32-Bit TX System RISC
TX39 Family
TMPR3911/3912

TOSHIBA CORPORATION

MIPS16, application Specific Extensions and R3000A 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.

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 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.

Preface

Thank you for choosing Toshiba semiconductor products. This is the 2000 edition of the databook for the TMPR3911/12 32-bit TX System RISC microprocessor.

This databook is designed to be easily understood by engineers who are designing the Toshiba TMPR3911/12 RISC microprocessor into their products for the first time. No special knowledge of this device is assumed – the contents includes basic information about the microprocessor and the application fields in which it is used. In addition, complete technical specifications are given, facilitating incorporation of the device into any given user application.

Toshiba are continually updating technical publications. Any comments and suggestions regarding any Toshiba document are most welcome and will be taken into account when subsequent editions are prepared. To receive updates to the information in this databook, or for additional information about the products described in it, please contact your nearest Toshiba office or authorized Toshiba dealer.

August 2003

Table of Contents

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.

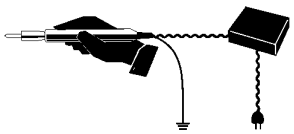
(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

(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).

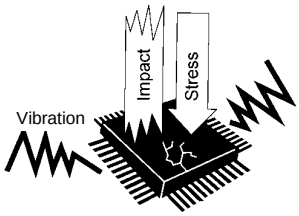
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.

- 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.

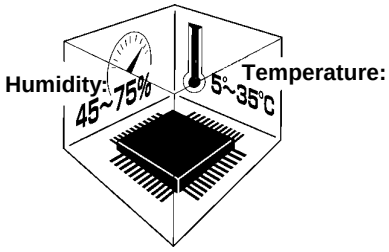
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

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.

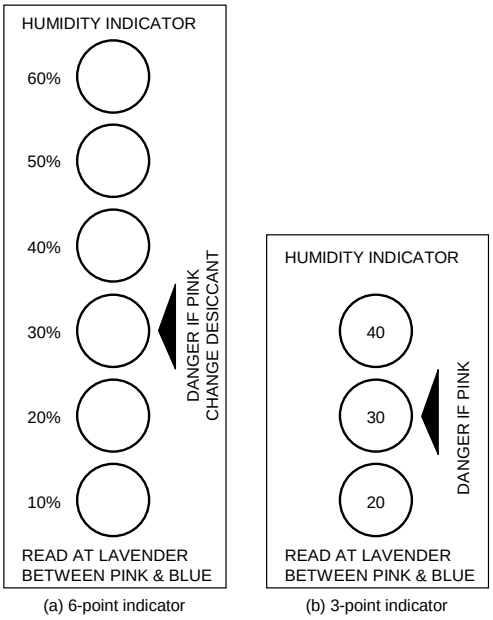


Figure 1 Humidity indicator

3.3 Design

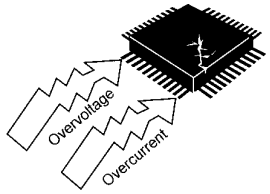
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will 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)

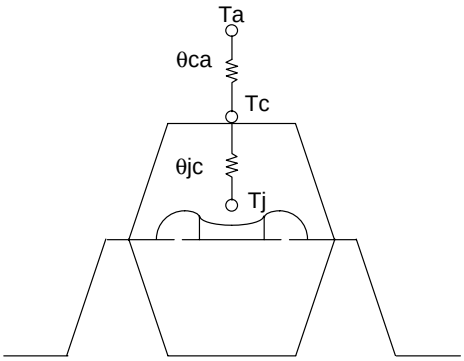


Figure 2 Thermal resistance of package

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

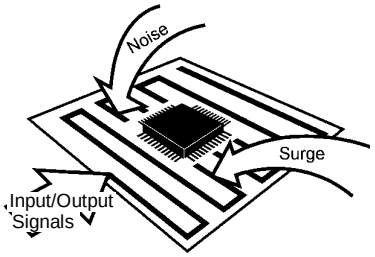
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with 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.

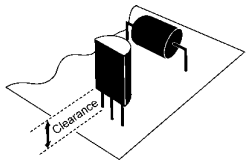
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

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 a soldering iron

Complete soldering within ten seconds for lead temperatures of up to 260°C, or within three seconds for lead temperatures of up to 350°C.

(2) Using medium infrared ray reflow

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

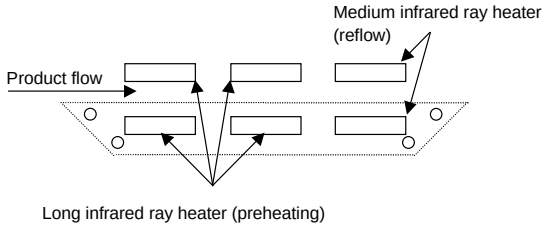


Figure 3 Heating top and bottom with long or medium infrared rays

- Complete the infrared ray reflow process within 30 seconds at a package surface temperature of between 210°C and 240°C.
- Refer to Figure 4 for an example of a good temperature profile for infrared or hot air reflow.

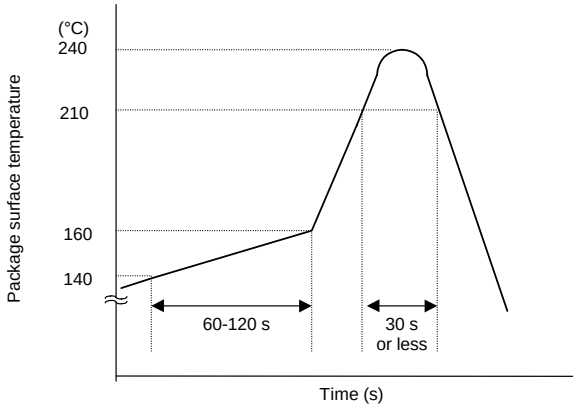


Figure 4 Sample temperature profile for infrared or hot air reflow

(3) Using hot air reflow

- Complete hot air reflow within 30 seconds at a package surface temperature of between 210°C and 240°C.
- For an example of a recommended temperature profile, refer to Figure 4 above.

(4) Using solder flow

- Apply preheating for 60 to 120 seconds at a temperature of 150°C.
- For lead insertion-type packages, complete solder flow within 10 seconds with the temperature at the stopper (or, if there is no stopper, at a location more than 1.5 mm from the body) which does not exceed 260°C.
- For surface-mount packages, complete soldering within 5 seconds at a temperature of 250°C or

less in order to prevent thermal stress in the device.

- Figure 5 shows an example of a recommended temperature profile for surface-mount packages using solder flow.

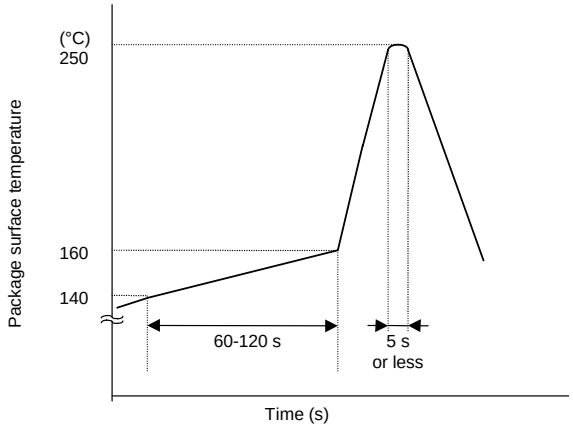


Figure 5 Sample temperature profile for solder flow

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.

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

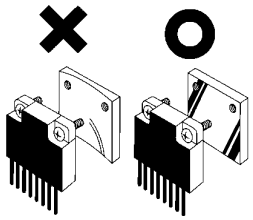
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects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

TMPR3911/3912

1. TMPR3911/12 Overview

1.1 Overview

The TMPR3911/12 is the single-chip, integrated digital ASSP for the Personal Digital Assistants (PDA). Figure 1.1.1 shows a block diagram of the overall PDA system. The TMPR3911/12 consists of the PDA system support logic, integrated with an embedded TX39/H Processor core designed by TOSHIBA.

The TMPR3911/12 consists of a TX39/H Processor core with 4 Kbytes of instruction cache memory and 1 Kbyte of data cache memory, plus integrated functions for interfacing to numerous system components and external I/O modules. The TX39/H Processor core is also augmented with a single-cycle multiply/accumulate module to allow integrated DSP functions, such as a software modem for high-performance standard data and fax protocols. The TMPR3911/12 also contains multiple DMA channels and a high-performance and flexible Bus Interface Unit (BIU) for providing an efficient means for transferring data between external system memory, cache memory, the Processor core, and external I/O modules. The types of external memory devices supported include dynamic random access memory (DRAM), synchronous dynamic random access memory (SDRAM), static random access memory (SRAM), Flash memory, read-only memory (ROM), and expansion cards (PCMCIA and/or MagicCard). The TMPR3911/12 also contains a System Interface Module (SIM) containing integrated functions for interfacing to numerous external I/O modules such as liquid crystal displays (LCD's), the TC35143 (which handles most of the analog functions of the system, including sound and telecom codecs and touchscreen ADC), ISDN/high-speed serial, infrared, wireless peripherals, Magicbus, etc. Lastly, The TMPR3911/12 contains support for implementation of power management for PDA system, whereby various TMPR3911/12 internal modules and external subsystems can be individually (under software control) powered up and down. The TMPR3911/12 contains the following overall features:

- high level of integration on a single chip, small board space, low pin count, low power, and high performance
  - Plastic LQFP208-pin package (TMPR3912AU-92)
    FBGA 217-pin package (TMPR3912XB-92)
    Plastic LQFP176-pin package (TMPR3911BU)
    FBGA177-pin package (TMPR3911BXB)
  - 32-bit TX39/H Processor core, cache memory, multiply-accumulate module, multi-channel DMA controller, bus interface unit and memory controller, power management, and other peripheral subsystems all on a single integrated chip
  - minimal number of inter-chip connections
  - Maximum operation frequency: 58 MHz (TMPR3911BU and TMPR3911BXB)
    92 MHz (TMPR3912AU-92 and XB-92)
- low power consumption
  - entire TMPR3911/12 operation is 3.3 V (in case of TMPR3911BU/BXB, I/O: 3.3 V, internal: 2.6 V)
  - real-time clock based on 32.768 kHz reference
  - Processor core clock stop state for low standby current
  - power-down modes for individual internal peripheral modules

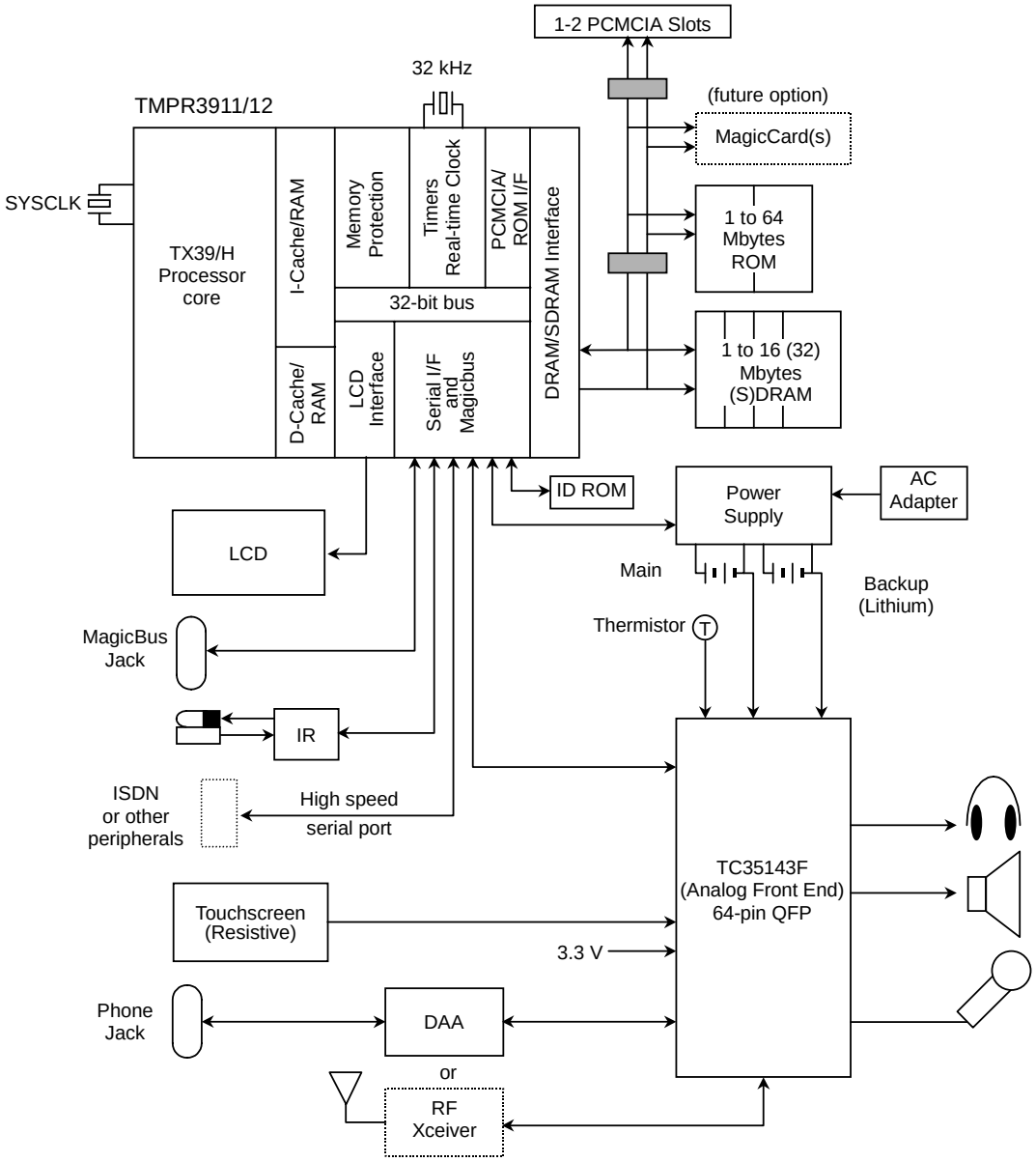


Figure 1.1.1 PDA Block Diagram

Figure 1.1.2 shows a block diagram of the TMPR3911/12. The key functions and features for each TMPR3911/12 module are itemized below.

Processor core Module

- TX39/H Processor core
  - full 32-bit operation (registers, instructions, addresses, etc.)
  - 32 general purpose 32-bit registers; 32-bit program counter
  - MIPS RISC Instruction Set Architecture (ISA) supported
  - Translation Look-aside Buffer (TLB) (4K Page size, 32 entries)
- on-chip cache
  - 4 Kbyte instruction cache (I-cache)
    - physical address tag and valid bit per cache line
    - programmable burst size
    - instruction streaming mode supported
    - direct-mapped
  - 1 Kbyte data cache (D-cache)
    - physical address tag and valid bit per cache line
    - programmable burst size
    - write-through
    - two-way set associative
  - cache address snoop mode supported for DMA
  - 4-level deep write buffer
- high-speed multiplier/accumulator
  - on-chip hardware multiplier
  - supports 32×32 multiplier operations, with 64-bit accumulator
  - existing multiply instructions are enhanced and new multiply and add instructions are added to the R3000A instruction set to improve the performance of DSP applications
- Processor core interface
  - handles data bus, address bus, and control interface between Processor core and rest of the TMPR3911/12 logic

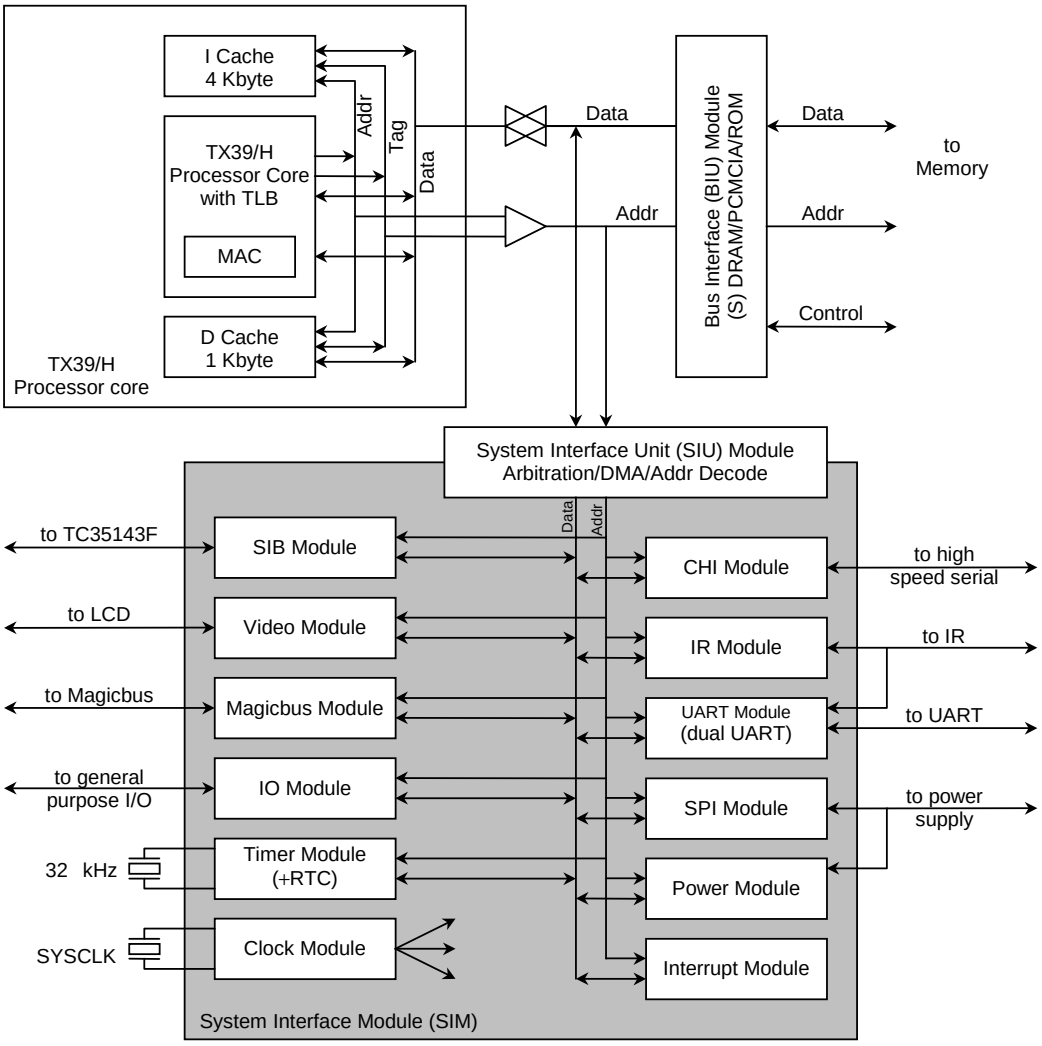


Figure 1.1.2 TMPR3911/12 Block Diagram

BIU Module

- system memory and the TMPR3911/12 Bus Interface Unit (BIU)
  - supports up to 2 banks of physical memory
  - supports self-refreshing DRAM and SDRAM
  - programmable parameters for each bank of DRAM or SDRAM (row/column address configuration, refresh, burst modes, etc.)
- programmable chip select memory access
  - 4 programmable (size, wait states, burst mode control) memory device and general purpose chip selects
    - available for system ROM, SRAM, Flash
    - available for external port expansion registers
  - 4 programmable (wait states, burst mode control) MagicCard or general purpose chip selects
    - the TMPR3911/12 provides the chip select and card detect signals
    - supports card insertion/removal timeouts
- supports up to 2 identical PCMCIA ports
  - the TMPR3911/12 and the TC35143 provide the control signals and accepts the status signals
  - appropriate connector keying and level-shifting buffers required for 3.3 V versus 5 V PCMCIA interface implementations

SIU Module

- multi-channel 32-bit DMA controller and System Interface Unit (SIU)
- independent DMA channels for video, Magicbus, SIB to/from TC35143 audio/telecom codecs, high-speed serial port, IR UART, and general purpose UART
- address decoding for submodules within System Interface Module (SIM)

Clock Module

- the TMPR3911/12 supports system-wide single crystal configuration, besides the 32 kHz RTC XTAL (reduces cost, power, and board space)
- common crystal rate divided to generate clock for CPU, video, sound, telecom, UARTs, etc.
- independent enabling or disabling of individual clocks under software control, for power management

CHI Module

- high-speed serial Concentration Highway Interface (CHI) contains logic for interfacing to external full-duplex serial time-division-multiplexed (TDM) communication peripherals
- supports ISDN line interface chips and other PCM/TDM serial devices
- CHI interface is programmable (number of channels, frame rate, bit rate, etc.) to provide support for a variety of formats
- supports data rates up to 4.096 Mbps
- independent DMA support for CHI receive and transmit

Interrupt Module

- contains logic for individually enabling, reading, and clearing all TMPR3911/12 interrupt sources
- interrupts generated from internal TMPR3911/12 modules or from edge transitions on external signal pins

I/O Module

- contains support for reading and writing the 7 bi-directional general purpose I/O pins and the 32 bi-directional multi-function I/O pins
- each I/O port can generate a separate positive and negative edge interrupt
- independently configurable I/O ports allow the TMPR3911/12 to support a flexible and wide range of system applications and configurations

IR Module

- IR consumer mode
  - allows control of consumer electronic devices such as stereos, TVs, VCRs, etc.
  - programmable pulse parameters
  - external analog LED circuitry
- IRDA communication mode
  - allows communication with other IRDA devices such as FAX machines, copiers, printers, etc.
  - supported by the UART module within the TMPR3911/12
  - external analog receiver preamplifier and LED circuitry
  - data rate = up to 115 kbps at 1 meter
- IR FSK communication mode
  - supported by the UART module within the TMPR3911/12
  - external analog IR chip(s) perform frequency modulation to generate the desired IR communication mode protocol
  - data rate = up to 36000 bps at 3 meters
- carrier detect state machine
  - periodically enables IR receiver to check if a valid carrier is present

Magicbus Module

- synchronous, serial 2-wire (clock and data), half-duplex communications protocol
- supports low-cost, low-power peripherals
- supports maximum data rate of 14.75 Mbps
- DMA support for Magicbus receive and transmit

Power Module

- power-down modes for individual internal peripheral modules
- serial (SPI port) power supply control interface supported
- power management state machine has 3 states: RUNNING, DOZING and SLEEP

SIB Module

- the TMPR3911/12 contains holding and shift registers to support the serial interface to the TC35143 ASIC and/or other optional codec devices
- interface compatible with slave mode 3 of the Crystal CS4216 codec
- synchronous, frame-based protocol
- the TMPR3911/12 always master source of clock and frame frequency and phase ; programmable clock frequency
- each SIB frame consists of 128 clock cycles, further divided into 2 subframes or words of 64 bits each (supports up to 2 devices simultaneously)
- independent DMA support for audio receive and transmit, telecom receive and transmit
- supports 8-bit or 16-bit mono telecom formats
- supports 8-bit or 16-bit mono or stereo audio formats
- independently programmable audio and telecom sample rates
- CPU read/write registers for subframe control and status

SPI Module

- provides interface to SPI peripherals and devices
- full-duplex, synchronous serial data transfers (data in, data out, and clock signals)
- the TMPR3911/12 supplies dedicated chip select and interrupt for an SPI interface serial power supply
- 8-bit or 16-bit data word lengths for the SPI interface
- programmable SPI baud rate

Timer Module

- Real Time Clock (RTC) and Timer
- 40-bit counter (30.517 μs granularity); maximum uninterrupted time = 388.36 days
- 40-bit alarm register (30.517 μs granularity)
- 16-bit periodic timer (0.868 μs granularity at SYSCLK = 75 MHz); maximum timeout = 56.8 ms
- interrupts on alarm, timer, and prior to RTC roll-over

UART Module

- 2 independent full-duplex UARTs
- programmable baud rate generator
- UARTB port used for serial control interface to external IR module
- UARTA port used for general purpose serial control interface
- UARTA and UARTB DMA support for receive and transmit

Video Module

- bit-mapped graphics
- supports monochrome, gray scale, or color modes
- time-based dithering algorithm for gray scale and color modes
- supports multiple screen sizes
- supports split and non-split displays
- variable size and relocatable video buffer
- DMA support for fetching image data from video buffer

1.2 References

(1) MIPS RISC Architecture, Gerry Kane and Joe Heinrich, Prentice Hall, 1992
This book is a comprehensive reference for the MIPS RISC Instruction Set Architecture (ISA). This book also describes implementation-specific architectural features for each of the MIPS RISC processor families. This book includes descriptions of CPU architecture, memory management, cache architecture, instruction set summary, exception processing, floating-point unit, and assembly language programming.

(2) PC Card Standard, Release 2.01, Personal Computer Memory Card International Association, November 1992
This document provides a description of the standard for PC Card interchangeability. This includes card and socket physical and mechanical specifications, connector pin description and electrical interface specifications, card data formats and header organization, and execute in place specifications. This is not a General Magic document, and availability of this document is controlled by the Personal Computer Memory Card International Association.

(3) TX39 Family Users Manual, TOSHIBA

(4) Computer Architecture: A Quantitative Approach
Second Edition
John L Hennessy & David A Patterson
Morgan Kaufmann Publishers, Inc.
ISBN 1-55860-329-8

(5) See MIPS Run
Dominic Sweetman
Morgan Kaufmann Publishers, Inc.
ISBN 1-55860-410-3

(6) MIPS Technologies Inc. Publications
http://www.mips.com/publications/

(7) SGI Document Archive
ftp://sgigate.sgi.com/pub/doc/

2. Pin Descriptions

2.1 Overview

The TMPR3912 contains 208 pins (LQFP) or 217 pins (FBGA) consisting of input, output, bi-directional, and power and ground pins. The TMPR3911 contains 176 pins (LQFP) or 177 pins (FBGA) consisting of input, output, bi-directional, and power and ground pins. The only difference between TMPR3911 and TMPR3912 is pin counts of power and ground pins. These pins are used to support various functions. The following sections describe the function of each pin including any special power-down considerations for each pin.

2.2 Pins

The TMPR3912 contains 137 signal pins, 3 spare pins, 34 power pins, and 34 ground pins. The TMPR3911 contains 137 signal pins, 3 spare pins, 19 power pins and 17 ground pins. Of the 137 signal pins, 32 of them are multi-function and can be independently programmed either as IO ports or for an alternate standard/normal function. As an IO port, any of these pins can be programmed as an input or output port, with the capability of generating a separate positive and negative edge interrupt. See Section 2.3 for a summary of the multi-function IO ports versus their standard functions.

2.2.1 Memory Pins

D[31:0]: INPUT/OUTPUT

These pins are the data bus signals for the system. These pins are normally outputs and only become inputs during reads, thus no resistors are required since the bus will only float for a short period of time during bus turn-around. For more information, See Section 4.3.2.

A[12:0]: OUTPUT

These pins are the address bus signals for the system. The address lines are multiplexed and can be connected directly to SDRAM and DRAM devices. To generate the full 26-bit address for static devices, an external latch must be used to latch the signals using the ALE signal. For static devices, address bits 25:13 are provided by the external latch and address bits 12:0 (directly connected from the TMPR3911/12's address bus) are held afterward by the TMPR3911/12 for the remainder of the address bus cycle.

ALE: OUTPUT

This pin is used as the address latch enable signal to latch A[12:0] using an external latch, for generating the upper address bits 25:13.

RD*: OUTPUT

This pin is used as the read signal for static devices. This signal is asserted for reads from MCS3-0*, CS3-0*, CARD2CS* and CARD1CS* for memory and attribute space.

WE*: OUTPUT

This pin is used as the write signal for the system. This signal is asserted for writes to MCS3-0*, CS3-0*, CARD2CS* and CARD1CS* for memory and attribute space, and for writes to DRAM and SDRAM.

CAS0* (WE0*): OUTPUT

This pin is used as the CAS signal for SDRAMs, the CAS signal for D[7:0] for DRAMs, and the write enable signal for D[7:0] for static devices.

CAS1* (WE1*): OUTPUT

This pin is used as the CAS signal for D[15:8] for DRAMs and the write enable signal for D[15:8] for static devices.

CAS2* (WE2*): OUTPUT

This pin is used as the CAS signal for D[23:16] for DRAMs and the write enable signal for D[23:16] for static devices. Also used as the bank select signal for SDRAM (applies only to TMPR3912XB-92/AU-92)

CAS3* (WE3*): OUTPUT

This pin is used as the CAS signal for D[31:24] for DRAMs and the write enable signal for D[31:24] for static devices.

RAS0*: OUTPUT

This pin is used as the RAS signal for SDRAMs and the RAS signal for Bank0 DRAMs.

RAS1* (DCS1*): OUTPUT

This pin is used as the chip select signal for Bank1 SDRAMs and the RAS signal for Bank1 DRAMs.

DCS0*: OUTPUT

This pin is used as the chip select signal for Bank0 SDRAMs.

DCKE: OUTPUT

This pin is used as the clock enable for SDRAMs.

DCLKIN: INPUT

This pin must be tied externally to the DCLKOUT signal and is used to match skew for the data input when reading from SDRAM and DRAM devices.

DCLKOUT: OUTPUT

This pin is the clock for the SDRAMs.

DQMH: OUTPUT

This pin is the upper data mask for a 16-bit SDRAM configuration.

DQML: OUTPUT

This pin is the lower data mask for a 16-bit SDRAM or 8-bit SDRAM configuration.

CS3-0*: OUTPUT

These pins are the Chip Select 3 through 0 signals. They can be configured to support either 32-bit or 16-bit ports.

MCS3-0*: OUTPUT

These pins are the MagicCard Chip Select 3 through 0 signals. They only support 16-bit ports.

CARD2CSH*, L*: OUTPUT

These pins are the Chip Select signals for PCMCIA card slot 2.

CARD1CSH*, L*: OUTPUT

These pins are the Chip Select signals for PCMCIA card slot 1.

CARDREG*: OUTPUT

This pin is the REG* signal for the PCMCIA cards.

CARDIORD*: OUTPUT

This pin is the IORD* signal for the PCMCIA IO cards.

CARDIOWR*: OUTPUT

This pin is the IOWR* signal for the PCMCIA IO cards.

CARDDIR*: OUTPUT

This pin is used to provide the direction control for bi-directional data buffers used for the PCMCIA slot(s). This signal will assert whenever CARD2CSH* or CARD2CSL* or CARD1CSH* or CARD1CSL* is asserted and a read transaction takes place.

CARD2WAIT*: INPUT

This pin is the card wait signal from PCMCIA card slot 2.

CARD1WAIT*: INPUT

This pin is the card wait signal from PCMCIA card slot 1.

2.2.2 Bus Arbitration Pins

DREQ*: INPUT

This pin is used to request external arbitration. The TMPR3911/12 memory transactions are halted and certain memory signals will be tri-stated when DGRNT* is asserted in order to allow an external master to access memory.

DGRNT*: OUTPUT

This pin is asserted in response to DREQ* to inform the external test logic or bus master that it can now begin to drive signals.

2.2.3 Clock Pins

SYSCLKIN: INPUT

This pin should be connected along with SYSCLKOUT to an external crystal which is the main TMPR3911/12 clock source.

SYSCLKOUT: OUTPUT

This pin should be connected along with SYSCLKIN to an external crystal which is the main TMPR3911/12 clock source.

C32KIN: INPUT

This pin along with C32KOUT should be connected to a 32.768 kHz crystal.

C32KOUT: OUTPUT

This pin along with C32KIN should be connected to a 32.768 kHz crystal.

BC32K: OUTPUT

This pin is a buffered output of the 32.768 kHz clock.

2.2.4 CHI Pins

CHIFS: INPUT/OUTPUT

This pin is the CHI frame synchronization signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3911/12 to be the master CHI sync source. As an input, this pin allows an external peripheral to be the master CHI sync source and the TMPR3911/12 CHI module will slave to this external sync.

CHICLK: INPUT/OUTPUT

This pin is the CHI clock signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3911/12 to be the master CHI clock source. As an input, this pin allows an external peripheral to be the master CHI clock source and the TMPR3911/12 CHI module will slave to this external clock.

CHIDOUT: OUTPUT

This pin is the CHI serial data output signal.

CHIDIN: INPUT

This pin is the CHI serial data input signal.

2.2.5 IO Pins

IO[6:0]: INPUT/OUTPUT

These pins are general purpose input/output ports. Each port can be independently programmed as an input or output port. Each port can generate a separate positive and negative edge interrupt. Each port can also be independently programmed to use a 16 to 24 ms debouncer.

MFIO[1:0]: INPUT/OUTPUT

These pins are multi-function input/output ports. Each port can be independently programmed as an input or output port, or can be programmed for multi-function use to support test signals (for debugging purposes only). Each port can generate a separate positive and negative edge interrupt. Note that 30 other multi-function pins are available for usage as multi-function input/output ports. These pins are named after their respective standard/normal function and are not listed here.

2.2.6 Magicbus Pins

MBUSCLK: INPUT/OUTPUT

This pin is the bi-directional Magicbus clock signal. MBUSCLK is an input signal whenever the TMPR3911/12 is in the slave mode and is an output signal whenever the TMPR3911/12 is in the master mode.

MBUSDATA: INPUT/OUTPUT

This pin is the bi-directional Magicbus data signal. MBUSDATA is an input signal whenever the TMPR3911/12 is in the slave mode and is an output signal whenever the TMPR3911/12 is in the master mode.

MBUSINT: INPUT

This pin is the Magicbus interrupt signal. This signal is used to interrupt the TMPR3911/12 whenever a peripheral has been attached to or detached from the bus, or whenever a peripheral has initiated an interrupt event.

2.2.7 Reset Pins

CPURES*: INPUT

This pin is used to reset the Processor core. This pin should be connected to a switch for initiating a reset in the event that a software problem might hang the Processor core. The pin should also be pulled up to VSTANDBY (external signal) through an external pull-up resistor.

PON*: INPUT

This pin serves as the Power On Reset signal for the TMPR3911/12. This signal must remain low when VSTANDBY (external signal) is asserted until VSTANDBY is stable. Once VSTANDBY is asserted, this signal should never go low unless all power is lost in the system.

2.2.8 Power Module Pins

ONBUTN: INPUT

Asserting this signal will cause PWRCS to be asserted if the PWROK signal is high.

PWRCS: OUTPUT

This signal is asserted automatically if the ONBUTN signal is asserted when the PWROK signal is high. Clearing the PWRCS bit in the Power Control Register by software deasserts this signal.

PWROK: INPUT

This pin should be asserted when the source of power is stable.

When the PWROKNMI bit in the Interrupt Status 6 Register is set, the deassertion of the PWROK signal causes the NMI (Non Maskable Interrupt).

PWRINT: INPUT

The rising edge and falling edge of signal to this pin causes PWRINT interrupt request.

VCC3: INPUT

This pin should be asserted when the source of power for external circuit is stable.

Asserting this pin starts oscillating the internal clock.

For more details of these pins, see chapter 12.

2.2.9 SIB Pins

SIBDIN: INPUT

This pin is used for serial input to the SIB module.

SIBDOUT: OUTPUT

This pin is used for serial output from the SIB module.

SIBSCLK: OUTPUT

This pin is the clock for SIB interface.

SIBSYNC: OUTPUT

This pin is the frame synchronization signal for SIB interface. This frame sync is asserted for one clock cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data.

SIBIRQ: INPUT

The rising edge and falling edge of signal to this pin causes SIBIRQ interrupt request.

SIBMCLK: INPUT/OUTPUT

This pin is the master clock source for the SIB module.

2.2.10 SPI Pins

SPICLK: OUTPUT

This pin is used to clock data in and out of the SPI slave device.

SPIOUT: OUTPUT

This pin contains the data that is shifted into the SPI slave device.

SPIIN: INPUT

This pin contains the data that is shifted out of the SPI slave device.

2.2.11 UART and IR Pins

TXD: OUTPUT

This pin is the UART transmit signal from the UARTA module.

RXD: INPUT

This pin is the UART receive signal to the UARTA module.

IROUT: OUTPUT

This pin is the UART transmit signal from the UARTB module or the Consumer IR output signal if Consumer IR mode is enabled.

IRIN: INPUT

This pin is the UART receive signal to the UARTB module.

RXPWR: OUTPUT

This pin is the receiver power output control signal to the external communication IR analog circuitry.

CARDET: INPUT

This pin is the carrier detect input signal from the external communication IR analog circuitry.

2.2.12 Video Pins

FRAME: OUTPUT

This pin is the frame synchronization pulse signal between the Video Module and the LCD, and is used by the LCD to return its pointers to the top of the display. The Video Module asserts FRAME after all the lines of the LCD have been shifted and transferred, producing a full frame of display.

DF: OUTPUT

This pin is the AC signal for the LCD. Since LCD plasma tends to deteriorate whenever subjected to a DC voltage, the DF signal is used by the LCD to alternate the polarity of the row and column voltages used to turn the pixels on and off. The DF signal can be configured to toggle on every frame or can be configured to toggle every programmable number of LOAD signals.

LOAD: OUTPUT

This pin is the line synchronization pulse signal between the Video Module and the LCD, and is used by the LCD to transfer the contents of its horizontal line shift register to the LCD panel for display. The Video Module asserts LOAD after an entire horizontal line of data has been shifted into the LCD.

CP: OUTPUT

This pin is the clock signal for the LCD. Data is pushed by the Video Module on the rising edge of CP and sampled by the LCD on the falling edge of CP.

VDAT[3:0]: OUTPUT

These pins are the data for the LCD. These signals are directly connected to the LCD for 4-bit non-split displays. For 4-bit split and 8-bit non-split displays, an external register is required to demultiplex the 4-bit data into the desired 8 parallel data lines needed for the LCD.

DISPON: OUTPUT

This pin is the display-on enable signal for the LCD.

VIDDONE: OUTPUT

This pin is used to externally synchronize events to periods when the video is not shifting.

2.2.13 Endian Pin

ENDIAN: INPUT

This pin is used to select the endianness. The high level input sets the endianness to the big endian. The low level input sets the endianness to the little endian.

2.2.14 Test Pins

TESTAIU: INPUT

If this pin is set to low during reset, ROM data bus width will be 16 bits wide.

If this pin is set to high during reset, ROM data bus width will be 32 bits wide.

When the DREQ* signal is asserted to request external arbitration, this pin must be set to low.

TESTCPU: INPUT

Connect to GND.

TESTIN: INPUT

Connect to GND.

2.2.15 Spare Pins

NC3-1: NO CONNECT

These pins are reserved for future use and should be left unconnected.

2.2.16 Power Supply Pins

VDD: +3.3 V

These pins are the power pins.
(In case of TMRP3912AU-92 and TMRP3912XB-92)

VSS: GND

These pins are the ground pins for the TMPR3911/12.

VDDH: +3.3 V

These pins are the power pins for I/O.
(In case of TMPR3911BU and TMPR3911BXB)

VDDL: +2.6 V

These pins are the power pins for internal logic.
(In case of TMPR3911BU and TMPR3911BXB)

VDD (PLL): +2.6 V

These pins are the power pins for PLL.
(In case of TMPR3911BU and TMPR3911BXB)

VSS (PLL): GND

These pins are the ground pins for PLL.
(In case of TMPR3911BU and TMPR3911BXB)

2.3 Pin Mode

This section contains tables summarizing various aspects of the pin usage for the TMPR3911/12. Table 2.3.1 lists the standard versus multi-function usage for each TMPR3911/12 pin, if applicable. The column showing the multi-function select indicates the corresponding bit in the MIOSEL register, as well as the default configuration of each multi-function pin during reset. The "Bus Arb State" column shows which pins are tri-stated whenever the DGRNT* signal is asserted in response to a DREQ* (external bus arbitration request).

Table 2.3.1 TMPR3911/12 Standard and Multi-Function Pin Mode (1/3)

TMPR3911/12 pin	standard function (I = input, O = output)	multi-function	multi-function select 1 = multi-function mode 0 = standard function mode		Bus Arb State
			MIOSEL	reset state	
D[31:0]	D[31:0] (I/O)	–	–	–	Hi-Z
A[12:0]	A[12:0] (I/O)	–	–	–	Hi-Z
ALE	ALE (O)	–	–	–	Hi-Z
RD*	RD* (O)	–	–	–	Hi-Z
WE*	WE* (O)	–	–	–	Hi-Z
CAS0* (WE0*)	CAS0* (O)	–	–	–	Hi-Z
CAS1* (WE1*)	CAS1* (O)	–	–	–	Hi-Z
CAS2* (WE2*)	CAS2* (O)	–	–	–	Hi-Z
CAS3* (WE3*)	CAS3* (O)	–	–	–	Hi-Z
RAS0*	RAS0* (O)	–	–	–	Hi-Z
RAS1* (DCS1)*	RAS1* (O)	–	–	–	Hi-Z
DCS0*	DCS0* (O)	–	–	–	Hi-Z
DCKE	DCKE (O)	–	–	–	Hi-Z
DCLKIN	DCLKIN (I)	–	–	–	–
DCLKOUT	DCLKOUT (O)	–	–	–	Hi-Z
DQMH	DQMH (O)	–	–	–	Hi-Z
DQML	DQML (O)	–	–	–	Hi-Z
DREQ*	DREQ* (I)	MIO[27]	MIOSEL[27]	0	–
DGRNT*	DGRNT* (O)	MIO[26]	MIOSEL[26]	0	–
SYSCLKIN	SYSCLKIN (I)	–	–	–	–
SYSCLKOUT	SYSCLKOUT (O)	–	–	–	–
C32KIN	C32KIN (I)	–	–	–	–
C32KOUT	C32KOUT (O)	–	–	–	–
BC32K	BC32K (O)	MIO[25]	MIOSEL[25]	1	–
VDAT[3]	VDAT[3] (O)	–	–	–	–
VDAT[2]	VDAT[2] (O)	–	–	–	–
VDAT[1]	VDAT[1] (O)	–	–	–	–
VDAT[0]	VDAT[0] (O)	–	–	–	–
CP	CP (O)	–	–	–	–
LOAD	LOAD (O)	–	–	–	–
DF	DF (O)	–	–	–	–
FRAME	FRAME (O)	–	–	–	–
DISPON	DISPON (O)	–	–	–	–
PWRCS	PWRCS (O)	–	–	–	–
PWRINT	PWRINT (I)	–	–	–	–

Table 2.3.2 TMPR3911/12 Standard and Multi-Function Pin Mode (2/3)

TMPR3911/12 pin	standard function (I = input, O = output)	multi-function	multi-function select 1 = multi-function mode 0 = standard function mode		Bus Arb State
			MIOSEL	reset state	
PWROK	PWROK (I)	–	–	–	–
ONBUTN	ONBUTN (I)	–	–	–	–
CPURES*	CPURES* (I)	–	–	–	–
PON*	PON* (I)	–	–	–	–
MBUSCLK	MBUSCLK (I/O)	–	–	–	–
MBUSDATA	MBUSDATA (I/O)	–	–	–	–
MBUSINT	MBUSINT (I)	–	–	–	–
TXD	TXD (O)	MIO[24]	MIOSEL[24]	0	–
RXD	RXD (I)	MIO[23]	MIOSEL[23]	0	–
CS0*	CS0* (O)	–	–		Hi-Z
CS1*	CS1* (O)	MIO[22]	MIOSEL[22]	0	–
CS2*	CS2* (O)	MIO[21]	MIOSEL[21]	0	–
CS3*	CS3* (O)	MIO[20]	MIOSEL[20]	0	
MCS0*	MCS0* (O)	MIO[19]	MIOSEL[19]	1	–
MCS1*	MCS1* (O)	MIO[18]	MIOSEL[18]	1	–
MCS2*	MCS2* (O)	MIO[17]	MIOSEL[17]	1	–
MCS3*	MCS3* (O)	MIO[16]	MIOSEL[16]	1	–
CHIFS	CHIFS (I/O)	MIO[31]	MIOSEL[31]	1	–
CHICLK	CHICLK (I/O)	MIO[30]	MIOSEL[30]	1	–
CHIDOUT	CHIDOUT (O)	MIO[29]	MIOSEL[29]	1	–
CHIDIN	CHIDIN (I)	MIO[28]	MIOSEL[28]	1	–
VCC3	VCC3 (I)	–	–	–	–
IO6	IO6 (I/O)	–	–	–	–
IO5	IO5 (I/O)	–	–	–	–
IO4	IO4 (I/O)	–	–	–	–
IO3	IO3 (I/O)	–	–	–	–
IO2	IO2 (I/O)	–	–	–	–
IO1	IO1 (I/O)	–	–	–	–
IO0	IO0 (I/O)	–	–	–	–
SPICLK	SPICLK (O)	MIO[15]	MIOSEL[15]	0	–
SPIOUT	SPIOUT (O)	MIO[14]	MIOSEL[14]	0	–
SPIIN	SPIIN (I)	MIO[13]	MIOSEL[13]	0	–
SIBSYNC	SIBSYNC (O)	–	–	–	–
SIBDOUT	SIBDOUT (O)	–	–	–	–
SIBDIN	SIBDIN (I)	–	–	–	–
SIBMCLK	SIBMCLK (I/O)	MIO[12]	MIOSEL[12]	0	–
SIBSCLK	SIBSCLK (O)	–	–	–	–
SIBIRQ	SIBIRQ (I)	–	–	–	–
RXPWR	RXPWR (O)	–	–	–	–
CARDET	CARDET (I)	–	–	–	–
IROUT	IROUT (O)	–	–	–	–
IRIN	IRIN (I)	–	–	–	–
TESTAIU	TESTAIU (I)	–	–	–	–
TESTCPU	TESTCPU (I)	–	–	–	–
TESTIN	TESTIN (I)	–	–	–	–
VIDDONE	VIDDONE (O)	–	–	–	–

Table 2.3.3 TMPR3911/12 Standard and Multi-Function Pin Mode (3/3)

TMPR3911/12 pin	standard function (I = input, O = output)	multi-function	multi-function select 1 = multi-function mode 0 = standard function mode		Bus Arb State
			MIOSEL	reset state	
CARDREG*	CARDREG* (O) (SHOWDINO CS*)	MIO[11]	MIOSEL[11]	1	–
CARDIOWR*	CARDIOWR* (O)	MIO[10]	MIOSEL[10]	1	–
CARDIORD*	CARDIORD* (O)	MIO[9]	MIOSEL[9]	1	–
CARD1CSL*	CARD1CSL* (O)	MIO[8]	MIOSEL[8]	1	–
CARD1CSH	CARD1CSH* (O)	MIO[7]	MIOSEL[7]	1	–
CARD2CSL*	CARD2CSL* (O)	MIO[6]	MIOSEL[6]	1	–
CARD2CSH*	CARD2CSH* (O)	MIO[5]	MIOSEL[5]	1	–
CARD1WAIT*	CARD1WAIT* (I)	MIO[4]	MIOSEL[4]	1	–
CARD2WAIT*	CARD2WAIT* (I)	MIO[3]	MIOSEL[3]	1	–
CARDDIR*	CARDDIR* (O)	MIO[2]	MIOSEL[2]	1	–
MFIO[1]	reserved	MIO[1]	MIOSEL[1]	1	–
MFIO[0]	reserved	MIO[0]	MIOSEL[0]	1	–
ENDIAN	ENDIAN (I)	–	–	–	–
NC[3:1]	SPARE	–	–	–	–
VDD	+3.3 V (*1)	–	–	–	–
VSS	GND	–	–	–	–
VDDH	+3.3 V (*2)	–	–	–	–
VDDL	+2.6 V (*2)	–	–	–	–

*1) In case of TMPR3912AU-92 and TMPR3912XB-92
*2) In case of TMPR3911BU and MTPR3911BXB

Table 2.3.4 lists various power-down states and conditions for each TMPR3911/12 pin. The “Power-Down Control” column shows the conditions which trigger a power-down for each respective pin. This column also shows the reset state for each of these conditions.

The “PON* state” column defines the state of each pin at power-on reset (PON*). This condition is defined as initial power up of the TMPR3911/12. This state is entered after power is applied for the very first time (VSTANDBY is turned on but VCC3 is still turned off). See figure 12.2.2.

The “1st-time power-up state” column defines the state of each pin after power-up. This mode is defined as VCC3 applied to the entire system and is initiated by the user pressing the ONBUTN while in the power-on reset (PON*) state. See figure 12.2.2.

The “power-down state” column defines the state of each pin during power-down mode.

Table 2.3.4 TMPR3911/12 Power-Down Pin Mode (1/3)

TMPR3911/12 pin	PON* state	1st time power-up state	power-down state	Power-Down Control
D[31:0]	LOW	LOW	LOW	MEMPOWERDOWN
A[12:0]	LOW	LOW	LOW	MEMPOWERDOWN
ALE	LOW	LOW	LOW	X
RD*	LOW	HI	LOW	POWERDOWN
WE*	LOW	LOW	LOW	MEMPOWERDOWN
CAS0* (WE0*)	LOW	LOW	LOW	MEMPOWERDOWN
CAS1* (WE1*)	LOW	LOW	LOW	MEMPOWERDOWN
CAS2* (WE2*)	LOW	LOW	LOW	MEMPOWERDOWN
CAS3* (WE3*)	LOW	LOW	LOW	MEMPOWERDOWN
RAS0*	LOW	LOW	LOW	MEMPOWERDOWN
RAS1* (DCS1*)	LOW	LOW	LOW	MEMPOWERDOWN
DCS0*	LOW	LOW	LOW	MEMPOWERDOWN
DCKE	LOW	LOW	LOW	MEMPOWERDOWN
DCLKIN*	IN	IN	IN	X
DCLKOUT	LOW	LOW	LOW	MEMPOWERDOWN
DQMH	LOW	LOW	LOW	MEMPOWERDOWN
DQML	LOW	LOW	LOW	MEMPOWERDOWN
DREQ	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[27]
DGRNT*	LOW	HI	SELECTABLE	POWERDOWN & MIOPD[26]
SYSCLKIN	OSC OFF	OSC ON	OSC OFF	POWERDOWN
SYSCLKOUT	OSC OFF	OSC ON	OSC OFF	POWERDOWN
C32KIN	OSC ON	OSC ON	OSC ON	X
C32KOUT	OSC ON	OSC ON	OSC ON	X
BC32K	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[25]
VDAT[3]	LOW	LOW	LOW	video module disabled
VDAT[2]	LOW	LOW	LOW	video module disabled
VDAT[1]	LOW	LOW	LOW	video module disabled
VDAT[0]	LOW	LOW	LOW	video module disabled

POWERDOWN = (VCCON & VCC3)*
VCCON: a bit in the Power Control Register.
VCC3: an input signal.
MEMPOWERDOWN: a bit in the Memory Configration 4 Register.
SELECTABLE: See Table 9.2.2 and 9.2.3.

Table 2.3.5 TMPR3911/12 Power-Down Pin Mode (2/3)

TMPR3911/12 pin	PON* state	1st time power-up state	power-down state	Power-Down Control
CP	LOW	LOW	LOW	video module disabled
LOAD	LOW	LOW	LOW	video module disabled
DF	LOW	LOW	LOW	video module disabled
FRAME	LOW	LOW	LOW	video module disabled
DISPON	LOW	LOW	LOW	video module disabled
PWRCS	LOW	HI	LOW	X
PWRINT	IN	IN	IN	X
PWROK	IN	IN	IN	X
ONBUTN	IN	IN	IN	X
CPURES*	IN	IN	IN	X
PON*	IN	IN	IN	X
MBUSCLK	LOW	LOW	LOW	magicbus module disabled
MBUSDATA	LOW	LOW	LOW	magicbus module disabled
MBUSINT	IN	IN	IN	X
TXD	LOW	LOW	SELECTABLE	POWERDOWN & MIOPD[24]
RXD	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[23]
CS0*	PULL-DOWN	HI	PULL-DOWN	POWERDOWN
CS1*	PULL-DOWN	HI	SELECTABLE	POWERDOWN & MIOPD[22]
CS2*	PULL-DOWN	HI	SELECTABLE	POWERDOWN & MIOPD[21]
CS3*	PULL-DOWN	HI	SELECTABLE	POWERDOWN & MIOPD[20]
MCS0*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[19]
MCS1*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[18]
MCS2*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[17]
MCS3*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[16]
CHIFS	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[31]
CHICLK	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[30]
CHIDOUT	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[29]
CHIDIN	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[28]
VCC3	PULL-DOWN	IN	PULL-DOWN	POWERDOWN
IO6	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[6]
IO5	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[5]
IO4	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[4]
IO3	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[3]
IO2	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[2]
IO1	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[1]
IO0	PULL-DOWN	IN	SELECTABLE	POWERDOWN & IOPD[0]
SPICLK	LOW	LOW	SELECTABLE	POWERDOWN & MIOPD[15]
SPIOUT	LOW	LOW	SELECTABLE	POWERDOWN & MIOPD[14]
SPIIN	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[13]
SIBYNC	LOW	LOW	LOW	POWERDOWN
SIBDOUT	LOW	LOW	LOW	POWERDOWN
SIBDIN	PULL-DOWN	IN	PULL-DOWN	POWERDOWN
SIBMCLK	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[12]
SIBSCLK	LOW	LOW	LOW	POWERDOWN
SIBIRQ	PULL-DOWN	IN	PULL-DOWN	POWERDOWN

POWERDOWN = (VCCON & VCC3)*
VCCON: a bit in the Power Control Register.
VCC3: an input signal.
MEMPOWERDOWN: a bit in the Memory Configration 4 Register.
SELECTABLE: See Table 9.2.2 and 9.2.3.

Table 2.3.6 TMPR3911/12 Power-Down Pin Mode (3/3)

TMPR3911/12 pin	PON* state	1st time power-up state	power-down state	Power-Down Control
RXPWR	LOW	LOW	LOW	POWERDOWN
CARDET	PULL-DOWN	IN	PULL-DOWN	POWERDOWN
IROUT	LOW	LOW	LOW	POWERDOWN
IRIN	PULL-DOWN	IN	PULL-DOWN	POWERDOWN
TESTAIU	IN	IN	IN	X
TESTCPU	IN	IN	IN	X
TESTIN	IN	IN	IN	X
VIDDONE	LOW	LOW	LOW	X
CARDREG*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[11]
CARDIOWR*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[10]
CARDIORD*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[9]
CARD1CSL*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[8]
CARD1CSH*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[7]
CARD2CSL*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[6]
CARD2CSH*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[5]
CARD1WAIT*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[4]
CARD2WAIT*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[3]
CARDDIR*	PULL-DOWN	IN	SELECTABLE	POWERDOWN & MIOPD[2]
MFIO[1]	IN	IN	SELECTABLE	POWERDOWN & MIOPD[1]
MFIO[0]	IN	IN	SELECTABLE	POWERDOWN & MIOPD[0]
ENDIAN	IN	IN	IN	X
NC[3:1]	X	X	X	X
VDD 19EACH/34EACH	X	X	X	X
VSS 17EACH/34EACH	X	X	X	X

POWERDOWN = (VCCON & VCC3)*
VCCON: a bit in the Power Control Register.
VCC3: an input signal.
MEMPOWERDOWN: a bit in the Memory Configration 4 Register.
SELECTABLE: See Table 9.2.2 and 9.2.3.

2.4 Pin Assignment

2.4.1 Pin assignment (in case of TMPR3912AU-92 and TMPR3912XB-92)

Table 2.4.1 Pin Assignment (1/3)

LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name	LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name
1	A1	—	VDD	45	M2	I/O	MIOX[0]
2	C3	I/O	D[0] (D[24])	46	M3	I/O	IO[6]
3	E5	—	VSS	47	L4	I/O	IO[5]
4	B1	I/O	D[1] (D[25])	48	N1	—	VSS
5	C2	I/O	D[2] (D[26])	49	N2	I/O	CHICLK
6	E4	—	VDD	50	M4	I/O	CHIFS
7	C1	I/O	D[3] (D[27])	51	P2	I	CHIDIN
8	D3	—	VSS	52	P1	O	CHIDOUT
9	D2	I/O	D[4] (D[28])	53	R1	—	VDD
10	F5	—	VDD	54	N3	I	RXD
11	D1	I/O	D[5] (D[29])	55	L5	O	TXD
12	E3	I/O	D[6] (D[30])	56	R2	I/O	IO[4]
13	E2	—	VSS	57		—	NC
14	F4	I/O	D[7] (D[31])	58	M5	I	IRIN
15	E1	—	VSS	59	R3	O	IROUT
16	F3	I/O	D[8] (D[16])	60	N4	—	VSS
17	F2	—	VDD	61	P4	—	VDD
18	G5	I/O	D[9] (D[17])	62	L6	I	CARDET
19	F1	I/O	D[10] (D[18])	63	R4	O	RXPWR
20	G4	—	VSS	64	N5	I/O	IO[3]
21	G3	I/O	D[11] (D[19])	65	P5	I/O	IO[2]
22	G2	—	VDD	66	M6	—	VSS
23	G1	I/O	D[12] (D[20])	67	R5	O	SPICLK
24	G6	I/O	D[13] (D[21])	68	N6	I	SPIIN
25	H6	—	VSS	69	P6	O	SPIOUT
26	H3	I/O	D[14] (D[22])	70	L7	—	VDD
27	H1	I/O	D[15] (D[23])	71	R6	I	TESTCPU
28	H2	—	VDD	72	M7	I	TESTIN
29	H4	I	ENDIAN	73	N7	O	VIDDONE
30	J6	I/O	MIOX[1]	74	P7	I	TESTAIU
31	J1	I	MBUSINT	75	R7	—	VSS
32	J2	I/O	MBUSDATA	76	K7	I	VCC3
33	J3	—	VSS	77	L8	O	BC32K
34	J4	I/O	MBUSCLK	78	N8	—	VDD
35	J5	—	VDD	79	R8	I	C32KIN
36	K1	—	VDD	80	P8	O	C32KOUT
37	K2	O	SIBMCLK	81	M8	—	VSS
38	K3	—	VSS	82	K9	O	PWRCS
39	K4	O	SIBSCLK	83	R9	I	PWRINT
40	L1	O	SIBSYNC	84	P9	I	PWROK
41	L2	I	SIBDIN	85		—	NC
42	L3	O	SIBDOUT	86	M9	I	ONBUTN
43	K5	—	VDD	87	L9	I	PON*
44	M1	I	SIBIRQ	88	R10	I	CPURES*

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

(3) The signal name change regarding DQMH/L is applied only to TMPR3912AU-92.

Table 2.4.2 Pin Assignment (2/3)

LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name	LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name
89	P10	—	VDD	137	G13	—	VDD
90	N10	O	DISPON	138	(G12)	I/O	D[28] (D[4])
91	M10	O	FRAME	139	(G11)	I/O	D[27] (D[3])
92	R11	—	VSS	140	F15	—	VSS
93	P11	O	DF	141	F14	I/O	D[26] (D[2])
94	N11	O	LOAD	142	F13	—	VSS
95	L10	O	CP	143	F12	I/O	D[25] (D[1])
96	R12	—	VSS	144	E15	—	VDD
97	P12	—	VDD	145	E14	I/O	D[24] (D[0])
98	N12	O	VDAT[0]	146	E13	I/O	D[23] (D[15])
99	M11	O	VDAT[1]	147	F11	—	VDD
100	R13	O	VDAT[2]	148	D15	I/O	D[22] (D[14])
101	P13	O	VDAT[3]	149	D14	—	VSS
102	M12	—	VSS	150	D13	I/O	D[21] (D[13])
103	P14	I/O	IO[1]	151	E12	—	VDD
104	R14	—	VDD	152	C15	I/O	D[20] (D[12])
105	R15	I	CARD2WAIT*	153	C14	I/O	D[19] (D[11])
106	N13	O	CARD2CSH*	154	D12	—	VSS
107	L11	O	CARD2CSL*	155	B14	I/O	D[18] (D[10])
108	P15	I/O	IO[0]	156	B15	—	VDD
109	N14	—	VSS (PLL)	157	A15	I/O	D[17] (D[9])
110	L12	O	CARDIORD*	158	C13	—	VSS
111	N15	O	CARDIOWR*	159	E11	I/O	D[16] (D[8])
112	M13	O	CARDREG*	160	A14	—	VDD
113	M14	I	CARD1WAIT*	161		—	NC
114	K11	—	VDD (PLL)	162	D11	O	CS0*
115	M15	O	CARDDIR*	163	A13	O	RD*
116	L13	—	VDD	164	C12	—	VSS
117	L14	O	CARD1CSL*	165	B12	—	VDD
118	K12	O	CARD1CSH*	166	E10	O	DGRNT*
119	L15	—	VSS	167	A12	I	DREQ*
120	K13	O	MCS3*	168	C11	O	ALE
121	K14	O	MCS2*	169	B11	O	WE*
122	J11	O	MCS1*	170	D10	—	VDD
123	K15	O	MCS0*	171	A11	I/O	A[12]
124	J12	O	CS3*	172	C10	I/O	A[11]
125	J13	O	CS2*	173	B10	—	VSS
126	J14	O	CS1*	174	E9	I/O	A[10]
127	J15	—	VDD	175	A10	I/O	A[9]
128	J10	I	SYSCLKIN	176	D9	—	VDD
129	H11	O	SYSCLKOUT	177	C9	I/O	A[8]
130	H13	—	VSS	178	B9	I/O	A[7]
131	H15	—	VSS	179	A9	—	VSS
132	H14	—	VDD	180	F9	I/O	A[6]
133	H12	I/O	D[31] (D[7])	181	E8	I/O	A[5]
134	G10	I/O	D[30] (D[6])	182	C8	—	VDD
135	G15	—	VSS	183	A8	I/O	A[4]
136	G14	I/O	D[29] (D[5])	184	B8	—	VSS

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

(3) The signal name change regarding DQMH/L is applied only to TMPR3912AU-92.

Table 2.4.3 Pin Assignment (3/3)

LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name	LQFP NO. (TMPR3912)	FBGA NO.	I/O	Signal Name
185	D8	I/O	A[3]	203	D5	—	VSS
186	F7	I/O	A[2]	204	A3	I	DCLKIN
187	A7	—	VDD	205	B3	O	DCLKOUT
188	B7	I/O	A[1]	206	D4	—	VDD
189	C7	I/O	A[0]	207	B2	O	DQMH (DQML)
190	D7	—	VSS	208	A2	O	DQML (DQMH)
191	E7	—	VSS	—	B13	—	NC
192	A6	O	DCS0*	—	F6	—	NC
193	B6	O	RAS1*	—	F8	—	NC
194	C6	O	RAS0*	—	F10	—	NC
195	D6	O	CAS3* (CAS0*)	—	G7	—	NC
196	A5	—	VDD	—	H5	—	NC
197	B5	O	CAS2* (CAS1*)	—	H10	—	NC
198	C5	O	CAS1* (CAS2*)	—	K6	—	NC
199	E6	O	CAS0* (CAS3*)	—	K8	—	NC
200	A4	—	VSS	—	K10	—	NC
201	B4	—	VDD	—	N9	—	NC
202	C4	O	DCKE	—	P3	—	NC

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

(3) The signal name change regarding DQMH/L is applied only to TMPR3912AU-92.

2.4.2 Pin Assignment (in case of TMPR3911BU)

Table 2.4.4 Pin Assignment (1/2)

NO.	I/O	Signal Name	NO.	I/O	Signal Name	NO.	I/O	Signal Name
1	—	VDDH	41	I/O	CHICLK	81	—	VSS
2	I/O	D[0] (D [24])	42	I/O	CHIFS	82	O	VDAT[0]
3	I/O	D[1] (D [25])	43	I	CHIDIN	83	O	VDAT[1]
4	I/O	D[2] (D [26])	44	O	CHIDOUT	84	O	VDAT[2]
5	I/O	D[3] (D [27])	45	I	RXD	85	O	VDAT[3]
6	—	VSS	46	O	TXD	86	—	VSS
7	I/O	D[4] (D [28])	47	I/O	IO[4]	87	I/O	IO[1]
8	—	VDDL	48	I	IRIN	88	—	VDDH
9	I/O	D[5] (D [29])	49	O	IROUT	89	I	CARD2WAIT*
10	I/O	D[6] (D [30])	50	—	VSS	90	O	CARD2CSH*
11	I/O	D[7] (D [31])	51	—	VDDL	91	O	CARD2CSL*
12	I/O	D[8] (D [16])	52	I	CARDET	92	I/O	IO[0]
13	I/O	D[9] (D [17])	53	O	RXPWR	93	—	VSS (PLL)
14	I/O	D[10] (D [18])	54	I/O	IO[3]	94	O	CARDIORD*
15	—	VSS	55	I/O	IO[2]	95	O	CARDIOWR*
16	I/O	D[11] (D [19])	56	O	SPICLK	96	O	CARDREG*
17	I/O	D[12] (D [20])	57	I	SPIIN	97	I	CARD1WAIT*
18	I/O	D[13] (D [21])	58	O	SPIOUT	98	—	VDD (PLL)
19	I/O	D[14] (D [22])	59	I	TESTCPU	99	O	CARDDIR*
20	I/O	D[15] (D [23])	60	I	TESTIN	100	O	CARD1CSL*
21		VDDH	61	O	VIDDONE	101	O	CARD1CSH*
22	I	ENDIAN	62	I	TESTAIU	102	O	MCS3*
23	I/O	MIOX[1]	63	I	VCC3	103	O	MCS2*
24	I	MBUSINT	64	O	BC32K	104	O	MCS1*
25	I/O	MBUSDATA	65	—	VDDH	105	O	MCS0*
26	—	VSS	66	I	C32KIN	106	O	CS3*
27	I/O	MBUSCLK	67	O	C32KOUT	107	O	CS2*
28	—	VDDH	68	—	VSS	108	O	CS1*
29	—	VDDL	69	O	PWRCS	109	—	VDDH
30	O	SIBMCLK	70	I	PWRINT	110	I	SYSCLKIN
31	—	VSS	71	I	PWROK	111	O	SYSCLKOUT
32	O	SIBSCLK	72	I	ONBUTN	112	—	VSS
33	O	SIBSYNC	73	I	PON*	113	—	VDDL
34	I	SIBDIN	74	I	CPURES*	114	I/O	D[31] (D [7])
35	O	SIBDOUT	75	—	VDDL	115	I/O	D[30] (D [6])
36	—	NC	76	O	DISPON	116	I/O	D[29] (D [5])
37	I	SIBIRQ	77	O	FRAME	117	I/O	D[28] (D [4])
38	I/O	MIOX[0]	78	O	DF	118	I/O	D[27] (D [3])
39	I/O	IO[6]	79	O	LOAD	119	—	VSS
40	I/O	IO[5]	80	O	CP	120	I/O	D[26] (D [2])

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

Table 2.4.5 Pin Assignment (2/2)

NO.	I/O	Signal Name	NO.	I/O	Signal Name	NO.	I/O	Signal Name
121	I/O	D[25] (D [1])	161	I/O	A[0]			
122	—	VDDH	162	—	VSS			
123	I/O	D[24] (D [0])	163	O	DCS0*			
124	I/O	D[23] (D [15])	164	O	RAS1*			
125	I/O	D[22] (D [14])	165	O	RAS0*			
126	—	VSS	166	O	CAS3* (CAS0*)			
127	I/O	D[21] (D [13])	167	O	CAS2* (CAS1*)			
128	I/O	D[20] (D [12])	168	O	CAS1* (CAS2*)			
129	I/O	D[19] (D [11])	169	O	CAS0* (CAS3*)			
130	—	VSS	170	—	VDDH			
131	I/O	D[18] (D [10])	171	O	DCKE			
132	—	VDDL	172	I	DCLKIN			
133	I/O	D[17] (D [9])	173	O	DCLKOUT			
134	—	VSS	174	—	NC			
135	—	NC	175	O	DQMH(DQML)			
136	I/O	D[16] (D [8])	176	O	DQML(DQMH)			
137	—	VDDH						
138	O	CS0*						
139	O	RD*						
140	—	VSS						
141	—	VDDL						
142	O	DGRNT*						
143	I	DREQ*						
144	O	ALE						
145	O	WE*						
146	I/O	A[12]						
147	I/O	A[11]						
148	I/O	A[10]						
149	I/O	A[9]						
150	—	VDDH						
151	I/O	A[8]						
152	I/O	A[7]						
153	—	VSS						
154	I/O	A[6]						
155	I/O	A[5]						
156	I/O	A[4]						
157	I/O	A[3]						
158	I/O	A[2]						
159	—	VDDL						
160	I/O	A[1]						

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

2.4.3 Pin Assignment (in case of TMPR3911BXB)

Table 2.4.6 Pin Assignment (1/2)

NO.	I/O	Signal Name	NO.	I/O	Signal Name	NO.	I/O	Signal Name
A1	I/O	D[1] (D[25])	L3	I	SIBDIN	D7	I/O	A[3]
B1	—	VDDH	M3	—	NC	M7	I	VCC3
C1	I/O	D[0] (D[24])	N3	I/O	IO[6]	N7	O	BC32K
D1	I/O	D[3] (D[27])	P3	I	IRIN	P7	I	C32KIN
E1	I/O	D[5] (D[29])	R3	O	TXD	R7	O	VIDDONE
F1	I/O	D[9] (D[17])	A4	O	DCLKOUT	A8	—	VSS
G1	I/O	D[12] (D[20])	B4	—	VDDH	B8	I/O	A[4]
H1	—	VDDH	C4	O	CAS1* (CAS2*)	C8	I/O	A[6]
J1	—	VDDH	D4	—	VSS	D8	I/O	A[5]
K1	O	SIBSYNC	E4	—	VDDL	M8	O	C32KOUT
L1	I	SIBIRQ	F4	I/O	D[10] (D[18])	N8	I	TESTAIU
M1	I/O	IO[5]	G4	I/O	D[14] (D[22])	P8	—	VSS
N1	I	CHIDIN	H4	I/O	MIOX[1]	R8	—	VDDH
P1	O	CHIDOUT	J4	I/O	MBUSDATA	A9	I/O	A[9]
R1	I/O	IO[4]	K4	—	VDDL	B9	I/O	A[7]
A2	O	DQML (DQMH)	L4	O	SIBSCLK	C9	I/O	A[8]
B2	—	NC	M4	—	VSS	D9	—	VDDH
C2	I/O	D[2] (D[26])	N4	I	CARDET	M9	O	PWRCS
D2	I/O	D[6] (D[30])	P4	I/O	IO[3]	N9	I	PWROK
E2	I/O	D[8] (D[16])	R4	O	IROUT	P9	I	PWRINT
F2	I/O	D[11] (D[19])	A5	O	CAS0* (CAS3*)	R9	I	ONBUTN
G2	I	ENDIAN	B5	O	CAS2* (CAS1*)	A10	O	WE*
H2	I	MBUSINT	C5	O	CAS3* (CAS0*)	B10	I/O	A[10]
J2	—	VSS	D5	O	RAS1*	C10	I/O	A[11]
K2	—	VSS	E5	—	NC	D10	I/O	A[12]
L2	O	SIBDOUT	M5	—	VDDL	M10	—	VDDL
M2	I/O	MIOX[0]	N5	I/O	IO[2]	N10	I	PON*
N2	I/O	CHICLK	P5	O	SPICLK	P10	I	CPURES*
P2	I/O	CHIFS	R5	O	RXPWR	R10	O	DISPON
R2	I	RXD	A6	O	RAS0*	A11	—	VDDL
A3	O	DQMH (DQML)	B6	O	DCS0*	B11	O	ALE
B3	I	DCLKIN	C6	—	VSS	C11	I	DREQ*
C3	O	DCKE	D6	I/O	A[0]	D11	O	DGRNT*
D3	I/O	D[4] (D[28])	M6	O	SPIOUT	M11	O	FRAME
E3	I/O	D[7] (D[31])	N6	I	TESTCPU	N11	O	LOAD
F3	—	VSS	P6	I	TESTIN	P11	O	DF
G3	I/O	D[15] (D[23])	R6	I	SPIIN	R11	—	VSS
H3	I/O	D[13] (D[21])	A7	I/O	A[1]	A12	—	VDDH
J3	I/O	MBUSCLK	B7	I/O	A[2]	B12	—	VSS
K3	O	SIBMCLK	C7	—	VDDL	C12	O	RD*

<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

Table 2.4.7 Pin Assignment (2/2)

NO.	I/O	Signal Name	NO.	I/O	Signal Name	NO.	I/O	Signal Name
D12	O	CS0*	P14	—	VSS			
E12	I/O	D[26] (D[2])	R14	—	VDDH			
F12	I/O	D[29] (D[5])	A15	—	NC			
G12	—	VDDL	B15	—	VDDL			
H12	O	SYSCLKOUT	C15	I/O	D[18] (D[10])			
J12	O	CS2*	D15	I/O	D[19] (D[11])			
K12	O	MCS3*	E15	I/O	D[22] (D[14])			
L12	—	VDD (PLL)	F15	I/O	D[25] (D[1])			
M12	O	CARDIORD*	G15	I/O	D[28] (D[4])			
N12	O	CP	H15	—	VDDH			
P12	O	VDAT[0]	J15	O	MCS0*			
R12	O	VDAT[3]	K15	O	CARD1CSH*			
A13	—	VSS	L15	I	CARD1WAIT*			
B13	I/O	D[16] (D[8])	M15	—	VSS (PLL)			
C13	I/O	D[21] (D[13])	N15	O	CARD2CSH*			
D13	I/O	D[23] (D[15])	P15	I	CARD2WAIT*			
E13	—	VDDH	R15	O	CARD2CSL*			
F13	I/O	D[27] (D[3])						
G13	I/O	D[30] (D[6])						
H13	O	CS3*						
J13	O	CS1*						
K13	O	MCS2*						
L13	O	CARDDIR*						
M13	O	CARDIOWR*						
N13	O	VDAT[1]						
P13	O	VDAT[2]						
R13	I/O	IO[1]						
A14	I/O	D[17] (D[9])						
B14	—	VSS						
C14	I/O	D[20] (D[12])						
D14	—	VSS						
E14	I/O	D[24] (D[0])						
F14	—	VSS						
G14	I/O	D[31] (D[7])						
H14	—	VSS						
J14	I	SYSCLKIN						
K14	O	MCS1*						
L14	O	CARD1CSL*						
M14	O	CARDREG*						
N14	I/O	IO[0]						

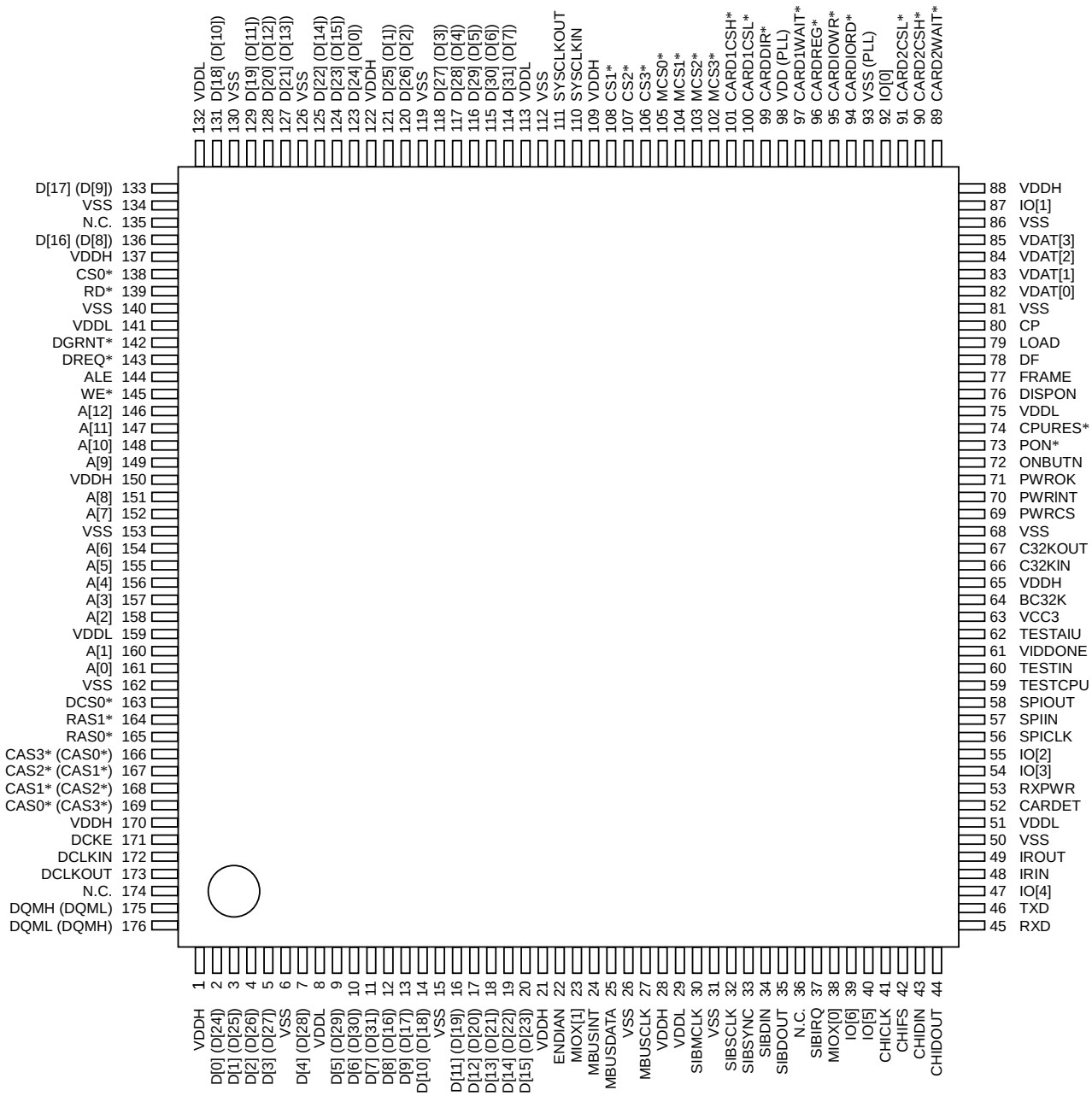
<NOTE>

(1) * Active-low signal

(2) ( ) indicates the signal name in the little endian mode

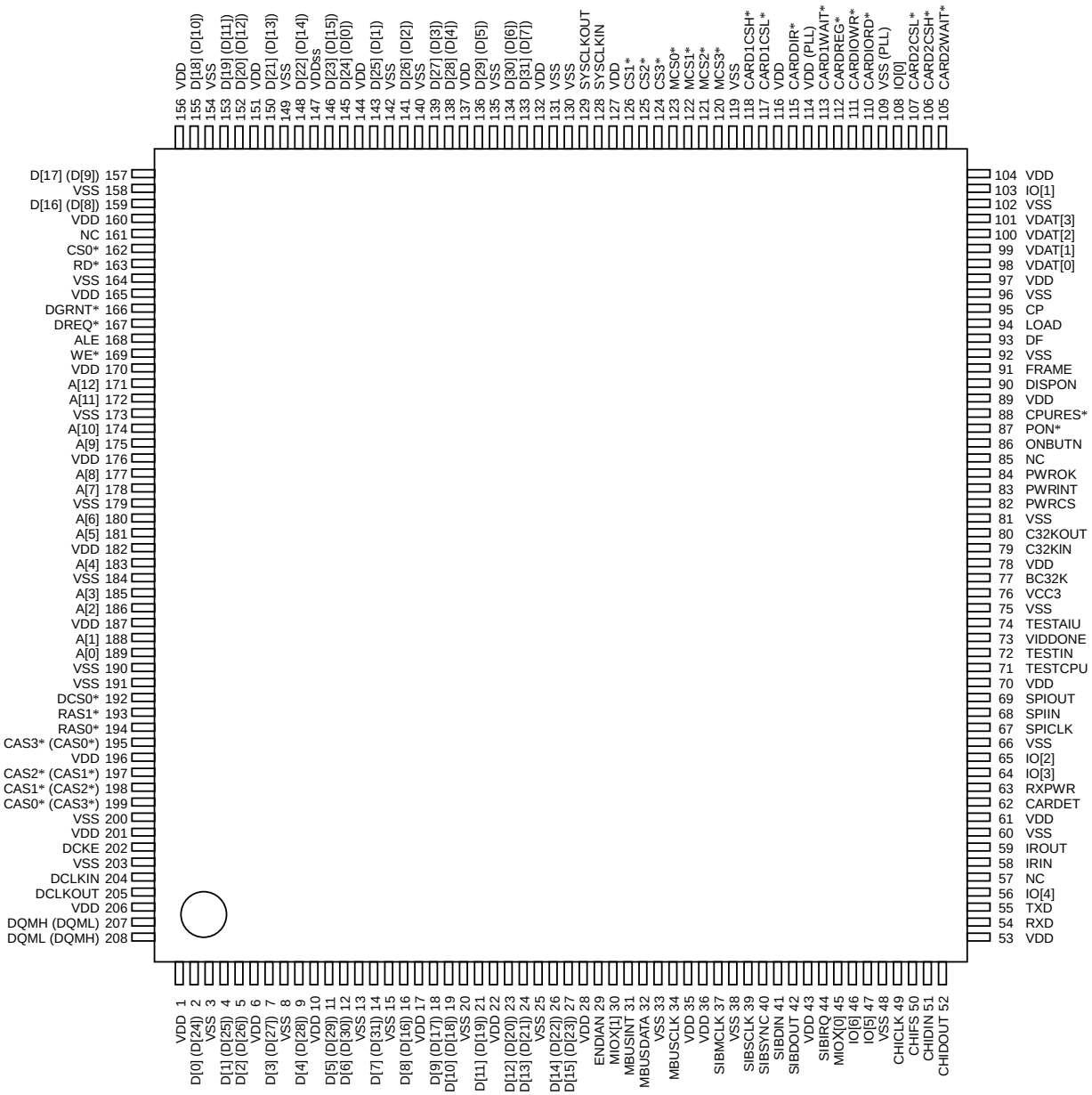
2.4.4 TMPR3911BU Pin Assignment

P-LQFP176-2424-0.50A



2.4.5 TMPR3912AU-92

P-LQFP208-2828-0.50A



2.4.6 TMPR3911BXB Ball Assignment

P-FBGA177-1313-0.80C4

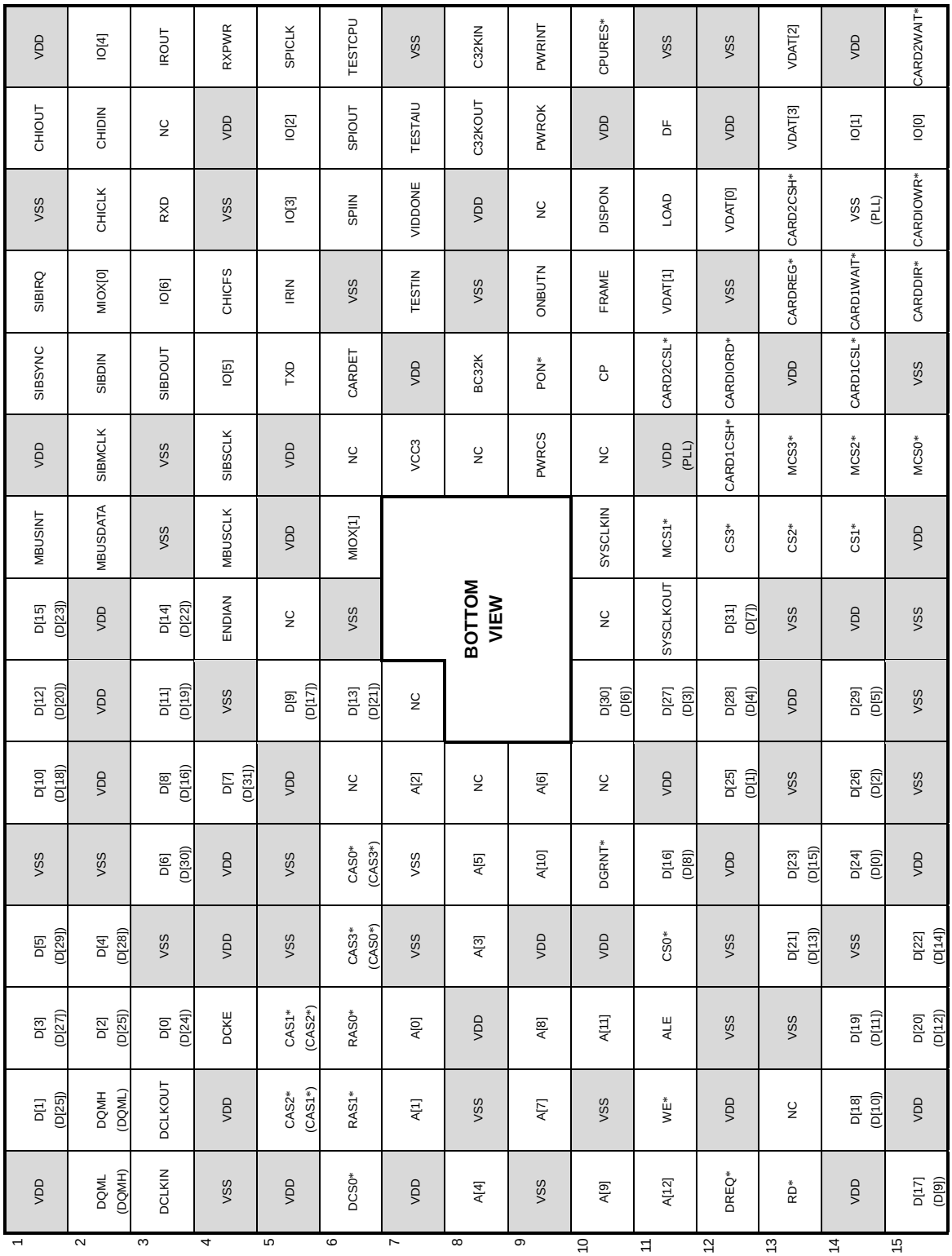
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R
1	3 D[1] (D[25])	1 VDDH	2 D[0] (D[24])	5 D[3] (D[27])	9 D[5] (D[29])	13 D[9] (D[17])	17 D[12] (D[20])	21 VDDH	28 VDDH	33 SIBSYNC	37 SIBIRQ*	40 IO[5]	43 CHIDIN	44 CHIDOUT	47 IO[4]
2	176 DQML (DQMH)	174 N.C.	4 D[2] (D[26])	10 D[6] (D[30])	12 D[8] (D[16])	16 D[11] (D[19])	22 ENDIAN	24 MBUSINT	26 VSS	31 VSS	35 SIBDOUT	38 MIOX[0]	41 CHICLK	42 CHIFS	45 RXD
3	175 DQMH (DQML)	172 DCLKIN	171 DCKE	7 D[4] (D[28])	11 D[7] (D[31])	15 VSS	20 D[15] (D[23])	18 D[13] (D[21])	27 MBUSCLK	30 SIBMCLK	34 SIBDIN	36 N.C.	39 IO[6]	48 IRIN	46 TXD
4	173 DCLKOUT	170 VDDH	168 CAS1* (CAS2*)	6 VSS	8 VDDL	14 D[10] (D[18])	19 D[14] (D[22])	23 MIOX[1]	25 MBUSDATA	29 VDDL	32 SIBSCLK	50 VSS	52 CARDET	54 IO[3]	49 IROUT
5	169 CAS0* (CAS3*)	167 CAS2* (CAS1*)	166 CAS3* (CAS0*)	164 RAS1*	N.C.							51 VDDL	55 IO[2]	56 SPICLK	53 RXPWR
6	165 RAS0*	163 DCS0*	162 VSS	161 A[0]								58 SPIOUT	59 TESTCPU	60 TESTIN	57 SPIIN
7	160 A[1]	158 A[2]	159 VDDL	157 A[3]								63 VCC3	64 BC32K	66 C32KIN	61 VIDDONE
8	153 VSS	156 A[4]	154 A[6]	155 A[5]				Bottom View				67 C32KOUT	62 TESTAIU	68 VSS	65 VDDH
9	149 A[9]	152 A[7]	151 A[8]	150 VDDH								69 PWRCS	71 PWROK	70 PWRINT	72 ONBUTN
10	145 WE*	148 A[10]	147 A[11]	146 A[12]								75 VDDL	73 PON*	74 CPURES*	76 DISPON
11	141 VDDL	144 ALE	143 DREQ*	142 DGRNT*								77 FRAME	79 LOAD	78 DF	81 VSS
12	137 VDDH	140 VSS	139 RD*	138 CS0*	120 D[26] (D[2])	116 D[25] (D[5])	113 VDDL	111 SYSCLKOUT	107 CS2*	102 MCS3*	98 VDD-PLL	94 CARDIORD*	80 CP	82 VDAT[0]	85 VDAT[3]
13	134 VSS	136 D[16] (D[8])	127 D[21] (D[13])	124 D[23] (D[15])	122 VDDH	118 D[27] (D[3])	115 D[30] (D[6])	106 CS3*	108 CS1*	103 MCS2*	99 CARDDIR*	95 CARDIOWR*	83 VDAT[1]	84 VDAT[2]	87 IO[1]
14	133 D[17] (D[9])	130 VSS	128 D[20] (D[12])	126 VSS	123 D[24] (D[0])	119 VSS	114 D[31] (D[7])	112 VSS	110 SYSCLKIN	104 MCS1*	100 CARD1CSL*	96 CARDREG*	92 IO[0]	86 VSS	88 VDDH
15	135 N.C.	132 VDDL	131 D[18] (D[10])	129 D[19] (D[11])	125 D[22] (D[14])	121 D[25] (D[1])	117 D[28] (D[4])	109 VDDH	105 MCS0*	101 CARD1CSH*	97 CARD1WAIT*	93 VSS-PLL	90 CARD2CSH*	89 CARD2WAIT*	91 CARD2CSL*

YY <- Pin No. of QFP
AAAA <- Singal Name on Big Endien Mode
(BBBB) <- Signal Name on Little Endien Mode

( ) should be used in Little-Endien mode. VDDH: 3.3 V/VDDL: 2.6 V

2.4.7 TMPR3912XB-92

P-FBGA217-1313-0.80B6



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	VDD	DQML (DQMH)	DCLKIN	VSS	VDD	DCS0*	VDD	A[4]	VSS	A[9]	A[12]	DREQ*	RD*	VDD	D[17] (D[9])
B	D[1] (D[25])	DQMH (DQML)	DCLKOUT	VDD	CAS2* (CAS1*)	RAS1*	A[1]	VSS	A[7]	VSS	WE*	VDD	NC	D[18] (D[10])	VDD
C	D[3] (D[27])	D[2] (D[25])	D[0] (D[24])	DCKE	CAS1* (CAS2*)	RAS0*	A[0]	VDD	A[8]	A[11]	ALE	VSS	VSS	D[19] (D[11])	D[20] (D[12])
D	D[5] (D[29])	D[4] (D[28])	VSS	VDD	VSS	CAS3* (CAS0*)	VSS	A[3]	VDD	VDD	CS0*	VSS	D[21] (D[13])	VSS	D[22] (D[14])
E	VSS	VSS	D[6] (D[30])	VDD	VSS	CAS0* (CAS3*)	VSS	A[5]	A[10]	DGRNT*	D[16] (D[8])	VDD	D[23] (D[15])	D[24] (D[0])	VDD
F	D[10] (D[18])	VDD	D[8] (D[16])	D[7] (D[31])	VDD	NC	A[2]	NC	A[6]	NC	VDD	D[25] (D[1])	VSS	D[26] (D[2])	VSS
G	D[12] (D[20])	VDD	D[11] (D[19])	VSS	D[9] (D[17])	D[13] (D[21])	NC			D[30] (D[6])	D[27] (D[3])	D[28] (D[4])	VDD	D[29] (D[5])	VSS
H	D[15] (D[23])	VDD	D[14] (D[22])	ENDIAN	NC	VSS				NC	SYSCLKOUT	D[31] (D[7])	VSS	VDD	VSS
J	MBUSINT	MBUSDATA	VSS	MBUSCLK	VDD	MIOX[1]				SYSCLKIN	MCS1*	CS3*	CS2*	CS1*	VDD
K	VDD	SIBMCLK	VSS	SIBSCLK	VDD	NC	VCC3	NC	PWRCS	NC	VDD (PLL)	CARD1CSH*	MCS3*	MCS2*	MCS0*
L	SIBSYNC	SIBDIN	SIBDOUT	IO[5]	TXD	CARDET	VDD	BC32K	PON*	CP	CARD2CSL*	CARDIORD*	VDD	CARD1CSL*	VSS
M	SIBIRQ	MIOX[0]	IO[6]	CHICFS	IRIN	VSS	TESTIN	VSS	ONBUTN	FRAME	VDAT[1]	VSS	CARDREG*	CARD1WAIT*	CARDDIR*
N	VSS	CHICLK	RXD	VSS	IO[3]	SPIIN	VIDDONE	VDD	NC	DISPON	LOAD	VDAT[0]	CARD2CSH*	VSS (PLL)	CARDIOWR*
P	CHIOUT	CHIDIN	NC	VDD	IO[2]	SPIOUT	TESTAIU	C32KOUT	PWROK	VDD	DF	VDD	VDAT[3]	IO[1]	IO[0]
R	VDD	IO[4]	IROUT	RXPWR	SPICLK	TESTCPU	VSS	C32KIN	PWRINT	CPURES*	VSS	VSS	VDAT[2]	VDD	CARD2WAIT*

( ) should be used in Little-Endien mode.

3. CPU Module

3.1 Overview

The TMPR3911/12 consists of an embedded TX39/H Processor core along with the System Interface Logic required to support the PDA System. These include a 4 Kbyte instruction cache and a 1 Kbyte data cache used to improve performance, a 32 × 32 multiplier with accumulator used to perform DSP functions to support features such as software modems.

Directly interfacing to the CPU core is the CPU Interface logic. The CPU Interface logic consists of a 4-deep Write Buffer used to speed up write transactions, DMA Arbitration logic used to arbitrate with the CPU core for SIU DMA requests and also used to provide cache coherency snooping during DMA operations, and Interface Controller logic used to provide the necessary control to the CPU core and to provide a simple interface for initiating memory transactions to the BIU.

3.2 CPU Core

3.2.1 Description

There are several enhancements for the PDA system. These additional key features of the CPU core are:

- Translation Look-aside Buffer (TLB) (4 Kbyte Page size, 32 Entries)
- 4 Kbyte instruction cache (I-cache)
  - 16 bytes (4 words) per line (256 lines total)
  - physical address tag per cache line
  - single valid bit per cache line
  - programmable burst size (16 bytes to 128 bytes)
  - instruction streaming mode supported
  - direct-mapped
- 1 Kbyte data cache (D-cache)
  - 4 bytes (1 word) per line (128 lines total)
  - physical address tag per cache line
  - single valid bit per cache line
  - programmable burst size (16 bytes to 128 bytes); burst refill can be disabled
  - write-through
  - two-way set associative
- cache address snoop mode supported for DMA
- 32×32 on-chip hardware multiplier
  - supports 32×32 (single-cycle) multiplier operations, with 64-bit accumulator
  - signed or unsigned data formats
  - accumulator supports sign-extension and overflow status
  - existing multiply instructions are enhanced and new multiply and add instructions are added to the MIPS I instruction set to improve the performance of DSP applications

3.3 CPU Interface

3.3.1 Block Diagram

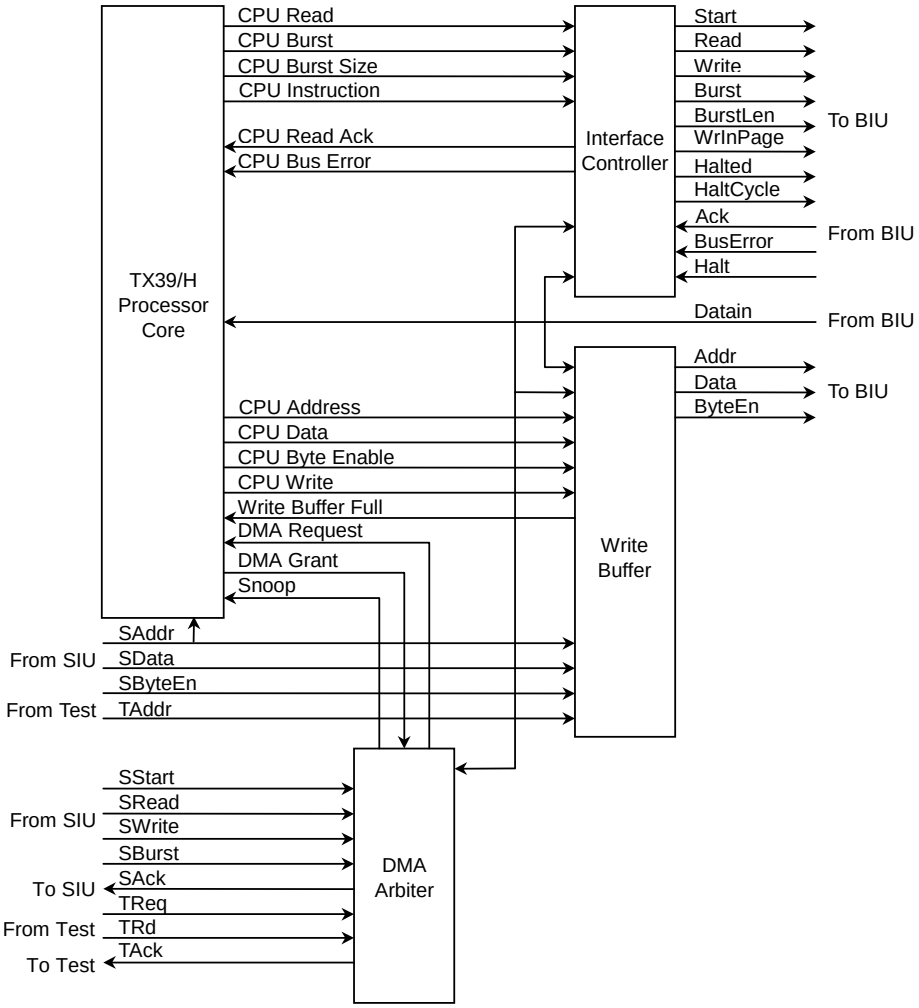


Figure 3.3.1 CPU Interface Block Diagram

The CPU Interface logic consists of the Interface Controller, the Write Buffer and the DMA Arbiter. These blocks provide the CPU core with the interface to the BIU to launch memory transactions as well as provide the SIU with the interface to launch DMA transactions to and from the BIU. Also provided is a Test Interface that allows reads and writes from and to internal function registers to bypass the CPU core.

3.3.2 Write Buffer

The CPU core writes into the Write Buffer. The Write Buffer can store up to four write transactions and contains registers to buffer the Address, Data and Byte Enables. As long as there is space in the Write Buffer, the processor will not stall. Once the Write Buffer is full, the CPU is stalled for subsequent writes until there is space in the Write Buffer. When the Write Buffer is not empty, the Write Buffer informs the Interface Controller to start a write transaction to the BIU. Once an acknowledge (Ack) is received from the BIU, the Write Buffer will proceed to the next Address, Data and Byte Enable in the buffer. A Bus Error from the BIU will also cause the Write Buffer to proceed to the next Address, Data and Byte Enable. If the addresses of two consecutive writes are within a 512-byte page, the Interface Controller will assert the WrInPage signal. This signal allows the BIU to keep the SDRAM and/or DRAM in page mode while writing within a 512-byte page.

3.3.3 Interface Controller

The Interface Controller logic contains a state machine to control the interface between the CPU core and the BIU. Read transactions are initiated when the CPU core asserts the CPU Read signal. Write transactions are initiated whenever the Write Buffer is not empty. If the CPU attempts to start a read cycle while the Write Buffer is not empty, the write transaction will complete first and the remaining writes in the write buffer are flushed before allowing the read cycle to occur. The one exception is if the read is an instruction fetch from a cacheable location. In this case, the read will take place before the write buffer is flushed. If the Interface Controller is in the middle of an In Page Write (WrInPage asserted), then one more write transaction must occur to take the BIU out of page mode before the read cycle can occur.

The Interface Controller initiates a read or write to the BIU by asserting the Start signal. The Read and Write signals will indicate whether the BIU should complete a read or write transaction. The CPU core will perform burst reads to fill the instruction and data caches. The CPU core can burst either 4, 8, 16 or 32 words depending on how the CPU core is configured. The state machine inside the Interface Controller will assert the Burst signal to the BIU and provide the burst size to the BIU using the BurstLen control signals. Bursting can greatly enhance the memory bandwidth because the BIU can keep the SDRAM and/or DRAM devices in page mode during a burst read.

The BIU will acknowledge that it is finished with a read or write by asserting the Ack signal. The Ack signal will also indicate that the data input DataIn is valid during a read. During a burst, there will be one Ack for each word of data. If the address that is provided to the BIU via the Addr signals is not mapped into any physical space that is supported by the Address Decoder logic in the BIU, the BusError signal will be asserted by the Watch Dog Timer in the BIU to terminate the bus cycle. The Interface Controller logic will use the BusError signal in place of the Ack signal to end the bus cycle and proceed to the next transaction. If the BusError occurs on a processor read, the CPU Bus Error signal will be asserted to indicate to the CPU that a Bus Error has occurred. For writes the CPU Bus Error will not be asserted since the write would have long ago been completed into the write buffer and the Bus Error will not provide meaningful information.

The BIU must halt the Interface Controller to prevent a bus cycle from starting in order to insert refresh, or when an external device requests the memory bus by asserting the DMAREQ* pin, or when the memory interface is powered down by asserting the MEMPOWERDOWN control bit in the Memory Configuration 4 register. The BIU will halt the Interface Controller by asserting the Halt signal. The Interface Controller must then assert the Halted signal once the state machine has completed the current memory transaction. Once the BIU sees the Halted signal asserted, it can proceed to insert refresh or provide a DMAGRNT* if an external device has requested the memory bus. Once refresh is finished or the external device de-asserts the DMAREQ* signal, the BIU will release the Halt signal. If the Halt signal is asserted due to the MEMPOWERDOWN signal being set then the BIU will not release the Halt signal until the HaltCycle signal is asserted by the Interface Controller.

The HaltCycle signal will be set whenever the Interface Controller starts a read or write cycle, but the Halt signal is asserted. When the BIU sees the HaltCycle signal asserted, the BIU will take one of two possible actions. On the one hand, if the address indicated by the Addr signals corresponds to an internal function register, the BIU will immediately de-assert the Halt signal to allow the transaction to begin. Once the Start signal is asserted, the BIU will immediately re-assert the Halt signal to prevent subsequent memory transactions from beginning. On the other hand, if the address does not correspond to an internal function register, the BIU must bring the memory interface out of power-down mode. This will take several cycles because the SDRAM and/or DRAM devices must be brought out of self refresh mode. Once the memory interface is brought out of power-down mode, the BIU will de-assert the Halt signal. The action taken if the address is a function register allows the CPU core to access internal function registers without having to take the memory interface out of memory power-down mode. This is useful when the processor is running a program out of cache that only requires access to internal function registers.

3.3.4 DMA Arbitration

The DMA arbiter allows the SIU to launch DMA transactions to the BIU. The SIU will initiate a DMA transaction by pulsing the SStart signal. Once this occurs, the DMA arbiter will assert the DMA Request signal to the CPU core. The CPU core will then assert the DMA Grant signal to indicate that the CPU is now idling. If the SIU is performing a DMA write (as indicated by the SWrite signal), the SAddr, SData and SByteEn signals will then be loaded into the Write Buffer once there is space available in the Write Buffer. As soon as the Write Buffer loads the signals, the DMA Arbitration logic will assert the SAck signal to indicate to the SIU that the DMA write is complete.

At the same time that the data is loaded into the Write Buffer, the Snoop signal is asserted to the CPU core. The snoop logic in the CPU core will invalidate the cache locations defined by the SAddr bits when the Snoop signal is asserted. This provides cache coherency during DMA to simplify managing the cache.

For a DMA read transaction, the SRead signal will be asserted by the SIU. The Write Buffer will always be flushed before allowing the DMA read to begin. Once the Write Buffer is empty, the DMA read will begin and the SAck signal will be asserted to end the DMA read once the BIU asserts the Ack signal. The SIU will receive data input via the DataIn bus. The SBurst signal from the SIU is used to initiate burst reads. The SIU will only request 4-word bursts, thus the BurstLen signals will be set to a burst length of 4 whenever the SBurst signal is asserted during DMA reads.

3.3.5 CPU Stop Mode

In order to reduce the power dissipation in the CPU core, the clock to the CPU core can be disabled. This is possible since the CPU core designs are implemented as fully static. The CPU core will normally be in Stop Mode and will only be active to process interrupts. The CPU enters to the CPU Stop Mode by setting the STOPCPU control bit in the Power Control Register. The STOPCPU control bit will go to the DMA Arbitration logic, which will assert DMA Request to the CPU core upon the STOPCPU signal being asserted. Once the CPU core responds with a DMA Grant, the DMA arbiter will assert a signal FSTOPCPU that is given to the Clock Generator to disable the clock to the CPU core.

3.3.6 TLB

The TLB in the TX39/H core is based on R3000.

The differences between the TLB in the TX39/H core and the TLB in the R3000 are as follows.

(1) Kseg2 TLB Mapping Area

The address from 0xFF00_0000 to 0xFFFE_FFFF in Kseg2 is a reserved area.

The following table shows the address mapping of the TX39/H core.

Segment	Virtual Address	Physical Address	Cacheable
Kseg2	0xFFFF_FFFF 0xFFFF_0000	TLB Mapped	Yes
Reserved	0xFFFE_FFFF 0xFF00_0000	0xFFFE_FFFF 0xFF00_0000	No
Kseg2	0xFEFF_FFFF 0xC000_0000	TLB Mapped	Yes
Kseg1	0xBFFF_FFFF 0xA000_0000	0x1FFF_FFFF 0x0000_0000	No
Kseg0	0x9FFF_FFFF 0x8000_0000	0x1FFF_FFFF 0x0000_0000	Yes
Kuseg	0x7FFF_FFFF 0xC000_0000	TLB Mapped	Yes

(2) Multi-hit Detection

Even though some of virtual address in the TLB entries are same, the TX39/H core does not detect multi-hit while the R3000 detects it.

(3) The Number of TLB Entries

The TX39/H core has 32 TLB entries while the R3000 has 64 TLB entries.

3.3.6.1 Memory Management System

The TX39/H provides a full-featured memory management unit (MMU) utilizing an on-chip Translation Lookaside Buffer (TLB) to provide very fast virtual memory accesses. This chapter describes the operation of the TLB and the CP0 registers that provide the software interface to the TLB. The memory mapping scheme supported by the TX39/H to translate virtual addresses to physical addresses is also described in detail.

3.3.6.2 Memory System Architecture

The TX39/H's virtual memory system logically expands the CPU's physical memory space by translating addresses composed in a large virtual address space into the physical memory space of the TX39/H.

Figure 3.3.2 shows the form of an TX39/H virtual address. The most significant 20 bits of a 32-bit virtual address are called the virtual page number, or VPN. The VPN allows mapping of 4 Kbyte pages, while the least significant 12 bits (the offset within a page) are passed along unchanged to form the physical address. The three most significant bits of VPN (bits 31..29) further define how addressed are mapped according to whether the TX39/H is in user mode or kernel mode (these modes are described in the paragraphs that follow).

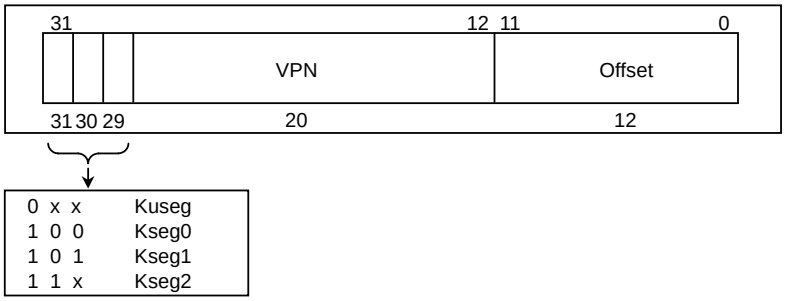


Figure 3.3.2 Virtual Address Format

A 6-bit process identifier field is appended to each virtual address to form unique virtual addresses for up to 64 processes. The mapping of these extended, process-unique virtual addresses to physical addresses need not be one-to-one; virtual addresses of two or more different processes may map to the same physical address.

3.3.6.3 Privilege States

The TX39/H provides two privilege states: the Kernel mode, which is analogous to the "supervisory" mode provided by many machines, and the User mode, where non-supervisory programs are executed. The TX39/H enters the Kernel mode whenever an exception is detected and remains in the Kernel mode until a Restore From Exception (*rfe*) instruction is executed.

Address mapping is different for Kernel and User modes. To simplify the management of user state from within the Kernel, the user-mode address space is a subset of the Kernel-mode address space. Figure 3.3.3 shows the virtual-to-physical memory map for both the User mode and Kernel mode segments.

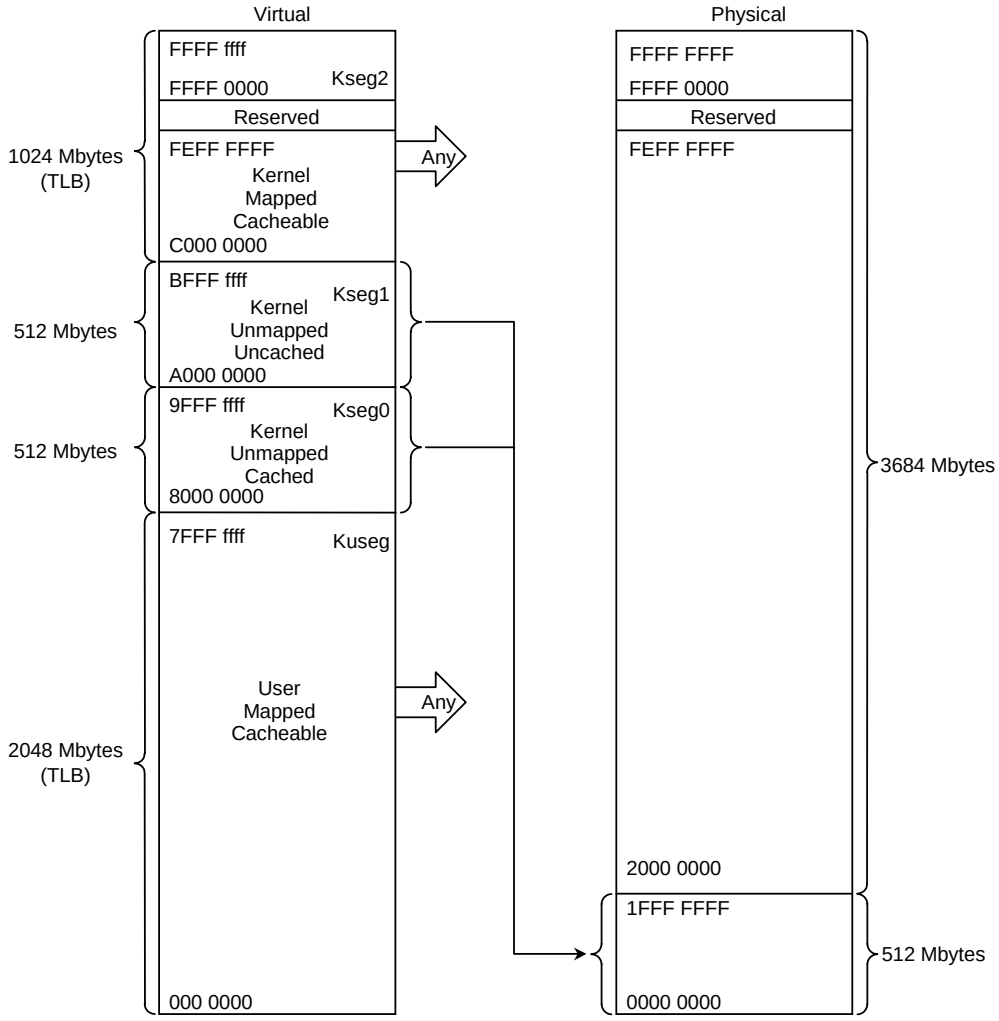


Figure 3.3.3 The TX39/H Virtual Memory Map

3.3.6.4 User-Mode Virtual Addressing

When the processor is operating in User mode, a single, uniform virtual address space (kuseg) of 2 Gbytes is available for users. All valid User-mode virtual addresses have the most-significant bit cleared to 0. An attempt to reference an address with the most-significant bit set while in the User mode causes an Address Error exception.

The 2 Gbyte User segment begins at address zero. The TLB maps all references to kuseg identically from Kernel and User modes and controls access cacheability. (The N bit in a TLB entry determines whether the reference will be cached.)

Kuseg is typically used to hold user code and data, and the current user process typically resides in kuseg.

3.3.6.5 Kernel-Mode Virtual Addressing

When the processor is operating in Kernel mode, three distinct virtual address spaces (in addition to kuseg) are simultaneously available. The three segments dedicated to the kernel are:

- kseg0. This cached, unmapped segment begins at virtual address 0x8000_0000 and is 512 Mbytes long.
- kseg1. This uncached, unmapped segment begins at virtual address 0xA000_0000 and is 512 Mbytes long.
- kseg2. This kernel-mapped, cacheable segment begins at virtual address 0xC000_0000 and is 1 Gbytes long.

Kseg0. When the most-significant three bits of the virtual address are “100”, the virtual address space selected is a 512-Mbyte kernel physical space (kseg0). The TX39/H direct-maps references within kseg0 onto the first 512 Mbytes of physical address space. These references use cache memory, but they do not use TLB entries. Thus, kseg0 is typically used for kernel executable code and some kernel data.

Kseg1. When the most-significant three bits of the virtual address are “101”, the virtual address space selected is a 512-Mbyte kernel physical space (kseg1). The processor directly maps kseg1 onto the first 512 Mbytes of physical address space and uses no TLB entries. Unlike kseg0, kseg1 uses uncashed references. An operating system typically uses kseg1 for I/O register, ROM code, and disk buffers.

Kseg2. When the most-significant two bits of the virtual address are “11”, the virtual address space selected is a 1024 Mbyte kernel virtual space (kseg2). Like kuseg, kseg2 uses TLB entries to map virtual addresses to arbitrary physical ones, with or without caching. (The N bit in a TLB entry determines whether the reference will be cached.) An operating system typically uses kseg2 for stacks and per-process data that it must remap on context switches, for user page tables (memory map), and for some dynamically allocated data areas. Kseg2 allows selective caching and mapping on a per-page basis, rather than requiring an all or nothing approach.

3.3.6.6 Virtual Memory and the TLB

Mapped virtual addresses are translated into physical addresses using a Translation Lookaside Buffer (TLB). The TLB is a fully associative memory device that holds 32 entries to provide mapping of 32 4 Kbyte pages. When address mapping is indicated (that is, when the access is in *kuseg* or *kseg2*), each TLB entry is simultaneously checked for a match with the extended virtual address.

The CPU supports up to four coprocessors. Coprocessor 0 (CP0), which is called the System Control Coprocessor, is implemented as an integral part of the TX39/H. CP0 supports address translation, exception handling, and other "privileged" operations. It consists of the 32-entry TLB plus the ten registers shown in Figure 3.3.4. The sections that follow describe how each of the four TLB-related registers is used. (Note: CP0 functions and registers associated with exception handling are described in Chapter 6 of the *32-Bit TX System RISC TX39 Family Architecture* manual.)

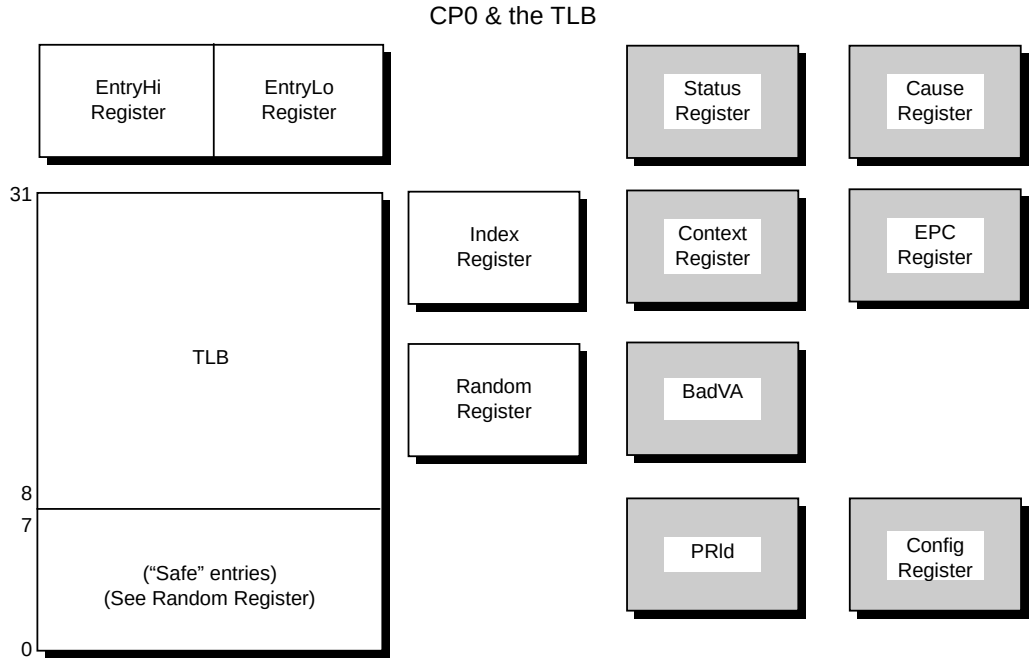


Figure 3.3.4 The CP0 Registers & the TLB

3.3.6.7 TLB Entries

Each TLB entry is 32 bits wide and its format is shown in Figure 3.3.5 Each of the fields of a TLB entry has a corresponding field in the EntryHi/EntryLo register pair described next. Refer to Figure 3.3.6 for a description of each of the TLB entry fields.

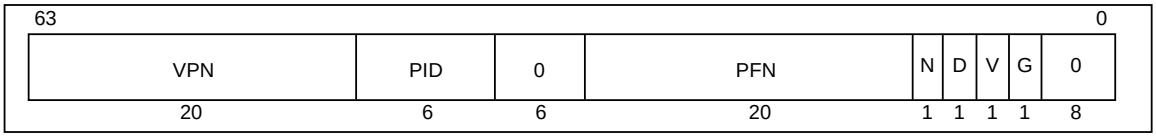


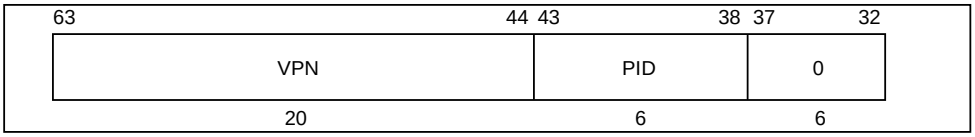
Figure 3.3.5 Format of a TLB Entry

3.3.6.8 EntryHi & EntryLo Registers (CP0 register No.10, No.2)

These two registers provide the data pathway through which the TLB is read, written, or probed. When address translation exceptions occur, these registers are loaded with relevant information about the address that caused the exception. The format of the *EntryHi* and *EntryLo* register pair a is the same as the format of TLB entry and is illustrated in Figure 3.3.6.

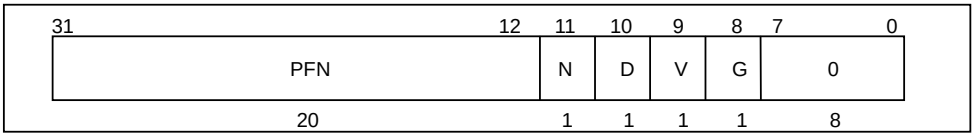
EntryLo is the natural form of a Page Table Entry (PTE); however, since PTEs are always loaded by system software, not by the TX39/H hardware, an operating system can use another format for memory-resident PTEs.

TLB EntryHi Register (CP0 register No.10)



VPN	Virtual Page Number. Bits 31..12 of virtual address.
PID	Process ID Process ID field. A 6-bit field which lets multiple processes share the TLB while each process has a distinct mapping of otherwise identical virtual page numbers.
0	Reserved. Currently ignores writes, returns zero when read.

TLB EntryLo Register (CP0 register No.2)



PFN	Page Frame Number. Bits 31..12 of the physical address. The TX39/H maps a virtual page to PFN.
N	Non-cacheable. If this bit is set, the page is marked as non-cacheable and the TX39/H directly accesses main memory instead of first accessing the cache.
D	Dirty. If this bit is set, the page is marked as “dirty” and therefore writable. This bit is actually a “write-protect” bit that software can use to prevent alteration of data. If an entry is accessed for a write operation when the D bit is cleared, the TX39/H causes a TLB Mod trap. The TLB entry is not modified on such a trap.
V	Valid. If this bit it set, it indicates that the TLB entry is valid; otherwise, a TLBL or TLBS Miss occurs.
G	Global. If this bit it set, the TX39/H ignores the PID match requirement for valid translation. In kseg2, the Global bit lets the kernel access all mapped data without requiring it to save or restore PID (Process ID) values.
0	Reserved. Currently ignores writes, returns zero when read.

Figure 3.3.6 The TLB EntryHi & EntryLo Registers

3.3.6.9 Virtual Address Translation

During virtual-to-physical address translation, the TX39/H compares PID and the highest 20 bits (the VPN) of the virtual address to the contents of the TLB. Figure 3.3.7 illustrates the TLB address translation process.

A virtual address matches a TLB entry when the virtual page number (VPN) field of the virtual address equals the VPN field of the entry, and either the Global (G) bit of the TLB entry is set, or the process identifier (PID) field of the virtual address (as held in the EntryHi register) matches the PID field of the TLB entry. While the Valid (V) bit of the entry must be set for a valid translation to take place, it is not involved in the determination of a matching TLB entry.

If a TLB entry matches, the physical address and access control bits (N, D, and V) are retrieved from the matching TLB entry. Otherwise, a TLB miss (or UTLB miss) exception occurs. If the access control bits (D and V) indicate that the access is not valid, a TLB modification or TLB miss exception occurs. If the N bit is set, the physical address that is retrieved is used to access main memory, bypassing the cache.

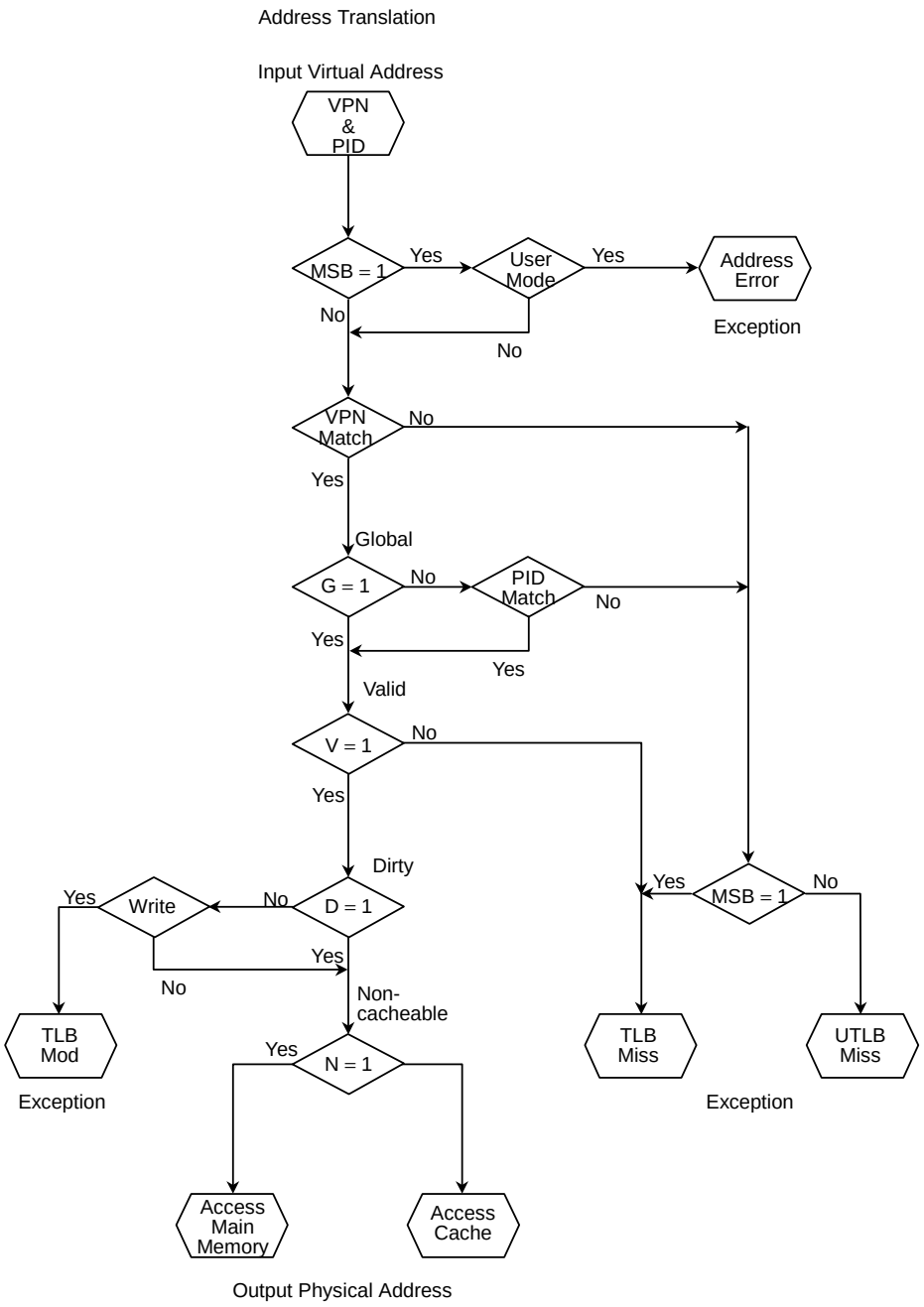


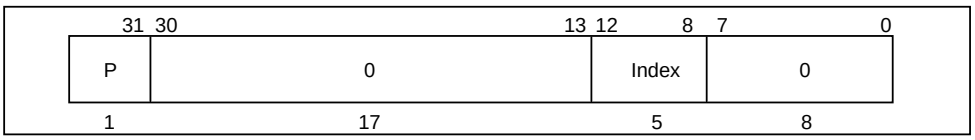
Figure 3.3.7 TLB Address Translation

3.3.6.10 The Index Register (CP0 register No.0)

The *Index* Register is a 32-bit, read/write register, which has 5 bits that index an entry in the TLB. The most significant bit of the register shows the success or failure of a TLB Probe (tlbp) instruction (described later in this chapter).

The *Index* register also specifies the TLB entry that will be affected by the TLB Read (tlbr) and TLB Write Index (tlbwi) instructions. Figure 3.3.8 shows the format of the *Index* register.

Index Register



P	Probe failure. Set to 1 when the last TLBProbe (tlbp) instruction was unsuccessful.
Index	Index to the TLB entry that will be affected by the TLBRead and TLBWrite instruction.
0	Reserved. Currently ignores writes, returns zero when read.

Figure 3.3.8 The Index Register

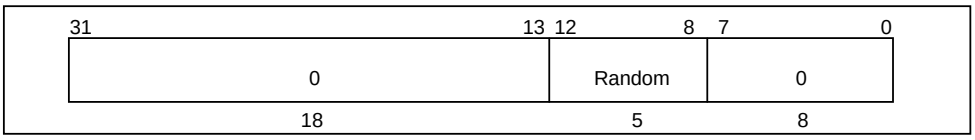
3.3.6.11 The Random Register (CP0 register No.1)

The *Random* Register is a 32-bit register. The format of the Random Register is shown in Figure 3.3.9. The five-bit *Random* filed indexes a random entry in the TLB. The TX39/H decrements the *Random* field during every machine cycle but constrains the value of this field to the TLB indexes from 31 to 8 (the counter wraps around from 8 to 31 skipping values 7 through 0).

The TLB Write Random (tlbwr) instruction is used to write the TLB entry that this register indexes. The first eight entries (0 to 7) are the “safe” entries because a tlbwr instruction can never replace the contents of these entries. Typically, these eight entries are reserved for use by the operating system.

Although normal operations never require it, the contents of this register can be read to verify proper operation of the process. To further simplify testing, the *Random* field is set to a value of 31 when the TX39/H is reset.

Random Register



Random	A random index with a value ranging from 8 to 31 to a TLB entry.
0	Reserved. Currently ignores writes, returns zero when read.

Figure 3.3.9 The Random Register

3.3.6.12 TLB Instructions

The instructions that the TX39/H provides for working with the TLB are listed in Table 3.3.1 and described briefly below.

Table 3.3.1 TLB Instructions

Op Code	Description
tlbp	Translation Lookaside Buffer Probe
tlbr	Translation Lookaside Buffer Read
tlbwi	Translation Lookaside Buffer Write Index
tlbwr	Translation Lookaside Buffer Write Random

Translation Lookaside Buffer Probe (tlbp).

This instruction probes the TLB to see if an entry matches the *EntryHi* register contents. If a match exists, the TX39/H loads the *Index* register with the index of the entry that matches the *EntryHi* register. When on match exists, the TX39/H sets the high order bit (the *P* bit) of the Index register.

Translation Lookaside Buffer Read (tlbr).

This instruction loads *EntryHi* and *EntryLo* registers with the contents of the TLB entry specified by the contents of the *Index* register.

Translation Lookaside Buffer Write Index (tlbwi).

This instruction loads specified TLB entry with the contents of the *EntryHi* and *EntryLo* registers. The contents of the Index register specify the TLB entry.

Translation Lookaside Buffer Write Random (tlbwr).

This instruction loads a pseudo-randomly specified TLB entry with the contents of the *EntryHi* and *EntryLo* registers. The contents of the *Random* register specify the TLB entry.

