TOSHIBA

64-Bit TX System RISC
TX49 Family
TX4939

Rev. 3.1

TOSHIBA CORPORATION

R4000/R4400/R5000 are a trademark of MIPS Technologies, Inc.

The information contained herein is subject to change without notice.

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others.

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.
It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..

The Toshiba products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).
These Toshiba products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of Toshiba products listed in this document shall be made at the customer's own risk.

The products described in this document may include products subject to the foreign exchange and foreign trade laws.

The products described in this document contain components made in the United States and subject to export control of the U.S. authorities. Diversion contrary to the U.S. law is prohibited.

TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law and regulations.

Preface

Thank you for new or continued patronage of TOSHIBA semiconductor products. This is the 2005 edition of the user's manual for the TX4939 64-bit RISC microprocessor.

This databook is written so as to be accessible to engineers who may be designing a TOSHIBA microprocessor into their products for the first time. No prior knowledge of this device is assumed. What we offer here is basic information about the microprocessor, a discussion of the application fields in which the microprocessor is utilized, and an overview of design methods. On the other hand, the more experienced designer will find complete technical specifications for this product.

Toshiba continually updates its technical information. Your comments and suggestions concerning this and other Toshiba documents are sincerely appreciated and may be utilized in subsequent editions. For updating of the data in this manual, or for additional information about the product appearing in it, please contact your nearest Toshiba office or authorized Toshiba dealer.

November 2005

List of Contents

List of Figures

List of Tables

Handling Precautions

1. Using Toshiba Semiconductors Safely

TOSHIBA are continually working to improve the quality and the reliability of their products.

Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property.

In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.

The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer’s own risk.

2. Safety Precautions

This section lists important precautions which users of semiconductor devices (and anyone else) should observe in order to avoid injury and damage to property, and to ensure safe and correct use of devices.

Please be sure that you understand the meanings of the labels and the graphic symbol described below before you move on to the detailed descriptions of the precautions.

[Explanation of labels]

⚠DANGER	Indicates an imminently hazardous situation which will result in death or serious injury if you do not follow instructions.
⚠WARNING	Indicates a potentially hazardous situation which could result in death or serious injury if you do not follow instructions.
⚠CAUTION	Indicates a potentially hazardous situation which if not avoided, may result in minor injury or moderate injury.

[Explanation of graphic symbol]

Graphic symbol	Meaning
	Indicates that caution is required (laser beam is dangerous to eyes).

2.1 General Precautions regarding Semiconductor Devices

⚠CAUTION
Do not use devices under conditions exceeding their absolute maximum ratings (e.g. current, voltage, power dissipation or temperature). This may cause the device to break down, degrade its performance, or cause it to catch fire or explode resulting in injury. Do not insert devices in the wrong orientation. Make sure that the positive and negative terminals of power supplies are connected correctly. Otherwise the rated maximum current or power dissipation may be exceeded and the device may break down or undergo performance degradation, causing it to catch fire or explode and resulting in injury. When power to a device is on, do not touch the device's heat sink. Heat sinks become hot, so you may burn your hand. Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger. When conducting any kind of evaluation, inspection or testing, be sure to connect the testing equipment's electrodes or probes to the pins of the device under test before powering it on. Otherwise, you may receive an electric shock causing injury. Before grounding an item of measuring equipment or a soldering iron, check that there is no electrical leakage from it. Electrical leakage may cause the device which you are testing or soldering to break down, or could give you an electric shock. Always wear protective glasses when cutting the leads of a device with clippers or a similar tool. If you do not, small bits of metal flying off the cut ends may damage your eyes.

2.2 Precautions Specific to Each Product Group

2.2.1 Optical semiconductor devices

⚠ DANGER

When a visible semiconductor laser is operating, do not look directly into the laser beam or look through the optical system. This is highly likely to impair vision, and in the worst case may cause blindness.
If it is necessary to examine the laser apparatus, for example to inspect its optical characteristics, always wear the appropriate type of laser protective glasses as stipulated by IEC standard IEC825-1.

⚠ WARNING

Ensure that the current flowing in an LED device does not exceed the device's maximum rated current.
This is particularly important for resin-packaged LED devices, as excessive current may cause the package resin to blow up, scattering resin fragments and causing injury.

When testing the dielectric strength of a photocoupler, use testing equipment which can shut off the supply voltage to the photocoupler. If you detect a leakage current of more than 100 μA, use the testing equipment to shut off the photocoupler's supply voltage; otherwise a large short-circuit current will flow continuously, and the device may break down or burst into flames, resulting in fire or injury.

When incorporating a visible semiconductor laser into a design, use the device's internal photodetector or a separate photodetector to stabilize the laser's radiant power so as to ensure that laser beams exceeding the laser's rated radiant power cannot be emitted.
If this stabilizing mechanism does not work and the rated radiant power is exceeded, the device may break down or the excessively powerful laser beams may cause injury.

2.2.2 Power devices

⚠ DANGER

Never touch a power device while it is powered on. Also, after turning off a power device, do not touch it until it has thoroughly discharged all remaining electrical charge.
Touching a power device while it is powered on or still charged could cause a severe electric shock, resulting in death or serious injury.

When conducting any kind of evaluation, inspection or testing, be sure to connect the testing equipment's electrodes or probes to the device under test before powering it on.
When you have finished, discharge any electrical charge remaining in the device.
Connecting the electrodes or probes of testing equipment to a device while it is powered on may result in electric shock, causing injury.

⚠WARNING

Do not use devices under conditions which exceed their absolute maximum ratings (current, voltage, power dissipation, temperature etc.).
This may cause the device to break down, causing a large short-circuit current to flow, which may in turn cause it to catch fire or explode, resulting in fire or injury.

Use a unit which can detect short-circuit currents and which will shut off the power supply if a short-circuit occurs.
If the power supply is not shut off, a large short-circuit current will flow continuously, which may in turn cause the device to catch fire or explode, resulting in fire or injury.

When designing a case for enclosing your system, consider how best to protect the user from shrapnel in the event of the device catching fire or exploding.
Flying shrapnel can cause injury.

When conducting any kind of evaluation, inspection or testing, always use protective safety tools such as a cover for the device.
Otherwise you may sustain injury caused by the device catching fire or exploding.

Make sure that all metal casings in your design are grounded to earth.
Even in modules where a device's electrodes and metal casing are insulated, capacitance in the module may cause the electrostatic potential in the casing to rise.
Dielectric breakdown may cause a high voltage to be applied to the casing, causing electric shock and injury to anyone touching it.

When designing the heat radiation and safety features of a system incorporating high-speed rectifiers, remember to take the device's forward and reverse losses into account.
The leakage current in these devices is greater than that in ordinary rectifiers; as a result, if a high-speed rectifier is used in an extreme environment (e.g. at high temperature or high voltage), its reverse loss may increase, causing thermal runaway to occur.
This may in turn cause the device to explode and scatter shrapnel, resulting in injury to the user.

A design should ensure that, except when the main circuit of the device is active, reverse bias is applied to the device gate while electricity is conducted to control circuits, so that the main circuit will become inactive.
Malfunction of the device may cause serious accidents or injuries.

⚠CAUTION

When conducting any kind of evaluation, inspection or testing, either wear protective gloves or wait until the device has cooled properly before handling it.
Devices become hot when they are operated. Even after the power has been turned off, the device will retain residual heat which may cause a burn to anyone touching it.

2.2.3 Bipolar ICs (for use in automobiles)

⚠CAUTION

If your design includes an inductive load such as a motor coil, incorporate diodes or similar devices into the design to prevent negative current from flowing in.
The load current generated by powering the device on and off may cause it to function erratically or to break down, which could in turn cause injury.

Ensure that the power supply to any device which incorporates protective functions is stable.
If the power supply is unstable, the device may operate erratically, preventing the protective functions from working correctly. If protective functions fail, the device may break down causing injury to the user.

3. General Safety Precautions and Usage Considerations

This section is designed to help you gain a better understanding of semiconductor devices, so as to ensure the safety, quality and reliability of the devices which you incorporate into your designs.

3.1 From Incoming to Shipping

3.1.1 Electrostatic discharge (ESD)

When handling individual devices (which are not yet mounted on a printed circuit board), be sure that the environment is protected against electrostatic electricity. Operators should wear anti-static clothing, and containers and other objects which come into direct contact with devices should be made of anti-static materials and should be grounded to earth via an 0.5- to 1.0-MΩ protective resistor.

Please follow the precautions described below; this is particularly important for devices which are marked “Be careful of static.”.

(1) Work environment

- When humidity in the working environment decreases, the human body and other insulators can easily become charged with static electricity due to friction. Maintain the recommended humidity of 40% to 60% in the work environment, while also taking into account the fact that moisture-proof-packed products may absorb moisture after unpacking.

- Be sure that all equipment, jigs and tools in the working area are grounded to earth.

- Place a conductive mat over the floor of the work area, or take other appropriate measures, so that the floor surface is protected against static electricity and is grounded to earth. The surface resistivity should be 10^4 to 10^8 Ω/sq and the resistance between surface and ground, 7.5×10^5 to 10^8 Ω

- Cover the workbench surface also with a conductive mat (with a surface resistivity of 10^4 to 10^8 Ω/sq, for a resistance between surface and ground of 7.5×10^5 to 10^8 Ω) . The purpose of this is to disperse static electricity on the surface (through resistive components) and ground it to earth. Workbench surfaces must not be constructed of low-resistance metallic materials that allow rapid static discharge when a charged device touches them directly.

- Pay attention to the following points when using automatic equipment in your workplace:

  (a) When picking up ICs with a vacuum unit, use a conductive rubber fitting on the end of the pick-up wand to protect against electrostatic charge.

  (b) Minimize friction on IC package surfaces. If some rubbing is unavoidable due to the device’s mechanical structure, minimize the friction plane or use material with a small friction coefficient and low electrical resistance. Also, consider the use of an ionizer.

  (c) In sections which come into contact with device lead terminals, use a material which dissipates static electricity.

  (d) Ensure that no statically charged bodies (such as work clothes or the human body) touch the devices.

(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

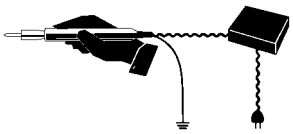
(f) Make sure that jigs and tools used in the assembly process do not touch devices.

(g) In processes in which packages may retain an electrostatic charge, use an ionizer to neutralize the ions.

- Make sure that CRT displays in the working area are protected against static charge, for example by a VDT filter. As much as possible, avoid turning displays on and off. Doing so can cause electrostatic induction in devices.
- Keep track of charged potential in the working area by taking periodic measurements.
- Ensure that work chairs are protected by an anti-static textile cover and are grounded to the floor surface by a grounding chain. (Suggested resistance between the seat surface and grounding chain is 7.5×10^5 to $10^{12}\Omega$.)
- Install anti-static mats on storage shelf surfaces. (Suggested surface resistivity is 10^4 to 10^8 Ω/sq; suggested resistance between surface and ground is 7.5×10^5 to 10^8 Ω.)
- For transport and temporary storage of devices, use containers (boxes, jigs or bags) that are made of anti-static materials or materials which dissipate electrostatic charge.
- Make sure that cart surfaces which come into contact with device packaging are made of materials which will conduct static electricity, and verify that they are grounded to the floor surface via a grounding chain.
- In any location where the level of static electricity is to be closely controlled, the ground resistance level should be Class 3 or above. Use different ground wires for all items of equipment which may come into physical contact with devices.

(2) Operating environment

- Operators must wear anti-static clothing and conductive shoes (or a leg or heel strap).
- Operators must wear a wrist strap grounded to earth via a resistor of about 1 MΩ.

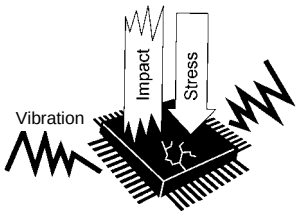


- Soldering irons must be grounded from iron tip to earth, and must be used only at low voltages (6 V to 24 V).
- If the tweezers you use are likely to touch the device terminals, use anti-static tweezers and in particular avoid metallic tweezers. If a charged device touches a low-resistance tool, rapid discharge can occur. When using vacuum tweezers, attach a conductive chucking pat to the tip, and connect it to a dedicated ground used especially for anti-static purposes (suggested resistance value: 10^4 to 10^8 Ω).
- Do not place devices or their containers near sources of strong electrical fields (such as above a CRT).

- When storing printed circuit boards which have devices mounted on them, use a board container or bag that is protected against static charge. To avoid the occurrence of static charge or discharge due to friction, keep the boards separate from one other and do not stack them directly on top of one another.
- Ensure, if possible, that any articles (such as clipboards) which are brought to any location where the level of static electricity must be closely controlled are constructed of anti-static materials.
- In cases where the human body comes into direct contact with a device, be sure to wear anti-static finger covers or gloves (suggested resistance value: 10^8 Ω or less).
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.
- If a wrist strap cannot be used for some reason, and there is a possibility of imparting friction to devices, use an ionizer.
- The transport film used in TCP products is manufactured from materials in which static charges tend to build up. When using these products, install an ionizer to prevent the film from being charged with static electricity. Also, ensure that no static electricity will be applied to the product's copper foils by taking measures to prevent static occuring in the peripheral equipment.

3.1.2 Vibration, impact and stress

Handle devices and packaging materials with care. To avoid damage to devices, do not toss or drop packages. Ensure that devices are not subjected to mechanical vibration or shock during transportation. Ceramic package devices and devices in canister-type packages which have empty space inside them are subject to damage from vibration and shock because the bonding wires are secured only at their ends.



Plastic molded devices, on the other hand, have a relatively high level of resistance to vibration and mechanical shock because their bonding wires are enveloped and fixed in resin. However, when any device or package type is installed in target equipment, it is to some extent susceptible to wiring disconnections and other damage from vibration, shock and stressed solder junctions. Therefore when devices are incorporated into the design of equipment which will be subject to vibration, the structural design of the equipment must be thought out carefully.

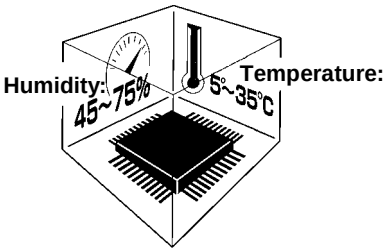
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

Furthermore, it is known that stress applied to a semiconductor device through the package changes the resistance characteristics of the chip because of piezoelectric effects. In analog circuit design attention must be paid to the problem of package stress as well as to the dangers of vibration and shock as described above.

3.2 Storage

3.2.1 General storage

- Avoid storage locations where devices will be exposed to moisture or direct sunlight.
- Follow the instructions printed on the device cartons regarding transportation and storage.
- The storage area temperature should be kept within a temperature range of 5°C to 35°C, and relative humidity should be maintained at between 45% and 75%.



- Do not store devices in the presence of harmful (especially corrosive) gases, or in dusty conditions.
- Use storage areas where there is minimal temperature fluctuation. Rapid temperature changes can cause moisture to form on stored devices, resulting in lead oxidation or corrosion. As a result, the solderability of the leads will be degraded.
- When repacking devices, use anti-static containers.
- Do not allow external forces or loads to be applied to devices while they are in storage.
- If devices have been stored for more than two years, their electrical characteristics should be tested and their leads should be tested for ease of soldering before they are used.

3.2.2 Moisture-proof packing

Moisture-proof packing should be handled with care. The handling procedure specified for each packing type should be followed scrupulously. If the proper procedures are not followed, the quality and reliability of devices may be degraded. This section describes general precautions for handling moisture-proof packing. Since the details may differ from device to device, refer also to the relevant individual datasheets or databook.

(1) General precautions
Follow the instructions printed on the device cartons regarding transportation and storage.

- Do not drop or toss device packing. The laminated aluminum material in it can be rendered ineffective by rough handling.
- The storage area temperature should be kept within a temperature range of 5°C to 30°C, and relative humidity should be maintained at 90% (max). Use devices within 12 months of the date marked on the package seal.

- If the 12-month storage period has expired, or if the 30% humidity indicator shown in Figure 1 is pink when the packing is opened, it may be advisable, depending on the device and packing type, to back the devices at high temperature to remove any moisture. Please refer to the table below. After the pack has been opened, use the devices in a 5°C to 30°C. 60% RH environment and within the effective usage period listed on the moisture-proof package. If the effective usage period has expired, or if the packing has been stored in a high-humidity environment, bake the devices at high temperature.

Packing	Moisture removal
Tray	If the packing bears the “Heatproof” marking or indicates the maximum temperature which it can withstand, bake at 125°C for 20 hours. (Some devices require a different procedure.)
Tube	Transfer devices to trays bearing the “Heatproof” marking or indicating the temperature which they can withstand, or to aluminum tubes before baking at 125°C for 20 hours.
Tape	Deviced packed on tape cannot be baked and must be used within the effective usage period after unpacking, as specified on the packing.

- When baking devices, protect the devices from static electricity.

- Moisture indicators can detect the approximate humidity level at a standard temperature of 25°C. 6-point indicators and 3-point indicators are currently in use, but eventually all indicators will be 3-point indicators.

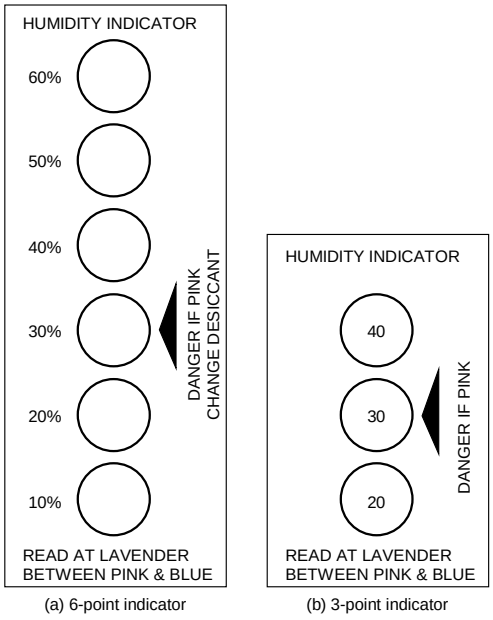


Figure 1 Humidity indicator

3.3 Design

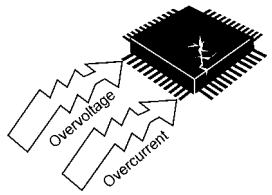
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will be used, including factors such as the ambient temperature, transient noise and voltage and current surges, as well as mounting conditions which affect device reliability. This section describes some general precautions which you should observe when designing circuits and when mounting devices on printed circuit boards.

For more detailed information about each product family, refer to the relevant individual technical datasheets available from Toshiba.

3.3.1 Absolute maximum ratings

⚠CAUTION Do not use devices under conditions in which their absolute maximum ratings (e.g. current, voltage, power dissipation or temperature) will be exceeded. A device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user.

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Although absolute maximum ratings differ from product to product, they essentially concern the voltage and current at each pin, the allowable power dissipation, and the junction and storage temperatures.



If the voltage or current on any pin exceeds the absolute maximum rating, the device's internal circuitry can become degraded. In the worst case, heat generated in internal circuitry can fuse wiring or cause the semiconductor chip to break down.

If storage or operating temperatures exceed rated values, the package seal can deteriorate or the wires can become disconnected due to the differences between the thermal expansion coefficients of the materials from which the device is constructed.

3.3.2 Recommended operating conditions

The recommended operating conditions for each device are those necessary to guarantee that the device will operate as specified in the datasheet.
If greater reliability is required, derate the device's absolute maximum ratings for voltage, current, power and temperature before using it.

3.3.3 Derating

When incorporating a device into your design, reduce its rated absolute maximum voltage, current, power dissipation and operating temperature in order to ensure high reliability.
Since derating differs from application to application, refer to the technical datasheets available for the various devices used in your design.

3.3.4 Unused pins

If unused pins are left open, some devices can exhibit input instability problems, resulting in malfunctions such as abrupt increase in current flow. Similarly, if the unused output pins on a device are connected to the power supply pin, the ground pin or to other output pins, the IC may malfunction or break down.

Since the details regarding the handling of unused pins differ from device to device and from pin to pin, please follow the instructions given in the relevant individual datasheets or databook.

CMOS logic IC inputs, for example, have extremely high impedance. If an input pin is left open, it can easily pick up extraneous noise and become unstable. In this case, if the input voltage level reaches an intermediate level, it is possible that both the P-channel and N-channel transistors will be turned on, allowing unwanted supply current to flow. Therefore, ensure that the unused input pins of a device are connected to the power supply (Vcc) pin or ground (GND) pin of the same device. For details of what to do with the pins of heat sinks, refer to the relevant technical datasheet and databook.

3.3.5 Latch-up

Latch-up is an abnormal condition inherent in CMOS devices, in which Vcc gets shorted to ground. This happens when a parasitic PN-PN junction (thyristor structure) internal to the CMOS chip is turned on, causing a large current of the order of several hundred mA or more to flow between Vcc and GND, eventually causing the device to break down.

Latch-up occurs when the input or output voltage exceeds the rated value, causing a large current to flow in the internal chip, or when the voltage on the Vcc (Vdd) pin exceeds its rated value, forcing the internal chip into a breakdown condition. Once the chip falls into the latch-up state, even though the excess voltage may have been applied only for an instant, the large current continues to flow between Vcc (Vdd) and GND (Vss). This causes the device to heat up and, in extreme cases, to emit gas fumes as well. To avoid this problem, observe the following precautions:

(1) Do not allow voltage levels on the input and output pins either to rise above Vcc (Vdd) or to fall below GND (Vss). Also, follow any prescribed power-on sequence, so that power is applied gradually or in steps rather than abruptly.

(2) Do not allow any abnormal noise signals to be applied to the device.

(3) Set the voltage levels of unused input pins to Vcc (Vdd) or GND (Vss).

(4) Do not connect output pins to one another.

3.3.6 Input/Output protection

Wired-AND configurations, in which outputs are connected together, cannot be used, since this short-circuits the outputs. Outputs should, of course, never be connected to Vcc (Vdd) or GND (Vss).

Furthermore, ICs with tri-state outputs can undergo performance degradation if a shorted output current is allowed to flow for an extended period of time. Therefore, when designing circuits, make sure that tri-state outputs will not be enabled simultaneously.

3.3.7 Load capacitance

Some devices display increased delay times if the load capacitance is large. Also, large charging and discharging currents will flow in the device, causing noise. Furthermore, since outputs are shorted for a relatively long time, wiring can become fused.

Consult the technical information for the device being used to determine the recommended load capacitance.

3.3.8 Thermal design

The failure rate of semiconductor devices is greatly increased as operating temperatures increase. As shown in Figure 2, the internal thermal stress on a device is the sum of the ambient temperature and the temperature rise due to power dissipation in the device. Therefore, to achieve optimum reliability, observe the following precautions concerning thermal design:

(1) Keep the ambient temperature (Ta) as low as possible.

(2) If the device's dynamic power dissipation is relatively large, select the most appropriate circuit board material, and consider the use of heat sinks or of forced air cooling. Such measures will help lower the thermal resistance of the package.

(3) Derate the device's absolute maximum ratings to minimize thermal stress from power dissipation.
$\theta ja = \theta jc + \theta ca$
$\theta ja = (Tj - Ta) / P$
$\theta jc = (Tj - Tc) / P$
$\theta ca = (Tc - Ta) / P$
in which θja = thermal resistance between junction and surrounding air (°C/W)
θjc = thermal resistance between junction and package surface, or internal thermal resistance (°C/W)
θca = thermal resistance between package surface and surrounding air, or external thermal resistance (°C/W)
Tj = junction temperature or chip temperature (°C)
Tc = package surface temperature or case temperature (°C)
Ta = ambient temperature (°C)
P = power dissipation (W)

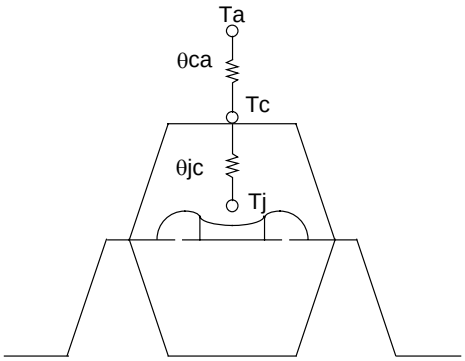


Figure 2 Thermal resistance of package

3.3.9 Interfacing

When connecting inputs and outputs between devices, make sure input voltage (V_{IL}/V_{IH}) and output voltage (V_{OL}/V_{OH}) levels are matched. Otherwise, the devices may malfunction. When connecting devices operating at different supply voltages, such as in a dual-power-supply system, be aware that erroneous power-on and power-off sequences can result in device breakdown. For details of how to interface particular devices, consult the relevant technical datasheets and databooks. If you have any questions or doubts about interfacing, contact your nearest Toshiba office or distributor.

3.3.10 Decoupling

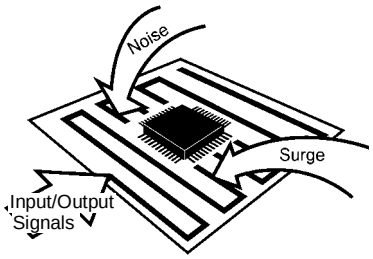
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with respect to high frequencies must be kept low. This can be accomplished by using thick and short wiring for the Vcc (Vdd) and GND (Vss) lines and by installing decoupling capacitors (of approximately 0.01 μF to 1 μF capacitance) as high-frequency filters between Vcc (Vdd) and GND (Vss) at strategic locations on the printed circuit board.

For low-frequency filtering, it is a good idea to install a 10- to 100-μF capacitor on the printed circuit board (one capacitor will suffice). If the capacitance is excessively large, however, (e.g. several thousand μF) latch-up can be a problem. Be sure to choose an appropriate capacitance value.

An important point about wiring is that, in the case of high-speed logic ICs, noise is caused mainly by reflection and crosstalk, or by the power supply impedance. Reflections cause increased signal delay, ringing, overshoot and undershoot, thereby reducing the device's safety margins with respect to noise. To prevent reflections, reduce the wiring length by increasing the device mounting density so as to lower the inductance (L) and capacitance (C) in the wiring. Extreme care must be taken, however, when taking this corrective measure, since it tends to cause crosstalk between the wires. In practice, there must be a trade-off between these two factors.

3.3.11 External noise

Printed circuit boards with long I/O or signal pattern lines are vulnerable to induced noise or surges from outside sources. Consequently, malfunctions or breakdowns can result from overcurrent or overvoltage, depending on the types of device used. To protect against noise, lower the impedance of the pattern line or insert a noise-canceling circuit. Protective measures must also be taken against surges.



For details of the appropriate protective measures for a particular device, consult the relevant databook.

3.3.12 Electromagnetic interference

Widespread use of electrical and electronic equipment in recent years has brought with it radio and TV reception problems due to electromagnetic interference. To use the radio spectrum effectively and to maintain radio communications quality, each country has formulated regulations limiting the amount of electromagnetic interference which can be generated by individual products.

Electromagnetic interference includes conduction noise propagated through power supply and telephone lines, and noise from direct electromagnetic waves radiated by equipment. Different measurement methods and corrective measures are used to assess and counteract each specific type of noise.

Difficulties in controlling electromagnetic interference derive from the fact that there is no method available which allows designers to calculate, at the design stage, the strength of the electromagnetic waves which will emanate from each component in a piece of equipment. For this reason, it is only after the prototype equipment has been completed that the designer can take measurements using a dedicated instrument to determine the strength of electromagnetic interference waves. Yet it is possible during system design to incorporate some measures for the prevention of electromagnetic interference, which can facilitate taking corrective measures once the design has been completed. These include installing shields and noise filters, and increasing the thickness of the power supply wiring patterns on the printed circuit board. One effective method, for example, is to devise several shielding options during design, and then select the most suitable shielding method based on the results of measurements taken after the prototype has been completed.

3.3.13 Peripheral circuits

In most cases semiconductor devices are used with peripheral circuits and components. The input and output signal voltages and currents in these circuits must be chosen to match the semiconductor device's specifications. The following factors must be taken into account.

(1) Inappropriate voltages or currents applied to a device's input pins may cause it to operate erratically. Some devices contain pull-up or pull-down resistors. When designing your system, remember to take the effect of this on the voltage and current levels into account.

(2) The output pins on a device have a predetermined external circuit drive capability. If this drive capability is greater than that required, either incorporate a compensating circuit into your design or carefully select suitable components for use in external circuits.

3.3.14 Safety standards

Each country has safety standards which must be observed. These safety standards include requirements for quality assurance systems and design of device insulation. Such requirements must be fully taken into account to ensure that your design conforms to the applicable safety standards.

3.3.15 Other precautions

(1) When designing a system, be sure to incorporate fail-safe and other appropriate measures according to the intended purpose of your system. Also, be sure to debug your system under actual board-mounted conditions.

(2) If a plastic-package device is placed in a strong electric field, surface leakage may occur due to the charge-up phenomenon, resulting in device malfunction. In such cases take appropriate measures to prevent this problem, for example by protecting the package surface with a conductive shield.

(3) With some microcomputers and MOS memory devices, caution is required when powering on or resetting the device. To ensure that your design does not violate device specifications, consult the relevant databook for each constituent device.

(4) Ensure that no conductive material or object (such as a metal pin) can drop onto and short the leads of a device mounted on a printed circuit board.

3.4 Inspection, Testing and Evaluation

3.4.1 Grounding

⚠CAUTION Ground all measuring instruments, jigs, tools and soldering irons to earth.
Electrical leakage may cause a device to break down or may result in electric shock.

3.4.2 Inspection Sequence

⚠CAUTION

① Do not insert devices in the wrong orientation. Make sure that the positive and negative electrodes of the power supply are correctly connected. Otherwise, the rated maximum current or maximum power dissipation may be exceeded and the device may break down or undergo performance degradation, causing it to catch fire or explode, resulting in injury to the user.

② When conducting any kind of evaluation, inspection or testing using AC power with a peak voltage of 42.4 V or DC power exceeding 60 V, be sure to connect the electrodes or probes of the testing equipment to the device under test before powering it on. Connecting the electrodes or probes of testing equipment to a device while it is powered on may result in electric shock, causing injury.

(1) Apply voltage to the test jig only after inserting the device securely into it. When applying or removing power, observe the relevant precautions, if any.

(2) Make sure that the voltage applied to the device is off before removing the device from the test jig. Otherwise, the device may undergo performance degradation or be destroyed.

(3) Make sure that no surge voltages from the measuring equipment are applied to the device.

(4) The chips housed in tape carrier packages (TCPs) are bare chips and are therefore exposed. During inspection take care not to crack the chip or cause any flaws in it.
Electrical contact may also cause a chip to become faulty. Therefore make sure that nothing comes into electrical contact with the chip.

3.5 Mounting

There are essentially two main types of semiconductor device package: lead insertion and surface mount. During mounting on printed circuit boards, devices can become contaminated by flux or damaged by thermal stress from the soldering process. With surface-mount devices in particular, the most significant problem is thermal stress from solder reflow, when the entire package is subjected to heat. This section describes a recommended temperature profile for each mounting method, as well as general precautions which you should take when mounting devices on printed circuit boards. Note, however, that even for devices with the same package type, the appropriate mounting method varies according to the size of the chip and the size and shape of the lead frame. Therefore, please consult the relevant technical datasheet and databook.

3.5.1 Lead forming

⚠CAUTION

① Always wear protective glasses when cutting the leads of a device with clippers or a similar tool. If you do not, small bits of metal flying off the cut ends may damage your eyes.

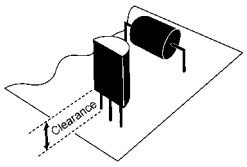
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

Semiconductor devices must undergo a process in which the leads are cut and formed before the devices can be mounted on a printed circuit board. If undue stress is applied to the interior of a device during this process, mechanical breakdown or performance degradation can result. This is attributable primarily to differences between the stress on the device's external leads and the stress on the internal leads. If the relative difference is great enough, the device's internal leads, adhesive properties or sealant can be damaged. Observe these precautions during the lead-forming process (this does not apply to surface-mount devices):

(1) Lead insertion hole intervals on the printed circuit board should match the lead pitch of the device precisely.

(2) If lead insertion hole intervals on the printed circuit board do not precisely match the lead pitch of the device, do not attempt to forcibly insert devices by pressing on them or by pulling on their leads.

(3) For the minimum clearance specification between a device and a printed circuit board, refer to the relevant device's datasheet and databook. If necessary, achieve the required clearance by forming the device's leads appropriately. Do not use the spacers which are used to raise devices above the surface of the printed circuit board during soldering to achieve clearance. These spacers normally continue to expand due to heat, even after the solder has begun to solidify; this applies severe stress to the device.



(4) Observe the following precautions when forming the leads of a device prior to mounting.

- Use a tool or jig to secure the lead at its base (where the lead meets the device package) while bending so as to avoid mechanical stress to the device. Also avoid bending or stretching device leads repeatedly.
- Be careful not to damage the lead during lead forming.
- Follow any other precautions described in the individual datasheets and databooks for each device and package type.

3.5.2 Socket mounting

(1) When socket mounting devices on a printed circuit board, use sockets which match the inserted device's package.

(2) Use sockets whose contacts have the appropriate contact pressure. If the contact pressure is insufficient, the socket may not make a perfect contact when the device is repeatedly inserted and removed; if the pressure is excessively high, the device leads may be bent or damaged when they are inserted into or removed from the socket.

(3) When soldering sockets to the printed circuit board, use sockets whose construction prevents flux from penetrating into the contacts or which allows flux to be completely cleaned off.

(4) Make sure the coating agent applied to the printed circuit board for moisture-proofing purposes does not stick to the socket contacts.

(5) If the device leads are severely bent by a socket as it is inserted or removed and you wish to repair the leads so as to continue using the device, make sure that this lead correction is only performed once. Do not use devices whose leads have been corrected more than once.

(6) If the printed circuit board with the devices mounted on it will be subjected to vibration from external sources, use sockets which have a strong contact pressure so as to prevent the sockets and devices from vibrating relative to one another.

3.5.3 Soldering temperature profile

The soldering temperature and heating time vary from device to device. Therefore, when specifying the mounting conditions, refer to the individual datasheets and databooks for the devices used.

(1) Using medium infrared ray reflow

- Heating top and bottom with long or medium infrared rays is recommended (see Figure 3).

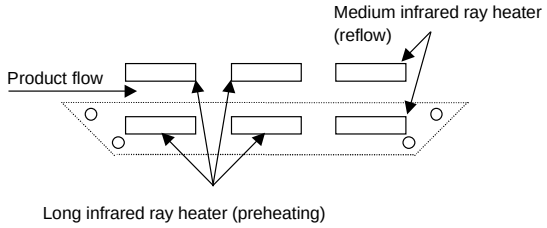


Figure 3 Heating top and bottom with long or medium infrared rays

- Complete the infrared ray reflow process for 30 to 50 seconds at a package surface temperature of between 230°C and 260°C.
- Refer to Figure 4 for an example of a good temperature profile for infrared or hot air reflow.

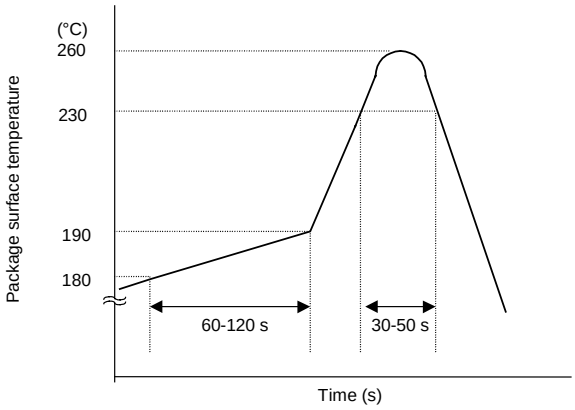


Figure 4 Sample temperature profile (Pb free) for infrared or hot air reflow

(2) Using hot air reflow

- Complete hot air reflow for 30 to 50 seconds at a package surface temperature of between 230°C and 260°C.
- For an example of a recommended temperature profile, refer to Figure 4 above.

3.5.4 Flux cleaning and ultrasonic cleaning

(1) When cleaning circuit boards to remove flux, make sure that no residual reactive ions such as Na or Cl remain. Note that organic solvents react with water to generate hydrogen chloride and other corrosive gases which can degrade device performance.

(2) Washing devices with water will not cause any problems. However, make sure that no reactive ions such as sodium and chlorine are left as a residue. Also, be sure to dry devices sufficiently after washing.

(3) Do not rub device markings with a brush or with your hand during cleaning or while the devices are still wet from the cleaning agent. Doing so can rub off the markings.

(4) The dip cleaning, shower cleaning and steam cleaning processes all involve the chemical action of a solvent. Use only recommended solvents for these cleaning methods. When immersing devices in a solvent or steam bath, make sure that the temperature of the liquid is 50°C or below, and that the circuit board is removed from the bath within one minute.

(5) Ultrasonic cleaning should not be used with hermetically-sealed ceramic packages such as a leadless chip carrier (LCC), pin grid array (PGA) or charge-coupled device (CCD), because the bonding wires can become disconnected due to resonance during the cleaning process. Even if a device package allows ultrasonic cleaning, limit the duration of ultrasonic cleaning to as short a time as possible, since long hours of ultrasonic cleaning degrade the adhesion between the mold resin and the frame material. The following ultrasonic cleaning conditions are recommended:

Frequency: 27 kHz ~ 29 kHz

Ultrasonic output power: 300 W or less (0.25 W/cm^2 or less)

Cleaning time: 30 seconds or less

Suspend the circuit board in the solvent bath during ultrasonic cleaning in such a way that the ultrasonic vibrator does not come into direct contact with the circuit board or the device.

3.5.5 No cleaning

If analog devices or high-speed devices are used without being cleaned, flux residues may cause minute amounts of leakage between pins. Similarly, dew condensation, which occurs in environments containing residual chlorine when power to the device is on, may cause between-lead leakage or migration. Therefore, Toshiba recommends that these devices be cleaned.
However, if the flux used contains only a small amount of halogen (0.05W% or less), the devices may be used without cleaning without any problems. No cleaning is recommended for TX4938.

3.5.6 Mounting tape carrier packages (TCPs)

(1) When tape carrier packages (TCPs) are mounted, measures must be taken to prevent electrostatic breakdown of the devices.

(2) If devices are being picked up from tape, or outer lead bonding (OLB) mounting is being carried out, consult the manufacturer of the insertion machine which is being used, in order to establish the optimum mounting conditions in advance and to avoid any possible hazards.

(3) The base film, which is made of polyimide, is hard and thin. Be careful not to cut or scratch your hands or any objects while handling the tape.

(4) When punching tape, try not to scatter broken pieces of tape too much.

(5) Treat the extra film, reels and spacers left after punching as industrial waste, taking care not to destroy or pollute the environment.

(6) Chips housed in tape carrier packages (TCPs) are bare chips and therefore have their reverse side exposed. To ensure that the chip will not be cracked during mounting, ensure that no mechanical shock is applied to the reverse side of the chip. Electrical contact may also cause a chip to fail. Therefore, when mounting devices, make sure that nothing comes into electrical contact with the reverse side of the chip.
If your design requires connecting the reverse side of the chip to the circuit board, please consult Toshiba or a Toshiba distributor beforehand.

3.5.7 Mounting chips

Devices delivered in chip form tend to degrade or break under external forces much more easily than plastic-packaged devices. Therefore, caution is required when handling this type of device.

(1) Mount devices in a properly prepared environment so that chip surfaces will not be exposed to polluted ambient air or other polluted substances.

(2) When handling chips, be careful not to expose them to static electricity.
In particular, measures must be taken to prevent static damage during the mounting of chips. With this in mind, Toshiba recommend mounting all peripheral parts first and then mounting chips last (after all other components have been mounted).

(3) Make sure that PCBs (or any other kind of circuit board) on which chips are being mounted do not have any chemical residues on them (such as the chemicals which were used for etching the PCBs).

(4) When mounting chips on a board, use the method of assembly that is most suitable for maintaining the appropriate electrical, thermal and mechanical properties of the semiconductor devices used.

* For details of devices in chip form, refer to the relevant device's individual datasheets.

3.5.8 Circuit board coating

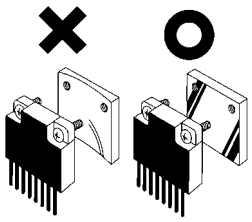
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects that may result and then choose the coating resin which results in the minimum level of stress to the device.

3.5.9 Heat sinks

(1) When attaching a heat sink to a device, be careful not to apply excessive force to the device in the process.

(2) When attaching a device to a heat sink by fixing it at two or more locations, evenly tighten all the screws in stages (i.e. do not fully tighten one screw while the rest are still only loosely tightened). Finally, fully tighten all the screws up to the specified torque.

(3) Drill holes for screws in the heat sink exactly as specified. Smooth the surface by removing burrs and protrusions or indentations which might interfere with the installation of any part of the device.



(4) A coating of silicone compound can be applied between the heat sink and the device to improve heat conductivity. Be sure to apply the coating thinly and evenly; do not use too much. Also, be sure to use a non-volatile compound, as volatile compounds can crack after a time, causing the heat radiation properties of the heat sink to deteriorate.

(5) If the device is housed in a plastic package, use caution when selecting the type of silicone compound to be applied between the heat sink and the device. With some types, the base oil separates and penetrates the plastic package, significantly reducing the useful life of the device.
Two recommended silicone compounds in which base oil separation is not a problem are YG6260 from Toshiba Silicone.

(6) Heat-sink-equipped devices can become very hot during operation. Do not touch them, or you may sustain a burn.

3.5.10 Tightening torque

(1) Make sure the screws are tightened with fastening torques not exceeding the torque values stipulated in individual datasheets and databooks for the devices used.

(2) Do not allow a power screwdriver (electrical or air-driven) to touch devices.

3.5.11 Repeated device mounting and usage

Do not remount or re-use devices which fall into the categories listed below; these devices may cause significant problems relating to performance and reliability.

(1) Devices which have been removed from the board after soldering

(2) Devices which have been inserted in the wrong orientation or which have had reverse current applied

(3) Devices which have undergone lead forming more than once

3.6 Protecting Devices in the Field

3.6.1 Temperature

Semiconductor devices are generally more sensitive to temperature than are other electronic components. The various electrical characteristics of a semiconductor device are dependent on the ambient temperature at which the device is used. It is therefore necessary to understand the temperature characteristics of a device and to incorporate device derating into circuit design. Note also that if a device is used above its maximum temperature rating, device deterioration is more rapid and it will reach the end of its usable life sooner than expected.

3.6.2 Humidity

Resin-molded devices are sometimes improperly sealed. When these devices are used for an extended period of time in a high-humidity environment, moisture can penetrate into the device and cause chip degradation or malfunction. Furthermore, when devices are mounted on a regular printed circuit board, the impedance between wiring components can decrease under high-humidity conditions. In systems which require a high signal-source impedance, circuit board leakage or leakage between device lead pins can cause malfunctions. The application of a moisture-proof treatment to the device surface should be considered in this case. On the other hand, operation under low-humidity conditions can damage a device due to the occurrence of electrostatic discharge. Unless damp-proofing measures have been specifically taken, use devices only in environments with appropriate ambient moisture levels (i.e. within a relative humidity range of 40% to 60%).

3.6.3 Corrosive gases

Corrosive gases can cause chemical reactions in devices, degrading device characteristics.
For example, sulphur-bearing corrosive gases emanating from rubber placed near a device (accompanied by condensation under high-humidity conditions) can corrode a device’s leads. The resulting chemical reaction between leads forms foreign particles which can cause electrical leakage.

3.6.4 Radioactive and cosmic rays

Most industrial and consumer semiconductor devices are not designed with protection against radioactive and cosmic rays. Devices used in aerospace equipment or in radioactive environments must therefore be shielded.

3.6.5 Strong electrical and magnetic fields

Devices exposed to strong magnetic fields can undergo a polarization phenomenon in their plastic material, or within the chip, which gives rise to abnormal symptoms such as impedance changes or increased leakage current. Failures have been reported in LSIs mounted near malfunctioning deflection yokes in TV sets. In such cases the device's installation location must be changed or the device must be shielded against the electrical or magnetic field. Shielding against magnetism is especially necessary for devices used in an alternating magnetic field because of the electromotive forces generated in this type of environment.

3.6.6 Interference from light (ultraviolet rays, sunlight, fluorescent lamps and incandescent lamps)

Light striking a semiconductor device generates electromotive force due to photoelectric effects. In some cases the device can malfunction. This is especially true for devices in which the internal chip is exposed. When designing circuits, make sure that devices are protected against incident light from external sources. This problem is not limited to optical semiconductors and EPROMs. All types of device can be affected by light.

3.6.7 Dust and oil

Just like corrosive gases, dust and oil can cause chemical reactions in devices, which will adversely affect a device's electrical characteristics. To avoid this problem, do not use devices in dusty or oily environments. This is especially important for optical devices because dust and oil can affect a device's optical characteristics as well as its physical integrity and the electrical performance factors mentioned above.

3.6.8 Fire

Semiconductor devices are combustible; they can emit smoke and catch fire if heated sufficiently. When this happens, some devices may generate poisonous gases. Devices should therefore never be used in close proximity to an open flame or a heat-generating body, or near flammable or combustible materials.

3.7 Disposal of Devices and Packing Materials

When discarding unused devices and packing materials, follow all procedures specified by local regulations in order to protect the environment against contamination.

4. Precautions and Usage Considerations

This section describes matters specific to each product group which need to be taken into consideration when using devices. If the same item is described in Sections 3 and 4, the description in Section 4 takes precedence.

4.1 Microcontrollers

4.1.1 Design

(1) Using resonators which are not specifically recommended for use

Resonators recommended for use with Toshiba products in microcontroller oscillator applications are listed in Toshiba databooks along with information about oscillation conditions. If you use a resonator not included in this list, please consult Toshiba or the resonator manufacturer concerning the suitability of the device for your application.

(2) Undefined functions

In some microcontrollers certain instruction code values do not constitute valid processor instructions. Also, it is possible that the values of bits in registers will become undefined. Take care in your applications not to use invalid instructions or to let register bit values become undefined.

TX4939

2005-11 Rev. 3.1

National Conventions Used in This Manual

Numeric Conventions

- Hexadecimal values are expressed as follows (example shown for decimal number 42):
  0x2A
- KB (kilobyte) = 1,024 bytes
  MB (megabyte) = 1,024 × 1,024 = 1,048,576 bytes
  GB (gigabyte) = 1,024 × 1,024 × 1,024 = 1,073,741,824 bytes

Data Conventions

- Byte: 8 bits
- Halfword: 2 consecutive bytes (16 bits)
- Word: 4 consecutive bytes (32 bits)
- Doubleword: 8 consecutive bytes (64 bits)

Signal Conventions

- Active-low signals are indicated by adding an asterisk (*) at the end of the signal name. (Example: RESET*)
- A signal is “asserted” when it is driven to the active voltage level. A signal is “deasserted” when it is driven to an inactive voltage level.

Register Conventions

- Properties of each bit in a register are expressed as follows.
  R: Read-only. Software cannot change the bit value.
  W: Write-only. The value of the bit is undefined if read.
  W1C Write 1 Clear. A write of 1 clears the bit, and a write of 0 has no effect.
  R/W: Read/Write
  R/W1C: Read/Write 1 Clear. The bit is both readable and writeable. A write of 1 clears the bit, and a write of 0 has no effect.
  R/W1S Read/Write 1 Set. The bit is both readable and writeable. A write of 1 sets the bit, and a write of 0 has no effect.
  R/W0C: Read/Write 0 Clear. The bit is both readable and writeable. A write of 0 clears the bit, and a write of 1 has no effect.
  RS/WC Read Set/Write Clear. The bit is both readable and writeable. The bit is set when read, and a write of an arbitrary value to the bit clears it.
  R/L: Read/Load. The value of this bit can only be changed through PCI configuration space programming, as described in Section 10.3.14, “Programming PCI Configuration Space Registers.”
- The notation <register name>.<bit/field name> is used to indicate a specific bit/field of a register.
  Example: CCFG.TOE
  CCFG.TOE refers to the Timeout Enable for Bus Error (TOE) field, located at bit 14 of the Chip Configuration Register (CCFG).

Handling of Reserved Bits and Registers

Reserved registers are not intended for general user access. Software should not write the reserved registers or use any values read from the reserved registers.

For all registers with fields marked as reserved, software should write a value of 0 to the reserved field and should not use any values read from the reserved fields.

Diagnostic Features

Any features labeled as “diagnostic features” are intended for evaluation purposes only, and the operation is not guaranteed.

References

64-Bit TX System RISC TX49/H2, TX49/H3, TX49/H4 Core Architecture

MIPS RISC Architecture, Gerry Kane and Joe Heinrich (ISBN 0-13-590472-2)

See MIPS Run, Dominic Sweetman (ISBN 1-55860-410-3)

MIPS Publications

PCI Local Bus Specification Revision 2.2

PCI Bus Power Management Interface Specification Revision 1.1

Audio CODEC ‘97 (AC ‘97) Revision 2.1

SmartMedia™ Physical Format Specifications Web-Online Version 1.00

SMIL (SmartMedia™ Interface Library) Software Edition Version 1.00

Chapter 1. Features

1.1. Abstract

The TX4939 is a micro controller for the 64-bit TX System RISC TX49 family. The TX4939 uses the TX49/H4 core as its CPU. The TX49/H4 core is a 64-bit RISC CPU core that Toshiba developed based on the R4000 architecture created by MIPS Technologies, Inc. (MIPS). Refer to the TX49 Core Architecture Manual for more information on the TX49/H4 core such as the Instruction set.

The TX4939 has as peripheral functions an External Bus Controller, a DDR SDRAM Controller, a PCI Controller, a DMA Controller, an Interrupt Controller, an AC-link Controller, Serial and Parallel ports, a Timer/Counter, and peripheral circuits such as Ethernet MAC, and RTC (Battery back-up capable).

1.2. Implemented Features

- *TX49/H4 Core (On-chip IEEE754 Compliant Single/Double Precision FPU)*
- *DDR SDRAM Controller (2 Channels: 32-bit/100-200MHz)*
- *External Bus Controller (8-bit,16-bit with 4-channels)*
- *NOR Flash Interface on External bus*
- *NAND Flash Controller*
- *32-bit PCI Controller (33 MHz/66 MHz) with 4 clock outputs, arbiter, and interrupts for 6 devices.*
- *PCI Boot and Satellite mode*
- *Direct Memory Access Controller (8 Channels [4 Channels are dedicated to ACLC])*
- *SIO (4 Channels, ch2 and ch3 are multiplexed with SPI)*
- *SPI (multiplexed with SIO ch2,ch3)*
- *8-bit Video Port (Synchronous Parallel Interface)*
- *Serial TS Video port (Max 3-ports)*
- *Timer/Counter (6 Channels)*
- *AC-link Controller / I2S(5.1ch) / I2S(2ch)*
- *Ether MAC with RMII (2 Channels)*
- *2-channel 100MB ATA IDE I/F*
- *Security Engine - DES/3DES/AES/MD5/SHA1/Modular Exponentiation Operation/Ex-OR for FEC*
- *Battery Backup RTC (48 bit linear counter, 250 bytes CMOS RAM)*
- *Extensive Power Management Feature*
- *Any Internal Controller can stop clock and keep RESET status.*
- *On-chip SRAM (2 KB)*
- *Interrupt controller (NMI, 4 external inputs for PCI and 3 external interrupt inputs for External Bus)*
- *Low Power Consumption (Typ. TBD)*
  The TX4939 operates at 1.2 V (Internal), 2.5V(DDR) and 3.3 V (I/O Block), and supports the low power consumption mode (Halt Mode).
- *Maximum Operating Frequency (for the CPU): 400 MHz*
- *IEEE1149.1 (JTAG) Support: Debugging Support Unit*
- *Built-in Clock Generator*
- *20 MHz Single Xtal Operation*
- *Generate All necessary Clock including Audio Sampling Clock Source*
- *Audio Sampling 96, 48, 44.1, 32, 24, 22.05, 16, 8 KHz (x512)*
- *Baud rate Clock source 14.7456 MHz*
- *Package*
  PBGA 456 pins: 1.00 mm ball pitch
  27 mm x 27 mm body size
  36 pins are thermal ball for heat dissipation

1.3. System Block Diagram

Figure 1-1 System Block Diagram shows TX4939 implemented features.

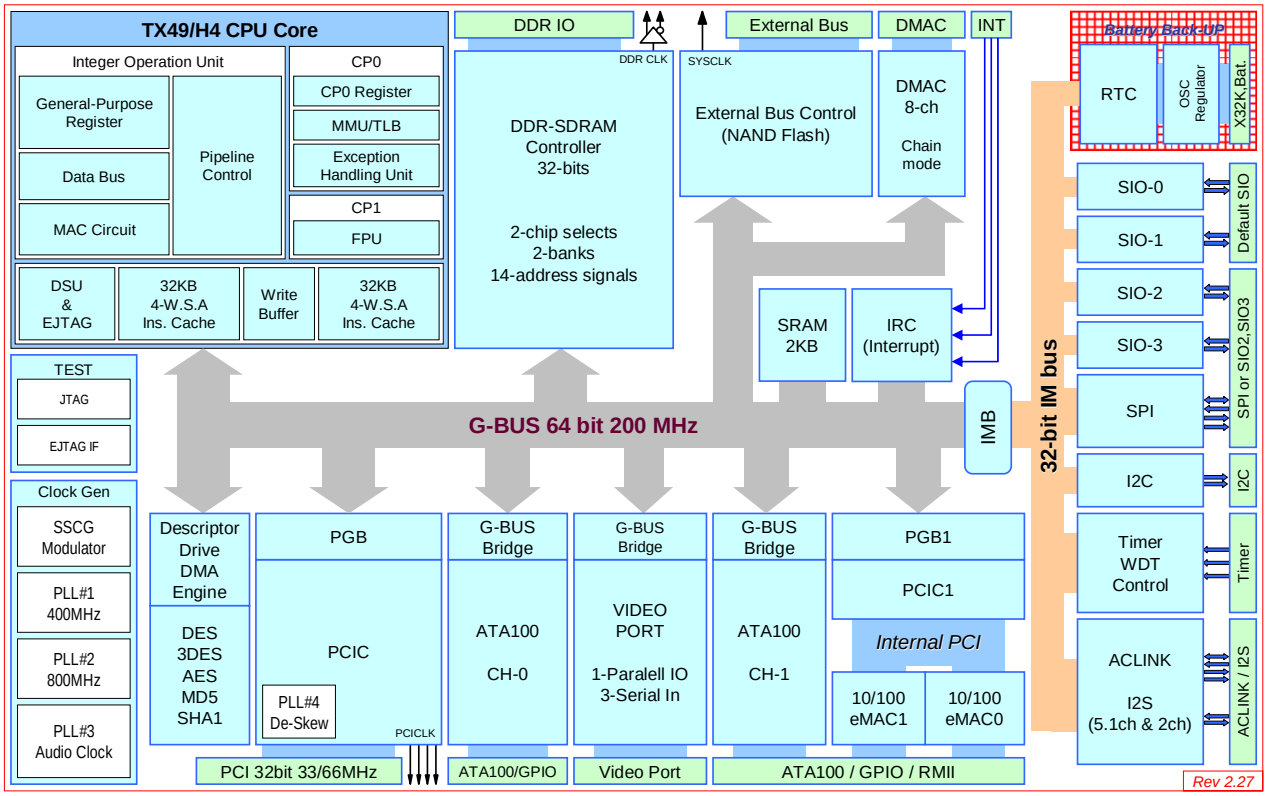


Figure 1-1 System Block Diagram

Due to pin limitation, some of features are not used simultaneously.

Table 1-1 Major Functional IPs shared by pin multiplexing.

Default Signal Name	Default Function	Primary Function	Secondary Function
ATA100-0	GPIO Ch-0 (30p)	ATA100/ATAPI CH-0	N/A
ATA100-1	GPIO Ch-1 (30p)	ATA100/ATAPI CH-1	Max. 2-ch 10/100 Ethernet MAC with RMII interface
Video Port	GPIO Ch-2 (17p)	1-Serial TS I/O, 1-Parallel TS/656 I/O	1-Serial TS I/O, 2-Serial TS Input
SPI Port	GPIO (4p)	SPI Interface	SIO2, SIO3
ACLINK	GPIO (6p)	ACLINK	I2S 2ch / I2S 5.1ch
I2C	GPIO (2p)	I2C	N/A

1

1.4. Example of Reference System

1.4.1. Two ATA100 for DVD Recorder

Figure 1-2 shows example system for high performance DVD recorder system. In this system, Ethernet port is an option. Dual ATA100 channels ensure enough through put for simultaneous DVD Recording and Playback.

One serial video port is used to receive TS stream from digital tuner or send TS stream other system.
One Parallel Video port is used to send OSD bitmap image to the display controller chip. By this way, OSD handling software will have broad portability independent from versions of display controller chip.

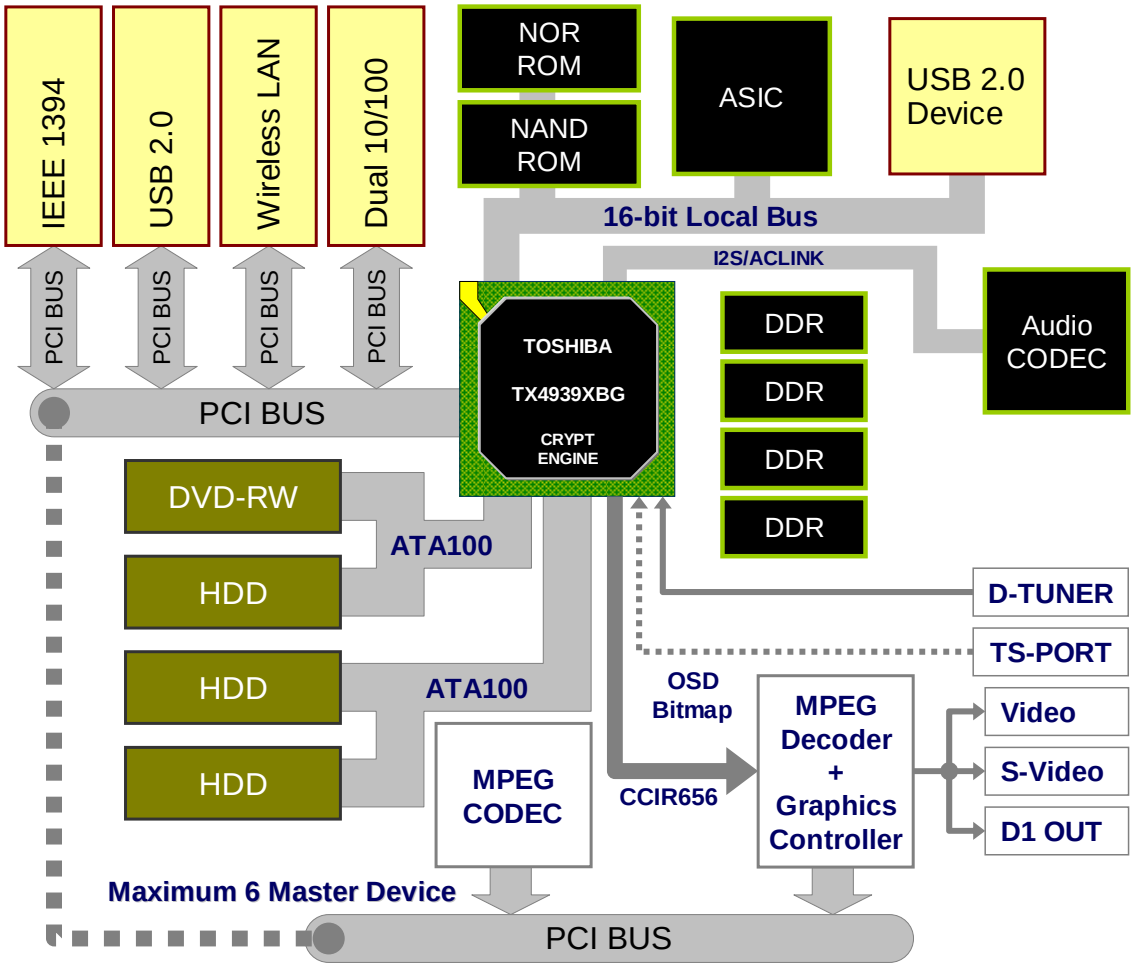


Figure 1-2 Two ATA100 for DVD Recorder

1.4.2. One ATA100 and Two Ethernet

Figure 1-3 shows typical multimedia station with HDD server function. Video contents can be distributed through various networks including bare TS stream device.

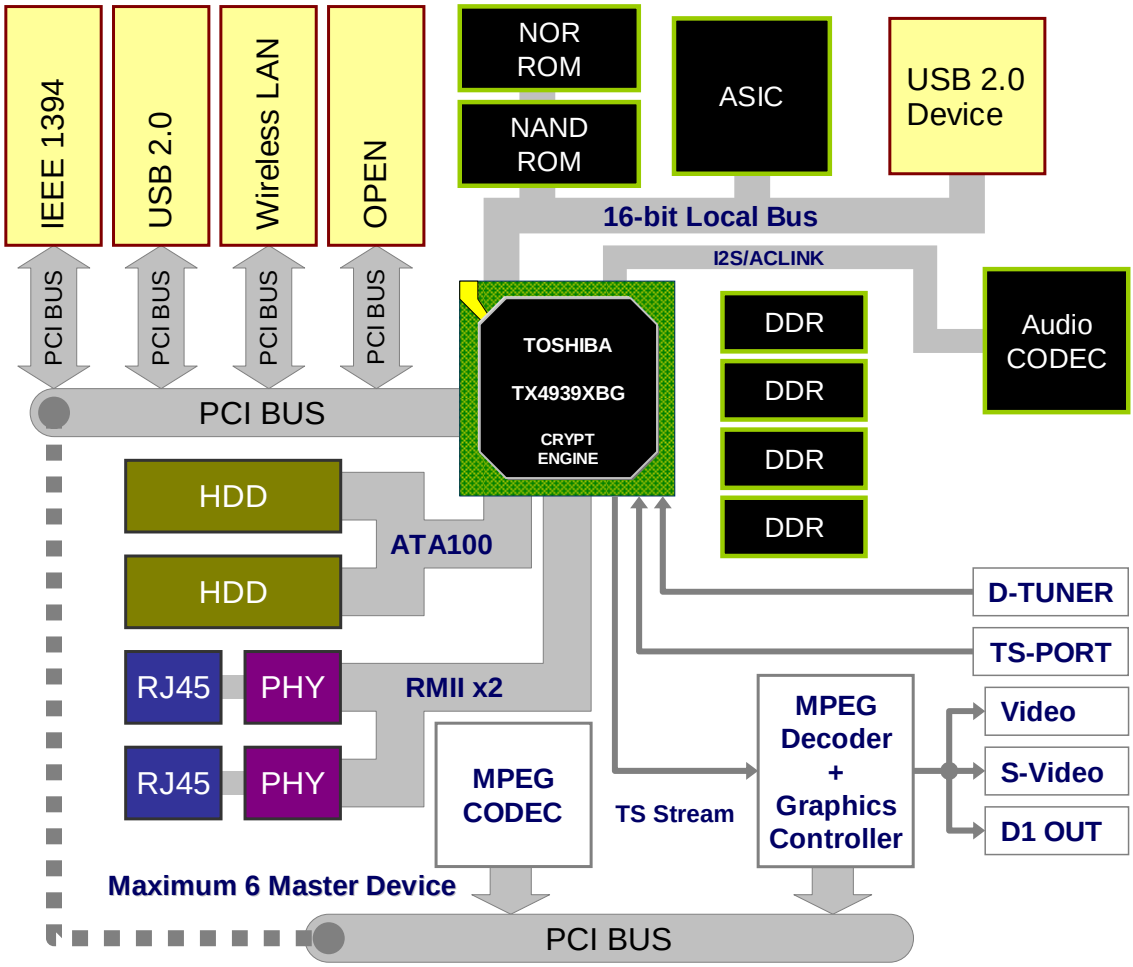


Figure 1-3 One ATA100 and Two Ethernet System

Chapter 2. Internal Block Diagram

2.1. TX4939 System Block Diagram

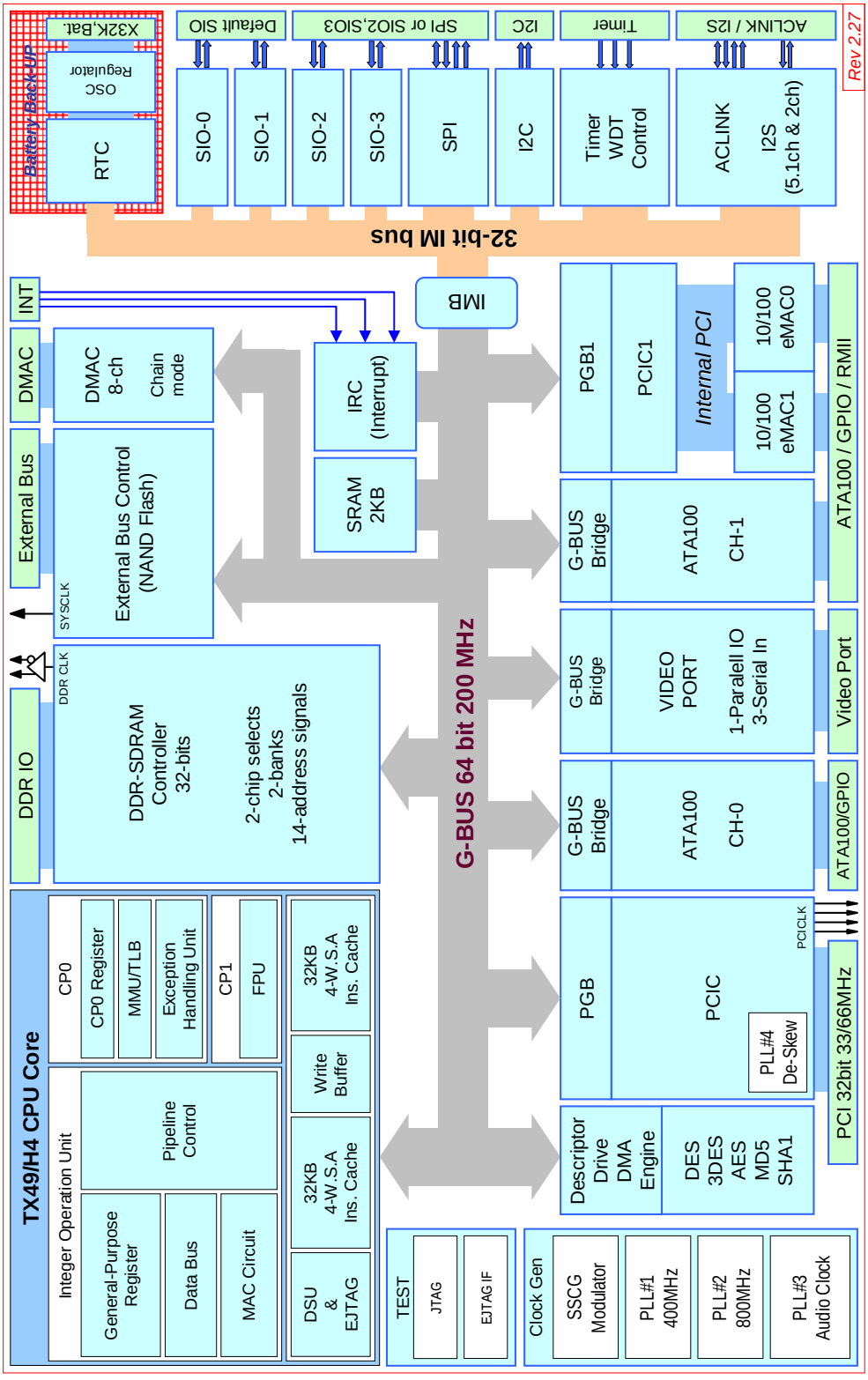


Figure 2-1 TX4939 Internal Block Diagram

2.2. TX49/H4 Core Features

The TX49/H4 core is a high-performance, low power consumption 64-bit RISC CPU core developed by Toshiba. This core has the following features.

- *64-bit Operation*
- *32/64-bit Integer General-purpose Registers*
- *64-GB Physical Address Space*
- *Optimized 5-stage Pipeline*
- *Instruction Set*
  Upwards Compatible with MIPS III ISA
  Added 3-operand multiply instruction, MAC (Multiply Accumulate) instruction, PREF (pre- fetch) instruction.
- *32-KB Instruction Cache, 32-KB Byte Data Cache*
  4-way Set Associative, Lock Function Support
- *MMU (Memory Management Unit)*
  48-double-entry (odd/even) Joint TLB
- *On-chip IEEE754-compliant Single and Double Precision FPU*
- *4-stage Write Buffer*
- *Debugging Support EJTAG (Enhanced JTAG)*
  Debugging Module (DSM)

2.3. Power Management Feature

TX4939 has extensive power management features, which will enable to develop ultimate low power system. Those features are:

- *Power Management in TX49/H4 CPU Core.*
- *Internal Controller Clock and Status Control*
- *Battery Back-Up Real Time Clock*

2.3.1. Strategy for Power Management

In general, there are two major strategies to reduce total chip power consumption, such as (1) switching the clock frequency between normal and some of reduced one and (2) switching the CPU operation mode between normal and power save mode. These changing will be controlled by CPU by observing the system situation.

In case of TX4939 SoC, because of the complexity of clock generator, which produces all necessary clocks out of one 20 MHz clock source by cascading two PLL, it is not practical to switch the clock frequency to another on the fly.
So that, CPU operation mode switching will be used for CPU power management in conjunction with peripheral operation mode switching described below.

2.3.2. Power Management for Internal Controller

The most of internal controller has power management facilities. Such as Clock Disable and Resetting. Table 2-1 lists the whole control bits.

Table 2-1 Periphheral Clock and Reset Control

CLOCK	RESET	DESCRIPTION	COMMENT
ETH2CKD	ETH2RST	Clock Stop or Hold Reset the Ethernet MAC 2	
ETH1CKD	ETH1RST	Clock Stop or Hold Reset the Ethernet MAC 1 controller.	
BROMCKD	BROMRST	Clock Stop or Hold Reset the BROM/SRAM controller.	
NDCCKD	NDCRST	Clock Stop or Hold Reset the NAND Flash controller.	
I2CCKD	I2CRST	Clock Stop or Hold Reset the I2C controller.	
ETH0CKD	ETHC0RST	Clock Stop or Hold Reset the Ethernet MAC 0 controller.	
SPICKD	SPIRST	Clock Stop or Hold Reset the SPI controller.	
SRAMCKD	SRAMRST	Clock Stop or Hold Reset the SRAM controller.	
PCIC1CKD	PCIC1RST	Clock Stop or Hold Reset the PCI controller 1.	
DMA1CKD	DMA1RST	Clock Stop or Hold Reset the DMA controller 1.	
ACLCKD	ACLRST	Clock Stop or Hold Reset the AC-link controller.	
ATA0CKD	ATA0RST	Clock Stop or Hold Reset the ATA0 controller.	
DMA0CKD	DMA0RST	Clock Stop or Hold Reset the DMA controller 0.	
PCICKD	PCICRST	Clock Stop or Hold Reset the PCI controller.	
I2SCKD	I2SRST	Clock Stop or Hold Reset the I2S controller.	
TM0CKD	TM0RST	Clock Stop or Hold Reset the TMR0 controller.	
TM1CKD	TM1RST	Clock Stop or Hold Reset the TMR1 controller.	
TM2CKD	TM2RST	Clock Stop or Hold Reset the TMR2 controller.	
SIOCKD	SIORST	Clock Stop or Hold Reset the SIO controller.	
CYPCKD	CYPRST	Clock Stop or Hold Reset the Cipher controller.	

2.3.3. Battery Back-Up Real Time Clock

TX4939 has a battery back-up Real Time Clock with alarm function. This alarm function generates interrupt signal both CPU and external terminal. The external interrupt signal can wake the CPU up by the help of external power control circuit.

This RTC is designed as low power circuitry which consumes 5 uA at maximum. This assures 10 years battery life with conventional Lithium battery with 200 mAH capacity. The ultimate low power system can be realized with this RTC.

2.4. TX4939 Peripheral Function Features

◆ *Direct Memory Access Controller (DMAC)*

Has an on-chip 8-channel DMA Controller. Three of channels execute DMA transfer to memory and I/O devices, and the rest of 5-channels are for internal resources (one for NAND Flash and 4 channels are dedicated for ACLC/I2S).

- □ *Each of the 3 memories and I/O device channels can respond to internal/external DMA requests.*
- □ *Has an on-chip 4-channel DMA Controller dedicated to ACLC/I2S and an on-chip 1-channel DMA Controller for internal resource*
- □ *Supports DMA transfer between the on-chip Serial I/O Controller and AC-Link Controller as internal DMA requests.*
- □ *Supports transfer between external I/O devices and memory with 16-bit or 8-bit data bus.*
- □ *Supports modes for copying between memories that have no address boundary restrictions. Burst transfers of up to 8 double words are possible for a single read or write operation.*
- □ *Supports modes for writing double word data in the specified memory region and filling memory.*
- □ *Supports Chain DMA transfer.*

◆ *DDR SDRAM Controller (DDRC)*

The DDR SDRAM Controller generates the clock and control signals that are required by the DDR SDRAM interface. The DDR SDRAM Controller has 2 on-chip channels and can support memory sizes of up to 2GB (1GB/Channel) by supporting various memory configurations.

NOTE: Motherboard for the memory configuration more than 4-chips or 1-GB should be carefully designed.

- □ *Memory Clock Frequency: from 100 MHz to 166 MHz, and 200MHz*
- □ *2 Independent Memory Channels*
- □ *Supports 2/ 4-bank 64 Mb, 128 Mb, 256 Mb, 512 Mb, and 1 Gb DDR SDRAM.*
- □ *256Mbit (x16) 2pcs (=64MB)*
- □ *512Mbit (x16) 2pcs (=128MB)*
- □ *256Mbit (x16) 4pcs (=128MB)*
- □ *1Gbit (x16) 4pcs (=512MB) (Max capacity with x16 chip)*
- □ *256Mbit (x8) 4pcs (=128MB)*
- □ *1Gbit (x8) 8pcs (=1GB) (Max capacity with x8 chip)*
- □ *Data bus width is 32 bits.*
  One kind of DDR SDRAM timing set for all channel
  Supports critical word first access of the TX49/H4 core.
  Low power consumption mode: Can select Self-refresh or Pre-charge Power Down.

- □ *Advanced Memory Mapping Technology*
  4 (four) independent DDR memory mapping windows enable mapping the DDR memory space to anywhere TX49 physical memory space with 16 MB resolution.

◆ *External Bus Interface*

the External bus has 8/16-bit data bus. Address bus consists of 6 dedicated and 16+3 external latched address signals

- □ *8/16-bit External bus operation with DMA assistance*
- □ *Direct Address SA [5:0] + Latched from SADB[15:0] + Latched from SA[5:0]*

8-bit bus mode	*A[21:6] can be latched from SADB[15:0]* *A[27:22] can be latched from SA[5:0]*	*4 MB* *256 MB*
16-bit bus mode	*A[22:7] can be latched from SADB[15:0]* *A[28:23] can be latched from SA[5:0]*	*8 MB* *512 MB*

- □ *4 independent channels*
  Channel 0 for NOR Flash
  Channel 1-3 for user peripherals
- □ *Supports access to ROM (mask ROM, page mode ROM, EPROM, EEPROM), SRAM, flash memory and I/O peripherals*
- □ *Selectable 1/3, 1/4, 1/5, and 1/6 of GBUS/DDRCLK for SYSCLK*
- □ *Supports the External Acknowledge Signal (ACK*) and External Ready Signal modes*
- □ *NAND Flash Controller with DMA burst transfer mode.*
- □ *Boot setting can be made from the following selections:*
  - Data bus width: 8-bit or 16-bit
  - ACK output or ACK* input*
  - BWE pin (byte enable or byte Write enable)
  - Boot channel clock frequency
- □ *3 channels of DMA are provided.*

◆ *PCI Controller*

Has an on-chip PCI Controller that complies with PCI External Bus Specification Revision 2.2.

- □ *Complies with PCI External Bus Specification Revision 2.2.*
- □ *32-bit PCI interface with maximum PCI clock frequency of 66 MHz.*
- □ *Supports both the target and initiator functions.*
- □ *Can change the address mapping between the internal bus and the PCI Bus.*
- □ *Has an on-chip PCI Bus arbiter. Can connect up to 6 external bus masters each.*
- □ *Has on-chip 1-channel DMA Controller dedicated to the PCI Controller (PCMAC).*
- □ *Support Satellite Mode*
- □ *Support PCI Boot*
  PCI BOOT address moved to 0xFFFE_0000

◆ *Serial I/O Port (SIO)*

Has on-chip 4-channel asynchronous Serial I/O Interface (full duplex SIO).

- □ *Full duplex SIO × 4 channels*

- □ *On-chip baud rate generator*
- □ *FIFO*
  On-chip 8-bit × 8-stage FIFO for transmission
  On-chip 13-bit × 16-stage FIFO (data: 8 bits, status: 5 bits) for reception

- *8-bit Video Port, 3-Serial Video ports*

  MPEG-2 stream IO port based on Bt.656 protocol
  CrYCb Video Capture and Streaming

- *SPI*

  - *Full duplex, synchronous serial data transfer (I/O data, clock signals)*
  - *Can specify 8-bit or 16-bit data length*
  - *Programmable SPI baud rate*
  - *Supports Master mode only*

- *Commercial IR(CiR) (This feature shares pin with SIO-3)*

  The commercial IR for remote controller.

- *Timer/Counter Control (TMR)*

  Has on-chip 3-channel timer/counter.

  - *32-bit up counter × 3 channels*
  - *Supports 3 modes (Interval Timer Mode, Pulse Generator Mode, Watchdog Timer Mode).*
  - *Timer Output Pins: 2 pins*
  - *Count Clock Input Pins: 1 pin*
  - *Watchdog External Reset Signals: 1 signal*

- *AC-link Controller (ACLC) (same function with TX4937/38, but pins are shared with I2S)*

  The AC-link Controller can connect to and manipulate audio and/or modem CODECs described in "Audio CODEC '97 Revision 2.1" (AC'97).

  - *Supports up to 2 CODECs.*
  - *Supports 16-bit PCM left-right channel recording and playback.*
  - *Supports 16-bit surround, center, and LFE channel playback.*
  - *Supports variable rate audio recording and playback.*
  - *Supports modem CODEC Line 1 and GPIO slot.*
  - *Supports AC-link low power mode, wakeup, and warm reset.*
  - *Supports sample data input/output by DMA transfer.*

- *I2S Audio CODEC Interface (I2S) (This feature shares pin with AC-Link)*

  The I2S controller supports two programmable I2S output/input, and one I2S output.
  The basic features of the I2S controller are outlined below:

  - *Two modes of operation (2 channels i/o mode, 3 Channel output mode)*
  - *Support 16, 18, 20 or 24 bits data.*
  - *Support Left or Right-justify with MSB first.*
  - *Support 32, 48 or 64 bit-wide time-slot.*

◆ *Interrupt Controller (IRC)*

The Interrupt Controller receives interrupt requests of the on-chip TX4939 peripheral circuits and external interrupt requests then generates interrupt requests to the TX49/H4 core. Also, the Interrupt Controller has a 16-bit flag register that generates interrupt requests to an external device or to the TX49/H4 core.

- □ *NMI and 7 external interrupt signal inputs from the on-chip peripheral circuits.*
- □ *Sets 8 priority interrupt levels for each interrupt input.*
- □ *Can select either the edge or the level in the interrupt detection mode for each external interrupt.*
- □ *Has as a flag register for interrupt requests an on-chip 16-bit read/write register. Can issue interrupt requests (IRC interrupt) to external devices or to the TX49/H4 core*

◆ *On-Chip SRAM*

- □ *2KB x 1 channel*
- □ *Specifiable base address*
- □ *Accessible by byte, half-word, full-word, or double-words*
- □ *Supports burst access as Cached-Code-Space*
- □ *Supports the critical word first function of the TX49/H4 core*

◆ *Battery Backup RTC*

- □ *48-bit liner counter for RTC function*
- □ *Upper 32-bit counts time of second up to 272 years*
- □ *Lower 16-bit represent decimal-place of second*
- □ *32-bit Alarm register*
- □ *Compare with RTC upper 32-bit all the time*
- □ *Alarm output signal*
- □ *256Byte internal registers*
- □ *6 Byte are assigned for data port of RTC and Alarm register*
- □ *250 Byte are assigned as CMOS RAM*

◆ *CIPHER Engine*

- □ *Supports DES/3DES/AES*
- □ *Supports MD5/SHA1*
- □ *Dedicated DMA with chain mode*
- □ *Reduced KEY exposure for CBC mode in AES*
  (a) 6 set of KEY and Initial-data register
  (b) KEY registers are W/O (write only)
  (c) keep initial data for each stream
  (d) Initial data will be updated for CBC mode
  (e) Decryption-KEY will be generated from Encryption-KEY without exposure
- □ *Supports Ex-OR operation*
- □ *Modular Exponentiation Operation Coprocessor Engine*

◆ *ATA 100 EIDE interface*

- □ *Have on-chip 2-channel ATA 100 EIDE interface*
- □ *Up to 100MB/s Data Transfer x 1 channel*
- □ *Up to 2 EIDE devices each*
- □ *Compliant with ATA/ATAPI-6*
- □ *Can become the Bus Master and transfer data to external memory*
- □ *Can select data bus widths of 16/32/64 bits*
- □ *Has 16 x 64-bits transmit FIFO and 16 x 64-bits receive FIFO*
- □ *Supports Scatter gather DMA (same as TC86C001F)*
- □ *Similar software visibility with TC86C001F*

◆ *Dual Ether MAC (EMAC) (same as TX4938 but Media Independent Interface is RMII)*

Has on-chip Dual Ether MAC.

- □ *Equivalent to the TC35815 except Media Independent Interface (RMII).*
- □ *Supports transfer rates of 100 Mbps and 10 Mbps.*
- □ *Employs the IEEE Std 802.3 standard CSMA/CD format when in the half duplex transfer mode.*
- □ *Employs IEEE Std 802.3x standard MAC control and Pause operation for flow control when in the full duplex transfer mode.*
- □ *RMII (Reduced Media Independent Interface).*

◆ *Extended EJTAG Interface*

The TX4939 Extended EJTAG (Extended Enhanced Joint Test Action Group) Interface provides two functions. The first function is IEEE1149.1 standard compliant JTAG boundary scan testing. The second function is real-time debugging by a Debugging Support Unit (DSU) that is built into the TX49/H4 core.

- □ *IEEE1149.1 JTAG Boundary Scan*
- □ *Can perform execution control (run, break, step, register/memory access) as a real-time debugging function that uses dedicated emulation probes and can use PC traces.*
- □ *PC-Trace dedicated pins are multiplexed with some of Video port signal pins.*

- □ *Tamper protection for EJTAG access.*

Chapter 3. Pin Assignment and Function

3.1. Pin Assign Table

Table 3-1 Pin Assignment Table

Class	Signal_Name	IO	BALL	Class	Signal_Name	IO	BALL	Class	Signal_Name	IO	BALL
S001	SA00	IO	C2	S002	BE1*	O	C1	S003	BE0*	O	D3
S004	CE3*	O	D2	S005	CE2*	O	D1	S006	CE1*	O	E4
S007	INT0*	I	E3	S008	INT1*	I	E2	S009	INT2*	I	E1
S010	NMI*	I	F4	S011	INTA*	I	F3	S012	INTB*	I	F2
S013	PCICLK1	IO	F1	S014	INTC*	I	G4	S015	INTD*	I	G3
S016	PCICLK2	IO	G1	S017	REQ5*	IO	H4	S018	GNT5*	IO	H3
S019	REQ4*	IO	H2	S020	PCICLK3	IO	H1	S021	GNT4*	IO	J4
S022	REQ3*	IO	J3	S023	GNT3*	IO	J2	S024	PCICLK4	IO	J1
S025	REQ2*	IO	K4	S026	GNT2*	IO	K3	S027	PCIRST	IO	K2
S028	REQ1*	IO	K1	S029	GNT1*	IO	L4	S030	GNT0*	IO	L3
S031	REQ0*	IO	L2	S032	PME*	I	L1	S033	PCIAD31	IO	M4
S034	PCIAD30	IO	M3	S035	PCIAD29	IO	M1	S036	PCIAD28	IO	N4
S037	PCIAD27	IO	N3	S038	PCIAD26	IO	N2	S039	PCIAD25	IO	N1
S040	PCIAD24	IO	P1	S041	CBE3*	IO	P2	S042	PCIAD23	IO	P3
S043	PCIAD22	IO	P4	S044	PCIAD21	IO	R1	S045	PCIAD20	IO	R3
S046	PCIAD19	IO	R4	S047	PCIAD18	IO	T1	S048	PCIAD17	IO	T2
S049	PCIAD16	IO	T3	S050	CBE2*	IO	T4	S051	PCICLKIN	I	U1
S052	FRAME*	IO	U2	S053	IRDY*	IO	U3	S054	TRDY*	IO	U4
S055	DEVSEL*	IO	V1	S056	STOP*	IO	V2	S057	LOCK*	IO	V3
S058	PERR*	IO	V4	S059	SERR*	IO	W1	S060	PAR	IO	W2
S061	CBE1*	IO	W3	S062	PCIAD15	IO	W4	S063	PCIAD14	IO	Y1
S064	PCIAD13	IO	Y3	S065	PCIAD12	IO	Y4	S066	PCIAD11	IO	AA1
S067	PCIAD10	IO	AA2	S068	M66EN	I	AA3	S069	PCIAD09	IO	AA4
S070	PCIAD08	IO	AB1	S071	CBE0*	IO	AB2	S072	PCIAD07	IO	AB3
S073	PCIAD06	IO	AB4	S074	PCIAD05	IO	AC1	S075	PCIAD04	IO	AC2
S076	PCIAD03	IO	AC3	S077	PCIAD02	IO	AD1	S078	PCIAD01	IO	AD2
S079	PCIAD00	IO	AE3	S080	A0_DASP*	IO	AF3	S081	A0_DCS1*	IO	AD4
S082	A0_DCS0*	IO	AE4	S083	A0_DA2	IO	AF4	S084	A0_DA0	IO	AC5
S085	A0_PDIAG*	IO	AD5	S086	A0_DA1	IO	AE5	S087	A0_DINTR	IO	AF5
S088	A0_DMACK*	IO	AC6	S089	A0_DIORDY	IO	AD6	S090	A0_DIOR*	IO	AE6
S091	A0_DIOW*	IO	AF6	S092	A0_DMAREQ	IO	AC7	S093	A0_DD15	IO	AD7
S094	A0_DD00	IO	AF7	S095	A0_DD14	IO	AC8	S096	A0_DD01	IO	AD8
S097	A0_DD13	IO	AE8	S098	A0_DD02	IO	AF8	S099	A0_DD12	IO	AC9
S100	A0_DD03	IO	AD9	S101	A0_DD11	IO	AE9	S102	A0_DD04	IO	AF9
S103	A0_DD10	IO	AC10	S104	A0_DD05	IO	AD10	S105	A0_DD09	IO	AE10
S106	A0_DD06	IO	AF10	S107	A0_DD08	IO	AC11	S108	A0_DD07	IO	AD11
S109	A0_DRST*	O	AE11	S110	A1_DASP*	IO	AF11	S111	A1_DCS1*	IO	AC12
S112	A1_DCS0*	IO	AD12	S113	A1_DA2	IO	AF12	S114	A1_DA0	IO	AC13
S115	A1_PDIAG*	IO	AD13	S116	A1_DA1	IO	AE13	S117	A1_DINTR	IO	AF13
S118	A1_DMACK*	IO	AF14	S119	A1_DIORDY	IO	AE14	S120	A1_DIOR*	IO	AD14
S121	A1_DIOW*	IO	AC14	S122	A1_DMAREQ	IO	AF15	S123	A1_DD15	IO	AD15
S124	A1_DD00	IO	AC15	S125	A1_DD14	IO	AF16	S126	A1_DD01	IO	AE16
S127	A1_DD13	IO	AD16	S128	A1_DD02	IO	AC16	S129	A1_DD12	IO	AF17
S130	A1_DD03	IO	AE17	S131	A1_DD11	IO	AD17	S132	A1_DD04	IO	AC17
S133	A1_DD10	IO	AF18	S134	A1_DD05	IO	AE18	S135	A1_DD09	IO	AD18
S136	A1_DD06	IO	AC18	S137	A1_DD08	IO	AF19	S138	A1_DD07	IO	AE19
S139	A1_DRST*	O	AD19	S140	VDCLKIN1	I	AC19	S141	VDCLKO1	O	AF20
S142	VDVLD1	IO	AD20	S143	VDPSN1	IO	AC20	S144	VDS1	IO	AF21
S145	VDCLKIN	I	AE21	S146	VDCLKO	O	AD21	S147	VDVLD	IO	AC21
S148	VDPSN	IO	AF22	S149	VD0	IO	AE22	S150	VD1	IO	AD22
S151	VD2	IO	AC22	S152	VD3	IO	AF23	S153	VD4	IO	AE23
S154	VD5	IO	AD23	S155	VD6	IO	AF24	S156	VD7	IO	AE24
S157	DRDQ0	IO	AD25	S158	DRDQ1	IO	AD26	S159	DRDQ2	IO	AC24
S160	DRDQ3	IO	AC25	S161	DRDM0	O	AC26	S162	DRDQS0	IO	AB23

Table 3-1 Pin Assignment Table

Class	Signal_Name	IO	BALL	Class	Signal_Name	IO	BALL	Class	Signal_Name	IO	BALL
S163	DRDQ4	IO	AB24	S164	DRDQ5	IO	AB25	S165	DRDQ6	IO	AB26
S166	DRDQ7	IO	AA23	S167	DRDQ8	IO	AA24	S168	DRDQ9	IO	AA25
S169	DRDQ10	IO	AA26	S170	DRDQ11	IO	Y23	S171	DRDM1	O	Y24
S172	DRDQS1	IO	Y26	S173	DRDQ12	IO	W23	S174	DRDQ13	IO	W24
S175	DRDQ14	IO	W25	S176	DRDQ15	IO	W26	S178	DRCKE	O	V24
S179	DRA12	O	V25	S180	DRA11	O	V26	S181	DRA8	O	U23
S182	DRA9	O	U24	S183	DRA6	O	U25	S184	DRA7	O	U26
S185	DRA4	O	T23	S186	DRA5	O	T24	S187	DRA2	O	T25
S188	DRA3	O	T26	S189	DRA0	O	R23	S190	DRVref1	I	R24
S191	DRA1	O	R26	S192	DRBA1	O	P23	S193	DRA10	O	P24
S194	DRA13	O	P25	S195	DRRAS*	O	P26	S196	DRBA0	O	N26
S197	DRCAS*	O	N25	S198	DRWE*	O	N24	S199	DRCS1*	O	N23
S200	DRCS0*	O	M26	S201	DRDQ16	IO	M24	S202	DRDQ17	IO	M23
S203	DRDQ18	IO	L26	S204	DRDQ19	IO	L25	S205	DRDM2	O	L24
S206	DRDQS2	IO	L23	S207	DRDQ20	IO	K26	S208	DRDQ21	IO	K25
S210	DRDQ22	IO	K24	S211	DRDQ23	IO	K23	S212	DRDQ24	IO	J26
S213	DRDQ25	IO	J25	S214	DRDQ26	IO	J24	S215	DRDQ27	IO	J23
S216	DRDM3	O	H26	S218	DRDQS3	IO	H25	S219	DRDQ28	IO	H24
S220	DRDQ29	IO	H23	S221	DRDQ30	IO	G26	S222	DRDQ31	IO	G24
S223	DRCKOUT	O	F24	S224	DRCKP	O	F26	S226	DRCKM	O	E26
S227	DRCKREF	I	E24	S228	PLL12VSS_A	I	D25	S229	PLL12VDD_A	I	D24
S232	PLL3VSS_A	I	C26	S233	PLL3VDD_A	I	C25	S235	MSTCLK	I	B24
S236	RESET*	I	A24	S237	MSTCLK2	I	C23	S238	div_RST*	I	B23
S239	TDO	O	A23	S240	TEST0*	I	D22	S241	HALTDOZE	O	C22
S242	TCK	I	B22	S243	TRST*	I	A22	S244	BYPASSPLL*	I	D21
S245	TMS	I	C21	S246	TDI	I	B21	S247	RXD0	IO	A21
S248	TXD0	IO	D20	S249	RXD1	IO	C20	S250	ALARM*	O	A20
S251	VDDR	I	D19	S252	RTCVDD	O	C19	S253	X32OUT	I	B19
S254	X32IN	I	A19	S255	TXD1	IO	D18	S256	ACRESET*	O	C18
S257	SYNC	O	B18	S258	SDOUT	O	A18	S259	SDIN1	I	D17
S260	SDIN0	I	C17	S261	BITCLK	I	B17	S262	I2C_CLK	IO	A17
S263	EJRST*	I	D16	S264	I2C_DO	IO	C16	S265	SPIIN	IO	B16
S266	SPIOUT	IO	A16	S267	SPICS0	IO	D15	S268	SPICLK	IO	C15
S269	GP00	IO	A15	S270	GP01	IO	D14	S271	GP02	IO	C14
S272	GP03	IO	B14	S273	TIMER0	O	A14	S274	TIMER1	O	A13
S275	WDRST*	O	B13	S276	NDCE*	O	C13	S277	NDLA	O	D13
S278	NDRB*	I	A12	S279	NRCE*	O	C12	S280	DMADONE*	IO	D12
S281	SADB00	IO	A11	S282	SADB01	IO	B11	S283	SADB02	IO	C11
S284	ACK*	IO	D11	S285	SADB03	IO	A10	S286	SADB04	IO	B10
S287	SADB05	IO	C10	S288	SADB06	IO	D10	S289	SYSCLK	O	A9
S290	SADB07	IO	B9	S291	SADB08	IO	C9	S292	SADB09	IO	D9
S293	ACE*	O	A8	S294	SADB10	IO	B8	S295	SADB11	IO	C8
S296	SADB12	IO	D8	S297	SADB13	IO	A7	S298	SADB14	IO	C7
S299	SADB15	IO	D7	S300	OE*	O	A6	S301	SWE*	O	B6
S302	DMAACK0	IO	C6	S303	DMAREQ0	I	D6	S304	DMAREQ2	I	A5
S305	DMAACK2	IO	B5	S306	SA02	IO	C5	S307	SA05	IO	D5
S308	DMAREQ1	I	A4	S309	DMAACK1	IO	B4	S310	SA03	IO	C4
S311	SA04	IO	A3	S312	SA01	IO	B3				

Table 3-2 Power Pin Assignment

P&G	Ball Names
VSS	A1, A2, A25, A26, B1, B2, B7, B12, B15, B20, B25, B26, C3, C24, D4, D23, D26, E5, E22, E23, E25, F23, F25, G2, G23, G25, L11, L12, L13, L14, L15, L16, M2, M11, M12, M13, M14, M15, M16, M25, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, R2, R11, R12, R13, R14, R15, R16, R25, T11, T12, T13, T14, T15, T16, V23, Y2, Y25, AB5, AB22, AC4, AC23, AD3, AD24, AE1, AE2, AE7, AE12, AE15, AE20, AE25, AE26, AF1, AF2, AF25, AF26
VDDC	E7, E8, E11, E12, E15, E16, E19, E20, G5, G22, H5, H22, L5, L22, M5, M22, R5, R22, T5, T22, W5, W22, Y5, Y22, AB7, AB8, AB11, AB12, AB15, AB16, AB19, AB20
VDD33	E6, E9, E10, E13, E14, E17, E18, E21, F5, J5, K5, N5, P5, U5, V5, AA5, AB6, AB9, AB10, AB13, AB14, AB17, AB18, AB21
VDD25	F22, J22, K22, N22, P22, U22, V22, AA22

3.2. Pin Alignment (TOP VIEW)

Package Outline TX4939XBG TOP VIEW

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	VSS	VSS	SA04	DMAREQ1	DMAREQ2	OE*	SADB13	ACE*	SYSCLK	SADB03	SADB00	NDRB*	TIMER1	TIMER0	GP00	SPIOUT	I2C_CLK	SDOUT	X32IN	ALARM*	RXD0	TRST*	TDO	RESET*	VSS	VSS
B	VSS	VSS	SA01	DMAACK1	DMAACK2	SWE*	VSS	SADB10	SADB07	SADB04	SADB01	VSS	WDRST*	GP03	VSS	SPIIN	BITCLK	SYNC	X32OUT	VSS	TDI	TCK	div_RST*	MSTCLK	VSS	VSS
C	BE1*	SA00	VSS	SA03	SA02	DMAACK0	SADB14	SADB11	SADB08	SADB05	SADB02	NRCE*	NDCE*	GP02	SPICLK	I2C_DO	SDIN0	ACRESET*	RTCVDD	RXD1	TMS	HALTDOZE	MSTCLK2	VSS	PLL3VDD_A	PLL3VSS_A
D	CE2*	CE3*	BE0*	VSS	SA05	DMAREQ0	SADB15	SADB12	SADB09	SADB06	ACK*	DMADONE*	NDLA	GP01	SPICS0	EJRST*	SDIN1	TXD1	VDDR	TXD0	BYPASSPLL	TEST0*	VSS	PLL12VDD_	PLL12VSS_	VSS
E	INT2*	INT1*	INT0*	CE1*	VSS	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VSS	VSS	DRCKREF	VSS	DRCKM
F	PCICLK1	INTB*	INTA*	NMI*	VDD33																	VDD25	VSS	DRCKOUT	VSS	DRCKP
G	PCICLK2	VSS	INTD*	INTC*	VDDC																	VDDC	VSS	DRDQ31	VSS	DRDQ30
H	PCICLK3	REQ4*	GNT5*	REQ5*	VDDC																	VDDC	DRDQ29	DRDQ28	DRDQS3	DRDM3
J	PCICLK4	GNT3*	REQ3*	GNT4*	VDD33																	VDD25	DRDQ27	DRDQ26	DRDQ25	DRDQ24
K	REQ1*	PCIRST	GNT2*	REQ2*	VDD33																	VDD25	DRDQ23	DRDQ22	DRDQ21	DRDQ20
L	PME*	REQ0*	GNT0*	GNT1*	VDDC						VSS	VSS	VSS	VSS	VSS	VSS						VDDC	DRDQS2	DRDM2	DRDQ19	DRDQ18
M	PCIAD29	VSS	PCIAD30	PCIAD31	VDDC						VSS	VSS	VSS	VSS	VSS	VSS						VDDC	DRDQ17	DRDQ16	VSS	DRCS0*
N	PCIAD25	PCIAD26	PCIAD27	PCIAD28	VDD33						VSS	VSS	VSS	VSS	VSS	VSS						VDD25	DRCS1*	DRWE*	DRCAS*	DRBA0
P	PCIAD24	CBE3*	PCIAD23	PCIAD22	VDD33						VSS	VSS	VSS	VSS	VSS	VSS						VDD25	DRBA1	DRA10	DRA13	DRRAS*
R	PCIAD21	VSS	PCIAD20	PCIAD19	VDDC						VSS	VSS	VSS	VSS	VSS	VSS						VDDC	DRA0	DRVref1	VSS	DRA1
T	PCIAD18	PCIAD17	PCIAD16	CBE2*	VDDC						VSS	VSS	VSS	VSS	VSS	VSS						VDDC	DRA4	DRA5	DRA2	DRA3
U	PCICLKIN	FRAME*	IRDY*	TRDY*	VDD33																	VDD25	DRA8	DRA9	DRA6	DRA7
V	DEVSEL*	STOP*	LOCK*	PERR*	VDD33																	VDD25	VSS	DRCKE	DRA12	DRA11
W	SERR*	PAR	CBE1*	PCIAD15	VDDC																	VDDC	DRDQ12	DRDQ13	DRDQ14	DRDQ15
Y	PCIAD14	VSS	PCIAD13	PCIAD12	VDDC																	VDDC	DRDQ11	DRDM1	VSS	DRDQS1
AA	PCIAD11	PCIAD10	M66EN	PCIAD09	VDD33																	VDD25	DRDQ7	DRDQ8	DRDQ9	DRDQ10
AB	PCIAD08	CBE0*	PCIAD07	PCIAD06	VSS	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VDD33	VDDC	VDDC	VDD33	VSS	DRDQS0	DRDQ4	DRDQ5	DRDQ6
AC	PCIAD05	PCIAD04	PCIAD03	VSS	A0_DA0	A0_DMACK*	A0_DMAREQ	A0_DD14	A0_DD12	A0_DD10	A0_DD08	A1_DCS1*	A1_DA0	A1_DIOW*	A1_DD00	A1_DD02	A1_DD04	A1_DD06	VDCLKIN1	VDPSN1	VDVLD	VD2	VSS	DRDQ2	DRDQ3	DRDM0
AD	PCIAD02	PCIAD01	VSS	A0_DCS1*	A0_PDIAG*	A0_DIORDY	A0_DD15	A0_DD01	A0_DD03	A0_DD05	A0_DD07	A1_DCS0*	A1_PDIAG*	A1_DIOR*	A1_DD15	A1_DD13	A1_DD11	A1_DD09	A1_DRST*	VDVLD1	VDCLKO	VD1	VD5	VSS	DRDQ0	DRDQ1
AE	VSS	VSS	PCIAD00	A0_DCS0*	A0_DA1	A0_DIOR*	VSS	A0_DD13	A0_DD11	A0_DD09	A0_DRST*	VSS	A1_DA1	A1_DIORDY	VSS	A1_DD01	A1_DD03	A1_DD05	A1_DD07	VSS	VDCLKIN	VD0	VD4	VD7	VSS	VSS
AF	VSS	VSS	A0_DASP*	A0_DA2	A0_DINTR	A0_DIOW*	A0_DD00	A0_DD02	A0_DD04	A0_DD06	A1_DASP*	A1_DA2	A1_DINTR	A1_DMACK*	A1_DMAREQ	A1_DD14	A1_DD12	A1_DD10	A1_DD08	VDCLKO1	VDS1	VDPSN	VD3	VD6	VSS	VSS

Figure 3-1 Pin Assignment on Package (TOP VIEW)

3.3. Pin Function

Note: In the following table, "PU" in the I/O column indicates the presence of internal pull-up resistance, and "OD" indicates open drain. The asterisk "*" at the end of a signal name indicates that that signal is an Active Low signal.

3.3.1. System Clock and RESET Signals

System Clock and RESET Signals		
Signal Name	**I/O**	**Function**
MSTCLK	Input	Master Clock Input 20 MHz System Clock is required. This clock is used to generate internal clock for data processing. This clock can be modulated by external SSCG generator.
MSTCLK2	Input	Master Clock Input2 20 MHz System Clock is required. This clock is used to generate internal clock for UART Baud rate clock and Audio sampling clock. This clock should not be modulated by external SSCG generator.
RESET*	Input	Reset This is system-reset signal.
HALTDOZE	Output	Halt/Doze Status Output This signal is asserted (outputs the "H" level) when in either the Halt mode or the Doze mode.
IOSRST*	Output	IO System Reset signal out This pin is multiplexed with PCICLK[4] and can be used to reset external IO system reset by software.
SYSRST*	Output	System Reset signal out This pin is multiplexed with PCICLK[3] and can be used to reset whole system or generate NMI to the CPU.

3.3.2. DDR SDRAM Interface Signals

DDR SDRAM Interface Signals		
Signal Name	**I/O**	**Function**
DRA[13:0]	Output	Address This address signals are for DDR SDRAM.
DRDQ[31:0]	I/O	Data Bus This is a 32-bit data bus dedicated for DDR SDRAM
DRDQS[3:0]	I/O	DDR Control Signal for each byte data
DRDM[3:0]	Output	DDR Control signal for each byte data
DRCS[1:0]*	Output	DDR Chip Select
DRVref1	Input	DDR Voltage reference input
DRCKP, DRCKM	Output	Differential DDR Clock Source
DRCKOUT	Output	Single-end DDR Reference Clock for deskew
DRCKREF	Input	DDR Clock deskew reference input
DRBA[1:0]	Output	DDR Bank Address
DRWE*, DRCAS*, DRRAS*	Output	DDR Control Signals
DRCKE	Output	DDR Control Signals Clock Enable signal

3.3.3. VIDEO Port Interface Signal

VIDEO Port Interface Signals		
Signal Name	**I/O**	**Function**
Parallel Video Port (Input/Output), (CCIR656, EN50083)		
VDCLKIN	I/O, PU	Video Port clock input
VDCLKO	I/O, PU	Video Port clock output
VD[7:0]	I/O, PU	Video Port
VDVLD	I/O, PU	Video Port Data Valid (Active High)
VDPSN	I/O, PU	Video Port sync (Active High).
Serial Video Port 1 (Input/Output, TS)		
VDCLKIN1	I/O, PU	Video Port clock input
VDCLKO1	I/O, PU	Video Port clock output
VDS1	I/O, PU	Video Port input/output
VDVLD1	I/O, PU	Video Port Data Valid (Active High)
VDPSN1	I/O, PU	Video Port sync (Active High).
Serial Video Port 2 (Input TS)		
VDCLKIN2	I/O, PU	Video Port clock input
VDCLKO2	I/O, PU	Video Port clock output
VDS2	I/O, PU	Video Port input
VDVLD2	I/O, PU	Video Port Data Valid (Active High)
VDPSN2	I/O, PU	Video Port sync (Active High).
Serial Video Port 3 (Input TS)		
VDCLKIN3	I/O, PU	Video Port clock input
VDCLKO3	I/O, PU	Video Port clock output
VDS3	I/O, PU	Video Port input
VDVLD3	I/O, PU	Video Port Data Valid (Active High)
VDPSN3	I/O, PU	Video Port sync (Active High).

Note: Since All Signals can be used as GPIO, only bidirectional IO is used.

3.3.4. ATA100 Channel 0 Interface

ATA100 Interface		
Signal Name	**I/O**	**Function**
A0_DD [15:0]	I/O	Data bidirectional
A0_DA [2:0]	I/O	Address output
A0_DCS [1:0]*	I/O	Chip Select output
A0_DIOW*	I/O	IO write output
A0_DIOR*	I/O	IO read output
A0_DMACK*	I/O	DMA Acknowledge output
A0_DRST*	I/O	Reset output
A0_DMAREQ	I/O	DMA Request input
A0_DIORDY	I/O	IO Channel Ready input
A0_DINTR	I/O	Interrupt Request input
A0_PDIAG*	I/O	Passed Diagnostics input
A0_DASP*	I/O	Slave Presence input

Note: Since All Signals can be used as GPIO, only bidirectional IO is used.

3.3.5. ATA100 Channel 1 Interface

ATA100 Interface		
Signal Name	**I/O**	**Function**
A1_DD [15:0]	I/O	Data bidirectional
A1_DA [2:0]	I/O	Address
A1_DCS [1:0]*	I/O	Chip Select
A1_DIOW*	I/O	IO write
A1_DIOR*	I/O	IO read
A1_DMACK*	I/O	DMA Acknowledge
A1_DRST*	I/O	Reset
A1_DMAREQ	I/O	DMA Request
A1_DIORDY	I/O	IO Channel Ready
A1_DINTR	I/O	Interrupt Request
A1_PDIAG*	I/O	Passed Diagnostics
A1_DASP*	I/O	Slave Presence

Note: Since All Signals can be used as GPIO, only bidirectional IO is used.

3.3.6. External Bus Interface Signals

External Bus Signals		
Signal Name	**I/O**	**Function**
SADB[15:0]	I/O PU	External Bus Address Data Bus (Lower 16-bit) 16/8-bit data bus for External bus. Address [21:6] will appear on this bus to latch by external latch devices.
BE[1:0]*	Output	Byte Enable Signal Access Control signal for upper and lower byte on External bus.
ACE*	Output	Address Clock Enable This is the Latch Enable signal for the upper ADDR address.
SA[5:0]	I/O PU	Lower Address Bits This is the lower address for External bus access. These pins are also used for Boot Configuration.
SYSCLK	Output	System Clock This is a device clock for external I/O.
NRCE* (CE[0])	Output	Chip Enable, dedicated for "NOR" Flash access as boot device.
CE[3:1]*	Output	Chip Enables for generic use
OE*	Output	Output Enable This is the Output Enable signal for ROM, SRAM, or I/O devices.
SWE*	Output	Write Enable This is the Write Enable signal for SRAM and I/O devices.
ACK* (READY)	Input PU	Acknowledge This is the Flow Control signal.
DMAREQ[2:0]	Input PU	DMA Request This is the DMA Transfer Request signal from external I/O devices.
DMAACK[2:0]	I/O PU	DMA Acknowledge This is the DMA Transfer Acknowledge signal for external I/O devices. This pin also used for the Boot Configuration.
DMADONE*	I/O PU	DMA Done This signal functions either as an output signal that signals the end of DMA transfer or as an input signal for ending DMA transfer.
Signals for NAND Controller		
NDCE*	Output	NAND Flash Chip Enable (NAND CE*)
NDLA	Output PU	NAND Flash Control Signal Latch Enable
NDRB*	Input	NAND Flash Ready/Busy (NAND R/B)
NDCLE	(Output)	NAND CLE that can be generated from SA[0] latched by NDLA
NDALE	(Output)	NAND ALE that can be generated from SA[1] latched by NDLA
NDCS0	(Output)	Extended NAND chip select, that can be generated from SA[2] latched by NDLA
NDCS1	(Output)	Extended NAND chip select, that can be generated from SA[3] latched by NDLA
NDRE*	(Output)	NAND RE* that can be generated from SA[4] latched by NDLA
NDWE*	(Output)	NAND WE* that can be generated from SA[5] latched by NDLA

3.3.7. ISA Interface Signals

External Bus Signals		
Signal Name	**I/O**	**Function**
SADB[15:0]	I/O PU	ISA Data Bus (16-bit) Address [21:6] will appear on this bus to latch by external latch devices.
SA[21:0]	Output	ISA Address Bus SA[5:0] is connecting direct to TX4939 SA[21:6] is connecting directly to external address latch devices.
BHE*	Output	Byte High Enable - Together with SA[0] to determine if this is a 8-bit or 16-bit cycle
IOW*	Output	IO Write Command.
IOR*	Output	IO Read Command.
WAIT	Input	WAIT command. ISA device use this signal to drag out the cycle, this signal is needed for slow ISA device.
CS*	Output	ISA Chip Select.

3.3.8. Default GPIO

Default GPIO Signals		
Signal Name	**I/O**	**Function**
GPIO00	I/O	Default GPIO 00
GPIO01	I/O	Default GPIO 01
GPIO02	I/O	Default GPIO 02
GPIO03	I/O	Default GPIO 03

3.3.9. PCI Interface Signals

PCI Interface		
Signal Name	**I/O**	**Function**
Signals for Host Mode		
PCICLK[4:1]	I/O	PCI Clock (Service Clock) These are the service PCI Clocks.
PCIRST	Output	PCI Bus Reset Signal (Service Output) Reset Output for PCI Bus target cards.
PCICLKIN	Input	PCI Clock Input This is the PCI Clock input.
PCIAD[31:0]	I/O	PCI Address and Data This is the Address/Data Multiplex Bus.
C_BE[3:0]*	I/O	Command and Byte Enable This is the Command and Byte Enable signal.
PAR	I/O	Parity This is the Parity signal for PCIAD[31:0] and C_BE[3:0]*. Parity is even.
FRAME*	I/O	Cycle Frame Indicates that a bus operation is being run.
IRDY*	I/O	Initiator Ready This signal indicates that the Initiator has finished transferring data.
TRDY*	I/O	Target Ready This signal indicates that the Target is ready to finish transferring data.
STOP*	I/O	Stop This signal indicates that the Target requests the Initiator to stop transferring data.
LOCK*	I/O	PCI Resource Clock This signal indicates that the PCI Bus Master has locked (established exclusive access to) a particular memory Target on the PCI Bus.
DEVSEL*	I/O	Device Select The Target asserts this response signal to access performed by the Initiator.
REQ[5:1]*	I/O	Request The PCI Arbiter's Bus Request inputs for the channel of 1 to 5.
REQ[0]*	I/O	Request In the internal arbiter mode this pin will be 6th arbiter REQ* input.
GNT[0]*	I/O	Grant In the internal arbiter mode this pin will be 6th arbiter GNT* output.
GNT[5:1]*	I/O	Grant The PCI Arbiter's Bus Grant Output for the channel of 1 to 5
PERR*	I/O	Data Parity Error This signal indicates that a data parity error occurred in a bus cycle other than a Special cycle.
SERR*	I/O	System Error This signal indicates that either an address parity error, a data parity error during a Special cycle, or a fatal error occurred.
M66EN	Input	66 MHz Clock Enable 1: Enable the 66 MHz operation mode. 0: Disable the 66 MHz operation mode.
PME*	Input	Power Management This signal indicates the Power Management mode status.
Additional Signals for External Arbiter and Satellite Mode (Note)		
REQ*	Output	This pin represents REQ* output signal of internal PCI core.
GNT*	Input	This pin represents GNT* input of internal PCI Core.
INTOUT*	Output	Extra Interrupt output.
IDSEL	Input	This pin represents IDSEL input for Satellite mode.

Note: With the PCI Host mode, both external and internal arbiter can be used. But the PCI Satellite mode, only external arbiter mode available.

3.3.10. Ethernet MAC Interface (RMII)

Ethernet MAC Channel 0 Interface		
Signal Name	**I/O**	**Function**
E0RCLKI	Input	Reference Clock Input
E0CRS_DV	Input	Carrier Sense/Receive Data Valid from PHY
E0RXD[1:0]	Input	Receive Data
E0TX_EN	Output	Transmit Enable
E0TXD[1:0]	Output	Transmit Data
E0RX_ER	Input	Receive Error
E0MDC	Output	Management data clock output
E0MDIO	I/O	Management data bi-directional signal

Ethernet MAC Channel 1 Interface		
Signal Name	**I/O**	**Function**
E1RCLKI	Input	Reference Clock Input
E1CRS_DV	Input	Carrier Sense/Receive Data Valid from PHY
E1RXD[1:0]	Input	Receive Data
E1TX_EN	Output	Transmit Enable
E1TXD[1:0]	Output	Transmit Data
E1RX_ER	Input	Receive Error
E1MDC	Output	Management data clock output
E1MDIO	I/O	Management data bi-directional signal

3.3.11. AC-Link Interface

AC-Link Interface		
Signal Name	**I/O**	**Function**
ACRESET*	Output	AC'97 Master Hardware Reset This is the AC'97 Master Hardware reset signal.
BITCLK	Input	12.288 MHz Serial Data Clock The BITCLK signal is an input clock from an external AC'97 device.
SDIN[0]	Input PD	Serial, Time Division Multiplexed, AC'97 Input Stream This is the serial Time Division Multiplexed AC'97 output stream input.
SDOUT	Output	Serial, Time Division Multiplexed, AC'97 Output Stream This is the serial Time Division Multiplexed AC'97 output stream output.
SYNC	Output	48 kHz Fixed Rate Sample Synchronous This is 48kHz Fixed sample synchronous signal.
SDIN[1]	Input PD	Serial, Time Division Multiplexed, AC'97 Input Stream This is the serial Time Division Multiplexed AC'97 output stream input.

3.3.12. I2S Interface 2-channel mode

I2S Interface		
Signal Name	**I/O**	**Function**
I2S_WS0	I/O	Channel 0 Word Select
I2S_SCK0	I/O	Channel 0 Bit Clock
I2S_SD0	I/O PD	Channel 0 Data
I2S_WS1	I/O	Channel 1 Word Select
I2S_SCK1	I/O	Channel 1 Bit Clock
I2S_SD1	I/O PD	Channel 1 Data

3.3.13. I2S Interface 5.1 channel mode

I2S Interface		
Signal Name	**I/O**	**Function**
I2S_WS	I/O	Word Select
I2S_SCK	I/O	Bit Clock
I2S_SD0	I/O PD	Channel 0 Data
I2S_SD1	I/O	Channel 1 Data
I2S_SD2	I/O PD	Channel 2 Data

3.3.14. I2C Interface

I2C Interface		
Signal Name	**I/O**	**Function**
I2C_CLK	O	Bit Clock
I2C_DO	I/O	Data

3.3.15. SPI Interface

I2C Interface		
Signal Name	**I/O**	**Function**
SPICLK	I/O, PU	Bit Clock (output)
SPIOUT	I/O, PU	Data Out
SPIIN	I/O, PU	Data In
SPICS0	I/O, PU	SPI Chip Select (output)

Note Since All Signals can be used as GPIO, only bidirectional IO is used.

3.3.16. RTC Interface

RTC Interface		
Signal Name	**I/O**	**Function**
RTCVDD	Input	Power Supply for RTC module. 2.7 V to 3.3 V
VDDR	Output	Regulated Power Supply for RTC logic
ALARM*	Output Open Drain	Alarm output. Open drain low active signal External pull-up should not be higher than RTCVDD voltage.
X32IN	Input	32KHz Crystal Input This is connected with 32Khz crystal.
X32OUT	Output	32KHz Crystal Output This is connected with 32Khz crystal.

Note: All RTC signals and RTCVDD are required independent Voltage from the system.

3.3.17. SIO Interface

SIO Interface		
Signal Name	**I/O**	**Function**
RXD0	Input PU	SIO channel 0 Reception Data This is the Serial data input signal.
TXD0	Tri-state Output PU	SIO channel 0 Transfer Data This is the Serial data output signal. PU is for Boot Configuration
CTS0	Input PU	SIO channel 0 Clear To Send signal
RTS0	Tri-state Output PU	SIO channel 0 Ready To Send signal.
RXD1	Input PU	SIO channel 1 Reception Data This is the Serial data input signal.
TXD1	Tri-state Output PU	SIO channel 1 Transfer Data This is the Serial data output signal. PU is for Boot Configuration
RXD2	Input PU	SIO channel 2 Reception Data This is the Serial data input signal.
TXD2	Tri-state Output PU	SIO channel 2 Transfer Data This is the Serial data output signal.
RXD3	Input PU	SIO channel 3 Reception Data This is the Serial data input signal.
TXD3	Tri-state Output PU	SIO channel 3 Transfer Data This is the Serial data output signal.

3.3.18. Timer Interface

Timer Interface		
Signal Name	**I/O**	**Function**
TIMER [1:0]	Output	Timer Output This is the timer output signal.
WDRST*	OD Output	Watchdog Reset This is the Watchdog Reset output signal.

3.3.19. Interrupt Signals

Interrupt Signals		
Signal Name	**I/O**	**Function**
NMI*	Input PU	Non-Maskable Interrupt This is the Non-Maskable Interrupt signal.
INT[2:0]*	Input PU	External Interrupt Request These are the external interrupt signals.
INTA* INTB* INTC* INTD*	Input PU	External Interrupt dedicated for PCI

3.3.20. PLL Power and Ground

PLL Power and Ground		
Signal Name	**I/O**	**Function**
PLL12Vdd_A	–	Power Supply for PLL1 and PLL2 This is the PLL analog power pin. PLL will be modified version of PLL3M32E, which will be 1.2V power supply voltage.
PLL12Vss_A	–	Ground for PLL1 and PLL2 This is the PLL analog ground pin.
PLL3Vdd_A	–	Power Supply for PLL3 (Audio) This is the PLL analog power pin. PLL will be modified version of PLL3M32E, which will be 1.2V power supply voltage.
PLL3Vss_A	–	Ground for PLL3 (Audio) This is the PLL analog ground pin.

3.3.21. TEST and EJTAG Debugging Interface

TEST and EJTAG Debugging Interface		
Signal Name	**I/O**	**Function**
TEST Signals		
TEST [0]*	Input PU	Test Mode Initiate This is the Test pin. Fix this pin to either Open or to the "H" level.
BYPASSPLL*	Input PU	Bypass PLL for test This input is used in the test mode. Fix this signal to the "H" level for the normal operation.
div_RST*	Input PU	Clock Generator Reset This signal initializes the clock generator.
JTAG and EJTAG signals		
TRST*	Input PD	JTAG Reset Input This signal is asynchronous reset input of the TAP Controller. This input has pull-down device, however it is recommended to attach additional pull down resister to prevent accidental assertion.
EJRST*	Input PD	EJTAG Reset Input This signal is asynchronous reset input of the Debugging Support Unit (DSU). This input has pull-down device, however it is recommended to attach additional pull down resister to prevent accidental assertion. When connecting an EJTAG probe, take a measure such as connecting pull-up resistance to prevent this signal from low-level. When this signal is deasserted, G-Bus time-out detection becomes invalid. Note for tamper protection. In production model, It is highly recommended to connect this terminal to low level right under package without any routing on the motherboard.
TCK	Input PU	JTAG or EJTAG Clock Input During EJTRST is high level, this clock goes only to EJTAG TAP and EJTRST is low level, it goes only JTAG TAP. This is the JTAG clock input signal. JTDI or JTMS data is fetched at the rising edge of this clock.
TMS	Input PU	JTAG Command Input This signal mainly controls the status shifts of the TAP Controller State Machine.
TDI/(DINT*)	Input PU	JTAG Data Input/Debugging Interrupt Input When PC Trace mode is not selected, this signal is a JTAG Data input. This signal inputs serial data to the JTAG Data/Instruction Registers. When PC Trace mode is selected, this signal is an Interrupt input signal used to cancel PC Trace mode for the debug unit.
TDO/(TPC[0])	Output	JTAG Data Output/PC Trace Output When PC Trace mode is not selected, this signal is a JTAG Data output. Data is output by means of serial scan. When PC Trace mode is selected, this signal outputs nonconsecutive program counter values synchronous to DCLK.
Additional Signals for PC Trace Function		
DINT*	Input	This is the Interrupt input signal that is used for changing the Debugging Unit state from the PC Trace mode to Off.
TPC[3:1]	Output	PC Trace Output This signal outputs nonconsecutive program counter values synchronous to DCLK.
DCLK	Output	Debugging Clock Output This signal is clock output for the real-time debugging system. When the PC Trace mode is On, TPC[3:1] and the PCST signal are outputted synchronously. This clock is 1/3 the frequency of the TX49/H4 Core operating clock (CPUCLK).
PCST [8:0]	Output	PC Trace Status Output These signals output information such as the PC Trace Status.

3.4. Pin Multiplexing

3.4.1. Pin Multiplex for GPIO (Miscellaneous)

PCI Controller has two modes. One is Host mode and the other is Satellite mode. In the Satellite mode, the internal arbiter cannot be used and external arbiter should be provided. Some of the internal arbiter terminal switched to other signals. This configuration can be set by CCRG.PCIARB (PCIARB bit in Chip Configuration Register).
Table 3-3 below shows these multiplexing.

Table 3-3 GPIO Signal Multiplexing

Default Signal Name	GPIO (Boot Default)	Function Select or Satellite Mode	Ball
GP00	GPIO00	Dedicated GPIO	A15
GP01	GPIO01	Dedicated GPIO	D14
GP02	GPIO02	MCLKA external input/output	C14
GP03	GPIO03	MCLKB external input/output	B14
ACRESET*	GPIO04	See 3.4.2	C18
BITCLK	GPIO05	See 3.4.2	B17
SDIN[0]	GPIO06	See 3.4.2	C17
SDOUT	GPIO07	See 3.4.2	A18
SYNC	GPIO08	See 3.4.2	B18
SDIN[1]	GPIO09	See 3.4.2	D17
I2C_CLK	GPIO10	I2C_CLK	A17
I2C_DO	GPIO11	I2C_DO	C16
SPIOUT	GPIO12	RXD2/ CTS0	A16
SPICS0	GPIO13	TXD2/ RTS0	D15
SPIIN	GPIO14	RXD3 / CIR	B16
SPICLK	GPIO15	TXD3	C15
REQ[0]*	REQ[0]*	REQ*	L2
GNT[0]*	GNT[0]*	GNT*	L3
REQ[1]*	REQ[1]*	INTOUT*	K1
GNT[1]*	GNT[1]*	IDSEL	L4
REQ[2]*	REQ[2]*	GPIO16	K4
GNT[2]*	GNT[2]*	GPIO17	K3
REQ[3]*	REQ[3]*	GPIO18	J3
GNT[3]*	GNT[3]*	GPIO19	J2
REQ[4]*	REQ[4]*	GPIO20	H2
GNT[4]*	GNT[4]*	GPIO21	J4
REQ[5]*	REQ[5]*	GPIO22	H4
GNT[5]*	GNT[5]*	GPIO23	H3

Note 1: SIO2, 3 are selected by the combinations of SPIMODE, SIO2MODE[1:0], SIO3MODE bits in Pin Configuration Register (Chapter 7)

Note 2: I2C function can be selected by setting I2CMODE bit in Pin Configuration Register (Chapter 7)

Note 3: GP00 to GP03 are Hiz mode at boot time.

3.4.2. Pin Multiplexing for ACLINK and I2S

Table 3-4 I2S Signals

Default Signal Name	GPIO (Boot Default	ACLINK mode	I2S 2ch	I2S 5.1ch
ACRESET*	GPIO04	ACRESET*	I2S_WS0	I2S_ WS
BITCLK	GPIO05	BITCLK	I2S_SCK0	I2S_SCK
SDIN[0]	GPIO06	SDIN[0]	I2S_SD0	I2S_SD0
SDOUT	GPIO07	SDOUT	I2S_WS1	I2_SD1
SYNC	GPIO08	SYNC	I2S_SCK1	--
SDIN[1]	GPIO09	SDIN[1]	I2S_SD1	I2S_SD2

Note: All GPIO port is Hiz mode at boot time.

3.4.3. Pin Multiplex for ATA100-0 (Channel 0)

At the Boot time, the ATA100-0 port will be configured as GPIO channel-0. This port will be activated by setting ATA0MODE bit in Pin Configuration Register (Chapter 7)

Table 3-5 Pin Multiplex for ATA100-0 Channel

Default Signal Name	GPIO (Boot default)	ATA100-0	Ball
A0_DASP*	G0PIO00	A0_DASP*	AF3
A0_DCS1*	G0PIO01	A0_DCS1*	AD4
A0_DCS0*	G0PIO02	A0_DCS0*	AE4
A0_DA2	G0PIO03	A0_DA2	AF4
A0_DA0	G0PIO04	A0_DA0	AC5
A0_PDIAG*	G0PIO05	A0_PDIAG*	AD5
A0_DA1	G0PIO06	A0_DA1	AE5
A0_DINTR	G0PIO07	A0_DINTR	AF5
A0_DMACK*	G0PIO08	A0_DMACK*	AC6
A0_DIORDY	G0PIO09	A0_DIORDY	AD6
A0_DIOR*	G0PIO10	A0_DIOR*	AE6
A0_DIOW*	G0PIO11	A0_DIOW*	AF6
A0_DMAREQ	G0PIO12	A0_DMAREQ	AC7
A0_DD15	G0PIO13	A0_DD15	AD7
A0_DD00	G0PIO14	A0_DD00	AF7
A0_DD14	G0PIO15	A0_DD14	AC8
A0_DD01	G0PIO16	A0_DD01	AD8
A0_DD13	G0PIO17	A0_DD13	AE8
A0_DD02	G0PIO18	A0_DD02	AF8
A0_DD12	G0PIO19	A0_DD12	AC9
A0_DD03	G0PIO20	A0_DD03	AD9
A0_DD11	G0PIO21	A0_DD11	AE9
A0_DD04	G0PIO22	A0_DD04	AF9
A0_DD10	G0PIO23	A0_DD10	AC10
A0_DD05	G0PIO24	A0_DD05	AD10
A0_DD09	G0PIO25	A0_DD09	AE10
A0_DD06	G0PIO26	A0_DD06	AF10
A0_DD08	G0PIO27	A0_DD08	AC11
A0_DD07	G0PIO28	A0_DD07	AD11
A0_DRST*	G0PIO29	A0_DRST*	AE11

Note : All GPIO port is Hiz mode at boot time.

3.4.4. Pin Multiplex for ATA100-1 (Channel 1)

At the Boot time, the ATA100-1 port will be configured as GPIO channel-1. This port will be activated by the setting combinations of ATA1MODE, ET0MODE, and ET1MODEbit in Pin Configuration Register (Chapter 7)

Table 3-6 Function Selecting Map

ATA1MODE	ET0MODE	ET1MODE	Function Result
0	0	0	All GPIO (Boot-up default)
0	1	0	Ethernet Channel 0 and GPIO
0	0	1	Ethernet Channel 1 and GPIO
0	1	1	Ethernet 0,1 and GPIO
1	0	0	ATA100-1 only

Table 3-7 Pin Multiplex for ATA100-1 Channel

Default Signal Name	GPIO (Boot Default)	ATA100-1	Ethernet MAC	Ball
A1_DASP*	G1PIO00	A1_DASP*	E0RCLKI	AF11
A1_DCS1*	G1PIO01	A1_DCS1*	-	AC12
A1_DCS0*	G1PIO02	A1_DCS0*	E0CRS_DV	AD12
A1_DA2	G1PIO03	A1_DA2	E0RXD[0]	AF12
A1_DA0	G1PIO04	A1_DA0	E0RXD[1]	AC13
A1_PDIAG*	G1PIO05	A1_PDIAG*	E0TX_EN	AD13
A1_DA1	G1PIO06	A1_DA1	E0TXD[0]	AE13
A1_DINTR	G1PIO07	A1_DINTR	E0TXD[1]	AF13
A1_DMACK*	G1PIO08	A1_DMACK*	E0RX_ER	AF14
A1_DIORDY	G1PIO09	A1_DIORDY	E0MDC	AE14
A1_DIOR*	G1PIO10	A1_DIOR*	E0MDIO	AD14
A1_DIOW*	G1PIO11	A1_DIOW*	E1RCLKI	AC14
A1_DMAREQ	G1PIO12	A1_DMAREQ	-	AF15
A1_DD15	G1PIO13	A1_DD15	E1CRS_DV	AD15
A1_DD00	G1PIO14	A1_DD00	E1RXD[0]	AC15
A1_DD14	G1PIO15	A1_DD14	E1RXD[1]	AF16
A1_DD01	G1PIO16	A1_DD01	E1TX_EN	AE16
A1_DD13	G1PIO17	A1_DD13	E1TXD[0]	AD16
A1_DD02	G1PIO18	A1_DD02	E1TXD[1]	AC16
A1_DD12	G1PIO19	A1_DD12	E1RX_ER	AF17
A1_DD03	G1PIO20	A1_DD03	E1MDC	AE17
A1_DD11	G1PIO21	A1_DD11	E1MDIO	AD17
A1_DD04	G1PIO22	A1_DD04	G1PIO22	AC17
A1_DD10	G1PIO23	A1_DD10	G1PIO23	AF18
A1_DD05	G1PIO24	A1_DD05	G1PIO24	AE18
A1_DD09	G1PIO25	A1_DD09	G1PIO25	AD18
A1_DD06	G1PIO26	A1_DD06	G1PIO26	AC18
A1_DD08	G1PIO27	A1_DD08	G1PIO27	AF19
A1_DD07	G1PIO28	A1_DD07	G1PIO28	AE19
A1_DRST*	G1PIO29	A1_DRST*	G1PIO29	AD19

Note: All GPIO port is Hiz mode at boot time.

3.4.5. Pin Multiplex for Video port

Video port (17 signal) will be GPIO Channel-2 at boot up time. This port can be configured by setting VPSMODE and VSSMODE bit in Pin Configuration Register (See Chapter 7)

PC Trace Mode is activated by Boot Configuration. Related mode bit is PTSEL, which is in the Chip Configuration Register (CCFG). Please see Chapter 7 for detail.

Table 3-8 Function Selecting Map

PTSEL	VSSMODE	VPPMODE	Function Result
0	0	0	All GPIO (Boot-up default)
0	0	1	1-Parallel ports and GPIO
0	1	0	3-Serial port and GPIO
0	1	1	1-Parallel and 1-Serial port
1	Don't care	Don't care	PC Trace Mode

Table 3-9 Pin Multiplex for Video ports

Default Signal Name	GPIO (Boot default)	PC Trace Mode	1-Serial, 1-Parallel	3-Serial Mode	Ball
VDCLKIN1(Note 1)	G2PIO00	G2PIO00	VDCLKIN1	VDCLKIN1	AC19
VDCLKO1	G2PIO01	DCLK	VDCLKO1	VDCLKO1	AF20
VDVLD1	G2PIO02	G2PIO02	VDVLD1	VDVLD1	AD20
VDPSN1	G2PIO03	TPC[1]	VDPSN1	VDPSN1	AC20
VDS1	G2PIO04	TPC[2]	VDS1	VDS1	AF21
VDCLKIN	G2PIO05	TPC[3]	VDCLKIN	VDCLKIN2	AE21
VDCLKO	G2PIO06	PCT[0]	VDCLKO	VDCLKO2	AD21
VDVLD	G2PIO07	PCT[1]	VDVLD	VDVLD2	AC21
VDPSN	G2PIO08	PCT[2]	VDPSN	VDPSN2	AF22
VD0	G2PIO09	PCT[3]	VD0	VDS2	AE22
VD1	G2PIO10	PCT[4]	VD1	VDCLKIN3	AD22
VD2	G2PIO11	PCT[5]	VD2	VDCLKO3	AC22
VD3	G2PIO12	PCT[6]	VD3	VDVLD3	AF23
VD4	G2PIO13	PCT[7]	VD4	VDPSN3	AE23
VD5	G2PIO14	PCT[8]	VD5	VDS3	AD23
VD6	G2PIO15	G2PIO15	VD6	G2PIO15	AF24
VD7	G2PIO16	G2PIO16	VD7	G2PIO16	AE24

Note 1: Serial port channel 1 can perform Input and Output.

Note 2: All GPIO port is High-Z mode at boot time.

3.4.6. Pin Multiplexing for ISA

ISA interface signals are sharing with NOR/NAND/RAM interface (EBC interface). Table 3-10 is the list all of signals that are used in ISA mode. Figure 9-1 in chapter 9 shows how ISA connects to EBC interface signals.

At the Boot time

- DMAREQ[2] are default to be input mode. The state of DMAREQ[2] is HIGH since there is Pull-Up on the internal IO pad.
- DMAACK[2] are default to be output mode. The state of DMAACK[2] is HIGH after RESET.

After Boot up, if the system has ISA connector on board then firmware must enable ISA mode for that channel by program to EBCCRn register bit 22. See Chapter 9 for more information

Table 3-10 ISA Signal Multiplexing

Default Signal Name	ISA	Ball Name
DMAREQ[2]	IOR*	A5
DMAACK[2]	IOW*	B5
BE[1]	BHE*	C1
ACK*/READY	WAIT	D11
SADB[15:0]	DATA[15:0]	A11,B11,C11,A10,B10 ,C10,D10,B9,C9,D9, B8,C8,D8,A7,C7,D7
SA[5:0]	SA[6:1]	C2,B3,C5,C4,A3,D5
BE[0]	SA[0]	D3

Note: Both DMAREQ[2] and DMAAC[2] used Pull-Up IO pad.

3.4.7. Pin Multiplexing for PCICLK [4:1]

Table 3-11 PCICLK signal multiplexing

Default Signal Name	BOOT Default	PCICLK ON	COMMENT
PCICLK[4]	IOSRST*	PCICLK	PCICLK is OFF and IOSRST* will be assert during RESET* assertion. During PCICLK[4] is ON, IOSRST function will be disengaged
PCICLK[3]	SYSRST*	PCICLK	PCICLK is OFF and SYSRST* will be deassert by RESET* assertion. During PCICLK[3] is ON, SYSRST function will be disengaged
PCICLK[2]	(PCICLK Stopped)	PCICLK	PCICLK is OFF at boot time
PCICLK[1]	(PCICLK Running)	PCICLK	PCICLK is ON at boot time

Chapter 4. Boot Configuration

4.1. Boot Configuration

In boot time, following major things will be configured by external signal conditions.

- *PCI Controller Mode*
  - *Host Mode*
  - *Satellite Mode*
- *Boot Device*
  - *NOR device on channel 0*
  - *ROM Device on PCI Bus (PCI Boot)*
- *CPU Clock Configuration*
  - *CPU Pipeline Clock*
  - *GBUS Clock Frequency*
  - *SYSCLK Frequency*
  - *NOR device timing set (NOR Boot)*
  - *PCI Bus Clock Frequency*
- *PC Trace Enable or not*
- *ETC.*

The SA [5:0], DMAACK [2:0], and SADB [15:0] signals (External bus signals) can also function as configuration signals for initially setting various functions upon booting the system. The states of the configuration signals immediately after the RESET* is de-asserted are read as initial values for the TX4939 internal registers.
A High signal level sets a value of 1 and a Low signal level sets a value of 0.

All configuration signals are provided with internal pull-up resistors. To drive a signal Low, pull down the corresponding pin on the board using an approx. 4.7 kΩ resistor. Driving a signal high does not require a pull-down resistor.
Any signals defined Reserved should not be pulled down.

Figure 4-1 shows Boot Configuration Timing

Table 4-1 describes each configuration signal.

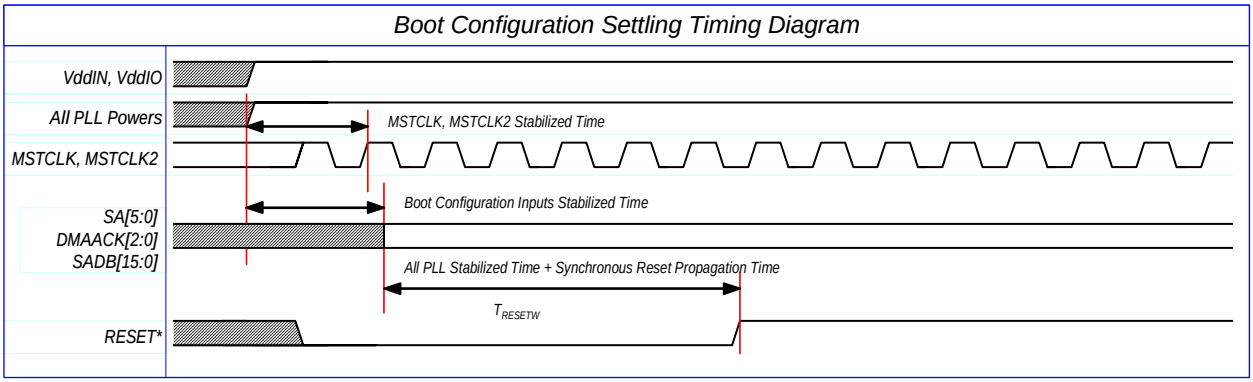


Figure 4-1 Boot Configuration Settling Timing

4.2. Boot Configuration Detail

Table 4-1 Boot Configuration Details

<table>
<tr><th>Signal</th><th>Description</th><th>Corresponding Register Bit</th></tr>
<tr><td>SA[2:0]</td><td>CPU Clock Setting (MULCLK[2:0] ==> ND[4:0]@PLL#2)
(YMULCLK[4:0] = ND[4:0])
<table>
<tr><td>SA[2:0]</td><td>ND[4:0]</td><td>PLL#2</td><td>CPU Clock</td></tr>
<tr><td>000</td><td>01000</td><td>600 MHz</td><td>300 MHz</td></tr>
<tr><td>001</td><td>01001</td><td>666 MHz</td><td>333 MHz</td></tr>
<tr><td>010</td><td>01010</td><td>733 MHz</td><td>366 MHz</td></tr>
<tr><td>011</td><td>01011</td><td>800 MHz</td><td>400 MHz</td></tr>
<tr><td>100</td><td>01100</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>101</td><td>01101</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>110</td><td>01110</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>111</td><td>00111</td><td>533 MHz</td><td>266 MHz</td></tr>
</table></td><td>MULCLK[2:0]</td></tr>
<tr><td>SA[4:3]</td><td>SYSCLK Setting</td><td>00 = DIV4 of Gbus clock
01 = DIV3 of Gbus clock
10 = DIV5 of Gbus clock
11 = DIV6 of Gbus clock</td><td>SYSSP[1:0]</td></tr>
<tr><td>SA[5]</td><td>SSCG Control</td><td>1 : Enable
0 : Disabled</td><td>SSCG</td></tr>
<tr><td>DMAACK[2]</td><td>PCI Host and Satellite selection setting
1: Host
0: Satellite</td><td></td><td>CCFG_PCIMODE</td></tr>
<tr><td>DMAACK[1:0]</td><td>CPUCLK/GBUSCLK Ratio Setting
YDIVMODE[2:0] = {DMAACK[1],DMAACK[1:0]}
<table>
<tr><td>DMAACK[1:0]</td><td>YDIVMODE[2:0]</td><td>CPUCLK/GBUSCLK</td></tr>
<tr><td>00</td><td>000</td><td>(1 / 2.0)</td></tr>
<tr><td>01</td><td>001</td><td>(1 / 3.0)</td></tr>
<tr><td>10</td><td>110</td><td>(1 / 5.0)</td></tr>
<tr><td>11</td><td>111</td><td>(1 / 6.0)</td></tr>
</table></td><td>YDIVMODE[2:0]</td></tr>
<tr><td>SADB[0]</td><td>PC Trace Configuration</td><td>1 = PC Trace Enable
0 = PC Trace Disable</td><td>CCFG_PTSEL</td></tr>
<tr><td>SADB[1]</td><td>Specifies the function of the BE[1:0]*/BWE[1:0]* pins upon booting.

1 = BWE[1:0]* (Byte Write Enable)
0 = BE[1:0]* (Byte Enable)</td><td></td><td>CCFG_BESEL</td></tr>
<tr><td>SADB[2]</td><td>Boot ACK* Input : Specifies the access mode for external bus controller channel 0.

1 = Normal mode
0 = External ACK mode</td><td></td><td>CCFG_ACKSEL</td></tr>
<tr><td>SADB[3]</td><td>Boot ROM Bus Width. Specifies the data bus width when booting from a memory device connected to the External bus controller.
<table>
<tr><td>SADB [3]</td><td>Boot ROM Bus Width</td></tr>
<tr><td>0</td><td>16 bit</td></tr>
<tr><td>1</td><td>8 bit</td></tr>
</table></td><td></td><td>CCFG_ROMW</td></tr>
<tr><td>SADB[4]</td><td>TX4939 Endian Mode : Specifies the TX4939 Endian mode.

1 = Big endian
0 = Little Endian</td><td></td><td>CCFG_ENDIAN</td></tr>
<tr><td>~SADB[5]</td><td>TX49/H4 Internal Timer Interrupt Disable : Specifies whether timer interrupts within the TX49/H4 core are enabled.

0 = Enable timer interrupts within the TX49/H4 core.
1 = Disable timer interrupts within the TX49/H4 core.</td><td></td><td>CCFG_TINTDIS</td></tr>
</table>

Signal	Description	Corresponding Register Bit
SADB[6]	Indicates the PCI bus arbiter selection setting 0 = Select external PCI bus arbiter 1 = Select built-in PCI bus arbiter	CCFG_ARB
SADB[7]	PCI Boot Option 0 = PCI Boot off 1 = PCI Boot on	CCFG_PCIBOOT
SADB[15:8]	Reserved for software setting.	CCFG_BCFG[7:0]

Chapter 5. Clock Generators

5.1. Overview

Figure 5-1 shows the top view of TX4939 internal clock generation structure. TX4939 generates all necessary clocks out of single 20 MHz clock source. There are two clock input, i.e. MSTCLK and MSTCLK2.
MSTCLK can be modulated by internal SSCG Modulator and the modulated clock is fed to the Primary Clock Generator. The Primary CG generates 400 MHz clock and divider generates 100 MHz and 66.66 MHz.

This 66.66 MHz clock is fed to Secondary Clock Generator, which generates CPUCLK, GBUSCLK, DDR Clock, and forth. The CPUCLK frequency and the ratio between CPUCLK and GBUS/DDR Clock are defined at boot configuration. See Chapter 4. Boot Configuration.

The 66.66 MHz is also used for PCI Clock either 66.66 MHz or 33.33 MHz. This PCI Clock distribution does not have any synchronization with PCI Logic. It simply serves as PCI Clock source. There are four (4) PCI Service Clocks.

Instead, the clock signal input to the PCICLKIN terminal has synchronizer (De-Skew Circuit), where automatically internal PCI clock synchronizes with this input clock.

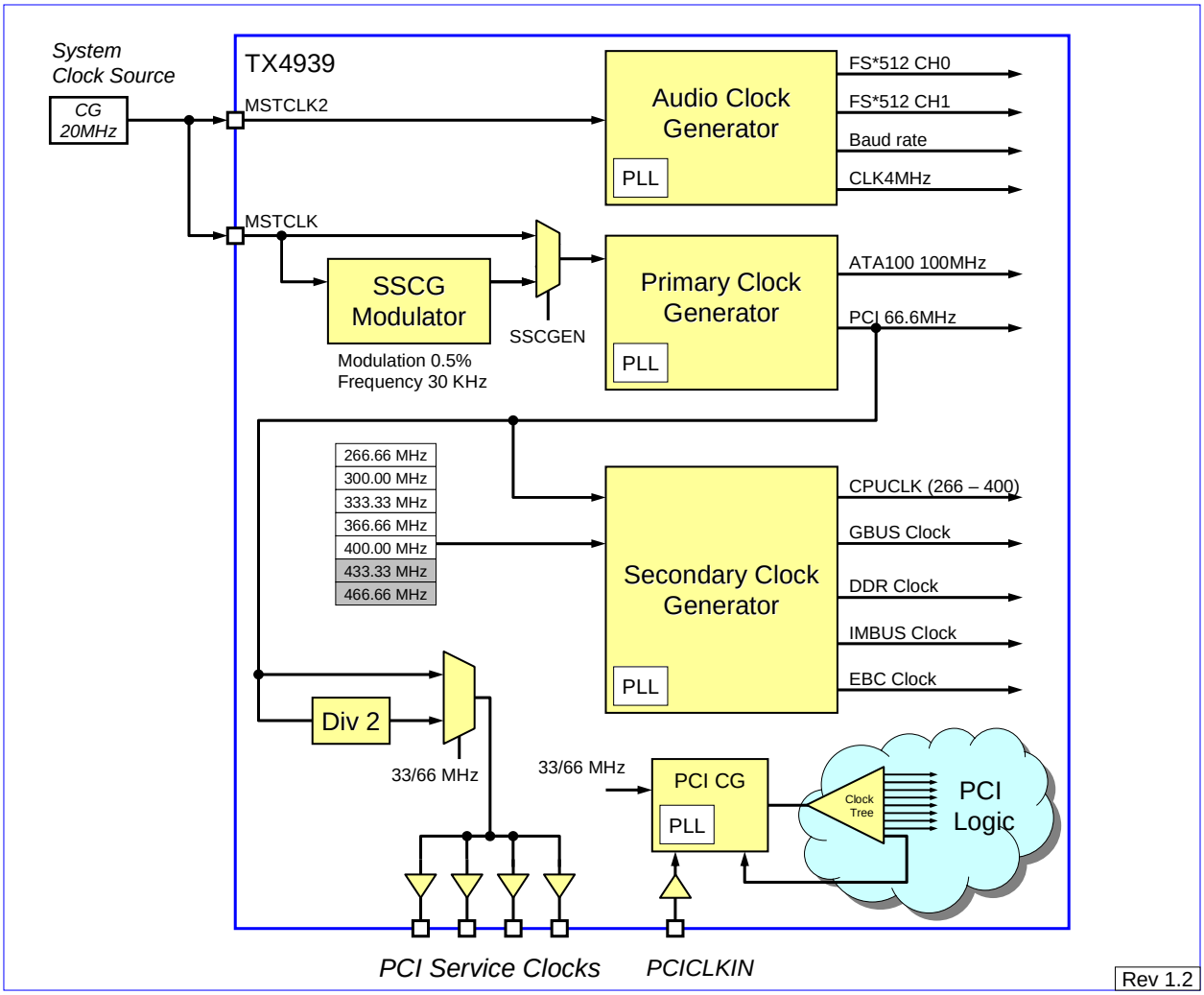


Figure 5-1 Overview of Clocking System in TX4939

5.2. Master Clock Generator

Internal Clock generator generates all necessary clocks out of 20 MHz source clock. There are two 20 MHz clock input terminals, such as MSTCLK and MSTCLK2. MSTCLK input is for generating ATA100 (100 MHz), PCICLK (66.6 MHz), CPUCLK (333 or 400 MHz), and the rest of clocks except UART and audio sampling clock. This clock generator has SSCG modulator to provide spread spectrum clock for reducing EMI of the system.

MSTCLK2 input is connected MCLK generator that generates Baud-rate clock for UART and two sets of audio sampling clocks. Figure 5-2 is the simplified diagram of master clock generator.

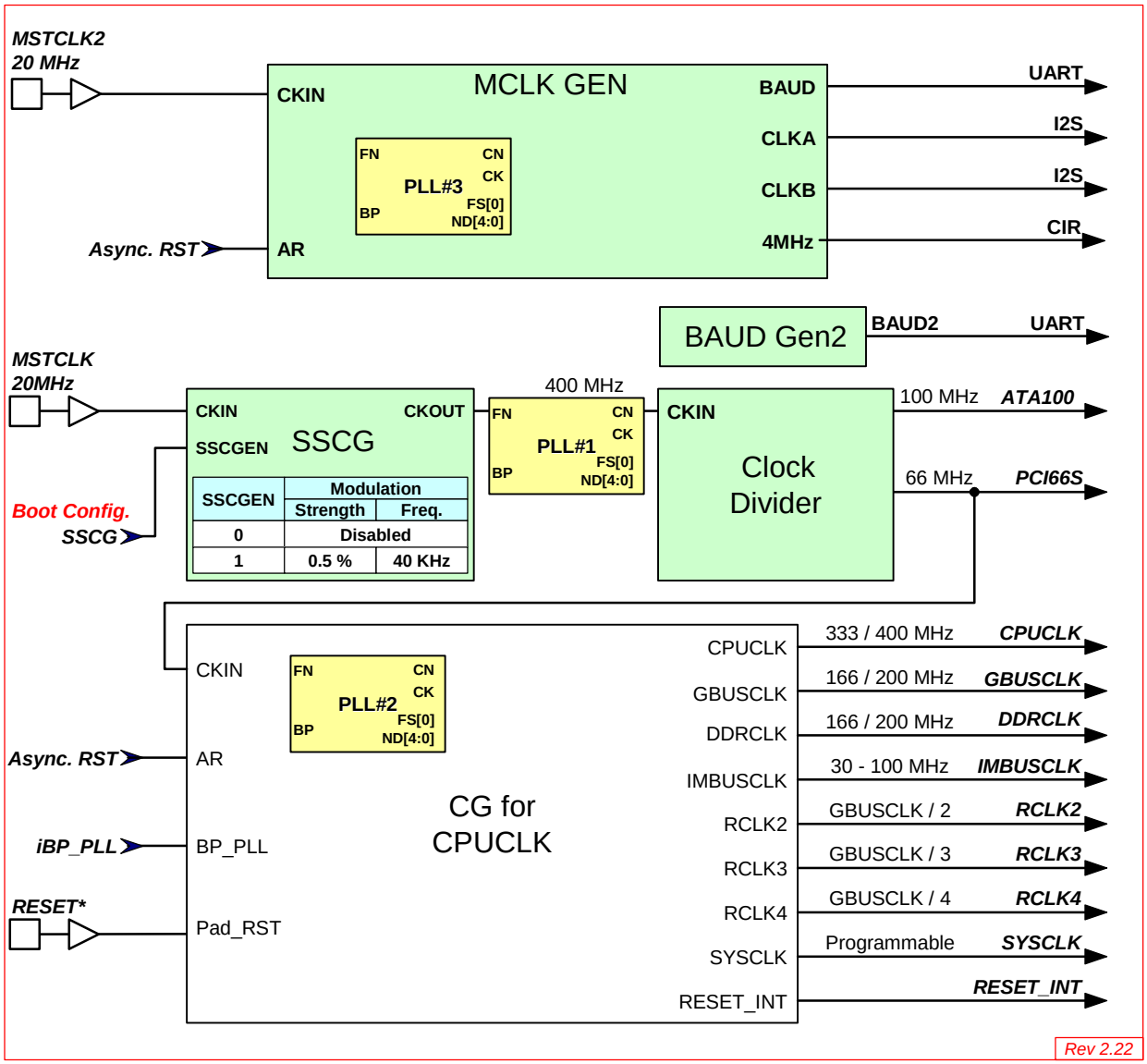


Figure 5-2 Master Clock Generator

5.3. Audio Clock Generator

5.3.1. Features

Audio Clock Generator (MCLK) generates multiple clocks for audio subsystem like I2S and optional baud rate clock. Diagram below shows the overall structure of the TX4939 Audio Clock Generator.

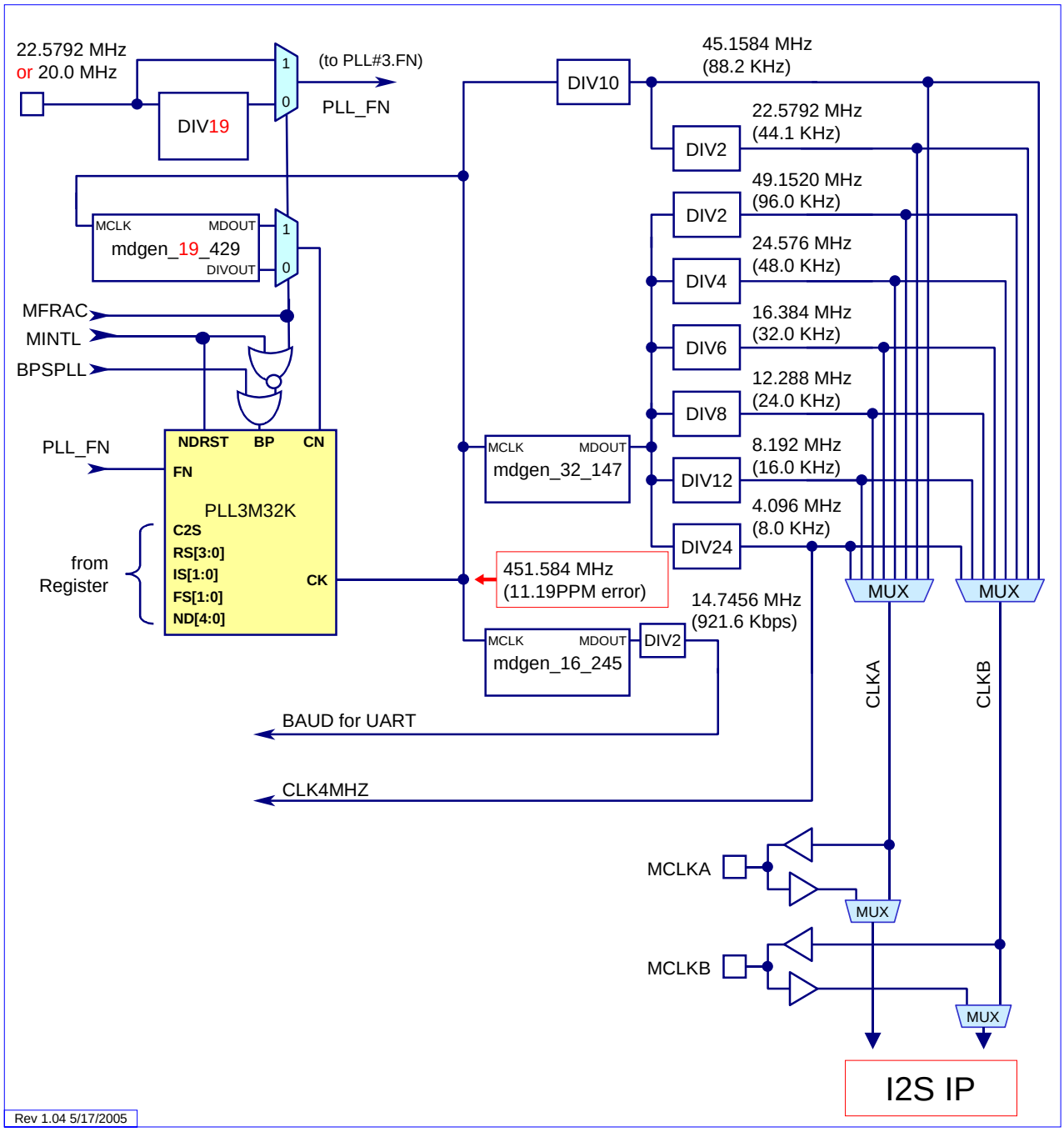


Figure 5-3 Diagram of Audio Clock Generator

5.3.2. Source Clock Generation

There are several method provided to generate source clock. One of method described in Table 5-1 should be chosen based on the requirement of source clock. The jitter value will be notified after sample evaluation.

Table 5-1 Methods for Source Clock Generation

MSTCLK2	Method	F_{SOURCE}	COMMENT (Issues)
20,000,000 Hz	Fractional N PLL	451,584,000 Hz	(1) Jitter generated by Fractional PLL. (2) Frequency Error (-11.2 PPM).
	Integer N PLL	451,584,000 Hz	(1) Jitter generated by Very High Frequency Multiplier. (2) Frequency Error (-11.2 PPM)
15,000,000 Hz	Fractional N PLL	338,688,000 Hz	(1) Jitter generated by Fractional PLL. (2) Frequency Error (-11.2 PPM).
	Integer N PLL	338,688,000 Hz	(1) Jitter generated by Very High Frequency Multiplier. (2) Frequency Error (-11.2 PPM)
22,579,200 Hz Note 1	Simple X20 Multiplier	451,584,000 Hz	(1) Clean Source Clock. (2) Extra Crystal Note 1
	Simple X15 Multiplier	338,688,000 Hz	(1) Clean Source Clock. (2) Extra Crystal Note 1

Note 1 Initial Version of TX4939 does not support this function.

The Audio Clock Generator generates either 451,584,000 Hz or 338,688,000 Hz Clock as source clock (F_{SOURCE}). The 451,584,000 Hz clock is for Fs*512 or Fs*256 as "Over Sampling Clock (F_{OS}), and 338,688,000 Hz is for Fs*768 or Fs*384. Table 5-2 describes those relations.

Table 5-2 Relation between source clock and result frequency

F_{SOURCE}	Divide	Divide	Multiply	F_{OS}	Fs*512	Fs*256
451,584,000 Hz	10	1	1	45,158,400 Hz	88,200 Hz	176,400 Hz
	20	1	1	22,579,200 Hz	44,100 Hz	88,200 Hz
	147	2	32	49,152,000 Hz	96,000 Hz	192,000 Hz
	147	4	32	24,576,000 Hz	48,000 Hz	96,000 Hz
	147	6	32	16,384,000 Hz	32,000 Hz	64,000 Hz
	147	8	32	12,288,000 Hz	24,000 Hz	48,000 Hz
	147	12	32	8,192,000 Hz	16,000 Hz	32,000 Hz
	147	24	32	4,096,000 Hz	8,000 Hz	16,000 Hz
	245	1	8	14,745,600 Hz	Baud rate Clock	
F_{SOURCE}	**Divide**	**Divide**	**Multiply**	**F_{OS}**	**Fs*768**	**Fs*384**
338,688,000 Hz	10	1	1	33,868,800 Hz	44,100 Hz	88,200 Hz
	20	1	1	16,934,400 Hz	22,050 Hz	44,100 Hz
	147	2	32	36,864,000 Hz	48,000 Hz	96,000 Hz
	147	4	32	18,432,000 Hz	24,000 Hz	48,000 Hz
	147	6	32	12,288,000 Hz	16,000 Hz	32,000 Hz
	147	8	32	9,216,000 Hz	12,000 Hz	24,000 Hz
	147	12	32	6,144,000 Hz	8,000 Hz	16,000 Hz
	147	24	32	3,072,000 Hz	4,000 Hz	8,000 Hz
	245	1	8	11,059,200 Hz	Baud rate Clock	

Depend on the Source Clock frequency, the Baud Rate Source Clock also varies. The Baud Rate Source Clock is either 14,745,600 Hz or 11,059,200 Hz. Table 5-3 describes the relation between UART Divisor and Baud Rate. In this table, "UART Divisor" represents the product of prescaler and divisor.

Table 5-3 Corresponding Prescaler value for each source clock.

F_{SOURCE}	BAUD RATE	Division	Pre-scalar	Divisor
14,745,600 Hz	921,600 bps	1	Bypass pre-scalar and divider	
	460,800 bps	2	1/2	1
	230,400 bps	4	1/2	2
	115,200 bps	8	1/2	4
	57,600 bps	16	1/2	8
	38,400 bps	24	1/2	12
	28,800 bps	32	1/2	16
	14,400 bps	64	1/2	32
	9,600 bps	96	1/2	48
	4,800 bps	192	1/2	96
	2,400 bps	384	1/2	192
	1,200 bps	768	1/8	96
	600 bps	1536	1/8	192
	300 bps	3072	1/32	96
F_{SOURCE}	BAUD RATE	Division	Pre-scalar	Divisor
11,059,200 Hz	921,600 bps	N/A	N/A	N/A
	460,800 bps	N/A	N/A	N/A
	230,400 bps	3	N/A	N/A
	115,200 bps	6	1/2	3
	57,600 bps	12	1/2	6
	38,400 bps	18	1/2	9
	28,800 bps	24	1/2	12
	14,400 bps	48	1/2	24
	9,600 bps	72	1/2	36
	4,800 bps	144	1/2	72
	2,400 bps	288	1/2	144
	1,200 bps	576	1/8	72
	600 bps	1152	1/8	144
	300 bps	2304	1/32	72

5.3.3. Frequency Error in the Generation of Source Clock out of 20 MHz

Followings are the explanation of the technology of fractional N PLL, which generate 451,584,000 Hz clock (F_{SOURCE}). Figure 5-4 shows the block diagram of this fractional N PLL.

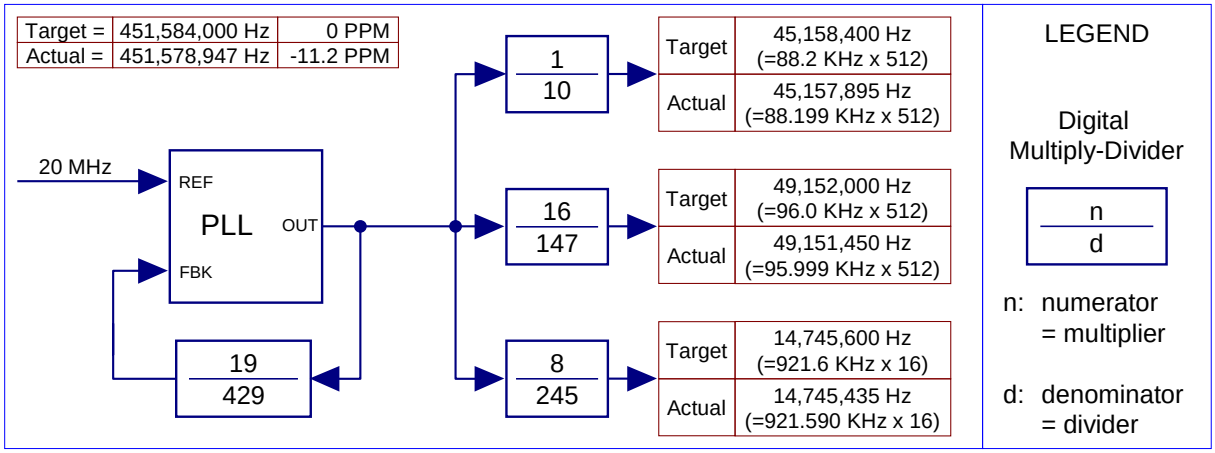


Figure 5-4 Block Diagram of Fractional N PLL

The ideal value of Multiply-Divider for this PLL is "625/7065" (n=625, d=7065). However, this value of multiply-divider would create undesirable phase jitter with very low frequency element. Therefore, the actual circuit uses "19/429" in stead. This approximation has -11.2 PPM error against target frequency of 451,584,000 Hz. Table 5-4 shows those actual frequencies.

Table 5-4 Frequency Error in the Final Clocks

Target Sampling Clock	Result Logic Clock	Actual Sampling Clock
96,000 Hz	49.151450 MHz	95,998.93 Hz
88,200 Hz	45.157895 MHz	88,199.01 Hz
48,000 Hz	24.575725 MHz	47,999.46 Hz
44,100 Hz	22.578947 MHz	44,099.51 Hz
32,000 Hz	16.383817 MHz	31,999.64 Hz
24,000 Hz	12.287863 MHz	23,999.73 Hz
22,050 Hz	11.289474 MHz	22,049.75 Hz
16,000 Hz	8.191908 MHz	15,999.82 Hz
8,000 Hz	4.09599 MHz	7,999.98 Hz
921,600 Hz	14.745435 MHz	921,589.69 Hz

It is the fact that the fractional N PLL might generate clock with some level of jitter. However, it depends on the application if this jitter value would be acceptable or not. In the case that the application required very clean clocks, this PLL should have alternate PLL mode, such as Integer PLL. Since 19/429 is about 1/22, the integer multiplier is better to close to 22. In this design, X20 is chosen for that mode. In this mode, 20 MHz external clock should be changed to 22.5792 MHz instead. This clock can be available from any of Audio Clock Generator.

5.3.4. Restriction of Audio Clock for Over Sampling

Since a digital clock multiplier is used, i.e. mdgen_32_147 in Figure 5-3 Diagram of Audio Clock Generator, the source clock, 451.584 MHz or 338.688 MHz has intrinsic jitter.

Note that this mdgen_32_147 is used to generate the source clock for 96KHz, 48KHz, 32KHz, 24KHz, 16KHz, and 8KHz Fs frequency. These clocks might be affected and not for 88.2KHz neither 44.1KHz.

Table 5-5 mdgen_32_147 Operation explains the operation of multiplier/divider. The mdgen_32_147 counts the input clock, i.e. 451.584 MHz and the counter value will return to zero when it reaches 147. Output pulse will be generated 32 times within every 147 count cycles. The column "SEQ" means the cycle number of 32 output clocks. The column "Real Value" is the theoretical figure of clock timing. Actual clock is generated at the timing displayed at the column "Pseudo". The column "Error" shows the difference between theoretical and actual value.

Table 5-5 mdgen_32_147 Operation

SEQ	Real Value	Pseudo	Error
0	0.0000	0	0.00 ns
1	4.5938	5	-0.90 ns
2	9.1875	9	0.42 ns
3	13.7813	14	-0.48 ns
4	18.3750	18	0.83 ns
5	22.9688	23	-0.07 ns
6	27.5625	28	-0.97 ns
7	32.1563	32	0.35 ns
8	36.7500	37	-0.55 ns
9	41.3438	41	0.76 ns
10	45.9375	46	-0.14 ns
11	50.5313	51	-1.04 ns
12	55.1250	55	0.28 ns
13	59.7188	60	-0.62 ns
14	64.3125	64	0.69 ns
15	68.9063	69	-0.21 ns
16	73.5000	74	-1.11 ns
17	78.0938	78	0.21 ns
18	82.6875	83	-0.69 ns
19	87.2813	87	0.62 ns
20	91.8750	92	-0.28 ns
21	96.4688	96	1.04 ns
22	101.0625	101	0.14 ns
23	105.6563	106	-0.76 ns
24	110.2500	110	0.55 ns
25	114.8438	115	-0.35 ns
26	119.4375	119	0.97 ns
27	124.0313	124	0.07 ns
28	128.6250	129	-0.83 ns
29	133.2188	133	0.48 ns
30	137.8125	138	-0.42 ns
31	142.4063	142	0.90 ns
32	147.0000	147	0.00 ns

These errors are considered as jitter in the clock. This multiplication factor is power of 2, these jitter will be disappeared after dividing by series of divide by 2. For example, the jitter will disappear at 32fs for 96KHz sampling. This jitter may not affect sampling operation at the normal over sampling timing. But it might affect the operation of Analog-Digital conversion with Sigma-Delta system.

Next page shows detail of this jitter.

Table 5-6 Jitter Value in case of 451.584 MHz Source Clock describes jitter values in various sampling clocks generated by divide by 2. The intermediate clock, i.e. 98.304 MHz, corresponds to 1,024fs for 96KHz sampling. This clock has large jitter in every steps. This jitter will disappear when it reaches to 32fs for 96KHz sampling (32fs = 3.072MHz). This frequency corresponds to 64fs for 48KHz sampling.

As jitter exists 64fs for 96KHz sampling, it may affect total harmonic distortion.

Table 5-6 Jitter Value in case of 451.584 MHz Source Clock

Source	(32/147)	Div 2	Div 2	Div 2	Div 2	Div 2	Div 2
451.584MHz	98.304MHz	49.152MHz	24.576MHz	12.288MHz	6.144MHz	3.072MHz	1.536MHz
192.000 KHz	512fs	256fs	128fs	64fs	32fs	16fs	8fs
96.000 KHz	1,024fs	512fs	256fs	128fs	64fs	32fs	16fs
48.000 KHz	2,048fs	1,024fs	512fs	256fs	128fs	64fs	32fs
Cycle - 00	-	-	-	-	-	-	-
Cycle - 01	-0.90ns	0.42ns	0.83ns	-0.55ns	-1.11ns	-	-
Cycle - 02	0.42ns	0.83ns	-0.55ns	-1.11ns	-	-	-
Cycle - 03	-0.48ns	-0.97ns	0.28ns	0.55ns	-1.11ns	-	-
Cycle - 04	0.83ns	-0.55ns	-1.11ns	-	-	-	-
Cycle - 05	-0.07ns	-0.14ns	-0.28ns	-0.55ns	-1.11ns	-	-
Cycle - 06	-0.97ns	0.28ns	0.55ns	-1.11ns	-	-	-
Cycle - 07	0.35ns	0.69ns	-0.83ns	0.55ns	-1.11ns	-	-
Cycle - 08	-0.55ns	-1.11ns	-	-	-	-	-
Cycle - 09	0.76ns	-0.69ns	0.83ns	-0.55ns	-1.11ns	-	-
Cycle - 10	-0.14ns	-0.28ns	-0.55ns	-1.11ns	-	-	-
Cycle - 11	-1.04ns	0.14ns	0.28ns	0.55ns	-1.11ns	-	-
Cycle - 12	0.28ns	0.55ns	-1.11ns	-	-	-	-
Cycle - 13	-0.62ns	0.97ns	-0.28ns	-0.55ns	-1.11ns	-	-
Cycle - 14	0.69ns	-0.83ns	0.55ns	-1.11ns	-	-	-
Cycle - 15	-0.21ns	-0.42ns	-0.83ns	0.55ns	-1.11ns	-	-
Cycle - 16	-1.11ns	-	-	-	-	-	-

In case of Source Clock=338.688 MHz mode, mdgen_32_147 generates 73.728 MHz, which corresponds to 768fs for 96KHz sampling. It has certain level of jitter up to 8fs for 96KHz sampling. This is described Table 5-7 Jitter Value in case of 338.688 MHz Source Clock below.

Table 5-7 Jitter Value in case of 338.688 MHz Source Clock

Source	(32/147)	Div 3	Div 2	Div 2	Div 2	Div 2	Div 2
338.688MHz	73.728MHz	24.576MHz	12.288MHz	6.144MHz	3.072MHz	1.536MHz	0.768MHz
192.000 KHz	384fs	128fs	64fs	32fs	16fs	8fs	4fs
96.000 KHz	768fs	256fs	128fs	64fs	32fs	16fs	8fs
48.000 KHz	1,536fs	512fs	256fs	128fs	64fs	32fs	16fs
Cycle - 00	-	-	-	-	-	-	-
Cycle - 01	-1.20ns	-0.65ns	-1.29ns	0.37ns	0.74ns	-1.48ns	-
Cycle - 02	0.55ns	-1.29ns	0.37ns	0.74ns	-1.48ns	-	-
Cycle - 03	-0.65ns	1.01ns	-0.92ns	1.11ns	-0.74ns	-1.48ns	-
Cycle - 04	1.11ns	0.37ns	0.74ns	-1.48ns	-	-	-
Cycle - 05	-0.09ns	-0.28ns	-0.55ns	-1.11ns	0.74ns	-1.48ns	-
Cycle - 06	-1.29ns	-0.92ns	1.11ns	-0.74ns	-1.48ns	-	-
Cycle - 07	0.46ns	1.38ns	-0.18ns	-0.37ns	-0.74ns	-1.48ns	-
Cycle - 08	-0.74ns	0.74ns	-1.48ns	-	-	-	-
Cycle - 09	1.01ns	0.09ns	0.18ns	0.37ns	0.74ns	-1.48ns	-
Cycle - 10	-0.18ns	-0.55ns	-1.11ns	0.74ns	-1.48ns	-	-
Cycle - 11	-1.38ns	-1.20ns	0.55ns	1.11ns	-0.74ns	-1.48ns	-
Cycle - 12	0.37ns	1.11ns	-0.74ns	-1.48ns	-	-	-
Cycle - 13	-0.83ns	0.46ns	0.92ns	-1.11ns	0.74ns	-1.48ns	-
Cycle - 14	0.92ns	-0.18ns	-0.37ns	-0.74ns	-1.48ns	-	-
Cycle - 15	-0.28ns	-0.83ns	1.29ns	-0.37ns	-0.74ns	-1.48ns	-
Cycle - 16	-1.48ns	-1.48ns	-	-	-	-	-

5.3.5. Audio Clock Control Register

Audio Clock Generator has two registers to control. These registers are 32-bits wide. WORD accesses are allowed for READ/WRITE (LW or SW). Those are shown Table 5-8 below.

Table 5-8 Audio Clock Control Registers

Offset Address	Definition	Description
0xE068	MCLKOSC	Audio Clock PLL set-up
0xE070	MCLKCTL	Audio Clock Operation Control

5.3.6. MCLKOSC Register

MCLKOSC Register controls PLL set-up and configuration.

MCLKOSC	31	30	29	28	27	26	25	24
FIELD Definition	RESERVED							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
DEFAULT	0	0	0	0	0	0	0	0

MCLKOSC	23	22	21	20	19	18	17	16
FIELD Definition	RESERVED						MFRAC	MINTL
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
DEFAULT	0	0	0	0	0	0	1	1

MCLKOSC	15	14	13	12	11	10	9	8
FIELD Definition	RESERVED		FS [1:0]		RS [3:0]			
R/W	R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	1	0	1	0	0

MCLKOSC	7	6	5	4	3	2	1	0
FIELD Definition	IS [1:0]		C2S	ND [4:0]				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	1	0	1	0	0	1

Figure 5-5 Definition of MCLKOSC Register

Table 5-9 describes recommended parameter values related PLL.
Following tables, i.e. Table 5-10, Table 5-11 and Figure 5-6 are describe more detail about those parameters. These values are generated based on simulation; values are subjected to change after sample evaluation.

Table 5-9 Recommended Values for PLL Parameter

Operation Mode	MSTCLK2	Source Clock	MFRAC	MINTL	FS[1:0]	RS[3:0]	IS[1:0]	C2S	ND[4:0]
Fractional N PLL	20.0000 MHz	451.584 MHz	1'b1	1'b1	2'b00	4'b0100	2'b01	1'b0	5'b00000
Integer N PLL			1'b0	1'b1	2'b00	4'b0100	2'b01	1'b0	5'b00000
Fractional N PLL	15.0000 MHz	338.688 MHz	1'b1	1'b1	2'b01	4'b0100	2'b01	1'b1	5'b00000
Integer N PLL			1'b0	1'b1	2'b01	4'b0100	2'b01	1'b1	5'b00000
Simple X20 PLL	22.5792 MHz	451.584 MHz	1'b1	1'b0	2'b00	4'b0100	2'b01	1'b0	5'b10011
Simple X15 PLL		338.688 MHz	1'b1	1'b0	2'b01	4'b0100	2'b01	1'b1	5'b01111

Note1: Those values will be updated after ES evaluation.
Note2: Current version will not support "Simple X20, X15 PLL mode.

Table 5-10 Bit Field Definitions of MCLKOSC Register

Bit Field	Symbol	Description
31:18	Reserved	Reserved
17	MFRAC	PLL Configuration (default value : MFRAC = 1, MINTL = 1) {MFRAC, MINTL} defines PLL Configuration of the Audio Clock Generator. MFRAC = 0, MINTL = 0: Audio Clock Generator is disabled MFRAC = 0, MINTL = 1: PLL generates 451.584 MHz with "Integer N PLL" mode (CG=20 MHz) MFRAC = 1, MINTL = 0: PLL operates simple X10 mode, 22.5792 MHz CG required. MFRAC = 1, MINTL = 1: PLL generates 451.584 MHz with "Fractional N PLL" mode (CG=20 MHz)
16	MINTL	
15:14	Reserved	Reserved
13:12	FS [1:0]	PLL frequency range select This parameter has a relation ship with C2S input. Refer Table 5-11
11:8	RS [3:0]	PLL Low Pass Filter R2 selection R2 will be the value of 2.4 + RS[3:0] x 1.2 (K ohm)
7:6	IS [1:0]	PLL Charge Pump Current (Ip) select IS[1:0] / Description 2'b00 — Charge Pump Current will be X3 of 8 uA (24uA) 2'b01 — Charge Pump Current will be X2 of 8 uA (16uA) 2'b10 — Charge Pump Current will be X1 of 8 uA (8uA) 2'b11 — Disable Charge Pump
5	C2S	PLL VCO Performance Select
4:0	ND [4:0]	PLL Multiply factor Multiplier is the value of ND [4:0] + 1.

Table 5-11 Parameter FS[1:0] and C2S

FS[1:0]	C2S	Description	Low Limit	High Limit
FS=00	C2S=H	No Output Divider	450 MHz	1,000 MHz
FS=00	C2S=L	Output will be divided by 2	315 MHz	700 MHz
FS=01	C2S=H	Output will be divided by 4	225 MHz	500 MHz
FS=01	C2S=L	Output will be divided by 8	158 MHz	350 MHz
FS=10	C2S=H	No Output Divider	113 MHz	250 MHz
FS=10	C2S=L	Output will be divided by 2	79 MHz	175 MHz
FS=11	C2S=H	Output will be divided by 4	56 MHz	125 MHz
FS=11	C2S=L	Output will be divided by 8	39 MHz	88 MHz

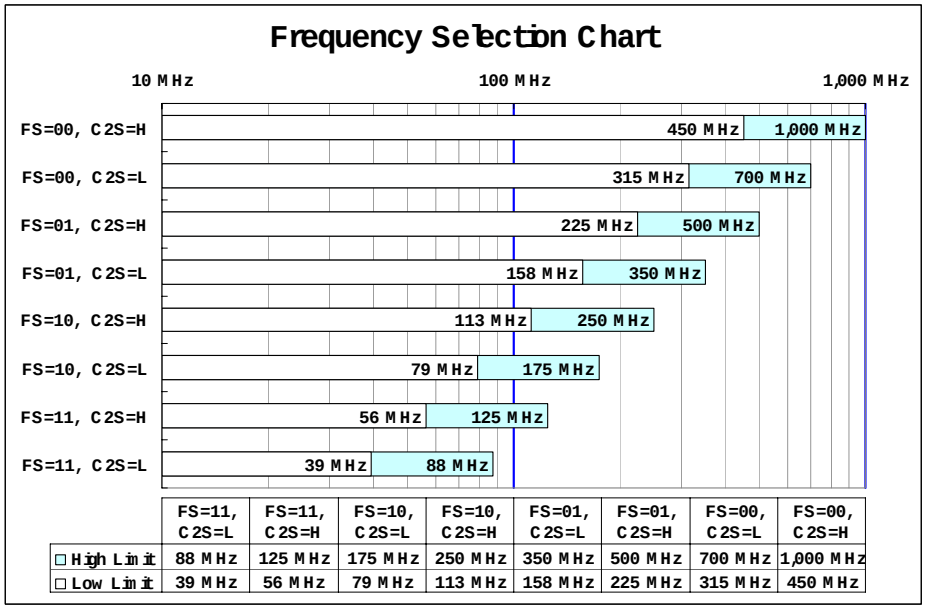


Figure 5-6 Frequency Selection related to FS[1:0] and C2S value

5.3.7. MCLKCTL Register

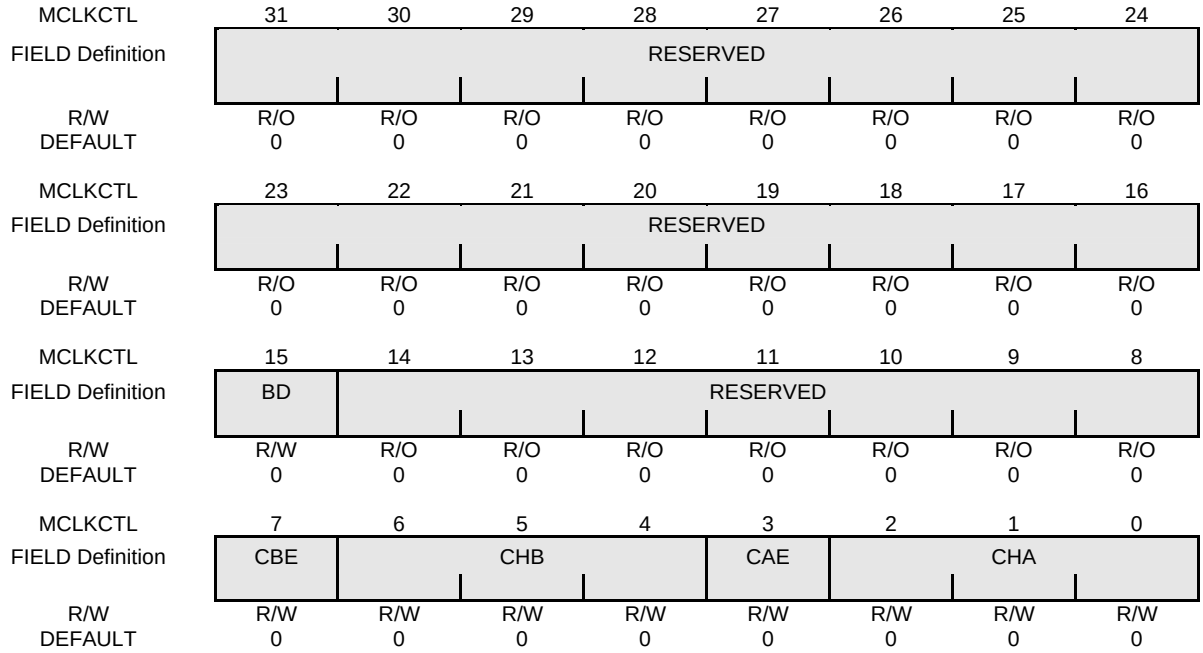


Figure 5-7 Definition of MCLKCTL Register

Table 5-12 Bit Field Definitions

Bit Field	Symbol	Description
31:16	N/A	Reserved
15	BDE	Baud Clock enable (1:Enable, 0:Disable)
14:8	N/A	Reserved
7	CBE	Channel-B Enable (1:Enable, 0:Disable)
6:4	CHB [2:0]	Channel B Frequency select CHB [2:0] = 3'b000: Select 22.5792 MHz (Fs = 44.1 KHz) CHB [2:0] = 3'b001: Select 24.576 MHz (Fs = 48.0 KHz) CHB [2:0] = 3'b010: Select 16.384 MHz (Fs = 32.0 KHz) CHB [2:0] = 3'b011: Select 12.288 MHz (Fs = 24.0 KHz) CHB [2:0] = 3'b100: Select 8.192 MHz (Fs = 16.0 KHz) CHB [2:0] = 3'b101: Select 4.096 MHz (Fs = 8.0 KHz) CHB [2:0] = 3'b110: Select 45.1584 MHz (Fs = 88.2 KHz) CHB [2:0] = 3'b111: Select 49.152 MHz (Fs = 96.0 KHz)
3	CAE	Channel-A Enable (1:Enable, 0:Disable)
2:0	CHA [2:0]	Channel A Frequency select CHA [2:0] = 3'b000: Select 22.5792 MHz (Fs = 44.1 KHz) CHA [2:0] = 3'b001: Select 24.576 MHz (Fs = 48.0 KHz) CHA [2:0] = 3'b010: Select 16.384 MHz (Fs = 32.0 KHz) CHA [2:0] = 3'b011: Select 12.288 MHz (Fs = 24.0 KHz) CHA [2:0] = 3'b100: Select 8.192 MHz (Fs = 16.0 KHz) CHA [2:0] = 3'b101: Select 4.096 MHz (Fs = 8.0 KHz) CHA [2:0] = 3'b110: Select 45.1584 MHz (Fs = 88.2 KHz) CHA [2:0] = 3'b111: Select 49.152 MHz (Fs = 96.0 KHz)

5.4. Second Baud rate Generator

TX4939 has second baud-rate generator. It generates baud-rate clock out of 100 MHz clock (ATA100 clock). This baud-rate can be used for UART with non-frequency accurate application such as debug or peripheral control.

This baud-rate generator uses a multiplier-divider that divides by 217 and multiplies by 32. It generates 14.74654377 MHz, which has 64-PPM error against 14.7456 MHz. In addition, it will be SSCG modulated when SSCGEN signal asserted. Because of this nature, it is not recommended to use this Baud-rate for communication through MODEM or external line. But it is suitable to use in the system to control peripherals like LCD panel or remote control equipment.

5.5. SSCG (Spread Spectrum Clock Generator)

5.5.1. SSCG UNIT

The SSCG UNIT in TX4939 modulates MSTCLK directly. This SSCG module does frequency modulation by pure digital circuit.

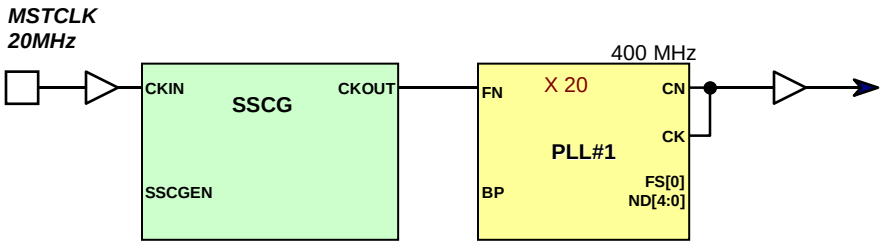


Figure 5-8 Location of SSCG UNIT

5.5.2. Modulation Profile

- *Modulation Magnitude = 0.5% (p-p)*
- *Modulation Frequency = 31.25 KHz*
- *Maximum Delay Time = 516 steps (19.09 nS)*

5.6. De-Skew Circuit

5.6.1. Theory of Operation

Figure 5-9 shows basic concepts of clock de-skew circuit. The module “dll_dskw1” is designed as follows.

A programmable delay line is implemented to delay the signal from CIN to OUT.
Whenever CIN input detect the clock edge, internal phase comparator starts to monitor both REF and SCK input for the clock edge. If SCK input detects the clock edge earlier than REF input, it decreases the delay value and if it is opposite, it increases the delay value. By this way, the pulse edge of P2 and P3 will be matched precisely.

For the delay line, Z7DELAYF ASIC delay line is used. It has 512 resolutions and each delay unit has 50 ps typical delay. In the full range of operational condition, the unit delay might vary from 30 ps to 90 ps. Therefore, the maximum adjustable range is about 15 ns.

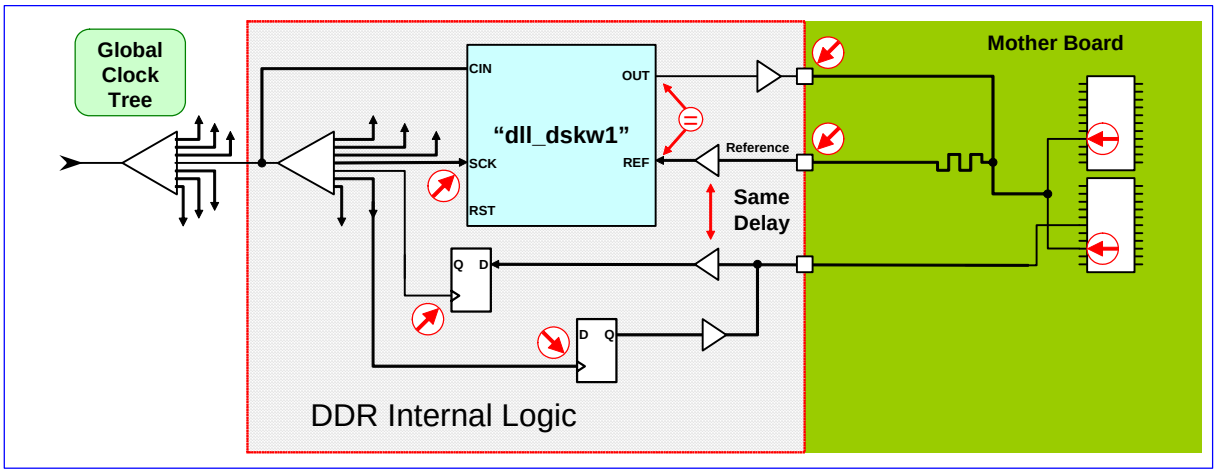


Figure 5-9 Concept of Clock De-Skew Circuit

5.6.2. DDR Clock De-Skew

The range of DDR Clock is from 100 MHz to 200 MHz. In terms of period, it is from 5 ns to 10 ns. Therefore, the “dll_dskw1” module can be used with straightforward manner. Figure 5-10 shows the DDR Clock De-Skew circuit for TX4939.

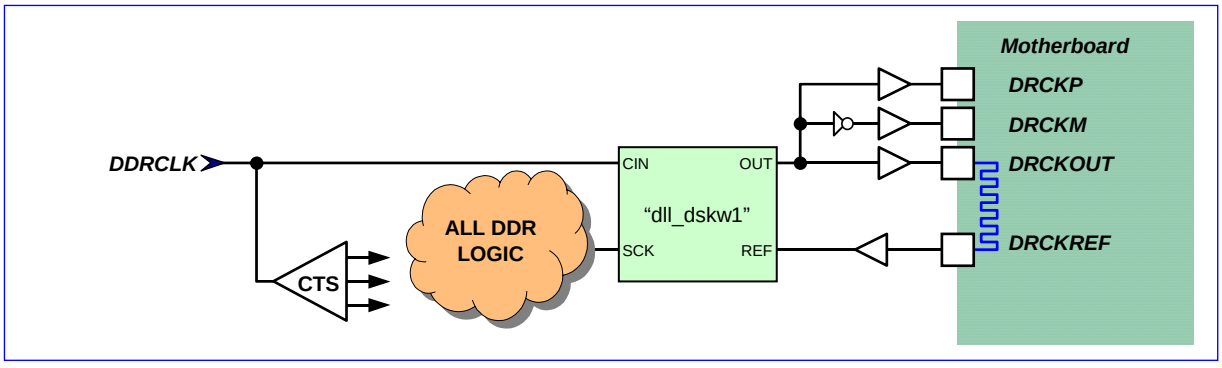


Figure 5-10 Structure of DDR Clock De-Skew Circuit

5.6.3. PCI Clock De-Skew

PCI De-Skew is performed by PLL circuit. Figure 5-11 shows this diagram and recommended PCI Clock distribution.

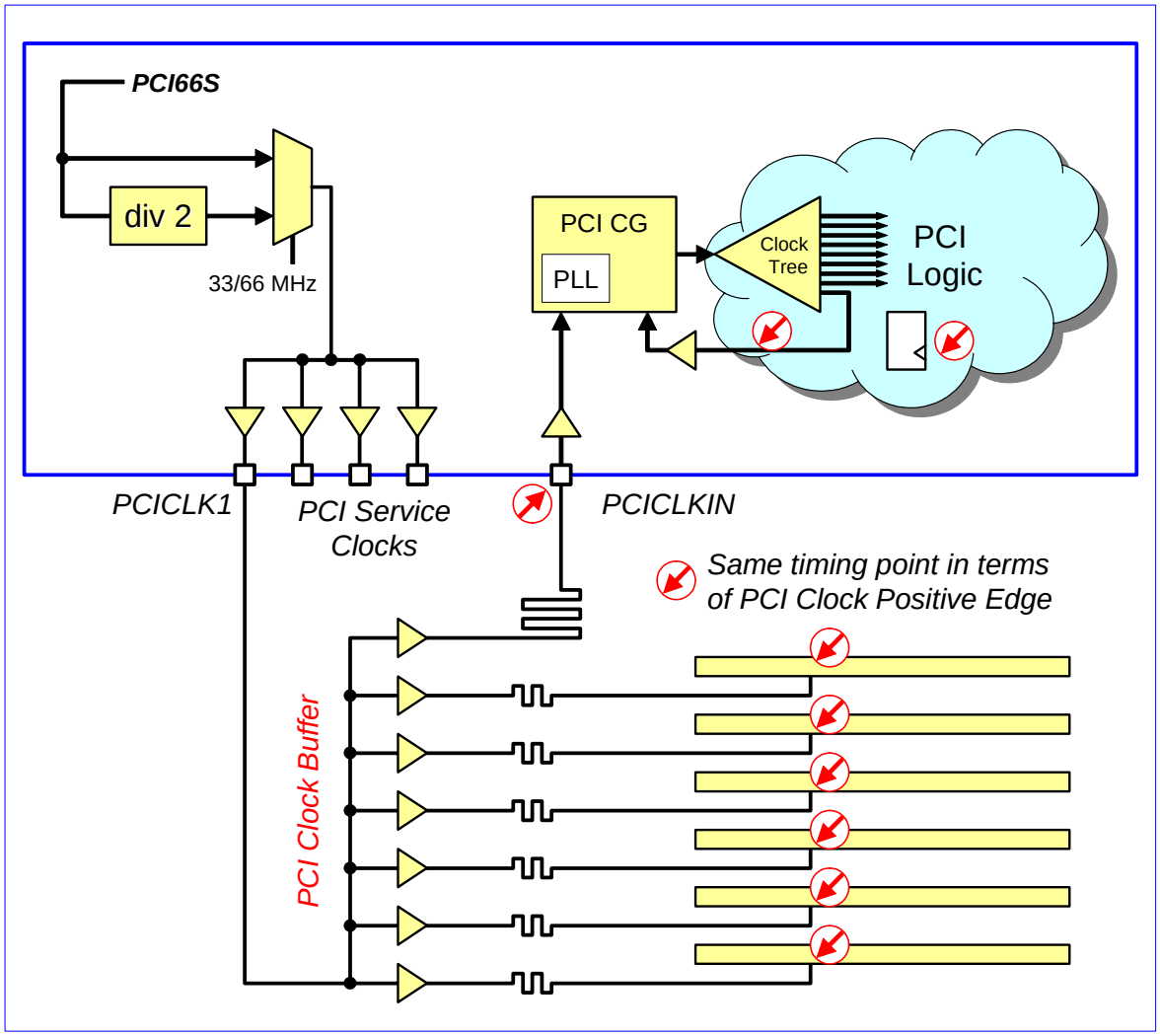


Figure 5-11 Structure of PCI Clock De-Skew Circuit

Chapter 6. Address Mapping

6.1. TX49 CPU Address Space

This chapter explains the address space of TX4939. Please refer to "64 bit TX System RISC TX49/H4 Core Architecture" about the details of mapping to a physical address from the virtual address of TX49/H4 core.

Here Figure 6-1 shows the default address mapping from logical addresses to physical addresses. There are two hardwired logical to physical translation tables exist. One is Boot Address related and the other is Internal Register related. Detailed description follows.

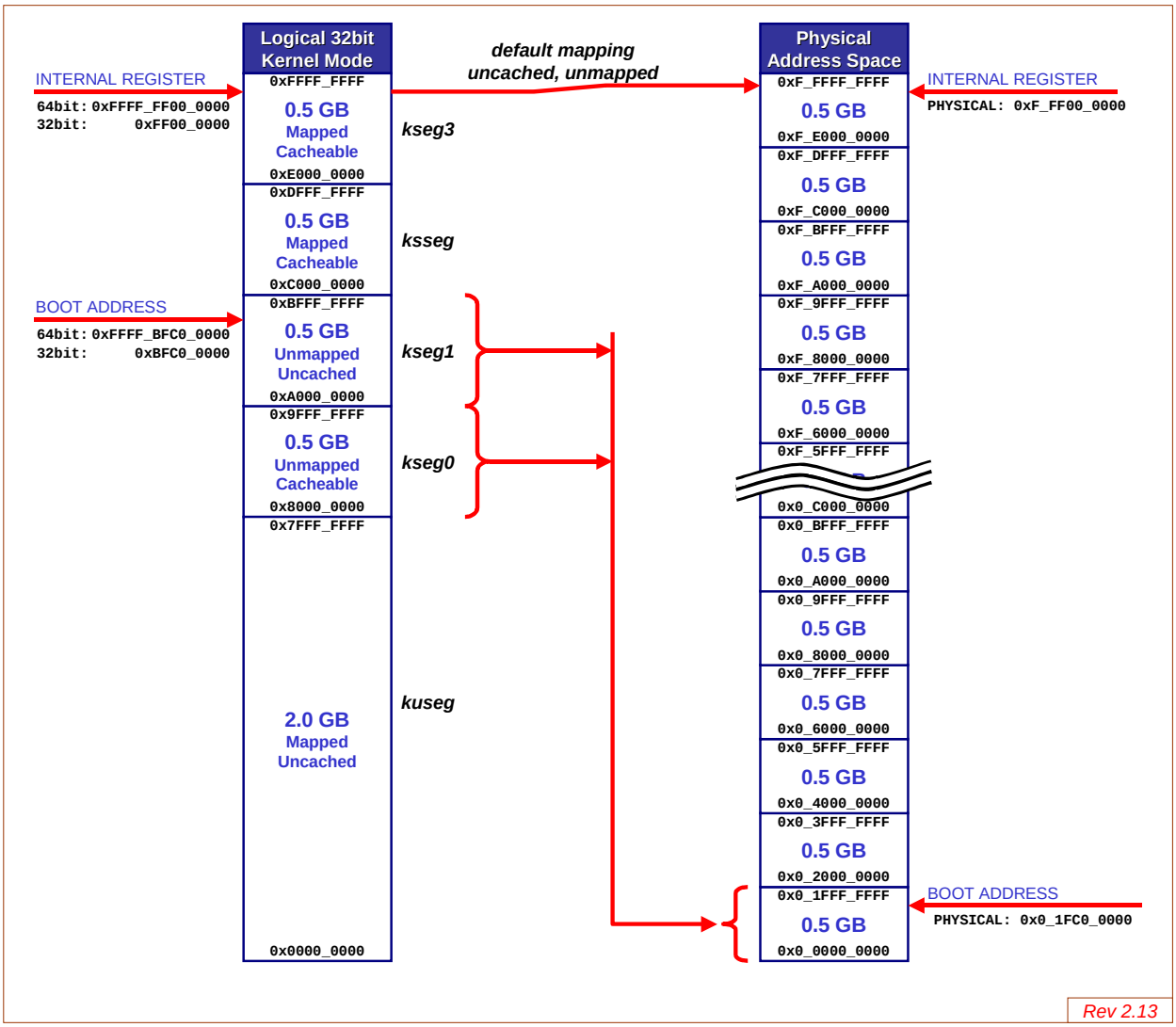


Figure 6-1 Logical to Physical Address Translation

6.2. Physical Address Map Overview

TX4939 supports up to 64G (2^{36}) bytes of physical address. Following resources are to be allocated in the physical address of the TX4939, such as (1) TX4939 Internal registers (refer to 6.5 Register Map), (2) DDR SDRAM, (3) External Devices such as ROM, I/O Devices, and (4) PCI Bus.

Each resource is to be allocated in any physical addresses by the register setup. Refer to the explanation of each controller for the details of the mapping. At initialization, only the internal registers and the memory space which stores the TX49/H4 core reset vectors are allocated shown as Figure 6-2. Usually ROM connected to the external bus controller channel 0 is used for the memory device that stores the reset vectors.

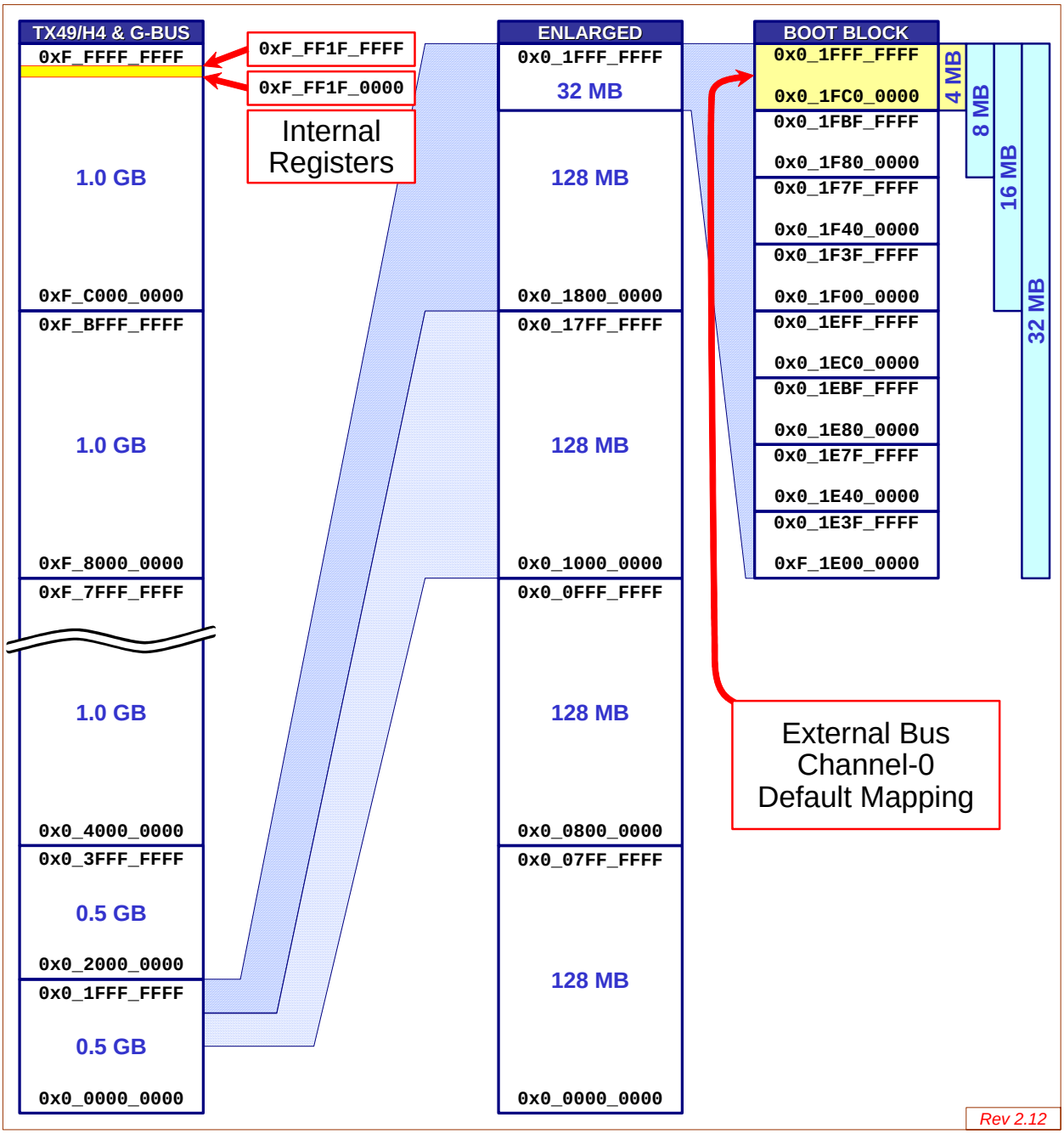


Figure 6-2 Physical Address Map at Initializing System

6.3. DDR SDRAM Mapping

6.3.1. Fundamentals

The DDR SDRAM Controller in TX4939 can support 1.0 GB memory capacity per channel. As this size exceeds the size of kseg0 and kseg1 (which size is 512 MB), there might be an issue to map this whole memory to the physical address 0x0_0000_0000.

The DDR SDRAM Mapping logic will provide system with the capability to split such DDR memory space exceed 512 MB to smaller region and map to any region of physical address space.

The granularity is intended to be 16 MB as target. However, it may increase if there are timing issues in the implementation. Also, to avoid design complexity, the maximum window size is 1.0 GB and window across the 1.0 GB boundary is prohibited.

6.3.2. Control Registers

Table 6-1 shows the registers for DDR Memory Mapping Control.

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-1 Internal Registers for DDR SDRAM Mapping Control

Offset Address		Register Size (bit)	Register Symbol	Register Name
DDR SDRAM Memory Mapping Control Register				
LD/SD LE/BE	**LW/SW LE/BE**			
0x8200	0x8200 0x8204	64	DRWINEN	Enable/Disable DDR Mapping Windows
0x8208	0x8208 0x820C	64	DRWIN00	DDR Mapping Window #0
0x8210	0x8210 0x8214	64	DRWIN01	DDR Mapping Window #1
0x8218	0x8218 0x821C	64	DRWIN02	DDR Mapping Window #2
0x8220	0x8220 0x8224	64	DRWIN03	DDR Mapping Window #3

6.3.3. DDR Mapping Window Control (DRWINEN) 0x8200

This register enables or disables each DDR Mapping Register. Before enabling any Mapping Window, i.e. Windows #0 to #3, corresponding registers, i.e. DRWIN00, DRWIN01, DRWIN02, and DRWIN03, has to be preset with precise value.

If failed, the system might cause serious malfunction and result is unknown.

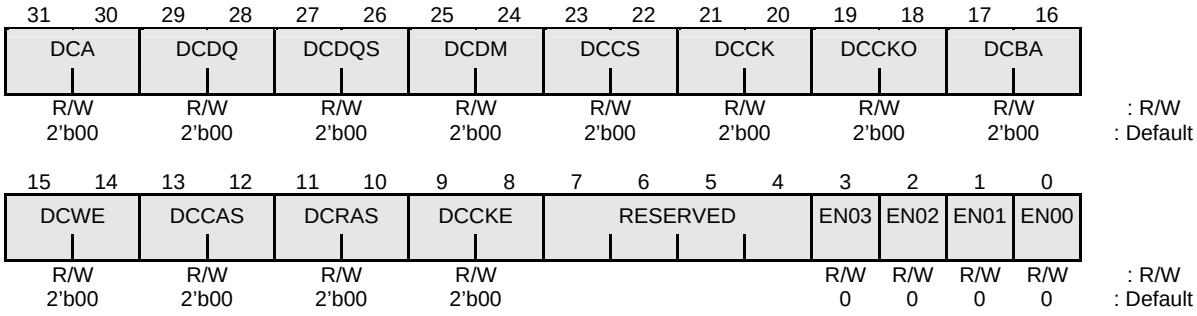


Figure 6-3 DDR Mapping Window Control

Table 6-2 DDR Mapping Window Control

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:32	—	Reserved	—	—	—
31:30	DCA	DC_DRA	Drive Strength Control of DRA[13:0] signal 2'b00 = 28% of maximum strength 2'b01 = 57% of maximum strength 2'b10 = 71% of maximum strength .. 2'b11 = maximum strength (100%)	2'b00	R/W
29:28	DCDQ	DSC_DRDQ	Drive Strength Control of DRDQ[31:0] Control is same as above	2'b00	R/W
27:26	DCDQS	DSC_DRDQS	Drive Strength Control of DRDQS[3:0] Control is same as above	2'b00	R/W
25:24	DCDM	DSC_DRDM	Drive Strength Control of DRDM[3:0] Control is same as above	2'b00	R/W
23:22	DCCS	DSC_DRCS	Drive Strength Control of DRCS[1:0] Control is same as above	2'b00	R/W
21:20	DCCK	DSC_DRCK	Drive Strength Control of DRCKP and DRCKP Control is same as above	2'b00	R/W
19:18	DCCKO	DSC_DRCKOUT	Drive Strength Control of DRCKOUT Control is same as above	2'b00	R/W
17:16	DCBA	DSC_DRBA	Drive Strength Control of DRBA[1:0] Control is same as above	2'b00	R/W
15:14	DCWE	DSC_DRWE	Drive Strength Control of DRWE Control is same as above	2'b00	R/W
13:12	DCCAS	DSC_DRCAS	Drive Strength Control of DRCAS Control is same as above	2'b00	R/W
11:10	DCRAS	DSC_DRRAS	Drive Strength Control of DRRAS Control is same as above	2'b00	R/W
9:8	DCCKE	DSC_DRCKE	Drive Strength Control of DRCKE Control is same as above	2'b00	R/W
3	EN03	ENWIN03	1'b1 = Enable Window #3, 1'b0 = Disable	0	R/W
2	EN02	ENWIN02	1'b1 = Enable Window #2, 1'b0 = Disable	0	R/W
1	EN01	ENWIN01	1'b1 = Enable Window #1, 1'b0 = Disable	0	R/W
0	EN00	ENWIN00	1'b1 = Enable Window #0, 1'b0 = Disable	0	R/W

6.3.4. DDR Mapping Window #n (n=0, 1, 2, 3) DRWIN00 0x8208

DRWIN01 0x8210
DRWIN02 0x8218
DRWIN03 0x8220

DRWIN00, DRWIN01, DRWIN02, and DRWIN03 control DDR Mapping condition independently.

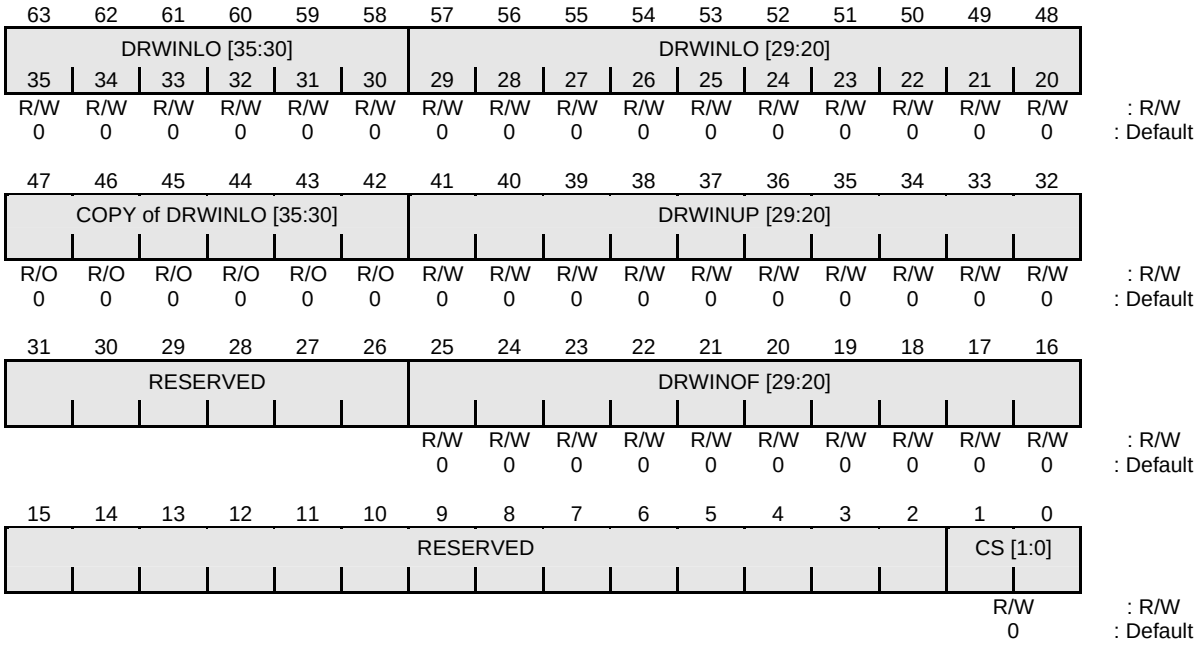


Figure 6-4 DDR Mapping Window Control

Table 6-3 DDR Mapping Window Control

Bit	Mnemonic	Field Name	Description	Initial Value	R/W

63:48	DRWINLO	DRWINLO [35:20]	Define the lower address of window #n in physical address.
			DRWINLO_ADRS = { DRWINUP [35:20], 20'HF_FFFF }
				0	R/W

47:42	DRWINUP	DRWINUP [35:30]	This register is the copy of DRWINLO [35:30]	0	R/O
41:32	DRWINUP	DRWINUP [29:20]	Define the upper address of window #n in physical address. DRWINUP _ADRS = { DRWINUP [35:20], 20'H0_0000 }	0	R/W
31:26	—	RESERVED	—	—	—
25:16	DRWINOF	DRWINOF [29:20]	Define the offset address of the target DDR memory space. DRWINOF _ADRS = { DRWINOF [29:20], 20'H0_0000 }	0	R/W
15:2	—	RESERVED	—	—	—
1:0	CS [1:0]	Chip Select [1:0]	Define corresponding DDR channel number with CS CS[1:0] = 0 Channel 0 (CS0) CS[1:0] = 1 Channel 0 (CS0) CS[1:0] = 2 Channel 1 (CS1) CS[1:0] = 3 Channel 1 (CS1) NOTE: CS setting should be consistent with DDR_CTRL15 setting.	0	R/W

When the system receives an effective physical address, (EPA), it checks following condition for enabled window(s).

IF ((EPA <= DRWINUP_ADRS) && (EPA >= DRWINLO_ADRS))

Then, if it is true, Corresponding DDR channel will be accessed with following DDR_EFFECTIVE_ADDRESS.

DDR_EFFECTIVE_ADDRESS = EPA – DRWINLO_ADRS + DRWINOF_ADRS

Figure 6-5 shows an example of DDR Memory Mapping. In this example, DDR Channel 0 (CS0) is split two blocks, one is 384 MB and the rest is 640 MB. The 384 MB portion is mapped to 0x0_0000_0000 to 0x0_17FF_FFFF. The rest is mapped to 0x0_8000_0000 to 0x0_A7FF_FFFF and creates continuous memory space with 0x0_4000_0000 to 0x0_7FFF_FFFF memory space, which is mapped from DDR Channel 1 (CS1).

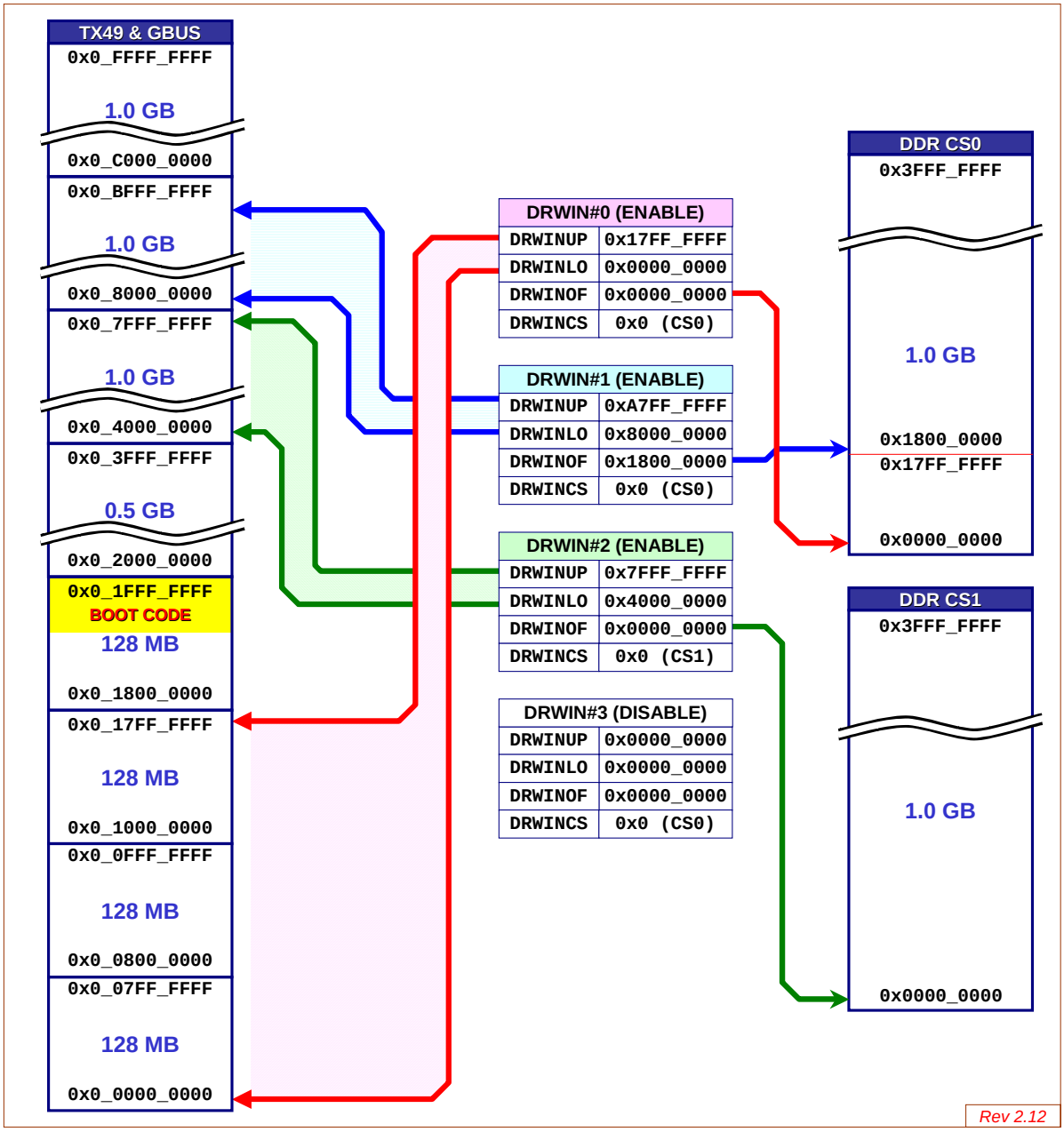


Figure 6-5 Example of DDR Memory Split-Mapping

6.4. PCI Address Space Mapping

This section describes new feature that is introduced TX4939 PCI Controller. It is related memory mapping from PCI Memory Space to GBUS Memory Space. This mapping will be used when any PCI Master Device (external devices) access any memory device attached GBUS (DDR Memory is an example) through PGB (PCI to GBUS Bridge).

TX4939 has three windows for this mapping. Three control registers are provided to define the mapping windows as same as TMPR4938XB, however each window size is now programmable. (Note: TMPR4938XB has three windows but each window has fixed size, such as 512 MB (Window#0), 16 MB (Window #1), and 1 MB (Window #2).

6.4.1. P2G Memory Space (n) PCI Lower Base Address Register (n=0,1,2)

These registers are corresponding to the PCI-GBUS Window #n. Those registers are P2GM0PLBASE, P2GM1PLBASE, P2GM2PLBASE,

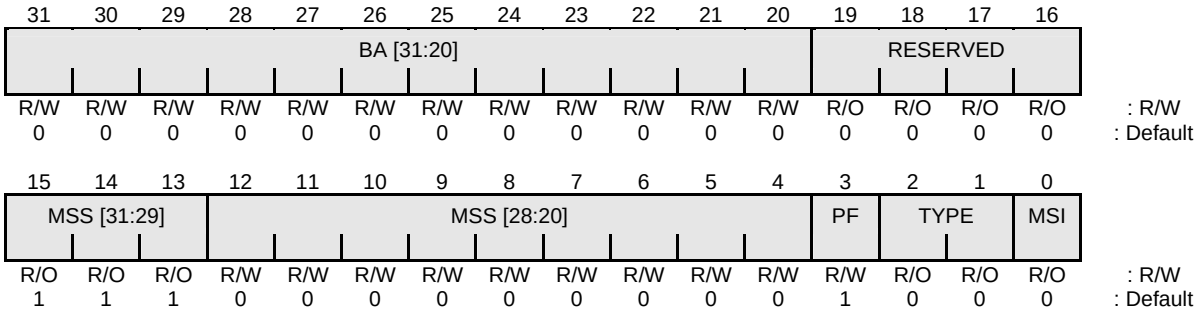


Figure 6-6 P2G Memory Space (n) PCI Lower Address Register

Table 6-4 P2G Memory Space (n) PCI Lower Address Register

Bit	Mnemonic	Field Name	Description	Default	R/W
31:20	BA [31:20]	Base Address	Base Address (Default=0x000) Sets the lower address of the PCI base address in Target Access Memory Space n. The size of Memory Space is defined by the value of MSS [23:20]. Effective Base Address (EBA) is given by following formula. EBA [31:20] = BA[31:20] & MSS [31:20] (bitwise AND)	0x000	R/W
19:16		RESERVED		4'b0000	R/O
15:13	MSS [31:29]	Size (Maximum)	Memory Space Size (Maximum, Default=3'b111, 512 MB).	3'b111	R/O
12:4	MSS [28:20]	Size	Memory Space Size (Default=0x000, 512 MB) See Table 6-5 for other corresponding.	0x000	R/W
3	PF	Prefetchable	Prefetchable 1: Indicates that memory is prefetchable. 0: Indicates that memory is not prefetchable.	1	R/W
2:1	TYPE	Type	Type (Default: 00) 00: Indicates that an address is within a 32-bit address region	0	R/O
0	MSI	Memory Space	Memory Space Indicator (Fixed Value: 0) 0: Indicates that this Base Address Register is for use by the PCI Memory Space	0	R/O

Table 6-5 Corresponding of Memory Space Size and MSS[31:20] Value

MSS [31:29] (Hexadecimal)	MSS [31:29] Bitwise												Memory Size
	31	30	29	28	27	26	25	24	23	22	21	20	
0xE00	1	1	1	0	0	0	0	0	0	0	0	0	512 MB
0xF00	1	1	1	1	0	0	0	0	0	0	0	0	256 MB
0xF80	1	1	1	1	1	0	0	0	0	0	0	0	128 MB
0xFC0	1	1	1	1	1	1	0	0	0	0	0	0	64 MB
0xFE0	1	1	1	1	1	1	1	0	0	0	0	0	32 MB
0xFF0	1	1	1	1	1	1	1	1	0	0	0	0	16 MB
0xFF8	1	1	1	1	1	1	1	1	1	0	0	0	8 MB
0xFFC	1	1	1	1	1	1	1	1	1	1	0	0	4 MB
0xFFE	1	1	1	1	1	1	1	1	1	1	1	0	2 MB
0xFFF	1	1	1	1	1	1	1	1	1	1	1	1	1 MB

6.5. Register Map Convention

6.5.1. Addressing

TX4939 internal registers are to be accessed through 64 K bytes address space that is based on physical address 0xF_FF1F_0000 or pointed address by RAMP register. Figure 6-7 shows how to generate internal register address. Physical address 1 and physical address 2 shown Figure 6-7 access the same register.

In TX49/H4 Core, the physical address from 0xF_FF00_0000 to 0xF_FF3F_FFFF are uncached mapped to the virtual address from 0xFF00_0000 to 0xFF3F_FFFF (32 bit mode) /from 0xFFFF_FFFF_FF00_0000 to 0xFFFF_FFFF_FF3F_FFFF (64 bit mode).

This space includes the region from 0xF_FF1F_0000 allocated TX4939 internal registers at initialization.

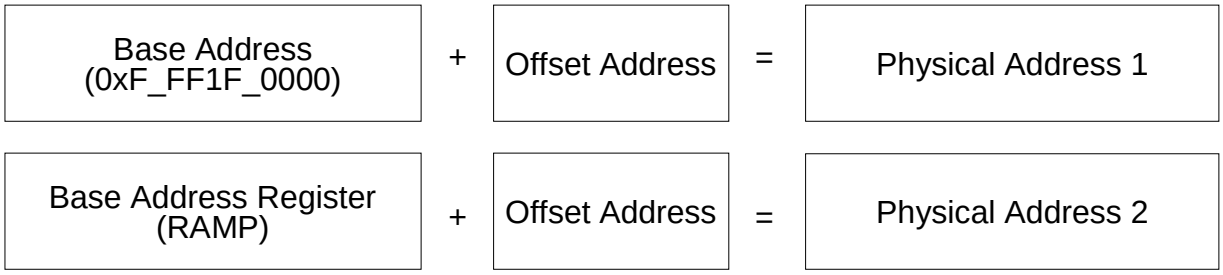


Figure 6-7 Generating Physical Address for a Internal Register

6.5.2. Endianness and Register size

As this system has been designed based on the philosophy of Bi-Endian, all register address is defined as such that any register less than 32 bits are accessed with 32 bit and 64 bit registers with 64 bit. However, it does not limit the 64 bit register to be accessed with 32 bit.

As far as all 32 bit register and less than 32 bit register should be accessed only with 32 bit operation. Other type of access may cause undefined result.

On the other hand, 64 bit registers can be accessed also with 32 bit operation. But the access address might change depending on the system Endianness. Table 6-6 and Table 6-7 describes the access address of 32 bit portion in the 64 bit register. Please note that access way is dependent on register.

Table 6-6 32-bit Size Access to 64-bit Register (SRAM, EBUSC, PCIC1, PCIC, DMAC0/1)

Address	Big Endian (Bit which are accessed)	Little Endian (Bit which are accessed)
0x*_****_***0	[63……..32] [31……..0]	[63……..32] [31……..0]
0x*_****_***8	########	########
0x*_****_***4	[63……..32] [31……..0]	[63……..32] [31……..0]
0x*_****_***C	########	########

(######## indicates 32 bits data (upper 32 bits or lower 32 bits) which are accessed.)

Table 6-7 32-bit Size Access to 64-bit Register (CRYPT, DDR, Config)

Address	Big/Little Endian (Bit which are accessed)
0x*_****_***0	[63……..32] [31……..0]
0x*_****_***8	########
0x*_****_***4	[63……..32] [31……..0]
0x*_****_***C	########

(######## indicates 32 bits data (upper 32 bits or lower 32 bits) which are accessed.)

6.6. Register Map

Please refer to “16.5 PCI Configuration Space Register” about PCI configuration register.

