TOSHIBA

32-Bit TX System RISC

TX19 Family

TMP1940CYAF/TMP1940FDBF

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

Toshiba offers a broad range of microcontrollers targeted for both commercial and industrial applications. The *TX System RISC TX19 Family* manual contains the detailed specifications of the TX1940, including the architecture, programming, capabilities, operation, electrical characteristics, packaging and so forth.

The TX1940 is a high-performance RISC processor based on the R3000A architecture and the MIPS16 Application Specific Extension pioneered by MIPS Technologies, Inc.

Recently, with the ever-growing market for lightweight portable devices, manufacturers of electronic systems have been seeking cost-effective, single-chip solutions to processor-based applications. Toshiba has designed the TX1940 to help customers achieve the best cost performance for their products.

Contents

TMP1940FDBF

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.

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.

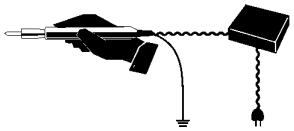
(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

(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).

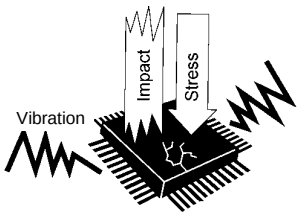
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.

- 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.

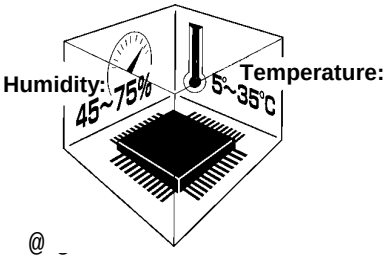
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

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.

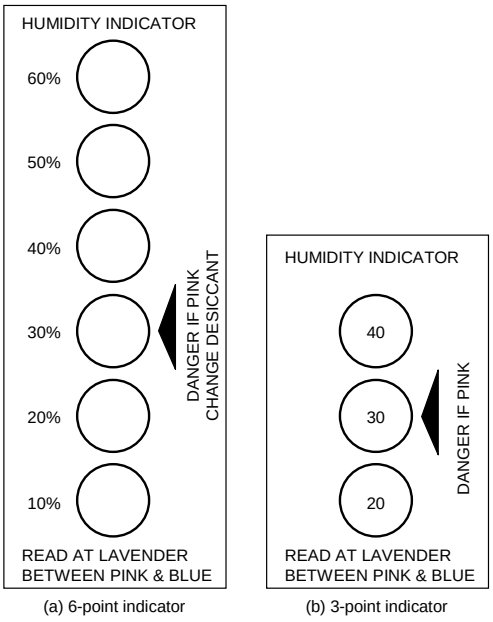


Figure 1 Humidity indicator

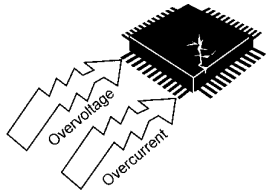
3.3 Design

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)

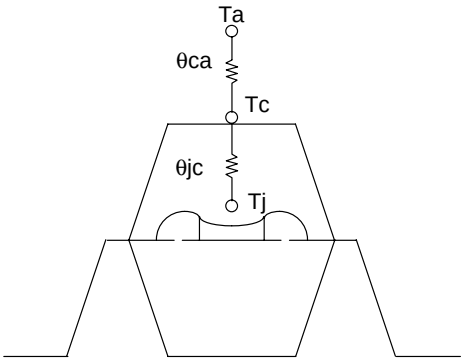


Figure 2 Thermal resistance of package

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

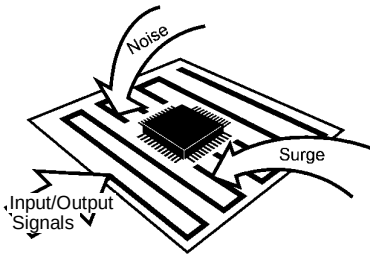
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with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

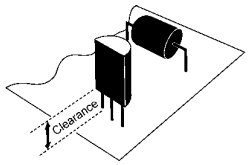
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

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).

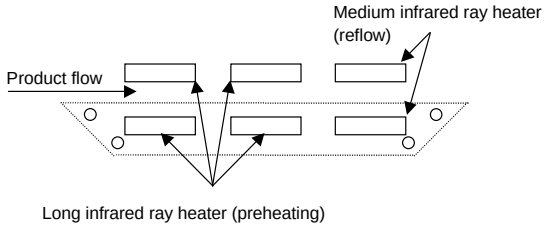


Figure 3 Heating top and bottom with long or medium infrared rays

- 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.

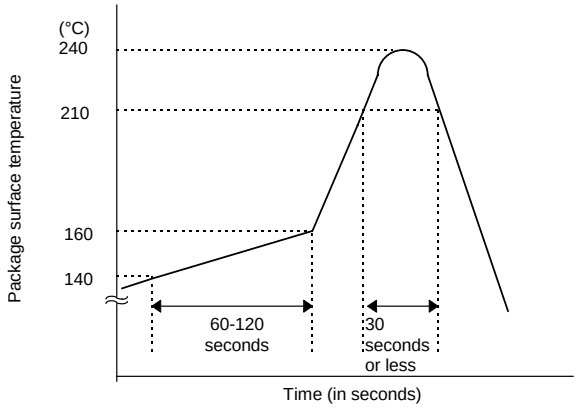


Figure 4 Sample temperature profile for infrared or hot air reflow

(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.

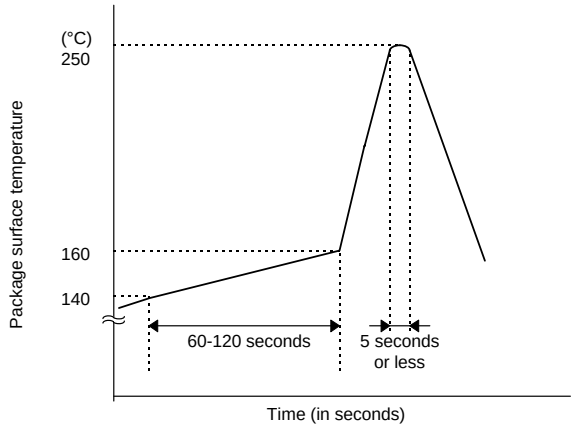


Figure 5 Sample temperature profile for solder flow

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

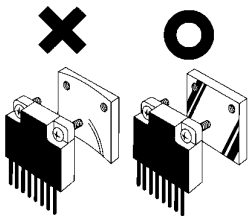
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects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

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.

(3) Scratch and puncture wounds by the point of a probe

The tips of probes and adaptors used in development tools are individually designed to be compatible with particular devices. Probes for some devices have sharp points. When you handle them bare-handed, take care not to suffer a scratch or puncture wound.

4.1.2 Reliability predictions for microcontroller devices

For microcontroller devices, the following junction temperature range is used for reliability predictions:

Tj = 0°C ~ 85°C

An estimation of the chip junction temperature, Tj, can be obtained from the equation:

Tj = Ta + Q ×•ja

where:

Ta = ambient temperature (°C)
The assumption is that the ambient temperature is not affected by any heat transfers from the device.
Q = chip's average power dissipation (W)
•ja = package thermal resistance (°C/W)

Note 1:	If you use a microcontroller device outside the 0 to 85°C range for long periods of time, contact your nearest Toshiba office or authorized Toshiba dealer.
Note 2:	For the •ja value, contact your nearest Toshiba office or authorized Toshiba dealer.

TOSHIBA

Part 1 TMP1940

TOSHIBA CORPORATION

32-Bit RISC Microprocessor TX19 Family

TMP1940CYAF

1. Features

The TX19 is a family of high-performance 32-bit microprocessors that offers the speed of a 32-bit RISC solution with the added advantage of a significantly reduced code size of a 16-bit architecture. The instruction set of the TX19 includes as a subset the 32-bit instructions of the TX39, which is based on the MIPS R3000A™ architecture. Additionally, the TX19 supports the MIPS16 Application-Specific Extensions (ASE) for improved code density.

The TMP1940 is built on a TX19 core processor and a selection of intelligent peripherals. The TMP1940 is suitable for low-voltage, low-power applications.

Features of the TMP1940 include the following:

(1) TX19 core processor

1) Two instruction set architecture (ISA) modes: 16-bit ISA for code density and 32-bit ISA for speed
   - The 16-bit ISA is object-code compatible with the code-efficient MIPS16 ASE.
   - The 32-bit ISA is object-code compatible with the high-performance TX39 family.

2) Combines high performance with low power consumption.

   — High performance
   - Single clock cycle execution for most instructions
   - 3-operand computational instructions for high instruction throughput
   - 5-stage pipeline
   - On-chip high-speed memory
   - DSP function: Executes 32-bit x 32-bit multiplier operations with a 64-bit accumulation in a single clock cycle.

   — Low power consumption
   - Optimized design using a low-power cell library
   - Programmable standby modes in which processor clocks are stopped

3) Fast interrupt response suitable for real-time control
   - Distinct starting locations for each interrupt service routine
   - Automatically generated vectors for each interrupt source
   - Automatic updates of the interrupt mask level

(2) 10-Kbyte on-chip RAM
256-Kbyte on-chip ROM
(The TMP1940FDBF has 512-Kbyte FE^{2}PROM and 16-Kbyte RAM.)

(3) External memory expansion
- 16-Mbyte off-chip address space for code and data
- External bus interface with dynamic bus sizing for 8-bit and 16-bit data ports

(4) 4-channel DMA controller
- Interrupt- or software-triggered

(5) 4-channel 8-bit timer

(6) 4-channel 16-bit timer

(7) 1-channel real-time counter (RTC)

(8) 4-channel general-purpose serial interface
Two channels support both UART and synchronous transfer modes and the other two channels are solely for UART.

(9) 1-channel serial bus interface
Either I^{2}C bus mode or clock-synchronous mode can be selected.

(10) 8-channel 10-bit A/D converter (with internal sample/hold)
Conversion time: 10.75 µs @32 MHz

(11) Watchdog timer

(12) 4-channel chip select/wait controller

(13) Interrupt sources
- 4 CPU interrupts: software interrupt instruction
- 32 internal interrupts: 7 priority levels, with the exception of the watchdog timer interrupt
- 11 external interrupts: 7 priority levels, with the exception of the NMI interrupt

(14) 77-pin input/output ports

(15) Four standby modes
- IDLE (HALT, DOZE), SLEEP, STOP

(16) Dual clocks
- Clock for low-power operation: Low-speed clock (32.768 kHz)
- RTC clock: Low-speed clock (32.768 kHz)

(17) Clock generator
- On-chip PLL (x4)
- Clock gear: Divides the operating speed of the CPU by 1/2, 1/4 or 1/8

(18) Little-endian

Higher address	31 24	23 16	15 8	7 0	Word address
↑	11	10	9	8	8
	7	6	5	4	4
Lower address	3	2	1	0	0

- Byte 0 is the lowest-order byte (bits 7-0).
- The address of a word data item is the address of its lowest-order byte (byte 0).

(19) Operating voltage range: 2.7 to 3.6 V

(20) Operating frequency

- 32 MHz (Vcc ≥ 3.0 V)
- 26 MHz (Vcc ≥ 2.7 V)

(21) Package

- 100-pin QFP (14 x 14 x 1.4 (t) mm, 0.5-mm pitch)

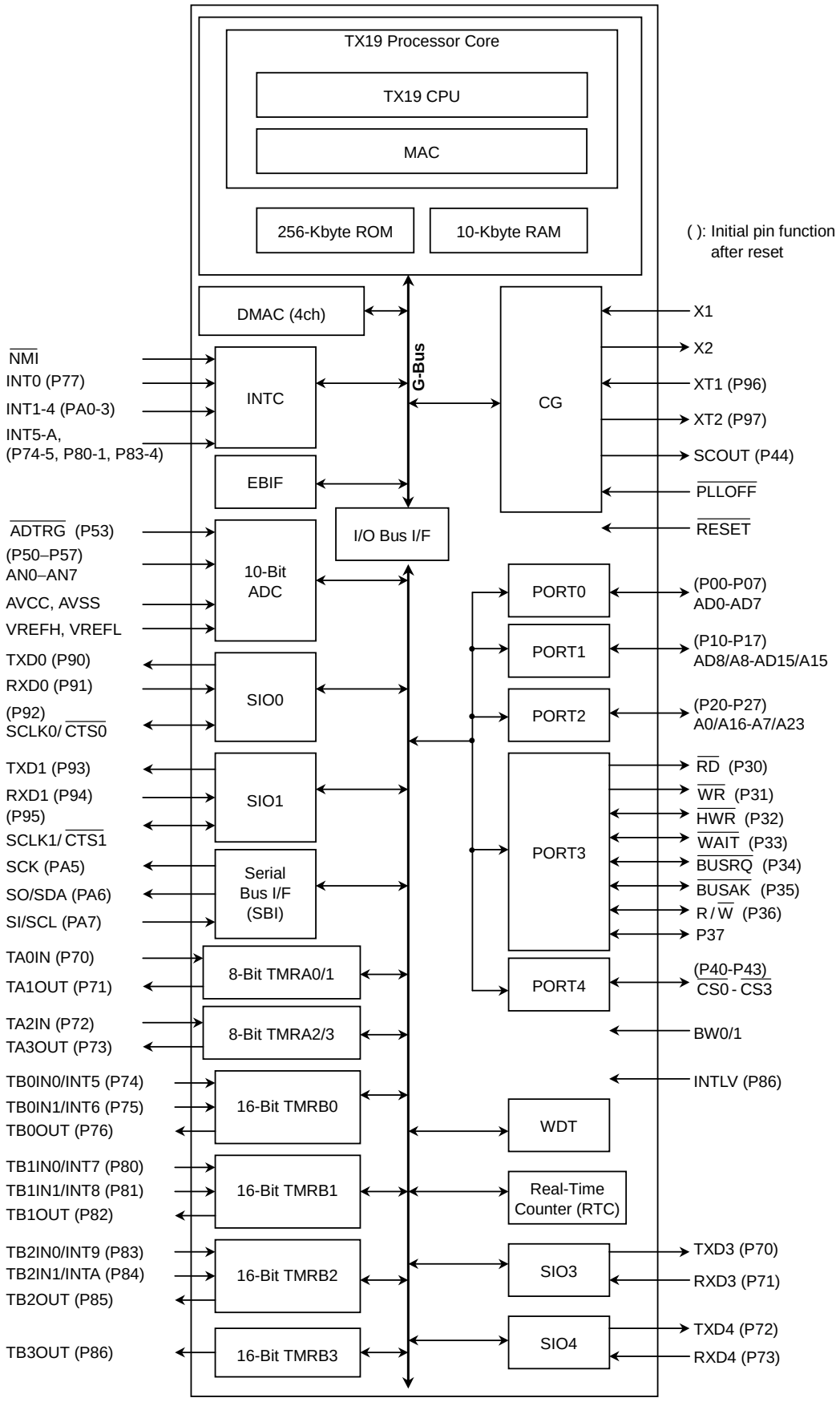


Figure 1.1 TMP1940CYAF Block Diagram

2. Signal Descriptions

This section contains pin assignments for the TMP1940CYAF as well as brief descriptions of the TMP1940CYAF input and output signals.

2.1 Pin Assignment

The following illustrates the TMP1940CYAF pin assignment.

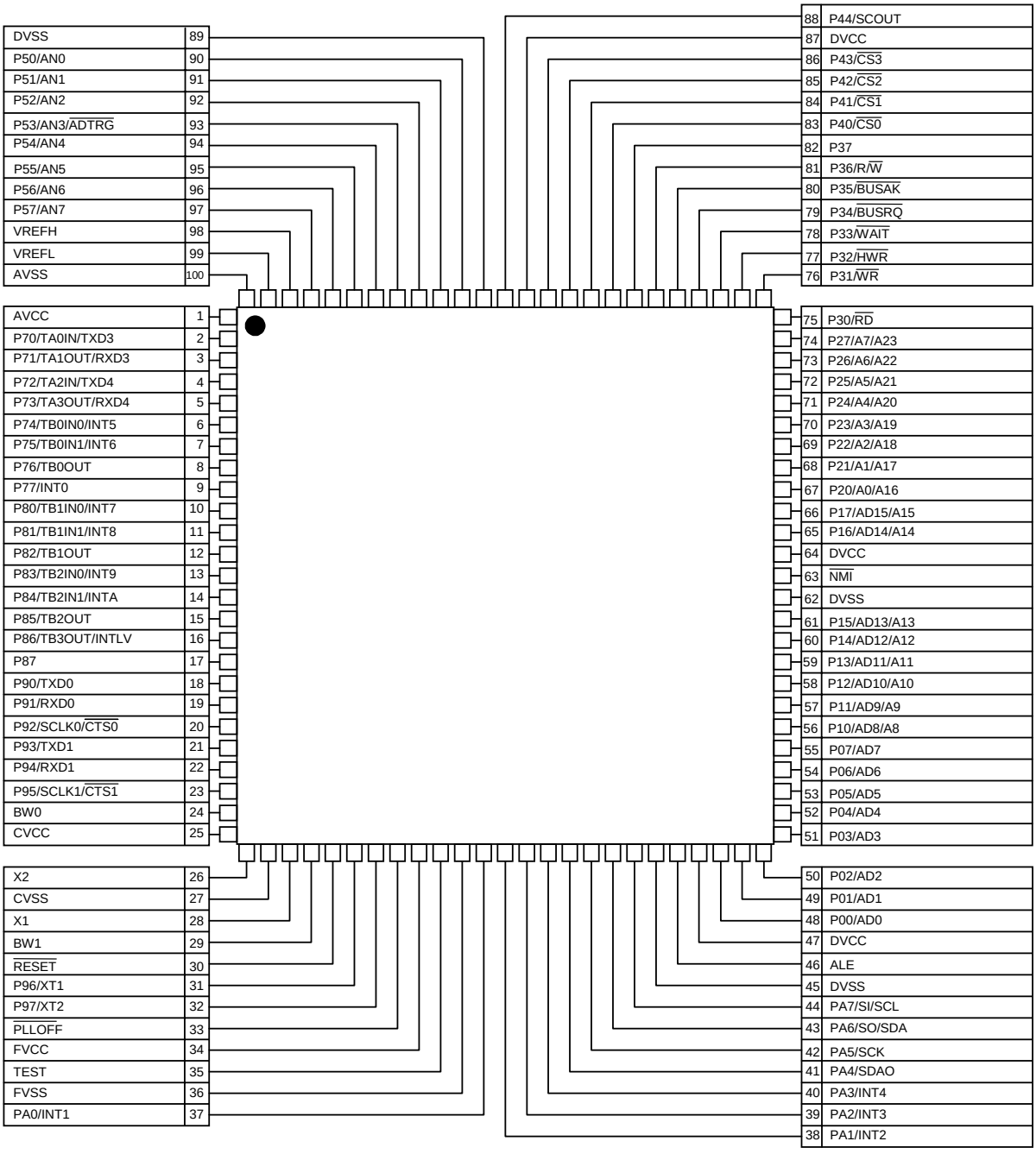


Figure 2.1 100-Pin LQFP Pin Assignment

2.2 Pin Usage Information

Table 2.1 lists the input and output pins of the TMP1940CYAF, including alternate pin names and functions for multi-function pins.

Table 2.1 Pin Names and Functions

Pin Name	# of Pins	Type	Function
P00–P07	8	Input/output	Port 0: Individually programmable as input or output
AD0–AD7		Input/output	Address (Lower): Bits 0-7 of the address/data bus
P10–P17	8	Input/output	Port 1: Individually programmable as input or output
AD8–AD15		Input/output	Address/Data (Upper): Bits 8-15 of the address/data bus
A8–A15		Output	Address: Bits 8-15 of the address bus
P20–P27	8	Input/output	Port 2: Individually programmable as input or output
A0–A7		Output	Address: Bits 0-7 of the address bus
A16–A23		Output	Address: Bits 16-23 of the address bus
P30	1	Output	Port 30: Output-only
$\overline{RD}$		Output	Read Strobe: Asserted during a read operation from an external memory device
P31	1	Output	Port 31: Output-only
$\overline{WR}$		Output	Write Strobe: Asserted during a write operation on D0-D7
P32	1	Input/output	Port 32: Programmable as input or output (with internal pull-up resister)
$\overline{HWR}$		Output	Higher Write Strobe: Asserted during a write operation on D8-D15
P33	1	Input/output	Port 33: Programmable as input or output (with internal pull-up resister)
$\overline{WAIT}$		Input	Wait: Causes the CPU to suspend external bus activity
P34	1	Input/output	Port 34: Programmable as input or output (with internal pull-up resister)
$\overline{BUSRQ}$		Input	Bus Request: Asserted by an external bus master to request bus mastership
P35	1	Input/output	Port 35: Programmable as input or output (with internal pull-up resister)
$\overline{BUSAK}$		Output	Bus Acknowledge: Indicates that the CPU has relinquished the bus in response to $\overline{BUSRQ}$.
P36	1	Input/output	Port 36: Programmable as input or output (with internal pull-up resister)
R / $\overline{W}$		Output	Read/Write: Indicates the direction of data transfer on the bus: 1 = read or dummy cycle, 0 = write cycle
P37	1	Input/output	Port 37: Programmable as input or output (with internal pull-up resister) This pin is used to select the operating mode during reset. The TMP1940CYAF enters NORMAL mode when this pin is sampled high at the rising edge of $\overline{RESET}$. This pin should not be pulled down to a logic 0 during a reset sequence. The TMP1940FDBF, which has an on-chip flash, uses this pin as an interface to the DSU tool. For details, refer to the TMP1940FDBF datasheet pages.
P40	1	Input/output	Port 40: Programmable as input or output (with internal pull-up resister)
$\overline{CS0}$		Output	Chip Select 0: Asserted low to enable external devices at programmed addresses
P41	1	Input/output	Port 41: Programmable as input or output (with internal pull-up resister)
$\overline{CS1}$		Output	Chip Select 1: Asserted low to enable external devices at programmed addresses
P42	1	Input/output	Port 42: Programmable as input or output (with internal pull-up resister)
$\overline{CS2}$		Output	Chip Select 2: Asserted low to enable external devices at programmed addresses
P43	1	Input/output	Port 43: Programmable as input or output (with internal pull-up resister)
$\overline{CS3}$		Output	Chip Select 3: Asserted low to enable external devices at programmed addresses
P44	1	Input/output	Port 44: Programmable as input or output
SCOUT		Output	System Clock Output: Drives out a clock signal at the same frequency as the CPU clock (high-speed or low-speed)
P50–P57	8	Input	Port 5: Input-only
AN0–AN7		Input	Analog Input: Input to the on-chip A/D Converter
$\overline{ADTRG}$		Input	A/D Trigger: Starts an A/D conversion (multiplexed with P53)
P70	1	Input/output	Port 70: Programmable as input or output
TA0IN		Input	8-Bit Timer 0 Input: Input to Timer 0
TXD3		Output	Serial Transmit Data 3: Programmable as a push-pull or open-drain output
P71	1	Input/output	Port 71: Programmable as input or output
TA1OUT		Output	8-Bit Timer 1 Output: Output from either Timer 0 or Timer 1
RXD3		Input	Serial Receive Data 3

Pin Name	# of Pins	Type	Function
P72	1	Input/output	Port 72: Programmable as input or output
TA2IN		Input	8-Bit Timer 2 Input: Input to Timer 2
TXD4		Output	Serial Transmit Data 4: Programmable as a push-pull or open-drain output
P73	1	Input/output	Port 73: Programmable as input or output
TA3OUT		Output	8-Bit Timer 3 Output: Output from either Timer 2 or Timer 3
RXD4		Input	Serial Receive Data 4
P74	1	Input/output	Port 74: Programmable as input or output
TB0IN0		Input	16-Bit Timer 0 Input 0: Count/capture trigger input to 16-bit Timer 0
INT5		Input	Interrupt Request 5: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P75	1	Input/output	Port 75: Programmable as input or output
TB0IN1		Input	16-Bit Timer 0 Input 1: Capture trigger input to 16-bit Timer 0
INT6		Input	Interrupt Request 6: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P76	1	Input/output	Port 76: Programmable as input or output
TB0OUT		Output	16-Bit Timer 0 Output: Output from 16-bit Timer 0
P77	1	Input/output	Port 77: Programmable as input or output
INT0		Input	Interrupt Request 0: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P80	1	Input/output	Port 80: Programmable as input or output
TB1IN0		Input	16-Bit Timer 1 Input 0: Count/capture trigger input to 16-bit Timer 1
INT7		Input	Interrupt Request 7: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P81	1	Input/output	Port 81: Programmable as input or output
TB1IN1		Input	16-Bit Timer 1 Input 1: Capture trigger input to 16-bit Timer 1
INT8		Input	Interrupt Request 8: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P82	1	Input/output	Port 82: Programmable as input or output
TB1OUT		Output	16-Bit Timer 1 Output: Output from 16-bit Timer 1
P83	1	Input/output	Port 83: Programmable as input or output
TB2IN0		Input	16-Bit Timer 2 Input 0: Count/capture trigger input to 16-bit Timer 2
INT9		Input	Interrupt Request 9: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P84	1	Input/output	Port 84: Programmable as input or output
TB2IN1		Input	16-Bit Timer 2 Input 1: Capture trigger input to 16-bit Timer 2
INTA		Input	Interrupt Request A: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P85	1	Input/output	Port 85: Programmable as input or output
TB2OUT		Output	16-Bit Timer 2 Output: Output from 16-bit Timer 2
$\overline{\text{BOOT}}$	1	Input	This pin function is used to select the operating mode during reset. The TMP1940CYAF enters NORMAL mode when this pin is sampled high at the rising edge of $\overline{\text{RESET}}$. This pin should not be pulled up to a logic 1 during a reset sequence. With the TMP1940FDBF, which has an on-chip flash, this pin is used to put the flash in Single-Boot mode. For details, refer to the TMP1940FDBF datasheet pages.
P86	1	Input/output	Port 86: Programmable as input or output
TB3OUT		Output	16-Bit Timer 3 Output: Output from 16-bit Timer 3
INTLV		Input	Interleave Mode: This pin function is used by the TMP1940FDBF with the on-chip flash. The TMP1940FDBF enters Interleave mode when this pin is sampled high at the rising edge of $\overline{\text{RESET}}$. During a reset sequence, this pin should be pulled up to a logic 1 when Interleave mode is used and pulled down to a logic 0 otherwise. For a description of when Interleave mode is required, refer to the TMP1940FDBF datasheet pages.
P87	1	Input/output	Port 87: Programmable as input or output This pin is used to select the operating mode during reset. This pin should be pulled down to a logic 0 during a reset sequence.
P90	1	Input/output	Port 90: Programmable as input or output
TXD0		Output	Serial Transmit Data 0: Programmable as a push-pull or open-drain output
P91	1	Input/output	Port 91: Programmable as input or output
RXD0		Input	Serial Receive Data 0

Pin Name	# of Pins	Type	Function
P92 SCLK0 $\overline{CTS0}$	1	Input/output Input/output Input	Port 92: Programmable as input or output Serial Clock Input/Output 0 Serial Clear-to-Send 0
P93 TXD1	1	Input/output Output	Port 93: Programmable as input or output Start Serial Transmit Data 1: Programmable as a push-pull or open-drain output
P94 RXD1	1	Input/output Input	Port 94: Programmable as input or output Serial Receive Data 1
P95 SCLK1 $\overline{CTS1}$	1	Input/output Input/output Input	Port 95: Programmable as input or output Serial Clock Input/Output 1 Serial Clear-to-Send 1
P96 XT1	1	Input/output Input	Port 96: Programmable as input or open-drain output Connection pin for a low-speed crystal
P97 XT2	1	Input/output Output	Port 97: Programmable as input or open-drain output Connection pin for a low-speed crystal
PA0–PA3 INT1–INT4	4	Input/output Input	Ports A0–A3: Individually programmable as input or output Interrupt Request 1–4: Individually programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PA4	1	Input/output	Port A4: Programmable as input or output
PA5 SCK	1	Input/output Input/output	Port A5: Programmable as input or output Clock input/output pin when the Serial Bus Interface is in SIO mode
PA6 SO SDA	1	Input/output Output Input/output	Port A6: Programmable as input or output Data transmit pin when the Serial Bus Interface is in SIO mode Data transmit/receive pin when the Serial Bus Interface is in I^2C mode; programmable as a push-pull or open-drain output
PA7 SI SCL	1	Input/output Input Input/output	Port A7: Programmable as input or output Data receive pin when the Serial Bus Interface is in SIO mode Clock input/output pin when the Serial Bus Interface is in I^2C mode; as an output, programmable as a push-pull or open-drain output
ALE	1	Output	Address Latch Enable (This signal is driven out only when external memory is accessed.)
$\overline{NMI}$	1	Input	Nonmaskable Interrupt Request: Causes an NMI interrupt on the falling edge
BW0–1	2	Input	Both BW0 and BW1 should be tied to logic 1.
TEST	1	Input	Test pin: This pin should be left open or tied to ground.
$\overline{PLLOFF}$	1	Input	This pin should be tied to logic 1 when the frequency multiplied clock from the PLL is used; otherwise, it should be tied to logic 0.
$\overline{RESET}$	1	Input	Reset (with internal pull-up resister): Initializes the whole TMP1940CYAF.
VREFH	1	Input	Input pin for high reference voltage for the A/D Converter. This pin should be connected to the AVCC pin when the A/D Converter is not used.
VREFL	1	Input	Input pin for low reference voltage for the A/D Converter. This pin should be connected to the AVSS pin when the A/D Converter is not used.
AVCC	1	—	Power supply pin for the A/D Converter. This pin should always be connected to power supply even when the A/D Converter is not used.
AVSS	1	—	Ground pin for the A/D Converter. This pin should always be connected to ground even when the A/D Converter is not used.
X1/X2	2	Input/output	Connection pins for a high-speed crystal
DVCC, CVCC	5	—	Power supply pins
DVSS, CVSS	5	—	Ground pins (0 V)

Note 1: The TMP1940FDBF, with on-chip flash memory, supports software debugging using a DSU ICE. When a DSU ICE is used, P37 and A0-A7 function as debug interface signals. For a detailed description, refer to the TMP1940FDBF datasheet pages. The TMP1940CYAF, with on-chip mask ROM, does not provide support for a DSU ICE.

Note 2: P37, P85, P86 and P87 should be held at the prescribed logic states for one system clock cycle before and after the rising edge of $\overline{RESET}$, with the $\overline{RESET}$ signal being stable in either logic state.

3. Core Processor

The TMP1940CYAF contains a high-performance 32-bit core processor called the TX19. For a detailed description of the core processor, refer to the *32-Bit TX System RISC TX19 Core Architecture* manual.

Functions unique to the TMP1940CYAF, which are not covered in the architecture manual, are described below.

3.1 Reset Operation

To reset the TMP1940CYAF, $\overline{\text{RESET}}$ must be asserted for at least 12 system clock periods after the power supply voltage and the internal high-frequency oscillator have stabilized. This time is typically 3 μs at 32 MHz when the on-chip PLL is utilized, and 6 μs otherwise. After a reset, either the PLL-multiplied clock or an external clock is selected, depending on the logic state of the $\overline{\text{PLLOFF}}$ pin. By default, the selected clock is geared down to 1/8 for internal operation.

The following occurs as a result of a reset:

- The System Control Coprocessor (CP0) registers within the TX19 core processor are initialized. For details, refer to the *32-Bit TX System RISC TX19 Core Architecture* manual.
- The Reset exception is taken. Program control is transferred to the exception handler at a predefined address. This predefined location is called exception vector, which directly indicates the start of the actual exception handler routine. The Reset exception is always vectored to virtual address 0xBFC0_0000 (which is the same as for the Nonmaskable Interrupt exception).
- All on-chip I/O peripheral registers are initialized.
- All port pins, including those multiplexed with on-chip peripheral functions, are configured as either general-purpose inputs or general-purpose outputs.

Note: A reset operation does not affect the contents of the on-chip RAM.

4. Memory Map

The mapping of virtual addresses to physical addresses is shown below.

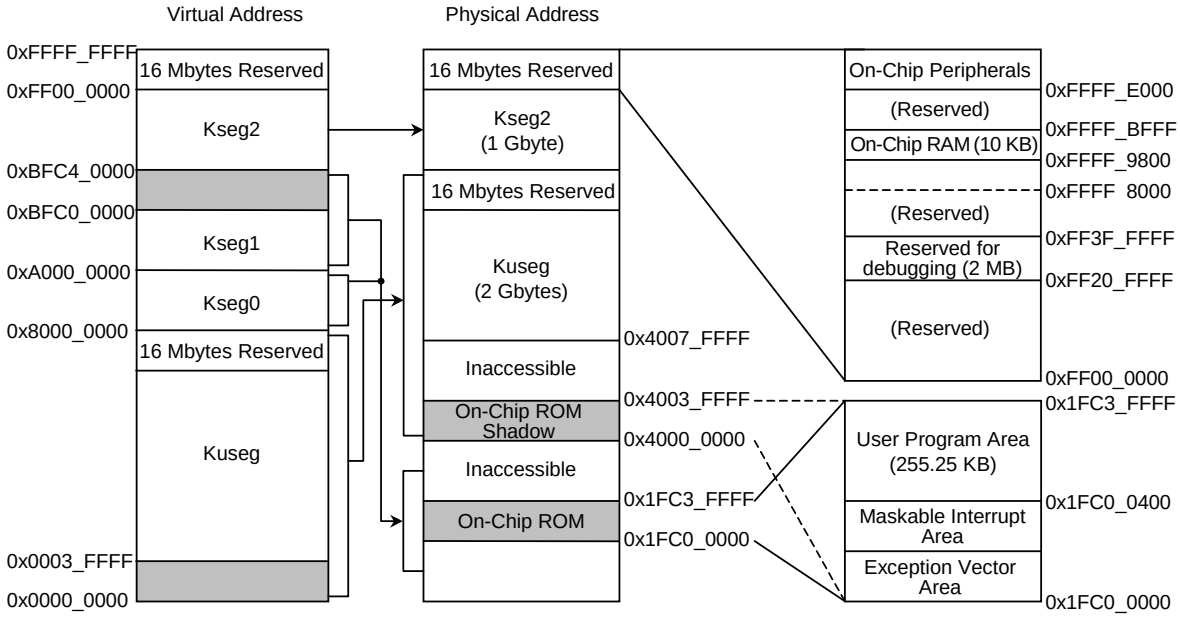


Figure 4.1 Memory Map

Note 1: **In the TMP1940CYAF, the on-chip 256-Kbyte ROM is mapped to the addresses from 0x1FC0_0000 through 0x1FC3_FFFF and the on-chip 10-Kbyte RAM is mapped to the addresses from 0xFFFF_9800 through 0xFFFF_BFFF. In the TMP1940FDBF, the on-chip 512-Kbyte flash ROM is mapped to the addresses from 0x1FC0_0000 through 0x1FC7_FFFF and the on-chip 16-Kbyte RAM is mapped to the addresses from 0xFFFF_8000 through 0xFFFF_BFFF.**

Note 2: **The on-chip ROM is located in a linear address space beginning at physical address 0x1FC0_0000. All types of exceptions are vectored to the on-chip ROM when the BEV bit of the System Control Coprocessor's Status register is set to the default value of 1. (When BEV=0, not all exception vectors reside in contiguous locations.) When external memory is used, the BEV bit can be cleared to 0. However, using the 32K-byte virtual address range beginning at 0x0000_0000 helps to improve code efficiency, as shown below. The shaded area starting at physical address 0x4000_0000 has a size equal to the on-chip ROM size. References to this range (mapped from the virtual address space starting at 0x0000_0000) are rerouted to the on-chip ROM.**

Examples: 32-bit ISA

- **Acessing the 0x0000_0000 + 32-KB region**

```
ADDIU   r2, r0, 7       ; r2 ← (0x0000_0007)
SW      r2, lo (_t) (r0)  ; 0x0000_xxxx ← (r2); Accessed with a single instruction
```

- **Accessing other regions**

```
LUI     r3, hi (_f)     ; ← Upper 16 bits of address are loaded into r3
ADDIU   r2, r0, 8       ; r 2 ← (0x0000_0008)
SW      r2, lo (_f) (r3)  ; Lower 16-bits of address must be added to upper 16 bits.
```

Note 3: **The TMP1940CYAF has access to only 16 Mbytes of external physical address space. The 16-Mbyte physical memory can be located anywhere within the CPU's 3.5-Gbyte physical address space through use of programmable chip select signals. However, any address references to the on-chip memory, on-chip peripheral or reserved regions override external memory access.**

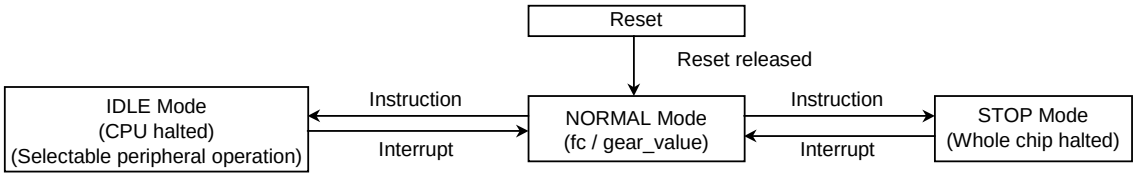
Note 4: **No instruction should be placed in the last four words of the physical address space.**

- **If only on-chip ROM is used:**
  0x1FC3_FFF0 thru 0x1FC3_FFFF of TMP1940CYAF's 256-Kbyte on-chip ROM, or
  0x1FC7_FFF0 thru 0x1FC7_FFFF in TMP1940FDBF's 512-Kbyte on-chip ROM
- **If ROM is added off-chip:**
  Last four words of the memory installed in the end-user system

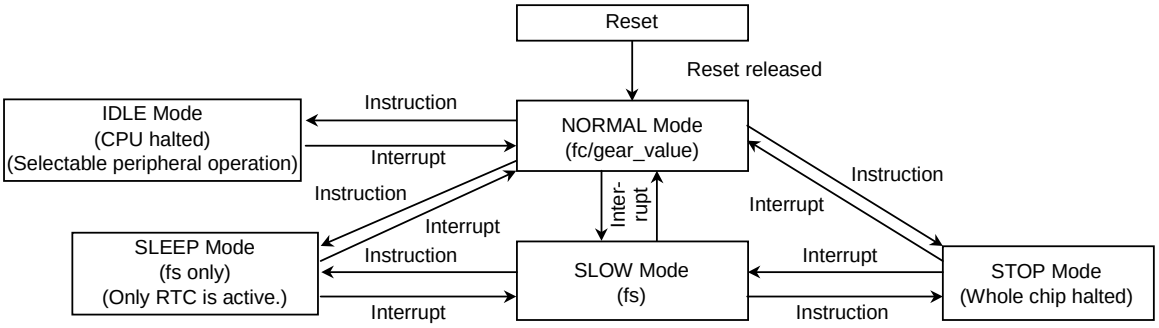
5. Clock/Standby Control

The TMP1940CYAF has two clocking modes: Single-Clock mode which operates off of the high-speed clock supplied from the X1/X2 pins, and Dual-Clock mode which operates off of the high-speed clock supplied from the X1/X2 pins and the low-speed clock supplied from the XT1/XT2 pins.

Figure 5.1 shows the transitions between clocking modes in Single-Clock mode and Dual-Clock mode.



(a) Single-Clock Mode



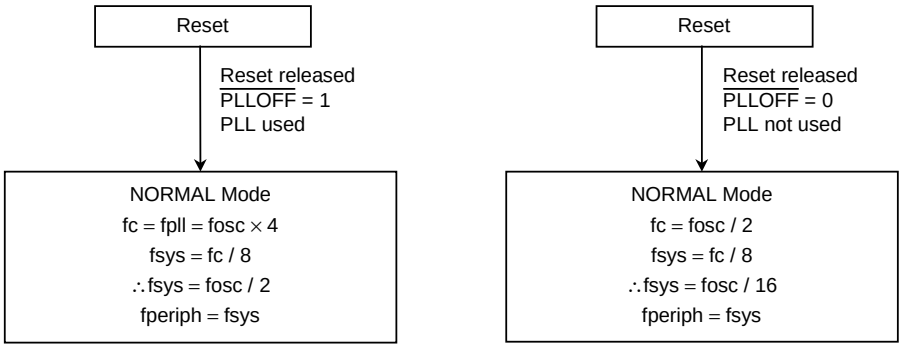
Note 1: Before a transition to SLOW or SLEEP mode can occur, the low-speed oscillator (fs) must be oscillating stably.

Note 2: After SLEEP mode is exited, the TMP1940CYAF returns to the mode it was in before entering SLEEP mode.

Note 3: After STOP mode is exited, the TMP1940CYAF returns to the mode specified by the System Control Register 0 (SYSCR0). See Section 5.2.

(b) Dual-Clock Mode

Figure 5.1 Standby Modes Flow Diagram



A. When the PLL clock is used

B. When the PLL is not used

fosc:	Clock frequency supplied via the X1 and X2 pins
fs:	Clock frequency supplied via the XT1 and XT2 pins
fpll:	PLL multiplied clock frequency (x4)
fc:	Clock frequency selected by the $\overline{\text{PLLOFF}}$ pin
fgear:	Clock frequency selected by the GEAR[1:0] bits in the SYSCR1
fsys:	System clock frequency selected by the SYSCK bit in the SYSCR1
fperiph:	Clock source for the prescalers inside on-chip peripherals

Figure 5.2 Default Clock Frequencies in NORMAL Mode

5.1 Clock Generation

5.1.1 Main System Clock

- A crystal can be connected between X1 and X2, or X1 can be externally driven with a clock.
- The on-chip PLL can be enabled or disabled (bypassed) during reset by using the $\overline{\text{PLLOFF}}$ pin. When the PLL is enabled, the input clock frequency is multiplied by four.
- The clock gear can be programmed to divide the clock by 2, 4 or 8. (The default is 1/8 on reset.)
- Input clock frequency

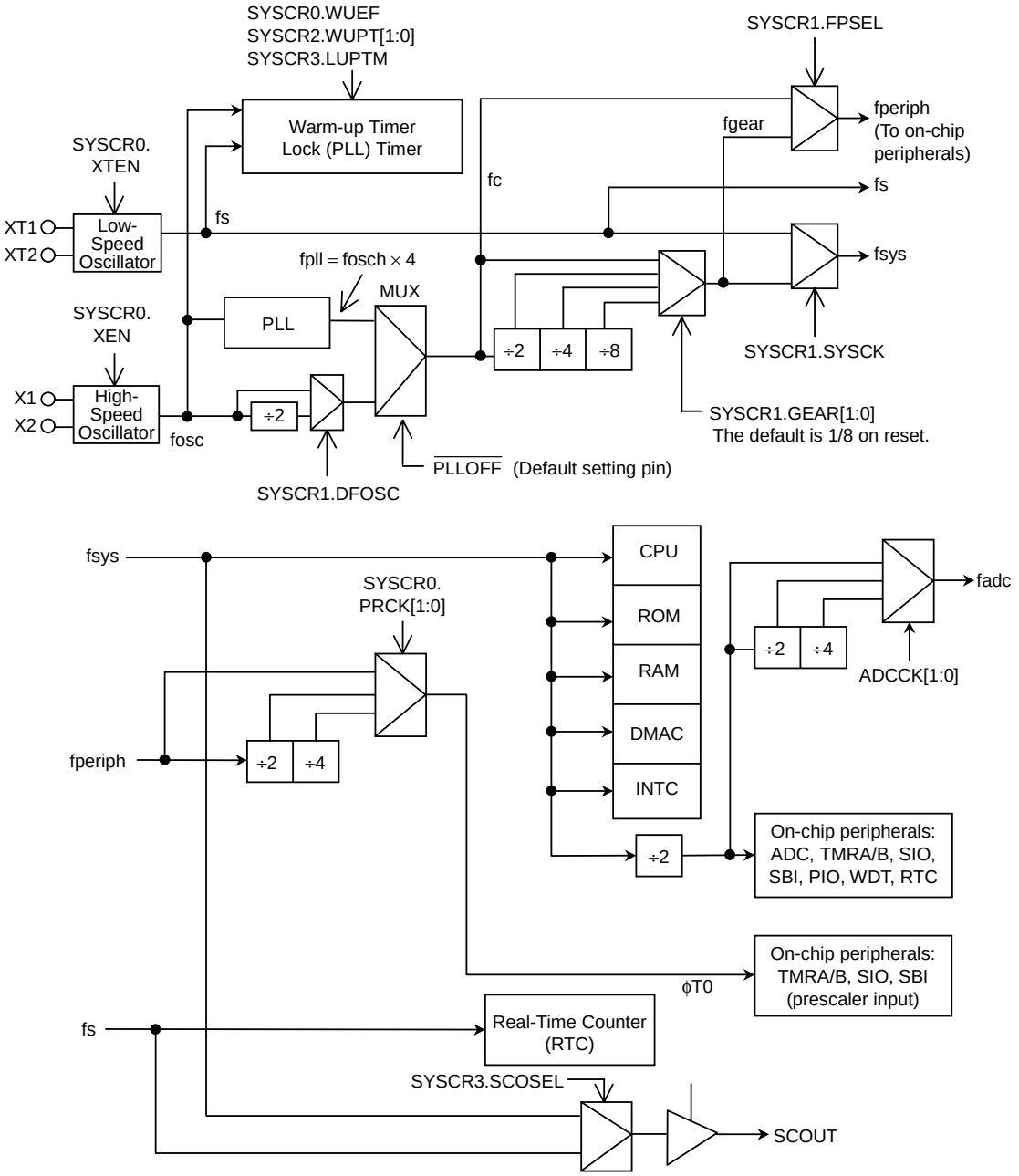
		Input Frequency Range	fmax	fmin
PLL ON (For both crystal and external clock)		5–8 MHz	32 MHz	2.5 MHz
PLL OFF	Crystal	16–20 MHz	20 MHz	1 MHz
	External clock	16–20 MHz	20 MHz	1 MHz
		20–32 MHz	16 MHz[1]	1.25 MHz

Note 1: The DFOSC bit in the SYSCR1 must be cleared to 0. The default is 0 on reset.

5.1.2 Subsystem Clock

- A 32.768-kHz crystal is connected between XT1 and XT2 (or XT1 can be externally driven with a clock.)
- SLOW mode: The CPU operates off of the low-speed clock.
- SLEEP mode: Only the Real-Time Counter (RTC) is operational.

5.1.3 Clock Source Block Diagrams



Note 1: When the clock gear is used to reduce the system clock frequency (fsys), the prescalars within on-chip peripherals must be programmed so that the prescaler output (ϕTn) satisfies the following relationship:

ϕTn < fsys / 2

Descriptions of each peripheral on the following sections include tables showing legal programming alternatives.

Note 2: When the low-speed clock (fs) is used as the system clock, all on-chip peripherals except the Watchdog Timer (WDT) and the Real-Time Counter (RTC) must be disabled.

Note 3: The presclar clock source (ϕTn) must not be changed while any of the peripherals to which it is supplied are running.

Figure 5.3 Clock Source Block Diagrams

5.2 Clock Generator (CG) Registers

5.2.1 System Clock Control Registers

		7	6	5	4	3	2	1	0
SYSCR0 (0xFFFF_EE00)	Name	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
	Read/Write	R/W							
	Reset Value	1	0	1	0	0	0	0	0
	Function	High-speed oscillator 0: Disable 1: Enable	Low-speed oscillator 0: Disable 1: Enable	High-speed oscillator after exiting STOP mode 0: Disable 1: Enable	Low-speed oscillator after exiting STOP mode 0: Disable 1: Enable	Clock select after exiting STOP mode 0: High-speed 1: Low-speed	Oscillator warm-up period (WUP) timer On writes: 0: Don't-care 1: Start WUP On reads: 0: Expired 1: Not expired	Prescaler clock select 00: fperiph/4 01: fperiph/2 10: fperiph 11: Reserved	
		15	14	13	12	11	10	9	8
SYSCR1 (0xFFFF_EE01)	Name	—	—	SYSCK	FPSEL	DFOSC	—	GEAR1	GEAR0
	Read/Write	—	—	R/W				R/W	
	Reset Value	—	—	0	0	0	—	1	1
	Function			System clock (fsys) select 0: High-speed (fgear) 1: Low-speed (fs)	fperiph select 0: fgear 1: fc	High-speed oscillator frequency divide factor 0: Divide-by-2 1: Divide-by-1		High-speed clock (fc) gear select 00: fc 01: fc/2 10: fc/4 11: fc/8	
		23	22	21	20	19	18	17	16
SYSCR2 (0xFFFF_EE02)	Name	DRVSOCH	DRVOSCL	WUPT1	WUPT0	STBY1	STBY0	—	DRVE
	Read/Write	R/W						—	R/W
	Reset Value	0	0	1	0	1	1	—	0
	Function	High-speed oscillator drive capability 0: High 1: Low	Low-speed oscillator drive capability 0: High 1: Low	Oscillator warm-up time 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		Standby mode select 00: Reserved 01: STOP mode 10: SLEEP mode 11: IDLE mode			1: Pins are driven in STOP mode.
		31	30	29	28	27	26	25	24
SYSCR3 (0xFFFF_EE03)	Name	—	SCOSEL	—	ALESEL	—	—	LUPFG	LUPTM
	Read/Write	—	R/W	—	R/W	—	—	R	R/W
	Reset Value	—	0	—	1	—	—	0	0
	Function		SCOUT output select 0: fs 1: fsys		ALE output width select 0: fsys × 0.5 1: fsys × 1.5			PLL lock 0: Locked 1: Unlocked	PLL lock time select 0: 2^{16}/input frequency 1: 2^{12}/input frequency

Note 1: The Config register in the CP0 has the Doze and Halt bits. Setting the Halt bit puts the TMP1940CYAF in one of the standby modes, as specified by the STBY1-STBY0 bits in the SYSCR2. Setting the Doze bit puts the TMP1940CYAF in IDLE mode, irrespective of the settings of the STBY1-STBY0 bits.

Note 2: When the PLL is not used, the LUPTM bit in the SYSCR3 must be set to 1 (2^{12}/input frequency).

Note 3: The WUPT1-WUPT0 bits in the SYSCR2 must not be changed during the oscillator warm-up period. The LUPTM bit in the SYSCR3 must not be changed during the PLL lock period.

Note 4: The following considerations relate to consecutive mode changes immediately after a warm-up event (e.g., SLEEP–NORMAL–SLEEP).

Hardware warm-up (with no software intervention)

(1) After having transitioned from STOP or SLEEP mode to NORMAL mode

- When the PLL is used

  A transition to a next mode can not occur until the PLL locks (SYSCR3.LUPFG=0) and at least five program instructions are executed (including the instruction to check the LUPFG flag).

- When the PLL is not used
  - When the oscillator warm-up time (SYSCR2.WUPT[1:0]) is programmed to 01 (2^8/input frequency)

    A transition to a next mode can not occur until the PLL locks (SYSCR3.LUPFG=0) and at least five program instructions are executed.
  - When the oscillator warm-up time (SYSCR2.WUPT[1:0]) is programmed to either 10 (2^{14}/input frequency) or 11 (2^{16}/input frequency)

    A transition to a next mode can not occur until at least five program instructions are executed.

(2) After having transitioned from STOP or SLEEP mode to SLOW mode

Once in SLOW mode, a transition to a next mode can occur immediately.

Software warm-up

(1) After having transitioned from SLOW mode to NORMAL mode

- When the PLL is used

  The NORMAL mode can be entered after the oscillator warm-up period timer has expired (i.e., after the SYSCR2.WUEF bit is cleared). A transition to a next mode can not occur until the PLL locks (SYSCR3.LUPFG=0) and at least five program instructions are executed (including the instruction to check the LUPFG flag).

- When the PLL is not used
  - When the oscillator warm-up time (SYSCR2.WUPT[1:0]) is programmed to either 01 (2^8/input frequency)

    The NORMAL mode can be entered after the oscillator warm-up period timer has expired (i.e., after the SYSCR2.WUEF bit is cleared). A transition to a next mode can not occur until the PLL locks (SYSCR3.LUPFG=0) and at least five program instructions are executed.
  - When the oscillator warm-up time (SYSCR2.WUPT[1:0]) is programmed to either 10 (2^{14}/input frequency) or 11 (2^{16}/input frequency)

    The NORMAL mode can be entered after the oscillator warm-up timer has expired (i.e., after the SYSCR2.WUEF bit is cleared). A transition to a next mode can not occur until at least five program instructions are executed.

(2) After having transitioned from NORMAL mode to SLOW mode

After the oscillator warm-up timer has expired (SYSCR2.WUEF=0), a transition to a next mode can not occur until at least five program instructions are executed.

5.2.2 ADC Conversion Clock

ADCCLK (0xFFFF_EE04)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	ADCCK1	ADCCK0
Read/Write	—	—	—	—	—	—	R/W	R/W
Reset Value	—	—	—	—	—	—	0	0
Function							ADC conversion clock (fadc) select 00: fsys/2 01: fsys/4 10: fsys/8 11: Don't use.	

Note: A/D conversion is executed using the clock selected by this register. Reduced conversion accuracy occurs unless the conversion time is set to 8.6 µs or more.

Relationships Between fsys Frequencies and A/D Conversion Times

fsys	Conversion Clock		
	fsys/2	fsys/4	fsys/8
32 MHz	Don't use.	10.75 µs	21.5 µs
20 MHz	8.6 µs	17.2 µs	34.4 µs
16 MHz	10.75 µs	21.5 µs	43.0 µs
10 MHz	17.2 µs	34.4 µs	68.8 µs
8 MHz	21.5 µs	43.0 µs	86.0 µs

Note: The conversion clock must not be changed while A/D conversion is in progress.

5.2.3 STOP/SLEEP Wake-up Interrupt Control Registers (INTCG Registers)

IMCGA0 (0xFFFF_EE10)

	7	6	5	4	3	2	1	0
Name	—	—	EMCG01	EMCG00	—	—	—	INT0EN
Read/Write	—	—	R/W		—	—	—	R/W
Reset Value	—	—	1	0	—	—	—	0
Function			Wake-up INT0 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT0 enable 0: Disable 1: Enable

IMCGA1 (0xFFFF_EE11)

	15	14	13	12	11	10	9	8
Name	—	—	EMCG11	EMCG10	—	—	—	INT1EN
Read/Write	—	—	R/W		—	—	—	R/W
Reset Value	—	—	1	0	—	—	—	0
Function			Wake-up INT1 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT1 enable 0: Disable 1: Enable

IMCGA2 (0xFFFF_EE12)

	23	22	21	20	19	18	17	16
Name	—	—	EMCG21	EMCG20	—	—	—	INT2EN
Read/Write	—	—	R/W		—	—	—	R/W
Reset Value	—	—	1	0	—	—	—	0
Function			Wake-up INT2 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT2 enable 0: Disable 1: Enable

		31	30	29	28	27	26	25	24
IMCGA3 (0xFFFF_EE13)	Name	—	—	EMCG31	EMCG30	—	—	—	INT3EN
	Read/Write	—	—	R/W		—	—	—	R/W
	Reset Value	—	—	1	0	—	—	—	0
	Function			Wake-up INT3 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT3 enable 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
IMCGB0 (0xFFFF_EE14)	Name	—	—	EMCG41	EMCG40	—	—	—	INT4EN
	Read/Write	—	—	R/W		—	—	—	R/W
	Reset Value	—	—	1	0	—	—	—	0
	Function			Wake-up INT4 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT4 enable 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
IMCGB1 (0xFFFF_EE15)	Name	—	—	—	—	—	—	—	—
	Read/Write	—	—	—	—	—	—	—	—
	Reset Value	—	—	1	0	—	—	—	0
	Function			Must be set to 10.					Must be set to 0.
		23	22	21	20	19	18	17	16
IMCGB2 (0xFFFF_EE16)	Name	—	—	—	—	—	—	—	—
	Read/Write	—	—	—	—	—	—	—	—
	Reset Value	—	—	1	0	—	—	—	0
	Function			Must be set to 10.					Must be set to 0.
		31	30	29	28	27	26	25	24
IMCGB3 (0xFFFF_EE17)	Name	—	—	EMCG71	EMCG70	—	—	—	INTRTCEN
	Read/Write	—	—	R/W		—	—	—	R/W
	Reset Value	—	—	1	0	—	—	—	0
	Function			Wake-up INTRTC sensitivity 00: Don't use. 01: Don't use. 10: Don't use. 11: Rising edge These bits must be set to 11.					INTRTC enable 0:Disable 1: Enable

Note 1: The edge/level sensitivity must be defined for an interrupt pin which is enabled as wake-up signaling to exit STOP/SLEEP mode.

Note 2: Interrupt programming must follow these steps:

1. Configure the pin as an interrupt input, if the pin is multiplexed with a general-purpose port.

2. Set the active state for the interrupt during initialization.

3. Clear any interrupt request.

4. Enable the interrupt.

Note 3: The above steps must be performed with the relevant interrupt pin disabled.

Note 4: The TMP1940CYAF has six interrupt sources which can be used for wake-up signaling to exit STOP/SLEEP mode: INT0 to INT4 (external interrupts) and INTRTC (internal RTC interrupt).

Note 5: When one of these interrupt sources is used for STOP/SLEEP wake-up signaling, its interrupt sensitivity defined in the CG block overrides the setting in the INTC block. In the INTC block, its senstivity must be set to the high level (which has no effect).

Example: Enabling the INT0 interrupt

IMCGA0.EMCG[01:00] = 10 } **CG block**
IMCGA0.INT0EN = 1 } **(Set the INT0 sensitivity to the falling edge)**
IMC0L.EIM[11:10] = 01 } **INTC block**
IMC0L.IL[12:10] = 101 } **(Set the interrupt sensitivity to the high level, and the interrupt priority level to 5.)**

All interrupt sources other than those used for STOP/SLEEP wake-up signaling are controlled by the INTC block.

5.2.4 Interrupt Request Clear Register

EICRCG (0xFFFF_EE20)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	ICRCG2	ICRCG1	ICRCG0
Read/Write	—	—	—	—	—	W		
Reset Value	—	—	—	—	—	—	—	—
Function						Clear interrupt request 000: INT0 100: INT4 001: INT1 101: Reserved 010: INT2 110: Reserved 011: INT3 111: INTRTC		

Note 1: Clearing the INT0-INT4 and INTRTC interrupt requests, if programmed for STOP/SLEEP wake-up signaling, requires two register settings: first, the EICRCG register in the CG block, and then the INTCLR register in the INTC block. The clearing of other interrupt sources is controlled through the INTCLR register alone.