4. BIU Module

4.1 Overview

The BIU provides an interface to the memory devices in the system, including static devices, as well as DRAM, HDRAM and SDRAM devices. An overall block diagram of the BIU is shown in Figure 4.1.1. The BIU provides control to support the following memory devices:

- SDRAM chip select (DCS0*, DCS1*) and/or DRAM (RAS0*, RAS1*)
  - 8-bit or 16-bit data bus width SDRAM configuration
  - 16-bit or 32-bit data bus width DRAM configuration
  - 16-bit or 32-bit data bus width HDRAM configuration
  - 4 Mbit, 16 Mbit, 64 Mbit and 128 Mbit (*) parts supported
    (*) 128 Mbit is supported only by TMPR3912AU-92/XB-92.
  - page mode reads and writes supported
  - independent refresh counters for each bank
  - self refreshing parts supported to retain memory when system is powered down
- 4 general purpose chip selects (CS3*-CS0*)
  - 16-bit or 32-bit ports
  - programmable wait states
  - read page mode
- 4 general purpose chip selects (MCS3*- MCS0*)
  - 16-bit ports
  - programmable wait states
  - read page mode
- 2 full PCMCIA slots
  - 8 bit or 16 bit ports
  - IORD and IOWR provided to support I/O cards
  - WAIT signal supported

Most of the BIU logic uses the CLK signal (×4 frequency of SYSCLK) as the clock, but in order to increase the memory bandwidth, SDRAM devices are accessed using the signal DCLKOUT (×8 frequency of SYSCLK). The DCLKOUT signal is also used to provide more precise timing for generating the control signals for the chip selects, as well as the SDRAM and DRAM control signals. Input data is sampled using the DCLKIN signal which is connected externally to the DCLKOUT signal. The external connection ensures that the input data and DCLKIN will not be skewed, which is necessary to meet the hold time specification on the SDRAM devices. Also, the external looping of DCLKOUT to DCLKIN helps to improve the access time of normal DRAMs, since the input data will be sampled closer to the end of the access.

4.1.1 Block Diagram

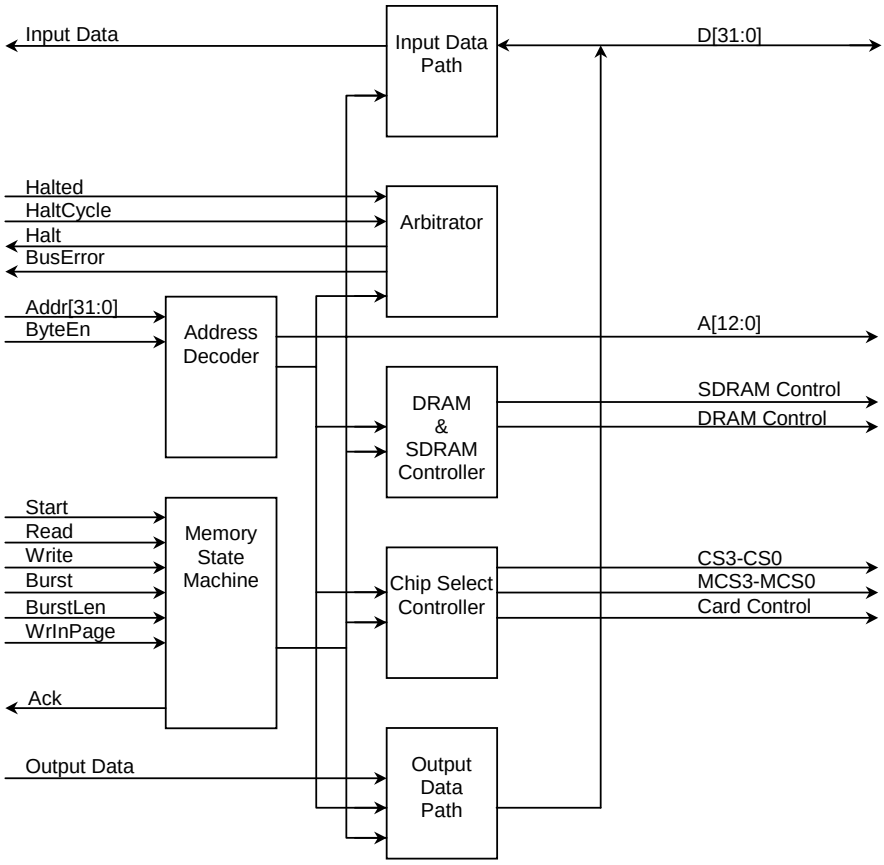


Figure 4.1.1 BIU Block Diagram

4.2 Address Decoder

The Address Decoder provides the decoding for the address space in the system. The Address Decoder also contains the logic to generate the address signals A[12:0] for the memory devices, along with Address Re-Mapper logic used to re-direct memory addresses.

4.2.1 System Address Map

The System Address Map is shown in Table 4.2.1. The address space is partitioned to support two banks (Bank 1 and Bank 0) of DRAM and/or SDRAM, four general purpose 16/32-bit chip selects (CS3-CS0), four general purpose 16-bit-only chip selects (MCS3-MCS0), two PCMCIA slots (Card 2 and Card 1), and internal function register access. The address space contains two primary regions: kseg0/kseg1 are only accessible in kernel mode, while kuseg is accessible in either user or kernel mode.

MCS3-MCS0 are available in kuseg space and are thus always available to either user or kernel mode accesses.

CS3-CS1 are located in both kseg and kuseg space, but accesses in kuseg space can be independently disabled for each chip select using the ENCS3USER, ENCS2USER, and ENCS1USER control bits in the Memory Configuration 0 Register. CS0 is used as the chip select for the boot ROM in the system. It is mapped starting at address $11000000 in kernel space, which will map into the TX39/H boot vector location at address $1FC00000. CS0 is also mapped in kuseg space to allow access to the ROM by programs operating in user mode.

Card 1 and Card 2 are mapped into kuseg space for memory accesses and kernel-only space for IO and Attribute accesses. To access the attribute space in a card, the CARD1IOEN or CARD2IOEN bits in the Memory Configuration 3 register must be cleared prior to accessing the Card space. For I/O accesses the CARD1IOEN or CARD2IOEN bits must be set prior to accessing the Card space.

The processor can write to the Mode Registers in the SDRAMs in kernel mode at address $10E00000 and $10F00000 for Bank 0 and Bank 1 SDRAM banks, respectively. Bank 0 and Bank 1 are each mapped into four different locations. Each location is just an image of the other locations. Each Bank can contain either DRAM or SDRAM devices, with the only restriction being that Bank 1 cannot contain SDRAM if Bank 0 contains DRAM.

A system can be built without DRAM or SDRAM and instead can contain SRAM as the main memory. In this case at least one bank of SRAM should be connected to CS1 and the ENCS1DRAM control bit in the Memory Configuration 0 Register should be set. Setting this bit will cause CS1 to map in place of Bank 0 and Bank 1, as shown in Table 4.2.1. This is required because exception vectors are mapped into physical address 0x0000_0080 (in case of BEU = 0 in TX39/H core STATUS register), thus it is required that some memory exist in this region to support exception handling. Accesses to reserved or non-enabled locations will respond with a Bus Error if the Watch Dog Timer is enabled.

Table 4.2.1 System Address Map (Physical Address)

Size	Address	Segment	Devices
16 Mbyte	FF00_0000	reserved	reserved
1 Gbyte	C000_0000	kseg2	reserved
1 Gbyte	7C00_0000	kuseg	reserved
64 Mbyte	7800_0000	kuseg	MCS3
64 Mbyte	7400_0000	kuseg	MCS2
64 Mbyte	7000_0000	kuseg	MCS1
64 Mbyte	6C00_0000	kuseg	MCS0
64 Mbyte	6800_0000	kuseg	Card 2 (Memory)
64 Mbyte	6400_0000	kuseg	Card 1 (Memory)
64 Mbyte	6000_0000	kuseg	CS3 (if enabled)
64 Mbyte	5C00_0000	kuseg	CS2 (if enabled)
64 Mbyte	5800_0000	kuseg	CS1 (if enabled)
128 Mbyte (*)	5000_0000	kuseg	CS0 (ROM)
128 Mbyte	4800_0000	kuseg	reserved
32 Mbyte	4600_0000	kuseg	DRAM BANK1
32 Mbyte	4400_0000	kuseg	DRAM BANK0
32 Mbyte	4200_0000	kuseg	CS1 (if enabled, else DRAM BANK 1)
32 Mbyte	4000_0000	kuseg	CS1 (if enabled, else DRAM BANK 0)
512 Mbyte	2000_0000	reserved	reserved
240 Mbyte (*)	1100_0000	kseg0, kseg1	CS0 (ROM)
1 Mbyte	10F0_0000	kseg0, kseg1	SDRAM BANK 1 Mode Register
1 Mbyte	10E0_0000	kseg0, kseg1	SDRAM BANK 0 Mode Register
2 Mbyte	10C0_0000	kseg0, kseg1	Internal Function Registers
4 Mbyte	1080_0000	kseg0, kseg1	CS3
4 Mbyte	1040_0000	kseg0, kseg1	CS2
4 Mbyte	1000_0000	kseg0, kseg1	CS1
64 Mbyte	0C00_0000	kseg0, kseg1	Card 2 (I/O or Attribute)
64 Mbyte	0800_0000	kseg0, kseg1	Card 1 (I/O or Attribute)
32 Mbyte	0600_0000	kseg0, kseg1	DRAM BANK1
32 Mbyte	0400_0000	kseg0, kseg1	DRAM BANK 0
32 Mbyte	0200_0000	kseg0, kseg1	CS1 (if enabled, else DRAM BANK 1)
32 Mbyte	0000_0000	kseg0, kseg1	CS1 (if enabled, else DRAM BANK 0)

(*) Note: Since external addresses (HA[25:0] are only 26 bit width, memory space which can be used physically is only 64 Mbyte of these CS0 area. Upper 5 bits of internal addresses (Addr[31:27]) are discarded, so the access to the remaining area overlaps to the physical 64 Mbyte area.

4.2.2 Address Generation

The Address Decoder logic generates the A[12:0] address bus for the memory devices. Addresses are multiplexed on the A[12:0] address bus. For static devices such as CS3-CS0, MCS3-MCS0, Card 2 and Card 1, the addresses are multiplexed such that Addr[25:13] is provided first-in-time followed by Addr[12:0], as shown in Figure 4.2.1. Addr[25:13] must be latched with an external latch using the ALE signal. The latched addresses concatenated with A[12:0] will provide a 26-bit address bus for the system, HA[25:0].

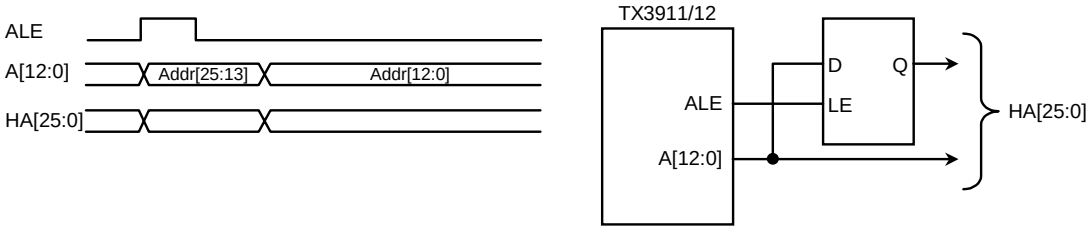


Figure 4.2.1 ALE Timing

DRAM and SDRAM devices contain multiplexed address buses. Thus, A[12:0] will be directly connected to the DRAM and SDRAM devices. In order to support a wide range of DRAM and SDRAM types and configurations, there are control bits in the Memory Configuration 0 Register that provide software configuration for the Row and Column address bit positions. These control bits are ROWSEL0[1:0] and COLSEL0[2:0] for Bank 0, and ROWSEL1[1:0] and COLSEL1[2:0] for Bank 1.

Table 4.2.2 shows the position that specific address bits are placed, depending on the configuration of the control bits. This flexibility in address bit positioning makes it possible to support 4-Mbit, 16-Mbit, and 64-Mbit DRAM and SDRAM devices with different bus width configurations. The two sets of control bits make it possible to support different devices in Bank 0 and Bank 1. The "ϕ" character in the table means that this bit position will be a zero and the "X" implies a "don't care" bit position.

Table 4.2.2 TMPR3912 DRAM and SDRAM Address Mapping

ROW A[12:0]	ROWSEL
18, 17:9	00
22, 18, 20, 19, 17:9	01
20, 22, 21, 19, 17:9	10
22, 23, 21, 19, 17:9	11

COL A[12:0]	COLSEL
22, 20, 18, 8:1	0000
19, 18, 8:2	0001
21, 20, 18, 8:2	0010
23, 22, 20, 18, 8:2	0011
24, 22, 20, 18, 8:2	0100
18, ϕ, X, 8:0	0101
22, X, ϕ, 21, 8:0	0110
18, ϕ, X, 21, 8:1	0111
22, X, ϕ, 23, 21, 8:1	1000
24, 23, 21, 8:2	1001

Example: When ROWSEL[1:0] = 01, the address bits Addr[22], Addr[18], Addr[20], Addr[19] and Addr[17:9] are outputed to A[12], A[11], A[10], A[9] and A[8:0] pins respectively as row address for DRAM and/or SDRAM.

When COLSEL[3:0] = 1000, the address bits Addr[22], undefined value, zero data, Addr[23], Addr[21], and Addr[8:1] are outputed to A[12], A[11], A[10], A[9], A[8], A[7:0] respectively as column address for DRAM and/or SDRAM.

4.2.3 Address Re-Mapper Logic

The Address Decoder logic provides an Address Re-Mapper that allows addresses to be re-directed. There are two separate sets of Re-Mapper registers provided. The Memory Configuration 5-8 Registers contain the STARTVAL2[31:9], MASK2[3:0] and RMAPADD2[31:9] control bits for one Re-Mapper, and the STARTVAL1[31:9], MASK1[3:0] and RMAPADD1[31:9] control bits for the other Re-Mapper. Each Re-Mapper contains its own enable bit defined by the ENRMAP2 and ENRMAP1 control bits in the Memory Configuration 0 Register. If the enable bits are not set, then no address Re-Mapping will occur. The Re-Mapper works as described in the following paragraph.

The Addr[31:2] address output of the CPU Interface is compared with the STARTVAL[31:9] bits. If the addresses compare, the upper address bits are replaced with the RMAPADD[31:9] bits. The lower bits[8:2] are always passed through. This provides a 512-byte block that can be re-mapped to any other 512-byte block. The MASK[3:0] bits are used to select either a 512, 1K, 2K, 4K or 8 Kbyte block. A bitwise “AND” of the MASK[3:0] bits with Addr[12:9] is performed before the address comparison. If the address compares to the STARTVAL bits then the address is replaced with the RMAPADD bits. The resulting address is then provided to the Address Decoder logic. Table 4.2.3 shows the resulting address when the comparison is true for the different possible MASK settings. The STARTVAL[12:9] bits are valid as shown by the “V” in the table. Otherwise the bits must be set to zero as shown.

Table 4.2.3 Address Re-Mapper

MASK[3:0]	STARTVAL[12:9]	Address Result
F	V, V, V, V	RMAPADD[31:9], Addr[8:2]
E	V, V, V, 0	RMAPADD[31:10], Addr[9:2]
C	V, V, 0, 0	RMAPADD[31:11], Addr[10:2]
8	V, 0, 0, 0	RMAPADD[31:12], Addr[11:2]
0	0, 0, 0, 0	RMAPADD[31:13], Addr[12:2]

The primary function of the Re-Mapper logic is to support Flash devices, which have slow write access times, are block-oriented, and require a lot of overhead. The re-mapping of Flash writes into normal memory space allows the Flash to be written much less frequently. Anytime a write occurs within the re-mapped range, the write will instead take place into the desired re-mapped memory location. The software can then setup memory protection to protect writes to the rest of the Flash, outside of the previously selected range or block. Once a memory protection exception occurs, the software can disable the Re-Mapper, move the data from normal memory to the Flash, then move the STARTVAL bits to point to a new address, corresponding to a new address block.

4.3 Chip Select Controller

4.3.1 Description

The Chip Select Controller provides the logic required to generate CS3-CS0, MCS3-MCS0, Card 2, and Card 1 control signals. CS3-CS0 can each independently be set as either 32-bit or 16-bit ports. MCS3-MCS0, are always 16-bit ports. Card 2 and Card 1 can be set as either 16-bit or 8-bit ports.

The access time for each chip select is individually programmed using the control bits in the Memory Configuration 1-3 Registers. Also provided for CS3-CS0 and MCS3-MCS0 is separate access time control for Read Page Mode that can be enabled or disabled independently for each chip select using the ENCSχPAGE (χ = 0, 1, 2, 3), ENMCSχPAGE (χ = 0, 1, 2, 3) control bits in the Memory Configuration 3 Register. The Read Page Mode provides burst read capability to support memory devices such as page mode ROMs. There is separate access time control provided for memory accesses to Card 2 or Card 1, or IO/Attribute accesses to Card 2 or Card 1. The address inputs for all 32-bit memory devices should be connected starting with HA[2] and all 16-bit memory devices should be connected starting with HA[1]. The PCMCIA cards will use all the address bits HA[25:0].

4.3.2 Access Mapping

This section describes the mapping between the byte lanes of the CPU registers (R[ ]) and the byte lanes of external signals (D[ ]).

Depending upon CPU endianness, the signal names of some pins are reorganized (see Table 4.3.1). In this section Xl[ ] means X[ ] in little-endian mode, Xb[ ] means X[ ] in big-endian mode. For example CASl[ ] means CAS[ ] in little-endian mode, and CASb[ ] means CAS[ ] in big-endian mode.

example:

In case of little-endian, the physical pins [133:145] are named Dl[7:0], whereas in big-endian case, the same physical pins are named Db[31:24].

Table 4.3.1 Signal Name in each Endian Mode

Pin No.			Little-endian	Big-endian	
3911BXB	3912AU-92	3912XB-92			
P7, N7, M6, R7, N6, M5, R6, P5	133-145	H12-E14	Dl[7:0]	Db[31:24]	
N4, R5, N3, P3, R4, R3, P1, N2	146-159	E13-E11	Dl[15:8]	Db[23:16]	
C7, D7, C8, A7, B6, D6, A6, B5	27-16	H1-F3	Dl[23:16]	Db[15:8]	
C5, B4, A5, C4, A4, B3, A1, A3	14-2	F4-C3	Dl[31:24]	Db[7:0]	
E3	195	D6	CASl[0]*	CASb[3]*	
E2	197	B5	CASl[1]*	CASb[2]*	
D3	198	C5	CASl[2]*	CASb[1]*	
E1	199	E6	CASl[3]*	CASb[0]*	
C1	207	B2	DQMLl	DQMHb	
B1	208	A2	DQMHl	DQMLb	
P11	117	L14	CARD1CSL*	CARD1CSL*	(*1)
R10	118	K12	CARD1CSH*	CARD1CSH*	(*1)
R15	107	L11	CARD2CSL*	CARD2CSL*	(*1)
R13	106	N13	CARD2CSH*	CARD2CSH*	(*1)

*1) not affected by endianness

Table 4.3.2, 4.3.3, 4.3.4, and 4.3.5 show the access mapping in little-endian mode, and Table 4.3.6, 4.3.7, 4.3.8, and 4.3.9 shows the access mapping in big-endian mode.

Note that in both little-endian configuration and big-endian configuration, each device is connected to the same physical pins.

Table 4.3.2 Access Mapping (SDRAM in Little-Endian mode) (X: undefined)

Access	HA[1:0]	Data Bus Width (*1)	DQMHl, DQMLl	Dl[31:24]	Dl[23:16]	Dl[15:8]	Dl[7:0]
Word(1/2)	00	16-bit	LL	R[15:8]	R[7:0]	X	X
Word(2/2)	10	16-bit	LL	R[31:24]	R[23:16]	X	X
Triple Byte(1/2)	01	16-bit	LH	R[15:8]	X	X	X
Triple Byte(2/2)	10	16-bit	LL	R[31:24]	R[23:16]	X	X
Triple Byte(1/2)	00	16-bit	LL	R[15:8]	R[7:0]	X	X
Triple Byte(2/2)	10	16-bit	HL	X	R[23:16]	X	X
Half-Word	00	16-bit	LL	R[15:8]	R[7:0]	X	X
Half-Word	10	16-bit	LL	R[15:8]	R[7:0]	X	X
Byte	00	16-bit	HL	X	R[7:0]	X	X
Byte	01	16-bit	LH	R[7:0]	X	X	X
Byte	10	16-bit	HL	X	R[7:0]	X	X
Byte	11	16-bit	LH	R[7:0]	X	X	X
Word(1/4)	00	8-bit	LH	R[7:0]	X	X	X
Word(2/4)	01	8-bit	LH	R[15:8]	X	X	X
Word(3/4)	10	8-bit	LH	R[23:16]	X	X	X
Word(4/4)	00	8-bit	LH	R[31:24]	X	X	X
Triple Byte(1/3)	01	8-bit	LH	R[15:7]	X	X	X
Triple Byte(2/3)	10	8-bit	LH	R[23:16]	X	X	X
Triple Byte(3/3)	11	8-bit	LH	R[31:24]	X	X	X
Triple Byte(1/3)	00	8-bit	LH	R[7:0]	X	X	X
Triple Byte(2/3)	01	8-bit	LH	R[15:8]	X	X	X
Triple Byte(3/3)	10	8-bit	LH	R[23:16]	X	X	X
Half-Word(1/2)	00	8-bit	LH	R[7:0]	X	X	X
Half-Word(2/2)	01	8-bit	LH	R[15:8]	X	X	X
Half-Word(1/2)	10	8-bit	LH	R[7:0]	X	X	X
Half-Word(2/2)	11	8-bit	LH	R[15:8]	X	X	X
Byte	00	8-bit	LH	R[7:0]	X	X	X
Byte	01	8-bit	LH	R[7:0]	X	X	X
Byte	10	8-bit	LH	R[7:0]	X	X	X
Byte	11	8-bit	LH	R[7:0]	X	X	X

*1) selected by Memory Configuration 0 Register BANKχCONF[1:0] (χ = 0, 1).

Table 4.3.3 Access Mapping (DRAM/HDRAM in Little-Endian Mode)

Access	HA[1:0]	Data Bus Width (*1)	CASl[3:0]* (*2)	Dl[31:24]	Dl[23:16]	Dl[15:8]	Dl[7:0]
Word	00	32-bit	LLLL	R[31:24]	R[23:16]	R[15:8]	R[7:0]
Triple Byte	00	32-bit	HLLL	X	R[23:16]	R[15:8]	R[7:0]
Triple Byte	01	32-bit	LLLH	R[31:24]	R[23:16]	R[15:8]	X
Half-Word	00	32-bit	HHLL	X	X	R[15:8]	R[7:0]
Half-Word	10	32-bit	LLHH	R[15:8]	R[7:0]	X	X
Byte	00	32-bit	HHHL	X	X	X	R[7:0]
Byte	01	32-bit	HHLH	X	X	R[7:0]	X
Byte	10	32-bit	HLHH	X	R[7:0]	X	X
Byte	11	32-bit	LHHH	R[7:0]	X	X	X
Word(1/2)	00	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Word(2/2)	10	16-bit	LLHH	R[31:24]	R[23:16]	X	X
Triple Byte(1/2)	01	16-bit	LHHH	R[15:8]	X	X	X
Triple Byte(2/2)	10	16-bit	LLHH	R[31:24]	R[23:16]	X	X
Triple Byte(1/2)	00	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Triple Byte(2/2)	10	16-bit	HLHH	X	R[23:16]	X	X
Half-Word	00	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Half-Word	10	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Byte	00	16-bit	HLHH	X	R[7:0]	X	X
Byte	01	16-bit	LHHH	R[7:0]	X	X	X
Byte	10	16-bit	HLHH	X	R[7:0]	X	X
Byte	11	16-bit	LHHH	R[7:0]	X	X	X

*1) selected by Memory Configuration 0 Register BANKχCONF[1:0] (χ = 0, 1).

*2) asserted for both read access and write access.

Table 4.3.4 Access Mapping (CSχand MCSχin Little-Endian Mode) (X: undefined)

Access	HA[1:0]	Port Size (*1)	CASl[3:0]* (*2)	Dl[31:24]	Dl[23:16]	Dl[15:8]	Dl[7:0]
Word	00	32-bit	LLLL	R[31:24]	R[23:16]	R[15:8]	R[7:0]
Triple Byte	00	32-bit	HLLL	X	R[23:16]	R[15:8]	R[7:0]
Triple Byte	01	32-bit	LLLH	R[31:24]	R[23:16]	R[15:8]	X
Half-Word	00	32-bit	HHLL	X	X	R[15:8]	R[7:0]
Half-Word	10	32-bit	LLHH	R[15:8]	R[7:0]	X	X
Byte	00	32-bit	HHHL	X	X	X	R[7:0]
Byte	01	32-bit	HHLH	X	X	R[7:0]	X
Byte	10	32-bit	HLHH	X	R[7:0]	X	X
Byte	11	32-bit	LHHH	R[7:0]	X	X	X
Word(1/2)	00	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Word(2/2)	10	16-bit	HHLL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	01	16-bit	HHLH	X	X	R[15:8]	X
Triple Byte(2/2)	10	16-bit	HHLL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	00	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Triple Byte(2/2)	10	16-bit	HHHL	X	X	X	R[23:16]
Half-Word	00	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Half-Word	10	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Byte	00	16-bit	HHHL	X	X	X	R[7:0]
Byte	01	16-bit	HHLH	X	X	R[7:0]	X
Byte	10	16-bit	HHHL	X	X	X	R[7:0]
Byte	11	16-bit	HHLH	X	X	R[7:0]	X

1) selected by Memory Configuration 0 Register CSχSIZE (CSχ) (χ = 0, 1, 2, 3).
Port size for MCSχ* (χ = 0, 1, 2, 3) are fixed to 16 bit width.

*2) asserted for write access only.

Table 4.3.5 Access Mapping (PCMCIA in Little-Endian Mode) (X: undefined)

Access	HA[1:0]	Port Size (*1)	CARDCSnH*, CARDCSnL*	DI[31:24]	DI[23:16]	DI[15:8]	DI[7:0]
Word(1/2)	00	16-bit	LL	X	X	R[15:8]	R[7:0]
Word(2/2)	10	16-bit	LL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	01	16-bit	LH	X	X	R[15:8]	X
Triple Byte(2/2)	10	16-bit	LL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	00	16-bit	LL	X	X	R[15:8]	R[7:0]
Triple Byte(2/2)	10	16-bit	HL	X	X	X	R[23:16]
Half-Word	00	16-bit	LL	X	X	R[15:8]	R[7:0]
Half-Word	10	16-bit	LL	X	X	R[15:8]	R[7:0]
Byte	00	16-bit	HL	X	X	X	R[7:0]
Byte	01	16-bit	LH	X	X	R[7:0]	X
Byte	10	16-bit	HL	X	X	X	R[7:0]
Byte	11	16-bit	LH	X	X	R[7:0]	X
Word(1/2)	00	8-bit	LL	X	X	R[15:8]	R[7:0]
Word(2/2)	10	8-bit	LL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	01	8-bit	HL	X	X	X	R[15:8]
Triple Byte(2/2)	10	8-bit	LL	X	X	R[31:24]	R[23:16]
Triple Byte(1/2)	00	8-bit	LL	X	X	R[15:8]	R[7:0]
Triple Byte(2/2)	10	8-bit	HL	X	X	X	R[23:16]
Half-Word	00	8-bit	LL	X	X	R[15:8]	R[7:0]
Half-Word	10	8-bit	LL	X	X	R[15:8]	R[7:0]
Byte	00	8-bit	HL	X	X	X	R[7:0]
Byte	01	8-bit	HL	X	X	X	R[7:0]
Byte	10	8-bit	HL	X	X	X	R[7:0]
Byte	11	8-bit	HL	X	X	X	R[7:0]

*1) selected by Memory Configuration 3 Register PORT8SEL

Table 4.3.6 Access Mapping (SDRAM in Big-Endian Mode)

Access	HA[1:0]	Data Bus Width (*1)	DQMHb, DQMLb	Db[31:24]	Db[23:16]	Db[15:8]	Db[7:0]
Word(1/2)	00	16-bit	LL	X	X	R[31:24]	R[23:16]
Word(2/2)	10	16-bit	LL	X	X	R[15:8]	R[7:0]
Triple Byte(1/2)	00	16-bit	LL	X	X	R[31:24]	R[23:16]
Triple Byte(2/2)	10	16-bit	LH	X	X	R[15:8]	X
Triple Byte(1/2)	01	16-bit	HL	X	X	X	R[23:16]
Triple Byte(2/2)	10	16-bit	LL	X	X	R[15:8]	R[7:0]
Half-Word	00	16-bit	LL	X	X	R[15:8]	R[7:0]
Half-Word	10	16-bit	LL	X	X	R[15:8]	R[7:0]
Byte	00	16-bit	LH	X	X	R[7:0]	X
Byte	01	16-bit	HL	X	X	X	R[7:0]
Byte	10	16-bit	LH	X	X	R[7:0]	X
Byte	11	16-bit	HL	X	X	X	R[7:0]
Word(1/4)	00	8-bit	HL	X	X	X	R[31:24]
Word(2/4)	01	8-bit	HL	X	X	X	R[23:16]
Word(3/4)	10	8-bit	HL	X	X	X	R[15:8]
Word(4/4)	11	8-bit	HL	X	X	X	R[7:0]
Triple Byte(1/3)	00	8-bit	HL	X	X	X	R[31:24]
Triple Byte(2/3)	01	8-bit	HL	X	X	X	R[23:16]
Triple Byte(3/3)	10	8-bit	HL	X	X	X	R[15:7]
Triple Byte(1/3)	01	8-bit	HL	X	X	X	R[23:16]
Triple Byte(2/3)	10	8-bit	HL	X	X	X	R[15:8]
Triple Byte(3/3)	11	8-bit	HL	X	X	X	R[7:0]
Half-Word(1/2)	00	8-bit	HL	X	X	X	R[15:8]
Half-Word(2/2)	01	8-bit	HL	X	X	X	R[7:0]
Half-Word(1/2)	10	8-bit	HL	X	X	X	R[15:8]
Half-Word(2/2)	11	8-bit	HL	X	X	X	R[7:0]
Byte	00	8-bit	HL	X	X	X	R[7:0]
Byte	01	8-bit	HL	X	X	X	R[7:0]
Byte	10	8-bit	HL	X	X	X	R[7:0]
Byte	11	8-bit	HL	X	X	X	R[7:0]

*1) selected by Memory Configuration 0 Register BANKxCONF[1:0].

Table 4.3.7 Access Mapping (DRAM/HDRAM in Big-Endian Mode)

Access	HA[1:0]	Data Bus Width (*1)	CASb[3:0]* (*2)	Db[31:24]	Db[23:16]	Db[15:8]	Db[7:0]
Word	00	32-bit	LLLL	R[31:24]	R[23:16]	R[15:8]	R[7:0]
Triple Byte	00	32-bit	LLLH	R[31:24]	R[23:16]	R[15:8]	X
Triple Byte	01	32-bit	HLLL	X	R[23:16]	R[15:8]	R[7:0]
Half-Word	00	32-bit	LLHH	R[15:8]	R[7:0]	X	X
Half-Word	10	32-bit	HHLL	X	X	R[15:8]	R[7:0]
Byte	00	32-bit	LHHH	R[7:0]	X	X	X
Byte	01	32-bit	HLHH	X	R[7:0]	X	X
Byte	10	32-bit	HHLH	X	X	R[7:0]	X
Byte	11	32-bit	HHHL	X	X	X	R[7:0]
Word(1/2)	00	16-bit	HHLL	X	X	R[31:24]	R[23:16]
Word(2/2)	10	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Triple Byte(1/2)	00	16-bit	HHLL	X	X	R[31:24]	R[23:16]
Triple Byte(2/2)	10	16-bit	HHLH	X	X	R[15:8]	X
Triple Byte(1/2)	01	16-bit	HHHL	X	X	X	R[23:16]
Triple Byte(2/2)	10	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Half-Word	00	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Half-Word	10	16-bit	HHLL	X	X	R[15:8]	R[7:0]
Byte	00	16-bit	HHLH	X	X	R[7:0]	X
Byte	01	16-bit	HHHL	X	X	X	R[7:0]
Byte	10	16-bit	HHLH	X	X	R[7:0]	X
Byte	11	16-bit	HHHL	X	X	X	R[7:0]

*1) selected by Memory Configuration 0 Register BANKxCONF[1:0].

*2) asserted for both read access and write access.

Table 4.3.8 Access Mapping (CSn and MCSn in Big-Endian Mode)

Access	HA[1:0]	Port Size (*1)	CASb[3:0]* (*2)	Db[31:24]	Db[23:16]	Db[15:8]	Db[7:0]
Word	00	32-bit	LLLL	R[31:24]	R[23:16]	R[15:8]	R[7:0]
Triple Byte	00	32-bit	LLLH	R[31:24]	R[23:16]	R[15:8]	X
Triple Byte	01	32-bit	HLLL	X	R[23:16]	R[15:8]	R[7:0]
Half-Word	00	32-bit	LLHH	R[15:8]	R[7:0]	X	X
Half-Word	10	32-bit	HHLL	X	X	R[15:8]	R[7:0]
Byte	00	32-bit	LHHH	R[7:0]	X	X	X
Byte	01	32-bit	HLHH	X	R[7:0]	X	X
Byte	10	32-bit	HHLH	X	X	R[7:0]	X
Byte	11	32-bit	HHHL	X	X	X	R[7:0]
Word(1/2)	00	16-bit	LLHH	R[31:24]	R[23:16]	X	X
Word(2/2)	10	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Triple Byte(1/2)	00	16-bit	LLHH	R[31:24]	R[23:16]	X	X
Triple Byte(2/2)	10	16-bit	LHHH	R[15:8]	X	X	X
Triple Byte(1/2)	01	16-bit	HLHH	X	R[23:16]	X	X
Triple Byte(2/2)	10	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Half-Word	00	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Half-Word	10	16-bit	LLHH	R[15:8]	R[7:0]	X	X
Byte	00	16-bit	LHHH	R[7:0]	X	X	X
Byte	01	16-bit	HLHH	X	R[7:0]	X	X
Byte	10	16-bit	LHHH	R[7:0]	X	X	X
Byte	11	16-bit	HLHH	X	R[7:0]	X	X

1) selected by Memory Configuration 0 Register CSχSIZE (CSn).
Port size for MCSχ* are fixed to 16 bit width.

*2) asserted for write access only.

Table 4.3.9 Access Mapping (PCMCIA in Big-Endian Mode)

Access	HA[1:0]	Port Size (*1)	CARDCSnL*, CARDCSnH*	Db[31:24]	Db[23:16]	Db[15:8]	Db[7:0]
Word(1/2)	00	16-bit	LL	R[31:24]	R[23:16]	X	X
Word(2/2)	10	16-bit	LL	R[15:8]	R[7:0]	X	X
Triple Byte(1/2)	00	16-bit	LL	R[31:24]	R[23:16]	X	X
Triple Byte(2/2)	10	16-bit	LH	R[15:8]	X	X	X
Triple Byte(1/2)	01	16-bit	HL	X	R[23:16]	X	X
Triple Byte(2/2)	10	16-bit	LL	R[15:8]	R[7:0]	X	X
Half-Word	00	16-bit	LL	R[15:8]	R[7:0]	X	X
Half-Word	10	16-bit	LL	R[15:8]	R[7:0]	X	X
Byte	00	16-bit	LH	R[7:0]	X	X	X
Byte	01	16-bit	HL	X	R[7:0]	X	X
Byte	10	16-bit	LH	R[7:0]	X	X	X
Byte	11	16-bit	HL	X	R[7:0]	X	X
Word(1/2)	00	8-bit	LL	R[31:24]	R[23:16]	X	X
Word(2/2)	10	8-bit	LL	R[15:8]	R[7:0]	X	X
Triple Byte(1/2)	00	8-bit	LL	R[31:24]	R[23:16]	X	X
Triple Byte(2/2)	10	8-bit	LH	R[15:8]	X	X	X
Triple Byte(1/2)	01	8-bit	LH	R[23:16]	X	X	X
Triple Byte(2/2)	10	8-bit	LL	R[15:8]	R[7:0]	X	X
Half-Word	00	8-bit	LL	R[15:8]	R[7:0]	X	X
Half-Word	10	8-bit	LL	R[15:8]	R[7:0]	X	X
Byte	00	8-bit	LH	R[7:0]	X	X	X
Byte	01	8-bit	LH	R[7:0]	X	X	X
Byte	10	8-bit	LH	R[7:0]	X	X	X
Byte	11	8-bit	LH	R[7:0]	X	X	X

*1) selected by Memory Configuration 3 Register PORT8SEL

4.3.3 CS3-CS0 and MCS3-MCS0 Timing

Figure 4.3.1 through Figure 4.3.4 show the timing for CS3-CS0 and MCS3-MCS0. The timing is the same for all these signals, with the only exception being that 32-bit port timing only applies to CS3-CS0 while 16-bit port timing applies to CS3-CS0 and MCS3-MCS0.

Timing is also shown for Read Page Mode accesses. If Read Page Mode is enabled, the chip select will be asserted for the entire read cycle. If the processor is performing a burst read to re-fill either a data or instruction cache line, the chip select will remain asserted for the entire burst read. The first access of a Read Page Mode access uses the ACCVAL1 control bits to define the access time, while the next three accesses will use the ACCVAL2 control bits to define the access time. The pattern will repeat until the read is complete.

Figure 4.3.1 shows the timing for a single Word read and write for a 32-bit memory device. The access time is defined using the ACCVAL1 control bits. There are separate ACCVAL1 control bits for each of the chip selects, such that each chip select has an independently configurable access time. The ACCVAL1 bits should be set to the desired number of CLK times for the chip select. The ACCVAL1 bits must not be set to $0.

Figure 4.3.2 shows the timing for a Word read from a 16-bit port. The access is split into two parts to read the full 32 bits. During the first read HA (1) will be low and during the second read HA (1) will be high. The input data path will latch both accesses and concatenate them together as 32 bits for the processor before asserting the Ack signal to the CPU Interface logic.

Figure 4.3.3 shows a 4-Word burst read from a 32-bit port. The ALE signal is only asserted at the beginning of the cycle since the upper address bits will not be changed during the burst. The burst will always start on a 16-byte boundary. The address bits will increment as follows: HA[3:2] = 00, HA[3:2] = 01, HA[3:2] = 10, then HA[3:2] = 11.

Figure 4.3.4 shows the same 4-Word burst read from a 32-bit port, except this time Read Page Mode is enabled. In this case the chip select will remain asserted for the entire access and the addresses will change to access the new address location. The first access uses the ACCVAL1 control bits, while the next three accesses use the ACCVAL2 control bits. If the burst had been longer then 4 Words, the ACCVAL1 control bits would have been used for every fourth access. This type of bursting is compatible with ROM devices that support page mode access. If the 4-Word burst would have been from a 16-bit port, eight accesses would have been required, but the ACCVAL1 bits would still be used for the first and fifth access, while the ACCVAL2 bits would have been used for the rest. If Read Page Mode is enabled for a particular chip select, then the ACCVAL1 and ACCVAL2 control bits must not be set to $0. The “up arrow” markings on DCLKOUT in Figure 4.3.1 through Figure 4.3.4 show the points where input data is sampled.

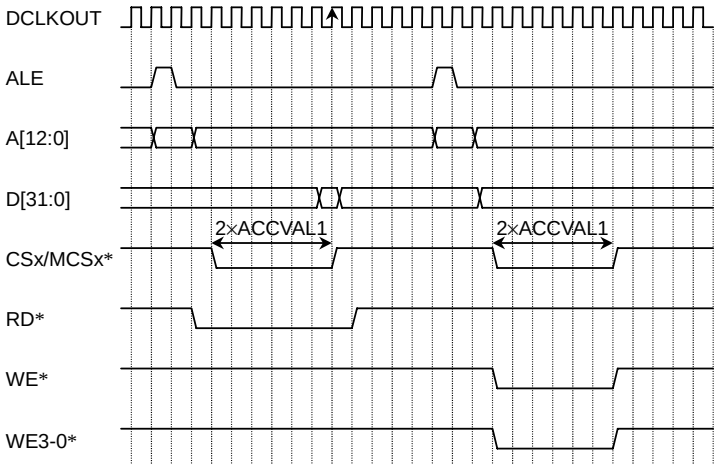


Figure 4.3.1 Word Read/Write 32-Bit Port

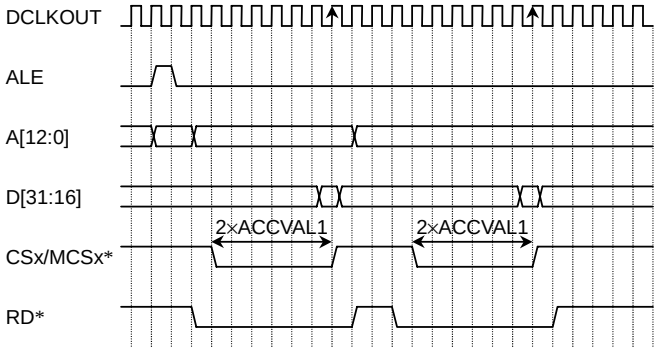


Figure 4.3.2 Word Read 16-Bit Port

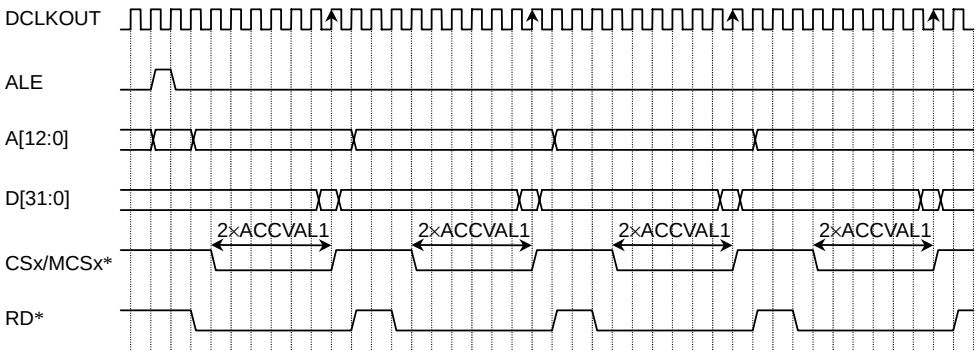


Figure 4.3.3 4-Word Burst Read 32-Bit Port

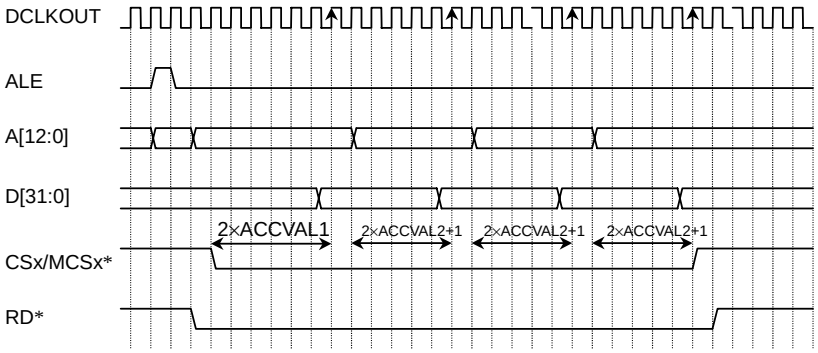


Figure 4.3.4 4-Word Burst Read 32-Bit Port Read with Page Mode Enabled

4.3.4 Card 2 and Card 1

There are three different types of accesses to the Card: Memory, Attribute and I/O. Memory space is accessed by reading or writing from or to the Card Memory locations. Attribute or IO space is accessed by reading or writing from or to the Card IO Attribute* locations. If an Attribute access is desired, the CARDxIOEN (χ = 1, 2) bit should be cleared. If an IO access is desired, the CARDxIOEN bit should be set.

To support bi-directional buffering of the data bus to the Cards, the CARDDIR* signal provides direction control for the bi-directional buffer. The CARDDIR* signal will be asserted during a read to either Card 2 or Card 1.

There is separate access time control provided for Memory access versus IO Attribute* access. CARDxACCVAL is used to control the access time for memory accesses and CARDxIOACCVAL is used to control the access time for IO or Attribute access. Adding one to either access value control setting will increase the access time by two CLK periods. A value of $1 will result in an access time of 1 CLK period. Figure 4.3.5 through Figure 4.3.7 show the different timing for Memory, Attribute and IO accesses. If a Word access is made to the Card, the access will be split into two accesses in the same manner as is done for CS3-CS0 and MCS3-MCS0. Read Page Mode is not supported for the Cards since it is not supported by the PCMCIA specification.

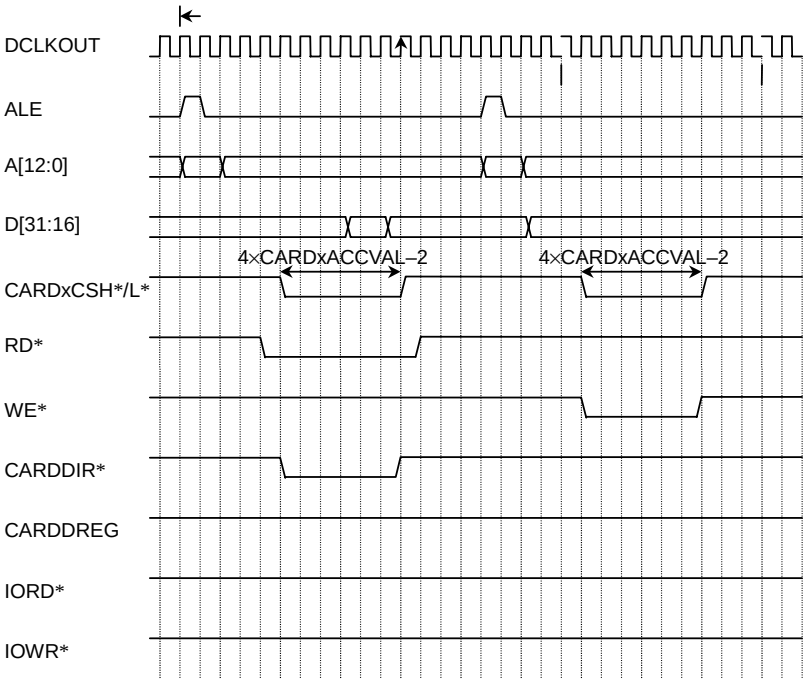


Figure 4.3.5 Memory Access to Card

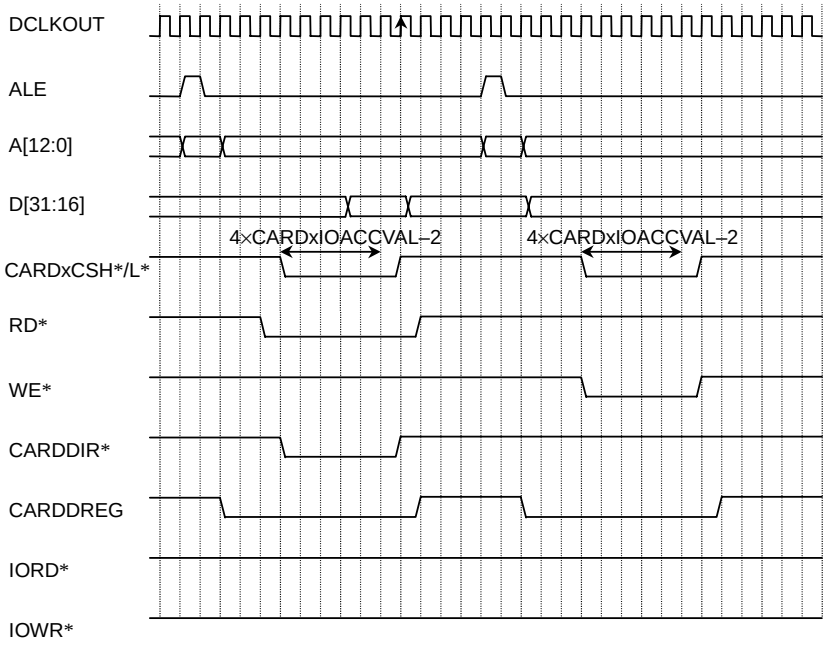


Figure 4.3.6 Attribute Access to Card

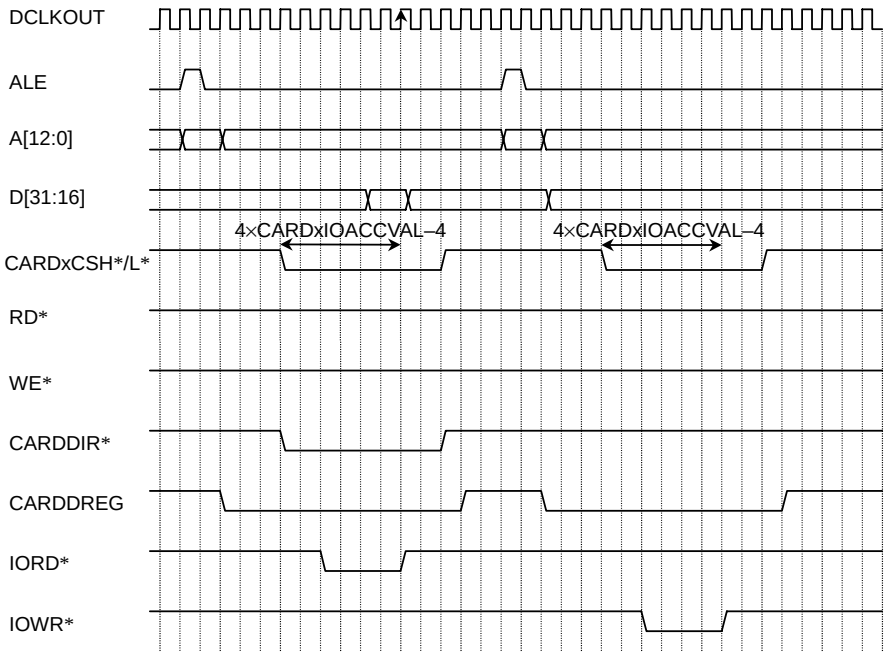


Figure 4.3.7 I/O Access to Card

The PCMCIA specification provides for a WAIT* signal that allows the Card to halt the bus cycle until the WAIT* signal is de-asserted. This allows the Card to control the access time. The WAIT* signal will assert at the beginning of the cycle and the cycle should not terminate until the WAIT* signal becomes de-asserted. There are separate WAIT* signals provided for each Card: WAIT2* and WAIT1*. The WAIT* signals are sampled on two consecutive rising edges of CLK. If the WAIT* signal is low, the access time counter will freeze until the WAIT* signal goes high. This function will only occur if the WAIT function is enabled using the CARDxWAITEN control bit. If the WAIT function is not enabled, the WAIT* signal is ignored. Figure 4.3.6 shows the timing for Card WAIT accesses.

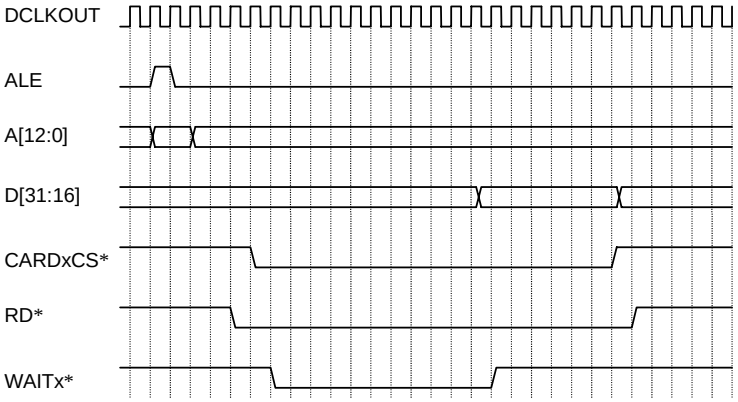


Figure 4.3.8 Card Wait Timing

4.3.5 CS0 Size Configuration

In order for the system to boot properly when powered up for the first time, it is necessary that CS0 and it's associated control be initialized properly since this chip select is used for the Boot ROM. When RESET is asserted, the ACCVAL1 and ACCVAL2 control bits for CS0 will be set to their maximum value and Read Page Mode will be disabled. The port size, 16-bit or 32-bit, is configured using the TESTAIU input pin. If the TESTAIU pin is high during RESET, CS0 is configured as a 32-bit port, and if TESTAIU is low during RESET, CS0 is configured as a 16-bit port. The CPU Module Section discusses the TESTAIU pin in more detail.

4.4 DRAM and SDRAM Controller

The DRAM and SDRAM Controller generates the required control signals to interface to DRAM and SDRAM devices. Two separate banks are supported and each bank can contain either DRAM or SDRAM, with the only restriction being that Bank 1 cannot contain SDRAM if Bank 0 contains DRAM. The DRAM and SDRAM Controller supports 4-Mbit, 16-Mbit and 64-Mbit devices in various bus configurations. In order to maximize the memory bandwidth, both the DRAM and SDRAM devices can be kept in page mode to complete burst read cycles and in-page write cycles.

4.4.1 Memory Chips Supported

Table 4.4.1 shows the different DRAM and SDRAM chips that are supported by the DRAM and SDRAM Controller. Also shown is the Row and Column configuration for the supported chips. SDRAM chips also contain two banks that are selectable through the address space.

Table 4.4.1 Supported DRAM and SDRAM Chips

Number of Bank	Row Address	Column Address	Memory Size	Memory Type	
–	9	9	256 K × 16	4-Mbit DRAM	
–	10	8	256 K × 16	4-Mbit DRAM	
–	10	9	512 K × 8	4-Mbit DRAM	
–	10	10	1 M × 16	16-Mbit DRAM	
–	11	9	1 M × 16	16-Mbit DRAM	
–	12	8	1 M × 16	16-Mbit DRAM	
–	11	10	2 M × 8	16-Mbit DRAM	
–	12	9	2 M × 8	16-Mbit DRAM	
2	11	9	2 M × 8	16-Mbit SDRAM	
2	11	8	1 M × 16	16-Mbit SDRAM	
–	11	11	4 M × 16	64-Mbit DRAM	
–	12	10	4 M × 16	64-Mbit DRAM	
–	12	11	8 M × 8	64-Mbit DRAM	
2	12	10	8 M × 8	64-Mbit SDRAM	
2	12	9	4 M × 16	64-Mbit SDRAM	
2	13	9	8 M × 8	64-Mbit SDRAM	
4	12	8	4 M × 16	64-Mbit SDRAM	*1
4	12	9	8 M × 16	128-Mbit SDRAM	*1

*1 Four-bank SDRAM

4.4.2 DRAM and SDRAM Configurations

Table 4.4.2 shows the various combinations of DRAM and SDRAM chips that are supported. Bank 1 and Bank 0 are independently programmable so each can contain any of the combinations. The only restriction is that Bank 1 cannot contain SDRAM if Bank 0 contains DRAM.

Table 4.4.2 Supported DRAM and SDRAM Configurations

ROW	COL	Memory Size	Memory Type	# of Chips	
18, 17:9	18, 8:1	256 K × 16	4-Mbit DRAM	1 × 256 K × 16	
20, 19, 17:9	20, 18, 8:1	1 M × 16	16-Mbit DRAM	1 × 1 M × 16	
18, 20, 19, 17:9	8:1	1 M × 16	16-Mbit DRAM	1 × 1 M × 16	
22, 21, 19, 17:9	22, 20, 18, 8:1	4 M × 16	64-Mbit DRAM	1 × 4 M × 16	
19, 17:9	19, 18, 8:2	256 K × 32	4-Mbit DRAM	2 × 256 K × 16	
19, 17:9	20, 18, 8:2	512 K × 32	4-Mbit DRAM	4 × 512 K × 8	
21, 19, 17:9	21, 20, 18, 8:2	1 M × 32	16-Mbit DRAM	2 × 1 M × 16	
18, 20, 19, 17:9	21, 8:2	1 M × 32	16-Mbit DRAM	2 × 1 M × 16	
22, 21, 19, 17:9	22, 20, 18, 8:2	2 M × 32	16-Mbit DRAM	4 × 2 M × 8	
23, 21, 19, 17:9	23, 22, 20, 18, 8:2	4 M × 32	64-Mbit DRAM	2 × 4 M × 16	
23, 21, 19, 17:9	24, 22, 20, 18, 8:2	8 M × 32	64-Mbit DRAM	4 × 8 M × 8	
18, 20, 19, 17:9	18, ϕ, X, 8:0	2 M × 8	16-Mbit SDRAM	1 × 2 M × 8	
22, 18, 20, 19, 17:9	22, X, ϕ, 21, 8:0	8 M × 8	64-Mbit SDRAM	1 × 8 M × 8	
18, 20, 19, 17:9	18, ϕ, X, X, 8:1	1 M × 16	16-Mbit SDRAM	1 × 1 M × 16	
18, 20, 19, 17:9	18, ϕ, X, 21, 8:1	2 M × 16	16-Mbit SDRAM	2 × 2 M × 8	
22, 18, 20, 19, 17:9	22, X, ϕ, X, 21, 8:1	4 M × 16	64-Mbit SDRAM	1 × 4 M × 16	
22, 18, 20, 19, 17:9	22, X, ϕ, 23, 21, 8:1	8 M × 16	64-Mbit SDRAM	2 × 8 M × 8	
22, 18, 20, 19, 17:9	X, ϕ, X, 8:0	8 M × 8	64-Mbit SDRAM	1 × 8 M × 8	
22, 18, 20, 19, 17:9	22, X, ϕ, X, X, 8:1	4 M × 16	64-Mbit SDRAM	1 × 4 M × 16	*1
22, 18, 20, 19, 17:9	22, X, ϕ, 23, X, 8:1	8 M × 16	64-Mbit SDRAM	2 × 8 M × 8	*1, *2

*1 Four-bank SDRAM

*2 When TMPR3911/12 and SDRAM are connected, A[8] and A[9] must be swapped.

4.4.3 DRAM

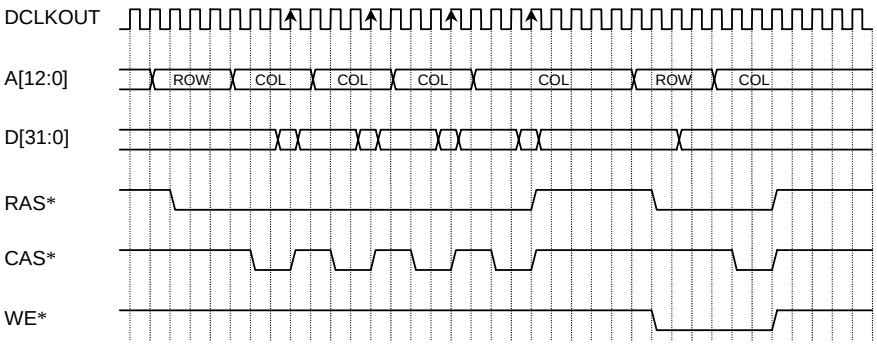
(1) DRAM

DRAM in Bank 1 is controlled using RAS1* and CAS3*-CAS0*, while DRAM in Bank 0 is controlled using RAS0* and CAS3*-CAS0*. The Memory Configuration 0 Register provides control bits that allow Bank 1 or Bank 0 to be independently set as either 32-bit or 16-bit DRAM configurations. The 32-bit DRAM configurations are connected to all 32 data bus bits D[31:0], while 16-bit DRAM configurations are connected to D[15:0] when the TMPR3912 is mapped as Big Endian. The 16-bit DRAM configurations will only use CAS1* and CAS0* for control, while 32-bit configurations will use all four CAS* lines. Word and Tri accesses to 16-bit configured DRAMs will be split into two accesses in the same manner as was done for the chip selects, except that the data is connected to D[15:0] instead of D[31:16]. Connecting DRAM devices to the lower data bus and chip select memory devices to the upper data bus helps to balance the capacitive loading on the data bus.

The DRAM timing is fixed to support 80 ns devices or faster. The only programmability is the ENBANK1OPT and ENBANK0OPT control bits in the Memory Configuration 4 Register that allow an extra clock of time for the first access to the DRAM. This option is implemented to support 80 ns DRAM devices with a clock greater than 70 MHz for the CPU and BIU. DRAMs faster than 80 ns will not need the ENBANKxOPT bit to be set.
(DRAMs faster than 160 ns will not need the ENBANKxOPT bit to be set in case of TMPR3911.)

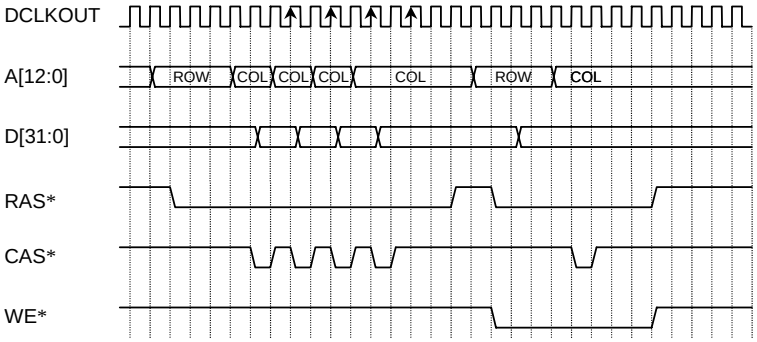
During a burst read cycle the RASx* line will be asserted to latch the ROW address into the DRAM, and the CAS* signals are then asserted for the required number of accesses. Page mode writes are supported when the ENWRINPAGE control bit is set in the Memory Configuration 0 Register. An in-page write will occur when the CPU Interface asserts the WrInPage signal during a write. If this signal is asserted, the DRAM will be kept in page mode until the WrInPage signal is no longer asserted.

Refresh is inserted using CAS* before RAS* refresh. The Refresh Control is generated by the Arbitration Logic. Figure 4.4.1 through Figure 4.4.4 show various timing examples, with refresh timing shown in Figure 4.4.1. The timing is the same for 32-bit and 16-bit ports, except there are twice as many accesses to the 16-bit port. For example, a Word read from a 16-bit port will result in a burst of two 16-bit reads to fetch the 32-bits of data.



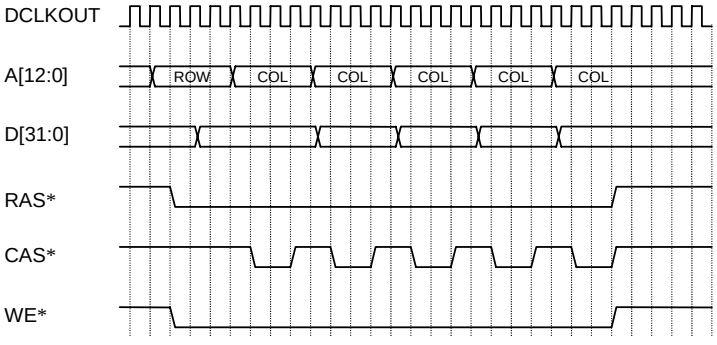
ENBANKxOPT = 0
ENBANKxHDRAM = 0

Figure 4.4.1 Burst Read followed by Word Write (DRAM)



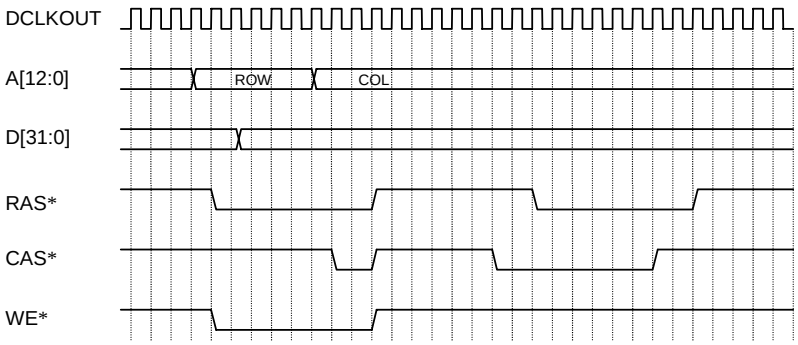
ENBANKxOPT = 0
ENBANKxHDRAM = 1

Figure 4.4.2 Burst Read followed by Word Write (HDRAM)



ENBANKxOPT = 0
ENBANKxHDRAM = 0

Figure 4.4.3 In-Page Write (DRAM)



ENBANKxOPT = 1
ENBANKxHDRAM = 0

Figure 4.4.4 Write with ENBANKOPT Set followed by Refresh Cycle (DRAM)

(2) DRAM Initialization

When the DRAM is first powered up there are several things that must be done before accessing the DRAM. First the Memory Configuration Registers must be properly configured for the chosen DRAM and bus configuration. Next, after a delay of at least 100 μs after the power is asserted to the DRAM, the refresh should be enabled. The DRAM should not be accessed until at least eight CAS* before RAS* refresh cycles occur. After this procedure is completed the DRAM can be accessed.

4.4.4 SDRAM

(1) SDRAM

SDRAM devices are supported in Bank 1 and Bank 0. Each Bank can independently be configured to support either 16-bit or 8-bit SDRAMs. Bank 1 uses RAS1* as the chip select for the SDRAM and Bank 0 uses DCS0*. All other control signals are shared between the two banks.

The clock for the SDRAM devices is the DCLKOUT signal which nominally runs at the clock twice the rate of the CLK signal. Reads from the SDRAMs occur on every DCLKOUT cycle, such that the reading of all 32 bits for 16-bit SDRAMs can happen within a single CLK cycle. A 16-bit SDRAM configuration provides the maximum memory bandwidth and once a read cycle begins, the CPU will not stall since 32 bits are read on each cycle. All 8-bit SDRAMs require 2 CLK cycles to read 32 bits of data. During a burst read the SDRAM devices are kept in page mode for the entire read cycle.

The SDRAM devices are also kept in page mode during in-page write cycles. All 16-bit SDRAMs can keep up at 32 bits per CLK cycle, which can provide a long write block without stalling the CPU. All 8-bit SDRAMs require twice the number of cycles. Tri, Half-Word and Byte writes are supported to the SDRAM devices using the DQMH and DQML signals. These signals provide a data mask so that only the desired bytes will be written. DQMH is used for the upper 8 bits in a 16-bit SDRAM configuration and DQML is used for the lower 8 bits. For 8-bit SDRAMs only DQML is used. The data mask signals DQMH and DQML are required because the SDRAM is setup to always read or write 32 bits of data at a time. The data mask signals will “mask out” the bytes that should not be written during partial Word stores.

(2) SDRAM Initialization

Prior to accessing an SDRAM device, an initialization procedure must be followed. First, all Memory Configuration Registers must be properly initialized for the chosen SDRAM and bus configuration. Next, after a pause of at least 200 μs after power is applied to the SDRAM, a command must be sent to precharge the SDRAM banks. The command is sent by writing to the SDRAM Bank 1 and/or Bank 0 Mode Register address with Data[31] set high when the TMPR3911/12 is mapped as big endian, or with Data[7] set high when the TMPR3911/12 is mapped as little endian. Setting this bit will cause a Precharge All command to be sent to the SDRAM instead of writing to the Mode Register. Next, the Mode Register must be properly initialized. Finally, refresh should be enabled and at least 8 refresh cycles must occur prior to reading or writing from or to the SDRAM.

(3) SDRAM Mode Register

An SDRAM contains a write-only Mode Register that defines the parameters for accessing the device. The mode register must be initialized prior to reading or writing from the SDRAM. They can be written via the SDRAM Bank 1 and SDRAM Bank 0 addresses as shown in the System Address Map. When writing to the mode register, the data from the CPU Interface on Data[12:0] (Data[20:16, 31:24]) is placed on A[12:0], and Data[31] (Data[7]) must be a zero when the TMPR3911/12 is mapped as big endian (little endian). This is done because the Mode Register receives its data on the address bus instead of the data bus. The SDRAM should be configured for the following settings: Burst Length = 4 (for 8-bit SDRAM) or Burst Length = 2 (for 16-bit SDRAM), CAS Latency = 3, Burst Type = Sequential, Write Mode = Burst Write.

(4) SDRAM Power Down Mode

The DCKE signal is provided for putting the SDRAM into Power Down Mode. The Power Down Mode feature can be enabled or disabled using the ENSDRAMPD control bit in the Memory Configuration 0 Register. When this bit is set, the DCKE signal will be held low whenever there are no accesses to the SDRAM device for 7 CLK periods. Once a read, write or refresh is initiated to the SDRAM, the DCKE signal is asserted to bring the device out of Power Down Mode. Power Down Mode helps to reduce power dissipation in the SDRAM device when there are no accesses. During Memory Power Down the SDRAM device is placed into self refresh mode for maximum power savings and the DCLKOUT signal.

(5) SDRAM Timing

Figure 4.4.5 through Figure 4.4.7 show a few examples of the SDRAM timing. The first chip select (assertion of CS*) is used to latch the ROW address. Subsequent chip selects will either read or write data, and the final chip select is used to precharge the bank. Note from Table 4.4.2 that address bit 10 for 16-Mbit parts and address bit 11 for 64-Mbit parts is always a zero during the column address. This is done to ensure that auto precharge is never latched during a read or write command and also will cause only the bank that is being accessed to be precharged during the precharge command. Since the burst size of the SDRAMs is always setup to 32 bits, reads will always be 32 bits without regard to the actual number of bytes requested by the CPU. Writes are also 32 bits, but the data mask signals DQMH and DQML will mask out the bytes that should not be written, thus allowing for Triple Byte, Half-Word and Byte writes.

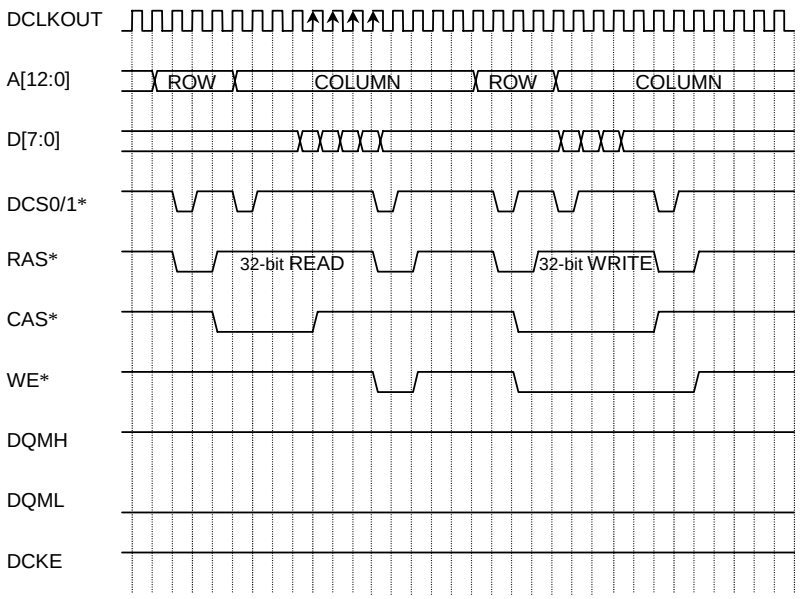


Figure 4.4.5 Word Read and Write to 8-bit SDRAM

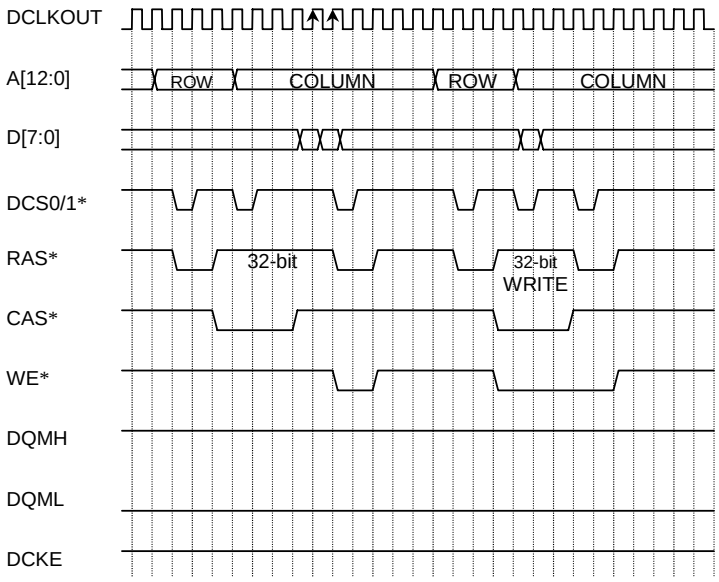


Figure 4.4.6 Word Read and Write to 16-Bit SDRAM

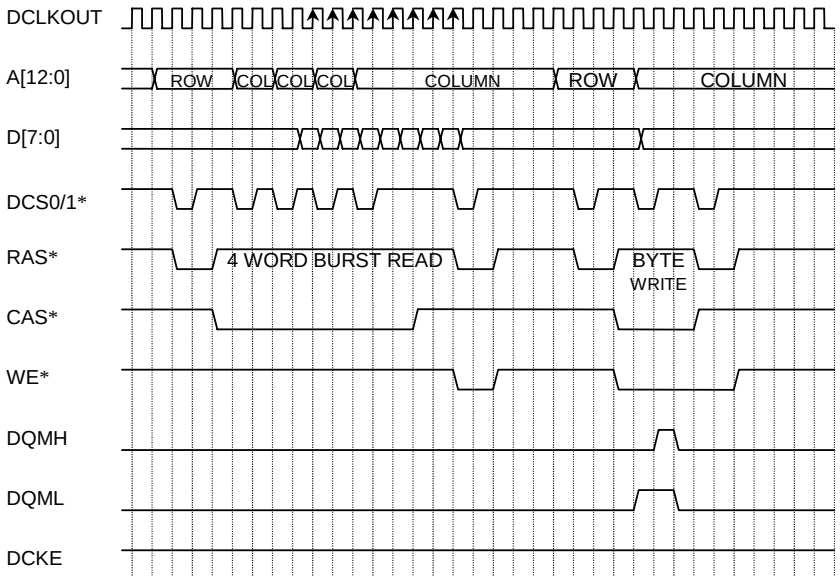


Figure 4.4.7 4-Word Burst Read and Byte Write to 16-Bit SDRAM

(6) Support of Four-Bank SDRAM (only by TMPR3911BU/BXB and TMPR3912XB-92/AU-92)

The TMPR3912XB-92/AU-92 supports four-bank SDRAM. When DRAM banks 0 and 1 are 16-bit SDRAM, Addr[21] is output from the CAS2* pin. Also, by setting ROW and COL as follows, Addr[22] can be output from the A[12] pin.

ROW: 22, 18, 20, 19, 17:9 (ROWSEL = 01)

COL: 22, X, ϕ, X, X, 8:1 (COLSEL = 1000)

The CAS2* pin (Addr[21]) and A[12] pin (Addr[22]) can be connected to the four-bank SDRAM bank select pin so that the CAS2* and A[12] pins are used for the bank select signal.

4.5 Arbitration

The Arbitration logic contains the Refresh Controller for Bank 1 and Bank 0, support for External Bus Master, the Watch Dog Timer for terminating hung memory accesses, and Memory Power Down logic for placing the SDRAM and/or DRAM into self refresh mode.

4.5.1 Refresh Controller

The Refresh Controller logic contains separate refresh counters for Bank 1 and Bank 0. Each counter is a 6-bit counter that uses the 32 kHz clock for the counter. The refresh rate is set using the RFSHVAL1[5:0] and RFSHVAL0[5:0] control bits in the Memory Configuration 4 Register. These values are loaded into the respective refresh counter. The refresh counter will down count and when it reaches a count of zero, refresh will be inserted. If the RFSHVALx control bits are set to $00, the counter is not used and instead refresh is generated based on both the rising and falling edge of the 32 kHz clock. This is required to generate a 15.26 µs refresh rate required by some DRAMs or SDRAMs. RFSHVALx control bits set to $01 will cause a refresh rate of 61.04 µs and additional values will increase the refresh rate by increments of 30.52 µs. When it is time for the Refresh Controller to insert refresh, the Halt signal will be asserted to the CPU Interface. Once the CPU Interface asserts the Halted signal, the Refresh Controller will generate the appropriate control signals that will cause the DRAM & SDRAM control logic to provide refresh for the memory devices. Once the Refresh Controller has completed inserting refresh, the Halt signal will be de-asserted so that the CPU Interface can continue running.

4.5.2 External Bus Master

The External Bus Master logic allows an external device to take control of the memory signals so that memory can be shared with an external device. The external device will request the bus by asserting the DREQ* signal. Once the DREQ* signal is asserted, the Halt signal will be asserted to the CPU Interface logic. Once the CPU Interface logic asserts the Halted signal, the External Bus Master logic will then assert the DGRNT* signal to inform the external device that it now controls the memory bus. At the same time that DGRNT* is asserted, the following signals are tri-stated: D[31:0], A[12:0], ALE, RD*, WE*, CAS3*-CAS0*, RAS1*, RAS0*, DCS0*, DCKE, DQMH, DQML and CS0*. DCLKOUT will be tri-stated if the ENDCLKOUTTRI control bit is set in the Memory Configuration 0 Register. Otherwise, DCLKOUT will continue to run. Once the external device de-asserts DREQ*, DGRNT* will be de-asserted, the memory signals will be driven by the TMPR3911/12, and the Halt signal will be de-asserted so the CPU Interface can begin launching memory transactions. The external device should not attempt to take over the memory signals for a long period of time, otherwise TMPR3911/12 DMA events and refresh will be interrupted. The External Bus Master logic will not respond with a DGRNT* if the memory interface is powered down.

4.5.3 Watch Dog Timer

The Watch Dog Timer prevents accesses to reserved memory locations from hanging the BIU. The Watch Dog Timer consists of a 10-bit down-counter whose initial value is determined by the WATCHTIMEVAL[3:0] control bits in the Memory Configuration 4 Register. The WATCHTIMEVAL[3:0] control bits are loaded into the upper 4 bits of the 10-bit counter and the lower 6 bits are loaded with $3F. The counter uses the CLK signal to count. The Watch Dog Timer rate is determined by the following equation:

$$\text{Watch Dog Timer Rate} = \frac{(\text{WATCHTIME\_VAL}[3:0]+1)*64}{f_{CLK}\ (*1)}$$

*1 f_{CLK} is the frequency of the CLK signal and is ×4 the frequency of SYSCLK (that is, when SYSCLK is 9.216 MHz, f_{CLK} is 36.864 MHz (*2); when SYSCLK is 11.520 MHz, f_{CLK} is 46.08 MHz.

*2 Note that these valve is the case that the RF function is not used. Therefore, In case of TMPR3911, f_{CLK} is a half of this valve, which means 18.432 MHz

The Watch Dog Timer is enabled using the ENWATCH control bit in the Memory Configuration 4 Register. When enabled, the counter will load whenever the CPU Interface asserts the Start signal to begin a memory transaction. If the Watch Dog Timer reaches a count of zero before the Memory State Machine asserts the Ack signal, the BusError signal is asserted to end the memory transaction. There are two possible ways that this can occur. First, if the address provided by the CPU interface to the Address Decoder is pointing to any of the reserved locations or to any of the chip selects that are not enabled, then the Memory State Machine will not respond with an Ack. Secondly, if the WAIT function is enabled to Card 2 or Card 1, then the WAIT2* or WAIT1* signals must de-assert before the Watch Dog Timer reaches a count of zero, otherwise the BusError signal will be asserted and the Card access will be aborted.

4.5.4 Memory Power Down

The memory interface is powered down whenever the MEMPOWERDOWN control bit is set in the Memory Configuration 4 Register. When this bit is set, the Halt signal will be asserted to the CPU Interface, the DRAM and/or SDRAM will be placed into self refresh mode, and all memory interface signals will be driven low. The memory interface will wake up whenever the CPU Interface asserts the HaltCycle signal and the address defined by the Addr[31:2] bits correspond to an address other than internal function registers. When the memory interface wakes up, the DRAM and/or SDRAM are taken out of self refresh mode and the Halt signal is de-asserted to allow the CPU Interface to launch memory transactions. Figure 4.5.1 and Figure 4.5.2 show timing for the wake-up and power-down of the memory interface.

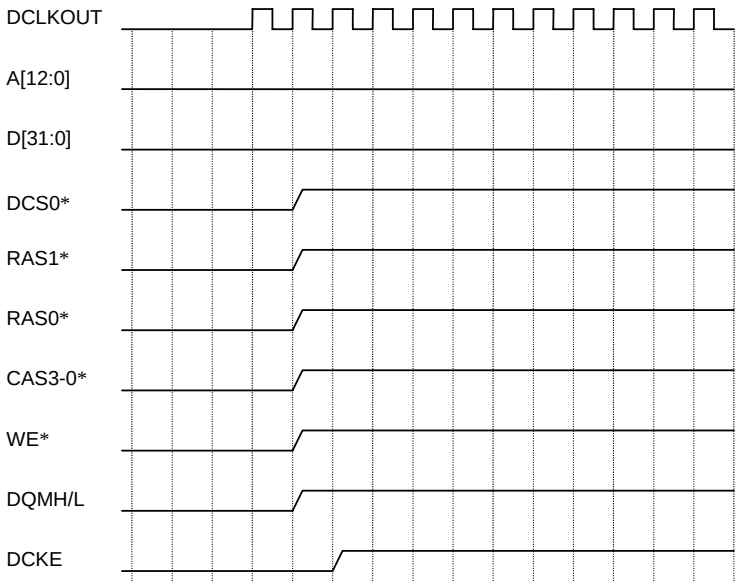


Figure 4.5.1 Memory Interface Wake-Up

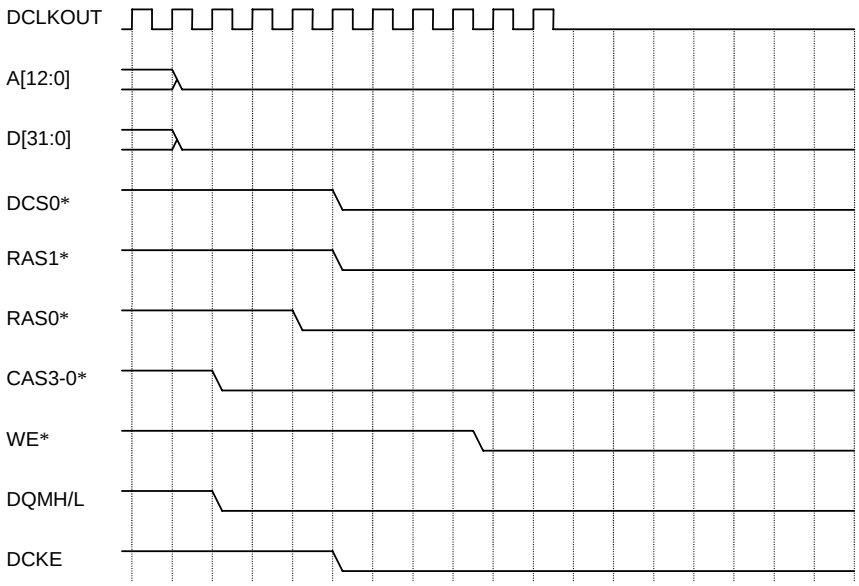


Figure 4.5.2 Memory Interface Power Down

When the memory interface is powered down all memory signals are driven low. The order in which they are driven low is important because DRAM devices enter self refresh mode by asserting CAS* before RAS*, and SDRAM devices enter self refresh mode by de-asserting DCKE while DCS0* and/or RAS1*, RAS0* and CAS0* are asserted with WE* not asserted. Once the SDRAM enters self refresh mode, DCLKOUT can be turned off. The ordering makes it possible to put both DRAM and SDRAM devices into self refresh mode with only one time setting required. When the memory interface comes out of self refresh mode, all signals are de-asserted. DCKE must de-assert one clock later in order to properly bring the SDRAM out of self refresh mode. After exiting self refresh mode, the Halt signal will not be released for several clocks to allow the DRAM and SDRAM to finish exiting self refresh mode.

4.6 Memory Connections

Data Bus pins (D[31:0]), CAS signal pins, and data mask pins (DQMH, DQML) change their functions in Big Endian and Little Endian. The following table shows the change of the name of each pin in Big Endian and Little Endian. (The endian can be selected using the ENDIAN pin, “1” level to the Big Endian, “0” level to the Little Endian.)

Pin No.					
TMPR3911BU	TMPR3911BXB	TMPR3912AU	TMPR3912XB	Big Endian	Little Endian
114	P7	133	H12	D[31]	D[7]
115	N7	134	G10	D[30]	D[6]
116	M6	136	G14	D[29]	D[5]
117	R7	138	G12	D[28]	D[4]
118	N6	139	G11	D[27]	D[3]
120	M5	141	F14	D[26]	D[2]
121	R6	143	F12	D[25]	D[1]
123	P5	145	E14	D[24]	D[0]
124	N4	146	E13	D[23]	D[15]
125	R5	148	D15	D[22]	D[14]
127	N3	150	D13	D[21]	D[13]
128	P3	152	C15	D[20]	D[12]
129	R4	153	C14	D[19]	D[11]
131	R3	155	B14	D[18]	D[10]
133	P1	157	A15	D[17]	D[9]
136	N2	159	E11	D[16]	D[8]
20	C7	27	H1	D[15]	D[23]
19	D7	26	H3	D[14]	D[22]
18	C8	24	G6	D[13]	D[21]
17	A7	23	G1	D[12]	D[20]
16	B6	21	G3	D[11]	D[19]
14	D6	19	F1	D[10]	D[18]
13	A6	18	G5	D[9]	D[17]
12	B5	16	F3	D[8]	D[16]
11	C5	14	F4	D[7]	D[31]
10	B4	12	E3	D[6]	D[30]
9	A5	11	D1	D[5]	D[29]
7	C4	9	D2	D[4]	D[28]
5	A4	7	C1	D[3]	D[27]
4	B3	5	C2	D[2]	D[26]
3	A1	4	B1	D[1]	D[25]
2	A3	2	C3	D[0]	D[24]
166	E3	195	D6	CAS3*	CAS0*
167	E2	197	B5	CAS2*	CAS1*
168	D3	198	C5	CAS1*	CAS2*
169	E1	199	E6	CAS0*	CAS3*
175	C1	207	B2	DQMH	DQML
176	B1	208	A2	DQML	DQMH

Figure 4.6.1 through Figure 4.6.8 illustrate the various signal connections between the TMPR3911/12 and external SDRAM, DRAM, and static memory devices. These figures show the connections for various bus width configurations (e.g., 8-bit, 16-bit, 32-bit).

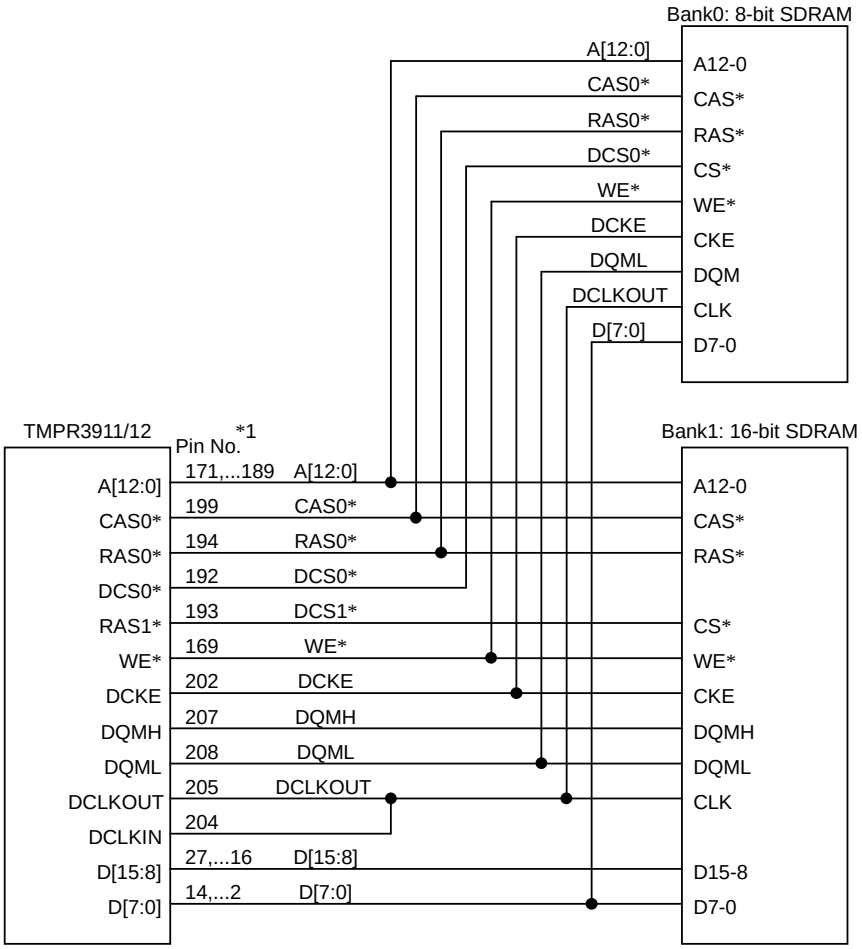


Figure 4.6.1 SDRAM Memory Connection (Big Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

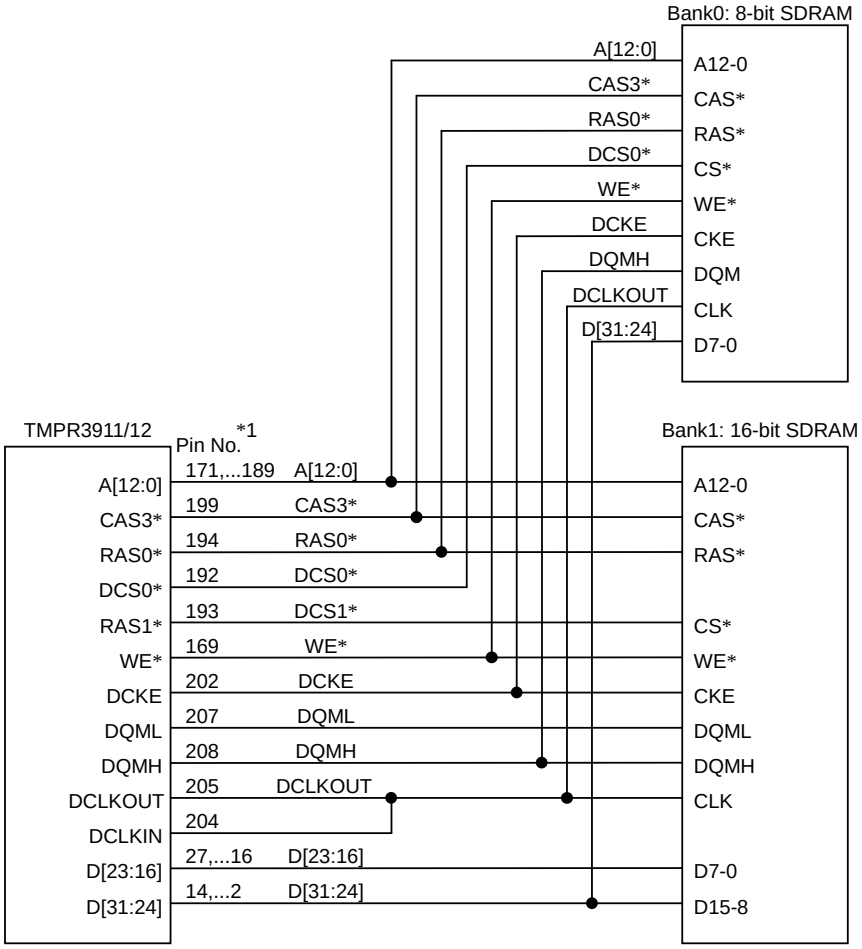


Figure 4.6.2 SDRAM Memory Connection (Little Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

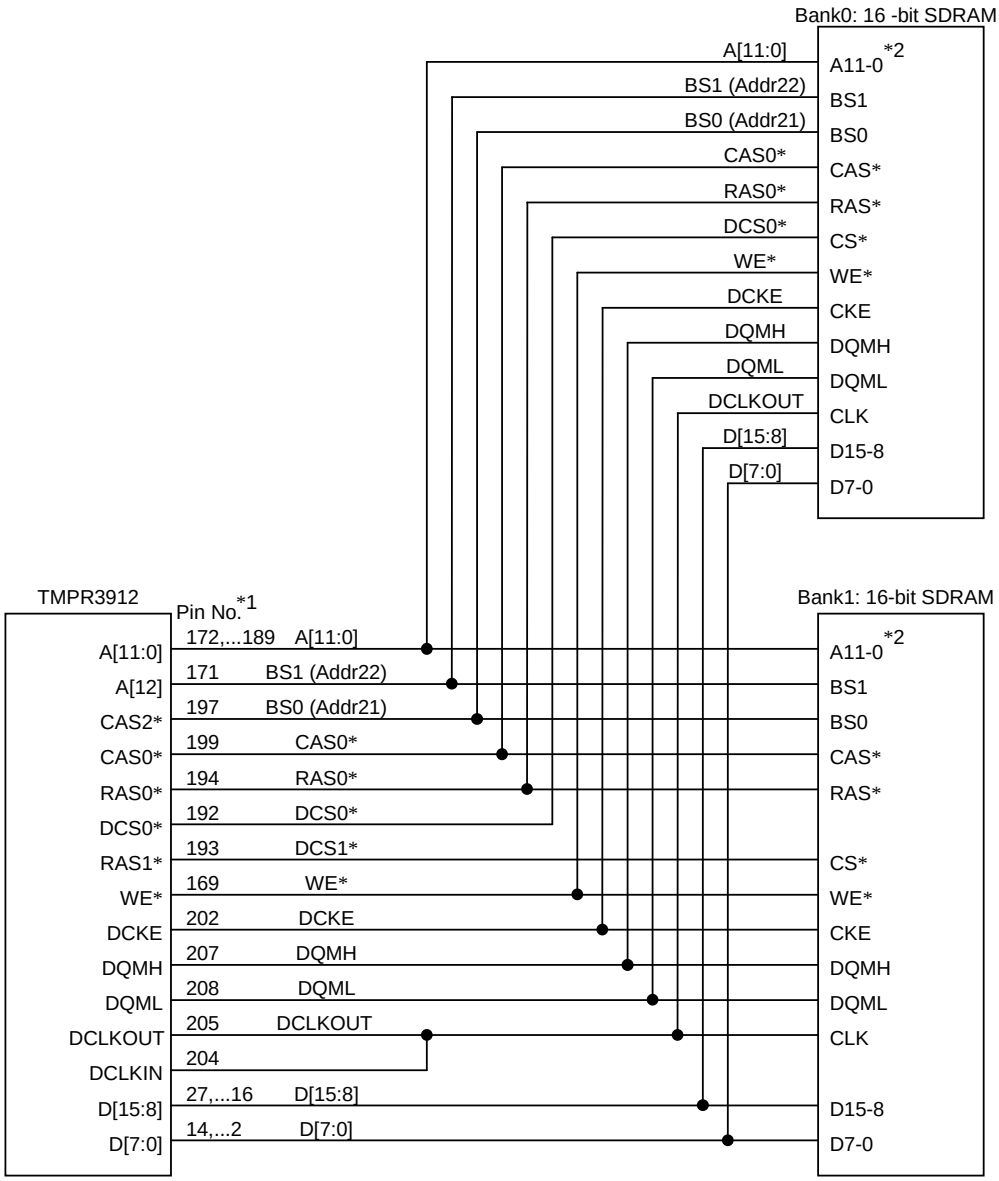


Figure 4.6.3 4-bank SDRAM Memory Connection (Big Endian)*
(supported only by TMPR3912XB-92/AU-92)

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

*2 A[8] and A[9] must be swapped when 128 Mbit SDRAM is used.

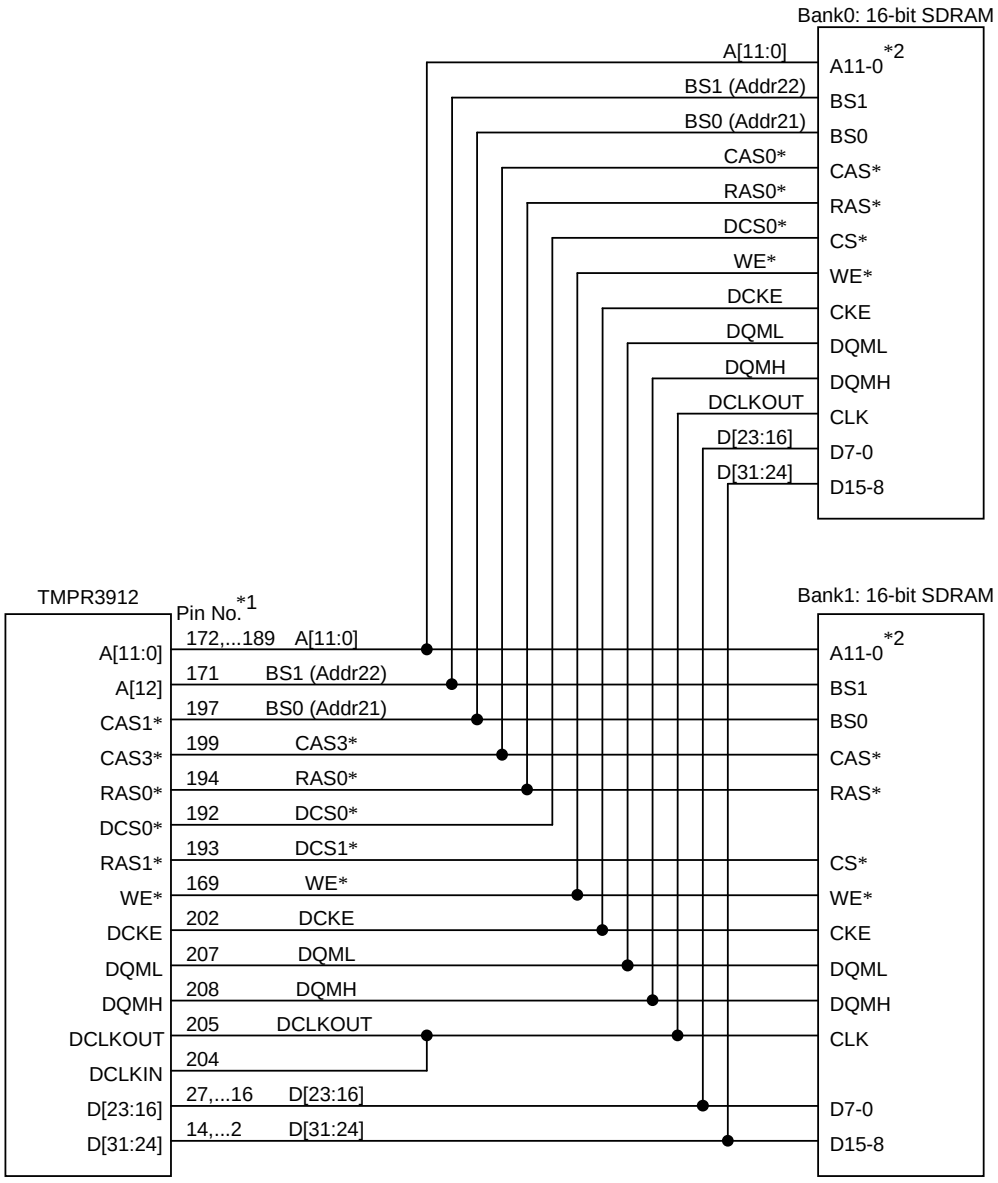


Figure 4.6.4 4-bank SDRAM Memory Connection (Little Endian)*
(supported only by TMPR3912XB-92/AU-92)

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

*2 A[8] and A[9] must be swapped when 128 Mbit SDRAM is used.

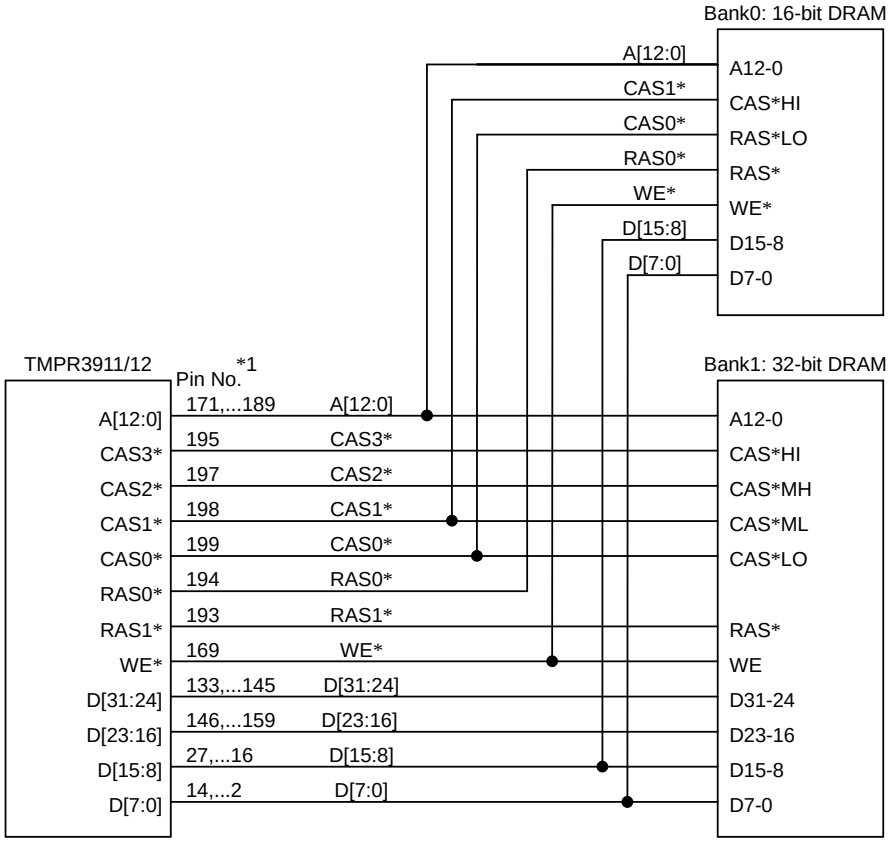


Figure 4.6.5 DRAM Memory Connection (Big Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

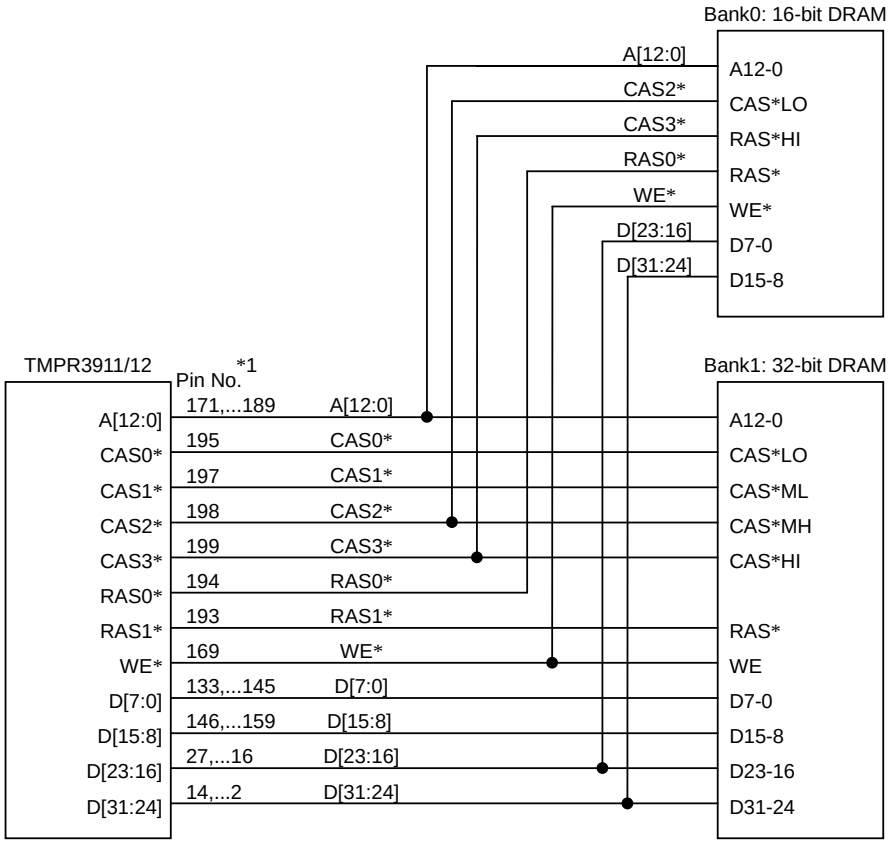


Figure 4.6.6 DRAM Memory Connection (Little Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

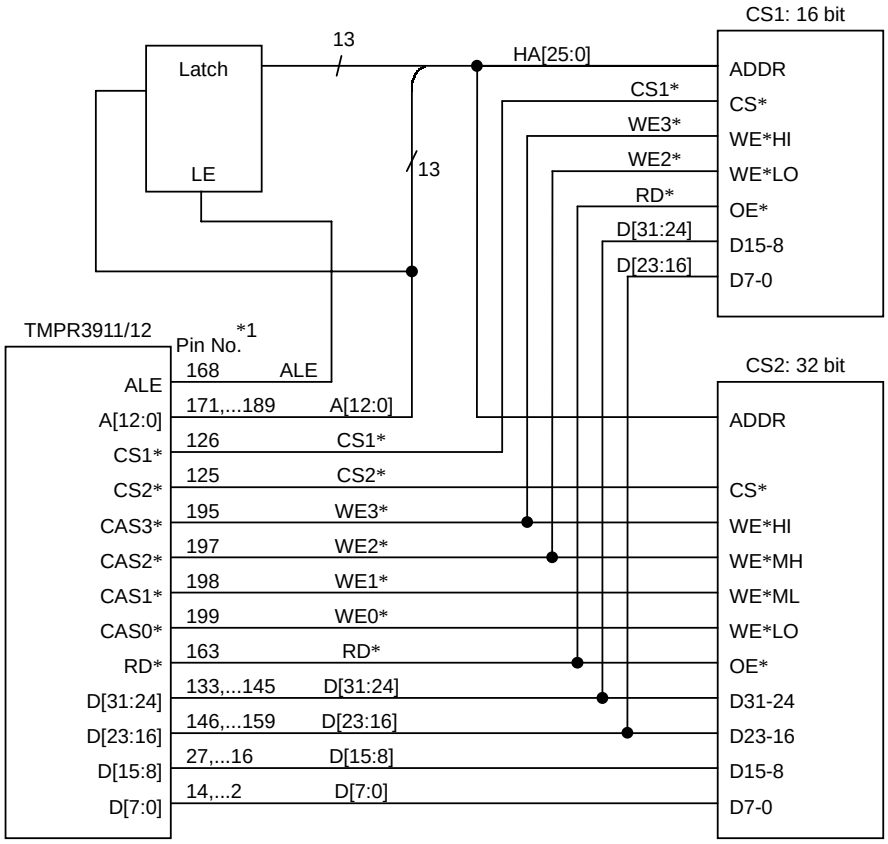


Figure 4.6.7 Chip Select Memory Connection (Big Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

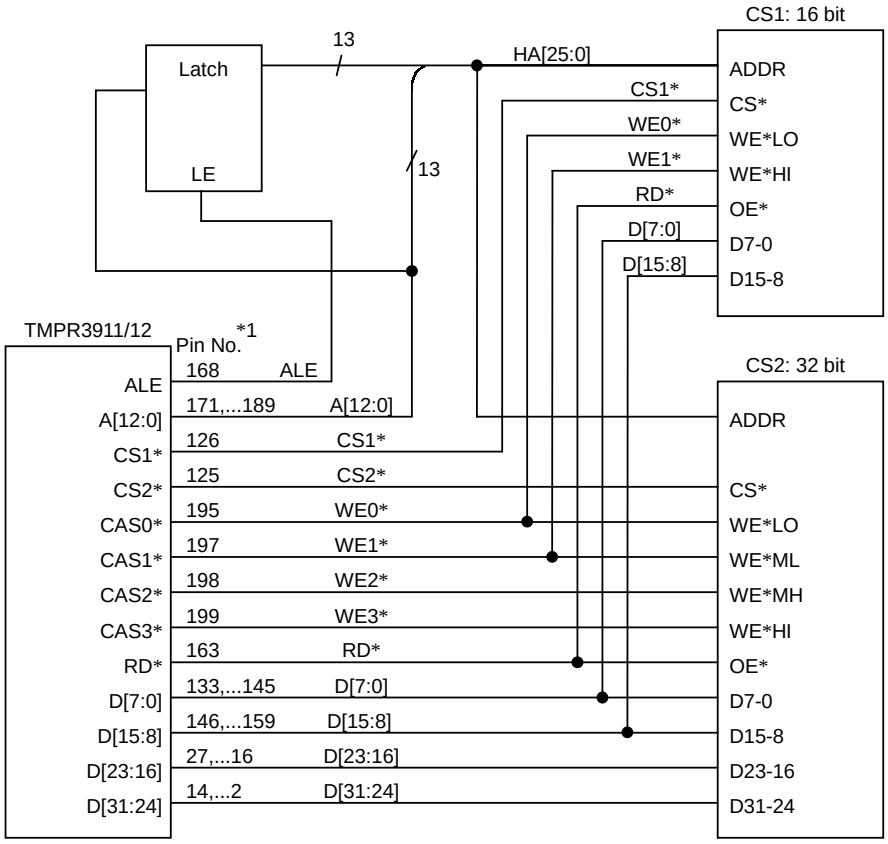


Figure 4.6.8 Chip Select Memory Connection (Little Endian)*

*1 Note that Pin No. in the figure above are those of TMPR3912AU-92 (LQFP208). In case of TMPR3911BU (LQFP176), TMPR3911BXB (FBGA177) and TMPR3912XB-92 (FBGA217), refer to the table in the Section 2.

4.7 BIU Registers

4.7.1 Memory Configuration 0 Register

OFFSET = $000:

Bit	Label	RESET	Read/Write
31	Reserved		
30	ENDCLKOUTTRI	0	R/W
29	DISDQMINIT	0	R/W
28	ENSDRAMPD	0	R/W
27	SHOWDINO	0	R/W
26	ENRMAP2	0	R/W
25	ENRMAP1	0	R/W
24	ENWRINPAGE	0	R/W
23	ENCS3USER	X	R/W
22	ENCS2USER	X	R/W
21	ENCS1USER	X	R/W
20	ENCS1DRAM	X	R/W
19-18	BANK1CONF[1:0]	X	R/W
17-16	BANK0CONF[1:0]	X	R/W
15-14	ROWSEL1[1:0]	X	R/W
13-12	ROWSEL0[1:0]	X	R/W
11-8	COLSEL1[3:0]	X	R/W
7-4	COLSEL0[3:0]	X	R/W
3	CS3SIZE	X	R/W
2	CS2SIZE	X	R/W
1	CS1SIZE	X	R/W
0	CS0SIZE	See Description	R/W

ENDCLKOUTTRI:

Setting this bit will cause the DCLKOUT pin to tri-state when DGRNT* is asserted during external bus arbitration. Otherwise, DCLKOUT will continue to operate.

DISDQMINIT:

If this bit is a cleared, the DQMH and DQML signals will go high when coming out of power-down and remain high until the first access to SDRAM. If this bit is set, this function is disabled and the DQMH and DQML signals will remain low.

ENSDRAMPD:

Setting this bit will cause the BIU to put the SDRAMs into Power Down mode by lowering DCKE whenever there are no accesses to the SDRAMs for 7 CLK periods.

SHOWDINO:

Setting this bit will cause internal reads and writes by the processor to function registers to show externally for debug. The RD* signal provides the read or write status, D[31:0] provide the data, A[12:0] provide the address, and the CARDREG* signal provides a chip select on the rising edge for sampling the signals.

ENRMAP2:

Setting this bit will enable Address Re-Mapper 2.

ENRMAP1:

Setting this bit will enable Address Re-Mapper 1.

ENWRINPAGE:

Setting this bit will cause the DRAMs and/or SDRAMs to remain in page mode when the WrInPage signal is asserted by the CPU Interface. If this bit is cleared, the WrInPage signal is ignored.

ENCS3USER:

Setting this bit will enable CS3 in the kuseg Space.

ENCS2USER:

Setting this bit will enable CS2 in the kuseg Space.

ENCS1USER:

Setting this bit will enable CS1 in the kuseg Space.

ENCS1DRAM:

Setting this bit will enable CS1 in Bank 1 and Bank 0 Space.

BANK1CONF[1:0]:

These bits provide the configuration for Bank 1 according to the following:

11 16-bit SDRAM

10 8-bit SDRAM

01 32-bit DRAM/HDRAM

00 16-bit DRAM/HDRAM

BANK0CONF[1:0]:

These bits provide the configuration for Bank 0 according to the following:

11 16-bit SDRAM

10 8-bit SDRAM

01 32-bit DRAM/HDRAM

00 16-bit DRAM/HDRAM

ROWSEL1[1:0]:

ROWSEL0[1:0]:

COLSEL1[3:0]:

COLSEL0[3:0]:

These bits determine the Row and Column configuration for addressing according to the following:

ROW	ROWSEL
18, 17:9	00
22, 18, 20, 19, 17:9	01
20, 22, 21, 19, 17:9	10
22, 23, 21, 19, 17:9	11

COL	COLSEL
22, 20, 18, 8:1	0000
19, 18, 8:2	0001
21, 20, 18, 8:2	0010
23, 22, 20, 18, 8:2	0011
24, 22, 20, 18, 8:2	0100
18, ϕ, X, 8:0	0101
22, X, ϕ, 21, 8:0	0110
18, ϕ, X, 21, 8:1	0111
22, X, ϕ, 23, 21, 8:1	1000
24, 23, 21, 8:2	1001
Reserved	Others

CS3SIZE:

Setting this bit defines CS3 to be a 32-bit port. Otherwise, a 16-bit port is assumed.

CS2SIZE:

Setting this bit defines CS2 to be a 32-bit port. Otherwise, a 16-bit port is assumed.

CS1SIZE:

Setting this bit defines CS1 to be a 32-bit port. Otherwise, a 16-bit port is assumed.

CS0SIZE:

Setting this bit defines CS0 to be a 32-bit port. Otherwise, a 16-bit port is assumed. On RESET, this signal is set high if the TESTAIU pin is tied high and low if the TESTAIU pin is tied low.

4.7.2 Memory Configuration 1 Register

OFFSET = $004

Bit	Label	RESET	Read/Write
31-28	MCS3ACCVAL1[3:0]	X	R/W
27-24	MCS3ACCVAL2[3:0]	X	R/W
23-20	MCS2ACCVAL1[3:0]	X	R/W
19-16	MCS2ACCVAL2[3:0]	X	R/W
15-12	MCS1ACCVAL1[3:0]	X	R/W
11-8	MCS1ACCVAL2[3:0]	X	R/W
7-4	MCS0ACCVAL1[3:0]	X	R/W
3-0	MCS0ACCVAL2[3:0]	X	R/W

MCS3ACCVAL1[3:0]:

These bits define the access time for MCS3. If ENMCS3PAGE is set, these bits will be used for the first access and MCS3ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS3PAGE is not set, these bits and MCS3ACCVAL2[3:0] should be set with the same value. If ENMCS3PAGE is set, these bits must be set to at least $1.

MCS3ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS3PAGE is set. If ENMCS3PAGE is not set, these bits and MCS3ACCVAL1[3:0] should be set with the same value. If ENMCS3PAGE is set, these bits must be set to at least $2.

MCS2ACCVAL1[3:0]:

These bits define the access time for MCS2. If ENMCS2PAGE is set, these bits will be used for the first access and MCS2ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS2PAGE is not set, these bits and MCS2ACCVAL2[3:0] should be set with the same value. If ENMCS2PAGE is set, these bits must be set to at least $1.

MCS2ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS2PAGE is set. If ENMCS2PAGE is not set, these bits and MCS2ACCVAL1[3:0] should be set with the same value. If ENMCS2PAGE is set, these bits must be set to at least $2.

MCS1ACCVAL1[3:0]:

These bits define the access time for MCS1. If ENMCS1PAGE is set, these bits will be used for the first access and MCS1ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS1PAGE is not set, these bits and MCS1ACCVAL2[3:0] should be set with the same value. If ENMCS1PAGE is set, these bits must be set to at least $1.

MCS1ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS1PAGE is set. If ENMCS1PAGE is not set, these bits and MCS1ACCVAL1[3:0] should be set with the same value. If ENMCS1PAGE is set, these bits must be set to at least $2.

MCS0ACCVAL1[3:0]:

These bits define the access time for MCS0. If ENMCS0PAGE is set, these bits will be used for the first access and MCS0ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS0PAGE is not set, these bits and MCS0ACCVAL2[3:0] should be set with the same value. If ENMCS0PAGE is set, these bits must be set to at least $1.

MCS0ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS0PAGE is set. If ENMCS0PAGE is not set, these bits and MCS0ACCVAL1[3:0] should be set with the same value. If ENMCS0PAGE is set, these bits must be set to at least $2.

4.7.3 Memory Configuration 2 Register

OFFSET = $008

Bit	Label	RESET	Read/Write
31-28	CS3ACCVAL1[3:0]	X	R/W
27-24	CS3ACCVAL2[3:0]	X	R/W
23-20	CS2ACCVAL1[3:0]	X	R/W
19-16	CS2ACCVAL2[3:0]	X	R/W
15-12	CS1ACCVAL1[3:0]	X	R/W
11-8	CS1ACCVAL2[3:0]	X	R/W
7-4	CS0ACCVAL1[3:0]	$F	R/W
3-0	CS0ACCVAL2[3:0]	$F	R/W

CS3ACCVAL1[3:0]:

These bits define the access time for CS3. If ENCS3PAGE is set, these bits will be used for the first access and CS3ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENCS3PAGE is not set, these bits and CS3ACCVAL2[3:0] should be set with the same value. If ENCS3PAGE is set, these bits must be set to at least $1.

CS3ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS3PAGE is set. If ENCS3PAGE is not set, these bits and CS3ACCVAL1[3:0] should be set with the same value. If ENCS3PAGE is set, these bits must be set to at least $2.

CS2ACCVAL1[3:0]:

These bits define the access time for CS2. If ENCS2PAGE is set, these bits will be used for the first access and CS2ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENCS2PAGE is not set, these bits and CS2ACCVAL2[3:0] should be set with the same value. If ENCS2PAGE is set, these bits must be set to at least $1.

CS2ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS2PAGE is set. If ENCS2PAGE is not set, these bits and CS2ACCVAL1[3:0] should be set with the same value. If ENCS2PAGE is set, these bits must be set to at least $2.

CS1ACCVAL1[3:0]:

These bits define the access time for CS1. If ENCS1PAGE is set, these bits will be used for the first access and CS1ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENCS1PAGE is not set, these bits and CS1ACCVAL2[3:0] should be set with the same value. If ENCS1PAGE is set, these bits must be set to at least $1.

CS1ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS1PAGE is set. If ENCS1PAGE is not set, these bits and CS1ACCVAL1[3:0] should be set with the same value. If ENCS1PAGE is set, these bits must be set to at least $2.

CS0ACCVAL1[3:0]:

These bits define the access time for CS0. If ENCS0PAGE is set, these bits will be used for the first access and CS0ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENCS0PAGE is not set, these bits and CS0ACCVAL2[3:0] should be set with the same value. If ENCS0PAGE is set, these bits must be set to at least $1.

CS0ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS0PAGE is set. If ENCS0PAGE is not set, these bits and CS0ACCVAL1[3:0] should be set with the same value. If ENCS0PAGE is set, these bits must be set to at least $2.

4.7.4 Memory Configuration 3 Register

OFFSET = $00C

Bit	Label	RESET	Read/Write
31-28	CARD2ACCVAL[3:0]	X	R/W
27-24	CARD1ACCVAL[3:0]	X	R/W
23-20	CARD2IOACCVAL[3:0]	X	R/W
19-16	CARD1IOACCVAL[3:0]	X	R/W
15	ENMCS3PAGE	X	R/W
14	ENMCS2PAGE	X	R/W
13	ENMCS1PAGE	X	R/W
12	ENMCS0PAGE	X	R/W
11	ENCS3PAGE	X	R/W
10	ENCS2PAGE	X	R/W
9	ENCS1PAGE	X	R/W
8	ENCS0PAGE	0	R/W
7	CARD2WAITEN	X	R/W
6	CARD1WAITEN	X	R/W
5	CARD2IOEN	X	R/W
4	CARD1IOEN	X	R/W
3	PORT8SEL	0	R/W
2	PORT2_8SEL	0	R/W
1	PORT1_8SEL	0	R/W
0	DCLKDISABLE	0	R/W

CARD2ACCVAL[3:0]:

These bits define the access time for Card 2 Memory Space.

CARD1ACCVAL[3:0]:

These bits define the access time for Card 1 Memory Space.

CARD2IOACCVAL[3:0]:

These bits define the access time for Card 2 IO and Attribute Space.

CARD1IOACCVAL[3:0]:

These bits define the access time for Card 1 IO and Attribute Space.

ENMCS3PAGE:

Setting this bit will enable Read Page Mode for MCS3.

ENMCS2PAGE:

Setting this bit will enable Read Page Mode for MCS2.

ENMCS1PAGE:

Setting this bit will enable Read Page Mode for MCS1.

ENMCS0PAGE:

Setting this bit will enable Read Page Mode for MCS0.

ENCS3PAGE:

Setting this bit will enable Read Page Mode for CS3.

ENCS2PAGE:

Setting this bit will enable Read Page Mode for CS2.

ENCS1PAGE:

Setting this bit will enable Read Page Mode for CS1.

ENCS0PAGE:

Setting this bit will enable Read Page Mode for CS0.

CARD2WAITEN:

Setting this bit will enable the CARD2WAIT* signal.

CARD1WAITEN:

Setting this bit will enable the CARD1WAIT* signal.

CARD2IOEN:

Setting this bit will cause accesses to Card 2 I/O space to assert the IORD* or IOWR* signals. This is used for accessing I/O cards. If this bit is not set, RD* or WE* is asserted, which provides Attribute space access.

CARD1IOEN:

Setting this bit will cause accesses to Card 1 I/O space to assert the IORD* or IOWR* signals. This is used for accessing I/O cards. If this bit is not set, RD* or WE* is asserted, which provides Attribute space access.

PORT8SEL, PORT2_8SEL, PORT1_8SEL:

These bits define the PCMCIA port size.

PORT8SEL	PORT2_8SEL	PORT1_8SEL	Port1	Port2
1	*	*	8 bit port access	8 bit port access
0	0	1	8 bit port access	16 bit port access
0	1	0	16 bit port access	8 bit port access
0	1	1	8 bit port access	8 bit port access
0	0	0	16 bit port access	16 bit port access

DCLKDISABLE:

Setting this bit to 1 disables output of DCLKOUT. Clearing the bit to 0 enables output of DCLKOUT except during memory power down (when MEMPORWERDOWN bit is set to 1).

4.7.5 Memory Configuration 4 Register

OFFSET = $010

Bit	Label	RESET	Read/Write
31	ENBANK1HDRAM	0	R/W
30	ENBANK0HDRAM	0	R/W
29	ENARB	0	R/W
28	DISSNOOP	0	R/W
27	CLRWRBUSERRINT	0	R/W
26	ENBANK1OPT	0	R/W
25	ENBANK0OPT	0	R/W
24	ENWATCH	0	R/W
23-20	WATCHTIMEVAL[3:0]	X	R/W
19-17	Reserved		
16	MEMPOWERDOWN	1	R/W
15	ENRFSH1	0	R/W
14	ENRFSH0	0	R/W
13-8	RFSHVAL1[5:0]	X	R/W
7-6	Reserved		
5-0	RFSHVAL0[5:0]	X	R/W

ENBANK1HDRAM:

Setting this bit will enable the TMPR3911/12 to support HDRAM devices in Memory Bank 1. The BANK1CONF bits in the Memory Configuration 0 Register will configure the TMPR3912 to support either a 16-bit HDRAM configuration or a 32-bit HDRAM configuration, in the same manner that standard DRAM is setup.

ENBANK0HDRAM:

Setting this bit will enable the TMPR3911/12 to support HDRAM devices in Memory Bank 0. The BANK0CONF bits in the Memory Configuration 0 Register will configure the TMPR3911/12 to support either a 16-bit HDRAM configuration or a 32-bit HDRAM configuration, in the same manner that standard DRAM is setup.

ENARB:

Setting this bit will always cause the CPU Interface logic to externally arbitrate with the CPU during DMA reads and writes when the DISSNOOP control bit is set.

DISSNOOP:

Setting this bit will prevent the CPU Interface logic from causing the SNOOP signal to assert to the CPU Core during DMA writes.

CLRWRBUSERRINT:

WRBUSERRINT will occur whenever a write is timed out by the Watch Dog Timer.

This interrupt is connected directly to INT[0] on the CPU and is completely independent of the rest of the TMPR3911/12 interrupt logic. When this interrupt is set, it will remain set until the CLRWRBUSERRINT bit is asserted. The bit must be set and subsequently cleared to clear the interrupt.

ENBANK1OPT:

Setting this bit will cause the BIU to insert an extra clock of delay between the assertion of RAS1* and the assertion of the CAS* signals for DRAM accesses. This is needed for interfacing to 80 ns DRAMs when running with a clock greater than 70 MHz.

ENBANK0OPT:

Setting this bit will cause the BIU to insert an extra clock of delay between the assertion of RAS0* and the assertion of the CAS* signals for DRAM accesses. This is needed for interfacing to 80 ns DRAMs when running with a clock greater than 70 MHz.

ENWATCH:

Setting this bit will enable the Watch Dog Timer.

WATCHTIMEVAL[3:0]:

These bits define the length of the Watch Dog Timer. The Watch Dog Timer rate is determined by the following equation:

$$\text{Watch Dog Timer Rate} = \frac{(\text{WATCHTIME}\ [3:0]+1)*64}{f_{CLK}}$$

MEMPOWERDOWN:

Setting this bit will cause the memory interface to be put into Memory Power Down mode which will cause the DRAMs and SDRAMs to be put into self-refresh and all memory signals will be driven low. The memory interface will remain powered down until the processor attempts to access a location in memory other than function registers.

ENRFSH1:

Setting this bit will cause the BIU to begin refreshing Bank 1 DRAMs or SDRAMs. This bit should not be set until meeting the start up specification requirements for the memory devices.

ENRFSH0:

Setting this bit will cause the BIU to begin refreshing Bank 0 DRAMs or SDRAMs. This bit should not be set until meeting the start up specification requirements for the memory devices.

RFSHVAL1[5:0]:

These bits define the refresh period for Bank 1 refresh according to the following:

$0 15.26 µs

$1 61.04 µs

$2 91.55 µs

$3 122.07 µs

RFSHVAL0[5:0]:

These bits define the refresh period for Bank 0 refresh according to the following:

$0 15.26 µs

$1 61.04 µs

$2 91.55 µs

$3 122.07 µs

:

4.7.6 Memory Configuration 5 Register

OFFSET = $014: write-only

Bit	Label	RESET	Read/Write
31-9	STARTVAL2[31:9]	X	W
8-4	Reserved		
3-0	MASK2[3:0]	X	W

STARTVAL2[31:9]: write-only

These bits define the start address for re-map region 2.

MASK2[3:0]: write-only

These bits define the mask for re-map region 2.

4.7.7 Memory Configuration 6 Register

OFFSET = $018: write-only

Bit	Label	RESET	Read/Write
31-9	STARTVAL1[31:9]	X	W
8-4	Reserved		
3-0	MASK1[3:0]	X	W

STARTVAL1[31:9]: write-only

These bits define the start address for re-map region 1.

MASK1[3:0]: write-only

These bits define the mask for re-map region 1.

4.7.8 Memory Configuration 7 Register

OFFSET = $01C: write-only

Bit	Label	RESET	Read/Write
31-9	RMAPADD2[31:9]	X	W
8-0	Reserved		

RMAPADD2[31:9]: write-only

These bits define the destination address for re-map region 2.

4.7.9 Memory Configuration 8 Register

OFFSET = $020: write-only

Bit	Label	RESET	Read/Write
31-9	RMAPADD1[31:9]	X	W
8-0	Reserved		

RMAPADD1[31:9]: write-only

These bits define the destination address for re-map region 1.

5. SIU Module

5.1 Overview

The SIU Module within the TMPR3911/12 provides the multi-channel 32-bit DMA controller used to select and arbitrate the DMA channels from each SIM DMA source. Independent DMA channels are provided for Magicbus, video, sound, telecom, CHI, UARTA, and UARTB sources.

The SIU Module also contains the address decoder for all of the submodules contained within the SIM. This decoder generates the read and write enable pulses for each of the internal function registers.

5.2 Implementation

5.2.1 Block Diagram

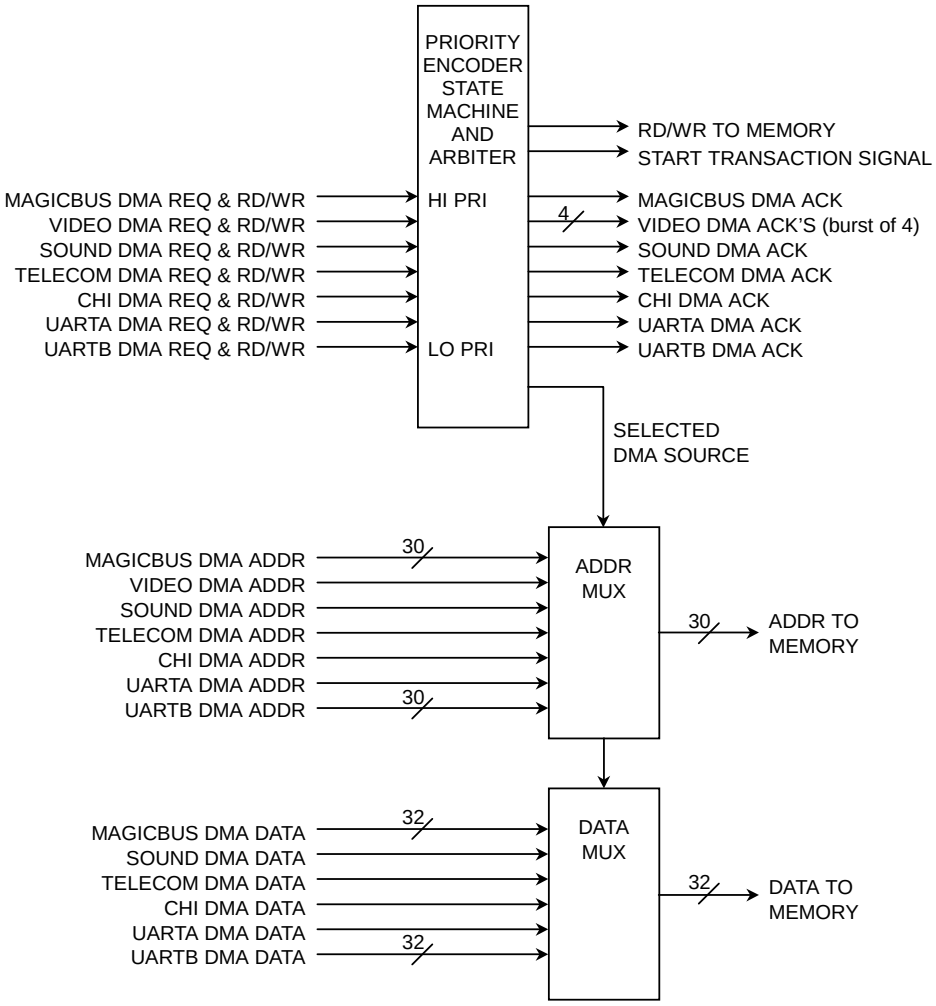


Figure 5.2.1 SIU DMA Controller Block Diagram

5.2.2 DMA Controller Description

Figure 5.2.1 shows a block diagram of the SIU DMA Controller. The DMA Controller receives the DMA requests and read/write status from all of the 7 possible DMA sources (Magicbus, video, sound, telecom, CHI, UARTA, and UARTB) and uses a Priority Encoder to arbitrate and select the highest priority request. If a new request is received and the DMA controller is currently busy, that request will remain pending and will be processed as soon as all other higher priority pending requests have been processed. The priority of the DMA sources and the DMA direction (read or write) supported are shown in Table 5.2.1.