Table 6-8 Register Map

Offset Address	Peripheral Controller
0x0000 to 0x2FFF	Reserved
0x3000 to 0x3FFF	ATA0
0x4000 to 0x4FFF	ATA1
0x5000 to 0x5FFF	NDFMC
0x6000 to 0x67FF	SRAMC
0x6800 to 0x6FFF	Crypt Engine
0x7000 to 0x7FFF	PCIC1 (For Ethernet MAC[1:0])
0x8000 to 0x8FFF	DDR (including DDR mapping register
0x9000 to 0x9FFF	EBUSC
0xA000 to 0xAFFF	VPC
0xB000 to 0xB7FF	DMAC0
0xB800 to 0xBFFF	DMAC1
0xC000 to 0xCFFF	Reserved
0xD000 to 0xDFFF	PCIC
0xE000 to 0xE7FF	CONFIG/GPIO
0xE800 to 0xEFFF	IRC
0xF000 to 0xF0FF	TMR0
0xF100 to 0xF1FF	TMR1
0xF200 to 0xF2FF	TMR2
0xF300 to 0xF3FF	SIO0, SIO2
0xF400 to 0xF4FF	SIO1, SIO3
0xF500 to 0xF50F	Reserved
0xF510 to 0xF6FF	Reserved
0xF700 to 0xF7FF	ACLC
0xF800 to 0xF8FF	SPIC
0xF900 to 0xF9FF	I2C
0xFA00 to 0xFAFF	I2S
0xFB00 to 0xFBFF	RTC
0xFC00 to 0xFCFF	CIR
0xFD00 to 0xFDFF	TMR3
0xFE00 to 0xFEFF	TMR4
0xFF00 to 0xFFFF	TMR5

6.6.1. Registers for ATA0

All registers below are mapped into ATA100 CORE registers one by one, except the DMA Command register is mapped to ATA100 CORE register, Start register and Direction Control register. Use byte, half-word, or word load/store to access register with 8, 16, or 32 bits word, respectively

Table 6-9 Internal Registers for ATA0

Offset Address		Register Size (bit)	Register Symbol	Register Name
ATA0				
LE	**BE**			
0x3000	0x3006	16	ATA0_DATA	DATA register ATA0 device
0x3001	0x3006	8	ATA0_Error_Ft	Error/Feature ATA0 device
0x3002	0x3005	8	ATA0_Sec	Sector Count ATA0 device
0x3003	0x3004	8	ATA0_LBA0	LBA Low ATA0 device
0x3004	0x3003	8	ATA0_LBA1	LBA Mid ATA0 device
0x3005	0x3002	8	ATA0_LBA2	LBA High ATA0 device
0x3006	0x3001	8	ATA0_Device	Device register ATA0 device
0x3007	0x3000	8	ATA0_St_Cmd	Status/Command ATA0 device
0x3402	0x3405	8	ATA0_Alt_DevCtl	Alternate Status/Device Control ATA0 device
0x3800	0x3807	8	ATA0_DMA_Cmd	DMA command for ATA0
0x3802	0x3805	8	ATA0_DMA_stat	DMA Status for ATA0
0x3804	0x3800	32	ATA0_PRD_Ptr	PRD Table Pointer for ATA0
0x3c00	0x3c06	16	ATA0_Sys_Ctl_1	System Control 1 for ATA0
0x3c02	0x3c04	16	ATA0_Sys_Ctl_2	System Control 2 for ATA0
0x3c08	0x3c0e	16	ATA0_Xfer_Cnt_1	Transfer Word Count 1 for ATA0
0x3c0a	0x3c0c	16	ATA0_Xfer_Cnt_2	Transfer Word Count 2 for ATA0
0x3c10	0x3c16	16	ATA0_Sec_Cnt	Sector Count for ATA0
0x3c18	0x3c1e	16	ATA0_Strt_AddL	Transfer Start Low Address for ATA0
0x3c20	0x3c26	16	ATA0_Strt_AddU	Transfer Start Upper Address for ATA0
0x3c28	0x3c2e	16	ATA0_Add_Ctl	Additional Control for ATA0
0x3c30	0x3c36	16	ATA0_Lo_BCnt	Lower Burst Count for ATA0
0x3c38	0x3c3e	16	ATA0_Up_BCnt	Upper Burst Count for ATA0
0x3c88	0x3c8e	16	ATA0_PIO_Acc	PIO Access Address for ATA0
0x3c90	0x3c96	16	ATA0_H_Rst_Tim	Host Reset Timer for ATA0
0x3c98	0x3c9e	16	ATA0_int_ctl	Interrupt Control for ATA0
0x3cb8	0x3cbe	16	ATA0_Pkt_Cmd	ATAPI Packet Command for ATA0
0x3cc0	0x3cc6	16	ATA0_Bxfer_cntH	Bus Transfer Count High for ATA0
0x3cc8	0x3cce	16	ATA0_Bxfer_cntL	Bus Transfer Count Low for ATA0
0x3cd0	0x3cd6	16	ATA0_Dev_TErr	Device Timing Error for ATA0
0x3cd8	0x3cde	16	ATA0_Pkt_xfer_ct	Packet Transfer Control for ATA0
0x3ce0	0x3ce6	16	ATA0_Strt_AddT	Transfer Start Top Address for ATA0

6.6.2. Registers for ATA1

All registers below are mapped into ATA100 CORE registers one by one, except the DMA Command register is mapped to ATA100 CORE register, Start register and Direction Control register. Use byte, half-word, or word load/store to access register with 8, 16, or 32 bits word, respectively

Table 6-10 Internal Registers for ATA1

Offset Address		Register Size (bit)	Register Symbol	Register Name
ATA1				
LE	**BE**			
0x4000	0x4006	16	ATA1_DATA	DATA register ATA1 device
0x4001	0x4006	8	ATA1_Error_Ft	Error/Feature ATA1 device
0x4002	0x4005	8	ATA1_Sec	Sector Count ATA1 device
0x4003	0x4004	8	ATA1_LBA0	LBA Low ATA1 device
0x4004	0x4003	8	ATA1_LBA1	LBA Mid ATA1 device
0x4005	0x4002	8	ATA1_LBA2	LBA High ATA1 device
0x4006	0x4001	8	ATA1_Device	Device register ATA1 device
0x4007	0x4000	8	ATA1_St_Cmd	Status/Command ATA1 device
0x4402	0x4405	8	ATA1_Alt_DevCtl	Alternate Status/Device Control ATA1 device
0x4800	0x4807	8	ATA1_DMA_Cmd	DMA command for ATA1
0x4802	0x4805	8	ATA1_DMA_stat	DMA Status for ATA1
0x4804	0x4800	32	ATA1_PRD_Ptr	PRD Table Pointer for ATA1
0x4c00	0x4c06	16	ATA1_Sys_Ctl_1	System Control 1 for ATA1
0x4c02	0x4c04	16	ATA1_Sys_Ctl_2	System Control 2 for ATA1
0x4c08	0x4c0e	16	ATA1_Xfer_Cnt_1	Transfer Word Count 1 for ATA1
0x4c0a	0x4c0c	16	ATA1_Xfer_Cnt_2	Transfer Word Count 2 for ATA1
0x4c10	0x4c16	16	ATA1_Sec_Cnt	Sector Count for ATA1
0x4c18	0x4c1e	16	ATA1_Strt_AddL	Transfer Start Low Address for ATA1
0x4c20	0x4c26	16	ATA1_Strt_AddU	Transfer Start Upper Address for ATA1
0x4c28	0x4c2e	16	ATA1_Add_Ctl	Additional Control for ATA1
0x4c30	0x4c36	16	ATA1_Lo_BCnt	Lower Burst Count for ATA1
0x4c38	0x4c3e	16	ATA1_Up_BCnt	Upper Burst Count for ATA1
0x4c88	0x4c8e	16	ATA1_PIO_Acc	PIO Access Address for ATA1
0x4c90	0x4c96	16	ATA1_H_Rst_Tim	Host Reset Timer for ATA1
0x4c98	0x4c9e	16	ATA1_int_ctl	Interrupt Control for ATA1
0x4cb8	0x4cbe	16	ATA1_Pkt_Cmd	ATAPI Packet Command for ATA1
0x4cc0	0x4cc6	16	ATA1_Bxfer_cntH	Bus Transfer Count High for ATA1
0x4cc8	0x4cce	16	ATA1_Bxfer_cntL	Bus Transfer Count Low for ATA1
0x4cd0	0x4cd6	16	ATA1_Dev_TErr	Device Timing Error for ATA1
0x4cd8	0x4cde	16	ATA1_Pkt_xfer_ct	Packet Transfer Control for ATA1
0x4ce0	0x4ce6	16	ATA1_Strt_AddT	Transfer Start Top Address for ATA1

6.6.3. Registers for NAND Controller (NDFMC)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-11 Internal Registers for NAND Controller (NDFMC)

Offset Address		Register Size (bit)	Register Symbol	Register Name
NAND Controller (NDFMC)				
LD/SD	**LW/SW**			
LE/BE	**LE/BE**			
0x5000	0x5000	32	NDFDTR	NAND Flash Memory Data Transfer Register
0x5008	0x5008	32	NDFMCR	NAND Flash Memory Mode Control Register
0x5010	0x5010	32	NDFSR	NAND Flash Memory Status Register
0x5018	0x5018	32	NDFISR	NAND Flash Memory Interrupt Status Register
0x5020	0x5020	32	NDFIMR	NAND Flash Memory Interrupt Mask Register
0x5028	0x5028	32	NDFSPR	NAND Flash Memory Strobe Pulse Width Register

6.6.4. Registers for SRAM Controller (SRAMC)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-12 Internal Registers for SRAM Controller (SRAMC)

Offset Address			Register Size (bit)	Register Symbol	Register Name
SRAM Controller (SRAMC)					
LD/SD	**LW/SW**				
LE/BE	**LE**	**BE**			
0x6000	0x6000 0x6004	0x6004 0x6000	64	SRAMCR	Internal SRAM Control Register

6.6.5. Registers for Crypt Engine Controller

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-13 Internal Registers for Crypto Controller

Offset Address		Register Size (bit)	Register Symbol	Register Name
Crypto Controller				
LD/SD	**LW/SW**			
LE/BE	**LE/BE**			
0x6800	0x6800 0x6804	64	CSR	Control and Status Register
0x6808	0x6808 0x680C	64	IDESPtr	Initial Descriptor Pointer Register
0x6810	0x6810 0x6814	64	CDESPtr	Current Cipher Descriptor Pointer Register
0x6818	0x6818 0x681C	64	BusErr	Bus Error Address Register
0x6820	0x6820 0x6824	64	Cip_tout	Time Out Register
0x6828	0x6828 0x682C	64	Cir	Context Index Register
0x6830	0x6830 0x6834	64	Cdr1	Context Data Register 1
0x6838	0x6838 0x683C	64	Cdr2	Context Data Register 2
0x6840	0x6840 0x6844	64	Cdr3	Context Data Register 3
0x6848	0x6848 0x684C	64	Cdr4	Context Data Register 4
0x6850	0x6850 0x6854	64	Cdr5	Context Data Register 5
0x6858	0x6858 0x685C	64	Cdr6	Context Data Register 6
0x6860	0x6860 0x6864	64	Cdr7	Context Data Register 7
0x6868	0x6868 0x686C	64	Cdr8	Context Data Register 8
0x6870	0x6870 0x6874	64	Cdr9	Context Data Register 9
0x6878	0x6878 0x687C	--	--	Reserved
0x6880	0x6880 0x6884	--	--	Reserved
0x6888	0x6888 0x688C	--	--	Reserved
0x6890	0x6890 0x6894	--	--	Reserved
0x6898	0x6898 0x689C	--	--	Reserved
0x68A0	0x68A0 0x68A4	--	--	Reserved

0x68A8	0x68A8 0x68AC	--	--	Reserved
0x68B0	0x68B0 0x68B4	64	RCSR	RNG Control and Status Register
0x68B8	0x68B8 0x68BC	32	RPR	RNG Parameter Register
0x68C0	0x68C0 0x68C4	64	RDR	RNG Debug Register
0x68C8	0x68C8 0x68CC	64	ROR1	RNG Output Register 1
0x68D0	0x68D0 0x68D4	64	ROR2	RNG Output Register 2
0x68D8	0x68D8 0x66DC	64	ROR3	RNG Output Register 3
0x68E0-0x68E8		--	--	Reserved
0x68F0	0x68F0	32	Xorslr0	XOR Source Lower Register
0x68F8	0x68F8	32	XorSUR1	XOR Source Upper Register

6.6.6. Registers for PCI Controller for ETHERC (PCIC1)

For 64-bit register, use LD/SD and LW/SW instructions to access register. For 32-bit register, use LW/SW instructions to access register.

Table 6-14 Internal Registers for PCI Controller for ETHERC (PCIC1)

Offset Address		Register Size (bit)	Register Symbol	Register Name
PCI Controller for ETHERC (PCIC1)				
LD/SD LE/BE	**LW/SW LE/BE**			
	0x7000	32	PCIID	ID Register (Device ID, Vender ID)
	0x7004	32	PCISTATUS	PCI Status Command Register (Status, Command)
	0x7008	32	PCICCREV	Class Code Revision ID Register (Class Code, Revision ID)
	0x700C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
	0x7010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
	0x7014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
	0x7018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
	0x701C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
	0x7020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
	0x7024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
	0x702C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vender ID)
	0x7034	32	PCICAPPTR	Capability Pointer Register (Capability Register)
	0x703C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
	0x7040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
	0x7080	32	G2PSTATUS	G2P Status Register
	0x7084	32	G2PMASK	G2P Interrupt Mask Register
	0x7088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
	0x708C	32	PCIMASK	PCI Status Interrupt Mask Register
	0x7090	32	P2GCFG	P2G Configuration Register
	0x7094	32	P2GSTATUS	P2G Status Register
	0x7098	32	P2GMASK	P2G Interrupt Mask Register
	0x709C	32	P2GCCMD	P2G Current Command Register
	0x7100	32	PBAREQPORT	PCI Bus Arbiter Request port Register
	0x7104	32	PBACFG	PCI Bus Arbiter Configuration Register
	0x7108	32	PBASTATUS	PCI Bus Arbiter Status Register
	0x710C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
	0x7110	32	PBABM	PCI Bus Arbiter Broken Master Register
	0x7114	32	PBACREQ	PCI Bus Arbiter Current Request Register
	0x7118	32	PBACGNT	PCI Bus Arbiter Current Grant Register
	0x711C	32	PBACSTATE	PCI Bus Arbiter Current State Register

0x7120	(LE) 0x7120 0x7124	(BE) 0x7124 0x7120	64	G2PM0GBASE	G2P Memory Space 0 G-bus Base Address Register
0x7128	(LE) 0x7128 0x712C	(BE) 0x712C 0x7128	64	G2PM1GBASE	G2P Memory Space 1 G-bus Base Address Register
0x7130	(LE) 0x7130 0x7134	(BE) 0x7134 0x7130	64	G2PM2GBASE	G2P Memory Space 2 G-bus Base Address Register
0x7138	(LE) 0x7138 0x713C	(BE) 0x713C 0x7138	64	G2PIOGBASE	G2P I/O Space G-bus Base Address Register
	0x7140		32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
	0x7144		32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
	0x7148		32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
	0x714C		32	G2PIOMASK	G2P I/O Space Address Mask Register
0x7150	(LE) 0x7150 0x7154	(BE) 0x7154 0x7150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
0x7158	(LE) 0x7158 0x715C	(BE) 0x715C 0x7158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
0x7160	(LE) 0x7160 0x7164	(BE) 0x7164 0x7160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
0x7168	(LE) 0x7168 0x716C	(BE) 0x716C 0x7168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
	0x7170		32	PCICCFG	PCI Controller Configuration Register
	0x7174		32	PCICSTATUS	PCI Controller Status Register
	0x7178		32	PCICMASK	PCI Controller Interrupt Mask Register
0x7180	(LE) 0x7180 0x7184	(BE) 0x7184 0x7180	64	P2GM0GBASE	P2G Memory Space 0 G-bus Base Address Register
0x7188	(LE) 0x7188 0x718C	(BE) 0x718C 0x7188	64	P2GM1GBASE	P2G Memory Space 1 G-bus Base Address Register
0x7190	(LE) 0x7190 0x7194	(BE) 0x7194 0x7190	64	P2GM2GBASE	P2G Memory Space 2 G-bus Base Address Register
0x7198	(LE) 0x7198 0x719C	(BE) 0x719C 0x7198	64	P2GIOGBASE	P2G I/O Space G-bus Base Address Register
	0x71A0		32	G2PCFGADRS	G2P Configuration Address Register
	0x71A4		32	G2PCFGDATA	G2P Interrupt Acknowledge Data Register
	0x71C8		32	G2PINTACK	G2P Interrupt Acknowledge Data Register
	0x71CC		32	G2PSPC	G2P Special Cycle Data Register

6.6.7. Registers for DDR SDRAM Controller (DDRC)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-15 Internal Registers for DDR SDRAM Controller (DDRC)

Offset Address		Register Size (bit)	Register Symbol	Register Name
DDR SDRAM Controller (DDRC)				
LD/SD	**LW/SW**			
LE/BE	**LE/BE**			
0x8000	0x8000	16	DDRCTL00	DDR_CTL_00
0x8008	0x8008	16	DDRCTL01	DDR_CTL_01
0x8010	0x8010	16	DDRCTL02	DDR_CTL_02
0x8018	0x8018	16	DDRCTL03	DDR_CTL_03
0x8020	0x8020	16	DDRCTL04	DDR_CTL_04
0x8028	0x8028	16	DDRCTL05	DDR_CTL_05
0x8030	0x8030	16	DDRCTL06	DDR_CTL_06
0x8038	0x8038	16	DDRCTL07	DDR_CTL_07
0x8040	0x8040	16	DDRCTL08	DDR_CTL_08
0x8048	0x8048	16	DDRCTL09	DDR_CTL_09
0x8050	0x8050	16	DDRCTL10	DDR_CTL_10
0x8058	0x8058	16	DDRCTL11	DDR_CTL_11
0x8060	0x8060	16	DDRCTL12	DDR_CTL_12
0x8068	0x8068	16	DDRCTL13	DDR_CTL_13
0x8070	0x8070	16	DDRCTL14	DDR_CTL_14
0x8078	0x8078	16	DDRCTL15	DDR_CTL_15
0x8080	0x8080	16	DDRCTL16	DDR_CTL_16
0x8088	0x8088	16	DDRCTL17	DDR_CTL_17
0x8090	0x8090	16	DDRCTL18	DDR_CTL_18
0x8098	0x8098	16	DDRCTL19	DDR_CTL_19
0x80A0	0x80A0	16	DDRCTL20	DDR_CTL_20
0x80A8	0x80A8	16	DDRCTL21	DDR_CTL_21
0x80B0	0x80B0	16	DDRCTL22	DDR_CTL_22
0x80B8	0x80B8	16	DDRCTL23	DDR_CTL_23
0x80C0	0x80C0	16	DDRCTL24	DDR_CTL_24
0x80C8	0x80C8	16	DDRCTL25	DDR_CTL_25
0x80D0	0x80D0	16	DDRCTL26	DDR_CTL_26
0x80D8	0x80D8	16	DDRCTL27	DDR_CTL_27
0x80E0	0x80E0	16	DDRCTL28	DDR_CTL_28
0x80E8	0x80E8	16	DDRCTL29	DDR_CTL_29
0x80F0	0x80F0	16	DDRCTL30	DDR_CTL_30
0x80F8	0x80F8	16	DDRCTL31	DDR_CTL_31
0x8100	0x8100	16	DDRCTL32	DDR_CTL_32
0x8108	0x8108	16	DDRCTL33	DDR_CTL_33
0x8110	0x8110	16	DDRCTL34	DDR_CTL_34
0x8118	0x8118	16	DDRCTL35	DDR_CTL_35
0x8120	0x8120	16	DDRCTL36	DDR_CTL_36
0x8128	0x8128	16	DDRCTL37	DDR_CTL_37
0x8130	0x8130	16	DDRCTL38	DDR_CTL_38
0x8138	0x8138	16	DDRCTL39	DDR_CTL_39
0x8140	0x8140	16	DDRCTL40	DDR_CTL_40
0x8148	0x8148	16	DDRCTL41	DDR_CTL_41
0x8150	0x8150	16	DDRCTL42	DDR_CTL_42
0x8158	0x8158	16	DDRCTL43	DDR_CTL_43
0x8160	0x8160	16	DDRCTL44	DDR_CTL_44
0x8168	0x8168	16	DDRCTL45	DDR_CTL_45
0x8170	0x8170	16	DDRCTL46	DDR_CTL_46
0x8200	0x8200 0x8204	64	DRWINEN	Enable/Disable DDR Mapping Windows
0x8208	0x8208 0x820C	64	DRWIN00	DDR Mapping Window #0
0x8210	0x8210 0x8214	64	DRWIN01	DDR Mapping Window #1
0x8218	0x8218 0X821C	64	DRWIN02	DDR Mapping Window #2
0x8220	0x8220 0X8224	64	DRWIN03	DDR Mapping Window #3

6.6.8. Registers for External Bus Controller (EBUSC)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-16 Internal Registers for External Bus Controller (EBUSC)

Offset Address			Register Size (bit)	Register Symbol	Register Name
External Bus Controller (EBUSC)					
LD/SD	**LW/SW**				
LE/BE	**LE**	**BE**			
0x9000	0x9000 0X9004	0x9004 0X9000	64	EBCCR0	EBUS Channel Control Register 0
0x9008	0x9008 0X900C	0x900C 0X9008	64	EBCCR1	EBUS Channel Control Register 1
0x9010	0x9010 0X9014	0x9014 0X9010	64	EBCCR2	EBUS Channel Control Register 2
0x9018	0x9018 0X901C	0x901C 0X9018	64	EBCCR3	EBUS Channel Control Register 3
0x9020	0x9020 0X9024	0x9024 0X9020	64	EBCCR4	EBUS Channel Control Register 4 (Reserved)
0x9028	0x9028 0X902C	0x902C 0X9028	64	EBCCR5	EBUS Channel Control Register 5 (Reserved)
0x9030	0x9030 0X9034	0x9034 0X9030	64	EBCCR6	EBUS Channel Control Register 6 (Reserved)
0x9038	0x9038 0X903C	0x903C 0X9038	64	EBCCR7	EBUS Channel Control Register 7 (Reserved)

6.6.9. Registers for Video Port Controller (VPC)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-17 Internal Registers for Video Port Controller (VPC)

Offset Address		Register Size (bit)	Register Symbol	Register Name
VIdeo Port Controller (VPC)				
LD/SD	**LW/SW**			
LE/BE	**LE/BE**			
0xA000	0xA000	32	CSR	Control and Status Register
0xA008	0xA008	32	CtrlA1	Video Port 1 Control A Register
0xA010	0xA010	32	CtrlB1	Video Port 1 Control B Register
0xA018	0xA018	32	IDESPtr1	Video Port 1 Initial Descriptor Pointer Register
0xA020	0xA020	32	CDESPtr1	Video Port 1 Current Descriptor Pointer Register
0xA028	0xA028	32	CtrlA2	Video Port 2 Control A Register
0xA030	0xA030	32	CtrlB2	Video Port 2 Control B Register
0xA038	0xA038	32	IDESPtr2	Video Port 2 Initial Descriptor Pointer Register
0xA040	0xA040	32	CDESPtr2	Video Port 2 Current Descriptor Pointer Register
0xA048	0xA048	32	CtrlA3	Video Port 3 Control A Register
0xA050	0xA050	32	CtrlB3	Video Port 3 Control B Register
0xA058	0xA058	32	IDESPtr3	Video Port 3 Initial Descriptor Pointer Register
0xA060	0xA060	32	CDESPtr3	Video Port 3 Current Descriptor Pointer Register
0xA068	0xA068	32	BusErr	Bus Error Address Register

6.6.10. Registers for DMA Controller (DMAC0)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-18 Internal Registers for DMA Controller (DMAC0)

Offset Address			Register Size (bit)	Register Symbol	Register Name
DMA Controller (DMAC0)					
LD/SD	**LW/SW**				
LE/BE	**LE**	**BE**			
0xB000	0xB000 0xB004	0xB004 0xB000	64	DM0CHAR0	DMAC0 Chain Address Register 0
0xB008	0xB008 0xB00C	0xB00C 0xB008	64	DM0SAR0	DMAC0 Source Address Register 0
0xB010	0xB010 0xB014	0xB014 0xB010	64	DM0DAR0	DMAC0 Destination Address Register 0
0xB018	0xB018 0xB01C	0xB01C 0xB018	64	DM0CNTR0	DMAC0 Count Register 0
0xB020	0xB020 0xB024	0xB024 0xB020	64	DM0SAIR0	DMAC0 Source Address Increment Register 0
0xB028	0xB028 0xB02C	0xB02C 0xB028	64	DM0DAIR0	DMAC0 Destination Address Increment Register 0
0xB030	0xB030 0xB034	0xB034 0xB030	64	DM0CCR0	DMAC0 Channel Control Register 0
0xB038	0xB038 0xB03C	0xB03C 0xB038	64	DM0CSR0	DMAC0 Channel Status Register 0
0xB040	0xB040 0xB044	0xB044 0xB040	64	DM0CHAR1	DMAC0 Chain Address Register 1
0xB048	0xB048 0xB04C	0xB04C 0xB048	64	DM0SAR1	DMAC0 Source Address Register 1
0xB050	0xB050 0xB054	0xB054 0xB050	64	DM0DAR1	DMAC0 Destination Address Register 1
0xB058	0xB058 0xB05C	0xB05C 0xB058	64	DM0CNTR1	DMAC0 Count Register 1
0xB060	0xB060 0xB064	0xB064 0xB060	64	DM0SAIR1	DMAC0 Source Address Increment Register 1
0xB068	0xB068 0xB06C	0xB06C 0xB068	64	DM0DAIR1	DMAC0 Destination Address Increment Register 1
0xB070	0xB070 0xB074	0xB074 0xB070	64	DM0CCR1	DMAC0 Channel Control Register 1
0xB078	0xB078 0xB07C	0xB07C 0xB078	64	DM0CSR1	DMAC0 Channel Status Register 1
0xB080	0xB080 0xB084	0xB084 0xB080	64	DM0CHAR2	DMAC0 Chain Address Register 2
0xB088	0xB088 0xB08C	0xB08C 0xB088	64	DM0SAR2	DMAC0 Source Address Register 2
0xB090	0xB090 0xB094	0xB094 0xB090	64	DM0DAR2	DMAC0 Destination Address Register 2
0xB098	0xB098 0xB09C	0xB09C 0xB098	64	DM0CNTR2	DMAC0 Count Register 2
0xB0A0	0xB0A0 0xB0A4	0xB0A4 0xB0A0	64	DM0SAIR2	DMAC0 Source Address Increment Register 2
0xB0A8	0xB0A8 0xB0AC	0xB0AC 0xB0A8	64	DM0DAIR2	DMAC0 Destination Address Increment Register 2
0xB0B0	0xB0B0 0xB0B4	0xB0B4 0xB0B0	64	DM0CCR2	DMAC0 Channel Control Register 2
0xB0B8	0xB0B8 0xB0BC	0xB0BC 0xB0B8	64	DM0CSR2	DMAC0 Channel Status Register 2
0xB0C0	0xB0C0 0xB0C4	0xB0C4 0xB0C0	64	DM0CHAR3	DMAC0 Chain Address Register 3
0xB0C8	0xB0C8 0xB0CC	0xB0CC 0xB0C8	64	DM0SAR3	DMAC0 Source Address Register 3
0xB0D0	0xB0D0 0xB0D4	0xB0D4 0xB0D0	64	DM0DAR3	DMAC0 Destination Address Register 3
0xB0D8	0xB0D8 0xB0DC	0xB0DC 0xB0D8	64	DM0CNTR3	DMAC0 Count Register 3
0xB0E0	0xB0E0 0xB0E4	0xB0E4 0xB0E0	64	DM0SAIR3	DMAC0 Source Address Increment Register 3
0xB0E8	0xB0E8 0xB0EC	0xB0EC 0xB0E8	64	DM0DAIR3	DMAC0 Destination Address Increment Register 3

0xB0F0	0xB0F0 0xB0F4	0xB0F4 0xB0F0	64	DM0CCR3	DMAC0 Channel Control Register 3
0xB0F8	0xB0F8 0xB0FC	0xB0FC 0xB0F8	64	DM0CSR3	DMAC0 Channel Status Register 3
0xB148	0xB148 0xB14C	0xB14C 0xB148	64	DM0MFDR	DMAC0 Memory Fill Data Register
0xB150	0xB150 0xB154	0xB154 0xB150	64	DM0MCR	DMAC0 Master Control Register

6.6.11. Registers for DMA Controller (DMAC1)

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-19 Internal Registers for DMA Controller (DMAC1)

Offset Address			Register Size (bit)	Register Symbol	Register Name
DMA Controller (DMAC1)					
LD/SD	**LW/SW**				
LE/BE	**LE**	**BE**			
0xB800	0xB800 0xB804	0xB804 0xB800	64	DM1CHAR0	DMAC1 Chain Address Register 0
0xB808	0xB808 0xB80C	0xB80C 0xB808	64	DM1SAR0	DMAC1 Source Address Register 0
0xB810	0xB810 0xB814	0xB814 0xB810	64	DM1DAR0	DMAC1 Destination Address Register 0
0xB818	0xB818 0xB81C	0xB81C 0xB818	64	DM1CNTR0	DMAC1 Count Register 0
0xB820	0xB820 0xB824	0xB824 0xB820	64	DM1SAIR0	DMAC1 Source Address Increment Register 0
0xB828	0xB828 0xB82C	0xB82C 0xB828	64	DM1DAIR0	DMAC1 Destination Address Increment Register 0
0xB830	0xB830 0xB834	0xB834 0xB830	64	DM1CCR0	DMAC1 Channel Control Register 0
0xB838	0xB838 0xB83C	0xB83C 0xB838	64	DM1CSR0	DMAC1 Channel Status Register 0
0xB840	0xB840 0xB844	0xB844 0xB840	64	DM1CHAR1	DMAC1 Chain Address Register 1
0xB848	0xB848 0xb84C	0xB84C 0xb848	64	DM1SAR1	DMAC1 Source Address Register 1
0xB850	0xB850 0Xb854	0xB854 0Xb850	64	DM1DAR1	DMAC1 Destination Address Register 1
0xB858	0xB858 0xB85C	0xB85C 0xB858	64	DM1CNTR1	DMAC1 Count Register 1
0xB860	0xB860 0xB864	0xB864 0xB860	64	DM1SAIR1	DMAC1 Source Address Increment Register 1
0xB868	0xB868 0xB86C	0xB86C 0xB868	64	DM1DAIR1	DMAC1 Destination Address Increment Register 1
0xB870	0xB870 0xB874	0xB874 0xB870	64	DM1CCR1	DMAC1 Channel Control Register 1
0xB878	0xB878 0xB87C	0xB87C 0xB878	64	DM1CSR1	DMAC1 Channel Status Register 1
0xB880	0xB880 0xB884	0xB884 0xB880	64	DM1CHAR2	DMAC1 Chain Address Register 2
0xB888	0xB888 0xB88C	0xB88C 0xB888	64	DM1SAR2	DMAC1 Source Address Register 2
0xB890	0xB890 0xB894	0xB894 0xB890	64	DM1DAR2	DMAC1 Destination Address Register 2
0xB898	0xB898 0xB89C	0xB89C 0xB898	64	DM1CNTR2	DMAC1 Count Register 2
0xB8A0	0xB8A0 0xB8A4	0xB8A4 0xB8A0	64	DM1SAIR2	DMAC1 Source Address Increment Register 2
0xB8A8	0xB8A8 0xB8AC	0xB8AC 0xB8A8	64	DM1DAIR2	DMAC1 Destination Address Increment Register 2
0xB8B0	0xB8B0 0xB8B4	0xB8B4 0xB8B0	64	DM1CCR2	DMAC1 Channel Control Register 2
0xB8B8	0xB8B8 0xB8BC	0xB8BC 0xB8B8	64	DM1CSR2	DMAC1 Channel Status Register 2
0xB8C0	0xB8C0 0xB8C4	0xB8C4 0xB8C0	64	DM1CHAR3	DMAC1 Chain Address Register 3

0xB8C8	0xB8C8 0xB8CC	0xB8CC 0xB8C8	64	DM1SAR3	DMAC1 Source Address Register 3
0xB8D0	0xB8D0 0xB8D4	0xB8D4 0xB8D0	64	DM1DAR3	DMAC1 Destination Address Register 3
0xB8D8	0xB8D8 0xB8DC	0xB8DC 0xB8D8	64	DM1CNTR3	DMAC1 Count Register 3
0xB8E0	0xB8E0 0xB8E4	0xB8E4 0xB8E0	64	DM1SAIR3	DMAC1 Source Address Increment Register 3
0xB8E8	0xB8E8 0xB8EC	0xB8EC 0xB8E8	64	DM1DAIR3	DMAC1 Destination Address Increment Register 3
0xB8F0	0xB8F0 0xB8F4	0xB8F4 0xB8F0	64	DM1CCR3	DMAC1 Channel Control Register 3
0xB8F8	0xB8F8 0xB8FC	0xB8FC 0xB8F8	64	DM1CSR3	DMAC1 Channel Status Register 3
0xB948	0xB948 0xB94C	0xB94C 0xB948	64	DM1MFDR	DMAC1 Memory Fill Data Register
0xB950	0xB950 0xB954	0xB954 0xB950	64	DM1MCR	DMAC1 Master Control Register

6.6.12. Registers for PCI Controller (PCIC)

Use Word (LW/SW) load/store to access register.

Table 6-20 Internal Registers for PCI Controller (PCIC)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller (PCIC)			
LW/SW **LE/BE**			
0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
0xD010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
0xD014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
0xD018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
0xD01C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
0xD020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
0xD024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
0xD040	32	G2PTOCNT	G2P Timeout Count register (Retry Timeout Value, TRDY Timeout Value)
0xD080	32	G2PSTATUS	G2P Status Register
0xD084	32	G2PMASK	G2P Interrupt Mask Register
0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
0xD08C	32	PCIMASK	PCI Status Interrupt Mask Register
0xD090	32	P2GCFG	P2G Configuration Register
0xD094	32	P2GSTATUS	P2G Status Register
0xD098	32	P2GMASK	P2G Interrupt Mask Register
0xD09C	32	P2GCCMD	P2G Current Command Register
0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for a diagnosis)
0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for a diagnosis)
0xD11C	32	PBACSTATE	PCI Bus Arbiter Current Status Register (for a diagnosis)

6.6.13. Registers for GBUS to PCI Interface

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-21 Internal Registers for GBUS to PCI Interface

Offset Address			Register Size (bit)	Register Symbol	Register Name
GBUS to PCI Interface					
LD/SD	**LW/SW**				
LE/BE	**LE/BE**				
0xD120	(LE) 0xD120 0xD124	(BE) 0xD124 0xD120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
0xD128	(LE) 0xD128 0xD12C	(BE) 0xD12C 0xD128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
0xD130	(LE) 0xD130 0xD134	(BE) 0xD134 0xD130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
0xD138	(LE) 0xD138 0xD13C	(BE) 0xD13C 0xD138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
	0xD140		32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
	0xD144		32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
	0xD148		32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
	0xD14C		32	G2PIOMASK	G2P I/O Space Address Mask Register
0xD150	(LE) 0xD150 0xD154	(BE) 0xD154 0xD150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
0xD158	(LE) 0xD158 0xD15C	(BE) 0xD15C 0xD158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
0xD160	(LE) 0xD160 0xD164	(BE) 0xD164 0xD160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
0xD168	(LE) 0xD168 0xD16C	(BE) 0xD16C 0xD168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
	0xD170		32	PCICCFG	PCI Controller Configuration Register
	0xD174		32	PCICSTATUS	PCI Controller Status Register
	0xD178		32	PCICMASK	PCI Controller Interrupt Mask register
0xD180	(LE) 0xD180 0xD184	(BE) 0xD184 0xD180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
0xD188	(LE) 0xD188 0xD18C	(BE) 0xD18C 0xD188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
0xD190	(LE) 0xD190 0xD194	(BE) 0xD194 0xD190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
0xD198	(LE) 0xD198 0xD19C	(BE) 0xD19C 0xD198	64	P2GIOGBASE	P2G I/O Space 0 G-Bus Base Address Register
	0xD1A0		32	G2PCFGADRS	G2P Configuration Address Register
	0xD1A4		32	G2PCFGDATA	G2P Configuration Data Register
	0xD1C8		32	G2PINTACK	G2P Interrupt Acknowledge Register
	0xD1CC		32	G2PSPC	G2P Special Cycle Data Register
	0xD1D0		32	PCICDATA0	PCI Configuration Data 0 Register
	0xD1D4		32	PCICDATA1	PCI Configuration Data 1 Register
	0xD1D8		32	PCICDATA2	PCI Configuration Data 2 Register
	0xD1DC		32	PCICDATA3	PCI Configuration Data 3 Register
0xD200	(LE) 0xD200 0xD204	(BE) 0xD204 0xD200	64	PDMCA	PDMAC Chain Address Register
0xD208	(LE) 0xD208 0xD20C	(BE) 0xD20C 0xD208	64	PDMGA	PDMAC G-Bus Address Register
0xD210	(LE) 0xD210 0xD214	(BE) 0xD214 0xD210	64	PDMPA	PDMAC PCI Bus Address Register

0xD218	(LE) 0xD218 0xD21C	(BE) 0xD21C 0xD218	64	PDMCTR	PDMAC Count Register
0xD220	(LE) 0xD220 0xD224	(BE) 0xD224 0xD220	64	PDMCFG	PDMAC Configuration Register
0xD228	(LE) 0xD228 0xD22C	(BE) 0xD22C 0xD228	64	PDMSTATUS	PDMAC Status Register

6.6.14. Registers for Chip Configuration

Use Double Word (LD/SD) or Word (LW/SW) load/store to access register.

Table 6-22 Internal Registers for Chip Configuration

Offset Address		Register Size (bit)	Register Symbol	Register Name
Configuration				
LD/SD	**LW/SW**			
LE/BE	**LE/BE**			
0xE000	0xE000 0xE004	64	CCFG	Chip Configuration Register
0xE008	0xE008 0xE00C	64	REVID	Chip Revision ID Register
0xE010	0xE010 0xE014	64	PCFG	Pin Configuration Register
0xE018	0xE018 0xE01C	64	TOEA	Timeout Error Access Address Register
0xE020	0xE020 0xE024	64	CLKCTR	Clock Control Register
0xE030	0xE030 0xE034	64	GARBC	G-Bus Arbiter Control Register
0xE048	0xE048 0xE04C	64	RAMP	Register Address Mapping Register
0xE050	0xE050 0xE054	64	-	Reserved
0xE058	0xE058 0xE05C	64	-	Reserved
0xE060	0xE060 0xE064	64	DSKWCTRL	DLL DeSkew Control Register
0xE068	0xE068 0xE06C	64	MCLKOSC	Audio Clock PLL set-up
0xE070	0xE070 0xE074	64	MCLKCTL	Audio Clock Operation Control
0xE100	0xE100 0xE104	64	GPIOMR1	GPIO Mode Register 1
0xE108	0xE108 0xE10C	64	GPIODR1	GPIO Data Register 1
0xE110	0xE110 0xE114	64	GPIOMR2	GPIO Mode Register 2
0xE118	0xE118 0xE11C	64	GPIODR2	GPIO Data Register 2

6.6.15. Registers for Timer(s)

Use Word (LW/SW) load/store to access register.

Table 6-23 Internal Registers for Timer(s)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Timer (Channel 0)			
LW/SW **LE/BE**			
0xF000	32	TMTCR0	Timer Control Register 0
0xF004	32	TMTISR0	Timer Interrupt Status Register 0
0xF008	32	TMCPRA0	Compare Address Register A 0
0xF00C	32	TMCPRB0	Compare Address Register B 0
0xF010	32	TMITMR0	Interval Timer Mode Register 0
0xF020	32	TMCCDR0	Divider Register 0
0xF030	32	TMPGMR0	Pulse Generator Mode Register 0
0xF0F0	32	TMTRR0	Timer Read Register 0
Timer (Channel 1)			
0xF100	32	TMTCR1	Timer Control Register 1
0xF104	32	TMTISR1	Timer Interrupt Status Register 1
0xF108	32	TMCPRA1	Compare Address Register A 1
0xF10C	32	TMCPRB1	Compare Address Register B 1
0xF110	32	TMITMR1	Interval Timer Mode Register 1
0xF120	32	TMCCDR1	Divider Register 1
0xF130	32	TMPGMR1	Pulse Generator Mode Register 1
0xF1F0	32	TMTRR1	Timer Read Register 1
Timer (Channel 2)			
0xF200	32	TMTCR2	Timer Control Register 2
0xF204	32	TMTISR2	Timer Interrupt Status Register 2
0xF208	32	TMCPRA2	Compare Register A 2
0xF210	32	TMITMR2	Interval Timer Mode Register 2
0xF220	32	TMCCDR2	Divider Register 2
0xF240	32	TMWTMR2	Watch Dog Timer Register 2
0xF2F0	32	TMTRR2	Timer Read Register 2
Timer (Channel 3)			
0XFD00	32	TMTCR3	Timer Control Register 3
0XFD04	32	TMTISR3	Timer Interrupt Status Register 3
0xFD08	32	TMCPRA3	Compare Address Register A 3
0xFD0C	32	TMCPRB3	Compare Address Register B 3
0xFD10	32	TMITMR3	Interval Timer Mode Register 3
0xFD20	32	TMCCDR3	Divider Register 3
0xFD30	32	TMPGMR3	Pulse Generator Mode Register 3
0xFDF0	32	TMTRR3	Timer Read Register 3
Timer (Channel 4)			
0xFE00	32	TMTCR4	Timer Control Register 4
0xFE04	32	TMTISR4	Timer Interrupt Status Register 4
0xFE08	32	TMCPRA4	Compare Address Register A 4
0xFE0C	32	TMCPRB4	Compare Address Register B 4
0xFE10	32	TMITMR4	Interval Timer Mode Register 4
0xFE20	32	TMCCDR4	Divider Register 4
0xFE30	32	TMPGMR4	Pulse Generator Mode Register 4
0xFEF0	32	TMTRR4	Timer Read Register 4
Timer (Channel 5)			
0xFF00	32	TMTCR5	Timer Control Register 5
0xFF04	32	TMTISR5	Timer Interrupt Status Register 5
0xFF08	32	TMCPRA5	Compare Register A 5
0xFF10	32	TMITMR5	Interval Timer Mode Register 5
0xFF20	32	TMCCDR5	Divider Register 5
0xFF40	32	TMWTMR5	Watch Dog Timer Register 5
0xFFF0	32	TMTRR5	Timer Read Register 5

6.6.16. Registers for Serial I/O

Use Word (LW/SW) load/store to access register.

Table 6-24 Internal Registers for Serial I/O (Channel 0)

Offset Address	Register Size (bit)	Register Symbol	Register Name
LW/SW LE/BE			
0xF300	32	SILCR0	Line Control Register 0
0xF304	32	SIDICR0	DMA/Interrupt Control Register 0
0xF308	32	SIDISR0	DMA/ Interrupt Status Register 0
0xF30C	32	SISCISR0	Status Change Interrupt Status Register 0
0xF310	32	SIFCR0	FIFO Control Register 0
0xF314	32	SIFLCR0	Flow Control Register 0
0xF318	32	SIBGR0	Baud Rate Control Register 0
0xF31C	32	SITFIFO0	Transmitter FIFO Register 0
0xF320	32	SIRFIFO0	Receiver FIFO Register 0

Table 6-25 Internal Registers for Serial I/O (Channel 2)

Offset Address	Register Size (bit)	Register Symbol	Register Name
LW/SW LE/BE			
0xF380	32	SILCR2	Line Control Register 2
0xF384	32	SIDICR2	DMA/Interrupt Control Register 2
0xF388	32	SIDISR2	DMA/ Interrupt Status Register 2
0xF38C	32	SISCISR2	Status Change Interrupt Status Register 2
0xF390	32	SIFCR2	FIFO Control Register 2
0xF394	32	SIFLCR2	Flow Control Register 2
0xF398	32	SIBGR2	Baud Rate Control Register 2
0xF39C	32	SITFIFO2	Transmitter FIFO Register 2
0xF3A0	32	SIRFIFO2	Receiver FIFO Register 2

Table 6-26 Internal Registers for Serial I/O (Channel 1)

Offset Address	Register Size (bit)	Register Symbol	Register Name
LW/SW LE/BE			
0xF400	32	SILCR1	Line Control Register 1
0xF404	32	SIDICR1	DMA/Interrupt Control Register 1
0xF408	32	SIDISR1	DMA/ Interrupt Status Register 1
0xF40C	32	SISCISR1	Status Change Interrupt Status Register 1
0xF410	32	SIFCR1	FIFO Control Register 1
0xF414	32	SIFLCR1	Flow Control Register 1
0xF418	32	SIBGR1	Baud Rate Control Register 1
0xF41C	32	SITFIFO1	Transmitter FIFO Register 1
0xF420	32	SIRFIFO1	Receiver FIFO Register 1

Table 6-27 Internal Registers for Serial I/O (Channel 3)

Offset Address	Register Size (bit)	Register Symbol	Register Name
LW/SW LE/BE			
0xF480	32	SILCR3	Line Control Register 3
0xF484	32	SIDICR3	DMA/Interrupt Control Register 3
0xF488	32	SIDISR3	DMA/ Interrupt Status Register 3
0xF48C	32	SISCISR3	Status Change Interrupt Status Register 3
0xF490	32	SIFCR3	FIFO Control Register 3
0xF494	32	SIFLCR3	Flow Control Register 3
0xF498	32	SIBGR3	Baud Rate Control Register 3
0xF49C	32	SITFIFO3	Transmitter FIFO Register 3
0xF4A0	32	SIRFIFO3	Receiver FIFO Register 3

6.6.17. Registers for Interrupt Controller (IRC)

Use Word (LW/SW) load/store to access register.

Table 6-28 Internal Registers for Interrupt Controller (IRC)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Interrupt Controller (IRC)			
LW/SW **LE/BE**			
0xE800	32	IRDEN	Interrupt Detection Enable Register
0xE808	32	ISCIPB	Interrupt Source and Cause Binding Register
0xE810	32	IRDM0	Interrupt Detection Mode Register 0
0xE818	32	IRDM1	Interrupt Detection Mode Register 1
0xE820	32	IRLVL0	Interrupt Level Register 0
0xE828	32	IRLVL1	Interrupt Level Register 1
0xE830	32	IRLVL2	Interrupt Level Register 2
0xE838	32	IRLVL3	Interrupt Level Register 3
0xE840	32	IRLVL4	Interrupt Level Register 4
0xE848	32	IRLVL5	Interrupt Level Register 5
0xE850	32	IRLVL6	Interrupt Level Register 6
0xE858	32	IRLVL7	Interrupt Level Register 7
0xE860	32	IRLVL8	Interrupt Level Register 8
0xE868	32	IRLVL9	Interrupt Level Register 9
0xE870	32	IRLVL10	Interrupt Level Register 10
0xE878	32	IRLVL11	Interrupt Level Register 11
0xE880	32	IRLVL12	Interrupt Level Register 12
0xE888	32	IRLVL13	Interrupt Level Register 13
0xE890	32	IRLVL14	Interrupt Level Register 14
0xE898	32	IRLVL15	Interrupt Level Register 15
0xE8A0	32	IRMSK	Interrupt Mask Register
0xE8A8	32	IREDC	Interrupt Edge Detection Clear Register
0xE8B0	32	IRPND0	Interrupt Pending Register 0
0xE8B8	32	IRCS	Interrupt Current Status Register
0xE8C0	32	IRPND1	Interrupt Pending Register 1
0xE8C8	32	IRDM2	Interrupt Detection Mode Register 2
0xE8D0	32	IRDM3	Interrupt Detection Mode Register 3
0xE8D8	32	IRDBR0	Interrupt Debug Register 0
0xE8E0	32	IRDBR1	Interrupt Debug Register 1
0xE8E8	32	IRDBEN	Interrupt Debug Enable Register
0xE900	32	IRFLAG0	Interrupt Request Flag Register 0
0xE908	32	IRFLAG1	Interrupt Request Flag Register 1
0xE910	32	IRPOL	Interrupt Request Polarity Control Register
0xE918	32	IRRCNT	Interrupt Request Control Register
0xE920	32	IRMASKINT	Interrupt Request Internal Interrupt Mask Register
0xE928	32	IRMASKEXT	Interrupt Request External Interrupt Mask Register

6.6.18. Registers for AC Link

Use Word (LW/SW) load/store to access register.

Table 6-29 Internal Registers for AC Link

Offset Address	Register Size (bit)	Register Symbol	Register Name
AC-link Controller (ACLC)			
LW/SW			
LE/BE			
0xF700	32	ACCTLEN	ACLC Control Enable Register
0xF704	32	ACCTLDIS	ACLC Control Disable Register
0xF708	32	ACREGACC	ACLC CODEC Register Access Register
0xF710	32	ACINTSTS	ACLC Interrupt Status Register
0xF714	32	ACINTMSTS	ACLC Interrupt Masked Status Register
0xF718	32	ACINTEN	ACLC Interrupt Enable Register
0xF71C	32	ACINTDIS	ACLC Interrupt Disable Register
0xF720	32	ACSEMAPH	ACLC Semaphore Register
0xF740	32	ACGPIDAT	ACLC GPI Data Register
0xF744	32	ACGPODAT	ACLC GPO Data Register
0xF748	32	ACSLTEN	ACLC Slot Enable Register
0xF74C	32	ACSLTDIS	ACLC Slot Disable Register
0xF750	32	ACFIFOSTS	ACLC FIFO Status Register
0xF780	32	ACDMASTS	ACLC DMA Request Status Register
0xF784	32	ACDMASEL	ACLC DMA Channel Selection Register
0xF7A0	32	ACAUDODAT	ACLC Audio PCM Output Data Register
0xF7A4	32	ACSURRDAT	ACLC Surround Data Register
0xF7A8	32	ACCENTDAT	ACLC Center Data register
0xF7AC	32	ACLFEDAT	ACLC LFE Data Register
0xF7B0	32	ACAUDIDAT	ACLC Audio PCM Input Data Register
0xF7B8	32	ACMODODAT	ACLC Modem Output Data Register
0xF7BC	32	ACMODIDAT	ACLC Modem Input Data Register
0xF7FC	32	ACREVID	ACLC Revision ID Register

6.6.19. Registers for Serial Peripheral Interface (SPI)

Use Word (LW/SW) load/store to access register.

Table 6-30 Internal Registers for Serial Peripheral Interface (SPI)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Serial Peripheral Interface (SPI)			
LW/SW			
LE/BE			
0xF800	32	SPMCR	SPI Master Control Register
0xF804	32	SPCR0	SPI Control Register 0
0xF808	32	SPCR1	SPI Control Register 1
0xF80C	32	SPFS	SPI Inter Frame Space Register
0xF810	32	-	Reserved
0xF814	32	SPSR	SPI Status Register
0xF818	32	SPDR	SPI Data Register
0xF81C	32	-	Reserved

6.6.20. Registers for I2C Controller

Use Word (LW/SW) load/store to access register.

Table 6-31 Internal Registers for I2C Controller

Offset Address	Register Size (bit)	Register Symbol	Register Name
I2C Controller			
LW/SW			
LE/BE			
0xF900	32	I2C_ICTSR	I2C Interrupt Control/Status Register
0xF904	32	I2C_TXRR	I2C Transmit/Receive Register
0xF908	32	I2C_CR	I2C Command Register
0xF90C	32	I2C_PRE	I2C Prescale Register
0xF920	32	I2C_CTR	I2C Control Register

6.6.21. Registers for I2S Controller

Use Word (LW/SW) load/store to access register.

Table 6-32 Internal Registers for I2S Controller

Offset Address	Register Size (bit)	Register Symbol	Register Name
I2S Controller			
LW/SW			
LE/BE			
0xFA00	32	I2SMCR	I2S Channel Main Control Register
0xFA04	32	I2SCCR	I2S Channel Control Register
0xFA08	32	I2SICTRL	I2S Interrupt Control Register
0xFA0C	32	I2SCOR	I2S Clock Option Register
0xFA80	32	I2SCH0D	Channel 0 Data
0xFA90	32	I2SCH1D	Channel 1 Data
0xFAA0	32	I2SCH2D	Channel 2 Data

6.6.22. Registers for RTC Controller

Use Word (LW/SW) load/store to access register.

Table 6-33 Internal Registers for RTC Controller

Offset Address	Register Size (bit)	Register Symbol	Register Name
RTC Controller			
LW/SW			
LE/BE			
0xFB00	32	RTCCTL	Control and Status Register
0xFB04	32	RTCADR	Address Register
0xFB08	32	RTCDAT	Data port to access the contents of RTC registers
0xFB0C	32	RTCTBC	Time Base Corrector Register

6.6.23. Registers for CIR Controller

Use Word (LW/SW) load/store to access register.

Table 6-34 Internal Register for CIR Controller

Offset Address	Register Size (bit)	Register Symbol	REGISTER NAME
CIR Controller			
LW/SW			
LE/BE			
0xFC00	32	IRC_CSR	IRC Control/Status Register

Chapter 7. Configuration Registers

7.1. Detailed Description

The configuration registers set up and control the basic functionality of the entire TX4939. Refer to Section 5.2 for details of each configuration register. Also refer to sections mentioned in the description about each bit field.

7.1.1. Detecting G-Bus Timeout

The G-bus is an internal bus of the TX4939. Access to each address on the G-Bus is completed upon a bus response from the accessed address. If an attempt is made to access an undefined physical address or if a hardware failure occurs, no bus response is made. If a bus response does not occur, the bus access will not be completed, leading to a system halt. To solve this problem, the TX4939 is provided with a G-Bus timeout detection function. This function forcibly stops bus access if no bus response occurs within the specified time.

Setting the G-Bus Timeout Error Detection bit (CCFG.TOE) of the chip configuration register enables the G-Bus timeout detection function. If a bus response does not occur within the G-Bus clock (GBUSCLK) cycle specified in the G-Bus Timeout Time field (CCFG.GTOT), the G-Bus timeout detection function makes an error response to force the bus access to end. The accessed address is stored to the timeout error access address register (TOEA).

If a timeout error is detected while the TX49/H4 core, as the bus master, is gaining write access to the G-Bus, the Write-Access Bus Error bit (CCFG.BEOW) is set. Enabling interrupt No. 2 in the interrupt controller makes it possible to post an interrupt to the TX49/H4 core. If a timeout error is detected while the TX49/H4 core is gaining read access to the bus, a bus error exception occurs in the TX49/H4 core.

If a timeout error is detected while another G-Bus master (the PCI controller or DMA controller) is accessing the G-Bus, an error bit in that controller is set, which can be used to post an interrupt. Refer to the descriptions of each controller for details.

If the TRST* signal is deasserted, it is assumed that an EJTAG probe is connected, so the G-Bus timeout detection feature is disabled.

7.2. Registers

All registers are 64 bit. Firmware must use DWord or Word instructions to access to these registers.

Table 7-1 lists the configuration registers.

Table 7-1 Configuration Register Mapping

Offset Address	Size in Bits	Register Symbol	Register Name
0xE000	64	CCFG	Chip Configuration Register
0xE008	64	REVID	Chip Revision ID Register
0xE010	64	PCFG	Pin Configuration Register
0xE018	64	TOEA	Timeout Error Access Address Register
0xE020	64	CLKCTR	Clock Control Register
0xE030	64	GARBC	G-Bus Arbiter Control Register
0xE048	64	RAMP	Register Address Mapping Register
0xE060	64	DSKWCTRL	DLL DeSkew Control Register
0xE068	64	MCLKOSC	Audio Clock PLL set-up
0xE070	64	MCLKCTL	Audio Clock Operation Control
0xE100	64	GPIOMR1	GPIO Mode Register 1
0xE108	64	GPIODR1	GPIO Data Register 1
0xE110	64	GPIOMR2	GPIO Mode Register 2
0xE118	64	GPIODR2	GPIO Data Register 2

Any address not defined in this table is reserved for future use.

7.2.1. Chip Configuration Register (CCFG) 0xE000

For the bit fields whose initial values are set by boot configuration (refer to Section 4.2), the initial input signal level and the corresponding register value are indicated.

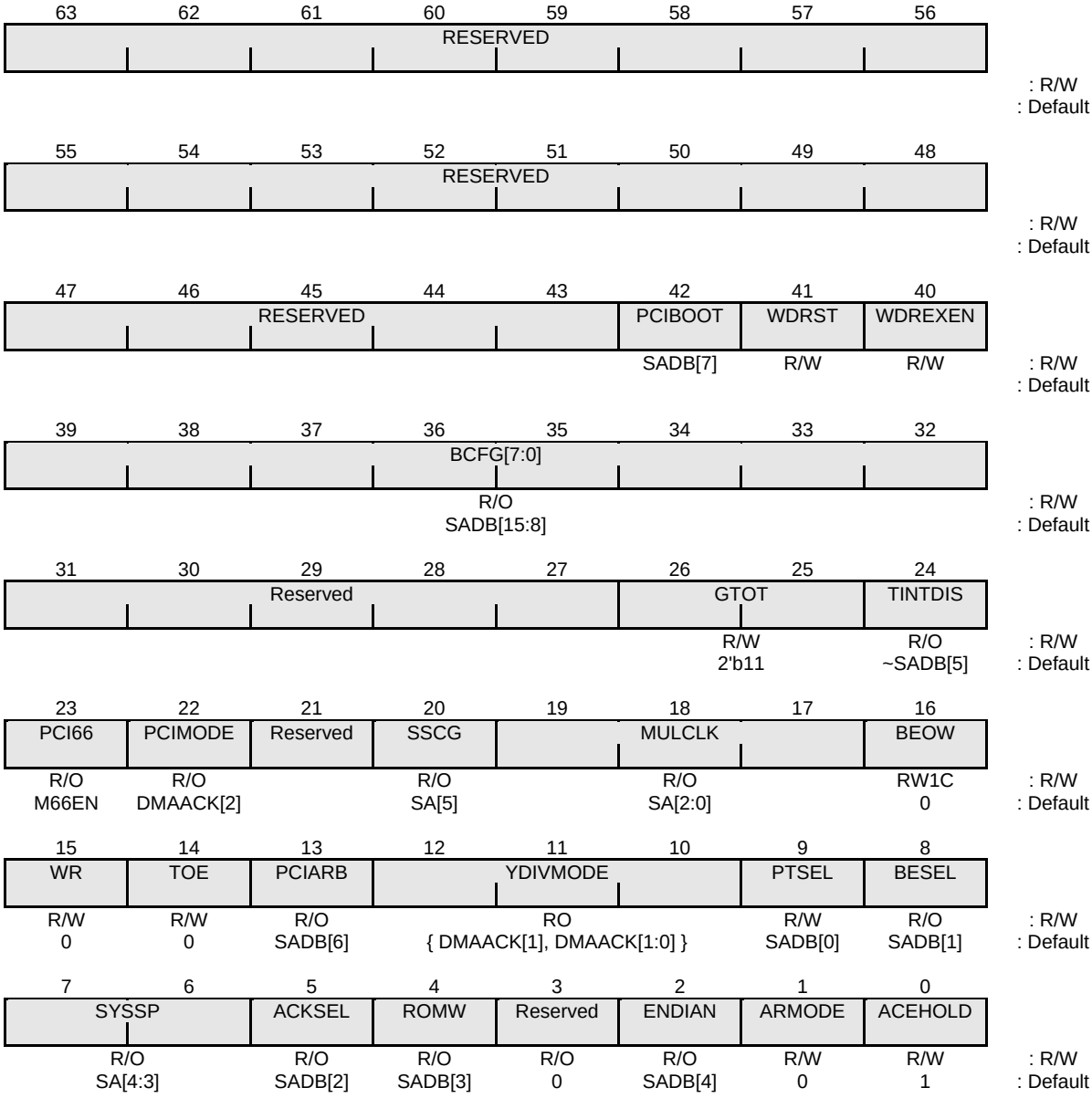


Figure 7-1 Chip Configuration Register

Table 7-2 Chip Configuration Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:43	—	Reserved	—	—	—
42	PCIBOOT	PCI Boot Configuration	PCI Boot Option 0 = PCI Boot off 1 = PCI Boot on	SADB[7]	R/O
41	WDRST	Watchdog Reset Status	Watch Dog Reset Status (Initial Value 0, RW1C) Indicates that a watchdog reset has occurred. Initialized when RESET* is asserted. 0 = No watchdog reset has occurred. 1 = A watchdog reset has occurred	0	RW1C
40	WDREXEN	Watchdog Reset External Output	Watch Dog Reset External Enable (Initial Value 0, R/W) Specifies whether to assert the WDRST* signal at a watchdog reset. Initialized when RESET* is asserted. 0 = Do not assert the WDRST* signal. 1 = Assert the WDRST* signal.	0	R/W
39:32	BCFG[7:0]	Boot Configuration	Set to 1 at a reset if the corresponding SADB[15:8] signal is high. Set to 0 at a reset if the corresponding SADB[15:8] signal is low.	SADB[15:8]	R/O
31:27	—	Reserved	—	—	—
26:25	GTOT	G-Bus Timeout Time	Specifies the number of G-Bus clock (GBUSCLK) cycles after which a bus timeout error will occur on the internal bus (G-Bus) of the TX4939. 11 = 4096 GBUSCLK 10 = 2048 GBUSCLK 01 = 1024 GBUSCLK 00 = 512 GBUSCLK	2'b11	R/W
24	TINTDIS	Disable TX49/H4 Core Timer Interrupt	Indicates a value for indicating whether to enable the TX49/H4 internal timer interrupt. This signal is inverted signal of TIMEREN 0: The TX49/H4 internal timer interrupt is enabled. 1: The TX49/H4 internal timer interrupt is disabled.	~SADB[5]	R/O
23	PCI66	PCI 66MHz Mode	Used to inform the device connected to the PCI bus that a 66 MHz operation is to be performed. 0 = Perform a 33 MHz operation. 1 = Perform a 66 MHz operation.	M66EN	R/O
22	PCIMODE	Host/Satellite Control	Indicates the PCI Host/Satellite selection setting 1 = Host 0 = Satellite	DMAACK[2]	R/O
21	—	Reserved	—	—	—
20	SSCG	SSCG Control	Specifies whether the Spread Spectrum Modulator is bypassed 0 = SSCG Disabled (Bypass) 1 = SSCG Enabled	SA[5]	R/O

Table 7-2 Chip Configuration Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
19:17	MULCLK	CPUCLK Frequency Multiplication Factor	Indicates information about the frequency multiplication factor of the TX49/H4 core clock (CPUCLK) to the MSTCLK. This field is set with a result of encoding an initial input value at SA[2:0]. The PLL incorporated in the TX4939 multiplies the MSTCLK and supplies the resulting frequency to the TX49/H4 core. The value set in YMULCLK [4:0] is reflected in the EC field of the TX49/H4 core Configuration register. MULCLK[2:0] ==> YMULCLK[4:0] = ND[4:0] MULCLK[2:0] / ND[4:0] / PLL#2 / CPU Clock 000 / 5'b01000 / 600 MHz / 300 MHz 001 / 5'b 01001 / 666 MHz / 333 MHz 010 / 5'b 01010 / 733 MHz / 366 MHz 011 / 5'b 01011 / 800 MHz / 400 MHz 100 / 5'b 01100 / Reserved / Reserved 101 / 5'b 01101 / Reserved / Reserved 110 / 5'b 01110 / Reserved / Reserved 111 / 5'b 00111 / 533 MHz / 266 MHz The following equation defines the CPUCLK frequency as a function of the MSTCLK frequency and YMULCLK: CPUCLK(f) = (25 x YMULCLK x MSTCLK(f)) / 36	SA[2:0]	R/O
16	BEOW	Write-Access Bus Error	Indicates that a timeout error has occurred in the internal bus (G-Bus) during a write bus transaction of the TX49/H4 core. This bit corresponds to interrupt No. 2 in the interrupt controller. 0 = No error has occurred. 1 = An error has occurred.	0	RW1C
15	WR	Watchdog Timer Mode	Specifies how information will be reported in watchdog timer mode. 0 = Generate an NMI exception. 1 = Generate a watchdog reset.	0	R/W
14	TOE	G-Bus Timeout Error Detection	Specifies whether to detect and report a bus timeout error in the internal bus (G-Bus) of the TX4939. 0 = Do not detect or report a bus timeout error. 1 = Detect and report a bus timeout error.	0	R/W
13	PCIARB	PCI Arbiter Selection	Indicates the PCI bus arbiter selection setting 0 = Select external PCI bus arbiter 1 = Select built-in PCI bus arbiter	SADB[6]	R/O
12:10	YDIVMODE	GBUSCLK Frequency Division Ratio	Specifies the frequency division ratio of the GBUS clock output (GBUSCLK) frequency to the clock frequency (CPUCLK) of the TX49/H4 core. 000:GBUSCLK frequency = CPUCLK frequency ÷ 2 001:GBUSCLK frequency = CPUCLK frequency ÷ 3 110:GBUSCLK frequency = CPUCLK frequency ÷ 5 111:GBUSCLK frequency = CPUCLK frequency ÷ 6	{ DMAACK[1], DMAACK[1:0] }	R/O
9	PTSEL	PC Trace Mode	PC Trace Mode Enable 0 = PC Trace Disable 1 = PC Trace Enable	SADB[0]	R/W
8	BESEL	BE function	Specifies the function of BE[1:0]*/BWE[1:0]* pins upon booting 0: BE[1:0]* (Byte Enable) 1: BWE[1:0]* (Byte Write Enable)	SADB[1]	R/O
7:6	SYSSP	SYSCLK frequency division ratio	Indicates the frequency division ratio of the SYSCLK frequency to the G-Bus clock frequency (GBUSCLK). 00: SYSCLK frequency = GBUSCLK frequency ÷ 4 01: SYSCLK frequency = GBUSCLK frequency ÷ 3 10: SYSCLK frequency = GBUSCLK frequency ÷ 5 11: SYSCLK frequency = GBUSCLK frequency ÷ 6	SA[4:3]	R/O
5	ACKSEL	Boot ACK* I/P	Specifies the access mode for external bus controller channel0 0 = External ACK mode 1 = Normal mode	SADB[2]	R/O
4	ROMW	Boot ROM Bus Width	Specifies the data bus width when booting from a memory device connected to the local bus controller ROMW = 1'b0 = 16 bits ROMW = 1'b1 = 8 bits	SADB[3]	R/O

Table 7-2 Chip Configuration Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
3	---	Reserved	---	---	R/O
2	ENDIAN	Endian	Indicates the TX4939 Endian mode setting. 0 = Little Endian mode 1 = Big Endian mode	SADB[4]	R/O
1	ARMODE	ACK*/READY Mode	Selects an ACK*/READY signal operation mode for the external bus controller (refer to Section 7.3.6). 0 = ACK*/READY dynamic mode 1 = ACK*/READY static mode	0	R/W
0	ACEHOLD	ACE Hold	Specifies the hold time of an address relative to the external bus controller ACE* signal (refer to Section 7.3.4). 0 = Switch the address at the same time when the ACE* signal is deasserted. 1 = Switch the address one clock cycle after the ACE* signal is deasserted.	1	R/W

7.2.2. Chip Revision ID Register (REVID) 0xE008

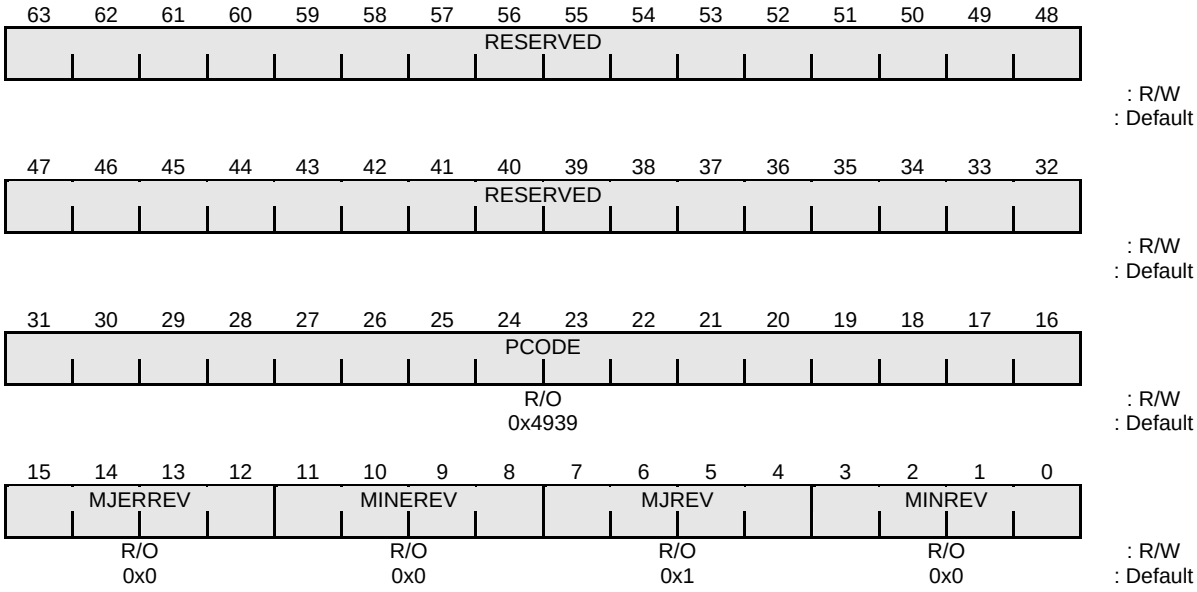


Figure 7-2 Chip Revision ID Register

Table 7-3 Chip Revision ID Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:32	—	Reserved	—	—	—
31:16	PCODE	Product Code	Indicates the product number. It is a fixed value.	0x4939	R
15:12	MJERREV	Major Extra Code	Indicates the major extra code.	0x0	R
11:8	MINEREV	Major Extra Code	Indicates the minor extra code.	0x0	R
7:4	MJREV	Major Revision Code	Indicates the major revision of the product. Contact Toshiba technical staff for the latest information.	0x1	R
3:0	MINREV	Minor Revision Code	Indicates the minor revision of the product. Contact Toshiba technical staff for the latest information.	0x0	R

7.2.3. Pin Configuration Register (PCFG) 0xE010

For the bit fields whose initial values are set by boot configuration (refer to Section 4.2), the initial input signal level and the corresponding register value are indicated.

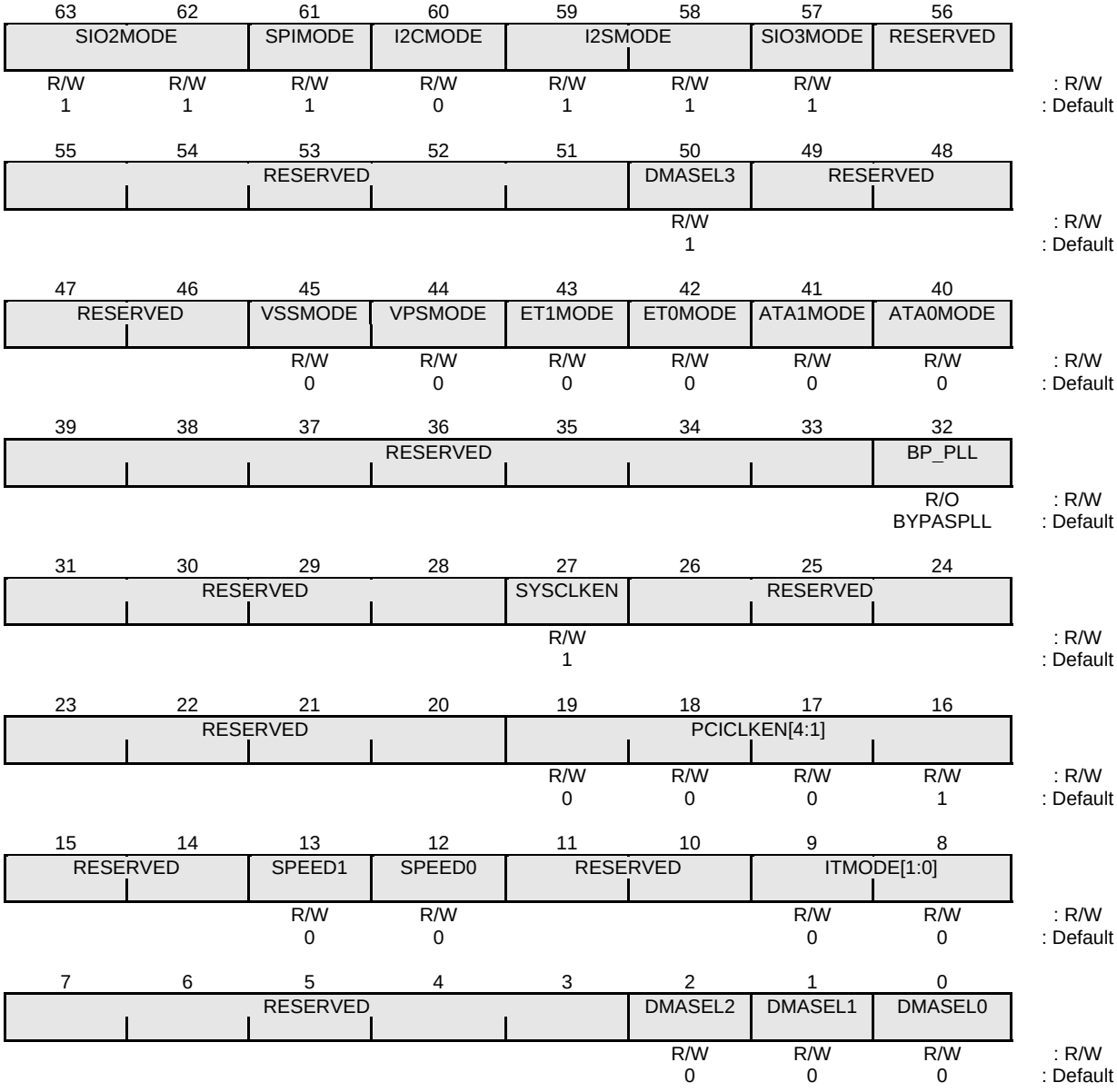


Figure 7-3 Pin Configuration Register

Table 7-4 Pin Configuration Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:62	SIO2MODE[1:0]	Setting of shared pins	SIO2 Shared-pin setting 1x: GPIO Mode (GPIO12, GPIO13) 01: SIO2 Mode (GPIO12 = RXD2, GPIO13 = TXD2) 00: SIO0 Mode (GPIO12 = CTS0, GPIO13 = RTS0)	11	R/W
61	SPIMODE	Setting of shared pin	SPI Shared-pin setting 1: SIO/GPIO Mode 0: SPI Mode	1	R/W
60	I2CMODE	Setting of shared pin	I2C Shared-pin setting 1 : I2C Mode 0 : GPIO Mode	0	R/W
59:58	I2SMODE	Setting of shared pin	I2S Shared-pin setting 11 : GPIO Mode 10 : I2S Mode 01 : I2S Mode 00 : ACLC Mode	11	R/W
57	SIO3MODE	Setting of shared pin	SIO3 Shared-pin setting 1 : GPIO Mode 0 : SIO Mode Used only when SPIMODE = 1	1	R/W
56:51	—	Reserved	—	—	—
50	DMASEL3	DMA Request Select 3	Selects a DMA request used by DMA controller 0 channel 3. 0: NDFC 1: SIO channel 0 transmission	1	R/W
49:46	—	Reserved	—	—	—
45:44	VSSMODE,VPSMODE	Setting of shared pin	Video Port Shared-pin setting {PTSEL,VSSMODE,VPSMODE} 000 : All GPIO 001 : 1-Parallel ports and GPIO 010 : 3-Serial port and GPIO 011 : 1-Parallel and 1-Serial port 1xx : PC Trace Mode Please refer to Table3-8 in Chapter 3	00	R/W
43	ET1MODE	Setting of shared pin	Ethernet MAC1 Shared-pin setting 1 : MAC1 Mode 0 : Other Mode Please refer to Table3-7 in Chapter 3	0	R/W
42	ET0MODE	Setting of shared pin	Ethernet MAC0 Shared-pin setting 1 : MAC0 Mode 0 : Other Mode Please refer to Table3-7 in Chapter 3	0	R/W
41	ATA1MODE	Setting of shared pin	ATA1 Shared-pin setting 1 : ATA1 Mode 0 : Other Mode Please refer to Table3-7 in Chapter 3	0	R/W
40	ATA0MODE	Setting of shared pin	ATA0 Shared-pin setting 1 : ATA0 Mode 0 : GPIO Mode	0	R/W
39:33	—	Reserved	—	—	—
32	BP_PLL	Bypass PLL	Indicates information about whether internal PLLs are on or off. 1 = The PLL is on 0 = The PLL is off..	BYPASSPLL*	R
31:28	—	Reserved	—	—	—
27	SYSCLKEN	SYSCLK Enable	Specifies whether to output the SYSCLK. 1 = Clock output 0 = Tri-state	1	R/W
26:20	—	Reserved	—	—	—
19	PCICLKEN[4]	PCICLK4 Enable	PCICLK4 Enable 0: IOSRST* 1: Clock output	0	R/W
18	PCICLKEN[3]	PCICLK3 Enable	PCICLK3 Enable 0: SYSRST* 1: Clock output	0	R/W
17	PCICLKEN[2]	PCICLK2 Enable	PCICLK2 Enable 0: L 1: Clock output	0	R/W
16	PCICLKEN[1]	PCICLK1 Enable	PCICLK1 Enable 0: L 1: Clock output	1	R/W

15:14	—	Reserved	—	—	—
13	SPEED1	RMII1 SPEED	RMII1 Speed Select 0: 10 Mbps transfer rate for RMII1 1: 100 Mbps transfer rate for RMII1 This bit used only if external PHY does not have pin to indicate the speed.	0	R/W
12	SPEED0	RMII0 SPEED	RMII0 Speed Select 0: 10 Mbps transfer rate for RMII0 1: 100 Mbps transfer rate for RMII0 This bit used only if external PHY does not have pin to indicate the speed.	0	R/W
11:10	—	Reserved	—	—	—
9:8	ITMODE[1:0]	Internal Test Mode	Internal Test Mode 00: Normal operation For normal operation, must keep these two bits 00. These bits are only for debug purpose.	00	R/W
7:3	—	Reserved	—	—	—
2	DMASEL2	DMA Request Select 2	Selects a DMA request used by DMA controller 0 channel 2. 0: DMAREQ[2] (external) 1: SIO channel 0 reception (internal)	0	R/W
1:0	DMASEL1 & DMASEL0	DMA Request Select 1 & Select 0	Selects a DMA request used by DMA controller 0 channel 1. 00: DMAREQ[1] (external) 01: Reserved	00	R/W

7.2.4. Timeout Error Access Address Register (TOEA) 0xE018

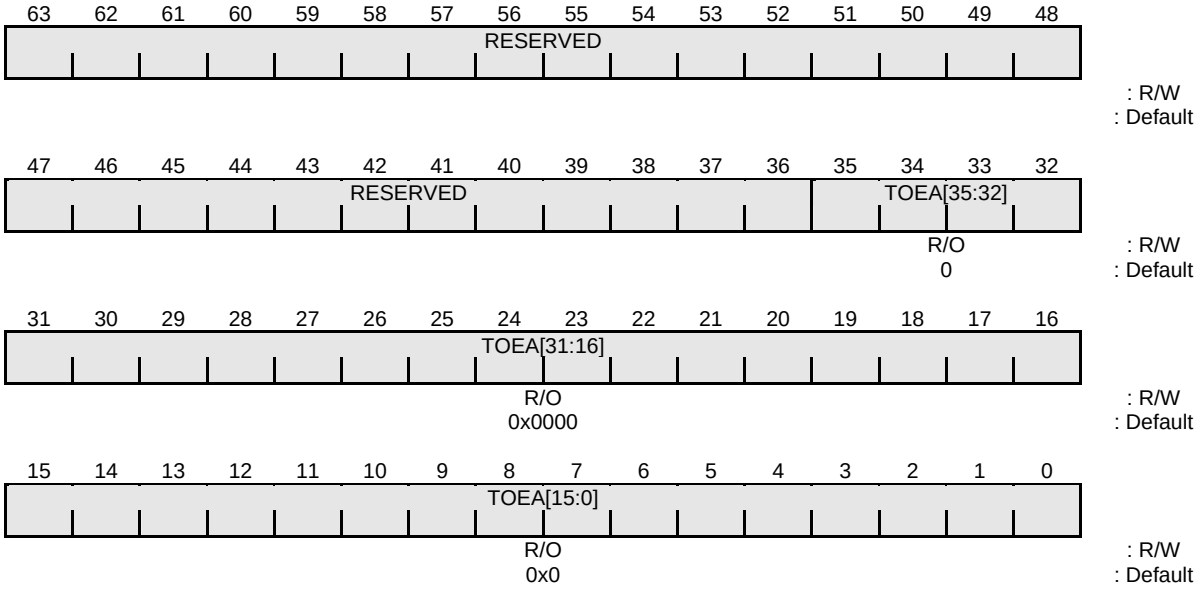


Figure 7-4 Timeout Error Access Register

Table 7-5 Timeout Error Access Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:36	—	Reserved	—	—	—
35:0	TOEA	Timeout Error Access Address	Holds the G-Bus address for the G-Bus cycle in which the latest G-Bus timeout error was detected.	0x0_0000_0000	R

7.2.5. Clock Control Register (CLKCTR) 0xE020

For the low-order 19 bits of the clock control register, canceling a reset requires that the corresponding reset bit be cleared by software. Before clearing them, wait at least **128** CPU clock cycles after they are set.

63	62	61	60	59	58	57	56	
RESERVED		TM5CKD	TM4CKD	TM3CKD	CIRCKD	SIO3CKD	SIO2CKD	
						R/W	R/W	: R/W
						0	0	: Default
55	54	53	52	51	50	49	48	
SIO1CKD	VPCCKD	RESERVED		ETH1CKD	ATA1CKD	BROMCKD	NDCCKD	
R/W	R/W			R/W	R/W	R/W	R/W	: R/W
0	0			0	0	0	0	: Default
47	46	45	44	43	42	41	40	
I2CCKD	ETH0CKD	SPICKD	SRAMCKD	PCI1CKD	DMA1CKD	ACLCKD	ATA0CKD	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: R/W
0	0	0	0	0	0	0	0	: Default
39	38	37	36	35	34	33	32	
DMA0CKD	PCICCKD	I2SCKD	TM0CKD	TM1CKD	TM2CKD	SIO0CKD	CYPCKD	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: R/W
0	0	0	0	0	0	0	0	: Default
31	30	29	28	27	26	25	24	
IOSRST	SYSRST	TM5RST	TM4RST	TM3RST	CIRRST	SIO3RST	SIO2RST	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: R/W
0	0	0	0	0	0	0	0	: Default
23	22	21	20	19	18	17	16	
SIO1RST	VPCRST	EPCIRST	RESERVED	ETH1RST	ATA1RST	BROMRST	NDCRST	
R/W	R/W	R/W		R/W	R/W	R/W	R/W	: R/W
0	0	0		0	0	0	0	: Default
15	14	13	12	11	10	9	8	
I2CRST	ETH0RST	SPIRST	SRAMRST	PCI1RST	DMA1RST	ACLRST	ATA0RST	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: R/W
0	0	0	0	1	0	0	0	: Default
7	6	5	4	3	2	1	0	
DMA0RST	PCICRST	I2SRST	TM0RST	TM1RST	TM2RST	SIO0RST	CYPRST	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: R/W
0	0	0	0	0	0	0	0	: Default

Figure 7-5 Clock Control Register

Table 7-6 Clock Control Register

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
63:62	—	Reserved	—	—	—
61	TMR5CKD	Timer 5 Clock Disable	Controls clock pulses for the TMR5 controller 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
60	TMR4CKD	Timer 4 Clock Disable	Controls clock pulses for the TMR4 controller 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
59	TMR3CKD	Timer 3 Clock Disable	Controls clock pulses for the TMR3 controller 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
58	CIRCKD	CIR Clock Disable	Controls clock pulses for the CIR controller 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
57	SIO3CKD	SIO3 Clock Disable	Controls clock pulses for the SIO3 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
56	SIO2CKD	SIO2 Clock Disable	Controls clock pulses for the SIO2 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
55	SIO1CKD	SIO1 Clock Disable	Controls clock pulses for the SIO1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
54	VPCCKD	VPC Clock Disable	Controls clock pulses for the VPC controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
53:52	—	Reserved	—	—	—
51	ETH1CKD	ETH1 Clock Disable	Controls clock pulses for the Ethernet MAC 1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
50	ATA1CKD	ATA1 Clock Disable	Controls clock pulses for the ATA1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
49	BROMCKD	BROM Clock Disable	Controls clock pulses for the BROM/SRAM controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
48	NDCCKD	NDC Clock Disable	Controls clock pulses for the NAND Flash controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
47	I2CCKD	I2C Clock Disable	Controls clock pulses for the I2C controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
46	ETH0CKD	ETHERC0 Clock Disable	Controls clock pulses for the Ethernet MAC 0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
45	SPICKD	SPIC Clock Disable	Controls clock pulses for the SPI controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
44	SRAMCKD	SRAMC Clock Disable	Controls clock pulses for the internal SRAM controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
43	PCIC1CKD	PCIC1 Clock Disable	Controls clock pulses for PCI controller 1. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
42	DMA1CKD	DMAC1 Clock Disable	Controls clock pulses for the DMA controller 1. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
41	ACLCKD	ACLC Clock Disable	Controls clock pulses for the AC-link controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
40	ATA0CKD	ATA0 Clock Disable	Controls clock pulses for the ATA0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
39	DMA0CKD	DMAC0 Clock Disable	Controls clock pulses for the DMA controller 0. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W

Table 7-6 Clock Control Register

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
38	PCICKD	PCIC Clock Disable	Controls clock pulses for the PCI controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
37	I2SCKD	I2S Clock Disable	Controls clock pulses for the I2S controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
36	TM0CKD	Timer 0 Clock Disable	Controls clock pulses for the TMR0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
35	TM1CKD	Timer 1 Clock Disable	Controls clock pulses for the TMR1 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
34	TM2CKD	Timer 2 Clock Disable	Controls clock pulses for the TMR2 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
33	SIO0CKD	SIO0 Clock Disable	Controls clock pulses for the SIO0 controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
32	CYPCKD	CYP Clock Disable	Controls clock pulses for the Cipher controller. 0 = Supply clock pulses. 1 = Do not supply clock pulses.	0	R/W
31	IOSRST	IOSRST Reset	Resets IO signal IOSRST_. 0 = Normal state 1 = Reset	0	R/W
30	SYSRST	SYSRST Reset	Resets IO signal SYSRST_. 0 = Normal state 1 = Reset	0	R/W
29	TM5RST	TMR5 Reset	Resets the TMR5 controller. 0 = Normal state 1 = Reset	0	R/W
28	TM4RST	TMR4 Reset	Resets the TMR4 controller. 0 = Normal state 1 = Reset	0	R/W
27	TM3RST	TMR3 Reset	Resets the TMR3 controller. 0 = Normal state 1 = Reset	0	R/W
26	CIRRST	CIR Reset	Resets the CIR controller. 0 = Normal state 1 = Reset	0	R/W
25	SIO3RST	SIO3 Reset	Resets the SIO3 controller. 0 = Normal state 1 = Reset	0	R/W
24	SIO2RST	SIO2 Reset	Resets the SIO2 controller. 0 = Normal state 1 = Reset	0	R/W
23	SIO1RST	SIO1 Reset	Resets the SIO1 controller. 0 = Normal state 1 = Reset	0	R/W
22	VPCRST	VPC Reset	Resets the VPC controller. 0 = Normal state 1 = Reset	0	R/W
21	EPCIRST	EPCIRST Reset	Resets IO signal PCIRST_. 0 = Normal state 1 = Reset	0	R/W
20	—	Reserved	—	—	—
19	ETH1RST	ETH1 Reset	Resets the Ethernet MAC 1 controller. 0 = Normal state 1 = Reset	0	R/W
18	ATA1RST	ATA1 Reset	Resets the ATA1 controller. 0 = Normal state 1 = Reset	0	R/W
17	BROMRST	BROM Reset	Resets the BROM controller. 0 = Normal state 1 = Reset	0	R/W
16	NDCRST	NDC Reset	Resets the NAND Flash controller. 0 = Normal state 1 = Reset	0	R/W

Table 7-6 Clock Control Register

Bit	Mnemonic	Field Name	Description	Initial Value	Read/Write
15	I2CRST	I2C Reset	Resets the I2C controller. 0 = Normal state 1 = Reset	0	R/W
14	ETHC0RST	ETHERC0 Reset	Resets the ETHER controller 0. 0 = Normal state 1 = Reset	0	R/W
13	SPIRST	SPI Reset	Resets the SPI controller. 0 = Normal state 1 = Reset	0	R/W
12	SRAMRST	SRAMC Reset	Resets the internal SRAM controller. 0 = Normal state 1 = Reset	0	R/W
11	PCIC1RST	PCIC1 Reset	Resets the PCI controller 1. 0 = Normal state 1 = Reset	1	R/W
10	DMA1RST	DMAC1 Reset	Resets the DMAC controller 1. 0 = Normal state 1 = Reset	0	R/W
9	ACLRST	ACLC Reset	Resets the AC-link controller. 0 = Normal state 1 = Reset	0	R/W
8	ATA0RST	ATA0 Reset	Resets the ATA0 controller. 0 = Normal state 1 = Reset	0	R/W
7	DMA0RST	DMAC Reset	Resets the DMA controller 0. 0 = Normal state 1 = Reset	0	R/W
6	PCICRST	PCIC Reset	Resets the PCI controller 0. 0 = Normal state 1 = Reset	0	R/W
5	I2SRST	I2S Reset	Resets the I2S controller. 0 = Normal state 1 = Reset	0	R/W
4	TM0RST	TMR0 Reset	Resets the TMR0 controller. 0 = Normal state 1 = Reset	0	R/W
3	TM1RST	TMR1 Reset	Resets the TMR1 controller. 0 = Normal state 1 = Reset	0	R/W
2	TM2RST	TMR2 Reset	Resets the TMR2 controller. 0 = Normal state 1 = Reset	0	R/W
1	SIO0RST	SIO0 Reset	Resets the SIO0 controller. 0 = Normal state 1 = Reset	0	R/W
0	CYPRST	CYP Reset	Resets the CYP controller. 0 = Normal state 1 = Reset	0	R/W

7.2.6. G-Bus Arbiter Control Register (GARBC) 0xE030

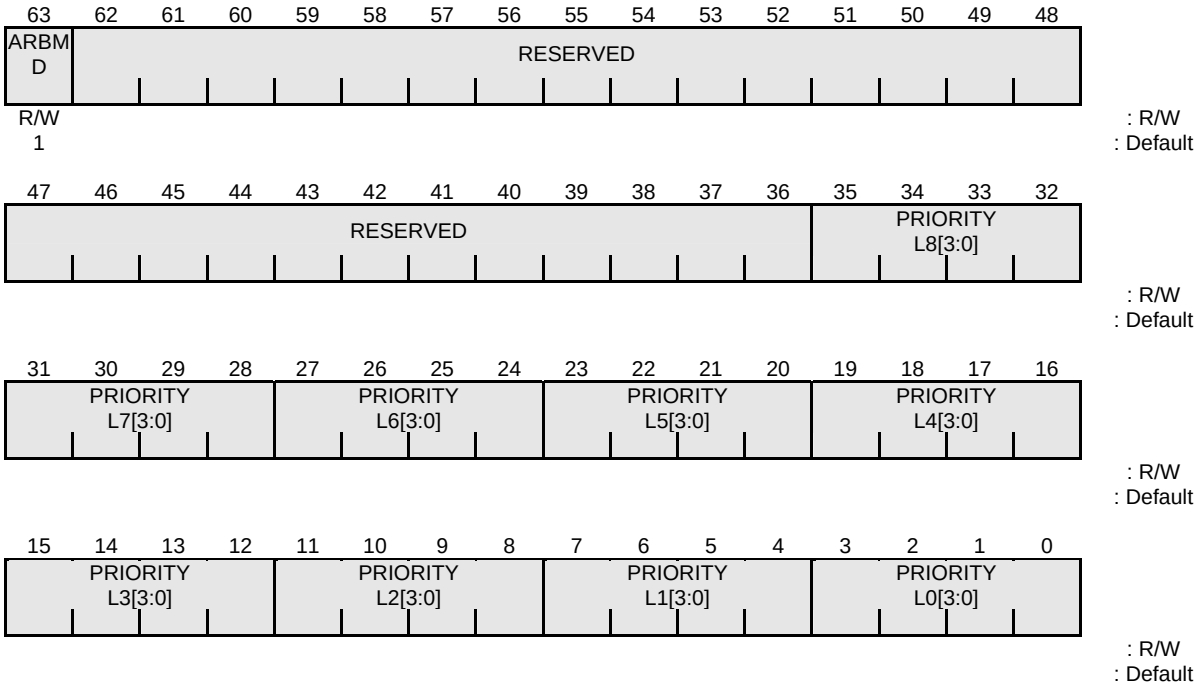


Figure 7-6 G-Bus Arbiter Control Register

Table 7-7 G-Bus Arbiter Control Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63	ARBMD	Arbitration Mode	Specifies how to prioritize G-Bus arbitration. 0 = Fixed priority. The G-Bus arbitration priority conforms to the content of the PRIORITY field (bits [23:0]). 1 = Round-robin (in a round-robin fashion, PCIC0 > PDMAC > DMAC0 > DMAC1 > PCIC1 > ATA0 > ATA1 > CYP > VPC) Note: Before accessing the PCI by DMAC, specify round-robin as the priority mode. **If fixed-priority mode is selected, a dead lock is likely to occur in PCI bus access.**	1	R/W
62:36	—	Reserved	—	—	—
35:0	PRIORITY	Arbitration Priority	Specifies the priority when ARBMD (bit [63]) specifies fixed-priority mode. [35:32] = Bus master with the highest priority [31:28] = Bus master with the second highest priority [27:24] = Bus master with the third highest priority [23:20] = Bus master with the fourth highest priority [19:16] = Bus master with the fifth highest priority [15:12] = Bus master with the sixth highest priority [11:8] = Bus master with the seventh highest priority [7:4] = Bus master with the eighth highest priority [3:0] = Bus master with the ninth highest priority The 3-bit codes for different Masters are: 0000 = PCI controller 0001 = PDMAC 0010 = DMAC0 0011 = DMAC1 0100 = PCI controller 1 0101 = ATA0 0110 = ATA1 0111 = CYP 1000 = VPC controller A priority of PCIC > PDMAC > DMAC0 > DMAC1 > PCIC1 > ATA0 > ATA1 > CYP > VPC is initially set up.	0000_0001_0010_0011_0100_0101_0110_0111_1000	R/W

7.2.7. Register Address Mapping Register (RAMP) 0xE048

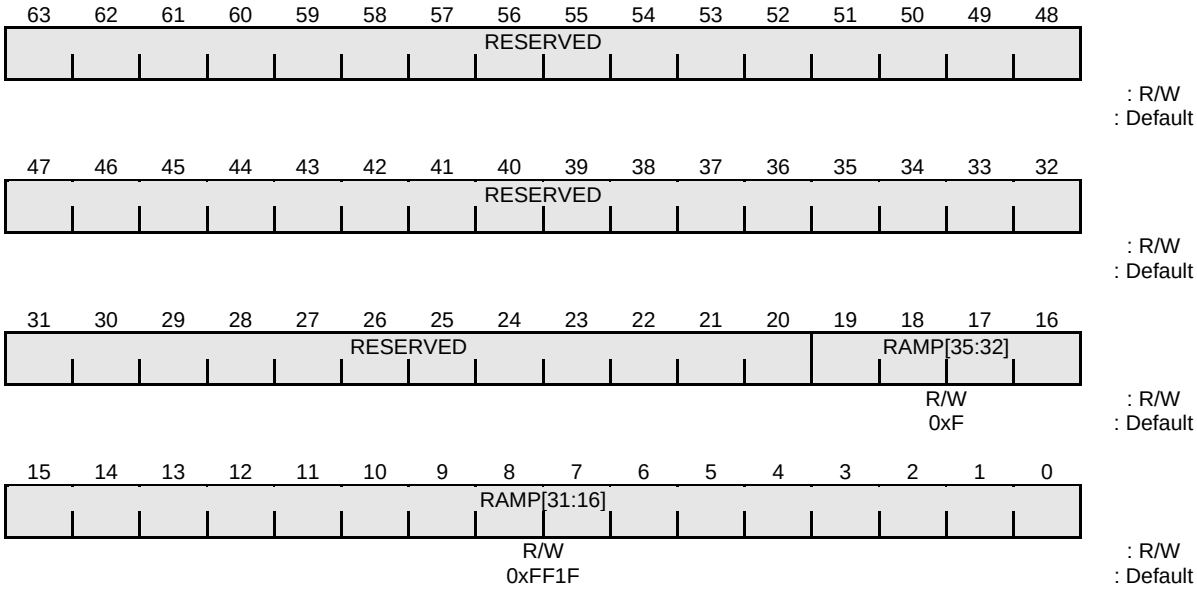


Figure 7-7 Register Address Mapping Register

Table 7-8 Register Address Mapping Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:20	—	Reserved	—	—	—
19:0	RAMP[35:16]	Register Address Mapping	This is a base address register for the TX4939 built-in registers. It holds the high-order 20 bits of a register address. The default built-in register base address is 0xF_FF1F_0000. Even after the content of the base address register is changed, the default value can be used to reference the built-in registers. (Refer to "4.2 Register Mapping".)	0xF_FF1F	R/W

7.2.8. DLL De-Skew Control Register (DSKWCTRL) 0xE060

This register controls the enabling of the locking mechanism for the PCI & DDR DLLs. This register's bits also control what kind of control signal to be used for enabling the DLL locking.

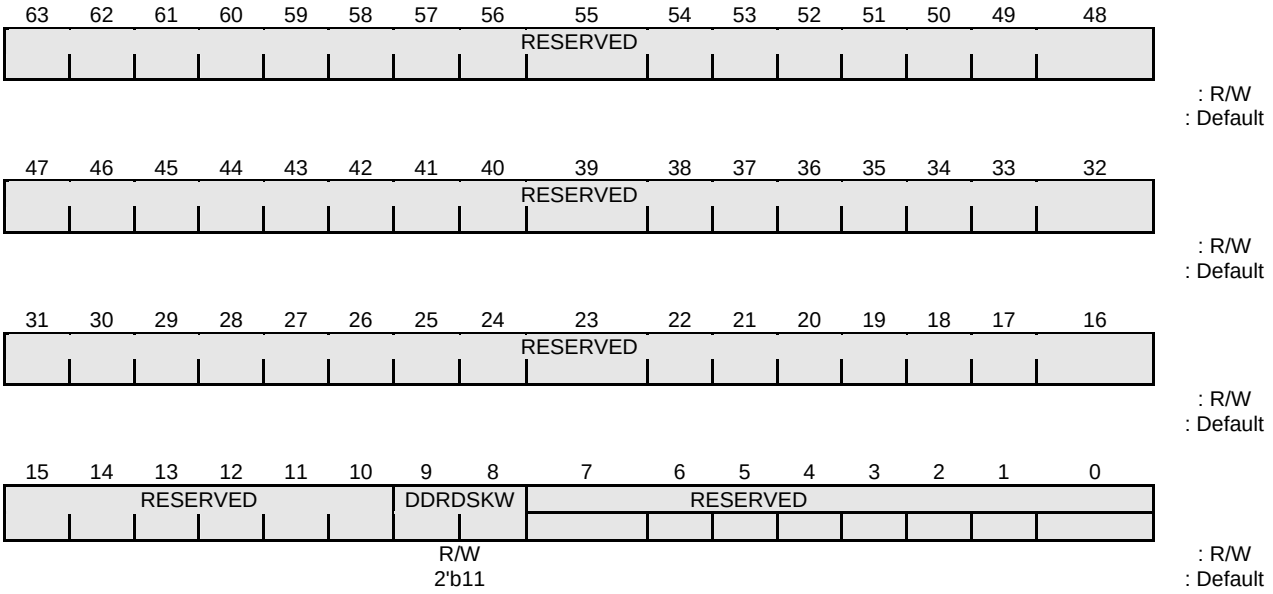


Figure 7-8 DLL De-Skew Control Register

Table 7-9 DLL De-Skew Control Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:10	—	Reserved	—	—	—
9:8	DDRDSKW	DDR DLL De-Skew Enable	Controls enabling of DDR DLL Locking. 2'b00 = Disable DLL Locking and force zero delay. 2'b01 = Reserved 2'b10 = Reserved 2'b11 = Always Enable Locking.	2'b11	R/W
7:0	—	Reserved	—	—	—

7.2.9. MCLKOSC Register 0xE068

MCLKOSC Register controls PLL set-up and configuration.

MCLKOSC	31	30	29	28	27	26	25	24
FIELD Definition	RESERVED							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
DEFAULT	0	0	0	0	0	0	0	0

MCLKOSC	23	22	21	20	19	18	17	16
FIELD Definition	RESERVED						MFRAC	MINTL
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/W
DEFAULT	0	0	0	0	0	0	1	1

MCLKOSC	15	14	13	12	11	10	9	8
FIELD Definition	RESERVED		FS [1:0]		RS [3:0]			
R/W	R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	1	0	1	0	0

MCLKOSC	7	6	5	4	3	2	1	0
FIELD Definition	IS [1:0]		C2S			ND [4:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	1	0	1	0	0	1

Figure 7-9 Definition of MCLKOSC Register

Table 7-10Bit Field Definitions of MCLKOSC Register

Bit Field	Symbol	Description
31:18	Reserved	Reserved
17	MFRAC	PLL Configuration (default value : MFRAC = 1, MINTL = 1) {MFRAC, MINTL} defines PLL Configuration of the Audio Clock Generator. MFRAC = 0, MINTL = 0: Audio Clock Generator is disabled MFRAC = 0, MINTL = 1: PLL generates 451.584 MHz with "Integer N PLL" mode (CG=20 MHz) MFRAC = 1, MINTL = 0: Reserved MFRAC = 1, MINTL = 1: PLL generates 451.584 MHz with "Fractional N PLL" mode (CG=20 MHz)
16	MINTL	
15:14	Reserved	Reserved
13:12	FS [1:0]	PLL frequency range select This parameter has a relation ship with C2S input.
11:8	RS [3:0]	PLL Low Pass Filter R2 selection R2 will be the value of 2.4 + RS[3:0] x 1.2 (K ohm)
7:6	IS [1:0]	PLL Charge Pump Current (Ip) select IS[1:0] / Description 2'b00 / Charge Pump Current will be X3 of 8 uA (24uA) 2'b01 / Charge Pump Current will be X2 of 8 uA (16uA) 2'b10 / Charge Pump Current will be X1 of 8 uA (8uA) 2'b11 / Disable Charge Pump
5	C2S	PLL VCO Performance Select
4:0	ND [4:0]	PLL Multiply factor Multiplier is the value of ND [4:0] + 1.

7.2.10. MCLKCTL Register 0xE070

MCLKCTL	31	30	29	28	27	26	25	24
FIELD Definition	RESERVED							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
DEFAULT	0	0	0	0	0	0	0	0

MCLKCTL	23	22	21	20	19	18	17	16
FIELD Definition	RESERVED							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
DEFAULT	0	0	0	0	0	0	0	0

MCLKCTL	15	14	13	12	11	10	9	8
FIELD Definition	BD	RESERVED						
R/W	R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
DEFAULT	0	0	0	0	0	0	0	0

MCLKCTL	7	6	5	4	3	2	1	0
FIELD Definition	CB	CHB				CHA		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Figure 7-10 Definition of MCLKCTL Register

Table 7-11 Bit Field Definitions

	Symbol	Description
31:16	N/A	Reserved
15	BDE	Baud Clock enable (1:Enable, 0:Disable)
14:8	N/A	Reserved
7	CBE	Channel-B Enable (1:Enable, 0:Disable)
6:4	CHB [2:0]	Channel B Frequency select CHB [2:0] = 3'b000: Select 22.5792 MHz (Fs = 44.1 KHz) CHB [2:0] = 3'b001: Select 24.576 MHz (Fs = 48.0 KHz) CHB [2:0] = 3'b010: Select 16.384 MHz (Fs = 32.0 KHz) CHB [2:0] = 3'b011: Select 12.288 MHz (Fs = 24.0 KHz) CHB [2:0] = 3'b100: Select 8.192 MHz (Fs = 16.0 KHz) CHB [2:0] = 3'b101: Select 4.096 MHz (Fs = 8.0 KHz) CHB [2:0] = 3'b110: Select 45.1584 MHz (Fs = 88.2 KHz) CHB [2:0] = 3'b111: Select 49.152 MHz (Fs = 96.0 KHz)
3	CAE	Channel-A Enable (1:Enable, 0:Disable)
2:0	CHA [2:0]	Channel A Frequency select CHA [2:0] = 3'b000: Select 22.5792 MHz (Fs = 44.1 KHz) CHA [2:0] = 3'b001: Select 24.576 MHz (Fs = 48.0 KHz) CHA [2:0] = 3'b010: Select 16.384 MHz (Fs = 32.0 KHz) CHA [2:0] = 3'b011: Select 12.288 MHz (Fs = 24.0 KHz) CHA [2:0] = 3'b100: Select 8.192 MHz (Fs = 16.0 KHz) CHA [2:0] = 3'b101: Select 4.096 MHz (Fs = 8.0 KHz) CHA [2:0] = 3'b110: Select 45.1584 MHz (Fs = 88.2 KHz) CHA [2:0] = 3'b111: Select 49.152 MHz (Fs = 96.0 KHz)

7.2.11. GPIO Mode Register 1(GPIOMR1) 0xE100

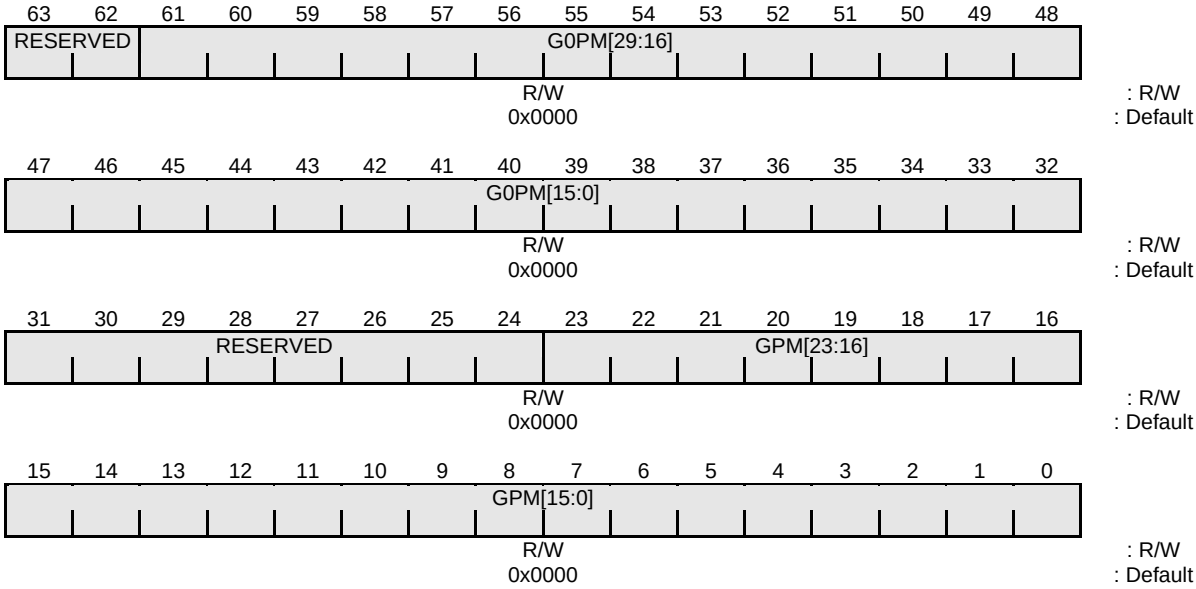


Figure 7-11 GPIOMR1 Register

Table 7-12 GPIOMR1 Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:62	G0PM	G0P Mode	RESERVED	0x0	R/W
61:32	G0PM	G0P Mode	G0PIO29-G0PIO00 Mode select 0: Tri-state mode (G0PD=1), Low output (G0PD=0) 1: output mode Please refer to GPIODR1 register	0x0000_0000	R/W
31:24	GPM	GP Mode	RESERVED	0x00	R/W
23:0	GPM	GP Mode	GPIO23-GPIO00 Mode select 0: Tri-state mode (GPD=1), Low output (GPD=0) 1: output mode Please refer to GPIODR1 register	0x00_0000	R/W

7.2.12. GPIO Data Register 1 (GPIODR1) 0xE108

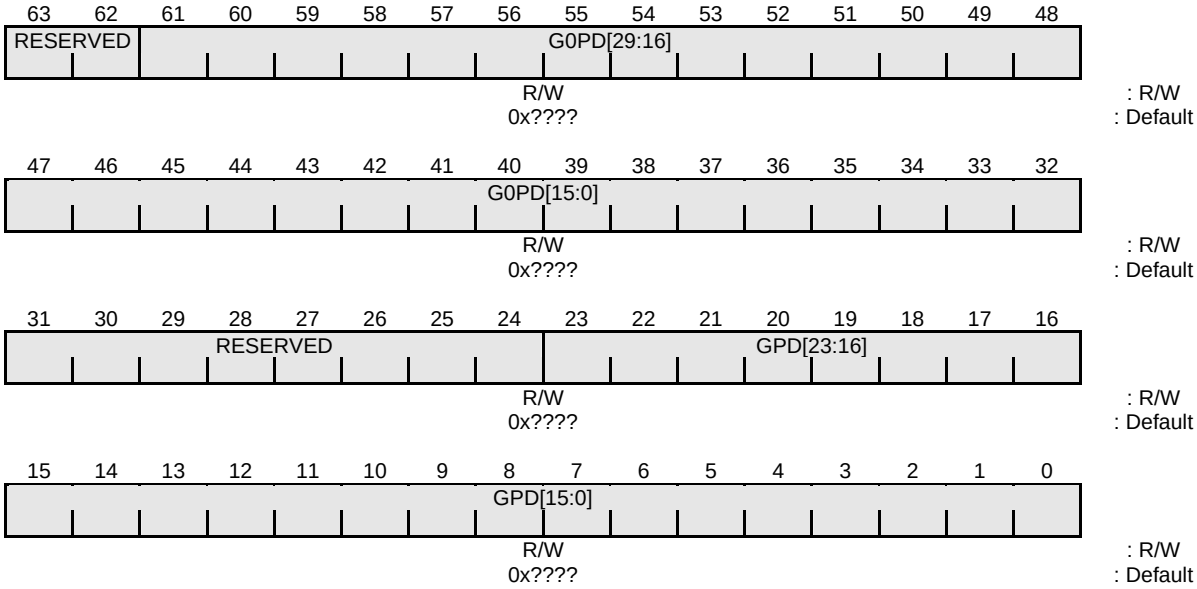


Figure 7-12 GPIODR1 Register

Table 7-13 GPIODR1 Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:62	G0PD	G0P Data	Reserved	0x?	R/W
61:32	G0PD	G0P Data	G0PIO29 – G0PIO00 Data/Status When Write to these bits: - If G0PM = 0 then if G0PD = 0, GPIOC drives LOW if G0PD = 1, GPIOC tri-state (default) - if G0PM = 1 then if G0PD = 0, GPIOC drives LOW if G0PD = 1, GPIOC drives HIGH (default) When read these bits, the read data are from the G0PIO signals.	0x????_????	R/W
31:24	GPD	GP Data	Reserved	0x??	R/W
23:0	GPD	GP Data	GPIO23 – GPIO00 Data/Status When Write to these bits: - If GPM = 0 then if GPD = 0, GPIOC drives LOW if GPD = 1, GPIOC tri-state (default) - if GPM = 1 then if GPD = 0, GPIOC drives LOW if GPD = 1, GPIOC drives HIGH (default) When read these bits, the read data are from the GPIO signals.	0x??_????	R/W

7.2.13. GPIO Mode Register 2(GPIOMR2) 0xE110

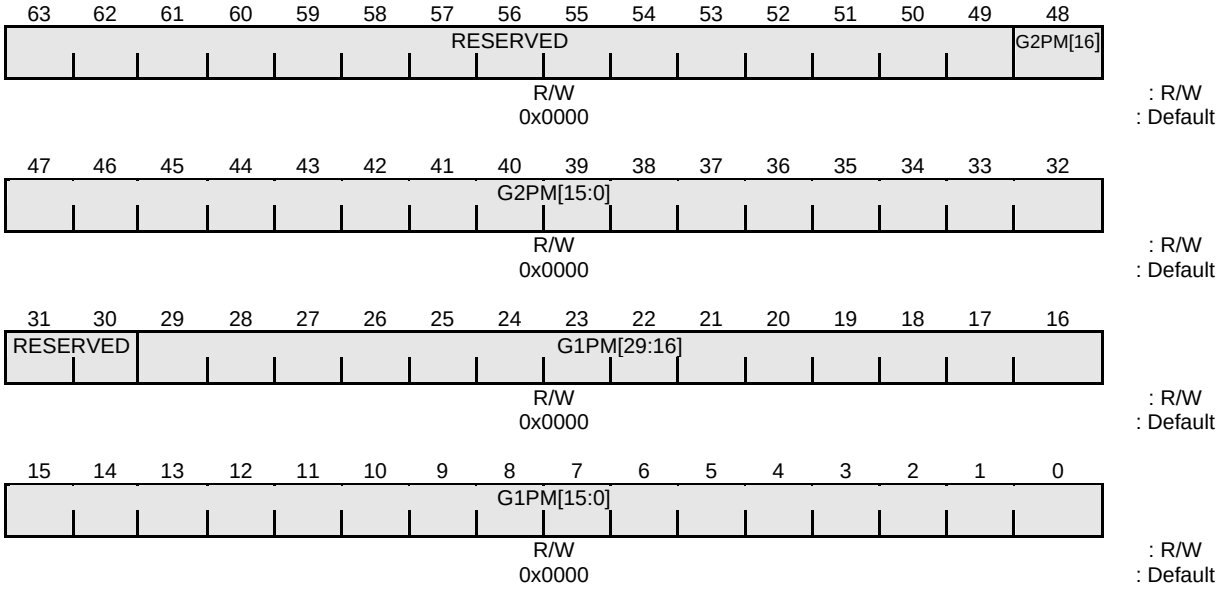


Figure 7-13 GPIOMR2 Register

Table 7-14 GPIOMR2 Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:49	G2PM	G2P Mode	Reserved	0x0000_0000	R/W
48:32	G2PM	G2P Mode	G2PIO16-G2PIO00 Mode select 0: Tri-state mode (G2PD=1), Low output (G2PD=0) 1: output mode Please refer to GPIODR2 register	0x0_0000	R/W
31:30	G1PM	G1P Mode	Reserved	0x0	R/W
29:0	G1PM	G1P Mode	G1PIO29-G1PIO00 Mode select 0: Tri-state mode (G1PD=1), Low output (G1PD=0) 1: output mode Please refer to GPIODR2 register	0x0000_0000	R/W

7.2.14. GPIO Data Register 2 (GPIODR2) 0xE118

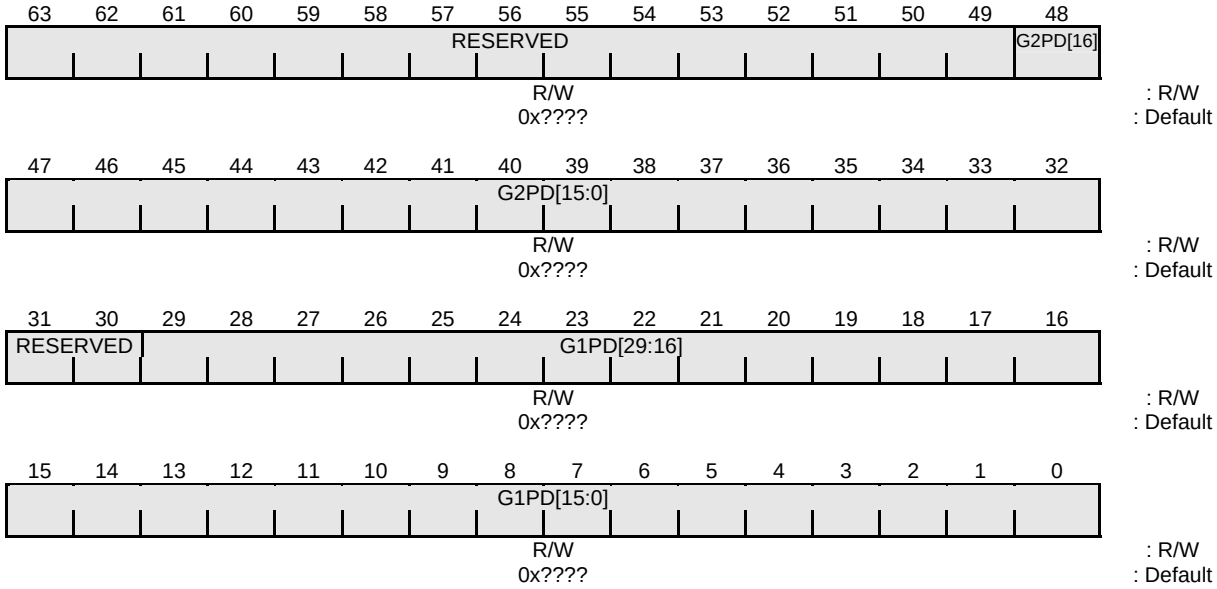


Figure 7-14 GPIODR2 Register

Table 7-15 GPIODR2 Register

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:49	G2PD	G2P Data	Reserved	0x????_????	R/W
48:32	G2PD	G2P Data	G2PIO16 – G2PIO00 Data/Status When Write to these bits: - If G2PM = 0 then if G2PD = 0, GPIOC drives LOW if G2PD = 1, GPIOC tri-state (default) - if G2PM = 1 then if G2PD = 0, GPIOC drives LOW if G2PD = 1, GPIOC drives HIGH (default) When read these bits, the read data are from the G2PIO signals.	0x?_????	R/W
31:30	G1PD	G1P Data	Reserved	0x?	R/W
29:0	G1PD	G1P Data	G1PIO29 – G1PIO00 Data/Status When Write to these bits: - If G1PM = 0 then if G1PD = 0, GPIOC drives LOW if G1PD = 1, GPIOC tri-state (default) - if G1PM = 1 then if G1PD = 0, GPIOC drives LOW if G1PD = 1, GPIOC drives HIGH (default) When read these bits, the read data are from the GPIO signals.	0x????_????	R/W

Chapter 8. Interrupt Controller

8.1. Characteristics

The TX4939 on-chip Interrupt Controller (IRC) receives interrupt requests from the TX4939 on-chip peripheral circuitry as well as external interrupt requests and in turn generates interrupt requests to the TX49/H4 processor core.

Also, this Interrupt Controller has a 16-bit flag register that generates interrupt requests either to the external devices or to the TX49/H4 core.

The Interrupt Controller has the following characteristics.

- *Supports interrupts from 47 types of on-chip peripheral circuits and a maximum of 3 external interrupt signal inputs.*
- *Sets 8 priority interrupt levels for each interrupt input.*
- *Can select either edge detection or level detection for each external interrupt when in the interrupt detection mode.*
- *The Interrupt Controller contains a 16-bit readable/write-able flag register which can be programmed to issue interrupt requests to external devices as well as to the TX49/H4 core (IRC interrupt).*

8.2. Block Diagram

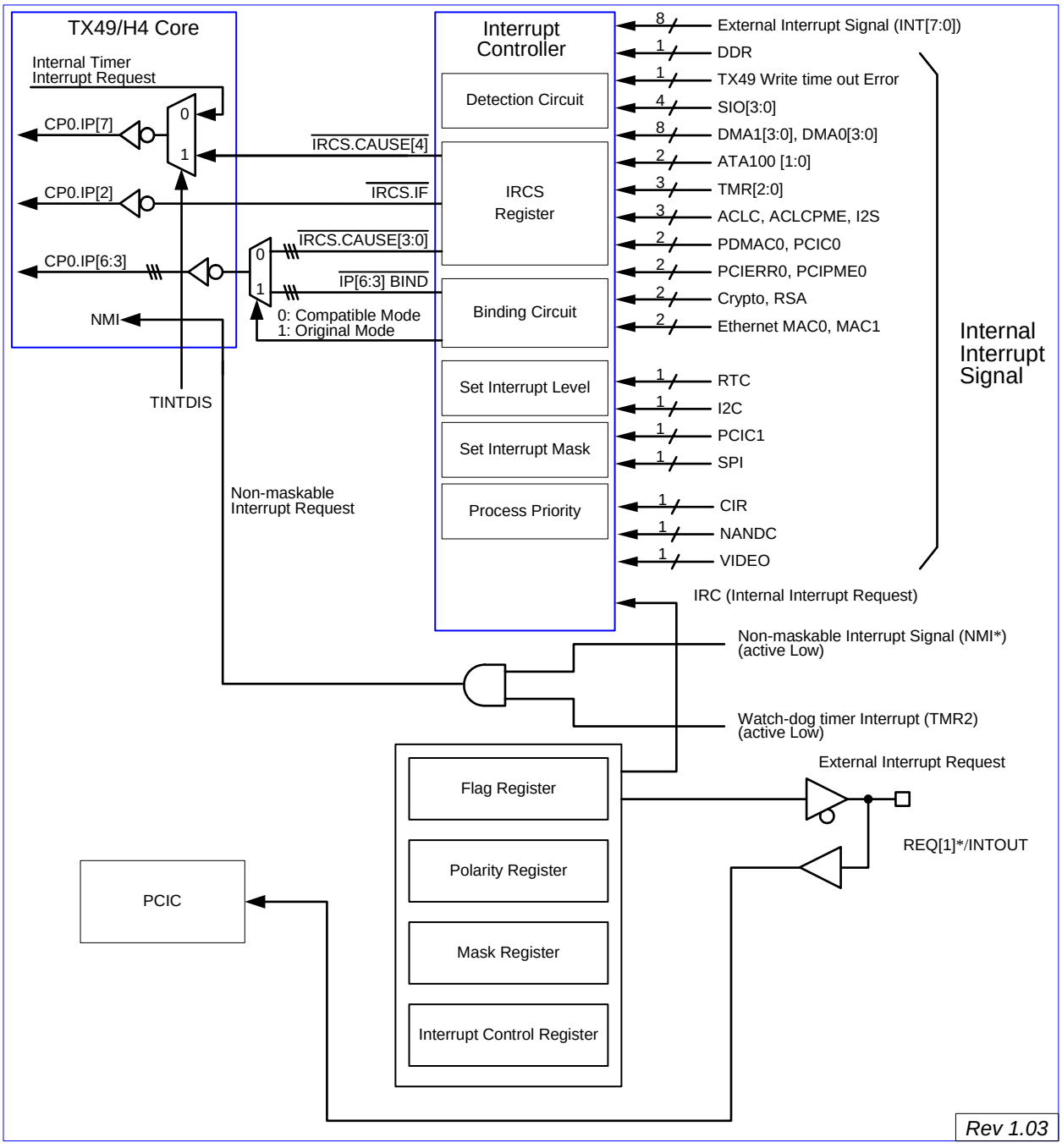


Figure 8-1 Interrupt Controller Outline

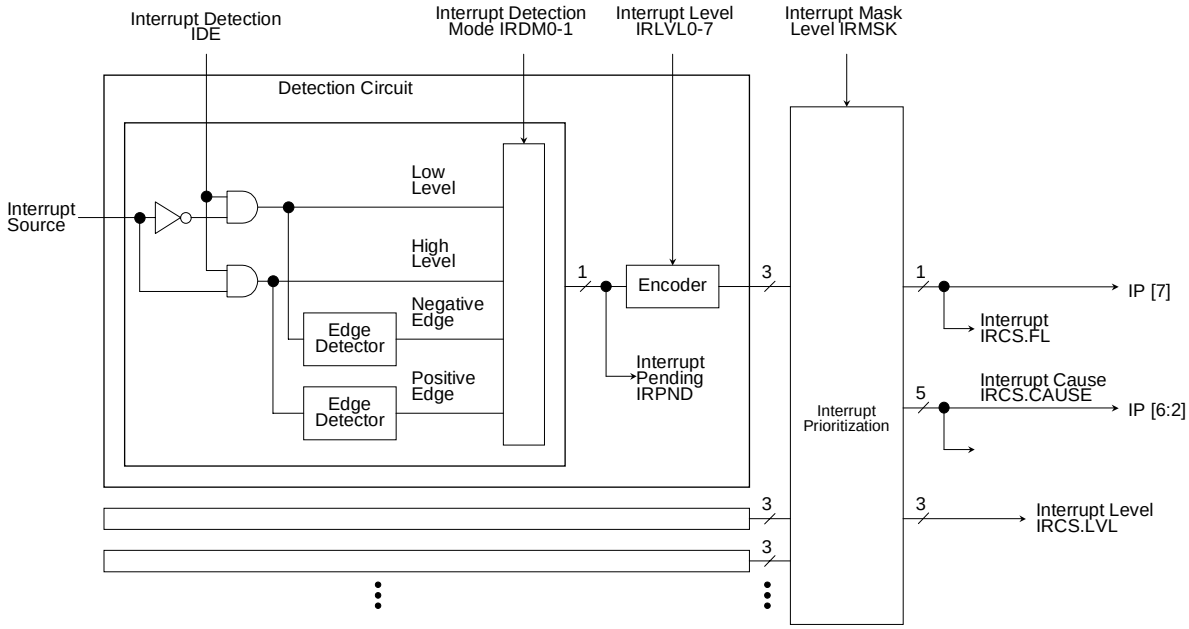


Figure 8-2 Internal Block Diagram of Interrupt Controller

8.3. Detailed Explanation

8.3.1. Interrupt Sources

The TX4939 has 47 interrupt sources, such as interrupts from 44 types of on-chip peripheral circuits and 3 external interrupt signals. Table 8-1 lists the interrupt sources and its interrupt numbers in the TX4939 system.

The priority encoder nominates highest priority interrupt and generates a 6-bit binary number corresponding to the interrupt number in Table 8-1. This number always appears in the field of IRCS register. (i.e. IRCS.CAUSE)

The way to inform interrupt occurrence to CPU has variations depend on the interrupt operation mode.

8.3.2. Interrupt Operation Mode

TX4939 interrupt controller has two modes of operation. One is former TX49 Series compatible mode and TX4939 original mode. These mode can be selected by setting the flag bit of “Interrupt Source and Cause IP Binding Register”. (See 8.4.2)

8.3.3. Compatible Mode

In this mode, the interrupt occurrence will activate CP0 IP[2] bit only. This feature is upward compatible with former TX49 Series like TX4938. The interrupt handler is supposed to read IRCS.CAUSE register to get the interrupt number.

Addition to this feature, selected interrupt occurrences can be directly informed to CPU by means of CP0 IP[6:3] bit. The assignment of interrupt source to designated IP bit can be performed by ISCIPB register.
(See 8.4.2 for detail explanation)

8.3.4. Original Mode

In this mode, the interrupt occurrence will activate CP0 IP[2] bit and CP0 IP[7:3], 5-bit will receive the interrupt number as follows:

If IRCS.CAUSE < 31, CP0 IP[7:3] <= (IRCS.CAUSE)
If IRCS.CAUSE > 31, CP0 IP[7:3] <= 31
Note IRCS.CAUSE value never takes 31 by design.

This structure created by simply expand TX4938 interrupt controller schema, which schema is only meaningful only when the internal timer is not used.

To detect the occurrence of interrupt by checking CP0 IP[2] and then, the interrupt handler can get the interrupt source number directly from CP0 IP[7:3] And if this register value is 31, interrupt handler is supposed to get exact interrupt source number from IRCS.CAUSE register.

In addition to above interrupt sources, the TX49/H4 core has a TX49/H4 core internal timer interrupt and two software interrupts, but these interrupts are directly reported to the TX49/H4 core independent of this Interrupt Controller. Please refer to the 64-bit TX System RISC TX49/H4 Core Architecture Manual for more information.

Table 8-1 TX4939 Interrupt Sources

Interrupt Number	Interrupt Source Defined Number	Detect Control	IRCS.CAUSE	Original Mode		Compatible Mode IP[7:2] (Note 1)
				IP[7:3]	IP[2]	
0	No Interrupt	-	0	0	0	6'b000000
1	DDR	Low Active	1	1	1	6'b0SSSS1
2	Tx49 Write Timeout	Low Active	2	2	1	6'b0SSSS1
3	INT0	Edge/Level	3	3	1	6'b0SSSS1
4	INT1	Edge/Level	4	4	1	6'b0SSSS1
5	INT2	Edge/Level	5	5	1	6'b0SSSS1
6	Ether0	Low Active	6	6	1	6'b0SSSS1
7	VIDEO	Low Active	7	7	1	6'b0SSSS1
8	CIR	Low Active	8	8	1	6'b0SSSS1
9	SIO0	Low Active	9	9	1	6'b0SSSS1
10	DMA0[0]	Low Active	10	10	1	6'b0SSSS1
11	DMA0[1]	Low Active	11	11	1	6'b0SSSS1
12	DMA0[2]	Low Active	12	12	1	6'b0SSSS1
13	DMA0[3]	Low Active	13	13	1	6'b0SSSS1
14	IRC	Low Active	14	14	1	6'b0SSSS1
15	PDMAC0	Low Active	15	15	1	6'b0SSSS1
16	TMR0	Low Active	16	16	1	6'b0SSSS1
17	TMR1	Low Active	17	17	1	6'b0SSSS1
18	TMR2	Low Active	18	18	1	6'b0SSSS1
19	ATA0	Low Active	19	19	1	6'b0SSSS1
20	ATA1	Low Active	20	20	1	6'b0SSSS1
21	ACLC	Low Active	21	21	1	6'b0SSSS1
22	DMA1[0]	Low Active	22	22	1	6'b0SSSS1
23	DMA1[1]	Low Active	23	23	1	6'b0SSSS1
24	DMA1[2]	Low Active	24	24	1	6'b0SSSS1
25	DMA1[3]	Low Active	25	25	1	6'b0SSSS1
26	Cipher	Low Active	26	26	1	6'b0SSSS1
27	INTA (Note 2)	Edge/Level	27	27	1	6'b0SSSS1
28	INTB (Note 2)	Edge/Level	28	28	1	6'b0SSSS1
29	INTC (Note 2)	Edge/Level	29	29	1	6'b0SSSS1
30	INTD (Note 2)	Edge/Level	30	30	1	6'b0SSSS1
31	## Not For Use	Low Active	####	31	1	6'b0SSSS1
32	## Reserved	Low Active	32	31	1	6'b0SSSS1
33	I2C	Low Active	33	31	1	6'b0SSSS1
34	SPI	Low Active	34	31	1	6'b0SSSS1
35	## Reserved	Low Active	35	31	1	6'b0SSSS1
36	PCIC1(PCI1ERR,PCI1PME)	Low Active	36	31	1	6'b0SSSS1
37	PCI0ERR	Low Active	37	31	1	6'b0SSSS1
38	PCI0PME	Low Active	38	31	1	6'b0SSSS1
39	NANDC	Low Active	39	31	1	6'b0SSSS1
40	ACLC_PME	Low Active	40	31	1	6'b0SSSS1
41	RTC	Low Active	41	31	1	6'b0SSSS1
42	RSA/RND	Low Active	42	31	1	6'b0SSSS1
43	Ether1	Low Active	43	31	1	6'b0SSSS1
44	SIO1	Low Active	44	31	1	6'b0SSSS1
45	SIO2	Low Active	45	31	1	6'b0SSSS1
46	SIO3	Low Active	46	31	1	6'b0SSSS1
47	I2S	Low Active	47	31	1	6'b0SSSS1
48	TMR3	Low Active	48	31	1	6'b0SSSS1
49	TMR4	Low Active	49	31	1	6'b0SSSS1
50	TMR5	Low Active	50	31	1	6'b0SSSS1

Note 1: Value of S in the expression 6'b0SSSS1 represents the selected interrupt source. In case IP[2] is "1", one or more than one of SSSS should be "1". (See 8.4.2)

Note 2: If these four interrupt signal are used for PCI, all of them should be programmed as Level detect mode. Edge detect mode is provided as general purpose interrupt control.

8.3.5. Interrupt Request Detection

In order to perform interrupt detection, each register of the Interrupt Controller is initialized, and then the IDE bit of the Interrupt Detection Enable Register (IRDEN) is set to “1.” All interrupts detected by the Interrupt Controller are masked when this bit is cleared.
It is possible to set each interrupt factor detection mode using Interrupt Detection Mode Register 0, 1, and 2 (IRDM0, IRDM1, and IRDM2). There are four detection modes: Low level, High level, falling edge, and rising edge.
The detected interrupt factors can be read out from the Interrupt Pending Register (IRPND) unless the IDE bit of IRDEN is not cleared.

8.3.6. Interrupt Level Assigning

Interrupt levels from 0 to 7 are assigned to each detected interrupt using the Interrupt Level Register (IRLVL0-7). Interrupt level 7 is the highest priority and interrupt level1 is the lowest priority. Level 0 interrupts will be masked (Table 8-2).
The priorities set by these interrupt levels will be given higher priority than the priorities provided for each interrupt source indicated in Table 8-1.

Table 8-2 Interrupt Levels

Priority	Interrupt Level (IRLVLn.ILm)
High	111
	110
	101
	100
	011
	010
Low	001
Mask	000

8.3.7. Interrupt Priority Assigning

When multiple interrupt requests exist, the Interrupt Controller selects the interrupt with the highest priority according to the priority level and interrupts number. Interrupt factors with an interrupt level lower than the interrupt level specified by the Interrupt Mask Level Register (IRMSK) will be excluded (masked).
When the interrupt with the highest priority is selected, then the interrupt number of that interrupt is set in the interrupt factor field (CAUSE) of the Interrupt Current Status Register (IRCS), the interrupt level is set in the Interrupt Level field (LVL), and the Interrupt Flag bit (IF) is set.
Priorities are assigned as follows.

- *When interrupt levels differ, the interrupt with the higher interrupt level has priority (Table 8-2)*
- *When multiple interrupts with the same interrupt level are simultaneously detected, the interrupt with the smaller interrupt number has priority.*

In addition, the interrupt priority assignments are re-evaluated under the following conditions. At this time, the interrupt with the highest priority is selected and the Interrupt Factor field (CAUSE) and Interrupt Level field (LVL) of the Interrupt Current Status Register (IRCS) are set again.

- *When an interrupt request with a higher interrupt level than that of the currently selected interrupt is detected. However, when the interrupt levels are equal, the Interrupt Level field (LVL) does not change even if the interrupt number is small.*
- *When the interrupt level (IRLVLn.ILm) of the currently selected interrupt changes to a value smaller than the current setting.*
- *When the currently selected interrupt is cleared (refer to 8.3.10 Clearing Interrupt Requests).*

8.3.8. Interrupt Notification of Original Mode

When the interrupt with the highest priority is selected, then the interrupt factor is reported to the Interrupt Current Status Register (IRCS) and an interrupt is reported to the TX49/H4 core.

Any interrupt occurrence will assert CP0.IP[2] bit. When IP[2] has value one IP[7:3] will show the interrupt source number described in Table 8-3 column of IP[7:3]. This value has meaning only when the internal timer interrupt is disabled. Since IP[7:3] is 5 digit binary, the interrupt that number is more than 31 will appear as 31. In this case, interrupt handler should refer the content of IRCS.CAUSE register.

The internal timer interrupt can be disable by setting TINTDIS. This value can be set at the boot time also.
See more detail “Chapter 5 Boot Configuration”.

Table 8-3 Interrupt Notification to IP[7:2] of the CP0 Cause Register

TINTDIS	IP[7]	IP[6:3]	IP[2]
0 : Internal Timer Interrupts: Valid	Internal Timer Interrupt Notification	If IRCS.CAUSE < 31, CP0 IP[6:3] <= 0x0F && (IRCS.CAUSE) If IRCS.CAUSE > 31, CP0 IP[6:3] <= 15 LOGICALLY, THIS DOES NOT HAVE MUCH MEANING.	Indicate occurrence of any interrupt
1 : Internal Timer Interrupts: Invalid	If IRCS.CAUSE < 31, CP0 IP[7:3] <= (IRCS.CAUSE) If IRCS.CAUSE > 31, CP0 IP[7:3] <= 31 Note: IRCS.CAUSE value never takes 31 by design.		Indicate occurrence of any interrupt

8.3.9. Interrupt Notification of Compatible Mode

Any interrupt occurrence will be informed to CP0 IP[2] same as former TX49 Series, i.e. TX4938.
However, the usage of IP[7:3] has been changed as follows.

IP[7] will be use for Internal timer interrupt exclusively.
IP[6:3] will not get the part of IRCS.CAUSE register instead it will used for selected interrupt indication. (See more detail in 8.4.2 Interrupt Source and Cause IP Binding Register (ISCIPB))

Table 8-4 Interrupt Notification to IP[7:2] of the CP0 Cause Register

TINTDIS	IP[7]	IP[6:3]	IP[2]
0 : Internal Timer Interrupts: Valid	Internal Timer Interrupt Notification	Indicate occurrence of selected interrupt	Indicate occurrence of any interrupt
1 : Internal Timer Interrupts: Invalid	No Internal Timer Interrupt Notification	Indicate occurrence of selected interrupt	Indicate occurrence of any interrupt

8.3.10. Clearing Interrupt Requests

Interrupt requests are cleared according to the following process.

- *When the detection mode is set to the High level or Low level:*
  - *Operation is performed to de-assert the request of a source that is asserting an interrupt request.*

- *When the detection mode is set to Rising edge or Falling edge*
  - *Edge detection requests are cleared by first specifying the interrupt source of the interrupt request to be cleared in the Edge Detection Clear Source field (EDCS0 or EDCS1) of the Interrupt Edge Detection Clear Register (IREDC) then writing the resulting value when the corresponding Edge Detection Clear Enable bit (EDCE0 or EDCE1) is set to “1.”*

8.3.11. Interrupt requests

It is possible to make interrupt requests to external devices and interrupt requests (IRC interrupts) to the TX49/H4 core by using a 16-bit interrupt request flag register. REQ[1]* signals are used as interrupt output signals. Consequently, external interrupt requests can only be used when in the PCI External Arbiter mode. Also, internal interrupt requests are assigned to interrupt number 14 of the Interrupt Controller (IRC).
The following six registers set the interrupts.

- *Interrupt Request Flag Register (IRFLAG0, IRFLAG1)*
- *Interrupt Request Polarity Control Register (IRPOL)*
- *Interrupt Request Mask Register (IRMASKINT, IRMASKEXT)*
- *Interrupt Request Control Register (IRRCNT)*

The following formulas derive the interrupt generation conditions:

Internal interrupt request =
(|((IRFLAG[15:0] ^ IRPOL[15:0]) & IRMASKINT[15:0]))^ IRRCNT.INTPOL

External interrupt request =
(|((IRFLAG[15:0] ^ IRPOL[15:0]) & IRMASKEXT[15:0]))^ IRRCNT.EXTPOL

In the above formulas, "^" indicates Exclusive OR operations and "|" indicates reduction operators that perform an OR operation on all bits.
Also, the External Interrupt OD Control bit (IRRCNT.OD) of the Interrupt Request Control Register can select whether the external interrupt supply signal is open drain output or totem pole output.

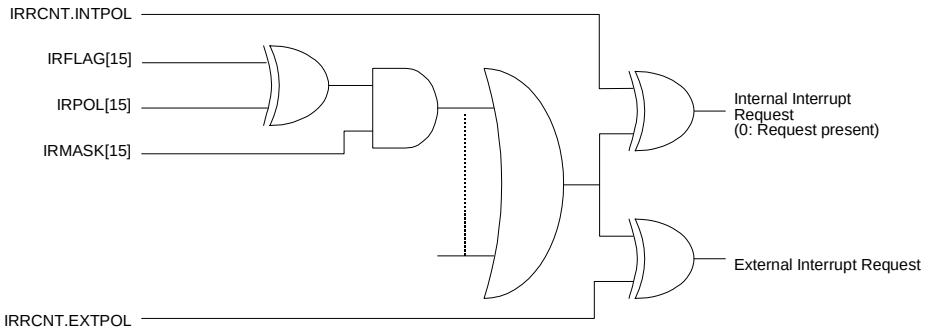


Figure 8-3 External Interrupt Request Logic

There are two flag registers: Flag Register 0 (IRFLAG0), and Flag Register 1 (IRFLAG1). These registers have two different Write methods. Accordingly, Writes to one register are reflects in the other.

Either "0" or "1" can be written to Flag Register 0

In the case of Flag Register 1 however, "1" can be written from the TX49/H4 core, but "0" cannot be written. On the other hand, bits that wrote "1" are cleared to "0" in the case of access from a device other than the TX49/H4 core (access from an external PCI device for example). The bit value at this time will not change even if "0" is written. This register sends interrupt notification from the TX49/H4 core to external devices. External devices can be used in applications that clear these interrupt notifications.

8.4. Registers

This register set has been moved from IMBUS (32-bit) to GBUS (64-bit) to decrease the latency of register access. Because of this change from former TX49 series, register size is remained as 32-bit, but the address alignment changed to 64-bit. As registers are attached lower 32-bit portion in GBUS, this address expressions are that of Little Endian Mode.

Table 8-5 Interrupt Control Registers

Address	Register	Register Name
0xE800	IRDEN	Interrupt Detection Enable Register
0xE808	ISCIPB	Interrupt Source and Cause Binding Register
0xE810	IRDM0	Interrupt Detection Mode Register 0
0xE818	IRDM1	Interrupt Detection Mode Register 1
0xE820	IRLVL0	Interrupt Level Register 0
0xE828	IRLVL1	Interrupt Level Register 1
0xE830	IRLVL2	Interrupt Level Register 2
0xE838	IRLVL3	Interrupt Level Register 3
0xE840	IRLVL4	Interrupt Level Register 4
0xE848	IRLVL5	Interrupt Level Register 5
0xE850	IRLVL6	Interrupt Level Register 6
0xE858	IRLVL7	Interrupt Level Register 7
0xE860	IRLVL8	Interrupt Level Register 8
0xE868	IRLVL9	Interrupt Level Register 9
0xE870	IRLVL10	Interrupt Level Register 10
0xE878	IRLVL11	Interrupt Level Register 11
0xE880	IRLVL12	Interrupt Level Register 12
0xE888	IRLVL13	Interrupt Level Register 13
0xE890	IRLVL14	Interrupt Level Register 14
0xE898	IRLVL15	Interrupt Level Register 15
0xE8A0	IRMSK	Interrupt Mask Register
0xE8A8	IREDC	Interrupt Edge Detection Clear Register
0xE8B0	IRPND0	Interrupt Pending Register 0
0xE8B8	IRCS	Interrupt Current Status Register
0xE8C0	IRPND1	Interrupt Pending Register 1
0xE8C8	IRDM2	Interrupt Detection Mode Register 2
0xE8D0	IRDM3	Interrupt Detection Mode Register 3
0xE8D8	IRDBR0	Interrupt Debug Register 0
0xE8E0	IRDBR1	Interrupt Debug Register 1
0xE8E8	IRDBEN	Interrupt Debug Enable Register
0xE900	IRFLAG0	Interrupt Request Flag Register 0
0xE908	IRFLAG1	Interrupt Request Flag Register 1
0xE910	IRPOL	Interrupt Request Polarity Control Register
0xE918	IRRCNT	Interrupt Request Control Register
0xE920	IRMASKINT	Interrupt Request Internal Interrupt Mask Register
0xE928	IRMASKEXT	Interrupt Request External Interrupt Mask Register

8.4.1. Interrupt Detection Enable Register (IRDEN) 0xE800

Figure 8-4 Interrupt Detection Enable Register

Table 8-6 Interrupt Detection Enable Register

Bit(s)	Mnemonic	Field Name	Explanation	Read/Write
31:1	—	—	Reserved	—
0	IDE	Interrupt Detection Enable	Interrupt Detection Enable (Default: 0) Enables interrupt detection. 0: Stop interrupts detection. 1: Start interrupts detection.	R/W

8.4.2. Interrupt Source and Cause IP Binding Register (ISCIPB) 0xE808

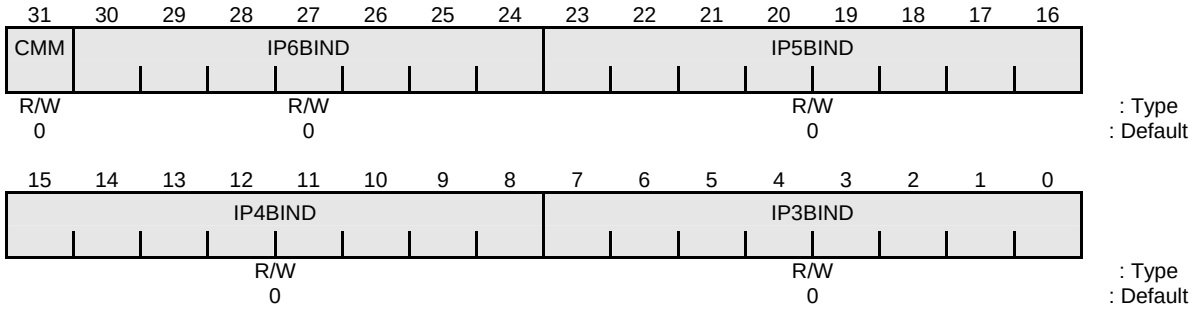


Figure 8-5 Interrupt Source and Cause IP Binding Register

Table 8-7 Interrupt Source and Cause IP Binding Register

Bit(s)	Mnemonic	Field Name	Explanation	R/W
31	CMM	Compatible Mode	Select Interrupt Controller Mode 0: Original Mode (Default) 1: Compatible Mode.	R/W
30:24	IP6BIND	IP[6] BIND	IP[6] interrupt source bind This bit has an effect only in the “Compatible Mode”, which is activated by setting “1” in CMM bit in this register. In Compatible Mode, when the interrupt source N gets a request, the request also propagates to CP0 IP[6] bit same time. Note: Interrupt source number is defined in Table 8-1 0: No Assignment (Default) N : Interrupt Source N (N>0) assigned this IP bit	R/W
23:16	IP5BIND	IP[5] BIND	IP[5] interrupt source bind (Ditto)	R/W
15:8	IP4BIND	IP[4] BIND	IP[4] interrupt source bind (Ditto)	R/W
7:0	IP3BIND	IP[3] BIND	IP[3] interrupt source bind (Ditto)	R/W

8.4.3. Interrupt Detection Mode Register 0 (IRDM0) 0xE810

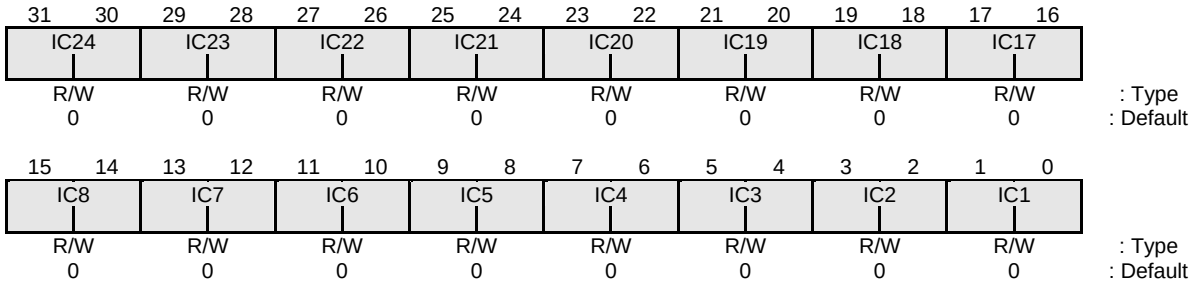


Figure 8-6 Interrupt Detection Mode Register 0

Table 8-8 Interrupt Detection Mode Register 0

Bits	Mnemonic	Field Name	Explanation	R/W
31:30	IC24	Interrupt Source Control 24	Interrupt Source Control 24 (Default: 00) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
29:28	IC23	Interrupt Source Control 23	Interrupt Source Control 23 (Default: 00) Ditto	R/W
27:26	IC22	Interrupt Source Control 22	Interrupt Source Control 22 (Default: 00) Ditto	R/W
25:24	IC21	Interrupt Source Control 21	Interrupt Source Control 21 (Default: 00) Ditto	R/W
23:22	IC20	Interrupt Source Control 20	Interrupt Source Control 20 (Default: 00) Ditto	R/W
21:20	IC19	Interrupt Source Control 19	Interrupt Source Control 19 (Default: 00) Ditto	R/W
19:18	IC18	Interrupt Source Control 18	Interrupt Source Control 18 (Default: 00) Ditto	R/W
17:16	IC17	Interrupt Source Control 17	Interrupt Source Control 17 (Default: 00) Ditto	R/W
15:14	IC8	Interrupt Source Control 8	Interrupt Source Control 8 (Default: 00) Ditto	R/W
13:12	IC7	Interrupt Source Control 7	Interrupt Source Control 7 (Default: 00) Ditto	R/W
11:10	IC6	Interrupt Source Control 6	Interrupt Source Control 6 (Default: 00) Ditto	R/W
9:8	IC5	Interrupt Source Control 5	Interrupt Source Control 5 (Default: 00) 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
7:6	IC4	Interrupt Source Control 4	Interrupt Source Control 4 (Default: 00) Ditto	R/W
5:4	IC3	Interrupt Source Control 3	Interrupt Source Control 3 (Default: 00) Ditto	R/W
3:2	IC2	Interrupt Source Control 2	Interrupt Source Control 2 (Default: 00) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
1:0	IC1	Interrupt Source Control 1	Interrupt Source Control 1 (Default: 00) Ditto	R/W

8.4.4. Interrupt Detection Mode Register 1 (IRDM1) 0xE818

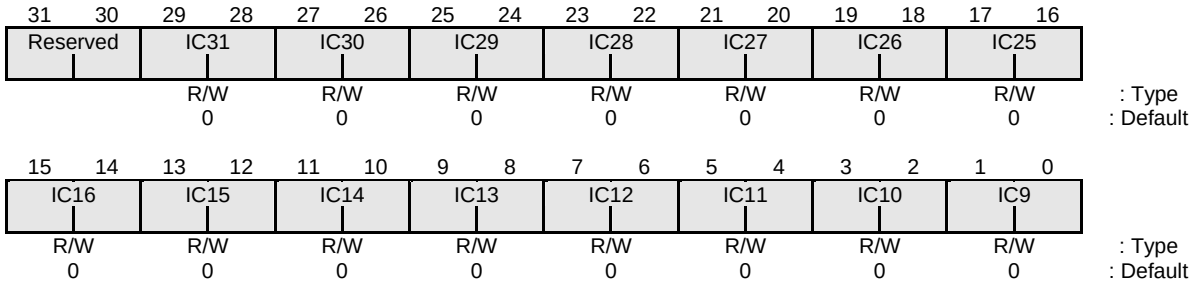


Figure 8-7 Interrupt Detection Mode Register 1

Table 8-9 Interrupt Detection Mode Register 1

Bit	Mnemonic	Field Name	Explanation	Read/Write
31:30	—	—	Reserved	—
29:28	IC31	Interrupt Source Control 31	Interrupt Source Control 31 (Default: 00, R/W) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
27:26	IC30	Interrupt Source Control 30	Interrupt Source Control 30 (Default: 00) 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active	R/W
25:24	IC29	Interrupt Source Control 29	Interrupt Source Control 29 (Default: 00, R/W) Ditto	R/W
23:22	IC28	Interrupt Source Control 28	Interrupt Source Control 28 (Default: 00, R/W) Ditto	R/W
21:20	IC27	Interrupt Source Control 27	Interrupt Source Control 27 (Default: 00, R/W) Ditto	R/W
19:18	IC26	Interrupt Source Control 26	Interrupt Source Control 26 (Default: 00, R/W) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
17:16	IC25	Interrupt Source Control 24	Interrupt Source Control 24 (Default: 00, R/W) Ditto	R/W
15:14	IC16	Interrupt Source Control 16	Interrupt Source Control 16 (Default: 00, R/W)) Ditto	R/W
13:12	IC15	Interrupt Source Control 15	Interrupt Source Control 15 (Default: 00, R/W)) Ditto	R/W
11:10	IC14	Interrupt Source Control 14	Interrupt Source Control 14 (Default: 00, R/W)) Ditto	R/W
9:8	IC13	Interrupt Source Control 13	Interrupt Source Control 13 (Default: 00, R/W)) Ditto	R/W
7:6	IC12	Interrupt Source Control 12	Interrupt Source Control 12 (Default: 00, R/W)) Ditto	R/W
5:4	IC11	Interrupt Source Control 11	Interrupt Source Control 11 (Default: 00, R/W)) Ditto	R/W
3:2	IC10	Interrupt Source Control 10	Interrupt Source Control 10 (Default: 00, R/W)) Ditto	R/W
1:0	IC9	Interrupt Source Control 9	Interrupt Source Control 9 (Default: 00, R/W)) Ditto	R/W

8.4.5. Interrupt Detection Mode Register 2 (IRDM2) 0xE8C8

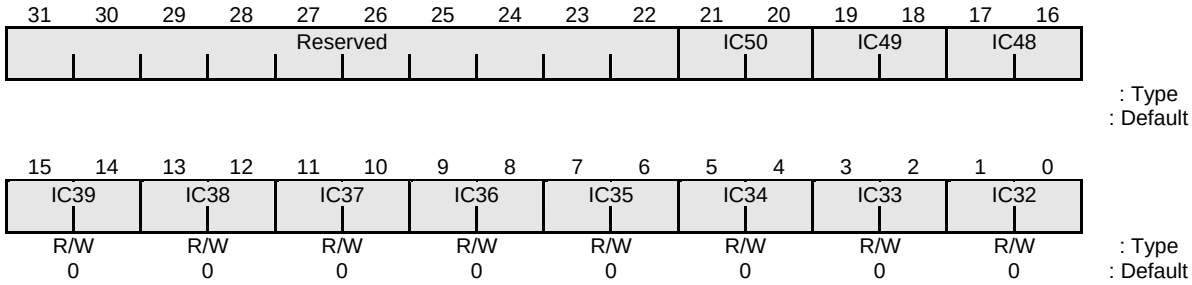


Figure 8-8 Interrupt Detection Mode Register 2

Table 8-10 Interrupt Detection Mode Register 2

Bits	Mnemonic	Field Name	Explanation	R/W
31:22	-	-	Reserved	-
21:20	IC50	Interrupt Source Control 50	Interrupt Source Control 50 (Default: 00) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
19:16	IC49	Interrupt Source Control 49	Interrupt Source Control 49 (Default: 00) Ditto	R/W
17:16	IC48	Interrupt Source Control 48	Interrupt Source Control 48 (Default: 00) Ditto	R/W
15:14	IC39	Interrupt Source Control 39	Interrupt Source Control 39 (Default: 00) Ditto	R/W
13:12	IC38	Interrupt Source Control 38	Interrupt Source Control 38 (Default: 00) Ditto	R/W
11:10	IC37	Interrupt Source Control 37	Interrupt Source Control 37 (Default: 00) Ditto	R/W
9:8	IC36	Interrupt Source Control 36	Interrupt Source Control 36 (Default: 00) Ditto	R/W
7:6	IC35	Interrupt Source Control 35	Interrupt Source Control 35 (Default: 00) Ditto	R/W
5:4	IC34	Interrupt Source Control 34	Interrupt Source Control 34 (Default: 00) Ditto	R/W
3:2	IC33	Interrupt Source Control 33	Interrupt Source Control 33 (Default: 00) Ditto	R/W
1:0	IC32	Interrupt Source Control 32	Interrupt Source Control 32 (Default: 00) Ditto	R/W

8.4.6. Interrupt Detection Mode Register3 (IRDM3) 0xE8D0

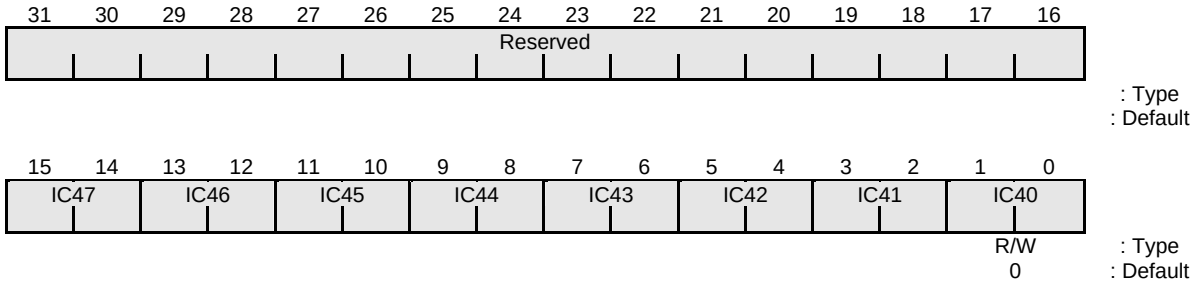


Figure 8-9 Interrupt Detection Mode Register 3

Table 8-11 Interrupt Detection Mode Register 3

Bits	Mnemonic	Field Name	Explanation	Read/Write
31:16	-	-	Reserved	-
15:14	IC47	Interrupt Source Control 47	Interrupt Source Control 47 (Default: 00) 00: Low level active 01: Reserved 10: Reserved 11: Reserved	R/W
13:12	IC46	Interrupt Source Control 46	Interrupt Source Control 46 (Default: 00) Ditto	R/W
11:10	IC45	Interrupt Source Control 45	Interrupt Source Control 45 (Default: 00) Ditto	R/W
9:8	IC44	Interrupt Source Control 44	Interrupt Source Control 44 (Default: 00) Ditto	R/W
7:6	IC43	Interrupt Source Control 43	Interrupt Source Control 43 (Default: 00) Ditto	R/W
5:4	IC42	Interrupt Source Control 42	Interrupt Source Control 42 (Default: 00) Ditto	R/W
3:2	IC41	Interrupt Source Control 41	Interrupt Source Control 41 (Default: 00) Ditto	R/W
1:0	IC40	Interrupt Source Control 40	Interrupt Source Control 40 (Default: 00) Ditto	R/W

8.4.7. Interrupt Mask Level Register (IRMSK) 0xE8A0

Figure 8-10 Interrupt Mask Level Register

Table 8-12 Interrupt Mask Level Register

Bits	Mnemonic	Field Name	Explanation	Read/Write
31:3	-	-	Reserved	-
2:0	IML	Interrupt Mask Level	Interrupt Mask Level (Default: 000) These bits specify the interrupt mask level. Masks interrupts with a mask level lower than the set mask level. 000: Interrupt mask level 0 (No interrupts masked) 001: Interrupt mask level 1 (Levels 2-7 enabled) 010: Interrupt mask level 2 (Levels 3-7 enabled) 011: Interrupt mask level 3 (Levels 4-7 enabled) 100: Interrupt mask level 4 (Levels 5-7 enabled) 101: Interrupt mask level 5 (Levels 6-7 enabled) 110: Interrupt mask level 6 (Level 7 enabled) 111: Interrupt mask level 7 (Interrupts disabled)	R/W

8.4.8. Interrupt Level Registers (IRLVLxx)

IRLVL Registers		31 30 29 28 27 26 25 24		23 22 21 20 19 18 17 16		15 14 13 12 11 10 09 08		07 06 05 04 03 02 01 00	
Name	OFFSET	Reserved	Assign	Reserved	Assign	Reserved	Assign	Reserved	Assign
IRLVL0	0xE820	—	IL18	—	IL17	—	IL2	—	IL1
IRLVL1	0xE828	—	IL20	—	IL19	—	IL4	—	IL3
IRLVL2	0xE830	—	IL22	—	IL21	—	IL6	—	IL5
IRLVL3	0xE838	—	IL24	—	IL23	—	IL8	—	IL7
IRLVL4	0xE840	—	IL26	—	IL25	—	IL10	—	IL9
IRLVL5	0xE848	—	IL28	—	IL27	—	IL12	—	IL11
IRLVL6	0xE850	—	IL30	—	IL29	—	IL14	—	IL13
IRLVL7	0xE858	—	---	—	IL31	—	IL16	—	IL15
IRLVL8	0xE860	—	IL49	—	IL48	—	IL33	—	IL32
IRLVL9	0xE868	—	—	—	IL50	—	IL35	—	IL34
IRLVL10	0xE870	—	—	—	—	—	IL37	—	IL36
IRLVL11	0xE878	—	—	—	—	—	IL39	—	IL38
IRLVL12	0xE880	—	—	—	—	—	IL41	—	IL40
IRLVL13	0xE888	—	—	—	—	—	IL43	—	IL42
IRLVL14	0xE890	—	—	—	—	—	IL45	—	IL44
IRLVL15	0xE898	—	—	—	—	—	IL47	—	IL46

Figure 8-11 Interrupt Level Registers

Table 8-13 Interrupt Level Register Field Definition (CASE of IRLVL0)

Bit	Mnemonic	Field Name	Explanation	Read/Write
31:27	—	—	Reserved	—
26:24	IL18	Interrupt Level 18	Interrupt Level of INT [18] (Default: 000) 000: Interrupt Level 0 (Interrupt disable) 001:Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7	R/W
23:19	—	—	Reserved	—
18:16	IL17	Interrupt Level 17	Interrupt Level of INT [17] (Default: 000) Ditto	R/W
15:11	—	—	Reserved	—
10:8	IL2	Interrupt Level 2	Interrupt Level of INT [2] (Default: 000) Ditto	R/W
7:3	—	—	Reserved	—
2:0	IL1	Interrupt Level 1	Interrupt Level of INT [1] (Default: 000) Ditto7	R/W

8.4.9. Interrupt Edge Detection Clear Register (IREDC) 0xE8A8

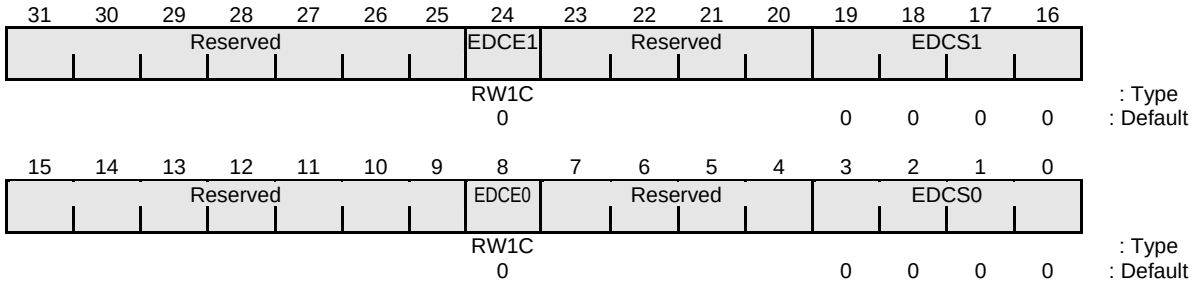


Figure 8-12 Interrupt Edge Detection Clear Register

Table 8-14 Interrupt Edge Dectection Clear Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:25	—	—	Reserved	—
24	EDCE1	Edge Detection Clear Enable 1	Edge Detection Clear Enable 1 (Default: 0) Clears edge detection of interrupts specified by the EDCS1 field 0: Does not clear. 1: Clears. Value always becomes “0” when this bit is read.	R/W1C
23:20	—	—	Reserved	—
19:16	EDCS1	Edge Detection Clear Source 1	Edge Detection Clear Source 1 (Default: 0x0) These bits specify the interrupt source to be cleared. 1111: Reserved 1110: Reserved 1101: Interrupt Channel [30] 1100: Interrupt Channel [29] 1011: Interrupt Channel [28] 1010: Interrupt Channel [27] 1001: Reserved 0000: Reserved	R/W
15:9	—	—	Reserved	—
8	EDCE0	Edge Detection Clear Enable 0	Edge Detection Clear Enable 0 (Default: 0) Clears edge detection of interrupts specified by the EDCS0 field. 0: Does not clear. 1: Clears. Value always becomes “0” when this bit is read.	R/W1C
7:5	—	—	Reserved	—
4:0	EDCS0	Edge Detection Clear Source 0	Edge Detection Clear Source 0 (Default: 0x0) These bits specify the interrupt source to be cleared. 1111: Reserved 0110: Reserved 0101: Reserved 0100: Interrupt Channel [5] 0011: Interrupt Channel [4] 0010: Interrupt Channel [3] 0001: Reserved 0000: Reserved	R/W

8.4.10. Interrupt Pending Register 0 (IRPND0) 0xE8B0

Indicates the status of each interrupt request regardless of the IRLVL 7-0 and IRMSK value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
—	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	: Type
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Default

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IS16	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	: Type
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Default

Figure 8-13 Interrupt Pending Register 0

Table 8-15 Interrupt Pending Register 0

Bit	Mnemonic	Field Name	Explanation	R/W
31	—	—	Reserved	—
30	IS31	Interrupt Status 31	IRINTREQ [31] Status (Default: 0, R) 1: Interrupt requests 0: No interrupt requests	R
29	IS30	Interrupt Status 30	IRINTREQ [30] Status Ditto	R
28	IS29	Interrupt Status 29	IRINTREQ [29] Status Ditto	R
27	IS28	Interrupt Status 28	IRINTREQ [28] Status Ditto	R
26	IS27	Interrupt Status 27	IRINTREQ [27] Status Ditto	R
25	IS26	Interrupt Status 26	IRINTREQ [26] Status Ditto	R
24	IS25	Interrupt Status 25	IRINTREQ [25] Status Ditto	R
23	IS24	Interrupt Status 24	IRINTREQ [24] status Ditto	R
22	IS23	Interrupt Status 23	IRINTREQ [23] status Ditto	R
21	IS22	Interrupt Status 22	IRINTREQ [22] status Ditto	R
20	IS21	Interrupt Status 21	IRINTREQ [21] status Ditto	R
19	IS20	Interrupt Status 20	IRINTREQ [20] status Ditto	R
18	IS19	Interrupt Status 19	IRINTREQ [19] status Ditto	R
17	IS18	Interrupt Status 18	IRINTREQ [18] status Ditto	R
16	IS17	Interrupt Status 17	IRINTREQ [17] status Ditto	R
15	IS16	Interrupt Status 16	IRINTREQ [16] status Ditto	R
14	IS15	Interrupt Status 15	IRINTREQ [15] status Ditto	R
13	IS14	Interrupt Status 14	IRINTREQ [14] status Ditto	R
12	IS13	Interrupt Status 13	IRINTREQ [13] status Ditto	R
11	IS12	Interrupt Status 12	IRINTREQ [12] status Ditto	R
10	IS11	Interrupt Status 11	IRINTREQ [11] status Ditto	R
9	IS10	Interrupt Status 10	IRINTREQ [10] status Ditto	R
8	IS9	Interrupt Status 9	IRINTREQ [9] status Ditto	R
7	IS8	Interrupt Status 8	IRINTREQ [8] status Ditto	R
6	IS7	Interrupt Status 7	IRINTREQ [7] status Ditto	R
5	IS6	Interrupt Status 6	IRINTREQ [6] status Ditto	R
4	IS5	Interrupt Status 5	IRINTREQ [5] status Ditto	R
3	IS4	Interrupt Status 4	IRINTREQ [4] status Ditto	R
2	IS3	Interrupt Status 3	IRINTREQ [3] status Ditto	R
1	IS2	Interrupt Status 2	IRINTREQ [2] status Ditto	R
0	IS1	Interrupt Status 1	IRINTREQ [1] status Ditto	R

8.4.11. Interrupt Pending Register 1 (IRPND1) 0xE8C0

Indicates the status of each interrupt request regardless of the IRLVL 7-0 and IRMSK value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved													IS50	IS49	IS48	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	: Type
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Default

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IS47	IS46	IS45	IS44	IS43	IS42	IS41	IS40	IS39	IS38	IS37	IS36	IS35	IS34	IS33	IS32	
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	: Type
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Default

Figure 8-14 Interrupt Pending Register 1

8

Table 8-16 Interrupt Pending Register 1

Bit	Mnemonic	Field Name	Explanation	R/W
31:19	—	—	Reserved	—
18	IS50	Interrupt Status 50	IRINTREQ [50] (Default: 0, R) 1: Interrupt requests 0: No interrupt requests	R
17	IS49	Interrupt Status 49	IRINTREQ [49] Ditto	R
16	IS48	Interrupt Status 48	IRINTREQ [48] Ditto	R
15	IS47	Interrupt Status 47	IRINTREQ [47] Ditto	R
14	IS46	Interrupt Status 46	IRINTREQ [46] Ditto	R
13	IS45	Interrupt Status 45	IRINTREQ [45] Ditto	R
12	IS44	Interrupt Status 44	IRINTREQ [44] Ditto	R
11	IS43	Interrupt Status 43	IRINTREQ [43] Ditto	R
10	IS42	Interrupt Status 42	IRINTREQ [42] Ditto	R
9	IS41	Interrupt Status 41	IRINTREQ [41] Ditto	R
8	IS40	Interrupt Status 40	IRINTREQ [40] Ditto	R
7	IS39	Interrupt Status 39	IRINTREQ [39] Ditto	R
6	IS38	Interrupt Status 38	IRINTREQ [38] Ditto	R
5	IS37	Interrupt Status 37	IRINTREQ [37] Ditto	R
4	IS36	Interrupt Status 36	IRINTREQ [36] Ditto	R
3	IS35	Interrupt Status 35	IRINTREQ [35] Ditto	R
2	IS34	Interrupt Status 34	IRINTREQ [34] Ditto	R
1	IS33	Interrupt Status 33	IRINTREQ [33] Ditto	R
0	IS32	Interrupt Status 32	IRINTREQ [32] Ditto	R

8.4.12. Interrupt Current Status Register (IRCS) 0xE8B8

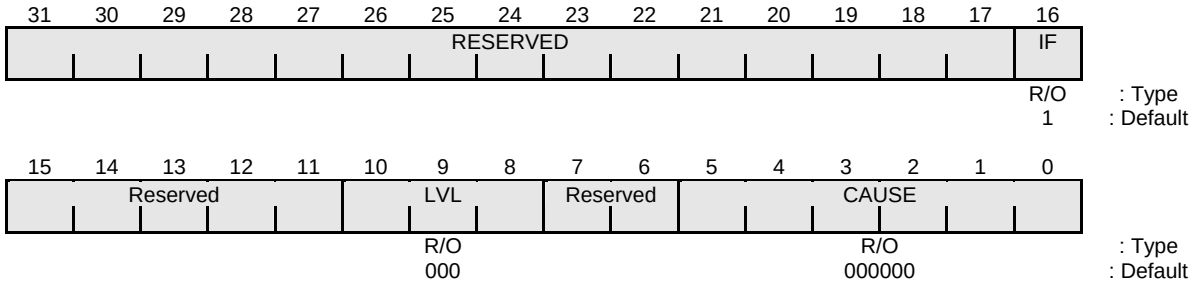


Figure 8-15 Interrupt Current Status Register

Table 8-17 Interrupt Current Status Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:17	—	—	Reserved	—
16	IF	Interrupt Flag	Interrupt Flag (Default: 1) This bit indicates the interrupt generation status. 0: Interrupt requests have been generated. 1: Interrupt requests have not been generated	R
15:11	—	—	Reserved	—
10:8	LVL	Interrupt Level	Interrupt Level (Default: 000) These bits specify the level of the interrupt request that was reported to the TX49/H4 core. [0] 000: Interrupt level 0 [1] 001: Interrupt level 1 [2] 010: Interrupt level 2 [3] 011: Interrupt level 3 [4] 100: Interrupt level 4 [5] 101: Interrupt level 5 [6] 110: Interrupt level 6 [7] 111: Interrupt level 7	R
7:6	—	—	Reserved	—
5:0	CAUSE	Interrupt Cause	Interrupt Cause (Default: 0x00) These bits specify the number of original interrupt cause specified in the column "IRCS.CAUSE Bits" in Table 8-1 TX4939 Interrupt Sources.	R

8

8.4.13. Interrupt Request Flag Register 0 (IRFLAG0) 0xE900

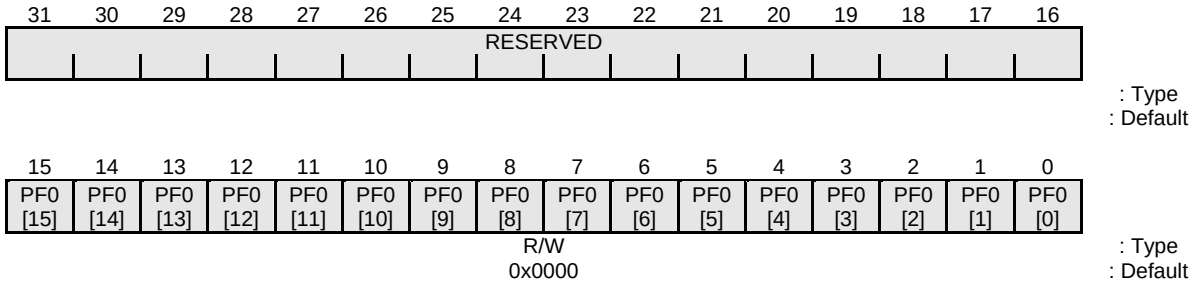


Figure 8-16 Interrupt Request Flag Register 0

Table 8-18 Interrupt Request Flag Register 0

Bit	Mnemonic	Field Name	Explanation	R/W
31:16	—	—	Reserved	—
15:0	PF0 [15:0]	Flag 0	Interrupt Request Flag 0 [15:0] (Default: 0x0000) Changes made to this register are reflected in Flag Register 1 also since they are the same registers. The bits in this field accept writes of both 1s and 0s.	R/W

8.4.14. Interrupt Request Flag Register 1 (IRFLAG1) 0xE908

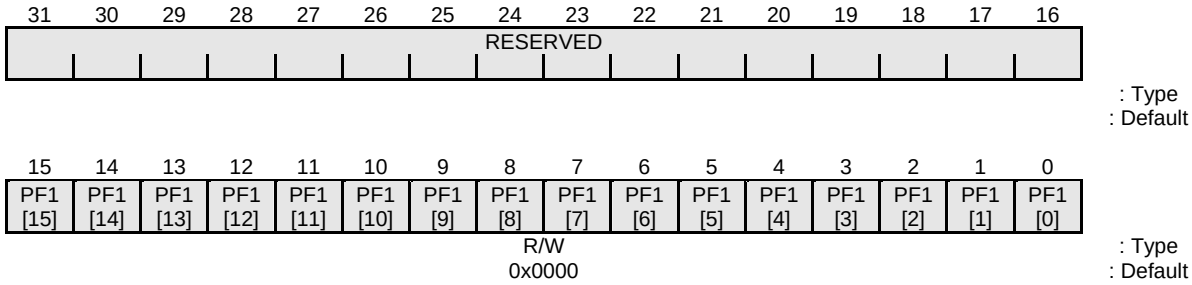


Figure 8-17 Interrupt Request Flag Register 1

Table 8-19 Interrupt Request Flag Register 1

Bit	Mnemonic	Field Name	Explanation	R/W
31:16	—	—	Reserved	—
15:0	PF1 [15:0]	Flag 1	Interrupt Request Flag 1 [15:0] (Default: 0x0000) Changes made to this register are reflected in Flag Register 0 also since they are the same registers. Both “0” and “1” can be written to Flag Register 0. Writes to Flag Register 1 operate as follows: Write From the TX49/H4 core 1: Set the flag bit 0: No change From other devices (DMAC, PCIC) 1: Clear the flag bit 0: No change Read: Read the flag bit	R/W

8

8.4.15. Interrupt Request Polarity Control Register (IRPOL) 0xE910

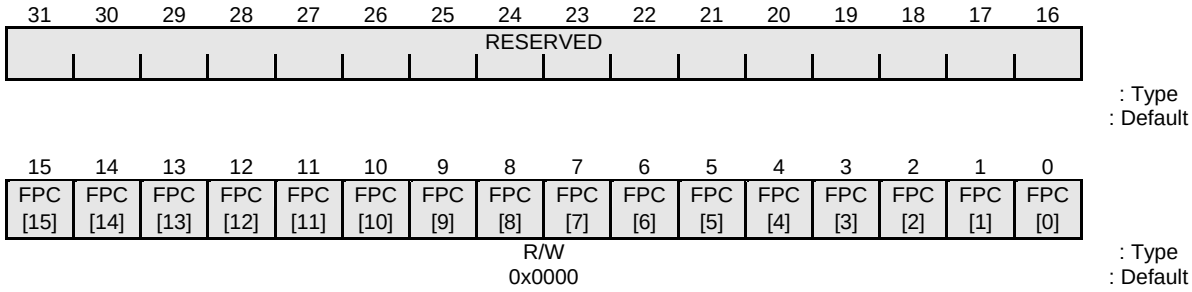


Figure 8-18 Interrupt Requests Polarity Control Register

Table 8-20 Interrupt Requests Polarity Control Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:16	—	—	Reserved	—
15:0	FPC [15:0]	Flag Polarity Control	Flag Polarity Control [15:0] (Default: 0x0000) These bits specify the polarity of the flag bit that generated the interrupt. An interrupt request is generated when the XOR of the FPC bit and the flag bit is “1.” Flag bit (PF) FPC bit Interrupt request 0 0 No 0 1 Yes 1 0 Yes 1 1 No	R/W

8.4.16. Interrupt Request Control Register (IRRCNT) 0xE918

Figure 8-19 Interrupt Request Control Register

Table 8-21 Interrupt Request Control Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:3	—	—	Reserved	—
2	ODC	External Interrupt OD Control	External Interrupt Open Drain Control (Default: 0) This bit specifies whether to make the external interrupt signal (IRC[2]*) an open drain pin or not. 0: Open drain (reset) 1: Totem pole	R/W
1	EXTPOL	External Interrupt Request Polarity Control	External Interrupt Polarity Control (Default: 1) This bit specifies the polarity of external interrupt requests. 0: Do not reverse polarity of interrupt requests. 1: Reverse polarity of interrupt requests	R/W
0	INTPOL	Internal Interrupt Request Polarity Control	Internal Interrupt Polarity Control (Default: 1) This bit specifies the polarity of internal interrupt requests. 0: Do not reverse polarity of interrupt requests. 1: Reverse polarity of interrupt requests	R/W

8.4.17. Interrupt Request Internal Interrupt Mask Register (IRMASKINT)

0xE920

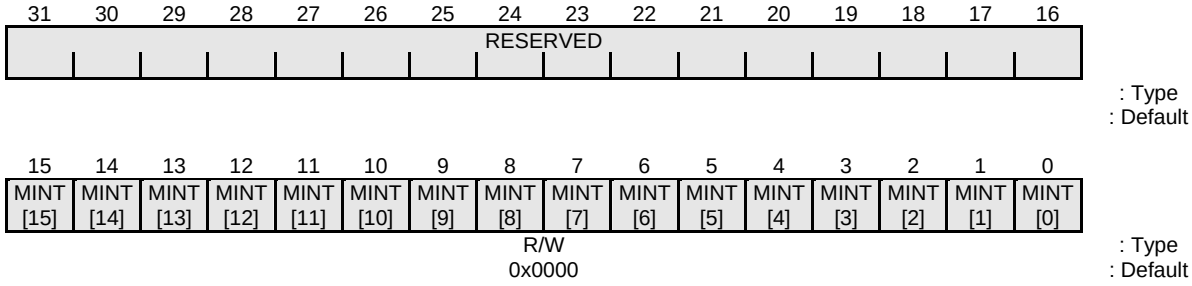


Figure 8-20 Interrupt Request Internal Interrupt Mask Register

Table 8-22 Interrupt Request Internal Interrupt Mask Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:16	—	—	Reserved	—
15:0	MINT [15:0]	Internal Request Mask	Internal Interrupt Mask (Default: 0x0000) These bits specify whether to use the corresponding flag bit as an internal interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (Reset) 1: Do not mask	R/W

8.4.18. Interrupt Request External Interrupt Mask Register (IRMASKEXT) 0xE928

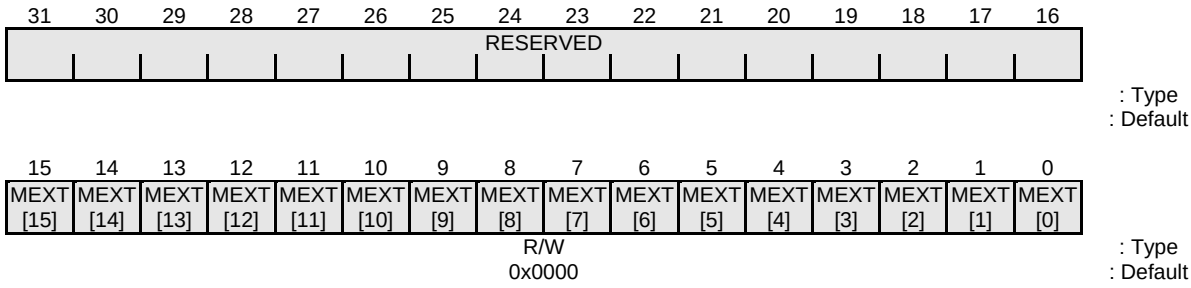


Figure 8-21 Interrupt Request External Interrupt Mask Register

Table 8-23 Interrupt Request External Interrupt Mask Register

Bit	Mnemonic	Field Name	Explanation	R/W
31:16	—	—	Reserved	—
15:0	MEXT [15:0]	External Request Mask	External Interrupt Mask (Default: 0x0000) These bits specify whether to use the corresponding flag bit as an external interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (reset) 1: Do not mask	R/W

8.4.19. Interrupt Debug Register 0 (IRDBR0) 0xE8D8

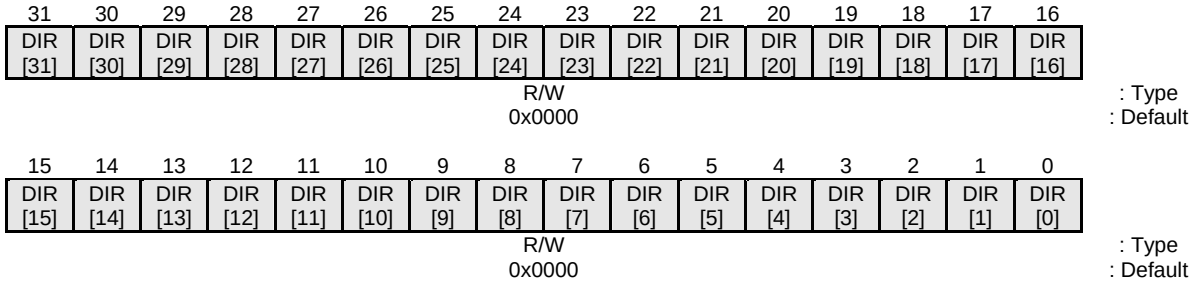


Figure 8-22 Interrupt Debug Register 0

Table 8-24 Interrupt Debug Register 0

Bit	Mnemonic	Field Name	Explanation	R/W
31:0	DIR [31:0]	Debug Interrupt Request	Debug Interrupt Request 0 (Default: 0x00000000) These bits are for software debug purposed only. A ‘1’ on each bit will cause the similar interrupt as in Table 8-1 0: No Interrupt 1: Interrupt Note: DIR[0] and DIR[31] are not used	R/W

8.4.20. Interrupt Debug Register 1 (IRDBR1) 0xE8E0

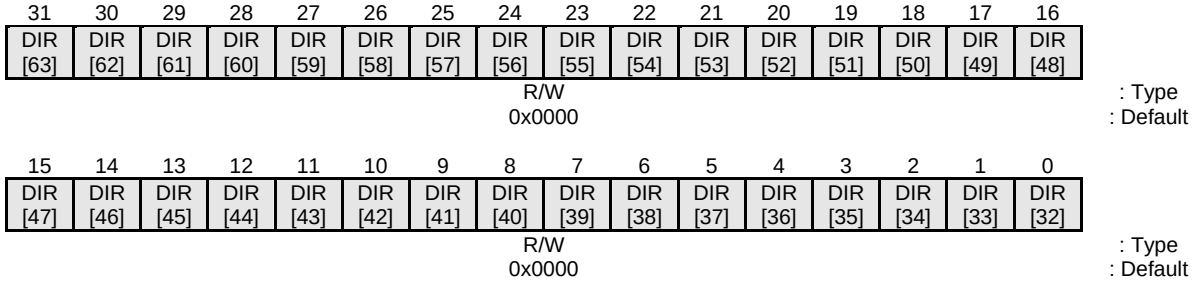


Figure 8-23 Interrupt Debug Register 1

Table 8-25 Interrupt Debug Register 1

Bit	Mnemonic	Field Name	Explanation	R/W
31:0	DIR [63:32]	Debug Interrupt Request	Debug Interrupt Request 1(Default: 0x00000000) These bits are for software debug purposed only. A '1' on each bit will cause the similar interrupt as in Table 8-1 0: No Interrupt 1: Interrupt Note: DIR[63:51], DIR[35], and DIR[32] are not used	R/W

8.4.21. Interrupt Debug Enable Register (IRDBEN) 0xE8E8

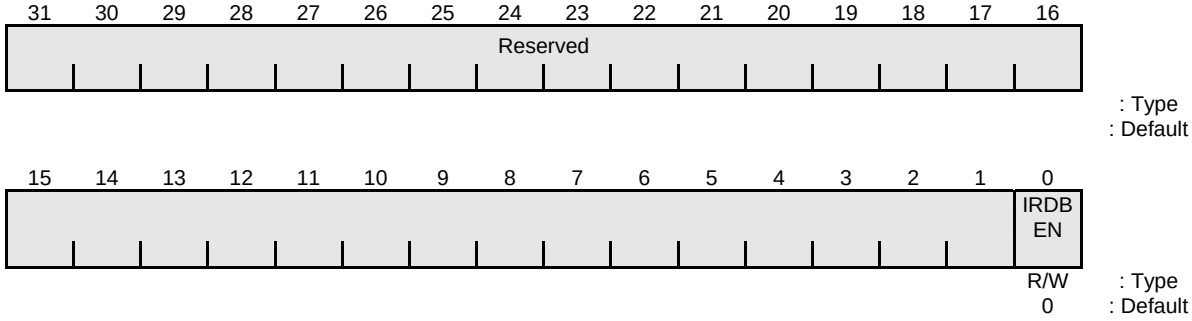


Figure 8-24 Interrupt Debug Enable Register

Table 8-26 Interrupt Debug Enable Register

Bit(s)	Mnemonic	Field Name	Explanation	R/W
31:1	—	—	Reserved	—
0	IRDBEN	Interrupt Debug Enable	Interrupt Debug Enable (Default: 0) Enables interrupt debug registers 0: Disable Interrupt debug registers 1: Enable Interrupt debug registers	R/W

Chapter 9. External Bus Interface

9.1. Basic Structure

External Bus Interface supports NOR-Flash interface (as boot device), NAND-Flash control, and two kinds of ASIC I/O interfaces. In terms of bus width, 8/16 bus width are available. Figure 9-1 shows typical configuration with 8/16-bit bus mode.

9.1.1. External Bus Connection Diagram

Figure 9-1 below shows an example of NOR-Flash, NAND-Flash, and ASIC connectivity on the External bus.

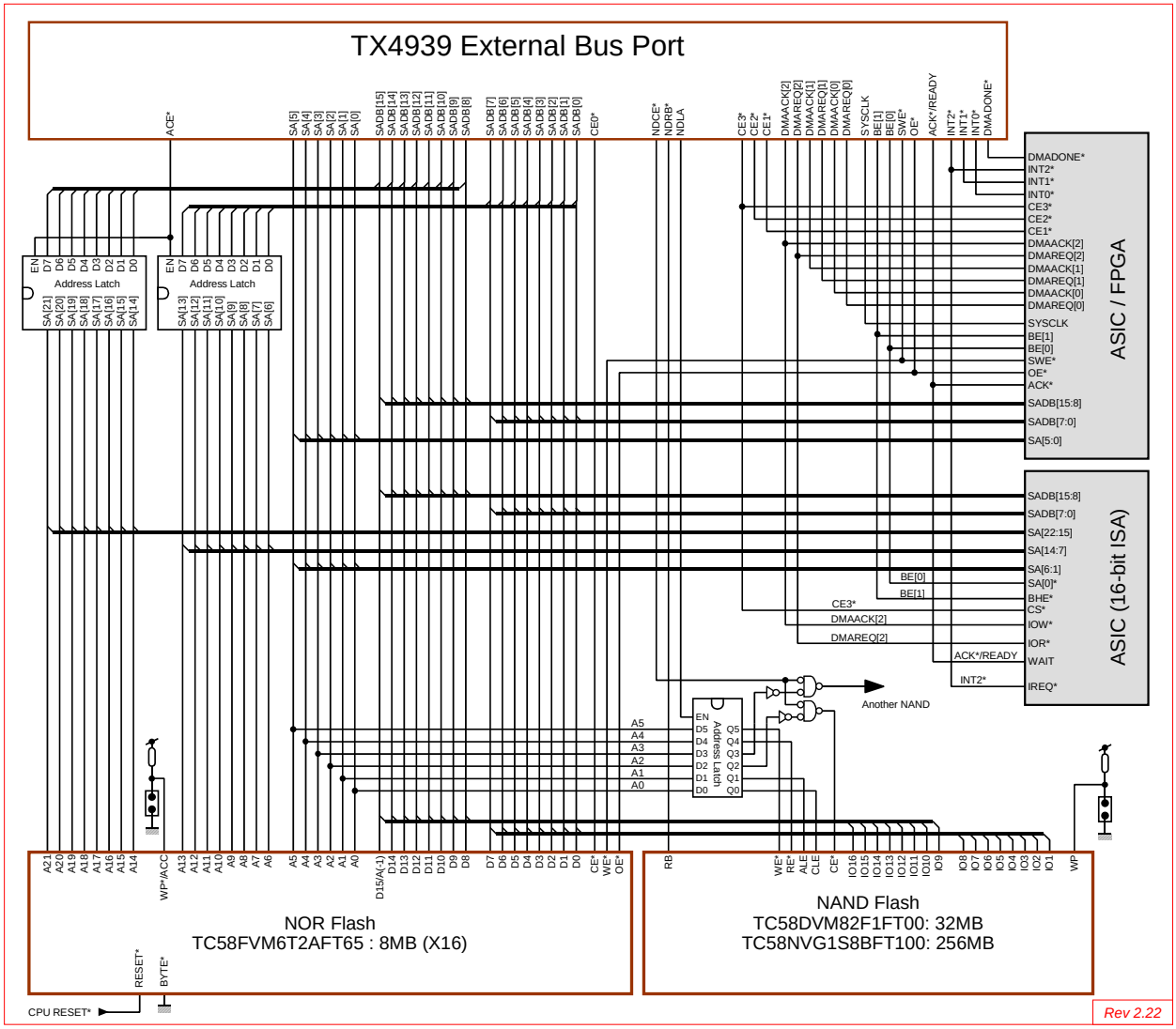


Figure 9-1 External Circuit for External Bus Interface for 8/16-bit mode.

Note: If ISA mode is enable then DMAREQ[2] and DMAACK[2] are dedicated for ISA interface.
Otherwise if ISA mode is not used then DMAREQ[2] and DMAACK[2] can be used for DMA transfer.

9.2. External Bus Controller

9.2.1. External Bus Channel Control Registers

The External Bus Controller (EBC) has four (4) independent channels. Each channel has own Channel Control Register (EBCCRn: n=0-3),that can be configured separately. Table 9-1 is the list of these registers. The channel 0 is dedicated for boot device.

These control registers can be accessed either WORD or DWORD; however, EBCCRn.ME bit should be enabled at the last timing of WORD access. Otherwise, undesirable memory access may happen.

Table 9-1 External Bus Control Registers

OFFSET ADDRESS	BIT WIDTH	REGISTER SYMBOL	REGISTER NAME
0x9000	64	EBCCR0	E-Bus Channel Control Register 0 (Boot Device)
0x9008	64	EBCCR1	E-Bus Channel Control Register 1
0x9010	64	EBCCR2	E-Bus Channel Control Register 2
0x9018	64	EBCCR3	E-Bus Channel Control Register 3
0x9020	64	EBCCR4	E-Bus Channel Control Register 4 (Reserved)
0x9028	64	EBCCR5	E-Bus Channel Control Register 5 (Reserved)
0x9030	64	EBCCR6	E-Bus Channel Control Register 6 (Reserved)
0x9038	64	EBCCR7	E-Bus Channel Control Register 7 (Reserved)

9.2.2. Boot Up Options

Since EBCCR0 is dedicated for boot device, EBCCR0 is set enable as boot channel. (EBCCR0.ME = 1'b1)

Table 9-2 Boot Configuration for Channel 0 (Subset of Table 4-1 Boot Configuration Details)

<table>
<tr><th>Signal</th><th>Description</th><th>Corresponding Register Bit</th></tr>
<tr><td>NO PIN ASSIGNED</td><td>Selects the operation mode of the ACK*/READY signal.

1 = ACK*/READY Static mode
0 = ACK*/READY Dynamic mode (Default)</td><td>CCFG.ARMODE</td></tr>
<tr><td>NO PIN ASSIGNED</td><td>Set the address hold time relative to the ACE* signal.

1 = Address changes 1 clock cycle after deassertion of the ACE* signal (Default)</td><td>CCFG.ACEHOLD</td></tr>
<tr><td>SADB[1]</td><td>Specifies the function of the BE[1:0]*/BWE[1:0]* pins upon booting.

1 = BWE[3:0]* (Byte Write Enable)
0 = BE[3:0]* (Byte Enable)</td><td>CCFG_BESEL</td></tr>
<tr><td>SADB[2]</td><td>Boot ACK* Input : Specifies the access mode for external bus controller channel 0.

1 = Normal mode
0 = External ACK mode</td><td>CCFG_ACKSEL</td></tr>
<tr><td>SADB[3]</td><td>Boot ROM Bus Width. Specifies the data bus width when booting from a memory device connected to the External bus controller.

<table><tr><td>SADB [3]</td><td>Boot ROM Bus Width</td></tr><tr><td>0</td><td>16 bit</td></tr><tr><td>1</td><td>8 bit</td></tr></table></td><td>CCFG_ROMW</td></tr>
</table>

9.2.3. Address Mapping

Each of the four channels can use the Base Address field (EBCCRn.BA[35:20]) and the Channel Size field (EBCCRn.CS[3:0]) of the External Bus Channel Control Register to map to any physical address.

A channel is selected when the following equation becomes True.

paddr[35:20] & !Mask[35:20] == BA[35:20] & !Mask[35:20]

In the above equation, paddr represents the accessed physical address, Mask[35:20] represents the address mask value selected from Table 9-3 from the Channel Size field value, the ampersand (&) represents the AND operation, and the exclamation mark (!) represents the Logical NOT for each bit.

Table 9-3 Address Mask

CS[3:0]	Channel Size	Address Mask[35:20]
0000	1 MB	0000_0000_0000_0000
0001	2 MB	0000_0000_0000_0001
0010	4 MB	0000_0000_0000_0011
0011	8 MB	0000_0000_0000_0111
0100	16 MB	0000_0000_0000_1111
0101	32 MB	0000_0000_0001_1111
0110	64 MB	0000_0000_0011_1111
0111	128 MB	0000_0000_0111_1111
1000	256 MB	0000_0000_1111_1111
1001	512 MB	0000_0001_1111_1111
1010	Reserved	Reserved
1011	Reserved	Reserved
1100	Reserved	Reserved
1101	Reserved	Reserved
1110	Reserved	Reserved
0111	Reserved	Reserved

9.2.4. External Address Output

SA [5:0] is dedicated address signal. The middle address will be provided through SADB [15:0].

External device latches the middle sixteen address bits and the upper six address bits by using the ACE* signal. Either the ACE* signal itself can be used as a Latch Enable signal or the upper and middle address can be latched at the rise of SYSCLK when the ACE* signal is being asserted.

The ADDR signal output is held for one clock cycle after the ACE* signal rise when the CCFG.ACEHOLD bit is set (default). The ADDR signal output is not held when the CCFG.ACEHOLD bit is cleared. This hold time setting is applied globally to all channels.

The ACE* signal of the upper address is always asserted for starting of every transaction. When there is a burst transfer, ACE* signal is only asserted for the first cycle.

9.2.5. Address Bit Corresponding in the 16-bit Mode

In case of 16-bit data bus width mode, 512 MB memory space (2^{29}) is accessible. Table 9-4 shows this corresponding.

Table 9-4 Address Bit Correspondin in the 16-bit Mode

Latched SADB Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Middle Address	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7

Latched SA Bit	5	4	3	2	1	0	Non-latched SA	5	4	3	2	1	0
Upper Address	28	27	26	25	24	23	Lower Address	6	5	4	3	2	1

Note: Address is expressed by BYTE ADDRESS

When a Single cycle that accesses 1-Byte or 1 half-word data is executed, 16-bit access is executed only once on the external bus. 16-bit access is executed twice when performing 1-word access. 16-bit access is executed four times when performing 1-double-word access. When a Burst cycle is executed, four 16-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than double-word data.

9.2.6. Address Bit Corresponding in the 8-bit Mode

In case of 8-bit data bus width mode, 256 MB memory space (2^{28}) is accessible. Table 9-5 shows this corresponding.

Table 9-5 Address Bit Correspondin in the 8-bit Mode

Latched SADB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Middle Address	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6

Latched SA	5	4	3	2	1	0	Non-latched SA	5	4	3	2	1	0
Upper Address	27	26	25	24	23	22	Lower Address	5	4	3	2	1	0

Note: Address is expressed by BYTE ADDRESS

When a Single cycle that accesses 1-Byte data is executed, 8-bit access is executed only once on the external bus. 8-bit access is executed twice when performing 1-half-word access. 8-bit access is executed four times when performing 1-word access. 8-bit access is executed eight times when performing 1-double-word access. When a Burst cycle is executed, eight 8-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than double-word data.

9.2.7. Access Mode

The following four modes are available as controller access modes.
These modes can be set separately for each channel.

- *Normal mode*
- *Page mode*
- *External ACK mode*
- *Ready mode*

Depending on the combination of modes in each channel, either of two modes in which the ACK*/Ready signal operates differently (ACK*/Ready Dynamic mode, ACK*/Ready Static mode) is selected by the ACK*/Ready Mode bit (CCFG.ARMODE) of the Chip Configuration Register. The mode selected is applied globally to all channels.

(1) ACK*/READY Dynamic mode (CCFG.ARMODE = 0)

This mode is selected in the initial state.
The ACK*/Ready signal automatically switches to either input or output according to the setting of each channel. When in the Normal mode or the Page mode, the ACK*/Ready signal is an output signal, and the internally generated ACK* signal is output. When in the External ACK* or Ready mode, the ACK*/Ready signal becomes an input signal. The ACK*/Ready signal outputs High if there is no access to the External Bus Controller. However, this signal may output Low during access to SDRAM.

(2) ACK*/Ready Static mode (CCFG.ARMODE = 0)

The internally generated ACK* signal is not output when in either the Normal mode or Page mode. Therefore, the ACK*/Ready signal will not become an output in any channel.

Access using Burst transfer by the internal bus (G-Bus) is supported when in a mode other than the Ready mode. However, the Ready mode is not supported.

Table 9-6 Operation Mode

	PM	RDY	PWT:WT	Mode	ACK*/READY Pin State	Access End Timing State	G-Bus Burst Access
ACK*/Ready Dynamic Mode	0	0	!3f	Normal	Output	Internally Generated ACK*	√
			3f	External ACK*	Input	ACK* Input	√
		1	—	READY	Input	Ready Input	—
	!0	0	—	Page	Output	Internally Generated ACK*	√
		1	—	Reserved	—	—	—
ACK*/Ready Static Mode	0	0	!3f	Normal	Hi-Z	Internally Generated ACK*	√
			3f	External ACK*	Input	ACK* Input	√
		1	—	READY	Input	Ready Input	—
	!0	0	—	Page	Hi-Z	Internally Generated ACK*	√
		1	—	Reserved	—	—	—

9.2.7.1. Normal Mode

When in this mode, the ACK*/Ready signal becomes an ACK* output when it is in the ACK*/Ready Dynamic mode. The ACK*/Ready signal becomes High-Z when it is in the ACK*/Ready Static mode.

Wait cycles are inserted according to the EBCCRn.PWT and EBCCRn.WT value at the access cycle. The Wait cycle count is 0 to 0x3e (becomes the external ACK mode when set to EBCCRn.PWT: WT = 0x3f).

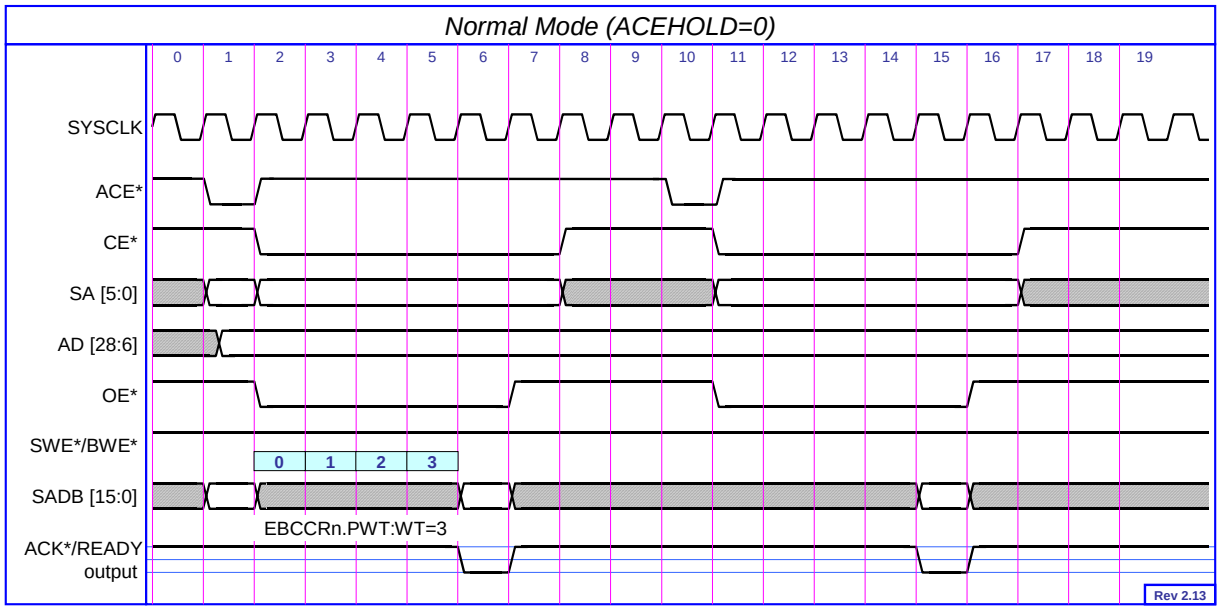


Figure 9-2 Normal Mode (ACEHOLD=0)

9.2.7.2. External ACK Mode

When in this mode, the ACK*/READY pin becomes ACK* input, and the cycle is ended by the ACK* signal from an external device. ACK* input is internally synchronized. Refer to Section "7.3.7.4 ACK* Input Timing" for more information regarding timing.

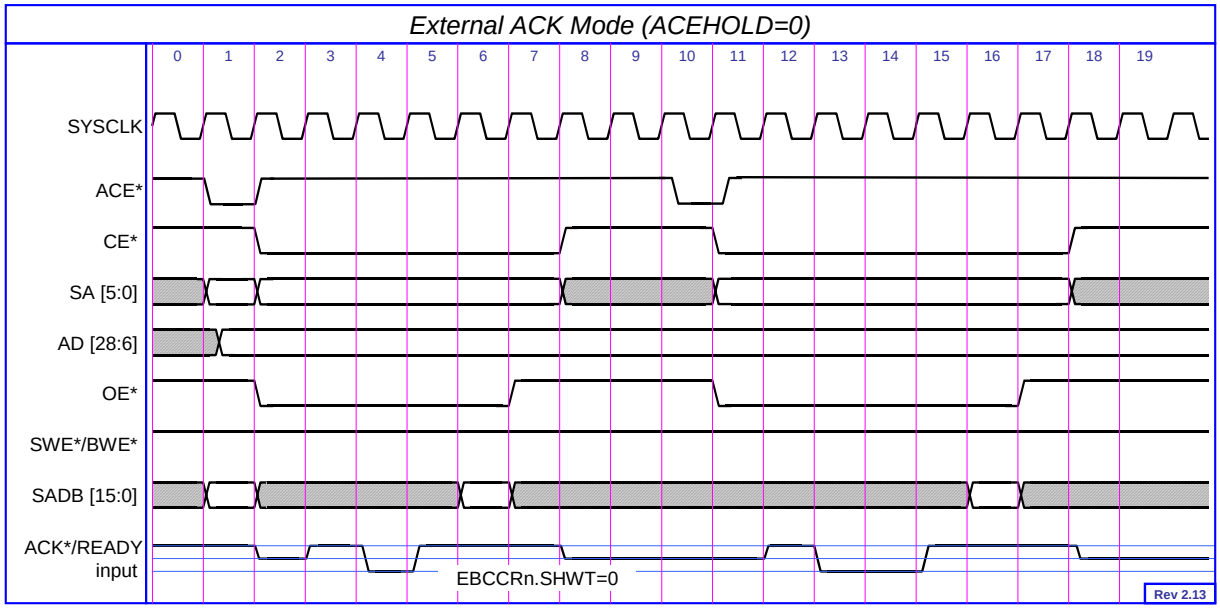


Figure 9-3 External ACK Mode (ACEHOLD=0)

9.2.7.3. Ready Mode

When in this mode, the ACK*/Ready pin becomes Ready input, and the cycle is ended by Ready input from an external device. Ready input is internally synchronized. Refer to Section “9.2.8.5 Ready Input Timing” for more information regarding timing.
When the Wait cycle count specified by EBCCRn.PWT:WT elapses, a check is performed to see whether the Ready signal was asserted. Since EBCCRn.WT[0] is used to indicate the ACK*/ Ready Static/Dynamic mode, it is not used for setting the Wait cycle count. Therefore, the Wait cycle count that can be set by the Ready mode is 0, 2, 4, 6 … 62. The Ready mode does not support Burst access by the internal bus.

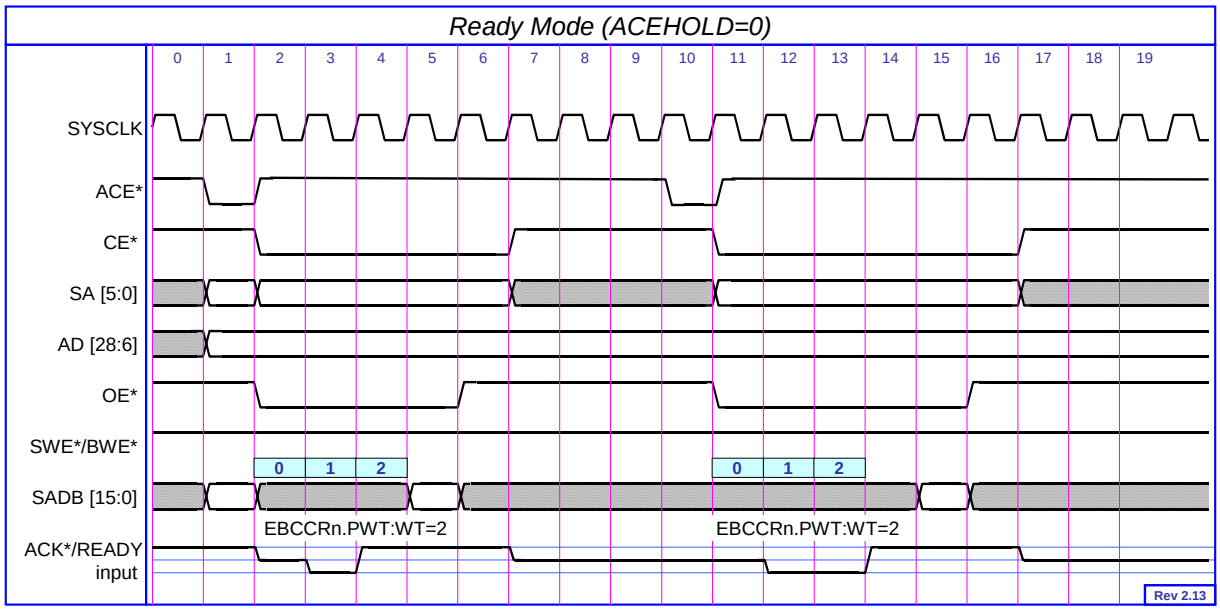


Figure 9-4 Ready Mode (ACEHOLD=0)

9.2.7.4. Page Mode

When in this mode, the ACK*/Ready pin becomes ACK* output when it is in the Dynamic mode. When it is in the ACK*/Ready Static mode, the ACK*/Ready signal becomes HiZ. Wait cycles are inserted into the access cycle according to the values of EBCCRn.PWT and EBCCRn.WT. The Wait cycle count in the first access cycle of Single access or Burst access is determined by the EBCCRn.WT value. The Wait cycle count can be set from 0 to 15. The Wait cycle count of subsequent Burst cycles is determined by the EBCCRn.PWT value. The Wait cycle count can be set from 0 to 3.

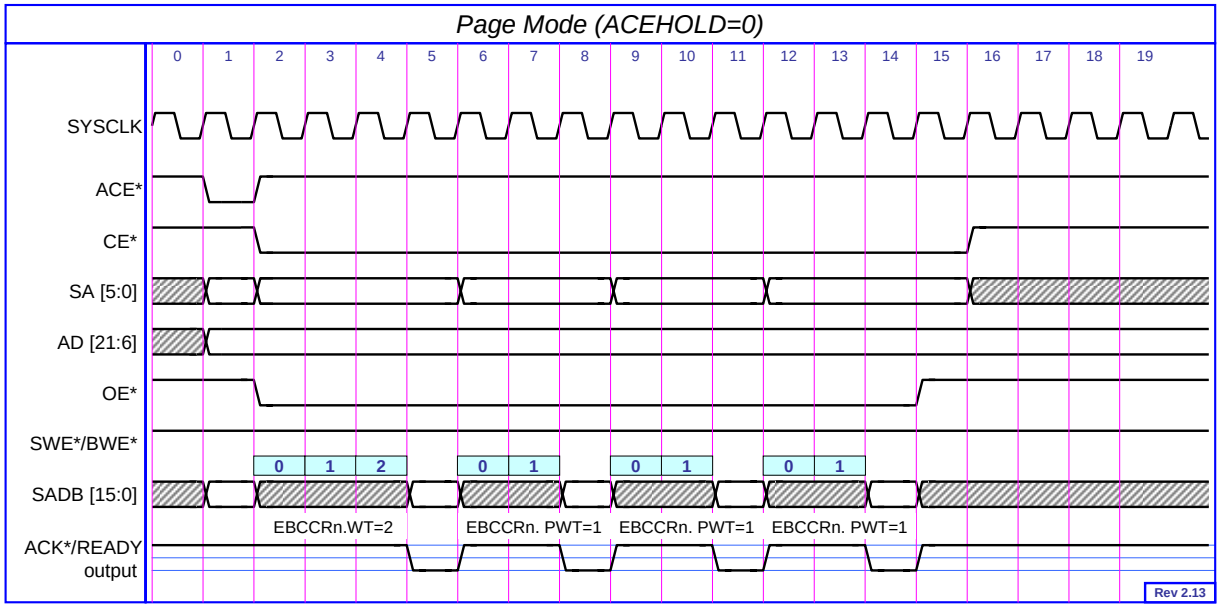


Figure 9-5 Page Mode (ACEHOLD=0)

9.2.8. Access Timing

9.2.8.1. SHWT Option

The SHWT option is selected when the SHWT (Setup/Hold Wait Time) field of the Channel Control Register is a value other than “0.” This option inserts a Setup cycle and a Hold cycle between the current signal and the next signal.

Setup cycle: CE* from ADDR, OE* from CE*, BWE* from CE*, SWE* from CE*.
Hold cycle: ADDR from CE*, CE* from OE*, CE* from BWE*, CE* from SWE*

This option is used for I/O devices that are generally slow. All Setup cycles and Hold cycles will be identical, so each cycle cannot be set individually. The SHWT mode cannot be used by the Page mode. The SHWT mode can be used by all other modes, but there is one restriction: the internal bus cannot use Burst access.

The hold cycles of SADB relative to SWE* and BWE* are fixed at one clock cycle, regardless of setting of the SHWT option. When the SHWT option is disabled, the setup cycles of SWE* and BWE* relative to CE*, and the hold cycles of SWE*and BWE* relative to CE* are one clock cycle.

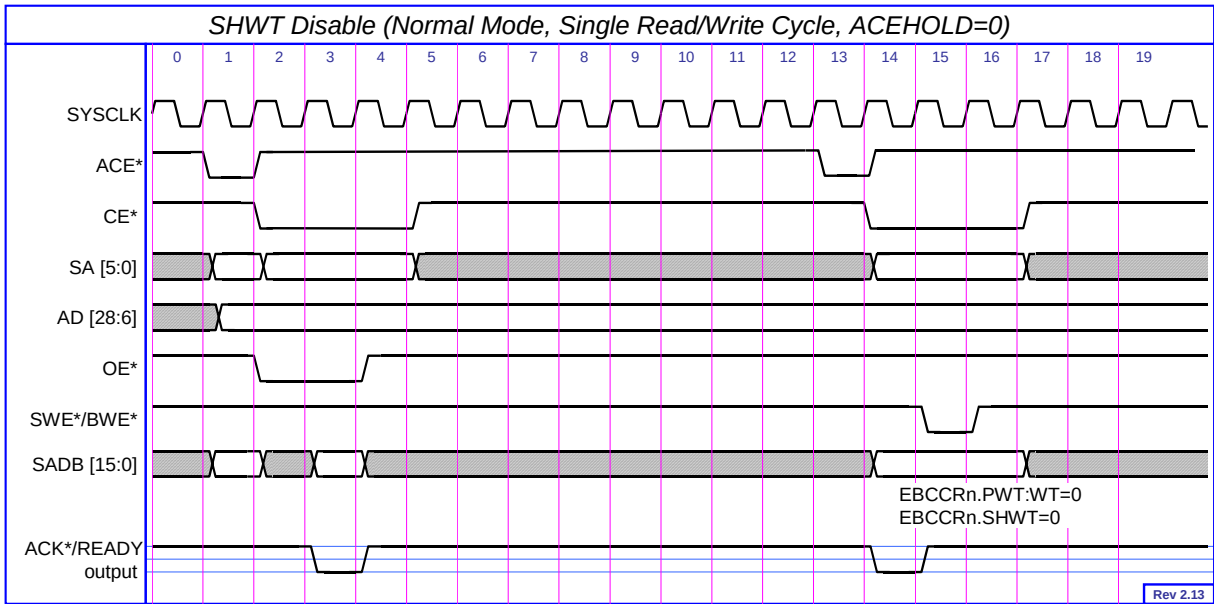


Figure 9-6 SWHT Disable (Normal Mode, Single Read/Write Cycle)

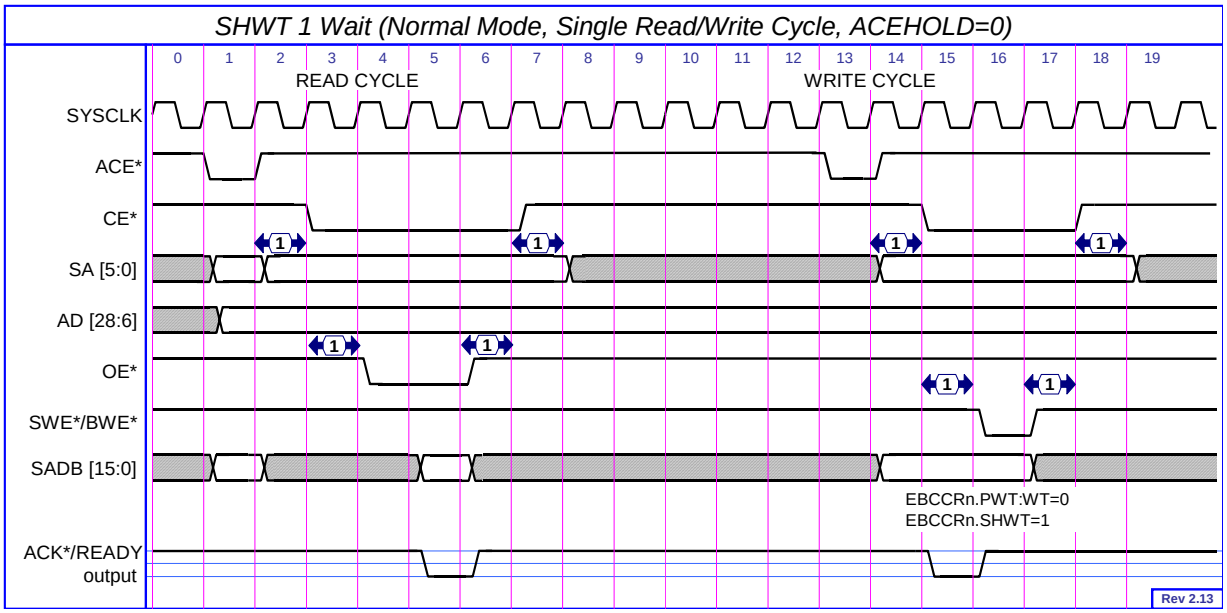


Figure 9-7 SHWT 1 Wait (Normal Mode, Single Read/Write Cycle)

9

9.2.8.2. ACK*/READY Input/Output Switching Timing

When in the ACK*/Ready Static mode, the ACK*/Ready signal is always an input signal. When in the ACK*/Ready Dynamic mode, the ACK*/Ready signal is an input signal when in the External ACK mode or the Ready mode, but is an output signal in all other modes.

During External ACK mode or Ready mode access, the ACK* signal becomes High-Z at the cycle where the CE* signal is asserted. At the end of the access cycle, the ACK* signal is output (driven) again one clock cycle after the CE* signal is de-asserted.

9.2.8.3. ACK* Output Timing (Normal Mode, Page Mode)

When in the Normal mode and Page mode of the ACK*/Ready Dynamic mode, the ACK* signal becomes an output signal and is asserted for one clock cycle to send notification to the external device of the data Read and data Write timing.

During the Read cycle, the data is latched at the rise of the next clock cycle after when the ACK* signal is asserted.

During the Write cycle, SWE*/BWE* is deasserted at the next clock cycle after when the ACK* signal is deasserted, and the data is held for one more clock cycle after that.

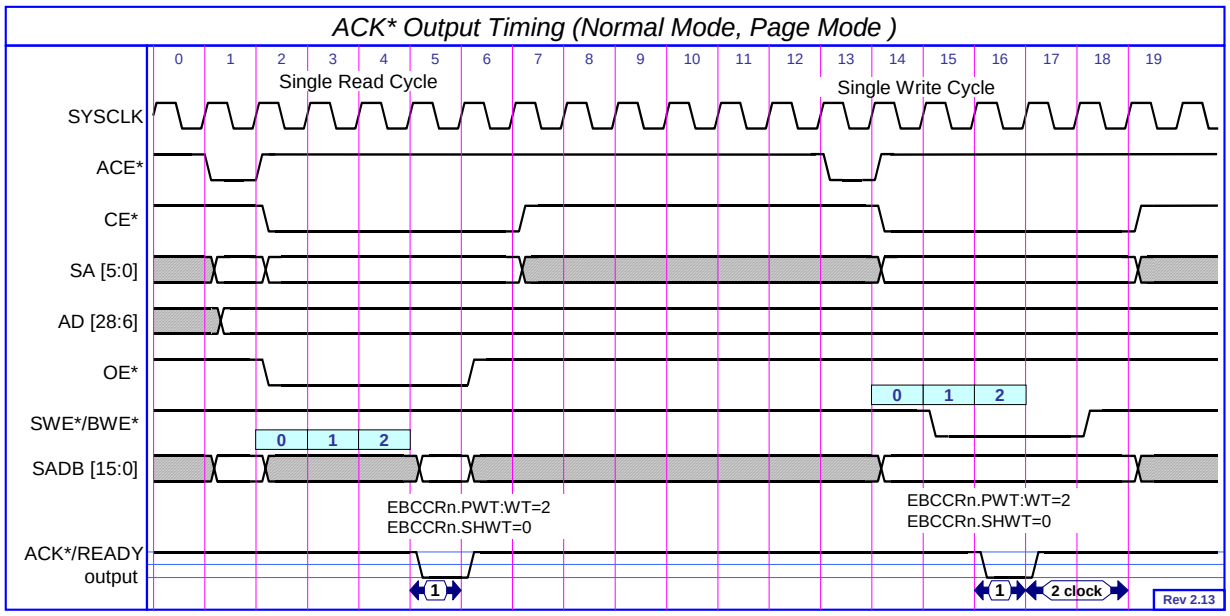


Figure 9-8 ACK* Output Timing (Single Read/Write Cycle)

9.2.8.4. ACK* Input Timing (External ACK Mode)

The ACK* signal becomes an input signal when in the external ACK mode.

During a Read cycle, data is latched two clock cycles after assertion of the ACK* signal is acknowledged ( ). During a Write cycle, assertion of the ACK* signal is acknowledged, SWE*/BWE* is deasserted three clock cycles later, then data is held for one clock cycle after that ( ).

The ACK* input signal is internally synchronized. Due to internal State Machine restrictions, ACK* cannot be acknowledged consecutively on consecutive clock cycles. External devices can assert ACK* across multiple clock cycles under the following conditions.

During Single access, the ACK* signal can be asserted before the end of the cycle during which CE* is dasserted. During Burst access, it is possible to assert the ACK* signal for up to three clock cycles during Reads and for up to five clock cycles during Writes. If the ACK* signal is asserted for a period longer than this, it will be acknowledged as the next valid ACK* signal.

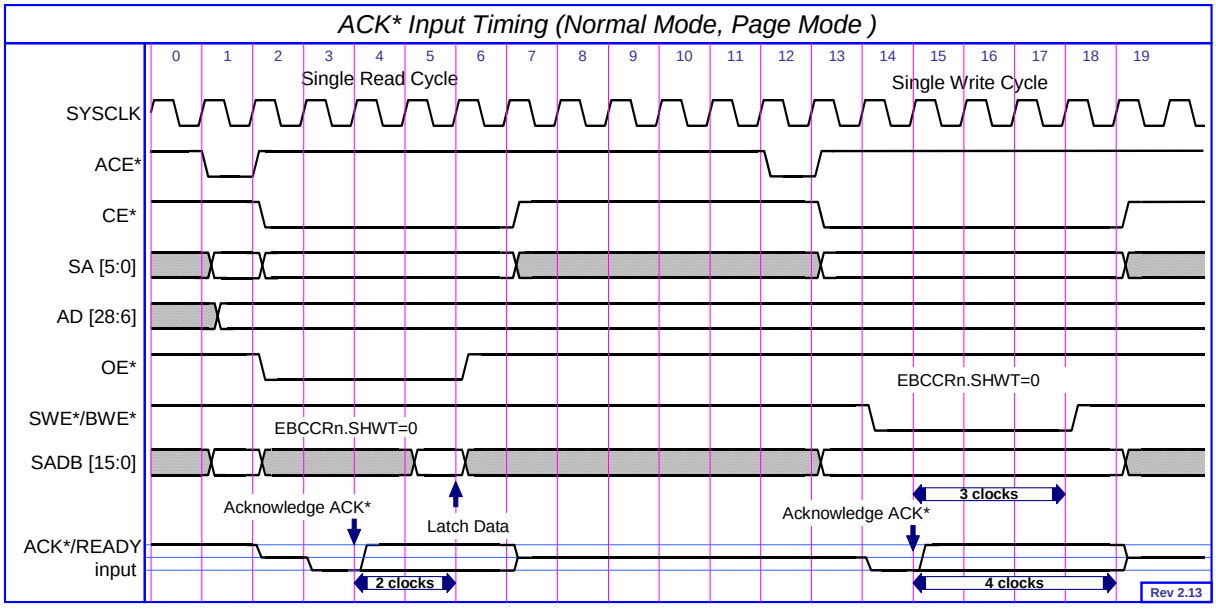


Figure 9-9 ACK* Input Timing (Single Read/Write Cycle)

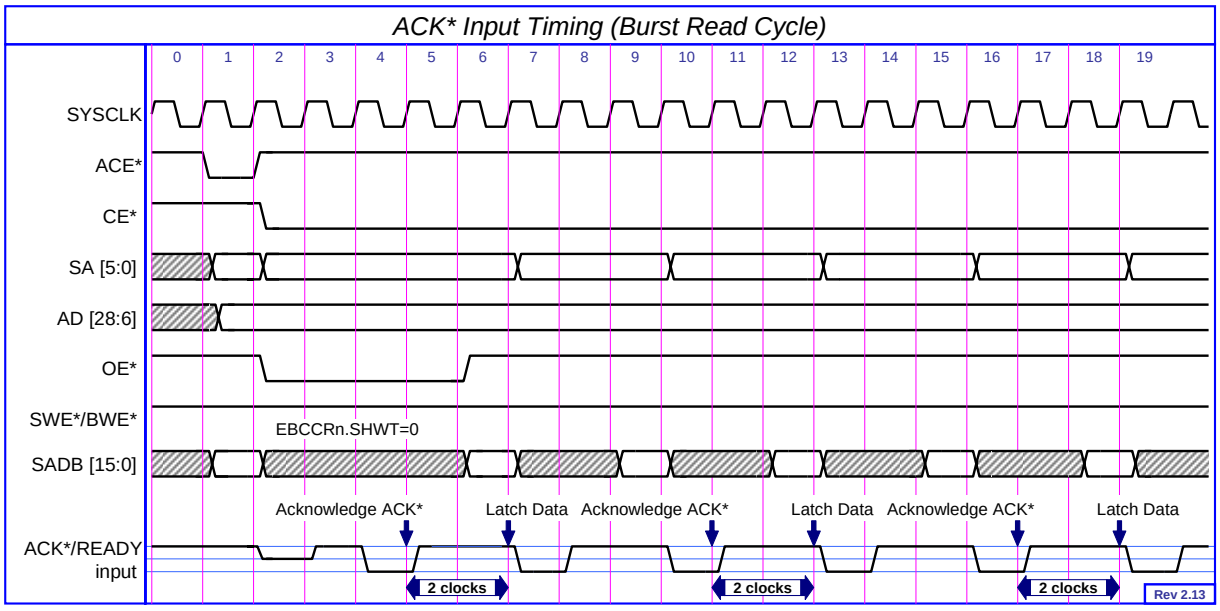


Figure 9-10 ACK Input Timing (Burst Reade Cycle)*

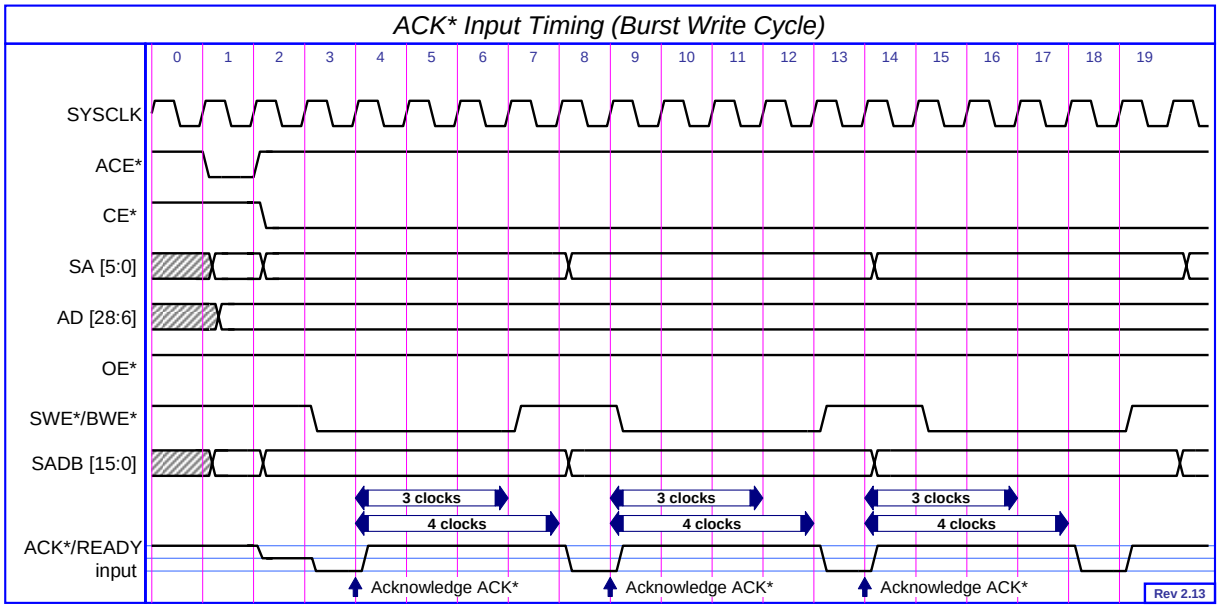


Figure 9-11 ACK Input Timing (Burst Write Cycle)*

9.2.8.5. Ready Input Timing

The ACK*/Ready pin is used as a Ready input when in the Ready mode. The Ready input timing is the same as the ACK* input timing explained in 9.2.8.4 ACK* Input Timing (External ACK Mode) with the two following exceptions.

Ready must be a High Active signal.
When in the Ready mode, the Wait cycle count specified by EBCCRn.PWT:WT must be inserted in order to delay the Ready signal check (see 9.2.7.3 Ready Mode).

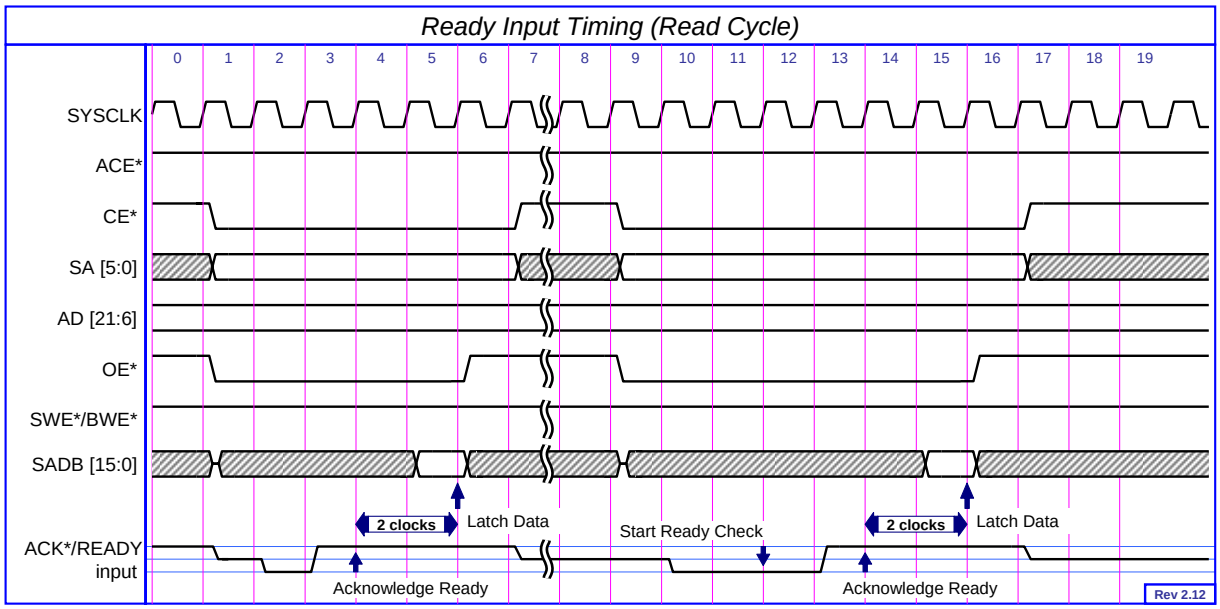


Figure 9-12 Ready Input Timing (Read Cycle)

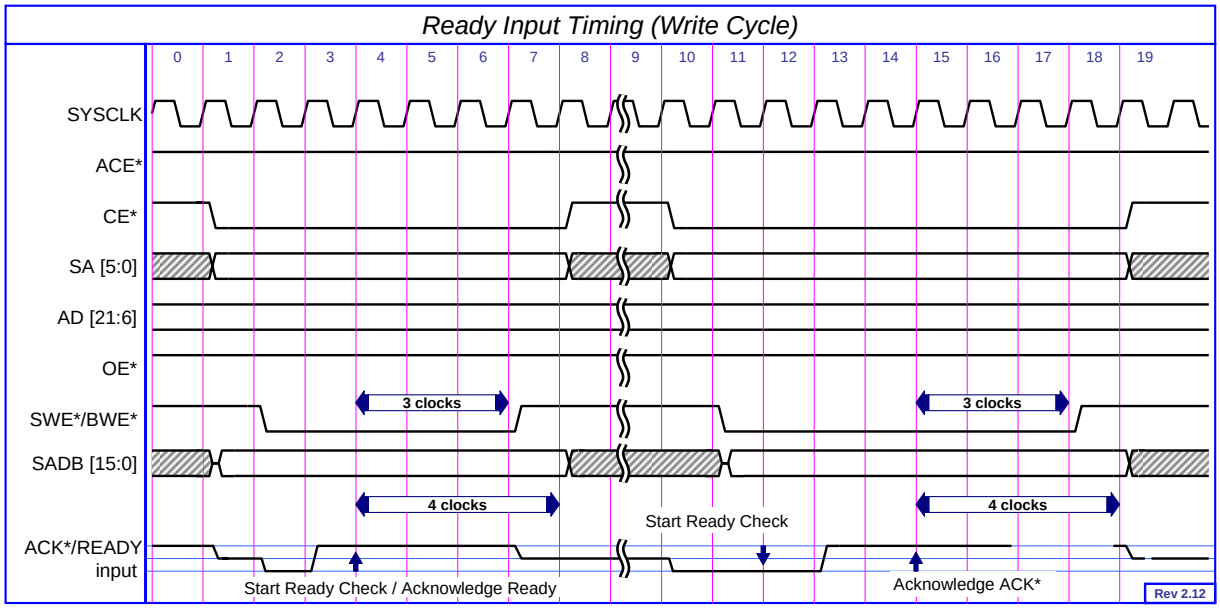


Figure 9-13 Ready Input Timing (Write Cycle)

9.2.9. Clock Options

External devices connected to the external bus can use the SYSCLK signal as the clock. The SYSCLK signal clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/3, 1/4, 1/5, 1/6. The SA[4:3] signal are used to set this frequency during reset, and the setting is reflected in the SYSCLK Division Ratio field (CCFG.SYSSP) of the Chip Configuration Register.

The operation reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel independent of the SYSCLK signal clock frequency: 1/3, 1/4, 1/5, 1/6. The external signal of the External Bus Controller operates synchronous to this operation clock. The Bus Speed field (EBCCRn.SP) of the External Bus Channel Control Register sets this frequency.

Please set the same value as CCFG.SYSSP to EBCCRn.SP when the external device uses the SYSCLK signal. If these two values do not match, then the channel, the operation reference clock, and the SYSCLK signal will no longer be synchronous and will not operate properly.