Note 2: In cases where INT0-INT4 are not used for STOP/SLEEP wake-up signaling, they are controlled by the INTC block in the same way as other interrupt sources. INTRTC is controlled by both the CG and INTC blocks, regardless of whether it is used for wake-up signaling.

5.3 System Clock Control Section

A system reset initializes the SYSCR0.XEN bit to 1, the SYSCR0.XTEN bit to 0 and the SYSCR1.GEAR[1:0] bits to 00, putting the TMP1940CYAF in Single-Clock mode. If the on-chip PLL is enabled, the PLL reference clock is always multiplied by four. By default, the system clock frequency (fsys) is geared down to fc/8, where fc = fosc × 4 (fosc is the oscillator frequency). For example, if an 8-MHz crystal is connected between the X1 and X2 pins, the fsys clock operates at 4 MHz (8 × 4 × 1/8).

The PLL output clock can be disabled by setting the $\overline{\text{PLLOFF}}$ pin low during reset. Regardless of the logic state of the $\overline{\text{PLLOFF}}$ pin, the fsys frequency is, by default, geared down to fc/8. A reset clears the SYSCR1.DFOSC bit to 0, setting fc to fosc/2. Therefore, for example, if a 20-MHz crystal is connected between the X1 and X2 pins, fsys becomes 20 × 1/2 × 1/8 = 1.25 MHz.

Alternatively, the X1 pin can be driven with an external clock. Since the fsys clock must have a 50% duty cycle, it is recommended to use the default DFOSC bit value of 0 (i.e., fc = fosc × 1/2). However, the divide-by-2 clock generator may be bypassed by setting the DFOSC bit after reset. This causes fc to be equal to fosc; i.e., fsys becomes double the rate available when a crystal is connected between X1 and X2.

5.3.1 Oscillation Stabilization Time When Switching Between NORMAL and SLOW Modes

When a crystal is connected between the X1 and X2 pins and/or the XT1 and XT2 pins, the integrated warm-up period timer is used to assure oscillation stability. The warm-up period can be selected through the WUPT1–WUPT0 bits of the SYSCR2 to suit the crystal used. The warm-up period timer can be started by software writing a 1 to the WUEF bit in the SYSCR0. This bit is self-clearing; it can be read to ascertain that the timer has expired.

Table 5.1 shows the warm-up periods required when the clocking is switched between NORMAL and SLOW modes.

Note 1: No warm-up is necessary when the TMP1940CYAF is driven by an external oscillator clock which is already stable.

Note 2: Because the warm-up period timer is clocked by the oscillator clock, any frequency fluctuations will lead to small timer errors. Table 5.1 should be considered as approximate values.

Note 3: Ensure that the PLL lock flag (SYSCR3.LUPFG) is cleared before starting the warm-up period timer.

Note 4: When a low-speed crystal is connected between XT1 (Port 96) and XT2 (Port 97), the following register settings are required to reduce power consumption:

When a crystal is connected between XT1 and XT2:
P9CR.P96C–P97C = 11
P9.P96–P97 = 00

When XT1 is driven with an external clock:
P9CR.P96C–P97C = 11
P9.P96–P97 = 10

Table 5.1 Warm-up Periods

Warm-up Period Select SYSCR2.WUPT[1:0]	High-Speed Clock (fosc)	Low-Speed Clock (fs)
01 (2^8/ oscillation frequency)	32 (μs)	7.8 (ms)
10 (2^{14}/ oscillation frequency)	2.048 (ms)	500 (ms)
11 (2^{16}/ oscillation frequency)	8.192 (ms)	2000 (ms)

Assumption: fosc = 8 MHz, fs = 32.768 kHz

Example: Switching from NORMAL mode to SLOW mode

SYSCR2.WUPT[1:0] = xx	Select warm-up period.
SYSCR0.XTEN = 1	Enable low-speed clock (fs) oscillation.
SYSCR0.WUEF = 1	Start warm-up period (WUP) timer.
Check SYSCR0.WUEF.	Wait until SYSCR0.WUEF is cleared (i.e., the WUP expires.)
SYSCR1.SYSCK = 1	Switch system clock speed to low speed (fs).
SYSCR0.XEN = 0	Disable high-speed clock (fosc) oscillation.

5.3.2 System Clock Output

Either the fsys or fs clock can be driven out from the P44/SCOUT pin. The P44/SCOUT pin is configured as SCOUT (system clock output) by programming the Port 4 registers as follows: P4CR.P44C=1 and P4FC.P44F=1. The output clock is selected through the SYSCR3.SCOSEL bit.

Table 5.2 shows the pin states in each clocking mode when the P44/SCOUT pin is configured as SCOUT.

Table 5.2 SCOUT Output States

SCOUT Select	NORMAL/ SLOW	Standby Modes		
		IDLE	SLEEP	STOP
SCOSEL = 0	The fs clock is driven out.			Held at either 1 or 0.
SCOSEL = 1	The fsys clock is driven out.			

NOTE: The phase difference between the system clock output signal (SCOUT) and the internal clock signal can not be guaranteed.

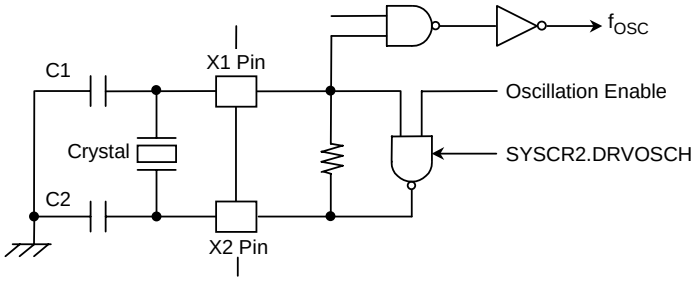
5.3.3 Reducing the Oscillator Clock Drive Capability

When a crystal is connected between the X1 and X2 pins and/or between XT1 and XT2 pins, oscillator noise and power consumption can be reduced through the programming of the SYSCR2.

Setting the SYSCR2.DRVOSCH bit reduces the drive capability of the high-speed oscillator. Setting the SYSCR2.DRVOSCL bit reduces the drive capability of the low-speed oscillator clock.

A reset clears both the DRVOSCH and DRVOSCL bits to 0, providing a high drive capability at power-up. Both the high-speed and low-speed oscillator clocks must have a high drive capability (i.e., DRVOSCH=0, DRVOSCL=0) when clocking modes are changed.

- Drive capability of the high-speed oscillator



- Drive capability of the low-speed oscillator

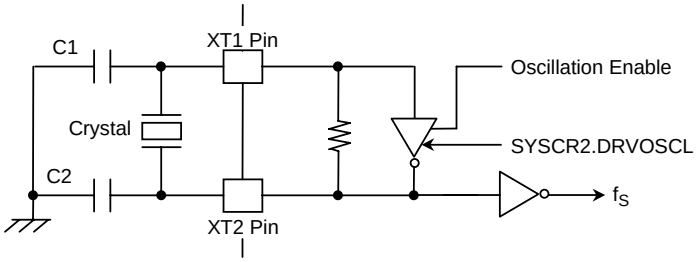


Figure 5.4 Oscillator Clock Drive Capabilities

5.4 Prescalar Clock Control Section

The TMRA01, TMRA23, TMRB0 to TMRB3, SIO0 to SIO4 (there is no SIO2), and SBI have a clock prescalar. The prescalar clock source (ϕT0) can be selected from fperiph/4, fperiph/2 and fperiph/1 through the PRCK[1:0] bits of the SYSCR0. fperiph can be selected from either fgear or fc through the FPSEL bit of the SYSCR1. The default reset values select fgear as fperiph, and fperiph/4 as ϕT0.

5.5 Clock Frequency Multiplication Section (PLL)

The on-chip PLL multiplies the frequency of the high-speed oscillator clock (fosc) by four to generate the fpll clock. At reset, the PLL is disabled. To use the PLL, the $\overline{\text{PLLOFF}}$ pin must be high when $\overline{\text{RESET}}$ is released.

Note: If the $\overline{\text{PLLOFF}}$ pin is low when $\overline{\text{RESET}}$ is released, the PLL will be disabled and the oscillator clock will be driven with no frequency multiplication.

Being an analog circuit, the PLL requires a certain duration of time (called lock time) to stabilize, like an oscillator. The oscillator warm-up period (WUP) timer is also used as the PLL lock timer. The LUPTM bit in the SYSCR3 must be programmed so that the following relationship is satisfied:

PLL lock time ≥ Oscillator warm-up time

At reset, the default lock-up time is 2^{16} / input frequency.

Setting the WUP timer control bit (SYSCR0.WUEF) starts the PLL lock timer. The SYSCR3.LUPTM bit remains set while the PLL is out of lock, and is cleared when the PLL locks.

In real-time applications whose software execution time is critical, once the PLL has gone out of lock in a standby mode, software must determine before resuming operation whether the PLL has locked (after the oscillator warm-up period timer has expired) in order to assure clock stability.

There is one thing to remember when changing the clock gear value.

The clock gear can be changed by the programming of the GEAR[1:0] bits of the SYSCR1. The RF[1:0] bits of the CPU's Config register need not be altered. It takes a few clock cycles for a gear change to take effect. Therefore, one or more instructions following the instruction that changed the clock gear value may be executed using the old clock gear value. If subsequent instructions need be executed with a new clock gear value, a dummy instruction (one that executes a write cycle) should be inserted after the instruction that modifies the clock gear value.

When the clock gear is used, the prescalars within on-chip peripherals must be programmed so that the prescaler output (ϕTn) satisfies the following relationship:

$$\phi Tn < fsys / 2$$

5.6 Standby Control Section

The TMP1940CYAF provides support for several levels of power reduction. While in NORMAL mode, setting the Halt bit of the Config register within the TX19 core processor causes the TMP1940CYAF to enter one of the standby modes — IDLE, SLEEP or STOP — as specified by the SYSCR2.STBY[1:0] bits. Setting the Doze bit of the Config register causes the TMP1940CYAF to enter IDLE (Doze) mode, irrespective of the setting of SYSCR2.STBY[1:0].

Prior to a transition to any of the standby modes, all interrupts other than those used for wake-up signaling must be disabled through the Interrupt Controller (INTC).

The characteristics of the IDLE, SLEEP and STOP modes are as follows:

IDLE: The CPU stops.

On-chip peripherals can be selectively enabled and disabled through use of a register bit in a given peripheral, as shown in Table 5.3.

Table 5.3 IDLE Mode Register Settings

Peripheral	IDLE Mode Bit
TMRA01	TA01RUN.I2TA01
TMRA23	TA23RUN.I2TA23
TMRB0	TB0RUN.I2TB0
TMRB1	TB1RUN.I2TB1
TMRB2	TB2RUN.I2TB2
TMRB3	TB3RUN.I2TB3
SIO0	SC0MOD1.I2S0
SIO1	SC1MOD1.I2S1
SIO3	SC3MOD1.I2S3
SIO4	SC4MOD1.I2S4
SBI	SBI0BR1.I2SBI0
ADC	ADMOD1.I2AD
WDT	WDMOD.I2WDT

Note 1: In Halt mode (i.e., a standby mode entered by setting the Halt bit in the Config register), the TMP1940CYAF freezes the TX19 core processor, preserving the pipeline state. In Halt mode, the TMP1940CYAF ignores any external bus requests; so it continues to assume bus mastership.

Note 2: In Doze mode (i.e., a standby mode entered by setting the Doze bit in the Config register), the TMP1940CYAF freezes the TX19 core processor, preserving the pipeline state. In Doze mode, the TMP1940CYAF recognizes external bus requests.

SLEEP: Only the internal low-speed oscillator and the RTC are operational.

STOP: The whole TMP1940CYAF stops.

5.6.1 TMP1940CYAF Operation in NORMAL and Standby Modes

Table 5.4 TMP1940CYAF Operation in NORMAL and Standby Modes

Operation Mode	Operating States
NORMAL	The TX19 core processor and peripherals operate at frequencies specified in the CG block.
IDLE (Halt)	The processor and DMAC operations stop; other on-chip peripherals can be selectively disabled.
IDLE (Doze)	Processor operation stops; the DMAC is operational; other on-chip peripherals can be selectively disabled.
SLEEP	Processor operation stops; of the on-chip peripherals, only the RTC is operational (at fs).
STOP	All processor and peripheral operations stop completely.

5.6.2 CG Operation in NORMAL and Standby Modes

Table 5.5 CG States in NORMAL and Standby Modes

Clock Source	Mode	Oscillator	PLL	Clock Supply to Peripherals	Clock Supply to CPU
Crystal	NORMAL	On	On	Yes	Yes
	SLOW	On	Off	Partially supplied (See Note.)	Yes
	IDLE (Halt)	On	On	Selectable	No
	IDLE (Doze)	On	On	Selectable	No
	SLEEP	fs only	Off	RTC only	No
	STOP	Off	Off	No	No
External Clock	NORMAL	Off	On	Yes	Yes
	SLOW	Off	Off	Partially supplied (See Note.)	Yes
	IDLE (Halt)	Off	On	Selectable	No
	IDLE (Doze)	Off	On	Selectable	No
	SLEEP	Off	Off	RTC only	No
	STOP	Off	Off	No	No

Note: The INTC, External Bus Interface (EBIF), I/O ports, WDT and RTC can operate in SLOW mode.

5.6.3 Processor and Peripheral Block Operation in Standby Modes

Table 5.6 Processor and Peripheral Blocks in Standby Modes

Circuit Block	Clock Source	IDLE (Doze)	IDLE (Halt)	SLEEP	STOP
TX19 Core Processor	fsys	Off	Off	Off	Off
DMAC		On	Off	Off	Off
INTC		On	On	Off	Off
EBIF		On	On	Off	Off
External Bus Mastership		On	On	Off	Off
I/O Ports		On	Off	Off	Off
ADC		Selectable on a block-by-block basis		Off	Off
SIO				Off	Off
I2C				Off	Off
Timer Counters				Off	Off
WDT				Off	Off
RTC	fs	On	On	On	Off
CG	—	On	On	On	Off

5.6.4 Wake-up Signaling

There are two ways to exit a standby mode: an interrupt request or reset signal. Availability of wake-up signaling depends on the settings of the Interrupt Mask Level bits, CMask[15:13], of the CP0 Status register and the current standby mode (see Table 5.7).

- Wake-up via Interrupt Signaling

  The operation upon return from a standby mode varies, depending on the interrupt priority level programmed before entering a standby mode. If the interrupt priority level is greater than the processor's interrupt mask level, execution resumes with the interrupt service routine. Upon completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated the standby mode (i.e., the instruction that set the Halt or Doze bit in the Config register).

  If the interrupt priority level is equal to or less than the processor's interrupt mask level, program execution resumes with the instruction that activated the standby mode. The interrupt is left pending.

  Nonmaskable interrupts are always serviced upon return from a standby mode, regardless of the current interrupt mask level.

- Wake-up via Reset Signaling

  Reset signaling always brings the TMP1940CYAF out of any standby mode. A wake-up from STOP mode must allow sufficient time for the oscillator to restart and stabilize (see Table 5.1).

  A reset does not affect the contents of the on-chip RAM, but initializes everything else, whereas an interrupt preserves all internal states that were in effect before the standby mode was entered.

Table 5.7 Wake-up Signaling Sources and Wake-up Operations

Interrupt Masking			Unmasked Interrupt (request_level > mask_level)			Masked Interrupt (request_level ≤ mask_level)		
Standby Mode			IDLE (Programmable)	SLEEP	STOP	IDLE (Programmable)	SLEEP	STOP
Wake-up Signaling Sources	Interrupts	NMI	✓	✓	✓[1]	✓	✓	✓[1]
		INTWDT	✓	–	–	✓	–	–
		INT0–4	✓	✓	✓[1]	❖	❖	❖[1]
		INTRTC	✓	✓	–	❖	❖	–
		INT5–A	✓	–	–	❖	–	–
		INTTA0–3	✓	–	–	❖	–	–
		INTTB00–31 INTTBOF0–3	✓	–	–	❖	–	–
		INTRX0–4 INTTX0–4	✓	–	–	❖	–	–
		INTS2	✓	–	–	❖	–	–
		INTAD	✓	–	–	❖	–	–
		INTDMA[2]	✓	–	–	❖	–	–
	$\overline{\text{RESET}}$		✓	✓	✓	✓	✓	✓

✓: Execution resumes with the interrupt service routine. ($\overline{\text{RESET}}$ initializes the whole TMP1940CYAF.)

❖: Execution resumes with the instruction that activated the standby mode. The interrupt is left pending.

–: Cannot be used to exit a standby mode.

Note 1: The TMP1940CYAF exits the standby mode after the warm-up period timer expires.

Note 2: INTDMA is accepted only in IDLE (Doze) mode.

Note 3: If the interrupt request level is greater than the mask level, an interrupt signal which is programmed as level-sensitive must be held active until interrupt processing begins. Otherwise, the interrupt will not be serviced successfully.

Note 4: If interrupts are disabled in the CPU, all interrupts other than those used for wake-up signaling must also be disabled in the Interrupt Controller (INTC) before a standby mode is entered. Otherwise, any interrupt could take the TMP1940CYAF out of the standby mode.

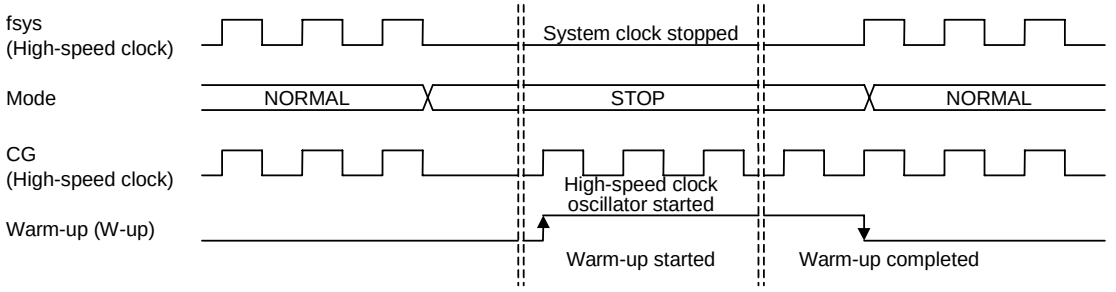
5.6.5 STOP Mode

The STOP mode stops the whole TMP1940CYAF, including the on-chip oscillator. Pin states in STOP mode depend on the setting of the SYSCR2.DRVE bit, as shown in Table 5.8. Upon detection of wake-up signaling, the warm-up period timer should be activated to allow sufficient time for the oscillator to restart and stabilize before exiting STOP mode. After that, the system clock output can restart. On exiting STOP mode, the TMP1940CYAF enters either NORMAL or SLOW mode, as programmed by the RXEN, RXTEN and RSYSCK bits of the SYSCR0.

These register bits must be programmed prior to the instruction that activates a standby mode. The warm-up period is chosen through the SYSCR2.WUPT[1:0] bits.

5.6.6 Returning from a Standby Mode

(1) Mode transitions from NORMAL to STOP to NORMAL

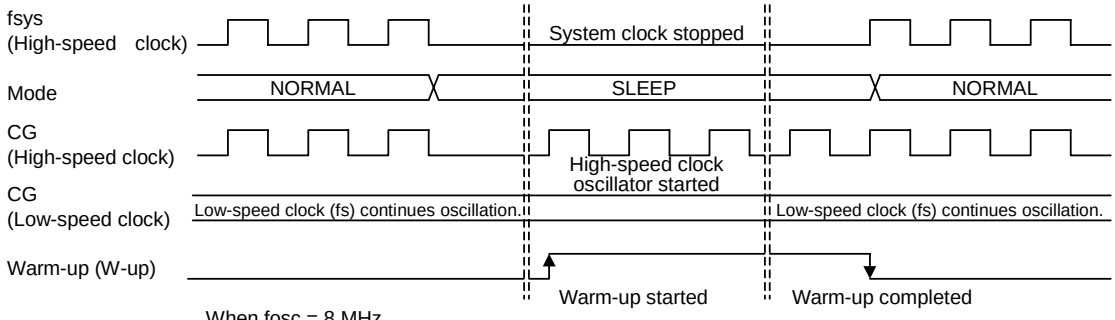


When fosc = 8 MHz

W-up Time Select SYSCR2.WUPT[1:0]	W-up Time (fc)
01 (2^8/fosc)	Don't use (Note)
10 (2^{14}/fosc)	2.048 ms
11 (2^{16}/fosc)	8.192 ms

Note: In the TMP1940FDBF, with an integrated flash memory, the WUPT[1:0] bits must not be set to 01 because this does not allow enough time for the internal system to resume.

(2) Mode transitions from NORMAL to SLEEP to NORMAL

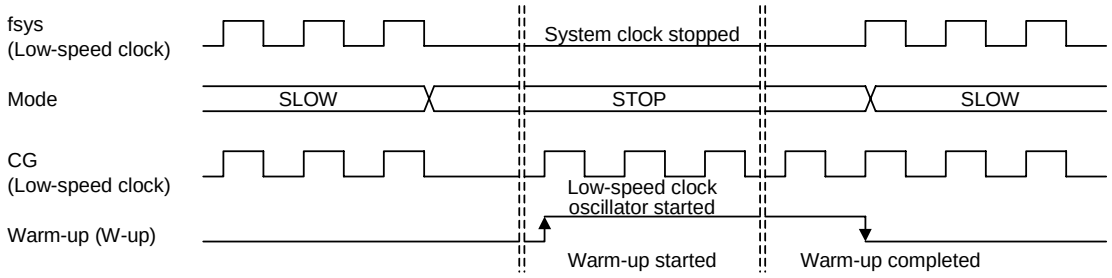


When fosc = 8 MHz

W-up Time Select SYSCR2.WUPT[1:0]	W-up Time (fc)
01 (2^8/fosc)	Don't use (Note)
10 (2^{14}/fosc)	2.048 ms
11 (2^{16}/fosc)	8.192 ms

Note: In the TMP1940FDBF, with an integrated flash memory, the WUPT[1:0] bits must not be set to 01 because this does not allow enough time for the internal system to resume.

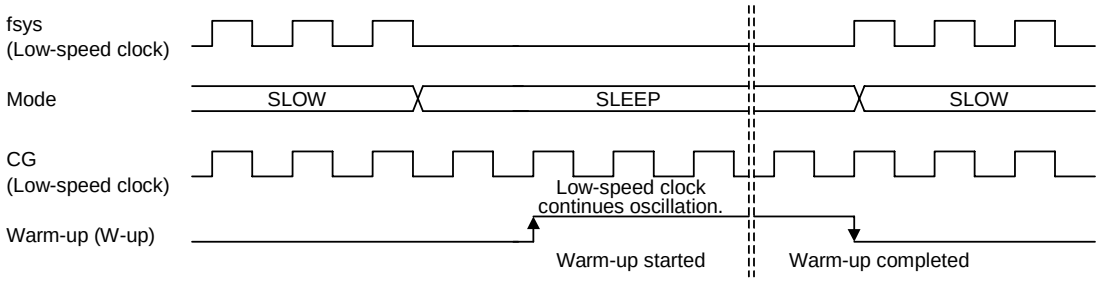
(3) Mode transitions from SLOW to STOP to SLOW



When fosc = 32.768 kHz

W-up Time Select SYSCR2.WUPT[1:0]	W-up Time (fc)
01 (2^8/fosc)	7.8 ms
10 (2^{14}/fosc)	500 ms
11 (2^{16}/fosc)	2000 ms

(4) Mode transitions from SLOW to SLEEP to SLOW



When fosc = 32.768 kHz

W-up Time Select SYSCR2.WUPT[1:0]	W-up Time (fc)
01 (2^8/fosc)	7.8 ms
10 (2^{14}/fosc)	500 ms
11 (2^{16}/fosc)	2000 ms

Note 1: Although the fs clock continues oscillation, a warm-up time must be specified.

Note 2: For the TMP1940FDBF with an on-chip flash, when the $\overline{\text{RESET}}$ signal is used for STOP/ SLEEP wake-up signaling, it must be held active for at least 500 µs for the internal system to stabilize.

Table 5.8 Pin States in STOP Mode

Pins	Input/Output	SYSCR2.DRVE = 0	SYSCR2.DRVE = 1
P00–07	Input mode	—	—
	Output mode	—	Output
	AD0–AD7	—	—
P10–17	Input mode	—	—
	Output mode	—	Output
	AD8–AD15	—	—
P20–27	Input mode	—	—
	Output mode, A0–A7/A16–A23	—	Output
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	Output pin	—	Output
P32–37	Input mode	PU*	Input
	Output mode	PU*	Output
P40–43	Input mode	PU*	Input
	Output mode	PU*	Output
P44 (SCOUT)	Input mode	—	Input
	Output mode	—	Output
P50–57	Input pin	—	—
P70–76	Input mode	—	Input
	Output mode	—	Output
P77 (INT0)	Input mode	—	Input
	Output mode	—	Output
	Input mode (INT0)	Input	Input
P80–87	Input mode	—	Input
	Output mode	—	Output
P90–95	Input mode	—	Input
	Output mode	—	Output
P96 (XT1) – P97 (XT2)	Input mode	—	Input
	Output mode	—	Output
	XT1, XT2	—	—
PA0–PA3	Input mode	—	Input
	Output mode	—	Output
	Input mode (INT1–INT4)	Input	Input
PA4–PA7	Input mode	—	Input
	Output mode	—	Output
$\overline{NMI}$	Input pin	Input	Input
ALE	Output pin	Output Low	Output Low
$\overline{RESET}$	Input pin	Input	Input
AM0, AM1	Input pin	Input	Input
X1	Input pin	—	—
X2	Output pin	Output High	Output High

—: Pins configured for input mode and input-only pins are disabled. Pins configured for output mode and output-only pins assume the high-Impedance state.

Input: The input gate is active; the input voltage must be held at either the high or low level to keep the input pin from floating.

Output: Pin direction is output.

PU*: Programmable pull-up. Because the input gate is always disabled, no overlap current flows while in high-impedance state.

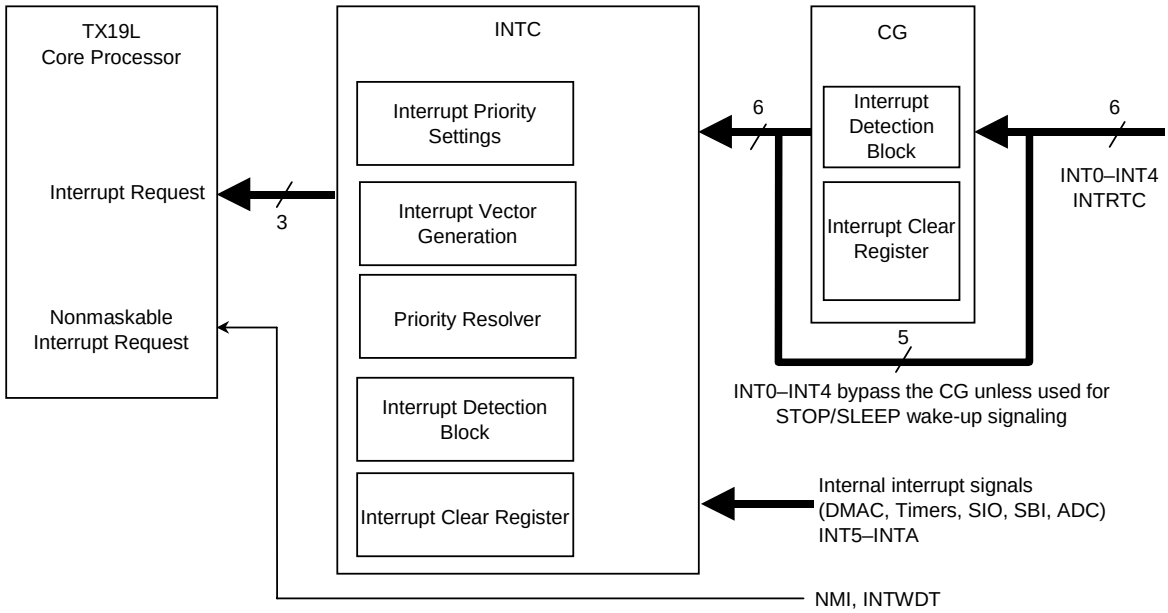
6. Interrupts

6.1 Overview

Interrupt processing is coordinated bewtween the CP0 Status register, the Interrupt Controller (INTC) and the Clock Generator (CG). The Status register contains the Interrupt Mask Level field (CMask[15:13]) and the Interrupt Enable bit (IEc). For interrupt processing, also refer to the *32-Bit TX System RISC TX19 Core Architecture* manual.

The TMP1940CYAF interrupt mechanism includes the following features:

- 4 CPU internal interrupts (software interrupts)
- 12 external interrupt pins ($\overline{\text{NMI}}$, INT0 through INTA)
- 32 on-chip peripheral interrupts
- Vector generation for each interrupt source
- Programmable priority for each interrupt source (7 levels)
- DMA trigger on interrupt



Note 1: There are interrupt enable and polarity bits in these registers:

- **Interrupt Mode Control registers (IMCxx) in the INTC**
- **IMCGxx registers in the CG**

Note 2: The TX1940CYAF provides six interrupt sources, INT0–INT4 and INTRTC, that can be used for STOP/SLEEP wake-up signaling. External interrupts INT5–INTA cannot function as wake-up signals.

Figure 6.1 General Interrupt Mechanism

The Interrupt Detection block monitors interrupt events. Each interrupt source can be individually programmed for active polarity and either level or edge sensitivity. The TMP1940CYAF interrupts are broadly grouped as follows:

- External interrupts INT0–INT4 and INTRTC
  - When enabled for STOP/SLEEP wake-up signaling

    The TMP1940CYAF awakens from STOP or SLEEP mode, if so programmed, when any of the external interrupts INT0–INT4 or INTRTC is asserted. The EMCGxx field in the IMCGxx register

defines the interrupt polarity. The INTxEN bit in the IMCGxx register controls whether these interrupt sources are enabled as wake-up signal sources (1=enable). If enabled, the interrupt polarity (EIMxx) field in the INTC's IMCxx register has no effect, but must be set to 01, or high level. The ILxx field in the IMCxx register determines the action taken after exiting STOP/SLEEP mode; i.e., whether execution resumes with an interrupt service routine.

- When disabled for STOP/SLEEP wake-up signaling

  If INT0–INT4 are disabled for STOP/SLEEP wake-up signaling, the INTC alone determines the polarity and enabling of these interrupt sources. INTRTC is programmed through both the CG and INTC, regardless of whether it is used for wake-up signaling.

- External interrupts INT5–INTA and internal interrupts except INTRTC

  These interrupts are programmable through the INTC.

The INTC collects interrupt events, prioritizes them and presents the highest-priority request to the TX19 core processor. Hardware interrupts are summarized below.

Interrupt		Programming	Interrupt Sensing
INT0–INT4		IMCGxx reg. in CG IMCx reg. in INTC	When enabled for STOP/SLEEP wake-up signaling, the polarity field in the INTC has no effect, but must always be set to "high-level." The actual sensitivity is programmed in the CG. When disabled for STOP/SLEEP wake-up signaling, interrupt sensitivity is programmed in the INTC. In either case, each interrupt source is individually configurable as negative or positive polarity, and as edge-triggered or level-sensitive.
INTRTC		IMCGxx reg. in CG IMCx reg. in INTC	In the INTC, the polarity must always be set to "high-level." The actual sensitivity must be configured as rising-edge triggered in the CG.
INT0–INTA		IMCx reg. in INTC	Configurable as negative or positive polarity, and as edge-triggered or level-sensitive.
On-Chip Peripherals	INTDMAn	IMCx reg. in INTC	Falling edge
	Other	IMCx reg. in INTC	Rising edge

Here are example register settings required to enable and disable the INT0 interrupt as a source of the STOP/SLEEP wake-up signal (negative-edge triggered).

- Enabling the interrupt

```
IMCGA0.EMCG[01:00] = 10     : Configure INT0 as negative-edge triggered  }
EICRCG.ICRCG[2:0] = 000     : Clear INT0 request                         } CG block
IMCGA0.INT0EN = 1           : Enable INT0 for wake-up signaling          }
IMC0L.EIM[11:10] = 01       : Configure  INT0 as high-level sensitive    }
INTCLR.EICLR[5:0] = 000001  : Clear INT0 request                         } INTC block
IMC0L.IL[12:10] = 101       : Set INT0 priority level to 5               }
Status.IEc = 1, Status.CMask = xxx                                       TX19 core processor
```

- Disabling the interrupt

```
Status.IEc = 0                                                           TX19 core processor
IMC0L.IL[12:10] = 000       : Disable INT0 interrupt
INTCLR.EICLR[5:0] = 000001  : Clear INT0 request
IMCGA0.INT0EN = 0           : Disable INT0 for wake-up signaling
EICRCG.ICRCG[2:0] = 000     : Clear INT0 request
```

6.2 Interrupt Sources

The TMP1940CYAF provides a reset interrupt, nonmaskable interrupts, and maskable interrupts:

- Reset and nonmaskable interrupts

  The $\overline{\text{RESET}}$ pin causes a Reset interrupt. The $\overline{\text{NMI}}$ pin functions as a nonmaskable interrupt. The on-chip Watchdog Timer (WDT) is also capable of being a source of a nonmaskable interrupt (INTWDT). Reset and nonmaskable interrupts are always vectored to virtual address 0xBFC0_0000.

- Maskable interrupts

  The TMP1940CYAF supports two types of maskable interrupts: software and hardware interrupts. Maskable interrupts are vectored to virtual addresses 0xBFC0_0210 through 0xBFC0_0260, as shown below.

Interrupt Source			Virtual Vector Address
Reset			0xBFC0_0000
Nonmaskable			
Maskable	Software	Swi0	0xBFC0_0210
		Swi1	0xBFC0_0220
		Swi2	0xBFC0_0230
		Swi3	0xBFC0_0240
	Hardware		0xBFC0_0260

Note 1: The above table shows the vector addresses when the BEV bit in the CP0 Status register is set to 1. When BEV=1, all exception vectors reside in the on-chip ROM space.

Note 2: Software interrupts are posted by setting one of the Sw[3:0] bits in the CP0 Cause register. Software interrupts are distinct from the "Software Set" interrupt which is one of the hardware interrupt sources. A Software Set interrupt is posted from the INTC to the TX19 core processor when the IL0[2:0] field in the INTC's IMC0 register is set to a non-zero value.

Table 6.1 Hardware Interrupt Sources

Interrupt Number	IVR[9:0]	Interrupt Source	Interrupt Control Register	Address
0	000	Software Set	IMC0L	0xFFFF_E000
1	010	INT0 pin		
2	020	INT1 pin	IMC0H	0xFFFF_E002
3	030	INT2 pin		
4	040	INT3 pin	IMC1L	0xFFFF_E004
5	050	INT4 pin		
6	060	Reserved	IMC1H	0xFFFF_E006
7	070	Reserved		
8	080	Reserved	IMC2L	0xFFFF_E008
9	090	Reserved		
10	0A0	INT5 pin	IMC2H	0xFFFF_E00A
11	0B0	INT6 pin		
12	0C0	INT7 pin	IMC3L	0xFFFF_E00C
13	0D0	INT8 pin		
14	0E0	INT9 pin	IMC3H	0xFFFF_E00E
15	0F0	INTA pin		
16	100	Reserved	IMC4L	0xFFFF_E010
17	110	Reserved		
18	120	Reserved	IMC4H	0xFFFF_E012
19	130	Reserved		
20	140	INTTA0: 8-Bit Timer 0	IMC5L	0xFFFF_E014
21	150	INTTA1: 8-Bit Timer 1		
22	160	INTTA2: 8-Bit Timer 2	IMC5H	0xFFFF_E016
23	170	INTTA3: 8-Bit Timer 3		
24	180	Reserved	IMC6L	0xFFFF_E018
25	190	Reserved		
26	1A0	Reserved	IMC6H	0xFFFF_E01A
27	1B0	Reserved		
28	1C0	INTTB00: 16-Bit Timer 0 (TB0RG0)	IMC7L	0xFFFF_E01C
29	1D0	INTTB01: 16-bit Timer 0 (TB0RG1)		
30	1E0	INTTB10: 16-bit Timer 1 (TB1RG0)	IMC7H	0xFFFF_E01E
31	1F0	INTTB11: 16-bit Timer 1 (TB1RG1)		
32	200	INTTB20: 16-bit Timer 2 (TB2RG0)	IMC8L	0xFFFF_E020
33	210	INTTB21: 16-bit Timer 2 (TB2RG1)		
34	220	INTTB30: 16-bit Timer 3 (TB3RG0)	IMC8H	0xFFFF_E022
35	230	INTTB31: 16-bit Timer 3 (TB3RG1)		
36	240	Reserved	IMC9L	0xFFFF_E024
37	250	Reserved		
38	260	Reserved	IMC9H	0xFFFF_E026
39	270	Reserved		
40	280	INTTBOF0: 16-Bit Timer 0 (Overflow)	IMCAL	0xFFFF_E028
41	290	INTTBOF1: 16-Bit Timer 1 (Overflow)		
42	2A0	INTTBOF2: 16-Bit Timer 2 (Overflow)	IMCAH	0xFFFF_E02A
43	2B0	INTTBOF3: 16-Bit Timer 3 (Overflow)		
44	2C0	Reserved	IMCBL	0xFFFF_E02C
45	2D0	Reserved		
46	2E0	Reserved	IMCBH	0xFFFF_E02E
47	2F0	Reserved		
48	300	INTRX0: SIO receive (Channel 0)	IMCCL	0xFFFF_E030
49	310	INTTX0: SIO transmit (Channel 0)		
50	320	INTRX1: SIO receive (Channel 1)	IMCCH	0xFFFF_E032
51	330	INTTX1: SIO transmit (Channel 1)		
52	340	INTS2: Serial Bus Interface (SBI)	IMCDL	0xFFFF_E034
53	350	Reserved		

Interrupt Number	IVR[9:0]	Interrupt Source	Interrupt Control Register	Address
54	360	INTRX3: SIO receive (Channel 3)	IMCDH	0xFFFF_E036
55	370	INTTX3: SIO transmit (Channel 3)		
56	380	INTRX4: SIO receive (Channel 4)	IMCEL	0xFFFF_E038
57	390	INTTX4: SIO transmit (Channel 4)		
58	3A0	INTRTC: RTC	IMCEH	0xFFFF_E03A
59	3B0	INTAD: A/D conversion complete		
60	3C0	INTDMA0: DMA complete (Channel 0)	IMCFL	0xFFFF_E03C
61	3D0	INTDMA1: DMA complete (Channel 1)		
62	3E0	INTDMA2: DMA complete (Channel 2)	IMCFH	0xFFFF_E03E
63	3F0	INTDMA3: DMA complete (Channel 3)		

6.3 Interrupt Detection

When enabled as a STOP/SLEEP wake-up signal, the polarities of INT0–INT4 are programmed in the EMCGxx field of the IMCGxx register within the CG; in this case, the EIMxx field of the IMCx register within the INTC has no effect; it must be set to “high-level sensitive,” though. When disabled as a wake-up singnal, the polarities of INT0–INT4 are programmed in the EIMxx field in the INTC’s IMCx register. The polarity of INTRTC is always programmed in both the CG and the INTC. All other interrupts are always programmed in the INTC’s IMCx register.

Each interrupt source is individually configurable as negative or positive polarity, and as edge-triggered or level-sensitive. When a selected transition is detected, an interrupt request is issued to the INTC (except for the NMI and INTWDT interrupts, which are directly delivered to the TX19 core processor).

It is the responsibility of software (an interrupt handler routine) to determine the cause of an interrupt and to clear the interrupt condition. INTRTC and INT0–INT4 used for STOP/SLEEP wake-up signaling require software access to two registers: the EICRCG register in the CG and the INTCLR register in the INTC. Other interrupts can be cleared by writing its IVR[9:4] value to the INTCLR register located within the INTC. For an external interrupt configured as level-sensitive, software must explicitly address the device in question and clear the interrupt condition. A level-sensitive interrupt signal must be held active until the TX19 core processor reads its interrupt vector from the Interrupt Vector Register (IVR).

6.4 Resolving Interrupt Priority

(1) Seven Interrupt Priority Levels

The Interrupt Mode Control registers (IMCF–IMC0) contain a 3-bit interrupt priority level (ILx) field for each interrupt source, which ranges from level 0 to level 7, with level 7 being the highest priority. Level 0 indicates that the interrupt is disabled.

(2) Interrupt Level Notification

When an interrupt event occurs, the INTC sends its priority level to the TX19 core processor. The processor can determine the priority level of an interrupt being requested by reading the IL field in the CP0 Cause register.

(3) Interrupt Vector (Interrupt Source Notification)

Whenever an interrupt request is made, the INTC automatically sets its vector in the IVR. The TX19 core processor can determine the exact cause of an interrupt by reading the IVR. If multiple interrupt requests occur at the same level, the interrupt with the smallest interrupt number is delivered (see Table 6.1). When no interrupt is pending, the IVR[9:4] field in the IVR contains a value of zero.

When the TX19 core processor responds to a request with an interrupt acknowledge cycle, the INTC forwards the interrupt vector for that interrupt request. At this time, the TX19 core processor saves the priority level value in the CMask field of the CP0 Status register.

6.5 Register Description

Table 6.2 INTC Register Map

Address	Symbol	Register Name	Corresponding Interrupt Number
0xFFFF_E060	INTCLR	Interrupt Request Clear Register	All (63 – 0)
0xFFFF_E040	IVR	Interrupt Vector Register	All (63 – 0)
0xFFFF_E03C	IMCF	Interrupt Mode Control Register F	63 – 60
0xFFFF_E038	IMCE	Interrupt Mode Control Register E	59 – 56
0xFFFF_E034	IMCD	Interrupt Mode Control Register D	55 – 52
0xFFFF_E030	IMCC	Interrupt Mode Control Register C	51 – 48
0xFFFF_E02C	IMCB	Interrupt Mode Control Register B	47 – 44
0xFFFF_E028	IMCA	Interrupt Mode Control Register A	43 – 40
0xFFFF_E024	IMC9	Interrupt Mode Control Register 9	39 – 36
0xFFFF_E020	IMC8	Interrupt Mode Control Register 8	35 – 32
0xFFFF_E01C	IMC7	Interrupt Mode Control Register 7	31 – 28
0xFFFF_E018	IMC6	Interrupt Mode Control Register 6	27 – 24
0xFFFF_E014	IMC5	Interrupt Mode Control Register 5	23 – 20
0xFFFF_E010	IMC4	Interrupt Mode Control Register 4	19 – 16
0xFFFF_E00C	IMC3	Interrupt Mode Control Register 3	15 – 12
0xFFFF_E008	IMC2	Interrupt Mode Control Register 2	11 – 8
0xFFFF_E004	IMC1	Interrupt Mode Control Register 1	7 – 4
0xFFFF_E000	IMC0	Interrupt Mode Control Register 0	3 – 0

6.5.1 Interrupt Vector Register (IVR)

This register indicates the vector for the interrupt source when there is an interrupt event.

IVR (0xFFFF_E040)

	7	6	5	4	3	2	1	0
Name	IVRL				—	—	—	—
Read/Write	R							
Reset Value	0	0	0	0	0	0	0	0
Function	Interrupt vector for the source of the current interrupt							
	15	14	13	12	11	10	9	8
Name	IVRH						IVRL	
Read/Write	R/W						R	
Reset Value	0	0	0	0	0	0	0	0
Function							Interrupt vector for the source of the current interrupt	
	23	22	21	20	19	18	17	16
Name	IVRM							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Name	IVRM							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function								

6.5.2 Interrupt Mode Control Registers (IMCF–IMC0)

These registers control the interrupt priority level, active polarity, either level or edge sensitivity, and DMA triggering.

IMC0L (0xFFFF_E000)

	7	6	5	4	3	2	1	0
Name	—	—	EIM01	EIM00	DM0	IL02	IL01	IL00
Read/Write	—	—	R/W					
Reset Value	—	—	0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level Must be set to 00.		DMA trigger 0: Disable 1: Enable	When DM0 = 0 Interrupt Number 0 (Software Set) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM0 = 1 DMAC channel select 000–011: Channel number (0–3) 100–111: Don't use.		
	15	14	13	12	11	10	9	8
Name	—	—	EIM11	EIM10	DM1	IL12	IL11	IL10
Read/Write	—	—	R/W					
Reset Value	—	—	0	0	0	0	0	0
Function			Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM0 = 0 Interrupt Number 1 (INT0 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM0 = 1 DMAC channel select 000–011: Channel number (0–3) 100–111: Don't use.		

IMC0H (0xFFFF_E002)

	23	22	21	20	19	18	17	16
Name	—	—	EIM21	EIM20	DM2	IL22	IL21	IL20
Read/Write	—	—	R/W					
Reset Value	—	—	0	0	0	0	0	0
Function			Same as above (INT1)		Same as above (INT1)	Interrupt Number 2 (INT1 pin) Same as above		
	31	30	29	28	27	26	25	24
Name	—	—	EIM31	EIM30	DM3	IL32	IL31	IL30
Read/Write	—	—	R/W					
Reset Value	—	—	0	0	0	0	0	0
Function			Same as above (INT2)		Same as above (INT2)	Interrupt Number 3 (INT2 pin) Same as above		

Note 1: Interrupt sensitivity must be programmed when interrupts are enabled.
Note 2: For a complete list of the Interrupt Mode Control registers, see Chapter 19.
Note 3: When an interrupt is used to trigger a DMAC channel, that DMAC channel must be put in Ready state after the programming of the INTC.

6.5.3 Interrupt Request Clear Register (INTCLR)

Loading the EICLR[5:0] field of this register with the IVRL[9:4] value of the IVR causes the corresponding interrupt to be cleared.

INTCLR 0xFFFF_E060)

	7	6	5	4	3	2	1	0
Name	—	—	EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
Read/Write	—	—	W					
Reset Value	—	—	—	—	—	—	—	—
Function	IVRL[9:4] value for an interrupt to be cleared							

Note1: An interrupt request must not be cleared before the TX19 core processor reads the IVR value.
Note2: Follow the steps below to disable a particular interrupt with the Interrupt Controller (INTC).

1. **Globally disable the acceptance of interrupts by the core processor by clearing the IEc bit of the Status register.**
2. **Disable a desired interrupt with the INTC by clearing the ILx[2:0] field of the IMCxx register.**
3. **Execute the SYNC instruction.**
4. **Enable the acceptance of interrupts by the core processor by setting the IEc bit of the Status register.**

```
Example:  mtc0   r0,   r31    ;  _DI ( ) ;
          sb     r0,   IMC**  ;  IMC** = 0 ;
          sync                ;  _SYNC ( ) ;
          mtc0   $sp,  r31    ;  _EI ( ) ;
```

7. I/O Ports

The TMP1940CYAF has 77 I/O port pins. All the port pins except a few share pins with alternate functions. They can be individually programmed as general-purpose I/O or dedicated I/O for the on-chip CPU or peripherals. Table 7.1 shows all the I/O port pins available on the TMP1940CYAF and their shared functions. (There is no Port 6.) Table 7.2 is a summary of register settings used to control the port pins.

Table 7.1 Programmable I/O Ports

Port	Pin Name	# of Pins	Direction	Pull Resistor	Direction Programmability	Alternate Functions
Port 0	P00–P07	8	Input/output	—	Bitwise	AD0–AD7
Port 1	P10–P17	8	Input/output	—	Bitwise	AD8–AD15/A8–A15
Port 2	P20–P27	8	Input/output	—	Bitwise	A0–A7/A16–A23
Port 3	P30	1	Output	—	Fixed	$\overline{RD}$
	P31	1	Output	—	Fixed	$\overline{WR}$
	P32	1	Input/output	Pullup	Bitwise	$\overline{HWR}$
	P33	1	Input/output	Pullup	Bitwise	$\overline{WAIT}$
	P34	1	Input/output	Pullup	Bitwise	$\overline{BUSRQ}$
	P35	1	Input/output	Pullup	Bitwise	$\overline{BUSAK}$
	P36	1	Input/output	Pullup	Bitwise	R/$\overline{W}$
	P37	1	Input/output	Pullup	Bitwise	
Port 4	P40	1	Input/output	Pullup	Bitwise	$\overline{CS0}$
	P41	1	Input/output	Pullup	Bitwise	$\overline{CS1}$
	P42	1	Input/output	Pullup	Bitwise	$\overline{CS2}$
	P43	1	Input/output	Pullup	Bitwise	$\overline{CS3}$
	P44	1	Input/output	—	Bitwise	SCOUT
Port 5	P50–P57	8	Input	—	Fixed	AN0–AN7/ $\overline{ADTRG}$ (P53)
Port 7	P70	1	Input/output	—	Bitwise	TA0IN/TXD3
	P71	1	Input/output	—	Bitwise	TA1OUT/RXD3
	P72	1	Input/output	—	Bitwise	TA2IN/TXD4
	P73	1	Input/output	—	Bitwise	TA3OUT/RXD4
	P74	1	Input/output	—	Bitwise	TB0IN0/INT5
	P75	1	Input/output	—	Bitwise	TB0IN1/INT6
	P76	1	Input/output	—	Bitwise	TB0OUT
	P77	1	Input/output	—	Bitwise	INT0
Port 8	P80	1	Input/output	—	Bitwise	TB1IN0/INT7
	P81	1	Input/output	—	Bitwise	TB1IN1/INT8
	P82	1	Input/output	—	Bitwise	TB1OUT
	P83	1	Input/output	—	Bitwise	TB2IN0INT9
	P84	1	Input/output	—	Bitwise	TB2IN1/INTA
	P85	1	Input/output	—	Bitwise	TB2OUT (/ $\overline{BOOT}$ in TMP1940FDBF)
	P86	1	Input/output	—	Bitwise	TB3OUT/INTLV
	P87	1	Input/output	—	Bitwise	
Port 9	P90	1	Input/output	—	Bitwise	TXD0
	P91	1	Input/output	—	Bitwise	RXD0
	P92	1	Input/output	—	Bitwise	SCLK0/ $\overline{CTS0}$
	P93	1	Input/output	—	Bitwise	TXD1
	P94	1	Input/output	—	Bitwise	RXD1
	P95	1	Input/output	—	Bitwise	SCLK1/ $\overline{CTS1}$
	P96	1	Input/output	—	Bitwise	XT1
	P97	1	Input/output	—	Bitwise	XT2
Port A	PA0–PA3	4	Input/output	—	Bitwise	INT1–INT4
	PA4	1	Input/output	—	Bitwise	
	PA5	1	Input/output	—	Bitwise	SCK
	PA6	1	Input/output	—	Bitwise	SO/SDA
	PA7	1	Input/output	—	Bitwise	SI/SCL

Table 7.2 I/O Port Programmability (1/2)

Port	Pin Name	Direction / Function	I/O Register Settings		
			Pn	PnCR	PnFC
Port 0	P00–P07	Input port	X	0	N/A
		Output port	X	1	
		AD0–AD7 bus lines	X	X	
Port 1	P10–P17	Input port	X	0	0
		Output port	X	1	0
		AD8–AD15 bus lines	X	0	1
		A8–A15 outputs	X	1	1
Port 2	P20–P27	Input port	X	0	0
		Output port	X	1	0
		A0–A7 outputs	X	0	1
		A16–A23 outputs	X	1	1
Port 3	P30	Output port	X	N/A	0
		$\overline{RD}$ output during external accesses	X		1
	P31	Output port	X	N/A	0
		$\overline{WR}$ output during external accesses	X		1
	P32–P37	Input port (with pullup disabled)	0	0	0
		Input port (with pullup enabled)	1	0	0
		Output port	X	1	0
	P32 (Note 1)	$\overline{HWR}$ output	X	1	1
	P33	$\overline{WAIT}$ input (with pullup disabled)	0	0	N/A
		$\overline{WAIT}$ input (with pullup enabled)	1	0	
	P34	$\overline{BUSRQ}$ input (with pullup disabled)	0	0	1
		$\overline{BUSRQ}$ input (with pullup enabled)	1	0	1
	P35	$\overline{BUSAK}$ output	X	1	1
	P36 (Note1)	R/$\overline{W}$ output	X	1	1
Port 4	P40–P43 (Note 1)	Input port (with pullup disabled)	0	0	0
		Input port (with pullup enabled)	1	0	0
		Output port	X	1	0
	P40	$\overline{CS0}$ output	X	1	1
	P41	$\overline{CS1}$ output	X	1	1
	P42	$\overline{CS2}$ output	X	1	1
	P43	$\overline{CS3}$ output	X	1	1
	P44	SCOUT output	X	1	1
Port 5	P50–P57	Input port	X	N/A	
		AN[0:7] inputs (Note 2)	X		
	P53	$\overline{ADTRG}$ input (Note 3)	X		
Port 7	P70–P77	Input port	X	0	0
		Output port	X	1	0
	P70	TA0IN input	X	0	1
		TXD3 output	X	1	1
	P71	TA1OUT output	X	1	1
		RXD3 input	X	0	1
	P72	TA2IN input	X	0	1
		TXD4 output	X	1	1
	P73	TA3OUT output	X	1	1
		RXD4 input	X	0	1
	P74	TB0IN0 input	X	0	1
		INT5 input	X	0	Setting unneeded
	P75	TB0IN1 input	X	0	1
		INT6 input	X	0	Setting unneeded
	P76	TB0OUT output	X	1	1
	P77	Wake-up INT0 input (Note 4)	X	0	1
		INT0 input (no wake-up)	X	0	Setting unneeded

Table 7.2 I/O Port Programmability (2/2)

Port	Pin Name	Function / Direction	I/O Register Settings		
			Pn	PnCR	PnFC
Port 8	P80–P87	Input port	X	0	0
		Output port	X	1	0
	P80	TB1IN0 input	X	0	1
		INT7 input	X	0	Setting unneeded
	P81	TB1IN1 input	X	0	1
		INT8 input	X	0	Setting unneeded
	P82	TB1OUT output	X	1	1
	P83	TB2IN0 input	X	0	1
		INT9 input	X	0	Setting unneeded
	P84	TB2IN1 input	X	0	1
		INTA input	X	0	Setting unneeded
	P85	TB2OUT output	X	1	1
	P86	TB3OUT output	X	1	1
Port 9	P90–P95	Input port	X	0	0
		Output port	X	1	0
	P90	TXD0 output	X	1	1
	P91	RXD0 input	X	0	N/A
	P92	SCLK0 output	X	1	1
		$\overline{\text{CTS0}}$ /SCLK0 input	X	0	1
	P93	TXD1 output	X	1	1
	P94	RXD1 input	X	0	N/A
	P95	SCLK1 output	X	1	1
		$\overline{\text{CTS1}}$/SCLK1 input	X	0	1
	P96–P97	Input port	X	0	N/A
		Output port (Note 5)	X	1	
		XT1–XT2 (Note 6)	X	0	
Port A	PA0–PA7	Input port	X	0	0
		Output port	X	1	0
	PA0–PA3	Wake-up INT1–INT4 inputs (Note 4)	X	0	Setting unneeded
		INT1–INT4 inputs (no wake-up)	X	0	
	PA5	SCK input	X	0	1
		SCK output	X	1	1
	PA6	SDA input	X	0	0
		SDA output (Note 5)/SO output	X	1	1
	PA7	SI input/SCL input	X	0	0
		SCL output (Note 7)	X	1	1

X: Don't care

Pn: Port n Register, PnCR: Port n Control Register, PnFC: Port n Function Register

Note 1: P32, P36 and P40–P43 have their internal pull-up resistors enabled when the corresponding PxFC register bit is set and when the bus is released.

Note 2: When P50–P57 are configured as analog channels of the ADC, the ADCH[2:0] field in A/D Mode Control Register 1 (ADMOD1) is used to select a channel(s). See Section 15.1.

Note 3: When P53 is configured as $\overline{\text{ADTRG}}$, the ADTRGE bit in the ADMOD1 register is used to enable and disable the external trigger input to the ADC.

Note 4: When INT0–INT4 are enabled for a wake-up from STOP mode with the SYSCR2.DRIVE bit cleared (undriven pins), the corresponding bit in the PnFC must be set.

Note 5: When P96–P97 are configured as output ports, they function as open-drain outputs.

Note 6: When P96–P97 are configured as XT1–XT2, the SYSCR0 register must be programmed to enable oscillation, etc.

Note 7: When PA6 and PA7 are configured as SDA and SCL outputs for the SBI, the ODEA[7:6] field in the Open-Drain Enable (ODE) register can be used to configure them as either push-pull or open-drain ouptuts. Upon reset, the default is push-pull. See Section 7.11.

7.1 Port 0 (P00–P07)

Eight Port 0 pins function as either discrete general-purpose I/O pins or the AD[0:7] bits of the address/data bus. The P0CR register controls the direction of the Port 0 pins. Upon reset, the P0CR register bits are cleared, configuring all Port 0 pins as inputs.

During external memory accesses, Port 0 pins are automatically configured as AD[0:7], with the P0CR register bits all cleared.