Table 5.2.1 DMA Sources

DMA Source	priority	DMA read (from memory)	DMA write (to memory)
Magicbus	highest	yes	yes
video (burst of 4 longs)	↑	yes	no
sound	↑	yes	yes
telecom	↑	yes	yes
CHI	↓	yes	yes
UART-A (1 byte)	↓	yes	yes
UART-B (1 byte)	lowest	yes	yes

The state machine for the Priority Encoder also generates the START transaction pulse and read versus write status signal to the CPU Interface for each DMA transaction. After the arbiter decodes all received DMA requests and selects the highest priority request as the DMA channel to be processed, the DMA address and data busses for the selected DMA channel are routed to the memory subsystem. All DMA sources transfer 1 word per transaction, except for the video which always transfers 4 consecutive words at a time (and thus returns 4 separate ACKs to the Video Module), and the UARTs which transfer only 1 byte at a time. The UARTs are thus tagged with the appropriate Byte Enable (corresponding to 1 of 4 byte positions) depending on the byte count in the DMA address counter within the respective UART Module.

The state machine also generates the required acknowledge (ACK) signal at the end of each DMA transaction. This ACK signal is used by the circuit corresponding to the respective DMA source to either latch the data from memory (for a read transaction) or to enable the data to memory (for a write transaction).

The SIU also contains several test bits (see Section 5.3) which allow detailed testing of the DMA Controller for various combinations of simultaneous DMA requests. These bits are used for IC testing and should never be set.

5.2.3 Address Decoder Description

Figure 5.2.2 shows a block diagram of the SIM Address Decoder. The Address Decoder generates the read and write enable pulses for each of the internal TMPR3911/12 control and status registers. The lower 8 bits of the internal address bus are decoded, along with the SIM chip select and read/write status signal, to generate any 1 of 256 possible write enable pulses or any 1 of 256 possible read enable pulses. These enable pulses are used by the circuit/module targeted by the respective read or write pulse to either latch the data from the CPU (for a write transaction) or to enable the data to the CPU (for a read transaction). See the next section for a full listing of all the internal function Registers, including their address offset and read versus write support. The address offset for function Registers is the offset from the base address ($10C00000) of the function Registers in the System Address Map.

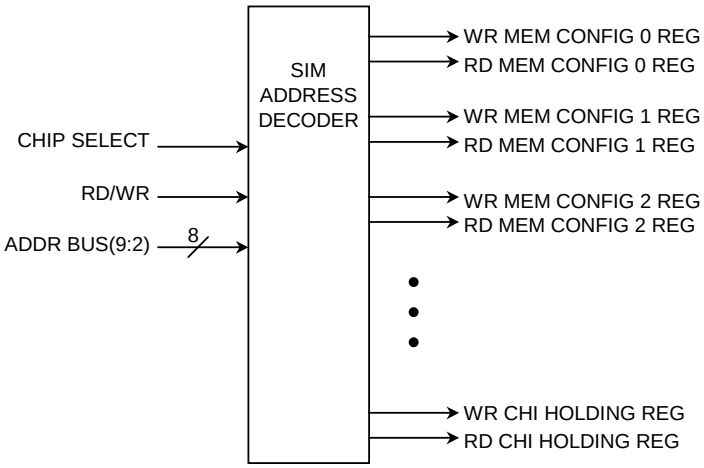


Figure 5.2.2 SIM Address Decoder Block Diagram

5.2.4 Internal Function Registers

Offset	Read/Write	Register
000	R/W	Memory Configuration 0
004	R/W	Memory Configuration 1
008	R/W	Memory Configuration 2
00C	R/W	Memory Configuration 3
010	R/W	Memory Configuration 4
014	W	Memory Configuration 5
018	W	Memory Configuration 6
01C	W	Memory Configuration 7
020	W	Memory Configuration 8
028	R/W	Video Control 1
02C	W	Video Control 2
030	W	Video Control 3
034	W	Video Control 4
038	W	Video Control 5
03C	W	Video Control 6
040	W	Video Control 7
044	W	Video Control 8
048	W	Video Control 9
04C	W	Video Control 10
050	W	Video Control 11
054	W	Video Control 12
058	W	Video Control 13
05C	W	Video Control 14
060	W	SIB Size
064	W	SIB Sound Receive Start
068	W	SIB Sound Transmit Start
06C	W	SIB Tel Receive Start
070	W	SIB Tel Transmit Start
074	R/W	SIB Control
078	R/W	SIB Sound Holding Register
07C	R/W	SIB Tel Holding Register
080	R/W	SIB SF0 Control
084	R/W	SIB SF1 Control
088	R	SIB SF0 Status
08C	R	SIB SF1 Status
090	R/W	SIB DMA Control
0A0	R/W	IR Control 1
0A4	W	IR Control 2
0A8	W	IR Holding Register
0B0	R/W	UARTA Control 1
0B4	W	UARTA Control 2
0B8	W	UARTA DMA Control 1
0BC	W	UARTA DMA Control 2
0C0	R	UARTA DMA Count
0C4	R/W	UARTA Holding Register
0C8	R/W	UARTB Control 1
0CC	W	UARTB Control 2
0D0	W	UARTB DMA Control 1
0D4	W	UARTB DMA Control 2
0D8	R	UARTB DMA Count
0DC	R/W	UARTB Holding Register

Offset	Read/Write	Register
0E0	R/W	Magicbus Control 1
0E4	W	Magicbus Control 2
0E8	W	Magicbus DMA Control 1
0EC	W	Magicbus DMA Control 2
0F0	R	Magicbus DMA Control
0F4	W	Magicbus Swap
0F8	R/W	Magicbus Holding Register
100	W	Clear Interrupt 1
100	R	Interrupt Status 1
104	W	Clear Interrupt 2
104	R	Interrupt Status 2
108	W	Clear Interrupt 3
108	R	Interrupt Status 3
10C	W	Clear Interrupt 4
10C	R	Interrupt Status 4
110	W	Clear Interrupt 5
110	R	Interrupt Status 5
114	R	Interrupt Status 6
118	R/W	Enable Interrupt 1
11C	R/W	Enable Interrupt 2
120	R/W	Enable Interrupt 3
124	R/W	Enable Interrupt 4
128	R/W	Enable Interrupt 5
12C	R/W	Enable Interrupt 6
140	R	RTC High
144	R	RTC Low
148	R/W	Alarm High
14C	R/W	Alarm Low
150	R/W	Timer Control
154	R/W	Periodic Timer
160	R/W	SPI Control
164	R/W	SPI Holding Register
180	R/W	IO Control
184	R/W	Multi-function IO Data Out
188	R/W	Multi-function IO Direction
18C	R	Multi-function IO Data In
190	R/W	Multi-function IO Select
194	R/W	I/O Power Down
198	R/W	Multi-function I/O Power Down
1C0	R/W	Clock Control
1C4	R/W	Power Control
1C8	R/W	SIU Test
1D8	R/W	CHI Control
1DC	R/W	CHI Pointer Enable
1E0	W	CHI Receive Pointer A
1E4	W	CHI Receive Pointer B
1E8	W	CHI Transmit Pointer A
1EC	W	CHI Transmit Pointer B
1F0	R/W	CHI Size
1F4	W	CHI Receive Start
1F8	W	CHI Transmit Start
1FC	R/W	CHI Holding Register
208	R	TX3911/12 Revision Register

5.3 SIU Registers

5.3.1 SIU Test Register

OFFSET = $1C8:

Bit	Label	RESET	Read/Write
31-8	Reserved		
7	ENDMATEST	0	R/W
6	ENNOTIMETEST	0	R/W
5-0	DMATESTWR[5:0]	0	R/W

ENDMATEST:

This bit is used for IC testing and should never be set.

ENNOTIMETEST:

This bit is used for IC testing and should never be set.

DMATESTWR[5:0]:

These bits are used for IC testing and should never be set.

5.3.2 TMPR3911/12 Revision Register

OFFSET = $208:

Bit	Label	RESET	Read/Write
31-8	Reserved		
7	READREVISION	–	R

READREVISION:

These bits indicate revisions of the TMPR3911/12. The following values are read out:

TMPR3912AU-92 : $01

TMPR3912XB-92 : $01

TMPR3911BU : $02

TMPR3911BXB : $02

6. Clock Module

6.1 Overview

The Clock Module within the TMPR3911/12 contains logic for generating the clocks for all other internal TMPR3911/12 modules, as well as for generating certain externally driven TMPR3911/12 output clocks. The Clock Module contains dividers for generating the correct rates for each clock and also contains logic for independently enabling or disabling individual clocks under software control.

DIVMOD in power control register in Power Module can be used to keep compatibility between TMPR3911 and TMPR3912 peripheral module, such as UART, timer and so on. Because once DIVMOD is set to high, peripheral master clock is generated by free clock, which means frequency of peripheral master clock is free of RF function.

6.1.1 Related Pins

SYSCLKIN: INPUT

This pin should be connected along with SYSCLKOUT to an external crystal which is the main TMPR3911/12 clock source.

SYSCLKOUT: OUTPUT

This pin should be connected along with SYSCLKIN to an external crystal which is the main TMPR3911/12 clock source.

6.2 Implementation

6.2.1 Block Diagram

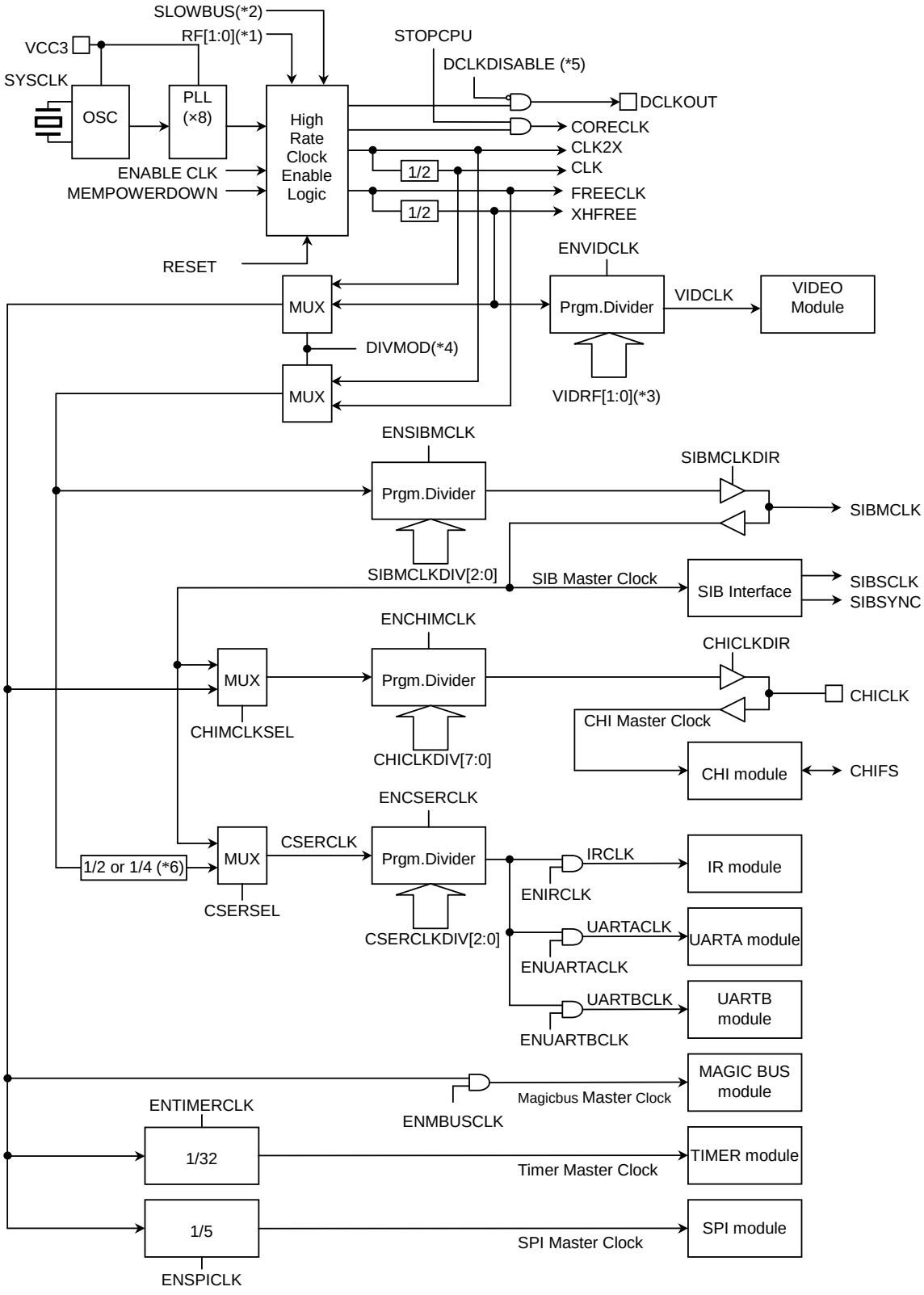


Figure 6.2.1 Clock Module Block Diagram

RF[1:0]*[1]	SLOWBUS*[2]	CORECLK	DCLKOUT	CLK2X	FREECLK	CLK	XHFREE
00	0	F	F	F	F	F/2	F/2
01	0	F/2	F/2	F/2	F	F/4	F/2
10	0	F/4	F/4	F/4	F	F/8	F/2
11	0	F/8	F/8	F/8	F	F/16	F/2
00	1	F/2	F/2	F/2	F	F/4	F/2
01	1	F/4	F/4	F/4	F	F/8	F/2
10	1	F/8	F/8	F/8	F	F/16	F/2
11	1	F/16	F/16	F/16	F	F/32	F/2

VIDRF[1:0]*[3]	VIDCLK
00	F/2
01	F/4
10	F/8
11	F/16

Note:

*1 RF[1:0] is defined by the Config Register in TX39/H Core.
*2 SLOWBUS is defined by the Power Control Register in Power Module.
*3 VIDRF[1:0] is defined by the Power Control Register in Power Module.
*4 DIVMOD is defined by the Power Control Register in Power Module.
*5 DCLKDISABLE is defined by the Memory Configuration 3 Register.
*6 1/2: TMPR3912XB-92/AU-92
1/4: TMP3911BU/BXB

6.2.2 Clock Module Description

Figure 6.2.1 shows a block diagram of the Clock Module. The main TMPR3911/12 clock source is from an external crystal, connected to the TMPR3911/12 via the SYSCLKIN/SYSCLKOUT pins. The TMPR3911/12 contains an oscillator which is internally connected to these pins. The output of the oscillator feeds a phase-locked-loop (or multiplier) circuit which takes the lower rate oscillator output and generates the higher-rate clocks required by the TMPR3911/12. For example, if the external SYSCLK crystal rate is 9.216 MHz, the PLL multiplier circuit performs an ×8 rate multiplication to generate a high-speed clock rate of 73.728 MHz.

Each of the individual clocks can also be enabled or disabled under software control, in order to reduce power consumption. All of these clocks default to a disabled state during reset.

The main high rate clocks generated by the Clock Module are the following:

- CLK2X – variable rate clock and same rate as the CORECLK; used for generating other lower rate clocks and for certain memory interface circuits in the BIU
- CLK – half rate of CLK2X used as the master clock for many of the TMPR3911/12 modules and for generating other lower rate clocks
- CORECLK – same rate as CLK2X and twice the rate of CLK (TMPR3911/12); used as master clock for CPU core.
- DCLKOUT – same rate as CLK2X; used as high-speed externally driven clock for SDRAMs
- FREECLK – fixed rate clock generated by PLL
- XHFREE – half rate of FREECLK
- The SIBMCLK pin can be configured as an output, for which the programmable rate is generated by dividing down from CLK2X. The SIBMCLK pin can also be configured as an input. In this mode, all SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to CLK2X. This mode allows optional decoupling of the SIB (as well as the CHI and UART) rates from the CPU core clock rate. The selected SIBMCLK source is then used as the SIB Master Clock for the SIB Module circuits, and is also used to generate the SIBSCLK and SIBSYNC externally driven TMPR3911/12 output signals.
- The CHICLK pin can be configured as an output, for which the programmable rate is generated by dividing down from either CLK or the SIB Master Clock. The CHICLK pin can also be configured as an input. In this mode, all CHI clocks are derived from an external peripheral source and the CHI Module will slave to this external clock. The selected CHICLK source is used as the CHI Master Clock for the CHI Module circuits, and is also used to generate the CHIFS externally driven TMPR3911/12 output signal.
- The programmable UART Master Clock is generated by dividing down from either CLK or the SIB Master Clock, and is used as the master clock for the baud generator circuit within each UART Module.
- The Clock Module also contains several fixed dividers for generating the master clocks for the Video, Magicbus, RTC timer, and SPI circuits. These clocks are divided down from either CLK2X or CLK.

6.3 Clock Registers

6.3.1 Clock Control Register

OFFSET = $1C0:

Bit	Label	RESET	Read/Write
31-24	CHICLKDIV[7:0]	0	R/W
23	Reserved *	0	–
22	Reserved *	0	–
21	CHIMCLKSEL	0	R/W
20	CHICLKDIR	0	R/W
19	ENCHIMCLK	0	R/W
18	ENVIDCLK	0	R/W
17	ENMBUSCLK	0	R/W
16	ENSPICLK	0	R/W
15	ENTIMERCLK	0	R/W
14	Reserved *	0	–
13	SIBMCLKDIR	0	R/W
12	Reserved	0	–
11	ENSIBMCLK	0	R/W
10-8	SIBMCLKDIV[2:0]	0	R/W
7	CSERSEL	1	R/W
6-4	CSERDIV[2:0]	0	R/W
3	ENCSERCLK	0	R/W
2	ENIRCLK	0	R/W
1	ENUARTACLK	0	R/W
0	ENUARTBCLK	0	R/W

*) These bits must be zero.

CHICLKDIV[7:0]:

These bits define the divide-modulus for the programmable divider used to generate CHICLK. The divide-modulus is equal to (CHICLKDIV+2). (See the following table.)

CHICLKDIV[7:0]	divide-modulus
0	2
1	3
2	4
⌣	⌣
n	n+2
⌣	⌣
253	255
254	256

CHIMCLKSEL:

Setting this bit to a logic “1” selects the external SIBMCLK source to be the CHI master clock. Clearing this bit to a logic “0” selects an internal clock source to be the CHI master clock.

CHICLKDIR:

This bit controls the direction of the CHICLK pin. Setting this bit to a logic “1” configures CHICLK to be an output (CHI master mode). Clearing this bit to a logic “0” configures CHICLK to be an input (CHI slave mode).

ENCHIMCLK:

This bit is used to enable or disable the CHICLK counter and CHICLK clock generation. Setting this bit to a logic “1” enables the CHICLK counter and CHICLK generation. Clearing this bit to a logic “0” disables the CHICLK counter and CHICLK generation, halting the clock to the CHI Module in order to reduce power consumption.

ENVIDCLK:

This bit is used to enable or disable the video master clock. Setting this bit to a logic “1” enables the VIDCLK. Clearing this bit to a logic “0” disables the VIDCLK, halting the clock to the Video Module in order to reduce power consumption.

ENMBUSCLK:

This bit is used to enable or disable the Magicbus master clock. Setting this bit to a logic “1” enables the Magicbus master clock. Clearing this bit to a logic “0” disables the Magicbus master clock, halting the clock to the Magicbus Module in order to reduce power consumption.

ENSPICLK:

This bit is used to enable or disable the SPICLK counter and SPI master clock generation. Setting this bit to a logic “1” enables the SPICLK counter and SPI master clock generation. Clearing this bit to a logic “0” disables the SPICLK counter and SPI master clock generation, halting the clock to the SPI Module in order to reduce power consumption.

ENTIMERCLK:

This bit is used to enable or disable the RTC periodic timer clock counter. Setting this bit to a logic “1” enables the timer clock counter. Clearing this bit to a logic “0” disables the timer clock counter.

SIBMCLKDIR:

This bit controls the direction of the SIBMCLK pin. Setting this bit to a logic “1” configures SIBMCLK to be an output. Clearing this bit to a logic “0” configures SIBMCLK to be an input.

ENSIBMCLK:

This bit is used to enable or disable the SIBMCLK counter and SIBMCLK clock generation. Setting this bit to a logic “1” enables the SIBMCLK counter and SIBMCLK generation. Clearing this bit to a logic “0” disables the SIBMCLK counter and SIBMCLK generation, halting the clock to the SIB Module in order to reduce power consumption.

SIBMCLKDIV[2:0]:

These bits define the divide-modulus for the programmable divider used to generate SIBMCLK. The divide-modulus is equal to (SIBMCLK+2). (See the following table.)

SIBMCLKDIV	divide-modulus
0	2
1	3
2	4
3	5
4	6
5	7
6	8

CSERSEL:

Clearing this bit to a logic “0” selects the external SIBMCLK source to be the CSERCLK master clock. Setting this bit to a logic “1” selects an internal clock source to be the CSERCLK master clock.

CSERDIV[2:0]:

These bits define the divide-modulus for the programmable divider used to generate CSERCLK. The divide-modulus is equal to (CSERDIV+2). (See the following table.)

CSERDIV	divide-modulus
0	2
1	3
2	4
3	5
4	6
5	7
6	8

ENCSERCLK:

This bit is used to enable or disable the CSERCLK counter. Setting this bit to a logic “1” enables the CSERCLK counter. Clearing this bit to a logic “0” disables the CSERCLK counter.

ENIRCLK:

This bit is used to enable or disable the IRCLK clock generation. Setting this bit to a logic “1” enables the IRCLK generation. Clearing this bit to a logic “0” disables the IRCLK generation, halting the clock to the IR Module in order to reduce power consumption.

ENUARTACLK:

This bit is used to enable or disable the UARTACLK clock generation. Setting this bit to a logic “1” enables the UARTACLK generation. Clearing this bit to a logic “0” disables the UARTACLK generation, halting the clock to the UARTA Module in order to reduce power consumption.

ENUARTBCLK:

This bit is used to enable or disable the UARTBCLK clock generation. Setting this bit to a logic “1” enables the UARTBCLK generation. Clearing this bit to a logic “0” disables the UARTBCLK generation, halting the clock to the UARTB Module in order to reduce power consumption.

Table 6.3.1 through Table 6.3.3 show examples of the frequencies for each clock.

Table 6.3.1 Clock Frequencies (1)

DIVMOD = 0, CHIMCLKSEL = 0, CSERSEL = 1, RF[1:0] = 00, SLOWBUS = 0

Clock	Frequency					
	Symbol	Calculation	Register setting value	Frequency (MHz)		
SYSCLK (External Crystal)	f_{IN}	–		7.3728	9.216	11.520
DCLKOUT (SDRAM Clock)	$f_{DCLKOUT}$	$f_{IN} \times 8$		58.9824	73.728	92.160
CLK	f_{CLK}	$f_{IN} \times 4$		29.4912	36.864	46.080
CLK2X	f_{CLK2X}	$f_{IN} \times 8$		58.9824	73.728	92.160
XHFREE	f_{XHFREE}	$f_{IN} \times 4$		29.4912	36.864	46.080
FREECLK	$f_{FREECLK}$	$f_{IN} \times 8$		58.9824	73.728	92.160
VIDCLK (VIDEO Master Clock)	f_{VIDCLK}	$f_{XHFREE} \div (2^{VIDRF[1:0]})$	VIDRF = 00	29.4912	36.864	46.080
			01	14.7456	18.432	23.040
			10	7.3728	9.216	11.520
			11	3.6864	4.608	5.760
SIBMCLK (SIB Master Clock)	$f_{SIBMCLK}$	$f_{CLK2X} \div$ (SIBMCLKDIV[2:0] + 2)	SIBMCLKDIV =			
			000	29.4912	36.864	46.080
			001	19.6608	24.576	30.720
			010	14.7456	18.432	23.040
			011	11.7965	14.746	18.432
			100	9.8304	12.288	15.360
			101	8.4261	10.532	13.166
			110	7.3728	9.216	11.520
			111	–	–	–
CHICLK (CHI Clock)	f_{CHICLK}	$f_{CLK} \div$ (CHICLKDIV[7:0] + 2)	CHICLKDIV =			
			$00	14.7456	18.432	23.040
			~	~	~	~
			$FE	0.1152	0.144	0.180
CSERCLK	$f_{CSERCLK}$	$f_{CLK2X} \div 4$ (3911BU/BXB)		14.7456	18.432	–
		$f_{CLK2X} \div 2$ (AU-92/XB-92)		–	–	46.08
IRCLK (IR Master Clock)	f_{IRCLK}	$f_{CSERCLK} \div$ (CSERDIV[2:0] + 2)	CSERDIV = 000	7.3728	9.216	23.04
UARTACLK (UART-A Master Clock)	$f_{UARTACLK}$	$f_{CSERCLK} \div$ (CSERDIV[2:0] + 2)	001	4.9152	6.144	15.36
			010	3.6864	4.608	11.52
UARTBCLK (UART-B Master Clock)	$f_{UARTBCLK}$	$f_{CSERCLK} \div$ (CSERDIV[2:0] + 2)	011	2.94912	3.6864	9.216
			100	2.4576	3.072	7.680
			101	2.1065	2.633	6.5829
			110	1.8432	2.304	5.760
			111	–	–	–
MBUSMCLK (Magic Bus Master Clock)	$f_{MBUSMCLK}$	f_{CLK}		29.4912	36.864	46.089
TIMERCLK (Timer Master Clock)	$f_{TIMERCLK}$	$f_{CLK} \div 32$		0.9216	1.152	1.440
SPIMCLK (SPI Master Clock)	$f_{SPIMCLK}$	$f_{CLK} \div 5$		5.8982	7.3728	9.216
SYSCLK (External Crystal)	f_{IN}	–		7.3728	9.216	11.520
DCLKOUT (SDRAM Clock)	$f_{DCLKOUT}$	$f_{IN} \times 4$		29.4912	36.864	46.080
CLK	f_{CLK}	$f_{IN} \times 2$		14.7456	18.432	23.040
CLK2X	f_{CLK2X}	$f_{IN} \times 4$		29.4912	36.864	46.080

Table 6.3.2 Clock Frequencies (2)

DIVMOD = 0, CHIMCLKSEL = 0, CSERSEL = 1,
RF[1:0] = 01, SLOWBUS = 0 or RF[1:0] = 00, SLOWBUS = 1

Clock	Frequency					
	Symbol	Calculation	Register setting value	Frequency (MHz)		
XHFREE	f_{XHFREE}	$f_{IN} \times 4$		29.4912	36.864	46.080
FREECLK	$f_{FREECLK}$	$f_{IN} \times 8$		58.9824	73.728	92.160
VIDCLK (VIDEO Master Clock)	f_{VIDCLK}	f_{XHFREE} ÷ (2^VIDRF[1:0])	VIDRF = 00	29.4912	36.864	46.080
			01	14.7456	18.432	23.040
			10	7.3728	9.216	11.520
			11	3.6864	4.608	5.760
SIBMCLK (SIB Master Clock)	$f_{SIBMCLK}$	f_{CLK2X} ÷ (SIBMCLKDIV[2:0] + 2)	SIBMCLKDIV =			
			000	14.7456	18.432	23.040
			001	9.8304	12.288	15.360
			010	7.3728	9.216	11.520
			011	5.8982	7.373	9.216
			100	4.9152	6.144	7.680
			101	4.2130	5.266	6.583
			110	3.6864	4.608	5.760
			111	–	–	–
CHICLK (CHI Clock)	f_{CHICLK}	f_{CLK} ÷ (CHICLKDIV[7:0] + 2)	CHICLKDIV =			
			$00	7.3728	9.216	11.520
			~	~	~	~
			$FE	0.0576	0.072	0.090
CSERCLK	$f_{CSERCLK}$	f_{CLK2X} ÷ 4 (3911BU/BXB)		7.3728	9.216	–
		f_{CLK2X} ÷ 2 (AU-92/XB-92)		–	–	23.040
IRCLK (IR Master Clock)	f_{IRCLK}	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)	CSERDIV = 000	3.6864	4.608	11.52
			001	2.4576	3.072	7.68
			010	1.8432	2.304	5.76
			011	1.4746	1.8432	4.608
			100	1.2288	1.536	3.840
			101	1.0533	1.3165	3.291
			110	0.9216	1.152	2.880
			111	–	–	–
UARTACLK (UART-A Master Clock)	$f_{UARTACLK}$	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)				
UARTBCLK (UART-B Master Clock)	$f_{UARTBCLK}$	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)				
MBUSMCLK (Magic Bus Master Clock)	$f_{MBUSMCLK}$	f_{CLK}		14.7456	18.432	23.040
TIMERCLK (Timer Master Clock)	$f_{TIMERCLK}$	f_{CLK} ÷ 32		0.4608	0.576	0.720
				2.9491	3.6864	4.608
SPIMCLK (SPI Master Clock)	$f_{SPIMCLK}$	f_{CLK} ÷ 5				
SYSCLK (External Crystal)	f_{IN}	–				
DCLKOUT (SDRAM Clock)	$f_{DCLKOUT}$	$f_{IN} \times 4$		29.4912	36.864	46.080
CLK	f_{CLK}	$f_{IN} \times 2$		14.7456	18.432	23.040
CLK2X	f_{CLK2X}	$f_{IN} \times 4$		29.4912	36.864	46.080
XHFREE	f_{XHFREE}	$f_{IN} \times 4$		29.4912	36.864	46.080
FREECLK	$f_{FREECLK}$	$f_{IN} \times 8$		58.9824	73.728	92.160

Table 6.3.3 Clock Frequencies (3)

DIVMOD = 1, CHIMCLKSEL = 0, CSERSEL = 1,
RF[1:0] = 01, SLOWBUS = 0 or RF[1:0] = 00, SLOWBUS = 1

Clock	Frequency					
	Symbol	Calculation	Register setting value	Frequency (MHz)		
VIDCLK (VIDEO Master Clock)	f_{VIDCLK}	f_{XHFREE} ÷ (2^VIDRF[1:0])	VIDRF = 00	29.4912	36.864	46.080
			01	14.7456	18.432	23.040
			10	7.3728	9.216	11.520
			11	3.6864	4.608	5.760
SIBMCLK (SIB Master Clock)	$f_{SIBMCLK}$	$f_{FREECLK}$ ÷ (SIBMCLKDIV[2:0] + 2)	SIBMCLKDIV =			
			000	29.4912	36.864	46.080
			001	19.6608	24.576	30.720
			010	14.7456	18.432	23.040
			011	11.7965	14.746	18.432
			100	9.8304	12.288	15.360
			101	8.4261	10.532	13.166
			110	7.3728	9.216	11.520
			111	–	–	–
CHICLK (CHI Clock)	f_{CHICLK}	f_{XHFREE} ÷ (CHICLKDIV[7:0] + 2)	CHICLKDIV =			
			$00	14.7456	18.432	23.040
			~	~	~	~
			$FE	0.1152	0.144	0.180
CSERCLK	$f_{CSERCLK}$	$f_{FREECLK}$ ÷ 4 (3911BU/BXB)		14.7456	18.432	–
		$f_{FREECLK}$ ÷ 2 (AU-92/XB-92)		–	–	46.08
IRCLK (IR Master Clock)	f_{IRCLK}	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)	CSERDIV =	7.3728	9.216	23.04
UARTACLK (UART-A Master Clock)	$f_{UARTACLK}$	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)	000	4.9152	6.144	15.36
			001	3.6864	4.608	11.52
UARTBCLK (UART-B Master Clock)	$f_{UARTBCLK}$	$f_{CSERCLK}$ ÷ (CSERDIV[2:0] + 2)	010	2.94912	3.6864	9.216
			011	2.4576	3.072	7.680
			100	2.1065	2.633	6.5829
			101	1.8432	2.304	5.760
			110	–	–	–
			111			
MBUSMCLK (Magic Bus Master Clock)	$f_{MBUSMCLK}$	f_{XHFREE}		29.4912	36.864	46.089
TIMERCLK (Timer Master Clock)	$f_{TIMERCLK}$	f_{XHFREE} ÷ 32		0.9216	1.152	1.440
SPIMCLK (SPI Master Clock)	$f_{SPIMCLK}$	f_{XHFREE} ÷ 5		5.8982	7.3728	9.216

7. CHI Module

7.1 Overview

The CHI Module within the TMPR3911/12 contains holding registers, shift registers, DMA support, and other logic to support interfacing to external full-duplex serial TDM communication peripherals, including ISDN communication devices and PCM/TDM serial highways. The TMPR3911/12 implementation of the CHI Module is based on the Concentration Highway Interface standard specified by Intel and AT&T, which is intended to allow glueless interface to various TDM highways used by numerous commercial products. The TMPR3911/12 CHI Module can also be used to support an intra-system high-speed serial DMA channel within a PIC system.

The TMPR3911/12 CHI Module utilizes a 4-signal interface consisting of clock, sync, transmit serial data, and receive serial data. The data is organized as a TDM format, with up to 64 timeslots (nominally 8 bits each) per frame, with a nominal frame rate of 8 kHz (8 bits × 8 kHz = 64 kbps nominal data rate per channel). Up to four input timeslots and up to four output timeslots can be independently selected in each half-frame. The CHI Module does not handle the timeslots which are not selected for input or output. The number of timeslots and the data rate (up to 4.096 Mbps) and frame rate (up to 64 kHz, depending on the system configuration) are programmable, providing flexibility for supporting various TDM communication peripherals. These timeslots are commonly used to carry voice, data, or control and status information.

The CHI Module provides full-duplex DMA support for receive and transmit (two DMA channels). The DMA buffers can be configured in a continuous (circular) buffer mode or a one-time (empty or fill, then stop) buffer mode. Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and efficiently empty or fill half of the DMA buffer in a ping-pong fashion. The DMA buffer size is programmable (up to a maximum of 16 Kbytes) and the receive and transmit buffers can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation). Also available is a direct CPU read/write mode for bypassing the DMA, allowing the CPU to read or write the CHI data on a sample by sample basis, if so desired.

7.1.1 Related Pins

CHIFS: INPUT/OUTPUT

This pin is the CHI frame synchronization signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3911/12 to be the master CHI sync source. As an input, this pin allows an external peripheral to be the master CHI sync source and the TMPR3911/12 CHI Module will slave to this external sync.

CHICLK: INPUT/OUTPUT

This pin is the CHI clock signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3911/12 to be the master CHI clock source. As an input, this pin allows an external peripheral to be the master CHI clock source and the TMPR3911/12 CHI Module will slave to this external clock.

CHIDOUT: OUTPUT

This pin is the CHI serial data output signal.

CHIDIN: INPUT

This pin is the CHI serial data input signal.

7.2 Interface Requirements

7.2.1 Frame Structure and Serial Timing

Each CHI frame (nominally 8 kHz rate) is time-division-multiplexed into several timeslots or channels. The total number of timeslots per frame is programmable, with a maximum of 64 timeslots allowed, and the number of timeslots is also restricted to an even number. Each timeslot is 8 bits, although 16-bit or 32-bit channels can be supported by accessing adjacent timeslots.

The TMPR3911/12 CHI Module supports a master or slave mode for both the clock (CHICLK) and sync (CHIFS). For the master mode, the TMPR3911/12 contains programmable dividers for generating the clock and/or sync signal, synchronously dividing down from the main core clock (CLK). For the slave mode, TMPR3911/12 accepts external clock and/or sync signals and utilizes “digital-PLL” type circuitry to stayed “locked” to the external source.

The CHI Module supports the following programmable features which allow support for various clock and sync timing formats:

- 1x versus 2x clock modes for CHICLK (2x clock mode uses two CHICLK periods per data bit)
- MSB-first versus LSB-first serial formats for transmit and receive
- rising versus falling edge (polarity) used for frame sync triggering
- CHIFS signal can be sampled on either rising or falling edge of CHICLK
- CHIDIN receive data can be sampled on either rising or falling edge of CHICLK
- CHIDOUT transmit data can be pushed on either rising or falling edge of CHICLK
- CHIDIN receive data can have programmable bit offset (timeslot 0 offset from CHIFS)
- CHIDOUT transmit data can have programmable bit offset (timeslot 0 offset from CHIFS)
- CHIDOUT transmit data (tri-state) output buffer enable is dynamically asserted for only active timeslots; for sleep mode CHIDOUT is always tri-stated
- for CHIFS master mode, the CHIFS pulse width and polarity is programmable

The TMPR3911/12 CHI Module allows for a programmable bit offset for both CHIDIN and CHIDOUT, which is related to the number of clock cycles between the start of timeslot 0 and CHIFS. This flexibility allows the TMPR3911/12 CHI Module to support a wide variety of interface clock and sync timing formats. The control bits for controlling the CHIDIN bit offset are CHIRXBOFF[3:0], while the control bits for controlling the CHIDOUT bit offset are CHITXBOFF[3:0].

Table 7.2.1 and Table 7.2.2 shows a summary matrix for the values of CERX and CETX for all possible settings of CHIRXBOFF and CHITXBOFF, respectively. These values are shown for various configurations of CHICLK mode (1x versus 2x), CHIFSEDGE, CHIRXEDGE, and CHITXEDGE. The CHIFSEDGE settings determine whether to use the rising edge (CHIFSEDGE = 1) or falling edge (CHIFSEDGE = 0) of CHICLK to sample CHIFS. The CHIRXEDGE settings determine whether to use the rising edge (CHIRXEDGE = 1) or falling edge (CHIRXEDGE = 0) of CHICLK to sample the CHIDIN input. The CHITXEDGE settings determine whether to use the rising edge (CHITXEDGE = 1) or falling edge (CHITXEDGE = 0) of CHICLK to push the CHIDOUT output. CERX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDIN is sampled. CETX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDOUT is pushed. The CHI frame structure and bit offsets are shown in Figure 7.2.1~7.2.7 for various clock and sync configurations.

Table 7.2.1 CERX Values for CHIRXBOFF Versus Clock and Edge Configurations

chiclk mode	chifs-edge	chirx-edge	CHIRXBOFF															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	0	1	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
1x	1	0	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
2x	1	1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
2x	0	1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
2x	1	0	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31

Table 7.2.2 CETX Values for CHITXBOFF Versus Clock and Edge Configurations

chiclk mode	chifs-edge	chitx-edge	CHITXBOFF															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	0	1	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
1x	1	0	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
2x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
2x	0	1	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	1	0	-1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29

When the CHI frame consists of only two time slots, not all the BOFF values are supported. The offsets must be within the half-frame.

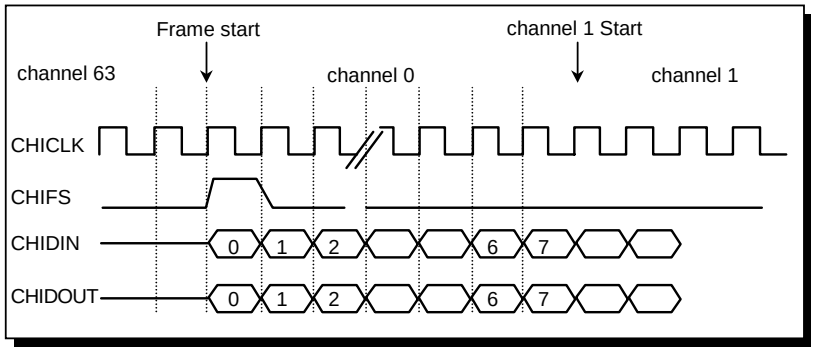


Figure 7.2.1 CHI Frame Structure Example

CHICLK 1X mode
CHIFS sampled on falling edge
CHIDIN sampled on falling edge; RXBOFF = 0; CERX = 0
CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = –1

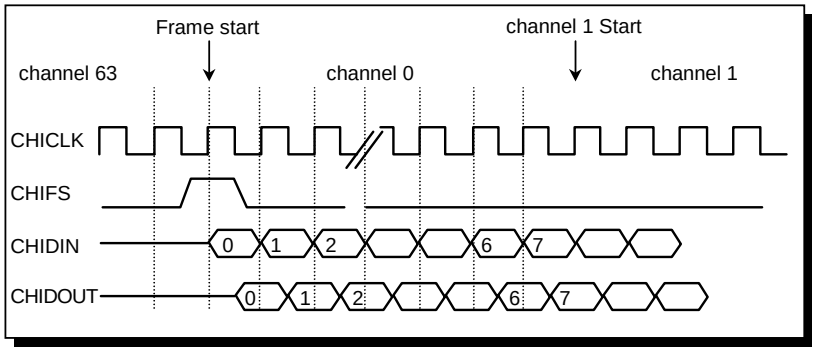


Figure 7.2.2 CHI Frame Structure Example

CHICLK 1X mode
CHIFS sampled on rising edge
CHIDIN sampled on falling edge; RXBOFF = 1; CERX = 1
CHIDOUT pushed on falling edge; TXBOFF = 1; CETX = 1

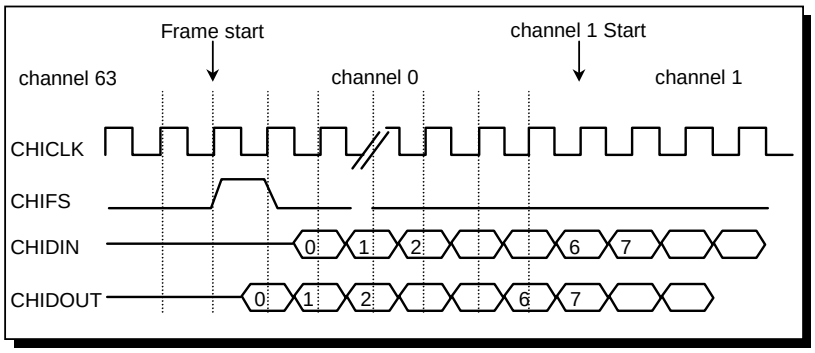


Figure 7.2.3 CHI Frame Structure Example

CHICLK 1X mode
CHIFS sampled on falling edge
CHIDIN sampled on rising edge; RXBOFF = 2; CERX = 3
CHIDOUT pushed on falling edge; TXBOFF = 0; CETX = 0

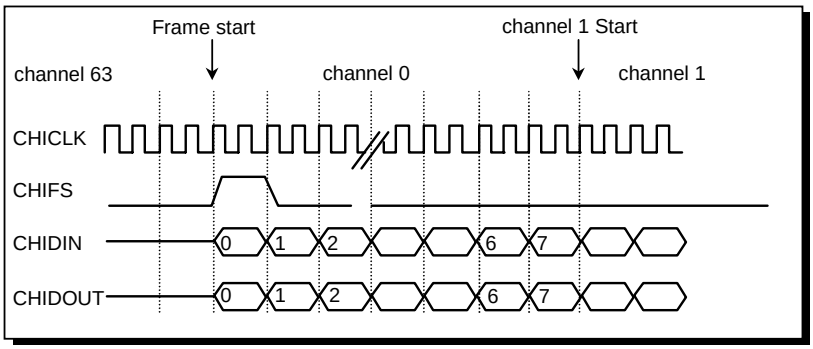


Figure 7.2.4 CHI Frame Structure Example

CHICLK 2X mode
CHIFS sampled on falling edge
CHIDIN sampled on rising edge; RXBOFF = 0; CERX = 1
CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = −1

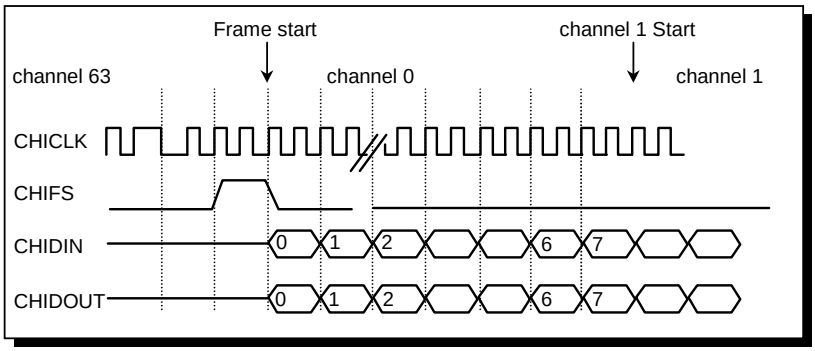


Figure 7.2.5 CHI Frame Structure Example

CHICLK 2X mode
CHIFS sampled on falling edge
CHIDIN sampled on rising edge; RXBOFF = 2; CERX = 5
CHIDOUT pushed on rising edge; TXBOFF = 2; CETX = 3

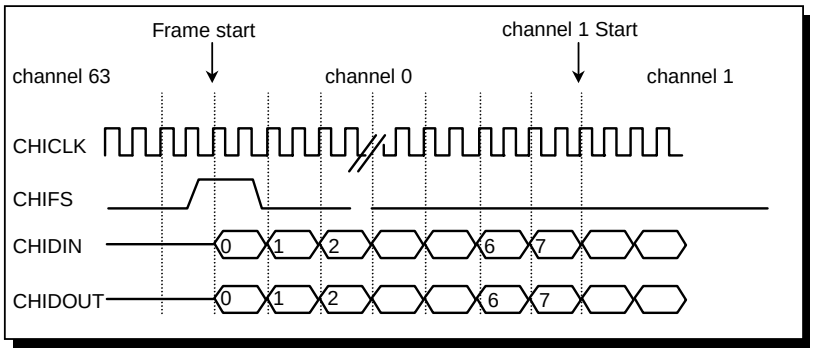


Figure 7.2.6 CHI Frame Structure Example

CHICLK 2X mode
CHIFS sampled on rising edge
CHIDIN sampled on rising edge; RXBOFF = 0; CERX = 2
CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = 0

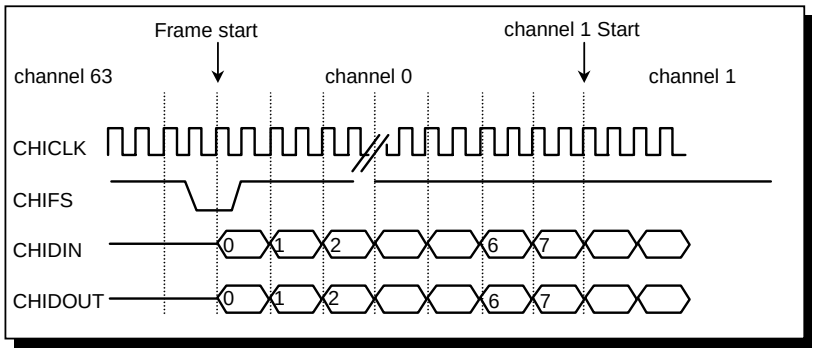


Figure 7.2.7 CHI Frame Structure Example

CHICLK 2X mode

CHIFS sampled on falling edge

CHIRXFSPOL = 1 (negative polarity)

CHIDIN sampled on rising edge; RXBOFF = 1; CERX = 3

CHIDOUT pushed on rising edge; TXBOFF = 1; CETX = 1

7.2.2 Configurations

The programmability of the clock, sync, bit offsets, and number of timeslots allows the CHI Module to support a wide range of configurations. Several of these configurations are commonly utilized as communication interfaces by numerous commercial products, some of which are briefly discussed below. Other configurations are available for application-specific usage.

The K2 Interface is an AT&T standard used as a serial inter-chip digital interface between U-interface transmission circuits and various system-wide interface circuits. The K2 Interface utilizes four pins (clock, sync, data in, and data out) and consists of eight 8-bit timeslots, with a frame rate of 8 kHz and a clock rate of 512 kHz.

The SLD Interface utilizes 3 pins (clock, sync, data), with the transmit and receive pins tied together, such that the data is sent bi-directionally in a ping-pong fashion. The SLD Interface consists of eight 8-bit timeslots, with 4 timeslots reserved for transmit and 4 timeslots reserved for receive. The frame rate is 8 kHz and the clock rate is 512 kHz.

The GCI Interface is a 4-pin variable-speed TDM highway utilizing anywhere from 4 to 48 timeslots. The frame rate is 8 kHz and the clock rate is 2x the data rate and varies from 512-kHz to 6.144 MHz.

The IOM-2 Interface is commonly used to interface to 4-pin ISDN line interface drivers. Each frame consists of twelve 8-bit timeslots, with a frame rate of 8 kHz and a (2x) clock rate of 1.536-MHz.

The CEPT Level-1 PCM Format is a common communications standard used for digital transmission of voice and data. Each frame consists of 32 8-bit timeslots, with a frame rate of 8-kHz and a (1x) clock rate of 2.048 MHz.

Table 7.2.3 shows a summary matrix of several example CHI configurations and their associated parameters.

Table 7.2.3 Example CHI Configurations (table values based on CLK = 36.864 MHz)

chiclk divide	time slots	chiclk mode	fs	chiclk rate	data rate	comments
4.5	64	2x	8 kHz	8.192 MHz	4.096 Mbps	slave mode only
6	48	2x	8 kHz	6.144 MHz	3.072 Mbps	GCI format
9	64	1x	8 kHz	4.096 MHz	4.096 Mbps	CHI format
9	8	1x	64 kHz	4.096 MHz	4.096 Mbps	hi-speed mode
10	64	1x	7.2 kHz	3.6864 MHz	3.6864 Mbps	
12	24	2x	8 kHz	3.072 MHz	1.536 Mbps	
12	48	1x	8 kHz	3.072 MHz	3.072 Mbps	
18	32	1x	8 kHz	2.048 MHz	2.048 Mbps	CEPT PCM format
24	12	2x	8 kHz	1.536 MHz	768 kbps	IOM-2 format
72	8	1x	8 kHz	512 MHz	512 kbps	K2, SLD formats

Note that the maximum achievable frame rate depends on the system configuration. If devices with long access time and/or 16 bit-wide data bus are used, the frame rate of 64 kHz may be unachievable because of the reduced bus band width. Using other interfaces in parallel with CHI and/or reducing the CPU clock frequencies will also reduce the band width available for CHI.

7.3 Implementation

7.3.1 Block Diagram

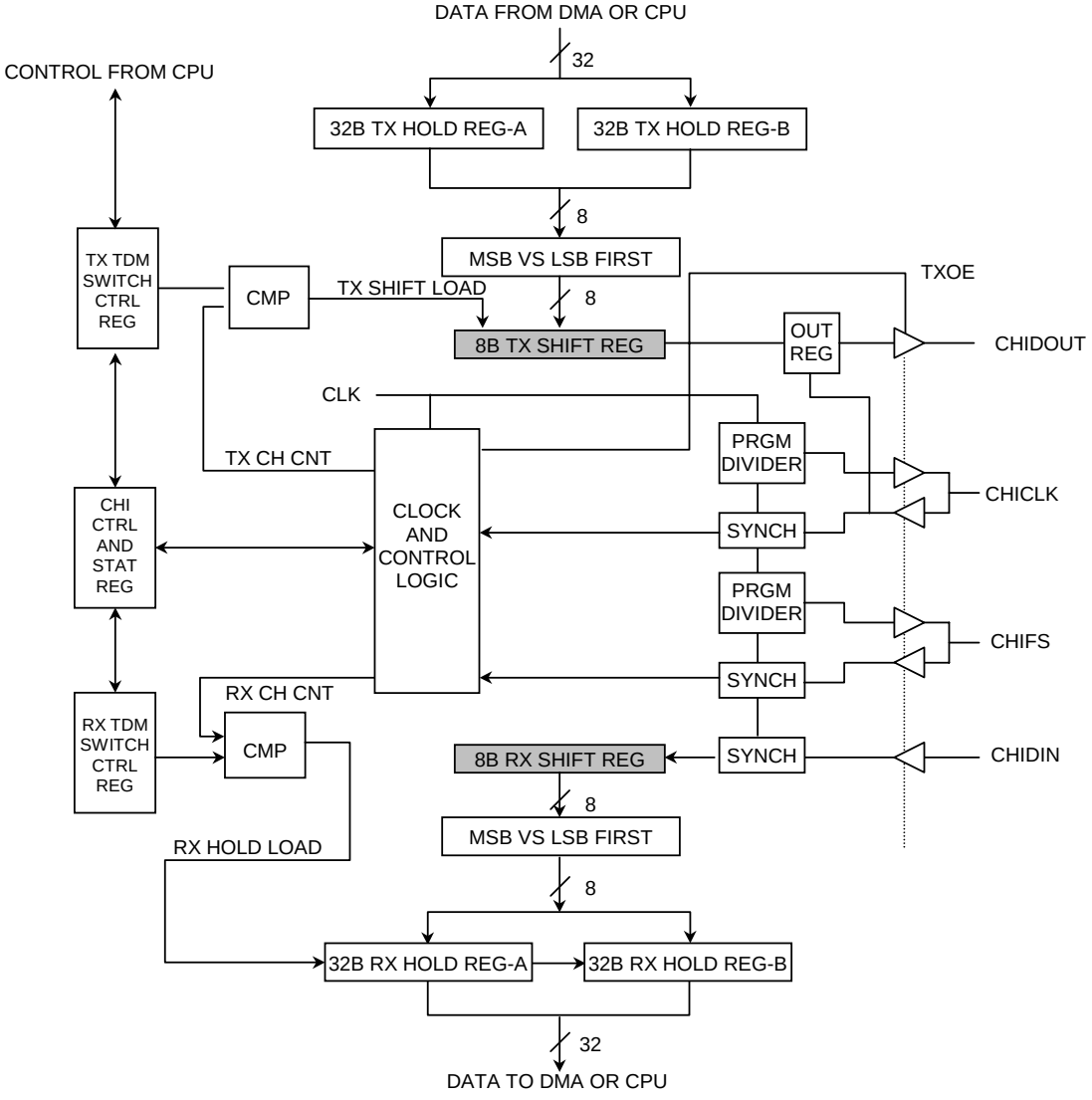


Figure 7.3.1 CHI Module Block Diagram

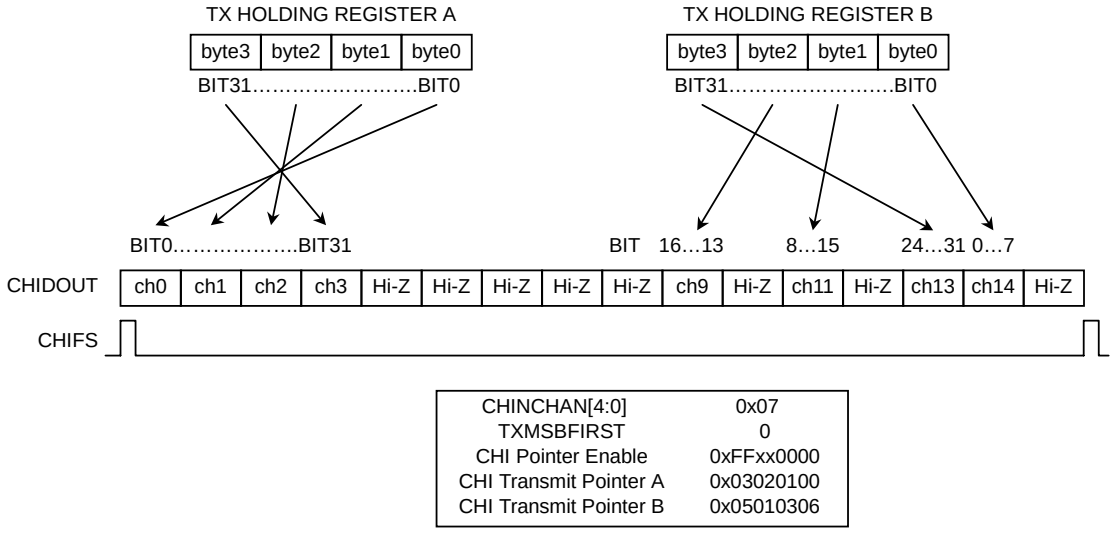
The CHI Module consists of holding registers (both transmit and receive), shift registers (both transmit and receive), DMA support, and other logic to support interfacing to various types of TDM highways. See Figure 7.3.1 for a block diagram of the CHI Module.

7.3.2 Transmitter

For the CHI transmit direction, Buffer-A and Buffer-B transmit holding registers are written either from the DMA circuit or directly from the CPU. Each of these 2 holding registers are 32-bits wide, and CHI control logic determines which byte from which holding register gets loaded at a given time into the 8-bit transmit shift register. In addition, the byte data loaded from the holding register to the shift register can be MSB-first or LSB-first. The reason for having Buffer-A and Buffer-B holding registers is that the CHI Module operates in a ping-pong fashion. Each frame of data is partitioned into 2 buffers (A and B); for example, with 64 timeslots total, the data is partitioned into 32 timeslots per buffer. The ping-pong operation allows one buffer to be updated (via the DMA or CPU) while the other buffer is being loaded into the shift register a byte at a time, depending on which timeslots are active. The ping-pong operation is transparent to the CPU or DMA interface, since the CHI Module automatically points to the correct A or B buffer at a given time and the CPU or DMA always accesses the same 32-bit holding register for all transactions.

The transmit TDM switch control register is used to select ANY 4 channels per buffer to be loaded from the holding register to the shift register. For example, if the CHI Module is configured for 32 timeslots per buffer (64 total timeslots), any 4 channels per buffer (8 total) can be selected out of the 32 available channels. The CHIDOUT signal is tri-stated during any of the non-selected channels. Each of the 8 selected channels also has an individual control bit for enabling/disabling the timeslot.

Figure 7.3.2 shows how the transmit TDM switch works, for example, with 16 total timeslots.



CHINCHAN[4:0]	0x07
TXMSBFIRST	0
CHI Pointer Enable	0xFFxx0000
CHI Transmit Pointer A	0x03020100
CHI Transmit Pointer B	0x05010306

Figure 7.3.2 Transmit TDM Switch Example

7.3.3 Receiver

For the CHI receive direction, Buffer-A and Buffer-B receive holding registers are read either by the DMA circuit or directly by the CPU. Each of these 2 holding registers are 32 bits wide, and CHI control logic determines which byte to which holding register gets loaded at a given time from the 8-bit receive shift register. In addition, the byte data loaded from the shift register to the holding register can be MSB-first or LSB-first. Similar to the transmit direction, the receive section also operates in a ping-pong fashion, allowing one buffer to be read (via the DMA or CPU) while the other buffer is being loaded from the shift register a byte at a time, depending on which timeslots are active.

The receive TDM switch control register (independent from the transmit TDM switch control register) is used to select ANY 4 channels per buffer to be loaded from the shift register to the holding register. Each of the 8 selected channels also has an individual control bit for enabling/disabling the timeslot.

An interrupt is available whenever a valid 32 bit word is available from the receive data holding register A. This also means a valid CHI output sample can be written to the transmit data holding register A. Similarly, an interrupt is also available whenever a valid 32 bit word is available from the receive data holding register B. This also means a valid CHI output sample can be written to the transmit data holding register B.

7.3.4 Clock and Control Generation

The CHI Module contains several programmable counters which are used to generate the various CHI internal and external control signals and clocks. See Figure 7.3.3 for a block diagram of the CHI clock and control generation circuit. As mentioned previously, CHICLK can be configured as either an output (master mode) or input (slave mode). As an output, CHICLK is derived by dividing down from CLK. In this mode, all CHI clocks are then synchronously locked to the main TMPR3911/12 system clock. As an input, CHICLK is generated from an external clock source, which is asynchronous with respect to CLK. The TMPR3911/12 CHI Module utilizes a digital-PLL circuit to stay “locked” to the external source, while still operating internally using CLK. CHIDIN and CHIDOUT are also synchronized between CLK and the externally-supplied CHICLK.

CHIFS can also be configured as either an output (master mode) or input (slave mode). As an output, CHIFS is derived by dividing down from CHICLK. For this mode, the CHIFS pulse width and polarity is also programmable. As an input, CHIFS is generated from an external sync source. The TMPR3911/12 CHI Module utilizes a digital-PLL circuit to stay “locked” to the external sync source, while still operating internally using CLK.

The programmable receive and transmit sync delay counters shown in Figure 7.3.3 are used to implement the bit offset feature described earlier. The bit offset control bits determine the number of clock cycles between the start of timeslot 0 and CHIFS. The receive and transmit sync delay counters are independent from each other, such that the receive and transmit serial data streams can have different bit offsets.

The programmable receive channel counter output is constantly compared with the receive TDM switch control register values, and whenever a match occurs, the byte of data is loaded from the receive shift register into the correct field within the receive holding register. Similarly, the programmable transmit channel counter output is constantly compared with the transmit TDM switch control register values, and whenever a match occurs, the byte of data is loaded from the correct field within the transmit holding register into the receive shift register.

All control registers, including the TDM switch control registers, must be unchanged while the CHI Module is enabled. If any register is changed during the CHI operation, the result is undefined.

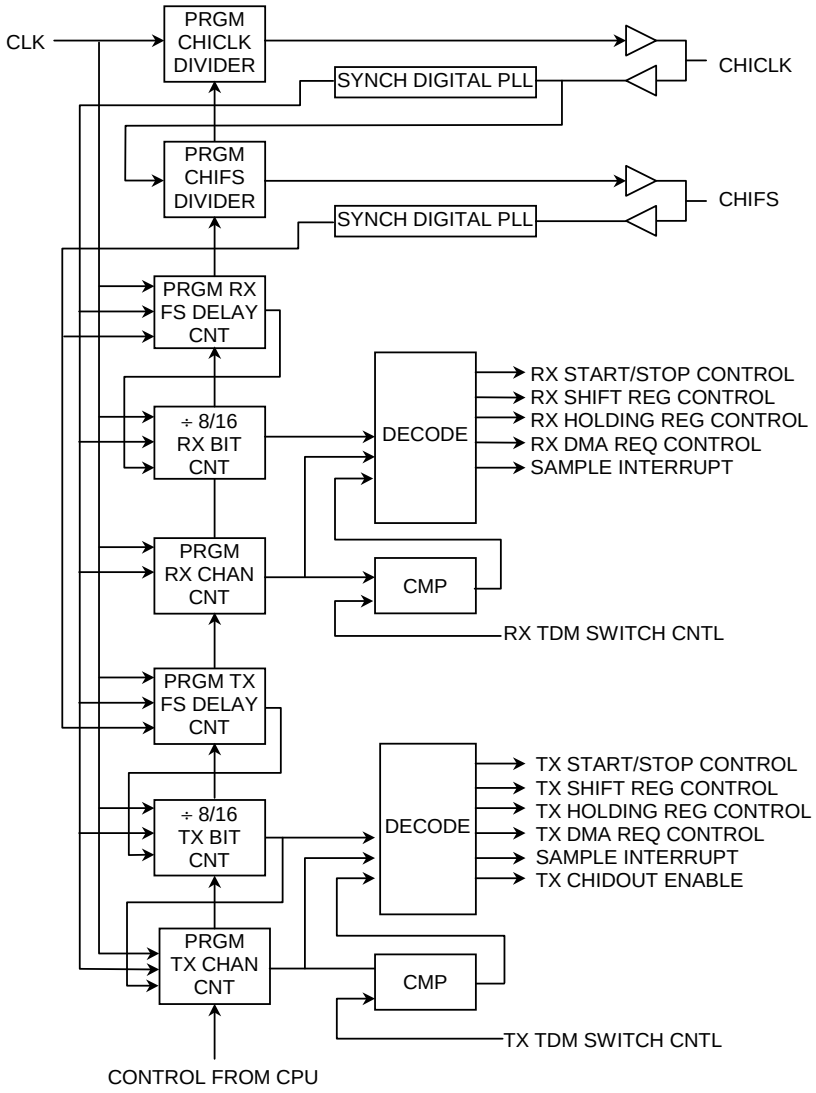


Figure 7.3.3 CHI Clock and Control Generation

7.3.5 DMA Address Generation

The CHI Module provides support for 2 full-duplex DMA channels: receive and transmit. The circuit used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts is shown in Figure 7.3.4.

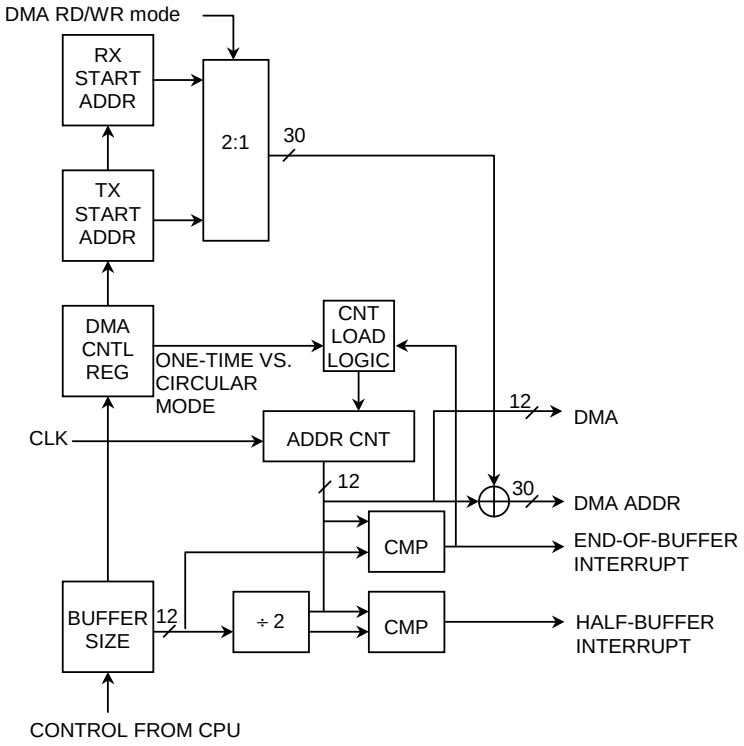


Figure 7.3.4 CHI DMA Address Generation

The DMA buffer size is programmable (up to a maximum of 16 Kbytes) and the receive and transmit buffer start addresses are also programmable (anywhere over the full 32-bit address space). Because there are separate start addresses, the receive and transmit buffers can be configured to either reside in different memory spaces or share the same memory space. The latter setup allows for overlapping buffers for loopback purposes or for optimum memory allocation, for which the DMA logic supports two full-duplex loopback modes. For one mode, receive DMA requests are issued first, followed by transmit DMA requests. This ordering allows a receive-to-transmit immediate loopback via the DMA buffer. For the second mode, transmit DMA requests are issued first, followed by receive DMA requests. Thus, received samples are written to the DMA buffer location immediately after transmit samples were read from that same location (which then became immediately available). This ordering allows a single circular DMA buffer to be used for both transmit and receive samples.

The DMA buffers can be configured in a circular buffer mode or a one-time buffer mode. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TMPR3911/12's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

Because the CHI Module reads and writes a byte at a time between the shift registers and the 32 bit holding registers, the software must pack and unpack these bytes to and from the 32 bit words in memory in order to multiplex and demultiplex each channel for processing. Table 7.3.1 shows the format and organization of the CHI channels within memory for DMA mode. Consecutive byte samples for a given channel reside in memory every 8th byte.

Note that the byte lanes are swapped between the little- and big-endian modes. In the little-endian mode, bits 31:24 of a word on the DMA buffer correspond to the byte lane with the address offset '+3' and therefore to the 'byte0' in the CHI TX/RX holding register. On the contrary, in the big-endian mode, bits 31:24 of a word on the DMA buffer correspond to the byte lane with the address offset '+0' and therefore to the 'byte3' in the CHI TX/RX holding register.

Table 7.3.1 CHI DMA Memory Organization

(relative) memory address	+0	+1	+2	+3
+0x0	buffA, byte3 sample 0	buffA, byte2 sample 0	buffA, byte1 sample 0	buffA, byte0 sample 0
+0x4	buffB, byte3 sample 0	buffB, byte2 sample 0	buffB, byte1 sample 0	buffB, byte0 sample 0
+0x8	buffA, byte3 sample 1	buffA, byte2 sample 1	buffA, byte1 sample 1	buffA, byte0 sample 1
+0xC	buffB, byte3 sample 1	buffB, byte2 sample 1	buffB, byte1 sample 1	buffB, byte0 sample 1
+0x10	buffA, byte3 sample 2	buffA, byte2 sample 2	buffA, byte1 sample 2	buffA, byte0 sample 2
etc.				

For a given channel, the minimum input-to-output latency for the CHI data path is (4 cycles) × (62.5 μs) = 250 μs, assuming an 8 kHz frame rate. These required 4 cycles are as follows:

- 1 cycle to load receive shift register into receive holding register
- 1 cycle for DMA of receive holding register data into receive memory space
- (insert here any application-specific time required for processing of received data and moving result to transmit memory space)
- 1 cycle for DMA of data in transmit memory space to transmit holding register
- 1 cycle to load transmit holding register data into transmit shift register

Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and utilize the DMA buffer in a ping-pong fashion. For transmit mode, the CPU can use these interrupts to fill or write one half of the buffer while the other half is being emptied by the DMA controller for transmitting out the CHI. Similarly, for receive mode, the CPU can use these interrupts to empty or read one half of the buffer while the other half is being filled by the DMA controller from received CHI input samples.

Also available is a direct CPU read/write mode for bypassing the DMA, allowing the CPU to read or write the CHI data on a sample by sample basis, if so desired. Separate DMA enables for receive and transmit allow DMA to be setup for receive only (transmit via CPU), transmit only (receive via CPU), receive and transmit, or none (receive and transmit via CPU).

The DMA circuit also provides an interrupt each time the DMA buffer pointer is incremented, which occurs whenever a new sample is read from and/or written to the DMA buffer. This interrupt may be useful for triggering a read of the DMA pointer status value, which is the actual 12-bit DMA address counter output. This value indicates exactly where the current address is pointing to in the overall DMA buffer.

7.3.6 Related Interrupts

CHI0_5INT:

Issues an interrupt whenever the CHI DMA buffer pointer has reached the halfway point.

CHI1_0INT:

Issues an interrupt whenever the CHI DMA buffer pointer has reached the end-of-buffer point.

CHIDMACNTINT:

Issues an interrupt each time the CHI DMA buffer pointer is incremented, which occurs whenever a new CHI sample is read from and/or written to the CHI DMA buffer.

CHIININTA:

Issues an interrupt whenever a valid CHI input sample is available from CHI RX Holding Register A; this also means a valid CHI output sample can be written to CHI TX Holding Register A.

CHIININTB:

Issues an interrupt whenever a valid CHI input sample is available from CHI RX Holding Register B; this also means a valid CHI output sample can be written to CHI TX Holding Register B.

CHIACTINT:

Issues an interrupt whenever CHICLK is active. This is used for CHI wakeup purposes.

CHIERRINT:

Issues an interrupt whenever a CHI error is received. This interrupt is triggered if CPU or DMA reading of the CHI RX Holding Registers does not keep up with the hardware filling of the CHI RX Holding Registers or if CPU or DMA writing of the CHI TX Holding Registers does not keep up with the hardware emptying of the CHI TX Holding Registers.

7.4 CHI Registers

7.4.1 CHI Control Register

OFFSET = $1D8:

Bit	Label	RESET	Read/Write
31-30	Reserved		
29	CHILOOP	0	R/W
28	CHIENTEST	0	R/W
27	CHIFSDIR	0	R/W
26-25	CHIFSWIDTH[1:0]	X	R/W
24-20	CHINCHAN[4:0]	X	R/W
19-16	CHITXBOFF[3:0]	X	R/W
15-12	CHIRXBOFF[3:0]	X	R/W
11	TXMSBFIRST	X	R/W
10	RXMSBFIRST	X	R/W
9	CHIRXFSPOL	X	R/W
8	CHITXFSPOL	X	R/W
7	CHIRXEDGE	X	R/W
6	CHITXEDGE	X	R/W
5	CHIFSEDGE	X	R/W
4	CHITXFSEDGE	X	R/W
3	CHICLK2XMODE	0	R/W
2	CHIRXEN	0	R/W
1	CHITXEN	0	R/W
0	ENCHI	0	R/W

CHILOOP:

This bit is used for IC testing and should not be set. Setting this bit to a logic “1” will cause the CHI serial transmitted data to be internally looped back to the CHI serial receive data path. The data is inverted when this mode is selected. Clearing this bit to a logic “0” selects the normal CHIDIN pin as the CHI serial receive data source.

CHIENTEST:

This bit is used for IC testing and should not be set.

CHIFSDIR:

This bit controls the direction of the CHIFS pin. Setting this bit to a logic “1” configures CHIFS to be an output (CHI sync master mode). Clearing this bit to a logic “0” configures CHIFS to be an input (CHI sync slave mode). Note that CHIRXFSPOL and CHIFSEDGE bits must be set to proper values even when CHIFSDIR is set to a logic “1” (CHI sync master mode).

CHIFSWIDTH[1:0]:

These bits are used to select pulse width for the CHIFS signal, relevant whenever the CHI Module is configured as master mode. The pulse width is counted by data bit width. (In clk2x mode, two CHICLKs correspond to one data bit.) The available CHIFS pulse widths are as follows:

CHIFSWIDTH	CHIFS pulse width
0	1 bit wide
1	2 bits wide
2	1 byte wide
3	half-frame wide

CHINCHAN[4:0]:

These bits are used to program the number of 8-bit channel timeslots per half-frame, up to 32 total per half-frame. The value loaded for CHINCHAN is the desired number of channels-1.

CHITXBOFF[3:0]:

These bits select the transmit data programmable bit offset, which is related to the number of clocks from the start of timeslot 0 (1st timeslot) transmit data to the CHIFS edge used to trigger the start of each CHI frame. The value loaded for CHITXBOFF must be chosen from Table 7.2.2 according to the configuration.

CHIRXBOFF[3:0]:

These bits select the receive data programmable bit offset, which is related to the number of clocks from the start of timeslot 0 (1st timeslot) receive data to the CHIFS edge used to trigger the start of each CHI frame. The value loaded for CHIRXBOFF must be chosen from Table 7.2.1 according to the configuration.

TXMSBFIRST:

This bit selects between MSB-first and LSB-first serial data formats for each byte of the CHI transmit data. Setting this bit to a logic “1” selects MSB-first. Clearing this bit to a logic “0” selects LSB-first.

RXMSBFIRST:

This bit selects between MSB-first and LSB-first serial data formats for each byte of the CHI receive data. Setting this bit to a logic “1” selects MSB-first. Clearing this bit to a logic “0” selects LSB-first.

CHIRXFSPOL:

This bit selects between positive (active high) or negative (active low) polarity for the received CHIFS signal pulse. This bit must be properly set in CHI sync master mode, as well as in CHI sync slave mode; In CHI sync master mode, CHIRXFSPOL must be equal to CHITXFSPOL. Setting this bit to a logic “1” selects negative polarity for the CHIFS pulse. In this case, the falling edge of the CHIFS pulse is used to trigger the start of the CHI frame period. Clearing this bit to a logic “0” selects positive polarity for the CHIFS pulse. In this case, the rising edge of the CHIFS pulse is used to trigger the start of the CHI frame period.

CHITXFSPOL:

This bit selects between positive (active high) or negative (active low) polarity for the transmitted CHIFS signal pulse, relevant whenever the CHI Module is configured as master mode. Setting this bit to a logic “1” selects negative polarity for the CHIFS pulse. Clearing this bit to a logic “0” selects positive polarity for the CHIFS pulse.

CHIRXEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive data CHIDIN. Setting this bit to a logic “1” selects rising edge. Clearing this bit to a logic “0” selects falling edge. CHIDIN must be stable before and after the specified CHICLK edge.

CHITXEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to clock out the transmit data CHIDOUT. Setting this bit to a logic “1” selects rising edge. Clearing this bit to a logic “0” selects falling edge.

CHIFSEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive frame sync CHIFS. This bit must be properly set in CHI sync master mode, as well as in CHI sync slave mode. Setting this bit to a logic “1” selects rising edge. Clearing this bit to a logic “0” selects falling edge. CHIFS must be stable before and after the specified CHICLK edge; In CHI sync master mode, CHIFSEDGE must be equal to inverted CHITXFSEDGE.

CHITXFSEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to clock out the transmit frame sync CHIFS, relevant whenever the CHI Module is configured as master mode. Setting this bit to a logic “1” selects rising edge. Clearing this bit to a logic “0” selects falling edge.

CHICLK2XMODE:

This bit selects between 1x and 2x clock modes. Setting this bit to a logic “1” selects 2x clock mode, which means that the CHICLK frequency equals twice the serial data bit rate. Clearing this bit to a logic “0” selects 1x clock mode, which means that the CHICLK frequency equals the serial data bit rate.

CHIRXEN:

This bit is used to enable/disable CHI receive processing in the direct CPU read/write mode, where the CPU reads the received data through the CHI RX holding register. Setting this bit to a logic “1” enables CHI receive processing. Clearing this bit to a logic “0” disables CHI receive processing, causing all received data to not be processed by the CHI module. This bit has no effect when RX DMA is enabled.

CHITXEN:

This bit is used to enable/disable CHI transmit processing in the direct CPU read/write mode, where the CPU writes the data to be transmitted through the CHI TX holding register. Setting this bit to a logic “1” enables CHI transmit processing. Clearing this bit to a logic “0” disables CHI transmit processing, causing the CHI serial transmitted data to be tri-stated. This bit has no effect when TX DMA is enabled.

ENCHI:

This bit is used to enable/disable the CHI module. Setting this bit to a logic “1” enables the CHI module. Clearing this bit to a logic “0” disables the CHI Module and keeps the module in a reset state but gives no effect on the CHI Control Register.

To begin the CHI operation, follow the procedure below.

1) Set up all the configuration registers except CHIRXEN, CHITXEN, ENDMARXCHI, ENDMATXCHI, and ENCHI bits.
2) Set either CHIRXEN or ENDMARXCHI, and/or, either CHITXEN or ENDMATXCHI, to a logic “1”.
3) Set ENCHI to a logic “1”.

To finish the CHI operation.

1) Clear ENCHI to a logic “0”.
2) Clear ENDMARXCHI and/or ENDMATXCHI to a logic “0”, if they were previously set.

7.4.2 CHI Pointer Enable Register

OFFSET = $1DC:

Bit	Label	RESET	Read/Write
31	CHITXPTRB3EN	X	R/W
30	CHITXPTRB2EN	X	R/W
29	CHITXPTRB1EN	X	R/W
28	CHITXPTRB0EN	X	R/W
27	CHITXPTRA3EN	X	R/W
26	CHITXPTRA2EN	X	R/W
25	CHITXPTRA1EN	X	R/W
24	CHITXPTRA0EN	X	R/W
23	CHIRXPTRB3EN	X	R/W
22	CHIRXPTRB2EN	X	R/W
21	CHIRXPTRB1EN	X	R/W
20	CHIRXPTRB0EN	X	R/W
19	CHIRXPTRA3EN	X	R/W
18	CHIRXPTRA2EN	X	R/W
17	CHIRXPTRA1EN	X	R/W
16	CHIRXPTRA0EN	X	R/W
15-0	Reserved	X	R/W

CHITXPTRB3EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB3. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRB2EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB2. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRB1EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB1. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRB0EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB0. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRA3EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA3. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRA2EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA2. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRA1EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA1. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHITXPTRA0EN:

This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA0. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRB3EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB3. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRB2EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB2. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRB1EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB1. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRB0EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB0. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRA3EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA3. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRA2EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA2. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRA1EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA1. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

CHIRXPTRA0EN:

This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA0. Setting this bit to a logic “1” enables the timeslot. Clearing this bit to a logic “0” disables the timeslot.

7.4.3 CHI Receive Pointer A Register

OFFSET = $1E0: write-only

Bit	Label	RESET	Read/Write
31-29	Reserved		
28-24	CHIRXPTRA3[4:0]	X	W
23-21	Reserved		
20-16	CHIRXPTRA2[4:0]	X	W
15-13	Reserved		
12-8	CHIRXPTRA1[4:0]	X	W
7-5	Reserved		
4-0	CHIRXPTRA0[4:0]	X	W

CHIRXPTRA3[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 3 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHIRXPTRA2[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 2 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHIRXPTRA1[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 1 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHIRXPTRA0[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 0 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

7.4.4 CHI Receive Pointer B Register

OFFSET = $1E4: write-only

Bit	Label	RESET	Read/Write
31-29	Reserved		
28-24	CHIRXPTRB3[4:0]	X	W
23-21	Reserved		
20-16	CHIRXPTRB2[4:0]	X	W
15-13	Reserved		
12-8	CHIRXPTRB1[4:0]	X	W
7-5	Reserved		
4-0	CHIRXPTRB0[4:0]	X	W

CHIRXPTRB3[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 3 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHIRXPTRB2[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 2 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHIRXPTRB1[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 1 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHIRXPTRB0[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 0 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

7.4.5 CHI Transmit Pointer A Register

OFFSET = $1E8: write-only

Bit	Label	RESET	Read/Write
31-29	Reserved		
28-24	CHITXPTRA3[4:0]	X	W
23-21	Reserved		
20-16	CHITXPTRA2[4:0]	X	W
15-13	Reserved		
12-8	CHITXPTRA1[4:0]	X	W
7-5	Reserved		
4-0	CHITXPTRA0[4:0]	X	W

CHITXPTRA3[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 3 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHITXPTRA2[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 2 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHITXPTRA1[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 1 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

CHITXPTRA0[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 0 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

7.4.6 CHI Transmit Pointer B Register

OFFSET = $1EC: write-only

Bit	Label	RESET	Read/Write
31-29	Reserved		
28-24	CHITXPTRB3[4:0]	X	W
23-21	Reserved		
20-16	CHITXPTRB2[4:0]	X	W
15-13	Reserved		
12-8	CHITXPTRB1[4:0]	X	W
7-5	Reserved		
4-0	CHITXPTRB0[4:0]	X	W

CHITXPTRB3[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 3 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2−1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHITXPTRB2[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 2 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2−1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHITXPTRB1[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 1 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2−1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

CHITXPTRB0[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 0 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2−1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

7.4.7 CHI Size Register

OFFSET = $1F0:

Bit	Label	RESET	Read/Write
31-30	Reserved		
29-18	CHIDMAPTR[13:2]	–	R
17-16	Reserved		
15	CHIBUFF1TIME	0	R/W
14	CHIDMALOOP	0	R/W
13-2	CHISIZE[13:2]	X	W
1	ENDMARXCHI	0	R/W
0	ENDMATXCHI	0	R/W

CHIDMAPTR[13:2]: read-only

These bits provide the status of the CHI DMA counter.

CHIBUFF1TIME:

The CHI DMA controller supports two buffer addressing modes depending on the state of this bit. When CHIBUFF1TIME is set to a logic “1”, the CHI DMA controller will stop executing when it reaches the end of the DMA buffer. When CHIBUFF1TIME is cleared to a logic “0”, the CHI DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner.

CHIDMALOOP:

The CHI DMA controller supports two full-duplex loopback modes depending on the state of this bit. When CHIDMALOOP is set to a logic “1”, the CHI DMA controller issues RX DMA requests first, followed by TX DMA requests. This ordering allows an RX-to-TX immediate loopback via the DMA buffer. When CHIDMALOOP is cleared to a logic “0”, the CHI DMA controller issues TX DMA requests first, followed by RX DMA requests. This ordering allows a single circular DMA buffer to be used for both TX and RX, if so desired.

CHISIZE (13:2): write-only

These bits define the size of the CHI DMA buffers (16 Kbytes maximum). Both the CHI RX buffer and the CHI TX buffer are the same size. The last address in the CHI RX DMA buffer is given by CHIRXSTART[31:2] + CHISIZE[13:2]. The last address in the CHI TX DMA buffer is given by CHITXSTART[31:2] + CHISIZE[13:2]. The value loaded into CHISIZE should be equal to the desired buffer length –1.

ENDMARXCHI:

This bit enables the CHI DMA receive function. Setting this bit to a logic “1” enables the DMA mode. Clearing this bit to a logic “0” disables the DMA mode. This bit should not be set until the CHIRXSTART, CHITXSTART, and CHISIZE registers are setup. To enable CHI DMA receive function, this bit must be set before the CHI Module is enabled (ENCHI asserted). Either ENDMARXCHI or ENDMATXCHI or both can be set at a time since the CHI DMA controller can support full duplex operation.

ENDMATXCHI:

This bit enables the CHI DMA transmit function. Setting this bit to a logic "1" enables the DMA mode. Clearing this bit to a logic "0" disables the DMA mode. This bit should not be set until the CHIRXSTART, CHITXSTART, and CHISIZE registers are setup. To enable CHI DMA transmit function, this bit must be set before the CHI Module is enabled (ENCHI asserted). Either ENDMARXCHI or ENDMATXCHI or both can be set at a time since the CHI DMA controller can support full duplex operation.

7.4.8 CHI RX Start Register

OFFSET = $1F4: write-only

Bit	Label	RESET	Read/Write
31-2	CHIRXSTART[31:2]	X	W
1-0	Reserved		

CHIRXSTART[31:2]: write-only

These bits define the start address for the CHI RX DMA buffer. The CHI RX buffer and CHI TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

7.4.9 CHI TX Start Register

OFFSET = $1F8: write-only

Bit	Label	RESET	Read/Write
31-2	CHITXSTART[31:2]	X	W
1-0	Reserved		

CHITXSTART[31:2]: write-only

These bits define the start address for the CHI TX DMA buffer. The CHI RX buffer and CHI TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

7.4.10 CHI TX Holding Register

OFFSET = $1FC: write-only

Bit	Label	RESET	Read/Write
31-0	CHITXHOLD[31:0]	X	W

CHITXHOLD[31:0]: write-only

These bits represent the CHI data to be transmitted. CHI data can be either written directly to this register by the CPU or transparently read from the CHI TX DMA buffer to this register. This register should only be loaded by the CPU after the CHIININTA or CHIININTB interrupt is asserted. The write immediately after CHIININTA updates the internal TX holding register A and the write immediately after CHIININTB updates the internal TX holding register B. Transmit data for bytes 3, 2, 1, and 0 are loaded into the 32-bit CHITXHOLD at locations [31:24], [23:16], [15:8], and [7:0], respectively. These data bytes correspond to the CHI timeslots as defined by the values in the CHITXPTRA and CHITXPTRB TDM switch registers.

7.4.11 CHI RX Holding Register

OFFSET = $1FC: read-only

Bit	Label	RESET	Read/Write
31-0	CHIRXHOLD[31:0]	X	R

CHIRXHOLD[31:0]: read-only

These bits represent the CHI data to be received. CHI data can be either read directly from this register by the CPU or transparently written to the CHI RX DMA buffer from this register. This register should only be read by the CPU after the CHIININTA or CHIININTB interrupt is asserted. The read immediately after CHIININTA sees the internal RX holding register A and the read immediately after CHIININTB sees the internal RX holding register B. Receive data for bytes 3, 2, 1, and 0 are stored into the 32-bit CHIRXHOLD at locations [31:24], [23:16], [15:8], and [7:0], respectively. These data bytes correspond to the CHI timeslots as defined by the values in the CHIRXPTRA and CHIRXPTRB TDM switch registers.

8. Interrupt Module

8.1 Overview

The Interrupt Module within the TMPR3911/12 contains logic for reading, enabling, and clearing all of the interrupt sources. These interrupts are either generated from internal TMPR3911/12 modules or from edge transitions on external signal pins. All of these latter interrupt types generate a separate positive and negative edge interrupt.

The status (logic state) of each interrupt is readable, allowing any interrupt event to be polled, if desired. Each interrupt status signal is also gated with the corresponding interrupt enable bit in order to generate the overall IRQ output to the CPU. The interrupt status register is reset using the corresponding clear bit from the corresponding clear interrupt register.

8.2 Implementation

8.2.1 Block Diagram

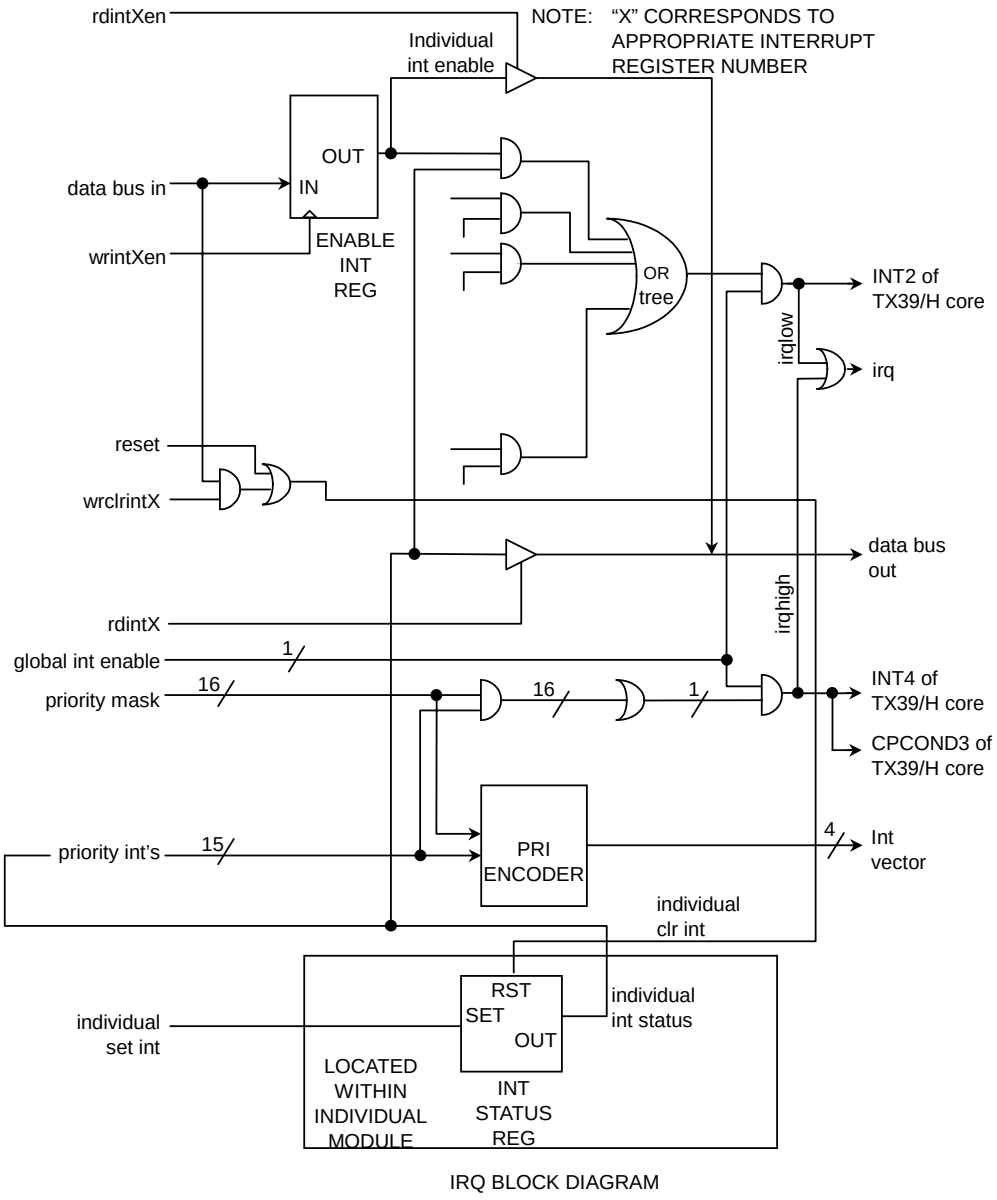


Figure 8.2.1 Interrupt Circuit Block Diagram

8.2.2 Interrupt Logic Description

Figure 8.2.1 shows a block diagram of the interrupt circuit for a single interrupt source. Every other interrupt source contains an identical circuit.

The overall Interrupt Module contains 5 main sets of registers (numbered 1 through 5), with a 6th register allocated for setting the global interrupt enable and the 16-bit Priority Mask, and for reading the Priority Interrupt Vector and IRQHIGH and IRQLOW status bits. Each bit within each of the 5 main sets then consists of 3 registers: Status, Enable, and Clear.

- Interrupt Status Register read-only
  - allows CPU to read logic state of interrupt status
  - informs the CPU which interrupts are pending
- Enable Interrupt Register read/write
  - allows CPU to enable a given interrupt
  - each individual interrupt has a corresponding enable bit
- Clear Interrupt Register write-only
  - allows CPU to clear a given pending interrupt
  - each individual interrupt has a corresponding clear bit
  - each individual interrupt clear address and bit position maps one-to-one with the interrupt status address and bit position (status transactions are CPU reads, clear transactions are CPU writes)

A given Interrupt Status Register is set by the corresponding individual interrupt event. For example, this event could be a positive (or negative) edge transition on an external TMPR3911/12 pin. Another example might be an event triggered by an internal TMPR3911/12 module, such as a particular DMA buffer reaching the end-of-buffer point.

The output of a given Interrupt Status Register is readable by the CPU, by reading the address which points to the corresponding Interrupt Status Register. A given interrupt is enabled by writing a “1” to the corresponding bit position in the corresponding Enable Interrupt Register. The Interrupt Status Register output bit is then gated with the enable bit from the corresponding Enable Interrupt Register. The output of this gate is then combined with all the other gated Interrupt Status bits using a wide bitwise “OR” tree in order to generate the IRQLOW status signal. In other words, if ANY of the gated Interrupt Status signals is asserted and the global interrupt enable GLOBALEN is asserted, then IRQLOW will be asserted.

A given pending interrupt is cleared by writing a “1” to the corresponding bit position in the corresponding Clear Interrupt Register. The output of this Clear Interrupt Register is used to reset the corresponding Interrupt Status Register.

On power on reset, the Global Enable is cleared to prevent any interrupts from occurring until the Enable Interrupt Registers are initialized. Also, all Clear Interrupts will be set for the duration of reset (thus interrupts cleared).

The Interrupt Module also contains 15 High Priority Interrupt sources. A 16-bit Priority Mask (located within the Enable Interrupt 6 Register) is used to enable any of these high priority interrupts, by writing a “1” to the corresponding bit position in the Enable Interrupt 6 Register, where bit 15 is the highest priority. Each bit of the 16-bit Priority Mask is gated with the corresponding 15 High Priority Interrupt signals and these gated signals are then combined using a bitwise 16-input “OR” tree in order to generate the IRQHIGH status signal. If IRQHIGH or IRQLOW is asserted, then the overall IRQ will be asserted.

A Priority Encoder circuit also compares the 16-bit Priority Mask with the 15 High Priority Interrupt signals and generates a 4-bit interrupt status vector which points to the highest priority pending interrupt from the set of 16 possible events. Level 15 is the highest priority and Level 1 is the lowest priority, with Level 0 corresponding to the standard interrupt handler.

The IRQHIGH signal is connected to interrupt bit 4 on the TX39/H Processor Core and the IRQLOW signal is connected to interrupt bit 2. The IRQHIGH signal is also connected to CPU Co-Processor Condition bit 3.

8.3 Interrupt Registers

8.3.1 Interrupt Status 1 Register

OFFSET = $100: read-only

Bit	Label	RESET	Read/Write
31	LCDINT	–	R
30	DFINT	–	R
29	CHI0_5INT	–	R
28	CHI1_0INT	–	R
27	CHIDMACNTINT	–	R
26	CHININTA	–	R
25	CHININTB	–	R
24	CHIACTINT	–	R
23	CHIERRINT	–	R
22	SND0_5INT	–	R
21	SND1_0INT	–	R
20	TEL0_5INT	–	R
19	TEL1_0INT	–	R
18	SNDDMACNTINT	–	R
17	TELDMACNTINT	–	R
16	LSNDCLIPINT	–	R
15	RSNDCLIPINT	–	R
14	VALSNDPOSINT	–	R
13	VALSNDNEGINT	–	R
12	VALTELPOSINT	–	R
11	VALTELNEGINT	–	R
10	SNDININT	–	R
9	TELININT	–	R
8	SIBSF0INT	–	R
7	SIBSF1INT	–	R
6	SIBIRQNEGINT	–	R
5	SIBIRQNEGINT	–	R
4-0	Reserved	–	R

LCDINT:

Issues an interrupt at the end of each video frame.

DFINT:

Issues an interrupt each time the video DF signal toggles.

CHI0_5INT:

Issues an interrupt whenever the CHI DMA buffer pointer has reached the halfway point.

CHI1_0INT:

Issues an interrupt whenever the CHI DMA buffer pointer has reached the end-of-buffer point.

CHIDMACNTINT:

Issues an interrupt each time the CHI DMA buffer pointer is incremented, which occurs whenever a new CHI sample is read from and/or written to the CHI DMA buffer.

CHIININTA:

Issues an interrupt whenever a valid CHI input sample is available from CHI RX Holding Register A; this also means a valid CHI output sample can be written to CHI TX Holding Register B.

CHIININTB:

Issues an interrupt whenever a valid CHI input sample is available from CHI RX Holding Register B; this also means a valid CHI output sample can be written to CHI TX Holding Register A.

CHIACTINT:

Issues an interrupt whenever CHICLK is active. This is used for CHI wakeup purposes.

CHIERRINT:

Issues an interrupt whenever a CHI error is received. This interrupt is triggered if CPU or DMA reading of the CHI RX Holding Registers does not keep up with the hardware filling of the CHI RX Holding Registers or if CPU or DMA writing of the CHI TX Holding Registers does not keep up with the hardware emptying of the CHI TX Holding Registers.

SND0_5INT:

Issues an interrupt whenever the sound DMA buffer pointer has reached the halfway point.

SND1_0INT:

Issues an interrupt whenever the sound DMA buffer pointer has reached the end-of-buffer point.

TEL0_5INT:

Issues an interrupt whenever the telecom DMA buffer pointer has reached the halfway point.

TEL1_0INT:

Issues an interrupt whenever the telecom DMA buffer pointer has reached the end-of-buffer point.

SNDDMACNTINT:

Issues an interrupt each time the sound DMA buffer pointer is incremented, which occurs whenever a new sound sample is read from and/or written to the sound DMA buffer.

TELDMACNTINT:

Issues an interrupt each time the telecom DMA buffer pointer is incremented, which occurs whenever a new telecom sample is read from and/or written to the telecom DMA buffer.

LSNDCLIPINT:

Issues an interrupt whenever the amplitude of the left channel sound data is clipping the codec A/D converter for SIB subframe 1.

RSNDCLIPINT:

Issues an interrupt whenever the amplitude of the right channel sound data is clipping the codec A/D converter for SIB subframe 1.

VALSNDPOSINT:

Issues an interrupt whenever the valid sound status flag transitions from a logic “0” to a logic “1”. This valid flag is triggered from SIB subframe 0 (if SELSNDSF1 = “0”) or from SIB subframe 1 (if SELSNDSF1 = “1”).

VALSNDNEGINT:

Issues an interrupt whenever the valid sound status flag transitions from a logic “1” to a logic “0”. This valid flag is triggered from SIB subframe 0 (if SELSNDSF1 = “0”) or from SIB subframe 1 (if SELSNDSF1 = “1”).

VALTELPOSINT:

Issues an interrupt whenever the valid telecom status flag transitions from a logic “0” to a logic “1”. This valid flag is triggered from SIB subframe 0 (if SELTELSF1 = “0”) or from SIB subframe 1 (if SELTELSF1 = “1”).

VALTELNEGINT:

Issues an interrupt whenever the valid telecom status flag transitions from a logic “1” to a logic “0”. This valid flag is triggered from SIB subframe 0 (if SELTELSF1 = “0”) or from SIB subframe 1 (if SELTELSF1 = “1”).

SNDININT:

Issues an interrupt whenever a valid sound input longword (32 bits) is available from the Sound RX Holding Register; this also means a valid sound output longword can be written to the Sound TX Holding Register.

TELININT:

Issues an interrupt whenever a valid telecom input longword (32 bits) is available from the Telecom RX Holding Register; this also means a valid telecom output longword can be written to the Telecom TX Holding Register.

SIBSF0INT:

Issues an interrupt at the start of every SIB subframe 0. This is used to initiate CPU reading of the SIB Subframe 1 Status Register (SF1STAT Register) and/or CPU writing of the SIB Subframe 0 Control Register (SF0AUX Register).

SIBSF1INT:

Issues an interrupt at the start of every SIB subframe 1. This is used to initiate CPU reading of the SIB Subframe 0 Status Register (SF0STAT Register) and/or CPU writing of the SIB Subframe 1 Control Register (SF1AUX Register).

SIBIRQPOSINT:

Issues an interrupt whenever the SIBIRQ pin transitions from a logic “0” to a logic “1”.

SIBIRQNEGINT:

Issues an interrupt whenever the SIBIRQ pin transitions from a logic “1” to a logic “0”.

8.3.2 Interrupt Status 2 Register

OFFSET = $104: read-only

Bit	Label	RESET	Read/Write
31	UARTARXINT	–	R
30	UARTARXOVERRUNINT	–	R
29	UARTAFRAMEERRINT	–	R
28	UARTABREAKINT	–	R
27	UARTAPARITYERRINT	–	R
26	UARTATXINT	–	R
25	UARTATXOVERRUNINT	–	R
24	UARTAEMPTYINT	–	R
23	UARTADMAFULLINT	–	R
22	UARTADMAHALFINT	–	R
21	UARTBRXINT	–	R
20	UARTBRXOVERRUNINT	–	R
19	UARTBFRAMEERRINT	–	R
18	UARTBBREAKINT	–	R
17	UARTBPARITYERRINT	–	R
16	UARTBTXINT	–	R
15	UARTBTXOVERRUNINT	–	R
14	UARTBEMPTYINT	–	R
13	UARTBDMAFULLINT	–	R
12	UARTBDMAHALFINT	–	R
11	MBUSTXBUFAVALLINT	–	R
10	MBUSTXERRINT	–	R
9	MBUSEMPTYINT	–	R
8	MBUSRXBUFAVALLINT	–	R
7	MBUSRXERRINT	–	R
6	MBUSDETINT	–	R
5	MBUSDMAFULLINT	–	R
4	MBUSDMAHALFINT	–	R
3	MBUSPOSINT	–	R
2	MBUSNEGINT	–	R
1-0	Reserved	–	R

UARTARXINT:

Issues an interrupt if the UARTA Receive Holding Register is loaded with data.

UARTARXOVERRUNINT:

Issues an interrupt if the UARTA Receive Holding Register is loaded twice before the interrupt is service.

UARTAFRAMEERRINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register contains a frame error.

UARTABREAKINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register is a break.

UARTAPARITYERRINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register contains a parity error.

UARTATXINT:

Issues an interrupt if the UARTA Transmit Holding Register is available.

UARTATXOVERRUNINT:

Issues an interrupt if the UARTA Transmit Holding Register is written to when the Transmit Holding Register is not available.

UARTAEMPTYINT:

Issues an interrupt if the UARTA Transmit Holding Register and Transmit Shift Register are both empty.

UARTADMAFULLINT:

Issues an interrupt if the UARTA DMA counter reaches the end of the buffer.

UARTADMAHALFINT:

Issues an interrupt if the UARTA DMA counter reaches the mid point of the buffer.

UARTBRXINT:

Issues an interrupt if the UARTB Receive Holding Register is loaded with data.

UARTBRXOVERRUNINT:

Issues an interrupt if the UARTB Receive Holding Register is loaded twice before the interrupt is service.

UARTBFRAMEERRINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register contains a frame error.

UARTBBREAKINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register is a break.

UARTBPARITYERRINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register contains a parity error.

UARTBTXINT:

Issues an interrupt if the UARTB Transmit Holding Register is available.

UARTBTXOVERRUNINT:

Issues an interrupt if the UARTB Transmit Holding Register is written to when the Transmit Holding Register is not available.

UARTBEMPTYINT:

Issues an interrupt if the UARTB Transmit Holding Register and Transmit Shift Register are both empty.

UARTBDMAFULLINT:

Issues an interrupt if the UARTB DMA counter reaches the end of the buffer.

UARTBDMAHALFINT:

Issues an interrupt if the UARTB DMA counter reaches the mid point of the buffer.

MBUSTXBUFAVAILINT:

This interrupt is set when the ENMBUS bit is first asserted and subsequently when the contents of the MBUS Transmitter Holding Register are transferred to the MBUS Shift Register. This interrupt is used to indicate that the MBUS Transmitter Holding Register is available to be written by the software.

MBUSTXERRINT:

Issues an interrupt if the Magicbus Transmit Holding Register is written to when the Transmit Holding Register is not available.

MBUSEMPTYINT:

Issues an interrupt if the Magicbus Transmit Holding Register and Transmit Shift Register are both empty.

MBUSRXBUFAVAILINT:

Issues an interrupt if the Magicbus Receive Holding Register is loaded with data.

MBUSRXERRINT:

Issues an interrupt if the Magicbus Receive Holding Register is loaded twice before the interrupt is serviced.

MBUSDETINT:

Issues an interrupt if the Magicbus Command Word detector triggers.

MBUSDMAFULLINT:

Issues an interrupt if the Magicbus DMA Counter reaches the end of the buffer.

MBUSDMAHALFINT:

Issues an interrupt if the Magicbus DMA Counter reaches the mid point of the buffer.

MBUSPOSINT:

Issues an interrupt if the MBUSINT signal goes from a logic “0” to a logic “1”.

MBUSNEGINT:

Issues an interrupt if the MBUSINT signal goes from a logic “1” to a logic “0”.

8.3.3 Interrupt Status 3 Register

OFFSET = $108: read-only

Bit	Label	RESET	Read/Write
31-0	MFIOPOSINT[31:0]	–	R

MFIOPOSINT[31:0]:

Issues an interrupt whenever any of the multi-function I/O pin(s) transition from a logic “0” to a logic “1”. There are a total of 32 multi-function I/O pins (31 thru 0), each of which corresponds to the respective bit within this Status Register. Each multi-function I/O pin can independently trigger an interrupt.

8.3.4 Interrupt Status 4 Register

OFFSET = $10C: read-only

Bit	Label	RESET	Read/Write
31-0	MFIONEGINT[31:0]	–	R

MFIONEGINT[31:0]:

Issues an interrupt whenever any of the multi-function I/O pin(s) transition from a logic “1” to a logic “0”. There are a total of 32 multi-function I/O pins (31 thru 0), each of which corresponds to the respective bit within this Status Register. Each multi-function I/O pin can independently trigger an interrupt.

8.3.5 Interrupt Status 5 Register

OFFSET = $110: read-only

Bit	Label	RESET	Read/Write
31	RTCINT	–	R
30	ALARMINT	–	R
29	PERINT	–	R
28	STPTIMERINT	–	R
27	POSPWRINT	–	R
26	NEGPWRINT	–	R
25	POSPWROKINT	–	R
24	NEGPWROKINT	–	R
23	POSONBUTNINT	–	R
22	NEGONBUTNINT	–	R
21	SPIBUFAVAILINT	–	R
20	SPIERRINT	–	R
19	SPIRCVINT	–	R
18	SPIEMPTYINT	–	R
17	IRCONSMINT	–	R
16	CARSTINT	–	R
15	POSCARINT	–	R
14	NEGCARINT	–	R
13-7	IOPOSINT[6:0]	–	R
6-0	IONEGINT[6:0]	–	R

RTCINT:

This interrupt is set whenever all 40 bits of the RTC counter reaches a value of “$FFFFFFFFFF” to alert the software that the counter is “rolling over”.

ALARMINT:

This interrupt is set whenever the RTC counter reaches a count that is equal to the value of the ALARM[39:0] bits set in the Alarm Register.

PERINT:

This interrupt is set whenever the Periodic Timer is enabled and the Periodic Timer counter reaches a count of zero.

STPTIMERINT:

This interrupt is set whenever the Stop Timer Counter counts up to the value set by the STPTIMERVAL[3:0] control bits.

STPTIMERINT:

This interrupt is set whenever the Stop Timer Counter counts up to the value set by the STPTIMERVAL[3:0] control bits.

POSPWRINT:

This interrupt is set whenever the PWRINT pin transitions from a logic “0” to a logic “1”.

NEGPWRINT:

This interrupt is set whenever the PWRINT pin transitions from a logic “1” to a logic “0”.

POSPWROKINT:

Issues an interrupt whenever the PWROK signal transitions from a logic “0” to a logic “1”.

NEGPWROKINT:

Issues an interrupt whenever the PWROK signal transitions from a logic “1” to a logic “0”.

POSONBUTNINT:

Issues an interrupt whenever the ONBUTN signal transitions from a logic “0” to a logic “1”. If the DBNCONBUTN control bit is set, then the interrupt will not set until the signal is debounced for 16-24 ms.

NEGONBUTNINT:

Issues an interrupt whenever the ONBUTN signal transitions from a logic “1” to a logic “0”. If the DBNCONBUTN control bit is set, then the interrupt will not set until the signal is debounced for 16-24 ms.

SPIBUFAVAILINT:

This interrupt is set whenever the ENSPI bit is first asserted and subsequently when the contents of the SPI Transmitter Holding Register are transferred to the SPI Shift Register. This interrupt is used to indicate that the SPI Transmitter Holding Register is available to be written by the software.

SPIERRINT:

This interrupt is set whenever the SPI Transmitter Holding Register is written, but the SPIBUFAVAILINT has not set to indicate that the register is available. This interrupt serves as an overrun indication for the software.

SPIRCVINT:

This interrupt is set whenever the contents of the SPI Shift Register are transferred to the SPI Receiver Holding Register. This interrupt is used to indicate that there is valid data in the SPI Receiver Holding Register to be read by the software.

SPIEMPTYINT:

This interrupt is set whenever the both the SPI Shift Register and the SPI Transmitter Holding Register are empty. This interrupt can be used by the software to determine when the SPI is idle.

IRCONSMINT:

Whenever the upper byte of data is loaded into the 7-bit Period Number Counter, the lower byte of data is loaded into an intermediate holding register. At this time the 16-bit IR Holding Register is empty. The IRCONSMINT interrupt is then set to inform the CPU that the IR Holding Register is available. The CPU must fill the next word of data into the IR Holding Register before the 7-bit Period Number Counter is finished counting the periods for both the upper and lower bytes of data.

CARSTINT:

This interrupt is set whenever the Carrier Detect State Machine detects the CARDET pin is high just before turning off the RXPWR pin.

POSCARINT:

This interrupt is set whenever CARDET pin transitions from a logic “0” to a logic “1”.

NEGCARINT:

This interrupt is set whenever CARDET pin transitions from a logic “1” to a logic “0”.

IOPOSINT (6:0):

Issues an interrupt whenever any of the general purpose I/O pin(s) transitions from a logic “0” to a logic “1”. There are a total of 7 general purpose I/O pins (6 thru 0), each of which corresponds to the respective bit within this Status Register. Each general purpose I/O pin can independently trigger an interrupt.

IONEGINT (6:0):

Issues an interrupt whenever any of the general purpose I/O pin(s) transitions from a logic “1” to a logic “0”. There are a total of 7 general purpose I/O pins (6 thru 0), each of which corresponds to the respective bit within this Status Register. Each general purpose I/O pin can independently trigger an interrupt.

8.3.6 Interrupt Status 6 Register

OFFSET = $114: read-only

Bit	Label	RESET	Read/Write
31	IRQHIGH	–	R
30	IRQLOW	–	R
29	PWROKNMI	0	R/W
28-6	Reserved		
5-2	INTVECT[3:0]	–	R
1-0	Reserved		

IRQHIGH:

This status bit is the bitwise “OR” of all the 16 possible high priority interrupts.

IRQLOW:

This status bit is the bitwise “OR” of all the interrupts contained in the Interrupt Status 1, 2, 3, 4, and 5 Registers.

PWROKNMI:

When this bit is set, input to the PWROK pin can be used as the NMI source for the TX39/H core. When the bit is cleared, NMI for the TX39/H core is disabled.

INTVECT[3:0]:

These 4 status bits indicate a vector pointing to the highest priority pending interrupt from the set of 16 possible high priority interrupt events. The set of high priority interrupts is listed below, where level 15 is the highest priority and level 1 is the lowest priority. Level 0 corresponds to the standard interrupt handler.

priority level	high priority interrupt source
15	POSPWROKINT or NEGPWROKINT
14	ALARMINT
13	PERINT
12	MBUSPOSINT or MBUSNEGINT
11	UARTARXINT
10	UARTBRXINT
9	MFIOPOSINT (19) or MFIOPOSINT (18) or MFIOPOSINT (17) or MFIOPOSINT (16)
8	MFIOPOSINT (1) or MFIOPOSINT (0) or IOPOSINT (6) or IOPOSINT (5)
7	MFIONEGINT (19) or MFIONEGINT (18) or MFIONEGINT (17) or MFIONEGINT (16)
6	MFIONEGINT (1) or MFIONEGINT (0) or IONEGINT (6) or IONEGINT (5)
5	MBUSDMAFULLINT
4	SNDDMACNTINT
3	TELDMACNTINT
2	CHIDMACNTINT
1	IOPOSINT (0) or IONEGINT (0)

8.3.7 Clear Interrupt 1 Register

OFFSET = $100: write-only

The Clear Interrupt 1 Register bit locations are mapped on a one to one basis with the Interrupt Status 1 Register. A logic “1” written to a specific bit location will clear the corresponding bit in the Status Register.

8.3.8 Clear Interrupt 2 Register

OFFSET = $104: write-only

The Clear Interrupt 2 Register bit locations are mapped on a one to one basis with the Interrupt Status 2 Register. A logic “1” written to a specific bit location will clear the corresponding bit in the Status Register.

8.3.9 Clear Interrupt 3 Register

OFFSET = $108: write-only

The Clear Interrupt 3 Register bit locations are mapped on a one to one basis with the Interrupt Status 3 Register. A logic “1” written to a specific bit location will clear the corresponding bit in the Status Register.

8.3.10 Clear Interrupt 4 Register

OFFSET = $10C: write-only

The Clear Interrupt 4 Register bit locations are mapped on a one to one basis with the Interrupt Status 4 Register. A logic “1” written to a specific bit location will clear the corresponding bit in the Status Register.

8.3.11 Clear Interrupt 5 Register

OFFSET = $110: write-only

The Clear Interrupt 5 Register bit locations are mapped on a one to one basis with the Interrupt Status 5 Register. A logic “1” written to a specific bit location will clear the corresponding bit in the Status Register.

8.3.12 Enable Interrupt 1 Register

OFFSET = $118: read/write

The Enable Interrupt 1 Register bit locations are mapped on a one to one basis with the Interrupt Status 1 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.13 Enable Interrupt 2 Register

OFFSET = $11C: read/write

The Enable Interrupt 2 Register bit locations are mapped on a one to one basis with the Interrupt Status 2 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.14 Enable Interrupt 3 Register

OFFSET = $120: read/write

The Enable Interrupt 3 Register bit locations are mapped on a one to one basis with the Interrupt Status 3 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.15 Enable Interrupt 4 Register

OFFSET = $124: read/write

The Enable Interrupt 4 Register bit locations are mapped on a one to one basis with the Interrupt Status 4 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.16 Enable Interrupt 5 Register

OFFSET = $128: read/write

The Enable Interrupt 5 Register bit locations are mapped on a one to one basis with the Interrupt Status 5 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.17 Enable Interrupt 6 Register

OFFSET = $12C: read/write

The Enable Interrupt 6 Register bit locations are mapped on a one to one basis with the Interrupt Status 6 Register. A logic “1” written to a specific bit location will enable the corresponding interrupt in the Status Register.

Bit	Label	RESET	Read/Write
31-19	Reserved		
18	GLOBALEN	0	R/W
17	IRQPRITEST	0	R/W
16	IRQTEST	0	R/W
15-0	PRIORITYMASK[15:0]	X	R/W

GLOBALEN:

This is used as a global interrupt enable for all interrupts, and is cleared upon reset, therefore disabling all interrupts.

IRQPRITEST:

This bit is used for IC testing and should not be set.

IRQTEST:

This bit is used for IC testing and should not be set.

PRIORITYMASK[15:0]:

These bits are an enable mask for the 16 possible high priority interrupts, where bit 15 is the highest priority. A logic “1” written to a specific bit location will enable the corresponding high priority interrupt.

9. I/O Module

9.1 Overview

The I/O Module within the TMPR3911/12 contains support for reading and writing the 7 bi-directional general purpose I/O pins and the 32 bi-directional multi-function I/O pins.

Each of the general purpose I/O pins can be independently configured as an input or output port. Each port can generate a separate positive and negative edge interrupt and each port can also be independently configured to use a debouncer.

Of the 137 signal pins found on the TMPR3911/12, 32 of them are multi-function and can be independently programmed either as I/O ports or for an alternate standard/normal function. This allows the TMPR3911/12 to support a flexible and wide range of system applications and configurations. As an I/O port, any of these pins can be programmed as an input or output port, with the capability of generating a separate positive and negative edge interrupt.

9.1.1 Related Pins

I/O[6:0]: INPUT/OUTPUT

These pins are general purpose input/output ports. Each port can be independently programmed as an input or output port. Each port can generate a separate positive and negative edge interrupt. Each port can also be independently programmed to use a 16 ms to 24 ms debouncer.

MFIO[1:0]: INPUT/OUTPUT

These pins are multi-function input/output ports. Each port can be independently programmed as an input or output port, or can be programmed for multi-function use to support test signals (for debugging purposes only). Each port can generate a separate positive and negative edge interrupt. Note that 30 other multi-function pins are available for usage as multi-function input/output ports. These pins are named according to their respective standard/normal function and are not listed here.

9.2 Implementation

9.2.1 Block Diagram

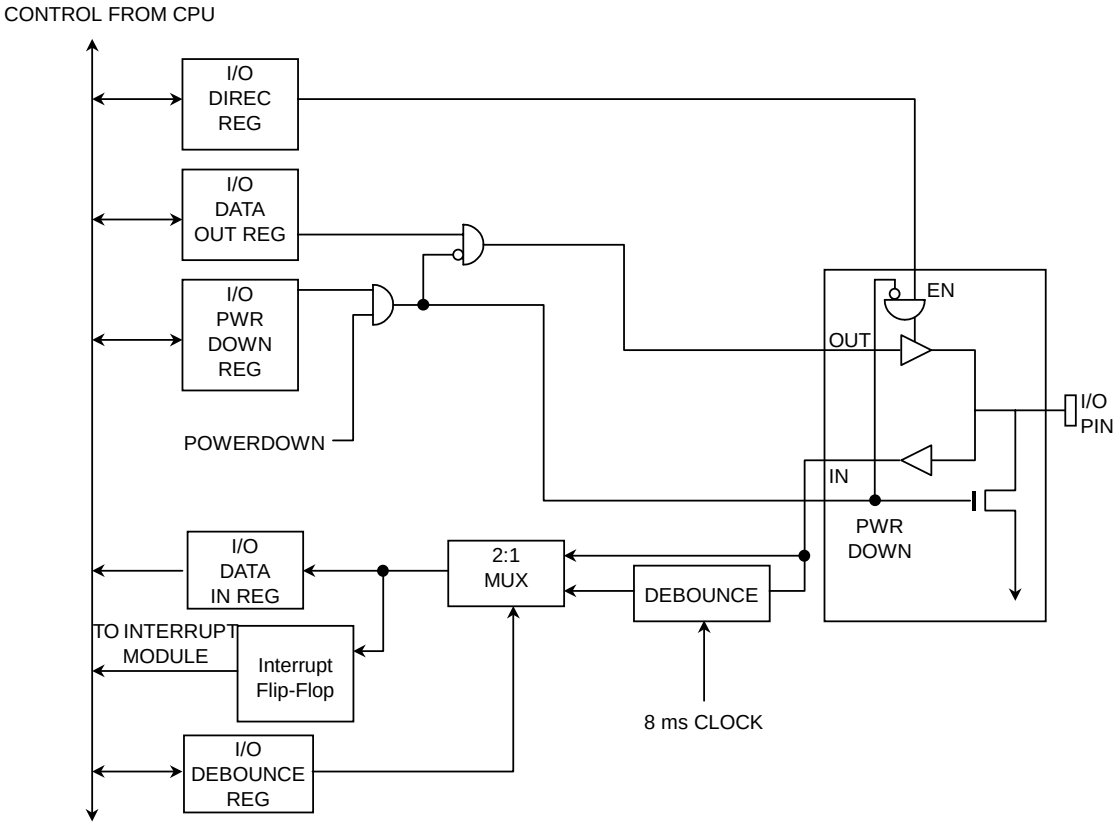


Figure 9.2.1 General Purpose I/O Port Block Diagram

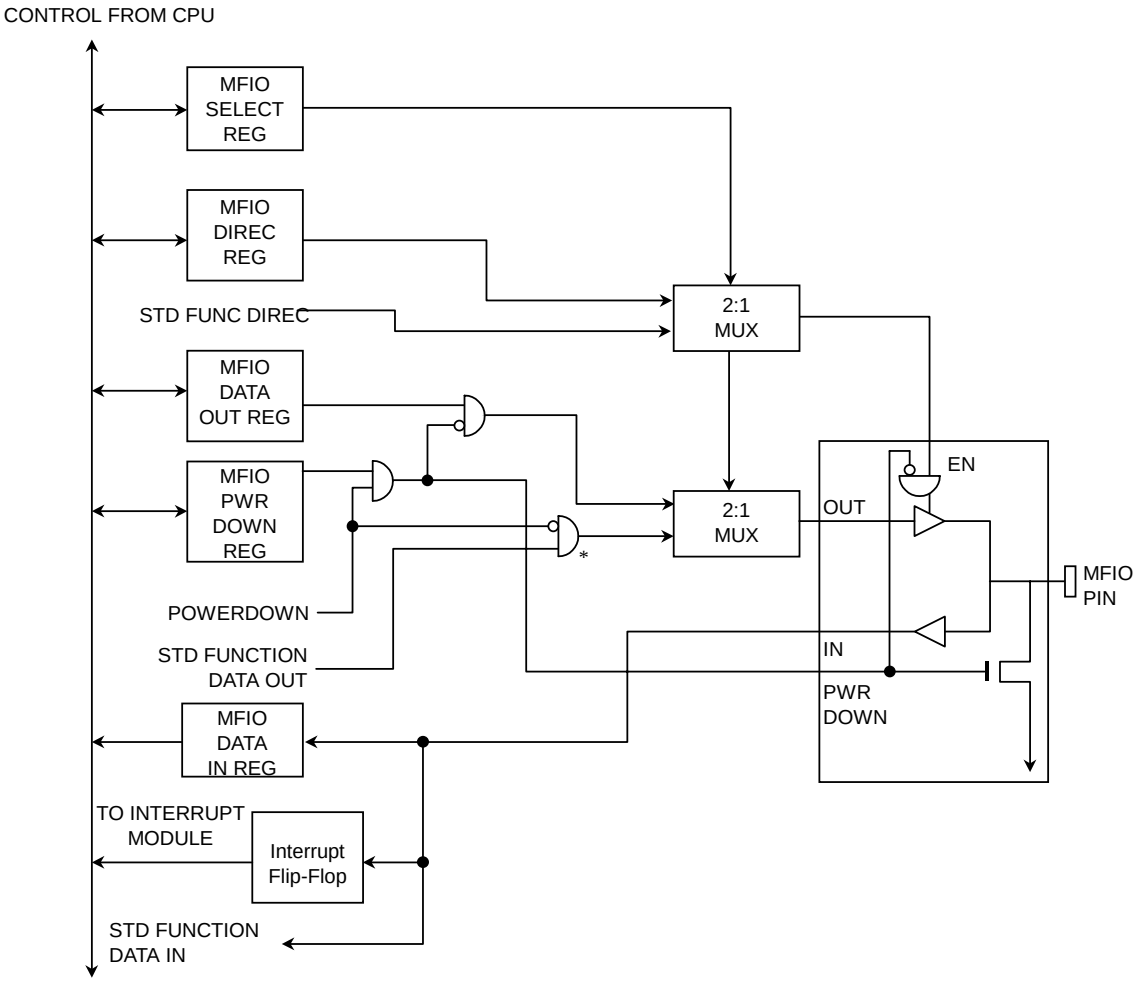


Figure 9.2.2 Multi-Function I/O Port Block Diagram

* CHIFS, CHICLK, CHIDOUT, BC32K, TXD, SPICLK, SPIOUT, SIBMCLK: each of above signals is directly connected with MUX without powerdown combination logic.

9.2.2 General Purpose I/O Ports

Each of the 7 general purpose I/O ports can be independently programmed as an input or output port. Each port can generate a separate positive and negative edge interrupt. Figure 9.2.1 shows a block diagram of a general purpose I/O port.

Each port consists of a bi-directional buffer connected to the appropriate TMPR3911/12 pin. For the input direction, the output signal from the input buffer is routed directly to a debounce circuit. This circuit performs a 16 ms to 24 ms debounce of the input signal. The debounce select control signal from the I/O Debounce Select Register is used to select between the debouncer output and the direct signal from the port input buffer, which bypasses the debouncer. This selected signal is then routed to the I/O Data Input Register and to the Interrupt Flip-Flop. Reading a specific bit location within the I/O Data Input Register returns the logic state of the respective general purpose I/O pin (either direct or debounced), regardless of whether that pin is configured as an output or input. If the pin is configured as an input, the value read is the logic state of the pin as driven by an external source. If the pin is configured as an output, the value read is the logic state of the pin as driven by the TMPR3911/12.

For the output direction, the input signal to the output buffer is routed from the appropriate bit within the I/O Data Output Register. The output enable control signal for the tri-state output buffer is routed from the I/O Direction Register.

The I/O Power-Down Register provides independent control bits for controlling the power-down state for each of the 7 general purpose I/O ports. If a particular I/O port is configured to power down, then the output will be gated low and a Pull-Down resistor will be enabled whenever the device powers down. Device power down occurs when either VCC3 or VCCON (internal signal) become de-asserted. The Pull-Down will prevent the input from floating if the general purpose I/O port is configured as an input.

9.2.3 Multi-function I/O Ports

Each of the 32 multi-function I/O ports can be independently programmed as an input or output port or can be programmed for an alternate standard/normal function. This allows the TMPR3911/12 to support a flexible and wide range of system applications and configurations. As an I/O port, any of these pins can generate a separate positive and negative edge interrupt. Figure 9.2.2 shows a block diagram of a multi-function I/O port.

Each port consists of a bi-directional buffer connected to the appropriate TMPR3911/12 pin. For the input direction, the output signal from the input buffer is routed directly to the MFI/O Data Input Register, to the Interrupt Flip-Flop, and to the appropriate module corresponding to the standard/normal function of the particular port, if this standard use is as an input signal. Reading a specific bit location within the MFIO Data Input Register returns the logic state of the respective multi-function I/O pin, regardless of whether that pin is configured as an output or input. If the pin is configured as an input, the value read is the logic state of the pin as driven by an external source. If the pin is configured as an output, the value read is the logic state of the pin as driven by the TMPR3911/12.

For the output direction, the input signal to the output buffer is routed from a multiplexer which selects between the appropriate bit within the MFIO Data Output Register and the signal driven by the appropriate module corresponding to the standard/normal function of the particular port, if this standard use is as an output signal. The output enable control signal for the tri-state output buffer is also routed from a multiplexer which selects between the appropriate bit within the MFIO Direction Register and the direction signal driven by the appropriate module corresponding to the standard use of the particular port, if this standard use is as an output signal.

The MFIO Power-Down Register provides independent control bits for controlling the power-down state for each of the 32 multi-function I/O ports. If a particular multi-function I/O port is configured to power down, then the output will be gated low and a Pull-Down resistor will be enabled whenever the device powers down. Device power down occurs when either VCC3 or VCCON (internal signal) become de-asserted. The Pull-Down will prevent the input from floating if the multi-function I/O port is configured as an input.

As mentioned previously, each of the multi-function I/O ports can be independently programmed for routing of signals for a standard/normal function to or from any given multi-function I/O port. Table 9.2.1 lists each of these standard functions for each multi-function I/O port. Note that depending on the reset state for the respective MFIO Select Register bits, the pin function is configured either as a multi-function port or as a standard function port.

Table 9.2.1 Multi-Function I/O Ports Versus Standard Functions

TMPR3911/12 pin	standard function (I = input, O = output)	multi-function I/O port	multi-function select (reset state)
CHIFS	CHIFS (I/O)	MIO[31]	MIOSEL[31] (1)
CHICLK	CHICLK (I/O)	MIO[30]	MIOSEL[30] (1)
CHIDOUT	CHIDOUT (O)	MIO[29]	MIOSEL[29] (1)
CHIDIN	CHIDIN (I)	MIO[28]	MIOSEL[28] (1)
DREQ*	DREQ* (I)	MIO[27]	MIOSEL[27] (0)
DGRNT*	DGRNT* (O)	MIO[26]	MIOSEL[26] (0)
BC32K	BC32K (O)	MIO[25]	MIOSEL[25] (1)
TXD	TXD (O)	MIO[24]	MIOSEL[24] (0)
RXD	RXD (I)	MIO[23]	MIOSEL[23] (0)
CS1*	CS1* (O)	MIO[22]	MIOSEL[22] (0)
CS2*	CS2* (O)	MIO[21]	MIOSEL[21] (0)
CS3*	CS3* (O)	MIO[20]	MIOSEL[20] (0)
MCS0*	MCS0* (O)	MIO[19]	MIOSEL[19] (1)
MCS1*	MCS1* (O)	MIO[18]	MIOSEL[18] (1)
MCS2*	MCS2* (O)	MIO[17]	MIOSEL[17] (1)
MCS3*	MCS3* (O)	MIO[16]	MIOSEL[16] (1)
SPICLK	SPICLK (O)	MIO[15]	MIOSEL[15] (0)
SPIOUT	SPIOUT (O)	MIO[14]	MIOSEL[14] (0)
SPIIN	SPIIN (I)	MIO[13]	MIOSEL[13] (0)
SIBMCLK	SIBMCLK (I/O)	MIO[12]	MIOSEL[12] (0)
CARDREG*	CARDREG* (O)	MIO[11]	MIOSEL[11] (1)
CARDIOWR*	CARDIOWR* (O)	MIO[10]	MIOSEL[10] (1)
CARDIORD*	CARDIORD* (O)	MIO[9]	MIOSEL[9] (1)
CARD1CSL*	CARD1CSL* (O)	MIO[8]	MIOSEL[8] (1)
CARD1CSH*	CARD1CSH* (O)	MIO[7]	MIOSEL[7] (1)
CARD2CSL*	CARD2CSL* (O)	MIO[6]	MIOSEL[6] (1)
CARD2CSH*	CARD2CSH* (O)	MIO[5]	MIOSEL[5] (1)
CARD1WAIT*	CARD1WAIT* (I)	MIO[4]	MIOSEL[4] (1)
CARD2WAIT*	CARD2WAIT* (I)	MIO[3]	MIOSEL[3] (1)
CARDDIR*	CARDDIR* (O)	MIO[2]	MIOSEL[2] (1)
MFIO[1]	(reserved)	MIO[1]	MIOSEL[1] (1)
MFIO[0]	(reserved)	MIO[0]	MIOSEL[0] (1)

Table 9.2.2 and Table 9.2.3 show the state of Power Down Mode for Multi-Function I/O Ports and I/O Ports, respectively.