9.2.10. ISA Mode (16-bit only)

TX4939 supports ISA I/O space access. Since the pins used in ISA mode are multiplexed pins, select ISA before use of these pins.

9.2.10.1. Address space

In the channel set to EBCCRn.ISA=1, 64 Kbyte from the start address of the channel is used for the ISA I/O space. (Refer to Figure 9-14). When access to the ISA I/O space is performed, IOR* and IOW* control signals are valid, and OE* and SWE* are invalid. Other control signals are usable, that is the same as the case other than the ISA mode.

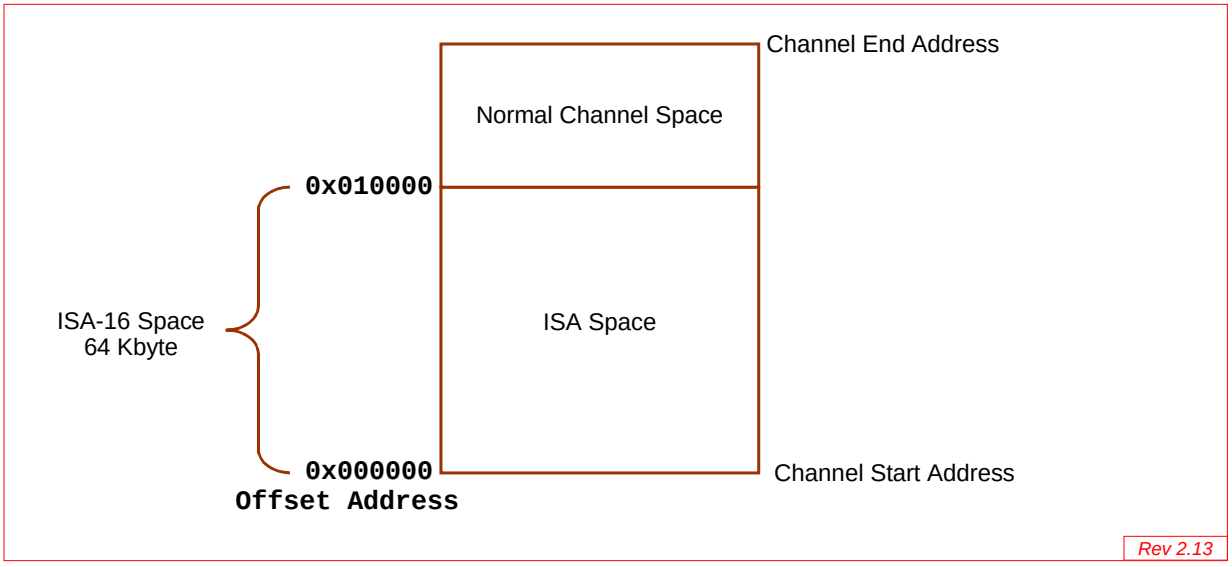


Figure 9-14 Physical Address Mapping of the ISA Spaces

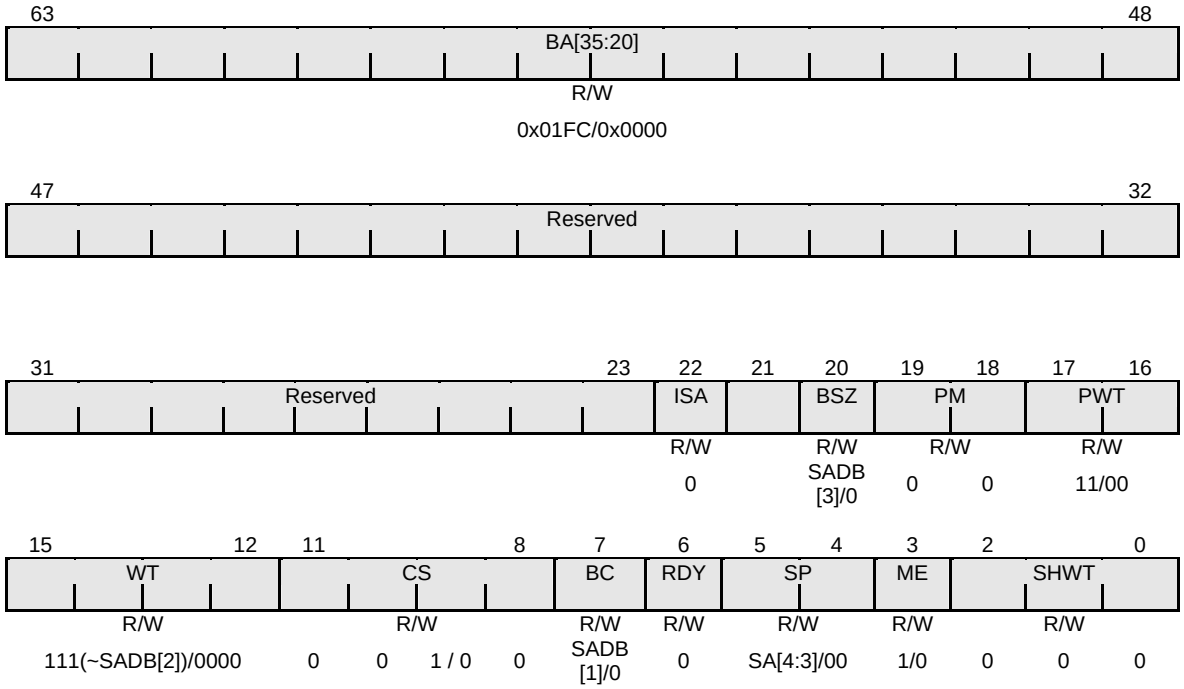
9.3. Register Detail

Table 9-7 External Bus Controller (EBUSC) Registers

Offset Address	Bit Width	Register Symbol	Register Name
0x9000	64	EBCCR0	E-Bus Channel Control Register 0
0x9008	64	EBCCR1	E-Bus Channel Control Register 1
0x9010	64	EBCCR2	E-Bus Channel Control Register 2
0x9018	64	EBCCR3	E-Bus Channel Control Register 3
0x9020	64	EBCCR4	Reserved
0x9028	64	EBCCR5	Reserved
0x9030	64	EBCCR6	Reserved
0x9038	64	EBCCR7	Reserved

9.3.1. External Bus Channel Control Register (EBCCRn)

Channel 0 can be used as Boot memory. Therefore, the default is set by the Boot signal. Channels 1 - 3 have the same register configuration as Channel 0, but they have different defaults than Channel 0.
When the EBCCRn is programmed using a sequence of 32-bit store instructions, the base address in the high-order 32-bit portion of the register must be written first, followed by the Master Enable bit in the low-order 32-bit portion.



Only in the case of Channel 0 are fields with different defaults in the “Channel 0/Other channel” state. SADB[ ] represents the corresponding Data[ ] signal value when the RESET* signal is de-asserted. SA[ ] represents the corresponding ADDR[ ] signal value when the RESET* signal is de-asserted.

Table 9-8 External Bus Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
63:48	BA[35:20]	Base Address	External Bus Control Base Address (Default: 0x01FC/0x0000) A physical address is used to specify the base address. The upper 16 bits [35:20] of the physical address are compared to the value of this field.	R/W
47:23		Reserved		—
22	ISA	ISA Mode	ISA Mode Set (Default: 0) Specifies the ISA-16 Mode. 0: ISA Mode Disable 1: ISA Mode Enable	R/W
47:21		Reserved		—
20	BSZ	Bus Width	External Bus Control Bus Size (Default: SADB[3]/0) Specifies the memory bus width. 0: 16-bit width 1: 8-bit width Note: SADB[3] is set to Channel 0 as the default.	R/W
19:18	PM Page Size	Page Mode	External Bus Control Page Mode Page Size (Default: 00) Specifies the Page mode (Page mode memory support) use and page size. 00: Normal mode 01: 4-page mode 10: 8-page mode 11: 16-page mode	R/W
17:16	PWT	Page Mode Wait time	External Bus Control Page Mode Wait Time (Channel 0=2'b11 / 00) Specifies the wait cycle count during Burst access when in the Page mode. 00: 0 wait cycles 10: 2 wait cycles 01: 1 wait cycle 11: 3 wait cycles Specifies a wait cycle count from 0 to 62 that matches WT when in the Normal mode or Ready mode. (See the WT item.)	R/W
15:12	WT	Normal Mode Wait Time	External Bus Control Normal Mode Wait Time (Default: [111 (~SADB[2])/[0000]) Specifies the wait cycle count in the first cycle of a Single Cycle or Burst access. Specifies the following wait cycle count when in the Page mode. 0000: 0 wait cycles 0100: 4 wait cycles 1000: 8 wait cycles 1100: 12 wait cycles 0001: 1 wait cycle 0101: 5 wait cycles 1001: 9 wait cycles 1101: 13 wait cycles 0010: 2 wait cycles 0110: 6 wait cycles 1010: 10 wait cycles 1110: 14 wait cycles 0011: 3 wait cycles 0111: 7 wait cycles 1011: 11 wait cycles 1111: 15 wait cycles Specifies a wait cycle count from 0 to 62 that matches PWT when in a mode other than the Page mode. PWT[1:0]: WT[3:0] 000000: 0 wait cycles 000001: 1 wait cycles 000011: 3 wait cycles : : 001110: 14 wait cycles 001111: 15 wait cycles : : 011110: 30 wait cycles 011111: 31 wait cycles 111111: External ACK mode Note 1: Value that is the reverse of SA[2] is set to the LSB of Channel 0 as the default. Note 2: If PWT:WT is set to 0x3f when PM = 00 and RDY = 0, the external bus enters the ACK* Input mode (External ACK mode) without the wait cycle count for the ACK* output being the maximum value. Note 3: WT[0] is used to select Dynamic/Static ACK*/Ready mode when in the Ready mode. Therefore, the Wait cycle count is an even number. Note 4: Set the WT wait cycle count to a value greater than the PWT Wait cycle count when in the Page mode.	R/W

Table 9-8 External Bus Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
11:8	CS	Channel Size	External Bus Control Channel Size (Default: 0010/0000) Specifies the channel memory size. 0000: 1 MB 0001: 2 MB 0010: 4 MB 0011: 8 MB 0100: 16 MB 0101: 32 MB 0110: 64 MB 0111: 128 MB 1000: 256 MB 1001: 512 MB 1010 – 1111: Reserved * The channel memory size can be set up to 512 MB when the memory bus width is 16 bits, or up to 256 MB when the memory bus width is 8 bits. No size larger than this can be set.	R/W
7	BC	Byte Control	External Bus Byte Control (Default:SADB[1]/0) Specifies whether to use the BWE*[1:0] signal as an asserted Byte Write Enable signal (BWE*[1:0]) only during a Write cycle, or to use it as an asserted Byte Enable signal (BE*[1:0]) that is asserted during both Read and Write cycles. 0: Byte Enable (BE *[1:0]) 1: Byte Write Enable (BWE*[1:0]) Note: SADB[1] is set to Channel 0 as the default.	R/W
6	RDY	Ready Input Mode	External Bus Control Ready Input Mode (Default: 0) Specifies whether to use the Ready mode. 0: Disable the Ready mode. 1: Enable the Ready mode. Note: The Ready mode cannot be used when the Page mode is selected.	R/W
5:4	SP	BOOTSP	External Bus Control Bus Speed (Default: SA[4:3] / 00) Specifies the External Bus speed. 00: 1/4 speed (1/4 of the GBUSCLK frequency) 01: 1/3 speed (1/3 of the GBUSCLK frequency) 10: 1/5 speed (1/5 of the GBUSCLK frequency) 11: 1/6 speed (1/6 of the GBUSCLK frequency)	R/W
3	ME	Master Enable	External Bus Control Master Enable (Default: 1 / 0) Enables a channel. 0: Disable channel 1: Enable channel Note: set to 1 for Channel 0 as the default.	R/W
2:0	SHWT	Set Up/Hold Wait Time	External Bus Control Setup/Hold Wait Time (Default: 000) Specifies the wait count when switching between the Address and Chip Enable signal, or the Chip Enable Signal and Write Enable/Output Enable signal. * 000: Disable 100: 4 wait cycles 001: 1wait cycle 101: 5 wait cycles 010: 2 wait cycles 110: 6 wait cycles 011: 3 wait cycles 111: 7 wait cycles * Set this bit field to “0” when using it in the Page mode or when performing Burst access.	R/W

9.4. Timing Diagrams

Please take the following points into account when referring to the timing diagrams.

The clock frequency of the SYSCLK signal can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/3, 1/4, 1/5, or 1/6. Also, the operating reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel: 1/3, 1/4, 1/5, or 1/6. (See 9.2.9.) The timing diagrams indicate the SYSCLK signal clock frequency and channel operating reference clock frequency as being equivalent.

Both the BWE* signal and BE* signal are indicated in all timing diagrams. The setting of the Channel Control Register (EBCCRn) determines whether the BWE* pin will function as BWE* or BE*.

All Burst cycles in the timing diagrams illustrate examples in which the address increases by increments of 1 starting from 0. However, cases where the CWF (Critical Word First) function of the TX49 core was used or the decrement burst function performed by the DMA Controller was used are exceptions.

The timing diagrams display each clock cycle currently being accessed using the symbols described in the following table.

SWn	Normal Wait Cycles
PWn	Page Wait Cycles
ASn	Set-up Time from SHWT Address Validation to CE Fall
CSn	Set-up Time from SHWT CE Fall to OE/SWE Fall
AHn	Hold Time from SHWT CE Rise to Address Change
CHn	Hold Time from SHWT OE/SWE Rise to CE Rise
ESn	Synch Cycles of the External Input Signal
ACEn	Address Clock Enable Cycles
Sn	Other Cycles

Shaded areas in the diagrams are undefined values.

9.4.1. ACE* Signal

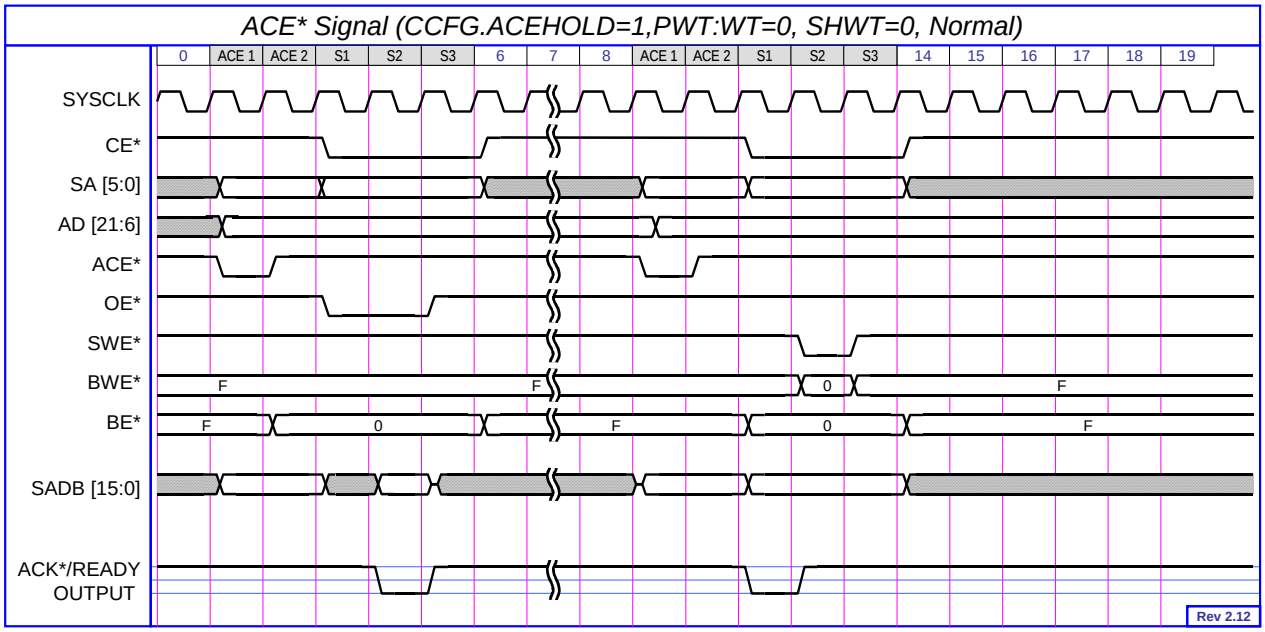


Figure 9-15 ACE Signal (CCFG.ACEHOLD=1, PWT: WT=0, SHWT=0, Normal)*

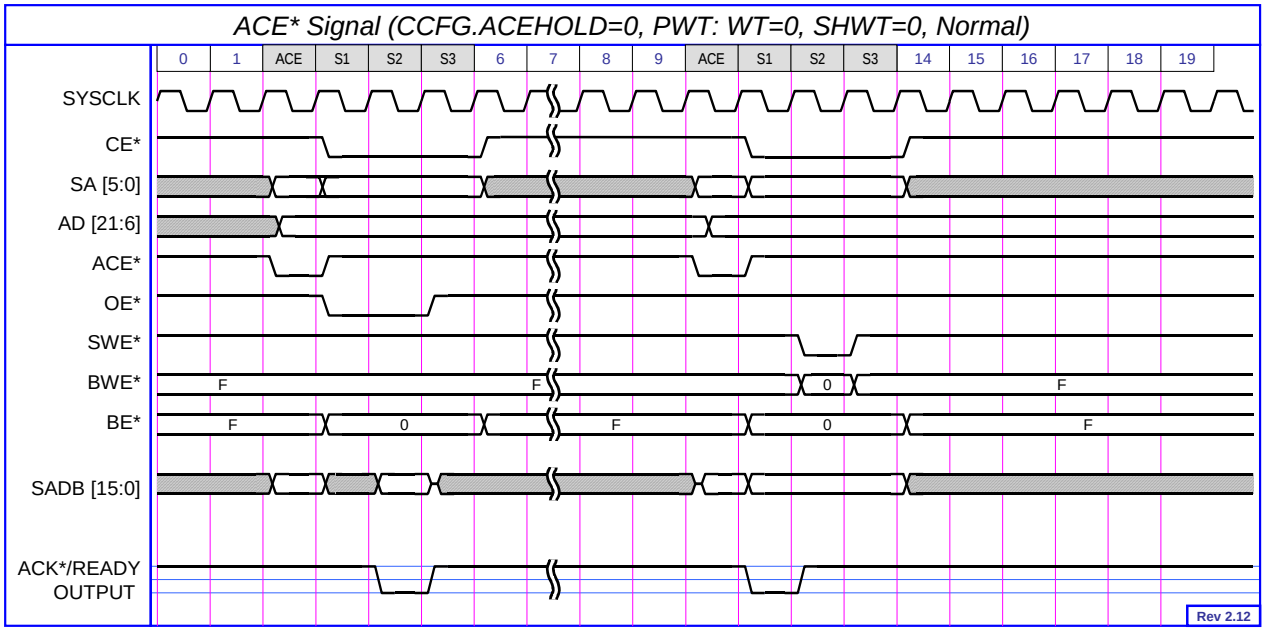


Figure 9-16 ACE Signal (CCFG.ACEHOLD=0, PWT: WT=0, SHWT=0, Normal)*

9.4.2. Normal Mode Access (Single, 16-bit bus)

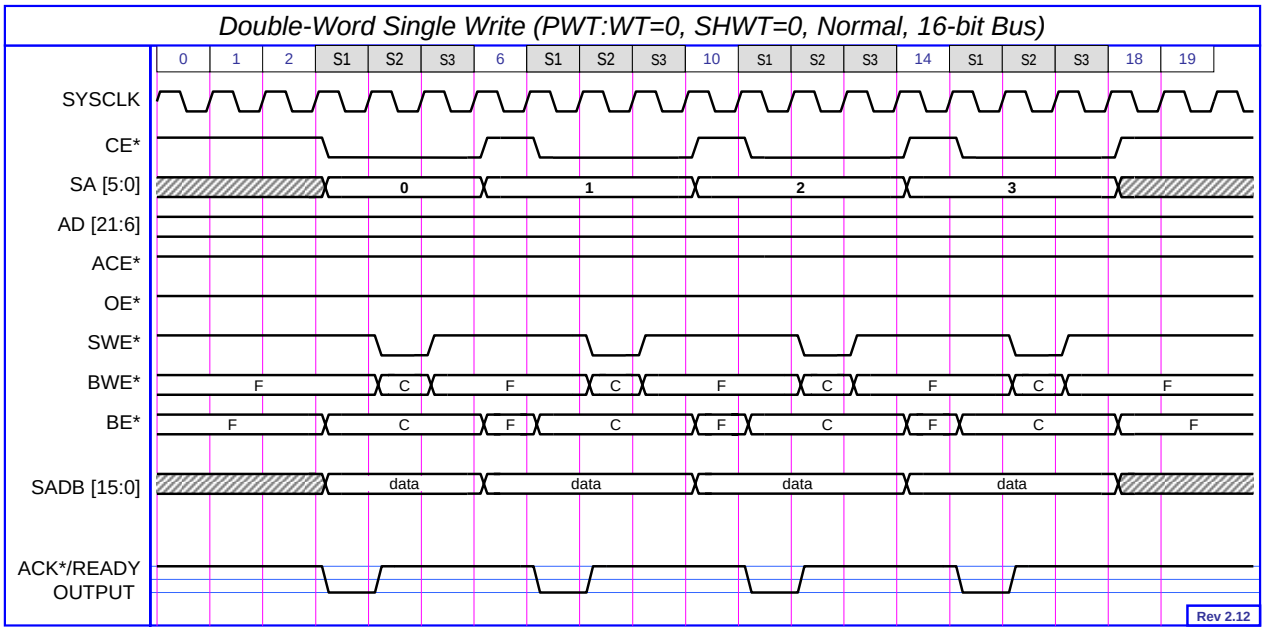


Figure 9-17 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

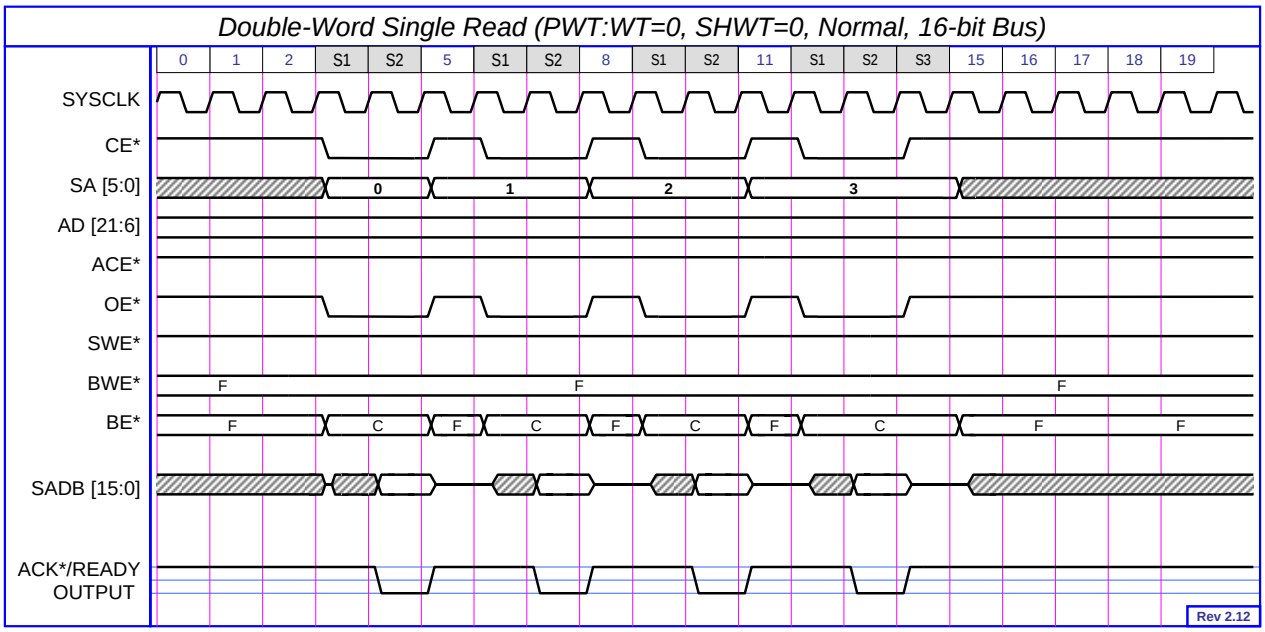


Figure 9-18 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

9

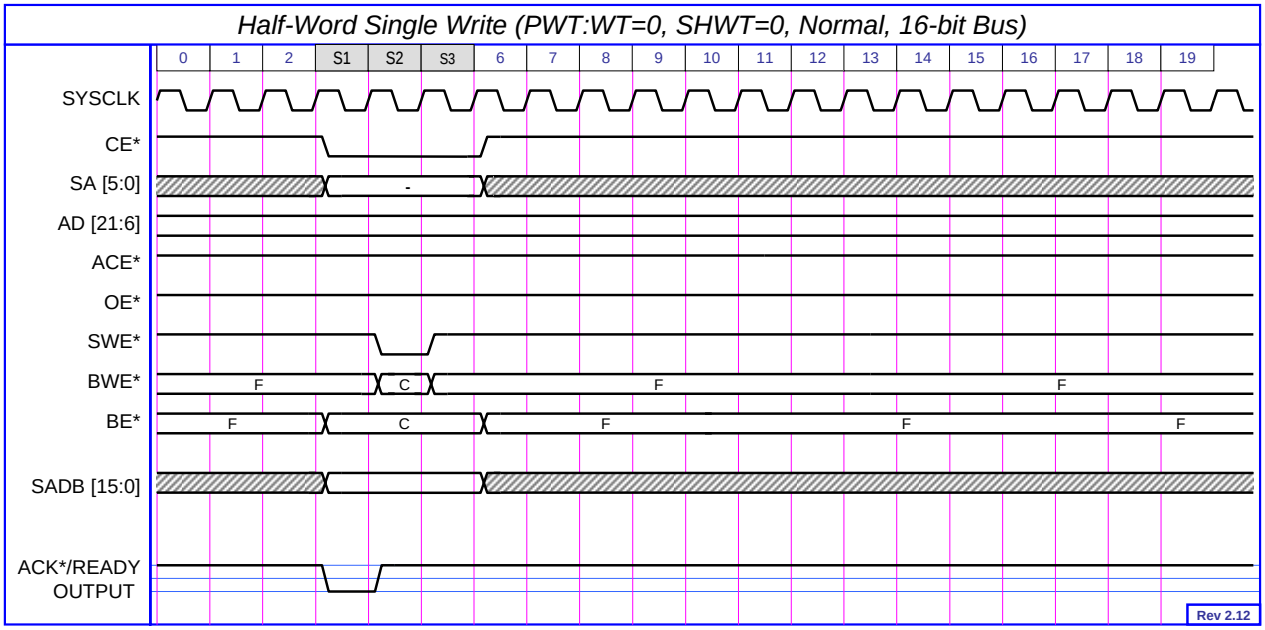


Figure 9-19 Half-word Single Write/Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

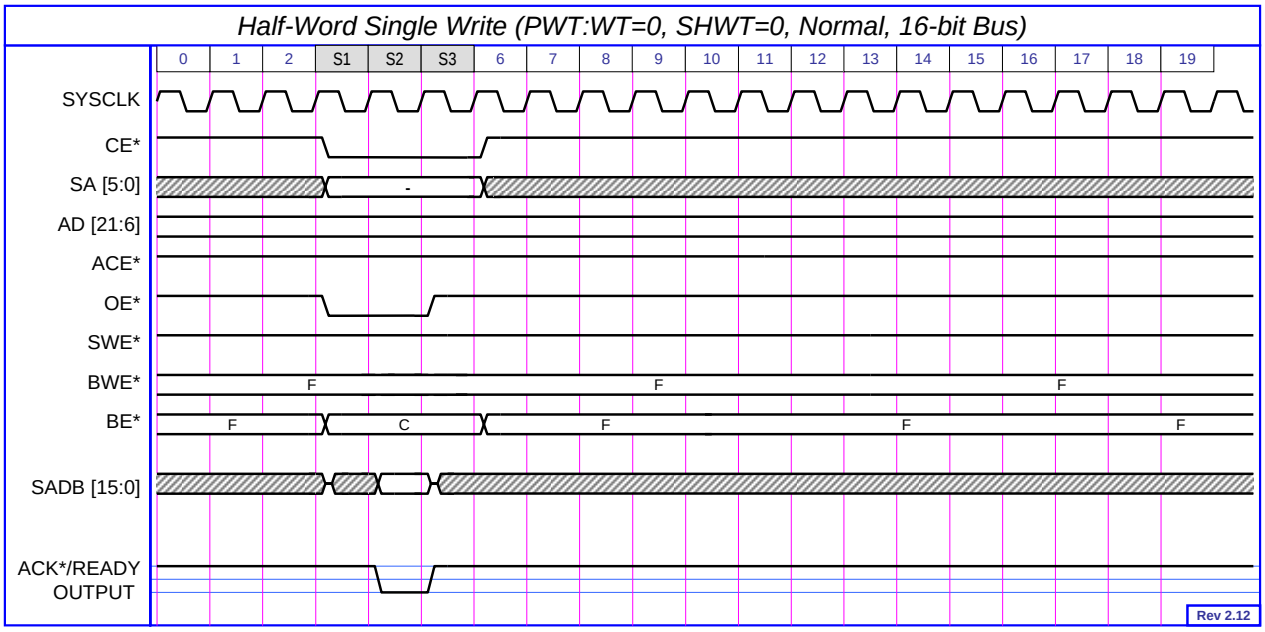


Figure 9-20 Half-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

9.4.3. Normal Mode Access (Burst, 16-bit Bus)

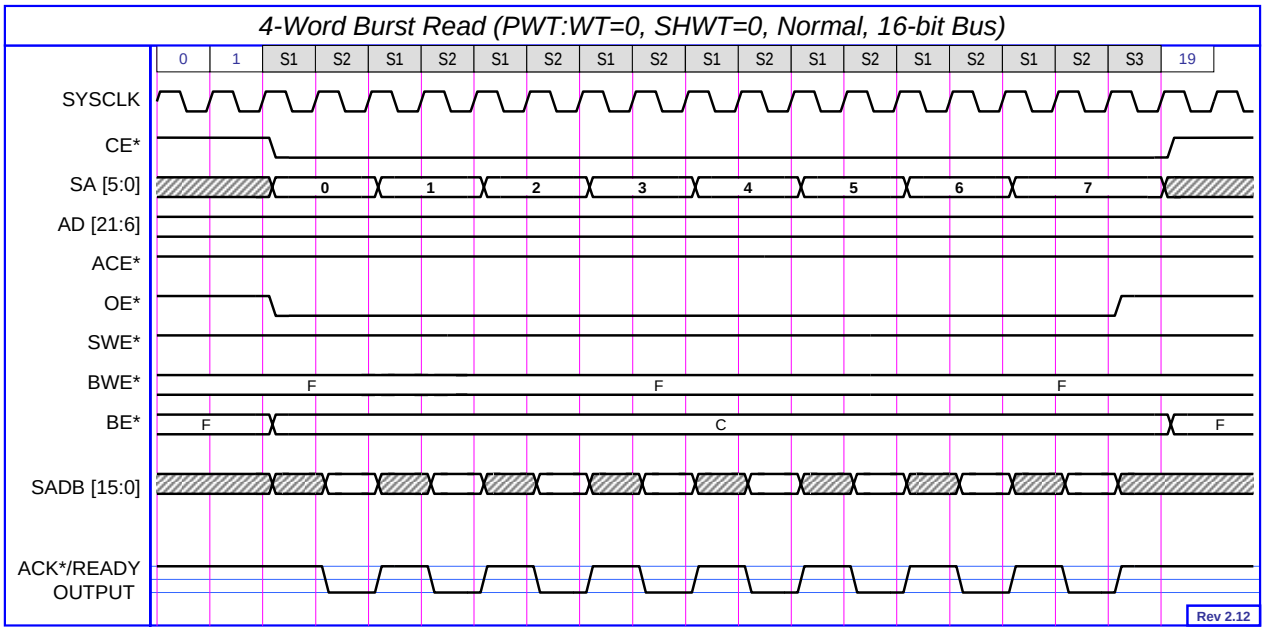


Figure 9-21 4-word Burst Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

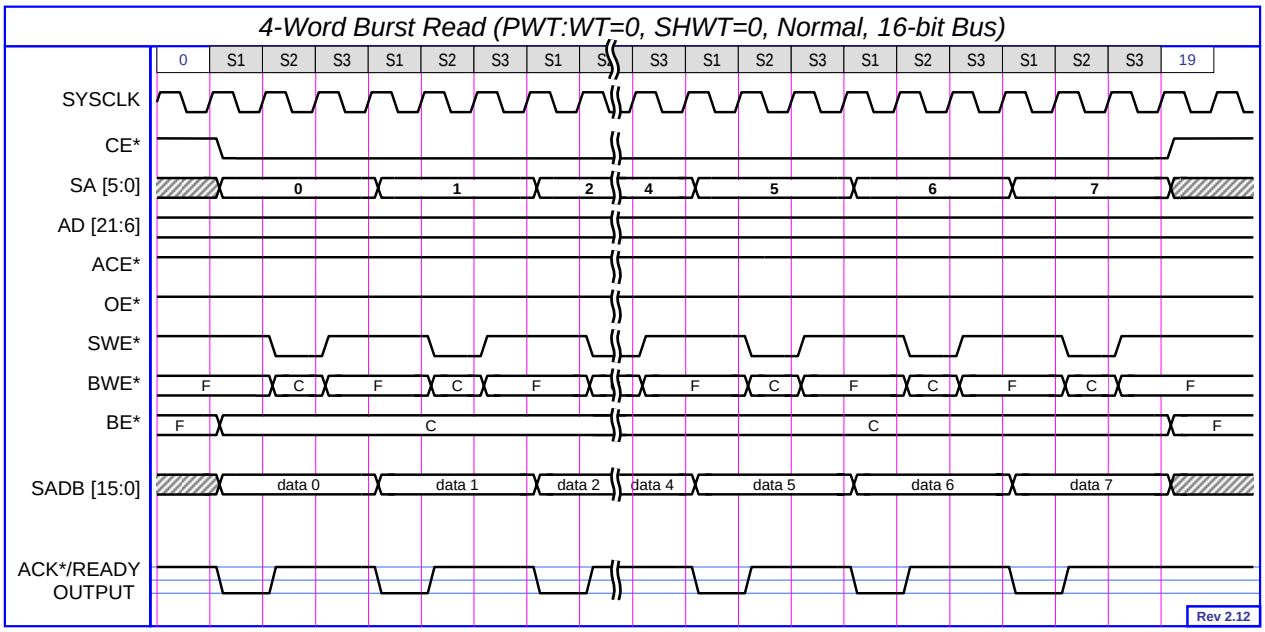


Figure 9-22 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)
(For 1/3 clock speed – SP=01, SADB setup time to assertion of SWE is ½ SYSCLK)

9.4.4. Normal Mode Access (Single, 8-bit Bus)

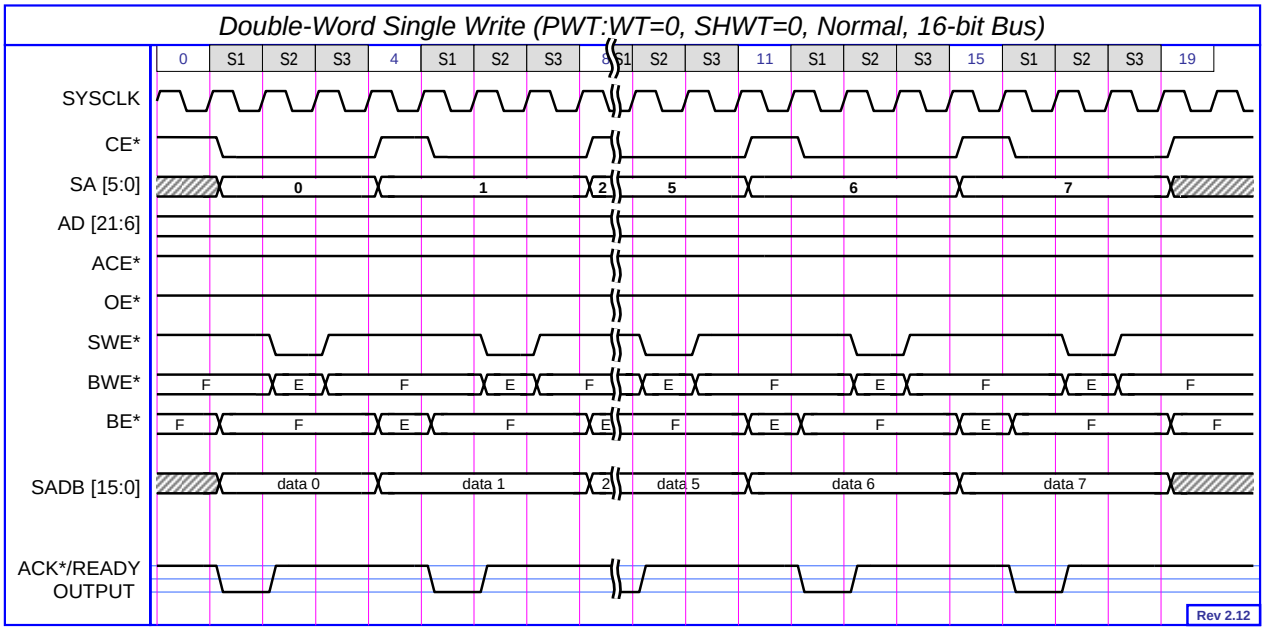


Figure 9-23 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

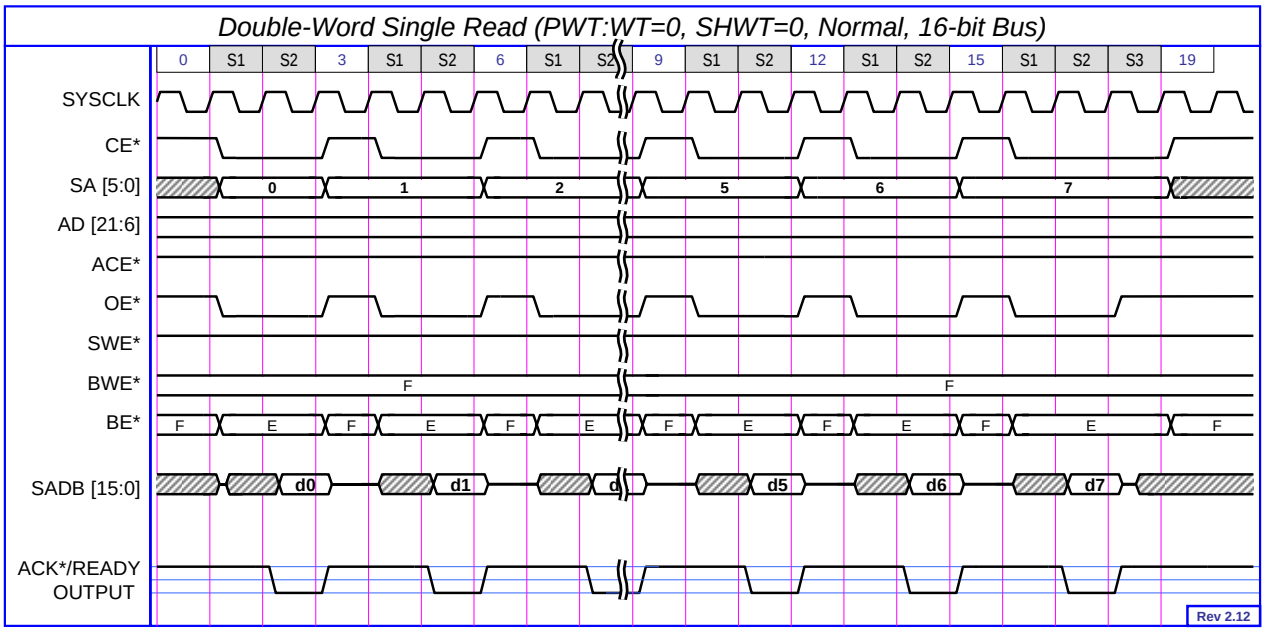


Figure 9-24 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

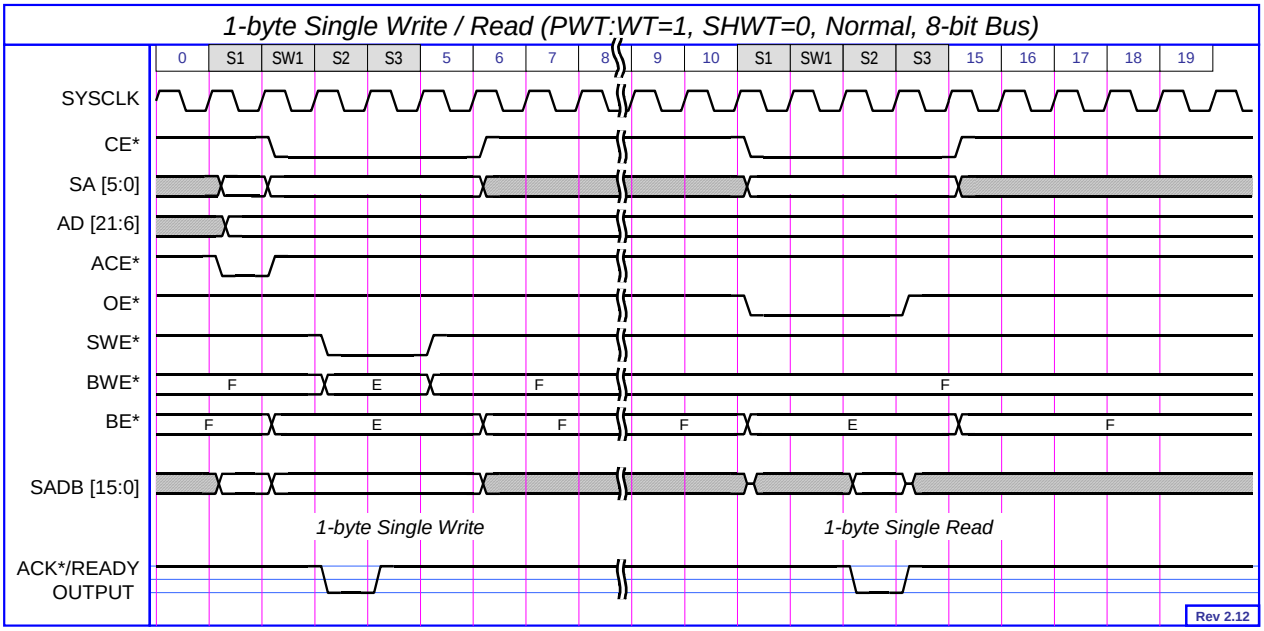


Figure 9-25 1-byte Single Write / Read (PWT: WT=1, SHWT=0, Normal, 8-bit Bus)

9.4.5. Normal Mode Access (Burst, 8-bit Bus)

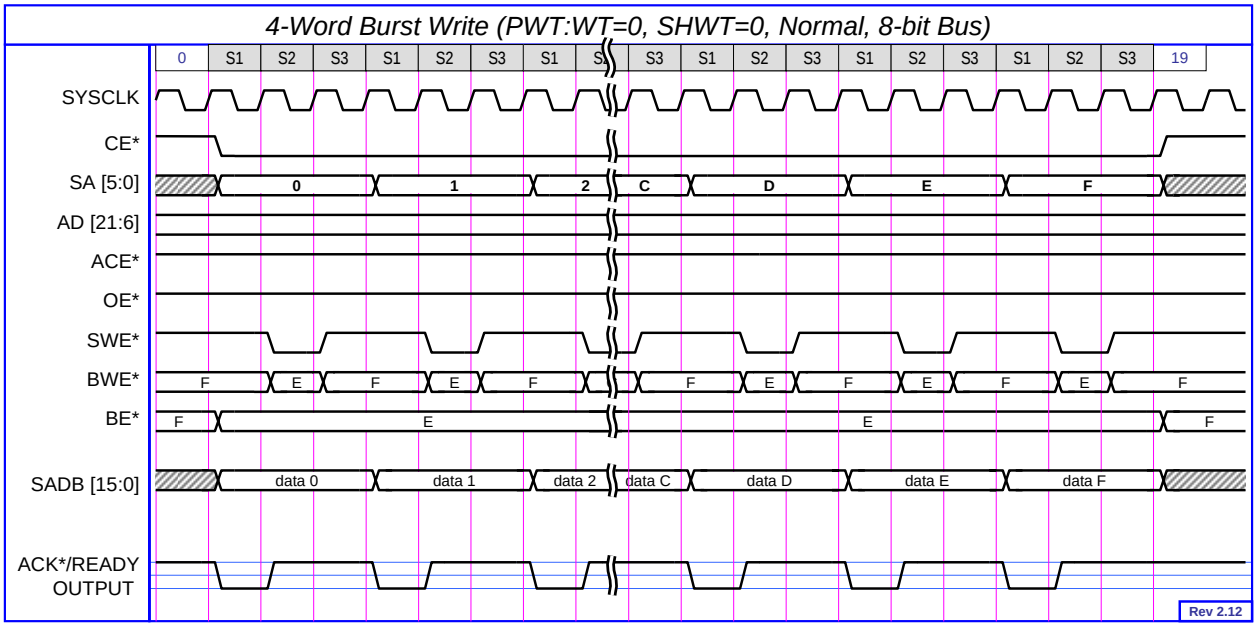


Figure 9-26 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)
(For 1/3 clock speed – SP=01, SADB setup time to assertion of SWE is ½ SYSCLK)

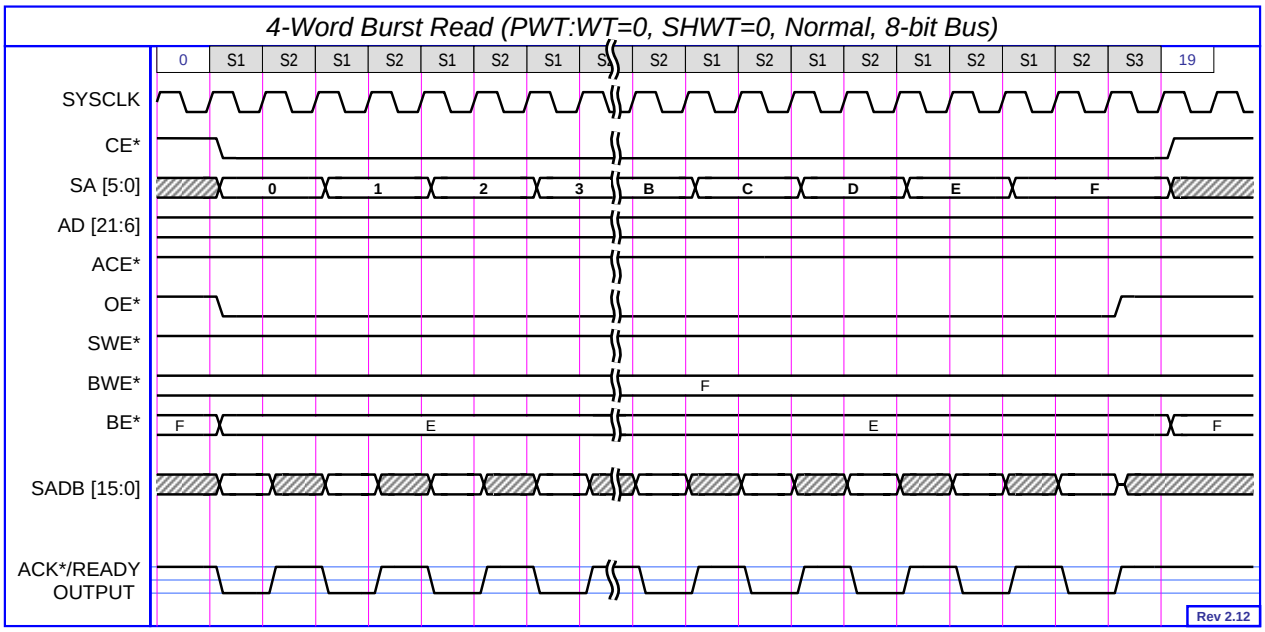


Figure 9-27 4-word Burst Read (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

9.4.6. Page Mode Access (Burst, 16-bit Bus)

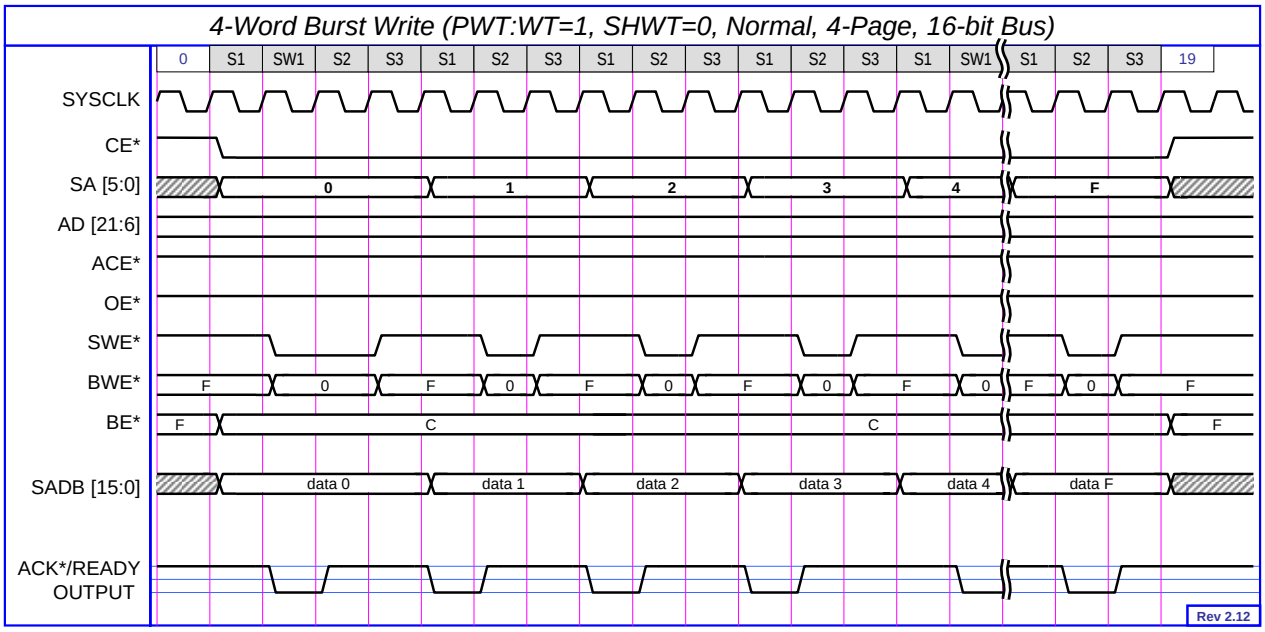


Figure 9-28 4-word Burst Write (WT=1, PWT=0, SHWT=0, 4-page, 16-bit Bus)
(For 1/3 clock speed – SP=01, SADB setup time to assertion of SWE is ½ SYSCLK)

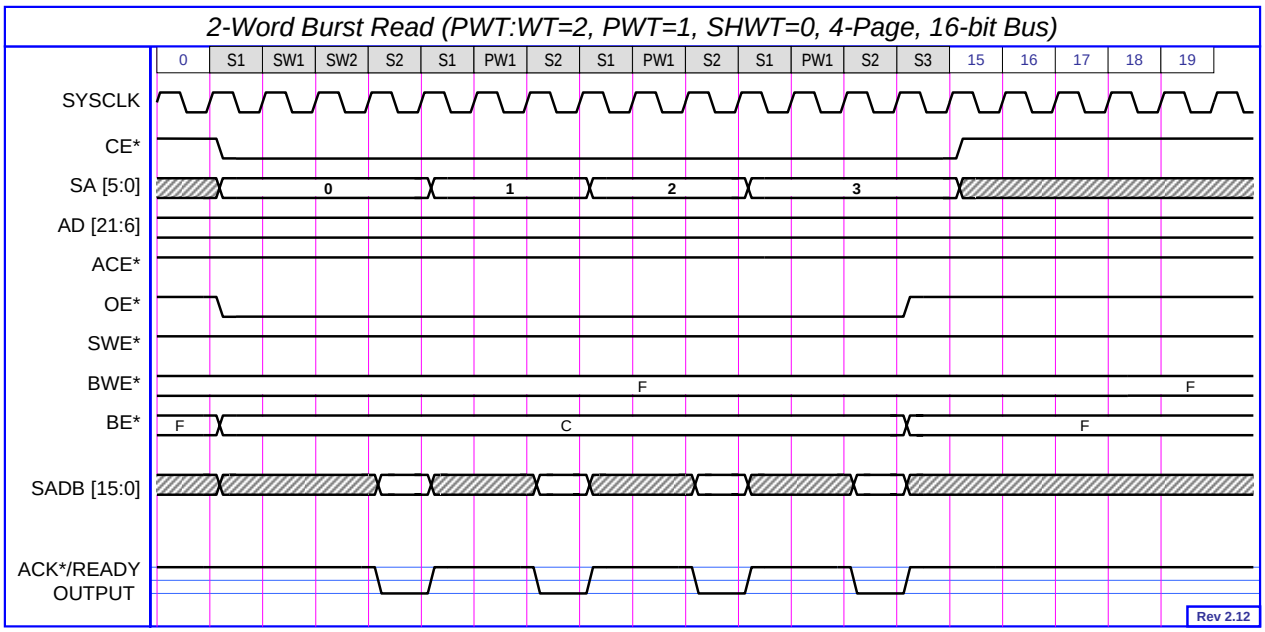
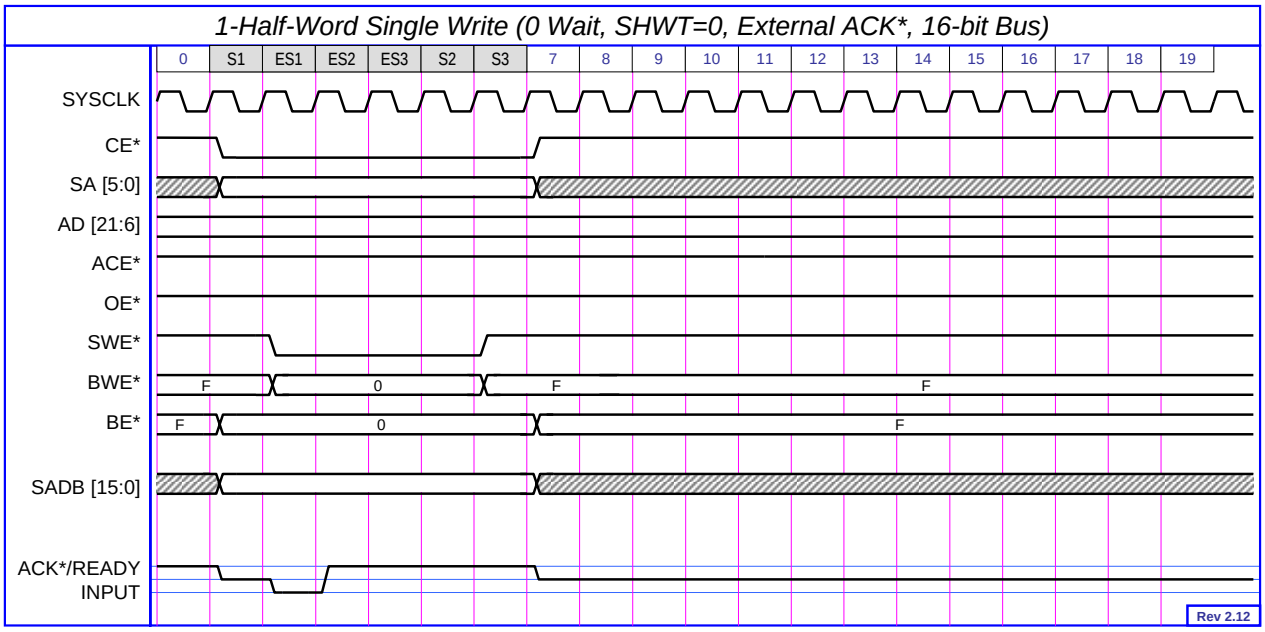


Figure 9-29 2-word Burst Read (WT=2, PWT=1, SHWT=0, 4-page, 16-bit Bus)

9.4.7. External ACK Mode Access (16-bit Bus)



Note 1: The TX4939 sets the ACK* signal to High Impedance in the S1 State.

Note 2: External devices drive the ACK* signal to Low (assert the signal) until the ES1 State.

Note 3: External devices drive the ACK* signal to High (deassert the signal) in the ES2 State. If an external device is late in asserting ACK*, then the Wait State is inserted for the amount of time the external device is late. If a certain condition is met, it is okay for the ACK* signal to be driven to Low for 1 clock cycle or more. See 9.2.8.4 ACK* Input Timing (External ACK Mode) for more information.

Figure 9-30 1-half-word Single Write (0 Wait, SHWT=0, External ACK*, 16-bit Bus)

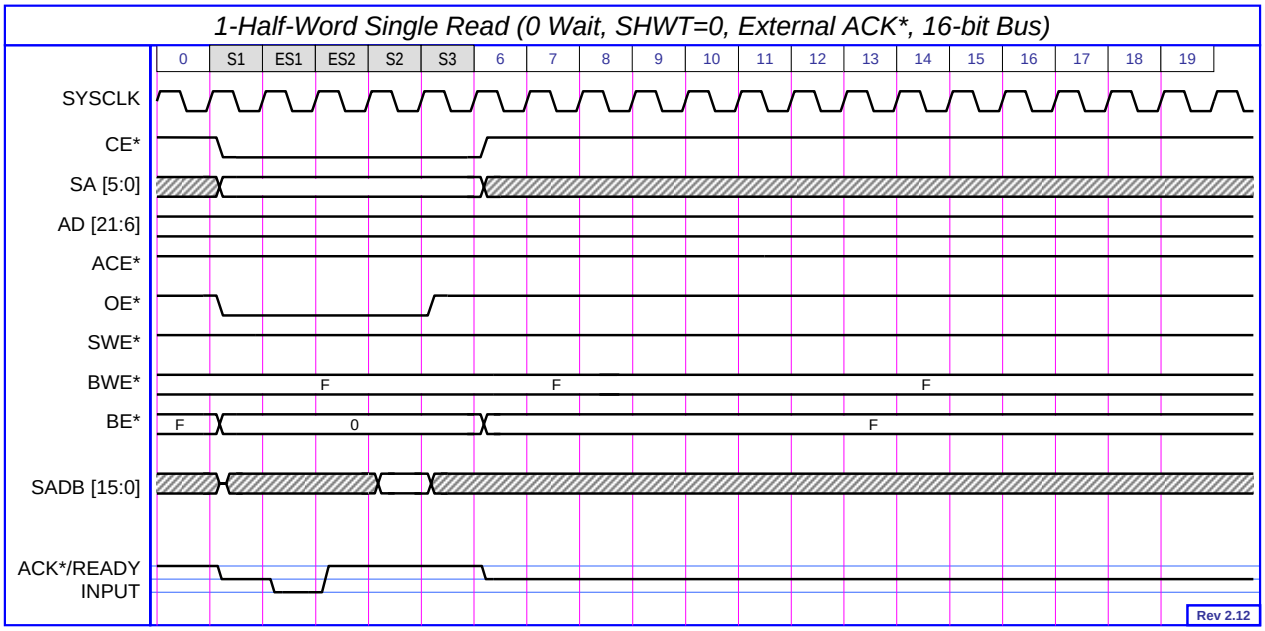


Figure 9-31 1-half-word Single Read (0 Wait, SHWT=0, External ACK*, 16-bit Bus)

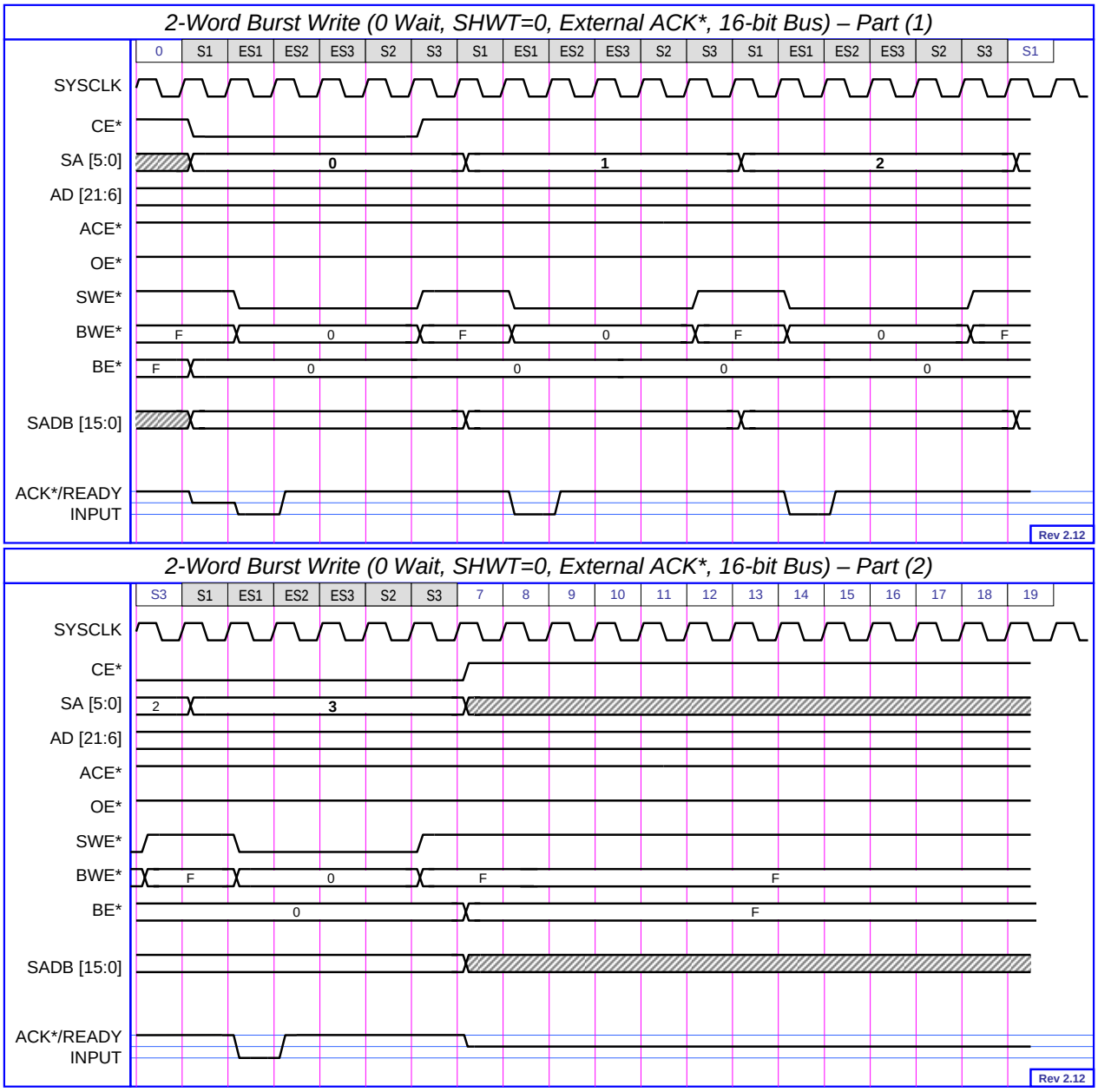


Figure 9-32 2-word Burst Write (0 Wait, SHWT=0, External ACK*, 16-bit Bus)
(For 1/3 clock speed – SP=01, SADB setup time to assertion of SWE is ½ SYSCLK)

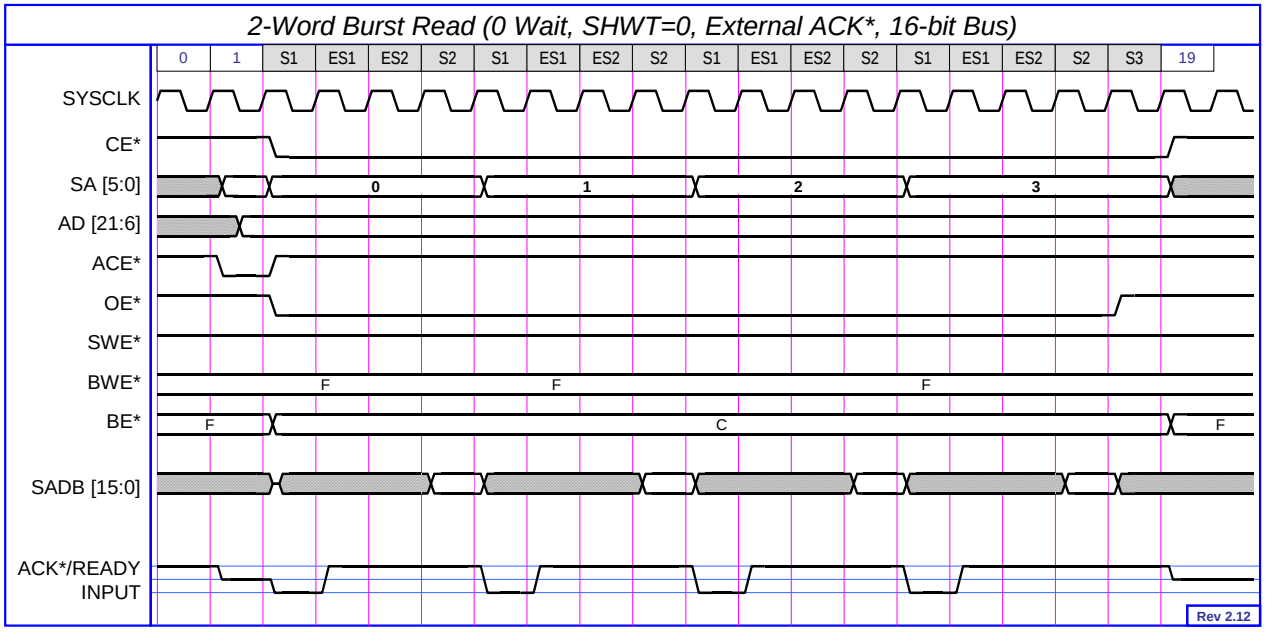


Figure 9-33 2-word Burst Read (0 Wait, SHWT=0, External ACK, 16-bit Bus)*

9

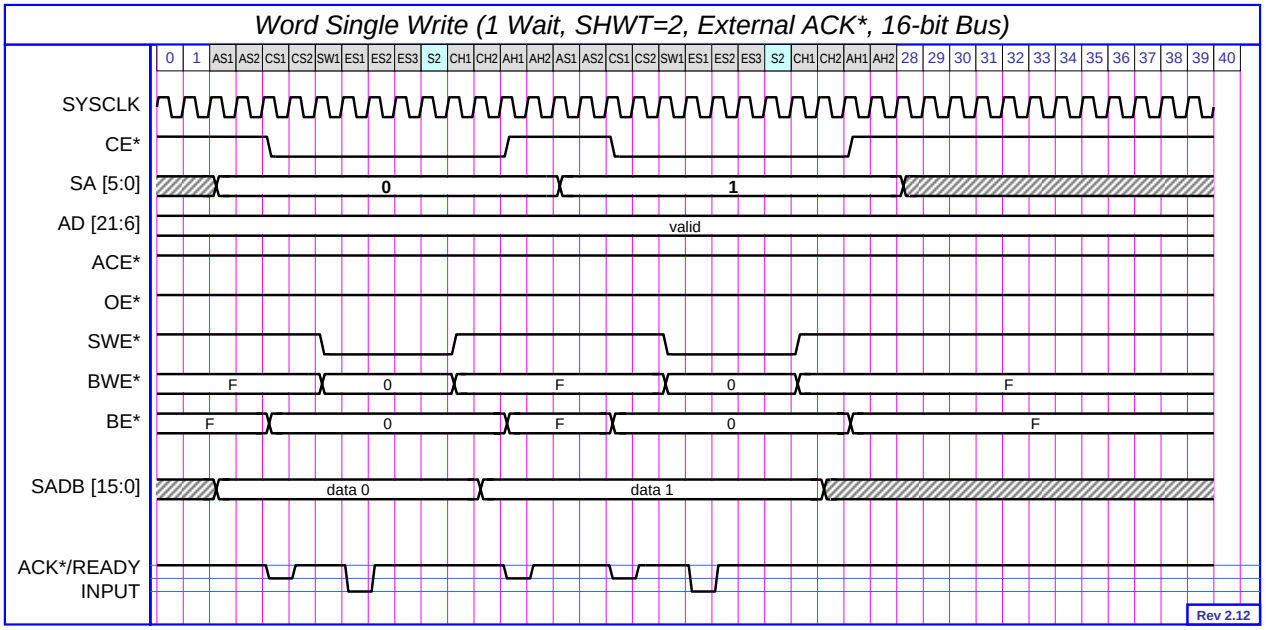


Figure 9-34 Word Single Write (1 Wait, SHWT=2, External ACK*, 16-bit Bus)

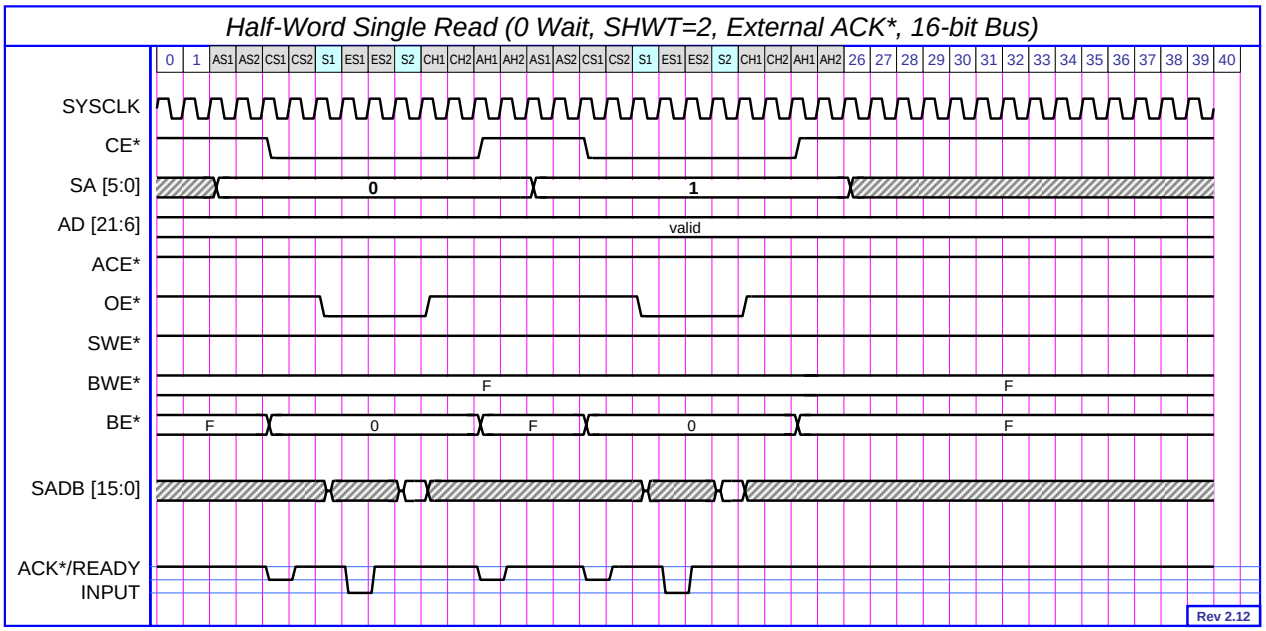


Figure 9-35 Half-word Single Read (0 Wait, SHWT=2, External ACK*, 16-bit Bus)

Note: TX4939 drives the ACK* signal when in the AH2, AS1, or AS2 State.

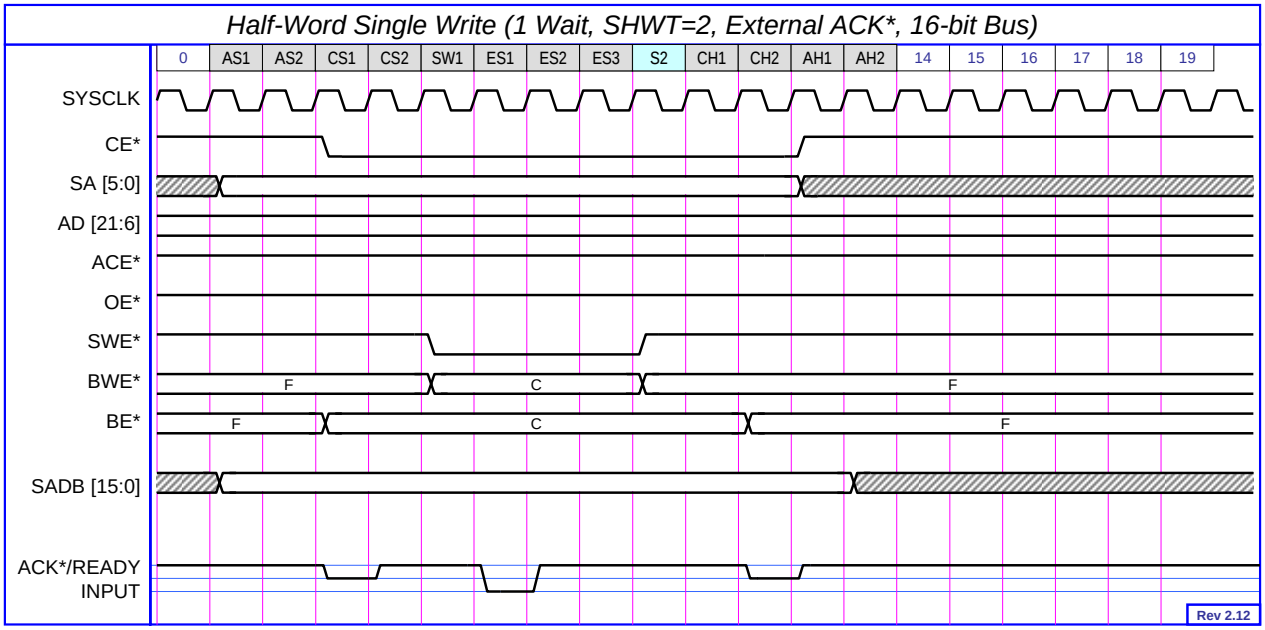


Figure 9-36 Half-word Single Write (1 Wait, SHWT=2, External ACK, 16-bit Bus)*

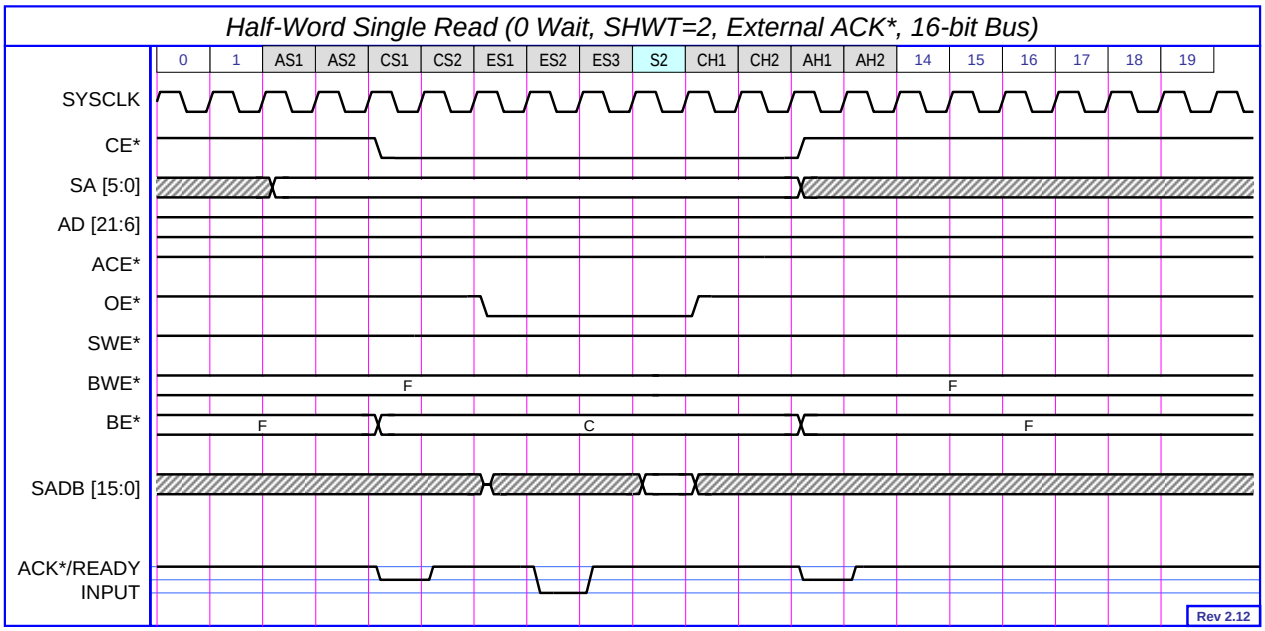


Figure 9-37 Half-word Single Read (0 Wait, SHWT=2, External ACK, 16-bit Bus)*

9.4.8. READY Mode Access (16-bit Bus)

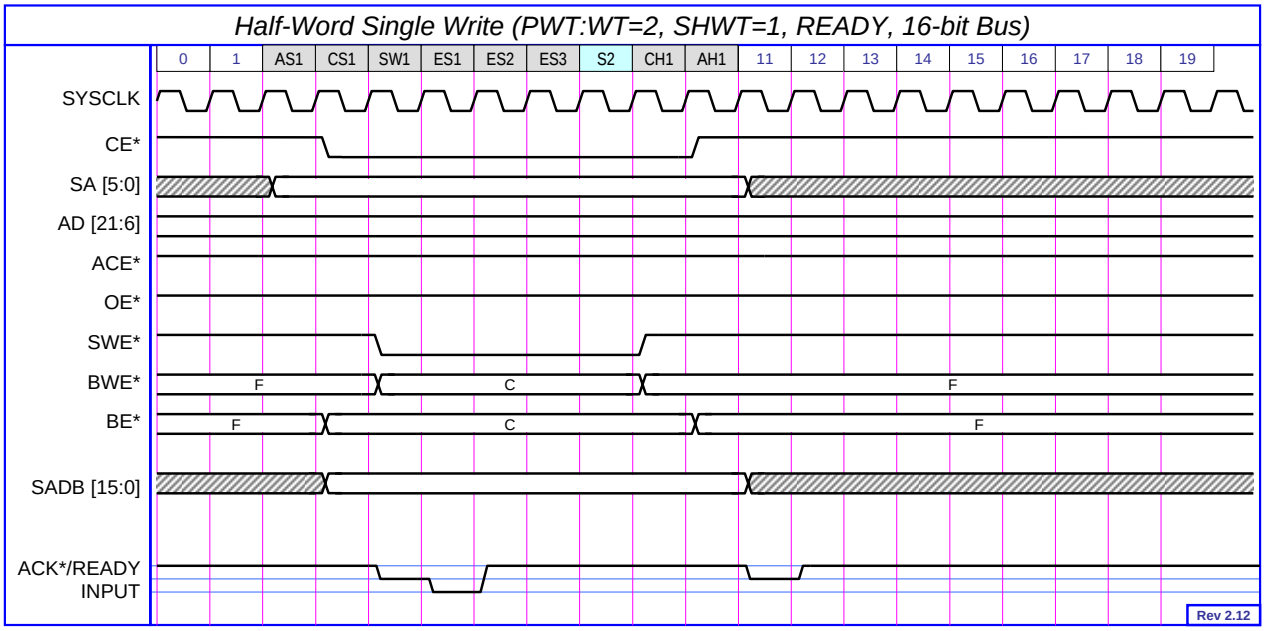


Figure 9-38 1-half-word Single Write (PWT: WT=2, SHWT=1, READY, 16-bit Bus)

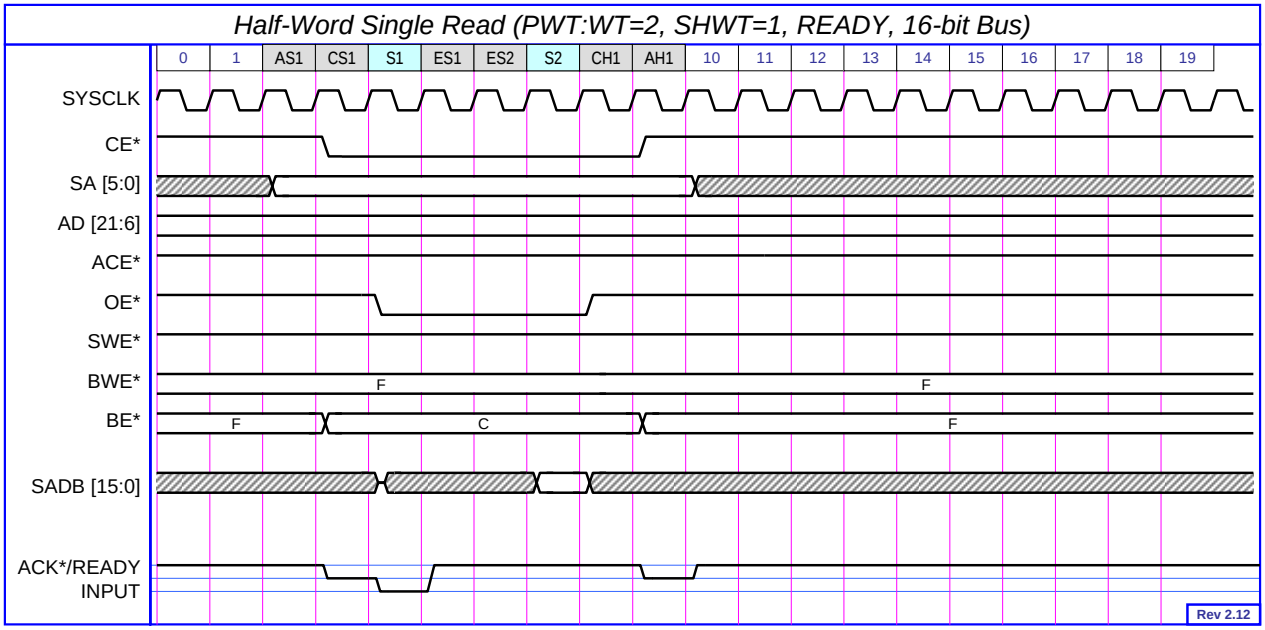


Figure 9-39 Half-word Single Read (PWT: WT=2, SHWT=1, READY, 16-bit Bus)

9.4.9. ISA IO Space Access (16-bit only)

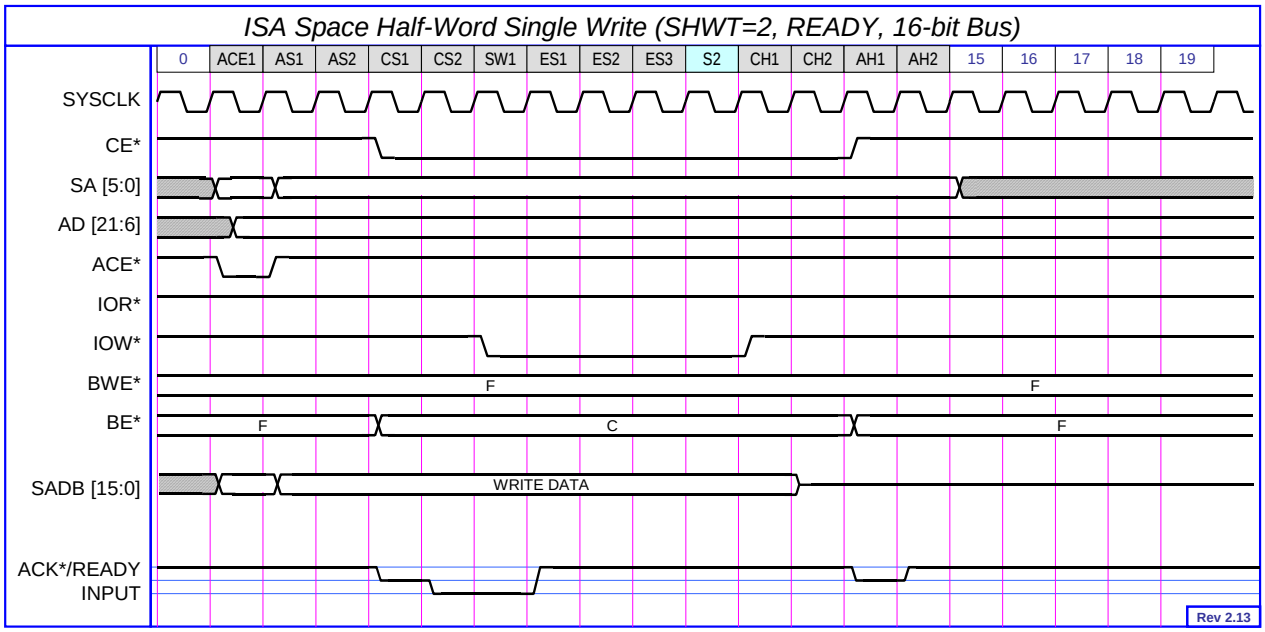


Figure 9-40 ISA Half-Word Single Write (SHWT=2, READY, 16-bit Bus)

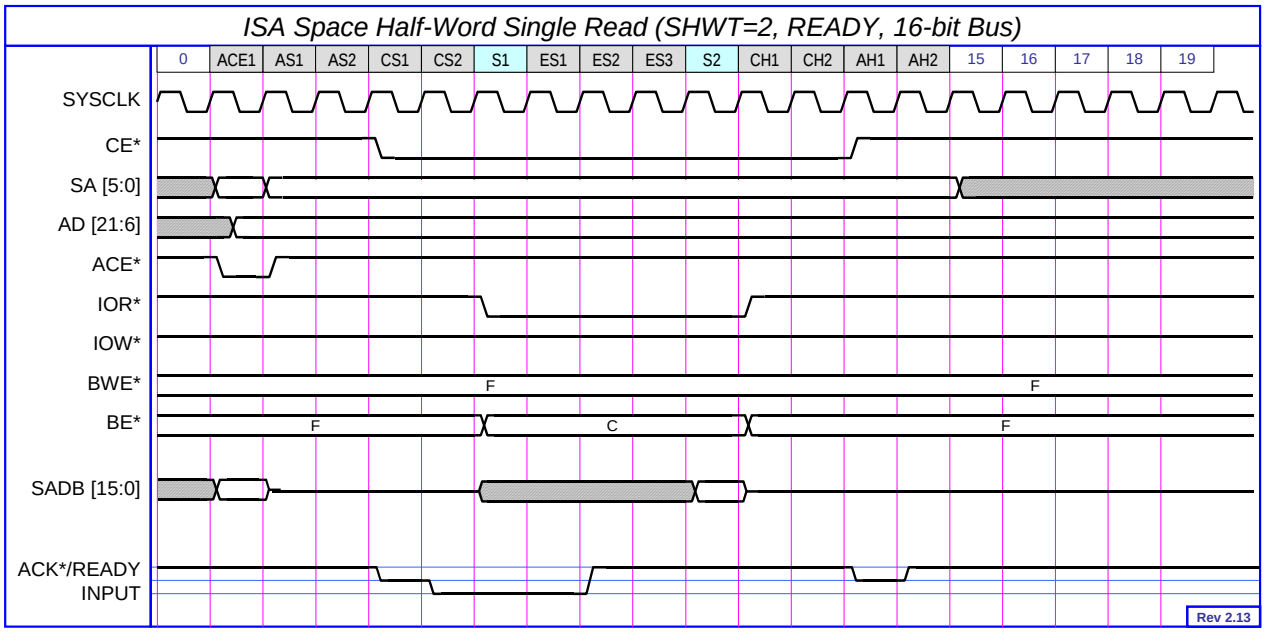


Figure 9-41 ISA Half-Word Single Read (SHWT=2, READY, 16-bit Bus)

Chapter 10. NAND Flash Memory Controller

10.1. Features

The TX4939XBG has a built-in NAND Flash Controller (NDFMC-II). The NDFMC-II generates the control signals required to interface with NAND flash memory. Also, the NDFMC-II performs ECC calculation and support DMA data transfer.

The NAND Flash Controller has the following features.

- *Indirect register access to NAND flash memory*
- *On-chip ECC calculation circuit*
- *DMA Operation for NAND READ data transfer*

10.2. Block diagram

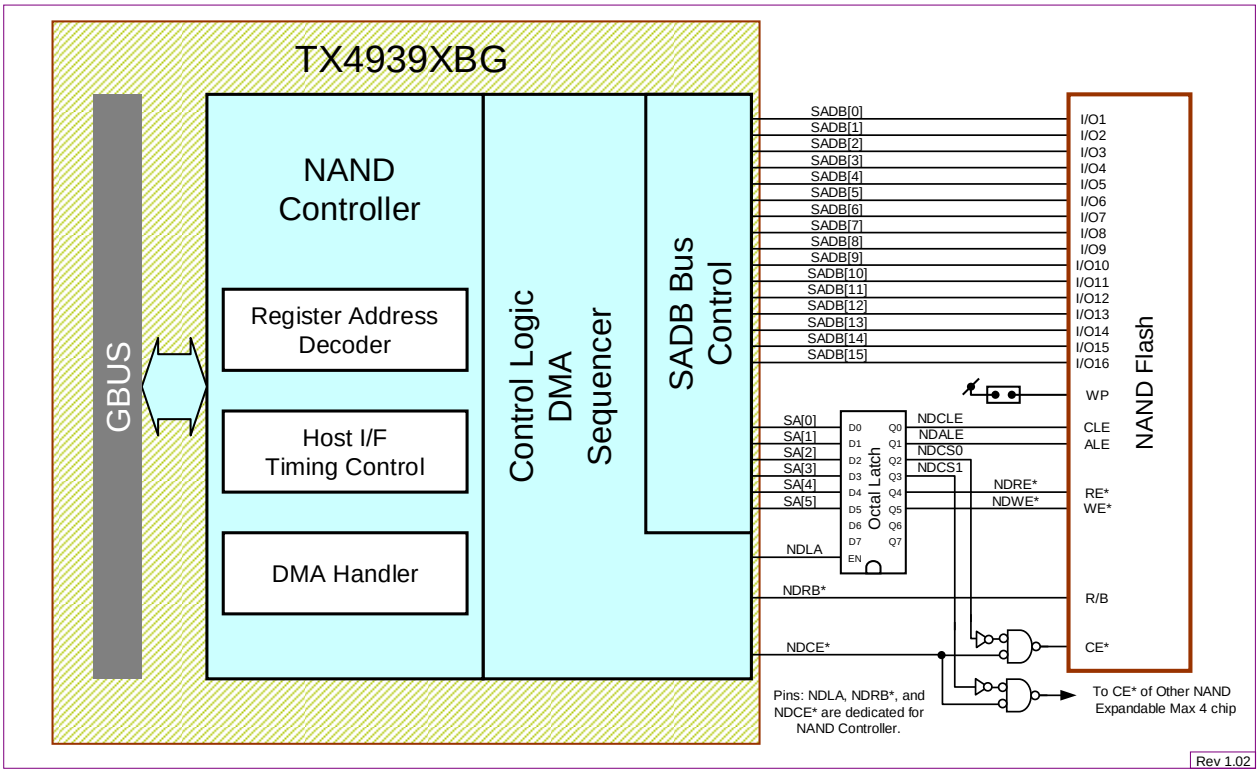


Figure 10-1 NAND Flash Memory Controller Block Diagram

10.2.1. Theory of Operation

NAND has 6 control signals, such as CE*, R/B, WE*, RE*, ALE, and CLE.

To reduce number of pins directly from TX4939XBG, external latch is used.
NDLA signal is the latch signal. As the R/B (Ready/Busy) signal is input, it uses one dedicated pin.
Since CE* signal needs to be controlled independently from other signal, it also use dedicated pin.

Eventually, the NDFMC (NAND Flash Memory Controller) shares SA [5:0] and SADB [15:0]. However, the assertion of ND_CE, which means certain NAND chip is online, does not hold these signals, i.e. SA, SADB, all the time.

Those buses will be used when NDFMC does data transfer or update the control signals, i.e. NDCLE, NDALE, NDRE*, and NDWE*. Actual arbitration with EBC might happen in this timing, which is associating with internal bus transactions.

10.3. Detailed Operation

10.3.1. Registers

Table 10-1 NDFMC Registers

Offset Address	Bit Width	Register Symbol	Register Name
0x5000	32	NDFDTR	NAND Flash Memory Data Transfer Register (R/W)
0x5008	32	NDFMCR	NAND Flash Memory Mode Control Register (R/W)
0x5010	32	NDFSR	NAND Flash Memory Status Register (Read) NAND Flash Memory Controller Reset (Write)
0x5018	32	NDFISR	NAND Flash Memory Interrupt Status Register(RO)
0x5020	32	NDFIMR	NAND Flash Memory Interrupt Mask Register (R/W)
0x5028	32	NDFSPR	NAND Flash Memory Strobe Pulse Width Register(R/W)

10.3.2. Convention for following explanation

NDFDTR and NDFMCR are the essential registers. Figure 10-2 shows field definition of them.

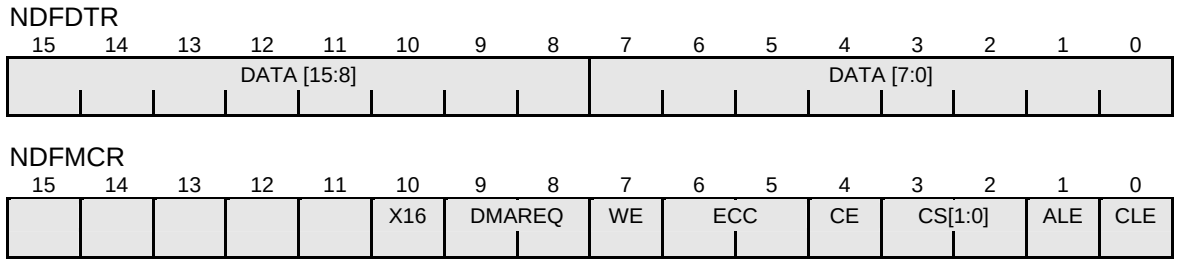


Figure 10-2 Field Definition of Registers NDFDTR and NDFMCR

Table 10-2 and Table 10-3 show Mnemonics used in this section. In addition, following expressions mean writing the parameter in to corresponding register.

```
$NDFMCR = CLE | CE ;  // Write 0x0011 to register NDFMCR
```

Table 10-2 Mnemonic Command Parameter for NDFDTR

Mnemonic	Value	Explanation
NDC_READ1	0x00	NAND Flash Command READ Data from 0 to 255 byte position
NDC_READ2	0x01	NAND Flash Command READ Data from 256 to 511 byte position.
NDC_READ3	0x50	NAND Flash Command READ Data from the redundant byte
NDC_WRITE	0x80	NAND Flash Command WRITE Data from the top of page.
NDC_AUTOP	0x10	NAND Flash Command Invoke Auto-Program Operation
NDC_STATS	0x70	NAND Flash Command READ Status
NDC_READID	0x90	NAND Flash Command ID READ operation
NDC_RESET	0xFF	NAND Flash Command Initialize NAND

Table 10-3 Mnemonic Parameter for NDFMCR

Mnemonic	Value	Explanation
ND_CLE	0x0001	Assert CLE
ND_ALE	0x0002	Assert ALE
ND_CH0	0x0000	Select NAND #0
ND_CH1	0x0004	Select NAND #2
ND_CH2	0x0008	Select NAND #3
ND_CH3	0x000C	Select NAND #4
ND_CE	0x0010	Activate NAND Controller
ND_ECCR	0x0060	Reset ECC Circuit
ND_ECCE	0x0020	ECC Enable
ND_ECCD	0x0000	ECC Disable
ND_ECCRD	0x0040	READ ECC generated by NDFMC or **UPDATE** external latch
ND_WE	0x0080	Activate Write Enable
ND_DMA1	0x0100	Activate 128 Byte DMA Transfer
ND_DMA2	0x0200	Activate 256 Byte DMA Transfer
ND_DMA3	0x0300	Activate 512 Byte DMA Transfer
ND_X16B	0x0400	Set to 16-bit Bus Mode

10.3.3. Accessing NAND Flash Memory (General Procedure)

With the NDFMC, you can access NAND flash memory by controlling registers. The NDFMC also has an ECC calculation function. See 10.4.1 for more information on ECC. This subsection describes the procedure for accessing NAND flash memory.

To access NAND Flash Memory, following steps are required.

STEP 1 Initialization	Write any Data to NDFSR, this initialize entire NAND Flash Controller. Define the Write Pulse Width and Hold Time by setting NDFSPR register. Write 0x0000 to NDFMCR followed by writing any value to NDFDTR, this operation generates ND_LA assertion to update the value of external latch.
STEP 2 Preparation	Select a NAND Chip by asserting ND_CE bit. ND_CS [1:0] provides additional selection for NAND chips. Once select a NAND chip, do not change these three parameters until the end of entire NAND access.
STEP 3-X Operation	By using NDFMCR and NDFDTR register do the read/write operation for NAND device. Repeat these operations. First, set signal pattern and command mode in the NDFMCR, then either read or write NDFDTR to access NAND flash memory. Reading from NDFDTR invokes a read cycle to external NAND flash memory. The read cycle to NDFDTR also ends when the read cycle to external NAND flash memory ends. During this bus transaction, assertions of ND_LA and ND_RE* are unconditional. Writing to NDFDTR invokes a write cycle to external NAND flash memory. The write cycle to NDFDTR also ends when the write cycle to external NAND flash memory ends. During this bus transaction, the assertion of ND_LA is unconditionally, but the assertion of ND_WE* is conditional. In case of “COMMAND” or “ADDRESS” phase with ALE=1 or CLE=1, the assertion of ND_WE* is unconditional. In other cases, the assertion of ND_WE* is controlled by WE bit in NDFMCR register. (See Note below)
STEP Last	Deassert ND_CE bit
(Note)	Write operation to NDFMCR will complete when the value is set to this register. Only the CE bit contents propagate to external terminal on the fly. However, the other signal bits are just set in the register only. They will not propagate to external latch at that time. Because of this, an intentional synchronization is required as UPDATE operation in some cases. For example, after ALE signal de-assertion, followed ND_RE* assertion should wait certain period. In this case, UPDATE operation with only ND_LA assertion is required.

10.3.4. Initialization and UPDATE

NAND Flash Controller should be initialized after power on and before use. Since six of NAND Controller signal use external latches, these latch statuses have to be synchronize with corresponding internal registers. This synchronization is the “UPDATE” operation in this document.
For initialization and UPDATE, three registers, such as NDFSR, NDFDTR, and NDFMCR are used.
Figure 10-2 shows field definition of these registers. Section 10.5.1, 10.5.2, 10.5.3 describe details.

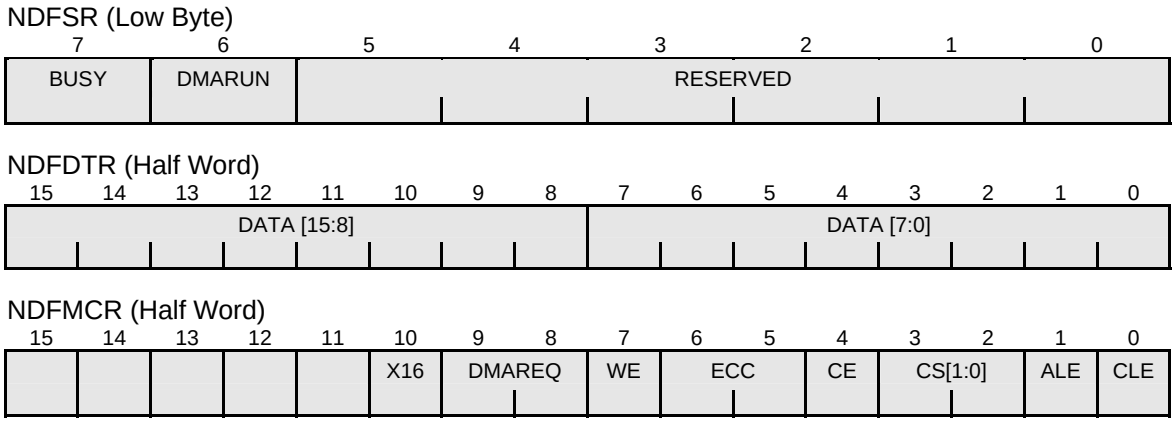


Figure 10-3 Field Definition of Registers NDFDTR and NDFMCR

Writing any data to NDFSR register (NAND Flash Memory Status Register =NAND Flash Memory Controller Reset) initializes entire NAND Flash Controller.

NDFMCR (NAND Flash Mode Control Register) consists of signal control bits and mode control bits. The signal control bits are WE, CE, CS [1:0], ALE, and CLE, those are transformed to NDRE*, NDWE*, NDCE* NDCS [1:0], ALE, and CLE.

Among these signals, the connection from CE to NDCE* signal is direct but the rests are through external latches. Control program has to update the logic status of external latches whenever it changes the contents of register. The name of this operation is UPDATE. Below is an example.

NDFMC initialize followed by UPDATE Sequence

```
$NDFSR  = 0x0000;            // initialize entire NAND Controller
$NDFSPR = 0x0106;            // Set SPW=6, HOLD=1 (optional)
$NDFMCR = 0x0000;            // Set WE=0, ALE=0, CLE=0 at least. (For simple UPDATE)
$NDFDTR = 0x0000;            // Dummy write to NDFDTR, This invoke NAND bus transaction
                             // with only ND_LA assertion.
```

Since this dummy write (to NDFDTR) invokes a real bus transaction on external bus, it has better result than software wait loop in terms of clock cycle consumed. (Note that the execution period of the software timer might fluctuate depend on the CPU Cache situation). The duration of this bus transaction consists of pulse width, hold time, and overhead. (where pulse width and hold time are defined by the register NDFSPR in terms of GBUSCLK period and minimum overhead is 4 GBUSCLK period). Please see corresponding register descriptions.

Following example shows on the fly UPDATE with wait cycle.

On the fly UPDATE

```
***
$NDFMCR = ND_CE|ND_CH0|ND_CLE;              // Assert the ND_CLE* for the Command mode.
$NDFDTR = NDC_READ1;                        // Data Read command.
$NDFMCR = ND_CE|ND_CH0|ND_ALE;              // ND_ALE* signal

$NDFDTR = <Times of necessary>;             // Set Addresses

$NDFMCR = ND_CE|ND_CH0;                     // De-assert ALE
for(i=0;i<4;i++)                            // UPDATE and Wait tWB (60ns x 4 = 240ns)
   $NDFDTR = 0x0000;
***
```

10.3.5. Write Sequence (8-bit Bus, Program Mode)

Write Sequence

```
//////////////////////////////////////////
// Write Sequence (512-Byte)            //
//////////////////////////////////////////
$NDFMCR = ND_CH0;                        // Select CH#0 and set WE=0 for prepare UPDATE
$NDFDTR = 0x0000;                        // UPDATE the Latch
$NDFMCR = ND_ECCR|ND_CH0;                // Initialize ECC circuit
$NDFMCR = ND_CE|ND_CH0|ND_CLE;           // Assert the ND_CLE* for the Command mode.
$NDFDTR = NDC_WRITE;                     // Write 0x80 (Data Write command).
$NDFMCR = ND_CE|ND_CH0|ND_ALE;           // Assert ND_ALE* for the Address mode.
$NDFDTR = 0x0FF & ( A );                 // Set A[7:0]
$NDFDTR = 0x0FF & ( A >> 9);             // Set A[16:9]
$NDFDTR = 0x0FF & ( A >> 17);            // Set A[24:17]
$NDFDTR = 0x01  & ( A >> 25);            // (A[25] if necessary)

$NDFMCR = ND_WE|ND_ECCEN|ND_CE|ND_CH0;   // Set ECC enable and enter Data mode.
for(i=0;i<512;i++)
{
   $NDFDTR = DATA[i];                    // Set 512-Byte data in order.
}

//////////////////////////////////////////
// Read ECC data from FDFMC             //
//////////////////////////////////////////
$NDFMCR = ND_ECCRD|ND_CE|ND_CH0;         // Calculated ECC Data Read mode.
for(I = 0;I < 6;i++)
{
   RDR[i] = $NDFDTR;
}

//////////////////////////////////////////
// Write 16-bytes Redundant Data        //
//////////////////////////////////////////
$NDFMCR= ND_WE|ND_ECCD|ND_CE|ND_CH0;     // Activate NDFMC on CH#0, and initialize
for(I = 0;I < 16;i++)                    // the ECC data of NDFMC.
{
   $NDFDTR = RDATA [i];
}

//////////////////////////////////////////
// Execute Auto-Program                 //
//////////////////////////////////////////

$NDFMCR = ND_ECCD|ND_CE|ND_CH0|ND_CLE;   // Enter the Command mode.
$NDFDTR = NDC_AUTOP;                     // Write 0x10 (Page Program command).
$NDFMCR = ND_CE|ND_CH0;                  // De-assert ND_WE and ND_CLE*.
$NDFDTR = 0x0000;                        // UPDATE (CLE)

//////////////////////////////////////////
// Wait until “BUSY” goes off           //
//////////////////////////////////////////

while ((0x01 & $NDFISR) == 1){}

//////////////////////////////////////////
// Read status                          //
//////////////////////////////////////////

$NDFMCR = ND_CE|ND_CH0|ND_CLE;           // Assert the ND_CLE* for Command Mode.
$NDFDTR = NDC_STATS;                     // Write 0x70 (Read Status command).
$NDFMCR = ND_CE|ND_CH0 ;                 // De-assert the ND_CLE*
$NDFDTR = 0x0000;                        // UPDATE CLE and short wait
Status  = $NDFDTR;                       // Read status value

$NDFMCR = 0x0000;                        // Release NAND
$NDFDTR = 0x0000;                        // UPDATE
//////////////////////////////////////////
// Repeat this process for other pages  //
//////////////////////////////////////////
```

10.3.6. Read Sequence (8-bit Bus, Program / DMA Mode)

Read Sequence (8-bit Bus, Program / DMA Mode)

```
//////////////////////////////////////////
// Read Sequence                        //
//////////////////////////////////////////

$NDFMCR = ND_CH0;                          // Select CH#0 and set WE=0 for prepare UPDATE
$NDFDTR = 0x0000;                          // UPDATE the Latch
$NDFMCR = ND_ECCR|ND_CH0;                  // Initialize ECC circuit
$NDFMCR = ND_CE|ND_CH0|ND_CLE;             // Assert Command Latch Enable

//////////////////////////////////////////
// Read Command and Address setting     //
//////////////////////////////////////////
$NDFDTR = NDC_READ1;                       // Data Read command.
$NDFMCR = ND_CE|ND_CH0|ND_ALE;             // ND_ALE* signal

$NDFDTR = 0x0FF & ( A );                   // Set A[7:0]
$NDFDTR = 0x0FF & ( A >> 9);               // Set A[16:9]
$NDFDTR = 0x0FF & ( A >> 17);              // Set A[24:17]
$NDFDTR = 0x001 & ( A >> 25);              // (A[25] if necessary) in order.
$NDFMCR = ND_CE|ND_CH0;                    // Prepare de-assert ALE
$NDFDTR = 0x0000;                          // UPDATE (ALE)

for(i=0;i<4;i++)                           // Wait tWB (60ns x 4 = 240ns)
   $NDFDTR = 0x0000;

while( (0x01 & $NDFSR) == 1){}             // Wait until Data Ready
$NDFMCR = ND_ECCE|ND_CE|ND_CH0;            // Set ECC Enable
//////////////////////////////////////////
// Read 512 byte data                   //
//////////////////////////////////////////
if(DMAMODE)
{
   $NDFMCR = ND_DMA3|ND_ECCE|ND_CE|ND_CH0;  // Invoke DMA Operation for 512 bytes.
}
else   // Program mode
{
   for(i = 0;i < 512;i++)
   {
      DATA[i] = $NDFDTR;
   }
}

//////////////////////////////////////////
// Get the ECC data calculated.         //
//////////////////////////////////////////
$NDFMCR = ND_CE|ND_CH0;                    // Set 0x10, Enter Data mode w/o ECC
for(i=0;i<16;i++)
{
   RDATA[i] = $NDFDTR;
}

//////////////////////////////////////////
// Read ECC calculated by NDFMC         //
//////////////////////////////////////////
$NDFMCR = ND_ECCR|ND_CE|ND_CH0;            // Set 0x50, and enter the ECC Data Read mode.

for(i=0;i<6;i++)
{
   RECC[i] = $NDFDTR;
}

$NDFMCR = 0x0000;                          // Release NAND
$NDFDTR = 0x0000;                          // UPDATE

//////////////////////////////////////////
// Compare the ECC data.                //
// If the data do not match, invoke the //
// necessary error process.             //
//////////////////////////////////////////
```

10.3.7. Read ID

Read ID (8-bit Bus, Program Mode)

```
//////////////////////////////////////////
// Read ID Sequence                     //
//////////////////////////////////////////

$NDFMCR = ND_CE|ND_CH0|ND_CLE;              // Select CH#0 and assert the ND_CLE*
$NDFDTR = NDC_READID;                       // Write 0x90 (ID Read command).
$NDFMCR = ND_CE|ND_CH0|ND_ALE;              // Assert the ND_ALE* for Address Mode.
$NDFDTR = 0x00;                             // Write 0x00 for ID Address.

$NDFMCR = ND_CE|ND_CH0;                     // De-assert ND_ALE.
$NDFDTR = 0x0000;                           // UPDATE

MakerCode = $NDFDTR;                        // Read the maker code.
DeviceCode = $NDFDTR                        // Read the device code.

$NDFMCR = 0x0000 ;                          // Prepare Release
$NDFDTR = 0x0000;                           // UPDATE
$NDFMCR = 0x00;                             // Release NAND
```

10.4. ECC and DMA Operation

10.4.1. ECC Generation

ECC data consists of line and column parity. The 512 bytes data is treated as two sets of 256 x 8 bit memory array. Each memory array can contain one bit error as correctable and two bit error detectable. Following table describes complete formula to generate ECC data. LPR00 = LPR[0] = EXOR(0XXXXXXX0,XXX) means an odd-parity with all data bit which address A8, A0 is 0 and the rest of addresses are 0 and 1.

ECC		A8	A7	A6	A5	A4	A3	A2	A1	A0		C2	C1	C0		COMMENT
LPR00	= EXOR (	0	X	X	X	X	X	X	X	0	,	X	X	X	)	
LPR01	= EXOR (	0	X	X	X	X	X	X	X	1	,	X	X	X	)	
LPR02	= EXOR (	0	X	X	X	X	X	X	0	X	,	X	X	X	)	
LPR03	= EXOR (	0	X	X	X	X	X	X	1	X	,	X	X	X	)	
LPR04	= EXOR (	0	X	X	X	X	X	0	X	X	,	X	X	X	)	
LPR05	= EXOR (	0	X	X	X	X	X	1	X	X	,	X	X	X	)	
LPR06	= EXOR (	0	X	X	X	X	0	X	X	X	,	X	X	X	)	
LPR07	= EXOR (	0	X	X	X	X	1	X	X	X	,	X	X	X	)	
LPR08	= EXOR (	0	X	X	X	0	X	X	X	X	,	X	X	X	)	
LPR09	= EXOR (	0	X	X	X	1	X	X	X	X	,	X	X	X	)	
LPR10	= EXOR (	0	X	X	0	X	X	X	X	X	,	X	X	X	)	
LPR11	= EXOR (	0	X	X	1	X	X	X	X	X	,	X	X	X	)	
LPR12	= EXOR (	0	X	0	X	X	X	X	X	X	,	X	X	X	)	
LPR13	= EXOR (	0	X	1	X	X	X	X	X	X	,	X	X	X	)	
LPR14	= EXOR (	0	0	X	X	X	X	X	X	X	,	X	X	X	)	
LPR15	= EXOR (	0	1	X	X	X	X	X	X	X	,	X	X	X	)	
CPR00	= EXOR (	0	X	X	X	X	X	X	X	X	,	X	X	0	)	
CPR01	= EXOR (	0	X	X	X	X	X	X	X	X	,	X	X	1	)	
CPR02	= EXOR (	0	X	X	X	X	X	X	X	X	,	X	0	X	)	
CPR03	= EXOR (	0	X	X	X	X	X	X	X	X	,	X	1	X	)	
CPR04	= EXOR (	0	X	X	X	X	X	X	X	X	,	0	X	X	)	
CPR05	= EXOR (	0	X	X	X	X	X	X	X	X	,	1	X	X	)	
LPR16	= EXOR (	1	X	X	X	X	X	X	X	0	,	X	X	X	)	
LPR17	= EXOR (	1	X	X	X	X	X	X	X	1	,	X	X	X	)	
LPR18	= EXOR (	1	X	X	X	X	X	X	0	X	,	X	X	X	)	
LPR19	= EXOR (	1	X	X	X	X	X	X	1	X	,	X	X	X	)	
LPR20	= EXOR (	1	X	X	X	X	X	0	X	X	,	X	X	X	)	
LPR21	= EXOR (	1	X	X	X	X	X	1	X	X	,	X	X	X	)	
LPR22	= EXOR (	1	X	X	X	X	0	X	X	X	,	X	X	X	)	
LPR23	= EXOR (	1	X	X	X	X	1	X	X	X	,	X	X	X	)	
LPR24	= EXOR (	1	X	X	X	0	X	X	X	X	,	X	X	X	)	
LPR25	= EXOR (	1	X	X	X	1	X	X	X	X	,	X	X	X	)	
LPR26	= EXOR (	1	X	X	0	X	X	X	X	X	,	X	X	X	)	
LPR27	= EXOR (	1	X	X	1	X	X	X	X	X	,	X	X	X	)	
LPR28	= EXOR (	1	X	0	X	X	X	X	X	X	,	X	X	X	)	
LPR29	= EXOR (	1	X	1	X	X	X	X	X	X	,	X	X	X	)	
LPR30	= EXOR (	1	0	X	X	X	X	X	X	X	,	X	X	X	)	
LPR31	= EXOR (	1	1	X	X	X	X	X	X	X	,	X	X	X	)	
CPR06	= EXOR (	1	X	X	X	X	X	X	X	X	,	X	X	0	)	
CPR07	= EXOR (	1	X	X	X	X	X	X	X	X	,	X	X	1	)	
CPR08	= EXOR (	1	X	X	X	X	X	X	X	X	,	X	0	X	)	
CPR09	= EXOR (	1	X	X	X	X	X	X	X	X	,	X	1	X	)	
CPR10	= EXOR (	1	X	X	X	X	X	X	X	X	,	0	X	X	)	
CPR11	= EXOR (	1	X	X	X	X	X	X	X	X	,	1	X	X	)	

For detail explanation of this ECC structure, please refer http://www.ssfdc.or.jp and, the document of "SmartMedia ECC Reference Manual Ver 2.1".

10.4.2. ECC Data Format

ECC generates LPR [31:0] and CPR [11:0] (totally 44-bit data). Use a program to check ECC for errors and correct any errors that occur.

You can read from the NDFDTR Register the ECC data calculated by NDFMC by setting the NDFMCR to either 0xD0 (Write mode) or 0x50 (Read mode). The ECC data you can read is 6-Byte data. Read NDFDTR six times. Data is read in the following order.

First data:	LPR[7:0]
Second data:	LPR[15:8]
Third data:	CPR[5:0], 2'b11
Fourth data:	LPR[23:16]
Fifth data:	LPR[31:24]
Sixth data:	CPR[11:6], 2'b11

As an example, this ECC data will be stuffed in the part of 16-byte redundant data space of NAND Flash. This redundant memory cell is 16-bytes. Those are numbered from D512 to D527. Figure 10-4 shows the location of this ECC. Please note that this structure is purely example and actual system might use its own structure.

D512	D513	D514	D515	D516	D517	D518	D519
Function Select Information Field				Invalid Data Flag	Fixed 0xFF	Fixed 0x00	Fixed 0x00

D520	D521	D522	D523	D524	D525	D526	D527
LPR[23:16]	LPR[31:24]	CPR[11:6]+11	Fixed 0x00	Fixed 0x00	LPR[7:0]	LPR[15:8]	CPR[5:0]+11

Figure 10-4 ECC Position in NAND Data Space.

10.4.3. DMA Operation

Refer to Section 14.3.7 in DMA Controller.
This DMA Operation uses DMA0 Channel 3. The burst size is 8-bytes (Double Word), from NDFMC to memory is supported. Before using DMA0 Operation DMA0 channel 3 should be properly set as follows.

DMA Request Polarity	Low Active	DM0CCR3.REQPOL = 0
DMA Acknowledge Polarity	Low Active	DM0CCR3.ACKPOL = 0
Request Detection	Level Detection	DM0CCR3.EGREQ = 0
Transfer Size	8 Byte	DM0CCR3.XFSZ = 011b
Transfer Direction	I/O to Memory	DM0CCR3.MEMIO = 0
Transfer Address Mode	Single	DM0CCR3.SNGAD = 1
I/O DMA Transfer Mode	External	DM0CCR3.EXTRQ = 1
Pin Config in shared DMA	Set to NDFMC	PCFG.DMASEL3 = 1'b0

Note: Destination Address (in memory) should be Double-Word aligned.

Once DMA transfer is requested, NDFMC generates DATA READ sequence to NAND memory continuously and is filling the FIFO (16-bytes). When 8-bytes data are filled, the controller assert DMAREQ signal to DMAC. Same time, controller release the external bus once and connect NAND memory again if there are rooms in the FIFO. If FIFO has no room, the controller disconnects NAND and waits until FIFO get a room.

10.4.4. Byte Sequence of DMA and ECC Generation

This has close relationship with System Endianness. Following figures describes these.

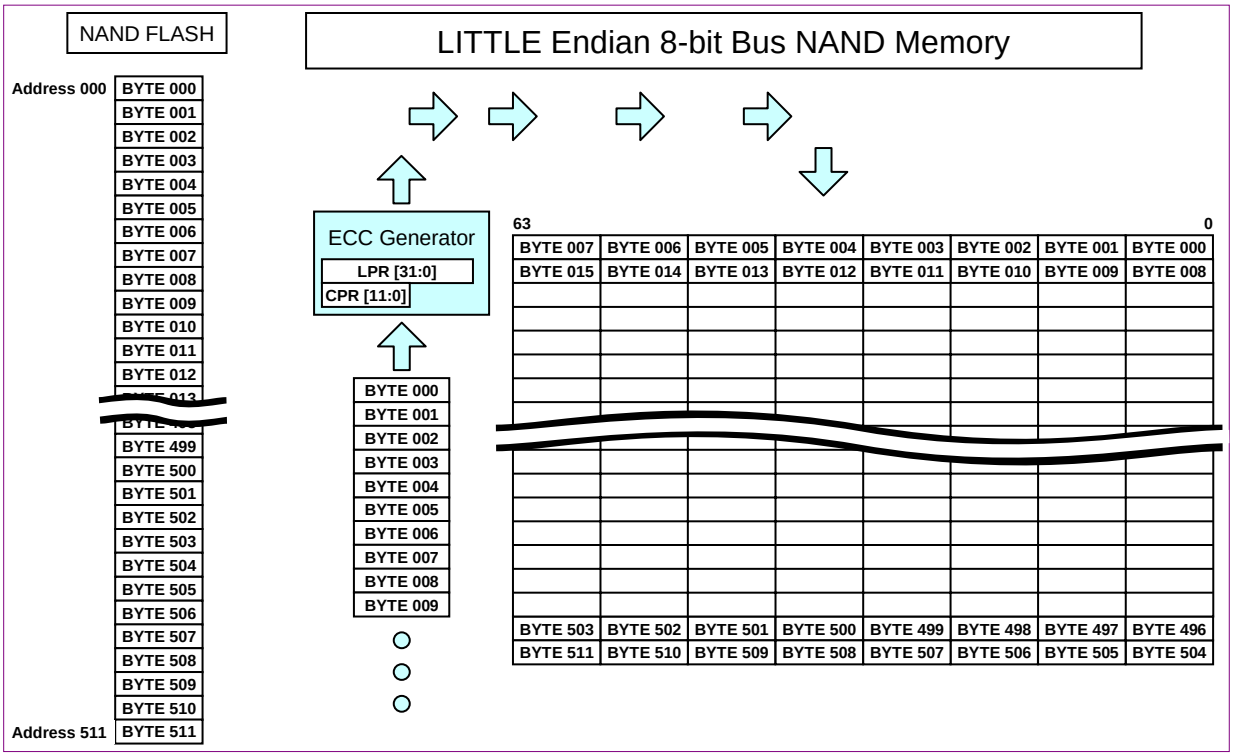


Figure 10-5 Byte Sequence of Little Endian and 8-bit bus

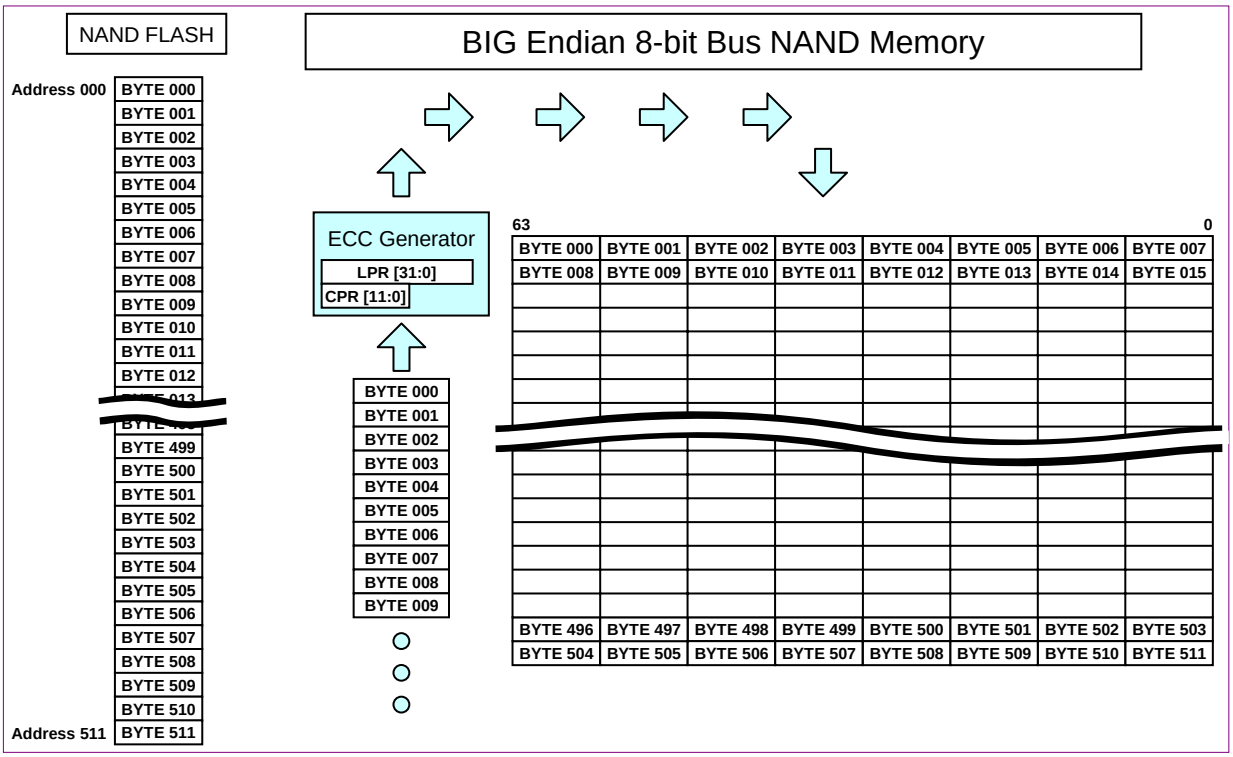


Figure 10-6 Byte Sequence of Big Endian and 8-bit bus

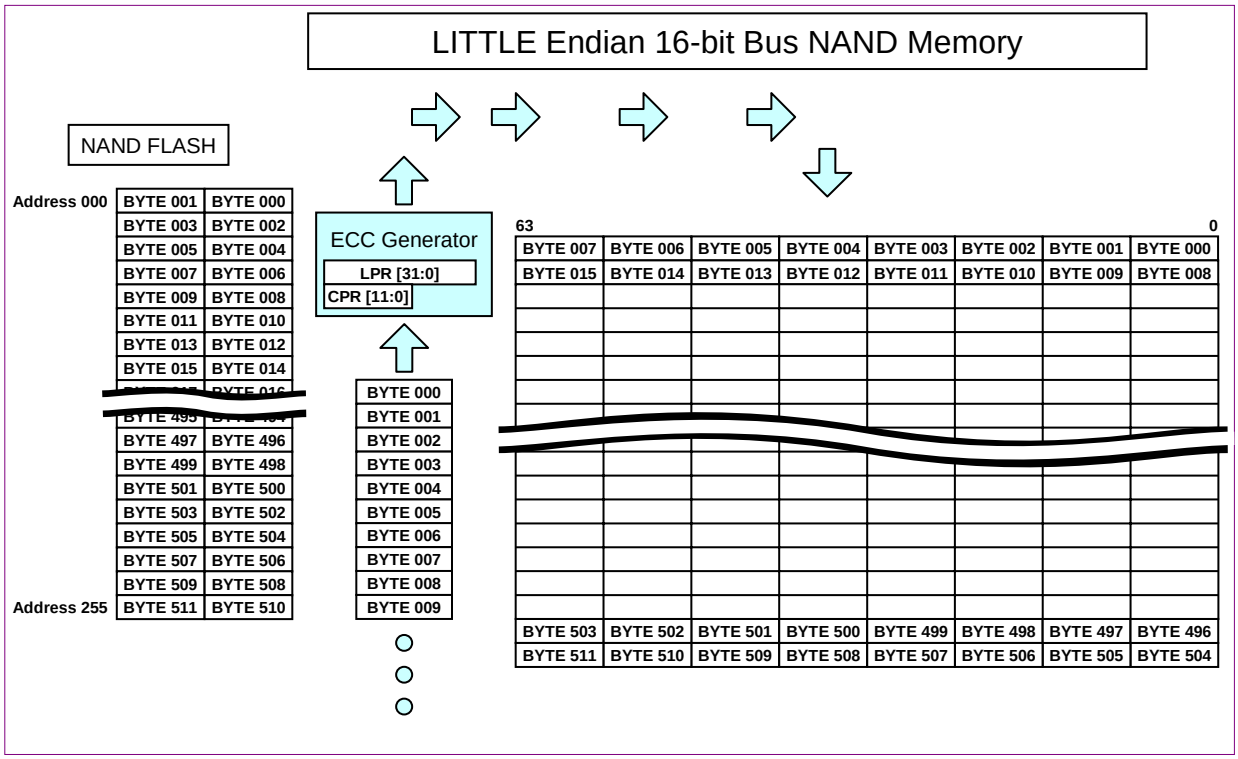


Figure 10-7 Byte Sequence of Little Endian and 16-bit bus

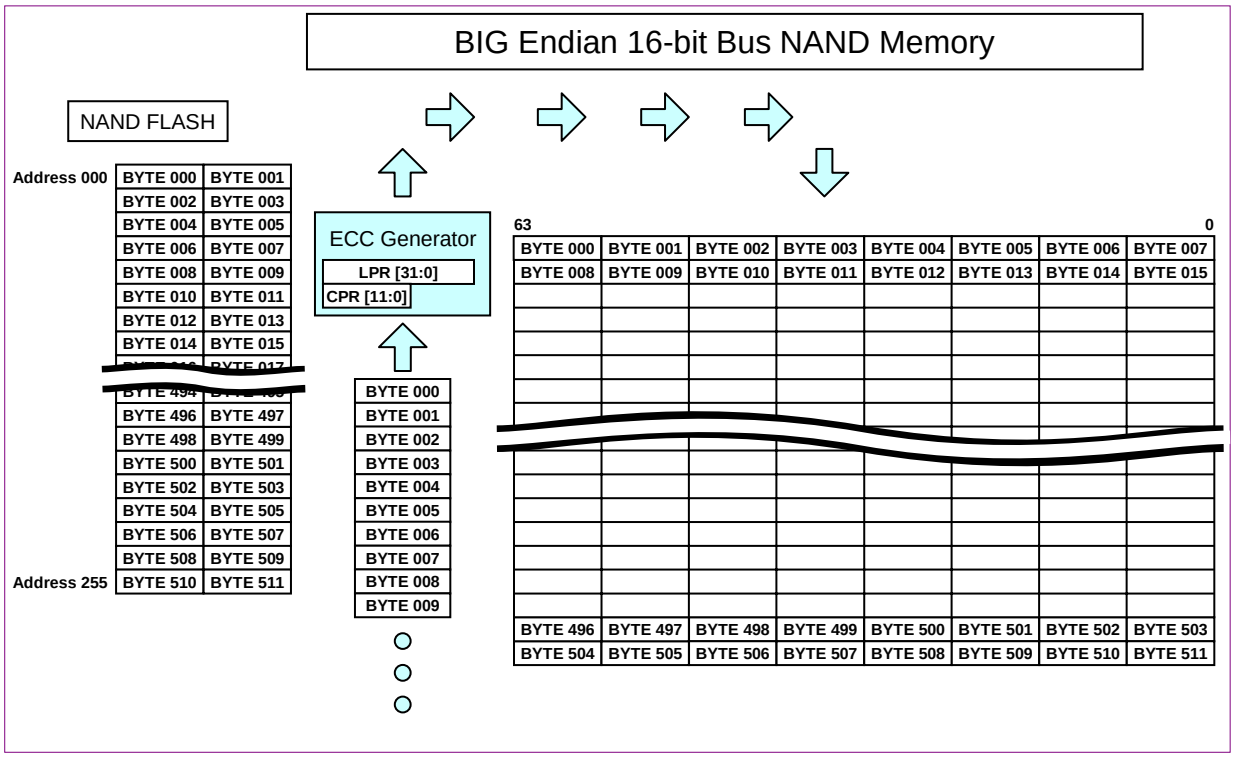


Figure 10-8 Byte Sequence of Big Endian and 16-bit bus

10.5. Registers Detail

10.5.1. NAND Flash Memory Data Transfer Register (NDFDTR) 0x5000

Read or Write access of this register invokes External Bus transaction. See 10.3 for detail.

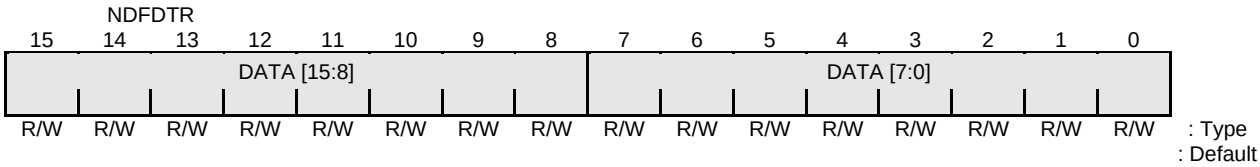


Figure 10-9 NAND Flash Memory Data Transfer Register (NDFDTR)

Table 10-4 NAND Flash Memory Data Transfer Register (NDFDTR)

Bits	Mnemonic	Field Name	Description
31 : 16	—	Reserved	—
15:8	DATA [15:8]	DATA [15:8]	Upper DATA BYTE in case of X16 NAND Flash at Data mode.
7 : 0	DATA [7:0]	DATA [7:0]	NAND Flash memory data Read: Reads NAND flash data when in the Data mode. Write: Writes transmission data in NAND flash memory in Data mode.

10.5.2. NAND Flash Memory Mode Control Register (NDFMCR) 0x5008

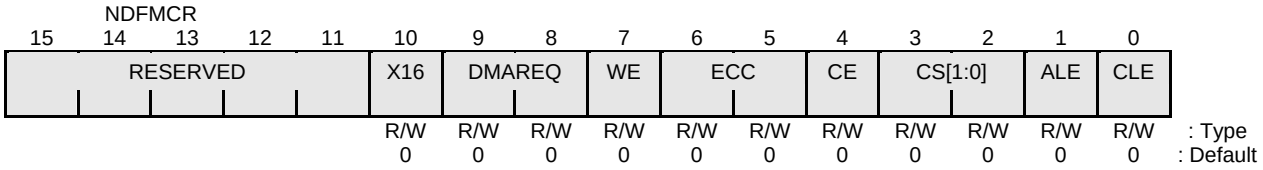


Figure 10-10 NAND Flash Memory Mode Control Register (NDFMCR)

Table 10-5 NAND Flash Memory Mode Control Register (NDFMCR)

Bits	Mnemonic	Field Name	Description
31 : 11	–	Reserved	–
10	X16	X16BUS	Indicate NAND Flash has 16-bit Bus. (Default: 0) 0: 8-bit Bus (Default) 1: 16-bit Bus.
9:8	DMAREQ	DMA Request	Initiate DMA data transfer (Default: 00) Invoke DMA transfer from/to NDFDTR port to memory in system. 00: No DMA 01: 128 Byte Transfer 10: 256 Byte Transfer 11: 512 Byte Transfer These bits will be cleared when all the requested bytes have been transferred from NAND Flash to FIFO
7	WE	Write Enable	Write Enable (Default: 0) When CPU references NDFDTR register, it generates external bus READ operation with non-conditional ND_LA and ND_RE* assertion. When CPU writes a value to NDFDTR register, it also generates external bus WRITE operation with ND_LA and conditional ND_WE* assertion. During Command and Address write cycle with either ALE or CLE assertion, ND_WE* signal assert non-conditional. In other cases, both ALE and CLE = 0, ND_WE* signal assertion is controlled by this bit as follows. 0: Disable ND_WE* assertion (See Note below) 1: Enable ND_WE* assertion Note Write operation to NDFDTR with WE=0, ALE=0, CLE=0, can be used to synchronize external latch with the contents of NDFMCR register.
6 : 5	ECC	ECC Control	ECC Control (Default: 00) This field specifies the operation of the ECC calculation circuit. 11: Reset the ECC circuit. 00: Disable the ECC circuit. 01: Enable the ECC circuit. 10: Read the ECC circuit the NDFMC calculated.
4	CE	Chip Enable	Chip Enable Default: 0) Enables access to NAND flash memory. Set this bit when accessing NAND flash memory. 0: Disable (ND_CE* high.) 1: Enable (ND_CE* low.)
3 : 2	CS [1:0]	Chip Select	Chip Select
1	ALE	Address Latch Enable	Address Latch Enable (Default: 0) Specifies the ND_ALE* signal. 0: Low 1: High
0	CLE	Command Latch Enable	Command Latch Enable (Default: 0) Specifies the value of the NC_CLE signal. 0: Low 1: High

10.5.3. NAND Flash Memory Status Register (NDFSR) 0x5010

A write to this register will reset entire NAND Flash Controller (NDFMC) and registers.
A read will return hardware status describes as follow.

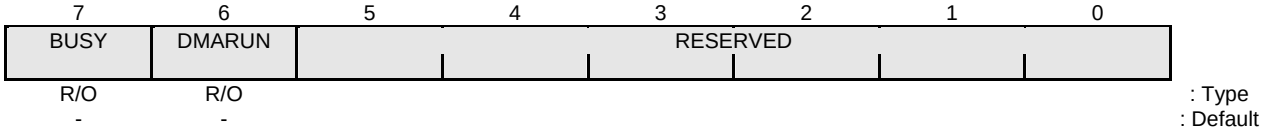


Figure 10-11 NAND Flash Memory Status Register (NDFSR)

Table 10-6 NAND Flash Memory Status Register (NDFSR)

Bits	Mnemonic	Field Name	Description
31 : 8	—	Reserved	—
7	BUSY	BUSY	BUSY (Default: Undefined) Indicates the status of NAND flash memory. 0: Ready 1: Busy
6	DMARUN	DMARUN	DMA Status Indicates the status of DMA channel working on NDFMC. 0: STOPPED 1: RUNNING

10

10.5.4. NAND Flash Memory Interrupt Status Register (NDFISR) 0x5018

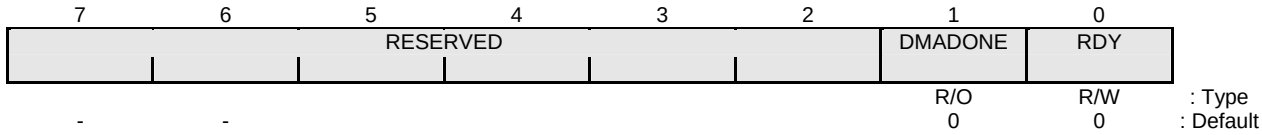


Figure 10-12 NAND Flash Memory Interrupt Status Register (NDFISR)

Table 10-7 NAND Flash Memory Interrupt StatusRegister (NDFISR)

Bits	Mnemonic	Field Name	Description
31 : 8	—	Reserved	—
7 : 2	-	Reserved	
1	DMADONE	DMADONE	DMA Status A ‘1’ indicates all the requested byte transfer which is specified in NDFMCR register have been transferred to FIFO. Write: 0: Don’t care 1: Clear to “0”
0	RDY	Ready	Ready (Default: 0) When the MRDY bit of the NDFIMR Register is "1", this bit is set to "1" when the ND_RB* signal changes from Low to High. Write "1" to clear this bit to "0". Read: 0: No change 1: ND_RB* signal changed from the Busy state to the Ready state. Write: 0: Don’t care 1: Clear to "0".

10.5.5. NAND Flash Memory Interrupt Mask Register (NDFIMR) 0x5020

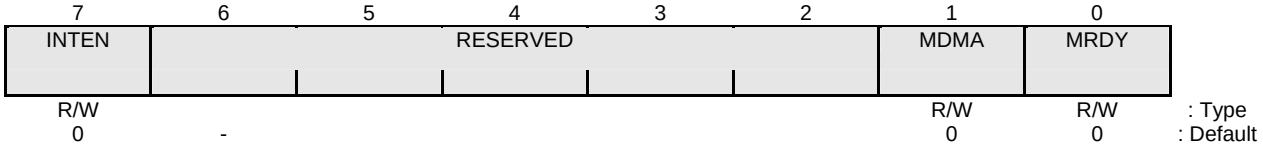


Figure 10-13 NAND Flash Memory Interrupt Mask Register (NDFIMR)

Table 10-8NAND Flash Memory Interrupt Mask Register (NDFIMR)

Bits	Mnemonic	Field Name	Description
31 : 8	—	Reserved	—
7	INTEN	Interrupt Enable	Interrupt Enable (Default: 0) This bit enables interrupts. If INTEN bit are set to "1", an interrupt occurs when the RDY or the DMADONE bit of the NDFISR Register becomes "1". 0: Disable 1: Enable
6 : 1	—	Reserved	
1	MDMA	Mask DMADONE interrupt	Mask DMADONE Interrupt (Default: 0) This bit masks the DMADONE bit of the NDFISR Register. If this bit is “1”, the DMADONE bit of the NDFISR Register is set when all the DMA requested byte have been transferred to FIFO. 0: Disable the NDFISR DMADONE bit. 1: Enable the NDFISR DMADONE bit.
0	MRDY	Mask RDY interrupt	Mask Ready Interrupt (Default: 0) This bit masks the RDY bit of the NDFISR Register. If this bit is "1", the RDY bit of the NDFISR Register is set when the ND_RB* signal changes from "0" to "1". 0: Disable the NDFISR RDY bit. 1: Enable the NDFISR RDY bit.

10.5.6. NAND Flash Memory Strobe Pulse Width Register (NDFSPR) 0x5028

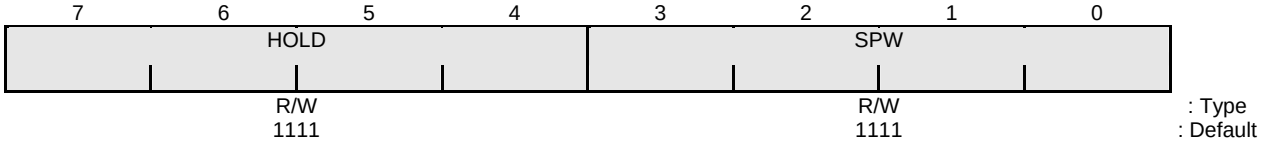


Figure 10-14 NAND Flash Memory Strobe Pulse Width Register (NDFSPR)

Table 10-9 NAND Flash Memory Strobe Pulse Width Register (NDFSPR)

Bits	Mnemonic	Field Name	Description
31 : 8	—	Reserved	—
7 : 4	HOLD	Hold Time	Hold Time (Default: 1111) This value defines the number of clock cycle from NDWE* or NDRE* de-assertion to SADB [15:0] bus and SA [5:0] bus release. This means so called hold-time for NAND write access with NDWE* signal. Actual hold time will be given as follows: Actual Hold Time = (HOLD + 1) x Period (GBUSCLK) where the value HOLD should not be zero (which is reserved), and Period (GBUSCLK) = Period of GBUSCLK ▪HOLD=1 with 200MHz GBUSCLK (period=5ns) results 10ns Hold Time ▪Need to consider delay time of an external Latch Device to determine HOLD Time
3 : 0	SPW	Strobe Pulse Width	Strobe Pulse Width (Default: 1111) This value defines the number of clock cycle for NDWE* or NDRE* assertion pulse width. Actual pulse width will be given as follows: Pulse Width of NDWE* = (SPW) x Period (GBUSCLK) Pulse Width of NDRE* = (SPW+1) x Period (GBUSCLK) where the value SPW should not be zero (which is reserved), and Period (GBUSCLK) = Period of GBUSCLK ▪ SPW=6 at 200MHz GBUSCLK (period=5ns) results 35 ns pulse width for NDRE* (read) and 30ns for NDWE* (write)

10.6. Timing diagram

10.6.1. Initialization Sequence

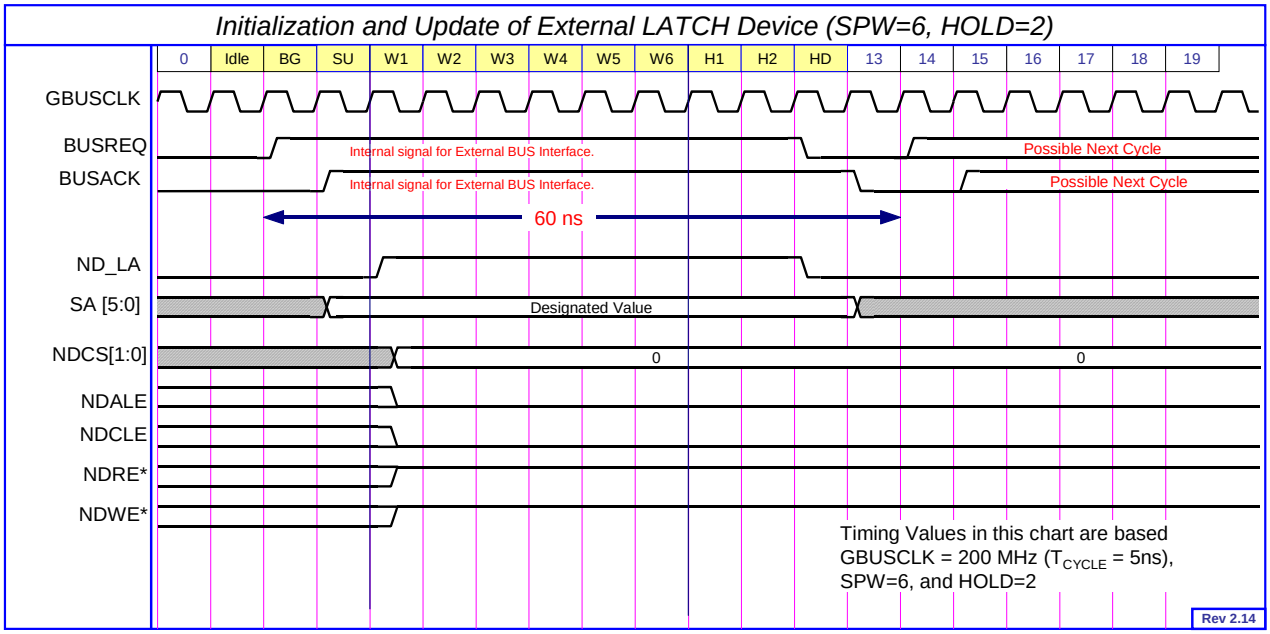


Figure 10-15 Initialization and Update Sequence

Required initialization Steps after power on:
Step 1 : Write to NDFSR. This will reset entire NDFMC registers.
Step 2 : Write to NDFSPR with the proper value of Write Pulse Width and Hold Time
Step 3 : Write any data to NDFDTR. This will generate ND_LA and synchronize the external latch status with internal value.

LEGEND for above and following timing chart

Idle	GBUS Idle state
BG	Bus Get cycle for GBUS
SU	Inherit set up cycle from SA[5:0] and SADB[15:0] to ND_LA, NDRE*, and NDWE*
Wn	PW state corresponds to the programmed pulse width (SPW). This will be from [PW1] to PW15]. NDWE* signal will be asserted exactly only this states, where NDRE* assertion will be kept until next H1 states.
Hn	HOLD state corresponds to the programmed hold time (HOLD). This will be from [H1] to [H15]. NDRE* signal assertion will be kept at H1 state.
HD	Inherit hold cycle from ND_LA de-assertion to SA [5:0] release. Since SADB [15:0] will be released at the same time with SA [5:0], for the hold time from NDRE* or NDWE* de-assertion to SADB [15:0] release has one additional cycle.

10.6.2. Data Read Sequence

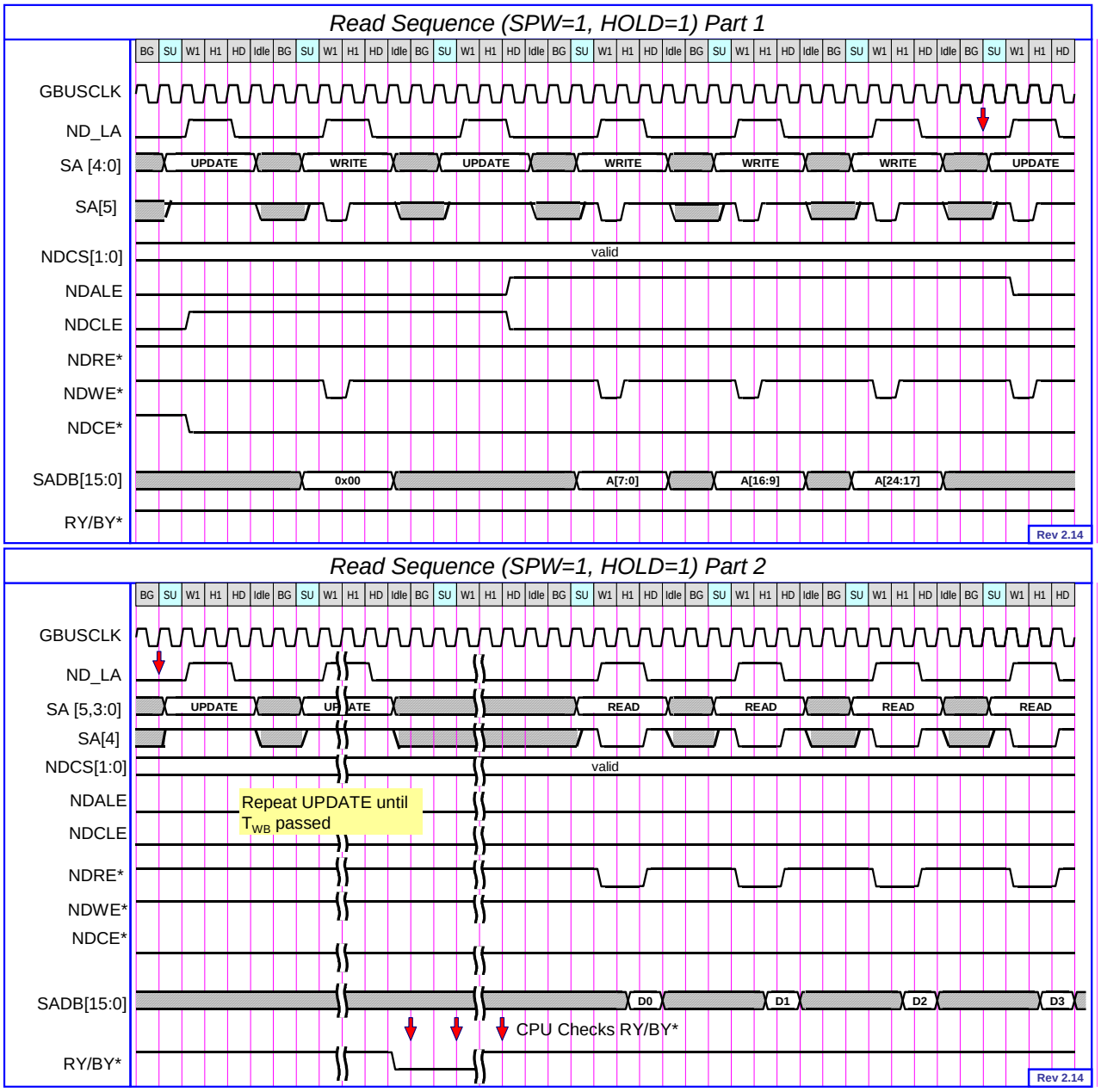


Figure 10-16 Command Cycles and Address Cycles and Read Cycle

10.6.3. Data Write Cycles

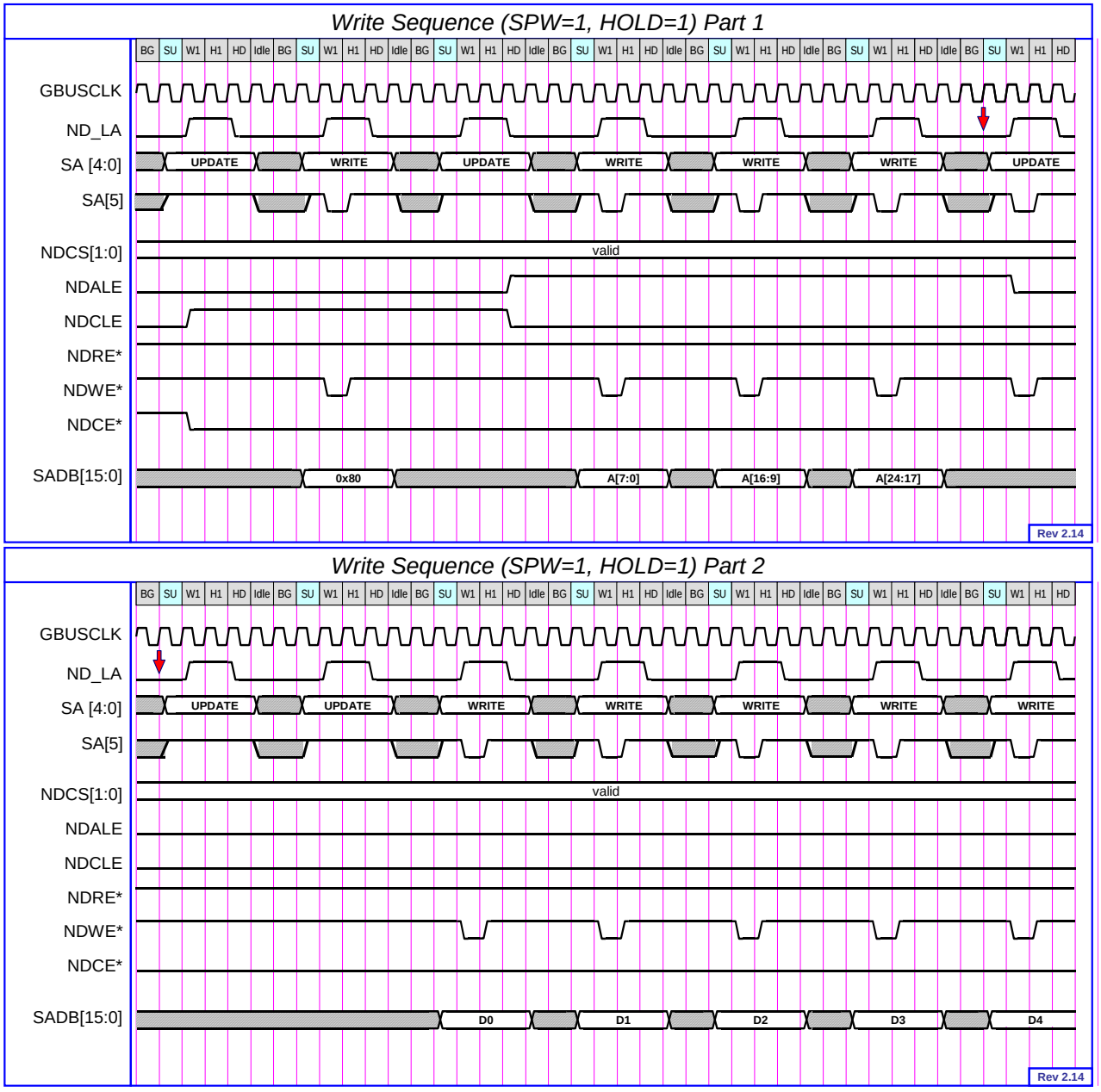


Figure 10-17 Data Write Cycles

Chapter 11. Real Time Clock

11.1. Feature of Real Time Clock Module (RTC)

Diagram below shows the overall structure of the TX4939 Real Time Clock.

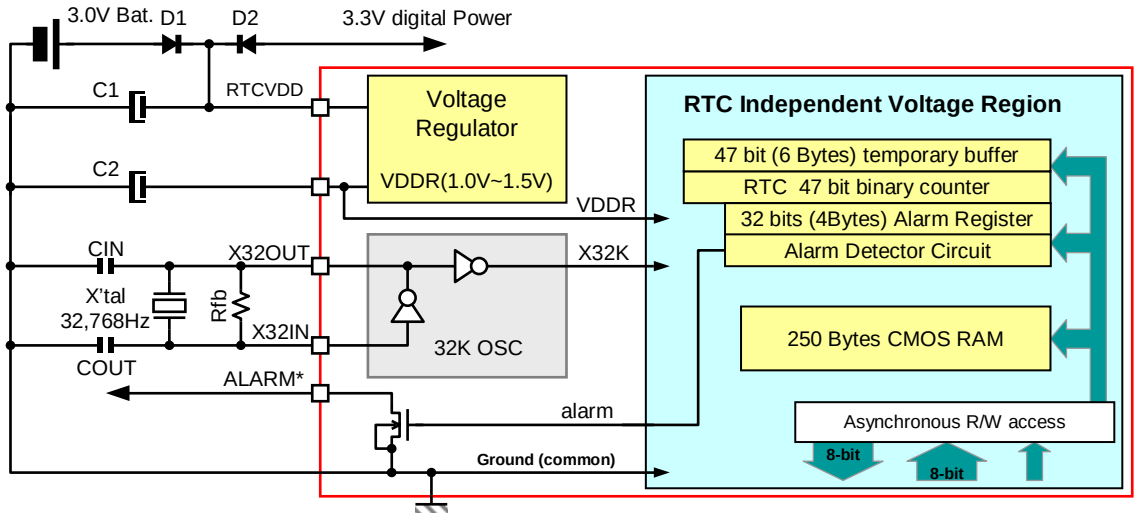


Figure 11-1 RTC Block Diagram

11.2. RTC IP Pin Description

The RTC has two groups of signal set. One is for external interface and the other is for internal interface. This internal interface consists of uni-directional 8 bits data bus and control signal to attach to IM-Bus.

11.2.1. RTC IP External Pin Description

Group	Signal Name	CNT	I/O	Description
Power Signal	RTCVDD	1	I	Independent RTC power supply. 2.7 V – 3.3 V
X'tal	X32IN	1	I	X'tal terminal (input)
	X32OUT	1	O	X'tal terminal (output)
Signal	ALARM*	1	O	Alarm Signal. (Low Active, Open Drain)
VDDR	VDDR	1	O	RTC Regulated Power Terminal. External decupling capacitance should be attached.

The actual voltage appears on VDDR terminal may differ depend on each device characteristic.

11.3. RTC IP Registers

11.3.1. RTC IP Internal Registers

RTC has 6 Bytes length time count register, 4 bytes length alarm register, 6 bytes length read/write buffer register, and 250 bytes universal registers.

Table 11-1 Non-Exposed Internal Registers

Name	Structure	Byte	Description
RTC	RTC [47:0]	6	RTC [47:0] is a 48-bits length register that counts two on every clock edge of 32.768 KHz oscillator clock. Therefore, the value of RTC [47:0] register is considered as fixed-point-representation. Such as RTC [47:16] is corresponds to the time count of second, starting from January 1st of 1970. RTC [15:0] is the decimal place of second.
ALARM	ALARM [47:16]	4	The contents of ALARM [47:16] will be compared with RTC [47:16] on every 32.768 KHz clock edge. Once the value is matched, alarm signal will be asserted. This alarm signal keep asserted until new value is set in the ALARM register.

Table 11-2 Exposed Internal Registers

Name	Structure	Byte	Description
RWB	RWB [47:0]	6	RWB [47:0] is a 48-bits register corresponds to the RTC register and ALARM register. The contents of ALARM [47:16] register will correspond to RWB [47:16] when the data exchange happens.
RAM	RAM [6:255] [7:0]	250	These registers are the entity of Non-Volatile CMOS memories.

Table 11-3 Address Mapping of Exposed Registers

	7	6	5	4	3	2	1	0
0x00	RAM [7]	RAM [6]	RWB[47:40]	RWB[39:32]	RWB[31:24]	RWB[23:16]	RWB[15:8]	RWB[7:0]
0x08	RAM [15]	RAM [14]	RAM [13]	RAM [12]	RAM [11]	RAM [10]	RAM [9]	RAM [8]
0x10	RAM [23]	RAM [22]	RAM [21]	RAM [20]	RAM [19]	RAM [18]	RAM [17]	RAM [16]
0x18	RAM [31]	RAM [30]	RAM [29]	RAM [28]	RAM [27]	RAM [26]	RAM [25]	RAM [24]
0x20	RAM [39]	RAM [38]	RAM [37]	RAM [36]	RAM [35]	RAM [34]	RAM [33]	RAM [32]
0x28	RAM [47]	RAM [46]	RAM [45]	RAM [44]	RAM [43]	RAM [42]	RAM [41]	RAM [40]
0x30	RAM [55]	RAM [54]	RAM [53]	RAM [52]	RAM [51]	RAM [50]	RAM [49]	RAM [48]
0x38	RAM [63]	RAM [62]	RAM [61]	RAM [60]	RAM [59]	RAM [58]	RAM [57]	RAM [56]
0x40	RAM [71]	RAM [70]	RAM [69]	RAM [68]	RAM [67]	RAM [66]	RAM [65]	RAM [64]
0x48	RAM [79]	RAM [78]	RAM [77]	RAM [76]	RAM [75]	RAM [74]	RAM [73]	RAM [72]
0x50	RAM [87]	RAM [86]	RAM [85]	RAM [84]	RAM [83]	RAM [82]	RAM [81]	RAM [80]
0x58	RAM [95]	RAM [94]	RAM [93]	RAM [92]	RAM [91]	RAM [90]	RAM [89]	RAM [88]
0x60	RAM [103]	RAM [102]	RAM [101]	RAM [100]	RAM [99]	RAM [98]	RAM [97]	RAM [96]
0x68	RAM [111]	RAM [110]	RAM [109]	RAM [108]	RAM [107]	RAM [106]	RAM [105]	RAM [104]
0x70	RAM [119]	RAM [118]	RAM [117]	RAM [116]	RAM [115]	RAM [114]	RAM [113]	RAM [112]
0x78	RAM [127]	RAM [126]	RAM [125]	RAM [124]	RAM [123]	RAM [122]	RAM [121]	RAM [120]
0x80	RAM [135]	RAM [134]	RAM [133]	RAM [132]	RAM [131]	RAM [130]	RAM [129]	RAM [128]
0x88	RAM [143]	RAM [142]	RAM [141]	RAM [140]	RAM [139]	RAM [138]	RAM [137]	RAM [136]
0x90	RAM [151]	RAM [150]	RAM [149]	RAM [148]	RAM [147]	RAM [146]	RAM [145]	RAM [144]
0x98	RAM [159]	RAM [158]	RAM [157]	RAM [156]	RAM [155]	RAM [154]	RAM [153]	RAM [152]
0xA0	RAM [167]	RAM [166]	RAM [165]	RAM [164]	RAM [163]	RAM [162]	RAM [161]	RAM [160]
0xA8	RAM [175]	RAM [174]	RAM [173]	RAM [172]	RAM [171]	RAM [170]	RAM [169]	RAM [168]
0xB0	RAM [183]	RAM [182]	RAM [181]	RAM [180]	RAM [179]	RAM [178]	RAM [177]	RAM [176]
0xB8	RAM [191]	RAM [190]	RAM [189]	RAM [188]	RAM [187]	RAM [186]	RAM [185]	RAM [184]
0xC0	RAM [199]	RAM [198]	RAM [197]	RAM [196]	RAM [195]	RAM [194]	RAM [193]	RAM [192]
0xC8	RAM [207]	RAM [206]	RAM [205]	RAM [204]	RAM [203]	RAM [202]	RAM [201]	RAM [200]
0xD0	RAM [215]	RAM [214]	RAM [213]	RAM [212]	RAM [211]	RAM [210]	RAM [209]	RAM [208]
0xD8	RAM [223]	RAM [222]	RAM [221]	RAM [220]	RAM [219]	RAM [218]	RAM [217]	RAM [216]
0xE0	RAM [231]	RAM [230]	RAM [229]	RAM [228]	RAM [227]	RAM [226]	RAM [225]	RAM [224]
0xE8	RAM [239]	RAM [238]	RAM [237]	RAM [236]	RAM [235]	RAM [234]	RAM [233]	RAM [232]
0xF0	RAM [247]	RAM [246]	RAM [245]	RAM [244]	RAM [243]	RAM [242]	RAM [241]	RAM [240]
0xF8	RAM [255]	RAM [254]	RAM [253]	RAM [252]	RAM [251]	RAM [250]	RAM [249]	RAM [248]

11.3.2. RTC IP Control Registers

RTC has four register ports. Only word access, i.e. LW or SW, are allowed. Other type of access may cause unexpected result. Table 11-4 shows these registers.

Table 11-4 Control IO ports

Offset Address	Register Size (bit)	Register Symbol	Register Name
RTC Controller			
0xFB00	32	RTCCTL	Control and Status Register
0xFB04	32	RTCADR	Address Register
0xFB08	32	RTCDAT	Data port to access the contents of RTC registers
0xFB0C	32	RTCTBC	Time Base Corrector Register

Note: These registers are 32-bits register but only lower 8-bits are used. Upper 24-bits are reserved. Upper 24-bits return "0" for read operation.

11.3.3. Control and Status Register (RTCCTL) 0xFB00

Data Read from this port returns the status and issue command before. Data Write to this port issues command if the command code is valid.

RTCCTL	7	6	5	4	3	2	1	0
Field Definition	ALME	ALMD	BUSY	Reserved	Reserved	COMMAND		
R/W	R/W	R/W	R/O	R/O	R/O	R/W		
DEFAULT	X	X	0	0	0	X	X	X

RTCCTL	Bit	Description
RTCCTL[7]	1	Alarm Enable Control. "1" on this bit enables ALARM function to detect alarm and generate an interrupt.
RTCCTL[6]	1	Alarm Detection Signal. If RTC[47:16] matches with ALARM [47:16], this FF will be set to "1" and hold this value until (1) Set Alarm command issued or (2) Alarm Enable Control is disabled. This bit can be used form alarm test. When this bit set to "1", pseudo alarm signal will be generated that is same as real alarm detection (only if RTCCTL[7] is set). Write "0" to this bit will do nothing.
RTCCTL[5]	1	BUSY. This bit will be set to "1" right after command write to RTCCTL and return to "0" when the command operation complete.
RTCCTL[4:3]	2	Reserved.
RTCCTL[2:0]	3	Command for register transferring. See Table 11-5 in detail.

Figure 11-2 Port RTCCTL

Following Table 11-5 describes detail of command code for RTCCTL register.

Table 11-5 RTCCTL Command Code

Operation	CODE	Description
NOP	0	This command can be used to set or reset the ALARM Bit safely.
Get Time	1	Copy RTC [47:0] contents to RWB [47:0] in next 32.768 KHz rising clock edge. RTCADR [7:0] will be set to zero.
Set Time	2	Copy RWB [47:16] contents to RTC [47:16] in next 32.768 KHz rising clock edge. The RTC [15:0] will be cleared to zero. In addition, RTCADR [7:0] will be reset to zero.
Get Alarm	3	Copy the ALARM [47:16] contents to RWB [47:16] in next 32.768 KHz rising clock edge. RTCADR [7:0] will be set to zero.
Set Alarm	4	Copy RWB [47:16] contents to ALARM [47:16] in second 32.768 KHz rising clock edge. RTCADR [7:0] will be set to zero. Same time, the alarm detection FF will be reset.
NOP	5	This command can be used to set or reset the ALARM Bit safely.
NOP	6	This command can be used to set or reset the ALARM Bit safely.
NOP	7	This command can be used to set or reset the ALARM Bit safely.

11.3.4. Address Register (RTCADR) 0xFB04

The RTCADR address the RAM contents described in Table 11-3 Address Mapping of Exposed Registers. This register is R/W.
Default: 8'hxx

11.3.5. Data port to access the contents of RTC Register (RTCDAT) 0xFB08

Any of exposed register can be accessed through this port with the value of RTCADR [7:0] register. After each read or write operation, RTCADR [7:0] will increment one automatically. After 255 of RTCADR [7:0], it returns zero.
Default: 8'hxx

11.3.6. Time Base Corrector Register (RTCTBC) 0xFB0C

Time Base Corrector register. This register is used to adjust the 48 bit time count value. Refer to 11.4 for more detail.

RTCTBC	7	6	5	4	3	2	1	0
Field Definition	PM	COMP						
R/W	R/W	R/W						
DEFAULT	0	0	0	0	0	0	0	0

RTCTBC	Bit	Description
RTCTBC[7]	1	PM Direction of Correction PM = 1: Delay the time base with COMP[6:0] (unit PPM) PM = 0: Advance the time base with COMP[6:0] (unit PPM)
RTCTBC[6:0]	7	COMP[6:0] Compensation Value COMP[6:0] can be 0 to 127 (unsigned integer) If COMP[6:0] is set to non-zero value, then time base correction will happen. Direction of Correction is defined by the value of PM bit.

11.4. Time Base Calibration

11.4.1. Theory of Operation

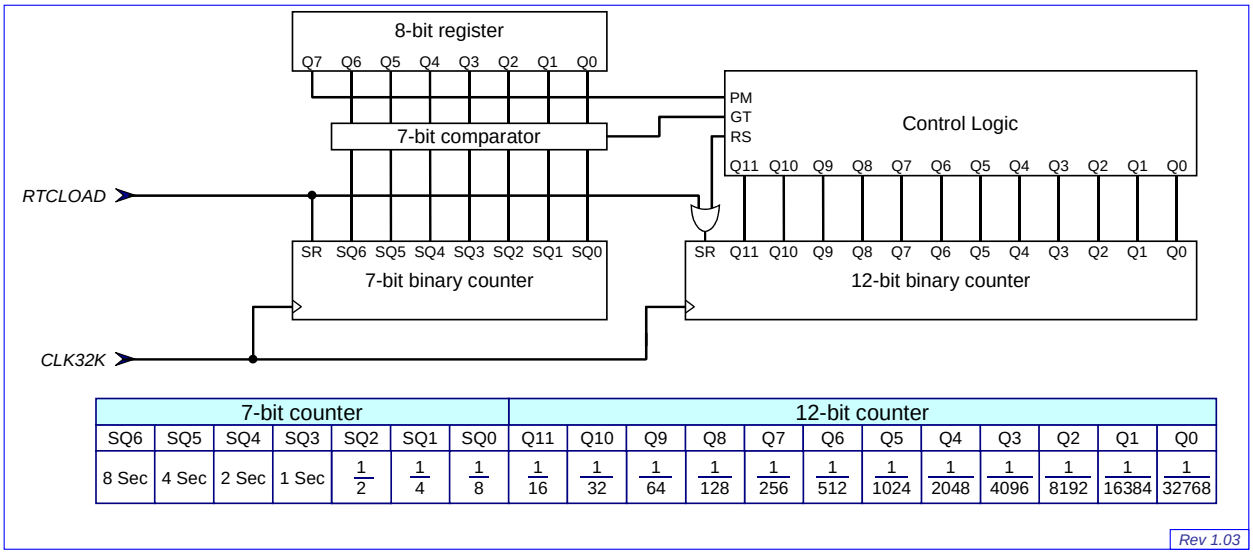
Real Time Clock uses an external crystal resonator. f_0 of this crystal resonator is 32,768 Hz. The accuracy can be selected from the grade. Those grades are ±20 PPM, ±50 PPM, and ±100 PPM.

On the other hand, the required accuracy of final product varies depend on application. In case of conventional Arm Watch, the accuracy is expressed by means of monthly error or annual error in terms of second unit. For example, ±15 seconds per month etc. Since a month (30days) consists of 2,592,000 seconds (=60 x 60 x 24 x 30), ± seconds per month corresponds to ±5.8 PPM (=15 / 2,592,000 x 1,000,000).

It is the common way to adjust frequency by using trimmer capacitance (i.e. variable capacitance) in factory. However, this adjustment work is not so convenient for factory automation.

This supplement provides an electrical means to adjust the 32,768 Hz frequency.

Figure 11-3 shows the skeleton of this time base corrector module.



7-bit counter							12-bit counter											
SQ6	SQ5	SQ4	SQ3	SQ2	SQ1	SQ0	Q11	Q10	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
8 Sec	4 Sec	2 Sec	1 Sec	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$	$\frac{1}{8192}$	$\frac{1}{16384}$	$\frac{1}{32768}$

Figure 11-3 Logic Diagram of Time Base Corrector

Internal to RTC there are 47-bit counter (RTC[47:1]). In Figure 11-3, 12-bit binary counter is corresponding to RTC[12:1]. And 7-bit binary counter is corresponding to RTC[19:13]. RTC[0] is always “0”. The 47-bit counter counts on every clock edge of 32.768 KHz oscillator clock (CLK32K).

This module is the part of divider circuit in RTC. It divides original 32,768 Hz clock binary manner. Since this divider chain consists of 19 flip-flops, actual divisor is 2^{19} (=524288). This number is approximately one million. Based on this condition, adding N clock(s) (N=1-127) per one round of this 19-stage counter delays 2N PPM in the RTC time base. Respectably, skipping 2N clock(s) advances 2N PPM.

When COMP[6:0] = 0000001 then N=1 or when COMP[6:0] = 1111111 then N=127

8-BIT REGISTER	7	6	5	4	3	2	1	0
FIELD Definition	PM	COMP [6:0]						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DEFAULT	0	0	0	0	0	0	0	0

Figure 11-4 Definition of 8-BIT REGISTER

Table 11-6 Bit Field Definitions

Bit Field	Symbol	Description
7	PM	Direction of Correction (Default = 0) PM = 1: Delay the time base with COMP [6:0] (unit PPM) PM = 0: Advance the time base with COMP [6:0] (unit PPM)
6:0	COMP[6:0]	Compensation Value (Default = 0) COMP [6:0] can be 0 to 127 (unsigned integer) If COMP [6:0] is set non-zero value, time base correction will happen. Direction of Correction is defined bt the value of PM bit.

The 7-BIT Counter counts 0 to 127, so that the output of 7-BIT_Comparator (i.e. GT) generates a same number of "1" signals with the value of COMP [6:0]. The control logic controls 12-BIT Counter. In normal case, this 12-BIT Counter counts 0 to 4095.

When the signal GT is 1 and PM equal 1, 12-BIT Counter will count …,4094,4095,0,0,1,2,… (repeat 0 twice). On the contrary, when the signal GT is 1 and PM equal 0, 12-BIT Counter will count …,4094,4095,0,2,3,… (skip 1). By this way, the time base correction will be done.

Chapter 12. Video Port

12.1. Feature

The video port controller(VPC) in TX4939 can be configured to support either 3 serial ports, or 1 parallel 8-bit port with 1 serial port. In the 3 serial port configuration, only one serial port can be configured as input or output while the other two serial ports are input only. In the 1 parallel port with 1 serial port configuration, both ports can be configured as input or output. Input mode supports receiving / storing externally digital video to DRAM. Output mode supports reading / sending DRAM data to external device.

- Dedicated DMA with chain mode
- configurable to 3 serial ports, or 1 parallel 8-bit port with 1 serial port
- 3 serial ports configuration
  - one serial port can be programmed to be input or output port
  - the other two serial ports are input only
- 8-bit parallel port and 1 serial port configuration
  - Programmable to be input or output port
- Support ITU-Bt.656 interface or EN50083-9's Synchronous serial and parallel interface

12.2. Block Diagram

Diagram below shows the overall structure of the TX4939 Video Port controller.

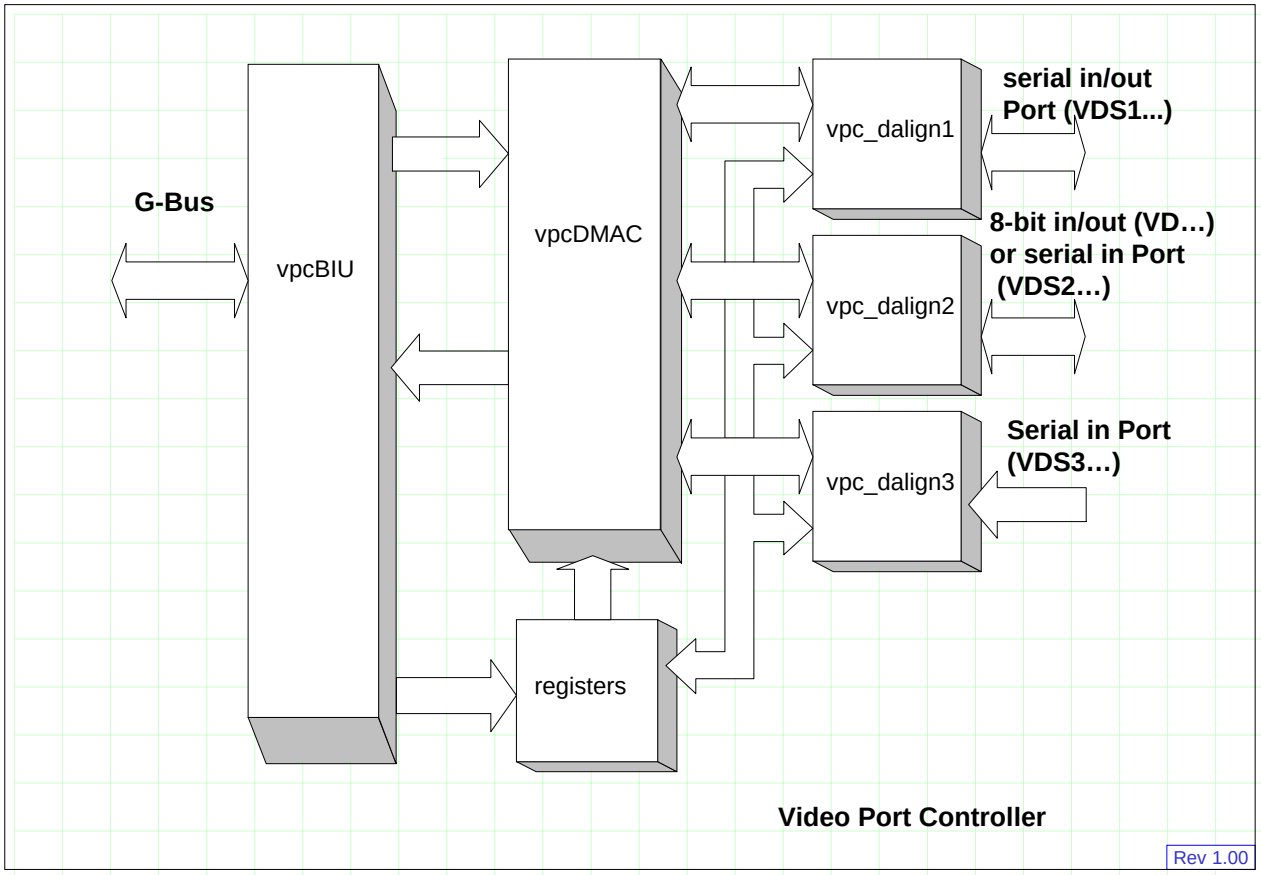


Figure 12-1 Block Diagram of Video Port Controller

12.3. Operations

Video Port Controller has two modes of operations.

- Output mode (transmit), in this mode the TX4939 is the master. VPC controller will begin to fetch the data from the memory and transfer it out to the video port.
- Input mode (capture), in this mode the TX4939 is the slave. VPC controller will begin to capture the data on the video port and transfer these data into the memory accordingly.

There are two errors condition that the VPC controller will check.

- Overflow error. When VPC is in capture mode, the overflow error occurs when there is an overflow in the capture FIFO. This condition is only happen when internal data bus can not transfer data out of the FIFO fast enough compare to the input data rate.
- Underflow error. When VPC is in transmit mode, the underflow error occurs when there is an underflow in the transmit FIFO. This condition is only happen when the internal data bus can transfer data into the FIFO fast enough compare to the output data rate.

Internal to the VPC controller, there is one 16x64 bit FIFO per port. This FIFO is shared for both transmit and capture mode.

12.3.1. Video Port DMA Controller

The Video Port DMA controller has the following features:

- Supports register access
- Supports master mode for G-bus protocol (single and burst transfer)
- Supports chain and link list DMA protocol
- Supports un-alignment byte data transfer for Transmit mode only. For Capture mode, all data are align to double word boundary (64-bit)

The VPC's DMA engine has one channel to transfer data between memory and the VPC engine. There are four tasks for the DMA.

- To get a descriptor
- To update the status in a descriptor when transfer complete
- To transfer received data from FIFO (receive mode) to memory
- To transfer transmit data from memory to FIFO (transmit mode).

There is one buffer per port inside VPC engine, this buffer shared for both transmit mode and receive mode.

Each video port has its own descriptor list. VPC's DMA engine processes the descriptor lists in round robin fashion.

12.3.2. 8-bit Parallel port

VPC supports two video formats. The two formats are ITU.Bt.656 and EN50083-9's Synchronous Parallel Interface. For the EN50083-9 format, data length and packet length are programmable from 128 bytes to 8192 bytes.

In TX4939, VDCLKIN clock is the source clock for VPC controller.

- VDCLKIN connect to VDCLKO directly.
- Internal controller uses VDCLKIN as clock source.

12.3.2.1. ITU.Bt656 Interface

This is a 8-bit ITU.Bt.656 (CCIR656) interface.

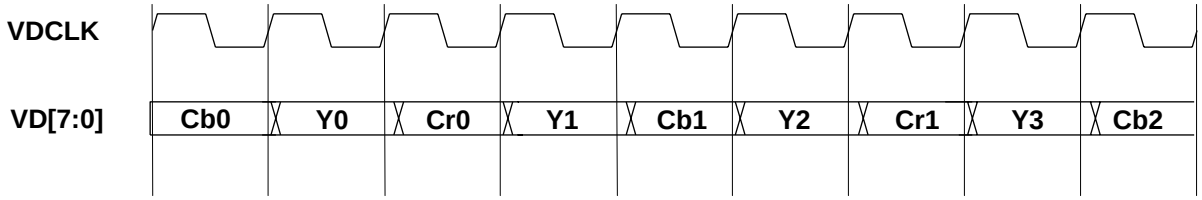


Figure 12-2 ITU.BT656 (CCIR656) Digital Video Timing

12.3.2.2. Synchronous Parallel Interface

The data to be tramsmitted are MPEG-2 Transport Packets with 188, 204 bytes or user defined. In the case of the 204 byte packet format packets may contains a 16 bytes "empty space", a VDVLD Signal serves to identify these dummy bytes. VDPSN flag is to identify the beginning of a packet. The data are synchronized to the VDCLK.

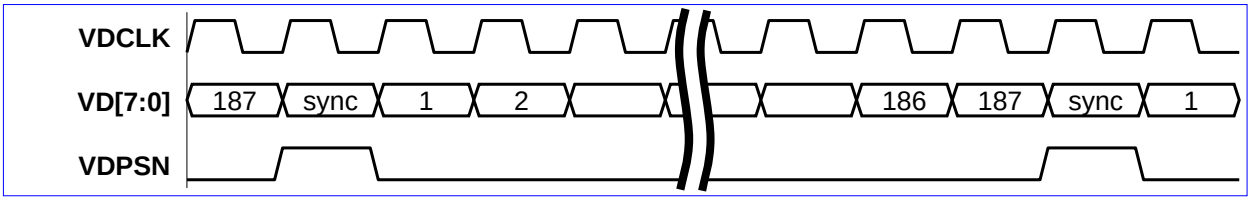


Figure 12-3 Transmission format with 188 Byte packets (VDVLD = 1)

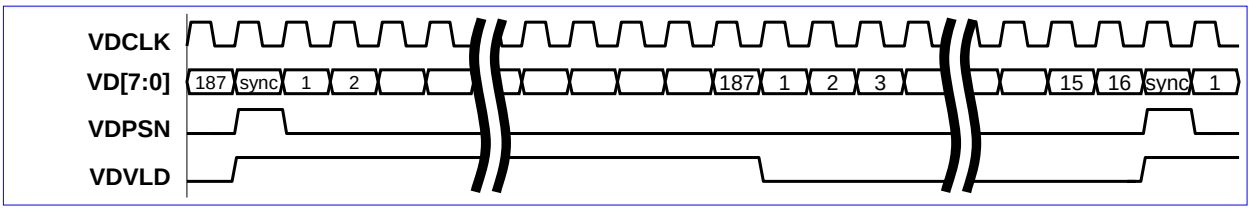


Figure 12-4 Transmission format with 204 Byte packets
(188 data bytes and 16 dummy bytes – dummy byte = 8'h00)

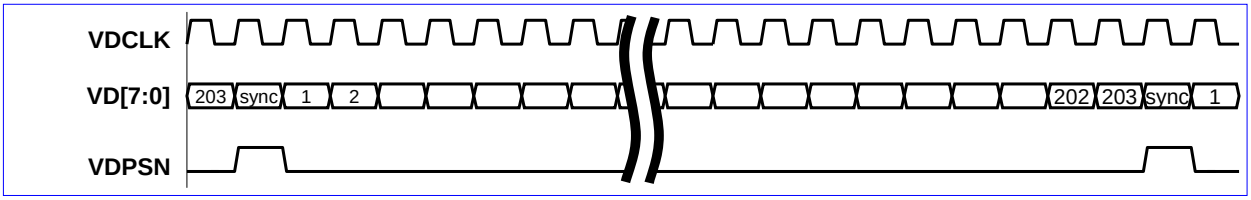


Figure 12-5 Transmission format with RS-coded packets (204 Bytes, VDVLD = 1)

12.3.3. Serial port

VPC serial interface support only the EN50083-9 format. Both data length and packet length are programmable from 128 bytes to 8192 bytes. Serial data is ordered from MSB to LSB. Width of VDPSN can be programmed to either 1-bit wide or 8-bit wide for output mode.

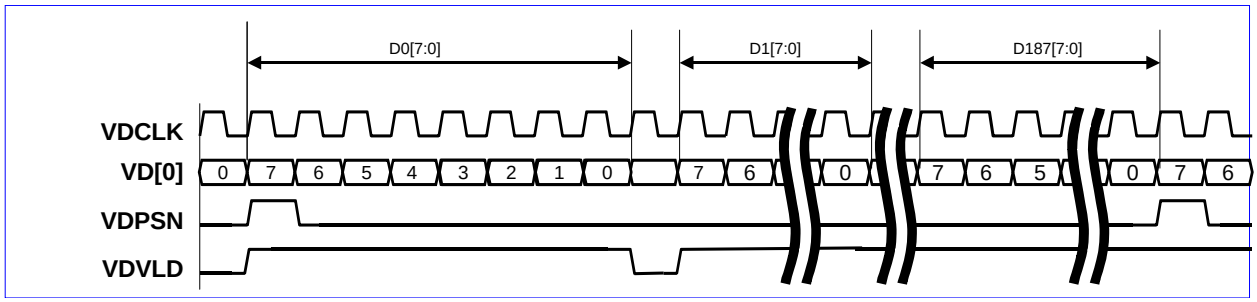


Figure 12-6 Serial transmission format with 188 Byte packets

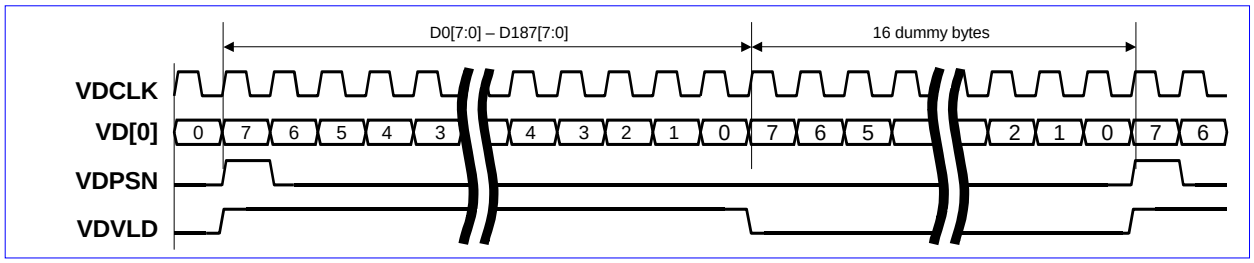


Figure 12-7 Serial transmission format with 204 Byte packets
(188 data bytes and 16 dummy bytes – dummy byte = 8'h00)

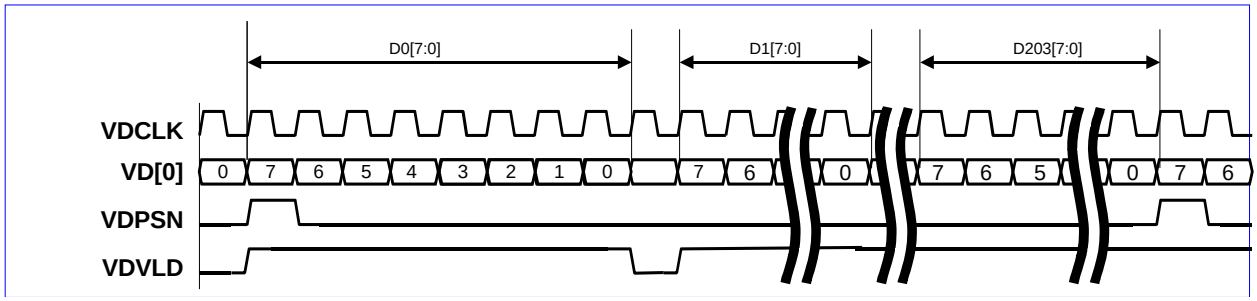


Figure 12-8 Serial transmission format with RS-coded packets (204 Bytes)

12.3.4. Data Format

12.3.4.1. ITU.Bt.656 Format

The video data format is in 4:2:2 color sub-sampling, each pair of color-difference samples will be intermixed with two luminance samples. The format for transmission shall be Cb, Y, Cr, Y. The following is a representation of the data format.

Byte0	Byte1	Byte2	Byte3
Cb0	Y0	Cr0	Y1
Cb1	Y2	Cr1	Y3
……			
Cb358	Y716	Cr358	Y717
Cb359	Y718	Cr359	Y719
……			

In main memory

- Transmit Video data can start at any byte address.
- Capture Video data must align to dword address.

12.3.4.2. MPEG2 data

- MPEG-2 Transport Stream (EN/ISO/IEC 13818-1)

```
Data:   | Syncbyte |          187 Data bytes          | Syncbyte |          187 Data bytes
Byte:     0     1  2  3  4 ----------------------------185 186 187   188     189 190 -------------------------
```

- MPEG-2 Transport Stream (ETS 300 421)

```
Data:   | Syncbyte |          203 Data bytes          | Syncbyte |          203 Data bytes
Byte:     0     1  2  3  4 ----------------------------201 202 203   204    205 206 207 -------------------
```

In main memory

- Transmit video data can start at any byte address.
- Capture Video data must align to dword address.

12.3.5. Transmit Window Option

For transmit mode, TX4939 has special option to transfer a particular window of video data out to the video port.

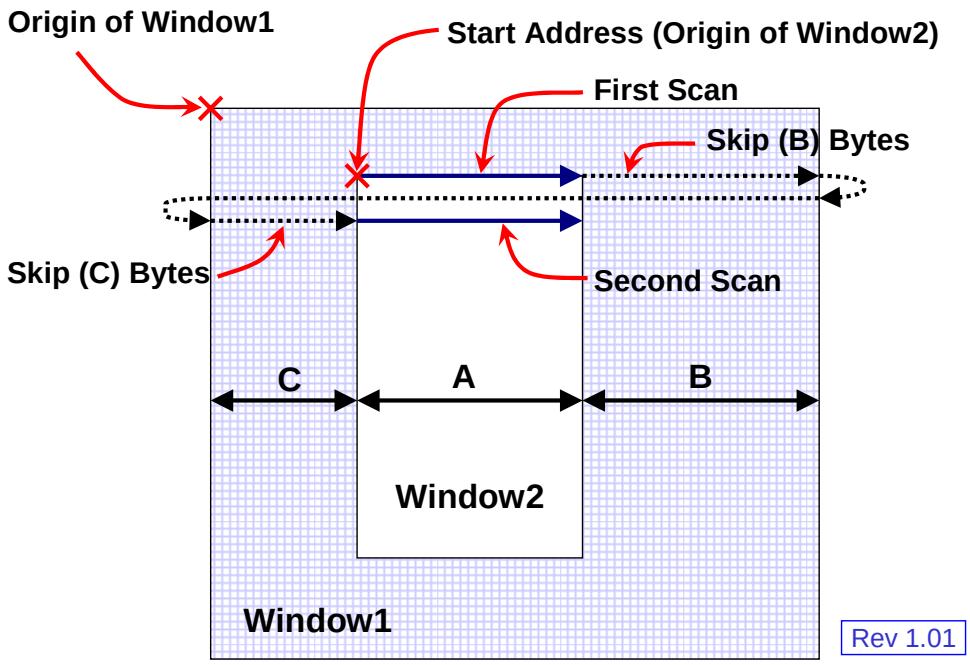


Figure 12-9 Transmit Window 2 data to video port

In Figure 12-9 is an example of the Transmit Window Option. In the system memory, there is Window 1 data. These video data are continuous in the memory. But the application want to transfer Window2 data output to the Video Port.

In the Control 1 Descriptor, there are two field

DMAGA[31:0] should be programmed to point to the Start Address(Origin of Window2) as shown in the Figure 12-9

Bcount[22:0] – this indicates the total number of bytes that the VPC want to transfer out. This bytecount is the same as number of byte in Window2 in Figure 12-9

Note: **Bcount[22:0] must be multple of validCount[11:0].**
When Bcount[11:0] is not multple of validCount[11:0], undefined behavioral will occur in VPC.

In the Control 2 Descriptor, there are two field

validCount[11:0] specify the valid byte which equivalent to symbol A in the Figure 12-9

Offset[11:0] specifies the offset of the next address for the video data. Which is equivalent to the value of symbol B plus symbol C in Figure 12-9

With the above setting, the VPC can transfer only the Video data in Window2 out to the Video port.

Example:

Memoy has the following byte data:

```
byte # - 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
data   - a b c d e f g h i j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z  A  B  C

byte # - 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
data   -  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W
```

With validCount[11:0] = 10(11 bytes), Offset[11:0] = 4 (5 bytes) and Bcount[22:0] = 32(33 bytes) data send out by VPC will have the following pattern assuming dma address(DMAGA[31:0] = 0) is pointing to byte 0:

```
a b c d e f g h i j  k q r s t u v w x y z A G  H  I  J  K  L  M  N  O  P  Q
```

Bytes #10 to #15 and #27 to #31 are skipped because of the Offset[11:0] = 4.

In this example, Bcount[22:0] can be up to

```
8 Mbyte - (8Mbyte % 11) - 1 = 8388599 (0x7FFFF7)
```

MPEG Transmit

If VPC is programmed to transmit packet in MPEG format with Vddata = 187(188 byte data packet) and Bcount[22:0] greater than 187(more than 188 bytes), the number of packets will be sent by VPC is

```
((Bcount + 1)  / (Vddata + 1)) + ((((Bcount + 1) % (Vddata + 1)) != 0) ? 1 : 0)
```

For example, if vdfor = 1, Vddata = 187 and bcount = 189 (190 bytes), 2 packets will be sent out by VPC. Second packet will have 0s after 2 bytes of valid data:

	0	**1**	**2**	**3**	**4**	**…**	**187**
packet #0	D0	D1	D2	D3	D4	…	D187
packet #1	D188	D189	0	0	0	…	0

12.3.6. Video Port Controller Registers

All registers are 32 bit . Firmware must use Word instruction to access to these registers.

Table 12-1 VPR Registers

Offset	Register	Width	R / W	Description
0xA000	CSR	[31:0]	R/W	Control and Status Register
0xA008	CtrlA1	[31:0]	R/W	Video Port 1 Control A Register
0xA010	CtrlB1	[31:0]	R/W	Video Port 1 Control B Register
0xA018	IDESPtr1	[31:0]	R/W	Video Port 1 Initial Descriptor Pointer Register
0xA020	CDESPtr1	[31:0]	R	Video Port 1 Current Descriptor Pointer Register
0xA028	CtrlA2	[31:0]	R/W	Video Port 2 Control A Register
0xA030	CtrlB2	[31:0]	R/W	Video Port 2 Control B Register
0xA038	IDESPtr2	[31:0]	R/W	Video Port 2 Initial Descriptor Pointer Register
0xA040	CDESPtr2	[31:0]	R	Video Port 2 Current Descriptor Pointer Register
0xA048	CtrlA3	[31:0]	R/W	Video Port 3 Control A Register
0xA050	CtrlB3	[31:0]	R/W	Video Port 3 Control B Register
0xA058	IDESPtr3	[31:0]	R/W	Video Port 3 Initial Descriptor Pointer Register
0xA060	CDESPtr3	[31:0]	R	Video Port 3 Current Descriptor Pointer Register
0xA068	BusErr	[31:0]	R	Bus Error Address Register

12.3.6.1. Control and Status Register (CSR) 0xA000

Table 12-2 Control and Status Register (CSR)

Bit(s)	Field	R/W	Default	Description
63:17	–	RO	0	Reserved
16	GBINT	RO	0	G-Bus Error Interrupt 0: No interrupt 1: G-Bus error This bit will be clear after read cycle.
15:6	–	RO	0	Reserved
5	SWAPO	R/W	0	Swap Transmit Data 0: no swapping 1: byte swapping enabled
4	SWAPI	R/W	0	Swap Receive Data 0: no swapping 1: byte swapping enabled
3	GINTE	R/W	0	Interrupt Enable 0: Disable All Interrupts. 1: Enable All Interrupts. This bit has to be program before start VPC engine
2	RSTD	R/W	0	Reset DMA Controller 0: Normal 1: Reset After set this bit, need to reset it back to “0” for normal operation.
1	RSTVPC	R/W	0	Reset Video Port interface logic 0: Normal 1: Reset After set this bit, need to reset it back to “0” for normal operation.
0	–	RO	0	Reserved

12.3.6.2. ControlA (CtrlA)

CtrlA1 0xA008
CtrlA2 0xA028
CtrlA3 0xA048

Table 12-3 ControlA Register (CtrlA)

Bit(s)	Field	R/W	Default	Description
63:10	–	RO	0	Reserved
9	VDPSN	R/W	0	VDPSN bit width selection in serial output mode. 0: VDPSN is 8-bit wide 1: VDPSN is 1-bit wide
8	PBUSY	RO	0	Port Busy Status 1: Port is busy 0: Port is idle This bit is useful for the transmit case in which DMA completion interrupt can be generated way ahead of the port transmission completion. System should check this status bit before issuing new operation to the same port. Otherwise, the new operation may terminate an ongoing transmission operation prematurely. NOTE: Default value is only valid when the video port's corresponding video clock is supplied while a reset is applied.
7	DCINT	RO	0	DMA Completion Interrupt 0: No interrupt 1: Transfer complete This bit will be clear after read cycle.
6	UOINT	RO	0	Underflow/Overflow Error Interrupt 0 : No error 1 : Underflow/Overflow error Overflow error only occurs when VPC is in capture mode. Underflow error only occurs when VPC is in transmit mode.
5:4	PDINT	R/W	01	Programmable DMA Completion Interrupt 00: Interrupt for every descriptor at completion of transfer. 01: Interrupt only when Next Descriptor pointer = Null at completion of transfer. 10: Interrupt only when Next Descriptor pointer = Null at completion of transfer. 11: Disable DMA Completion Interrupt
3	Vdvldp	R/W	0	VDVLD polarity select 0: Active high (default) 1: Active low
2	Vdmode	R/W	0	Video Port Mode***** 0: output mode (Master) – transmit (default) 1: input mode (Slave) – receive
1	VDFOR	R/W	0	Video Format 0: ITU.Bt.656 (default) 1: MPEG-2 Transport Stream
0	ENVPC	R/W	0	Start VPC Engine 0: Idle 1: Start When write "1" to this bit, VPC engine will begin to operate.

Note: Port configuration is selected by TX4939's configuration setting.

***** Video Port Mode is only applicable to video port 1 and video port 2. This bit is programmable for video port 3 but since **video port 3 is input port only, software must program this bit to 1**. Programming 0 will cause undefined behavioral.

12.3.6.3. ControlB (CtrlB)

CtrlB1 0xA010
CtrlB2 0xA030
CtrlB3 0xA050

Table 12-4 ControlB Register (CtrlB)

Bit(s)	Field	R/W	Default	Description
63:29	–	RO	0	Reserved
28:16	Vddata	R/W	187	Video data length 0-126: unsupported* 127: 128 Byte packets 128: 129 Byte packets 129: 130 Byte packets …. 187: 188 Byte packets (default) …. 8190: 8191 Byte packets 8191: 8192 Byte packets This field is only used when select MPEG-2 Transport Stream as video format.
15:13	–	RO	0	Reserved
12:0	Vdpacket	R/W	187	Video packet length 0-126: unsupported* 127: 128 Byte packets 128: 129 Byte packets 129: 130 Byte packets …. 187: 188 Byte packets (default) …. 8190: 8191 Byte packets 8191: 8192 Byte packets This field is only used when select MPEG-2 Transport Stream as video format.

Note: Vdpacket must be equal to or greater than Vddata. Programming Vdpacket to be smaller than Vddata will cause undefined behavioral in VPC.

*Programming Vddata and Vdpacket to a value less than 127 may cause undefined behavioral in VPC.

12.3.6.4. Initial Descriptor Pointer Register

IDESPtr1 0xA018
IDESPtr2 0xA038
IDESPtr3 0xA058

Table 12-5 Initial Descriptor Pointer Register (IDESPtr)

Bit(s)	Field	R/W	Default	Description
63:32	–	RO	0	Reserved
31:0	IDESPtr	R/W	0	Initial Descriptor Pointer. When ENVPC set to “1” the DMA engine begin to fetch the first Descriptor at this address

12.3.6.5. Current Descriptor Pointer Register

CDESPtr1 0xA020
CDESPtr2 0xA040
CDESPtr3 0xA060

Table 12-6 Current Descriptor Pointer Register (CDESPtr)

Bit(s)	Field	R/W	Default	Description
63:32	–	RO	0	Reserved
31:0	CDESPtr	RO	0	Current Descriptor Pointer. - This register contents the address of the Descriptor that the DMA engine is currently pointing to.

12.3.6.6. Bus Error Address Register (BusErr) 0xA068

Table 12-7 Bus Error Address Register (BusErr)

Bit(s)	Field	R/W	Default	Description
63:32	–	RO	0	Reserved
31:2	--	RO	0	G-bus error address[31:2]
1	--	RO	0	Command status during bus error. 0: Write operation 1: Read operation
0	--	RO	0	When set, indicates the address is valid, i.e. Bus Error has been captured.

Note:

1- Programmer must set TOE bit in the CCFG register to enable timeout interrupts.
2- The VPC actually does not generate any bus errors. So this is not handled under BusError handler in CPU. Instead it is handled under regular interrupts. Care should be observed in this case for EPC register in COP0
3- Enable GBINT in VPC CSR. This will cause an INT to be generated for the CPU
4- This is now an INT handler not a BUSERR (Both share the same vector but handled differently)
5- Inside the handler, software must read CSR to clear GBINT and issue software reset (DMAC reset - RSTD), to clear BusErr register.

12.3.7. Descriptor Format

All Video Port's descriptors are 32-bit, must use LW and SW to access to these descriptor.

Table 12-8 VPC Descriptor Table

Description	Offset
Input/Output DMA Address Descriptor	'h00/'h08
Next Descriptor Pointer Descriptor	'h04/'h0c
Control 1 Descriptor	'h08/'h10
Control 2 Descriptor	'h0c/'h14

Table 12-9 Input Source Address Descriptor

Field	Name	Description
31:0	DMAGA[31:0]	DMA G-Bus Address[31:0] Byte addressable

Table 12-10 Next Descriptor Pointer Descriptor

Field	Name	Description
31:2	nxtptr[31:2]	Address of the next descriptor. Null ('h0000_0000) = End of chain.
1:0	-	Reserved.

Table 12-11 Control 1 Descriptor

Field	Name	Description
31	----	Reserved.
30:8	Bcount[22:0]	Byte Count - up to 8M byte 23'h000000 = 1 byte 23'h000001 = 2 bytes 23'h000002 = 3 bytes …… 23'h7FFFFF = 8M bytes - For Capture mode minimum byte count must be 8 bytes.
7:3	---	Reserved.
2:1	Error	VPC will set these bits when there is an error occurred when process this Descriptor 00 = no error 01 = Underflow/Overflow error 10 = reserved. 11 = reserved. When error occurred, DMA engine will skip the rest of the operation until it see end of Descriptor. Before starting process next descriptor.
0	OWN	When set, indicates that the descriptor is owned by VPC controller. When reset, indicates that the descriptor is owned by the CPU. VPC controller clears this bit when it completes the DMA.

Table 12-12 Control 2 Descriptor

Field	Name	Description
31:20	validCount[11:0]	Byte Valid Count – up to 4K byte 12'h000 = disable 12'h007 = 8 byte (minimum) 12'h008 = 9 byte …… 12'hFFF = 4096 bytes When validcount[11:0] = 12'h000, which means all video data are continuous. If validCount[11:0] are not zero then once validCount reach, the next video data address = current address + offset. Once the Valid count reach, then it then starts to count from 0 again.
19:8	Offset[11:0]	Offset - up to 4K 12'h000 = 1 byte 12'h001 = 2 bytes …… 12'hFFF = 4096 bytes Only valid when validcount[11:0] are not zero. Once validCount reach then the next video data address = current address + offset.
7:0	---	Reserved.

Notes: Control 2 Descriptor is only use when VPC is in transmit mode. When VPC is in capture mode, programming the validCount[11:0] to non-zero value may cause error in VPC engine.

12.3.8. Big Endian Support

VPC is designed to process data in 64-bit(double word) chunk in Little Endian format. Byte ordering within a double word in Little Endian format is shown below:

Byte #	7	6	5	4	3	2	1	0
Address 0x00000000	h	g	f	e	d	c	b	a
Address 0x00000008	p	o	n	m	l	k	j	i

For output mode, VPC will transmit byte data from byte 0 to byte 7 starting from address 0x00000000 then address 0x00000008.

Data output seen on VD[7:0] would be

a b c d e f g h I j k l m n o p q

For input mode, VPC will pack the byte data received from byte 0 to byte 7 to form a double word to write to system memory.

In Big Endian mode, VPC assumes byte ordering is reverse that of Little Endian format:

Byte #	0	1	2	3	4	5	6	7
Address 0x00000000	a	b	c	d	e	f	g	h
Address 0x00000008	I	j	k	l	m	n	o	p

VPC has byte swapper logic to swap these byte data to Little Endian format before processing. There is one swapper for output and one swapper for input. The input and output swappers are both off normally in Little Endian mode and is turned on automatically when Big Endian mode is selected at the 4939 chip level. The CSR register has 2 bits, SWAPI and SWAPO, to allow software to turn these 2 swappers on or off independently as shown in the tables below.

	Little Endian		Big Endian	
SWAPI	0	1	0	1
Input Swapper	off	on	on	off

	Little Endian		Big Endian	
SWAPO	0	1	0	1
Output Swapper	off	on	on	off

Chapter 13. Timer/Counter Controller

13.1. Features

The TX4939 has an on-chip 6-channel timer/counter.

- *32-bit Up Counter: 6 Channels*
  - *Featured as down-counter like usage*
- *Interval Timer Mode (Channel 0, 1, 2, 3, 4, 5)*
- *Pulse Generator Mode (Channel 0, 1)*
- *Watchdog Timer Mode (Channel 2)*
- *Timer Output Signal (TIMER[1:0]) × 2*
- *Watchdog Timer Reset Output (WDRST*): × 1*

Note: This timer function has additional feature from TX4938's TIMER as follows.

The register TMTRRn (Timer Read Register) was read only.
In TX4939, this register accepts data write to set appropriate 32-bit data as a counter starting value.

Although, this register is now R/W (Read/Write), but the register name remain same for the software compatibility.

13.2. Block Diagram

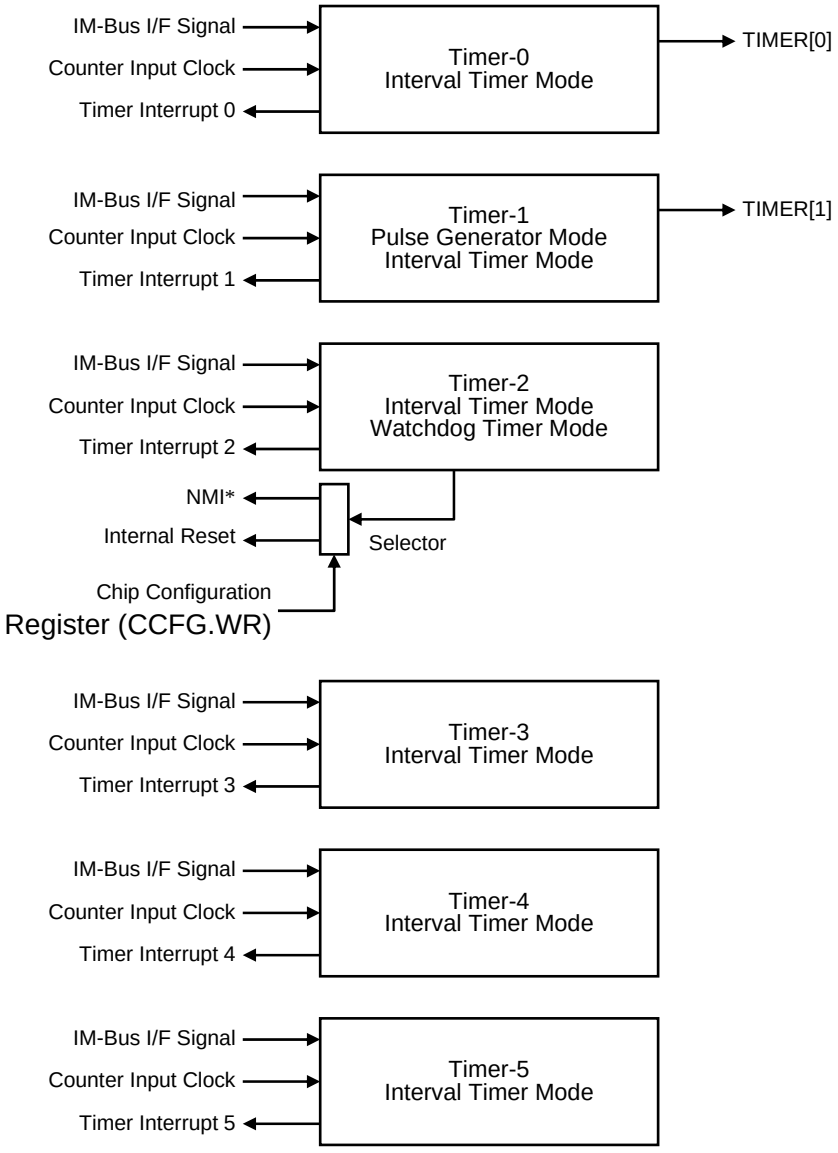


Figure 13-1 Connecting Timer Module Inside the TX4939

13

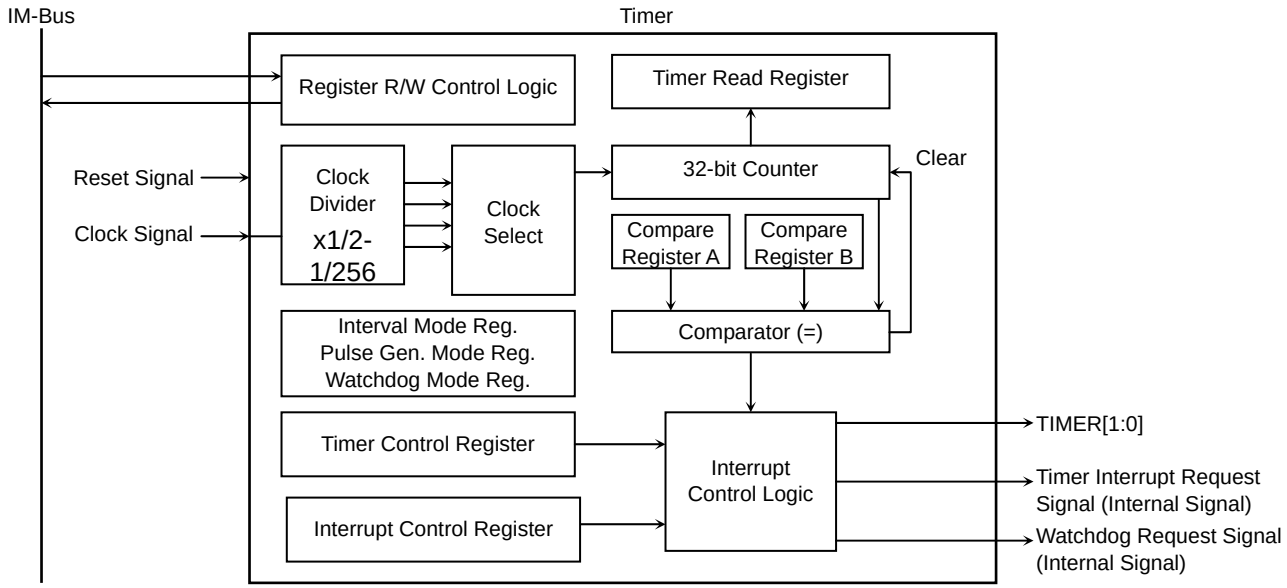


Figure 13-2 Timer Internal Block Diagram

13

13.3. Detailed Explanation

13.3.1. Overview

The TX4939 has an on-chip 6-channel 32-bit timer/counter. Each channel supports the following modes.

- *Interval Timer Mode (Timer 0, 1, 2, 3, 4, 5)*
  This mode periodically generates interrupts.
- *Pulse Generator Mode (Timer 0, 1)*
  This is the pulse signal output mode.
- *Watchdog Timer Mode (Timer 2)*
  This mode is used to monitor system abnormalities.

13.3.2. Counter Clock

The clock used for counting can be set to a frequency that is 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of the internal clock (IMBUSCLK) frequency, or can be selected from nine counter input signal (TCLK) types. Divide Register *n* (TMCCDRn) and the Counter Clock Select bit (TMTCRn.CCS) are used to select the counter clock. In this situation, IMBUSCLK is the internal clock signal which is the G-Bus clock divided by 2. See “Chapter 5 Clock Generator” for more information.

The counter input signal (TCLK) is used by three channels. Using TCLK makes it possible to count external events. The External Clock Edge bit (TMTCRn.ECES) can be used to select the clock rising/falling count.

Note that TX4939 does not support TCLK.

Set the TCLK clock frequency to 45% or less of IMBUSCLK (TCLK = 22.5 MHz or less when IMBUSCLK = 50 MHz). The following table shows example count times when using 50 MHz IMBUSCLK.

Table 13-1 Divide Value and Count (IMBUSCLK = 100 MHz)

Divide Rate	TMCCDRn. CCD	Counter Clock Frequency (Hz)	Resolution (ns)	Max. Set Time (sec.)	TMCPRAn Value for 1 sec.
2	000	50.0 M	20.00	85.90	50000000
4	001	25.0 M	40.00	171.80	25000000
8	010	12.5 M	80.00	343.60	12500000
16	011	6.25 M	160.00	687.19	6250000
32	100	3.125 M	320.00	1374.39	3125000
64	101	1.5625 K	640.00	2748.78	1562500
128	110	781.25 K	1280.00	5497.56	781250
256	111	390.625 K	2560.00	10995.12	390625

13.3.3. Counter

Each channel has an independent 32-bit counter. Set the Timer Count Enable bit (TMTCRn.TCE) and the 32-bit counter will start counting.

Clear the Timer Count Enable bit to stop the counter. If the Counter Reset Enable bit (TMTCRn.CRE) is set, then the counter will be cleared also. The Watchdog Timer Disable bit (TMWTRM2.WDIS) must be set in order to stop and clear this counter when in the Watch Dog Timer mode.

Also, reading the Timer Read Register (TMTRR) makes it possible to fetch the counter value.

13.3.4. Interval Timer Mode

The Interval Timer mode is used to periodically generate interrupts. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to “00” sets the timer to the Interval Timer mode. This mode can be used by all timers.

When the count value matches the value of Compare Register A (TMCPRAn), the Interval Timer TMCPRA Status bit (TMTISRn.TIIS) of the Timer Interrupt Status Register is set. When the Interval Timer Interrupt Enable bit (TMITMRn.TIIE) of the Interval Timer Mode Register is set, timer interrupts occur. When a “0” is written to the Interval Timer TMCPRA Status bit (TMTISRn.TIIS), TIIS is cleared and timer interrupts stop.

If the Timer Zero Clear Enable bit (TMITMRn.TZCE) is set, the counter is cleared to 0 if the count value matches the Compare Register A (TMCPRAn) value. Count operation stops when the Timer Zero Clear Enable bit (TMITMRn.TZCE) is cleared.

The level of the TIMER[1:0] output signal stays in the initial state (Low) in this mode. Output is undefined when changing from the Pulse Generator mode to this mode. Figure 13-3 shows an outline of the count operation and generation of interrupts when in the Interval Timer mode and Figure 13-4 shows the operation when using an external input clock.

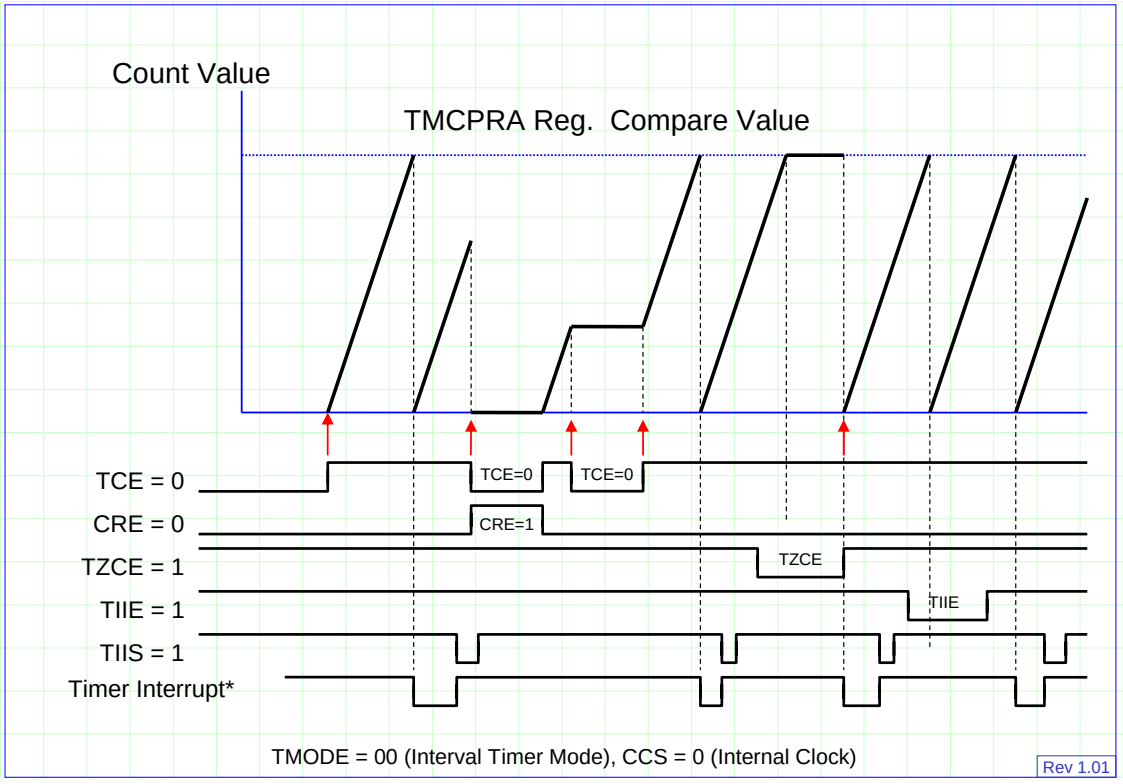


Figure 13-3 Operation Example of Interval Timer (Using Internal Clock)

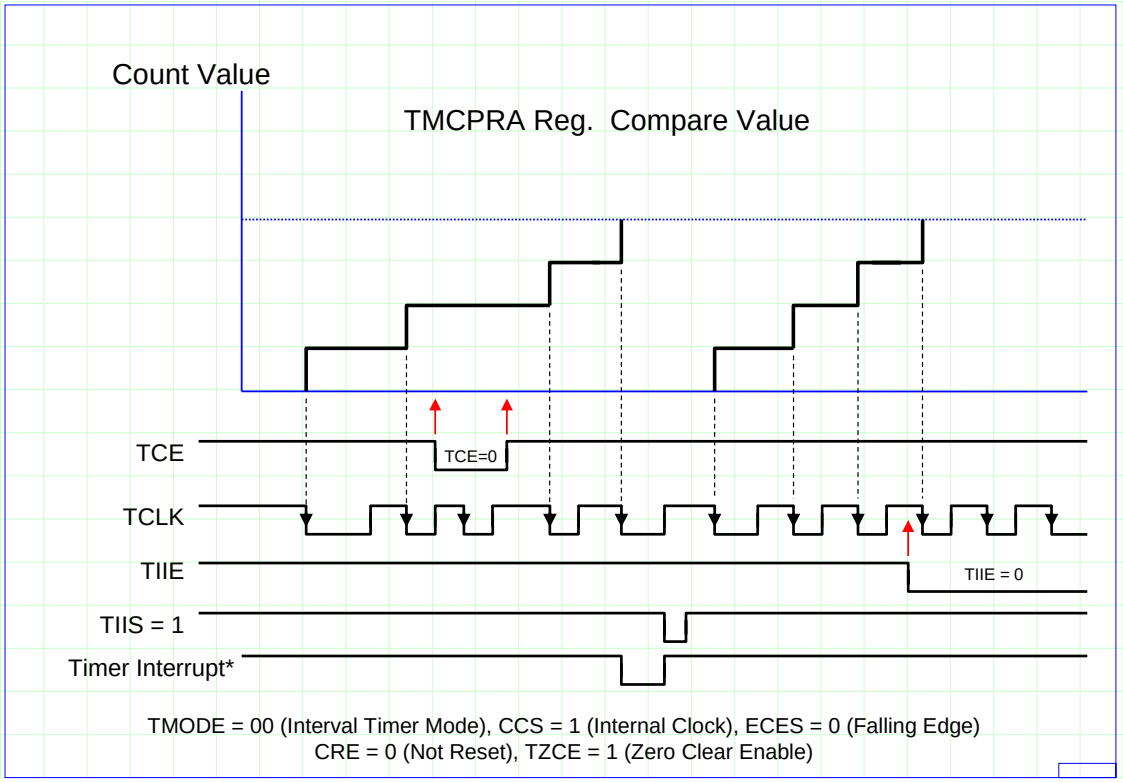


Figure 13-4 Operation Example of the Interval Timer (External Input Clock: Falling Edge Operation)

13.4. Pulse Generator Mode

When in the Pulse Generator mode, use Compare Register A (TMCPRAn) and Compare Register B (TMCPRBn) to output a particular period and particular duty square wave to the TIMER[n] signal. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to “01” sets the timer to the Pulse Generator mode. Timer 0 and Timer 1 can be used, but Timer 2 cannot.

The initial state of the TIMER[n] signal can be set by the Flip Flop Default bit (TMPGMRn.FFI) of the Pulse Generator Mode Register.

The TIMER[n] output signal reverses when the counter value matches the value set in Compare Register A (TMCPRAn). The TIMER[n] output signal reverse again, clearing the counter when the counter continues counting and the value set in Compare Register B (TMCPRBn) and the counter value match. Consequently, a value greater than that in Compare Register A (TMCPRAn) must not be set in Compare Register B (TMCPRBn).

Interrupts can be generated in the Pulse Generator mode as well. However, this is not standard practice.

The Pulse Generator TMCPRA Status bit (TMTISRn.TPIAS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register A (TMCPRAn). Timer interrupts are generated when the TMCPRA Interrupt Enable bit (TMPGMRn.TPIAE) of the Pulse Generator Mode Register is set.

Similarly, the Pulse Generator TMCPRB Status bit (TMTISRn.TPIBS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register B (TMCPRBn). Timer interrupts are generated when the TMCPRB Interrupt Enable bit (TMPGMRn.TPIBE) of the Pulse Generator Mode Register is set.

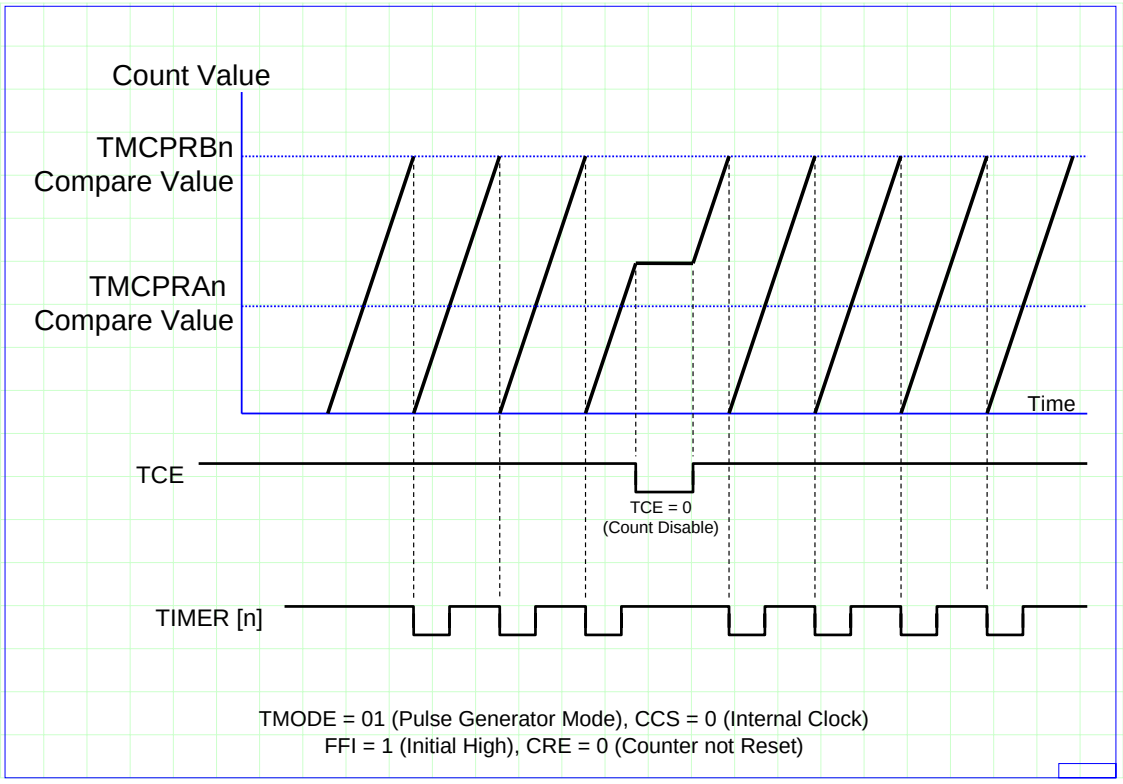


Figure 13-5 Operation Example of the Pulse Generator Mode

13.5. Watchdog Timer Mode

The Watchdog Timer mode is used to monitor system anomalies. The software periodically clears the counter and judges an anomaly to exist if the counter is not cleared within a specified period of time. Then, either the TX4939 is internally reset or an NMI is signaled to the TX49/H4 core. Set the Timer mode field (TMTCR2.TMODE) of the Timer Control Register to “10” to set the timer to the Watchdog Timer mode. This mode can only be used by Timer 2.

Use the Watchdog Reset bit (WR) of the Chip Configuration Register (CCFG) to select whether to perform an internal reset or signal an NMI. Set this bit to “1” to select Watchdog Reset, or set it to “0” to select NMI Signaling.

When the timer count reaches the value programmed in Compare Register A (TMCPRA2), the Watchdog Timer TMCPRA Match Status bit in the Timer Interrupt Status Register (TMTISR2.TWIS) is set. Either the watchdog timer reset or NMI is issued if the Timer Watchdog Enable bit in the Watchdog Timer Mode Register (TMWTMR2.TWIE) is set.

When the watchdog timer reset is selected, the Watchdog Reset Status bit in the Chip Configuration Register (CCFG.WDRST) is set. If the Watchdog Reset External Output bit in the Chip Configuration Register (CCFG.WDREXEN) is cleared, the entire TX4939 is initialized but the configuration registers. If the CCFG.WDREXEN bit is set, the WDRST* signal is asserted and remains asserted until the RESET* signal is asserted.

There are three ways of stopping NMI signaling from being performed.

Clear the Watchdog Timer Interrupt Status bit (TMTISR2.TWIS) of the timer Interrupt Status Register.
Clear the counter by writing “1” to the Watchdog Timer Clear bit (TMWTMR2.TWC) of the Watchdog Timer Mode Register.
Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE) while the Watchdog Timer Disable bit (TMWTMR2.WDIS) is still set.

It is possible to stop the counter when in the Watchdog Timer mode by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register while the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set to “1”.

It is also possible to stop the counter by clearing the Counter Clock Divide Cycle Enable bit (TMTCR2.CCDE) of the Timer Control Register when the internal clock is being used as the counter clock.

It is not possible to directly write “0” to the Watchdog Timer Disable bit (TMWTMR2.WDIS). There are two ways to clear this bit.

Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.WDIS)
Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register

In Watchdog Timer mode, the TIMER[1:0] outputs remain at logic high.