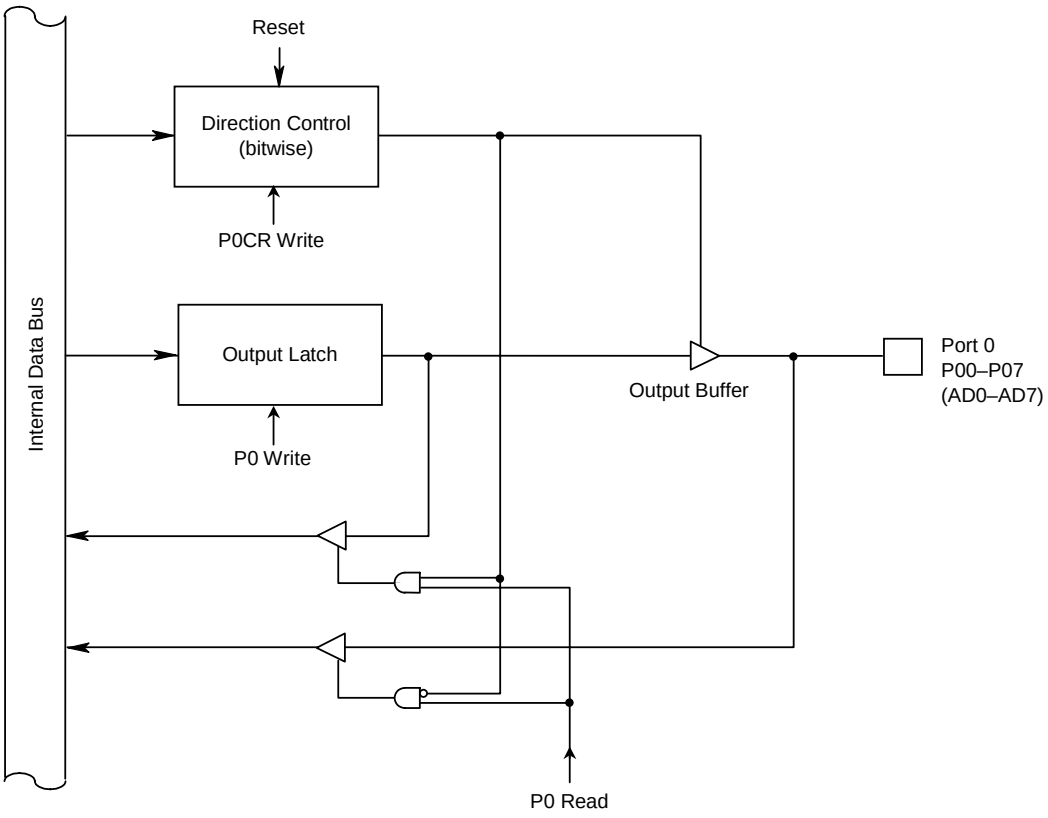


Figure 7.1 Port 0 (P00–P07)

Port 0 Register

		7	6	5	4	3	2	1	0
P0	Name	P07	P06	P05	P04	P03	P02	P01	P00
(0xFFFF_F000)	Read/Write	R/W							
	Reset Value	Input mode (The Output Latch is cleared to 0.)							

Port 0 Control Register

		7	6	5	4	3	2	1	0
P0CR	Name	P07	P06	P05	P04	P03	P02	P01	P00
(0xFFFF_F002)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	0: IN, 1: OUT (Functions as AD7–AD0 during external memory accesses, with all bits cleared.)							

Port 0 Direction Settings

0	Input
1	Output

Figure 7.2 Port 0 Registers

7.2 Port 1 (P10–P17)

Eight Port 1 pins can be individually programmed to function as discrete general-purpose I/O pins, the AD[8:15] bits of the address/data bus or the A[8:15] bits of the address bus. The P1CR and P1FC registers select the direction and function of the Port 1 pins. Upon reset, the Output Latch (P1) is cleared, and the P1CR and P1FC register bits are cleared to all 0s, configuring all Port 1 pins as input port pins.

For external memory accesses, Port 1 pins must be configured as AD[8:15] or A[8:15].

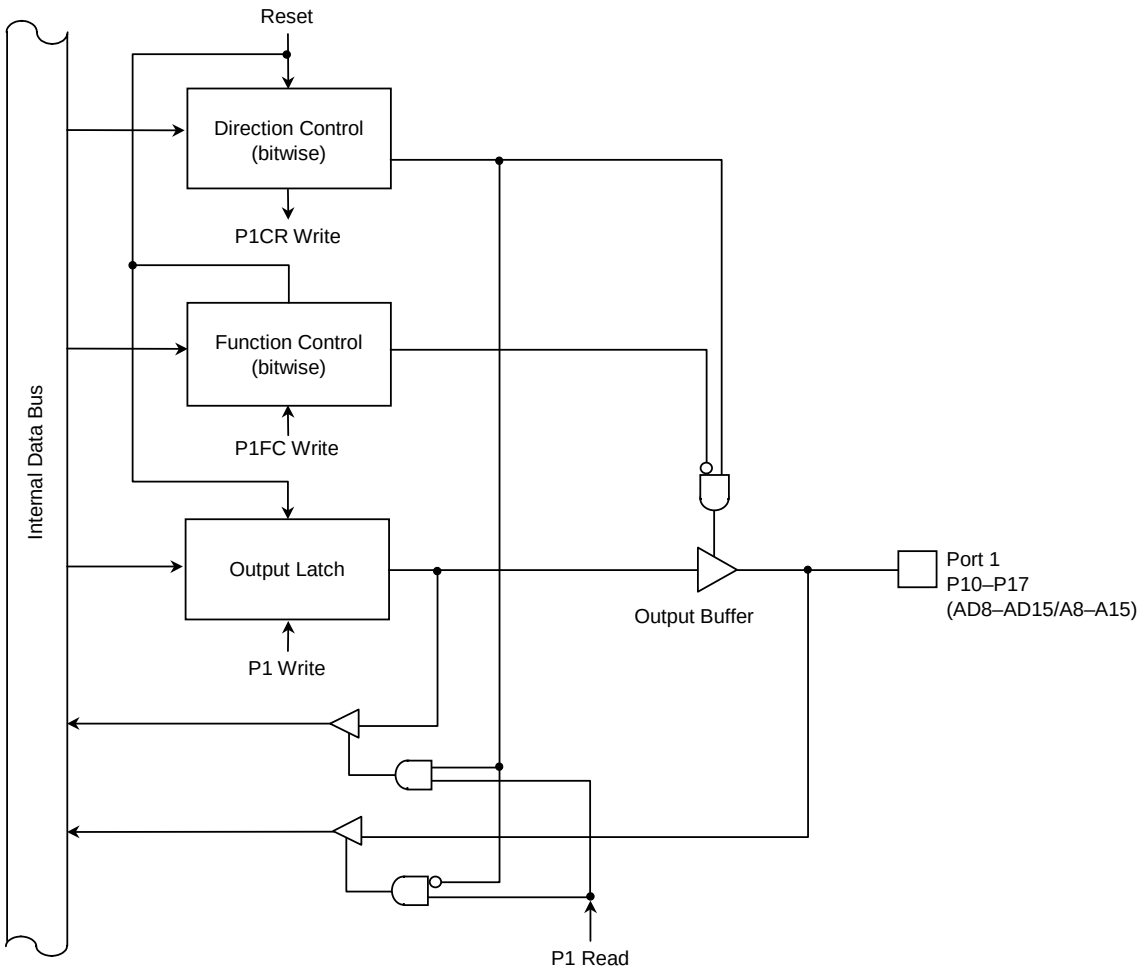


Figure 7.3 Port 1 (P10–P17)

Port 1 Register

P1 (0xFFFF_F001)

	7	6	5	4	3	2	1	0
Name	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
Reset Value	Input mode (The Output Latch is cleared to 0.)							

Port 1 Control Register

P1CR (0xFFFF_F004)

	7	6	5	4	3	2	1	0
Name	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	Refer to P1FC.							

Port 1 Function Register

P1FC (0xFFFF_F005)

	7	6	5	4	3	2	1	0
Name	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00: Input port, 01: Output port, 10: AD15–8, 11: A15–8							

Port 1 Function Settings

P1CR.P1xC	P1FC.P1xF	
	0	1
0	Input port	Address/Data bus (AD15–AD8)
1	Output port	Address bus (A15–A8)

Figure 7.4 Port 1 Registers

7.3 Port 2 (P20–P27)

Eight Port 2 pins can be individually programmed to function as discrete general-purpose I/O pins, the A[0:7] bits of the address bus or the A[16:23] bits of the address bus. The P2CR and P2FC registers select the direction and function of the Port 2 pins. Upon reset, the Output Latch (P2) is set to all 1s, and the P2CR and P2FC register bits are cleared, configuring all Port 2 pins as input port pins.

For external memory accesses, Port 2 pins must be configured as A[0:7] or A[16:23].

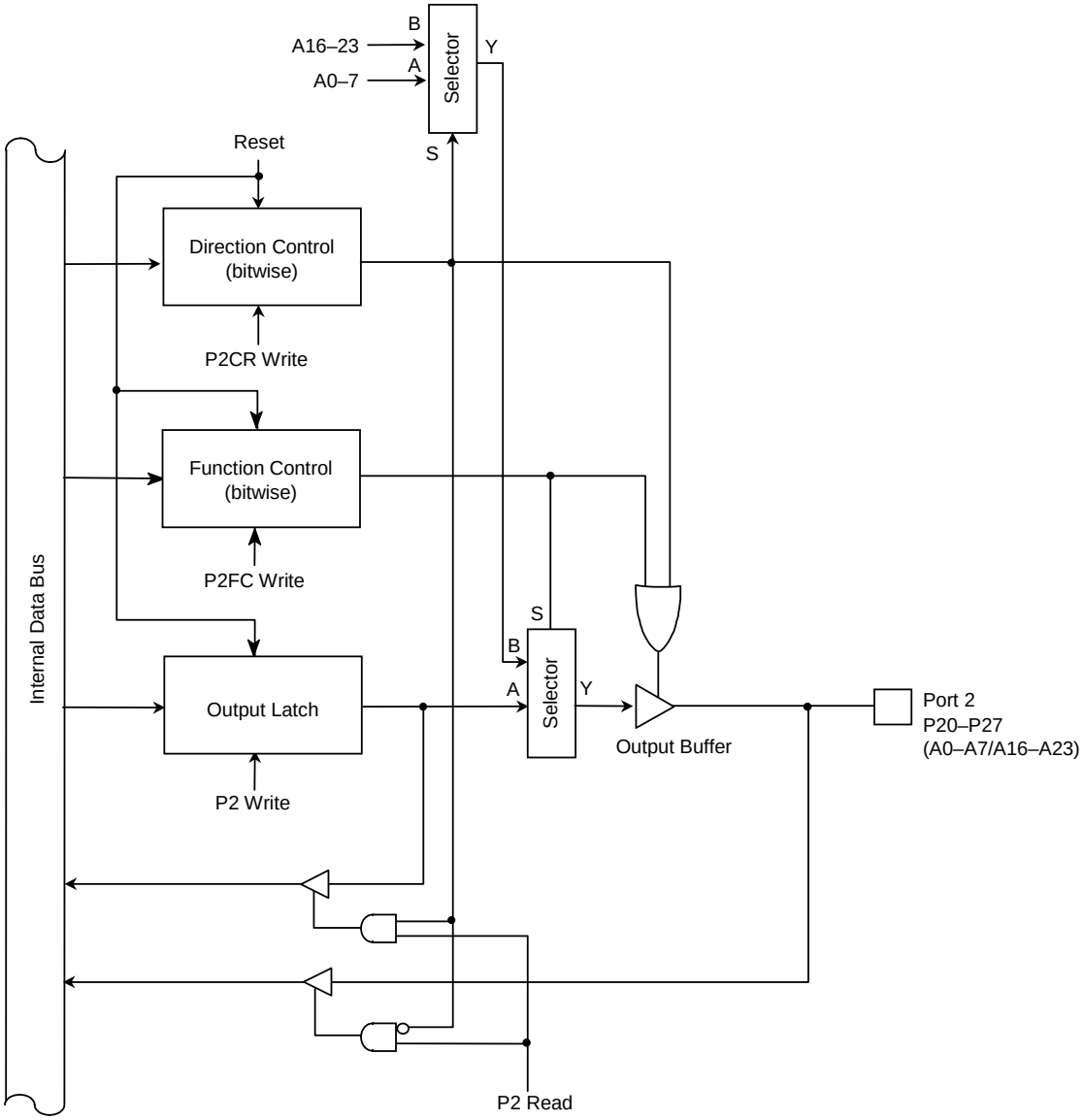


Figure 7.5 Port 2 (P20~P27)

Port 2 Register

		7	6	5	4	3	2	1	0
P2	Name	P27	P26	P25	P24	P23	P22	P21	P20
(0xFFFF_F012)	Read/Write	R/W							
	Reset Value	Input mode (The Output Latch set to 1.)							

Port 2 Control Register

		7	6	5	4	3	2	1	0
P2CR	Name	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
(0xFFFF_F014)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	Refer to P2FC.							

Port 2 Function Register

		7	6	5	4	3	2	1	0
P2FC	Name	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
(0xFFFF_F015)	Read/Write	W							
	Reset Value	0	0	0	0	0	0	0	0
	Function	P2FC/P2CR = 00: Input port, 01: Output port, 10: A7–0, 11: A23–16							

Port 2 Function Settings

P2CR. P2xC	P2FC.P2xF	
	0	1
0	Input port	Address bus (A7–A0)
1	Output port	Address bus (A23–A16)

Figure 7.6 Port 2 Registers

7.4 Port 3 (P30–P37)

Eight Port 3 pins can be individually programmed to function as either discrete general-purpose I/O pins or CPU control/status pins. In either case, P30 and P31 are output-only pins.

The P3CR and P3FC registers select the direction and function of the Port 3 pins. Upon reset, the P3CR and P3FC register bits are cleared, configuring P30 and P31 as output port pins and P32–P37 as input port pins with pullup enabled. (Bits 0 and 1 in the P3CR and bit 3 in the P3FC are unused.) Upon reset, the Output Latch (P3) is set to all 1s; so a logic 1 appears on P30 and P31.

When P30 is configured as $\overline{RD}$ (P3FC.P30F=1), the Read Strobe signal is activated when external address space is accessed. Likewise, when P31 is configured as $\overline{WR}$ (P3FC.P31F=1), the Write Strobe signal is activated when external address space is accessed.

P35 can be configured as $\overline{BUSAK}$. While $\overline{BUSAK}$ is asserted, the internal pullup resistors for P32 and P36 are enabled, if they are configured as $\overline{HWR}$ (P3FC.P32F=1) and R/$\overline{W}$ (P3FC.P36F=1) respectively.

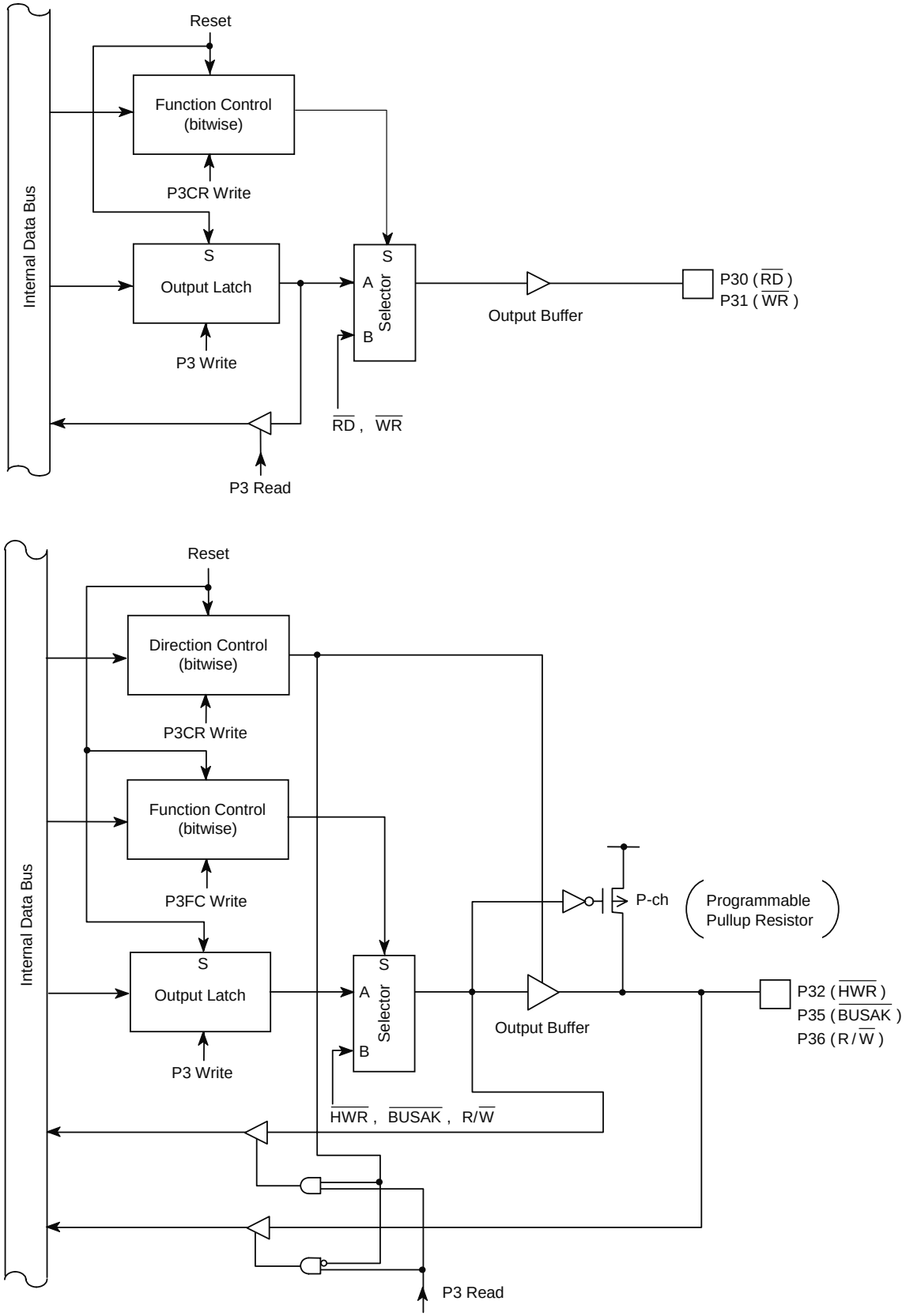


Figure 7.7 Port 3 (P30, P31, P32, P35, P36)

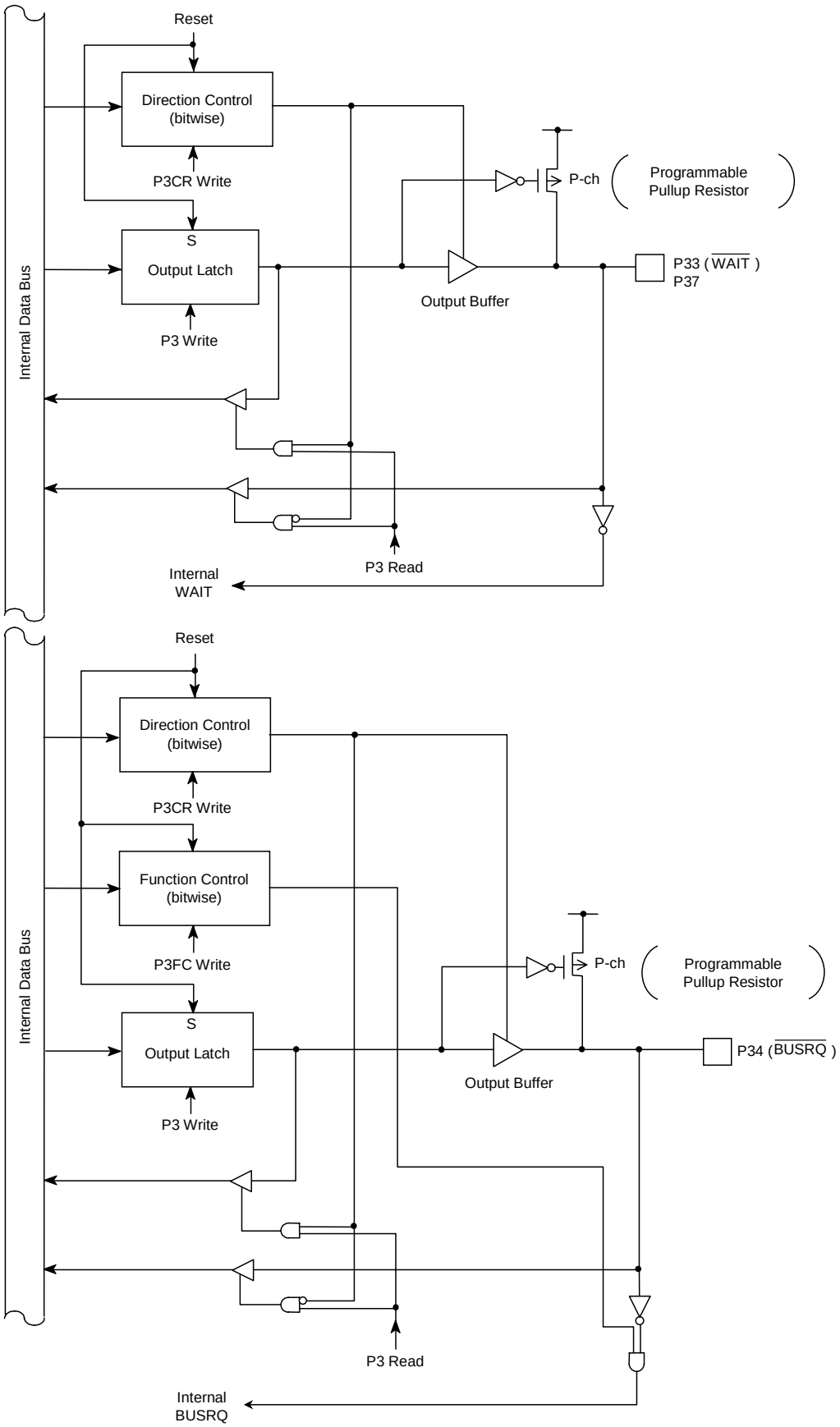


Figure 7.8 Port 3 (P33, P34, P37)

Port 3 Register

P3 (0xFFFF_F018)

	7	6	5	4	3	2	1	0
Name	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W							
Reset Value	Input mode						Output mode	
	1 (Pullup)	1 (Pullup)	1 (Pullup)	1 (Pullup)	1 (Pullup)	1 (Pullup)	1	1

Port 3 Control Register

P3FC (0xFFFF_F01B)

	7	6	5	4	3	2	1	0
Name	P37C	P36C	P35C	P34C	P33C	P32C	—	—
Read/Write	W						—	—
Reset Value	0	0	0	0	0	0	—	—
Function	0: IN 1: OUT							

Port 3 Direction Settings

0	Input
1	Output

Port 3 Function Register

P3FC (0xFFFF_F01B)

	7	6	5	4	3	2	1	0
Name	—	P36F	P35F	P34F	—	P32F	P31F	P30F
Read/Write	W							
Reset Value	—	0	0	0	—	0	0	0
Function		0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$

$\overline{BUSRQ}$ Settings

P3FC.P34F	1
P3CR.P34C	0

$\overline{BUSAK}$ Settings

P3FC.P35F	1
P3CR.P35C	1

R/$\overline{W}$ Settings

P3FC.P36F	1
P3CR.P36C	1

P30 ($\overline{RD}$) Function Settings

P30F	P30	
	0	1
0	Output a 0.	Output a 1.
1	Assert $\overline{RD}$ only during external accesses.	

P31 ($\overline{WR}$) Function Settings

P31F	P31	
	0	1
0	Output a 0.	Output a 1.
1	Asserts $\overline{WR}$ only during external accesses	

$\overline{HWR}$ Settings

P3FC.P32F	1
P3CR.P32C	1

Figure 7.9 Port 3 Registers

7.5 Port 4 (P40–P44)

P40–P43 can be individually programmed to function as either discrete general-purpose I/O pins or programmable chip select ($\overline{CS0}$ – $\overline{CS3}$) pins. P44 can be programmed to function as either a general-purpose I/O pin or a system clock output (SCOUT) pin.

The P4CR and P4FC registers select the direction and function of the Port 4 pins. Upon reset, the P4CR and P4FC register bits are cleared, configuring all the Port 4 pins as input port pins; P40–P43 have an internal pullup resistor. Upon reset, the Output Latch (P4) is set to all 1s.

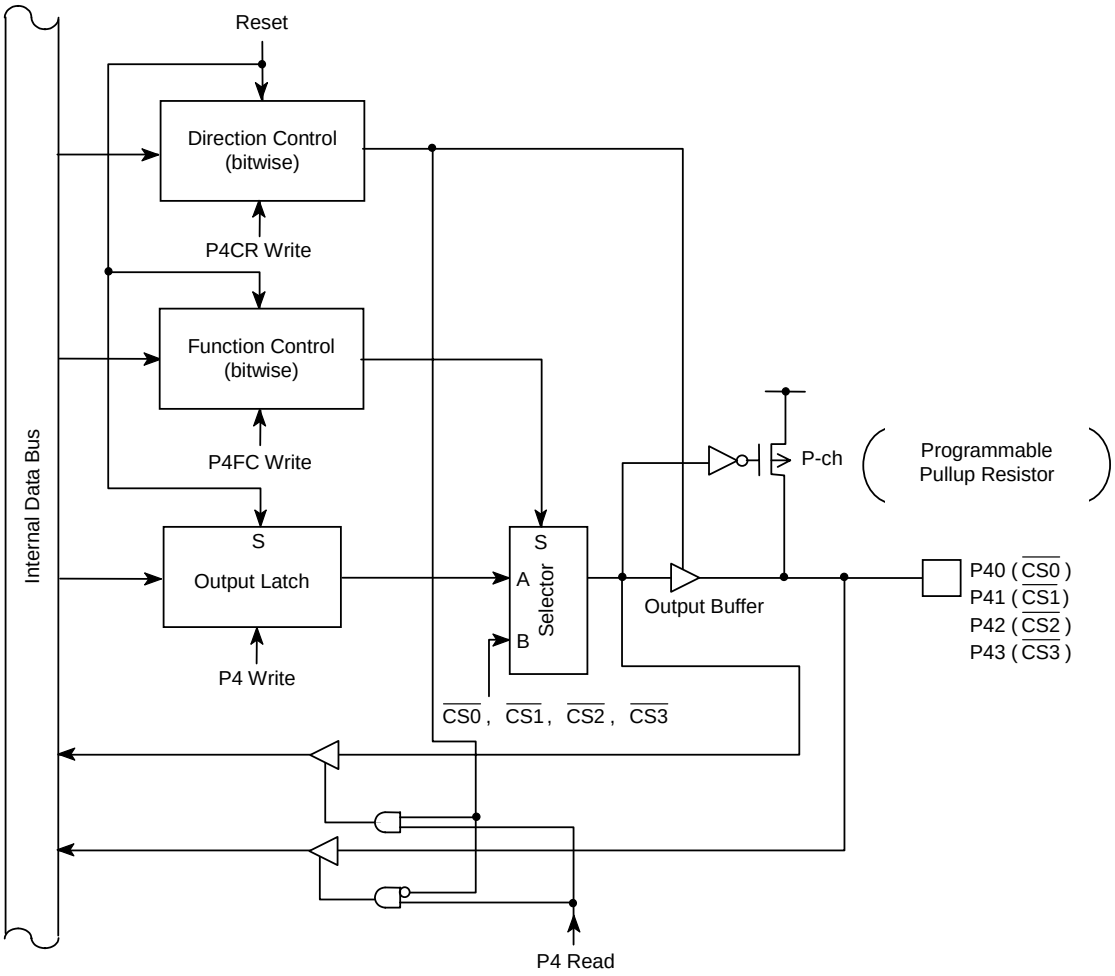


Figure 7.10 Port 4 (P40–P43)

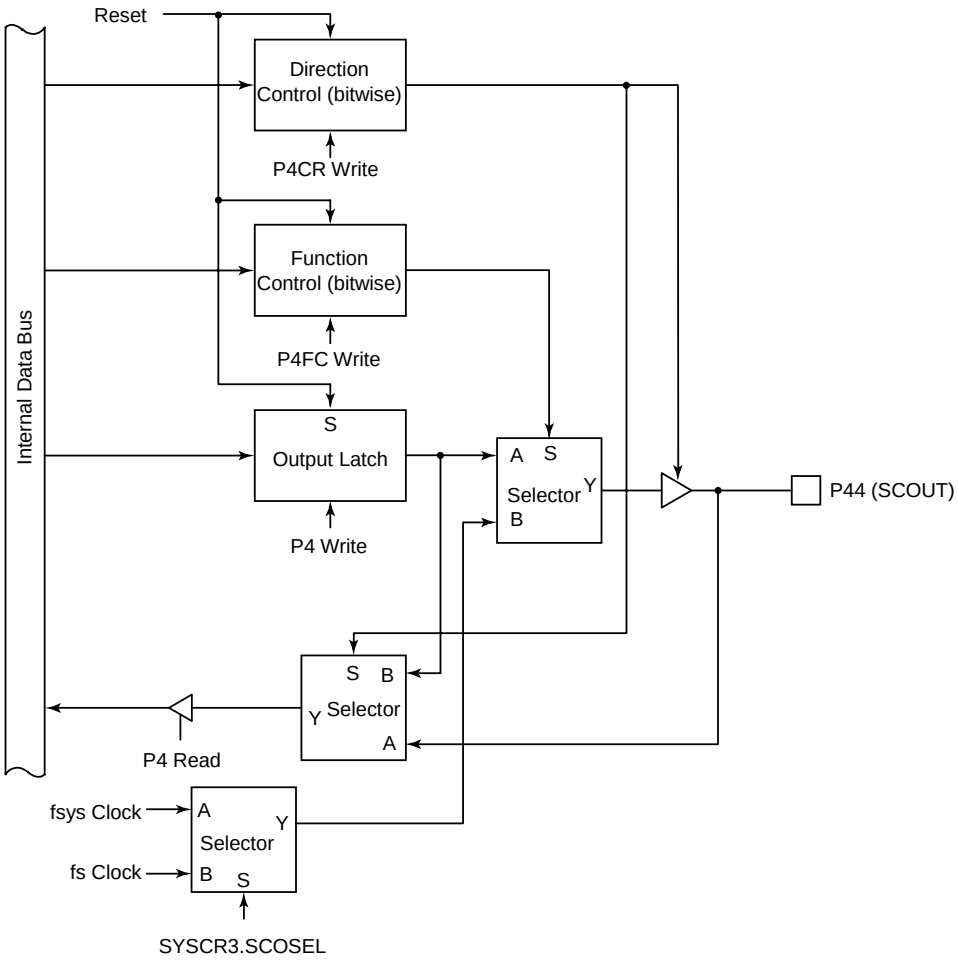


Figure 7.11 Port 4 (P44)

Port 4 Register

		7	6	5	4	3	2	1	0
P4	Name	—	—	—	P44	P43	P42	P41	P40
(0xFFFF_F01E)	Read/Write	—	—	—	R/W				
	Reset Value	—	—	—	Input mode				
		—	—	—	1	1 (Pullup)	1 (Pullup)	1 (Pullup)	1 (Pullup)

Port 4 Control Register

		7	6	5	4	3	2	1	0
P4CR	Name	—	—	—	P44C	P43C	P42C	P41C	P40C
(0xFFFF_F020)	Read/Write	—	—	—	W				
	Reset Value	—	—	—	0	0	0	0	0
		—	—	—	0: IN			1: OUT	

Port 4 Function Register

		7	6	5	4	3	2	1	0
P4FC	Name	—	—	—	P44F	P43F	P42F	P41F	P40F
(0xFFFF_F021)	Read/Write	—	—	—	W				
	Reset Value	—	—	—	0	0	0	0	0
	Function				0: Port 1: SCOUT		0: Port 1: $\overline{CS}$		

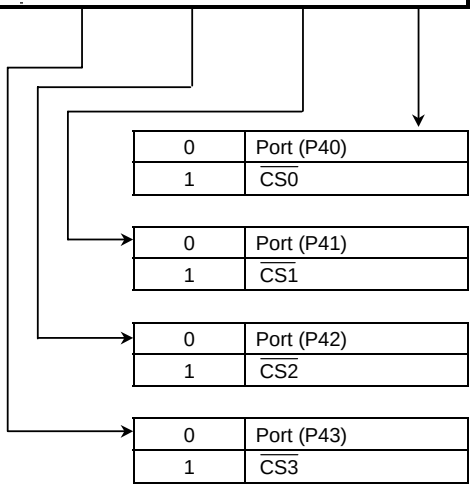


Figure 7.12 Port 4 Registers

7.6 Port 5 (P50–P57)

Eight Port 5 pins are input-only pins shared with the analog input pins of the A/D Converter (ADC). P53 is also shared with the A/D trigger input pin.

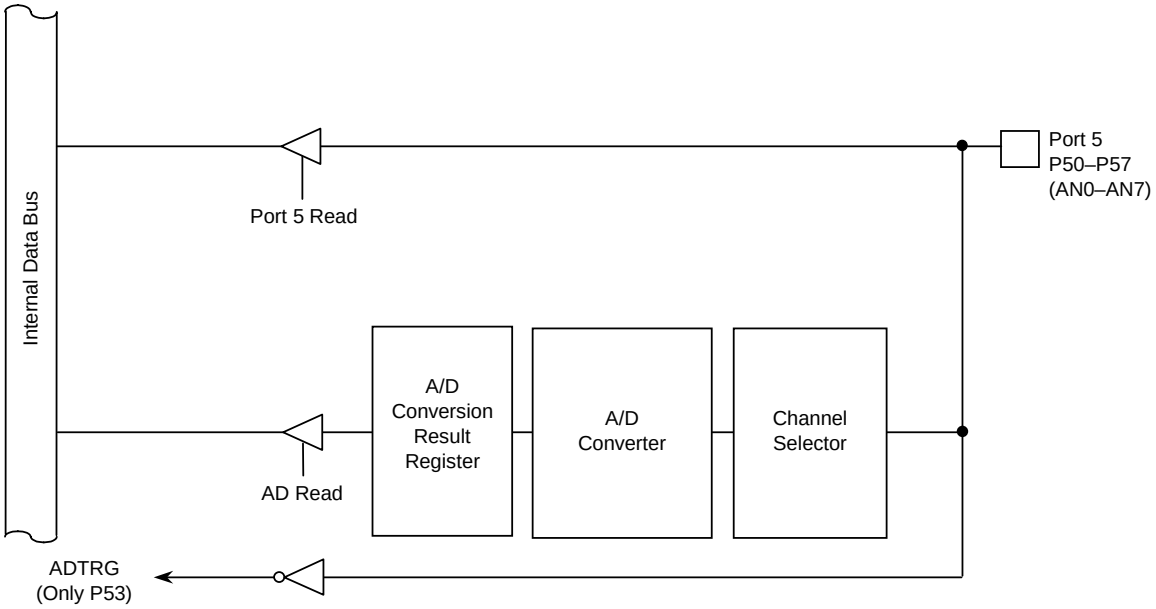


Figure 7.13 Port 5 (P50–P57)

Port 5 Register

P5 (0xFFFF_F025)

	7	6	5	4	3	2	1	0
Bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
Read/Write	R							
After reset	Input mode							

Figure 7.14 Port 5 Register

Note 1: A/D Mode Control Register 1 (ADMOD1) is used to select an A/D converter input channel(s) and to enable the A/D trigger input. See Section 15.1.

Note 2: When P53 is used as the A/D trigger Input ($\overline{\text{ADTRG}}$) pin, P53 (AN3) can not function as an analog input.

7.7 Port 7 (P70–P77)

Eight Port 7 pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. Upon reset, all Port 7 pins are configured as input port pins. Alternatively, P70 and P72 can each be programmed as either the TXD output from an SIO channel or the clock input (TA0IN or TA2IN) to an 8-bit timer. P71 and P73 can each be programmed as either the RXD input to an SIO channel or the timer output (TA1OUT or TA3OUT) from an 8-bit timer. P74 and P75 can each be programmed as either the clock input (TB0IN0 or TB0IN1) to a 16-bit timer or an external interrupt request pin (INT5 or INT6). P76 can be programmed as the timer flip-flop output (TB0OUT) from a 16-bit timer. P77 can be programmed as an external interrupt request pin (INT0).

The P7CR and P7FC registers select the direction and function of the Port 7 pins. A reset sets the Output Latch (P7) to all 1s, and clears the P7CR and P7FC register bits, configuring all Port 7 pins as input port pins. When INT0 is used as a wake-up from STOP mode with the SYSCR2.DRVE bit cleared, the P7FC.P77F bit must be set to 1.

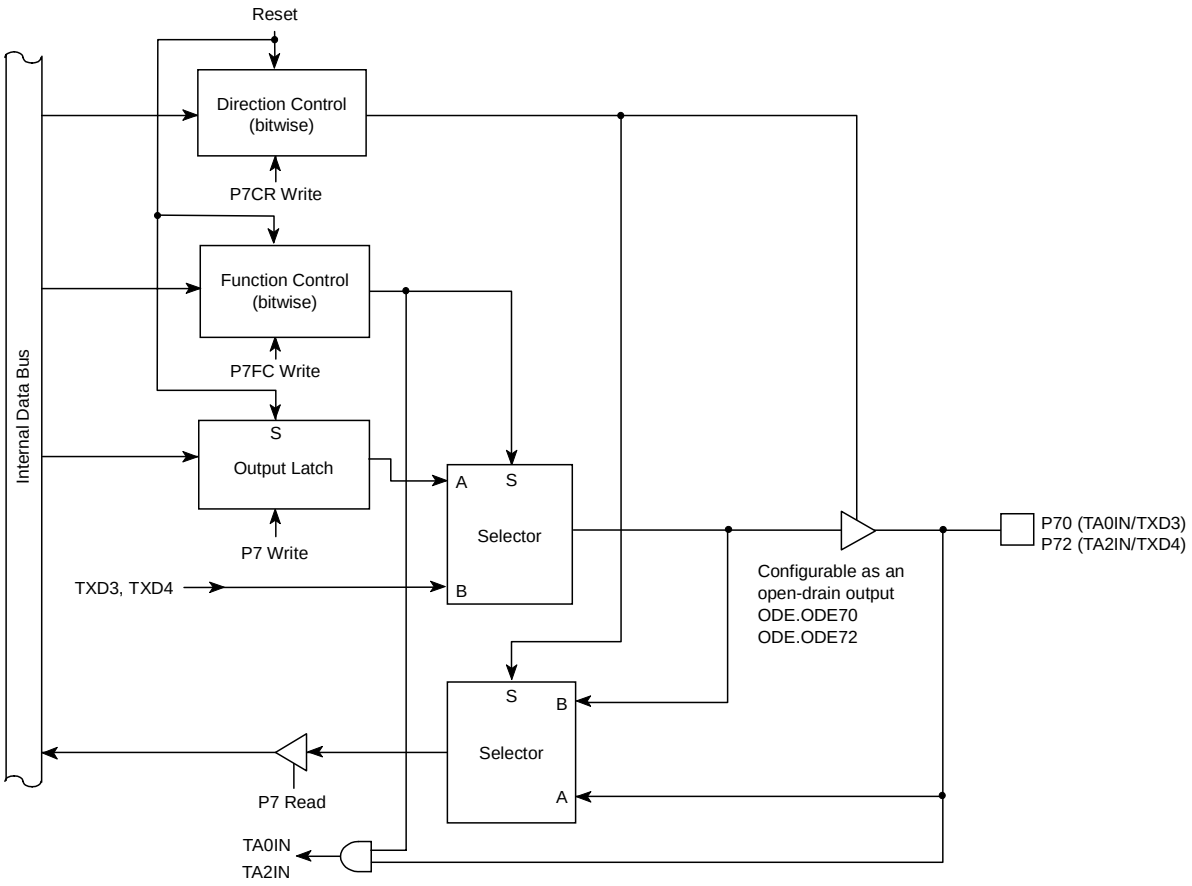


Figure 7.15 Port 7 (P70, P72)

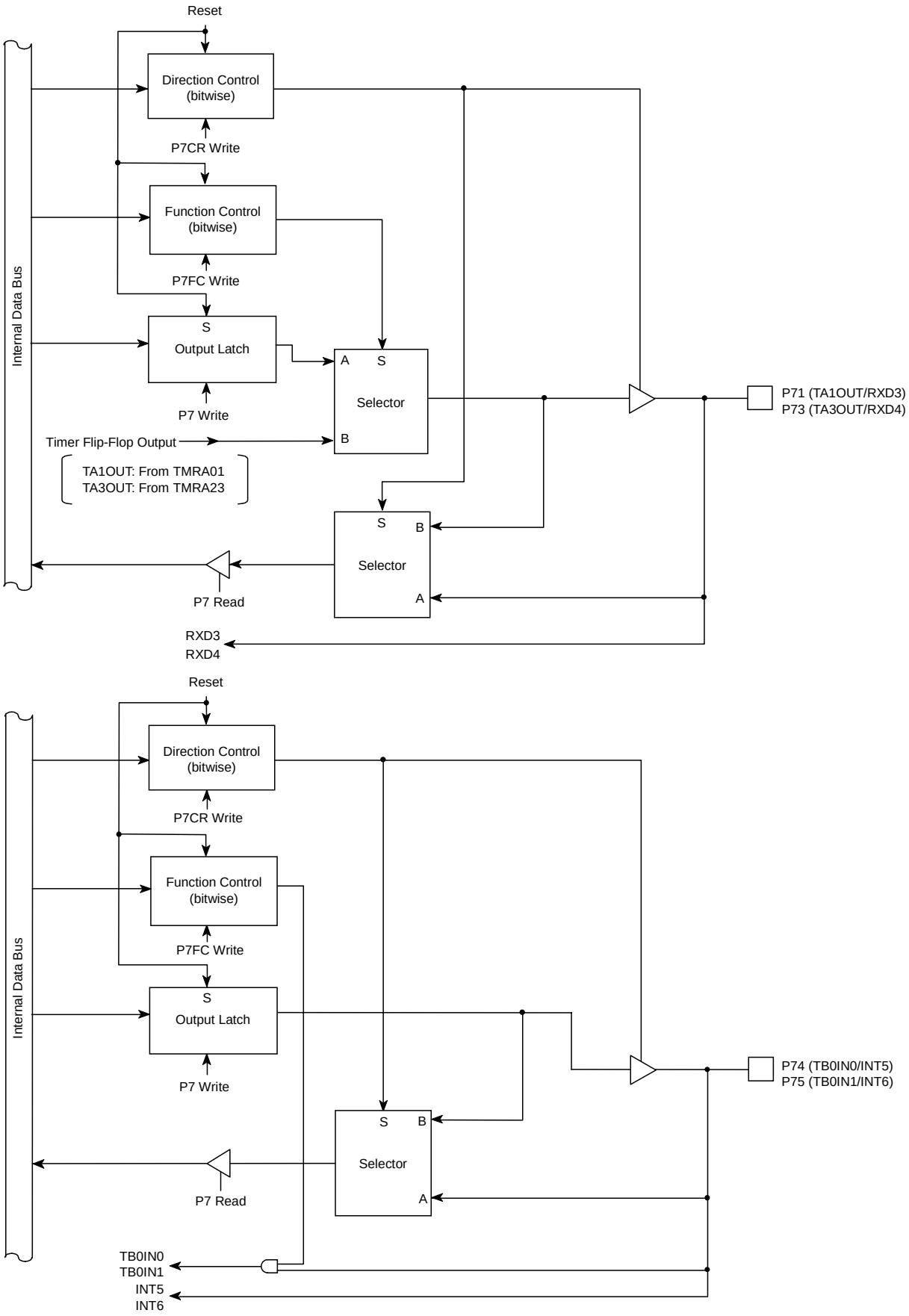


Figure 7.16 Port 7 (P71, P73, P74, P75)

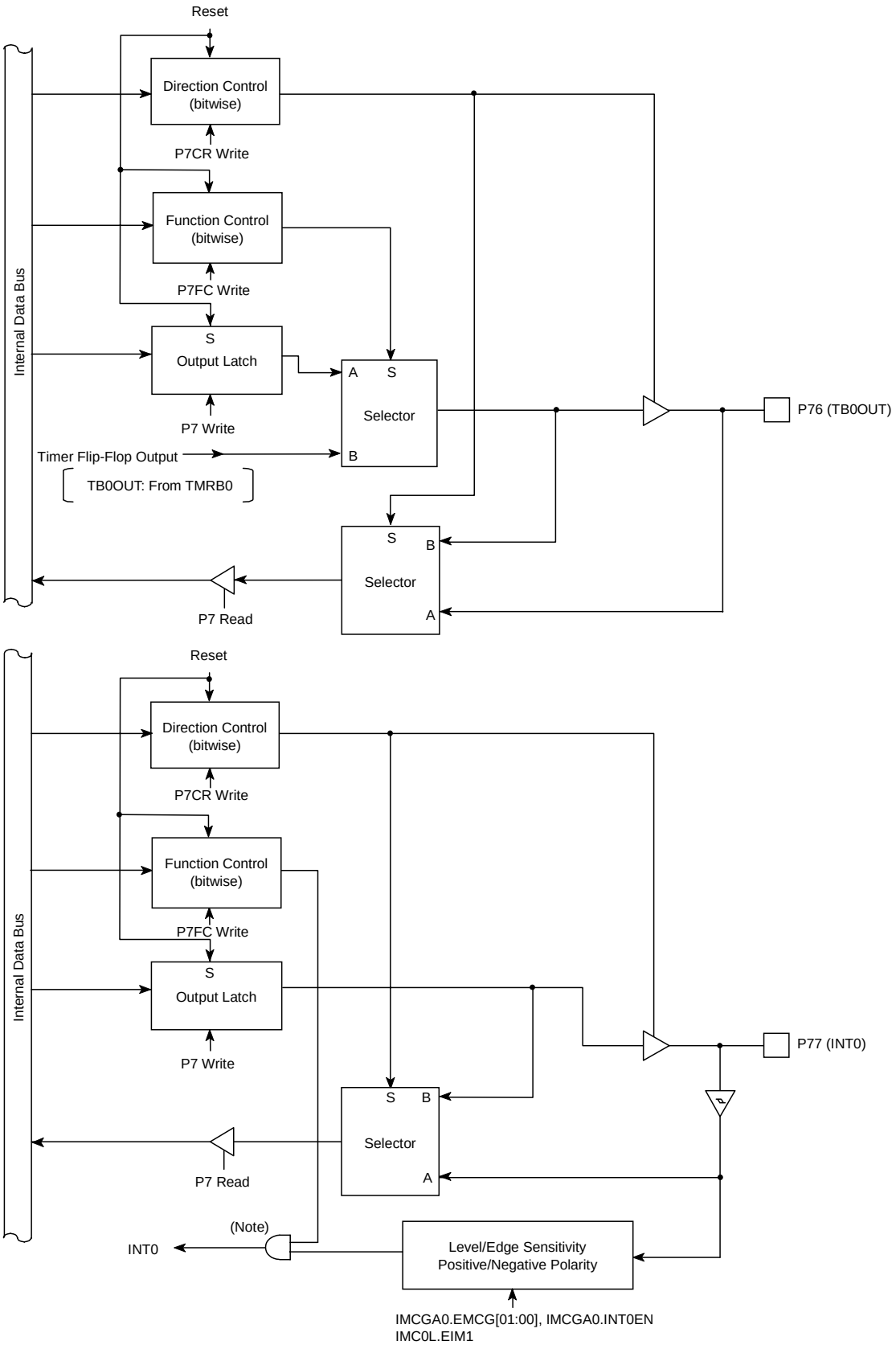


Figure 7.17 Port 7 (P76, P77)

Port 7 Register

P7 (0xFFFF_F02B)

	7	6	5	4	3	2	1	0
Name	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W							
Reset Value	Input mode (The Output Latch is set to 1.)							
	1	1	1	1	1	1	1	1

Port 7 Control Register

P7CR (0xFFFF_F02E)

	7	6	5	4	3	2	1	0
Name	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	0: IN 1: OUT							

Port 7 Direction Settings

0	Input
1	Output

Port 7 Function Register

P7FC (0xFFFF_F02F)

	7	6	5	4	3	2	1	0
Name	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	0: Port 1: Wake-up INT0	0: Port 1: TB0OUT	0: Port 1: TB0IN1	0: Port 1: TB0IN0	0: Port 1: TA3OUT 1: RXD4	0: Port 1: TA2IN 1: TXD4	0: Port 1: TA1OUT 1: RXD3	0: Port 1: TA0IN 1: TXD3

INT0 Settings

P7FC.P77F	1
P7CR.P77C	0

Note: Required to exit STOP mode, with SYSCR2.DRVE cleared. Otherwise, unneeded.

TB0OUT Settings

P7FC.P76F	1
P7CR.P76C	1

TB0IN1 Settings

P7FC.P75F	1
P7CR.P75C	0

TB0IN0 Settings

P7FC.P74F	1
P7CR.P74C	0

RXD4 Settings / TA3OUT Settings

P7FC.P73F	1	P7FC.P73F	1
P7CR.P73C	0	P7CR.P73C	1

TA2IN Settings / TXD4 Settings

P7FC.P72F	1	P7FC.P72F	1
P7CR.P72C	0	P7CR.P72C	1

RXD3 Settings / TA1OUT Settings

P7FC.P71F	1	P7FC.P71F	1
P7CR.P71C	0	P7CR.P71C	1

TA0IN Settings / TXD3 Settings

P7FC.P70F	1	P7FC.P70F	1
P7CR.P70C	0	P7CR.P70C	1

Figure 7.18 Port 7 Registers

7.8 Port 8 (P80–P87)

Eight Port 8 pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. Upon reset, all Port 8 pins are configured as input port pins, and the Output Latch (P8) is set to all 1s. Port 8 pins (except P87) can be programmed as clock inputs to 16-bit timers, timer flip-flop outputs from 16-bit timers, or external interrupt request pins (INT7 through INTA).

Setting the P8FC register bits configures the Port 8 pins for dedicated functions. A reset clears all the P8CR and P8FC register bits, configuring all Port 8 pins as input port pins.

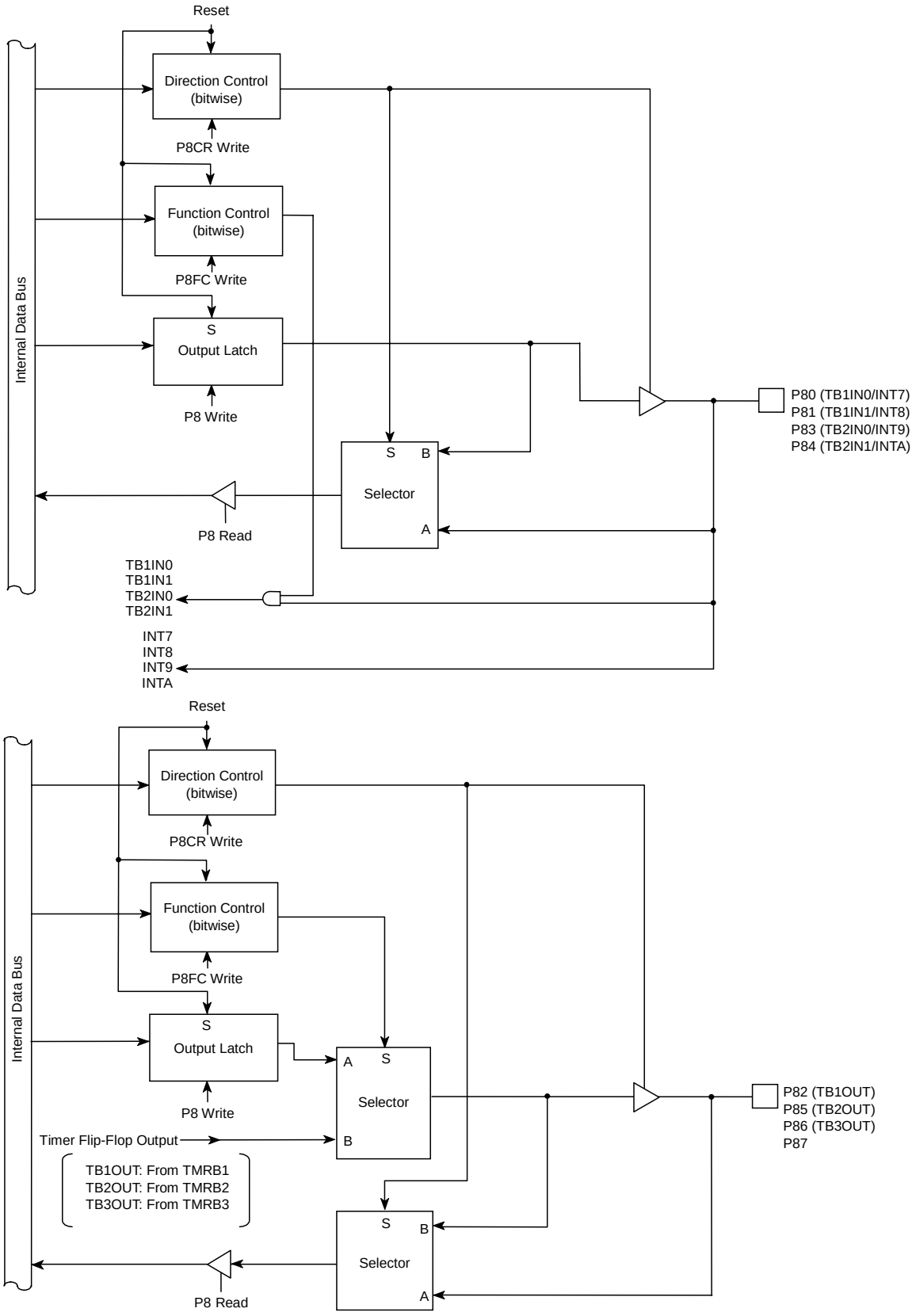


Figure 7.19 Port 8 (P80~P87)

Port 8 Register

P8 (0xFFFF_F030)

	7	6	5	4	3	2	1	0
Name	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
Reset Value	Input mode (The Output Latch is set to 1.)							

Port 8 Control Register

P8CR (0xFFFF_F032)

	7	6	5	4	3	2	1	0
Name	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	0: IN 1: OUT							

Port 8 Direction Settings

0	Input
1	Output

Port 8 Function Register

P8FC (0xFFFF_F033)

	7	6	5	4	3	2	1	0
Name	—	P86F	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	W							
Reset Value	—	0	0	0	0	0	0	0
Function	Must be written as 0.	0: Port 1: TB3OUT	0: Port 1: TB2OUT	0: Port 1: TB2IN1	0: Port 1: TB2IN0	0: Port 1: TB1OUT	0: Port 1: TB1IN1	0: Port 1: TB1IN0

TB1OUT Settings

P8FC.P82F	1
P8CR.P82C	1

TB3OUT Settings

P8FC.P86F	1
P8CR.P86C	1

TB2OUT Settings

P8FC.P85F	1
P8CR.P85C	1

Figure 7.20 Port 8 Registers

7.9 Port 9 (P90–P97)

- P90–P95

  P90–P95 can be individually programmed to function as discrete general-purpose or dedicated I/O pins. Upon reset, P90–P95 are configured as input port pins, and the corresponding Output Latch (P9) bits are set to 1.

  Setting the bits in the P9FC register configures the corresponding pin for SIO input or output pins. A reset clears the relevant P9CR and P9FC bits, configuring P90–P95 as input port pins.

- P96–P97

  P96 and P97 function as general-purpose I/O pins. As output ports, P96 and P97 are configured as open-drain outputs.

  Upon reset, the relevant Output Latch (P9) bits are set to 1, and the P9CR register bits are set, causing P96 and P97 to assume the high-impedance state.

  P96 and P97 can also be used as the XT1 and XT2 pins; in this case, a low-frequency crystal is connected between XT1 and XT2 to provide for Dual-Clock mode, which is controlled through System Clock Control Registers 0 and 1 (SYSCR0 and SYSCR1).

(1) P90 (TXD0) and P93 (TXD1)

P90 and P93 can be programmed to function as either general-purpose I/O pins or TXD output pins for SIO channels. P90 and P93 are configurable as open-drain outputs.

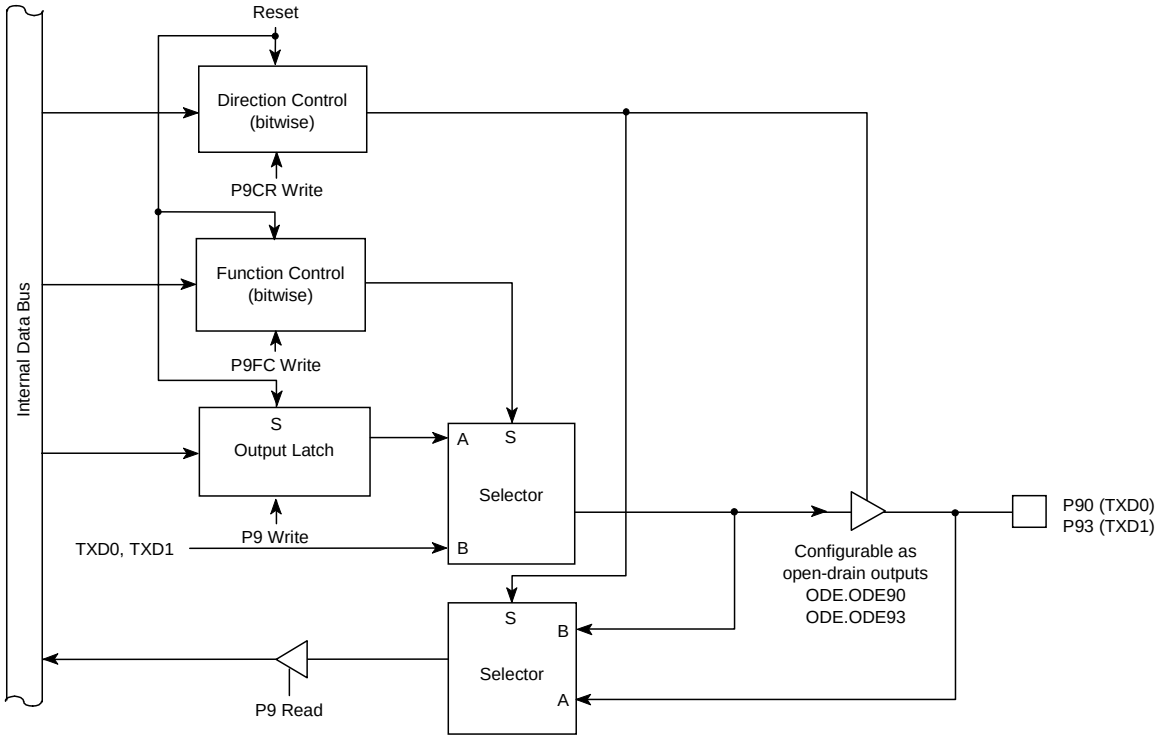


Figure 7.21 Port 9 (P90, P93)

(2) P91 (RXD0) and P94 (RXD1)

P91 and P94 can be programmed to function as either general-purpose I/O pins or RXD input pins for SIO channels.