Table 9.2.2 Multi-Function I/O Ports Versus Standard Functions (Power Down State)

TMPR3911/12 pin	standard function (I = input, O = output)	standard function direction (1 = output, 0 = input)	Power Down State			
			MIOPD=0 MIOSEL=0 MIODIREC=X	MIOPD=0 MIOSEL=1 MIODIREC=0	MIOPD=0 MIOSEL=1 MIODIREC=1	MIOPD=1 MIOSEL=X MIODIREC=X
CHIFS	CHIFS (I/O)	CHIFSDIR	CHIFS (I/O)	Hi-Z	MIODOUT[31]	PULL-DOWN
CHICLK	CHICLK (I/O)	CHICLKDIR	CHICLK (I/O)	Hi-Z	MIODOUT[30]	PULL-DOWN
CHIDOUT	CHIDOUT (O)	CHIENOUT	CHIDOUT (O)	Hi-Z	MIODOUT[29]	PULL-DOWN
CHIDIN	CHIDIN (I)	0	Hi-Z	Hi-Z	MIODOUT[28]	PULL-DOWN
DREQ*	DREQ* (I)	0	Hi-Z	Hi-Z	MIODOUT[27]	PULL-DOWN
DGRNT*	DGRNT* (O)	1	0	Hi-Z	MIODOUT[26]	PULL-DOWN
BC32K	BC32K (O)	1	BC32K (O)	Hi-Z	MIODOUT[25]	PULL-DOWN
TXD	TXD (O)	1	TXD (O)	Hi-Z	MIODOUT[24]	PULL-DOWN
RXD	RXD (I)	0	Hi-Z	Hi-Z	MIODOUT[23]	PULL-DOWN
CS1*	CS1* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[22]	PULL-DOWN
CS2*	CS2* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[21]	PULL-DOWN
CS3*	CS3* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[20]	PULL-DOWN
MCS0*	MCS0* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[19]	PULL-DOWN
MCS1*	MCS1* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[18]	PULL-DOWN
MCS2*	MCS2* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[17]	PULL-DOWN
MCS3*	MCS3* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[16]	PULL-DOWN
SPICLK	SPICLK (O)	1	SPICLK (O)	Hi-Z	MIODOUT[15]	PULL-DOWN
SPIOUT	SPIOUT (O)	1	SPIOUT (O)	Hi-Z	MIODOUT[14]	PULL-DOWN
SPIIN	SPIIN (I)	0	Hi-Z	Hi-Z	MIODOUT[13]	PULL-DOWN
SIBMCLK	SIBMCLK (I/O)	SIBMCLKDIR	SIBMCLK (I/O)	Hi-Z	MIODOUT[12]	PULL-DOWN
CARDREG*	CARDREG* (O)	1	0	Hi-Z	MIODOUT[11]	PULL-DOWN
CARDIOWR*	CARDIOWR* (O)	1	0	Hi-Z	MIODOUT[10]	PULL-DOWN
CARDIORD*	CARDIORD* (O)	1	0	Hi-Z	MIODOUT[9]	PULL-DOWN
CARD1CSL*	CARD1CSL* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[8]	PULL-DOWN
CARD1CSH*	CARD1CSH* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[7]	PULL-DOWN
CARD2CSL*	CARD2CSL* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[6]	PULL-DOWN
CARD2CSH*	CARD2CSH* (O)	/POWERDOWN	Hi-Z	Hi-Z	MIODOUT[5]	PULL-DOWN
CARD1WAIT*	CARD1WAIT* (I)	0	Hi-Z	Hi-Z	MIODOUT[4]	PULL-DOWN
CARD2WAIT*	CARD2WAIT* (I)	0	Hi-Z	Hi-Z	MIODOUT[3]	PULL-DOWN
CARDDIR*	CARDDIR* (O)	1	0	Hi-Z	MIODOUT[2]	PULL-DOWN
MFIO[1]	(reserved)	—	—	Hi-Z	MIODOUT[1]	PULL-DOWN
MFIO[0]	(reserved)	—	—	Hi-Z	MIODOUT[0]	PULL-DOWN

NOTE: CHIFSDIR, CHICLKDIR, SIBMCLKDIR, POWERDOWN: Internal Signal

Table 9.2.3 General Purpose IO (Power Down State)

TMPR3911/12 pin	IOPD = 0 IODIREC = 0	IOPD = 0 IODIREC = 1	IOPD = 1 IODIREC = X
IO[6]	Hi-Z	IODOUT[6]	PULL-DOWN
IO[5]	Hi-Z	IODOUT[5]	PULL-DOWN
IO[4]	Hi-Z	IODOUT[4]	PULL-DOWN
IO[3]	Hi-Z	IODOUT[3]	PULL-DOWN
IO[2]	Hi-Z	IODOUT[2]	PULL-DOWN
IO[1]	Hi-Z	IODOUT[1]	PULL-DOWN
IO[0]	Hi-Z	IODOUT[0]	PULL-DOWN

9.2.4 Related Interrupts

MFIOPOSINT[31:0]:

Issues an interrupt whenever any of the multi-function I/O pin(s) transition from a logic “0” to a logic “1”. There are a total of 32 multi-function I/O pins (31 thru 0), each of which corresponds to the respective bit within the Interrupt Status 3 Register. Each multi-function I/O pin can independently trigger an interrupt.

MFIONEGINT[31:0]:

Issues an interrupt whenever any of the multi-function I/O pin(s) transition from a logic “1” to a logic “0”. There are a total of 32 multi-function I/O pins (31 thru 0), each of which corresponds to the respective bit within the Interrupt Status 4 Register. Each multi-function I/O pin can independently trigger an interrupt.

IOPOSINT[6:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic “0” to a logic “1”. There are a total of 7 general purpose I/O pins (6 thru 0), each of which corresponds to the respective bit within the Interrupt Status 5 Register. Each general purpose I/O pin can independently trigger an interrupt.

IONEGINT[6:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic “1” to a logic “0”. There are a total of 7 general purpose I/O pins (6 thru 0), each of which corresponds to the respective bit within the Interrupt Status 5 Register. Each general purpose I/O pin can independently trigger an interrupt.

9.3 I/O Registers

9.3.1 I/O Control Register

OFFSET = $180:

Bit	Label	RESET	Read/Write
31	Reserved		
31-24	IODEBSEL[6:0]	X	R/W
23	Reserved		
22-16	IODIREC[6:0]	0	R/W
15	Reserved		
14-8	IODOUT[6:0]	X	R/W
7	Reserved		
6-0	IODIN[6:0]	–	R

IODEBSEL[6:0]:

These bits select the debounce mode for the 7 general purpose I/O pins. Setting a specific bit location to a logic “1” causes the respective general purpose I/O port signal to be filtered by a debounce circuit for the input signal direction. Clearing a specific bit location to a logic “0” causes the respective general purpose I/O port signal to bypass the input debounce circuit.

IODIREC[6:0]:

These bits control the direction of the 7 general purpose I/O pins. Setting a specific bit location to a logic “1” configures the respective general purpose I/O pin to be an output. Clearing a specific bit location to a logic “0” configures the respective general purpose I/O pin to be an input.

IODOUT[6:0]:

These bits correspond to the data output values for the 7 general purpose I/O pins. Setting or clearing a specific bit location controls the logic state of the respective general purpose I/O pin, if that pin is configured as an output.

IODIN[6:0]: read-only

These bits correspond to the data input values for the 7 general purpose I/O pins. Reading a specific bit location returns the logic state of the respective general purpose I/O pin, regardless of whether that pin is configured as an output or input.

9.3.2 MFIO Data Output Register

OFFSET = $184:

Bit	Label	RESET	Read/Write
31-0	MFIODOUT[31:0]	X	R/W

MFIODOUT[31:0]:

These bits correspond to the data output values for the 32 multi-function I/O ports. Setting or clearing a specific bit location controls the logic state of the respective multi-function I/O port, if that pin is configured as an output.

9.3.3 MFIO Direction Register

OFFSET = $188:

Bit	Label	RESET	Read/Write
31-0	MFIODIREC[31:0]	0	R/W

MFIODIREC[31:0]:

These bits control the direction of the 32 multi-function I/O ports. Setting a specific bit location to a logic “1” configures the respective multi-function I/O port to be an output. Clearing a specific bit location to a logic “0” configures the respective multi-function I/O port to be an input.

9.3.4 MFIO Data Input Register

OFFSET = $18C: read-only

Bit	Label	RESET	Read/Write
31-0	MFIODIN[31:0]	–	R

MFIODIN[31:0]: read-only

These bits correspond to the data input values for the 32 multi-function I/O pins. Reading a specific bit location returns the logic state of the respective multi-function I/O pin, regardless of whether that pin is configured as an output or input.

9.3.5 MFIO Select Register

OFFSET = $190:

Bit	Label	RESET	Read/Write
31-0	MFIOSEL[31:0]	$F20F0FFF	R/W

MFIOSEL[31:0]:

These bits correspond to the select mode for the 32 multi-function I/O pins. Setting a specific bit location to a logic “1” configures the respective multi-function I/O pin to be selected as a bi-directional I/O port. Clearing a specific bit location to a logic “0” configures the respective multi-function I/O pin to be selected as a standard/normal pin function.

9.3.6 I/O Power-Down Register

OFFSET = $194:

Bit	Label	RESET	Read/Write
31-7	Reserved		
6-0	IOPD[6:0]	$7F	R/W

IOPD[6:0]:

These bits control the power-down state for the 7 general purpose I/O pins. Setting a specific bit location to a logic “1” configures the respective general purpose I/O pin to power down when the TMPR3911/12 powers down. Clearing a specific bit location to a logic “0” configures the respective general purpose I/O pin to be in an active state at all times.

9.3.7 MFIO Power-Down Register

OFFSET = $198:

Bit	Label	RESET	Read/Write
31-0	MFIOPD[31:0]	$FAF03FFC	R/W

MFIOPD[31:0]:

These bits control the power-down state for the 32 multi-function I/O ports. Setting a specific bit location to a logic “1” configures the respective multi-function I/O port to power down when the TMPR3911/12 powers down. Clearing a specific bit location to a logic “0” configures the respective multi-function I/O port to be in an active state at all times.

10. IR Module

10.1 Overview

The IR Module consists of three parts. The first is the Consumer Interface, which provides an easy mechanism for generating pulses to either turn on or turn off a light emitting diode (LED). The pattern in which the diode is turned on and off is used to encode commands which emulate remote control key presses that the consumer device recognizes. The software is responsible for programming the IR Module parameters to emulate desired consumer remote control protocols and key presses. The Consumer Interface provides universal control of common consumer devices such as televisions, VCR's, hi-fi equipment, appliances, etc.

The second part of the IR Module is the Communication Interface. The IR Module can be used in one of two data communication modes: using a frequency shift keyed (FSK) modulation scheme or using a scheme based on the protocol standardized by the InfraRed Data Association (IrDA).

The Communication Interface is implemented using a standard UART protocol with an adjustable baud rate. This interface operates half-duplex in one direction at a time. The UARTB Module is used for both the FSK and IrDA data communication modes (see Section 16 for a detailed description of the UART Module).

For the FSK mode transmit direction, external communication IR analog circuitry encodes the UART output data using an FSK modulation scheme. For the receive direction, external communication IR analog circuitry amplifies the received photo-diode signal and demodulates the FSK signal before feeding the UART input.

For the IrDA mode transmit direction, the UART output data directly drives the external analog LED circuit. For the receive direction, the received photo-diode signal is externally amplified before feeding the UART input. Additional control bits in the UART Module also allow for various narrow pulse options to support the IrDA 1.0 standard.

The final part of the IR Module is the Carrier Detect State Machine Interface. This interface is used to periodically enable the receiver in the external communication IR analog circuitry to detect whether there is a valid carrier present. If a valid carrier is present then an interrupt will be issued so the system can begin receiving data.

10.1.1 Related Pins

IROUT: OUTPUT

This pin is the UART transmit signal from the UARTB Module or the Consumer IR output signal if Consumer IR mode is enabled.

IRIN: INPUT

This pin is the UART receive signal to the UARTB Module.

RXPWR: OUTPUT

This pin is the receiver power output control signal to the external communication IR analog circuitry.

CARDET: INPUT

This pin is the carrier detect input signal from the external communication IR analog circuitry.

10.2 Consumer IR

10.2.1 Requirements

Consumer IR remote controllers send a burst of pulses whenever a button is pressed. The number of bursts along with the length of each burst defines the key that has been pressed.

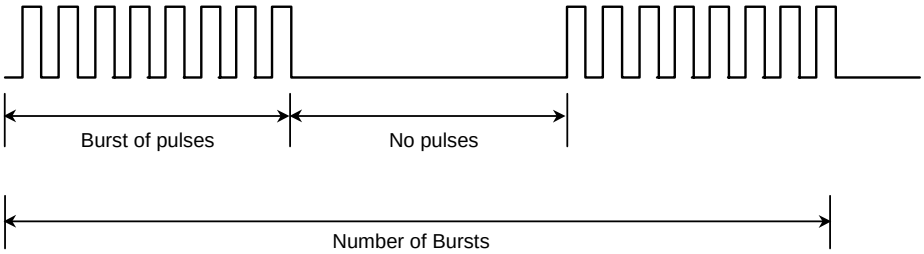


Figure 10.2.1 Consumer IR Pulses

The frequency of the pulses and the pulse width is fixed for a given remote control but can vary between different remote controls from different manufacturers. The number of pulses in a burst, the amount of time when there are no pulses, and the number of times that the burst is repeated are used to distinguish which key has been pressed for a given remote control.

10.2.2 Implementation

The Consumer IR Interface is designed to assist the CPU in shaping the pulses that are required to emulate a key press on a remote control. The logic has been implemented such that the frequency (typical consumer remote controllers use a carrier frequency in the 30 to 50 kHz range) and width of a pulse can be programmed by the CPU for a given remote control. The processor can then vary the number of bursts in a pulse and the amount of time when there are no pulses on a byte by byte basis in order to emulate the various key presses.

10.2.3 Block Diagram

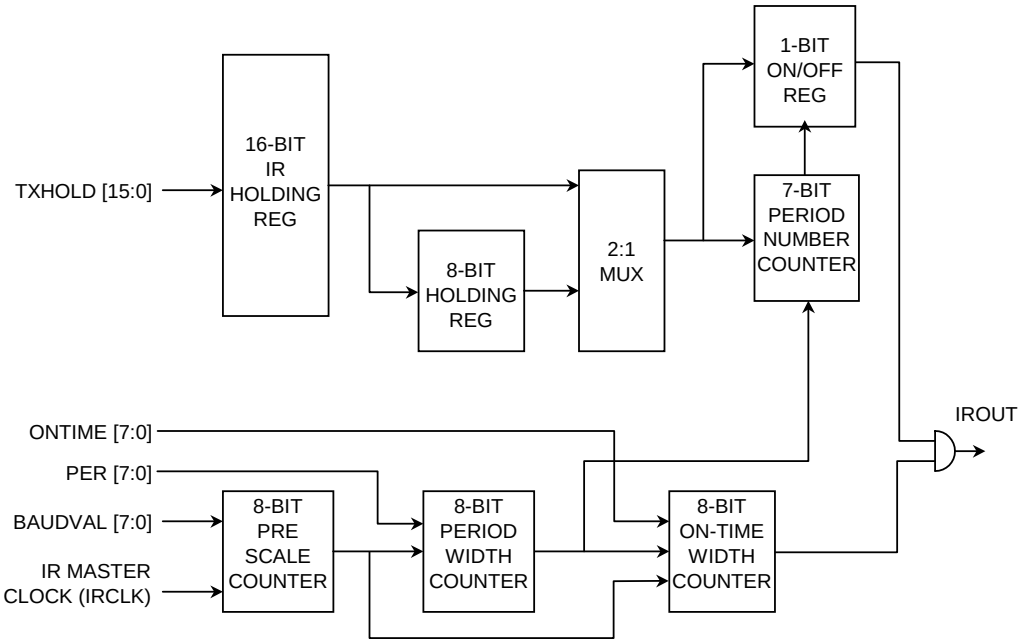


Figure 10.2.2 Consumer IR Block Diagram

10.2.4 Functional Description

The Consumer IR logic remains idle until the CPU activates the IR logic by writing to the IR Control 1 Register to enable the Consumer mode (ENCONSM = “1”). Prior to enabling the Consumer mode, the CPU must configure the Baud Value (BAUDVAL), Period Value (PER), and OnTime Value (ONTIME) fields located in the IR Control 2 Register. BAUDVAL[7:0] is used to pre-scale IR master clock (IRCLK) (since typical consumer IR data rates are in the kHz range instead of the MHz range). PER[7:0] is used to define the period of the pulses for a burst, representing the number of pre-scaled counts per period. ONTIME[7:0] defines the amount of time that the pulse is on for the given period, representing the number of pre-scaled counts for which the pulse is on. ONTIME should always be less than PER or the IROUT signal will never transition low.

Transmit data = number of pulse and no-pulse width are written by the CPU into a 16-bit holding register, IR Holding Register, TXHOLD field. When the 7-bit Period Number Counter (down-counter) reaches zero, it will load the lower 7 bits of the holding register’s upper byte into the Period Number Counter and the upper bit into the 1-bit ON/OFF Register.

If the upper bit is set to a “1” then the IROUT signal will be a “1” until the OnTime Width Counter reaches it’s terminal count. The OnTime Width Counter freezes once it reaches it’s terminal count in order to prevent the signal from transitioning until the counter is loaded again. The IROUT signal will continue to transition according to the Period Value and OnTime Value until the 7-bit Period Number Counter reaches zero. At this time the lower 7 bits of the holding register’s lower byte is loaded into the Period Number Counter and the upper bit of the lower byte is loaded into the 1-bit ON/OFF Register. If the upper bit is set to a “0” this time, then the IROUT signal will remain low for as many period counts as are defined by the value loaded into the 7-bit Period Number Counter. Thus the CPU controls the pulse generation by defining a number of pulses to burst (upper bit set to “1”, lower 7-bit data defines the number of pulses) or an amount of time when there are no pulses (upper bit set to “0”, lower 7-bit data defines the number of pulse times when IROUT is “0”) on a byte by byte basis. The OnTime Value and Period Value timing relationships are shown in Figure 10.2.3.

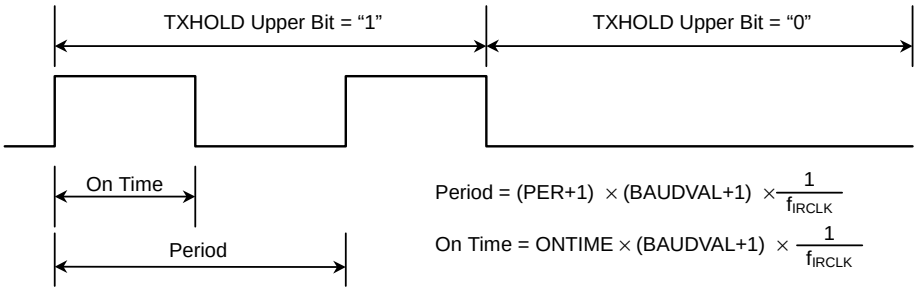


Figure 10.2.3 Consumer IR Signal Generation

10.2.5 Related Interrupts

IRCONSMINT:

Whenever the upper byte of data is loaded into the 7-bit Period Number Counter, the lower byte of data is loaded into an intermediate holding register. At this time the 16-bit IR Holding Register is empty. The IRCONSMINT interrupt is then set to inform the CPU that the IR Holding Register is available. The CPU must fill the next word of data into the IR Holding Register before the 7-bit Period Number Counter is finished counting the periods for both the upper and lower bytes of data.

10.3 Two-Way Communication Via IR

10.3.1 Requirements

The Communication Interface is implemented using a standard UART protocol, consisting of a start bit, 8 bits of data (LSB first), then a stop bit. The data rate is adjustable using a programmable baud rate counter. The Communication Interface consists of both a transmitter and a receiver and the UART is operated half-duplex in one direction at a time. The UARTB Module is used for both the FSK and IrDA data communication modes and contains DMA support for transmit and receive data (see Section 16 for a detailed description of the UART Module).

For the FSK mode transmit direction, external communication IR analog circuitry encodes the UARTB output data using an FSK modulation scheme, which modulates at two different frequencies depending on whether the data is a “1” or a “0”. The carrier frequency is 1375 kHz and the mark and space frequencies are 1325 kHz and 1425 kHz, respectively. For the receive direction, external communication IR analog circuitry amplifies the received photo-diode signal and demodulates and decodes the FSK signal before feeding the UARTB input. The transmitter and receiver portions of the external communication IR analog circuitry contain individual power-down control for power management purposes. The FSK communication mode operates at a data rate of 2400 to 38400 bps at 3 meters.

For the IrDA mode transmit direction, the UARTB output data directly drives the external analog LED circuit. For the receive direction, the received photo-diode signal is externally amplified before feeding the UARTB input. Additional control bits in the UART Module also allow for various narrow pulse options to support the IrDA 1.0 standard. When configured for these options, the UART transmit output pulses are narrower than normal and the UART receiver circuit also expects the same narrow pulse format for the input data. The IrDA communication mode operates at a data rate up to 115.2 kbps at 1 meter.

10.3.2 Block Diagram

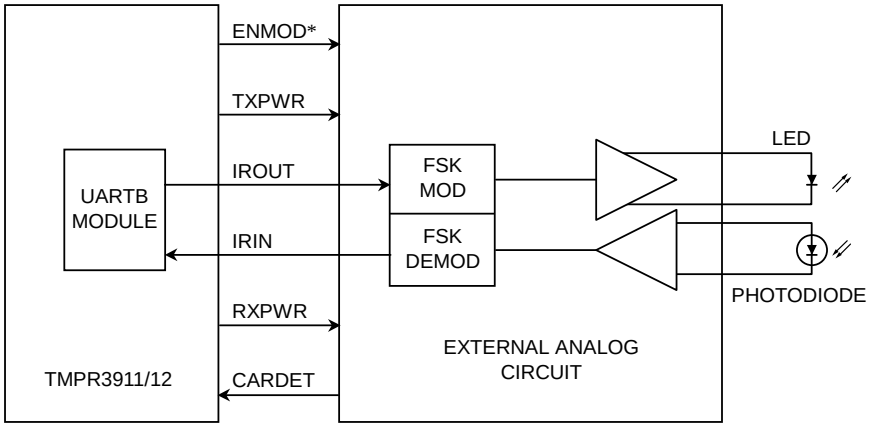


Figure 10.3.1 FSK Communication Circuit

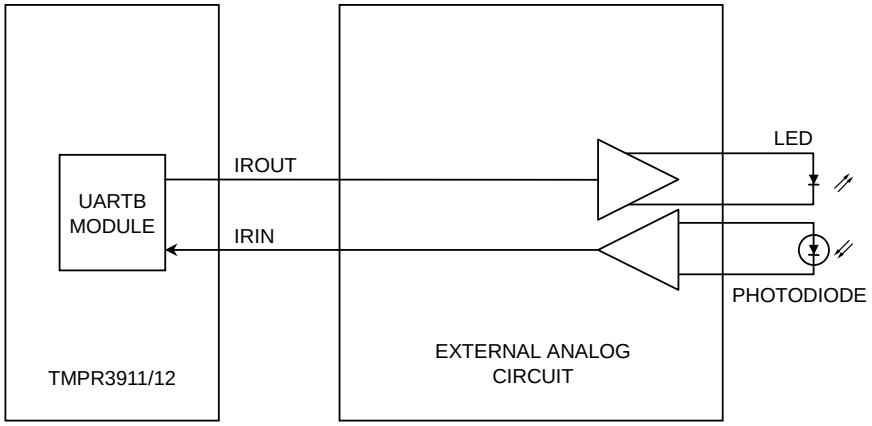


Figure 10.3.2 IrDA Communication Circuit

10.4 Carrier Detect State Machine

10.4.1 Requirements

The external communication IR analog circuitry contains a circuit which detects if there is a valid carrier present. It is not desirable to leave this circuit enabled all the time because it dissipates too much power. Thus the circuit is powered on and off with the RXPWR pin. It is possible for the CPU to periodically enable the RXPWR pin, then wait to see if a carrier is present (CARDET pin = "1"), but this requires a lot of processor overhead. In order to alleviate the overhead, the IR Module contains a circuit that periodically enables the RXPWR pin, waits a specified period of time, samples the CARDET pin, then disables the RXPWR pin. If the CARDET pin is a "1" when sampled then an interrupt is issued to the CPU indicating that there is a valid carrier present. After enabling the RXPWR pin the CARDET pin cannot be sampled for a specified amount of time because the Carrier Detect Logic in the external communication IR analog circuitry takes time to settle. The time between successive enabling of the RXPWR pin is programmable depending on the desired sample rate.

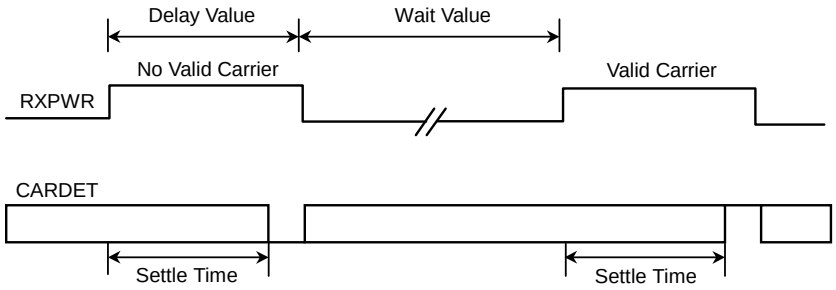


Figure 10.4.1 Carrier Detect State Timing

10.4.2 Block Diagram

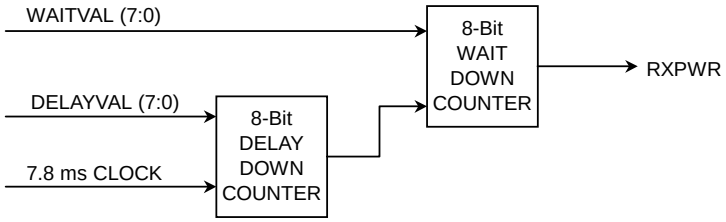


Figure 10.4.2 Carrier Detect Block Diagram

10.4.3 Functional Description

DELAYVAL[7:0] is loaded into the 8-bit Delay Down Counter whenever the counter reaches a count of zero. The terminal count of the Delay Down Counter enables the 8-bit Wait Down Counter which loads WAITVAL[7:0] whenever the Wait Down Counter reaches a count of zero. The RXPWR pin is asserted whenever the Wait Down Counter reaches a count of zero. Thus the period at which RXPWR is periodically asserted is derived by:

(DELAYVAL + 1) * (WAITVAL + 1) *7.8 ms

The RXPWR pin will remain asserted until the Wait Down Counter counts again which will occur at:

(DELAYVAL + 1) *7.8 ms

At the same time the RXPWR pin is deasserted the CARDET pin is sampled and an interrupt is set if a valid carrier is present.

10.4.4 Related Interrupts

CARSTINT:

This interrupt is set whenever the Carrier Detect State Machine samples the CARDET pin- = “1” just before turning off the RXPWR pin.

POSCARINT:

This interrupt is set whenever CARDET pin transitions from a logic “0” to a logic “1”.

NEGCARINT:

This interrupt is set whenever CARDET pin transitions from a logic “1” to a logic “0”.

10.5 IR Registers

10.5.1 IR Control 1 Register

OFFSET = $0A0:

Bit	Label	RESET	Read/Write
31-25	Reserved		
24	CARDET	–	R
23-16	BAUDVAL[7:0]	X	R/W
15-5	Reserved		
4	TESTIR	0	R/W
3	DTINVERT	0	R/W
2	RXPWR	0	R/W
1	ENSTATE	0	R/W
0	ENCONSM	0	R/W

CARDET: read-only

This bit provides the status of the CARDET (carrier detect) input pin.

BAUDVAL[7:0]:

These bits are used to pre-scale the IR master clock (IRCLK). The bits are loaded into an 8-bit down-counter that counts at the IRCLK rate. The resulting baud clock is used to clock the Period Width Counter and OnTime Width Counter.

TESTIR:

This bit is used to speed up the counters for IC testing and should not be set.

DTINVERT:

Setting this bit to a logic “1” will cause the IROUT signal to be active low instead of active high.

RXPWR:

This bit is connected to the RXPWR pin.

ENSTATE:

Setting this bit to a logic “1” enables the Communication IR carrier detect state machine.

ENCONSM:

Setting this bit to a logic “1” enables the Consumer IR function.

10.5.2 IR Control 2 Register

OFFSET = $0A4: write-only

Bit	Label	RESET	Read/Write
31-24	PER[7:0]	X	W
23-16	ONTIME[7:0]	X	W
15-8	DELAYVAL[7:0]	X	W
7-0	WAITVAL[7:0]	X	W

PER[7:0]: write-only

These bits are used to define the period of the Consumer IR pulses. The bits are loaded into an 8-bit down-counter that counts at the rate defined by the Baud Value. The resulting period is given by the following equation. (f_{IRCLK} is the frequency of the IR master clock (IRCLK).)

$$Period = (PER + 1) * (BAUDVAL + 1) * (1/f_{IRCLK})$$

ONTIME[7:0]: write-only

These bits are used to define the On Time of the Consumer IR pulses. These bits are loaded into an 8-bit down-counter that counts at the rate defined by the Baud Value. The resulting On Time is given by the following equation.

$$On\ Time = ONTIME * (BAUDVAL + 1) * (1/f_{IRCLK})$$

DELAYVAL[7:0]: write-only

These bits are used to define the amount of time that the RXPWR pin will be a logic “1” when the Carrier State Machine logic is enabled. The bits are loaded into an 8-bit down-counter that counts at a 7.8 ms rate. Thus the time is programmable from 7.8 ms to 2.0 seconds.

$$Delay\ Time = (DELAYVAL + 1) * 7.8\ ms$$

WAITVAL[7:0]: write-only

These bits are used to define the amount of time to wait before asserting the RXPWR pin to a logic “1” when the Carrier State Machine logic is enabled. The bits are loaded into an 8-bit down-counter that counts whenever the DELAYVAL counter reaches a count of zero. Thus the time is programmable from 7.8 ms to 511 seconds.

$$Wait\ Time = (DELAYVAL + 1) * (WAITVAL + 1) * 7.8\ ms$$

10.5.3 IR Holding Register

OFFSET = $0A8: write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	TXHOLD[15:0]	X	W

TXHOLD[15:0]: write-only

These bits are used to define the number of pulses in a burst or the amount of time to keep the IROUT pin low. The lower seven bits of the upper byte and the lower seven bits of the lower byte are used as values to load into a 7-bit counter that counts the number of periods as defined by the Period Value. The upper bit in each byte if set to a logic “1” causes pulses on the IROUT pin and if set to a logic “0” causes the IROUT pin to remain low for a given count.

Before setting the ENCONSM bit the TXHOLD register must be pre-loaded with the first valid word of data.

11. Magicbus Module

11.1 Overview

Magic Port is a multifunction expansion port for a wide range of serial peripheral connections. Magicbus is a serial interface standard which includes an advanced communications protocol for interfacing to low-cost and low-power daisy-chainable peripherals and docks. The Magicbus implementation is intended to support peripherals such as scanners, modems, battery chargers, base stations (including intelligent answering machines, telephones, fast chargers, etc.), storage devices, etc. Magicbus features high-speed data transfer rates (up to 14.75 Mbps), daisy-chainable peripherals (up to 6 maximum), and hot plugging. A low-cost Magicbus Interface Circuit (MBIC) is also available for use in peripherals. This interface can be implemented as a discrete circuit, as a standalone ASIC, or integrated with a microcontroller core (e.g., 68HC05 type).

Magicbus consists of a synchronous, serial two-wire (clock and data), half-duplex communications protocol. The clock and data signals are bi-directional, such that the TMPR3911/12 Magicbus Module can be dynamically configured as the bus master or slave at a given time. The protocol also includes an interrupt source for initiating all peripheral servicing, as well as when peripherals have been attached to or detached from the bus. TMPR3911/12's Magicbus Module contains independent transmitter and receiver logic to implement the half-duplex serial data transfer, with DMA support provided for both the transmit and receive directions. The independent transmitter and receiver allow a peripheral to control the data rate of the receiving shift register, since the receiver uses the peripheral's transmit clock as it's clock source.

11.1.1 Related Pins

MBUSCLK: INPUT/OUTPUT

This pin is the bi-directional Magicbus clock signal. MBUSCLK is an input signal whenever the TMPR3911/12 is in the slave mode and is an output signal whenever the TMPR3911/12 is in the master mode.

MBUSDATA: INPUT/OUTPUT

This pin is the bi-directional Magicbus data signal. MBUSDATA is an input signal whenever the TMPR3911/12 is in the slave mode and is an output signal whenever the TMPR3911/12 is in the master mode.

MBUSINT: INPUT

This pin is the Magicbus interrupt signal. This signal is used to interrupt the TMPR3911/12 whenever a peripheral has been attached to or detached from the bus, or whenever a peripheral has initiated an interrupt event.

11.2 Interface Requirements

11.2.1 Timing Requirements

Magicbus consists of a synchronous, serial two-wire (clock and data), half-duplex communications protocol. The clock and data signals are bi-directional, such that the bus supports a single Master at a given time, with all other bus members configured as Slaves. The dynamic configuration of Master versus Slave assignments is controlled by the Magicbus protocol. Data is pushed onto the bus using the positive edge of MBUSCLK and the Slaves sample the data using the negative edge of MBUSCLK, as shown in Figure 11.2.1.

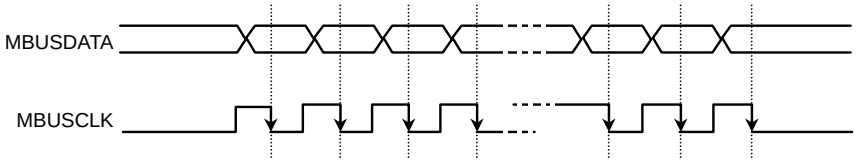


Figure 11.2.1 MBUSDATA and MBUSCLK Timing

11.2.2 Data Formats

Magicbus commands and data sequences are comprised of groups of either 16-bit or 32-bit data, which are shifted serially with the most significant bit (MSB) transmitted first in time. Figure 11.2.2 shows the timing diagram for a 16-bit group of data. Each of these 16-bit or 32-bit groups is shifted serially in either a seamless, non-interrupted fashion, or with delay inserted between every 8 bits, 16 bits, or 32 bits of the data stream. Magicbus Command Words consist of 16-bit packets containing fields which specify information such as peripheral address, command code, and parity. All peripherals monitor and read these Command Words to determine whether they are the addressed peripheral and, if so, what is the exact command to be executed.

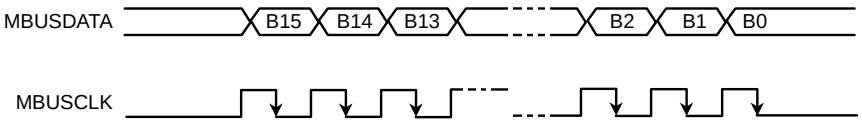


Figure 11.2.2 16-bit Data Group

11.2.3 Command Words

The Magicbus protocol dictates that when a Command Word is to be transmitted, the clock and data signal functions are swapped, whereby MBUSDATA contains the clock signal and MBUSCLK contains the data signal. This Swap Mode is used to differentiate Command Words from “raw data”. Each peripheral MBIC contains circuitry which detects whenever the clock and data signal functions have been swapped. The first 2 bits of each Command Word are Command Detection Bits which re-synchronize the bus to reset the mode of any swapped clock/data lines and also prevent any past bus errors from corrupting current and future data transmissions. Figure 11.2.3 shows a timing diagram of the Command Word and the swapping of the clock and data signal functions.

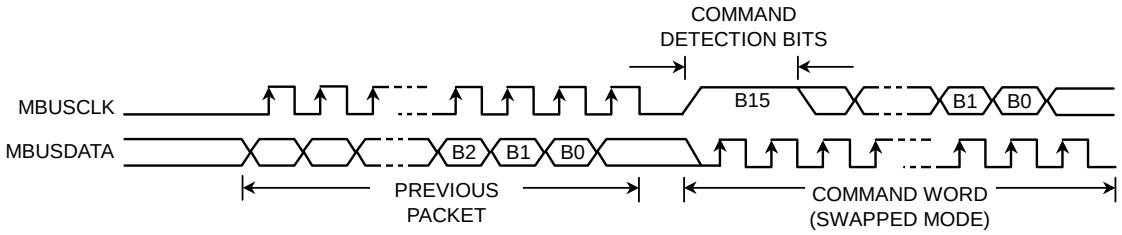


Figure 11.2.3 Magicbus Command Word

11.3 Implementation

11.3.1 Block Diagram

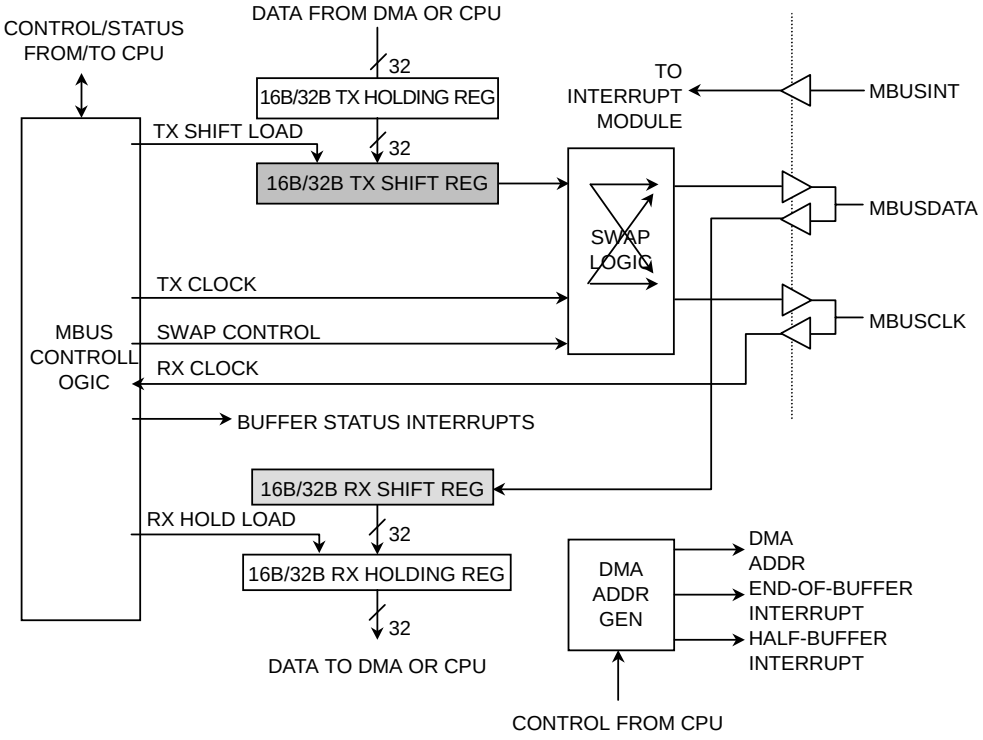


Figure 11.3.1 Magicbus Module Block Diagram

11.3.2 Transmitter

The Magicbus Transmitter consists of a Holding Register and Shift Register which support the serial data transmission from the Magicbus Module to attached Magicbus peripheral devices. The 32-bit Transmit Holding Register is loaded with parallel data from either the DMA circuit or directly from the CPU. The Magicbus Module provides an interrupt (MBUSTXBUFAVAILINT) whenever the ENMBUS control bit is enabled or when the Transmit Holding Register is available (i.e., empty). When the Magicbus Module is configured for long mode (LONG = “1”), all 32 bits of the Transmit Holding Register are valid. When the Magicbus Module is configured for word mode (LONG = “0”), only the lower 16 bits of the Transmit Holding Register are valid.

The Magicbus Transmit Holding Register can be accessed at one of 2 address locations. One location corresponds to Normal Mode and the other location corresponds to Swap Mode. As mentioned previously, the Swap Mode is used for transferring Command Words, for which the clock and data signals are swapped.

The contents of the Transmit Holding Register then feed the 32-bit Transmit Shift Register. This data is transferred in parallel whenever the Transmit Shift Register is empty and available. The Magicbus Module provides an interrupt (MBUSEMPTYINT) whenever the Transmit Holding Register and Transmit Shift Register are both available (i.e., empty).

The Magicbus Module also issues an interrupt (MBUSTXERRINT) if the Transmit Holding Register is written to when the Holding Register is not available (i.e., not empty). In this case, not enough time was provided for the contents of the Transmit Holding Register to be transferred to the Transmit Shift Register.

The contents of the Transmit Shift Register are then shifted out serially to the MBUSDATA output port (except for Command Words which use Swap Mode). When the Magicbus Module is configured for long mode (LONG = “1”), all 32 bits of the Transmit Shift Register are valid. When the Magicbus Module is configured for word mode (LONG = “0”), only the first 16 bits of the Transmit Shift Register are valid.

The Magicbus Module supports 4 different combinations of clock polarity and phase, as determined by the CLKPOL and PHAPOL control bits. Figure 11.3.2 shows a timing diagram of the clock and data relationships for these 4 combinations. This flexibility allows data to be pushed and sampled on either the rising or falling edges of the clock. However, the actual Magicbus protocol and specification specifies the use of only one of these combinations: CLKPOL = “0” and PHAPOL = “1”, which means that MBUSDATA is always pushed on the rising edge of MBUSCLK and always sampled on the falling edge of MBUSCLK.

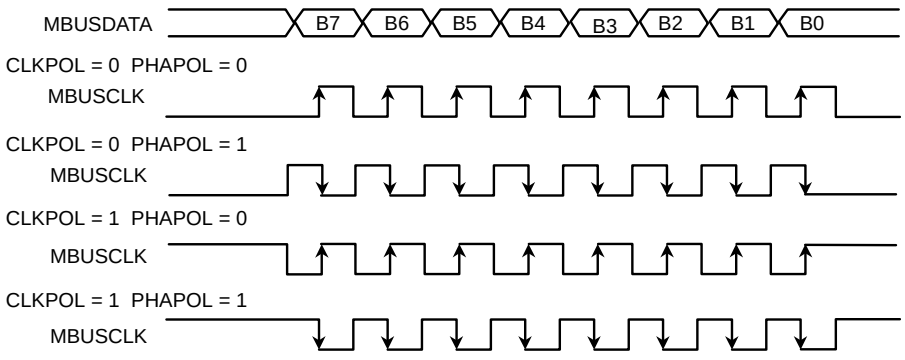


Figure 11.3.2 Magicbus Clock and Data Relationships

The transmit clock rate is also programmable, using the Magicbus Baud Rate Generator. The Magicbus specification supports data rates from 250 kbps to 14.75 Mbps. See Section 11.3.4 for a detailed description of the Magicbus Baud Rate Generator.

As mentioned previously, the transmit data consists of 16-bit or 32-bit groups. Seamless operation is possible, whereby each data bit for each group is transmitted in a continuous and non-interrupted fashion. This seamless operation results in the fastest possible data rate for Magicbus. In order to maintain seamless operation, the Transmit Holding Register must be reloaded before the contents of the Transmit Shift Register have been completely shifted out.

In order to accommodate slower peripherals which cannot keep up with fast Magicbus data rates, the Magicbus Module contains a programmable Inter-Packet Delay Counter. This counter allows the insertion of a programmable number of baud clocks of delay between bytes, words, or longs of the data stream, as shown in Figure 11.3.3. See Section 11.3.4 for a detailed description of the Inter-Packet Delay Counter.

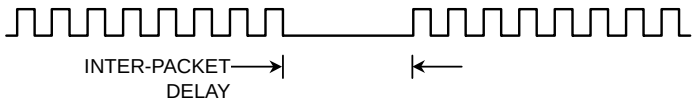


Figure 11.3.3 Magicbus Inter-Packet Delay

11.3.3 Receiver

The Magicbus Receiver consists of a Shift Register and Holding Register which support the receiving of serial data from attached Magicbus peripheral devices to the Magicbus Module. Data from the MBUSDATA input port is shifted in serially to the 32-bit Receive Shift Register. The receive clock from the MBUSCLK port is used to clock the serial data through the Receive Shift Register. This allows a peripheral to control the data rate of the Magicbus Receiver (allowing support for slower peripherals), since the receiver uses the peripheral's transmit clock as it's clock source. When the Magicbus Module is configured for long mode (LONG = "1"), all 32 bits of the Receive Shift Register are valid. When the Magicbus Module is configured for word mode (LONG = "0"), only the first 16 bits of the Receive Shift Register are valid.

The contents of the Receive Shift Register are then fed in parallel to the 32-bit Receive Holding Register. The data is transferred whenever the Receive Shift Register has finished shifting in either 16 or 32 bits of data depending on the selected mode.

The Receive Holding Register is read by either the DMA circuit or directly by the CPU. The Magicbus Module provides an interrupt (MBUSRXBUFAVAILINT) whenever the Receive Holding Register is loaded with data (i.e., full). When the Magicbus Module is configured for long mode (LONG = "1"), all 32 bits of the Receive Holding Register are valid. When the Magicbus Module is configured for word mode (LONG = "0"), only the lower 16 bits of the Receive Holding Register are valid.

The Magicbus Module issues an interrupt (MBUSRXERRINT) if the Receive Holding Register is loaded twice from the Receive Shift Register before the MBUSRXBUFAVAILINT interrupt is serviced and the data in the Receive Holding Register was never read. This situation might occur if the DMA circuit or the CPU cannot keep up with the incoming data rate of the Magicbus port.

11.3.4 Control Logic

The Magicbus Module contains several sets of counters which are used to generate the various internal control signals and clocks needed for the transmitter and receiver sections. A block diagram of the control logic is shown in Figure 11.3.4.

The ENMBUS control bit is used to enable the Magicbus Module by triggering the On/Off Logic circuit. When the ENMBUS control bit is deasserted to disable the Magicbus Module, the module will not shutdown until the Transmit Holding Register and Transmit Shift Register are empty plus a couple of Magicbus baud clocks of delay. This allows the Magicbus Module to cleanly complete any pending transactions before shutdown of the module. The MBUSON status bit indicates whether the Magicbus Module is still enabled or not.

The programmable Magicbus Baud Rate Generator provides variable control of the transmit clock rate, as determined by the BAUDRATE[7:0] control register value preloaded into the Baud Rate Generator. The output of the Baud Rate Generator is the Magicbus baud clock, which is used to operate the Transmit Shift Register and provides the MBUSCLK whenever TMPR3911/12 is configured as the Magicbus bus master. The Magicbus specification supports data rates from 250 kbps to 14.75 Mbps. As mentioned previously, the Magicbus Module supports 4 different combinations of clock polarity and phase, as determined by the CLKPOL and PHAPOL control bits. The Magicbus specification specifies the use of only one of these combinations: CLKPOL = “0” and PHAPOL = “1”, which means that MBUSDATA is always pushed on the rising edge of MBUSCLK and always sampled on the falling edge of MBUSCLK.

The Magicbus Transmitter control logic consists of a Transmit Bit Counter and a programmable Transmit Inter-Packet Delay Counter. The Transmit Bit Counter is used to count the number of bits to be shifted out the Transmit Shift Register. This counter operates as a divide-by-16 (for word mode) or a divide-by-32 (for long mode), and is also used to generate the load pulse for transferring the contents of the Transmit Holding Register to the Transmit Shift Register.

Depending on the configuration of the ENBYTEWAIT and ENWORDWAIT control bits, the Transmit Bit Counter will also trigger the counting of the Transmit Inter-Packet Delay Counter, which is used in order to accommodate slower peripherals which cannot keep up with fast Magicbus data rates. This counter allows the insertion of a programmable number of baud clocks of delay (based on the DELAYVAL[7:0] control register value) between either bytes, words, or longs. Delay is always inserted after each long (32-bit) group of bits, but is only inserted between bytes or words if the respective wait enable control bits (ENBYTEWAIT and ENWORDWAIT) are asserted. In addition, if no delay is desired, then the DELAYVAL[7:0] register value should be set to “0”.

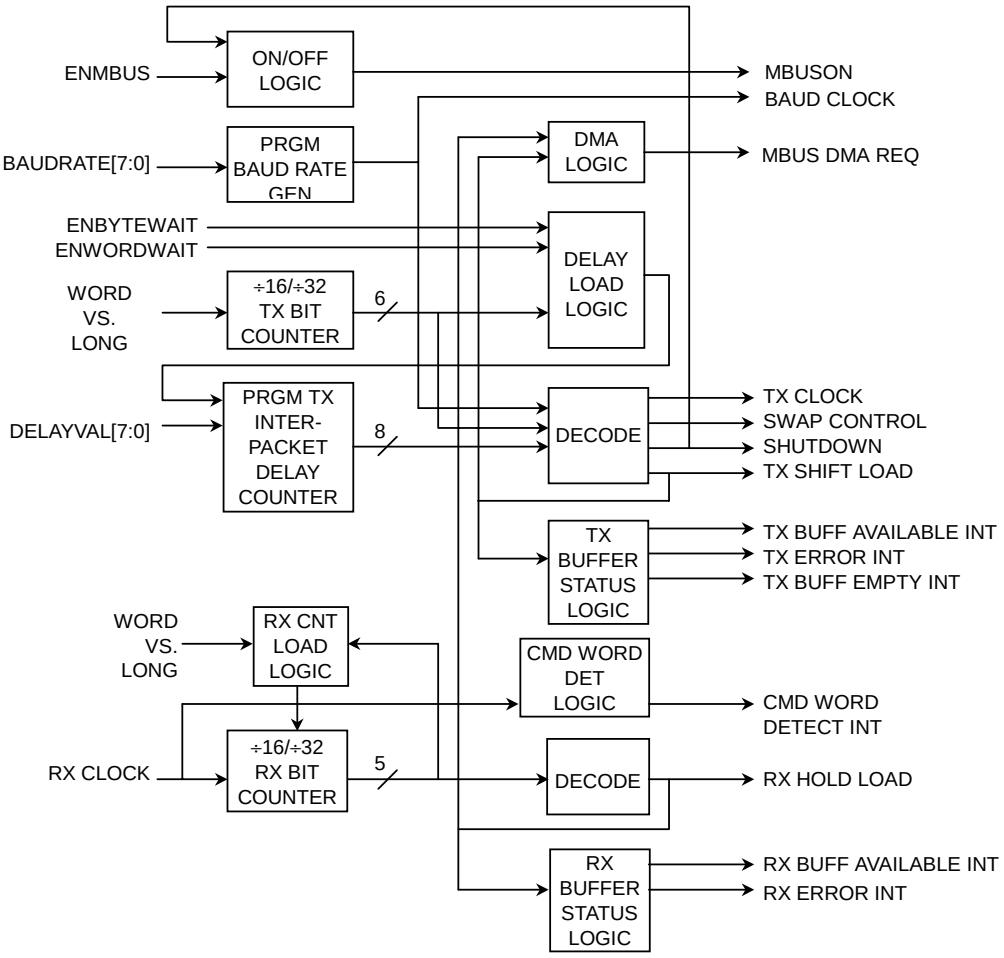


Figure 11.3.4 Magicbus Control Logic

The Magicbus Receiver control logic consists of a Receive Bit Counter and a circuit for detecting the receipt of a Command Word, whereby the clock and data signal functions are swapped. The Magicbus Receiver circuitry (including the Receive Bit Counter and the Receive Shift Register) uses the receive clock from the MBUSCLK port. This allows a peripheral to control the data rate of the Magicbus Receiver (allowing support for slower peripherals), since the receiver uses the peripheral's transmit clock as it's clock source.

The Receive Bit Counter is used to count the number of bits to be shifted into the Receive Shift Register. This counter operates as a divide-by-16 (for word mode) or a divide-by-32 (for long mode), and is also used to generate the load pulse for transferring the contents of the Receive Shift Register to the Receive Holding Register.

The Command Word detect logic monitors the MBUSCLK and MBUSDATA input lines and generates an interrupt (MBUSDETINT) whenever a Command Word is detected.

11.3.5 DMA Address Generation

The Magicbus Module provides support for both transmit and receive DMA. Since the Magicbus Module operates half-duplex, a single DMA channel is shared between transmit or receive, with only one direction using the buffer at a given time. The circuit used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts is shown in Figure 11.3.5.

The DMA buffer size is programmable (up to a maximum of 1 MByte) and the buffer start address is also programmable (anywhere over the full 32-bit address space). The DMA buffer can be configured in a circular buffer mode or a one-time buffer mode, depending on the configuration of the ENDMALOOP control bit. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TMPR3911/12's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and utilize the DMA buffer in a ping-pong fashion. For transmit mode, the CPU can use these interrupts to fill or write one half of the buffer while the other half is being emptied by the DMA controller for transmitting Magicbus data. Similarly, for receive mode, the CPU can use these interrupts to empty or read one half of the buffer while the other half is being filled by the DMA controller from received Magicbus data.

Also available is a direct CPU read/write mode for bypassing the DMA, allowing the CPU to read or write the Magicbus data on a word by word (or long by long) basis, if so desired. Separate DMA enables are provided for receive and transmit.

The DMA circuit also provides the DMA pointer status value, which is the actual 18-bit DMA address counter output. This value indicates exactly where the current address is pointing to in the overall DMA buffer, and is available via reading of the DMACNT[19:2] status register bits.

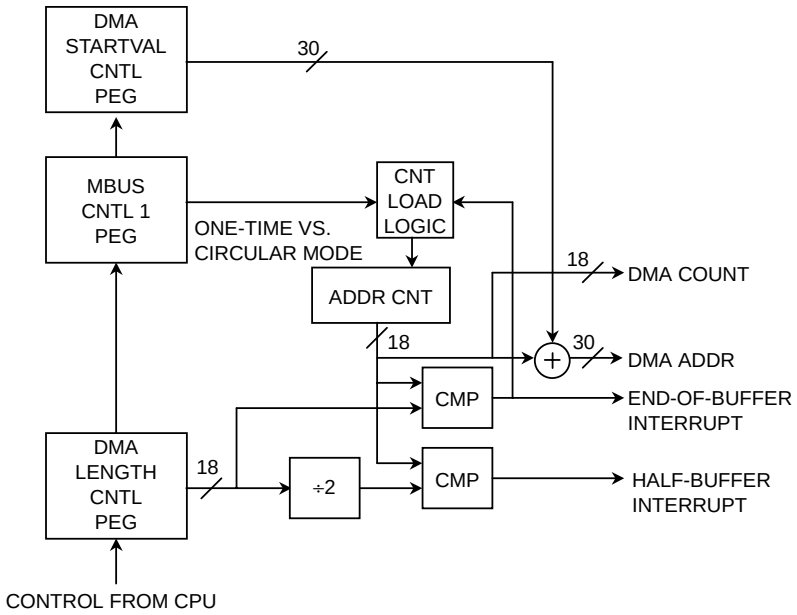


Figure 11.3.5 Magicbus DMA Address Generation

11.3.6 Related Interrupts

MBUSTXBUFAVAILINT:

This interrupt is set when the ENMBUS bit is first asserted and subsequently when the contents of the MBUS Transmitter Holding Register are transferred to the MBUS Shift Register. This interrupt is used to indicate that the MBUS Transmitter Holding Register is available to be written by the software.

MBUSTXERRINT:

Issues an interrupt if the Magicbus Transmit Holding Register is written to when the Transmit Holding Register is not available.

MBUSEMPTYINT:

Issues an interrupt if the Magicbus Transmit Holding Register and Transmit Shift Register are both empty.

MBUSRXBUFAVAILINT:

Issues an interrupt whenever the Magicbus Receive Holding Register is loaded with data.

MBUSRXERRINT:

Issues an interrupt if the Magicbus Receive Holding Register is loaded twice before the interrupt is serviced.

MBUSDETINT:

Issues an interrupt whenever the Magicbus Command Word detector triggers.

MBUSDMAFULLINT:

Issues an interrupt if the Magicbus DMA Counter reaches the end of the buffer.

MBUSDMAHALFINT:

Issues an interrupt if the Magicbus DMA Counter reaches the mid point of the buffer.

MBUSPOSINT:

Issues an interrupt if whenever the MBUSINT signal goes from a logic “0” to a logic “1”.

MBUSNEGINT:

Issues an interrupt if whenever the MBUSINT signal goes from a logic “1” to a logic “0”.

11.4 Magicbus Registers

11.4.1 Magicbus Control 1 Register

OFFSET = $0E0:

Bit	Label	RESET	Read/Write
31	MBUSON	0	R
30	EMPTY	1	R
29	MBUSINT	–	R
28-17	Reserved		
16	ENDMARX	0	R/W
15	ENDMATX	0	R/W
14	ENDMATEST	0	R/W
13	ENDMALOOP	0	R/W
12	LOOPBACK	0	R/W
11	RCVPHAPOL	0	R/W
10	DTIDLE	0	R/W
9	ENBYTEWAIT	0	R/W
8	ENWORDWAIT	0	R/W
7	DETPHAPOL	0	R/W
6	DETCLKPOL	0	R/W
5	PHAPOL	0	R/W
4	CLKPOL	0	R/W
3	SLAVE	0	R/W
2	FSLAVE	0	R/W
1	LONG	0	R/W
0	ENMBUS	0	R/W

MBUSON: read-only

When the ENMBUS bit is disabled, the module will not shutdown until the Magicbus Transmit Holding Register and Transmit Shift Register are empty plus a couple of Magicbus baud clocks of delay. This bit provides the status as to whether the module is still enabled or not.

EMPTY: read-only

This bit is asserted if the Magicbus Transmit Holding Register and Transmit Shift Register are both empty and the delay counter is finished inserting delay.

MBUSINT: read-only

This bit provides the status of the MBUSINT input pin.

ENDMARX:

This bit enables the DMA receive function. This bit should not be set until the Magicbus DMA Control 1 Register and Magicbus DMA Control 2 Register are setup and the module is enabled. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

ENDMATX:

This bit enables the DMA transmit function. This bit should not be set until the Magicbus DMA Control 1 Register and Magicbus DMA Control 2 Register are setup, the module is enabled and the empty flag is set. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

ENDMATEST:

This bit is used for IC testing and should not be set.

ENDMALOOP:

The DMA controller supports two modes, depending on the state of this bit. When ENDMALOOP is a logic “0”, the DMA controller will stop executing when it reaches the end of the DMA buffer. When ENDMALOOP is a logic “1”, the DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating.

LOOPBACK:

Setting this bit will cause the Magicbus Transmitter clock and data to be looped back into the receive clock and data. The data is inverted when this mode is selected.

RCVPHAPOL:

Setting this bit will cause the Magicbus Receiver clock to be inverted.

DTIDLE:

This bit defines the value that the MBUSDATA pin will idle to in between transmitted characters.

ENBYTEWAIT:

Setting this bit will cause a delay to be inserted between each byte in transmitted data. The length of the delay is defined by the value in the DELAYVAL bits.

ENWORDWAIT:

Setting this bit will cause a delay to be inserted between each word in transmitted data. The length of the delay is defined by the value in the DELAYVAL bits.

DETPHAPOL:

Setting this bit will cause the receive data pin to the Command Word clock to be inverted.

DETCLKPOL:

Setting this bit will cause the receive clock pin to the Command Word clear to be inverted.

PHAPOL:

Setting this bit will cause the Magicbus Transmitter to clock data out on the rising edge of MBUSCLK to be sampled by the peripherals on the falling edge of MBUSCLK.

CLKPOL:

Setting this bit will invert the MBUSCLK pin.

SLAVE:

Setting this bit will cause the MBUSCLK and MBUSDATA pins to be inputs and will enable the Magicbus Receiver circuit. The transition from master to slave will not take place until the Magicbus Transmit Holding Register and Transmit Shift Register are empty.

FSLAVE:

Setting this bit forces Slave Mode without regard to the status of the Magicbus Transmit Holding Register and Transmit Shift Register.

LONG:

Setting this bit will cause the Magicbus Transmitter and Magicbus Receiver to handle 32 bits per buffer. Otherwise 16 bits is assumed. This bit must be set for DMA transactions.

ENMBUS:

Setting this bit will enable the Magicbus Module. When this bit is cleared the Magicbus Module is kept in a reset state. But clearing this bit gives no effect on the Magicbus Control Register. This bit should not be set until all other control bits are setup.

11.4.2 Magicbus Control 2 Register

OFFSET = $0E4: write-only

Bit	Label	RESET	Read/Write
31-24	DELAYVAL[7:0]	X	W
23-16	BAUDRATE[7:0]	X	W
15-0	Reserved		

DELAYVAL[7:0]: write-only

These bits define the number of Magicbus baud clocks of delay to insert between either bytes, words, or longs. Delay is always inserted after a long, but is only inserted between bytes or words if the respective enables (ENBYTEWAIT or ENWORDWAIT) are set. If no delay is desired then the register should be set to zero.

BAUDRATE[7:0]: write-only

These bits define the baudrate of the Magicbus Transmitter. The following equation determines the baudrate ($f_{MBUSMCLK}$ is the frequency of the Magicbus master clock.):

$$\text{baudrate} = f_{MBUSMCLK} \div (\text{BAUDRATE}[7:0] * 2 + 2)$$

11.4.3 Magicbus DMA Control 1 Register

OFFSET = $0E8: write-only

Bit	Label	RESET	Read/Write
31-2	DMASTARTVAL[31:2]	X	W
1-0	Reserved		

DMASTARTVAL[31:2]: write-only

These bits define the start address for the DMA buffer.

11.4.4 Magicbus DMA Control 2 Register

OFFSET = $0EC: write-only

Bit	Label	RESET	Read/Write
31-20	Reserved		
19-2	DMALENGTH[19:2]	X	W
1-0	Reserved		

DMALENGTH[19:2]: write-only

These bits define the length of the DMA buffer. The last address in the DMA buffer is given by DMASTARTVAL[31:2] + DMALENGTH[19:2]. The value loaded into DMALENGTH should be equal to the desired buffer length –1.

11.4.5 Magicbus DMA Count

OFFSET = $0F0: read-only

Bit	Label	RESET	Read/Write
31-20	Reserved		
19-2	DMACNT[19:2]	X	R
1-0	Reserved		

DMACNT[19:2]: read-only

These bits provide the status of the DMA counter.

11.4.6 Magicbus Transmit Holding Register

OFFSET = $0F8: Normal write-only

OFFSET = $0F4: Swap write-only

Bit	Label	RESET	Read/Write
31-0	TXDATA[31:0]	X	W

TXDATA[31:0]: write-only

These bits are the data to be transmitted. This register should only be loaded after the MBUSTXBUFAVAILINT interrupt is set. Writing to the Swap location will cause the clock and data lines to be swapped for this character of data. For word mode, only bits 15:0 are valid.

11.4.7 Magicbus Receive Holding Register

OFFSET = $0F8: read-only

Bit	Label	RESET	Read/Write
31-0	RXDATA[31:0]	X	R

RXDATA[31:0]: read-only

These bits are the receive data. The bits are valid after the MBUSRXBUFAVAILINT interrupt is set. For word mode, only bits 15:0 are valid.

12. Power Module

12.1 Overview

12.1.1 System Power Control Overview

The System Power Supply must provide several power supplies to various parts of the system. The system may use the SPI serial control interface, along with the PWRCS signal, to control turning on or off the various power supplies. The software will only turn on the power supplies that are required to perform a particular task. The TMPR3911/12 must always be provided power as long as there is a good main or backup battery in the system, or if a battery charger is plugged in. Similarly, there may be other components in the system that must always be powered. These components, along with the TMPR3911/12, should be powered by the VSTANDBY line.

DRAM or SDRAM should be powered by the VCCDRAM line. The VCCDRAM power line should remain dead when the VSTANDBY power line is first activated, until PWRCS is asserted to turn on the system for the first time. After the system boots for the first time, the software can decide not to turn off VCCDRAM when the system is subsequently turned off, in order to preserve the contents of memory.

Most of the other components in the system, including ROM, and system buffers are powered by the VCC3 line. The VCC3 power line is powered only when the system is on. Other switchable power lines are provided for components such as the LCD, PC Card, etc. Each of these lines will normally be turned off by the software when powering down the system. When the system is on, these lines will only be turned on if the particular module is being used.

In addition to the System Power Supply control, the Power Module also contains logic to support the TMPR3911/12 initialization, clock enable function, CPU Stop Mode, stop timer, and power failure logic. These features are described in detail in the following sub-sections.

12.1.2 Power State Transition Diagram

Figure 12.1.1 shows the power state transition diagram of the TMPR3911/12 system. The details of transitions are described in he following subsections.

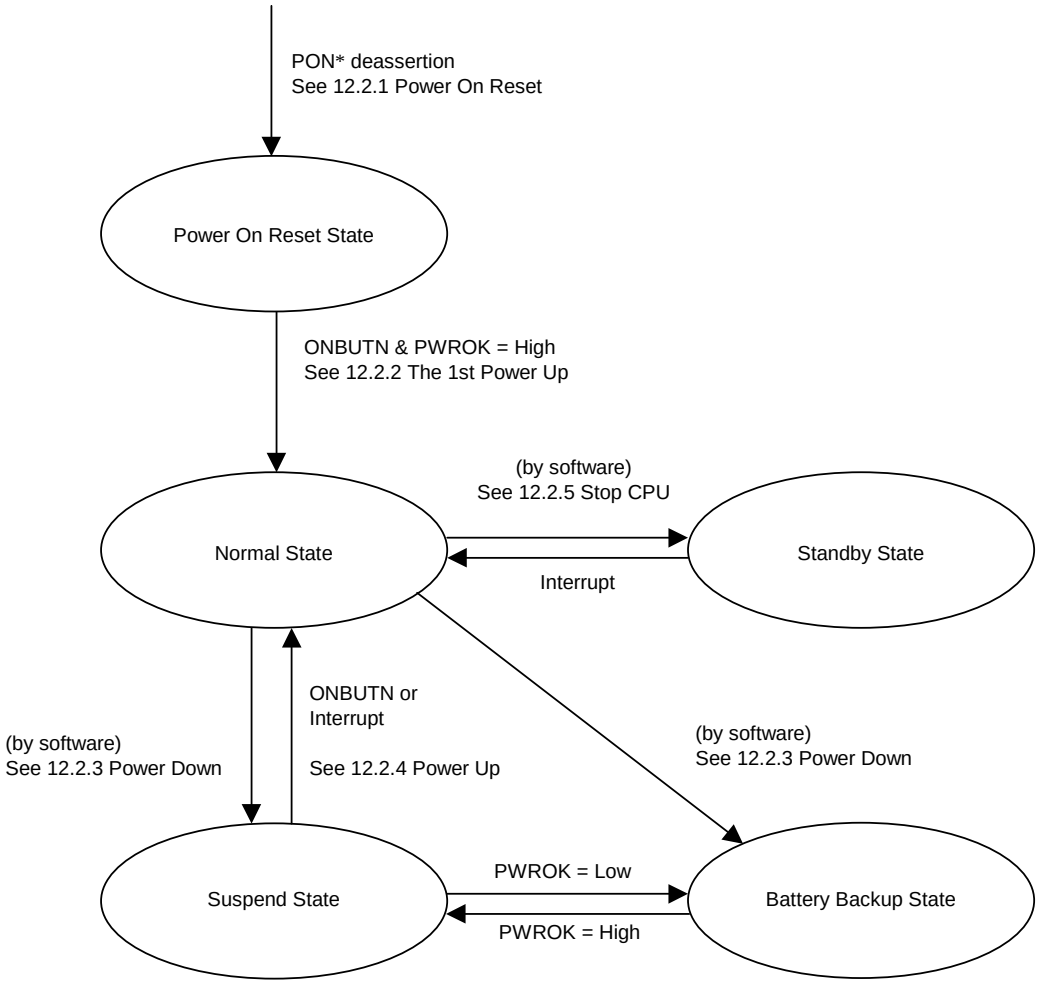


Figure 12.1.1 Power State Transition Diagram

12.1.3 Power Supply Lines in the System

The description in this section assumes that the system has the following power supply lines.

Table 12.1.1 shows the status of each power supply line on each power state described in 12.1.2

In addition to these power supply lines a System Power Supply may provide other power supply lines according to particular system requirements. For example a system may have a dedicated power supply line for a PC Card slot to supply power during the Suspend State.

VSTANDBY: (VDD)

This line provides power for the TMPR3911/12 and other components in the system that must never be powered off. This line should always be alive if there is either a good main battery or a good backup battery, or if a battery charger is plugged in.

VCCDRAM:

This line provides power for the DRAM and/or SDRAM. This line must be off when VSTANDBY is first powered, and remain off until the system is powered up by the assertion of PWRCS. When the software subsequently powers down the system it may choose to keep this line powered to preserve the contents of memory.

VCC3:

This line provides power for the ROM, system buffers, and other components in the system. This line will be powered by the System Power Supply when PWRCS is asserted and will always be powered off when the system is powered down.

Table 12.1.1 Power Supply Status

Power State	Power On Reset State	Normal State	Standby State	Suspend State	Battery Backup State
VSTANDBY	On	On	On	On	On
VCCDRAM	Off	On	On	On	On
VCC3	Off	On	On	Off	Off

12.1.4 Related Pins

PON*: INPUT

This pin serves as the Power On Reset signal for the TMPR3911/12. This signal must remain low when VSTANDBY is asserted until VSTANDBY is stable. Once VSTANDBY is powered, this signal should never go low unless all power is lost in the system.

ONBUTN: INPUT

This pin is used as the On Button for the system. Asserting this signal will cause PWRCS to be asserted to request the System Power Supply to turn on the system power. PWRCS can not be asserted if the PWROK signal is low.

PWROK: INPUT

This pin provides a status from the System Power Supply that there is a valid source of power in the system. This signal typically will be asserted if there is a battery charger supplying power or if the main battery is good and the battery door is closed. If the PWROK signal is low when the system is powered off, the PWRCS signal is not asserted by either the assertion of the ONBUTN signal or the interrupt detection. If the PWROK signal goes low while the device is on, the software should immediately shut down the system since power is about to be lost. When PWROK goes low, there must be serious warning so that the software can shut down the system before power is actually lost.

When the PWROKNMI bit in the Interrupt Status 6 Register is set, the deassertion of the PWROK signal causes the NMI (Non Maskable Interrupt).

PWRCS: OUTPUT

This signal is asserted automatically if the ONBUTN signal is asserted when the PWROK signal is high. Clearing the PWRCS bit in the Power Control Register by software deasserts this signal.

The usage of this signal depends on the implementation of the System Power Supply. For example, the assertion of this signal may usually cause the System Power Supply to turn VCCDRAM and VCC3 on to power up the system.

PWRINT: INPUT

This pin is used by the System Power Supply to alert the software that some status has changed in the System Power Supply and the software should read the status from the System Power Supply to find out what has changed. These will be low priority events, unlike the PWROK status, which is a high priority emergency case.

VCC3: INPUT

This pin provides the status of the power supply for the ROM, system buffers, and other components in the system. This signal will be asserted by the System Power Supply when PWRCS is asserted, and will always be turned off when the system is powered down.

12.2 Description

12.2.1 Power On Reset

When VSTANDBY is powered on, the PON* signal must be kept low until VSTANDBY is stable, in order to provide a good Power On Reset to the TMPR3911/12. The assertion of PON* will set a RESET flip-flop, as shown in Figure 12.2.1. The RESET flip-flop will remain set until the ONBUTN is pressed for the first time. This functionality provides a long reset time to the TMPR3911/12 when VSTANDBY is first asserted. The long reset time is needed to allow the 32 kHz oscillator to stabilize. The RESET signal is used to initialize all registers and logic in the TMPR3911/12.

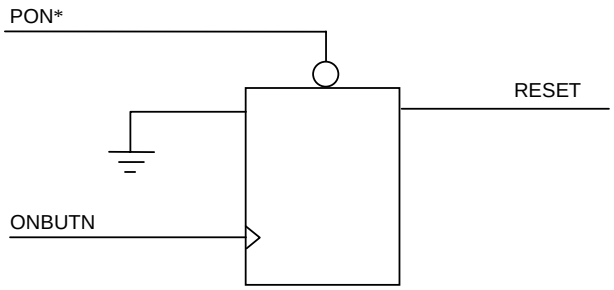


Figure 12.2.1 Reset Flip-Flop

12.2.2 The 1st Power Up

The 1st Power Up sequence is shown in Figure 12.2.2 When the ONBUTN is pressed, the PWRCS signal will be asserted if the PWROK signal is high. Once the PWRCS signal is asserted, the System Power Supply powers VCCDRAM and VCC3 in order to power up the system. Once VCC3 is powered, the high frequency oscillator and PLL that generate the clock signals for the CPU and BIU start clocking. When the oscillator and PLL are first enabled, the oscillation can vary widely until the oscillator and PLL become stable. The stabilization time can vary between 1 ms and 10 ms, depending on the oscillator and PLL of a particular TMPR3911/12 implementation. In order to prevent the TMPR3911/12 from using the clock before the clock becomes stable, a circuit in the Power Module will provide an internal signal ENSYSCLK that will disable the clock until the oscillator and PLL become stable. The circuit uses the 32 kHz signal as its reference, since this is known to be good because of the long Power On Reset time. The ENSYSCLK is asserted 16-24 ms after VCC3 is asserted at the 1st power up. Once the ENSYSCLK signal is asserted, the CPU will start instruction execution.

If the PWROK signal is low when the ONBUTN is pressed, the PWRCS signal will not be asserted.

The “PON* state” and the “1st-time power-up state” in the Figure 12.2.2 are defined on page 2-17.

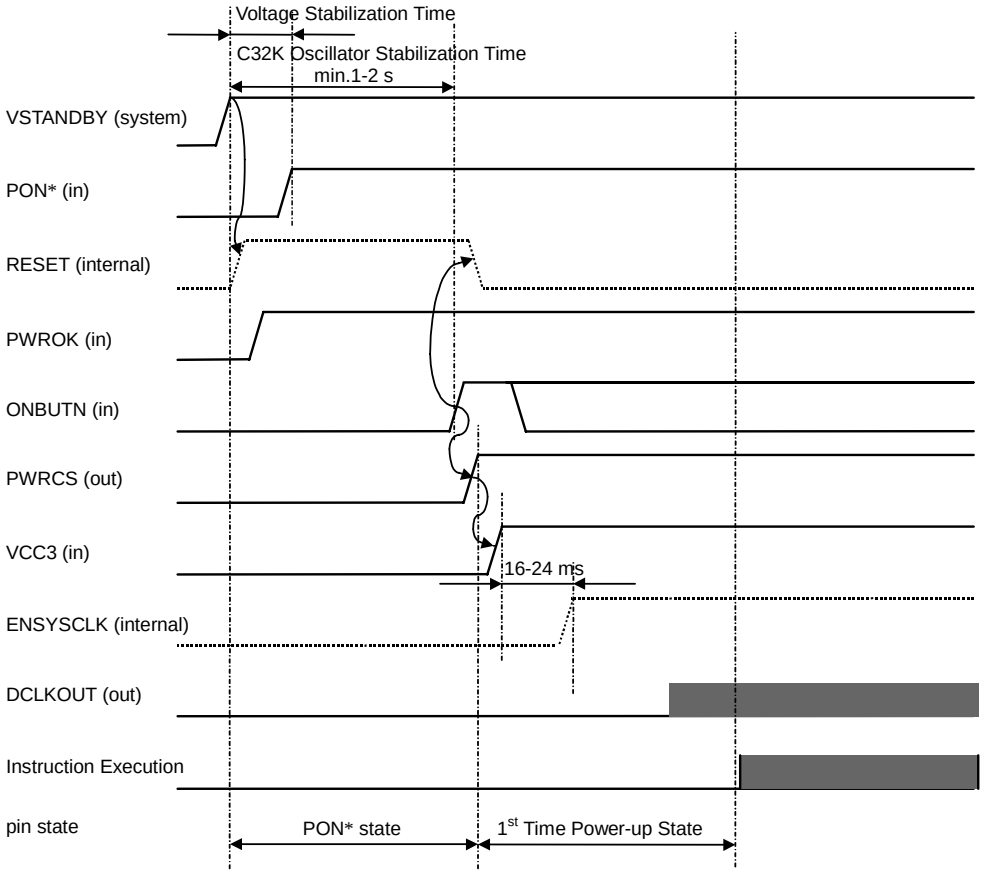


Figure 12.2.2 The 1st Power Up Sequence

12.2.3 Power Down

A state transition into the Suspend State is called 'Power Down'. Typically this is triggered by assertion of the ONBUTN signal to power off the system. In the Suspend State VSTANDBY and VCCDRAM are powered but VCC3 is not powered. Some system may have some other power supply lines which are powered during the Suspend State. For example a system may supply power to a PC Card during the Suspend State.

If the system is powered up and the PWROK signal is deasserted, the system must be powered down as quickly as possible to preserve the contents of DRAM and/or SDRAM. All power supply lines, which are not required to preserve valuable data, must be powered off. This state is called the Battery Backup State.

The system is powered down by software in the following sequence.

1. All modules in the TMPR3911/12 are disabled and all clocks in the Clock Module are turned off.
2. The interrupts for the wake-up events such as power switch, alarm (RTC), and serial DCD are enabled, and the other interrupts are disabled.
   When the PWROK signal is deasserted, some system may disable all interrupts to prevent unexpected power up after power will be supplied again.
3. The MEMPOWERDOWN bit in the Memory Configuration 4 Register is set in order to set the DRAM and/or SDRAM into self-refresh mode. The following instructions must be executed from cache memory. See Section 0 for more details.
4. By clearing the PWRCS and VCCON bits in the Power Control Register simultaneously the PWRCS signal is deasserted. After the PWRCS is deasserted, the System Power Supply shuts down VCC3 power line. Clearing VCCON bit results in negating ENSYSCLK and stopping all internal clocks except 32kHz clock.

A wait loop for a few milliseconds follows. A few instructions may be executed before power off. The rest of the wait loop is executed after the next power up to guarantee that some devices in the system are stable after power up.

12.2.4 Power Up

When the system is off if rising edge of the ONBUTN signal or an enabled interrupt occurs, and the PWROK signal is asserted, then the system will power up. If assertion of the ONBUTN signal or an enabled interrupt occurs, but the PWROK signal is deasserted, then the system will not power up at that time. But since the interrupt is latched, the system will power up once the PWROK signal is asserted. On the other hand, the assertion of the ONBUTN signal is not a latched event unless the POSONBUTNINT or NEGONBUTNINT interrupts are enabled. Therefore, the assertion of the ONBUTN signal will not wake up the system once PWROK signal is asserted.

Once the ENSYSCLK (internal signal) goes low, the clock to the CPU is stopped and the CPU will remain suspended at its last state. When the device powers back up, the ENSYSCLK will be asserted 16-24 ms after VCC3 is asserted or 4-6 ms after VCC3 is asserted, depending on the state of the SELC2MS control bit in the Power Control Register. Then the CPU will resume the instruction execution from where it stopped. It is probably desirable to loop for some time to make sure that the battery voltage has stabilized and the PWROK signal will not suddenly go low as a result of a substantial load added to the battery. Once the loop has completed, the software must clear the FORCESHUTDWN bit prior to accessing any address that is not in the cache or the internal function registers (See Section 12.2.6).

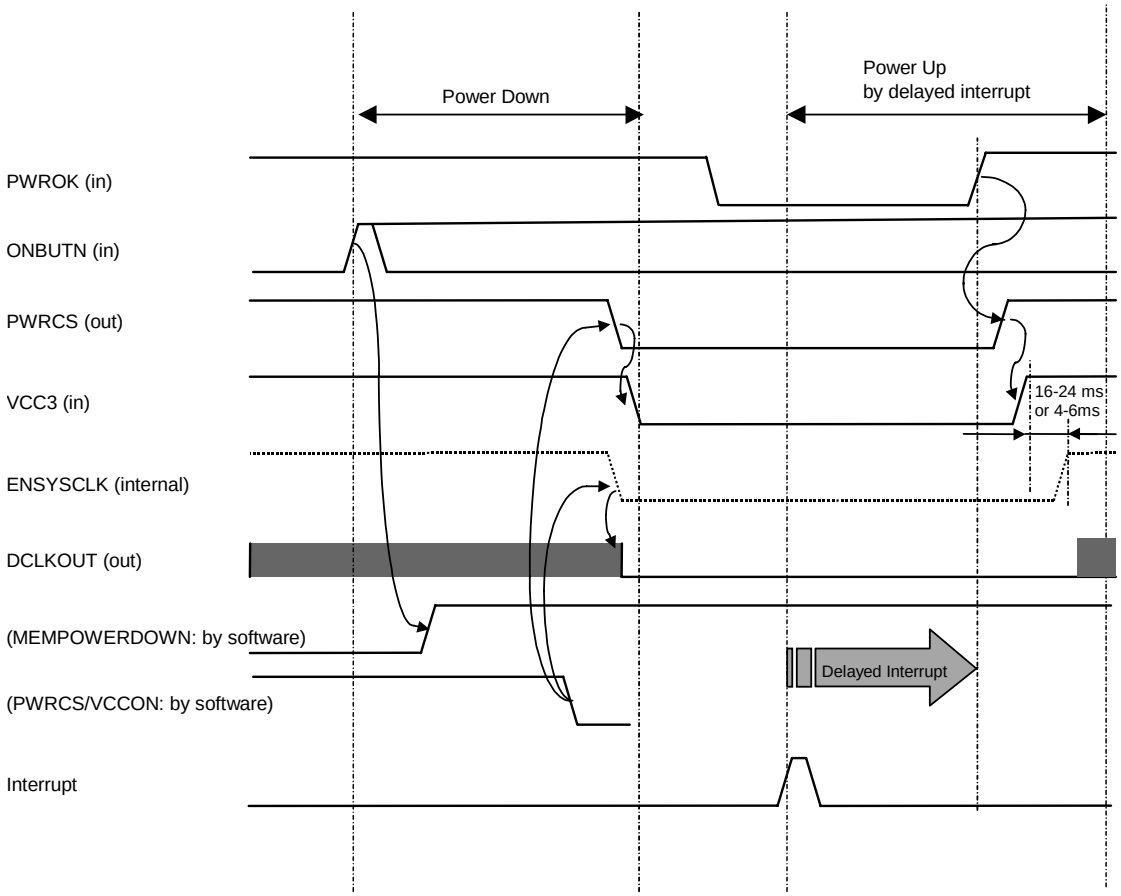


Figure 12.2.3 Power Down and Power Up Sequence

12.2.5 Stop CPU

In order to reduce the power consumption in the TMPR3911/12, it is desirable to stop the clock to the CPU when the CPU does not have to execute instructions. This is possible since the CPU core is fully static and will simply resume the instruction execution from where it stopped when the clock is re-enabled. The states in which just the CPU clock stops are called the Standby State or the Idle State.

The system goes into the Standby State by software in the following sequence.

The IEc bit in the Status Register of the CPU core is cleared to disable interrupt.

1. The software may change RF bits in the Config Register of the CPU core and SLOWBUS bit in the Power Control Register to reduce clock frequencies of DCLKOUT, CORECLK, CLK2X, and CLK (See Section 6.2.1) for power saving. The reduced frequency must be determined to maintain the system activity (DMA, peripheral device operation, etc.) in the Standby State.
2. The software can stop the clock to the CPU by setting the STOPCPU bit in the Power Control Register. Once this bit is set, the clock to the CPU will be stopped and remains stopped until an enabled interrupt clears the STOPCPU bit. This bit should not be set when powering down the system because the ENSYSCLK signal is a more dominant clock shutdown signal for the entire TMPR3911/12 chip.
3. A program loop waiting for the STOPCPU bit to be set follows. Some instructions are executed, then the CPU core stops.

When any enabled interrupt events occur the system goes back to the Normal State promptly. The following software sequence should be done.

1. The CPU core restarts the execution of the program loop waiting for the STOPCPU bit to be cleared and exits from the loop promptly since the STOPCPU bit is cleared.
2. Resume the value of the RF bits and SLOWBUS bit.
3. The IEc bit in the Status Register of the CPU core is set to enable interrupt.
4. The CPU core jumps to the interrupt handler for the interrupt that waked up the system.

The IEc bit is used to resume the RF bits and SLOWBUS bit before jumping to the interrupt handler.

12.2.6 Forced Shutdown

The Power Module provides a feature called Forced Shutdown that will force the system to power down automatically if the PWROK signal goes low before the software clears the FORCESHUTDWN bit. The main purpose of this feature is to protect the contents of DRAM or SDRAM. The timing sequence for Forced Shutdown is shown in Figure 12.2.4. This feature can be enabled or disabled by setting the ENFORCESHUTDWN bit in the Power Control Register. This feature is useful if a particular battery provides a PWROK indication when there is no load on the battery, but quickly deasserts the PWROK signal once the battery has some load by asserting VCC3. When this feature is disabled, the system will continue the instruction execution and enable the persistent memory even if the PWROK signal is deasserted, and will eventually stop by the power failure and lose the data in the memory.

The FORCESHUTDWN bit is set whenever PWRCS is asserted as a result of the ONBUTN being pressed or an enabled interrupt being set when the device is off. If the PWROK signal goes low prior to the software clearing the FORCESHUTDWN bit, the Power Module will simply deassert the PWRCS signal and the System Power Supply will turn off the VCC3 power supply.

The instruction to clear the FORCESHUTDWN bit must be executed prior to bringing the memory interface out of self-refresh mode. This implies that the instruction has to be pre-loaded into the cache before the previous system shutdown, since the memory interface will exit self refresh mode once an address location is accessed in the system that is not in the cache or not to a function register. The very first time that the system is turned on after the assertion of VSTANDBY, it is not possible to support this function since the cache is not initialized. This is not a problem because the main purpose of this feature is to protect the contents of DRAM or SDRAM, which does not exist when the device is turned on for the first time.

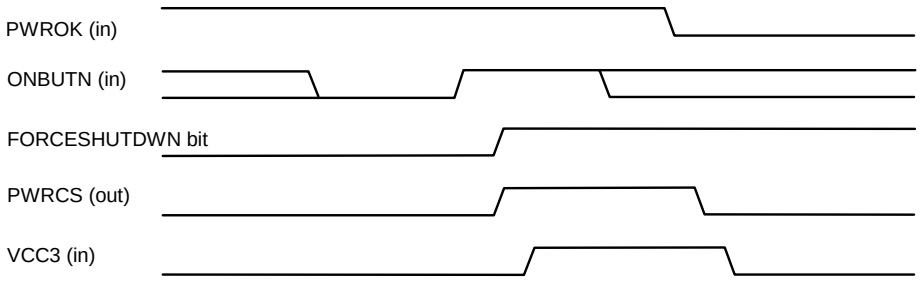


Figure 12.2.4 Forced Shutdown Sequence

12.2.7 Memory Power Down

By setting the MEMPOWERDOWN bit in the Memory Configuration 4 Register, the memory controller goes into Memory Power Down Mode. All memory signals will be driven low and DRAMs and/or SDRAMs will go into self-refresh mode. The software can only access to the cache memory and the internal function registers. The first access outside them clears the MEMPOWERDOWN bit and DRAMs and/or SDRAMs will go out of the self-refresh mode. Thus it is necessary that all instructions following the assertion of the MEMPOWERDOWN bit need to be executed on the cache memories.

12.2.8 Stop Timer

The Stop Timer is used as a coarse timer to bring the CPU out of Stop Mode. The Stop Timer consists of a 4-bit up counter and a comparator to generate the STPTIMERINT interrupt. The Stop Timer is enabled using the ENSTPTIMER control bit in the Power Control Register. When the ENSTPTIMER control bit is not set the counter will be reset to zero. Once the ENSTPTIMER bit is set the counter will count up using an 8 ms clock as the input. Once the counter reaches a value that is equal to the STPTIMERVAL[3:0] control bits, the STPTIMERINT interrupt is set. The Stop Timer will provide a maximum duration of 120 ms, in steps of 8 ms.

12.2.9 Power Module Interrupts

STPTIMERINT:

This interrupt is set whenever the Stop Timer Counter counts up to the value set by the STPTIMERVAL[3:0] control bits.

POSPWRINT:

This interrupt is set when the PWRINT pin changes from a logic “0” to a logic “1”.

NEGPWRINT:

This interrupt is set when the PWRINT pin changes from a logic “1” to a logic “0”.

POSPWROKINT:

Issues an interrupt whenever the PWROK signal changes from a logic “0” to a logic “1”.

NEGPWROKINT:

Issues an interrupt whenever the PWROK signal changes from a logic “1” to a logic “0”.

POSONBUTNINT:

Issues an interrupt whenever the ONBUTN signal changes from a logic “0” to a logic “1”. If the DBNCONBUTN control bit is set then the interrupt will not be set until the signal is debounced for 16-24 ms.

NEGONBUTNINT:

Issues an interrupt whenever the ONBUTN signal changes from a logic “1” to a logic “0”. If the DBNCONBUTN control bit is set then the interrupt will not be set until the signal is debounced for 16-24 ms.

12.3 Power Registers

12.3.1 Power Control Register

OFFSET = $ 1C4:

Bit	Label	RESET	Read/Write
31	ONBUTN	–	R
30	PWRINT	–	R
29	PWROK	–	R
28-27	VIDRF[1:0]	0	R/W
26	SLOWBUS	0	R/W
25	DIVMOD	0	R/W
24-16	Reserved		
15-12	STPTIMERVAL[3:0]	X	R/W
11	ENSTPTIMER	0	R/W
10	ENFORCESHUTDWN	0	R/W
9	FORCESHUTDWN	0	R/W
8	FORCESHUTDWNOCC	0	R/W
7	SELC2MS	0	R/W
6	Reserved		
5	BPDBVCC3	0	R/W
4	STOPCPU	0	R/W
3	DBNCONBUTN	0	R/W
2	COLDSTART	1	R/W
1	PWRCS	0	R/W
0	VCCON	0	R/W

ONBUTN: read-only

This bit provides the status of the ONBUTN signal.

PWRINT: read-only

This bit provides the status of the PWRINT signal.

PWROK: read-only

This bit provides the status of the PWROK signal.

STPTIMERVAL[3:0]:

When the Stop Timer is enabled the STPTIMERINT interrupt will be set when the counter is equal to the value set by these bits.

ENSTPTIMER:

This bit is used to enable the Stop Timer.

VIDRF[1:0]:

These bits select the divide-ratio of VIDCLK, which is the master clock used for the video module.

VIDRF	divide-ratio
0	1
1	2
2	4
3	8

SLOWBUS:

Setting this bit will cause the internal clocks except FREECLK and XHFEE to be divided by 2.

DIVMOD:

Setting this bit will cause the clocks in each function module such as an UART to be independent of RF (1:0). (See Section 6, Figure 6.2.1 and 6.2.2)

ENFORCESHUTDWN:

The FORCESHUTDWN bit is set whenever the VCCON bit and the PWRCS bit are set by either the ONBUTN signal assertion or an enabled interrupt occurrence. If the PWROK signal goes low before the processor clears the FORCESHUTDWN bit, then the VCCON bit and the PWRCS bit are cleared and the PWRCS signal will go low if the ENFORCESHUTDWN bit is set. If the ENFORCESHUTDWN bit is not set, then the FORCESHUTDWN bit will have no effect.

FORCESHUTDWN:

If the ENFORCESHUTDWN bit is set, then this bit must be cleared after the initial boot, but prior to memory interface waking up procedure. See Section 12.2.6 for details.

FORCESHUTDWNOCC:

This bit is set if a Forced Shutdown occurs due to the PWROK signal going low prior to the processor clearing the FORCESHUTDWN bit. This bit provides status to the processor to indicate that the event occurred. The bit should be cleared once the status has been checked, so that it can be used again for the next power up.

SELC2MS:

The assertion of PWRCS signal will cause the System Power Supply to assert the VCC3 power line. Once VCC3 is powered, the oscillator and PLL inside of the TMPR3911/12 will wake up and generate the main system clock, but for a certain period of time the clock will be unstable until the oscillator and PLL lock on the correct frequency. The Power Module will prevent the clock from driving any logic until the clock becomes stable. The reference clock for the debouncing delay time is provided by the C2MS (2 ms clock) and the C8MS (8 ms clock) internal signals generated by the RTC from the 32 kHz oscillator. If SELC2MS is set to a logic “1”, the clock will be disabled for a minimum of 4 ms and a maximum of 6 ms after VCC3 is powered. If SELC2MS is set to a logic “0”, the clock will be deasserted for a minimum of 16 ms and a maximum of 24 ms after VCC3 is powered.

BPDBVCC3:

If this bit is set, the clock will be enabled immediately following the assertion of VCC3, instead of waiting for the delay mentioned in the SELC2MS bit description.

STOPCPU:

Setting this bit will cause the clock to the CPU core to be disabled to allow low power operation. The bit is cleared whenever an enabled interrupt is set. This bit should not be set when powering down the system. It is only used when the system is powered up, but the CPU does not have to execute instructions.

DBNCONBUTN:

If this bit is set, then the POSONBUTNINT and NEGONBUTNINT interrupts and the ONBUTN wake up function will only occur after a minimum of 16 ms and a maximum of 24 ms of debouncing of the ONBUTN signal has taken place to allow direct “undebounced” button connection. If this bit is cleared, then the interrupts and power up occur immediately upon transition of the ONBUTN signal.

COLDSTART:

This bit is set by RESET and provides status to the processor that a Power On Reset has occurred.

PWRCS:

This bit is set whenever the ONBUTN signal is asserted or an enabled interrupt occurs while the PWROK signal is high. When powering down the system, this bit must be cleared simultaneously with the VCCON bit. All enabled interrupts must be cleared prior to powering down the system, or the PWRCS and VCCON signals will not go low.

VCCON:

This bit is set whenever the ONBUTN is pressed or an enabled interrupt occurs if the PWROK signal is high. When powering down the system, this bit must be cleared simultaneously with the PWRCS bit. All enabled interrupts must be cleared prior to powering down the system, or the PWRCS and VCCON signals will not go low.

13. SIB Module

13.1 Overview

The SIB Module within the TMPR3911/12 contains holding registers, shift registers, and other logic to support interfacing to the analog-front-end device: TC35143 and/or other optional external codec devices.

The overall sound subsystem for a PIC (Personal Information Communicator) allows playing and recording of sounds, and consists of a single-channel 12-bit codec within TC35143, TC35143 interface circuits for direct connection to an external microphone (MIC) and speaker, and DMA support within the TMPR3911/12.

Similarly, the overall telecom subsystem for a PIC allows support of high-performance software modems, and consists of a single-channel 16-bit codec within TC35143, which also includes optional echo cancellation and interface circuits for connection to an external Data Access Arrangement (DAA), as well as an auxiliary input/output port for supporting future wireless interfaces. The DAA provides the front-end interface circuitry needed between the analog telephone network and the telecom codec for wireline configurations. The front-end circuit can also include ring detect, off-hook detect, and connect detect functions.

Audio and telecom data and control/status information (such as sampling rate, gain control, muting, clip detect, etc.) is passed between the TMPR3911/12 and TC35143 (and/or other external codec devices) via the SIB. Within the TMPR3911/12, the SIB Module logic provides DMA support and control/status registers for data transfers between external system memory, the CPU core, and the SIB. SIB data transfer is always synchronous and is frame-based, with the TMPR3911/12 side always the master source of the clock and frame frequency and phase.

For the sound subsystem, TC35143 allows the system to digitize (using the sound codec's ADC) the MIC input, as well as to convert (using the sound codec's DAC) sampled sounds (either pre-stored, synthesized, or previously recorded) stored in system memory to an analog audio output, for routing to either a speaker and/or headphone output. For the telecom subsystem, TC35143 allows the system to digitize (using the telecom codec's ADC) the telecom analog baseband receive input, and the subsequent modem baseband signal processing can be implemented using entirely a software-based approach. TC35143 then allows the system to convert (using the telecom codec's DAC) the processed baseband transmit output to an analog telecom output for routing back out the wireline or wireless port.

The SIB Module provides full-duplex DMA support for sound receive and transmit, as well as independent DMA support for telecom receive and transmit (four total independent DMA channels). The DMA buffers can be configured in a continuous (circular) buffer mode or a one-time (empty or fill, then stop) buffer mode. Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and efficiently empty or fill half of the DMA buffer in a ping-pong fashion. The DMA buffer size is programmable (up to a maximum of 16 KBytes) and the receive and transmit buffers can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation). Also available is a direct CPU read/write mode for bypassing the DMA, allowing the CPU to read or write the sound or telecom data on a sample by sample basis, if so desired.

13.1.1 Related Pins

SIBDIN: INPUT

This pin contains the input data shifted from TC35143 and/or external codec device.

SIBDOUT: OUTPUT

This pin contains the output data shifted to TC35143 and/or external codec device.

SIBSCLK: OUTPUT

This pin is the serial clock sent to TC35143 and/or external codec device to synchronize the data transfer.

SIBSYNC: OUTPUT

This pin is the frame synchronization signal sent to TC35143 and/or external codec device. This frame sync is asserted for one clock cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data.

SIBIRQ: INPUT

This pin is a general purpose input port used for the SIB interrupt source from TC35143. This interrupt source can be configured to generate an interrupt on either a positive and/or negative edge.

SIBMCLK: INPUT/OUTPUT

This pin is the master clock source for the SIB logic. This pin can be used as output mode or input mode.

In output mode, SIBMCLK can be configured as a high-rate output master clock source required by certain external codec devices. In this mode all SIB clocks are synchronously slaved to the main TMPR3911/12 system clock CLK2X.

In input mode, SIBMCLK can be configured as an input slave clock source. In this mode, all SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to CLK2X. Also, for this mode, SIBMCLK can still be optionally used as a high-rate master clock source required by certain external codec devices.

13.2 Block Diagram

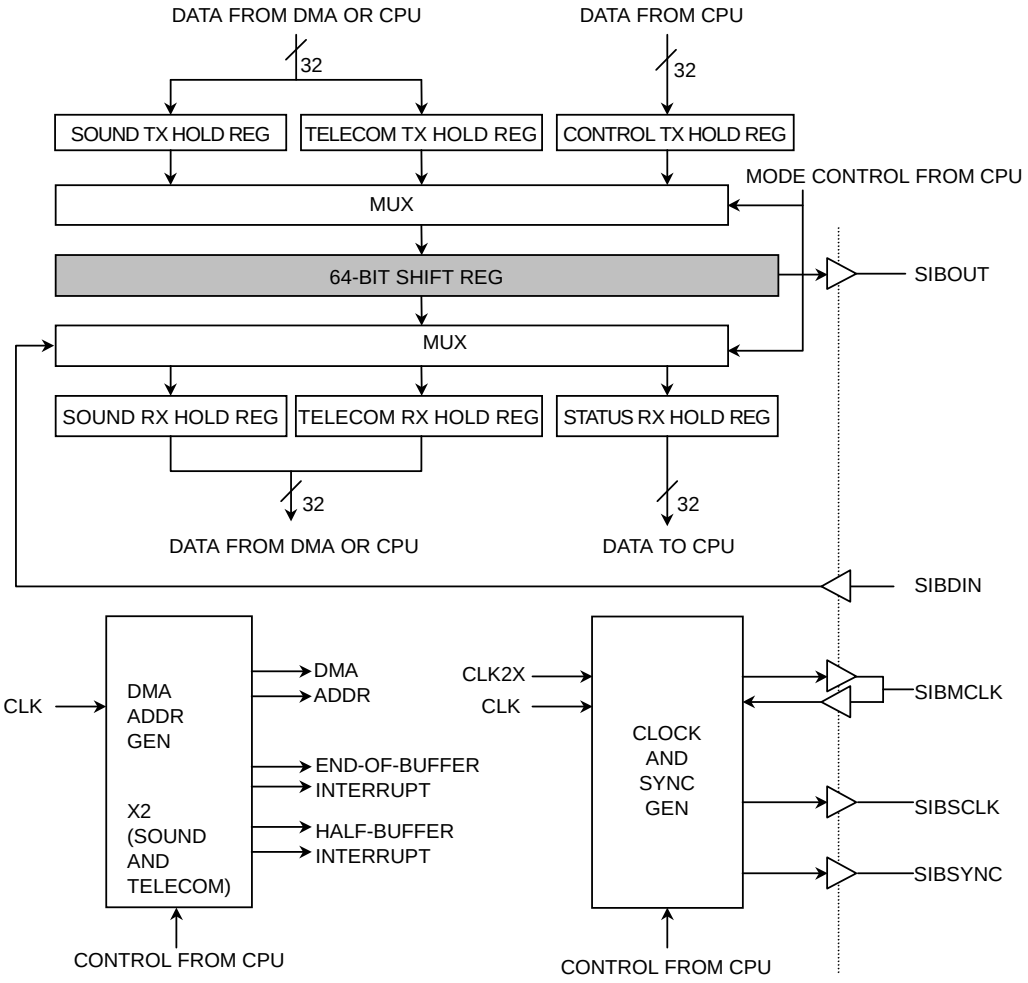


Figure 13.2.1 SIB Module Block Diagram

13.2.1 Holding and Shift Registers

The SIB Module logic consists of holding registers (both transmit and receive) and a serial shift register to support the transfer of sound and telecom data and control/status information between the TMPR3911/12 and TC35143 (and/or other external codec devices) via the SIB (see Figure 13.2.1 for a block diagram of the SIB Module).

For the SIB transmit direction, the sound and telecom transmit holding registers are written either from the DMA circuit or directly from the CPU, while the subframe 0 and subframe 1 control holding registers are always written directly from the CPU. These holding registers are processed by an array of multiplexers which select which fields of the various 32-bit transmit holding registers are loaded into the 64-bit shift register, such that an entire subframe is loaded in parallel and then shifted out the serial output SIBDOUT. The sequence of loading the shift register from the various holding register fields is determined by the programmed sound and telecom sample rates, mono versus stereo mode, and 8-bit versus 16-bit modes.

Conversely, for the SIB receive direction, the sound and telecom receive holding registers are read either by the DMA circuit or directly by the CPU, while the subframe 0 and subframe 1 status holding registers are always read directly by the CPU. The 64-bit shift register containing an entire subframe of received data from the serial input SIBDIN is processed by an array of multiplexers which select which fields of the various 32-bit receive holding registers are loaded from the shift register. The sequence of loading the various fields of the receive holding registers from the shift register is determined by the programmed sound and telecom sample rates, mono versus stereo mode, left versus right mono source, and 8-bit versus 16-bit modes.

13.2.2 Clock and Sync Generation

The SIB Module logic contains several programmable counters which are used to generate the various SIB internal and external control signals and clocks. See Figure 13.2.2 for a block diagram of the SIB clock and sync generation circuit. As mentioned previously, SIBMCLK can be configured as either an input or output. As an output, SIBMCLK is derived by dividing down from CLK2X. In this mode, all SIB clocks are then synchronously slaved to the main TMPR3911/12 system clock and only one main XTAL source is required for the entire system (besides the 32 kHz RTC XTAL), resulting in the lowest cost system configuration with regards to clock sources. As an input, SIBMCLK is generated from an external oscillator source, which is asynchronous with respect to CLK2X. This more expensive configuration allows the SIB (and also potentially the CHI and dual-UART circuits) to operate independent of the frequency used for the CPU core. This allows flexible system design options with respect to the CPU operating frequency, along with the ability to generate the required or desired UART baud rates and audio/telecom sampling frequencies.

The programmable SIBSCLK rate is derived by dividing down from SIBMCLK. Nominal rates for a TC35143-only configuration are CLK2X ÷ 5 (TMPR3912AU-92/XB-92) and SIBSCLK = 9.216 MHz (equals SIBMCLK ÷ 2). The SIBSCLK rate is then used to enable a set of counters (word counter and subframe/frame counter) to generate the SIBSYNC frame rate of 72 kHz (equals SIBSCLK ÷ 128), as well as various internal SIB clocks and control signals.

The subframe rate is used to enable independent sets of programmable sound and telecom sample rate counters which generate various control signals based on the desired sample rates. For instance, this logic correctly generates the sound and telecom valid flag bits which are asserted within the SIBDOUT serial output stream every time a valid sample is being transmitted.

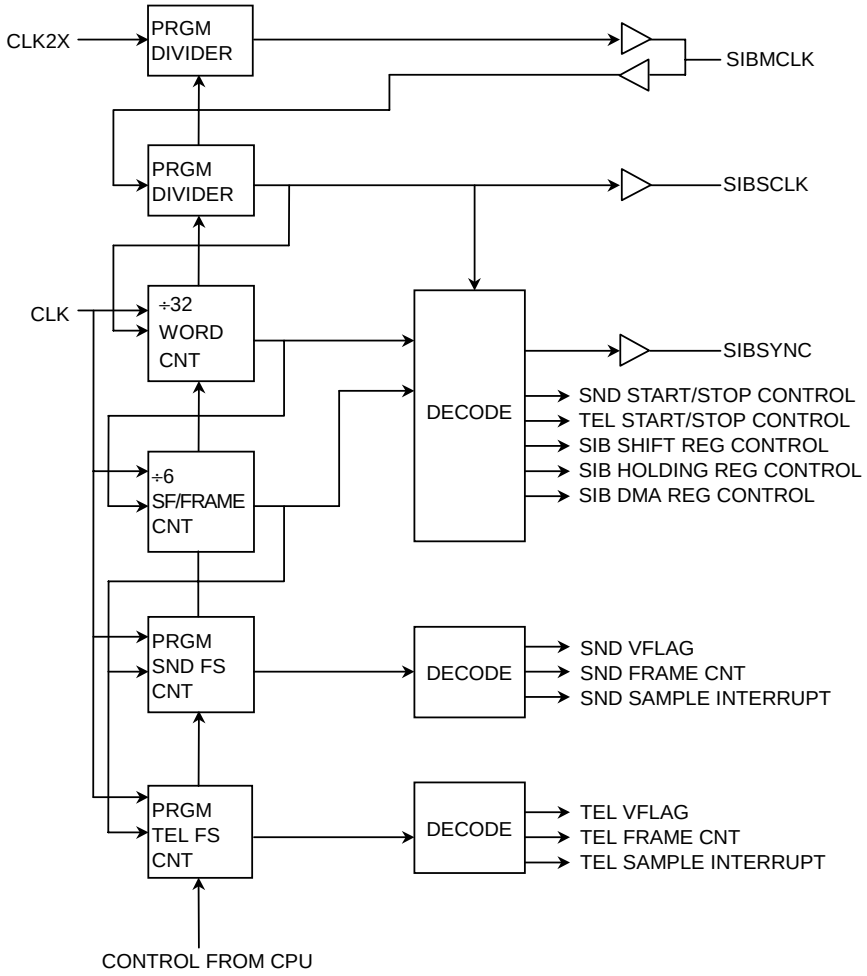


Figure 13.2.2 SIB Clock and Sync Generation

13.3 Interface Requirements

13.3.1 Frame Structure

Each SIB frame consists of 128 SIBSCLK, further divided into 2 subframes: Subframe 0 and 1. The SIB frame structure is shown in Figure 13.3.1.

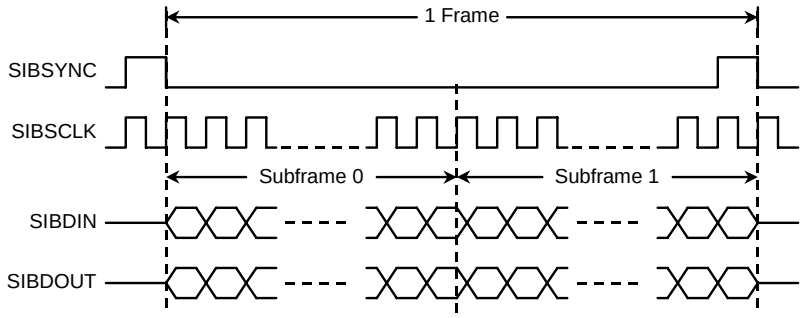


Figure 13.3.1 SIB Framing Structure

13.3.2 Timing Requirements

The TMPR3911/12 always samples receive data (SIBDIN) on the falling edge of SIBSCLK, whereas the transmit data from TC35143 is always output on the rising edge of SIBSCLK. Similarly, the TMPR3911/12 always output transmit data (SIBDOUT) on the rising edge of SIBSCLK, whereas the transmit data to TC35143 is always sampled on the falling edge of SIBSCLK in TC35143.

The SIBSYNC signal is asserted for one SIBCLK cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data. SIBSYNC is always sampled by TC35143 on the falling edge of SIBSCLK. The timing relationship for the SIBSCLK, SIBSYNC, and data signals (SIBDIN, SIBDOUT) is shown in Figure 13.3.2.

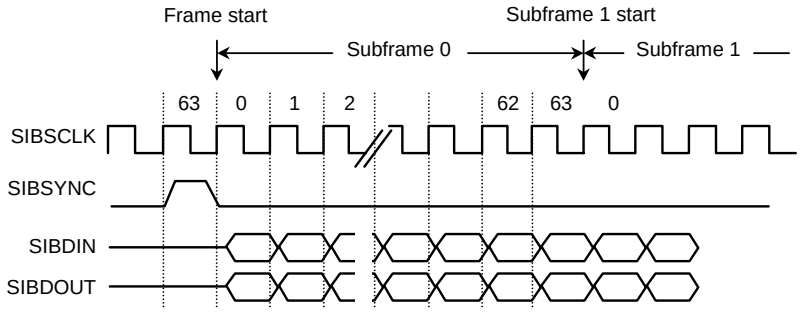


Figure 13.3.2 SIB Timing Relationships

13.3.3 Configurations

Many configuration options exist for the SIB. The SIB Module control bits SELSNDSF1 and SELTELSF1 are used to determine the desired SIB configuration. See Table 13.3.1 for a summary matrix of these possible configurations. The lowest cost configuration utilizes TC35143 only, assigned to subframe 0 of the SIB, while subframe 1 is not used. For this configuration, the system utilizes the audio and telecom codecs within TC35143. The SIBSCLK rate is fixed and the independent sound and telecom sample rates are controlled via TMPR3911/12 and TC35143 programmable counters.

Table 13.3.1 SIB Configuration Matrix (1)

TMPR3912AU-92/XB-92; 11.520 MHz crystal to be used. RF[1:0] = 00, SLOWBUS = 0.

sound codec	telecom codec	TMPR3912 clk2x [1]	sibsclk	sibmclk out or sound OSC in [2]	SIB frame rate [3]	comments
TC35143; Fs = 8K, 11.05K, 16K, or 22.15K	TC35143; Fs = 7.2K, 8K, or 9.6K	92.16M	9.216M (÷10)	18.432M out (÷5)	72K	TC35143 only; lowest cost; sibsclk = fixed; sound & telecom Fs control; via TC35143 prgm [4]; SIB subframe0 = TC35143; SIB subframe1 = not used; SELSNDSF1 = 0; SELTELSF1 = 0;

Table 13.3.2 SIB Configuration Matrix (2)

The TMPR3911BU/BXB SIB does not provide support for a 7.3728-MHz crystal. To use the SIB, use a 6.912-MHz crystal. RF[1:0]=00, SLOWBUS=0.

sound codec	telecom codec	TMPR3912 clk2x [1]	sibsclk	sibmclk out or sound OSC in [2]	SIB frame rate[3]	comments
TC35143; Fs = 8 k 11.05 k, 16 k, or 22.15 k	TC35143; Fs = 7.2 k, 8 k, or 9.6 k	55.296 M	9.216 M (÷6)	18.432 M out (÷3)	72 k	TC35143 only; lowest cost; sibsclk = fixed; sound & telecom Fs control; via TC35143 prgm [4]; SIB subframe1 = not used; SELSNDSF1 = 0; SELTELSF1 = 0;

NOTES :

[1] TMPR3911/12 clk2x is internal high-rate system-wide clock

[2] optional external sound OSC can be connected to TMPR3911/12 sibmclk (to allow decoupling of SIB clocks from clk2x; else this pin can be configured as buffered sibmclk output which is synchronously divided down from clk2x (sibmclk allows potential support of codecs requiring a high-rate fixed master clock + a sample-rate-dependent serial clock); sibsclk serial clock is always synchronously divided down from sibmclk

[3] SIB frame rate = sibsclk ÷ 128; ratio for average number of SIB-frames per valid-data-frame is dependent upon sound & telecom Fs

[4] TC35143 audio/telecom Fs divider programmability: SIB-frame/valid-data-frame RATIO

(sibsclk × 2) ÷ (40 × 64) = 7.2K	5
(sibsclk × 2) ÷ (36 × 64) = 8K	4.5
(sibsclk × 2) ÷ (30 × 64) = 9.6K	3.75
(sibsclk × 2) ÷ (15 × 64) = 19.2K (1.6% error vs. 18.9K for CD-XA)	1.875
(sibsclk × 2) ÷ (13 × 64) = 22.154K (0.47% error vs. 22.050K for CD-Audio)	1.625
(sibsclk × 2) ÷ (12 × 64) = 24K	1.5

13.3.4 Sample Rates

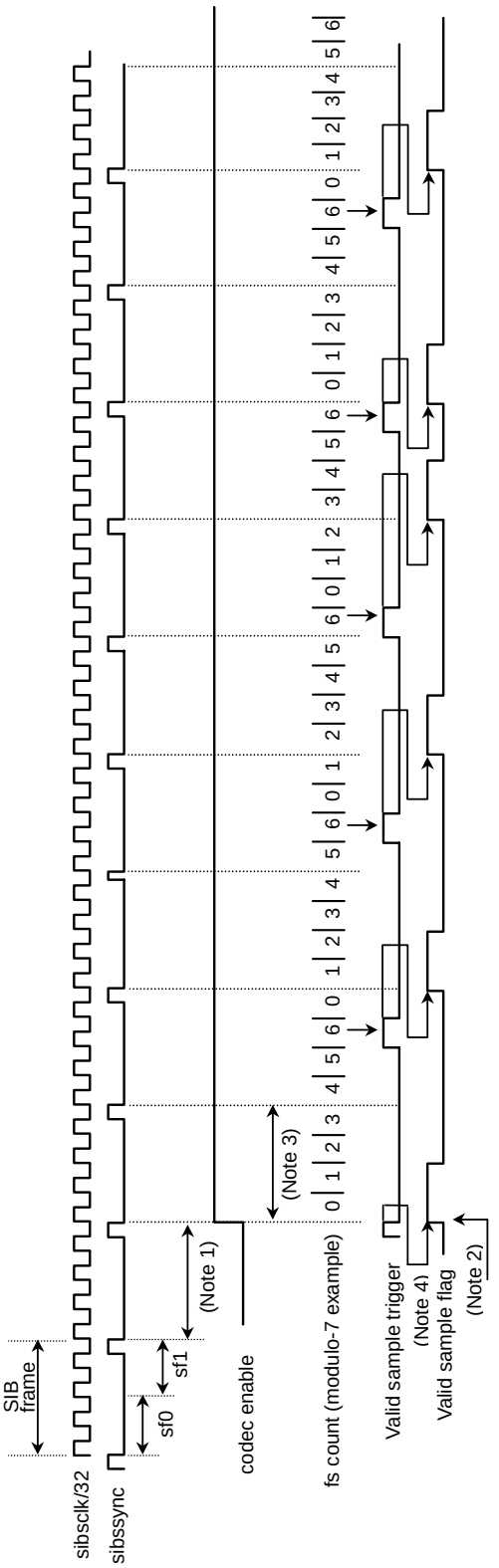
The SNDFSDIV and TELFSDIV control bits are used to independently configure the sampling rate for the audio and telecom codecs, respectively. These control bits set the modulus of the audio and telecom sample rate counters within the SIB Module. TC35143 also contains its own set of audio and telecom sample rate counters for its internal use and these counters must be both frequency and phase synchronized to the sample rate counters within the TMPR3911/12 in order to guarantee correct operation. The synchronization of these counters is triggered from the codec enable command transmitted by the TMPR3911/12 and received by TC35143, as shown in Figure 13.3.3.

The sample rate is calculated using the following (note that the FSDIV value loaded into the counter is the desired count modulus −1):

$$Fs = (SIBSCLK \times 2) \div ((FSDIV + 1) \times 64)$$

For example, if SIBSCLK = 9.216 MHz and FSDIV = 39 (modulo 40), then:

$$Fs = (9.216\ MHz \times 2) \div ((39 + 1) \times 64)$$
$$= 7.2\ kHz$$



Note 1: during this frame, audio (and/or telecom) codec input or output enable command is transmitted by TMPR3911/12 and received by TC35143F
Note 2: at start of first frame after codec is enabled, audio (and/or telecom) counters are reset; this ensures that TMPR3911/12 and TC35143F fs counters are immediately phase-synchronized
Note 3: during this frame, first valid DAC sample is transmitted by TMPR3911/12 (received by TC35143F) or first valid ADC sample is transmitted by TC35143F (received by TMPR3911/12); valid DAC and ADC samples are always transmitted during the same subframe (full duplex)
Note 4: audio (and/or telecom) vflag bits are asserted to TC35143F for one subframe each time a new DAC sample is being transmitted (TMPR3911/12 will also latch a new ADC sample from TC35143F during the same subframe); since the audio (and/or telecom) programmable sample rate is derived from sibsclk/32 (= 4x frame rate), valid samples are synchronized from the periodic fs count rollover to the next available SIB subframe
Above timing diagram shows fs count and valid sample flag generation for example configuration where frequency divisor = 6 (modulo-7 divide).

Figure 13.3.3 SIB Sample Rate Setup Timing

13.3.5 Enable/Disable Sequencing

As mentioned in the previous section, the synchronization of the sample rate counters within TMPR3911/12 and TC35143 is triggered from the codec enable command transmitted by TMPR3911/12 and received by TC35143. The following steps illustrate the full sequence of events needed to enable and disable the audio and/or telecom processing paths, while ensuring synchronization of these sample rate counters (the example below shows a configuration with TC35143 for sound and telecom, using subframe 0 only):

INITIAL ENABLE SEQUENCE:

(1) SIB is initially disabled (ENSIB = 0)

(2) setup SIB control registers to desired settings (SIB clock rates, sample rate dividers, data formats (8-bit vs. 16-bit, etc.), DMA settings, subframe 0 enable = ENSF0

(3) write initial data and control values to transmit holding registers (SNDTXHOLD, TELTXHOLD, SF0AUX); SF0AUX control register sets up TC35143 audio and telecom codecs, including sample rate dividers (to match TMPR3911/12 sample rates), gains, etc. codecs are not enabled at this point

(4) enable SIB (assert ENSIB = 1)

(5) SIB then begins transmitting and receiving data; TC35143 control registers get configured with desired settings; codecs still not enabled at this point

SND (or TEL) ENABLE SEQUENCE:

(6) setup software to trigger from SIBSF0INT

(7) immediately after SIBSF0INT trigger, software must write to SF0AUX to assert Audio (or Telecom) Codec Enable, and during same subframe (i.e., before SIBSF1INT event occurs), software must assert ENSND (or ENTEL)

(8) after next SIBSF0INT and during the subframe 0 period, the Codec Enable Command is transmitted from TMPR3911/12 and received by TC35143

(9) the Audio (or Telecom) Codec within TC35143 is then enabled at the start of the next frame after the Enable Command is sent, at which time the sample rate counters within both TMPR3911/12 and TC35143 begin counting in a synchronized fashion

(10) during this first frame after codec is enabled, the first DAC sample is transmitted by TMPR3911/12 (received by TC35143) and/or the first ADC sample is transmitted by TC35143 (received by TMPR3911/12); DAC and ADC samples are always transmitted during the same subframe in a full-duplex manner (as mentioned in the previous section, DAC and ADC samples are synchronized from the periodic sample rate counter rollover to the next available SIB subframe)

(11) check valid status from TC35143 (indicates whether ADC's are settled, all turn-on delays have been met, etc.) which indicates valid samples

SND (or TEL) DISABLE SEQUENCE:

(12) software writes to SF0AUX to de-assert Audio (or Telecom) Codec Enable, in order to disable codec

(13) software then de-asserts ENSND (or ENTEL) to disable sound (or telecom) processing on TMPR3911/12 side

13.3.6 Data Formats

The TMPR3911/12's SIB Module logic contains transmit and receive holding registers which are used by the DMA circuit or CPU to write and read sound and telecom data and SIB control and status register data. The SNDTXHOLD and SNDRXHOLD registers are used for transmit and receive sound data corresponding to the appropriate subframe (0 or 1), according to the configuration as determined by the SELSNDSF1 control bit. Similarly, the TELTXHOLD and TELRXHOLD registers are used for transmit and receive telecom data corresponding to the appropriate subframe (0 or 1), according to the configuration as determined by the SELTELSF1 control bit. All of these holding registers are 32 bits wide, and the appropriate shift register fields for each subframe are written to or read from based on the programmed sound and telecom samples rates, mono versus stereo mode, and 8-bit versus 16-bit modes.

The SF0AUX and SF1AUX control holding registers and SF0STAT and SF1STAT status holding registers are also 32 bits wide. Each of these control holding registers updates the control fields for the respective subframe once per frame. These transmit holding registers load the same control data into the shift register until the CPU updates the contents of the holding register. Similarly, the status holding registers are updated from the received status data in the shift register once per frame. The contents of the receive holding registers are available to be read by the CPU at any time.

The SIB data format for transfers to and from TC35143 is shown in Figure 13.3.4. Each word consists of fields containing audio data, telecom data, and control register address and data. During each frame, data is transferred bi-directionally (using SIBDIN and SIBDOUT) for all of these fields. For instance, the audio data field is written once every frame (from the TMPR3911/12 to TC35143) via SIBDOUT and also returns audio data (from TC35143 to the TMPR3911/12) at the same time via SIBDIN.

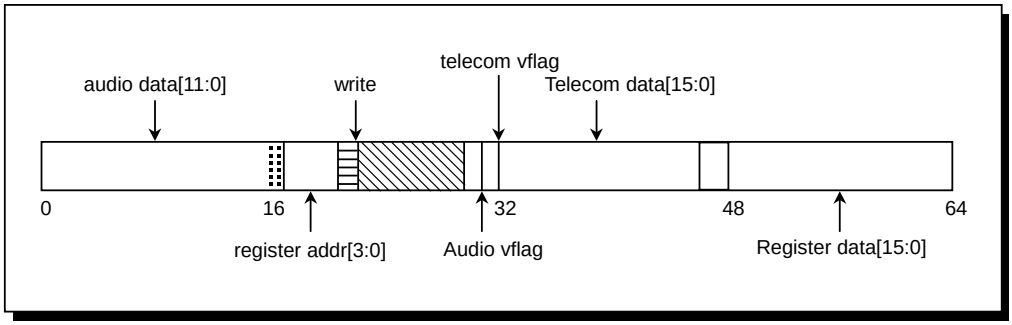


Figure 13.3.4 Data Format for TC35143 Word Interface

The SF0AUX control holding register and SF0STAT status holding register map directly to bits 16-31 and 48-63 of the subframe 0 word format. See Table 13.3.3 which shows the mapping of SF0AUX and SF0STAT bit positions to the respective TC35143 fields. For TC35143, the audio and telecom "vflag" bits (valid flags) in the transmit direction are dynamically asserted to TC35143 for one subframe each time a new sample is being transmitted. The audio and telecom "vflag" bits in the receive direction are statically asserted from TC35143 to the TMPR3911/12 only after the ADC's are settled, all turn-on delays have been met, etc. The TMPR3911/12 then uses these valid flags to determine when the receive data from TC35143 is truly valid.

Table 13.3.3 SIB Subframe 0 Control/Status Bits and Fields

SF0AUX/SF0STAT bits	TC35143 word bits	field definition
- - -	0-15	audio data
31	16	reserved for register extension; must write to 0 if not used
30-27	17-20	register address[3:0]
26	21	write bit
25-18	22-29	reserved, must write to 0
17	30	audio valid flag
16	31	telecom valid flag
- - -	32-47	telecom data
15-0	48-63	register data[15:0]

The last 16 bits of the TC35143 word is made up of control register data. The exact contents and definition of this field is defined by the register address field and the "write" bit. For a read cycle ("write" bit = 0), the address bits determine which TC35143 register to read and this read data is sent by TC35143 within the control register data field of SIBDIN during the same frame as the read request occurred. The separation between the address and data fields should make this possible. In addition, during a read cycle, the control register data field of SIBDOUT is ignored by TC35143. For a write cycle ("write" bit = 1), the control register data contents of SIBDOUT are written to the TC35143 register pointed to by the register address field.

13.3.7 Subframe Formats

SIB subframe 0 is nominally reserved for interfacing to TC35143. Four possible sample-size modes are supported for sound and telecom data (these are all mono sound and telecom formats):

sound	telecom
16-bit	16-bit
16-bit	8-bit
8-bit	16-bit
8-bit	8-bit

SIB subframe 1 is nominally reserved for interfacing to an external codec device.

For the external codec device, four possible sample-size modes are supported:

sound	telecom
16-bit stereo	16-bit
16-bit mono	8-bit
8-bit stereo	
8-bit mono	

Table 13.3.4 SIB Holding and Shift Register Formats (Subframe 0)

Subframe 0 Sound 16 Bit, Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
1	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]
2	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
3	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]

Subframe 0 Sound 16 Bit, Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:24],	8b0sf0aux[15:0]
1	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[23:16], 8b0	sf0aux[15:0]
2	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[15:8], 8b0	sf0aux[15:0]
3	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[7:0], 8b0	sf0aux[15:0]

Subframe 0 Sound 8 Bit, Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
1	sndtxhold[23:16], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]
2	sndtxhold[15:8], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
3	sndtxhold[7:0], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]

Subframe 0 Sound 8 Bit, Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:24], 8b0	sf0aux[15:0]
1	sndtxhold[23:16], 8b0	sf0aux[31:18], VS, VT	teltxhold[23:16], 8b0	sf0aux[15:0]
2	sndtxhold[15:8], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:8], 8b0	sf0aux[15:0]
3	sndtxhold[7:0], 8b0	sf0aux[31:18], VS, VT	teltxhold[7:0], 8b0	sf0aux[15:0]

NOTE: “VS” = valid sound flag bit; “VT” = valid telecom flag bit
“8b0” = 8-bit field of zero values
“SR” = Shift Register

Table 13.3.5 SIB Holding and Shift Register Formats (Subframe 1 Sound)

Subframe 1 Sound 16 Bit Stereo:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
1	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
2	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
3	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]

Subframe 1 Sound 16 Bit Mono:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[31:16]	sf1aux[15:0]
1	sndtxhold[15:0]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
2	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[31:16]	sf1aux[15:0]
3	sndtxhold[15:0]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]

Subframe 1 Sound 8 Bit Stereo:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
1	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]
2	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
3	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]

Subframe 1 Sound 8 Bit Mono:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[31:24], 8b0	sf1aux[15:0]
1	sndtxhold[23:16], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
2	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[15:8], 8b0	sf1aux[15:0]
3	sndtxhold[7:0], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]

NOTE: “8b0” = 8-bit field of zero values
“SR” = Shift Register

Table 13.3.6 SIB Holding and Shift Register Formats (Subframe 1 Telecom)

Subframe 1 Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	16b0	sf1aux[31:17], VT	teltxhold[31:16]	sf1aux[15:0]
1	16b0	sf1aux[31:17], VT	teltxhold[15:8]	sf1aux[15:0]
2	16b0	sf1aux[31:17], VT	teltxhold[31:16]	sf1aux[15:0]
3	16b0	sf1aux[31:17], VT	teltxhold[15:8]	sf1aux[15:0]

Subframe 1 Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	16b0	sf1aux[31:17], VT	teltxhold[31:24], 8b0	sf1aux[15:0]
1	16b0	sf1aux[31:17], VT	teltxhold[23:16], 8b0	sf1aux[15:0]
2	16b0	sf1aux[31:17], VT	teltxhold[15:8], 8b0	sf1aux[15:0]
3	16b0	sf1aux[31:17], VT	teltxhold[7:0], 8b0	sf1aux[15:0]

NOTE: “VS” = valid sound flag bit; “VT” = valid telecom flag bit
“8b0” = 8-bit field of zero values; “16b0” = 16-bit field of zero values
“SR” = Shift Register

13.4 DMA Address Generation

The SIB Module provides support for four independent DMA channels: sound receive and transmit and telecom receive and transmit. Two identical circuits (one for sound DMA and one for telecom DMA) are used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts (see Figure 13.4.1).

The DMA buffer size is programmable (up to a maximum of 16 Kbytes) and the receive and transmit buffer start addresses are also programmable (anywhere over the full 32-bit address space). Because there are separate start addresses, the receive and transmit buffers can be configured to either reside in different memory spaces or share the same memory space. The latter setup allows for overlapping buffers for loopback purposes or for optimum memory allocation, for which the DMA logic supports two full-duplex loopback modes. For one mode, receive DMA requests are issued first, followed by transmit DMA requests. This ordering allows a receive-to-transmit immediate loopback via the DMA buffer. For the second mode, transmit DMA requests are issued first, followed by receive DMA requests. Thus, received samples are written to the DMA buffer location immediately after transmit samples were read from that same location (which then became immediately available). This ordering allows a single circular DMA buffer to be used for both transmit and receive samples.