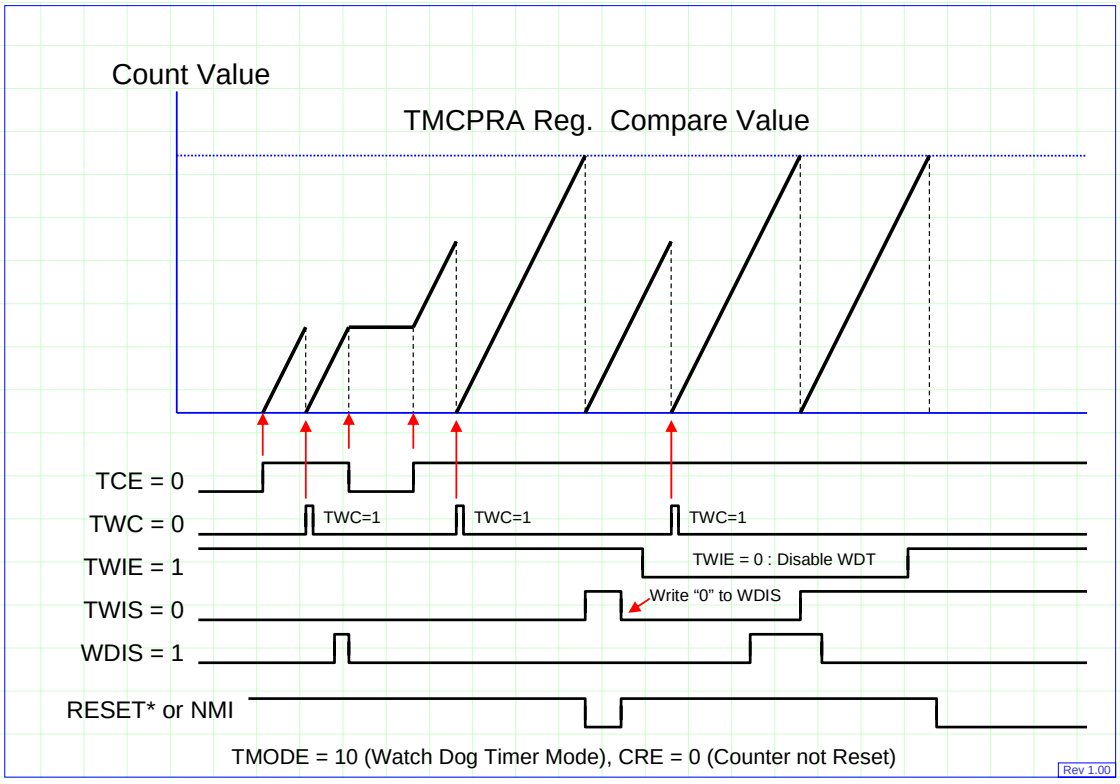


Figure 13-6 Operation Example of the Watchdog Timer Mode

13.6. Registers

Table 13-2 Timer Register List

Offset Address	Register Symbol	Register Name
		Timer 0 (TMR0)
0xF000	TMTCR0	Timer Control Register 0
0xF004	TMTISR0	Timer Interrupt Status Register 0
0xF008	TMCPRA0	Compare Register A 0
0xF00C	TMCPRB0	Compare Register B 0
0xF010	TMITMR0	Interval Timer Mode Register 0
0xF020	TMCCDR0	Divide Cycle Register 0
0xF030	TMPGMR0	Pulse Generator Mode Register 0
0xF040	TMWTMR0	(Reserved)
0xF0F0	TMTRR0	Timer Read Register 0
		Timer 1 (TMR1)
0xF100	TMTCR1	Timer Control Register 1
0xF104	TMTISR1	Timer Interrupt Status Register 1
0xF108	TMCPRA1	Compare Register A 1
0xF10C	TMCPRB1	Compare Register B 1
0xF110	TMITMR1	Interval Timer Mode Register 1
0xF120	TMCCDR1	Divide Cycle Register 1
0xF130	TMPGMR1	Pulse Generator Mode Register 1
0xF140	TMWTMR1	(Reserved)
0xF1F0	TMTRR1	Timer Read Register 1
		Timer 2 (TMR2)
0xF200	TMTCR2	Timer Control Register 2
0xF204	TMTISR2	Timer Interrupt Status Register 2
0xF208	TMCPRA2	Compare Register A 2
0xF20C	TMCPRB2	(Reserved)
0xF210	TMITMR2	Interval Timer Mode Register 2
0xF220	TMCCDR2	Divide Cycle Register 2
0xF230	TMPGMR2	(Reserved)
0xF240	TMWTMR2	Watchdog Timer Mode Register 2
0xF2F0	TMTRR2	Timer Read Register 2
		Timer 3 (TMR3)
0xFD00	TMTCR3	Timer Control Register 3
0xFD04	TMTISR3	Timer Interrupt Status Register 3
0xFD08	TMCPRA3	Compare Register A3
0xFD0C	TMCPRB3	(Reserved)
0xFD10	TMITMR3	Interval Timer Mode Register 3
0xFD20	TMCCDR3	Divide Cycle Register 3
0xFD30	TMPGMR3	(Reserved)
0xFD40	TMWTMR3	(Reserved)
0xFDF0	TMTRR3	Timer Read Register 3
		Timer 4 (TMR4)
0xFE00	TMTCR4	Timer Control Register 4
0xFE04	TMTISR4	Timer Interrupt Status Register 4
0xFE08	TMCPRA4	Compare Register A 4
0xFE0C	TMCPRB4	(Reserved)
0xFE10	TMITMR4	Interval Timer Mode Register 4
0xFE20	TMCCDR4	Divide Cycle Register 4
0xFE30	TMPGMR4	(Reserved)
0xFE40	TMWTMR4	(Reserved)
0xFEF0	TMTRR4	Timer Read Register 4
		Timer 5 (TMR5)
0xFF00	TMTCR5	Timer Control Register 5
0xFF04	TMTISR5	Timer Interrupt Status Register 5
0xFF08	TMCPRA5	Compare Register A 5
0xFF0C	TMCPRB5	(Reserved)
0xFF10	TMITMR5	Interval Timer Mode Register 5
0xFF20	TMCCDR5	Divide Cycle Register 5
0xFF30	TMPGMR5	(Reserved)
0xFF40	TMWTMR5	(Reserved)
0xFFF0	TMTRR5	Timer Read Register 5

13.6.1. Timer Control Register n (TMTCRn)

TMTCR0 (0xF000), TMTCR1 (0xF100), TMTCR2 (0xF200), TMTCR3 (0xFD00), TMTCR4 (0xFE00), TMTCR5 (0xFF00)

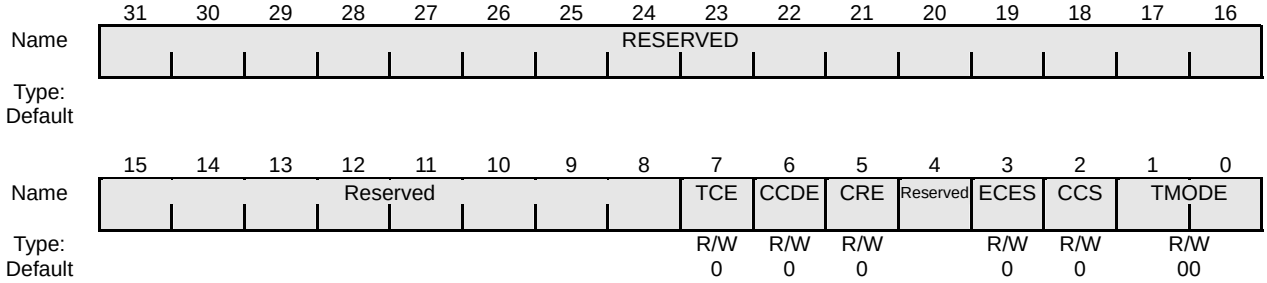


Figure 13-7 Timer Control Register

Table 13-3 Timer Control Register

Bit	Mnemonic	Field Name	Description	R/W
31:8		Reserved		—
7	TCE	Timer Counter Enable	Timer Counter Enable (Default: 0) This field controls whether the counter runs or stops. When in the Watchdog mode, counter operation only stops when the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set. When the Watchdog Timer Disable bit is cleared, the value of this Timer Count Enable bit becomes “0”, but the count continues. 0: Stop counter 1: Counter Operation When TCE bit changes from 1 to 0, CRE value will be evaluated once. And if and only if CRE value is 1, counter will be reset to 0.	R/W
6	CCDE	Counter Clock Divider Enable	Counter Clock Divide Enable (Default: 0) This bit enables the divide operation of the internal clock (IMBUSCLK). The counter stops if this bit is set to “0” when the internal bus clock is in use. 0: Disable 1: Enable	R/W
5	CRE	Counter Reset	Counter Reset Enable (Default: 0) The value of CRE will be evaluated once when TCE bit changes from 1 to 0. Counter value will be reset to zero if and only if CRE value is 1. Such operation that TCE changes from 1 to 0 and same time, CRE changes 0 to 1, by single register write, it will stop and reset the counter.	R/W
4		Reserved		—
3	ECES	External Clock Edge Select	External Clock Edge Select (Default: 0) This bit specifies the counter operation edge when using the counter input signal (TCLK). 0: Falling edge of the counter input signal (TCLK) 1: Rising edge of the counter input signal (TCLK)	R/W
2	CCS	Counter Clock Select	Counter Clock Select (Default: 0) This bit specifies the timer clock. 0: Internal clock (IMBUSCLK) 1: External input clock (TCLK) **Please note that TX4939 does not support external clock**	R/W
1:0	TMODE	Timer Mode	Timer Mode (Default: 00) This bit specifies the timer operation mode. 11: Reserved 10: Watchdog Timer mode (Timer 2), Reserved (Timer 0, 1,3,4,5) 01: Pulse Generator mode (Timer 0, 1), Reserved (Timer 2,3,4,5) 00: Interval Timer mode	R/W

13.6.2. Timer Interrupt Status Register n (TMTISRn)

TMTISR0 (0xF004), TMTISR1 (0xF104), TMTISR2 (0xF204), TMTISR3 (0xFD04), TMTISR4 (0xFE04), TMTISR5 (0xFF04)

Figure 13-8 Timer Interrupt Status Register

Table 13-4 Timer Interrupt Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:4		Reserved		—
3	TWIS	Watchdog Timer Status	Watchdog Timer TMCPRA Match Status (Default: 0) (This bit is Reserved other than TMTISR2 Register.) When in the Watchdog Timer mode, this bit is set when the counter value matches Compare Register A2 (TMCPRA2). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Negate interrupt 1: Invalid	R/W0C
2	TPIBS	Pulse Generator TMCPRB Status	Pulse Generator TMCPRB Match Status (Default: 0) (This bit is Reserved in case of TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register Bn (TMCPRBn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C
1	TPIAS	Pulse Generator TMCPRA Status	Pulse Generator TMCPRA Match Status (Default: 0) (This bit is Reserved in case of TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C

13

0	TIIS	Interval Timer TMCPRA Status	Interval Timer TMCPRA Match Status (Default: 0) When in the Interval Timer mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid	R/W0C

13.6.3. Compare Register An (TMCPRAn)

TMCPRA0 (0xF008), TMCPRA1 (0xF108), TMCPRA2 (0xF2008), TMCPRA3 (0xFD08), TMCPRA4 (0xFE08), TMCPRA5 (0xFF08)

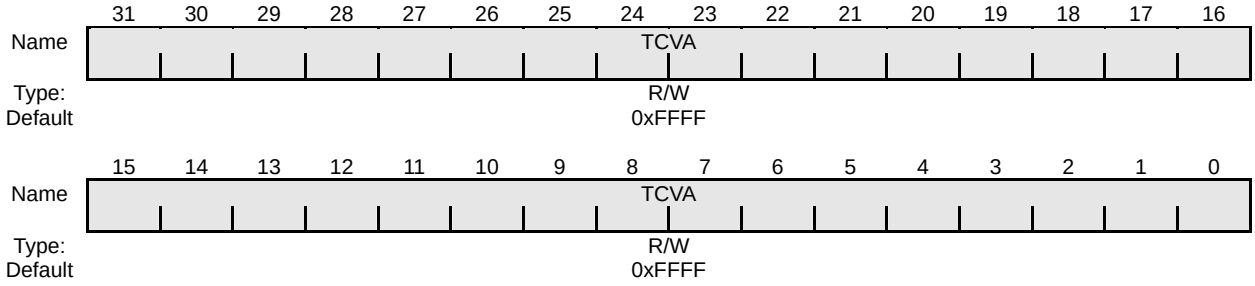


Figure 13-9 Compare Register A

Table 13-5 Compare Register A

Bits	Mnemonic	Field Name	Description	R/W
31:0	TCVA	Timer Compare Register A	Timer Compare Register A (Default: 0xFFFFFFFF) Sets the timer compare value as a 32-bit value. This register can be used in all modes.	R/W

13.6.4. Compare Register Bn (TMCPRBn)

TMCPRB0 (0xF00C), TMCPRB1 (0xF10C),

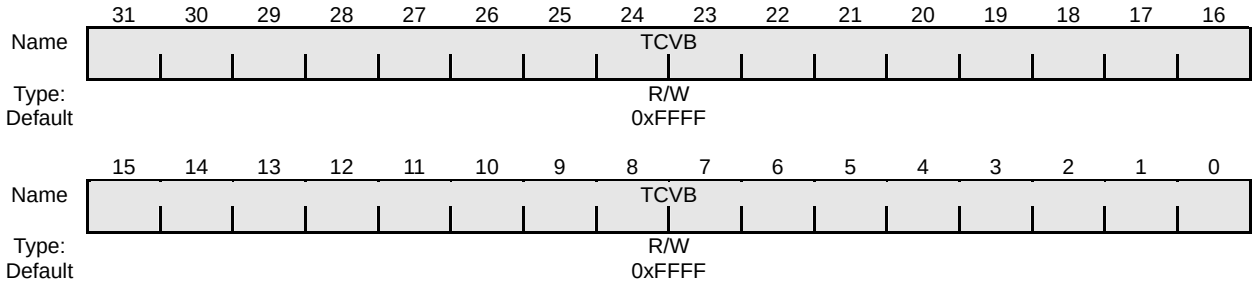


Figure 13-10 Compare Register B

Table 13-6 Compare Register B

Bits	Mnemonic	Field Name	Description	R/W
31:0	TCVB	Timer Compare Value B	Timer Compare Value B (Default: 0xFFFFFFFF) Sets the timer compare value as a 32-bit value. This register can only be used when in the Pulse Generator mode. Please set a value greater than that in Compare Register A.	R/W

13.6.5. Interval Timer Mode Register *n* (TMITMRn)

TMITMR0 (0xF010), TMITMR1 (0xF110), TMITMR2 (0xF210), TMITMR3 (0xFD10), TMITMR4 (0xFE10), TMITMR5 (0xFF10)

Figure 13-11 Interval Timer Mode Register

Table 13-7 Interval Timer Mode Register

Bit	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15	TIIE	Interval Timer Interrupt Enable	Timer Interval Interrupt Enable (Default: 0) Sets Interval Timer TMCPRA Interrupt Enable/Disable. 0: Disable (mask) 1: Enable	R/W
14:1		Reserved		—
0	TZCE	Interval Timer Clear Enable	Interval Timer Zero Clear Enable (Default: 0) This bit specifies whether or not to clear the counter to “0” after the count value matches Compare Register A. Count stops at this value if it is not cleared. 0: Do not clear 1: Clear	R/W

13.6.6. Divide Register *n* (TMCCDRn)

TMCCDR0 (0xF020), TMCCDR1 (0xF120), TMCCDR2 (0xF220), TMCCDR3 (0xFD20), TMCCDR4 (0xFE20), TMCCDR5 (0xFF20)

Figure 13-12 Divide Register

Table 13-8 Divide Register

Bits	Mnemonic	Field Name	Description	R/We
31:3		Reserved		—
2:0	CCD	Counter Clock Divide Value	Counter Clock Divide (Default: 000) These bits specify the divide value when using the internal clock (IMBUSCLK) as the counter input clock source. The binary value *n* is divided by 2^{n+1}. 000: Divide by 2^1 (f/2) 001: Divide by 2^2 (f/4) 010: Divide by 2^3 (f/8) 011: Divide by 2^4 (f/16) 100: Divide by 2^5 (f/32) 101: Divide by 2^6 (f/64) 110: Divide by 2^7 (f/128) 111: Divide by 2^8 (f/256)	R/W

13

13.6.7. Pulse Generator Mode Register *n* (TMPGMRn)

TMPGMR0 (0xF030), TMPGMR1 (0xF130)

Figure 13-13 Pulse Generator Mode Register

Table 13-9 Pulse Generator Mode Register

Bit	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15	TPIBE	TMCPRB Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRB Enable (Default: 0) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRB and the counter value match. 0: Mask 1: Do not mask	R/W
14	TPIAE	TMCPRA Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRA Enable (Default: 0) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRA and the counter value match. 0: Mask 1: Do not mask	R/W
13:1		Reserved		—
0	FFI	Flip Flop Default	Initial TIMER Output Level (Default: 0) This bit specifies the TIMER[n] signal default when in the Pulse Generator mode. 0: Low 1: High	R/W

13.6.8. Watchdog Timer Mode Register *n* (TMWTMRn)

TMWTMR2 (0xF240)

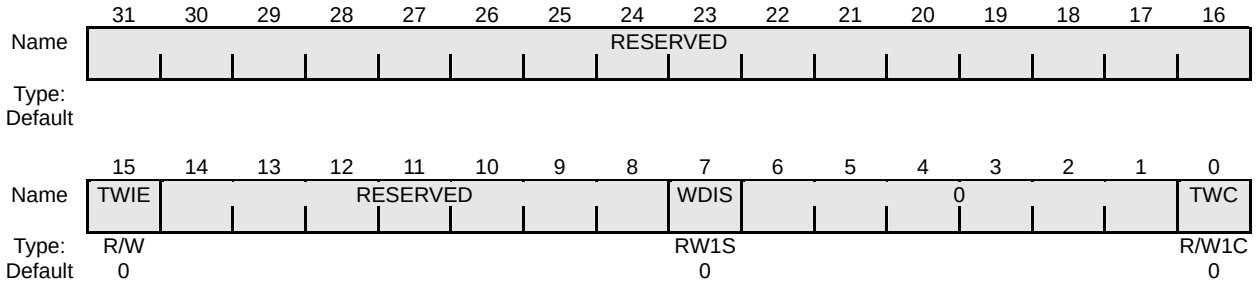


Figure 13-14 Watchdog Timer Mode Register

Table 13-10 Watchdog Timer Mode Register

Bit	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15	TWIE	Watchdog Timer Signaling Enable	Timer Watchdog Enable (Default: 0) This bit sets NMI signaling enable/disable either when in the Watchdog Timer mode or during a reset. This bit cannot be cleared when the Watchdog Timer Disable bit (WDIS) is “0”. 0: Disable (mask) 1: Enable	R/W
14:8		Reserved		—
7	WDIS	Watchdog Timer Disable	Watchdog Timer Disable (Default: 0) Only when this bit is set can the counter be stopped by clearing the Watchdog Timer Signaling Enable bit (TWIE) or by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register. Writing “0” to this bit is not valid. This bit can be cleared in either of the following ways. Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE). Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register.	R/W1S
6:1		Reserved		—
0	TWC	Watchdog Timer Clear	Watchdog Timer Clear (Default: 0) Setting this bit to “1” clears the counter. Writing “0” to this bit is not valid. This bit is always read as “0”.	R/W1C

13.6.9. Timer Read Register *n* (TMTRRn)

TMTRR0 (0xF0F0), TMTRR1 (0xF1F0), TMTRR2 (0xF2F0), TMTRR3 (0xFDF0), TMTRR4 (0xFEF0), TMTRR5 (0xFFF0)

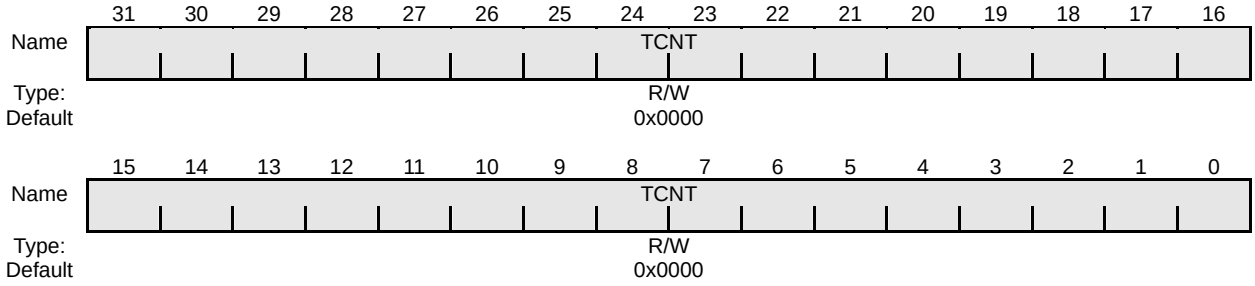


Figure 13-15 Timer Read Register 0

Table 13-11 Timer Read Register n

Bits	Mnemonic	Field Name	Description	R/W
31:0	TCNT	Timer Counter	Timer Counter (Default: 0x00000000) This Register is a 32-bit counter. Read operation on this register returns the counter value. Write operation to this register force the counter value to the write data. The write data should be less than the value of the register TMCPRAn and the timer operates normally except starting counter value. If this write data value is greater or equal to the value of TMCPRAn, the result is undefined.	R/W

13

Chapter 14. DMA Controller

14.1. Features

The TX4939 contains two set of four-channel DMA Controller (DMAC0, DMAC1) that executes DMA (Direct Memory Access) with memory and I/O devices.

The DMA Controller has the following characteristics.
DMAC has on-chip 8-channel DMA controller. Three channels are for transferring to memory and I/O devices. The other channels are for internal resources (4 channels are dedicated for ACLC/I2S).

- *<DMAC0, DMAC1>*
  - *Has four on-chip DMA channels*
  - *Supports Memory-Memory Copy modes that do not have address boundary limitations. Burst transfer of up to eight double words is possible for each Read or Write operation.*
  - *Supports Memory Fill mode that writes double-word data to the specified memory region*
  - *Supports Chained DMA Transfer*
  - *On-chip signed 24-bit address count up registers for both the source address and destination address*
  - *On-chip 26-bit Byte Count Register for each channel*
  - *One of two methods can be selected for determining access priority among multiple channels: Round Robin or Fixed Priority*
  - *Big Endian or Little Endian mode can be set separately for each channel*

- *<DMAC0>*
  - *Supports external I/O devices with 8-, 16-, and 32-bit Data Bus widths and transfer between memory devices.*
  - *Supports DMA on-chip Serial I/O Controllers, and NAND Flash Memory Controller*

- *<DMAC1>*
  - *Supports DMA on-chip AC-Link Controllers*
  - *Supports DMA on-chip I2S Controllers*

14.2. Block Diagram

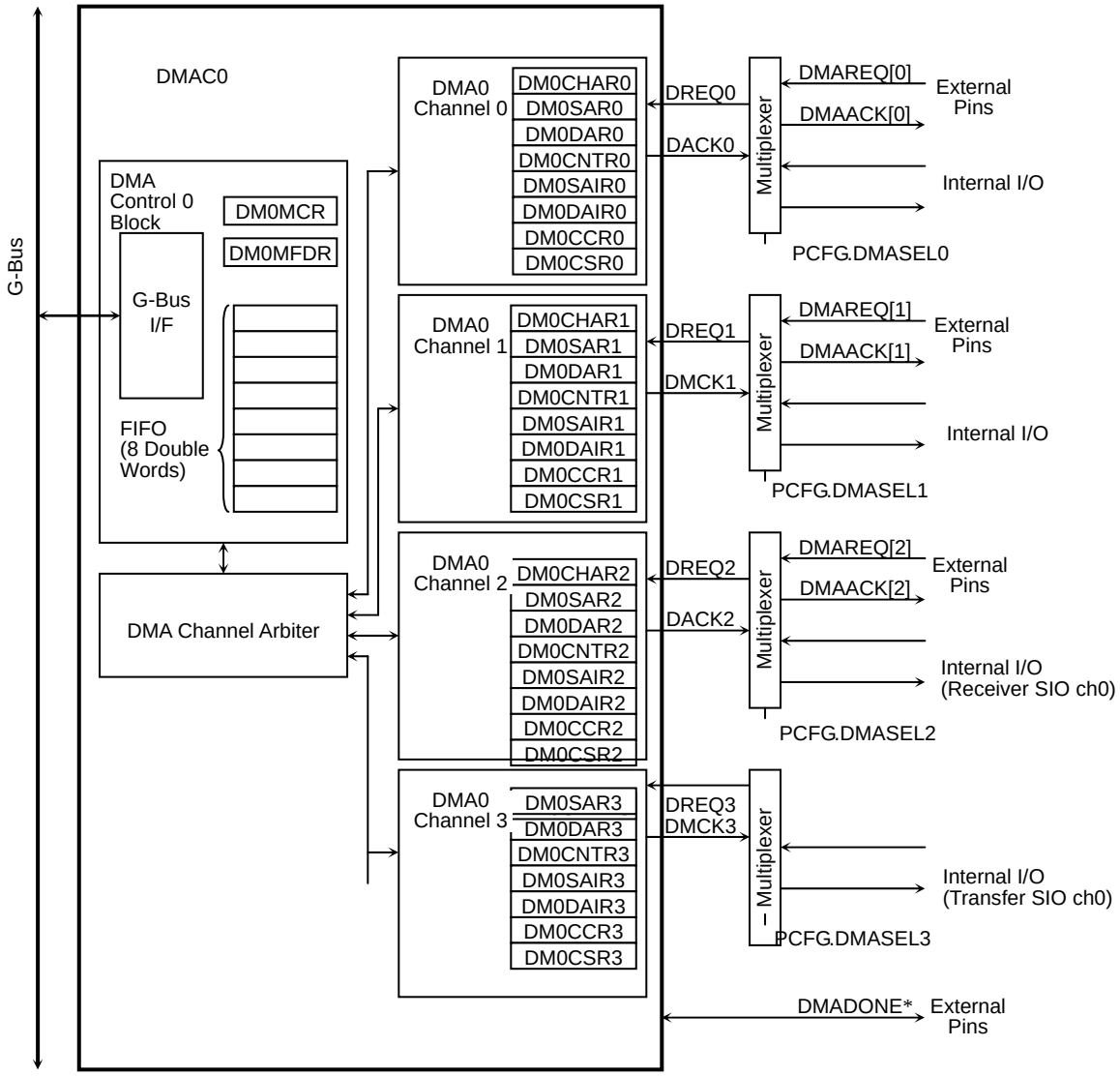


Figure 14-1 DMA0 Controller Block Diagram

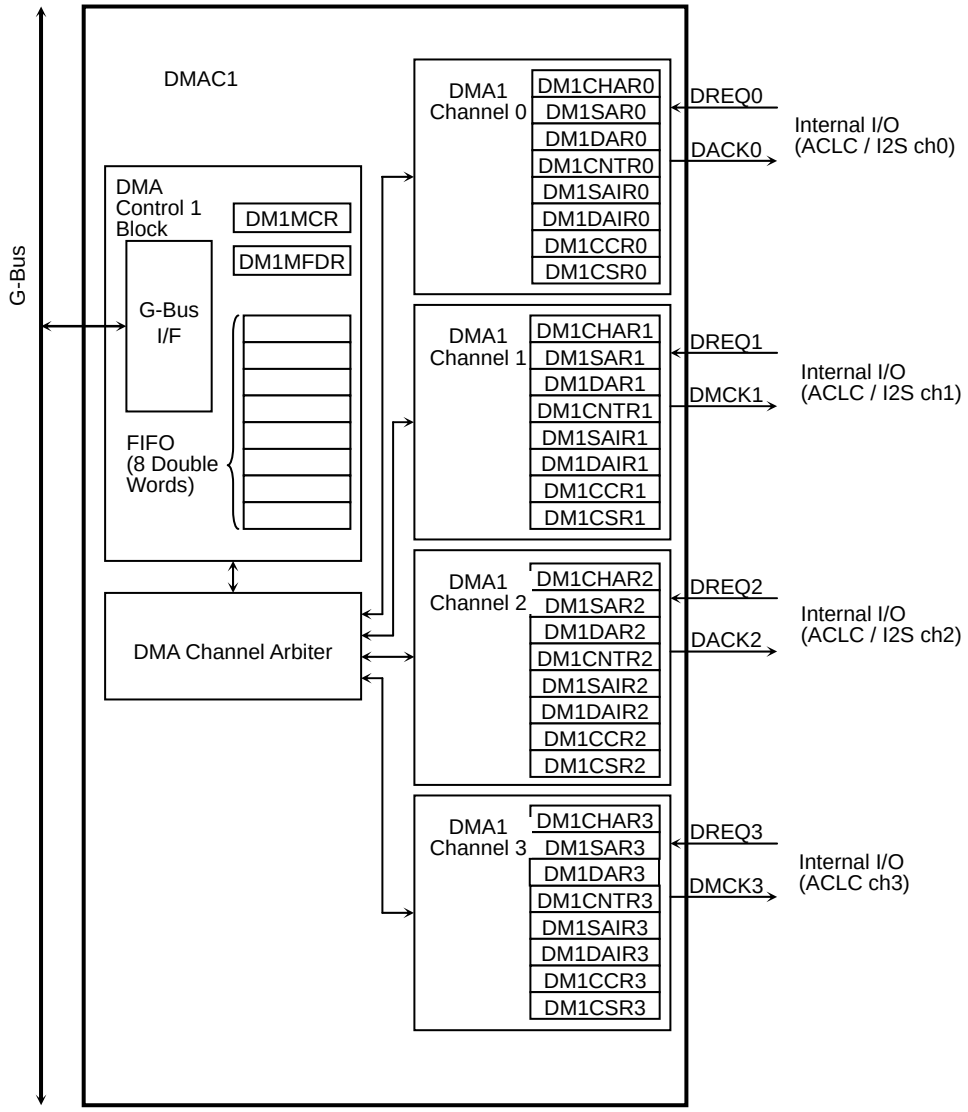


Figure 14-2 DMA1 Controller Block Diagram

14.3. Detailed Explanation

14.3.1. Transfer Mode

The DMA Controller (DMAC0, DMAC1) supports five transfer mode types (refer to Table 14-1 DMA Controller Transfer Modes below). The setting of the External Request bit (DMCCRn.EXTRQ) of the DMA Channel Control Register selects whether transfer with an I/O device is a DMA transfer.

I/O DMA Transfer Mode (DMCCRn.EXTRQ = “1”)
Perform DMA transfer with either an external device connected to the External Bus Controller or an on-chip I/O device (ACLC/I2S or SIO or NDFMC)

Memory Transfer Mode (DMCCRn.EXTRQ = “0”)
Either copies data between memory devices or fills data in memory.

Table 14-1 DMA Controller Transfer Modes

DMA Controller	Transfer Mode	DMCCRn EXTREQ	PCFG DMASEL	DMCCRn SNGAD	DMSAR	DMDAR	Ref.
DMAC0	External I/O (Dual Address)	1	0	0	√	√	14.3.3 14.3.8
	Internal I/O (SIO)	1	1	0	√	√	14.3.4 14.3.8
	Internal- I/O (NDFMC)	1	1	1	√		14.3.4
	Memory-Memory Copy	0	—	0	√	√	14.3.4 14.3.8
DMAC1	Internal I/O (ACLC/I2S)	1	—	0	√	√	14.3.4 14.3.8
	Memory-Memory Copy	0	—	0	√	√	14.3.4 14.3.8

14.3.2. On-chip Registers

The DMA Controller has two shared registers that are shared by four channels. Section 0 explains each register in detail.

- Shared Registers
  DMMCR: DMA Master Control Register
  DMMFDR: DMA Memory Fill Data Register

- DMA Channel Register
  DMCHARn: DMA Chained Address Register
  DMSARn: DMA Source Address Register
  DMDARn: DMA Destination Address Register
  DMCNTRn: DMA Count Register
  DMSAIRn: DMA Source Address Increment Register
  DMDAIRn: DMA Destination Address Increment Register
  DMCCRn: DMA Channel Control Register
  DMCSRn: DMA Channel Status Register

14.3.3. External I/O DMA Transfer Mode

The External I/O DMA Transfer Mode performs DMA transfer with external I/O devices that are connected to the External Bus Controller.

14.3.3.1. External Interface

External I/O devices signal DMA requests to the DMA Controller by asserting the DMA Transfer Request Signal (DMAREQ[n]). On the other hand, the DMA Controller accesses external I/O devices by asserting the DMA Transfer Acknowledge Signal (DMAACK[n]).

The DMA Transfer Request signal (DMAREQ[n]) can use the Request Polarity bit (REQPOL) of the DMA Channel Control Register (DMCCRn) to select the signal polarity for each channel, and can use the Edge Request bit (EGREQ) to select either edge detection or level detection for each channel. The DMA Transfer Acknowledge signal (DMAACK[n]) can also use the Acknowledge Polarity bit (ACKPOL) to select the polarity.

Please assert/deassert the DMAREQ[n] signal as follows below.

When level detection is set (DMCCRn.EGREQ = 0)

The DMAREQ[n] signal must be continuously asserted until one SYSCLK cycle after the DMAACK[n] signal is asserted. Also, the DMAREQ[n] signal must be deasserted before the CE*/CS* signal is deasserted. If this signal is asserted too soon, DMA transfer will not be performed. If this signal is asserted or deasserted too late, unexpected DMA transfer may result.

During Dual Address transfer, we recommend detecting assertion of the CE* signal for the external I/O device that is currently asserting DMAACK[n], then deasserting DMAREQ[n].

When edge detection is set (DMCCRn.EGREQ = 1)

Please set up assertion of the DMAREQ[n] signal so the DMAREQ[n] signal is asserted after the DMAACK[n] signal corresponding to a previously asserted DMAREQ[n] signal is deasserted. The DMAREQ[n] signal will not be detected even if it is asserted before DMAACK[n] is deasserted.

*** is a timing diagram that shows the timing of external DMA access. In this timing diagram, both the DMAREQ[n] signal and the DMAACK[n] signal are set to Low active (DMCCRn.REQPL = 0, DMCCRn.ACKPOL = 0).
The DMAACK[n] and DMADONE[n] signals, which are DMA control signals, are synchronized to SDCLK. When these signals are used by an external I/O device that is synchronous to SYSCLK, it is necessary to take clock skew into account. The DMAACK[n] signal is asserted either at the SYSCLK cycle, the same as with assertion of the CE*/CS* signal, or before that. In addition, it is deasserted after the last ACK*/READY signal is deasserted.
When the DMADONE* signal (refer to 14.3.3.3) is used as an output signal, it is asserted for at least one SYSCLK cycle while the DMAACK[n] signal is asserted either during the same SYSCLK cycle that the CE*/CS* signal is deasserted or during a subsequent SYSCLK cycle. When the DMADONE* signal is used as an input signal, it must be asserted for one SYSCLK cycle while the DMAACK[n] signal is being asserted.

14.3.3.2. Dual Address Transfer

If the Single Address bit (DMCCRn.SNGAD) has been cleared, access to external I/O devices and to external memory is each performed continuously. Each access is the same as normal access except when the DMAACK[n] signal is asserted.

Please refer to “14.3.8 Dual Address Transfer” for information regarding setting the register.

14.3.3.3. DMADONE* Signal

The DMADONE* signal operates as either the DMA stop request input signal or the DMA done signalling output signal, or may operate as both of these signals depending on the setting of the DONE Control Field (DNCTRL) of the DMA Channel Control Register (DMCCRn).

The DMADONE* signal is shared by four channels. The DMADONE* channel is valid for a channel when the DMAACK[n] signal for that channel is asserted.

If the DMADONE* channel is set to be used as an output signal (DMCCRn.DNCTRL = 10/11), it will operate as follows depending on the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn).

When the Chain End bit (CHDN) is set, the DMADONE* signal is only asserted when the DMAACK[n] signal for the last DMA transfer in the Link List Command Chain is asserted.

When the Chain End bit (CHDN) is cleared, the DMADONE* signal is asserted when the DMAACK[n] signal for the last data transfer in a DMA transfer specified by the current DMA Channel Register is asserted. Namely, if the Link List Command chain is used, there is one assertion at the end of each data transfer specified by each Descriptor.

If the DMADONE* signal is set to be used as an input signal (DMCCRn.DNCTRL = 01/11), DMA transfer can be set to end normally when the external device asserts the DMADONE* signal when the DMAACK[n] signal of channel *n* is asserted. DMADONE* is asserted during DMAACK[n] is not asserted, then unexpected operation occurs. When DMA transfer is terminated by the DMADONE* assertion of the external device, the External DONE Assert bit (DMCSRn.EXTDN) of the DMA Channel Status Register is set regardless of the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn). Operation is as follows depending on the setting of the Chain End bit (CHDN).

When the Chain End bit (CHDN) is set, all DMA transfer for that chain is terminated. At this time, the Normal Chain End bit (NCHNC) and the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register are both set and the Transfer Active bit (DMCCRn.XFACT) of the DMA Channel Control Register is cleared.

When the Chain End bit (CHDN) is cleared, only DMA transfer specified by the current DMA Channel Register ends normally, and only the Normal Transfer End bit (NTRNFC) is set. When the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn) is set, chain transfer is executed and DMA transfer continues. When the Chain Enable bit (CHNEN) is cleared, the Transfer Active bit (DMCCRn.XFACT) is cleared and the Normal Chain End bit (NCHNC) is set.

Three clock cycles are required from external assertion of the DMADONE* signal to disabling of new DMA access. Operation will not stop even if the bus operation in progress is a Single transfer or a Burst transfer. For example, if the DMADONE* signal is asserted during Read operation of Dual Address transfer, the corresponding Write bus operation will also be executed.

If the DMADONE* pin is set to become both input and output for channel *n* (DMCCRn.DNCTRL = “11”), the DMADONE* signal becomes an open drain signal when the channel becomes active. When used by this mode, the DMADONE* signal must be pulled up by an external source. When in this mode, the External DONE Assert bit (DMCSRn.EXTDN) is not only set when asserted by an external device, but is also set when asserted by the TX4939.

14.3.4. Internal I/O DMA Transfer Mode

Performs DMA with the on-chip Serial I/O Controller, the AC-link Controller or the I2S Controller. Set the DMA Channel Control Register (DMCCRn) as follows.

DMCCRn.EXTRQ = 1: I/O DMA Transfer mode
DMCCRn.SNGAD = 0: Dual Address Transfer

Refer to “14.3.8 Dual Address Transfer” and “DMA transfer (Serial I/O Controller)” or “DMA operation (AC-link Controller)” or “DMA Interface (I2S Controller)” for more information.
Refer to the Boot Configuration section for selection AC-link or I2S

Performs DMA with the on chip NAND Flash Memory Controller (NDFMC) Set the DMA Channel 3 Control Register (DMCCRn) as follows.
DMCCRn.EXTRQ = 1: I/O DMA Transfer mode
DMCCRn.SNGAD = 1: Single Address Transfer
Refer to DMA Operation section in NDFMC for more information

14.3.5. Memory-Memory Copy Mode

It is possible to copy memory from any particular address to any other particular address when in the Memory-Memory Copy mode.

Set the DMA Channel Control Register (DMCCRn) as follows.
DMCCRn.EXTRQ = 0: Memory Transfer mode
DMCCRn.SNGAD = 0: Dual Address mode

Furthermore, when in the Memory-Memory Copy mode it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to “14.3.8 Dual Address Transfer” for information regarding the setting of other registers.

14.3.6. Memory Fill Transfer Mode

When in the Memory Fill Transfer mode, double word data set in the DMA Memory Fill Data Register (DMMFDR) is written to the data region specified by the DMA Source Address Register (DMSARn). This data can be used for initializing the memory, etc.

Set the DMA Channel Control Register (DMCCRn) as follows.

- *DMCCRn.EXTRQ = 0: Memory transfer mode*
- *DMCCRn.SNGAD = 1: Single Address Transfer*
- *DMCCRn.MEMIO = 0: Transfer from I/O to memory*

In addition, when in the Memory Fill Transfer mode, it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to “14.3.7 Single Address Transfer” for information regarding the setting of other registers.

By using this function together with the memory Write function that writes to multiple SDRAM Controller memory channels simultaneously (refer to Section 9.3.4), it is possible to initialize memory even more efficiently.

14.3.7. Single Address Transfer

Note: The Single Address Transfer is only supported for internal NAND Flash Controller (NDFMC)

This section explains register settings during Single Address transfer (DMCCRn.SNGAD = 1). This applies to the following DMA Transfer modes.

- *External I/O (Single Address) Transfer*
- *Memory Fill Transfer*

14.3.7.1. Channel Register Settings During Single Address Transfer

Table 14-2 Channel Register Setting Restrictions During Single Address Transfer shows restrictions of the Channel Register settings during Single Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set and DMA transfer is not performed.

For Burst transfer, +8, 0, or –8 can be set to the DMA Source Address Increment Register (DMSAIRn). Setting 0 is only possible during transfer from memory to external I/O. A Configuration Error will result if the value “0” is set during transfer from external I/O to memory or during Memory Fill transfer.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 2 bytes or larger, then set the DMA Source Address Register (DMSARn) with 1 to 3 low-order bits complemented.

- *If the transfer size is 2 bytes, set the DMSARn with the low-order 1 bit complemented.*
- *If the transfer size is 4 bytes, set the DMSARn with the low-order 2 bits complemented.*
- *If the transfer size is 8 bytes or larger, set the DMSARn with the low-order 3 bits complemented.*

Example: When the transfer address is 0x0_0001_0000, the DMA Source Address Register (DMSARn) is as follows below.

- *DMSAIRn setting is “0” or greater: 0x0_0001_0000*
- *DMSAIRn setting is a negative value: 0x0_0001_0007*

During Single Address transfer, the DMA Destination Address Register (DMDARn) and DMA Destination Address Increment Register (DMDAIRn) settings are ignored.

Table 14-2 Channel Register Setting Restrictions During Single Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[2:0]		DMSAIRn[2:0]	DMCNTRn[2:0]
	DMSAIRn is “0” or greater	DMSAIRn setting is a negative value		
1 Byte	***	***	***	***
2 Bytes	**0	**0	**0	**0
4 Bytes	*00	*00	*00	*00
8 Bytes	000	111	000	000
4 Double Words	000	111	8/0/-8	000
8 Double Words				
16 Double Words				
32 Double Words				

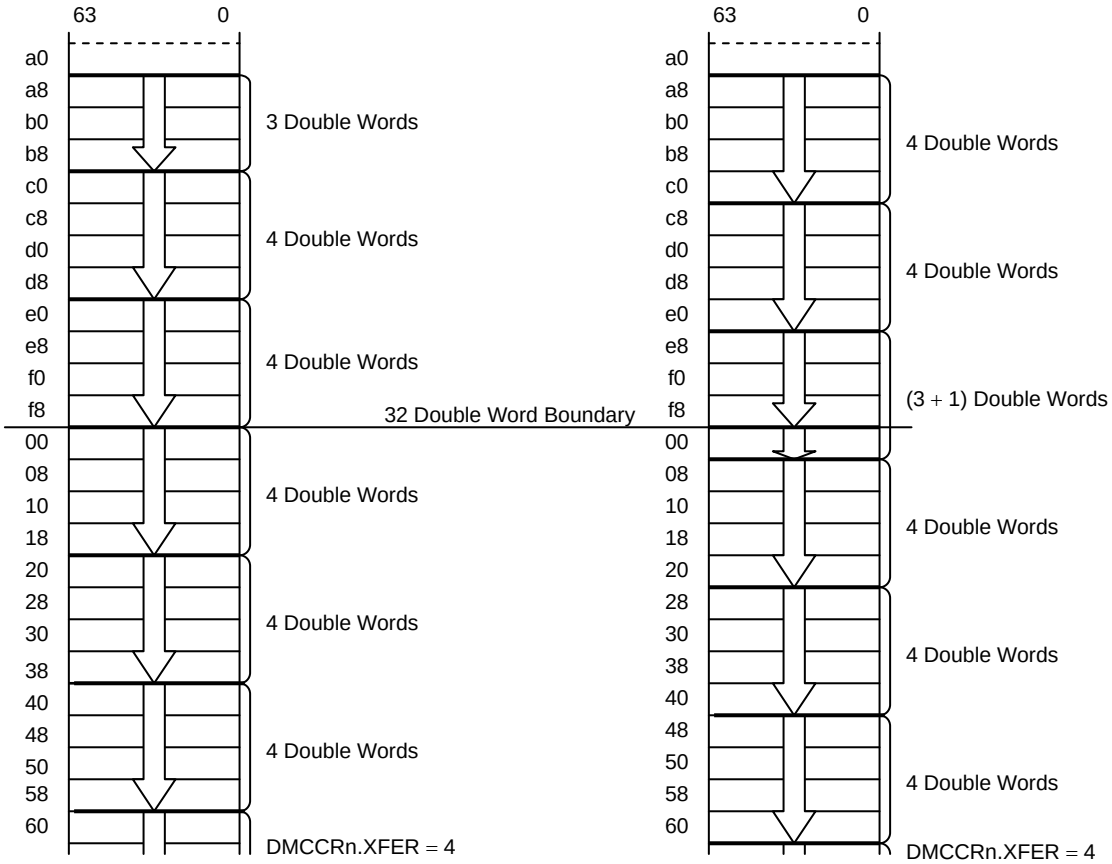
14.3.7.2. Burst Transfer During Single Address Transfer

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32-double word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, the DMA Controller executes multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register.

Figure 14-3 shows the Single Address Burst transfer status when the lower 8 bits of the Transfer Start address are 0xA8 and the transfer setting size (DMCCRn.XFSZ) is set to 4 double words.

Panel (a) of this figure shows the situation when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is “0”. In this case, first a three-double word transfer is performed up to the address aligned to the transfer setting size. Then, four-double word transfer specified by the transfer setting size is repeated. This setting is normally used.

On the other hand, panel (b) shows when the Transfer Size Mode bit (DMCCRn.USEXFSWZ) is “1”. In this case, transfer is repeated according to the transfer setting size. Three-double word transfer and one-double word transfer is only performed consecutively without releasing bus ownership when transfer spans across a 32-double word boundary.



(a) DMCCRn.USEXFSZ = “0” ***(b) DMCCRn.USEXFSZ = “1”***

Figure 14-3 Non-aligned Single Address Burst Transfer

14.3.8. Dual Address Transfer

This section explains the register settings for Dual Address transfer (DMCCRn.SNGAD = 0). This applies to the following DMA transfer modes.

- *External I/O (Dual Address) transfer*
- *Internal I/O DMA transfer*
- *Memory-Memory Copy transfer*

14.3.8.1. Channel Register Settings During Dual Address Transfer

Table 14-3 shows restrictions of the Channel Register settings during Dual Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set, and DMA transfer is not performed.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 8 bytes or larger, then a value will be set in the DMA Source Address Register (DMSARn) that reflects as follows.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer size is 2 bytes or larger, set the DMA Source Address Register (DMSARn) as follows:

- *If the transfer size is 2 bytes, set the DMSARn with the low-order 1 bit complemented.*
- *If the transfer size is 4 bytes, set the DMSARn with the low-order 2 bits complemented.*
- *If the transfer size is 8 bytes or larger, set the DMSARn with the low-order 3 bits complemented.*

Likewise, if the setting of the DMA Destination Address Increment Register (DMDAIRn) is negative and the transfer size is 2 bytes or larger, set the DMA Destination Address Register (DMDARn) as follows:

- *If the transfer size is 2 bytes, set the DMDARn with the low-order 1 bit complemented.*
- *If the transfer size is 4 bytes, set the DMDARn with the low-order 2 bits complemented.*
- *If the transfer size is 8 bytes or larger, set the DMDARn with the low-order 3 bits complemented.*

Example: When the transfer address is 0x0_0001_0000, the DMA Source Address Register (DMSARn) is as follows below.

- *DMSAIRn setting is "0" or greater: 0x0_0001_0000*
- *DMSAIRn setting is a negative value: 0x0_0001_0007*

Table 14-3 Channel Register Setting Restrictions During Dual Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[2:0]		DMDARn[2:0]		DMSAIRn	DMDAIRn	DMCNTRn	DMCCRn REVBYTE
	DMSAIRn setting is 0 or greater	DMSAIRn setting is a negative value	DMDAIRn setting is 0 or greater	DMDAIRn setting is a negative value				
1 Byte	***	***	***	***	***	***	***	0
2 Bytes	**0	**0	**0	**1	**0	**0	**0	0
4 Bytes	*00	*00	*00	*11	*00	*00	*00	0
8 Bytes, 4 / 8 Double Wods (DMMCR.FIFUM[n]= 0)	000	111	000	111	000	000	000	0/1
4 / 8 Double Words (DMMCR.FIFUM[n]= 1)	000	111	000	111	8/0/-8	8/-8 ‡	000	0/1
	***	—	***	—	8	8	***	0
	—	***	—	***	-8	-8		0
16 Double Words	Cannot be set (Configuration Error)							
32 Double Words	Cannot be set (Configuration Error)							

‡: 8, 0, or -8 can be specified when the Destination Burst Inhibit bit (DMCCRn.DBINH) is set.

14.3.8.2. Burst Transfer During Dual Address Transfer

The DMA Controller has a 64-bit 8-stage FIFO on-chip that is connected to the internal bus (G-Bus) for Burst transfer during Dual Address transfer. Since this FIFO employs a shifter, it is possible to perform transfer of any address or data size. Burst transfer is only performed when 4 Double Words or 8 Double Words is set by the Transfer Setting Size field (DMCCRn.XFSZ) and the FIFO Use Enable bit (DMMCRn.FIFUM[n]) of the DMA Master Control Register is set.

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32-double word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, it is necessary to divide the transfer into multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register and whether or not the address offset relative to the Transfer Setting size (DMCCRn.XFSZ) is equivalent to the source address and destination address combined.

Figure 14-4 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to “1”, the lower 8 bits of the Transfer Start address for the transfer source are set to 0xA8, the lower 8 bits of the Transfer Start address for the transfer destination are set to 0x38, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 Double Words.

Transfer repeats according to the transfer setting size, regardless of the different address offsets. However, transfers that span across 32-double word boundaries are divided. Since data remains in the on-chip FIFO when in this mode, it becomes possible to share the on-chip FIFO among multiple DMA channels.

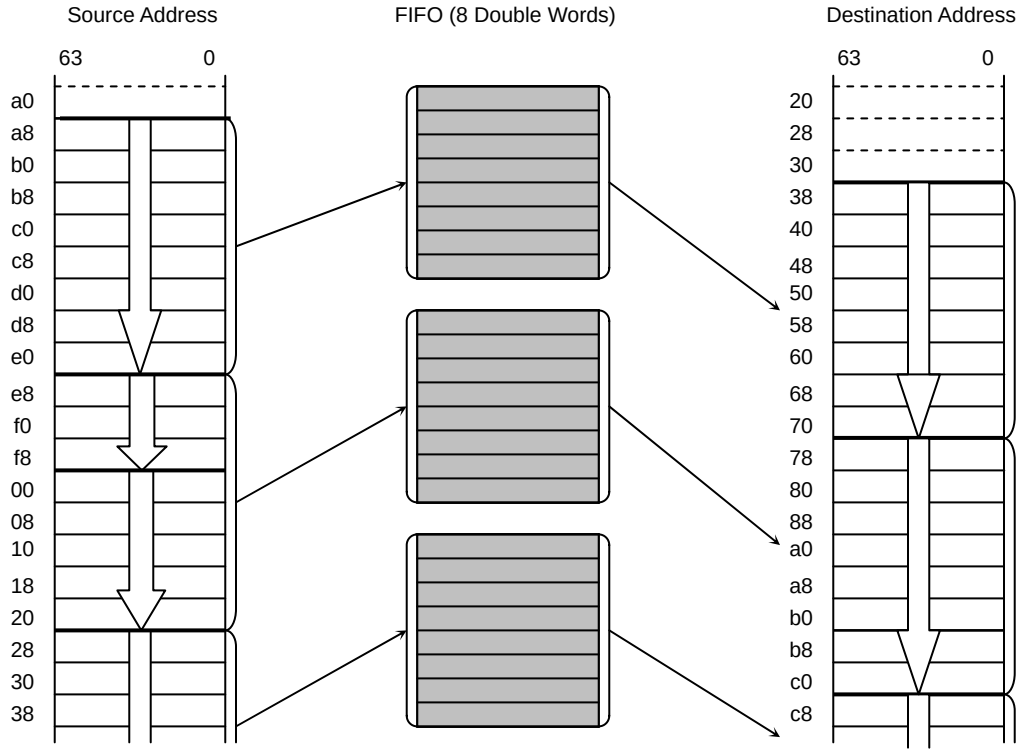


Figure 14-4 Dual Address Burst Transfer (DMCCRn.USEXFSZ = 1)

Figure 14-5 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to “0”, the lower 8 bits of the Transfer Start address for the transfer source are set to 0xA8, the lower 8 bits of the Transfer Start address for the transfer destination are set to (a) 0x28/(b) 0x30, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 double words.

Panel (a) of this figure shows when the address offset is equivalent. In this case, first transfer of three double words is performed up to the address that is aligned with the transfer setting size. Then, transfer of eight double words that is specified by the transfer setting size is repeated.

On the other hand, panel (b) show when the address offset is not equivalent. In this case, first only data up to the address that is aligned with the transfer setting size is read to the on-chip FIFO. Then, data is written up to the address that is aligned with the transfer setting size as long as data remains in the on-chip FIFO. Efficiency decreases since the transfer size is divided. Also, since data may remain in the on-chip FIFO, Burst transfer of a Dual Address that uses the on-chip FIFO simultaneously with another channel cannot be performed.

Using the Burst Inhibit bit makes it possible to mix Burst transfer with 8-Double-Word Single transfer. This in turn makes it possible to perform Burst access only for memory access during DMA transfer with external I/O devices that cannot perform Burst transfer.

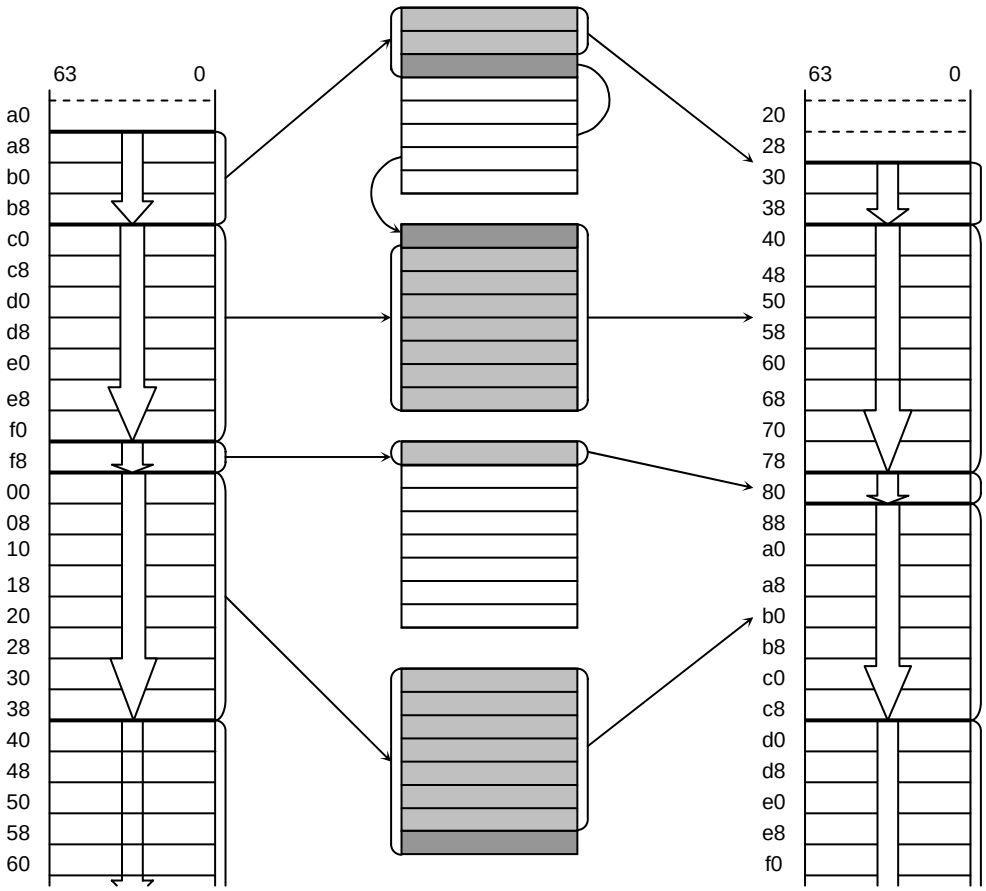
When the Source Burst Inhibit bit (DMCCRn.SBINH) is set, data read from the Source Address to the on-chip FIFO is divided into multiple 8-byte Single Read transfers, then transfer is executed.

When the Destination Burst Inhibit bit (DMCCRn.DBINH) is set, data written from the FIFO to the Destination Address is divided into multiple 8-byte Single Write transfers, then transfer is executed.

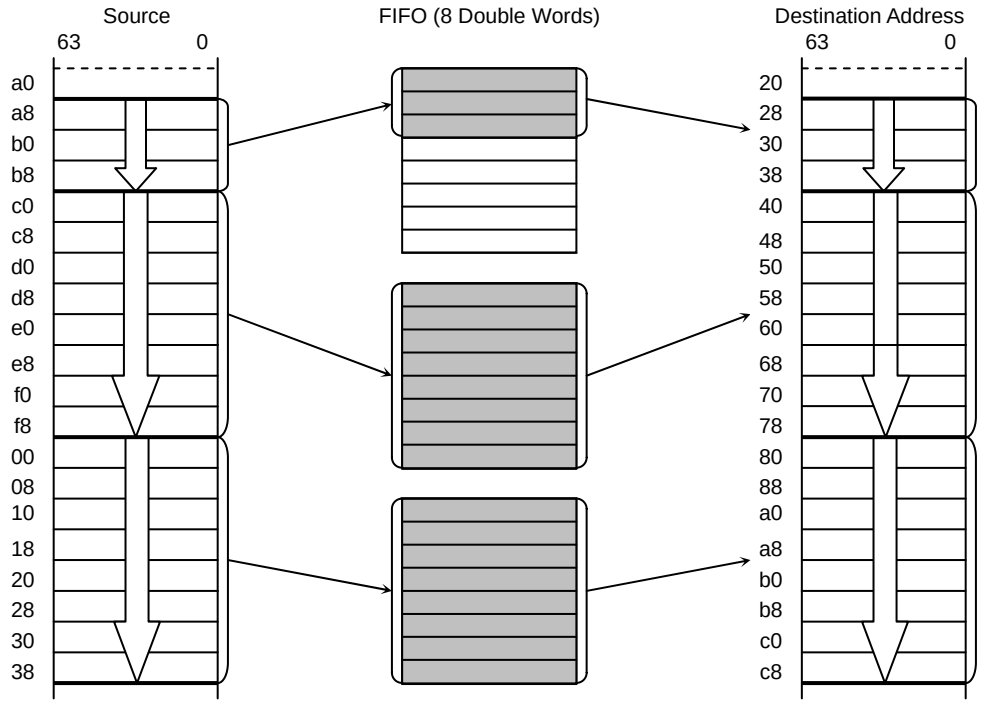
14.3.8.3. Double Word Byte Swapping

When the Reverse Byte bit (REVBYTE) of the DMA Channel Configuration Register (DMCCRn) is set, read double word data is written after byte swapping is performed. For example, if the read data is “0x01234567_89ABCDEF”, then the data “0xEFCDAB89_67452301” is written.

The Reverse Byte bit can only be set when the REVBYTE column of Table 14-3 is set so “0/1” is indicated.



(a) Address offset is equivalent



(b) Address offset differs

Figure 14-5 (b) Dual Address Burst Transfer (DMCCRn.USEXFSZ = 0)

14.3.9. DMA Transfer

The sequence of DMA transfer that uses only the DMA Channel Register is as follows below.

(1) Select DMA request signal

When performing external I/O or internal I/O DMA, set the DMA Request Select field (PCFG.DMASEL) of the Pin Configuration Register.

(2) Set the Master Enable bit

Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

(3) Set the Address Register and Count Register
Set the five following register values.

- *DMA Source Address Register (DMSARn)*
- *DMA Destination Address Register (DMDARn)*
- *DMA Count Register (DMCNTRn)*
- *DMA Source Address Increment Register (DMSAIRn)*
- *DMA Destination Address Increment Register (DMDAIRn)*

(4) Set Chain Address Register

Set “0” to the DMA Chain Address Register (DMCHARn).

(5) Clear the DMA Channel Status Register (DMCSRn)

Clear when status from the previous DMA transfer remains.

(6) Set the DMA Channel Control Register (DMCCRn)

(7) Initiate DMA transfer

DMA transfer is started by setting the Transfer Active bit (XFACT) of the DMA Channel Control Register.

(8) Signal completion

When DMA data transfer ends normally, the Normal Transfer Complete bit (NTRNFC) of the DMA Channel Status Register (DMCSRn) is set. An interrupt is signalled if the Transfer Complete Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set.

If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status Register and the transfer is interrupted. If the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set, then the interrupt is signaled.

14.3.10. Chain DMA Transfer

Table 14-4 shows the data structure in memory that the DMA Command Descriptor has. When the Simple Chain bit (SMPCHN) of the DMA Channel Control Register (DMCCRn) is set, only the initial four double words are used. DMSAIRn, DMDAIR, DMCCRn, and DMCSRn use the settings from when DMA started. In addition, all eight double words are used when the Simple Chain bit (SMPCHN) is cleared.

Saving the start memory address of another DMA Command Descriptor in the Offset 0 Chain Address field makes it possible to construct a chain list of DMA Command Descriptors (Figure 14-6). Set “0” in the Chain Address field of the DMA Command Descriptor at the end of the chain list.

When DMA transfer that is specified by one DMA Command Descriptor ends, the DMA Controller automatically reads the next DMA Command Descriptor indicated by the Chain Address Register (Chain transfer), then continues DMA transfer. Continuous DMA transfer that uses multiple Descriptors connected into such a chain-like structure is called Chain DMA transfer.

Since the DMA Channel Status Register is also overwritten during Chain transfer when the DMA Simple Chain bit (SMPCHN) is cleared, be sure not to unnecessarily clear necessary bits.

Placing DMA Command Descriptors at addresses that do not span across 32-double-word boundaries in memory is efficient since they are read by one G-Bus Burst Read operation.

Table 14-4 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	DMA Chain Address Register (DMCHARn)
0x08	Source Address	DMA Source Address Register (DMSARn)
0x10	Destination Address	DMA Destination Address Register (DMDARn)
0x18	Count	DMA Count Register (DMCNTRn)
0x20	Source Address Increment	DMA Source Address Increment Register (DMSAIRn)
0x28	Destination Address Increment	DMA Destination Address Increment Register (DMDAIRn)
0x30	Channel Control	DMA Channel Control Register (DMCCRn)
0x38	Channel Status	DMA Channel Status Register (DMCSRn)

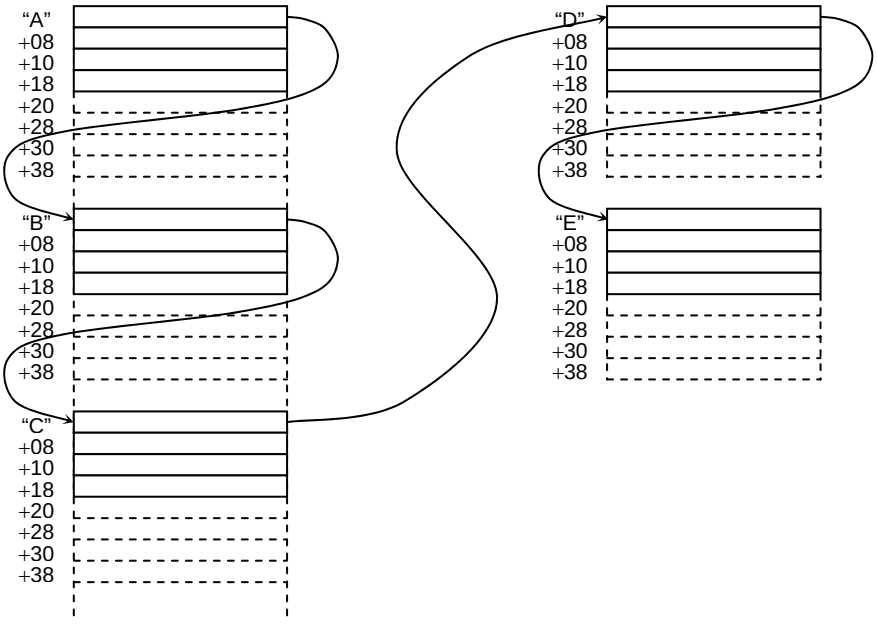


Figure 14-6 DMA Command Descriptor Chain

The sequence of Chain DMA transfer is as follows below.

(1) Select DMA request signal

When performing external I/O or internal I/O DMA, set the DMA Request Select field (PCFG.DMASEL) of the Pin Configuration Register.

(2) Set the Master Enable bit

Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

(3) Structure of the DMA command Descriptor chain

Construct the DMA Command Descriptor Chain in memory.

(4) Set the Count Register

Set “0” to the DMA Count Register (DMCNTRn) .
Sets the DMA Source Address Increment Register (DMSAIRn) and DMA destination Address Increment Register (MMDAIRn).

(5) Clear the DMA Channel Status Register (DMCSRn)

Clear the status of the previous DMA transfer.

(6) Set the DMA Channel Control Register (DMCCRn).

(7) Initiate DMA transfer

Setting the address of the DMA Command Descriptor at the beginning of the chain list in the DMA Chain Address Register (DMCHARn) automatically initiates DMA transfer. First, the value stored in each field of the DMA Command descriptor at the beginning of the Chain List is read to each corresponding DMA Channel register (Chain transfer), then DMA transfer is performed according to the read value.

When a value other than “0” is stored in the DMA Chain Address Register (DMCHARn), data of the size stored in the DMA Count Register (DMCNTRn) is completely transferred, then the DMA Command Descriptor value of the memory address specified by the DMA Chain Address Register is read.

In addition, if the Chain Address field value read the Descriptor 0, the DMA Chain Address Register value is not updated. All previous values (Data Command Descriptor Addresses with the value “0” in the Chain Address field when the values were read) are held.

0 Value judgement is performed when the lower 32 bits of the DMA Chain Address Register are rewritten. If the value is not “0” at this time, DMA transfer is automatically initiated. Therefore, please write to the upper 32 bits first when writing to the DMA Chain Address Register using 32-bit Store instructions.

(8) Signal completion

Set the Normal Chain End bit (NCHNC) of the DMA Channel Status Register (DMCSRn) when DMA data transfer of all Descriptor Chains is complete. An interrupt is signalled if the Chain End Interrupt Enable bit (INTENC) of the DMA Channel Control Register (DMCCRn) is set at this time.

In addition, the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register (DMCSRn) is set each time DMA data transfer specified by each DMA Command Descriptor ends normally. An interrupt is signalled if the Transfer End Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set at this time.

If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status register and transfer is interrupted. An interrupt is signalled if the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set.

14.3.11. Dynamic Chain Operation

It is possible to add DMA Command Descriptor chains to the DMA Command Descriptor chain while Chain DMA transfer is in progress. This is performed according to the following procedure.

(1) Construct the DMA Command Descriptor chain

Construct the DMA Command Descriptor chain to be added to memory.

(2) Add a DMA Command Descriptor chain

Substitute the address of the Command Descriptor at the beginning of the Descriptor Chain to be added into the Chain Address field of the Descriptor at the end of the DMA Command Descriptor chain that is currently performing DMA transfer.

(3) Check the Chain Enable bit

Read the value of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn). If that value is “0”, then write the Chain Address field value of the DMA Command Descriptor that is indicated by the address stored in the DMA Chain Address Register (DMCHARn).

14.3.12. Interrupts

An interrupt number (10 – 13) of the Interrupt Controller is mapped to each channel. In addition, there are completion interrupts for when transfer ends normally and error interrupts for when transfer ends abnormally for each channel. When an interrupt occurs, then the bit that corresponds to either the Normal Interrupt Status field (DIS[3:0]) or the Error Interrupt Status field (EIS[3:0]) of the DMA Master Control Register (DMMCR) is set.

Figure 14-7 shows the relationship between the Status bit and Interrupt Enable bit for each interrupt cause. Refer to the explanation for each Status bit for more information regarding each information cause.

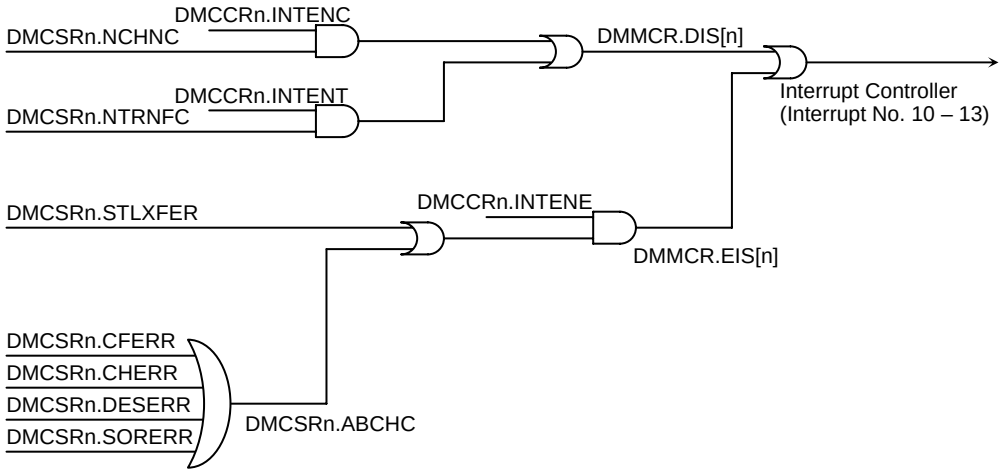


Figure 14-7 DMA Controller Interrupt Signal

14.3.13. Transfer Stall Detection Function

If the period from when a certain channel last performs internal bus access to when the next internal bus access is performed exceeds the Transfer Stall Detection Interval field (STLTIME) of the DMA Channel Control Register (DMCCRn), the Transfer Stall Detection bit (STLXFER) of the DMA Channel Status Register (DMCSRn) is set. An error interrupt is signalled if the Error Interrupt Enable bit (DMCCRn.INTENE) is set.

In contrast to other error interrupts, DMA transfer is not stopped. Normal DMA transfer is executed if bus ownership can be obtained. Furthermore, clearing the Transfer Stall Detection field (STLXFER) resumes transfer stall detection as well.

Setting the Transfer Stall Detection Interval field (STLTIME) to “000” disables the Transfer Stall Detection function.

14.3.14. Arbitration Among DMA Channels

The DMA Controller has an on-chip DMA Channel Arbiter that arbitrates bus ownership among four DMA channels that use the internal bus (G-Bus). There are two methods for determining priority: the round robin method and the fixed priority method. (See Figure 14-8.) The Round Robin Priority bit (RRPT) of the DMA Master Control Register (DMMCR) selects the priority method.

Fixed priority (DMMCR.RRPT = 0)
As shown below, Channel 0 has the highest priority and Channel 3 has the lowest priority.
CH0 > CH1 > CH2 > CH3

Round Robin method (DMMCR.RRPT = 1)
The last channel to perform DMA transfer has the lowest priority.

- *After CH0 DMA transfer execution: CH1 > CH2 > CH3 > CH0*
- *After CH1 DMA transfer execution: CH2 > CH3 > CH0 > CH1*
- *After CH2 DMA transfer execution: CH3 > CH0 > CH1 > CH2*
- *After CH3 DMA transfer execution: CH0 > CH1 > CH2 > CH3*

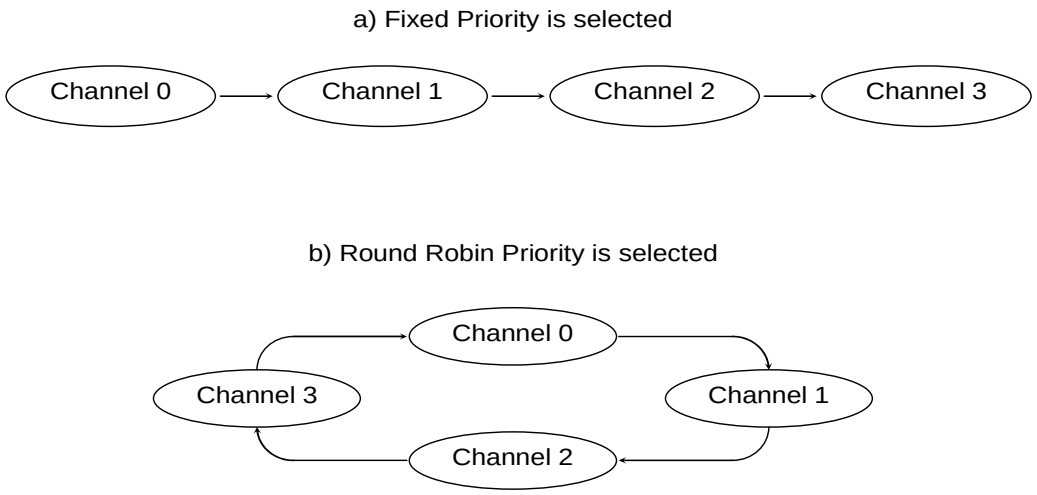


Figure 14-8 DMA Channel Arbitration

14.3.15. Restrictions in Access to PCI Bus

The PCI Controller detects a bus error if the DMA Controller performs one of the following accesses to the PCI Bus.

Burst transfer exceeding 8 double words (PCICSTATUS.TLB)
Address Increment value –8 Burst transfer (PCICSTATUS.NIB)
Address Increment Value 0 Burst transfer (PCICSTATUS.ZIB)
Dual Address Burst transfer when the setting for DMSARn, DMDARn, or DMCNTRn is not a double word boundary (PCICSTATUS.IAA)

In addition, Single Address transfers between an external I/O device and the PCI Bus are not supported. Data transfer is not performed, but no error is detected.

14.4. DMA Controller Registers

Table 14-5 DMA Controller 0 Registers

Offset Address	Bit Width	Mnemonic	Register Name
0xB000	64	DM0CHAR0	DMA Chain Address Register 0
0xB008	64	DM0SAR0	DMA Source Address Register 0
0xB010	64	DM0DAR0	DMA Destination Address Register 0
0xB018	64	DM0CNTR0	DMA Count Register 0
0xB020	64	DM0SAIR0	DMA Source Address Increment Register 0
0xB028	64	DM0DAIR0	DMA Destination Address Increment Register 0
0xB030	64	DM0CCR0	DMA Channel Control Register 0
0xB038	64	DM0CSR0	DMA Channel Status Register 0
0xB040	64	DM0CHAR1	DMA Chain Address Register 1
0xB048	64	DM0SAR1	DMA Source Address Register 1
0xB050	64	DM0DAR1	DMA Destination Address Register 1
0xB058	64	DM0CNTR1	DMA Count Register 1
0xB060	64	DM0SAIR1	DMA Source Address Increment Register 1
0xB068	64	DM0DAIR1	DMA Destination Address Increment Register 1
0xB070	64	DM0CCR1	DMA Channel Control Register 1
0xB078	64	DM0CSR1	DMA Channel Status Register 1
0xB080	64	DM0CHAR2	DMA Chain Address Register 2
0xB088	64	DM0SAR2	DMA Source Address Register 2
0xB090	64	DM0DAR2	DMA Destination Address Register 2
0xB098	64	DM0CNTR2	DMA Count Register 2
0xB0A0	64	DM0SAIR2	DMA Source Address Increment Register 2
0xB0A8	64	DM0DAIR2	DMA Destination Address Increment Register 2
0xB0B0	64	DM0CCR2	DMA Channel Control Register 2
0xB0B8	64	DM0CSR2	DMA Channel Status Register 2
0xB0C0	64	DM0CHAR3	DMA Chain Address Register 3
0xB0C8	64	DM0SAR3	DMA Source Address Register 3
0xB0D0	64	DM0DAR3	DMA Destination Address Register 3
0xB0D8	64	DM0CNTR3	DMA Count Register 3
0xB0E0	64	DM0SAIR3	DMA Source Address Increment Register 3
0xB0E8	64	DM0DAIR3	DMA Destination Address Increment Register 3
0xB0F0	64	DM0CCR3	DMA Channel Control Register 3
0xB0F8	64	DM0CSR3	DMA Channel Status Register 3
0xB148	64	DM0MFDR	DMA Memory Fill Data Register
0xB150	64	DM0MCR	DMA Master Control Register

Table 14-6 DMA Controller 1 Registers

Offset Address	Bit Width	Mnemonic	Register Name
0xB800	64	DM1CHAR0	DMA Chain Address Register 0
0xB808	64	DM1SAR0	DMA Source Address Register 0
0xB810	64	DM1DAR0	DMA Destination Address Register 0
0xB818	64	DM1CNTR0	DMA Count Register 0
0xB820	64	DM1SAIR0	DMA Source Address Increment Register 0
0xB828	64	DM1DAIR0	DMA Destination Address Increment Register 0
0xB830	64	DM1CCR0	DMA Channel Control Register 0
0xB838	64	DM1CSR0	DMA Channel Status Register 0
0xB840	64	DM1CHAR1	DMA Chain Address Register 1
0xB848	64	DM1SAR1	DMA Source Address Register 1
0xB850	64	DM1DAR1	DMA Destination Address Register 1
0xB858	64	DM1CNTR1	DMA Count Register 1
0xB860	64	DM1SAIR1	DMA Source Address Increment Register 1
0xB868	64	DM1DAIR1	DMA Destination Address Increment Register 1
0xB870	64	DM1CCR1	DMA Channel Control Register 1
0xB878	64	DM1CSR1	DMA Channel Status Register 1
0xB880	64	DM1CHAR2	DMA Chain Address Register 2
0xB888	64	DM1SAR2	DMA Source Address Register 2
0xB890	64	DM1DAR2	DMA Destination Address Register 2
0xB898	64	DM1CNTR2	DMA Count Register 2
0xB8A0	64	DM1SAIR2	DMA Source Address Increment Register 2
0xB8A8	64	DM1DAIR2	DMA Destination Address Increment Register 2
0xB8B0	64	DM1CCR2	DMA Channel Control Register 2
0xB8B8	64	DM1CSR2	DMA Channel Status Register 2
0xB8C0	64	DM1CHAR3	DMA Chain Address Register 3
0xB8C8	64	DM1SAR3	DMA Source Address Register 3
0xB8D0	64	DM1DAR3	DMA Destination Address Register 3
0xB8D8	64	DM1CNTR3	DMA Count Register 3
0xB8E0	64	DM1SAIR3	DMA Source Address Increment Register 3
0xB8E8	64	DM1DAIR3	DMA Destination Address Increment Register 3
0xB8F0	64	DM1CCR3	DMA Channel Control Register 3
0xB8F8	64	DM1CSR3	DMA Channel Status Register 3
0xB948	64	DM1MFDR	DMA Memory Fill Data Register
0xB950	64	DM1MCR	DMA Master Control Register

14.4.1. DMA Master Control Register (DM0MCR, DM1MCR)

Offset address: DMAC0 0xB150, DMAC1 0xB950 This register controls the entire DMA Controller.

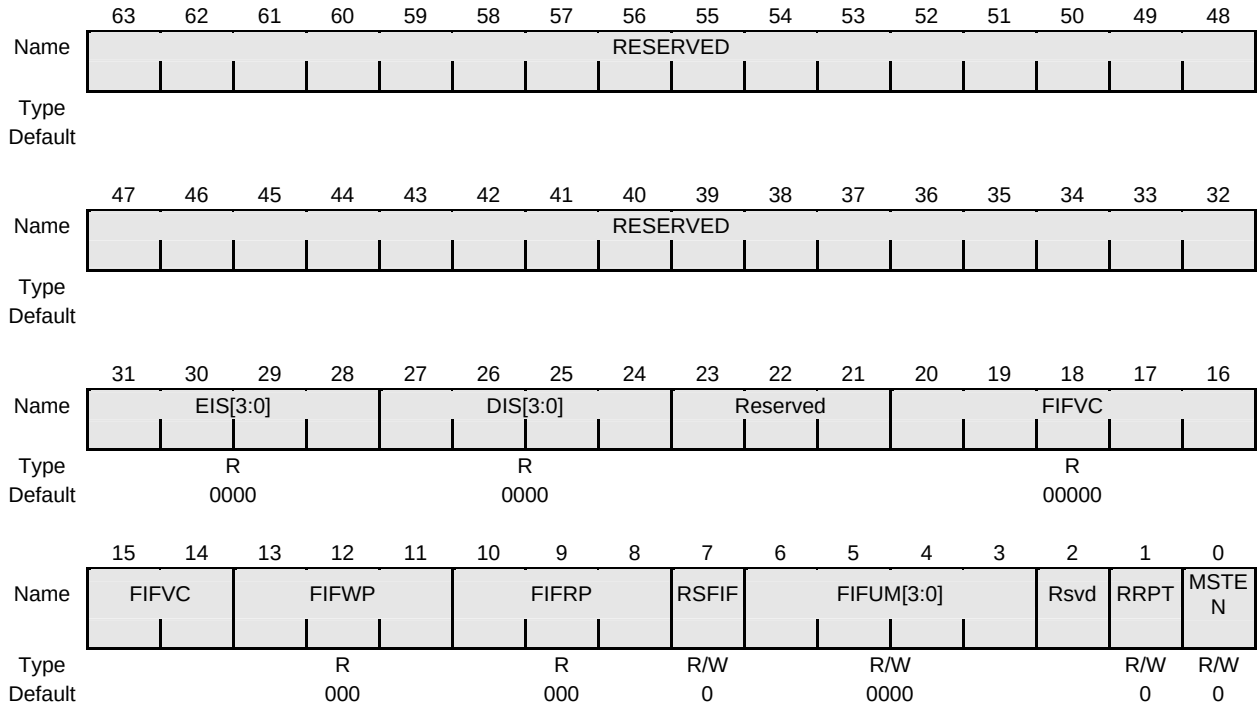


Figure 14-9 DMA Master Control Register

Table 14-7 DMA Master Control Register

Bit	Mnemonic	Field Name	Description	R/W
63:32		Reserved		—
31:28	EIS[3:0]	Error Interrupt Status	Error Interrupt Status [3:0] (Default: 0x0) These four bits indicate the error interrupt status of each channel. EIS[n] corresponds to channel *n*. 1: There is an error interrupt in the corresponding channel. 0: There is no error interrupt in the corresponding channel.	R
27:24	DIS[3:0]	Normal Completion Interrupt Status	Done Interrupt Status [3:0] (Default: 0x0) These four bits indicate the transfer completion (transfer complete or chain ended) interrupt status of each channel. DIS[n] corresponds to channel *n*. 1: There is a transfer completion interrupt in the corresponding channel. 0: There is no transfer completion interrupt in the corresponding channel.	R
23:21		Reserved		—
20:14	FIFVC	FIFO Valid Entry Count	FIFO Valid Entry Count (Default: 0000000) These read only bits indicate the byte count of data that were written to FIFO but not read out from the FIFO.	R
13:11	FIFWP	FIFO Write Pointer	FIFO Write Pointer (Default: 000) These read only bits indicate the next write position in FIFO. This is a diagnostic function.	R
10:8	FIFRP	FIFO Read Pointer	FIFO Read Pointer (Default: 000) These read only bits indicate the next read position in FIFO. This is a diagnostic function.	R
7	RSFIF	Reset FIFO	Reset FIFO (Default: 0) This bit is used for resetting FIFO. When this bit is set to “1”, the FIFO read pointer, FIFO write pointer and FIFO valid entry count are initialized to “0”. If an error occurs during DMA transfer, use this bit when data remains in the FIFO (when the FIFO Valid entry Count Field is not “0”) to initialize the FIFO.	R/W

Table 14-7 DMA Master Control Register

Bit	Mnemonic	Field Name	Description	R/W
6:3	FIFUM[3:0]	FIFO Use Enable [3:0]	FIFO Use Enable [3:0] (Default: 0x0) Each channel specifies whether to use 8-double word FIFO in Dual Address transfer. FIFUM[n] corresponds to channel *n*. Refer to "14.3.8.2 Burst Transfer During Dual Address Transfer" for more information.	R/W
2		Reserved		—
1	RRPT	Round Robin Priority	Round Robin Priority (Default: 0) Specifies the method for determining priority among channels. 1: Round Robin method. Priority of the last channel used is the lowest, and the next previous channel has the next lowest priority. Round robin is in the order Channel 0 > Channel 1 > Channel > Channel 3. 0: Fixed Priority. Priority is fixed in the order Channel 0 > Channel 1 > Channel 2 > Channel 3.	R/W
0	MSTEN	Master Enable	Master Enable (Default: 0) This bit enables the DMA Controller. 1: Enable 0: Disable Note: If the entire DMA Controller is disabled, then all internal logic including the Bus Interface Logic and State Machine are reset.	R/W

14.4.2. DMA Channel Control Register (DM0CCRn, DM1CCRn)

Offset address: DMAC0 0xB030 (ch. 0) / 0xB070 (ch. 1) / 0xB0B0 (ch. 2) / 0xB0F0 (ch. 3)
DMAC1 0xB830 (ch. 0) / 0xB870 (ch.1) / 0xB8B0 (ch. 2) / 0xB8F0 (ch. 3)

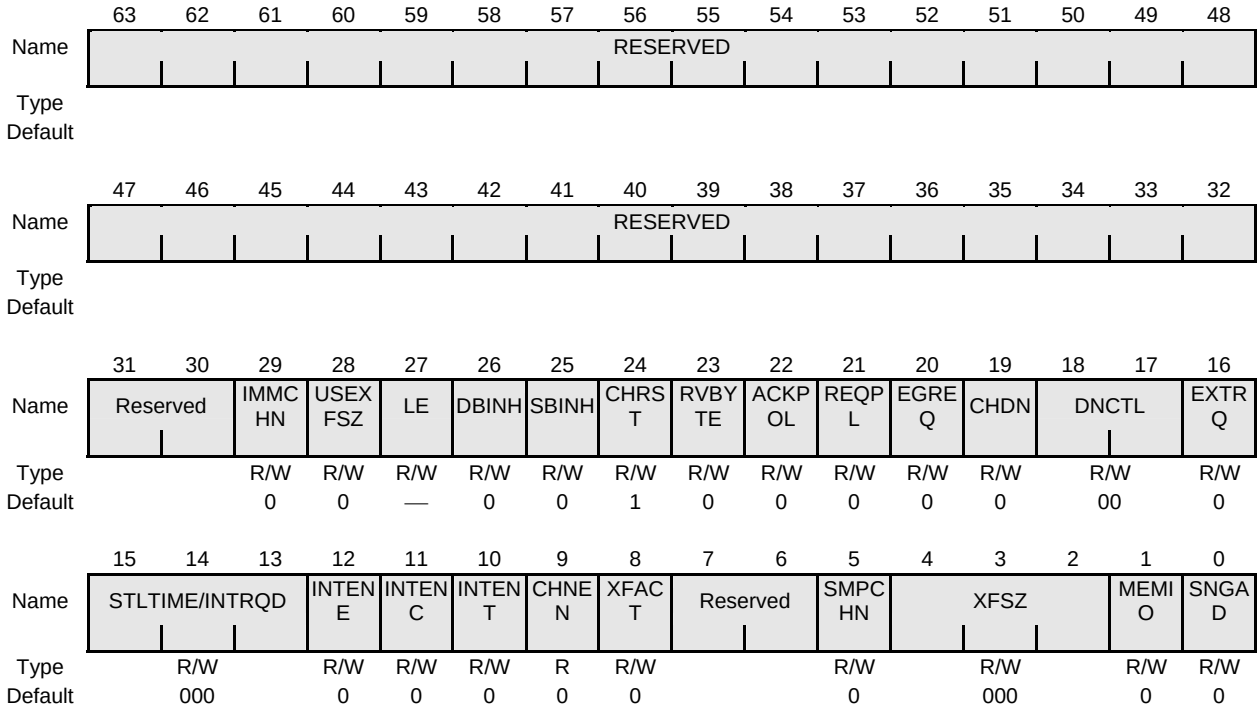


Figure 14-10 DMA Channel Control Register

Table 14-8 DMA Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
63:32		Reserved		—
29	IMMCHN	Immediate Chain	Immediate Chain (Default: 0) Always set this bit to “1”. 1: means that DMAC will automatically enter chain mode from the last transfer so the bus is held and there is no penalty to reacquire the bus. 0: means that the DMAC gives up the bus between the last transfer on a channel and the chain operation for that channel.	R/W
28	USEXFSZ	Transfer Set Size Mode	Use Transfer Set Size (Default: 0) Selects the DMA channel operation mode during Burst DMA transfer. Refer to “14.3.7.2” and “14.3.8.2 Burst Transfer During Dual Address Transfer” for more information. 1: The DMA Controller always transfers the amount of data set in DMCCRn.XFSZ for each bus operation. Since alignment to the boundary of the DMCCRn.XFSZ in the address is not forced when in this mode, transfers that exceed 32-double-word boundaries are divided into two operations. 0: The DMA Controller calculates the transfer size so the address set in DMSARn and DMDARn (only during Dual Address transfer) can be aligned to the boundary of the size set in DMCCRn.XFSZ, then transfers data according to that size. Note: In Dual Address Transfer mode, programming this bit to 0 is valid only when both the contents of the DMSARn and the DMDARn are on doubleword boundaries *and* the contents of the DMCNTRn is a multiple of eight bytes.	R/W
27	LE	Little Endian	Little Endian (Default: value that is the opposite of the G-Bus Endian (CCFG.ENDIAN) This bit sets the Endian of the channel. Please use the default value as is. 1: Channel operates in the Little Endian mode 0: Channel operates in the Big Endian mode	R/W

Table 14-8 DMA Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
26	DBINH	Destination Burst Inhibit	Destination Burst Inhibit (Default: 0) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Write cycle to the address set from FIFO to DMDARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “14.3.8.2 Burst Transfer During Dual Address Transfer” for more information. The settings of this bit have no effect during Single Address transfers. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.	R/W
25	SBINH	Source Burst Inhibit	Source Burst Inhibit (Default: 0) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Read cycle to the FIFO from the address set to DMSARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “14.3.8.2 Burst Transfer During Dual Address Transfer” for more information. The settings of this bit have no effect during Single Address transfers. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.	R/W
24	CHRST	Channel Reset	Channel Reset (Default: 1) This bit is used fo initializing channels. The DMCCRn.XFACT, DMCCRn.CHNEN, and DMCSRn bits are all cleared. In addition, all channel logic and interrupts from channels are cleared and bus ownership requests to the DMA Channel Arbiter are also reset. The software must clear this bit before operating a channel. 1: Reset channel 0: Enable channel	R/W
23	REVBYTE	Reverse Byte	Reverse Bytes (Default: 0) This bit specifies whether to reverse the byte order during a Dual Address transfer when the Transfer Setting Size field (DMCCRn.XFSZ) setting is 8 bytes or more. Refer to “14.3.8.3 Double Word Byte Swapping” for more information. 1: Reverses the byte order. 0: Does not reverse the byte order.	R/W
22	ACKPOL	Acknowledge Polarity	Acknowledge Polarity (Default: 0) Specifies the polarity of the DMAACK[n] signal. 1: Asserts when the DMAACK[n] signal is High 0: Asserts when the DMAACK[n] signal is Low	R/W
21	REQPL	Request Polarity	Request Polarity (Default: 0) Specifies the polarity of the DMAREQ[n] signal. 1: Asserts when the DMAREQ[n] signal is High. 0: Asserts when the DMAREQ[n] signal is Low.	R/W
20	EGREQ	Edge Request	Edge Request (Default: 0) Specifies the method for detecting DMA requests by the DMAREQ[n] signal. 1: DMAREQ[n] signal is Edge Detect. 0: DMAREQ[n] signal is Level Detect.	R/W
19	CHDN	Chain Complete	Chain Done (Default: 0) Selects control by the DMADONE* signal. See “14.3.3.3” for more information. 1: Assertion of the DMADONE* signal controls the overall Chain DMA transfer. 0: Assertion of the DMADONE* signal controls DMA transfer according to the DMA Channel Register setting at that time.	R/W
18:17	DNCTL	DONE Control	Done Control (Default: 00) Specifies the input/output mode of the DMADONE* signal. Refer to “14.3.3.3 DMADONE* Signal” for more information. 00: DMADONE* signal becomes the input signal, but input is ignored. 01: DMADONE* signal becomes the input signal. 10: DMADONE* signal becomes the output signal. 11: DMADONE* signal becomes the open drain input/output signal.	R/W

Table 14-8 DMA Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
16	EXTRQ	External Request	External Request (Default: 0) Sets the Request Transfer mode. 1: I/O DMA transfer mode This bit is used by the External I/O DMA Transfer mode and the Internal I/O DMA Transfer mode. A channel requests internal bus ownership when the I/O device asserts the DMA request signal. 0: Memory Transfer mode This bit is used by the Memory-Memory Copy Transfer mode and the Memory Fill Transfer mode. A channel requests internal bus ownership when the value of DMCSRn.WAITC becomes “0”.	R/W
15:13	STLTIME / INTRQD	Transfer Stall Detection Interval/Internal Request Delay	When in the I/O DMA Transfer mode (DMCCRn.EXTRQ is “1”) Stalled Transfer Detect Time (Default: 000) Sets the detection interval for a lack of bus ownership. If this channel *n* releases bus ownership then the interval it does not have ownership exceeds the clock count set by this field, then DMCSRn.STLXFER is set to “1”. Refer to “0 Transfer Stall Detection Function” for more information. 000: Does not detect stalled transfers. 001: Sets 960 (15 × 64) clocks as the detection interval 010: Sets 4032 (63 × 64) clocks as the detection interval 011: Sets 16320 (255 × 64) clocks as the detection interval 100: Sets 65472 (1023 × 64) clocks as the detection interval 101: Sets 262080 (4095 × 64) clocks as the detection interval 110: Sets 1048512 (16383 × 64) clocks as the detection interval 111: Sets 4194240 (65535 × 64) clocks as the detection interval When in the Memory Transfer mode (DMCCRn.EXTRQ is “0”) Internal Request Delay (Default: 000) Sets the delay time from when bus ownership is released to the next bus ownership request. Bus ownership is released, the set delay time elapses, then a bus ownership request is generated from the channel. 000: Always requests bus ownership when this channel is active. (Bus ownership is released after bus operation ends) 001: Set 16 clocks as the delay time 010: Set 32 clocks as the delay time 011: Set 64 clocks as the delay time 100: Set 128 clocks as the delay time 101: Set 256 clocks as the delay time 110: Set 512 clocks as the delay time 111: Set 1024 clocks as the delay time	R/W
12	INTENE	Error Interrupt Enable	Interrupt Enable on Error (Default: 0) Enables interrupts when the Error End bit (DMCSRn.ABCHC) or the Transfer Stall Detection bit (DMCSRn.STLXFER) is set. 1: Generates interrupts. 0: Does not generate interrupts.	R/W
11	INTENC	Chain End Interrupt Enable	Interrupt Enable on Chain Done (Default: 0) This bit enables interrupts when the Chain End bit (DMCSRn.NCHNC) is set. 1: Generate interrupts. 0: Do not generate interrupts.	R/W
10	INTENT	Transfer End Interrupt Enable	Interrupt Enable on Transfer Done (Default: 0) This bit enables interrupts when the Transfer End bit (DMCSRn.NTRNFC) is set. 1: Generate interrupts. 0: Do not generate interrupts.	R/W

Table 14-8 DMA Channel Control Register

Bit	Mnemonic	Field Name	Description	R/W
9	CHNEN	Chain Enable	Chain Enable (Default: 0) This bit indicates whether Chain operation is being performed. Read Only. This bit is cleared when either the Master Enable bit (DMMCR.MSTEN) is cleared or the Channel Reset bit (DMCCRn.CHRST) is set. This bit is set if a value other than “0” is set when the CPU writes to the DMA Chain Address Register (DMCHARn) or when a Chain transfer writes DMA Command Descriptor. This bit is then cleared when “0” is set to the DMA Chain Address Register (DMCHARn). 1: If transfer completes due to the current DMA Channel Register setting, a DMA Command Descriptor is loaded in the DMA Channel Register from the specified DMA Chain Address Register (DMCHARn) address, then DMA transfer continues. 0: Further transfer does not start even if transfer completes due to the current DMA Channel Register setting.	R
8	XFACT	Transfer Active	Transfer Active (Default: 0) DMA transfer is performed according to the DMA Channel Register setting when this bit is set. This bit is automatically set when a value other than “0” is set in the DMA Chain Address Register (DMCHARn). DMA transfer is then initiated. This bit is automatically cleared either when DMA transfer ends normally it is stopped due to an error. 1: Perform DMA transfer. 0: Do not perform DMA transfer.	R/W
7:6		Reserved		—
5	SMPCHN	Simple Chain	Simple Chain (Default: 0) This bit selects the DMA Channel Register that loads data from DMA Command Descriptors during Chain DMA transfer. 1: Data is only loaded to the four following DMA Channel Registers: the Chain Address Register (DMCHARn), the Source Address Register (DMSARn), the Destination Address Register (DMDARn), and the Count Register (DMCNTRn). 0: Data is loaded to all eight DMA Channel Registers.	R/W
4:2	XFSZ	Transfer Set Size	Transfer Set Size (Default: 000) These bits set the transfer data size of each bus operation in the internal bus. When the transfer set size is set to four double words or greater, the data size actually transferred during a single bus operation does not always match the transfer set size. Refer to “14.3.7.2” and “14.3.8.2 Burst Transfer During Dual Address Transfer” for more information. 000: 1 byte 001: 2 byte 010: 4 byte 011: 8 bytes (1double word) 100: 4 double words 101: 8 double words 110: 16 double words (Single Address transfer only) 111: 32 double words (Single Address transfer only)	R/W
1	MEMIO	Memory to I/O	Memory to I/O (Default: 0) This bit specifies the transfer direction during Single Address transfer (DMCCRn.SNGAD = 1). Clear this bit when in the Memory Fill Transfer mode. The setting of this bit is ignored when Dual Address transfer is set (DMCCRn.SNGAD = 0). 1: From memory to I/O 0: From I/O to memory	R/W
0	SNGAD	Single Address	Single Address (Default: 0) This bit specifies whether the transfer method is Single Address transfer or Dual Address transfer. 1: Single Address transfer 0: Dual Address transfer	R/W

14.4.3. DMA Channel Status Register (DM0CSRn, DM1CSRn))

Offset Address: DMAC0 0xB038 (ch. 0) / 0xB078 (ch. 1) / 0xB0B8 (ch. 2) / 0xB0F8 (ch. 3)
DMAC1 0xB838 (ch. 0) / 0xB878 (ch. 1) / 0xB8B8 (ch. 2) / 0xB8F8 (ch. 3)

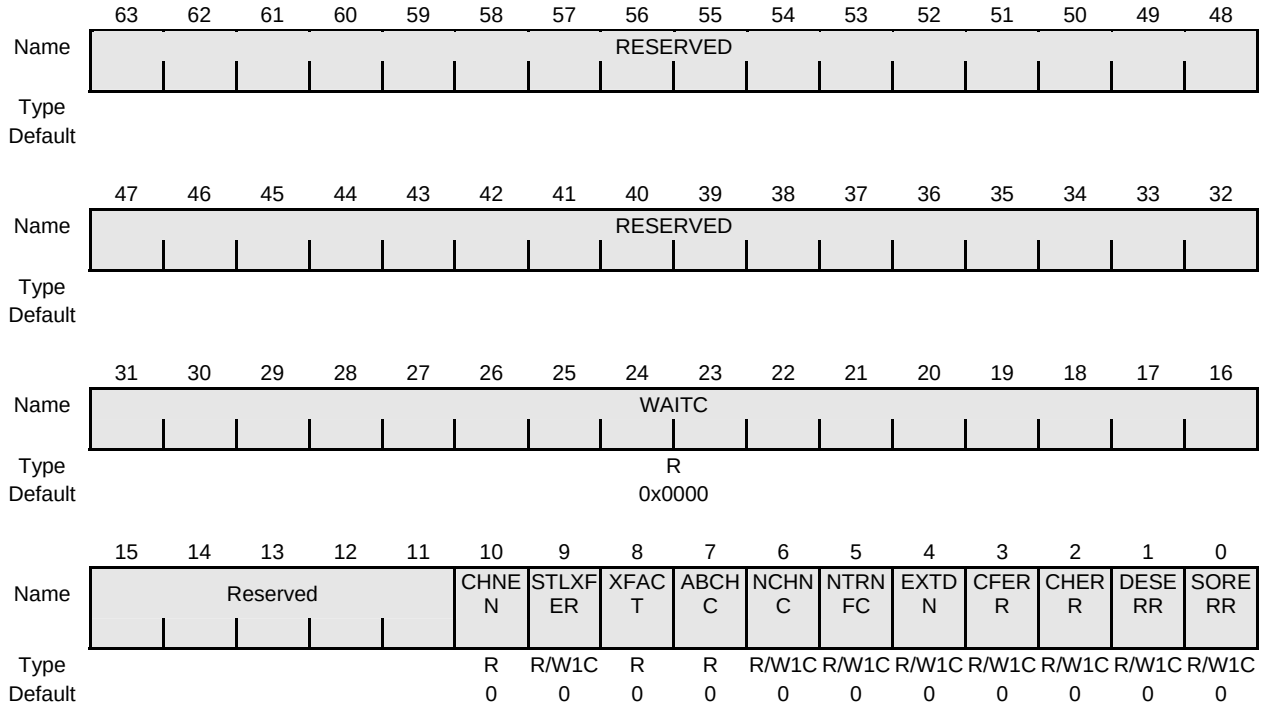


Figure 14-11 DMA Channel Status Register

Table 14-9 DMA Channel Status Register

Bit	Mnemonic	Field Name	Description	R/W
63:32		Reserved		—
31:16	WAITC	Wait Counter	Wait Counter (Default: 0x0000) This is a diagnostic function. • I/O DMA transfer mode (DMCCRn.EXTRQ = “1”) This counter is decremented by 1 at each 64 G-Bus cycles. After channel *n* releases bus ownership, this counter sets the default (the value that is the detection interval clock cycle count set by the Transfer Stall Detection Interval field (DMCCRn.STLTIME) divided by 64). The Transfer Stall Detect bit (DMCSRn.STLXFER) is set when the interval during which bus ownership is not held reaches the set clock cycle. The counter is reset to the default and stops counting. Clearing the Transfer Stall Detect bit (DMCSRn.STLXFER) resumes the count and starts stall detection. • Memory transfer mode (DMCCRn.EXTRQ = “0”) This counter is decremented by 1 at each G-Bus cycle. After bus ownership is released, the counter is set to the delay clock cycle count set by the Internal Request Delay field (DMCCRn.INTRQD). When the counter reaches “0” the count stops and channel *n* requests bus ownership.	R
15:11		Reserved		—
10	CHNEN	Chain Enable	Chain Enable (Default: 0) This value is a copy of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn).	R
9	STLXFER	Transfer Stall Detect	Stalled Transfer Detect (Default: 0) This bit indicates whether the interval during which bus ownership is not held exceeds the value set by the Transfer Stall Detect Interval field (DMCCRn.STLTIME) after bus ownership is released when in the I/O DMA transfer mode. 1: Indicates that the interval during which bus ownership was not held exceeds the DMCCRn.STLTIME setting. 0: The interval during which bus ownership was not held did not exceed the setting since this bit was last cleared.	R/W1C

Table 14-9 DMA Channel Status Register

Bit	Mnemonic	Field Name	Description	R/W
8	XFACT	Transfer Active	Transfer Active (Default: 0) This value is a copy of the Transfer Active bit (XFACT) of the DMA Channel Control Register (DMCCRn).	R
7	ABCHC	Error Complete	Error Completion (Default: 0) This bit indicates whether an error occurred during DMA transfer. This bit indicates the logical sum of the four error bits (CFERR, CHERR, DESERR, SORERR) in DMCSRn[3:0]. 1: DMA transfer ends due to an error. 0: No error occurred since this bit was last cleared.	R
6	NCHNC	Chain Complete	Normal Chain Completion (Default: 0) When performing chain DMA transfer, This bit indicates whether all DMA data transfers in the DMA Descriptor chain are complete. 1: All DMA data transfers in the DMA Descriptor chain ended normally. Or, DMA transfer that did not use a DMA Descriptor chain ended normally. 0: DMA transfer has not ended normally since this bit was last cleared.	R/W1C
5	NTRNFC	Transfer Complete	Normal Transfer Completion (Default: 0) This bit indicates whether DMA transfer ended according to the current DMA Channel Register setting. 1: DMA transfer ended normally. 0: DMA transfer has not ended since this bit was last cleared.	R/W1C
4	EXTDN	External DONE Asserted	External Done Asserted (Default: 0) This bit indicates whether an external I/O device asserted the DMADONE* signal. When the DMADONE* signal is set to bidirectional, this bit is also set when the TX4939 asserts the DMADONE* signal. 1: DMADONE* signal was asserted. 0: DMADONE* signal was not asserted.	R/W1C
3	CFERR	Configuration Error	Configuration Error (Default: 0) Indicates whether an illegal register setting was made. 1: There was a configuration error. 0: There was no configuration error.	R/W1C
2	CHERR	Chain Bus Error	Chain Bus Error (Default: 0) This bit indicates whether a bus error occurred while reading a DMA Command Descriptor. 1: Bus error occurred. 0: No bus error occurred.	R/W1C
1	DESERR	Destination Error	Destination Bus Error (Default: 0) This bit indicates whether a bus error occurred during a destination bus Write operation (a Write to a set DMDARn address). 1: Bus error occurred. 0: No bus error occurred.	R/W1C
0	SORERR	Source Bus Error	Source Bus Error (Default: 0) This bit indicates whether a bus error occurred during either a source bus Read or Write operation (A Read or Write to a set DMSARn address). 1: Bus error occurred. 0: No bus error occurred.	R/W1C

14.4.4. DMA Source Address Register (DM0SARn, DM1SARn)

Offset Address: DMAC0 0xB008 (ch. 0) / 0xB048 (ch. 1) / 0xB088 (ch. 2) / 0xB0C8 (ch. 3)
DMAC1 0xB808 (ch. 0) / 0xB848 (ch. 1) / 0xB888 (ch. 2) / 0xB8C8 (ch. 3)

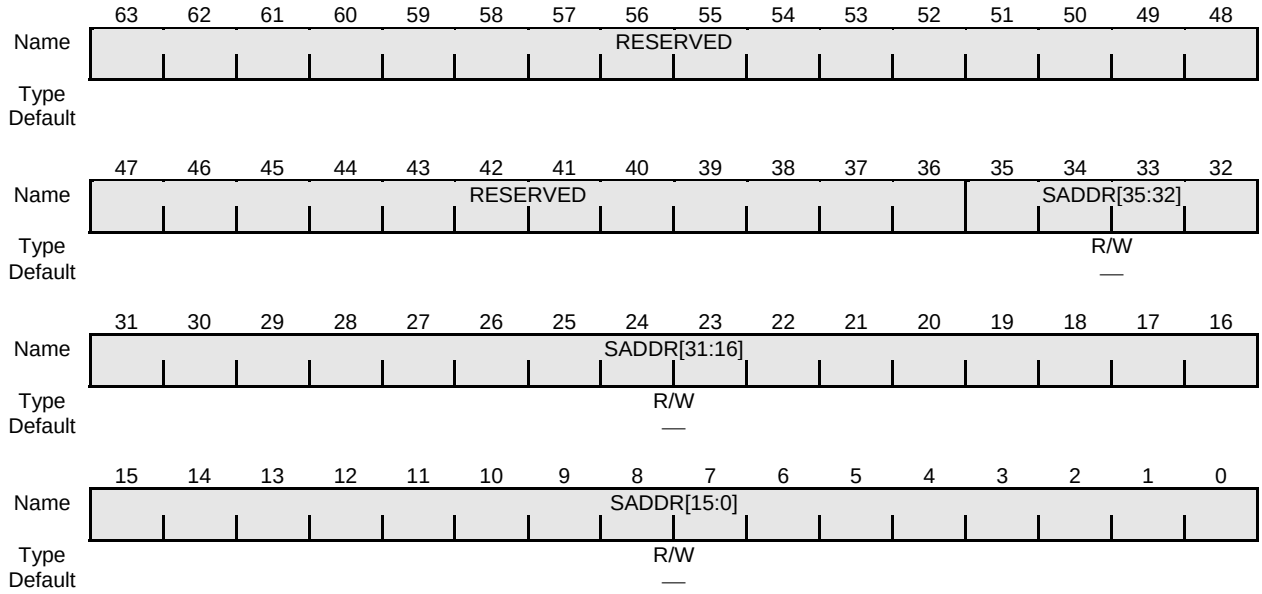


Figure 14-12 DMA Source Address Register

Table 14-10 DMA Source Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:36	—	Reserved		—
35:0	SADDR	Source Address	Source Address (Default: Undefined) This field sets the physical address of the transfer source during Dual Address transfer. This field sets the physical address of memory access during Single Address transfer. This field is used for either Memory-to-I/O or I/O-to-Memory transfers. Refer to “14.3.7.1” and “14.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes once for each bus operation only by the size that was transferred. During Single transfer, the value only changes by the value specified by the DMA Source Address Increment Register (DMSAIRn).	R/W

14.4.5. DMA Destination Address Register (DM0DARn, DM1DARn)

Offset Address: DMAC0 0xB010 (ch. 0) / 0xB050 (ch. 1) / 0xB090 (ch. 2) / 0xB0D0 (ch. 3)
DMAC1 0xB810 (ch. 0) / 0xB850 (ch. 1) / 0xB890 (ch. 2) / 0xB8D0 (ch. 3)

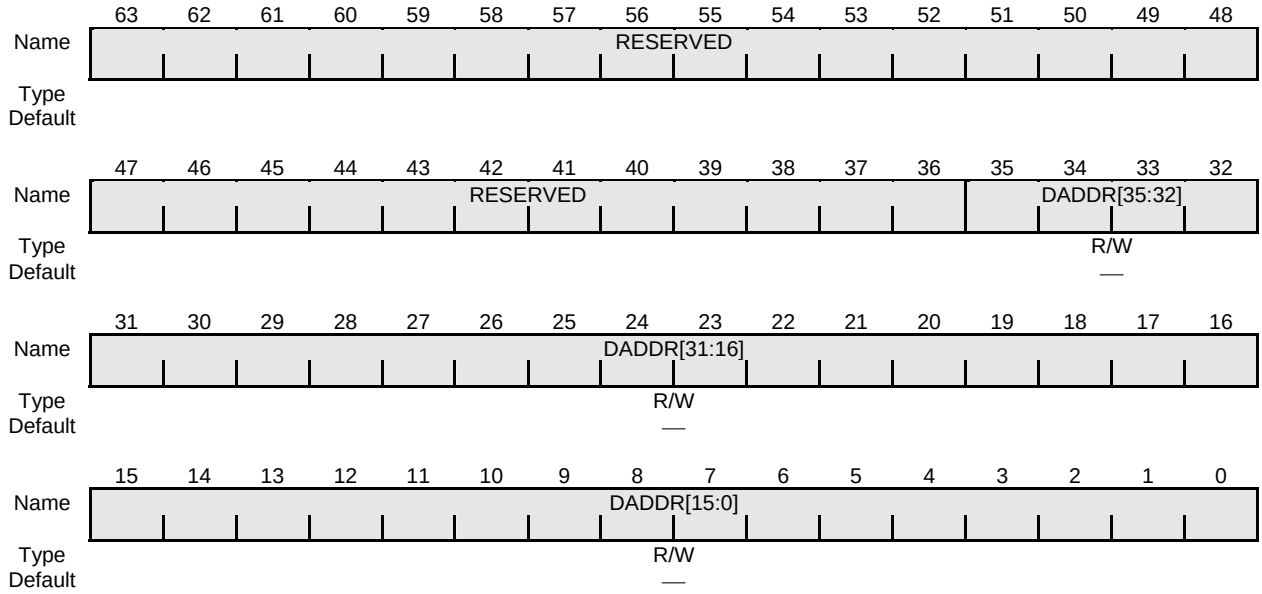


Figure 14-13 DMA Destination Address Register

Table 14-11 DMA Destination Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:36		Reserved		—
35:0	DADDR	Destination Address	Destination Address (Default: undefined) This register sets the physical address of the transfer destination during Dual Address transfer. This register is ignored during Single Address transfer. Refer to “14.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes only by the size of data transferred during each single bus operation. During Single transfer, the value only changes by the value specified by the DMA Destination Address Increment Register (DMDAIRn).	R/W

14.4.6. DMA Chain Address Register (DM0CHARn, DM1CHARn)

Offset Address: DMAC0 0xB000 (ch. 0) / 0xB040 (ch. 1) / 0xB080 (ch. 2) / 0xB0C0 (ch. 3)
DMAC1 0xB800 (ch. 0) / 0xB840 (ch. 1) / 0xB880 (ch. 2) / 0xB8C0 (ch. 3)

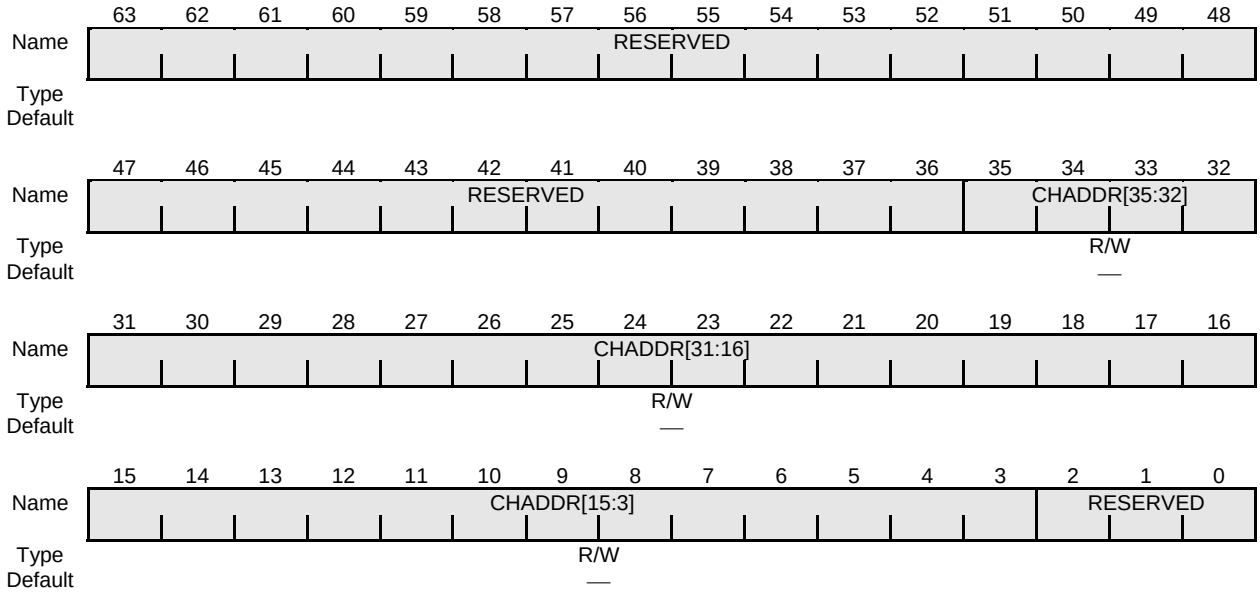


Figure 14-14 DMA Chain Address Register

Table 14-12 DMA Chain Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:36		Reserved		—
35:3	CHADDR	Chain Address	Chain Address (Default: undefined) When Chain DMA transfer is executed, this register sets the physical address of the next DMA Command Descriptor to be read. If DMA transfer according to the current Channel Register setting ends and the Chain Enable bit (DMCCRn.CHNEN) is set, then the DMA Command Descriptor is loaded in the Channel Register starting from the address indicated by this register. When a value other than “0” is set in this register, the Chain Enable bit (DMCCRn.CHNEN) and the Transfer Active bit (DMCCRn.XFACT) are set. When “0” is set in this register, only the Chain Enable bit (DMCCRn.CHNEN) is cleared. When the Chain Address field value reads a DMA Command Descriptor of 0, the value of this register is not updated and the value before that one (address of the Data Command Descriptor when the value of the Chain Address field being read was “0”) is held.	R/W
2:0		Reserved		R/W

14.4.7. DMA Source Address Increment Register (DM0SAIRn, DM1SAIRn)

Offset Address: DMAC0 0xB020 (ch. 0) / 0xB060 (ch. 1) / 0xB0A0 (ch. 2) / 0xB0E0 (ch. 3)
DMAC1 0xB820 (ch. 0) / 0xB860 (ch. 1) / 0xB8A0 (ch. 2) / 0xB8E0 (ch. 3)

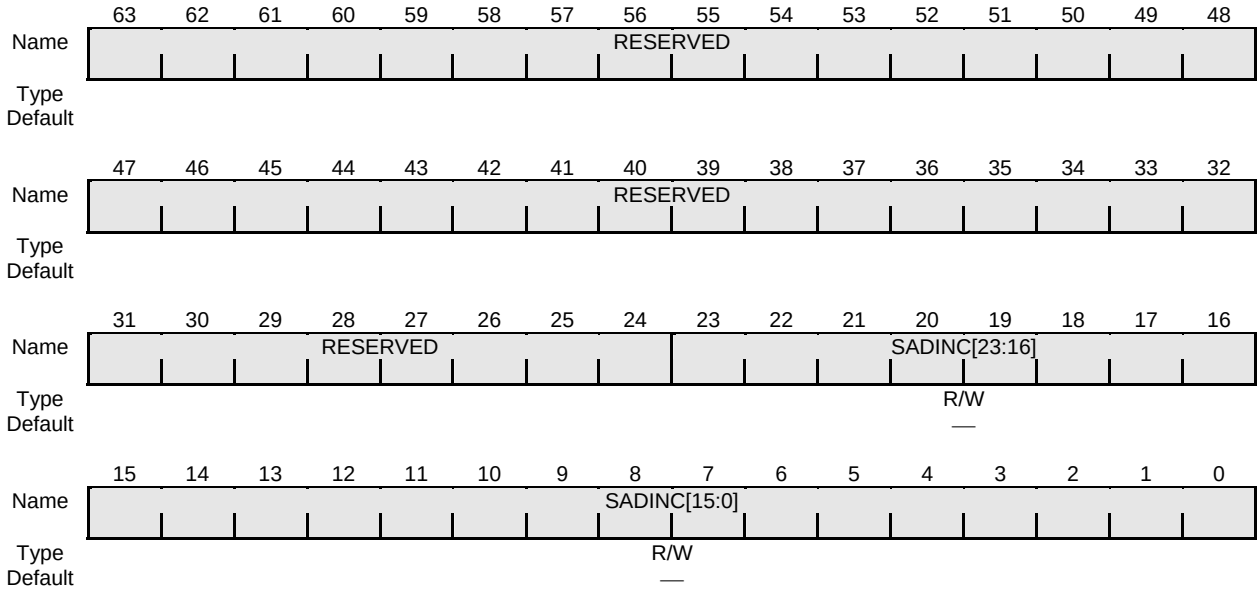


Figure 14-15 DMA Source Address Increment Register

Table 14-13 DMA Source Address Increment Register

Bit	Mnemonic	Field Name	Description	R/W
63:24		Reserved		—
23:0	SADINC	Source Address Increment	Source Address Increment (Default: undefined) This field sets the increase/decrease value of the DMA Source Address Register (DMSARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "14.3.7.1" and "14.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.	R/W

14.4.8. DMA Destination Address Increment Register (DM0DAIRn, DM1DAIRn)

Offset Address: DMAC0 0xB028 (ch. 0) / 0xB068 (ch. 1) / 0xB0A8 (ch. 2) / 0xB0E8 (ch. 3)
DMAC1 0xB828 (ch. 0) / 0xB868 (ch. 1) / 0xB8A8 (ch. 2) / 0xB8E8 (ch. 3)

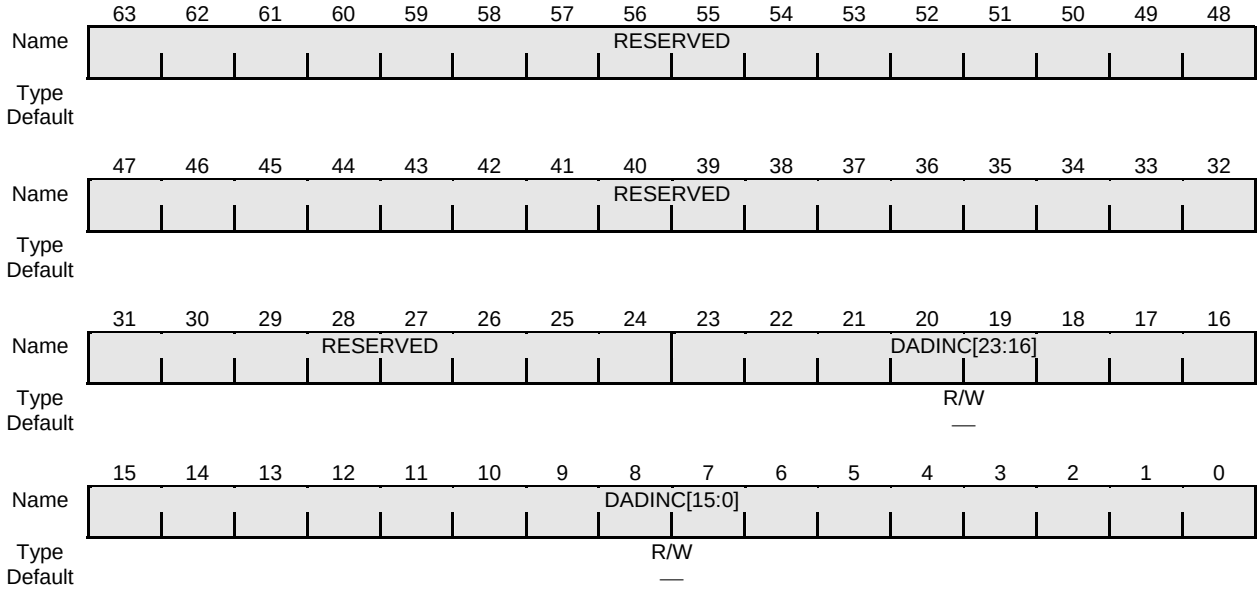


Figure 14-16 DMA Destination Address Increment Register

Table 14-14 DMA Destination Address Increment Register

Bit	Mnemonic	Field Name	Description	R/W
63:24		Reserved		—
23:0	DADINC	Destination Address Increment	Destination Address Increment (Default: undefined) This field sets the increase/decrease value of the DMA Destination Address Register (DMDARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "14.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.	R/W

14.4.9. DMA Count Register (DM0CNTRn, DM1CNTRn)

Offset Address: DMAC0 0xB018 (ch. 0) / 0xB058 (ch. 1) / 0xB098 (ch. 2) / 0xB0D8 (ch. 3)
DMAC1 0xB818 (ch. 0) / 0xB858 (ch. 1) / 0xB898 (ch. 2) / 0xB8D8 (ch. 3)

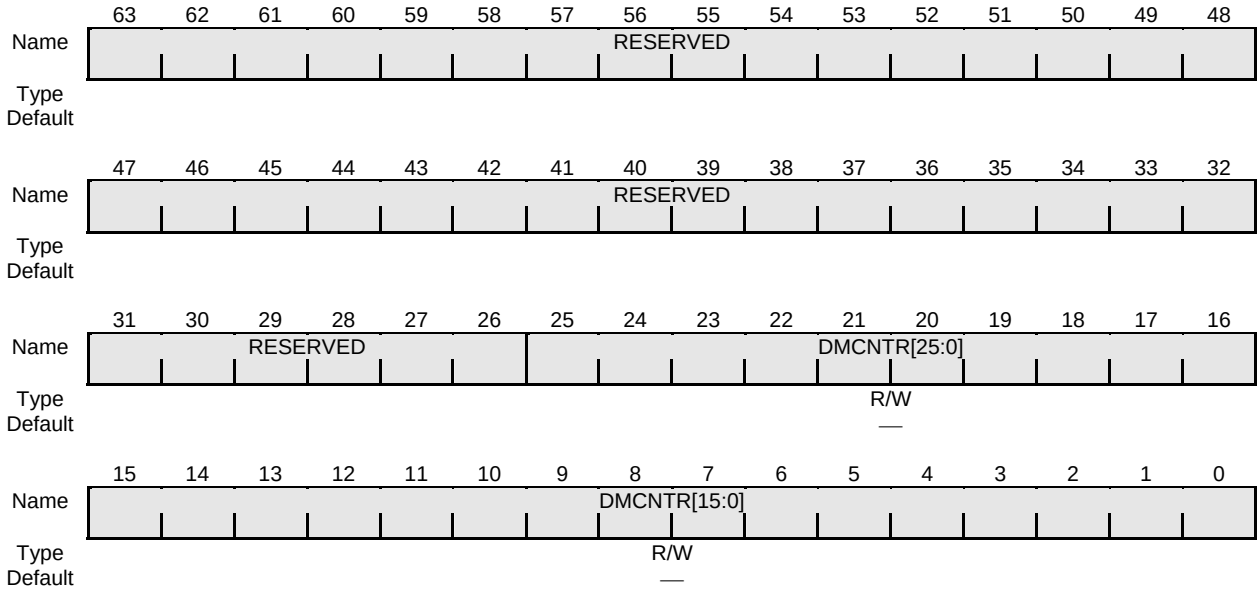


Figure 14-17 DMA Count Register

Table 14-15 DMA Count Register

Bit	Mnemonic	Field Name	Description	R/W
63:26		Reserved		—
25:0	DMCNTR	Count	Count Register (Default: undefined) This register sets the byte count that is transferred by the DMA Channel Register setting. The value is a 26-bit unsigned data that is decremented only by the size of the data transferred during a single bus operation. Refer to “14.3.8.1 Channel Register Settings During Dual Address Transfer” for more information.	R/W

14.4.10. DMA Memory Fiill Data Register (DM0MFDR, DM1MFDR)

Offset Address: DMAC0 0xB148, DMAC1 0xB948

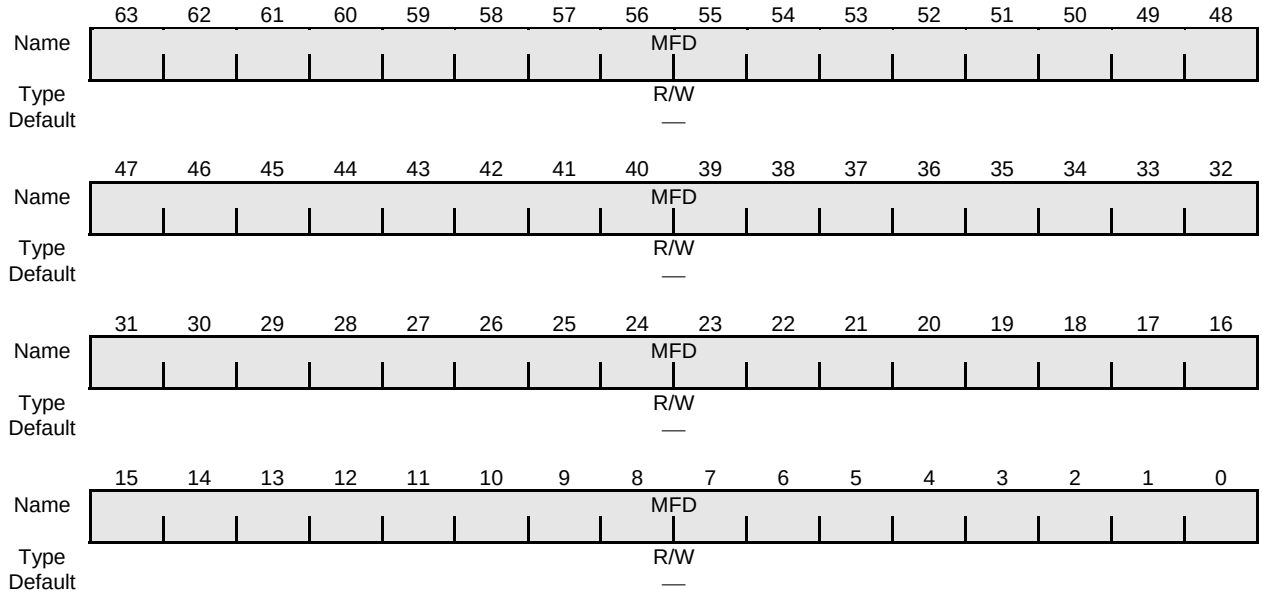


Figure 14-18 DMA Memory Fill Data Register

Table 14-16 DMA Memory Fill Data Register

Bit	Mnemonic	Field Name	Description	R/W
63:0	MFD	Memory Fill Data	Memory Fill Data (Default: undefined) This register, which stores double-word data written to memory when in the Memory Fill Transfer mode, is shared between all channels.	R/W

14.5. Timing Diagrams

This section contains timing diagrams for the external I/O DMA transfer mode. The DMAREQ[n] signals and DMAACK[n] signals in the timing diagrams are set to Low Active.

14.5.1. External I/O Device – SRAM Dual Address Transfer

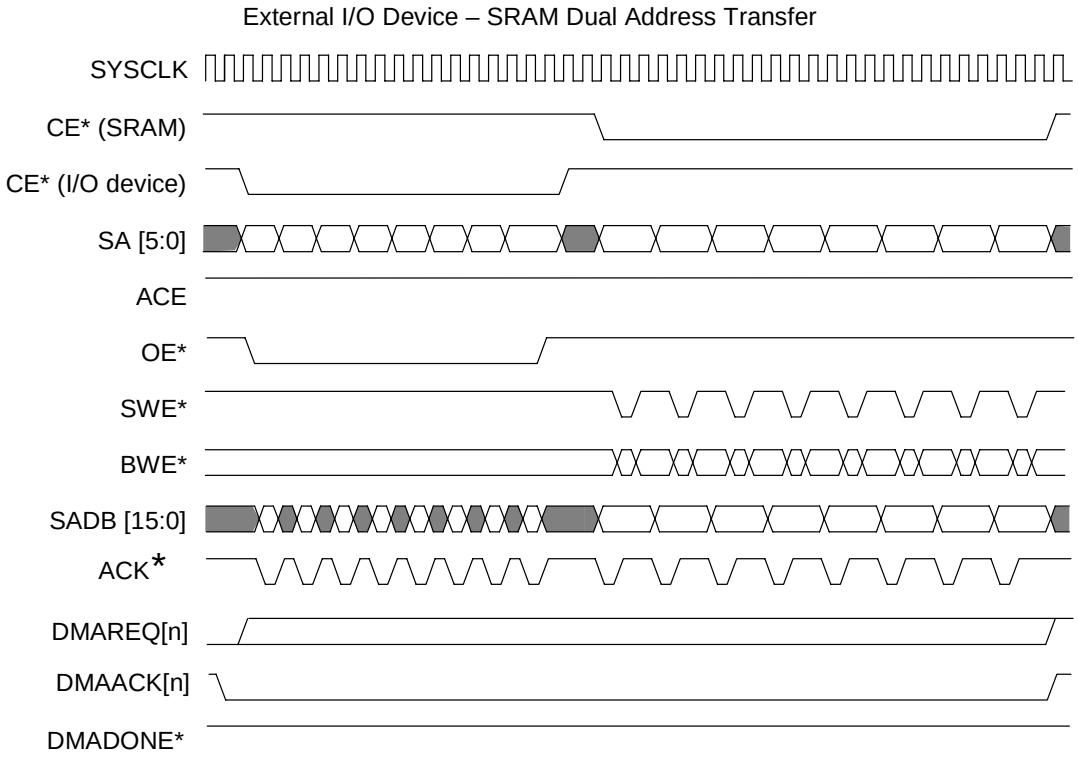


Figure 14-19 Dual Address Transfer from External I/O Device to SRAM (4-word Burst Transfer to 16-bit Bus SRAM)

Dual Address Transfer from Memory to External I/O Device

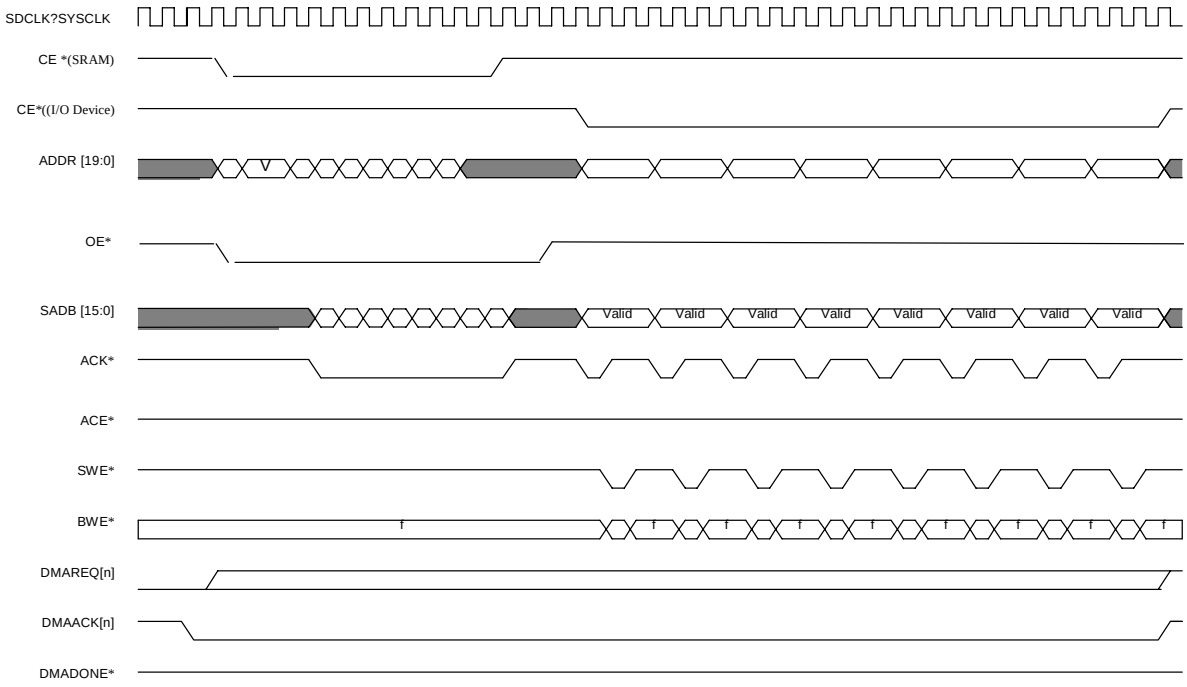


Figure 14-20 Dual Address Transfer from Memory to External I/O Device (4-word Burst Transfer from 16-bit Bus SRAM)

14

14.5.2. External I/O Device – SRAM Dual Address Transfer

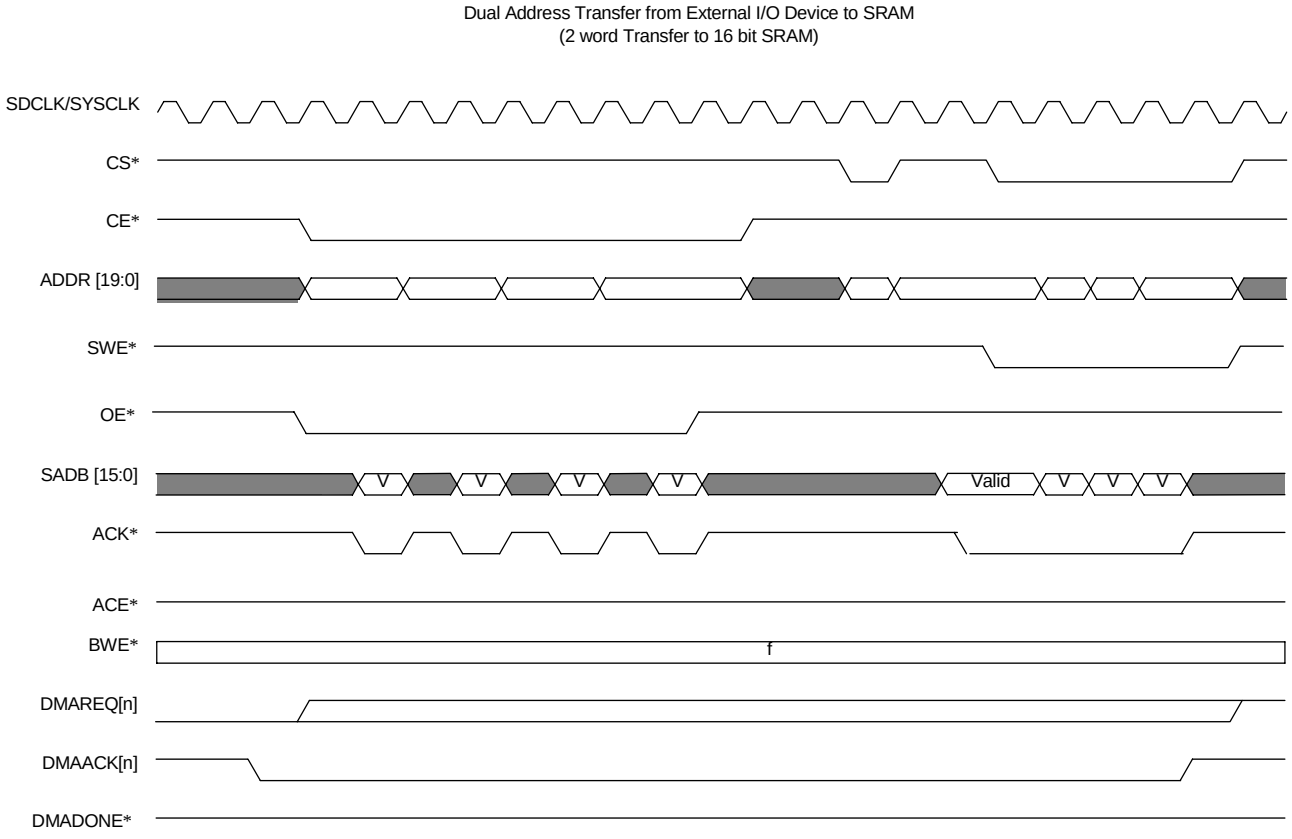


Figure 14-21 Dual Address Transfer from External I/O Device to SRAM (2-word Burst Transfer to 16-bit SRAM)

14.5.3. External I/O Device (Non-burst) – Memory Dual Address Transfer

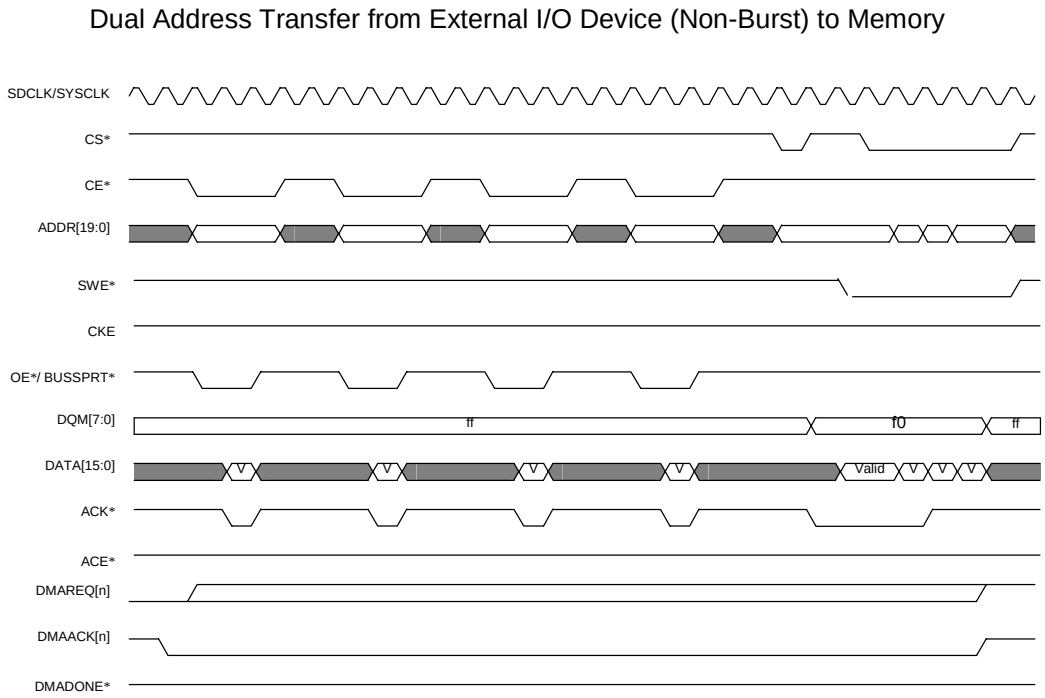


Figure 14-22 Dual Address Transfer from External I/O Device (Non-Burst) to Memory (2-word Burst Transfer to 16-bit SRAM: Set DMCCRn.SBINH to "1")

Dual Address Transfer from Memory to External I/O Device

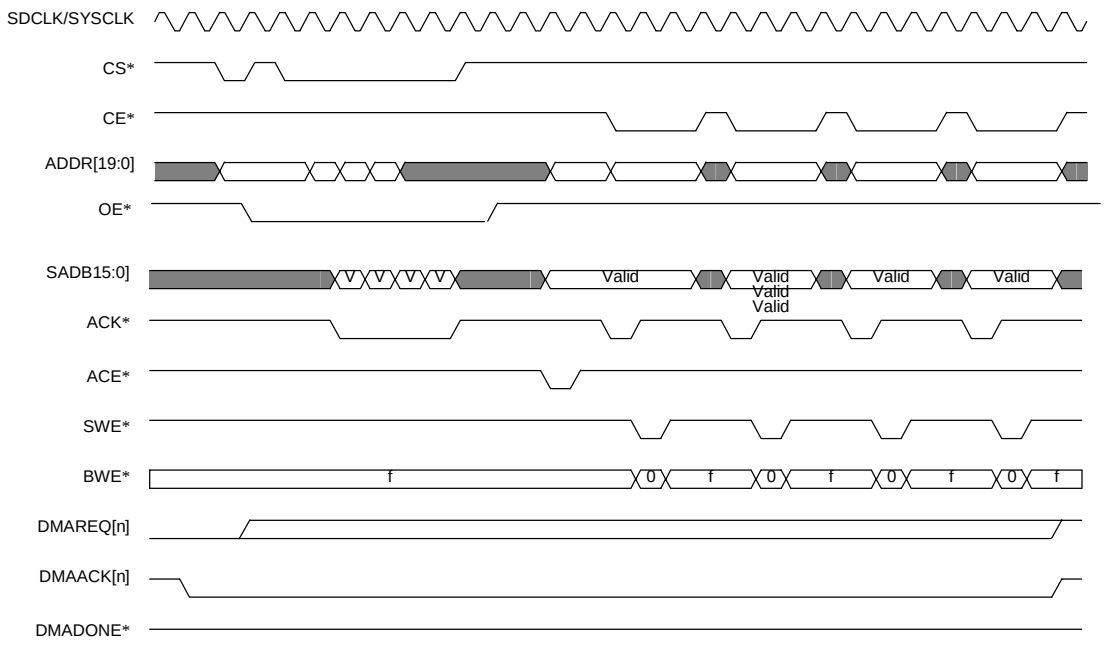


Figure 14-23 Dual Address Transfer from Memory to External I/O Device (2-word Burst Transfer from 16-bit SRAM: Set DMCCRn.DBINH to "1")

Chapter 15. DDR SDRAM Controller

15.1. Features

The DDR SDRAM Controller generates the clock and control signals that are required by the DDR SDRAM interface. The DDR SDRAM Controller has 2 on-chip channels and can support memory sizes of up to 2 GB (1 GB/Channel) by supporting various memory configurations.

- *Memory Clock Frequency: from 100 MHz to 200 MHz*
- *2 Independent Memory Channels*
- *Supports 2- or 4-bank 16 Mb, 64 Mb, 128 Mb, 256 Mb, 512 Mb or 1 Gb DDR SDRAM*
- *Can use registered DIMM*
- *Memory data bus width is 32 bits (word or W)*
- *Internal data bus width is 64 bits (double-word or DW)*
- *Supports critical first word access of the TX49/H4 core*
- *Low power consumption mode: Can select Self-refresh or Pre-charge Power Down*
- *Supports DMAC special Burst access (address decrement/fix)*
- *Can write to any byte during Single or Burst Write operation. This feature is controlled by the DQM signal.*
- *Supports burst length of up to 32DWs on GBUS*
- *Programmable Drive Strength for DDR signals*

15.2. Register Map

Offset Address	Bit Width	Register Symbol	Register Name
0x8000	16	DDRCTL00	DDR_CTL_00
0x8008	16	DDRCTL01	DDR_CTL_01
0x8010	16	DDRCTL02	DDR_CTL_02
0x8018	16	DDRCTL03	DDR_CTL_03
0x8020	16	DDRCTL04	DDR_CTL_04
0x8028	16	DDRCTL05	DDR_CTL_05
0x8030	16	DDRCTL06	DDR_CTL_06
0x8038	16	DDRCTL07	DDR_CTL_07
0x8040	16	DDRCTL08	DDR_CTL_08
0x8048	16	DDRCTL09	DDR_CTL_09
0x8050	16	DDRCTL10	DDR_CTL_10
0x8058	16	DDRCTL11	DDR_CTL_11
0x8060	16	DDRCTL12	DDR_CTL_12
0x8068	16	DDRCTL13	DDR_CTL_13
0x8070	16	DDRCTL14	DDR_CTL_14
0x8078	16	DDRCTL15	DDR_CTL_15
0x8080	16	DDRCTL16	DDR_CTL_16
0x8088	16	DDRCTL17	DDR_CTL_17
0x8090	16	DDRCTL18	DDR_CTL_18
0x8098	16	DDRCTL19	DDR_CTL_19
0x80A0	16	DDRCTL20	DDR_CTL_20
0x80A8	16	DDRCTL21	DDR_CTL_21
0x80B0	16	DDRCTL22	DDR_CTL_22
0x80B8	16	DDRCTL23	DDR_CTL_23
0x80C0	16	DDRCTL24	DDR_CTL_24
0x80C8	16	DDRCTL25	DDR_CTL_25
0x80D0	16	DDRCTL26	DDR_CTL_26
0x80D8	16	DDRCTL27	DDR_CTL_27
0x80E0	16	DDRCTL28	DDR_CTL_28
0x80E8	16	DDRCTL29	DDR_CTL_29
0x80F0	16	DDRCTL30	DDR_CTL_30
0x80F8	16	DDRCTL31	DDR_CTL_31
0x8100	16	DDRCTL32	DDR_CTL_32
0x8108	16	DDRCTL33	DDR_CTL_33
0x8110	16	DDRCTL34	DDR_CTL_34
0x8118	16	DDRCTL35	DDR_CTL_35
0x8120	16	DDRCTL36	DDR_CTL_36
0x8128	16	DDRCTL37	DDR_CTL_37
0x8130	16	DDRCTL38	DDR_CTL_38
0x8138	16	DDRCTL39	DDR_CTL_39
0x8140	16	DDRCTL40	DDR_CTL_40
0x8148	16	DDRCTL41	DDR_CTL_41
0x8150	16	DDRCTL42	DDR_CTL_42
0x8158	16	DDRCTL43	DDR_CTL_43
0x8160	16	DDRCTL44	DDR_CTL_44
0x8168	16	DDRCTL45	DDR_CTL_45
0x8170	16	DDRCTL46	DDR_CTL_46
0x8200	64	DRWINEN	Enable/Disable DDR Mapping Windows
0x8208	64	DRWIN00	DDR Mapping Window #0
0x8210	64	DRWIN01	DDR Mapping Window #1
0x8218	64	DRWIN02	DDR Mapping Window #2
0x8220	64	DRWIN03	DDR Mapping Window #3

15.3. DDR SDRAM Interface

15.3.1. Pin Signals

DDR SDRAM Interface Signals		
Signal Name	**I/O**	**Function**
DRA[13:0]	Output	Address This address signals are for DDR SDRAM.
DRDQ[31:0]	I/O	Data Bus This is a 32-bit data bus dedicated for DDR SDRAM
DRDQS[3:0]	I/O	DDR Control Signal for each byte data
DRDM[3:0]	Output	DDR Control signal for each byte data
DRCS[1:0]*	Output	DDR Chip Select
DRVref	Input	DDR Voltage reference input
DRCKP, DRCKM	Output	Differential DDR Clock Source
DRCKOUT	Output	Single-end DDR Reference Clock for deskew
DRCKREF	Input	DDR Clock deskew reference input
DRBA[1:0]	Output	DDR Bank Address
DRWE*, DRCAS*, DRRAS*	Output	DDR Control Signals
DRCKE	Output	DDR Control Signals Clock Enable signal

15.4. Read

READ bursts are initiated with a READ command, as shown in Figure 15-1.

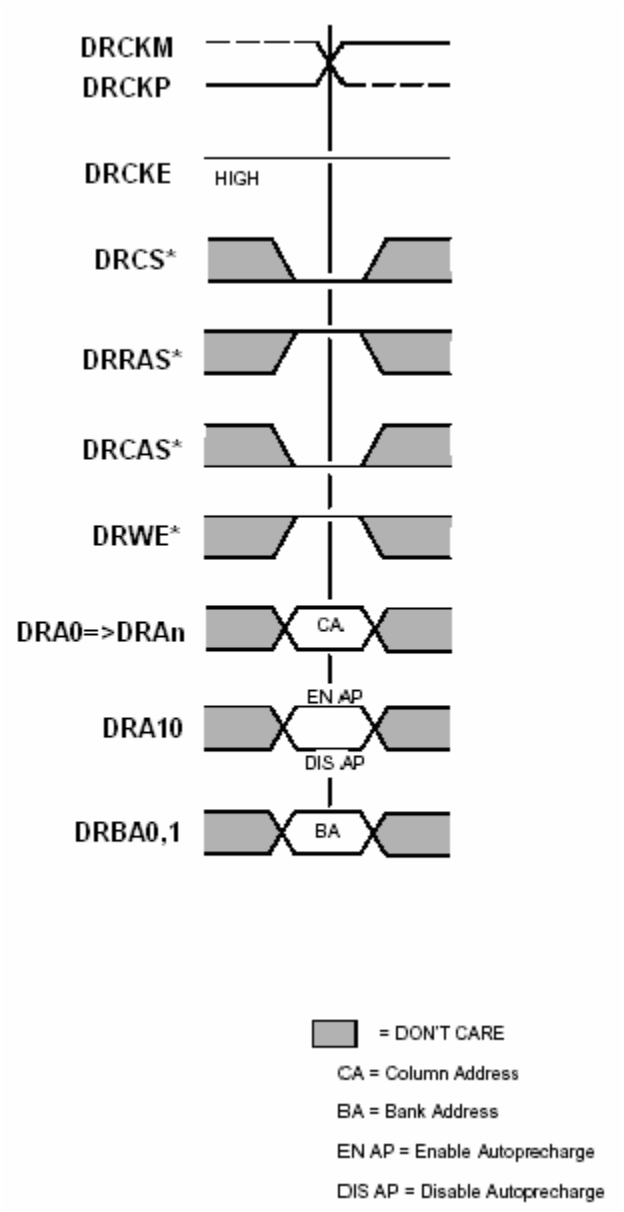


Figure 15-1 Read Command

Figure 15-2 shows an example of burst read of 4 with a CAS latency of 2 (CL=2).

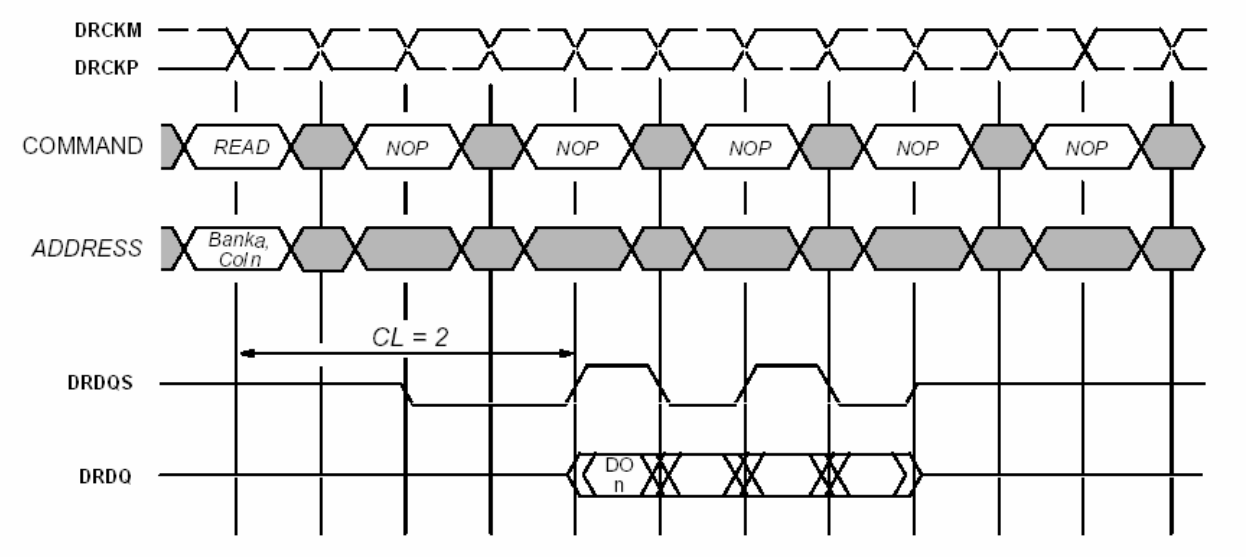


Figure 15-2 Burst Read of 4

15.4.1. Write

WRITE bursts are initiated with a WRITE command, as shown in Figure 15-3.

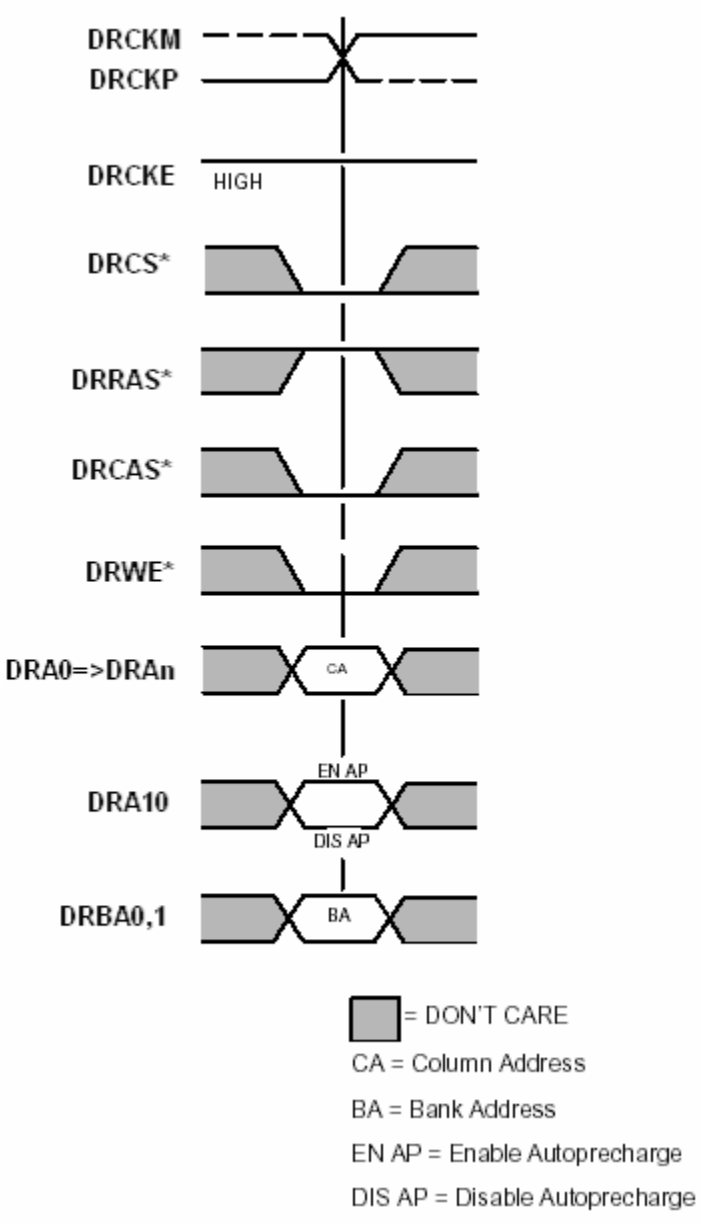


Figure 15-3 Write Command

Figure 15-4 shows an example of burst write of 4.

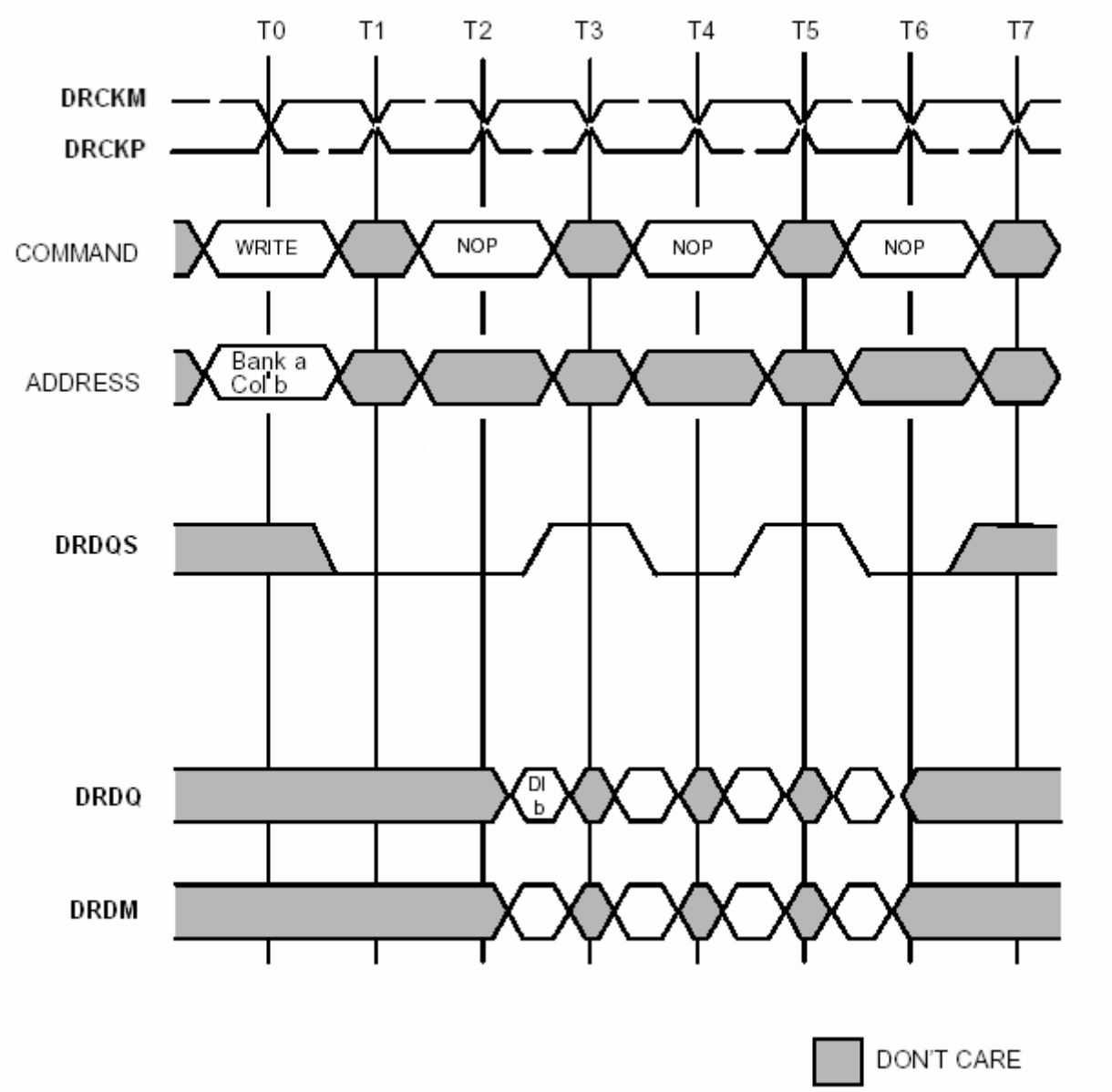


Figure 15-4 Example of Burst Write 4

15.5. Precharge

The PRECHARGE command is illustrated in Figure 15-5.

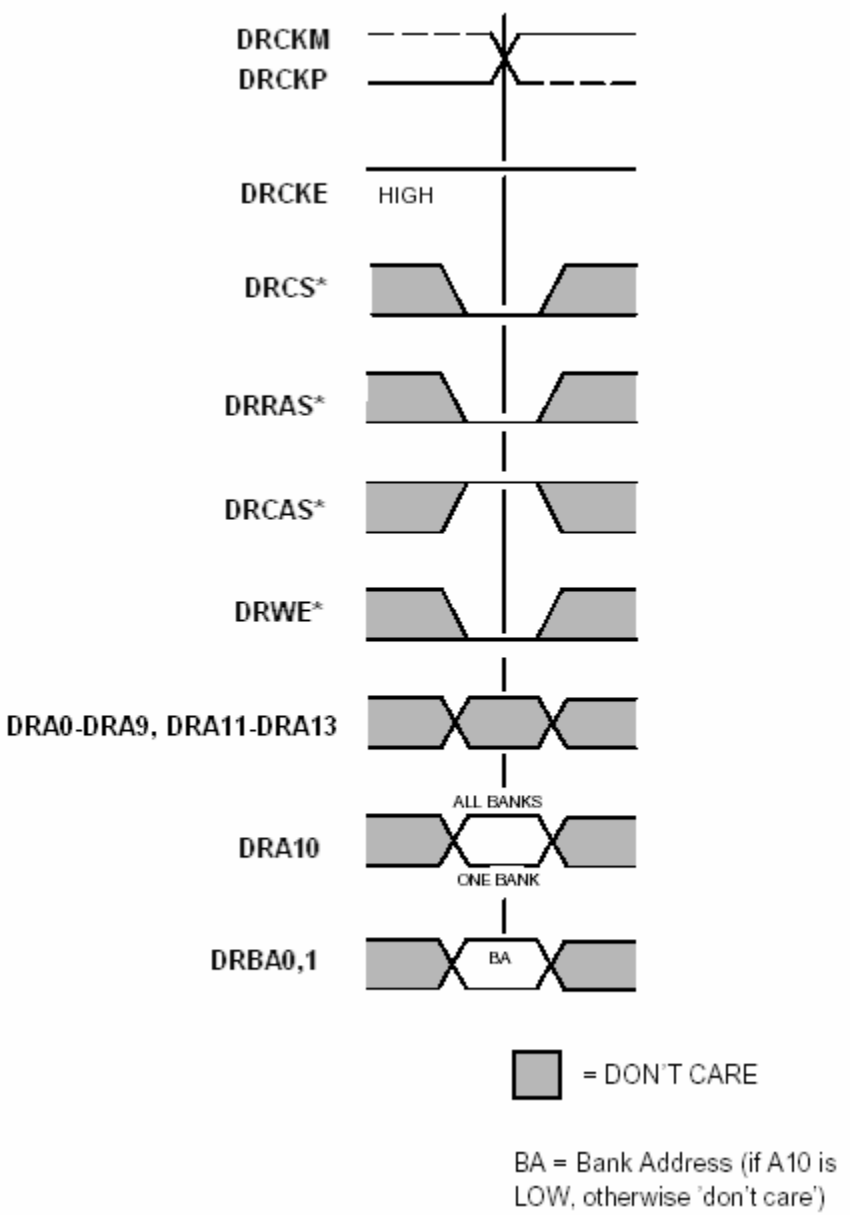


Figure 15-5 Precharge Command

15.6. Power-Down

The POWER-DOWN command is illustrated in Figure 15-6.

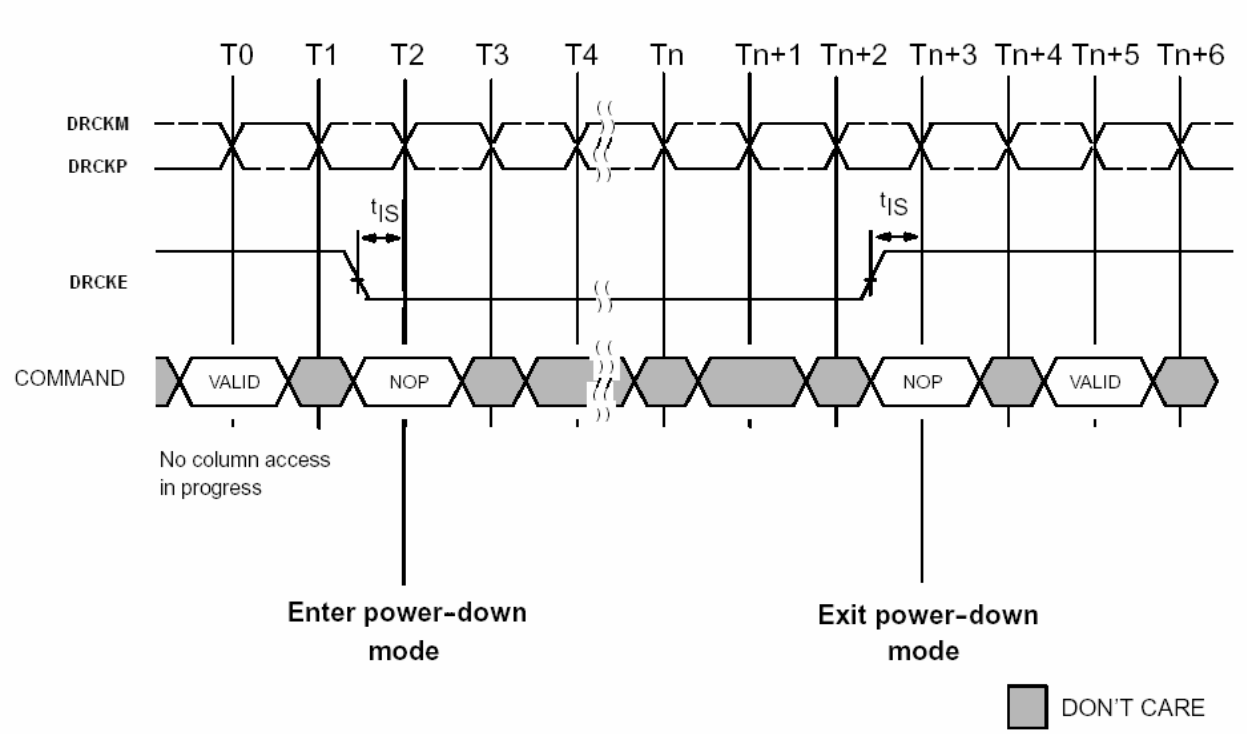


Figure 15-6 Power-Down Command

15

15.7. Memory Access

15.7.1. Address Mapping

It is possible to map the DRAM address to an arbitrary physical address using the DDR Mapping Window registers. When the system receives an effective physical address, (EPA), it checks following condition for enabled window(s).

IF ((EPA <= DRWINUP_ADRS) && (EPA >= DRWINLO_ADRS))

Then, if it is true, Corresponding DDR channel will be accessed with following DDR_EFFECTIVE_ADDRESS.

DDR_EFFECTIVE_ADDRESS = EPA – DRWINLO_ADRS + DRWINOF_ADRS

Each window has a resolution of 1MB.

Figure 15-7 shows an example of DDR Memory Mapping. In this example, DDR Channel 0 (CS0) is split two blocks, one is 384 MB and the rest is 640 MB. The 384 MB portion is mapped to 0x0_0000_0000 to 0x0_17FF_FFFF. The rest is mapped to 0x0_8000_0000 to 0x0_A7FF_FFFF and creates continuous memory space with 0x0_4000_0000 to 0x0_7FFF_FFFF memory space, which is mapped from DDR Channel 1 (CS1).

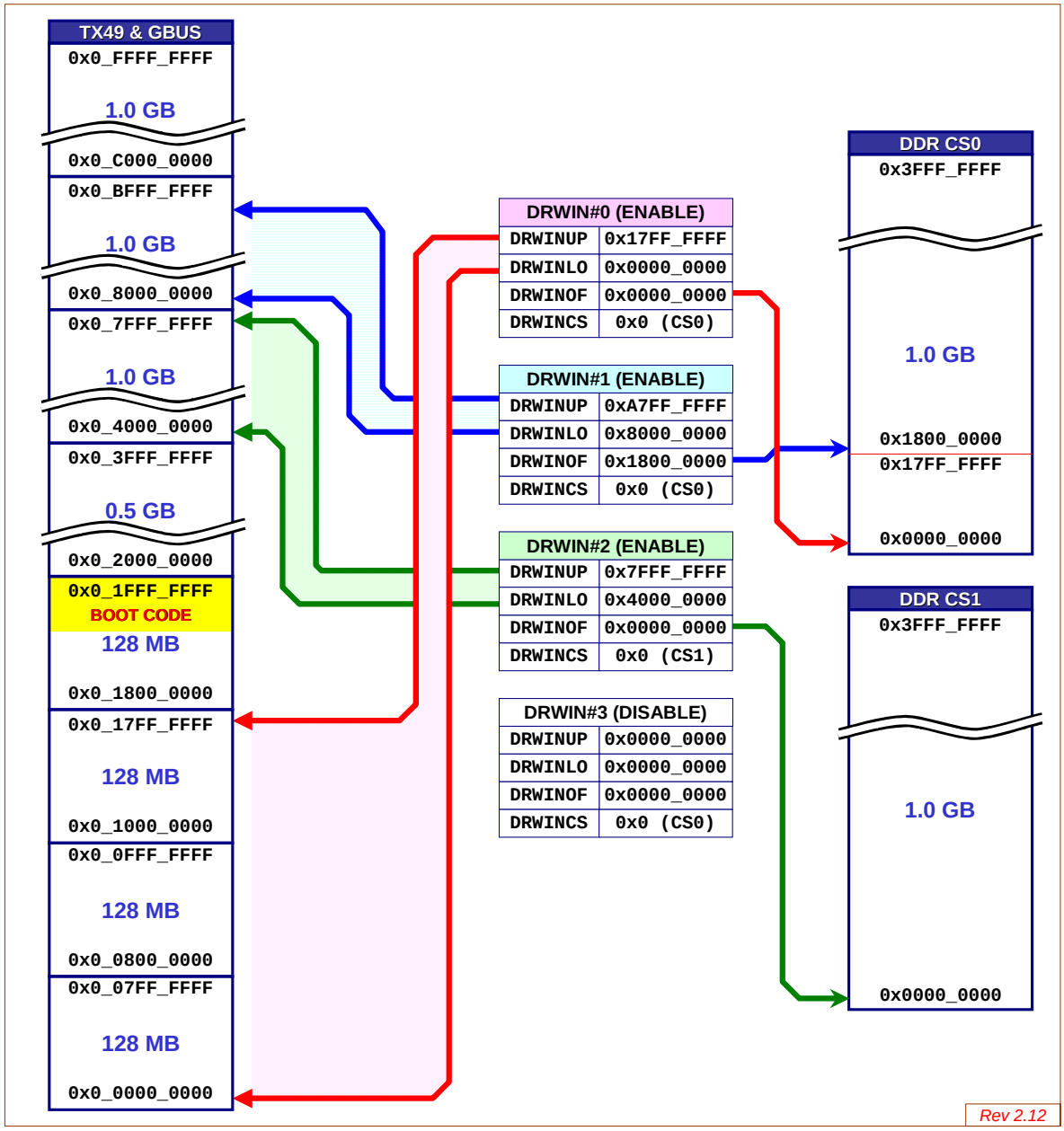


Figure 15-7 Example of DDR Memory Split-Mapping

15.7.2. Wrapper Register

Register must be accessed using 32-bit or 64-bit load and store instructions.

15.7.2.1. Wrapper Control Register

Offset Address	Bit Width	Register Symbol	Register Name
0x8200	64	DRWINEN	Enable/Disable DDR Mapping Windows
0x8208	64	DRWIN00	DDR Mapping Window #0
0x8210	64	DRWIN01	DDR Mapping Window #1
0x8218	64	DRWIN02	DDR Mapping Window #2
0x8220	64	DRWIN03	DDR Mapping Window #3

15.7.2.2. DDR Mapping Window Control (DRWINEN)

This registers enables or disables each DDR Mapping Register also defines the drive strength of DDR memory signals. Before enabling any Mapping Window, i.e. Windows #0 to #3, corresponding registers, i.e. DRWIN00, DRWIN01, DRWIN02, and DRWIN03, has to be preset with precise value.

If failed, the system might cause serious malfunction and result is unknown.

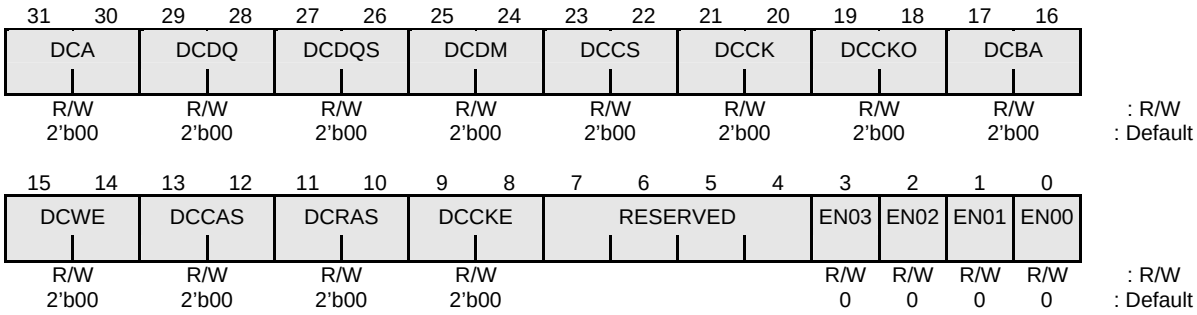


Figure 15-8 DDR Mapping Window Control

Table 15-1 DDR Mapping Window Control

Bit	Mnemonic	Field Name	Description	Initial Value	R/W
63:32	—	Reserved	—	—	—
31:30	DCA	DC_DRA	Drive Strength Control of DRA[13:0] signal 2'b00 = 28% of maximum strength 2'b01 = 57% of maximum strength 2'b10 = 71% of maximum strength .. 2'b11 = maximum strength (100%)	2'b00	R/W
29:28	DCDQ	DSC_DRDQ	Drive Strength Control of DRDQ[31:0] Control is same as above	2'b00	R/W
27:26	DCDQS	DSC_DRDQS	Drive Strength Control of DRDQS[3:0] Control is same as above	2'b00	R/W
25:24	DCDM	DSC_DRDM	Drive Strength Control of DRDM[3:0] Control is same as above	2'b00	R/W
23:22	DCCS	DSC_DRCS	Drive Strength Control of DRCS[1:0] Control is same as above	2'b00	R/W
21:20	DCCK	DSC_DRCK	Drive Strength Control of DRCKP and DRCKP Control is same as above	2'b00	R/W
19:18	DCCKO	DSC_DRCKOUT	Drive Strength Control of DRCKOUT Control is same as above	2'b00	R/W
17:16	DCBA	DSC_DRBA	Drive Strength Control of DRBA[1:0] Control is same as above	2'b00	R/W
15:14	DCWE	DSC_DRWE	Drive Strength Control of DRWE Control is same as above	2'b00	R/W
13:12	DCCAS	DSC_DRCAS	Drive Strength Control of DRCAS Control is same as above	2'b00	R/W
11:10	DCRAS	DSC_DRRAS	Drive Strength Control of DRRAS Control is same as above	2'b00	R/W
9:8	DCCKE	DSC_DRCKE	Drive Strength Control of DRCKE Control is same as above	2'b00	R/W
3	EN03	ENWIN03	1'b1 = Enable Window #3, 1'b0 = Disable	0	R/W
2	EN02	ENWIN02	1'b1 = Enable Window #2, 1'b0 = Disable	0	R/W
1	EN01	ENWIN01	1'b1 = Enable Window #1, 1'b0 = Disable	0	R/W
0	EN00	ENWIN00	1'b1 = Enable Window #0, 1'b0 = Disable	0	R/W

15.7.2.3. DDR Mapping Window #n (n=0, 1, 2, 3)

DRWIN00, DRWIN01, DRWIN02, and DRWIN03 control DDR Mapping condition independently.

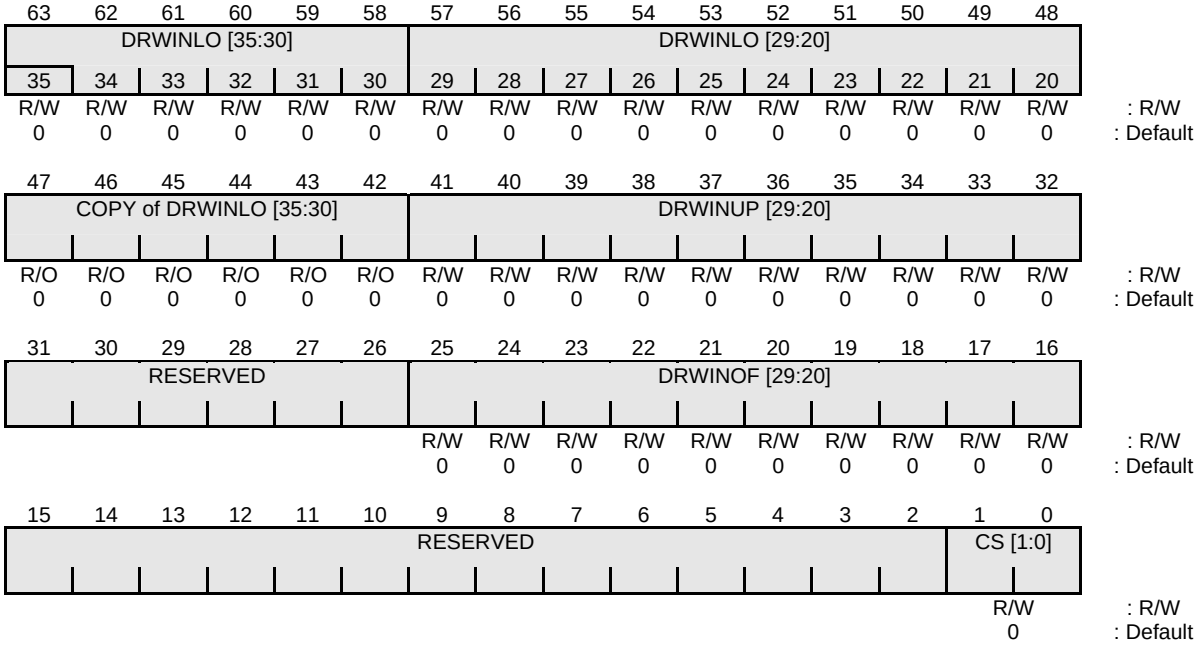


Figure 15-9 DDR Mapping Window Control

Table 15-2 DDR Mapping Window Control

Bit	Mnemonic	Field Name	Description	Initial Value	R/W

63:48	DRWINLO	DRWINLO [35:20]	Define the lower address of window #n in physical address.
			DRWINLO_ADRS = { DRWINUP [35:20], 20'HF_FFFF }
				0	R/W

47:42	DRWINUP	DRWINUP [35:30]	This register is the copy of DRWINLO [35:30]	0	R/O
41:32	DRWINUP	DRWINLO [29:20]	Define the upper address of window #n in physical address. DRWINUP _ADRS = { DRWINUP [35:20], 20'H0_0000 }	0	R/W
31:26	—	RESERVED	—	—	—
25:16	DRWINOF	DRWINOF [29:20]	Define the offset address of the target DDR memory space. DRWINOF _ADRS = { DRWINOF [29:20], 20'H0_0000 }	0	R/W
15:2	—	RESERVED	—	—	—
1:0	CS [1:0]	Chip Select [1:0]	Define corresponding DDR channel number with CS CS[1:0] = 2'b00 Channel 0 (CS0) CS[1:0] = 2'b01 Channel 0 (CS0) CS[1:0] = 2'b10 Channel 1 (CS1) CS[1:0] = 2'b11 Channel 1 (CS1) NOTE: CS setting should be consistent with DDR_CTRL15 setting.	0	R/W

When the system receives an effective physical address, (EPA), it checks following condition for enabled window(s).

IF ((EPA <= DRWINUP_ADRS) && (EPA >= DRWINLO_ADRS))

Then, if it is true, Corresponding DDR channel will be accessed with following DDR_EFFECTIVE_ADDRESS.

DDR_EFFECTIVE_ADDRESS = EPA – DRWINLO_ADRS + DRWINOF_ADRS

15.8. DDR SDRAM Controller

The DDR SDRAM controller is configured with the following features:

- *Maximum Byte Request = 256 bytes*
- *Register Data Width = 16 bits*
- *Half cas latency support*
- *Register DLL Value – DLL value is stored in a user-readable register*
- *Drive data/dqs when idle – allows the user (under register control) to have the controller drive the data and dqs signals during the times in which the controller is idle*
- *Support Registered DIMMs*
- *Self Refresh (register controllable)*
- *Masked Writes*
- *DLL Bypass Option*
- *Command Queue Depth = 4*
- *Write Queue Depth = 1 (no FIFO)*
- *Read Queue Depth = 1 (no FIFO)*
- *Supported clock frequencies from 100 MHz to 200 MHz*
- *Wrap command support for critical first word access*
- *Auto Precharge (register controllable)*
- *Memory Data Width = 32 bits*
- *Maximum Chip Selects = 2 (supports 2 physical banks)*
- *DQ:DQS Ratio = 8 (every data byte has one DQS signal)*
- *Maximum Columns Supported = 12 (supports x8 x16 DDR SDRAM)*
- *Maximum Address Pins = 14 (supports 1Gbit DDR SDRAM)*
- *Address Order = CS, Row, Bank*

The DDR SDRAM controller has the following advanced features:

- *Fully pipelined command, read, and write data interface to the controller*
- *Advanced bank look-ahead features for high memory throughput*
- *Programmable register interface to control memory device parameters and protocols including auto precharge*
- *Full initialization of memory upon reset of the controller*
- *Built in adjustable Delay Compensation Circuitry (DCC or DLL) for reliable data send and capture timing*

15.8.1. Initialization protocol

The controller is designed such that it requires the following sequence for correct operation after all power to the ASIC and to the memory devices are stable. The controller does not have any circuitry that controls the activation of power and ground to the system. Once the power is stable to the memory devices and the ASIC, the following procedure is required to initialize the controller. The controller will initialize the memory devices automatically once this procedure is completed.

- *Assert the reset pin. All programmable registers will be set to zero when the reset pin is asserted.*
- *De-assert the reset pin synchronously to the controller clock.*
- *Reset DDR deskew PLL by writing 0 and then 3 to the DDRDSKW field of the DLL De-Skew Control Register (see chapter on Configuration Registers).*
- *Issue write register commands to program all the registers to configure the DRAM protocols and the settings for the DCC in the controller. The Start register must be kept at 0 during this initialization step.*
  Register values to be used for a specific memory part can be obtained by running the register_gen.tcl tickle script with the corresponding memory part's SOMA file as input.
- *Write a 1 into the Start register. Once this register is written with a 1, the controller will go through the initialization sequence specified by the parameters written into the registers of the controller.*

The controller will generate an interrupt (INT_STATUS[2] in DDR_CTL_01 register) to the CPU once the DRAM initialization is complete and the internal DCC is locked.

15.8.2. Supported DDR SDRAM Configurations

This controller supports the SDRAM configurations listed below in Table 15-3.

Table 15-3 Supported DDR SDRAM Configration

DDR SDRAM Configuration			Row Address (bit)	Column Address (bit)
64Mbit	4-bank	1M x 16	12	8
128 Mbit	4-bank	1M x 32	12	8
		2M x 16	12	9
		4M x 8	12	10
256 Mbit	4-bank	2M x 32	12	9
		4M x 16	13	9
		8M x 8	13	10
512 Mbit	4-bank	8M x 16	13	10
		16M x 8	13	11
1 Gbit	4-bank	16M x 16	14	10
		32M x 8	14	11

15.8.3. Delay Line Tuning

Write Datapath

Essentially, for the write datapath the parameters need to be adjusted according the "flight times" while maintaining the 90 degree phase between the write data and the write DQS:

For the write datapath, there are two registers that control the capture of data at the DRAM interface:

- TC300C_WR_DQS_SHIFT[8:0] - Controls the delay for the clk_wr signal which is used to clock the write data such that the phase between the write data and write_dqs is always 90 degrees. The unit of this parameter is degree. For example, to delay by half of a clock cycle(equivalent to a 180 degree phase shift) TC300C_WR_DQS_SHIFT should be programmed to 180(0xB4). This is fine tunable post silicon to accommodate for DQ routing.

- TC300C_dqs_out_shift[8:0] - Controls the delay for the DQS signal for write. Again, the unit of the parameter is degree.

Getting the DQS signal to arrive at a certain point in a clock cycle at the DRAM is a function of the generation of the DQS signal and the physical delays in transmitting this signal from one point to another.

t_flight = t_io_cell + t_driver + t_board.

The write data sent along with the DQS strobe must be aligned such that the strobe rises and falls within the valid region of the data with maximum setup and hold characteristics. This translates into the write data being clocked 1/4 cycle (90 degree) before the rising edge of the DQS strobe.

The values for TC300C_WR_DQS_SHIFT and TC300C_DQS_OUT_SHIFT are determined by the min and max physical delays of t_io_cell, t_driver, and t_board.

The goal is to have the DQS signal arrive at the DRAMs as close to the rising edge of the clock as possible. Once the typical delay values are calculated for these delays, the parameter values are chosen such that the clk_wr signal arrives at the IO cell 1 cycle minus this value.

Recommended Delay Settings

t_flight	clk_wr delay	DQS delay
much less than ¼ cycle	3/4 clk (270 degree)	1 clk (360 degree)
close to ¼ cycle	1/2 clk (180 degree)	3/4 clk (270 degree)
close to ½ cycle	1/4 clk (90 degree)	1/2 clk (180 degree)

Example setting of parameters

Below is an example of the calculations necessary to choose the correct value of TC300C_WR_DQS_SHIFT and TC300C_DQS_OUT_SHIFT:

- Clock frequency = 200MHz (5ns)
- t_io_cell(typ) = 100 ps (+/- 20%)
- t_driver(typ) = 1500 ps (+/- 30%)
- t_board(typ) = 500ps (+/- 10%)
- tdqss = +/- 0.25clk

Typical delay is 2100ps or 0.42clk.
This means that adding an additional 1/2 clock would place the DQS signal within 0.08clk of the rising edge.
So, setting TC300C_DQS_OUT_SHIFT to:

TC300C_DQS_OUT_SHIFT = 180

would accomplish this delay.

To maintain the 1/4 clk phase difference between the write data and write DQS, TC300C_WR_DQS_SHIFT shoud be set to:

TC300C_WR_DQS_SHIFT = 180 - 90 = 90

The min and max variation of the arrival time is +/- 520ps.
This is 0.10 of a clock cycle. This means that the DQS signal would arrive 0.82 to 1.02 of a clock cycle around the clock edge.
This is within the 0.75 to 1.25 specification of the DRAM.

Read Datapath

For the read path, there are 4 delay parameters that need to be programmed:

1.) DDR_CTL_39, TC300C_DLL_DQS_DELAY_0 for byte 0 or bit[7:0]
2.) DDR_CTL_40, TC300C_DLL_DQS_DELAY_1 for byte 1 or bit[15:8]
3.) DDR_CTL_41, TC300C_DLL_DQS_DELAY_2 for byte 2 or bit[23:16]
4.) DDR_CTL_42, TC300C_DLL_DQS_DELAY_3 for byte 3 or bit[31:24]

These parameters are best calculated by software.
First the software should write a 32-bit data to a memory location.
Then the software would form a loop, to sweep the delay from 0 to 360 degree while attempting to read each byte. The optimal parameter for each byte is then calcuated as follow:

TC300C_DLL_DQS_DELAY_x = MIN_DELAY + ((MAX_DELAY - MIN_DELAY) / 2)
where
MIN_DELAY = the smallest delay value in which the byte being read is successful
MAX_DELAY = the biggest delay value in which the byte being read is successful.

15.9. Register Map

Register must be accessed using 32-bit or 64-bit load and store instructions.

Offset Address	Bit Width	Register Symbol	Register Name	Access	Parameter(s)
0x8000	16	DDRCTL00	DDR_CTL_00	-	Reservred
0x8008	16	DDRCTL01	DDR_CTL_01	WR, RD	INT_ACK, INT_STATUS
0x8010	16	DDRCTL02	DDR_CTL_02	RD, RW	OUT_OF_RANGE_LENGTH, INT_MASK
0x8018	16	DDRCTL03	DDR_CTL_03	RW, RD	START, OUT_OF_RANGE_TYPE
0x8020	16	DDRCTL04	DDR_CTL_04	RD, RD	MAX_COL_REG, MAX_ROW_REG
0x8028	16	DDRCTL05	DDR_CTL_05	RW, RD	INITAREF, MAX_CS_REG
0x8030	16	DDRCTL06	DDR_CTL_06	RW, RW	CASLAT, CASLAT_LIN
0x8038	16	DDRCTL07	DDR_CTL_07	RW, RW	TWR, BSTLEN
0x8040	16	DDRCTL08	DDR_CTL_08	RW, RW	TPDEX, TRRD
0x8048	16	DDRCTL09	DDR_CTL_09	RW, RW	TRAS_MIN, TRC
0x8050	16	DDRCTL10	DDR_CTL_10	RW, RW	TEMRS, TRP
0x8058	16	DDRCTL11	DDR_CTL_11	RW, RW	TRFC, TMRD
0x8060	16	DDRCTL12	DDR_CTL_12	RW, RW	TWTR, TDLL
0x8068	16	DDRCTL13	DDR_CTL_13	RW, WR	AREFRESH, WRITEINTERP
0x8070	16	DDRCTL14	DDR_CTL_14	RW, RW	NO_CMD_INIT
0x8078	16	DDRCTL15	DDR_CTL_15	RW, RW	POWER_DOWN, CS_MAP
0x8080	16	DDRCTL16	DDR_CTL_16	RW, RW	SREFRESH, TXSR
0x8088	16	DDRCTL17	DDR_CTL_17	RW, RW	AP, DRIVE_DQ_DQS
0x8090	16	DDRCTL18	DDR_CTL_18	RW, RW	INTRPTREADA, CONCURRENTAP
0x8098	16	DDRCTL19	DDR_CTL_19	RW, RW	INTRPTAPBURST, INTRPTWRITEA
0x80A0	16	DDRCTL20	DDR_CTL_20	RW, RW	TRCD_INT, TRAS_LOCKOUT
0x80A8	16	DDRCTL21	DDR_CTL_21	RW, RW	DLL_BYPASS_MODE, REG_DIMM_ENABLE
0x80B0	16	DDRCTL22	DDR_CTL_22	RW, RW	COLUMN_SIZE, ADDR_PINS
0x80B8	16	DDRCTL23	DDR_CTL_23	RD, RW	DLL_LOCK APREBIT
0x80C0	16	DDRCTL24	DDR_CTL_24	-	Reserved
0x80C8	16	DDRCTL25	DDR_CTL_25	-	Reserved
0x80D0	16	DDRCTL26	DDR_CTL_26	-	Reserved
0x80D8	16	DDRCTL27	DDR_CTL_27	RD	VERSION
0x80E0	16	DDRCTL28	DDR_CTL_28	RW	TREF
0x80E8	16	DDRCTL29	DDR_CTL_29	RW	TRAS_MAX
0x80F0	16	DDRCTL30	DDR_CTL_30	RW	TINIT
0x80F8	16	DDRCTL31	DDR_CTL_31	RW	EMRS_DATA
0x8100	16	DDRCTL32	DDR_CTL_32	RD	OUT_OF_RANGE_ADDR[15:0]
0x8108	16	DDRCTL33	DDR_CTL_33	RD	OUT_OF_RANGE_ADDR[30:16]
0x8110	16	DDRCTL34	DDR_CTL_34	RD	TC300C_DLL_LOCK
0x8118	16	DDRCTL35	DDR_CTL_35	RW	TC300C_WR_DQS_SHIFT
0x8120	16	DDRCTL36	DDR_CTL_36	RW	TC300C_DQS_OUT_SHIFT
0x8128	16	DDRCTL37	DDR_CTL_37	RW	TC300C_WR_DQS_SUB
0x8130	16	DDRCTL38	DDR_CTL_38	RW	TC300C_DQS_OUT_SUB
0x8138	16	DDRCTL39	DDR_CTL_39	RW	TC300C_DLL_DQS_DELAY_0
0x8140	16	DDRCTL40	DDR_CTL_40	RW	TC300C_DLL_DQS_DELAY_1
0x8148	16	DDRCTL41	DDR_CTL_41	RW	TC300C_DLL_DQS_DELAY_2
0x8150	16	DDRCTL42	DDR_CTL_42	RW	TC300C_DLL_DQS_DELAY_3
0x8158	16	DDRCTL43	DDR_CTL_43	RW	TC300C_DLL_DQS_SUB_0
0x8160	16	DDRCTL44	DDR_CTL_44	RW	TC300C_DLL_DQS_SUB_1
0x8168	16	DDRCTL45	DDR_CTL_45	RW	TC300C_DLL_DQS_SUB_2
0x8170	16	DDRCTL46	DDR_CTL_46	RW	TC300C_DLL_DQS_SUB_3

15.10. Registers

Note: Reserve bits will return 0s when read.

DDR_CTL_00 = 0x8000

Name	Bits	Default	Range	Description
-	5:00	-	-	Reserved
-	7:6	-	-	Reserved
-	13:08	-	-	Reserved
-	15:14	-	-	Reserved

DDR_CTL_01 = 0x8008

Name	Bits	Default	Range	Description
INT_STATUS	3:0	0x0	0x0-0xf	Status of interrupt features in the controller. Status of all possible interrupts generated by the controller. The MSB of this register is the OR of all the lower bits. The Int_status bits correspond to these interrupts. 0 = Single out of range address detected 1 = Multiple out of range address detected 2 = DRAM initialization complete 3 = Logical OR of all lower bits READONLY
-	7:4	-	-	Reserved
INT_ACK	10:8	0x0	0x0-0x7	Clear mask of INT_STATUS register. This register controls clearing the Int_status register. If any of these bits are set to a one when writing this register, the corresponding bit in the Int_status register will be set to 0. Any bits written width a 0 will not alter the bit in the Int_status register. WRITEONLY
-	15:11	-	-	Reserved

DDR_CTL_02 = 0x8010

Name	Bits	Default	Range	Description
INT_MASK	3:0	0x0	0x0-0x7	Mask for controller_int signals from the INT_STATUS register. The mask applied to the outputs Int_status register that are logically ORed and reflected to the controller_int pin on the ASIC interface.
-	7:3	-	-	Reserved
OUT_OF_RANGE_LENGTH	15:8	0x00	0x0-0xff	Length of command that caused Out of Range interrupt. The user length for an out of range request to the memory devices. READONLY

DDR_CTL_03 = 0x8018

Name	Bits	Default	Range	Description
OUT_OF_RANGE_TYPE	2:0	0x0	0x0-0x7	Type of command that caused Out of Range interrupt. READONLY
-	7:3	-	-	Reserved
START	8:8	0x0	0x0-0x1	Start bit to control command processing in the controller Start register. With this register set to 0, the controller will not issue any commands to the DRAMs or respond to any pin activity except for reading and writing registers. Once this register is set to 1 the controller will then respond to inputs from the ASIC.
-	15:9	-	-	Reserved

DDR_CTL_04 = 0x8020

Name	Bits	Default	Range	Description
MAX_ROW_REG	3:0	0xe	0xe	Maximum width of memory address bus. Maximum number of row bits configured for the controller. This value can be used to set the Addr_pins. Addr_pins = Max_row_reg - <number of row bits in memory device> READONLY
-	7:4	-	-	Reserved
MAX_COL_REG	11:8	0xc	0xc	Maximum width of column address in DRAMs. Maximum number of column bits configured for the controller. This value can be used to set the Column_size. Column_size = Max_col_reg - <number of column bits in memory device> READONLY
-	15:12	-	-	Reserved

DDR_CTL_05 = 0x8028

Name	Bits	Default	Range	Description
MAX_CS_REG	1:0	0x2	0x2	Maximum number of chip selects available. Maximum number of chip selects configured for the controller. This value can be used to set the CS_diff. CS_diff = Max_cs_reg - <number of ranks of memory connected to the controller> The number of ranks of memory must be a power of 2. READONLY
-	7:2	-	-	Reserved
INITAREF	11:8	0x0	0x0-0xf	Number of autorefresh commands to execute during initialization of DRAMS. The number of auto-refresh commands needed by the DRAM devices to satisfy the initialization sequence.
-	15:12	-	-	Reserved

DDR_CTL_06 = 0x8030

Name	Bits	Default	Range	Description
CASLAT_LIN	3:0	0x0	0x0-0xf	Cas latency in half cycles. Cas Latency linear value. This field encodes the actual cas latency in ½ cycle increments. 0000 - 0010 RESERVED 0011 = 1.5 cycles 0100 = 2 cycles 0101 = 2.5 cycles 0110 = 3 cycles 0111 = 3.5 cycles 1000 = 4 cycles 1001-1111 RESERVED
-	7:4	-	-	Reserved
CASLAT	10:8	0x0	0x0-0x7	Encoded cas latency sent to DRAMs during initialization. Cas Latency encoding to be programmed into the DRAMs upon initialization. Please refer to the encoding specified in the DRAM spec sheet. This encoding should correspond to the Caslat_Linear setting. See the regconfig files in the release for actual settings for each particular device.
-	15:11	-	-	Reserved

DDR_CTL_07 = 0x8038

Name	Bits	Default	Range	Description
BSTLEN	2:0	0x0	0x0-0x7	Encoded burst length sent to DRAMs during initialization. Burst length encoding that will be programmed into the DRAMs upon initialization. Encoding is as follows: 2 words = 001 4 words = 010 8 words = 011 All other codes are reserved
-	7:3	-	-	Reserved
TWR	10:8	0x0	0x0-0x7	DRAM TWR parameter in cycles. Write Recovery time in cycles
-	15:11	-	-	Reserved

DDR_CTL_08 = 0x8040

Name	Bits	Default	Range	Description
TRRD	2:0	0x0	0x0-0x7	DRAM TRRD parameter in cycles. Activate to Activate delay for different banks in cycles
-	7:3	-	-	Reserved
TPDEX	10:8	0x0	0x0-0x7	DRAM TPDEX parameter in cycles. Power-down exit command period in cycles.
-	15:10	-	-	Reserved

DDR_CTL_09 = 0x8048

Name	Bits	Default	Range	Description
TRC	4:0	0x00	0x0-0x1f	DRAM TRC parameter in cycles. Active to Active command period for the same bank in cycles
-	7:5	-	-	Reserved
TRAS_MIN	11:8	0x0	0x0-0xf	DRAM TRAS_MIN parameter in cycles. Minimum Row activate time in cycles
-	15:12	-	-	Reserved

DDR_CTL_10 = 0x8050

Name	Bits	Default	Range	Description
TRP	3:0	0x0	0x0-0xf	DRAM TWR parameter in cycles. Precharge command time in cycles
-	7:4	-	-	Reserved
TEMRS	9:8	0x0	0x0-0x3	DRAM TERMS parameter in cycles. Extended mode register set time in cycles
-	15:10	-	-	Reserved

DDR_CTL_11 = 0x8058

Name	Bits	Default	Range	Description
TMRD	1:0	0x0	0x0-0x3	DRAM TMRD parameter in cycles. Mode register set command time in cycles
-	7:2	-	-	Reserved
TRFC	12:8	0x00	0x0-0x1f	DRAM TRFC parameter in cycles. Refresh command time in cycles
-	15:13	-	-	Reserved

DDR_CTL_12 = 0x8060

Name	Bits	Default	Range	Description
TDLL	7:0	0x00	0x0-0xff	DRAM TDLL parameter in cycles. Dll lock time for DDR SDRAM in cycles
TWTR	10:8	0x0	0x0-0x7	DRAM TWTR parameter in cycles. The number of cycles needed to switch from a write to a read operation dictated by the DDR SDRAM specification.
-	15:11	-	-	Reserved

DDR_CTL_13 = 0x8068

Name	Bits	Default	Range	Description
WRITEINTERP	0:0	0x0	0x0-0x1	Enable DRAM feature that allows controller to interrupt write bursts to the DRAMs. Read interrupt write parameter. When enabled the controller is able to interrupt a write burst with a read command. Some memory device do not allow this functionality. 0 = The device does not support read commands interrupting write commands. 1 = The device does support read commands interrupting write commands.
-	7:1	-	-	Reserved
AREFRESH	8:8	0x0	0x0-0x1	Execute an autorefresh command immediately or at the next DRAM burst boundary. When this register is written with a “1”, and auto precharge command will be issued to the DRAM devices. If there are any open banks, the controller will automatically close these banks before issuing the auto refresh command. This register will always read back “0”. WRITEONLY
-	15:9	-	-	Reserved

DDR_CTL_14 = 0x8070

Name	Bits	Default	Range	Description
NO_CMD_INIT	0:0	0x0	0x0-0x1	Enable disabling DRAM commands until TDLL has expired during initialization. 0 = Issue only REF and PRE commands during DLL initialization of the DRAMs 1 = Do not issue any type command during DLL initialization of the DRAMs
-	7:1	-	-	Reserved
-	15:8	-	-	Reserved

DDR_CTL_15 = 0x8078

Name	Bits	Default	Range	Description
CS_MAP	1:0	0x0	0x0-0x3	Number if active chip selects used in address decoding. This register is programmed with a mask that determines which chip select pins are active. The user address CS field will be mapped into the active chip selects indicated by this register in ascending order from lowest to highest. This allows the controller to map the entire contiguous user address into any group of chip selects. Bit 0 of this register corresponds to chip select[0]. 2'b01 = chip select[0] is enabled 2'b10 = chip select[1] is enabled 2'b11 = Both chip select[0] and chip select[1] are enabled
-	7:2	-	-	Reserved
POWER_DOWN	8:8	0x0	0x0-0x1	Place DRAMs in power down state. When this register is written with a "1", the controller will disable the clock enable pin to the DRAM devices as soon as the current burst for the current transaction, if any, the controller is processing, has completed. Any subsequent commands in the command queue will be suspended until this register is written with a "0". The controller will continue to issue refresh commands and any precharge commands necessary on open banks while in power down mode. This means that the DRAM will automatically be brought out of power down mode for these operations.
-	15:9	-	-	Reserved

DDR_CTL_16 = 0x8080

Name	Bits	Default	Range	Description
TXSR	7:0	0x00	0x0-0xff	DRAM TXSR parameter in cycles. Self refresh exit time in cycles
SREFRESH	8:8	0x0	0x0-0x1	Place DRAMs in self-refresh mode. When this register is written with a "1", the DRAM device(s) will be placed in self-refresh mode. In order to put the DRAMs in self refresh mode, the current burst for the current transaction, if any, will complete, all banks will be closed, the self refresh command will be issued to the DRAM, and the clock enable pin will be de-asserted. The system will remain in self-refresh mode until this register is written with a "0". The DRAM devices will return to normal operating mode after the self-refresh exit time (txsr) of the device and any dll initialization time for the DRAM is reached. The controller will resume processing of commands where it was interrupted if it were processing a transaction when the register was written.
-	15:9	-	-	Reserved

DDR_CTL_17 = 0x8088

Name	Bits	Default	Range	Description
DRIVE_DQ_DQS	0:0	0x0	0x0-0x1	Enable feature to keep output enables on dq and dqs drives active when controller is idle. This parameter selects if the dq and dqs output enables will driven active when the controller is in an idle state. 0 - Leave the output enables in High-Z when idle 1 - Drive the output enables active when idle.
-	7:1	-	-	Reserved
AP	8:8	0x0	0x0-0x1	Enable autorefresh mode of controller. Enable Auto Precharge mode for DRAM devices. 0 = Auto precharge mode disabled. All read and write transactions are issued such that the banks stay open until another request needs the bank or the max open time (tras_max) has expired or a refresh command closes all banks. 1 = Auto precharge mode enabled. All read and write transactions are terminated by an auto precharge command. If a transaction consists of multiple read or write commands, only the last command is issued with an auto precharge.
-	15:9	-	-	Reserved

DDR_CTL_18 = 0x8090

Name	Bits	Default	Range	Description
CONCURRENTAP	0:0	0x0	0x0-0x1	Allow controller to issue commands to other banks while a bank is in the process of autorefresh. Enable Concurrent Auto precharge. Some DRAM devices do not allow one bank to be auto pre charged while another bank is reading or writing. The JEDEC standard allows concurrent auto precharge. 0 = Concurrent Auto precharge disabled. 1 = Concurrent Auto precharge enabled.
-	7:1	-	-	Reserved
INTRPTREADA	8:8	0x0	0x0-0x1	Allow the controller to interrupt a read with autoprecharge command with another read command. Enable interrupting of a read with auto precharge command with another read command to the same bank before the first read command is completed. 0 = Disable interrupting a read with auto precharge with another read command to the same bank. 1 = Enable interrupting a read with auto precharge with another read command to the same bank.
-	15:9	-	-	Reserved

DDR_CTL_19 = 0x8098

Name	Bits	Default	Range	Description
INTRPTWRITEA	0:0	0x0	0x0-0x1	Allow the controller to interrupt a write command with autoprecharge with another write command. Enable interrupting of a write with auto precharge command with another read or write command to the same bank before the first write command is completed. 0 = Disable interrupting a write with auto precharge with another read or write command to the same bank. 1 = Enable interrupting a write with auto precharge with another read or write command to the same bank.
-	7:1	-	-	Reserved
INTRPTAPBURST	8:8	0x0	0x0-0x1	Allow the controller to interrupt a command with autoprecharge with another command. Enable interrupting an auto precharge command with another command for a different bank. If enabled the current operation will be interrupted but the bank will be pre charged as if the current operation were allowed to continue. 0 = Disable interrupting an auto precharge operation on a different bank. 1 = Enable interrupting an auto precharge operation on a different bank.
-	15:9	-	-	Reserved

DDR_CTL_20 = 0x80A0

Name	Bits	Default	Range	Description
TRAS_LOCKOUT	0:0	0x0	0x0-0x1	Allow the controller to execute autoprecharge commands before the TRAS_MIN parameter has expired. Tras lockout setting for the DRAM device. 0 = Tras lockout not supported by memory device 1 = Tras lockout supported by memory device
-	7:1	-	-	Reserved
TRCD_INT	11:8	0x0	0x0-0xf	DRAM TRCD parameter in cycles. Ras to Cas delay in cycles
-	15:11	-	-	Reserved

DDR_CTL_21 = 0x80A8

Name	Bits	Default	Range	Description
REG_DIMM_ENABLE	0:0	0x0	0x0-0x1	Enable registered DIMM operation of the controller. This register controls the address and command pipeline of the controller when configured for register DIMM operations. 0 = Normal non registered operation. 1 = Enable register DIMM operation.
-	7:1	-	-	Reserved
DLL_BYPASS_MODE	8:8	0x0	0x0-0x1	Enable the DLL bypass feature of the controller. When set to 1, the values programmed into dll_dqs_delay_x, dqs_out_shift, and wr_dqs_shift become absolute values rather than fractional values of delays in the delay chains. In this mode the DLL locking mechanism is bypassed. If the delay programmed into the delay registers is greater than the number of delay elements in the delay chain, then the delay is set to the maximum number of delay elements in the delay chain. 0 = Normal operational mode 1 = Bypass DLL master delay line.
-	15:9	-	-	Reserved

DDR_CTL_22 = 0x80B0

Name	Bits	Default	Range	Description
ADDR_PINS	2:0	0x0	0x0-0x7	Number of address pins used by DRAM devices. The difference between the maximum number of address pins (14) and the actual number of address pins on the device connected to the controller. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based upon the value of this parameter. For more detail, see Section 15.13.
-	7:3	-	-	Reserved
COLUMN_SIZE	10:8	0x0	0x0-0x7	Number of column bits used for DRAM devices. The difference between the maximum column width (12) and the actual number of column pins in the devices connected to the controller. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based upon the value of this parameter. For more detail see Section 15.13..
-	15:11	-	-	Reserved

DDR_CTL_23 = 0x80B8

Name	Bits	Default	Range	Description
APREBIT	3:0	0x0	0x0-0xf	Location of autoprecharge bit in DRAM address. The bit position for the auto precharge signal for the DDR SDRAM device connected to the controller.
-	7:4	-	-	Reserved
-	13:8	-	-	Reserved
-	15:14	-	-	Reserved

DDR_CTL_24 = 0x80C0

Name	Bits	Default	Range	Description
-	6:0	-	-	Reserved
-	7	-	-	Reserved
-	14:8	-	-	Reserved
-	15	-	-	Reserved

DDR_CTL_25 = 0x80C8

Name	Bits	Default	Range	Description
-	6:0	-	-	Reserved
-	7	-	-	Reserved
-	14:8	-	-	Reserved
-	15	-	-	Reserved

DDR_CTL_26 = 0x80D0

Name	Bits	Default	Range	Description
-	6:0	-	-	Reserved
-	15:7	-	-	Reserved

DDR_CTL_27 = 0x80D8

Name	Bits	Default	Range	Description
VERSION	15:0	0x2031	0x2031	Controller version number DDR version number of the controller. READONLY

DDR_CTL_28 = 0x80E0

Name	Bits	Default	Range	Description
TREF	11:0	0x000	0x0-0xfff	DRAM TREF parameter in cycles. Cycles between refresh commands.
-	15:12	-	-	Reserved

DDR_CTL_29 = 0x80E8

Name	Bits	Default	Range	Description
TRAS_MAX	15:0	0x0000	0x0-0xffff	DRAM TRAS_MAX parameter in cycles. Maximum Row active time in cycles

DDR_CTL_30 = 0x80F0

Name	Bits	Default	Range	Description
TINIT	15:0	0x0000	0x0-0xffff	DRAM TINIT parameter in cycles. Initialization time for DDR SDRAM in cycles

DDR_CTL_31 = 0x80F8

Name	Bits	Default	Range	Description
EMRS_DATA	13:0	0x0000	0x0-0x3fff	Extended mode register data written during initialization or Write modereg command. The contents of this register will be loaded into the DRAM during initialization. Consult the DRAM specification for the correct settings for this register.
-	15:14	-	-	Reserved

DDR_CTL_32 = 0x8100

Name	Bits	Default	Range	Description
OUT_OF_RANGE_ADDR[15:0]	15:0	0x0000	0x0-0xffff	Address from command that caused an Out of Range interrupt. This is the lower 16 bits of the user address for an out of range request to the memory devices. READONLY

DDR_CTL_33 = 0x8108

Name	Bits	Default	Range	Description
OUT_OF_RANGE_ADDR[30:16]	14:0	0x0000	0x0-0x7fff	Address from command that caused an Out of Range interrupt. This is the upper 15 bits of the user address for an out of range request to the memory devices. READONLY
-	15	-	-	Reserved

DDR_CTL_34 = 0x8110

Name	Bits	Default	Range	Description
TC300C_DLL_LOCK	8:0	0x00	0x0-0x1ff	Number of delay elements of TC300C master DLL lock. The actual number of delay elements of the TC300C macro delay line used to capture one full clock cycle. This parameter is automatically updated every 8 clock cycles. READONLY
-	15:9	-	-	Reserved

DDR_CTL_35 = 0x8118

Name	Bits	Default	Range	Description
TC300C_WR_DQS_SHIFT	8:0	0x00	0x0-0x1ff	Fraction of a cycle to delay the clk_wr signal in the controller. This parameter controls the amount of delay introduced to the write datapath to ensure the correct capture of data at the DRAM interface. It selects the number of 1/360ths of the system clock to delay the write dq data by to the DRAM.
-	15:9	-	-	Reserved

DDR_CTL_36 = 0x8120

Name	Bits	Default	Range	Description
TC300C_DQS_OUT_SHIFT	8:0	0x00	0x0-0x1ff	Fraction of a cycle to delay the write dqs signal to the DRAMs during writes. This parameter controls the amount of delay introduced to the write dqs signal to ensure the correct capture of data at the DRAM interface. It selects the number of 1/360ths of the system clock to delay the write dqs by to the DRAM.
-	15:9	-	-	Reserved

DDR_CTL_37 = 0x8128

Name	Bits	Default	Range	Description
TC300C_WR_DQS_SUB	4:0	0x00	0x0-0x1f	Offset delay subtracted from the TC300C_WR_DQS_SHIFT delay of the clk_wr signal in the controller. The number of delay element subtracted from the number calculated from the TC300C_WR_DQS_SHIFT parameter. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

DDR_CTL_38 = 0x8130

Name	Bits	Default	Range	Description
TC300C_DQS_OUT_SUB	4:0	0x00	0x0-0x1f	Offset delay subtracted from the TC300C_DQS_OUT_SHIFT delay of the write dqs signal to the DRAMs during writes. The number of delay element subtracted from the number calculated from the TC300C_DQS_OUT_SHIFT parameters. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

DDR_CTL_39 = 0x8138

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_DELAY_0	8:0	0x00	0x0-0x1ff	Fraction of a clock cycle to delay the dqs signal from the DRAMs during reads. The number of 1/360ths of the system clock to delay the dqs signals from the DDR SDRAM devices to center the edges of the dqs signal to capture the read data in the middle of the valid window. This parameter controls the amount of delay to introduce to the dqs path for dq[7:0] of the read data.
-	15:9	-	-	Reserved

DDR_CTL_40 = 0x8140

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_DELAY_1	8:0	0x00	0x0-0x1ff	Fraction of a clock cycle to delay the dqs signal from the DRAMs during reads. The number of 1/360ths of the system clock to delay the dqs signals from the DDR SDRAM devices to center the edges of the dqs signal to capture the read data in the middle of the valid window. This parameter controls the amount of delay to introduce to the dqs path for dq[15:8] of the read data.
-	15:9	-	-	Reserved

DDR_CTL_41 = 0x8148

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_DELAY_2	8:0	0x00	0x0-0x1ff	Fraction of a clock cycle to delay the dqs signal from the DRAMs during reads. The number of 1/360ths of the system clock to delay the dqs signals from the DDR SDRAM devices to center the edges of the dqs signal to capture the read data in the middle of the valid window. This parameter controls the amount of delay to introduce to the dqs path for dq[23:16] of the read data.
-	15:9	-	-	Reserved

DDR_CTL_42 = 0x8150

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_DELAY_3	8:0	0x00	0x0-0x1ff	Fraction of a clock cycle to delay the dqs signal from the DRAMs during reads. The number of 1/360ths of the system clock to delay the dqs signals from the DDR SDRAM devices to center the edges of the dqs signal to capture the read data in the middle of the valid window. This parameter controls the amount of delay to introduce to the dqs path for dq[31:24] of the read data.
-	15:9	-	-	Reserved

DDR_CTL_43 = 0x8158

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_SUB_0	4:0	0x00	0x0-0x1f	Delay offset subtracted from TC300C_DLL_DQS_DELAY_0 delay of the dqs signal from the DRAMs during reads. The number of delay element subtracted from the number calculated from the TC300C_DLL_DQS_DELAY_0 parameter. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

DDR_CTL_44 = 0x8160

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_SUB_1	4:0	0x00	0x0-0x1f	Delay offset subtracted from TC300C_DLL_DQS_DELAY_1 delay of the dqs signal from the DRAMs during reads. The number of delay element subtracted from the number calculated from the TC300C_DLL_DQS_DELAY_1 parameter. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

DDR_CTL_45 = 0x8168

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_SUB_2	4:0	0x00	0x0-0x1f	Delay offset subtracted from TC300C_DLL_DQS_DELAY_2 delay of the dqs signal from the DRAMs during reads. The number of delay element subtracted from the number calculated from the TC300C_DLL_DQS_DELAY_2 parameter. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

DDR_CTL_46 = 0x8170

Name	Bits	Default	Range	Description
TC300C_DLL_DQS_SUB_3	4:0	0x00	0x0-0x1f	Delay offset subtracted from TC300C_DLL_DQS_DELAY_3 delay of the dqs signal from the DRAMs during reads. The number of delay element subtracted from the number calculated from the TC300C_DLL_DQS_DELAY_3 parameter. This parameter allows finer control of the exact number of delay element to be used.
-	15:5	-	-	Reserved

Following de-assertion of the reset pin, the registers in the controller must be configured before the DRAM devices can be accessed.

15.11. Read Data Capture

The read data capture logic is responsible for capturing the dq outputs from the DRAM devices with the delayed version of dqs strobe signals from the DRAM devices and then passing the data back to the system clock domain. The dqs strobes are first sent through the delay lines to ensure that the rising and falling edges of the strobes are in the middle of the valid window of data.

DDR DRAM devices send a dqs strobe signal coincident with the read data so that the read data can be reliably captured by a controller. The edges of this strobe are aligned with the data by the DRAM device. Both the rising and falling edges of this strobe are used to capture data. It is assumed that the wires for the data and the associated dqs signals are routed with the same length between the capture logic and the DRAM device. This will cause the rising and falling edges of the dqs strobe to arrive at the capture logic at the same time the data is in transition. The raw dqs signal cannot be used as a clock to a flip-flop to register the data since the data is not stable at the time the dqs edges are rising and falling.

A delayed version of the dqs signal must be used to capture the data. The amount of delay added to the dqs signals should be calculated such that it gives the maximum margin to capture the read data. This calculated value is best described as a relative number based on the period of the system clock since the frequency of the dqs signal is matched to the system clock by the DRAMs. In the case shown in Figure 15-10 below, the delay is set to approximately 25% of the system clock. The DLL will keep this relative delay constant so the read data from the DRAMs can be reliably captured.

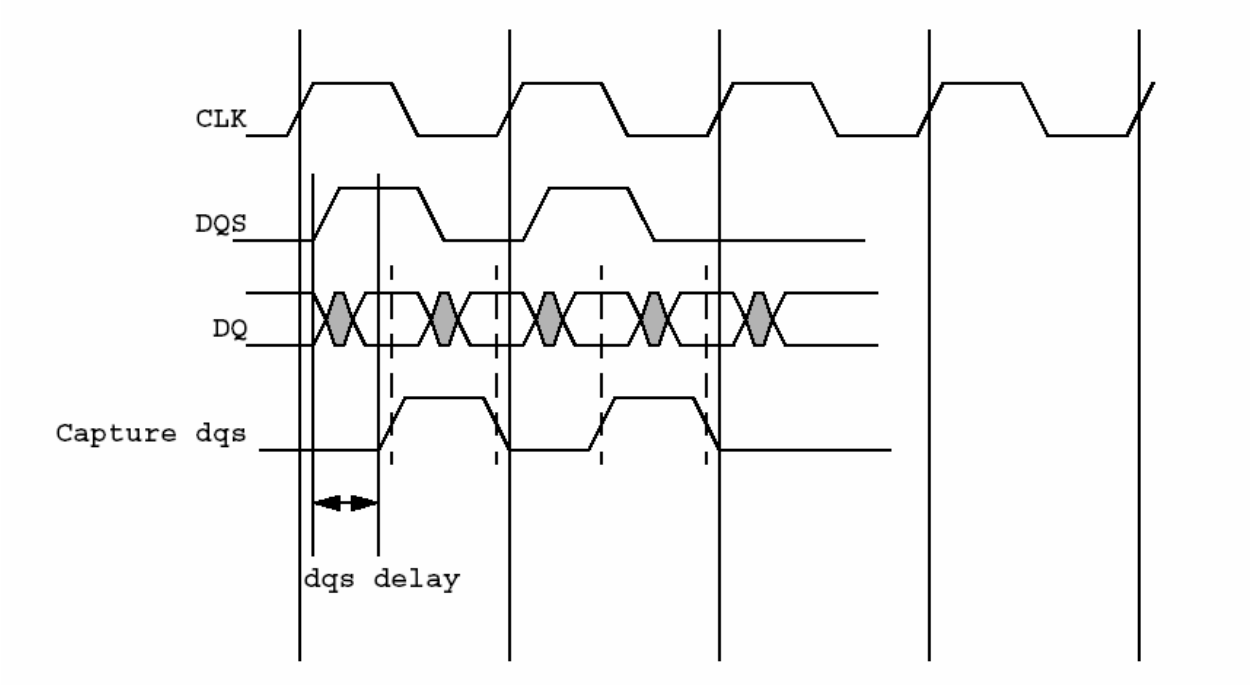


Figure 15-10 dqs Read Timing

The amount of delay introduced to the read dqs signal from DRAM is controlled by the TC300C_dll_dqs_delay_*x* and TC300C_dll_dqs_sub_*x* parameters. Where *x* can have a value of 0, 1, 2 and 3 corresponding to each of the 4 DQ bytes.

Once the data from the DRAMs are registered in with the delayed dqs signal, the output of these registers must be transferred to registers that are clocked by the main controller clock.

15.12. Write Data Timing

Some DRAM devices require that the dqs data strobe arrive at the DRAM devices within a window around the clock that the DRAM receives. Figure 15-11 below describes this relationship. The value for tdqss is specified in fractions of the clock cycle. Most DRAMs specify these values between +/- 0.25 and 0.2 of a clock cycle. This translates to a valid window of between 0.4 and 0.5 of a clock cycle.

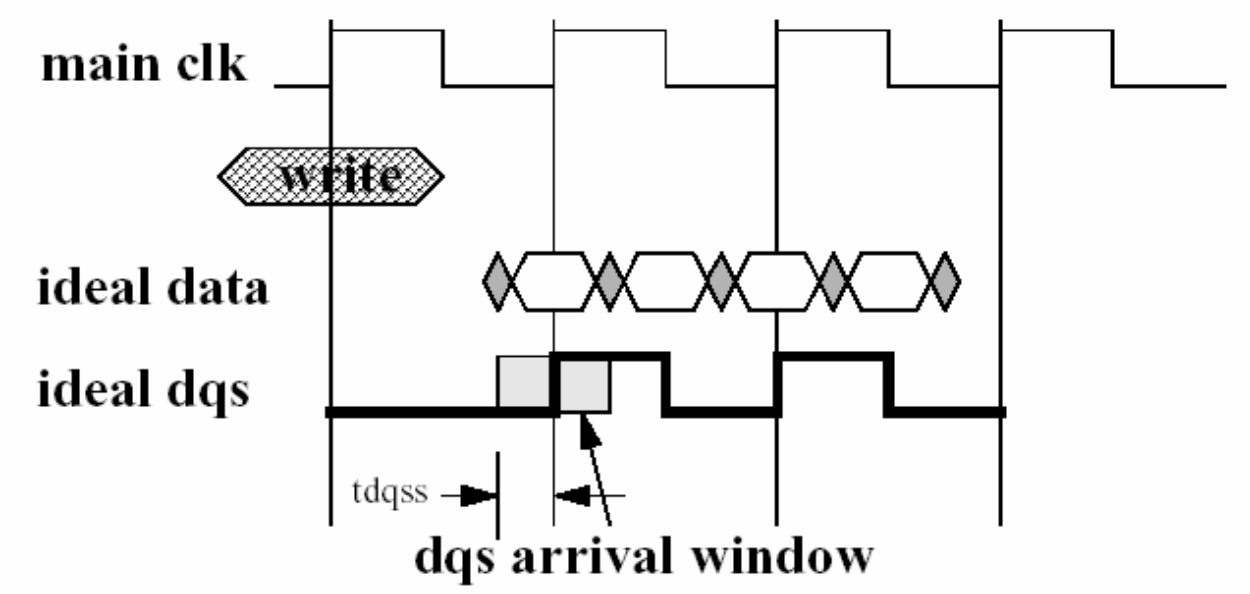


Figure 15-11 dqs Arrival Window

The data transfer timing from the controller to the DRAM for writes is similar to the read transfer from the DRAMs to the controller. There are two differences. The first is that the DRAM devices expect the dqs signal to be shifted by the controller to allow the DRAM the maximum margin for capturing the data with the dqs signal sent to the DRAMs from the controller. The second condition is that the first rising edge of the dqs signal sent from the controller must occur near the rising edge of the clk at the DRAM. This is called the arrival window. DRAMs typically specify this window as 0.8clk to 1.2clk. See Figure 15-12 below for details.

The DLL maintains two delay lines to accomplish the sending of write data and the write dqs strobe:

The first delay line delays the main clock by the amount needed to have the write dqs strobe transition as near to the clock edge at the DRAM as possible under typical operating conditions. The amount of delay is controlled by the TC300C_dqs_out_shift and TC300C_dqs_out_sub parameters.

The second delay line adjusts the clock that is used to output the write data. This clock should be adjusted to maximize the setup and hold requirements around the dqs write strobe. The amount of delay is controlled by the TC300C_wr_dqs_shift and TC300C_wr_dqs_sub parameters.

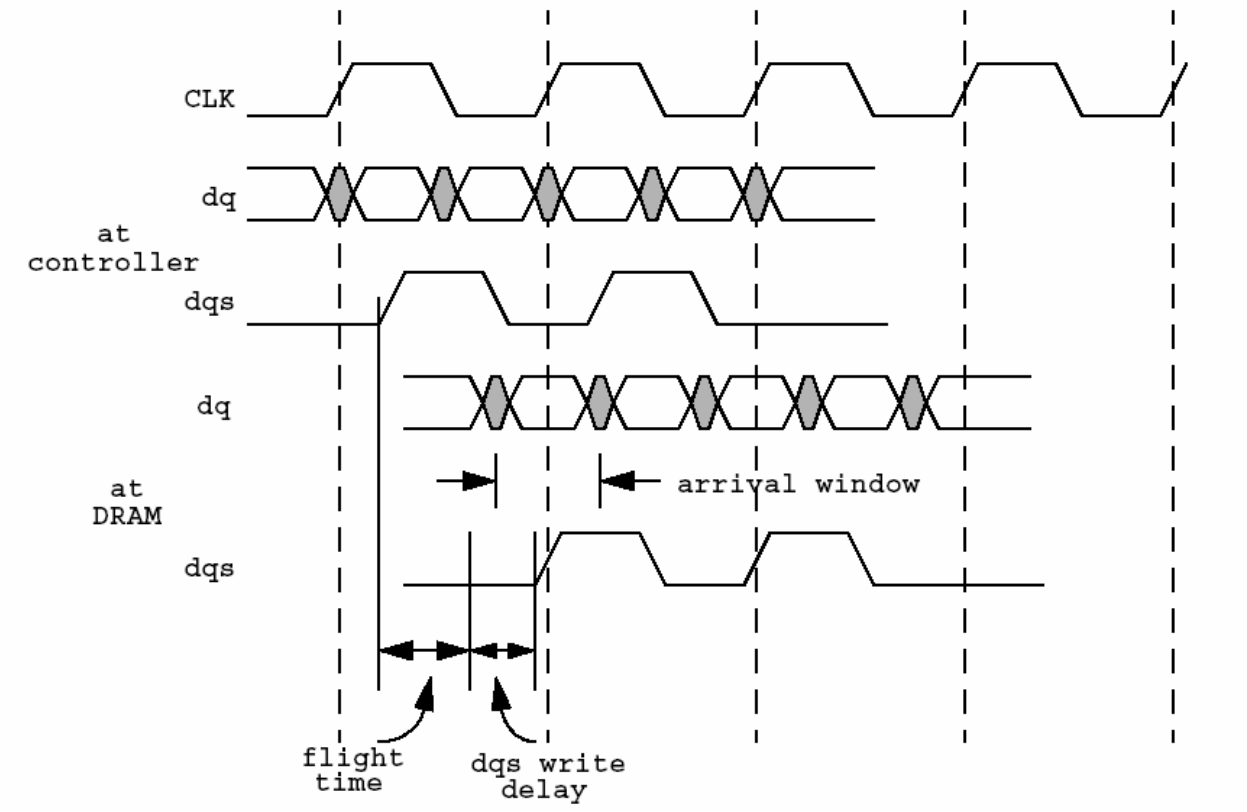


Figure 15-12 dqs Write Timing

15.13. DDR Controller Address Mapping

The DDR Databahn memory controller automatically maps the incoming user address to the available DRAM memory connected to the controller in a contiguous block starting at user address 0 and ending at the highest available address according to the size and number of DRAM devices connected to the controller. This mapping is determined by the pre-chosen address mapping of the Chip Select, Row, Bank, Column signals plus setting certain fields in the internal Databahn registers.

In this controller the address map is ordered as follow:

- *Chip Select, Row, Bank, Column, Datapath*

The Bank, Chip Select, and Datapath widths are fixed while the widths of the other two portions of the memory map are programmable using the device address width registers in the controller. This width can be reduced from the maximum by selecting values for the register settings that are less than the maximum configured.

For this controller, the bit width of Chip Select, Row, Bank, Column and Datapath are as follow:

- *Chip Select = 1*
- *Row = 14*
- *Bank = 2*
- *Column = 12*
- *Datapath = 2*

The maximum allowable address space and mapping into the DRAM devices for the controller is shown in Figure 15-13 below. The numbers on top of the map correspond to the address posted to the controller. This map corresponds to a memory device with 14 row address bits and 12 column bits.

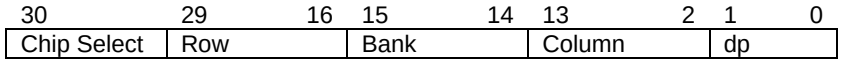


Figure 15-13 Controller Memory Map: Maximum

The DDR Databahn memory processor can support memories with address pins widths and column size widths of up to 7 bits less each than the maximum configured. This allows the DDR Databahn memory processor to function with a wide variety of memory device sizes.

The register settings for addr_pins and column_size control the address map used to decode the user address to the DRAM chip selects and row and column addresses. It is assumed that the values in these registers never exceed the maximum values configured. From the above example, if the memory processor is wired to devices with 12 address pins and 10 column bits, the maximum accessible memory space would be 128MB, and the address map is shown in Figure 15-14, “Alternate memory map,” below. Note that address bits 30 through 27 are not used. These bits are ignored when generating the address to the DRAMs.

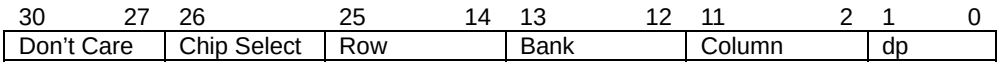


Figure 15-14 Alternate Memory Map

15.14. DDR Controller Interrupt

An interrupt will be generated when the physical address is out of bounds and the interrupt is unmasked. The external interrupt will assert when one or both of the following conditions occur:

- *address out of bounds (For starting address and/or (starting address+length) out of bounds)*
- *address out of bounds over-run (For starting address and/or (starting address+length) out of bounds)*

An address out of bounds occurs when an out of bounds address is detected, and the status register doesn't indicate that a previous out of bounds address has been detected and not acknowledged. If another out of bounds address is detected before the first address out of bounds is acknowledged, then an address out of bounds over-run will occur. Each of these conditions (interrupt sources) will have a unique mask bit and acknowledge bit in the interrupt mask and interrupt acknowledge registers. The user can enable none, 1, or both of the conditions (sources) depending on their system requirements.

Table 15-4 below describes the interrupt control and status parameters in more details.

Table 15-4 Interrupt Parameter Definition

Parameter Name	Description	Bit 0	Bit 1	Bit 2
int_status	The first 2 bits describe the interrupt source, and the 3rd bit is asserted when any unmasked interrupt occurs.	Single address out of range error occurred	Multiple address out of range errors occurred	(bit0 | bit 1) Register indicates that an interrupt has occurred
int_mask	The interrupt mask register contains mask bit for each interrupt source. When the corresponding mask bit is set to 1, the interrupt is masked (i.e. disabled).	Mask single out of range interrupt	Mask multiple out of range interrupt	Inhibit interrupt pin. Registers still function the same , but interrupt pin is never asserted. For polling systems
int_ack	Similar to the interrupt mask register, the interrupt acknowledge register contains acknowledge bits for each interrupt source. The user can clear a single interrupt or multiple interrupts with one write.	Acknowledge single address out of range	Acknowledge multiple out of range interrup	None

15

Out of Range Address, Length and Command Type Registers (out_of_range_addr, out_of_range_length, out_of_range_type)
These register contain the erroneous address, length and the command type. When multiple interrupts occur before the initial interrupt is acknowledged, only the initial out of range address and length are saved. Therefore, the 2nd –N th address and length pairs are lost if the 1st interrupt hasn't been acknowledged/cleared.

Steps in interrupt generation / acknowledge
When an out of bounds access is detected, the following actions occur:
1) Out of bounds interrupt signal is asserted.
2) The command will enter into the command queue and will execute as normal.
3) The status register will update to indicate an out of range access occurred.
4) The out of range address, length, and command type is saved in the controller register space.
5) The interrupt is cleared when the user writes to the interrupt acknowledge register.
To check for an out of bounds address, the starting address is checked, and the (starting address + length) is checked. Therefore, even if the instruction starts inbounds, if the length of the transfer puts the access out of bounds, an exception will occur.

Chapter 16. PCI Controller

16.1. Features

The TX4939 PCI Controller functions as a bus bridge between the TX4939 External PCI (or internal PCI) and the internal bus (G-Bus).