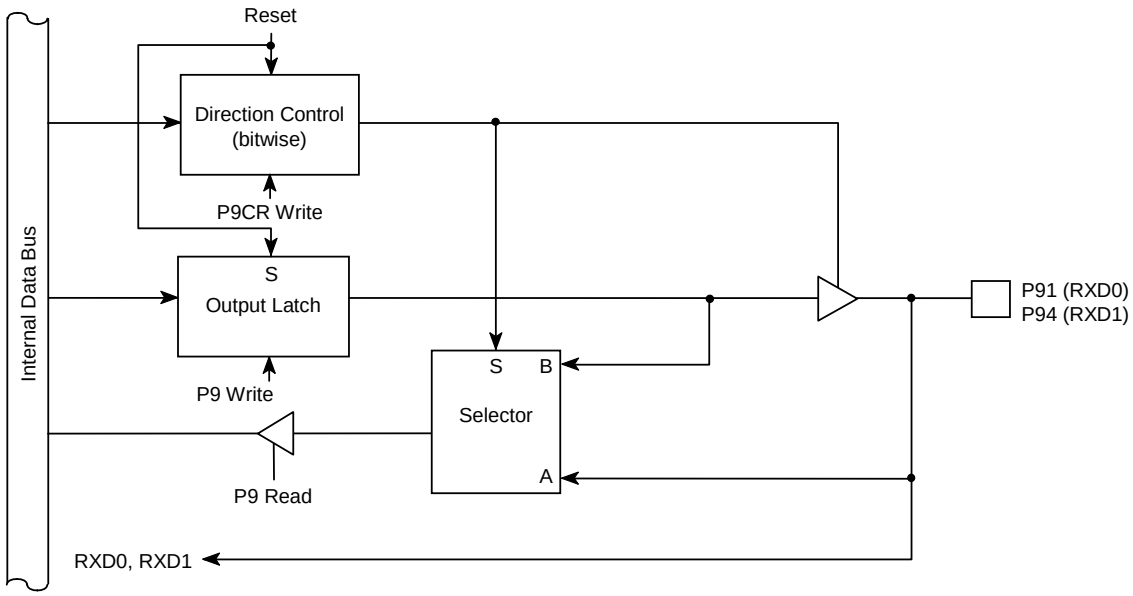


Figure 7.22 Port 9 (P91, P94)

(3) P92 (SCLK0/ $\overline{\text{CTS0}}$) and P95 (SCLK1/ $\overline{\text{CTS1}}$)

P92 and P95 can be programmed to function as general-purpose I/O pins, or SCLK clock input or output pins or $\overline{\text{CTS}}$ input pins for SIO channels.

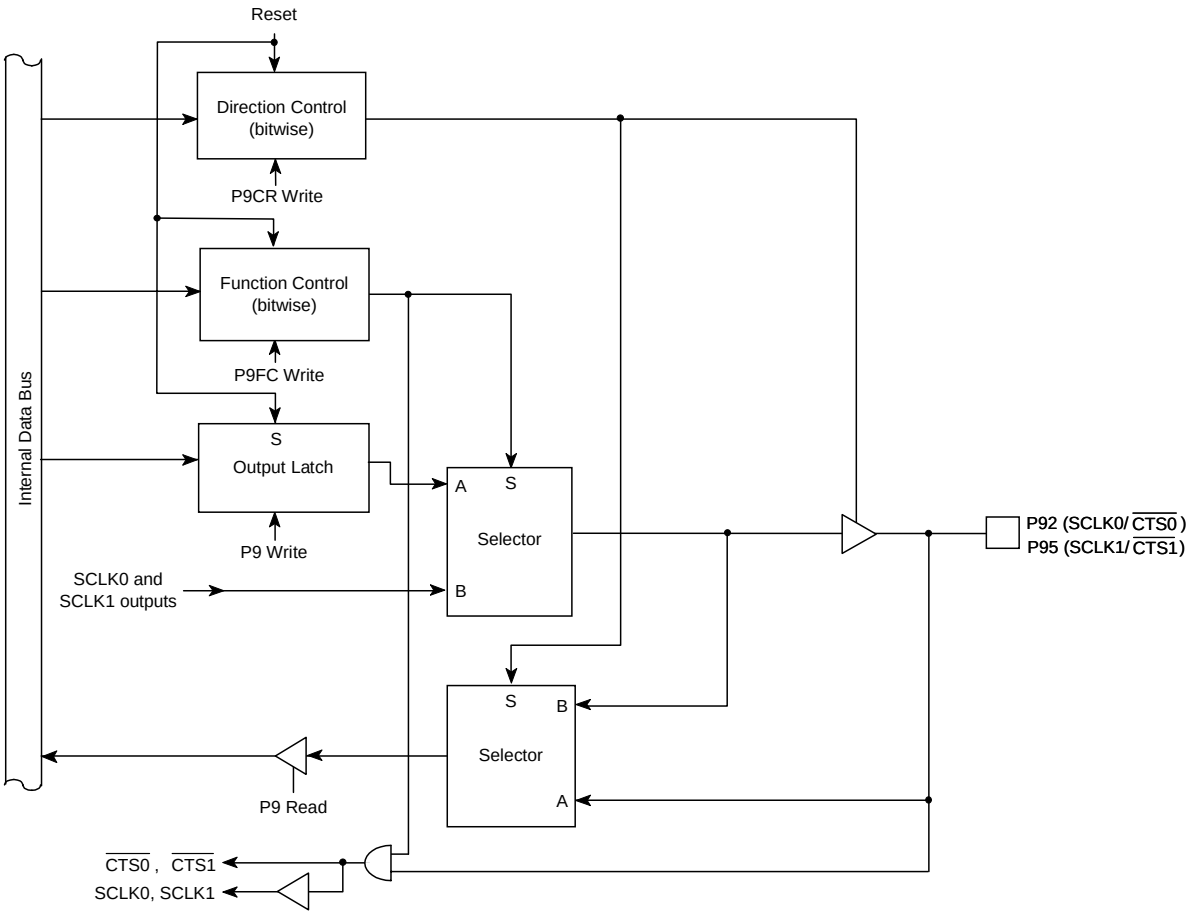


Figure 7.23 Port 9 (P92, P95)

(4) P96 (XT1) and P97 (XT2)

P96 and P97 function as general-purpose I/O pins. Alternatively, P96 and P97 can be used as the XT1 and XT2 pins for connecting a low-frequency crystal.

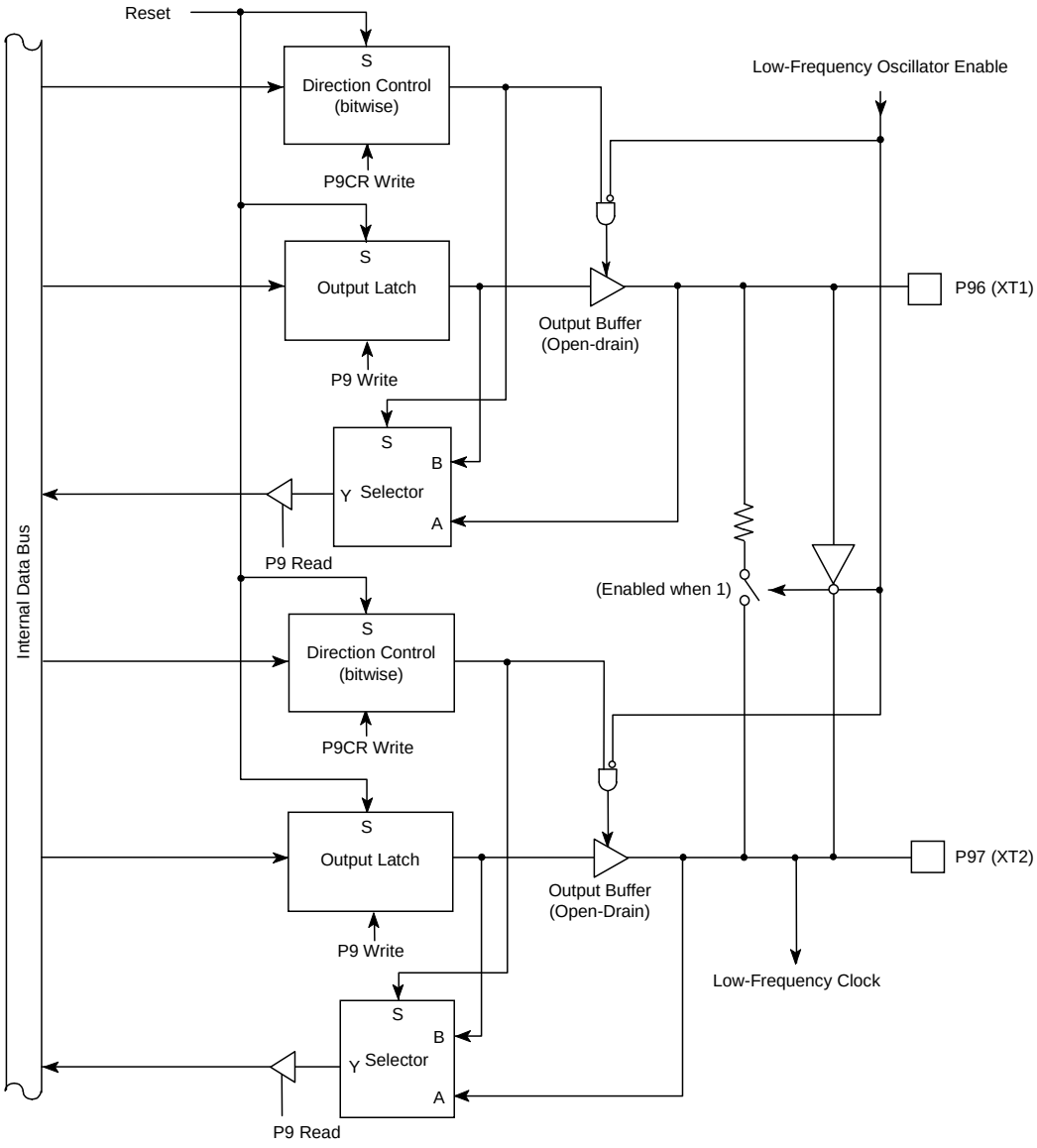


Figure 7.24 Port 9 (P96, P97)

Port 9 Register

P9 (0xFFFF_F031)

	7	6	5	4	3	2	1	0
Name	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
Reset Value	Output mode		Input mode					
	1	1	1	1	1	1	1	1

Port 9 Control Register

P9CR (0xFFFF_F034)

	7	6	5	4	3	2	1	0
Name	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write	W							
Reset Value	1	1	0	0	0	0	0	0
Function	0: IN 1: OUT							

Port 9 Direction Settings

0	Input
1	Output

Port 9 Function Register

P9FC (0xFFFF_F035)

	7	6	5	4	3	2	1	0
Name	—	—	P95F	—	P93F	P92F	—	P90F
Read/Write	W							
Reset Value	—	—	0	—	0	0	—	0
Function			0: Port 1: SCLK1 output or $\overline{CTS1}$ / SCLK1 input		0: Port 1: TXD1	0: Port 1: SCLK0 output or $\overline{CTS0}$ / SCLK0 input		0: Port 1: TXD0

$\overline{CTS1}$/SCLK1 Input Settings

P9FC.P95F	1
P9CR.P95C	0

SCLK1 Output Settings

P9FC.P95F	1
P9CR.P95C	1

TXD0 Output Settings

P9FC.P90F	1
P9CR.P90C	1

TXD1 Output Settings

P9FC.P93F	1
P9CR.P93C	1

$\overline{CTS1}$/SCLK0 Input Settings

P9FC.P92F	1
P9CR.P92C	0

SCLK0 Output Settings

P9FC.P92F	1
P9CR.P92C	1

Note 1: Setting bit 0 of the Open-Drain Enable (ODE) register configures theTXD0 pin as an open-drain output. Setting bit 1 of the ODE register configures the TXD1 pin as an open-drain output. See Section 7.11.
The P91/RXD0 and P94/RXD1 pins do not have bits for selecting pin functions. These pins can be continuously used as shared input port and serial data input pins.

Note 2: Low-speed oscillator consideration
When a low-frequency crystal is connected between XT1 (P96) and XT2 (P97), the following register settings are required to reduce power consumption:

When a crystal is connected between XT1 and XT2:
P9CR.P96C–P97C = 11
P9.P96–P97 = 00

When XT1 is driven with an external clock:
P9CR.P96C–P97C = 11
P9.P96–P97 = 10

Figure 7.25 Port 9 Registers

7.10 Port A (PA0–PA7)

Eight Port A pins can be individually programmed to function as discrete general-purpose or dedicated I/O pins. Upon reset, all Port A pins are configured as input port pins.

Alternatively, PA0–PA3 can be programmed as external interrupt request pins (INT1–INT4), and PA5–PA7 as the Serial Bus Interface (SBI) pins.

Setting the PAFC register bits configures the corresponding Port 8 pins for dedicated functions. A reset clears all the PACR and PAFC register bits, configuring all Port A pins as input port pins.

When INT1–INT4 are used as a wake-up from STOP mode with the SYSCR2.DRVE bit cleared, the corresponding bits in the PAFC register must be set to 1.

In the TMP1940FDBF with an on-chip flash, Port A can act as an interface to the DSU ICE. For a detailed description, see the TMP1940FDBF datasheet pages.

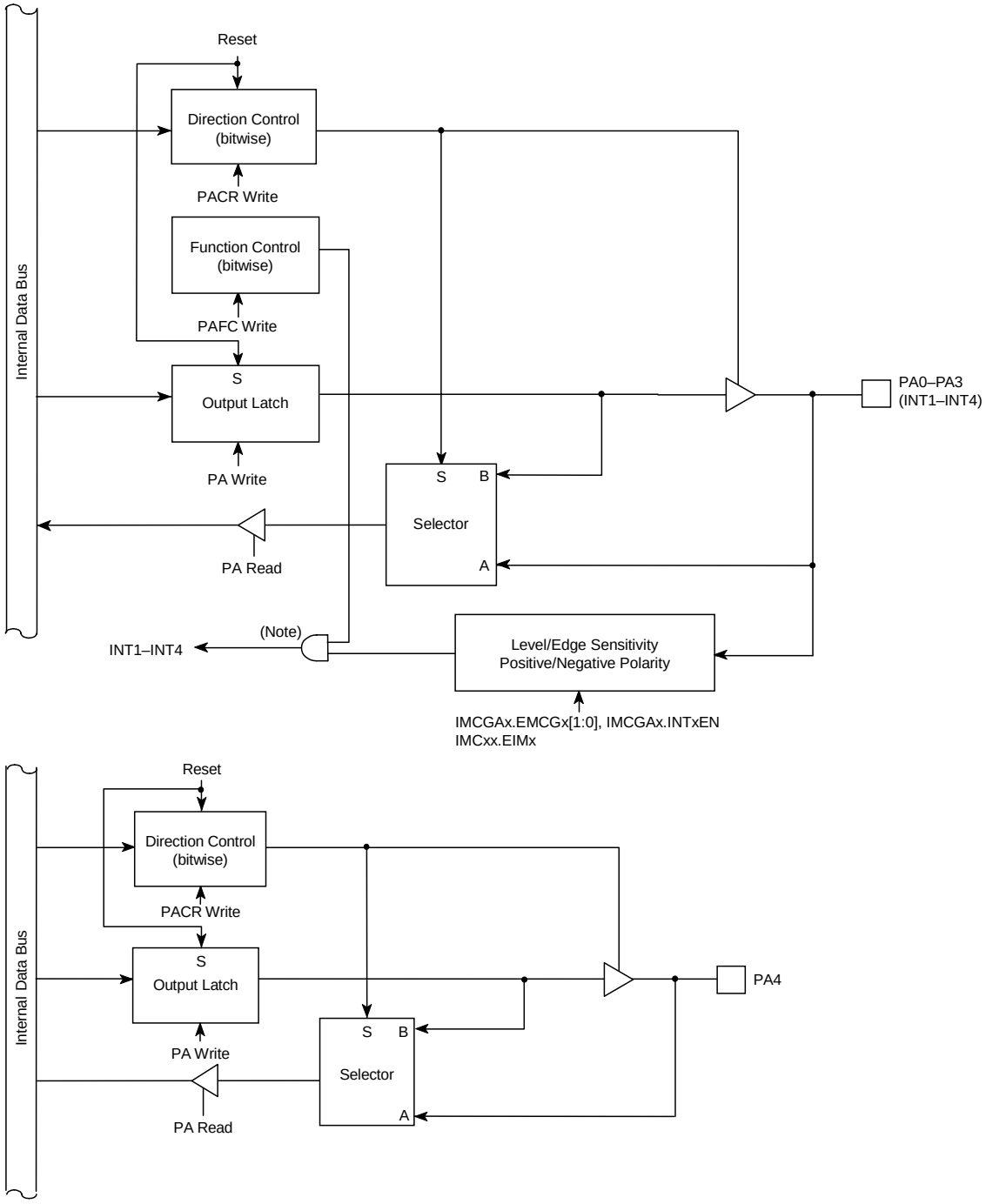


Figure 7.26 Port A (PA0–PA4)

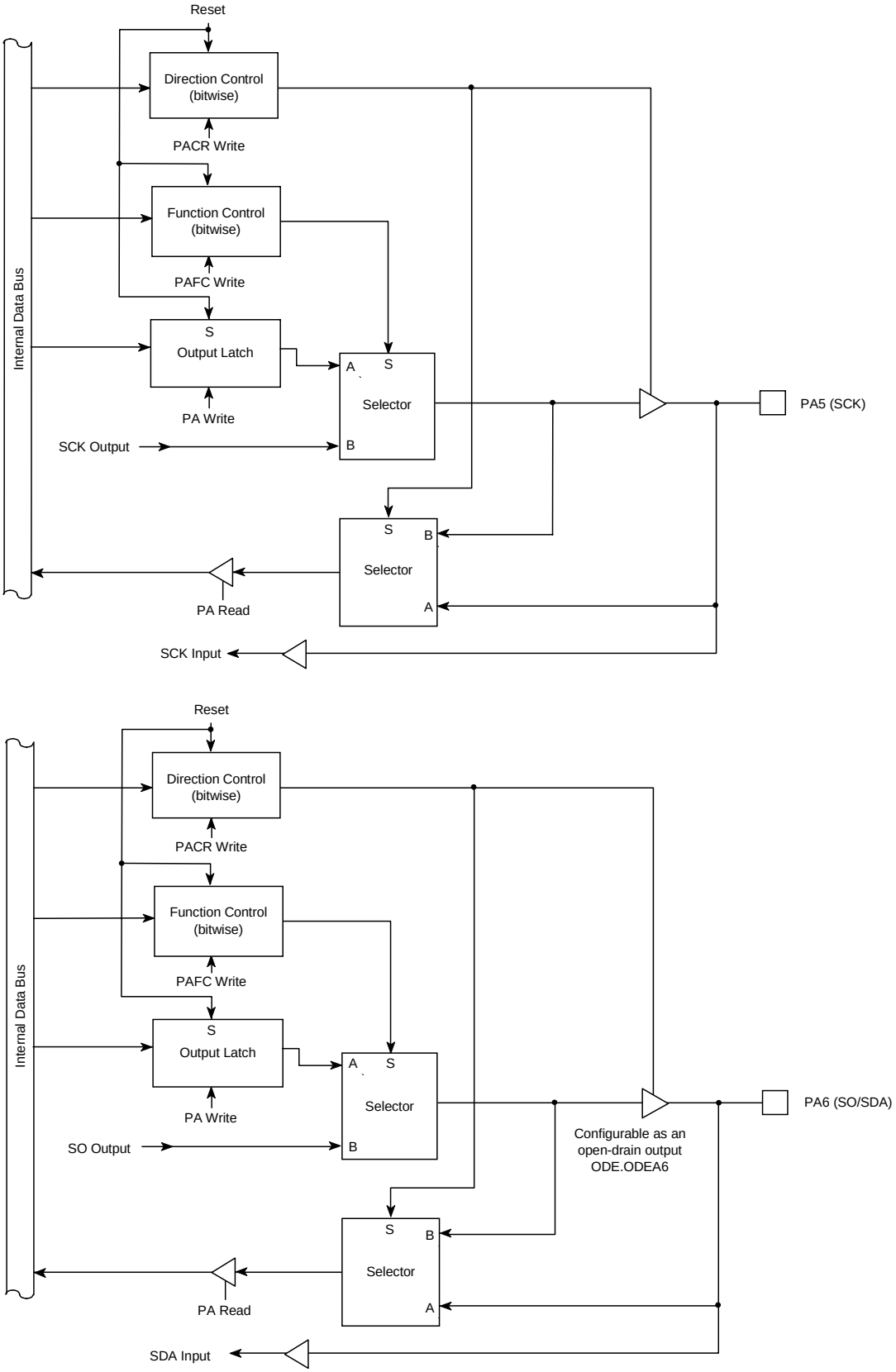


Figure 7.27 Port A (PA5–PA6)

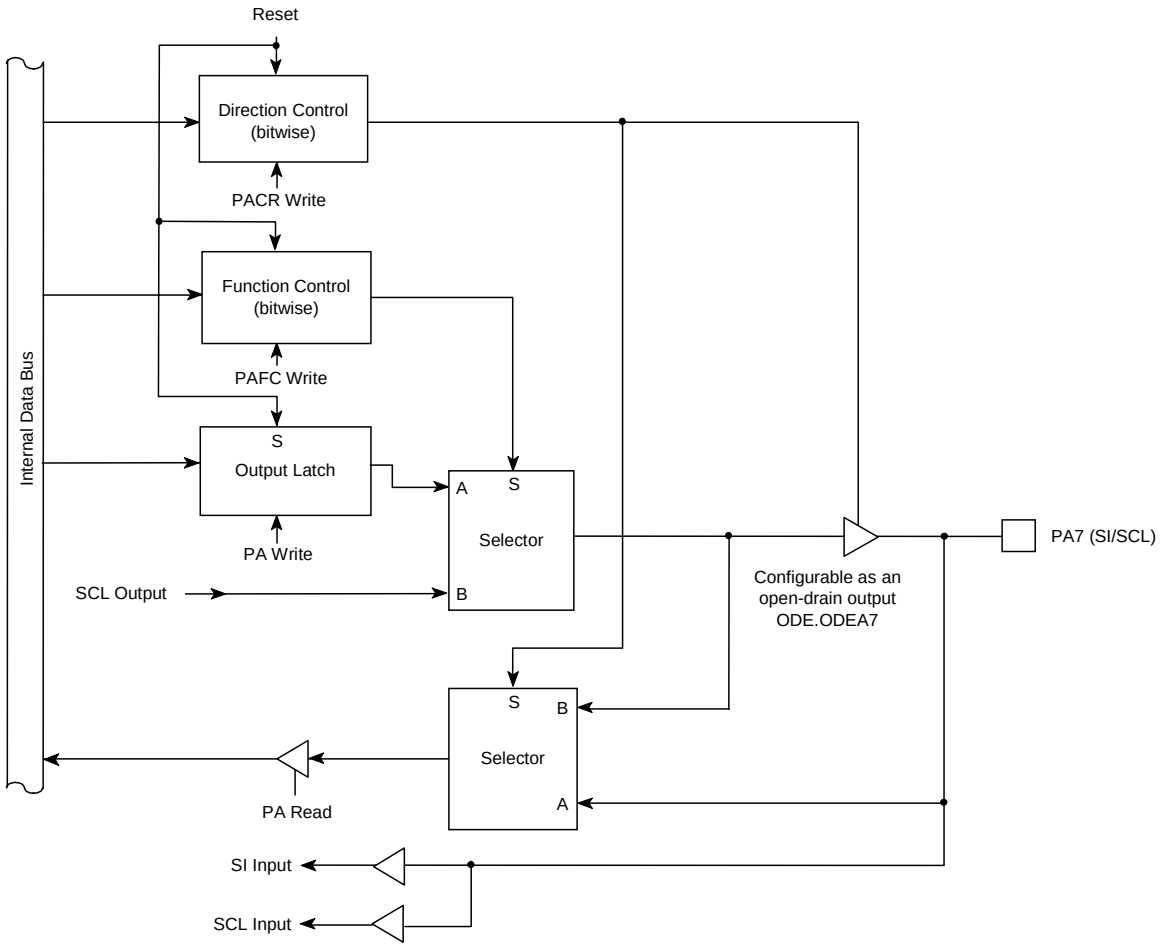


Figure 7.28 Port A (PA7)

Port A Register

PA (0xFFFF_F036)

	7	6	5	4	3	2	1	0
Name	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R/W							
Reset Value	Input mode (The Output Latch set to 1.)							

Port A Control Register

PACR (0xFFFF_F038)

	7	6	5	4	3	2	1	0
Name	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	0: IN 1: OUT							

Port A Direction Settings

0	Input
1	Output

Port A Function Register

PAFC (0xFFFF_F039)

	7	6	5	4	3	2	1	0
Name	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
Read/Write	W							
Reset Value	0	0	0					
Function	0: Port 1: SCL output	0: Port 1: SDA/SO output	0: Port 1: SCK output	Must be written as 0.	0: Port 1: Wake-up INT4 input	0: Port 1: Wake-up INT3 input	0: Port 1: Wake-up INT2 input	0: Port 1: Wake-up INT1 input

Wake-up INT1–INT4 Input Settings

PAFC.PAxF	1
PACR.PAxC	0

Note: Required to exit STOP mode, with SYSCR2.DRVE cleared. Otherwise, unneeded.

SCK Output Settings

PAFC.PA5F	1
PACR.PA5C	1

SDA/SO Output Settings

PAFC.PA6F	1
PACR.PA6C	1

SCL Output Settings

PAFC.PA7F	1
PACR.PA7C	1

Figure 7.29 Port A Registers

7.11 Open-Drain Output Control

The TXD output pins (P70, P72, P90 and P93) of the SIO, and the SO/SDA (PA6) and SI/SCL (PA7) pins of the Serial Bus Interface (SBI) can be configured as either push-pull or open-drain outputs.

Open-Drain Enable Register

		7	6	5	4	3	2	1	0
ODE (0xFFFF_F050)	Name	—	—	ODE72	ODE70	ODEA7	ODEA6	ODE93	ODE90
	Read/Write	—	—	R/W					
	Reset Value	—	—	0	0	0	0	0	0
	Function			P72 0: Push-pull 1: Open-drain	P70 0: Push-pull 1: Open-drain	PA7 0: Push-pull 1: Open-drain	PA6 0: Push-pull 1: Open-drain	P93 0: Push-pull 1: Open-drain	P90 0: Push-pull 1: Open-drain

Figure 7.30 Open-Drain Enable Register

8. External Bus Interface

The TMP1940CYAF contains external bus interface logic that handles the transfer of information between the internal busses and the memory or peripherals in the external address space. It consists of the External Bus Interface (EBIF) logic and the Chip Select/Wait Controller.

The CS/Wait Controller provides four programmable chip select signals, with variable block sizes. The chip select function supports automatic wait-state generation and data bus sizing (8-bit or 16-bit) for each of the four address blocks and the rest of the external address locations.

The EBIF logic controls the timing of the external bus, based on the settings of the CS/Wait Controller. The EBIF logic also performs dynamic bus sizing and bus arbitration.

(1) Wait-state generation

Individually programmable for each address block

- Automatic insertion of up to seven wait cycles
- $\overline{\text{WAIT}}$ pin

(2) Data bus width

Individually programmable (8-bit or 16-bit) for each address block

(3) Read recovery cycles

Individually programmable (to up to 2 cycles) for each address block. Read recovery cycles are dummy cycles inserted between two consecutive external bus cycles.

(4) ALE pulse width

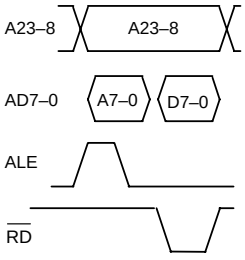
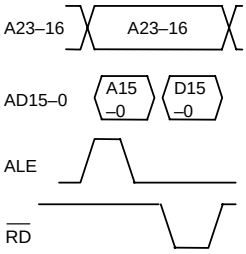
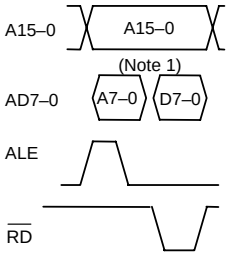
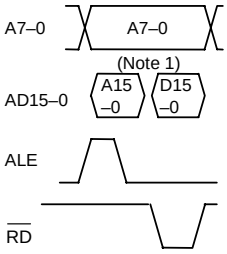
Selectable ALE pulse width (0.5 or 1.5 cycles). This setting applies to all the address blocks.

(5) Bus arbitration

8.1 Address and Data Buses

8.1.1 Supported Configurations

For external memory interface, Port 0 (AD0–AD7), Port 1 (AD8–AD15/A8–A15) and Port 2 (A16–A23/A0–A7) pins can be configured as the address and data buses. The TMP1940CYAF supports the following four bus configurations.

		A	B	C	D
Address Lines		24 Max (16 Mbytes)	24 Max (16 Mbytes)	16 Max (64 Mbytes)	8 Max (256 Bytes)
Data Lines		8	16	8	16
Multiplexed Address/Data Lines		8	16	0	0
Pin Functions	Port 0	AD0–AD7	AD0–AD7	AD0–AD7	AD0–AD7
	Port 1	A8–A15	AD8–AD15	A8–A15	AD8–AD15
	Port 2	A16–A23	A16–A23	A0–A7	A0–A7
Timing Diagram					

Note 1: Because the data bus is multiplxed with the address bus, even in the C and D configurations, address bits also appear on the AD bus prior to the data being accepted or provided.

Note 2: Upon reset, all of Ports 0–2 are configured as general-purpose input ports; programming is required to use them as address or data bus pins.

Note 3: Address and data bus configurations are selectable through the programming of the P1CR, P1FC, P2CR and P2FC registers.

8.1.2 States of the Address Bus During On-Chip Address Accesses

While an on-chip address is being accessed, the address bus maintains the previous address externally presented. During this time, the address/data bus assumes the high-impedance state.

8.2 External Bus Operation

This section describes external bus operations. In the timing diagrams which follow, A23–A16 is the address bus, and AD15–AD0 is the address/data bus.

This section only provides a functional description of the bus; refer to Section 18, *AC Electrical Characteristics*, for detailed timing specifications.

8.2.1 Basic Bus Operation

While the TMP1940CYAF provides a total of three clock cycles to perform a read or write, it also allows the bus cycle to be extended by inserting wait states.

Figure 8.1 shows external bus read timing. Figure 8.2 shows external bus write timing. While an on-chip address is being accessed, the external address bus maintains the previous value with the ALE pin kept inactive. During this time, the address/data bus assumes the high-impedance state, and bus control signals such as $\overline{RD}$ and $\overline{WR}$ remain inactive.

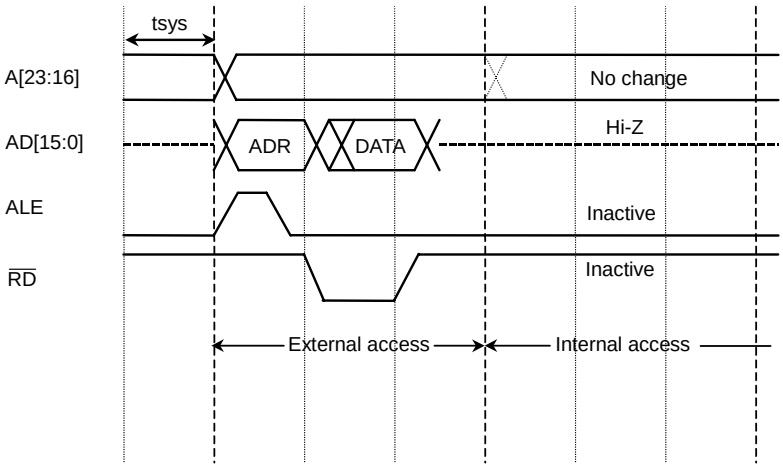


Figure 8.1 Read Cycle Timing

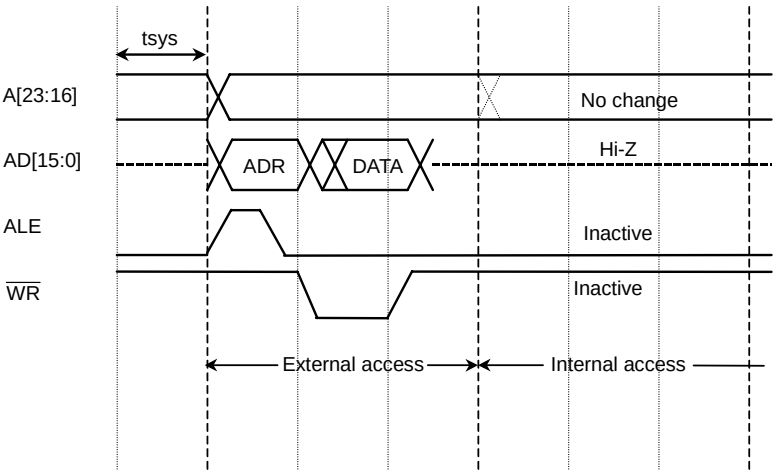


Figure 8.2 Write Cycle Timing

Note: tsys is the system clock period.

8.2.2 Wait Timing

The CS/Wait Controller provides two ways to insert wait states in a bus cycle.

Each address block can be programmed either:

- to insert required number of wait state cycles (up to seven cycles), or
- to use the $\overline{WAIT}$ pin to insert wait states dynamically on a cycle basis

Following are bus cycle timing diagrams with wait states.

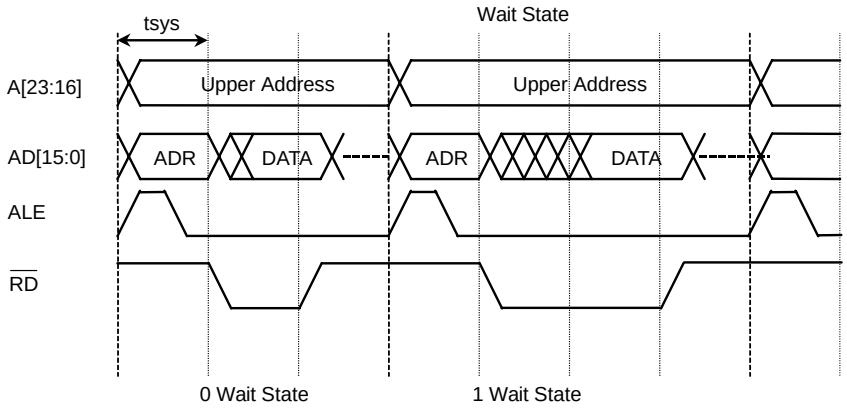


Figure 8.3 Read Cycle Timing (with Zero and One Wait State Cycle)

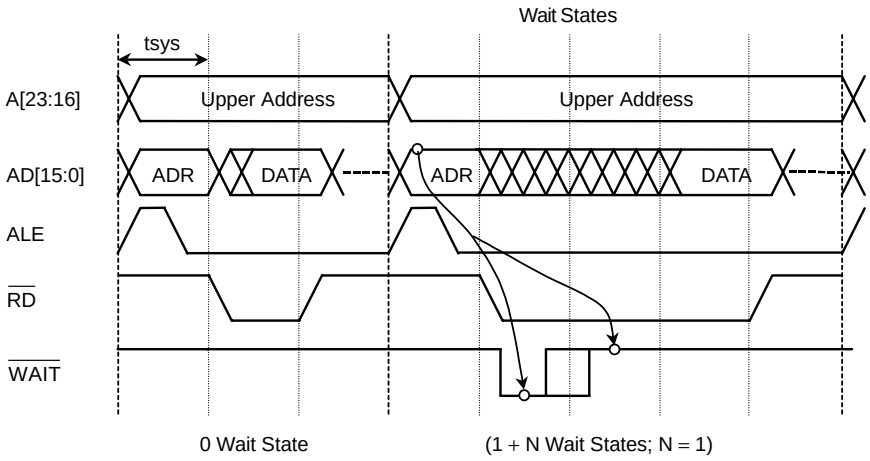


Figure 8.4 Read Cycle Timing (with 1 + N Wait States; N=1)

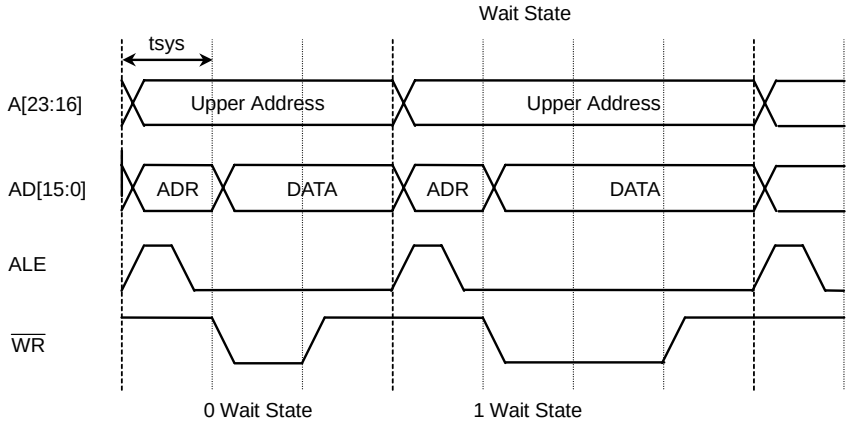


Figure 8.5 Write Cycle Timing (with Zero and One Wait State Cycle)

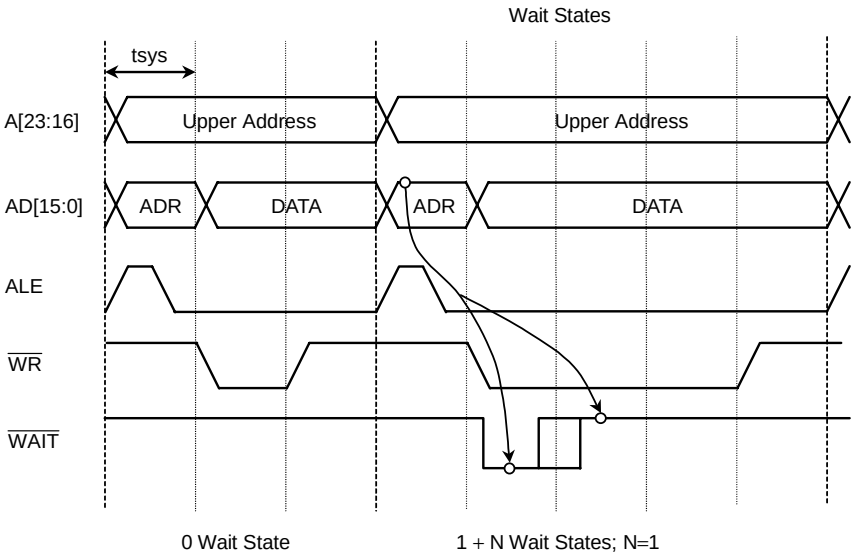


Figure 8.6 Write Cycle Timing (with 1 + N Wait State Cycles; N=1)

8.2.3 ALE Pulse Width

The ALE pulse width is programmed to 0.5 or 1.5 clock cycles through the ALESEL bit of the SYSCR3 register within the CG. The default is 1.5 cycles. This setting applies to the whole external address space.

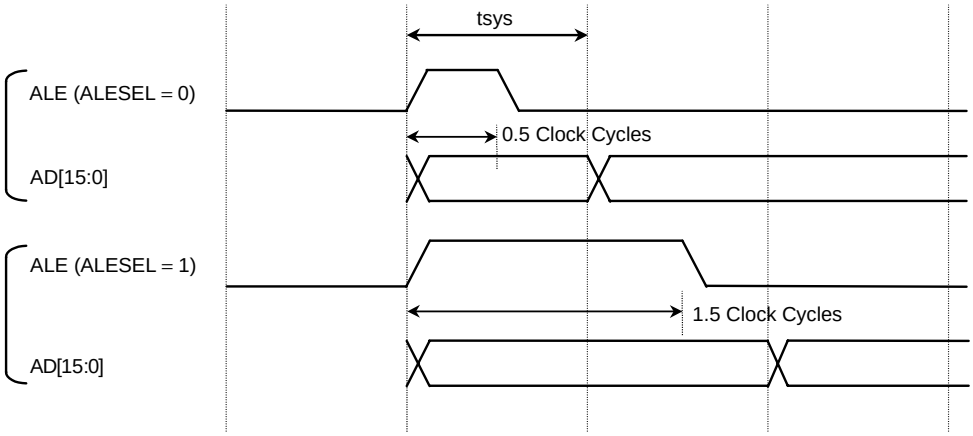


Figure 8.7 ALE Pulse Width

Figure 8.8 shows read cycle timing, with the ALE width programmed to 0.5 and 1.5 clock cycles.

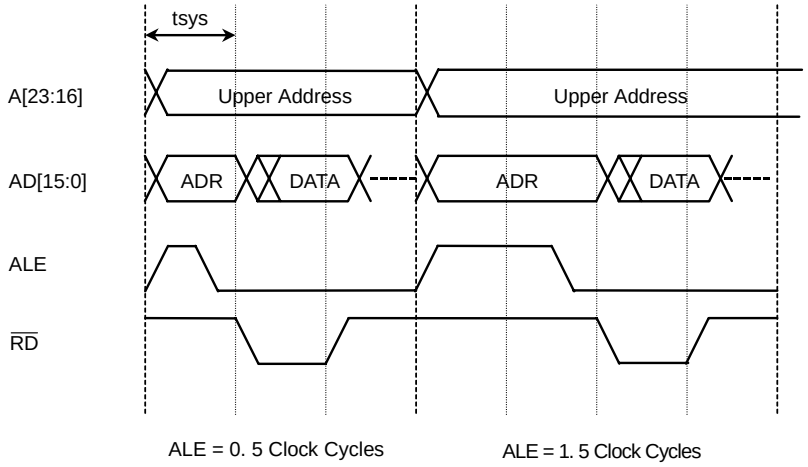


Figure 8.8 Read Cycle Timing (ALE = 0.5 and 1.5 Clock Cycles)

8.2.4 Read Recovery Time

Following an external bus read cycle, a certain recovery time may be required before initiating the next external bus cycle. To allow for a read recovery time, one or two dummy cycles can be inserted between back-to-back bus cycles. (Dummy cycles can only be inserted immediately after a read.)

- Between an external read and an external read: Programmable
- Between an external read and an external write: Programmable
- After an external write: No dummy cycle

Dummy cycle insertion is programmable in the CS/Wait Controller.

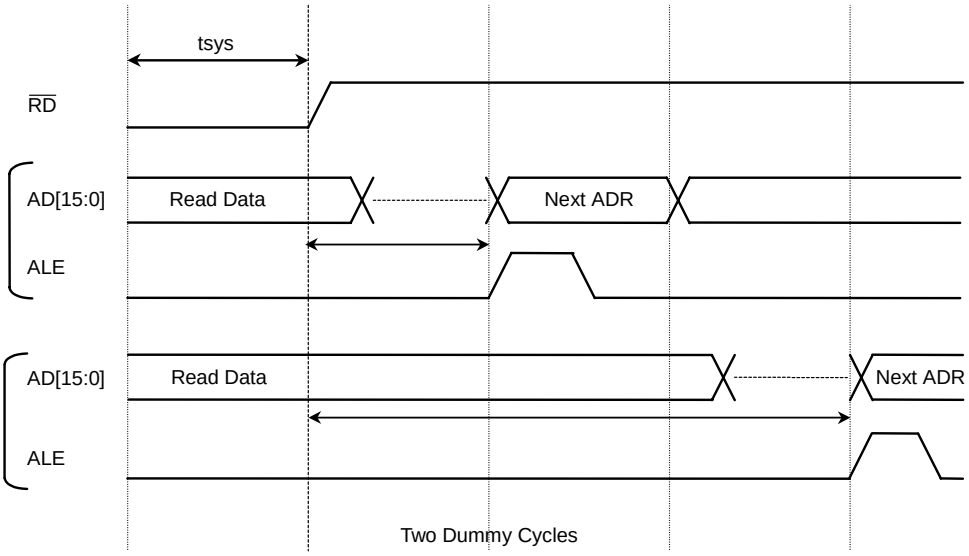


Figure 8.9 Read Recovery Time

Dummy cycles insert idle cycles between transfers to enable slow off-chip peripherals to remove data from the data bus before the next transfer begins. This provides a sufficient time after the $\overline{RD}$ strobe for the previous read is deasserted until the address for the next read or write is placed on the address bus. Figure 8.10 shows bus cycle timing with one and two dummy cycles inserted into bus cycles.

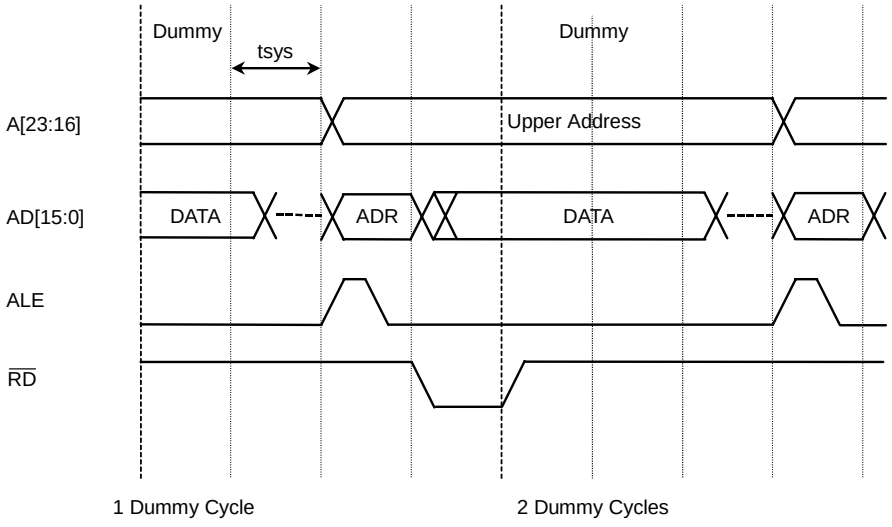


Figure 8.10 Read Cycle Timing (with Dummy Cycles Inserted)

8.3 Bus Arbitration

The TMP1940CYAF provides support for an external bus master to take control of the external bus. Two bus arbitration control signals, $\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$, are used to determine the bus master. One or more of the external devices on the bus can have the capability of becoming bus master for the external bus, but not the TMP1940CYAF internal bus.

8.3.1 Bus Access Control

External bus masters can gain control of the external bus, but not the TMP1940CYAF internal bus (G-Bus). Thus, external bus masters cannot access the TMP1940CYAF's on-chip memory and peripherals. The External Bus Interface (EBIF) logic in the TMP1940CYAF manages the arbitration of the external bus; the CPU and on-chip DMAC do not participate in any way in this bus arbitration. During external bus mastership, the CPU and the on-chip DMAC can access the internal memory (RAM and ROM) and registers.

Once an external device assumes bus mastership, the CPU or the on-chip DMAC has no way to regain the bus until the external bus master releases the bus. If the CPU or the on-chip DMAC issues an external memory access request, it is forced to wait until the TMP1940CYAF regains the bus. Therefore, should $\overline{\text{BUSRQ}}$ be left asserted for a long time, the TMP1940CYAF might suffer system lockups.

8.3.2 Bus Arbitration Flow

External devices capable of becoming bus masters assert $\overline{\text{BUSRQ}}$ to request the bus. The TMP1940CYAF samples $\overline{\text{BUSRQ}}$ at the end of each external bus cycle, as seen on its internal bus (G-Bus). When the TMP1940CYAF has made an internal decision to grant the bus, it asserts $\overline{\text{BUSAK}}$ to indicate to the requesting device that the bus is available. At the same time, the TMP1940CYAF puts the address bus, the data bus and bus control signals in the high-impedance state.

A load or store may require multiple bus cycles, depending on the port size of the addressed device (dynamic bus sizing). In that case, the TMP1940CYAF does not grant the bus until the entire transfer is complete.

The TMP1940CYAF, if so programmed, automatically inserts dummy cycles between back-to-back bus cycles to allow for sufficient read recovery time. In dummy cycles, the TMP1940CYAF has already internally initiated a bus cycle on the G-Bus for the next external access. The TMP1940CYAF can only accept an external bus request at the boundary of an internal G-Bus bus cycle. Therefore, if $\overline{\text{BUSRQ}}$ is asserted during a dummy cycle, the TMP1940CYAF grants the bus after it completes the next external bus cycle.

An external bus master must keep $\overline{\text{BUSRQ}}$ asserted until it is granted the bus.

A timing diagram of the bus arbitration sequence is shown in Figure 8.11.

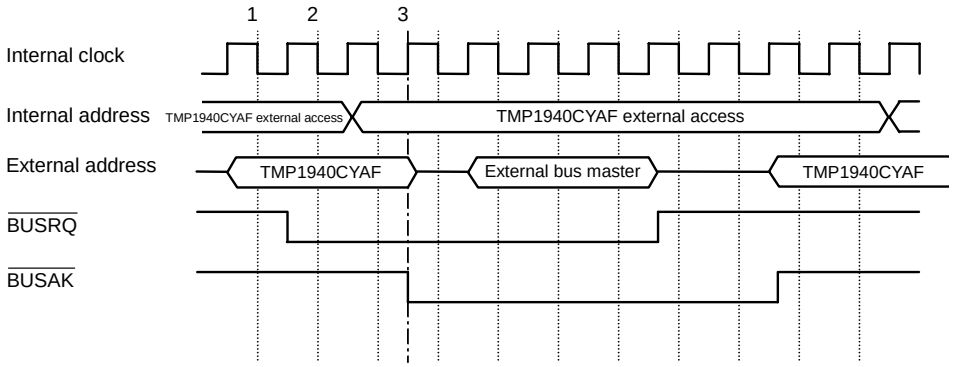


Figure 8.11 Bus Arbitration Timing Diagram

1. $\overline{\text{BUSRQ}}$ is sampled high.
2. The TMP1940CYAF recognizes the assertion of $\overline{\text{BUSRQ}}$.
3. The TMP1940CYAF asserts $\overline{\text{BUSAK}}$ at the completion of the current bus cycle. The external bus master recognizes $\overline{\text{BUSAK}}$ and assumes bus mastership to start a bus transfer.

8.3.3 Relinquishing the bus

When the external bus master has completed its bus transactions, it deasserts $\overline{\text{BUSRQ}}$ to relinquish the bus to the TMP1940CYAF. Figure 8.12 shows the timing for an external bus master to relinquish the bus.

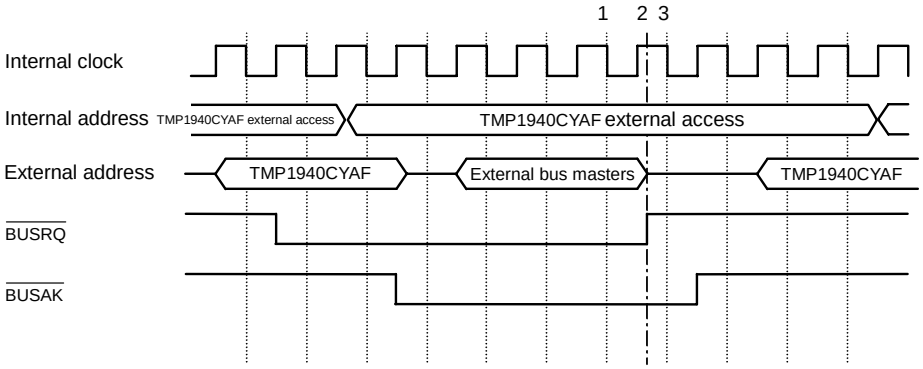


Figure 8.12 External Bus Master Relinquishing the Bus

1. The external bus master has control of the bus.
2. When the external bus master no longer needs the bus, it deasserts $\overline{\text{BUSRQ}}$.
3. In response to the deassertion of $\overline{\text{BUSRQ}}$, the TMP1940CYAF deasserts $\overline{\text{BUSAK}}$.

9. Chip Select/Wait Controller

The TMP1940CYAF supports direct connections to ROM and SRAM devices.

The TMP1940CYAF provides four programmable chip select signals. Programmable features include variable block sizes, data bus width, wait state insertion, and dummy cycle insertion for back-to-back bus cycles.

$\overline{CS0} - \overline{CS3}$ (multiplexed with P40–P43) are the chip select output pins for the CS0–CS3 address ranges. These chip select signals are generated when the CPU or on-chip DMAC issues an address within the programmed ranges. The P40–P43 pins must be configured as $\overline{CS0} - \overline{CS3}$ by programming the Port A Control (P4CR) register and the Port 4 Function (P4FC) register.

Chip select address ranges are defined in terms of a base address and an address mask. There is a Base/Mask Address (BMAn) register for each of the four chip select signals, where n is a number from 0 to 3.

There is also a set of three Chip Select/Wait Control registers, B01CS, B23CS and BEXCS, each of which consists of a master enable bit, a data bus width bit, a wait state field and a dummy cycle field.

External memory devices can also use the $\overline{WAIT}$ pin to insert wait states and consequently prolong read and write bus cycles.

9.1 Programming Chip Select Ranges

Each of the four chip select address ranges is defined in the BMAn register. The basic chip select model allows one of the chip select output signals ($\overline{CS0} - \overline{CS3}$) to assert when an address on the address bus falls within a particular programmed range. The B01CS register defines specific operations for $\overline{CS0}$ and $\overline{CS1}$, and the B23CS register defines specific operations for $\overline{CS2}$ and $\overline{CS3}$ (see Section 9.2).

9.1.1 Base/Mask Address Registers (BMA0–BMA3)

The organizations of the BMAn registers are shown in Figure 9.1 and Figure 9.2. The base address (BAn) field specifies the starting address for a chip select. Any set bit in the address mask field (MAn) masks the corresponding base address bit. The address mask field determines the block size of a particular chip select line. The address is compared on every bus cycle.

(1) Base address

The base address (BAn) field specifies the upper 16 bits (A31–A16) of the starting address for a chip select. The lower 16 bits (A15–A0) are assumed to be zero. Thus, the base address is any multiple of 64 Kbytes starting at 0x0000_0000. Figure 9.3 shows the relationships between starting addresses and the BMAn values.

(2) Address mask

The address mask field defines whether any particular bits of the address should be compared or masked. Any set bit masks the corresponding base address bit. The address compare logic uses only the address bits that are not masked (i.e., mask bit cleared to 0) to detect an address match. Address bits that can be masked (i.e., supported block sizes) differ for the four chip select spaces as follows:

CS0 and CS1 spaces: A29–A14

CS2 and CS3 spaces: A30–A15

The address mask field defines the block size of a particular chip select line.

Note: Use physical addresses in the BMAn registers.

Base/Mask Address Registers

BMA0 (0xFFFF_E400)

	7	6	5	4	3	2	1	0
Name	MA0 (A29 – A14)							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS0 block size			0: The address compare logic uses this address bit.				
	15	14	13	12	11	10	9	8
Name	MA0 (A29 – A14)							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	1	1
Function	Must be written as 0.							
	23	22	21	20	19	18	17	16
Name	BA0							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23–A16 of the starting address for CS0							
	31	30	29	28	27	26	25	24
Name	BA0							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31–A24 of the starting address for CS0							

BMA1 (0xFFFF_E404)

	7	6	5	4	3	2	1	0
Name	MA1 (A29 – A14)							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS1 block size			0: The address compare logic uses this address bit.				
	15	14	13	12	11	10	9	8
Name	MA1 (A29 – A14)							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	1	1
Function	Must be written as 0.							
	23	22	21	20	19	18	17	16
Name	BA1							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23–A16 of the starting address for CS1							
	31	30	29	28	27	26	25	24
Name	BA1							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31–A24 of the starting address for CS1							

Note: Bits 10–15 in the BMA0 and BMA1 must be written as zeros. The CS0 and CS1 block sizes can vary from 16 Kbytes to 1 Gbytes. However, the TMP1940CYAF supports only 16 Mbytes of external address space. Therefore, bits 10–15 in the BMA0 and BMA1 must be cleared so that A24–A29 of an address will not be masked.

Figure 9.1 Base/Mask Address Registers (BMA0 and BMA1)

BMA2 (0xFFFF_E408)

	7	6	5	4	3	2	1	0
Name	MA2 (A30 – A15)							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS2 block size			0: The address compare logic uses this address bit.				
	15	14	13	12	11	10	9	8
Name	MA2 (A30 – A15)							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	1
Function	Must be written as 0.							
	23	22	21	20	19	18	17	16
Name	BA2							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23–A16 of the starting address for CS2							
	31	30	29	28	27	26	25	24
Name	BA2							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31–A24 of the starting address for CS2							

BMA3 (0xFFFF_E40C)

	7	6	5	4	3	2	1	0
Name	MA3 (A30 – A15)							
Read/Write	R/W							
Reset Value	1	1	1	1	1	1	1	1
Function	CS3 block size			0: The address compare logic uses this address bit.				
	15	14	13	12	11	10	9	8
Name	MA3 (A30 – A15)							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	1
Function	Must be written as 0.							
	23	22	21	20	19	18	17	16
Name	BA3							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A23–A16 of the starting address for CS3							
	31	30	29	28	27	26	25	24
Name	BA3							
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	A31–A24 of the starting address for CS3							

Note: Bits 9–15 in the BMA2 and BMA3 must be written as zeros. The CS2 and CS3 block sizes can vary from 32 Kbytes to 1 Gbytes. However, the TMP1940CYAF supports only 16 Mbytes of external address space. Therefore, bits 9–15 in the BMA0 and BMA1 must be cleared so that A24–A30 of an address will not be masked.