The DMA buffers can be configured in a circular buffer mode or a one-time buffer mode. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TMPR3911/12's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and utilize the DMA buffer in a ping-pong fashion. For transmit mode, the CPU can use these interrupts to fill or write one half of the buffer while the other half is being emptied by the DMA controller for transmitting out the SIB. Similarly, for receive mode, the CPU can use these interrupts to empty or read one half of the buffer while the other half is being filled by the DMA controller from received SIB input samples.

Also available is a direct CPU read/write mode for bypassing the DMA, allowing the CPU to read or write the sound or telecom data on a sample by sample basis, if so desired. Separate DMA enables for receive and transmit allow DMA to be setup for receive only (transmit via CPU), transmit only (receive via CPU), receive and transmit, or none (receive and transmit via CPU).

The sound and telecom DMA circuits also provide an interrupt each time the respective DMA buffer pointer is incremented, which occurs whenever a new sample is read from and/or written to the DMA buffer. This interrupt may be useful for triggering a read of the DMA pointer status value, which is the actual 12-bit DMA address counter output. This value indicates exactly where the current address is pointing to in the overall DMA buffer.

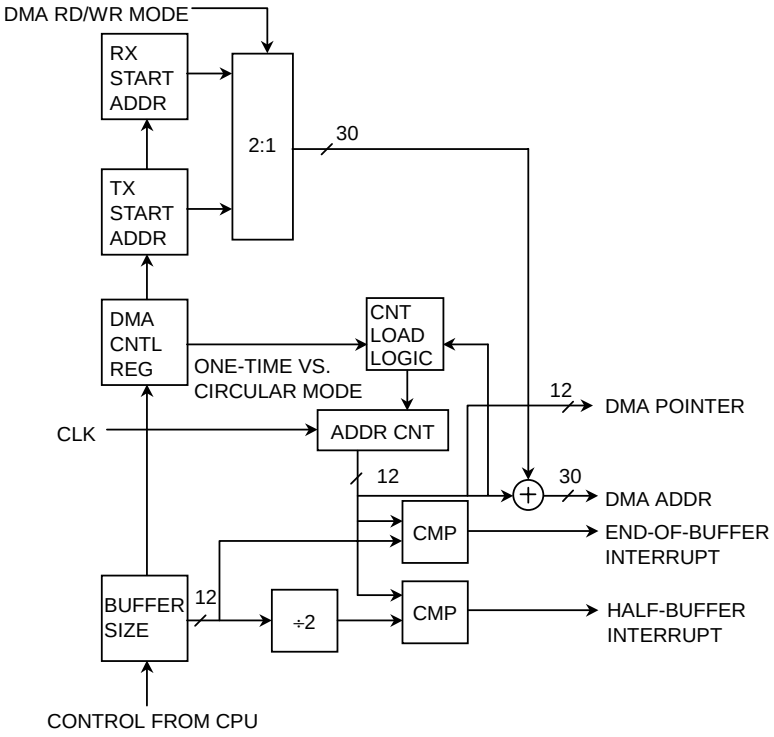


Figure 13.4.1 SIB DMA Address Generation

13.5 Related Interrupts

SND0_5INT:

Issues an interrupt whenever the sound DMA buffer pointer has reached the halfway point.

SND1_0INT:

Issues an interrupt whenever the sound DMA buffer pointer has reached the end-of-buffer point.

TEL0_5INT:

Issues an interrupt whenever the telecom DMA buffer pointer has reached the halfway point.

TEL1_0INT:

Issues an interrupt whenever the telecom DMA buffer pointer has reached the end-of-buffer point.

SNDDMACNTINT:

Issues an interrupt each time the sound DMA buffer pointer is incremented, which occurs whenever a new sound sample is read from and/or written to the sound DMA buffer.

TELDMACNTINT:

Issues an interrupt each time the telecom DMA buffer pointer is incremented, which occurs whenever a new telecom sample is read from and/or written to the telecom DMA buffer.

LSNDCLIPINT:

Issues an interrupt whenever the amplitude of the left channel sound data is clipping the codec A/D converter for SIB subframe 1.

RSNDCLIPINT:

Issues an interrupt whenever the amplitude of the right channel sound data is clipping the codec A/D converter for SIB subframe 1.

VALSNDPOSINT:

Issues an interrupt whenever the valid sound status flag transitions from a logic “0” to a logic “1”. This valid flag is triggered from SIB subframe 0 (if SELSNDSF1 = “0”) or from SIB subframe 1 (if SELSNDSF1 = “1”).

VALSNDNEGINT:

Issues an interrupt whenever the valid sound status flag transitions from a logic “1” to a logic “0”. This valid flag is triggered from SIB subframe 0 (if SELSNDSF1 = “0”) or from SIB subframe 1 (if SELSNDSF1 = “1”).

VALTELPOSINT:

Issues an interrupt whenever the valid telecom status flag transitions from a logic “0” to a logic “1”. This valid flag is triggered from SIB subframe 0 (if SELTELSF1 = “0”) or from SIB subframe 1 (if SELTELSF1 = “1”).

VALTELNEGINT:

Issues an interrupt whenever the valid telecom status flag transitions from a logic “1” to a logic “0”. This valid flag is triggered from SIB subframe 0 (if SELTELSF1 = “0”) or from SIB subframe 1 (if SELTELSF1 = “1”).

SNDININT:

Issues an interrupt whenever a valid sound input longword (32 bits) is available from the Sound RX Holding Register; this also means a valid sound output longword can be written to the Sound TX Holding Register.

TELININT:

Issues an interrupt whenever a valid telecom input longword (32 bits) is available from the Telecom RX Holding Register; this also means a valid telecom output longword can be written to the Telecom TX Holding Register.

SIBSF0INT:

Issues an interrupt at the start of every SIB subframe 0. This is used to initiate CPU reading of the SIB Subframe 1 Status Register (SF1STAT Register) and/or CPU writing of the SIB Subframe 0 Control Register (SF0AUX Register).

SIBSF1INT:

Issues an interrupt at the start of every SIB subframe 1. This is used to initiate CPU reading of the SIB Subframe 0 Status Register (SF0STAT Register) and/or CPU writing of the SIB Subframe 1 Control Register (SF1AUX Register).

SIBIRQPOSINT:

Issues an interrupt whenever the SIBIRQ pin transitions from a logic “0” to a logic “1”.

SIBIRQNEGINT:

Issues an interrupt whenever the SIBIRQ pin transitions from a logic “1” to a logic “0”.

13.6 SIB Registers

13.6.1 SIB Size Register

OFFSET = $060: write-only

Bit	Label	RESET	Read/Write
31-30	Reserved		
29-18	SNDSIZE[13:2]	X	W
17-14	Reserved		
13-2	TELSIZE[13:2]	X	W
1-0	Reserved		

SNDSIZE[13:2]: write-only

These bits define the size of the sound DMA buffers (16 Kbytes maximum). Both the sound RX buffer and the sound TX buffer are the same size. The last address in the sound RX DMA buffer is given by SNDRXSTART[31:2] + SNDSIZE[13:2]. The last address in the sound TX DMA buffer is given by SNDTXSTART[31:2] + SNDSIZE[13:2]. The value loaded into SNDSIZE should be equal to the desired buffer length –1.

Note that this register is write only.

TELSIZE[13:2]: write-only

These bits define the size of the telecom DMA buffers (16 Kbytes maximum). Both the telecom RX buffer and the telecom TX buffer are the same size. The last address in the telecom RX DMA buffer is given by TELRXSTART[31:2] + TELSIZE[13:2]. The last address in the telecom TX DMA buffer is given by TELTXSTART[31:2] + TELSIZE[13:2]. The value loaded into TELSIZE should be equal to the desired buffer length –1.

Note that this register is write only.

13.6.2 SIB Sound RX Start Register

OFFSET = $064: write-only

Bit	Label	RESET	Read/Write
31-2	SNDRXSTART[31:2]	X	W
1-0	Reserved		

SNDRXSTART[31:2]: write-only

These bits define the start address for the sound RX DMA buffer. The sound RX buffer and sound TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

13.6.3 SIB Sound TX Start Register

OFFSET = $068: write-only

Bit	Label	RESET	Read/Write
31-2	SNDTXSTART[31:2]	X	W
1-0	Reserved		

SNDTXSTART[31:2]: write-only

These bits define the start address for the sound TX DMA buffer. The sound RX buffer and sound TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

13.6.4 SIB Telecom RX Start Register

OFFSET = $06C: write-only

Bit	Label	RESET	Read/Write
31-2	TELRXSTART[31:2]	X	W
1-0	Reserved		

TELRXSTART[31:2]: write-only

These bits define the start address for the telecom RX DMA buffer. The telecom RX buffer and telecom TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

13.6.5 SIB Telecom TX Start Register

OFFSET = $070: write-only

Bit	Label	RESET	Read/Write
31-2	TELTXSTART[31:2]	X	W
1-0	Reserved		

TELTXSTART[31:2]: write-only

These bits define the start address for the telecom TX DMA buffer. The telecom RX buffer and telecom TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).

13.6.6 SIB Control Register

OFFSET = $074:

Bit	Label	RESET	Read/Write
31	SIBIRQ	–	R
30	ENCNTTEST	0	R/W
29	ENDMATEST	0	R/W
28	SNDMONO	X	R/W
27	RMONOSNDIN	X	R/W
26-24	SIBSCLKDIV[2:0]	X	R/W
23	TEL16	X	R/W
22-16	TELFSDIV[6:0]	X	R/W
15	SND16	X	R/W
14-8	SNDFSDIV[6:0]	X	R/W
7	SELTELSF1	X	R/W
6	SELSNDSF1	X	R/W
5	ENTEL	0	R/W
4	ENSND	0	R/W
3	SIBLOOP	0	R/W
2	ENSF1	0	R/W
1	ENSF0	0	R/W
0	ENSIB	0	R/W

SIBIRQ: read-only

This bit provides the logic state of the SIBIRQ input pin. The SIBIRQ pin is active-high from TC35143.

ENCNTTEST:

This bit is used for IC testing and should not be set.

ENDMATEST:

This bit is used for IC testing and should not be set.

SNDMONO:

This bit is used to configure the sound input and output format for SIB subframe 1 as mono versus stereo. Setting this bit to a logic “1” selects mono mode. Clearing this bit to a logic “0” selects stereo mode.

RMONOSNDIN:

This bit is used to select left versus right channel as the mono sound source for SIB subframe 1, whenever the sound is configured as mono mode. Setting this bit to a logic “1” selects the right channel as the mono sound source. Clearing this bit to a logic “0” selects the left channel as the mono sound source.

SIBSCLKDIV[2:0]:

These bits select the start count value and the stop count value for the 4-bit programmable counter used to generate SIBSCLK, which is derived by dividing down SIBMCLK; SIBSCLK is the serial bit clock used for the SIB. Since the MSB of the counter output is used for SIBSCLK, the start count and stop count values are chosen to provide (as close as possible) a 50% duty cycle SIBSCLK. The table used to compute these counter start and stop values are as follows:

SIBSCLKDIV	start value	stop value	divide-modulus
0	7	8	2
1	6	8	3
2	6	9	4
3	5	9	5
4	5	10	6
5	4	11	8
6	3	12	10
7	2	13	12

TEL16:

This bit is used to configure the SIB telecom input and output format as 8-bit versus 16-bit mode. Setting this bit to a logic “1” selects 16-bit mode. Clearing this bit to a logic “0” selects 8-bit mode.

TELFSDIV[6:0]:

These bits select the divider modulus for the 7-bit programmable counter used to generate the telecom sample rate clock, which is derived by dividing down an internal SIB clock of rate equal to the 4 times the SIB frame rate or 2 times the SIB subframe rate (where each frame consists of 128 serial data bits). This programmable divider consists of a down-counter which counts down from TELFSDIV to zero, so the value loaded for TELFSDIV should be the desired divider modulus –1.

For example, if SIBMCLK = 18.432 MHz and SIBSCLK = 9.216 MHz (= SIBMCLK ÷ 2), in order to generate a telecom sample rate of 7.2 kHz, the value loaded for TELFSDIV should = 39, since 9.216 MHz ÷ (40 × 32) = 7.2 kHz.

SND16:

This bit is used to configure the SIB sound input and output format as 8-bit versus 16-bit mode. Setting this bit to a logic “1” selects 16-bit mode. Clearing this bit to a logic “0” selects 8-bit mode.

SNDFSDIV[6:0]:

These bits select the divider modulus for the 7-bit programmable counter used to generate the sound sample rate clock, which is derived by dividing down an internal SIB clock of rate equal to the 4 times the SIB frame rate or 2 times the SIB subframe rate (where each frame consists of 128 serial data bits). This programmable divider consists of a down-counter which counts down from SNDFSDIV to zero, so the value loaded for SNDFSDIV should be the desired divider modulus −1.

For example, if SIBMCLK = 18.432 MHz and SIBSCLK = 9.216 MHz (= SIBMCLK ÷ 2), in order to generate a sound sample rate of 24 kHz, the value loaded for SNDFSDIV should = 11, since 9.216 MHz ÷ (12 × 32) = 24 kHz.

SELTELSF1:

This bit is used to select between SIB subframe 0 and subframe 1 for the telecom data source and destination. Setting this bit to a logic “1” selects SIB subframe 1 as the telecom source. Clearing this bit to a logic “0” selects SIB subframe 0 as the telecom source.

SELSNDSF1:

This bit is used to select between SIB subframe 0 and subframe 1 for the sound data source and destination. Setting this bit to a logic “1” selects SIB subframe 1 as the sound source. Clearing this bit to a logic “0” selects SIB subframe 0 as the sound source.

ENTEL:

This bit is used to enable/disable telecom processing for the SIB module. Setting this bit to a logic “1” enables telecom processing. Clearing this bit to a logic “0” disables telecom processing. This bit should not be set until after the SIB module is setup, then ENSIB asserted.

Special timing restrictions are required whenever an enable/disable command is sent to TC35143 and/or other external codec device (see Figure 13.3.3). First, after SIBSF0INT (if SELTELSF1 = “0”) or SIBSF1INT (if SELTELSF1 = “1”) is asserted, the CPU then writes a telecom codec enable command to TC35143 via the SF0AUX (if SELTELSF1 = “0”) or SF1AUX (if SELTELSF1 = “1”) register. The CPU then asserts ENTEL. Both of these CPU transactions must occur within the same SIB frame period, which is before the next SIBSF0INT (if SELTELSF1 = “0”) or SIBSF1INT (if SELTELSF1 = “0”). This ensures that the sample rate counters within the TMPR3911/12 and TC35143 are fully phase-synchronized.

ENSND:

This bit is used to enable/disable sound processing for this SIB module. Setting this bit to a logic "1" enables sound processing. Clearing this bit to a logic "0" disables sound processing. This bit should not be set until after the SIB module is setup, then ENSIB asserted.

Special timing restrictions are required whenever an enable/disable command is sent to TC35143 and/or other external codec device (see Figure 13.3.3). First, after SIBSF0INT (if SELSNDSF1 = "0") or SIBSF1INT (if SELSNDSF1 = "1") is asserted, the CPU then writes a sound codec enable command to TC35143 via the SF0AUX (if SELSNDSF1 = "0") or SF1AUX (if SELSNDSF1 = "1") register. The CPU then asserts ENSND. Both of these CPU transactions must occur within the same SIB frame period, which is before the next SIBSF0INT (if SELSNDSF1 = "0") or SIBSF1INT (if SELSNDSF1 = "0"). This ensures that the sample rate counters within the TMPR3911/12 and TC35143 are fully phase-synchronized.

SIBLOOP:

This bit is used for IC testing and should not be set. Setting this bit to a logic "1" will cause the SIB serial transmitted data to be internally looped back to the SIB serial receive data path. The data is inverted when this mode is selected. Setting this bit to a logic "0" selects the normal SIBDIN pin as the SIB serial receive data source.

ENSF1:

This bit is used to enable/disable SIB subframe 1 processing. Setting this bit to a logic "1" enables SIB subframe 1. Clearing this bit to a logic "0" disables SIB subframe 1, causing the SIB serial transmitted data during this subframe to be zeroed and all received data during this subframe to not be processed by the SIB module. This bit should be set before ENSIB is asserted.

ENSF0:

This bit is used to enable/disable SIB subframe 0 processing. Setting this bit to a logic "1" enables SIB subframe 0. Clearing this bit to a logic "0" disables SIB subframe 0, causing the SIB serial transmitted data during this subframe to be zeroed and all received data during this subframe to not be processed by the SIB module. This bit should be set before ENSIB is asserted.

ENSIB:

This bit is used to enable/disable the SIB module. Setting this bit to a logic "1" enables the SIB module. Clearing this bit to a logic "0" disables the SIB module and keeps the module in a reset state but gives no effect on the SIB Control Register.

13.6.7 SIB Sound TX Holding Register

OFFSET = $078: write-only

Bit	Label	RESET	Read/Write
31-0	SNDTXHOLD[31:0]	X	W

SNDTXHOLD[31:0]: write-only

These bits represent the sound data to be transmitted. Sound data can be either written directly to this register by the CPU or transparently read from the sound TX DMA buffer to this register. This register should only be loaded by the CPU after the SNDININT interrupt is asserted. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 13.3.4 ~ 13.3.3 for a summary of these formats.

13.6.8 SIB Sound RX Holding Register

OFFSET = $078: read-only

Bit	Label	RESET	Read/Write
31-0	SNDRXHOLD[31:0]	–	R

SNDRXHOLD[31:0]: read-only

These bits represent the sound data to be received. Sound data can be either read directly from this register by the CPU or transparently written to the sound RX DMA buffer from this register. This register should only be read by the CPU after the SNDININT interrupt is set. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 13.3.4 ~ 13.3.3 for a summary of these formats.

13.6.9 SIB Telecom TX Holding Register

OFFSET = $07C: write-only

Bit	Label	RESET	Read/Write
31-0	TELTXHOLD[31:0]	X	W

TELTXHOLD[31:0]: write-only

These bits represent the telecom data to be transmitted. Telecom data can be either written directly to this register by the CPU or transparently read from the telecom TX DMA buffer to this register. This register should only be loaded by the CPU after the TELININT interrupt is asserted. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 13.3.4~13.3.7 for a summary of these formats.

13.6.10 SIB Telecom RX Holding Register

OFFSET = $07C: read-only

Bit	Label	RESET	Read/Write
31-0	TELRXHOLD[31:0]	–	R

TELRXHOLD[31:0]: read-only

These bits represent the telecom data to be received. Telecom data can be either read directly from this register by the CPU or transparently written to the telecom RX DMA buffer from this register. This register should only be read by the CPU after the TELININT interrupt is set. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 13.3.4~13.3.7 for a summary of these formats.

13.6.11 SIB Subframe 0 Control Register

OFFSET = $080:

Bit	Label	RESET	Read/Write
31-0	SF0AUX[31:0]	X	R/W

SF0AUX[31:0]:

These bits represent the control data to be transmitted during SIB subframe 0. This register can be loaded by the CPU asynchronously with respect to the SIB frame timing (although special timing restrictions are required whenever an enable/disable command is sent to TC35143 and/or other external codec device). Whenever this register is not updated, the previous contents are transmitted during consecutive SIB frames.

13.6.12 SIB Subframe 1 Control Register

OFFSET = $084:

Bit	Label	RESET	Read/Write
31-0	SF1AUX[31:0]	X	R/W

SF1AUX[31:0]:

These bits represent the control data to be transmitted during SIB subframe 1. This register can be loaded by the CPU asynchronously with respect to the SIB frame timing (although special timing restrictions are required whenever an enable/disable command is sent to TC35143 and/or other external codec device). Whenever this register is not updated, the previous contents are transmitted during consecutive SIB frames.

13.6.13 SIB Subframe 0 Status Register

OFFSET = $088: read-only

Bit	Label	RESET	Read/Write
31-0	SF0STAT[31:0]	–	R

SF0STAT[31:0]: read-only

These bits represent the status data to be read during SIB subframe 0. This register can be read by the CPU asynchronously with respect to the SIB frame timing. Whenever this register is not updated from TC35143 and/or other external codec device, the previous contents are transmitted during consecutive SIB frames.

13.6.14 SIB Subframe 1 Status Register

OFFSET = $08C: read-only

Bit	Label	RESET	Read/Write
31-0	SF1STAT[31:0]	–	R

SF1STAT[31:0]: read-only

These bits represent the status data to be read during SIB subframe 1. This register can be read by the CPU asynchronously with respect to the SIB frame timing. Whenever this register is not updated from TC35143 and/or other external codec device, the previous contents are transmitted during consecutive SIB frames.

13.6.15 SIB DMA Control Register

OFFSET = $090:

Bit	Label	RESET	Read/Write
31	SNDBUFF1TIME	0	R/W
30	SNDDMALOOP	0	R/W
29-18	SNDDMAPTR[13:2]	–	R
17	ENDMARXSND	0	R/W
16	ENDMATXSND	0	R/W
15	TELBUFF1TIME	0	R/W
14	TELDMALOOP	0	R/W
13-2	TELDMALOOP[13:2]	–	R
1	ENDMARXTEL	0	R/W
0	ENDMATXTEL	0	R/W

SNDBUFF1TIME:

The sound DMA controller supports two buffer addressing modes depending on the state of this bit. When SNDBUFF1TIME is set to a logic “1”, the sound DMA controller will stop executing when it reaches the end of the DMA buffer. When SNDBUFF1TIME is cleared to a logic “0”, the sound DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner.

SNDDMALOOP:

The sound DMA controller supports two full-duplex loopback modes depending on the state of this bit. When SNDDMALOOP is set to a logic “1”, the sound DMA controller issues RX DMA requests first, followed by TX DMA requests. This ordering allows an RX-to-TX immediate loopback via the DMA buffer. When SNDDMALOOP is cleared to a logic “0”, the sound DMA controller issues TX DMA requests first, followed by RX DMA requests. This ordering allows a single circular DMA buffer to be used for both TX and RX, if so desired.

SNDDMAPTR[13:2]: read-only

These bits provide the status of the sound DMA counter.

ENDMARXSND:

This bit enables the sound DMA receive function. Setting this bit to a logic “1” enables the DMA mode. Clearing this bit to a logic “0” disables the DMA mode. This bit should not be set until the SNDRXSTART, SNDTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted) and ENSND is set. Either ENDMARXSND or ENDMATXSND or both can be set at a time since the sound DMA controller can support full duplex operation.

ENDMATXSND:

This bit enables the sound DMA transmit function. Setting this bit to a logic “1” enables the DMA mode. Clearing this bit to a logic “0” disables the DMA mode. This bit should not be set until the SNDRXSTART, SNDTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted) and ENSND is set. Either ENDMARXSND or ENDMATXSND or both can be set at a time since the sound DMA controller can support full duplex operation.

TELBUFF1TIME:

The telecom DMA controller supports two buffer addressing modes depending on the state of this bit. When TELBUFF1TIME is set to a logic “1”, the telecom DMA controller will stop executing when it reaches the end of the DMA buffer. When TELBUFF1TIME is cleared to a logic “0”, the telecom DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner.

TELDMALOOP:

The telecom DMA controller supports two full-duplex loopback modes depending on the state of this bit. When TELDMALOOP is set to a logic “1”, the telecom DMA controller issues RX DMA requests first, followed by TX DMA requests. This ordering allows an RX-to-TX immediate loopback via the DMA buffer. When TELDMALOOP is cleared to a logic “0”, the telecom DMA controller issues TX DMA requests first, followed by RX DMA requests. This ordering allows a single circular DMA buffer to be used for both TX and RX, if so desired.

TELDMAPTR[13:2]: read-only

These bits provide the status of the telecom DMA counter.

ENDMARXTEL:

This bit enables the telecom DMA receive function. Setting this bit to a logic “1” enables the DMA mode. Clearing this bit to a logic “0” disables the DMA mode. This bit should not be set until the TELRXSTART, TELTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted) and ENTEL is set. Either ENDMARXTEL or ENDMATXTEL or both can be set at a time since the telecom DMA controller can support full duplex operation.

ENDMATXTEL:

This bit enables the telecom DMA transmit function. Setting this bit to a logic “1” enables the DMA mode. Clearing this bit to a logic “0” disables the DMA mode. This bit should not be set until the TELRXSTART, TELTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted) and ENTEL is set. Either ENDMARXTEL or ENDMATXTEL or both can be set at a time since the telecom DMA controller can support full duplex operation.

14. SPI Module

14.1 Overview

The SPI is a serial interface consisting of clock, data out, and data in. The SPI is used to interface to devices such as serial power supplies, serial A/D converters, and other devices that contain simple serial clock and data interfaces. The TMPR3911/12 only supports master mode and generates the SPI clock to the slaves. Multiple slave devices can share the SPI by using a unique chip select for each slave device. The chip select can be generated using one of the general purpose I/O ports on the TMPR3911/12 or TC35143, or using some other output port available in the system. When a device is selected by asserting it's chip select, the device will shift data in using the SPICLK and SPIOUT signals and the device will shift data out using the SPIIN signal. When a device is not selected then the data output connected to SPIIN must be tri-stated so other devices can share the SPIIN signal. The SPI Module contains registers which provide programmability for SPICLK rate, MSB first versus LSB first, clock polarity, data phase polarity, and byte mode versus word mode operation.

14.1.1 Related Pins

SPICLK: OUTPUT

This pin is used to clock data in and out of the slave device.

SPIOUT: OUTPUT

This pin contains the data that is shifted into the slave device.

SPIIN: INPUT

This pin contains the data that is shifted out of the slave device.

14.2 Description

14.2.1 Block Diagram

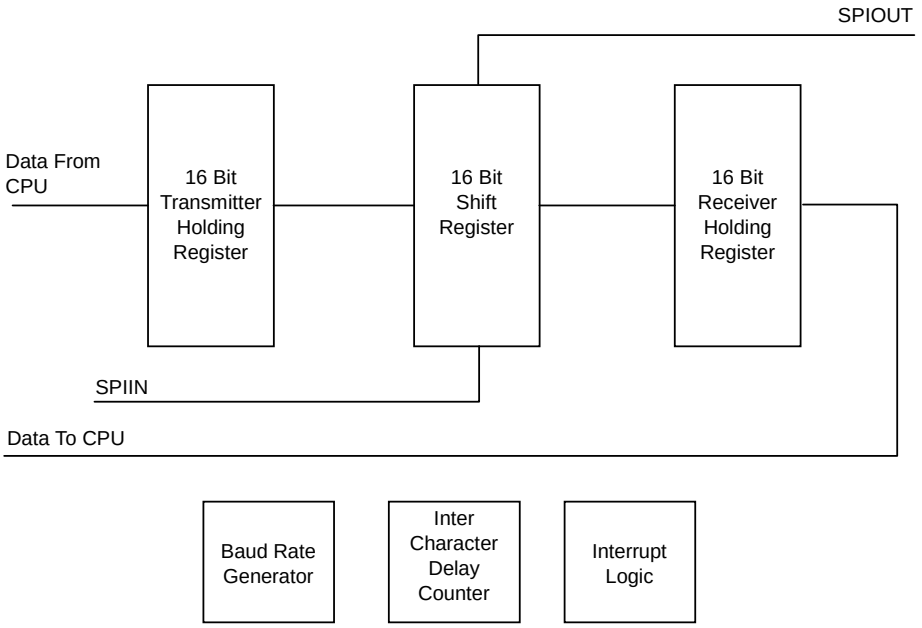


Figure 14.2.1 SPI Block Diagram

The SPI Module primarily consists of a 16-bit Shift Register, a 16-bit Transmitter Holding Register, a 16-bit Receiver Holding Register, a Baud Rate Generator, an Inter Character Delay Counter, and Interrupt Logic. A block diagram of the SPI Module is shown in Figure 14.2.1.

14.2.2 Baud Rate Generator

The rate of the SPICLK signal is determined by the value of the BAUDRATE[3:0] bits in the SPI Control Register. The BAUDRATE[3:0] bits are used by the Baud Rate Generator to divide the SPI master clock generated by the Clock Module logic. The typical frequency of the SPI master clock is shown in the table below.

Table 14.2.1

	SPI Master clock frequency ($f_{SPIMCLK}$)	Main System clock frequency	Comment
TMPR3912AU-92/XB-92	9.216 MHz	46.080 MHz	f_{SYSCLK} = 11.520 MHz, RF[1:0] = 00, SLOWBUS = 0
TMPR3911BU/BXB	5.8982 MHz	29.4912 MHz	f_{SYSCLK} = 7.3728 MHz, RF[1:0] = 00, SLOWBUS = 0

14.2.3 Transmitter/Receiver

The SPI Module is kept in a reset state until the ENSPI bit is set in the SPI Control Register. Before setting the ENSPI bit, all other control bits in the SPI Control Register should be set to the desired values. Once the ENSPI bit is set, the SPIBUFAVAILINT interrupt will be asserted to indicate that the SPI Transmitter Holding Register is available. The SPI logic will then wait until the software writes to the SPI Transmitter Holding Register.

Once the software writes to the Transmitter Holding Register then the contents will be transferred to the Shift Register and shifted out to the slave device. While data is shifting out to the slave device using the SPIOUT signal, data will shift in using the SPIIN signal. Once the data has finished shifting, the contents of the Shift Register will be loaded into the Receiver Holding Register and the SPIRCVINT Interrupt will be asserted to indicate that there is valid receive data in the Receiver Holding Register. Once the contents of the Transmitter Holding Register are transferred to the Shift Register, the SPIBUFAVAILINT interrupt is again asserted to indicate that the Transmitter Holding register is once again available. Therefore the software is supposed to do the following procedure every time it attempts to write data into the Transmitter Holding Register.

1) Check if SPIBUFAVAILINT is a logic “1”. If not, wait for its assertion.
2) Clear the SPIBUFAVAILINT status bit.
3) Write data into the Transmitter Holding Register.

Thus, as long as the software can keep the Transmitter Holding Register serviced before the data shifts out of the Shift Register, the SPI can maintain seamless data transfer. If the software fails to keep up with the transfer rate, then the SPI will simply wait until the next data is written to the Transmitter Holding Register.

At the end of every series of transmission the software is supposed to negate the chip select signal for the target device by the following procedure.

1) Check if SPIBUFAVAILINT is a logic “1”. If not, wait for its assertion.
2) Leave the SPIBUFAVAILINT status bit set (do not clear the bit).
3) Clear the SPIEMPTYINT status bit.
4) Wait for SPIEMPTYINT assertion.
5) Negate the chip select signal.

The SPI supports either 8-bit per character or 16-bit per character operation, as defined by the WORD bit in the SPI Control Register. The software can also select whether the MSB or LSB should shift first using the LSB control bit in the SPI Control Register. Another set of control bits (CLKPOL and PHAPOL) select the polarity of the SPICLK and determine whether data should be sampled on the rising or falling edge of the SPICLK. SPI timing is shown in Figure 14.2.2.

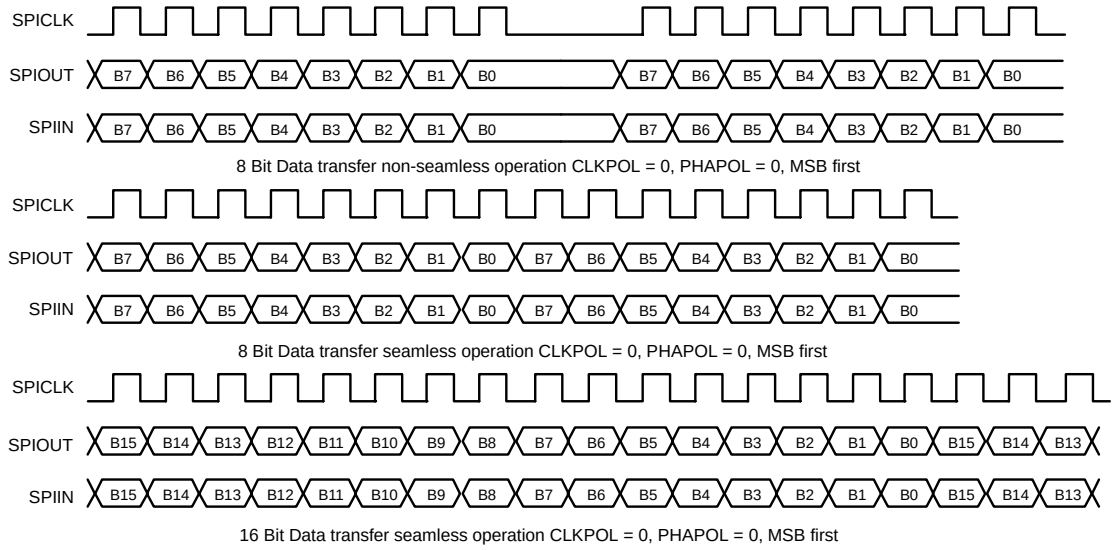


Figure 14.2.2 SPI Timing

14.2.4 CLKPOL/PHAPOL

The CLKPOL and PHAPOL bits in the SPI Control Register determine the idle phase of SPICLK and the valid clock edge for sampling data. Figure 14.2.3 shows the four possible combinations.

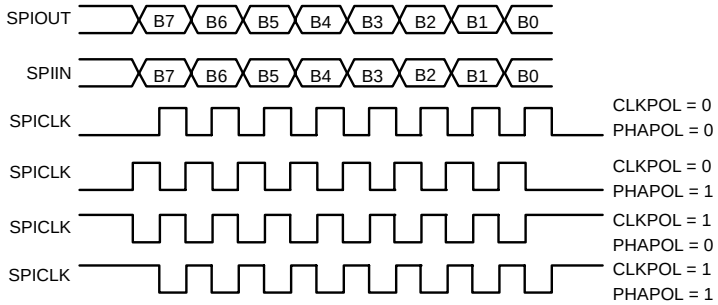


Figure 14.2.3 CLKPOL/PHAPOL Timing

14.2.5 Inter Character Delay Counter

Sometimes it is desirable to guarantee a minimum time between groups of data. The Inter Character Delay Counter is used to provide delay between groups of data. If WORD mode is selected in the SPI Control Register, delay will be inserted after 16 bits of data are shifted. If WORD mode is not selected, delay will be inserted after 8 bits of data are shifted, as shown in Figure 14.2.4. Inter character delay is added by setting the DELAYVAL[3:0] bits to a value other than $0. The number stored in these bits will directly correspond to the number of SPICLK periods of delay that will be inserted between characters. A zero value for these bits will imply seamless operation and the SPI will shift data and provide clocks continuously as long as the software keeps up with the transmitter rate.

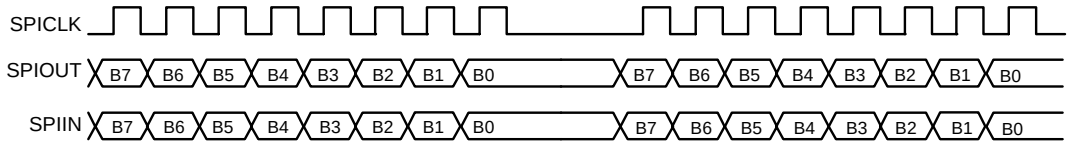


Figure 14.2.4 Inter Character Delay Timing

14.2.6 SPI Interrupts

SPIBUFAVAILINT:

This interrupt is set when the ENSPI bit is first asserted and subsequently when the contents of the SPI Transmitter Holding Register are transferred to the SPI Shift Register. This interrupt is used to indicate that the SPI Transmitter Holding Register is available to be written by the software. See Section 8 for information on how to read and clear interrupts.

SPIERRINT:

This interrupt is set whenever the SPI Transmitter Holding Register is written, but the SPIBUFAVAILINT has not set to indicate that the register is available. This interrupt serves as a overrun indication for the software. See Section 8 for information on how to read and clear interrupts.

SPIRCVINT:

This interrupt is whenever the contents of the SPI Shift Register are transferred to the SPI Receiver Holding Register. This interrupt is used to indicate that there is valid data in the SPI Receiver Holding Register to be read by the software. See Section 8 for information on how to read and clear interrupts.

SPIEMPTYINT:

This interrupt is set whenever the both the SPI Shift Register and the SPI Transmitter Holding Register are empty. This interrupt can be used by the software to determine when the SPI is idle. See Section 8 for information on how to read and clear interrupts.

14.3 SPI Registers

14.3.1 SPI Control Register

OFFSET = $160:

Bit	Label	RESET	Read/Write
31-18	Reserved		
17	SPION	0	R
16	EMPTY	1	R
15-12	DELAYVAL[3:0]	X	R/W
11-8	BAUDRATE[3:0]	X	R/W
7-6	Reserved		
5	PHAPOL	0	R/W
4	CLKPOL	0	R/W
3	Reserved		
2	WORD	0	R/W
1	LSB	0	R/W
0	ENSPI	0	R/W

SPION: read-only

When the ENSPI bit is disabled, the SPI will not shut down until the Transmitter Holding Register and Shift Register are both empty, in order to make sure that any data still in the SPI is shifted out. This status bit allows the software to know when the module has shut down as a result of clearing the ENSPI control bit.

EMPTY: read-only

This bit is asserted if both the Transmitter Holding Register and Shift Register are empty and the Inter Character Delay Counter is finished inserting delay. This status bit allows the software to know when the SPI is idle.

DELAYVAL[3:0]:

These bits define the number SPICLK periods of delay to insert between 16-bit or 8-bit characters, depending on whether WORD mode is selected or not. The value of these bits directly corresponds to the number of SPICLK periods of delay. Thus, a value of $0 will provide seamless operation with no inter character delay.

BAUDRATE[3:0]:

These bits define the rate of the SPICLK. These bits divide the SPI master clock to generate the desired SPICLK rate, as given by the following equation ($f_{SPIMCLK}$ is the frequency of the SPI master clock.):

$$\text{SPICLK Rate} = \frac{f_{SPIMCLK}}{BAUDRATE[3:0]*2+2}$$

PHAPOL:

Setting this bit will cause the transmitter to clock data out on the rising edge of SPICLK, to be sampled by the peripherals on the falling edge of SPICLK.

CLKPOL:

Setting this bit will cause the SPICLK signal to idle high instead of low.

WORD:

Setting this bit will cause the SPI to shift 16 bits of data in and out. If this bit is cleared, 8 bits of data are shifted.

LSB:

Setting this bit will cause the data to be shifted out to the slave devices and in from the slave devices LSB-first instead of MSB-first.

ENSPI:

Setting this bit will enable the SPI Module. This bit should only be set after all other control bits have been set to the desired values. When this bit is cleared, the SPI will remain active until the Shift Register and Transmitter Holding Register are both empty. Once both registers are empty, the SPI will shut down and all the logic will be held in a reset state but gives no effect on the SPI Control Register.

14.3.2 SPI Transmitter Holding Register

OFFSET = $164: write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	TXDATA[15:0]	X	W

TXDATA[15:0]: write-only

These bits are the data that is loaded into the Transmitter Holding Register. This register should only be loaded after the SPIBUFAVAILINT interrupt is asserted. If WORD mode is not set, only bits 7:0 are valid.

14.3.3 SPI Receiver Holding Register

OFFSET = $164: read-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	RXDATA[15:0]	X	R

RXDATA[15:0]: read-only

These bits are the receive data in the Receiver Holding Register. The bits are only valid after the SPIRCVINT interrupt is asserted. If WORD mode is not set, only bits 7:0 are valid.

15. Timer Module

15.1 Overview

The Timer Module consists of two portions. The first is a 40-bit Real Time Clock (RTC) counter that uses a 32.768 kHz clock. The counter will provide a maximum count of 388 days. Also included is a 40-bit alarm register for the RTC that allows the software to set an alarm at any desired count of the RTC counter. The RTC will generate two interrupts for the CPU. The first is the ALARMINT that will generate an interrupt whenever the RTC reaches the value set by the alarm. The second is the RTCINT that will generate an interrupt whenever the RTC counter "rolls over" after reaching a count of 388 days.

The second portion of the Timer Module is the Periodic Timer used by the software to generate periodic interrupts for monitoring system events. The Periodic Timer uses the TIMERCLK generated by the Clock Module, which is generated by dividing the CLK by 32. The Periodic Timer contains a programmable 16-bit counter. When enabled, the counter will count down and generate an interrupt (PERINT) whenever reaching a count of zero.

15.1.1 Related Pins

C32KIN: INPUT

This pin along with C32KOUT should be connected to a 32.768 kHz crystal.

C32KOUT: OUTPUT

This pin along with C32KIN should be connected to a 32.768 kHz crystal.

BC32K: OUTPUT

This pin is a buffered output of the 32.768 kHz clock.

15.2 RTC

15.2.1 RTC Block Diagram

The RTC consists of five 8-bit ripple counters, a 40-bit equality comparator, roll over detect logic and two interrupt flip-flops, as shown in Figure 15.2.1.

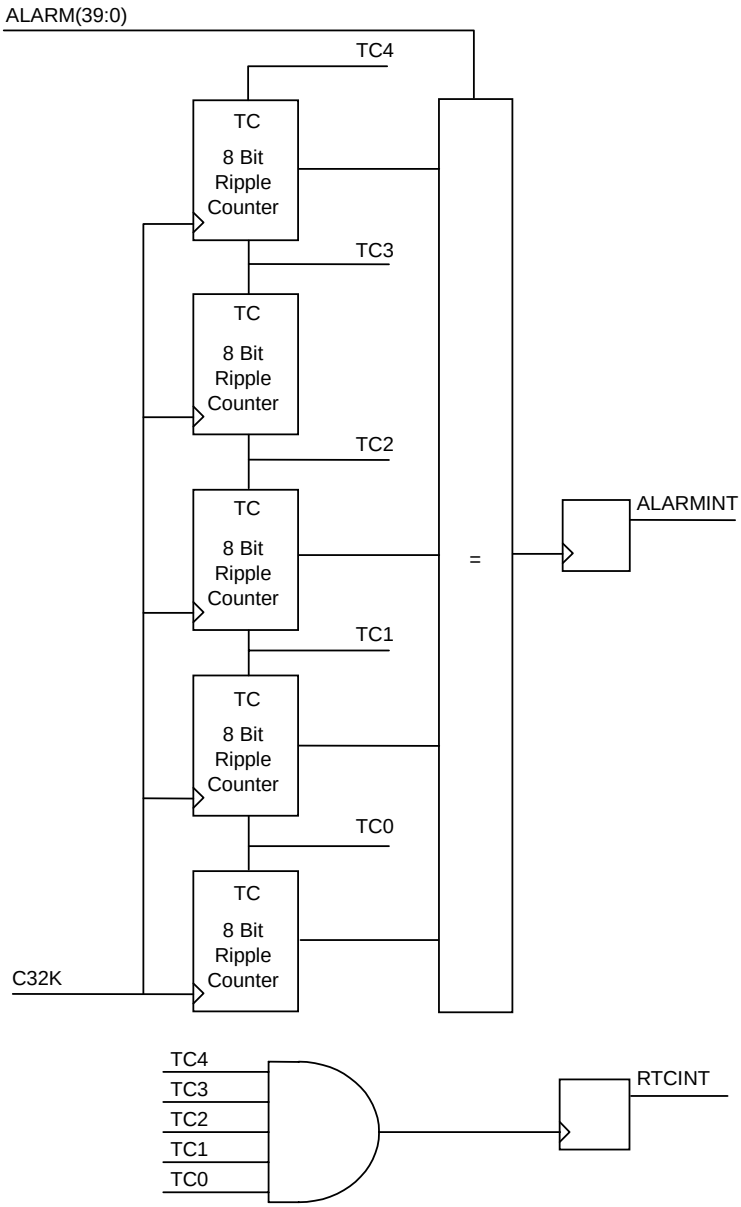


Figure 15.2.1 RTC Block Diagram

15.2.2 RTC Description

The RTC contains five 8-bit ripple counters connected in series. The first counter counts on each C32K clock, while each successive counter only counts when the previous count stage has reached a count of “$FF”. Once all the counters reach a count of “$FFFFFFFFFF”, the RTCINT interrupt will assert to indicate that the counter is “rolling over”. Given a 40-bit counter for the RTC and an input clock C32K of 32.768 kHz, the time until the RTCINT interrupt will assert is 388 days.

The software can generate an alarm interrupt (ALARMINT) by setting the ALARM[39:0] bits in the Alarm Register. Whenever the RTC becomes equal to the value set in the Alarm Register, the ALARMINT will be triggered. The value of the RTC counter can be read via the RTC Register. The RTC counter is split into groups of five 8-bit counters to simplify IC testing.

15.2.3 RTC Interrupts

RTCINT:

This interrupt is set whenever all 40 bits of the RTC counter reach a value of “$FFFFFFFFFF” to alert the software that the counter is “rolling over”. See Section 8 for information on how to read and clear interrupts.

ALARMINT:

This interrupt is set whenever the RTC counter reaches a count that is equal to the value of the ALARM[39:0] bits set in the Alarm Register. See Section 8 for information on how to read and clear interrupts.

15.3 Periodic Timer

15.3.1 Periodic Timer Block Diagram

The Periodic Timer consists of a 16-bit down counter, along with a zero detect and an interrupt flip-flop, as shown in Figure 15.3.1.

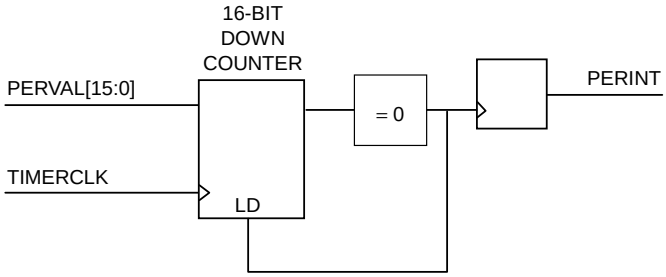


Figure 15.3.1 Periodic Timer Block Diagram

15.3.2 Periodic Timer Description

The Periodic Timer is used by the software to generate periodic interrupts needed to monitor system events. The Periodic Timer contains a 16-bit down counter whose initial value is programmed by the software via the PERVAL[15:0] control bits in the Periodic Timer Register. The counter uses the TIMERCLK signal generated by the Clock Module as the clock for the counter. With a system clock of 36.864 MHz, the TIMERCLK will normally be set to 1.152 MHz. This implies a maximum Periodic Timer count duration of 56.89 ms with a granularity of 0.87 µs. The Periodic Timer is enabled by asserting the ENPERTIMER control bit in the Timer Control Register. Once enabled, the counter will load the PERVAL[15:0] control bits and down count to zero. Once the count of zero is reached, the PERINT interrupt is asserted and the counter re-loads the PERVAL[15:0] control bits. The value of the Periodic Timer counter can be read by the software via the PERCNT[15:0] status bits in the Periodic Timer Register.

15.3.3 Periodic Timer Interrupts

PERINT:

This interrupt is set whenever the Periodic Timer is enabled and the Periodic Timer counter reaches a count of zero. See Section 8 for information on how to read and clear interrupts.

15.4 Timer Registers

15.4.1 RTC Register

OFFSET = $140: read-only

Bit	Label	RESET	Read/Write
31-8	Reserved		
7-0	RTC[39:32]	X	R

OFFSET = $144: read-only

Bit	Label	RESET	Read/Write
31-0	RTC[31:0]	X	R

RTC[39:0]: read-only

These bits provide the status of the 40-bit RTC counter. The software must read these bits twice and compare the values to ensure that the counter is not read while the counter is counting since the CPU clock is not synchronous with the RTC counter clock. If the result of the comparison is not equal, the software must read the register again to read the correct count value.

15.4.2 Alarm Register

OFFSET = $148:

Bit	Label	RESET	Read/Write
31-8	Reserved		
7-0	ARARM[39:32]	X	R/W

OFFSET = $14C:

Bit	Label	RESET	Read/Write
31-0	ARARM[31:0]	X	R/W

ALARM[39:0]:

Whenever the RTC counter reaches a count that is equal to these bits, the ALARMINT interrupt will be set.

15.4.3 Timer Control Register

OFFSET = $150:

Bit	Label	RESET	Read/Write
31-8	Reserved		
7	FREEZEPRE	0	R/W
6	FREEZERTC	0	R/W
5	FREEZETIMER	0	R/W
4	ENPERTIMER	0	R/W
3	RTCCLR	0	R/W
2	TESTC8MS	0	R/W
1	ENTESTCLK	0	R/W
0	ENRTCTST	0	R/W

FREEZEPRE:

Setting this bit will cause the lower 8 bits of the RTC counter to freeze. The lower 8 bits of the RTC counter are also used to generate an 8 ms reference signal for the IR Carrier Detect State Machine and the debouncers in the Power Module and I/O Module. Thus, setting this bit will also cause these modules to lose their 8 ms reference signal. This bit can be used by the software to freeze the RTC counter during software debugging.

FREEZERTC:

Setting this bit will cause the upper 32 bits of the RTC counter to freeze. This bit can be used by the software to freeze the RTC counter during software debugging.

FREEZETIMER:

Setting this bit will cause the Periodic Timer counter to freeze. This bit can be used by the software to freeze the Periodic Timer counter during software debugging.

ENPERTIMER:

Setting this bit will cause the Periodic Timer to load the PERVAL[15:0] control bits and begin down counting. The PERINT interrupt will only be asserted if this bit is enabled.

RTCCLR:

Setting this bit to a logic “1” will cause all 40 bits of the RTC counter to initialize to “$0000000000”. The RTC counter will stay cleared and the counter will not start counting until this bit is cleared back to a logic “0”.

TESTC8MS:

Setting this bit will cause the 8 ms reference pulse used by the IR logic and the debouncer circuits to run at 61 μs instead of 8 ms. This bit is used for IC testing to speed up the counters and should normally never be set by the software.

ENTESTCLK:

Setting this bit will cause the 32 kHz input clock for the RTC counter to be driven by the TIMERCLK instead of the 32 kHz input pin. This bit is used for IC testing to speed up the amount of time needed to test the RTC counter and should normally never be set by the software.

ENRTCTST:

Setting this bit will cause all five of the 8-bit counters that comprise the RTC counter to count together. This provides a mechanism for fully exercising all the counters in a timely fashion for IC testing. The software may wish to set this bit in order to test the RTCINT interrupt since normally it would take 388 days to generate this interrupt.

15.4.4 Periodic Timer Register

OFFSET = $154:

Bit	Label	RESET	Read/Write
31-16	PERCNT[15:0]	X	R
15-0	PERVAL[15:0]	X	R/W

PERCNT[15:0]: read-only

These bits provide the status of the Periodic Counter. The software should continually read this register until two consecutive reads provide the same value, in order to ensure that the register is not read while the counter is counting, since the TIMERCLK is not synchronous with the CPU clock.

PERVAL[15:0]:

These bits are loaded into the Periodic Timer counter when the counter is enabled or when the counter reaches a count of zero. The PERINT time is expressed by the following formula ($f_{TIMERCLK}$ is the frequency of the TIMERCLK).

$$\text{Interrupt Rate} = \frac{\texttt{PERVAL[15:0]}+1}{f_{TIMERCLK}}$$

16. UART Module

16.1 Overview

The UART Module contains two identical, fully independent, full duplex UARTs: UARTA and UARTB. Each UART contains the following features:

- adjustable baud rate counter from 230 kHz down to 225 Hz
- single buffered transmit register
- double buffered receive register
- DMA for either transmit or receive
- full duplex
- even, odd or no parity
- 7 or 8 bits per character
- one or two stop bits per character
- pulse option mode to support IRDA Infrared protocol

16.1.1 Related Pins

TXD: OUTPUT

This pin is the UART transmit signal from the UARTA module.

RXD: INPUT

This pin is the UART receive signal to the UARTA module.

IROUT: OUTPUT

This pin is the UART transmit signal from the UARTB module or the Consumer IR output signal if Consumer IR mode is enabled.

IRIN: INPUT

This pin is the UART receive signal to the UARTB module.

16.2 Overall Operation

The operation of each UART is controlled through six registers: Control 1 Register, Control 2 Register, DMA Control 1 Register, DMA Control 2 Register, DMA Count Register, and Transmit/Receive Holding Register.

16.2.1 Power On/Off

Each UART is powered on and off with the ENUART bit of the Control 1 Register. When the ENUART bit is cleared, the UART will power off only after all data in its transmit shift and holding registers has been clocked out. The on/off status of the UART can be monitored via the UARTON bit of the Control 1 Register. While powered off, the UART is completely disabled.

In addition to being powered on, each UART must also have its clock enabled to operate. The UART clock enables, as well as the UART clock frequency, is controlled by the Clock Module (see Section 6).

16.2.2 Baud Rate and Communication Parameters

The baud rate, bits per symbol (7 or 8), parity type (none, even, or odd), and line polarity (normal or inverted) apply to both the transmitter and receiver and may not be independently selected for each direction. All of these except the baud rate are selected with individual bits of the Control 1 Register (see Register Descriptions in Section 16.5).

The baud rate for each UART is determined by the BAUDRATE value in the Control 2 Register and the UART clock frequency ($f_{UARTCLK}$). The following equation determines the baud rate:

$$\text{Baud Rate} = f_{UARTCLK} \div ((\text{BAUDRATE} + 1) * 16)$$

The UART clock frequency is determined by the Clock Module configuration (see Section 6). Table 16.2.1 shows the value that should be written to the BAUDRATE control bits to generate the standard baud rates.

Table 16.2.1 Standard UART Baud Rates

Baud rate	BAUDRATE (when $f_{UARTCLK}$ = 1.8432 MHz)	BAUDRATE (when $f_{UARTCLK}$ = 3.6864 MHz)	BAUDRATE (when $f_{UARTCLK}$ = 9.216 MHz)
115200	0	1	4
57600	1	2	9
38400	2	5	14
19200	5	11	29
9600	11	23	59
4800	23	47	119
2400	47	95	239
1200	95	191	479
600	191	383	959
300	383	767	1919

16.2.3 Interrupt Operation

Each UART may be configured to generate CPU interrupts for a number of events. When any of these events occur, a corresponding status bit will be set in the Interrupt Status 2 Register (see Section 8 for details on the Interrupt Module). Table 16.2.2 lists the UART events that can generate interrupts. The entries are in terms of the mnemonic used in this document and the actual event. In the mnemonic the lower-case 'n' is replaced with 'A' for UARTA or 'B' for UARTB.

Table 16.2.2 UART Interrupt Events

mnemonic	interrupting event
UARTnRXINT	receiver holding register becomes full
UARTnRXOVERRUNINT	receiver shift register becomes full when both holding register and PRXHOLD are full
UARTnFRAMEERRINT	receiver does not detect stop bit-character not 0x0
UARTnBREAKINT	receiver detects BREAK condition-character is 0x0 and no stop bit
UARTnPARITYERRINT	receiver detects parity bit error
UARTnTXINT	transmit holding register becomes empty
UARTnTXOVERRUNINT	transmit holding register written to when full
UARTnEMPTYINT	transmit shift register becomes empty when holding register is also empty
UARTnDMAFULLINT	DMA controller reaches end of specified buffer
UARTnDMAHALFINT	DMA controller reaches half-way point in specified buffer

While these events cause a bit to be set in Interrupt Status 2 Register, they will not cause an interrupt unless they are configured to do so. Interrupts are enabled through the Enable Interrupt 2 Register. Interrupt status bits are cleared through the Clear Interrupt 2 Register. Note that interrupt status bits are set and need to be cleared regardless of whether or not the interrupt is enabled. See Section 8 for more general information about the TMPR3911/12 interrupts.

16.2.3.1 Responding to Interrupt Status Bits

When responding to an event (whether via interrupt or polling), it is important to clear the interrupt status bit before taking the action that will re-enable the event. This will ensure that an event will not be missed.

For example, when the Receive Holding Register becomes full, the UARTnRXINT bit will get set. If the Receive Holding Register is read before the UARTnRXINT bit is cleared, it is possible that sometime after the Receive Holding Register becomes empty and before the UARTnRXINT bit is cleared, a second character will be transferred to the Receive Holding Register. This transfer will also cause the UARTnRXINT bit to be set. Obviously, if the CPU (still reacting to the first event) then clears the UARTnRXINT bit, the second event will be missed. As a result, the CPU will not be "aware" that the Receive Holding Register is full and an overrun interrupt will eventually occur.

16.2.3.2 Related Interrupts

UARTARXINT:

Issues an interrupt whenever the UARTA Receive Holding Register is loaded with data.

UARTARXOVERRUNINT:

Issues an interrupt if the UARTA Receive Holding Register is loaded twice before the interrupt is service.

UARTAFRAMEERRINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register contains a frame error.

UARTABREAKINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register is a break.

UARTAPARITYERRINT:

Issues an interrupt if the current data in the UARTA Receive Holding Register contains a parity error.

UARTATXINT:

Issues an interrupt if the UARTA Transmit Holding Register is available.

UARTATXOVERRUNINT:

Issues an interrupt if the UARTA Transmit Holding Register is written to when the Transmit Holding Register is not available.

UARTAEMPTYINT:

Issues an interrupt if the UARTA Transmit Holding Register and Transmit Shift Register are both empty.

UARTADMAFULLINT:

Issues an interrupt if the UARTA DMA counter reaches the end of the buffer.

UARTADMAHALFINT:

Issues an interrupt if the UARTA DMA counter reaches the mid point of the buffer.

UARTBRXINT:

Issues an interrupt whenever the UARTB Receive Holding Register is loaded with data.

UARTBRXOVERRUNINT:

Issues an interrupt if the UARTB Receive Holding Register is loaded twice before the interrupt is service.

UARTBFRAMEERRINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register contains a frame error.

UARTBBREAKINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register is a break.

UARTBPARITYERRINT:

Issues an interrupt if the current data in the UARTB Receive Holding Register contains a parity error.

UARTBTXINT:

Issues an interrupt if the UARTB Transmit Holding Register is available.

UARTBTXOVERRUNINT:

Issues an interrupt if the UARTB Transmit Holding Register is written to when the Transmit Holding Register is not available.

UARTBEMPTYINT:

Issues an interrupt if the UARTB Transmit Holding Register and Transmit Shift Register are both empty.

UARTBDMAFULLINT:

Issues an interrupt if the UARTB DMA counter reaches the end of the buffer.

UARTBDMAHALFINT:

Issues an interrupt if the UARTB DMA counter reaches the mid point of the buffer.

16.2.4 DMA Operation

Each UART has one DMA channel which may be used with either the transmitter or the receiver. DMA operation involves specifying a memory buffer and enabling DMA. Once enabled, characters will flow either from the memory buffer to the transmitter, or from the receiver to the memory buffer.

The memory buffer is specified in terms of its start address in physical address space, and its length in bytes. Note that the CPU must know the mapping between virtual and physical addresses. The start address must be written to the DMA Control 1 Register. The two LSBs are ignored, thereby forcing the buffer to begin on a long word boundary. The buffer length minus one must be written to the DMA Control 2 Register.

Once the buffer has been specified, DMA may be enabled. This is done with either ENDMARX or ENDMATX in the Control 1 Register. Only one of these two bits should be set at any one time. Operation is undefined if both are set. Once enabled, UART DMA will begin. For the transmitter, this means that when the Transmit Holding Register becomes empty, the DMA controller will transfer a byte from the specified buffer to the Transmit Holding Register, and update its internal buffer pointer. For the receiver, this means that when the Receive Holding Register becomes full, the DMA controller will transfer a byte from the Receive Holding Register to the specified buffer, and update its internal buffer pointer.

The ENDMALOOP control bit in the Control 1 Register determines when DMA will stop. If this bit is cleared, DMA will stop when the end of the buffer is reached. If this bit is set, the DMA controller will run continuously, looping back to the start of the buffer when the buffer end is reached. In either case, DMA can be explicitly halted by clearing the ENDMARX or ENDMATX bit.

The DMA Count Register is a read-only register which contains the UART DMA controller's buffer counter status. The counter counts up so that the value in this status register indicates how much of the buffer remains to be accessed (on this pass) by the DMA controller.

There are two interrupt status bits associated with the DMA controller: UARTnDMAFULLINT and UARTnDMAHALFINT. These are set as the DMA controller buffer pointer reaches the end and the half-way point of the buffer.

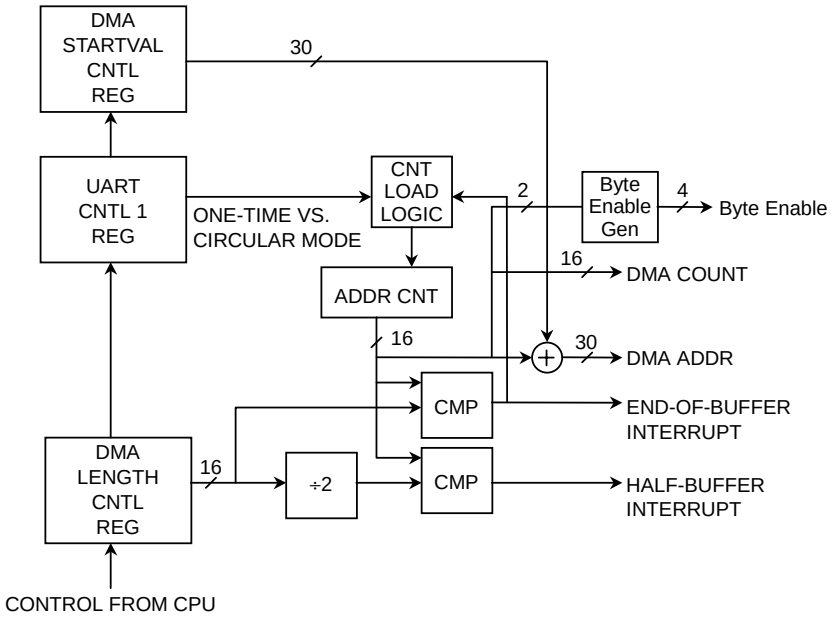


Figure 16.2.1 UART DMA

16.2.5 Internal Loopback

The LOOPBACK bit of the Control 1 Register, when set, places the UART in an internal loopback mode. When in this mode, the transmit and receive lines of the UART are internally tied together, the external transmit line is driven high, and the external receive line is ignored. This loopback mode is for testing purposes only.

16.3 Transmitter Operation

Figure 16.3.1 shows a block diagram of the transmitter portion of each UART. Characters to be transmitted are written by the CPU to the Transmit Holding Register. This should only be done when the Transmit Holding Register is empty which is indicated by either the EMPTY bit of the Control 1 Register, or the UARTnTXINT interrupt. Writing a character to the Transmit Holding Register when it is not empty results in a transmitter overrun.

The character in the Transmit Holding Register is transferred into the Transmit Shift Register when the Transmit Shift Register becomes empty. Transferred with the character is 1 start bit, 1 or 2 stop bits (depending on the setting of TWOSTOP in the Control 1 Register), and 0 or 1 parity bits (depending on the setting of ENPARITY and EVENPARITY in the Control 1 Register). Beginning at the next baud period following the transfer, the character is shifted out to the transmit line. The start bit is shifted first, followed by the LSB through MSB of the (7-bit or 8-bit) character, followed by the parity bit (if any), and ending with the stop bit(s). As the last stop bit is shifted out, the transmitter will load the next character from the Transmit Holding Register if one has been written. If the Transmit Holding Register is empty, the Transmit Shift Register will continue to shift out stop bits (logic level “1”) until a character is written by the CPU into the Transmit Holding Register.

Depending on the configuration, as few as nine bits (7-bit characters, no parity, 1 stop bit), and as many as twelve bits (8-bit characters, parity enabled, 2 stop bits) are transmitted for every character.

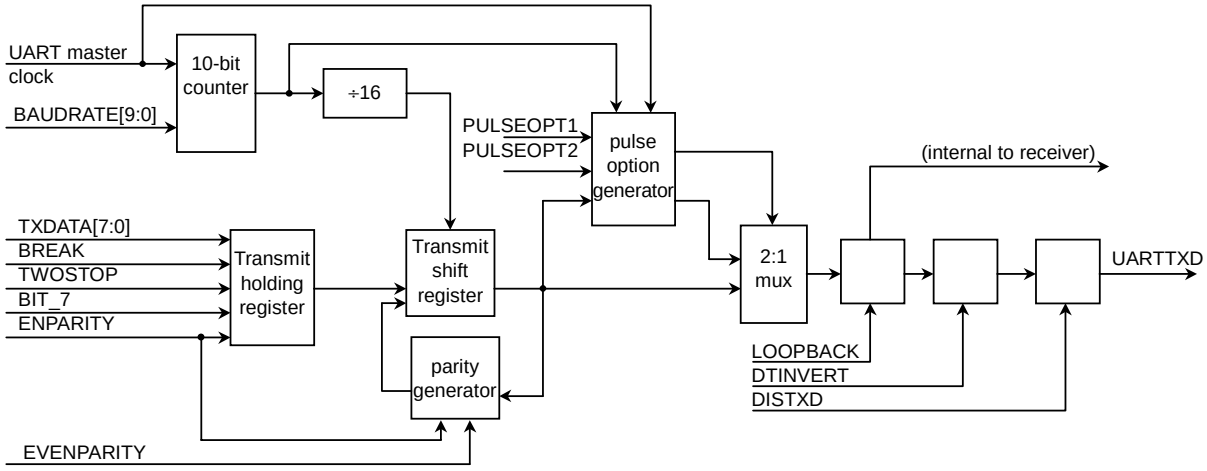


Figure 16.3.1 UART Transmitter

16.3.1 Transmitter Pulse Output Operation

In normal operation (neither pulse option selected), the transmit line is tied directly to the Transmit Shift Register output. Each transmitted bit causes the transmit line to go high or low for a full baud period.

To support the IRDA Infrared protocol, there are two pulse options which result in a modified signal on the output line. When either of these options are selected, transmitted zeros will cause the transmit line to go low for only a fraction of the baud period. Ones are transmitted normally. The difference between the two pulse option modes is the duration that the line will stay low for each transmitted zero.

Pulse Option 1 (selected with PULSEOPT1 of the Control 1 Register) results in a duration of six UART clock cycles. The UART clock frequency is controlled by the Clock Module (see Section 6), When the UART clock frequency is 3.6864 MHz, the low pulse is 1.63 μs (= 6 ÷ 3.6864). Note that this duration is irrespective of the baud rate.

Pulse Option 2 (selected with PULSEOPT2 of the Control 1 Register) results in a duration of 3/16 of the selected baud period.

The mode of operation with both pulse options selected (both PULSEOPT1 and PULSEOPT2 asserted) is reserved for a future implementation.

16.3.2 Transmitter Disable Operation

The transmit line may be disabled with the DISTXD bit of the Control 1 Register. In this mode the transmit line is held at logic level “0”. Otherwise, the transmitter operates normally as described above.

16.3.3 Transmitter BREAK Operation

A BREAK may be transmitted by writing the value 0x100 to the Transmit Holding Register. The BREAK is transmitted after the character (if any) in the Transmit Shift Register is sent. The BREAK condition continues until explicitly terminated by the CPU. The CPU may continue to write the value 0x100 to the Transmit Holding Register in order to control the duration of the BREAK precisely. The BREAK condition is terminated when the CPU writes the value 0x00 to the Transmit Holding Register.

16.3.4 Transmitter Overrun

The transmit overrun condition occurs if the CPU writes a character to the Transmit Holding Register when it is not empty. This condition is indicated by the UARTnTXOVERRUNINT interrupt status bit. The character in the Transmit Holding Register resulting from an overrun is undefined.

16.4 Receiver Operation

Figure 16.4.1 shows a block diagram of the receiver portion of each UART. In normal operation, the receiver waits for a start bit on the receive line. One baud cycle after receiving the start bit, the next 7 or 8 bits (depending on BIT_7 in the Control 1 Register) are clocked into the Receive Shift Register. If parity is enabled (depending on ENPARITY and EVENPARITY in the Control 1 Register), the next bit is compared with the expected parity. As the stop bit is clocked in and checked, the assembled character is transferred from the Receive Shift Register to the PRXHOLD Register.

The PRXHOLD Register is a buffer between the Receive Shift Register and the Receive Holding Register. The PRXHOLDFULL bit of the Control 1 Register, when asserted, indicates that the PRXHOLD Register is full.

On the first UART clock period that the Receive Holding Register is empty and the PRXHOLD Register is full, the character is transferred to the Receive Holding Register. This transfer results in the RXHOLDFULL bit of the Control 1 Register and the UARTnRXINT interrupt status bit both asserting.

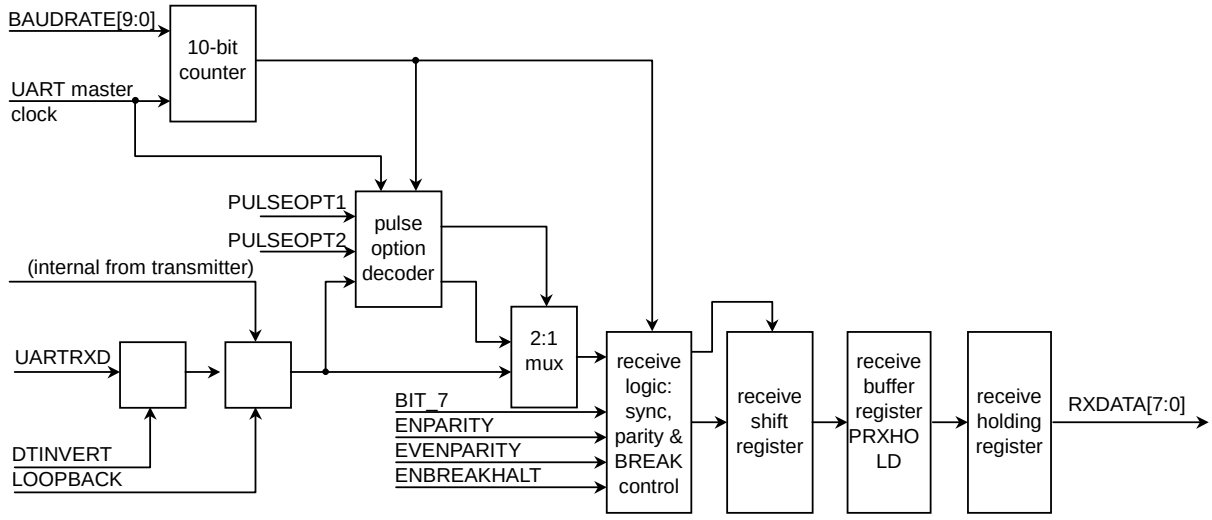


Figure 16.4.1 UART Receiver

16.4.1 Receiver BREAK Operation

The receiver detects a BREAK condition when the received character is 0x00 and there is no stop bit. When this occurs the receiver generates a UARTnBREAKINT interrupt. The interrupt does not assert until the BREAK character is transferred to the Receive Holding Register. In other words, the interrupt is held off until all characters preceding the BREAK are read by the CPU.

Additionally, the operation of the receiver in response to a BREAK is affected by the ENBREAKHALT bit of the Control 1 Register. Normally (with the ENBREAKHALT bit cleared), the receiver will start receiving characters again as soon as a stop bit is detected on the line. However, if the ENBREAKHALT bit is set, the receiver shuts down until the BREAK character is read by the CPU from the Receive Holding Register. While shut down, the receiver ignores all data on the receive line.

16.4.2 Receiver Frame Error Condition

A receiver frame error occurs when the receiver does not detect a stop bit following the character. This condition results in a UARTnFRAMEERRINT interrupt being generated. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the frame error is transferred to the Receive Holding Register.

16.4.3 Receiver Overrun Condition

A receiver overrun occurs when a character is received and both the PRXHOLD Register and Receive Holding Register are full. This condition occurs at the point that the stop bit of the new character is clocked in and results in a UARTnRXOVERRUNINT interrupt. The new character will overwrite the character in the PRXHOLD Register. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the overrun error is transferred to the Receive Holding Register.

16.4.4 Receiver Parity Error Condition

A receiver parity error occurs when parity is enabled and the receiver detects the wrong polarity of the bit following the character and preceding the stop bit. This condition results in a UARTnPARITYERRINT interrupt being generated. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the parity error is transferred to the Receive Holding Register.

16.4.5 Receiver Pulse Operation

In normal operation the received signal corresponds to the polarity of the data for the entire baud period. When the received signal is from a pulsed transmitter, zeros are low for only a fraction of the baud period. If either pulse option is selected (with PULSEOPT1 or PULSEOPT2 of the Control 1 Register) the receiver reconstructs the data stream by extending low pulses to the full baud period.

16.5 UART Registers

16.5.1 UART Control 1 Register

OFFSET = $0B0: UARTA

OFFSET = $0C8: UARTB

Bit	Label	RESET	Read/Write
31	UARTON	0	R
30	EMPTY	1	R
29	PRXHOLDFULL	0	R
28	RXHOLDFULL	0	R
27-16	Reserved		
15	ENDMARX	0	R/W
14	ENDMATX	0	R/W
13	TESTMODE	0	R/W
12	ENBREAKHALT	0	R/W
11	ENDMATEST	0	R/W
10	ENDMALOOP	0	R/W
9	PULSEOPT2	0	R/W
8	PULSEOPT1	0	R/W
7	DTINVERT	0	R/W
6	DISTXD	1	R/W
5	TWOSTOP	0	R/W
4	LOOPBACK	0	R/W
3	BIT_7	0	R/W
2	EVENPARITY	0	R/W
1	ENPARITY	0	R/W
0	ENUART	0	R/W

UARTON: read-only

When the ENUART bit is disabled, the module will not shut down until the Transmit Holding Register and Transmit Shift Register are empty plus a couple of clocks. This bit provides the status as to whether the module is still enabled or not.

EMPTY: read-only

This bit is high if the Transmit Holding Register and Transmit Shift Register are both empty.

PRXHOLDFULL: read-only

The receive data path consists of an 8-bit shift register plus two 8-bit holding registers. Whenever the receiver finishes receiving a character, it will transfer the contents into the PRXHOLD Register. The contents of this register will then be loaded into the RXHOLD Register whenever this RXHOLD Register is empty. The RXHOLD Register is emptied by reading a byte from the Receive Holding Register. This PRXHOLDFULL bit provides status as to whether there is a valid byte of data in the PRXHOLD Register.

RXHOLDFULL: read-only

This bit provides the status of the RXHOLD Register.

ENDMARX:

This bit enables the DMA receive function. This bit should not be set until the UARTDMACNTL1 and UARTDMACNTL2 Registers are setup and the module is enabled. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

ENDMATX:

This bit enables the DMA transmit function. This bit should not be set until the UARTDMACNTL1 and UARTDMACNTL2 registers are setup, the module is enabled and the empty flag is set. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

TESTMODE:

This bit is used for IC testing and should never be set.

ENBREAKHALT:

Setting this bit will cause the receiver to halt after receiving a break, until the Receive Holding Register is emptied and the UARTRXD signal goes to the marking state. Otherwise, the receiver will halt until the UARTRXD signal goes to the marking state without regard to the status of the Receive Holding Register.

ENDMATEST:

This bit is used for IC testing and should not be set.

ENDMALOOP:

The DMA controller supports two modes depending on the state of this bit. When ENDMALOOP is low, the DMA controller will stop executing when it reaches the end of the DMA buffer. When ENDMALOOP is high, the DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating.

PULSEOPT2:

Setting this bit will cause the transmitted data to pulse low for three baud clocks instead of the normal 16 baud clocks. Setting either this bit or PULSEOPT1 will cause the receiver to expect the data to be a pulsed input.

PULSEOPT1:

Setting this bit will cause the transmitted data to pulse low for a fixed six UART clocks instead of the normal 16 baud clocks.

DTINVERT:

Setting this bit will cause the UARTTXD and UARTRXD signals to be inverted.

DISTXD:

Setting this bit will cause the UARTTXD signal to go low.

TWOSTOP:

Setting this bit will cause the transmitter to transmit two stop bits instead of the normal one stop bit.

LOOPBACK:

Setting this bit will cause the transmitted data to internally loop back to the receive data. The UARTTXD pin is held high when this bit is set.

BIT_7:

Setting this bit selects 7-bit per character mode instead of 8-bit per character mode.

EVENPARITY:

Setting this bit selects even parity instead of odd parity, if the ENPARITY bit is set.

ENPARITY:

Setting this bit will cause parity to be generated and received.

ENUART:

Setting this bit will enable the UART Module. When this bit is cleared the module is kept in a reset state but clearing this bit gives no effect on the UART Control 1 Register. This bit should not be set until all other control bits are setup.

16.5.2 UART Control 2 Register

OFFSET = $0B4: UARTA write-only

OFFSET = $0CC: UARTB write-only

<TMPR3911BU, TMPR3911BXB>

Bit	Label	RESET	Read/Write
31-10	Reserved		
9-0	BAUDRATE[9:0]	X	W

<TMPR3912AU-92, TMPR3912XB-92>

Bit	Label	RESET	Read/Write
31-11	Reserved		
10-0	BAUDRATE[10:0]	X	W

BAUDRATE[9:0]

BAUDRATE[10:0]: write-only

These bits define the baud rate of the transmitter and receiver. The following equation determines the baud rate:

$$\text{Baud Rate} = f_{UARTCLK} \div ((\text{BAUDRATE} + 1) * 16)$$

16.5.3 UART DMA Control 1 Register

OFFSET = $0B8: UARTA write-only

OFFSET = $0D0: UARTB write-only

Bit	Label	RESET	Read/Write
31-2	DMASTARTVAL[31:2]	X	W
1-0	Reserved		

DMASTARTVAL[31:2]: write-only

These bits define the start address (physical address) for the DMA buffer.

16.5.4 UART DMA Control 2 Register

OFFSET = $0BC: UARTA write-only

OFFSET = $0D4: UARTB write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	DMALENGTH[15:0]	X	W

DMALENGTH[15:0]: write-only

These bits define the length of the DMA buffer. The DMALENGTHH field should be loaded with a value equal to “desired buffer length – 1”. The last address in the DMA buffer is given by DMASTARTVAL[31:2] + DMALENGTH[15:0].

16.5.5 UART DMA Count

OFFSET = $0C0: UARTA read-only

OFFSET = $0D8: UARTB read-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	DMACNT[15:0]	X	R

DMACNT[15:0]: read-only

These bits provide the status of the DMA counter.

16.5.6 UART Transmit Holding Register

OFFSET = $0C4: UARTA write-only

OFFSET = $0DC: UARTB write-only

Bit	Label	RESET	Read/Write
31-9	Reserved		
8	BREAK	X	W
7-0	TXDATA[7:0]	X	W

BREAK: write-only

Setting this bit, along with writing $00 to TXDATA[7:0], will cause a break to be generated. The break will continue until this bit is cleared, along with TXDATA[7:0] maintained at $00. This register should only be loaded after the UARTTXINT interrupt is set. The break will flow through the Transmit Holding Register, thus it will not start the break until the current character that is being transmitted from the Transmit Shift Register is finished.

TXDATA[7:0]: write-only

These bits are the data to be transmitted. This register should only be loaded after the UARTTXINT interrupt is set. For 7-bit mode, only bits 6:0 are valid.

16.5.7 UART Receiver Holding Register

OFFSET = $0C4: UARTA read-only

OFFSET = $0DC: UARTB read-only

Bit	Label	RESET	Read/Write
31-8	Reserved		
7-0	RXDATA[7:0]	X	R

RXDATA[7:0]: read-only

These bits are the receive data. The bits are valid after the UARTRXINT interrupt is set or when the RXHOLD flag is asserted. For 7-bit mode, only bits 6:0 are valid.

17. Video Module

17.1 Overview

The Video Module within the TMPR3911/12 contains logic for transferring bit-mapped graphics data between a video buffer located in system memory to an external LCD. The LCD must be refreshed at a rate of 50 to 100 frames per second to maintain a steady picture on the display.

Data is clocked into the LCD's shift register using parallel data lines (4 or 8 bits wide). Once a horizontal line of data has been shifted into the shift register, this line of pixels is transferred to successive lines of the LCD panel. This sequence is repeated until all the lines of the display have been shifted and transferred, producing a full frame of display. At the end of the frame the pointers in the LCD return to the top of the display.

Different LCD's have different requirements on the number of horizontal and vertical pixels, the number of parallel lines required for the data interface, interface timing, and refresh rate. The Video Module inside the TMPR3911/12 is programmable in order to support many different types of LCD's. The Video Module also contains logic that produces 4-level or 16-level gray scale on a monochrome LCD using a time-based dithering algorithm. Also supported is an 8-bit per pixel color mode for interfacing to color LCD's.

17.1.1 Related Pins

FRAME: OUTPUT

This pin is the frame synchronization pulse signal between the Video Module and the LCD, and is used by the LCD to return its pointers to the top of the display. The Video Module asserts FRAME after all the lines of the LCD have been shifted and transferred, producing a full frame of display.

DF: OUTPUT

This pin is the AC signal for the LCD. Since LCD plasma tends to deteriorate whenever subjected to a DC voltage, the DF signal is used by the LCD to alternate the polarity of the row and column voltages used to turn the pixels on and off. The DF signal can be configured to toggle on every frame or can be configured to toggle every programmable number of LOAD signals.

LOAD: OUTPUT

This pin is the line synchronization pulse signal between the Video Module and the LCD, and is used by the LCD to transfer the contents of it's horizontal line shift register to the LCD panel for display. The Video Module asserts LOAD after an entire horizontal line of data has been shifted into the LCD.

CP: OUTPUT

This pin is the clock signal for the LCD. Data is pushed by the Video Module on the rising edge of CP and sampled by the LCD on the falling edge of CP.

VDAT[3:0]: OUTPUT

These pins are the data for the LCD. These signals are directly connected to the LCD for 4-bit non-split displays. For 4-bit split and 8-bit non-split displays, an external register is required to demultiplex the 4-bit data into the desired 8 parallel data lines needed for the LCD.

DISPON: OUTPUT

This pin is the display-on enable signal for the LCD.

VIDDONE: OUTPUT

This pin is used to externally synchronize events to periods when the video is not shifting.

17.2 Interface Requirements

17.2.1 Display Types

The Video Module interface supports 3 types of LCD's: 4-bit split, 4-bit non-split, and 8-bit non split. Figure 17.2.1 shows these 3 different display types for monochrome displays and Figure 17.2.2 shows these 3 different display types for color displays.

A 4-bit split display uses 8 parallel data lines to shift data to both the upper and lower halves of the display simultaneously, with four bits of data shifted to the upper half and four bits of data shifted to the lower half. The end of frame is reached when each half of the display has been shifted and transferred. The Video Module within the TMPR3911/12 contains control logic which generates the appropriate video buffer addressing required for split displays (see Section 17.3.5). Since the TMPR3911/12 only provides 4 pins for video output data (VDAT[3:0]), an external 4-bit register is required to demultiplex the 4-bit data into the desired 8 parallel data lines (4 upper, 4 lower) needed for the LCD, as shown in Figure 17.2.4.

A 4-bit non-split display uses 4 parallel data lines to shift data to successive single horizontal lines of the display at a time, until the entire frame has been shifted and transferred. No external register is required for this configuration; the 4 pins for video output data from the TMPR3911/12 can be directly connected to the LCD.

An 8-bit non-split display uses 8 parallel data lines to shift data to successive single horizontal lines of the display at a time, until the entire frame has been shifted and transferred. This operates the same as the 4-bit non-split display, except that 8 parallel lines are shifted at a time instead of 4 parallel lines. As also required for the 4-bit split displays, an external 4-bit register is required to demultiplex the 4-bit video output data from the TMPR3911/12 into the desired 8 parallel data lines needed for the 8-bit non-split LCD, as shown in Figure 17.2.4.

Monochrome displays require 1 bit of video data per pixel. Figure 17.2.1 shows the pixel ordering of the parallel data lines for the 3 types of monochrome displays. Color displays require 3 bits (red, green, and blue) of video data per pixel, resulting in a horizontal shift register of length 3 times the number of pixels per horizontal line. This RGB-tri is shifted to the LCD as consecutive bits via the parallel data lines. Figure 17.2.2 shows the RGB-tri and pixel ordering of the parallel data lines for the 3 types of color displays.

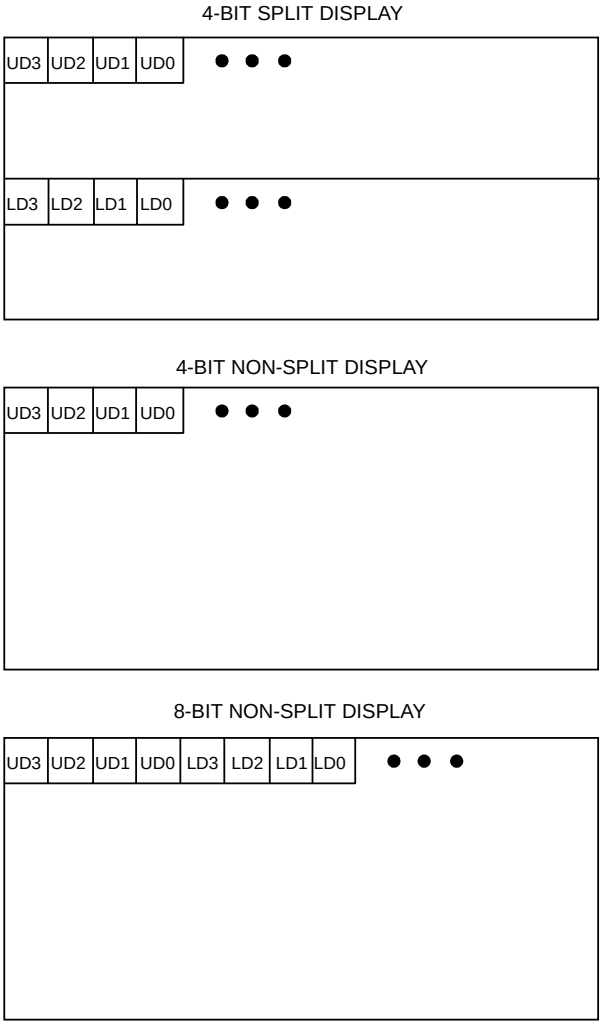


Figure 17.2.1 Monochrome Display Types

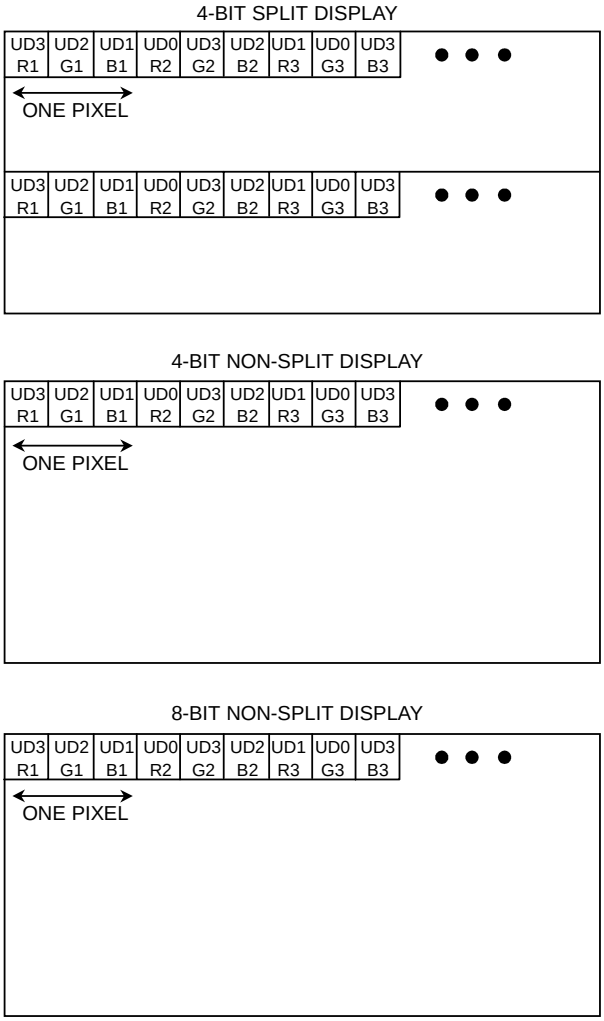


Figure 17.2.2 Color Display Types

17.2.2 Timing Requirements

Data is shifted from the TMPR3911/12 to the LCD using the VDAT[3:0] signals. The CP signal is used to clock the data into the LCD's shift register. After each horizontal line of data has been shifted into the LCD's shift register, the LOAD signal is asserted to cause the line to be displayed on the LCD panel. Then after all of the lines have been displayed, the FRAME signal is asserted to cause the LCD's line pointer to start over at the top of the display. Figure 17.2.3 shows the various timing requirements for the LCD interface (the example shown is for 4-bit non-split displays).

The DF signal provides an AC reference for the display. Since LCD plasma tends to deteriorate whenever subjected to a DC voltage, the DF signal is used by the LCD to alternate the polarity of the row and column voltages used to turn the pixels on and off. The DF signal can be configured to toggle on every frame (DFMODE = 0) or can be configured to toggle every programmable number of LOAD signals (DFMODE = 1), with this toggle period determined by the DFVAL[7:0] control register setting. Figure 17.2.3 shows an example for DFMODE = 0 and for DFMODE = 1 with DFVAL[7:0] = $1.

The rate of the CP clock signal is controlled by the BAUDVAL[4:0] control register setting. The number of CP clock pulses per line busted between successive LOAD pulses is determined by the number of pixels per line programmed into the HORZVAL[8:0] control register. The VIDRATE[9:0] control register setting determines the frequency of the LOAD pulses, which corresponds to the video line rate. This line rate, along with the number of LCD lines programmed into the LINEVAL[9:0] control register, then determines the video frame rate.

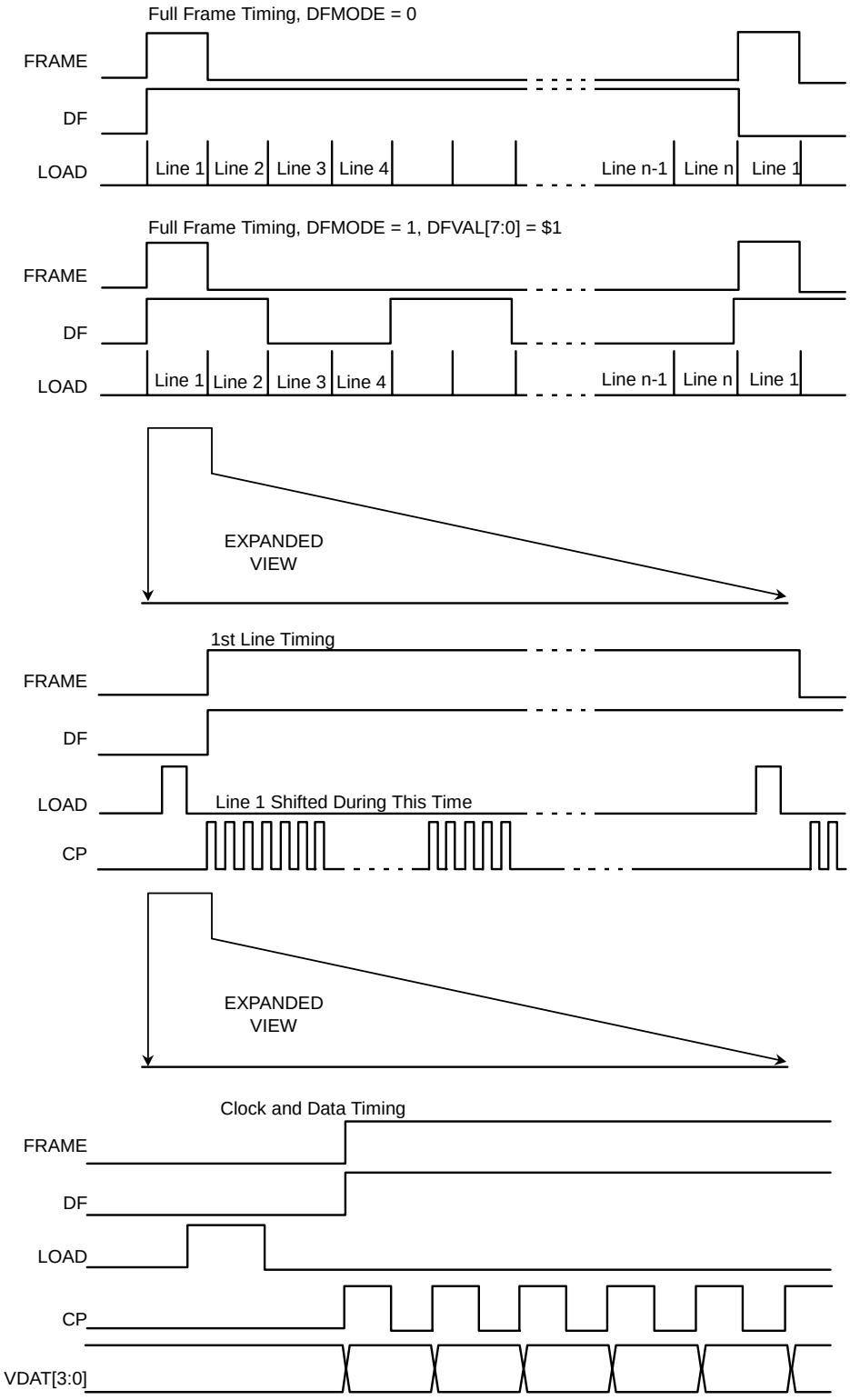


Figure 17.2.3 LCD Timing

As mentioned previously, since the TMPR3911/12 only provides 4 pins for video output data (VDAT[3:0]), an external 4-bit register is required to demultiplex the 4-bit data into the desired 8 parallel data lines needed for both 4-bit split and 8-bit non-split LCD's, as shown in Figure 17.2.4. To support these configurations, the Video Module drives the VDAT output data in a time-multiplexed format, alternating 4 bits of upper data (UD) with 4 bits of lower data (LD) on successive data cycles. The UD data is latched using the external register with CP used as the register's clock. Then, the LCD can clock in all 8 bits in parallel (UD plus LD, which are the VDAT signals directly connected to the LCD) using the falling edge of CP, as shown in Figure 17.2.5.

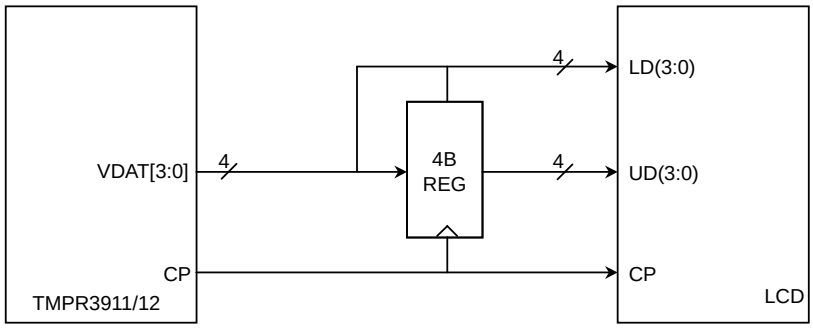


Figure 17.2.4 External Register Configuration for 4-Bit Split and 8-Bit Non-Split Displays

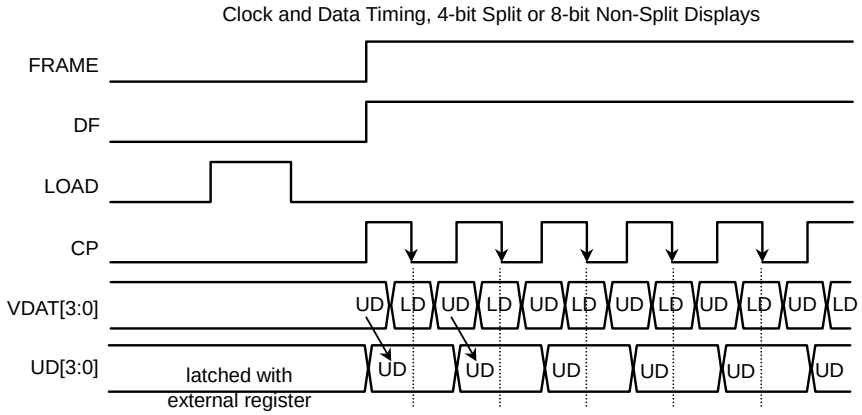


Figure 17.2.5 LCD Timing for 4-Bit Split and 8-Bit Non-Split Displays

17.3 Implementation

17.3.1 Block Diagram

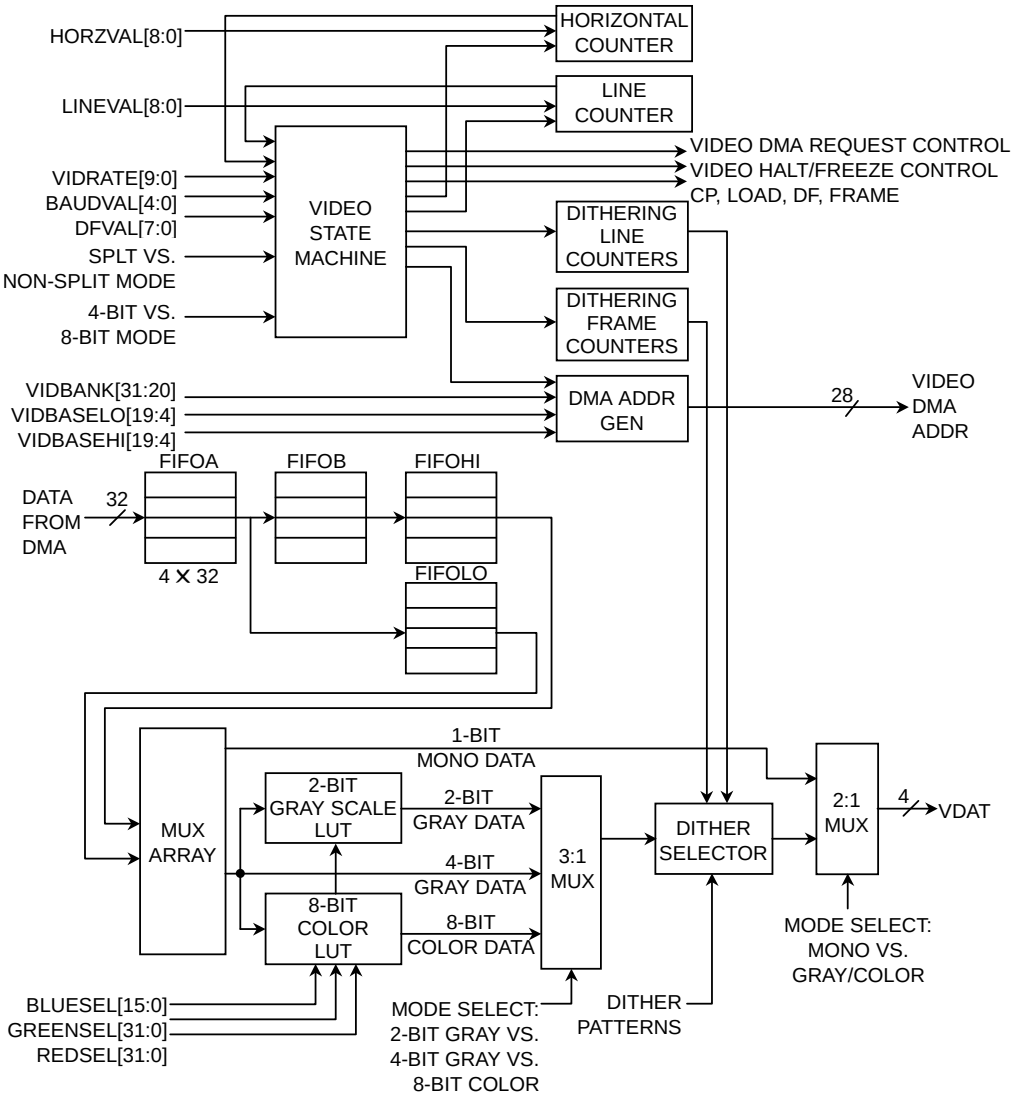


Figure 17.3.1 Video Module Block Diagram

17.3.2 Video State Machine

The Video State Machine contains programmable logic which allows the TMPR3911/12 to support the different interface timing and rates commonly found in different LCD's. This is implemented using several sets of programmable counters.

The CP clock rate is controlled by the Baud Rate Counter, which is preloaded with the BAUDVAL[4:0] control register setting. This counter generates the baud clock by dividing down from the main video clock (which is the same rate as CLK). The Baud Rate Counter is free-running, except whenever the HALT or FREEZEFRAME conditions are asserted. The HALT condition occurs if the video DMA requests cannot be serviced fast enough by the memory subsystem. In this case, HALT is asserted and the video baud clock is stopped, such that video data will momentarily be stopped from being sent to the LCD. The FREEZEFRAME condition is asserted in response to the assertion of the ENFREEZEFRAME control bit. This causes the video baud clock to stop and the video to stop shifting cleanly at the end of the current frame.

The actual CP clock signal is generated by gating the video baud clock with the valid shift period, as determined by the count output of the Horizontal Counter. The clock used to shift the video output data also depends on the display type (4-bit split, 4-bit non-split, or 8-bit non split), since the 4-bit split and 8-bit non split types require data to be shifted as twice the CP rate such that an external register can be used to demultiplex the 4-bit data into the desired 8 parallel data lines.

The video line rate and LOAD pulse frequency is controlled by the Video Rate Counter, which is preloaded with the VIDRATE[9:0] control register setting. This counter generates the LOAD pulse by counting the programmed number of video baud clocks. In addition, setting the LOADDLY control bit will result in an additional baud clock delay to be inserted for the FRAME and CP signals with respect to the LOAD signal, as shown in Figure 17.3.2. Some LCD's may require this timing restriction.

The toggle rate of the DF signal is controlled by the DF Counter, which is preloaded with the DFVAL[7:0] control register setting. The DF Counter will count each time a LOAD pulse is generated. If the DFMODE control bit is set to logic "0", the DF signal is configured to toggle on every frame. If the DFMODE control bit is set to logic "1", the DF signal is configured to toggle every time the DF Counter rolls over.

The FRAME pulse generation is controlled by both the Horizontal Counter and Line Counter (see Sections 17.3.3 and 17.3.4). Together, these counters determine the video frame rate. The FRAME pulse is asserted for a duration of the entire first line at a frequency of once per frame.

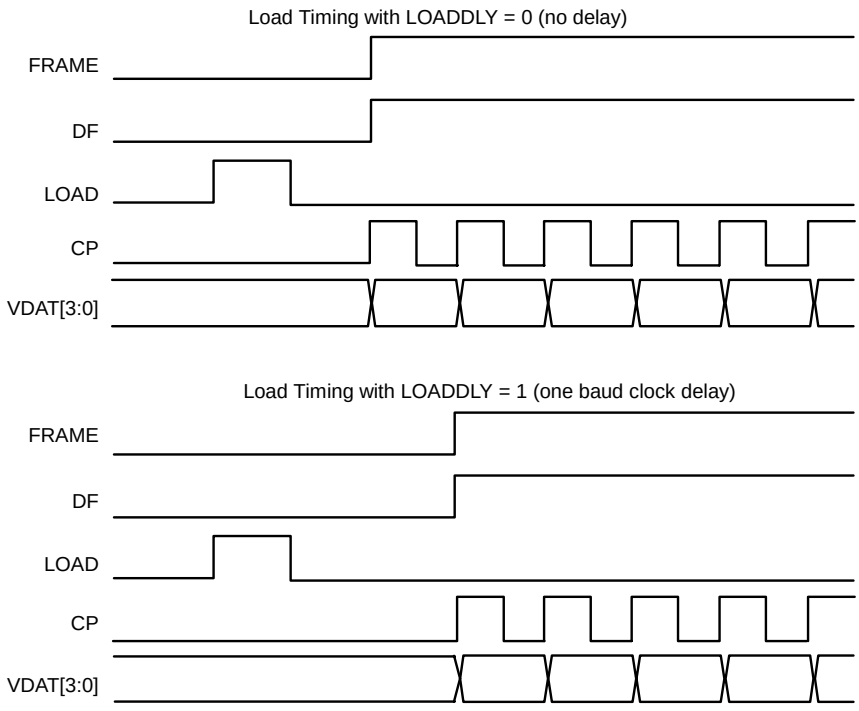


Figure 17.3.2 LOADDLY Timing

The Video State Machine also generates the video DMA request signals used to fetch the video image data from memory. Each data fetch sequence consists of one DMA request followed by 4 successive video DMA transactions, since video DMA data transfers are busted as 4 longwords (32 bits each) at a time from system memory to the FIFO buffers within the Video Module. The video DMA requests are initiated whenever the FIFO buffers are available to accept more data (see Section 17.3.6).

17.3.3 Horizontal Counter

The Horizontal Counter is used to count the number of bits that are shifted (in a burst fashion) into the LCD's line shift register. For each new line, the Horizontal Counter is preloaded with the HORZVAL[8:0] control register setting and counts each time the 4 bits (or 8 bits for 8-bit non-split LCD's) of video output data are shifted out. The HORZVAL value should equal the horizontal size of the LCD divided by 4 (or 8 for 8-bit non-split LCD's) −1. Thus, the 9-bit Horizontal Counter supports a maximum horizontal size of 2048 pixels.

Each time the Video Rate Counter reaches it's terminal count, the Video State Machine asserts the LOAD pulse in order to transfer the entire line shift register contents to the LCD panel. Thus, the Video Rate Counter preload value VIDRATE[9:0] should be configured with a value greater than the Horizontal Counter preload value HORZVAL[8:0] in order to load the entire line shift register contents to the LCD after all the bits have been shifted into the line shift register. Figure 17.2.3 shows an example of the relative timing for the LOAD pulse versus the burst of CP clocks used to shift a horizontal line of data.

17.3.4 Line Counter

The Line Counter is used to count the number of lines that are displayed by the LCD. Each time the Line Counter reaches it's terminal count, the Video State Machine asserts the FRAME pulse in order to begin a new frame and cause the pointers in the LCD to return to the top of the display. For each new frame, the Line Counter is preloaded with the LINEVAL(9:0) control register setting and counts each time a LOAD pulse is generated. The LINEVAL value should equal the number of lines for the LCD −1. The 10-bit Line Counter supports a maximum vertical size of 1024 lines.

17.3.5 DMA Address Generation

The DMA Address Generation circuit consists of logic which generates the address for fetching video data from the video buffer located in system memory. The video buffer can be relocated anywhere in memory (over the full 32-bit address space) by configuring the VIDBANK[31:20], VIDBASEHI[19:4], and VIDBASELO[19:4] control register settings. See Figure 17.3.3 for a block diagram of the video DMA address generation circuit.

For a split LCD, the VIDBANK and VIDBASEHI bits are concatenated to provide the start address for the upper (HI) portion of the video buffer, while the VIDBANK and VIDBASELO bits are concatenated to provide the start address for the lower (LO) portion of the video buffer. For a non-split LCD, the VIDBANK and VIDBASEHI bits are concatenated to provide the start address for the video buffer.

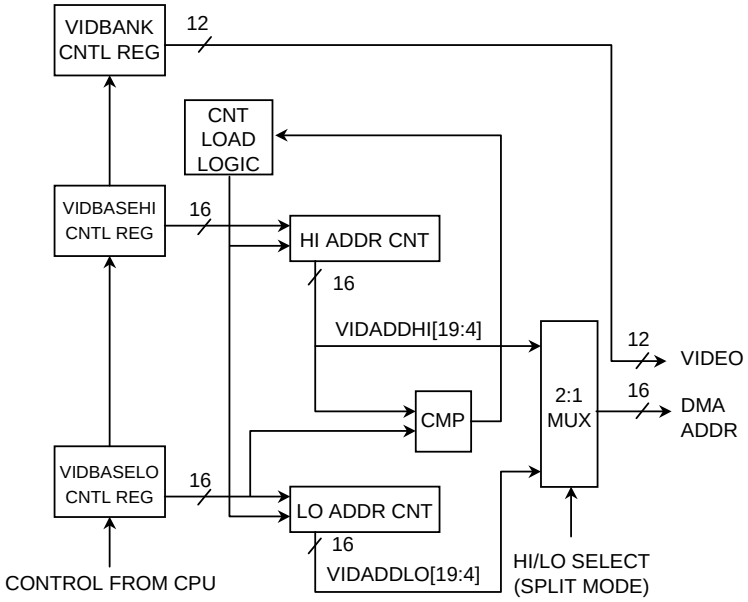


Figure 17.3.3 Video DMA Address Generation

The DMA Address Generation circuit consists of 2 sets of counters, the HI Address Counter and the LO Address Counter, which generate the address for the HI and LO portions of the video buffer, respectively. Both of these counters are incremented whenever the video FIFO buffer is filled from a burst of DMA-initiated data transfers. For a split LCD, the VIDBANK and VIDADDHI bits are concatenated to provide the address for the upper (HI) portion of the video buffer, while the VIDBANK and VIDADDLO bits are concatenated to provide the address for the lower (LO) portion of the video buffer. A 2:1 multiplexer is used to select between these two combined addresses, depending on whether the current video data fetch is for a HI or LO portion of the buffer. For a non-split LCD, the VIDBANK and VIDADDHI bits are concatenated to provide the address for the video buffer. Figure 17.3.4 shows the DMA address fields for split and non-split display formats. Video data is fetch as a burst of 4 longwords at a time and takes advantage of the fast-page mode available for DRAM or SDRAM.

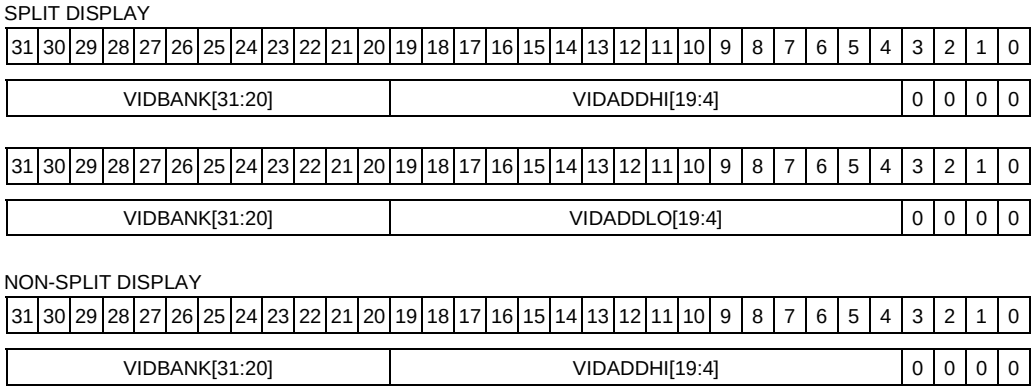


Figure 17.3.4 Video DMA Address Fields

The VIDBANK control bits provide a bank location for the video buffer. The video bank can be located anywhere in memory over the full 32-bit address space, in increments of 1 MByte. These bits are latched at the beginning of each frame, making it possible to change these bits while the Video Module is active and have the video address switch to the new bank at the start of the next frame. In addition, the video buffer start address can be located anywhere within the 1 MByte bank, in increments of 16 bytes.

The HI and LO Address Counters are both up-counters. Thus, VIDBASEHI must be configured to a lower value than VIDBASELO. The HI and LO Address Counters are reloaded with the VIDBASEHI and VIDBASELO values, respectively, whenever the HI Address Counter output VIDADDHI equals the VIDBASELO value. Therefore, VIDBASELO value must always be set to one address greater than the end address of the upper (HI) portion of the display. For non-split displays, the VIDBASELO value must always be set to one address greater than the end address of the display.

17.3.6 Video FIFO

The Video Module contains several sets of First-In First-Out (FIFO) buffers to support video DMA burst data transfers. As shown in Figure 17.3.1, these FIFOs are organized as 4 sets of FIFOs. Each set of FIFOs is further subdivided into 4 latches (4 × 32-bits or a total of 128 bits wide).

The first set of FIFOs is referred to as FIFOA. Video data is loaded one longword (32 bits) at a time from the memory subsystem consecutively into each of the 4 latches of FIFOA. FIFOA then feeds both FIFOB and FIFOLO, while FIFOB feeds FIFOHI, with all of these data transfers performed 128 bits in parallel at a time. For example, the 1st latch of FIFOA directly loads into the 1st latch of FIFOB, at the same time the 2nd latch of FIFOA directly loads into the 2nd latch of FIFOB, etc. FIFOHI and FIFOLO are used to separately buffer the video data for the upper (HI) and lower (LO) portions of a split display, if this mode is enabled.

For a non-split display, FIFOLO is not used. In this mode, FIFOA accepts and buffers the 4-longword burst of data from the memory subsystem. Whenever FIFOA is full, and FIFOB is empty, the 4 longwords from FIFOA will then be transferred to FIFOB. Similarly, whenever FIFOA is full again and FIFOB is full, and FIFOHI is empty, the 4 longwords from FIFOB will then be transferred to FIFOHI. Then, the data is emptied from FIFOHI one longword at a time as the data is fetched for eventual shifting out to the LCD.

For a split display, both FIFOHI and FIFOLO are used. In this mode, FIFOA accepts and buffers the 4-longword burst of data from the memory subsystem. Whenever FIFOA is full, and FIFOB is empty, the 4 longwords from FIFOA will then be transferred to FIFOB. Then whenever FIFOA and FIFOB are both full, and FIFOHI and FIFOLO are both empty, the 4 longwords from FIFOB will then be transferred to FIFOHI at the same time the 4 longwords from FIFOA are transferred to FIFOLO. This is the mechanism used for separately buffering the video data for the upper (HI) and lower (LO) portions of the split display. As explained in the previous section, the video DMA Address Generation logic ensures that the correct HI or LO buffer address is generated at the correct time. Finally, the data is emptied from FIFOHI or FIFOLO one longword at a time via a multiplexer array, based on whether the current output cycle corresponds to an upper or lower portion of the split display.

17.3.7 Gray Scale LUT

The Video Module supports monochrome, 4-level or 16-level gray scale, and 8-bit per pixel color modes. The gray scale and color modes are implemented using a programmable lookup table (LUT) for selecting the shades of gray or color, followed by a programmable time-based dithering algorithm in the Dither Selector circuit. The monochrome mode bypasses these circuits and basically serializes the FIFOHI (and FIFOLO if a split display is used) output data into 4-bit streams for shifting to the LCD.

The gray scale modes of the Video Module are implemented by varying the duty cycle (i.e., number of frames) for which a given pixel is turned on, giving the monochrome display an appearance of gray scale. In order to reduce the noticeable flicker caused by turning on and off adjacent pixels at the same time, a time-based dithering algorithm is used.

Two gray scale modes are supported by the Video Module: 2-bit gray (4-level) or 4-bit gray (16-level). Since the LUT used in the Video Module consists of 16 total entries or gray scales, the 4-bit gray scale processing path does not require a LUT (all 16 gray scales are available and used in this mode). The 4-bit gray scale mode requires that the number of bits stored in the video buffer is equal to 4 times the number of pixels, with each pixel corresponding to 4 consecutive data bits stored in the video buffer. Each set of these 4 gray scale bits maps to one pixel of dithered data generated by the Dither Selector.

The 2-bit gray mode does use a LUT, which allows any 4 gray levels to be selected out of the 16 total possible gray levels. The 2-bit gray scale mode requires that the number of bits stored in the video buffer is equal to 2 times the number of pixels, with each pixel corresponding to 2 consecutive data bits stored in the video buffer. The 2-bit gray scale LUT uses the BLUESEL[15:0] control bits as the programmable LUT entries. The 16-bit BLUESEL field is divided into 4 nibbles: BLUESEL[15:12], BLUESEL[11:8], BLUESEL[7:4], and BLUESEL[3:0]. Each nibble then corresponds to one of the 4 gray scales selected out of 16 possible, with each 2 pre-gray-scale bits from memory used to lookup one of these 4 gray scales at a time (2 to 4 mapping). Figure 17.3.5 shows a block diagram of the 2-bit gray scale LUT logic. Each set of the 4 gray scale bits from the LUT output maps to one pixel of dithered data generated by the Dither Selector.

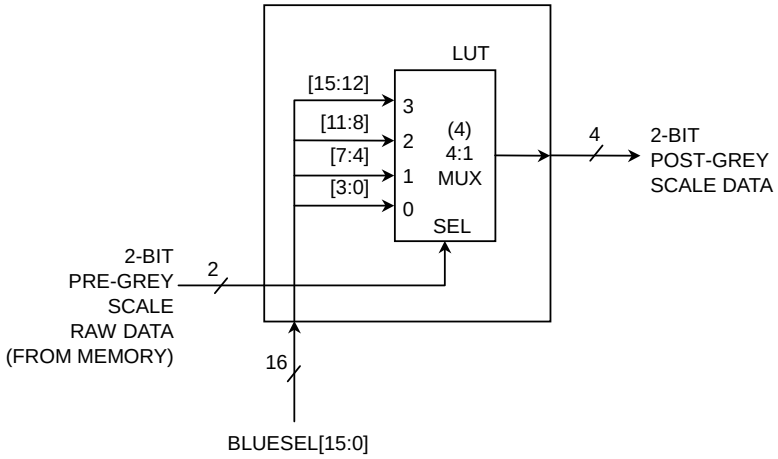


Figure 17.3.5 2-BIT Gray Scale LUT

17.3.8 Color LUT

The Video Module supports an 8-bit per pixel color mode for interfacing to color LCD's. The color mode generates 16 different shades using the time-based dithering algorithm. The 8-bits used per pixel are encoded into 3 bits of red (R), 3 bits of green (G), and 2 bits of blue (B). The color mode requires that 8 bits (3:3:2 for R:G:B) are stored in the video buffer for every RGB-tri (one pixel) of dithered data shifted to the color LCD.

The color mode uses separate LUTs for red, green, and blue. The red LUT uses the REDSEL[31:0] control bits as the programmable LUT entries. The 32-bit REDSEL field is divided into 8 nibbles: REDSEL[31:28], REDSEL[27:24], … , REDSEL[7:4], and REDSEL[3:0]. Each nibble then corresponds to one of the 8 shades selected out of 16 possible, with each 3 pre-red bits from memory used to lookup one of these 4 shades at a time (3 to 4 mapping). The green LUT is identical to the red LUT, except the LUT uses the GREENSEL[31:0] control bits as the programmable LUT entries. The blue LUT uses the BLUESEL[15:0] control bits as the programmable LUT entries, and each LUT output nibble corresponds to one of the 4 shades selected out of 16 possible, with each 2 pre-blue bits from memory used to lookup one of these 4 shades at a time (2 to 4 mapping). Figure 17.3.6 shows a block diagram of the 8-bit per pixel color LUT logic.

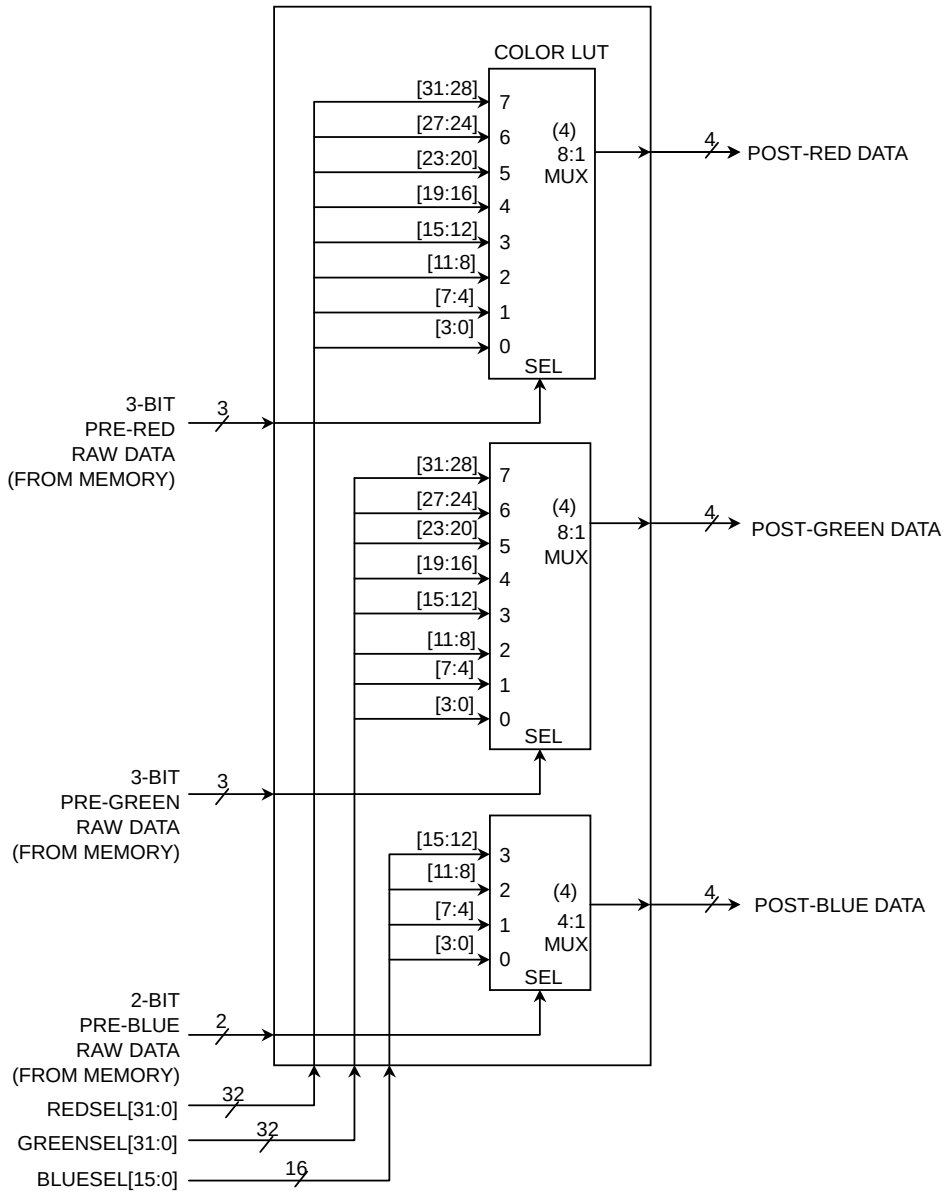


Figure 17.3.6 8-Bit Per Pixel Color LUT

17.3.9 Dithering

The Dither Selector is based on a scheme which varies the duty cycle (i.e., number of frames) for which a given pixel is turned on, giving the display an appearance of multiple shades. In order to reduce the noticeable flicker caused by turning on and off adjacent pixels at the same time, a time-based dithering algorithm is used to vary the pattern of adjacent pixels every frame.

A block diagram of the Dither Selector is shown in Figure 17.3.7. The Dither Selector contains a set of Dithering Line Counters and a set of Dithering Frame Counters. The set of Dithering Line Counters contains 4 counters which count modulo-3, modulo-4, modulo-5, and modulo-7 every horizontal line. The set of Dithering Frame Counters contains 4 counters which count modulo-3, modulo-4, modulo-5, and modulo-7 each video frame. The output of the Dithering Line Counters are used to lookup the programmable dither patterns configured in Video Control Registers 8 through 14, generating a 16-bit Enable Pattern. Each new line count looks up a different field of the appropriate Video Control Register, resulting in a time-varying and dithered pattern of on and off pixel control with the desired duty cycle. Section 17.4 lists recommended values for each of these dither patterns.

The pre-dithered video data is processed by sending 16 adjacent bits at a time through 4 identical Enable Select circuits, each of which processes 4 bits of pre-dithered data. Each 4 bits of pre-dithered data is used to lookup one of the 16 possible enable patterns, which thus determine whether or not a given pixel is turned on or off for the given frame and line. Thus every 4 bits of pre-dithered data produces 1 bit of dithered data. Table 17.3.1 shows the duty cycle ratios used by the Enable Select circuits, which correspond to the 16 possible shades (of gray or color) available.

Table 17.3.1 Dither Duty Cycles

pre-dithered data nibble	duty cycle
15	1
14	6/7
13	4/5
12	3/4
11	5/7
10	2/3
9	3/5
8	4/7
7	2/4
6	3/7
5	2/5
4	1/3
3	2/7
2	1/5
1	1/7
0	0

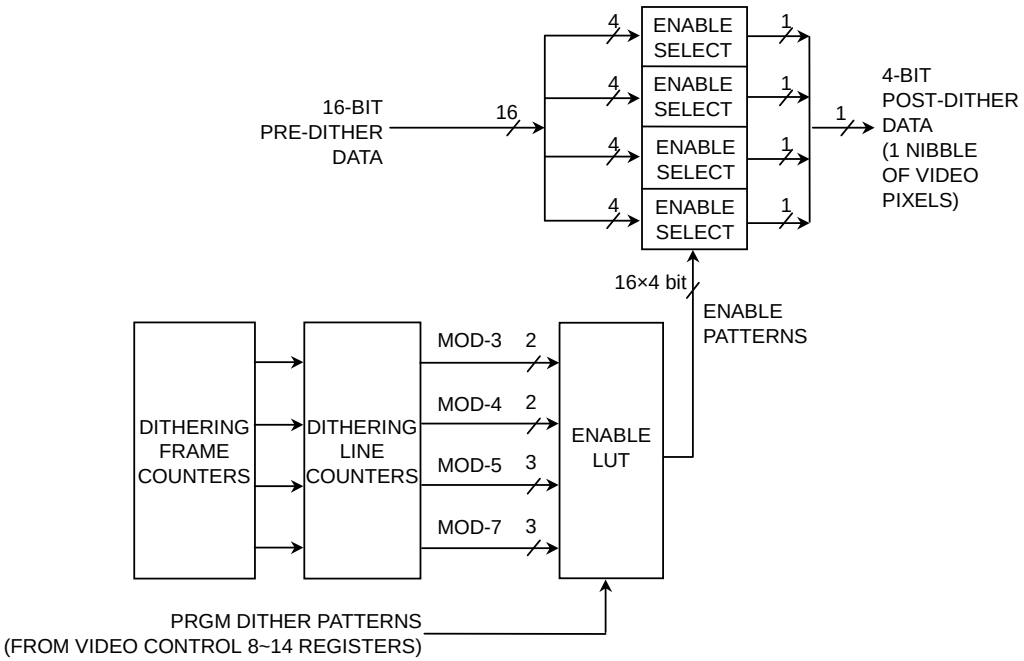


Figure 17.3.7 Dither Selector Block Diagram

17.3.10 Related Interrupts

LCDINT:

Issues an interrupt at the end of each video frame.

DFINT:

Issues an interrupt each time the video DF signal toggles.

17.4 Video Registers

17.4.1 Video Control 1 Register

OFFSET = $028:

Bit	Label	RESET	Read/Write
31-22	LINECNT[9:0]	0	R
21	LOADDLY	0	R/W
20-16	BAUDVAL[4:0]	X	R/W
15-9	VIDDONEVAL[16:0]	X	R/W
8	ENFREEZEFRAME	0	R/W
7-6	BITSEL[1:0]	0	R/W
5	DISPSPLIT	0	R/W
4	DISP8	0	R/W
3	DFMODE	0	R/W
2	INVVID	0	R/W
1	DISPON	0	R/W
0	ENVID	0	R/W

LINECNT[9:0]: read-only

These bits provide the status of the line counter.

LOADDLY:

Setting this bit will cause the FRAME and CP signals to be delayed by an additional baud clock relative to the LOAD signal. This is only required by some kinds of LCDs that have a long specification time between LOAD and CP.

BAUDVAL[4:0]:

These bits determine the rate of the CP clock. These bits must be set to a minimum of $1 and should be set to as high a value as possible to interface to the selected LCD. The following equation provides clock rate:

$$\text{CP Rate} = \frac{f_{VIDCLK}}{BAUDVAL[4:0]*2+2}$$

VIDDONEVAL[6:0]

These bits determine the time between the end of video shifting and the assertion of VIDDONE signal. These bits must be set to at least a valve of $1. Setting those bits to a valve of $1 will cause a delay from the video finishing shifting to the assertion of VIDDONE of six CLK periods (normally 162 ns). Each increment to these bits will provide an additional 16 CLK periods (normally 434 ns).

ENFREEZEFRAME:

Setting this bit will cause the Video logic to freeze at the end of the current frame. This feature is used to blip a single frame. The video logic will not start transferring data again until this bit is cleared and the ENVID bit is toggled.

BITSEL[1:0]:

These bits define the bit depth for the Video Module to generate according to the following:

$3 8-bit per pixel Color Mode

$2 4-bit per pixel Gray Scale Mode

$1 2-bit per pixel Gray Scale Mode

$0 Monochrome Mode

DISPSPLIT:

This bit should be set if interfacing to a split LCD.

DISP8:

This bit should be set if interfacing to an 8-bit non-split LCD.

DFMODE:

If this bit is a logic “0” then the DF signal will be toggled on each frame. If this bit is a logic “1” then the DF signal will toggle at the rate defined by the DFVAL (7:0) bits.

INVVID:

Setting this bit will cause the VDAT[3:0] signals to be inverted.

DISPON:

This bit is directly connected to the DISPON signal pin.

ENVID:

Setting this bit will enable the video logic. This bit should not be set until all other control bits are setup.

17.4.2 Video Control 2 Register

OFFSET = $02C: write-only

Bit	Label	RESET	Read/Write
31-22	VIDRATE[9:0]	X	W
21	Reserved		
20-12	HORZVAL[8:0]	X	W
11-10	Reserved		
9-0	LINEVAL[9:0]	X	W

VIDRATE[9:0]: write-only

These bits determine the frequency at which the LOAD pulse is generated, which in turn sets the Frame Rate. The Line Rate and Frame Rate are given by the following equations:

$$\text{Line Rate} = \frac{\texttt{CP Rate}}{\texttt{VIDRATE[9 :0]+1}}$$

$$\text{Frame Rate} = \frac{\texttt{LineRate}}{\texttt{LINEVAL[9 :0]+1}}$$

HORZVAL[8:0]: write-only

These bits define the horizontal size of the LCD according to the following equation:

HORZVAL = (HorzSize ÷ 4 – 1) for 4-bit split or non-split LCD

HORZVAL = (HorzSize ÷ 8 – 1) for 8-bit non-split LCD

LINEVAL[9:0]: write-only

These bits define the number of lines for the LCD.

LINEVAL = (#of Lines – 1) for a non-split LCD

LINEVAL = (#of Lines ÷ 2 – 1) for a split LCD

17.4.3 Video Control 3 Register

OFFSET = $030: write-only

Bit	Label	RESET	Read/Write
31-20	VIDBANK[31:20]	X	W
19-4	VIDBASEHI[19:4]	X	W
3-0	Reserved		

VIDBANK[31:20]: write-only

These bits provide the upper bits of the address from which video data is fetched from memory. These bits are concatenated with the upper and lower address counters to provide the actual address. These bits are latched at the beginning of each frame, thus it is possible to change these bits “on the fly” and the address will not change until the beginning of the next frame.