16.1.1. Overall

- *Compliant to "PCI Local Bus Specification Revision 2.2"*
- *PCI Bus: 32-bit data bus; Internal Bus: 64-bit data bus*
- *Maximum PCI bus clock operating frequency: 66 MHz*
- *Dual address cycle support (40-bit PCI address space)*
- *Supports both the Initiator and Target functions*
- *Supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1.*
- *On-chip PCI Bus Arbiter, can connect to a maximum of six (6) external bus masters*
- *1-channel on-chip DMA Controller (PDMAC) dedicated to the PCI Controller*
- *Supports PCI clock output "enable/disable" mode*
- *The Internal Bus clock and PCI Bus clock are asynchronous and can be set independently*
- *Mounted a retry function on the Internal Bus side also in order to avoid deadlock on the PCI Bus.*

16.1.2. Initiator Function

- *Single and Burst transfer from the Internal Bus to the PCI Bus*
- *Supports memory, I/O, configuration, special cycle, and interrupt acknowledge transactions.*
- *Address mapping between the Internal Bus and the PCI Bus can be modified*
- *Mounted 16-stage 64-bit data one FIFO each for Read and Write*
- *Post Write function enables quick termination of a maximum of four Write transactions by the G-Bus without waiting for completion on the PCI Bus.*
- *Endian switching function*

16.1.3. Target Function

- *Single and Burst transfer from the PCI Bus to the Internal Bus*
- *Supports memory, I/O, and configuration cycles*
- *Supports high-speed back-to-back transactions on the PCI Bus*
- *Address mapping between the PCI Bus and the Internal bus can be modified*
- *Mounted 16-stage 64-bit data FIFO for Read*
- *Mounted 20-stage 64-bit data FIFO for Write*
- *Post Write function enables quick termination of a maximum of five (5) Write transactions by the PCI Bus without waiting for completion on the G-Bus.*
- *Read Burst length (pre-fetch data size) on the Internal Bus when reading a pre-fetchable space can be made programmable*
- *Endian switching function*

16.1.4. PCI Arbiter

- *Supports six (6) external PCI bus masters*
- *Uses the Programmable Fairness algorithm (two levels with different priorities for six (6) external and one internal round-robin request/grant pairs)*
- *Supports bus parking*
- *Bus master uses the Most Recently Used algorithm*
- *Unused slots and broken masters can be automatically disabled after Power On reset*
- *On-chip arbitration function can be disabled and external arbiter can be used.*

16.1.5. PDMAC (PCI DMA Controller)

- *Direct Memory Access (DMA) Controller dedicated to 1-channel PCI*
- *Is possible to transfer data using minimal G-Bus bandwidth*
- *Data can be transferred bidirectionally between the G-Bus and the PCI Bus*
- *Specifying a physical address on the PCI Bus and an address on the G-Bus makes it possible to automatically transfer data between the PCI Bus and the G-Bus*
- *Supports the Chain DMA mode, in which a Descriptor containing chain-shaped addresses and a transfer size is automatically read from memory while DMA transfer continuous*
- *On-chip 4-stage 64-bit data buffer*

16.1.6. Miscellaneous

- *PCI BOOT*
  Boot from PCI Memory Space (0xFFFE_0000)

- *Satellite Mode*
  External Host is able to configure the TX4939 based Adapter System

- *External PCI Clock acceptable*

16.2. Block Diagram

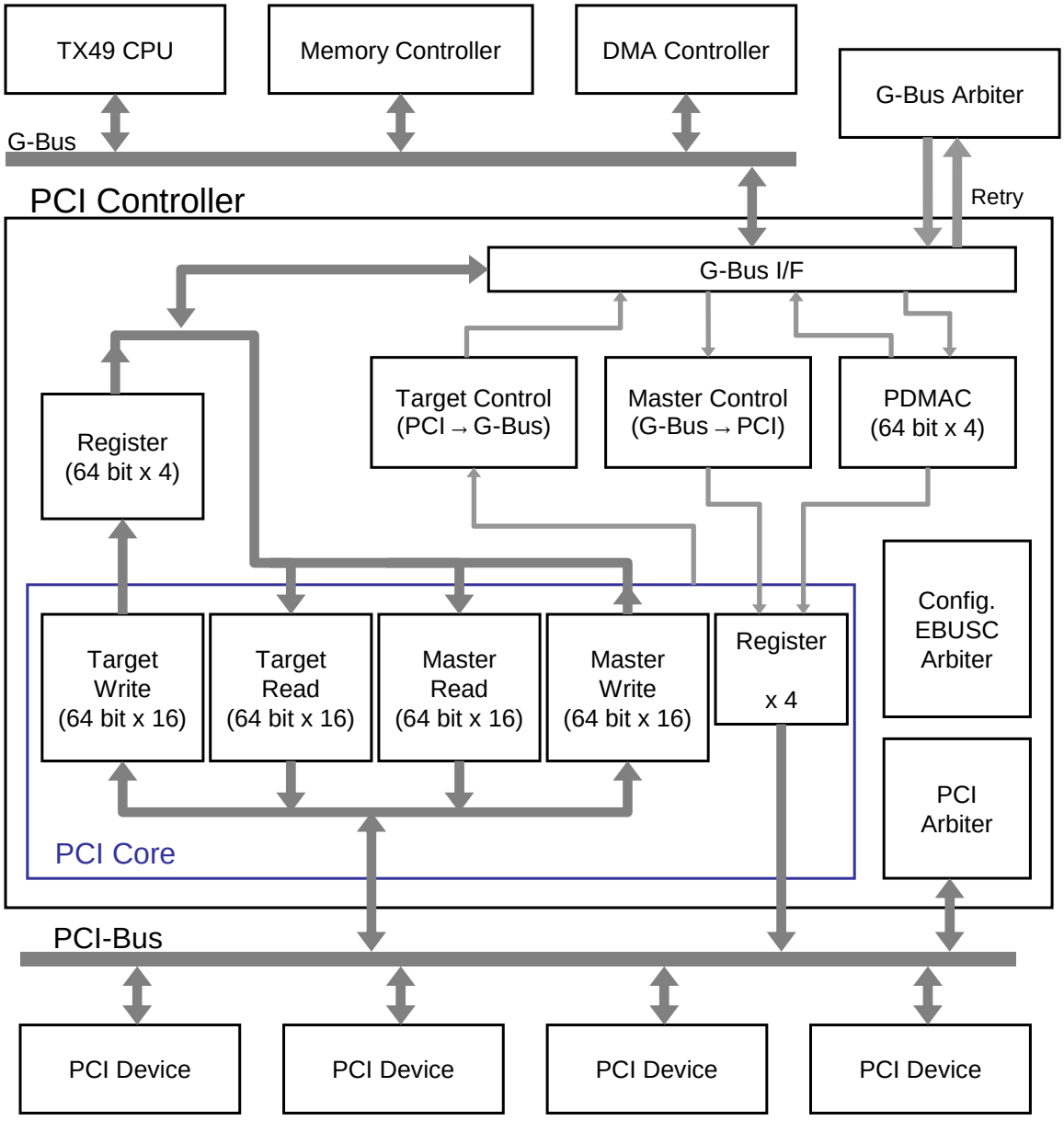


Figure 16-1 PCI Controller Block Diagram

16.3. Detailed Explanation

16.3.1. Terminology Explanation

The following terms are used in this chapter.

- *Initiator*
  Means the bus Master of the PCI Bus. The TX4939 operates as the initiator when it obtains the PCI Bus and issues PCI access.

- *Target*
  Means the bus Slave of the PCI Bus. The TX4939 operates as the target when an external PCI device on the PCI Bus executes PCI access to the TX4939.

- *Host Mode*
  One PCI Host device exists for one PCI Bus. The PCI Host device uses a PCI configuration space to perform PCI configuration on other PCI devices on the PCI Bus. The TX4939 can be configured as Host Mode by boot configuration. Please refer to Boot Configuration Chapter.

- *Satellite Mode*
  A PCI device other than the PCI Host device accepts configuration from the PCI Host device. With this mode, the TX4939 will be used a PCI Adapter card. This state is referred to as the Satellite mode which can be configured by boot configuration. Please refer to Boot Configuration Chapter.

- *PCI BOOT*
  In boot time, the TX49 CPU fetches instruction from GBUS physical address 0x0_1FC0_0000. In normal boot mode, this GBUS memory space is mapped to the boot device on channel 0. In case of PCI BOOT mode, this memory space mapped to the PCI memory space. So that the TX4939 system boot from the device that is mapped such PCI memory space.

- *DWORD, QWORD*
  DWORD expresses 32-bit words, and QWORD expresses 64-bit words. According to conventions observed regarding MIPS architecture, this manual uses the following expressions:

  Byte: 8-bit
  Half-word: 16-bit
  Word: 32-bit
  Double-word: 64-bit

16.3.2. Satellite Mode

The TX4939 with Satellite Mode will provide a PCI Adapter card. Compare with Host Mode, there are some restrictions described in Table 16-1 below.

Table 16-1 Restrictions TX4939 in Satellite Mode

<table>
<tr><th>Item#</th><th>Area</th><th>Description</th><th>Comment</th></tr>
<tr><td>1</td><td>Arbiter</td><td>Internal arbiter can not be used in Satellite Mode. Instead, non used pins for internal arbiter will be used different function as follows.
<table>
<tr><th colspan="2">Pin Name at Host Mode</th><th colspan="3">Pin Usage at Satellite Mode</th></tr>
<tr><td>REQ[0]*</td><td>Input</td><td>REQ*</td><td>Output</td><td>PCI Bus request from the TX4939</td></tr>
<tr><td>GNT[0]*</td><td>Output</td><td>GNT*</td><td>Input</td><td>PCI Bus granted signal from Host arbiter</td></tr>
<tr><td>REQ[1]*</td><td>Input</td><td>INTOUT</td><td>Output</td><td>Interrupt output from the TX4939 interrupt controller</td></tr>
<tr><td>GNT[1]*</td><td>Output</td><td>IDSEL</td><td>Input</td><td>PCI IDSEL input for PCI Configuration Register access.</td></tr>
<tr><td>REQ[2:5]*</td><td>Input</td><td>GPIO[x]</td><td>I/O</td><td>All signal can be used as extra GPIO</td></tr>
<tr><td>GNT[2:5]*</td><td>Output</td><td>GPIO[x]</td><td>I/O</td><td>All signal can be used as extra GPIO</td></tr>
</table></td><td></td></tr>
<tr><td>2</td><td>PME</td><td>Power Management in Satellite Mode is not supported in this version.</td><td></td></tr>
<tr><td>3</td><td>PCICLK</td><td>In the current version of the TX4939, PCICLK can only be 33.33 MHz</td><td></td></tr>
</table>

16.3.3. PCI Boot

PCI BOOT Mode can be set during boot up. Please see Boot Configuration chapter for detail.
Two windows of the memory space from the G-Bus to the PCI Bus space are used when in the PCI Boot mode. The defaults of several registers are changed as indicated Table 16-2 below.

Table 16-2 PCI BOOT Related Register Assignment

Register Names	Values
G-Bus base address (G2GBASE):	0x0_1FC0_0000
Space size (G2PM2MASK):	128 KB
PCI Bus base address (G2PM2PBASE):	0x00_FFFE_0000
Initiator Memory Space 2 Enable (PCICCFG.G2PM2EN):	1
Bus Master bit (PCISTATUS.BM)	1 [Only when in the Host mode]
Target Configuration Access Ready (PCICSTAUTS.TCAR)	1 [Only when in the Satellite mode]

Note: The on-chip PCI Bus Arbiter cannot be used when the PCI Boot mode is being used while in the Satellite mode.

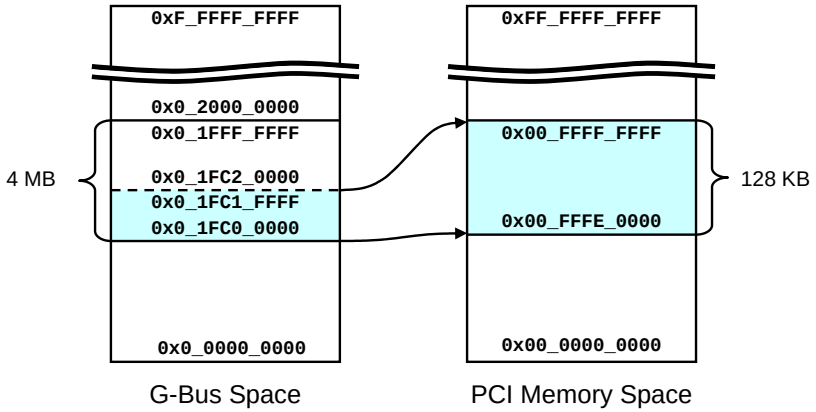


Figure 16-2 PCI BOOT default mapping

16.3.4. Sample PCI Adapter Configuration

Figure 16-3 shows example of PCI Adapter system by the TX4939 with Satellite Mode.

In this configuration, the TX4939 is in Satellite Mode and PCI BOOT is activated.
After PCI Host (ASIC SoC) configured PCI BOOT Device by mapping the part of Boot ROM as memory device on PCI Memory Space (i.e. 0xFFFE_0000 – 0xFFFF_FFFF), PCI Host might release RESET_OUT for the TX4939.

Then the TX4939 will boot from CPU address 0x1FC0_0000, which internally mapped to 0xFFFE_0000 of the PCI Memory Space.

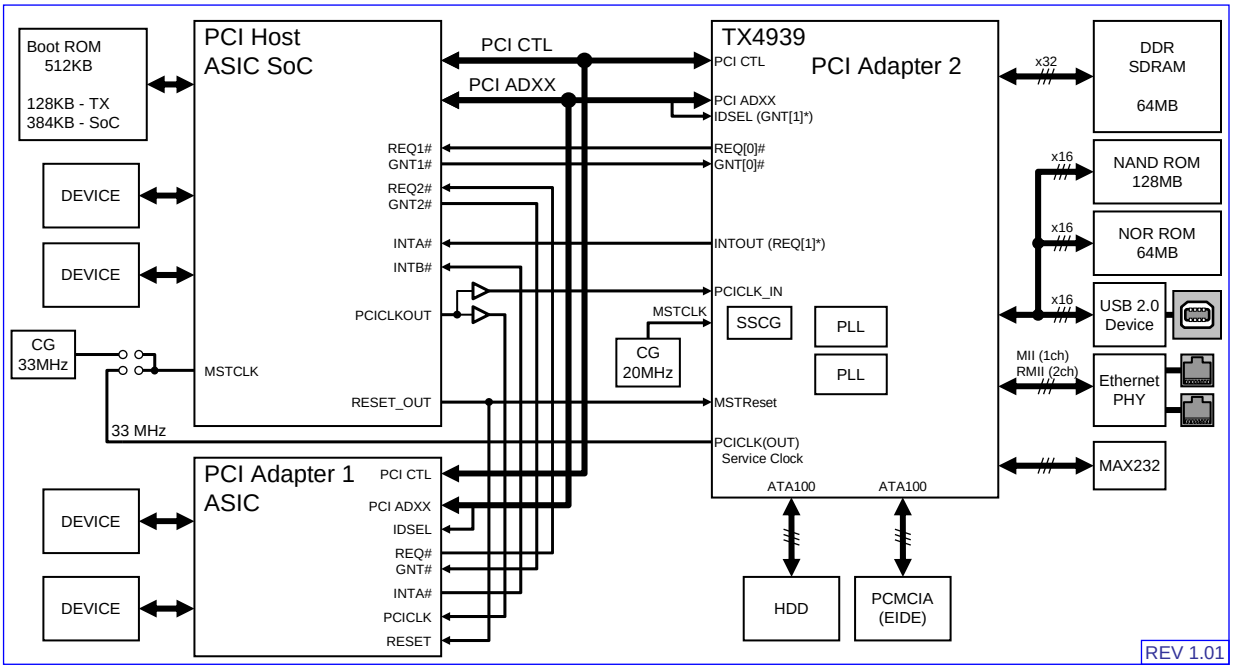


Figure 16-3 Block Diagram of Sample PCI Adapter

The TX4939 needs 20MHz Clock as master clock. This master clock can be modulated by internal SSCG (Spread Spectrum Clock Generator), then generates 400MHz fixed clock first. A 100MHz and 66.66MHz clock will be derived from this 400 MHz for ATA100 and PCI.

The TX4939 has 6 PCI Clock output (Service Clock either 33 MHz or 66 MHz). This internal 66.66 MHz clock is fed to second PLL, which generates CPU, GBUS, DDR, and External bus clocks. CPU pipeline clock can be any frequency integer multiple of 33.33 MHz and range is from 266.66 MHz to 400 MHz.

16.3.5. On-chip Register

The PCI Controller on-chip register contains the PCI Configuration Space Register and the PCI Controller Control Register. The registers that can be accessed vary according to whether the current mode is the Host mode or the Satellite mode.

An external PCI Host device only accesses the PCI Configuration Space Register when in the Satellite mode. This register is defined in the PCI Bus Specifications. A PCI configuration cycle is used to access this register. This register cannot be accessed when in the Host mode. Section 16.4 “PCI Configuration Space Register” explains each register in detail.
The PCI Controller Control Register is only accessed by the TX49 core and cannot be accessed from the PCI Bus.

Registers in the PCI Controller Control Register that include an offset address in the range from 0xD000 to 0xD07F can only be accessed when in the Host mode and cannot be accessed when in the Satellite mode. These registers correspond to PCI Configuration Space Registers that an external PCI Host device accesses when in the Satellite mode. Exceptions to above access limitations are to address 0xD010, 0xD018, 0xD020 and 0xD024, which correspond to memory 0,1,2, base address registers and io base register. These four registers are Read/Write accessible in Host mode and Read Only accessible in satellite mode. Section 16.4 “PCI Controller Control Register” explains each register in detail. Figure 16-4 illustrates the register map when in the Host mode. Figure 16-5 illustrates the register map when in the Satellite mode.

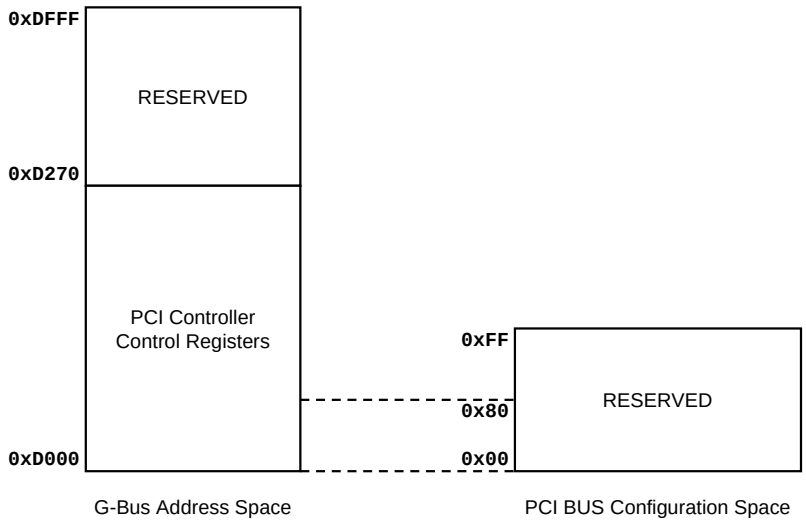


Figure 16-4 Register Map (Host Mode)

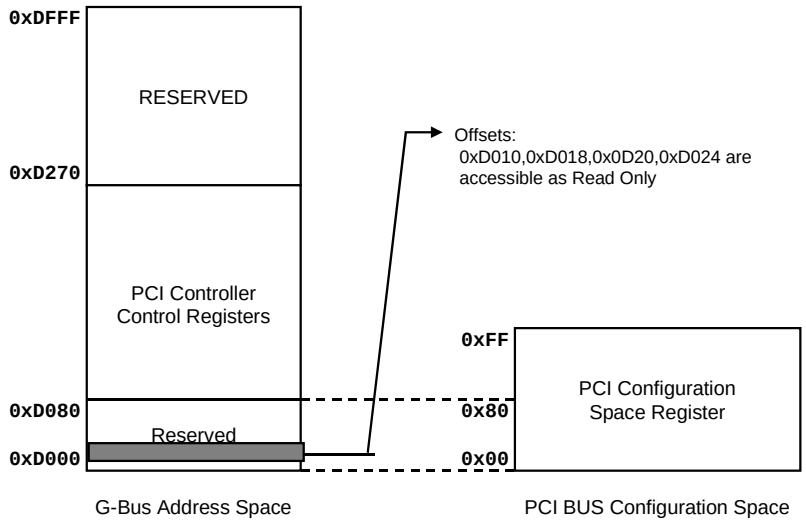


Figure 16-5 Register Map (Satellite Mode)

16.3.6. Supported PCI Bus Commands

Table 16-3 shows the PCI Bus commands that the PCI Controller supports.

Table 16-3 Supported PCI Bus Commands

C/BE Value	PCI Command	As Initiator (Note2)	As Target (Note2)
0000	Interrupt Acknowledge	Host	—
0001	Special Cycle	Host	—
0010	I/O Read	Host & Satellite	Host & Satellite
0011	I/O Write	Host & Satellite	Host & Satellite
0100	(Reserved)	—	—
0101	(Reserved)	—	—
0110	Memory Read (Note 1)	Host & Satellite	Host & Satellite
0111	Memory Write (Note 1)	Host & Satellite	Host & Satellite
1000	(Reserved)	—	—
1001	(Reserved)	—	—
1010	Configuration Read	Host	Satellite
1011	Configuration Write	Host	Satellite
1100	Memory Read Multiple	Host & Satellite	Host & Satellite
1101	Dual Address Cycle	Host & Satellite	Host & Satellite
1110	Memory Read Line	Host & Satellite	Host & Satellite
1111	Memory Write and Invalidate	Host & Satellite	Host & Satellite

Note1: The byte enable signals are asserted as necessary during memory read and memory write cycles same as I/O Read, I/O Write and Single Access commands. During burst memory reads, four byte enable signals will be asserted.

Note2: Host means “supported only at Host mode”.
Satellite means “supported only at Satellite mode”.
Host & Satellite means “supported at both mode”.

- *I/O Read, I/O Write, Memory Read, Memory Write*
  This command executes Read/Write access to the address mapped on the G-Bus and PCI Bus.

- *Memory Read Multiple, Memory Read Line*
  The Memory Read Multiple command is issued if all of the following conditions are met when the Initiator function is operating and Burst Read access is issued from the G-Bus to the PCI Bus.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (2) The Read data word count is larger than the value set in the Cache Line Size Field.

  Also, the Read Memory Line command is issued when all of the following conditions are met.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (2) The Read data word count is smaller than the value set in the Cache Line Size Field.

  The Memory Read command is issued if these conditions are not met, namely, if “0” is set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register. In the case of the target, a normal G-Bus cycle is issued to the address mapped from the PCI Bus to the G-Bus.

- *Memory Write and Invalidate*
  When the TX4939 operates as the initiator, the PCI Controller issue the Memory Write and Invalidate command if all of the following conditions are met when write access from the G-Bus to the PCI Bus occurs.

  (1) The Memory Write and Invalidate Enable bit (PCISTATUS.MWIEN) of the PCI Status Command Register is set.

  (2) A value other than "0" was set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (3) The word count of the Write data is larger than the value set in the Cache Line Size field. The Memory Write command is issued if these conditions are not met.

  When the TX4939 operates as the target, the Memory Write and Invalidate command is converted into G-Bus Write access.

  Note that the TX4939 does not support the cache memory Snoop function.

- *Dual address cycle*
  When the TX4939 operates as the initiator, the PCI Controller executes dual access cycles if the PCI Bus address exceeds 0x00_FFFF_FFFF.

  When the TX4939 operates as the target, normal G-Bus cycles are executed to the address mapped from the PCI Bus to the G-Bus.

- *Configuration Read, Configuration Write (Only be used by Initiator at Host mode)*

  The configuration cycles are issued on the PCI Bus by either reading or writing from/to the G2P Configuration Data Register (G2PCFGDATA) after writing the configuration space address to the G2P Configuration Address Register. The TX4939 supports both "Type 0" and "Type 1" configuration transactions.

  On systems that have PCI card slots, the PCI Host device checks each PCI card slot during system initialization to see if any PCI device exists, then set the Configuration Space Register of the devices that do exist. If a PCI Configuration Read operation is performed for devices that do not exist, then by default a Bus Error exception will be generated since there is no PCI Bus response. Clearing the Bus Error Response During Initiator Read bit (PCICFG.IRBER) of the PCI Controller Configuration Register makes it possible to execute a Read transaction without causing a Bus Error. All bits of the data read at this time will be set to "1".

- *Accepting Configuration Read and Write (Only be accepted by Target at Satellite mode).*

  Configuration cycles will be accepted as the target only when in the Satellite mode. After reset, Retry response to PCI Configuration access will continue until the software sets the Target Configuration Access Ready Bit (PCICFG.TCAR) of the PCI Controller Configuration Register. Please use the software to set this bit after the software initialization process ends and the software is ready to accept PCI configuration.

- *Interrupt Acknowledge*
  This command issues interrupt acknowledge cycles as an initiator. Interrupt acknowledge cycles are executed on the PCI Bus when the G2P Interrupt Acknowledge Data Register (G2PINTACK) is read. The value returned by this Read becomes the interrupt acknowledge cycle data.

  The TX4939 does not support interrupt acknowledge cycles as the target.

- *Special Cycle*
  This command issues specialy cycles as the initiator. This command issues special cycles on the PCI Bus when writing to the G2P Special Cycle Data Register (G2PSPC). The written value is output as the special cycle data.

  The TX4939 does not support special cycles as the target.

16.3.7. Initiator Access (G-Bus → PCI Bus Address Conversion)

During PCI initiator access, the G-Bus address of the Burst transaction issued by the G-Bus that was converted into the PCI Bus address is used to issue a Burst transaction on the PCI Bus. 36-bit physical addresses (G-Bus addresses) are used on the G-Bus. Also, 40-bit PCI Bus addresses are used on the PCI Bus.

Three memory access windows and one I/O access window can be set in the G-Bus space (Figure 16-6). The size of each window is variable. When Burst transactions are issued to these access windows on the G-Bus, then that G-Bus address is converted into a PCI Bus address that is used to issue a Burst transaction to the PCI Bus as the initiator. PCI memory access is issued when the access window is the memory access window. PCI I/O access is issued when the access window is the I/O access window. Dual access cycles are also issued to the PCI Bus when the PCI Bus address exceeds 0x00_FFFF_FFFF.

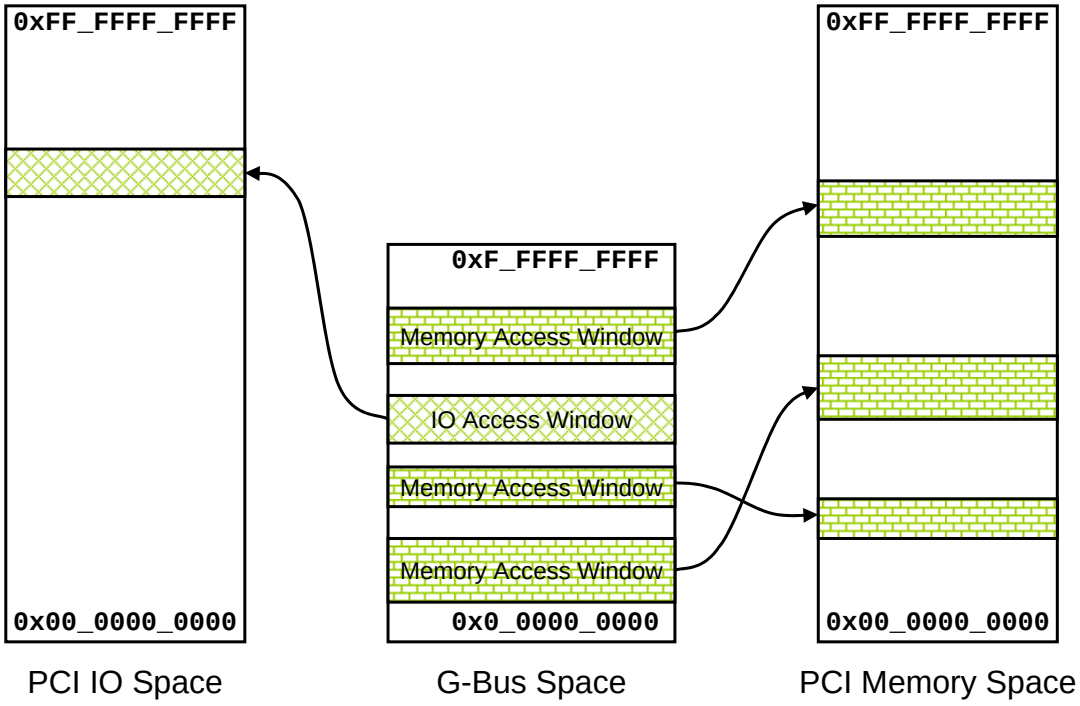


Figure 16-6 Initiator Access Memory Window

When expressed as a formula, conversion of a G-Bus address (GBusAddr[35:0]) into a PCI Bus Address (PCIAddr[39:0]) is as follows below. GBASE[35:8], PBASE[39:8], and AM[35:8] each represent the setting register of the corresponding access window indicated below in Table 16-4. The "&" symbol indicates a logical AND for each bit, "||" indicates a logical OR for each bit, "!" indicates logical NOT, and "|" indicates bit linking.

```
If (GBusAddr[35:8] & ! AM[35:8] == GBASE[35:8] & ! AM[35:8]) then
      PCIAddr[39:0] =        PBASE[39:36]
                           | ((PBASE[35:8] & ! AM[35:8]) || (GBusAddr[35:8] & AM[35:8]))
                           | GBusAddr[7:0];
```

Table 16-4 Initiator Access Space Address Mapping Register

	G-Bus Base Address GBASE[35:8]	PCI Bus Base Address PBASE[39:8]	Address Mask AM[35:8]
Memory Space 0	G2PM0GBASE.BA[35:8]	G2PM0PBASE.BA[39:8]	G2PM0MASK.AM[35:8]
Memory Space 1	G2PM1GBASE.BA[35:8]	G2PM1PBASE.BA[39:8]	G2PM1MASK.AM[35:8]
Memory Space 2	G2PM2GBASE.BA[35:8]	G2PM2PBASE.BA[39:8]	G2PM2MASK.AM[35:8]
I/O Space	G2PIOGBASE.BA[35:8]	G2PIOPBASE.BA[39:8]	G2PIOMASK.AM[35:8]

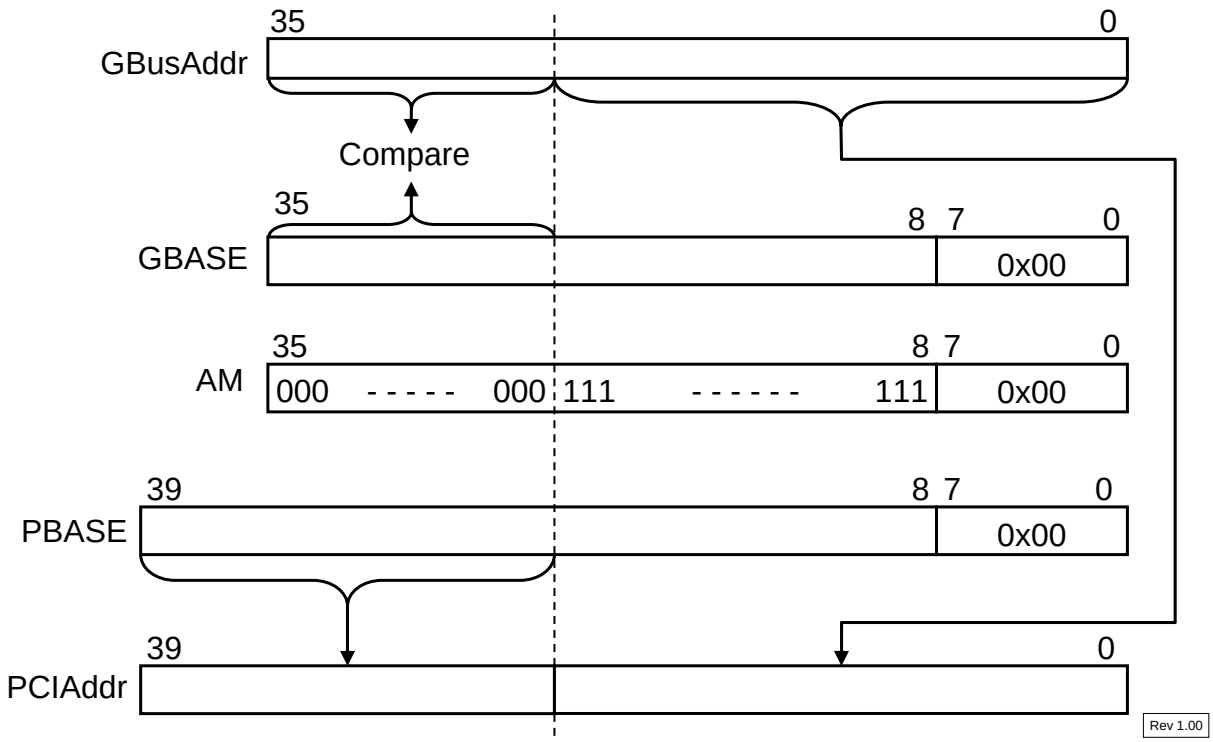


Figure 16-7 Address Conversion for Initiator (GBus -> PCI Bus Address Conversion)

It is possible to set each space to valid/invalid or to perform Word Swap (see 16.3.10 “Endian Switching Function). Table 16-5 shows the settings registers for these properties.

When 64-bit access is made to the initiator memory space, two 32-bit Burst accesses are issued on the PCI Bus. 64-bit access to the I/O space is not supported.

Also, operation is not guaranteed if resources in the PCI space were made cacheable and were then accessed when the Critical Word First function of the TX49/H4 core was enabled.

Table 16-5 Initiator Access Space Properties Register

	Enable	Endian Swap
Memory Space 0	*BusMasterEnable* & PCICCFG.G2PM0EN	G2PM0GBASE.BSWAP, G2PM0GBASE.EXFER
Memory Space 1	*BusMasterEnable* & PCICCFG.G2PM1EN	G2PM1GBASE.BSWAP, G2PM1GBASE.EXFER
Memory Space 2	*BusMasterEnable* & PCICCFG.G2PM2EN	G2PM2GBASE.BSWAP, G2PM2GBASE.EXFER
I/O Space	*BusMasterEnable* & PCICCFG.G2PIOEN	G2PIOGBASE.BSWAP, G2PIOGBASE.EXFER

BusMasterEnable: PCI State Command Register Bus Master Bit (PCISTATUS.BM)

16.3.8. Target Access (PCI Bus → G-Bus Address Conversion)

During PCI target access, the PCI Bus address of the Bus transaction issued by the PCI Bus is converted into a G-Bus address and is used to issue a Bus transaction on the G-Bus. 40-bit PCI Bus addresses are used on the PCI Bus. Also, 36-bit physical addresses are used on the G-Bus.

Three memory access windows and one I/O access window can be set in the PCI bus space (Figure 16-8).
When the Bus transactions to these access windows are issued on the PCI Bus, these Bus transactions are accepted as PCI target devices. The PCI Bus Address is converted into G-Bus addresses, and then corresponding Bus transactions are issued to the G-Bus.
The memory space window responds to the PCI memory space access command. The I/O space window responds to the PCI I/O space access command.

Note: Byte swapping is always disabled when prefetch mode is disabled. When the G-Bus is configured for big-endian mode, the order of bits in a 32-bit word does not change during a PCI transfer. (The byte ordering changes.)

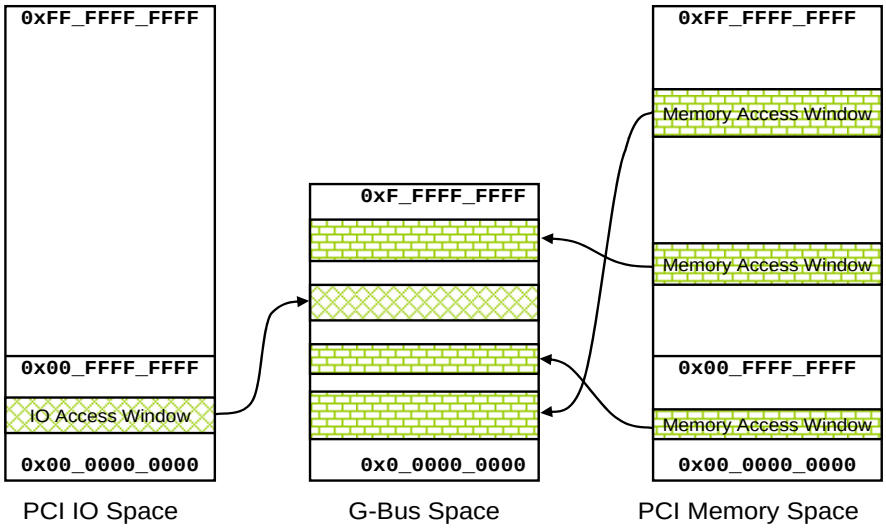


Figure 16-8 Target Access Memory Window

When expressed as a formula, conversion of a PCI Bus Address (PCIAddr[39:0]) into a G-Bus address (GBusAddr[35:0]) is as follows below. GBASE[35:8], and PBASE[39:8] each represent the setting register of the corresponding access window indicated below in Table 16-7. The "&" symbol indicates a logical AND for each bit, and "|" indicates bit linking.

Memory space 0, 1, and 2 If (PCIAddr[39:n] == P2GM0PUBASE.BA[39:32] | P2GM0PLBASE.BA[31:n] then
GBusAddr[35:0] = P2GM0GBASE[35:n] | PCIAddr[(n-1):0];

Note: Memory space 0,1 and 2 in the TX4939 has fixed bit linking as given below
GbusAddr[35:0] = P2GM{X}GBASE[35:29]|PCIAddr[28:0], where X = 0,1,2

I/O space If (PCIAddr[31:8] == P2GIOPBASE.BA[31:8]) then
GBusAddr[35:0] = P2GIOGBASE[35:8] | PCIAddr[7:0];

Table 16-6 Corresponding of Memory Space Size and MSS[31:20] Value

MSS[31:20]	31	30	29	28	27	26	25	24	23	22	21	20	n	Memory Size
0xE00	1	1	1	0	0	0	0	0	0	0	0	0	29	512 MB
0xF00	1	1	1	1	0	0	0	0	0	0	0	0	28	256 MB
0xF80	1	1	1	1	1	0	0	0	0	0	0	0	27	128 MB
0xFC0	1	1	1	1	1	1	0	0	0	0	0	0	26	64 MB
0xFE0	1	1	1	1	1	1	1	0	0	0	0	0	25	32 MB
0xFF0	1	1	1	1	1	1	1	1	0	0	0	0	24	16 MB
0xFF8	1	1	1	1	1	1	1	1	1	0	0	0	23	8 MB
0xFFC	1	1	1	1	1	1	1	1	1	1	0	0	22	4 MB
0xFFE	1	1	1	1	1	1	1	1	1	1	1	0	21	2 MB
0xFFF	1	1	1	1	1	1	1	1	1	1	1	1	20	1 MB

Note: MSS[31:20] Field locates inside the register of P2GM(m)CFG. See detail in 16.4.6

Table 16-7 Target Access Space Address Mapping Register

	Space Size	PCI Address	PCI Bus Base Address PBASE	G-Bus Base Address GBASE
Memory Space 0	1 – 512MB	40-bit	P2GM0PUBASE.BA[39:32] | P2GM0PLBASE.BA[31:n]	P2GM0GBASE.BA[35:n]
Memory Space 1	1 – 512MB	40-bit	P2GM1PUBASE.BA[39:32] | P2GM1PLBASE.BA[31:n]	P2GM1GBASE.BA[35:n]
Memory Space 2	1 – 512MB	32-bit	P2GM2PUBASE.BA[39:32] | P2GM2PLBASE.BA[31:n]	P2GM2GBASE.BA[35:n]
I/O Space	256 B	32-bit	P2GIOPBASE.BA[31:8]	P2GIOGBASE.BA[35:8]

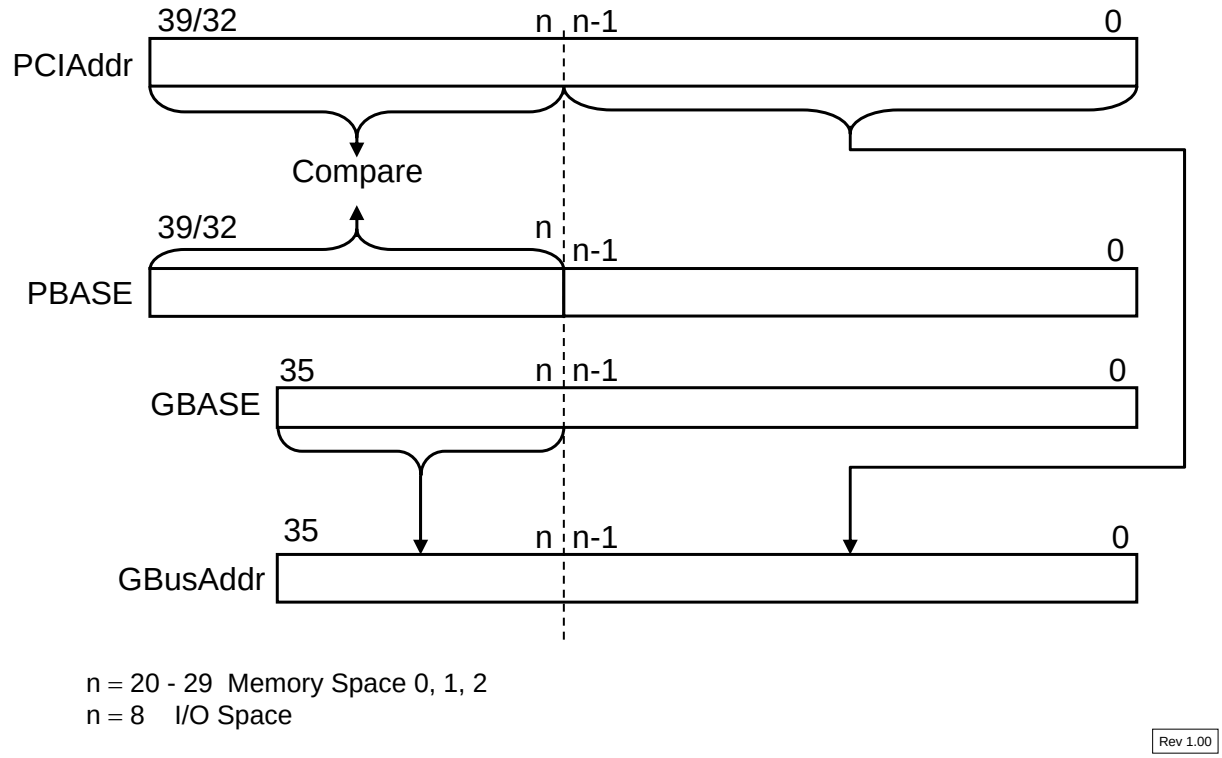


Figure 16-9 Address Conversion for Target (PCI Bus (PCI Bus → G-Bus Address Conversion)

16

It is possible to set each space to valid/invalid, pre-fetch Read to valid/invalid, or to perform Word Swap (see16.3.10). Table 16-8 shows the settings registers for these properties.
When pre-fetch Reads are set to valid, data transfer is performed on the G-Bus according to the size set by the Target Pre-fetch Read Burst Length Field (P2GCFG.TPRBL) of the P2G Configuration Register during a PCI target Read transaction. This is performed using accesses to resources that will not be affected even if a pre-read such as memory is performed. Also, PCI Burst Reads to memory spaces that were set to I/O space and pre-fetch disable are not supported.

Note: Always use PCI single reads. Don't use burst reads.

Table 16-8 Target Access Space Properties Register

	Enable	Pre-fetch (Initial State)	Endian Swap
Memory Space 0	PCICCFG.TCAR & *MemEnable* & P2GM0GBASE.P2GM0EN	P2GCFG.MEM0PD (valid)	P2GM0GBASE.BSWAP, P2GM0GBASE.EXFER
Memory Space 1	PCICCFG.TCAR & *MemEnable* & P2GM1GBASE.P2GM1EN	P2GCFG.MEM1PD (valid)	P2GM1GBASE.BSWAP, P2GM1GBASE.EXFER
Memory Space 2	PCICCFG.TCAR & *MemEnable* & P2GM2GBASE.P2GM2EN	P2GCFG.MEM2PD (invalid)	P2GM2GBASE.BSWAP, P2GM2GBASE.EXFER
I/O Space	PCICCFG.TCAR & *IOEnable* & P2GIOGBASE.P2GIOEN	Always invalid	P2GIOGBASE.BSWAP, P2GIOGBASE.EXFER

MemEnable: PCI State Command Register Memory Space bit (PCISTATUS.MEMSP)

IOEnable: PCI State Command Register I/O Space bit (PCISTATUS.IOSP)

16.3.9. Post Write Function

The Post Write function improves system performance by completing the original bus Write transaction without waiting for the other bus to complete its transaction when the first bus issues a Write transaction. Initiator Write can Post Write a maximum of four (4) Write transactions, and Target Write can Post Write a maximum of five (5) Write transactions.

Due to compatibility issues with old PC software in the PCI specifications, performing Post Writes with Initiator Configuration Write and Target I/O Write is not recognized. However, the TX4939 PCI Controller can even perform Post Writes to these functions. In order to guarantee that these Writes are completed by the target device, please execute Reads to the device that performed the Write, then either refer to the read value (so the TX49/H4 core can support non-blocking load) or execute the SYNC instruction.

16.3.10. Endian Switching Function

The TX4939 supports both the Little Endian mode and the Big Endian mode. On the other hand, the PCI Bus is defined in Little Endian logic only. Therefore, when the TX4939 is in the Big Endian mode, either the software or the hardware must perform some kind of conversion when exchanging data with the PCI Bus.

The hardware provided in this Endian Switching is described in Figure 16-10 below. Those swapping are controlled by EXFER bit and BSWAP bit in corresponding registers, such as G2PMnGBASE, G2PIOGBASE, P2GMnGBASE, and P2GIOGBASE. The BSWAP is 36th bit and the EXFER is 37th bit of those registers. Also the power on default value of these bits are set according to the CPU Endianness.

Endian switching during initiator access is specified by the Swap bit (BSWAP, EXFER) of the G-Bus Base Address Register (G2PMnGBASE, G2PIOGBASE) of the access window for each initiator access (see Table 16-5).
Endian switching during target access is specified by the Swap bit (BSWAP, EXFER) of the G-Bus Base Address Register (P2GMnGBASE, P2GIOGBASE) of the access window for each target access (see Table 16-8).
And when the PDMAC is used, the Swap bit (BSWAP, EXFER) of the PDMAC Control Register is specified accordingly to above Swap bit setting.

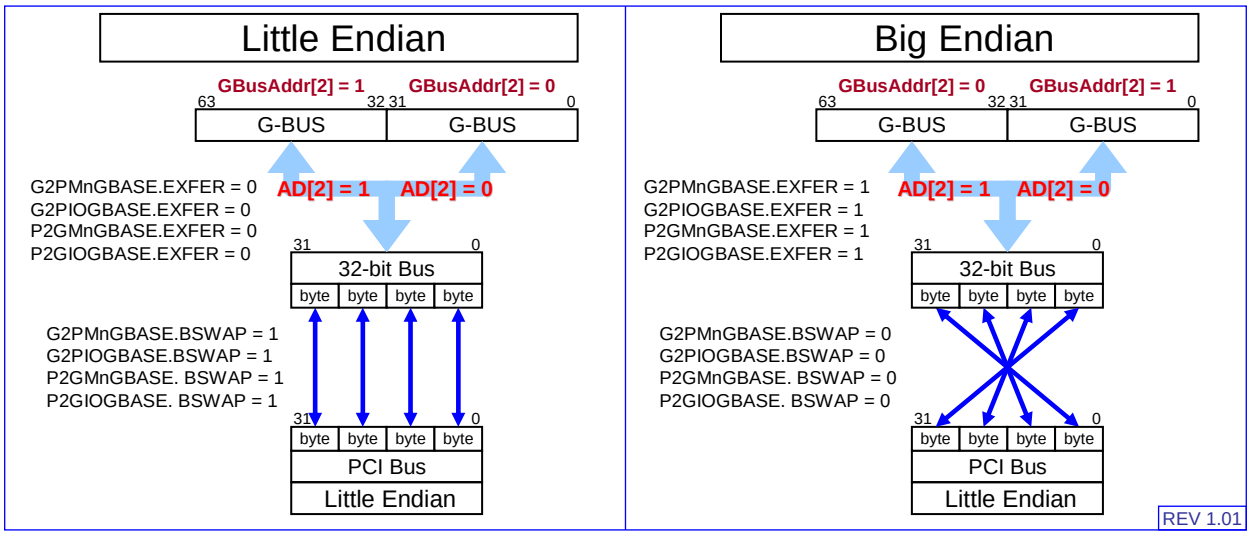


Figure 16-10 Endian Switching.

In case of Little Endian, there is no issue. But in case of Big Endian, swapping will happen between G-Bus and PCI Bus. These swapping, even it is commonly used, just support DMA data transfer. Because of this, any read or write operation to PCI by CPU should have appropriate counter measure.

Commonly used counter measure are “swap the data before write” and “swap the data after read”. Those detail will be found in any Bi-Endian coded PCI Peripheral driver.

16.3.11. 66 MHz Operation Mode

The TX4939 PCI Controller supports 66 MHz PCI. The way the TX4939 decides whether to operate the PCI in 33/66 MHz mode is described below.

- *The M66EN pin is a ground pin in 33MHz card connectors.*
- *If there is no 33MHz card connected to the PCI system, then this pin remains un-driven high.*
- *If at least one card connected to the PCI system is a 33MHz connector, then the ground track at the place of M66EN pin position pulls the M66EN signal down.*
- *The TX4939 PCI clock generator detects the polarity of the M66EN signal. If the M66EN signal is pulled high, the clock generator generates 66MHz clock, else if it is low, it generates 33 MHz clock.*

The clock for the internal PCI Bus is always at 66MHz.
It is possible to read the state of the external PCI Bus clock from the 66 MHz Drive Status bit (P2GSTATUS.M66EN) of the P2G Status Register.

PCI Controller can be reset by either the TX4939 Master Reset or PCI hardware reset register.
The software uses a hardware reset (PCICCFG.HRST of the PCI Controller Configuration Register) to reset the PCI Controller. The PCIRST signal to external devices can be asserted when asserting EPCIRST bit in clock control register.

16.3.12. Power Management

The TX4939 PCI Controller supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1.
The PCI Host device controls the system status by reporting the power management state to the PCI Satellite device. Also, the PCI Satellite device uses the PME* signal to report requests for changing the power management state or to report to the PCI Host device that a power management event has occurred.

16.3.12.1. Power Management State

In the case of the PCI Bus Power Management Interface Specifications, four power management states are defined from State D0 to State D3. The TX4939 supports states D0 through D3. Figure 16-11 illustrates the power management state transition.
After Power On Reset, or when transitioning from the $D3_{HOT}$ state to the D0 state, the power management state becomes uninitialized D0. If initialized by the system software at this point, the state transitions to D0 Active.
The TX4939 uses the software to change the system status after a status change is detected.

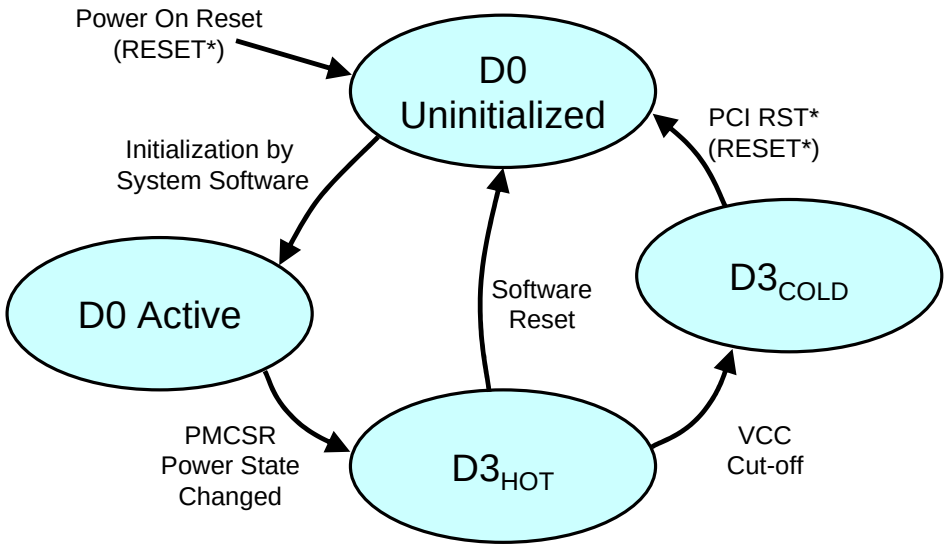


Figure 16-11 Transition of the Power Management States

16.3.12.2. PME* Signal (Satellite Mode) [Note This Function is Disabled]

The following PMEs (Power Management Events) are reported when in the Satellite mode.

- *The PCI Host device sets the PME_En bit of the PMCSR Register in the TX4939 configuration space. This makes it possible for the TX4939 to assert the PME* signal.*
- *Then, the PME_En Set bit (P2GSTATUS.PMEES) of the P2G Status Register is set. Furthermore, it also becomes possible to generate PME_En Set interrupts. The PME_En bit value can be read from the PME_En bit (PCISSTATUS.PMEEN) of the Satellite Mode PCI Status Register.*
- *Writing "1" to the PME bit (P2GCFG.PME) of the P2G Configuration Register sets the PME_Status bit of the PMCSR Register, then asserts the PME* signal, which is the open drain signal. PME is then reported to the PCI Host device.*
- *The PCI Host device checks the PMCSR PME_Status bit of each PCI device, then specifies the PCI device that asserted the PME* signal.*
- *After the process corresponding to PME ends, the PCI Host device writes "1" to the TX4939 PME_Status bit that reported PME, thereby reporting the end of the process. As a result, the PME_Status bit of the PMCSR Register is cleared and the PME* signal is deasserted.*

  Then, the PME Status Clear bit (P2GSTATUS.PMECLR) of the P2G Status Register is set. It is also possible to generate PME Status Clear interrupts.

16.3.12.3. PME* Signal* (Host Mode)

The PME Detection bit (PCICSTATUS.PMED) of the PCI Controller Status Register is set when an external satellite device asserts the PME* signal while the TX4939 is in the Host mode. It is also possible to generate PME Detection interrupts at this time.

- *The TX4939 software should then check the PMCSR PME_Status bit of each PCI device on PCI Bus to detect which PCI device asserted the PME* signal.*
- *Once the device which asserted the PME* signal is identified, then the process (routine to service the PME* assertion) corresponding to the PCI device which asserted the PME* should be started by the TX4939 software.*
- *After the process corresponding to PME ends, the TX4939 writes "1" to the PME_Status bit of the PCI device that reported PME, thereby reporting the end of the process. As a result, the PME_Status bit of the PMCSR Register in the PME reporting PCI device is cleared and the PME* signal is deasserted.*

16.3.13. PDMAC (PCI DMA Controller)

The PCI DMA Controller (PDMAC) is a one-channel PCI Director Memory Access (DMA) controller. Data can be transferred bidirectionally between the G-Bus and the PCI Bus.
Note: The PDMAC can only access the DDR controller on the G-Bus. It does not provide support for access to other controllers on the G-Bus.

16.3.13.1. Simple (Non-Chain) DMA Transfer

The steps to follow for a non-chain DMA transfer mode are described below.

(1) Address Register and Count Register Setting
Set values for the following three registers.

- *PDMAC G-Bus Address Register (PDMGA)*
- *PDMAC PCI Bus Address Register (PDMPA)*
- *PDMAC Count Register (PDMCTR)*

(2) Chain Address Register Setting
Set “0” to the PDMAC Chain Address Register (PDMCA).

(3) PDMAC Status Register (PDMSTATUS) Clearing
Clear any remaining status from a previous DMA transfer.

(4) PDMAC Control register (PDMCFG) Setting
Clear the Channel Reset bit (CHRST), and make settings such as the data transfer direction (XFRDIRC), and the data transfer unit size (XFRSIZE).

(5) DMA Transfer Initiation
Set the Transfer Active bit (XFRACT) of the PDMAC Control Register initiates DMA transfer.

(6) Termination Report
When the DMA data transfer terminates normally, the Normal Data Transfer Completion bit (NTCMP) of the PDMAC Status Register (PDMSTATUS) is set. An interrupt is then reported if the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) of the PDMAC Control Register is set.
If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Control register is set.

16.3.13.2. Chain DMA

DMA Command Descriptors are 4 QWORD (32-Byte) data structures indicated in Table 16-9 that are placed in memory. Storing the starting memory address of another DMA Command Descriptor in the Offset 0 Chain Address Field makes it possible to configure a chain list for the DMA command Descriptor. Set “0” in the Chain Address field of the DMA Command Descriptor at the end of the chain list.
When the DMA transfer specified by one DMA Command Descriptor ends, the PDMAC reads the next DMA Command Descriptor that the Chain Address field automatically points to, then continues the DMA transfer. Such continuous DMA transfer that uses multiple descriptors in a chain format is referred to as the Chain DMA mode.
When a DMA Command Descriptor is placed to an address that does not extend across a 32 QWORD boundary in memory, this transfer method is more efficient since data can be read by a single G-Bus Burst Read transaction.

Table 16-9 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	PDMAC Chain Address Register (PDMCA)
0x08	G-Bus Address	PDMAC G-Bus Address Register (PDMGA)
0x10	PCI Bus Address	PDMAC PCI Bus Address Register (PDMPA)
0x18	Count	PDMAC Count Register (PDMCTR)

The DMA transfer procedure is as follows when in the Chain DMA mode.

(1) Count Register Setting
Sets “0” to the PDMAC Count Register (PMDCTR).

(2) DMA Command Descriptor Chain Construction
Constructs the DMA Command Descriptor Chain in memory.

(3) PDMAC Status Register (PDMSTATUS) Clearing
Clears any remaining status from a previous DMA transfer.

(4) PDMAC Control Register (PDMCFG) Setting
Clears the Channel Regster bit (CHRST) and makes settings such as the data transfer direction (XFRDIRC) and the data transfer unit size (XFRSIZE).

(5) DMA Transfer Initiation
Setting the address of the DMA Command descriptor that is at the beginning of the Chain List in the PDMAC Chain Address Register (PDMCA) automatically initiates DMA transfer.
First, the values stored in each field of the DMA Command Descriptor that is at the beginning of the Chain List are read to each corresponding PDMAC Register, then DMA transfer is performed according to the read values.
If a value other than “0” is stored in the PDMAC Chain Address Register (PDMCA), data transfer of the size stored in the PDMAC Count Register is complete, then the DMA Command Descriptor value for the memory address specified by the PDMAC Chain Address Register is read.
When the Chain Address field value reads a descriptor of “0”, the PDMAC Chain Address Register value is not updated and the previous value (address of the Data Command Descriptor at which the Chain Address field value is “0” when read) is held.
0 value judgement is performed when the lower 32 bits of the PDMAC Chain Address Register are rewritten. DMA transfer is automatically initiated if the value was not “0”. Therefore, please write to the upper 32 bits first when writing to the PDMAC Chain Address Register using a 32-bit Store instruction.

(6) Termination Report
When DMA data transfer of all descriptor chains terminates normally, the Normal Chain Complete bit (NCCMP) of the PDMAC Status Register is set. An interrupt is reported if the Chain Termination Interrupt Enable bit (MCCMPIE) of the PDMAC Control register (PDMCFG) is set.
Also, the Normal Data Transfer Complete bit (NTCMP) of the DPMAC Status Register is set each time the DMA data transfer specified by a DMA Command Descriptor terminates normally. An interrupt is reported if the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) of the PDMAC Control Register (PDMCFG) is set.
If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Control register is set.

16.3.13.3. Dynamic Chain Operation

It is possible to dynamically add other DMA Command Descriptor Chains to a DMA Command Descriptor Chain that is currently being processed when executing DMA data transfer. This is done according to the following procedure.

(1) DMA Command Descriptor Chain Construction
Constructs a DMA Command Descriptor Chain in memory.

(2) Addition of DMA Command Descriptor Chains
Substitutes the address of the command descriptor that is at the beginning of the descriptor chain to be added into the Descriptor Chain Address field at the end of the DMA Command Descriptor Chain that is currently performing DMA transfer.

(3) Chain Enable bit checking
Reads the value of the Chain Enable bit (CHNEN) in the PDMAC Control Register (PDMCFG). If the read value is “0”, then the Chain Address field value of the DMA Command Descriptor indicated by the address stored in the PDMAC Chain Address Register (PDMCA) is written to the PDMAC Chain Address Register (PDMCA)

16.3.13.4. Data Transfer Size

The Transfer Size field (PDMCFG.XFERSIZE) of the PDMAC Control Register specifies the transfer size of each G-Bus transaction in a DMA transfer. The transfer size can be selected from one of the following: 1 DWORD, 1 QWORD, or 4 QWORD (Burst transfer).
1 QWORD or 4 QWORD can only be selected as the transfer size when the setting of the PDMAC G-Bus Address Register (PDMGA) and the PDMAC PCI Bus Address Register (PDMPA) is a 64-bit address boundary and the PDMAC Count Register (PDMCTR) setting is an 8-byte multiple. 1 DWORD must be selected as the transfer size in all other cases.

16.3.14. Error Detection, Interrupt Reporting

The PCI Controller reports the four following types of interrupts to the Interrupt Controller (IRC).

- *PDMAC Interrupt* *(Interrupt Number: 15, PDMAC)*
- *Power Management Interrupt* *(Interrupt Number: 38, PCI0PME)*
- *Error Detection Interrupt* *(Interrupt Number: 37, PCI0ERR)*

When each cause is detected, an interrupt is reported if the corresponding Status bit is set, and the corresponding Interrupt Enable Bit is set. The following tables list the name of each interrupt cause, the Status bit, and the Interrupt Enable bit. Please refer to the explanation of each Status bit for more information regarding each interrupt cause.

16.3.14.1. PDMAC Interrupts

Table 16-10 PDMAC Interrupts

Name	Status Bit		Interrupt Enable Bit	
Normal Chain Termination	PDMSTATUS	NCCMP	PDMCFG	NCCMPIE
Normal Data Transfer Termination		NTCMP		NTCMPIE
Inter-Transfer Stall Time Reached		STLTRF		ERRIE
Configuration Error		CFGERR		
PCI Fatal Error		PCIERR		
G-Bus Chain Error		CHNERR		
G-Bus Data Error		DATAERR		

16.3.14.2. Power Management Interrupts

Table 16-11 Power Management Interrupts

Name	Status Bit		Interrupt Enable Bit	
PME Detect	PCICSTATUS	PME	PCICMASK	PMEIE

16.3.14.3. Error Detection Interrupts

Table 16-12 Error Detection Interrupts

Name	Status Bit		Interrupt Enable Bit	
Parity Error Detect	PCISTATUS / PCISSTATUS	DPE	PCIMASK	DPEIE
System Error Report		SSE		SSEIE
Master Abort Receive		RMA		RMAIE
Target Abort Receive		RTA		RTAIE
Target Abort Report		STA		STAIE
Master Data Parity Error		MDPE		MDPEIE
TRDY Timeout Error	G2PSTATUS	IDTTOE	G2PMASK	IDTTOEIE
Retry Timeout Error		IDRTOE		IDRTOEIE
Broken Master Detect	PBASTATUS	BMD	PBAMASK	BMDIE
Long Burst Transfer Detect	PCICSTATUS	TLB	PCICMASK	TLBIE
Negative Increase Burst Transfer Detect		NIB		NIBIE
Zero Increase Burst Transfer Detect		ZIB		ZIBIE
PERR* Detect		PERR		PERRIE
SERR* Detect		SERR		SERRIE
G-Bus Bus Error Detect		GBE		GBEIE

16.3.15. PCI Bus Arbiter

Configuration settings during boot up select whether to use the on-chip PCI Bus arbiter (Internal PCI Bus Arbiter mode) or to use the External PCI Bus arbiter (External PCI Bus Arbiter mode).
When in the Internal PCI Bus Abiter mode, setting the PCI Bus Arbiter Enable bit (PBACFG.PBAEN) of the PCI Bus Arbiter Configuration Register starts operation.
The on-chip PCI Bus arbiter can arbitrate eight sets of PCI Bus usage requests from the Bus Master. Seven (7) ports are used: one for the PCI Controller bus master and six (6) for External Bus masters. The one remaining port is reserved for future expansion.

16.3.15.1. Request Signal, Grant Signal

The six external Bus Masters are connected to the REQ[5:0] signals and the GNT[5:0]* signals. Also, when in the External PCI Bus Master mode, the REQ[0]* signal becomes the PCI Bus Request Output signal and the GNT[0]* signal becomes the Bus Usage Permission Input Signal. Furthermore, the REQ[1]* signal can be used as an interrupt output signal to the external devices (see 14.3.7 for more information). This is explained in the Figure 16-12.

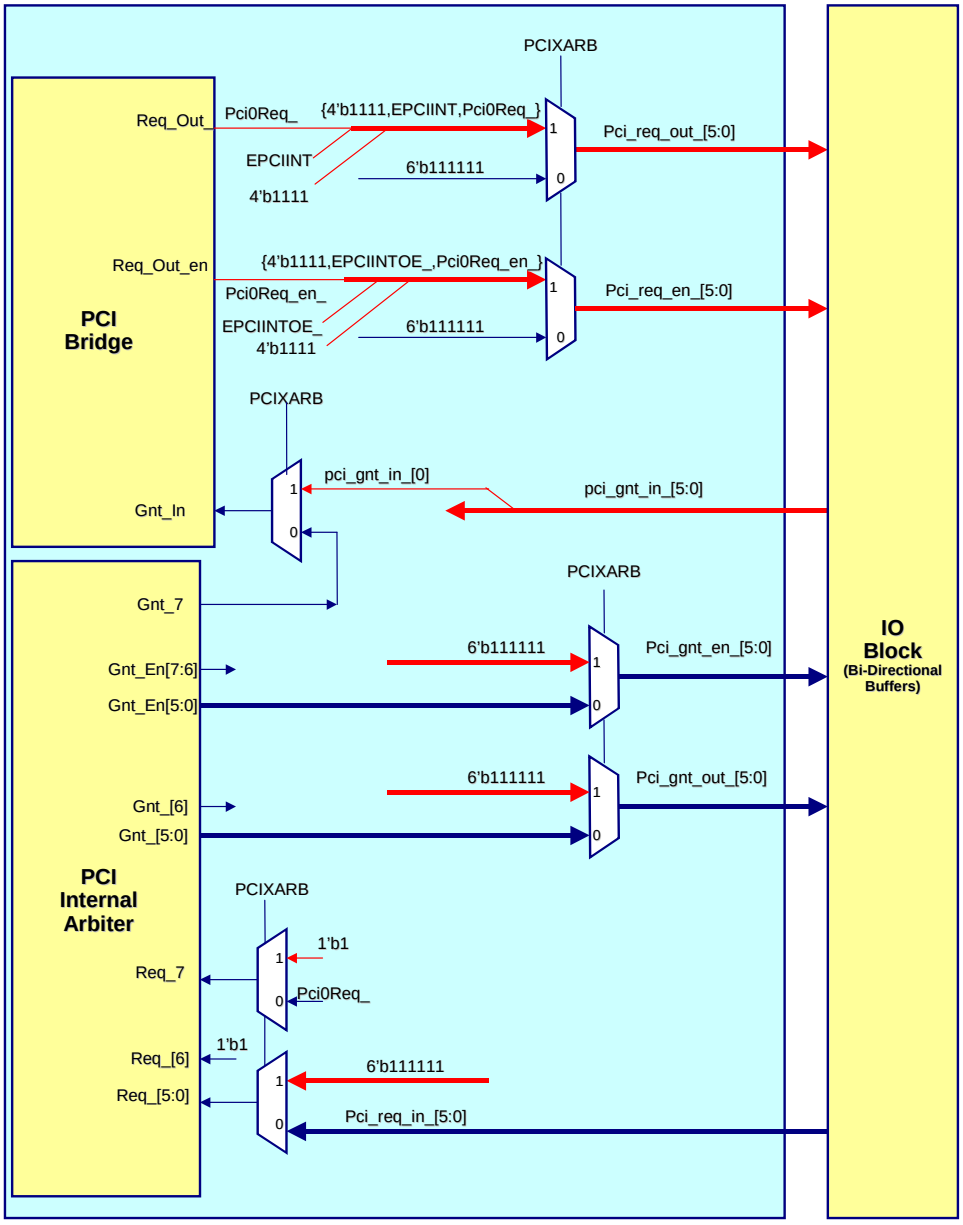


Figure 16-12 PCI Req/Gnt connections in Int/Ext Arbiter Mode

16.3.15.2. Priority Control

As illustrated below in Figure 16-13, a combination of two round-robin sequences is used as the arbitration algorithm that determines the priority of Internal PCI Bus arbiter bus requests. The round-robin with the lower priority (Level 2) consists of Masters W - Z, and the round-robin with the high priority (Level 1), consists of Master A - D. The PCI Bus Arbiter Request Port Register (PBAREQPORT) specifies whether to allocate the PCI Controller and the six External Bus Masters to Masters A - D or W - Z.

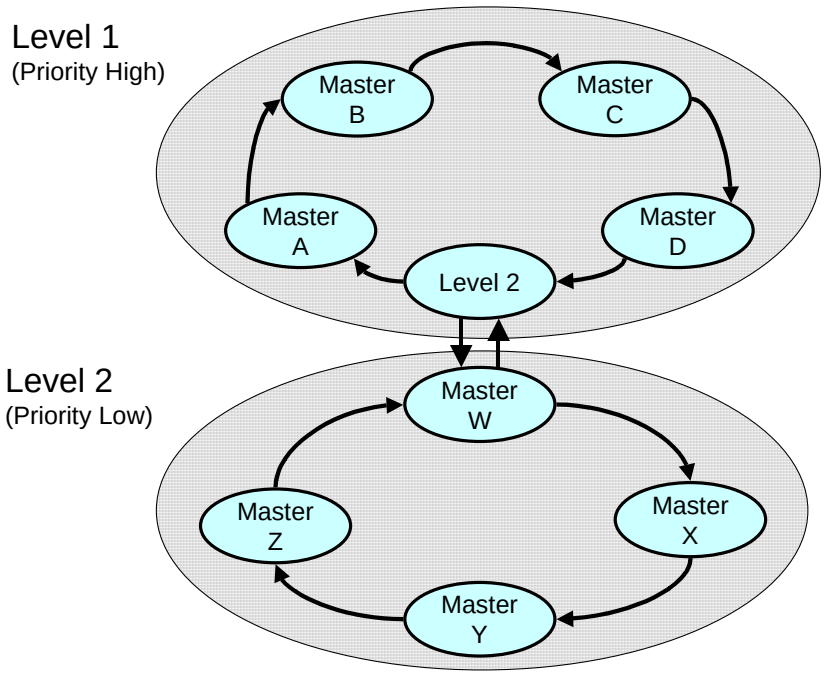


Figure 16-13 PCI Bus Arbitration Priority

The Bus Master priority is determined based on the Level 1 round-robin sequence. However, when Level 2 is used inside Level 1, the Level 2 Bus Master priority is determined based on the Level 2 round-robin sequence.
All 8 Bus Masters cannot be used on the TX4939. However, the Bus Master priority would be as follows if we assume there is a hypothetical device that can use all 8 Bus Masters and all 8 Bus Masters (Masters A – D, W – Z) simultaneously requested the bus.

→ A → B → C → D → W
→ A → B → C → D → X
→ A → B → C → D → Y
→ A → B → C → D → Z
→ A (returns to the beginning)

Since the priority can only transition in the order indicated by the above arrows (or the arrows in Figure 16-13, if we assume that the three Bus Masters A, B, and W exist, then Master B will obtain the bus first. If A and W then simultaneously request the bus, then PCI Bus ownership will transition in the order B → W → A.

16.3.15.3. Bus Parking

The On-chip PCI Bus Arbiter supports bus parking.
The last PCI Bus Master is made the Park Master when the Fix Park Master bit (FIXPM) of the PCI Bus Arbiter Configuration Register (PBACFG) is cleared (in the default state). When this bit is set, the Internal PCI Bus Arbiter Request A Port (Master A) becomes the Park Master.

16.3.15.4. Broken Master Detect

The TX4939 On-chip PCI Bus Arbiter has a function for automatically detecting broken masters.
If the PCI Bus Master requests and is granted the bus when the PCI Bus is in the Idle state, this master must assert the FRAME* signal within 16 PCI block cycles and start a transaction. The PCI Bus Arbiter recognizes any device that breaks this rule as a broken bus master and removes that device from the bus arbitration sequence.
This detection function is enabled when the Broken Master Check Enable bit (BMCEN) of the PCI Bus Arbiter Configuration Register (PBACFG) is set. When a broken master is detected, the Broken Master Detection bit (PBSTATUS.BMD) of the PCI Bus Arbiter Status Register is set and the bit in the PCI Bus Arbiter Broken Master Register (PBABM) that corresponds to that master is set. Then it also becomes possible to report an interrupt.

16.3.15.5. Special Programming

There may be some devices among PCI bus masters that operate differently from typical PCI devices. PCI devices with the following characteristics can be made usable by changing the programming of the PCI bus arbiter.

(1) Bus masters that can not re-assert REQ unless GNT is once deasserted after deasserting REQ
- *Assign the bus master to a request port other than Port A through the PBAREQPORT register (at 0xD100). (Assign the TX4939 to Port A.)*
- *Enable the Fixed Parked Master (FIXPA) bit in the PBACFG register (at 0xD104).*

(2) Bus masters that initiate a PCI transaction even when the deassertion of GNT has taken away their bus mastership before the start of the transaction
- *Assign the bus master to request port A, B, C or D through the PBAREQPORT register (at 0xD100).*

For example, a bus master with both of the above characteristics can be used by configuring the PCI bus arbiter as follows:
- *Set the internal PCI bus arbiter to the fixed parked master.*
- *Assign the TX4939 to request port A.*
- *Assign the bus master to request port B.*

If this bus master is connected to REQ[3] and broken master checking is to be enabled, values to be written to the PBACFG and PBAREQPORT registers are as follows:

PBACFG (at 0xD104):	0x0000000B
PBAREQPORT (at 0xD100):	0x73546210

16.3.16. Set Configuration Space

The access to the PCI Configuration space of external PCI devices is explained in section 16-7.
The registers in the PCI registers space (which also has registers corresponding to the standard PCI Configuration Space) inside the PCI Bridge can be accessed by doing simple GBus read/writes from a GBus master with the address corresponding to the PCI register space.

16.3.17. PCI Clock

The Clock of PCI Controller logic is designed as asynchronous from the GBUS clock. This internal clock is generated from the clock connected to the PCICLKIN terminal such way that the internal clock will have minimum skew from the inputted clock signal at PCICLKIN terminal.

16.4. PCI Controller Control Register

Table 16-13 lists the registers contained in the PCI Controller Control Register. Parentheses in the register names indicate the corresponding PCI Configuration Space Register.

Table 16-13 PCI Controller Control Register

Section	Address	Size	Mnemonic	Register Name
16.4.1	0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
16.4.2	0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
16.4.3	0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
16.4.4	0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
16.4.5	0xD010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
16.4.7	0xD014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
16.4.5	0xD018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
16.4.8	0xD01C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
16.4.5	0xD020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
16.4.9	0xD024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
16.4.10	0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
16.4.11	0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
16.4.12	0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
16.4.13	0xD040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
16.4.14	0xD080	32	G2PSTATUS	G2P Status Register
16.4.15	0xD084	32	G2PMASK	G2P Interrupt Mask Register
16.4.16	0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register
16.4.17	0xD08C	32	PCIMASK	PCI Status Interrupt Mask Register
16.4.18	0xD090	32	P2GCFG	P2G Configuration Register
16.4.19	0xD094	32	P2GSTATUS	P2G Status Register
16.4.20	0xD098	32	P2GMASK	P2G Interrupt Mask Register
16.4.21	0xD09C	32	P2GCCMD	P2G Current Command Register
16.4.22	0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
16.4.23	0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
16.4.24	0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
16.4.25	0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
16.4.26	0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
16.4.27	0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for diagnositics)
16.4.28	0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for diagnostics)
16.4.29	0xD11C	32	PBACSTATE	PCI Bus Arbiter Currrent State Register (for diagnostics)
16.4.30	0xD120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
16.4.31	0xD128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
16.4.32	0xD130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
16.4.33	0xD138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
16.4.34	0xD140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
16.4.35	0xD144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
16.4.36	0xD148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
16.4.37	0xD14C	32	G2PIOMASK	G2P I/O Space Address Mask Register
16.4.38	0xD150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
16.4.39	0xD158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
16.4.40	0xD160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
16.4.41	0xD168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
16.4.42	0xD170	32	PCICCFG	PCI Controller Configuration Register
16.4.43	0xD174	32	PCICSTATUS	PCI Controller Status Register
16.4.44	0xD178	32	PCICMASK	PCI Controller Interrupt Mask Register
16.4.45	0xD180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
16.4.46	0xD188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
16.4.47	0xD190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
16.4.48	0xD198	64	P2GIOGBASE	P2G I/O Space G-Bus Base Address Register
16.4.49	0xD1A0	32	G2PCFGADRS	G2P Configuration Address Register
16.4.50	0xD1A4	32	G2PCFGDATA	G2P Configuration Data Register
16.4.51	0xD1C8	32	G2PINTACK	G2P Interrupt Acknowldge Data Register
16.4.52	0xD1CC	32	G2PSPC	G2P Special Cycle Data Register
16.4.53	0xD1D0		PCICDATA0	Configuration Data 0 Register

Table 16-13 PCI Controller Control Register

Section	Address	Size	Mnemonic	Register Name
16.4.54	0xD1D4		PCICDATA1	Configuration Data 1 Register
16.4.55	0xD1D8		PCICDATA2	Configuration Data 2 Register
16.4.56	0xD1DC		PCICDATA3	Configuration Data 3 Register
16.4.6	0xD1E0	32	P2GM0CFG	P2G Memory Space 0 Configuration Register
16.4.6	0xD1E4	32	P2GM1CFG	P2G Memory Space 1 Configuration Register
16.4.6	0xD1E8	32	P2GM2CFG	P2G Memory Space 2 Configuration Register
16.4.57	0xD200	64	PDMCA	PDMAC Chain Address Register
16.4.58	0xD208	64	PDMGA	PDMAC G-Bus Address Register
16.4.59	0xD210	64	PDMPA	PDMAC PCI Bus Address Register
16.4.60	0xD218	64	PDMCTR	PDMAC Count Register
16.4.61	0xD220	64	PDMCFG	PDMAC Configuration Register
16.4.62	0xD228	64	PDMSTATUS	PDMAC Status Register

16.4.1. ID Register (PCIID)

The Device ID field corresponds to the Device ID Register in the PCI Configuration Space, and the Vendor ID field corresponds to the Vendor ID register of the PCI Configuration Space.

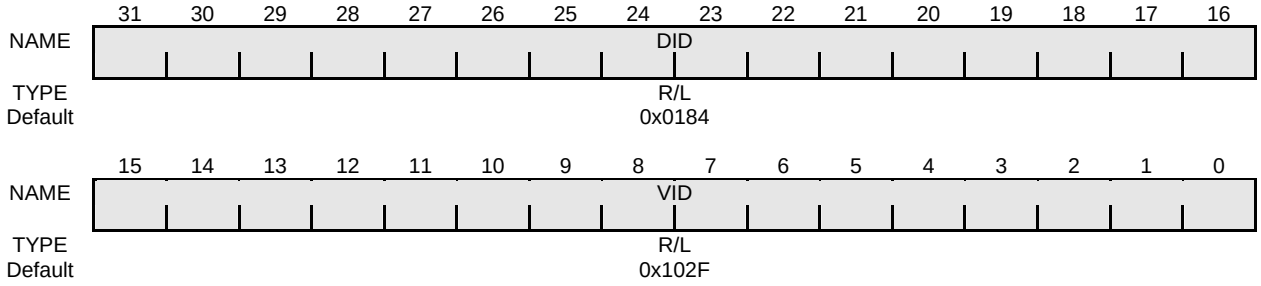


Figure 16-14 ID Registers

Table 16-14 ID Registers

Bits	Mnemonic	Field Name	Description	R/W
31:16	DID	Device ID	Device ID (Default: 0x0184/0x0185) This register indicates the ID that is allocated to a device. The ID can be changed by loading data from a configuration EEPROM during initialization. PCIC (external PCI bus controller) = 0x0184 PCIC1(internal PCI bus controller) = 0x0185	R/L
15:0	VID	Vendor ID	Vendor ID (Default: 0x102F) This register indicates the device product that is allocated by PCI SIG.	R/L

16.4.2. PCI Status, Command Register (PCISTATUS)

The upper 16 bits correspond to the Status Register in the PCI Configuration Space, and the lower 16 bits correspond to the Command Register in the PCI Configuration Space.

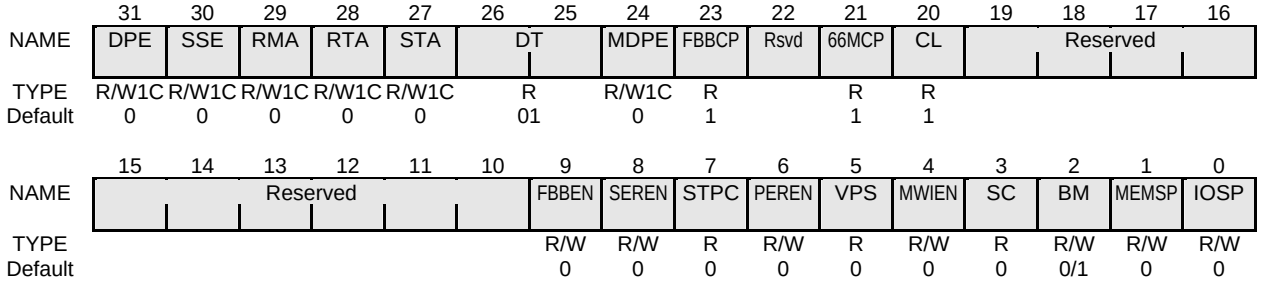


Figure 16-15 PCI Status, Command Register

Table 16-15 PCI Status, Command Register

Bit	Mnemonic	Field Name	Description	R/W
31	DPE	Detected Parity Error	Detected Parity Error (Default: 0) Indicates that a parity error was detected. A parity error is detected in the three following situtations: Detected a data parity error as the Read command PCI initiator. Detected a data parity error as the Write command PCI target. Detected an address parity error. This bit is set regardless of the setting of the Parity Error Response bit (PCISTATUS.PEREN) of the PCI Status, Command Register. 1: Detected a parity error. 0: Did not detect a parity error.	R/W1C
30	SSE	Signaled System Error	Signaled System Error (Default: 0) Detects either an address parity error or a special cycle data parity error. This bit is set when the SERR* signal is asserted. 1: Asserted the SERR* signal 0: Did not assert the SERR* signal.	R/W1C
29	RMA	Received Master Abort	Received Master Abort (Default: 0) This bit is set when a Master Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator (except for special cycles). 1: Transaction was aborted by a Master Abort. 0: Transaction was not aborted by a Master Abort.	R/W1C
28	RTA	Received Target Abort	Received Target Abort (Default: 0) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator. 1: Transaction was aborted by a Target Abort. 0: Transaction was not aborted by a Target Abort.	R/W1C
27	STA	Signaled Target Abort	Signaled Target Abort (Default: 0) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI target. 1: Bus transaction was aborted by a Target Abort. 0: Bus transaction was not aborted by a Target Abort.	R/W1C
26:25	DT	DEVSEL Timing	DEVSEL Timing (Fixed Value: 01) Three DEVSEL assert timings are defined in the PCI 2.2 Specifications: 00b = Fast; 01b = Medium; 10b = Slow; 11b = Reserved). With the exception of Read Configuration and Write Configuration, when the PCI Controller is the PCI target, the DEVSEL signal is asserted to a certain bus command and indicates the slowest speed for responding to the PCI Bus Master.	R
24	MDPE	Master Data Parity Error	Master Data Parity Error (Default: 0) Indicates the a parity error occurred when the PCI Controller is the PCI initiator. This bit is not set when the PCI Controller is the target. This bit is set when all of the three following conditions are met. It has been detected that the PERR* signal was set either directly or indirectly. The PCI Controller is the Bus Master for a PCI Bus transaction during which an error occurred. The Parity Error Response bit of the PCI Status Command Register (PCISTATUS.PEREN) has been set.	R/W1C
23	FBBCP	Fast Back-to-Back Capable	Fast Back-to-Back Capable (Fixed Value: 1) Indicates whether target access of a fast back-to-back transaction can be accepted. Is fixed to “1”.	R

Table 16-15 PCI Status, Command Register

Bit	Mnemonic	Field Name	Description	R/W
22		Reserved		—
21	66MCP	66 MHz Capable	66 MHz Capable (Fixed Value: 1) Indicates the 66 MHz operation is possible. Is fixed to “1”.	R
20	CL	Capabilities List	Capabilities List (Fixed Value: 1) Indicates that the capabilities list is being implemented. Is fixed to “1”.	R
19:10		Reserved		—
9	FBBEN	Fast Back-to-Back Enable	Fast Back-to-Back Enable (Default: 0) Indicates that issuing of fast back-to-back transactions has been enabled. 1: Enable 0: Disable	R/W
8	SEREN	SERR* Enable	SERR* Enable (Default: 0) Enables/Disables the SERR* signal. The SERR* signal reports that either a PCI Bus address parity error or a special cycle data parity error was detected. The SERR* signal is only asserted when the Parity Error Response bit is set and this bit is set. 1: Enable 0: Disable	R/W
7	STPC	Stepping Control	Stepping Control (Fixed Value: 0) Indicates that stepping control is not being supported.	R
6	PEREN	Parity Error Response	Parity Error Response (Default 0) Sets operation when a PCI address/data parity error is detected. A parity error response (either when the Parity Error Response bit (PCISTATUS.PEREN) of the PERR* Signal Assert or PCI Status, Command Register is set, or the SERR* signal is asserted) is performed only when this bit is set. When this bit is cleared, the PCI Controller ignores all parity errors and continues the transaction process as if the parity of that transaction was correct. 1: Parity error response is performed. 0: Parity error response is not performed.	R/W
5	VPS	VGA Palette Snoop	VGA Palette Snoop (Fixed Value: 0) Indicates that the VGA palette snoop function is not supported.	R
4	MWIEN	Memory Write and Invalidate Enable	Memory Write and Invalidate Enable (Default: 0) Controls whether to use the Memory Write and Invalidate command instead of the Memory Write command when the PCI Controller is the initiator.	R/W
3	SC	Special Cycles	Special Cycles (Fixed Value: 0) Indicates that special cycles will not be accepted as PCI targets.	R
2	BM	Bus Master	Bus Master (Default: 0/1) The default is only “1” when in the Host mode. 1: Operates as the Bus Master. 0: Does not operate as the Bus Master.	R/W
1	MEMSP	Memory Space	Memory Space (Default: 0) 1: Respond to PCI memory access. 0: Do not respond to PCI memory access.	R/W
0	IOSP	I/O Space	I/O Space (Default: 0) 1: Respond to PCI I/O access. 0: Do not respond to PCI I/O access.	R/W

16.4.3. Class Code, Revision ID Register (PCICCREV)

The Class Code field corresponds to the Class Code Register of the PCI Configuration Space, and the Revision ID field corresponds to the Revision ID Register of the PCI Configuration Space.

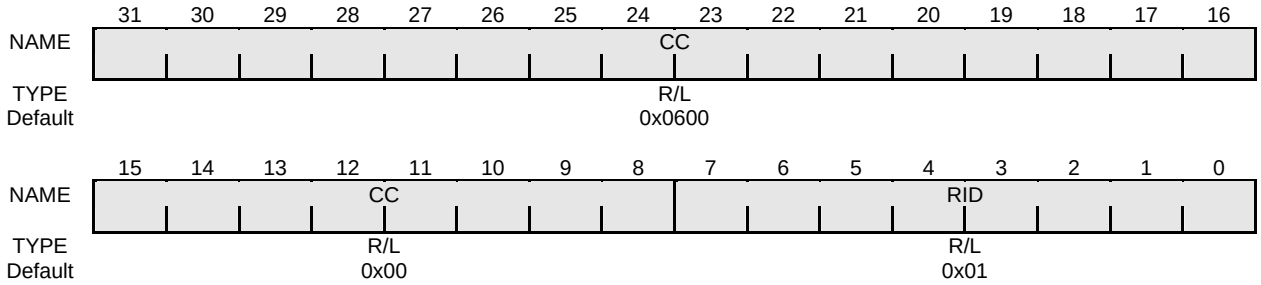


Figure 16-16 Class Code, Revision ID Register

Table 16-16 Class Code, Revision ID Register

Bits	Mnemonic	Field Name	Description	R/W
31:8	CC	Class Code	Class Code (Default: 0x060000) Classifies the device types. The default is 060000h, which defines the PCI Controller as a Host bridge device.	R/L
7:0	RID	Revision ID	Revision ID (Default: 0x01) Indicates the device revision ID. Please contact our Engineering Department for the exact value.	R/L

16.4.4. PCI Configuration 1 Register (PCICFG1) 0xD00C

The following fields correspond to the following registers.

- *BIST field → BIST Register of the PCI Configuration Space*
- *Header Type field → Header Type Register in the PCI Configuration Space*
- *Latency Timer field → Latency Timer Register of the PCI Configuration Space*
- *Cache Line Size field → Cache Line Size Register of the PCI Configuration Space.*

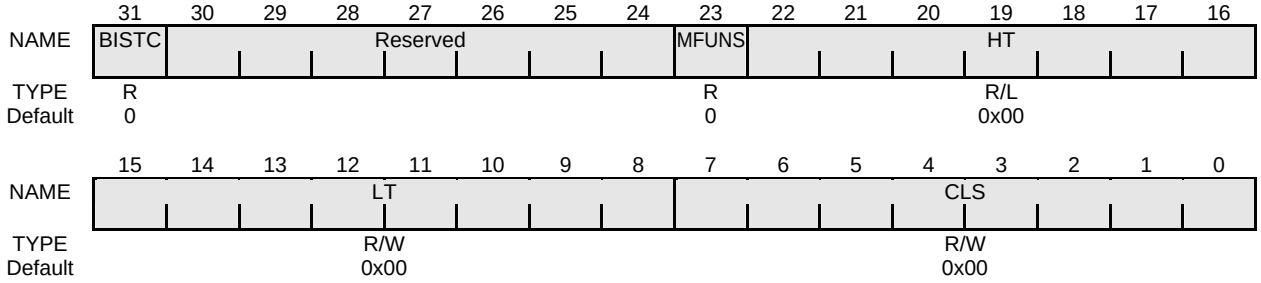


Figure 16-17 PCI Configuration 1 Register

Table 16-17 PCI Configuration 1 Register

Bit	Mnemonic	Field Name	Description	R/W
31	BISTC	BIST Capable	BIST Capable (Fixed Value: 0) Indicates that the BIST function is not being supported.	R
30:24		Reserved		—
23	MFUNS	Multi-Function	Multi-Function (Fixed Value: 0) 0: Indicates that the device is a single-function device.	R
22:16	HT	Header Type	Header Type (Default: 0x00) Indicates the Header type. 0000000: Header Type 0	R/L
15:8	LT	Latency Timer	Latency Timer (Default: 0x00) Sets the latency timer value. Specifies the PCI Bus clock count during which to abort access when the GNT* signal is deasserted during PCI access. Since the lower two bits are fixed to “0”, cycle counts can only be specified in multiples of 4.	R/W
7:0	CLS	Cache Line Size	Cache Line Size (Default: 0x00) Is used to select the PCI Bus command during a Burst Read transaction. See “16.3.6 Supported PCI Bus Commands” for more information.	R/W

16.4.5. P2G Memory Space (m) PCI Lower Base Address Register

P2GM0PLBASE, P2GM1PLBASE, P2GM2PBASE

This register corresponds to the Memory Space (m:m=0,1,2) Lower Base Address Register at offset address 0x10[18h,20h] of the PCI Configuration Space.
This register is accessible both Host Mode and Satellite Mode. In Satellite Mode this register is read only.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	BA [31:20]												RESERVED			
TYPE	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/O	R/O	R/O	R/O
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RESERVED												PF	TYPE		MSI
TYPE	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O		R/O
Default	0	0	0	0	0	0	0	0	0	0	0	0	1/0	00		0

Figure 16-18 P2G Memory Space (m) PCI Lower Base Address Register (m=0,1,2)

Table 16-18 P2G Memory Space (m) PCI Lower Base Address Register (m=0,1,2)

Bit	Mnemonic	Field Name	Description	Default	R/W
31:20	BA [31:20]	Base Address	Base Address [Host Mode] (Default=0x000) Sets the lower address of the PCI base address in Target Access Memory Space n. The size of Memory Space is defined by the value of P2GM(m)CFG.MSS. The default size of the three memory windows are, P2GM0PLBASE Default:512MB P2GM1PLBASE Default:16MB P2GM2PBASE Default:1MB Please see 16.4.6 for more details. Note : This bit field will be R/O in Satellite Mode.	0x000	R/W or R/O
19:4		RESERVED		0x0000	R/O
3	PF	Prefetchable	Prefetchable 1: Indicates that memory is prefetchable. 0: Indicates that memory is not prefetchable. The default value of PF for the three memory windows are P2GM0PLBASE Default:1 P2GM1PLBASE Default:1 P2GM2PBASE Default:0 Please see 16.4.6 for more details.	1/0	R/O
2:1	TYPE	Type	Type (Default: 00) 00: Indicates that an address is within a 32-bit address region	0	R/O
0	MSI	Memory Space	Memory Space Indicator (Fixed Value: 0) 0: Indicates that this Base Address Register is for use by the PCI Memory Space	0	R/O

16.4.6. P2G Memory Space (m) Configuration Register

P2GM0CFG, P2GM1CFG, P2GM2CFG

This register configures Memory Space of PCI memory windows (m=0,1,2) Lower Base Address Register at offset address 0x10[18h, 20h] of the PCI Configuration Space.

The size of PCI Memory Window is defined by the value of P2GM(m)CFG.MSS. The Effective Base Address (EBA) is given by following formula,

EBA [31:20] = P2GM(m)PLBASE.BA & P2GM(m)CFG.MSS (bitwise AND)

This register is accessible both in Host Mode and Satellite Mode.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RESERVED															
TYPE	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	MSS [31:29]			MSS [28:20]									PFCFG	RESERVED		
TYPE	R/O	R/O	R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/O
Default	1	1	1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	0	0	0

Figure 16-19 P2G Memory Space (m) Configuration (m=0,1,2)

Table 16-19 P2G Memory Space (m) Configuration (m=0,1,2)

Bit	Mnemonic	Field Name	Description	Default	R/W
31:16		RESERVED		0x0000	R/O
15:13	MSS [31:29]	Size (Maximum)	Memory Space Size (Maximum, Default=3'b111, 512 MB).	3'b111	R/O
12:4	MSS [28:20]	Size	Memory Space Size See Table 16-20 for other corresponding. The default values of MSS[28:20] for the three configuration registers are P2GM0CFG Default: 9'h000 P2GM1CFG Default: 9'h1F0 P2GM2CFG Default: 9'h1FF	1/0	R/W
3	PFCFG	Prefetchable Configuration	Prefetchable Configuration 1: Indicates that memory is prefetchable. 0: Indicates that memory is not prefetchable. The default values of PFCFG for the three configuration registers are P2GM0CFG Default: 1 P2GM1CFG Default: 1 P2GM2CFG Default:0	1/0	R/W
2:0		RESERVED		0x0	R/O

Table 16-20 Corresponding of Memory Space Size and MSS[31:20] Value

MSS[31:20]	31	30	29	28	27	26	25	24	23	22	21	20	n	Memory Size
0xE00	1	1	1	0	0	0	0	0	0	0	0	0	29	512 MB
0xF00	1	1	1	1	0	0	0	0	0	0	0	0	28	256 MB
0xF80	1	1	1	1	1	0	0	0	0	0	0	0	27	128 MB
0xFC0	1	1	1	1	1	1	0	0	0	0	0	0	26	64 MB
0xFE0	1	1	1	1	1	1	1	0	0	0	0	0	25	32 MB
0xFF0	1	1	1	1	1	1	1	1	0	0	0	0	24	16 MB
0xFF8	1	1	1	1	1	1	1	1	1	0	0	0	23	8 MB
0xFFC	1	1	1	1	1	1	1	1	1	1	0	0	22	4 MB
0xFFE	1	1	1	1	1	1	1	1	1	1	1	0	21	2 MB
0xFFF	1	1	1	1	1	1	1	1	1	1	1	1	20	1 MB

16.4.7. P2G Memory Space 0 PCI Upper Base Address Register (P2GM0PUBASE)

This register is unused since the PCI Controller does not support the target dual-address cycle. It is forbidden to write to this register.

16.4.8. P2G Memory Space 1 PCI Upper Base Address Register (P2GM1PUBASE)

This register is unused since the PCI Controller does not support the target dual-address cycle. It is forbidden to write to this register.

16.4.9. P2G I/O Space PCI Base Address Register (P2GIOPBASE)

This register corresponds to the I/O Space Base Address at offset address 0x24 of the PCI Configuration Space.
In case of Satellite Mode, this register will be read only.

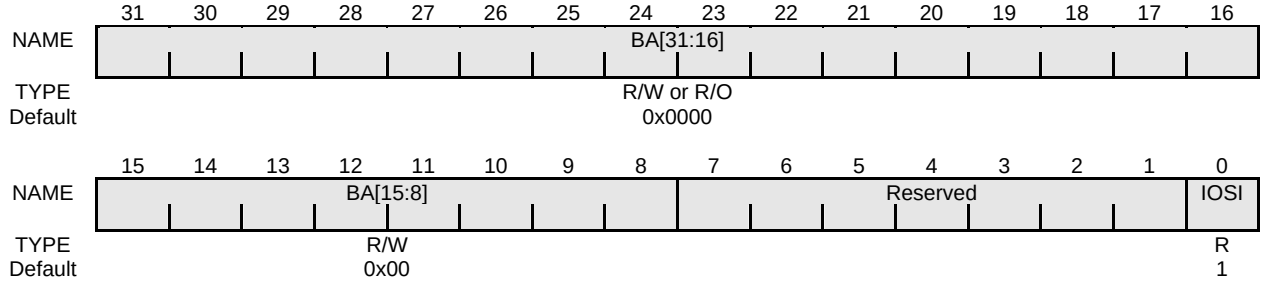


Figure 16-20 P2G I/O Space PCI Base Address Register

Table 16-21 P2G I/O Space PCI Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
31:8	BA[31:8]	Base Address	Base Address (Default: 0x00) Sets the PCI base address of the Target Access I/O Space. The size of this I/O space is fixed at 256 Bytes. In Satellite Mode, this field will be read only.	R/W or R/O
7:1		Reserved		—
0	IOSI	I/O Space	I/O Space Indicator (Fixed Value: 1) 1: Indicates that this Base Address Register is for use by the PCI I/O Space.	R

16.4.10. Subsystem ID Register (PCISID)

The Subsystem ID field corresponds to the Subsystem ID Register of the PCI Configuration Space, and the Subsystem Vendor ID field corresponds to the Subsystem Vendor ID Register of the PCI Configuration Space.

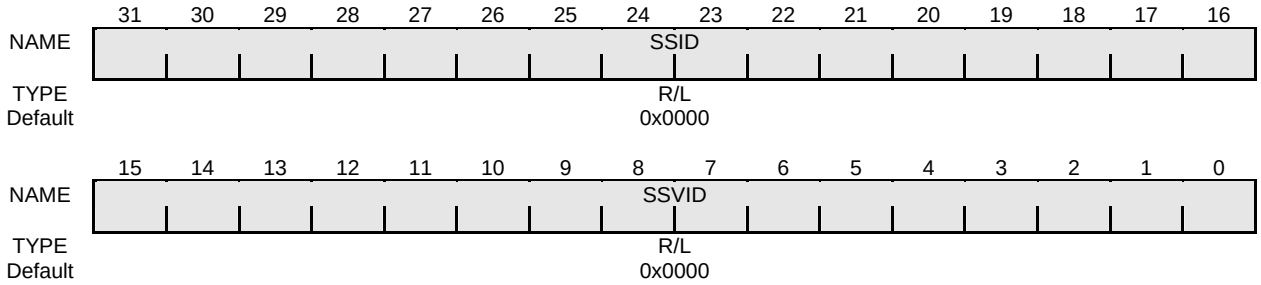


Figure 16-21 Subsystem ID Register

Table 16-22 Subsystem ID Register

Bits	Mnemonic	Field Name	Description	R/W
31:16	SSID	Subsystem ID	Subsystem ID (Default: 0x0000) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board.	R/L
15:0	SSVID	Subsystem Vendor ID	Subsystem Vendor ID (Default: 0x0000) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board.	R/L

16.4.11. Capabilities Pointer Register (PCICAPPTR)

The Capabilities Pointer field corresponds to the Capabilities Pointer Register of the PCI Configuration Space.

Figure 16-22 Capabilities Pointer Register

Table 16-23 Capabilities Pointer Register

Bits	Mnemonic	Field Name	Description	R/W
31:8		Reserved		—
7:0	CAPPTR	Capabilities Pointer	Capabilities Pointer (Fixed Value: 0xDC) Indicates as an offset value the starting address of the capabilities list that indicates extended functions.	R

16.4.12. PCI Configuration 2 Register (PCICFG2)

The following fields correspond to the following registers:

- *Max. Latency field → Max_Lat Register of the PCI Configuration Space*
- *Min. Grant field → Min_Gnt Register of the PCI Configuration Space*
- *Interrupt Pin field → Interrupt Pin Register of the PCI Configuration Space*
- *Interrupt Line field → Interrupt Line Register of the PCI Configuration Space*

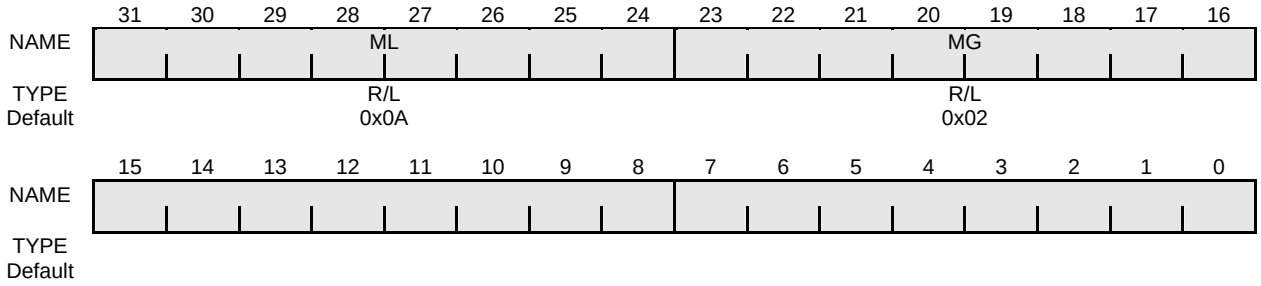


Figure 16-23 PCI Configuration 2 Register

Table 16-24 PCI Configuration 2 Register

Bits	Mnemonic	Field Name	Description	R/W
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Default: 0x0A) 00h: Does not use this register to determine PCI Bus priority. 01h-FFh: Specifies the time interval for requesting bus ownership. In units of 250 ns, assuming the PCICLK is 33 MHz.	R/L
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Default: 0x02) 00h: Is not used to calculate the latency timer value. 01h-FFh: Sets the time required for Burst transfer. In units of 250 ns, assuming the PCICLK is 33 MHz.	R/L
15:8	IP	Interrupt Pin	Interrupt Pin (Default: 0x01) Valid values: 00 - 04h 00h: Do not use interrupt signals. 01h: Use Interrupt signal INTA* 02h: Use Interrupt signal INTB* 03h: Use Interrupt signal INTC* 04h: Use Interrupt signal INTD* 05h - FFh: Reserved	R/L
7:0	IL	Interrupt Line	Interrupt Line (Default: 0x00) This is a readable/writable 8-bit register. The software uses this register to indicate information such as the interrupt signal connection information. Operation of the TX4939 is not affected.	R/W

16.4.13. G2P Timeout Count Register (G2PTOCNT)

The Retry Timeout field corresponds to the Retry Timeout Value Register of the PCI Configuration Space, and the TRDY Timeout field corresponds to the TRDY Timeout Value Register of the PCI Configuration Space.

Figure 16-24 G2P Timeout Count Register

Table 16-25 G2P Timeout Count Register

Bits	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15:8	RETRYTO	Retry Timeout	Retry Time Out (Default: 0x80) Sets the maximum number of retries to accept when operating as the initiator on the PCI Bus. Ends with an error when receiving more retry terminations than the set maximum number. Setting a “0” disables this timeout function. Note: Generally, disable retry time-out detection by setting this field to zero. Some PCI devices invoke more than 128 retries at normal times.	R/W
7:0	TRDYTO	TRDY Timeout	TRDY Time Out (Default: 0x80) Sets the maximum value of the time to wait for assertion of the TRDY* signal when operating as the initiator on the PCI Bus. Setting a “0” disables this timeout function. Note: Generally, disable TRDY time-out detection by setting this field to zero. Some PCI devices exhibit a TRDY delay longer than 128 PCI clocks at normal times.	R/W

16.4.14. G2P Status Register (G2PSTATUS)

Figure 16-25 G2P Status Register

Table 16-26 G2P Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:2		Reserved		—
1	IDTTOE	TRDY Timeout Error	Initiator Detected TRDY Time Out Error (Default: 0x0) This bit is set when the initiator detects a TRDY timeout.	R/W1C
0	IDRTOE	Retry Timeout Error	Initiator Detected Retry Time Out Error (Default: 0x0) This bit is set when the initiator detects a Retry timeout.	R/W1C

16.4.15. G2P Interrupt Mask Register (G2PMASK)

Figure 16-26 G2P Interrupt Mask Register

Table 16-27 G2P Interrupt Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:2		Reserved		—
1	IDTTOEIE	TRDY Timeout Error Interrupt Enable	Initiator Detected TRDY Time Out Interrupt Enable (Default: 0x0) The initiator generates an interrupt when it detects a TRDY timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
0	IDRTOEIE	Retry Timeout Error Interrupt Enable	Initiator Detected Retry Time Out Interrupt Enable (Default: 0x0) The initiator generates an interrupt when it detects a Retry timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W

16.4.16. Satellite Mode PCI Status Register (PCISSTATUS)

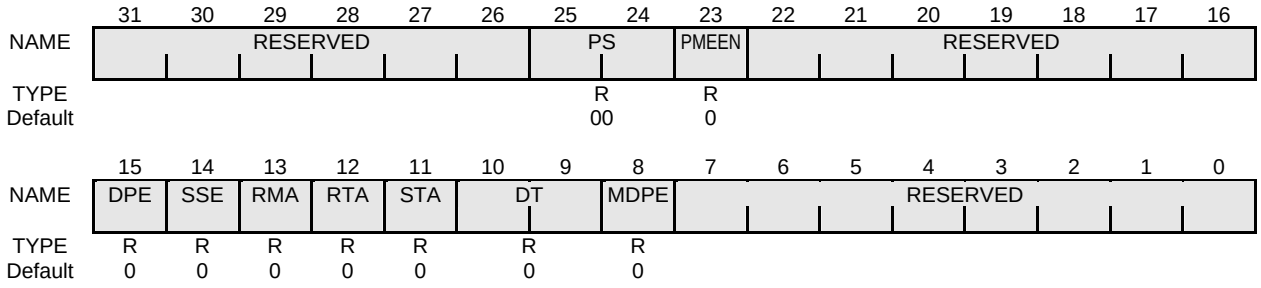


Figure 16-27 Satellite Mode PCI Status Register

Table 16-28 Satellite Mode PCI Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:24		Reserved		—
25:24	PS	Power State	PowerState (Default: 0x0) This is a shadow register of the PowerState field in the PMCSR Register.	R
23	PMEEN	PME Enable	PME_En (Default: 0x0) This is a shadow register of the PME_En bit of the PMCSR Register.	R
22:16		Reserved		—
15	DPE	Detected Parity Error	Detected Parity Error (Default: 0x0) This is a shadow register of the PCISTATUS.DPE bit.	R
14	SSE	Signaled System Error	Signaled System Error (Default: 0x0) This is a shadow register of the PCISTATUS.SSE bit.	R
13	RMA	Received Master Abort	Received Master Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RMA bit.	R
12	RTA	Received Target Abort	Received Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RTA bit.	R
11	STA	Signaled Target Abort	Signaled Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.STA bit.	R
10:9	DT	Set DEVSEL Timing	DEVSEL Timing (Fixed Value: 0x1) This is a shadow register of the PCISTATUS.DT field.	R
8	MDPE	Data Parity Detected	Master Data Parity Error Detected (Default: 0x0) This is a shadow register of the PCISTATUS.MDPE bit.	R
7:0		Reserved		—

16.4.17. PCI Status Interrupt Mask Register (PCIMASK)

Figure 16-28 PCI Status Interrupt Mask Register

Table 16-29 PCI Status Interrupt Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15	DPEIE	Detected Parity Error Interrupt Enable	Detected Parity Error Interrupt Enable (Default: 0x0) Generates an interrupt when a parity error is detected. Usually, this interrupt is masked and a Master Data Parity error signals the error to the system. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
14	SSEIE	Signaled System Error Interrupt Enable	Signaled System Error Interrupt Enable (Default: 0x0) Generates an interrupt when a system error is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
13	RMAIE	Received Master Abort Interrupt Enable	Received Master Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Master Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
12	RTAIE	Received Target Abort Interrupt Enable	Received Target Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Target Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
11	STAIE	Signaled Target Abort Interrupt Enable	Signaled Target Abort Interrupt Enable (Default: 0x0) Generates an interrupt when a Target Abort is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
13	RMA	Received Master Abort	Received Master Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RMA bit.	R/W
12	RTA	Received Target Abort	Received Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.RTA bit.	R/W
11	STA	Signaled Target Abort	Signaled Target Abort (Default: 0x0) This is a shadow register of the PCISTATUS.STA bit.	R/W
10:9		Reserved		—
8	MDPEIE	Master Data Parity Detected Interrupt Enable	Master Data Parity Detected Interrupt Enable (Default: 0x0) Generates an interrupt when data parity is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
7:0		Reserved		—

16.4.18. P2G Configuration Register (P2GCFG)

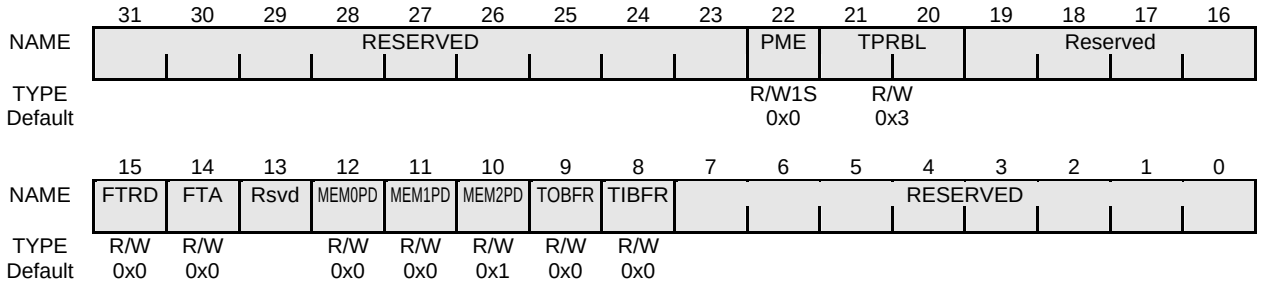


Figure 16-29 P2G Configuration Register

Table 16-30 P2G Configuration Register

Bit	Mnemonic	Field Name	Description	R/W
31:23		Reserved		—
22	PME	PME	PME (Default: 0x0) This bit is invalid since the PME* signal is an input signal.	R/W1S
21:20	TPRBL	Target Prefetch Read Burst Length	Target Prefetch Read Burst Length (Default: 0x3) These bits set the number of DWORDS (32-bit words) to be read into the data FIFO when prefetching is valid during a target memory Read operation. Extra data transferred to the data FIFO is deleted when performing a memory Read operation of a PCI Bus transfer that is smaller than the set size. This setting is invalid when prefetching is disabled. 0x00: Access and transfer each 2 DWORDs of data to the target read FIFO. 0x01: Access and transfer each 4 DWORDs of data to the target read FIFO. 0x10: Access and transfer each 6 DWORDs of data to the target read FIFO. 0x11: Access and transfer each 8 DWORDs of data to the target read FIFO.	R/W
19:16		Reserved		—
15	FTRD	Force Target Retry/Disconnect	Force Target Retry/Disconnect (Default: 0x0) The PCI Controller executes Retry Termination on a PCI Read access transaction if this bit is set to “1”. This is a diagnostic function.	R/W
14	FTA	Force Target Abort	Force Target Abort (Default: 0x0) The PCI Controller executes a Target Abort on a PCI Read access transaction if this bit is set to “1”. This is a diagnostic function.	R/W
13		Reserved		—
12	MEM0PD	Memory 0 Window Prefetch Disable	Memory 0 Window Prefetch Disable (Default: 0x0) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 0 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled.	R/W
11	MEM1PD	Memory 1 Window Prefetch Disable	Memory 1 Window Prefetch Disable (Default: 0x0) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 1 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change.	R/W
10	MEM2PD	Memory 2 Window Space Prefetch Disable	Memory 2 Window Prefetch Disable (Default: 0x1) Prefetching during a G-Bus Burst Read transfer cycle to the Memory 2 Space is disabled when this bit is set to “1”. PCI Burst Read transactions are not supported when prefetching is disabled. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change.	R/W
9	TOBFR	Target Out-Bound FIFO Reset	Target Out-Bound FIFO Reset (Default: 0x0) The PCI Controller flushes the CORE internal Target Out-Bound FIFO when “1” is written to this bit. This bit always reads out “0” when it is read. This is a diagnostic function.	R/W
8	TIBFR	Target In-Bound FIFO Reset	Target In-Bound FIFO Reset (Default: 0x0) The PCI Controller flushes the CORE internal Target In-Bound FIFO when “1” is written to this bit. This bit always read out “0” when it is read. This is a diagnostic function.	R/W
7:0		Reserved		—

16.4.19. P2G Status Register (P2GSTATUS)

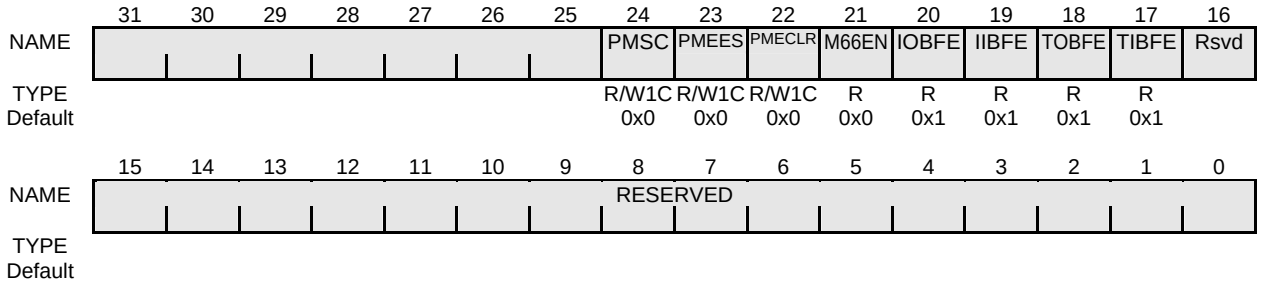


Figure 16-30 P2G Status Register

Table 16-31 P2G Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:25		Reserved		—
24	PMSC	PM State Change Detected	Power Management State Change (Default: 0x0) ***This bit is not valid in the TX4939.*** "1" is set to this bit when the PowerState field of the Power Management Register (PMCSR) is rewritten. This bit is cleared to "0" when a "1" is written to it.	R/W1C
23	PMEES	PME_En Set Detected	PME_En Set (Default: 0x0) ***This bit is not valid in the TX4939.*** This bit is set to "1" when the PME_En bit of the PMCSR Register is set to "1". When this bit is set, it indicates that the PCI Master (Host) device enabled PME* signal output. 1: Indicates that the PME_En bit is set. 0: Indicates that the PME_En bit is not set. This bit is cleared to "0" when a "1" is written to it.	R/W1C
22	PMECLR	PME Status Clear Detected	PME_Status Clear (Default: 0x0) ***This bit is not valid in the TX4939.*** This bit indicates that the PME_Status bit of the PMCSR Register was cleared. 1: Indicates that the PME_Status bit was cleared. 0: Indicates that the PME_Status bit was not cleared. This bit is cleared to "0" when a "1" is written to it.	R/W1C
21	M66EN	66 MHz Drive Status	M66EN Status (Default: 0x0) This bit indicates the current status of the M66EN signal. This bit can only be read. Writes to this bit are invalid. 1: The M66EN signal is asserted. 0: The M66EN signal is deasserted.	R
20	IOBFE	Initiator Out-Bound FIFO Empty	Initiator Out-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Initiator Out-Bound FIFO is empty. 0: Indicates that the Initiator Out-Bound FIFO is not empty. This is a diagnostic function.	R
19	IIBFE	Initiator In-Bound FIFO Empty	Initiator In-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Initiator In-Bound FIFO is empty. 0: Indicates that the Initiator In-Bound FIFO is not empty. This is a diagnostic function.	R
18	TOBFE	Target Out-Bound FIFO Empty	Target Out-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Target Out-Bound FIFO is empty. 0: Indicates that the Target Out-Bound FIFO is not empty. This is a diagnostic function.	R
17	TIBFE	Target In-Bound FIFO Empty	Target In-Bound FIFO Empty (Default: 0x1) 1: Indicates that the Target In-Bound FIFO is empty. 0: Indicates that the Target In-Bound FIFO is not empty. This is a diagnostic function.	R
16:0		Reserved		—

16.4.20. P2G Interrupt Mask Register (P2GMASK)

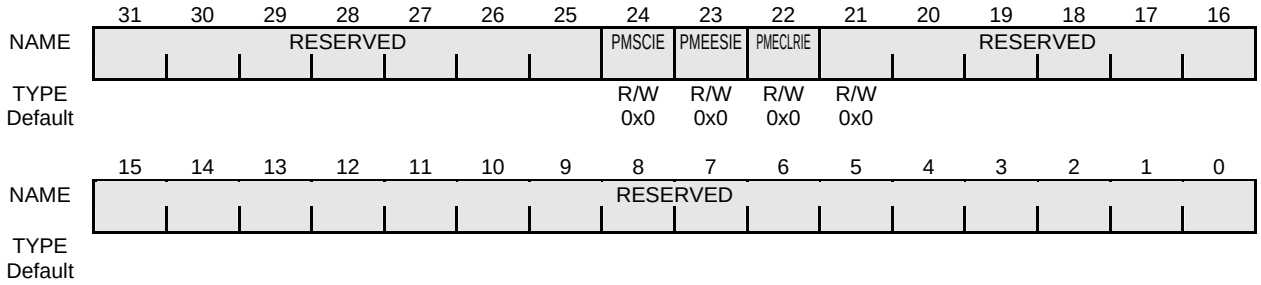


Figure 16-31 P2G Interrupt Mask Register

Table 16-32 P2G Interrupt Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:25		Reserved		—
24	PMSCIE	Power Management State Change Interrupt Enable	Power Management State Change Interrupt Enable (Default: 0x0) Generates an interrupt when the PowerState field of the Power Management Register (PMCSR) is rewritten. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
23	PMEESIE	PME_En Set Interrupt Enable	PME_En Set Interrupt Enable (Default: 0x0) Generates an interrupt when the PME_En bit of the PMCSR Register is set. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
22	PMECLRIE	PME Status Clear Interrupt Enable	PME_Status Clear Interrupt Enable (Default: 0x0) Generates an interrupt when the PME_Status bit of the PMCSR Register is cleared. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
21		Reserved		—
20:0		Reserved		—

16.4.21. P2G Current Command Register (P2GCCMD)

Figure 16-32 P2G Current Command Register

Table 16-33 P2G Current Command Register

Bits	Mnemonic	Field Name	Description	R/W
31:4		Reserved		—
3:0	TCCMD	Target Current Command Register	Target Current Command (Default: 0x9) Indicates the PCI command within the target access process that is currently in progress. This is a diagnostic function.	R

16.4.22. PCI Bus Arbiter Request Port Register

(PBAREQPORT)

This register sets the correlation between each PCI Bus request source (PCI Controller and REQ[5:0]) and each Internal PCI Bus Arbiter Request port (Master A - D, W - Z) (see Figure 16-13).
When changing these settings, each of the eight field values must always be set to different values. After changing this register, the Broken Master Register (BM) value becomes invalid since the bit mapping changes.
This register is only valid when using the on-chip PCI Bus Arbiter.

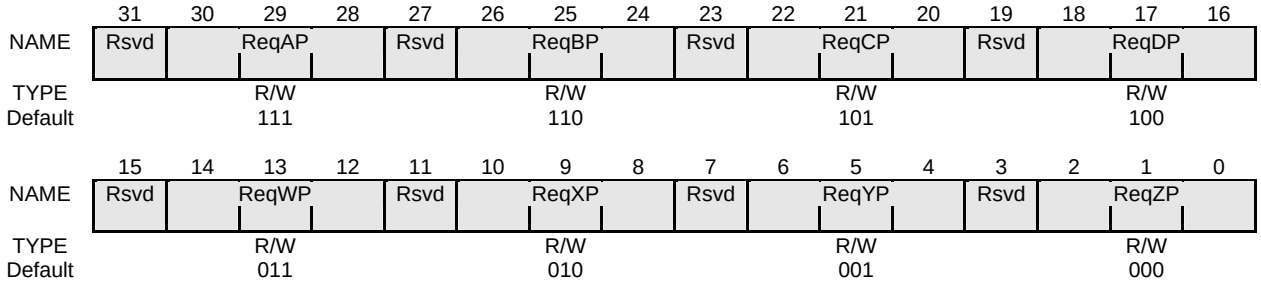


Figure 16-33 PCI Bus Arbiter Request Port Register

Table 16-34 PCI Bus Arbiter Request Port Register

Bit	Mnemonic	Field Name	Description	R/W
31		Rsvd		—
30:28	ReqAP	Request A Port	Request A Port (Default: 111) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request A Port (Master A). 111: Makes the PCI Controller Master A. 110: Rsvd 101: Makes REQ*[5] Master A. 100: Makes REQ*[4] Master A. 011: Makes REQ*[3] Master A. 010: Makes REQ*[2] Master A. 001: Makes REQ*[1] Master A. 000: Makes REQ*[0] Master A.	R/W
27		Rsvd		—
26:24	ReqBP	Request B Port	Request B Port (Default: 110) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request B Port (Master B). 111: Makes the PCI Controller Master B. 110: Rsvd 101: Makes REQ*[5] Master B 100: Makes REQ*[4] Master B 011: Makes REQ*[3] Master B 010: Makes REQ*[2] Master B. 001: Makes REQ*[1] Master B. 000: Makes REQ*[0] Master B.	R/W
23		Rsvd		—
22:20	ReqCP	Request C Port	Request C Port (Default: 101) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request C Port (Master C). 111: Makes the PCI Controller Master C. 110: Rsvd 101: Makes REQ*[5] Master C 100: Makes REQ*[4] Master C. 011: Makes REQ*[3] Master C. 010: Makes REQ*[2] Master C. 001: Makes REQ*[1] Master C. 000: Makes REQ*[0] Master C.	R/W
19		Rsvd		—

Table 16-34 PCI Bus Arbiter Request Port Register

Bit	Mnemonic	Field Name	Description	R/W
18:16	ReqDP	Request D Port	Request D Port (Default: 100) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request D Port (Master D). 111: Makes the PCI Controller Master D. 110: Rsvd 101: Makes REQ*[5] Master D. 100: Makes REQ*[4] Master D. 011: Makes REQ*[3] Master D. 010: Makes REQ*[2] Master D. 001: Makes REQ*[1] Master D. 000: Makes REQ*[0] Master D.	R/W
15		Rsvd		—
14:12	ReqWP	Request W Port	Request W Port (Default: 011) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request W Port (Master W). 111: Makes the PCI Controller Master W. 110: Rsvd 101: Makes REQ*[5] Master . 100: Makes REQ*[4] Master W. 011: Makes REQ*[3] Master W. 010: Makes REQ*[2] Master W. 001: Makes REQ*[1] Master W. 000: Makes REQ*[0] Master W.	R/W
11		Rsvd		—
10:8	ReqXP	Request X Port	Request X Port (Default: 010) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request X Port (Master X). 111: Makes the PCI Controller Master X. 110: Rsvd 101: Makes REQ*[5] Master X. 100: Makes REQ*[4] Master X. 011: Makes REQ*[3] Master X. 010: Makes REQ*[2] Master X. 001: Makes REQ*[1] Master X. 000: Makes REQ*[0] Master X.	R/W
7		Rsvd		—
6:4	ReqYP	Request Y Port	Request Y Port (Default: 001) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Y Port (Master Y). 111: Makes the PCI Controller Master Y. 110: Rsvd 101: Makes REQ*[5] Master Y. 100: Makes REQ*[4] Master Y. 011: Makes REQ*[3] Master Y. 010: Makes REQ*[2] Master Y. 001: Makes REQ*[1] Master Y. 000: Makes REQ*[0] Master Y.	R/W
3		Rsvd		—
2:0	ReqZP	Request Z Port	Request Z Port (Default: 000) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Z Port (Master Z). 111: Makes the PCI Controller Master Z. 110: Rsvd 101: Makes REQ*[5] Master Z. 100: Makes REQ*[4] Master Z. 011: Makes REQ*[3] Master Z. 010: Makes REQ*[2] Master Z. 001: Makes REQ*[1] Master Z. 000: Makes REQ*[0] Master Z.	R/W

16.4.23. PCI Bus Arbiter Configuration Register (PBACFG)

This register is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-34 PCI Bus Arbiter Configuration Register

Table 16-35 PCI Bus Arbiter Configuration Register

Bit	Mnemonic	Field Name	Description	R/W
31:4		Rsvd		—
3	FIXPA	Fixed Park Master	Fixed Park Master (Default: 0) Selects the method for determining the Park Master. 0: The last Bus Master becomes the Park Master. 1: Internal PCI Bus Arbiter Request Port A is the Park Master.	R/W
2	RPBA	Reset PCI Bus Arbiter	Reset PCI Bus Arbiter (Default: 0) Resets the PCI Bus Arbiter. However, the PCI Bus Arbiter Register settings are saved. Please use the software to clear this bit. 1: The PCI Bus Arbiter is currently being reset. 0: The PCI Bus Arbiter is not currently being reset.	R/W
1	PBAEN	PCI Bus Arbiter Enable	PCI Bus Arbiter Enable (Default: 0) This is the Bus Arbiter Enable bit. After Reset, External PCI Bus requests to the PCI Arbiter cannot be accepted until this bit is set to “1”. The PCI Controller is the default Parking Master after Reset. 1: Enables the PCI Bus Arbiter. 0: Disables the PCI Bus Arbiter.	R/W
0	BMCEN	Broken Master Check Enable	Broken Master Check Enable (Default: 0) Controls Broken Master detection. 1: Enables the Broken PCI Bus Master check. 0: Disables the Broken PCI Bus Master check.	R/W

16.4.24. PCI Bus Arbiter Status Register (PBASTATUS)

This register is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-35 PCI Bus Arbiter Status Register

Table 16-36 PCI Bus Arbiter Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:1		Rsvd		—
0	BM	Broken Master Detected	Broken Master Detected (Default: 0) This bit indicates that a Broken Master was detected. This bit is set to “1” if even one of the bits in the PCI Bus Arbiter Broken Master Register (PBABM) is “1”. 1: Indicates that a Broken Master was detected. 0: Indicates that no Broken Master has been detected.	R/W1C

16.4.25. PCI Bus Arbiter Interrupt Mask Register (PBAMASK)

This register is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-36 PCI Bus Arbiter Interrupt Mask Register

Table 16-37 PCI Bus Arbiter Interrupt Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:1		Rsvd		—
0	BMIE	Broken Master Detected Interrupt Enable	Broken Master Detected Interrupt Enable (Default: 0) Generates an interrupt when a Broken Master is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W

16.4.26. PCI Bus Arbiter Broken Master Register (PBABM)

This register indicates the acknowledged Broken Master. This register sets the bit that corresponds to the PCI Master device that was acknowledged as the Broken Master when the Broken Master Check Enable bit (BMCEN) in the PCI Bus Arbiter Configuration Register (PBACFG) is set.
Regardless of the value of the Broken Master Check Enable bit, a PCI Master device is removed from the arbitration scheme when “1” is written to the corresponding BM bit.
This register must be cleared to “0” since bit mapping changes, making this register value invalid when the PCI Bus Arbiter Request Port Register (PBAREQPORT) is changed.
This register is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-37 PCI Bus Arbiter Broken Master Register

Table 16-38 PCI Bus Arbiter Broken Master Register

Bit	Mnemonic	Field Name	Description	R/W
31:8		Rsvd		—
7	BM_A	Broken Master	Broken Master A (Default: 0) Indicates whether PCI Bus Master A is a Broken Master. 1: PCI Bus Master A was acknowledged as a Broken Master. 0: PCI Bus Master A was not acknowledged as a Broken Master.	R/W
6	BM_B	Broken Master	Broken Master B (Default: 0) Indicates whether PCI Bus Master B is a Broken Master. 1: PCI Bus Master B was acknowledged as a Broken Master. 0: PCI Bus Master B was not acknowledged as a Broken Master.	R/W
5	BM_C	Broken Master	Broken Master C (Default: 0) Indicates whether PCI Bus Master C is a Broken Master. 1: PCI Bus Master C was acknowledged as a Broken Master. 0: PCI Bus Master C was not acknowledged as a Broken Master.	R/W
4	BM_D	Broken Master	Broken Master D (Default: 0) Indicates whether PCI Bus Master D is a Broken Master. 1: PCI Bus Master D was acknowledged as a Broken Master. 0: PCI Bus Master D was not acknowledged as a Broken Master.	R/W
3	BM_W	Broken Master	Broken Master W (Default: 0) Indicates whether PCI Bus Master W is a Broken Master. 1: PCI Bus Master W was acknowledged as a Broken Master. 0: PCI Bus Master W was not acknowledged as a Broken Master.	R/W
2	BM_X	Broken Master	Broken Master X (Default: 0) Indicates whether PCI Bus Master X is a Broken Master. 1: PCI Bus Master X was acknowledged as a Broken Master. 0: PCI Bus Master X was not acknowledged as a Broken Master.	R/W
1	BM_Y	Broken Master	Broken Master Y (Default: 0) Indicates whether PCI Bus Master Y is a Broken Master. 1: PCI Bus Master Y was acknowledged as a Broken Master. 0: PCI Bus Master Y was not acknowledged as a Broken Master.	R/W
0	BM_Z	Broken Master	Broken Master Z (Default: 0) Indicates whether PCI Bus Master Z is a Broken Master. 1: PCI Bus Master Z was acknowledged as a Broken Master. 0: PCI Bus Master Z was not acknowledged as a Broken Master.	R/W

16.4.27. PCI Bus Arbiter Current Request Register (PBACREQ)

This register is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-38 PCI Bus Arbiter Current Request Register

Table 16-39 PCI Bus Arbiter Current Request Register

Bits	Mnemonic	Field Name	Description	R/W
31:8		Rsvd		—
7:0	CPCIBRS	Current PCI Bus Request Status	Current PCI Bus Request Status (Default: 0x00) This register indicates the status of the current PCI Bus Request Input Signal (PCI Controller and REQ*[5:0]). CPCIBRS[7] corresponds to the PCI Controller and CPCIBRS[5:0] correspond to REQ*[5:0].	R/W

16.4.28. PCI Bus Arbiter Current Grant Register (PBACGNT)

This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

Figure 16-39 PCI Bus Arbiter Current Grant Register

Table 16-40 PCI Bus Arbiter Current Grant Register

Bits	Mnemonic	Field Name	Description	R/W
31:8		Rsvd		—
7:0	CPCIBGS	Current PCI Grant Status	Current PCI Bus Grant Status (Default: 0x80) This register indicates the current PCI Bus Grant output signal (PCI Controller and GNT*[5:0]). CPCIBGS[7] corresponds to the PCI Controller, and CPCIBGS[5:0] correspond to GNT*[5:0].	R/W

16.4.29. PCI Bus Arbiter Current State Register (PBACSTATE)

This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

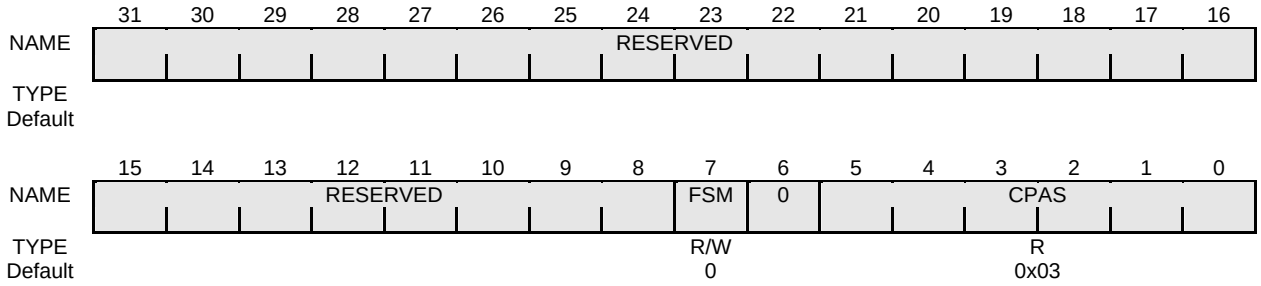


Figure 16-40 PCI Bus Arbiter Current State Registe

Table 16-41 PCI Bus Arbiter Current State Registe

Bit	Mnemonic	Field Name	Description	R/W
31:8		Rsvd		—
7	FSM	Observe PCI Arbiter State Machine	Observe PCI Arbiter Finite State Machine (Default: 0) Specifies which State Machine to observe. 1: Observe the Level 1 State Machine. 0: Observe the Level 2 State Machine.	R/W
6		Rsvd		—
5:0	CPAS	Current PCI Bus Arbiter State	Current PCI Bus Arbiter State (Default: 0x03) Displays the State Machine that was selected by the FSM bit. Please refer to Figure 16-13 for an explanation of Arbitration Flow. When FSM =1: 0x00: Preparation state for transferring bus ownership to PCI Agent A. 0x01: State in which Grant A is provided to PCI Agent A when PCI Bus ownership is being held elsewhere. 0x02: State in which Grant A is provided to PCI Agent A when PCI Bus ownership is not being held elsewhere. 0x03: The agent that was provided Grant A exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. 0x04 – 0x07: Same as 0x00 – 0x03 but for PCI Agent B 0x08 – 0x0B: Same as 0x00 – 0x03 but for PCI Agent C 0x0C – 0x0F: Same as 0x00 – 0x03 but for PCI Agent D 0x10 : Preparation state for transferring bus ownership to PCI Agent Level 2. 0x11: State in which Grant Level 2 is provided to PCI Agent Level 2 when PCI Bus ownership is being held elsewhere. 0x12: State in which Grant Level 2 is provided to PCI Agent Level 2 when PCI Bus ownership is not being held elsewhere. 0x13: The agent that was provided Grant Level 2 exists in this state. If there is bus ownership, the PCI Bus Arbiter transfers bus ownership to another agent. When FSM=0, the FSM=1 description is replaced as follows: A→W, B→X, C→Y, D→Z, Level 2→N/A.	R

16.4.30. G2P Memory Space 0 G-Bus Base Address Register (G2PM0GBASE)

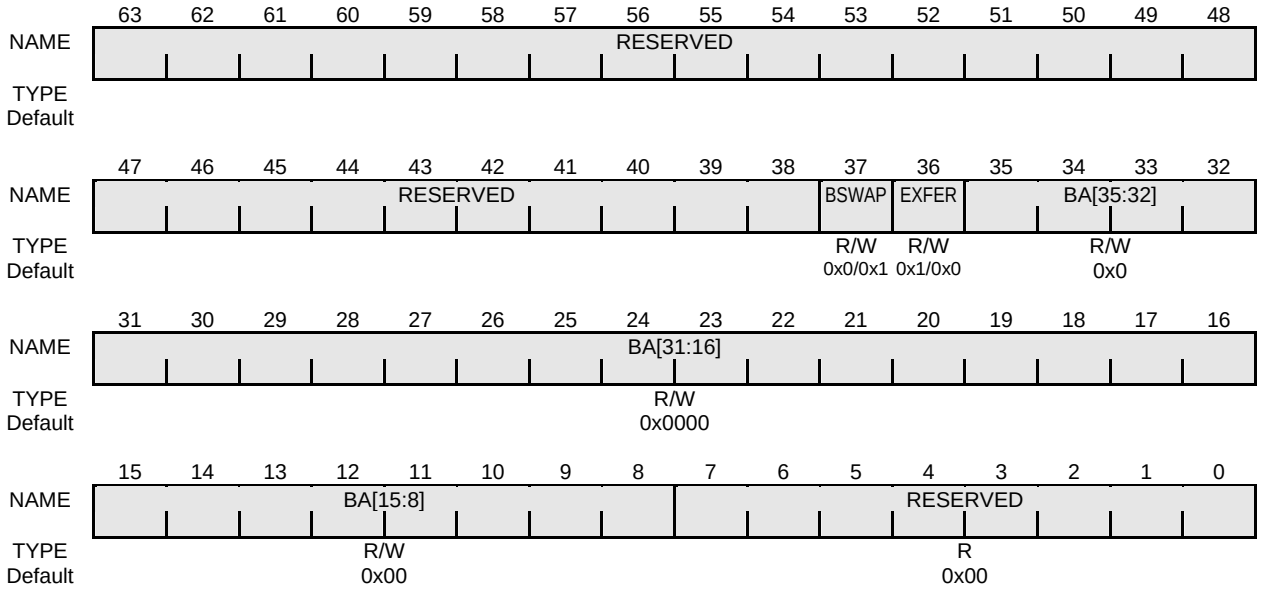


Figure 16-41 G2P Memory Space 0 G-Bus Base Address Register

Table 16-42 G2P Memory Space 0 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:38		Rsvd		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "1" when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:8	BA[35:8]	Base Address	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of Memory Space 0 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Rsvd		R

16.4.31. G2P Memory Space 1 G-Bus Base Address Register (G2PM1GBASE)

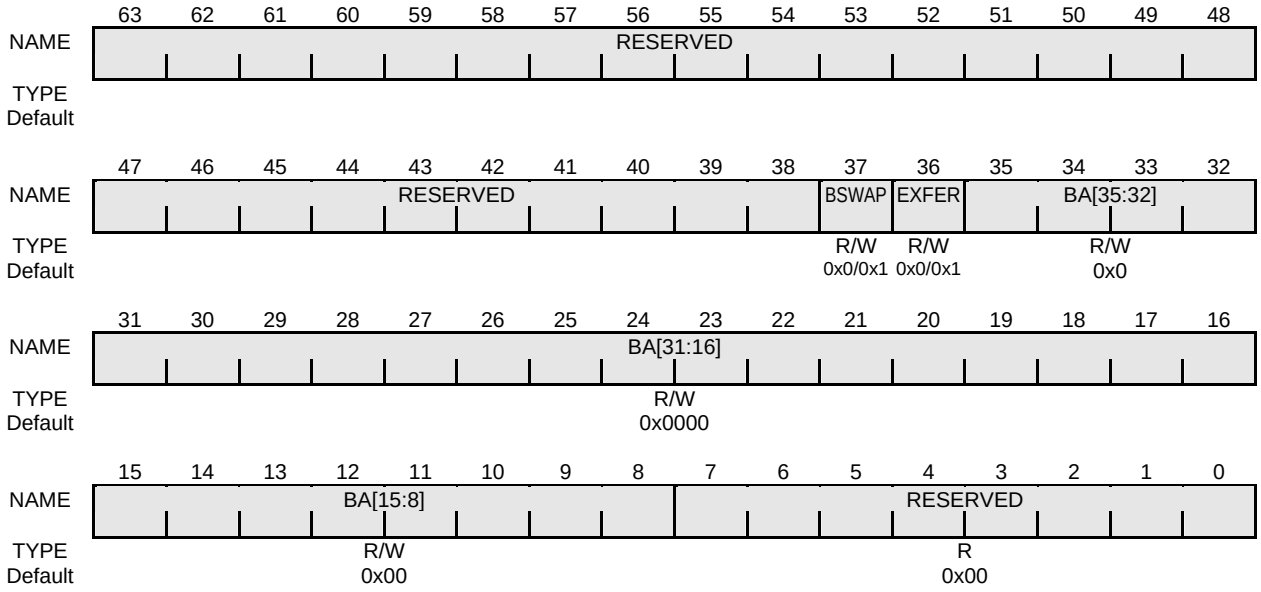


Figure 16-42 G2P Memory Space 1 G-Bus Base Address Register

Table 16-43 G2P Memory Space 1 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:38		Rsvd		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 1. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "1" when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 1. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:8	BA[35:8]	Memory Space Base Address 1	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of Memory Space 1 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Rsvd		R

16.4.32. G2P Memory Space 2 G-Bus Base Address Register (G2PM2GBASE)

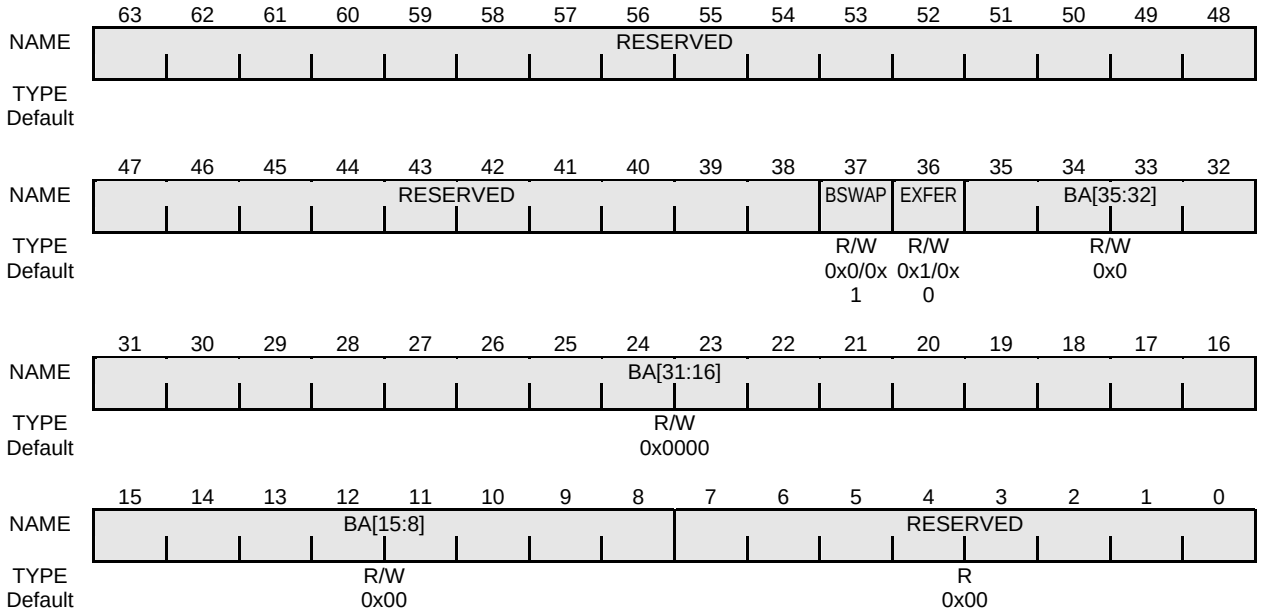


Figure 16-43 G2P Memory Space 2 G-Bus Base Address Register

Table 16-44 G2P Memory Space 2 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:38		Rsvd		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:8	BA[35:8]	Base Address	Base Address Default: 0x0_0000_00 (not PCI boot mode) Default:0x0_1fc00_00 (PCI boot mode) Sets the G-Bus base bus address of Memory Space 2 for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Rsvd		R

16.4.33. G2P I/O Space G-Bus Base Address Register (G2PIOGBASE)

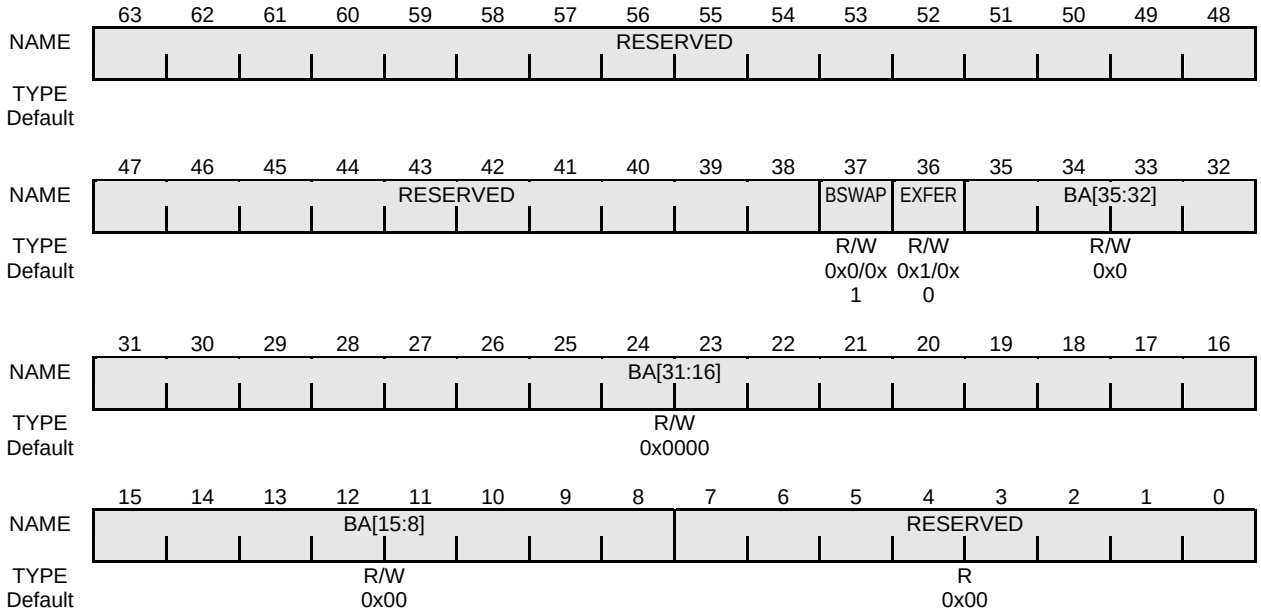


Figure 16-44 G2P I/O Space G-Bus Address Register

Table 16-45 G2P I/O Space G-Bus Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:38		Rsvd		—
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of the I/O space. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to the I/O Memory Space through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of the I/O Space. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:8	BA[35:8]	Base Address	Base Address (Default: 0x0_0000_00) Sets the G-Bus base bus address of the I/O Memory Space for initiator access. Can set the base address in 256-byte units.	R/W
7:0		Rsvd		R

16.4.34. G2P Memory Space 0 Address Mask Register (G2PM0MASK)

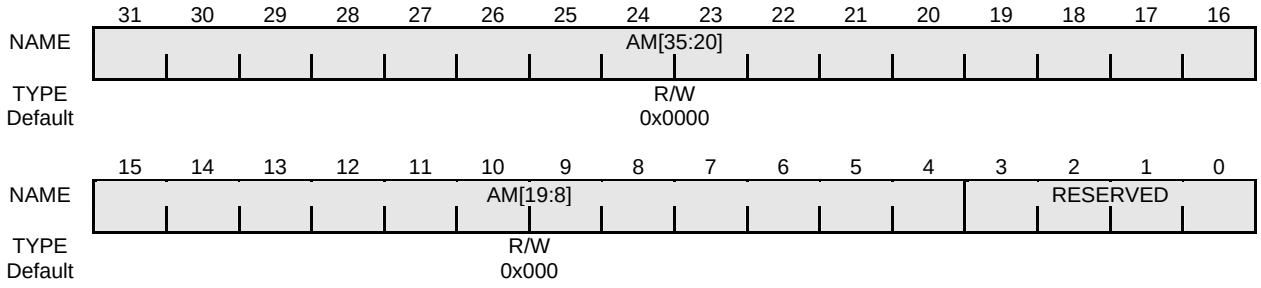


Figure 16-45 G2P Memory Space 0 Address Mask Register

Table 16-46 G2P Memory Space 0 Address Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 16.3.7 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value of G2PM0MASK[31:0] becomes 0x00FF_FFF0.	R/W
3:0		Rsvd		R

16.4.35. G2P Memory Space 1 Address Mask Register (G2PM1MASK)

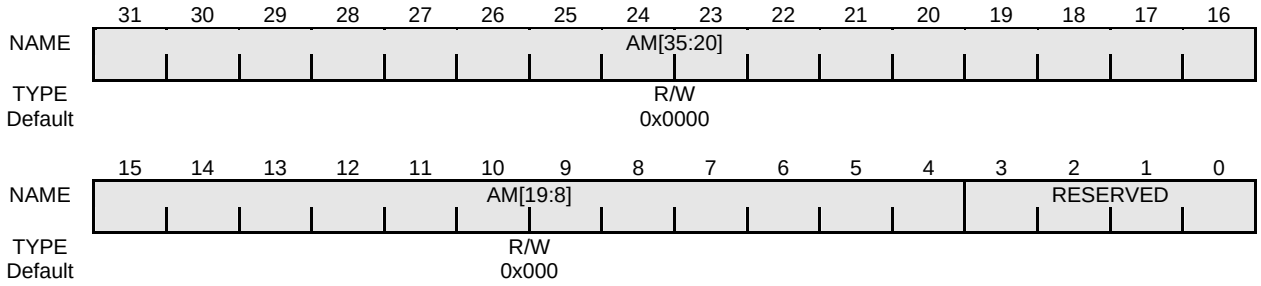


Figure 16-46 G2P Memory Space 1 Address Mask Register

Table 16-47 G2P Memory Space 1 Address Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 16.3.7 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value of G2PM1MASK[31:0] becomes 0x00FF_FFF0.	R/W
3:0		Rsvd		R

16.4.36. G2P Memory Space 2 Address Mask Register (G2PM2MASK)

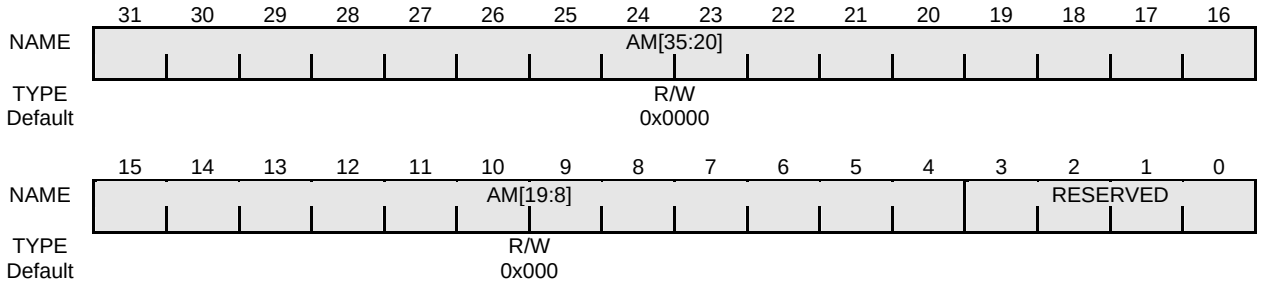


Figure 16-47 G2P Memory Space 2 Address Mask Register

Table 16-48 G2P Memory Space 2 Address Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask Default: 0x0_0000_00 (not PCI boot mode Default: 0x0_0001_ff (PCI boot mode) Sets the bits to be subject to address comparison. See 16.3.7 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value of G2PM2MASK[31:0] becomes 0x00FF_FFF0.	R/W
3:0		Rsvd		R

16.4.37. G2P I/O Space Address Mask Register (G2PIOMASK)

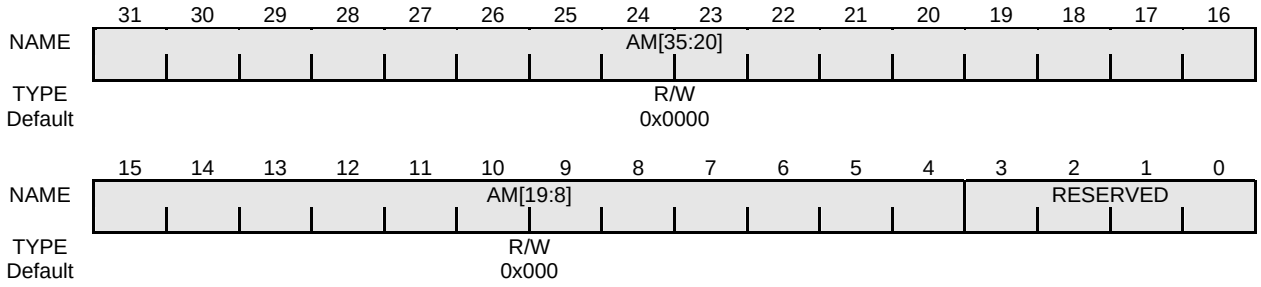


Figure 16-48 G2P I/O Space Address Mask Register

Table 16-49 G2P I/O Space Address Mask Register

Bits	Mnemonic	Field Name	Description	R/W
31:4	AM[35:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Default: 0x0_0000_00) Sets the bits to be subject to address comparison. See 16.3.7 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value of G2PIOMASK[31:0] becomes 0x00FF_FFF0.	R/W
3:0		Rsvd		R

16.4.38. G2P Memory Space 0 PCI Base Address Register (G2PM0PBASE)

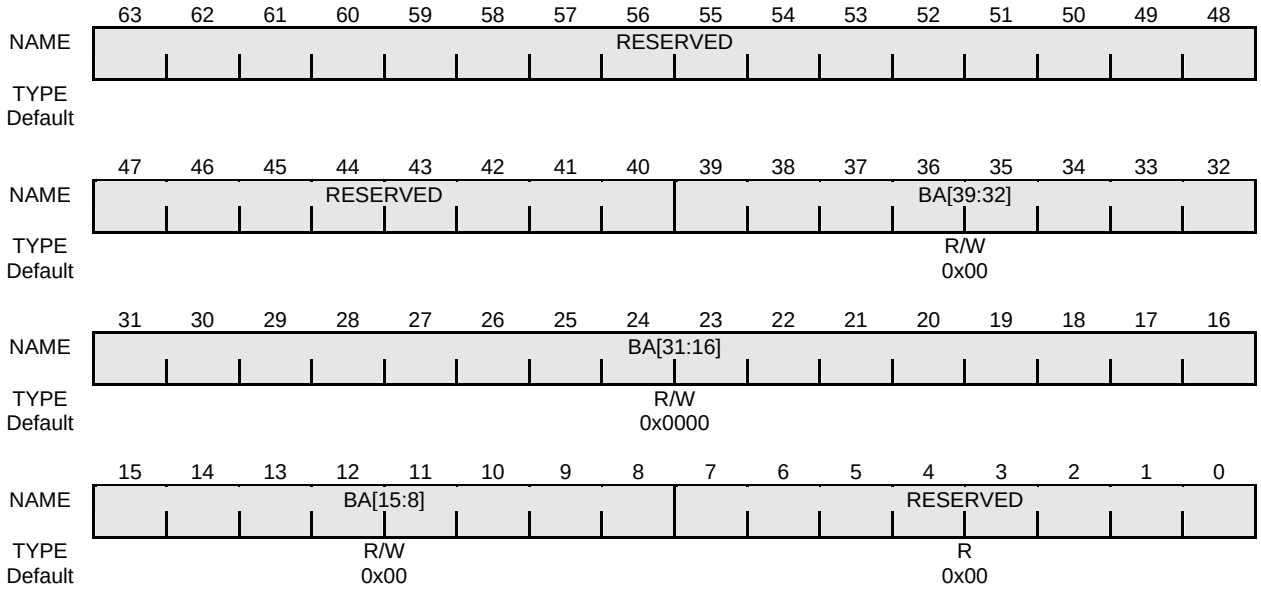


Figure 16-49 G2P Memory Space 0 G-Bus Base Address Register

Table 16-50 G2P Memory Space 0 G-Bus Base Address Register

Bits	Mnemonic	Field Name	Register	R/W
63:40		Rsvd		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of Memory Space 0 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Rsvd		R

16.4.39. G2P Memory Space 1 PCI Base Address Register (G2PM1PBASE)

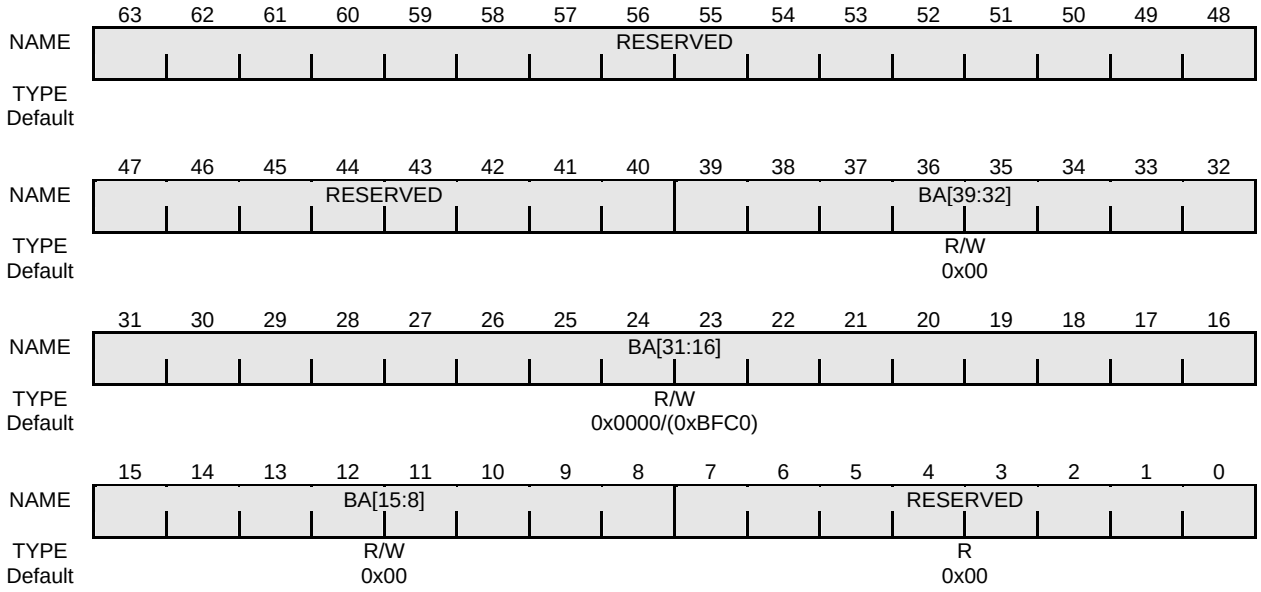


Figure 16-50 G2P Memory Space 1 G-Bus Base Address Register

Table 16-51 G2P Memory Space 1 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:40		Rsvd		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of Memory Space 1 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Rsvd		R

16.4.40. G2P Memory Space 2 PCI Base Address Register (G2PM2PBASE)

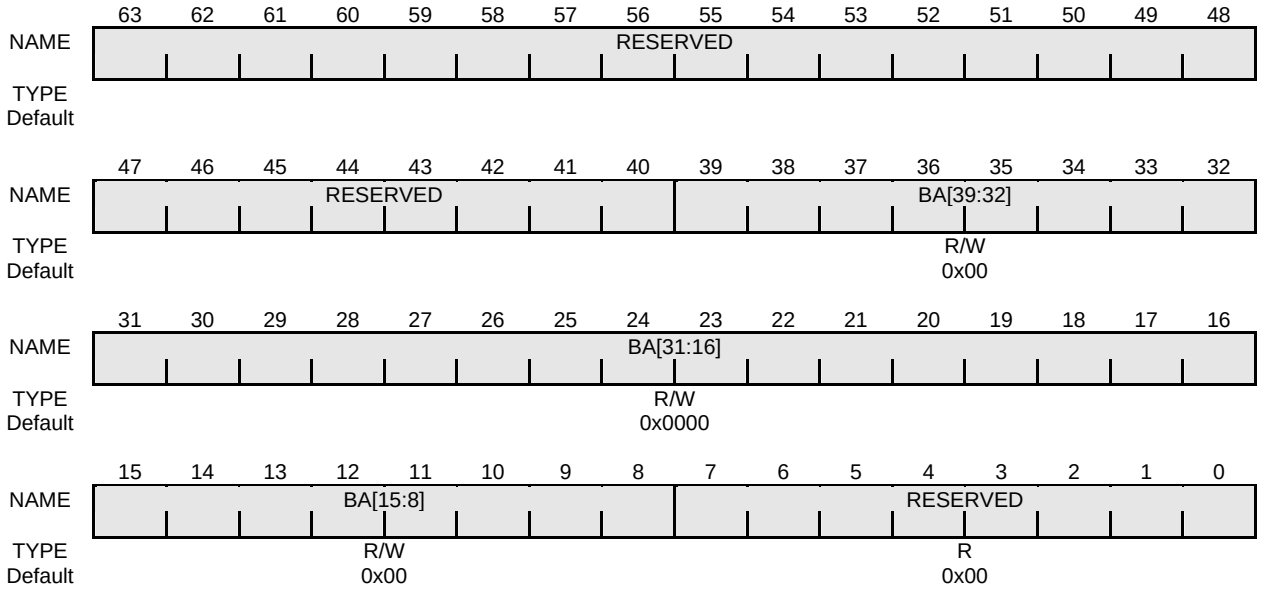


Figure 16-51 G2P Memory Space 2 G-Bus Base Address Register

Table 16-52 G2P Memory Space 2 G-Bus Base Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:40		Rsvd		—
39:8	BA[39:8]	Base Address	Base Address Default: 0x00_0000_00 (not PCI boot mode) Default:0x00_fffe_00 PCI boot mode) Sets the PCI Base address of Memory Space 2 for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Rsvd		R

16.4.41. G2P I/O Space PCI Base Address Register (G2PIOPBASE)

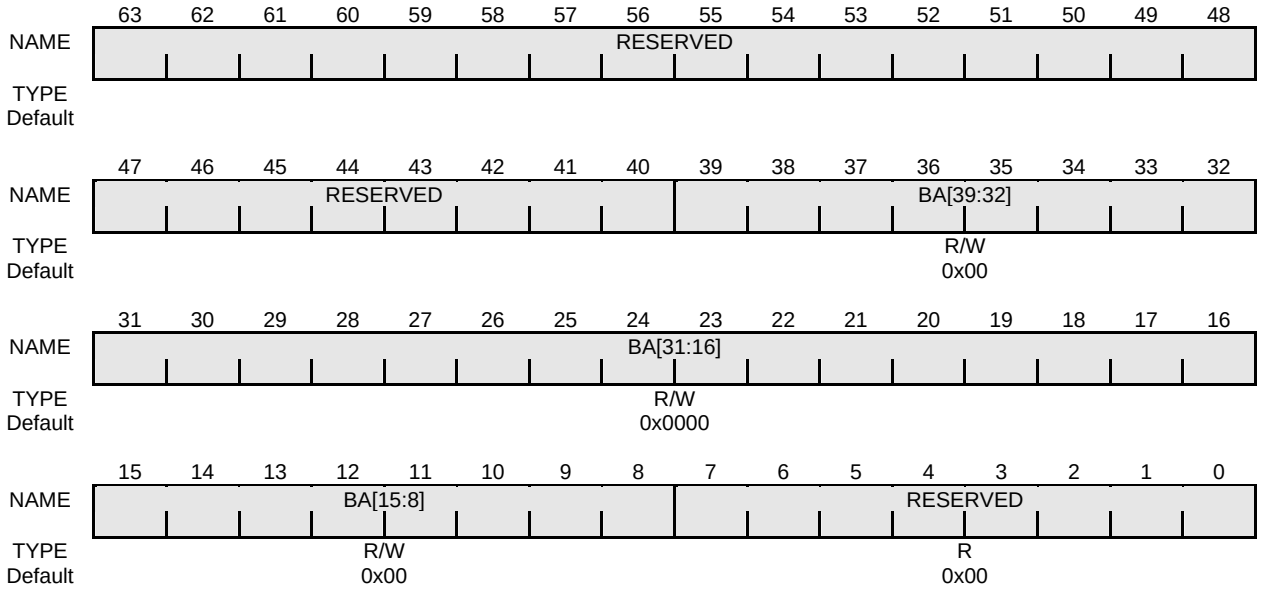


Figure 16-52 G2P I/O Space G-Bus Address Register

Table 16-53 G2P I/O Space G-Bus Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:40		Reserved		—
39:8	BA[39:8]	Base Address	Base Address (Default: 0x00_0000_00) Sets the PCI Base address of the I/O Space for initiator access. Can set the base address in 256-Byte units.	R/W
7:0		Reserved		R

16

16.4.42. PCI Controller Configuration Register (PCICCFG)

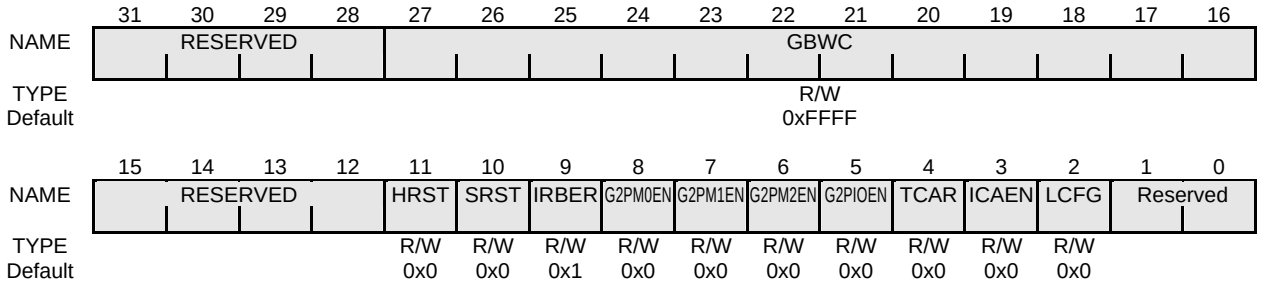


Figure 16-53 PCI Controller Configuration Register

Table 16-54 PCI Controller Configuration Register

Bit	Mnemonic	Field Name	Description	R/W
31:28		Rsvd		—
27:16	GBWC	G-Bus Wait Counter Setting	G Bus Wait Counter (Default: 0xFFF) Sets the Retry response counter at the G-Bus during a PCI initiator Read transaction. When the initiator Read access cycle exceeds the setting of this counter, a Retry response is sent to the G-Bus and the G-Bus is released. PCI Read operation continues. This counter uses the G-Bus clock (GBUSCLK) when operating. When 0x000 is set, a Retry response is not sent to the G-Bus by a long response cycle count. When the G-Bus timeout count is used with the value other than the initial value 4096 GBUSCLK, G-BUS timeout may occur before a Retry response is sent. When G-Bus timeout of the configuration register (CCFG.GTOT) is used with the value other than the initial value (11), set the following maximum values to the register. GTOT value Maximum value of the register 10 (2048 GBUSCLK) : 0x7f0 01 (1024 GBUSCLK) : 0x3f0 00 (512 GBUSCLK) : 0x1f0	R/W
15:12		Rsvd		—
11	HRST	Hardware Reset	Hard Reset (Default: 0x0) Performs PCI Controller hardware reset control. This bit is automatically cleared when Reset ends. This is a diagnostic function. The PCI Controller cannot be accessed for 32 G-Bus clock cycles after this bit is set. 1: Perform a hardware reset on the PCI Controller. 0: Do not perform a hardware reset on the PCI Controller.	R/W
10	SRST	Software Reset	Soft Reset (Default: 0x0) Performs PCI Controller software reset control. Data is also reloaded to the Configuration Space Register from the Configuration Data Register. Please set this bit after the EEPROM Load End bit (PCICSTATUS.E2PDONE) is set. Also, please use the software to clear this bit at least four PCI Bus Clock cycles after Reset. Other registers of the PCI Controller cannot be accessed while this bit is set. This bit differs from the Hardware Reset bit (HRST). The following register values are not initialized. G2P Status Register (G2PSTATUS) PCI Bus Arbiter Status Register (PBASTATUS) PCI Controller Status Register (PCICSTATUS) Software Reset bit (PCICCFG.SRST) Load Configuration Register bit (PCICCFG.LCFG) 1: The PCI Controller is reset by the software. 0: The PCI Controller is not reset by the software.	R/W

Table 16-54 PCI Controller Configuration Register

Bit	Mnemonic	Field Name	Description	R/W
9	IRBER	Bus Error Response Setting During Initiator Read	Initiator Read Bus Error Response (Default: 0x1) Bus error responses on the G-Bus are controlled when the following phenomena indicated by the PCI Status, Command Register (PICSTATUS) and the G2P Status Register (G2PSTATUS) occur during initiator Read access. Detected Parity Error (PCISTATUS.DPE) Received Master Abort (PCISTATUS.RMA) Received Target Abort (PCISTATUS.RTA) Initiator Detected TRDY Time Out Error (G2PSTATUS.IDTTOE) Initiator Detected Retry Time Out Error (G2PSTATUS.IDRTOE) 1: Responds with a Bus error on the G-Bus. 0: Does not respond with a Bus error on the G-Bus. (Normally terminates the Read transaction on the G-Bus. Read data is invalid.)	R/W
8	G2PM0EN	Initiator Memory Space 0 Enable	Initiator Memory Space 0 Enable (Default: 0x0) Controls PCI initiator access to Memory Space 0. 1: Memory Space 0 is valid. 0: Memory Space 0 is invalid.	R/W
7	G2PM1EN	Initiator Memory Space 1 Enable	Initiator Memory Space 1 Enable (Default: 0x0) Controls PCI initiator access to Memory Space 1. 1: Memory Space 1 is valid. 0: Memory Space 1 is invalid.	R/W
6	G2PM2EN	Initiator Memory Space 2 Enable	Initiator Memory Space 2 Enable Default: 0x0 (not PCI boot mode) Default: 0x1 (PCI boot mode) Controls PCI initiator access to Memory Space 2. 1: Memory Space 2 is valid. 0: Memory Space 2 is invalid.	R/W
5	G2PIOEN	Initiator I/O Space Enable	Initiator I/O Space Enable (Default: 0x0) Controls PCI initiator access to the I/O Space.. 1: I/O Space is valid. 0: I/O Space is invalid.	R/W
4	TCAR	Target Configuration Access Ready	Target Configuration Access Ready Satellite mode: Default: 0x1 - when in PCI Boot mode 0x0 - when not in PCI Boot mode PCI controller receives a target access, when this bit is 1 and PCISTATUS.E2PDONE bit is 1. Configuration access from the PCI Bus can be accepted during PCI Boot up after initialization from EEPROM or after each initialization ends. Please use the software to set this bit after initialization ends. Retry response to PCI configuration access is performed until this bit is set. 1: Responds to PCI target access. 0: Performs a Retry response to PCI target access. Host mode: Default:0x0 The TX4939 will perform a Retry response when this bit is not set.	R/W
3	ICAEN	Initiator Configuration Access Enable	Initiator Configuration Access Enable (Default: 0x1) Controls initiator PCI configuration access using the G2P Configuration Address Register (G2PCFGADRS) and the G2P Configuration Data Register (G2PCFGDATA). This is a diagnostic function. 1: Initiator configuration access is possible. 0: Initiator configuration access is not possible.	R/W
2	LCFG	Load Configuration Data Register	Load PCI Configuration Data Register (Default: 0x0) When a software reset is performed on this bit using the Software Reset bit (PCICFG.SRST) when this bit is already set, data is loaded to the Configuration Space Register from the Configuration Data 0/1/2/3 Register. 1: Load from the Configuration Data 0/1/2/3 Register. 0: Load from EEPROM.	R/W
1:0		Rsvd		—

16.4.43. PCI Controller Status Register (PCICSTATUS)

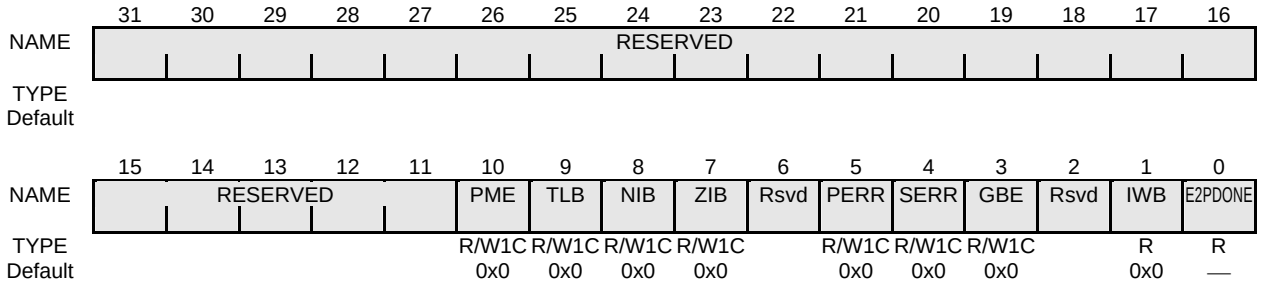


Figure 16-54 PCI Controller Status Register

Table 16-55 PCI Controller Status Register

Bit	Mnemonic	Field Name	Description	R/W
31:11		Rsvd		—
10	PME	PME Detect	PME Detect (Default: 0x0) This bit indicates that assertion of the PME* signal was detected. 1: Indicates that assertion of the PME* signal was detected. 0: Indicates that assertion of the PME* signal was not detected.	R/W1C
9	TLB	Long Burst Transfer Detect	Too Long Burst Detect (Default: 0x0) Indicates that a Burst transfer by the on-chip DMA Controller exceeding 16 DWORDs was detected. 1: Indicates that a Burst transfer exceeding 16 DWORDs was detected. 0: Indicates that no Burst transfer exceeding 16 DWORDs was detected.	R/W1C
8	NIB	Negative Increment Burst Detect	Negative Increment Burst Detect (Default: 0x0) Indicates that Burst transfer by the on-chip DMA Controller in the negative direction was detected. 1: Indicates that a Burst transfer in the negative direction was detected. 0: Indicates that no Burst transfer in the negative direction was detected.	R/W1C
7	ZIB	Zero Increment Burst Detect	Zero Increment Burst Detect (Default: 0x0) Indicates that Burst transfer by the on-chip DMA Controller without an address increment was detected. 1: Indicates that a Burst transfer without an address increment was detected. 0: Indicates that no Burst transfer without an address increment was detected.	R/W1C
6		Rsvd		—
5	PERR	PERR* Detected	PERR* Occurred (Default: 0x0) Indicates that the Parity Error signal (PERR*) was asserted. This bit is a monitor status bit that records assertion of the PERR* signal even if the TX4939 is not accessing PCI. 1: Indicates that the PERR* signal was asserted. 0: Indicates that the PERR* signal was not asserted.	R/W1C
4	SERR	SERR* Detected	SERR* Occurred (Default: 0x0) Indicates that the System Error signal (SERR*) was asserted. This bit is a monitor status bit that records assertion of the SERR* signal even if the TX4939 is not accessing PCI. 1: Indicates that the SERR* signal was asserted. 0: Indicates that the SERR* signal was not asserted.	R/W1C
3	GBE	G-Bus Error Detect	G-Bus Error Detect (Default: 0x0) Indicates that a G-Bus Error occurred in the G-Bus Master cycle of the PCI Controller. This error is indicated when a timeout occurs on the G-Bus. This bit is only set by Master cycle Bus Errors. 1: Indicates that a G-Bus Error was detected. 0: Indicates that no G-Bus Error was detected.	R/W1C
2		Rsvd		—
1	IWB	Initiator Write Busy	Initiator Write Busy (Busy: 0x0) Indicates that a Write cycle was in progress when a Write cycle to the PCI Bus was executed. While a Write cycle is in progress, no error status to that Write cycle is reflected. Therefore, this bit is used to confirm the status when it changes from “1” to “0” after the Write cycle ends. 1: Indicates that a Write cycle is in progress. 0: Indicates that no Write cycle is in progress.	R

Table 16-55 PCI Controller Status Register

Bit	Mnemonic	Field Name	Description	R/W
0	E2PDONE	EEPROM Load Done	EEPROM Load Done (Default--) When using EEPROM, this bit indicates that data loading from EEPROM is complete. This bit is set to “1”. 1: Indicates that data loading from EEPROM is complete. 0: Indicates that data loading from EEPROM is not complete.	R

16.4.44. PCI Controller Interrupt Mask Register (PCICMASK)

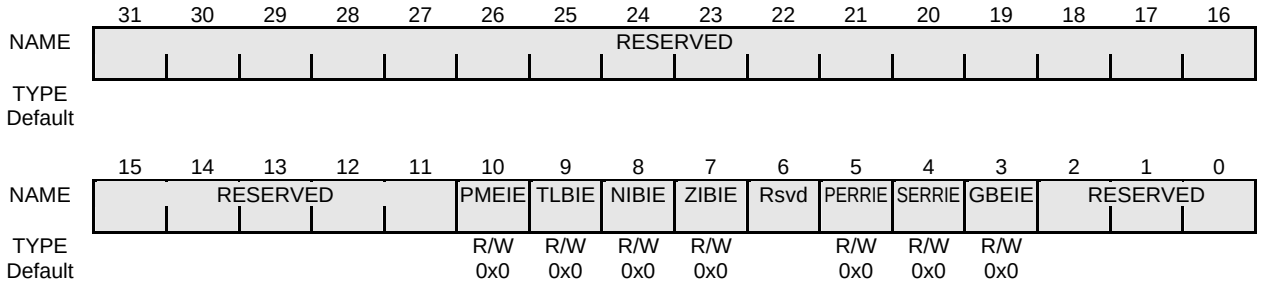


Figure 16-55 PCI Controller Interrupt Mask Register

Table 16-56 PCI Controller Interrupt Mask Register

Bit	Mnemonic	Field Name	Description	R/W
31:11		Rsvd		—
10	PMEIE	PME Detect Interrupt Enable	PME* Signal Interrupt Enable (Default: 0x0) This bit generates an interrupt when input of the PME* signal is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
9	TLBIE	Long Burst Transfer Detect Interrupt	Too Long Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Burst transfer by the on-chip DMA Controller exceeding 8 DWORDs was detected. 1: Generates an interrupt. 0: Does not generate an interrupt	R/W
8	NIBIE	Negative Increment Burst Transfer Detect Interrupt Enable	Negative Increment Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a negative direction Burst transfer by the on-chip DMA Controller is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
7	ZIBIE	Zero Increment Burst Transfer Detect Interrupt Enable	Zero Increment Burst Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Burst transfer by the on-chip DMA Controller without an address increment is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
6		Rsvd		—
5	PERRIE	PERR* Detect Interrupt Enable	PERR* Interrupt Enable (Default: 0x0) This bit generates an interrupt when the Parity Error signal (PERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
4	SERRIE	SERR* Detect Interrupt Enable	SERR* Interrupt Enable (Default: 0x0) This bit generates an interrupt when the System Error signal (SERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
3	GBEIE	G-Bus Bus Error Detect Interrupt Enable	G-Bus Bus Error Interrupt Enable (Default: 0x0) This bit generates an interrupt when a Bus Error is asserted while the PCI Controller is the G-Bus Master. 1: Generates an interrupt. 0: Does not generate an interrupt.	R/W
2:0		Rsvd		—

16.4.45. P2G Memory Space 0 G-Bus Base Address Register (P2GM0GBASE)

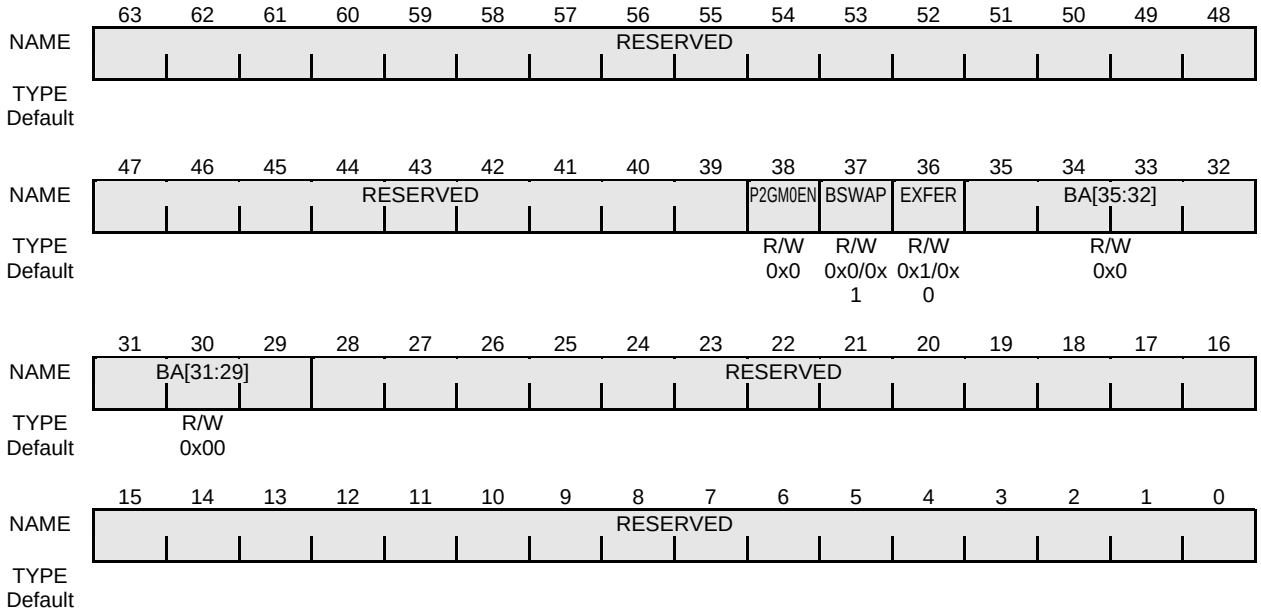


Figure 16-56 P2G Memory Space 0 G-Bus Base Address Register

Table 16-57 P2G Memory Space 0 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:39		Rsvd		—
38	P2GM0EN	Memory Space 0 Enable	Target Memory Space 0 Enable (Default: 0x0) Controls whether Memory Space 0 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 0 Lower Base Address Register or the Memory Space 0 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates Memory Space 0 for target access. 0: Invalidates Memory Space 0 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 0 for target access.. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 0 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:29	BA[35:29]	Base Address	Base Address 0 (Default: 0x000) Sets the G-Bus base bus address of Memory Space 0 for target access. Can set the base address in 512-MB units.	R/W
28:0		Rsvd		—

16.4.46. P2G Memory Space 1 G-Bus Base Address Register (P2GM1GBASE)

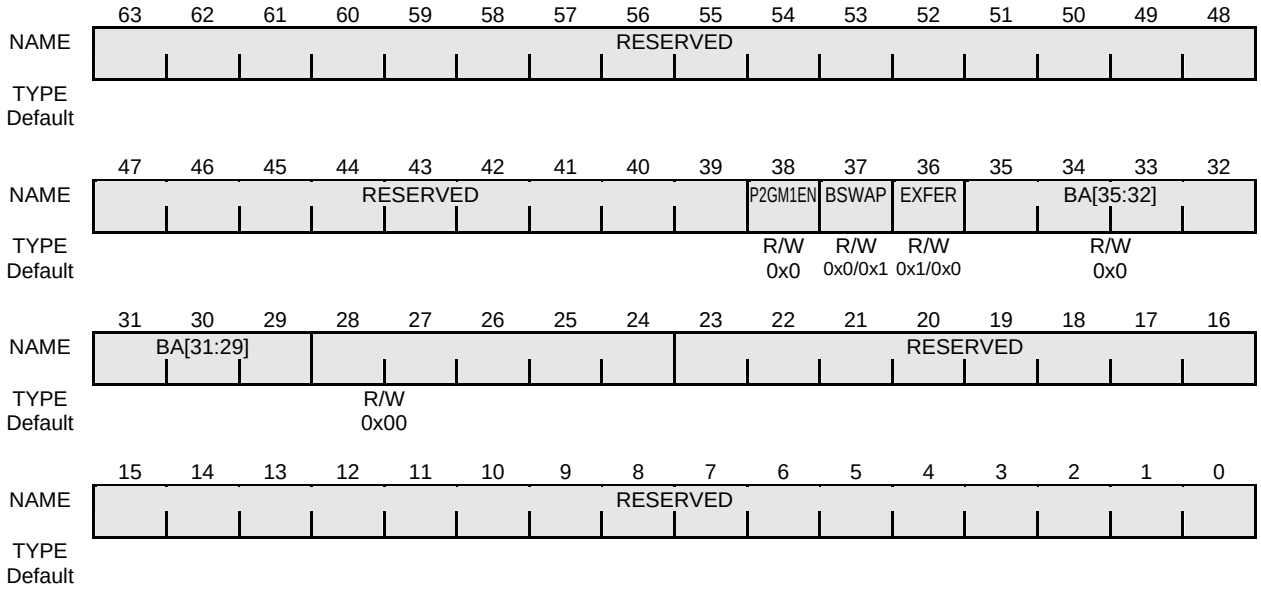


Figure 16-57 P2G Memory Space 1 G-Bus Base Address Register

Table 16-58 P2G Memory Space 1 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:39		Rsvd		—
38	P2GM1EN	Memory Space 1 Enable	Target Memory Space 1 Enable (Default: 0x0) Controls whether Memory Space 1 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 1 Lower Base Address Register or the Memory Space 1 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “1” is returned to Reads as a response. 1: Validates Memory Space 1 for target access. 0: Invalidates Memory Space 1 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 1 for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 1 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:29	BA[35:29]	Memory Space Base Address 1	Base Address 0 (Default: 0x000) Sets the G-Bus base bus address of Memory Space 1 for target access. Can set the base address in 512-MB units.	R/W
28:24		Rsvd		R/W
23:0		Rsvd		—

16.4.47. P2G Memory Space 2 G-Bus Base Address Register (P2GM2GBASE)

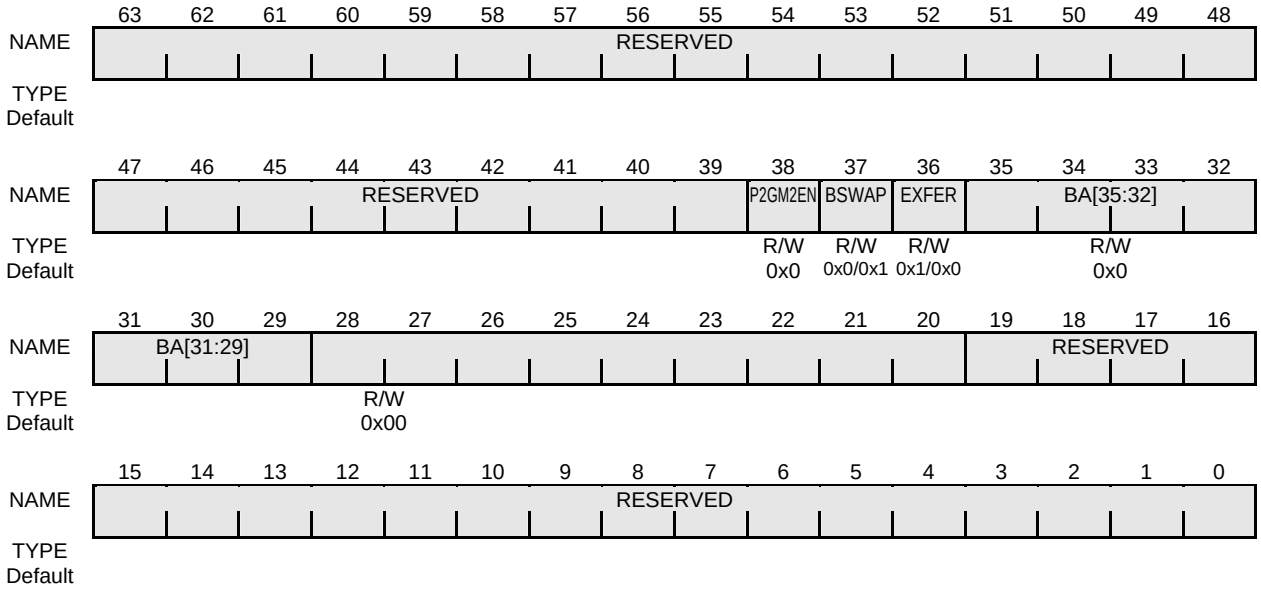


Figure 16-58 P2G Memory Space 2 G-Bus Base Address Register

Table 16-59 P2G Memory Space 2 G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:39		Rsvd		—
38	P2GM2EN	Memory Space 2 Enable	Target Memory Space 2 Enable (Default: 0x0) Controls whether Memory Space 2 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 2 Lower Base Address Register or the Memory Space 2 Upper Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates Memory Space 2 for target access. 0: Invalidates Memory Space 2 for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of Memory Space 2 for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to Memory Space 2 through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of Memory Space 2 for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:29	BA[35:29]	Memory Space Base Address 2	Base Address 2 (Default: 0x000) Sets the G-Bus base bus address of Memory Space 2 for target access. Can set the base address in 512-MB units.	R/W
28:20		Rsvd		R/W
19:0		Rsvd		—

16.4.48. P2G I/O Space G-Bus Base Address Register (P2GIOGBASE)

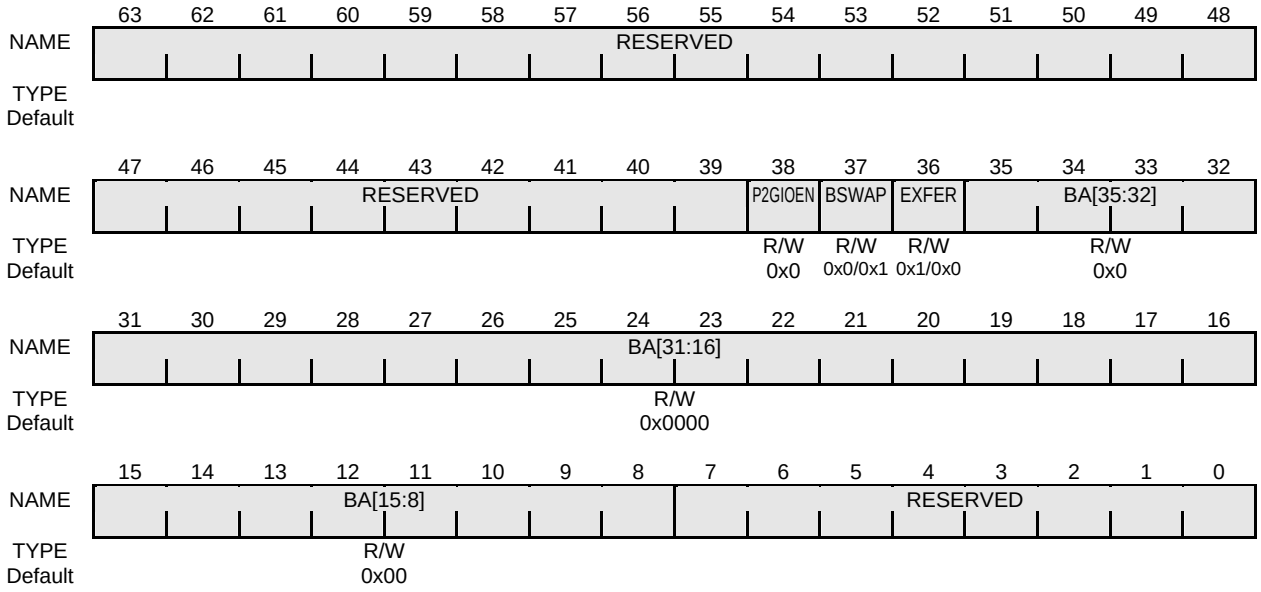


Figure 16-59 P2G I/O Space G-Bus Base Address Register

Table 16-60 P2G I/O Space G-Bus Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
63:39		Rsvd		—
38	P2GIOEN	I/O Space Enable	Target I/O Space Enable (Default: 0x0) Controls whether the I/O Space for target access is valid or invalid. When this bit is set to invalid, Writes to the I/O Space Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates I/O Space for target access. 0: Invalidates I/O Space for target access.	R/W
37	BSWAP	Byte Swap	Byte Swap Disable (Default: Little Endian Mode: 0x1; Big Endian Mode: 0x0) Sets the byte swapping of the I/O Space for target access. 1: Do not perform byte swapping. 0: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “1” when in the Big Endian Mode, the byte order of transfer to the I/O Space through DWORD (32-bit) access will not change. See detail in 16.3.10 Endian Switching Function	R/W
36	EXFER	Endian Transfer	Endian Transfer (Default: Little Endian Mode: 0x0; Big Endian Mode: 0x1) Sets the Endian Transfer of the I/O Space for target access. 1: Performs Endian Transfer. 0: Does not perform Endian Transfer. Please use the default state. See detail in 16.3.10 Endian Switching Function	R/W
35:8	BA[35:8]	Memory Space Base Address 2	Base Address 2 (Default: 0x000) Sets the G-Bus base bus address of the I/O Space for target access. Can set the base address in 256-byte units.	R/W
7:1		Rsvd		—

16.4.49. G2P Configuration Address Register(G2PCFGADRS)

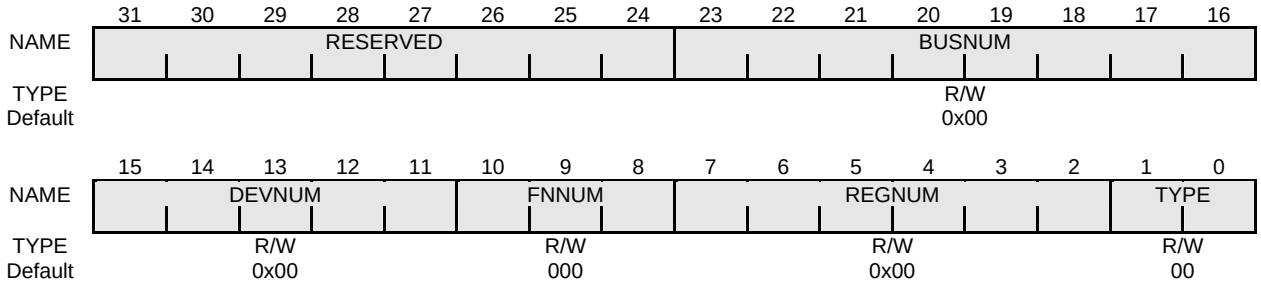


Figure 16-60 G2P Configuration Address Register

Table 16-61 G2P Configuration Address Register

Bit	Mnemonic	Field Name	Description	R/W
31:24		Rsvd		—
23:16	BUSNUM	Bus Number	Bus Number (Default: 0x00) Indicates the target PCI Bus Number (one of 256).	R/W
15:11	DEVNUM	Device Number	Device Number (Default: 0x00) This field is used to identify the target physical device number. (This is one number out of 32 devices. 21 of these 32 devices are used.) When in the address phase of Type 0 configuration access, AD[31:11] of the upper 21 address lines are used as the IDSEL signal. 0x00: Use AD [11] as IDSEL. 0x01: Use AD [12] as IDSEL. 0x02: Use AD [13] as IDSEL. : : 0x13: Use AD [30] as IDSEL. 0x14: Use AD [31] as IDSEL. 0x15 - 0x1F: Rsvd	R/W
10:8	FNNUM	Function Number	Function Number (Default: 000) This field is used to identify the target logic function number (one out of 8).	R/W
7:2	REGNUM	Register Number	Register Number (Default: 0x00) This field is used to identify the DWORD (one out of 64) inside the Configuration Space of the target function	R/W
1:0	TYPE	Type	Type (Default; 00) This field is used to identify the address type in the address phase of the target function configuration cycle. 0x0: Type 0 configuration (Use the AD[31:11] signal as the IDSEL signal.) 0x1: Type 1 configuration (Output all bits unchanged as the address to the AD[ ] signal.)	R/W

16.4.50. G2P Configuration Data Register (G2PCFGDATA)

This is the only register that supports Byte access and 16-bit Word access. The upper address bit of the PCI Configuration Space is specified by the G2P Configuration Address Register (G2PCFGADRS). The lower two bits of the address are specified by the lower two bits of the offset address in this register as shown in Table 16-62.

Table 16-62 PCI Configuration Space Access Address

Access Size	Configuration Space Address [1:0]	Offset Address	
		Little Endian Mode	Big Endian Mode
32-bit	00	0xD1A4	0xD1A4
16-bit	00	0xD1A4	0xD1A6
	10	0xD1A6	0xD1A4
8-bit	00	0xD1A4	0xD1A7
	01	0xD1A5	0xD1A6
	10	0xD1A6	0xD1A5
	11	0xD1A7	0xD1A4

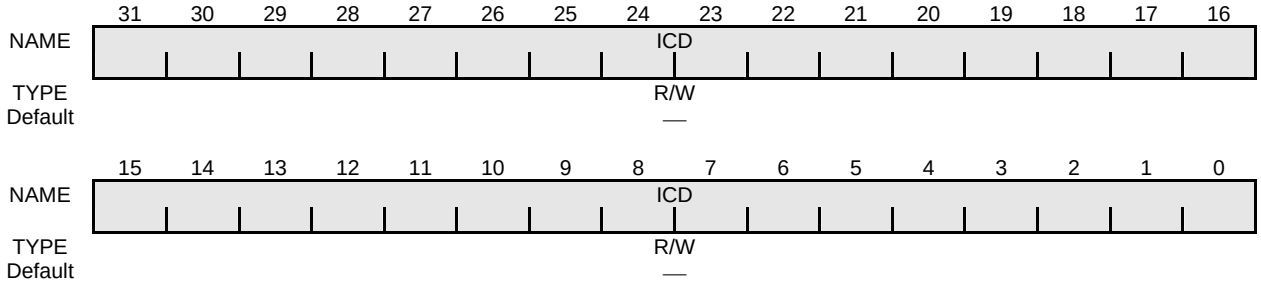


Figure 16-61 G2P Configuration Data Register

Table 16-63 G2P Configuration Data Register

Bits	Mnemonic	Field Name	Description	R/W
31:0	ICD	Initiator Configuration Data	Initiator Configuration Data Register (Default--) This is a data port that is used when performing initiator PCI configuration access. PCI configuration Read or Write transactions are issued when this register is read to or written from.	R/W

16

16.4.51. G2P Interrupt Acknowledge Data Register (G2PINTACK)

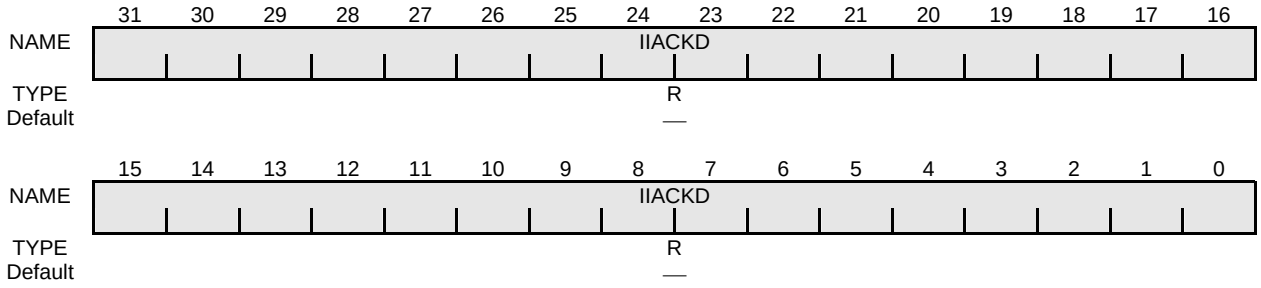


Figure 16-62 G2P Interrupt Acknowledge Data Register

Table 16-64 G2P Interrupt Acknowledge Data Register

Bits	Mnemonic	Field Name	Description	R/W
31:0	IIACKD	Initiator Interrupt Acknowledge Address Port	Initiator Interrupt Acknowledge Address Port (Default--) An Interrupt Acknowledge cycle is generated on the PCI Bus when this register is read. The data that is returned by this Read transaction becomes the Interrupt Acknowledge data.	R

16.4.52. G2P Special Cycle Data Register (G2PSPC)

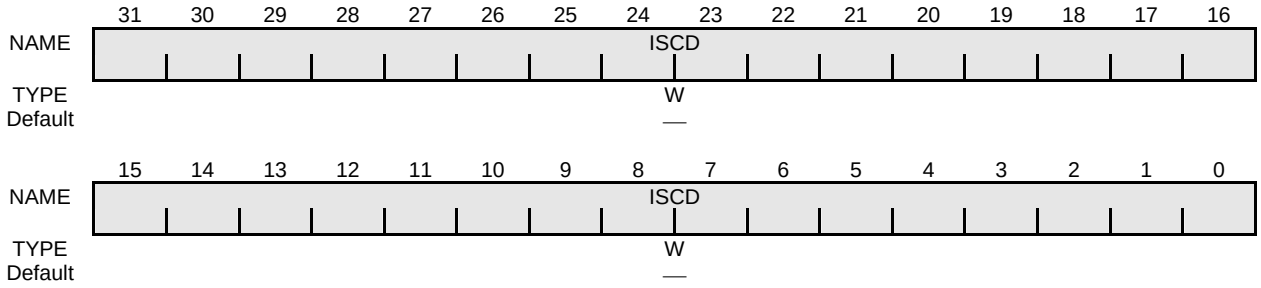


Figure 16-63 G2P Special Cycle Data Register

Table 16-65 G2P Special Cycle Data Register

Bits	Mnemonic	Field Name	Description	R/W
31:0	ISCD	Initiator Special Cycle Data Port	Initiator Special Cycle Data Port (Default--) When this register is written to, Special Cycles are generated on the PCI Bus depending on the data that is written.	W

16.4.53. Configuration Data 0 Register (PCICDATA0)

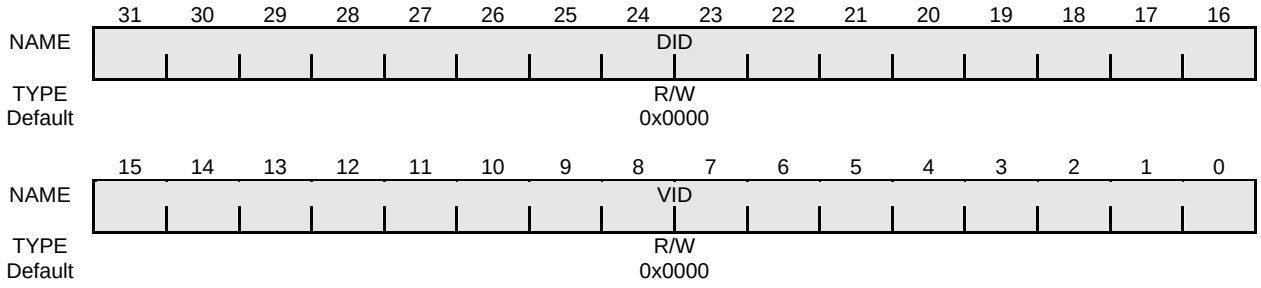


Figure 16-64 ID Register

Table 16-66 ID Register

Bits	Mnemonic	Field Name	Description	R/W
31:16	DID	Device ID	Device ID (Default: 0x0000) This is the data loaded in the Device ID Register of the PCI Configuration Space.	R/W
15:0	VID	Vendor ID	Vendor ID (Default: 0x0000) This is the data loaded in the Vendor ID Register of the PCI Configuration Space.	R/W

16.4.54. Configuration Data 1 Register (PCICDATA1)

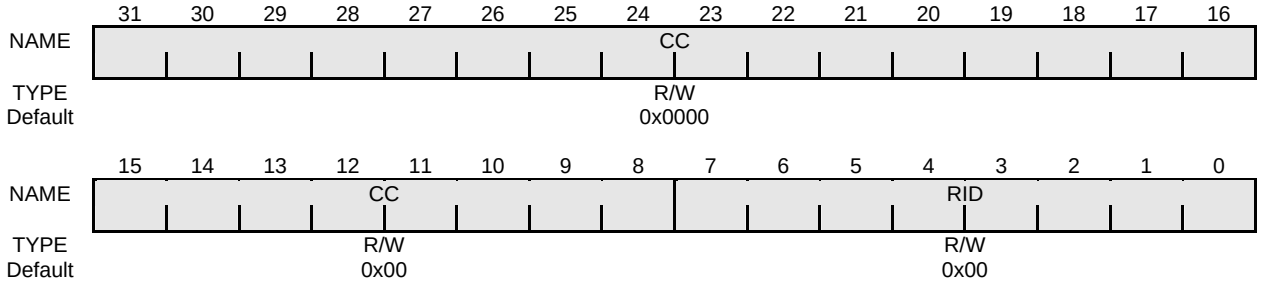


Figure 16-65 Class Code/Revision ID Register

Table 16-67 Class Code/Revision ID Register

Bis	Mnemonic	Field Name	Description	R/W
31:8	CC	Class Code	Class Code (Default: 0x000000) This is the data loaded in the Class Code Register of the PCI Configuration Space.	R/W
7:0	RID	Revision ID	Revision ID (Default: 0x00) This is the data loaded in the Revision ID Register of the PCI Configuration Space.	R/W

16.4.55. Configuration Data 2 Register (PCICDATA2)

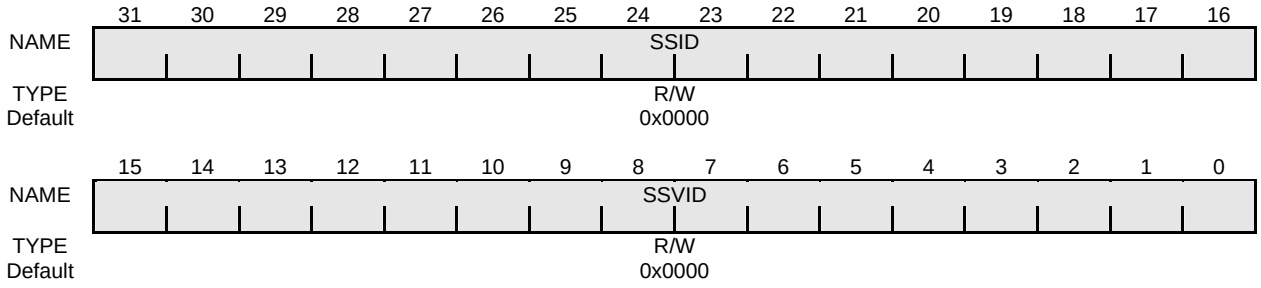


Figure 16-66 Sub System ID Register

Table 16-68 Sub System ID Register

Bits	Mnemonic	Field Name	Description	R/W
31:16	SSID	Sub System ID	Subsystem ID (Default: 0x0000) This is the data loaded in the Sub System ID Register of the PCI Configuration space.	R/W
15:0	SSVID	Sub System Vendor ID	Subsystem Vendor ID (Default: 0x0000) This is the data loaded in the Sub System Vendor ID Register of the PCI Configuration space.	R/W

16.4.56. Configuration Data 3 Register (PCICDATA3)

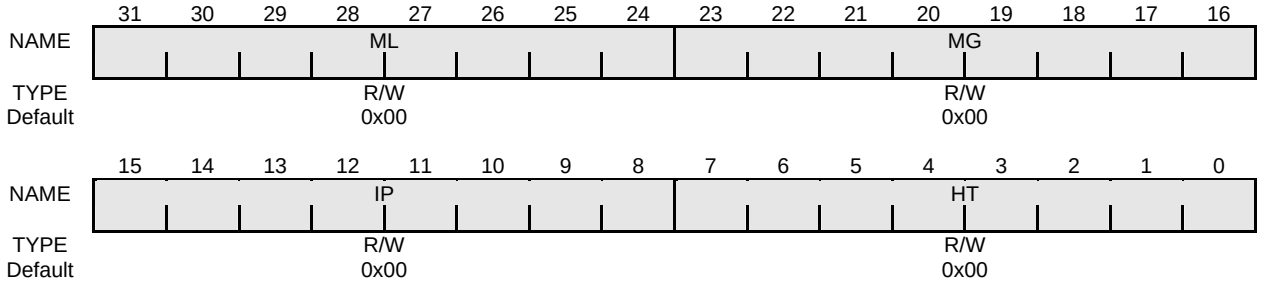


Figure 16-67 PCI Configuration Register 2

Table 16-69 PCI Configuration Register 2

Bits	Mnemonic	Field Name	Description	R/W
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Default: 0x00) This is the data loaded in the Max_Lat Register of the PCI Configuration Space.	R/W
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Default: 0x00) This is the data loaded in the Min_Gnt Register of the PCI Configuration Space.	R/W
15:8	IP	Interrupt Pin	Interrupt Pin (Default: 0x00) This is the data loaded in the Interrupt Pin Register of the PCI Configuration Space.	R/W
7:0	HT	Header Type	Header Type (Default: 0x00) This is the data loaded in the Header Type Register of the PCI Configuration Space.	R/W

16.4.57. PDMAC Chain Address Register (PDMCA)

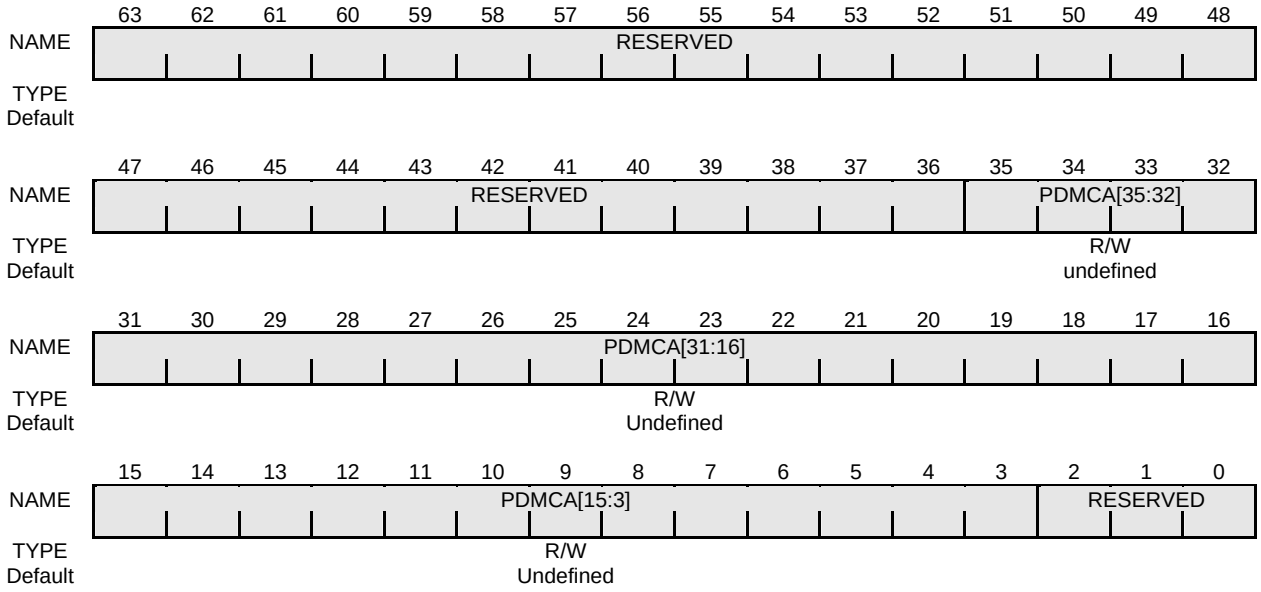


Figure 16-68 PDMAC Chain Address Register

Table 16-70 PDMAC Chain Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:36		Rsvd		—
35:3	PDMCA	Chain Address	PDMAC Chain Address (Default is undefined) The address of the next PDMAC Data Command Descriptor to be read is specified by a G-Bus physical address on a 64-bit address boundary. This register value is held without being affected by a Reset. 0 value judgement is performed when the lower 32 bits of this register are rewritten. DMA transfer is automatically initiated if the result is not “0”.	R/W
2:0		Rsvd		—

16.4.58. PDMAC G-Bus Address Register (PDMGA)

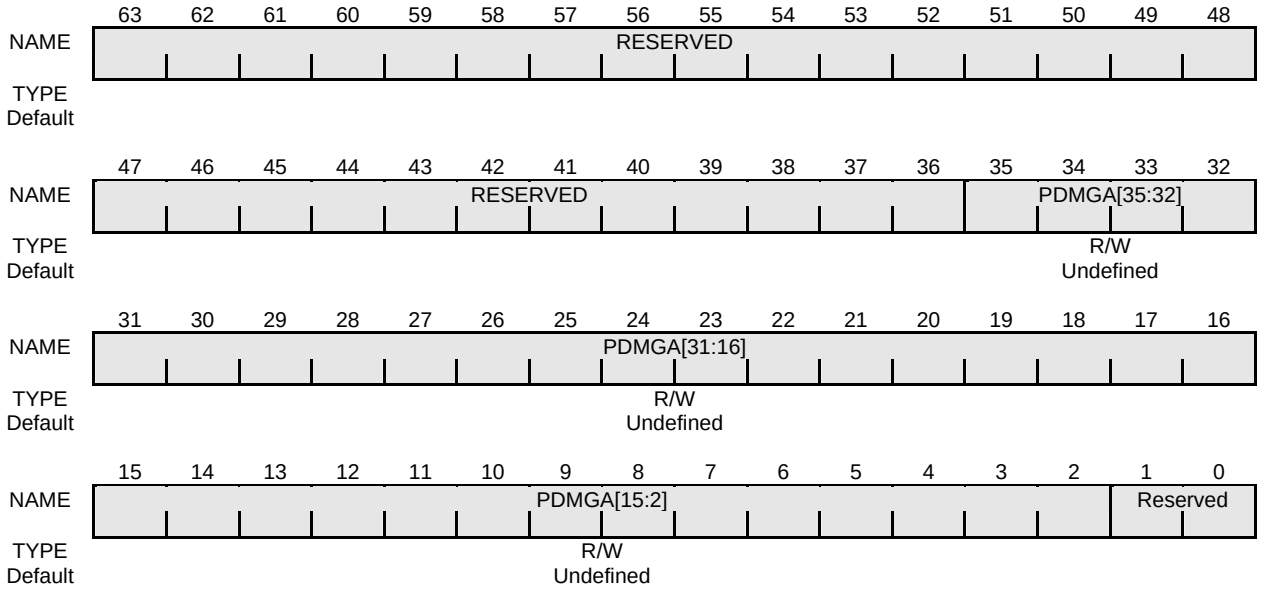


Figure 16-69 G-Bus Address Register

Table 16-71 G-Bus Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:36		Rsvd		—
35:2	PDMGA	G-Bus Address	PDMAC G-Bus Address (Default is undefined) The G-Bus DMA transfer address is specified by a G-Bus physical address on a 32-bit address boundary. This register value is used for G-Bus Read access during DMA transfer from the G-Bus to the PCI Bus, or it is used for G-Bus Write access during DMA transfer from the PCI Bus to the G-Bus. This register value is held without being affected by a Reset.	R/W
1:0		Rsvd		—

16.4.59. PDMAC PCI Bus Address Register (PDMPA)

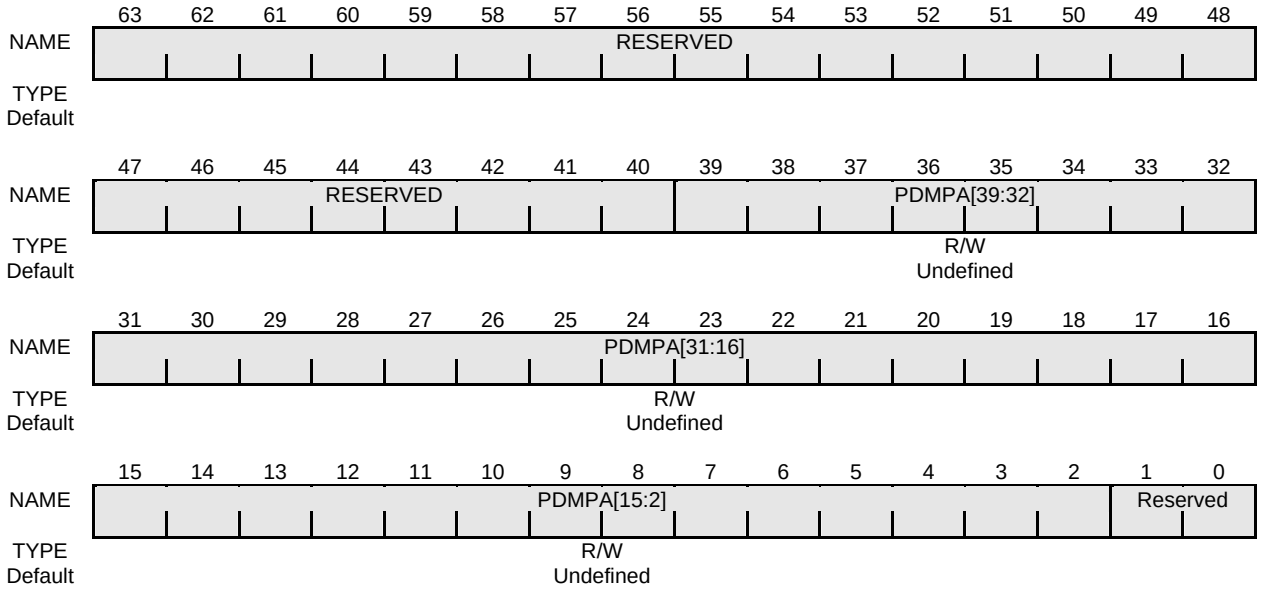


Figure 16-70 PCI Bus Address Register

Table 16-72 PCI Bus Address Register

Bits	Mnemonic	Field Name	Description	R/W
63:38		Reserved		—
39:2	PDMPA	PCI Bus Address	PDMAC PCI-Bus Address (Default is undefined) The PCI Bus DMA transfer address is specified by a PCI Bus physical address on a 32-bit address boundary. This register value is held without being affected by a Reset. Note: This register value is used for PCI Bus Write access during DMA transfer from the G-Bus to the PCI Bus, or it is used for PCI Bus Read access during DMA transfer from the PCI Bus to the G-Bus.	R/W
1:0		Rsvd		—

16.4.60. PDMAC Count Register (PDMCTR)

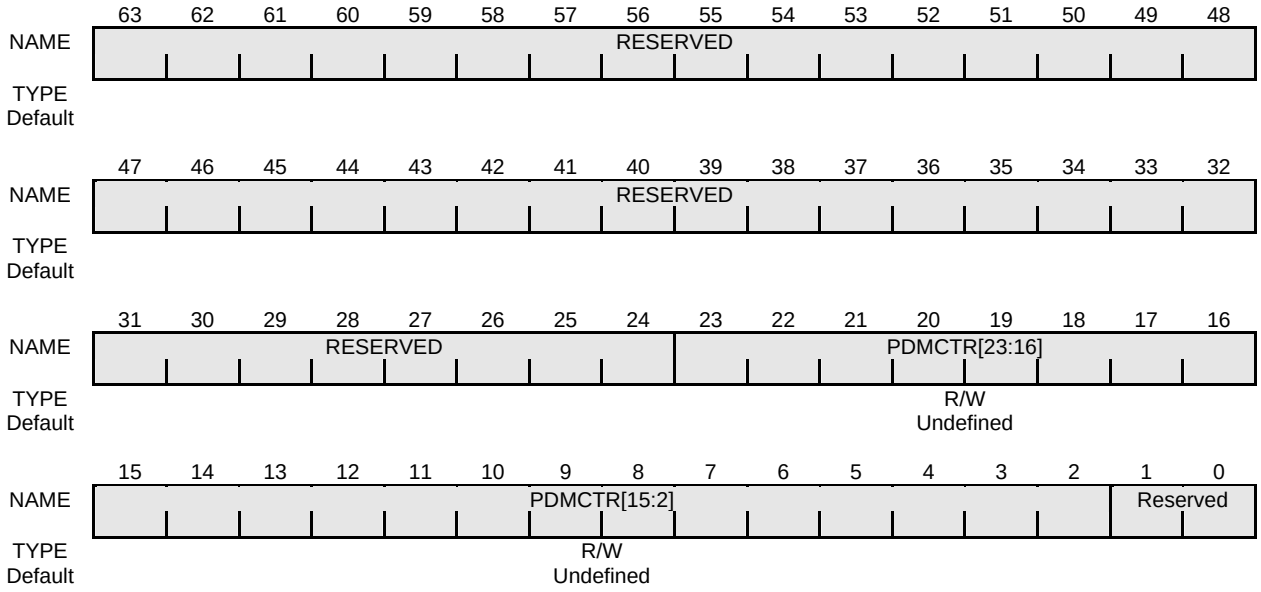


Figure 16-71 Count Register

Table 16-73 Count Register

Bits	Mnemonic	Field Name	Description	R/W
63:24		Rsvd		—
23:2	PDMCTR	Transfer Byte Count	PDMAC Transfer Count (Default is undefined) Sets an uncoded 24-bit transfer byte count in 32-bit word units. Also, the setting of this register must always be a multiple of the transfer size specified inside the PDMAC Control Register. No data transfer is performed if a count of “0” is set. This byte count value is calculated from the transferred byte size as the PDMAC performs a DMA transfer. This register value is held without being affected by a Reset.	R/W
1:0		Rsvd		—

16.4.61. PDMAC Control Register (PDMCFG)

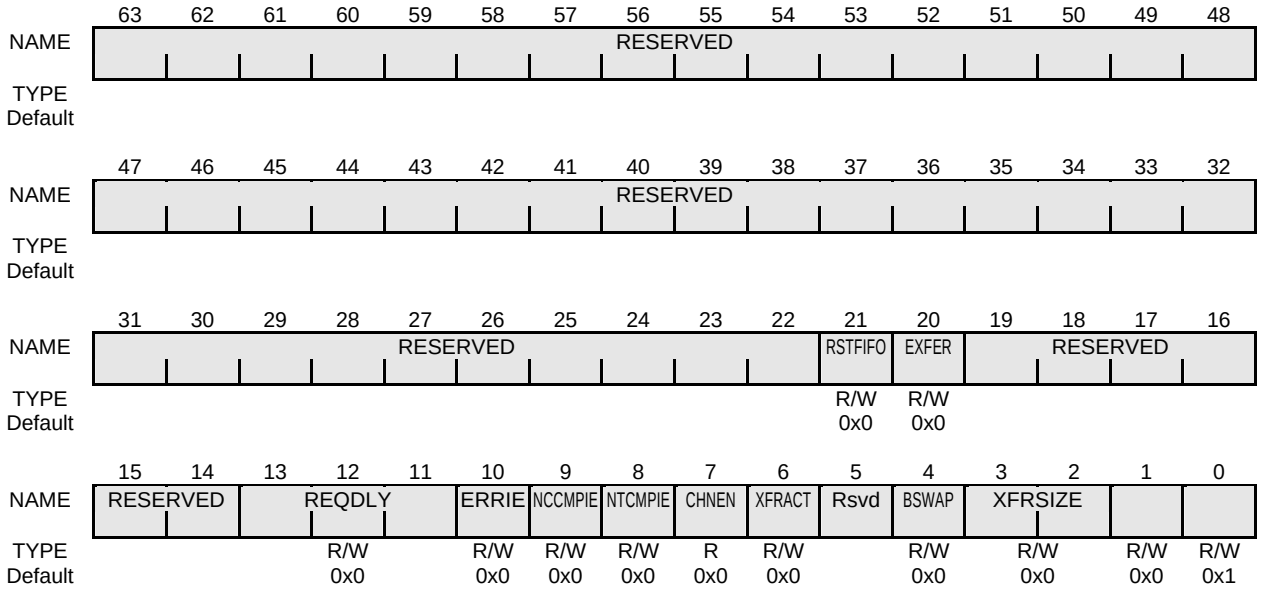


Figure 16-72 PDMAC Control Register

Table 16-74 PDMAC Control Register

Bit	Mnemonic	Field Name	Description	R/W
63:22		Rsvd		—
21	RSTFIFO	Reset FIFO	Reset FIFO (Default: 0x0) Initializes the Read pointer and Write pointer to the FIFO in the PDMAC, and sets the FIFO hold count to “0”. Please use the software to clear this bit when it is set. This is a function for a diagnosis. Usually, it is not used. 1: Performs FIFO reset. 0: Does not perform FIFO reset.	R/W
20	EXFER	Endian Transfer	Endian Transfer (Default: 0x0) Specifies whether to perform Endian transfer. Please use the default as is. Set up EXFER as follows according to a Endian setup of G-Bus. 1: G-Bus in Little Endian 0: G-Bus in Big Endian	R/W
19:14		Rsvd		—
13:11	REQDLY	Request Delay Time	Request Delay (Default: 0x0) G-Bus transactions for DMA transfer must be performed separated at least by the interval this field specifies. 000: Continuously try to perform G-Bus transfer. 001: 16 G-Bus clocks 010: 32 G-Bus clocks 011: 64 G-Bus clocks 100: 128 G-Bus clocks 101: 256 G-Bus clocks 110: 512 G-Bus clocks 111: 1024 G-Bus clocks	R/W
10	ERRIE	Error Detect Interrupt Enable	Interrupt Enable on Error (Default: 0x0) 1: PDMAC generates an error during error detection. 0: PDMAC does not generate an error during error detection.	R/W
9	NCCMPIE	Normal Chain Complete Interrupt Enable	Interrupt Enable on Chain Done (Default: 0x0) 1: PDMAC generates an interrupt when the current chain is complete. 0: PDMAC does not generate an interrupt when the current chain is complete.	R/W
8	NTCMPIE	Normal Data Transfer Complete Interrupt Enable	Interrupt Enable on Transfer Done (Default: 0x0) 1: PDMAC generates an interrupt when the current data transfer is complete. 0: PDMAC does not generate an interrupt when the current data transfer is complete.	R/W

Table 16-74 PDMAC Control Register

Bit	Mnemonic	Field Name	Description	R/W
7	CHNEN	Chain Enable	Chain Enable (Default: 0x0) (Read Only) When the current data transfer is complete, this field reads the next data command Descriptor from the address indicated by the PDMAC Chain Address Register then indicates whether to continue the transfer or not. This bit is only set to “1” when either a CPU Write process or a Descriptor Read process sets a value other than “0” in the PDMAC Chain Address Register. This bit is cleared to “0” if either the Channel Reset bit is set, or “0” is set in the PDMAC Chain Address Register by a CPU Write or a Descriptor Read process. The above 0 value judgement is not performed when the TX49/H4 core stores the upper 32 bits in the PDMAC Chain Address Register. 1: Reads the next data command Descriptor. 0: Does not read the next data command Descriptor.	R
6	XFRACT	Transfer Active	Transfer Active (Default: 0x0) Specifies whether to perform DMA transfer or not. Setting this bit after setting the appropriate value in the register group initiates DMA data transfer. This bit is not set if the PDMAC Count Register value is “0” and the Chain Enable bit is cleared when “1” is written to this bit. Even when a value other than “0” is written to the Chain Address Register, “1” is set to this bit and DMA transfer automatically starts. The above 0 value judgement is not performed when the TX49/H4 core stores the upper 32 bits in the PDMAC Chain Address Register. Data transfer will be stopped after a short delay if this bit is cleared while the data transfer is in progress. This bit is automatically cleared to “0” either when data transfer ends normally or is stopped by an error. Never clear XFRACT by software, because it stops guaranteeing a normal operation,. 1: Perform data transfer. 0: Do not perform data transfer.	R/W
5		Rsvd		—
4	BSWAP	Byte Swap Within DWORD	Swap Bytes in DWORD (Default: 0x0) Specifies whether to perform 32-bit data byte swapping. Please leave this bit at “0” for normal usage. Setting this bit when in the Big Endian mode executes data transfer so the byte order of the 32-bit data on the PCI Bus (which is Little Endian) does not change. 1: Swap the byte order of each 32-bit DWORD data, then transfer. 0: Transfer without swapping the byte order of each 32-bit DWORD data.	R/W
3:2	XFRSIZE	Transfer Size	Transfer Size (Default: 0x0) Specifies the data transfer size in one G-Bus transaction on the G-Bus. 00: 1 DWORD (32-bit) 01: 1 QWORD (64-bit) 10: 4 QWORD (Burst transfer) 11: Rsvd	R/W
1	XFRDIRC	Transfer Direction	Transfer Direction (Default: 0x0) Specifies the DMA data transfer direction. 1: Transfers data from the G-Bus to the PCI Bus. 0: Transfers data from the PCI Bus to the G-Bus.	R/W
0	CHRST	Channel Reset	Channel Reset (Default: 0x1) Resets the DMA channel. This bit must be cleared by the software in advance so the channel can start the data transfer. This reset function is not supported when PDMAC is in operation. Ensure that the Transfer Active (XFARCT) bit in the PDMSTATUS register is cleared prior to resetting the DMA channel. For chained DMA, also ensure either the Abnormal Chain Complete (ACCMP) or Normal Chain Complete (NCCMP) bit in the PDMSTATUS register is set. 1: All logic and State Machines are reset. 0: The channel becomes valid.	R/W

16.4.62. PDMAC Status Register (PDMSTATUS)

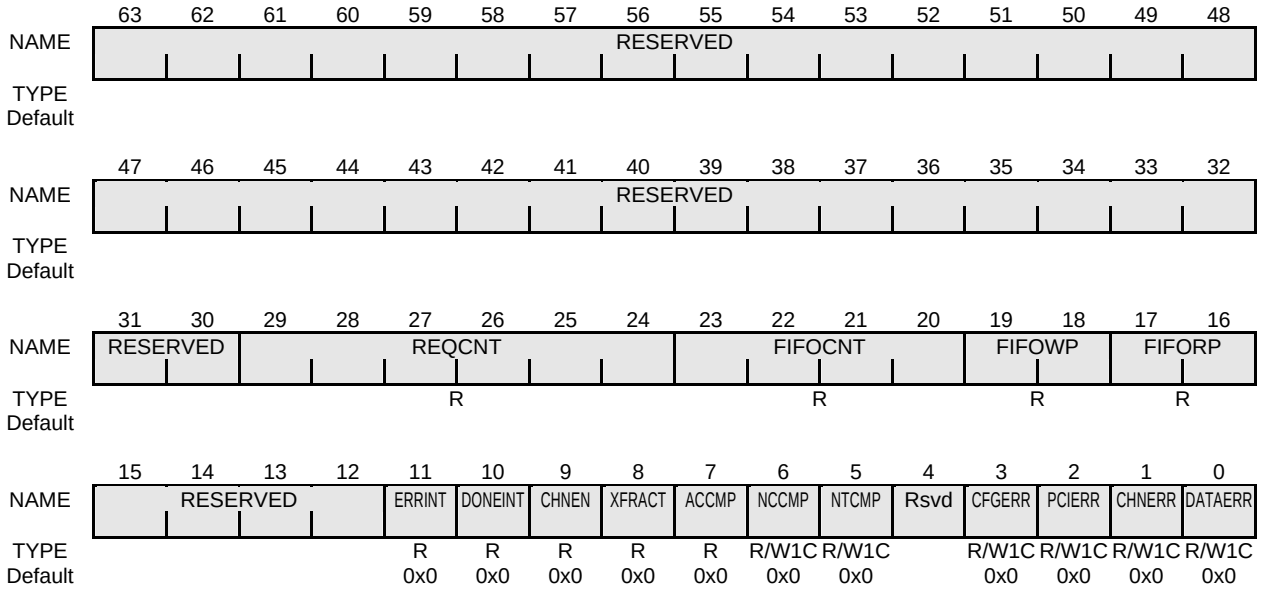


Figure 16-73 Status Register

Table 16-75 Status Register

Bit	Mnemonic	Field Name	Description	R/W
63:30		Rsvd		—
29:24	REQCNT	Request Delay Time Counter	Request Delay Counter (Default: 0x00) This field indicates the request delay time counter value as 16 × *n* when the 6-bit value of this field is *n*.	R
23:20	FIFOCNT	FIFO Hold Count	FIFO Valid Entry Count (Default: 0x0) This field indicates the number of bytes that was written in the FIFO but not yet read. This is a diagnostic function.	R
19:18	FIFOWP	FIFO Write Pointer	FIFO Write Pointer (Default: 0x0) This field indicates the next Write position in the FIFO. This is a diagnostic function.	R
17:16	FIFORP	FIFO Read Pointer	FIFO Read Pointer (Default: 0x0) This field indicates the next Read position in the FIFO. This is a dianostic function.	R
15:12		Rsvd		—
11	ERRINT	Error Interrupt Status	Error Interrupt Status (Default: 0x0) Indicates whether to signal an error interrupt. 1: An error interrupt request exists. 0: No error interrupt request exists.	R
10	DONEINT	Normal Transfer Complete Interrupt Status	Normal Transfer Complete Interrupt Status (Default: 0x0) Indicates whether a Normal Transfer Complete Interrupt is signaled. This bit becomes “1” when either the Normal Chain Complete bit (NCCMP) is set and the Normal Chain Complete Interrupt Enable bit (NCCMPIE) is set, or when the Normal Data Transfer Complete bit (NTCMP) is set and the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) is set. 1: A Normal Transfer Complete Interrupt request exists. 0: No Normal Transfer Complete Interrupt request exists.	R
9	CHNEN	Chain Enable	Chain Enable (Default: 0x0) This bit is a copy of the Chain Enable bit in the PDMAC Control Register.	R
8	XFRACT	Transfer Active	Transfer Active (Default: 0x0) This bit is a copy of the Transfer Active bit in the PDMAC Control Register.	R
7	ACCMP	Abnormal Chain Completion	Abnormal Chain Complete (Default: 0x0) 1: Indicates that the Chain transfer ended in an error state. In other words, this reflects an OR operation of the PDMAC Status Register bits [3:0]. 0: Indicates that no error has occurred in the Chain transfer since the previous error bit was cleared. Note: Bits [3:0] of the PDMAC Status Register must be cleared in order to clear this bit.	R

Table 16-75 Status Register

Bit	Mnemonic	Field Name	Description	R/W
6	NCCMP	Normal Chain Completion	Normal Chain Complete (Default: 0x0) 1: Indicates that the Chain transfer ended in the Normal state. 0: Indicates that Chain transfer has not ended since this bit was previously cleared.	R/W1C
5	NTCMP	Normal Data Transfer Complete	Normal Data Transfer Complete (Default: 0x0) 1: Indicates that the data transfer specified by the PDMAC Register ended in the Normal state. 0: Indicates that data transfer has not ended since this bit was previously cleared.	R/W1C
4		Rsvd		—
3	CFGERR	Configuration Error	Configuration Error (Default: 0x0) 1: Indicates that either the current setting of the control portion in the Control Register and the Address/Count Register are not consistent with each other or the PDMAC stipulation is not being obeyed. DMA transfer stops. 0: Indicates that the current setting of the control portion in the Control Register can be tolerated.	R/W1C
2	PCIERR	PCI Fatal Error	PCI Fatal Error (Default: 0x0) 1: Indicates that an error was signaled on the PCI Bus during the Chain process. 0: Indicates that no error has been signaled on the PCI Bus since this bit was previously cleared.	R/W1C
1	CHNERR	G-Bus Chain Error	G-Bus Chain Bus Error (Default: 0x0) 1: Indicates that a G-Bus error occurrred during the Chain process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the Chain process since this bit was cleared.	R/W1C
0	DATAERR	G-Bus Data Error	G-Bus Data Bus Error (Default: 0x0) 1: Indicates that a G-Bus error occurred during the data transfer process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the data transfer process since this bit was cleared.	R/W1C

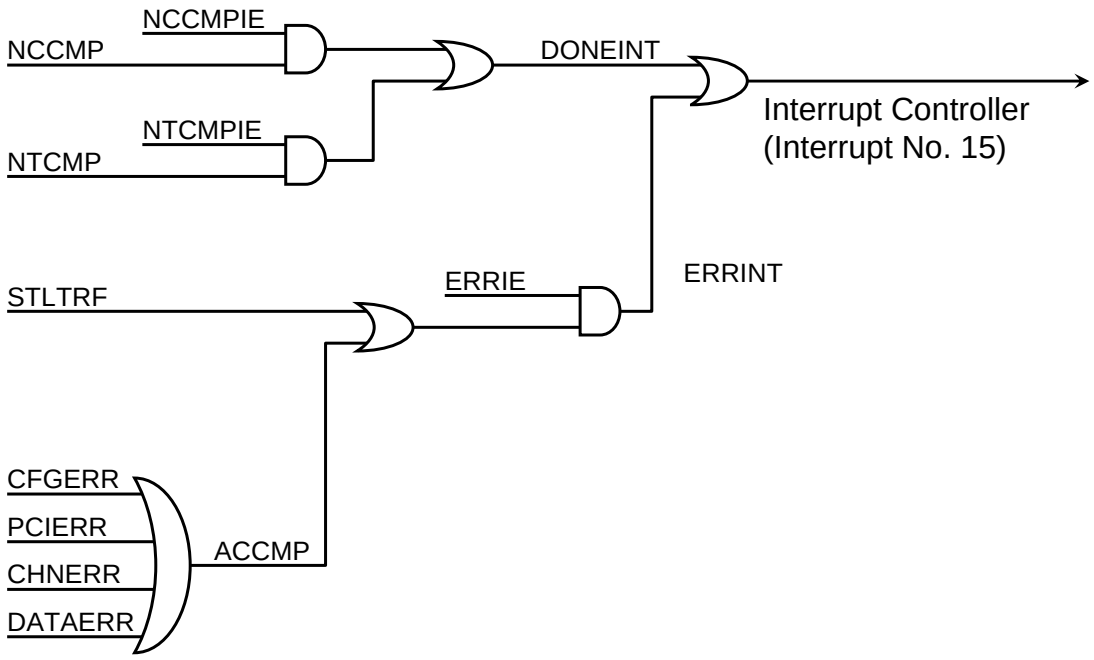


Figure 16-74 PDMAC Interrupt Signaling

16.5. PCI Configuration Space Register

16.5.1. Configuration Space Register Map

The PCI Configuration Space Register is accessed using PCI Configuration cycles by way of an external PCI host device only when in the Satellite mode. Table 16-76 lists registers contained within the PCI Configuration Space Register. The registers in the table with a shaded background are those whose values can be rewritten using EEPROM.