Figure 9.2 Base/Mask Address Registers (BMA2 and BMA3)

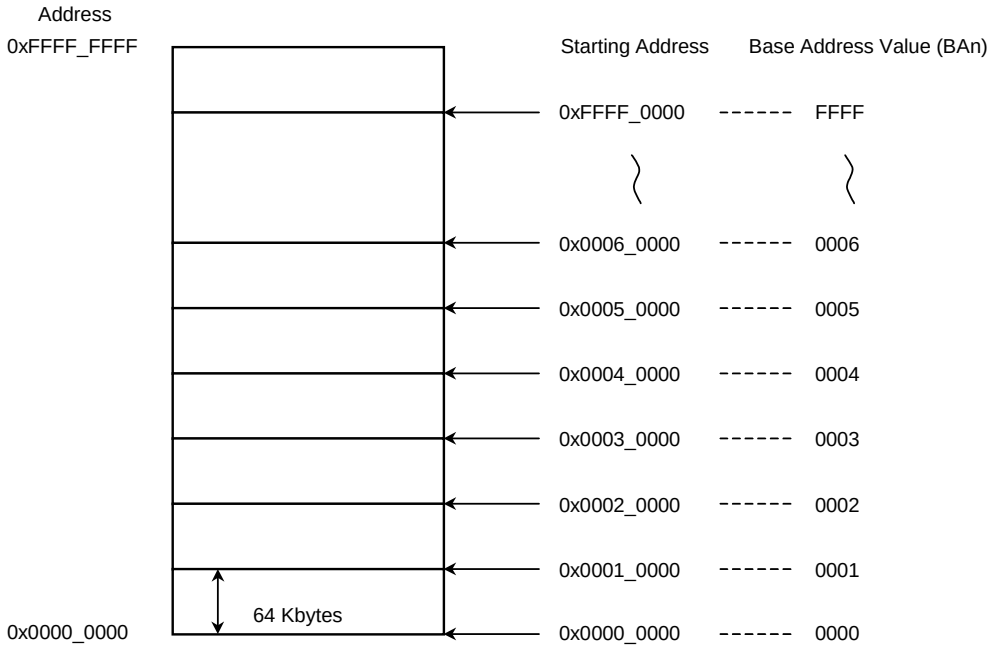
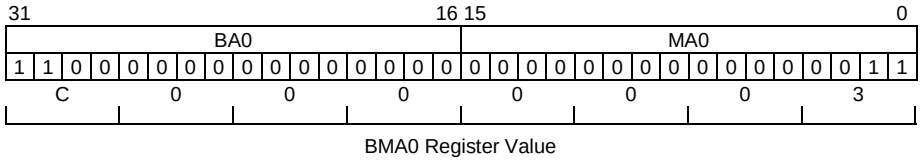


Figure 9.3 Relationships Between Starting Addresses and Base Address Register Values

9.1.2 Base Address and Address Mask Value Calculations

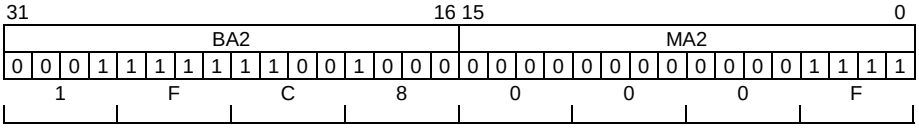
- Program the BMA0 register as follows to cause $\overline{CS0}$ to be asserted in the 64 Kbytes of address space starting at 0xC000_0000.



The BA0 field specifies the upper 16 bits of the starting address, or 0xC000. The MA0 field determines whether the A29–A14 bits of the address should be compared or masked. The A31 and A30 bits are always compared. Bits 15–10 of the MA0 field must be cleared so that the A29–A24 bits are always compared.

When the BMA0 register is programmed as shown above, the A31–A16 bits of the address are compared to the value of the BA0 field. Consequently, the 64-Kbyte address range between 0xC000_0000 and 0xC000_FFFF is defined as the CS0 space.

- Program the BMA2 register as follows to cause $\overline{CS2}$ to be asserted in the 512 Kbytes of address space starting at 0x1FC8_0000.

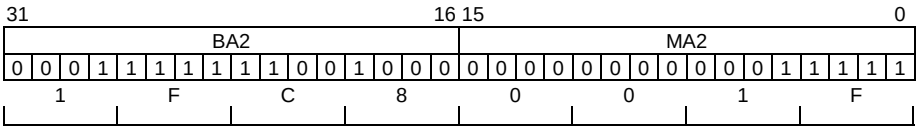


BMA2 Register Value

The BA2 field specifies the upper 16 bits of the starting address, or 0x1FC8. The MA2 field determines whether the A30–A15 bits of the address should be compared or masked. The A31 bit is always compared. Bits 15–9 of the MA2 field must be cleared so that the A30–A24 bits are always compared.

When the BMA2 register is programmed as shown above, the A31–A19 bits of the address are compared to the value of the BA2 field. Consequently, the 512-Kbyte address range between 0x1FC8_0000 and 0x1FCF_FFFF is defined as the CS2 space.

- Program the BMA2 register as follows to cause $\overline{CS2}$ to be asserted in the 1 Mbytes of address space starting at 0x1FC8_0000.



BMA2 Register Value

The BA2 field specifies the upper 16 bits of the starting address, or 0x1FC8. The MA2 field determines whether the A30–A15 bits of the address should be compared or masked. The A31 bit is always compared. Bits 15–9 of the MA2 field must be cleared so that the A30–A24 bits are always compared.

When the BMA2 register is programmed as shown above, the A31–A20 bits of the address are compared to the value of the BA2 field. Note, however, that the 512-Kbyte range between 0x1FC0_0000 and 0x1FC7_FFFF is reserved for the on-chip ROM. Consequently, the 512Kbyte address range between 0x1FC8_0000 and 0x1FCF_FFFF is defined as the CS2 space.

Note: The TMP1940CYAF does not assert any $\overline{CSn}$ signal in the following address ranges:

0x1FC_0000 through 0x1FC7_FFFF

0x4000_0000 through 0x4007_FFFF

0xFFFF_8000 through 0xFFFF_BFFF

Table 9.1 shows the programmable block sizes for CS0 to CS3. Even if the user has accidentally programmed more than one chip select line to the same area, only one chip select line is driven because of internal line priorities. CS0 has the highest priority, and CS3 the lowest.

Example:

The starting address of the CS0 space is progammed as 0xC000_0000 with a size of 16 Kbytes.
The starting address of the CS1 space is programmed as 0xC000_0000 with a size of 64 Kbytes.

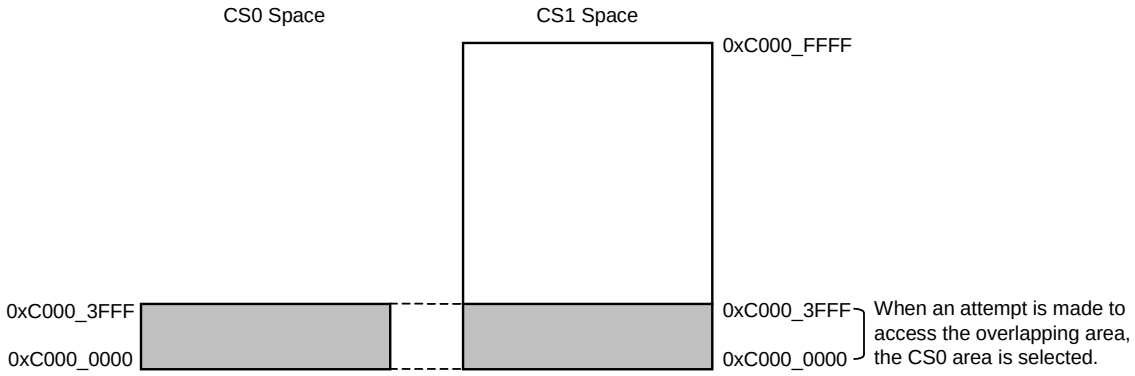


Table 9.1 Supported Block Sizes

CS Space	Size (bytes)										
	16 K	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M	16 M
CS0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CS1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CS2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CS3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

9.2 Chip Select/Wait Control Registers

The organizations of the Chip Select/Wait Control registers are shown in Figure 9.4 to Figure 9.5. Each of these registers consist of a chip select type field, a master enable bit, a data bus width bit, a wait state field and a dummy cycle field.

The B01CS register defines the CS0 and CS1 lines; the B23CS register defines the CS2 and CS3 lines; and the BEXCS register defines the access characteristics for the rest of the address locations.

Chip Select/Wait Control Registers

B01CS (0xFFFF_E480)

	7	6	5	4	3	2	1	0
Name	B0OM		—	B0BUS	B0W			
Read/Write	W		—	W				
Reset Value	0	0	—	0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
	15	14	13	12	11	10	9	8
Name	—	—	—	—	B0E	—	B0RCV	
Read/Write	—	—	—	—	W	—	W	
Reset Value	—	—	—	—	0	—	0	0
Function					CS0 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	
	23	22	21	20	19	18	17	16
Name	B1OM		—	B1BUS	B1W			
Read/Write	W		—	W				
Reset Value	0	0	—	0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
	31	30	29	28	27	26	25	24
Name	—	—	—	—	B1E	—	B1RCV	
Read/Write	—	—	—	—	W	—	W	
Reset Value	—	—	—	—	0	—	0	0
Function					CS1 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

Figure 9.4 Chip Select/Wait Control Registers

B23CS (0xFFFF_E484)

	7	6	5	4	3	2	1	0
Name	B2OM		—	B2BUS	B2W			
Read/Write	W		—	W				
Reset Value	0	0	—	0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{\text{WAIT}}$ pin Don't use any other value.			

	15	14	13	12	11	10	9	8
Name	—	—	—	—	B2E	B2M	B2RCV	
Read/Write	—	—	—	—	W			
Reset Value	—	—	—	—	1	0	0	0
Function					CS2 enable 0: Disable 1: Enable	CS2 space select 0: Whole 4-Gbyte space 1: CS space	Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

	23	22	21	20	19	18	17	16
Name	B3OM		—	B3BUS	B3W			
Read/Write	W		—	W				
Reset Value	0	0	—	0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{\text{WAIT}}$ pin Don't use any other value.			

	31	30	29	28	27	26	25	24
Name	—	—	—	—	B3E	—	B3RCV	
Read/Write	—	—	—	—	W	—	W	
Reset Value	—	—	—	—	0	—	0	0
Function					CS3 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

Figure 9.5 Chip Select/Wait Control Registers

BEXCS (0xFFFF_E488)

	7	6	5	4	3	2	1	0
Name	BEXOM			BEXBUS	BEXW			
Read/Write	W			W				
Reset Value	0	0		0	0	1	0	1
Function	Chip select output waveform 00: ROM/RAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Sets the number of Wait cycles 0000–0111: 0–7 wait states 1111: (1 + N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			

	15	14	13	12	11	10	9	8
Name							BEXRCV	
Read/Write							W	
Reset Value							0	0
Function							Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

Figure 9.6 Chip Select/Wait Control Registers

9.3 Application Example

Figure 9.7 shows an example usage of the TMP1940CYAF programmable chip selects. In this example, 128 Kbytes of ROM and 256 Kbytes of RAM are connected off-chip through a 16-bit data bus.

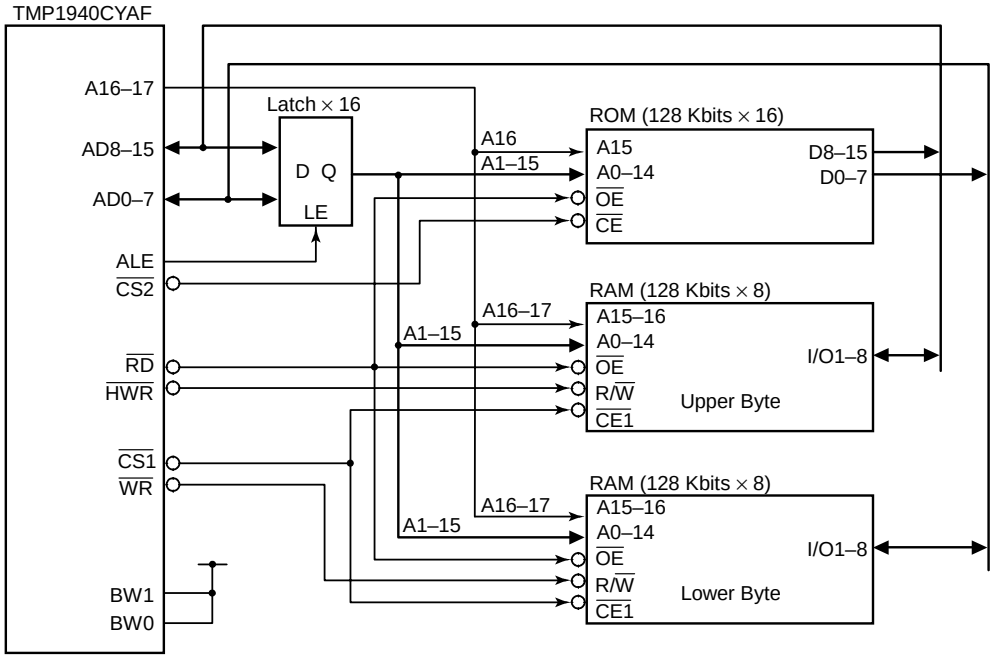


Figure 9.7 External Memory Connections (ROM Width = 16 bits, RAM Width = 16 bits)

Both $\overline{CS1}$ and $\overline{CS2}$ are shared with Port 4 pins. Upon reset, all Port 4 pins are configured as input port pins. To use them as chip select pins, set appropriate bits in the Port 4 Control (P4CR) register and the Port 4 Function (P4FC) register to 1.

10. DMA Controller (DMAC)

The TX1940CYAF contains a four-channel DMA controller.

10.1 Features

The TMP1940CYAF DMAC has the following features:

(1) Four independent DMA channels

(2) Two types of bus requests, with and without bus snooping

(3) Transfer requests:
Internal transfer requests: Software initiated
External transfer requests: Hardware signals from on-chip peripherals and external interrupt pins

(4) Dual-address mode

(5) Memory-to-memory, memory-to-I/O, and I/O-to-memory transfers

(6) Transfer width:

- Memory: 32-bit (8-bit and 16-bit memory devices are supported through the programming of the CS/Wait Controller.)
- I/O peripherals: 8-, 16-, and 32-bit

(7) Address pointers can increment, decrement or remain constant. The user can program the bit positions at which address incrementation or decrementation occurs.

(8) Fixed channel priority

10.2 Implementation

10.2.1 On-Chip DMAC Interface

Figure 10.1 shows how the DMAC is internally connected with the TX19 core processor and the Interrupt Controller (INTC).

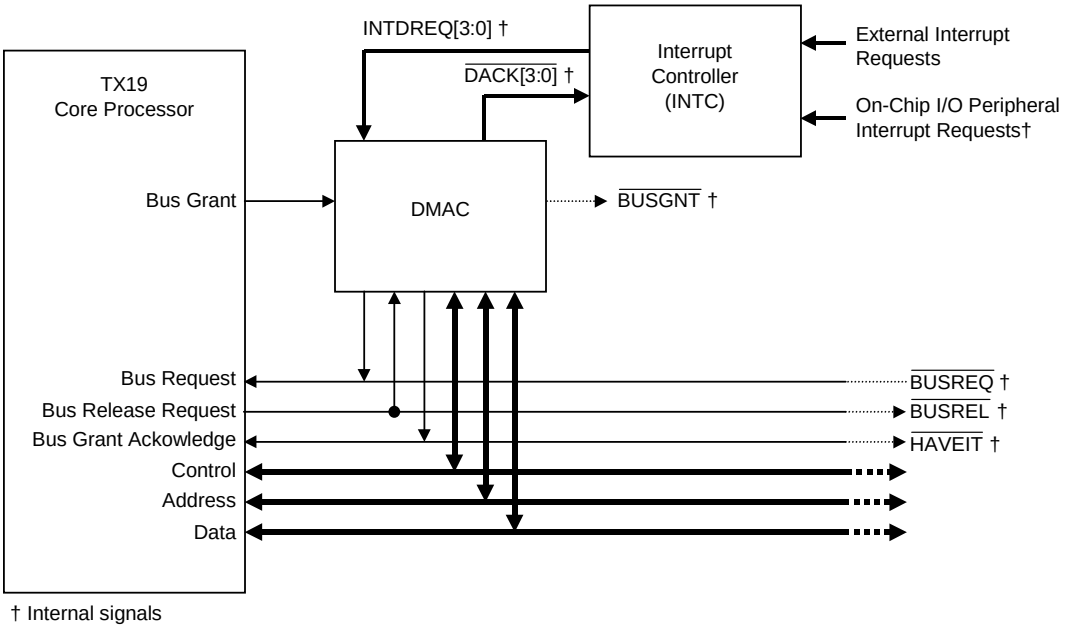


Figure 10.1 DMAC Connections within the TMP1940CYAF

The DMAC provides four independently programmable channels. With each DMA channel, there are two associated signals: a DMA request (INTDREQn) and a DMA acknowledge ($\overline{\text{DACKn}}$), where n is a channel number from 0 to 3. INTDREQn is an input to the DMAC coming from the INTC, and DACKn is an output signal from the DMAC going to the INTC.

Channel priority is fixed. Channel 0 has the highest priority, and Channel 3 has the lowest priority.

The TX19 core processor supports bus snooping. When snooping is enabled, the TX19 core processor grants the processor data bus to the DMAC, so that the DMAC can access the on-chip RAM and ROM connected to the processor. Snooping can be enabled and disabled under software control. The DMAC bus snooping is discussed in the next subsection in more details.

There are two bus request signals from the DMAC going to the TX19 core processor, SREQ and GREQ. GREQ is a bus request without snooping. SREQ is a bus request with snooping.

Note: DMA channel priority exists only among those using the same type of bus request signal (SREQ or GREQ). For example, once a given DMA channel has acquired bus mastership using SREQ, no other DMA channel can assume bus mastership using GREQ until the ongoing DMA transaction is completed.

10.2.2 DMAC Block

The DMAC block diagram is shown in Figure 10.2.

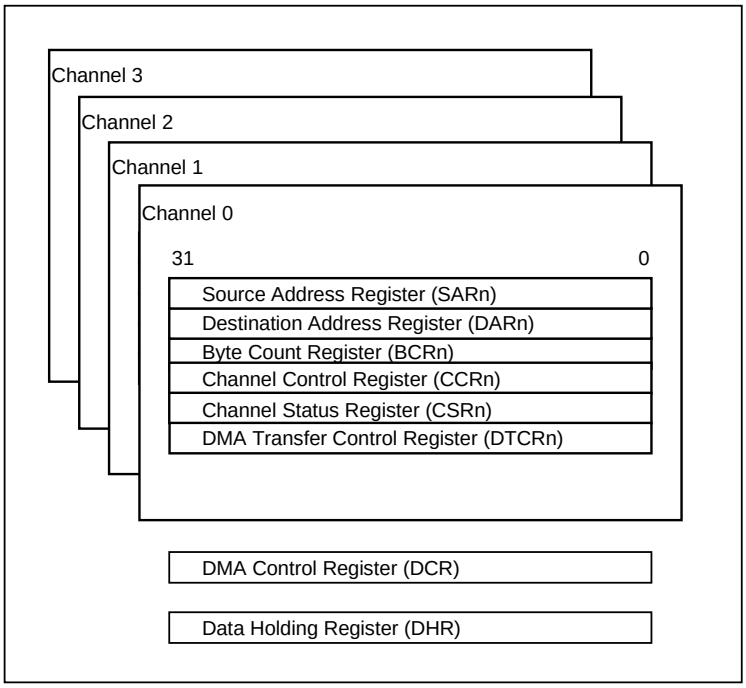


Figure 10.2 DMAC Block Diagram

10.2.3 Bus Snooping

The TX19 core processor supports snoop operations.

If snooping is enabled, the TX19 core processor grants the processor data bus to the DMAC. Because the DMAC takes control of the processor data bus, the TX19 stops operating during snoop operations until the DMAC relinquishes the bus to the processor. Snooping allows the DMAC to access the on-chip RAM and ROM, and thus to use them as a DMA source or destination device.

The DMAC allows the enabling and disabling of the snooping function by software.

If snooping is disabled, the DMAC can not access the on-chip RAM and ROM. However, regardless of whether snooping is enabled or disabled, the DMAC assumes mastership of the TMP1940CYAF on-chip bus (G-Bus) during DMA transfers. Therefore, as long as DMA transfers are in progress, the TX19 core processor can not access memory or I/O peripherals via the G-Bus; any attempt to do so causes the processor pipeline to stall.

Note: If snooping is disabled, the TX19 core processor does not grant mastership of the processor data bus to the DMAC. Therefore, if the on-chip RAM or ROM is specified as a source or destination for DMA transfers, a DMA acknowledge signal will never be returned, causing bus lockup.

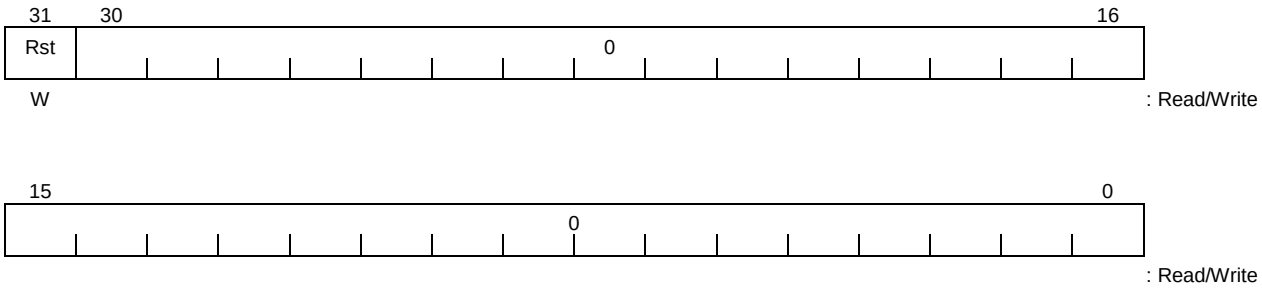
10.3 Register Description

The DMAC has twenty-six 32-bit registers. The DMAC register map is shown in Table 10.1.

Table 10.1 DMAC Registers

Address	Symbol	Register Name
0xFFFF_E200	CCR0	Channel Control Register (Ch. 0)
0xFFFF_E204	CSR0	Channel Status Register (Ch. 0)
0xFFFF_E208	SAR0	Source Address Register (Ch. 0)
0xFFFF_E20C	DAR0	Destination Address Register (Ch. 0)
0xFFFF_E210	BCR0	Byte Count Register (Ch. 0)
0xFFFF_E218	DTCR0	DMA Transfer Control Register (Ch. 0)
0xFFFF_E220	CCR1	Channel Control Register (Ch. 1)
0xFFFF_E224	CSR1	Channel Status Register (Ch. 1)
0xFFFF_E228	SAR1	Source Address Register (Ch. 1)
0xFFFF_E22C	DAR1	Destination Address Register (Ch. 1)
0xFFFF_E230	BCR1	Byte Count Register (Ch. 1)
0xFFFF_E238	DTCR1	DMA Transfer Control Register (Ch. 1)
0xFFFF_E240	CCR2	Channel Control Register (Ch. 2)
0xFFFF_E244	CSR2	Channel Status Register (Ch. 2)
0xFFFF_E248	SAR2	Source Address Register (Ch. 2)
0xFFFF_E24C	DAR2	Destination Address Register (Ch. 2)
0xFFFF_E250	BCR2	Byte Count Register (Ch. 2)
0xFFFF_E258	DTCR2	DMA Transfer Control Register (Ch. 2)
0xFFFF_E260	CCR3	Channel Control Register (Ch. 3)
0xFFFF_E264	CSR3	Channel Status Register (Ch. 3)
0xFFFF_E268	SAR3	Source Address Register (Ch. 3)
0xFFFF_E26C	DAR3	Destination Address Register (ch. 3)
0xFFFF_E270	BCR3	Byte Count Register (Ch. 3)
0xFFFF_E278	DTCR3	DMA Transfer Control Register (Ch. 3)
0xFFFF_E280	DCR	DMA Control Register (All channels)
0xFFFF_E28C	DHR	Data Holding Register (All channels)

10.3.1 DMA Control Register (DCR)



Bits	Mnemonic	Field Name	Description
31	Rst	Reset	Performs a software reset of the DMAC. When the Rst bit is set to 1, all the DMAC internal registers are initialized to their reset values. Any transfer requests are removed and all the four DMA channels are put in Idle state. 0: Don't-care 1: Resets the DMAC.

Note 1: When the snoop request is disabled (CCRn.SReq=0), a software reset of the DMAC must be performed in the following sequence:

1. **Disable interrupts.**
2. **Execute NOP four times.**
3. **Perform a software reset.**
4. **Perform a software reset again.**
5. **Re-enable interrupts.**

Execute steps 3 and 4 consecutively.

Note 2: If the software reset command is written to the DCR register immediately after the completion of the last transfer cycle of a DMA transaction, the DMA-done interrupt will not be cleared. In this case, the software reset only initializes channel registers, etc.

Note 3: Don't issue a software reset command to the DCR register via a DMA transfer.

Figure 10.3 DMA Control Register (DCR)

10.3.2 Channel Control Registers (CCRn)

31	30					25	24	23	22	21	20	19	18	17	16	
Str	0						—	NIEn	AblEn	—	—	—	—	Big	—	
W							W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	: Read/Write
								1	1	1	0	0	0	1	0	: Reset Value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
—	ExR	PosE	Lev	Sreq	ReIEN	SIO	SAC		DIO	DAC		TrSiz		DPS		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W		R/W		R/W		: Read/Write
0	0	0	0	0	0	0	00		0	00		00		00		: Reset Value

Bits	Mnemonic	Field Name	Description
31	Str	Channel Start	Reset value: — Enables a DMA channel. Setting this bit puts the DMA channel in Ready state. DMA transfer starts as soon as a transfer request is received. Only a write of 1 is valid, and a write of 0 has no effect on this bit. A 0 is returned on read. 1: Enables a DMA channel.
24	—	Reserved	This bit is reserved and must be written as 0.
23	NIEn	Normal Completion Interrupt Enable	Reset value = 1 1: Enables an interrupt when the channel finishes a transfer without an error condition. 0: Does not enable an interrupt when the channel finishes a transfer without an error condition.
22	AbIEn	Abnormal Termination Interrupt Enable	Reset value = 1 1: Enables an interrupt when the channel encounters a transfer error. 0: Does not enable an interrupt when the channel encounters a transfer error.
21	—	Reserved	This bit is reserved and must be written as 0.
20	—	Reserved	This bit is reserved and must be written as 0.
19	—	Reserved	This bit is reserved and must be written as 0.
18	—	Reserved	This bit is reserved and must be written as 0.
17	Big	Big-Endian	Reset value = 1 1: The DMA channel operates in big-endian mode. 0: The DMA channel operates in little-endian mode. In the TMP1940CYAF, this bit must be cleared to 0.
16	—	Reserved	This bit is reserved and must be written as 0.
15	—	Reserved	This bit is reserved and must be written as 0.
14	ExR	External Request Mode	Reset value = 0 Selects a transfer request mode. 1: External transfer requests (interrupt-driven) 0: Internal transfer requests (software-initiated)
13	PosE	Positive Edge	Reset value = 0 Defines the polarity of the internal DMA request signal (INTDREQn) for the channel. This bit is valid for external transfer requests (i.e., when ExR=1), and has no effect on internal transfer requests (i.e., when ExR=0). In the TMP1940CYAF, the PosE bit must be cleared, and the Lev bit must be set.
12	Lev	Level Mode	Reset value = 0 Specifies whether external transfer requests are level-senstiive or edge-triggered. This bit is valid for external transfer requests (i.e., when ExR=1), and has no effect on internal transfer requests (i.e., when ExR=0). In the TMP1940CYAF, this bit must be set.

Figure 10.4 Channel Control Registers (CCRn) (1/2)

Bit	Mnemonic	Field Name	Description
11	SReq	Snoop Request	Reset value = 0 Controls whether or not to request bus mastership with snooping. If set, the TX19 core processor's snoop function becomes valid, allowing the DMAC to use the processor's data bus. If cleared, the snoop function is disabled. 1: The snoop function is enabled (i.e., SREQ is used as a bus request signal). 0: The snoop function is disabled (i.e., GREQ is used as a bus request signal).
10	RelEn	Bus Release Request Enable	Reset value = 0 Controls whether or not to respond to the bus release request signal from the TX19 core processor. This bit is valid when the DMAC uses GREQ as a bus request signal. This bit has no meaning or effect when the DMAC uses SREQ as a bus request signal because, in that case, the TX19 core processor does not have the capability to generate a bus release request signal. 1: The DMAC will respond to the bus release request signal from the TX19 core processor, if it has control of the bus. The DMAC will relinquish the bus when the current DMA bus cycle completes. 0: The DMAC will ignore the bus release request signal from the TX19 core processor.
9	SIO	I/O Source	Reset value = 0 Specifies the type of the source device. 1: I/O device 0: Memory
8:7	SAC	Source Address Count	Reset value = 00 Selects the manner in which the source address changes after each cycle. 1x: Fixed (remains unchanged) 01: Decremented 00: Incremented
6	DIO	I/O Destination	Reset value = 0 Specifies the type of the destination device. 1: I/O device 0: Memory
5:4	DAC	Destination Address Count	Reset value = 00 Selects the manner in which the destination address changes after each cycle. 1x: Fixed (remains unchanged) 01: Decremented 00: Incremented
3:2	TrSiz	Transfer Size	Reset value = 00 Specifies the amount of data to be transferred in response to a DMA request. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)
1:0	DPS	Device Port Size	Reset value = 00 Specifies the port size of a source or destination I/O device. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)

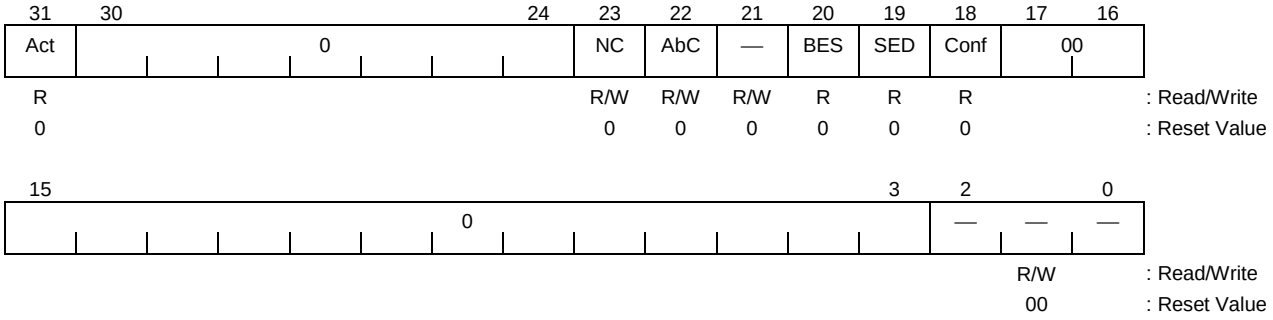
Figure 10.4 Channel Control Registers (CCRn) (2/2)

Note 1: The DPS field has no meaning or effect on memory-to-memory transfers.

Note 2: To access on-chip peripherals, the transfer size (TrSiz) must be equal to the device port size (DPS).

Note 3: The CCRn register must be programmed before placing the DMAC in Ready state.

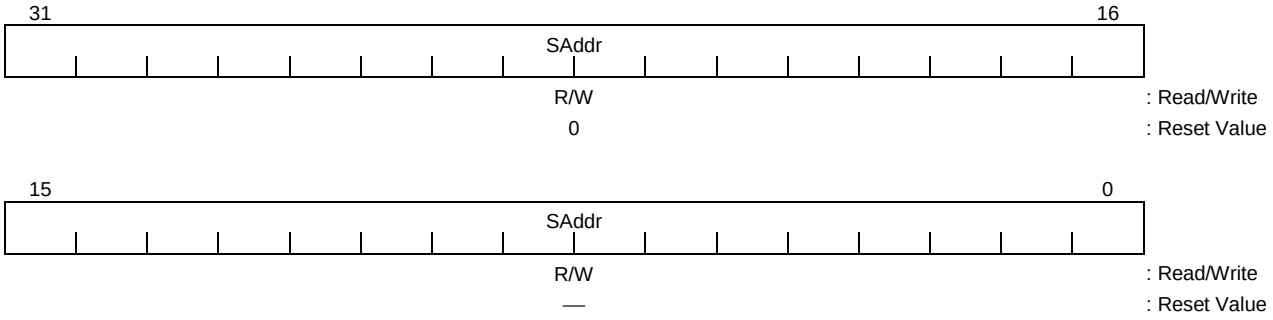
10.3.3 Channel Status Registers (CSRn)



Bit	Mnemonic	Field Name	Description
31	Act	Channel Active	Reset value = 0 Indicates whether or not the DMA channel is in Ready state. 1: The DMA channel is in Ready state. 0: The DMA channel is not in Ready state.
23	NC	Normal Completion	Reset value = 0 If set, the DMA channel has terminated by normal completion. If the NIEn bit in the CCRn is set, an interrupt is generated. The NC bit is cleared by writing a 0 to it. Clearing the NC bit causes the interrupt to be cleared. The NC bit must be cleared prior to starting the next transfer. An attempt to set the Str bit in the CCRn when NC=1 will cause an error. A write of 1 has no effect on this bit. 1: The DMA channel has terminated by normal completion. 0: The DMA channel has not terminated by normal completion.
22	AbC	Abnormal Completion	Reset value = 0 If set, the DMA channel has terminated with an error. If the AbIEn bit in the CCRn is set, an interrupt is generated. The AbC bit is cleared by writing a 0 to it. Clearing the AbC bit causes the interrupt to be cleared. The AbC bit must be cleared prior to starting the next transfer. An attempt to set the Str bit in the CCRn when AbC=1 will cause an error. A write of 1 has no effect on this bit. 1: The DMA channel has terminated with an error. 0: The DMA channel has not terminated with an error.
21	—	Reserved	This bit is reserved and must be written as 0.
20	BES	Source Bus Error	Reset value = 0 1: A bus error has occurred during the source read cycle. 0: A bus error has not occurred during the source read cycle.
19	BED	Destination Bus Error	Reset value = 0 1: A bus error has occurred during the destination write cycle. 0: A bus error has not occurred during the destination write cycle.
18	Conf	Configuration Error	Reset value = 0 1: A configuration error is present. 0: No configuration error is present.
2:0	—	Reserved	These bits are reserved and must be written as 0s.

Figure 10.5 Channel Status Registers (CSRn)

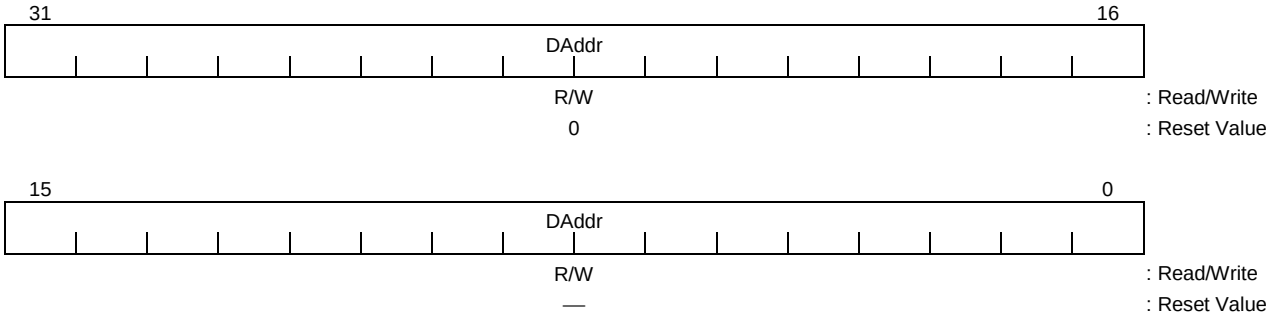
10.3.4 Source Address Registers (SARn)



Bit	Mnemonic	Field Name	Description
31:0	SAddr	Source Address	Reset value: — Contains the physical address of the source device. The address changes as programmed in the SAC and TrSiz fields in the CCRn and the SACM field in the DTCRn.

Figure 10.6 Source Address Registers (SARn)

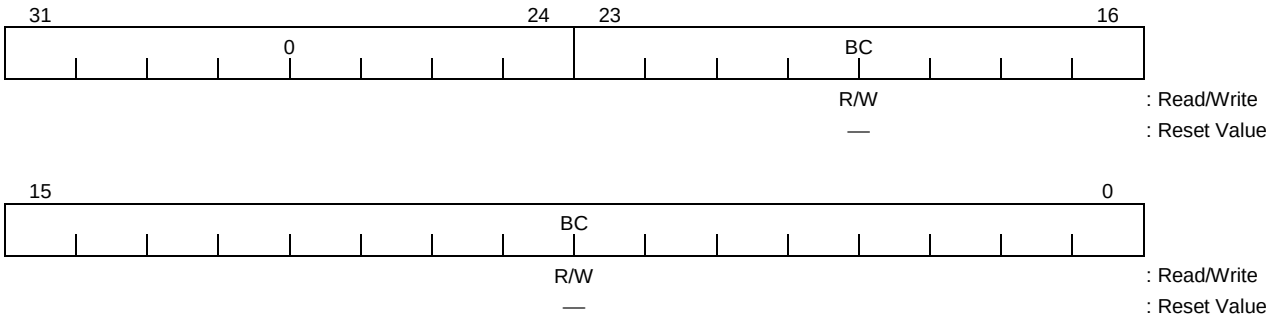
10.3.5 Destination Address Registers (DARn)



Bit	Mnemonic	Field Name	Description
31:0	DAddr	Destination Address	Reset value: — Contains the physical address of the destination device. The address changes as programmed in the DAC and TrSiz fields in the CCRn and the DACM field in the DTCRn.

Figure 10.7 Destination Address Registers (DARn)

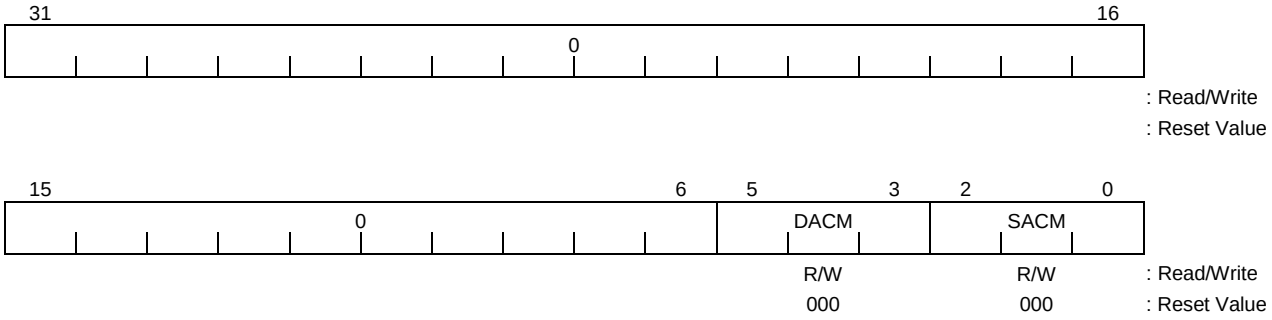
10.3.6 Byte Count Registers (BCRn)



Bit	Mnemonic	Field Name	Description
23:0	BC	Byte Count	Reset value: — Contains the number of bytes left to transfer on a DMA channel. The count is decremented by 1, 2 or 4 (as determined by the TrSiz field in the CCRn register) for each successful transfer.

Figure 10.8 Byte Count Registers (BCRn)

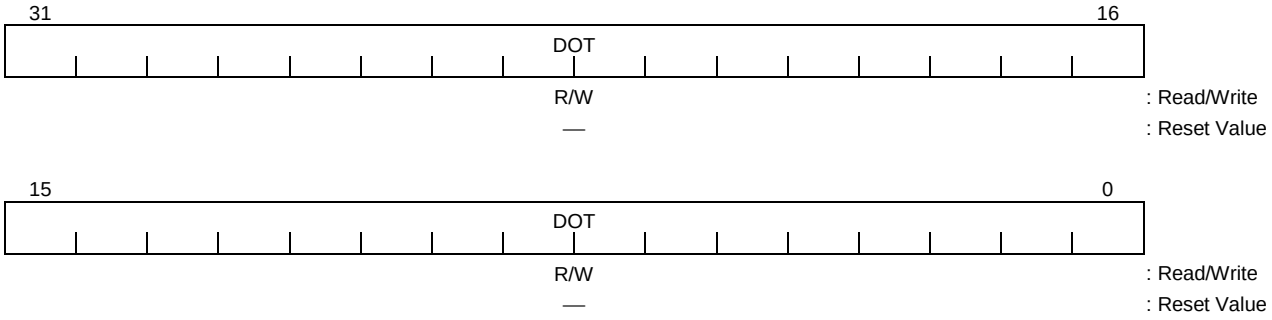
10.3.7 DMA Transfer Control Registers (DTCRn)



Bit	Mnemonic	Field Name	Description
5:3	DACM	Destination Address Count Mode	Selects the manner in which the destination address is incremented or decremented. 000: Counting begins with bit 0 of the DARn. 001: Counting begins with bit 4 of the DARn. 010: Counting begins with bit 8 of the DARn. 011: Counting begins with bit 12 of the DARn. 100: Counting begins with bit 16 of the DARn. 101: Reserved 110: Reserved 111: Reserved
2:0	SACM	Source Address Count Mode	Selects the manner in which the source address is incremented or decremented. 000: Counting begins with bit 0 of the SARn. 001: Counting begins with bit 4 of the SARn. 010: Counting begins with bit 8 of the SARn. 011: Counting begins with bit 12 of the SARn. 100: Counting begins with bit 16 of the SARn. 101: Reserved 110: Reserved 111: Reserved

Figure 10.9 DMA Transfer Control Registers (DTCRn)

10.3.8 Data Holding Register (DHR)



Bit	Mnemonic	Field Name	Description
31:0	DOT	Data on Transfer	Reset value: — Contains data read from the source address during a dual-address operation.

Figure 10.10 Data Holding Register (DHR)

10.4 Operation

This section describes the operation of the DMAC.

10.4.1 Overview

The DMAC is a high-speed 32-bit DMA controller used to quickly move large blocks of data between I/O peripherals and memory without intervention of the TX19 core processor.

(1) Devices Supported for the Source and Destination

The DMAC handles data transfers from memory to memory and between memory and I/O peripherals. The device from which data is transferred is referred to as a source device, and the device to which data is transferred is referred to as a destination device. Both memory and I/O peripherals can be a source or destination device. The DMAC supports data transfers from memory to I/O peripherals, from I/O peripherals to memory, and from memory to memory, but not from I/O peripherals to I/O peripherals.

DMA protocols for memory and I/O peripherals differ in that when accessing an I/O peripheral, the DMAC asserts the $\overline{\text{DACKn}}$ (n = channel number) signal to indicate that data is being transferred in response to a previous transfer request. Because each DMA channel has only one $\overline{\text{DACKn}}$ signal, the DMAC can not handle data transfers between two I/O peripherals.

Interrupt requests can be programmed to be a trigger to initiate a DMA process instead of requesting an interrupt to the TX19 core processor. If so programmed, the Interrupt Controller (INTC) forwards a DMA request to the DMAC (see 10.4.6, *Interrupts*). The DMA request coming from the INTC is cleared when the INTC receives a $\overline{\text{DACKn}}$ from the DMAC. Consequently, a DMA request for a transfer to/from an I/O peripheral is cleared after each DMA bus cycle (i.e., every time the number of bytes programmed into the CCRn.TrSiz field is transferred). On the other hand, during memory-to-memory transfer, the $\overline{\text{DACKn}}$ signal is not asserted until the byte count register (BCRn) reaches zero. Therefore, memory-to-memory transfer can continuously move large blocks of data in response to a single DMA request.

For example, data transfers between the TMP1940CYAF on-chip peripheral and on- or off-chip memory is discontinued after every DMA bus cycle. Nonetheless, until the BCRn register reaches zero, the DMAC remains in Ready state to wait for the next transfer request.

(2) Exchanging Bus Mastership (Bus Arbitration)

In response to a DMA request, the DMAC issues a bus request to the TX19 core processor. When the DMAC receives a bus grant signal from the TX19 core processor, it assumes bus mastership to service the DMA request. There are two bus request signals from the DMAC going to the TX19 core processor. One is a bus request without snooping (GREQ), and the other is a bus request with snooping (SREQ). The SReq bit in the CCRn register is used to select a bus request signal to use for each DMA channel.

While the DMAC has control of the bus, the TX19 core processor may issue a bus release request to the DMAC. The RelEn bit of the CCRn register controls whether to honor this request on a channel-by-channel basis. This setting has a meaning only when a DMA channel uses GREQ (i.e., a bus request without snooping). It has no meaning or effect when a DMA channel uses SREQ (i.e., a bus request with snooping) because, in this case, the TX19 core processor does not have the capability to generate a bus release request.

The DMAC relinquishes the bus to the TX19 core processor when there is no pending DMA request to be serviced.

Note 1: The NMI interrupt is left pending while the DMAC has control of the bus.

Note 2: Don't place the TMP1940CYAF in Halt powerdown mode while the DMAC is operating.

(3) Transfer Request Generation

Each DMA channel supports two types of request generation methods: internal and external.

Internal requests are those generated within the DMAC. The DMA channel is started as soon as the Str bit in the CCRn register is set. The channel immediately requests the bus and begins transferring data.

If a channel is programmed for external request and the Str bit is set, the transfer request signal (INTDREQn) must be asserted by the Interrupt Controller before the channel requests the bus and begins a transfer. Although INTDREQn can be programmed for level/edge sensitivity, the TMP1940CYAF requires INTDREQn to be low-level sensitive.

(4) Data Transfer Modes

The TMP1940CYAF DMAC supports dual-address transfers, but not single-address transfers.

The dual-address mode allows data to be transferred from memory to memory and between memory and an I/O peripheral. In this mode, the DMAC explicitly addresses both the source and destination devices. The DMAC also generates a $\overline{\text{DACK}}$n signal when accessing an I/O peripheral.

In dual-address mode, a transfer takes place in two DMA bus cycles: a source read cycle and a destination write cycle. In the source read cycle, the data being transferred is read from the source address and put into the DMAC internal Data Holding Register (DHR). In the destination write cycle, the DMAC writes data in the DHR to a destination address.

(5) DMA Channel Operation

The DMAC has four independent DMA channels 0 to 3. Setting the Start (Str) bit in the CCRn (n = 0–3) enables a particular channel and puts it in Ready state.

When a DMA request is detected in any of the channels in Ready state, the DMAC arbitrates for the bus and begins a transfer. When no DMA request is pending, the DMAC relinquishes the bus to the TX19 core processor and returns to Ready state. The channel can terminate by normal completion or from an error of a bus cycle. When a channel terminates, that channel is put in Idle state. Interrupts can be generated by error termination or by normal channel termination.

Figure 10.11shows a general state transitions of a DMA channel.

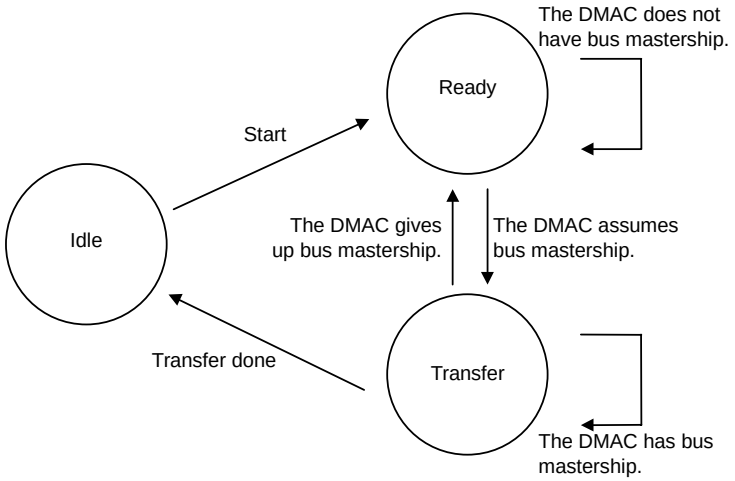


Figure 10.11 DMA Channel State Transitions

(6) Summary of Transfer Modes

The DMAC can perform data transfers as follows according to the combination of mode settings.

Transfer Request	Edge/Level	Address Mode	Data Flow
Internal	—	Dual	Memory-to-memory
External	Low Level	Dual	Memory-to-memory
			Memory-to-I/O
			I/O-to-memory

(7) Address Change Options

Address pointers can increment, decrement or remain constant. The SAC and DAC fields in the CCRn respectively select address change directions for the Source Address Register (SARn) and the Destination Address Register (DARn). While memory addresses can be programmed to increment, decrement or remain constant, I/O addresses must be programmed to remain constant.

The SACM and DACM fields in the DTCRn provide options to program bit positions at which the source and destination addresses are incremented or decremented after each transfer. The bit position can be bit 0, 4, 8, 12 or 16. Use of bit 0 is the regular increment/decrement mode in which the address changes by 1, 2 or 4, according to the source or destination size. Two examples of how other increment/decrement modes affect address changes are show below.

Example 1: When address bit 0 is selected in the SACM field and address bit 4 is selected in the DACM field

SAC: Programmed to increment the source address
DAC: Programmed to increment the destination address
TrSiz: Programmed to a transfer size of 32 bits
Source address: 0xA000_1000
Destination address: 0xB000_0000
SACM: 000 → Bit 0 is the source address bit at which address incrementation occurs.
DACM: 001 → Bit 4 is the destination address bit at which address incrementation occurs.

	Source	Destination
1st transfer	0xA000_1000	0xB000_0000
2nd transfer	0xA000_1004	0xB000_0010
3rd transfer	0xA000_1008	0xB000_0020
4th transfer	0xA000_100C	0xB000_0030
	…	…

Example 2: When address bit 8 is selected in the SACM field and address bit 0 is selected in the DACM field

SAC: Programmed to decrement the address
DAC: Programmed to decrement the address
TrSiz: Programmed to a transfer size of 16 bits
Source address: 0xA000_1000
Destination address: 0xB000_0000
SACM: 010 → Bit 8 is the source address bit at which address decrementation occurs.
DACM: 000 → Bit 0 is the destination address bit at which address decrementation occurs.

	Source	Destination
1st transfer	0xA000_1000	0xB000_0000
2nd transfer	0x9FFF_FF00	0xAFFF_FFFE
3rd transfer	0x9FFF_FE00	0xAFFF_FFFC
4th transfer	0x9FFF_FD00	0xAFFF_FFFA
	…	…

10.4.2 Transfer Request Generation

A DMA request must be issued for the DMAC to initiate a data transfer. Each DMA channel in the DMAC supports two types of request generation method: internal and external. In either request generation mode, once a DMA channel is started, a DMA request causes the DMAC to arbitrate for the bus and begin transferring data.

- Internal Request Generation

  A channel is programmed for internal request by clearing the ExR bit in the CCRn. In internal request generation mode, a transfer request is generated as soon as the Str bit in the CCRn is set.

  An internally generated request keeps a transfer request pending until the transfer is complete. If no transition to a higher-priority DMA channel or a bus master occurs, the channel will use 100% of the available bus bandwidth to transfer all data continuously.

  Internally generated requests support only memory-to-memory transfer.

- External Request Generation

  A channel is programmed for external request by setting the ExR bit in the CCRn. In external request generation mode, setting the Str bit in the CCRn puts the channel in Ready state. While in Ready state, assertion of the INTDREQn signal (where n is the channel number) coming from the Interrupt Controller (INTC) causes a transfer request to be generated. Externally generated requests support data transfers from memory to memory and between memory and an I/O peripheral.

  INTDREQn can be programmed for either edge or level sensitivity through the PosE bit in the CCRn. However, in the TMP1940CYAF, INTDREQn is an active-low, level-sensitive signal. Therefore, the PosE bit must be cleared to 0.

  The transfer size, i.e., the amount of data to be transferred in response to a transfer request, is programmed in the TrSize field in the CCRn. The transfer size can be 32 bits, 16 bits or 8 bits.

  A transfer request is removed by assertion of the $\overline{\text{DACKn}}$ signal (where n is the channel number). $\overline{\text{DACKn}}$ is asserted: 1) when an I/O peripheral bus cycle has completed and 2) when the Byte Count Register (BCRn) has reached zero in memory-to-memory transfer. Consequently, a memory-to-I/O or I/O-to-memory transfer request terminates after one DMA bus cycle completes, whereas memory-to-memory transfer can continuously move large blocks of data in response to a single DMA request.

  The INTC might clear INTDREQn before the DMAC accepts it and begins a data transfer. It must be noted that, even if that happens, a DMA bus cycle might be executed after the interrupt request has been cleared.

10.4.3 DMA Address Modes

The TMP1940CYAF supports only dual-address mode in which both the source and destination devices are explicitly addressed.

In dual-address mode, two bus transfers occur: a read from a source device and a write to the destination device. In the source read cycle, data is read from the source address and placed in the DMAC internal Data Holding Register (DHR). Then, in the destination write cycle, the data held in the DHR is written to the destination address.

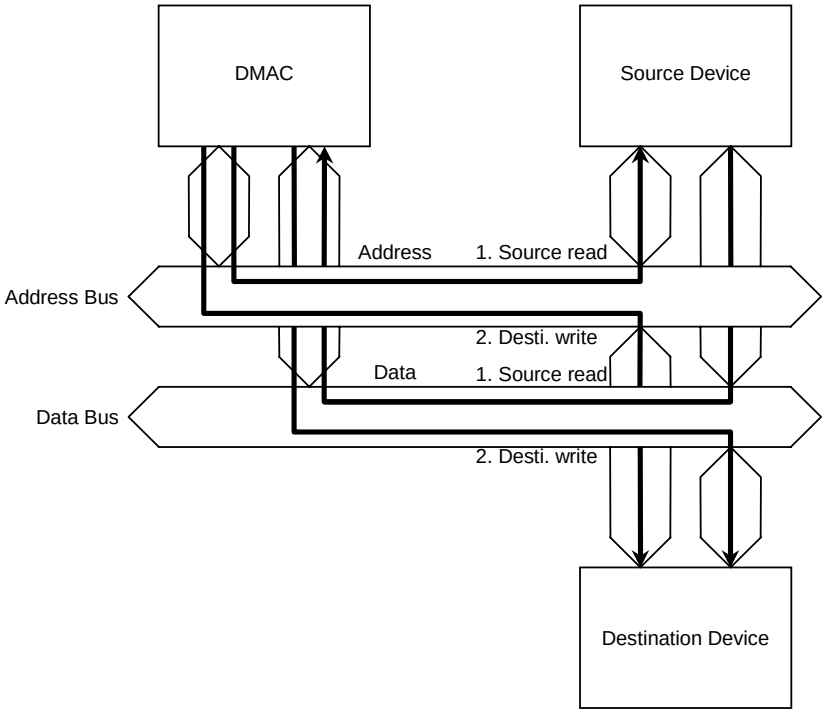


Figure 10.12 Dual-Address Transfer Mode

The transfer size programmed into the CCRn.TrSiz field determines the amount of data that is transferred from a source device to a destination device in response to a DMA request. The transfer size can be 32 bits, 16 bits or 8 bits.

The internal DHR is a 32-bit register that serves as a buffer for the data being transferred from a source device to a destination device during dual-address mode.

Memory accesses occur in a manner to fulfill the CCRn.TrSiz setting. Remember that the CS/Wait Controller supports either 16-bit or 8-bit bus accesses for external memory. If the DMA transfer size is programmed to 32 bits in CCRn.TrSiz, DMA read and write cycles each take up to four bus cycles to complete. A 16-bit data bus, as programmed in the CS/Wait Controller, requires two independent bus cycles to complete a 32-bit transfer. Likewise, an 8-bit data bus requires four independent bus cycles to complete a 32-bit transfer.

Memory-to-I/O and I/O-to-memory DMA transfers are governed by the setting of the CCRn.DPS field in addition to the setting of CCRn.TrSiz. The DPS field defines the port size of a source or destination I/O peripheral. The I/O port size can be 32 bits, 16 bits or 8 bits.

If the transfer size is equal to the I/O port size, an I/O access takes a single read or single write cycle. If the I/O port size is less than the programmed transfer size, the internal 32-bit DHR serves as a buffer for the data being transferred. For example, assume that the transfer size is programmed to 32 bits. If the source I/O port size is 8 bits and the destination memory width is 32 bits, then four 8-bit read cycles occur, followed by a 32-bit write cycle. (If the destination is an external memory with a 16-bit data bus,

the write cycle takes two bus cycles.) The 32 bits of data are buffered in the DHR until the destination write cycle occurs.

Source and destination addresses can be programmed to increment or decrement after each transfer. The SARn and DARn change, if so programmed, after each data transfer, depending on the transfer size, i.e., the programmed TrSiz value. The BRCn is decremented by TrSiz for each data transfer.

It is forbidden to program the device port size (DPS) to a value greater than the DMA transfer size (TrSiz).

The relationships between TrSiz and DPS are summarized below.

Table 10.2 DMA Transfer Sizes and Device Port Sizes (in Dual-Address Mode)

TrSiz	DPS	# of I/O Bus Cycles
0x (32 bits)	0x (32 bits)	1
0x (32 bits)	10 (16 bits)	2
0x (32 bits)	11 (8 bits)	4
10 (16 bits)	0x (32 bits)	Don't use.
10 (16 bits)	10 (16 bits)	1
10 (16 bits)	11 (8 bits)	2
11 (8 bits)	0x (32 bits)	Don't use.
11 (8 bits)	10 (16 bits)	Don't use.
11 (8 bits)	11 (8 bits)	1

Note: The DMAC does not incremnt or decrement the address for I/O peripherals. Therefore, if, for example, TrSiz is programmed to 16 bits and DPS is programmed to 8 bits, both the first and second bus cycles access the lower eight bits of the I/O data bus.

10.4.4 DMA Channel Operation

Each DMA channel is started by setting the Str bit in the CCRn to 1. Once started, the DMAC checks the channel setups for configuration errors. If no configuration error is present, the channel enters Ready state.

When a DMA request is detected while in Ready state, the DMAC arbitrates for the bus and begins transferring data.

The channel can terminate by normal completion or from an error.

(1) Channel Startup

A DMA channel is started by setting the Str bit in the CCRn.

Once started, the DMAC checks the channel setups for configuration errors. If a configuration error is detected, the channel terminates abnormally. If no configuration error is present, the channel enters Ready state. Once a channel enters Ready state, the Act bit in the CSRn is set to 1.

If the channel is programmed for internal request, the channel requests the bus and starts transferring data immediately. If the channel is programmed for external request, INTDREQn must be asserted before the channel requests the bus.