VIDBASEHI[19:4]: write-only

These bits provide the start address for the upper address counter.

17.4.4 Video Control 4 Register

OFFSET = $034: write-only

Bit	Label	RESET	Read/Write
31-24	DFVAL[7:0]	X	W
23-20	FRAMEMASKVAL[3:0]	X	W
19-4	VIDBASELO[19:0]	X	W
3-0	Reserved		

DFVAL[7:0]: write-only

These bits define the rate at which the DF signal will toggle if the DFMODE bit is set. The DF counter counts on each LOAD pulse, thus the DF Rate is given by the following equation:

$$\text{DF Rate} = \frac{\texttt{LineRate}}{\texttt{DFVAL[7:0]}+1}$$

VIDBASELO[19:4]: write-only

These bits provide the start address for the lower address counter. If a non-split LCD is used, these bits must be set to “1” plus the last address of the video buffer.

FRAMEMASKVAL[3:0]:

These bits determine the number of lines for which the VIDDONE signal will not assert following the assertion of the FRAME signal. If these bits are set to $0 then the VIDDONE signal will assert after every line completes shifting. Setting these bits to $1 will cause the VIDDONE signal to not assert after the line that immediately follows the assertion of the FRAME signal. Increases in these bits will cause subsequent lines to not assert the VIDDONE signal. Use of these bits allows the VIDDONE synchronization pulses to be generated away torn the periods after the video FRAM and DF transitions occur.

17.4.5 Video Control 5 Register

OFFSET = $038: write-only

Bit	Label	RESET	Read/Write
31-0	REDSEL[31:0]	X	W

REDSEL[31:0]: write-only

The Video Module logic generates 16 different shades using a time-based dithering algorithm. The 8-bit per pixel color mode encodes 3 bits of Red, 3 bits of Green, and 2 bits of Blue. These bits define which of the sixteen shades each of the 8 possible red combinations will choose. In other words, 3 bits are expanded into 4 bits using these values as the lookup table.

3-BIT PRE-RED DATA	4-BIT POST-RED DATA
111	REDSEL[31:28]
110	REDSEL[27:24]
101	REDSEL[23:20]
100	REDSEL[19:16]
011	REDSEL[15:12]
010	REDSEL[11:8]
001	REDSEL[7:4]
000	REDSEL[3:0]

17.4.6 Video Control 6 Register

OFFSET = $03C: write-only

Bit	Label	RESET	Read/Write
31-0	GREENSEL[31:0]	X	W

GREENSEL[31:0]: write-only

The Video Module logic generates 16 different shades using a time-based dithering algorithm. The 8-bit per pixel color mode encodes 3 bits of Red, 3 bits of Green, and 2 bits of Blue. These bits define which of the sixteen shades each of the 8 possible green combinations will choose. In other words, 3 bits are expanded into 4 bits using these values as the lookup table.

3-BIT PRE-GREEN DATA	4-BIT POST-GREEN DATA
111	GREENSEL[31:28]
110	GREENSEL[27:24]
101	GREENSEL[23:20]
100	GREENSEL[19:16]
011	GREENSEL[15:12]
010	GREENSEL[11:8]
001	GREENSEL[7:4]
000	GREENSEL[3:0]

17.4.7 Video Control 7 Register

OFFSET = $040: rite-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	BLUESEL[15:0]	X	W

BLUESEL[15:0]: write-only

The Video Module logic generates 16 different shades using a time-based dithering algorithm. The 8-bit per pixel color mode encodes 3 bits of Red, 3 bits of Green, and 2 bits of Blue. These bits define which of the sixteen shades each of the 4 possible blue combinations will choose. In other words, 2 bits are expanded into 4 bits using these values as the lookup table. The BLUESEL values are also used as the LUT for the 2-bit gray scale mode.

2-BIT PRE-BLUE DATA	4-BIT POST-BLUE DATA
11	BLUESEL[15:12]
10	BLUESEL[11:8]
01	BLUESEL[7:4]
00	BLUESEL[3:0]

17.4.8 Video Control 8 Register

OFFSET = $044: write-only

Bit	Label	RESET	Read/Write
31-12	Reserved		
11-0	PAT2_3L[11:0]	X	W

PAT2_3 [11:0]: write-only

These bits define the (2 out of 3) patterning for the dithering algorithm. The (1 out of 3) patterning for the dithering algorithm is the inverse of the (2 out of 3) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT2_3[11:8]	0111
1	PAT2_3[7:4]	1101
2	PAT2_3[3:0]	1010

17.4.9 Video Control 9 Register

OFFSET = $048: write-only

Bit	Label	RESET	Read/Write
31-16	PAT3_4[15:0]	X	W
15-0	PAT2_4[15:0]	X	W

PAT3_4[15:0]: write-only

These bits define the (3 out of 4) patterning for the dithering algorithm.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT3_4[15:12]	0111
1	PAT3_4[11:8]	1101
2	PAT3_4[7:4]	1011
3	PAT3_4[3:0]	1110

PAT2_4[15:0]: write-only

These bits define the (2 out of 4) patterning for the dithering algorithm.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT2_4[15:12]	1010
1	PAT2_4[11:8]	0101
2	PAT2_4[7:4]	1010
3	PAT2_4[3:0]	0101

17.4.10 Video Control 10 Register

OFFSET = $04C: write-only

Bit	Label	RESET	Read/Write
31-20	Reserved		
19-0	PAT4_5[19:0]	X	W

PAT4_5[19:0]: write-only

These bits define the (4 out of 5) patterning for the dithering algorithm. The (1 out of 5) patterning for the dithering algorithm is the inverse of the (4 out of 5) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT4_5[19:16]	0111
1	PAT4_5[15:12]	1101
2	PAT4_5[11:8]	1111
3	PAT4_5[7:4]	1011
4	PAT4_5[3:0]	1110

17.4.11 Video Control 11 Register

OFFSET = $050: write-only

Bit	Label	RESET	Read/Write
31-20	Reserved		
19-0	PAT3_5[19:0]	X	W

PAT3_5[19:0]: write-only

These bits define the (3 out of 5) patterning for the dithering algorithm. The (2 out of 5) patterning for the dithering algorithm is the inverse of the (3 out of 5) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT3_5[19:16]	0111
1	PAT3_5[15:12]	1010
2	PAT3_5[11:8]	0101
3	PAT3_5[7:4]	1010
4	PAT3_5[3:0]	1101

17.4.12 Video Control 12 Register

OFFSET = $054: write-only

Bit	Label	RESET	Read/Write
31-28	Reserved		
27-0	PAT6_7[27:0]	X	W

PAT6_7(27:0): write-only

These bits define the (6 out of 7) patterning for the dithering algorithm. The (1 out of 7) patterning for the dithering algorithm is the inverse of the (6 out of 7) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT6_7[27:24]	1111
1	PAT6_7[23:20]	1011
2	PAT6_7[19:16]	1111
3	PAT6_7[15:12]	1101
4	PAT6_7[11:8]	1111
5	PAT6_7[7:4]	1110
6	PAT6_7[3:0]	0111

17.4.13 Video Control 13 Register

OFFSET = $058: write-only

Bit	Label	RESET	Read/Write
31-28	Reserved		
27-0	PAT5_7[27:0]	X	W

PAT5_7[27:0]: write-only

These bits define the (5 out of 7) patterning for the dithering algorithm. The (2 out of 7) patterning for the dithering algorithm is the inverse of the (5 out of 7) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT5_7[27:24]	0111
1	PAT5_7[23:20]	1011
2	PAT5_7[19:16]	0101
3	PAT5_7[15:12]	1010
4	PAT5_7[11:8]	1101
5	PAT5_7[7:4]	1110
6	PAT5_7[3:0]	1111

17.4.14 Video Control 14 Register

OFFSET = $05C: write-only

Bit	Label	RESET	Read/Write
31-28	Reserved		
27-0	PAT4_7[27:0]	X	W

PAT4_7[27:0]: write-only

These bits define the (4 out of 7) patterning for the dithering algorithm. The (3 out of 7) patterning for the dithering algorithm is the inverse of the (4 out of 7) patterning.

LINE	PATTERN FIELD	RECOMMENDED PATTERN
0	PAT4_7[27:24]	1011
1	PAT4_7[23:20]	1001
2	PAT4_7[19:16]	1101
3	PAT4_7[15:12]	1100
4	PAT4_7[11:8]	0110
5	PAT4_7[7:4]	0110
6	PAT4_7[3:0]	0011

18. Electrical Characteristics

18.1 Electrical Characteristics of TMPR3911BU/BXB

18.1.1 Absolute maximum ratings (TMPR3911BU/BXB)

V_{SS} = 0 V (GND)

Parameter	Symbol	Rating	Unit
Supply voltage (I/O)	V_{DD}	V_{SS} – 0.5 to 4.5	V
Supply voltage (Internal Logic)	V_{DDL}	V_{SS} – 0.5~V_{DDH}	V
Input voltage	V_{IN}	V_{SS} – 0.5 to V_{DDH} + 0.5	V
Storage temperature	T_{STG}	–55 to 125	°C
Maximum dissipation (Ta = 70°C)	P_D	1	W

Note: Using an LSI at specifications higher than the maximum ratings can cause permanent damage to the LSI. For normal operation, use under the recommended operating conditions. Exceeding the recommended operating conditions may affect the reliability of the LSI.

18.1.2 Recommended operating conditions (TMPR3911BU/BXB)

V_{SS} = 0 V (GND)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Power Supply voltage	V_{DDH}		3.0	3.3	3.6	V
Power Supply voltage	V_{DDL}		2.4	2.6	2.8	3.3
Input voltage (*1)	V_{IH1}	V_{DDH} = 3.6 V	$V_{DDH} \times 0.8$	—	V_{DDH} + 0.3	V
	V_{IL1}	V_{DDH} = 3.0 V	–0.3	—	$V_{DDH} \times 0.1$	V
Input voltage (*2)	V_{IH2}	V_{DDH} = 3.6 V	2.4	—	V_{DDH} + 0.3	V
	V_{IL2}	V_{DDH} = 3.0 V	–0.3	—	0.6	V
SYSCLK Input Frequency	f_{SYSCLK}		4.5	—	7.3728	MHz
C32K Input Frequency	f_{32K}		—	32.768	—	KHz
Operating temperature	T_{OPR}		–20	—	70	°C

(*1) SYSCLKIN, DCLKIN, MBUSINT

(*2) Other inputs

18.1.3 DC characteristics (TMPR3911BU/BXB)

(Ta = –20°C to 70°C, V_{DDH} = 3.3 V ± 0.3 V, VDDL = 2.6 V ± 0.2 V)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating current (1)	IDD_{VDDH}	V_{IN} = V_{DDH} or V_{SS} I_{OH} = I_{OL}=0 mA f_{SYSCLK} = 7.3728 MHz	—	45	50	mA
	IDD_{VDDL}			88	95	
Doze Mode current (2)	$IDDD_{VDDH}$	V_{IN} = V_{DDH} or V_{SS} I_{OH} = I_{OL} = 0 mA f_{SYSCLK} = 7.3728 MHz	—	15	20	mA
	$IDDD_{VDDL}$		—	30	35	
Sleep Mode current (3)	$IDDS1_{VDDH}$	V_{IN} = V_{DDH} or V_{SS} I_{OH} = I_{OL} = 0 mA	—	1	10	µA
	$IDDS1_{VDDL}$		—	19	110	
Static current (4)	$IDDS2_{VDDH}$	V_{IN} = V_{DDH} or V_{SS} I_{OH} = I_{OL} = 0 mA	—	1	10	µA
	$IDDS2_{VDDL}$		—	9	90	
Input Leakage current	I_{IN}	V_{IN} = V_{DDH} or V_{SS}	–10	—	10	µA
Output voltage (5)	V_{OH1}	V_{DDH} = 3.0 V, I_{OH} = –4 mA	V_{DD} – 0.6	—	—	V
	V_{OL1}	V_{DDH} = 3.0 V, I_{OL} = 4 mA	—	—	+0.4	V
Output voltage (6)	V_{OH2}	V_{DDH} = 3.0 V, I_{OH} = –8 mA	V_{DD} – 0.6	—	—	V
	V_{OL2}	V_{DDH} = 3.0 V, I_{OL} = 8 mA	—	—	+0.4	V
Output voltage (7)	V_{OH3}	V_{DDH} = 3.0 V, I_{OH} = –16 mA	V_{DD} – 0.6	—	—	V
	V_{OL3}	V_{DDH} = 3.0 V, I_{OL} = 16 mA	–	—	+0.3	V
Output voltage (8)	V_{OH4}	V_{DDH} = 3.0 V, I_{OH} = –24 mA	V_{DD} – 0.6	—	—	V
	V_{OL4}	V_{DDH} = 3.0 V, I_{OL} = 24 mA	—	—	+0.4	V
Input current (Pull-down resistor)	I_{IHP}	V_{DD} = Max V_{IN} = V_{DD}	20	—	120	µA

(1) Based on TOSHIBA original benchmark program measurement.

(2) CPU is stopped (STOPSPU = 1). Internal peripherals are running.

(3) RTC is running. Internal peripherals are stopped.

(4) RTC is stopped. Internal peripherals are stopped.

(5) D[31;0], RAS0*, RAS1*, DCS0*, DCKE, DQMH, DQML, DREQ*, DGRNT*, BC32K, VDAT[3:0], CP, LOAD, DF, FRAME, DISPON, VIDDONE, PWRCS, TXD, RXD, CS3*-CS0*, CHIFS, CHICLK, CHIDOUT, CHIDIN, IO[6;0], SPICLK, SPIOUT, SPIIN, SIBSYNC, SIBDOUT, SIBMCLK, SIBCLK, RXPWR, IROUT, CARD1WAIT*, CARD2WAIT*, MIOX[2:0]

(6) A[12:0], ALE, RD*, WE*, CAS3*-CAS0*, CARDREG*, CARDIOWR*, CARDIORD*, CARD1CSL*, CARD1CSH*, CARD2CSL*, CARD2CSH*

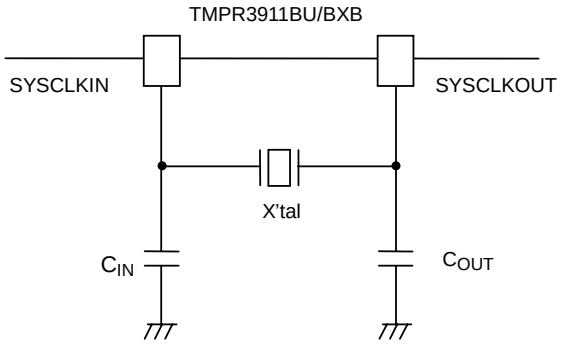
(7) DCLKOUT

(8) MBUSCLK, MBUSDATA

18.1.4 Crystal oscillator characteristics (TMPR3911BU/BXB)

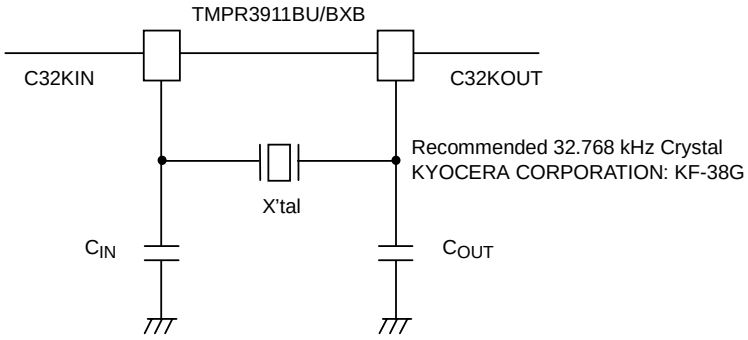
18.1.4.1 Crystal oscillator conditions

(1) 7.3728 MHz crystal



Parameter	Symbol	Recommended value		Unit
		Min	Max	
Crystal Oscillator frequency	f_{IN}	4.5	7.3728	MHz
External capacitors	C_{IN}, C_{OUT}	10	33	pF

(2) 32.768 kHz crystal



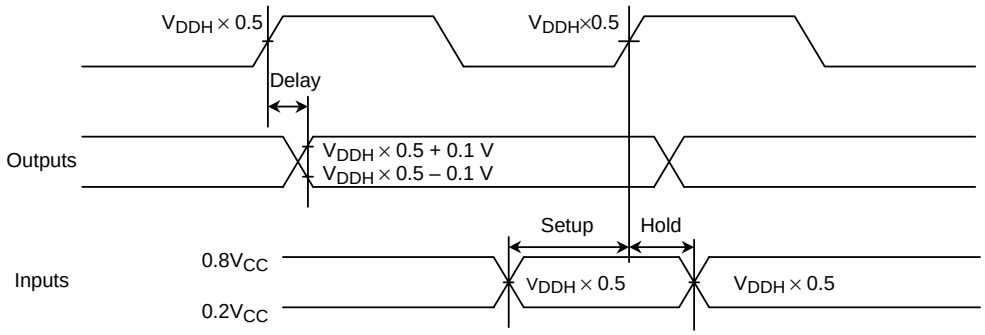
Parameter	Symbol	Recommended value		Unit
		Min	Max	
External capacitors	C_{IN}, C_{OUT}	10	33	pF

18.1.4.2 Electrical specifications

(V_{SS} = 0 V, V_{DD} = 3.3 V)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Crystal stabilization time 7.3728 MHz	$T_{STA-59M}$	T.B.D.	—	—	T.B.D.	ms
Crystal stabilization time 32 kHz	$T_{STA-32k}$	T.B.D.	—	—	T.B.D.	s

18.1.5 TMPR3911BU/BXB Timing



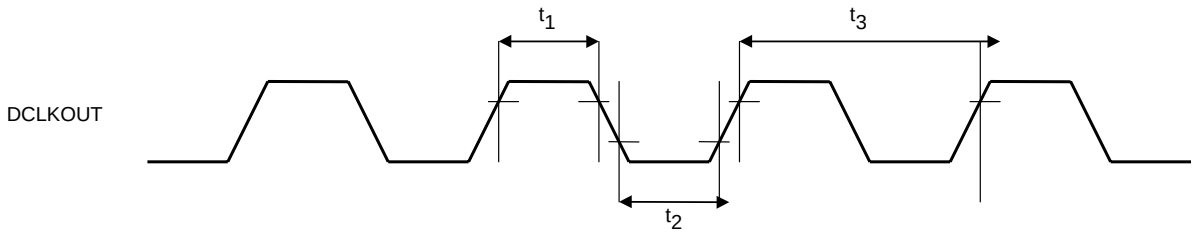
18.1.6 AC characteristics (TMPR3911BU/BXB)

The following operating conditions apply to all values specified in this section.
T_a = –20 ºC to 70 ºC, V_{DDH} = 3.3 ± 0.3 V, V_{DDL} = 2.6 ± 0.2 V, External Capacitance = 40 pF

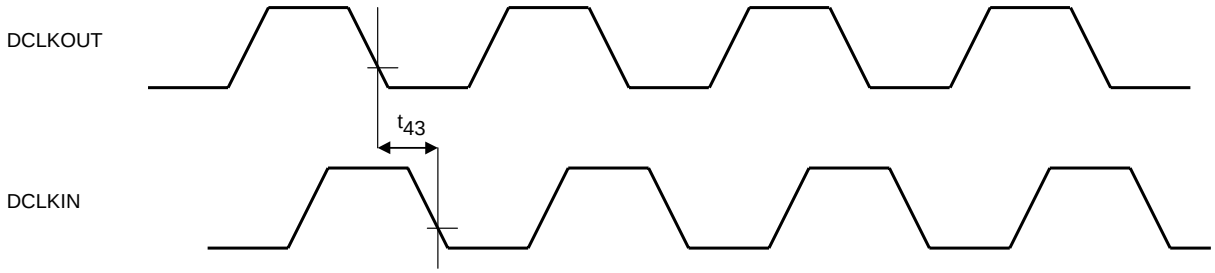
<Memory Interface>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	DCLKOUT high time	—	7.4	—	ns
t_2	DCLKOUT low time	—	7.4	—	ns
t_3	DCLKOUT period	—	16.9	—	ns
t_4	Delay DCLKOUT to ALE	Rising	—	5	ns
t_5	Delay DCLKOUT to ALE	Falling	—	3	ns
t_6	Delay DCLKOUT to A[12:0]	—	0	8	ns
t_7	Delay DCLKOUT to D[31:16]	—	—	8	ns
t_8	Delay DCLKOUT to D[15:0]	—	1	8	ns
t_9	Delay DCLKOUT to CS3-0*	Rising	—	10	ns
t_{10}	Delay DCLKOUT to CS3-0*	Falling	—	10	ns
t_{11}	Delay DCLKOUT to RD*	Rising	—	8	ns
t_{12}	Delay DCLKOUT to RD*	Falling	—	7	ns
t_{13}	Delay DCLKOUT to WE*	Rising	0	5	ns
t_{14}	Delay DCLKOUT to WE*	Falling	—	4	ns
t_{15}	Delay DCLKOUT to CAS3-0*	Rising	0	2.5	ns
t_{16}	Delay DCLKOUT to CAS3-0*	Falling	—	2.5	ns
t_{17}	Delay DCLKOUT to CARDxCSx*	Rising	—	9	ns
t_{18}	Delay DCLKOUT to CARDxCSx*	Falling	—	8	ns
t_{19}	Delay DCLKOUT to CARDDIR*	Rising	—	12	ns
t_{20}	Delay DCLKOUT to CARDDIR*	Falling	—	11	ns
t_{21}	Delay DCLKOUT to CARDREG*	Rising	—	9	ns
t_{22}	Delay DCLKOUT to CARDREG*	Falling	—	10	ns
t_{23}	Delay DCLKOUT to CARDIORD*	Rising	—	10	ns
t_{24}	Delay DCLKOUT to CARDIORD*	Falling	—	9	ns
t_{25}	Delay DCLKOUT to CARDIOWR*	Rising	—	9	ns
t_{26}	Delay DCLKOUT to CARDIOWR*	Falling	—	9	ns
t_{27}	Delay DCLKOUT to RAS0*	Rising	0	6	ns
t_{28}	Delay DCLKOUT to RAS0*	Falling	—	6	ns
t_{29}	Delay DCLKOUT to RAS1*	Rising	1	8	ns
t_{30}	Delay DCLKOUT to RAS1*	Falling	1	9	ns
t_{31}	Delay DCLKOUT to DQMH/L	Rising	0	8	ns
t_{32}	Delay DCLKOUT to DQMH/L	Falling	0	9	ns
t_{33}	Delay DCLKOUT to DCS0*	Rising	1	7	ns
t_{34}	Delay DCLKOUT to DCS0*	Falling	1	6	ns
t_{35}	Delay DCLKOUT to DCKE	Rising	0	8	ns
t_{36}	Delay DCLKOUT to DCKE	Falling	0	8	ns
t_{37}	Delay DCLKOUT to MCS3-0*	Rising	—	10	ns
t_{38}	Delay DCLKOUT to MCS3-0*	Falling	—	10	ns
t_{39}	D[31:16] to DCLKIN Setup time	—	1	—	ns
t_{40}	DCLKIN to D[31:16] Hold time	—	3.5	—	ns
t_{41}	D[15:0] to DCLKIN Setup time	—	1	—	ns
t_{42}	DCLKIN to D[15:0] Hold time	—	3.5	—	ns
t_{43}	DCLKOUT to DCLKIN Board Delay time	—	0	3	ns

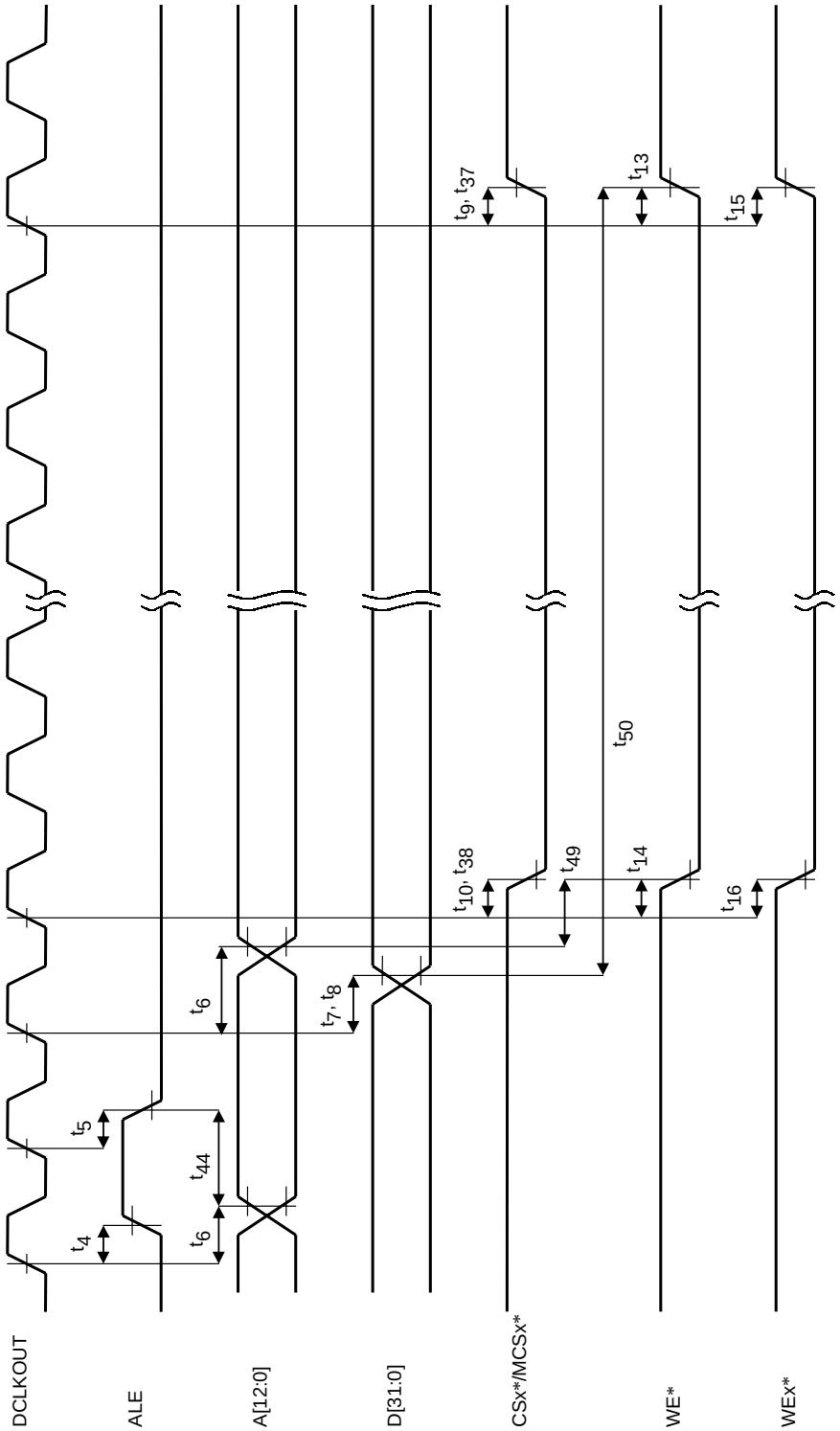
Symbol	Parameter	Calculation
t_{44}	Address valid to ALE falling edge	$t_3 + t_5 - t_6$
t_{45}	ALE falling edge to Address hold	$t_3 + t_6 - t_5$
t_{46}	Access time for read	2*ACCVAL1* $t_3 - (t_{10}$ or $t_{38}) - (t_{39}$ or $t_{41})$
t_{47}	Write pulse width	2*ACCVAL1* $t_3 + t_{13}$ or t_{14}
t_{48}	Address valid to CSX/MCSX* falling edge	At least $t_3 - t_6 + (t_{10}$ or $t_{38})$
t_{49}	Address valid to WE* falling edge	$t_3 - t_6 + t_{14}$
t_{50}	DATA valid to WE* rising edge	2*ACCVAL1* $t_3 + t_3 - (t_7$ or $t_8) + t_{13}$
t_{51}	WE* rising edge to DATA hold	At least t_3
t_{52}	Access time for 2nd/3rd/4th read	2*ACCVAL2* $t_3 + t_3 - t_6 - (t_{39}$ or $t_{41})$
t_{53}	Access time for read (Card)	2*((2*CARDxACCVAL) – 1)* $t_3 - t_{18} - t_{39}$
t_{54}	Address valid to CARDCS* falling edge	3* $t_3 - t_6 + t_{18}$
t_{55}	Write pulse width (Card)	2*((2*CARDxACCVAL) – 1)* $t_3 + t_{13} - t_{14}$
t_{56}	DATA valid to WE* rising edge (Card)	2*((2*CARDxACCVAL) – 1)* t_3 + 3* $t_3 - t_7 + t_{13}$
t_{57}	WE* rinsing edge to DATA hold(Card)	At least t_3
t_{58}	DATA valid to IOWR* rising edge	2*((2*CARDIOxACCVAL) – 1)* t_3 + 3* $t_3 - t_7 + t_{25}$
t_{59}	IOWR* rinsing edge to DATA hold	At least t_3
t_{60}	IOWR* pulse width	2*((2*CARDxIOACCVAL) – 1)* t_3 – 2* $t_3 - t_{25} - t_{26}$
t_{61}	RAS* falling edge to CAS* falling edge	(4 or 6)* $t_3 + t_{16} - (t_{28}$ or $t_{30})$
t_{62}	Address valid to RAS* falling edge	$t_3 + (t_{28}$ or $t_{30}) - t_6$
t_{63}	Address valid to CAS* falling edge	$t_3 + t_{16} - t_6$
t_{64}	CAS* pulse width	(1 or 2)* $t_3 + t_{15} - t_{16}$
t_{65}	DATA valid to CAS* rising edge	(4 or 5)* $t_3 + t_{15} - (t_7$ or $t_8)$
t_{66}	CAS* rising edge to DATA hold	At least t_3
t_{67}	Control signal active to DCKE	$t_3 + t_{35}$ – (Delay for each signal)
t_{68}	DATA valid to CAS* rising edge (In-Page)	3* $t_3 + t_{15} - (t_7$ or $t_8)$
t_{69}	Access time for read (IO Card)	2*((2*CARDxIOACCVAL) – 1)* t_3 – 2* $t_3 - t_{39} - t_{24}$



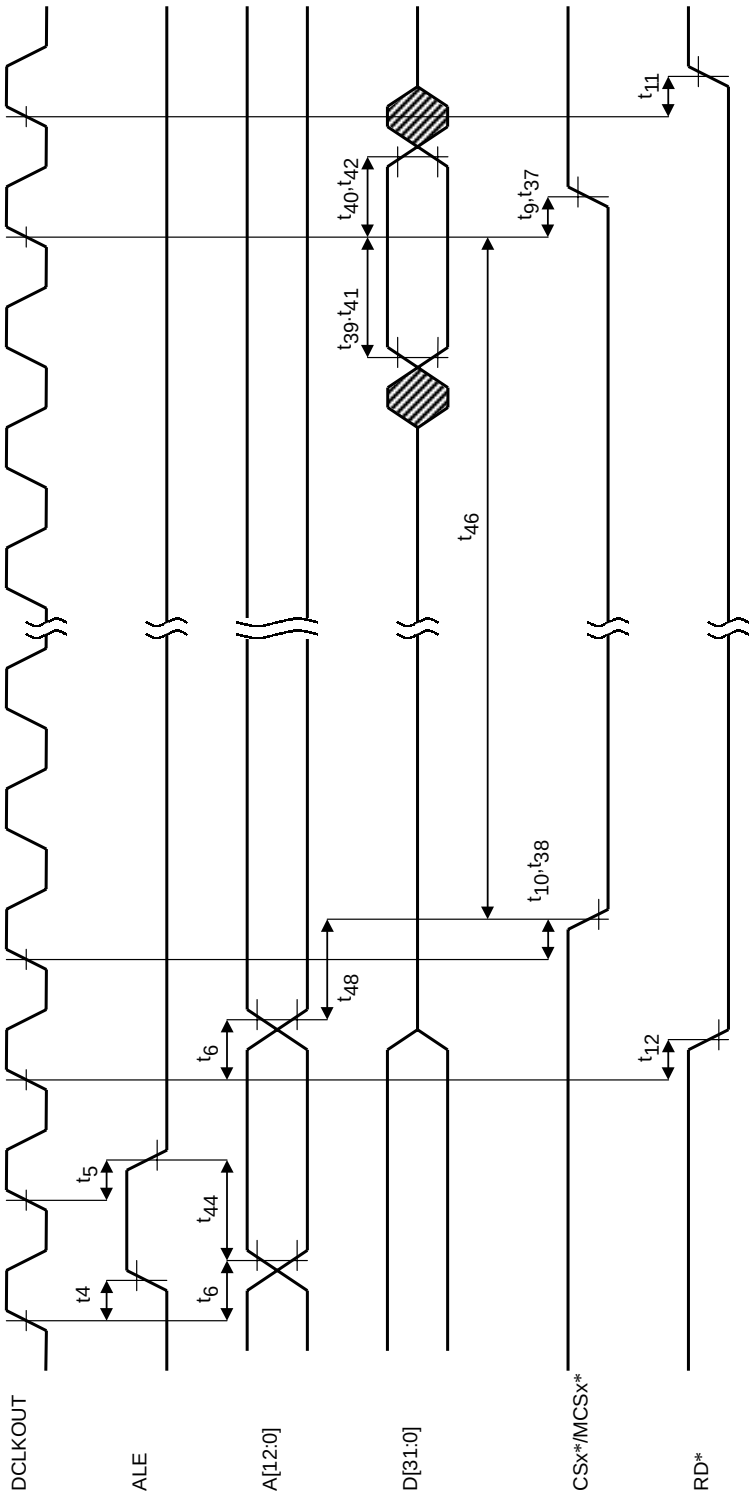
DCLKOUT Timing



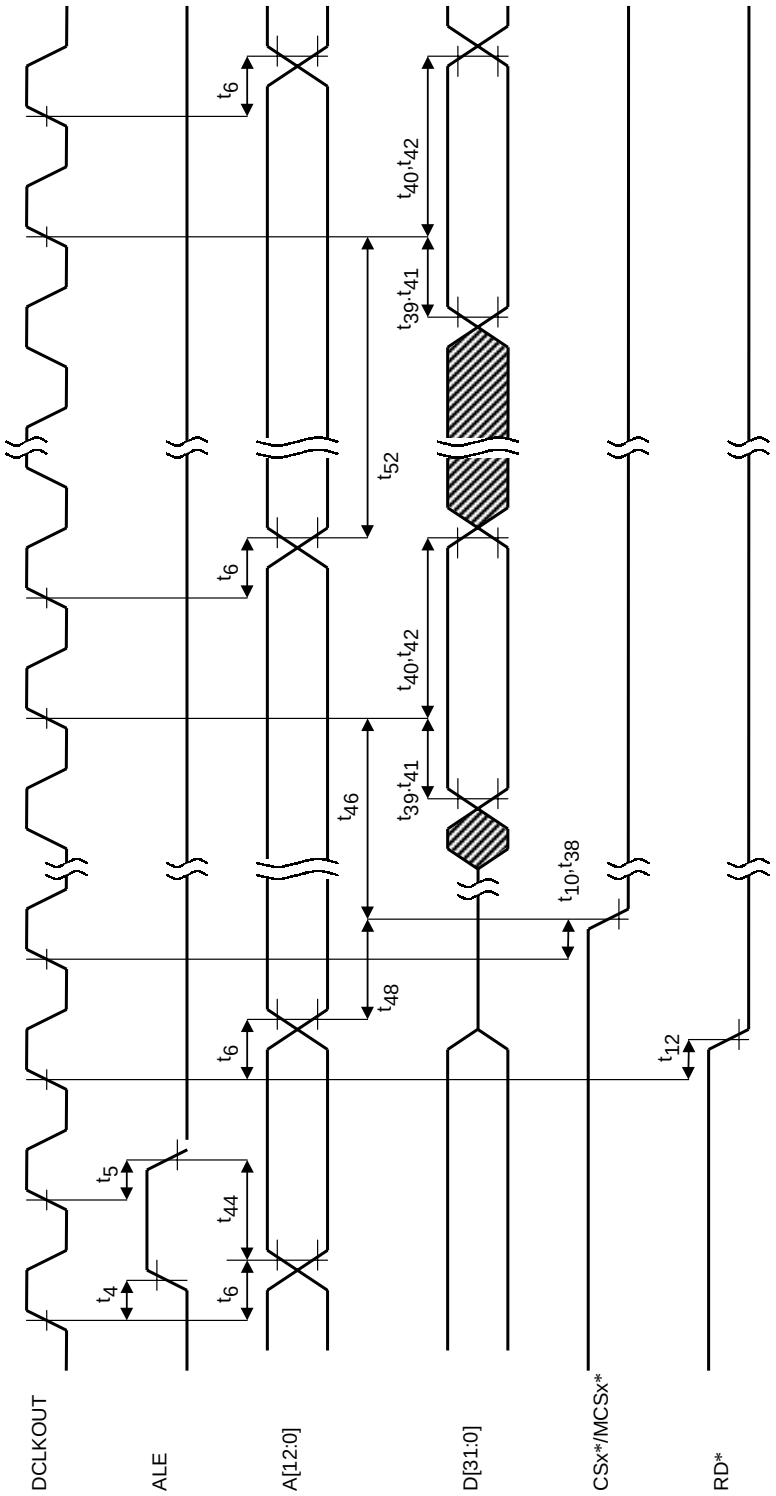
DCLKOUT to DCLKIN Timing



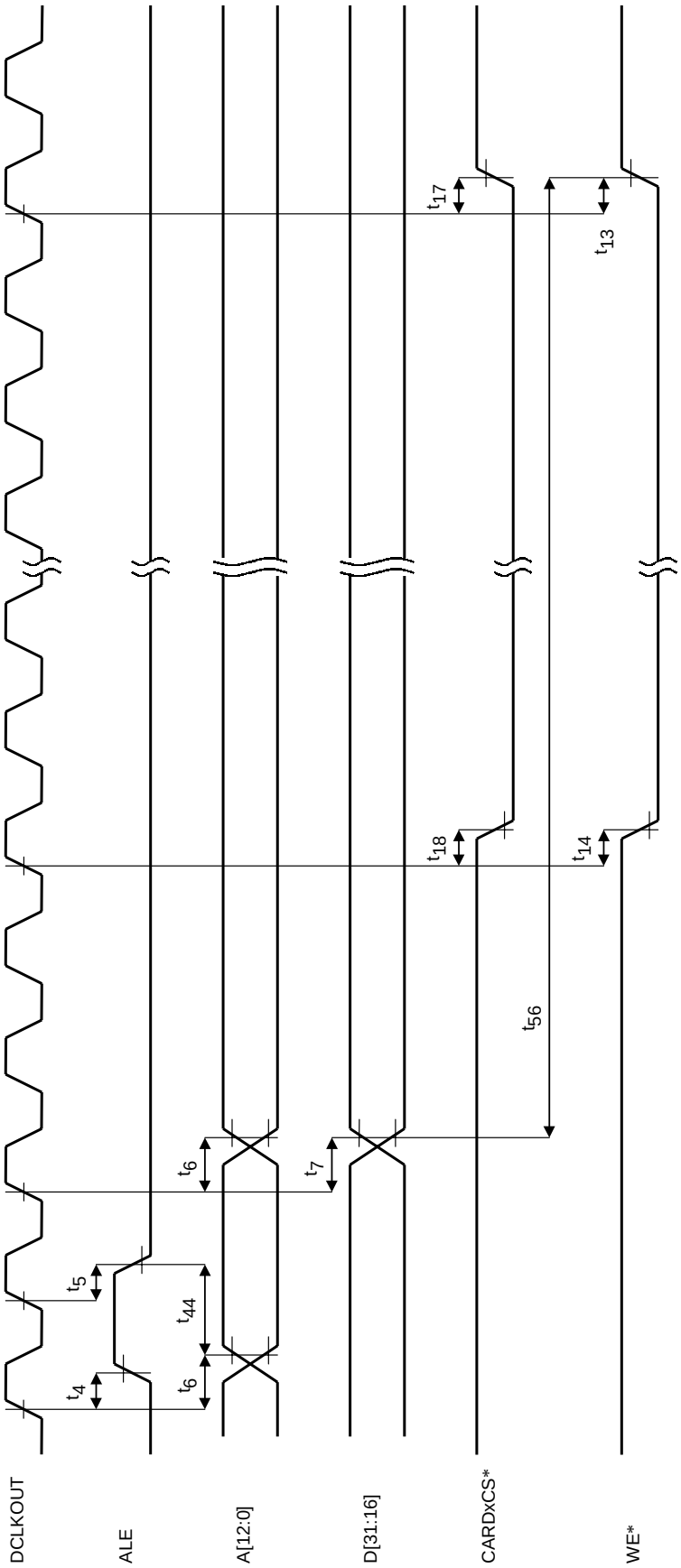
CS/MCS Device Access Timing(1) - Write Operation



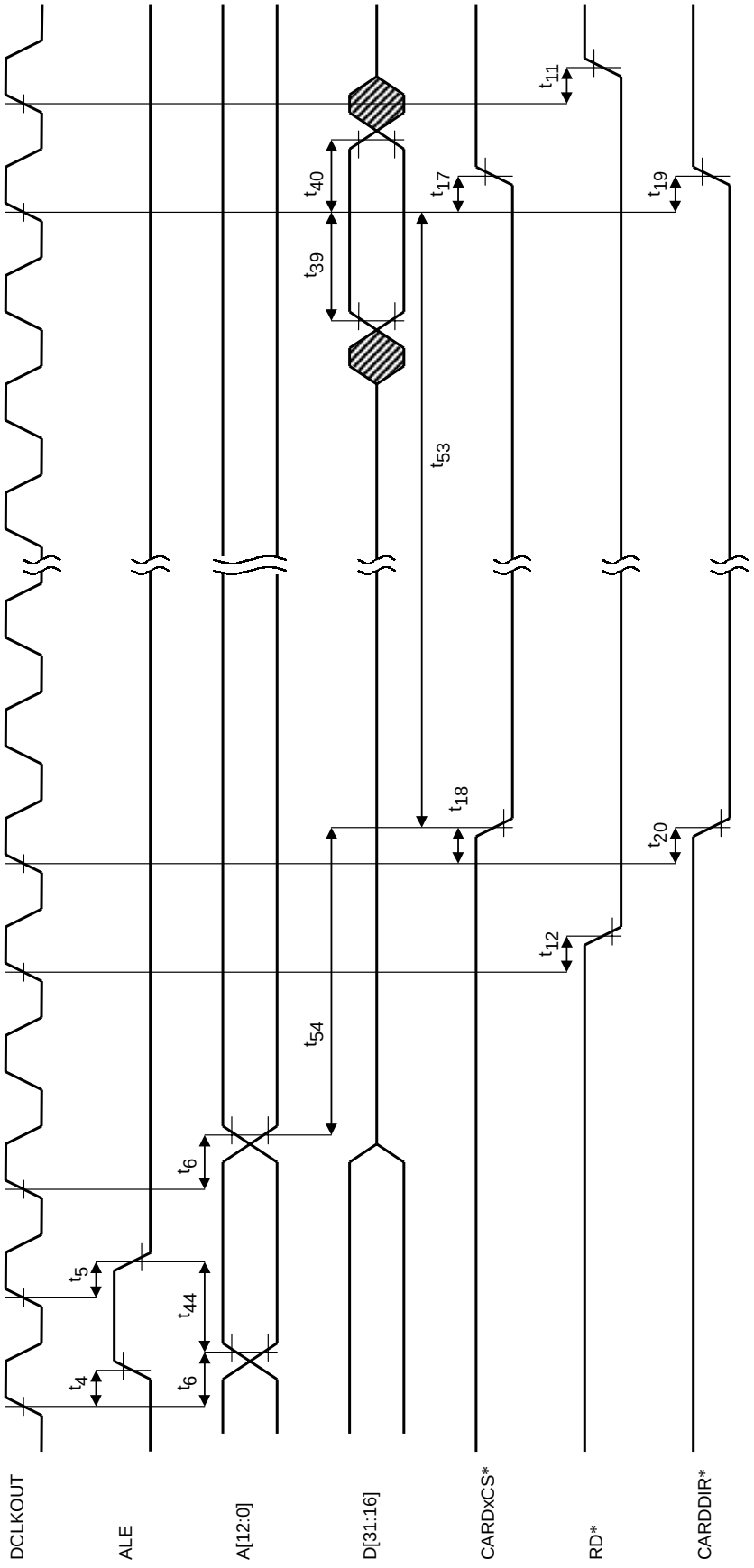
CS/MCS Device Access Timing(2) - Read Operation



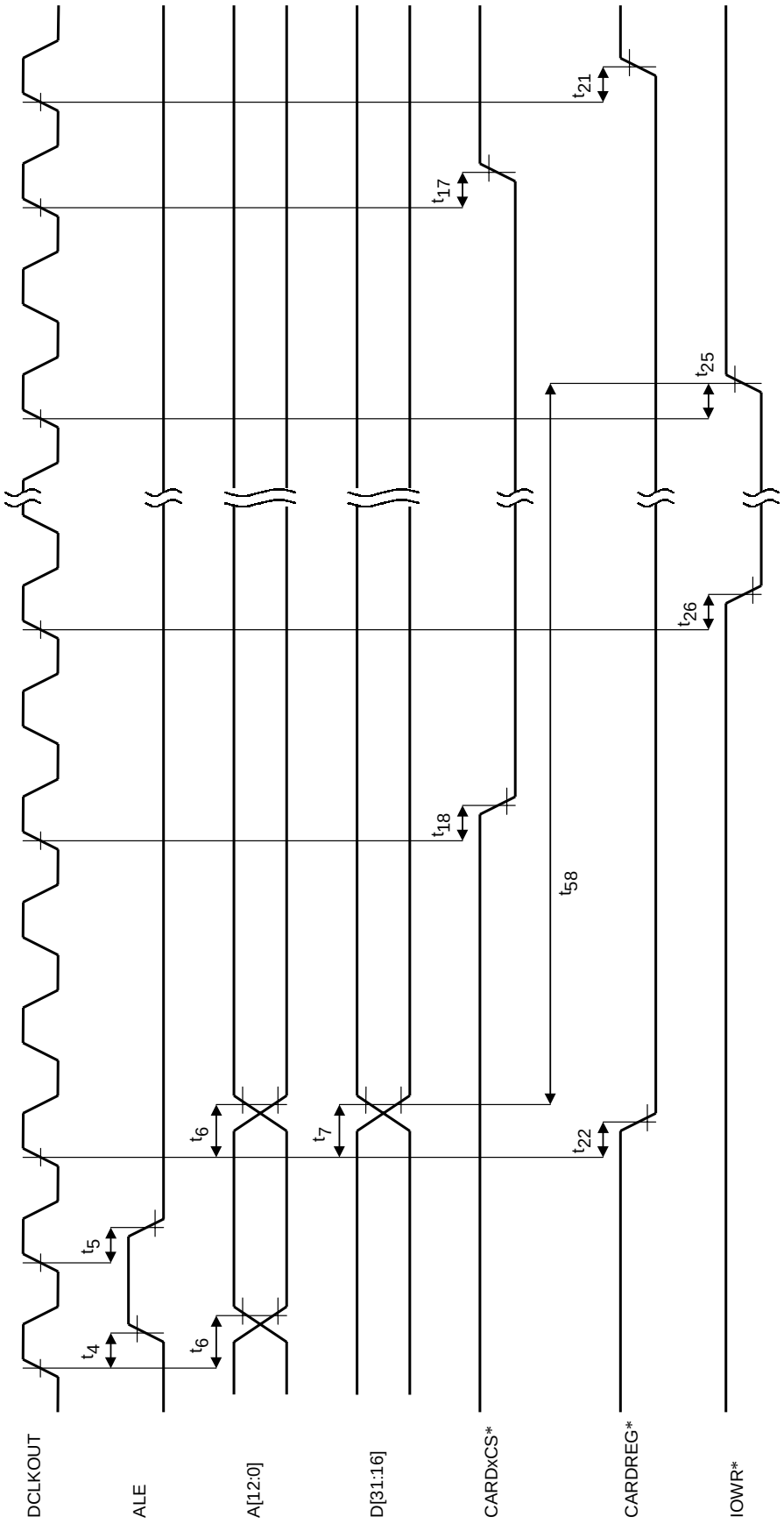
CS/MCS Device Access Timing(3) - Burst Read Operation with Page Mode Enabled



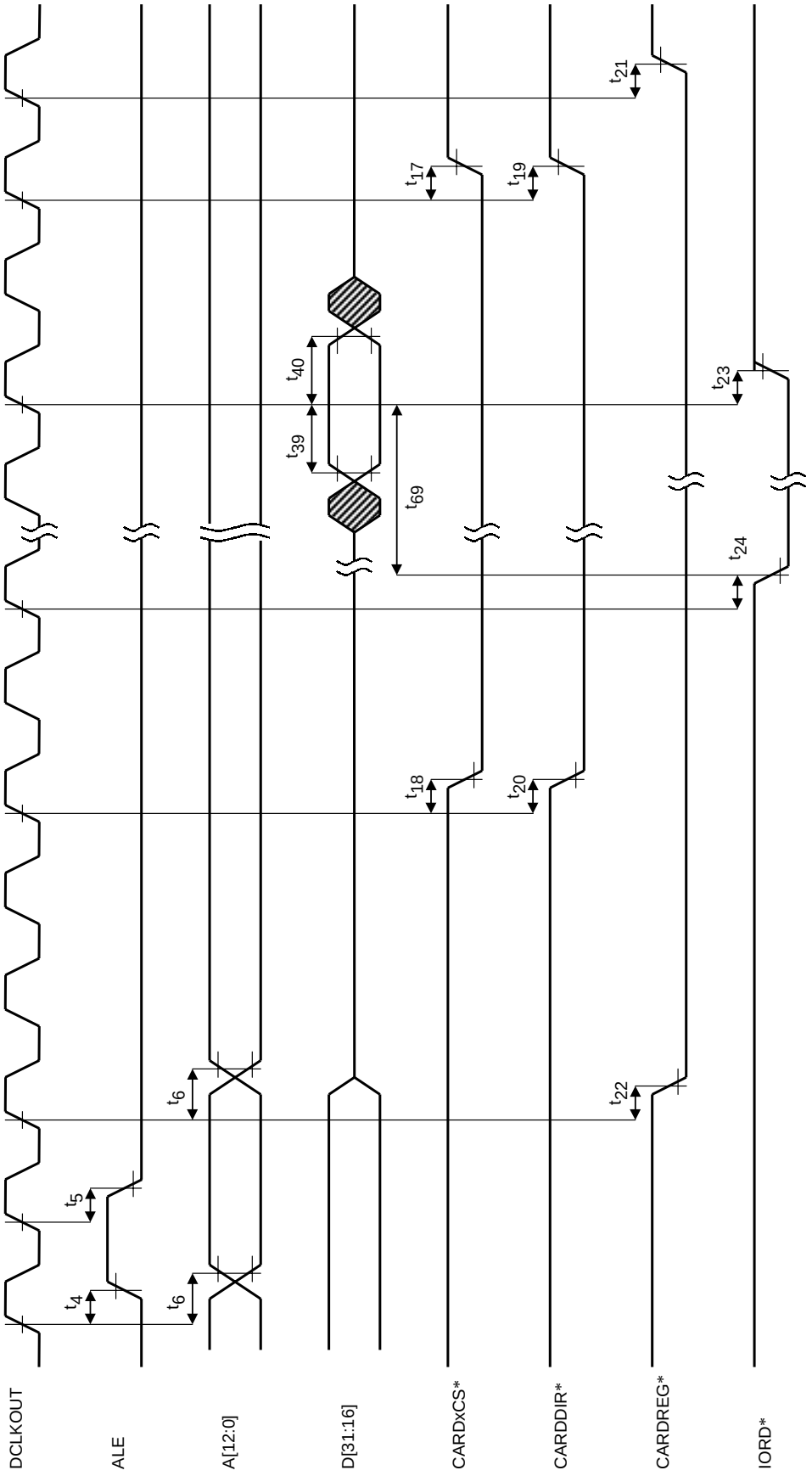
CARD Device Memory Access Timing(1) - Write Operation



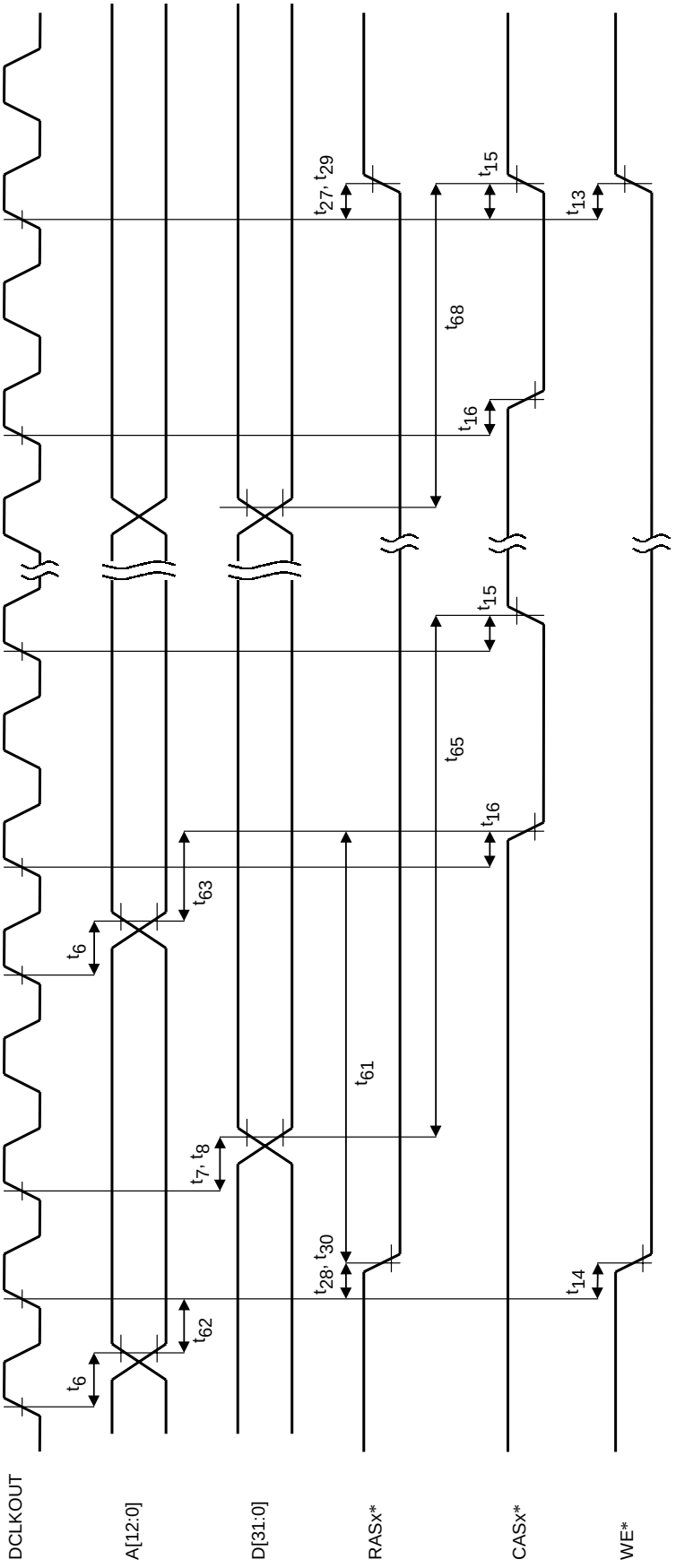
CARD Device Memory Access Timing(2) - Read Operation



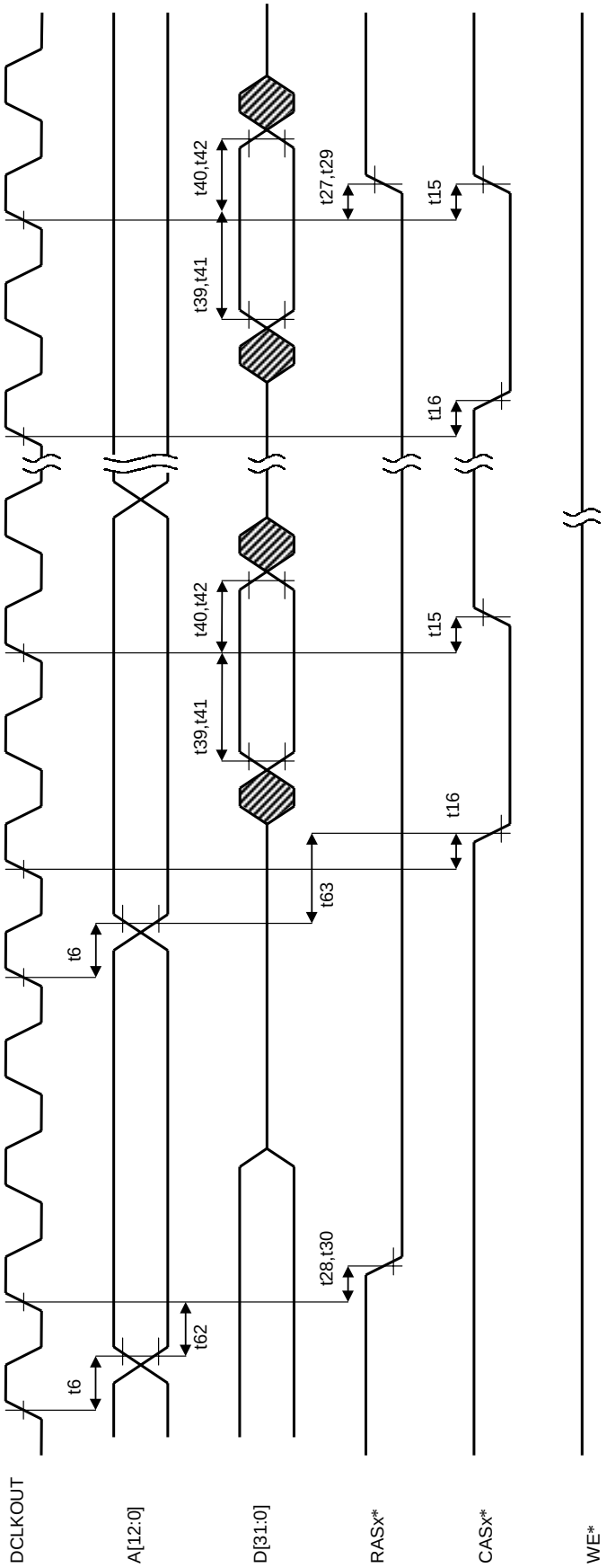
CARD Device I/O Access Timing(1) - Write Operation



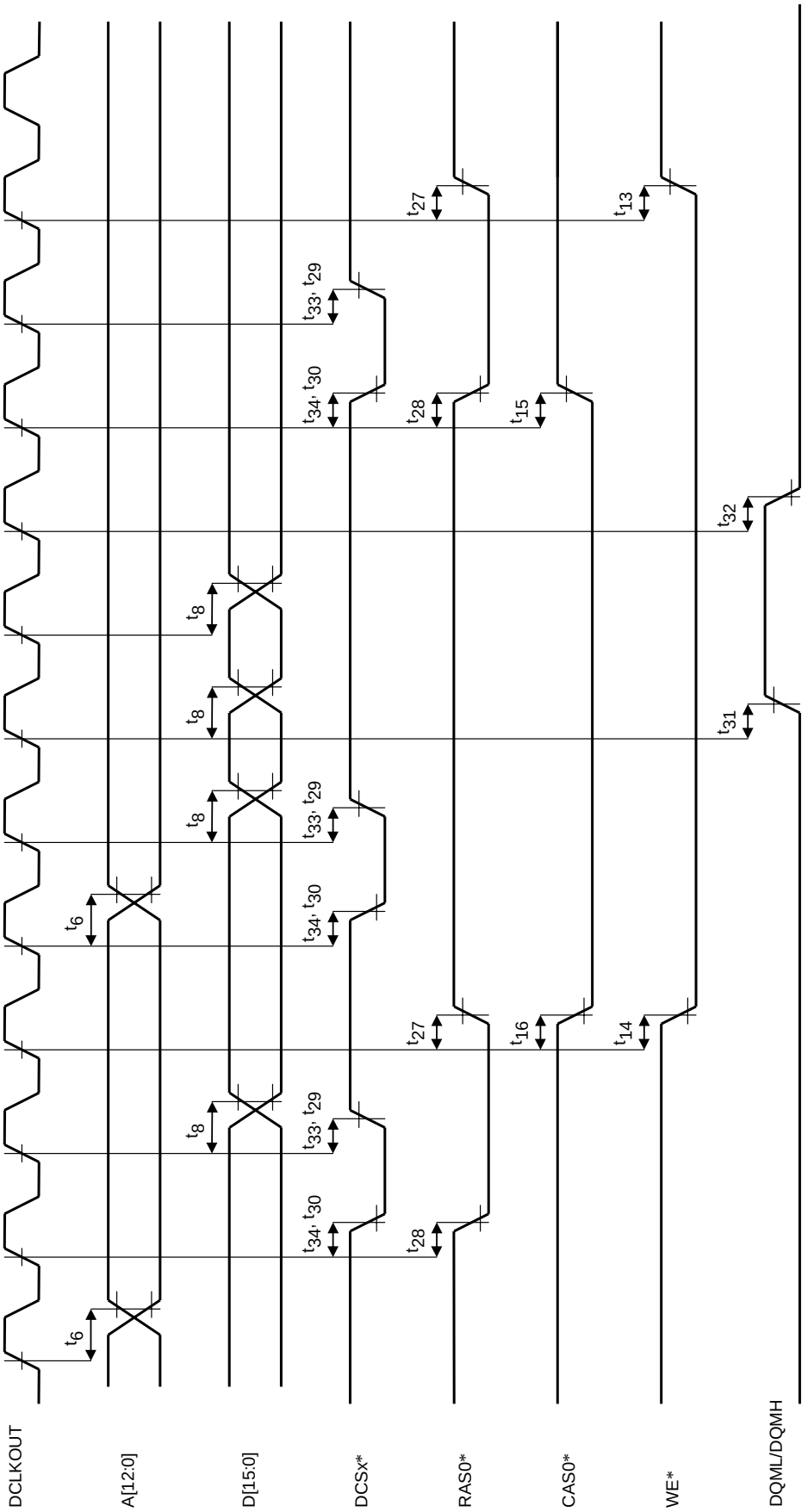
CARD Device I/O Access Timing(2) - Read Operation



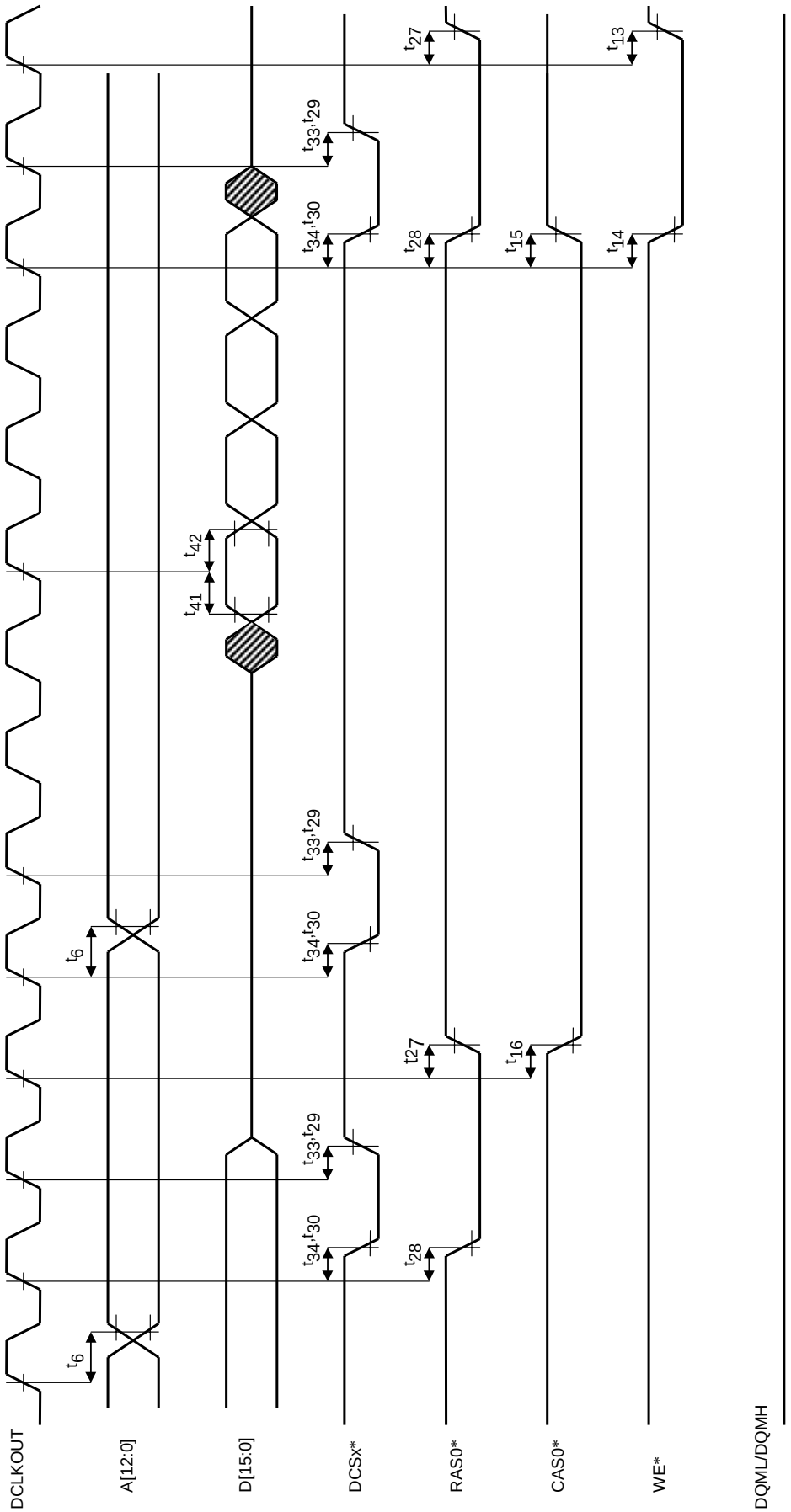
DRAM Access Timing(1) - Write Operation



DRAM Access Timing(2) - Read Operation



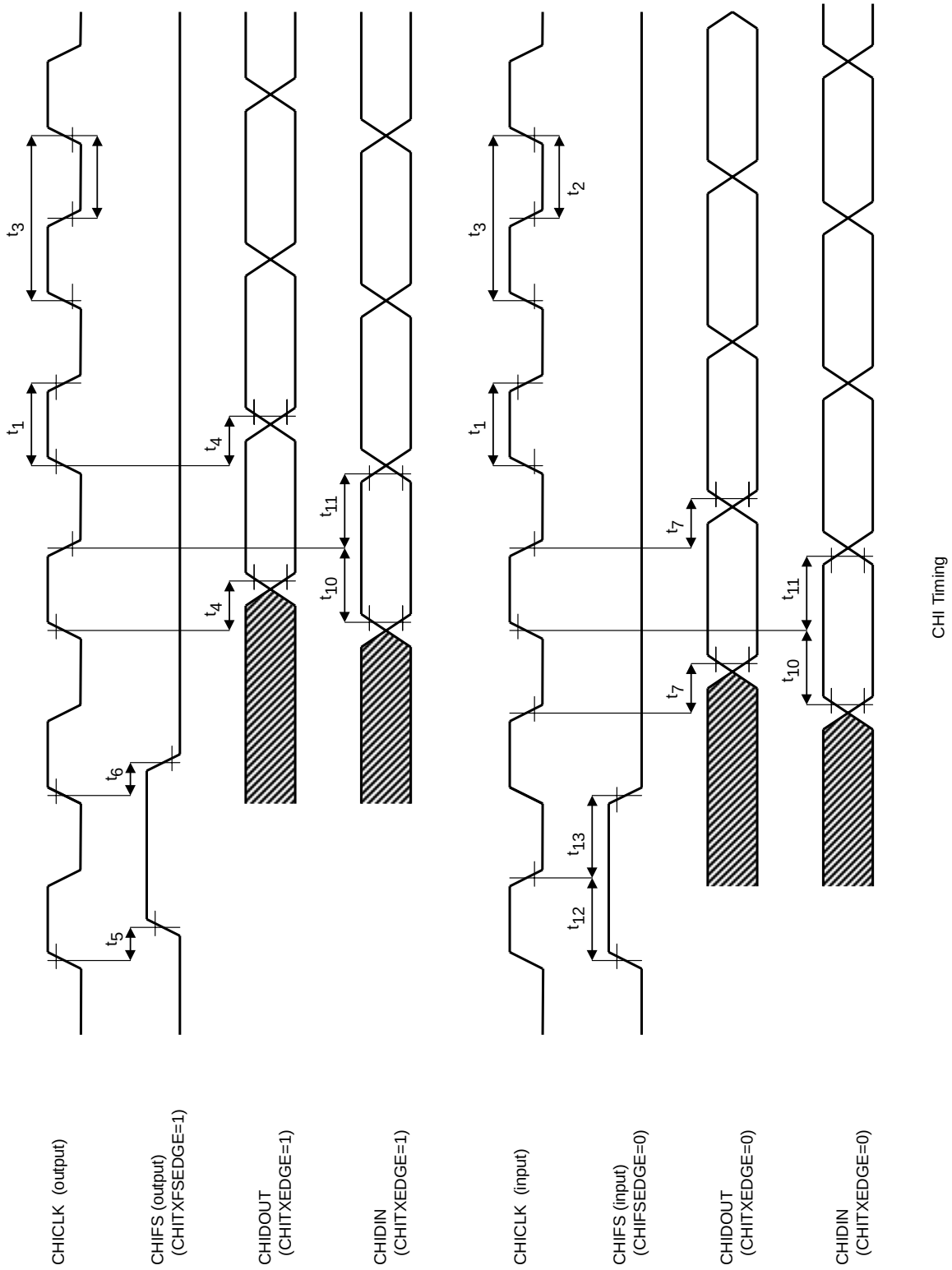
SDRAM Access Timing(1) - Write Operation



SDRAM Access Timing(2) - Read Operation

<CHI>

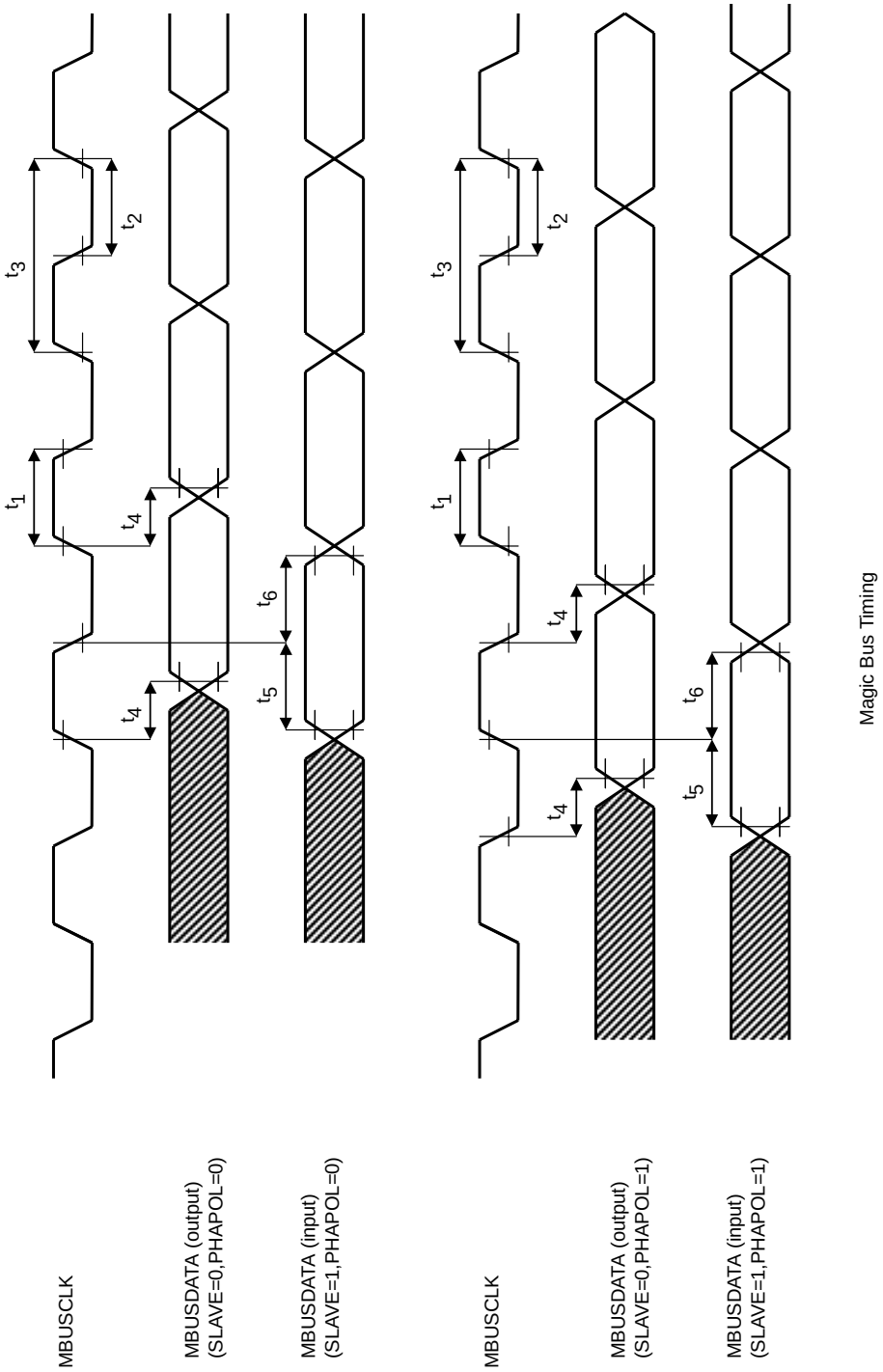
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	CHICLK high time	—	100	—	ns
t_2	CHICLK low time	—	100	—	ns
t_3	CHICLK period	—	225	—	ns
t_4	Delay CHICLK Rising/Falling to CHIDOUT (Master)	—	—	5	ns
t_5	Delay CHICLK Rising/Falling to CHIFS (Master)	Rising	—	5	ns
t_6	Delay CHICLK Rising/Falling to CHIFS (Master)	Falling	—	5	ns
t_7	Delay CHICLK Rising/Falling to CHIDOUT (Slave)	—	—	15	ns
t_8	Delay CHICLK Rising/Falling to CHIFS (Slave)	Rising	—	15	ns
t_9	Delay CHICLK Rising/Falling to CHIFS (Slave)	Falling	—	15	ns
t_{10}	CHIDIN to CHICLK Rising/Falling Setup time	—	20	—	ns
t_{11}	CHIDIN to CHICLK Rising/Falling Hold time	—	20	—	ns
t_{12}	CHIFS to CHICLK Rising/Falling Setup time (Slave)	—	20	—	ns
t_{13}	CHIFS to CHICLK Rising/Falling Hold time (Slave)	—	20	—	ns



CHI Timing

<Magic Bus>

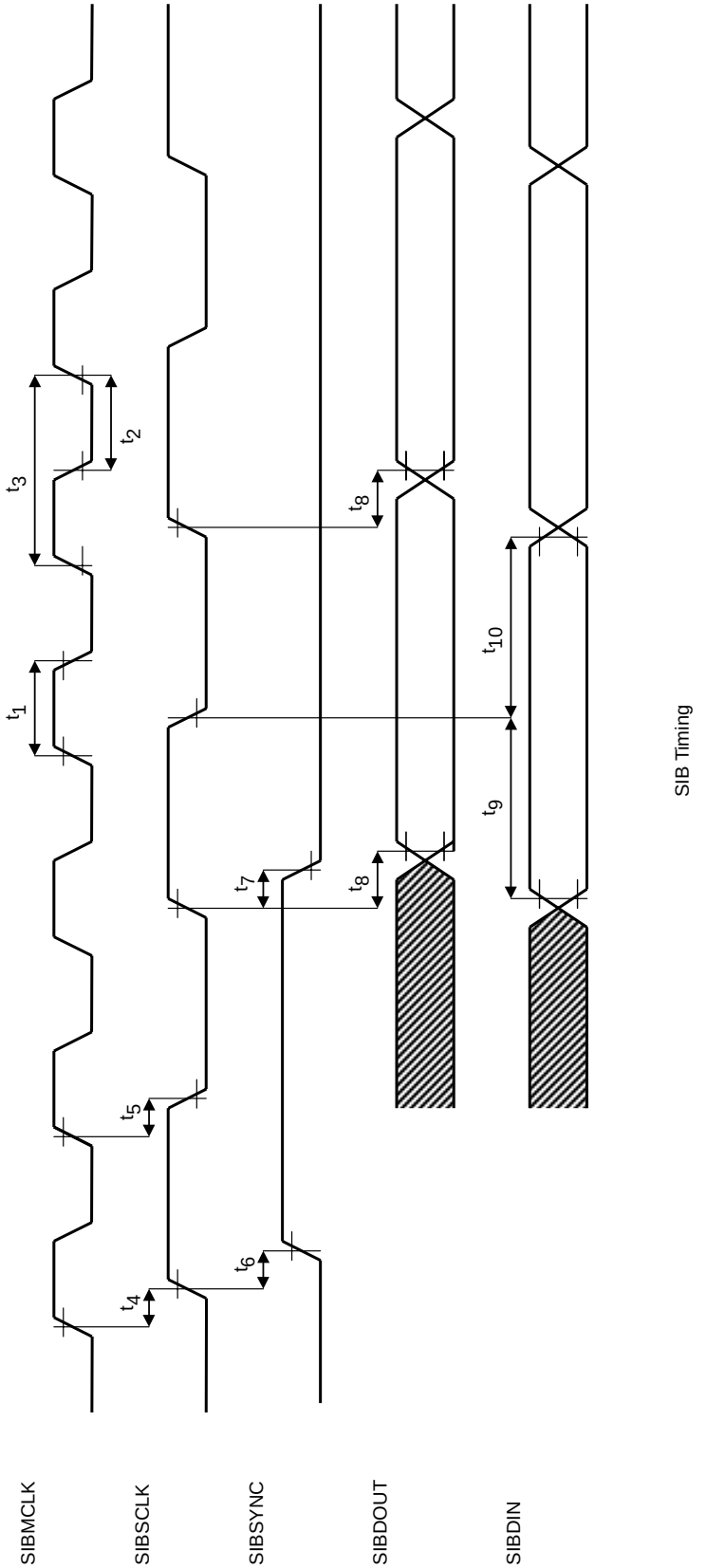
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	MBUSCLK high time	—	30	—	ns
t_2	MBUSCLK low time	—	30	—	ns
t_3	MBUSCLK period	—	67	—	ns
t_4	Delay MBUSCLK Rising/Falling to MBUSDATA (Master)	—	—	2	ns
t_5	MBUSDATA to MBUSCLK Rising/Falling Setup time (Slave)	—	4	—	ns
t_6	MBUSDATA to MBUSCLK Rising/Falling Hold time (Slave)	—	5	—	ns



Magic Bus Timing

<SIB>

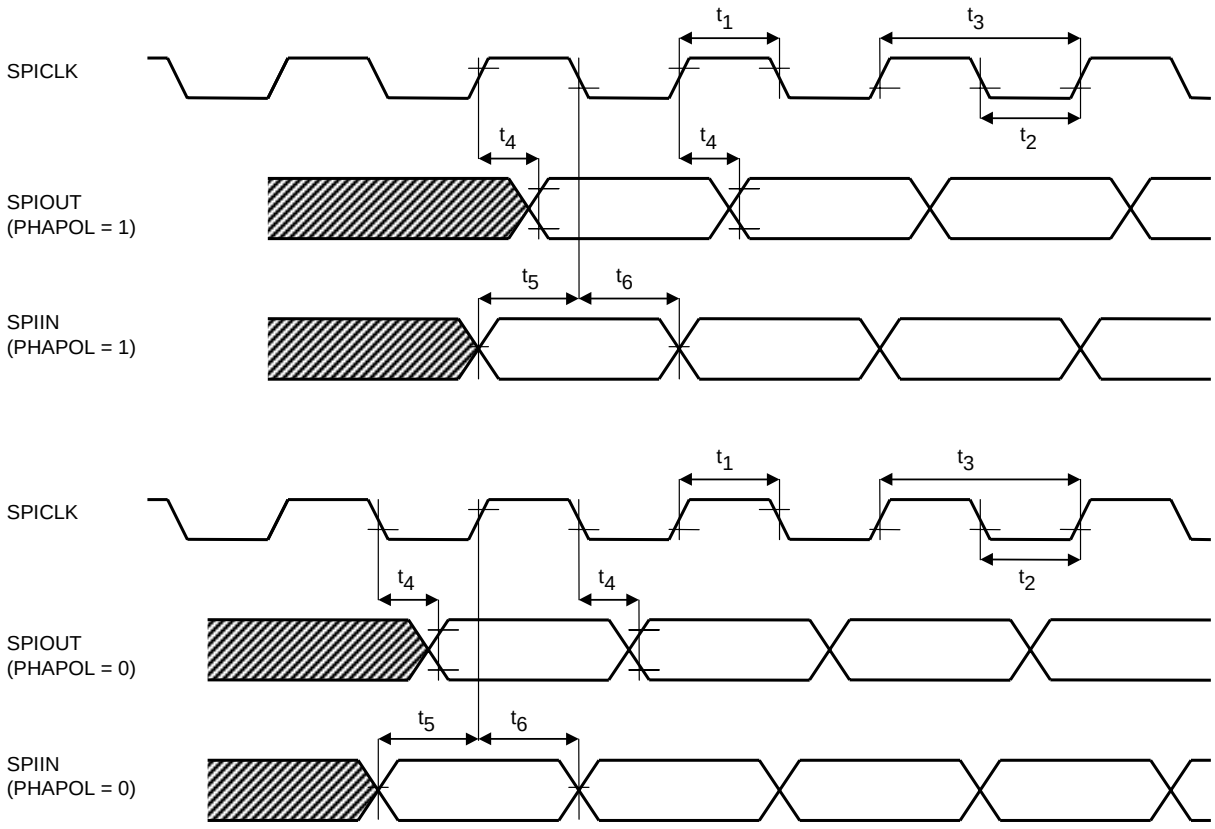
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	SIBMCLK high time	—	20	—	ns
t_2	SIBMCLK low time	—	20	—	ns
t_3	SIBMCLK period	—	50	—	ns
t_4	Delay SIBMCLK (Master) to SIBSCLK	Rising	—	5	ns
t_5	Delay SIBMCLK (Master) to SIBSCLK	Falling	—	5	ns
t_6	Delay SIBSCLK Rising to SIBSYNC	Rising	—	2	ns
t_7	Delay SIBSCLK Rising to SIBSYNC	Falling	—	2	ns
t_8	Delay SIBSCLK Rising to SIBDOUT	—	—	2	ns
t_9	SIBDIN to SIBSCLK Falling Setup time	—	20	—	ns
t_{10}	SIBDIN to SIBSCLK Falling Hold time	—	0	—	ns



SIB Timing

<SPI>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	SPICLK high time	—	160	—	ns
t_2	SPICLK low time	—	160	—	ns
t_3	SPICLK period	—	340	—	ns
t_4	Delay SPICLK Rising/Falling to SPIOUT	—	—	5	ns
t_5	SPIIN to SPICLK Rising/Falling Setup time	—	15	—	ns
t_6	SPIIN to SPICLK Rising/Falling Hold time	—	15	—	ns

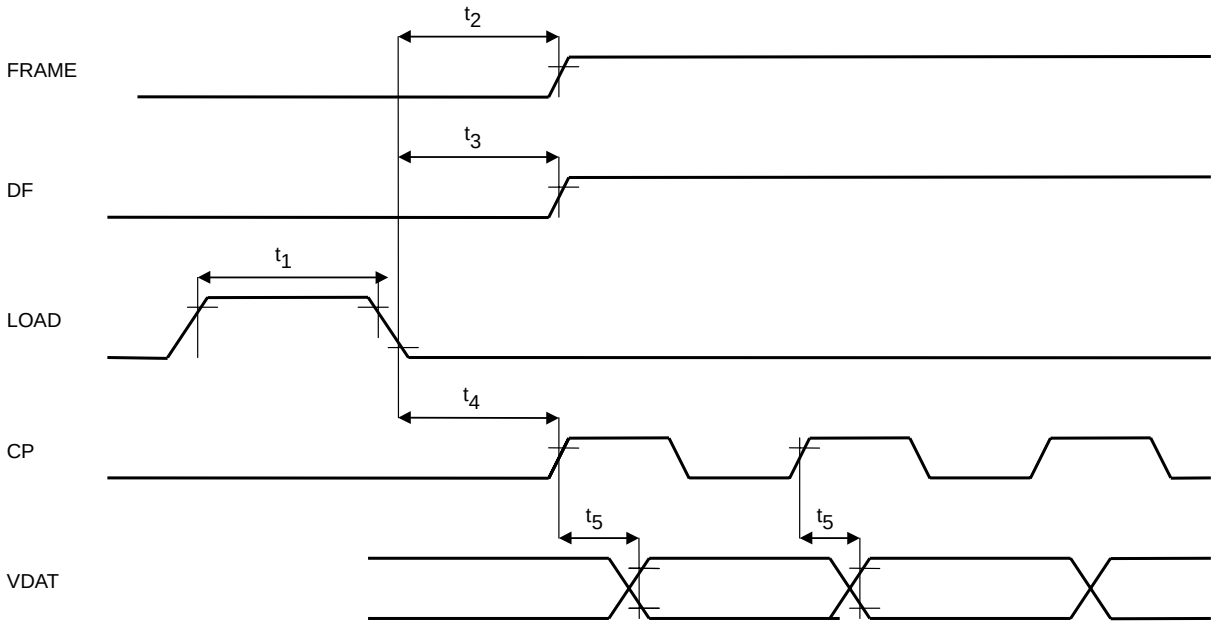


SPI Timing

<VIDEO>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	LOAD Pulse width	—	125	—	ns
t_2	Delay LOAD Falling to FRAME	—	125	—	ns
t_3	Delay LOAD Falling to DF	—	125	—	ns
t_4	Delay LOAD Falling to CP	—	125	—	ns
t_5	Delay CP Rising to VDAT [3:0]	—	—	5	ns

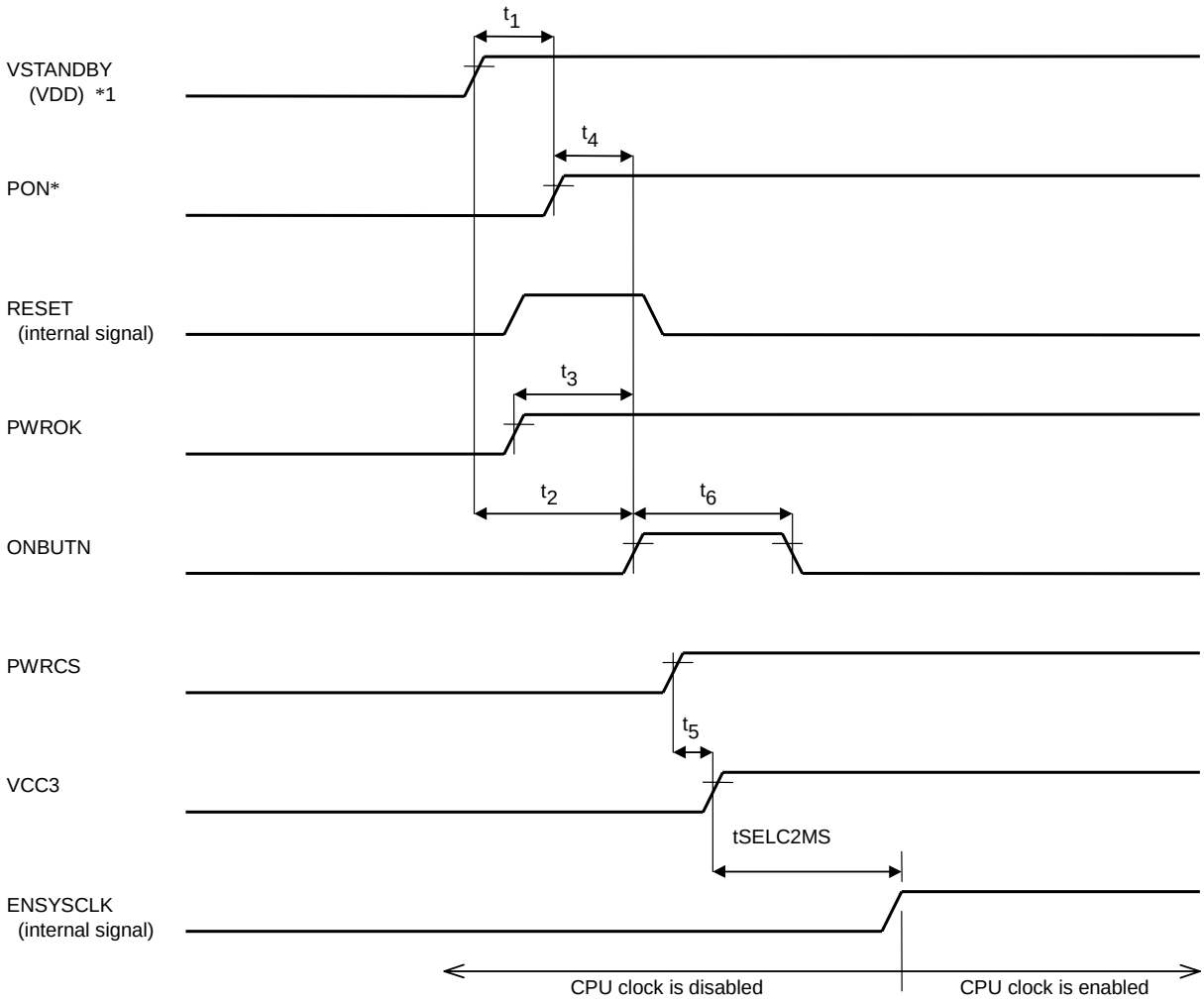
Note: Values shown assume a 58 MHz clock for the CPU. Min and Max values are programmable using Video Control Registers.



VIDEO Timing

<POWER>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	VSTANDBY to PON* Rising	—	50	—	ms
t_2	VSTANDBY to ONBUTN delay time	—	2	—	s
t_3	PWROK Rising to ONBUTN Rising	—	2	—	s
t_4	PON* Rising to ONBUTN Rising	—	1	—	us
t_5	PWRCS Rising to VCC3 Rising	—	0	—	s
t_6	ONBUTN pulse width	—	1	—	us

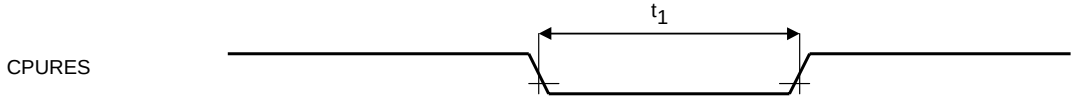


Power Timing

*1 This signal is power for the TMPR3911BU/BXB and other components in the system that must never lose power. That means VDD itself for TMPR3911BU/BXB.

<CPU RESET>

Item	Parameter	Rising/Falling	Min	Max	Unit
t_1	CPURES* low time	—	10	—	ns



CPU Reset Timing

18.2 Electrical Characteristics of TMPR3912AU-92/XB-92

18.2.1 Absolute maximum ratings (TMPR3912AU-92/XB-92)

V_{SS} = 0 V (GND)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{DD}	V_{SS} – 0.5 to 4.5	V
Input voltage	V_{IN}	V_{SS} – 0.5 to V_{DD} + 0.5	V
Storage temperature	T_{STG}	–55 to 125	°C
Maximum dissipation (Ta = 70°C)	P_D	1	W

Note: Using an LSI at specifications higher than the maximum ratings can cause permanent damage to the LSI. For normal operation, use under the recommended operating conditions. Exceeding the recommended operating conditions may affect the reliability of the LSI.

18.2.2 Recommended operating conditions (TMPR3912AU-92/XB-92)

V_{SS} = 0 V (GND)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Power Supply voltage	V_{DD}		3.0	3.3	3.6	V
Operating temperature	T_{OPR}		0	—	70	°C

18.2.3 DC characteristics (TMPR3912AU-92/XB-92)

(Ta = 0°C~70°C, V_{DD} = 3.3 V ± 0.3 V)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating current [8]	I_{DD}	$V_{IN} = V_{DD}$ or V_{SS} V_{DD} = Max $I_{OH} = I_{OL}$=0 mA f_{IN} [7] = 10 MHz	—	110	130	mA
Static current	$I_{DDS}1$	$V_{IN} = V_{DD}$ or V_{SS} V_{DD} = Max $I_{OH} = I_{OL}$ = 0 mA SLEEP mode & RTC stop mode	—	10	100	µA
	$I_{DDS}2$	$V_{IN} = V_{DD}$ or V_{SS} V_{DD} = Max $I_{OH} = I_{OL}$ = 0 mA SLEEP mode & RTC Running mode	—	20	120	µA
Input Leakage current	I_{IN}	$V_{IN} = V_{DD}$ or V_{SS}	–10	—	10	µA
Input voltage [1]	V_{IH1}	V_{DD} = 3.6 V	$V_{DD} \times 0.8$	—	V_{DD} + 0.3	V
	V_{IL1}	V_{DD} = 3.0 V	–0.3	—	$V_{DD} \times 0.2$	V
Input voltage [2]	V_{IH2}	V_{DD} = 3.6 V	2.4	—	V_{DD} + 0.3	V
	V_{IL2}	V_{DD} = 3.0 V	–0.3	—	0.6	V
Input voltage [3]	V_{OH1}	V_{DD} = 3.0 V, I_{OH} = –4 mA	V_{DD} – 0.6	—	—	V
	V_{OL1}	V_{DD} = 3.0 V, I_{OL} = 4 mA	—	—	0.4	V
Output voltage [4]	V_{OH2}	V_{DD} = 3.0 V, I_{OH} = –8 mA	V_{DD} – 0.6	—	—	V
	V_{OL2}	V_{DD} = 3.0 V, I_{OL} = 8 mA	–	—	0.4	V
Output voltage [5]	V_{OH3}	V_{DD} = 3.0 V, I_{OH} = –16 mA	V_{DD} – 0.6	—	—	V
	V_{OL3}	V_{DD} = 3.0 V, I_{OL} = 16 mA	—	—	0.4	V
Output voltage [6]	V_{OH4}	V_{DD} = 3.0 V, I_{OH} = –24 mA	V_{DD} – 0.6	—	–	V
	V_{OL4}	V_{DD} = 3.0 V, I_{OL} = 24 mA	—	—	0.4	V
Input current (Pull-down resistor)	I_{IHP}	V_{DD} = Max $V_{IN} = V_{DD}$	20	—	120	µA

(1) SYSCLKIN , TESTCPU,TESTAIU,TESTIN

(2) Other inputs

(3) D[31:0], RAS0*, RAS1*, DCS0*, DCKE*, DQMH, DQML, DREQ*, DGRNT*, BC32K, VDAT[3:0], CP, LOAD, DF, FRAME, DISPON, VIDDONE, PWRCS, TXD, RXD, CS3-0*, CHIFS, CHICLK, CHIDOUT, CHIDIN, IO[6:0], SPICLK, SPIOUT, SPIIN, SIBSYNC, SIBDOUT, SIBMCLK, SIBCLK, RXPWR, IROUT, CARD1WAIT*, CARD2WAIT*, MIOX[2:0]

(4) A[12:0], ALE, RD*, WE*, CAS3-0*, CARDREG*, CARDIOWR*, CARD1CSL*, CARD1CSH*, CARD2CSL*, CARD2CSH*

(5) DCLKOUT

(6) MBUSCLK, MBUSDATA

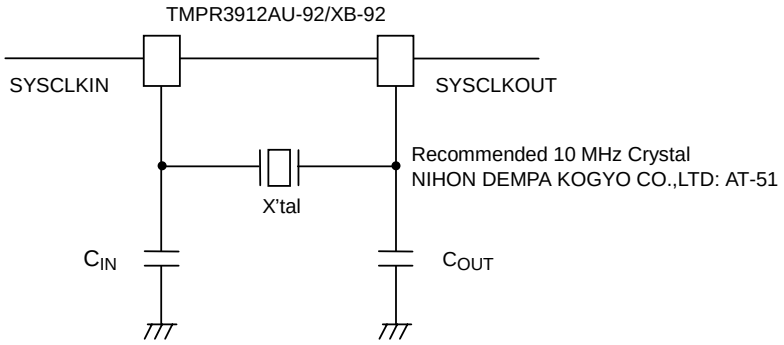
(7) f_{IN} is the frequency of the clock input from pins SYSCLKIN and SYSCLKOUT

(8) The operating current is the average current value when the TX3911/12 is operated in conjunction with the Toshiba TX System RISC Reference System with which the TX3911/12 is operated in CPU Continuous Operation Mode 40% of the time and in Low Power Consumption Mode (in which the CPU is stopped) 60% of the time.

18.2.4 Crystal oscillator characteristics (TMPR3912AU-92/XB-92)

18.2.4.1 Recommended oscillator conditions

(1) 10 MHz crystal

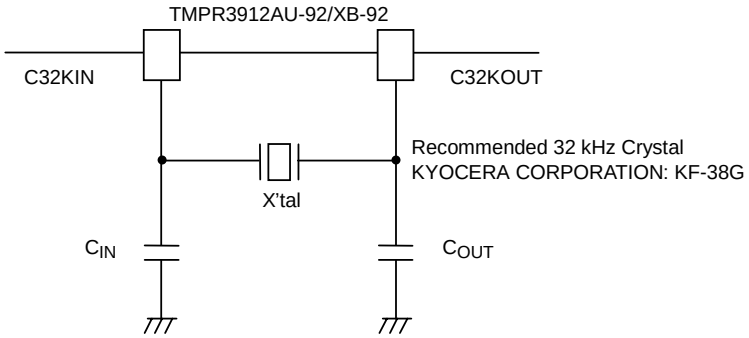


Parameter	Symbol	Recommended value		Unit
		Min	Max	
Crystal Oscillator frequency	f_{IN}	8.25	11.52	MHz
External capacitors	C_{IN}, C_{OUT}	10	33	pF

Please note that there are some considerations on the location of the external crystal as follows.

1. Please place the crystal as close to the TMPR3912AU-92/XB-92 as possible.
2. Please place the crystal as far from data bus lines as possible.
3. Please surround the crystal area with GND.

(2) 32 kHz crystal



Parameter	Symbol	Recommended value		Unit
		Min	Max	
External capacitors	C_{IN}, C_{OUT}	10	33	pF

Please note that there are some considerations on the location of the external crystal as follows.

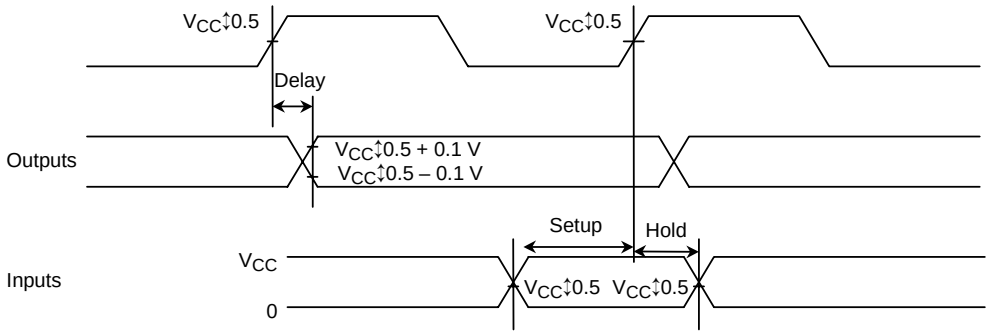
1. Please place the crystal as close to the TMPR3912AU-92/XB-92 as possible.
2. Please place the crystal as far from data bus lines as possible.
3. Please surround the crystal area with GND.

18.2.4.2 Electrical specifications

(V_{SS} = 0 V, V_{DD} = 3.3 V)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Crystal stabilization time 10 MHz	$T_{STA-10M}$	f = 8.25 MHz~9.216 MHz X'tal: AT-51 Cin = Cout = 10 pF~33 pF	—	—	10	ms
Crystal stabilization time 32 kHz	$T_{STA-32k}$	f = 32 kHz X'tal: KF-38G Cin = Cout = 10 pF~33 pF	—	—	2	s

18.2.5 Definition of AC specification (TMPR3912AU-92/XB-92)



18.2.6 AC characteristics (TMPR3912AU-92/XB-92)

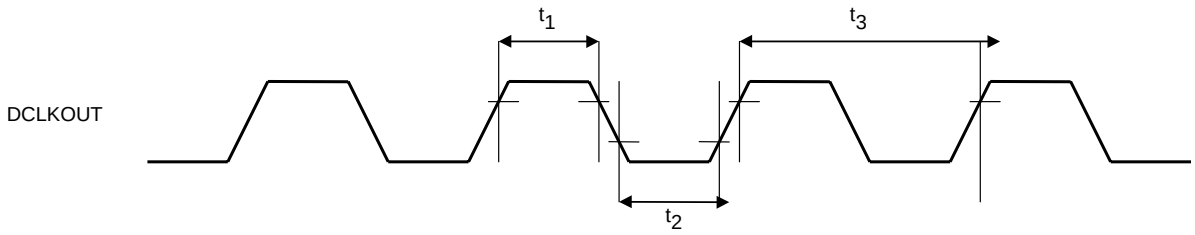
The following operating conditions apply to all values specified in this section.

T_a = 0~70 ºC, V_{DD} = 3.3 ± 0.3 V, External Capacitance = 40 pF

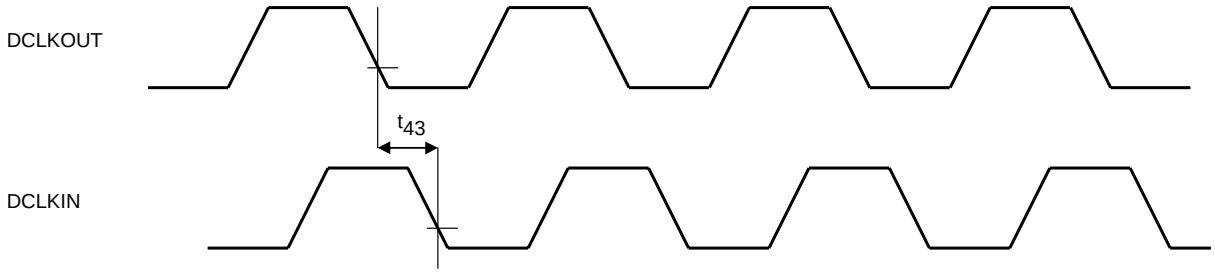
<Memory Interface>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	DCLKOUT high time	—	4.3	—	ns
t_2	DCLKOUT low time	—	4.3	—	ns
t_3	DCLKOUT period	—	10.8	—	ns
t_4	Delay DCLKOUT to ALE	Rising	—	5	ns
t_5	Delay DCLKOUT to ALE	Falling	—	3	ns
t_6	Delay DCLKOUT to A[12:0]	—	– 4	7	ns
t_7	Delay DCLKOUT to D[31:16]	—	—	8	ns
t_8	Delay DCLKOUT to D[15:0]	—	1	6.5	ns
t_9	Delay DCLKOUT to CS3-0*	Rising	—	10	ns
t_{10}	Delay DCLKOUT to CS3-0*	Falling	—	10	ns
t_{11}	Delay DCLKOUT to RD*	Rising	—	8	ns
t_{12}	Delay DCLKOUT to RD*	Falling	—	7	ns
t_{13}	Delay DCLKOUT to WE*	Rising	– 4	5	ns
t_{14}	Delay DCLKOUT to WE*	Falling	—	4	ns
t_{15}	Delay DCLKOUT to CAS3-0* (WE3-0*)	Rising	– 4	2.5	ns
t_{16}	Delay DCLKOUT to CAS3-0* (WE3-0*)	Falling	—	2.5	ns
t_{17}	Delay DCLKOUT to CARDxCSx*	Rising	—	9	ns
t_{18}	Delay DCLKOUT to CARDxCSx*	Falling	—	8	ns
t_{19}	Delay DCLKOUT to CARDDIR*	Rising	—	12	ns
t_{20}	Delay DCLKOUT to CARDDIR*	Falling	—	11	ns
t_{21}	Delay DCLKOUT to CARDREG*	Rising	—	9	ns
t_{22}	Delay DCLKOUT to CARDREG*	Falling	—	10	ns
t_{23}	Delay DCLKOUT to CARDIORD*	Rising	—	10	ns
t_{24}	Delay DCLKOUT to CARDIORD*	Falling	—	9	ns
t_{25}	Delay DCLKOUT to CARDIOWR*	Rising	—	9	ns
t_{26}	Delay DCLKOUT to CARDIOWR*	Falling	—	9	ns
t_{27}	Delay DCLKOUT to RAS0*	Rising	– 4	6	ns
t_{28}	Delay DCLKOUT to RAS0*	Falling	—	6	ns
t_{29}	Delay DCLKOUT to RAS1* (DCS1*)	Rising	1.0	6	ns
t_{30}	Delay DCLKOUT to RAS1* (DCS1*)	Falling	1.0	6	ns
t_{31}	Delay DCLKOUT to DQMH/L	Rising	1.0	6.5	ns
t_{32}	Delay DCLKOUT to DQMH/L	Falling	1.0	6.5	ns
t_{33}	Delay DCLKOUT to DCS0*	Rising	1.0	7	ns
t_{34}	Delay DCLKOUT to DCS0*	Falling	1.0	6	ns
t_{35}	Delay DCLKOUT to DCKE	Rising	1.0	8	ns
t_{36}	Delay DCLKOUT to DCKE	Falling	1.0	8	ns
t_{37}	Delay DCLKOUT to MCS3-0*	Rising	—	10	ns
t_{38}	Delay DCLKOUT to MCS3-0*	Falling	—	10	ns
t_{39}	D[31:16] to DCLKIN Setup time	—	1	—	ns
t_{40}	D[31:16] to DCLKIN Hold time	—	2.5	—	ns
t_{41}	D[15:0] to DCLKIN Setup time	—	1	—	ns
t_{42}	D[15:0] to DCLKIN Hold time	—	2.5	—	ns
t_{43}	DCLKOUT to DCLKIN Board Delay time	—	0	3	ns

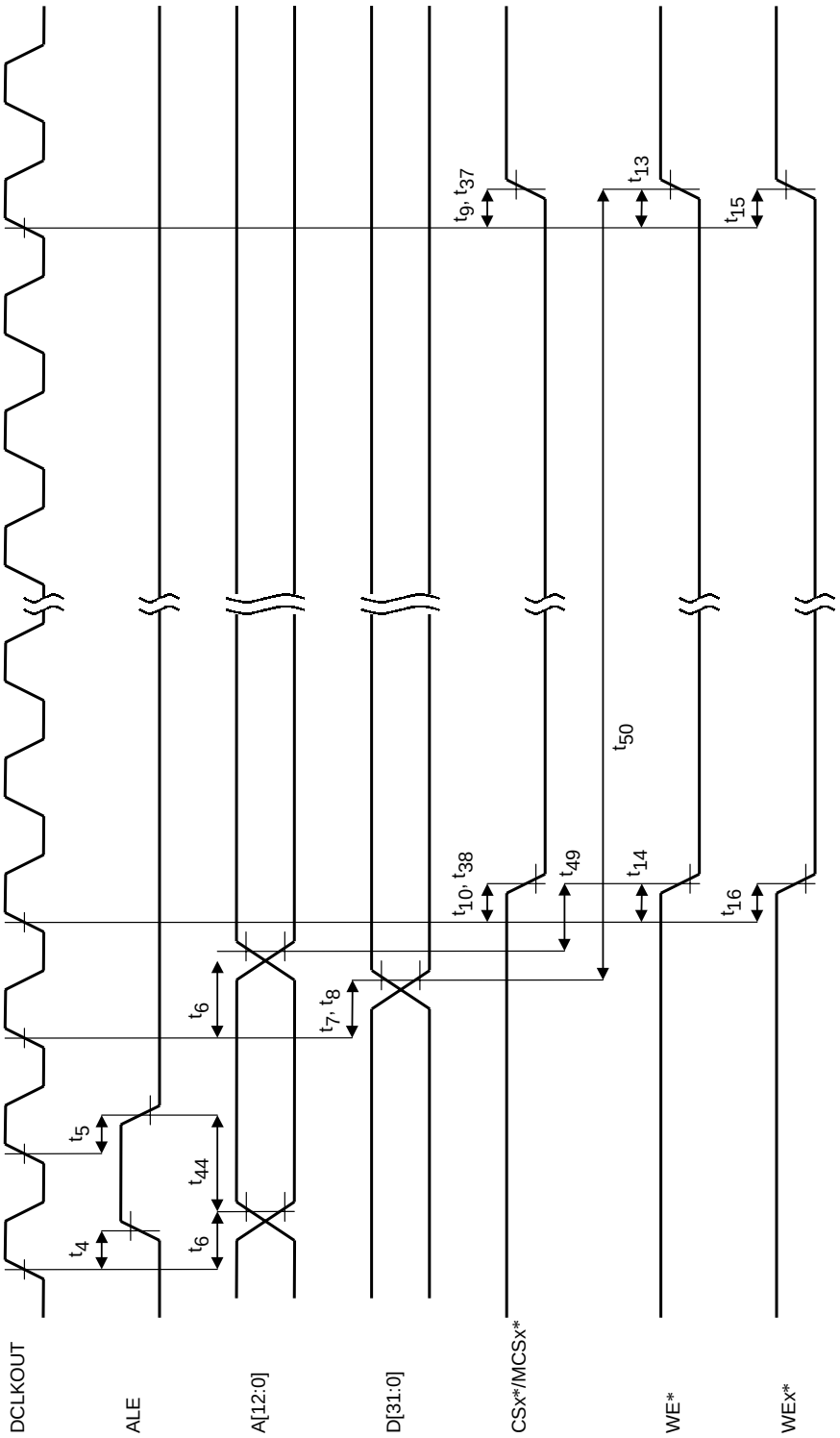
Symbol	Parameter	Calculation
t_{44}	Address valid to ALE falling edge	$t_3 + t_5 - t_6$
t_{46}	Access time for read	2*ACCVAL1* $t_3 - (t_{10}$ or $t_{38}) - (t_{39}$ or $t_{41})$
t_{48}	Address valid to CSX/MCSX* falling edge	At least $t_3 - t_6 + (t_{10}$ or $t_{38})$
t_{49}	Address valid to WE* falling edge	$t_3 - t_6 + t_{14}$
t_{50}	DATA valid to WE* rising edge	2*ACCVAL1* $t_3 + t_3 - (t_7$ or $t_8) + t_{13}$
t_{52}	Access time for 2nd/3rd/4th read	2*ACCVAL2* $t_3 + t_3 - t_6 - (t_{39}$ or $t_{41})$
t_{53}	Access time for read (Card)	2*((2*CARDxACCVAL) – 1)* $t_3 - t_{18} - t_{39}$
t_{54}	Address valid to CARDCS* falling edge	3* $t_3 - t_6 + t_{18}$
t_{55}	Write pulse width (Card)	2*((2*CARDxACCVAL) – 1)* $t_3 + t_{13} - t_{14}$
t_{56}	DATA valid to WE* rising edge (Card)	2*((2*CARDxACCVAL) – 1)* t_3 + 3* $t_3 - t_7 + t_{13}$
t_{58}	DATA valid to IOWR* rising edge	2*((2*CARDIOxACCVAL) – 1)* t_3 + 3* $t_3 - t_7 + t_{25}$
t_{61}	RAS* falling edge to CAS* falling edge	(4 or 6)* $t_3 + t_{16} - (t_{28}$ or $t_{30})$
t_{62}	Address valid to RAS* falling edge	$t_3 + (t_{28}$ or $t_{30}) - t_6$
t_{63}	Address valid to CAS* falling edge	$t_3 + t_{16} - t_6$
t_{65}	DATA valid to CAS* rising edge	(4 or 5)* $t_3 + t_{15} - (t_7$ or $t_8)$
t_{67}	Control signal active to DCKE	$t_3 + t_{35}$ – (Delay for each signal)
t_{68}	DATA valid to CAS* rising edge (In-Page)	3* $t_3 + t_{15} - (t_7$ or $t_8)$
t_{69}	Access time for read (IO Card)	2*((2*CARDxIOACCVAL) – 1)* t_3 – 2* $t_3 - t_{39} - t_{24}$



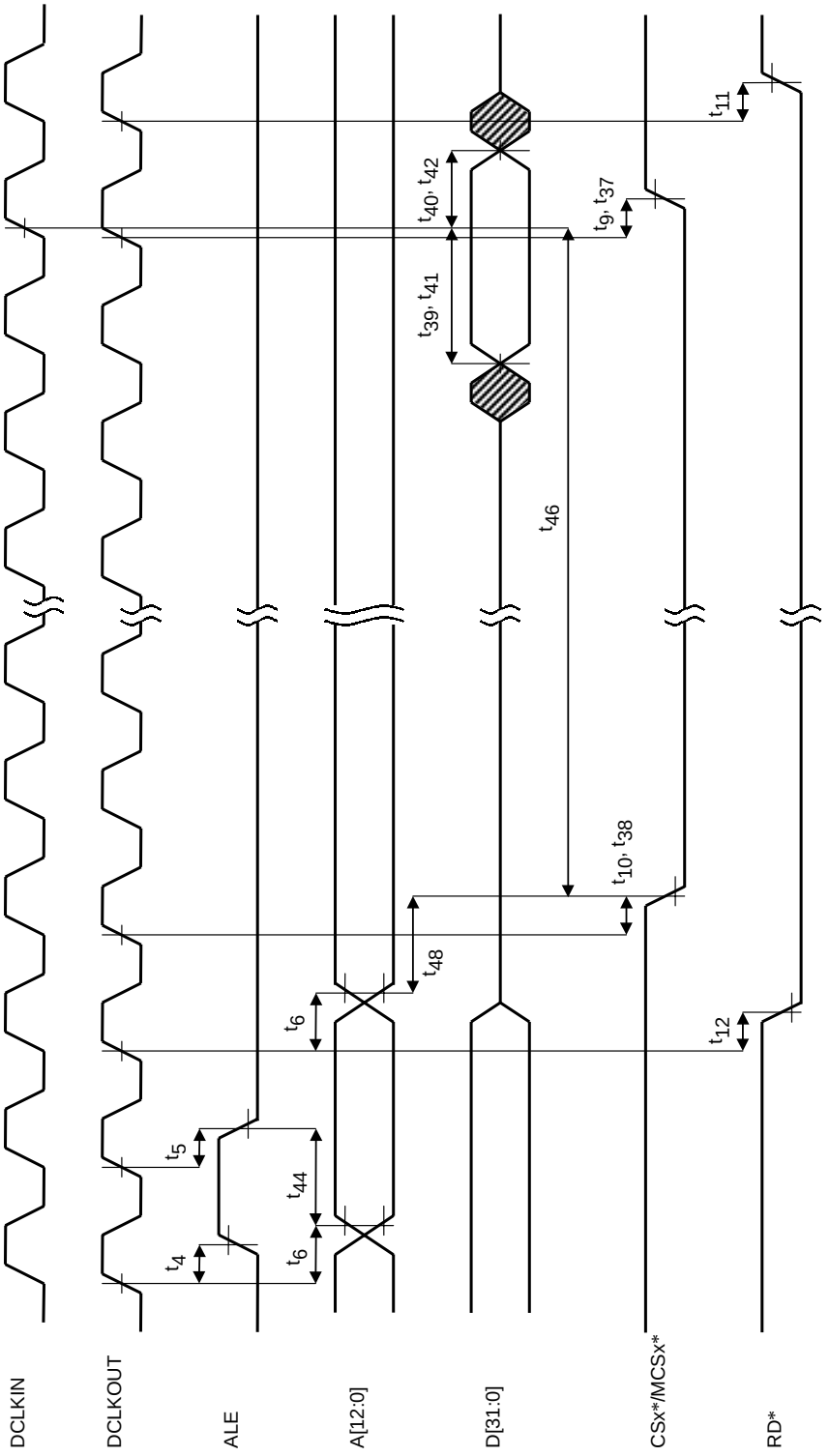
DCLKOUT Timing



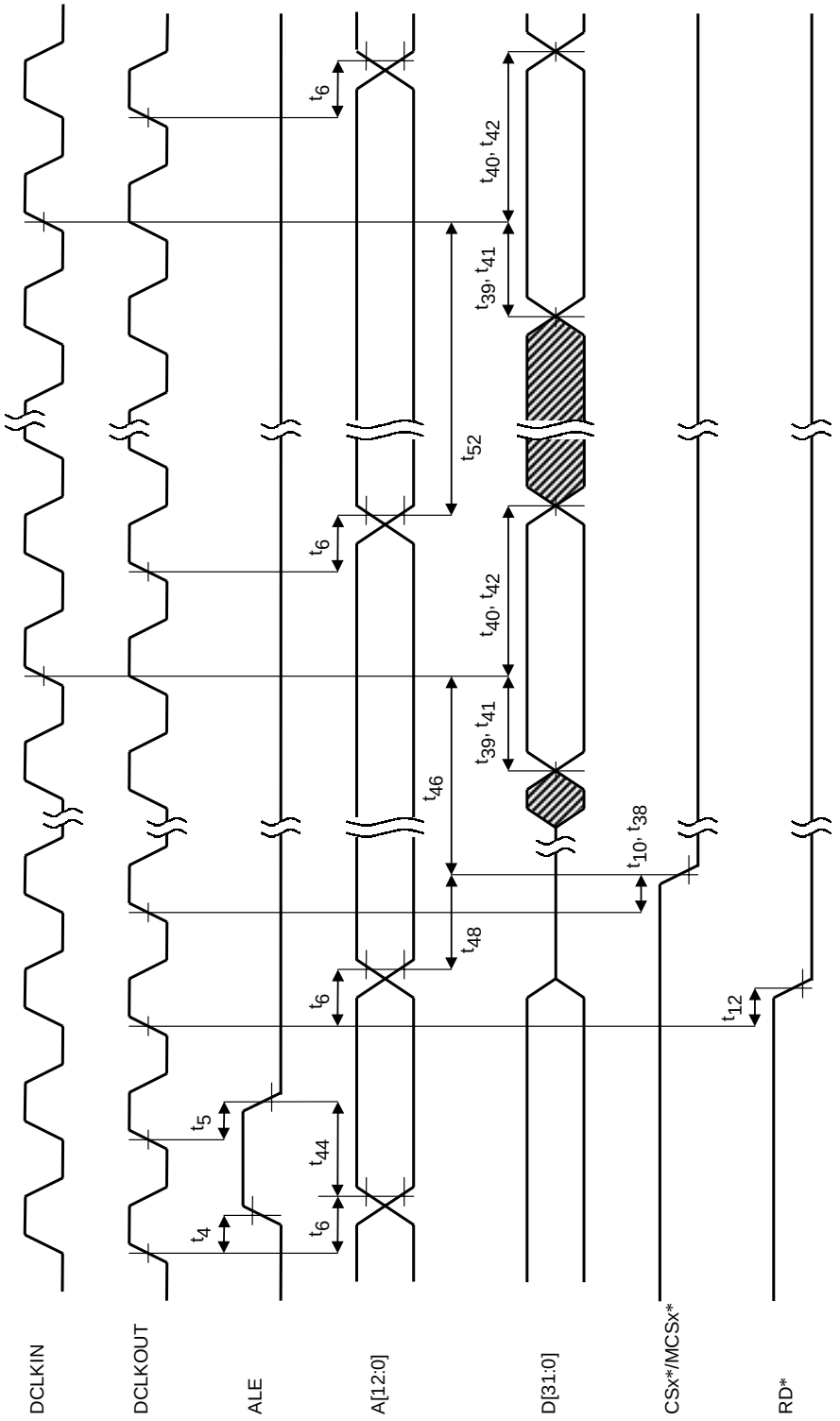
DCLKOUT to DCLKIN Timing



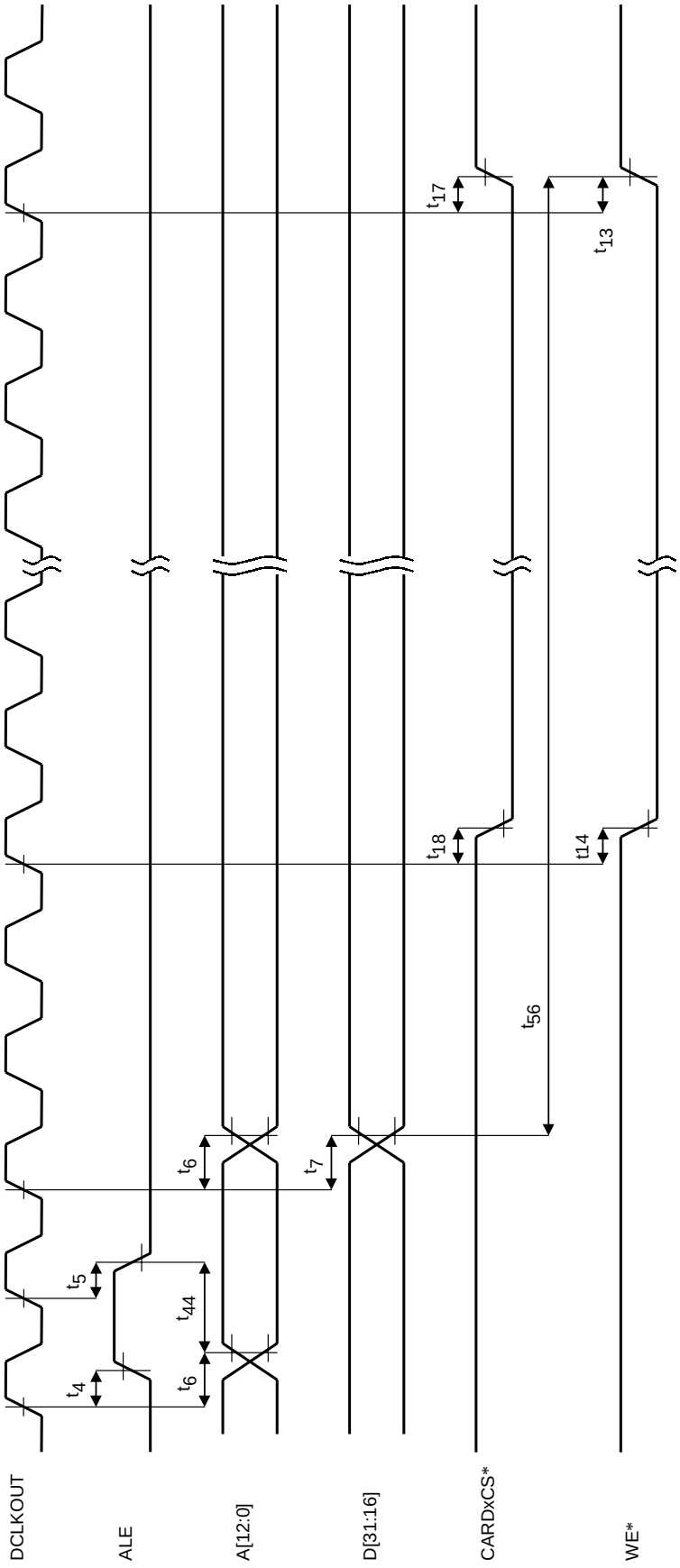
CS/MCS Device Access Timing(1) - Write Operation



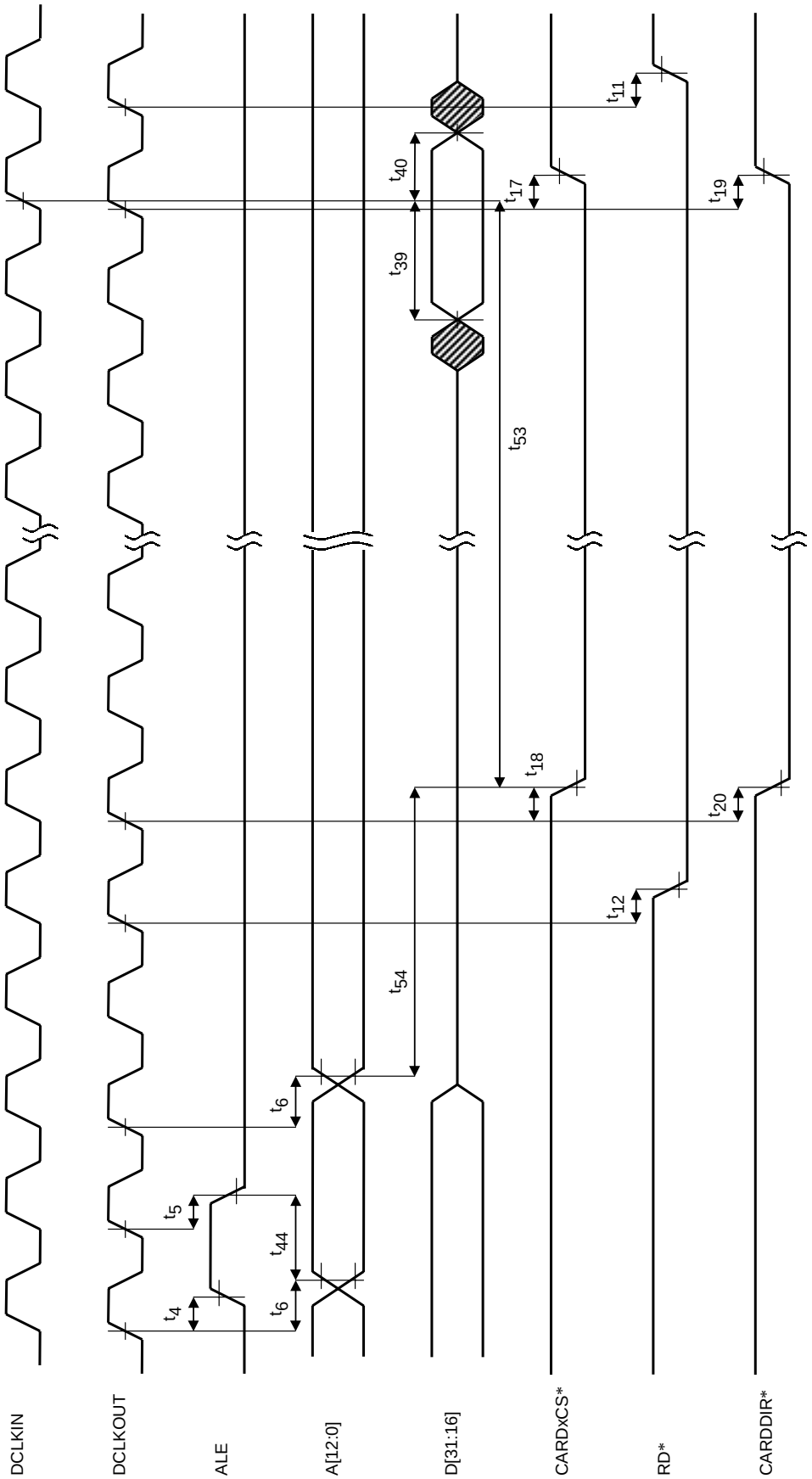
CS/MCS Device Access Timing(2) - Read Operation



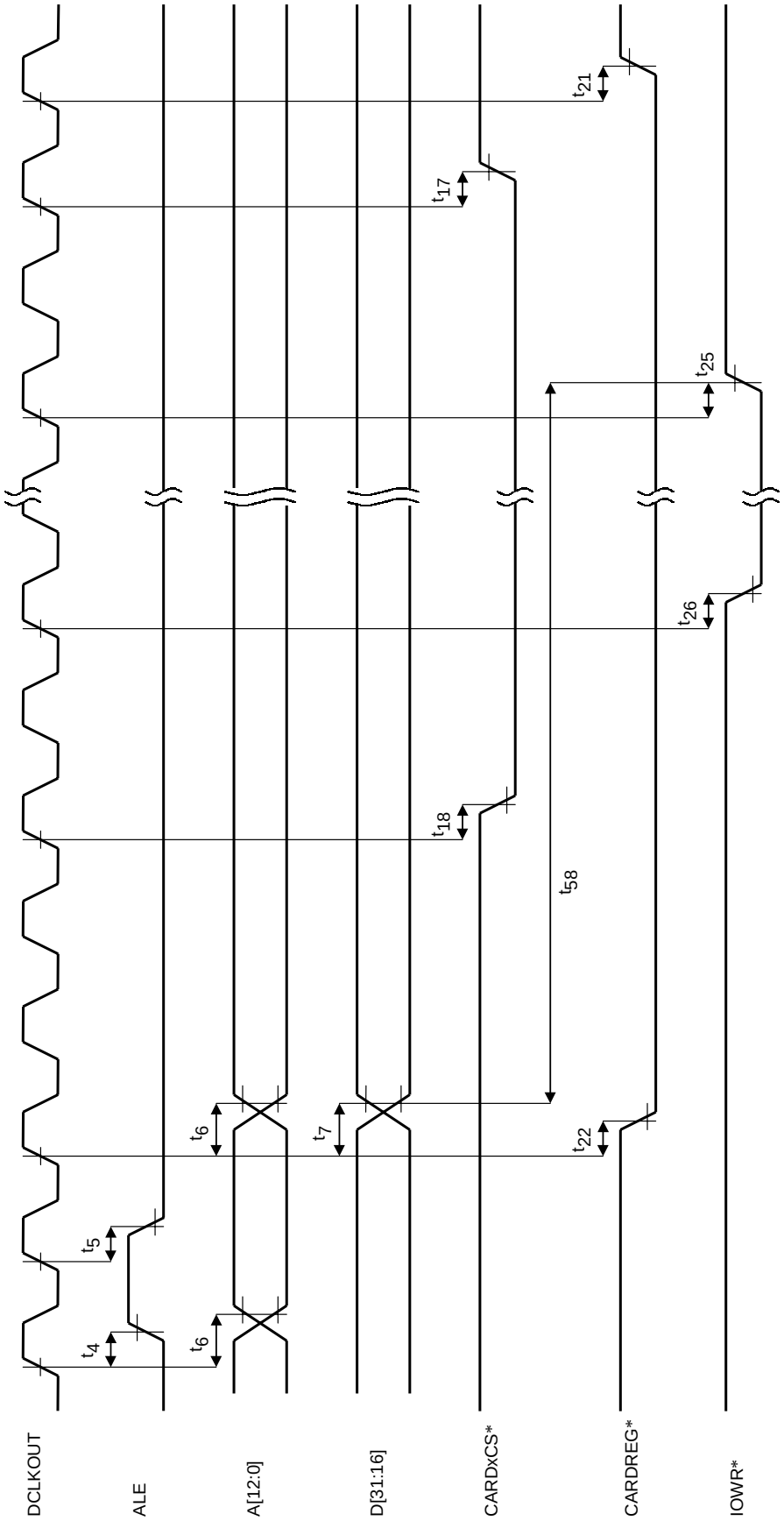
CS/MCS Device Access Timing(3) - Burst Read Operation with Page Mode Enabled