Registers at addresses 0x00 through 0x41 can use the corresponding PCI Controller Control Register to access from the TX49/H4 core when in the Host mode. Please refer to the explanation of the corresponding PCI Controller Control registers for more information about these registers. This section only describes the registers that are accessed from the PCI Configuration Space.

Also, it is possible to read some of the fields in the Status Register and PMCSR register from the Satellite Mode PCI Status Register.

Please refer to the PCI Bus Specifications for more information on the PCI Configuration Register.

Table 16-76 PCI Configuration Space Register

Offset Address	PCI Register WORD (32-bit)				Corresponding Register
	31 24	23 16	15 8	7 0	
00h	Device ID		Vendor ID		PCIID
04h	Status		Command		PCISTATUS
08h	Class Code			Revision ID	PCICCREV
0Ch	BIST	Header Type	Latency Timer	Cache Line Size	PCICFG1
10h	Memory Space 0 Lower Base Address				P2GM0PLBASE
14h	Memory Space 0 Upper Base Address				P2GM0PUBASE
18h	Memory Space 1 Lower Base Address				P2GM1PLBASE
1Ch	Memory Space 1 Upper Base Address				P2GM1PUBASE
20h	Memory Space 2 Base Address				P2GM2PBASE
24h	I/O Space Base Address				P2GIOPBASE
28h	Reserved				—
2Ch	Subsystem ID		Subsystem Vendor ID		PCISID
30h	Reserved				—
34h	Reserved			Capabilities Pointer (Cap_Ptr)	PCICAPPTR
38h	Reserved				—
3Ch	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	PCICFG2
40h	Reserved		Retry Timeout Value	TRDY Timeout Value	G2PTOCNT
44h-DBh	Reserved				—
DCh	Power Management Capabilities (PMC)		Next Item Ptr (Next_Item_Ptr)	Capability ID (Cap_ID)	—
E0h	Reserved	Reserved	Power Management Control/Status Register (PMCSR)		—
E4h-FFh	Reserved				—

16.5.2. Memory Space (m) Lower Base Address (m=0, 1, 2)

Memory Space 0 Lower Base Address (10h)
Memory Space 1 Lower Base Address (18h)
Memory Space 2 Lower Base Address (20h)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	BA [31:29]			BA [28:20]									BA [19:16]			
TYPE	R/W	R/W	R/W	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/W or R/O	R/O	R/O	R/O	R/O
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	BA [15:4]												PF	TYPE		MSI
TYPE	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/W	R/O		R/O
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0		0

Figure 16-75 Memory Space (m) Lower Base Address Register

Table 16-77 Memory Space (m) Lower Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
31:29	BA [31:29]	Base Address [31:29]	Base Address [31:29] These bits are always R/W and defines Base Address of PCI Memory windows 0.	R/W
28:20	BA [28:20]	Base Address [28:20]	Base Address [28:20] or Offset The TYPE of this bit vary depend on the value of corresponding bit in the register field of P2GM(m)PLBASE.MSS. See Table 16-78 below. Corresponding bit = 0 : R/O and this bit is the part of offset address. Corresponding bit = 1 : R/W and this bit is part of Base Address.	R/W or R/O
19:4	BA [19:4]	Base Address	Base Address [19:4] and Offset of PCI Address	R/O
3	PF	Prefetchable	Prefetchable 0 : Indicate that the memory is not prefetchable. 1 : Indicate that the memory is prefetchable.	R/O
2:1	Type	Type	Type (Default = 00) Locate anywhere in lower 4GB.	R/O
0	MSI	Memory Space	Memory Space Indicator 0 : Memory Space	R/O

Table 16-78 Correspondence of BA [28:20] field

MSS[31:20]	31	30	29	28	27	26	25	24	23	22	21	20	n	Memory Size
0xE00	1	1	1	0	0	0	0	0	0	0	0	0	29	512 MB
0xF00	1	1	1	1	0	0	0	0	0	0	0	0	28	256 MB
0xF80	1	1	1	1	1	0	0	0	0	0	0	0	27	128 MB
0xFC0	1	1	1	1	1	1	0	0	0	0	0	0	26	64 MB
0xFE0	1	1	1	1	1	1	1	0	0	0	0	0	25	32 MB
0xFF0	1	1	1	1	1	1	1	1	0	0	0	0	24	16 MB
0xFF8	1	1	1	1	1	1	1	1	1	0	0	0	23	8 MB
0xFFC	1	1	1	1	1	1	1	1	1	1	0	0	22	4 MB
0xFFE	1	1	1	1	1	1	1	1	1	1	1	0	21	2 MB
0xFFF	1	1	1	1	1	1	1	1	1	1	1	1	20	1 MB

16.5.3. IO Space Base Address (24h)

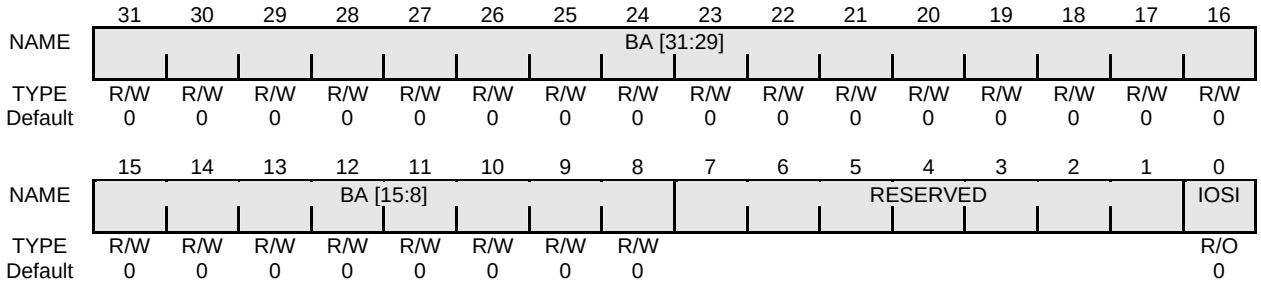


Figure 16-76 IO Space Base Address Register

Table 16-79 IO Space Base Address Register

Bit	Mnemonic	Field Name	Description	R/W
31:8	BA [31:8]	Base Address [31:8]	IO SpaceBase Address [31:8] Define the IO Space Base Address	R/W
7:1	RESERVED	RESERVED	RESERVED	R/O
0	IOSI	IO Space	IO Space Indicator 1 : IO Space	R/O

16.5.4. Capability ID Register (Cap_ID) 0xDC

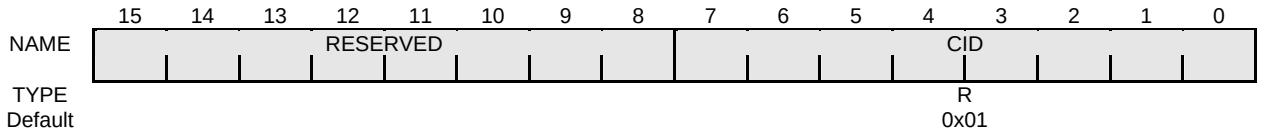


Figure 16-77 Capability ID Register

Table 16-80 Capability ID Register

Bits	Mnemonic	Field Name	Description	R/W
15:8		Reserved		—
7:0	CID	Capability ID	Capability ID (Default: 0x01) Indicates that a list is the link list of the Power Management Register.	R

16.5.5. Next Item Pointer Register (Next_Item_Ptr) 0xDD

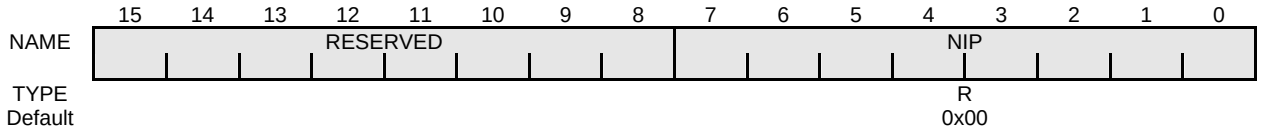


Figure 16-78 Next Item Pointer Register

Table 16-81 Next Item Pointer Register

Bits	Mnemonic	Field Name	Description	R/W
15:8		Reserved		—
7:0	NIP	Next Item Pointer	Next Item Pointer (Default: 0x0) This is the Next Item pointer. Indicates the end of a list.	R

16.5.6. Power Management Capability Register (PMC) 0xDE

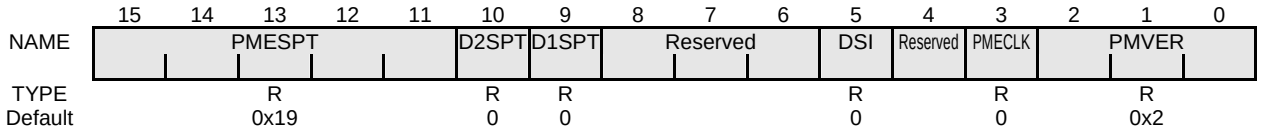


Figure 16-79 PMC Register

Table 16-82 PMC Register

Bit	Mnemonic	Field Name	Description	R/W
15:11	PMESPT	PME Output Support	PME_ Support (Fixed Value: 0x09) Indicates that the PME* signal can be output from the state where the bit is set to “1”. Bit 15: Can output the PME* signal from the D3cold state. Bit 14: Can output the PME* signal from the D3hot state. Bit 13: Can output the PME* signal from the D2 state. Bit 12: Can output the PME* signal from the D1 state. Bit 11: Can output the PME* signal from the D0 state. Note: With the TX4939 PCI Controller, it is possible to output the PME* signal from the D0 and the D3hot states.	R
10	D2SPT	D2 Support	D2_Support (Fixed Value: 0) 0: Indicates that the D2 state is not supported.	R
9	D1SPT	D1 Support	D1_Support (Fixed Value: 0) 0: Indicates that the D1 state is not supported.	R
8:6		Reserved		—
5	DSI	DSI	DSI (Fixed Value: 0) 0: Indicates that Device Specific Initialization is not required.	R
4		Reserved		—
3	PMECLK	PME Clock	PME Clock (Fixed Value: 0) 0: Indicates that the PCI Clock is not required to assert the PME* signal.	R
2:0	PMVER	Power Management I/F Version	Version (Fixed Value: 0x2) 2: Indicates compliance with "PCI Power Management Interface Specification" Version 1.1.	R

16.5.7. Power Management Control/Status Register (PMCSR) 0xE0

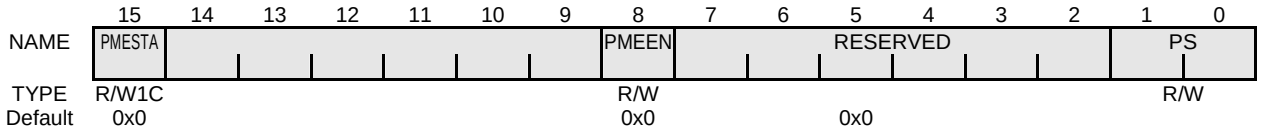


Figure 16-80 PMCSR Register

Table 16-83 PMCSR Register

Bit	Mnemonic	Field Name	Description	R/W
15	PMESTA	PME Status	PME_Status (Default: 0x0) Indicates the existence of a PME (Power Management Event) . 1: There is a PME. 0: There is no PME. The value of this bit becomes “1” when Writing a “1” to the PME bit (P2GCFG.PME) of the P2G Configuration Register. This bit is cleared when the Host Bridge writes a “1”. It is possible to signal a PME* Clear Interrupt at this time.	R/W1C
14:9		Reserved		—
8	PMEEN	PME Enable	PME_En (Default: 0x0) Sets PME* signal assertion to enable or disable. 1: Enables assertion of the PME* signal. 0: Disables assertion of the PME* signal. The PME_En set bit of the P2G Status Register (P2GSTATUS.PMEES) is set when this bit is set. At this time, it is possible to signal the PME_En set interrupt.	R/W
7:2		Reserved		—
1:0	PS	Power State	PowerState (Default: 0x0) Sets the Power Management state. The Power Management State Change bit (P2GSTATUS.PMSC) of the P2G Status Register is set when the value of this field is changed. It also becomes possible to generate a Power State Change Interrupt at this time. The TX4939an read the value of this field from the PowerState field (PCISSTATUS.PS) of the Satellite Mode PCI Status Register. 00b: D0 (no change) 01b: D1 :Reserved 10b: D2 :Reserved 11b: D3hot	R/W

16.6. A Malfunction of PCI Controller

16.6.1. Outline of the problem

If PCI to G-Bus transfer (PCI initiator read) by PDMAC, PCI read access by G-Bus bus master (B), and PCI bus access (read or write) by G-Bus bus master (A) are executed simultaneously, PCI read access by G-Bus bus master (B) could not finish.

16.6.2. Condition of the problem occurrence

(1) PDMAC PCI Initiator reads data from a device on PCI bus. (Read from PCI bus and write to G-Bus.)

(2) If an access (read or write) requirement (*) to a device on PCI bus by G-Bus bus master (A) is issued before PDMAC PCI initiator's read transfer (1) finish, the required access by G-Bus bus master (A) will be retried on G-Bus.

By the way, the retry happened by overflow of PCI access command queue could cause the problem. The followings are candidates to cause that retry.

The access requirement (*) by G-Bus bus master (A) is :

- *Burst Write Transfer*
- *Single write transfer in case of that three times single write transfers are executed before PDMAC PCI initiator read transfer (1)*
- *(Single/Burst) read transfer in case of that three times single write transfers are executed before PDMAC PCI initiator read transfer (1)*

(3) PDMAC initiator read transfer (1) finish on PCI bus.

(4) Though read transfer on PCI bus by G-Bus bus master (B) is executed, it will be retried, and then the retry can't be accepted. Then retry → refuse the request → retry → refuse the request → … This iteration will be continued permanently.

16.6.3. Work-around

In case of that PCI to G-Bus transfer (PCI initiator read) by PDMAC and PCI read access by G-Bus bus master (B) are executed simultaneously, any PCI bus access (read or write) is forbidden.

(Any problem doesn't occur if the number of G-Bus bus master which executes PCI read access simultaneously with PDMAC is one, or if the G-Bus bus masters execute PCI write access regardless of the number of the G-Bus bus masters).

Chapter 17. ATA100 ATAP Interface

17.1. Overview

This document is the functional specification for the ATA Host Interface Controller used in TX4939.

- *Compliant with ATA/ATAPI-6.*
- *Can interface with the TX49 G-Bus.*
- *Can become the Bus Master and transfer data to external memory.*
- *Data bus widths of 64 bits, can select Byte Big/Little Endian.*
- *Can select incrementing/decrementing/fixed address of the transfer address.*
- *Realizes Auto DMA mode. (When issuing a DMA command, settings such as the transfer direction and transfer start are unnecessary. After the command is issued, this controller can realize data transfer and only has to wait for a Transfer End interrupt.)*
- *Sets up the reset timer.*
- *Has 16 × 64-bit transmit FIFO and 16 × 64-bit receive FIFO. Can absorb the transfer rate difference between when transferring to/from the device and when transferring to/from the external memory.*
- *Can issue Timing Error interrupts.*
- *Can Support Scatter Gather DMA.*

The following table lists the transfer modes that this interface supports.

Operating Frequency	100 MHz
Supported Mode (PIO)	Mode 0 (Cycle Time: 600 nsec)
	Mode 1 (Cycle Time: 383 nsec)
	Mode 2 (Cycle Time: 240 nsec)
	Mode 3 (Cycle Time: 180 nsec)
	Mode 4 (Cycle Time: 120 nsec)
Supported Mode (Multiword DMA)	Mode 0 (Cycle Time: 480 nsec)
	Mode 1 (Cycle Time: 150 nsec)
	Mode 2 (Cycle Time: 120 nsec)
Supported Mode (Ultra DMA)	Mode 0 (Cycle Time: 240 nsec)
	Mode 1 (Cycle Time: 160 nsec)
	Mode 2 (Cycle Time: 120 nsec)
	Mode 3 (Cycle Time: 90 nsec)
	Mode 4 (Cycle Time: 60 nsec)
	Mode 5 (Cycle Time: 40 nsec)

17.2. Block Diagram

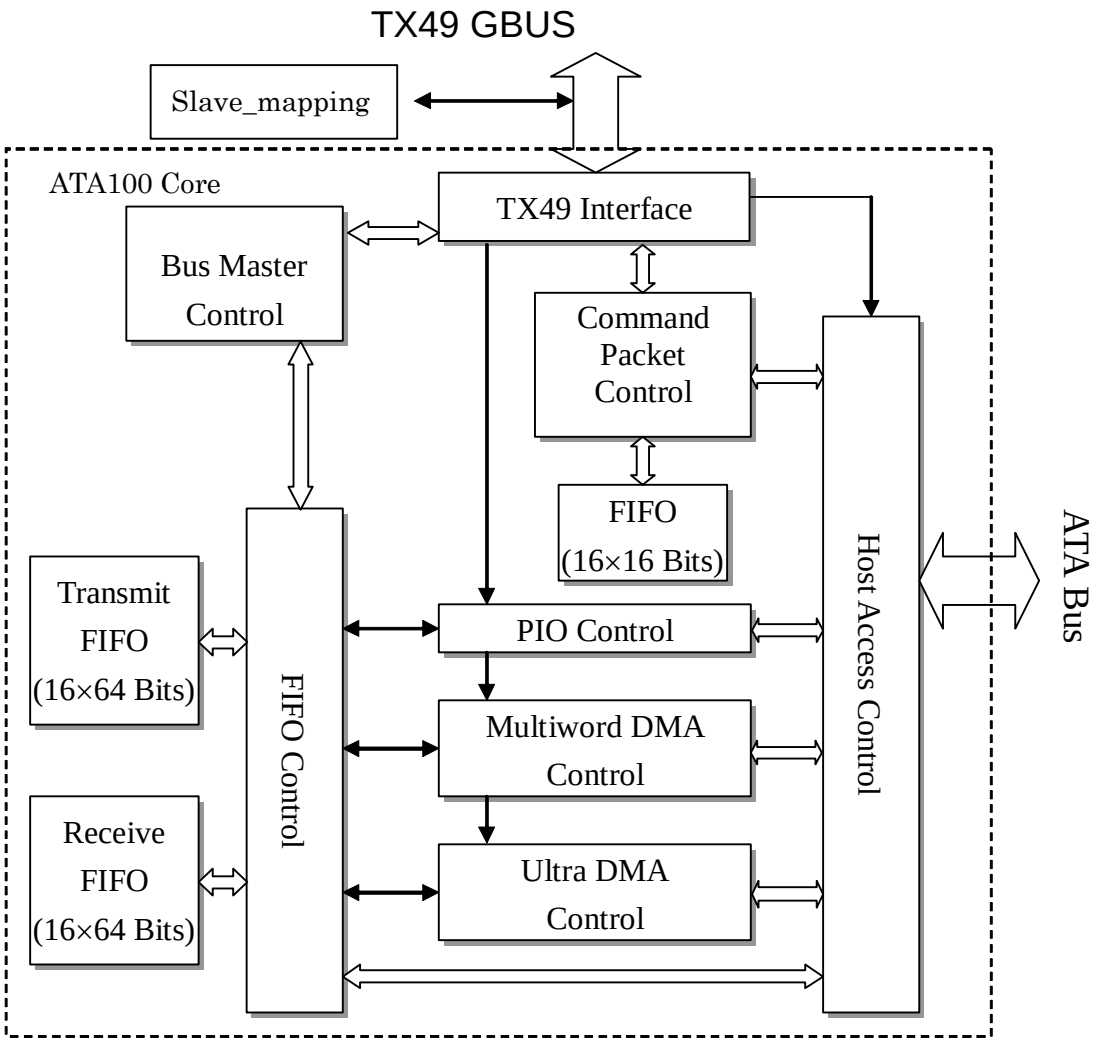


Figure 17-1 ATA100 Controller Block Diagram

17

17.3. Register List

17.3.1. Register address map

Use byte, half-word, or word load/store to access register with 8, 16, or 32 bits word, respectively

Gbus Address		Width (bit)	Access Type	Register Name	Location/Comments
ATA0/1					
LE	**BE**				
000	006	16	R/W	DATA	ATA100 Core/ATA Shadow register *1
001	006	8	R	Error	ATA100 Core/ATA Shadow register *1
001	006	8	W	Feature	ATA100 Core/ATA Shadow register *1
002	005	8	R/W	Sector Count	ATA100 Core/ATA Shadow register
003	004	8	R/W	LBA0	ATA100 Core/ATA Shadow register
004	003	8	R/W	LBA1	ATA100 Core/ATA Shadow register
005	002	8	R/W	LBA2	ATA100 Core/ATA Shadow register
006	001	8	R/W	Device/Head	ATA100 Core/ATA Shadow register *3
007	000	8	R	Status	ATA100 Core/ATA Shadow register
007	000	8	W	Command	ATA100 Core/ATA Shadow register
400	406	16	N/A	Reserved	N/A writes ignored
402	405	8	R	Alt Statusl	ATA100 Core/ATA Shadow register
402	405	8	W	Device Control	ATA100 Core/ATA Shadow register
403	404	8	N/A	Reserved	N/A writes ignored
800	807	8	R/W	DMA Command	Slave mapping/Mapped to ATA100 Core*4
801	806	8	N/A	Reserved	N/A writes ignored
802	805	8	R/W	DMA Status	Slave mapping *4
803	804	8	N/A	Reserved	N/A writes ignored
804	800	32	R/W	PRD table Ptr	Slave mapping
808	80C	32	N/A	Reserved	N/A writes ignored
80C	808	32	N/A	Reserved	N/A writes ignored
c00	c06	16	R/W	Sys Control	Slave mapping*3,*5/Mapped to ATA100 Core
c08	c0e	16	R/W	Xfer Word Cnt 1	Slave mapping*6/Mapped to ATA100 Core
c0a	c0c	16	R/W	Xfer Word Cnt 2	Slave mapping*6/Mapped to ATA100 Core
c10	c16	16	R/W	Sector Cnt	ATA100 Core
c18	c1e	16	R/W	Xfer Start Lower Address	ATA100 Core
c20	c26	16	R/W	Xfer Start Upper Address	ATA100 Core
c28	c2e	16	R/W	Additional Control	ATA100 Core
c30	c36	16	R/W	Lower Burst Counter	ATA100 Core
c38	c3e	16	R/W	Upper Burst Count	ATA100 Core
c88	c8e	16	R/W	PIO Access Address	ATA100 Core
c90	c96	16	R/W	Host Reset Timer	ATA100 Core
c98	c9e	16	R/W	Interrupt Control	ATA100 Core
cb8	cbe	16	WO	ATAPI Packet Command	ATA100 Core
cc0	cc6	16	RO	Bus Xfer Count High	ATA100 Core
cc8	cce	16	RO	Bus Xfer Count Low	ATA100 Core
cd0	cd6	16	R/W	ATA Device Timing Error	ATA100 Core
cd8	cde	16	R/W	Packet Xfer Control	ATA100 Core
ce0	ce6	16	R/W	Xfer Start Top Address	ATA100 Core

Note :

1 Data register in ATA shadow register is 16 bits; all others are 8 bits. All ATA shadow registers are accessed on Gbus byte boundaries. The Data register should be accessed with a word address of 0 and the two least significant byte enables asserted. This will read or write the full 16-bit data register. If only the least significant byte enable is asserted, the high-order byte of the data to or from the data register will be set to zero. If the Gbus word address is 0 and byte enable 1 is the least significant one asserted, this is an access of the Error/Feature register.

2 The ATA Controller does not support a 64-bit read/write to registers. So the read/write to DMA Command, Status and PRD Table Pointer can not be combined. Use 32-bit load/store to read/write to PRD Table Pointer.

3 Every time user does a write to the Device/Head register, user must do a write to the System Control Register for next command/data transfer immediately after the write to the Device/Head register.

4 A write to the Command register in the slave mapping module will cause to write the new value of the Direction bit in the ATA100 Core and to write the new value of the Start bit in the ATA100 Core. Since the Status and Command registers are within the same 32-bit word, it is possible to clear the status bits and write the R/W and Start bit in the same Gbus write operation. But please take consideration on access order. The status bits will be cleared first, then the R/W bit will be written to the correct value and finally the start bit will be written. When both the R_OR_WCTR bit (direction bit) and START_STOPBM bit (start bit) are written in the same Gbus write cycle, the R_OR_WCTR bit will be written before the START_STOPBM bit.

5 The write to Device/Head register clears any other bits in System Control register except command and data transfer mode field.

6 There are also 2 Transfer Word Count registers, 1 and 2. The mapping module maintains an internal bit to keep track the value of DEV bit of Device/Head ATA device register. If this bit is 0, it writes the transfer word count 1, otherwise it writes the transfer word count 2 to transfer word count register in ATA100 Core.
But it is impossible to read Transfer Word Counter register in ATA100 Core itself.

7 All ATA100 Core registers are 16 bits. The addresses of these registers are on 64-bit boundaries. Use only load/store half word (16-bits) to access these registers. The ATA100 Core does not support byte access to these registers.

8 User needs to specify the burst count registers as described in section 17.3.3 for variable burst length.

9 If the PRD_NOT_EN in DMA Command register is on, the Transfer Start Address needs to be specified.

17.3.2. Registers in slave mapping

17.3.2.1. DMA Command Register (800h)

Bit:	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name:	Reserved								Reserved			PRD_NOT_EN	R_OR_WCTR	Reserved		START_STOPBM
R/W:	RO								RO			R/W	RW	RO		RW
Default	00000000								000			0	0	00		0

Bits	Mnemonic	Field Name	Description
D7:D5		Reserved	
D4	PRD_NOT_EN	PRD_NOT_EN	PRD not enable, if this is '1' the DMA will use the Transfer Start Address for Source or Destination address as described in ATA100 Core
D3	R_OR_WCTR	R_OR_WCTR	This bit specifies a direction of data transfer 0 : Host → Device 1 : Device → Host This bit is mapped to Direction bit in ATA100 Core
D2:D1		Reserved	
D0	START_STOPBM	START_STOPBM	Data transfer starts by setting this bit to 1. Data to be transferred is specified in both Transfer Word Counter register and Sector Counter register. This bit will be cleared when transfer terminates. And following is the clear condition for this bit. a) Power On Reset b) Soft Reset (bit[15] of System Control Register) c) Termination of specified Data transfer including command packet Data Transfer termination except above condition does not clear this bit. This bit is mapped to Start bit in ATA100 Core

Figure 17-2 Command Register

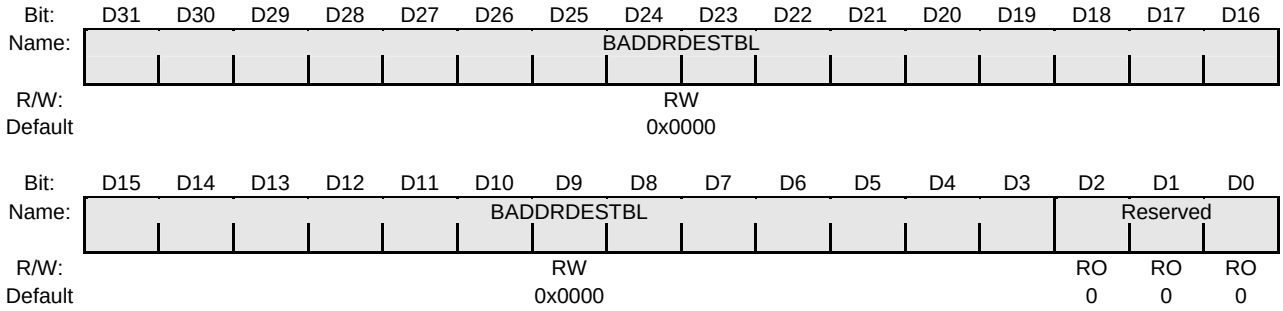
17.3.2.2. DMA Status Register (802h)

Bit:	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name:	Reserved									DR1_DMACP	DR0_DMACP	Reserved		INT_IDE	ERROR	BM_ACT
R/W:	RO									RW	RW	RO		RWC	RWC	RO
Default	000000000									0	0	0		0	0	0

Bits	Mnemonic	Field Name	Description
D7		Reserved	Reserved Fixed to "0".
D6	DR1_DMACP	DR1_DMACP	DR1_DMACP [Drive1 (Slave) DMA Capable] Is set to "1" when a slave drive can perform DMA.
D5	DR0_DMACP	DR0_DMACP	DR0_DMACP [Drive0 (Master) DMA Capable] Is set to "1" when the master drive can perform DMA.
D4:D3		Reserved	
D2	INT_IDE	INT_IDE	INT_IDE (RWC) [Interrupt] Is "1" when data transfer completes. This bit is cleared by writing "1" to it. When this bit is set to '1', the following bits of the ATA Interrupt Controller Register will be reset: bits **[15:8]** (Mask Address Error INT, Mask Reach Multiple INT, Mask DEV Timing Error, Mask Ultra DMA DEV Terminate, Mask Timer INT, Mask Bus Error, Mask Data Transfer End, Mask Host INT), and bits **[1:0]** (Data Transfer End, Host INT).
D1	ERROR	ERROR	ERROR (RWC) Is set to "1" when an error occurs during data transfer to or from the system memory. This bit is cleared by writing a "1" to it. Refer to the status register in the configuration space for details about the error.
D0	BM_ACT	BM_ACT	BM_ACT(RO) [Bus Master IDE Active] Is set to "1" when the command register start bit is set to "1." This bit is set to "0" either when the start bit is set to "0" (aborted) or when the last transfer in the current cycle is executed and the area descriptor EOT is set (terminated normally).

Figure 17-3 Status Register

17.3.2.3. PRD Table Pointer Register (804h–807h)



Bits	Mnemonic	Field Name	Description
D31:D3	BADDRDESTBL	BADDRDESTBL	BADDRDESTBL Specifies the upper 29 bits of the PRD berth address. This address must be specified in 8-Byte alignment. The Descriptor Table must not cross a 64K boundary in memory.
D2:D0		Reserved	

Figure 17-4 PRD Table Pointer Register

17.3.2.3.1. PRD

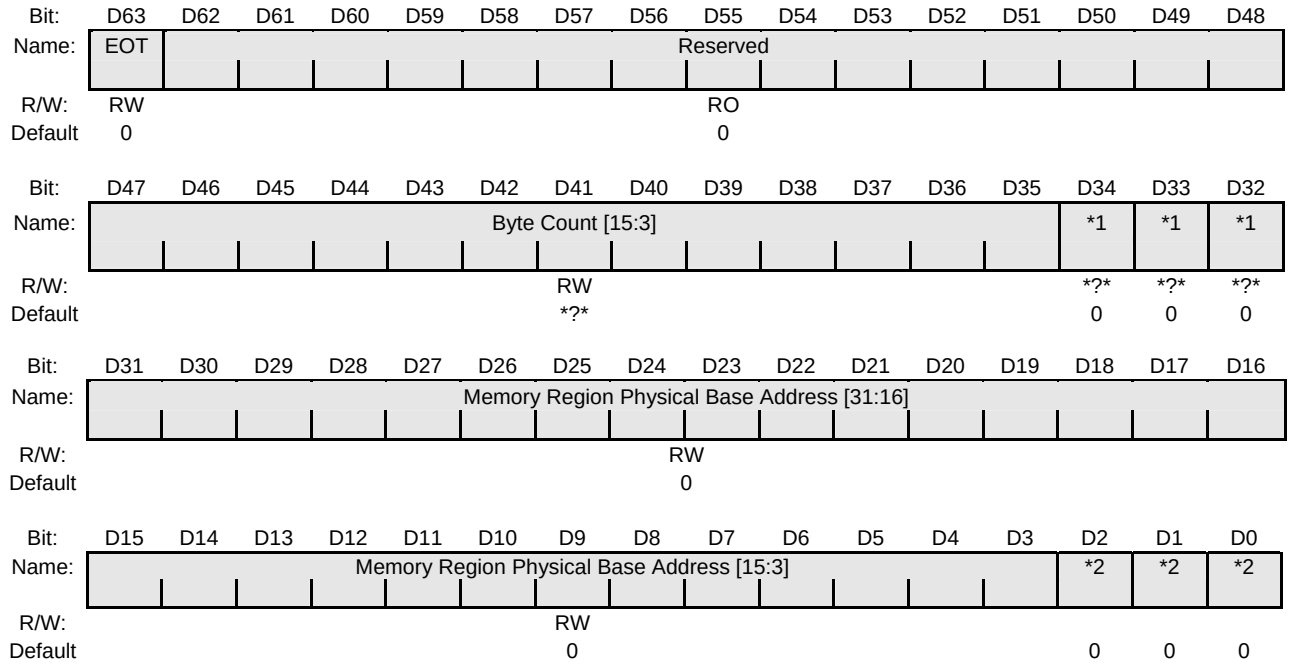
The PRD (Physical Region Descriptor) is a table that is used to define the physical memory area used in bus master transfers. It is provided for handling situations where the memory area can only be used in small divided sections when performing bus master transfer. The minimum configuration unit of a PRD is two double-words, with each PRD located at contiguous addresses so that they can be read by the hardware. The software prepares PRDs in a memory area.

17.3.2.3.2. PRD Table

The PRD table has its PRD data arranged in contiguous locations beginning with the address specified by the base address. One PRD consists of two double-words (64 bits). The base address is specified by an I/O register.

Offset	
0x00	PRD
0x08	PRD
0x10	PRD
0x18	PRD (Last data)

17.3.2.3.3. PRD Configuration



*1 : Byte Count[0]
*2 : Memory Region Physical Base Address[0]

Bits	Mnemonic	Field Name	Description
D63	EOT	EOT	EOT This bit designates the end of the table. Set this bit to “1” for the last entry of the table. The DMA controller checks this bit and terminates DMA processing when it is “1.”
D62:D48		Reserved	
D47:D35	Byte Count	Byte Count	Byte Count [15:3] These bits specify the size of the memory area. The 3 least significant bits of this byte count must be “0” since the address needs to be double word-aligned (8 bytes)
D31:D3	Memory Region Physical Base Address	Memory Region Physical Base Address	Memory Region Physical Base Address [31:3] These bits specify the start address of the memory area in which to store data during DMA transfer. The 3 least significant bits of this address must be “0” since the address needs to be double word-aligned (8-bytes).

Figure 17-5 PRD Configuration

17.3.2.4. System Control Register (C00h)

This register is the same as System Control register in ATA100 CORE. The register is shared with both Master device and Slave device.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Soft Reset	FIFO Reset	PDIAGN	DASPN	ATA Hard Reset	Command Transfer Mode Select		
W	W	R	R	R/W	R/W	R/W	R/W
0	0	1	1	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Data Transfer Mode Select				Break Enable	End Break	Auto DMA Enable	Access Now
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
0	0	0	0	0	0	0	0

Figure 17-6 System Control Register

Table 17-1 System Control Register

BIT	NAME	Description
Bit[15]:	Soft Reset	This bit resets the controller.
Bit[14]:	FIFO Reset	This bit resets the FIFO control block. You do not have to set this bit during normal operation. (This bit is only used during the controller test.) Note: When you set this bit, after the G-Bus BSTARTN signal is asserted, do not execute register read operation until 7-12 GBUSCLK cycles pass.
Bit[13]:	PDIAGN	You can use this bit to monitor the PDIAGN signal on the ATA Bus.
Bit[12]:	DASPN	You can use this bit to monitor the DASPN signal on the ATA Bus.
Bit[11]:	ATA Hard Reset	You can activate a Reset signal that is sent to the ATA Bus by setting this bit. This bit is automatically cleared when the time set by the Host Reset Timer Register elapses. However, you must take note that after this bit is set, the device will ignore any commands that are set until this bit is cleared.
Bit[10:8]	Command Transfer Mode Select [2:0]	You can use this field to select the transfer mode that you will use when accessing ATA Registers such as command registers on the target device. Access to the Data Register from TX4939 follows the Data transfer timing. 000: Selects PIO Mode 0 (Minimum Cycle Time: 600 ns) 001: Selects PIO Mode 1 (Minimum Cycle Time: 383 ns) 010: Selects PIO Mode 2 (Minimum Cycle Time: 330 ns) 011: Selects PIO Mode 3 (Minimum Cycle Time: 180 ns) 100: Selects PIO Mode 4 (Minimum Cycle Time: 120 ns) 101, 110, 111: Reserved The minimum cycle times mentioned above are specific values Example: The following transfer modes are selected when using Command Transfer Mode Select to select PIO Mode 3 and Data Transfer Mode Select to select Ultra DMA Mode 1. 1. ATA registers except the Data Register → PIO Mode 3 2. Data Register → PIO Mode 3 (Timing is the same as when using Data Transfer Mode Select to select PIO Mode 3.) 3. Data Transfer → Ultra DMA Mode 1

Table 17-1 System Control Register

BIT	NAME	Description
Bit[7:4]:	Data Transfer Mode Select [3:0]	You can use this field to select the data transfer mode that the system uses. 0000: Selects PIO Mode 0 (Cycle Time: 600 ns) 0001: Selects PIO Mode 1 (Cycle Time: 383 ns) 0010: Selects PIO Mode 2 (Cycle Time: 240 ns) 0011: Selects PIO Mode 3 (Cycle Time: 180 ns) 0100: Selects PIO Mode 4 (Cycle Time: 120 ns) 0101: Selects Multiword DMA Mode 0 (Cycle Time: 480 ns) 0110: Selects Multiword DMA Mode 1 (Cycle Time: 150 ns) 0111: Selects Multiword DMA Mode 2 (Cycle Time: 120 ns) 1000: Selects Ultra DMA Mode 0 (16.6) (Cycle Time: 240 ns) 1001: Selects Ultra DMA Mode 1 (25.0) (Cycle Time: 160 ns) 1010: Selects Ultra DMA Mode 2 (33.3) (Cycle Time: 120 ns) 1011: Selects Ultra DMA Mode 3 (44.4) (Cycle Time: 90 ns) 1100: Selects Ultra DMA Mode 4 (66.6) (Cycle Time: 60 ns) 1101: Selects Ultra DMA Mode 5 (100.0) (Cycle Time: 40 ns) 1110-1111: Reserved The cycle times mentioned above are specific values.
Bit[3]:	Break Enable	You can suspend PIO Mode data transfer. When setting this bit, you can suspend in units of the set sector count data transfer that was initiated by the READ MULTIPLE, READ MULTIPLE EXT, WRITE MULTIPLE, or WRITE MULTIPLE EXT command, and you can suspend in 512-byte units data transfer that was initiated by all other commands. Then an interrupt is issued. You can restart transfer by setting bit [2](End Break). In other words, set bit [3] when initializing and set both bit [2] and bit [3] when restarting. While data transfer is suspended by this bit, you can access the device ATA Registers.
Bit[2]:	End Break	When bit [3](Break Enable) has interrupted transfer as described above, setting this bit makes it possible to clear the suspend state. During a read, this bit is the status bit that indicates transfer is being suspended.
Bit[1]:	Auto DMA Enable	You can automatically set the transfer direction or initiate transfer when setting this bit while using one of the following commands to execute data transfer: READ DMA, READ DMA EXT, WRITE DMA, WRITE DMA EXT. However, the Data Transfer Mode selected by bits [7:4] must select Multiword DMA or Ultra DMA mode.
Bit[0]:	Access Now	The internal State Machine indicates that a device is currently being accessed.

<CAUSION>
The write to the register by the Device/Head register may cause an unexpected function by write wrong data to the register. So please rewrite to the System Control register after write to the Device/Head register to secure write to System Control register in ATA100 Core.

17.3.2.5. Transfer Word Count 1 Register (C08h):

This register sets the transfer word count in ATA100 Core for a master device. The register description is the same as the transfer word count register in ATA100 Core below.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Transfer Word Count[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Transfer Word Count[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-7 Transfer Word Count 1 Register

This register sets the transferred word count for each sector. When the size of 1 sector transferred to the device is 512 bytes, you do not need to change this register from its default state. Set bits [10:8] to the following values: 001 → 512 bytes; 010 → 1024 bytes; 100 → 2048 bytes.

Calculate the amount of data transferred by a single command as shown below.

Total Transfer Word Count = Transfer Word Count [15:0] × Sector Count [15:0]

When any sector data transfer starts, the transfer word number for each sector to be performed is set in this register and this number is decremented when each transfer is performed.

Note 1) You must set “00h” to the lower order of the Transfer Word Count when setting a value other than 1 to the Sector Count.

17.3.2.6. Transfer Word Count 2 Register (C0Ah):

This register sets the transfer word count in ATA100 Core for a slave device. The register description is the same as the transfer word count register in ATA100 Core below.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Transfer Word Count[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Transfer Word Count[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-8 Transfer Word Count 2 Register

This register sets the transferred word count for each sector. When the size of 1 sector transferred to the device is 512 bytes, you do not need to change this register from its default state. Set bits [10:8] to the following values: 001 → 512 bytes; 010 → 1024 bytes; 100 → 2048 bytes.

Calculate the amount of data transferred by a single command as shown below.

Total Transfer Word Count = Transfer Word Count [15:0] × Sector Count [15:0]

When any sector data transfer starts, the transfer word number for each sector to be performed is set in this register and this number is decremented when each transfer is performed.

Note 1) You must set “00h” to the lower order of the Transfer Word Count when setting a value other than 1 to the Sector Count.

17.3.3. ATA100 Core Control Registers

17.3.3.1. Sector Count Register (C10h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Sector Count[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Sector Count[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	1

BIT	NAME	Description
Bit[15:0]:	Sector Count	You can use this register to determine the number of sectors to transfer. The Transfer Word Count Register determines the transfer byte count for each sector. Also, setting “0000h” to this register results in the same operation as if “10000h” was set. When transfer starts, the total sector number of transfers to be performed is set in this register and this number is decremented when each sector transfer is performed.

17.3.3.2. Transfer Start Lower Address Register (C18h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Transfer Start Address[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Transfer Start Address [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This register is used only if PRD_NOT_EN in DMA Command Register is ‘1’. The 3 LSB’s should be 0.
This register sets the lower read/write start address of the transfer that this controller outputs to the G-Bus.

17.3.3.3. Transfer Start Upper Address Register (C20h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Transfer Start Address[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Transfer Start Address [23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This register is used only if PRD_NOT_EN in DMA Command Register is ‘1’.
This register sets the upper read/write start address of the transfer that this controller outputs to the G-Bus.

17.3.3.4. Additional Control Register (C28h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used		Not Used		Bus Mode		All Pause	All Idle
R	R	R/W	R/W	R	R	R	R
1	1	0	0	1	0	0	1
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	Transfer Pause	Not Used	Byte Swap	Endian Control		Not Used	
-	R/W	-	R/W	R/W	R/W	R	R
0	0	0	GENDIAN	0	0	0	1

Figure 17-9 Additional Control Register

Table 17-2 Additional Control Register

BIT	NAME	Description
Bit[13:12]:	Not Used	User should write only 00 to this field. Other values will be undefined
Bit[11:10]:	Bus Mode	This field determines the bus width when transferring data as the Bus Master. 11: Reserved 10: 64-bit bus. Only support 64-bit bus. Note: Data transfer rate on the ATA Bus is influenced by these bits.
Bit[9]:	All Pause	You can use this bit to acknowledge that the receiving of data from the device has been paused. Also, this bit is only set when bit [6](Transfer Pause) is set. 1: Transfer had been paused. 0: Normal state
Bit[8]:	All Idle	You can use this bit to acknowledge that the sending/receiving of data to/from the device has been stopped. Setting bit [6](Transfer Pause) does not set this bit. 1: Transfer has stopped 0: Normal state
Bit[7]:	-	Reserved
Bit[6]:	Transfer Pause	You can pause transfer from the device by setting this bit. 1: Pause transfer 0: Normal state
Bit[5]:	Not Used	Reserved
Bit[4];	Byte Swap Note 1	1: Swap all the bytes on 64 bit GBUS DATA 0: No Swap. Default is the System GBUS Endian control bit Note: This bit is applied on GBUS DATA in/out to ATA control for data transfer only by ATA DMA
Bit[3:2]:	Endian Control Note 1	This field determines the Endian for when transferring data as the Bus Master. The following example shows a 64-bit bus with a correlation between the ATA Bus data and G-Bus data of `aabb_ccdd_eeff_ggiih`: 11: 1st Data `ccddh` 2nd Data `aabbh` 3rd Data `ggiih` 4th Data `eeffh` 10: 1st Data `aabbh` 2nd Data `ccddh` 3rd Data `eeffh` 4th Data `ggiih` 01: 1st Data `eeffh` 2nd Data `ggiih` 3rd Data `aabbh` 4th Data `ccddh` 00: 1st Data `ggiih` 2nd Data `eeffh` 3rd Data `ccddh` 4th Data `aabbh`

Note 1: Byte Swap Bit[4] and Endian Control Bit[3:2] can control Endianness of ATA100 Interface.

CPU Endianness	Recommended Value for Bit[4:2]	Endianness of ATA100
Little Endian	Bit[4] = 0, Bit[3] = 0, Bit[2] = 0	Little Endian
	Bit[4] = 1, Bit[3] = 0, Bit[2] = 0	Big Endian
Big Endian	Bit[4] = 1, Bit[3] = 0, Bit[2] = 0	Little Endian
	Bit[4] = 0, Bit[3] = 1, Bit[2] = 0	Big Endian

17.3.3.5. Lower Burst Count Register (C30h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Burst Count[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Burst Count[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-10 Lower Burst Count Register

This register sets the lower order of the Burst transfer word count for when this controller becomes the Bus Master and transfers data. This register sets the maximum word count of data that can be transferred in a single Burst transfer. The value must be set more than 4.

17.3.3.6. Upper Burst Count Register (C38h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Burst Count[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Burst Count[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-11 Upper Burst Count Register

This register sets the upper order of the Burst transfer word count for when this controller becomes the Bus Master and transfers data.

17.3.3.7. Data Register ATA Shadow (000h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
DATA Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DATA Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-12 Data Register ATA Shadow

You can use this register when reading/writing to Data Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 0h.

17.3.3.8. Error/Feature Register ATA Shadow (001h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Error/Feature Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-13 Error/Feature Register ATA Shadow

You can use this register when reading the Error Register or writing to the device Features Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 1h

17.3.3.9. Sector Count Register ATA Shadow (002h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sector Count Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-14 Sector Count Register ATA Shadow

You can use this register when reading/writing to Sector Count Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 2h

17.3.3.10. LBA Low Register ATA Shadow (003h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBA Low Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-15 LBA Low Register ATA Shadow

You can use this register when reading/writing to LBA Low Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 3h.

17.3.3.11. LBA Mid Register ATA Shadow (004h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBA Mid Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-16 LBA Mid Register ATA Shadow

You can use this register when reading/writing to LBA Mid Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 4h.

17.3.3.12. LBA High Register ATA Shadow (005h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBA High Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-17 LBA High Register ATA Shadow

You can use this register when reading/writing to LBA High Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 5h.

17.3.3.13. Device/Head Register ATA Shadow (006h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-18 Device Register ATA Shadow

You can use this register when reading/writing to Device/Head Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 0, CS1N → 1, HA[2:0] → 6h.

17.3.3.14. Command/Status Register ATA Shadow (007h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command/Status Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-19 Command/Status Register ATA Shadow

You can use this register when reading Status Register or writing to the device Command Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses:
CS0N → 0, CS1N → 1, HA[2:0] → 7h.

17.3.3.15. Alternate Status Register ATA Shadow (402h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Alternate Status/Device Control Register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-20 Alternate Status Register ATA Shadow

You can use this register when reading Alternate Status Register or writing to Device Control Register of the device. Access to the device ATA Register begins when TX4939 accesses this register.

The following signals correspond to the following register addresses: CS0N → 1, CS1N → 0, HA[2:0] → 6h.

17.3.3.16. PIO Access Address Register (C88h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used							
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Not Used			CS1N	CS0N	HA[2:0]		
-	-	-	R/W	R/W	R/W	R/W	R/W
0	0	0	1	1	0	0	0

Figure 17-21 PIO Access Address Register

Setting this register makes it possible to forcibly set values such as the address that is provided to the device. This setting becomes valid when not accessing the device ATA registers nor performing data transfer. During data transfer, the values in the following table are automatically used.

Table 17-3 PIO Access Address Register

BIT	NAME	Description
Bit[4]:	CS1N	This bit determines bit 1 of Chip Select.
Bit[3]:	CS0N	This bit determined bit 0 of Chip Select.
Bit[2:0]:	HA[2:0]	This field determines the ATA address Bus.

Transfer Mode	CS0N	CS1N	HA[2:0]
PIO	0	1	0
Multiword DMA	1	1	0
Ultra DMA	1	1	0

17.3.3.17. Host Reset Timer Register (C90h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
RESET Timer							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	1	0	1	1
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
RESET Timer							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	0	1	1	1	0	0	0

Figure 17-22 Host Reset Timer Register

When executing Hard Reset using bit [11](ATA Hard Reset) of the System Control Register, this controller negates Reset signals transmitted on the ATA Bus when the time this register sets elapses. This register sets this time value in 10 nsec increments. The default for this register is 30 μsec (the ATA Spec. is 25 μsec).

You can also use this register as a timer for ATA soft reset. This timer is also invoked when Soft Reset using bit [2] of the Device Control Register is issued. This Controller is configured to issue an interrupt simultaneous to when the set time is reached.

17.3.3.18. Interrupt Control Register (C98h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Mask Address Error INT	Mask Reach Multiple INT	Mask DEV Timing Error	Mask Ultra DMA DEV Terminate	Mask Timer INT	Mask Bus Error	Mask Data Transfer End	Mask Host INT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Address Error INT	Reach Multiple INT	DEV Timing Error	Ultra DMA DEV Terminate	Timer INT	Bus Error	Data Transfer End	Host INT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-23 Interrupt Control Register

Table 17-4 Interrupt Control Register

BIT	NAME	Description
Bit[15]:	Mask Address Error INT	Interrupt mask. You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates that this Controller is write-accessed with address where any registers do not exist. Bit [7](Address Error INT) can be set regardless of this bit.
Bit[14]:	Mask Reach Multiple INT	Interrupt mask. You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates PIO transfer is in the Break state. Bit [6](Reach Multiple INT) can be set regardless of this bit.
Bit[13]:	Mask DEV Timing Error	Interrupt mask. You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates timing of a control signal from the device is out of spec. when performing data transfer with the device. Bit [5](DEV Timing Error) can be set regardless of this bit.
Bit[12]:	Mask Ultra DMA DEV Terminate	Interrupt mask. You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates the device terminated Ultra DMA transfer. Bit [4](Ultra DMA Terminate) can be set regardless of this bit.
Bit[11]:	Mask Timer INT	Interrupt mask. You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates the Host Reset timer reached the defined value. Bit [3](Timer INT) can be set regardless of this bit.
Bit[10]:	Mask Bus Error	Interrupt mask . You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates the Bus error occurred. Bit [2](Bus Error) can be set regardless of this bit.
Bit[9]:	Mask Data Transfer End	Interrupt mask . You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates data transfer ended. Bit [1](Data transfer End) can be set regardless of this bit.
Bit[8]:	Mask Host INT	Interrupt mask . You can prevent the INTN signal asserting by setting this bit “1”. The interrupt indicates the INTRQ signal from the device is active. Bit [0](Host INT) can be set regardless of this bit.
Bit[7]:	Address Error INT	Interrupt status that indicates this Controller is write-accessed with address where any registers do not exist. Setting “1” to this bit clears this interrupt status.
Bit[6]:	Reach Multiple INT	Interrupt status that indicates PIO transfer is in the Break state. Setting “1” to this bit clears this interrupt status.

Table 17-4 Interrupt Control Register

BIT	NAME	Description
Bit[5]:	DEV Timing Error	Interrupt status that indicates timing of a control signal from the device is out of spec. when performing data transfer with the device. Setting “1” to this bit clears this interrupt status. The specifications for confirming timing are as follows: 1. PIO Mode tB (IORDY Pulse Width) 2. Ultra DMA Mode tFS (First Strobe Time) tLI (Limited Interlock Time) tRFS (Ready to Final Strobe Time) tRP (Minimum Time to assert STOP or negate DMARQ) tSS (Time from STROBE edge to negation of DMARQ or assertion of STOP) The Table 17-5 shows the threshold values for detecting timing errors. However, error detection is not performed in the case of tRFS since the time period from when HDMARDY is negated to when STOP is asserted is shorter than the error detection threshold when in Host terminating an Ultra DMA data-in burst Mode0-3.
Bit[4]:	Ultra DMA DEV Terminate	Interrupt status that indicates the device terminated Ultra DMA transfer. Setting “1” to this bit clears this interrupt status.
Bit[3]:	Timer INT	Interrupt status that indicates the Host Reset timer reached the defined time. Setting “1” to this bit clears this interrupt status.
Bit[2]:	Bus Error	Interrupt status that indicates the Bus error occurred when this controller becomes the Bus Master and executes data transfer. Setting “1” to this bit clears this interrupt status. Before clearing this bit, set “1” to bit [14](FIFO Reset) of System Control Register and reset FIFO. The Start bit (bit[0]) in DMA Command register is not cleared when the bus error occurred, so you may set “1” to bit [15] (Soft Reset) of System Control1 register to reset this Controller.
Bit[1]:	Data Transfer End	Interrupt status that indicates data transfer ended. This bit is also set when the transfer of command packets ends. Setting “1” to this bit clears this interrupt status.
Bit[0]:	Host INT	This bit indicates that INTRQ, which is the interrupt signal from the device, is active. Setting “1” to this bit clears this interrupt status.

Table 17-5 Timing Error Threshold

Detection Item							
t_B (PIO Mode)	Spec. Value	1250(Max)					
	Error Detection Value	>1270					
t_{FS} (UDMA Mode)	Mode	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
	Spec. Value (MAX)	230	200	170	130	120	90
	Error Detection Value	270	240	210	170	160	130
t_{LI} (UDMA Mode)	Mode	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
	Spec. Value (MAX)	150	150	150	100	100	75
	Error Detection Value	180	180	180	130	130	110
t_{RFS} (UDMA Mode)	Mode	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
	Spec. Value (MAX)	75	70	60	60	60	50
	Error Detection Value	260	180	140	160	110	80
t_{RP} (UDMA Mode)	Mode	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
	Spec. Value (Min)	160	125	100	100	100	85
	Error Detection Value	140	100	80	80	80	60
t_{SS}(UDMA Mode)	Mode	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
	Spec. Value (Min)	50	50	50	50	50	50
	Error Detection Value	30	30	30	30	30	30

(in nsec)

Note: When the INT_IDE of DMA Status Register is set to “1”, the following bits of Interrupt Control Register will be reset: bits [15:8] (Mask Address Error INT, Mask Reach Multiple INT, Mask DEV Timing Error, Mask Ultra DMA DEV Terminate, Mask Timer INT, Mask Bus Error, Mask Data Transfer End, Mask Host INT), and bits [1:0] (Data Transfer End, Host INT).

17.3.3.19. ATAPI Packet Command Register (CB8h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Packet[15:8]							
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Packet[7:0]							
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0

Figure 17-24 ATAPI Packet Command Register

Table 17-6 ATAPI Packet Command Register

BIT	NAME	Description
Bit[15:0]:	Packet	Use this register to issue the ATAPI command packet to the device. This Register consists of a 16-stage FIFO. You can transfer a command packet to the device by setting the prescribed command packet in FIFO, setting the transfer count in bits [11:8](Packet Transfer Count) of the Packet Transfer Control Register, then setting bit [0](Packet Start) of the Packet Transfer Control Register to “1”.

17.3.3.20. Bus Transfer Count High Register (CC0h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used					Bus Master Transfer Count[26:24]		
-	-	-	-	-	R	R	R
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Bus Master Transfer Count[23:16]							
R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0

Figure 17-25 Bus Transfer Count High Register

You can know the transfer count when transferring data on the G-Bus when this ATA controller is the Bus Master. The total transfer count is set in this controller. When transfer starts, the total number of transfers to be performed is set in this register and this number is decremented when each transfer is performed.

17.3.3.21. Bus Transfer Count Low Register (CC8h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Bus Master Transfer Count[15:8]							
R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Bus Master Transfer Count[7:0]							
R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0

Figure 17-26 Bus Transfer Count Low Register

This is the lower register that is used with the Bus Transfer Count High Register.

17.3.3.22. ATA Device Timing Error Register (CD0h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used							
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Not Used		PIO_tB	DMA_tFS	DMA_tLI	DMA_tRFS	DMA_tRP	DMA_tSS
-	-	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figure 17-27 ATA Device Timing Error Register

You can use this register to confirm the status when performing ATA data transfer and timing of a control signal that the device provides to this controoler is out of spec. Writing “1” to each bit clears the status.

Table 17-7 ATA Device Timing Error Register

BIT	NAME	Description
Bit[5]:	PIO_t_B	Timing that is out of the tB spec occurred during PIO mode transfer.
Bit[4]:	DMA_t_{FS}	Timing that is out of the tFS spec occurred during Ultra DMA mode transfer.
Bit[3]:	DMA_t_{LI}	Timing that is out of the tLI spec occurred during Ultra DMA mode transfer.
Bit[2]:	DMA_t_{RFS}	Timing that is out of the tRFS spec occurred during Ultra DMA mode transfer. However, error detection is not implemented when in Host terminate an Ultra DMA data-in burst Mode0-3.
Bit[1]:	DMA_t_{RP}	Timing that is out of the tRP spec occurred during Ultra DMA mode transfer.
Bit[0]:	DMA_t_{SS}	Timing that is out of the tSS spec occurred during Ultra DMA mode transfer.

17.3.3.23. Packet Transfer Control Register (CD8h)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used			Packet Transfer Mode	Packet Transfer Count			
-	-	-	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Not Used							Packet Start
-	-	-	-	-	-	-	R/W
0	0	0	0	0	0	0	0

Figure 17-28 Packet Transfer Control Register

This register makes Packet transfer related settings.

Table 17-8 Packet Transfer Control Register

BIT	NAME	Description
Bit[12]:	Packet Transfer Mode	You can use this bit to set the transfer mode of the command packet. 1: Data transfer timing (Execute transfer using the transfer mode set by bits [7:4] (Data Transfer Mode Select) of the System Control Register.) 0: PIO data transfer timing (Packet data is transferred in PIO mode when the data transfer mode is set to Multiword DMA or Ultra DMA. The same transfer mode selected by bits [10:8] (Command Transfer Mode Select) of the System Control Register is used.)
Bit[11:8]:	Packet Transfer Count	You can use this bit to set the command packet transfer count. Set the value that is equal to the transfer count – 1.
Bit[0]:	Packet Start	You can use this bit to initiate transfer of the command packet.

17.3.3.24. Transfer Start Top Address Register (CE0h)

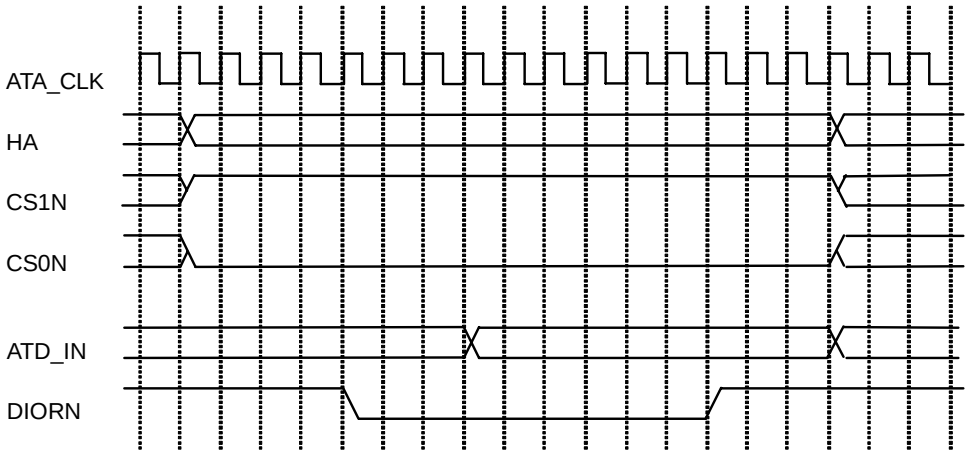
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Not Used							
-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Not Used				Transfer Start Address [35:32]			
-	-	-	-	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

This register sets the read/write start top start address of the transfer that this controller outputs to the G-Bus.

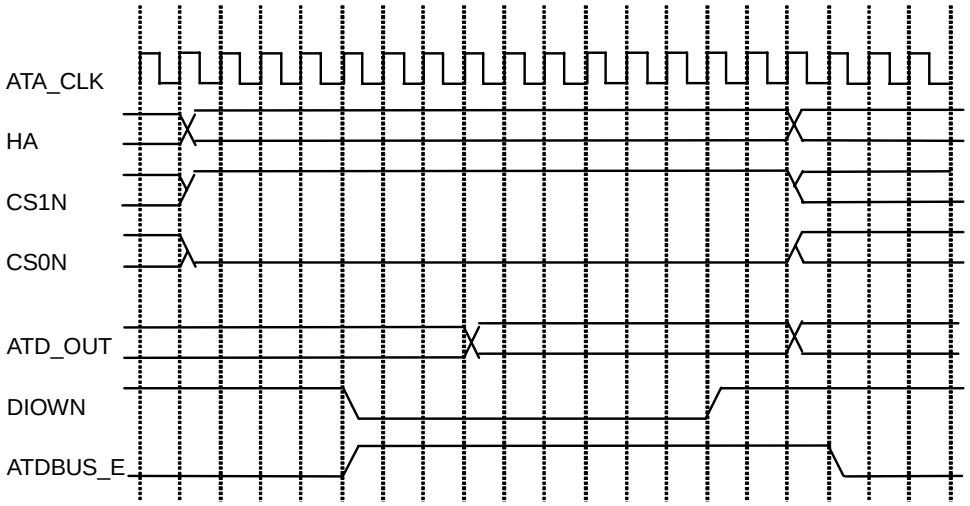
17.4. Actual Operation Timing

17.4.1. Data transfer in the PIO mode

Data transfer in the PIO Mode (Read)



Data Transfer in the PIO Mode (Write)



The above figures are timing diagrams of read/write transfer by PIO Mode 4. Following are the steps for setting up the software used when transferring data.

(1) Use bits [7:4] of the System Control Register to select the PIO mode.

(2) Use the Transfer Word Count Register to set the word count of data to be transferred with a device.

(3) Use the Transfer Sector Count Register to set the sector count of data to be transferred with a device.

(4) This controller becomes the Bus Master then the 36-bit start address at which data transfer starts is set in the Transfer Start Lower Register, Transfer Start Upper Register, and the Transfer Start Top Register. Then, the maximum number of words that this controller transfers as the Bus Master using a single Burst transfer is set in the Burst Count Upper Register and Burst Count Lower Register..

(5) Use bit [3] (R_ORWCTR) of the DMA Command Register to determine the data transfer direction.

(6) Execute the device recognition sequence. (See below for details.)

(7) Use the ATA Shadow Register to set commands or parameters such as the LBA value.

(8)-1 For read commands: Wait for an interrupt from the device, read the Status Register of the ATA Shadow Register, then confirm that the DRQ bit is set. (If the DRQ bit has not been set yet, wait until it is set.)

(8)-2 For write commands: Read the Status Register of the ATA Shadow Register immediately after a command is issued. Confirm that the DRQ bit is set. (If the DRQ bit has not been set yet, wait until it is set.)

(9) Set bit [0] (START_STOPBM) of the DMA Command Register, then start data transfer.

(10) Wait for an interrupt from the device or transfer end interrupt.

(11)-1 For read commands: Acknowledge that transfer has ended if a transfer end interrupt is issued. Then, shift to the next process. If there is an interrupt from the device, then read the Status Register of the ATA Shadow Register, acknowledge that the DRQ bit is set (in this case, this controller is also in the interrupt status that is indicated at bit [6](Reach Multiple INT) of the Interrupt Control Register), set bit [3] and bit [2] of the System Control Register, start transfer of the next sector and return to process (10).

(11)-2 For write commands: Acknowledge an interrupt from the device. If a transfer end interrupt is inserted simultaneous to an interrupt from the device, then read the Status Register of the ATA Shadow Register, confirm that the transfer has ended, then transition to the next process. If no transfer end interrupt was issued, confirm that the DRQ bit is set (in this case, this controller is also in the interrupt status that is indicated at bit [6](Reach Multiple INT) of the Interrupt Control Register), set bit [3] and bit [2] of the System Control Register, start transfer of the next sector, then return to process (10).

The device recognition sequence is a process that detects any devices connected to the ATA Bus and fetches information on those devices.

(1) Read the Status Register of the ATA Shadow Register, and confirm that the BSY bit is “0”.
(2) Read the Status Register of the ATA Shadow Register, and confirm that the DRQ bit is “0”. An error has occurred if the DRQ bit is “1”.
(3) Set the DEV bit of the Device/Head Register of the ATA Shadow Register.
(4) Confirm that the BSY bit of the Status Register of the ATA Shadow Register is “0”.
(5) Confirm that the DRQ bit of the Status Register of the ATA Shadow Register is “0”. An error has occurred if the DRQ bit is “1”.
(6) Read the LBA Mid Register and LBA High Register of the ATA Shadow Register, and confirm Signature.
(7) Execute the IDENTIFY DEVICE command and fetch the device information.

Note 1) The specification requires the device recognition sequence to be performed each time you set up the software used for transferring data.
Note 2) The specification specifies that the BSY and DRQ bits confirmed from the Status Register of the ATA Shadow Register are confirmed in the following order: clear BSY → set DRQ.

The following table shows the PIO transfer timing in each mode.

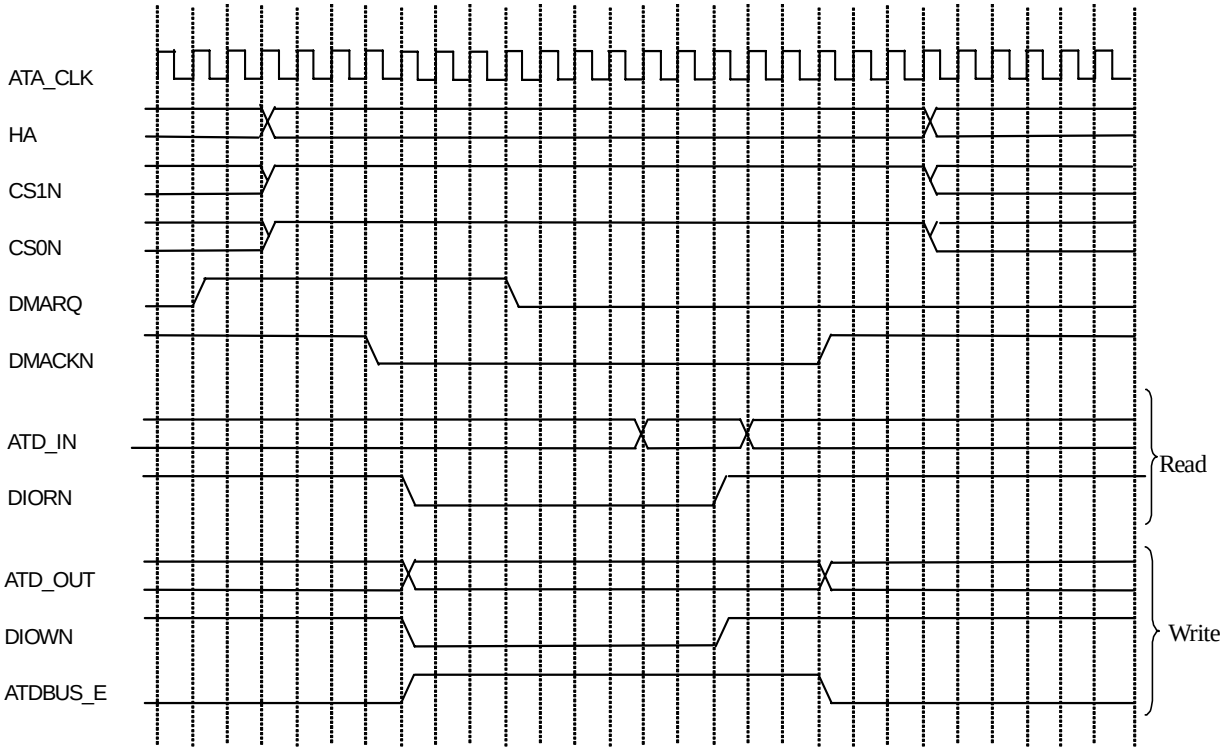
Table 17-9 PIO Transfer Timing

Mode	HA, CS Valid - DIORN/DIOWN Enable	DIORN/DIOWN Width	DIORN/DIOWN Disable - HA, CS Invalid	Cycle Time
Mode0	130 nsec	180 nsec	380 nsec	600 nsec
Mode1	100 nsec	150 nsec	210 nsec	390 nsec
Mode2	50 nsec	120 nsec	110 nsec	240 nsec
Mode3	50 nsec	100 nsec	70 nsec	180 nsec
Mode4	40 nsec	90 nsec	20 nsec	120 nsec

Note 1) HA, CS0N, CS1N: Are automatically set as follows at the point when PIO transfer starts: HA=0h, CS0N=0, CS1N=1.

Note 2) ATD_IN[15:0]:
During read → Send to the FIFO in the controller at the DIORN rise.
ATD_OUT[15:0]:
During write→ Define data simultaneous to the DIOWN fall. Hold the value for 2 ATA_CLK (100MHz) cycles after the DIOWN rise.

17.4.2. Multiword DMA mode data transfer



The above figure is a timing diagram that shows read/write transfer by Multiword DMA Mode 2. Following are the steps for setting up the software used when transferring data.

(1) Use bits [7:4] of the System Control Register. Select the Multiword DMA mode and set bit [1] of the System Control Register.

(2) – (7) (The same as for PIO transfer.)

(8) Wait for a transfer end interrupt and an interrupt from the device. Confirm that the transfer process has ended, then transition to the next process.

17

Use bits [7:4] of the System Control Register. Select the Multiword DMA mode and set bit [1] of the System Control Register. (The same as for PIO transfer.)
Wait for a transfer end interrupt or an interrupt from the device. Confirm that the transfer process has ended, then transition to the next process.

Following is a description of the meaning of each control signal.

DMARQ: DMA request signal that is input from the device. This signal starts the data transfer process when it is input to the Controller.
DMACKN: Receives the DMARQ signal, then starts the actual transfer process when this controller completes preparations for data transfer.
DIORN: This signal is used as a strobe signal when transferring data from the device to the controller.
DIOWN: This signal is used as a strobe signal when transferring data from the controller to the device.

The following table shows the transfer timing when using Multiword DMA.

Table 17-10 Multiword DMA Transfer Timing

Mode	HA, CS Valid - DIORN/DIOWN Enable	DIORN/DIOWN Width	DIORN/DIOWN Disable - HA, CS Invalid	Cycle Time
Mode0	90 nsec	260 nsec	210 nsec	480 nsec
Mode1	60 nsec	100 nsec	80 nsec	160 nsec
Mode2	40 nsec	90 nsec	70 nsec	120 nsec

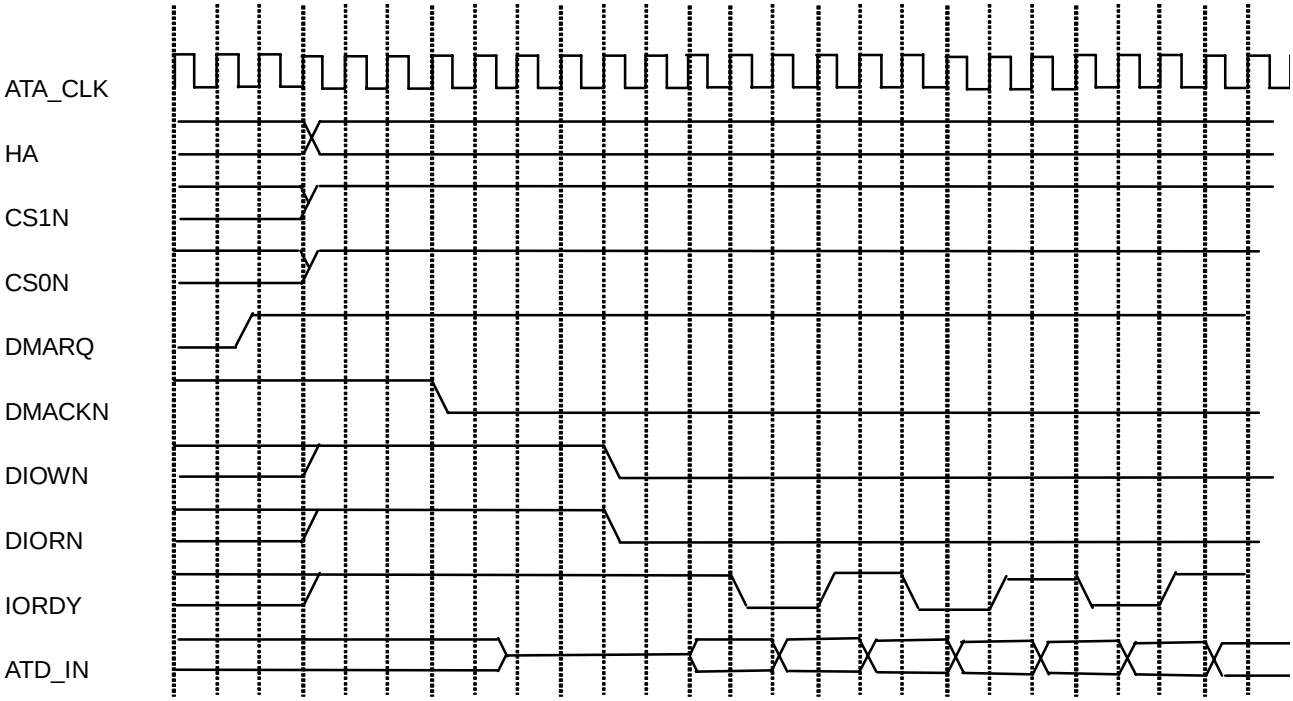
Note 1) HA, CS0N, CS1N: Are automatically set as follows at the point when Multiword DMA transfer starts: HA=0h, CS0N=1, CS1N=1.

Note 2) ATD_IN[15:0]:
During read → Send to the FIFO in ATA100 core at the DIORN rise.

ATD_OUT[15:0]:
During write→ Define data simultaneous to the DIOWN fall. Hold the value for 3 ATA_CLK (100MHz) cycles after the DIOWN rise.

17.5. Ultra DMA mode data transfer

17.5.1. When a read command is issued to the device (transfer start position)



The above figure is a timing diagram that shows the transfer start position during read transfer in Ultra DMA Mode 5. Following are the steps for setting up the software used when transferring data.

(1) Use bits [7:4] of the System Control Register, select the Ultra DMA mode, then set bit [1] of the System Control Register.

(2) – (7) (The same as for PIO transfer.)

(8) Wait for a transfer end interrupt and an interrupt from the device. Then, confirm that the transfer process ended and transition to the next process.

Following is a description of the meaning of each control signal.

DMARQ: DMA request signal that is inputted from the device. This signal starts the data transfer process when it is inputted to the Controller.

DMACKN: Receives the DMARQ signal, then starts the actual transfer process when this controller completes its data transfer preparations.

DIORN: This signal is usually used as a strobe signal when transferring data from the device to the controller. During read transfer in Ultra DMA however, this signal is used as the DMA Ready signal and outputs “High” to pause the data transfer.

DIOWN: This signal is usually used as a strobe signal when transferring data from the controller to the device. During read transfer in Ultra DMA however, this signal is used as the DMA Stop signal and outputs “High” to terminate the data transfer.

IORDY: This signal is input to this controller as a strobe signal used when transferring data from the device to the controller. Sampling data at both edges of this signal makes it possible to send data to the controller.

Following is the flow of each control signal at the transfer start position:

A Wait for DMARQ to become active (“High”).

B Set HA to “0h” and CS0N and CS1N to “1” when transfer preparations inside the controller are complete.

C Wait for the specified time, then make the DMACKN active (set to “0”).

D Wait for the specified time, then set DIORN and DIOWN to “0”.

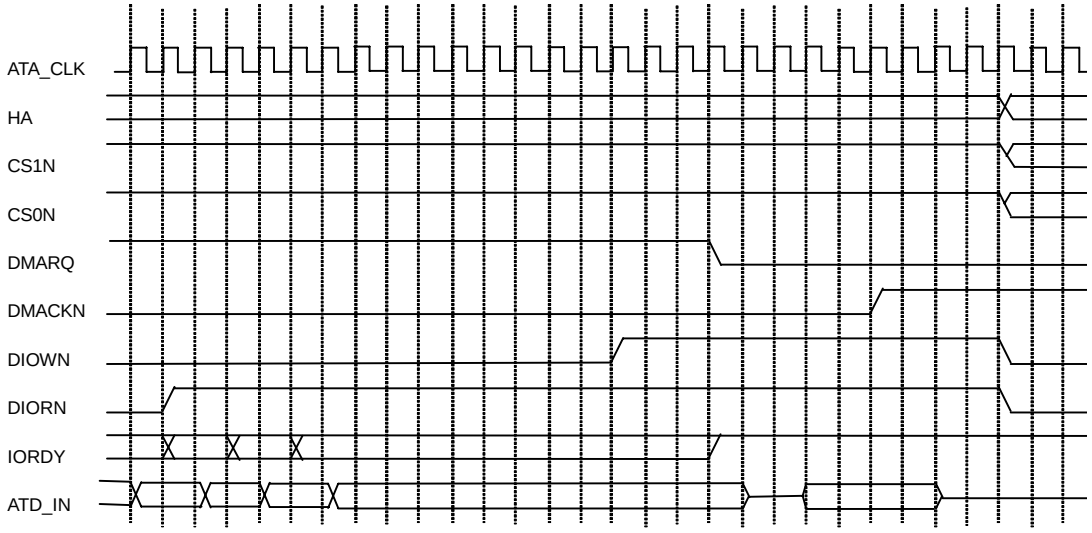
E Wait for IORDY to change, then send data to the controller.

The following table shows the timing parameters when transfer starts.

Table 17-11 Timing Parameters when Ultra DMA Transfer Starts

Mode	HA, CS Valid - DMACKN “0”	DMACKN “0” - DIORN/DIOWN “0”
Mode0	30 nsec	40 nsec
Mode1	30 nsec	40 nsec
Mode2	30 nsec	40 nsec
Mode3	30 nsec	40 nsec
Mode4	30 nsec	40 nsec
Mode5	30 nsec	40 nsec

17.5.2. When read command is issued to the device (Transfer End Position)



The above figure is a timing diagram that shows the transfer end position during read transfer by Ultra DMA Mode 5. Following is the flow of each control signal at the transfer end position.

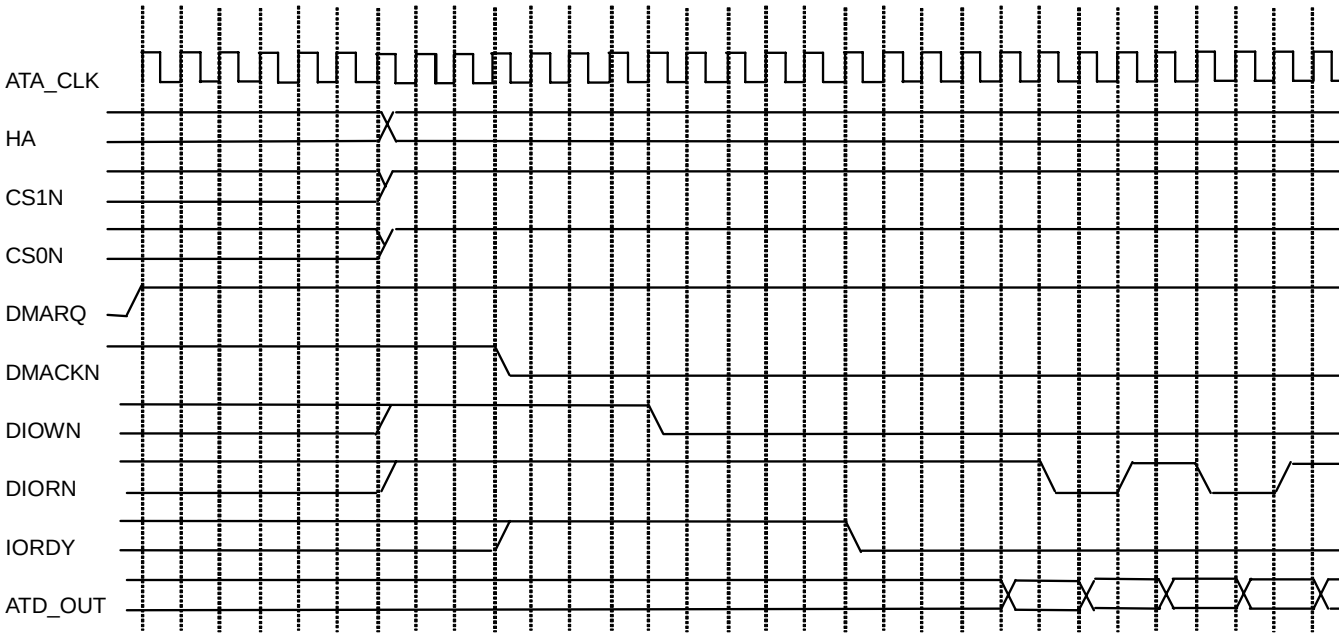
A- Set DIORN to “1” when all data for the set Sector Count × 512 Bytes is sent to the controller.
B- Wait for the specified time, then set DIOWN to “1”.
C- Wait until DMARQ is cleared.
D- CRC data generated inside the Controller is transmitted to the ATA Data Bus.
E- Set DMACKN to “1”. The device will then latch the CRC data that is transmitted from this controller.
F- Release CS0N, CS1N and HA.

The following table shows the parameters when transfer ends.

Table 17-12 Parameters when Ultra DMA Transfer Ends

Mode	DIORN “1” - DIOWN “1”	DMARQ “0”- DMACKN “1”	DMACKN “1” - HA,CS Release	IORDYN cycle (Input Signal)	CRC Data Setup (Min.)	CRC Data Hold
Mode0	180 nsec	130-140 nsec	40 nsec	240 nsec (Min)	90 nsec	30 nsec
Mode1	150 nsec	120-130 nsec	40 nsec	160 nsec (Min)	80 nsec	30 nsec
Mode2	140 nsec	100-110 nsec	40 nsec	120 nsec (Min)	60 nsec	30 nsec
Mode3	140 nsec	90-100 nsec	40 nsec	90 nsec (Min)	50 nsec	30 nsec
Mode4	140 nsec	70-80 nsec	40 nsec	60 nsec (Min)	30 nsec	30 nsec
Mode5	140 nsec	60-70 nsec	40 nsec	40 nsec (Min)	20 nsec	30 nsec

17.5.3. When Write Command is issued to the device (Transfer Start Position)



The above figure is timing diagram that shows the transfer start position during write transfer by Ultra DMA Mode 5. The steps for setting up the software used when transferring data are exactly the same as for when issuing a read command. The meaning of each signal on the ATA Bus changes: IORDY becomes the transfer Ready signal that the device provides to this controller; DIORN becomes the Strobe signal when transferring data to the device.

Following is the flow of each control signal at the transfer start position.

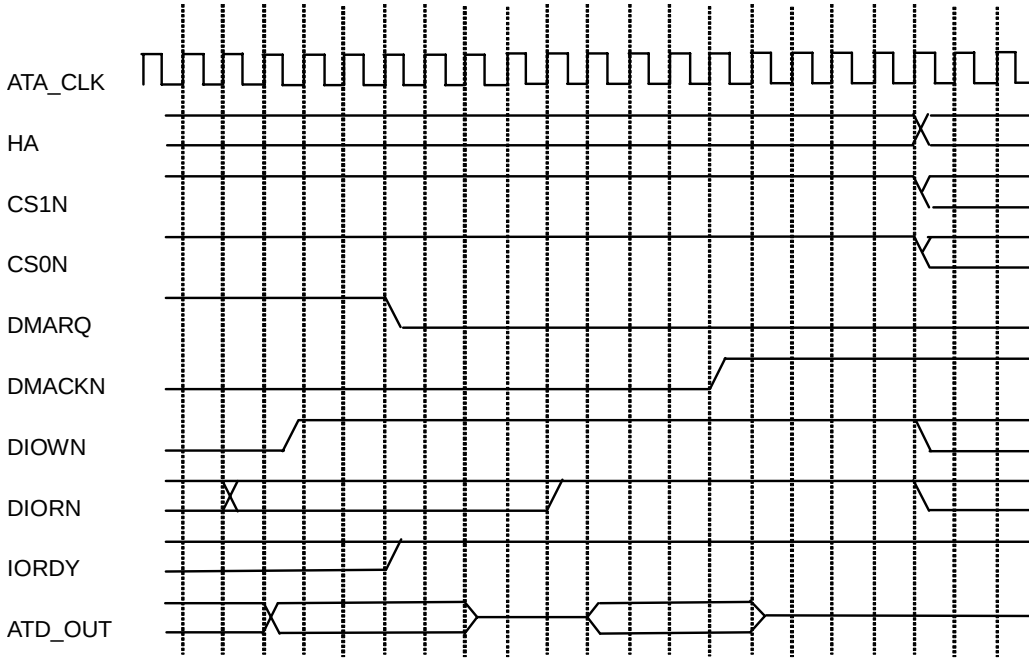
A- Wait until DMARQ becomes active (“High”).
B- When transfer preparations inside the Controller are complete, set HA to “0h”, and CS0N and CS1N to “1”.
C- Wait for the specified time, then make the DMACKN active (“0”).
D- Wait until IORDY becomes active (“Low”).
E- Wait for the specified time, then transfer data to the device while changing the state of DIORN.

The following table shows the parameters when transfer starts.

Table 17-13 Parameters when Ultra DMA Transfer Starts

Mode	HA, CS Valid - DMACKN “0”	DMACKN “0” - DIOWN “0”	IORDY “0” - DIORN “0”	DIORN Pulse Width	ATDBUS Setup	ATDBUS Hold
Mode0	40 nsec	40 nsec	140-150 nsec	120 nsec	110 nsec	10 nsec
Mode1	40 nsec	40 nsec	100-110 nsec	80 nsec	70 nsec	10 nsec
Mode2	40 nsec	40 nsec	80-90 nsec	60 nsec	50 nsec	10 nsec
Mode3	40 nsec	40 nsec	60-70 nsec	“H”: 40 “L”: 50	“H”: 30 “L”: 40	10 nsec
Mode4	40 nsec	40 nsec	50-60 nsec	30 nsec	20 nsec	10 nsec
Mode5	40 nsec	40 nsec	40-50 nsec	20 nsec	10 nsec	10 nsec

17.5.4. When Write Command is issued to the device (Transfer End Position)



The above figure is a timing diagram that shows the transfer end position during write transfer by Ultra DMA Mode 5. Following is the flow of each control signal.

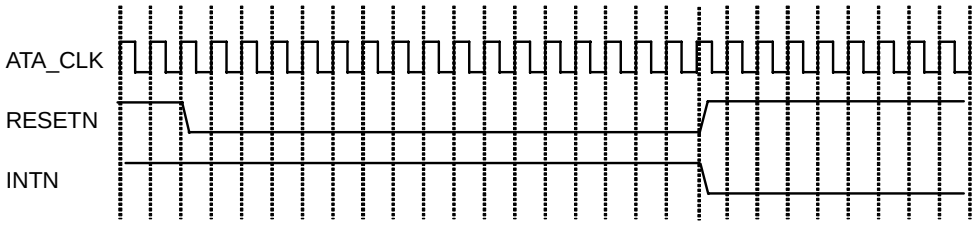
A- Set DIOWN to "1" when all data for the set Sector Count × 512 Bytes is sent from the controller.
B- Wait until DMARQ is cleared and IORDY is cleared.
C- Send the CRC data generated in the controller to the ATA data bus.
D- Wait for the specified time, then set DMACKN to "1". The device now latches the CRC data sent from this controller.
E- Release CS0N, CS1N and HA.

The following table shows the parameters when transfer ends.

Table 17-14 Parameters when Ultra DMA Transfer Ends

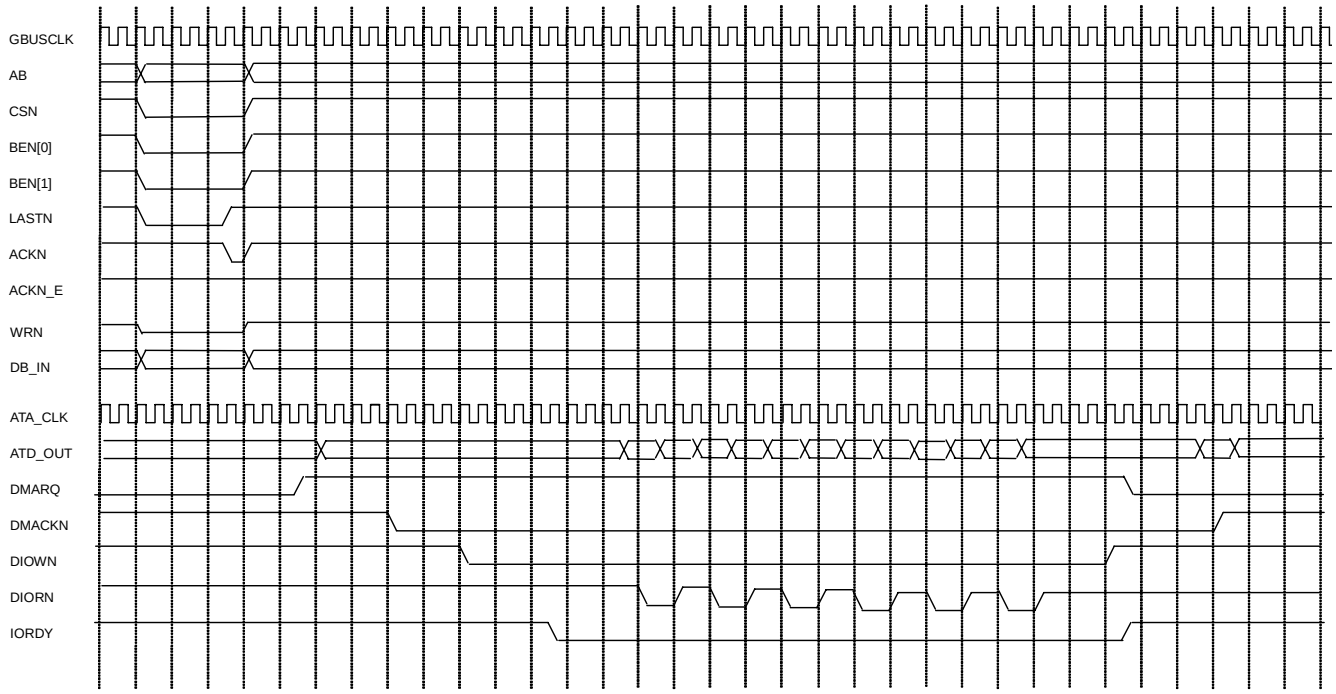
Mode	DMARQ "0"-DMACKN "1"	DMACKN "1"-HA, CS Release	CRC Data Setup (Min)	CRC Data Hold
Mode0	140-150 nsec	30 nsec	100 nsec	20 nsec
Mode1	130-140 nsec	30 nsec	90 nsec	20 nsec
Mode2	110-120 nsec	30 nsec	70 nsec	20 nsec
Mode3	100-110 nsec	30 nsec	60 nsec	20 nsec
Mode4	80-90 nsec	30 nsec	40 nsec	20 nsec
Mode5	70-80 nsec	30 nsec	30 nsec	20 nsec

17.5.5. Issuing Timing of Reset Signal



The above figure is a timing chart that shows the RESETN signal that is input to the device by setting bit [11] of the System Control Register to “1”. As you can see from the above timing chart, the system is configured so you can send an interrupt signal to the microcontroller simultaneous to automatically clearing the above bit when the time set by the Host Reset Timer Register elapses.

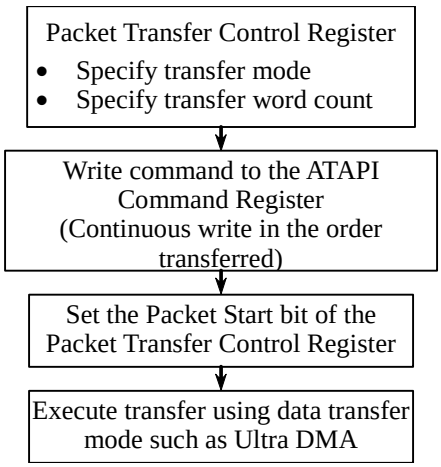
17.5.6. Command Packet Transmission Timing



The above figure is a timing chart that shows the situation when using the ATAPI Packet Command Register described in Section 17.3.3.19 to issue a command packet to the device. Ultra DMA is used as the transfer protocol. Also, Ultra DMA uses the mode selected by bits [7:4] of the System Control Register described in Section 17.3.2.4.

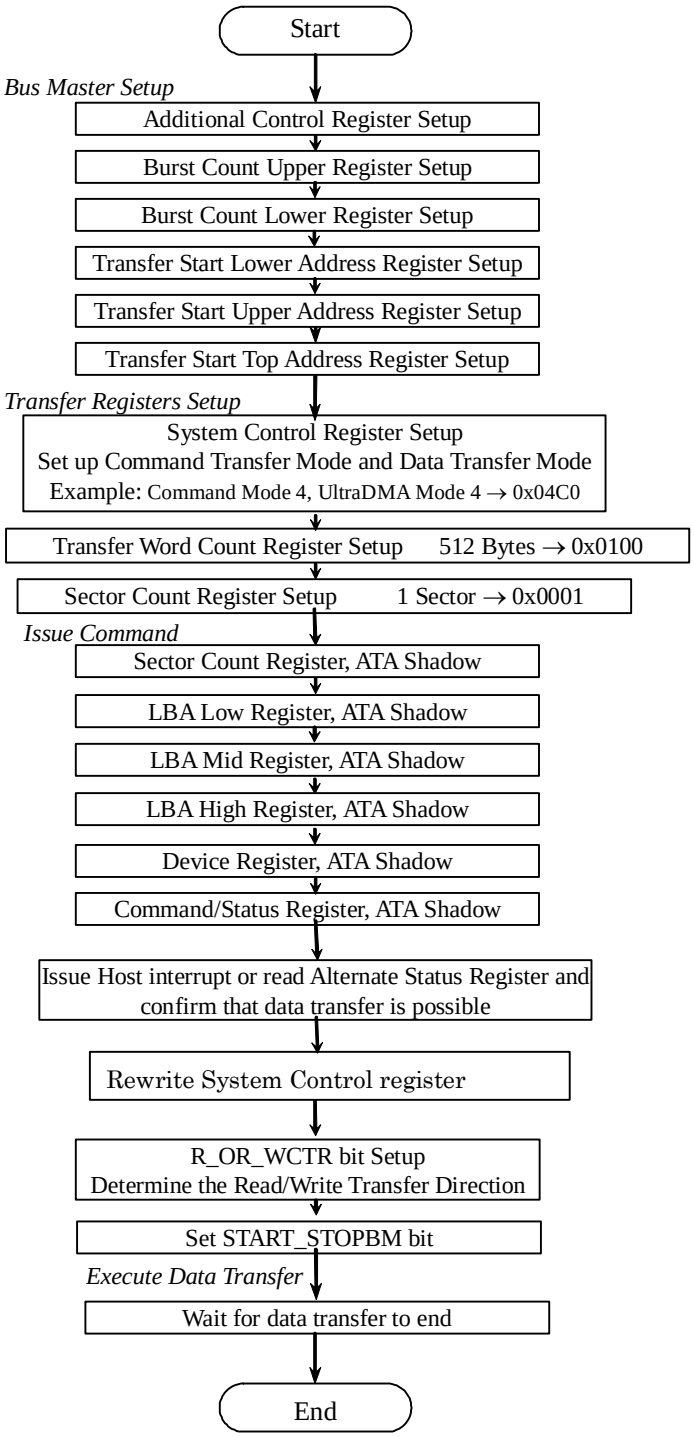
Command packets usually consist of 12 bytes, and these 12 bytes accumulate in the FIFO in the Controller when 6 consecutive accesses to the ATAPI Packet Command Register are performed. The command packet is sent to the device after accumulated in the FIFO in the controller.

In addition to the timing shown here, you can transmit data using the same timing as register transfer using the setting of bit [12](Packet Transfer Mode) of the Packet Transfer Control Register. Specifically, you can perform this transfer by using the following setup routine to execute the transfer process.



17.6. Basic Operation Setup

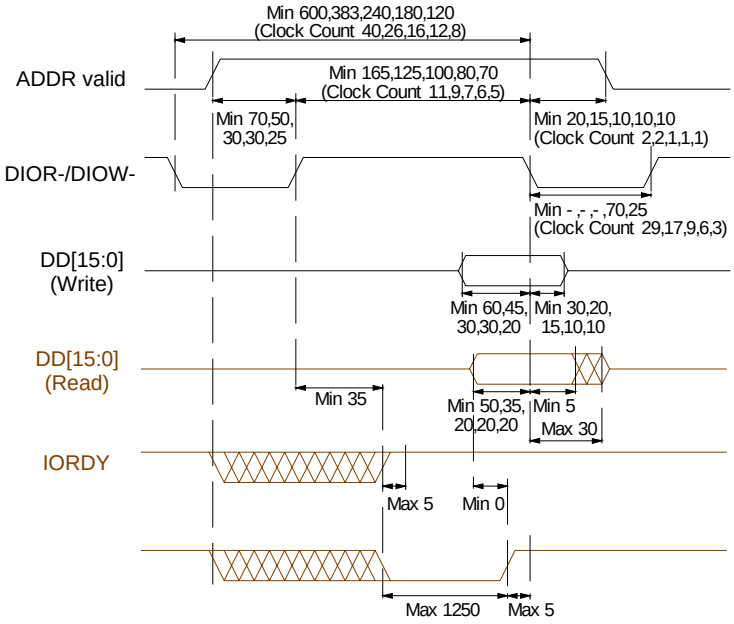
This setup is for PRD not enable. In PRD, a PRD table must be setup in place of Transfer Start Address.



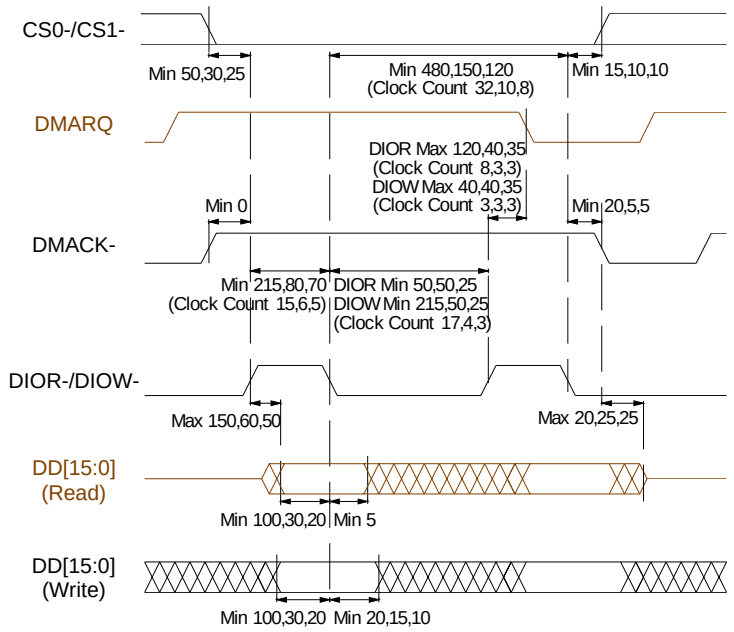
Note: R_OR_WCTR and START_STOPBM of the DMA Command Start Register settings are unnecessary when a WRITE DMA, READ DMA, WRITE DMA EXT or READ DMA EXT command is issued to the device and bit [2] (Auto DMA Enable) of the System Control Register is set.

17.7. ATA Bus Access Timing (ATA/ATAPI-6 Spec. Values)

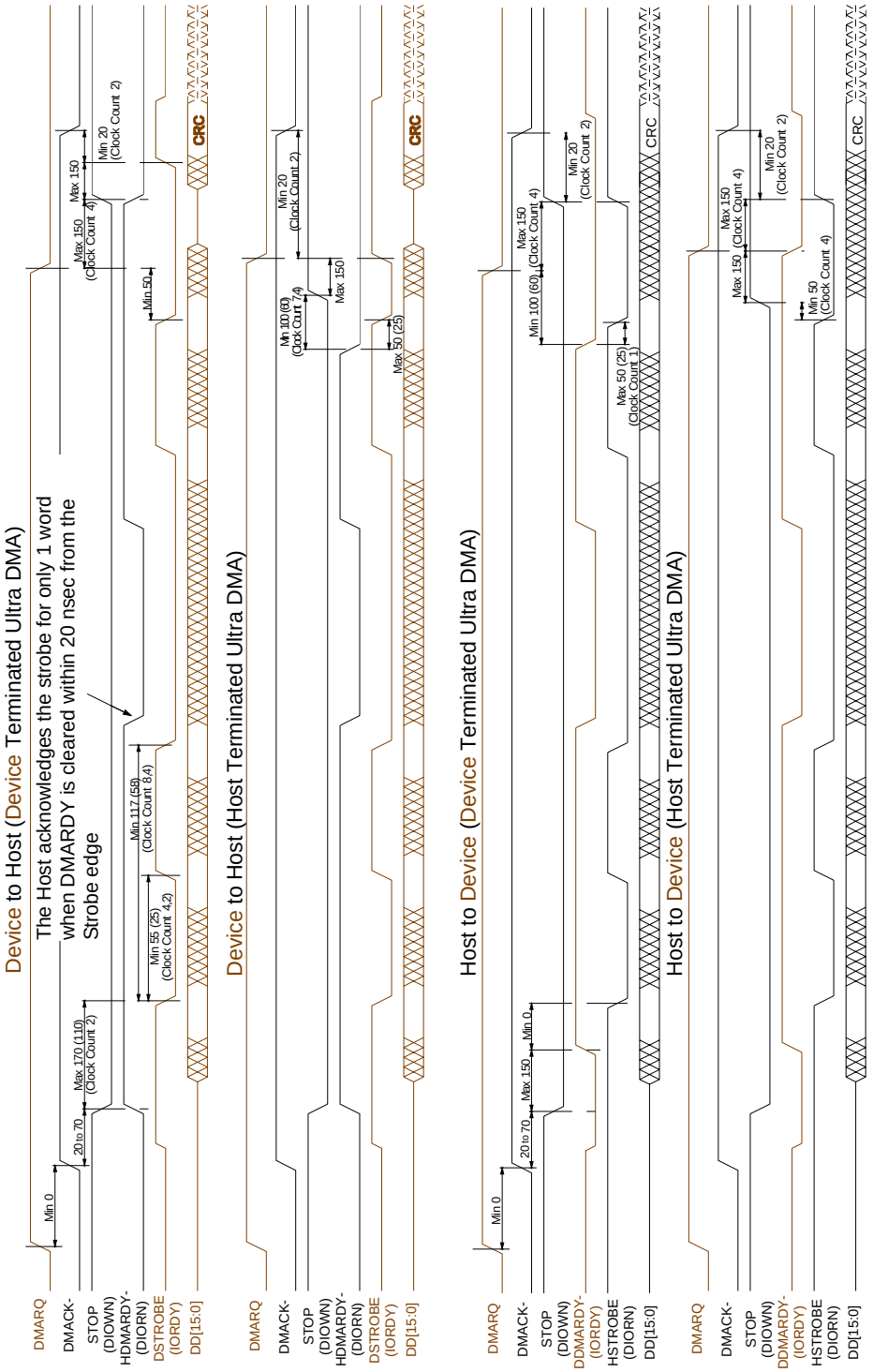
17.7.1. PIO Mode



17.7.2. Multiword DMA Mode



Ultra DMA Mode



Chapter 18. Dual Ethernet MAC Controller

18.1. Features

TX4939 Ethernet MAC has the following characteristics,

- *Selectable 1 or 2 Ethernet channels.*
- *Each Ethernet channel resides as a unique PCI device on an internal 32 bit 66 MHz PCI bus.*
- *Supports transfer rates of 100 Mbps and 10 Mbps per Ethernet channel.*
- *Equivalent to the TC35815CF except Reduced Media Independent Interface (RMII).*
- *Employs the IEEE Std 802.3 standard CSMA/CD format when in the half duplex transfer mode.*
- *Employs IEEE Std 802.3x standard MAC control and Pause operation for flow control when in the full duplex transfer mode.*
- *Has RMII(Reduced Media Independent Interface)*

18.2. Block diagram

Each of the two Ethernet controllers in TX4939 resides behind an internal 32 bit 66Mhz PCI bus. The system level block diagram of the two Ethernet controllers is given in Figure 18-1.

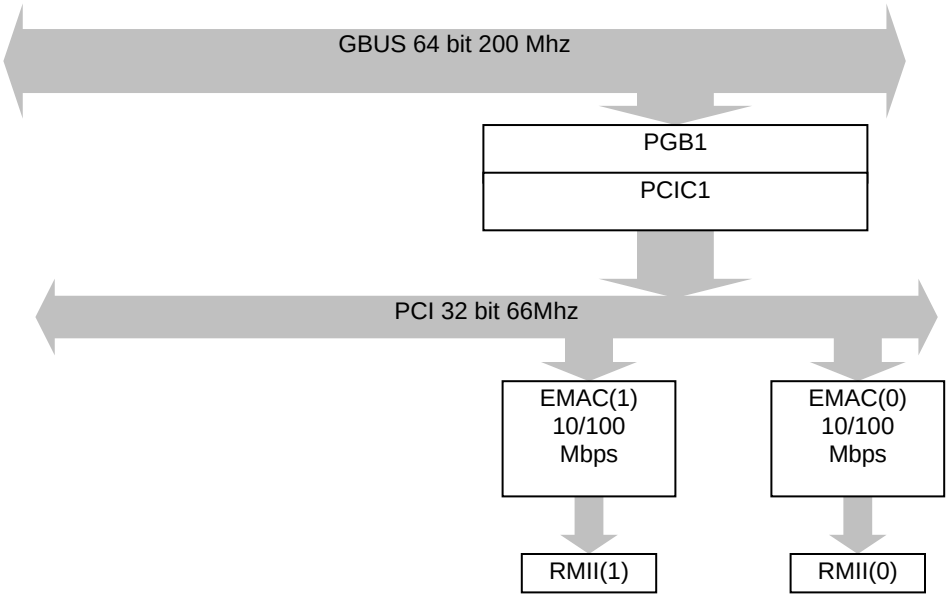


Figure 18-1 System Level block diagram of Quad eMAC in TX4939

Depending on the ATA1MODE, ET0MODE and ET1MODE configuration bits (Chapter 7), 1, or 2 Ethernet channels are usable. Table 18-1 gives the list of usable eMAC channels dependent on ATA1MODE, ET0MODE, and ET1MODE.

Table 18-1 Usable eMAC channels based on Product Mode selection

ATA1MODE	ET0MODE	ET1MODE	Function Result
0	0	0	All GPIO (Boot-up default)
0	1	0	Ethernet Channel 0 and GPIO
0	0	1	Ethernet Channel 1 and GPIO
0	1	1	Ethernet 0, 1 and GPIO
1	0	0	ATA100-1, no GPIO

Note: Unusable Ethernet channels are not accessible through the internal PCI Bus.

The block diagram of a single TX4939 Ethernet Controller is given in Figure 18-2.

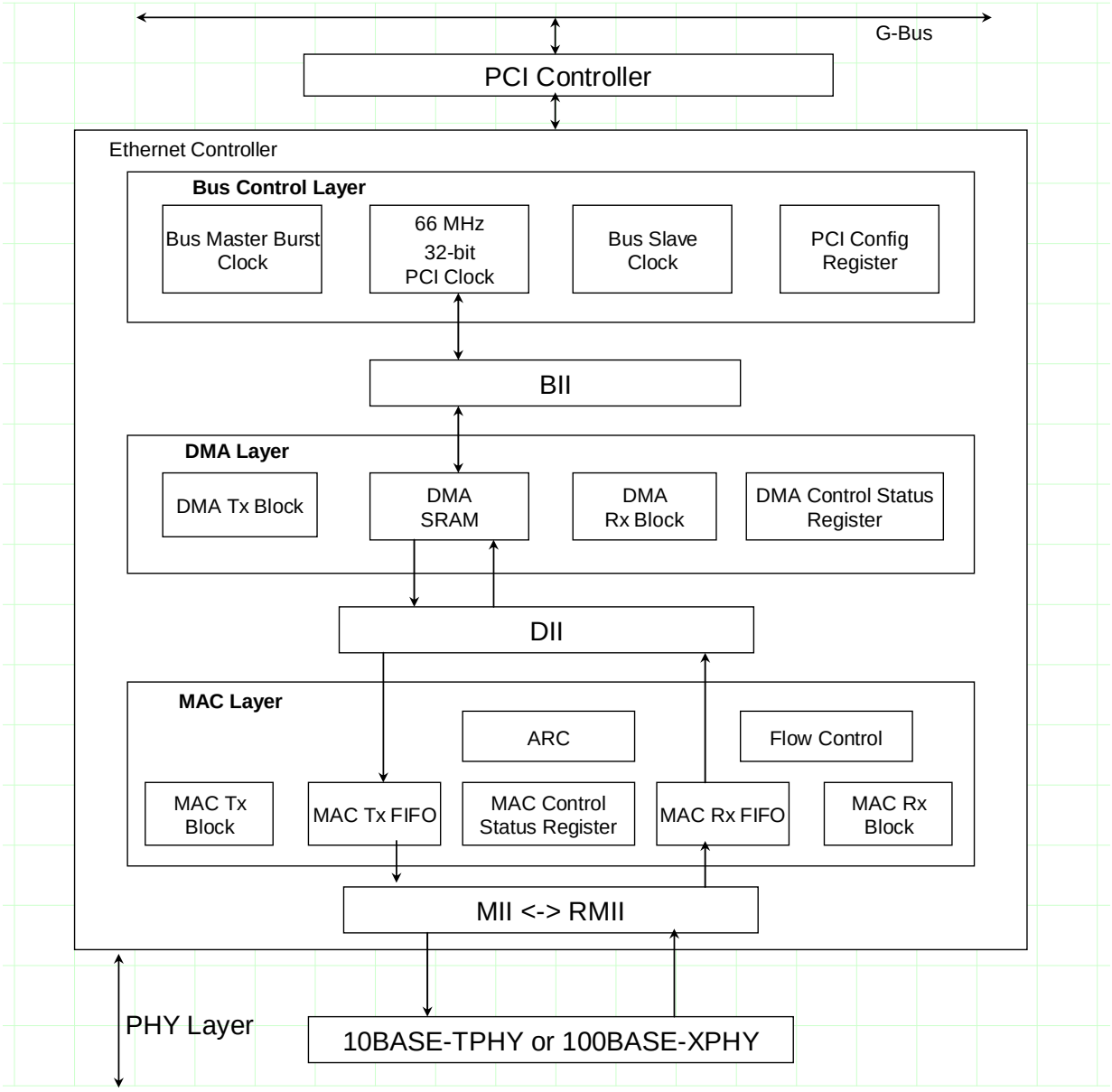


Figure 18-2 Ethernet Controller Block Diagram

18.3. Detailed explanation

The two Ethernet Controllers in TX4939 are identical to each other both in behavior and application and differ only in the method used to access each of the two Ethernet Controllers register space and their respective interrupt mapping. This difference is due to the fact that each of the two Ethernet Controllers are instantiated as a unique PCI device on the internal PCI Bus thereby requiring unique IDSEL and Interrupt signals for each of the two Ethernet Controllers. The following is a detailed explanation of one TX4939 Ethernet Controller. The description provided in the following sections is applicable to all two Ethernet Controllers unless specified otherwise.

18.3.1. Accessing the Ethernet Controller

Both Ethernet Controllers are connected to the chip-internal PCI Bus. The chip-internal PCI Bus is connected to the G-Bus via the PCI Controller (PCIC1). Therefore, you should access the two Ethernet Controllers via PCI Controller No. 2.

PCI Controller No. 2 is a circuit that is equivalent to the PCI Controller No. 1(PCIC). The Ethernet Controllers only supports the PCI Controller No. 2 functions that it requires to operate. Following are the differences with PCIC.

- *Host mode only. (Does not support the Satellite mode.)*
- *Does not support EEPROM for Configuration data storage*
- *Does not support an external PCI arbiter*
- *Does not support type 1 configuration*
- *Does not support the Interrupt Acknowledge command*
- *Does not support the Special Cycle command*

18.3.1.1. PCI Controller No. 2 control registers

Table 18-2 lists the PCI Controller No. 2 registers. For the details of each register or the operation of the PCI Controller, see Chapter 16.

Table 18-2 PCI Controller Control Registers

Section	Address	Size	Mnemonic	Register Name
18.4.2.1 18.4.2.2	0x7000	32	PCIID	ID Register (Device ID, Vendor ID)
18.4.2.3 18.4.2.4	0x7004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
18.4.2.5	0x7008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
18.4.2.6	0x700C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
	0x7010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
	0x7014	32	P2GM0PUBASE	P2G Memory Space 0 PCI Upper Base Address Register (Base Address 0 Upper)
	0x7018	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
	0x701C	32	P2GM1PUBASE	P2G Memory Space 1 PCI Upper Base Address Register (Base Address 1 Upper)
	0x7020	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
	0x7024	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
	0x702C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
	0x7034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
	0x703C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
	0x7040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
	0x7080	32	G2PSTATUS	G2P Status Register
	0x7084	32	G2PMASK	G2P Interrupt Mask Register
	0x7088	32	PCISSTATUS	Satellite Mode PCI Status Register
	0x708C	32	PCIMASK	PCI Status Interrupt Mask Register
	0x7090	32	P2GCFG	P2G Configuration Register
	0x7094	32	P2GSTATUS	P2G Status Register
	0x7098	32	P2GMASK	P2G Interrupt Mask Register
	0x709C	32	P2GCCMD	P2G Current Command Register
	0x7100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
	0x7104	32	PBACFG	PCI Bus Arbiter Configuration Register
	0x7108	32	PBASTATUS	PCI Bus Arbiter Status Register
	0x710C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
	0x7110	32	PBABM	PCI Bus Arbiter Broken Master Register
	0x7114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for diagnositics)
	0x7118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for diagnostics)
	0x711C	32	PBACSTATE	PCI Bus Arbiter Currrent State Register (for diagnostics)
	0x7120	64	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
	0x7128	64	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
	0x7130	64	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
	0x7138	64	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
	0x7140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
	0x7144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
	0x7148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
	0x714C	32	G2PIOMASK	G2P I/O Space Address Mask Register
	0x7150	64	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
	0x7158	64	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
	0x7160	64	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
	0x7168	64	G2PIOPBASE	G2P I/O Space PCI Base Address Register
	0x7170	32	PCICCFG	PCI Controller Configuration Register
	0x7174	32	PCICSTATUS	PCI Controller Status Register
	0x7178	32	PCICMASK	PCI Controller Interrupt Mask Register
	0x7180	64	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
	0x7188	64	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
	0x7190	64	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
	0x7198	64	P2GIOGBASE	P2G I/O Space G-Bus Base Address Register
	0x71A0	32	G2PCFGADRS	G2P Configuration Address Register
	0x71A4	32	G2PCFGDATA	G2P Configuration Data Register
	0x71C8	32	G2PINTACK	G2P Interrupt Acknowledge Data Register
	0x71CC	32	G2PSPC	G2P Special Cycle Data Register
	0x71D0		PCICDATA0	Configuration Data 0 Register

Section	Address	Size	Mnemonic	Register Name
	0x71D4		PCICDATA1	Configuration Data 1 Register
	0x71D8		PCICDATA2	Configuration Data 2 Register
	0x71DC		PCICDATA3	Configuration Data 3 Register
	0x71E0	32	P2GM0CFG	P2G Memory Space 0 Configuration Register
	0x71E4	32	P2GM1CFG	P2G Memory Space 1 Configuration Register
	0x71E8	32	P2GM2CFG	P2G Memory Space 2 Configuration Register
	0x7200	64	PDMCA	PDMAC Chain Address Register
	0x7208	64	PDMGA	PDMAC G-Bus Address Register
	0x7210	64	PDMPA	PDMAC PCI Bus Address Register
	0x7218	64	PDMCTR	PDMAC Count Register
	0x7220	64	PDMCFG	PDMAC Configuration Register
	0x7228	64	PDMSTATUS	PDMAC Status Register

18.3.1.2. Interrupt signaling

PCI Controller No. 2 signals the following interrupt to the Interrupt Controller (IRC):

PCIC1 Interrupt (Interrupt Number: 36, PCIC1INT)

This interrupt is similar to a combination of two PCI Controller 0 interrupt signals into a single interrupt signal. For the content of each interrupt, see subsection 10.3.11.

The two Ethernet Controllers signals the following interrupts to the Interrupt Controller (IRC):

Ethernet Channel 0 Interrupt (Interrupt Number: 6)
Ethernet Channel 1 Interrupt (Interrupt Number: 43)

For the content of these interrupts, see 18.4.3.9 Interrupt Source Register.

18.3.1.3. IDSEL signal connections

Internal PCI Bus AD (Address, Data) signals are connected to Ethernet Controller IDSEL signals. The connection destination follows below.

Ethernet Channel 0 IDSEL signal: Internal PCI Bus AD[31]
Ethernet Channel 1 IDSEL signal: Internal PCI Bus AD[30]

18.3.2. Data structure

The Ethernet Controller exchanges control information and data using the following data structure.

- *Frame Descriptors*
- *Buffer Descriptors*
- *Data Buffer*

Figure 18-3 shows the relationships of the above data structures.

A frame descriptor consists of a 4-byte pointer that points to the next frame, a field dedicated to a system or application program, a frame status field, a control field for all frames, and a corresponding group buffer descriptor. The buffer descriptor contains a pointer to the data buffer and buffer control information. The data buffer is a data storage area in Byte units that is stored in either the Little Endian or Big Endian order. For details on these data structures, see 18.3.7 Memory configuration.

The DMA engine responds to the transfer of data that is positioned at the byte boundaries. Frame descriptors must be aligned to 16-Byte boundaries. Also, buffer descriptors must be aligned to 8-Byte boundaries. The DMA engine Burst transfers 4-Byte aligned data as much as possible. However, whether data buffers are aligned to 4-Byte boundaries or not does not have much effect on performance. The DMA engine accesses only words or parts of words not aligned to 4-Byte boundaries at the start and end of Block transfer. Part way through Block transfer however, the DMA engine performs word-unit 4-Byte access.

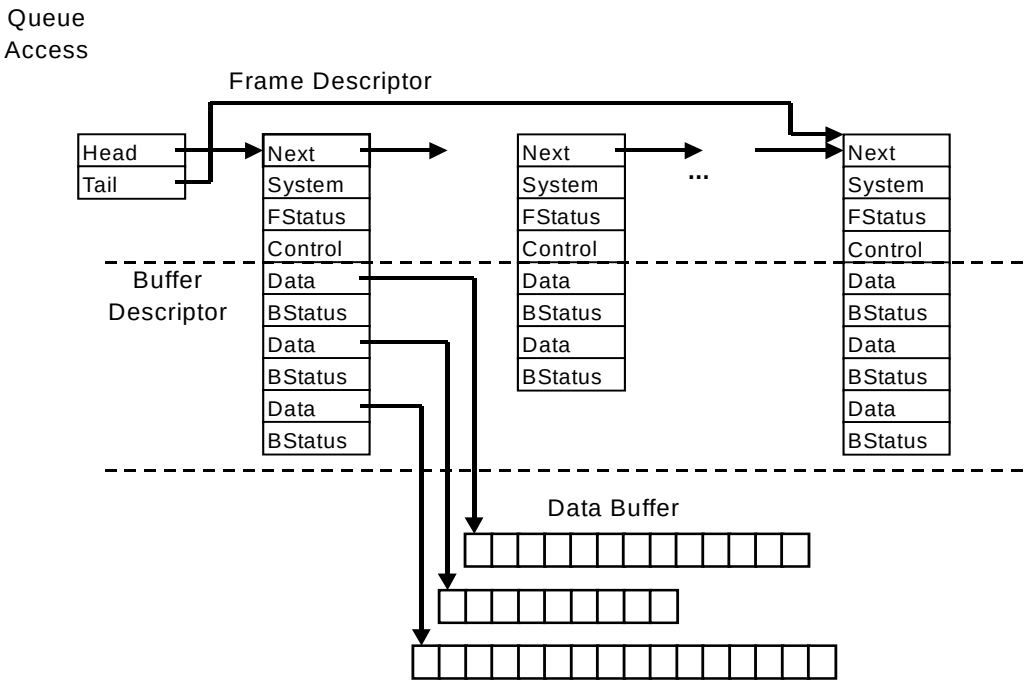


Figure 18-3 Data Structure Outline

18.3.3. System control model

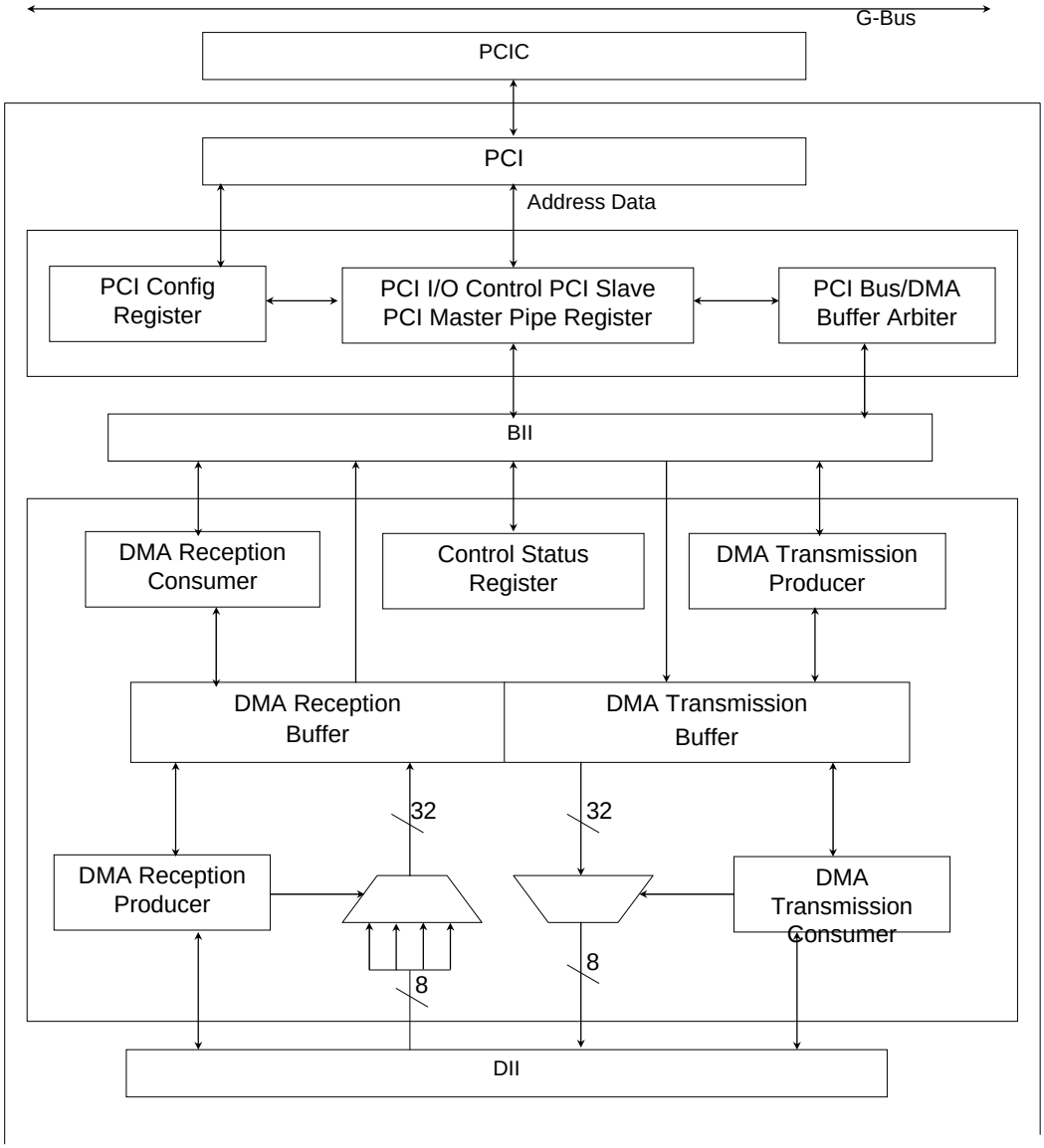
The Ethernet Controller as two programmable modes: the Interrupt Drive mode and the Polling mode.

- *Interrupt Drive mode*
  This mode generates an interrupt each time a packet or group packet transmits or receives.
- *Polling mode*
  This mode always polls to see if there is a subsequent packet to transmit or receive. When data or a descriptor is added to the frame descriptor queue, the Ethernet Controller performs transmission. The Host CPU checks the status of the frame descriptor for the received packet and judges reception completion. The CPU does not use an interrupt.

By enabling or disabling interrupts for each frame descriptor, the Ethernet Controller can arrange the stage layout to enable the handling of multiple packets between interrupts. In this way, the ability of the system to handle multiple packets at once and to cache the handling instructions makes it possible to reduce interrupt handling overhead and improve performance.
Furthermore, unless major errors occur, you can set the Ethernet Controller to not issue interrupts. In this situation, you have to put several controls in the Enable state so start/end of traffic and polling in the Idle state is handled efficiently. For details, see 18.3.9 DMA Operation.

18.3.4. Functional overview

Figure 18-2 is a diagram of the entire Ethernet system. Figure 18-4 and Figure 18-5 are more detailed function block diagrams of the Ethernet Controller. Figure 18-4 is a function block diagram of DMA. Figure 18-5 is a function block diagram of MAC.



Rev 2.20

Figure 18-4 DMA Function Block

Figure 18-4 and Figure 18-5 both show the relationship with the DMA-independent interface (DII). DII is an internal interface that has simple handshake signals and interfaces MAC-internal FIFO with the DMA block-internal DMA buffer. The two FIFOs in the MAC are 1 Byte wide. The DMA block contains DMA buffers that are wider and deeper than MAC. Figure 18-4 shows the relationship with the bus-independent interface (BII). The BII is an internal interface that has Bus Request/Bus Grant protocols for Burst access performed by the Bus Master. The BII makes it possible to connect the DMA engine to a 32-bit PCI Bus Controller or another bus controller.

18.3.4.1. Overview of PCI and DMA

Figure 18-4 shows the PCI Bus configuration, blocks that perform control, DMA buffer and data exchange.

- *PCI I/O Control Block*
  Generates and acknowledges PCI control signals.
- *PCI Slave Control Block*
  Acknowledges and controls transactions when the Ethernet Controller is the target device.
- *PCI Master Control Block*
  Starts and controls transactions requested by the DMA engine of the Ethernet Controller.
- *PCI Pipe Register Group*
  Buffers data so the DMA engine can continue Burst transfer with 1-1-1-1 timing over a long period of time.
- *PCI Configuration Register*
  Flexibly performs PCI system setup.
- *DMA Control Status Register*
  Sets up and controls DMA.
- *Arbiter Block*
  Arbitrates access to the PCI Bus or the DMA buffer.
- *Producer Block, Consumer Block*
  Controls transmission/reception data output/input to/from the DMA buffer.

The DMA buffer the DMA engine controls has a transmission area and a reception area. The DMA Transmission buffer stores the data and status information of multiple packets that are currently being transmitted. The DMA Reception buffer stores the data and status information of multiple packets that were received. Each buffer has a Producer Block that controls the data stored in a buffer and a Consumer Block that controls data removed from a buffer.
The PCI Bus/DMA Buffer arbiter determines whether the consumer or producer State Machine has higher priority access permission to the PCI Bus/DMA Buffer. The priority changes dynamically and, if possible during Burst transfer, gives priority to the State Machine that controls the PCI Bus. However, when either the MAC Reception FIFO becomes nearly full or the MAC Transmission FIFO becomes nearly empty during transmission, priority is given to that FIFO. In other situations, the round-robin scheme is used to provide service in a fair manner.
The DMA Controller block consists of the circuits required for the Bus Master to perform read/write operations via the PCI Bus.

- *Busrt Size Control Circuit*
  Optimizes PCI and system performance.
- *Transmission Threshold Control Circuit*
  Matches transmission latency to PCI Bus latency.
- *Big Endian Byte Translation Circuit*
  Supports data transfer to Big Endian format processors.
- *Buffer Distribution/Sharing Control Circuit*
  Can distribute then store one frame in multiple buffers. Can also store multiple frames in a single buffer and increase the usage efficiency.
- *Polling Control Circuit*
  Polls transmission packets. Is optional.
- *Transmission Wakeup Control Circuit*
  Performs control required to start transmission when the data is prepared.
- *Early Notification Circuit*
  Circuit that generates signals for starting to handle reception data before the data ends.
- *Interrupt Enable Control Circuit*
  Adjusts controller operation as the protocol requires.

18.3.4.2. MAC Overview

Figure 18-5 is a function block diagram of MAC. MAC consists of a Transmission Block, Reception Block, Control/Status Register group, Flow Control Block, and a Serial Controller. The Serial Controller is an MII station management interface.

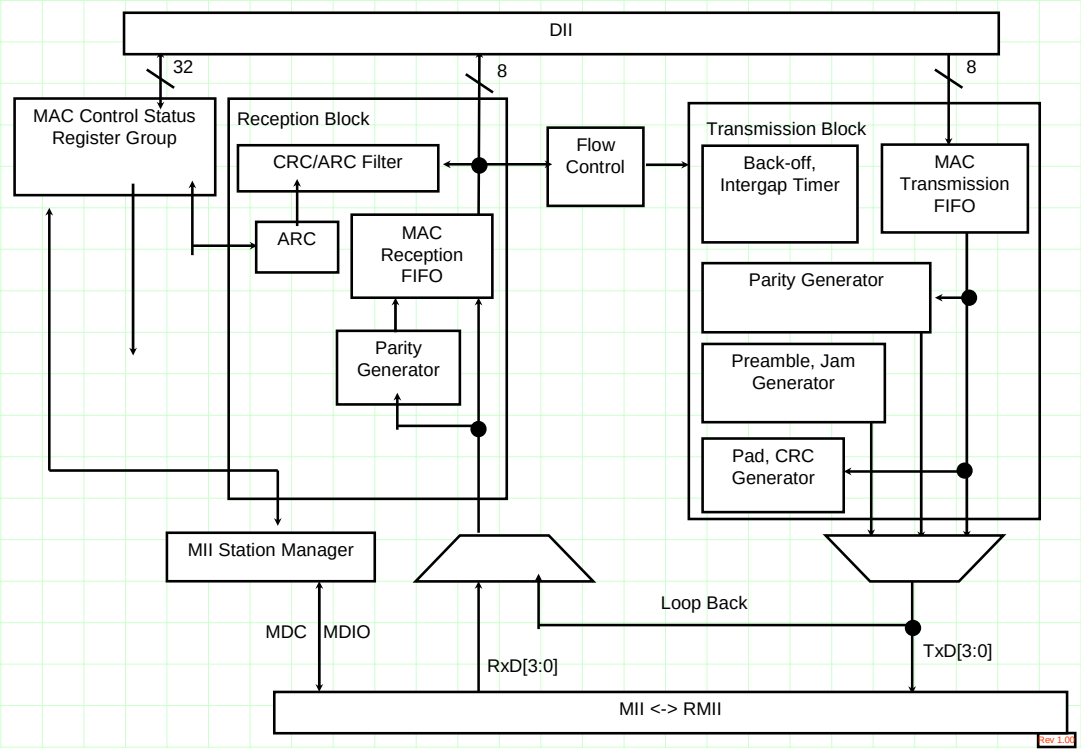


Figure 18-5 MAC Function Block

The media-independent interface (MII) transfers signals between the 100Base-T compatible physical layer and the Transmission/Reception Block. MII is described in the IEEE 802.3 standard.
The Transmission Block buffers data transmitted from the MAC Transmission FIFO, assembles packets, and then transfers them to the MII. Included in the Transmission Block are circuits that generate Preamble Bytes, Jam Bytes, Pad Bytes, and CRC values. Also included is a parity check circuit, a timer that creates back-off delay when a collision occurs, and a time that creates a cap between packets that were transmitted.
The Reception Block expands packets received from MII, and then stores them in MAC Reception FIFO. Included in the Reception Block are a circuit that checks the CRC value, a circuit that generates parity to protect data in the FIFO, and a circuit that checks the packet length. Also included is an address recognition circuit (ARC) that, based on the receiving address, judges whether to accept or discard a packet.
You can use a loop back circuit to separate MAC from the MII and physical layer and perform MAC layer testing.
The MAC function block performs control to perform network manipulation such as the following:

- *Control to include a stop request at the end of a packet currently being transmitted or received, and enable or disable the transmission/reception circuit*
- *Interrupt enable/disable control for each condition*
- *Address recognition control for up to 21 addresses*
- *Counter, status bit for collecting network management data*
- *Loop back or other control that aids network failure diagnosis*
- *Pause operation enable control for pausing the transmitter when a MAC control frame described by Pause operation is received*
- *MAC control frame transmission control for enabling Pause or other MAC control frame generation even when the transmitter has paused*
- *MAC control frame pass through control for enabling hardware or software to handle MAC control frames in another format*

18.3.5. DMA function block

18.3.5.1. DMA Transmission Controller

The DMA Transmission Controller consists of two State Machines: Producer and Consumer. The Producer checks the frame descriptor of the transmission queue, then controls data transfer from the transmission queue to the DMA Transmission buffer. Also, after transmission ends, the Producer controls the writing of transmission status information that expresses the MAC status information for the transmitted data. The Consumer controls transfer of data from the MAC transmission engine or a large-capacity DMA transmission buffer to a small MAC transmission FIFO.

18.3.5.2. DMA Reception Controller

The DMA Reception Controller consists of two State Machines: Producer and Consumer. Producer controls data transfer from MAC Reception FIFO to the DMA reception buffer. Consumer allocates buffers from the empty buffer list, writes the frame descriptor and related buffer descriptor of the packet in the free descriptor area, then controls data transfer via the PCI Bus from the DMA Reception buffer to the system memory.

18.3.6. MAC function blocks

18.3.6.1. MAC Transmission Block

The Transmission Block takes charge of data transmission. The Transmission Block is compliant with the IEEE 802.3 carrier sense multiple access with collision detection method (CSMA/CD) protocol. This block also supports full duplex modes that can simultaneously perform transmission and reception. The Transmission Block consists of the following parts.

- *Transmission FIFO, FIFO Control Counter*
- *Preamble, Jam Oscillator*
- *Pad Byte, CRC Generator*
- *Parity Checker*
- *Back Off, Intergap Timer*

For details of the Transmission Block, see 18.3.8 MAC operation.

18.3.6.2. MAC Reception Block

The Reception Block takes charge of data reception. The Reception Block is compliant with the IEEE 802.3 carrier sense multiple access with collision detection method (CSMA/CD) protocol. This block also supports full duplex modes that can simultaneously perform transmission and reception. The Reception Block consists of the following parts.

- *Reception FIFO, FIFO Control Counter*
- *Address Recognition ARC Block*
- *CRC Generator, Tester*
- *Parity Generator*

For details on the Reception block, see 18.3.8 MAC operation.

18.3.6.3. Flow Control Block

The Flow Control Block has the following functions:

- *Recognize MAC control frames the Reception Block received*
- *Transmit MAC control frames (even when the transmitter is paused)*
- *Pause operation timer and counter*
- *Command/Status Register Interface*
- *Options for handing off MAC control frames to the software driver*

The reception circuit in the Flow Control Block recognizes the MAC control frame, and then performs the following Pause operation. First, the Data Length/Data Type field must have a specific value for a MAC control frame. Second, the ARC must recognize the destination address. Third, the frame length must by 64 Bytes including CRC. Fourth, the CRC test result must confirm the correctness of the frame. Finally, the frame must include a valid Pause operation code and operand.
If the Data Length/Data Type field does not contain a specific value that expresses a MAC control frame, MAC performs no operation and the packet is treated as a typical packet. If the ARC cannot recognize the destination address, MAC ignores the packet. If the packet length including CRC is not 64 Bytes, MAC does not perform Pause operation. In this case, if Pass Through is in the Enabled state, the driver is handed off.
If the Control bit of the Transmission Status Register is set, you can detect full duplex mode Pause operation or other MAC control function frames even if the transmitter itself is in the Pause state.
The Flow Control Block has two timers for Pause operation and a Control/Status Register for each timer. One timer, register pair is used when the received packet pauses the transmitter. The other timer, register pair is used to estimate the pause status of the transmission destination after the transmitter transmits a Pause command. The Command/Status Register Interface is used to access the Transmission Control Register, the Transmission Status Register, the Reception Control Register, and the Reception Status Register. You can use these registers to start transmission of a MAC Control frame, enable or disable the MAC control function, or access the Flow Control counter.
You can use the Control bit to select whether to completely handle MAC Control frames inside the Ethernet Controller or hand them off to the software driver. Therefore, default flow control is possible even if the software driver itself does not support flow control.

18.3.6.4. MAC Control Register, Status Registers

MAC has a group of Control Registers and Status Registers. These registers are used to control the Transmission Block or Reception Block, display MAC status, are used in the communication interface with CAM, and are used in the interface with the MII Station Manager. These registers can also access from PCI using the memory map or I/O map. For details on the MAC Control Register or Status Register, see 18.4.5 MAC Control, Status Register group.

18.3.6.5. MII Station manager

MAC in the Ethernet Controller handles station management data signals (MDIO and MDC) from the MII Controller, but it does not interpret them. Using a serial interface defined by MII, the MII Station Manager reads and writes to/from the Control Registers and Status Registers in a PHY device whose configuration is set up.
When it is necessary to access these registers to negotiate configuration using an application specialized for a bridge, router, switching hub, etc., you can use the MAC Control Registers and Status Registers to trigger reads and writes through the Station Management Data Interface. For details, see 18.4.5.5 Station Management Registers..

18.3.6.6. Reception packet alignment

When storing Reception packets in the buffer, the Ethernet Controller can skip by the set byte count. By default, the Ethernet Controller does not skip Reception packets and these packets are on double-word boundaries. You can set the DMA Control Register to have the first buffer skip 1-3 Bytes. This function is convenient when aligning packets when internally necessary by handling IPs and decoding.

18.3.7. Memory configuration

This subsection describes the data structure the PCI-mounted Ethernet Controller uses when exchanging data with the Host system. The data structures are stored in system memory. There are three basic data structures:

- *Frame descriptor*
- *Buffer descriptor*
- *Data buffer*

These data structures are used in the following manners:

- *Transmission queue:* *Listing the frame descriptors of packets ready for transmission*
- *Reception queue:* *Listing the frame descriptors of received packets*
- *Buffer list:* *Listing the frame descriptors that have unused buffers for receiving data*

Figure 18-3 shows an outline of each data structure. This subsection describes each data structure in detail. Depending on the type of data structure, some queues in use may include different information. Each following item will describes this in detail. During continuous polling operation, a queue never becomes empty once it is created. One frame descriptor is always appended to the end of a queue. This dummy frame descriptor is used in processes that generate a descriptor to be transmitted. For details, see 18.3.9 DMA Operation. To start transmission, the system sets the address of the first frame descriptor in the Transmission queue in the Transmission Frame Pointer Register. The Ethernet Controller transmits while tracing the Transmission queue, then updates the status of the transmitted packet. The owner bit of the Frame Descriptor Status Field and Frame Descriptor Control field indicates “Transmission complete”. This enables the system software to handle the queue in situations such as when releasing the buffer. The Ethernet Controller fetches the buffer from the buffer list, and then writes the new frame descriptor or new buffer descriptor in a free descriptor area. Subsection 18.3.9 DMA operation describes this.

18.3.7.1. Frame descriptor

A frame descriptor consists of a pointer to the next frame descriptor in the queue, a System Data field, a Frame Length field, a Control field, and a Status field.
Table 18-3 shows the frame descriptor format.

Table 18-3 Frame Descriptor Format

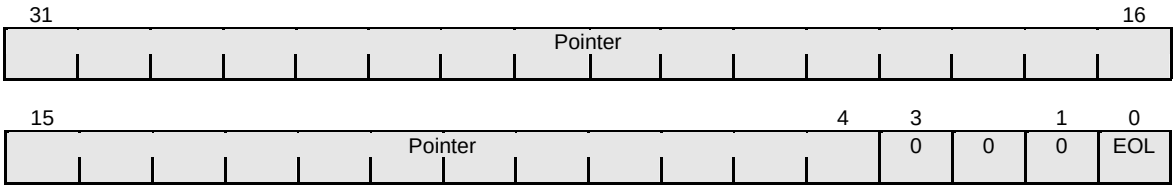
Byte 3	Byte 2	Byte 1	Byte 0	Offset
FDNext				00 h
FDSystem				04 h
FDStat				08 h
FDCtl		FDLength		0C h

FDNext	Next frame descriptor	Address of the next frame descriptor in the queue
FDSystem	Frame system data	Is used by the system or application software
FDStat	Frame descriptor status	Status field of this frame descriptor
FDCtl	Frame descriptor control	Control field of this frame descriptor
FDLength	Frame descriptor length	Field that expresses the length of this field

The Ethernet Controller retains the Frame System Data field (FDSystem). The FDSystem field can be used by system or application programs. The initial value of the frame descriptor written to the Reception queue is fetched from the frame descriptor of the current buffer list.
Depending on the queue type, the usage of the FDNext, FDCtl, FDStat, or FDLength fields varies. See the following items for an explanation.

18.3.7.1.1. FDNext field (next frame descriptor)

The FDNext field stores a flag that indicates the end of list (EOL) or a pointer to the next frame descriptor in the same queue. The frame descriptor must be aligned to a 16-byte boundary. In other words, bits 0-3 of the valid pointer must be "0".



Bit(s)	Mnemonic	Field Name	Description
31 : 4	Pointer	28-bit Pointer	28 bit Pointer When EOL=0, the upper 28 bits of the address of the next frame descriptor in this queue is stored.
3 : 1			Is fixed to 000
0	EOL	End Of List flag	EOL 0: Pointer is valid. 1: End of list. You have to wait until this flag is cleared.

Figure 18-6 FDNext Field

Regardless of the queue type, by setting the EOL bit, you can use the FDNext field stop the list consumer (consumption side processing mechanism). The consumer must wait for the list producer (generation side processing mechanism) to clear the EOL bit and store a valid pointer. In the case of a buffer list queue, you can use the FDNext field to create a chain to the next buffer pool from any buffer pool. When not creating a buffer list chain, the software driver has to set its own address in the FDNext field. Doing so makes it possible for the Ethernet Controller to recheck so it can reuse the same buffer area. You can also set the EOL bit and stop the Ethernet Controller.

18.3.7.1.2. FDSystem field (frame descriptor system data)

The FDSystem field is a 32-bit field that is reserved for use by the system software. You can use this field to store pointers to the information table, pointers to C++ virtual functions, etc.
The Transmission queue does not use the FDSystem field.
In the case of a Reception queue, the Ethernet Controller copies the content of the FDSystem field of a frame descriptor that retains a buffer list to which is allocated the first buffer descriptor that stores the frame.

18.3.7.1.3. FDStat field (frame descriptor status)

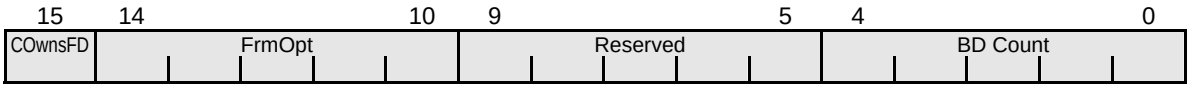
The Transmission queue or Reception queue uses the FDStat field to display the transmission/reception complete status. For the meaning of each Status bit, see the descriptions of the TX_Stat Register and Rx_Stat Register in subsection 18.4.5 MAC Control, Status Register group.
The reception empty buffer list does not use the FDStat field.

18.3.7.1.4. FDLength field (frame descriptor length)

The Transmission queue does not use the FDLength field.
In the case of the Reception queue, the Ethernet Controller sets the FDLength field to the total packet length.
In the case of a buffer list, the FDLength field is used to count the number of empty buffer descriptors allocated to the queue. The Ethernet Controller accesses the buffer list frame descriptor through the Buffer List Frame Pointer Register. When the Ethernet Controller encounters a buffer that it does not own, it sets the BL_Ex bit of the Interrupt Source Register and then waits for the system to clear it. The Ethernet Controller treats the FDLength field value as the upper limit, then reads the buffer descriptor. As the Ethernet Controller nears the end of the list, it reads out the next frame descriptor that the FDNext field indicates, as described in 18.4.3.8 Free Descriptor Area (FDA) Register.

18.3.7.1.5. FDCtl field (frame descriptor control)

This item shows the FDCtl field configuration and application. The Transmission queue or Reception queue uses the COwnsFD bit to synchronize the Ethernet Controller and System process. The Transmission queue or buffer list uses the FrmOpt field. The Transmission queue or Reception queue uses the BDCount field (the buffer list uses the Frame Descriptor Length field (FDLength) to enable the use of buffer pools larger than the BDCount field permits). For each reception packet, an excessive buffer descriptor error is issued when more than 28 buffer descriptors are used.



Bit(s)	Mnemonic	Field Name	Description
15	COwnsFD	Frame Descriptor Owner	COwnsFD 1: The Ethernet Controller owns the frame descriptor after the system sets the COwnsFD bit. 0: The system owns the frame descriptor after the Ethernet Controller clears the COwnsFD bit.
14 : 10	FrmOpt	Frame Option	Frame Option Control option for each frame (see the following description)
4 : 0	BDCount	Buffer Descriptor Count	BDCount This is the number of allocated buffer descriptors (1-28).

Figure 18-7 FDCtl Field

The Transmission queue uses the FrmOpt field and sets the transmission characteristics of each packet.

- *10000* *Big Endian order*
- *01000* *Issue interrupt after transmission*
- *00100* *Do not add CRC*
- *00010* *For short frames, do not add PAD*

You can use combinations of the above bits to set various transmission characteristics. For example, "01110" means "Little Endian, issue interrupt after transmission, no CRC, no padding for short packets".
In applications such as hubs where transmission packets are being received from both Big Endian and Little Endian transmission sources, controlling Big Endian for each packet is useful. In computer applications, it is easier to use the global Big Endian control bit described in 18.4.3.1 DMA Control Register.
Unless you set a global enable bit like that described in 18.4.3.6 Reception Fragment Size Register, the Reception buffer list uses the option set in the FrmOpt field to control packing and the endian.

- *10000* *Big Endian order*
- *00001* *Enables packing for this frame's buffer, ignores any global enable bit*

When packing is enabled, the RxFragSize Register controls the packing algorithm.

18.3.7.2. Buffer Descriptor

Each buffer descriptor consists of a pointer to the data buffer, a control byte, a status byte, and a 2-byte Buffer Length field. Table 18-4 indicates the buffer descriptor format.

Table 18-4 Buffer Descriptor Format

Byte 3	Byte 2	Byte 1	Byte 0	Offset
BuffData				00h
BDCtl	BDStat	BuffLength		04h

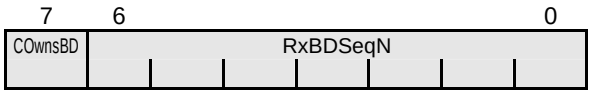
BuffData	Buffer Data Pointer	Address of the buffer that stores data (32 bits)
BDCtl	Buffer Descriptor Control	Control of this buffer descriptor
BDStat	Buffer Descriptor Status	Status of this buffer descriptor
Buff Length	Buffer Length	Field that expresses the length of this buffer

When a buffer is listed in the buffer list, the BuffData field indicates the buffer head and the BuffLength field indicates the size of buffers allocated as unused buffers. When the buffer is used in the Transmission queue or Reception queue, the BuffData field indicates the descriptor head and the BuffLength field indicates the data length.
When a buffer is placed in the buffer list, the system software is responsible for allocating the Buffer Length field and setting the size.
Similar to the frame descriptor, usage of the BDCtl field or BDStat field varies depending on the queue type.

Note: As shown in step 4 of 18.3.8.4 Frame transmission procedure, more than 8 bytes of data is required as a Transmission packet data. One Transmission packet cannot consist of multiple data buffers, so you have to implement your design so the BuffLength (buffer length) total is 8 bytes or more.

18.3.7.2.1. BDCtl field (buffer descriptor control)

The transmission queue does not use the BDCtl field. In the case of the reception queue, the number of buffer descriptors per buffer area is set in the BDCtl field. The first buffer descriptor in a frame is number 0, and the subsequent buffer descriptors are number 1 and number 2.



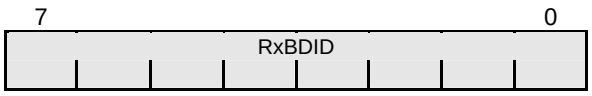
Bit(s)	Mnemonic	Field Name	Description
7	COwnsBD	Buffer Descriptor Owner	COwnsBD 1: The Ethernet Controller owns the buffer descriptor. When the system sets the COwnsBD bit, the buffer can freely receive data. 0: The system owns the buffer descriptor. When the Ethernet Controller clears the COwnsBD bit, it indicates that the buffer is full.
6 : 0	RxBDSeqN	Reception Buffer Descriptor Number	RxBDSeqN After receiving data, this field indicates the running number of this buffer in the current buffer area.

Figure 18-8 BDCtl Field

In an empty buffer list, the BDCtl field is used to record buffer ownership. This makes it possible to allocate the buffer descriptor and synchronize release of the buffer. This also prevents the Ethernet Controller from using the buffer in a loop before the system empties the buffer.

18.3.7.2.2. BDStat field (buffer descriptor status)

In the case of a transmission queue, the Ethernet Controller does not use the BDStat field. In the case of a reception queue, the BDStat field is used as the buffer ID. This value is copied from an empty buffer queue.



Bits	Mnemonic	Field Name	Description
7 : 0	RXBDID	Reception Buffer Descriptor ID	RXBDID Value of the buffer descriptor identification number

Figure 18-9 BDStat Field

In the case of an empty buffer queue, the BDStat field is used to hand the buffer descriptor identification number to the Ethernet Controller.

Note: You can only apply buffer descriptor identification numbers freely if there are no more than 256 buffers in a single buffer pool.

18.3.8. MAC operation

This subsection describes MAC operation in detail with respect to the following content:

- *Format of MAC frames and packets*
- *Initializing MAC*
- *Access MAC Registers*
- *Frame transmission procedure*
- *Frame reception procedure*
- *Operation of the address recognition circuit (ARC)*
- *Pause operation in full duplex transfers*
- *Signaling transmission/reception errors*
- *Accessing station management data*

18.3.8.1. Format of MAC frames and packets

Figure 18-10 shows the format of IEEE 802.3 Ethernet packets. Standard packets have the following fields.

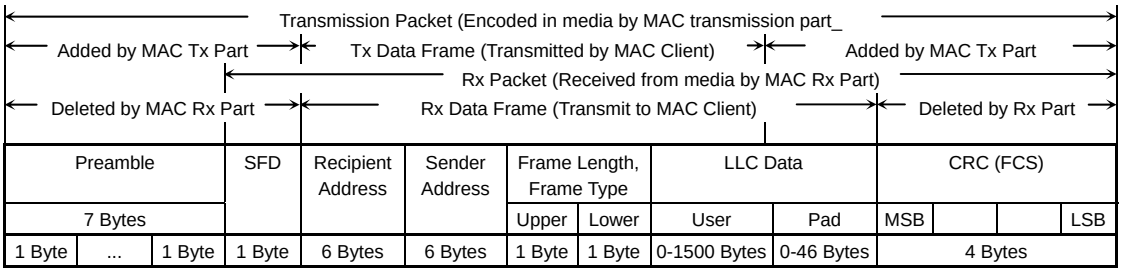


Figure 18-10 Ethernet Packet Frame Fields

- *Preamble— repeat the same byte 7 times*
  Each byte is fixed to 10101010 and is transmitted from left to right. The preamble is sometimes shortened during transmission and is not necessary when receiving data.
- *Start Frame Delimiter (SFD)— 1 Byte*
  The SFD is fixed to 10101011 and is transmitted from left to right. It is necessary when receiving data.
- *Recipient address— 6 Bytes*
  This is a unicast or multicast address. ARC can use the recipient address and can freely perform address filtering.
- *Sender address— 6 Bytes*
  MAC does not check the sender address. However, to make the sender address a valid station address, the first bit (LSB) of the first Byte transmitted must be "0".
- *Frame length or frame type— 2 Bytes*
  Is transmitted upper byte first. In the IEEE802.3 standard, a value of 1500 or less is defined as the frame length, and any value greater than 1535 is defined as the frame type. If the value is 1500 or less, it indicates the byte count of the logic link control (LLC) data in the Data field. MAC recognizes 8808h as the MAC control frame and specially recognizes 8100h as a tagged VLAN frame.
-
- *Logic link control (LLC) data— 0-1500 Bytes*
  LLC data consists of two fields: User Data and Pad Data.
- *User Data— 0-1500 Bytes*
- *Pad Data— 0-46 Bytes*
  When the user data is less than 46 Bytes long, MAC can add padding bytes and make the LLC data 46 Bytes long.

- *Cyclic redundancy check (CRC)— 4 Bytes*
  Also referred to as the frame check sequence (FCS), this value is calculated from all other fields except the preamble, SFD, and CRC itself.

  The preamble, SFD, pad data, and CRC are added on by the transmission side. You can also pad data using the software. CRC can use the Transmission Control Register to suppress padding. You can use the Reception Control Register to control deletion of CRC. You can delete pad data using either the DMA engine or the software driver.

  Except for CRC, the MAC transmits each byte starting from the least significant byte. In this document, all bytes transmitted or received are collectively referred to as "packets". The term "frame" refers to a part that you provide during transmission or a part that is provided to you during reception.

  Standard IEEE802.3 frames are transformed by various factors and options.
-
- *Depending on the PHY, there are cases where the preamble length is not 7 Bytes. Sometimes a repeater shortens the preamble. The transmission part sends the preamble in a standard format, but the reception part does not care even if there is no preamble or the preamble is more than 7 Bytes long. SFD follows the preamble.*
- *When in the Short Packet mode, LLC data less than 46 Bytes long is allowed. In this case, there are options for suppressing padding when transmitting data or for allowing the reception of short packets.*
- *When in the Long Packet mode, LLC data larger than 1500 Bytes is allowed. There is an option for allowing the reception of long packets.*
- *There is an option for suppressing the addition of CRC fields.*
- *There is also an option for allowing the reception of packets that do not have a valid CRC field.*

18.3.8.1.1. Format of Recipient Addresses

Bit 0 of a recipient address specifies the address type. This bit indicates whether the address is an individual address or a group address. Group addresses are also referred to as multicast addresses. Individual addresses are also referred to as unicast addresses. Broadcast addresses are special group addresses that are FF-FF-FF-FF-FF-FF hexadecimal addresses.
Bit 1 distinguishes locally managed addresses from globally managed addresses. If an address is managed globally (universal), bit 1 is set to "0". If an address is allocated locally, bit 1 is set to "1". Bit 1 is set to "1" for broadcast addresses.

Recipient address (first Byte)

Bit(s)	Mnemonic	Field Name	Description
7 : 2	Rest	Remaining Bit	Rest Remaining bits of the first Byte are the recipient bits.
1	U/L	Universal/Local	U/L 0: Universal address 1: Local address
0	I/G	Individual/Group	I/G 0: Individual address 1: Group address

Figure 18-11 Format of Recipient Addresses

18.3.8.1.2. Special flow control recipient address

The IEEE802.3 standard specially prescribes a recipient address of 01-80-C2-01-00-01 for PAUSE manipulation packets to realize full duplex flow control. In order for the MAC to receive packets including the special recipient address for PAUSE manipulation, you have to set the address in one entry in ARC memory, enable that entry, and then activate ARC itself.
The details are described in the following paragraphs. However, part of the ARC entry uses the SdPause bit of the Transmission Control Register and is also used when generating flow control frames.

18.3.8.2. Initializing MAC

After powering up or after performing reset, the MAC Control Status Register is initialized as described in section 18.4 Registers.
The transmission collision count and ARC data is not initialized when you power up or reset the Ethernet Controller. The transmission collision count is reset when a new packet is transmitted. Initialize ARC memory before enable ARC.

18.3.8.3. MAC Register access

Access to the MAC Register is controlled through the PCI Bus Interface. For details on read access and write access of the register, including the MAC Control Register, see 18.3.9 DMA operation.

18.3.8.3.1. Clearing special registers

The Lost Packet Error Count Register is cleared when it is read. This makes it possible to synchronize it to software drivers that tabulate the total error count.
The Transmission Status Register and Reception Status Register are cleared at the beginning of the next packet. Therefore, values read from the Register Interface may not be stable. These register values are stored in the FDStatus field of the frame descriptor in memory for each transmission or reception packet. You have to use the software to check the status value retained in the system data structure.

18.3.8.4. Frame transmission procedure

1. To transmit a frame, the Transmission Enable bit (TxEn) of the Transmission Control Register must be set, and the Transmission Halt Request bit (TxHalt) must be cleared. Furthermore, the Halt Transmission Immediately bit (TxHalt) and the Halt Request bit (HaltReq) of the MAC Control Register must be cleared. Usually, the above conditions (such as storing a valid frame descriptor address in the Transmission Frame Descriptor address) are set after initializing the DMA Controller. In this way, MAC instructs the DMA engine to transmit a frame to MAC Transmission FIFO. At this time, the DMA Transmission Controller controls transfers to the MAC Transmission FIFO.

2. The MAC Transmission Block starts transmitting data in the FIFO. However, the first 64 Bytes are held in the FIFO until the net is fetched. Then, the MAC Transmission Block requests the next data and continues to transmit until the DMA Transmission Controller indicates the end of transmission data. The MAC Transmission Block generates pad bytes when necessary, adds CRC to the end of a packet, then ends transmission. The MAC Transmission Block will then set the Transmission Complete bit (Comp) of the Transmission Status Register to signal that transmission has ended. Finally, depending on the Interrupt Enable Register settings, an interrupt may occur.

3. Data transfer via the RMII Interface is driven by a 50MHz or 5.0 MHz RMII Reference Clock (ExRCLK).

4. The MAC Transmission Block must not start transmitting to the net until 8-byte data is stored in the MAC Transmission FIFO. The first 8 bytes transmitted are the preamble and Start Frame Delimiter (SFD), so the allowable DMA latency when starting transmission is 16-byte time. The DMA Transmission Block does not transmit data to the MAC Transmission FIFO until either the entire packet is stored in the DMA Transmission buffer or the byte count set in the Transmission Threshold Register is stored in the DMA Transmission buffer. If a Transmission Underrun error is generated, you can avoid an underrun by increasing the setting of the Transmission Threshold Register.

5. The MAC Transmission Block performs a parity check. If a Parity error is generated, the MAC Transmission Block aborts transmission, then resets the FIFO and sets the Transmission Parity Error bit (TxPar) of the Transmission Status Register.

18.3.8.4.1. IEEE 802.3 transmission protocol

The MAC Transmission Block consists of three State Machines. The Main State Machine executes MAC layer protocol and controls the other two State Machines (Gap State Machine and Back Off State Machine). The Gap State Machine fetches the timing gap between packets, then counts them. The Back Off State Machine executes the Back Off or Resend algorithms of the 802.3 CSMA/CD protocol.

18.3.8.4.2. Interpacket gap (IPG) timing

When in the half duplex mode, the Gap State Machine measures 96-bit time from the point when the Carrier Sense signal is deasserted. This time becomes the interpacket gap. Gap State Machine splits the 96-bit time into 64-bit and 32-bit time, then precisely controls the timing at which transmission starts. If there is traffic with the first 64-bit time, the Gap State Machine resets the counter to 0, then starts counting from the beginning. If there is traffic during the remaining 32-bit time, the count continues as is and signals that 96-bit time has elapsed.
When in the full duplex mode, Gap State Machine starts counting along with transmission completion, then signals transmission completion after 96-bit time passes.

18.3.8.4.3. Collision process and back off

When the Main State Machine detects a collision, it starts up the counter of the Back Off State Machine, waits for the Back Off time to elapse, then attempts to resend the packet that caused the collision. The Back Off time is a multiple of 512-bit time (including $\times$ 0). When a collision happens in the same packet, the Main State Machine advances the internal Trial Count counter by 1 each time this happens. Then, it causes an 11-bit pseudorandom number generator to generate random numbers and output a subset of it. Each time this is retried, the subset increases in size by 1 bit. In this way, the following formula is invoked by the hardware.

$$0 \leq r < 2^k$$
$$k = \min(n, 10)$$

r is the slot time count that the MAC must wait when a collision occurs. *n* is the number of retransmission attempts. For example, after the first collision, n=1 and *r* is a random number between 0 and 1. In this case, the pseudorandom number generator is 1-bit wide and generates either 0 or 1 as a random number. For the second and subsequent attempts, *r* is a random number between 0 and 3. In other words, State Machine is n=2, so the pseudorandom number generator looks at the lower 2 bits and generates random numbers between 0 and 3.

To reduce the statistical probability of similar random number sequences being generated between multiple MACs using the same random number generator, MAC uses the CRC value of the packet that was previously transmitted successfully to recalculate a basic random number sequence.

18.3.8.4.4. RMII transmission operation

If there is data to be transmitted, there is no problem with the interpacket gap, and MII preparations are complete (in other words, there are no collisions for full duplex or for half duplex there are no collisions and there is no CrS),MAC transmits the preamble and SFD. After that, unless Short Packet transmission is enabled, MAC transmits 64 Bytes of data regardless of the packet length. If the packet length is less than 64 Bytes, the MAC Transmission Block pads the LLC Data field with zeroes. If CRC generation is enabled, the MAC Transmission Block appends CRC to the end of the packet. If a collision occurs during the first 64 Bytes (7-Byte preamble + SFD + 56-Byte frame), the MAC Transmission Block aborts transmission and transmits a jam pattern (a 32-bit sequence of ones). In this case, it advances the transmission retry counter by 1, then transfers control to the Back Off State Machine. After the Back Off time elapses, the MAC Transmission Block tries to resend the packet if there is no problem with the interpacket gap.

If no collisions occur, the MAC Transmission Block transmits the rest of the packet. If the first 64 Bytes are transmitted without any collision, the MAC Transmission Block provides the DMA engine with permission to overwrite these 64 Bytes. After transmitting the 64 Bytes, the MAC Transmission Block transmits the reset of the packet and appends CRC to the end of it. If an underrun occurs in the FIFO or there is a collision that occurs more than 16 times, the MAC Transmission Block makes no attempt to resend the packet and makes preparations to transmit the next packet registered in the queue. If a transmission error occurs, the MAC Transmission Block sets the appropriate bit of the Transmission Status Register. Also, depending on the Transmission Control Register settings, an interrupt may occur.

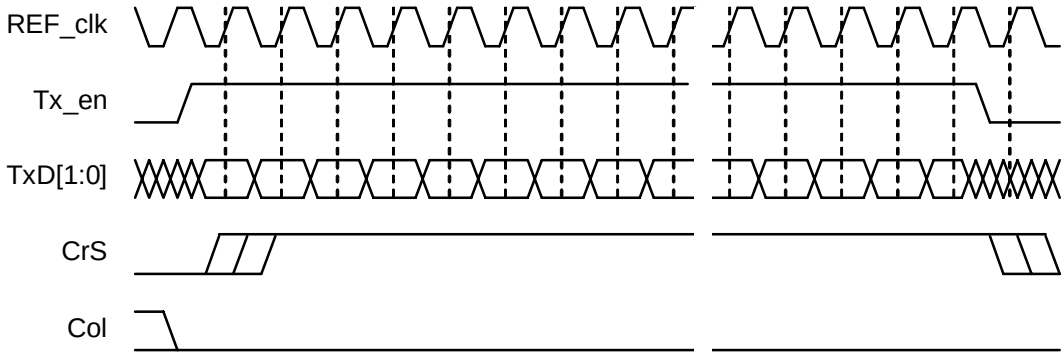


Figure 18-12 Transmission with no Collisions

Figure 18-12 shows the transmission timing of packets with RMII. Tx_en and TxD[1:0] change after the rising edge of REF_clk, then are stable until sampled at the next rising edge. When in the full duplex transfer mode, CrS and Col become undefined. When in the half duplex transfer mode, CrS must be asserted after Tx_en and must be asserted during packet transmission. If CrS is deasserted during packet transmission, a Carrier Lost error will occur. Col retaining the “L” setting indicates that no collisions have occurred.

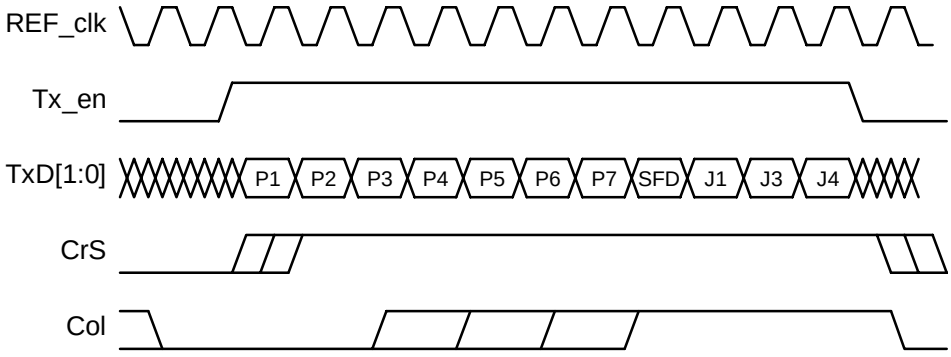


Figure 18-13 Transmission when Collision Occurred in the Preamble

Figure 18-13 shows the method the Transmission Block uses to handle collisions. It adds 4-Byte jam data before deasserting Tx_en. If a collision occurs in the preamble, the jam data is added on after SFD ends.

18.3.8.5. Frame reception procedure

To receive a frame, the Reception Enable bit (RxEn) of the Reception Control Register must be set, and the Reception Halt Request bit (RxHalt) must be cleared. Furthermore, the Halt Transmission Immediately bit (TxHalt) and the Halt Request bit (HaltReq) of the MAC Control Register must be cleared. Usually, before setting the above code, the DMA Controller is initialized by storing a valid address in the Buffer List Form Pointer Register or initializing a Free Descriptor Area Register and a Free Descriptor Size Register for example. This allows the DMA Reception Controller to control transfers from the MAC Reception FIFO. For the method of initializing data structures to enable reception, see the descriptions in subsections 18.3.9 DMA Operation and 18.3.9.5 Receiving frames.

When enabled, the MAC Reception Block always monitors the data stream coming in from RMII. When in the Loop-back mode, the data stream is inputted from the MAC Transmission Block via RMII.

The MAC Reception Block receives a 0- to 7-Byte preamble and the Start Frame Delimiter (SFD). The MAC Reception Block checks whether the first nibble received is a preamble, then checks whether there is an SFD in the first 8 Bytes of the packet. Except for the preamble, the MAC Reception Block treats a packet that does not have the SFD as its first Byte as a packet fragment and deletes it.

Following after the SFD is the first nibble of the recipient address. The MAC Reception Block generates parity when it receives 1 Byte of data, then stores it with the data in the MAC Reception FIFO. After that, it signals the existence of reception data. The MAC Reception Block collects subsequent nibbles into Byte units and stores them in the appropriate FIFO. The DMA Reception Controller reads them from the MAC Reception FIFO in Byte units, performs a parity check, and then transfers the data to the DMA Reception buffer. When either the MAC Reception FIFO becomes empty or it transmits the last Byte of a packet and ends transmission, the MAC Reception Block signals this fact.

If PHY asserts both the Rx_DV and Rx_er signals while receiving a frame, the MAC Reception Block reports that a CRC error occurred in the current packet.

When the MAC Reception Block receives the recipient address, the ARC tries to recognize the received address. If the ARC refuses a packet, the MAC Reception Block signals this fact and the DMA Reception Block destroys the data packet.

18.3.8.6. Address Recognition Circuit (ARC) operation

To read from or write to ARC memory, first the system software sets the ARC Address Register, and then it reads and writes the ARC Data Register. All Bytes are written to regardless of the Byte Enable status. To write either the upper 2 Bytes or lower 2 Bytes of a double word, the driver software has to handle the data to correctly and simultaneously write 2 adjoining Bytes. The Ethernet Controller does not support Read, Modify, or Write cycles to internal DMA RAM.

Figure 18-14 shows how the MAC reads entries from the ARC. ARC entries are in the Big Endian Byte order. In other words, if #0-0 is the first Byte of the first entry, then #0-5 is the sixth and therefore last Byte of the first entry. After entry #20 are the 2 Bytes Rsv-2 and Rsv-3, then the 2 double words MC#1 and MC#2. MC#1 and MC#2 cannot be used by ARC operation, but they are used when generating the MAC control frame. Item 18.3.8.7 Pause operation during full duplex transfer describes this.

Byte 3	Byte 2	Byte 1	Byte 0	
#0-0	#0-1	#0-2	#0-3	00h
#0-4	#0-5	#1-0	#1-1	04h
#1-2	#1-3	#1-4	#1-5	08h
#2-0	#2-1	#2-2	#2-3	0Ch
#2-4	#2-5	#3-0	#3-1	10h
#3-2	#3-3	#3-4	#3-5	14h
#4-0	#4-1	#4-2	#4-3	18h
#4-4	#4-5	#5-0	#5-1	1Ch
#5-2	#5-3	#5-4	#5-5	20h
				...
#18-0	#18-1	#18-2	#18-3	6Ch
#18-4	#18-5	#19-0	#19-1	70h
#19-2	#19-3	#19-4	#19-5	74h
#20-0	#20-1	#20-2	#20-3	78h
#20-4	#20-5	Rsv-2	Rsv-3	7Ch
MC#1-0	MC#1-1	MC#1-2	MC#1-3	80h
MC#2-0	MC#2-1	MC#2-2	MC#2-3	84h

Figure 18-14 ARM Memory Map

18.3.8.7. Pause operation during full duplex transfer

18.3.8.7.1. Local Pause operation

To enable Pause operation during full duplex transfer, you have to set a special multicast address for the MAC control frame in ARC memory and set the corresponding bit in the ARC Enable Register. You can store the special multicast address for the MAC control frame anywhere in ARC memory, but there are cases where you will have to be careful of the storage location to optimize the usage efficiency of ARC memory (see 18.3.8.7.2 Remote Pause operation). The MAC reception circuit recognizes full duplex Pause operation when the following conditions are met.

- *A particular value (0x8808) is set in the Frame Length/Frame Type field of the MAC control frame.*
- *The ARC recognizes a packet.*
- *The packet is 64 Bytes long.*
- *The Operation field specifies Pause operation (0x0001).*

After recognizing full duplex transfer Pause operation, the MAC reception circuit loads the operand values into the Pause Count Register then instructs both the MAC and DMA engines to pause when processing of the current packet ends. If no packet is currently being processed, then both engines immediately pause.

The pause circuit manages the Pause Count Register and counts down the pause time. When the pause time count becomes 0, the pause circuit issues an instruction to end pausing and operation of the transmission circuit resumes.

If another Pause operation is recognized during Pause operation, the Pause Count Register is reset by a new operand value. An operand value of 0 will abort the Pause operation currently in progress.

18.3.8.7.2. Remote Pause operation

The program can freely set the MAC control frame of the Ethernet Controller, so even when performing Pause operation or using MAC control in the future, it will still support this function.

For remote Pause operation or transmission of other MAC control frames, follow the procedure below.

- *Set the recipient address to #0 of the ARC memory.*
- *Set the sender address to #1 of the ARC memory.*
- *To #20 of the ARC memory set the special MAC control type value (0x0808), the Pause operation opcode (0x0001), and the operand value (pause time: 0x0000 to 0xFFFF). Write 0x0000 to the 2 Bytes after ARC #20.*
- *Write 0x0000_0000 to MC#1 and MC#2. Then, remote Pause transmission is complete at one bus operation.*
- *Write to the Transmission Control Register and set the SdPause bit.*

Usually, the recipient address is the special multicast address of the MAC control frame and the sender address is the local station address. You can enable these ARC entries to be used for address filtering. Since ARC entry #20 does not include a valid network address, you must not enable it when using it as part of a flow control transmission.

When transmission ends, the transmission status is written in the Transmission Control Frame Status Register. If the End of Transmission Control bit (bit 10) of the Interrupt Enable Control Register is set, the DMA engine will issue interrupts.

18.3.8.8. Error display

The Error flags and Abnormal Operation flags are divided into those that are set during transmission and those that are set during reception. These flags are each set in the Transmission Status Register (Tx_Stat) and the Reception Status Register (Rx_Stat), respectively. Also, the Missing Packet Count Register (Miss_Cnt) contains the number of deleted packets and is used to manage the system network. See 18.4

Registers for the format of these flags and the counter.

18.3.8.8.1. Transmission error display

Transmission operation ends when an entire packet (preamble, SFD, data, CRC) is transferred to a physical medium without any collisions occurring. If an internal error or network error occurs, the MAC Transmission Block reports the content of that error.

If one of the states described below occurs, transmission is aborted and the Status bit is set. After the Status bit is set, an interrupt occurs if the corresponding Interrupt Enable bit in the Transmission Control Register is set.

Table 18-5 Transmission Error Display (1/2)

Error Display	Description
MAC Transmission Parity Error	The data that the DMA Transmission Controller transmits to the MAC Transmission FIFO via DII is protected by the Parity bit. If a Parity error occurs, the DMA Transmission Controller halts transmission if the TxParErr bit of the Transmission Status Register is set and interrupts are enabled.
MAC Transmission FIFO Underrun	The MAC Transmission FIFO has a capacity of 80 Bytes so even if a collision occurs, it holds 64 Bytes for retransmission and can support DMA latency up to 1.28 μs (128-bit late time=16-Byte time). The DMA Transmission Controller has sufficient bandwidth, so if an underrun occurs in the MAC Transmission FIFO, this usually indicates that there is a problem with the latency of the PCI Bus. When such an underrun occurs, the Underrun bit of the Transmission Status Register is set.
Carrier Sense Lost	Carrier Sense (CrS) is monitored from the beginning of the Start Frame Delimiter to the last transmission Byte. This error indicates that transmission was not aborted even though CrS either does not exist or was lost due to a network fault or other cause. When in the Loop-back mode, Tx_en drives CrS. With full duplex transfers, Carrier Sense Lost is not asserted since CrS is not handed off to the Transmission Block. When Carrier Sense is lost, the LostCrS bit of the Transmission Status Register is set.
Excessive Collision	If the MAC encounters a collision during transmission, it backs off, updates the collision counter, and then tries to retransmit the data after a specific time interval elapses. When the counter becomes “16”, transmission is aborted if transmission is attempted 16 times and a collision occurred every time. A network fault could cause excessive collisions. When excessive collisions occur, the ExColl bit of the Transmission Status Register is set.
Late Collision (Collision Outside Window)	When the network is operating properly, the MAC detects a collision in the first 64 Bytes of the transmitted data. When a collision occurs after this time elapses (when one occurs outside the window), this means that there is a network fault. In this case, the LateColl bit of the Transmission Status Register is set and packet transmission is aborted. In other words, transmission is not retried after a late collision occurs.
SQE	When in the 10 Mbps mode, “heartbeat” is checked at the end of the transmitted packet. A heartbeat is a short collision signal that occurs within the first 40-bit late time period after the transmission ends. When a heartbeat is not detected, the SQErr bit of the Transmission Status Register is set.
Defer	This state indicates that it is necessary to defer transmission since the transmission route was already in use when packet transmission was attempted. This is not an error, but unless an error occurs, this state is used to indicate the state of the transmission route. When the Defer state occurs, the TxDefer bit of the Transmission Status Register is set.
Excessive Deferral	There are cases where it is necessary to defer transmission since the transmission route was already in use when the MAC attempted to transmit a packet the first time. If the deferral time is longer than MAX_DEFERRAL (2.4288 ms when in the 10 Mbps mode, 0.24288 ms when in the 100 Mbps mode) and the NoExDef bit of the Transmission Control Register is disabled, then the ExDefer bit of the Transmission Status Register is set. When the Excessive Deferral state occurs, transmission is not aborted, but a network fault may have occurred. It is possible to transmit an Excessive Deferral interrupt before packet transmission is complete and display it in the IntExDefer bit of the Interrupt Factor Register.

18.3.8.8.2. Reception error display

After detecting the Start Frame Delimiter (SFD), the MAC writes the data received from the physical medium to the MAC Reception FIFO. The MAC Reception Block checks whether an overflow occurred in the MAC Reception FIFO during reception. Also, when reception ends, it checks for external errors (Alignment, CRC, Maximum Frame Length Exceeded).

Table 18-6 Reception Error Display

Error Display	Description
MAC Reception Parity Error	Once data enters the MAC Reception FIFO, it is protected by the Parity bit. If the RxParErr bit of the Reception Status Register is set and interrupts are enabled when a Parity error occurs, the MAC Reception Block aborts reception.
Alignment Error	After reception ends, the MAC Reception Block checks whether reception packets are properly framed at the 8-bit boundaries. If they are not properly framed and CRC is invalid, the data is corrupted as it passes along the network and the MAC Reception Block signals an Alignment error. The MAC Reception Block also signals a CRC error at this time. The AlignErr bit and CRCErr bit of the Reception Status Register are set.
CRC Error	After reception ends, if the MAC Reception Block checks CRC and there was an error, it signals that error. The CRC error, Frame Alignment error, and the Maximum Frame Length Exceeded error are network errors that the Reception Block detects. These errors could be detected in the following combinations. - CRC error only - Alignment error and CRC error only - Maximum Frame Length Exceeded error and CRC error only - Alignment error, Maximum Frame Length Exceeded error, and CRC error
Overflow Error	During reception, the data is first stored in the MAC Reception FIFO, and then is transferred to the DMA Reception Controller. If the MAC Reception FIFO becomes full due to a cause such as excessive system latency, then the MAC Reception Block sets the Overflow bit of the Reception Status Register.
Maximum Frame Length Exceeded Error	The MAC Reception Block checks the packet length when reception ends. If it receives a frame that is longer than the maximum frame length of 1518 Bytes when the Long Frame mode is not enabled, the MAC Reception Block signals this error.
MII Error	When PHY detects a media error such as a coding violation, it signals MAC by asserting Rx_er. When MAC acknowledges the assertion of Rx_er, it deletes the received packet. A CRC error is forcibly issued and reception of the packet is terminated. There are also cases where an Alignment error or Minimum Frame Length error are detected.

18.3.8.9. Accessing station management data

Following is the basic sequence of events when accessing station management data.

- *The system software checks the Busy bit and confirms that MD is not Busy.*
- *For a write operation, you have to write data to the Data Register before setting up the Control Register.*
- *The software writes the MDC address, the Read or Write flag, and sets the Busy bit.*
- *The Ethernet Controller ends operation then clears the Busy bit.*
- *In the case of a read operation, the system software can read the Data Register after it detects that the Busy bit was cleared.*

18.3.9. DMA operation

This subsection describes the programming of the PCI-mounted Ethernet Controller. The programming details differ slightly depending on whether you select batch processing as the control mode or select Continuous Polling. Unless otherwise stated, the descriptions in this subsection assume the use of Continuous Polling.

Following is the structure of this subsection.

- *Initial setup of PCI*
- *Initial setup of DMA and MAC*
- *Initializing the queues*
- *Transmitting frames*
- *Receiving frames*
- *Handling interrupts*

Note: **If you use the SRAM in the DMA Controller without initializing it, Packet errors may be mistakenly detected during packet reception. Implement the following initialization procedure to prevent this phenomenon from occurring.**

(1) Prepare for initial setup
Set (write "1" to) the TestMode bit (bit 13) of the DMA_Ctl (0x00) Register.

(2) Initialize on-chip memory
Implement the following on all addresses (0x000-0x3FF) of the on-chip SRAM.
(i) Set the on-chip SRAM address in ARC_Adr (0x60). (ARC_Loc field: 0x000-0x3FF)
(ii) Write initialization data 0x0000_0000 to ARC_Data (0x064). (32-bit write)

You can implement the following to confirm the written data.
Read ARC_Data(0x64). (32-bit read)

(3) Cancel the Test mode
Reset (write "0" to) 1 in the TestMode bit (bit 13) of the DMA_Ctl (0x00) Register.

18.3.9.1. Initial setup of PCI

When initially setting up the system, you can use the IDSel signal to write to the PCI Configuration Register. This makes it possible for the system to map the Ethernet Controller to a memory boot address space then transfer data to the boot address space to output the IDSel signal.
Registers such as the following require initial setup.

- *PCI I/O Base Address Register or PCI Memory Base Address Register: To map the register to an I/O address space or memory space*
- *PCI Command Register: To customize the PCI function*
- *There are also situations where the following register requires initial setup.*
- *PCI Interrupt Register: To customize latency or signal interrupt factors to external pins*

18.3.9.2. Initial setup of DMA and MAC

After initially setting up the PCI, the DMA Control Register and MAC Control Register are usually mapped to an I/O address space or memory address space and are available for reading and writing.

Registers such as the following require initial setup.

- *DMA Transmission Frame Pointer: To start transmission*
- *DMA Buffer List Frame Pointer: To provide buffer for receiving data*
- *DMA Free Descriptor Area Base Register or DMA Free Descriptor Area Size Register: To initialize the reception signaling area*
- *DMA Transmission Polling Control Register: To customize polling of transmission packets*
- *DMA Transmission Threshold Register: To customize transmission latency handling*
- *MAC Transmission Control Register: To change the default transmission settings*
- *MAC Reception Control Register: To change the default reception settings*
- *MAC ARC Control Register: To customize the recognition conditions of station addresses and multicast addresses*
- *MAC ARC Address Register and MAC ARC Data Register: To set the filtering of station addresses and other addresses*
- *MAC ARC Enable Register: To enable each ARC entry after setting up the ARC*

There are also situations where the following registers require initial setup.

- *MAC Control Register: To customize the MAC configuration*
- *DMA Transmission Control Register: To customize the Burst size*

18.3.9.3. Initializing the queues

The system has to set up the transmission queue, buffer list, and reception descriptor area before starting up the Ethernet Controller.

18.3.9.3.1. Initializing the transmission queue

The Transmission Block has two operation modes: Batch Processing and Continuous Polling. When in the Batch Processing mode, the system software prepares a link list of the frame descriptors to be transmitted. The final descriptor must have "1" set in the EOL (end of list) field. When the last frame descriptor is transmitted, the Transmission Frame Pointer Register reads EOL and transmission ends. To resume transmission after this, the system resets the Transmission Frame Pointer Register.

When in the Continuous Polling mode, the system software also prepares a link list of the frame descriptors to be transmitted. However, the last frame descriptor is a dummy frame descriptor. The beginning of a link list is only a dummy frame descriptor, so it is okay for it to be empty. The system owns the dummy descriptor to prevent the Ethernet Controller from accessing it. Transmitting a new packet overwrites the dummy frame descriptor. This is explained in 18.4.4 Flow Control Register group.

18.3.9.3.2. Initializing the buffer list

The buffer list queue is either a single frame descriptor or multiple frame descriptors combined into a link list containing a list of the empty buffer descriptors that is initialized. It is okay for the buffer list to have one of the following configurations.

1. One frame descriptor containing many free buffer descriptors
2. A link list of frame descriptors
3. Wrap-around queue in which the last frame descriptor points to the first frame descriptor

In configurations 1 and 2 above, the EOL bit of the FDNext field is set while in configuration 3, the FDNext field of the last frame descriptor points to the first frame descriptor. You can use the Reception Buffer Fragment Size Register to globally pack a buffer. Also, you can use the Control field (FDCtl) of a frame descriptor to select whether to pack in buffer area units. Setting up the Buffer ID field when packing a buffer is useful in managing memory. For details on packing buffers,

see 18.4.3.6 Reception Fragment Size Register, 18.4.3.8 Free Descriptor Area (FDA) Register, and 18.3.7.2.1 BDCtl field (buffer descriptor control).

18.3.9.3.3. Initializing a reception descriptor area

You can initialize a reception descriptor by writing in the Free Descriptor Area Base Register or the Free Descriptor Area Size Register. The Ethernet Controller starts writing to the reception queue in the reception descriptor area according to these registers.

18.3.9.4. Transmitting frames

Paragraph 18.3.9.3.1 Initializing the transmission queue described transmission in a batch process. For each frame transmission batch, the system initializes the transmission queue and sets a Transmission Frame Pointer Register at the head of the queue.

In the case of continuous polling transmission, the frame descriptor list ends at the dummy frame descriptor owned by the system. The Ethernet Controller enters the Polling mode when it reaches a dummy record. In this mode, the Ethernet Controller periodically checks the frame descriptor Control (FDCtl) field and waits for the system to set the COwnsFD bit. The Transmission Polling Counter Register controls the polling frequency.

To transmit frames in the Continuous Polling mode, the system writes the frame descriptor of the frame to be transmitted at the end of the transmission queue. The system overwrites the old dummy frame descriptor, creates a new dummy frame descriptor, and sets the FDNext field of the old frame descriptor in a new dummy frame descriptor. After that, it sets the COwnsFD bit of the old frame descriptor and transfers ownership to the Ethernet Controller.

18.3.9.4.1. Signaling transmission completion

The system can fetch transmission completion information in various ways.

- *Request an interrupt.*
- *Poll the FDCtl field of the transmitted frame descriptor and confirm system ownership.*
- *Poll the Transmission Frame Pointer Register.*

You can set interrupts to occur either at the end of each frame or at the end of selected frames. When polling the Transmission Frame Pointer Register, it has an invalid value (a value set in FDNext while EOL=1) in the Batch Processing mode or has a dummy frame descriptor address when in the Continuous Polling mode.

18.3.9.5. Receiving frames

For the MAC to receive a frame, the system software has to perform the following operations.

- *As described in 18.3.9.3 Initializing the queues, initialize a free buffer list or free descriptor area.*
- *Write a dummy frame descriptor in a free descriptor area. Set the COwnsFD bit of the FDCtl field so the Ethernet Controller becomes the owner.*
- *Initialize the Reception Frame Pointer Register to the address of the dummy frame descriptor in the free descriptor area.*

There are two ways for the system software to get notification of a reception frame:

- *Request an interrupt for each reception frame.*
- *Poll the dummy frame descriptor and check whether the COwnsFD bit is set.*

You can enable interrupts by setting the Reception Complete Interrupt Enable bit of the Reception Control Register.

The system must perform the following processes after it receives a frame.

- *Process the frame descriptor, and then release it for reuse.*
- *Release the buffer returned from the above protocol layer and add it to the free buffer list.*

18.3.9.5.1. Processing reception frame descriptors

Free descriptor areas are used by FIFOs. However, the frame processing and time required for returning the corresponding buffer differ depending on the application. Then, the frame descriptor mapped by the Ethernet Controller copies that content to a different area, the frame descriptor of the free descriptor area is released in the order received, and the copied descriptor contents are handed to the upper layer of the protocol stack.

18.3.9.5.2. Releasing buffers

There are two methods of mapping a buffer:

- *Start a new frame in a new buffer (Single Frame mode).*
- *Place several frames or parts of a frame in one buffer (Packed Buffer mode).*

The Control field of the Buffer Fragment Size Register or Frame Descriptor controls the method of mapping the buffer. The Single Frame mode has the advantage of allowing easy memory management, but has the disadvantage of decreasing the memory usage efficiency. Conversely, the Packed Buffer mode has the advantage when considering memory usage efficiency, but has the disadvantage since memory management becomes complex.
When in the Packed Buffer mode, multiple frames or parts of frames may be placed in the same buffer area, so you must keep several points in mind when managing the memory. The Ethernet Controller counts the number of buffers created in the same buffer area then provides that value as the RxBDSeqN value of the BDCtl field in the buffer descriptor. Then, the system software counts the returned fragments, confirms that all fragments were returned, then releases them.
The buffer ID value (RxBDID) is copied from the buffer descriptor of the free buffer list queue to the buffer descriptor in the reception frame queue. You can use up to 256 buffer IDs. When you require more buffer IDs than that, there are several techniques you can use. For example, you can specify the buffer ID expansion bit in the FDSystem field. You can also use the upper bit of the buffer pointer.

18.3.9.6. Handling interrupts

Interrupts generally use one common interrupt line. To confirm whether this PCI device is the interrupt source, use the system software to read the Interrupt Factor Register. Depending on the content of the Interrupt Factor Register, you may have to read other registers such as the Transmission Status Register or Reception Status Register.

18.4. Registers

18.4.1. Overview

This subsection describes the Ethernet Controller registers you can access. These registers are grouped as follows below.

- *PCI Configuration Register Group*
- *DMA Control, Status Register Group*
- *Flow Control Register Group*
- *MAC Control, Status Register Group*

During normal operation, once you finish setting them up, few registers require direct access. Transmission/Reception operation is performed using a cyclic queue with a ring-like structure.

Control information and status information is transmitted by the data structures described in 18.3.7 Configuration. You have to initialize the DMA Control Registers before starting any transfer operation.

You have to access the MAC Registers when it requires a special configuration such as address filtering by the ARC. When using an interrupt drive type, you have to access several of the DMA Registers or MAC Registers from inside the Interrupt Handler, enable or disable interrupts, check the interrupt factors, or clear the Interrupt Condition bit.

Also, the Flow Control Register can use a driver to access them in order to monitor the run status of Pause commands issued from a local terminal or remote terminal.

18.4.1.1. Accessing registers

Except for those that use part of RAM, the reserved bits of a register are initialized to "0" or "1". To maintain compatibility even if the method of using the registers changes in the future, do not change the values of the reserved bits when using the software to write to registers.
Also, do not implement programming that depends on the values of the reserved bits.

18.4.1.2. Overview of PCI Configuration Registers

Table 18-7 lists the name, mnemonic, address, size, and access type of each PCI Configuration Register. PCI Configuration Registers are used by standard Master/Slave PCI devices.

Included in PCI Configuration Registers are Device ID Registers, Control Registers, registers that display status information, and registers that make various settings. These registers are setting during initialization.

Table 18-7 PCI Configuration Registers

Address	Mnemonic	Register Name
00h	Vend_ID	Vendor ID Register
02h	Dev_ID	Device ID Register
04h	PCI_Cmd	PCI Command Register
06h	PCI_Stat	PCI Status Register
08h	PCI_Clas	Class Code Register
0Ch	PCI_Ctl	PCI Control Register
10h	IO_BaseA	I/O Base Address Register
14h	MLo_BaseA	Memory Base Address Register
2Ch	Sub_Vend_ID	Subsystem Vendor ID Register
2Eh	Sub_ID	Subsystem ID Register
34h	PCI_Cap_Ptr	PCI Function Pointer
3Ch	PCI_Int	PCI Interrupt Register
40h	PM_Cap	Power Management Function
44h	PM_CSR	Power Management Control Status

The address of a PCI Configuration Register is valid when the input signal IDSel is asserted. The addresses of DMA Registers and MAC Registers are valid when IDSel is not asserted and the upper bits of the addresses match either the I/O Base Address Register (IO_BaseA) or the Memory Base Address Register (MLo_BaseA).

18.4.1.3. Overview of DMA Control, Status Registers

Table 18-8 shows the name, mnemonic, address, size, and access type of each DMA Control, Status Register. DMA Control, Status Registers control the transmission queue, reception queue, free buffer list, and free descriptor area. Registers for controlling the fragment size and polling rate are also available.

Table 18-8 DMA Control, Status Registers

Address	Mnemonic	Register Name
00h	DMA_Ctl	DMA Control Register
04h	TxFrmPtr	Transmission Frame Pointer
08h	TxThrsh	Transmission Threshold Register
0Ch	TxPollCtl	Transmission Polling Control Register
10h	BLFrmPtr	Buffer List Frame Pointer
14h	RxFragSize	Reception Fragment Size Register
18h	Int_En	Interrupt Enable Register
1Ch	FDA_Bas	Free Descriptor Area Base Register
20h	FDA_Lim	Free Descriptor Area Size Register
24h	Int_Src	Interrupt Function Register

18.4.1.4. Overview of MAC Control, Status Registers

MAC layer registers are broken down into the Flow Control Register and MAC Control, Status Register groups. Table 18-9 shows the name, mnemonic, address, size, and access type of each Flow Control Register.

Table 18-9 Flow Control Registers

Address	Mnemonic	Register Name
30h	PauseCnt	Pause Count Register
34h	RemPauCnt	Remote Pause Count Register
38h	TxConFrmStat	Transmission Control Frame Status Register

Table 18-10 shows the name, mnemonic, address, size, and access type of each MAC Control, Status Register. Similar to the ARC Control Register and Error Count Register, some registers are accessed by the system software driver when the MAC is activated. A standard DMA engine controls the MAC Transmission Control Register, MAC Transmission Status Register, MAC Reception Control Register, and MAC Reception Status Register after the system software driver sets them up.

Table 18-10 MAC Control, Status Register

Address	Mnemonic	Register Name
40h	MAC_Ctl	MAC Control Register
44h	ARC_Ctl	ARC Control Register
48h	Tx_Ctl	Transmission Control Register
4Ch	Tx_Stat	Transmission Status Register
50h	Rx_Ctl	Reception Control Register
54h	Rx_Stat	Reception Status Register
58h	MD_Data	Station Management Data Register
5Ch	MA_CA	Station Management Control, Address Register
60h	ARC_Adr	ARC Address Register
64h	ARC_Data	ARC Data Register
68h	ARC_Ena	ARC Enable Register
7Ch	Miss_Cnt	Missed Error Count Register

MAC layer Control Registers include registers such as the Master MAC Control Register, Transmission/ Reception Control Register, and the ARC Control Register.

18.4.1.5. Register address maps

Figures 18-15 through 18-18 show the memory addresses of the PCI Configuration Registers, DMA Control, Status Registers, Flow Control Registers, and MAC Control, Status Registers. System memory addresses are listed in the Little Endian mode.

	Byte 3	Byte 2	Byte 1	Byte 0	
	Dev_ID		Vend_ID		00
	PCI_Stat		PCI_Cmd		04
PCI_Clas	Base_Cl	Sub_Cl	Prog_IF	Rev_ID	08
PCI_Ctl	BIST	Hdr_Typ	Lat_Timr	Cache_Sz	0C
	IO_BaseA				10
	MLo_BaseA				14
	Reserved (MHi_BaseA)				18
	Reserved				...
	Sub_ID		Sub_Vend_ID		2C
	Reserved				30
	Reserved			Cap_Ptr	34
	Reserved				38
PCI_Int	Max_Lat	Min_Gnt	Int_Pin	Int_Line	3C
	PM_Cap				40
	PM_CSR				44

Figure 18-15 Address Map of PCI Configuration Registers

Byte 3	Byte 2	Byte 1	Byte 0	
Reserved	DMA_Ctl			00
TxFrmPtr				04
Reserved		TxThrsh		08
Reserved		TxPollCtl		0C
BLFrmPtr				10
Reserved		RxFragSize		14
Reserved		Int_En		18
FDA_Bas				1C
Reserved		FDA_Lim		20
Reserved	Int_Src			24

Figure 18-16 Address Map of DMA Control, Status Registers

Byte 3	Byte 2	Byte 1	Byte 0	
Reserved		PauseCnt		30h
Reserved		RemPauCnt		34h
Reserved	TxConFrmStat			38h

Figure 18-17 Address Map of Flow Control Registers

Reserved	MAC_Ctl		40
Reserved	Reserved	ARC_Ctl	44
Reserved	Tx_Ctl		48
Reserved	Tx_Stat		4C
Reserved	Rx_Ctl		50
Rx_Stat			54
Reserved	MD_Data		58
Reserved	MD_CA		5C
Reserved	ARC_Adr		60
ARC_Data[3:0]			64
Reserved	ARC_Ena[2:0]		68
Reserved	Reserved		6C
Reserved	Reserved		70
Reserved			...
Reserved	Miss_Cnt		7C

Figure 18-18 Address Map of MAC Control, Status Registers

18.4.2. PCI Configuration Register group

Each PCI device has a configuration address space totaling 256 Bytes. The PCI specification prescribes the first 72 Bytes of this space in advance. An Ethernet Controller that has the wakeup service only uses the first 74-Byte address space.

18.4.2.1. Vendor ID Register (Vend_ID) 0x00

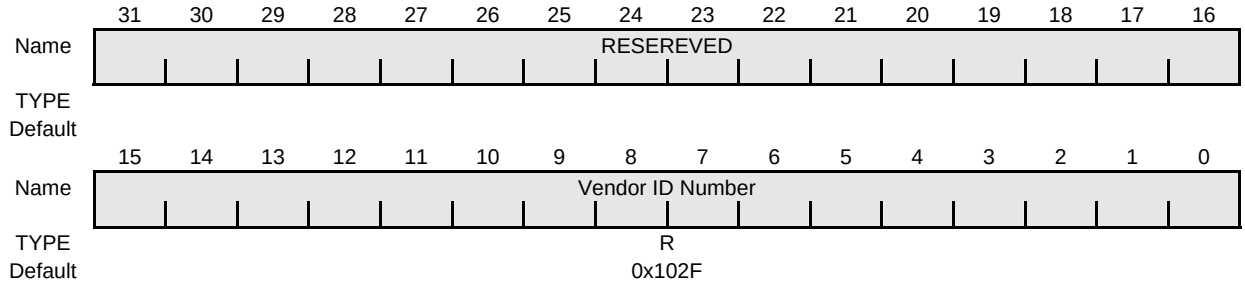


Figure 18-19 Vendor ID Register

The Vendor ID Register stores the ID code PCI SIG, which denotes Toshiba as the component manufacturer. This field is set to 0x102F.

18.4.2.2. Device ID Register (Dev_ID) 0x02

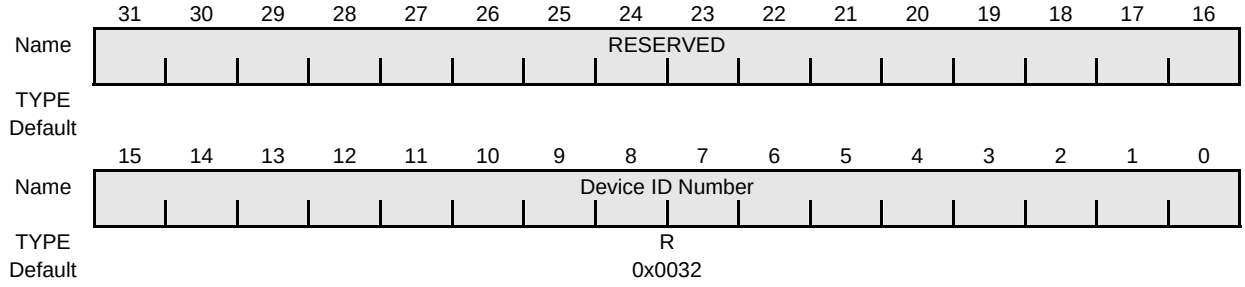
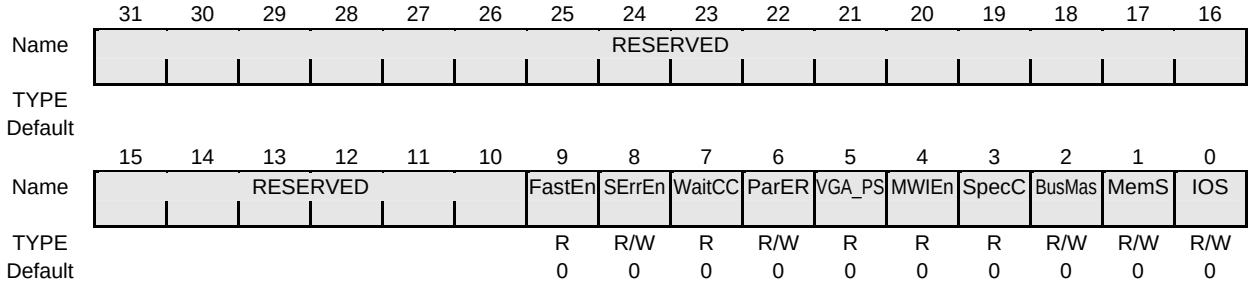


Figure 18-20 Device ID Register

The Device ID Register stores a 16-bit value for identifying a particular device. The device ID number Toshiba allocated is 0x0032. The Class Code Register stores the device's revision ID.

18.4.2.3. PCI Command Register (PCI_Cmd) 0x04



Bit(s)	Mnemonic	Field Name	Description
15 : 10		Reserved	
9	FastEn	Fast Back-to-back Enable	FastEn (fixed to "0", R) The Ethernet Controller cannot perform two consecutive bus transactions.
8	SErrEn	System Error Enable	SerrEn (Default: 0, R/W) Enables the system error (SERR#) driver.
7	WaitCC	Wait Cycle Control	WaitCC (fixed to "0", R) The Ethernet Controller does not generate address/data stepping.
6	ParER	Parity Error Response	ParER (Default: 0, R/W) The device responds to a parity error.
5	VGA_PS	VGA Palette Snoop	VGA_PS (fixed to "0", R) Does not perform special VGA Palette Snoop.
4	MWIEn	Memory Write and Invalidate Enable	MWIEn (Default: 0, R) The Ethernet Controller does not issue the Memory Write and Invalidate command.
3	SpecC	Special Cycle	SpecC (fixed to "0", R) The Ethernet Controller ignores special cycles.
2	BusMas	Bus Master	BusMas (Default: 0, R/W) The device can operate as a Bus Master.
1	MemS	Memory Space	MemS (Default: 0, R/W) The device responds to memory access.
0	IOS	I/O Space	IOS (Default: 0, R/W) The device responds accesses to the I/O space.

Figure 18-21 PCI Command Register

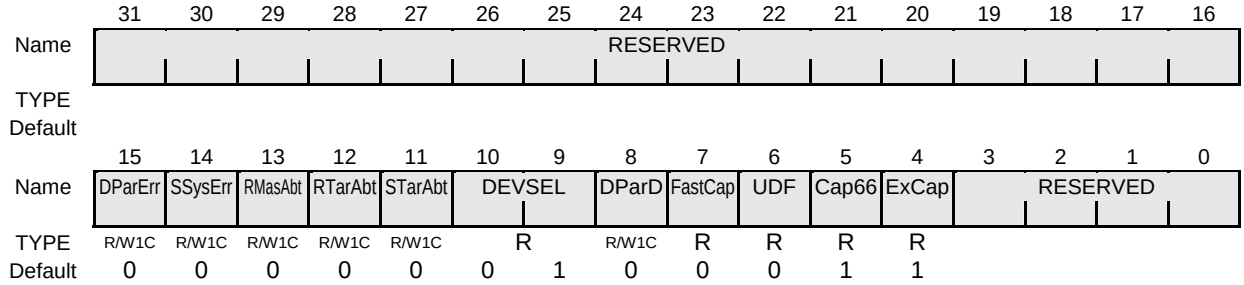
Hardware reset initializes the PCI Command Register to 0x0000. Software reset does not initialize the register contents.

The PCI Command Register defines the method of generating and responding to PCI cycles. For details, see Subsection 6.2.2 Device Control in the PCI 2.2 specification.
For the Ethernet Controller to function properly, you have to set the BusMas bit and MemS bit or IOS bit.

Set the SErrEn bit and the ParER bit to match the system.
The FastEn, WaitCC, VGA_PS, MWIEn and SpecC bits are always "0". The Ethernet Controller ignores any writes to these bits.

The Ethernet Controller has only one address comparator. The address comparator shares memory access and I/O access as described below. If MemS=0 and IOS=1, the Ethernet Controller responds to any accesses made to the I/O space. If MemS=0 and IOS=0, the Ethernet Controller does not respond to any accesses made to the I/O space or to the memory space. If MemS=1, then the Ethernet Controller only responds to accesses made to the memory space regardless of the IOS value. Also, you have to initialize the corresponding base address registers in addition to MemS and IOS.

18.4.2.4. PCI Status Register (PCI_Stat) 0x06



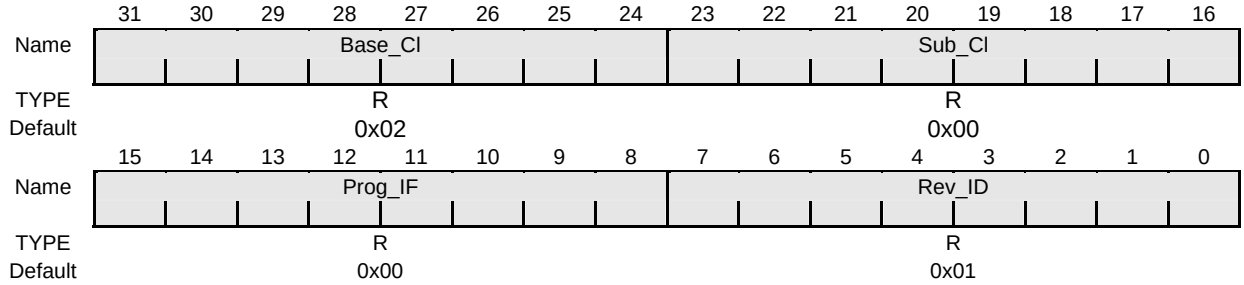
Bit(s)	Mnemonic	Field Name	Description
15	DParErr	Detected Parity Error	DParErr (Default: 0, R/W) Indicates that a parity error was detected. This bit is set if a parity error is detected even when the register's Parity Error Response bit is not set. This also applies to parity errors occurring during an address cycle.
14	SSysErr	Signal System Error	SSysErr (Default: 0, R/W) This bit is set when the device asserts SERR#.
13	RMasAbt	Receive Master Abort	RMasAbt (Default: 0, R/W) Indicates that the current device is the Bus Master and that Master Abort ended the Bus Master transaction (excluding special cycles).
12	RTarAbt	Receive Target Abort	RTarAbt (Default: 0, R/W) Indicates that the current device is the Bus Master and that Target Abort ended the Bus Master transaction.
11	STarAbt	Signal Target Abort	STarAbt (Default: 0, R/W) Indicates the current device is the target and that Target Abort ended the Bus Master transaction.
10 : 9	DEVSEL	Device Select Timing	DEVSEL (Fixed to "01", R) The latest output timing of DEVSEL# when the Ethernet Controller is accessed as the target device is "medium speed". This field is encoded as follows. 00: Fast 01: Medium Speed 10: Slow
8	DParD	Detect Data Parity Error	DParD (Default: 0, R/W) This bit is set when all of the three following conditions are met. PERR# was asserted as the Bus Master, but the target asserted it. The agent that set PERR# was the Bus Master. The Parity Error Response bit of the PCI Control Register is set to "1".
7	FastCap	Fast Back-to-back Capable	FastCap (fixed to "0", R) The Ethernet Controller cannot perform consecutive transactions from different agents.
6	UDF	User Defined Function	UDF (fixed to "0", R) Indicates that there are no user-defined functions in the Ethernet Controller.
5	Cap66	66 MHz Operation Capable	Cap66 (fixed to "1", R) Indicates that the Ethernet Controller can operate at 33-66 MHz.
4	ExCap	Extended Function	ExCap (fixed to "1", R) Indicates that the Ethernet Controller has extended functions.
3 : 0		Reserved	

Figure 18-22 PCI Status Register

Software reset initializes the PCI Status Register to 0x0230.

The PCI Status Register retains status information on events relating to the PCI Bus. There are several points about this register that you should keep in mind. First, writing "1" to a bit in this register clears it to "0". Also, the value of each bit does not change when you write "0" to it. This enables the system and PCI device to simultaneously update the status information. For details, see Subsection 6.2.3 Device Status of the PCI 2.2 specification.

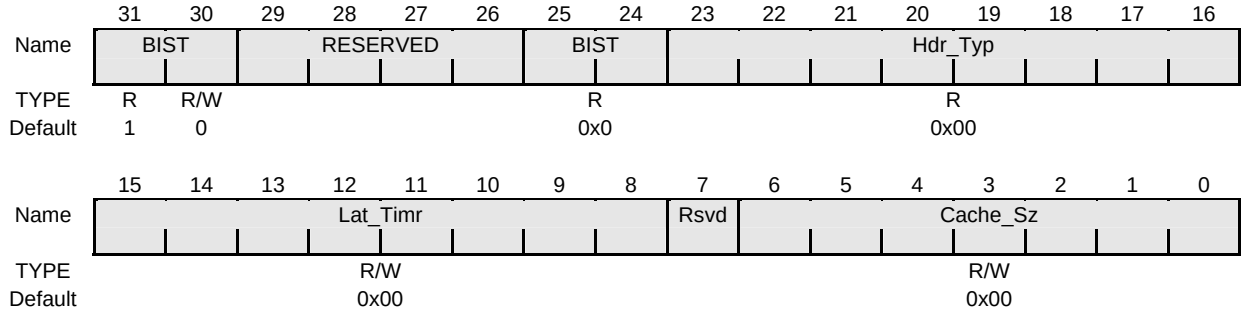
18.4.2.5. Class Code Register (PCI_Clas) 0x08



Bits	Mnemonic	Field Name	Description
31 : 24	Base_Cl	Base Class ID	Base_Cl (fixed to "0x02", R)
23 : 16	Sub_Cl	Subclass ID	Sub_Cl (fixed to "0x00", R) Is defined in the Ethernet Controller.
15 : 8	Prog_IF	Programming Interface	Prog_IF (fixed to "0x00", R) The register-level programming interface is not defined.
7 : 0	Rev_ID	Revision ID	Rev_ID (fixed to "0x01", R) Is set to "0x01" for this Ethernet Controller.

Figure 18-23 Class Code Register

18.4.2.6. PCI Control Register (PCI_Ctl) 0x0C



Bits	Mnemonic	Field Name	Description
31 : 24	BIST	Embedded Self-Test	BISAT (Default: 0x80, R/W) Controls whether to invoke BIST during startup.
23 : 16	Hdr_Typ	Header Type	Hdr_Typ (fixed to "0x00", R) This is a single-function device. The range 0x10-0x3F of the configuration space is the standard layout.
15 : 8	Lat_Timr	Latency Timer	Lat_Tmr (DefaultA: 0x00, R/W) Sets the time the Ethernet Controller operates as the Bus Master as a PCI Bus Clock count.
7		Reserved	
6 : 0	Cache_Sz	Cache Line Size	Cache_Sz (Default: 0x00, R/W) Sets the system cache line size.

Figure 18-24 PCI Control Register

Hardware reset initializes the PCI Control Register to 0x8000_0000. Software reset does not change the register contents.

BIST is used for testing the buffers on the chip. Bit 31 of the BIST field is read only and is fixed to "1". In other words, the Ethernet Controller supports BIST. Bit 30 is used to start invoking BIST. Writing "1" to bit 30 starts a test. This bit is cleared when a test ends. If an error occurs in a test, then either bit 25 or bit 24 is set. Bit 25 is set if a parity error occurs when starting up SRAM inside DMA. Bit 24 indicates that there is a fault in the SRAM in DMA. In other words, it indicates that the read data and the expected value (read data) do not match. Bits 29:26 are reserved.

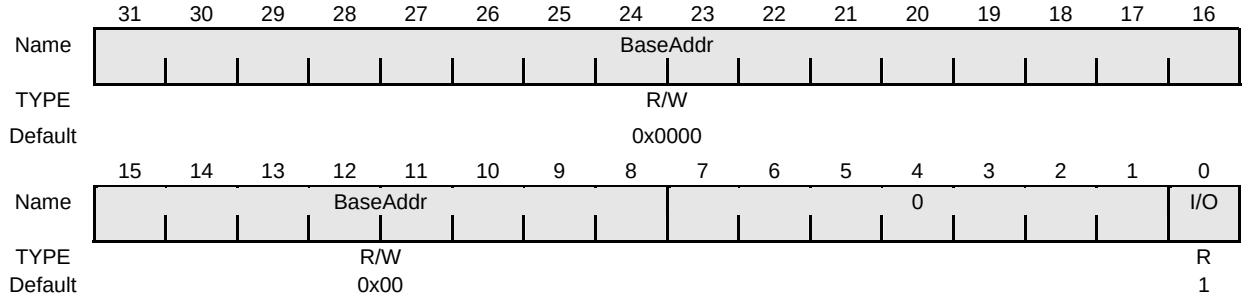
The software driver invokes BIST during initialization. (Note: Invoking BIST affects operation since RAM data and RAM registers are overwritten.)
When the clock is 33 MHz, testing 1 K × 4-Byte memory requires approximately 123 μs.

The default value of the latency timer is 0. You can use a program to set the latency timer.
After performing a hardware reset, the cache line size is initialized to 0. You must use the software driver to set an appropriate default value. In most cases, the recommended value is 8 double words (32 Bytes). The maximum value of the cache line size is 127 double words. The cache line size is used to select the Memory Read Multiple command or Memory Read Line command and perform Burst reads.

18.4.2.7. I/O, Memory Base Address Registers

I/O Base Address Register (IO_BaseA) 0x10

The I/O Base Address Register and Memory Base Address Register are used to map the DMA and MAC Control Registers or Status Registers to the I/O address space or system memory space. Both the I/O address space and memory address space have a maximum size of 32 bits. In addition to setting the Base Address Register, you have to set the corresponding Control bits in the PCI Command Register.

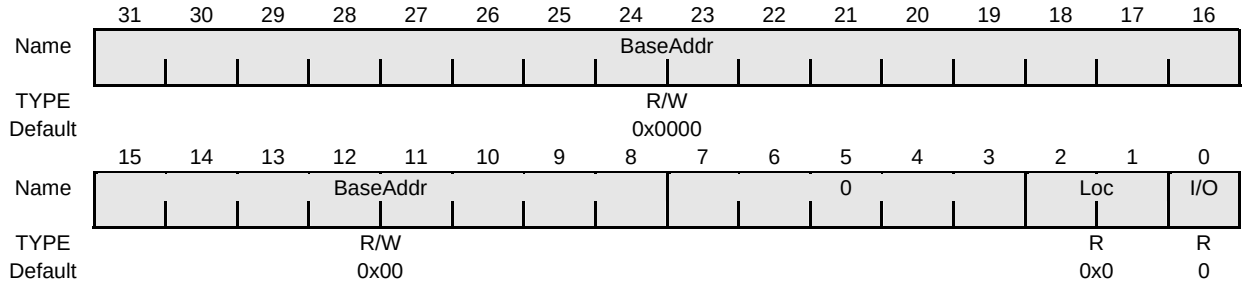


Bit(s)	Mnemonic	Field Name	Description
31 : 8	BaseAddr	Base Address	BaseAddr (Default: 0x000000, R/W) This field sets the upper 24 bits of the base address
7 : 1			Fixed to “0”.
0	I/O	I/O Flag	I/O (fixed to “1”, R) Indicates that this base address is relative to the I/O space.

Figure 18-25 I/O Base Address Register

Hardware resets initialize the I/O Base Address Register to 0x0000_0001. Software resets do not change the register contents.

Memory Address Register (MLo_BaseA) 0x14



Bit(s)	Mnemonic	Field Name	Description
31 : 8	BaseAddr	Base Address	BaseAddr (Default: 0x00_0000, R/W) This field sets the upper 24 bits of the base address.
7 : 3			Fixed to “0”.
2 : 1	Loc	Location Bit	Loc (Default: 00, R) This field indicates that memory is placed in a 32-bit address space.
0	I/O	I/O Flag	Memory (Default: 0, R) Indicates that this base address is relative to the memory space.

Figure 18-26 Memory Base Address Register

Hardware resets initialize the Memory Base Address Register to 0x0000_0000. Software resets do not change the register contents.

18.4.2.8. Subsystem Vendor ID Register, Subsystem ID Register

Subsystem Vendor ID Number Register (Sub_Vend_ID) 0x2C

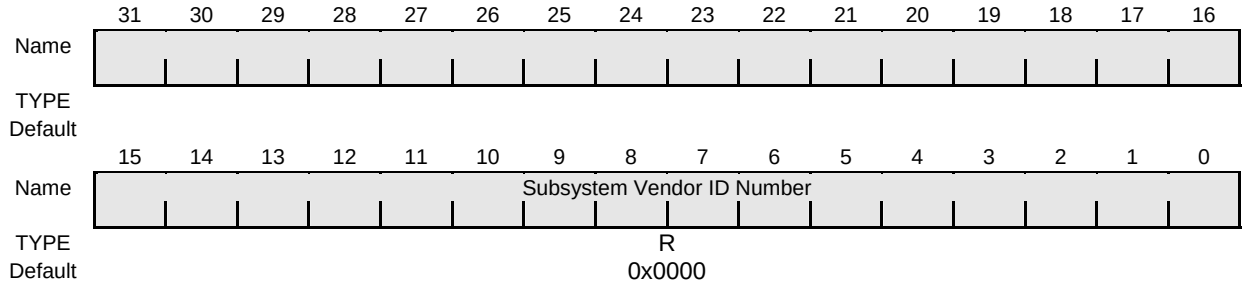


Figure 18-27 Subsystem Vendor ID Number Register

The Subsystem Vendor ID Register is set to 0x0000.

Subsystem ID Register (Sub_ID) 0x2E

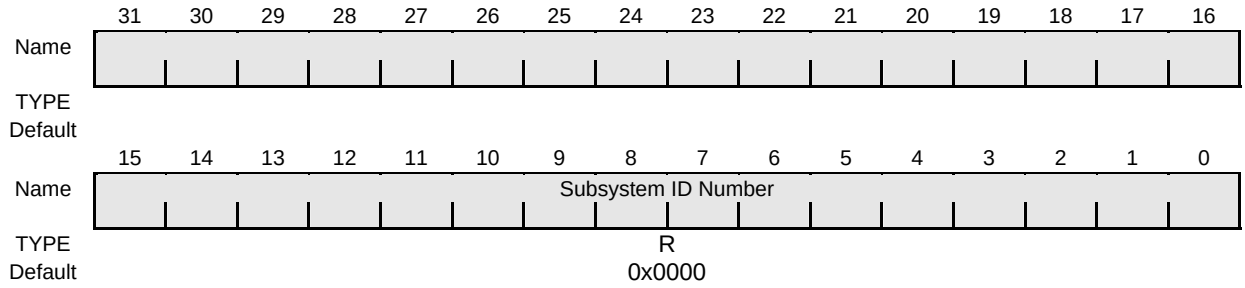


Figure 18-28 Subsystem ID Register

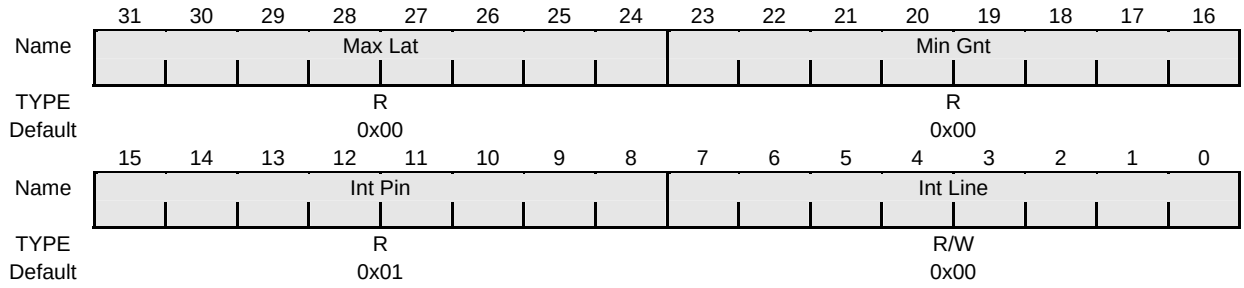
The Subsystem ID Register stores a 16-bit value for identifying a particular device. In the case of the TX4939, this value is 0x0000.

18.4.2.9. PCI Function Pointer Register (PCI_Cap_Ptr) 0x34

Bits	Mnemonic	Field Name	Description
31 : 8		Reserved	
7 : 0	CapPtr	Function Pointer	CapPtr (Fixed to “0x40”, R) This field is the offset of the first entry in the function list.

Figure 18-29 PCI Function Pointer Register

18.4.2.10. PCI Interrupt Register (PCI_Int) 0x3C



Bits	Mnemonic	Field Name	Description
31 : 24	Max_Lat	Maximum Latency	Max_Lat (Fixed to “0x00”, R) Sets the latency timer value (unit: 1/4 μs).
23 : 16	Min_Gnt	Minimum Grant	Min_Gnt (Fixed to “0x00”, R) Assuming the PCI clock is 33 MHz, this field sets the time required for Burst transfer (unit: 1/4 μs).
15 : 8	Int_Pin	Interrupt Pin	Int_Pin (Fixed to “0x01”, R) Is set to INTA#.
7 : 0	Int_Line	Function Pointer	Int_Line (Default: 0x00, R/W) Is set by the system. Is the connection information of the interrupt line.

Figure 18-30 PCI Interrupt Register

When the value of Max_Lat and Min_Gnt is 0, you do not use the PCI Interrupt Register to determine Maximum Latency and Minimum Grant.
To check whether this PCI device caused an interrupt, use the software to access the Interrupt Source Register. For details, see 18.4.3.9 Interrupt Source Register.

18.4.2.11. PCI Power Management Registers

Power Management Function Register (PM_Cap) 0x40

Bits	Mnemonic	Field Name	Description
31 : 0		Reserved	Reserved The Ethernet Controller does not support the Power Management Function Register.

Figure 18-31 Power Management Function Register

Power Management Control, Status Register (PM_CSR) 0x44

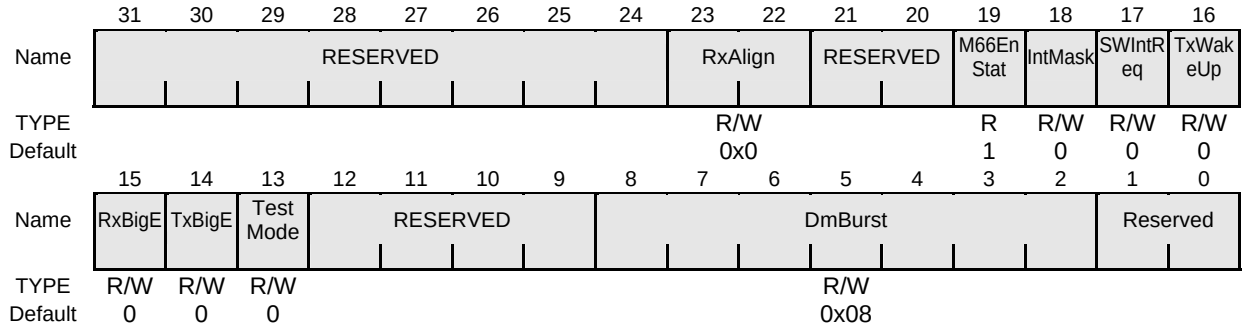
Bits	Mnemonic	Field Name	Description
31 : 0		Reserved	Reserved The PCI Controller does not support the Power Management Control, Status Register.

Figure 18-32 Power Management Control, Status Register

18.4.3. DMA Control, Status Register group

The DMA engine and system software jointly manage three queues: the transmission queue, the reception queue, and the buffer list. The transmission queue is used for frame descriptors that are ready for transmission and are standing by. The reception queue is used for frame descriptors that have been received and are waiting for processing by the system software. The buffer list is a buffer descriptor queue. Buffer descriptors describe system memory areas that can be used to store reception data. The free descriptor area (FDA) is a memory area that the Ethernet Controller can write transmission queue frame descriptors and buffer list descriptors to.