(2) Channel Termination

A DMA channel can terminate by normal completion or from an error. The status of a DMA operation can be determined by reading the CSRn.

A channel terminates abnormally when an attempt is made to set the Str bit in the CCRn when the NC or AbC bit in the CSRn is set.

Normal Termination

A DMA channel terminates by normal completion in the following case. Normal completion always occurs at the boundary of transfers programmed into the CCRn.TrSize field.

- Data transfers have terminated, with the BCRn decremented to 0.

Abnormal Termination

The paragraphs that follow summarize the cases in which a DMA channel terminates from an error.

- Configuration errors

  A configuration error results when the channel initialization contains inconsistencies or errors. A configuration error is reported before any data transfer takes place; therefore, in case of a configuration error, the SARn, DARn and BCRn remain unaltered. When a DMA channel has terminated from a configuration error, the AbC and Conf bits in the CSRn are set. A configuration error occurs for the following cases:

  – Both the CCRn.SIO and CCRn.DIO bits are set.
  – The CCRn.Str bit is set when the NC or AbC bit in the CSRn is set.
  – The BCRn contains a value that is not an integer multiple of the transfer size programmed into the CCRn.TrSiz field.
  – The SARn or DARn contains a value that is not an integer multiple of the transfer size programmed into the CCRn.TrSiz field.
  – The CCRn.TrSiz and CCRn.DPS fields contain illegal combinations.
  – The CCRn.Str bit is set when the the BCRn contains a value of zero.

- Bus errors

  When a DMA channel has terminated from a bus error, the AbC bit and the BES or BED bit in the CSRn is set.

  – A bus error has been reported during a source read or destination write cycle.

Note: The contents of the BCRn, SARn and DARn are not guaranteed when a channel has terminated due to a bus error. Chapter 19 lists the reserved addresses that, if accessed, cause a bus error.

10.4.5 DMA Channel Priority

The DMAC provides a fixed priority for the four channels, with channel 0 always having the highest priority and channel 3 the lowest. For example, when transfer requests occur on channels 0 and 1 simultaneously, the channel 0 request is serviced first. The channel 1 request is left pending. So that the channel 1 request is serviced, it must be maintained until data transfer completes on channel 0.

Remember that the internally generated request is kept until the servicing of the request is finished.

External transfer requests come from the Interrupt Controller (INTC). The INTC can program any interrupts to be used as a DMA trigger instead of as an interrupt request. If such an interrupt is programmed for edge sensitivity, the INTC internally maintains a transfer request. However, a level-sensitive interrupt is not held in the INTC; thus the interrupt request signal must remain asserted until the servicing of the DMA request begins.

A higher-priority channel always gets the attention of the DMAC. If a transfer request occurs on channel 0 while a request on channel 1 is being serviced, the servicing of the channel 1 request is suspended temporarily in order to service the channel 0 request first. After the channel 0 request has been serviced, channel 1 resumes the remaining data transfer.

Channel transitions take place at the boundary of a transfer size programmed for the current channel being serviced; that is, after all data in the DHR are written to a destination.

Note: DMA channel priority exists only among those using the same type of bus request signal (SREQ or GREQ).

10.4.6 Interrupts

The DMAC can generate an interrupt request (INTDMAn) to the TX19 core processor on completion of a channel operation: either by normal channel termination or by abnormal termination of a bus cycle.

- Normal Completion Interrupt

  When a channel operation terminates by normal completion, the NC bit in the CSRn is set to 1. At this time, if the NIEn bit in the CCRn is set, an interrupt request is generated to the TX19 core processor.

- Abnormal Completion Interrupt

  When a channel operation terminates abnormally, the AbC bit in the CSRn register is set to 1. At this time, if the AbIEn bit in the CCRn register is set, an interrupt request is generated to the TX19 core processor.

10.4.7 Data Packing and Unpacking

In dual-address mode, the internal 32-bit DHR allows the data to be packed and unpacked by the DMAC if the programmed transfer size is not equal to the device port size.

For example, if a source I/O peripheral is 8-bits wide and a destination memory device is 32-bits wide, four byte-read cycles occur. The four bytes of data are buffered in the DHR before a destination word-write cycle occurs.

The following illustrates the byte ordering for packing and unpacking of data.

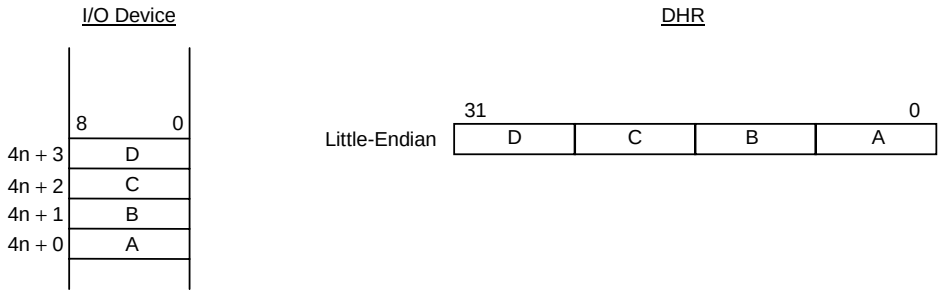


Figure 10.13 Data Packing and Unpacking

10.5 DMA Transfer Timing

All DMAC operations are synchronous to the rising edges of the internal system clock.

10.5.1 Dual-Address Mode

- Memory-to-memory transfer

  Figure 10.14 shows a DMA cycle from one external 16-bit memory to another, with the transfer size programmed to 16 bits. A block of data is transferred until the BCRn register reaches 0.

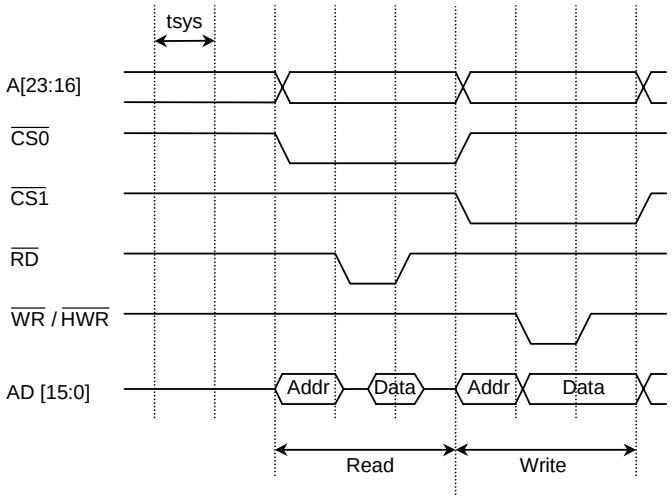


Figure 10.14 Memory-to-Memory Transfer (Dual-Address Mode)

- Memory-to-I/O transfer

  Figure 10.15 shows a DMA cycle from a 16-bit memory to an 8-bit I/O peripheral, with the transfer size programmed to 16 bits.

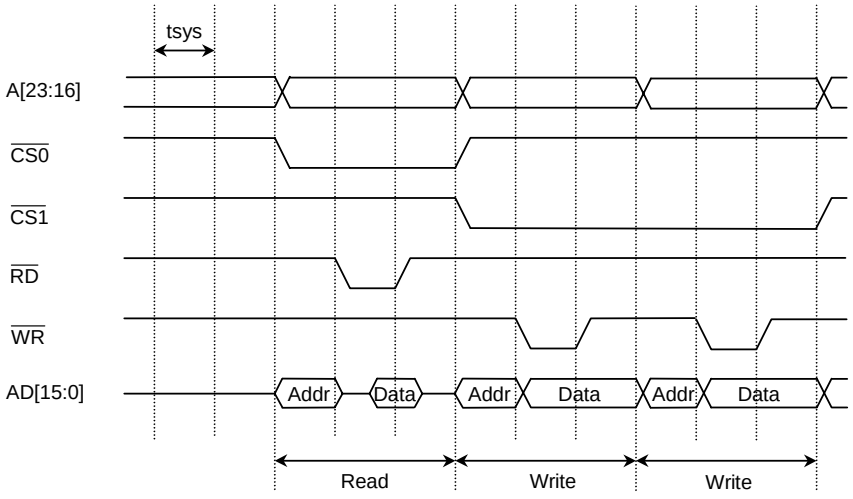


Figure 10.15 Memory-to-I/O Transfer (Dual-Address Mode)

- I/O-to-memory transfer

  Figure 10.16 shows a DMA cycle from an 8-bit I/O peripheral to a 16-bit memory, with the transfer size programmed to 16 bits.

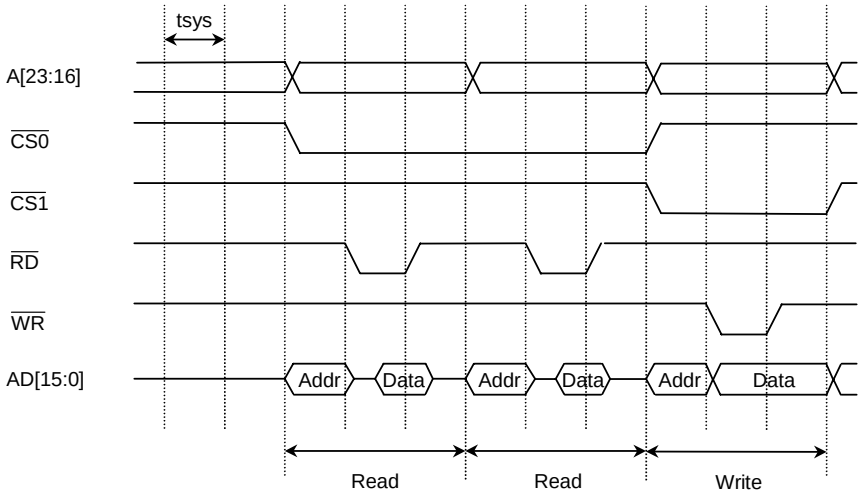


Figure 10.16 I/O-to-Memory Transfer (Dual-Address Mode)

10.6 Programming Example

The following illustrates the programming required to transfer data from an SIO receive buffer (SCnBUF) to the on-chip RAM. The assumptions are as follows:

DMAC Settings:

- DMA channel used: Channel 0
- Source address: SC1BUF
- Destination address: 0xFFFF_9800 (physical address)
- Number of bytes transferred: 256

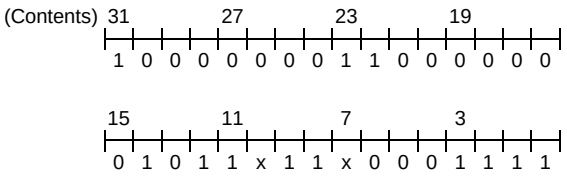
SIO Settings:

- Data format: 8 bits, UART
- SIO channel used: Channel 1
- Transfer rate: 9600 bps

DMA channel 0 is used for the transfer. The SIO1 receive interrupt is used as a trigger to start the DMA channel.

DMA channel 0 settings:

```
DCR     ←  0x8000_0000                 /* Reset DMAC * /
IMCFL   ←  15          7         0     /* Bit positions */
           xxxx, xxxx, xx10, x100      /* Interrupt level = 4 (arbitrary) * /
INTCLR  ←  0x3c                        /*IVR[9:4]; clear INTDMA0 * /
DTCR0   ←  0x0000_0000                 /* DACM = 000 * /
                                       /* SACM = 000 * /
SAR0    ←  0xFFFF_F208                 /* Physical address of SC1BUF */
DAR0    ←  0xFFFF_9800                 /* Physical address of destination */
BCR0    ←  0x0000_00FF                 /* 256 (Number of bytes to be transferred) */
CCR0    ←  0x80c0_5b0f
```



SIO channel 1 settings:

```
IMCCH   ←  31                  16     /* Bit positions */
           xxxx, xxxx, xx11, 1000      /* Use INTRX1 as a DMA trigger and select DMA ch. 0 * /
INTCLR  ←  0x32                        /* IVR[9:4]; clear INTRX1 * /
SC1MOD0 ←  0x09                        /* UART mode, 8-bit data format, baud rate generator * /
SC1CR   ←  0x00
BR1CR   ←  0x1d                        /* @fc = 32 MHz (approx. 9615 bps) */
SC1MOD0 ←  0x29                        /* Enable receiver * /
```

11. 8-Bit Timers (TMRAs)

The TMP1940CYAF has a four-channel 8-bit timer (TMRA0–TMRA3), which is comprised of two modules named TMRA01 and TMRA23. The TMRA01 contains the TMRA0 and the TMRA1, and the TMRA23 contains the TMRA2 and TMRA3. Each timer module has the following operating modes:

- 8-bit interval timer mode
- 16-bit interval timer mode
- 8-bit programmable pulse generation (PPG) mode (Variable frequency, variable duty cycle)
- 8-bit pulse width modulated (PWM) signal generation mode (Fixed frequency, variable duty cycle)

Figure 11.1 and Figure 11.2 are block diagrams of the TMRA01 and TMRA23 respectively. The main components of a timer channel are an 8-bit up-counter, an 8-bit comparator and an 8-bit timer register. Two timer channels share a prescalar and a timer flip-flop.

A total of six 8-bit registers provide control over the operating modes and timer flip-flops for the TMRA01 and the TMRA23 each, which can be independently programmed. The TMRA01 and the TMRA23 are functionally equivalent. In the following sections, any references to the TMRA01 also apply to the TMRA23.

Table 11.1 gives the pins and registers for the two timer modules.

Table 11.1 Pins and Registers for the TMRA01 and the TMRA23

		TMRA01	TMRA23
External Pins	External clock input	TA0IN (Shared with P70)	TA2IN (Shared with P72)
	Timer flip-flop output	TA1OUT (Shared with P71)	TA3OUT (Shared with P73)
Registers (Addresses)	Timer Run register	TA01RUN (0xFFFF_F100)	TA23RUN (0xFFFF_F108)
	Timer registers	TA0REG (0xFFFF_F102) TA1REG (0xFFFF_F103)	TA2REG (0xFFFF_F10A) TA3REG (0xFFFF_F10B)
	Timer Mode register	TA01MOD (0xFFFF_F104)	TA23MOD (0xFFFF_F10C)
	Timer Flip-Flop Control register	TA1FFCR (0xFFFF_F105)	TA3FFCR (0xFFFF_F10D)

11.1 Block Diagrams

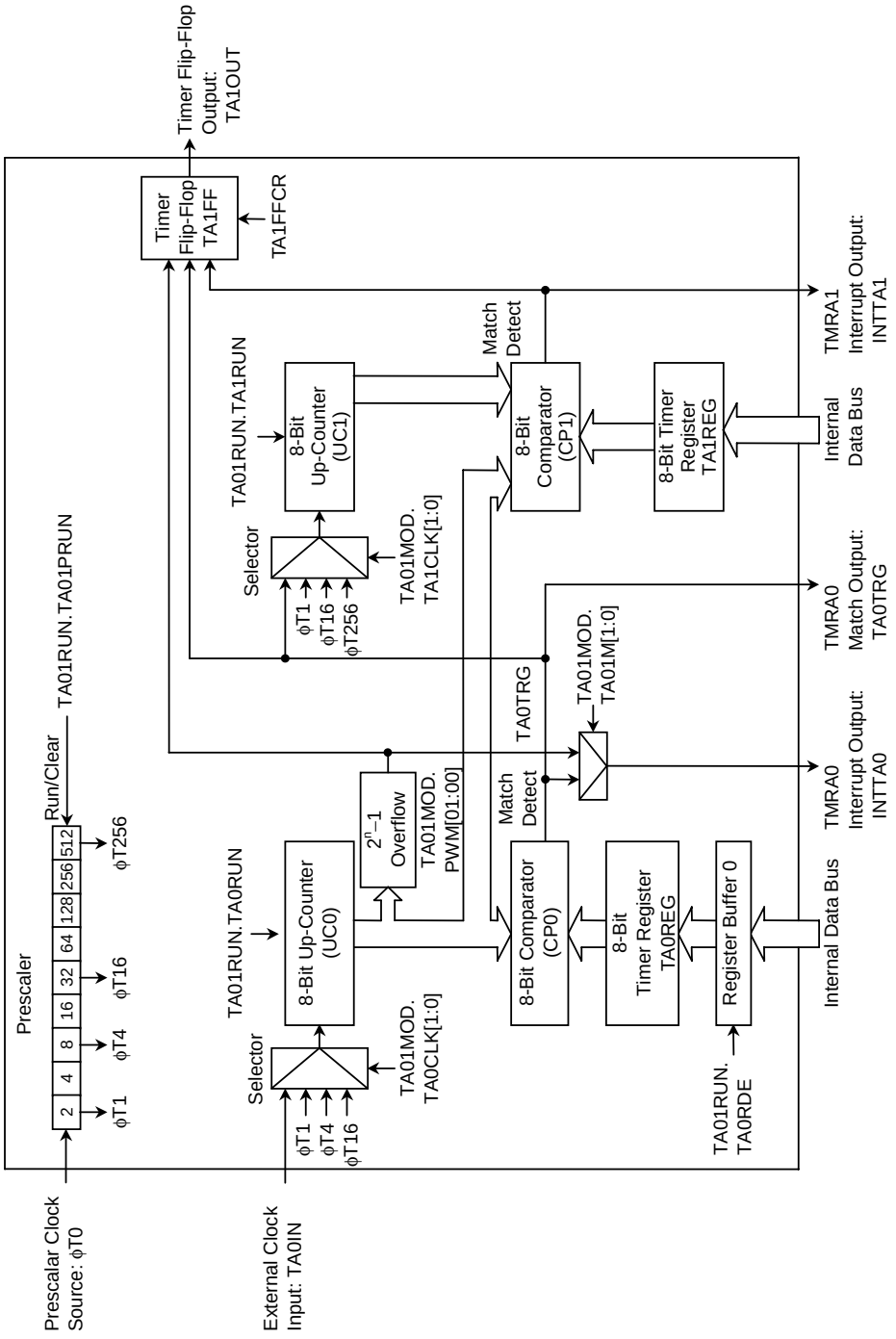


Figure 11.1 TMRA01 Block Diagram

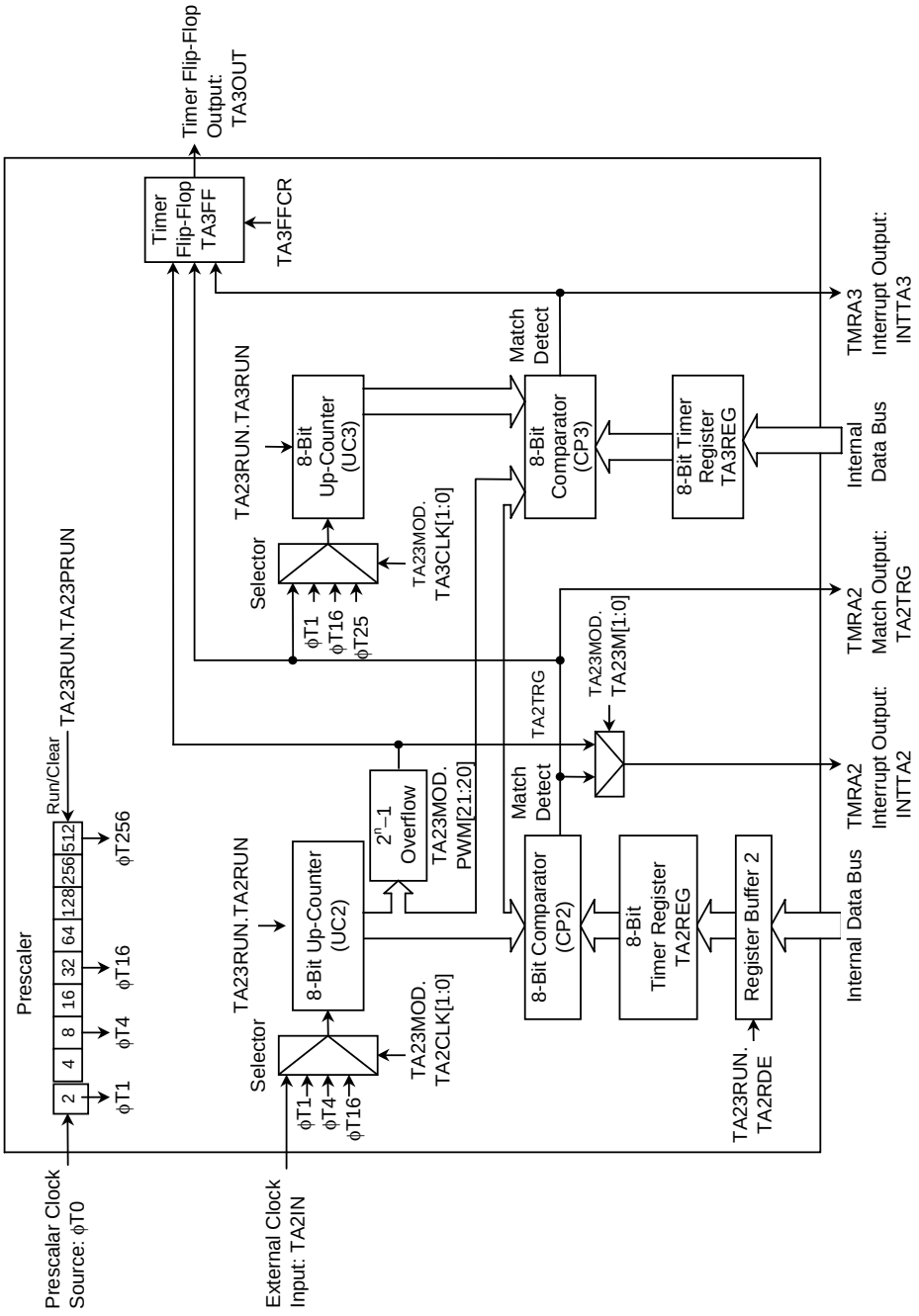


Figure 11.2 TMRA23 Block Diagram

11.2 Timer Components

11.2.1 Prescaler

The TMRA01 has a 9-bit prescalar that slows the rate of a clocking source to the counters. The prescalar clock source (ϕT0) can be selected from fperiph, fperiph/2 and fperiph/4 by programming the PRCK[1:0] field of the SYSCR0 located within the CG. fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The TA01PRUN bit in the TA01RUN register allows the enabling and disabling of the prescalar for the TMRA01. A write of 1 to this bit starts the prescalar. A write of 0 to this bit clears and halts the prescalar.

Prescalar output taps can be divide-by-2 (ϕT1), divide-by-8 (ϕT4), divide-by-32 (ϕT16) and divide-by-512 (ϕT256). Table 11.2 shows prescalar output clock resolutions (@fc = 32 MHz).

Table 11.2 Prescalar Output Clock Resolutions

@fc = 32MHz

Peripheral Clock Select SYSCR1.FPSEL	Clock Gear Value SYSCR1.GEAR[1:0]	Prescaler Clock Source SYSCR0.PRCK[1:0]	Prescalar Output Clock Resolution			
			ϕT1	ϕT4	ϕT16	ϕT256
0 (fgear)	00 (fc)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^9$ (16 μs)
	01 (fc/2)	00 (fperiph/4)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{12}$ (128 μs)
		01 (fperiph/2)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		10 (fperiph)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
	10 (fc/4)	00 (fperiph/4)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)	$fc/2^{13}$ (256 μs)
		01 (fperiph/2)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{12}$ (128 μs)
		10 (fperiph)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
	11 (fc/8)	00 (fperiph/4)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32 μs)	$fc/2^{14}$ (512 μs)
		01 (fperiph/2)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)	$fc/2^{13}$ (256 μs)
		10 (fperiph)	—	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{12}$ (128 μs)
1 (fc)	00 (fc)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^9$ (16 μs)
	01 (fc/2)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^9$ (16 μs)
	10 (fc/4)	00 (fperiph/4)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	—	$fc/2^5$ (1.0 μs)	$fc/2^9$ (16 μs)
	11 (fc/8)	00 (fperiph/4)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	—	—	$fc/2^6$ (2.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	—	$fc/2^5$ (1.0 μs)	$fc/2^9$ (16 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that ϕTn < fsys/2 is satisfied.

Note 2: Do not change the clock gear value while the timer is running.

Note 3: The — character means "Don't use."

11.2.2 Up-Counters (UC0 and UC1)

The timer module contains two 8-bit binary up-counters, each of which is driven by a clock independently selected by the TA01MOD register.

The clock input to the UC0 is either one of three prescalar outputs (ϕT1,ϕT4, ϕT16) or the external clock applied to the TA0IN pin. Which clock is to use is programmed into the TA0CLK[1:0] field of the TA01MOD register.

Possible clock sources for the UC1 depend on the selected operating mode. In 16-bit interval timer mode, the clock input to the UC1 is always the UC0 overflow output. In other operating modes, the clock input to the UC1 is either one of three prescalar outputs (ϕT1,ϕT16, ϕT256) or the TMRA0 comparator match-detect output.

The TA0RUN and TA1RUN bits in the TA01RUN register are used to start counting and to stop and clear the counter. Upon reset, the up-counter is set to 00H and the whole timer module is disabled.

11.2.3 Timer Registers (TA0REG and TA1REG)

Each timer register is an 8-bit register containing a time constant. When the up-counter reaches the time constant value in the timer register, the comparator block generates a match-detect signal. When the time constant is set to 00H, a match occurs upon a counter overflow.

One of the two timer registers, TA0REG, is double-buffered. The double-buffering function can be enabled and disabled through the programming of the TA0RDE bit in the TA01RUN: 0=disable, 1=enable.

If double-buffering is enabled, the TA0REG latches a new time constant value from the register buffer. This takes place upon detection of a 2^n–1 overflow in PWM mode and upon a match between the UC0 and the TA1REG in PPG mode. Double-buffering must be disabled in interval timer modes.

A reset clears the TA01RUN.TA0RDE bit to 0, disabling the double-buffering function. To use this function, the TA01RUN.TA0RDE bit must be set to1 after loading the TA0REG with a time constant. When TA01RUN.TA0RDE=1, the next time constant can be written to the register buffer.

Figure 11.13 illustrates the double-buffer structure for the TA0REG.

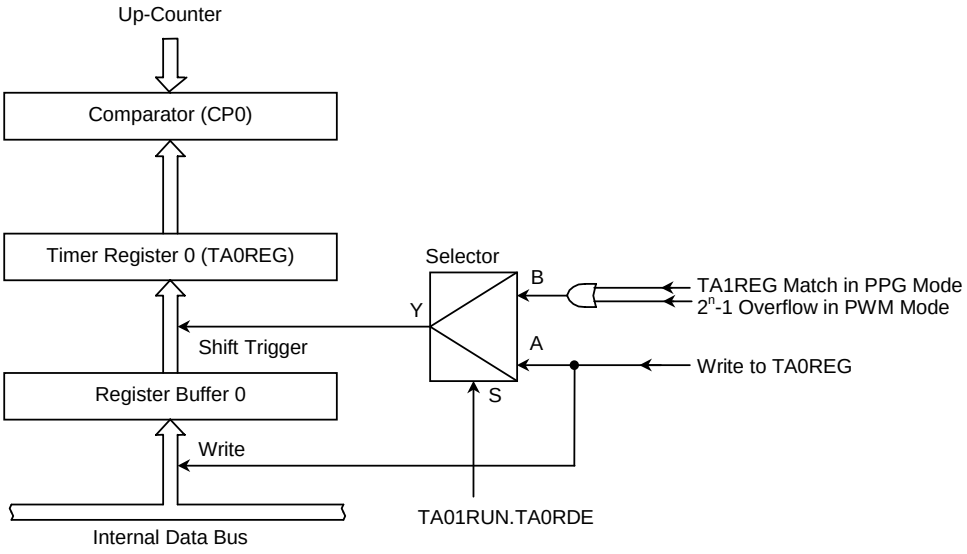


Figure 11.3 Timer Register 0 (TA0REG) Structure

Note: The timer register and the corresponding register buffer are mapped to the same address. When TA01RUN.TA0RDE=0, a time constant value is written to both of the timer register and the register buffer; when TA01RUN.TA0RDE=1, a time constant value is written only to the register buffer.

The addresses of the timer registers are as follows:

TA0REG: 0xFFFF_F102

TA1REG: 0xFFFF_F103

TA2REG: 0xFFFF_F10A

TA3REG: 0xFFFF_F10B

The timer registers are write-only registers.

11.2.4 Comparators (CP0 and CP1)

The comparator compares the output of the 8-bit up-counter with a time constant value in the 8-bit timer register. When a match is detected, an interrupt (INTTA0/INTTA1) is generated and the timer flip-flop is toggled, if so enabled.

11.2.5 Timer Flip-Flop (TA1FF)

The timer flip-flop (TA1FF) is toggled, if so enabled, each time the comparator match-detect output is asserted. The toggling of the timer flip-flop can be enabled and disabled through the programming of the TAFF1IE bit in the TA1FFCR.

A reset clears the TAFF1IE bit, disabling the toggling of the TA1FF. The TA1FF can be initialized to 1 or 0 by writing 01 or 10 to the TAFF1C[1:0] field in the TA1FFCR. Additionally, a write of 00 by software causes the TA1FF to be toggled to the opposite value.

The value of the TA1FF can be driven onto the TA1OUT pin, which is multiplexed with P71. The Port 7 registers (P7CR and P7FC) must be programmed to configure the P71/TA1OUT pin as TA1OUT.

11.3 Register Description

TMRA01 Run Register

TA01RUN (0xFFFF_F100)

	7	6	5	4	3	2	1	0
Name	TA0RDE	—	—	—	I2TA01	TA01PRUN	TA1RUN	TA0RUN
Read/Write	R/W	—	—	—	R/W			
Reset Value	0	—	—	—	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable				IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run	Timer Run/Stop Control 0: Stop & clear 1: Run	

I2TA01: Timer on/off in IDLE mode
TA01PRUN: Prescaler
TA1RUN: TMRA1
TA0RUN: TMRA0

Note: Bits 4, 5 and 6 are read as undefined.

TMRA23 Run Register

TA23RUN (0xFFFF_F108)

	7	6	5	4	3	2	1	0
Name	TA2RDE	—	—	—	I2TA23	TA23PRUN	TA3RUN	TA2RUN
Read/Write	R/W	—	—	—	R/W			
Reset Value	0	—	—	—	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable				IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run	Timer Run/Stop Control 0: Stop & clear 1: Run	

I2TA23: Timer on/off in IDLE mode
TA23PRUN: Prescaler
TA3RUN: TMRA3
TA2RUN: TMRA2

Note: Bits 4, 5 and 6 are read as undefined.

Figure 11.4 Timer Run Registers

TMRA01 Mode Register

TA01MOD (0xFFFF_F104)

	7	6	5	4	3	2	1	0
Name	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		TMRA1 clock source 00: TA0TRG 01: φT1 10: φT16 11: φT256		TMRA0 clock source 00: TA0IN input 01: φT1 10: φT4 11: φT16	

TMRA0 clock source

00	External input (TA0IN)
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

TMRA1 clock source

	TA01MOD.TA01M[1:0]≠01	TA01MOD.TA01M[1:0]=01
00	TMRA0 match output	TMRA0 overflow output (16-Bit Timer Mode)
01	φT1	
10	φT16	
11	φT256	

Period select in 8-bit PWM mode

00	Reserved
01	$(2^6-1) \times$ clock source
10	$(2^7-1) \times$ clock source
11	$(2^8-1) \times$ clock source

TMRA01 operating mode

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM generation (TMRA0) & 8-bit timer (TMRA1)

Figure 11.5 Timer Mode Register

TMRA23 Mode Register

TA23MOD (0xFFFF_F10C)

	7	6	5	4	3	2	1	0
Name	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
Read/Write	R/W							
Reset Valu	0	0	0	0	0	0	0	0
Function	Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		TMRA3 clock source 00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA2 clock source 00: TA2IN 01: ϕT1 10: ϕT4 11: ϕT16	

TMRA2 clock source

00	External input (TA2IN)	
01	ϕT1	(Prescaler)
10	ϕT4	(Prescaler)
11	ϕT16	(Prescaler)

TMRA3 clock source

	TA23MOD.TA23M[1:0]≠01	TA23MOD.TA23M[1:0]=01
00	TMRA2 match output	TMRA2 overflow output (16-Bit Timer Mode)
01	ϕT1	
10	ϕT16	
11	ϕT256	

Period select in 8-bit PWM mode

00	Reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

TMRA23 operating mode

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM generation (TMRA2) & 8-bit timer (TMRA3)

Figure 11.6 Timer Mode Register

TMRA01 Timer Flip-Flop Control Register

TA1FFCR (0xFFFF_F105)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	1	1	0	0
Function					00: Toggles TA1FF. (software toggle) 01: Sets TA1FF to 1. 10: Clears TA1FF to 0. 11: Don't-care This field is always read as 11.		TA1FF toggle enable 0: Disable 1: Enable	TA1FF toggle trigger 0: TMRA0 1: TMRA1

Selects a signal to toggle Timer Flip-Flop 1 (TA1FF)
(Don't-care in other than 8-bit timer mode)

0	Toggled by TMRA0
1	Toggled by TMRA1

Note: Bits 4 to 7 are read as undefined.

Figure 11.7 TMRA01 Flip-Flop Control Register

TMRA23 Flip-Flop Control Register

TA3FFCR (0xFFFF_F10D)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	TAFF3C1	TAFF3C0	TAFF3IE	TAFF3IS
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	1	1	0	0
Function					00: Toggles TA3FF (software toggle). 01: Sets TA3FF to 1 10: Clears TA3FF to 0 11: Don't care This field is always read as 11.		TA3FF toggle enable 0: Disable 1: Enable	TA3FF trigger 0: TMRA2 1: TMRA3

Selects a signal to toggle Timer Flip-Flop 3 (TA3FF) (Don't-care in other than 8-bit timer mode)

0	Toggled by TMRA2
1	Toggled by TMRA3

Note: Bits 4 to 7 are read as undefined values.

Figure 11.8 TMRA23 Flip-Flop Control Register

11.4 Operating Modes

11.4.1 8-Bit Interval Timer Mode

The TMRA0 and the TMRA1 can be independently programmed as 8-bit interval timers. Programming these timers should only be attempted when the timers are not running.

(1) Generating Periodic Interrupts

In the following example, the TMRA1 is used to accomplish periodic interrupt generation. First, stop the TMRA1 (if it is running). Then, set the operating mode, clock source and interrupt interval in the TA01MOD and TA1REG registers. Then, enable the INTTA1 interrupt and start the TMRA1.

Example: Generating the INTTA1 interrupt at a 20-µs interval (fc = 32 MHz)

Clocking conditions:

System clock: High-speed (fc)

Prescaler clock: fperiph/4 (fperiph = fsys)

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	–	X	X	–	–	0	–	Stops and clears the TMRA1.
TA01MOD	← 0	0	X	X	1	0	X	X	Selects 8-bit interval timer mode and φT1 as the clock source (which provides a 0.25-µs resolution @fc=32 MHz.)
TA1REG	← 0	1	0	1	0	0	0	0	Sets the time constant value in the TA1REG. 20 µs ÷ φT1 = 80 (50H)
IMC5LH	← X	X	1	1	0	1	0	1	Enables INTTA1 and sets the interrupt level to 5. INTTA1 must always be programmed to be rising-edge triggered.
TA01RUN	← –	X	X	X	–	1	1	–	Starts the TMRA1.

X = Don't care, – = No change

Refer to Table 11.2 when selecting a timer clock source.

Note: The clock inputs to the TMRA0 and the TMRA1 can be one of the following:
TMRA0: TA0IN input, φT1, φT4 or φT16
TMRA1: Match-detect signal from the TMRA0, φT1, φT16 or φT256

(2) Generating a SquareWave with a 50% Duty Cycle

The 8-bit interval timer mode can be used to generate square-wave output. This is accomplished by toggling the timer flip-flop (TA1FF) periodically. The TA1FF state can be driven out to the TA1OUT pin. Both the TMRA0 and the TMRA1 can be used as square-wave generators. The following shows an example using the TMRA1.

Example: Generating square-wave output with a 1.5-µs period on the TA1OUT pin (fc = 32 MHz)

Clocking conditions:

System clock: High-speed (fc)
High-speed clock gear: ×1 (fc)
Prescaler clock: fperiph/4 (fperiph = fsys)

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	X	X	X	–	–	0	–	Stops and clears the TMRA1.
TA01MOD	← 0	0	X	X	0	1	–	–	Selects 8-bit interval timer mode and φT1 as the clock source (which provides a 0.25-µs resolution @fc=32 MHz).
TA1REG	← 0	0	0	0	0	0	1	1	Sets the time constant value in the TA1REG. 1.5 µs ÷ φT1 ÷ 2 = 3
TA1FFCR	← X	X	X	X	1	0	1	1	Clears the TA1FF to 0 and selects the TMRA1 match-detect output as a toggle-trigger signal.
P7CR	← –	–	–	–	–	–	1	–	Configures P71 as the TA1OUT output pin.
P7FC	← –	–	–	–	–	–	1	–	
TA01RUN	← –	X	X	X	–	1	1	–	Starts the TMRA1.

X = Don't care, – = No change

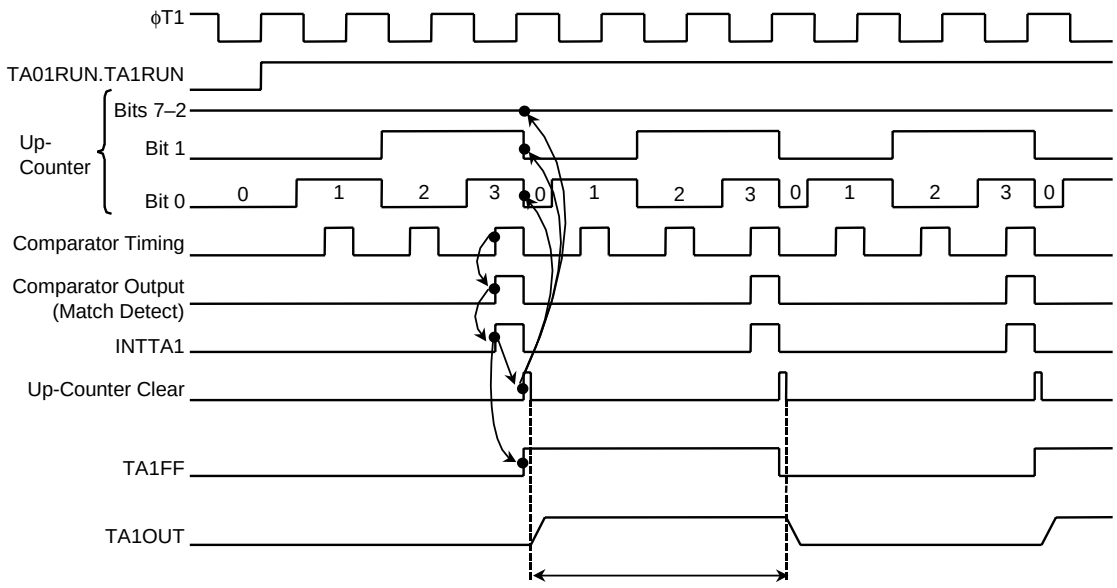


Figure 11.9 Square-Wave Generation (50% Duty Cycle)

(3) Using the TMRA0 Match-Detect Output as a Trigger for the TMRA1

Set the TMRA01 in 8-bit interval timer mode. Select the TMRA0 comparator match-detect output (TA0TRG) as the clock source for the TMRA1.

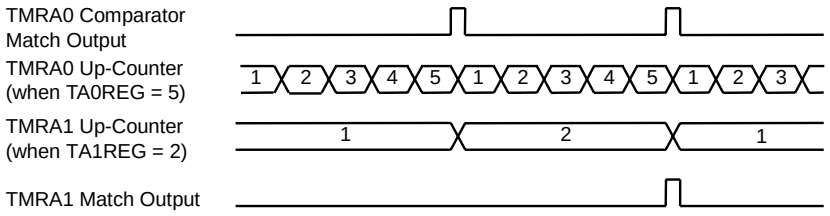


Figure 11.10 Using the TMRA0 Match-Detect Output as a Trigger for the TMRA1

11.4.2 16-Bit Interval Timer Mode

The TMRA0 and the TMRA1 are cascadable to form a 16-bit interval timer. The TMRA01 is put in 16-bit interval timer mode by programming the TA01M[1:0] field in the TA01MOD register to 01.

In 16-bit interval timer mode, the TMRA1 is clocked by the counter overflow output from the TMRA0. In this mode, the TA1CLK[1:0] bits in the TA01MOD register are don't-cares. The clock input to the TMRA0 can be selected from an external clock and one of three prescalar outputs (see Table 11.2).

Write the lower eight bits of a time constant value to the TA0REG and the upper eight bits to the TA1REG. Programming these registers should only be attempted when the timers are not running.

Example: Generating the INTTA1 interrupt at a 0.2-second interval (fc = 32 MHz)

Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	×1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

Under the above conditions, ϕT16 has a period of 4.0 μs @ 32 MHz. When ϕT16 is used as the TMRA0 clock source, the required time constant value is calculated as follows:

$0.2\ s \div 4.0\ \mu s = 50000 = C350H$

Thus, the TA1REG is to be set to C3H and the TA0REG to 50H.

Every time the up-counter UC0 reaches the value in the TA0REG, the TMRA0 comparator generates a match-detect output, but the TMRA0 continues counting up. A match between the UC0 and the TA0REG does not cause an INTTA0 interrupt.

Every time the up-counter UC1 reaches the value in the TA1REG, the TMRA1 comparator generates a match-detect output. When the TMRA0 and TMRA1 match-detect outputs are asserted simultaneously, both the up-counters (UC0 and UC1) are reset to 00H and an interrupt is generated on INTTA1. Also, if so enabled, the timer flip-flop (TA1FF) is toggled.

Example: TA1REG = 04H and TA0REG = 80H

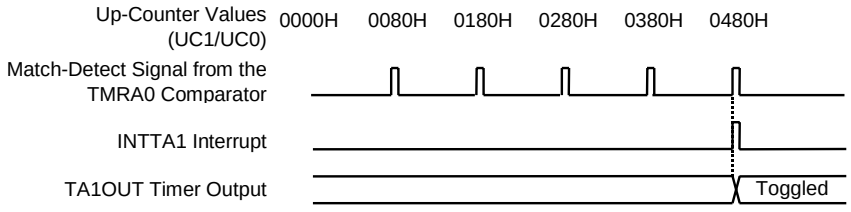


Figure 11.11 Timer Output in 16-Bit Interval Timer Mode

11.4.3 8-Bit Programmable Pulse Generation (PPG) Mode

The 8-bit PPG mode can be used to generate a square wave with any frequency and duty cycle, as shown below. The pulse can be high-going and low-going, as determined by the initial setting of the timer flip-flop (TA1FF). This mode is supported by the TMRA0, but not by the TMRA1. The square-wave output is driven to the TA1OUT pin (which is multiplexed with P71).

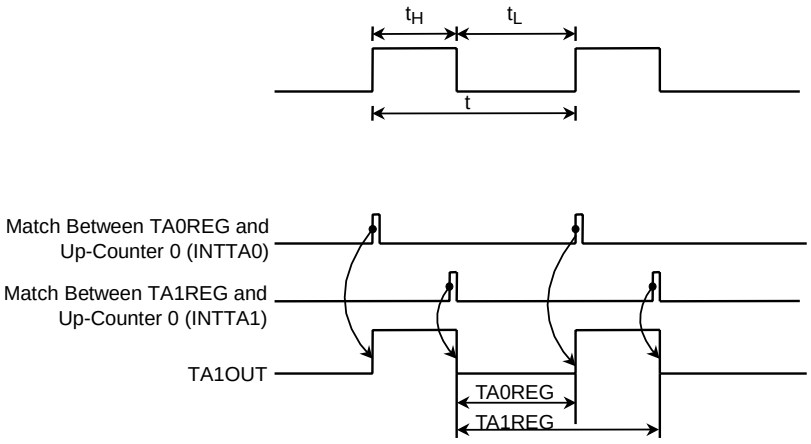


Figure 11.12 8-Bit PPG Output Waveform

In this mode, a square wave is generated by toggling the timer flip-flop (TA1FF). The TA1FF changes state every time a match is detected between the UC0 and the TA0REG and between the UC0 and the TA1REG.

The TA0REG must be set to a value less than the TA1REG value.

In this mode, the TMRA1 up-counter (UC1) can not be independently used; however, the TMRA1 must be put in a running state by setting the TA1RUN bit in the TA01RUN register to 1.

Figure 11.3 shows a functional diagram of 8-bit PPG mode.

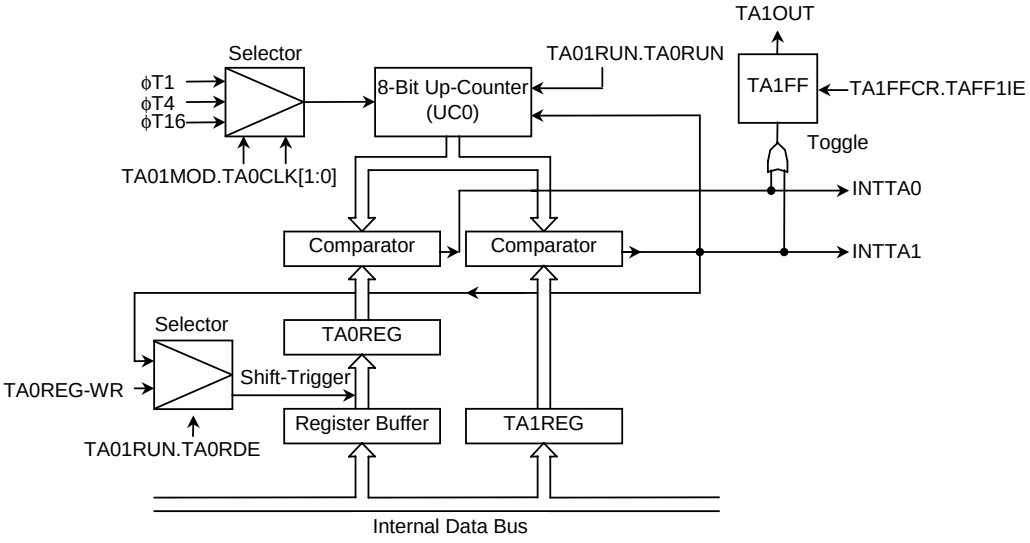


Figure 11.13 Functional Diagram of 8-Bit PPG Mode

In 8-bit PPG mode, if the double-buffering function is enabled, the TA0REG value can be changed dynamically by writing a new value into the register buffer. Upon a match between the TA1REG and the UC0, the TA0REG latches a new value from the register buffer.

The TA0REG can be loaded with a new value upon every match, thus making it easy to generate a square wave with virtually any (and variable) duty cycle.

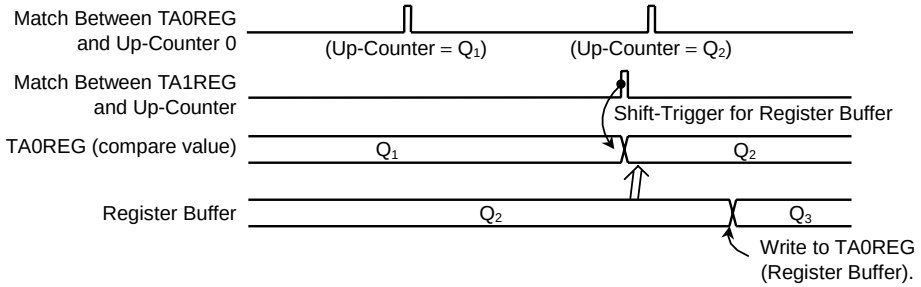
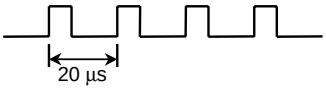


Figure 11.14 Register Buffer Operation

Example: Generating a 50-kHz square wave with a 25% duty cycle (fc = 32 MHz)



Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	×1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

The time constant values to be loaded into the TA0REG and TA1REG are determined as follows:

A 50-kHz waveform has a period of 20 µs. Under the above clocking conditions, ϕT1 has a 0.25-µs resolution (@fc = 32 MHz). When ϕT1 is used as the timer clock source, the TA1REG should be loaded with:

20 µs ÷ 0.25 µs = 80 (50H)

With a 25% duty cycle, the high pulse width is calculated as 20 µs × 1/4 = 5 µs. Thus, the TA0REG should be loaded with:

5 µs ÷ 0.25 µs = 20 (14H)

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← 0	X	X	X	–	0	0	0	Stops and clears the TMRA0.
TA01MOD	← 1	0	X	X	X	X	0	1	Selects 8-bit PPG mode and ϕT1 as the clock source.
TA0REG	← 0	0	0	1	0	1	0	0	Writes 14H.
TA1REG	← 0	1	0	1	0	0	0	0	Writes 50H.
TA1FFCR	← X	X	X	X	0	1	1	X	Sets the TA1FF to 1 and enables toggling.
									If these bits are set to 10, a low-going pulse is generated.
P7CR	← –	–	–	–	–	–	1	–	Configures P71 as the TA1OUT output pin.
P7FC	← –	–	–	–	–	–	1	–	
TA01RUN	← 1	X	X	X	–	1	1	1	Starts the TMRA0 and the TMRA1.

X = Don't care, – = No change

11.4.4 8-Bit PWM Generation Mode

The TMRA0 can be used as a pulse-width modulated (PWM) signal generator with up to 8 bits of resolution. This mode is supported by the TMRA0, but not by the TMRA1. The PWM signal is driven out on the TA1OUT pin (which is multiplexed with P71).

While the TMRA01 is in this mode, the TMRA1 is usable as an 8-bit interval timer. However, the TMRA0 match-detect output can not be used as a clock source for the TMRA1, and the timer output is not available for the TMRA1.

The timer flip-flop toggles when the up-counter (UC0) reaches the TA0REG value and when a 2^n-1 counter overflow occurs, where n is programmable to 6, 7 or 8 through the PWM[01:00] field in the TA01MOD register. The UC0 is reset to 00H upon a 2^n-1 overflow.

In 8-bit PWM generation mode, the following must be satisfied:

(TA0REG value) < (2^n-1 counter overflow value)

(TA0REG value) ≠ 0

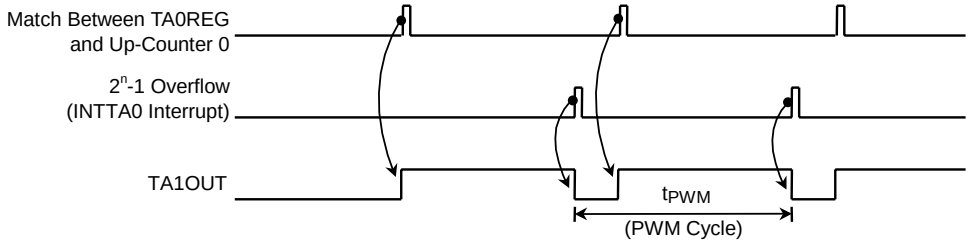


Figure11.15 8-Bit PWM Signal Generation

Figure 11.16 shows a functional diagram of 8-bit PWM generation mode.

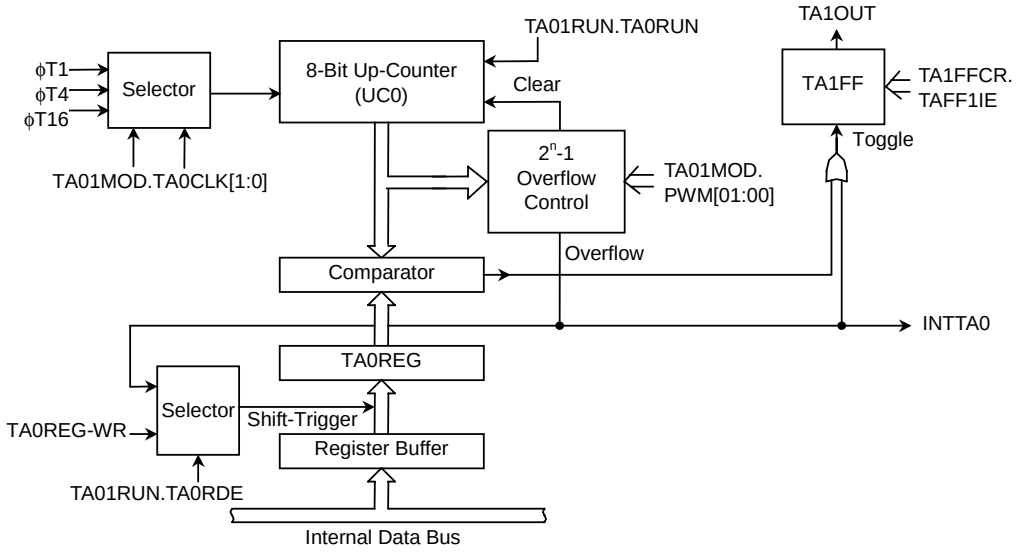


Figure 11.16 Functional Diagram of 8-Bit PWM Generation Mode

In 8-bit PWM generation mode, if the double-buffering function is enabled, the TA0REG value (i.e., the duty cycle) can be changed dynamically by writing a new value into the register buffer. Upon a 2^n-1 counter overflow, the TA0REG latches a new value from the register buffer.

The TA0REG can be loaded with a new value upon every counter overflow, thus generating a PWM signal with variable duty cycle.

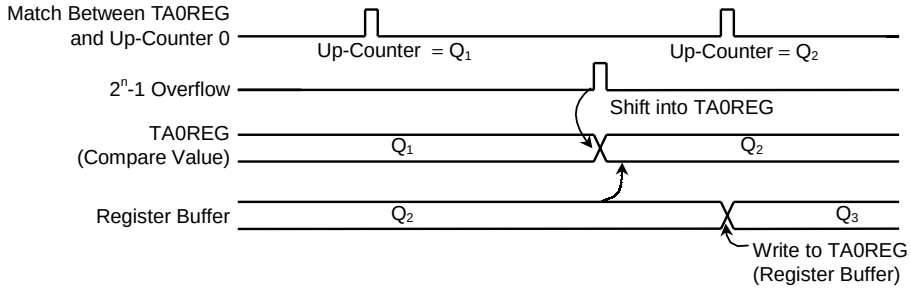
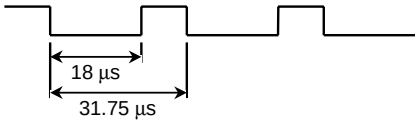


Figure11.17 Register Buffer Operation

Example: Generating a PWM signal as shown below on the TA1OUT pin (fc = 32 MHz)



Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	×1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

Under the above conditions, ϕT1 has a 0.25-µs period (@fc = 32 MHz).

31.75 µs ÷ 0.25 µs = 127

which is equal to $2^7 - 1$.

18 µs ÷ 0.25 µs = 72 = 48H

Hence, the time constant value to be programmed into the TA0REG is 48H.

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	X	X	X	–	–	–	0	Stops and clears the TMRA0.
TA01MOD	← 1	1	1	0	–	–	0	1	Selects 8-bit PWM mode (period = 2^7–1) and ϕT1 as the clock source.
TA0REG	← 0	1	0	0	1	0	0	0	Writes 48H.
TA1FFCR	← X	X	X	X	1	0	1	X	Clears the TA1FF to 0 and enables toggling.
P7CR	← –	–	–	–	–	–	1	–	Configures P71 as the TA1OUT output pin.
P7FC	← –	–	–	–	–	–	1	–	
TA01RUN	← 1	X	X	X	–	1	–	1	Starts the TMRA0.

X = Don't care, – = No change

Table 11.3 PWM Period

@fc = 32 MHz

Peripheral Clock Select SYSCR1. FPSEL	Clock Gear Value SYSCR1. GEAR[1:0]	Prescaler Clock Source SYSCR0. PRCK[1:0]	PWM Period								
			$2^6 - 1$			$2^7 - 1$			$2^8 - 1$		
			ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
0 (fgear)	00 (fc)	00 (fperiph/4)	15.8 µs	63 µs	252 µs	31.8 µs	127 µs	508 µs	63.8 µs	255 µs	1020 µs
		01 (fperiph/2)	7.9 µs	31.5 µs	126 µs	15.9 µs	63.5 µs	254 µs	31.9 µs	127.5 µs	510 µs
		10 (fperiph)	—	15.8 µs	63 µs	—	31.8 µs	127 µs	—	63.8 µs	255 µs
	01 (fc/2)	00 (fperiph/4)	31.5 µs	126 µs	504 µs	63.5 µs	254 µs	1016 µs	127.5 µs	510 µs	2040 µs
		01 (fperiph/2)	15.8 µs	63 µs	252 µs	31.8 µs	127 µs	508 µs	63.8 µs	255 µs	1020 µs
		10 (fperiph)	—	31.5 µs	126 µs	—	63.5 µs	254 µs	—	127.5 µs	510 µs
	10 (fc/4)	00 (fperiph/4)	63 µs	252 µs	1008 µs	127 µs	508 µs	2032 µs	255 µs	1020 µs	4080 µs
		01 (fperiph/2)	31.5 µs	126 µs	504 µs	63.5 µs	254 µs	1016 µs	127.5 µs	510 µs	2040 µs
		10 (fperiph)	—	63 µs	252 µs	—	127 µs	508 µs	—	255 µs	1020 µs
	11 (fc/8)	00 (fperiph/4)	126 µs	504 µs	2016 µs	254 µs	1016 µs	4064 µs	510 µs	2040 µs	8160 µs
		01 (fperiph/2)	63 µs	252 µs	1008 µs	127 µs	508 µs	2032 µs	255 µs	1020 µs	4080 µs
		10 (fperiph)	—	126 µs	504 µs	—	254 µs	1016 µs	—	510 µs	2040 µs
1 (fc)	00 (fc)	00 (fperiph/4)	15.8 µs	63 µs	252 µs	31.8 µs	127 µs	508 µs	63.8 µs	255 µs	1020 µs
		01 (fperiph/2)	7.9 µs	31.5 µs	126 µs	15.9 µs	63.5 µs	254 µs	31.9 µs	127.5 µs	510 µs
		10 (fperiph)	—	15.8 µs	63 µs	—	31.8 µs	127 µs	—	63.8 µs	255 µs
	01 (fc/2)	00 (fperiph/4)	15.8 µs	63 µs	252 µs	31.8 µs	127 µs	508 µs	63.8 µs	255 µs	1020 µs
		01 (fperiph/2)	—	31.5 µs	126 µs	—	63.5 µs	254 µs	—	127.5 µs	510 µs
		10 (fperiph)	—	15.8 µs	63 µs	—	31.8 µs	127 µs	—	63.8 µs	255 µs
	10 (fc/4)	00 (fperiph/4)	—	63 µs	252 µs	—	127 µs	508 µs	—	255 µs	1020 µs
		01 (fperiph/2)	—	31.5 µs	126 µs	—	63.5 µs	254 µs	—	127.5 µs	510 µs
		10 (fperiph)	—	—	63 µs	—	—	127 µs	—	—	255 µs
	11 (fc/8)	00 (fperiph/4)	—	63 µs	252 µs	—	127 µs	508 µs	—	255 µs	1020 µs
		01 (fperiph/2)	—	—	126 µs	—	—	254 µs	—	—	510 µs
		10 (fperiph)	—	—	63 µs	—	—	127 µs	—	—	255 µs

Note 1: The prescaler's output clock ϕTn must be selected so that ϕTn < fsys/2 is satisfied.