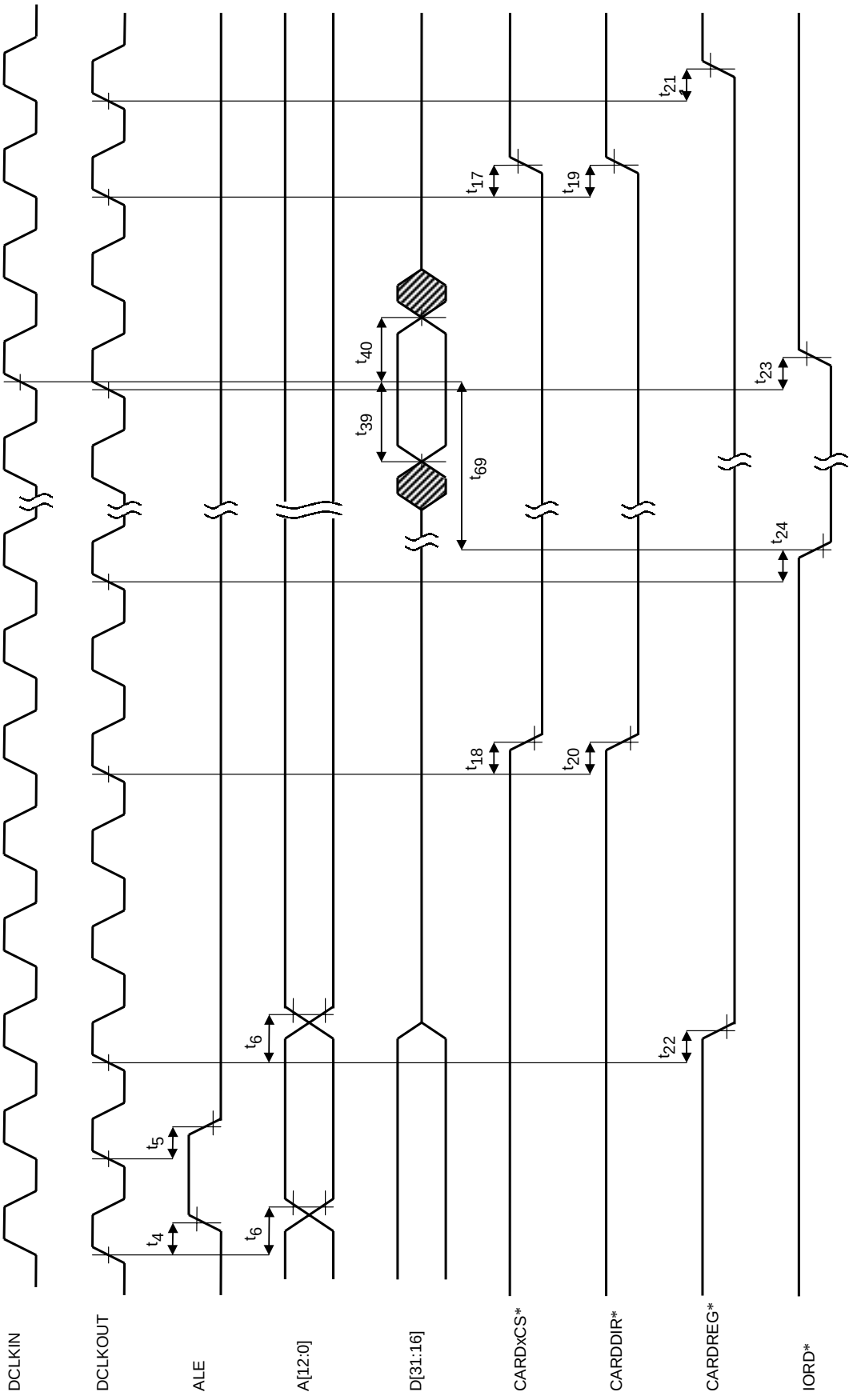
CARD Device Memory Access Timing(1) - Write Operation



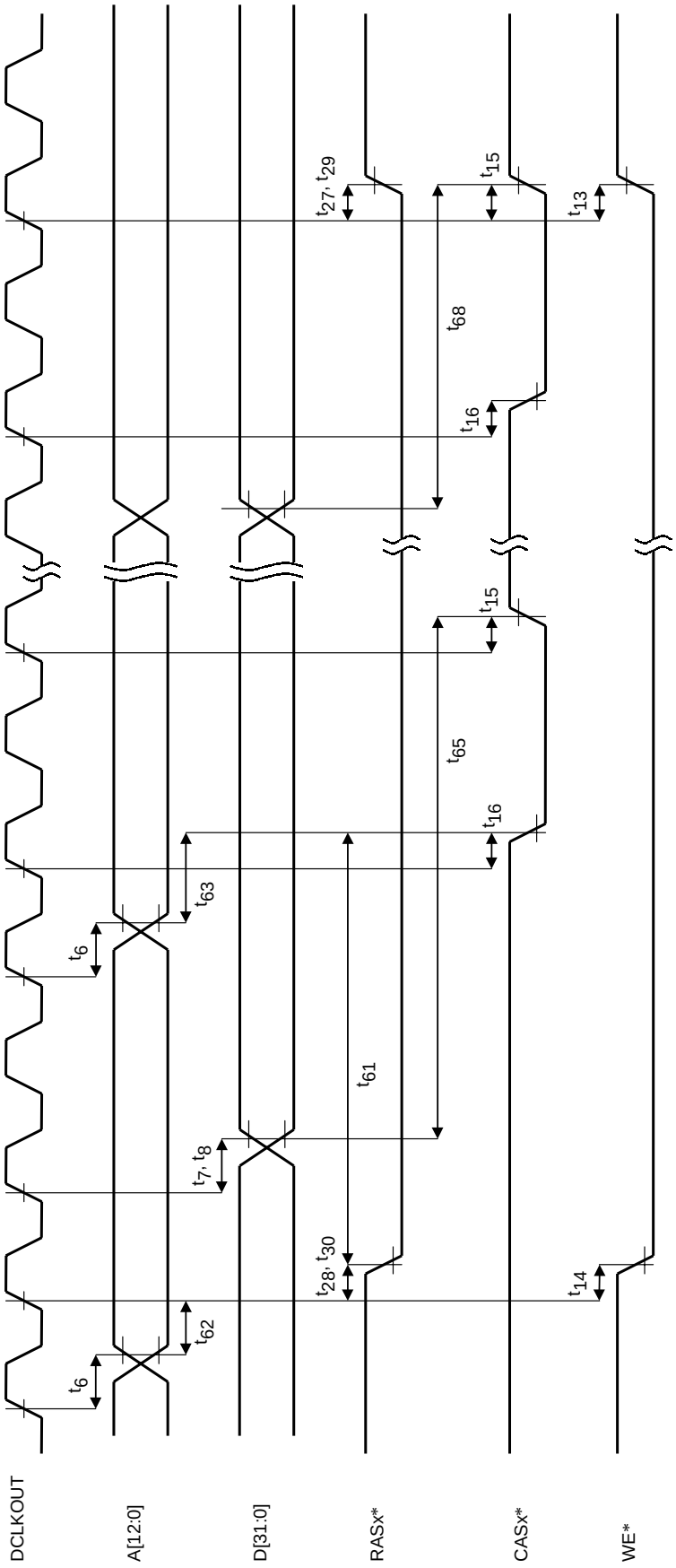
CARD Device Memory Access Timing(2) - Read Operation



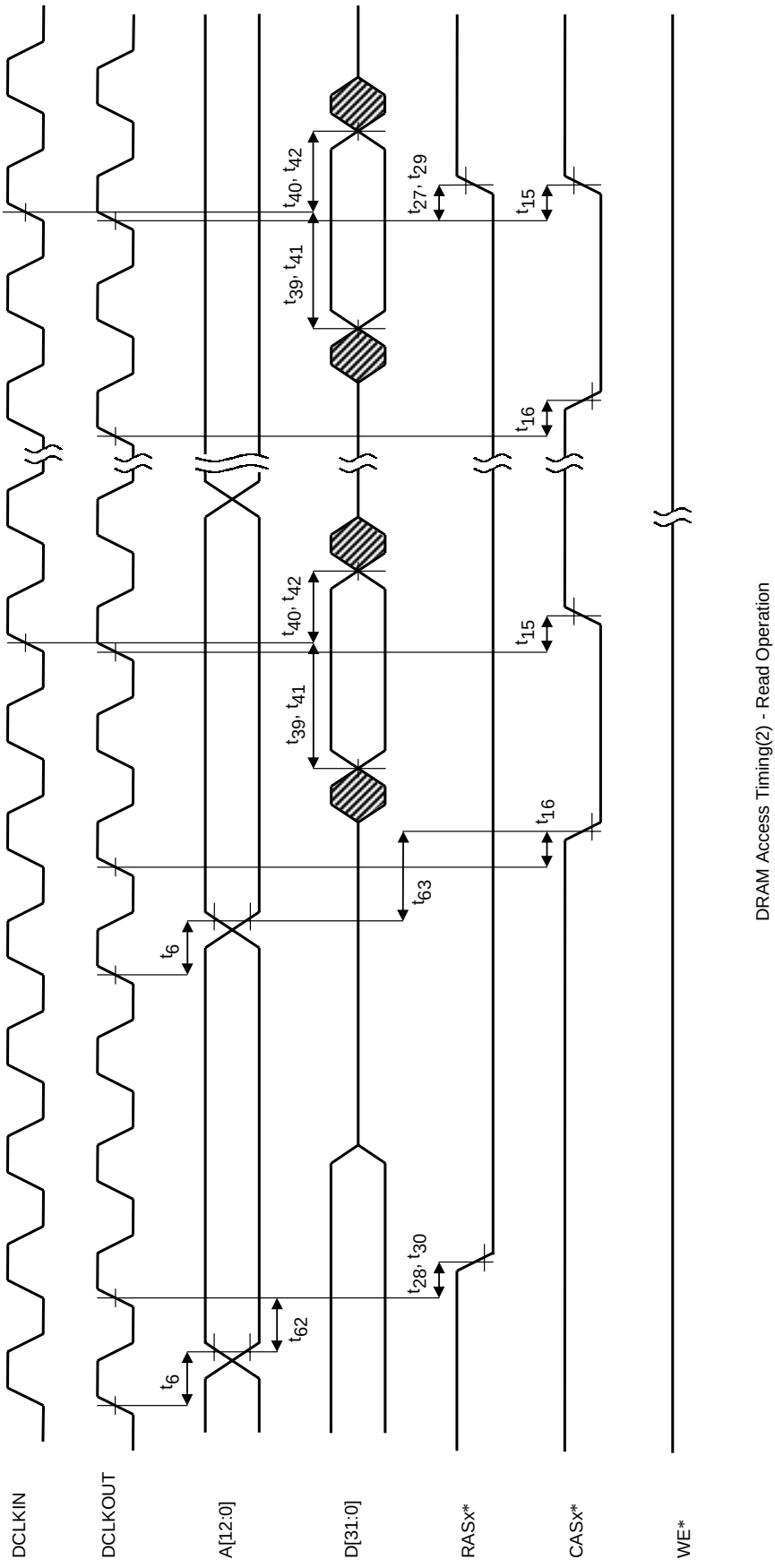
CARD Device I/O Access Timing(1) - Write Operation



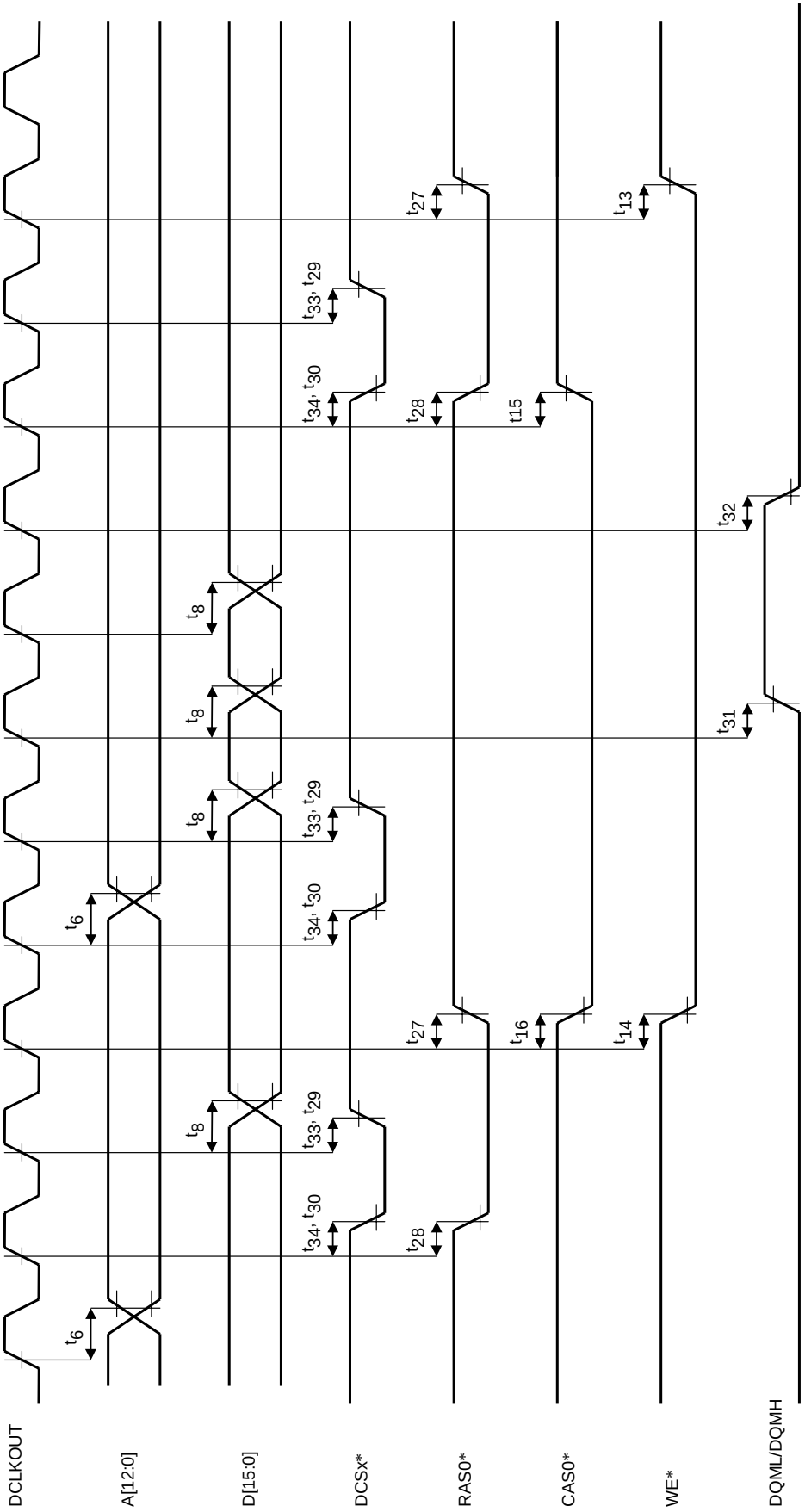
CARD Device I/O Access Timing(2) - Read Operation



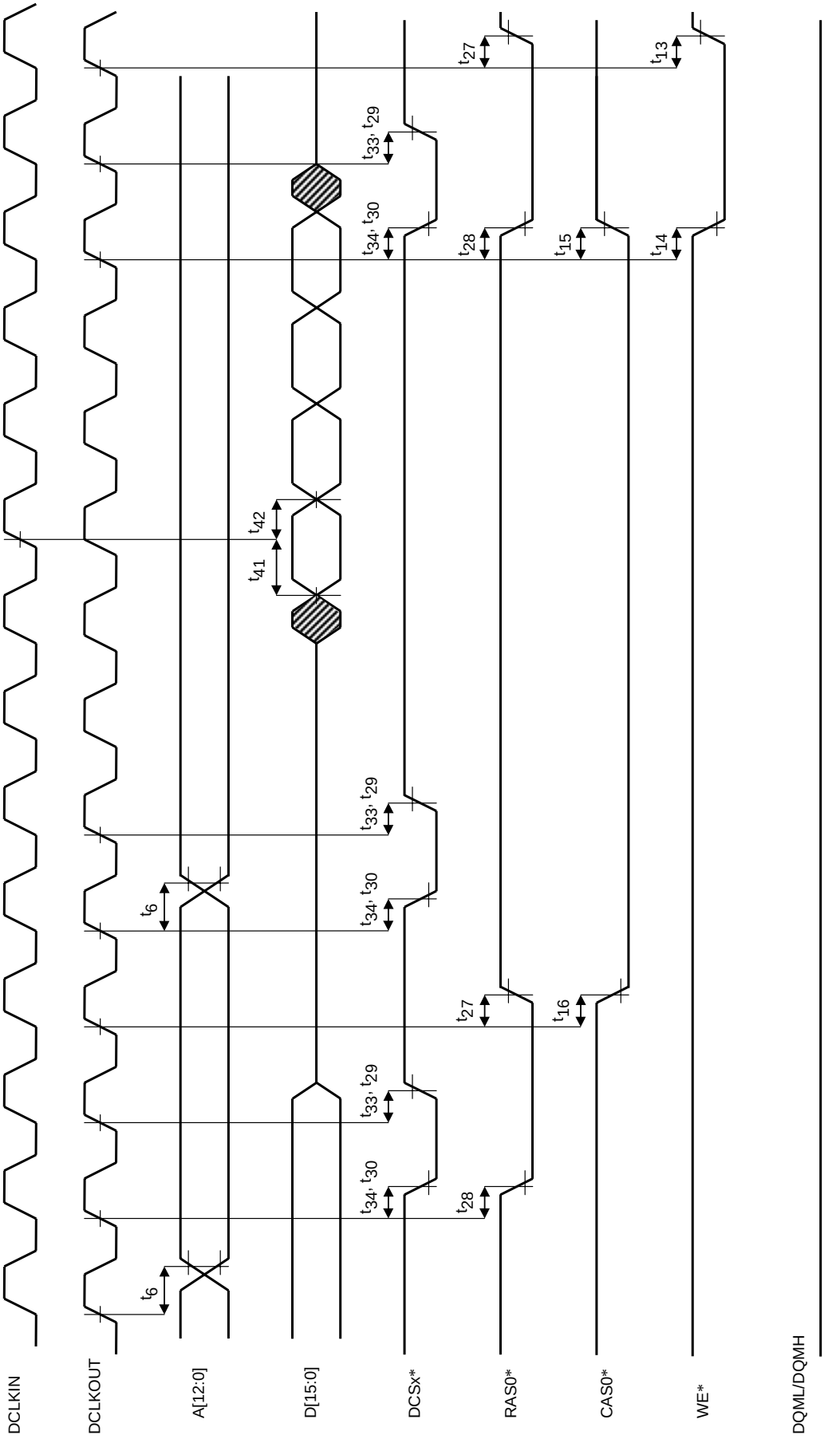
DRAM Access Timing(1) - Write Operation



DRAM Access Timing(2) - Read Operation



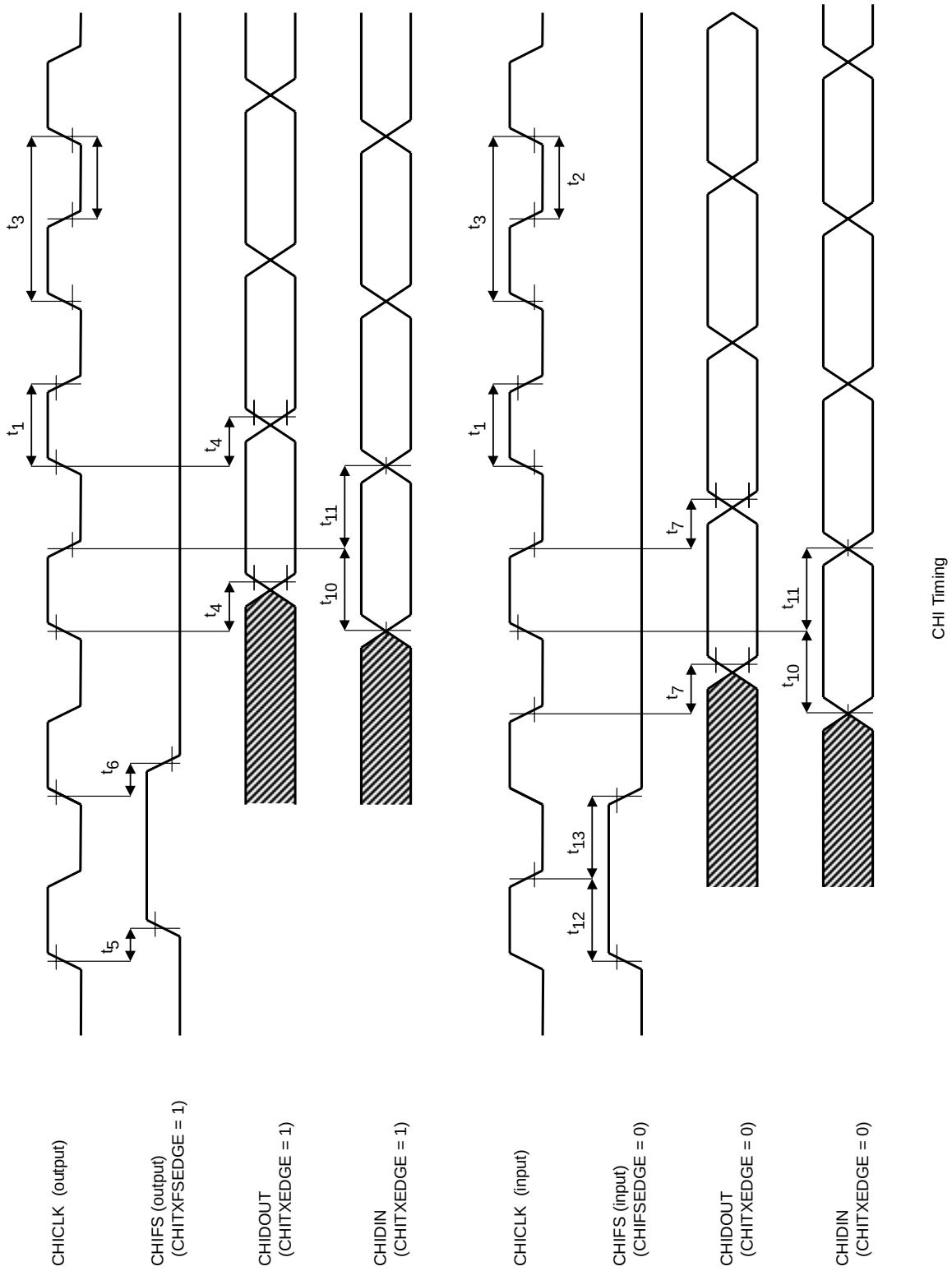
SDRAM Access Timing(1) - Write Operation



SDRAM Access Timing(2) - Read Operation

<CHI>

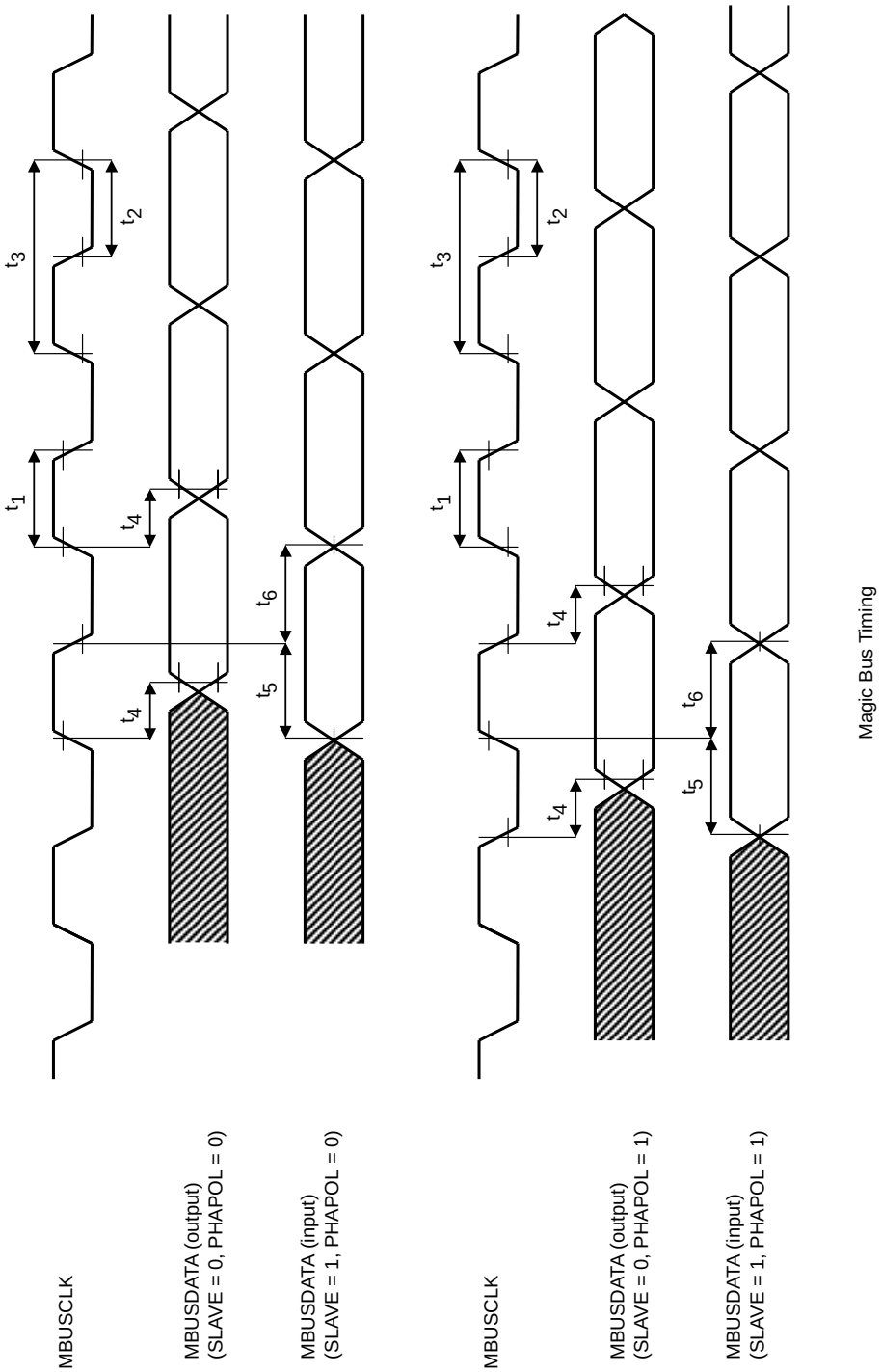
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	CHICLK high time	—	100	—	ns
t_2	CHICLK low time	—	100	—	ns
t_3	CHICLK period	—	225	—	ns
t_4	Delay CHICLK Rising/Falling to CHIDOUT (Master)	—	—	5	ns
t_5	Delay CHICLK Rising/Falling to CHIFS (Master)	Rising	—	5	ns
t_6	Delay CHICLK Rising/Falling to CHIFS (Master)	Falling	—	5	ns
t_7	Delay CHICLK Rising/Falling to CHIDOUT (Slave)	—	—	15	ns
t_8	Delay CHICLK Rising/Falling to CHIFS (Slave)	Rising	1.5	15	ns
t_9	Delay CHICLK Rising/Falling to CHIFS (Slave)	Falling	20	15	ns
t_{10}	CHIDIN to CHICLK Rising/Falling Setup time	—	20	—	ns
t_{11}	CHIDIN to CHICLK Rising/Falling Hold time	—	20	—	ns
t_{12}	CHIFS to CHICLK Rising/Falling Setup time (Slave)	—	20	—	ns
t_{13}	CHIFS to CHICLK Rising/Falling Hold time (Slave)	—	20	—	ns



CHI Timing

<Magic Bus>

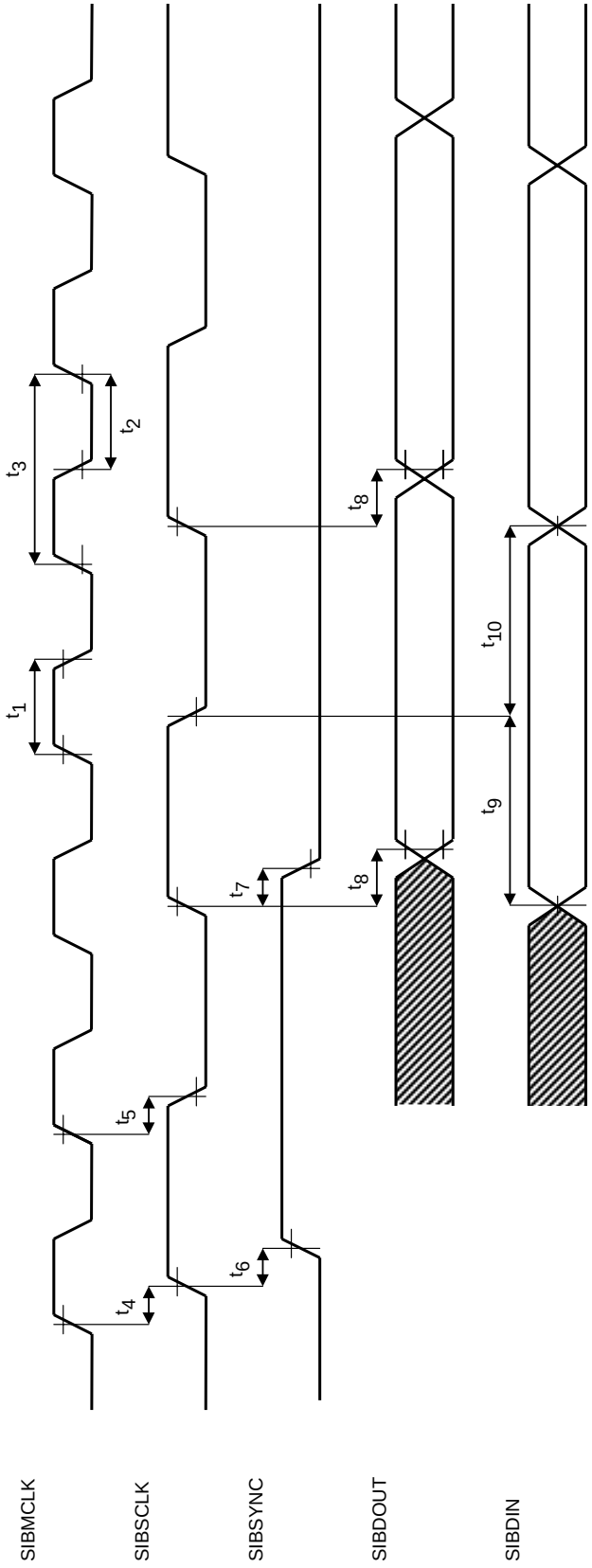
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	MBUSCLK high time	—	10	—	ns
t_2	MBUSCLK low time	—	10	—	ns
t_3	MBUSCLK period	—	25	—	ns
t_4	Delay MBUSCLK Rising/Falling to MBUSDATA (Master)	—	—	2	ns
t_5	MBUSDATA to MBUSCLK Rising/Falling Setup time (Slave)	—	4	—	ns
t_6	MBUSDATA to MBUSCLK Rising/Falling Hold time (Slave)	—	5	—	ns



Magic Bus Timing

<SIB>

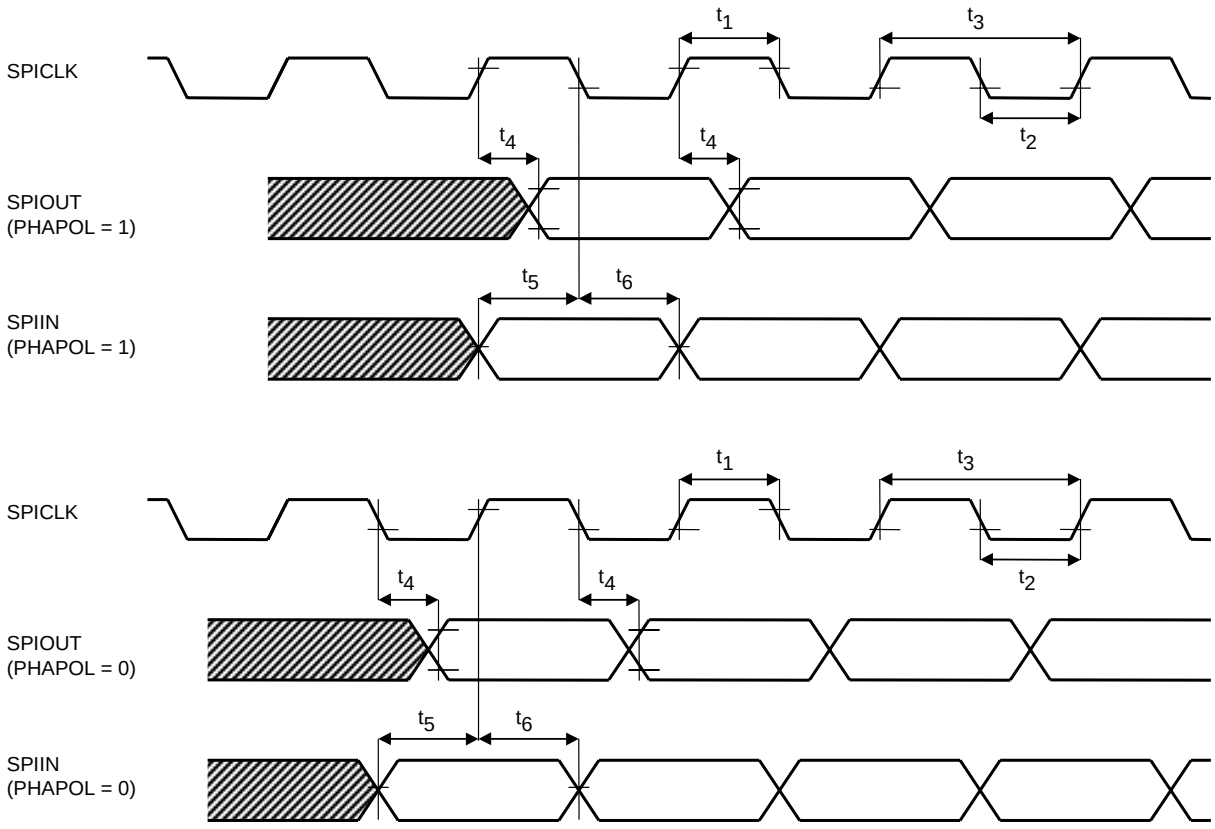
Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	SIBMCLK high time	—	20	—	ns
t_2	SIBMCLK low time	—	20	—	ns
t_3	SIBMCLK period	—	50	—	ns
t_4	Delay SIBMCLK (Master) to SIBSCLK	Rising	—	5	ns
t_5	Delay SIBMCLK (Master) to SIBSCLK	Falling	—	5	ns
t_6	Delay SIBSCLK Rising to SIBSYNC	Rising	—	2	ns
t_7	Delay SIBSCLK Rising to SIBSYNC	Falling	—	2	ns
t_8	Delay SIBSCLK Rising to SIBDOUT	—	—	2	ns
t_9	SIBDIN to SIBSCLK Falling Setup time	—	20	—	ns
t_{10}	SIBDIN to SIBSCLK Falling Hold time	—	0	—	ns



SIB Timing

<SPI>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	SPICLK high time	—	120	—	ns
t_2	SPICLK low time	—	120	—	ns
t_3	SPICLK period	—	250	—	ns
t_4	Delay SPICLK Rising/Falling to SPIOUT	—	—	5	ns
t_5	SPIIN to SPICLK Rising/Falling Setup time	—	15	—	ns
t_6	SPIIN to SPICLK Rising/Falling Hold time	—	15	—	ns

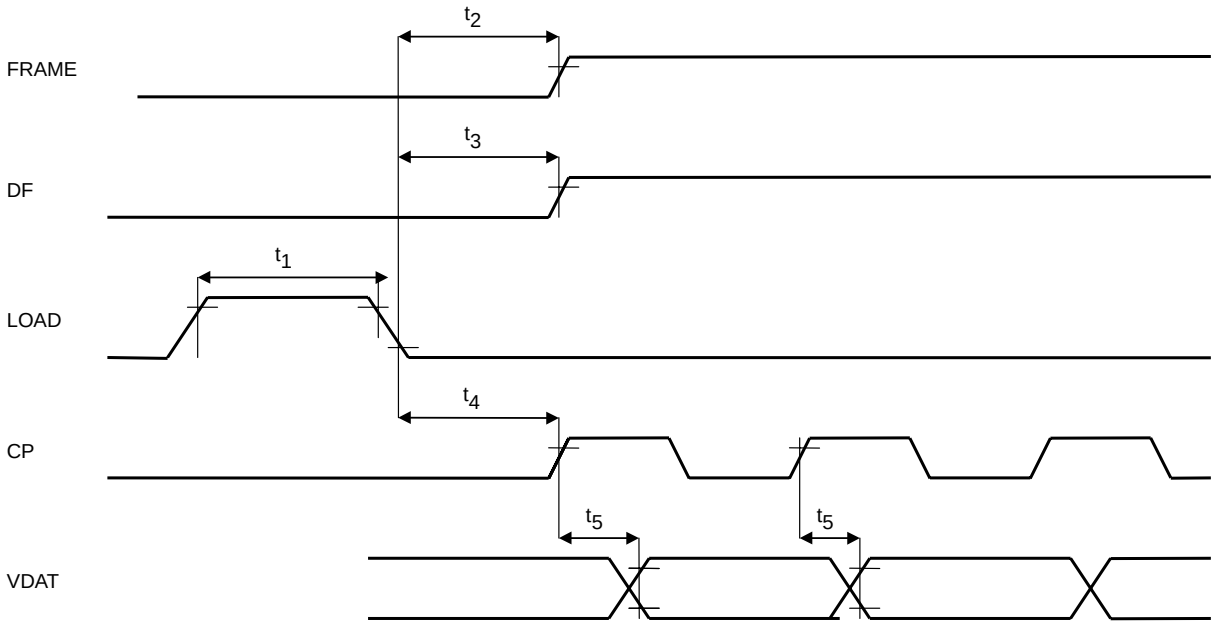


SPI Timing

<VIDEO>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	LOAD Pulse width	—	80	—	ns
t_2	Delay LOAD Falling to FRAME	—	80	—	ns
t_3	Delay LOAD Falling to DF	—	80	—	ns
t_4	Delay LOAD Falling to CP	—	80	—	ns
t_5	Delay CP Rising to VDAT [3:0]	—	—	5	ns

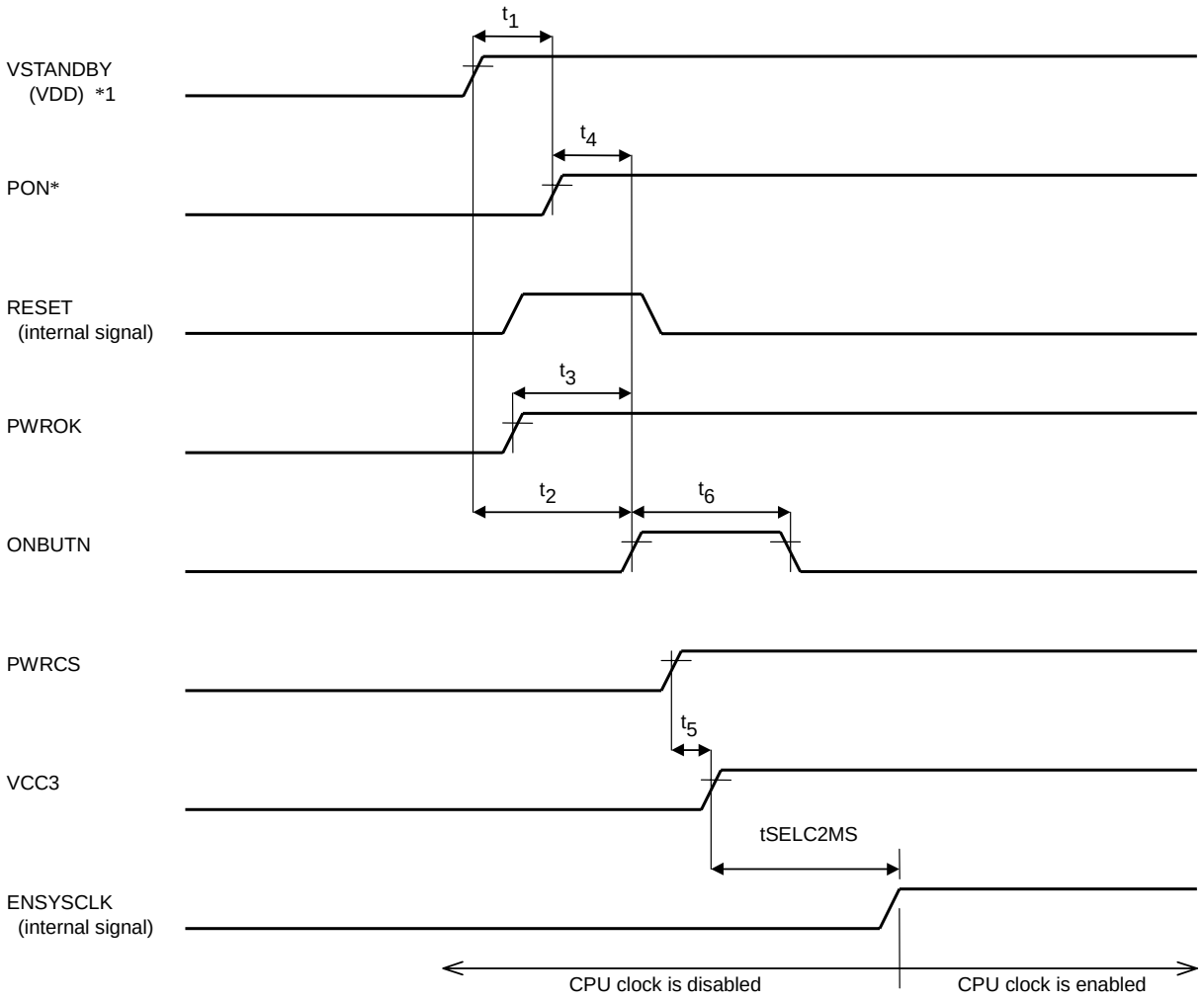
Note: Values shown assume a 92 MHz Video master clock. Min and Max values are programmable using BAUDVAL [4:0] at Video Control Registers 1.



VIDEO Timing

<POWER>

Symbol	Parameter	Rising/Falling	Min	Max	Unit
t_1	VSTANDBY to PON* Rising	—	50	—	ms
t_2	VSTANDBY to ONBUTN delay time	—	2	—	s
t_3	PWROK Rising to ONBUTN Rising	—	2	—	s
t_4	PON* Rising to ONBUTN Rising	—	1	—	us
t_5	PWRCS Rising to VCC3 Rising	—	0	—	s
t_6	ONBUTN pulse width	—	1	—	us

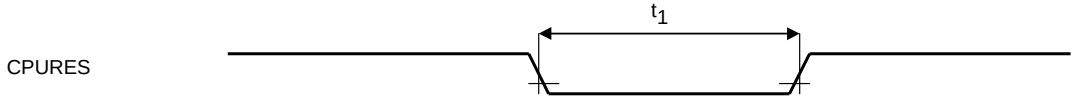


Power Timing

*1 This signal is power for the TMPR3912 and other components in the system that must never lose power. That means VDD iteself for TMPR3912.

<CPU RESET>

Item	Parameter	Rising/Falling	Min	Max	Unit
t_1	CPURES* low time	—	10	—	ns

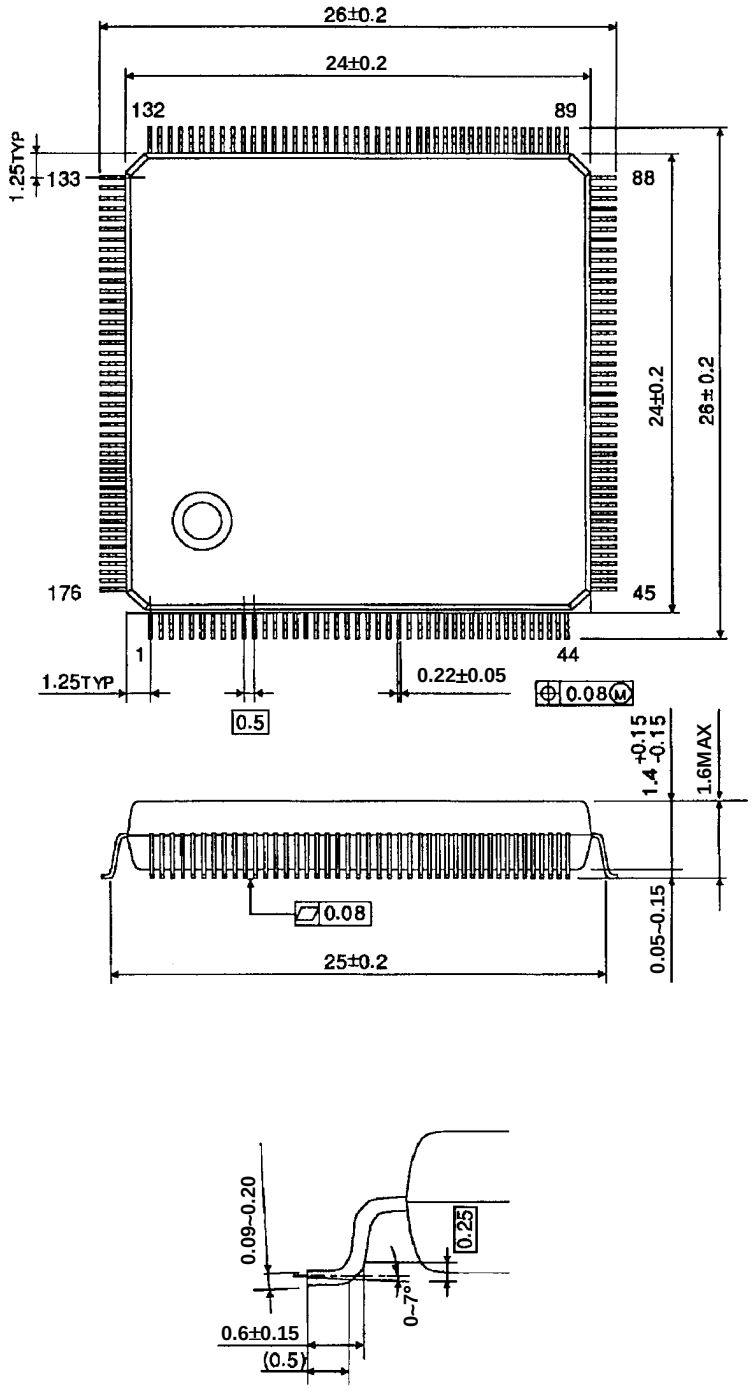


CPU Reset Timing

19. Package Dimension

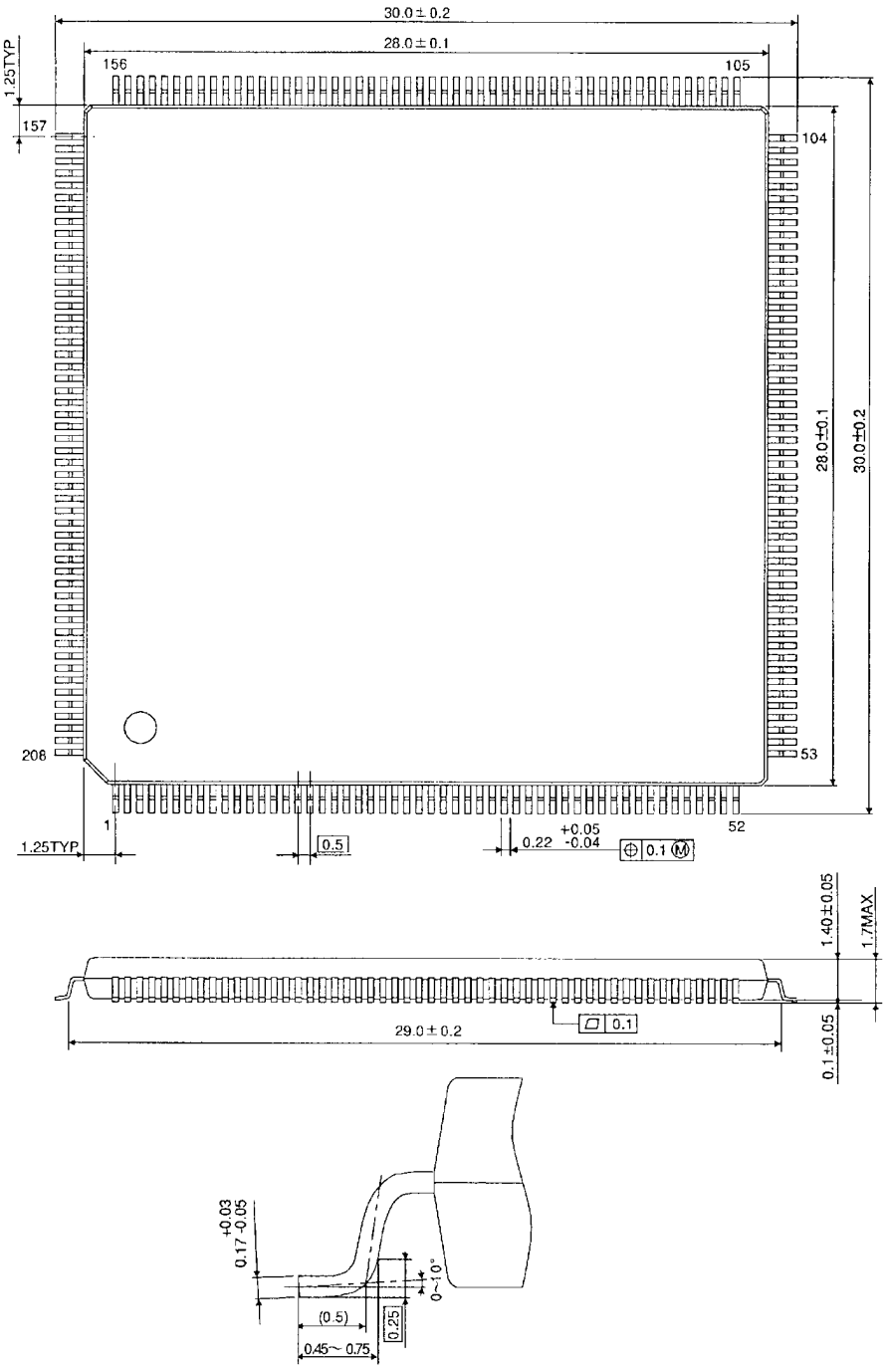
19.1 TMPR3911BU

P-LQFP176-2424-0.50A Unit: mm



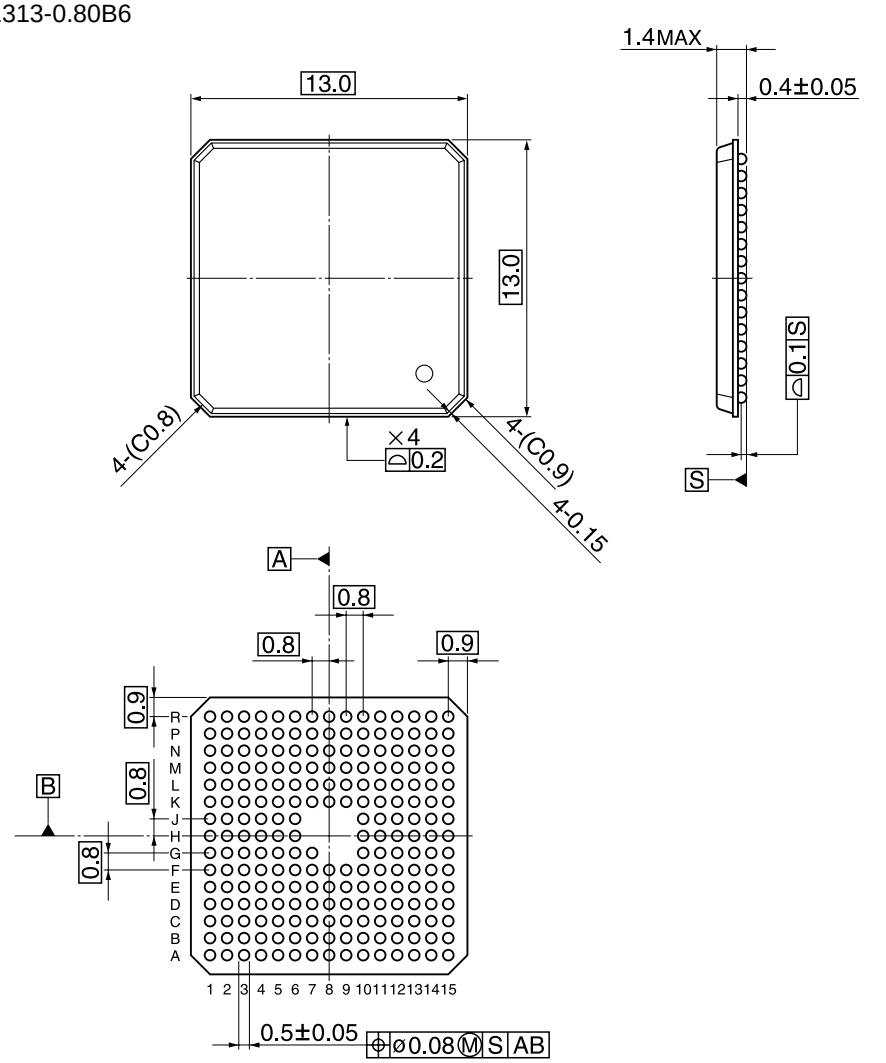
19.2 TMPR3912AU-92

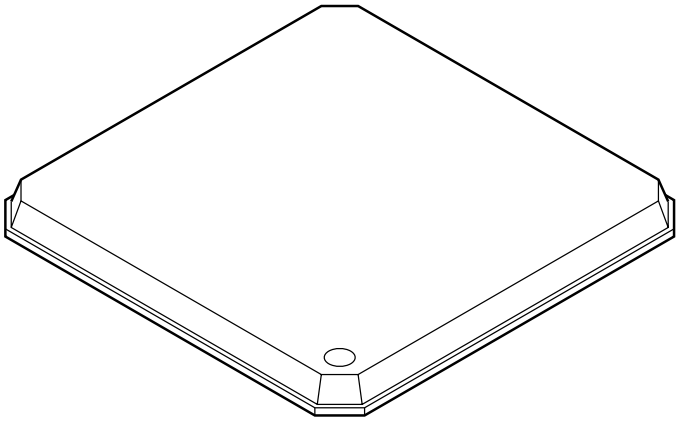
P-LQFP208-2828-0.50A Unit: mm



19.3 TMPR3912XB-92

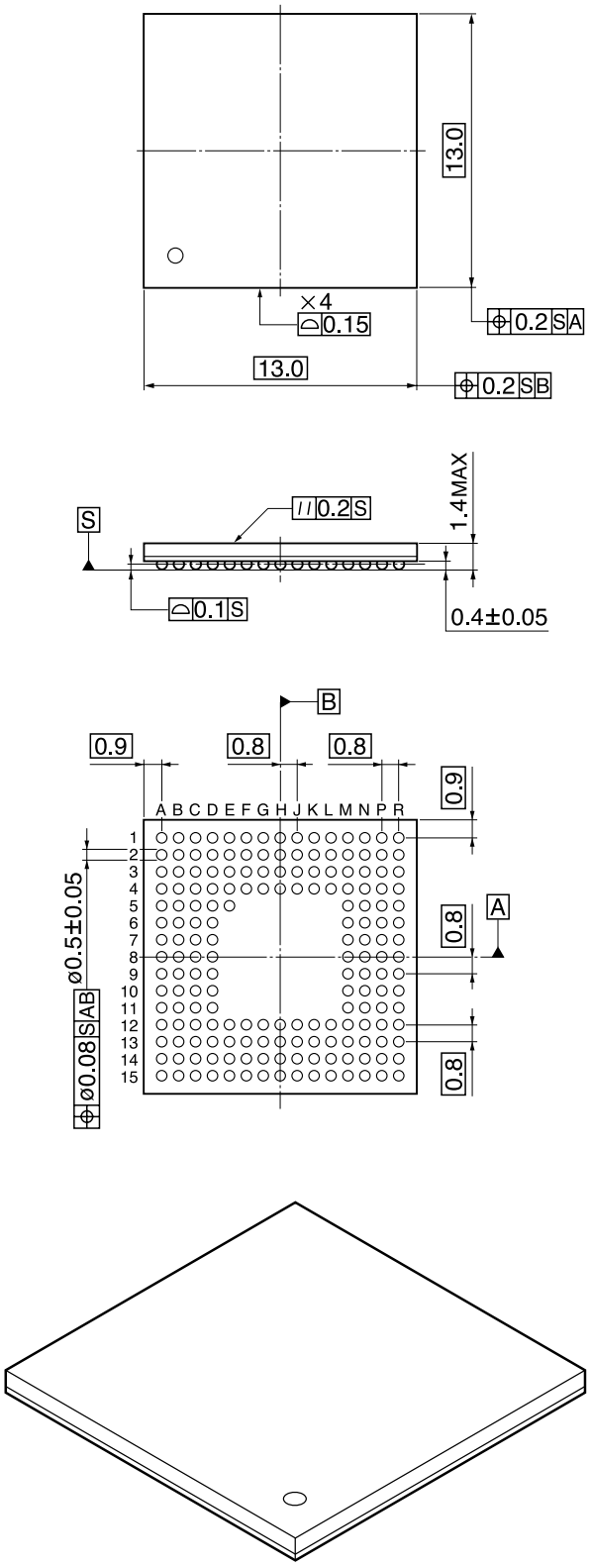
P-FBGA217-1313-0.80B6 Unit: mm





19.4 TMPR3911BXB

P-FBGA177-1313-0.80C4 Unit: mm



Appendix A. Differences among TMPR3911/3912

Table A-1 The Differences among TMPR3911BU/BXB and TMPR3912AU-92/XB-92

	TMPR3912AU-92 TMPR3912XB-92	TMRP3911BU TMPR3911BXB
CPU clock	92.160 MHz	58.9824 MHz
SYSCLKIN clock	11.520 MHz	7.3728 MHz
Independent set up for the port size of the PCMCIA card slots 1 and 2	Available	Available
DCLKOUT Output Control	Available	Available
TX3911/12 Revision Register Read Data	$01	$02
NMI by PWROK input	Available	Available
Clock Rate of CSERCLK Master Clock	CLK2X / 2	CLK2X / 4
Baud Rate Register of UART	11 bits (BAUDRATE[10:0])	10 bits (BAUDRATE[9:0])
4 bank SDRAM support	Supported	Supported
Package	LQFP208 (AU-92) FBGA217 (XB-92)	LQFP176 (BU) FBGA177 (BXB)

Appendix B. Access Timing Chart

Index

Note: Depending on CPU endianness, the signal names of some pins are reorganized (see Table 4.3.1). In the charts in this appendix, Xb[ ] means X[ ] pin in big endian mode, XI[ ] means X[ ] pin in little endian mode. For example, Db[7:0]/DI[31:24] means D[7:0] in big endian mode and D[31:24] in little endian mode. Note that these signals (Db[7:0] and DI[31:24]) are assigned to same pins.

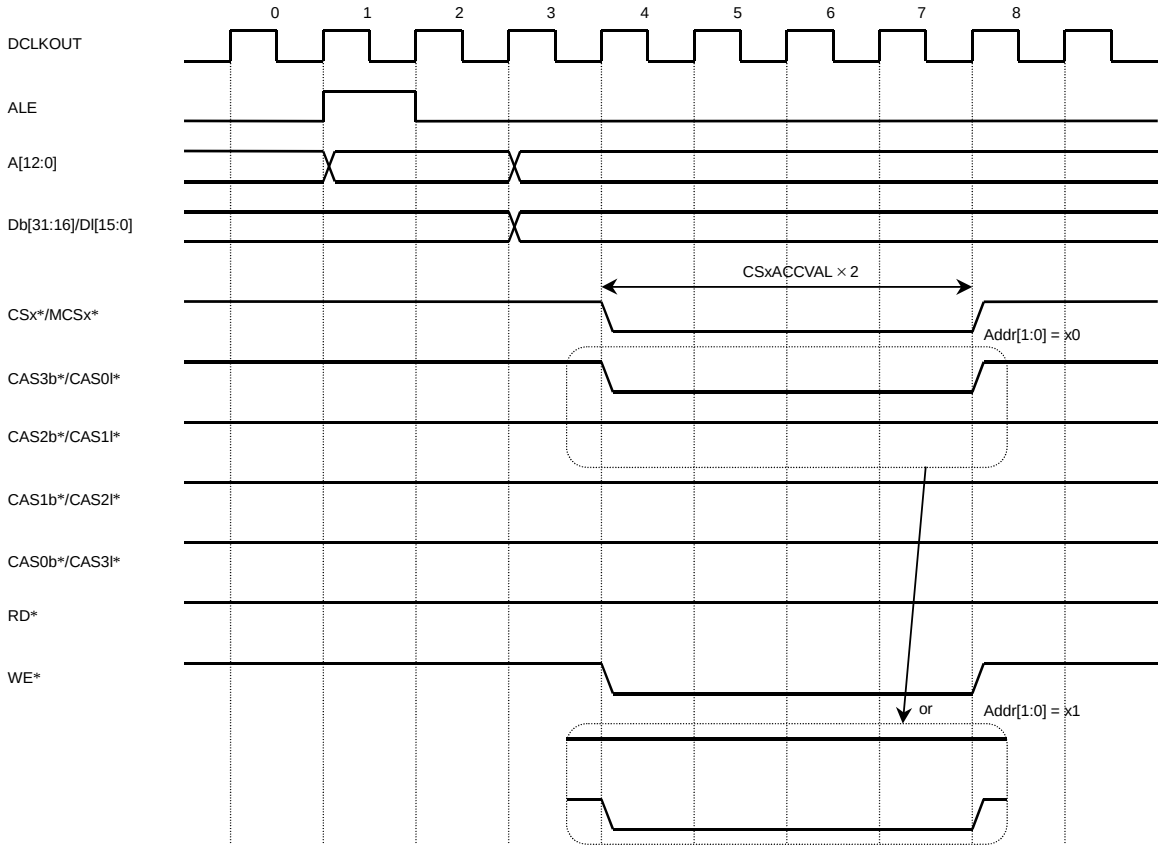


Figure B.1 16-bit Chip Select Device: Write Operation (1) (Byte)

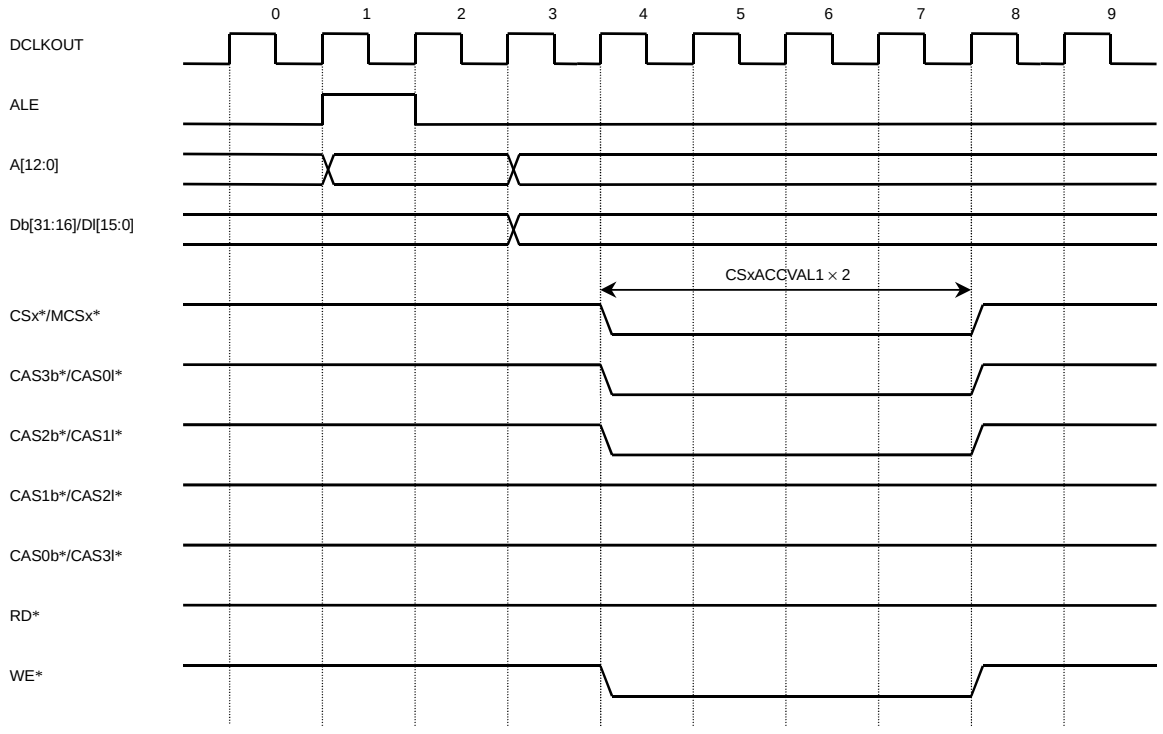


Figure B.2 16-bit Chip Select Device: Write Operation (2) (Halfword)

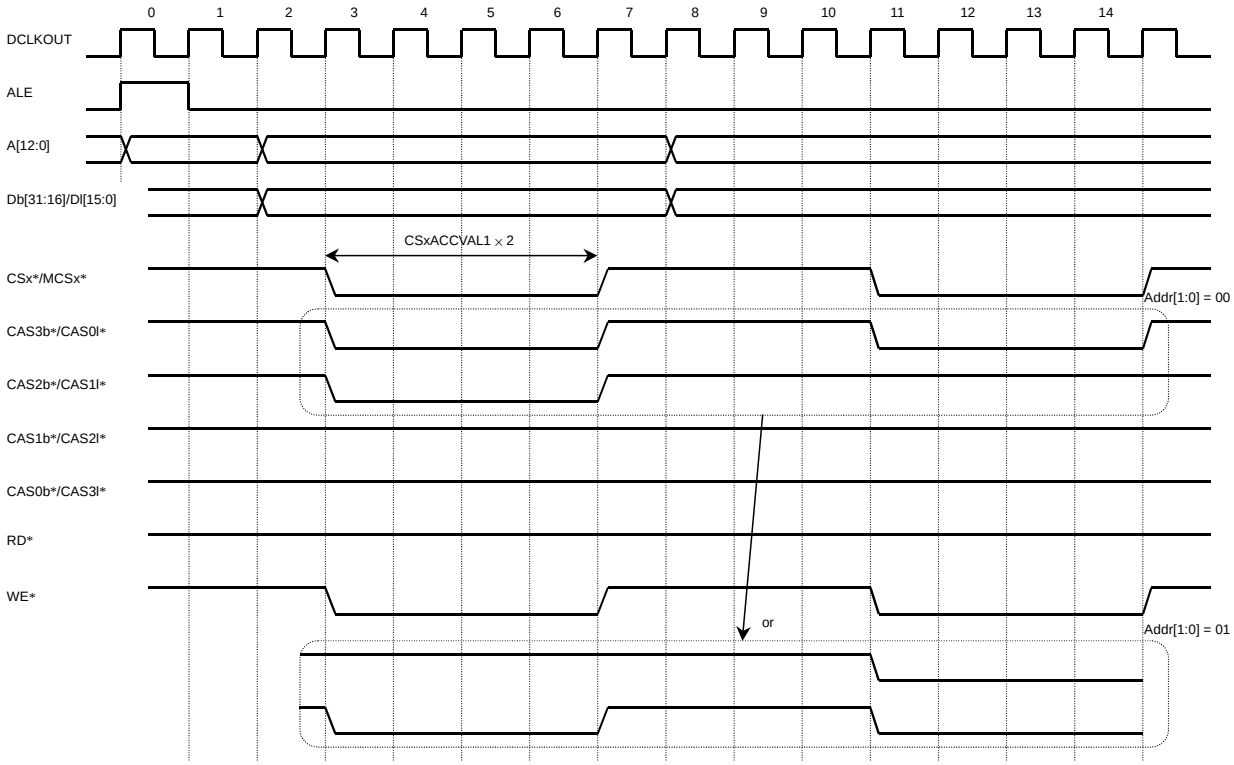


Figure B.3 16-bit Chip Select Device: Write Operation (3) (Triplebyte)

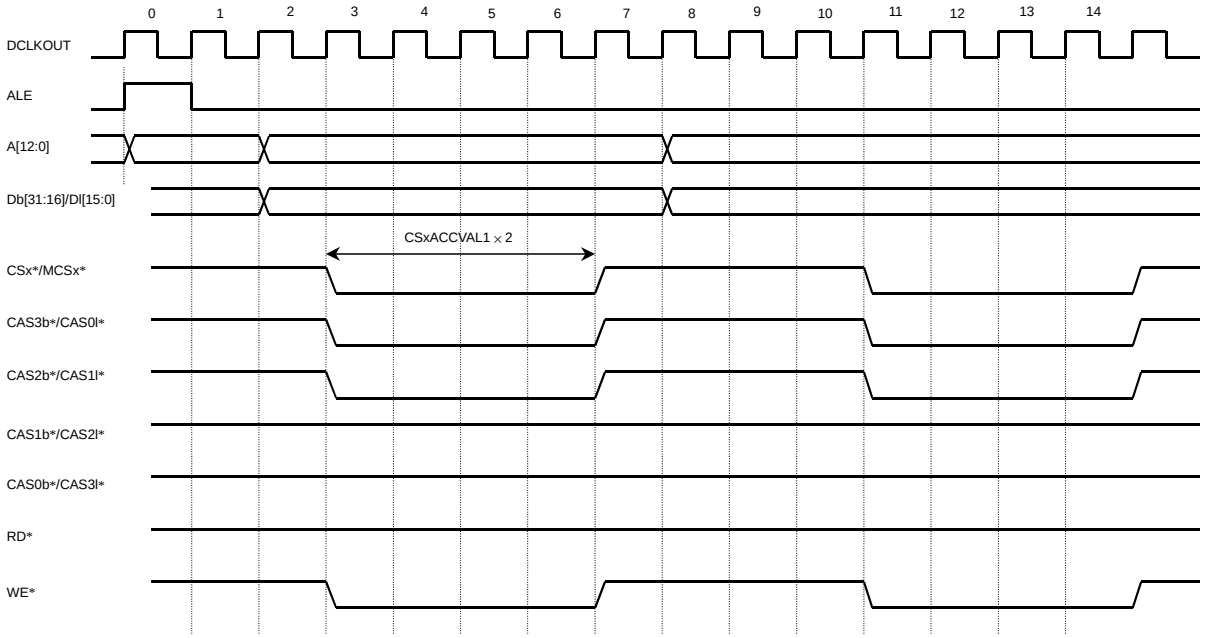


Figure B.4 16-bit Chip Select Device: Write Operation (4) (Word)

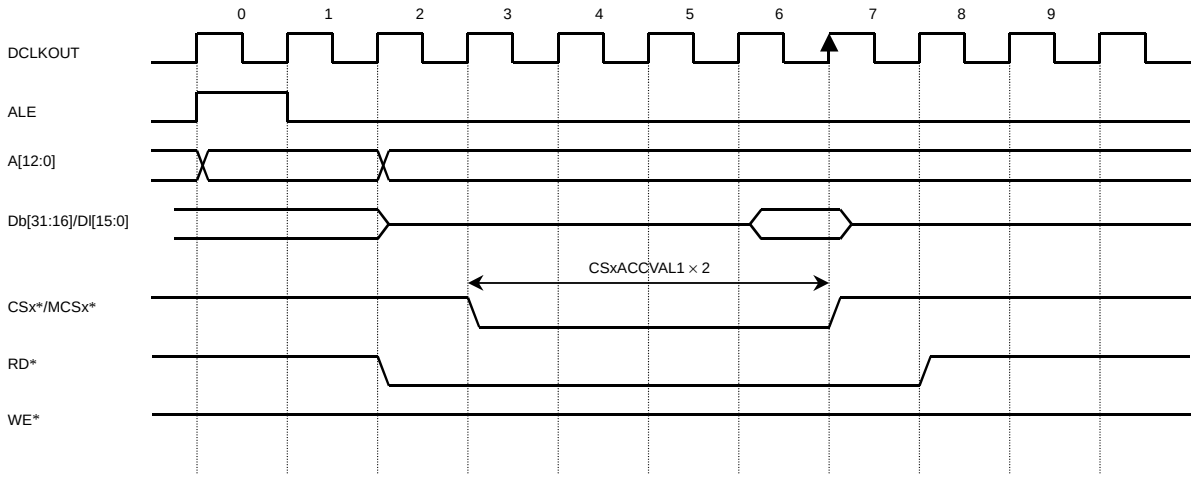


Figure B.5 16-bit Chip Select Device: Read Operation (1) (Byte)

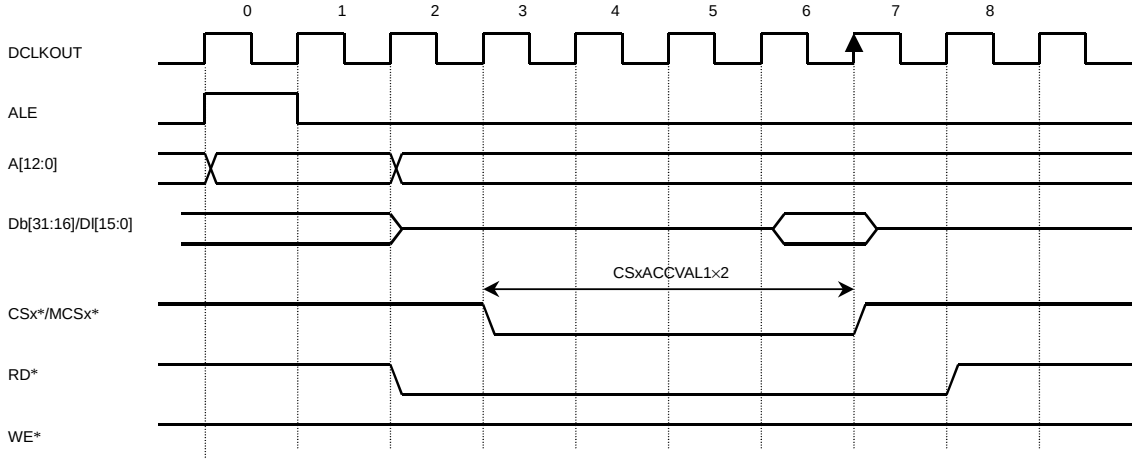


Figure B.6 16-bit Chip Select Device: Read Operation(2) (Halfword)

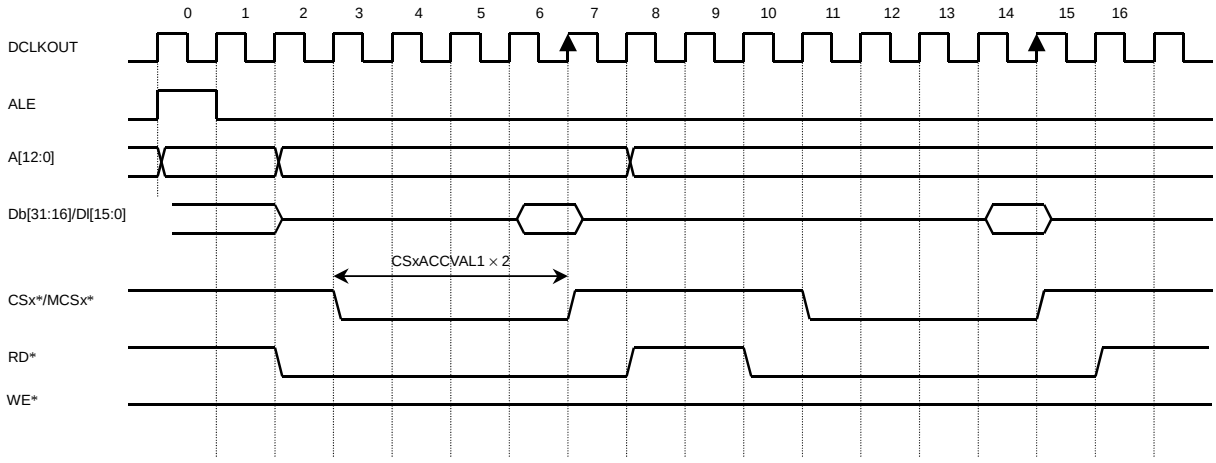


Figure B.7 16-bit Chip Select Device: Read Operation (3) (Triplebyte)

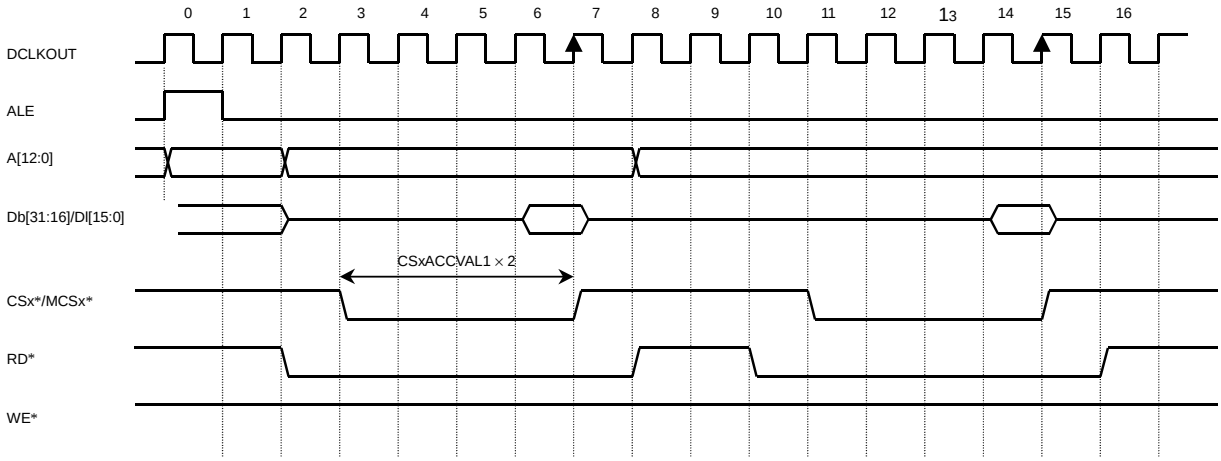


Figure B.8 16-bit Chip Select Device: Read Operation (4) (Word)

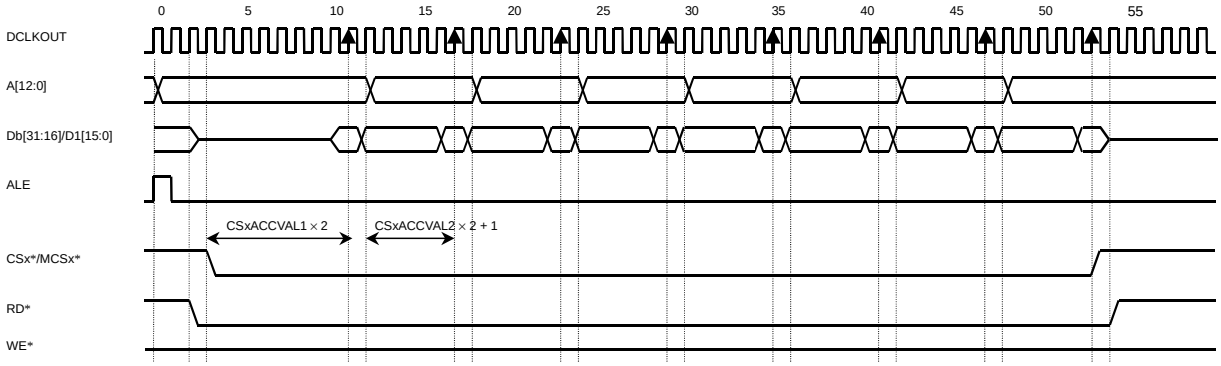


Figure B.9 16-bit Chip Select Device: Read Operation (5) (4 Word Burst)

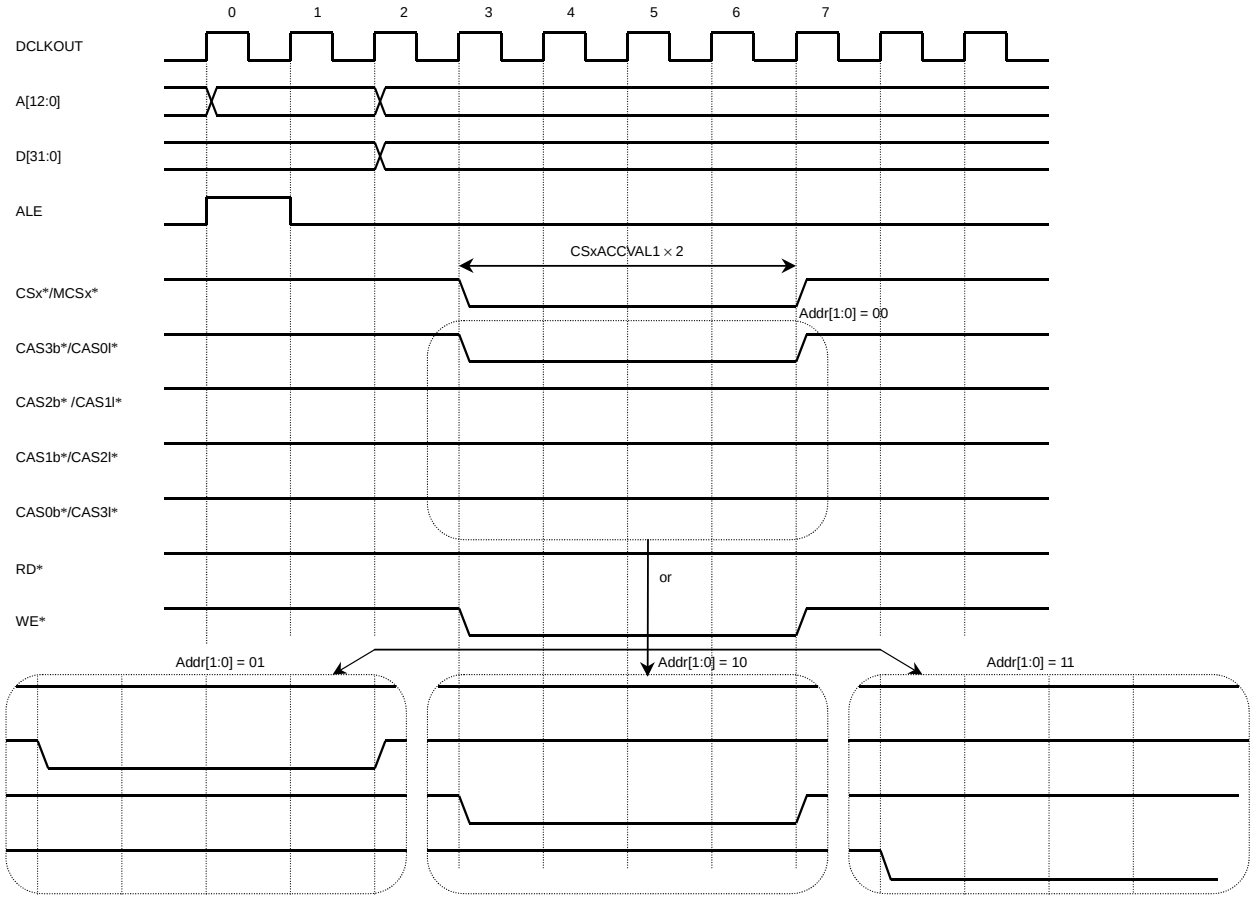


Figure B.10 32-bit Chip Select Device: Write Operation (1) (Byte)

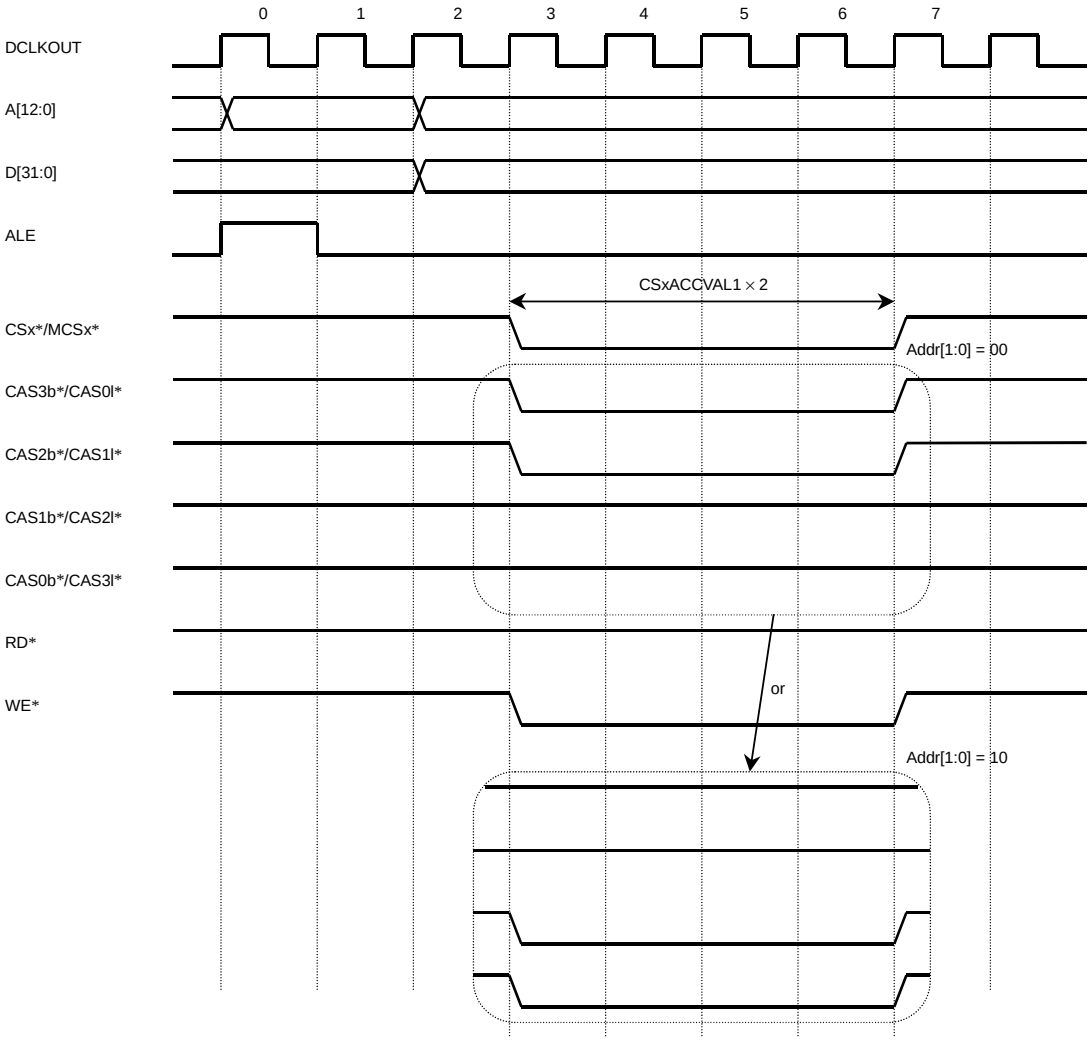


Figure B.11 32-bit Chip Select Device: Write Operation (2) (Halfword)

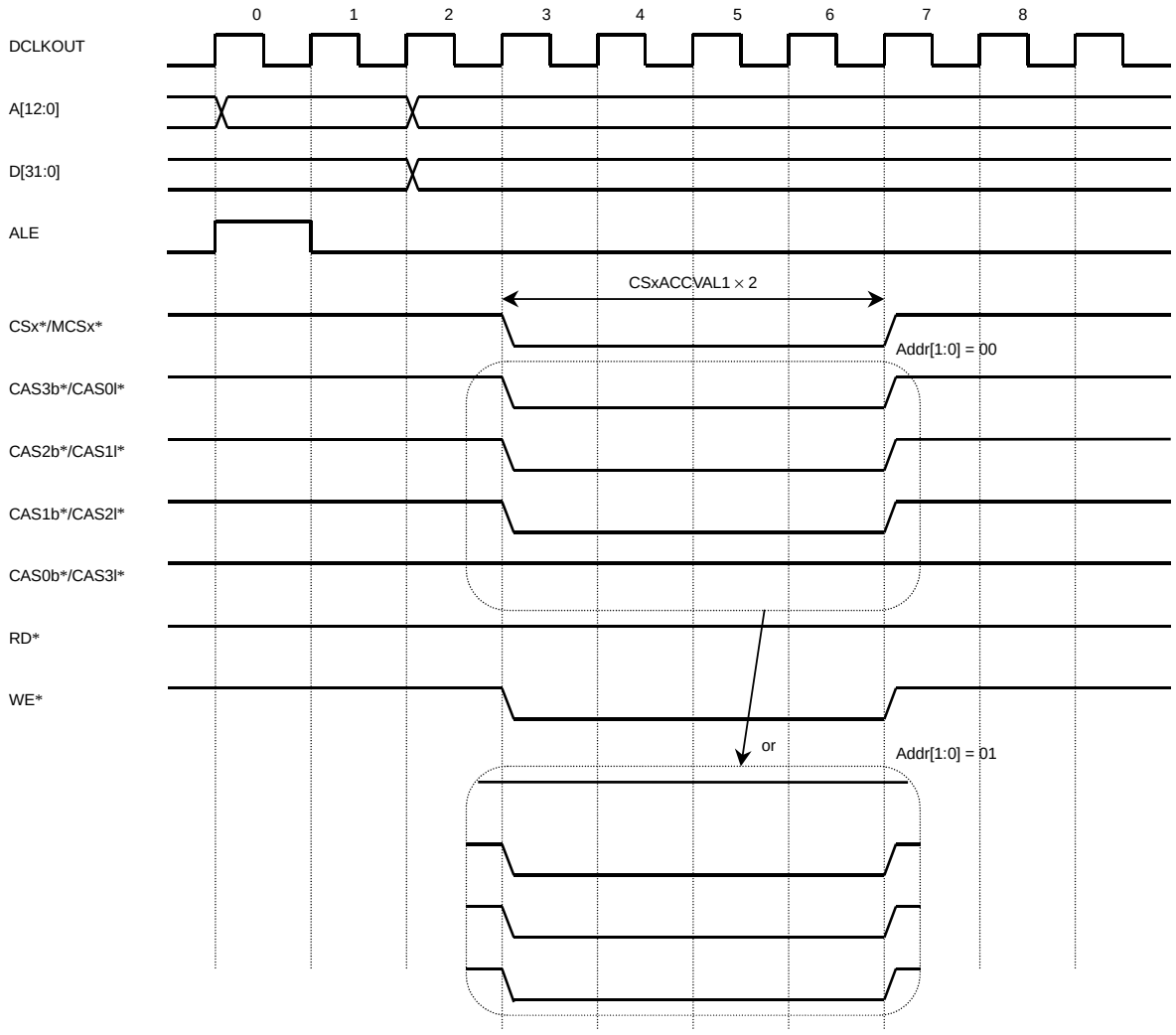


Figure B.12 32-bit Chip Select Device: Write Operation (3) (Triplebyte)

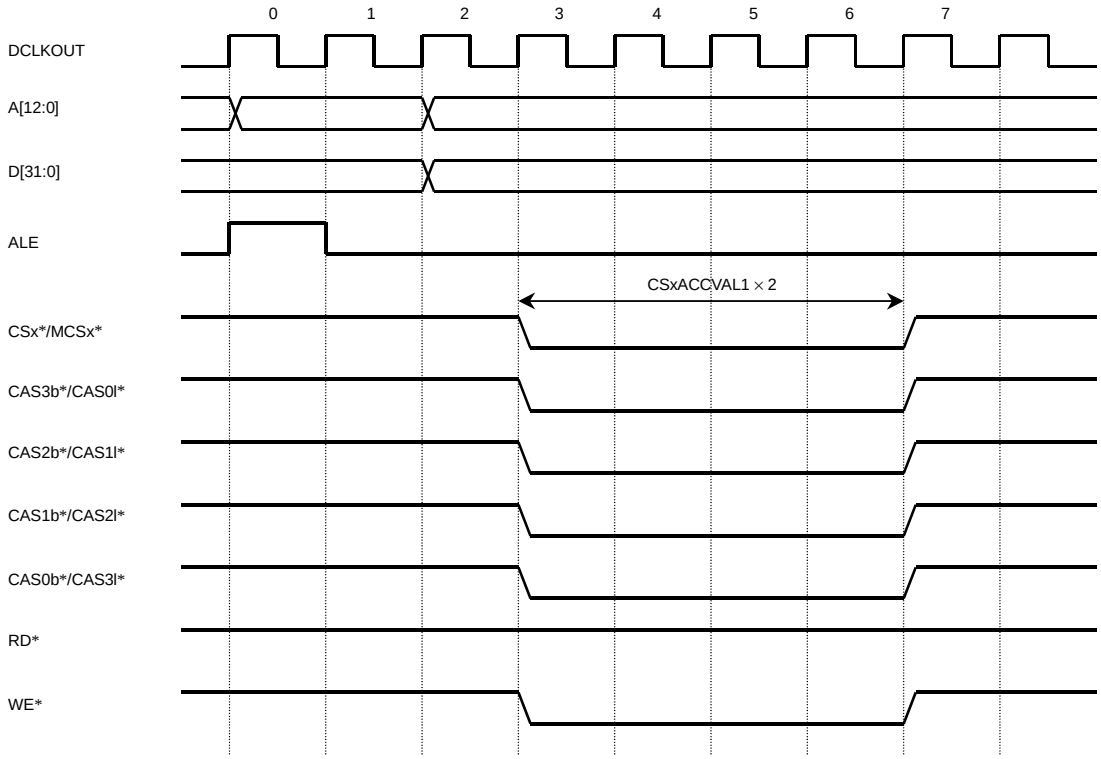


Figure B.13 32-bit Chip Select Device: Write Operation (4) (Word)

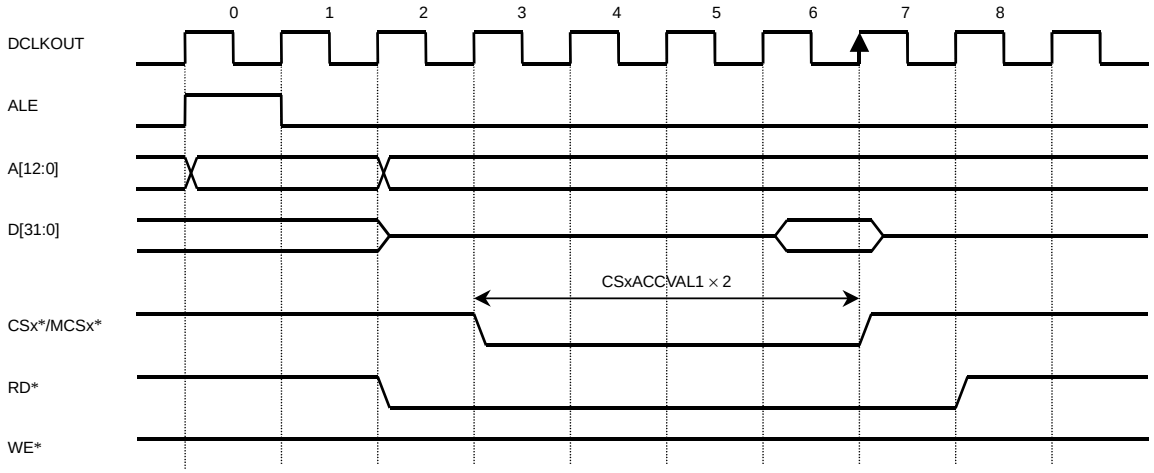


Figure B.14 32-bit Chip Select Device: Read Operation (1) (Byte/Halfword/Triplebyte/Word)

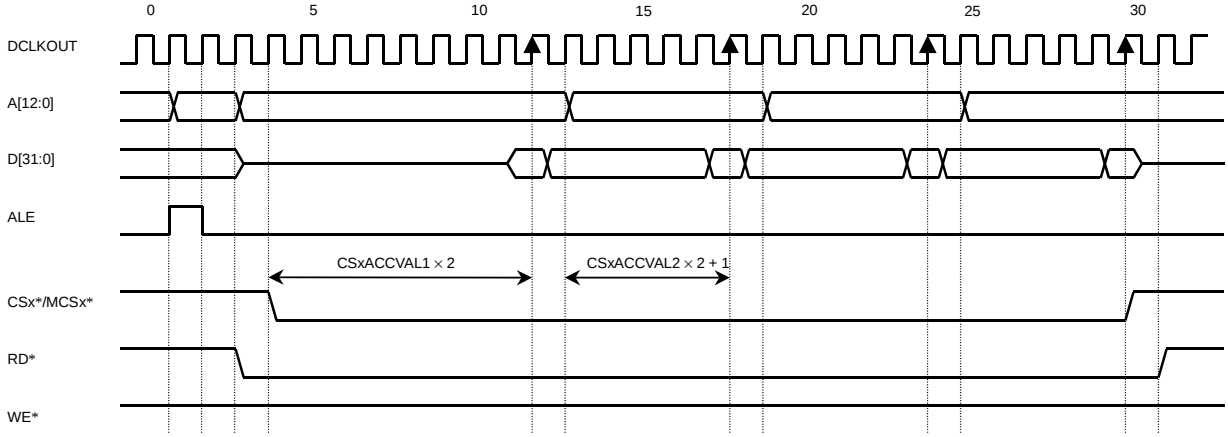


Figure B.15 32-bit Chip Select Device: Read Operation (2) (4 Word Burst)

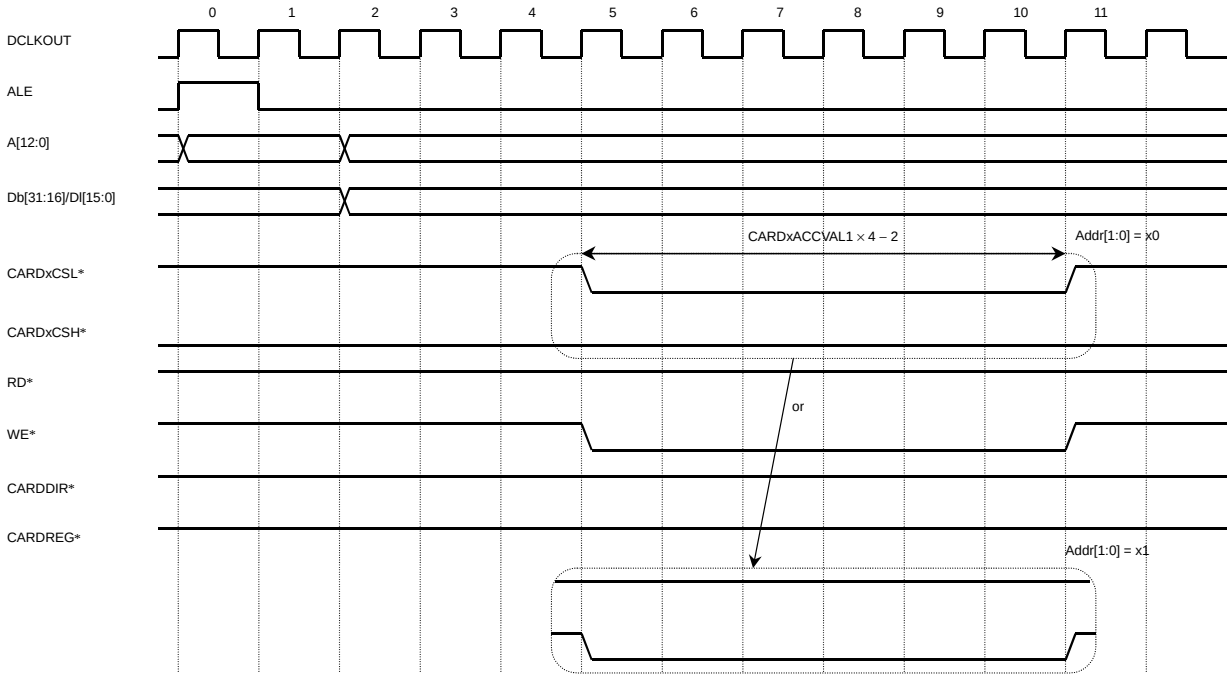


Figure B.16 16-bit CARD Device: Memory Access; Write Operation (1) (Byte)

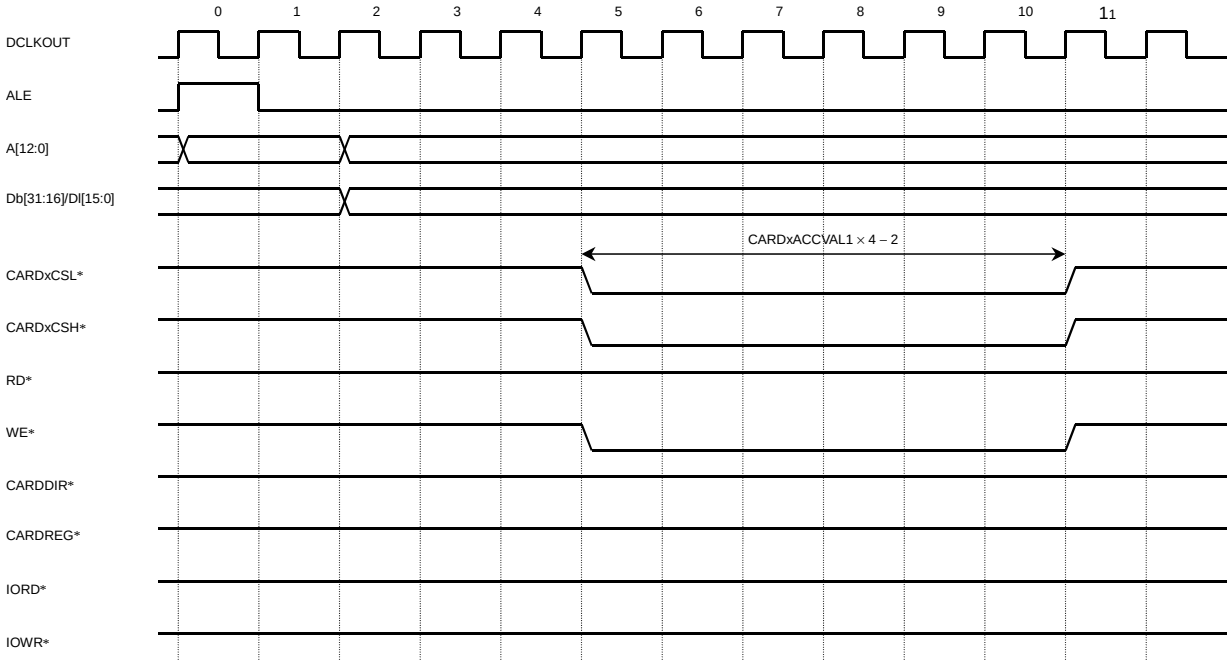


Figure B.17 16-bit CARD Device: Memory Access; Write Operation (2) (Halfword)

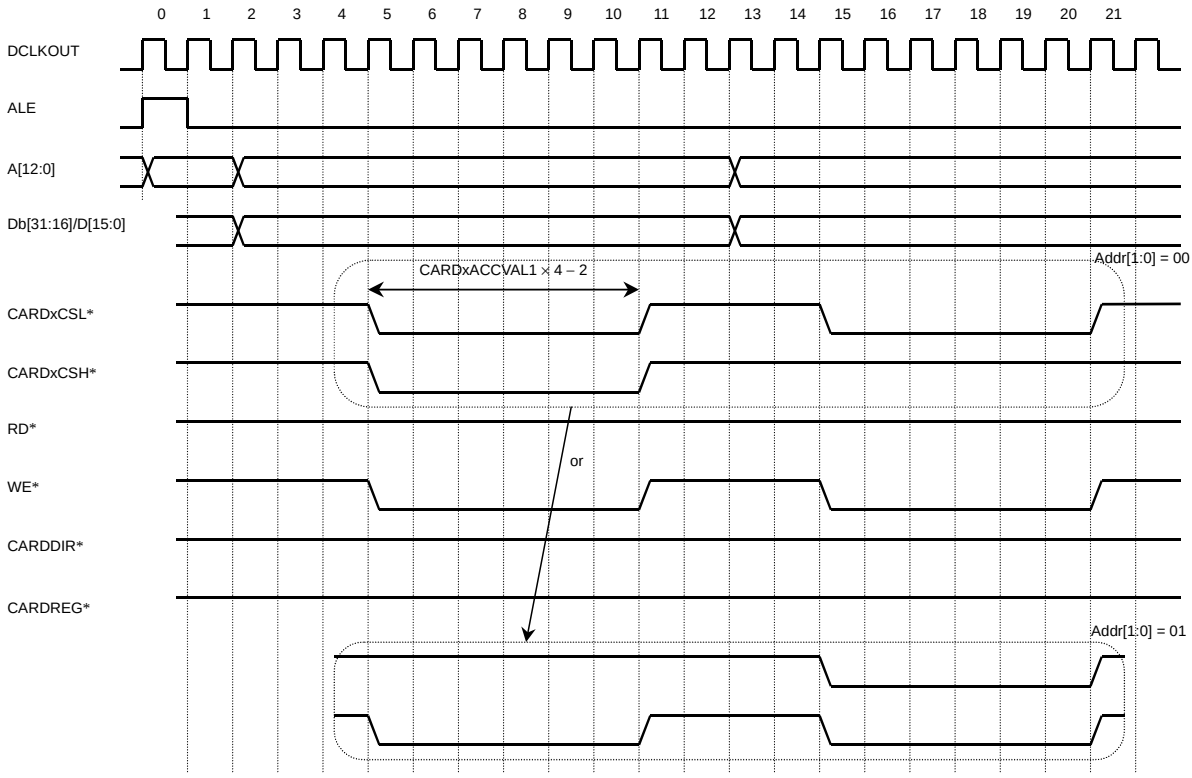


Figure B.18 16-bit CARD Device: Memory Access; Write Operation (3) (Triplebyte)

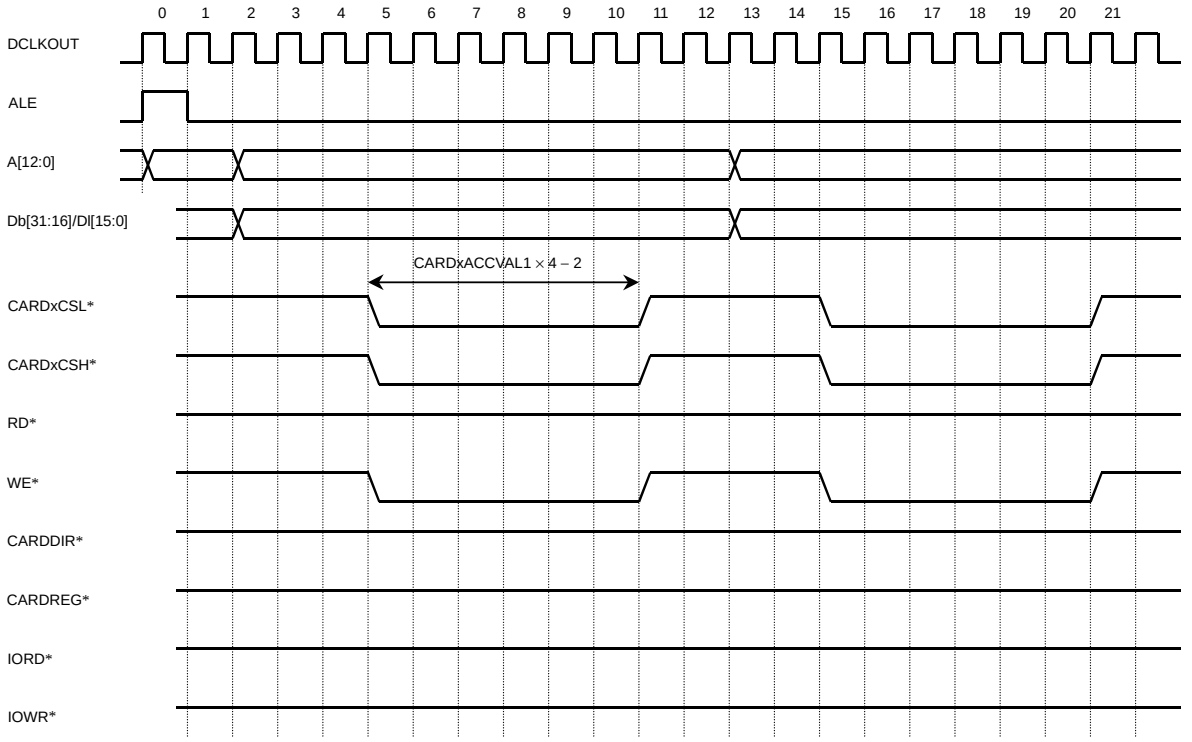


Figure B.19 16-bit CARD Device: Memory Access; Write Operation (4) (Word)

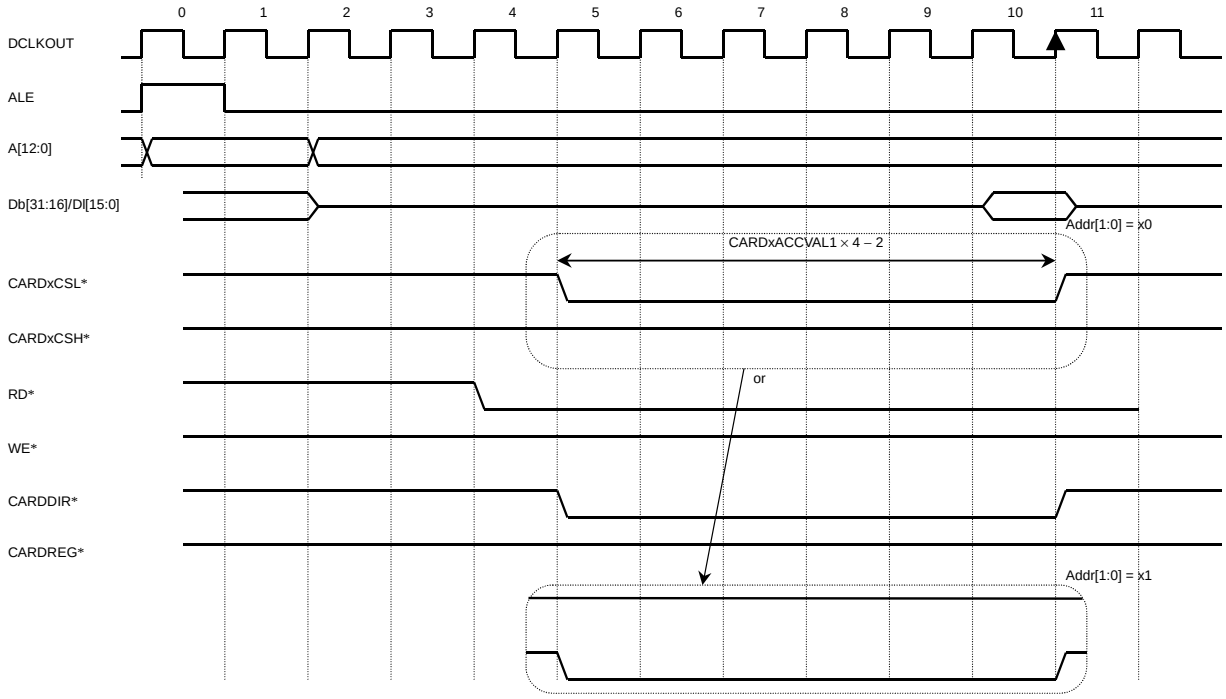


Figure B.20 16-bit CARD Device: Memory Access; Read Operation (1) (Byte)

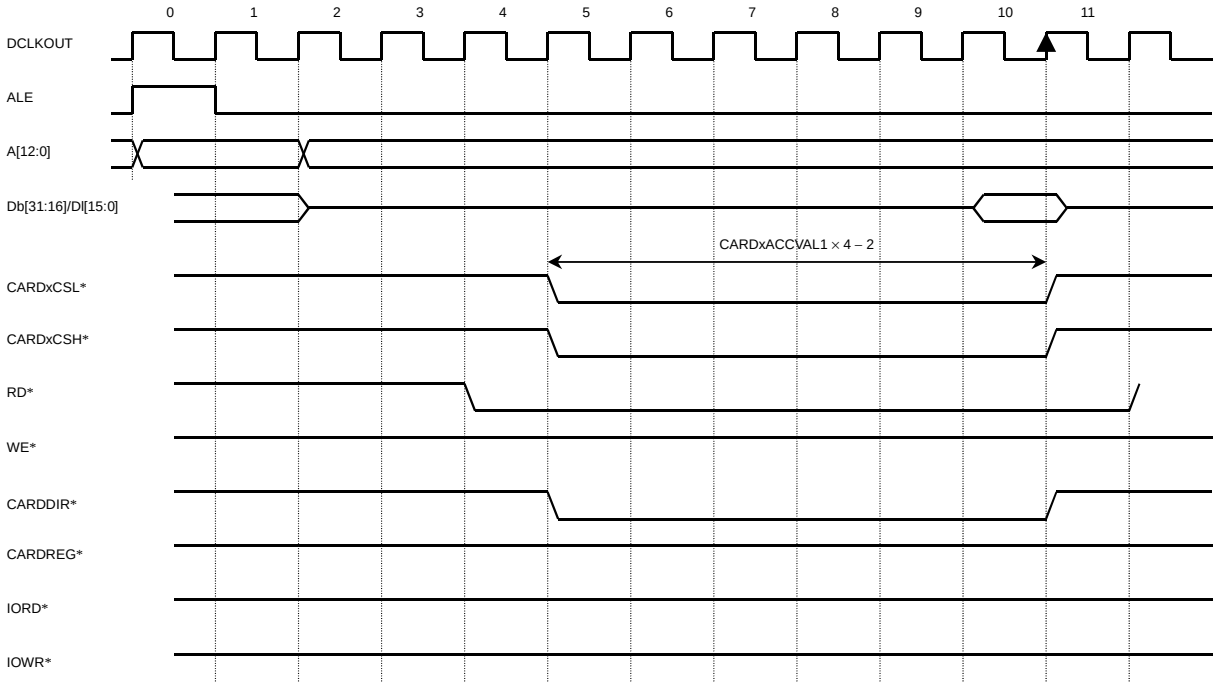


Figure B.21 16-bit CARD Device: Memory Access; Read Operation (2) (Half Word)

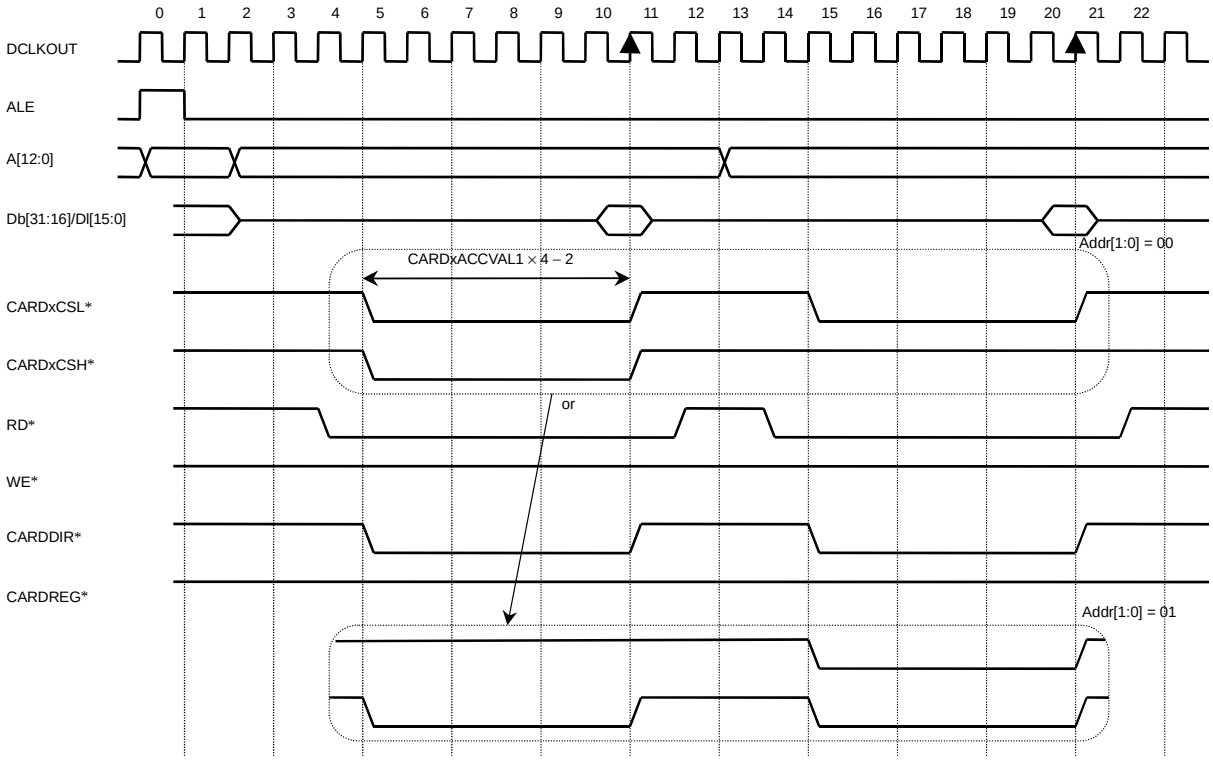


Figure B.22 16-bit CARD Device: Memory Access; Read Operation (3) (Triplebyte)

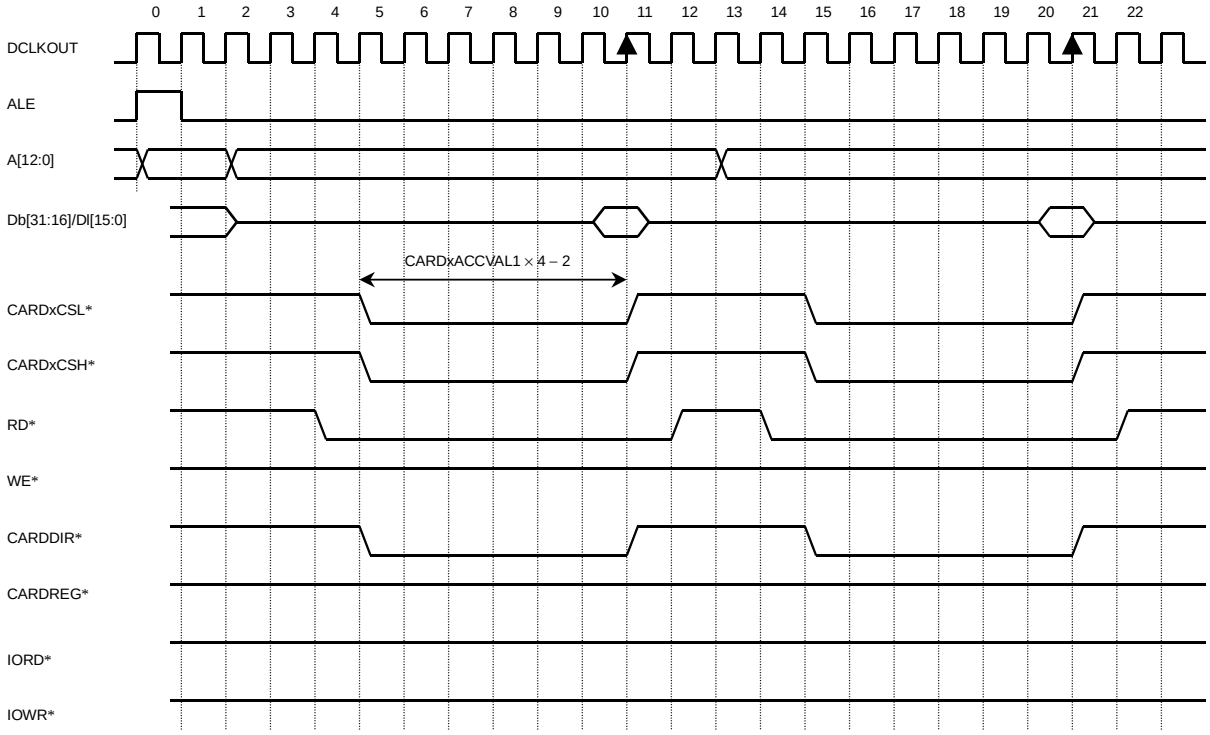


Figure B.23 16-bit CARD Device: Memory Access; Read Operation (4) (Word)

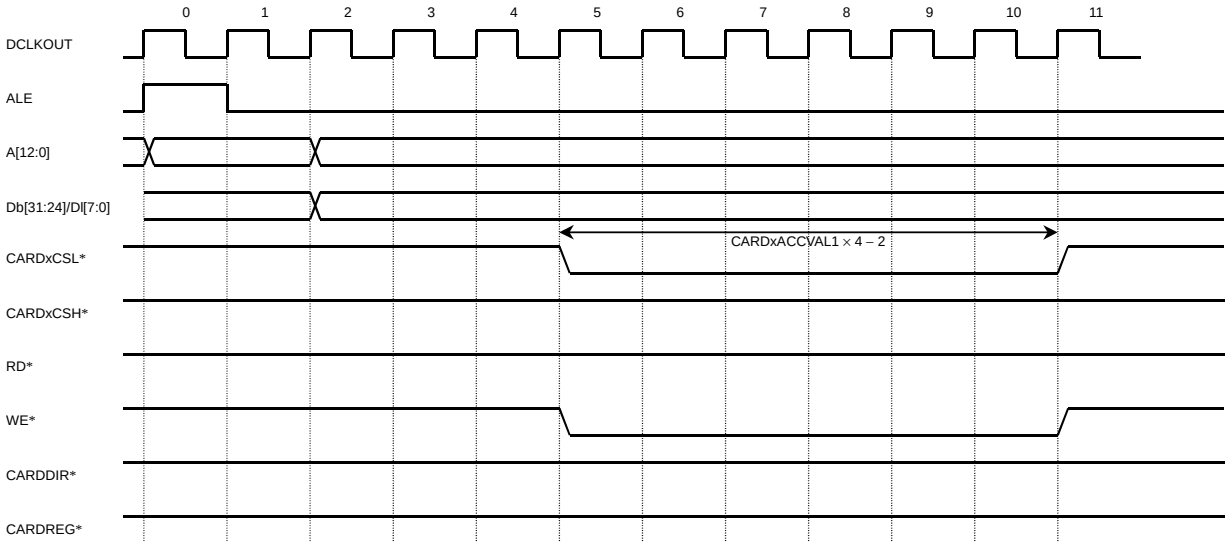


Figure B.24 8-bit CARD Device: Memory Access; Write Operation (1) (Byte)

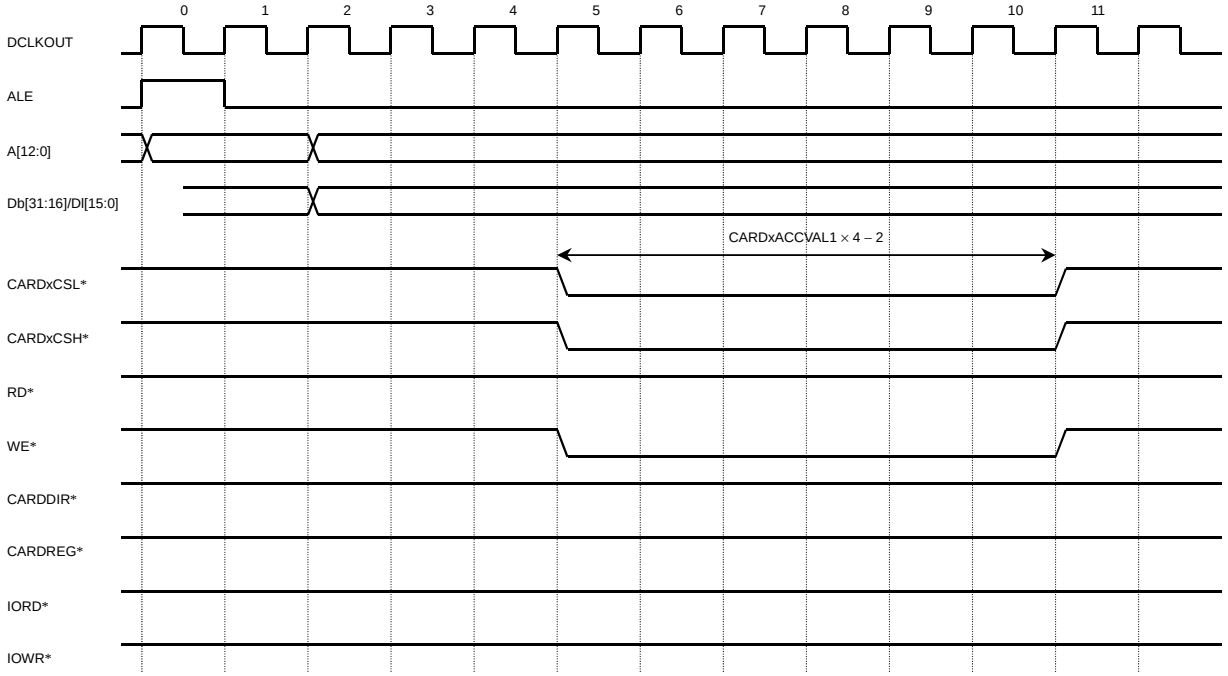


Figure B.25 8-bit CARD Device: Memory Access; Write Operation (2) (Halfword)

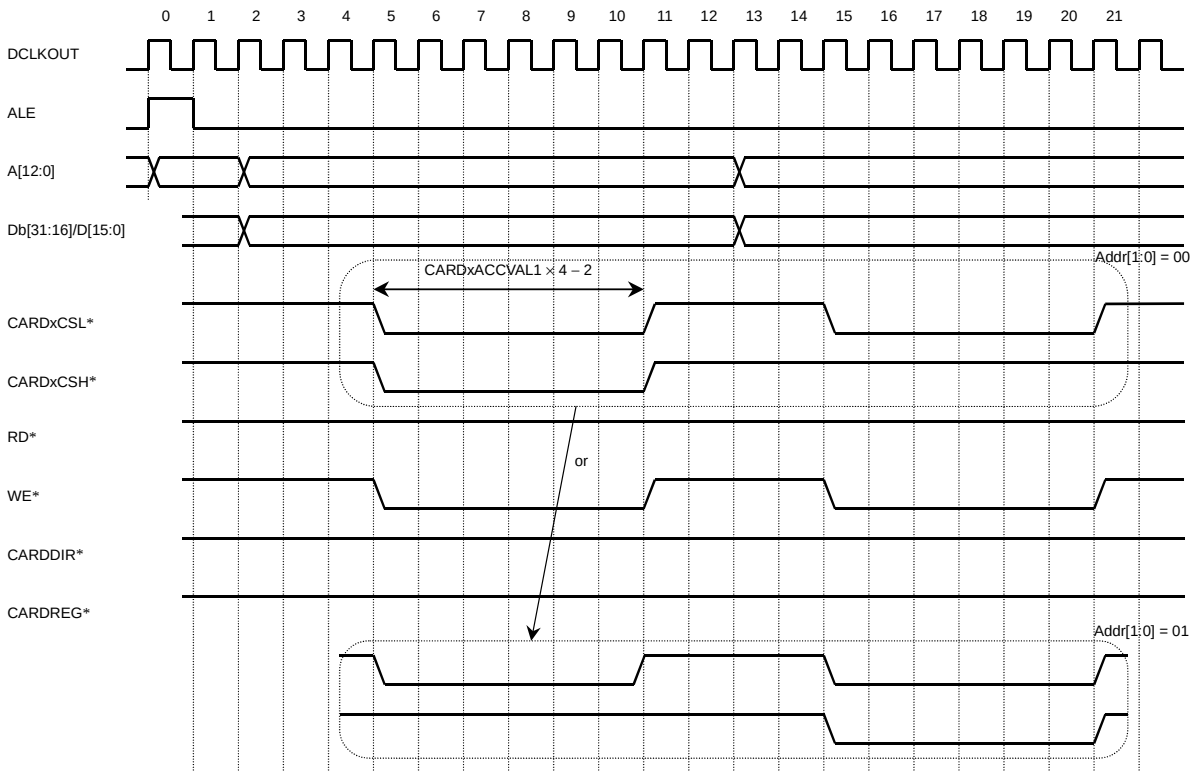


Figure B.26 8-bit CARD Device: Memory Access; Write Operation (3) (Triplebyte)