18.4.3.1. DMA Control Register (DMA_Ctl) 0x00



Bit(s)	Mnemonic	Field Name	Description
31 : 24		Reserved	
23 : 22	RxAlign	Reception Alignment	D2Supp (Default: 0x0, R/W) This field controls the alignment of reception packets as follows: 01: Skips the first byte of the first buffer. 10: Skips the first or second byte of the first buffer. 11: Skips the first through third byte of the first buffer.
21 : 20		Reserved	
19	M66EnStat	66 MHz Enable State	M66EnStat (Fixed to 1, R) 1: Indicates that the PCI Bus operates at a maximum of 66 MHz. 0: Indicates that the PCI Bus operates at 33 MHz or less.
18	IntMask	Interrupt Mask	IntMask (Default: 0, R/W) Setting this bit to “1” disables the interrupt request signal.
17	SWIntReq	Software Interrupt Request	SWIntReq (Default: 0, R/W) Setting this bit to “1” issues an interrupt request.
16	TxWakeUp	Transmission Wake Up	TxWakeUp (Default 0, R/W) Setting this bit to “1” ends the current polling cycle and starts transmission.
15		Reserved	
14		Reserved	
13	TestMode	Test Mode	TestMode (Default: 0, R/W) Enables the test mode function.
12 : 9		Reserved	
8 : 2	DmBurst	DMA Burst Size	DmBurst (Default: 0x08, R/W) Indicates the Burst size of data transfer executed in the Master mode.
1 : 0		Reserved	

Figure 18-33 DMA Control Register

Hardware resets initialize the DMA Control Register as follows:

- *0000_1020h:* *When the PCI Bus speed is 0-33 MHz*
- *0008_1020h:* *When the PCI Bus speed is 33-66 MHz*

0x20 of the lower byte denotes the default (32 Bytes = 8 double words) of the DMBurst field.

The DMA Control Register controls data transfer functions in the Master mode such as the Burst side, Big Endian handling, and the test mode. This register also controls various DMA functions during transmission such as wake up and software interrupts.

The DmBurst field controls the data transfer size with which the PCI Bus was used when operating in the Master mode. Bits 8:0 set the data transfer size (DMA Burst size), but the lower 2 bits are fixed to “0” and must be a multiple of 4. After a hardware reset, the default value becomes 32 Bytes, in other words 8 double words. You can change this value using a software driver. You cannot set the DmBurst field to “0”. Writing of “0” to this field is ignored. Generally, a multiple of the PCI cache line size is set in the DmBurst field. If the Burst size is 4, 8 or 12 when in the 100 Mbps full duplex mode, you have to take into consideration throughput reduction of the PCI Bus.

The TestMode bit enables test functions such as those that enable reading or writing to all areas of the internal DMA buffer or those that display internal status information in the reserved bits of a register.

The TxBigE bit and RxBigE bit enable the transmitting or receiving of data with Big Endian mode devices. However, it is important to note that only the data (bytes in the area designated by the buffer descriptor) is handled in the Big Endian mode. Control information such as frame descriptors and buffer descriptors are always in a format unique to the PCI Bus. In other words, they are in the Little Endian format.

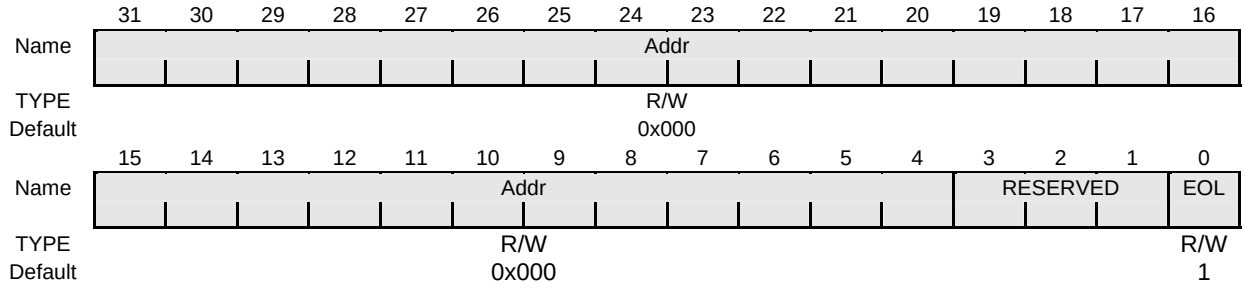
The TxWakeUp bit enables immediate data transmission without waiting for the end of the current polling cycle. Setting the TxWakeUp bit to “1” while the transmitter is polling aborts the current polling cycle. When the current polling cycle ends, the current TxWakeUp bit is cleared. Any writing of “0” to the TxWakeUp bit is ignored.

Software interrupts are made available to support software drivers.

The IntMask bit disables all interrupt sources. Therefore, even when the processing of interrupt sources is in progress, the software driver can put interrupts in the Enable state again.

The M66EnStat bit is used to control the MAC clock divide circuit when driving serial addresses of the MII station manager.

18.4.3.2. Transmission Frame Pointer Register 0x04



Bit(s)	Mnemonic	Field Name	Description
31 : 4		Address	Addr (Default: 0x000_0000, R/W) Retains the address of the first frame descriptor to be transmitted.
3 : 1		Reserved	
0	EOL	End of List	EOL (Default: 1, R/W) When this bit is set to “1”, the Address field is ignored. The Ethernet Controller waits for the system to clear it.

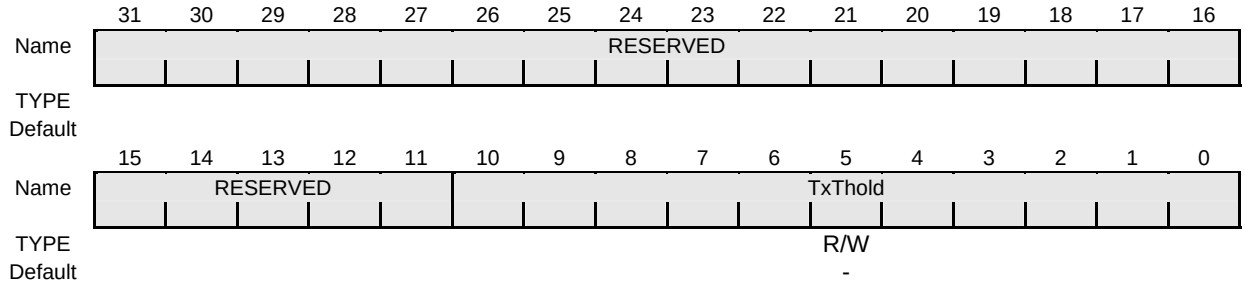
Figure 18-34 Transmission Frame Pointer Register

Software resets initialize the Transmission Frame Pointer Register to 0x0000_0001.

Software resets set the EOL bit to “1”. To enable polling of a transmission or the packet to be transmitted, the system has to set this register in a properly initialized frame descriptor. The address has to be aligned to a 16-Byte boundary. Therefore, bits 0-3 must be “0”.

For information on polling control methods, see the description in 18.4.3.4 Transmission Polling Control Register.

18.4.3.3. Transmission Threshold Register (TxThrsh) 0x08



Bits	Mnemonic	Field Name	Description
31 : 11		Reserved	
10 : 0	TxThold	Transmission Threshold Register	TxThold (Default: -, R/W) The Transmission Threshold Register controls buffer latency when transmitting packets.

Figure 18-35 Transmission Threshold Register

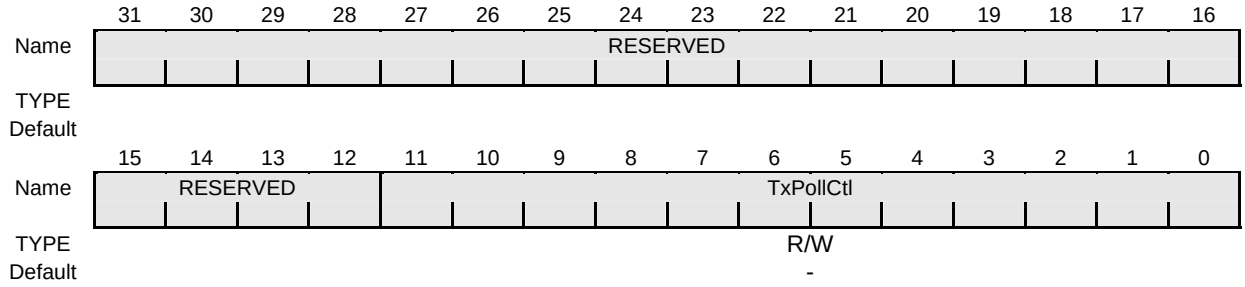
The Transmission Threshold Register uses part of internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Threshold Register controls buffer latency when transmitting packets. If the threshold value is not “0”, data transfer to the MAC starts either when the number of data bytes set by the threshold value (TxThold) has accumulated in the DMA Transmission buffer or when an entire packet is stored in the DMA Transmission buffer. When the threshold value is “0”, data is transferred to the MAC immediately after the data is fetched from the PCI Bus. The software driver initializes the Transmission Threshold Register.

When the threshold value (TxThold) is too small, the DMA Transmission buffer immediately becomes free due to the PCI Bus latency. This situation is displayed in the MAC transmission status, so the system software increases the threshold value.

Be careful to not make the threshold value greater than 1620. Having a threshold value greater than 1620 when in the Long Packet mode causes buffer memory to become full before transmission is enabled, therefore causing the transmission circuit to hang.

18.4.3.4. Transmission Polling Control Register (TxPollCtl) 0x0C



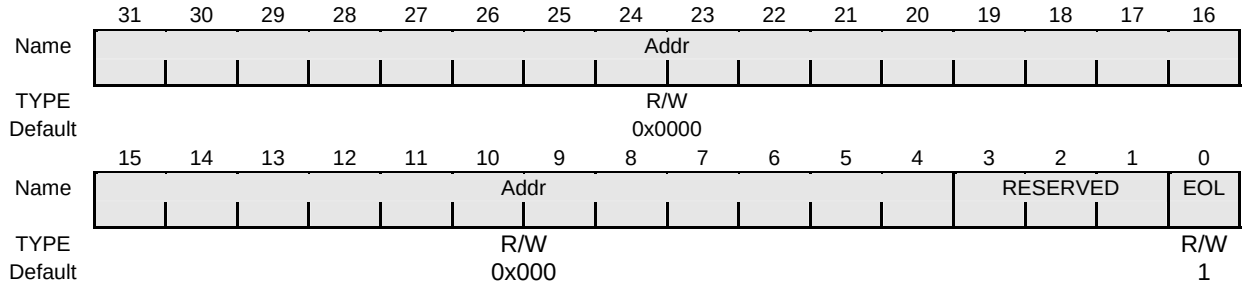
Bits	Mnemonic	Field Name	Description
31 : 12		Reserved	
11 : 0	TxPollCtl	Transmission Polling Control Register	TxPollCtl (Default: -, R/W) The Transmission Polling Control Register controls the frequency at which packets to be transmitted are polled.

Figure 18-36 Transmission Polling Control Register

The Transmission Polling Control Register uses part of internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Polling Control Register controls the frequency at which packets to be transmitted are polled. The TxPollCtl field operates as a counter. When a value is set to it, it decrements until it reaches 0. When it reaches 0, it polls to see if preparations for the next transmission packet are complete. No polling is performed when 0 is set in the TxPollCtl field. When the clock frequency is 33 MHz, the counter units equate to 61.44 μs. The software driver initializes this register. The software driver also sets the count value.

18.4.3.5. Buffer List Frame Pointer (BLFrmPtr) 0x10



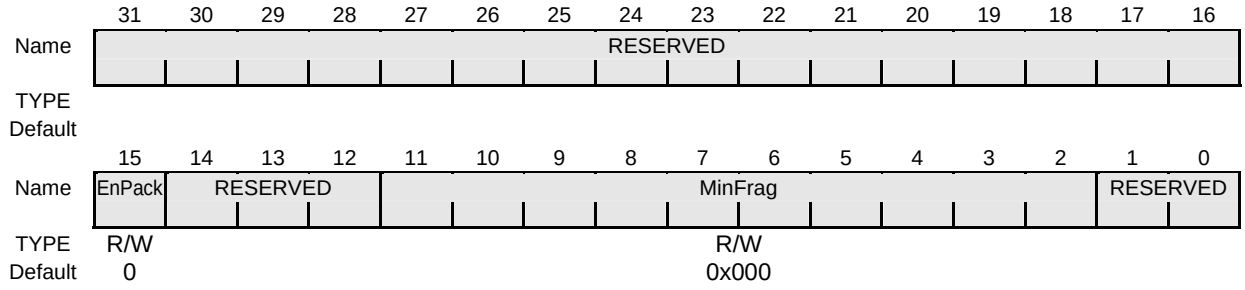
Bit(s)	Mnemonic	Field Name	Description
31 : 4		Address	Addr (Default: 0x000_0000, R/W)
4 : 1		Reserved	
0	EOL	End of List	EOL (Default: 1, R/W) When this bit is set to “1”, the Address field is ignored. You have to wait until the system clears this bit.

Figure 18-37 Buffer List Frame Pointer

Software resets initialize the Buffer List Frame Pointer to 0x0000_0001.

When it fetches a free buffer descriptor, the Buffer List Frame Pointer holds the address of the first frame descriptor to be read. To enable data reception, the system has to set this register in a properly initialized frame descriptor. The address must be aligned to a 16-Byte boundary. In other words, bits 0-3 must be “0”.

18.4.3.6. Reception Fragment Size Register (RxFragSize) 0x14



Bit(s)	Mnemonic	Field Name	Description
31 : 16		Reserved	
15	EnPack	Enable Packing	EnPack (Default: 0, R/W) 1: Uses the MinFrag value for buffer packing control. 0: Uses the FDCtl field of the frame descriptor for buffer packing control.
14 : 12		Reserved	
11 : 2	MinFrag	Minimum Fragments	MinFrag (Default: 0x000, R/W) The minimum byte count when partially writing to a buffer that contains data.
1 : 0		Reserved	

Figure 18-38 Reception Fragment Size Register

Hardware resets initialize the Reception Fragment Size Register to 0x0000_0000. Software resets do not change the register contents.

The Reception Fragment Size Register specifies the size of minimum data fragments that the Ethernet Controller generates. The minimum fragment size must be a multiple of 4. Therefore, the lower 2 bits are always “0”. You can use the EnPack bit to globally enable packing, or you can enable packing in buffer-area units. For more information on the enabling of packing in buffer-area units, see the description in 18.3.7.1.5 FDCtl field (frame descriptor control).

The Ethernet Controller always stores reception data in addresses aligned to 4-Byte boundaries. Therefore, the last 1-3 bytes of a frame may be unused.

When enabling packing, the MinFrag value must be greater than 0 for the Ethernet Controller to function properly. When using packing, use a software driver to set the MinFrag field and EnPack bit.

When not enabling packing, the MinFrag value must remain at “0”.

Packing is performed when more than the Minimum Fragment Size + 4 Bytes in the buffer is free. Packing is not performed when the amount of free space in the buffer is less than the Minimum Fragment Size. In this case, data is stored starting from the next buffer.

18.4.3.7. Interrupt Enable Register (Int_En) 0x18

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED															
TYPE																
Default																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED				NRAbtEn	TxCtlCmpEn	DmParErrEn	DParDEn	EarNotEn	DParErrEn	SSvsErrEn	RmasAbtEn	RTargAbtEn	STargAbtEn	BLExEn	FDAExEn
TYPE					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Field Name	Description
31 : 12		Reserved	
11	NRAbtEn	Non-recoverable Abort Enable	NRAbtEn (Default: 0, R/W) Enables interrupts when a non-recoverable abort occurs internally.
10	TxCtlCmpEn	MAC Control Frame Transmission Complete Enable	TxCtlCmpEn (Default: 0, R/W) Enables interrupts when transmission of the MAC control frame is complete.
9	DmParErrEn	DMA Parity Error Enable	DmParErrEn (Default: 0, R/W) Enables interrupts if a parity error is detected when reading or writing from/to DMA-internal RAM.
8	DParDEn	Data Parity Detection Enable	DParDEn (Default: 0, R/W) Enables interrupts when bit 8 of the PCI Status Register is set.
7	EarNotEn	Early Notification Enable	EarNotEn (Default: 0, R/W) When receiving a reception packet, issues an interrupt not only when reaching the tail of a packet, but also issues an interrupt when writing the first buffer or its descriptor.
6	DParErrEn	Parity Error Detection Enable	DParErrEn (Default: 0, R/W) Enables interrupts if a parity error is detected during PCI Bus transfer while the Ethernet Controller is accessing the Bus Master.
5	SSvsErrEn	System Error Notification Enable	SSvsErrEn (Default: 0, R/W) Enables interrupts if the Ethernet Controller signals a system error.
4	RmasAbtEn	Master Abort Reception Enable	RmasAbtEn (Default: 0, R/W) Enables interrupts if a Master Abort is received while the Ethernet Controller is operating as the Target.
3	RTargAbtEn	Target Abort Reception Enable	RTargAbtEn (Default: 0, R/W) Enables interrupts if a Target Abort is received while the Ethernet Controller is operating as the Target.
2	STargAbtEn	Target Abort Notification Enable	STargAbtEn (Default: 0, R/W) Enables interrupts if a Target Abort is issued while the Ethernet Controller is operating as the Target.
1	BLExEn	Buffer List Exhaustion Notification Enable	BLExEn (Default: 0, R/W) Enables interrupts when the buffer list is completely used up. In other words, enables interrupts when the Ethernet Controller encounters descriptors owned by the system that still remain in the buffer list.
0	FDAExEn	Free Descriptor Area Exhaust Notification Enable	FDAExEn (Default: 0, R/W) Enables interrupts when the free descriptor area is totally used up. In other words, enables interrupts when the Ethernet Controller encounters blocks owned by the system that still remain in the FDA.

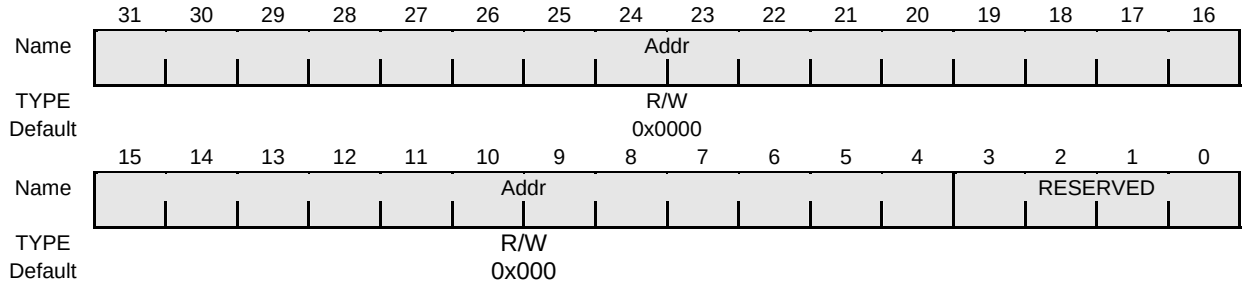
Figure 18-39 Interrupt Enable Register

Hardware resets initialize the Interrupt Enable Register to 0x0000_0000. Software resets do not change the register contents.
The Interrupt Enable Register controls whether to issue an interrupt in response to errors detected by the DMA engine or in response to some other conditions.

The Early Notification Enable (EarNotEn) bit is shared by applications that are required to reduce latency. When handling Early Notification, note that the frame descriptor becomes invalid. Only the first buffer descriptor is valid if an Early Notification interrupt occurs.

18.4.3.8. Free Descriptor Area (FDA) Registers

Free Descriptor Area Base Register (FDA_Bas) 0x1C



Bits	Mnemonic	Field Name	Description
31 : 4		Address	Addr (Default: 0x000_0000, R/W) The Free Descriptor Area Base Register contains the start address of the area for writing the frame descriptors and buffer descriptors of reception packets. The address must be a multiple of 16 Bytes. Therefore, bits 0-3 are fixed to "0".
3 : 0		Reserved	

Figure 18-40 Free Descriptor Area Base Register

Hardware resets initialize the Free Descriptor Area Base Register to 0x0000_0000. Software resets do not change the register contents.

Free Descriptor Area Size Register (FDA_Lim) 0x20

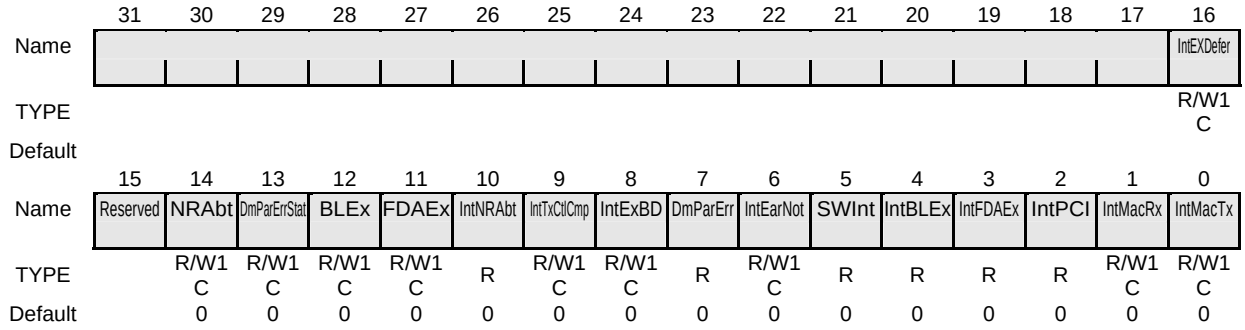
Bits	Mnemonic	Field Name	Description
31 : 16		Reserved	
15 : 4	Count/Offset	Count/Offset	Count/Offset (Default: 0x000, R/W) This field sets the size of the Reception Descriptor Area in 16-Byte units. You could also consider the lower 16 bits as being an offset from the base address.
3 : 0		Reserved	

Figure 18-41 Free Descriptor Area Size Register

Hardware resets initialize the Free Descriptor Area Size Register to 0x0000_0000. Software resets do not change the register contents.

Note: You have to specify in the FDA_Lim Register the minimum offset value of the free descriptor area from which it is safe to start the next frame descriptor. You have to secure an area that is sufficient for storing a maximum size packet that includes one frame descriptor and the maximum number of buffer descriptors. For example, if the maximum number of buffer descriptors required to store a maximum length frame is 28, then this area requires a capacity of 256 (16 + 28 × 8 in 16-Byte units) Bytes.

18.4.3.9. Interrupt Source Register (Int_Src) 0x24



Bit(s)	Mnemonic	Field Name	Description
31 : 17		Reserved	
16	IntEXDefer		IntEXDefer (Default: 0, R/W1C) This bit is set if an excessive delay is detected and the EnExDefer (Excessive Defer Enable) bit of the Transmission Control Register (Tx_Ctl) is set.
15		Reserved	
14	NRAbt		NRAbt (Default: 0, R/W1C) This bit is set to "1" when a non-recoverable abort occurs.
13	DmParErrStat		DmParErrStat (Default: 0, R/W1C) This bit is set to "1" when a DMA parity error occurs.
12	BLEx		BLEx (Default: 0, R/W1C) This bit is set to "1' when the buffer list (BL) becomes full.
11	FDAEx		FDAEx (Default: 0, R/W1C) This bit is set to "1" when the free descriptor area (FDA) becomes full.
10	IntNRAbt		IntNRAbt (Default: 0, R) The interrupt source is a non-recoverable abort state.
9	IntTxCtlCmp		IntTxCtlCmp (Default: 0, R/W1C) The interrupt source is the completion of MAC control frame transmission.
8	IntExBD		IntExBD (Default: 0, R/W1C) The interrupt source is excessive buffer descriptors (more than 28 buffer descriptors).
7	DmParErr		DmParErr (Default: 0, R) The interrupt source is a DMA parity error.
6	IntEarNot		IntEarNot (Default: 0, R/W1C) The interrupt source is early notification.
5	SWInt		SWInt (Default: 0, R) The interrupt source is a software interrupt request.
4	IntBLEx		IntBLEx (Source: 0, R) The interrupt source is the buffer list being completely used up.
3	IntFDAEx		IntFDAEx (Default: 0, R) The interrupt source is the free descriptor area being completely used up.
2	IntPCI		IntPCI (Default: 0, R) The PCI Status Register (PCI_Stat) displays the interrupt source.
1	IntMacRx		IntMacRx (Default: 0, R/W1C) The MAC Reception Status Register (Rx_Stat) displays the interrupt source.
0	IntMacTx		IntMacTx (Default: 0, R/W1C) The MAC Transmission Status Register (Tx_Stat) displays the interrupt source.

Figure 18-42 Interrupt Source Register

Writing "1" to W1Clr clears it to "0". This bit denotes the bits that clear interrupts. Any writing of "0" to these bits is ignored. "R" indicates that a bit is Read Only. Either clearing the source that set these bits to "1" or resetting Ethernet control clears these bits to "0".

Software resets initialize the Interrupt Source Register to 0x0000_0000.
The system software reads the Interrupt Source Register to check whether there are any interrupts related to Ethernet control. Also, the Interrupt Source Register contains several status bits that can only be displayed here. If bits 15:0 are all

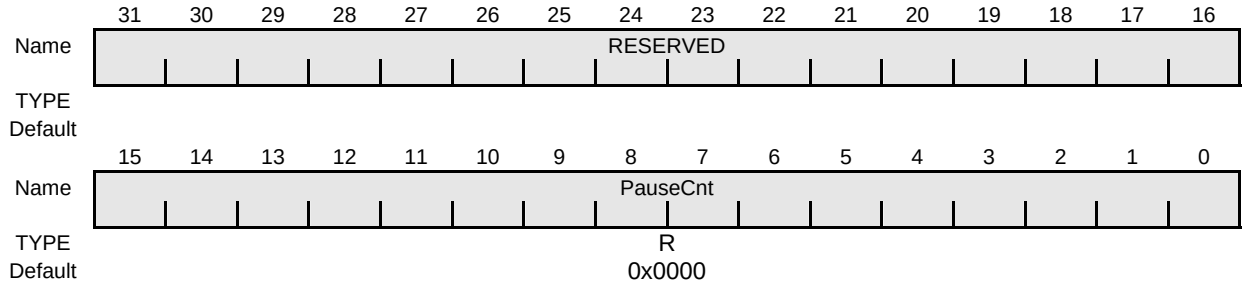
"0", this indicates that the Ethernet Controller did not issue any interrupts. If the Ethernet Controller causes any interrupts, then setting the IntMask bit of the DMA Control Register makes it possible to mask any subsequent interrupts that the Ethernet Controller issues.

Bit 8 is set when there are more than 28 buffer descriptors are in a single frame descriptor. If excessive buffer descriptors occur, the controller aborts transmission or reception operation. To resume transmission or reception operation, you have to perform a software reset and rebuild the transmission queue, reception queue, and buffer list. Bit 13 is set when a parity error is detected in the DMA RAM. However, bit 7 is only set if the DParErrEn bit of the Interrupt Enable Register is set.

To make it possible to handle as many completed transmission frames or reception frames as possible, the software driver has to clear the IntMACTx bit or the IntMACRx bit. Since the frame that is the interrupt source is handled as a part of the immediately previous interrupt, there are cases where there are no complete frames in an interrupt. Take this into account when creating your software driver.

18.4.4. Flow Control Register group

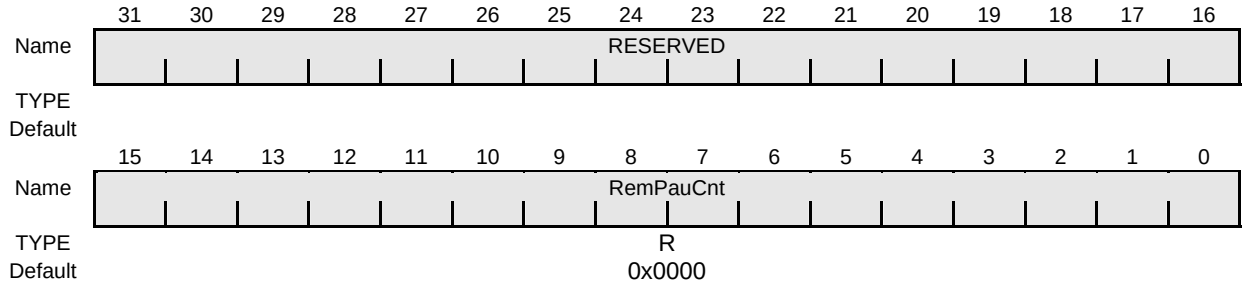
Pause Count Register (PauseCnt) 0x30



Bits	Mnemonic	Field Name	Description
31 : 16		Reserved	
15 : 0	PauseCnt	Reception Pause Count	PauseCnt (Default: 0, R) This is the number of time slots that the transmitter pauses as a result of receiving a MAC control Pause operation packet.

Figure 18-43 Pause Count Register

Remote Pause Count Register (RemPauCnt) 0x34



Bits	Mnemonic	Field Name	Description
31 : 16		Reserved	
15 : 0	RemPauCnt	Remote Pause Count	RemPauCnt (Default: 0, R) This is the number of time slots that the remote MAC pauses as a result of transmitting a Pause operation packet.

Figure 18-44 Remote Pause Count Register

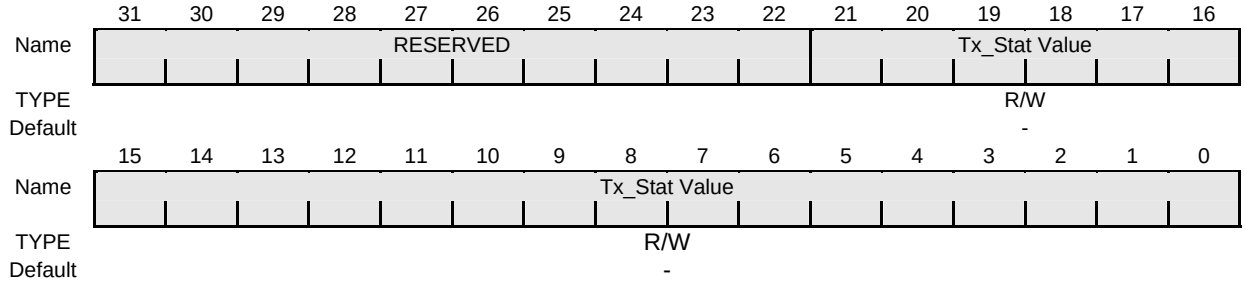
Software resets initialize the Pause Count Register and Remote Pause Count Register to 0x0000_0000.

The Pause Count Register displays the current value of the received Pause count. When a Pause operation MAC control frame is received, the specified Pause count is set and the count is decremented at each unit time stamp.

When the count is "0", it indicates that MAC is not pausing. The Remote Pause Count Register displays a rough value of the pause counter that the remote station has based on the transmitted Pause command.

In either case, the unit is 1 slot time. In other words, it is 512-bit late time.

Transmission Control Frame Status Register (TxCtlFrmStat) 0x38



Bits	Mnemonic	Field Name	Description
31 : 22		Reserved	
21 : 0	Tx_Stat Value	Transmission Control Frame Status	Tx_Stat (Default: -, R/W) The Transmission Control Frame Status Register displays via the SdPause bit of the Transmission Control Register the status from when the MAC control frame was transmitted to a remote station.

Figure 18-45 Transmission Control Frame Status Register

The Transmission Control Frame Status Register uses part of the internal RAM, so it is not affected by hardware resets or software resets.

The Transmission Control Frame Status Register displays via the SdPause bit of the Transmission Control Register the status when the MAC control frame was transmitted to a remote station.

After transmission ends, the software driver fetches the status from this register. The software can reset this register before starting transmission of the MAC control frame. The software can use the TxCtlCmpEn bit of the Interrupt Enable Register to issue an interrupt when transmission of the MAC control frame ends.

For information on the bit fields, see the explanation of the Transmission Status Register (Tx_Stat) in 18.4.5.3 Transmission Control, Status Registers.

18.4.5. MAC Control, Status Register group

18.4.5.1. MAC Control Register (MAC_Ctl) 0x40

Bit(s)	Mnemonic	Field Name	Description
31 : 14		Reserved	
13	EnMissRoll	Missing Error Counter Rollover Enable	EnMissRoll (Default: 0, R/W) Issues an interrupt when the count value of the Missing Error Count Register rolls over from 0x7FFF to 0x8000.
12:11		Reserved	
10	MissRoll	Missing Error Counter Rollover	MissRoll (Default: 0, R) Indicates that the count value of the Missing Error Count Register rolled over from 0x7FFF to 0x8000. (Read only)
9:7		Reserved	
6 : 5	Conn	Connection Mode	Conn (Default: 00, R/W) This field selects the connection mode. 00: Automatic (default) 01: Reserved 10: MII (MII clock determines the transfer rate) 11: Reserved
4	MacLoop	MAC Loop Back	MacLoop (Default: 0, R/W) Directly provides the transmission signal as the input of the reception circuit without sending it outside the Ethernet Controller.
3	FullDup	Full Duplex Mode	FullDup (Default: 0, R/W) Set this bit to "1" for full duplex.
2	Reset	Software Reset	Reset (Default: 0, R/W) Resets all State Machines and FIFOs of the Ethernet Controller.
1	HaltImm	Immediate Halt	HaltImm (Default: 0, R/W) Immediately halts transmission or reception when set to “1”. When receiving data, if this bit is set after MAC starts processing the recipient address, reception operation for the current packet continues and the data is transferred to system memory. At this time, if the RxHalted bit of the Reception Status Register is set, it indicates that the system sent a Reception Halt Request while a packet was being received. If this bit is set before starting processing of the recipient address, reception operation is immediately halted. The RxHalted bit will then be set.
0	HaltReq	Halt Request	HaltReq (Default: 0, R/W) Halts transmission/reception when the packet currently in progress ends.

Figure 18-46 MAC Control Register

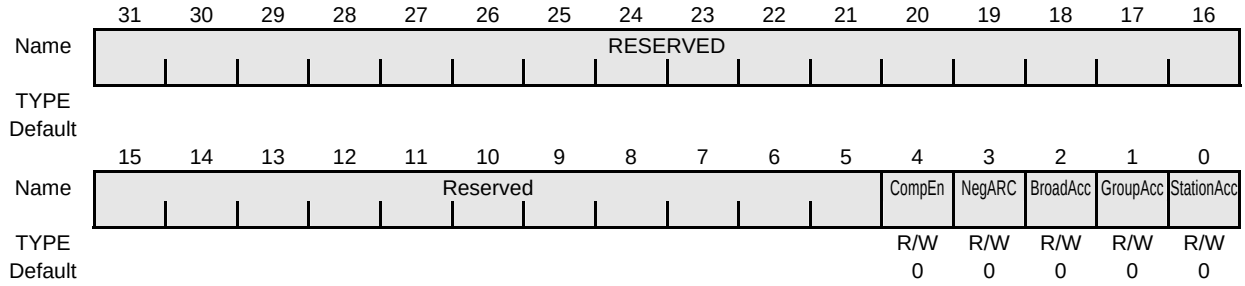
Hardware resets initialize the MAC Control Register to 0x8000. Setting the Reset bit (bit 2) executes software reset. Starting software reset clears bit 2. The other bits do not affect software resets.

The MAC Control Register is used to display total control and status information of MAC. The MissRoll bit is the status bit. All other bits are control bits.

After the Reset bit is set, 4 MII transmission or reception clock cycles pass, then software reset is executed for several cycles. Therefore, after writing to the Reset bit, do not access the Ethernet Controller until 320 ns pass for 100 Mbps transfer or until 3,200 ns pass for 10 Mbps transfer. Before performing reset, you can use the MAC Transmission Control Register (Tx_Ctl) or MAC Reception Control Register (Rx_Ctl) to issue a Halt Request to end the current network transaction.

The MissRoll bit is set when the counter rolls over from 0x7FFF to 0x8000, and is reset when the software reads the Missing Error Count Register. See 18.4.5.7 Missing Error Count Register for an explanation.
Some PHYs do not support full duplex transfer. The MacLoop bit has higher priority than the FullDup bit.

18.4.5.2. ARC Control Register 0x44



Bit(s)	Mnemonic	Field Name	Description
31 : 5		Reserved	
4	CompEn	Compare Enable	CompEn (Default: 0, R/W) Enables the Compare mode.
3	NegARC	Negative ARC	NegARC (Default: 0, R/W) 0: Accepts packets the ARC recognized, but refuses all other packets. 1: Refuses packets the ARC recognized, but accepts all other packets.
2	BroadAcc	Broadcast Accept	BroadAcc (Default: 0, R/W) Accepts all packets that have a broadcast address.
1	GroupAcc	Group Accept	GroupAcc (Default: 0, R/W) Accepts all packets that have a multicast group address.
0	StationAcc	Station Accept	StationAcc (Default: 0, R/W) Accepts all packets that have a unicast station address.

Figure 18-47 ARC Control Register

Hardware resets initialize the ARC Control Register to 0x0000. Software resets do not change the register contents. The ARC recognizes the three following Ethernet address types.

- *Station addresses: The first byte is even such as in 00-00-00-00-00-00.*
- *Broadcast addresses: Defined as FF-FF-FF-FF-FF-FF.*
- *Multicast group address: The first byte is odd such as in 01-00-00-00-00-00. However, not FF-FF-FF-FF-FF-FF.*

When the CompEn bit is set and the ARC's Compare mode is validated, the recipient address of the reception packet is compared with the address stored in ARC memory. For information on the structure of ARC memory, see 18.3.8.6 Address Recognition Circuit (ARC) operation.

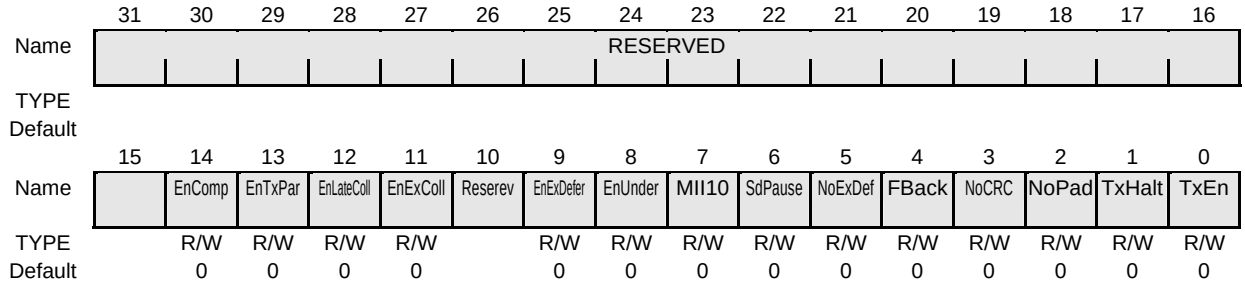
When the CompEn bit is cleared, the ARC unconditionally halts address comparisons. When an Accept bit (StationAcc, GroupAcc, or BroadAcc) is set, packets the ARC refused are also accepted. To refuse all packets, clear all bits of the ARC Control Register. To put MAC in the Promiscuous mode and accept all normal packets, set the ARC to accept all three of the above address types. You can also put MAC in the Promiscuous mode by setting the Negative ARC bit and clearing the CompEn bit.

When the ARC Compare mode is enabled, addresses for filtering reception messages are read from ARC memory. ARC memory has a 6-Byte structure for each entry.

You can set it to Valid or Invalid for each entry as described in ARC Enable Register on page 86.

18.4.5.3. Transmission Control, Status Registers

Transmission Control Register (Tx_Ctl) 0x48



Bit(s)	Mnemonic	Field Name	Description
31 : 15		Reserved	
14	EnComp	Enable Complete	EnComp (Default: 0, R/W) Issues an interrupt either when MAC transmits or destroys 1 packet.
13	EnTxPar	Enable Transmission Parity	EnTxPar (Default: 0, R/W) Issues an interrupt when a parity error occurs in MAC Transmission FIFO.
12	EnLateColl	Enable Late Collisions	EnLateColl (Default: 0, R/W) Issues an interrupt if a collision occurs after more than 512-bit late time (64-Byte) time passes.
11	EnExColl	Enable Excessive Collisions	EnExColl (Default: 0, R/W) Issues an interrupt if collisions occur in the same packet 16 times.
10		Reserved	
9	EnExDefer	Enable Excessive Defer	EnExDefer (Default: 0, R/W) Issues interrupts when MAC causes a MAX_DEFERRAL time delay. MAX_DEFERRAL = 0.24288 ms for 100 Mbps = 2.4288 ms for 10 Mbps
8	EnUnder	Underline Enable	EnUnder (Default: 0, R/W) Issues an interrupt when the MAC Transmission FIFO becomes empty during transmission.
7	MII10	MII 10 Mbps Mode	MII10 (Default: 0, R/W) Setting this bit to "1" enables SQE checking.
6	SdPause	PAUSE Transmission	SdPause (Default: 0, R/W) Transmits either the Pause command or another MAC control frame.
5	NoExDef	No Excessive Deferrals	NoExDef (Default: 0, R/W) Suppresses excessive deferral checking.
4	FBack	Fast Back Off	Fback (Default: 0, R/W) Uses a fast back-off timer during testing.
3	NoCRC	No Suppress	NoCRC (Default: 0, R/W) Does not add CRC to the end of a packet.
2	NoPad	No Padding	NoPad (Default: 0, R/W) Does not generate Pad Bytes even for packets with less than 64 Bytes.
1	TxHalt	Transmission Halt Request	TxHalt (Default: 0, R/W) Halts transmission if the current packet ends regardless of the packet type.
0	TxEn	Transmission Enable	TxEn (Default: 0, R/W) Immediately halts transmission when cleared to "0".

Figure 18-48 Transmission Control Register

Hardware resets initialize the Transmission Control Register to 0x0000. Software resets clear the TxEn bit, but do not clear any other bits. Do not set the FBack (Fast Back-off) bit to "1" when in the normal operation mode.

The SdPause (Pause Transmission) bit is automatically cleared when transmission of the MAC control frame ends. Any writing of "0" to this bit is ignored.

To issue an interrupt for each packet, set the EnComp bit or all MAC Error Enable bits. You can also set interrupts to occur when a specific state occurs.

Transmission Status Register (Tx_Stat) 0x4C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED										TxPAUSE	TxMACC	TxVLAN	TxBCast	TxMCast	SQErr
TYPE											R	R	R	R	R	R
Default											0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TxHalted	Comp	TxPar	LateColl	Reserved	Reserved	ExDefer	Under	IntTx	Paused	TxDefer	ExColl	TxColl			
TYPE	R	R	R	R			R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0			0	0	0	0	0	0	0	0	0	0

Bit(s)	Mneminc	Field Name	Description
31 : 22		Reserved	
21	TxPAUSE	Pause Packet Communication	TxPAUSE (Default: 0, R) This bit is set when MAC transmits a MAC control Pause packet.
20	TxMACC	MAC Control Packet Transmission	TxMACC (Default: 0, R) This bit is set when MAC transmits a MAC control packet.
19	TxVLAN	VLAN Tag Packet Transmission	TxVLAN (Default: 0, R) This bit is set when MAC transmits a VLAN tag packet.
18	TxBCast	Broadcast Transmission	TxBCast (Default: 0, R) This bit is set when MAC transmits a broadcast packet.
17	TxMCast	Multicast Transmission	TxMCast (Default: 0, R) This bit is set when MAC transmits a multicast packet.
16	SQErr	Signal Quality Error	SQErr (Default: 0, R) Indicates that the Heartbeat signal could not be monitored at the end of a transmission.
15	TxHalted	Transmission Halted	TxHalted (Default: 0, R) Halts transmission by either clearing the TxEn bit of the Transmission Control Register or setting the HaltImm bit of the MAC Control Register.
14	Comp	Complete	Comp (Default: 0, R) Indicates that MAC either transmitted or destroyed one packet.
13	TxPar	Transmission Parity Error	TxPar (Default: 0, R) Indicates that a parity error was detected in MAC Transmission FIFO.
12	LateColl	Late Collision	LateColl (Default: 0, R) Indicates that 512-bit late time (64-Byte time) or more elapsed before a collision occurred.
11		Reserved	
10		Reserved	
9	ExDefer	Excessive Deferral	ExDefer (Default: 0, R) Indicates that MAC deferred transmission for more than MAX_DEFERRAL.
8	Under	Underrun	Under (Default: 0 ,R) Indicates that MAC Transmission FIFO became empty during transmission.
7	IntTx	Transmission Interrupt	IntTx (Default: 0, R) This bit is set when interrupt conditions specified by the Transmission Control Register are met in a packet transmission.
6	Paused	Transmitter Paused	Paused (Default: 0, R) This bit is set if transmission is paused after the current packet ends.
5	TxDefer	Transmission Deferred	TxDefer (Default: 0, R) Indicates that a packet was kept waiting due to transmission delay.
4	ExColl	Excessive Collisions	ExColl (Default: 0, R) This bit is set if collisions occur 16 times in the same packet. Instead of transmitting that packet, processing of the next packet transmission starts.
3 : 0	TxColl	Transmission Collision Count	TxColl (Default: 0, R) Number of collisions that occurred when transmitting one packet.

Figure 18-49 Transmission Status Register (2/2)

Software resets initialize the Transmission Status Register to 0x00_0000. Also, this register is cleared at the beginning of each transmission packet.

The Transmission Status flag is set each time the applicable event occurs. Also, an interrupt occurs if the corresponding bit of the Transmission Control Register is set.
The lower 5 bits of the Transmission Status Register indicate the collision count of the packet. In other words, when ExColl=1, TxColl becomes 0. If TxColl is not 0, then ExColl=0.

The MAX_DEFERRAL time is 0.24288 ms for 100 Mbps and 2.42880 ms for 10 Mbps. If the TxMCast bit (bit 17) and the TxBCast bit (bit 18) are both "0", they indicate that a unicast packet was transmitted.

18.4.5.4. Reception Control, Status Register

Reception Control Register (Rx_Ctl) 0x50

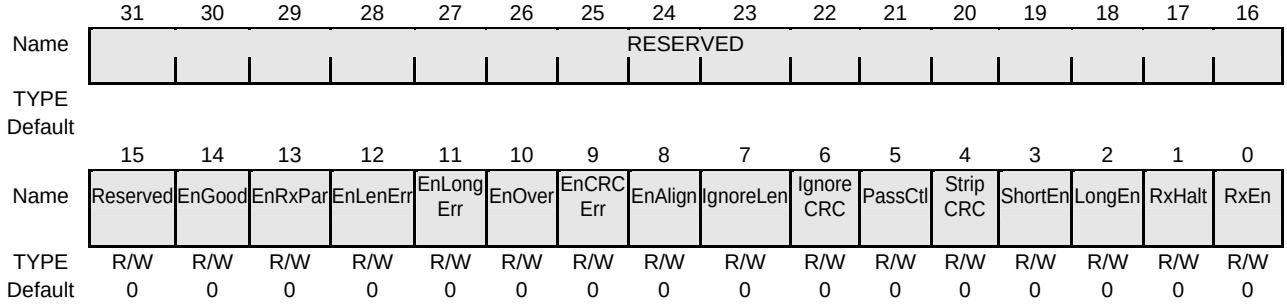


Figure 18-50 Reception Control Register

Table 18-11 Reception Control Register

Bit(s)	Mnemonic	Field Name	Description
31 : 15		Reserved	
14	EnGood	Transmission Enable	EnGood (Default: 0, R/W) Issues an interrupt if a packet is received without any error.
13	EnRxPar	Enable Reception Parity	EnRxPar (Default: 0, R/W) Issues an interrupt if a parity error is detected in the MAC Reception FIFO.
12	EnLenErr	Enable Frame Length Errors	EnLenErr(Default: 0, R/W) Issues an interrupt if a Frame Length error is detected.
11	EnLongErr	Enable Long Errors	EnLongErr (Default: 0, R/W) If the LongEn bit is not set, an interrupt is issued when a frame longer than 1518 Bytes (1522 Bytes for VLAN) is received.
10	EnOver	Enable Overflows	EnOver (Default: 0, R/W) Issues an interrupt if the MAC Reception FIFO becomes full when receiving a packet.
9	EnCRCErr	Enable CRC Errors	EnCRCErr (Default: 0, R/W) Issues an interrupt either when CRC receives an invalid packet or PHY asserts Rx_er while receiving a packet.
8	EnAlign	Enable Alignment	EnAlign (Default: 0, R/W) Issues an interrupt if CRC receives an invalid packet with a length that is not a multiple of 8.
7	IgnoreLen	Ignore Frame Length	IgnoreLen (Default: 0, R/W) Does not check the frame length.
6	IgnoreCRC	Ignore CRC Value	IgnoreCRC (Default: 0, R/W) Does not check CRC.
5	PassCtl	Pass MAC Control Frame	PassCtl (Default: 0, R/W) Passes a received MAC control frame to the system.
4	StripCRC	Strip CRC Value	StripCRC (Default: 0, R/W) Checks CRC, but removes it from the message.
3	ShortEn	Enable Short	ShortEn (Default: 0, R/W) Enables the reception of frames shorter than 64 Bytes. 1
2	LongEn	Long Enable	LongEn (Default: 0, R/W) Enables the reception of frames longer than 1518 Bytes (1522 Bytes in the case of VLAN). 1
1	RxHalt	Reception Halt Request	RxHalt (Default: 0, R/W) Halts reception after the current packet ends regardless of the packet type.

1 The above frame length does not include a preamble and a Start Frame Delimiter (SFD). For details, see 18.3.8.1 Format of MAC frames and packets.

Table 18-11 Reception Control Register

Bit(s)	Mnemonic	Field Name	Description
0	RxEn	Reception Enable	RxEn (Default: 0, R/W) This bit halts reception when cleared to "0". If this bit is cleared after MAC starts processing the recipient address, reception operation for the current packet continues and data is transferred to the system memory. At this time, the TxHalted bit of the Reception Status Register is set and indicates that the system issued a Reception Halt Request while receiving a packet. If this bit is cleared before processing of the recipient address starts, reception operation immediately halts. At this time, the RxHalted bit is set.

Hardware resets initialize the Reception Control Register to 0x0000. Software resets clear the RxEn bit, but no other bits are changed.
To issue an interrupt for each packet, set the EnGood bit or all Error Enable bits. You can also set the Reception Control Register to only issue interrupts when a particular state occurs.

Reception Status Register (Rx_Stat) 0x54

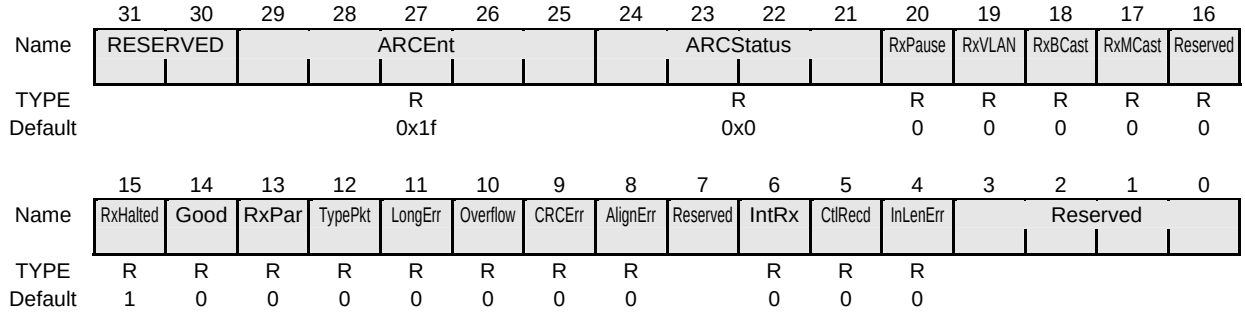


Figure 18-51 Reception Status Register

Table 18-12 Reception Status Register

Bit(s)	Mnemonic	Field Name	Description
31 : 30		Reserved	
29 : 25	ARCEnt	ARC Entry	ARCEnt (Default: 0x1F, R) This field stores the ARC entry index if the address matches. If the address does not match, all bits become "1".
24 : 21	ARCStatus	ARC Status	ARCStatus (Default: 0, R) Denotes ARC operation (see the following encoding).
20	RxPause	PAUSE Packet Reception	RxPause (Default: 0, R) This bit is set when the reception packet is a MAC control PAUSE packet.
19	RxVLAN	VLAN Tagged Packet Reception	RxVLAN (Default: 0, R) This bit is set when the reception packet is a VLAN tagged packet.
18	RxBCast	Broadcast Reception	RxBCast (Default: 0, R) This bit is set when the reception packet is a broadcast packet.
17	RxMCast	Multicast Reception	RxMCast (Default: 0, R) This bit is set when the reception packet is a multicast packet.
16		Reserved	
15	RxHalted	Reception Halted	RxHalted (Default: 1, R) This bit indicates that reception was halted either when the RxEn bit of the Reception Control Register was cleared or the HaltImm bit of the MAC Control Register was set.
14	Good	Normal Reception	Good (Default: 0, R) Indicates that a packet was received without any errors occurring.
13	RxPar	Reception Parity Error	RxPar (Default: 0, R) Indicates that a parity error was detected in the MAC Reception FIFO.
12	TypePkt	Type Packet	TypePkt (Default: 0, R) The value of the Frame Length field is greater than 1500 (no frame length check is performed). The software can use this bit to check the Protocol Type field.
11	LongErr	Long Error	LongErr (Default: 0, R) Indicates that a frame longer than 1518 Bytes (1522 Bytes for VLAN)1 was received. However, if the LongEn bit of the Reception Control Register is set, this bit is not set.
10	Overflow	Overflow	Overflow (Default: 0, R) Indicates that the MAC Reception FIFO is full and a reception byte is missing.
9	CRCErr	CRC Error	CRCErr (Default: 0, R) Indicates that either the CRC value at the end of the packet does not match the calculated value or PHY asserted Rx_er while receiving a packet.
8	AlignErr	Alignment Error	AlignErr (Default: 0, R) Indicates that the frame length is not an 8-bit multiple and that CRC is invalid.
7		Reserved	

1 The above frame length does not include a preamble and a Start Frame Delimiter (SFD). For details, see 18.3.8.1 Format of MAC frames and packets.

Table 18-12 Reception Status Register

Bit(s)	Mnemonic	Field Name	Description
6	IntRx	Reception Interrupt	IntRx (Default: 0, R) This bit is set when interrupt conditions are met due to the reception of a packet. If the EnGood bit of the Reception Control Register is set, notification is also sent for packets received without any errors. **Note: If the Address Recognition Circuit (ARC) destroys a received packet when the EnGood bit or EnCRCErr bit of the Reception Control Register is set, this IntRx bit is also set if the Good or CRCErr bit of the Reception Status Register is set. In this case however, no interrupt notification is sent and the IntMacRx bit of the Int_Src Register is not set.**
5	CtlRecd	Control Frame Received	CtlRecd (Default: 0, R) This bit is set if the ARC recognizes an address when the received packet is the MAC control frame (type=0x8808).
4	InLenErr	In Range Frame Length Error	InLenErr (Default: 0, R) If the value of the Frame Length field is 46 or less, this bit is set when the reception packet size is not 64 Bytes. Also, if the value of the Frame Length field is between 47 and 1500, this bit is set when the reception packet size is not the value of the Frame Length field + 18 (14 when in the Strip CRC mode).
3 : 0		Reserved	

Software resets initialize the Reception Status Register to 0x3E00_8000. This register is also cleared at the beginning of each reception packet.

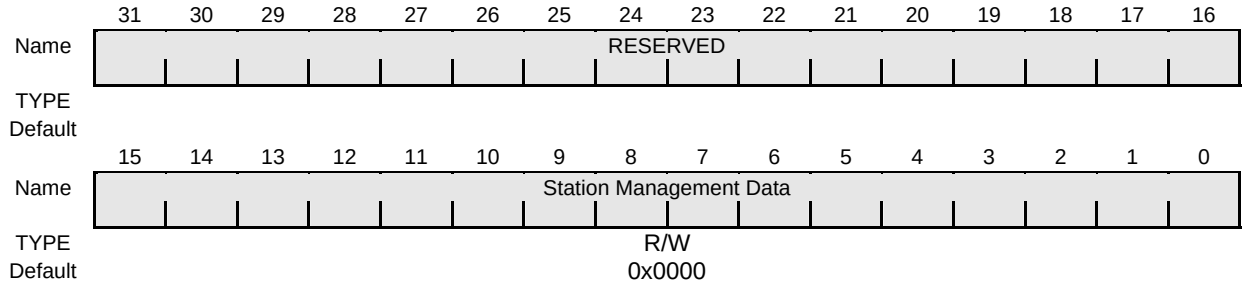
The Reception Status flag is set each time the applicable event occurs. Once the Reception Status flag is set, it remains set until the next packet is reached. When the corresponding bit of the Reception Control Register is set, an interrupt is issued.
The CtlRecd bit is set when the packet type is 0x8808 and the ARC recognizes an address. If both the RxMCast bit (bit 17) and the RxBCast bit (bit 18) are "0", they indicate that a unicast packet was received.
The ARCStatus field is encoded as follows below.

0000: Toss. MAC control frame received. PassCtl=0.
0001: Toss. The packet length is less than the minimum packet length, but is 6 Bytes or more. Keep if that is not the case.
0010: Toss. Matches ARC. NOT filtering.
0011: Reserved
0100: Toss. External CAM bit. NOT filtering.
0101: Reserved
0110: Toss. Address does not match. No external CAM. Compare disable. AND filtering.
0111: Toss. State in which the packet length is too short, the ARC result is Invalid, and the ShortEn bit is not set.
1000: Keep. Broadcast, multicast, or unicast accepting is enabled and the address matches.
1xx1: Reserved
1010: Keep. ARC matches. AND filtering.
1100: Keep. External CAM bit. AND filtering.
1110: Keep. ARC does not match. AND filtering.

Note The minimum packet length is 64 Bytes if ShortEn is not asserted, or is 14 Bytes if ShortEn is asserted.
Racing occurs between the internal ARC and external CAM. There are cases where the first signal to notify a hit blocks the other signal that notifies a hit and the ARC's Entry Status bit changes.

18.4.5.5. Station Management Registers

Station Management Data Register (MD_Data) 0x58

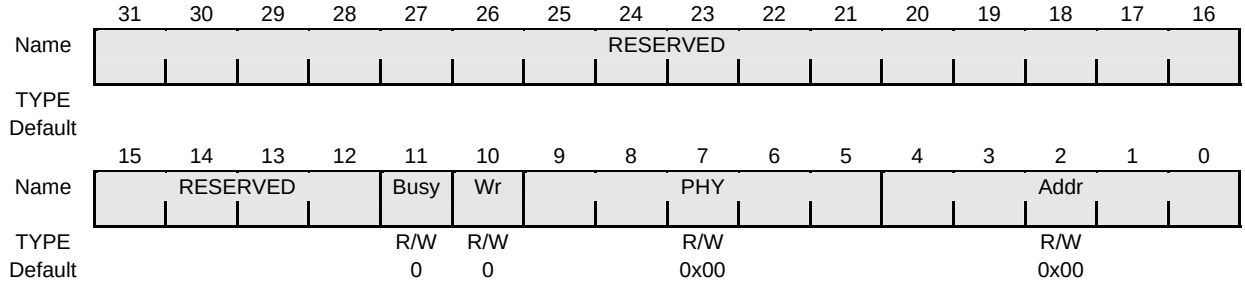


Bits	Mnemonic	Field Name	Description
31 : 16		Reserved	
15 : 0		Station Management Data	(Default: 0x0000, R/W) The MII item of the IEEE802.3 specification that relates the 100Base-T or 100 Mbps Ethernet defines the format of the Stage Management Data Register. For information on other hardware-dependent registers, see the separate PHY data sheet.

Figure 18-52 Station Management Data Register

Software resets initialize the Station Management Data Register to 0x0000.

Station Management Control, Address Register (MD_CA) 0x5C



Bit(s)	Mnemonic	Field Name	Description
31 : 12		Reserved	
11	Busy	Busy Bit	Busy (Default: 0, R/W) This bit is set when operation starts. The Ethernet Controller clears this bit when the operation ends.
10	Wr	Write	Wr (Default: 0, R/W) This bit is set to "1" during writes and is cleared to "0" during reads.
9 : 5	PHY	PHY Address	PHY (Default: 0, R/W) Address of all PHY to be read or written
4 : 0	Addr	Address	Addr (Default: 0, R/W) Address of PHY-internal register to be read or written.

Figure 18-53 Station Management Control Address Register

Before accessing the PHY Control Register, use the software to check the Busy bit of the Station Management Control, Address Register and confirm that it is not set.

The Ethernet Controller supports reading or writing of Station Management data to PHY. The setting of the Station Management Control, Address Register does not affect the operation of the Ethernet Controller.

18.4.5.6. Address Recognition Circuit (ARC) Access Register

ARC Address Register (ARC_Adr) 0x60

Bits	Mnemonic	Field Name	Description
31 : 12		Reserved	
11 : 2	ARC_Loc	ARC Address	ARC_Loc (Default: 0x00, R/W) ARC address: 4 bytes
1 : 0		Reserved	

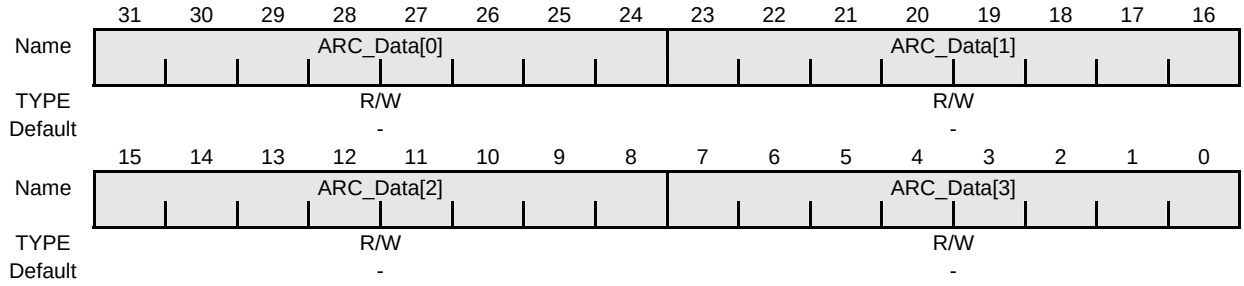
Figure 18-54 ARC Address Register

Software resets initialize the ARC Address Register to 0x0000.

During normal operation, the ARC Address Register and ARC Data Register can perform read or write operation to all ARC areas including two double-word positions immediately after flow control ARC. (See Figure 18-14.) During normal operation, writes to all other memory positions are invalid.

When the TestMode bit of the DMA Control Register is set, you can use the ARC Address Register and perform read or write operation on all RAM areas in the DMA Block.

ARC Data Register (ARC_Data) 0x64



Bits	Mnemonic	Field Name	Description
31 : 24	ARC_Data[0]	ARC Data Register [0]	ARC_Data[0] (Default -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
23 : 16	ARC_Data[1]	ARC Data Register [1]	ARC_Data[1] (Default -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
15 : 8	ARC_Data[2]	ARC Data Register [2]	ARC_Data[2] (Default: -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.
7 : 0	ARC_Data[3]	ARC Data Register [3]	ARC_Data[3] (Default: -, R/W) 4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to.

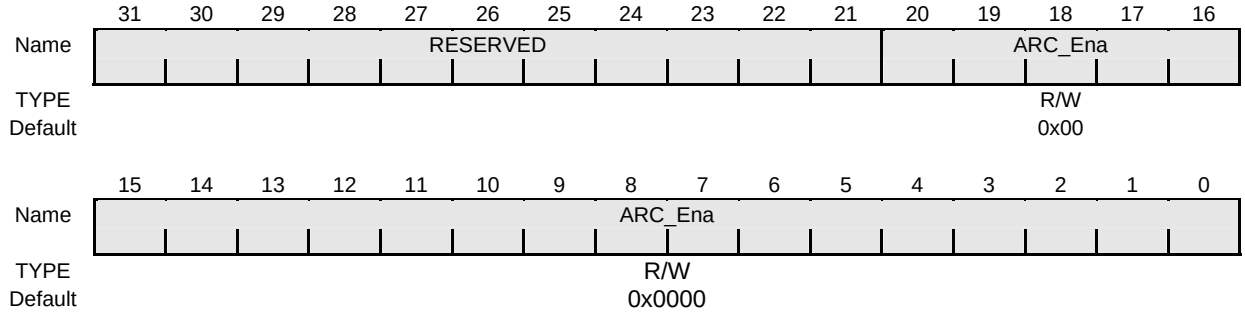
Figure 18-55 ARC Data Register

4-Byte data of ARC memory is accessed each time the ARC Data Register is read or written to. In other words, the PCI Byte Enable signal is ignored. When changing only 2 Bytes of a 4-Byte word, the system software must perform a Read/Modify/Write.

The ARC Data Register stores copies of data in ARC at the address specified by the ARC Address Register. You can read the content of the ARC Data Register as many times as you require. Writing data to this register changes the selected ARC data.

Note: In contrast to transmission/reception data in the Master mode, data transferred via the ARC Interface is interpreted in Big Endian.

ARC Enable Register (ARC _Ena) 0x68



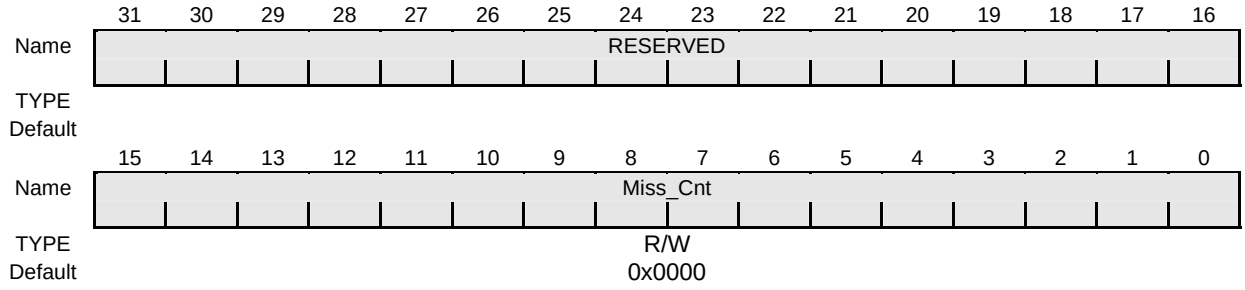
Bits	Mnemonic	Field Name	Description
31 : 21		Reserved	
20: 0	ARC_Ena	ARC Enable	ARC_Ena (Default: 0x0_0000, R/W) Sets the bits corresponding to an entry valid as address filtering.

Figure 18-56 ARC Enable Register

Hardware resets initialize the ARC Enable Register to 0x00_0000. Software resets do not change the contents of this register.

This register indicates which entry to validate as address filtering. This register can validate up to 21 entries numbered from 0 to 20.

18.4.5.7. Missed Error Count Register (Miss_Cnt) 0x7C



Bits	Mnemonic	Field Name	Description
31 : 16		Reserved	
15 : 0	Miss_Cnt	Missed Error Count	Miss_Cnt (Default: 0x0000, R/W) This register counts the number of valid packets MAC refused due to MAC Reception FIFO overflow, parity error, or the clearing of the Reception Enable bit (RxEn). This count does not include the packets refused by ARC.

Figure 18-57 Missed Error Count Register

Hardware resets initialize the Missed Error Count Register to 0x0000_0000. Software resets do not change the contents of this register.

The Missed Error Count Register displays the number of packets destroyed by various error types. This register provides the information required for station management along with the status information of the transmission/reception packet. Performing read access to the Missed Error Count Register clears it. Therefore, the software is responsible for increasing the bit count and retaining the precise total error count.

The MissRoll bit of the MAC Control Register is set when the Missed Error Count Register rolls over from 0x7FFF to 0x8000. Also, in this case an interrupt occurs if the EnMissRoll bit is set.

To generate interrupts more frequently from the Station Management software, you can set the Missed Error Count Register to a value that is close to the final count value of 0x7FFF. For example, if you set this register to 0x7F00, an interrupt occurs when errors have occurred 256 times.

0x74 through 0x78 is reserved to maintain compatibility with previous Toshiba products.

Chapter 19. QUAD SIO

19.1. Features

The TX4939 asynchronous Serial I/O (SIO) interface has four full duplex UART channels (SIO). SIO has the following features.

- *Configurable 2 or 4 SIO channels*
- *Full duplex transmission (simultaneous transmission and reception)*
- *On-chip baud rate generator*
- *Supports hardware flow control using RTS*/CTS* for SIO channel 0 only.*
- *Transmit FIFO: 8 bits × 8 stages*
- *Reception FIFO: 13 bits × 16 stages (data: 8 bits, status: 5 bits)*
- *Supports DMA transfer*
- *Supports multi-controller systems*
- *Supports Master/Slave operation*

19.2. Block Diagram

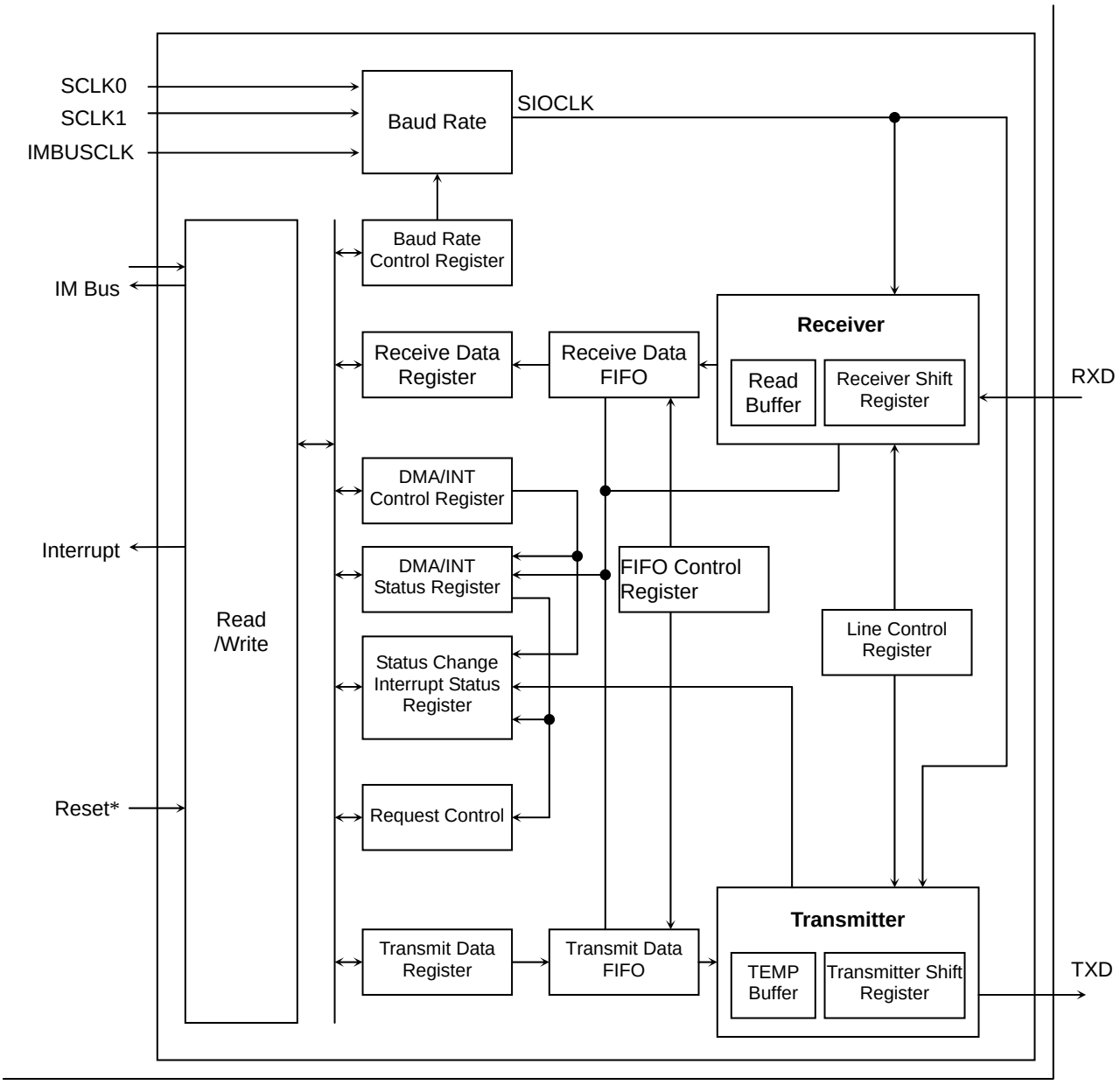


Figure 19-1 SIO Internal Block Diagram

19.3. Detailed Explanation

TX4939 has dedicated ports for SIO0 and SIO1 while sharing SIO2 and SIO3 ports with SPI and GPIO. Depending on the requirement, SIO2 and SIO3 ports can be activated by writing to Pin Configuration Register (PCFG) described in Chapter 7. Configuration Registers after boot up. TX4939 also allows SIO0 hardware flow control signals based on PCFG. Activation of SIO0 hardware flow control signals, SIO2 and SIO3 ports is independent of Product Mode Selection (see Chapter 4. Boot Configuration, Table 4-1 Boot Configuration Details, for Product Mode Selection). Table 19-1 gives PCFG settings that are needed to activate SIO2 and SIO3 and SIO0 CTSB/RSTB (hardware flow control signals).

Table 19-1 Usable SIO channels based on PCFG settings

PCFG[63:61,57]	Active Interface	SIO0 (CTSB/RTSB)	SIO2	SIO3
4'bxx0x	SPI	Unusable	Unusable	Unusable
4'b1x11	GPIO	Unusable	Unusable	Unusable
4'b0010	SIO	Usable	Unusable	Usable
4'b0011	SIO0/GPIO	Usable	Unusable	Unusable
4'b0110	SIO2/SIO3	Unusable	Usable	Usable
4'b0111	SIO2/GPIO	Unusable	Usable	Unusable
4'b1x10	GPIO, SIO3	Unusable	Unusable	Usable

Note: PCFG[63:61,57] defaults to GPIO interface after boot up. The behavior of unusable SIO channels as given in Table 19-1 is not defined when unusable SIO channels are accessed through IMBUS.

The following sections describe a single SIO channel that is also applicable to the remaining 3 SIO channels unless otherwise specified.

19.3.1. Overview

During reception, serial data that are input as an RXD signal from an external source are converted into parallel data, then are stored in the Receive FIFO buffer. Parallel data stored in the FIFO buffer are fetched by either CPU or DMA transfer.

During transmission, parallel data written to the Transmit FIFO buffer by CPU or DMA transfer are converted into serial data, then are output as a TXD signal.

19.3.2. Data Format

The TX4939 SIO can use the following data formats.

Data Length : 8/7 bits
Stop Bit : 1/2 bits
Parity Bit : Yes/No
Parity Format : Even/Odd
Start Bit : Fixed to 1 bit

Figure 19-2 illustrates the data frame when making each setting.

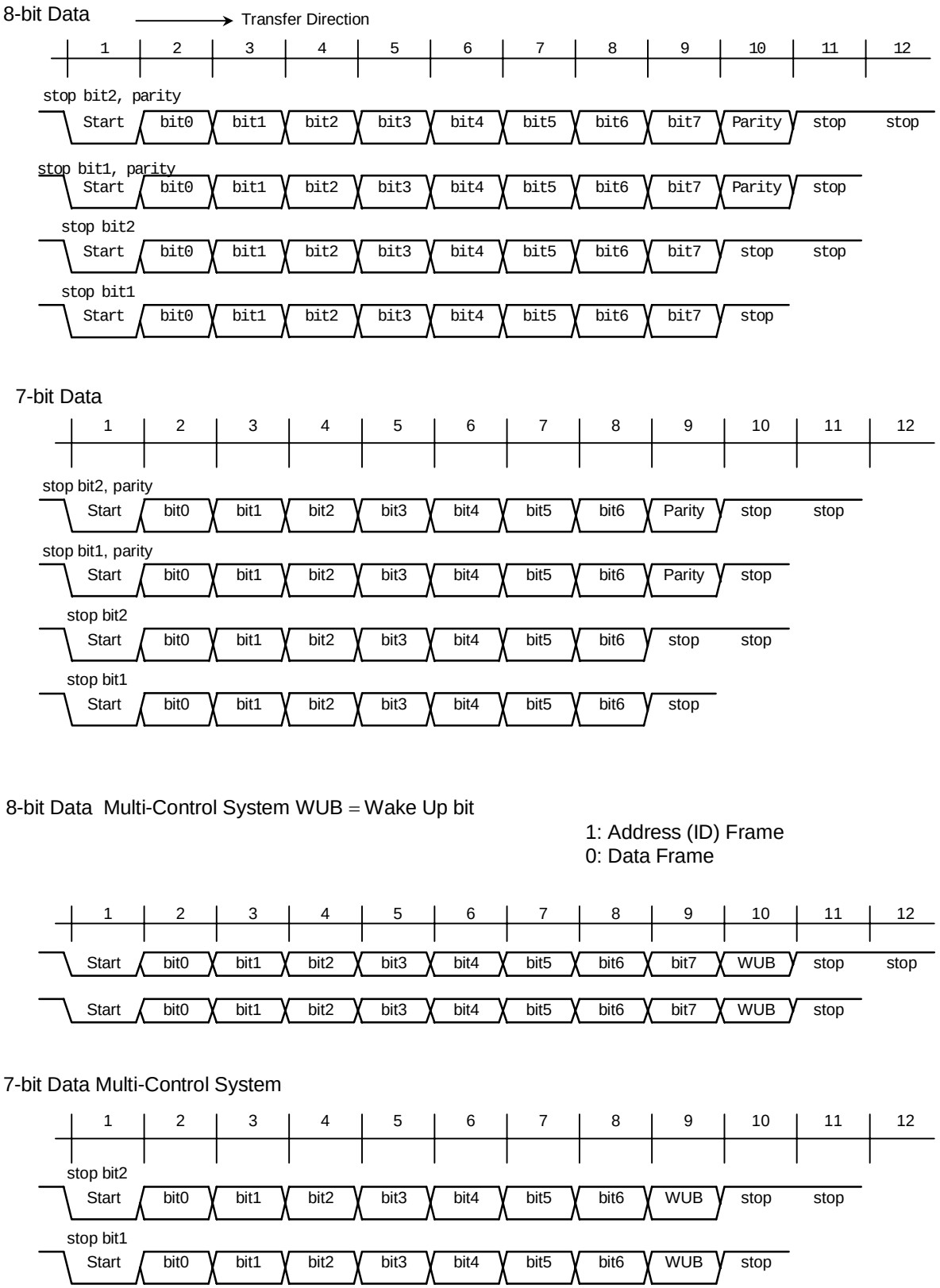


Figure 19-2 Data Frame Configuration

19.3.3. Serial Clock Generator

Generates the Serial Clock (SIOCLK). SIOCLK determines the serial transfer rate and has a frequency that is 16× the baud rate. One of the following can be selected as the source for the Serial Clock (SIOCLK).

- *Internal System Clock (IMBUSCLK)*
- *Internal Baud rate Clock0 (SCLK0)*
- *Internal Baud rate Clock1 (SCLK1)*
- *Baud rate generator circuit output*

The IMBUSCLK frequency can be selected from frequencies that are 1/4, 1/6, 1/10, or 1/12 the frequency of the CPU clock. SCLK0 (baud rate clock0) is generated by TX4939 Audio Clock Generator with a frequency of 14.7456Mhz. SCLK1 (baud rate clock1) is generated using ATA100 clock by the Master Clock Generator with a frequency of 14.7465Mhz. Please refer to Chapter 5. Clock Generators for SCLK0, SCLK1 (baud rate clocks) generation information.

The baud rate generator is a circuit that divides these clock signals according to the following formula.

$$Baud\ Rate = \frac{f_c}{Prescale \times Divisor \times 16}$$

fc: Clock frequency of IMBUSCLK or Clock frequency of SCLK0,SCLK1
Prescalar Value: 2, 8, 32, 128
Divide Value: 1, 2, 3,…255

Table 19-2 and Table 19-3 gives example settings of divide values relative to representative baud rates. Figure 19-3 shows the block diagram of the baud rate generator used to generate SIOCLK.

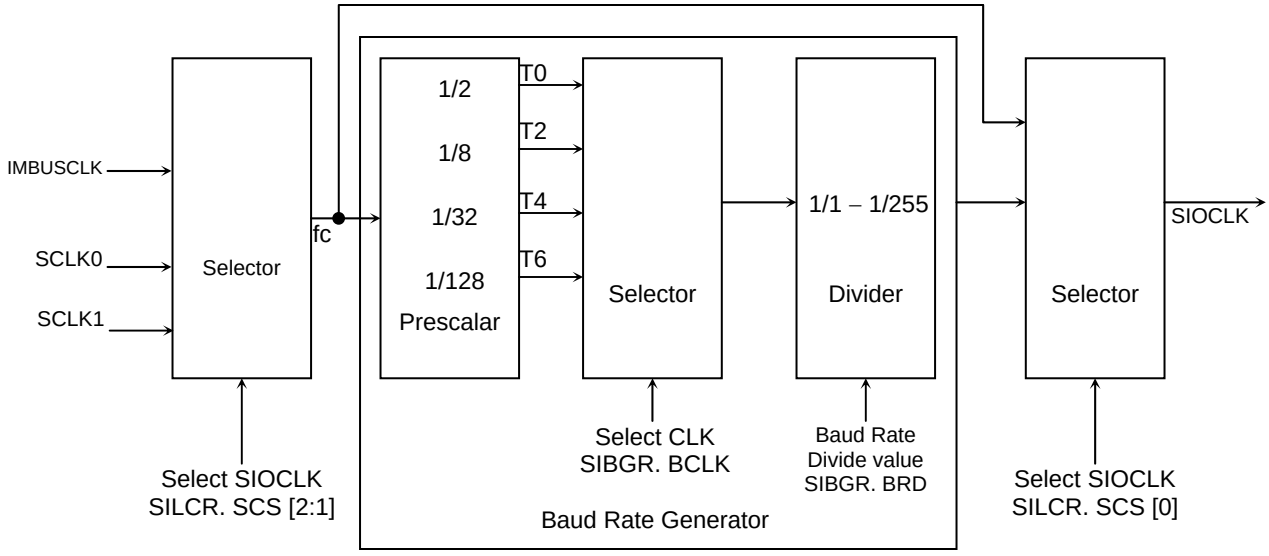


Figure 19-3 Baud Rate Generator and SIOCLK Generator

It is possible to correctly receive data if the error of the baud rate set by this controller is within 3.12 % of the target baud rate (communication baud rate).

Table 19-2 gives example divide value settings and percentage error from targeted baud rates when IMBUSCLK is used to generate SIOCLK (Figure 19-3). Exact target baud rates cannot be achieved when using IMBUSCLK to generate SIOCLK.

Table 19-2 Example Divide Value Settings when using IMBUSCLK (and error [%] from target baud rate value)

fc [MHz]	Target Bps	Prescalar Value (SIBGR.BCLK) and Divide Value (SIBGR.BRD)			
		2	8	32	128
IMBUSCLK = 100 (GBUSCLK=200)	300				163(0.147%)
	600				81(-0.469%)
	1200			163(0.147%)	41(0.756%)
	2400			81(-0.469%)	
	4800		163(0.147%)	41(0.756%)	
	9600		81(-0.469%)		
	14400	217(-0.006%)	54(-0.469%)		
	28800	109(0.452%)	27(-0.469%)		
	57600	54(-0.469%)			
	115200	27(-0.469%)			
	230400	14(3.119%)			
	460800	7(3.119%)			
	921600				
IMBUSCLK = 75 (GBUSCLK=150)	300				122 (0.058 %)
	600			244 (0.058 %)	61 (0.058 %)
	1200			122 (0.058 %)	
	2400		244 (0.058 %)	61 (0.058 %)	
	4800		122 (0.058 %)		
	9600		61 (0.058 %)		
	14400	163 (-0.147 %)	41 (-0.756 %)		
	28800	81 (0.469 %)			
	57600	41 (-0.756 %)			
	115200	20 (1.725 %)			
	230400	10 (1.725 %)			
	460800	5 (1.725 %)			
	921600				

Table 19-3 gives example divide values and jitter per bit time for targeted baud rates when SCLK0 is used to generate SIOCLK (Figure 19-3). Exact target baud rates are achieved when using SCLK0 to generate SIOCLK. However please note that SCLK0 inherently has jitter due to the process in which it is generated. SCLK0 (baud rate clock) generation information can be obtained from Chapter 5. Clock Generators.

Table 19-3 Example Divide Value Settings when using SCLK0 (and jitter value per bit time [%])

Fc [MHz]	Target bps	Bypass baud rate generator (Note 2)	Prescalar Value (SIBGR.BCLK) and Divide Value (SIBGR.BRD)			
			2	8	32	128
SCLK0 = 14.7456 (**Note 1**)	300				96(0.0001%)	24(0.0001%)
	600			192(0.0001%)	48(0.0001%)	12(0.0001%)
	1200			96(0.0002%)	24(0.0002%)	6(0.0002%)
	2400		192(0.0004%)	48(0.0004%)	12(0.0004%)	3(0.0004%)
	4800		96(0.0008%)	24(0.0008%)	6(0.0008%)	
	9600		48(0.0017%)	12(0.0017%)	3(0.0017%)	
	14400		32(0.0025%)	8(0.0025%)	2(0.0025%)	
	28800		16(0.0050%)	4(0.0050%)	1(0.0050%)	
	57600		8(0.0100%)	2(0.0100%)		
	115200		4(0.0200%)	1(0.0200%)		
	230400		2(0.0400%)			
	460800		1(0.0800%)			
	921600	**Note 2** (0.1600%)				

Note 1: SCLK0 is not generated by default. MCLKCTL:BDE needs to be set for SCLK0 generation. SCLK0 (baud rate clock) generation information can be obtained from Chapter 5. Clock Generators.

Note 2: The baud rate generator circuit (Figure 19-3) is bypassed and SIOCLK is sourced by SCLK0 when SILCR.SCS is set to 3'b010.

Table 19-4 gives example divide value settings and percentage error from targeted baud rates when SCLK1 is used to generate SIOCLK (Figure 19-3). Exact target baud rates cannot be achieved when using SCLK1 to generate SIOCLK.

Table 19-4 Example Divide Value Settings when using SCLK1 (and error [%] from target baud rate value)

fc [MHz]	Target Bps	Bypass baud rate generator (Note 2)	Prescalar Value (SIBGR.BCLK) and Divide Value (SIBGR.BRD)			
			2	8	32	128
SCLK1 = 14.7465 (**Note 1**)	300				96(-0.006%)	24(-0.006%)
	600			192(-0.006%)	48(-0.006%)	12(-0.006%)
	1200			96(-0.006%)	24(-0.006%)	6(-0.006%)
	2400		192(-0.006%)	48(-0.006%)	12(-0.006%)	3(-0.006%)
	4800		96(-0.006%)	24(-0.006%)	6(-0.006%)	
	9600		48(-0.006%)	12(-0.006%)	3(-0.006%)	
	14400		32(-0.006%)	8(-0.006%)	2(-0.006%)	
	28800		16(-0.006%)	4(-0.006%)	1(-0.006%)	
	57600		8(-0.006%)	2(-0.006%)		
	115200		4(-0.006%)	1(-0.006%)		
	230400		2(-0.006%)			
	460800		1(-0.006%)			
	921600	**Note 2** (-0.006%)				

Note 1: SCLK1 (baud rate clock) generation information can be obtained from Chapter 5. Clock Generators

Note 2: The baud rate generator circuit (Figure 19-3) is bypassed and SIOCLK is sourced by SCLK1 when SILCR.SCS is set to 3'b100.

19.3.4. Data Reception

When the Serial Data Reception Disable bit (RSDE) of the Flow Control Register (SIFLCRn) is set to "0", reception operation starts after the RXD signal start bit is detected. Start bits are detected when the RXD signal transitions from the High state to the Low state. Therefore, the RXD signal is not interpreted as a start bit if it is Low when the Serial Data Reception Disable bit is set to "0".

The received data are stored in the Receive FIFO. The Reception Data Full bit (RDIS) of the DMA/Interrupt Status Register (SIDISRn) is set if the byte count of the stored reception data exceeds the value set by the Receive FIFO Request Trigger Level field (RDIL) of the FIFO Control Register (SIFCRn).

An interrupt is signaled when the Reception Data Interrupt Enable bit (RIE) of the DMA/Interrupt Control Register (SIDICRn) is set. The received data can be read from the Receive FIFO Data Register (SIRFIFOn).

In addition, DMA transfer is initiated when the Reception Data DMA Enable bit (RDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

19.3.5. Data Transmission

Data stored in the Transmission Data FIFO are transmitted when the Serial Data Transmission Disable bit (TSDE) of the Flow Control Register (SIFLCRn) is set to "0".

If the available space in the Transmit FIFO is greater than the byte count set by the Transmit FIFO Request Trigger Level (TDIL) of the Control Register (SIFCRn), the transmission data empty bit (TDIS) of the DMA/Interrupt Status Register (SIDISRn) is set.

An interrupt is signaled when the Transmission Data Interrupt Enable bit (TIE) of the DMA/Interrupt Control Register (SIDICRn) is set.

In addition, DMA transfer is initiated when the Transmission Data DMA Enable bit (TDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

19.3.6. DMA Transfer

The four SIO channels share two DMA channels, one for reception and transmission respectively. In TX4939, only one SIO channel can make use of DMA operations at a time. The SIO channel that is to use DMA operations can be selected by writing to Chip Configuration Register (CCFG) described in Chapter 7. Chip Configuration. The DMA Request Select field (INTDMA[7:0]) of the Pin Configuration Register (PCFG) can be used to allocate DMA channels for the reception and transmission channel in the following manner,

SIO Channel 0 Reception	DMA Channel 2
SIO Channel 0 Transmission	DMA Channel 3.

Set the DMA Channel Control Register of the DMA Controller as described below.

DMA Request Polarity	Low Active	DMCCRn.ACKPOL = 0
DMA Acknowledge Polarity	Low Active	DMCCRn.REQPOL = 0
Request Detection	Level Detection	DMCCRn.EGREQ = 0
Transfer Size	1 Byte	DMCCRn.XFSZ = 000b
Transfer Address Mode	Dual	DMCCRn.SNGAD = 0

In the case of transmission channels, the address of the Transmit FIFO Register (SITFIFOn) is set in the DMAC Destination Address Register (DMDARn). In the case of reception channels, the address of the Receive FIFO Register (SIRFIFOn) is set in the DMAC Source Address Register (DMSARn). Please set the addresses specified in “19.4.8 Transmit FIFO Register” and “19.4.9 Receive FIFO Register” since the set address differs depending on the Endian mode.

19.3.7. Flow Control

This section applies to SIO0 channel only.
SIO0 supports hardware flow control that uses the RTS*/CTS* signal.
The CTS* (Clear to Send) input signal indicates that data can be received from the reception side when it is Low. Setting the Transmission Enable Select bit (TES) of the Flow Control Register (SIFLCRn) makes transmission flow control that uses the CTS* signal more effective.

It is also possible to generate status change interrupts by changing the state of the CTS* signal. The conditions in which interrupts are generated can be selected by the CTSS Active Condition field of the DMA/Interrupt Control Register (SIDICRn).

Setting the RTS* (Request to Send) output signal to High requests the transmission side to pause transmission. Transmission resumes when the reception side becomes ready and the RTS* signal is set to Low.

Setting the Reception Enable Select bit (RCS) of the flow Control Register (SIFLCRn) makes reception flow control that uses the RTS* signal more effective. The RTS* signal pin status becomes High when data of the byte count set by the RTS Active Trigger Level field (RTSTL) of the Flow Control Register (SIFLCRn) accumulates in the Receive FIFO. The RTS* signal can also be made High by setting the RTS Software Control bit (RTSSC) of the Flow Control Register (SIFLCRn). Setting this bit requests the transmission side to pause transmission.

19.3.8. Reception Data Status

Status data such as the following is also stored in the Receive FIFO.

- *Overrun error*
  An overrun error is generated if all 16-stage Receive FIFO buffers become full and more data is transferred to the Reception Read buffer. When this occurs, the Overrun Status bit is set by the last stage of the Receive FIFO.

- *Parity error*
  A parity error is generated when a parity error is detected in the reception data.

- *Framing error*
  A framing error is generated when “0” is detected at the first stop bit of the reception data.

- *Break reception*
  A break is detected when a framing error occurs in the reception data and all data in a single frame are “0”. When

this occurs, 2 frames (2 Bytes) of 0x00 data are stored in the Receive FIFO.

The Reception Error Interrupt bit (SIDISR.ERI) of the DMA/Interrupt Status Register (SIDISRn) is set when one of the following errors is detected: an overrun error, a parity error, or a framing error. An interrupt is signaled if the Reception Error Interrupt Enable bit of the DMA/Interrupt Control Register (SIDICRn) is set.

The UART Break Detect bit (UBRKD) and the Receiving Break bit (RBRKD) of the Status Change Interrupt Status Register (SISCISR) is set when a break is detected. The UART Break Detect bit (UBRKD) remains set until it is cleared by the software. The Receiving Break bit (RBRKD) is automatically cleared when a frame is received that is not a break.

The status of the next reception data to be read is set to the Overrun Error bit (UOER), Parity Error bit (UPER), Framing Error bit (UFER), and the Receive Break bit (RBRKD). Each of these statuses is updated when reception data is read from the Receive FIFO Register (SIRFIFOn).

During DMA transfer, an error is signaled and DMA transfer stops with error data remaining in the Receive FIFO if either an error (Framing Error, Parity Error, or Overrun Error) or a Reception time out (TOUT) is detected. If a Reception Error occurs during DMA transfer, use the Receive FIFO Reset bit (RFRST) of the FIFO Control Register (SIFCRn) to clear the Receive FIFO. However, a software reset will be required if a reception overrun error has occurred. Refer to “19.3.10 Software Reset” for more information.

19.3.9. Reception Time Out

A Reception time out is detected and the Reception Time Out bit (TOUT) of the DMA/Interrupt Status Register (SIDISR) is set under the following conditions.

- *Non-DMA transfer mode (SIDICRn.RDE = 0):*
  When at least 1 Byte of reception data exists in the Receive FIFO and the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed

- *DMA transfer mode (SIDICRn.RDE = 1):*
  When the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed regardless of whether reception data exists in the Receive FIFO

19.3.10. Software Reset

It is necessary to reset the FIFO and perform a software reset in the following situations.

- *After transmission data is set in FIFO, etc., transmission started but stopped before its completion*

- *An overrun occurred during data reception*

Software reset is performed by setting the Software Reset bit (SWRST) of the FIFO Control Register (SIFCR). This bit automatically returns to “0” after initialization is complete. This bit must be set again since all SIO registers are initialized by software resets.

19.3.11. Error Detection/Interrupt Signaling

An interrupt is signaled if an error or an interrupt cause is detected, the corresponding status bit is set and the corresponding Interrupt Enable bit is set.

The following figure shows the relationship between the status bit for each interrupt cause and each interrupt enable bit. Please refer to the explanation for each status bit for more information about each interrupt cause.

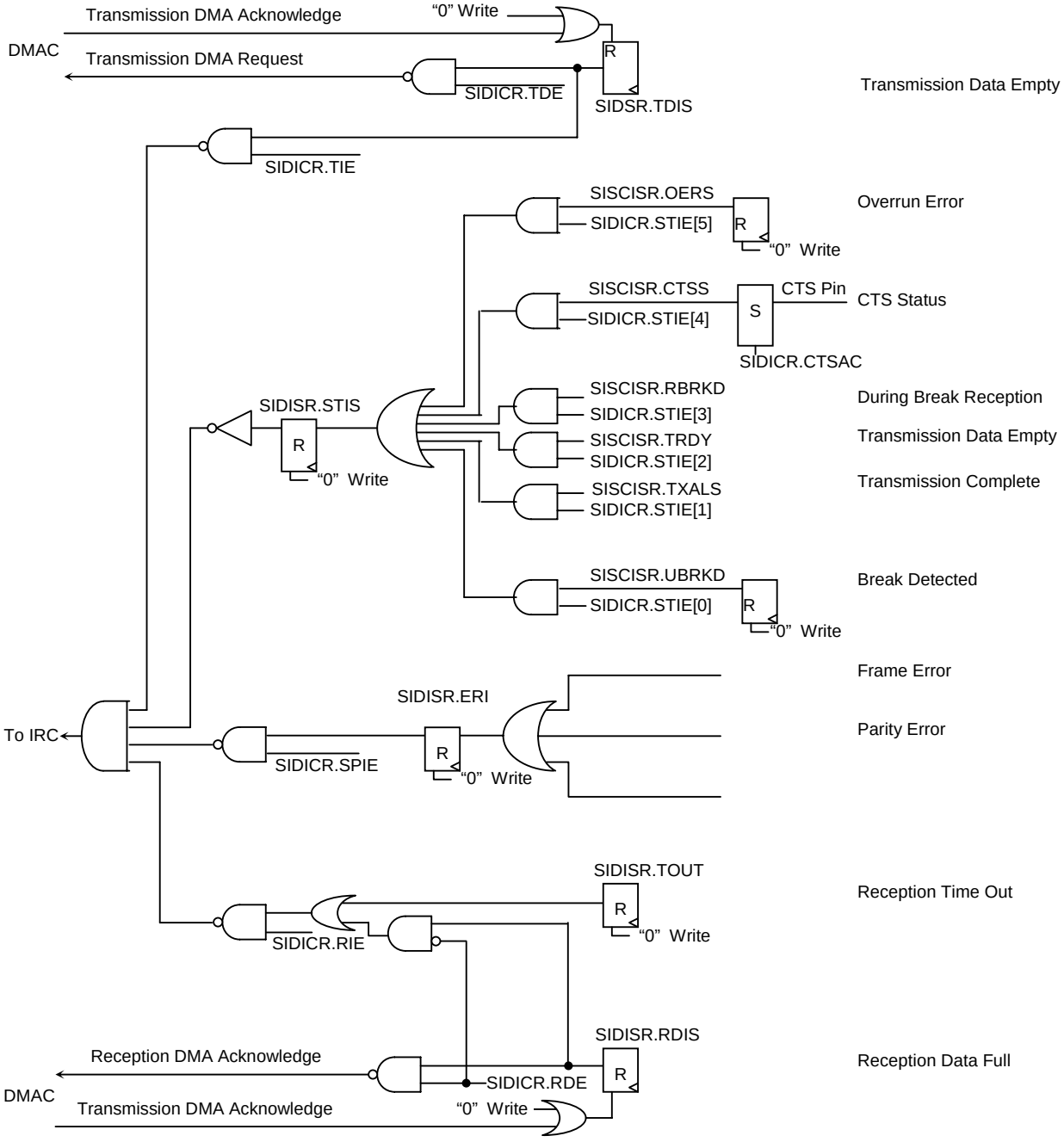


Figure 19-4 Relationship Between Interrupt Status Bits and Interrupt Signals

19.3.12. Multi-Controller System

The Multi-Controller System consists of one Master Controller, and multiple Slave Controllers as shown below in Figure 19-5.

In the case of the Multi-Controller System, the Master Controller transmits an address (ID) frame to all Slave Controllers, then transmits and receives data with the selected Slave Controller. Slave Controllers that were not selected will ignore this data.

Data frames whose data frame Wake Up bits (WUB) are “1” are handled as address (ID) frames. Data frames whose Wake Up bit (WUB) is “0” are handled as data frames.

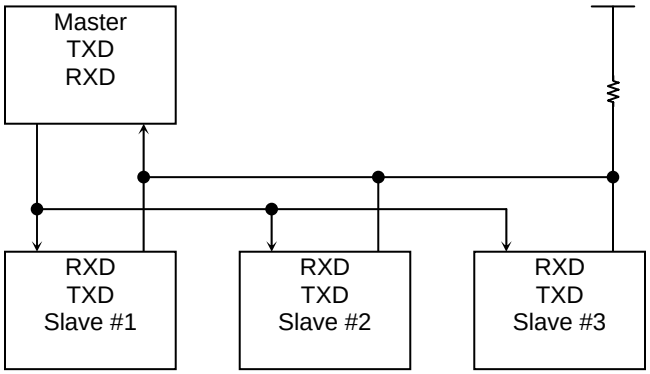


Figure 19-5 Example Configuration of Multi-Controller System

The data transfer procedure for the Multi-Controller System is as follows.

(1) The Master and Slave Controllers set the Mode field (UMODE) of the Line Control Register (SILCR) to “10” or “11” to set the Multi-Controller System mode. Also, the Slave Controller sets the open drain enable bit (UODE) of the Line Control Register (SILCR), setting the TXD output signal to open drain output.

(2) The Slave Controller sets the Reception Wake Up bit (RWUB) of the Line Control Register (SILCR), making it possible to receive address (ID) frames from the Master Controller.

(3) The Master Controller sets the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), and transmits the address (ID) of the selected Slave Controller. This causes the address (ID) frame to be transmitted. The Reception after Address Transmission Wake Up bit (RWUB) is cleared, enabling reception of data frames.

(4) Since the Reception Wake Up bit (RWUB) is set, the Slave Controller generates an interrupt to the CPU by receiving an address (ID) frame. The CPU compares its own address (ID) and the received data together. If they do not match, the Reception Wake Up bit (RWUB) is cleared, making data frame reception possible.

(5) The Master Controller and the selected Slave Controller clear the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), then set the mode that transmits data frames.

(6) Transmit/Receive data between the Master Controller and the selected Slave Controller. Then, Slave Controllers that were not selected ignore data frames since the Reception Wake Up bit (RWUB) is still set.

Note: Break signaling cannot be done by TX4939 as a slave in a Multi-Controller system”

19.4. Registers

With the exception of DMA access to the Transmit FIFO Register or the Receive FIFO Register, please use Word access when accessing register in the Serial I/O Port.

Table 19-5 SIO Registers

Offset Address	Mnemonic	Register Name
SIO0 (Channel 0)		
0xF300	SILCR0	Line Control Register 0
0xF304	SIDICR0	DMA/Interrupt Control Register 0
0xF308	SIDISR0	DMA/Interrupt Status Register 0
0xF30C	SISCISR0	Status Change Interrupt Status Register 0
0xF310	SIFCR0	FIFO Control Register 0
0xF314	SIFLCR0	Flow Control Register 0
0xF318	SIBGR0	Baud Rate Control Register 0
0xF31C	SITFIFO0	Transmit FIFO Register 0
0xF320	SIRFIFO0	Receive FIFO Register 0

Offset Address	Mnemonic	Register Name
SIO1 (Channel 1)		
0xF400	SILCR1	Line Control Register 1
0xF404	SIDICR1	DMA/Interrupt Control Register 1
0xF408	SIDISR1	DMA/Interrupt Status Register 1
0xF40C	SISCISR1	Status Change Interrupt Status Register 1
0xF410	SIFCR1	FIFO Control Register 1
0xF414	SIFLCR1	Flow Control Register 1
0xF418	SIBGR1	Baud Rate Control Register 1
0xF41C	SITFIFO1	Transmit FIFO Register 1
0xF420	SIRFIFO1	Receive FIFO Register 1

Offset Address	Mnemonic	Register Name
SIO2 (Channel 2)		
0xF380	SILCR2	Line Control Register 2
0xF384	SIDICR2	DMA/Interrupt Control Register 2
0xF388	SIDISR2	DMA/Interrupt Status Register 2
0xF38C	SISCISR2	Status Change Interrupt Status Register 2
0xF390	SIFCR2	FIFO Control Register 2
0xF394	SIFLCR2	Flow Control Register 2
0xF398	SIBGR2	Baud Rate Control Register 2
0xF39C	SITFIFO2	Transmit FIFO Register 2
0xF3A0	SIRFIFO2	Receive FIFO Register 2

Offset Address	Mnemonic	Register Name
SIO3 (Channel 3)		
0xF480	SILCR3	Line Control Register 3
0xF484	SIDICR3	DMA/Interrupt Control Register 3
0xF488	SIDISR3	DMA/Interrupt Status Register 3
0xF48C	SISCISR3	Status Change Interrupt Status Register 3
0xF490	SIFCR3	FIFO Control Register 3
0xF494	SIFLCR3	Flow Control Register 3
0xF498	SIBGR3	Baud Rate Control Register 3
0xF49C	SITFIFO3	Transmit FIFO Register 3
0xF4A0	SIRFIFO3	Receive FIFO Register 3

19.4.1. Line Control Register 0,1,2,3

(SILCR0, SILCR1, SILCR2, SILCR3)

Table 19-6 Address offsets for Line Control Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF300	SILCR0	Line Control Register 0
SIO1	0xF400	SILCR1	Line Control Register 1
SIO2	0xF380	SILCR2	Line Control Register 2
SIO3	0xF480	SILCR3	Line Control Register 3

This register specifies the format of asynchronous transmission/reception data.

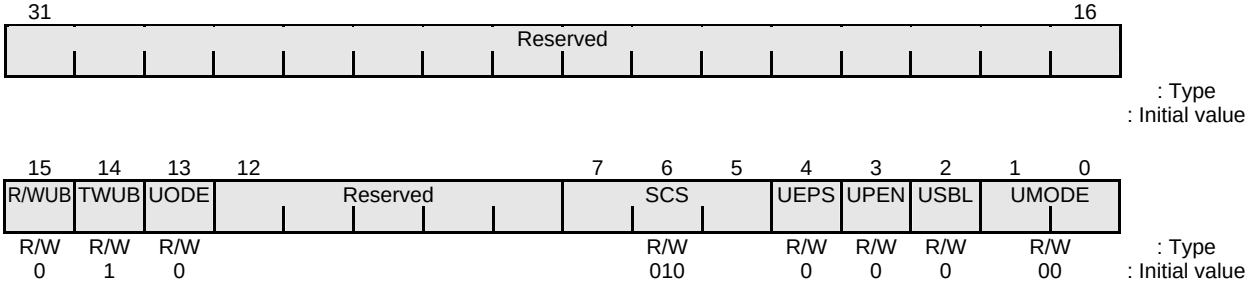


Figure 19-6 Line Control Register

Table 19-7 Line Control Register

Bit	Mnemonic	Field Name	Description	R/W
31:16		Reserved		—
15	RWUB	Receive Wake Up Bit	Wake Up Bit for Receive (Default: 0) When in the Multi-Controller System mode, this field selects whether to receive address (ID) frames whose Wake Up bits (WUB) are “1” or to receive data frames whose Wake Up bits (WUB) are “0”. This value is undefined when not in the Multi-Controller System mode. 0: Receive data frames. 1: Receive address (ID) frames.	R/W
14	TWUB	Transmit Wake Up Bit	Wake Up Bit for Transmit (Default: 1) When in the Multi-Controller System mode, this field specifies the Wake Up bit (WUB). This value is undefined when not in the Multi-Controller System mode. 0: Data frame transfer (WUB = 0) 1: Address (ID) frame transfer (WUB = 1)	R/W
13	UODE	Open Drain Enable	TXD Open Drain Enable (Default: 0) This field selects the output mode of the TXD signal. When in the Multi-Controller System mode, the Slave Controller must set the TXD signal to Open Drain. 0: Totem pole output 1: Open drain output	R/W
12:8		Reserved		—
7:5	SCS	Clock Select	SIO Clock Select (Default: 010) This field selects the serial transfer clock. The clock frequency that is the serial transfer clock divided by 16 becomes the baud rate (bps). 000: Internal clock (IMBUSCLK) 001: Baud rate generator output that divided IMBUSCLK 010: Internal Baud rate clock0 (SCLK0) 011: Baud rate generator output that divided SCLK0 100: Internal Baud rate clock1 (SCLK1) 101: Baud rate generator output that divided SCLK1 others: Reserved	R/W
4	UEPS	Even Parity Select	SIO Even Parity Select (Default: 0) This field selects the parity mode. 0: Odd parity 1: Even parity	R/W
3	UPEN	Parity Check Enable	SIO Parity Enable (Default: 0) This field selects whether to perform the parity check. This bit must be cleared in multidrop systems (i.e., when the UMODE field is 10 or 11.) 0: Disable the parity check 1: Enable the parity check	R/W
2	USBL	Stop Bit Length	SIO Stop Bit Length (Default: 0) This field specifies the stop bit length. 0: 1 bit 1: 2 bit	R/W
1:0	UMODE	Mode	SIO Mode (Default: 00) This field sets the data frame mode. 00: 8-bit data length 01: 7-bit data length 10: Multi-Controller 8-bit data length 11: Multi-Controller 7-bit data length	R/W

19.4.2. DMA/Interrupt Control Register 0,1,2,3

(SIDICR0, SIDICR1, SIDICR2, SIDICR3)

Table 19-8 Address offsets for DMA/Interrupt Control Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF304	SIDICR0	DMA/Interrupt Control Register 0
SIO1	0xF404	SIDICR1	DMA/Interrupt Control Register 1
SIO2	0xF384	SIDICR2	DMA/Interrupt Control Register 2
SIO3	0xF484	SIDICR3	DMA/Interrupt Control Register 3

This register uses either DMA or interrupts to execute the Host Interface.

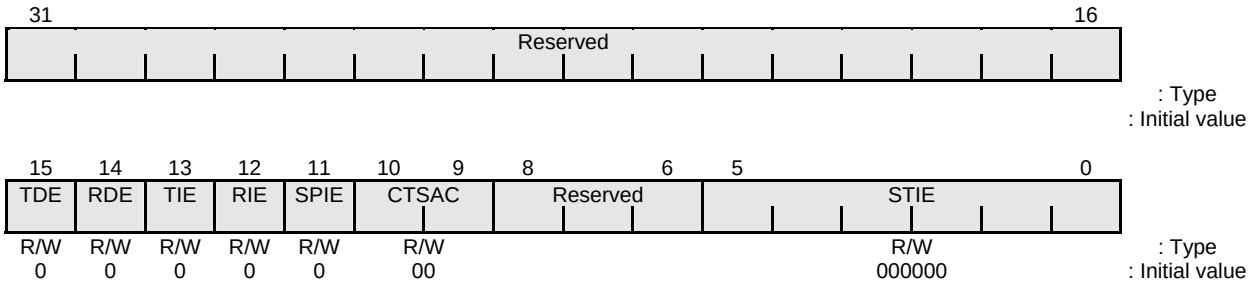


Figure 19-7 DMA/Interrupt Control Register

Table 19-9 DMA/Interrupt Control Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	TDE	Transmit DMA Transfer Enable	Transmit DMA Enable (Default: 0) This field sets whether to use DMA in the method for writing transmission data to the Transmit FIFO. 0: Do not use DMA. 1: Use DMA.	R/W
14	RDE	Receive DMA Transfer Enable	Receive DMA Enable (Default: 0) This field sets whether to use DMA in the method for reading reception data from the Receive FIFO. 0: Do not use DMA. 1: Use DMA.	R/W
13	TIE	Transmit Data Empty Interrupt Enable	Transmit Data Empty Interrupt Enable (Default: 0) When there is open space in the Transmit FIFO, this field sets whether to signal an interrupt. Set “0” when in the DMA Transmit mode (TDE = 1). 0: Do not signal an interrupt when there is open space in the Transmit FIFO. 1: Signal an interrupt when there is open space in the Transmit FIFO.	R/W
12	RIE	Reception Data Full Interrupt Enable	Receive Data Full Interrupt Enable (Default: 0) This field sets whether to signal interrupts when reception data is full (SIDISRn.RDIS = 1) or a reception time out (SIDISRn.TOUT = 1) occurs. Set to “0” when in the DMA Receive mode (RDE = 1). 0: Do not signal interrupts when reception data is full/reception time out occurred. 1: Signal interrupts when reception data is full/reception time out occurred.	R/W
11	SPIE	Reception Error Interrupt Enable	Receive Data Error Interrupt Enable (Default: 0) This field sets whether to signal interrupts when a reception error (Frame Error, Parity Error, Overrun Error) occurs (SIDISR.ERI = 1). 0: Do not signal reception error interrupts. 1: Signal reception error interrupts.	R/W
10:9	CTSAC	CTSS Active Condition	CTSS Active Condition (Default: 00) This field specifies status change interrupt request conditions using the CTS Status (CTSS) of the Status Change Interrupt Status Register. 00: Do not detect CTS signal changes. 01: Rising edge of the CTS pin 10: Falling edge of the CTS pin 11: Both edges of the CTS pin CTS pin is supported only for SIO0.	R/W
8:6		Reserved		—
5:0	STIE	Status Change Interrupt Enable	Status Change Interrupt Enable (Default: 0x00) This field sets the set conditions of the Status Change bit (STIS) of the DMA/Interrupt Status Register (SIDISR). The condition is selected depending on which bit of the Status Change Interrupt Status Register (SISCISR) is set. (Multiple selections are possible.) An SIO interrupt is asserted when STIC is “1”. 000000: Do not detect status changes. 1*****: Set “1” to STIS when the Overrun bit (OERS) is “1”. *1****: Set “1” to STIS when a change occurs in a condition set by the CTSS Active Condition field (CTSAC) in the CTS Status bit (CTSS). **1***: Set “1” to STIS when the Break bit (RBRKD) becomes “1”. ***1**: Set “1” to STIS when the Transmit Data Empty bit (TRDY) becomes “1”. ****1*: Set “1” to STIS when the Transmission Complete bit (TXALS) becomes “1”. *****1: Set “1” to STIS when the Break Detection bit (UBRKD) becomes “1”.	R/W

19.4.3. DMA/Interrupt Status Register 0,1,2,3

(SIDISR0, SIDISR1, SIDISR2, SIDISR3)

Table 19-10 Address offsets for DMA/Interrupt Status Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF308	SIDISR0	DMA/Interrupt Status Register 0
SIO1	0xF408	SIDISR1	DMA/Interrupt Status Register 1
SIO2	0xF388	SIDISR2	DMA/Interrupt Status Register 2
SIO3	0xF488	SIDISR3	DMA/Interrupt Status Register 3

This register indicates the DMA or interrupts status information.

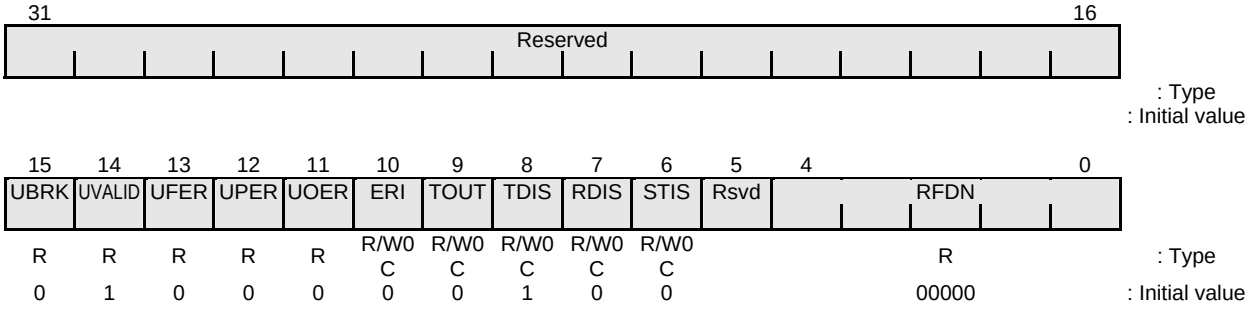


Figure 19-8 DMA/Interrupt Status Register

Table 19-11 DMA/Interrupt Status Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:16		Reserved		—
15	UBRK	Receive Break	SIO Break (Default: 0) This field indicates the break reception status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: No breaks 1: Detect breaks	R
14	UVALID	Receive FIFO Available Status	SIO Available Data (Default: 1) This field indicates whether or not data exists in the Receive FIFO (SIRFIFO). 0: Data exists in the Receive FIFO. 1: No data exists in the Receive FIFO.	R
13	UFER	Frame Error	SIO Frame Error (Default: 0) This field indicates the frame error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no frame errors. 1: There are frame errors.	R
12	UPER	Parity Error	SIO Parity Error (Default: 0) This field indicates the parity error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no parity errors. 1: There are parity errors.	R
11	UOER	Overrun Error	SIO Overrun Error (Default: 0) This register indicates the overrun status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no overrun errors. 1: There are overrun errors.	R
10	ERI	Reception Error Interrupt	Receive Data Error Interrupt (Default: 0) This bit is immediately set to “1” when a reception error (Frame Error, Parity Error, or Overrun Error) is detected.	R/W0C
9	TOUT	Reception Time Out	Time Out (Default: 0) This bit is set to “1” when a reception time out occurs.	R/W0C
8	TDIS	Transmission Data Empty	Transmit DMA/Interrupt Status (Default: 1) This bit is set when available space of the amount set by the Transmit FIFO Request Trigger Level (TDIL) of the FIFO Control Register (SIFCR) exists in the Transmit FIFO.	R/W0C
7	RDIS	Reception Data Full	Receive DMA/Interrupt Status (Default: 0) This bit is set when valid data of the amount set by the Receive FIFO Request Trigger Level (RDIL) of the FIFO Control register (SIFCR) is stored in the Receive FIFO.	R/W0C
6	STIS	Status Change	Status Change Interrupt Status (Default: 0) This bit is set when at least one of the interrupt statuses selected by the Status Change Interrupt Condition field (STIE) of the DMA/Interrupt Control Register (SIDICR) becomes “1”.	R/W0C
5		Reserved		—
4:0	RFDN	Reception Data Stage Status	Receive FIFO Data Number (Default: 00000) This field indicates how many stages of reception data remain in the Receive FIFO (0 – 16 stages).	R

19.4.4. Status Change Interrupt Status Register 0,1,2,3

(SISCISR0, SISCISR1, SISCISR2, SISCISR3)

Table 19-12 Address offsets for Status Change Interrupt Status Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF30C	SISCISR0	Status Change Interrupt Status Register 0
SIO1	0xF40C	SISCISR1	Status Change Interrupt Status Register 1
SIO2	0xF38C	SISCISR2	Status Change Interrupt Status Register 2
SIO3	0xF48C	SISCISR3	Status Change Interrupt Status Register 3

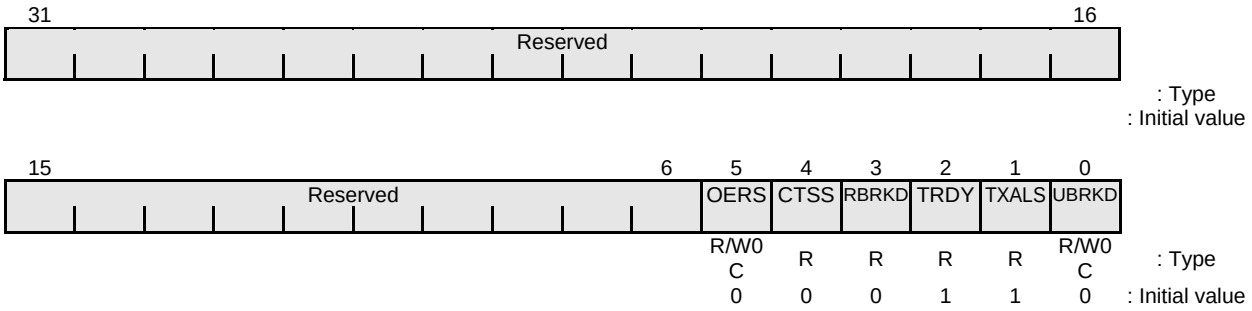


Figure 19-9 Status Change Interrupt Status Register

Table 19-13 Status Change Interrupt Status Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:6	—	Reserved	—	—
5	OERS	Overrun Error	Overrun Error Status (Default: 0) This bit is immediately set to “1” when an overrun error is detected. This bit is cleared when a “0” is written.	R/W0C
4	CTSS	CTS Status	CTS Terminal Status (Default: 0) This field indicates the status of the CTS signal. 1: The CTS signal is High. 0: The CTS signal is Low. This field is supported by SIO0 only.	R
3	RBRKD	Receiving Break	Receive Break (Default: 0) This bit is set when a break is detected. This bit is automatically cleared when a frame that is not a break is received. 1: Current Status is Break. 0: Current Status is not Break.	R
2	TRDY	Transmission Data Empty	Transmit Ready (Default: 1) This bit is set to “1” if at least one stage in the Transmit FIFO is free.	R
1	TXALS	Transmission Complete	Transmit All Sent (Default: 1) This bit is set to “1” if the Transmit FIFO and all transmission shift registers are empty.	R
0	UBRKD	Break Detected	SIO Break Detect (Default: 0) This bit is set when a break is detected. Once set, this bit remains set until cleared by writing a “0” to it.	R/W0C

19.4.5. FIFO Control Register 0,1,2,3

(SIFCR0, SIFCR1, SIFCR2, SIFCR3)

Table 19-14 Address offsets for Fifo Control Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF310	SIFCR0	Fifo Control Register 0
SIO1	0xF410	SIFCR1	Fifo Control Register 1
SIO2	0xF390	SIFCR2	Fifo Control Register 2
SIO3	0xF490	SIFCR3	Fifo Control Register 3

This register controls of the Transmit/Receive FIFO buffer.

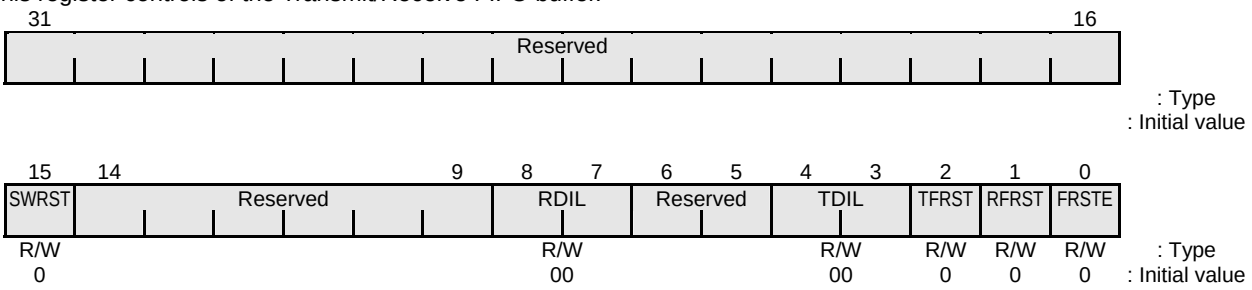


Figure 19-10 FIFO Control Register

Table 19-15 FIFO Control Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:16	—	Reserved	—	—
15	SWRST	Software Reset	Software Reset (Default: 0) This field performs SIO resets except for the FIFOs. Setting this bit to “1” initiates the reset. Set registers are also initialized. This bit returns to “0” when initialization is complete. 0: Normal operation 1: SIO software reset	R/W
14:9	—	Reserved	—	—
8:7	RDIL	Receive FIFO Request Trigger Level	Receive FIFO DMA/Interrupt Trigger Level (Default: 00) This register sets the level for reception data transfer from the Receive FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: 12 Bytes	R/W
6:5	—	Reserved	—	—
4:3	TDIL	Transmit FIFO Request Trigger Level	Transmit FIFO DMA/Interrupt Trigger Level (Default: 00) This register sets the level for transmission data transfer to the Transmit FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: Setting disabled	R/W
2	TFRST	Transmit FIFO Reset	Transmit FIFO Reset (Default: 0) The Transmit FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Software has to clear this bit to start normal operation. 0: During operation 1: Reset Transmit FIFO	R/W
1	RFRST	Receive FIFO Reset	Receive FIFO Reset (Default: 0) The Receive FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Software has to clear this bit to start normal operation. 0: During operation 1: Reset Receive FIFO	R/W
0	FRSTE	FIFO Reset Enable	FIFO Reset Enable (Default: 0) This field is the Reset Enable for the Transmit/Receive FIFO buffer. The FIFO is reset by combining the Transmit FIFO Reset bit (TFRST) and Receive FIFO Reset bit (RRST). 0: During operation 1: Reset Enable	R/W

19.4.6. Flow Control Register 0,1,2,3

(SIFLCR0, SIFLCR1, SIFLCR2, SIFLCR3)

Table 19-16 Address offsets for Flow Control Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF314	SIFLCR0	Flow Control Register 0
SIO1	0xF414	SIFLCR1	Flow Control Register 1
SIO2	0xF394	SIFLCR2	Flow Control Register 2
SIO3	0xF494	SIFLCR3	Flow Control Register 3

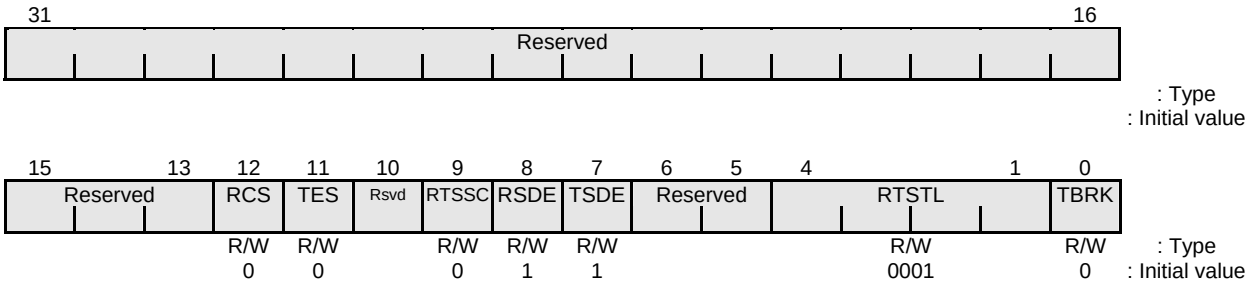


Figure 19-11 Flow Control Register

Table 19-17 Flow Control Register

Bit	Mnemonic	Field Name	Description	R/W
31:13		Reserved		—
12	RCS	RTS Signal Control Select	RTS Control Select (Default: 0) This field sets the reception flow control using RTS output signals. 0: Disable flow control using RTS signals. 1: Enable flow control using RTS signals. This field is supported by SIO0 only.	R/W
11	TES	CTS Signal Control Select	CTS Control Select (Default: 0) This field sets the transmission flow control using CTS input signals. 0: Disable flow control using CTS signals. 1: Enable flow control using CTS signals. This field is supported by SIO0 only.	R/W
10		Reserved		—
9	RTSSC	RTS Software Control	RTS Software Control (Default: 0) This register is used for software control of RTS output signals. 0: Set the RTS signal to Low (can receive data). 1: Sets the RTS signal to High (transmission pause request) This field is supported by SIO0 only.	R/W
8	RSDE	Serial Data Reception Disable	Receive Serial Data Disable (Default: 1) This is the Serial Data Disable bit. When this bit is cleared, data reception starts after the start bit is detected. The RTS signal will not become High even if this bit is cleared. (This field is supported by SIO0 only**)** 0: Enable (can receive data) 1: Disable (halt reception)	R/W
7	TSDE	Serial Data Transmit Disable	Transmit Serial Data Disable (Default: 1) This is the Serial Data Transmission Disable bit. When this bit is cleared, data transmission starts. When set, transmission stops after completing transmission of the current frame. 0: Enable (can transmit data) 1: Disable (halt transmission)	R/W
6:5		Reserved		—
4:1	RTSTL	RTS Active Trigger Level	RTS Trigger Level (Default: 0001) The RTS hardware control assert level is set by the reception data stage count of the Receive FIFO. 0000: Disable setting 0001: 1 : 1111: 15 This field is supported by SIO0 only.	R/W
0	TBRK	Break Transmission	Break Transmit (Default: 0) Transmits a break. The TXD signal is Low while TBRK is set to “1”. 0: Disable (clear break) 1: Enable (transmit break)	R/W

19.4.7. Baud Rate Control Register 0,1,2,3

(SIBGR0, SIBGR1, SIBGR2, SIBGR3)

Table 19-18 Address offsets for Baud Rate Control Register in TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF318	SIBGR0	Baud Rate Control Register 0
SIO1	0xF418	SIBGR1	Baud Rate Control Register 1
SIO2	0xF398	SIBGR2	Baud Rate Control Register 2
SIO3	0xF498	SIBGR3	Baud Rate Control Register 3

This register selects the clock that is provided to the baud rate generator and set the divide value.

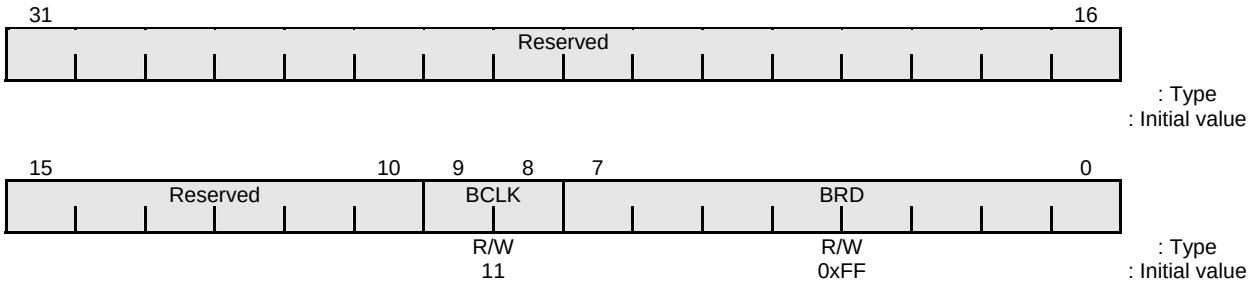


Figure 19-12 Baud Rate Control Register

Table 19-19 Baud Rate Control Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:10		Reserved		—
9:8	BCLK	Baud Rate Generator Clock	Baud Rate Generator Clock (Default: 11) This field sets the input clock for the baud rate generator. 00: Select prescalar output T0 (IMBUSCLK/2) 01: Select prescalar output T2 (IMBUSCLK/8) 10: Select prescalar output T4 (IMBUSCLK/32) 11: Select prescalar output T6 (IMBUSCLK/128)	R/W
7:0	BRD	Baud Rate Divide Value	Baud Rate Divide Value (Default: 0xFF) This field set divide value BRG of the baud rate generator. This value is expressed as a binary value.	R/W

19.4.8. Transmit FIFO Register 0,1,2,3

(SITFIFO0, SITFIFO1, SITFIFO2, SITFIFO3)

Table 19-20 Address offsets for Transmit FIFO Register in the TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF31C	SITFIFO0	Transmit Fifo Register 0
SIO1	0xF41C	SITFIFO1	Transmit Fifo Register 1
SIO2	0xF39C	SITFIFO2	Transmit Fifo Register 2
SIO3	0xF49C	SITFIFO3	Transmit Fifo Register 3

When using the DMA Controller to perform DMA transmission, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

Little Endian: 0xF31C (Ch.0), 0xF35C (Ch.1), 0xF39C (Ch.2), 0xF3DC (Ch.3)
Big Endian: 0xF31F (Ch.0), 0xF35F (Ch.1), 0xF39F (Ch.2), 0xF3DF (Ch.3)

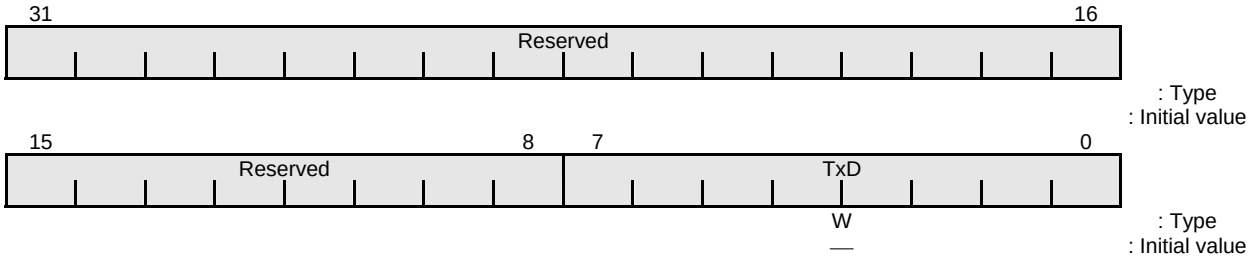


Figure 19-13 Transmit FIFO Register

Table 19-21 Transmit FIFO Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	TxD	Transmission Data	Transmit Data Data written to this register are written to the Transmit FIFO.	W

19.4.9. Receive FIFO Register 0,1,2,3

(SIRFIFO0, SIRFIFO1, SIRFIFO2, SIRFIFO3)

Table 19-22 Address offsets for Receive FIFO Register in the TX4939

Channel	Address Offset	Mnemonic	Register Name
SIO0	0xF320	SIRFIFO0	Receive Fifo Register 0
SIO1	0xF420	SIRFIFO1	Receive Fifo Register 1
SIO2	0xF3A0	SIRFIFO2	Receive Fifo Register 2
SIO3	0xF4A0	SIRFIFO3	Receive Fifo Register 3

When using the DMA Controller to perform DMA reception, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

Little Endian: 0xF320 (Ch.0), 0xF360 (Ch.1), 0xF3A0 (Ch.2), 0xF3E0 (Ch.3)
Big Endian: 0xF323 (Ch.0), 0xF363 (Ch.1), 0xF3A3 (Ch.2), 0xF3E3 (Ch.3)

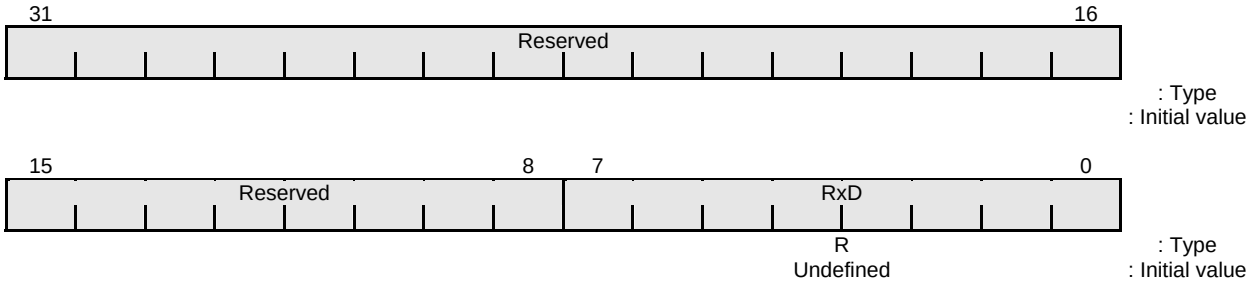


Figure 19-14 Receive FIFO Register

Table 19-23 Receive FIFO Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:8		Reserved		—
7:0	RxD	Reception Data	Receive Data This field reads reception data from the Receive FIFO. Reading this register updates the Reception Data Status.	R

Chapter 20. SPI Interface

20.1. Features

The SPI is a serial interface that consists of clock, data output, and data input. The SPI is used to interface with serial power, serial A/D converters, other devices including simple serial clocks and data interfaces. The TX4939 only operates as a Master. It generates SPI clocks to Slaves. Multi-slave devices can share the SPI by using a unique Chip Select for each Slave device. TX4939 SPI provides a dedicated Chip Select port to access a SPI device.

To access more than one SPI device, the system needs to use other output ports to generate unique Chip Selects. If the Chip Select of a device is asserted and the device is selected, that device uses the SPICLK and SPIOUT signals to shift data in, and then uses the SPIIN signal to shift data out. If the device is not selected, the data output connected to SPIIN must be put into the tri-state and other devices must be able to share the SPIIN signal.

The SPI module contains registers that can program the SPI CLK rate, MSB first or LSB first, clock polarity, data phase polarity, and Byte mode or Word mode operation.

The SPI module has the following characteristics:

- *Selectable clock phase and polarity*
- *Transfer Size: 8-bit or 16-bit*
- *4-frame Transmitter Buffer and 4-frame Receiver Buffer*
- *Master Operation*
- *Interframe Delay Time Counter*
- *MSB/LSB First*

20.2. Block diagram

The SPI Module mainly consists of a 16-bit SPI Data Register (SPDR), a 16-bit Transmitter Buffer, a 16-bit Receiver Buffer, a 16-bit Shift Register, a Baud Rate Generator, an Interframe Delay Time counter, and interrupt logic. Figure 20-1 is a block diagram of the SPI Module.

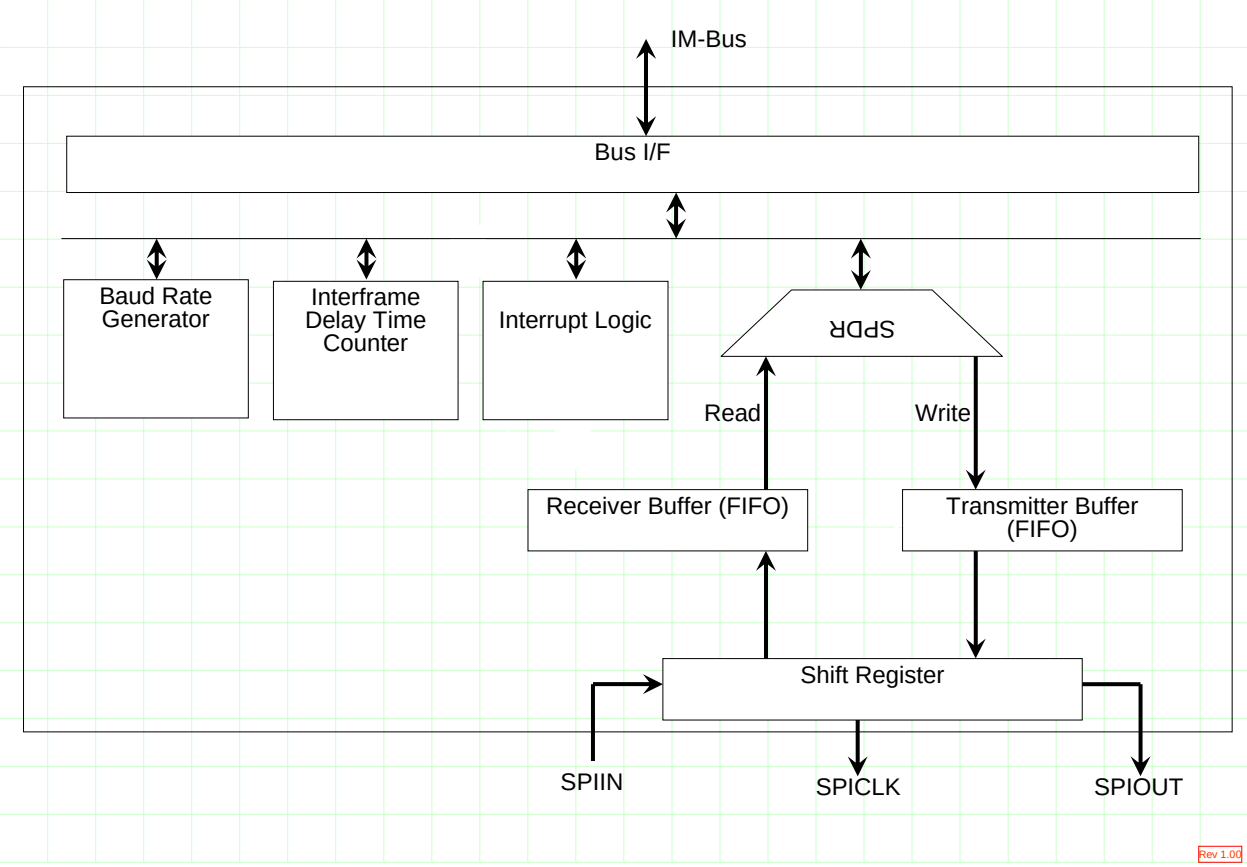


Figure 20-1 SPI Block Diagram

20.3. Operational description

SPI interface ports in TX4939 are shared with SIO2, SIO3 or GPIO ports. The active interface ports are selected by writing to Pin Configuration Register (PCFG) described in Chapter 7. Configuration Registers after boot up. Table 20-1 gives PCFG settings that are needed to select SPI ports as the active interface.

Table 20-1 Selecting SPI as the active interface port in TX4939

PCFG[61,57:56]	Active Interface	SPI Interface
3'b011	SPI	Usable
Others	GPIO or SIO2 orSIO3	Unusable

Note: PCFG[61,57:56] defaults to GPIO interface after boot up. The behavior of SPI is not defined when SPI is accessed through the IMBUS and SPI interface port is not selected in PCFG.

The following sections give a detailed description of SPI operations.

20.3.1. Operation modes

The SPI Module has the two following operation modes:

Configuration Mode (OPMODE = "01"):
You can only rewrite the lower byte (bits[7:0]) of SPI Control Register 0 (SPCR0) and all bits of SPI Control Register 1 (SPCR1) when in this mode. Also, the SPSTP bit, Receiver FIFO and Transmitter FIFO are cleared and the SPI Module is reset when in this mode. Setting this mode forcibly terminates even the transfer of a frame that is currently in progress.

Active Mode (OPMODE = "10")
The module operates in this mode during normal operation. You can execute transfers when in this mode.

Sleep mode (OPMODE = '11') : In this mode, the SPI module will be set in reset state.

20.3.2. Transmitter/Receiver

The SPI Module is in the Reset state when it is in the Configuration Mode. When in this mode, set the lower byte (bits[7:0]) of SPI Control Register 0 (SPCR0) and SPI Control Register 1 (SPCR1) to the desired value before changing the operation mode to the Active Mode. The SPI Module can start transferring data once it is in the Active Mode. Transfer starts when data is written to the SPI Data Register (SPDR).

The data written to the SPI Data Register (SPDR) is sent to the Shift Register, which then outputs the data to the Slave device. When the data is outputted from the SPIOUT pin, it is simultaneously fetched from the SPIIN pin. When fetching of the data is complete, the content of the Shift Register is loaded into the Receiver Buffer, the SRRDY bit of the SPI Status Register (SPSR) becomes "1", then the Reception Buffer is notified that there is reception data. When the RBSI bit of the SPI Status Register (SPSR) is set to "1", an interrupt occurs when the accumulated reception data reaches the level set by the RXIFL bit of SPI Control Register 0 (SPCR0).

When the content of the Transmitter Buffer is transferred to the Shift Register, the STRDY bit of the SPI Status Register (SPSR) becomes "1" and notification is sent that the Transmitter Buffer is available for use again. When the TBSI bit of the SPI Status Register (SPSR) is set to "1", an interrupt occurs when the accumulated transmission data reaches the level set by the TXIFL bit of SPI Control Register 0 (SPCR0). Therefore, the software executes the following steps each time it writes data to the Transmitter Buffer.

(1) Check whether the STRDY bit or the TBSI bit is "1". If neither bit is "1", wait until one of them becomes "1".

(2) Write data to the SPI Data Register (SPDR).

In this way, depending on the software used, the SPI Module can continue to seamlessly transmit data as long as the Transmitter Buffer is in the Run state until the data is shifted out from the Shift Register. If the software cannot keep up with the transfer rate, the SPI Module waits until the next data is written to the SPI Data Register (SPDR).

When one series of transmission is complete, the software deasserts the Chip Select signal of the Target device according to the following procedure.

(1) Check whether the SRRDY bit or the RBSI bit is “1”. If neither bit is “1”, wait until one of them is asserted.

(2) Check whether SIDLE is “1”. If it is not “1”, wait until it is asserted.

(3) Deassert the Chip Select signal.

The SPI Module supports either 8-bit or 16-bit per character operation as defined by the SSZ bit of SPI Control Register 1 (SPCR1). Also, the software can use the SBOS bit of SPI Control Register 0 (SPCR0) to select whether to shift MSB or LSB first. The other combination of control bits (SPHA and SPOL) determines the transfer format. See 0 for the transfer format.

20.3.3. Baud Rate Generator

The SPI Module operates IMBUSCLK as the Master Clock (SPI Master Clock). This module divides IMBUSCLK to generate (using the Baud Rate Generator) the SPICLK used for SPI transmission/reception. You can set this divide rate and can change the SPICLK to any frequency. Use SER[7:0] of SPI Control Register 1 (SPCR1) to set the divide rate. The following table shows the divide rate (SER[7:0]) and SPICLK frequency for when IMBUSCLK is 100 MHz or 83.25 MHz. (IMBUSCLK frequency is calculated as ¼ of the CPUCLK frequency).

Table 20-2 SPICLK Frequency

SER[7:0]	SPI Clock Rate (IMBUSCLK: 100 MHz)	SPI Clock Rate (IMBUSCLK: 83.25 MHz)
00000001b	25.00 MHz	20.81 MHz
00000010b	16.67 MHz	13.88 MHz
00000011b	12.50 MHz	10.41 MHz
00000100b	10.00 MHz	8.33 MHz
00000101b	8.33 MHz	6.94 MHz
…		
00001001b	5.00 MHz	4.16 MHz
…		
00010011b	2.50 MHz	2.08 MHz
…		
11111111b	195.31 KHz	162.60 KHz

20.3.4. Transfer format

During SPI transfer, serial data transmission (shift out) and reception (shift in) is executed simultaneously. At this time, serial data is shifted or sampled synchronous to the serial clock.
The SPHA bit and SPOL bit of SPI Control Register 0 (SPCR0) determine the transfer format. The setting of the SPHA bit broadly classifies the protocol that is used. Each protocol is described below in items 20.3.4.1 and 0.

20.3.4.1. SPHA = 0 format

Figure 20-2 shows the transfer format when SPHA is “0”.

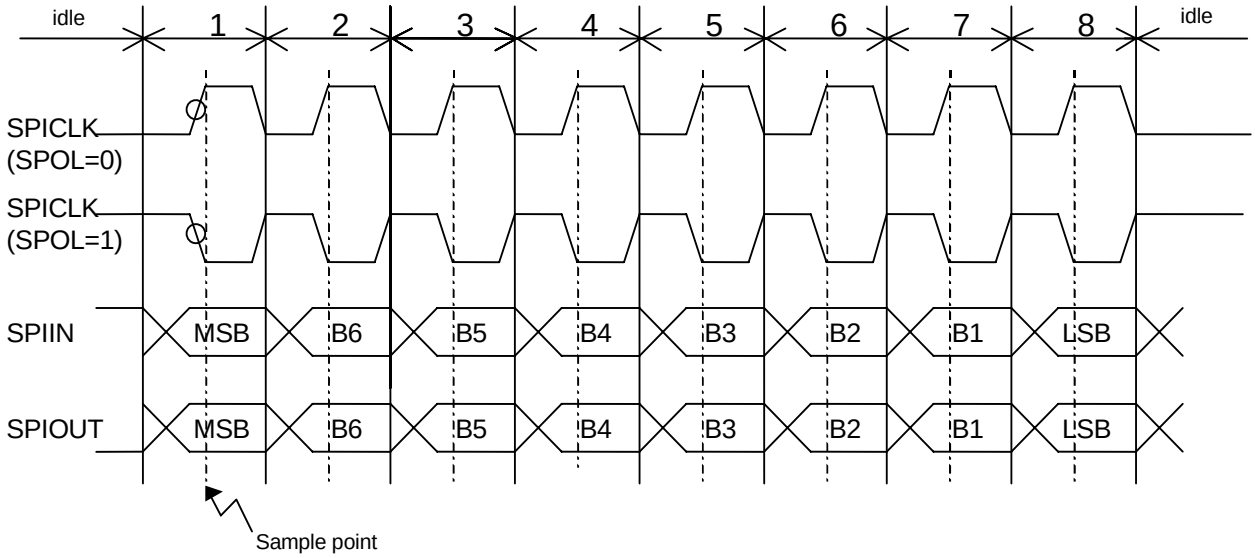


Figure 20-2 Transfer Format when SPHA is “0”

When in this format, data is sequentially fetched from the first clock edge after the Idle state. When the SPOL bit is “0”, data is fetched at the rising edge. When this bit is “1”, data is fetched at the falling edge. The SPIIN or SPIOUT signal is switched at the second clock edge of SPICLK. The signal is switched at the SPICLK falling edge when the SPOL bit is “0”, or is switched at the SPICLK rising edge when the SPOL bit is “1”. When the SPOL bit is “0”, SPICLK is at the Low level during the Idle state. When the SPOL bit is “1”, SPICLK is at the High level during the Idle state.

20.3.4.2. SPHA = 1 format

Figure 20-3 shows the transfer format when SPHA is “1”.

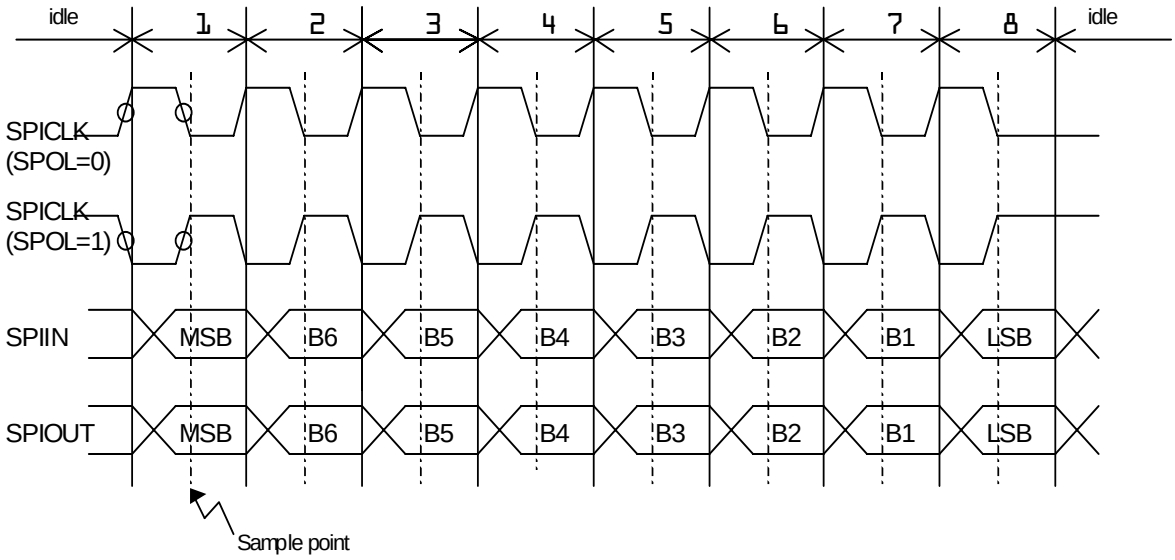


Figure 20-3 Transfer Format when SPHA is “1”

When in this format, data is sequentially fetched from the second clock edge after the Idle state. When the SPOL bit is “0”, the data is fetched from the second falling edge. When the SPOL bit is “1”, the data is fetched from the second rising edge. When the SPOL bit is “0”, SPICLK is at the Low level during the Idle state. When the SPOL bit is “1”, SPICLK is at the High level during the Idle state.

20.3.5. Interframe Delay Time Counter

There are cases where it is preferable to shorten the time between data groups. In such cases, the Interframe Delay Time Counter is used to specify the delay time between data groups. When 16 bits is selected as the data size by SPI Control Register 1, the delay time is inserted after 16-bit data is shifted. When 8 bits is selected, the delay time is inserted after 8-bit data is shifted. When IFS bits [9:0] are set to a value other than “0”, a delay is inserted between the characters. The length of the delay inserted between characters is also changed by the IFSPSE bit of SPI Control Register 0 (SPCR0) as shown below.

When SPCR0.IFSPSE=1: SPI Master Clock Cycle × IFS[9:0] × 32
When SPCR0.IFSPSE=0: SPI Master Clock Cycle × IFS[9:0]

If the IFS value is “0”, seamless operation is performed. The SPI Module continues to shift data and supply a clock as long as the software does not fall behind the transfer rate of the transmitter.

20.3.6. Buffer configuration

The SPI Module has a Transmitter Buffer and a Receiver Buffer. The buffers use FIFOs to store data. Each FIFO can store 4-frame data.

The Transmitter Buffer stores the value written to the SPI Data Register. Either when in the Idle state or when the current transmission is complete, the first data written to the Transmitter Buffer is transferred to the Shift Register.

On the other hand, data received in the Shift Register is stored in the Receiver Buffer each time transfer ends.
You can issue interrupts each time data equal in size to the data in the buffer accumulates.

20.3.7. SPI system errors

The SPI Module signals the following system errors during transfer.

20.3.7.1. Overrun error (SPOE)

An overrun error is issued when an attempt is made to write the next data to the Transmitter Buffer regardless of whether the Transmitter Buffer is full. The data to be newly written at this time is not written to the Transmitter Buffer. Also, the SPOE bit of the SPSR Register becomes “1”.

20.3.8. Interrupts

The SPI Module has three types of interrupt sources. The result of OR operation performed on three interrupt sources is inputted to the Interrupt Controller (IRC) as SPI interrupts. Check the SPI Status Register (SPSR) to see which interrupts occurred.

Type	Status bits	Mask-able bit
System error or idle	SPOE, SIDLE	SOEIE, SILIE
Receive Buffer Fill	RBSI	RBSIE
Transmit Buffer Fill	TBSI	TBSIE

Use System errors or Idle interrupts for error detection or Idle state interrupts. Use Receiver Buffer Fill interrupts and Transmitter Buffer Fill interrupts when setting new transmission data to the buffer or when reading receiver data from the buffer.

20.4. Registers

Access SPI Module registers using 32-bit access. Operation is not guaranteed when accessing these registers using any other access size. Do not write “0” to any undefined bits.

Table 20-3 SPI Module Registers

Offset Address	Bit Width	Register Symbol	Register Name
0xF800	32	SPMCR	SPI Master Control Register
0xF804	32	SPCR0	SPI Control Register 0
0xF808	32	SPCR1	SPI Control Register 1
0xF80C	32	SPFS	SPI Interframe Delay Time Register
0xF810	32	—	(Reserved)
0xF814	32	SPSR	SPI Status Register
0xF818	32	SPDR	SPI Data Register
0xF81C	32	—	(Reserved)

20.4.1. SPI Master Control Register (SPMCR) 0xF800

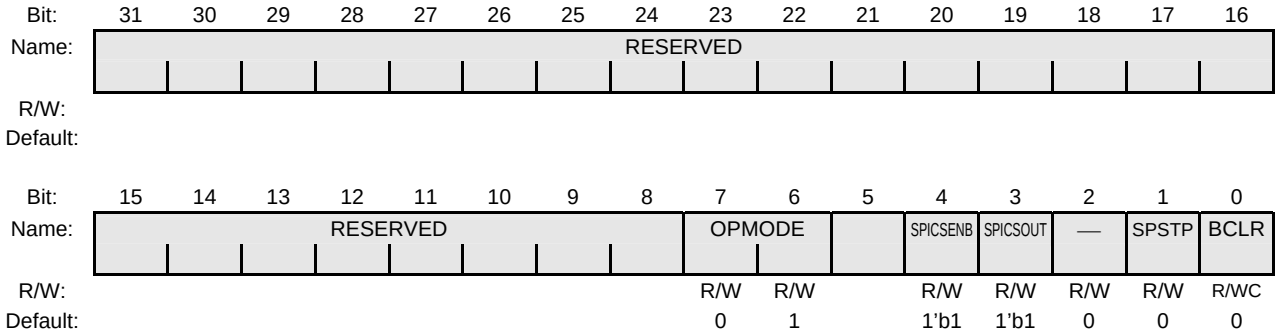


Figure 20-4 SPI Master Control Register (SPMCR)

Table 20-4 SPI Master Control Register (SPMCR)

Bit(s)	Mnemonic	Field Name	Description
31 : 8	—	Reserved	—
7 : 6	OPMODE	Operation Mode	Operation Mode (Default: 01) Sets the operation mode. 00: Don't care. 01: Configuration Mode 10: Active Mode 11: Sleep Mode
5	—	Reserved	—
4	SPICSENB	Chip Select Enable	Chip Select Enable (Default 1'b1) 1'b1: Chip Select (SPICSOUT) disabled 1'b0: Chip Select (SPICSOUT) enabled
3	SPICSOUT	Chip Select	Chip Select (Default: 1'b1) 1'b1: Active high Chip Select 1'b0: Active low Chip Select
2	—	Reserved	Do not write "1" to this bit.
1	SPSTP	SPI Stop	SPI Stop (Default: 0) When "1" is written to this bit, the SPI Module does not transfer any more data after the current frame is complete. You can only set this bit when in the Active Mode. Entering the Configuration Mode clears this bit. 0: Normal operation 1: Halt after the current transfer ends.
0	BCLR	SPI Buffer Clear	SPI Buffer Clear (Default: 0) Use this bit to clear the Transmitter FIFO and Receiver FIFO. Writing "1" to this bit initializes the FIFO. Write "1" to this bit after the SPI Module enters the Idle state (SIDLE=1). BCLR can be asserted only if SPSTP has already been asserted. Always outputs "0" when read. Write: 0: Don't care 1:Clear the FIFO

20.4.2. SPI Control Register 0 (SPCR0) 0xF804

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name:	RESERVED															
R/W:																
Default:																

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name:	TXIFL		RXIFL		SILIE	SOEIE	RBSIE	TBSIE	0	0	0	IFSPSE	1	SBOS	SPHA	SPOL
R/W:	R/W		R/W		R/W	R/W	R/W	R/W	R/O	R/O	R/O	R/W	R/O	R/W	R/W	R/W
Default:	0		0		0	0	0	0	0	0	0	0	0	0	0	0

Figure 20-5 SPI Control Register 0 (SPCR0)

Table 20-5 SPI Control Register 0 (SPCR0)

Bit(s)	Mnemonic	Field Name	Description
31 : 16	—	Reserved	—
15 : 14	TXIFL	Transmit Interrupt Fill Level	Transmit Interrupt Fill Level (Default: 00) You can issue interrupts according to the number of free space in the Transmitter FIFO. This field sets the free space count of the Transmitter FIFO at which interrupts are issued. 00: Issue an interrupt when there is 1 or more free space in the Transmitter FIFO. 01: Issue an interrupt when there are 2 or more free spaces in the Transmitter FIFO. 10: Issue an interrupt when there are 3 or more free spaces in the Transmitter FIFO. 11: Issue an interrupt when there are 4 free spaces in the Transmitter FIFO.
13 : 12	RXIFL	Receive Interrupt Fill Level	Receive Interrupt Fill Level (Default: 00) You can issue interrupts according to the number of data in the Receiver FIFO. This field sets the number of Receiver FIFO data at which to issue an interrupt. 00: Issue an interrupt when there is 1 or more data in the Receiver FIFO. 01: Issue an interrupt when there are 2 or more data in the Receiver FIFO. 10: Issue an interrupt when there are 3 or more data in the Receiver FIFO. 11: Issue an interrupt when there are 4 data in the Receiver FIFO.
11	SILIE	SPI IDLE Interrupt Enable	SPI IDLE Interrupt Enable (Default: 0) Enables SPI Idle interrupts. 0: Disable 1: Enable
10	SOEIE	SPI Overrun Interrupt Enable	SPI IDLE Overrun Enable (Default: 0) Enables SPI Overrun interrupts. 0: Disable 1: Enable
9	RBSIE	Receive Buffer Fill Interrupt Enable	Receive Buffer Fill Interrupt Enable (Default: 0) This bit specifies whether to signal an interrupt to the Interrupt Controller based on the number of data actually in the Receiver FIFO or to only display it as a status. 0: Disable (mask) 1: Enable
8	TBSIE	Transmit Buffer Fill Interrupt Enable	Transmit Buffer Fill Interrupt Enable (Default: 0) This bit specifies whether to signal an interrupt to the Interrupt Controller based on the number of free data actually in the Transmitter FIFO or to only display it as a status. 0: Disable (mask) 1: Enable
7:5	—	Reserved	—
4	IFSPSE	Inter Frame Space prescaler enable	Inter Frame Space prescaler Enable (Default: 0) Enables prescaler of the Interframe Delay Time Counter. 0: Disable (× 1) 1: Enable (× 32)
3	—	Reserved	—
2	SBOS	SPI Bit Order Select	SPI Bit Order Select (Default: 0) This bit specifies the bit order of the transfer data.

Table 20-5 SPI Control Register 0 (SPCR0)

Bit(s)	Mnemonic	Field Name	Description
			0: LSB first (transfer starting from the least significant bit) 1: MSB first (transfer starting from the most significant bit)
1	SPHA	SPI Phase	SPI Clock Phase (Default: 0) Selects the clock phase. 0: Samples at the first clock edge, then shifts at the second edge. 1: Shifts at the first clock edge, then samples at the second edge.
0	SPOL	SPI Polarity	SPI Clock Polarity (Default: 0) Selects the SPICLK polarity. 0: High Active (SPICLK is Low when idle) 1: Low Active (SPICLK is High when idle)

Note 1: You can only write to bits 4, 2, 1, or 0 when the SPI Module is in the Configuration Mode.
Note 2: The SPOL and SPHA bits select the SPICLK phase and the clock edge at which to sample data. For details, see 20.3.4

20.4.3. SPI Control Register 1 (SPCR1) 0xF808

Figure 20-6 SPI Control Register 1 (SPCR1)

Table 20-6 SPI Control Register 1 (SPCR1)

Bits	Mnemonic	Field Name	Description
31 : 16	—	Reserved	—
15 : 8	SER	SPI Data Rate	SPI Data Rate (Default: 000000b) This field sets the transfer bit rate. The transfer bit rate is calculated according to the following equation. $f_{BR} = f_{SPI}/2\ (n + 1)$ f_{BR} : SPICLK Frequency f_{SPI} : SPI Master Clock Frequency n : SER (Setting “0” is not permitted) (See 20.3.3 for SER and clock frequency examples.)
7 : 5	—	Reserved	—
4 : 0	SSZ	SPI Transfer Size	SPI Transfer Size (Default: 00000b) Selects the transfer size. 0x08: 8 bits 0x10: 16 bits Other values: Reserved (not settings are permitted)

Note 1: You can only write to this register when the SPI Module is in the Configuration Mode.

20.4.4. SPI Interframe Delay Time Counter (SPFS) 0xF80C

Figure 20-7 SPI Interframe Delay Time Counter (SPFS)

Table 20-7 SPI Interframe Delay Time Counter (SPFS)

Bits	Mnemonic	Field Name	Description
31 : 10	—	Reserved	—
9 : 0	IFS	Inter Frame Space	Inter Frame Space (Default: 0x00) This register sets the delay time to insert between two consecutive frames. When this register is set to "0", transfer ends, the next transmission data is loaded from the Transmitter FIFO to the Transmitter Buffer, and then the next transfer is performed. (Since the time data is loaded into the buffer is not "0", the delay time from the last clock until the first clock of the next transfer is not actually "0".) When setting the IFSPSE bit of the SPCR0 Register to "0" and not using prescaler, the Interframe delay time is the value calculated using the following formula: (15 ns - 15.5 μs when SPI Master Clock Frequency is 66 MHz) When setting the IFSPSE bit of the SPCR0 Register to "1" and using prescalar, the Interframe delay time is the value calculated using the following formula: (480 ns - 496 μs when SPI Master Clock Frequency is 66 MHz) Only write to this register when the SIDLE bit is "0".

20.4.5. SPI Status Register (SPSR) 0xF814

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name:	RESERVED															
R/W:																
Default:																

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name:	TBSI	RBSI	TBS			RBS			SPOE	RESERVED			IFSD	SIDLE	STRDY	SRRDY
R/W:	R	R	R			R			R/C				R	R	R	R
Default:	1	0	000			000			0				0	1	1	0

Figure 20-8 SPI Status Register (SPSR)

Table 20-8 SPI Status Register (SPSR)

Bit(s)	Mnemonic	Field Name	Description
31 : 16	—	Reserved	—
15	TBSI	Transmit Buffer Status Indicator	Transmit Buffer Status Indicator (Default: 1) Indicates the status of interrupts according to the number of free data in the Transmitter FIFO. 0: No interrupt 1: Interrupt
14	RBSI	Receive Buffer Status Indicator	Receive Buffer Status Indicator (Default: 0) Indicates the status of interrupts according to the number of free data in the Receiver FIFO. 0: No interrupts 1: Interrupts
13 : 11	TBS	Transmit Buffer Status	Transmit Buffer Status (Default: 000) Indicates how many data are in the Transmit FIFO. 000: No data 001: 1 010: 2 011: 3 100: 4 (FIFO is full) 101 – 111: NA
10 : 8	RBS	Receive Buffer Status	Receive Buffer Status (Default: 000) Indicates how many data are in the Receiver FIFO. 000: No data 001: 1 010: 2 011: 3 100: 4 (FIFO is full) 101 – 111: NA
7	SPOE	SPI Overrun Error	SPI Overrun Error (Default: 0) Indicates that an overrun occurred in the Transmitter FIFO. You can clear this bit by writing "1" to it. You can also clear this flag by putting the SPI Module in the Configuration Mode. Read: 0: No errors 1: Generate overrun errors Write: 0: Don't care 1: Clear
6 : 4	—	Reserved	—

Table 20-8 SPI Status Register (SPSR)

Bit(s)	Mnemonic	Field Name	Description
3	IFSD	SPI Inter Frame Space Delay Indicator	SPI Inter Frame Space Delay Indicator (Default: 0) This bit becomes "1" when transfer of a frame ends and transfer of the next frame is deferred by the Interframe Delay Time Counter. 0: No interframe cycle 1: Interframe cycle
2	SIDLE	SPI Idle Indicator	SPI Idle Indicator (Default: 1) This bit becomes "1" either when the Transmitter FIFO is empty or the SPSTP bit is "1" and there is no transfer in progress. 0: Run 1: Idle
1	STRDY	SPI Transmit Ready	SPI Transmit Ready (Default: 1) Indicates that there is space in the Transmitter FIFO. 0: Transmitter FIFO is full. 1: Transmitter FIFO has space.
0	SRRDY	SPI Receive Ready	SPI Receive Ready (Default: 0) Indicates that there is reception data in the Receiver FIFO. This bit is cleared when the SPDR Register is read and there is no longer any valid data in the Receiver FIFO. 0: Receiver FIFO is empty. 1:There is data in the Receiver FIFO.

20.4.6. SPI Data Register (SPDR) 0xF818

Figure 20-9 SPI Data Register (SPDR)

Table 20-9 SPI Data Register (SPDR)

Bits	Mnemonic	Field Name	Description
31 : 16	—	Reserved	—
15 : 0	DR	SPI Data Register	SPI Data Register (Default: 0x00) The data written to this register is stored in the Transfer Buffer. When the Shift Register is empty, data is written to the Shift Register and transmission begins. You can read the Receiver FIFO data when you read this register. When the transfer data size is 8 bits, use the lower 8 bits of this register. When reading, the upper side (bits[15:8]) are "0".

Chapter 21. CIR Controller

21.1. Features

The TX4939 has an infrared Controller. This controller is interface to IMB bus protocol.

- *1 infrared channel*
- *Programmable sample period*
- *Interface to IMB bus protocol*
- *Supports interrupt*

21.2. Block Diagram

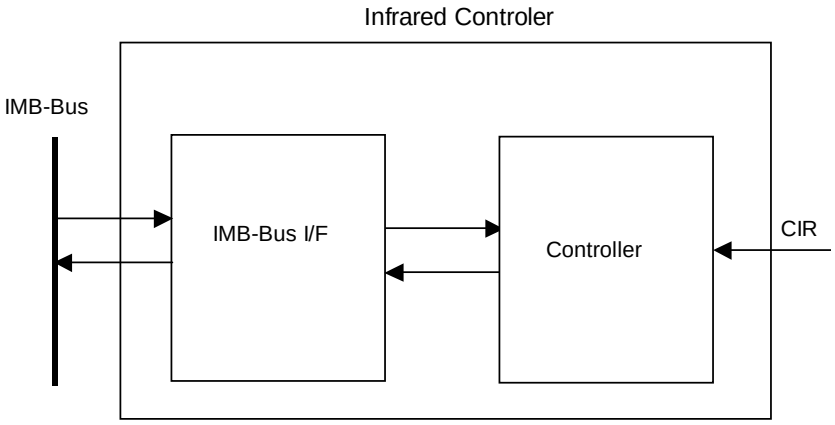


Figure 21-1 CIR Block Diagram

21.3. Functional Description

21.3.1. Theory of Operation

All infrared (IR) remote controls use some kind of IR signal. The remotes transmit pulses of IR light to send the signal. Demodulated signal connects to CIR signal of TX4939.

1) Pulse-Width-Coded Signals vary the length of pulses to code the information. If the pulse width is short then it corresponds to a logical zero or a low. If the pulse width is long then it corresponds to a logical one or a high.

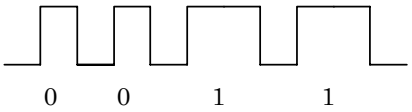


Figure 21-2 Pulse-Coded Signal

2) Space-Coded Signals vary the length of the spaces between pulses to code the information. In the Space-Coded Signals, if the space width is short then it corresponds to a logical zero or a low. If the space width is long then it corresponds to a logical one or a high.

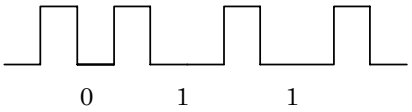


Figure 21-3 Space-Coded Signal

The CIR samples the HIGH time and LOW time of the infrared signal. It then records these count (high and low) value into registers. Upon L to H transition, it interrupt the CPU so that CPU can read these count value and determine if the code is “1” or “0”.

The Interrupt can be program to either trigger at L to H transition or H to L transition of IR signal (default is L to H transition).

The sample clock can be programmable. Refer to CIR register definition for more detail.

21.3.2. Signal Muxing

In TX4939, CIR signal is muxing with GPIO14/RXD3/SPIIN. In order to select CIR mode, firmware must select GPIO mode (SPIMODE = 1 and SIO3MODE = 1) and then select GPIO14 as input mode (GPIOIMR[14]=0, GPIODR[14]=1). Please refer to Chapter 7 Configuration for more detail.

21.4. Register Definition

21.4.1. CIR Control/Status Register (CIR_CSR) 0xFC00

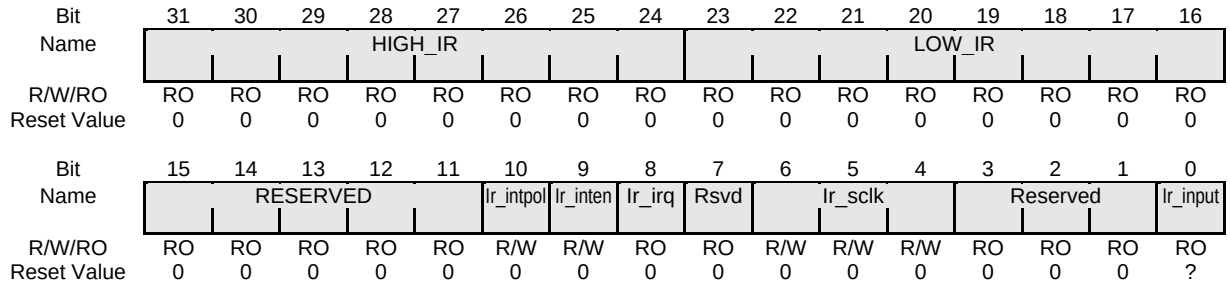


Figure 21-4 CIR Control/Status Register

Table 21-1 CIR Control/Status Register

Bit	Length	Field Name	Description
31:24	8	HIGH_IR	HIGH time count for Infrared Remote input. If Count value reaches 8'hff, it will remain at 8'hff. - If Ir_intpol = 0: This count register will get reload with new count value when there is a L to H transition on CIR signal. - If Ir_intpol = 1 then this count register will get reload with new count value when there is a H to L transition on CIR signal.
23:16	8	LOW_IR	LOW time count for Infrared Remote input If Count value reaches 8'hff, it will remain at 8'hff. - If Ir_intpol = 0: This count register will get reload with new count value when there is a L to H transition on CIR signal. - If Ir_intpol = 1 then this count register will get reload with new count value when there is a H to L transition on CIR signal.
15:11	5	Reserved	Reserved.
10	1	Ir_intpol	Infrared Interrupt option 0: Select L to H transition (default) 1: Select H to L transition
9	1	Ir_inten	Infrared Interrupt Enable 0: Disable (default) 1: Enable
8	1	Ir_irq	Infrared Remote Interrupt 0: No interrupt/not Valid 1: interrupt/Valid Reading this register will clear this bit. If Ir_inten is disable ("0"), firmware can treat this bit as valid bit. If this bit is "1" then the count value is valid.
7	1	---	Reserved
6:4	3	Ir_sclk	Infrared Remote Sample clock. Based on CLK4MHZ from clock module (sys_cg), please refer to clock spec. for more information about CLK4MHZ. 000: = 4.096 Mhz / 256 (default) 001: = 4.096 Mhz / 128 010: = 4.096 Mhz / 64 ---------- 110: = 4.096 Mhz / 4 111: = 4.096 Mhz / 2
3:1	3	---	Reserved
0	1	Ir_input	Infrared Remote Input signal

Chapter 22. I2C Controller

22.1. Overview

I2C Controller is an interface between IMBus and I2C protocol.

The basic features of the I2C are outlined below:

- *Interface to 32-bit IMBus.*
- *One I2C output.*
- *Single Master Operation.*
- *Support Byte mode.*
- *Interrupt or bit-polling driven data-transfers.*
- *Supports 7 and 10 bit addressing.*
- *Start/Stop/Repeated Start/Acknowledge generation.*
- *Software programmable clock frequency.*
- *GPIO mode.*

Figure 22-1 shows the block diagram of the I2C Interface module.

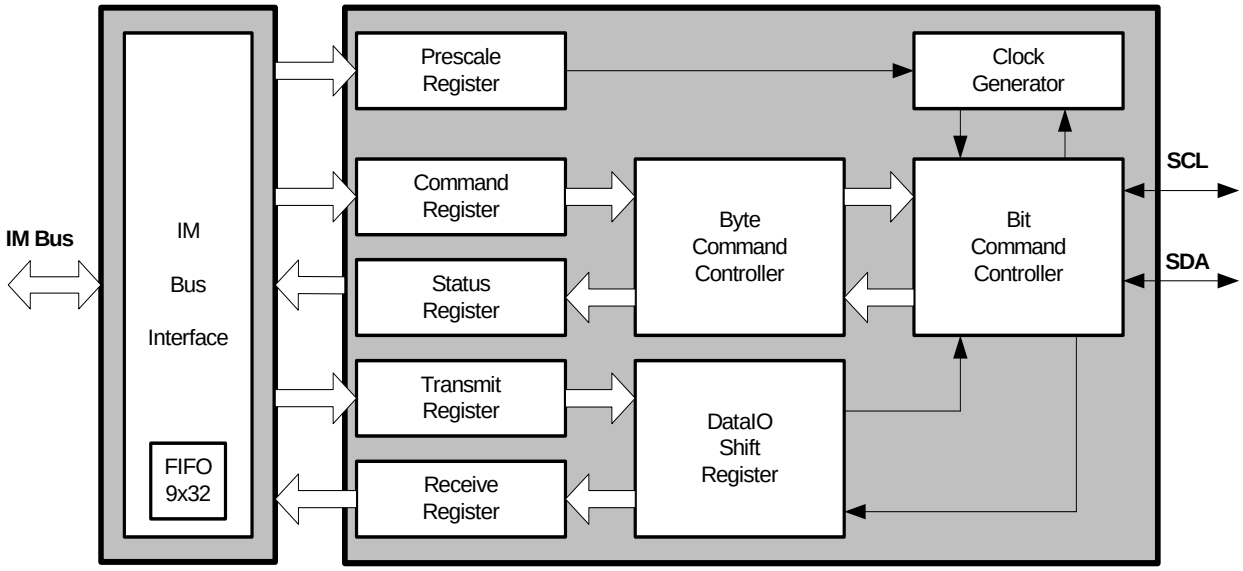


Figure 22-1 Block Diagram of I2C Controller

22.2. I2C Register Definition

All registers are 32 bits. Only word access is allowed. I.e. all instructions are LW or SW instructions. Other type of access may cause unexpected operation.

Offset	Register	Width	R / W	Description
0xF900	I2C_ICTSR	[31:0]	R/W	I2C Interrupt Control/Status Register
0xF904	I2C_TXRR	[31:0]	R/W	I2C Transmit/Receive Register
0xF908	I2C_CR	[31:0]	WO	I2C Command Register
0xF90C	I2C_PRE	[31:0]	R/W	I2C Prescale Register
0xF920	I2C_CTR	[31:0]	R/W	I2C Control Register

22.2.1. I2C Interrupt Control/Status Register (I2C_ICTSR) 0xF900

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		Field Name	Description
31:16		Reserved	
15:8		Reserved	
7		I2C_ASTS	I2C Ack Flag 0: Acknowledge received. 1: No Acknowledge received
6		I2C_DSTS	I2C Done flag 0: Idle/Done 1: BUSY
5		I2C_AINTFL	I2C Ack interrupt flag 0: Normal 1: Interrupt occur Write “1” to clear, write “0” has no affect.
4		I2C_DINTFL	I2C Done interrupt flag 0: Normal 1: Interrupt occur Write “1” to clear, write “0” has no affect
3:2		Reserved	
1		I2C_AEN	I2C Detect ACK interrupt enable 0: Disabled 1: Enabled When this bit set, controller will generate interrupt when it detects missing ACK during transfer. This bit is disabled when I2C controller is in Byte mode.
0		I2C_DEN	I2C control done interrupt enable 0: Disabled 1: Enabled When this bit set, controller will generate interrupt when it is done with the transfer.

22.2.2. I2C Transmit/Receive Register (I2C_TXRR) 0xF904

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		Field Name	Description
31:8		---	Reserved.
7:1		TXRR[7:1]	Write: Next bytes to transmit via I2C Read: Read back the receive data (bit [7:1])
0		TXRR[0]	Write: In case of a data transfer this bit represent the data's LSB. Incase of a slave address transfer this bit represents the RW bit "1" = Reading from slave "0" = Write to slave Read: Read back the receive data (bit[0]).

22.2.3. I2C Command Register (I2C_CR) 0xF908

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset Value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Note: X means don't care

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset Value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Note: X means don't care

Bit		Field Name	Description
31:8		Reserved	
7		STA	STA, generate (repeated) start condition
6		STO	STO, generate stop condition
5		RD	RD, read from slave
4		WR	WR, write to slave
3		ACK	ACK, when a receiver, send ACK (ACK = "0") or NACK (ACK = "1")
2:0		Reserved	

22.2.4. I2C Prescale Register (I2C_PRE) 0xF90C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit		Field Name	Description
31:16		Reserved	Reserved
15:0		Prescale	Prescale [7:0] = Clock Prescale register lo-byte [15:8] = Clock Prescale register hi-byte prescale = [IMBusCLK / (5 * SCL)] - 1

22.2.4.1. Prescale Register

Register I2C_PRE[15:0] are used to prescale the SCL clock line. Due to the structure of the I^2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when the "I2C mode" is disable.

Example: wb_clk_i = 32MHz, desired SCL = 100KHz

$$prescale = \frac{32\ MHz}{5 * 100\ KHz} - 1 = 63\ (dec) = 3F\ (hex)$$

Reset value: 0xFFFF

22.2.5. I2C Control Register (I2C_CTR) 0xF920

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset Value	0	0	0	0	?	?	1	1	0	0	0	0	0	0	0	0

Bit		Field Name	Description
31:17		Reserved	
16		I2Creset	I2C Reset. 0: normal (default) 1: Reset all pointers and all internal state machine.
15:12		Reserved	
11		GIOSDAi	SDA signal status This bit shows the status of SDA signal (H or L)
10		GIOSCLi	SCL signal status This bit shows the status of SCL signal (H or L)
9		GIOSDAC	SDA control 0: drive SDA = “0” 1: release SDA Only valid when GPIO enable bit is set
8		GIOSCLC	SCL control 0: drive SCL = “0” 1: release SCL Only valid when GPIO enable bit is set
7:3		Reserved	
2		GPIO enable	GPIO Mode Enable 0: Disabled (default) 1: Enabled GPIO Mode If this bit is set, I2C controller will be disable.
1		I2C Mode Select	I2C Mode Select 0: Byte Mode (defaut) 1: Reserved Firmware must not write “1” to this bit.
0		I2C enable	I2C Mode Enable 0: Disable I2C logic 1: Enabled I2C Mode If this bit is set, I2C controller will be enable.

22.3. System Configuration

The I2C system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers. All devices connected to these two signals must have open drain or open collector outputs. The logic AND function is exercised on both lines with external pull-up resistors.

The I2C core is a single master device; therefore, it starts generating a clock as soon as it is released from reset, this being the slowest clock possible (Clock Prescale = 0xFFFF). The user should program this register to the desired value before starting any transfers.

Data is transmitted synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (see START and STOP signals).

22.3.1. I2C Protocol

Normally, a standard communication consists of four parts:

(1) START signal generation
(2) Slave address transfer
(3) Data transfer
(4) STOP signal generation

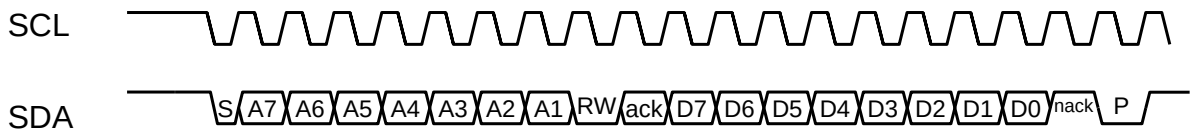


Figure 22-2 I2C Protocol

22.3.2. START signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a high-to-low transition of SDA while SCL is high. The START signal denotes the beginning of a new data transfer.
A Repeated START is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The core generates a START signal when the STA-bit in the Command Register is set and the RD or WR bits are set. Depending on the current status of the SCL line, a START or Repeated START is generated.

22.3.3. Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a seven-bits calling address followed by a RW bit. The RW bit signals the slave data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

Note: The core supports 10bit slave addresses by generating two address transfers. See the Philips I2C specifications for more details.

The core treats a Slave Address Transfer as any other write action. Store the slave device's address in the Transmit Register and set the WR bit. The core will then transfer the slave address on the bus.

22.3.4. Data Transfer

22.3.4.1. Byte Mode:

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a No Acknowledge, the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does not acknowledge the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

To write data to a slave, store the data to be transmitted in the Transmit Register and set the WR bit. To read data from a slave, set the RD bit. During a transfer the core set the TIP flag, indicating that a Transfer is In Progress. When the transfer is done the I2C_DSTS (done) flag is reset, the I2C_DSTS done flag set and, when interrupt enabled, an interrupt generated. The Receive Register contains valid data after the I2C_DSTS flag has been set. The user may issue a new write or read command when the I2C_DSTS flag is reset.

22.3.5. STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a low-to-high transition of SDA while SCL is at logical ‘1’.

22.3.6. Arbitration Procedure

Since the I2C core supports single master configurations only, no Arbitration logic is added to the core. Only clock synchronization is supported since slave devices can use this protocol for clock stretching.

22.3.7. Clock Synchronization

Since the logical AND function is performed on the I2C signals, a high to low transition on SCL or SDA affects all devices connected to the bus. The SCL clock signal can be synchronized between multiple masters using this feature. Each device starts counting its SCL low period when the current master drives SCL low. Once a device’s clock has gone low, it holds the SCL line low until the clock-high-state is reached.

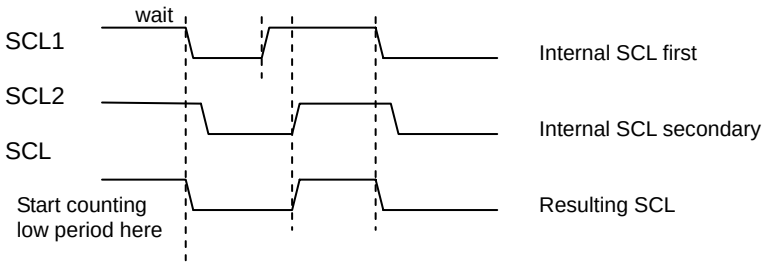


Figure 22-3 Clock Synchronization

22.3.8. Clock Stretching

Slave devices can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal low period is stretched, thus inserting wait-states.

22.4. Architecture

The I2C core is built around four primary blocks; the Clock Generator, the Byte Command Controller, the Bit Command Controller and the DataIO Shift Register.
All other blocks are used for interfacing or for storing temporary values.

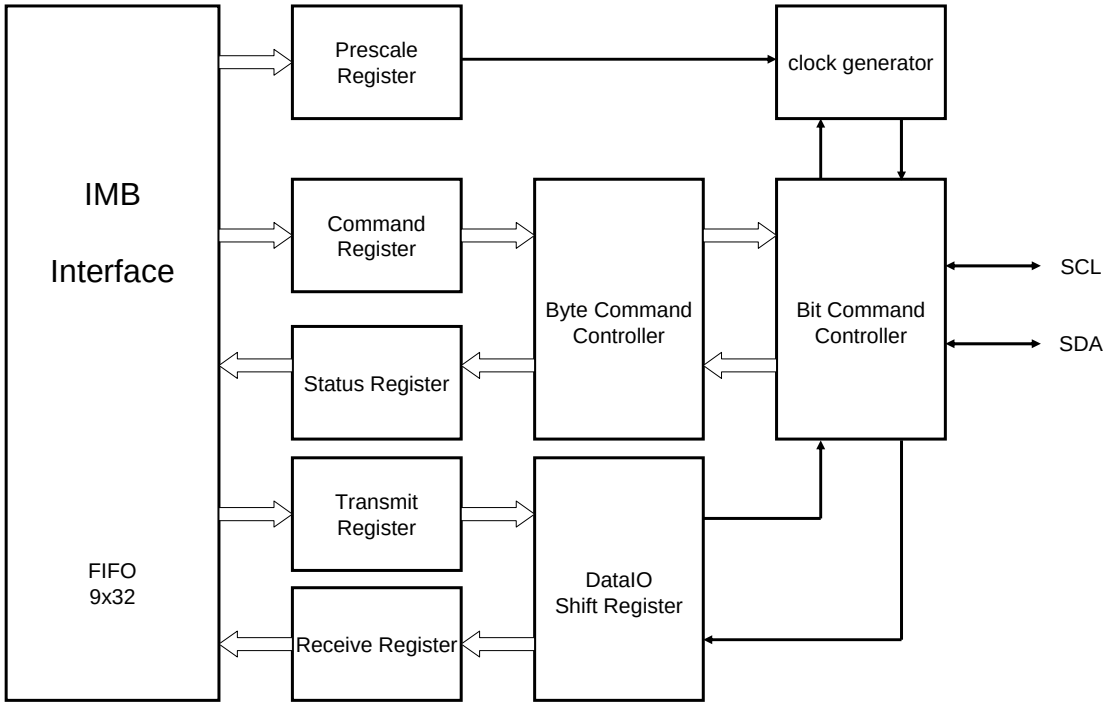


Figure 22-4 Internal structure I2C

22.5. Clock Generator

The Clock Generator generates an internal 4*Fscl clock enable signal that triggers all synchronous elements in the Bit Command Controller. It also handles clock stretching needed by some slaves.

22.6. Byte Command Controller

22.6.1. Byte Mode:

The Byte Command Controller handles I2C traffic at the byte level. It takes data from the Command Register and translates it into sequences based on the transmission of a single byte. By setting the START, STOP, and READ bit in the Command Register, for example, the Byte Command Controller generates a sequence that results in the generation of a START signal, the reading of a byte from the slave device, and the generation of a STOP signal. It does this by dividing each byte operation into separate bit-operations, which are then sent to the Bit Command Controller.

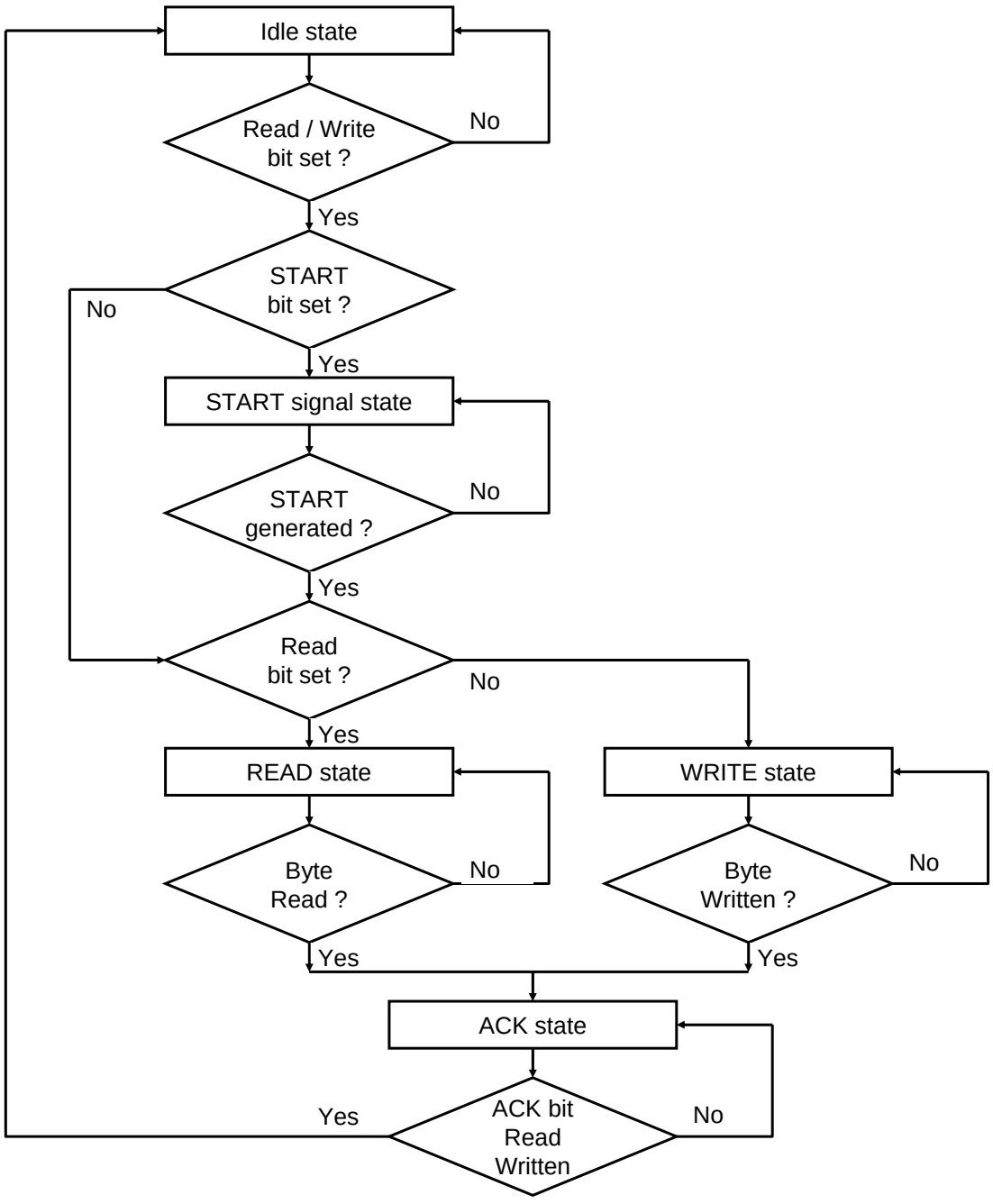


Figure 22-5 Control Flow

22.7. Bit Command Controller

The Bit Command Controller handles the actual transmission of data and the generation of the specific levels for START, Repeated START, and STOP signals by controlling the SCL and SDA lines. The Byte Command Controller tells the Bit Command Controller which operation has to be performed. For a single byte read, the Bit Command Controller receives 8 separate read commands. Each bit-operation is divided into 4 pieces, except for a STOP operation which is divided into 3 pieces.

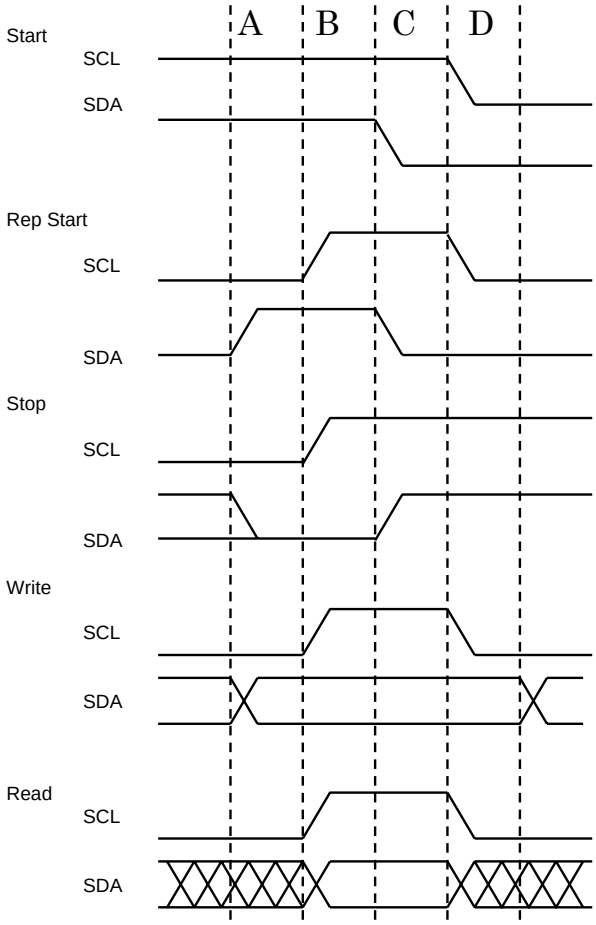


Figure 22-6 Bit Command Control

22.8. Data IO Shift Register

The DataIO Shift Register contains the data associated with the current transfer. During a read action, data is shifted in from the SDA line. After a byte has been read the contents are copied into the Receive Register. During a write action, the Transmit Register's contents are copied into the DataIO Shift Register and are then transmitted onto the SDA line.

22.9. Programming Examples

22.9.1. Example 1 (Byte Mode)

Write 1 byte of data to a slave.

Slave address = 0x51 (b"1010001")
Data to write = 0xAC

I2C Sequence:

(1) generate start command
(2) write slave address + write bit
(3) receive acknowledge from slave
(4) write data
(5) receive acknowledge from slave
(6) generate stop command

Commands:

(1) write 0xA2 (address + write bit) to Transmit Register, set STA bit, set WR bit.

-- wait for interrupt or I2C_DSTS flag to negate –

(2) read I2C_ASTS (ACK) bit from Control/Status Register, should be '0'.
write 0xAC to Transmit register, set STO bit, set WR bit.

-- wait for interrupt or I2C_DSTS flag to negate –

(3) read I2C_DSTS (ACK) bit from Status Register, should be '0'.

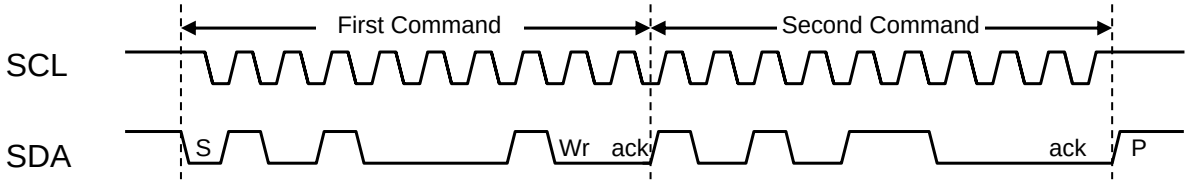


Figure 22-7 Byte Mode

22.9.2. Example 2 (Byte Mode)

Read a byte of data from an I2C memory device.

Slave address = 0x4E
Memory location to read from = 0x20

I2C sequence:

(1) generate start signal
(2) write slave address + write bit
(3) receive acknowledge from slave
(4) write memory location
(5) receive acknowledge from slave
(6) generate repeated start signal
(7) write slave address + read bit
(8) receive acknowledge from slave
(9) read byte from slave
(10) write no acknowledge (NACK) to slave, indicating end of transfer
(11) generate stop signal

Commands:

(1) write 0x9C (address + write bit) to Transmit Register, set STA bit, set WR bit.

-- wait for interrupt or I2C_DSTS (done) flag to negate --

(2) read I2C_ASTS (ACK) bit from Control/Status Register, should be '0'.
write 0x20 to Transmit register, set WR bit.

-- wait for interrupt or I2C_DSTS (done) flag to negate --

(3) read I2C_ASTS (ACK) bit from Control/Status Register, should be '0'.
write 0x9D (address + read bit) to Transmit Register, set STA bit, set WR bit.

-- wait for interrupt or I2C_DSTS (done) flag to negate --

(4) set RD bit, set ACK to '1' (NACK), set STO bit

-- wait for interrupt or I2C_DSTS (done) flag to negate --

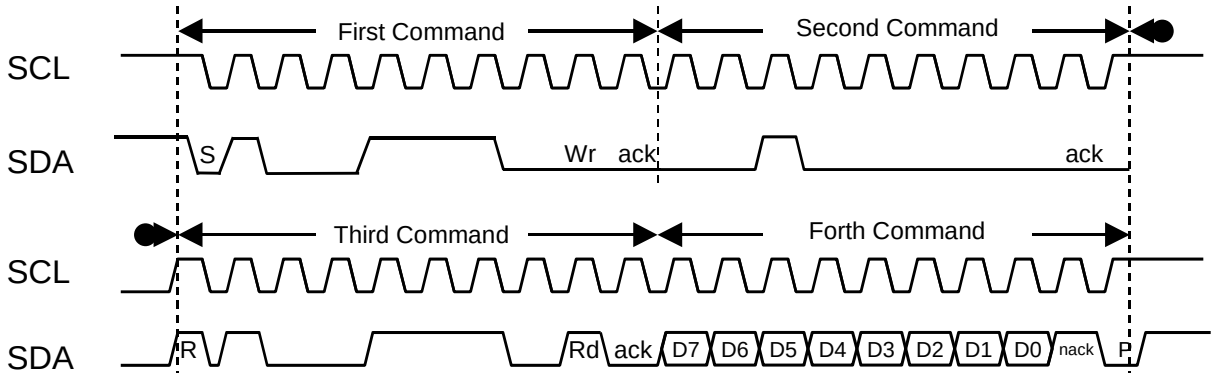


Figure 22-8 Byte Mode

Chapter 23. I2S Controller

23.1. Overview

I2SC is an interface between G-bus and I^2S protocol. This controller supports two programmable I^2S output/input, and one I^2S output.

The basic features of the I2SC are outlined below:

- *Interface to 32-b IMB bus.*
- *Two modes of operation (2 channels i/o mode, 3 Channel output mode)*
- *Support 16, 18, 20 or 24 bits data.*
- *Support Left or Right-justify with MSB first or LSB first.*
- *Support 32, 48 or 64 bit-wide time-slot.*

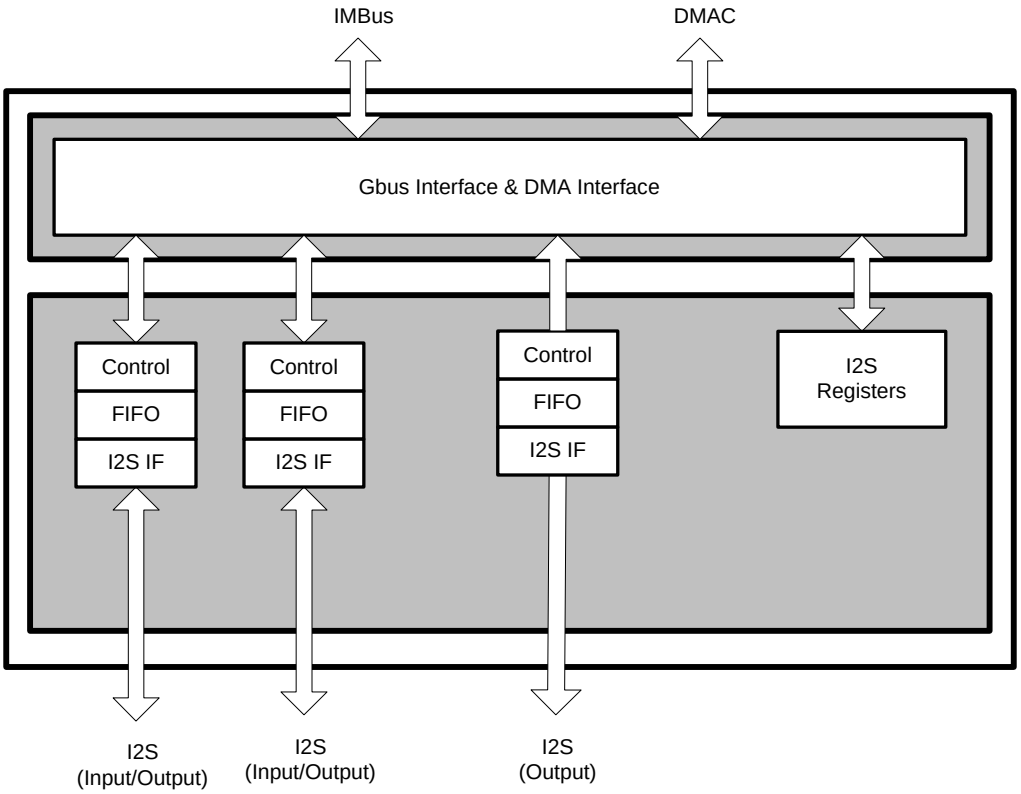


Figure 23-1 the block diagram of the I2SC Interface module.

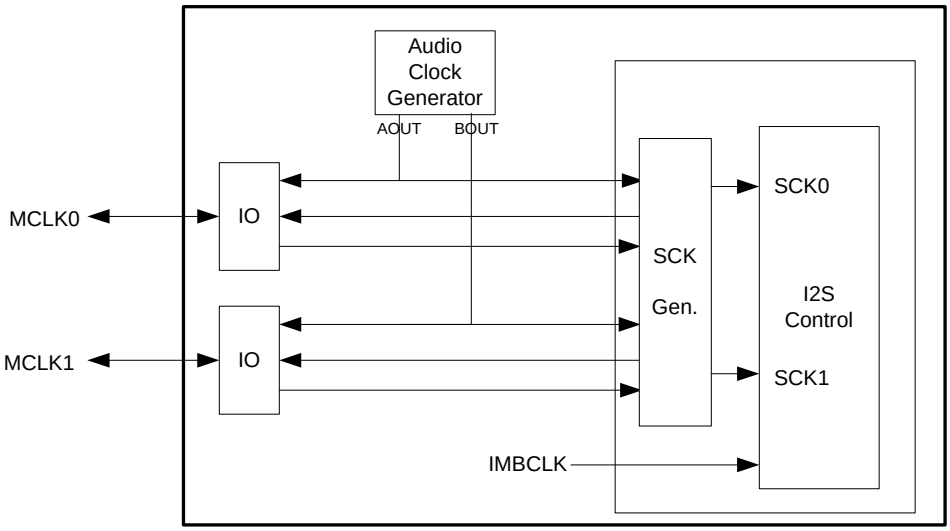


Figure 23-2 the block diagram for I2S clock diagram.

23.2. I2S Function

23.2.1. I²S Interface

I²S bus is a 3-pin serial link consisting of a line for two time multiplexed data channels (left and right), a word select line and a clock line. Since the transmitter and receiver have the same clock signal for data transmission, the transmitter as the master, has to generate the bit clock (SCK), and word-select (WS). Figure 23-3 illustrates simple system configurations and the basic interface timing.

Note that the time-slot (1/fs) could be 64, 48, or 32 bit-wide. Input data could be 16, 18, 20 or 24 bits and could be left- or right-justify with MSB first or LSB first. There are three formats shown in Figure 23-3. Figure 1.3(a) illustrates Standard Data Format (Sony Format) with Left-Channel "H" and Right-channel "L", (b) Left Justified Format with Left-channel "L" and Right-channel "H". For I²S data format, there is one clock delay to latch the data bit.

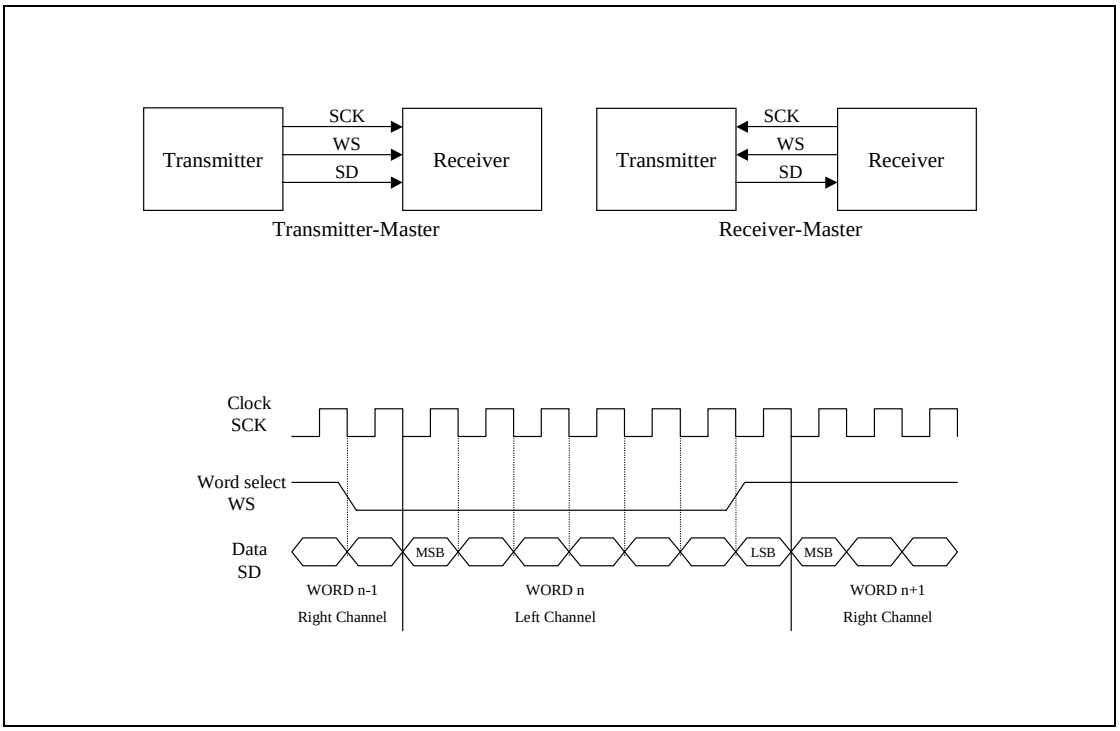


Figure 23-3 I²S Interface

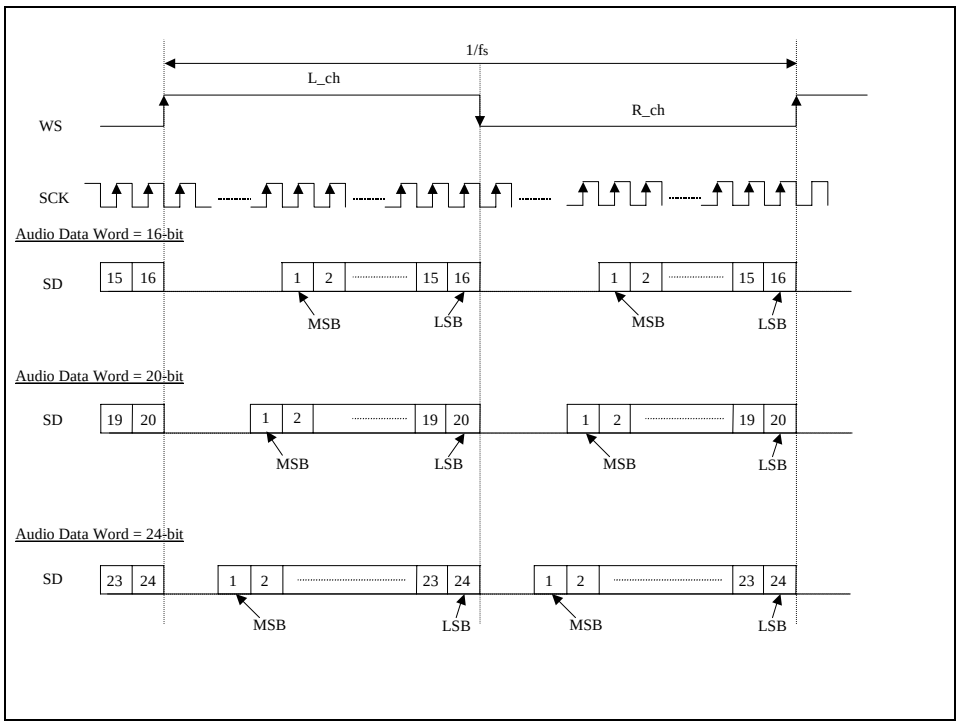


Figure 23-4 Data input timing of standard format (Sony format); L_ch=H, R_CH=L

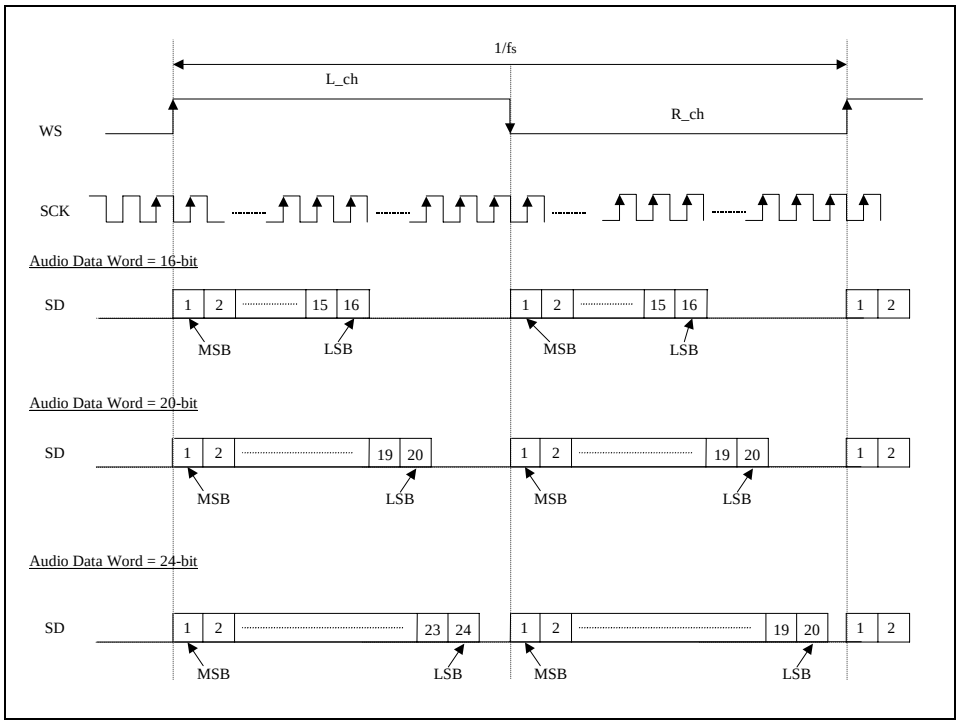


Figure 23-5 Data input timing of Left-Justified format; L_ch=H, R_ch=L

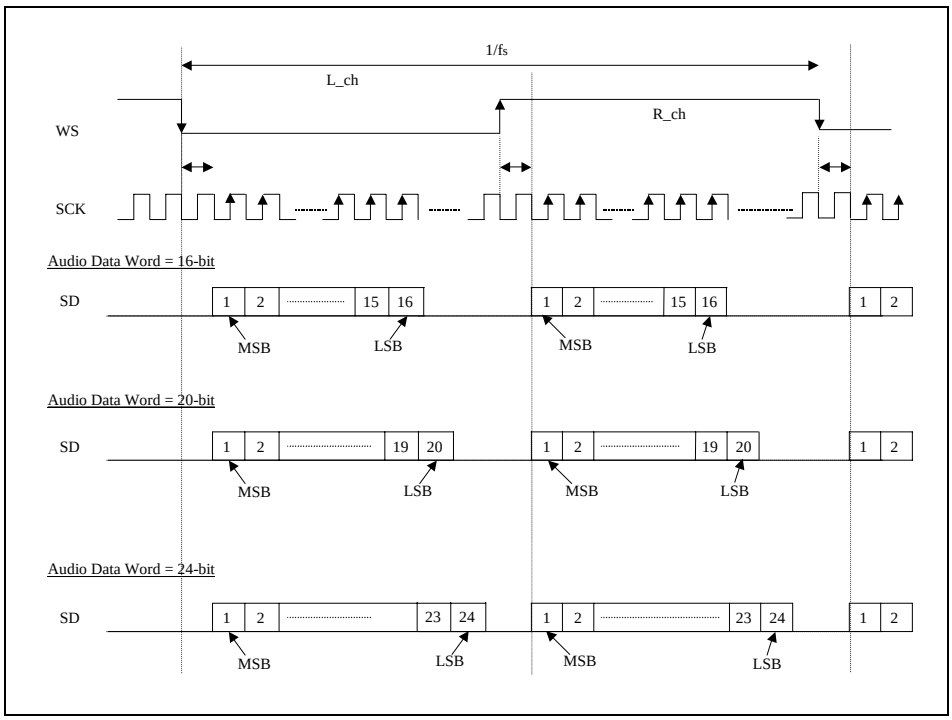


Figure 23-6 Data input timing of I2S data format (Phillips format); L_ch=L, R_ch=H

Each channel has its own control registers bits to control select format it interfaces to.

As mention before, I2SC controller support 16, 18, 20 or 24 bits data format. Below illustrates how these data map within 32 bits data.

16 bits format:	First Data =	Bit[15:0] = [MSB:LSB]	
	Second Data =	Bit[31:16] = [MSB:LSB]	
18 bits format:	Data =	Bit[17:0] = [MSB:LSB]	Bit[31:18] are not used.
20 bits format:	Data =	Bit[19:0] = [MSB:LSB]	Bit[31:20] are not used.
24 bits format:	Data =	Bit[23:0] = [MSB:LSB]	Bit[31:24] are not used.

There is a register bit to swap the MSB and LSB order for each channel.
Below illustrates how these data map within 32 bits data with Data Channel Swap Option is enable

16 bits format:	First Data =	Bit[15:0] = [LSB:MSB]	
	Second Data =	Bit[31:16] = [LSB:MSB]	
18 bits format:	Data =	Bit[17:0] = [LSB:MSB]	Bit[31:18] are not used.
20 bits format:	Data =	Bit[19:0] = [LSB:MSB]	Bit[31:20] are not used.
24 bits format:	Data =	Bit[23:0] = [LSB:MSB]	Bit[31:24] are not used.

In I2SC controller the data transfer is in Little Indian. The following illustrates how data and address appears on G bus.

D[31:0]	D[63:32]
A[3:2]= 2'b00	A[3:2]=2'b01

23.2.2. Mode of operation

There are two modes of operation
2 channel i/o mode
3 channel output mode

For 2 channels i/o mode: there are two pairs (SCK, WS, SD) of i2s signal. These two channels can be program to be input or output. At reset, default will be input mode.
For 3 channel output mode: There are common SCK,WS and separate three data line (SD0, SD1, SD2). SCK0 and WS0 are the common SCK and WS. WS1 becomes SD2. And SCK1 is not use. The following is the pin mux table

Signal name/Mode	2 Channel i/o Mode	3 Channel Output Mode
I2S_SCK0	I2S_SCK0	I2S_SCK
I2S_WS0	I2S_WS0	I2S_WS
I2S_SD0	I2S_SD0	I2S_SD0
I2S_SCK1	I2S_SCK1	N/A
I2S_WS1	I2S_WS1	I2S_SD2
I2S_SD1	I2S_SD1	I2S_SD1

23.2.3. MCLK and SCK Clock

SCK derived from Master Clock (MCLK). Internal to TX4939, there are an Audio Clock Generator. This Clock Generator will generate two Master Clock (MCLK0 and MCLK1). MCLK0 uses in 3 channel output modes. Both MCLK0 and MCLK1 use in 2 channels i/o mode.

The source of MCLK0 and MCLK1 can be from internal Audio Clock Generator or external clock source. I2S Channel Control Register has two bits to control these options for MCLK0 and MCLK1. Bit[23] selects clock source for MCLK1, bit[[7] selects clock source for MCLK0.

Bit[23]	MCLK1	MCLK1 IO pad
0	Slave - External clock	Input
1	Master - Internal clock	Output

Bit[7]	MCLK0	MCLK0 IO pad
0	Slave - External clock	Input
1	Master - Internal clock	Output

* Internal Audio Clock Generator generates the following frequency

24.5760Mhz, 22.5792Mhz, 16.3840Mhz, 14.7456Mhz, 12.2880Mhz, 8.1920Mhz, 4.0960Mhz

* There are two options for SCK clock source.

First option is Master mode, SCK is internally derived from the Master Clock (MCLK). The divider is used to generate SCK clock. Divider value is in I2SDIV register.

Second option is Slave mode. SCK is from external source. MCLK is not use in this mode.

* The following shows the divider value for different sample rate and different time slot.

Sample Rate	Time Slot		
	32-bit	48-bit	64-bit
768x	24	N/A	12
512x	N/A	N/A	8
384x	12	8	6
256x	8	N/A	4
192x	6	4	3
128x	4	N/A	2
96x	3	2	N/A
64x	2	N/A	1

Figure 23-7 shows the relation between MCLK and SCK with Divider value = 2,3,4

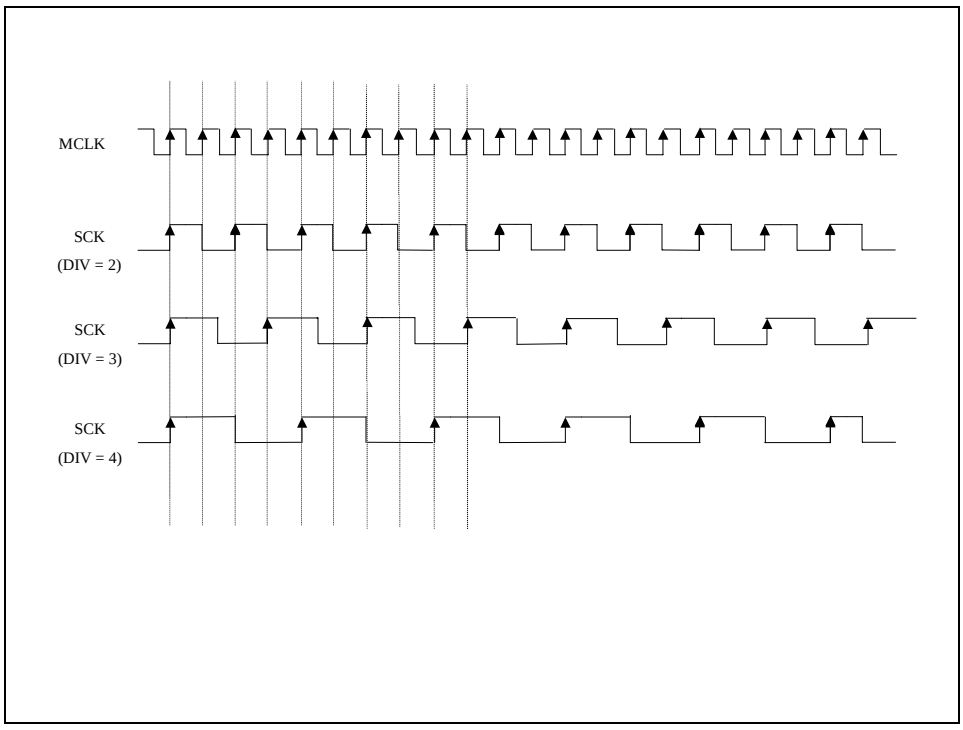


Figure 23-7 MCLK and SCK with Divider Value

23.2.4. MCLK and WS Detection

In I2S controller, there is logic that checks MCLK and WS clock frequency.

For MCLK detection, fast clock (150 MHz) uses to sample the MCLK clock. At every rising edge of MCLK, the counter begins to count (from “0”) base on 150 MHz clock. And also at every rising edge of MCLK, this count value get compare with the MCLK check value (register bits). If the count value is greater then MCLK check value, which means MCLK is out of range, Interrupt will get set. Otherwise it continues sampling MCLK. Once out of range detect, counter logic will stop counting.

For the WS detection, the logic simply samplings the WS with SCK clock. At every rising and falling edge of WS, counter starts counting from “0” base on SCK clock. It also compares the counter value at every rising edge of WS. If counter value match the setting in the control register (time slot – 32, 48, 64) means WS frequency is correct. If counter value does not match, WS interrupt sets to high and counter logic detection stops counting.

23.3. DMA Interface

23.3.1. Overview

This section describes the mechanism for transmission and reception of I2S data. An overview is described first. The DMA (Direct Memory Access) operation, error detection and recovery procedure follow.

Figure 23-8 and Figure 23-9 show conceptual views of the sample-data transmission and reception mechanisms.

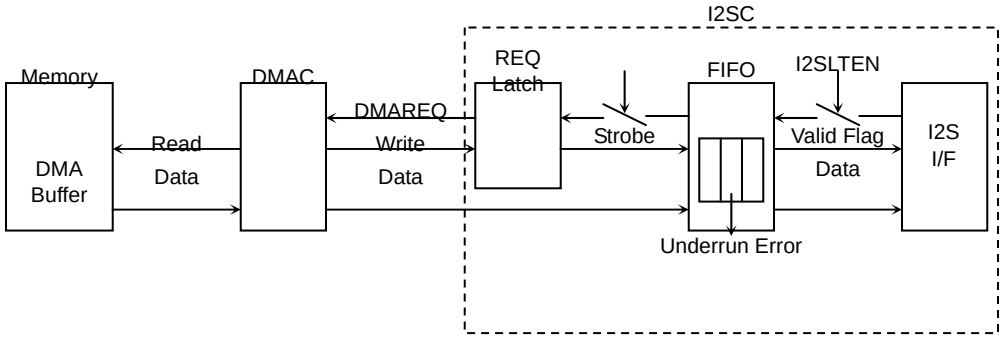


Figure 23-8 Sample-data Transmission Mechanism

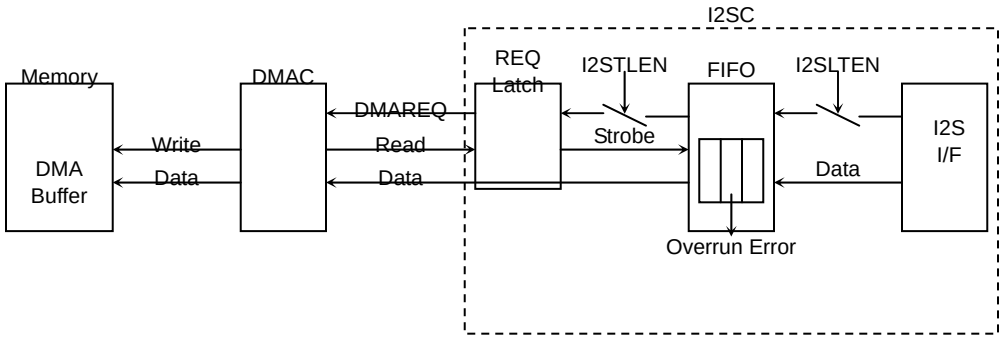


Figure 23-9 Sample-data Reception Mechanism

Transmission or reception through each stream can be independently activated or deactivated under control of I2SC Channel Control Register (start channel bits).

I2S controller is equipped with a separate FIFO for each channel. The data to transmit is prefetched from memory to FIFO through DMA. The received data is buffered on FIFO and then stored to memory through DMA. In this stage, each DMA is independently activated or deactivated under control of I2SC Channel Main Control Register (I2SMCR).

23.3.2. DMA Channel Mapping:

I2SC controller uses three DMA request channels. There is a dedicated DMA channel for each port.

For 2 channel i/o mode

I2S channel #0 uses DMAC1 port #0
I2S channel #1 uses DMAC1 port #1

For 3 Channel Output mode

I2S channel #0 uses DMAC1 port #0
I2S channel #1 uses DMAC1 port #1
I2S channel #2 uses DMAC1 port #2

23.3.3. DMA Operation:

Setup DMA Channel will be the same as AC-link Controller.
For an output channel, assign an i2s channel address to DMAC destination address register (DMDARn).
For an input channel, assign an i2s channel address to DMAC source address register (DMSARn).

The lower 16 bit address for i2s channels are :

Channel 0 : xFA80
Channel 1 : xFA90
Channel 2 : xFAA0

23.4. I²S Register Definition

23.4.1. Register Definition

All registers are 32 bits. Only word access allows. i.e. all instructions are LW or SW instructions. Other type of access may cause unexpected operation.

Offset	Register	Width	R / W	Description
0xFA00	I2SMCR	[31:0]	R/W	I2S Channel Main Control Register
0xFA04	I2SCCR	[31:0]	R/W	I2S Channel Control Register
0xFA08	I2SICTRL	[31:0]	R/W	I2S Interrupt Control Register
0xFA0C	I2SCOR	[31:0]	R/W	I2S Clock Option Register
0xFA80	I2SCH0D	[31:0]	R	Channel 0 Data *
0xFA90	I2SCH1D	[31:0]	R	Channel 1 Data *
0xFAA0	I2SCH2D	[31:0]	R	Channel 2 Data *

Note: * These registers are used by DMA engine to transfer data from/to I2S. Software should not read these.

23.4.2. I2S Channel Main Control Register (I2SMCR) 0xFA00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset Value	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		Field Name	Description
D31:3		---	Reserved
D2		---	Start Channel 2
D1		---	Start Channel 1
D0		---	Start Channel 0 0: Stop (default) 1: Start * When Start enabled. The first transfer/receive data is when WS = L. * When Stop enabled. The I2S controller stops when it finishes with the current FIFO.

23.4.3. I2S Channel Control Register (I2SCCR) 0xFA04

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	R/W	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

Bit		Field Name	Description
D31		---	I2S mode option 0: 2 channel i/o mode (default) 1: 3 channel output mode
D30:28		---	Reserved.
D27		---	WS Check Enable (Channel 1 i/o mode) 0: Disable (default) 1: Enable When 3 Channel Mode is enable, this bit is not used
D26		---	MCLK Check Enable (Channel 1 i/o mode) 0: Disable 1: Enable When 3 Channel Mode is enable, this bit is not used
D25		---	Clock Data Option (Channel 1 i/o mode) 0: Falling edge (default) 1: Rising edge When 3 Channel Mode is enable, this bit is not used Notes: Falling edge selected: WS is toggle at the falling edge of SCK Rising edge selected: WS is toggle at the rising edge of SCK Output mode: Falling edge selected: clock SD at falling edge Rising edge selected: clock SD at rising edge Input mode: Falling edge selected: sample SD at rising edge Rising edge selected: sample SD at falling edge
D24		---	Data Channel Swap Option (Channel 1 i/o mode) 0: Normal (default) 1: Swap (MSB <--> LSB) When 3 Channel Mode is enable, this bit is not used
D23		---	MCLK Mode Option (Channel 1 i/o mode) 0: Slave mode. Receive MCLK1 1: Master mode. Drive MCLK1 (Default) When 3 Channel Mode is enable, this bit is not used
D22		---	Receiver-Master Mode Option (Channel 1 i/o mode) 0: Slave mode. Receive SCK, WS 1: Master mode. Drive SCK, WS (Default) When 3 Channel Mode is enable, this bit is not used
D21		---	Clock Delay Option (Channel 1 i/o mode) 0: Latch data on the first clock (default) 1: Latch data on the second clock For Left-Justify mode only When 3 Channel Mode is enable, this bit is not used
D20		---	Invert WS Option (Channel 1 i/o mode) 0: As normal (default) 1: Invert WS Invert WS right at the IO pad. When 3 Channel Mode is enable, this bit is not used
D19		---	Left/Right Justify Select Option (Channel 1 i/o mode) 0: Left-Justify (default) 1: Right-Justify When 3 Channel Mode is enable, this bit is not used
D18:17		---	Data Select Option (Channel 1 i/o mode) 00: 16 bits data (default) 01: 18 bits data 10: 20 bits data 11: 24 bits data When 3 Channel Mode is enable, these bits are not used

Bit		Field Name	Description
D16		---	Channel 1 i/o Mode Option 0: Output channel mode 1: Input channel mode (default) When 3 Channel Mode is enable, this bit is not used.
D15:12		---	Reserved.
D11		---	WS Check Enable (Channel 0 i/o and 3 Channel Output mode) 0: Disable (default) 1: Enable
D10		---	MCLK Check Enable (Channel 0 i/o and 3 Channel Output mode) 0: Disable (default) 1: Enable
D9		---	Clock Data Option (Channel 0 i/o and 3 Channel Output mode) 0: Falling edge (default) 1: Rising edge Notes: Falling edge selected: WS is toggle at the falling edge of SCK Rising edge selected: WS is toggle at the rising edge of SCK Output mode: Falling edge selected: clock SD at falling edge Rising edge selected: clock SD at rising edge Input mode: Falling edge selected: sample SD at rising edge Rising edge selected: sample SD at falling edge
D8		---	Data Swap Option (Channel 0 i/o and 3 Channel Output mode) 0: Normal (default) 1: Swap (MSB <--> LSB)
D7		---	MCLK Mode Option (Channel 0 i/o and 3 Channel Output mode) 0: Slave mode. Receive MCLK0 1: Master mode. Drive MCLK0 (Default)
D6		---	Receiver-Master Mode Option (Channel 0 i/o and 3 Channel Output mode) 0: Slave mode. Receive SCK, WS 1: Master mode. Drive SCK, WS (Default)
D5		---	Clock Delay Option (Channel 0 i/o and 3 Channel Output mode) 0: Latch data on the first clock (default) 1: Latch data on the second clock For Left-Justify mode only
D4		---	Invert WS Option (Channel 0 i/o and 3 Channel Output mode) 0: As normal (default) 1: Invert WS
D3		---	Left/Right Justify Select Option (Channel 0 i/o and 3 Channel Output mode) 0: Left-Justify (default) 1: Right-Justify
D2:1		---	Data Select Option (Channel 0 i/o and 3 Channel Output mode) 00: 16 bits data (default) 01: 18 bits data 10: 20 bits data 11: 24 bits data
D0		---	Channel 0 i/o Mode Option 0: Output channel Mode 1: Input channel Mode(default) When 3 Channel Output Mode is enable, this bit is not used.