Note 2: Do not change the clock gear value while the timer is running.

Note 3: The — character means "Don't use."

11.4.5 Operating Mode Summary

Table 11.4 shows the settings for the TMRA01 for each of the operating modes.

Table 11.4 Register Settings for Each Operating Mode

Register	TA01MOD				TA1FFCR
Field	TA01M[1:0]	PWM[01:00]	TA1CLK[1:0]	TA0CLK[1:0]	TAFF1IS
Function	Interval Timer Mode	PWM Period	UC1 Clock Source	UC0 Clock Source	Timer Flip-Flop Toggle-Trigger
8-Bit Timer × 2ch	00	—	Match output from UC0 φT1, φT16, φT256 (00, 01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	0: UC0 output 1: UC1 output
16-Bit Timer Mode	01	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	—
8-Bit PPG × 1ch	10	—	—	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	—
8-Bit PWM × 1ch 8-Bit Timer × 1ch (Note)	11	2^6-1, 2^7-1, 2^8-1 (01, 10, 11)	φT1, φT16, φT256 (01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	PWM output

– = Don't care

Note: In 8-bit PWM generation mode, the UC1 can be used as an 8-bit timer. However, the match-detect output from the UC0 can not be used as a clock source for the UC1, and the timer output is not avaialble for the UC1.

12. 16-Bit Timer/Event Counters (TMRBs)

The TMP1940CYAF has a 16-bit timer/event counter consisting of four identical channels (TMRB0–TMRB3). Each channel has the following three basic operating modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) mode

Each channel has the capture capability used to latch the value of the counter. The capture capability allows:

- Frequency measurement
- Pulse-width measurement
- Time difference measurement

Figure 12.1 to Figure 12.4 are block diagrams of the TMRB0 to TMRB3.

The main components of a TMRBn block are a 16-bit up-counter, two 16-bit timer registers (one of which is double-buffered), two 16-bit capture registers, two comparators, capture control logic, a timer flip-flop and its associated control logic.

Each channel is independently programmable and functionally equivalent except that the TMRB3 has no external clock/capture trigger inputs. Table 12.1 gives the pins and registers for the four channels. In the following sections, any references to the TMRB0 also apply to all the other channels.

Table 12.1 Pins and Registers for the Four TMRBn Channels

		TMRB0	TMRB1	TMRB2	TMRB3
External Pins	External clock / Capture trigger inputs	TB0IN0 (Shared with P74) TB0IN1 (Shared with P75)	TB1IN0 (Shared with P80) TB1IN1 (Shared with P81)	TB2IN0 (Shared with P83) TB2IN1 (Shared with P84)	—
	Timer flip-flop output	TB0OUT0 (Shared with P76)	TB1OUT0 (Shared with P82)	TB2OUT (Shared with P85)	TB3OUT (Shared with P86)
Registers (Addresses)	Timer Run register	TB0RUN (0xFFFF_F180)	TB1RUN (0xFFFF_F190)	TB2RUN (0xFFFF_F1A0)	TB3RUN (0xFFFF_F1B0)
	Timer Mode register	TB0MOD (0xFFFF_F182)	TB1MOD (0xFFFF_F192)	TB2MOD (0xFFFF_F1A2)	TB3MOD (0xFFFF_F1B2H
	Timer Flip-Flop Control register	TB0FFCR (0xFFFF_F183)	TB1FFCR (0xFFFF_F193)	TB2FFCR (0xFFFF_F1A3)	TB3FFCR (0xFFFF_F1B3)
	Timer registers	TB0RG0L (0xFFFF_F188) TB0RG0H (0xFFFF_F189) TB0RG1L (0xFFFF_F18A) TB0RG1H (0xFFFF_F18B)	TB1RG0L (0xFFFF_F198) TB1RG0H (0xFFFF_F199) TB1RG1L (0xFFFF_F19A) TB1RG1H (0xFFFF_F19B)	TB2RG0L (0xFFFF_F1A8) TB2RG0H (0xFFFF_F1A9) TB2RG1L (0xFFFF_F1AA) TB2RG1H (0xFFFF_F1AB)	TB3RG0L (0xFFFF_F1B8) TB3RG0H (0xFFFF_F1B9) TB3RG1L (0xFFFF_F1BA) TB3RG1H (0xFFFF_F1BB)
	Capture registers	TB0CP0L (0xFFFF_F18C) TB0CP0H (0xFFFF_F18D) TB0CP1L (0xFFFF_F18E) TB0CP1H (0xFFFF_F18F)	TB1CP0L (0xFFFF_F19C) TB1CP0H (0xFFFF_F19D) TB1CP1L (0xFFFF_F19E) TB1CP1H (0xFFFF_F19F)	TB2CP0L (0xFFFF_F1AC) TB2CP0H (0xFFFF_F1AD) TB2CP1L (0xFFFF_F1AE) TB2CP1H (0xFFFF_F1AF)	TB3CP0L (0xFFFF_F1BC) TB3CP0H (0xFFFF_F1BD) TB3CPIL (0xFFFF_FIBE) TB3CPIH (0xFFFF_FIBF)

12.1 Block Diagrams

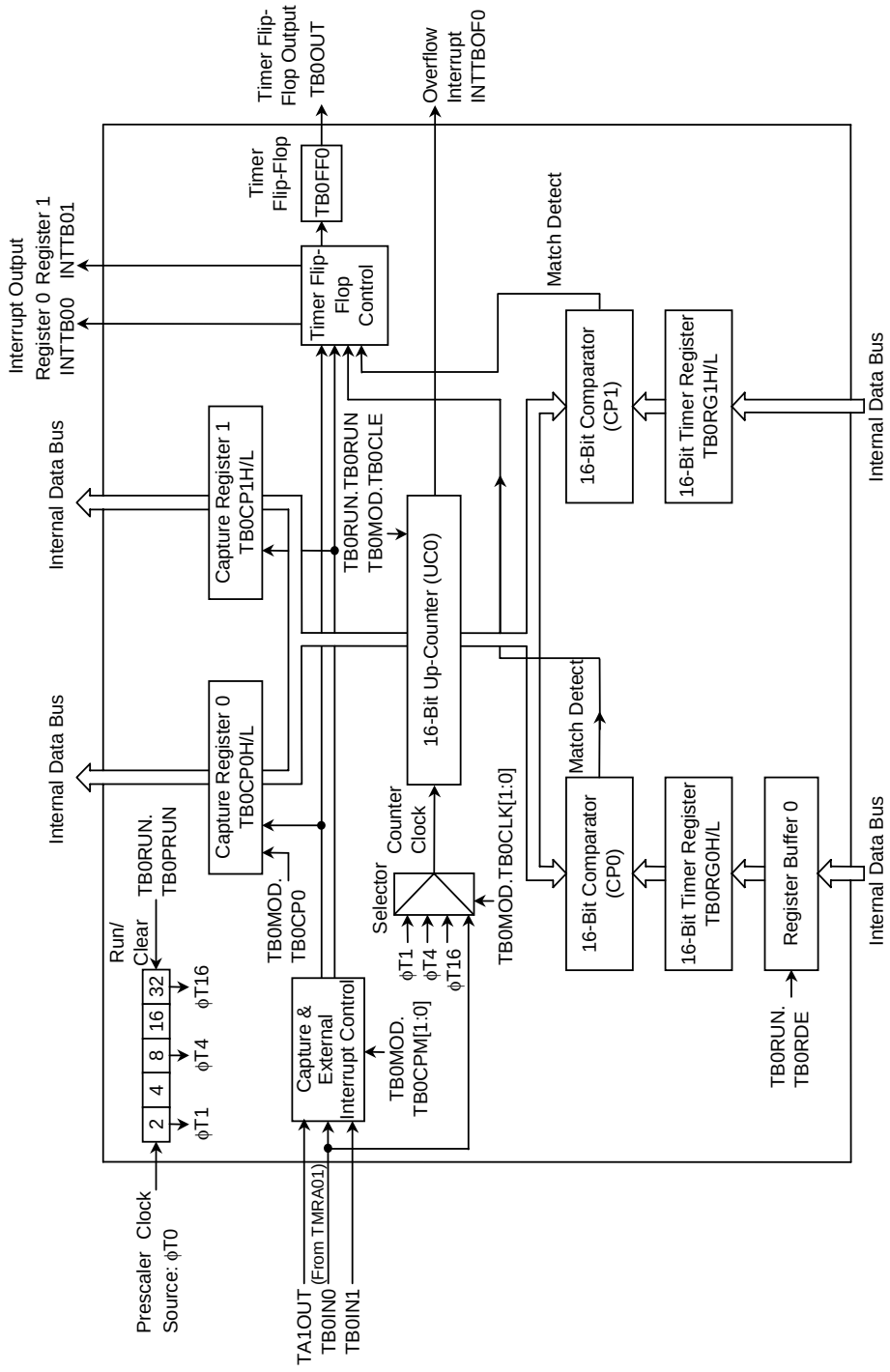


Figure 12.1 TMRB0 Block Diagram

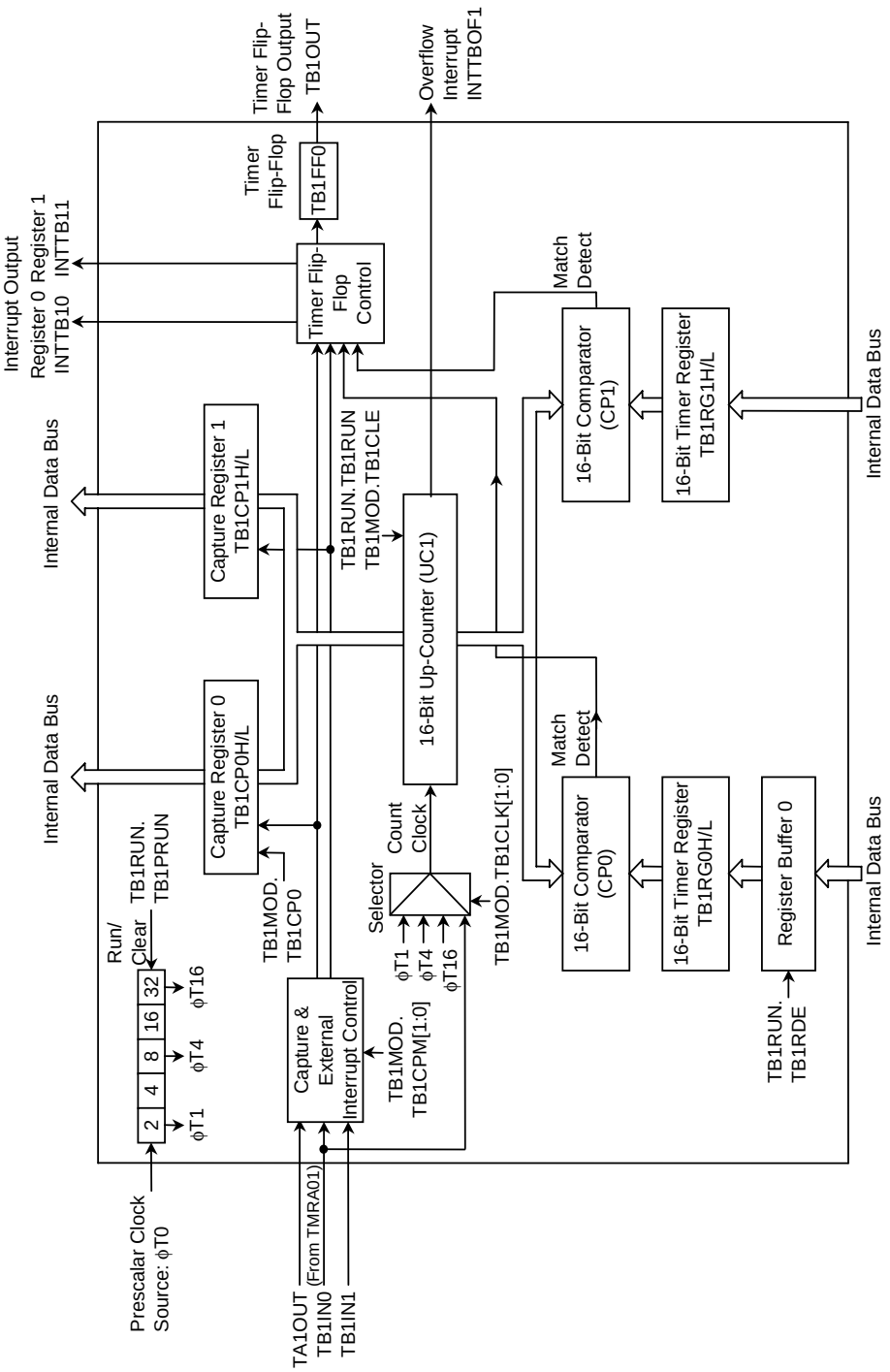


Figure 12.2 TMRB1 Block Diagram

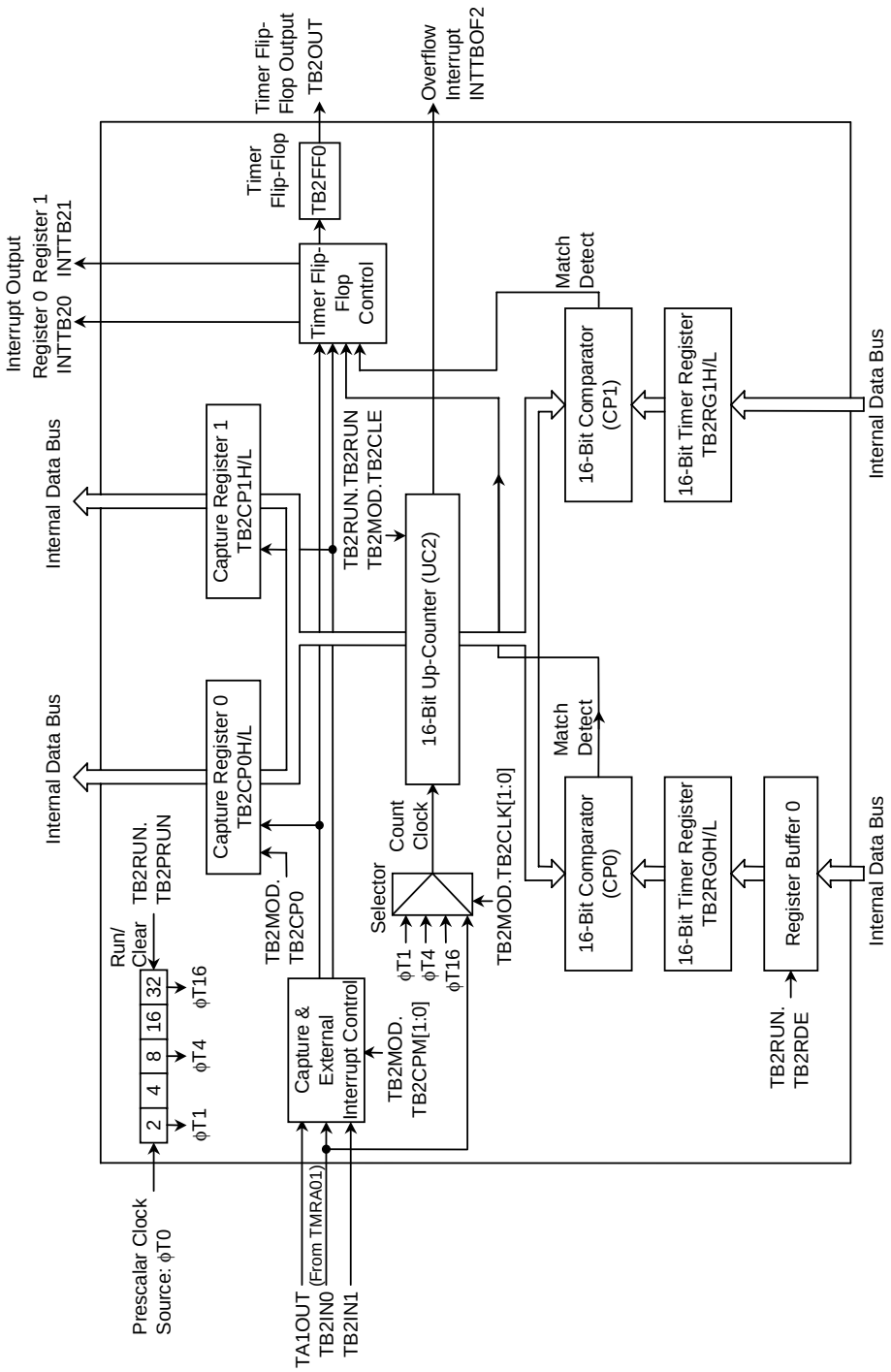


Figure 12.3 TMRB2 Block Diagram

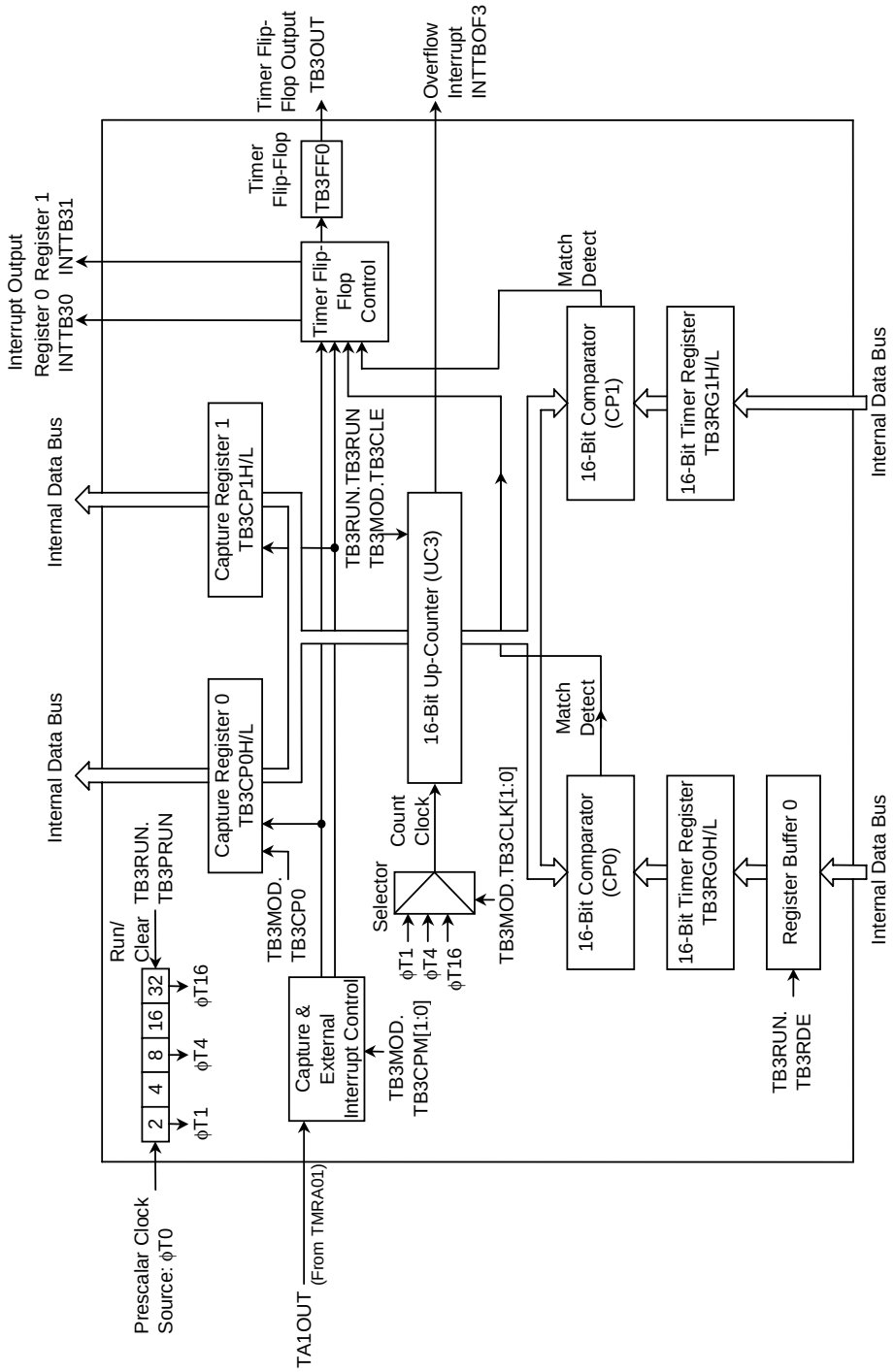


Figure 12.4 TMRB3 Block Diagram

12.2 Timer Components

12.2.1 Prescaler

The TMRB0 has a 5-bit prescalar that slows the rate of a clocking source to the counter. The prescalar clock source (ϕT0) can be selected from fperiph, fperiph/2 and fperiph/4 by programming the PRCK[1:0] field of the SYSCR0 located within the CG. fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The TB0RUN bit in the TB0RUN register allows the enabling and disabling of the TMRB0 prescalar. A write of 1 to this bit starts the prescalar. A write of 0 to this bit clears and halts the prescalar.

Prescalar output taps can be divide-by-2 (ϕT1), divide-by-8 (ϕT4) and divide-by-32 (ϕT16). Table 12.2 shows prescalar output clock resolutions (@fc = 32 MHz).

Table 12.2 Prescaler Output Clock Resolutions

@fc = 32 MHz

Peripheral Clock Select SYSCR1.FPSEL	Clock Gear Value SYSCR1.GEAR[1:0]	Prescaler Clock Source SYSCR0.PRCK[1:0]	Prescaler Output Clock Resolution		
			ϕT1	ϕT4	ϕT16
0 (gear)	00 (fc)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		01 (fperiph/2)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)
	01 (fc/2)	00 (fperiph/4)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
	10 (fc/4)	00 (fperiph/4)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)
		01 (fperiph/2)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		10 (fperiph)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
	11 (fc/8)	00 (fperiph/4)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32 μs)
		01 (fperiph/2)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)
		10 (fperiph)	—	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
1 (fc)	00 (fc)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		01 (fperiph/2)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)
	01 (fc/2)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		01 (fperiph/2)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)
	10 (fc/4)	00 (fperiph/4)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		01 (fperiph/2)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
		10 (fperiph)	—	—	$fc/2^5$ (1.0 μs)
	11 (fc/8)	00 (fperiph/4)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		01 (fperiph/2)	—	—	$fc/2^6$ (2.0 μs)
		10 (fperiph)	—	—	$fc/2^5$ (1.0 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that the relationship ϕTn < fsys/2 is satisfied.

Note 2: Do not change the clock gear value while the timer is running.

Note 3: The — character means "Don't use."

12.2.2 Up-Counter (UC0)

The TMRB0 contains a 16-bit binary up-counter, which is driven by a clock selected by the TB0CLK[1:0] field in the TB0MOD register. The clock input to the UC0 is either one of three prescalar outputs (ϕT1,ϕT4, ϕT16) or the external clock applied to the TB0IN0 pin. The clock input can be selected through the programming of the TB0CLK[1:0] field in the TB0MOD register.

The TB0RUN bit in the TB0RUN register is used to start the UC0 and to stop and clear the UC0. The UC0 is cleared to 0000H, if so enabled, when it reaches the value in the TB0RG1H/L register. The TB0CLE bit in the TB0MOD register allows the user to enable and disable this clearing. If it is disabled, the UC0 acts as a free-running counter.

An overflow interrupt (INTTBOF0) is generated upon a counter overflow.

Note: Programming the TB0CLK[1:0] and TB0CLE bits in the TB0MOD register should only be attempted when the timer is not running.

12.2.3 Timer Registers (TB0RG0H/L and TB0RG1H/L)

Each timer channel has two 16-bit timer registers containing a time constant. When the up-counter reaches the time constant value in each timer register, the associated comparator block generates a match-detect signal.

Each of the timer registers (TB0RG0H/L, TB0RG1H/L) can be written with either a halfword-store instruction or a series of two byte-store instructions. When byte-store instructions are used, the low-order byte must be stored first, followed by the high-order byte. The 16-bit timer registers are often simply referred to as TB0RG0 and TB0RG1 without the H and L suffix.

One of the two timer registers, TB0RG0, is double-buffered. The double-buffering function can be enabled and disabled through the programming of the TB0RDE bit in the TB0RUN: 0=disable, 1=enable.

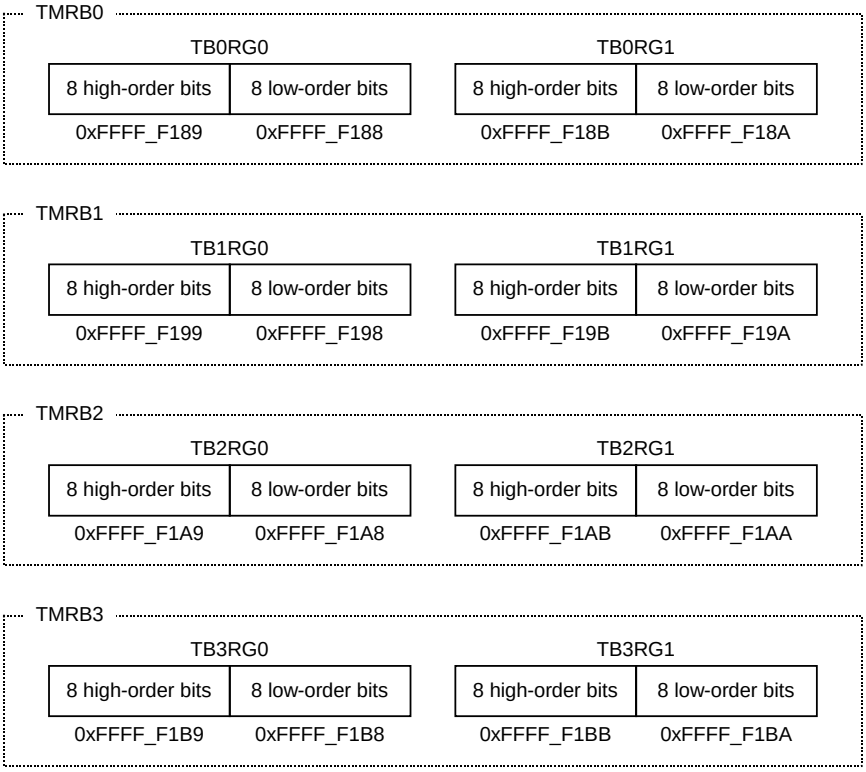
If double-buffering is enabled, the TB0RG0 latches a new time constant value from the register buffer. This takes place when a match is detected between the UC0 and the TB0RG1.

Upon reset, the contents of the TB0RG0 and TB0RG1 are undefined; thus, they must be loaded with valid values before the timer can be used. A reset clears the TB0RUN.TB0RDE bit to 0, disabling the double-buffering function. To use this function, the TB0RUN.TB0RDE bit must be set to 1 after loading the TB0RG0 and TB0RG1 with time constants. When TB0RUN.TB0RDE=1, the next time constant can be written to the register buffer.

Note 1: The TB0RG0 and the corresponding register buffer are mapped to the same address (0xFFFF_F188 thru 0xFFFF_F189). When TB0RUN.TB0RDE=0, a time constant value is written to both the TB0RG0 and the register buffer; when TB0RUN.TB0RDE=1, a time constant value is written only to the register buffer. Therefore, the double-buffering function should be disabled when writing an initial time constant to the timer register.

Note 2: Programming the TB0RDE bit should only be attempted when the timer is not running.

The following diagram shows the addresses of each timer register.



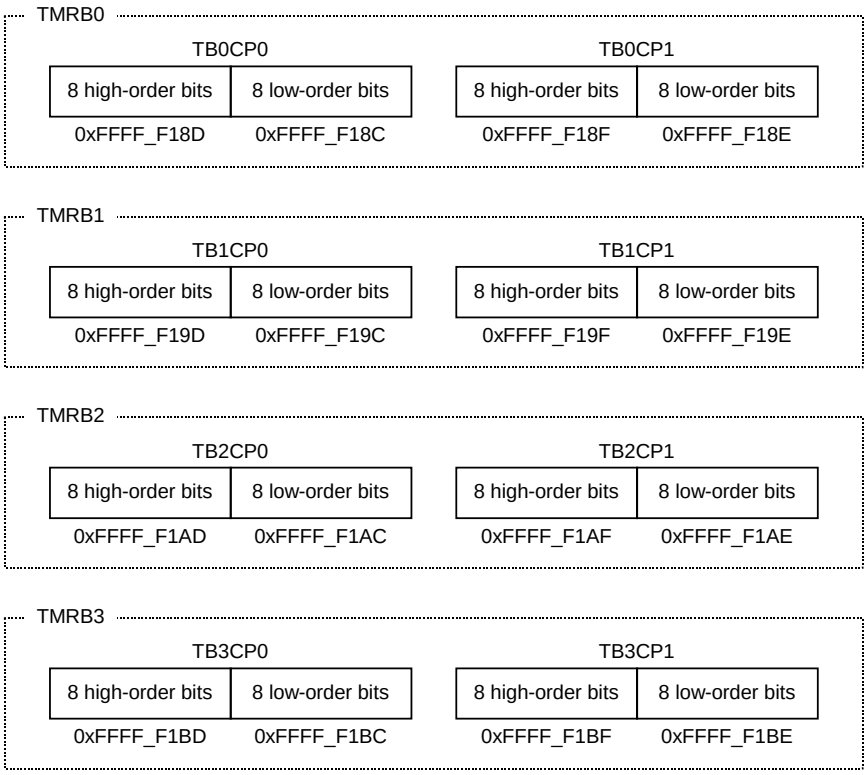
The Timer registers are write-only registers and cannot be read.

12.2.4 Capture Registers (TB0CP0H/L and TB0CP1H/L)

The capture registers are 16-bit registers used to latch the value of the up-counter (UC0).

Each of the capture registers can be read with either a halfword-load instruction or a series of two byte-load instructions. When byte-load instructions are used, the low-order byte must be read first, followed by the high-order byte. The 16-bit capture registers are often simply referred to as TBnCP and TBnCP1 without the H and L suffix.

The following diagram shows the addresses of each capture register.



The Capture registers are read-only registers and cannot be written by software.

12.2.5 Capture Control Logic

The capture control logic controls the capture of an up-counter (UC0) value into the capture registers (TB0CP0 and TB0CP1). The TB0CPM[1:0] field in the TB0MOD register selects a capture trigger input to be sensed by the capture control logic.

Futhermore, a counter value can be captured under software control; a write of 0 to the TB0MOD.TB0CP0 bit causes the current UC0 value to be latched into the TB0CP0. To use the capture capability, the prescalar must be running (i.e., TB0RUN.TB0PRUN=1).

Note 1: Reading the eight low-order bits of a capture register disables the capture capability. Reading the eight high-order bits thereafter re-enables the capture capability. The reading of a whole capture register should be completed during an interval between active transitions on the defined capture trigger input.

Note 2: Don't stop the timer after only reading the eight low-order bits of a capture register. If this is done, the capture capability continues to remain in the disabled state even after the timer is restarted.

Note 3: When the TB0IN0 pin is selected as a capture trigger input, it can not function as a timer clock source.

12.2.6 Comparators (CP0 and CP1)

The TMRB0 contains two 16-bit comparators. The CP0 block compares the output of the up-counter (UC0) with a time constant value in the TB0RG0. The CP1 block compares the output of the UC0 with a time constant value in the TB0RG1. When a match is detected, an interrupt (INTTB00/INTTB01) is generated.

12.2.7 Timer Flip-Flop (TB0FF0)

The timer flip-flop (TB0FF0) is toggled, if so enabled, upon assertion of match-detect signals from the comparators and latch signals from the capture control logic. The toggling of the TB0FF0 can be enabled and disabled through the programming of the TB0C1T1, TB0C0T1, TB0E1T1 and TB0E0T1 bits in the TB0FFCR register.

Upon reset, the TB0FF0 assumes an undefined state. The TB0FF0 can be initialized to 1 or 0 by writing 01 or 10 to the TB0FF0C[1:0] field in the TB0FFCR. A write of 01 to this field sets the TB0FF0; a write of 10 to this field clears the TB0FF0. Additionally, a write of 00 causes the TB0FF0 to be toggled to the opposite value.

The value of the TB0FF0 can be driven onto the TB0OUT pin, which is multiplxed with P76. The Port 7 registers (P7CR and P7FC) must be programmed to configure the P76/TB0OUT pin as TB0OUT.

Note: Programming the TB0FF0C[1:0] field should only be attempted when the timer is not running.

12.3 Register Description

TMRB0 Run register

TB0RUN (0xFFFF_F180)

	7	6	5	4	3	2	1	0
Name	TB0RDE	—	—	—	I2TB0	TB0PRUN	—	TB0RUN
Read/Write	R/W	R/W	—	—	R/W	R/W	—	R/W
Reset Value	0	0	—	—	0	0	—	0
Function	Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run

I2TB0: Timer on/off in IDLE mode
TB0PRUN: Prescaler
TB0RUN: TMRB0

Note: Bits 1, 4 and 5 are read as undefined.

TMRB1 Run register

TB1RUN (0xFFFF_F190)

	7	6	5	4	3	2	1	0
Name	TB1RDE	—	—	—	I2TB1	TB1PRUN	—	TB1RUN
Read/Write	R/W	R/W	—	—	R/W	R/W	—	R/W
Reset Value	0	0	—	—	0	0	—	0
Function	Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run

I2TB1: Timer on/off in IDLE mode
TB1PRUN: Prescaler
TB1RUN: TMBR1

Note: Bits 1, 4 and 5 are read as undefined.

Figure 12.5 Timer Run Registers

TMRB2 Run register

TB2RUN (0xFFFF_F1A0)

	7	6	5	4	3	2	1	0
Name	TB2RDE	—	—	—	I2TB2	TB2PRUN	—	TB2RUN
Read/Write	R/W	R/W	—	—	R/W	R/W	—	R/W
Reset Value	0	0	—	—	0	0	—	0
Function	Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run

I2TB2: Timer on/off in IDLE mode
TB2PRUN: Prescaler
TB2RUN: TMRB2

Note: Bits 1, 4 and 5 are read as undefined.

TMRB3 Run register

TB3RUN (0xFFFF_F1B0)

	7	6	5	4	3	2	1	0
Name	TB3RDE	—	—	—	I2TB3	TB3PRUN	—	TB3RUN
Read/Write	R/W	R/W	—	—	R/W	R/W	—	R/W
Reset Value	0	0	—	—	0	0	—	0
Function	Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run

I2TB3: Timer on/off in IDLE mode
TB3PRUN: Prescaler
TB3RUN: TMRB3

Note: Bits 1, 4 and 5 are read as undefined.

Figure 12.6 Timer Run Registers

TMRB0 Mode Register

TB0MOD (0xFFFF_F182)

	7	6	5	4	3	2	1	0
Name	—	—	TB0CP0	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
Read/Write	R/W		W*	R/W				
Reset Value	0	0	1	0	0	0	0	0
Function	Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB0IN0↑TB0IN1↑ 10: TB0IN0↑TB0IN0↓ 11: TA1OUT↑TA1OUT↓		UC0 clear control 0: Disable 1: Enable	TMRB0 clock source 00: TB0IN0 input 01: φT1 10: φT4 11: φT16	

Up-counter (UC0) clear control

0	Disabled
1	UC0 is reset upon a match with TB0RG1.

Capture triggers

00	Capture disabled
01	Latches UC0 value into TB0CP0 at rising edges of TB0IN0 Latches UC0 value into TB0CP1 at rising edges of TB0IN1.
10	Latches UC0 value into TB0CP0 at rising edges of TB0IN0. Latches UC0 value into TB0CP1 at falling edges of TB0IN0.
11	Latches UC0 value into TB0CP0 at rising edges of TA1OUT. Latches UC0 value into TB0CP1 at falling edges of TA1OUT.

Software capture

0	Latches UC0 value into TB0CP0.
1	Don't care

Figure 12.7 TMRB0 Mode Register

TMRB1 Mode Register

TB1MOD (0xFFFF_F192)

	7	6	5	4	3	2	1	0
Name	—	—	TB1CP0	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
Read/Write	R/W		W*	R/W				
Reset Value	0	0	1	0	0	0	0	0
Function	Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB1IN0↑TB1IN1↑ 10: TB1IN0↑TB1IN0↓ 11: TA1OUT↑TA1OUT↓		UC1 clear control 0: Disable 1: Enable	TMRB1 clock source 00: TB1IN0 input 01: φT1 10: φT4 11: φT16	

Up-counter (UC1) clear control

0	Disabled
1	UC1 is reset upon a match with TB1RG1.

Capture triggers

00	Capture disabled
01	Latches UC1 value into TB1CP0 at rising edges of TB1IN0 Latches UC1 value into TB1CP1 at rising edges of TB1IN1.
10	Latches UC1 value into TB1CP0 at rising edges of TB1IN0. Latches UC1 value into TB1CP1 at falling edges of TB1IN0.
11	Latches UC1 value into TB1CP0 at rising edges of TA1OUT. Latches UC1 value into TB1CP1 at falling edges of TA1OUT.

Software capture

0	Latches UC1 value into TB1CP0.
1	Don't care

Figure 12.8 TMRB1 Mode Register

TMRB2 Mode Register

TB2MOD (0xFFFF_F1A2)

	7	6	5	4	3	2	1	0
Name	—	—	TB2CP0	TB2CPM1	TB2CPM0	TB2CLE	TB2CLK1	TB2CLK0
Read/Write	R/W		W*	R/W				
Reset Value	0	0	1	0	0	0	0	0
Function	Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB2IN0↑TB2IN1↑ 10: TB2IN0↑TB2IN0↓ 11: TA1OUT↑TA1OUT↓		UC2 clear control 0: Disable 1: Enable	TMRB2 clock source 00: TB2IN0 input 01: φT1 10: φT4 11: φT16	

Up-counter (UC2) clear control

0	Disabled
1	UC2is reset upon a match with TB2RG1.

Capture triggers

	Capture Triggers
00	Capture disabled
01	Latches UC2 value into TB2CP0 at rising edges of TB2IN0. Latches UC2 value into TB2CP1 at rising edges of TB2IN1.
10	Latches UC2 value into TB2CP0 at rising edges of TB2IN0. Latches UC2 value into TB2CP1 at falling edges of TB2IN0.
11	Latches UC2 value into TB2CP0 at rising edges of TA1OUT. Latches UC2 value into TB2CP1 at falling edges of TA1OUT.

Software capture

0	Latches UC2 value into TB2CP0.
1	Don't care

Figure 12.9 TMRB2 Mode Register

TMRB3 Mode Register

TB3MOD (0xFFFF_F1B2)

	7	6	5	4	3	2	1	0
Name	—	—	TB3CP0	TB3CPM1	TB3CPM0	TB3CLE	TB3CLK1	TB3CLK0
Read/Write	R/W		W*	R/W				
Reset Value	0	0	1	0	0	0	0	0
Function	Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: Disabled 10: Disabled 11: TA1OUT↑TA1OUT↓		UC3 clear control 0: Disable 1: Enable	TMRB3 clock source 00: TB3IN0 input 01: ϕT1 10: ϕT4 11: ϕT16	

Up-counter (UC3) clear control

0	Disabled
1	UC3 is reset upon a match with TB3RG1.

Capture triggers

00	Capture disabled
01	Capture disabled
10	Capture disabled
11	Latches UC3 value into TB3CP0 at rising edges of TA1OUT. Latches UC3 value into TB3CP1 at falling edges of TA1OUT.

Software capture

0	Latches UC3 value into TB3CP0.
1	Don't care

Figure 12.10 TMRB3 Mode Register

TMRB0 Timer Flip-Flop Control Register

TB0FFCR (0xFFFF_F183)

	7	6	5	4	3	2	1	0
Name	—	—	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
Read/Write	W*		R/W				W*	
Reset Value	1	1	0	0	0	0	1	1
Function	Must be written as 11. * This field is always read as 11.		TB0FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB0FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
			UC0 →TB0CP1	UC0 → TB0CP0	UC0 = TB0RG1	UC0 = TB0RG0		

Timer flip-flop (TB0FF0) control

00	Toggles TB0FF0. (software toggle)
01	Sets TB0FF0 to 1.
10	Clears TB0FF0 to 0.
11	Don't care (read as 11)

When UC0 reaches TB0RG0 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC0 reaches TB0RG1 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC0 value is latched into TB0CP0 (see Note).

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC0 value is latched into TB0CP1.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

Note: Capturing the counter value into TB0CP0 via a software capture also generates a toggle-trigger to TB0FF0.

Figure 12.11 TMRB0 Timer Flip-Flop Control Register

TMRB1 Timer Flip-Flop Control Register

TB1FFCR (0xFFFF_F193)

	7	6	5	4	3	2	1	0
Name	—	—	TB1C1T1	TB1C0T1	TB1E1T1	TB1E0T1	TB1FF0C1	TB1FF0C0
Read/Write	W*		R/W				W*	
Reset Value	1	1	0	0	0	0	1	1
Function	Must be written as 11. * This field is always read as 11.		TB1FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB1FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
			UC1 → TB1CP1	UC1 → TB1CP0	UC1 = TB1RG1	UC1 = TB1RG0		

Timer flip-flop (TB1FF0) control

00	Toggles TB1FF0. (software toggle)
01	Set TB1FF0 to 1.
10	Clears TB1FF0 to 0.
11	Don't care (read as 11)

When UC1 reaches TB1RG0 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC1 reaches TB1RG1 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC1 value is latched into TB1CP0 (see Note).

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC1 value is latched into TB1CP1.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

Note: Capturing the counter value into TB1CP0 via a software capture also generates a toggle-trigger to TB1FF0.

Figure 12.12 TMRB1 Timer Flip-Flop Control Register

TMRB2 Timer Flip-Flop Control Register

TB2FFCR (0xFFFF_F193)

	7	6	5	4	3	2	1	0
Name	—	—	TB2C1T1	TB2C0T1	TB2E1T1	TB2E0T1	TB2FF0C1	TB2FF0C0
Read/Write	W*		R/W				W*	
Reset Value	1	1	0	0	0	0	1	1
Function	Must be written as 11. * This field is always read as 11.		TB2FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB2FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
			UC2 → TB2CP1	UC2 → TB2CP0	UC2 = TB2RG1	UC2 = TB2RG0		

Timer flip-flop (TB2FF0) control

00	Toggles TB2FF0. (software toggle)
01	Set TB2FF0 to 1.
10	Clears TB2FF0 to 0.
11	Don't care (read as 11)

When UC2 reaches TB2RG0 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC2 reaches TB2RG1 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC2 value is latched into TB2CP0 (see Note).

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC2 value is latched into TB2CP1.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

Note: Capturing the counter value into TB2CP0 via a software capture also generates a toggle-trigger to TB2FF0.

Figure 12.13 TMRB2 Timer Flip-Flop Control Register

TMRB3 Timer Flip-Flop Control Register

TB3FFCR (0xFFFF_F1B3)

	7	6	5	4	3	2	1	0
Name	—	—	TB3C1T1	TB3C0T1	TB3E1T1	TB3E0T1	TB3FF0C1	TB3FF0C0
Read/Write	W*		R/W				W*	
Reset Value	1	1	0	0	0	0	1	1
Function	Must be written as 11. * This field is always read as 11.		TB3FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB3FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
			UC3 → TB3CP1	UC3 → TB3CP0	UC3 = TB3RG1	UC3 = TB3RG0		

Timer flip-flop (TB3FF0) control

00	Toggles TB3FF0. (software toggle)
01	Set TB3FF0 to 1.
10	Clears TB3FF0 to 0.
11	Don't care (read as 11)

When UC3 reaches TB3RG0 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC3 reaches TB3RG1 value.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC3 value is latched into TB3CP0 (see Note).

0	Toggle-trigger disabled
1	Toggle-trigger enabled

When UC3 value is latched into TB3CP1.

0	Toggle-trigger disabled
1	Toggle-trigger enabled

Note: Capturing the counter value into TB3CP0 via a software capture also generates a toggle-trigger to TB3FF0.

Figure 12.14 TMRB3 Timer Flip-Flop Control Register

12.4 Operating Modes

12.4.1 16-Bit Interval Timer Mode

In the following example, the TMRB0 is used to accomplish periodic interrupt generation. The interval time is set in Timer Register 1 (TB0RG1), and the INTTB01 interrupt is enabled.

```
                  7  6  5  4  3  2  1  0
TB0RUN    ← 0  0  X  X  –  0  X  0    Stops the TMRB0.
IMC7LL    ← X  X  1  1  0  0  0  0    Enables INTTB01, sets its priority level to 4 and disables
IMC7LH    ← X  X  1  1  0  1  0  0    INTTB00.
TB0FFCR   ← 1  1  0  0  0  0  1  1    Disables the timer flip-flop toggle-trigger.
TB0MOD    ← 0  0  1  0  0  1  *  *    Selects a prescalar output clock as the timer clock source
                  (** = 01, 10, 11)   and disables the capture function.
TB0RG1    ← *  *  *  *  *  *  *  *    Sets the interval time.
            *  *  *  *  *  *  *  *    (16 bits)
TB0RUN    ← 0  0  X  X  –  1  X  1    Starts the TMRB0.
```

X = Don't care, – = No change

12.4.2 16-Bit Event Counter Mode

This mode is used to count events by interpreting the rising edges of the external counter clock (TB0IN0) as events.

The up-counter (UC0) counts up on each rising clock edge. The counter value is be latched into a capture register under software control. To determine the number of events (i.e., cycles) counted, the value in the capture register must be read.

```
                  7  6  5  4  3  2  1  0
TB0RUN    ← 0  0  X  X  –  0  X  0    Stops the TMRB0.
P7CR      ← –  –  –  0  –  –  –  –    Configures the P74 pin for input mode.
P7FC      ← –  –  –  1  –  –  –  –
IMC7LL    ← X  X  1  1  0  0  0  0    Enables INTTB01 (interrupt level = 4) and disables INTTB00.
IMC7LH    ← X  X  1  1  0  1  0  0
TB0FFCR   ← 1  1  0  0  0  0  1  1    Disables the timer flip-flop toggle-trigger.
TB0MOD    ← 0  0  1  0  0  1  0  0    Selects the TB0IN0 input as the timer clock source.
TB0RG1    ← *  *  *  *  *  *  *  *    Sets a count value (16 bits).
TB0RUN    ← 0  0  X  X  –  1  X  1    Starts the TMRB0.
```

X = Don't care, – = No change

Note: Even when the timer is used for event counting, the prescaler must be programmed to run (i.e., the TB0RUN.TB0PRUN bit must be set to 1).

12.4.3 16-Bit Programmable Pulse Generation (PPG) Mode

The 16-bit PPG mode can be used to generate a square wave with any frequency and duty cycle. The pulse can be high-going and low-going, as determined by the initial setting of the timer flip-flop (TB0FF0).

A square wave is generated by toggling the timer flip-flop every time the up-counter UC0 reaches the values in each timer register (TB0RG0 and TB0RG1). The square-wave output is driven to the TB0OUT pin. In this mode, the following relationship must be satisfied:

(TB0RG0 value) < (TB0RG1 value)

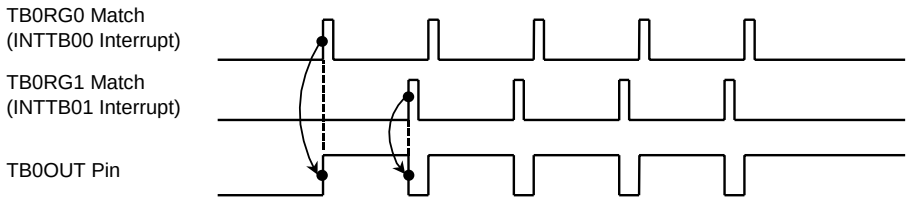


Figure 12.15 PPG Output Waveform

If the double-buffering function is enabled, the TB0RG0 value can be changed dynamically by writing a new value into the register buffer. Upon a match between the TB0RG1 and the UC0, the TB0RG0 latches a new value from the register buffer. The TB0RG0 can be loaded with a new value upon every match, thus making it easy to generate a square wave with virtually any duty cycle.

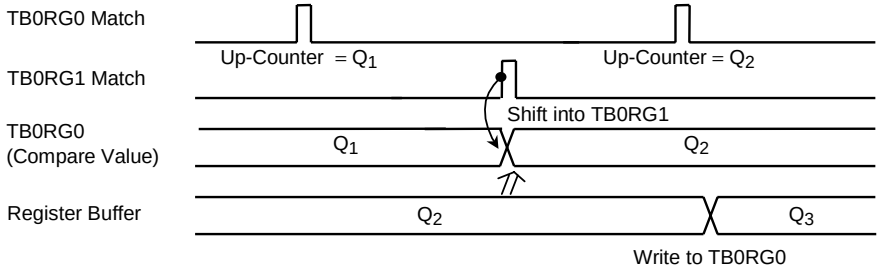


Figure 12.16 Register Buffer Operation

Figure 12.17 shows a functional diagram of 16-bit PPG mode.

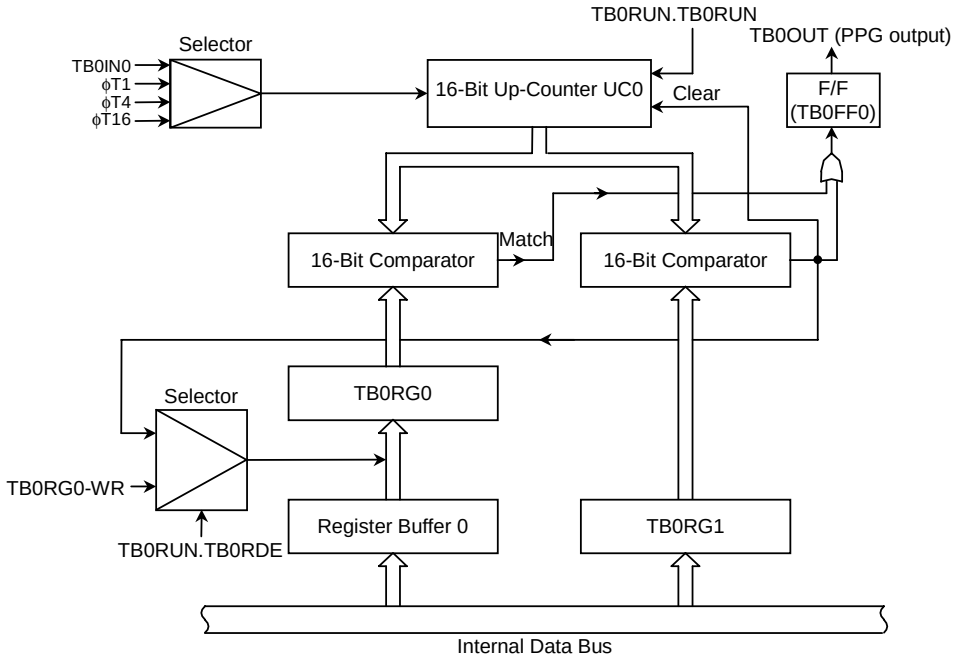


Figure 12.17 Functional Diagram of 16-Bit PPG Mode

The following is an example of running the timer in 16-bit PPG mode.

```
                  7  6  5  4  3  2  1  0
TB0RUN     ←      0  0  X  X  –  0  X  0
TB0RG0     ←      *  *  *  *  *  *  *  *
TB0RG1     ←      *  *  *  *  *  *  *  *
TB0RUN     ←      1  0  X  X  –  0  X  0
TB0FFCR    ←      X  X  0  0  1  1  1  0
TB0MOD     ←      0  0  1  0  0  1  *  *
                        (** = 01, 10, 11)
P7CR       ←      –  1  –  –  –  –  –  –
P7FC       ←      –  1  –  –  –  –  –  –
TB0RUN     ←      1  0  X  X  –  1  X  1
```

Disables the TB0RG0 double-buffering and stops the TMRB0.

Defines the duty cycle (16 bits).

Defines the cycle period (16 bits).

Enables the TB0RG0 double-buffering. (The duty cycle and cycle period are changed by the INTTB01 interrupt.)

Toggles the TB0FF0 when a match is detected between UC0 and TB0RG0 and between UC0 and TB0RG1. Initially clears the TB0FF0 to 0.

Selects a prescaler output clock as the timer clock source and disables the capture function.

Configures the P76 pin as TB1OUT.

Starts the TMRB0.

X = Don't care, – = No change

12.4.4 Timing and Measurement Functions Using the Capture Capability

The capture capability of the TMRBn provides versatile timing and measurement functions, including the following:

- One-shot pulse generation using an external trigger pulse
- Frequency measurement
- Pulse width measurement
- Time difference measurement

(1) One-Shot Pulse Generation Using an External Trigger Pulse

The TMRBn can be used to produce a one-time pulse as follows.

The 16-bit up-counter (UC0) is programmed to function as a free-running counter, clocked by one of the prescalar outputs. The TB0IN0 pin is used as an active-high external trigger pulse input for latching the counter value into Capture Register 0 (TB0CP0).

The TB0IN0 pin is shared with P74 and INT5. The Interrupt Controller (INTC) must be programmed to generate an INT5 interrupt upon detection of a rising edge on the TB0IN0/INT5 pin. A one-shot pulse has a delay and width controlled by the values stored in the timer registers (TB0RG0 and TB0RG1). Programming the TB0RG0 and TB0RG1 is the responsibility of the INT5 interrupt handler. The TB0RG0 is loaded with the sum of the TB0CP0 value (c) plus the pulse delay (d) – i.e., (c) + (d). The TB0RG1 is loaded with the sum of the TB0RG0 value plus the pulse width (p) – i.e., (c) + (d) + (p).

Next, the TB0E1T1 and TB0E0T1 bits in the Timer Flip-Flop Control register (TB0FFCR) are set to 11, so that the timer flip-flop (TB0FF0) will toggle when a match is detected between the UC0 and the TB0RG0 and between the UC0 and the TB0RG1. With the TB0FF0 toggled twice, a one-shot pulse is produced. Upon a match between the UC0 and the TB0RG1, the TMRB0 generates the INTTB01 interrupt, which must disable the toggle-trigger for the TB0FF0.

Figure 12.18 depicts one-shot pulse generation, with annotations showing (c), (d) and (p).

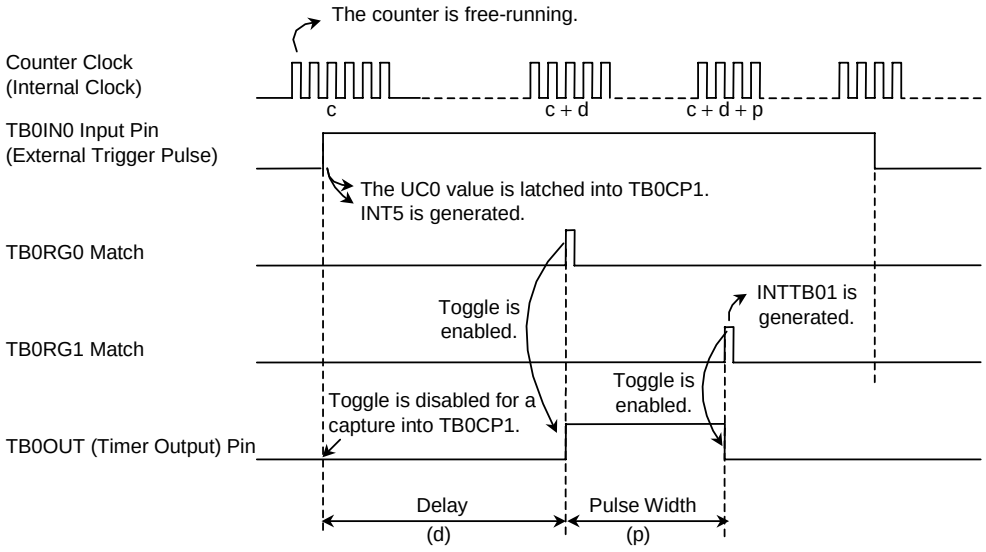


Figure 12.18 One-Shot Pulse Generation (with a Delay)

Example: Generating a one-shot pulse with a width of 2 ms and a delay of 3 ms on assertion of an external trigger pulse on the TB0IN0 pin

Clocking conditions:

System clock: High-speed (fc)
High-speed clock gear: ×1 (fc)
Prescaler clock: fperiph/4 (fperiph = fsys)

Settings in the main routine

```
                7  6  5  4  3  2  1  0
TB0MOD    ← X  X  1  0  1  0  0  1
TB0FFCR   ← X  X  0  0  0  0  1  0
P7CR      ← –  1  –  –  –  –  –  –
P7FC      ← –  1  –  –  –  –  –  –
IMC2HL    ← X  X  1  1  0  1  0  0
IMC7LL    ← X  X  1  1  0  0  0  0
IMC7LH    ← X  X  1  1  0  0  0  0
TB0RUN    ← –  0  X  X  –  1  X  1
```

- TB0MOD: Places the counter in free-running mode. / Selects φT1 as the counter clock source. / Latches UC0 value into TB0CP0 at rising edges of the TB0IN0 input.
- TB0FFCR: Clears TB0FF0 to 0. / Disables the toggle-trigger for TB0FF0.
- P7CR, P7FC: Configures the P76 pin as TB1OUT.
- IMC2HL, IMC7LL, IMC7LH: Enables INT5 and disables INTTB00 and INTTB01.
- TB0RUN: Starts the TMRB0.