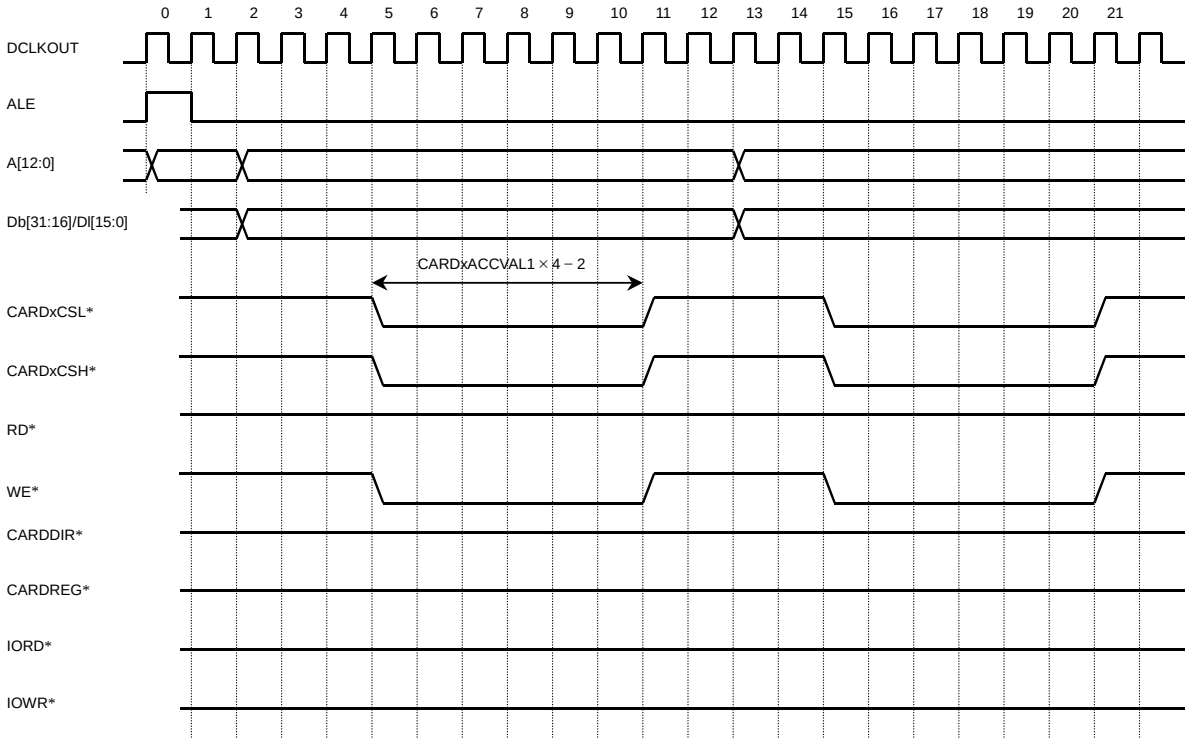


Figure B.27 8-bit CARD Device: Memory Access; Write Operation (4) (Word)

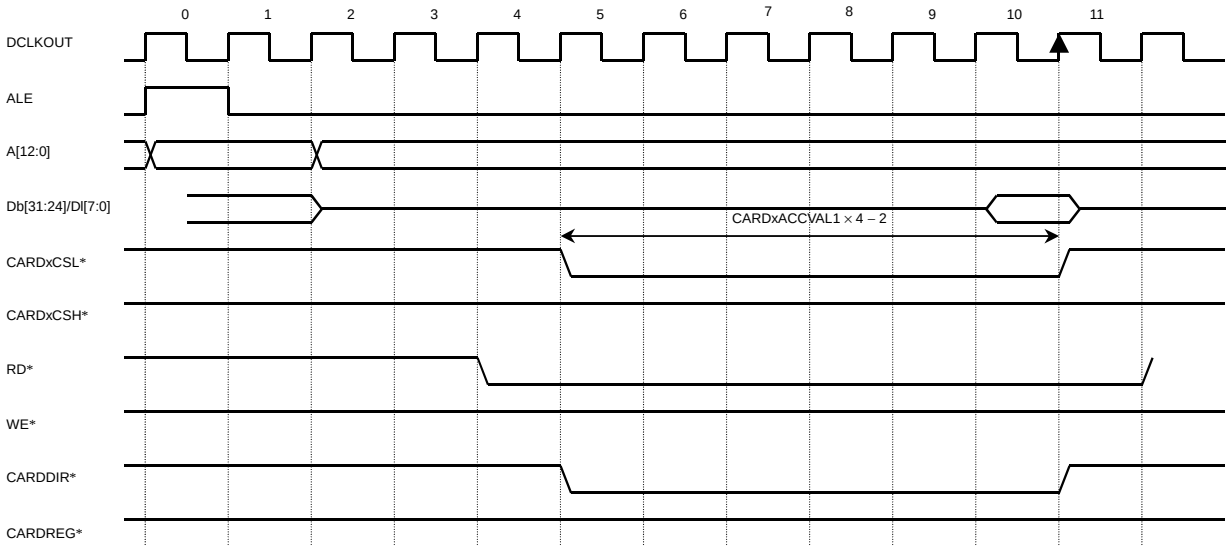


Figure B.28 8-bit CARD Device: Memory Access; Read Operation (1) (Byte)

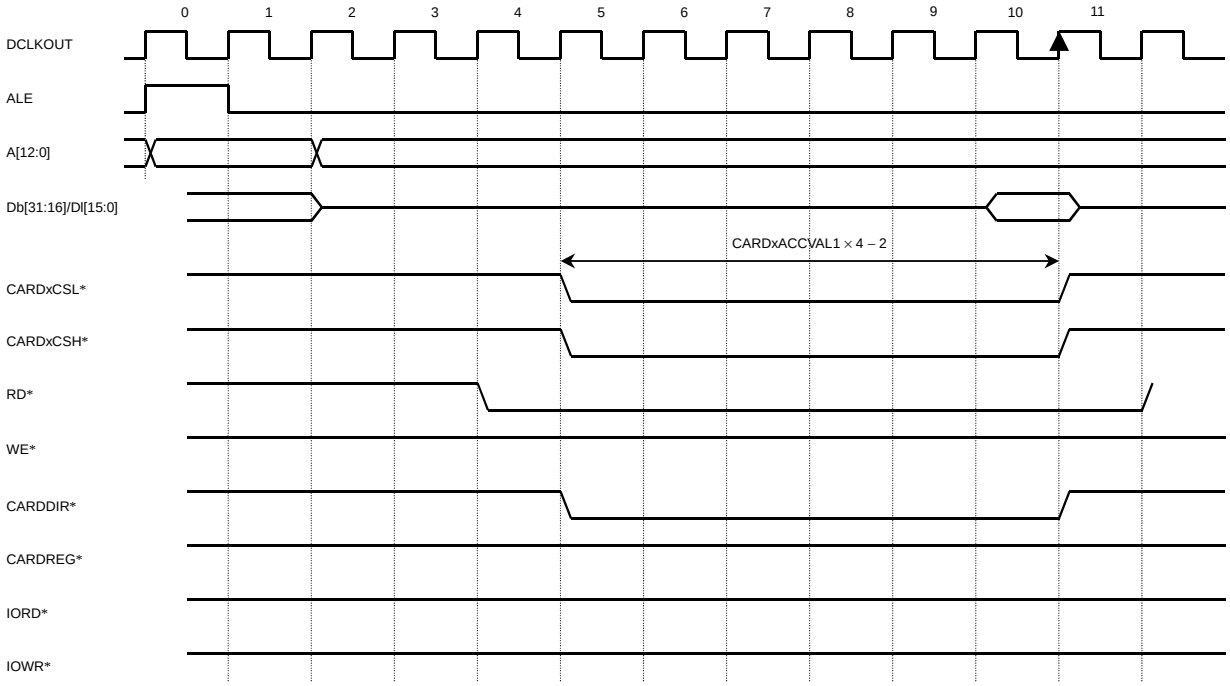


Figure B.29 8-bit CARD Device: Memory Access; Read Operation (2) (Half Word)

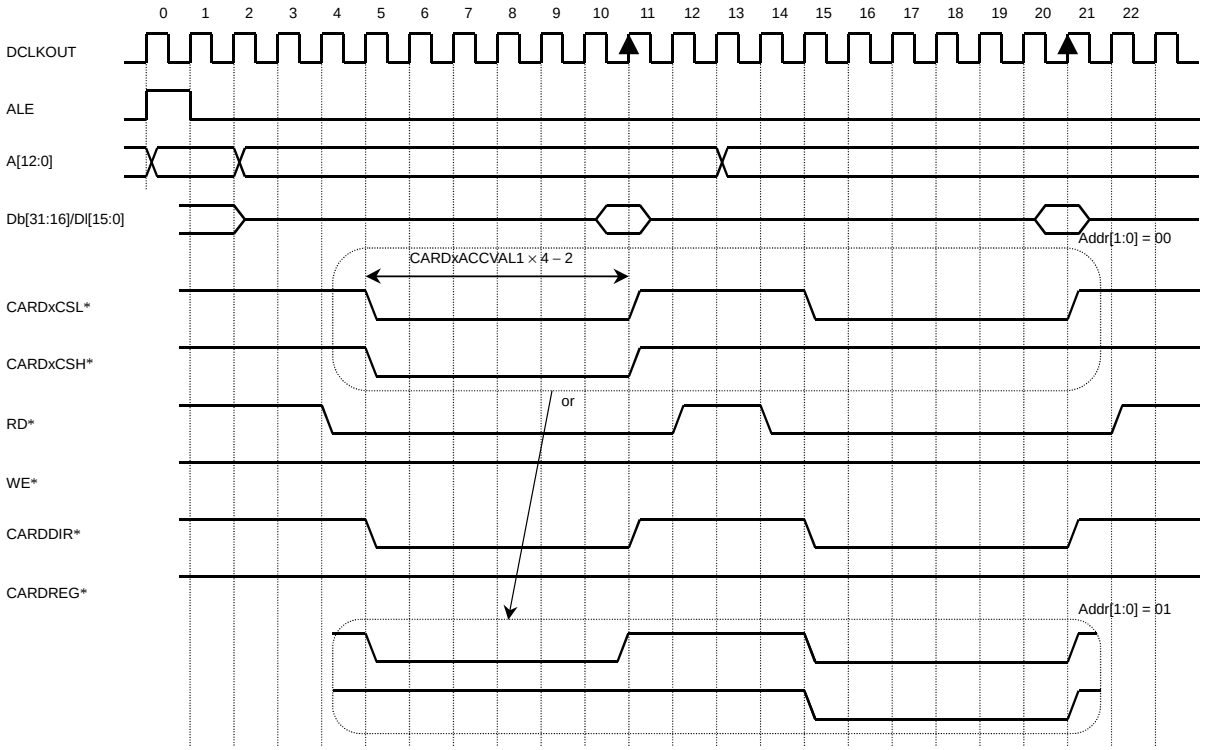


Figure B.30 8-bit CARD Device: Memory Access; Read Operation (3) (Triplebyte)

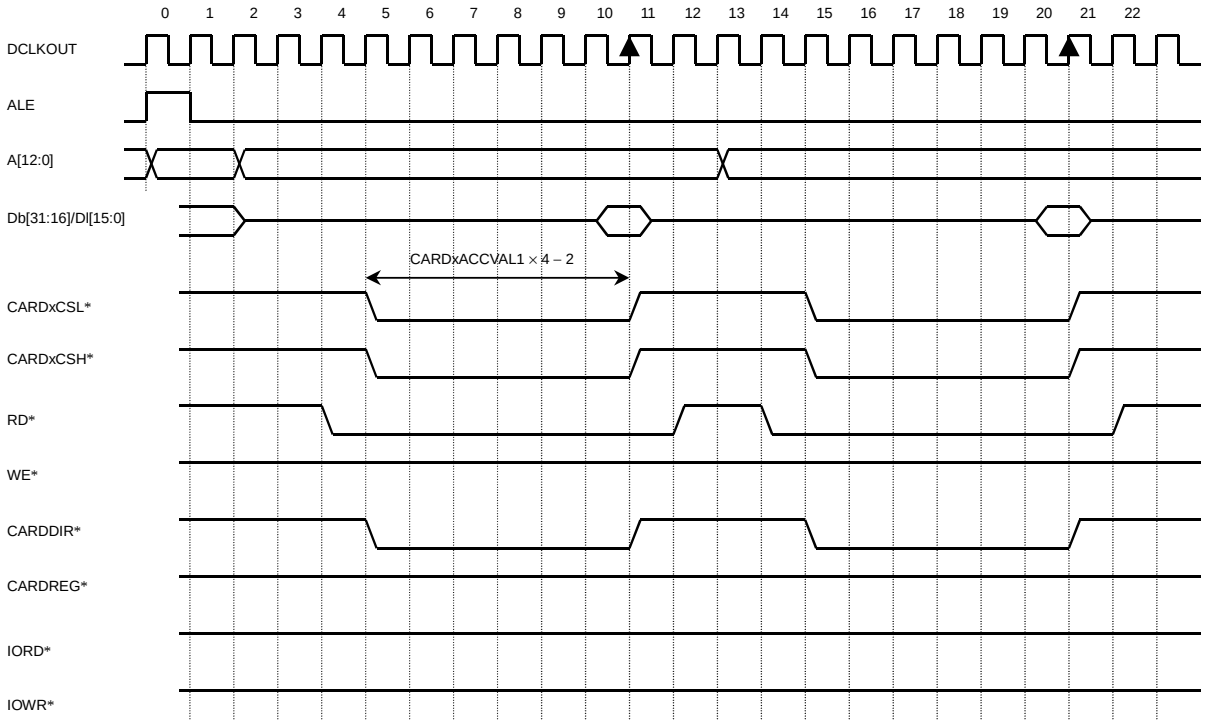


Figure B.31 8-bit CARD Device: Memory Access; Read Operation (4) (Word)

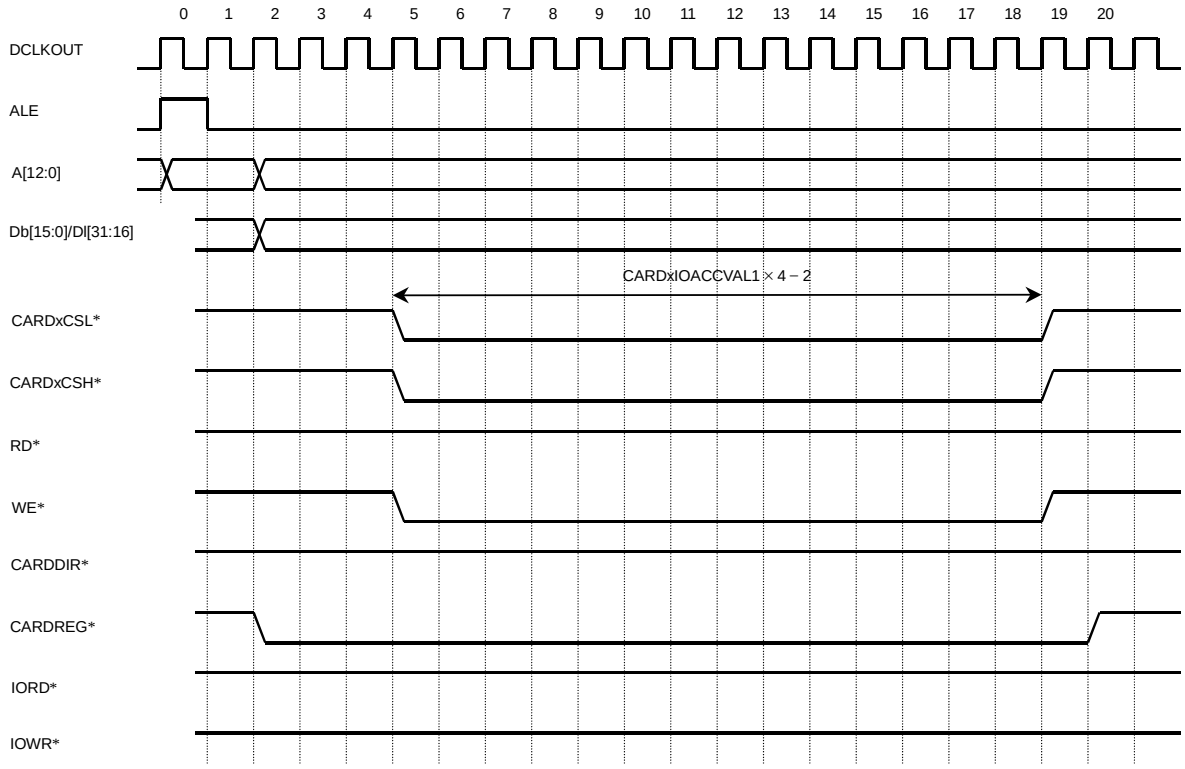


Figure B.32 16-bit CARD Device: Attribute Access; Write Operation (Halfword)

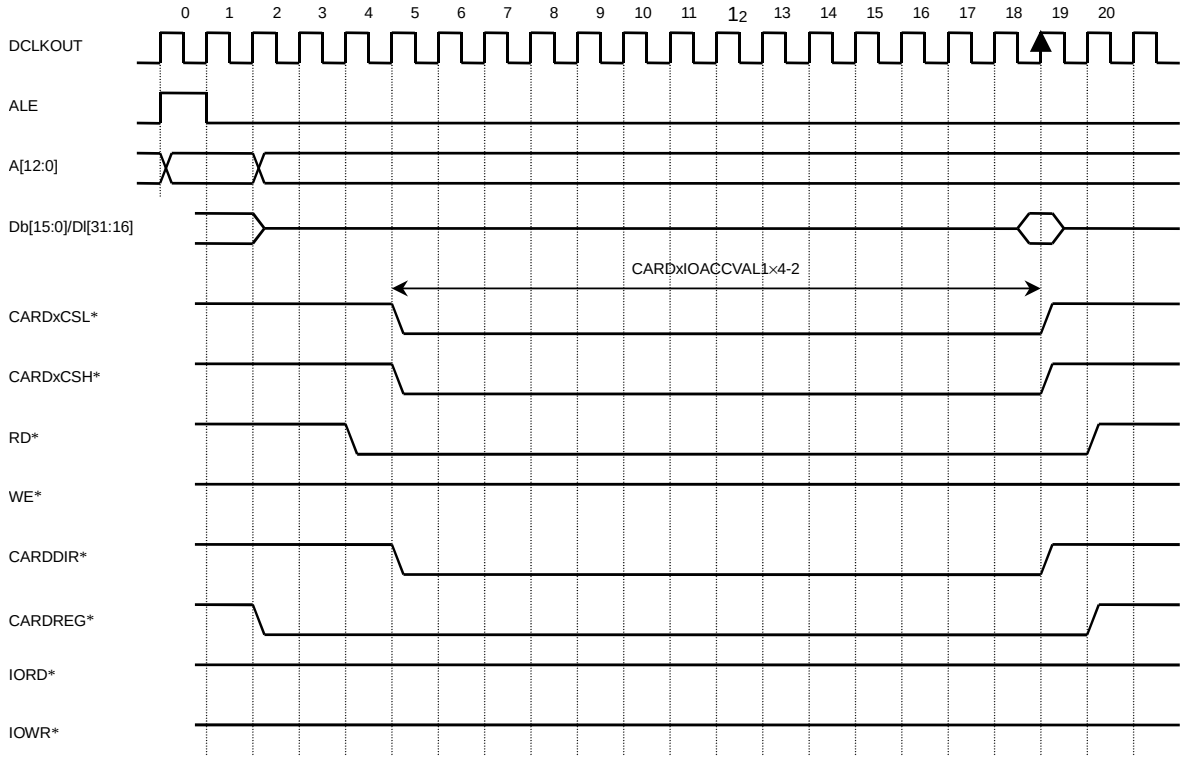


Figure B.33 16-bit CARD Device: Attribute Access; Read Operation (Halfword)

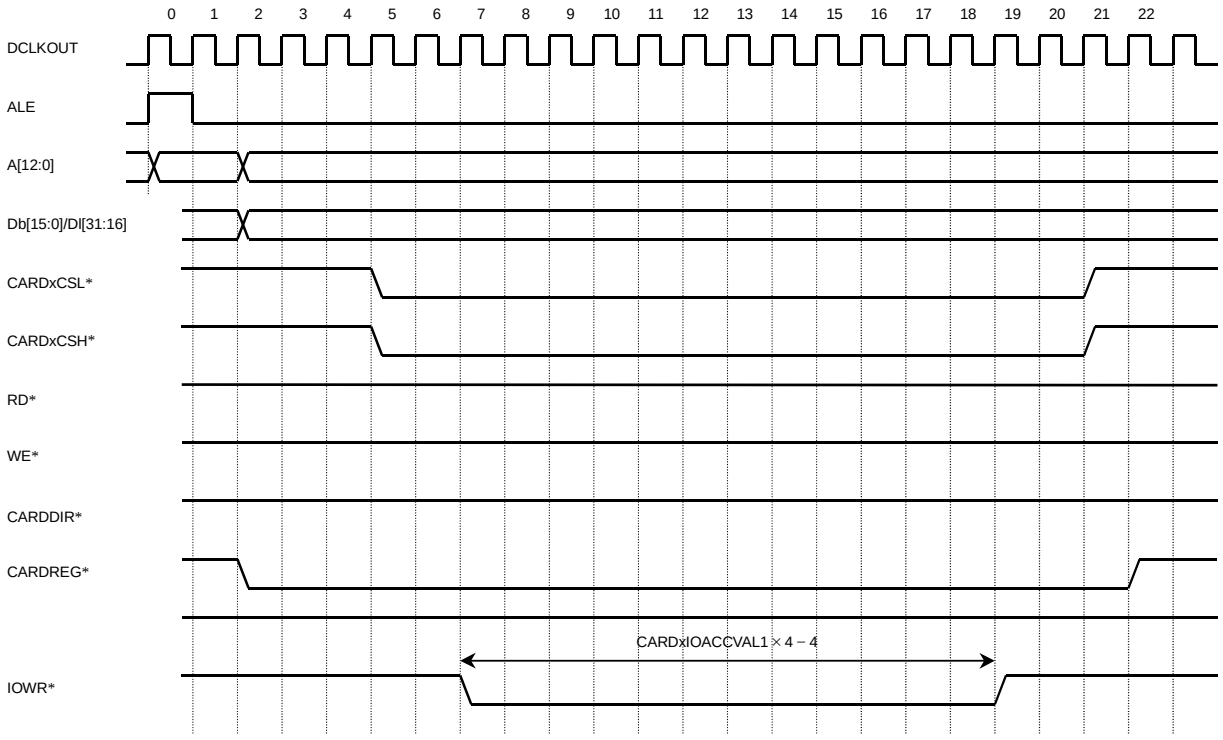


Figure B.34 16-bit CARD Device: I/O Access; Write Operation (Half word)

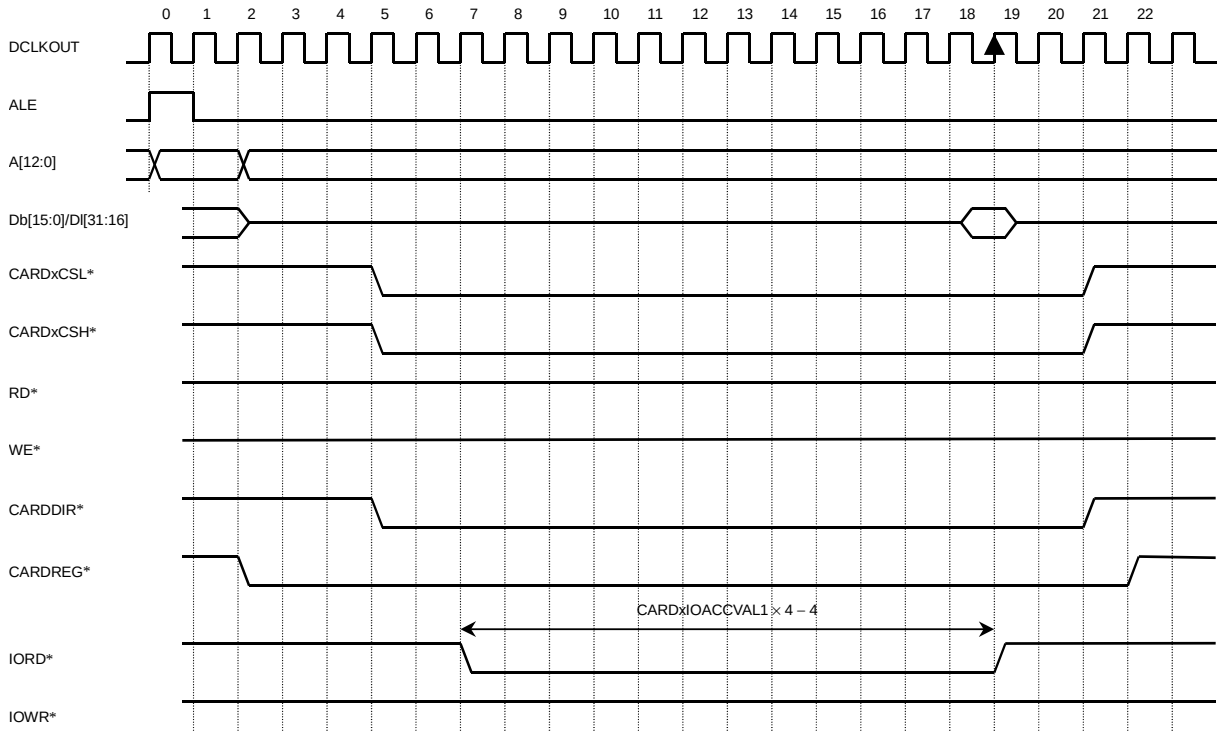
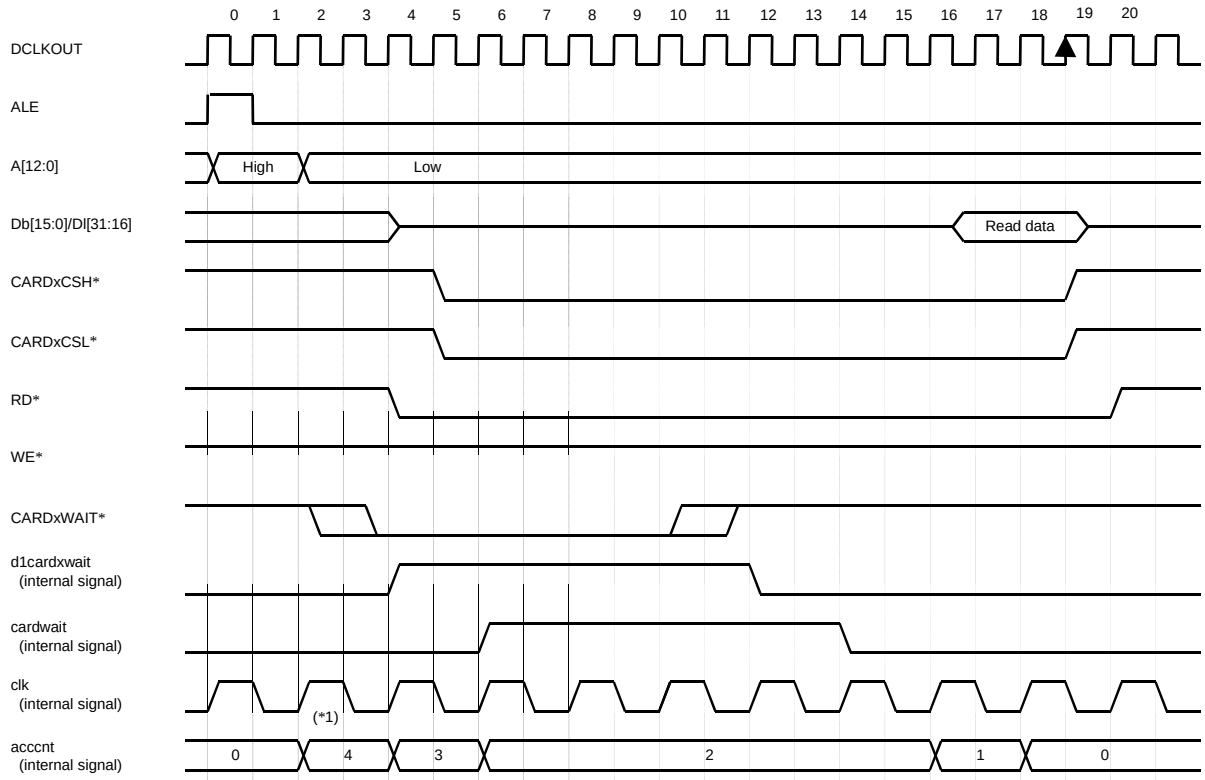


Figure B.35 16-bit CARD Device: I/O Access; Read Operation (Halfword)



Note:(*1) Value of CARDxACCVAL*2 is loaded into ineternal counter (“acccnt”). This chart is a case CARDxACCVAL = 2.

Figure B.36 16-bit CARD Device: Memory Access; Read Operation with CARDxWAIT

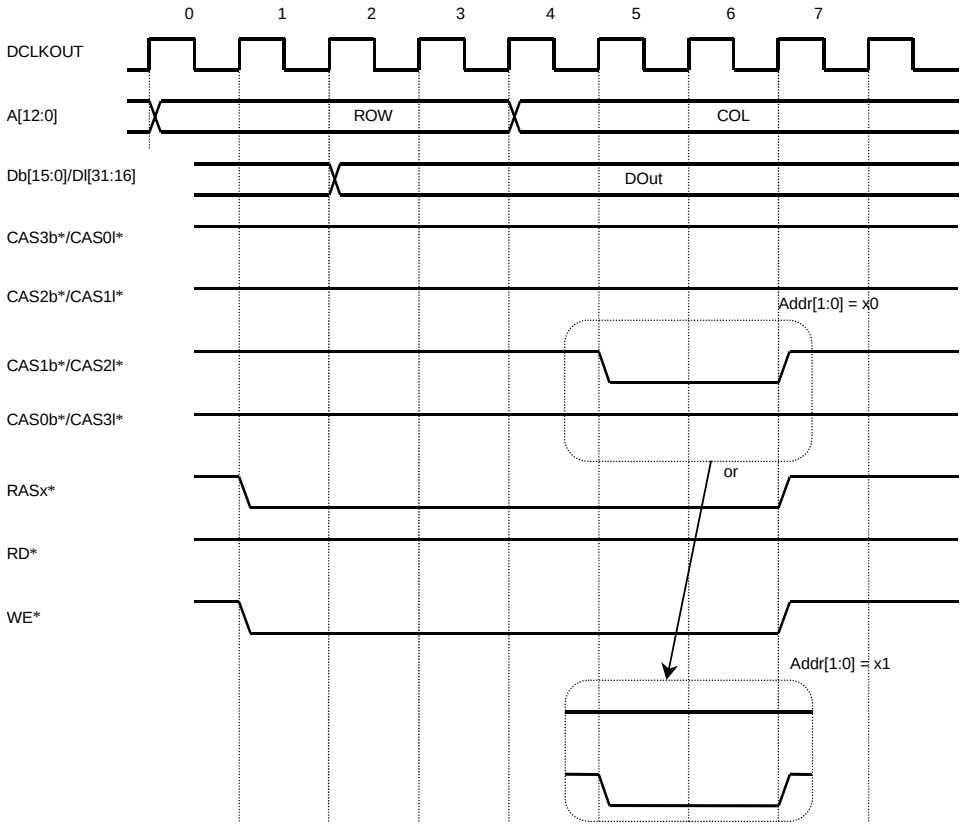


Figure B.37 16-bit DRAM: Write Operation (1) (Byte)

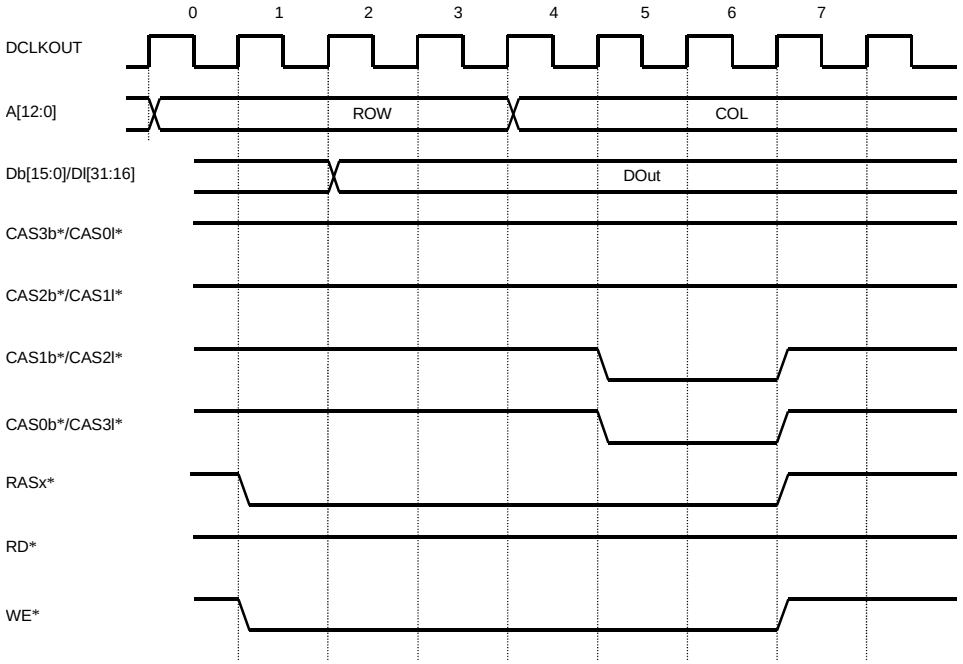


Figure B.38 16-bit DRAM: Write Operation (2) (Halfword)

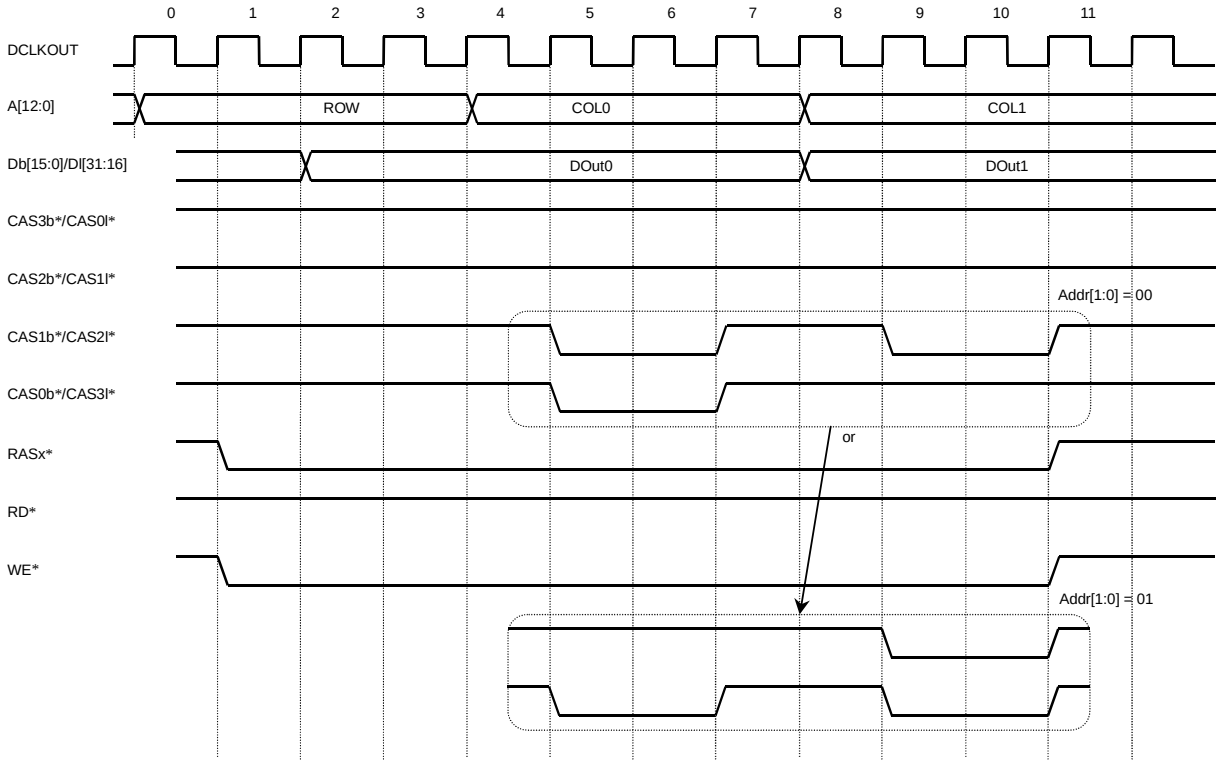


Figure B.39 16-bit DRAM: Write Operation (3) (Triplebyte)

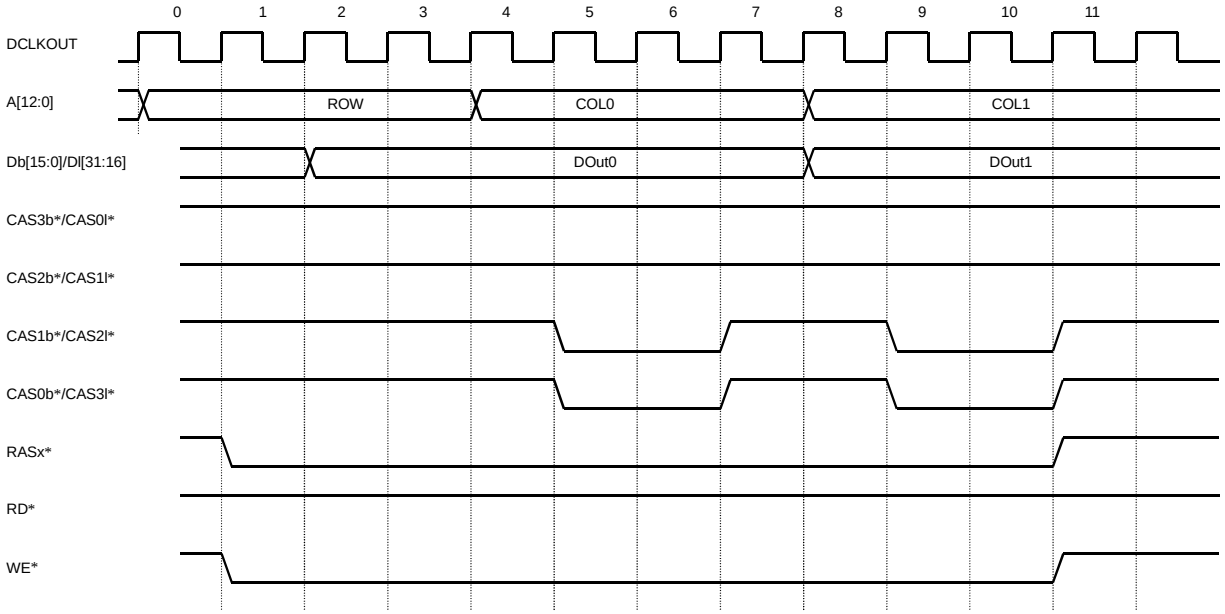


Figure B.40 16-bit DRAM: Write Operation (4) (Word)

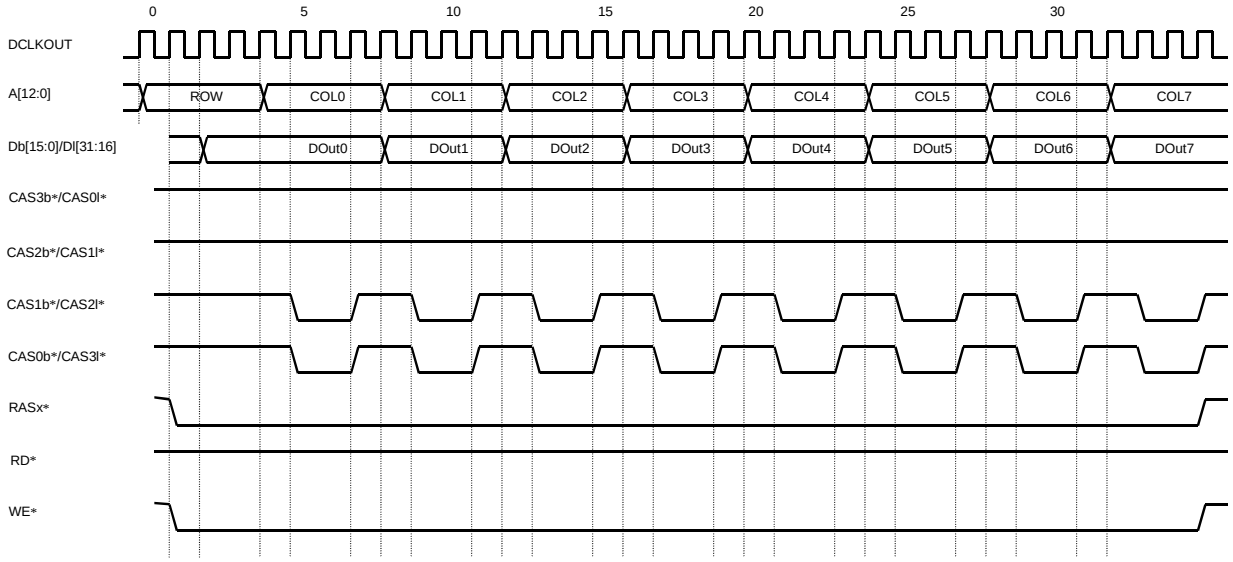


Figure B.41 16-bit DRAM: Write Operation (5) (4 Word Burst, Write In-Page)

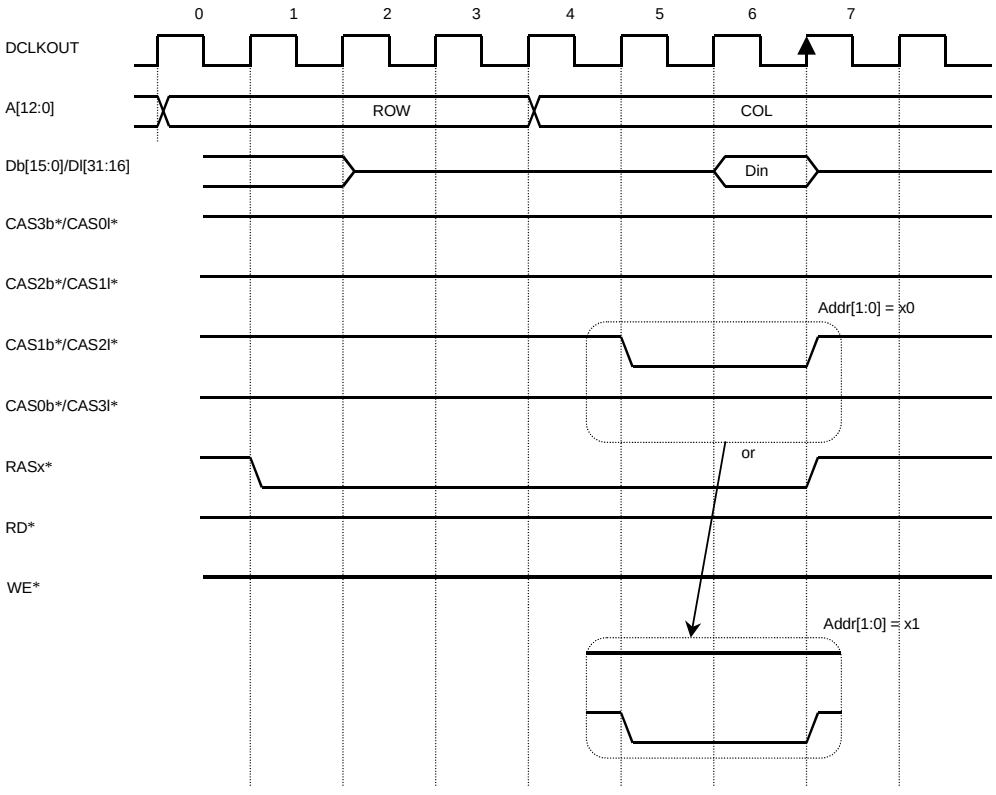


Figure B.42 16-bit DRAM: Read Operation (1) (Byte)

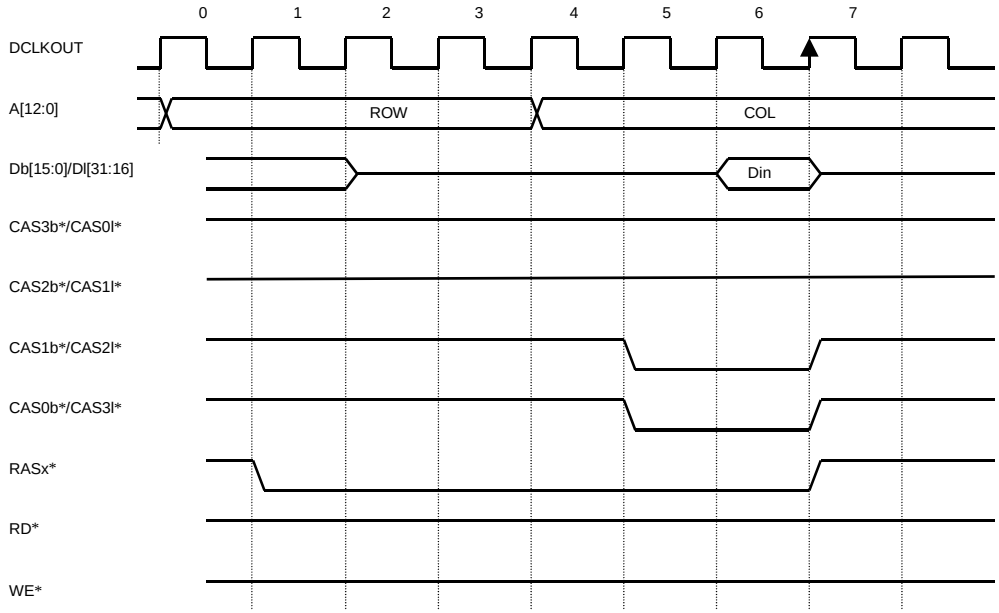


Figure B.43 16-bit DRAM: Read Operation (2) (Halfword)

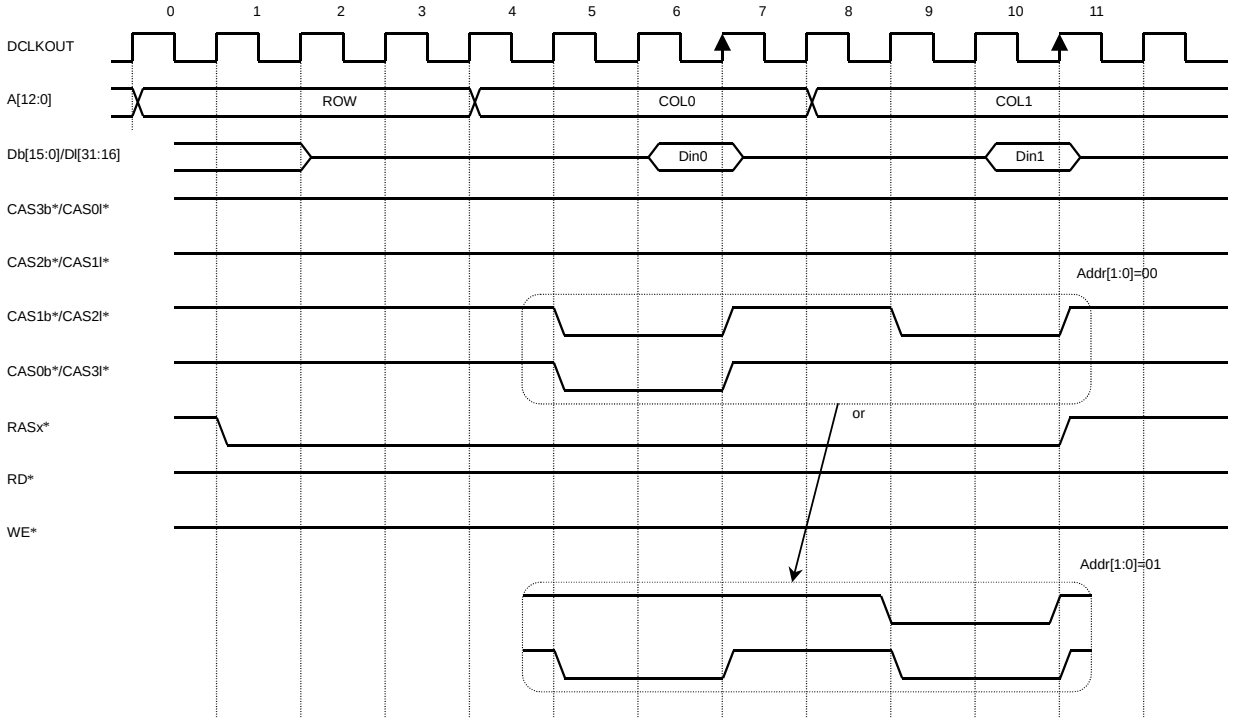


Figure B.44 16-bit DRAM: Read Operation (3) (Triplebyte)

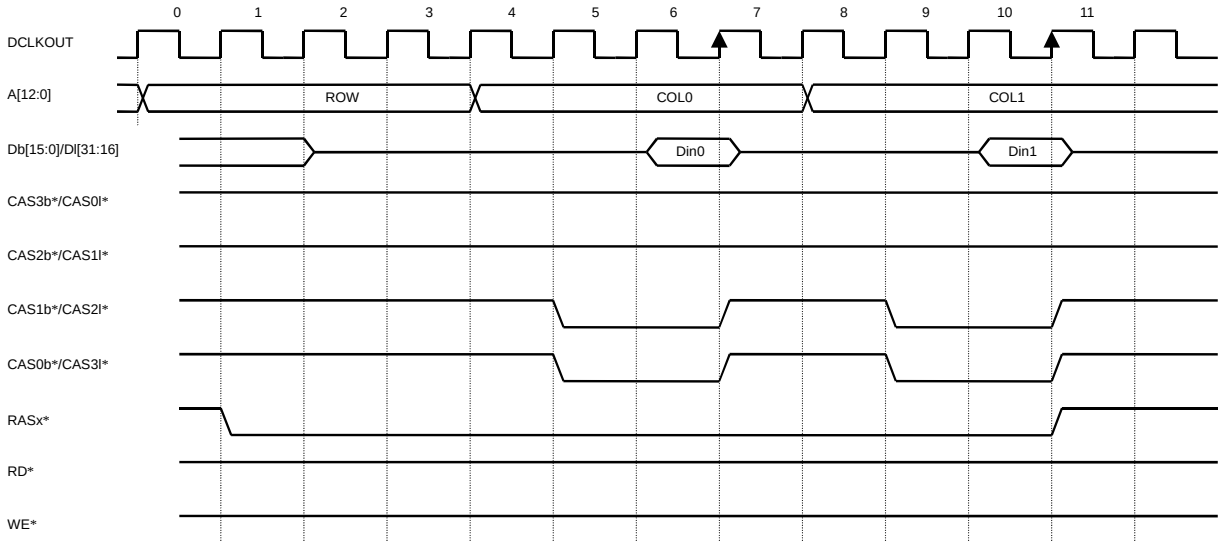


Figure B.45 16-bit DRAM: Read Operation (4) (Word)

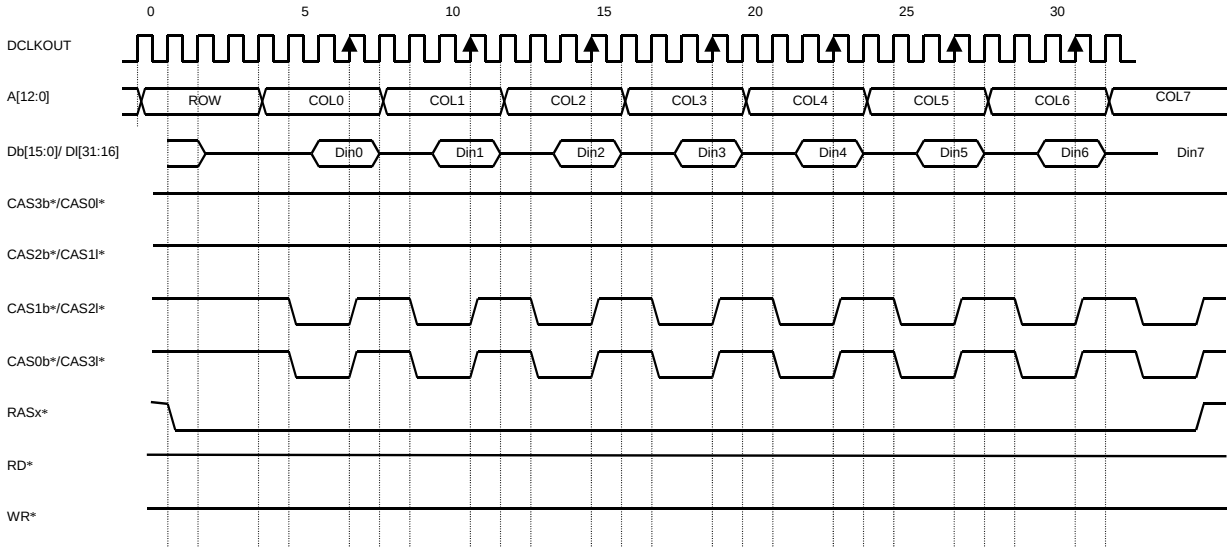


Figure B.46 16-bit DRAM: Read Operation (5) (4 Word Burst)

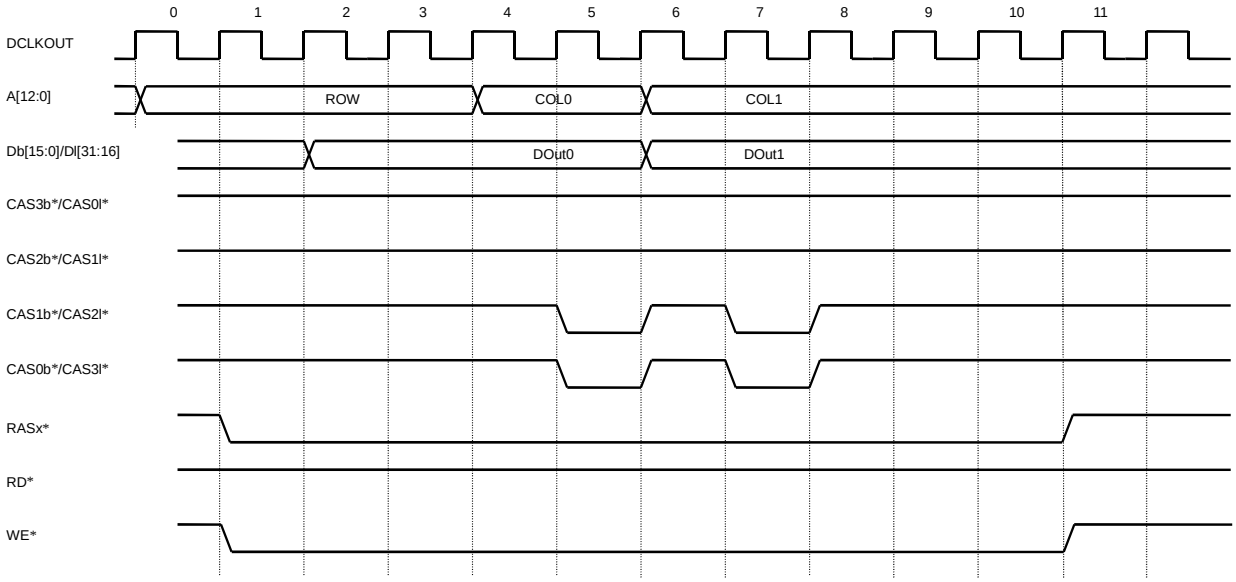


Figure B.47 16-bit DRAM: Write Operation (Word, ENBANKxHDRAM = 1)

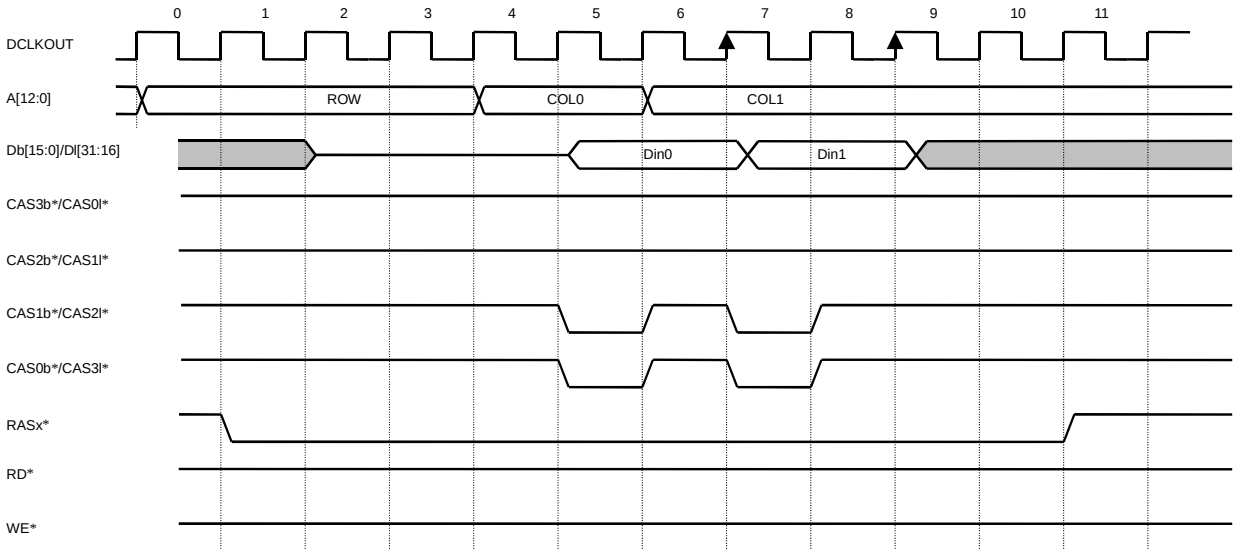


Figure B.48 16-bit DRAM: Read Operation (Word, ENBANKxHDRAM = 1)

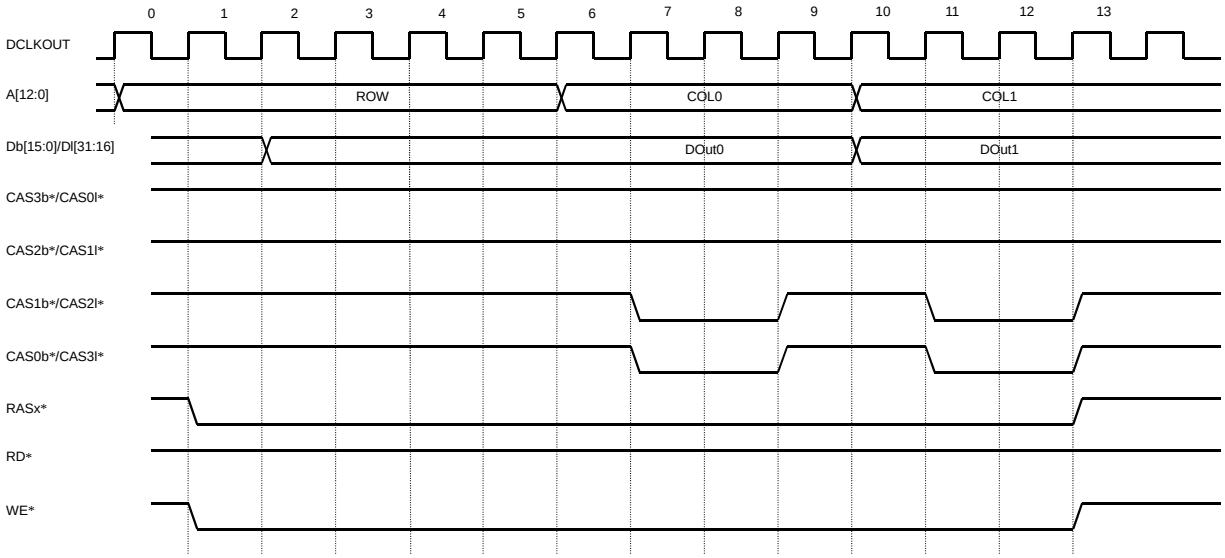


Figure B.49 16-bit DRAM: Write Operation (Word, ENBANKxOPT = 1)

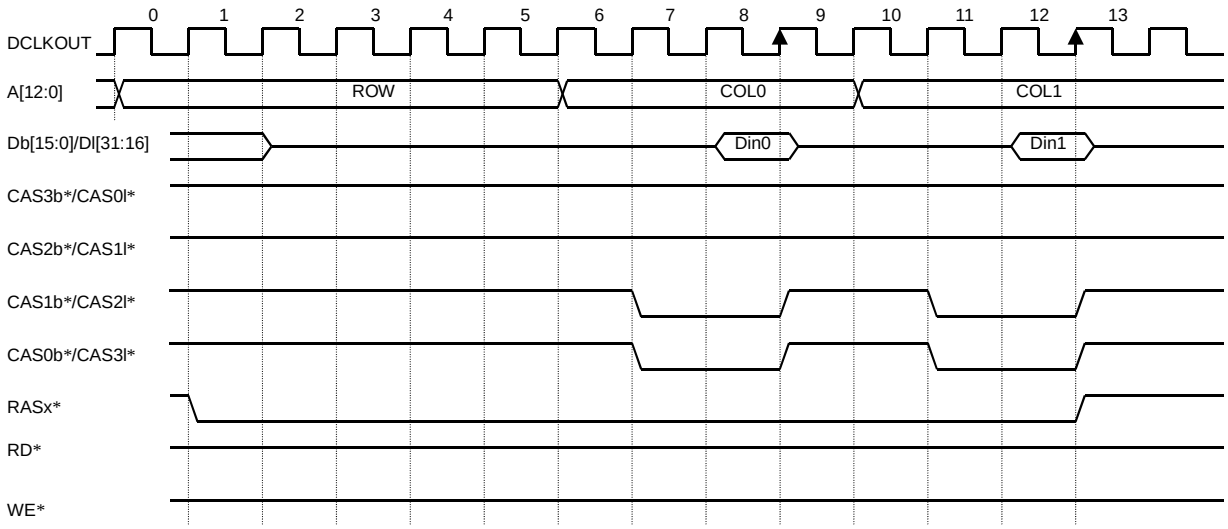


Figure B.50 16-bit DRAM: Read Operation (Word, ENBANKxOPT = 1)

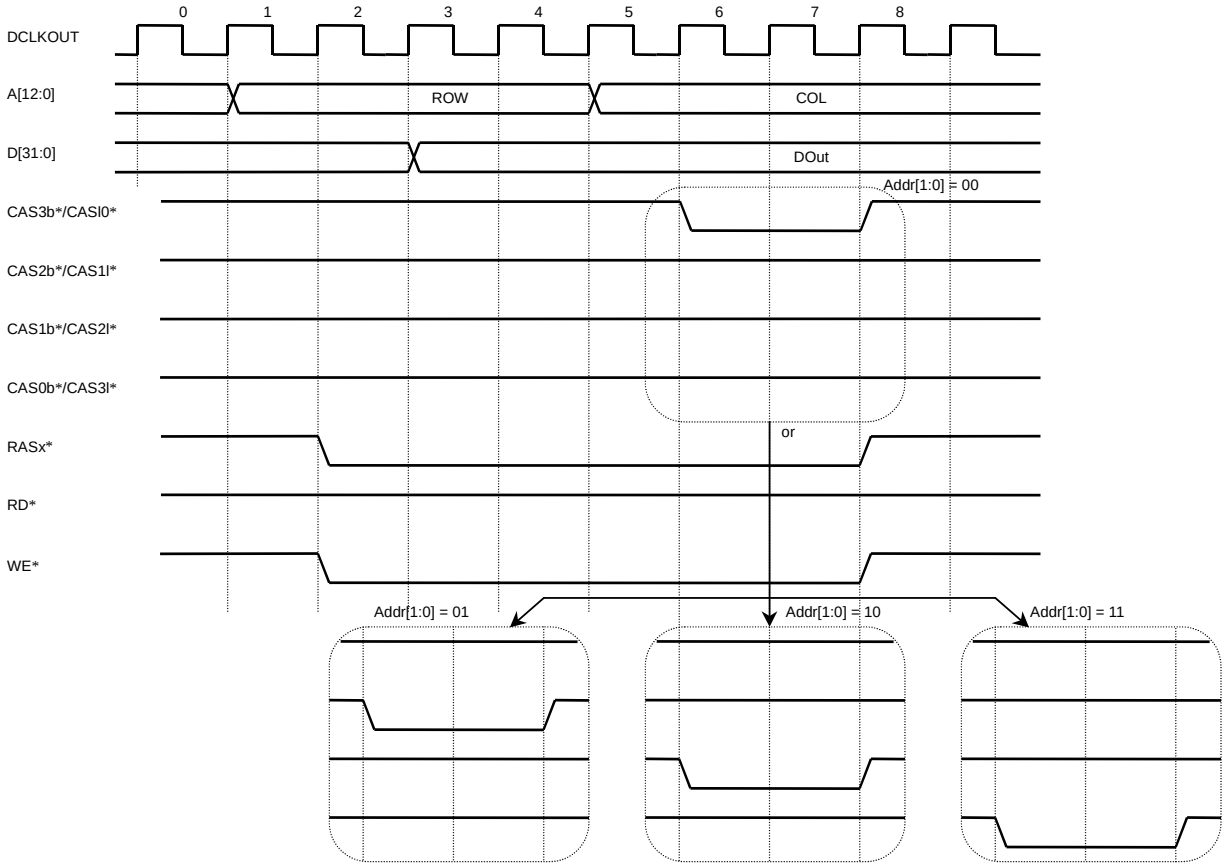


Figure B.51 32-bit DRAM: Write Operation (1) (Byte)

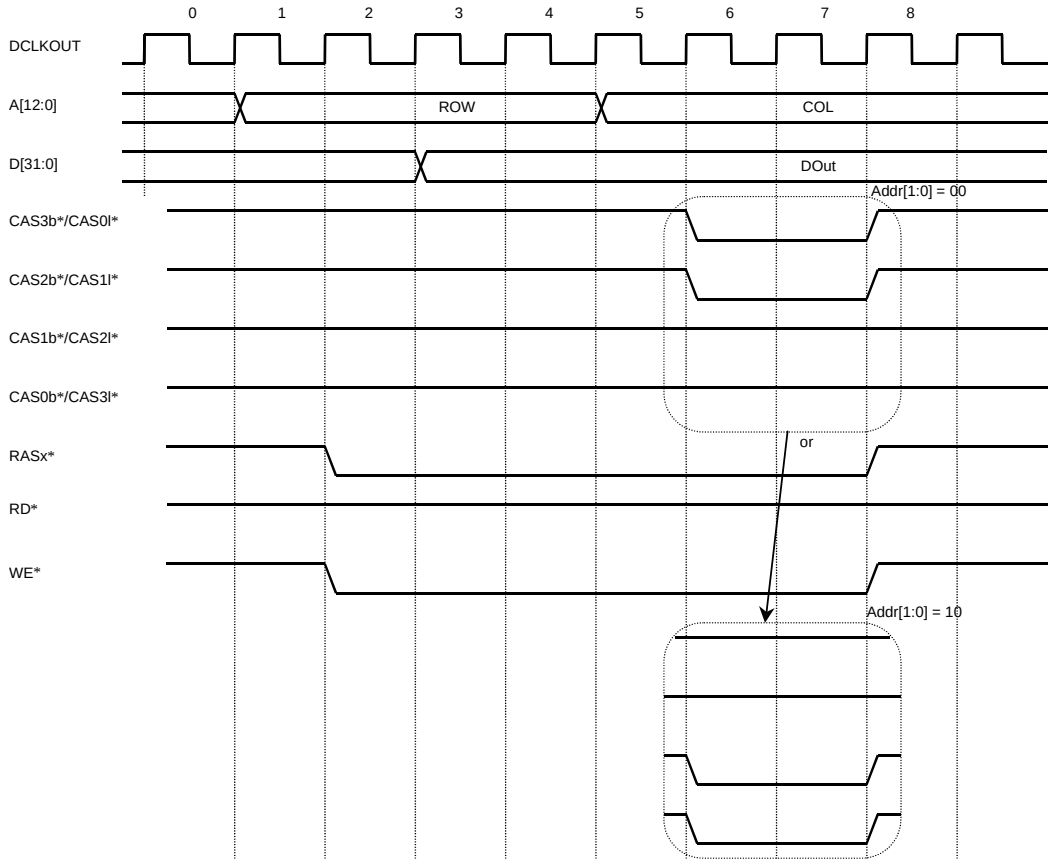


Figure B.52 32-bit DRAM: Write Operation (2) (Halfword)

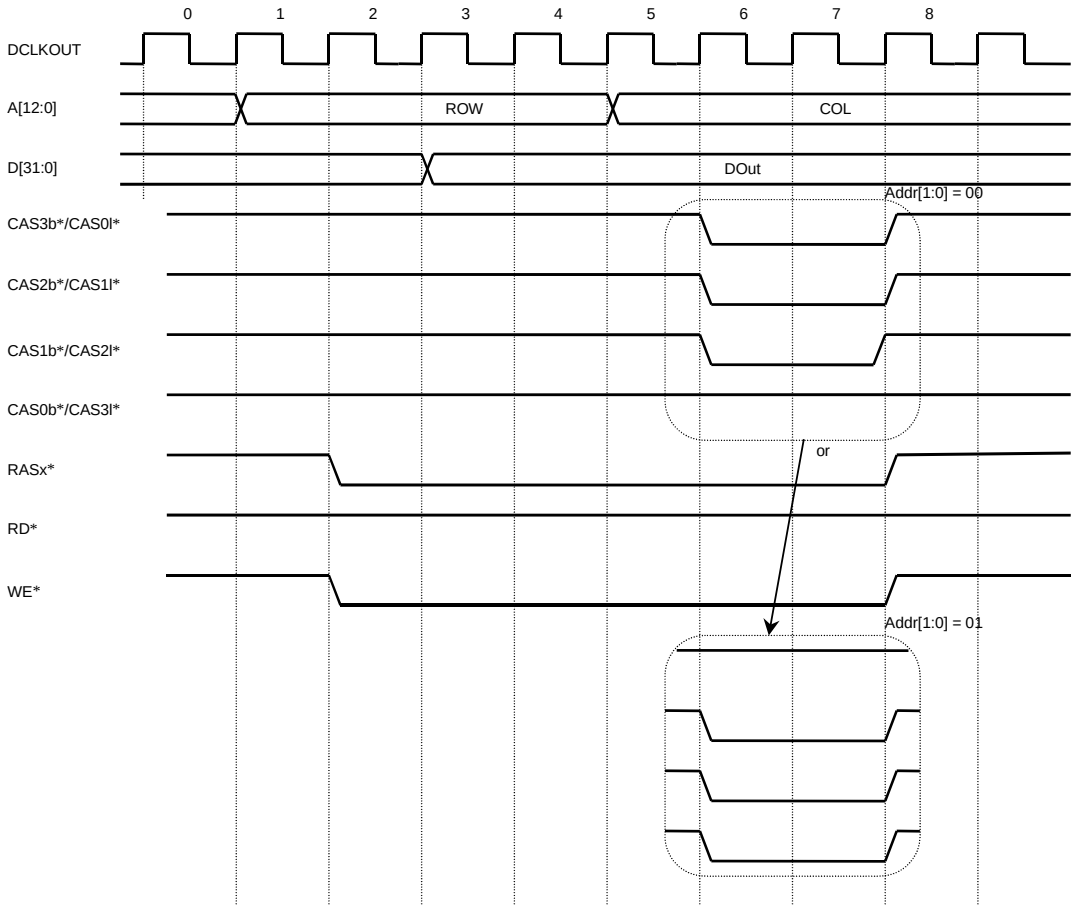


Figure B.53 32-bit DRAM: Write Operation (3) (Triplebyte)

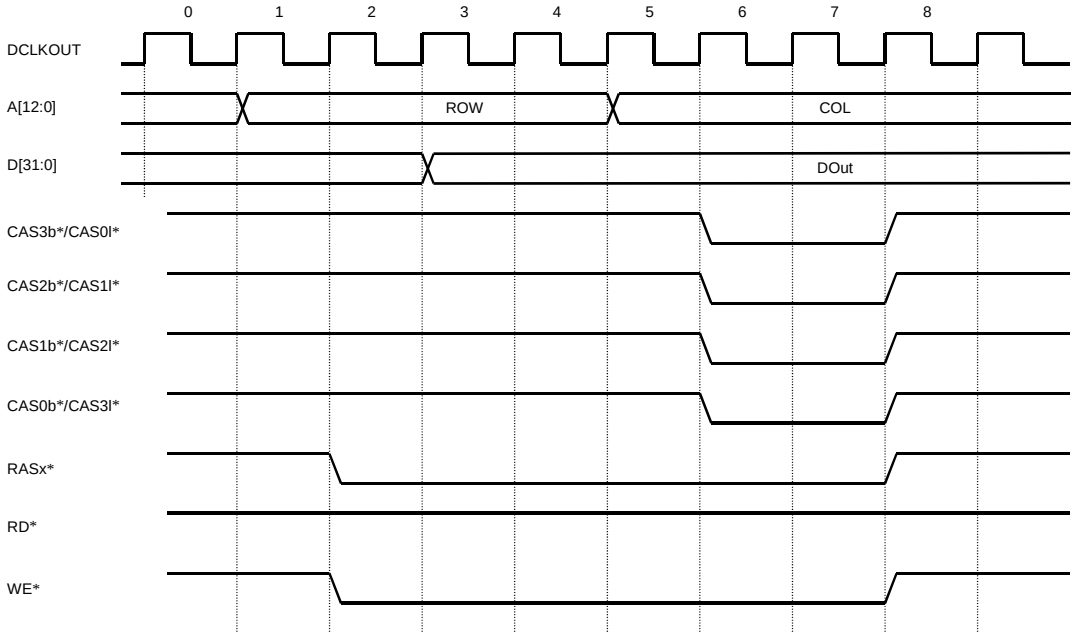


Figure B.54 32-bit DRAM: Write Operation (4) (Word)

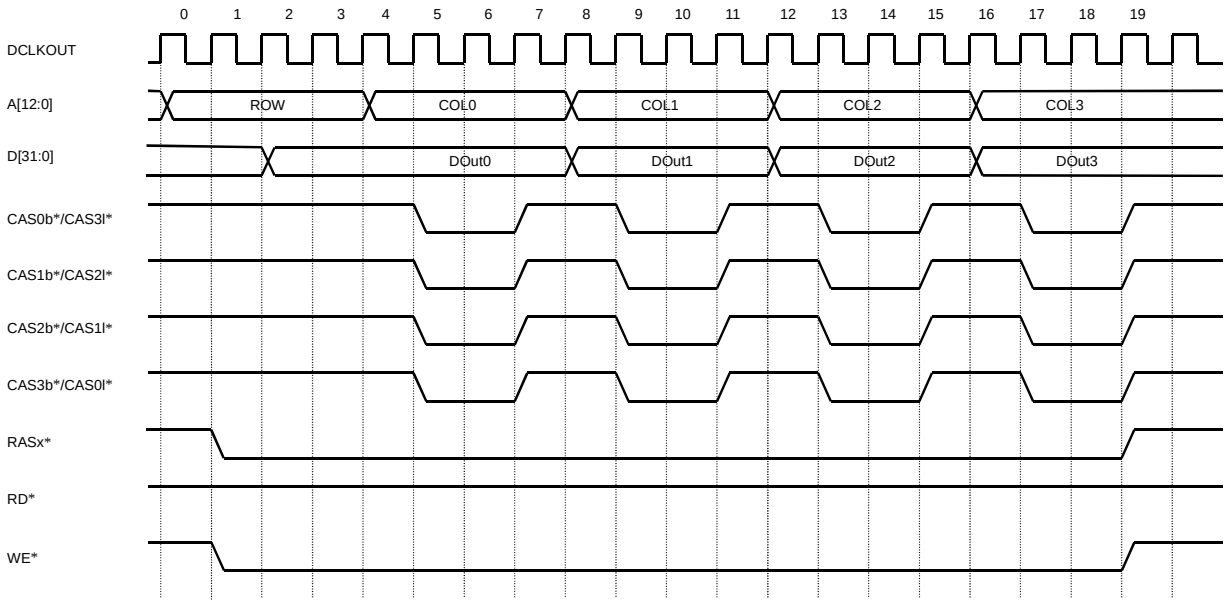


Figure B.55 32-bit DRAM: Write Operation (5) (4 Word Burst, Write In-Page)

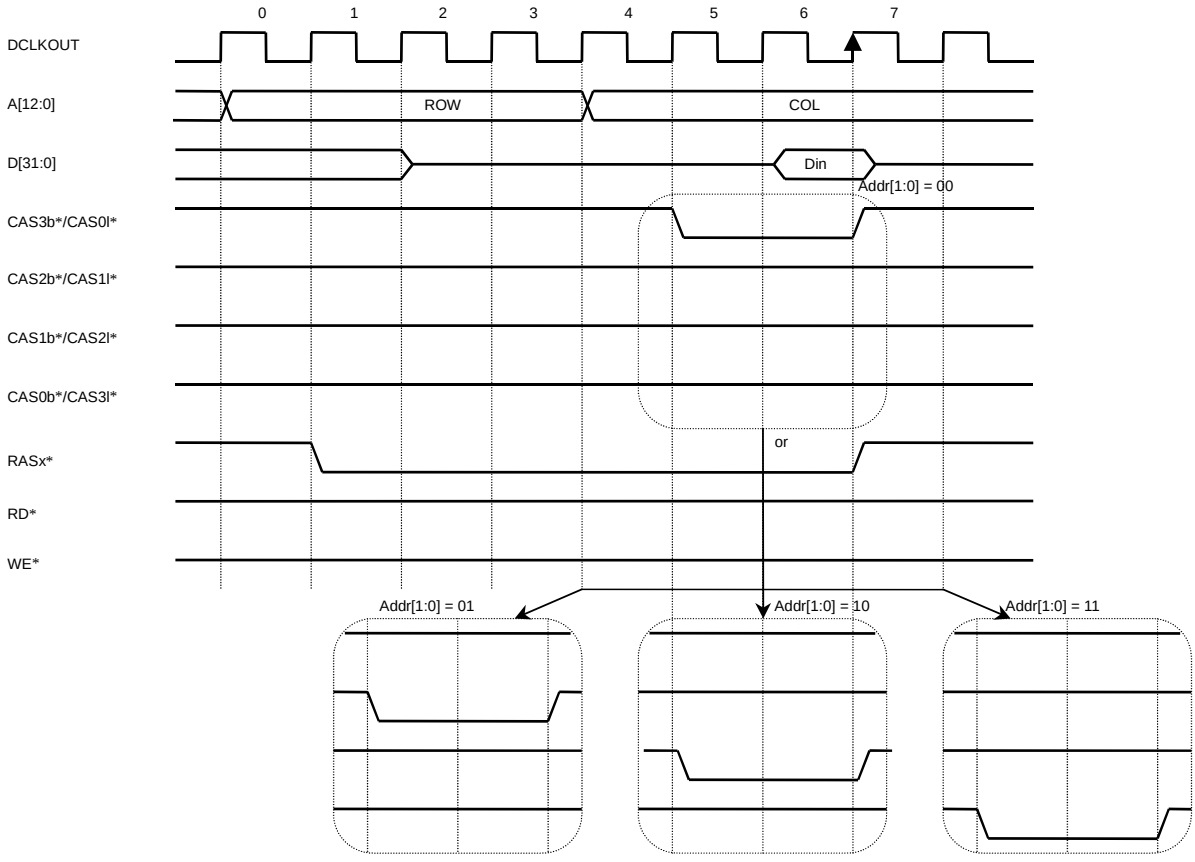


Figure B.56 32-bit DRAM: Read Operation (1) (Byte)

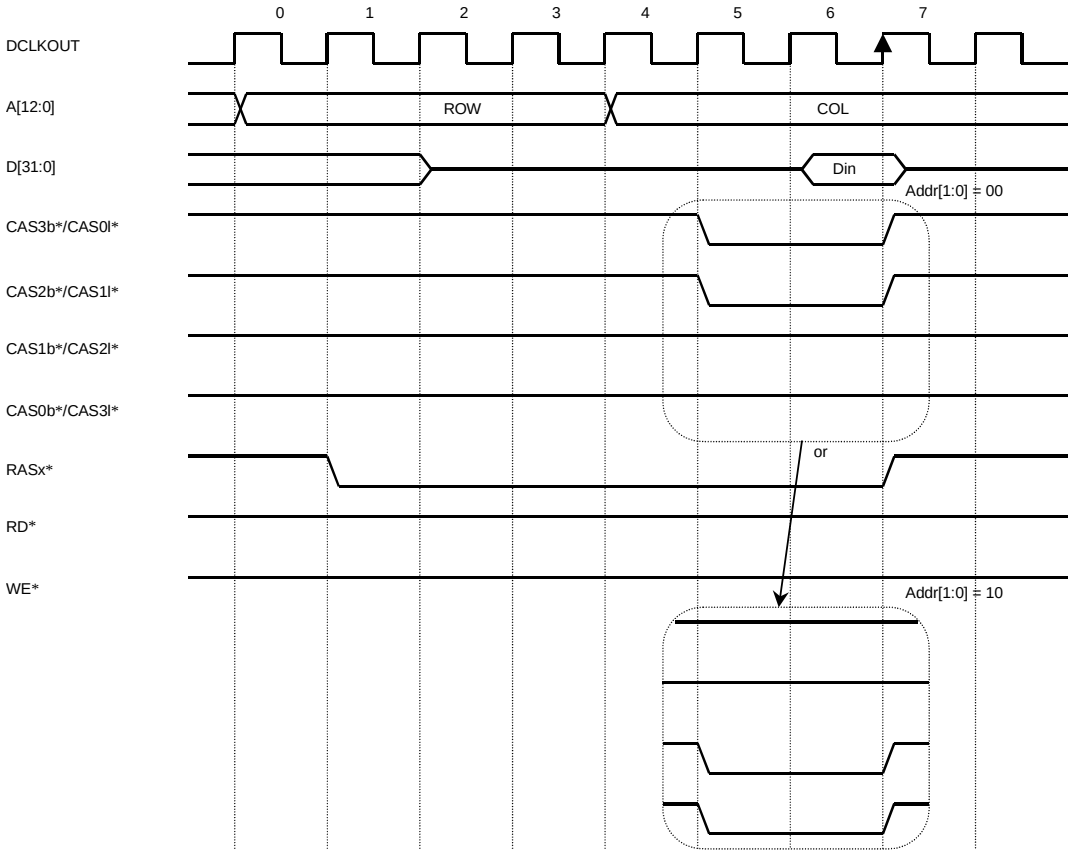


Figure B.57 32-bit DRAM: Read Operation (2) (Halfword)

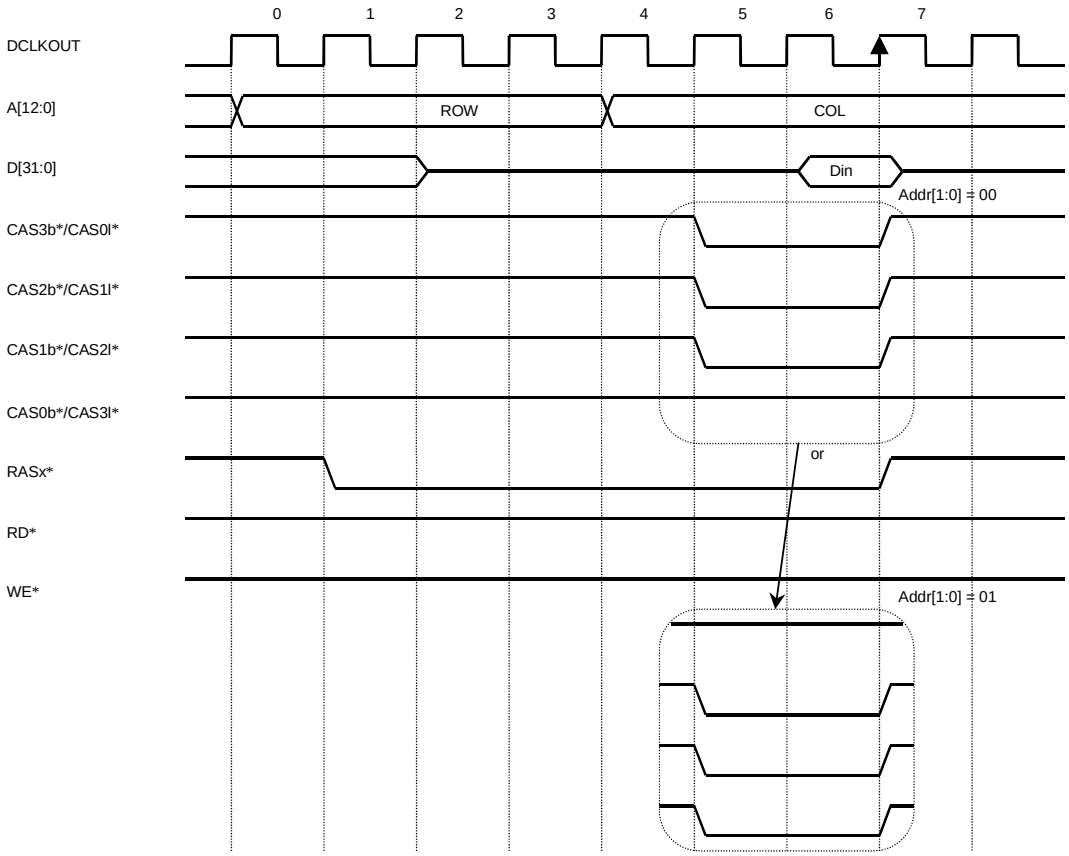


Figure B.58 32-bit DRAM: Read Operation (3) (Triplebyte)

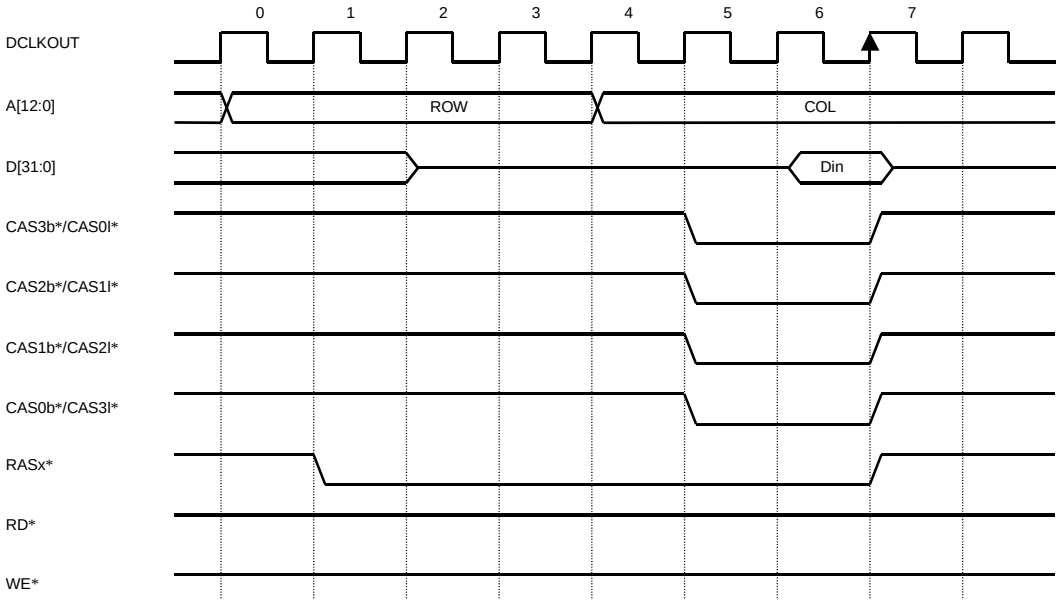


Figure B.59 32-bit DRAM: Read Operation (4) (Word)

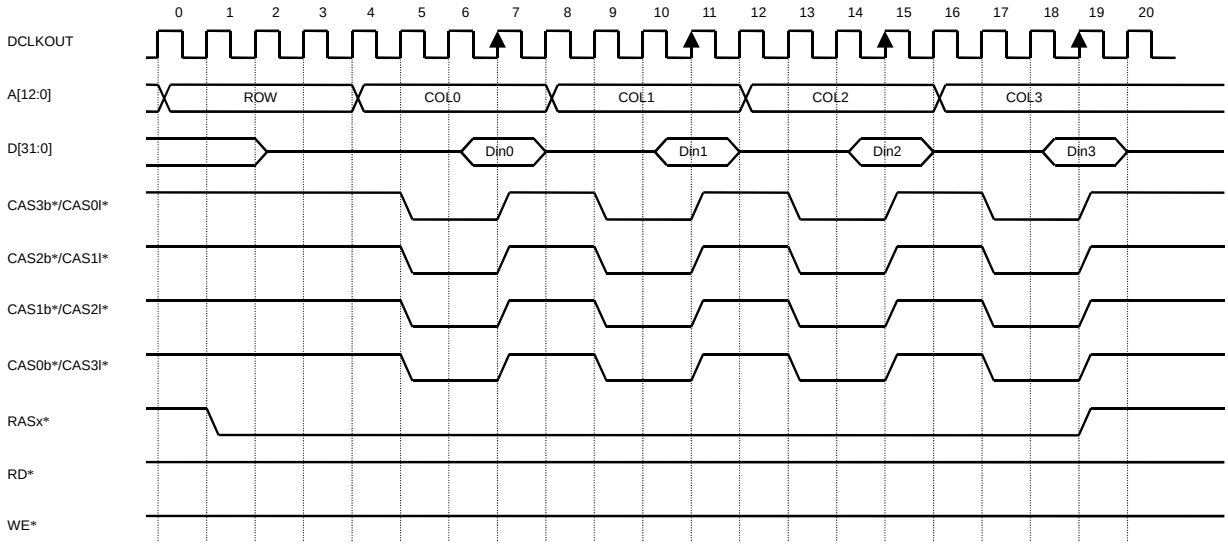


Figure B.60 32-bit DRAM: Read Operation (5) (4 Word Burst)

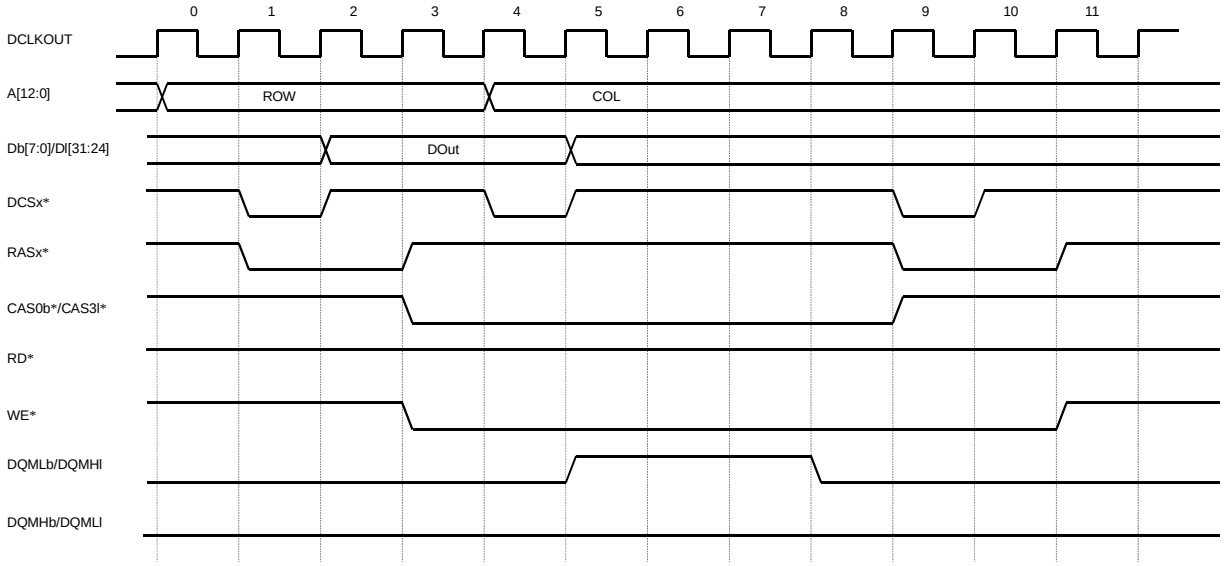


Figure B.61 8-bit SDRAM: Write Operation (1) (Byte)

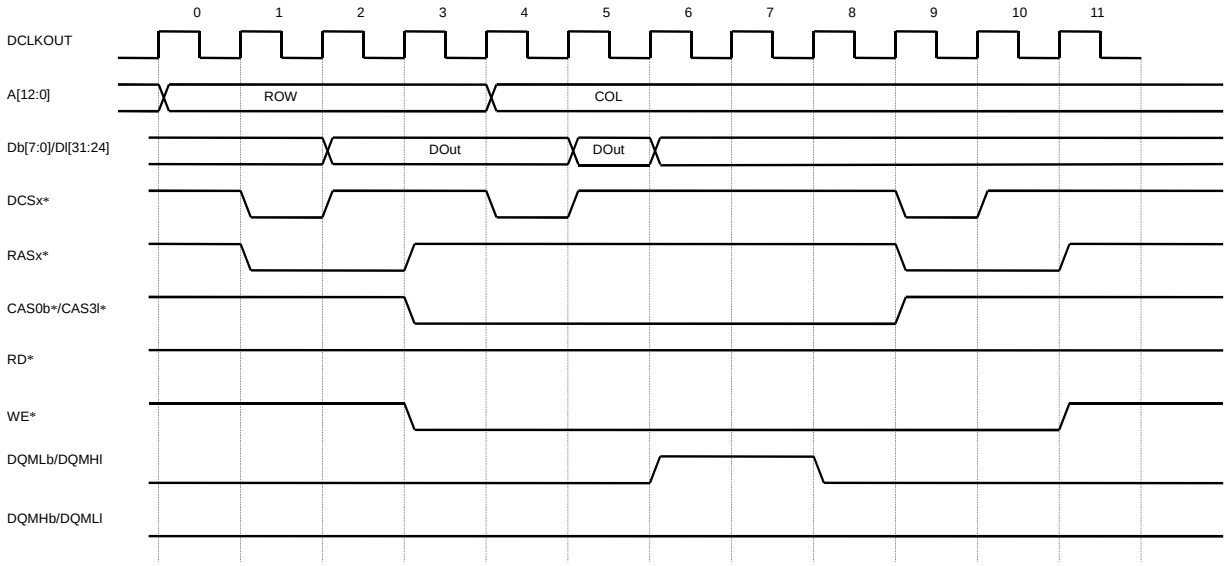


Figure B.62 8-bit SDRAM: Write Operation (2) (Halfword)

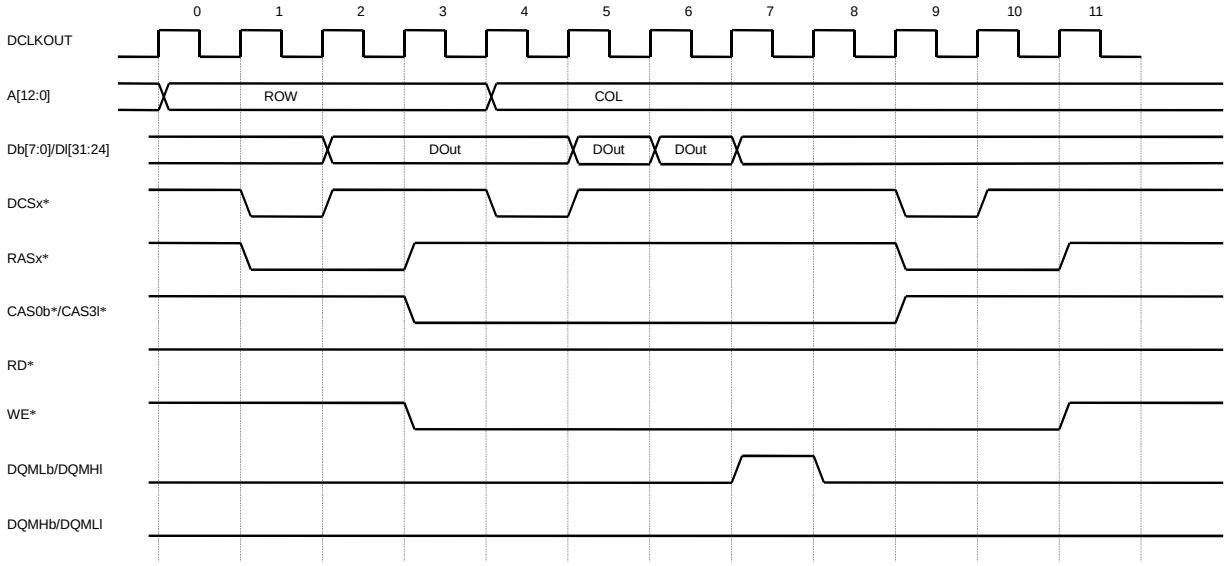


Figure B.63 8-bit SDRAM: Write Operation (3) (Triplebyte)

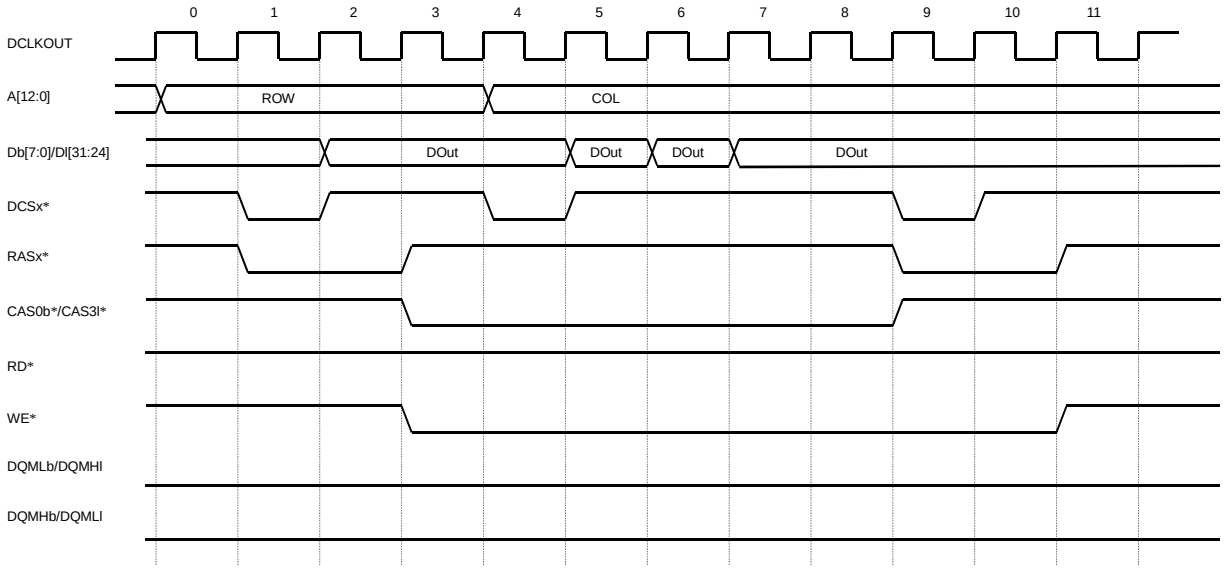


Figure B.64 8-bit SDRAM: Write Operation (4) (Word)

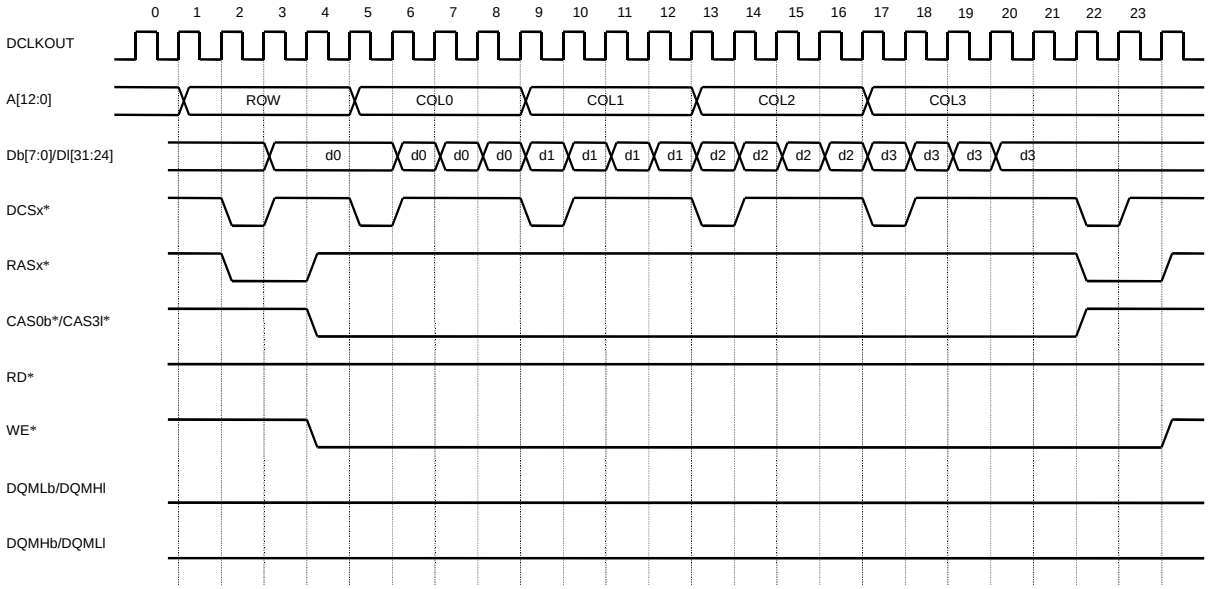
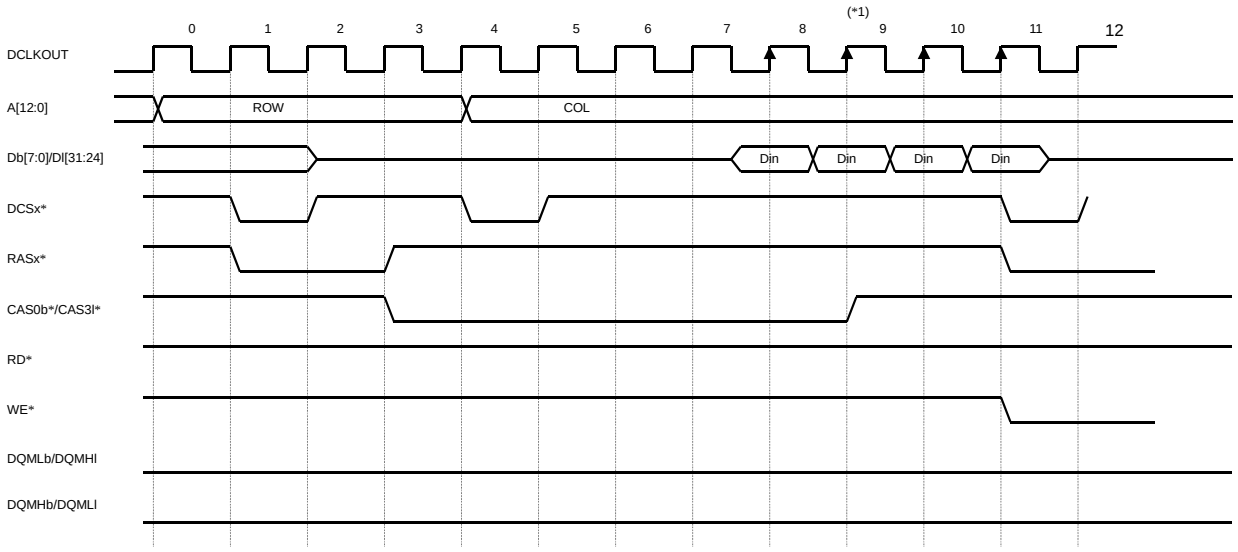


Figure B.65 8-bit SDRAM: Write Operation (5) (4 Word Burst, Write In-Page)



Note (*1): Reading is always be 32bits, without regard to the actual bytes requested by CPU.

Figure B.66 8-bit SDRAM: Read Operation (1) (Byte/Halfword/Tripleword/Word)

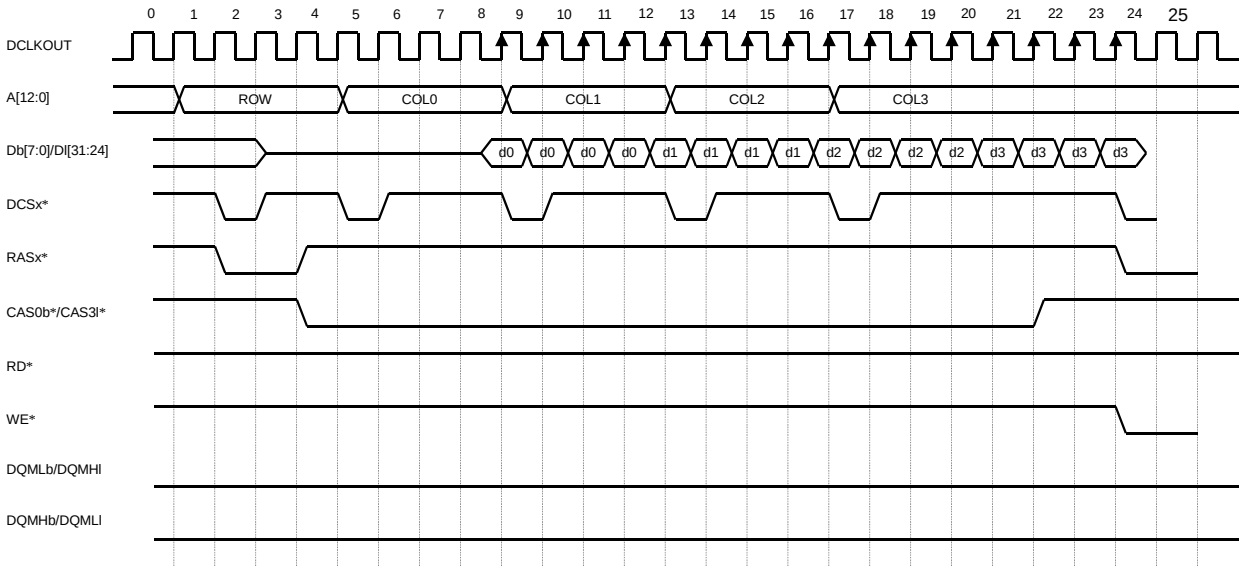


Figure B.67 8-bit SDRAM: Read Operation (2) (4 Word Burst)

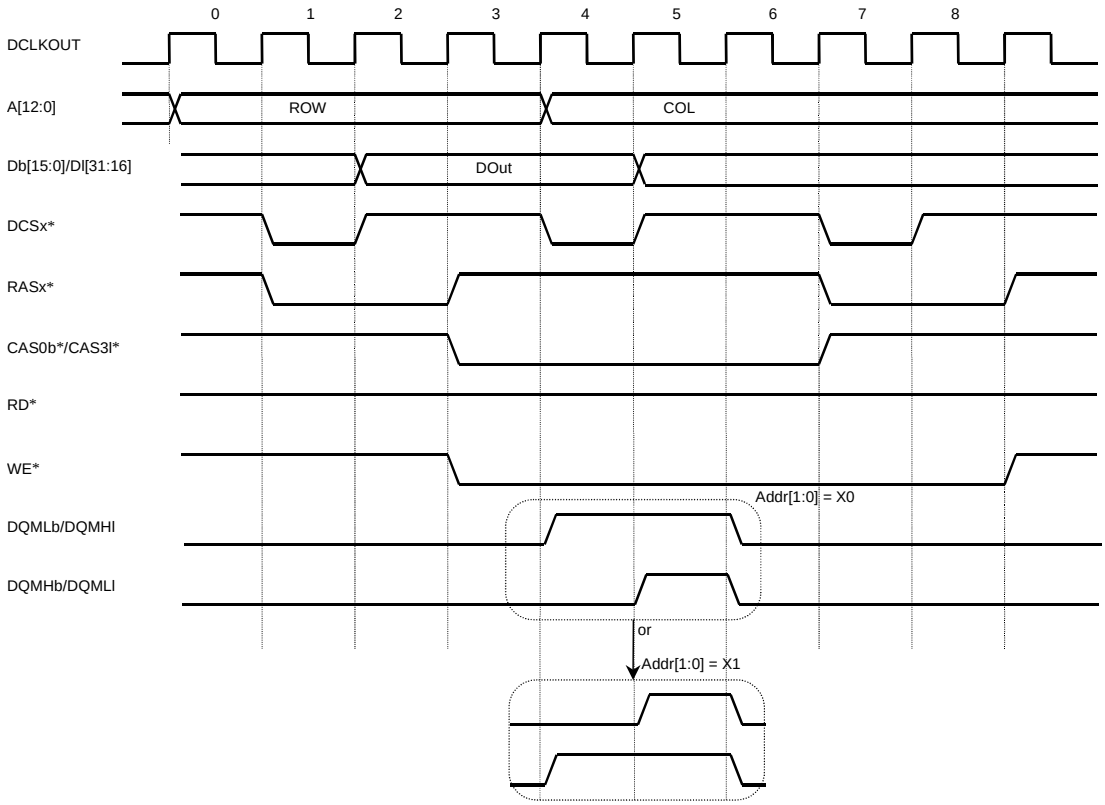


Figure B.68 16-bit SDRAM: Write Operation (1) (Byte)

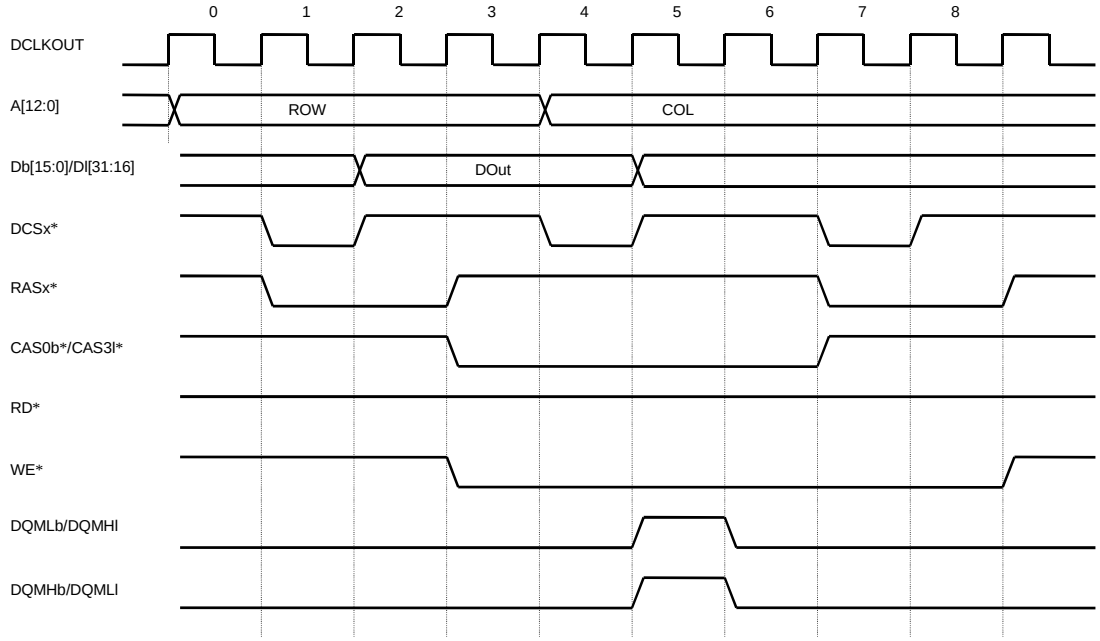


Figure B.69 16-bit SDRAM: Write Operation (2) (Halfword)

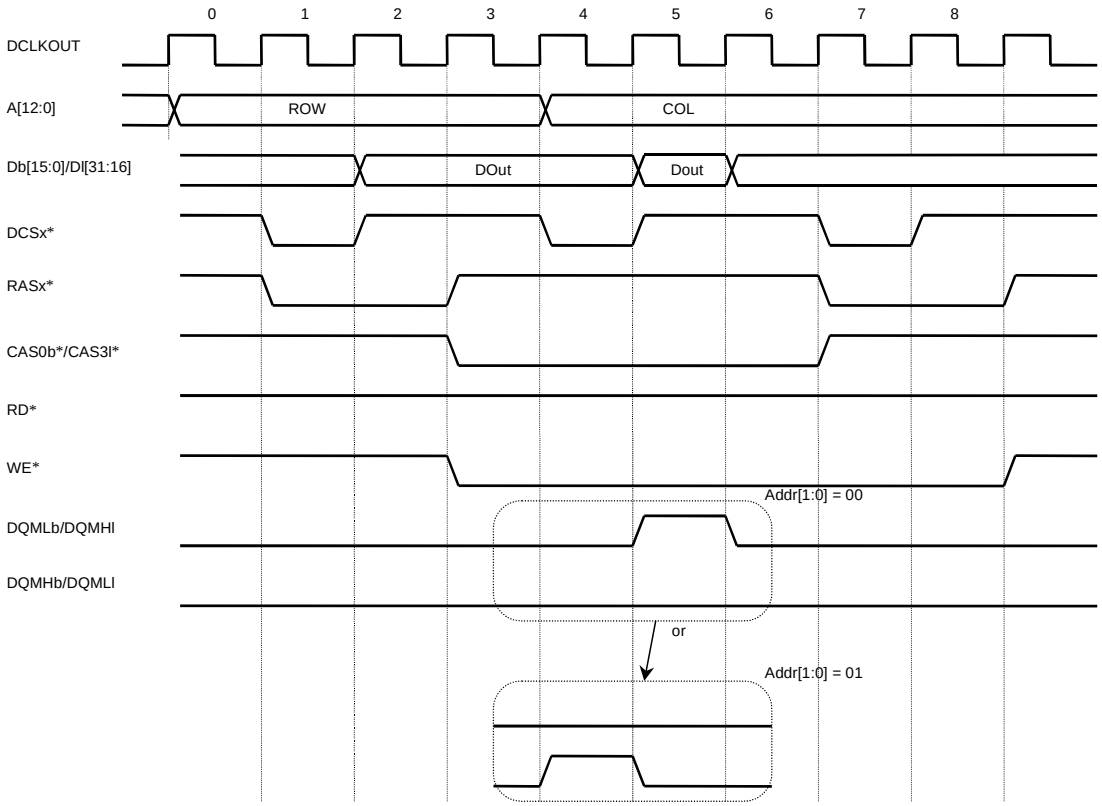


Figure B.70 16-bit SDRAM: Write Operation (3) (Triplebyte)

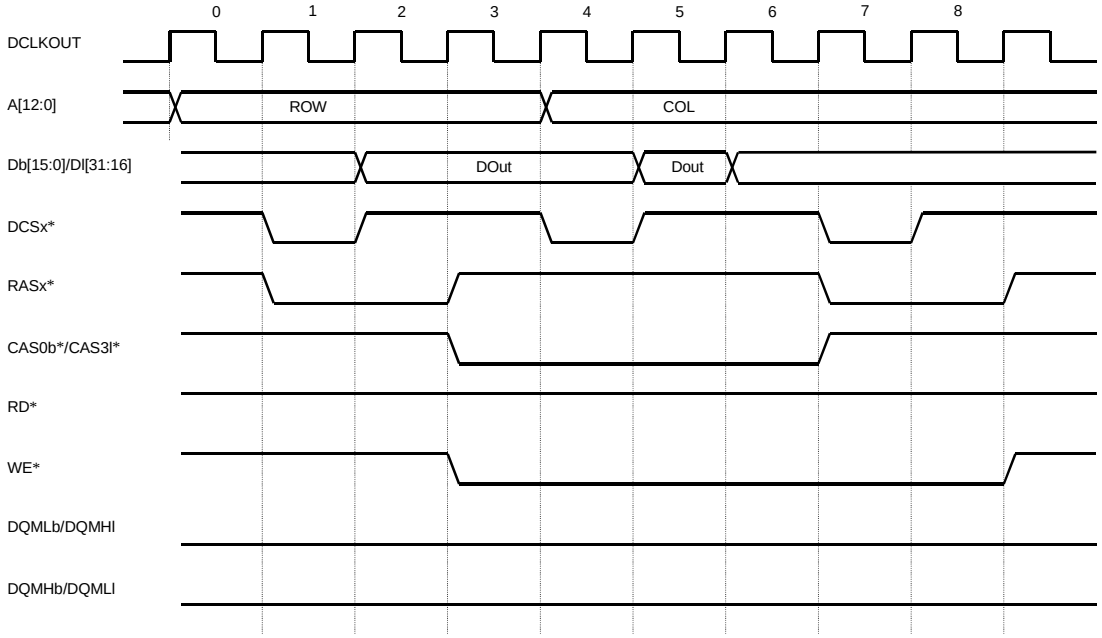


Figure B.71 16-bit SDRAM: Write Operation (4) (Word)

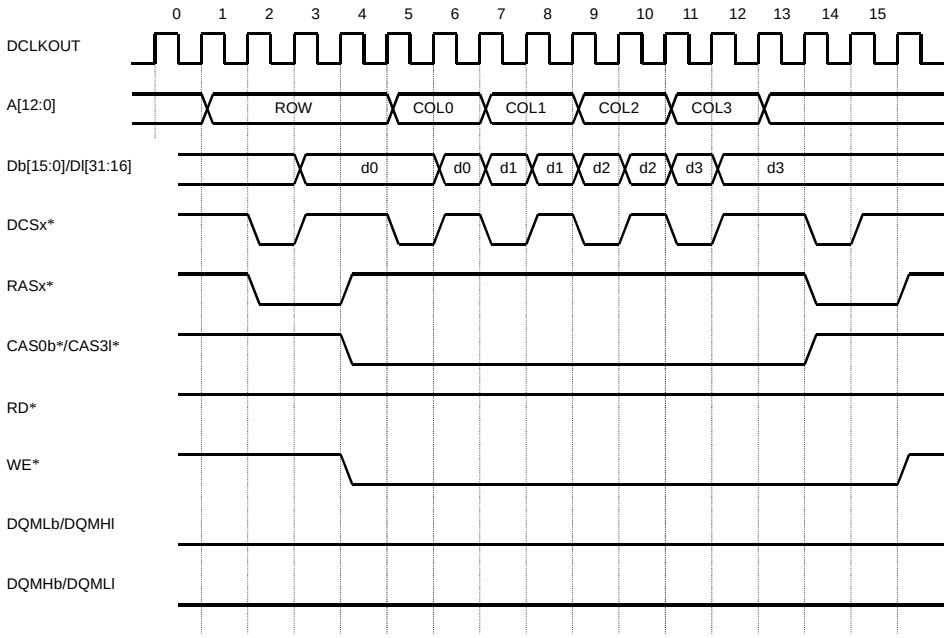
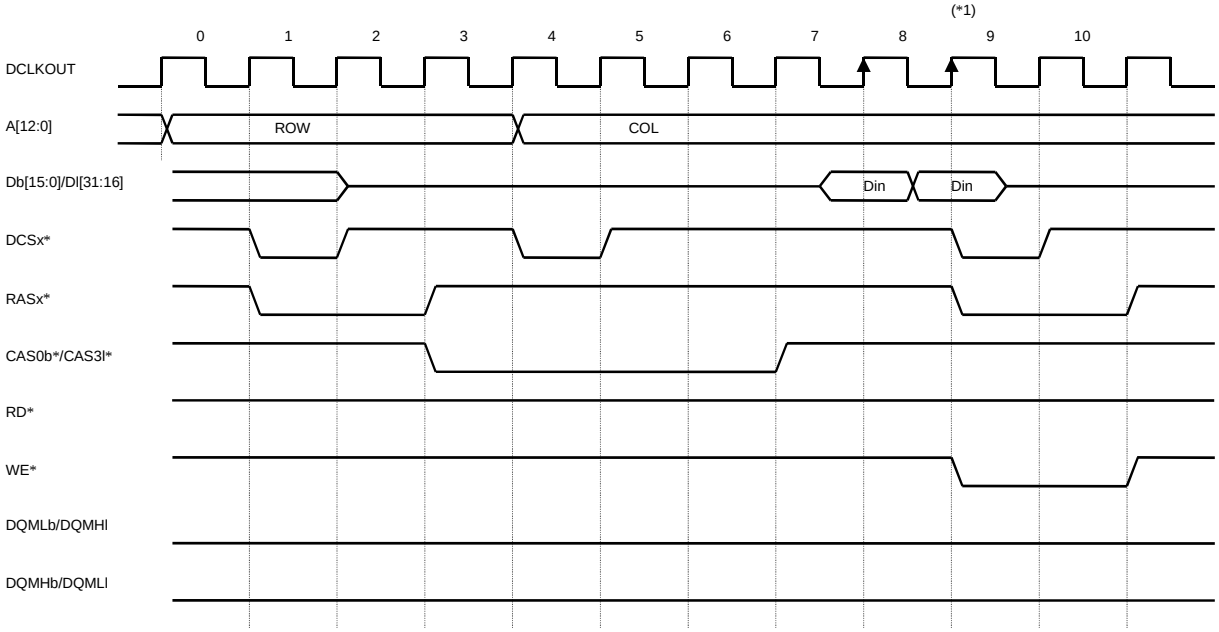


Figure B.72 16-bit SDRAM: Write Operation (5) (4 Word Burst,Write In-Page)



Note (*1): Reading is always be 32bits, without regard to the actual bytes requested by CPU.

Figure B.73 16-bit SDRAM: Read Operation (1) (Byte/Halfword/Tripleword/Word)

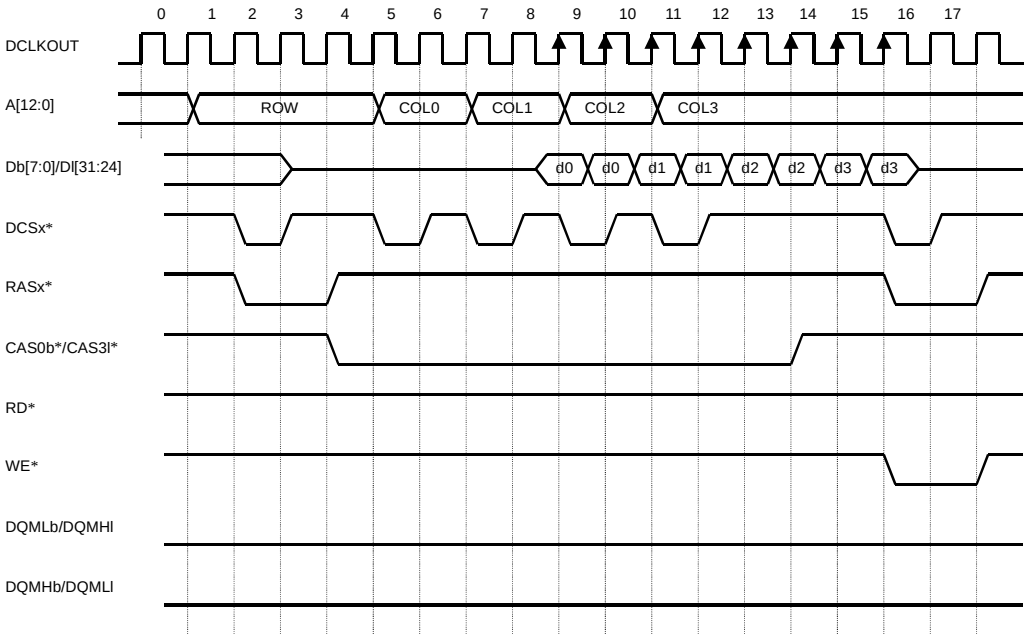
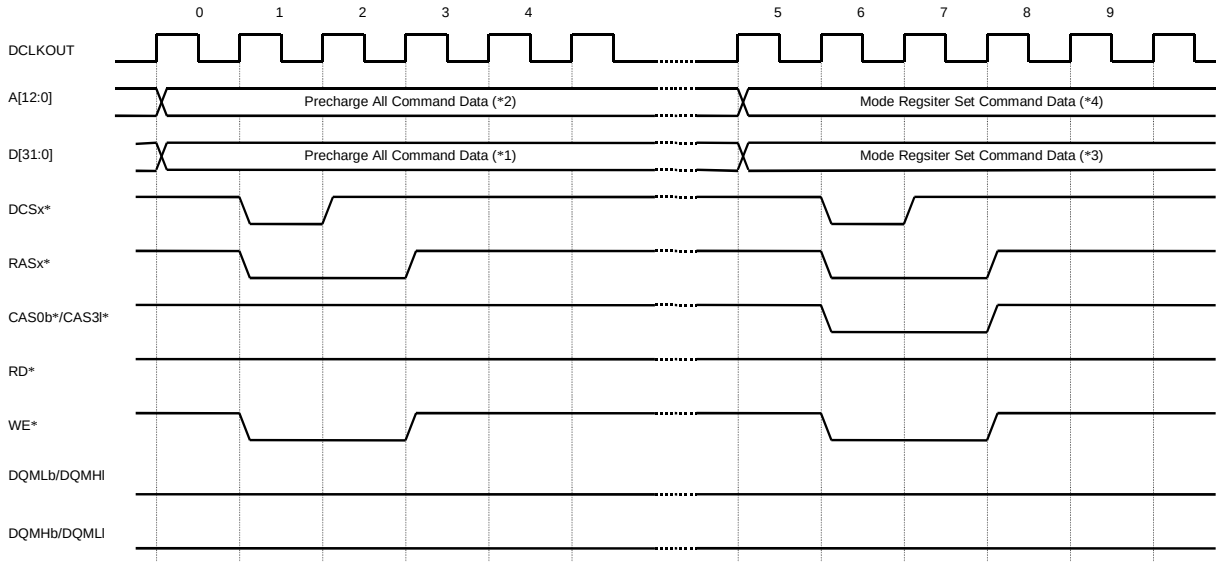


Figure B.74 16-bit SDRAM: Read Operation (2) (4 Word Burst)



Note: (*1) Data[31] (when Big Endian) or Data[7] (when Little Endian) must be “1”.
Note: (*2) Values of Data[12:0] (when Big Endian) or Data[20:16],Data[31:24] (when Little Endian) are outputed to A[12:0].
Note: (*3) Data[31] (when Big Endian) or Data[7] (when Little Endian) must be “0”.
Note: (*4) Values of Data[12:0] (when Big Endian) or Data[20:16],Data[31:24] (when Little Endian) are outputed to A[12:0].

Figure B.75 SDRAM Precharge All & SDRAM Mode Register Set

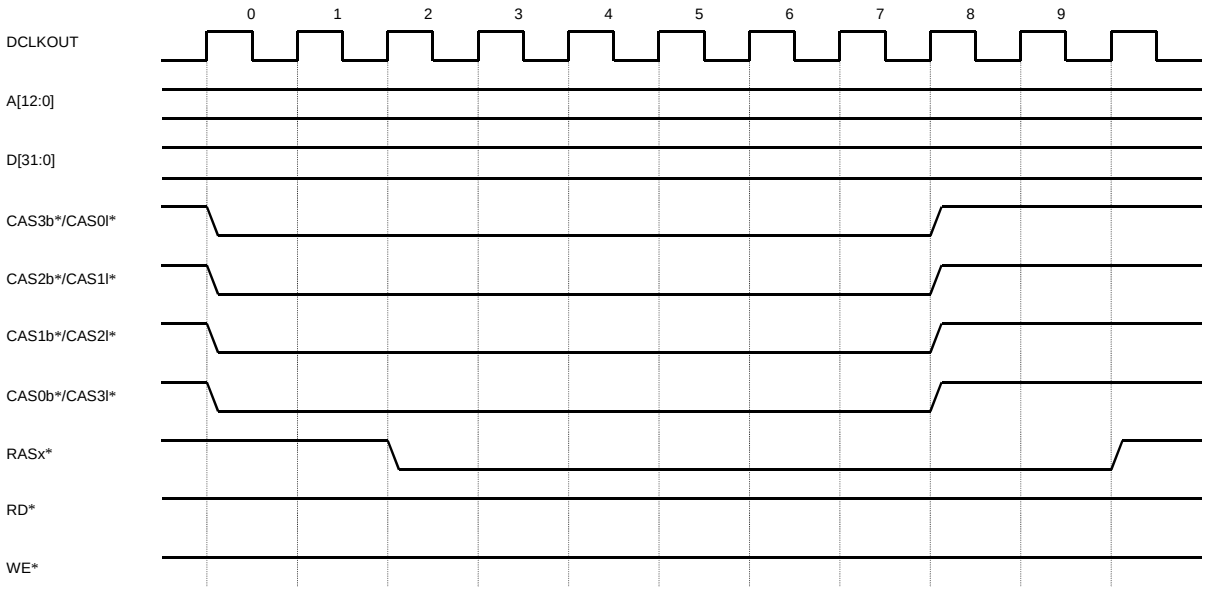


Figure B.76 DRAM Refresh Cycle

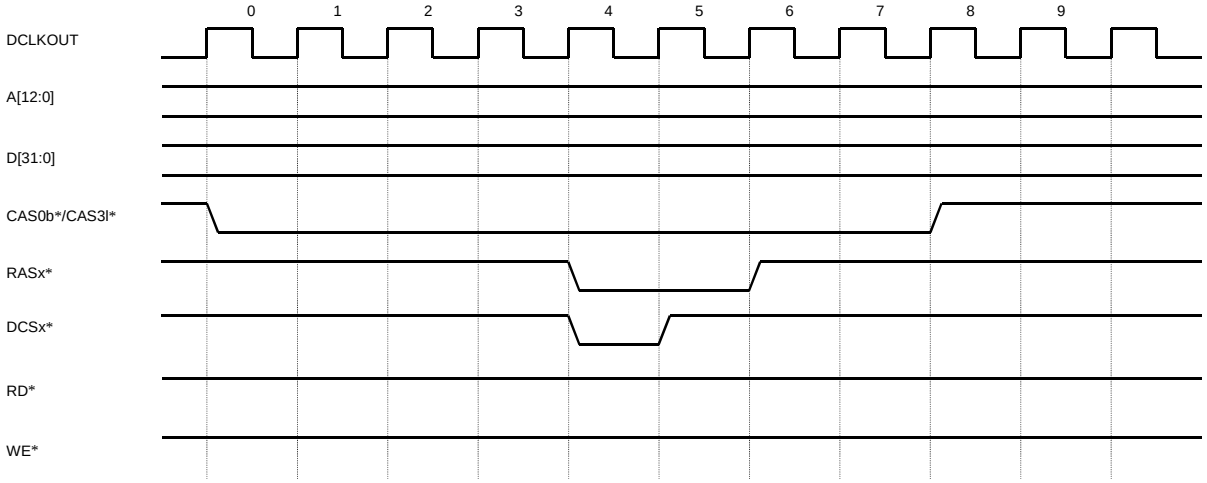


Figure B.77 SDRAM Refresh Cycle