23.4.4. I2S Interrupt Control Register (I2SICTRL) 0xFA08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		Field Name	Description
D31:22		---	Reserved
D21		---	Underrun/Overrun Interrupt Mask for Channel 1 i/o mode 0: Normal 1: Mask (default) When 3 Channel Mode is enable, this bit must set to “1”.
D20		---	Underrun/Overrun Interrupt Mask for Channel 0 i/o mode and 3 channel output mode
D19		---	WS1 Error Detect Interrupt Mask 0: Normal 1: Mask (default) When 3 Channel Mode is enable, this bit must set to “1”.
D18		---	WS0 Error Detect Interrupt Mask
D17		---	MCLK1 Error Detect Interrupt Mask 0: Normal 1: Mask (default) When 3 Channel Mode is enable, this bit must set to “1”.
D16		---	MCLK0 Error Detect Interrupt Mask 0: Normal 1: Mask (default)
D15:6		---	Reserved
D5		---	Underrun/Overrun Interrupt Status for Channel 1 i/o mode 0: Normal 1: Interrupt Write “1” to clear the interrupt
D4		---	Underrun/Overrun Interrupt Status for Channel 0 i/o and 3 channel output mode 0: Normal 1: Interrupt Write “1” to clear the interrupt
D3		---	WS1 Error Detect Interrupt Status 0: Normal 1: Interrupt Write “1” to clear the interrupt
D2		---	WS0 Error Detect Interrupt Status 0: Normal 1: Interrupt Write “1” to clear the interrupt
D1		---	MCLK1 Error Detect Interrupt Status 0: Normal 1: Interrupt Write “1” to clear the interrupt
D0		---	MCLK0 Error Detect Interrupt Status 0: Normal 1: Interrupt Write “1” to clear the interrupt

23.4.5. I2S Clock Option Register (I2SCOR) 0xFA0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W/RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W/RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit		Field Name	Description
D31:29		---	Reserved
D28:24		---	MCLK1 check value. Used for Channel 1 i/o mode only. 5'b00000 check for 1 5'b00001 check for 2 …… …….. 5'b11111 check for 32 (default) These five bits value will get compare with the MCLK0 counter. If the counter value is greater than the check value then the MCLK0 Interrupt will get set.
D23:22		---	Time-Slot Select Option for WS1. Used for Channel 1 i/o mode only. 00: 32 bit-wide (default) 01: 48 bit-wide 10: 64 bit-wide 11: reserved.
D21:20		---	Reserved
D19:16		---	MCLK1 divider factor. Used for Channel 1 I/O mode only. 4'b0000 SCK_OUT = MCLK1 / 1 (default) 4'b0001 SCK_OUT = MCLK1 / 1 4'b0010 SCK_OUT = MCLK1 / 2 4'b0011 SCK_OUT = MCLK1 / 3 4'b0100 SCK_OUT = MCLK1 / 4 4'b0101 SCK_OUT = MCLK1 / 6 4'b0110 SCK_OUT = MCLK1 / 8 4'b0111 SCK_OUT = MCLK1 / 9 4'b1000 SCK_OUT = MCLK1 / 12 4'b1001 SCK_OUT = MCLK1 / 24 4'b1010 SCK_OUT = MCLK1/1 ---------- ---------- 4'b1111 SCK_OUT = MCLK1/1 A MCLK1 divider factor. The SCK1_OUT for channel 1 is derived from MCLK1 clock. This register contains the divisor factor.
D15:13		---	Reserved
D12:8		---	MCLK0 check value. Used for Channel 0 i/o mode and 3 channels output mode. 5'b00000 check for 1 5'b00001 check for 2 …… …….. 5'b11111 check for 32 (default) These five bits value will get compare with the MCLK0 counter. If the counter value is greater than the check value then the MCLK0 Interrupt will get set.
D7:6		---	Time-Slot Select Option for WS0. Used for Channel 0 i/o mode and 3 channels output mode. 00: 32 bit-wide (default) 01: 48 bit-wide 10: 64 bit-wide 11: reserved.
D5:4		---	Reserved

Bit		Field Name	Description
D3:0		---	MCLK0 divider factor. Used for Channel 0 I/O mode and 3 channels output mode. 4'b0000 SCK_OUT = MCLK0 / 1 (default) 4'b0001 SCK_OUT = MCLK0 / 1 4'b0010 SCK_OUT = MCLK0 / 2 4'b0011 SCK_OUT = MCLK0 / 3 4'b0100 SCK_OUT = MCLK0 / 4 4'b0101 SCK_OUT = MCLK0 / 6 4'b0110 SCK_OUT = MCLK0 / 8 4'b0111 SCK_OUT = MCLK0 / 9 4'b1000 SCK_OUT = MCLK0 / 12 4'b1001 SCK_OUT = MCLK0 / 24 4'b1010 SCK_OUT = MCLK0/1 ---------- ---------- 4'b1111 SCK_OUT = MCLK0/1 A MCLK0 divider factor. The SCK0_OUT for channel 0 is derived from MCLK0 clock. This register contains the divisor factor.

23.5. Interface Signals

Signal Name	I/O	Signal Description
IMB-Bus Interface Signals		
xx	I	xx
I2S Interface Signals		
I2S_MCLK1	I/O	Master Clock 1. Used for channel i/o mode only.
I2S_MCLK0	I/O	Master Clock 0. Used for channel 0 i/o mode and 3 channel output mode.
I2S_SCK0	I/O	Channel 0 Clock Line
I2S_WS0	I/O	Channel 0 Word Select
I2S_SD0	I/O	Channel 0 Data Line
I2S_SCK1	I/O	Channel 1 Clock Line When select 3 channel output mode, this signal is not use
I2S_WS1/SD2	I/O	Channel 1 Word Select When select 3 channel ouput mode, this signal become Channel 2 Data Line.
I2S_SD1	I/O	Channel 1 Data Line
Interrupt		
I2S_INT	O	I2S Interrupt signal.

Chapter 24. ACLINK Controller

24.1. Features

ACLC, AC-link controller module can be connected to audio and/or modem CODECs described in the “Audio CODEC ’97 Revision 2.1” (AC’97) defined by Intel and can operate them. Refer to the following Web site for more information regarding the AC’97 specification.

http://developer.intel.com/ial/scalableplatforms/audio/

Its features are summarized as follows.

- *Up to two CODECs are supported.*
- *AC’97 compliant CODEC register access protocol is supported.*
- *CODEC register access completion is recognized by polling or interrupt.*
- *Recording and playback of 16-bit PCM Left&Right channels are supported.*
- *Recording can be selected from PCM L&R or Mic.*
- *Playback of 16-bit Surround, Center, and LFE channels is supported.*
- *Variable Rate Audio recording is supported.*
- *Variable Rate Audio playback is supported.*
- *Line 1 and GPIO slots for Modem CODEC are supported.*
- *AC-link low-power mode, wake-up, and warm-reset are supported*
- *Sample-data I/O via DMA transfer is supported.*

24.2. Configuration

Figure 24-1 illustrates the ACLC configuration.

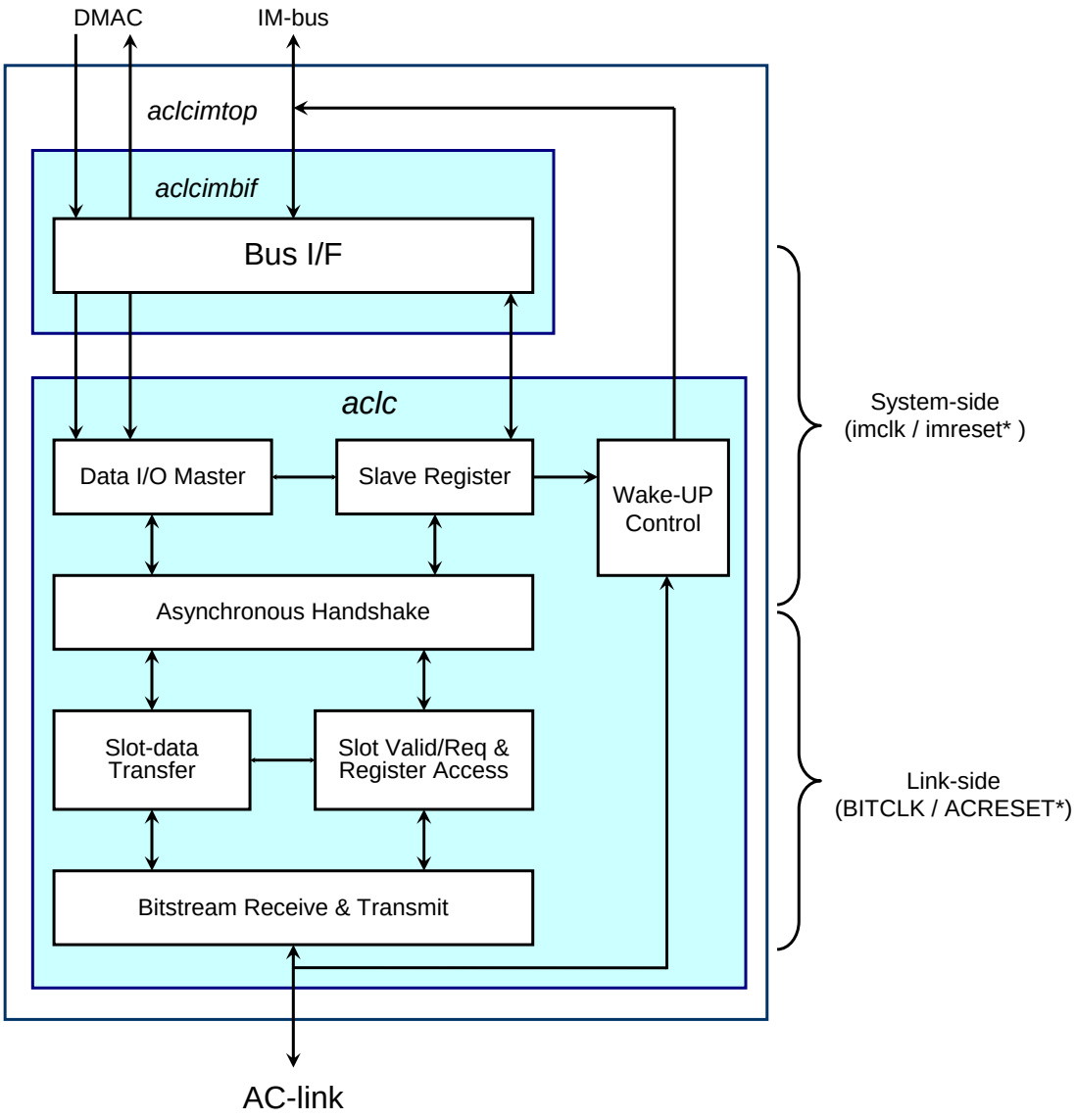


Figure 24-1 ACLC Module Configuration

24.3. Functional Description

ACLC provides four mechanisms to operate AC'97-compliant CODEC(s):

- *AC-link status control (start-up and low-power mode)*
- *CODEC register access*
- *Sample-data transmission and reception*
- *GPIO operation*

This section first describes the CODEC connection, chip configuration, and overall usage-flow. Then AC-link start-up sequence and the other mechanisms will be described. Using low-power mode comes last.

24.3.1. CODEC Connection

The ACLC module has two SDIN (named as SDATA_IN in the AC'97 specification) signals and supports up to two CODECs to be connected. This section shows some system configuration diagrams for typical usages. Note that the diagrams shown here is intended to provide conceptual understanding and some components may be necessary on the actual circuit board to ensure proper electrical signals. The diagrams assume CODECs compliant with the CODEC ID strapping recommendation described in the section D.5.2 of the AC'97 revision 2.1 specification.

24.3.1.1. Stereo Audio and Optional Modem Connection

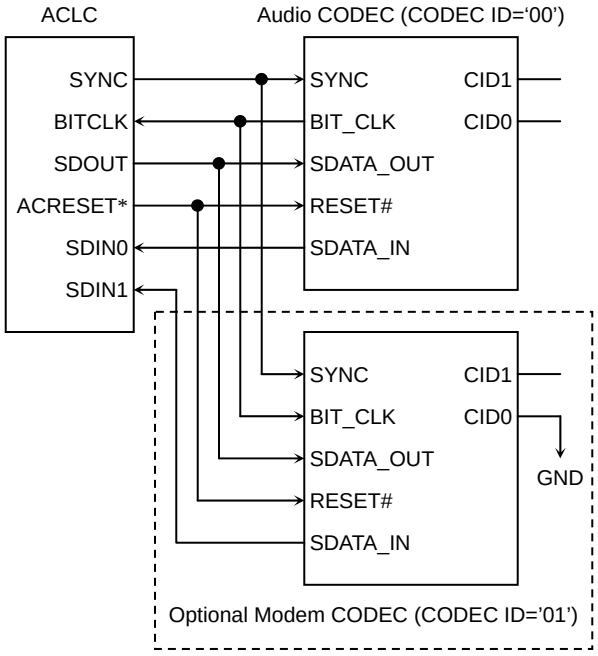


Figure 24-2 Stereo Audio and Optional Modem Connection Diagram

24.3.1.2. 5.1 Channel Audio Connection

This sample assumes one CODEC with four DACs mapped to stereo front (3&4) and stereo rear (7&8) slots, and another CODEC with two DACs mapped to center (6) and LFE (9) slots.

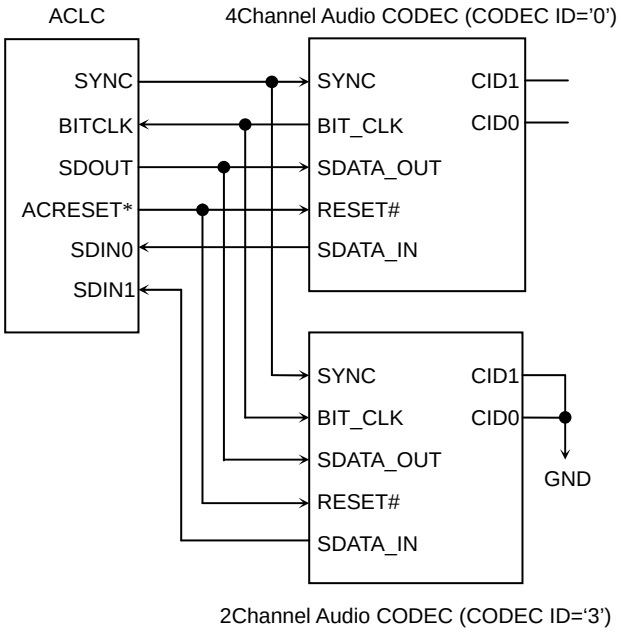
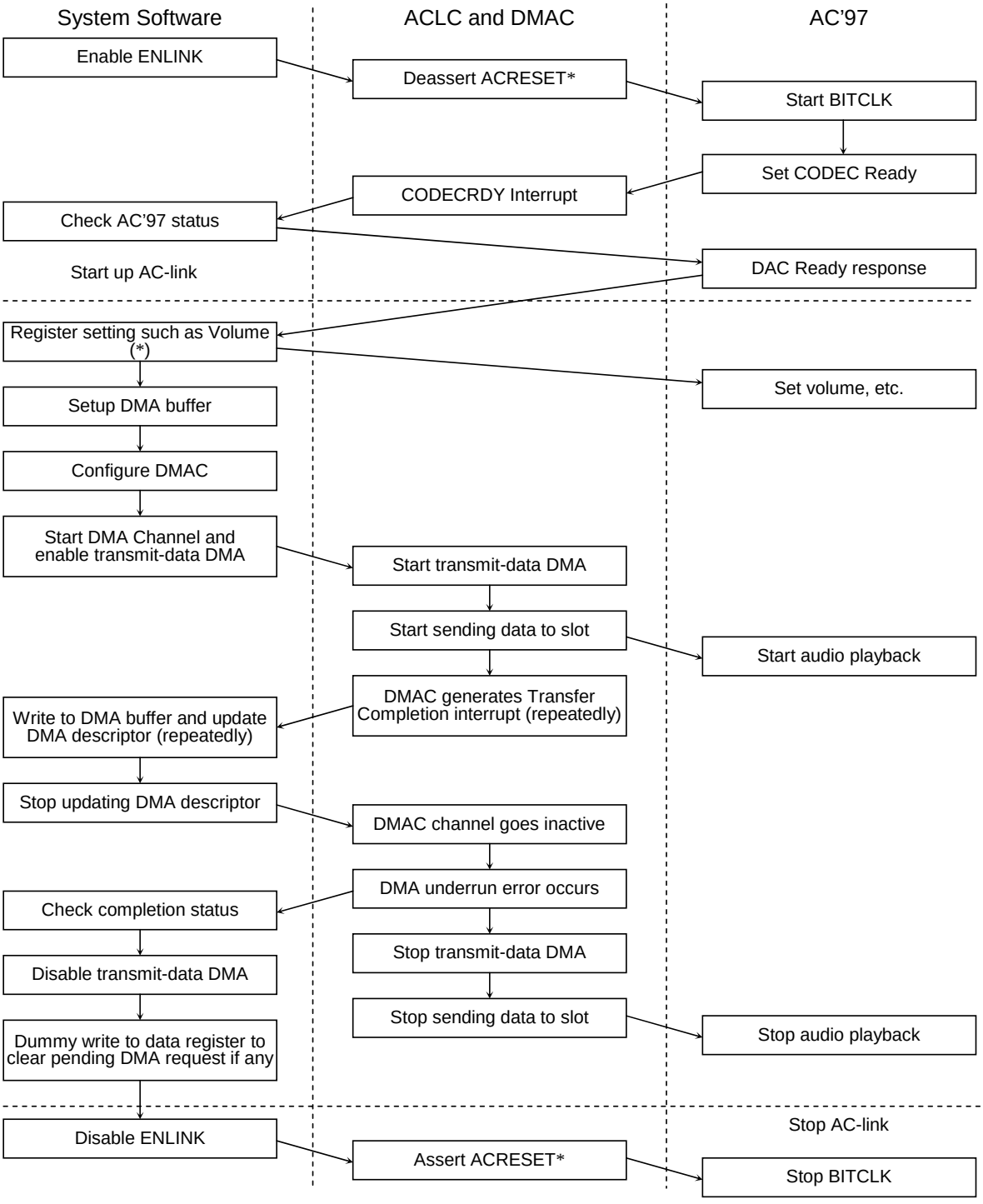


Figure 24-3 5.1 Channel Audio Connection Diagram

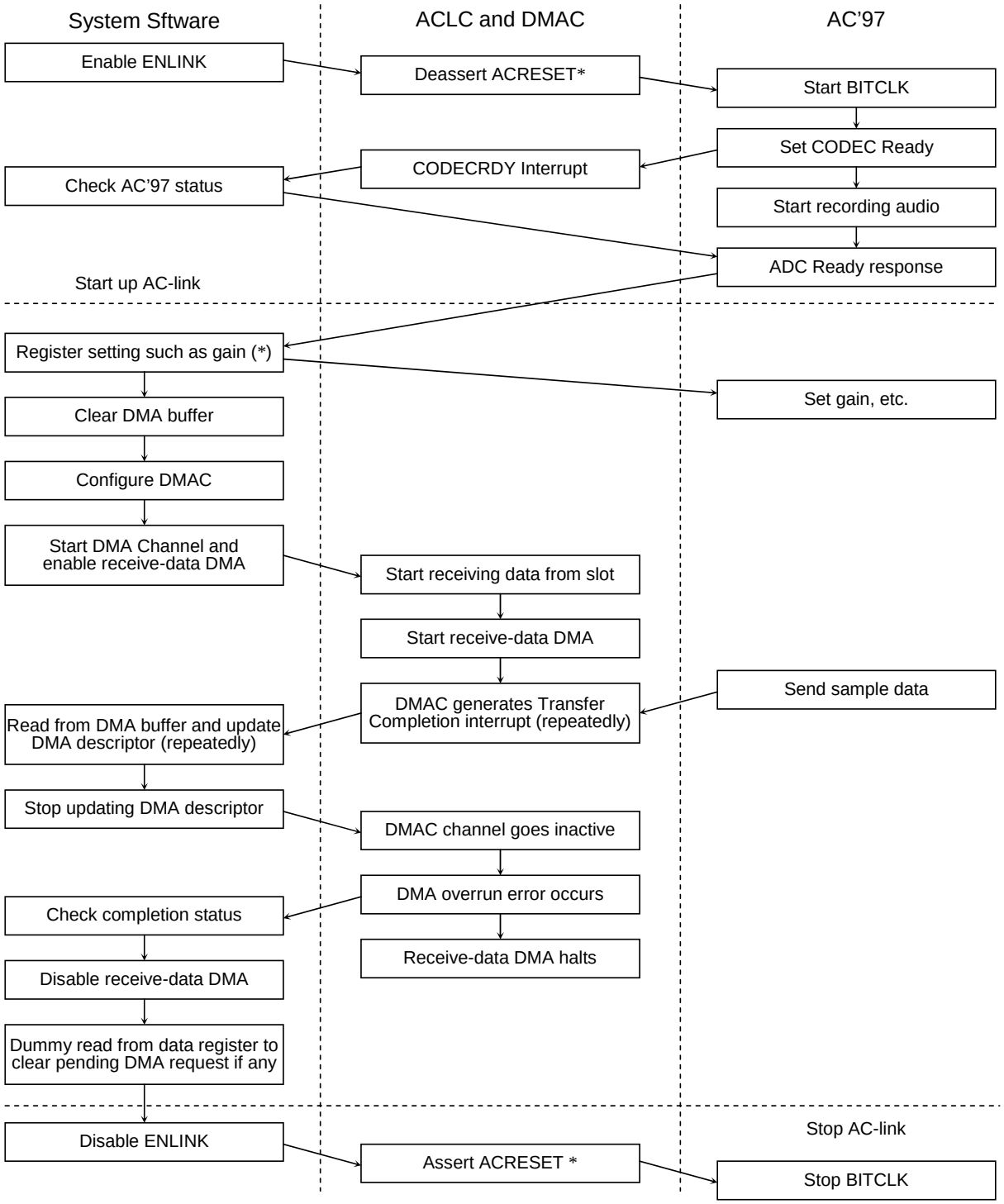
24.3.2. Usage Flow

This section outlines a process flow when using the AC'97 connected to ACLC. Refer to the subsequent sections for the details of each operation performed in this process flow. The diagrams below describe the audio playback and recording processes. The modem transmission and reception can be done in a similar way.



(*) Register settings such as volume can be made during data playback.

Figure 24-4 Audio Playback Process Flow



(*) Register settings such as volume can be made during data playback.

Figure 24-5 Audio Recording Process Flow

24.3.3. AC-link Start Up

Figure 24-6 shows the conceptual sequence of AC-link start-up.

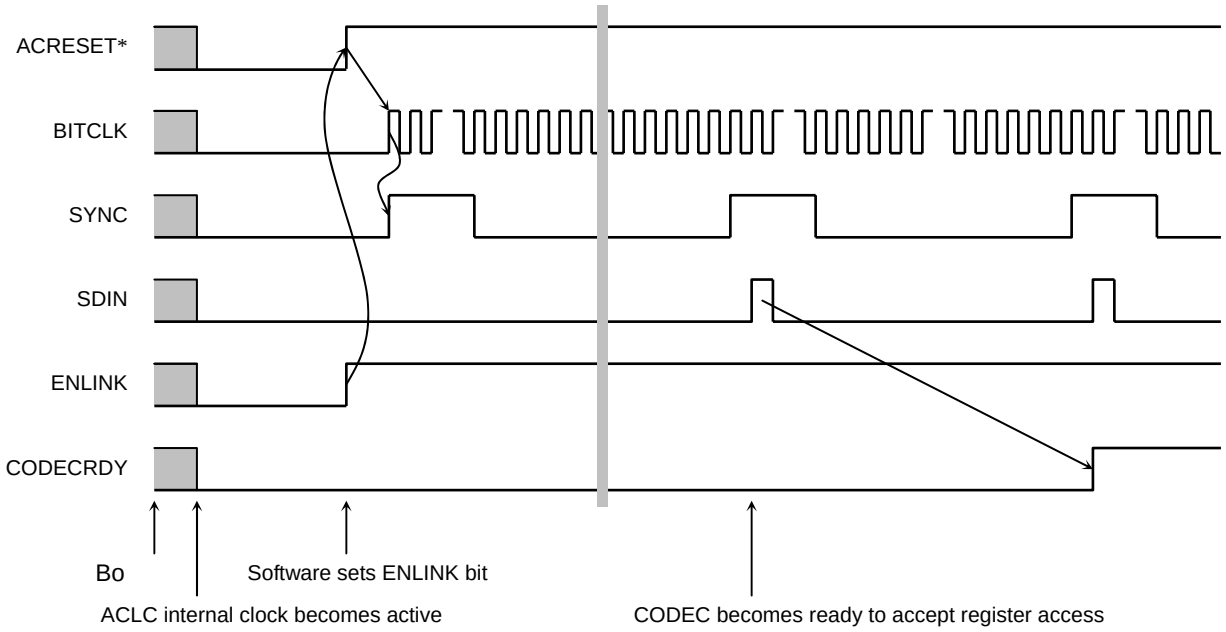
The ACLC Control Enable Register's Enable AC-link bit is used to deassert/assert the ACRESET* signal to the link side (including AC-link). This bit defaults to '0', so the CPU asserts the ACRESET* signal when it boots up.

The AC'97 specification requires that the reset assertion period is 1μs or longer. The software is responsible for controlling the length of this period.
The AC'97 specification also requires that the primary CODEC stops the AC-link clock (BITCLK) signal during the period from ACRESET* signal assertion to 162.8ns after ACRESET* signal deassertion. ACLC assumes the primary CODEC meet this requirement.

Deasserting the link-side reset makes the primary CODEC start driving the BITCLK signal. When the BITCLK signal is provided, ACLC starts the SYNC signal output, which indicates the start of the AC-link frame, and starts the frame-length counting.

When a CODEC becomes ready to receive access to its own register, the CODEC sets the "CODEC Ready" bit of the Tag slot. When ACLC detects that this bit has been set, the ACLC Interrupt Status Register (ACINTSTS)'s CODEC[1:0] Ready (CODEC[1:0]RDY) bit is set. The system software is able to recognize the readiness of the CODEC(s) by detecting this event by way of either polling or interrupt.

In case of 5.1 channel audio connection example (Figure 24-3), because the secondary CODEC is connected to the SDIN1 signal of ACLC, the software must watch ACINTSTS.CODEC1RDY bit to determine the CODEC's readiness for the register access.



Note: The number of BITCLK cycles relative to other signals is not to scale.

Figure 24-6 Cold Reset and CODEC Ready Recognition

24.3.4. CODEC Register Access

By accessing registers in the CODEC, the system software is able to detect or control the CODEC state. This section describes how to read and write CODEC registers via ACLC. For details about AC'97 register set and proper sequence to operate CODEC, refer to the AC'97 specification and target CODEC datasheet.

It takes several frame periods for a read or write access to complete. Taking this into account, ACLC is equipped with a function for reporting CODEC register access completion as status-change or interrupt.

In order to read an AC'97 register, write the access destination CODEC ID and register address in ACLC CODEC Register Access Register (ACREGACC) with its CODECRD bit set to "1". After the ACLC Interrupt Status Register (ACINTSTS)'s REGACC Ready (REGACCRDY) bit is set, the software is able to get the data returned from the AC'97 by reading the ACREGACC register and issue another access.

In order to write to an AC'97 register, write the access destination CODEC ID, register address, and the data in ACLC's ACREGACC register with ACREGACC.CODECRD bit set to "0". After the ACINTSTS.REGACCRDY bit has been set, the software is able to issue another access.

In case of 5.1 channel audio connection example (Figure 24-3), because the secondary CODEC has CODEC ID of '3', the software must write '3' into ACREGACC.CODECID field when it issues secondary CODEC register access.

24.3.5. Sample-data Transmission and Reception

This section describes the mechanism for transmission and reception of PCM audio and modem wave-data. An overview is described first. The DMA (Direct Memory Access) operation, error detection and recovery procedure follow. A special case using slot activation control is described last.

24.3.5.1. Overview

Figure 24-7 and Figure 24-8 show conceptual views of the sample-data transmission and reception mechanisms.

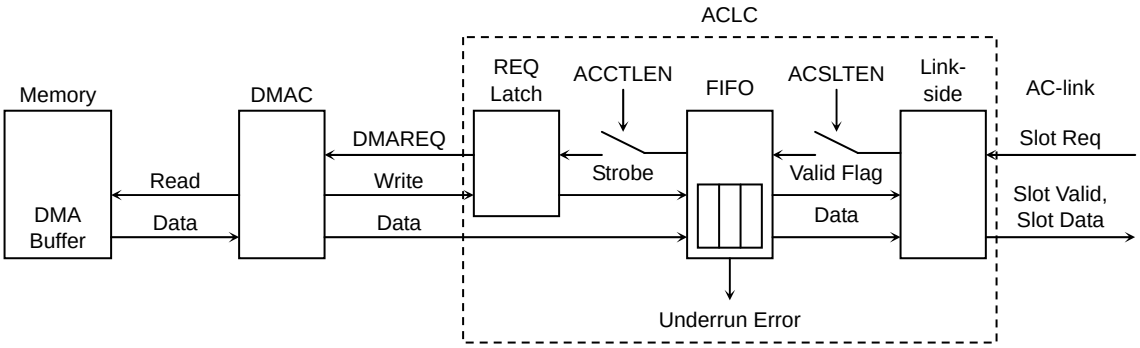


Figure 24-7 Sample-data Transmission Mechanism

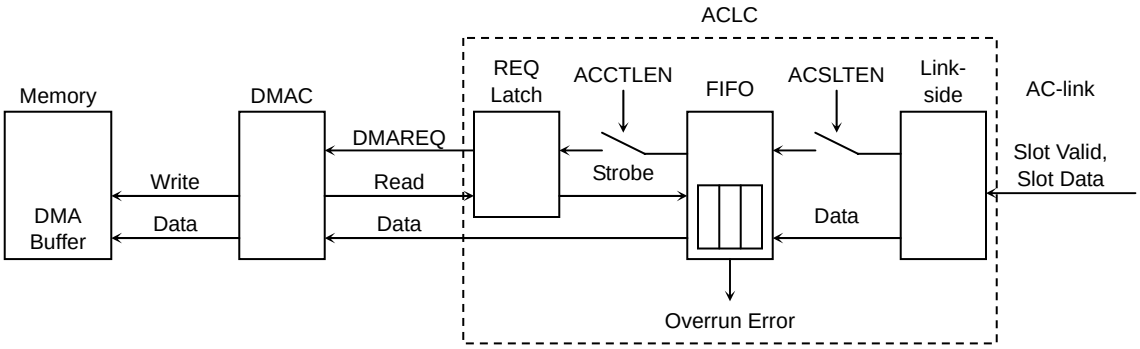


Figure 24-8 Sample-data Reception Mechanism

The CODEC requests ACLC to transmit and receive sample-data via ‘slot-request’ and ‘slot-valid’ bit-fields on the SDIN signal of AC-link.

For transmission, ACLC transmits the data with ‘slot-valid’ tag set. For reception, ACLC captures the slot-data.

Transmission or reception through each stream can be independently activated or deactivated under control of ACLC Slot Enable Register (ACSLTEN).

ACLC is equipped with a separate FIFO for each data-stream. The data to transmit is prefetched from memory to FIFO through DMA. The received data is buffered on FIFO and then stored to memory through DMA. In this stage, each DMA is independently activated or deactivated under control of ACLC Control Enable Register (ACCTLEN).

24.3.6. DMA Channel Mapping

ACLC uses four DMA request channels. These DMA channels are allocated to four out of seven data-streams, or slots, on the AC-link frame, according to ACLC DMA Channel Selection Register (ACDMASEL) setting as shown in Table 24-1. The pin configuration register allocates these DMA channels of ACLC to the DMAC (DMA controller) channels according to Pin Configuration Register (PCFG)'s DMA Request Selection (DMASEL[7:0]) bits as described in section 5.1.3.

Table 24-1 DMA Channel Mapping Modes

	ACDMASEL			
AC-link Slot Number	**0**	**1**	**2**	**3**
PCM L&R out (3&4)	ACLC ch0	ACLC ch0	ACLC ch0	ACLC ch0
Surround L&R out (7&8)		ACLC ch1	ACLC ch1	ACLC ch1
Center out (6)			ACLC ch2	
LFE out (9)			ACLC ch3	ACLC ch3
PCM L&R in (3&4) or Mic in (6)	ACLC ch1			ACLC ch2
Modem Line1 out (5)	ACLC ch2	ACLC ch2		
Modem Line 1 in (5)	ACLC ch3	ACLC ch3		

24.3.6.1. Sample-data Format

ACLC transmits/receives 16 bits per sample for each data slot shown in Table 24-1. The data resides on the first 16 bits of the 20 bits assigned to each slot on AC-link. Each sample-data register allows access by word (32-bit) unit only. Therefore the DMA count must be a multiple of word. Note that the transmit-data DMA count also must be the FIFO depth (refer to ***) or more for a reason described later.

For audio PCM front and surround streams, every data-word is loaded with a couple of left and right samples. For audio MIC stream, valid data is loaded in the same field as the left sample while the other field is filled with '0'. For audio center, LFE, and modem line 1 streams, two consecutive samples are packed into every word.

The data format at the sample-data register is arranged so that the data format on the DMA buffer follows the rules below.

- *Each sample data is put in the byte order in which the CPU operates (big- or little-endian).*
- *Samples are put in the time-sequential order at increasing addresses on memory.*
- *For a DMA channel which couples left and right samples, each left sample precedes the corresponding right sample.*

Refer to the later section for the register format.

Figures below show the format of DMA buffer for each type of DMA channel. #0, #1, … means the sample's sequential number for the AC-link slot. Subscript 'L' means lower 8-bit of each sample and subscript 'H' means upper 8-bit.

Table 24-2 Front and Surround DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_L	Left#0_H	Right#0_L	Right#0_H
+4	Left#1_L	Left#1_H	Right#1_L	Right#1_H
+8	Left#2_L	Left#2_H	Right#2_L	Right#2_H
:	:	:	:	:

Table 24-3 Center, LFE, and Modem DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	#1_L	#1_H
+4	#2_L	#2_H	#3_L	#3_H
+8	#4_L	#4_H	#5_L	#5_H
:	:	:	:	:

Table 24-4 Mic DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	0	0
+4	#1_L	#1_H	0	0
+8	#2_L	#2_H	0	0
:	:	:	:	:

Table 24-5 Front and Surround DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_H	Left#0_L	Right#0_H	Right#0_L
+4	Left#1_H	Left#1_L	Right#1_H	Right#1_L
+8	Left#2_H	Left#2_L	Right#2_H	Right#2_L
:	:	:	:	:

Table 24-6 Center, LFE, and Modem DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_H	#0_L	#1_H	#1_L
+4	#2_H	#2_L	#3_H	#3_L
+8	#4_H	#4_L	#5_H	#5_L
:	:	:	:	:

Table 24-7 Mic DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
z	#0_H	#0_L	0	0
+4	#1_H	#1_L	0	0
+8	#2_H	#2_L	0	0
:	:	:	:	:

24.3.6.2. DMA Operation

When ACLC's REQ latch (refer to Figure 24-7 and Figure 24-8) needs to read or write sample-data, it issues a DMA request. When DMAC acknowledges the request by performing write- or read-access to the ACLC sample-data register, ACLC deasserts the request. Therefore, the software must properly set up DMAC so that the source or destination points to the corresponding sample-data register for the DMA channel.

Setup the DMA Channel Control Registers (DMCCRn) in DMAC as follows.

Immediate chain	Enable	DMCCRn.IMMCHN = 1 [Note]
DMA request polarity	Low-active	DMCCRn.REQPOL = 0
DMA acknowledge polarity	Low-active	DMCCRn.ACKPOL = 0
Request sense	Level-sensitive	DMCCRn.EGREQ = 0
Sample chain	1 word	DMCCRn.SMPCHN = 1
Transfer size	1 word	DMCCRn.XFSZ = 010b
Transfer address mode	Dual	DMCCRn.SNGAD = 0

Note: Use this setting when DMA chain operation is utilized

For a transmission channel, assign the address of ACLC Audio PCM Output/Surround/Center/LFE/Modem Output Register (ACAUDO/SURR/CENT/LFE/MODODAT) to the DMAC destination address register (DMDARn). For a reception channel, assign the address of ACLC Audio input/Modem Input Register (ACAUDI/MODIDAT) to the DMAC source address register (DMSARn).

When any DMA request is pending, the REQ latch will not deasserted the request until the corresponding sample-data register is accessed. Just unsetting ACLC Control Enable Register (ACCTLEN)'s DMA Enable (xxxxDMA) bit corresponding to the DMA will not clear the REQ latch.

The procedure to continuously push or pull the sample-data stream through the chain DMA operation follows the DMAC specification. Refer to section 8.3.10 for this respect.

24.3.6.3. Sample-data FIFO

For a transmission stream, as long as ACLC Control Enable Register (ACCTLEN) allows that transmission and the FIFO has any room to fill data in, the FIFO issues a request via the REQ latch. On the other side, when a transmission FIFO receives a data-request from the link-side, it provides data with valid-flag set if it has any valid data. If it has no valid data, it responds with valid-flag unset and an underrun error bit is set.

At the transmit-data DMA start-up, until the FIFO becomes full, it responds to the link-side with valid-flag unset, in order to maximize the buffering effect. Therefore, the DMA size must be the FIFO depth or more.

Table 24-8 Transmission FIFO Depth

Data-stream	FIFO Depth (Word)
PCM L&R out	3
Surround L&R out	3
Center out	2
LFE out	2
Modem Line 1 out	1

The link-side drives the slot-valid bit and slot-data on AC-link. When underrun occurs, these bits are driven to all '0'.

For a reception stream, as long as the FIFO has any valid data, the FIFO issues a request via the REQ latch. On the other side, when ACCTLEN allows that reception and the link-side issues a data strobe, the FIFO stores the valid data. If the FIFO is full when it receives a data strobe, the data is discarded and an overrun error bit is set.

24.3.6.4. Error Detection and Recovery

In most usages, since the CODEC continuously requests sample-data transmission and reception, after DMA is finished, underrun and overrun will occur. The procedure described below allows the software to determine whether an error has occurred during DMA operation.

The software sets ACLC Control Enable Register (ACCTLEN)'s Error Halt Enable (xxxxEHLT) bit before it starts a DMA channel. After it starts the DMA channel, it waits until ACLC Interrupt Status Register (ACINTSTS)'s Underrun or Overrun Error (xxxxERR) bit is set. When the event is detected, the software checks DMA Channel Control Register (DMCCRn)'s Transfer Active (XFACT) bit and ACLC DMA Request Status Register (ACDMASTS)'s Request (xxxxDMA) bit and determines the DMA completion status as follows.

Table 24-9 DMA Completion Status Determination

DMCCRn.XFACT	ACDMASTS.xxxxDMA	Completion Status
Inactive	Pending	No Error during DMA
Inactive	Not Pending	Underrun or Overrun
Active	*	Underrun or Overrun

To recover from error, disable and enable the stream via ACCTLEN, and restart the DMA.

24.3.6.5. Slot Activation Control

In case ACLC is required to begin transmission or reception of multiple streams at the same time, slot activation control will be useful. To use this feature, the software must deactivate the relevant streams first, enable ACLC Control Enable Register (ACCTLEN), make sure the transmission FIFO becomes full by checking ACLC FIFO Status Register (ACFIFOSTS)'s Full (xxxxFULL) bit, and finally enable ACLC Slot Enable Register (ACSLTEN). This procedure assures that all the reception streams are activated at a frame and all the transmission streams begin to respond to the slot-request bits of that frame.

Note that access to ACSLTEN and ACLC Slot Disable Register (ACSLTDIS) needs special care to synchronize with the link-side. Refer to the register description for detail.

Since operating ACCTLEN register and DMAC without touching ACSLTEN is sufficient for most usages, the initial ACSLTEN value enables all the transmission and reception through the slots by default.

24.3.6.6. Variable Rate Limitation

To improve compatibility with existing AC'97 CODECs and controllers on the market, ACLC combines sample-data for the slots 3 and 4 into one DMA channel, and similarly for the slots 7 and 8. This feature effectively considers that the slot request bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame, and similarly for the slots 7 and 8. ACLC also considers that the slot valid bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame.

24.3.7. GPIO Operation

ACLC supports the slot 12 for the MC'97 (Modem Codec) GPIO.

The slot 12 is shadowed in the ACLC GPI Data Register (ACGPIDAT) and ACLC GPO Data Register (ACGPODAT) in the following way:

- *ACLC copies the slot 12 input data into the ACGPIDAT register, if the slot 12 input is marked by the CODEC as valid in the AC-link frame period.*
- *ACLC generates the slot 12 output data from the ACGPODAT register and mark it as valid, if the slot 12 is required from the CODEC in the previous AC-link frame.*

This shadowing function is enabled as long as ACSLTEN allows.

The bit 0 of the slot 12 is defined as 'GPIO_INT' and can cause ACLC to request an interrupt.

24.3.8. Interrupt

ACLC generate two kinds of interrupt to the interrupt controller as below.

- *ACLC Interrupt*
  Logical OR of all the valid bits of ACLC Interrupt Masked Status Register (ACINTMSTS) is connected. Refer to the section 24.4.5.

- *ACLCPME Interrupt*
  This interrupt shows the wake-up from CODEC in AC-link low-power mode.
  Refer to the description for ACLC Control Enable Register (ACCTLEN)'s Wake-up Enable (WAKEUP) bit in section 24.4.6.

24.3.9. AC-link Low-power Mode

The AC'97 specification makes provision for saving power during system suspension by powering-down both the controller and CODEC except the minimum circuit to detect modem RING/Caller-ID event and wake up the system. AC'97 CODEC is required to go into the low-power mode when they receive a special register-write access. In this mode, the AC-link controller must drive all output signals to low level to allow the CODEC digital I/O power cut.

ACLC provides 'AC-link low-power mode' setting. When this mode is enabled by ACLC Control Enable Register (ACCTLEN)'s Enable AC-link Low-power Mode (LOWPWR) bit, all the output signals except the ACRESET* signal to the AC-link are forced to low level.

The AC-link will be reactivated out of the low-power mode when the SYNC signal is driven high for 1 μs or longer by the AC-link controller while the BITCLK signal is inactive. The software is responsible for controlling the length of this period.

ACLC also provides the 'wake-up' function. While this function is enabled by ACCTLEN Register's Enable Wake-up (WAKEUP) bit, high-level input at any SDIN[x] signal will force ACLCPME interrupt assertion.

When ACLCPME interrupt is recognized, the software must disable the low-power mode and assert warm reset to the AC-link via ACCTLEN Register's Enable Warm Reset (WRESET) bit. After the warm reset is deasserted, the CODEC will start providing the BITCLK signal, and then ACLC will generate the SYNC signal for usual AC-link frames.

Refer to section B.5.1 of AC'97 specification revision 2.1 for the power-down and wake-up sequence in AC-link power-down mode.

24.4. Registers

The base address for the ACLC registers is described in section 4.2. Only word (32-bit) accesses are allowed. These registers return to their initial values when the module gets reset by power-on or configuration-register operation. The 'Disable AC-link' operation initializes the ACREGACC, ACGPIDAT, ACGPODAT, and ACSLTEN registers while keeping the other registers.

Do not access any location which is not mentioned in this section.

All the register bits marked as 'Reserved' are reserved. The value of the reserved bit when read is undefined. When any register is written, write to the reserved bit(s) the same value as the previous value read.

Table 24-10 ACLC Registers

Address	Mnemonic	Register Name	Type	Initial Value
0xF700	ACCTLEN	ACLC Control Enable Register	R/W1S	0x00000000
0xF704	ACCTLDIS	ACLC Control Disable Register	W1C	—
0xF708	ACREGACC	ACLC CODEC Register Access Register	R/W	0x00000000
0xF710	ACINTSTS	ACLC Interrupt Status Register	R/W1C	0x00000010
0xF714	ACINTMSTS	ACLC Interrupt Masked Status Register	R	0x00000000
0xF718	ACINTEN	ACLC Interrupt Enable Register	R/W1S	0x00000000
0xF71C	ACINTDIS	ACLC Interrupt Disable Register	W1C	—
0xF720	ACSEMAPH	ACLC Semaphore Register	RS/WC	0x00000000
0xF740	ACGPIDAT	ACLC GPI Data Register	R	0x00000000
0xF744	ACGPODAT	ACLC GPO Data Register	R/W	0x00000000
0xF748	ACSLTEN	ACLC Slot Enable Register	R/W1S	0x000003DF
0xF74C	ACSLTDIS	ACLC Slot Disable Register	W1C	—
0xF750	ACFIFOSTS	ACLC FIFO Status Register	R	0x00000000
0xF780	ACDMASTS	ACLC DMA Request Status Register	R	0x00000000
0xF784	ACDMASEL	ACLC DMA Channel Selection Register	R/W	0x00000000
0xF7A0	ACAUDODAT	ACLC Audio PCM Output Data Register	W	—
0xF7A4	ACSURRDAT	ACLC Surround Data Register	W	—
0xF7A8	ACCENTDAT	ACLC Center Data Register	W	—
0xF7AC	ACLFEDAT	ACLC LFE Data Register	W	—
0xF7B8	ACMODODAT	ACLC Modem Output Data Register	W	—
0xF7B0	ACAUDIDAT	ACLC Audio PCM Input Data Register	R	0xXXXXXXXX
0xF7BC	ACMODIDAT	ACLC Modem Input Data Register	R	0xXXXXXXXX
0xF7FC	ACREVID	ACLC Revision ID Register	R	0x00000203

24.4.1. ACLC Control Enable Register 0xF700

This register is used to check the setting of various ACLC features and to enable them.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name:	Reserved								MODIEHLT	MODOEHLT	Reserved	AUDIEHLT	LFEEHLT	CENTEHLT	SURREHLT	AUDOEHLT
R/W:									R/W1S	R/W1S		R/W1S	R/W1S	R/W1S	R/W1S	R/W1S
Default:									0	0		0	0	0	0	0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name:	MODIDMA	MODODMA	Reserved	AUDIDMA	LFEDMA	CENTDMA	SURRDMA	AUDODMA	Reserved		RDYCLR	MICSEL	WRESET	WAKEUP	LOWPWR	ENLINK
R/W:	R/W1S	R/W1S		R/W1S	R/W1S	R/W1S	R/W1S	R/W1S		W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S
Default:	0	0		0	0	0	0	0		0	0	0	0	0	0	0

Figure 24-9 ACCTLEN Register

Table 24-11 ACCTLEN Register

Bit	Mnemonic	Field Name	Description	R/W
31:24	—	Reserved		—
23	MODIEHLT	Enable Modem Receive-data DMA Error Halt	MODIEHLT: Enable Modem Receive-data DMA Error Halt. R 0: Indicates that MODIDMA error halt is disabled. 1: Indicates that MODIDMA error halt is enabled. W1S 0: No effect 1: Enables MODIDMA error halt. When MODIDMA overrun occurs, subsequent DMA will not be issued.	R/W1S
22	MODOEHLT	Enable Modem Transmit-data DMA Error Halt	MODOEHLT: Enable Modem Transmit-data DMA Error Halt. R 0: Indicates that MODODMA error halt is disabled. 1: Indicates that MODODMA error halt is enabled. W1S 0: No effect 1: Enables MODODMA error halt. When MODODMA underrun occurs, subsequent DMA will not be issued.	R/W1S
21	—	Reserved		—
20	AUDIEHLT	Enable Audio Receive-data DMA Error Halt	AUDIEHLT: Enable Audio Receive-data DMA Error Halt. R 0: Indicates that AUDIDMA error halt is disabled. 1: Indicates that AUDIDMA error halt is enabled. W1S 0: No effect 1: Enables AUDIDMA error halt. When AUDIDMA overrun occurs, subsequent DMA request will not be issued.	R/W1S
19	LFEEHLT	Enable Audio LFE Transmit-data DMA Error Halt	LFEEHLT: Enable Audio LFE Transmit-data DMA Error Halt. R 0: Indicates that LFEDMA error halt is disabled. 1: Indicates that LFEDMA error halt is enabled. W1S 0: No effect 1: Enables LFEDMA error halt. When LFEDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S

Table 24-11 ACCTLEN Register

Bit	Mnemonic	Field Name	Description	R/W
18	CENTEHLT	Enable Audio Center Transmit-data DMA Error Halt	CENTEHLT: Enable Audio Center Transmit-data DMA Error Halt. R 0: Indicates that CENTDMA error halt is disabled. 1: Indicates that CENTDMA error halt is enabled. W1S 0: No effect 1: Enables CENTDMA error halt. When CENTDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
17	SURREHLT	Enable Audio Surround L&R Transmit-data DMA Error Halt	SURREHLT: Enable Audio Surround L&R Transmit-data DMA Error Halt. R 0: Indicates that SURRDMA error halt is disabled. 1: Indicates that SURRDMA error halt is enabled. W1S 0: No effect 1: Enables SURRDMA error halt. When SURRDMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
16	AUDOEHLT	Enable Audio PCM L&R Transmit-data DMA Error Halt	AUDOEHLT: Enable Audio PCM L&R Transmit-data DMA Error Halt. R 0: Indicates that AUDODMA error halt is disabled. 1: Indicates that AUDODMA error halt is enabled. W1S 0: No effect 1: Enables AUDODMA error halt. When AUDODMA underrun occurs, subsequent DMA request will not be issued.	R/W1S
15	MODIDMA	Enable Modem Receive-data DMA	MODIDMA: Enable Modem Receive-data DMA. R 0: Indicates that modem receive-data DMA is disabled. 1: Indicates that modem receive-data DMA is enabled. W1S 0: No effect 1: Enables modem receive-data DMA.	R/W1S
14	MODODMA	Enable Modem Transmit-data DMA	MODODMA: Enable Modem Transmit-data DMA. R 0: Indicates that modem transmit-data DMA is disabled. 1: Indicates that modem transmit-data DMA is enabled. W1S 0: No effect 1: Enables modem transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]	R/W1S
13	—	Reserved		—
12	AUDIDMA	Enable Audio Receive-data DMA	AUDIDMA: Enable Audio Receive-data DMA. R 0: Indicates that audio receive-data DMA is disabled. 1: Indicates that audio receive-data DMA is enabled. W1S 0: No effect 1: Enables audio receive-data DMA.	R/W1S
11	LFEDMA	Enable Audio LFE Transmit-data DMA	LFEDMA: Enable Audio LFE Transmit-data DMA. R 0: Indicates that audio LFE transmit-data DMA is disabled. 1: Indicates that audio LFE transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio LFE transmit-data DMA. Note : DMA size must be internal FIFO depth or more.	R/W1S

Table 24-11 ACCTLEN Register

Bit	Mnemonic	Field Name	Description	R/W
10	CENTDMA	Enable Audio Center Transmit- data DMA	CENTDMA: Enable Audio Center Transmit-data DMA. R 0: Indicates that audio Center transmit-data DMA is disabled. 1: Indicates that audio Center transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Center transmit-data DMA. Note : DMA size must be internal FIFO depth or more.	R/W1S
9	SURRDMA	Enable Audio Surround L&R Transmit-data DMA	SURRDMA: Enable Audio Surround L&R Transmit-data DMA. R 0: Indicates that audio Surround L&R transmit-data DMA is disabled. 1: Indicates that audio Surround L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Surround L&R transmit-data DMA. Note : DMA size must be internal FIFO depth or more.	R/W1S
8	AUDODMA	Enable Audio PCM L&R Transmit-data DMA	AUDODMA: Enable Audio PCM L&R Transmit-data DMA. R 0: Indicates that audio PCM L&R transmit-data DMA is disabled. 1: Indicates that audio PCM L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio PCM L&R transmit-data DMA. Note : DMA size must be internal FIFO depth or more.	R/W1S
7:6	—	Reserved		—
5	RDYCLR	Clear CODEC Ready Bit	RDYCLR: Clear CODEC Ready Bit W1C 0: No effect 1: Clear CODEC[1:0] ready bits Note: This bit should only be written to reevaluate the CODEC ready status after power-down command is sent to CODEC.	W1S
4	MICSEL	MIC Selection	MICSEL: MIC Selection. R 0: Indicates that PCM L&R (Slot 3&4) is selected for audio reception. 1: Indicates that MIC (Slot 6) is selected for audio reception. W1S 0: No effect 1: Selects MIC (Slot 6) for audio reception.	R/W1S

Table 24-11 ACCTLEN Register

Bit	Mnemonic	Field Name	Description	R/W
3	WRESET	Assert Warm Reset	WRESET: Assert Warm Reset. R 0: Indicates that warm reset is not asserted. 1: Indicates that warm reset is asserted. W1S 0: No effect 1: Asserts warm reset. Note 1: Do not assert warm reset during normal operation. Note 2: The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).	R/W1S
2	WAKEUP	Enable Wake-up	WAKEUP: Enable Wake-up. R 0: Indicates that wake-up from low-power mode is disabled. 1: Indicates that wake-up from low-power mode is enabled. While any SDIN signal is driven high, ACLC asserts ACLCPME interrupt request to the interrupt controller. W1S 0: No effect 1: Enables wake-up from low-power mode. Note: Do not enable wake-up during normal operation.	R/W1S
1	LOWPWR	Enable AC-link low-power mode	LOWPWR: Enable AC-link Low-power Mode. R 0: SYNC and SDOUT signals are not forced to low. 1: SYNC and SDOUT signals are forced to low. W1S 0: No effect 1: Forces SYNC and SDOUT signals low. Note: Do not enable AC-link low-power mode during normal operation.	R/W1S
0	ENLINK	Enable AC-link	ENLINK: Enable AC-link. R 0: Indicates that the ACRESET* signal to AC-link is asserted. 1: Indicates that the ACRESET* signal to AC-link is not asserted. W1S 0: No effect 1: Deasserts the ACRESET* signal to AC-link Note: The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).	R/W1S

24.4.2. ACLC Control Disable Register 0xF704

This register is used to disable various ACLC features.

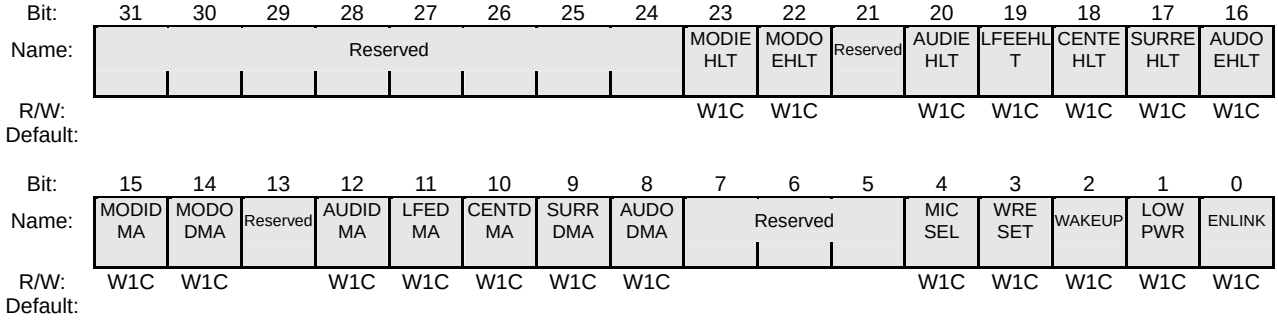


Figure 24-10 ACCTLDIS Register

Table 24-12 ACCTLDIS Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:24	—	Reserved		—
23	MODIEHLT	Disable Modem Receive-data DMA Error Halt	MODIEHLT: Disable Modem Receive-data DMA Error Halt. W1C 0: No effect 1: Disables MODIDMA error halt. MODIDMA request(s) will continue to be issued even after MODIDMA overrun occurs.	W1C
22	MODOEHLT	Disable Modem Transmit-data DMA Error Halt	MODOEHLT: Disable Modem Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables MODODMA error halt. MODODMA request(s) will continue to be issued even after MODODMA underrun occurs.	W1C
21		Reserved		—
20	AUDIEHLT	Disable Audio Receive-data DMA Error Halt	AUDIEHLT: Disable Audio Receive-data DMA Error Halt. W1C 0: No effect 1: Disables AUDIDMA error halt. AUDIDMA request(s) will continue to be issued even after AUDIDMA overrun occurs.	W1C
19	LFEEHLT	Disable Audio LFE Transmit-data DMA Error Halt	LFEEHLT: Disable Audio LFE Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables LFEDMA error halt. LFEDMA request(s) will continue to be issued even after LFEDMA underrun occurs.	W1C
18	CENTEHLT	Disable Audio Center Transmit-data DMA Error Halt	CENTEHLT: Disable Audio Center Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables CENTDMA error halt. CENTDMA request(s) will continue to be issued even after CENTDMA underrun occurs.	W1C
17	SURREHLT	Disable Audio Surround L&R Transmit-data DMA Error Halt	SURREHLT: Disable Audio Surround L&R Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables SURRDMA error halt. SURRDMA request(s) will continue to be issued even after SURRDMA underrun occurs.	W1C

Table 24-12 ACCTLDIS Register

Bit	Mnemonic	Field Name	Description	Read/Write
16	AUDOEHLT	Disable Audio PCM L&R Transmit-data DMA Error Halt	AUDOEHLT: Disable Audio PCM L&R Transmit-data DMA Error Halt. W1C 0: No effect 1: Disables AUDODMA error halt. AUDODMA request(s) will continue to be issued even after AUDODMA underrun occurs.	W1C
15	MODIDMA	Disable Modem Receive-data DMA	MODIDMA: Disable Modem Receive-data DMA. W1C 0: No effect 1: Disables modem receive-data DMA.	W1C
14	MODODMA	Disable Modem Transmit-data DMA	MODODMA: Disable Modem Transmit-data DMA. W1C 0: No effect 1: Disables modem transmit-data DMA.	W1C
13		Reserved		—
12		—	AUDIDMA: Disable Audio Receive-data DMA. W1C 0: No effect 1: Disables audio receive-data DMA.	W1C
11		—	LFEDMA: Disable Audio LFE Transmit-data DMA. W1C 0: No effect 1: Disables audio LFE transmit-data DMA.	W1C
10		—	CENTDMA: Disable Audio Center Transmit-data DMA. W1C 0: No effect 1: Disables audio Center transmit-data DMA.	W1C
9		—	SURRDMA: Disable Audio Surround L&R Transmit-data DMA. W1C 0: No effect 1: Disables audio Surround L&R transmit-data DMA.	W1C
8		—	AUDODMA: Disable Audio PCM L&R Transmit-data DMA. W1C 0: No effect 1: Disables audio PCM L&R transmit-data DMA.	W1C
7:5		Reserved		—
4		—	MICSEL: MIC Selection W1C 0: No effect 1: Selects PCM L&R (Slot 3&4) for audio reception	W1C
3		—	WRESET: Deassert Warm Reset. W1C 0: No effect 1: Deasserts warm reset. Note : The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).	W1C
2		—	WAKEUP: Disable Wake-up. W1C 0: No effect 1: Disables wake-up from low-power mode.	W1C
1		—	LOWPWR: Disable AC-link Low-power Mode. W1C 0: No effect 1: Releases SYNC and SDOUT signals from low.	W1C

Table 24-12 ACCTLDIS Register

Bit	Mnemonic	Field Name	Description	Read/Write
0		—	ENLINK: Disable AC-link. W1C 0: No effect 1: Asserts the ACRESET* signal to AC-link. Note : The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).	W1C

IMPORTANT NOTE

Clear xxxxDMA bits in ACCTLEN to "0" by using this register to disable transmit/receive-data DMA and to stop transmission/reception by the AC-link. Note that if these bits are cleared while output-slot data is flowing in the FIFO, ACLC may output a wrong data as the last sample. This behavior will not occur if the software waits for data-flow completion by detecting underrun before it disables the corresponding slot.

24.4.3. ACLC CODEC Register Access Register 0xF708

CODEC registers can be accessed through this register.

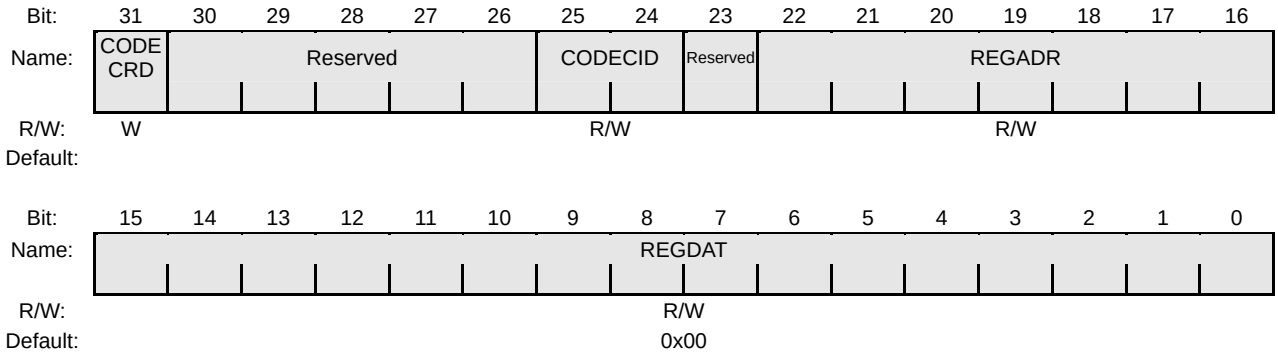


Figure 24-11 ACREGACC

Table 24-13 ACREGACC

Bit	Mnemonic	Field Name	Description	Read/Write
31	CODECRD	AC'97 register read access	CODECRD: AC'97 register read access Write 0: Indicates a write access. 1: Indicates a read access. Note : The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).	W
30:26	—	Reserved		—
25:24	CODECID	AC'97 CODEC ID	CODECID: AC'97 CODEC ID Specifies the CODEC ID of the read/write access destination. Write The values "0" through "3" can be specified as the CODEC ID, but the number of CODECs actually supported depends on the configuration.	W
23	—	Reserved		—
22:16	REGADR	AC'97 register address	REGADR: AC'97 register address Read Read address. Valid address can be read after read access is complete. Write Specifies the read/write access destination address.	R/W
15:0	REGDAT	AC'97 register data	REGDAT: AC'97 register data Read Valid data can be read after read access is complete. Write Write Data	R/W

IMPORTANT NOTE

This register must not be read from or written to until access completion is reported through the ACINTSTS register.

24.4.4. ACLC Interrupt Status Register 0xF710

This register shows various kinds of AC-link and ACLC status.

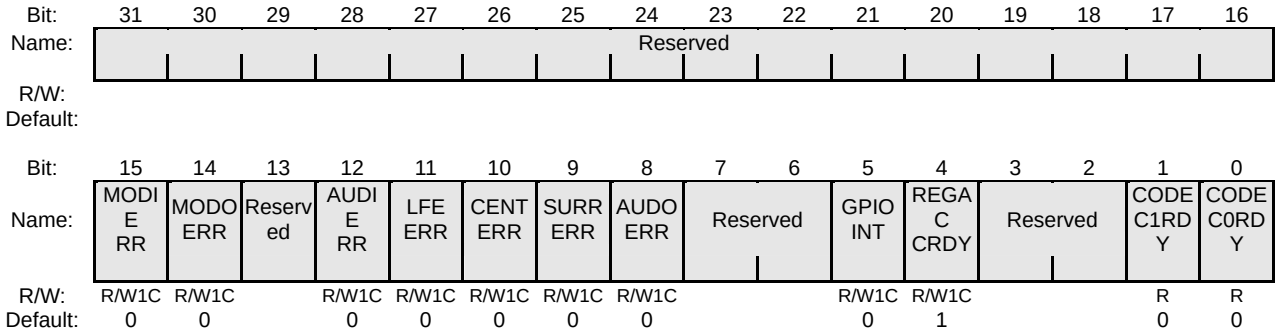


Figure 24-12 ACINTSTS Register

Table 24-14 ACINTSTS Register

Bit	Mnemonic	Field Name	Description	R/W
31:16	—	Reserved		—
15	MODIERR	Modem Receive-data DMA Overrun	MODIERR: Modem Receive-data DMA Overrun Read 0: No activity 1: Indicates that the modem receive-data DMA overran. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
14	MODOERR	Modem Transmit-data DMA Underrun	MODOERR: Modem Transmit-data DMA Underrun Read 0: No activity 1: Indicates that the modem transmit-data DMA underrun. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
13	—	Reserved		—
12	AUDIERR	Audio Receive-data DMA Overrun	AUDIERR: Audio Receive-data DMA Overrun Read 0: No activity 1: Indicates that the audio receive-data DMA overran. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
11	LFEERR	Audio LFE Transmit-data DMA Underrun	LFEERR: Audio LFE Transmit-data DMA Underrun Read 0: No activity 1: Indicates that the audio LFE transmit-data DMA underrun. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
10	CENTERR	Audio Center Transmit-data DMA Underrun	CENTERR: Audio Center Transmit-data DMA Underrun Read 0: No activity 1: Indicates that the audio center transmit-data DMA underrun. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C

Table 24-14 ACINTSTS Register

Bit	Mnemonic	Field Name	Description	R/W
9	SURRERR	Audio Surround L&R Transmit-data DMA Underrun	SURRERR: Audio Surround L&R Transmit-data DMA Underrun Read 0: No activity 1: Indicates that the audio surround L&R transmit-data DMA underrun. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
8	AUDOERR	Audio PCM L&R Transmit-data DMA Underrun	AUDOERR: Audio PCM L&R Transmit-data DMA Underrun Read 0: No activity 1: Indicates that the audio PCM L&R transmit-data DMA underrun. W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
7:6	—	Reserved		—
5	GPIOINT	GPIO Interrupt	GPIOINT: GPIO Interrupt Read 0: No activity 1: Indicates that the incoming slot 12 bit[0] is ‘1’ (the modem CODEC GPIO interrupt). W1C 0: No activity 1: This bit is cleared when “1” is written to it.	R/W1C
4	REGACCRDY	ACREGACC Ready	REGACCRDY: ACREGACC Ready Read 0: No activity 1: Indicates that the ACREGACC register is ready to get the value (in case the previous operation was a read access) and to initiate another R/W access to an AC’97 register. The result of reading or writing to the ACREGACC register before the completion notification is undefined. This bit is cleared if “1” is written to it. W1C 0: No activity 1: This bit automatically becomes ‘0’ when the ACREGACC register is written.	R/W1C
3:2	—	Reserved		—
1	CODEC1RDY	CODEC1 Ready	CODEC1RDY: CODEC1 Ready Read 0: No activity 1: Indicates that the CODEC Ready bit of SDIN1 Slot0 is set.	R
0	CODEC0RDY	CODEC0 Ready	CODEC0RDY: CODEC0 Ready Read 0: No activity 1: Indicates that the CODEC Ready bit of SDIN0 Slot0 is set.	R

24.4.5. ACLC Interrupt Masked Status Register 0xF714

Every bit in this register is configured as follows:

ACINTMSTS = ACINTSTS & ACINTEN

Bit placement is the same as for the ACINTSTS register. The logical OR of all bits in this register is used as ACLC interrupt request to the interrupt controller.

24.4.6. ACLC Interrupt Enable Register 0xF718

Interrupt request enable (R/W1S). Bit placement is the same as for the ACINTSTS register. Its initial value is all '0'. When a value is written to this register, the bit in the position where "1" was written is set to "1."

24.4.7. ACLC Interrupt Disable Register 0xF71C

Interrupt request enable clear (W1C). Bit placement is the same as for the ACINTSTS register. When a value is written to this register, the ACINTEN register bit in the position where a "1" was written is cleared to "0."

24.4.8. ACLC Semaphore Register 0xF720

This register is used for mutual exclusion control for resource.

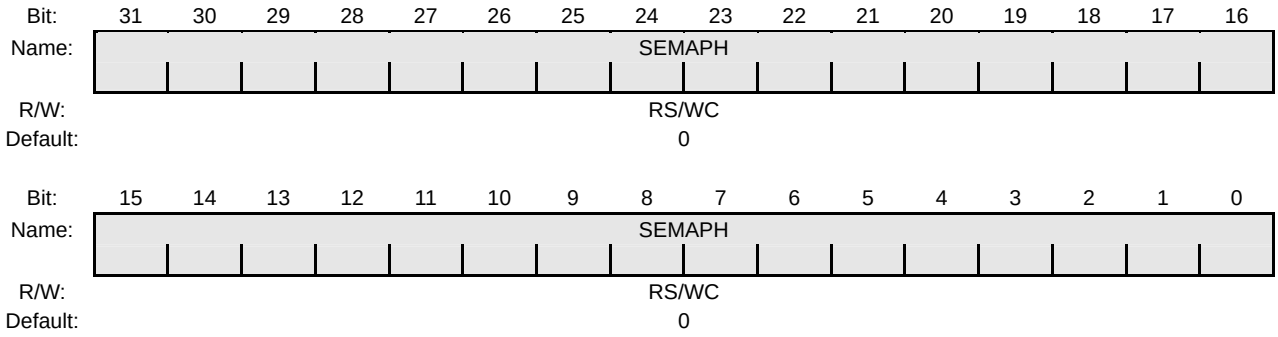


Figure 24-13 ACSEMAPH Register

Table 24-15 ACSEMAPH Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:0	SEMAPH	Semaphore flag	SEMAPH: Semaphore flag. RS 0: Indicates that the semaphore is unlocked. The read operation to this register will atomically set the bit[0] to lock the semaphore. 1: Indicates that the semaphore is locked. WC X: Writing any value to this register clears the bit[0] to release the semaphore.	RS/WC

This register is provided primarily for the mutual exclusion between the audio and modem drivers to share the common resources of ACLC, such as the ACREGACC register and the link-control bits in the ACCTLEN/DIS register.

24.4.9. ACLC GPI Data Register 0xF740

This register shows GPIO (slot 12) input data.

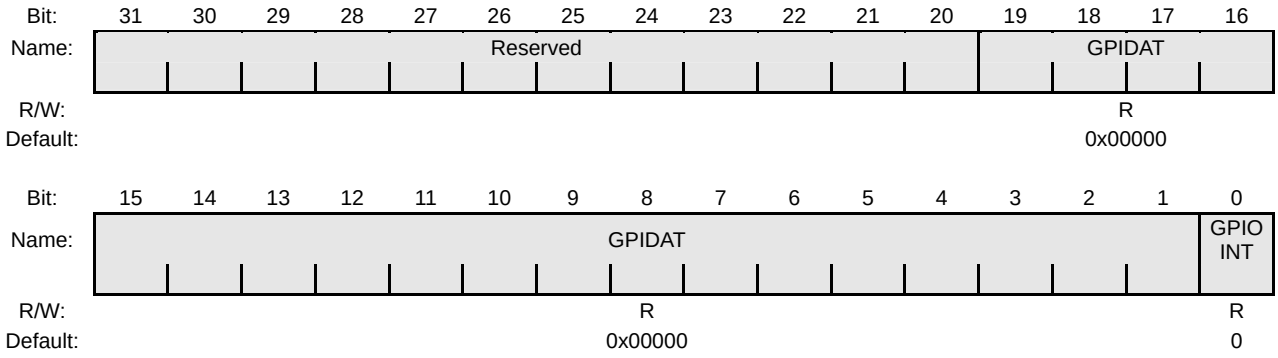


Figure 24-14 ACGPIDAT Register

Table 24-16 ACGPIDAT Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:20	—	Reserved		—
19:1	GPIDAT	GPIO-In data	GPIDAT: GPIO-In data R Read data. The incoming slot 12 bits[19:1] are shadowed here.	R
0	GPIOINT	GPIO Interrupt Indication	GPIOINT: GPIO Interrupt Indication R GPIO Interrupt. The incoming slot 12 bit[0] is shadowed here.	R

24.4.10. ACLC GPO Data Register 0xF744

This register specifies GPIO (slot 12) output data.

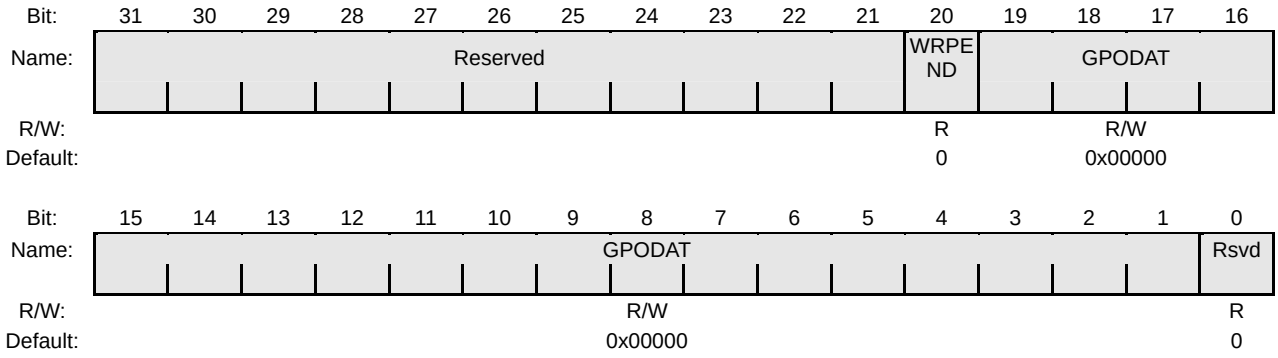


Figure 24-15 ACGPODAT Register

Table 24-17 ACGPODAT Register

Bit	Mnemonic	Field Name	Description	R/W
31:20	—	Reserved		—
20	WRPEND	Write Pending	WRPEND: Write Pending R 0: Indicates that the previous write operation is complete and the ACGPODAT register is ready to be written. 1: Indicates that the previous write operation is not complete and the ACGPODAT register is not yet ready to be written.	R
19:1	GPODAT	GPIO-Out data	GPODAT: GPIO-Out data R Reads back the value previously written to this field. W Writes data to the outgoing slot 12 bits[19:1].	R/W
0	—	—	R Reads always ‘0’.	R

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACGPODAT.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

24.4.11. ACLC Slot Enable Register 0xF748

This register enables independently the AC-link slot data streams.

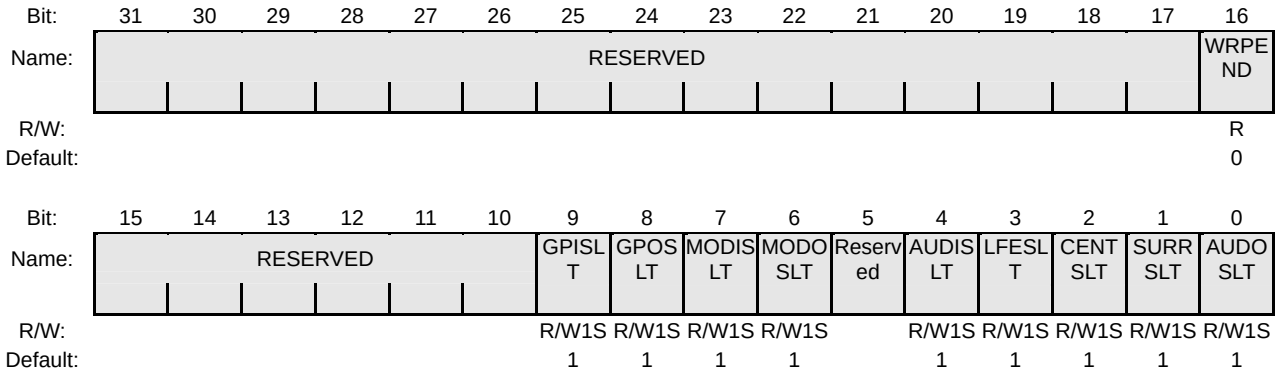


Figure 24-16 ACSLTEN Register

Table 24-18 ACSLTEN Register

Bit	Mnemonic	Field Name	Description	R/W
31:17	—	Reserved		—
16	WRPEND	Write Pending	WRPEND: Write Pending R 0: Indicates that the previous write operation is complete and the ACSLTEN and ACSLTDIS registers are ready to be accessed. 1: Indicates that the previous write operation is not complete and the ACSLTEN and ACSLTEDIS registers are not yet ready to be accessed.	R
15:10	—	Reserved		—
9	GPISLT	Enable GPI slot reception	GPISLT: Enable GPI slot reception. R 0: Indicates that GPI slot reception is disabled. 1: Indicates that GPI slot reception is enabled. W1S 0: No effect 1: Enables GPI slot reception.	R/W1S
8	GPOSLT	Enable GPO Slot transmission	GPOSLT: Enable GPO Slot transmission. R 0: Indicates that GPO slot transmission is disabled. 1: Indicates that GPO slot transmission is enabled. W1S 0: No effect 1: Enables GPO slot transmission.	R/W1S
7	MODISLT	Enable Modem slot reception	MODISLT: Enable Modem slot reception. R 0: Indicates that modem slot reception is disabled. 1: Indicates that modem slot reception is enabled. W1S 0: No effect 1: Enables modem slot reception.	R/W1S

Table 24-18 ACSLTEN Register

Bit	Mnemonic	Field Name	Description	R/W
6	MODOSLT	Enable Modem slot transmission	MODOSLT: Enable Modem slot transmission. R 0: Indicates that modem slot transmission is disabled. 1: Indicates that modem slot transmission is enabled. W1S 0: No effect 1: Enables modem slot transmission.	R/W1S
5	—	Reserved		—
4	AUDISLT	Enable Audio slot reception	AUDISLT: Enable Audio slot reception. R 0: Indicates that audio slot reception is disabled. 1: Indicates that audio slot reception is enabled. W1S 0: No effect 1: Enables audio slot reception.	R/W1S
2	CENTSLT	Enable Audio Center slot transmission	CENTSLT: Enable Audio Center slot transmission. R 0: Indicates that audio Center slot transmission is disabled. 1: Indicates that audio Center slot transmission is enabled. W1S 0: No effect 1: Enables audio Center slot transmission.	R/W1S
1	SURRSLT	Enable Audio Surround L&R slot transmission	SURRSLT: Enable Audio Surround L&R slot transmission. R 0: Indicates that audio Surround L&R slot transmission is disabled. 1: Indicates that audio Surround L&R slot transmission is enabled. W1S 0: No effect 1: Enables audio Surround L&R slot transmission.	R/W1S
0	AUDOSLT	Enable Audio PCM L&R slot transmission	AUDOSLT: Enable Audio PCM L&R slot transmission. R 0: Indicates that audio PCM L&R Slot transmission is disabled. 1: Indicates that audio PCM L&R Slot transmission is enabled. W1S 0: No effect 1: Enables audio PCM L&R slot transmission.	R/W1S

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

24.4.12. ACLC Slot Disable Register 0xF74C

This register disables independently the AC-link slot data streams.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name:	RESERVED															
R/W:																
Default:																

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name:	RESERVED						GPI SLT	GPO SLT	MODI SLT	MODO SLT	Rsvd	AUDI SLT	LFESL T	CENT SLT	SURR SLT	AUDO SLT
R/W:							W1C	W1C	W1C	W1C		W1C	W1C	W1C	W1C	W1C
Default:																

Figure 24-17 ACSLTDIS Register

Table 24-19 ACSLTDIS Register

Bit	Mnemonic	Field Name	Description	R/W
31:10	—	Reserved		—
9	GPISLT	Disable GPI slot reception	GPISLT: Disable GPI slot reception. W1C 0: No effect 1: Disables GPI slot reception.	W1C
8	GPOSLT	Disable GPO Slot transmission	GPOSLT: Disable GPO Slot transmission. W1C 0: No effect 1: Disables GPO slot transmission.	W1C
7	MODISLT	Disable Modem slot reception	MODISLT: Disable Modem slot reception. W1C 0: No effect 1: Disables modem slot reception.	W1C
6	MODOSLT	Disable Modem slot transmission	MODOSLT: Disable Modem slot transmission. W1C 0: No effect 1: Disables modem slot transmission.	W1C
5	—	Reserved		—
4	AUDISLT	Disable Audio slot reception	AUDISLT: Disable Audio slot reception. W1C 0: No effect 1: Disables audio slot reception.	W1C
3	LFESLT	Disable Audio LFE slot transmission	LFESLT: Disable Audio LFE slot transmission. W1C 0: No effect 1: Disables audio LFE slot transmission.	W1C

Table 24-19 ACSLTDIS Register

Bit	Mnemonic	Field Name	Description	R/W
2	CENTSLT	Disable Audio Center slot transmission	CENTSLT: Disable Audio Center slot transmission. W1C 0: No effect 1: Disables audio Center slot transmission.	W1C
1	SURRSLT	Disable Audio Surround L&R slot transmission	SURRSLT: Disable Audio Surround L&R slot transmission. W1C 0: No effect 1: Disables audio Surround L&R slot transmission.	W1C
0	AUDOSLT	Disable Audio PCM L&R slot transmission	AUDOSLT: Disable Audio PCM L&R slot transmission. W1C 0: No effect 1: Disables audio PCM L&R slot transmission.	W1C

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

24.4.13. ACLC FIFO Status Register 0xF750

This register indicates the AC-link slot data FIFO status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name:	RESERVED															
R/W:																
Default:																

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name:	Reserved	MODOFULL	Reserved		LFEFULL	CENTFULL	SURRFULL	AUDOFULL	MODIFILL	MODOFILL	Reserved	AUDIFILL	LFEFILL	CENTFILL	SURRFILL	AUDOFILL
R/W:		R			R	R	R	R	R	R		R	R	R	R	R
Default:		0			0	0	0	0	0	0		0	0	0	0	0

Figure 24-18 ACFIFOSTS Register

Table 24-20 ACFIFOSTS Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:15	—	Reserved		—
14	MODOFULL	Modem Transmit-data Full	MODOFULL: Modem Transmit-data Full. R 0: Indicates modem transmit-data FIFO is not full. 1: Indicates modem transmit-data FIFO is full.	R
13:12	Reserved	Reserved		—
11	LFEFULL	Audio LFE Transmit-data Full	LFEFULL: Audio LFE Transmit-data Full. R 0: Indicates audio LFE transmit-data FIFO is not full. 1: Indicates audio LFE transmit-data FIFO is full.	R
10	CENTFULL	Audio Center Transmit-data Full	CENTFULL: Audio Center Transmit-data Full. R 0: Indicates audio Center transmit-data FIFO is not full 1: Indicates audio Center transmit-data FIFO is full.	R
9	SURRFULL	Audio Surround L&R Transmit-data Full	SURRFULL: Audio Surround L&R Transmit-data Full. R 0: Indicates audio Surround L&R transmit-data FIFO is not full. 1: Indicates audio Surround L&R transmit-data FIFO is full.	R
8	AUDOFULL	Audio PCM L&R Transmit-data Full	AUDOFULL: Audio PCM L&R Transmit-data Full. R 0: Indicates audio PCM L&R transmit-data FIFO is not full. 1: Indicates audio PCM L&R transmit-data FIFO is full.	R
7	MODIFILL	Modem Receive-data Filled	MODIFILL: Modem Receive-data Filled. R 0: Indicates modem receive-data FIFO is empty. 1: Indicates modem receive-data FIFO is not empty.	R

Table 24-20 ACFIFOSTS Register

Bit	Mnemonic	Field Name	Description	Read/Write
6	MODOFILL	Modem Transmit-data Filled	MODOFILL: Modem Transmit-data Filled. R 0: Indicates modem transmit-data FIFO is empty. 1: Indicates modem transmit-data FIFO is not empty.	R
5	—	Reserved		—
4	AUDIFILL	Audio Receive-data Filled	AUDIFILL: Audio Receive-data Filled. R 0: Indicates audio receive-data FIFO is empty. 1: Indicates audio receive-data FIFO is not empty.	R
3	LFEFILL	Audio LFE Transmit-data Filled	LFEFILL: Audio LFE Transmit-data Filled. R 0: Indicates audio LFE transmit-data FIFO is empty. 1: Indicates audio LFE transmit-data FIFO is not empty.	R
2	CENTFILL	Audio Center Transmit-data Filled	CENTFILL: Audio Center Transmit-data Filled. R 0: Indicates audio Center transmit-data FIFO is empty. 1: Indicates audio Center transmit-data FIFO is not empty.	R
1	SURRFILL	Audio Surround L&R Transmit-data Filled	SURRFILL: Audio Surround L&R Transmit-data Filled. R 0: Indicates audio Surround L&R transmit-data FIFO is empty. 1: Indicates audio Surround L&R transmit-data FIFO is not empty.	R
0	AUDOFILL	Audio PCM L&R Transmit-data Filled	AUDOFILL: Audio PCM L&R Transmit-data Filled. R 0: Indicates audio PCM L&R transmit-data FIFO is empty. 1: Indicates audio PCM L&R transmit-data FIFO is not empty	R

24.4.14. ACLC DMA Request Status Register 0xF780

This register indicates the AC-link slot data DMA request status.

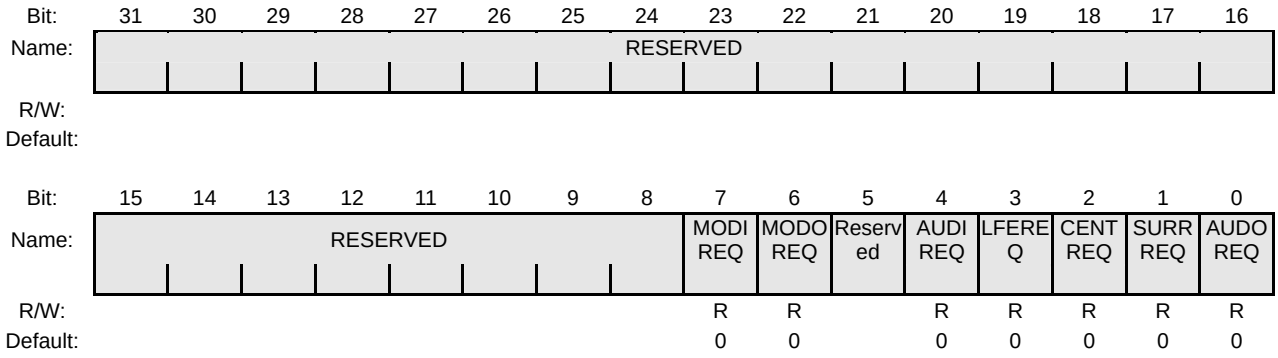


Figure 24-19 ACDMASTS Register

Table 24-21 ACDMASTS Register

Bit	Mnemonic	Field Name	Description	R/W
31:8	—	Reserved		—
7	MODIREQ	Modem Data Reception Request	MODIREQ: Modem Data Reception Request R 0: No request is pending. 1: Request is pending.	R
6	MODOREQ	Modem Data Transmission Request	MODOREQ: Modem Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
5	—	Reserved		—
4	AUDIREQ	Audio Data Reception Request	AUDIREQ: Audio Data Reception Request R 0: No request is pending. 1: Request is pending.	R
3	LFEREQ	Audio LFE Data Transmission Request	LFEREQ: Audio LFE Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
2	CENTREQ	Audio Center Data Transmission Request	CENTREQ: Audio Center Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
1	SURRREQ	Audio Surround L&R Data Transmission Request	SURRREQ: Audio Surround L&R Data Transmission Request R 0: No request is pending. 1: Request is pending.	R
0	AUDOREQ	Audio PCM L&R Data Transmission Request	AUDOREQ: Audio PCM L&R Data Transmission Request R 0: No request is pending. 1: Request is pending.	R

This read-only register shows if any DMA request is pending for each data I/O channel. A DMA request can be pending after the software deactivates the DMAC channel or disables DMA by ACCTLDIS register bit to complete DMA operation. In this case, write or read the sample data register (ACAUDODAT and others) to clear the DMA request.

24.4.15. ACLC DMA Channel Selection Register 0xF784

This register is used to select and check the channel allocation for AC-link slot data DMA.

Figure 24-20 ACDMASEL Register

Table 24-22 ACDMASEL Register

Bit	Mnemonic	Field Name	Description	R/W
31:2	—	Reserved		—
1:0	ACDMASEL	DMA Channel Selection	ACDMASEL: DMA Channel Selection W 0: PCM L&R out, Audio in, and Modem out&in. 1: PCM L&R out, Surround L&R out, and Modem out&in. 2: PCM L&R out, Surround L&R out, Center out, and LFE out. 3: PCM L&R out, Surround L&R out, Center out, and Audio in.	R/W

This register selects DMA channel mapping mode. The software is recommended to make sure no DMA request is pending before changing this register value.

24.4.16. ACLC Audio PCM Output Data Register 0xF7A0
ACLC Surround Data Register 0xF7A4

These registers are used to write audio PCM and surround L&R output data.

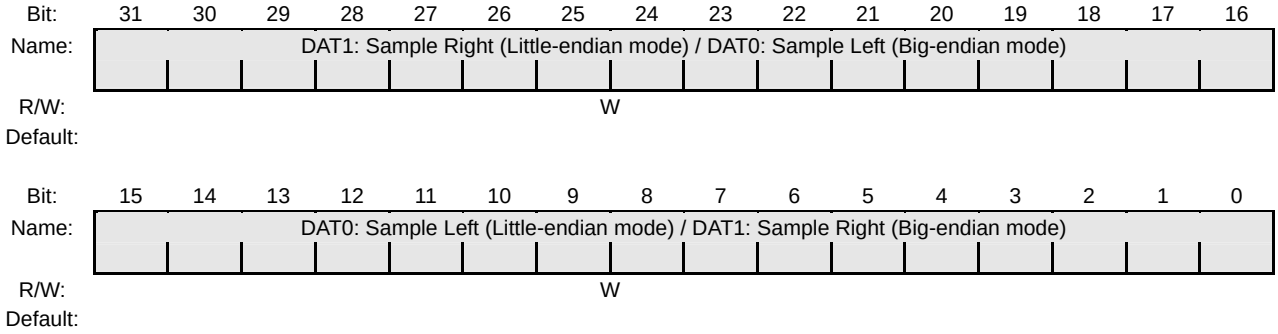


Figure 24-21 ACAUDODAT/ACSURRDAT Register

Table 24-23 ACAUDODAT/ACSURRDAT Register

Bit	Mnemonic	Field Name	Description			Read/Write
31:16	—	—	W	DAT1: Sample Right	DAT0: Sample Left	W
15:0	—	—	W	DAT0: Sample Left	Left DAT1: Sample Right	W

24.4.17. ACLC Center Data Register 0xF7A4
ACLC LFE Data Register 0xF7AC
ACLC Modem Output Data Register 0xF7B8

These registers are used to write audio center, LFE, and modem output data.

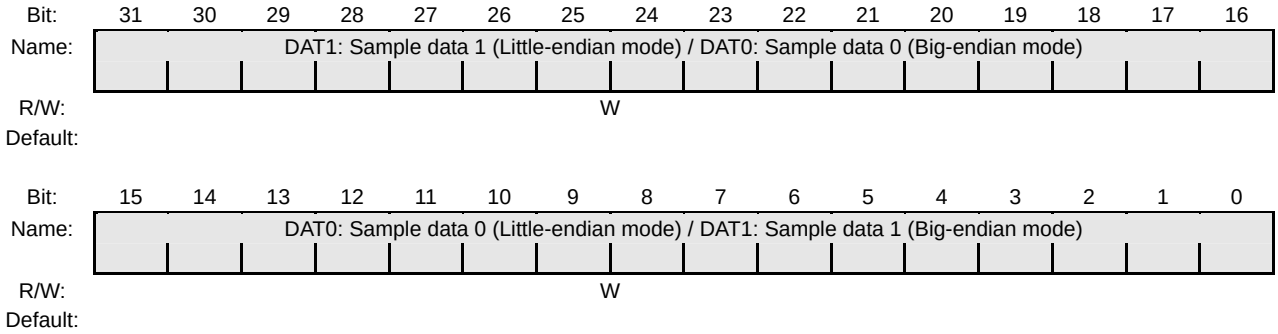


Figure 24-22 ACCENDAT/ACLFEDAT/ACMODODAT Register

Table 24-24 ACCENDAT/ACLFEDAT/ACMODODAT Register

Bit	Mnemonic	Field Name	Description			Read/Write
31:16	—	—	W	DAT1: Sample data 1	DAT0: Sample data 0	W
15:0	—	—	W	DAT0: Sample data 0	DAT1: Sample data 1	W

24.4.18. ACLC Audio PCM Input Data Register 0xF7B0

This register is used to read audio PCM input data.

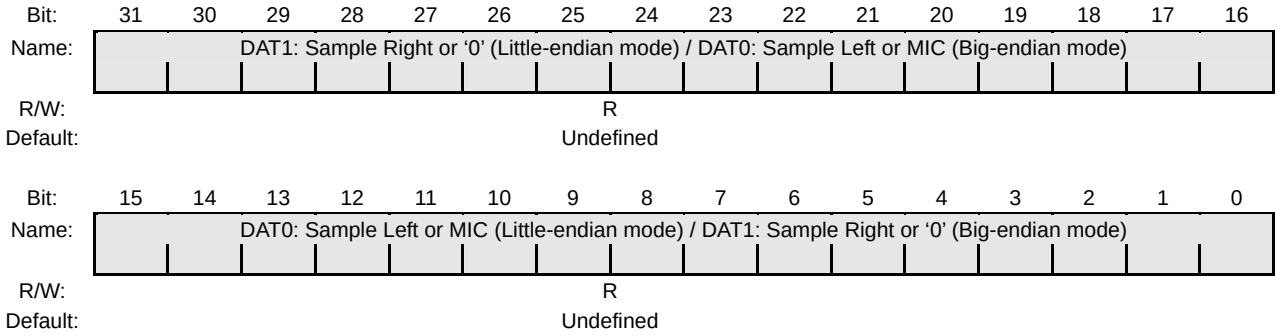


Figure 24-23 ACAUDIDAT Register

Table 24-25 ACAUDIDAT Register

Bit	Mnemonic	Field Name	Description			Read/Write
31:16	—	—	R	DAT1: Sample Right or '0'	DAT0: Sample Left or MIC	R
15:0	—	—	R	DAT0: Sample Left or MIC	DAT1: Sample Right or '0'	R

24.4.19. ACLC Modem Input Data Register 0xF7BC

This register is used to read modem input data.

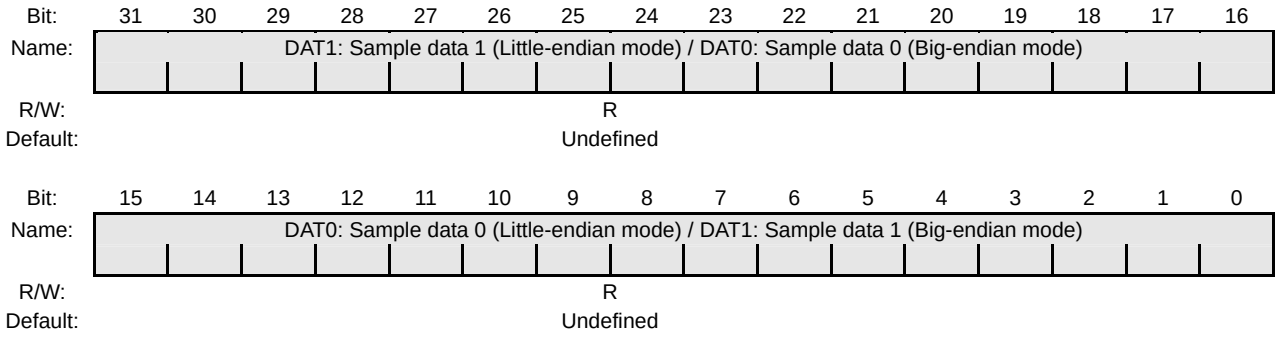


Figure 24-24 ACMODIDAT Register

Table 24-26 ACMODIDAT Register

Bit	Mnemonic	Field Name	Description			Read/Write
31:16	—	—	R	DAT1: Sample data 1	DAT0: Sample data 0	R
15:0	—	—	R	DAT0: Sample data 0	DAT1: Sample data 1	R

24.4.20. ACLC Revision ID Register 0xF7FC

This register is used to read ACLC module's revision ID.

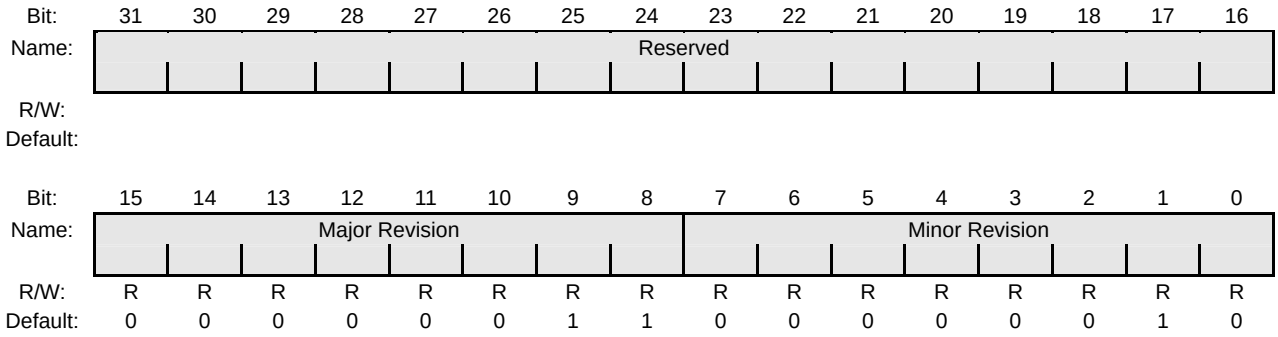


Figure 24-25 ACREVID Register

Table 24-27 ACREVID Register

Bit	Mnemonic	Field Name	Description	Read/Write
31:16	—	Reserved		—
15:8	—	—	Major Revision R Contact Toshiba technical staff for an explanation of the revision value.	R
7:0	—	—	Minor Revision R Contact Toshiba technical staff for an explanation of the revision value.	R

This read-only register shows the revision of ACLC module. Note that this number is not related to the AC'97 specification revision.

Chapter 25. On-Chip SRAM

25.1. Characteristics

The TX4939 has 1-channel of on-chip, fast 2-KB SRAM. You can map it to any address space by setting the Base Address Register. You can access SRAM from each G-Bus Master: the TX49/H4 core, DMAC0, DMAC1, PCI0, and PCI1.

- *2 KB × 1 channel*
- *Specifiable base address*
- *Accessible in Byte, half-word, word, or double-word units*
- *Supports Burst access1*

25.2. Block diagram

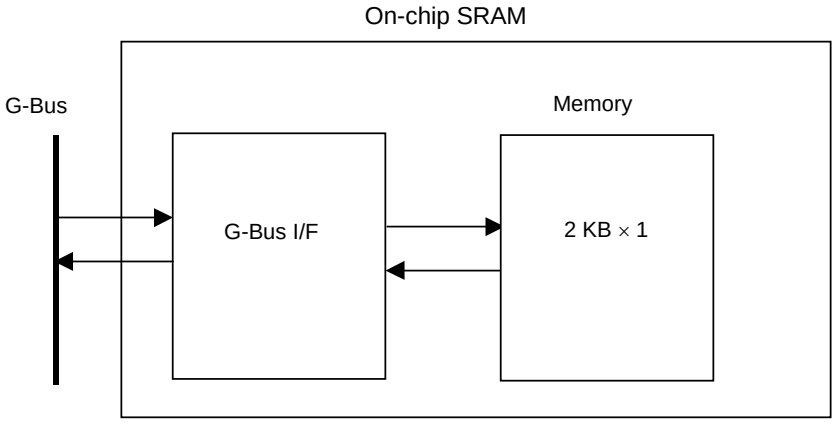


Figure 25-1 On-chip SRAM Block Diagram

1 : On-chip SRAM does not support special Burst access (address decrementing/fixed) of DMAC0, DMAC1.

25.3. Detailed explanation

25.3.1. Base address

Specify the base address of the on-chip SRAM in 2-KB boundary units (specify the upper 25 bits of the physical address). Operation is not guaranteed if the address overlaps with an address space set by SDRAMC, EBUSC, etc.

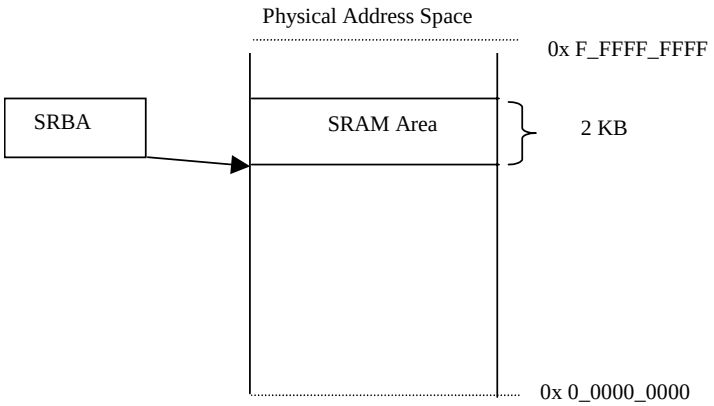


Figure 25-2 Base Address Specification

25.3.2. Access cycle count

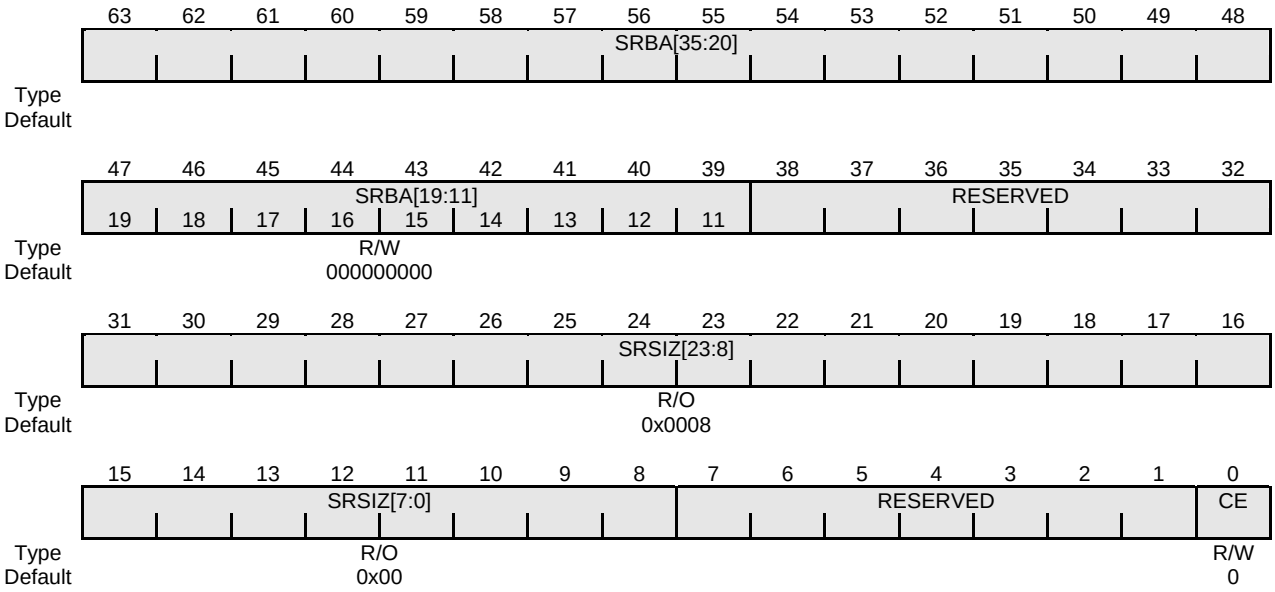
The following table shows the GBUSCLK cycle count required to access on-chip SRAM.

Table 25-1 Access Cycle Count

Access Type	GBUSCLK Cycle Count (n= Burst Size (1-8))
Single Read	3
Single Write	3
Burst Read	2+n
Burst Write	2+n

25.4. Register

25.4.1. On-chip SRAM Control Register 0x6000



Bit(s)	Mnemonic	Field Name	Description
63:39	SRBA [35:11]	On-chip SRAM Base Address	SRAM Base Address (Default: 0x0, R/W) This field specifies the base address of on-chip SRAM. The upper 25 bits [35:11] of the physical address are compared with the value of this field. A 2-KB physical address space with the base address SRBA0[35:11] is mapped to on-chip SRAM.
38:32		Reserved	
31:8	SRSIZ [23:0]	On-chip SRAM Size	SRAM Size (Default: 0x00_0800, R/O) Displays the on-chip SRAM size (2K Bytes).
7:1		Reserved	
0	CE	Channel Enable	Channel enable (Default: 0x0 , R/W) This bit specifies whether to enable a channel. When using on-chip SRAM, set "1". 0: Disable 1: Enable

Figure 25-3 On-Chip SRAM Control Register

Chapter 26. CRYPT Engine

26.1. Feature

The CRYPT Engine consists of three modules, such as CIPHER engine, Modular Exponentiation Operation Coprocessor engine and Random Number Generator (RNG) engine.

26.1.1. CIPHER Engine

- *Supports DES/3DES/AES*
- *Supports MD5/SHA1*
- *Dedicated DMA with chain mode*
- *Reduced KEY exposure for CBC mode in AES*
  (a) 6 set of KEY and Initial-data register
  (b) KEY registers are W/O (write only)
  (c) keep initial data for each stream
  (d) Initial data will be updated for CBC mode
  (e) Decryption-KEY will be generated from Encryption-KEY without exposure
- *Supports Ex-OR operation*

26.1.2. Modular Exponentiation Operation Coprocessor Engine

- *Supports up to 1152bits modular exponentiation operation*

26.1.3. Random Number Generator (RNG) Engine

- Has high degree of randomness

26.2. CIPHER Theory of Operations

Cipher controller supports three types of encryption. They are DES, 3DES and AES algorithms.

Cipher controller also supports two types of hash algorithms. They are MD5 and SHA1.

Cipher controller can perform the following:

a) Only encrypt/decrypt (either using DES, 3DES, or AES algorithm)
b) Only hash (either using MD5 or SHA1 algorithm)
c) Encrypt/Decrypt and hash together (either DES, 3DES or AES with either MD5 or SHA1)

When hash engine is enabled, hash output data length is different for MD5 and SHA1.

- For MD5, hash output data is 128 bits (16 bytes).
- For SHA1, hash output data is 160 bits (20 bytes.)

Cipher controller continuously operates and stops after the completion of the descriptor with Next descriptor address = Null.

DES Input/Output data can start at any address. DES data length must be multiple of 64 bits.
AES Input/Output data can start at any address. AES data length must be multiple of 128 bits.

All Input/Output byte count (IbyteCount/ObyteCount) must be “0” or at least “8 bytes”.
Message of 0 byte count is not supported for hashing.

26.2.1. XOR Function

In the Cipher controller, there are two XOR modes. Mode 0 is to XOR the memory data with a 64 bit register data. Mode 1 is to XOR two memory data and output back the result back to memory location.

Mode 0: There are two XOR registers (upper and lower) to form 64 bits. The Cipher controller will perform an Exclusive OR function between the fetch data (bypass mode) to these 64 XOR register bits. It then will write back the result to the destination location. Note that, if XOR register bits are “0” then the output data will remain the same.

Mode 1: In the Control Descriptor, bit 4 is a select bit for this feature (0: normal, 1: Select XOR mode 1). Once XOR mode 1 is selected, the controller will fetch data from Input Source Address Descriptor and XOR it with the data from XOR Input Source Address Descriptor 2 then it will write back this data to the Output Destination Address Descriptor. In this mode, Output Byte Count and Input Byte Count must be the same.

26.2.2. CIPHER DMA Controller

The CIPHER DMA Controller has the following features:

- *Supports register access*
- *Supports master mode for G-bus protocol (single and burst transfer)*
- *Supports chain and link list DMA protocol*
- *Supports un-alignment byte data transfer for both input and output DMA transfer*

The CIPHER's DMA engine has one channel to transfer data between memory and the CIPHER engine. There are four tasks for the DMA:

1. To get a descriptor
2. To update the status in a descriptor when transfer complete
3. To transfer transmitted data from Transmit FIFO to memory (TxFIFO(8bytes x 16) – output buffer)
4. To transfer received data from memory to Receive FIFO (RxFIFO (8bytes x 16) – input buffer).

26.2.2.1. DMAC Theory of Operations

The CIPHER's DMA engine deploys the following algorithm:

1. Firmware programs the six sets of Keys and Initial data.
2. The ENCR bit of the CSR register is set to start the DMA controller (DMAC).
3. DMAC fetches the first descriptor located at a memory location pointed to by the CDESPtr register.
4. DMAC is ping-pong between input buffer and output buffer to make sure deadlock does not occur.
5. DMAC completes transmit data transfer (when both input and output byte counts reach "0").
6. DMAC proceeds to the next descriptor if the next descriptor pointer is not 'h0000_0000. Otherwise, DMAC will stop and clear the ENCR bit in the CSR register.
7. When G-bus error occurs or Time-out occurs, DMA controller will stop all transfer.

26.2.2.2. CIPHER DMA Controller Registers

All registers are 32 bit. Firmware must use Word instruction to access to these registers.

Table 26-1 Cipher DMA Control Registers

Offset	Register	Width	R / W	Description
00h	CSR	[31:0]	R/W	Control and Status Register
08h	IDESPtr	[31:0]	R/W	Initial Descriptor Pointer Register
10h	CDESPtr	[31:0]	R	Current Cipher Descriptor Pointer Register
18h	BusErr	[31:0]	R	Bus Error Address Register
20h	Cip_tout	[31:0]	R/W	Time out Register
28h	Cir	[31:0]	R/W	Context Index Register
30h	Cdr1	[31:0]	WO	Context Data Register 1
38h	Cdr2	[31:0]	WO	Context Data Register 2
40h	Cdr3	[31:0]	WO	Context Data Register 3
48h	Cdr4	[31:0]	WO	Context Data Register 4
50h	Cdr5	[31:0]	WO	Context Data Register 5
58h	Cdr6	[31:0]	WO	Context Data Register 6
60h	Cdr7	[31:0]	WO	Context Data Register 7
68h	Cdr8	[31:0]	WO	Context Data Register 8
70h	Cdr9	[31:0]	WO	Context Data Register 9
78h	--	--	--	Reserved
80h	--	--	--	Reserved
88h	--	--	--	Reserved
90h	--	--	--	Reserved
98h	--	--	--	Reserved
A0h	--	--	--	Reserved
A8h	--	--	--	Reserved
E0h-E8h	--	--	--	Reserved
F0h	XORSLR0	[31:0]	R/W	XOR Source Lower Register
F8h	XORSUR1	[31:0]	R/W	XOR Source Upper Register

26.2.2.3. Control and Status Register (CSR)

Table 26-2 Control and Status Register (CSR)

Bit(s)	Field	R/W	Default	Description
31:28	–	RO	0	Reserved
27	SAESO	R/W	0	Swap output AES data 0: Swap 1: No swap (for testing only) For both Big or Little endian, this bit must be “0”.
26	SAESI	R/W	0	Swap Input AESdata 0: Swap 1: No swap (for testing only) For both Big or Little endian, this bit must be “0”.
25	SDESO	R/W	0	Swap output DES data 0: Swap 1: No swap (for testing only) For both Big or Little endian, this bit must be “0”.
24	SDESI	R/W	0	Swap Input DES data 0: Swap 1: No swap (for testing only) For both Big or Little endian, this bit must be “0”.
23	--	RO	0	Reserved
22:20	INDXBST	RO	0	Operate Index B Indicator This field indicates the current operate index B register set 000 Engine idle 001 Operating with Context register set #1 010 Operating with Context register set #2 …….. …… 110 Operating with Context register set #6 111 Reserved
19	TOINT	RO	0	Time out Interrupt 0: No interrupt 1: Time out error This bit will be clear after read.

Table 26-2 Control and Status Register (CSR)

Bit(s)	Field	R/W	Default	Description
18	DCINT	RO	0	DMA Completion Interrupt 0: No interrupt 1: Transfer complete This bit will be clear after read.
17	-	-	-	Reserved
16	GBINT	RO	0	G-Bus Error Interrupt 0: No interrupt 1: G-Bus error This bit will be clear after read.
15:13	INDXAST	RO	0	Operate Index A Indicator This field indicates the current operate index A register set 000 Engine idle 001 Operating with Context register set #1 010 Operating with Context register set #2 …….. …….. 110 Operating with Context register set #6 111 Reserved
12:11	CSWAP	R/W	00	Cipher Swap option 00: Normal 01: Swap Input data (for testing only) 10: Swap Output data (for testing only) 11: Swap both Input and Output data (for testing only) When = 2'b00, sysBigEndian control the swap logic. Other selection, these two bits control the swap logic.
10	--	--	0	Reserved.
9	CDIV	R/W	0	modular exponentiation operation coprocessor engine clock select 0: Div 2 (GBUSCLK DIV2) 1: Div 1 (same as GBUSCLK)
8	--	--	0	Reserved.
7:6	PDINT	R/W	01	Programmable DMA Completion Interrupt 00: Interrupt for every descriptor at completion of transfer. 01: Interrupt only when Next Descriptor pointer = Null or when "End_packet" bit set at completion of transfer. 10: Interrupt only when Next Descriptor pointer = Null at completion of transfer. 11: Disable DMA Completion Interrupt
5	--	--	0	Reserved.
4	--	--	0	Reserved.
3	GINTE	R/W	0	Interrupt Enable 0: Disable All Interrupts. 1: Enable All Interrupts. This bit has to be program before start Cipher engine
2	RSTD	R/W	0	Reset DMA Controller 0: Normal 1: Reset After set this bit, need to reset it back to "0" for normal operation.
1	RSTC	R/W	0	Reset Cipher engine 0: Normal 1: Reset After set this bit, need to reset it back to "0" for normal operation.
0	ENCR	R/W	0	Start Cipher Engine 0: Idle 1: Start When write "1" to this bit, Cipher engine will begin to operate.

26.2.2.4. Initial Descriptor Pointer Register (IDESPtr)

Table 26-3 Initial Descriptor Pointer Register (IDESPtr)

Bit(s)	Field	R/W	Default	Description
31:0	--	R/W	32'h0	Initial Descriptor Pointer This pointer will point to the first descriptor address.

26.2.2.5. Current Cipher Descriptor Pointer Register (CDESPtr)

Table 26-4 Current Cipher Descriptor Pointer Register (CDESPtr)

Bit(s)	Field	R/W	Default	Description
31:0	--	RO	32'h00	When Crypt engine is running. This register points to the address of the descriptor that the engine is current processing.

26.2.2.6. Time Out Register (cip_tout)

Table 26-5 Time Out Register (cip_tout)

Bit(s)	Field	R/W	Default	Description
31:8	–	RO	0	Reserved
7:0	--	R/W	8'h00	Programmable Time Out If timer reach 0 before see "ACK" from Cipher engine. Then it will generate Timeout error. Upon receive "ACK" then reload timer and start count down again. 8'h00 = disable timer 8'h01 – 8'hff = timer count These counters clock by GBUSCLK

Notes: This register is for testing purpose only

26.2.2.7. Bus Error Address Register (BusErr)

Table 26-6 Bus Error Address Register (BusErr)

Bit(s)	Field	R/W	Default	Description
31:20	–	RO	0	Reserved
31:2	--	RO	0	G-bus error address[31:2]
1	--	RO	0	Command status during bus error. 0: Write operation 1: Read operation
0	--	RO	0	When set, indicates the address is valid, i.e. Bus Error has been captured.

Notes: need to issues software reset (DMAC reset - RSTD),to clear BusErr register.

26.2.2.8. XOR Source Lower Register 0(XORSLR)

Table 26-7 XOR Source Lower Register 0(XORSLR)

Bit(s)	Field	R/W	Default	Description
31:0	--	R/W	0	XOR Source Lower Register Lower 32 bit of the XOR source data which used to XOR with the bit stream

26.2.2.9. XOR Source Upper Register 0(XORSUR)

Table 26-8 XOR Source Upper Register 0(XORSUR)

Bit(s)	Field	R/W	Default	Description
31:0	--	R/W	0	XOR Source Upper Register Upper32 bit of the XOR source data which used to XOR with the bit stream

26.2.2.10. Context Index Register (cir)

Table 26-9 Context Index Register (cir)

Bit(s)	Field	R/W	Default	Description
31:3	–	R/O	0	Reserved
2:0	Cptr[2:0]	R/W	0	Context Index Pointer 3'b000 = no access 3'b001 = Context register set #1 ……. 3'b110 = Context register set #6 3'b111 = no access

26.2.2.11. Context Data Register (cdr)

Table 26-10 Context Data Register (cdr)

Bit(s)	Field	R/W	Default	Description
31:0	Cdata	WO	0	Context Data register

In the Cipher controller, there are six Context register sets. In order to write to the context register set. Software must write to the Context Index register which select a set (1-6) then write to the Context Data register 1 to Context Data register 9.

The Context Index register contains pointer to the Context Register set (1 to 6). When write to the Context Data register, it will write to the Context Data Register which pointed by Context Index register.

Each context register set can be either for DES, AES or MD5/SHA1 register format. The controller samples des_ctrl[1:0] of the context register set to know which format is this context register set for.

Des_ctrl[1:0]	Configuration
2'b00	DES format
2'b01	AES format
2'b10	MD5 format
2'b11	SHA1 format

26.2.2.12. DES Context Data Register Format

Table 26-11 DES Context Data Register Format

Offset	Width	Name	Description
30h	[31:0]	des_key1l	Key 1 lower register
38h	[31:0]	des_key1u	Key 1 upper register
40h	[31:0]	des_key2l	Key 2 lower register
48h	[31:0]	des_key2u	Key 2 upper register
50h	[31:0]	des_key3l	Key 3 lower register
58h	[31:0]	des_key3u	Key 3 upper register
60h	[31:0]	des_idbusl	Des Initial lower value
68h	[31:0]	des_idbusu	Des Initial upper value
70h	[31:0]	des_ctrl	DES Control register [1:0] Engine select (0: DES, 1 :AES, 2: MD5, 3: SHA1) [3:2] Reserved [4] TDMS (1: Triple DES, 0: Single DES) [5] CMS (1: CBC mode, 0: ECB mode) [6] DMS (1: Decrypt, 0: Encrypt) [31:7] Reserved.

26.2.2.13. AES Context Data Register Format

Table 26-12 AES Context Data Register Format

Offset	Width	Name	Description
30h	[31:0]	aes_key0l	Key 0 register (Lower key)
38h	[31:0]	aes_key1	Key 1 register
40h	[31:0]	aes_key2	Key 2 register
48h	[31:0]	aes_key3u	Key 3 register (Upper key)
50h	[31:0]	aes_idat0	AES initial data 0 (Lower data)
58h	[31:0]	aes_idat1	AES initial data 1
60h	[31:0]	aes_idat2	AES initial data 2
68h	[31:0]	aes_idat3	AES initial data 3 (Upper data)
70h	[31:0]	aes_ctrl	AES Control register [1:0] Engine select (0: DES, 1 :AES, 2: MD5, 3: SHA1) [3:2] Reserved [4] Reserved. [5] CMS (1: CBC mode, 0: ECB mode) [6] DMS (1: Decrypt, 0: Encrypt) [7] Update key option (1: Update 0: no update) Once this bit set, the first decrypt data will be save back to aes_key Calculate_data[31:0] → aes_key0l Calculate_data[63:32] → aes_key1 Calculate_data[95:64] → aes_key2 Calculate_data[127:96] → aes_key3u [31:8] reserved.

Note: 128bit Cipher Key = {aes_key3u[31:0], aes_key2[31:0], aes_key1[31:0], aes_key0l[31:0]}

26.2.2.14. MD5/SHA1 Context Data Register Format

Table 26-13 MD5/SHA1 Context Data Register Format

Offset	Width	Name	Description
30h	[31:0]	Hash_lbcnt	Hash lower bit count Bit[2:0] are fixed to “0”
38h	[31:0]	Hash_ubcnt	Hash upper bit count
40h	[31:0]	Ini_val_a	Initial Value A
48h	[31:0]	Ini_val_b	Initial Value B
50h	[31:0]	Ini_val_c	Initial Value C
58h	[31:0]	Ini_val_d	Initial Value D
60h	[31:0]	Ini_val_e	Initial Value E
68h	[31:0]	---	Reserved.
70h	[31:0]	Ctrl	MD5/SHA1 Control register [1:0] Engine select (0: DES, 1 :AES, 2: MD5, 3: SHA1) [31:2] Reserved

26.3. CIPHER Operation Mode

26.3.1. CBC Mode

There are two mode in DES and AES engine, they are ECB mode and CBC mode. For CBC mode, it needs to use initial data for the first calculation, thereafter it uses the previous output result for the initial data. Therefore in CBC mode of DES and AES, the controller will update the Initial Data field after each calculation.

26.3.2. AES Special Mode

For AES engine, the controller has a special mode that allows the controller to update the KEYs field with the caculated data. With this feature, the original KEY can be encrypted. Then with the AES decrypt mode, it decrypts the encrypted KEY to get the original KEY and update the KEY0,1,2,3 fields.
aes_ctrl[7] is the control register bit for this special mode. After this bit is set, the first output data for this index will be saved back into this index KEY fields

Calculate_data[31:0] → aes_key0
Calculate_data[63:32] → aes_key1
Calculate_data[95:64] → aes_key2
Calculate_data[127:96] → aes_key3

26.4. CIPHER Descriptor

All Cipher descriptors are 32-bit,

Table 26-14 Cipher Descriptor Table

Description	Offset
Input Source Address Descriptor	'h00/'h08
Output Destination Address Descriptor	'h04/'h0c
Next Descriptor Pointer Descriptor	'h08/'h10
Control Descriptor	'h0c/'h14
Index Descriptor	'h10/'h18
XOR Input Source Address Descriptor 2	'h14/'h1c

Note: New descriptor must start at double word boundary.

Table 26-15 Input Source Address Descriptor

Field	Name	Description
31:0	SourceGA[31:0]	Source G-Bus Address[31:0] Byte addressable

Table 26-16 Output Destination Address Descriptor

Field	Name	Description
31:0	DestGA[31:0]	Destination G-Bus Address[31:0] Byte addressable

Table 26-17 Next Descriptor Pointer Descriptor

Field	Name	Description
31:2	nxtptr[31:2]	Address of the next descriptor. Null ('h0000_0000) = End of chain.
1:0	-	Reserved.

Table 26-18 Control Descriptor

Field	Name	Description
31:21	ObyteCount[10:0]	Output Byte Count – up to 2K-1 byte 11'h007-11'h001 not valid (may hang cipher controller) 11'h000 = 0 byte 11'h008 = 8 byte …… 11'h7ff = 2047 bytes Output byte count must be at least "0" or "8 bytes".
20:10	IbyteCount[10:0]	Input Byte Count - up to 2K-1 byte 11'h007-11'h001 not valid (may hang cipher controller) 11'h000 = 0 byte 11'h008 = 8 bytess …… 11'h7ff = 2047 bytes Input byte count must be at least "0" or "8 bytes". Please note that during Hash calculation, input byte count must be more than equal "8 bytes"
9	Start_packet	This bit = H, indicates start of a new packet.
8	End_packet	This bit = H indicates end of this packet. Next descriptor belongs to different packet. When Hash is enable and End_packet is set, this indicates the end for hash algorithm. In this descriptor the ObyteCount[10:0] must be set to the exact # of byte of the authentication data (for MD5 = 16 byte, for SHA1 = 20 bytes)
7:5	---	Reserved
4	Xor_sel	XOR Mode 1 Select. 0: Normal 1: Select XOR Mode 1.
3	Last_data	This bit = H indicates end of this data type. Next descriptor is belong to the next data type.
2:1	Error	Cipher will set these bits when there is an error occurred when process this packet 00 = no error 01 = time out error 10 = error in MD5/SHA1 engine 11 = reserved. When error occurred, DMA engine will skip the rest of the operation until it see end of packet. Before starting process next packet, Cipher controller will issues reset to all engines (DES, AES, MD5, SHA1). If time out occurred, TOINT gets set then interrupt CPU If error in MD5/SHA1 engine, HAINT gets set then interrupt CPU
0	---	Reserved

Table 26-19 Index Descriptor

Field	Name	Description
31:7	---	Reserved
6:4	IndexB[2:0]	IndexB pointer. This index uses for Hash engine. 000 = None, do not hash (pass through) 001 = Select Index1 key set ……. 110 = Select Index6 key set 111 = Reserved
[3]	--	Reserved
2:0	IndexA[2:0]	IndexA pointer. This index uses for Encryption engine. 000 = None, do not encrypt/decrypt (pass through) 001 = Select Index1 key set ……. 110 = Select Index6 key set 111 = Reserved

Notes: If only Encryption then program only IndexA, IndexB = 000
If only Hash then program only IndexB, IndexA = 000
If Encryption and Hash together then program both IndexA and IndexB

26.5. CIPHER Bi-Endian support

DMA controller supports both little endian and big endian format. For the CIPHER engine, we are assume that it only take in little endian format.

26.5.1. Little endian

In memory, all the process data are organized in 64 bit format. The following describes the 64 bits data format for little endian mode

Bits	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]
Byte	7	6	5	4	3	2	1	0

In case of the byte are not align to 8 byte boundary. The next byte data will shift in at the right side of the previous data

Example:

Input Data (from memory)

Data1

Byte	7	6	5	4	3	2	1	0
	valid	valid	valid	valid	valid	x	x	x

Data2

Byte	7	6	5	4	3	2	1	0
	valid	valid	valid	valid	valid	valid	valid	valid

Data3

Byte	7	6	5	4	3	2	1	0
	x	x	valid	valid	valid	valid	valid	valid

Break (transfer complete)

Data4

Byte	7	6	5	4	3	2	1	0
	valid	valid	valid	valid	x	x	x	x

Data5

Byte	7	6	5	4	3	2	1	0
	x	valid	valid	valid	valid	valid	valid	valid

Final Input Data (transfer to Cipher engine)

Byte	7	6	5	4	3	2	1	0
Data1	2-2	2-1	2-0	1-7	1-6	1-5	1-4	1-3
Data2	3-2	3-1	3-0	2-7	2-6	2-5	2-4	2-3
Data3	5-0	4-7	4-6	4-5	4-4	3-5	3-4	3-3
Data4	x	x	5-6	5-5	5-4	5-3	5-2	5-1

26.5.2. Big Endian

In memory, all the process data are organized in 64 bit format. The following describes the 64 bits data format for big endian mode

Bits	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]
Byte	0	1	2	3	4	5	6	7

In case of the byte are not align to 8 byte boundary. The next byte data will shift in at the left side of the previous data

Example:

Input Data (from memory)

Data1

Byte	0	1	2	3	4	5	6	7
	x	x	x	valid	valid	valid	valid	valid

Data2

Byte	0	1	2	3	4	5	6	7
	valid	valid	valid	valid	valid	valid	valid	valid

Data3

Byte	0	1	2	3	4	5	6	7
	valid	valid	valid	valid	valid	valid	x	x

Break (transfer complete)

Data4

Byte	0	1	2	3	4	5	6	7
	x	x	valid	valid	valid	valid	valid	valid

Data5

Byte	0	1	2	3	4	5	6	7
	valid	valid	valid	valid	valid	x	x	x

Final Input Data (before swap to little endian for Cipher engine)

Byte	0	1	2	3	4	5	6	7
Data1	1-3	1-4	1-5	1-6	1-7	2-0	2-1	2-2
Data2	2-3	2-4	2-5	2-6	2-7	3-0	3-1	3-2
Data3	3-3	3-4	3-5	4-2	4-3	4-4	4-5	4-6
Data4	4-7	5-0	5-1	5-2	5-3	5-4	x	x

Final Input Data (transfer to Cipher engine)

Convert to little endian format for Cipher engine

Byte	7	6	5	4	3	2	1	0
Data1	2-2	2-1	2-0	1-7	1-6	1-5	1-4	1-3
Data2	3-2	3-1	3-0	2-7	2-6	2-5	2-4	2-3
Data3	4-6	4-5	4-4	4-3	4-2	3-5	3-4	3-3
Data4	x	x	5-4	5-3	5-2	5-1	5-0	4-7

26.6. CIPHER Programming Descriptor examples.

26.6.1. Example 1

Encrypt/Decrypt 3DES

1st descriptor:	ObyteCount	#	(block data #1)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	0	
	Start_packet	1	
	End_packet	0	
	Last_data	0	
2nd descriptor:	ObyteCount	#	(block data #2)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	0	
	Start_packet	0	
	End_packet	0	
	Last_data	0	
3rd descriptor:	ObyteCount	#	(block data #3 – end of packet)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	0	
	Start_packet	0	
	End_packet	1	
	Last_data	1	

26.6.2. Example 2

Output SHA1 Authentication

1st descriptor:	ObyteCount	0	(block data #1)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	1	
	End_packet	0	
	Last_data	0	
2nd descriptor:	ObyteCount	20	(block data #2 – end of packet)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	1	
	Last_data	1	

26.6.3. Example 3

Encrypt/Decrypt 3DES and Output SHA1 Authentication

1st descriptor:	ObyteCount	#	(block data #1)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	1	
	End_packet	0	
	Last_data	0	
2nd descriptor:	ObyteCount	#+20	(block data #2 – end of packet)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	0	
	End_packet	1	
	Last_data	1	

26.6.4. Example 4

Encrypt/Decrypt 3DES and Output MD5 Authentication with header and trailer

1st descriptor:	ObyteCount	0	(block data #1 – for header data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	1	
	End_packet	0	
	Last_data	0	
2nd descriptor:	ObyteCount	0	(block data #2 – for header data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	1	
3rd descriptor:	ObyteCount	#	(block data #3 – 3DES/MD5)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	0	
4th descriptor:	ObyteCount	#	(block data #4 – 3DES/MD5)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	1	
5th descriptor:	ObyteCount	0	(block data #5 – trailer data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	0	
6th descriptor:	ObyteCount	16	(block data #6 – trailer data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	1	
	Last_data	1	

26.6.5. Example 5

Encrypt/Decrypt 3DES and Output MD5 Authentication with header and trailer (different starting address for Authentication data)

1st descriptor:	ObyteCount	0	(block data #1 – for header data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	1	
	End_packet	0	
	Last_data	0	
2nd descriptor:	ObyteCount	0	(block data #2 – for header data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	1	
3rd descriptor:	ObyteCount	#	(block data #3 – 3DES/MD5)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	0	
4th descriptor:	ObyteCount	#	(block data #4 – 3DES/MD5)
	IbyteCount	#	
	IndexA	1-6	
	IndexB	1-6	
	Start_packet	0	
	End_packet	0	
	Last_data	1	
5th descriptor:	ObyteCount	0	(block data #5 – trailer data – MD5)
	IbyteCount	#	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	1	
	Last_data	1	
6th descriptor:	ObyteCount	16	(Authentication data)
	IbyteCount	0	
	IndexA	0	
	IndexB	1-6	
	Start_packet	0	
	End_packet	1	
	Last_data	1	

26.7. Random Number Generator (RNG) Engine

26.7.1. RNG Registers

Offset	Register	Width	R / W	Description
B0h	RCSR	[63:0]	R/W	RNG Control and Status Register
B8h	RPR	[31:0]	R/W	RNG Parameter Register
C0h	RDR	[63:0]	R/W	RNG Debug Register
C8h	ROR1	[63:0]	R/W	RNG Output Register 1
D0h	ROR2	[63:0]	R/W	RNG Output Register 2
D8h	ROR3	[63:0]	R/W	RNG Output Register 3

NOTE1: In an access to 64 bit register by 32bit instruction, address SWAP function works in write mode but it does not work in read mode.

NOTE2: Please do not write any data to upper Read Only 32bit field. Because lower 32 bit field would be written by mistake by writing data to upper 32 bit field.

26.7.1.1. RNG Control and Status Register (RCSR)

Table 26-20 RNG Control and Status Register (RCSR)

Bit(s)	Field	R/W	Default	Description
63:4	–	RO	0	Reserved
3	R_INTE	R/W	0	RNG Interrupt Enable 0: Disable 1: Enable This bit has to be program before start RNG controller To clear the interrupt, write “0” to R_INTE bit. Then write “1” back if there is another RNG operation.
2	R_RST	R/W	0	Reset Random Number Generator Controller 0: Normal 1: Reset After set this bit, need to reset it back to “0” for normal operation
1	R_FIN	RO	0	End Signal 0: Normal 1: Done After the number of clocks specified by the Rando Number Output Timing Specify signal (fintm) passes, this signal becomes active (“1”) and random numbers are output by “R_RN”
0	R_ST	R/W	0	RNG Start Bit 0: Idle 1: Start This bit will get reset when R_FIN is asserted.

26.7.1.2. RNG Parameter Register (RPR)

Table 26-21 RNG Parameter Register (RPR)

Bit(s)	Field	R/W	Default	Description
31:16	Lfsrltchtm	R/W	0x003F	Seed Latch Timing Specify signal
15:0	Fintm	R/W	0x005A	Random Number Output Timing Specify signal

26.7.1.3. RNG Debug Register (RDR)

Table 26-22 RDG Debug Register (RDR)

Bit(s)	Field	R/W	Default	Description
63:32	Rnstcntr	RO	?	0/1 Balance Value Output signal
31:16	Rdstore	RO	?	Simple Random Number Evaluation Measurement signal
15	Rrnerror	RO	1	0/1 Balance Error Detection signal
14	Rstrerr	RO	1	Randomizing Circuit Error Detection signal
13:9	-	RO	0	Reserved

8	Tstmode	R/W	0	Test Mode Set signal 0: Normal 1: Test mode For debug purpose only
7	Slfxdyn	R/W	0	Randomizing Circuit Input Switching signal
6	Errclr	R/W	0	0/1 Balanced Error Detection Clear signal
5	Ysdmclr	R/W	0	Simple Random Number Evaluation Measurement Clear signal
4	Sterrclr	R/W	0	Randomizing Circuit Error Clear signal
3:0	Thhlderr	R/W	0	If the fault threshold input signal “seed” circuit output 0/1 balance was judged to have been disrupted, the recommended value of the signal wire that inputs the threshold is the decimal value 13 (0xD hexadecimal)

26.7.1.4. RNG Output Register 1 (ROR1)

Table 26-23 RNG Output Register 1 (ROR1)

Bit(s)	Field	R/W	Default	Description
63:0	RN1[63:0]	RO	?	Random Number Output signal RN1[63:0] = rn[63:0]

26.7.1.5. RNG Output Register 2 (ROR2)

Table 26-24 RNG Output Register 2 (ROR2)

Bit(s)	Field	R/W	Default	Description
63:0	RN2[63:0]	RO	?	Random Number Output signal RN2[63:0] = rn[127:64]

26.7.1.6. RNG Output Register 3 (ROR3)

Table 26-25 RNG Output Register 3 (ROR3)

Bit(s)	Field	R/W	Default	Description
63:0	RN3[63:0]	RO	?	Random Number Output signal RN3[63:0] = rn[191:128]

Chapter 27. EJTAG Interface

27.1. Extended EJTAG Interface

The TX4939 Extended EJTAG (Enhanced Joint Test Action Group) Interface provides two real-time debugging functions. One is the IEEE1149.1 standard compliant JTAG Boundary Scan Test, and the other is the Debugging Support Unit (DSU) that is built into the TX49/H4 core.

- *JTAG Boundary Scan Test*
  - *IEEE1149.1 compatible TAP Controller*
  - *Supports the following five instructions: EXTEST, SAMPLE/PRELOAD, IDCODE, BYPASS, HIGHZ*

- *Real-time Debugging*
  - *Real-time debugging using an emulation probe (made by Corelis or YDC)*
  - *Execution control (run, break, step, register/memory access)*
  - *Real-time PC tracing*

Please contact your local Toshiba Sales representative for more information regarding how to connect the emulation probe.

The two functions of the Extended EJTAG Interface operate in one of two modes.

- *PC Trace Mode*
  - *Execution control (fun, pause, access single steps, access internal register/system memory)*
  - *JTAG Boundary Scan Test*

- *Real-time Mode*
  - *Real-time PC tracing*

Table 27-1 EJTAG Interface Function and Operation Code

PC Tracing Mode	Off	On
JTAG Boundary Scan	Boundary Scan Test	—
Real-time Debugging	Execution Control	Real-time PC Tracing

27.2. JTAG Boundary Scan Test

27.2.1. JTAG Controller and Register

The Extended EJTAG Interface contains a JTAG Controller (TAP Controller) and a Control Register. This section explains only those portions that are unique to the TX4939. Please refer to the TX49/H4 Core Architecture Manual for all other portion not covered here. Please contact your local Toshiba Sales representative for more information regarding the required BSD files when performing the JTAG Boundary Scan Test.

- *Instruction Register (Refer to 27.2.2)*
- *Data Register*
  - *Boundary Scan Register (Refer to 27.2.3)*
  - *Bypass Register*
  - *Device ID Register (Refer to 27.2.4)*
  - *JTAG Address Register*
  - *JTAG Data Register*
  - *JTAG Control Register*
  - *EJTAG Mount Register*
- *Test Access Port Controller (TAP Controller)*

27.2.2. Instruction Register

The JTAG Instruction Register consists of an 8-bit shift register. This register is used for selecting either one or both of the test to be performed and the Test Data Register to be accessed. The Data Register is selected according to the instruction code in Table 27-2. Refer to the TX49/H4 Core Architecture Manual for more information regarding each instruction.

Table 27-2 Bit Configuration of JTAG Instruction Register

Instruction Code MSB → LSB	Instruction	Selected Data Register
00000000 (0x00)	EXTEST	Boundary Scan Register
00000001 (0x01)	SAMPLE/PRELOAD	Boundary Scan Register
00000010 (0x02)	Reserved	Reserved
00000011 (0x03)	IDCODE	Device ID Register
00000100 - 00001111	Reserved	Reserved
00010000 (0x10)	HIGHZ	Bypass Register
00010001 - 01111111	Reserved	Reserved
10000000 - 11111110	Refer to the TX49/H4 Core Architecture Manual	
11111111 (0xFF)	BYPASS	Bypass Register

Figure 27-1 shows the format of the Instruction Register.

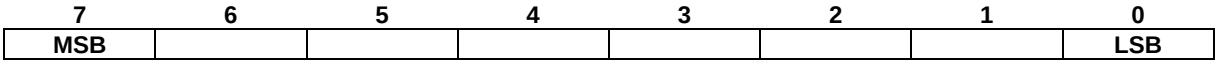


Figure 27-1 Instruction Register

The instruction code is shifted to the Instruction Register starting from the Least Significant Bit.

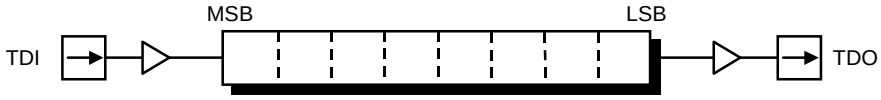


Figure 27-2 Shift Direction of the Instruction Register

27.2.3. Boundary Scan Register

The Boundary Scan Register contains a single 256-bit shift register to which all TX4939 I/O signals except for power supply, TDI, TCK, TDO, TMS, TRST*, and TEST[4]* are connected. Figure 27-3 shows the bits of the Boundary Scan Register.

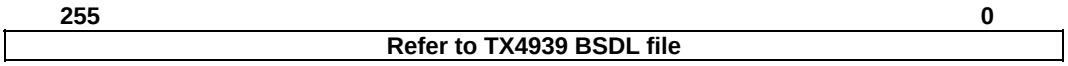


Figure 27-3 Boundary Scan Register

TDI input is fetched to the Least Significant Bit (LSB) of the Boundary Scan Register and the Most Significant Bit (MSB) of the Boundary Scan Register is sent from the TDO output.

27.2.4. Device ID Register

The Device ID Register is a 32-bit shift register. This register is used for reading the ID code that expresses the IC manufacturer, part number, and version from the IC and sending it to a serial device. The following figure shows the configuration of the Device ID Register.

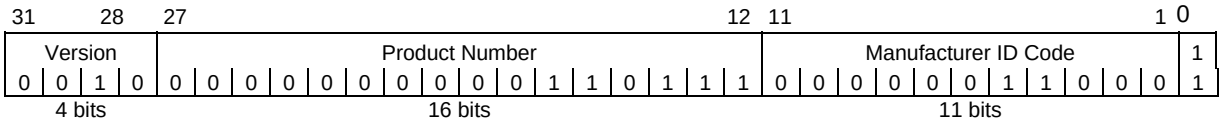


Figure 27-4 Device ID Register

The device ID code for the TX4939 is 0x2003_7031. However, the four top bits of the Version field may be changed. The device ID code is shifted out from the Least Significant Bit.

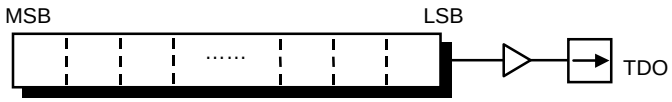


Figure 27-5 Shift Direction of the Device ID Register

27.2.5. Initializing the Extended EJTAG Interface

The Extended EJTAG Interface is not reset by asserting the RESET* signal. Operation of the TX49/H4 core is not guaranteed if the Extended EJTAG Interface is not reset. This interface is initialized by either of the following methods.

- *Assert the TRST* signal.*
  After clearing the processor reset, set the TMS input to High for five consecutive rising edges of the TCK input. The reset state is maintained if TMS is able to maintain the High state.

  The above methods must be performed while the MASTERCLK signal is being input. Also, externally fix the TRST signal to GND when not using an emulation probe. The G-Bus Time Out Detection function is disabled when the TRST* signal is deasserted.*

27.2.6. Features

1. Utilizes JTAG interface compatible with IEEE Std. 1149.1.

2. Processor access to external processor probe to execute from the external trace memory during debug exception and boot time. This is to eliminate system memory for debugging purpose.

3. Supports DMA access through JTAG interface to internal processor bus to access internal registers, host system peripherals and system memory.

4. Debug functions
   Instruction Address Break
   Data Bus break
   Processor Bus Break
   Reset, NMI, Interrupt Mask

5. Instructions for Debug
   SDBBP, DERET, CTC0, CFC0

6. CP0 Registers for Debug
   Debug, DEPC, DESAVE

27.2.7. EJTAG interface

This interface consists of two modes of operation a Run Time Mode and Real Time Mode. The Run Time Mode provides functions such as processor Run, STOP, Single Step, and access to internal registers and system memory. The Real Time mode provides additional status pins used in conjunction with JTAG pins for Real Time Trace information.

Table 27-3 JTAG Interface

PIN NAME	I / O	FUNCTION
JTDI	I	**JTAG data input / Debug interrupt input** Run-time mode: Input serial data to JTAG data/instruction registers. Real-time mode: Interrupt input to change the debug unit state from real-time mode to run-time mode.
JTCK	I	**JTAG clock input** Clock input for JTAG. The JTDI and JTMS data are latched on rising edges of this clock.
JTDO/TPC(0)	O	**JTAG data output / Trace PC output** Data is serially shifted out from this pin. / Outputs a non-sequential program counter value synchronously with DCLK.
JTMS	I	**JTAG command** Controls mainly the status transition of the TAP controller state machine. When the serial input data is a JTAG command, apply a high signal (= 1) to this pin.
DCLK	O	**Debug clock (1/3 CPU clock)** Clock output for a real-time debug system. Timings of the serial monitor bus and PC trace interface signals all are defined by this debug clock DCLK. This Debug clock frequency is 1/3 that of CPUCLK.
PCST(8:0)	O	**PC trace status** Outputs PC trace status information and serial monitor bus mode.
TPC(3:1) Note 1	O	**Trace PC output** Outputs a non-sequential program counter value synchronously with DCLK.
TRST*	I	**Test reset input** Reset input for a real-time debug system. When TRST* is asserted (= 0), the debug support unit (DSU) is initialized.

Note1: Leave TPC (3-1) pins open when not using them as PC trace outputs for debugging.

27.2.8. JTAG Interface

Standard JTAG interface is used for on chip debugging during Run Time mode. The TX49 Debug Support Unit has following registers.

- *Instruction Register*
- *Bypass Register*
- *Boundary-Scan Register*
- *Device Identification Register*
- *Implementation Register*
- *JTAG_Data_Register*
- *JTAG_Address_Register*
- *JTAG_Control_Register*

27.2.9. Processor Access Overview

The core processor can access external processor probe for reading and writing to external monitor memory, registers and other external resources.

In addition the processor can execute from the external monitor memory located from 0xf_ff20 0000 to 0xf_ff2f ffff when the ProbEnb bit is set and the processor probe is turned ON. Any access to the monitor location from 0xf_ff20 0000 to 0xf_ff3f ffff are only allowed when the processor is in the debug mode (DM = 1).

27.2.10. Instruction

The instruction is a 8 bit field. Instructions for the TX49 Debug Support Unit are encoded between 0x80 and 0x9f and other codes are reserved for Toshiba Standard JTAG instructions (Includes EXTEST, SAMPLE/PRELOAD and IDCODE) and so on. Instructions are decoded as follows.

Table 27-4 Instruction

Hex Value	Instruction	Description
0x00	EXTEST	Boundary Scan Register
0x01	SAMPLE/PRELOAD	Boundary Scan Register
0x02	Reserve	Reserve
0x03	IDCODE	Device Identification Register
0x83	EJTAG_ImpCode	Select Implementation Register
0x88	JTAG_ADDRESS_IR	Select JTAG_Address Register
0x89	JTAG_DATA_IR	Select JTAG_Data Register
0x8A	JTAG_CONTROL_IR	Select JTAG_Control Register
0x8B	JTAG_ALL_IR	Select JTAG_All Register
0x90	PCTRACE	PCIRACE Instruction
0xFF	BYPASS	Select BYPASS Register

Note: 0x80 ~ 0x9F are other code. Please do not use.

Any unused instruction between 0x80 and 0x9f defaulted to BYPASS instruction.

27.2.11. Debug Unit

- *Extended Instructions*
  - *SDBBP*
  - *DERET*
  - *CTC0*
  - *CFC0*
- *Extended Debug Registers in CP0*
  - *Debug Register*
  - *Debug Exception PC (DEPC)*
  - *Debug SAVE*

27.2.12. Register Map

Table 27-5 Register Map

Address	Mnemonic	Description
0xf ff30 0000	DCR	Debug Control Register
0xf ff30 0008	IBS	Instruction Break Status
0xf ff30 0010	DBS	Data Break Status
0xf ff30 0018	PBS	Processor Break Status
0xf ff30 0100	IBA0	Instruction Break Address 0
0xf ff30 0108	IBC0	Instruction Break Control 0
0xf ff30 0110	IBM0	Instruction Break Address Mask 0
0xf ff30 0300	DBA0	Data Break Address 0
0xf ff30 0308	DBC0	Data Break Control 0
0xf ff30 0310	DBM0	Data Break Address Mask 0
0xf ff30 0318	DB0	Data Break Value 0
0xf ff30 0600	PBA0	Processor Bus Break Address 0
0xf ff30 0608	PBD0	Processor Bus Break Data 0
0xf ff30 0610	PBM0	Processor Bus Break Mask 0
0xf ff30 0618	PBC0	Processor Bus Break Control 0

27.3. Processor Bus Break Function

This function is to monitor the interface to core and provide debug interruption or trace trigger for a given physical address and data.

27.4. Debug Exception

Three kinds of debug exception are supported.

- Debug Single Step (DSS bit)
- Debug Breakpoint Exception (SDBBP Instruction)
- JTAG Break Exception (Jtagbrk bit in JTAG_Control_Register)

Note: During real time debugging, first two functions are disabled.

27.4.1. Debug Single Step (DSS)

When the debug register DSS bit is set, this exception has been raised each time one instruction is executed.

27.4.2. Debug Breakpoint exception (Dbp)

This exception is raised when SDBBP instruction is executed.

27.4.3. JTAG Break Exception

This exception is raised when JTAG unit set the Jtagbrk in JTAG_Control_Register.

27.4.4. Debug Exception Handling

Updates DEPC and Debug register.
Registers other than DEPC and Debug register retain their values.

27.4.5. Branching to debug handler

If the ProbEnb bit in JTAG_Control_Register[15] is set, the debug exception vector is located at
PC: 0xffff ffff ff20 0200

If the ProbEnb bit in JTAG_Conctrol_Register[15] is cleared, the debug exception vector is located at
PC: 0xffff ffff bfc0 0400

27.4.6. Exception handling when in Debug Mode (DM bit is set)

All interrupts including NMI are masked. When the NMI interrupt has occurred during Debug mode, it is stored internally and the NMI interrupt is taken after debug handler is finished (DM is clear)

27.5. Real Time PC TRACE Output

In real time mode non-sequential Program Counter and trace information are outputted on TPC[3:0] and PCST[8:0] at 1/3 of the processor clock speed.

Chapter 28. Electrical Characteristics

28.1. Absolute Maximum Rating

Note 1

Item	Symbol	Rating	Unit
Supply Voltage 1 (for 3.3V I/O Pins)	$VDD33_{Max}$	–0.3 - 3.9	V
Supply Voltage 2 (for 2.5V I/O Pins)	$VDD25_{Max}$	–0.3 - 3.2	V
Supply Voltage 3 (for Internal Circuit)	$VDDC_{Max}$	–0.3 - 2.0	V
Input Voltage 1 (Note 2)	V_{IN1}	–0.3 - VDD33 + 0.3 V	V
Input Voltage 2 (Note 3)	V_{IN2}	–0.3 - VDD25 + 0.3 V	
Storage Temperature	T_{STG}	-40 - 125	°C
Maximum Power Dissipation	P_D	T.B.D.	W

Note 1: The Absolute Maximum Rating is a rating that must never be exceeded, even for an instant. Not a single Absolute Maximum Rating value can be exceeded. If any Absolute Maximum Rating value is exceeded, the product may be damaged or weakened, or damage or combustion may cause personal injury. Always be sure to design your application devices so the Absolute Maximum Rating is never exceeded.

Note 2: Even be sure that VDD33 + 0.3 V does not exceed the maximum rating for $VDD33_{Max}$.

Note 3: Even be sure that VDD25 + 0.3 V does not exceed the maximum rating for $VDD25_{Max}$.

28.2. Recommended Operating Conditions

Note 1

Item		Symbol	Condition	Min.	Typ.	Max.	Unit
Supply Voltage	I/O	VDD33		3.1	3.3	3.5	V
	I/O (DDR333 below)	VDD25		2.3	2.5	2.7	V
	I/O (DDR400)	VDD25		2.5	2.6	2.7	V
	Internal Circuit	VDDC		1.188	1.25	1.312	V
Operating Temperature 1 (Package Temperature)		T_C		0	-	85	°C

Note 1: The Recommended Operating Conditions are usage conditions that are recommended so that a product can operate normally and maintain constant product quality. If a product is used with operating conditions that exceed even one of the Recommended Operating Conditions, the product may malfunction. Always be sure to design your application devices so they will operate within the range of the Recommended Operating Conditions.

28.3. DC Characteristics

28.3.1. DC Characteristics of Pins (Except PCI I/F)

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V or VDD25 = 2.6 V ± 0.1 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Low-level Input Voltage	V_{IL1}	(Note 1)	-0.3		0.8	V
High-level Input Voltage	V_{IH1}	(Note 1)	2.0		VDD33+0.3	V
Low-level Output Current	I_{OL1}	(Note 2) V_{OL} =0.4V	4			mA
High-level Output Current	I_{OH1}	(Note 2) V_{OH} =2.4V	-		-4	mA
Reference Voltage	Vref	DRVref1, VDD25 = 2.5V or 2.6V	0.49*VDD25		0.51*VDD25	V
Low-level Input Voltage	V_{IL2}	(Note 3) VDD25 = 2.5V or 2.6V	-0.3		Vref-0.15	V
High-level Input Voltage	V_{IH2}	(Note 3) VDD25 = 2.5V or 2.6V	Vref+0.15		VDD25+0.3	V
Low-level Output Current	I_{OL2}	(Note 4) V_{OL} =0.4V	Note 10			mA
High-level Output Current	I_{OH2}	(Note 4) V_{OH} =1.8V	-		Note 10	mA
Low-level Input Leak Current	I_{IL1}	(Note 5) V_{IN} =VSS	-10		10	µA
Low-level Input Leak Current	I_{IL2}	(Note 6) V_{IN} =VSS	-200		-10	µA
High-level Input Leak Current	I_{IH1}	(Note 7) V_{IN} =VDD33	-10		10	µA
High-level Input Leak Current	I_{IH2}	(Note 8) V_{IN} =VDD33	10		200	µA
Hi-z Output Leak Current	I_{OZ}	(Note 9)	-10		10	µA
Operating Power Supply Current (Internal Power)	I_{CCInt}	VDD33 = 3.3 V, VDD25 = 2.6 V, VDDC = 1.3 V, CPU Clock = 400 MHz, DDR Clock = 200 MHz, Pin Load Capacitance = 25 pF	T.B.D.	T.B.D.	T.B.D.	mA
Operating Power Supply Current (VDD33)	I_{CCIO1}		T.B.D.	T.B.D.	T.B.D.	mA
Operating Power Supply Current (VDD25)	I_{CCIO2}		T.B.D.	T.B.D.	T.B.D.	mA

Note 1: All input pins, all bi-directional pins (during input) except for PCI/DDR Interface signals.

Note 2: ACE*, SA[5:0], BE[1:0]*, CE[7:0]*, DMAACK[2:0], DMADONE*, HALTDOZE, SWE*, SYSCLK, TIMER[1:0], TXD, SADB[15:0], DCLK, PCST[8:0], TDO, TPC[3:1]

Note 3: DDR input and bi-directional signals. DRDQ[31:0], DRDQS[3:0], DRCKREF

Note 4: DDR output and bi-directional signals. DRA[13:0], DRDQ[31:0], DRDQS[3:0], DRDM[3:0], DRCS[1:0]*, DRCKP, DRCKM, DRCKOUT, DRBA[1:0], DRWE*, DRCAS*, DRRAS*, DRCKE

Note 5: All input without Pull-Up device include bi-directional during input, such as MSTCLK, MSTCLK2, RESET*, DRDQ[31:0], DRDQS[3:0], DRCKREF, All ATA100 signals, NDRB*, WAIT, GPIO[3:0], All PCI Signals, BITCLK, All ACLINK or I2S signals, I2C_D0, TRST*, EJRST*, DINT*

Note 6: All input with Pull-Up device include bi-directional during input, such as All Video Port Signals, SADB[15:0], SA[5:0], ACK*, DMAREQ[2:0], DMAACK[2:0], DMADONE*, NDLA, All SPI signal, RXD[3:0], NMI*, INT[2:0]*, INT[D:A]*, TEST[0]*, BYPASSPLL, div_RST*, TCK, TMS, TDI, DINT*

Note 7: All input without Pull-Down device include bi-directional during input, such as, MSTCLK, MSTCLK2, HALTDOZE, DRDQ[31:0], DRDQS[3:0], DRCKREF, All Video Port signals, All ATA100 signals, All input of EBC signals, NDRB*, WAIT, GPIO[3:0], All PCI signals, BITCLK, I2S_WS0, I2S_SCK0, I2S_WS1, I2S_SCK1, I2S_WS, I2S_SCK, I2S_SD1, I2C_DO, All SPI signals, RXD[3:0], NMI*, INT[2:0]*, INT[D:A]*, TEST[0]*, BYPASSPLL*, div_RST*, TCK, TMS, TDI, DINT*,

Note 8: All input with Pull-Down device include bi-directional during input, SDIN[1:0], I2S_SD0(I2S 2ch), I2S_SD1(I2S 2ch), I2S_SD0(I2S 5.1ch), I2S_SD2(I2S 5.1ch), TRST*, EJRST*

Note 9: TXD[3:0], All ATA100 signals, ALARM*

Note 10: Those values are programmable.

28.3.2. DC Characteristics of Pins (PCI I/F)

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Low-level Input Voltage	V_{ILPCI}	(Note 1)	-0.5		0.9	V
High-level Input Voltage	V_{IHPCI}	(Note 1)	1.8		VDD33+0.3	V
High-level Output Voltage	V_{OHPCI}	(Note 2) I_{OUT} = -500µA	VDD33 x 0.9		-	V
Low-level Output Voltage	V_{OLPCI}	(Note 2) I_{OUT} = 1500µA			VDD33 x 0.1	V
Input Leak Current	I_{IHPCI} I_{ILPCI}	$0 < V_{IN} <$ VDD33	-10		10	µA

Note 1: ID_SEL, PCIICLKIN, C_BE[3:0]*, DEVSEL*, FRAME*, GNT[5:0]*, IRDY*, LOCK*, M66EN, PAR, PCIAD[31:0], PERR*, PME*, REQ[5:0], SERR*, STOP*, TRDY*

Note 2: All PCI interface signals except ID_SEL, LOCK* and PCICLKIN

28.4. AC Characteristics

28.4.1. MSTCLK, MSTCLK2 AC Characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Typ	Max	Unit
Master Clock Input Cycle	f_{MCP}			50		nS
Master Clock Input Frequency	f_{MCK}			20		MHz
Master Clock High Time	t_{MCH}		23		27	nS
Master Clock Low Time	t_{MCL}		23		27	nS
CPUCLK Frequency [2]	$_{CPU}$		200	300	400	MHz
Master Clock Rise Time	t_{MCR}		1		2	nS
Master Clock Fall Time	t_{MCF}		1		2	nS

Note 1: TX4939 operation is only guaranteed when the power is stable, PLL secures the PLL oscillation stability time tMCP_PLL and is in the Enable state.

Note 2:

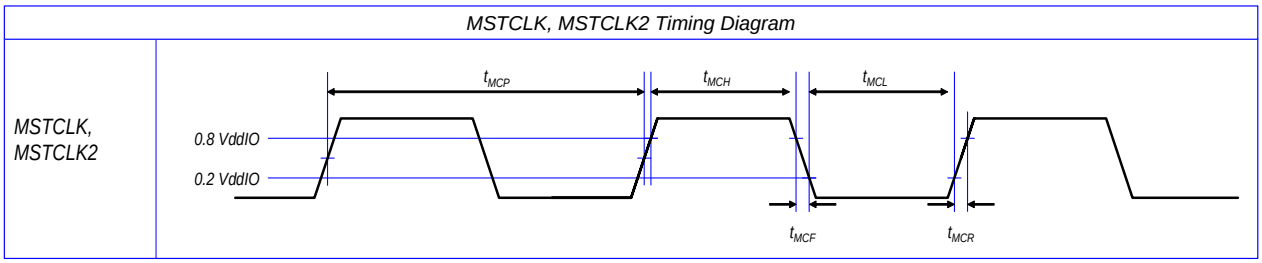


Figure 28-1 MSTCLK Timing Diagram

28.4.2. Power ON AC Characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
RESET* Width Time	T_{RESETW}		10		mS

RESET* Width Time consists of all PLL stabilized Time and logic reset operation time.

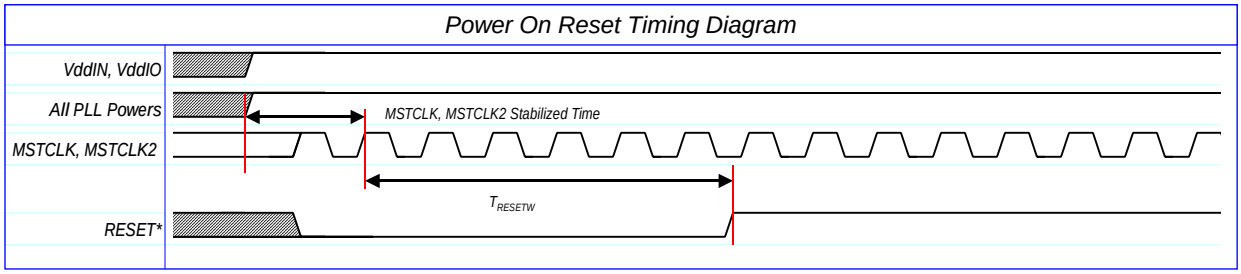


Figure 28-2 Power On Reset Timing Diagram

28.4.3. DDR SDRAM Interface AC Characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V or 2.6 ± 0.1 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Symbol	Parameters	Min.	Max.	Unit	Note
tCK	DRCKOUT cycle time	5.000	10.000	ns	
tCH	DRCKOUT high pulse width	0.45*tCK	0.55*tCK	ns	
tCL	DRCKOUT low pulse width	0.45*tCK	0.55*tCK	ns	
tCKS	DRCKOUT output skew	-0.350	0.350	ns	1,2,3
tDQSH	DRDQS high pulse width	0.45*tCK	0.55*tCK	ns	1
tDQSL	DRDQS low pulse width	0.45*tCK	0.55*tCK	ns	1
tDQS	DRCKOUT to DRDQS	-0.350	0.350	ns	1,4,5
tDSS	DRDQS falling edge to DRCKOUT rising edge	0.45*tCK -0.35	-	ns	1,6,7
tDSH	DRCKOUT rising edge to DRDQS falling edge	0.45*tCK -0.35	-	ns	1,6,7
tDQSQV	DRDQS to DRDQ shift (when data becomes valid)	-{0.5* tDQSHmax - 0.638}	-{0.5* tDQSHmin + 0.638}	ns	1,8,9
tDQSQIV	DRDQS to DRDQ shift (when data becomes invalid)	{0.5* tDQSHmin - 0.638}	{0.5* tDQSHmax + 0.638}	ns	1,8,9
t2	ADDR/CMD† to DRCKOUT	-0.350	0.350	ns	1,10,11
t3	DRDQ edge arrival relative to DRDQS	-{tCK/4 - [0.350+0.2* (tCK/4)]}	{tCK/4 - [0.350+0.2* (tCK/4)]}	ns	12,13,14

† ADDR/CMD = DRA[], DRCS[1:0]*, DRCKP, DRCKM, DRWE*, DRCAS*, DRRAS*, DRCKE

Note:

1: Write cycle timing parameter
2: The skew consists of pad output skew (+ 250ps) and package routing skew between any two clock pairs (+ 100ps).
3: tCKS Timing Parameter, refer to Figure 28-4
4: The timing consists of pad output skew (+ 250ps) and package routing skew between any DRCKOUT to any DRDQS (+ 100ps).
5: tDQS Timing parameter, refer Figure 28-5.
6: The skew consists of pad output skew (+ 250ps) and package routing skew between any DRCKOUT to any DRDQS (+ 100ps). Minimum DRDQS pulse width is 45% of DRCKOUT.
7: tDSS, tDSH timing parameters, refer to Figure 28-6.
8: During write, DRDQ signals are driven quarter clock earlier such that DRDQS is placed in the center of data eye window. The skew consists of pad output skew (+ 250ps), package routing skew between any DRDQS signals and it's associated DRDQ signals (+ 75ps) and maximum clock granularity (+ 312.5 ps).
9: tDQSQV and tDQSQIV timing parameters apply only within DRDQS and its associated DRDQ signals. Refer to Figure 28-7.
10: The skew consists of pad output skew (+ 250 ps) and package routing skew (+ 100 ps) between any DRCKOUT pair to any ADDR/CMD signal. Maximum clock granularity skew is 312.5 ps.
11: t2 Timing parameter, applies to registered DIMM Environment Only - ADDR/CMD signals are launched 1/2 clock cycle early. The granularity term does not apply here. Refer to Figure 28-8.
12: Read cycle timing parameter.
13: The PDL placement uncertainty is 20%. Package skew between DRDQS and its associated DRDQs is 75ps. The sum of setup/hold time & receiver uncertainty is 275ps.
14: t3 timing parameter, refer to Figure 28-9.

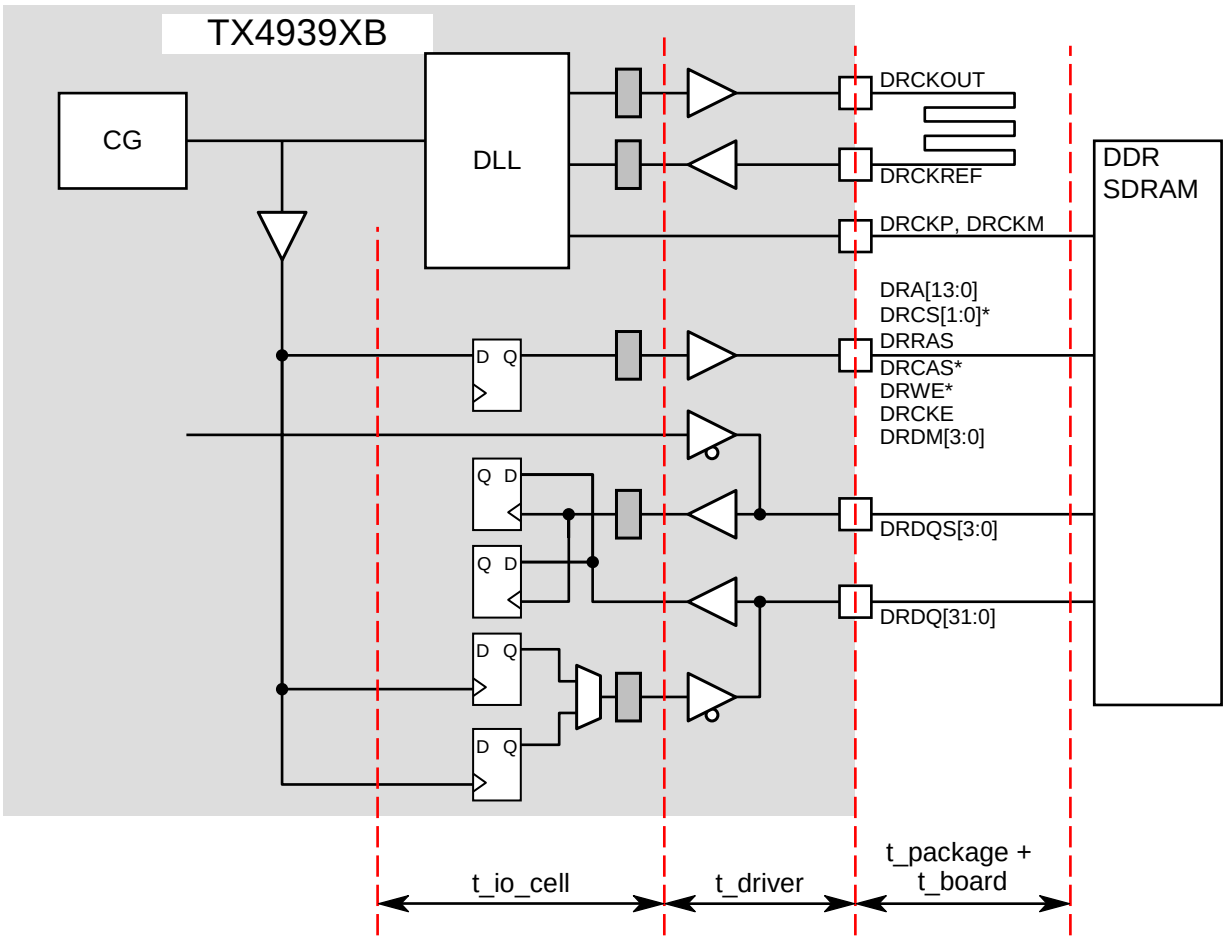


Figure 28-3 Block Diagram of DDR SDRAM Controller

DRCKOUT Output Skew

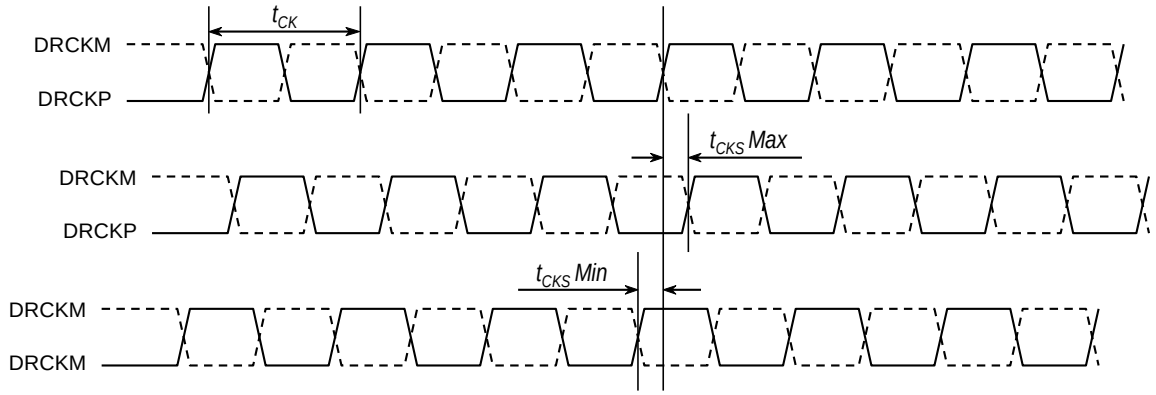


Figure 28-4 DRCKOUT Output Skew

DRDQS Timing Parameter

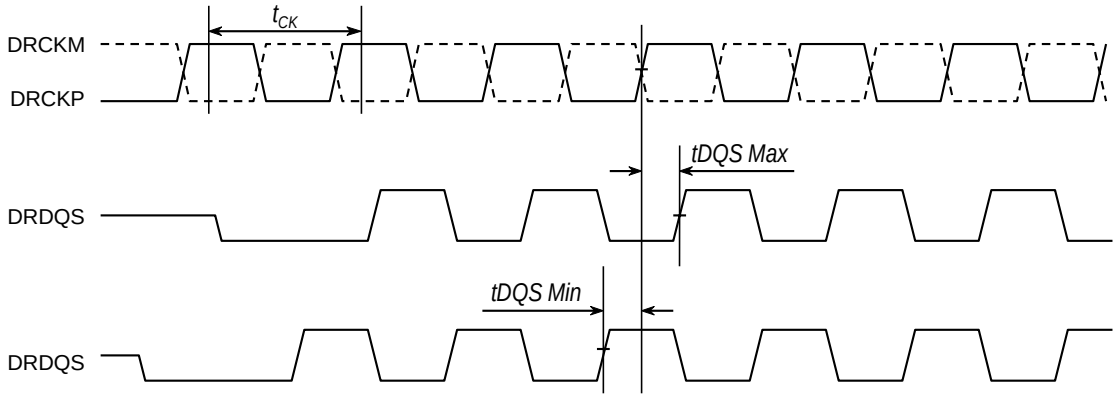


Figure 28-5 DRDQS Timing Parameter

tDSS/tDSH Timing Parameter

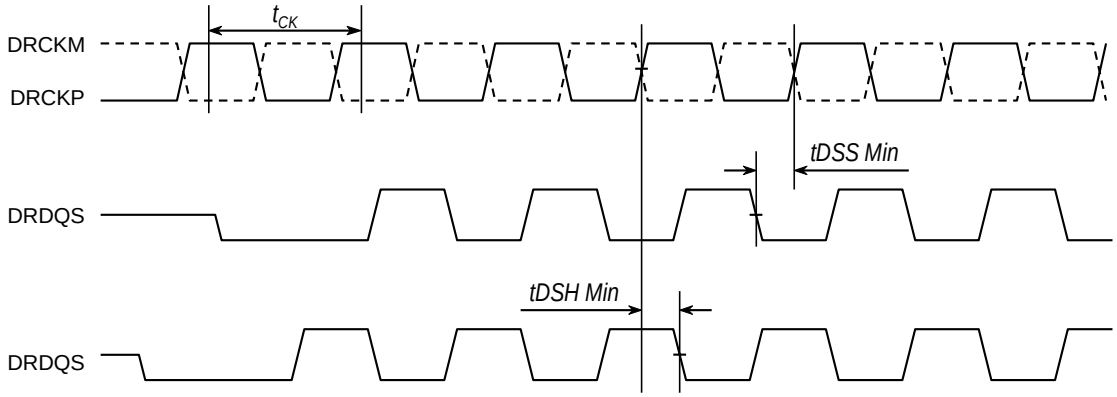


Figure 28-6 tDSS/tDSH Timing Parameters

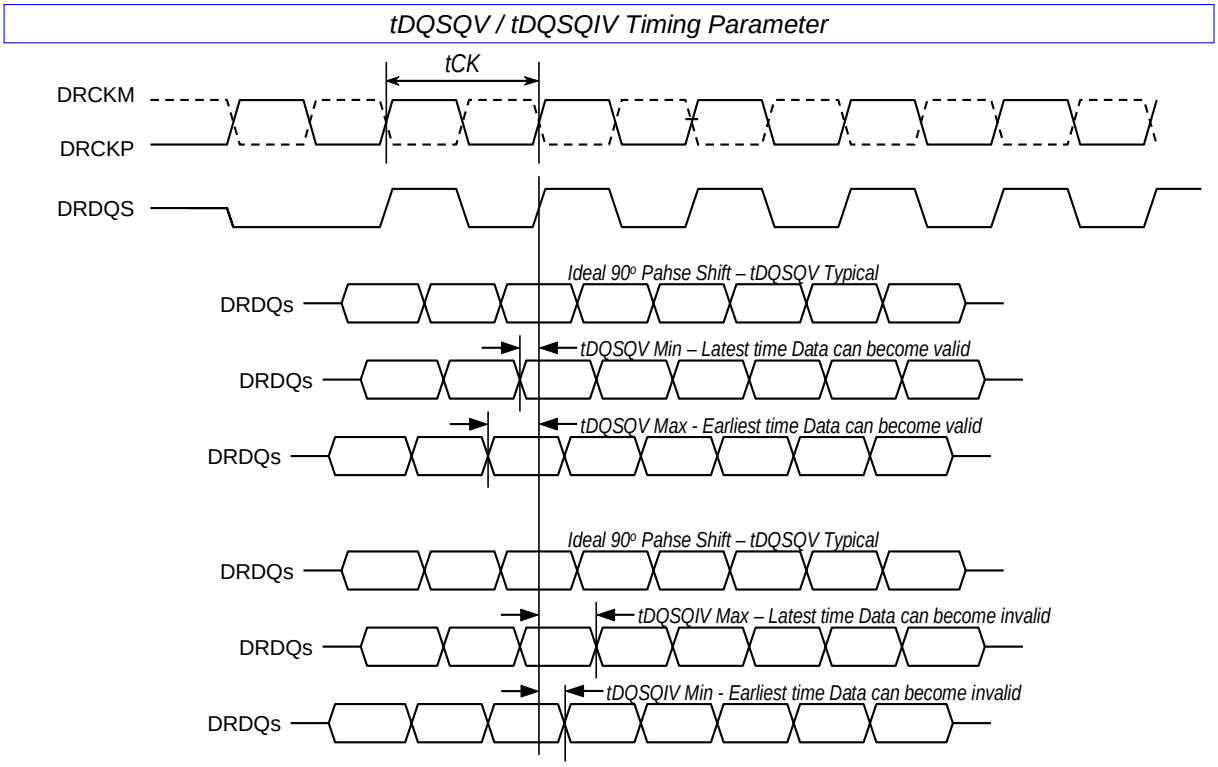


Figure 28-7 tDQSQV/tDQSQIV Timing Parameters

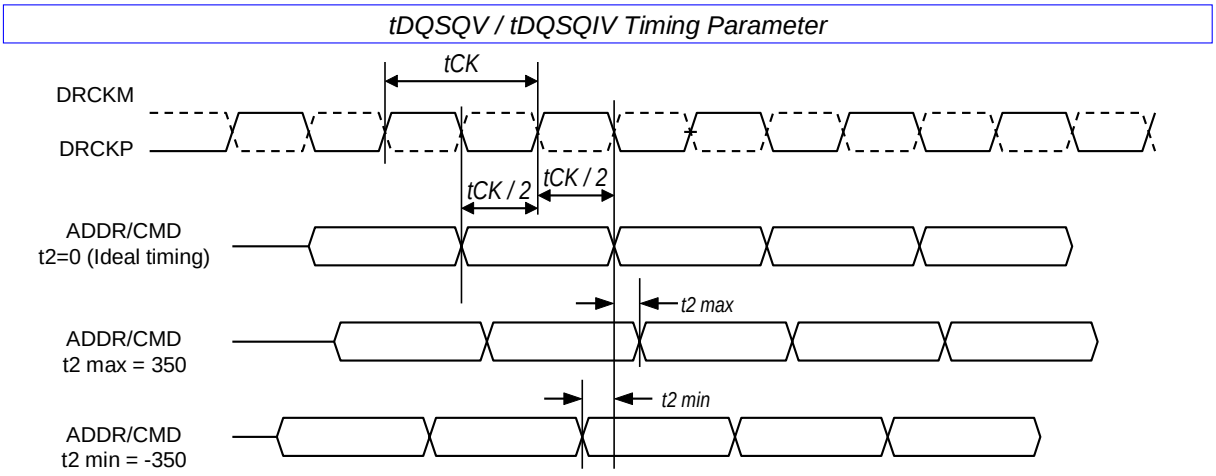


Figure 28-8 ADDR/CMD to DRCKOUT Timing Parameter (Registered DIMMs)

DRDQS Edge Arrival Relative to DRDQ

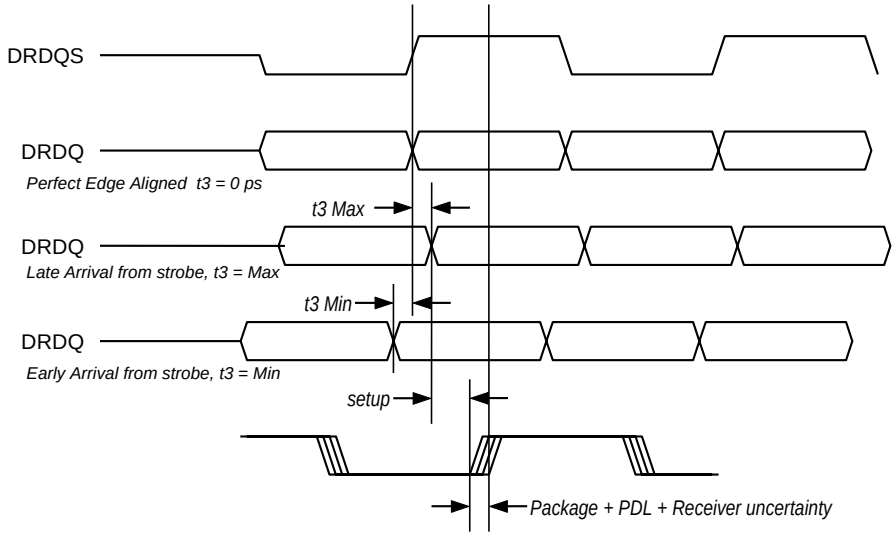


Figure 28-9 DRDQS Edge Arrival Relative to DRDQ

28.4.4. External Bus Interface AC Characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
SYSCLK Cycle Time	t_{CYC_SYSCLK}	Buffer fixed: CL=50 pF, 4 mA	10	—	ns
SYSCLK High Time	t_{HIGH_SYSCLK}	Buffer fixed: CL=50 pF, 4 mA	4	—	ns
SYSCLK Low Time	t_{LOW_SYSCLK}	Buffer fixed: CL=50 pF, 4 mA	4	—	ns
SA[5:0] Output Delay	t_{VAL_ADDR2}	For CL=50 pF, 4 mA buffer	1.5	8.5	ns
CE[7:0]* Output Delay	t_{VAL_CE}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
OE* Output Delay	t_{VAL_OE}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
SWE* Output Delay	t_{VAL_SWE}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
BWE[1:0]* Output Delay	t_{VAL_BWE}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
ACE Output Delay	t_{VAL_ACE}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
SADB[15:0] Output Delay (H→L, L→H)	t_{VAL_BUS}	For CL=50 pF, 4 mA buffer	1.5	8.5 (1)	ns
SADB[15:0] Output Delay (Hi-Z→Valid)	$t_{VAL\_DATA2ZV}$	For CL=50 pF, 4 mA buffer	1.5	8.5	ns
DATA[15:0] Ouput Delay (Valid→Hi-Z)	$t_{VAL\_DATA2VZ}$	For CL=50 pF, 4 mA buffer	-1	8.5	ns
SADB[15:0] Input Setup Time	t_{SU_DATA2}		6.0	—	ns
SADB[15:0] Input Hold Time	t_{HO_DATA2}		0.5	—	ns
ACK* Output Delay (H→L, L→H)	t_{VAL_ACK}	Buffer fixed: CL=50 pF, 4 mA	1.5	8.5	ns
ACK* Input Setup Time	t_{SU_ACK}		6.0	—	ns
ACK* Input Hold Time	t_{HO_ACK}		0.5	—	ns

Note 1: When the speed of the Local Bus is set to 1/3,1/4,1/5, 1/6 speed, write data delay has 10 ns more for synchronization

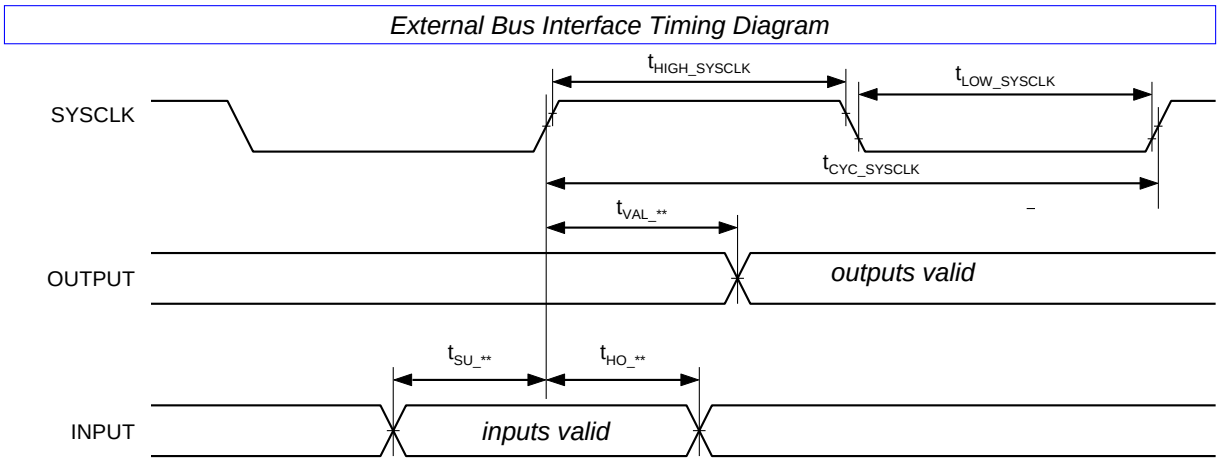


Figure 28-10 External Bus Interface Timing Diagram

28.4.5. PCI Interface AC Characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Signal Name	I/O	PCI Bus Spec.	Load (pF)	Symbol	Explanation	Min. (ns)	Max. (ns)
PCICLKIN	I	66 MHz	-	T_{CYC66}	Input Clock Cycle Time	15	30
				T_{HIGH66}	Input Clock High Time	6	-
				T_{LOW66}	Input Clock Low Time	6	-
				T_{SLEW66}	Input Clock Slew Rate [V/ns]	1.5	4
		33 MHz	-	T_{CYC33}	Input Clock Cycle Time	30	40
				T_{HIGH33}	Input Clock High Time	11	-
				T_{LOW33}	Input Clock Low Time	11	-
				T_{SLEW33}	Input Clock Slew Rate [V/ns]	1	4
PCICLK[6:0]	O	66 MHz	50	T_{CYCO66}	Output Clock Cycle Time	15	30
				$T_{HIGHO66}$	Output Clock High Time	6	-
				T_{LOWO66}	Output Clock Low Time	6	-
		33 MHz	70	T_{CYCO33}	Output Clock Cycle Time	30	40
				$T_{HIGHO33}$	Output Clock High Time	11	-
				T_{LOWO33}	Output Clock Low Time	11	-
		-	50	T_{SKW}	Output Clock Skew Point-To-Point Connection	0	TBD
PCIAD[31:0] C_BE[3:0] PAR FRAME* IRDY* TRDY* STOP* DEVSEL* PERR* SERR* LOCK* ID_SEL PME*	I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I/O I O3	66 MHz	30	T_{VAL66}	Output Delay (Bus Connection)	1	7
			-	T_{SU66}	Setup Time (Bus Connection)	3.5	-
			-	T_{H66}	Hold Time (Bus Connection)	0	-
		33 MHz	70	T_{VAL33}	Output Delay (Bus Connection)	1	11
			-	T_{SU33}	Setup Time (Bus Connection)	7	-
			-	T_{H33}	Hold Time (Bus Connection)	0	-
REQ[5:1]* GNT[5:1]*	I/O I/O	66 MHz	30	T_{PPD66}	Output Delay (Point-To-Point Connection)	1	7
			-	T_{PPS66}	Setup Time Point-To-Point Connection	5	-
			-	T_{PPH66}	Hold Time Point-To-Point Connection	0	-
		33 MHz	70	T_{PPD33}	Output Delay Point-To-Point Connection	1	12
			-	T_{PPS33}	Setup Time Point-To-Point Connection	10	-
			-	T_{PPH33}	Hold Time Point-To-Point Connection	0	-

Note 1: M66EN is treated as a static signal.

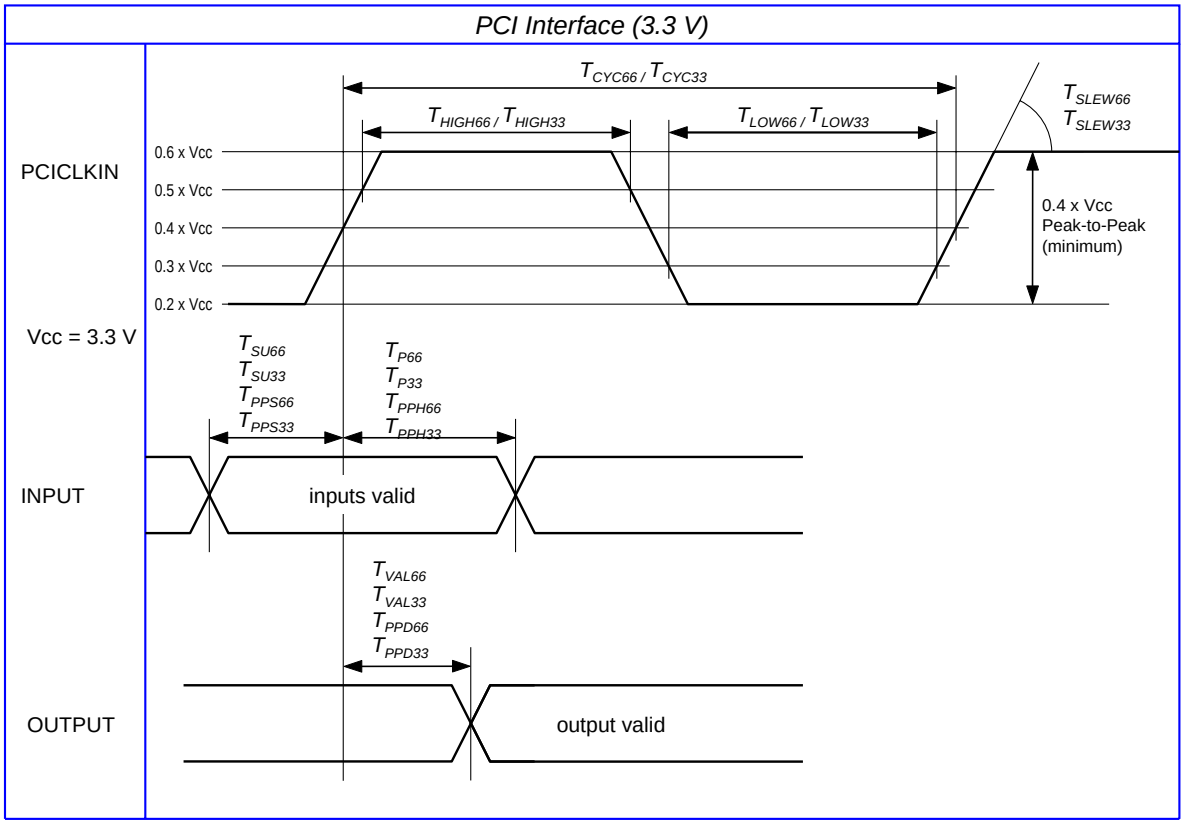


Figure 28-11 PCI Interface (3.3 V)

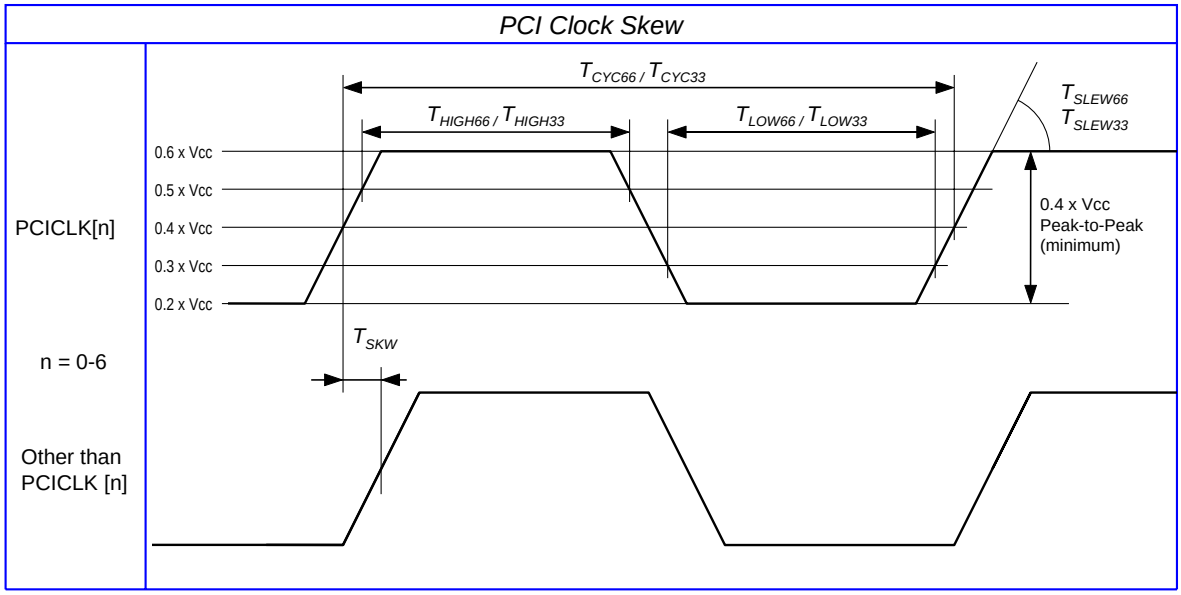


Figure 28-12 PCI Clock Skew

28.4.6. AC-link Interface AC characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Condition	Min.	Max.	Unit
BITCLK High Time	t_{HIGH_BCLK}		36	45	ns
BITCLK Low Time	t_{LOW_BCLK}		36	45	ns
SYNC Output Delay Time	t_{VAL_SYNC}	BITCLK reference, CL = 55 pF	—	15	ns
SDOUT Output Delay Time	t_{VAL_SDOUT}	BITCLK reference, CL = 55 pF	—	15	ns
SDIN[1:0] Input Setup Time	t_{SU_DSIN}	BITCLK reference	10	—	ns
SDIN[1:0] Input Hold Time	t_{HO_DSIN}	BITCLK reference	10	—	ns

AC-Link Interface Timing Diagram

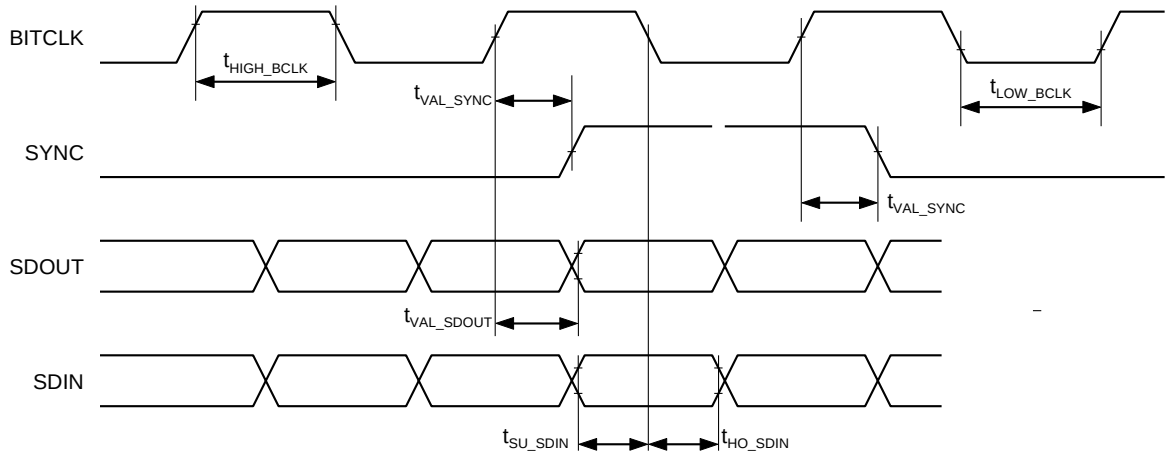


Figure 28-13 Timing Diagram: AC-link Interface

28.4.7. SPI AC characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
SPICLK Cycle Time	f_{CYC_SPICLK}		$2 \times t_{MCP}$	—	ns
SPICLK Frequency	f_{SPICLK}		—	f_{MCK}	MHz
SPICLK High Time	t_{HIGH_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPICLK Low Time	t_{LOW_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPIOUT Output Delay Time	t_{VAL_SPIOUT}		—	TBD	ns
SPIIN Input Setup Time	t_{SU_SPIIN}		TBD	—	ns
SPIIN Input Hold Time	t_{HO_SPIIN}		TBD	—	ns

SPI Interface Timing Diagram

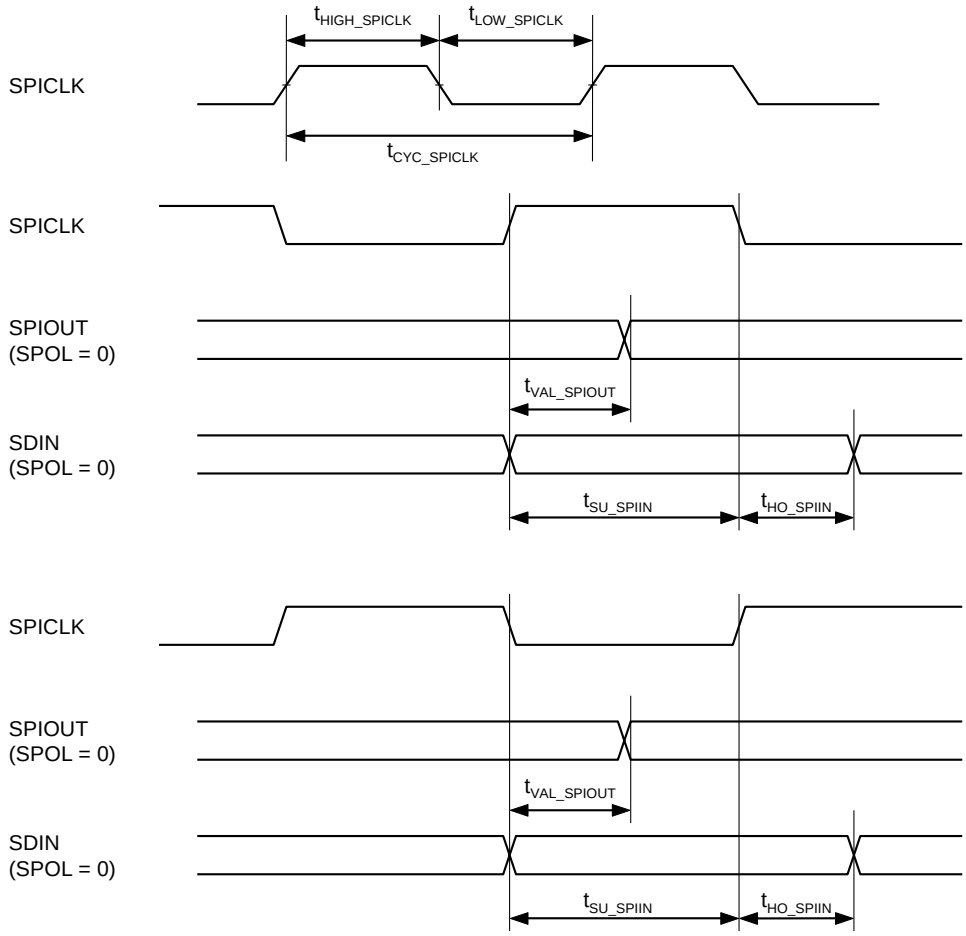


Figure 28-14 Timing Diagram: SPI Interface

28.4.8. AC characteristics of ATA Interface

DESCRIPTION		MIN	MAX
SRISE	Rising edge slew rate for any signal on AT interface (see note)		1.25 V/ns
SFALL	Falling edge slew rate for any signal on AT interface (see note)		1.25 V/ns
Chost	Host interface signal capacitance at the host connector		25 pf
Cdevice	Device interface signal capacitance at the device Connector		20 pf

NOTE : SRISE and SFALL shall meet this requirement when measured at the sender's connector from 10-90% of full signal amplitude with all capacitive loads from 15 pf through 40 pf where all signals have the same capacitive load value.

28.4.8.1. Register transfers to/from device

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Register transfer timing parameters		Mode 0 ns	Mode 1 ns	Mode 2 ns	Mode 3 ns	Mode 4 ns	Note
t_0	Cycle time (min)	600	383	330	180	120	1,4,5
t_1	Address valid to DIOR-/DIOW- Setup (min)	70	50	30	30	25	
t_2	DIOR-/DIOW- pulse width 8-bit (min)	290	290	290	80	70	1
t_{2i}	DIOR-/DIOW- recovery time (min)	-	-	-	70	25	1
t_3	DIOW- data setup (min)	60	45	30	30	20	
t_4	DIOW- data hold (min)	30	20	15	10	10	
t_5	DIOR- data setup (min)	50	35	20	20	20	
t_6	DIOR- data hold (min)	5	5	5	5	5	
t_{6z}	DIOR- data tristate (max)	30	30	30	30	30	2
t_9	DIOR-/DIOW- to address valid Hold (min)	20	15	10	10	10	
t_{RD}	Read Data Valid to IORDY active (if IORDY initially low after tA) (min)	0	0	0	0	0	
t_A	IORDY Setup time	35	35	35	35	35	3
t_B	IORDY Pulse Width (max)	1250	1250	1250	1250	1250	
t_C	IORDY Pulse Width (max)	5	5	5	5	5	

NOTE 1 t_0 is the minimum total cycle time, t_2 is the minimum DIOR-/DIOW- assertion time, and t_{2i} is the minimum DIOR-/DIOW- negation time. A host implementation shall lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

NOTE 2 This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.

NOTE 3 The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at the t_A after the activation of DIOR- or DIOW-, then t_5 shall be met and t_{RD} is not applicable. If the device is driving IORDY negated at the time t_A after the activation of DIOR- or DIOW-, then t_{RD} shall be met and t_5 is not applicable.

NOTE 4 ATA/ATAPI standards prior to ATA/ATAPI-5 inadvertently specified an incorrect value for mode 2 time t_0 by utilizing the 16-bit PIO value

NOTE 5 Mode shall be selected no faster than the highest mode supported by the slowest device.

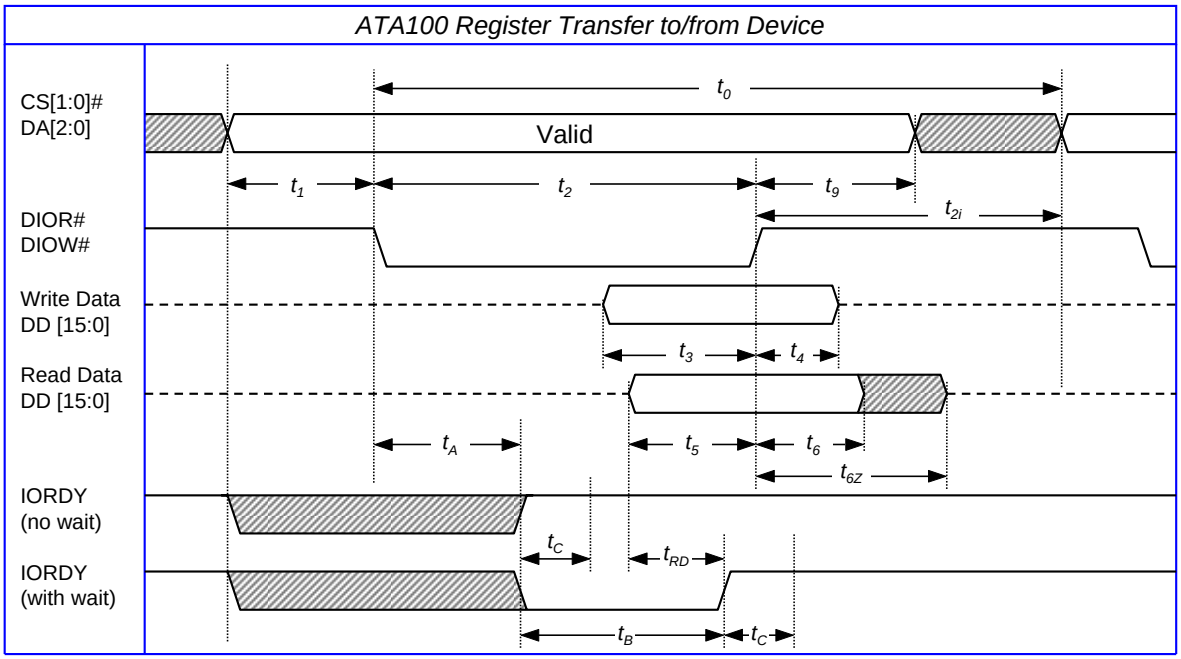


Figure 28-15 Register Transfer to/from device

28.4.8.2. Multiword DMA data transfer

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.2 V ± 0.062 V, VSS = 0 V)

Multiword DMA timing parameters			Mode 0 ns	Mode 1 ns	Mode 2 ns	Note
t_0	Cycle time	(min)	480	150	120	see note
t_D	DIOR-/DIOW- asserted pulse width	(min)	215	80	70	see note
t_E	DIOR- data access	(max)	150	60	50	
t_F	DIOR- data hold	(min)	5	5	5	
t_G	DIOR-/DIOW- data setup	(min)	100	30	20	
t_H	DIOW- data hold	(min)	20	15	10	
t_J	DMACK to DIOR-/DIOW- setup	(min)	0	0	0	
t_K	DIOR-/DIOW- to DMACK hold	(min)	20	5	5	
t_{KR}	DIOR- negated pulse width	(min)	50	50	25	see note
t_{KW}	DIOW- negated pulse width	(min)	215	50	25	see note
t_{LR}	DIOR- to DMARQ delay	(max)	120	40	35	
t_{LW}	DIOW- to DMARQ delay	(max)	40	40	35	
t_M	CS(1:0) valid to DIOR-/DIOW-	(min)	50	30	25	
t_N	CS(1:0) hold	(min)	15	10	10	
t_Z	DMACK- to read data released	(max)	20	25	25	

NOTE : t_0 is the minimum total cycle time, t_D is the minimum DIOR-/DIOW- assertion time, and t_K (t_{KR} or t_{KW}, as appropriate) is the minimum DIOR-/DIOW- negation time.
A host shall lengthen t_D and/or t_K to ensure that t_0 is equal to the value reported in the devices IDENTIFY DEVICE data.

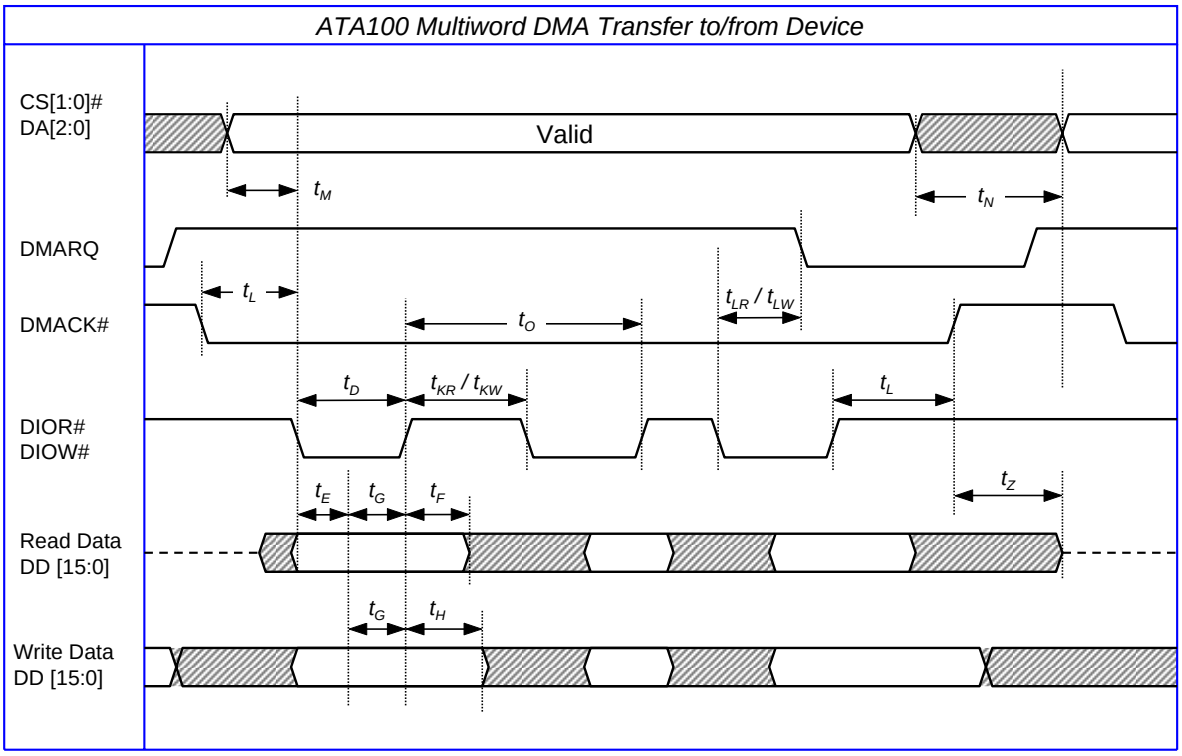


Figure 28-16 Multiword DMA data Transfer to/from device

28.4.8.3. Ultra DMA data burst timing requirements

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Name	Mode 0 (ns)		Mode 1 (ns)		Mode 2 (ns)		Mode 3 (ns)		Mode 4 (ns)		Comment (see Notes 1 and 2)
	min	max	min	max	min	max	min	max	min	max	
$t_{2CYCTYP}$	240		160		120		90		60		Typical sustained average two cycle time
t_{CYC}	112		73		54		39		25		Cycle time allowing for asymmetry and clock variations (from STROBE edge to STROBE edge)
t_{2CYC}	230		154		115		86		57		Two cycle time allowing for clock variations (from rising edge to next rising edge or from falling edge to next falling edge of STROBE)
t_{DS}	15		10		7		7		5		Data setup time at recipient
t_{DH}	5		5		5		5		5		Data hold time at recipient
t_{DVS}	70		48		30		20		6		Data valid setup time at sender (from data valid until STROBE edge) (see Note 4)
t_{DVH}	6		6		6		6		6		Data valid hold time at sender (from STROBE edge until data may become invalid) (see Note 4)
t_{FS}	0	230	0	200	0	170	0	130	0	120	First STROBE time (for device to first negate DSTROBE from STOP during a data in burst)
t_{LI}	0	150	0	150	0	150	0	100	0	100	Limited interlock time (see Note 3)
t_{MLI}	20		20		20		20		20		Interlock time with minimum (see Note 3)
t_{UI}	0		0		0		0		0		Unlimited interlock time (see Note 3)
t_{AZ}		10		10		10		10		10	Maximum time allowed for output drivers to release (from asserted or negated)
t_{ZAH}	20		20		20		20		20		Minimum delay time required for output
t_{ZAD}	0		0		0		0		0		drivers to assert or negate (from released)
t_{ENV}	20	70	20	70	20	70	20	55	20	55	Envelope time (from DMACK- to STOP and HDMARDY- during data in burst initiation and from DMACK to STOP during data out burst initiation)
t_{SR}		50		30		20		NA		NA	STROBE-to-DMARDY- time (if DMARDY- is negated before this long after STROBE edge, the recipient shall receive no more than one additional data word)
t_{RFS}		75		70		60		60		60	Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY-)
t_{RP}	160		125		100		100		100		Minimum time to assert STOP or negate DMARQ.
t_{IORDYZ}		20		20		20		20		20	Maximum time before releasing IORDY
t_{ZIORDY}	0		0		0		0		0		Minimum time before driving STROBE (see note 5)
t_{ACK}	20		20		20		20		20		Setup and hold times for DMACK- (before assertion or negation)
t_{SS}	50		50		50		50		50		Time from STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst)

NOTE 1 Timing parameters shall be measured at the connector of the sender or receiver to which the parameter applies. For example, the sender shall stop generating STROBE edges t_{RFS} after the negation of DMARDY-. Both STROBE and
DMARDY- timing measurements are taken at the connector of the sender.

NOTE 2 All timing measurement switching points (low to high and high to low) shall be taken at 1.5V.

NOTE 3 t_{UI} , t_{MLI} , and t_{LI} indicate sender-to-recipient or recipient-to-sender interlocks, i.e., either sender or recipient is waiting for the other to respond with a signal before proceeding. t_{UI} is an unlimited interlock that has no maximum time value.
t_{MLI} is a limited time-out that has a defined minimum. t_{LI} is a limited time-out that has a defined maximum.

NOTE 4 The test load for t_{DVS} and t_{DVH} shall be a lumped capacitor load with no cable or receivers. Timing for t_{DVS} and t_{DVH} shall be met for all capacitive loads from 15 to 40 pf where all signals have the same capacitive load value.

NOTE 5 t_{ZIORDY} may be greater than t_{ENV} since the device has a pull up on IORDY- giving it a known state when released.

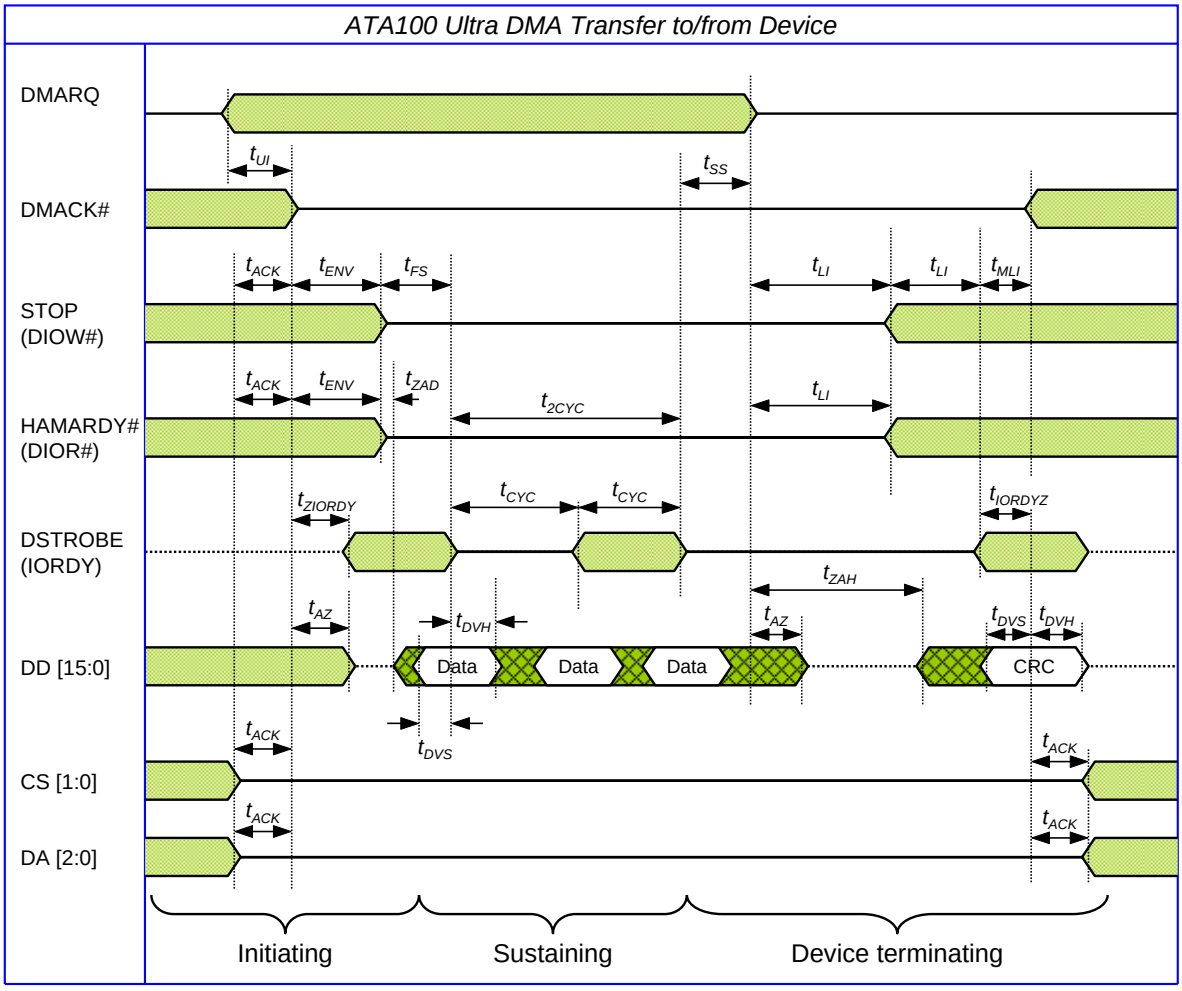


Figure 28-17 Ultra DMA data Transfer to/from device

28.4.9. Ethernet Interface (RMII) AC characteristics

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
E*REF_CLK Cycle Time	$t_{CYC\_REF\_CLK}$	**Typical period = 20ns**	—	—	ns
E*REF_CLK High Time	$t_{HIGH\_REF\_CLK}$	—	7 (TBD)	—	ns
E*REF_CLK Low Time	$t_{LOW\_REF\_CLK}$	—	7 (TBD)	—	ns
E*MDC Cycle Time	t_{CYC_MDC}	—	424 (TBD)	—	ns
E*MDC High Time	t_{HIGH_MDC}	—	212 (TBD)	—	ns
E*MDC Low Time	t_{LOW_MDC}	—	212 (TBD)	—	ns
Reception Data Setup Time	t_{SU_RX}	E*REF_CLK reference	4 (TBD)	—	ns
Reception Data Hold Time	t_{HO_RX}	E*REF_CLK reference	2 (TBD)	—	ns
Transmission Data Setup Time	-	E*REF_CLK reference	4 (TBD)	—	ns
Transmission Data Hold Time	-	E*REF_CLK reference	2 (TBD)	—	ns
Transmission Data Output Delay Time	t_{val_TX}	E*TXCLK reference CL=50 pF	0 (TBD)	10 (TBD)	ns
E*MDIO Output Delay Time	t_{val_MDIO}	E*MDC reference CL=50 pF	10 (TBD)	300 (TBD)	ns
E*MDIO Setup Time	t_{SU_MDIO}	E*MDC reference	124 (TBD)	—	ns
E*MDIO Hold Time	t_{HO_MDIO}	E*MDC reference	0 (TBD)	—	ns

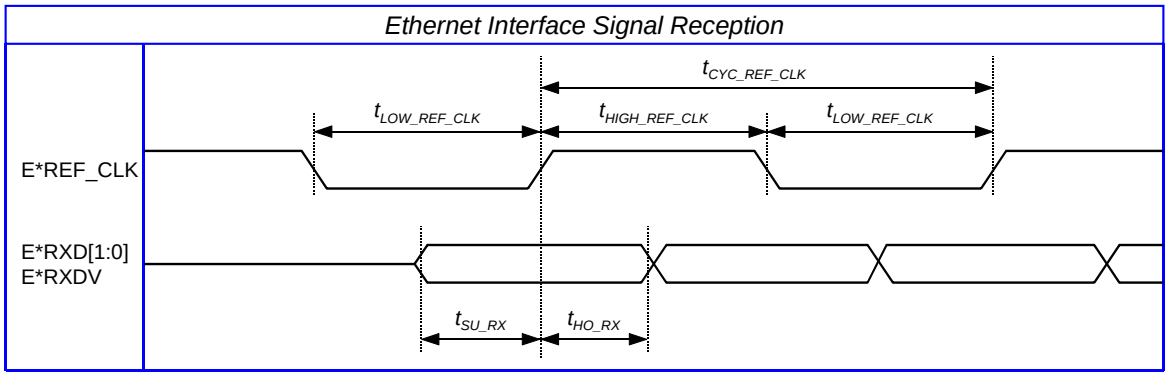


Figure 28-18 Ethernet Interface Signal Reception

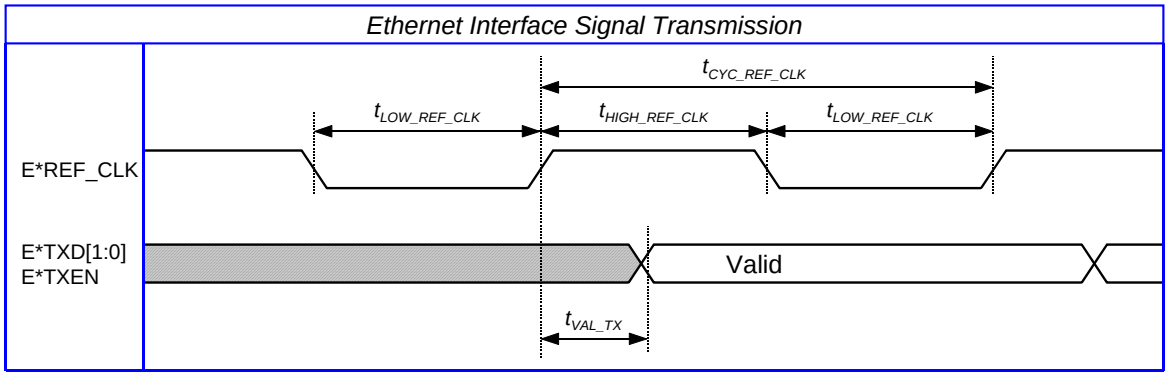


Figure 28-19 Ethernet Interface Signal Transmission

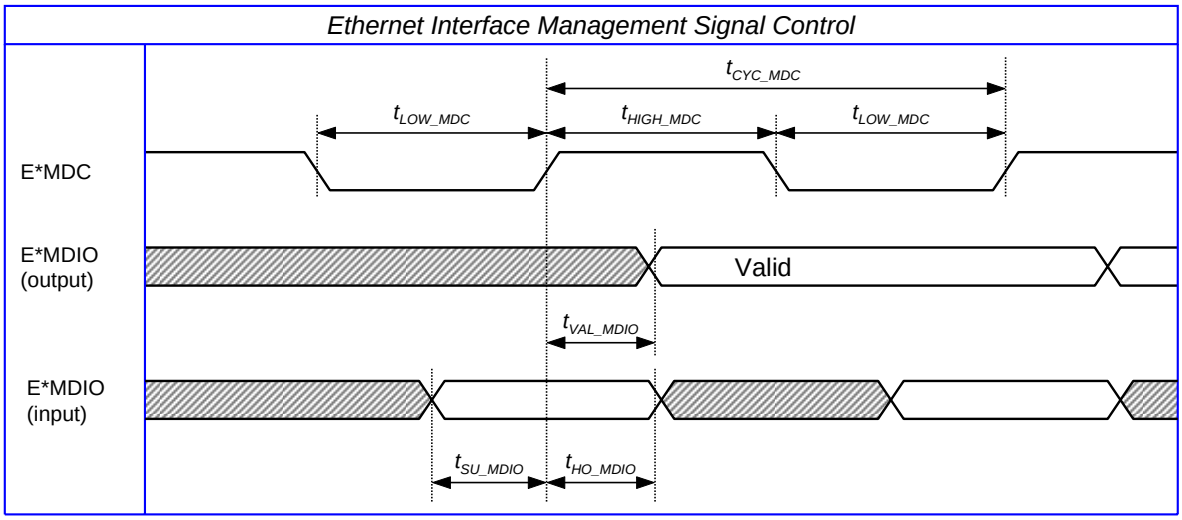


Figure 28-20 Ethernet Interface Management Signal Control

28.4.10. AC Characteristics of Video Port

28.4.10.1. Transport Data Input

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.5 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
VDINCLK0, VDOCLK0, VDINCLK1, VDOCLK1, VDINCLK2, VDOCLK2 frequency	f_{TICK}	Parallel mode		27	MHz
		Serial mode		100	MHz
TS0DATA, TS1DATA, TS2DATA setup time on rising edge	t_{TIS}	Serial mode	3		ns
TS0DATA, TS1DATA, TS2DATA hold time on rising edge	t_{TIH}	Serial mode	3		ns
TS0DATA, TS1DATA, TS2DATA setup time on rising edge	t_{TIS}	Parallel mode	5*		ns
TS0DATA, TS1DATA, TS2DATA hold time on rising edge	t_{TIH}	Parallel mode	4*		ns

* Parallel mode setup and hold time numbers only apply to VD[7:6] which are not used in serial mode.

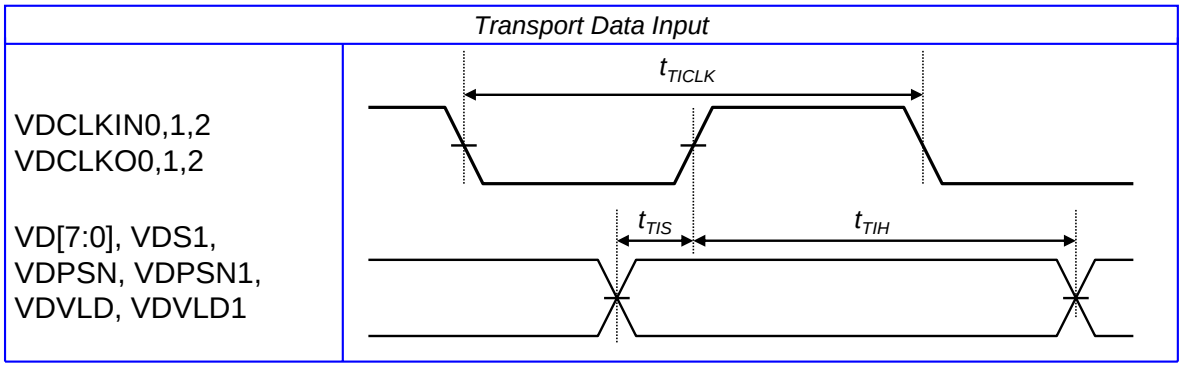


Figure 28-21 Video Port Transport data input

28.4.10.2. Transport Data Output

(Tc = 0 - 85°C, VDD33 = 3.3 V ± 0.2 V, VDD25 = 2.6 V ± 0.2 V, VDDC = 1.25 V ± 0.062 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Max.	Unit
VDCLKO0, VDCLKO1, VDCLKO2, frequency	f_{TOCK}			100	MHz
Data to VDCLKO0, VDCLKO1, VDCLKO2 delay time	t_{TODC}			3	ns
Data VDCLKO delay time	t_{TODC}	Parallel mode		28*	ns

* Parallel mode delay time number only applies to VD[7:6] which are not used in serial mode.

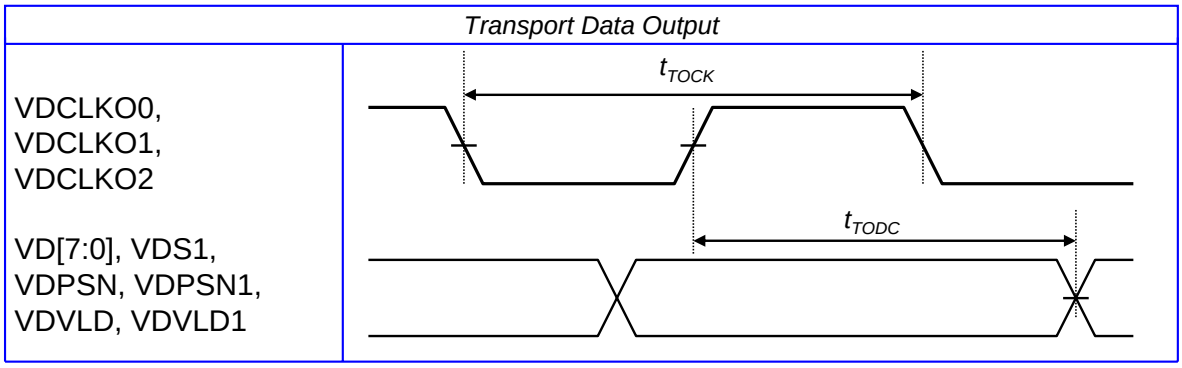
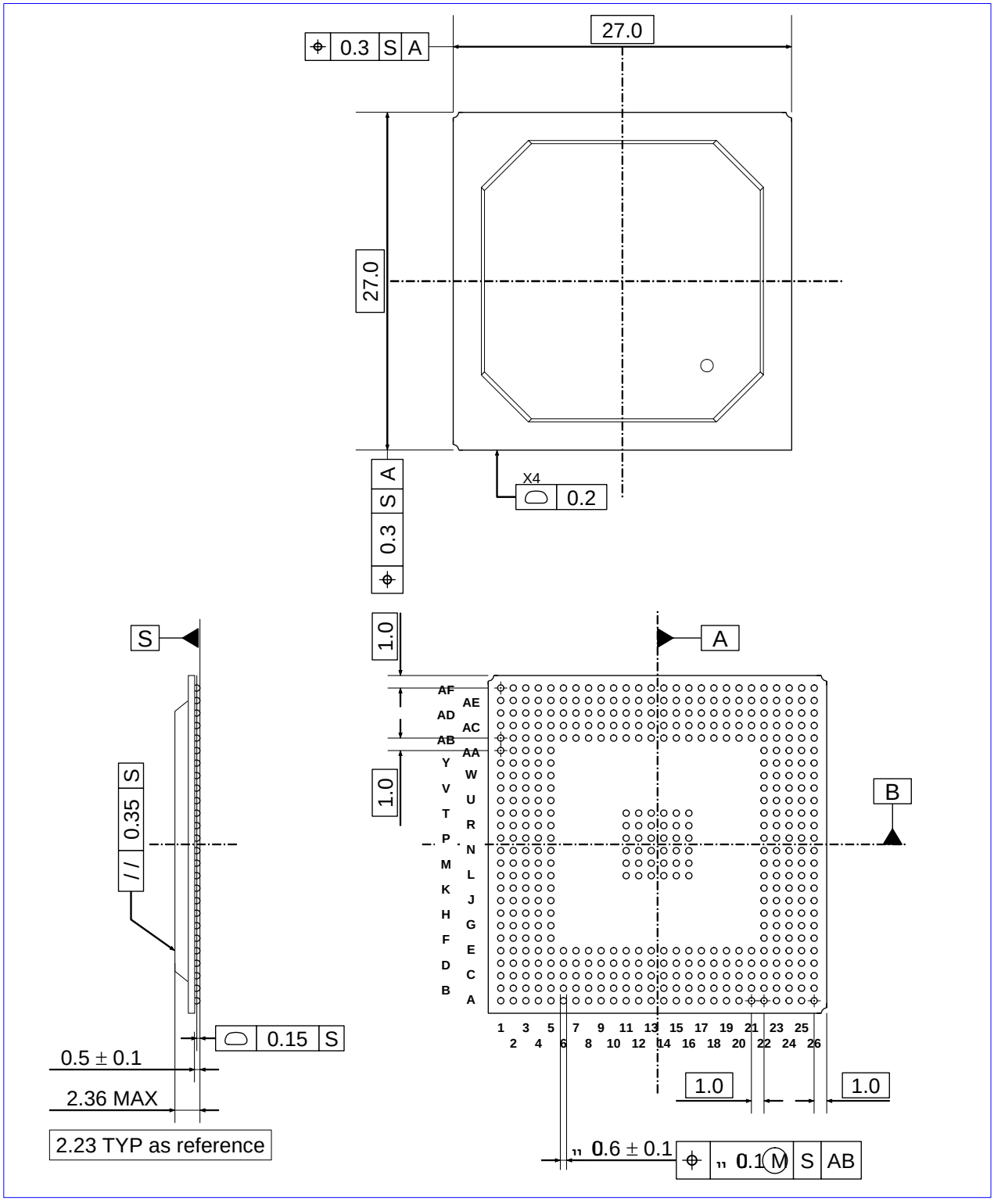


Figure 28-22 Video port Transport data output

Chapter 29. Package Outer Appearance

29.1. Package Drawing



29.2. Recommended Motherboard Footprint

Since this package, i.e. PBGA-2727-100P has thermal balls in the center of package, motherboard need to have lower thermal resistance from the footprint of these balls to ambient. Figure 29-1 shows recommended footprint for this processor.

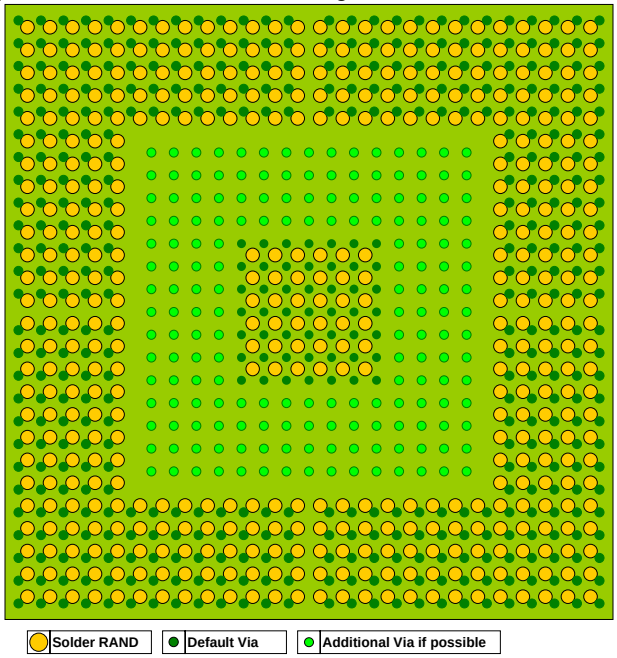


Figure 29-1 Recommended Footprint for heat disspation

There are 36 thermal balls in this package. The Via around these thermal balls are considered as thermal via. In this condition, 49 Via around thermal balls are considered default. Each thermal Via's thermal resistance can be calculated by following formula:

$$\text{Thermal Resistance} = \frac{L}{K_{copper} \times (R^2 - (R-T)^2) \times \pi}$$

where L = length of pipe (=PCB thickness), K_{copper} = thermal conductance of Copper (=395 W/mK)
R = semidiameter of through-hole, T = Copper plating thickness, π = 3.1415926

Since the heat dissipates not only center ball but also balls around peripheral, these thermal via acts additional heat dissipation path. Therefore it is recommended to build more via at the area of "Additional Via" in Figure 29-1.
Also it is recommended to increase copper coverage of the internal planes both thermal ball area and additional area.
Table 29-1 is the result of calculation for thermal resistance of each Via and 49 Via.

Table 29-1 Thermal Resistance of Via and Vias

PCB thickness = 1.6 mm	Thermal Resistance	
Plating thickness = 0.015 mm	Per Via	With 49 Via
Diameter of Via = 0.25	365.8 DegC/W	7.5 DegC/W
Diameter of Via = 0.30	301.6 DegC/W	6.2 DegC/W
Diameter of Via = 0.35	256.6 DegC/W	5.2 DegC/W
Diameter of Via = 0.40	223.3 DegC/W	4.6 DegC/W
Diameter of Via = 0.45	197.6 DegC/W	4.0 DegC/W
Diameter of Via = 0.50	177.2 DegC/W	3.6 DegC/W