Settings in INT5

```
TB0RG0    ← TB0CP0 + 3ms/φT1
TB0RG1    ← TB0RG0 + 2ms/φT1
TB0FFCR   ← X  X  –  –  1  1  –  –
IMC7LH    ← X  X  1  1  0  1  0  0
```

- TB0FFCR: Enables the TB0FF0 toggle-trigger for TB0RG0 and TB0RG1 matches.
- IMC7LH: Enables INTTB01.

Settings in INTTB01

```
TB0FFCR   ← X  X  –  –  0  0  –  –
IMC7LH    ← X  X  1  1  0  0  0  0
```

- TB0FFCR: Disables the TB0FF0 toggle-trigger for TB0RG0 and TB0RG1 matches.
- IMC7LH: Disables INTTB01.

X = Don't care, – = No change

If no delay is necessary, enable the TB0FF0 toggle-trigger for a capture of the UC0 value into the TB0CP0. Use the INT5 interrupt to load the TB0RG1 with a sum of the TB0CP0 value (c) plus the pulse width (p) and to enable the TB0FF0 toggle-trigger for a match between the UC0 and TB0RG1 values. A match generates the INTTB01 interrupt, which then is to disable the TB0FF0 toggle-trigger.

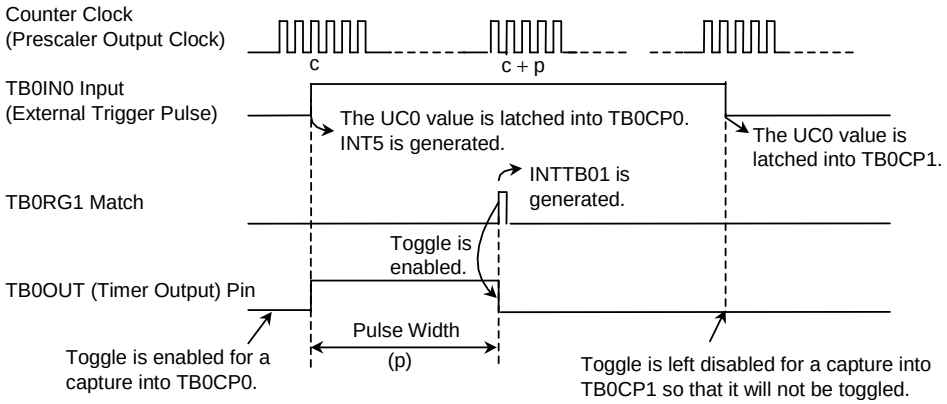


Figure 12.19 One-Shot Pulse Generation (without a Delay)

(2) Frequency Measurement

The capture function can be used to measure the frequency of an external clock. Frequency measurement requires a 16-bit TMRBn channel running in event counter mode and the 8-bit TMRA01. The timer flip-flop (TA1FF) in the TMRA01 is used to define the duration during which a measurement is taken.

Select the TB0IN0 pin as the clock source for the TMRB0. Set the TB0CPM[1:0] field in the TB0MOD to 11 to select the TA1FF output signal from the TMRA01 as a capture trigger input. This causes the TMRB0 to latch the 16-bit up-counter (UC0) value into Capture Register 0 (TB0CP0) on the low-to-high transition of the TA1FF and into Capture Register 1 (TB0CP1) on the next high-to-low transition of the TA1FF.

Either the INTTA0 or INTTA1 interrupt generated by the 8-bit timer can be used to make a frequency calculation.

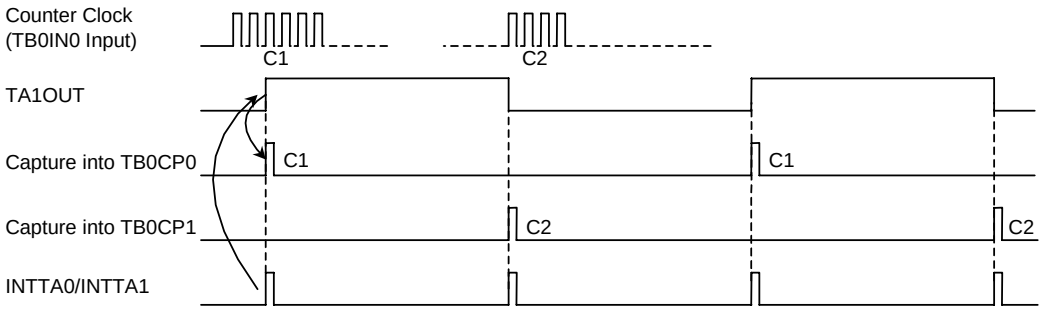


Figure 12.20 Frequency Measurement

For example, if the TA1FF of the 8-bit timer is programmed to be at logic 1 for a period of 0.5 seconds and the difference between the values captured into the TB0CP0 and TB0CP1 is 100, then the TB0IN0 frequency is calculated as 100 ÷ 0.5 s = 200 Hz.

(3) Pulse Width Measurement

The capture function can be used to measure the pulse width of an external clock. The external clock is applied to the TB0IN0 pin. The up-counter (UC0) is programmed to operate as a free-running counter, clocked by one of the prescalar outputs. The capture function is used to latch the UC0 value into Capture Register 0 (TB0CP0) at the clock rising edge and into Capture Register 1 (TB0CP1) at the next clock falling edge. The TB0IN0 input is shared with the INT5 input; the Interrupt Controller (INTC) is to be programmed to generate the INT5 interrupt at the falling edge of the TB0IN0 input.

Multplying the counter clock period by the difference between the values captured into the TB0CP0 and TB0CP1 gives the high pulse width of the TB0IN0 clock.

For example, if the prescalar output clock has a period of 0.5 μs and the difference between the TB0CP0 and TB0CP1 is 100, the high pulse width is calculated as 0.5 μs × 100 = 50 μs.

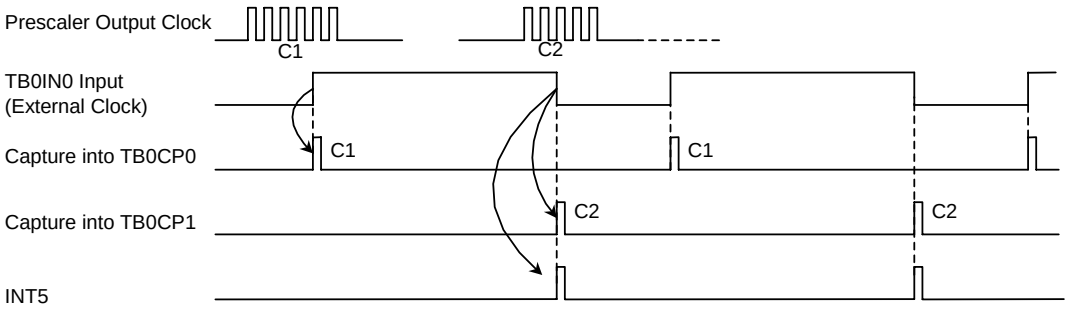


Figure 12.21 Pulse Width Measurement

The low pulse width can be measured by the second INT5 interrupt. This is accomplished by multiplying the counter clock period by the difference between the TB0CP0 value at the first C2 and the TB0CP1 value at the second C1.

(4) Time Difference Measurement

The capture function can be used to measure the time difference between two event occurrences. The 16-bit up-counter (UC0) is programmed to operate as a free-running counter. The UC0 value is latched into Capture Register 0 (TB0CP0) on the rising edge of TB0IN0. The TB0IN0 pin is shared with INT5; the Interrupt Controller (INTC) is to be programmed to generate the INT5 interrupt at this time.

Then, the UC0 value is latched into Capture Register 1 (TB0CP1) on the rising edge of TB0IN1. The TB0IN1 pin is shared with INT6; the INTC is to be programmed to generate the INT6 interrupt at this time.

The time difference between the two events that occurred on the TB0IN0 and TB0IN1 pins is calculated by multiplying the counter clock period by the difference between the TB0CP1 and TB0CP0 values.

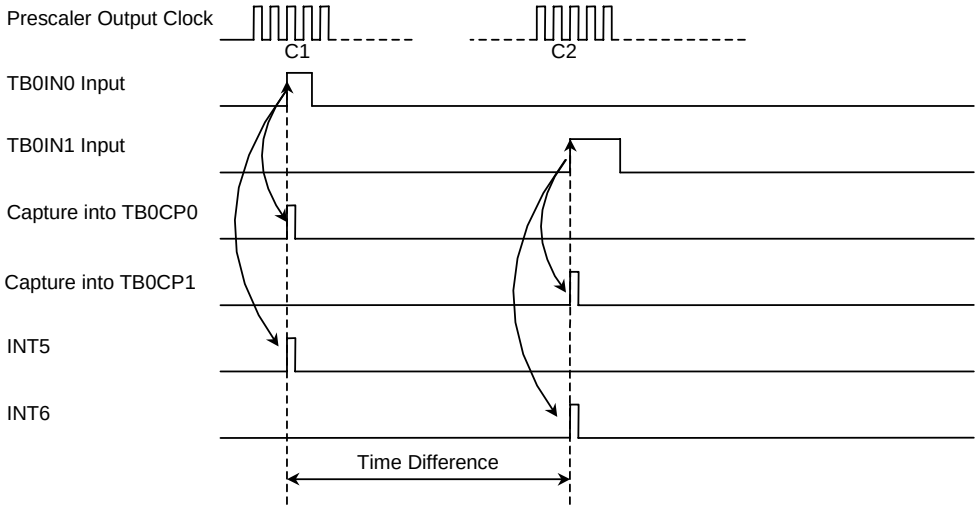


Figure 12.22 Time Difference Measurement

13. Serial I/O (SIO)

The TMP1940CYAF serial I/O contains four channels named SIO0, SIO1, SIO3 and SIO4 (there is not SIO2). The SIO0 and SIO1 provide Universal Asynchronous Receiver/Transmitter (UART) mode and synchronous I/O Interface mode. The SIO2 and SIO3 provide only UART mode.

- I/O Interface Mode

  Mode 0: Transmits/receives a serial clock (SCLK) as well as data streams for a synchronous clock mode of operation.

- UART mode

  Mode 1: 7 data bits

  Mode 2: 8 data bits

  Mode 3: 9 data bits

In Mode 1 and Mode 2, each character can include a parity bit. In Mode 3, an SIO channel operates in a wake-up mode for multidrop applications in which a master station is connected to several slave stations through a serial link.

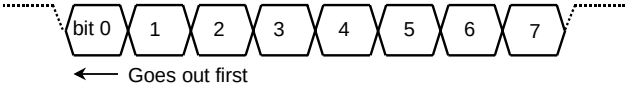
Figure 13.2 to Figure 13.5 are block diagrams of each SIO channel. The main components of an SIO channel are a clock prescalar, a serial clock generator, a receive buffer, a receive controller, a transimit buffer and a transmit controller.

Each SIO channel is independently programmable, and functionally equivalent with a few exceptions listed below. In the following sections, any references to the SIO0 also apply to the other channels.

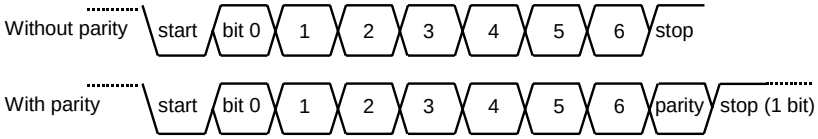
Table 13.1 Differences Between the SIO Channels

	SIO0	SIO1	SIO3	SIO4
Pins Used	TXD0 (P90) RXD0 (P91) $\overline{\text{CTS0}}$ /SCLK0 (P92)	TXD1 (P93) RXD1 (P94) $\overline{\text{CTS1}}$/SCLK1 (P95)	TXD3 (P70) RXD3 (P71)	TXD4 (P72) RXD4 (P73)
I/O Interface Mode	Available	Available	Not available	Not available

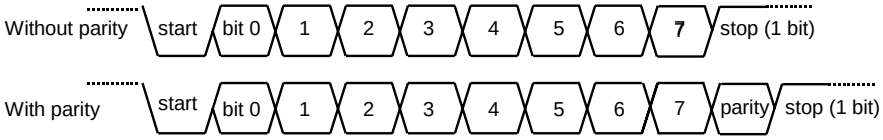
- Mode 0 (I/O Interface Mode)



- Mode 1 (7-Bit UART Mode)



- Mode 2 (8-Bit UART Mode)



- Mode 3 (9-Bit UART Mode)

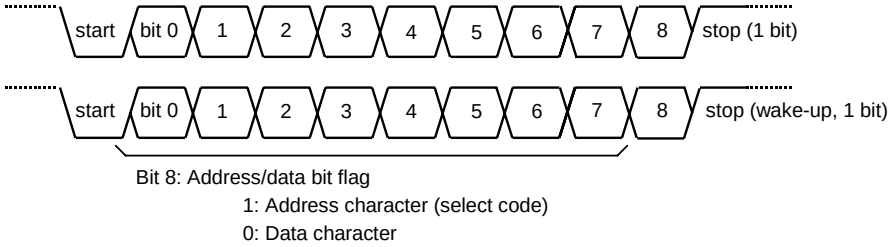


Figure 13.1 Data Formats

13.1 Block Diagrams

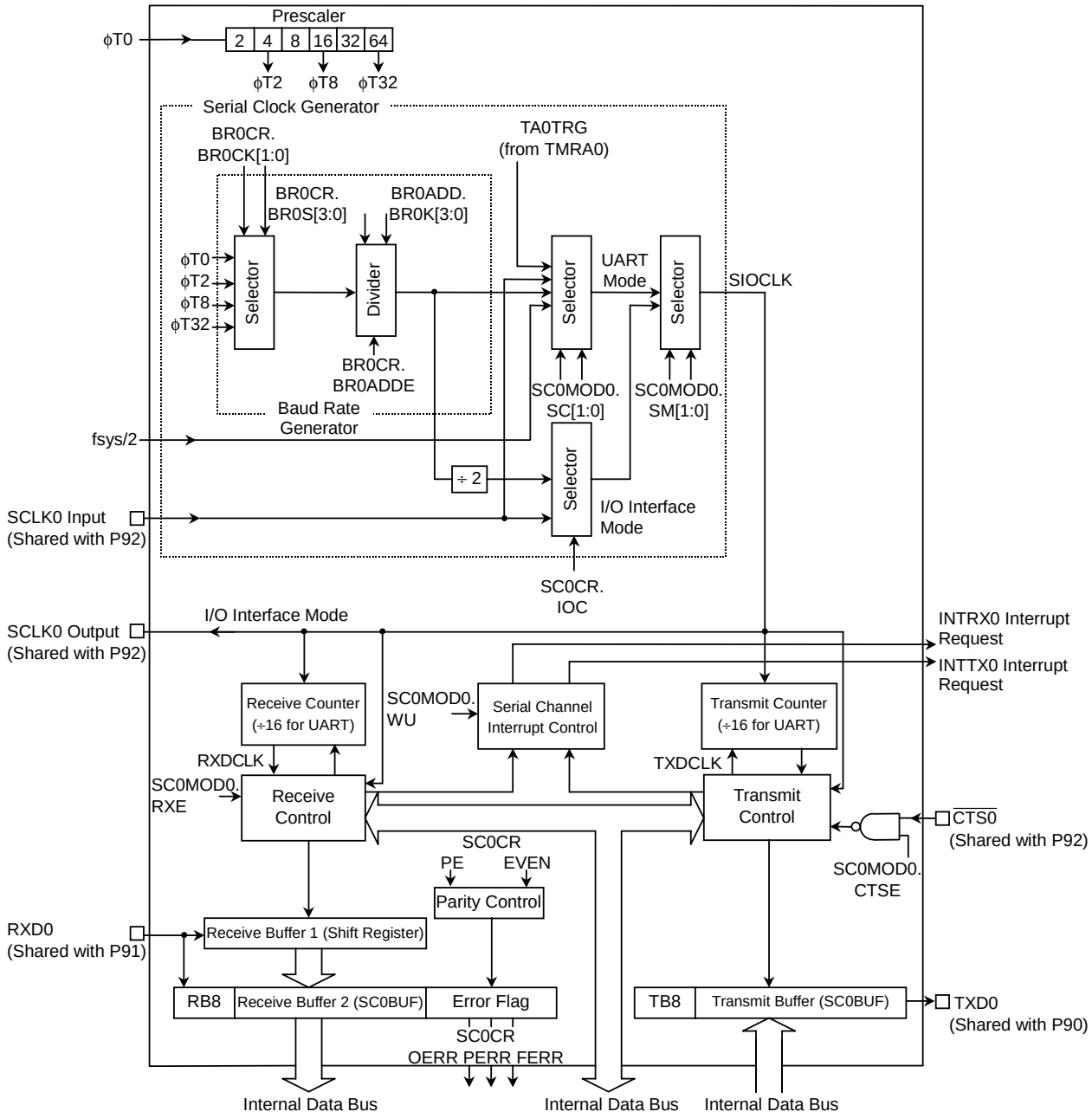


Figure 13.2 SIO0 Block Diagram

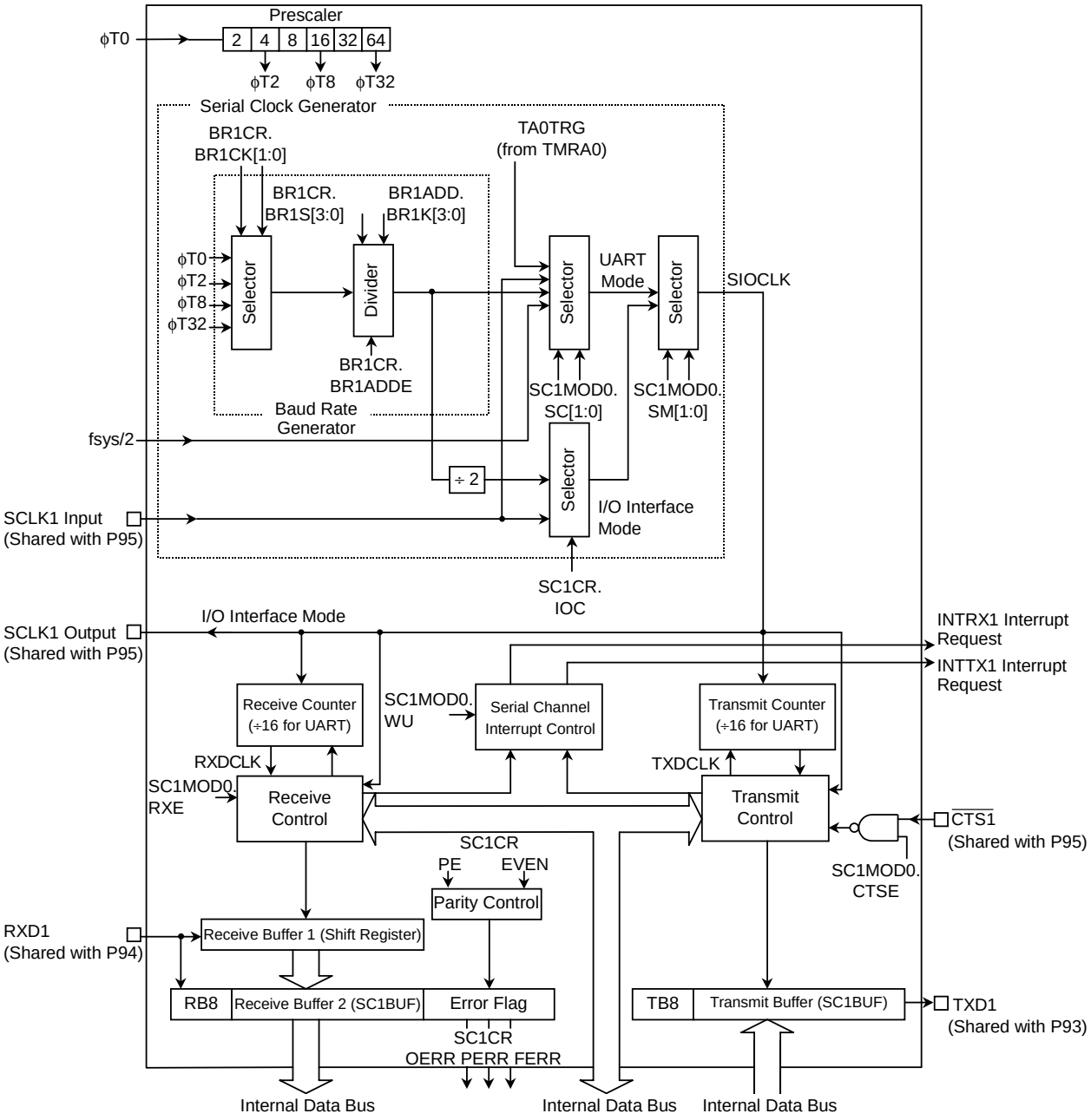


Figure 13.3 SIO1 Block Diagram

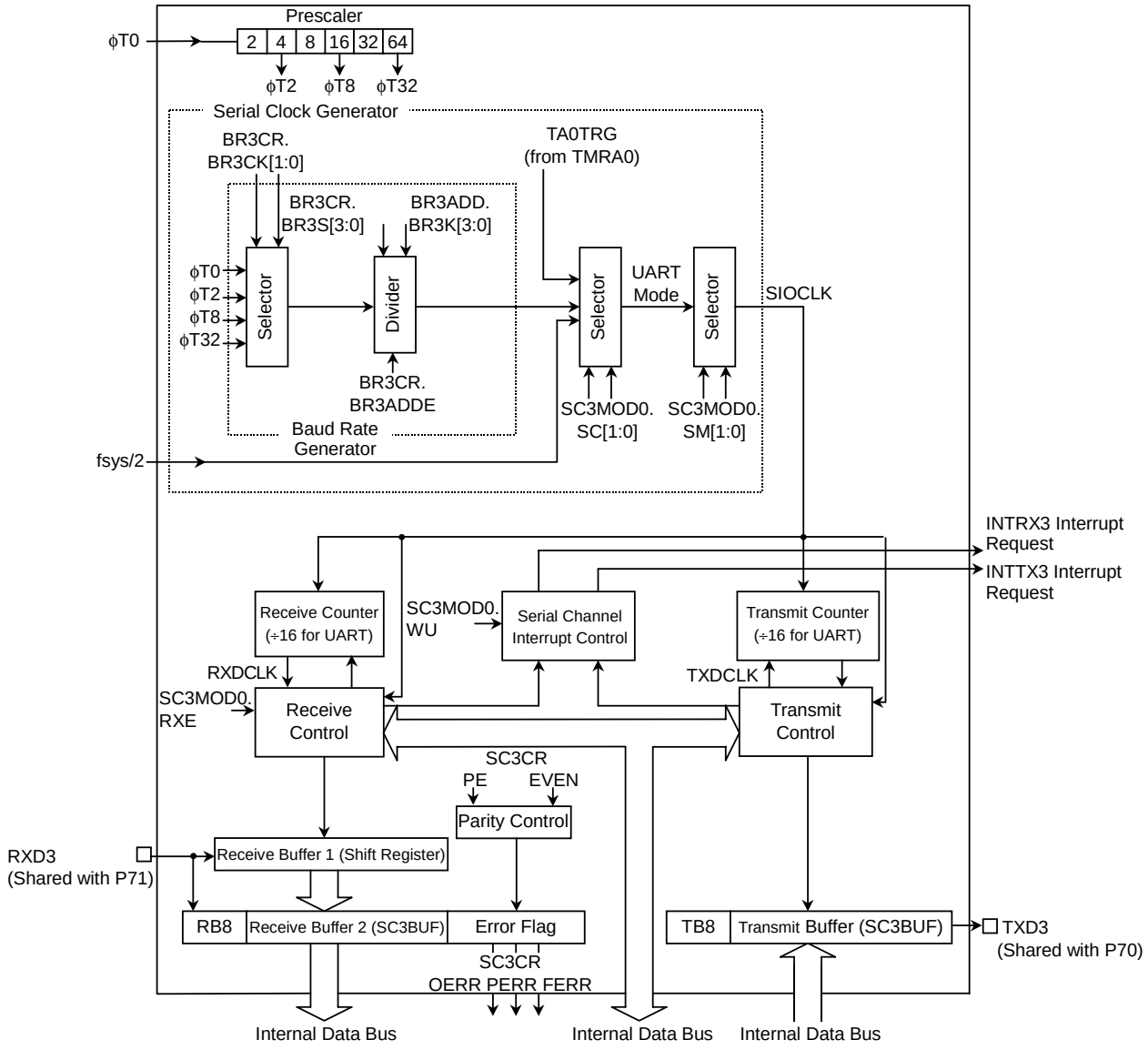


Figure 13.4 SIO3 Block Diagram

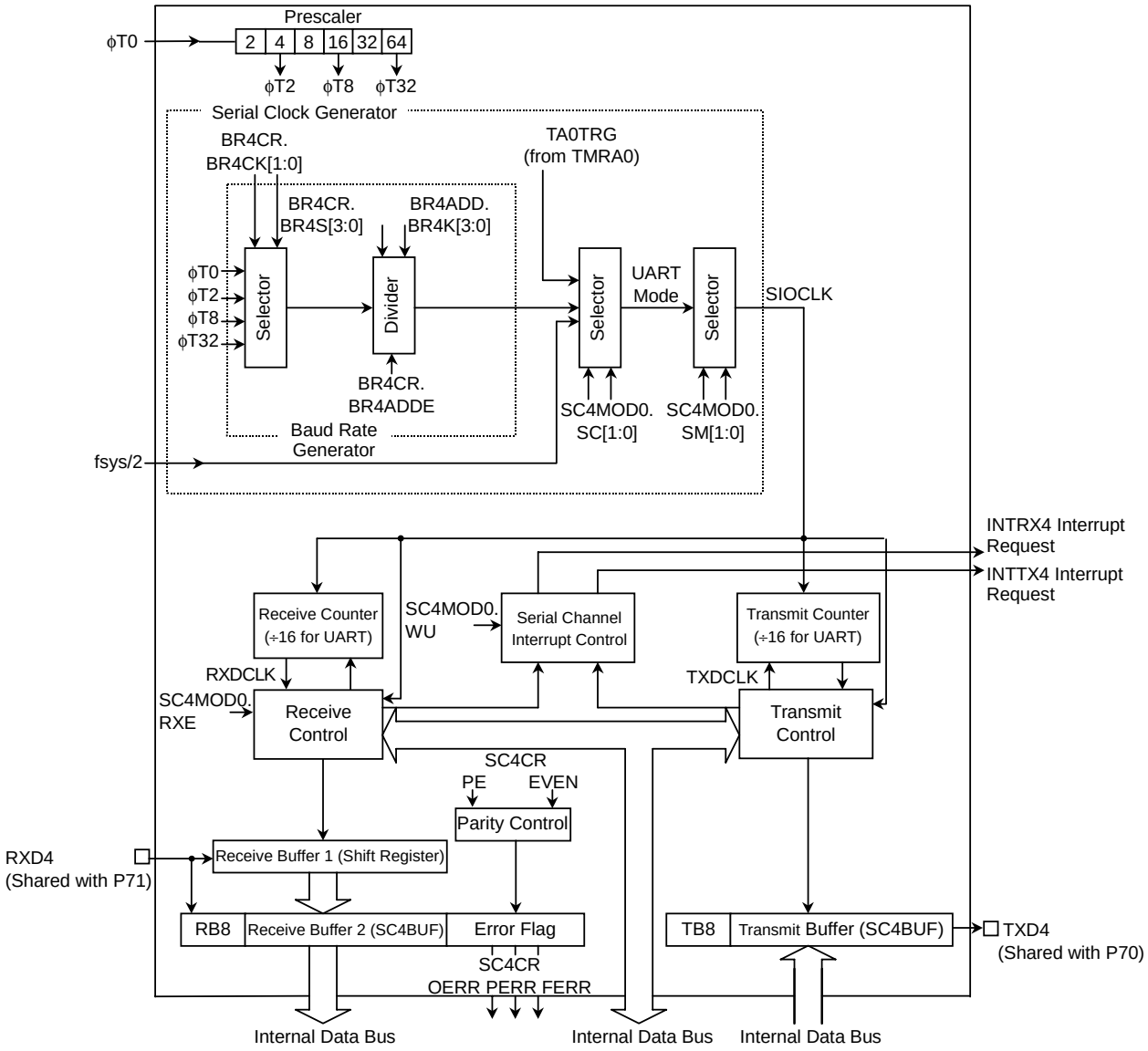


Figure 13.5 SIO4 Block Diagram

13.2 SIO Components

13.2.1 Prescaler

The SIO0 has a 6-bit prescalar that slows the rate of a clocking source to the serial clock generator. The prescalar clock source (ϕT0) can be selected from fperiph, fperiph/2 and fperiph/4 by programming the PRCK[1:0] field of the SYSCR0 located within the CG. fperiph can be selected from fgear (geared clock) and fc (non-geared clock) by programming the FPSEL bit of the SYSCR1 located within the CG.

The serial clock is selectable from several clocks; the prescalar is only enabled when the baud rate generator output clock is selected as a serial clock. Table 13.2 shows prescalar output clock resolutions (@fc = 32 MHz).

Table 13.2 Prescaler Output Clock Resolutions

@ fc = 32 MHz

Peripheral Clock Select SYSCR1.FPSEL	Clock Gear Value SYSCR1.GEAR[1:0]	Prescaler Clock Source SYSCR0.PRCK[1:0]	Prescaler Output Clock Resolution			
			ϕT0	ϕT2	ϕT8	ϕT32
0 (fgear)	00 (fc)	00 (fperiph/4)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
	01 (fc/2)	00 (fperiph/4)	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)
		01 (fperiph/2)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		10 (fperiph)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
	10 (fc/4)	00 (fperiph/4)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32 μs)
		01 (fperiph/2)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)
		10 (fperiph)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
	11 (fc/8)	00 (fperiph/4)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)	$fc/2^{11}$ (64 μs)
		01 (fperiph/2)	—	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)	$fc/2^{10}$ (32 μs)
		10 (fperiph)	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)	$fc/2^9$ (16 μs)
1 (fc)	00 (fc)	00 (fperiph/4)	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	$fc/2^2$ (0.125 μs)	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
	01 (fc/2)	00 (fperiph/4)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	—	$fc/2^3$ (0.25 μs)	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
	10 (fc/4)	00 (fperiph/4)	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	—	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	—	$fc/2^4$ (0.5 μs)	$fc/2^6$ (2.0 μs)
	11 (fc/8)	00 (fperiph/4)	—	—	$fc/2^6$ (2.0 μs)	$fc/2^8$ (8.0 μs)
		01 (fperiph/2)	—	—	$fc/2^5$ (1.0 μs)	$fc/2^7$ (4.0 μs)
		10 (fperiph)	—	—	—	$fc/2^6$ (2.0 μs)

Note 1: The prescaler's output clock ϕTn must be selected so that the relationship ϕTn < fsys/2 is satisfied.

Note 2: Do not change the clock gear value while the timer is running.

Note 3: The — character means "Don't use."

Prescalar output taps can be divide-by-1 (ϕT0), divide-by-4 (ϕT2), divide-by-16 (ϕT8) and divide-by-64 (ϕT32).

13.2.2 Baud Rate Generator

(1) Baud Rate Generator Configuration

The frequency used to transimit and receive data through the SIO0 is derived from the baud rate generator. The clock source for the baud rate generator can be selected from the 6-bit prescalar outputs (ϕT0, ϕT2, ϕT8, ϕT32) through the programming of the BR0CK[1:0] field in the BR0CR.

The baud rate generator contains a clock divider that can divide the selected clock by 1, n + (m / 16), or 16 (where n is an integer between 2 and 15, and m is an integer between 0 and 15). The clock divisor is programmed into the BR0ADDE and BR0S[3:0] bits in the BR0CR and the BR0K[3:0] bits in the BR0ADD.

- UART Mode

a. When BR0CR.BR0ADDE = 0

When the BR0CR.BR0ADDE bit is cleared, the BR0ADD.BR0K[3:0] field has no meaning or effect. In this case, the baud rate generator input clock is divided down by a value of N (1 to 16) programmed in the BR0CR.BR0S[3:0] field.

b. When BR0CR.BR0ADDE = 1

Setting the BR0CR.BR0ADDE bit enables the N + (16 – K) / 16 clock division function. The baud rate generator input clock is divided down according to the value of N (2 to 15) programmed in the BR0CR.BR0S[3:0] field and the value of K (1 to 15) programmed in the BR0ADD.BR0K[3:0] field.

Note: Setting N to 0 or 16 disables the N + (16 – K) / 16 clock division function. When N = 0 or 16, the BR0CR.BR0ADDE bit must be cleared.

- I/O Interface Mode

I/O Interface mode can not utilize the N + (16 – K) / 16 clock division function. The BR0CR.BR0ADDE must be cleared, so the baud rate generator input clock is divided down by a value of N (1 to 16) programmed in the BR0CR.BR0S[3:0] field.

(2) Baud Rate Calculations

- UART Mode

$$\text{Baud Rate} = \frac{\text{baud rate generator input clock}}{\text{baud rate generator divisor}} \div 16$$

When the clock input to the baud rate generator is 8-MHz ϕT0, the maximum baud rate is 500 kbps (with no clock division by the baud rate generator).

The baud rate generator can by bypassed if the user wants to use the fsys/2 clock as a serial clock. In this case, the maximum baud rate is 1 Mbps @fsys = 32 MHz.

- I/O Interface Mode

$$\text{Baud Rate} = \frac{\text{baud rate generator input clock}}{\text{baud rate generator divisor}} \div 2$$

When the clock input to the baud rate generator is 8-MHz ϕT0, the maximum baud rate is 2 Mbps (with the clock divided by 2 by the baud rate generator).

(3) Calculation Examples

- Integral Clock Division (Divide-by-N)
  fperiph = 24.576-MHz fc
  ϕT0 = fperiph/4
  Baud rate generator input clock: ϕT2
  Clock divisor N (BR0CR.BR0S[3:0]) = 10
  BR0CR.BR0ADDE = 0

  Clocking conditions

  System clock: High-speed (fc)
  High-speed clock gear: ×1 (fc)
  Prescaler clock: fperiph/4 (fperiph = fsys)

  The baud rate is determined as follows:

  $$\text{Baud Rate} = \frac{fc/16}{10} \div 16$$

  $$= 24.576 \times 10^6 \div 16 \div 10 \div 16 = 9600 \text{ (bps)}$$

  Note: Clearing the BR0CR.BR0ADDE bit to 0 disables the N + (16 – K) / 16 clock division function. At this time, the BR0ADD.BR0K[3:0] field is ignored.

- N + (16 – K) / 16 Clock Division (UART mode only)
  fperiph = 19.2-MHz fc
  ϕT0 = fperiph/4
  Baud rate generator input clock: ϕT2
  N (BR0CR.BR0S[3:0]) = 7
  K (BR0ADD.BR0K[3:0]) = 3
  BR0CR.BR0ADDE = 1

  Clocking conditions

  System clock: High-speed (fc)
  High-speed clock gear: ×1 (fc)
  Prescaler clock: fperiph/4 (fperiph = fsys)

  The baud rate is determined as follows:

  $$\text{Baud Rate} = \frac{fc/16}{7 + \frac{(16-3)}{16}} \div 16$$

  $$= 19.2 \times 10^6 \div 16 \div (7 + \frac{13}{16}) \div 16 = 9600 \text{ (bps)}$$

Table 13.3 and Table 13.4 show the UART baud rates obtained with various combinations of clock inputs and clock divisor values.

(4) Using an External Clock as a Serial Clock

The SIO0 and SIO1 can use an external clock as a serial clock, bypassing the baud rate generator. When an external clock is used, the baud rate is determined as shown below.

- UART Mode

  Baud Rate = external clock input ÷ 16

  The external clock period must be greater than or equal to 4/fsys. Therefore, when fsys = 32 MHz, the maximum baud rate is 500 kbps (32 ÷ 4 ÷ 16).

- I/O Interface Mode

  Baud Rate = external clock input clock

  The external clock period must be greater than 16/fsys. Therefore, when fsys = 32 MHz, the maximum baud rate is 2 Mbps (32 ÷16). For the timing parameters, refer to Section 18.6, *Serial Channel Timing*.

Table 13.3 UART Baud Rate Selection

When the baud rate generator is used and BR0CR.BR0ADDE = 0 Unit: kbps

fc (MHz)	Divisor N (Programmed in BR0CR.BR0S[3:0])	Baud Rate Generator Input Clock			
		ϕT0 (fc/4)	ϕT2 (fc/16)	ϕT8 (fc/64)	ϕT32 (fc/256)
19.6608	1	307.200	76.800	19.200	4.800
	2	153.600	38.400	9.600	2.400
	4	76.800	19.200	4.800	1.200
	8	38.400	9.600	2.400	0.600
	0	19.200	4.800	1.200	0.300
24.576	5	76.800	19.200	4.800	1.200
	A	38.400	9.600	2.400	0.600
29.4912	1	460.800	115.200	28.800	7.200
	2	230.400	57.600	14.400	3.600
	3	153.600	38.400	9.600	2.400
	4	115.200	28.800	7.200	1.800
	6	76.800	19.200	4.800	1.200
	C	38.400	9.600	2.400	0.600

Note: This table assumes: fsys = fc, clock gear = fc/1, prescaler clock source = fperiph/4

Table 13.4 UART Baud Rate Selection

When the TMRA0 timer trigger output is used and the TMRA0 input clock is ϕT1 Unit: kbps

TA0REG0	fc (MHz)					
	29.4912	24.576	24	19.6608	16	12.288
1H	230.4	192	187.5	153.6	125	96
2H	115.2	96	93.75	76.8	62.5	48
3H	76.8	64	62.5	51.2	41.67	32
4H	57.6	48	46.88	38.4	31.25	24
5H	46.08	38.4	37.5	30.72	25	19.2
6H	38.4	32	31.25	25.6	20.83	16
8H	28.8	24	23.44	19.2	15.63	12
AH	23.04	19.2	18.75	15.36	12.5	9.6
10H	14.4	12	11.72	9.6	7.81	6
14H	11.52	9.6	9.38	7.68	6.25	4.8

Note 1: I/O Interface mode can not utilize the trigger output signal from the 8-bit timer TMRA0 as a serial clock.

Note 2: This table assumes: fsys = fc, clock gear = fc/1, and prescaler clock source = fperiph/4

When the 8-bit timer TMRA0 is used to generate a serial clock, the baud rate is determined by the following equation:

$$\text{Baud Rate} = \frac{\text{clock frequency selected by SYSCR0.PRCK[1:0]}}{\text{TA0REG} \times \underline{2} \times 16}$$

↑ When theTMRA0 clock source is ϕT1.

13.2.3 Serial Clock Generator

This block generates a basic clock (SIOCLK) that controls the transimit and receive circuit.

- I/O Interface Mode

  When the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the output clock from the baud rate generator is divided by two to generate the SIOCLK clock. When the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the external SCLK0 clock is used as the SIOCLK clock; the SC0CR.SCLKS bit determines the active clock edge.

- UART Mode

  The SIOCLK clock is selected from a clock produced by the baud rate generator, the system clock (fsys/2), the trigger output signal from the 8-bit timer TMRA0, and the external SCLK0 clock, according to the setting of the SC0MOD0.SC[1:0] field.

13.2.4 Receive Counter

The receive counter is a 4-bit binary up-counter used in UART mode. This counter is clocked by SIOCLK. The receiver utilizes 16 clocks for each received bit, and oversamples each bit three times around their center (with 7th to 9th clocks). The value of a bit is determined by voting logic which takes the value of the majority of three samples. For example, if the three samples of a bit are 1, 0 and 1, then that bit is interpreted as a 1; if the three samples of a bit are 0, 0 and 1, then that bit is interpreted as a 0.

13.2.5 Receive Controller

- I/O Interface Mode

  If the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the receive controller samples the RXD0 input at the rising edge of the shift clock driven out from the SCLK0 pin. If the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the receive controller samples the RXD0 input at either the rising or falling edge of the SCLK0 clock, as programmed in the SC0CR.SCLKS bit.

- UART Mode

  The receive controller contains the start bit detection logic. Once a valid start bit is detected, the receive controller begins sampling the incoming data streams. The start bit, each data bit and the stop bit are sampled three times for 2-of-3 majority voting.

13.2.6 Receive Buffer

The receive buffer is double-buffered to prevent overrun errors. Received data is serially shifted bit by bit into Receive Buffer 1. When a whole character (i.e., 7 or 8 bits, as programmed) is loaded into Receive Buffer 1, it is transferred to Receive Buffer 2 (SC0BUF), and a receive-done interrupt (INTRX0) is generated.

- I/O Interface Mode

  The double-buffer structure can be used in full-duplex mode, but not in half-duplex mode. For details, refer to Section 13.4.

- UART Mode

  The CPU reads a character from Receive Buffer 2 (SC0BUF). Receive Buffer 1 can accept a new character through the RXD0 pin before the CPU picks up the previous character in Receive Buffer 2. However, the CPU must read Receive Buffer 2 before Receive Buffer 1 is filled with a new character. Otherwise, an overrun error occurs, causing the character previouly in Receive Buffer 1 to be lost. Even in that case, the contents of Receive Buffer 2 and the SC0CR.RB8 bit are preserved.

The SC0CR.RB8 bit holds the parity bit for an 8-bit UART character and the most-significant bit (i.e., address/data flag) bit for a 9-bit UART character.

In 9-bit UART mode, the receiver wake-up feature allows the slave station in a multidrop system to wake up whenever an address character is received. Setting the SC0MOD0.WU bit enables the wake-up feature. When the SC0CR.RB8 bit has received an address/data flag bit set to 1, the receiver generates the INTRX0 interrupt.

13.2.7 Transmit Counter

The transmit counter is a 4-bit binary up-counter used in UART mode. Like the receive counter, the transmit counter is also clocked by SIOCLK. The transmitter generates a transimit clock (TXDCLK) pulse every 16 SIOCLK pulses.

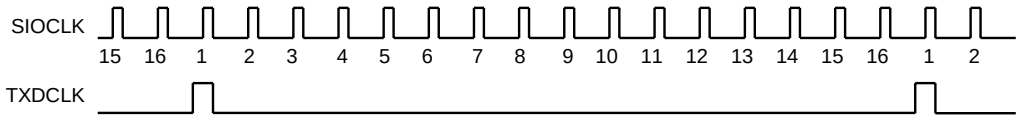


Figure 13.6 Transimit Clock Generation

13.2.8 Transmit Controller

- I/O Interface Mode

  If the SCLK0 pin is configured as an output by clearing the SC0CR.IOC bit to 0, the transimit controller shifts out each bit in the transmit buffer to the TXD0 pin at the rising edge of the shift clock driven out on the SCLK0 pin. If the SCLK0 pin is configured as an input by setting the SC0CR.IOC bit to 1, the transimit controller shifts out each bit in the transmit buffer to the TXD0 pin at either the rising or falling edge of the SCLK0 input, as programmed in the SC0CR.SCLKS bit.

- UART Mode

  Once the CPU loads a character into the transimit buffer, the transmit controller begins transmission at the next rising edge of TXDCLK, producing a transmit shift clock (TXDSFT).

Handshaking

The SIO0 and SIO1 have the clear-to-send ($\overline{CTS}$) pin. If the $\overline{CTS}$ operation is enabled, the $\overline{CTS}$ input must be low in order for the character to be transmitted. This feature can be used for flow control to prevent overrun in the receiver. The SC0MOD.CTSE bit enables and disables the $\overline{CTS}$ operation.

If the $\overline{CTS}$ pin goes high in the middle of a transmission, the transimit controller stops transmission upon completion of the current character until $\overline{CTS}$ again goes low. If so enabled, the transmit controller generates the INTTX0 interrupt to notify the CPU that the transmit buffer is empty. After the CPU loads the next character into the transmit buffer, the transmit controller remains in idle state until it detects $\overline{CTS}$ going low.

Although the SIO0 and SIO1 do not have the $\overline{RTS}$ pin, any general-purpose port pins can serve as the $\overline{RTS}$ pin. The receiving device uses the $\overline{RTS}$ output to control the $\overline{CTS}$ input of the transmitting device. Once the receiving device has received a character, $\overline{RTS}$ should be set to high in the receive-done interrupt handler to temporarily stop the transmitting device from sending the next character. This way, the user can easily implement a two-way handshake protocol.

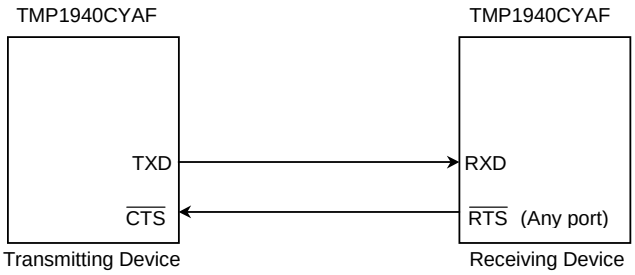
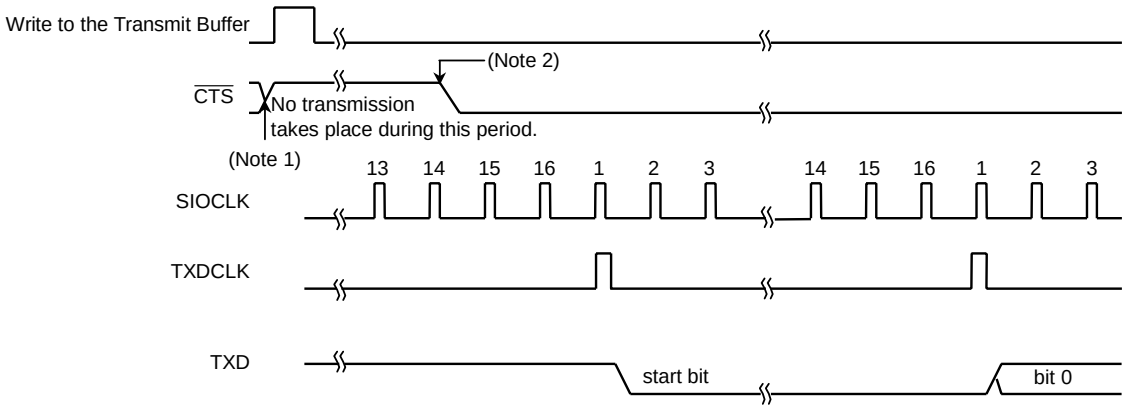


Figure 13.7 Handshaking Signals



Note 1: **When $\overline{CTS}$ goes high in the middle of transmission, the transmitter stops transmission after the current character has been sent.**

Note 2: **The transmitter starts tansmission at the first falling edge of the TXDCLK clock after the $\overline{CTS}$ signal goes low.**

Figure 13.8 Clear-To-Send ($\overline{CTS}$) Signal Timing

13.2.9 Transmit Buffer

Once the CPU loads a character into the transmit buffer (SC0BUF), it is shifted out on the TXD0 output, with the least-significant bit first, clocked by the transmit shift clock from the transmit controller. When the transmit buffer is empty and ready to be loaded with the next character, the INTTX0 interrupt is generated to the CPU. A character can not be written to the transmit buffer in the middle of a transmission.

13.2.10 Parity Controller

For transmit operations, setting the SC0CR.PE enables parity generation in 7- and 8-bit UART modes. The SC0CR.EVEN bit selects either even or odd parity.

If enabled, the parity controller automatically generates parity for the character in the transmit buffer (SC0BUF). In 7-bit UART mode, the TB7 bit in the SC0BUF holds the parity bit. In 8-bit UART mode, the TB8 bit in the SC0MOD holds the parity bit. The parity bit is set after the character has been transmitted. The SC0CR.PE and SC0CR.EVEN bits must be programmed prior to a write to the transmit buffer.

For receive operations, the parity controller automatically computes the expected parity when a character in Receive Buffer 1 is transferred to Receive Buffer 2 (SC0BUF). The received parity bit is compared to the SC0BUF.RB7 bit in 7-bit UART mode and to the SC0CR.RB8 bit in 8-bit UART mode. If a character is received with incorrect parity, the SC0CR.PERR bit is set.

13.2.11 Error Flags (UART mode only)

The SC0CR has the following error flag bits that indicate the status of the received character for improved data reception reliability.

- Overrun error (OERR)

  An overrun error is reported if all bits of a new character are received into Receive Buffer 1 when Receive Buffer 2 (SC0BUF) still contains a valid character.

- Parity error (PERR)

  A parity error is reported when the parity bit attached to a character received on the RXD pin does not match the expected parity computed from the character transferred to Receive Buffer 2 (SC0BUF).

- Framing error (FERR)

  A framing error is reported when a 0 is detected where a stop bit was expected. (The middle three of the 16 samples are used to determine the bit value.)

Note 1:	**Even if an error is present in a received character, the receive operation for the next character continues normally.**
Note 2:	**Error flags are kept until read.**

13.2.12 Signal Generation Timing

(1) UART Mode

Receive Operation

	9 Data Bits	8 Data Bits with Parity	8 Data Bits with No Parity 7 Data Bits with Parity 7 Data Bits with No Parity
Interrupt	Middle of the stop bit	Middle of the stop bit	Middle of the stop bit
Framing Error	Middle of the stop bit	Middle of the stop bit	Middle of the stop bit
Parity Error	—	Middle of the last bit (i.e., parity bit)	Middle of the last bit (i.e., parity bit)
Overrun Error	Middle of the last bit (i.e., bit 8)	Middle of the last bit (i.e., parity bit)	Middle of the stop bit

Transmit Operation

	9 Data Bits	8 Data Bits with Parity	8 Data Bits with No Parity 7 Data Bits with Parity 7 Data Bits with No Parity
Interrupt	Immediately before the stop bit is shifted out	Immediately before the stop bit is shifted out	Immediately before the stop bit is shifted out

(2) I/O Interface Mode

Transmit Interrupt	SCLK Output Mode	Immediately after the rising edge of the last SCLK pulse (See Figure 13.29)
	SCLK Input Mode	Immediately after the rising or falling edge of the last SCLK pulse, as programmed (See Figure 13.30)
Receive Interrupt	SCLK Output Mode	When a received character has been transferred to Receive Buffer 2 (SC0BUF) (i.e., immediately after the last SCLK pulse) (See Figure 13.31)
	SCLK Input Mode	When a received character has been transferred to Receive Buffer 2 (SC0BUF) (i.e., immediately after the last SCLK pulse) (See Figure 13.32)

Note 1: Don't modify any control register during transmit or receive operations.

Note 2: Don't disable receive operations by clearing the SC0MOD0.RXE bit while any character is being received.

13.3 Register Description

SC0MOD0 (0xFFFF_F202)

	7	6	5	4	3	2	1	0
Name	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Bit 8 of a transmitted character	Handshake control 0: Disables CTS operation 1: Enables CTS operation	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: I/O Interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: External clock (SCLK0 input)	

Wake-up function

	9-Bit UART Mode	Other Modes
0	Interrupt on every received character	Don't care
1	Interrupt only when RB8 = 1	

Handshake ($\overline{CTS}$) control

0	Disable (Accepts data streams at all times)
1	Enable

Note: In I/O Interface mode, a serial clock is selected by the SIO0 Control Register (SC0CR).

Figure 13.9 SIO0 Mode Register 0 (SC0MOD0)

SC1MOD0 (0xFFFF_F20A)

	7	6	5	4	3	2	1	0
Name	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Bit 8 of a transmitted character	Handshake control 0: Disables CTS operation 1: Enables CTS operation	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: I/O Interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: External clock (SCLK1 input)	

Wake-up function

	9-Bit UART Mode	Other Modes
0	Interrupt on every received character	Don't care
1	Interrupt only when RB8 = 1	

Handshake ($\overline{CTS}$) control

0	Disable (Accepts data streams at all times)
1	Enable

Note: In I/O Interface mode, a serial clock is selected by the SIO1 Control Register (SC1CR).

Figure 13.10 SIO1 Mode Register 0 (SC1MOD0)

SC3MOD0 (0xFFFF_F282)

	7	6	5	4	3	2	1	0
Name	TB8	—	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Bit 8 of a transmitted character	Must be written as 0.	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: Reserved 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: Don't care	

Wake-up function

	9-Bit UART Mode	Other Modes
0	Interrupt on every received character	Don't care
1	Interrupt only when RB8 = 1	

Figure 13.11 SIO3 Mode Register 0 (SC3MOD0)

SC4MOD0 (0xFFFF_F28A)

	7	6	5	4	3	2	1	0
Name	TB8	—	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Bit 8 of a transmitted character	Must be written as 0.	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: Reserved 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: Don't care	

Wake-up function

	9-Bit UART Mode	Other Modes
0	Interrupt on every received character	Don't care
1	Interrupt only when RB8 = 1	

Figure 13.12 SIO4 Mode Register 0 (SC4MOD0)

SC0CR (0xFFFF_F201)

	7	6	5	4	3	2	1	0
Name	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared when read)			R/W	
Reset Value		0	0	0	0	0	0	0
Function	Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	0: SCLK0 1: SCLK0	0: Baud rate generator 1: SCLK0 input

Input clock in I/O Interface mode

0	Baud rate generator
1	SCLK0 input

Active edge for the SCLK0 input

0	Data is transmitted/received on the SCLK0 rising edge.
1	Data is transmitted/received on the SCLK0 falling edge.

Framing error flag
Parity error flag
Overrun error flag
} These bits are cleared to 0 when read.

Input clock in I/O Interface mode

0	Odd parity
1	Even parity

Note 1: All error flags are cleared to 0 when read.

Note 2: When SCLK0 is configured as an output, the SCLKS bit must be cleared (rising-edge triggered).

Figure 13.13 SIO0 Control Register (SC0CR)

SC1CR (0xFFFF_F209)

	7	6	5	4	3	2	1	0
Name	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared when read)			R/W	
Reset Value		0	0	0	0	0	0	0
Function	Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	0: SCLK1 ↑ 1: SCLK1 ↓	0: Baud rate generator 1: SCLK1 input

Input clock in I/O Interface mode

0	Baud rate generator
1	SCLK1 input

Active edge for the SCLK1 input

0	Data is transmitted/received on the SCLK1 rising edge.
1	Data is transmitted/received on the SCLK1 falling edge.

Framing error flag
Parity error flag
Overrun error flag
} These bits are cleared to 0 when read.

Input clock in I/O Interface mode

0	Odd parity
1	Even parity

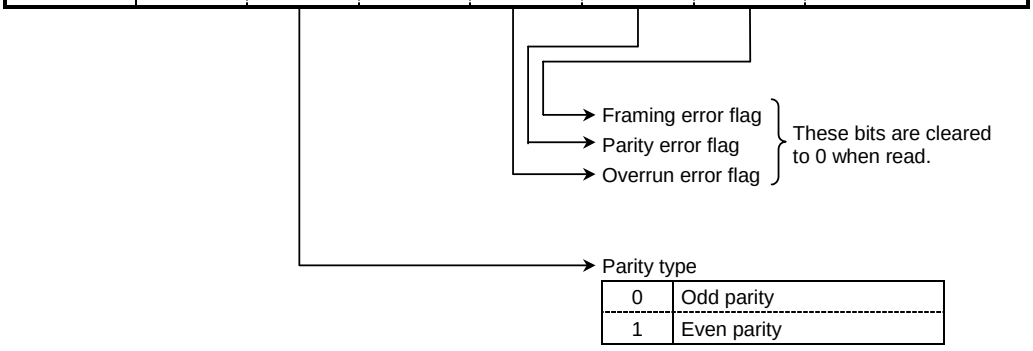
Note 1: All error flags are cleared to 0 when read.

Note 2: When SCLK1 is configured as an output, the SCLKS bit must be cleared (rising-edge triggered).

Figure 13.14 SIO1 Control Register (SC1CR)

SC3CR (0xFFFF_F281)

	7	6	5	4	3	2	1	0
Name	RB8	EVEN	PE	OERR	PERR	FERR	—	—
Read/Write	R	R/W		R (Cleared when read)			R/W	
Reset Value		0	0	0	0	0	0	0
Function	Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	Must be written as 00.	

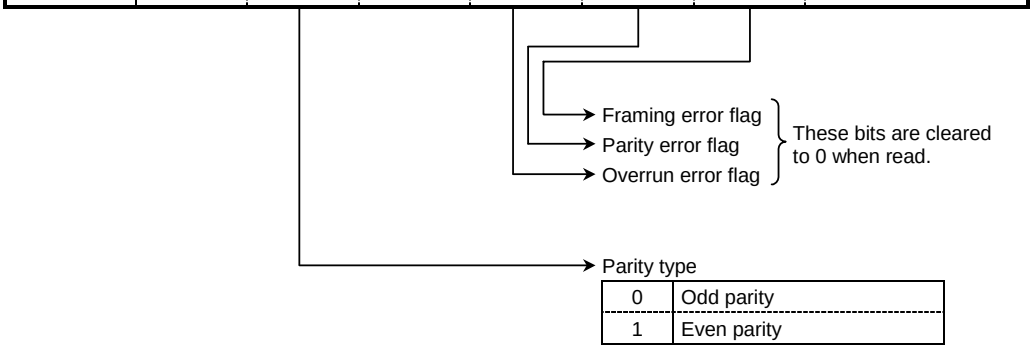


Note: All error flags are cleared to 0 when read.

Figure 13.15 SIO3 Control Register (SC3CR)

SC4CR (0xFFFF_F289)

	7	6	5	4	3	2	1	0
Name	RB8	EVEN	PE	OERR	PERR	FERR	—	—
Read/Write	R	R/W		R (Cleared when read)			R/W	
Reset Value		0	0	0	0	0	0	0
Function	Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	Must be written as 00.	



Note: All error flags are cleared to 0 when read.

Figure 13.16 SIO4 Control Register (SC4CR)

BR0CR (0xFFFF_F203)

	7	6	5	4	3	2	1	0
Name	—	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: φT0 01: φT2 10: φT8 11: φT32		Clock divisor value N			

Clock source for baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BR0ADD (0xFFFF_F204)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	0	0	0	0
Function					Value of K in N+(16–K)/16			

Clock divisor value for baud rate generator

BR0ADD. BR0K[3:0]	BR0CR.BR0ADDE = 1		BR0CR.BR0ADDE = 0
	BR0CR. BR0S[3:0]		
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) thru 1111 (N = 15)	0001 (N = 1) (Only UART) thru 1111 (N = 15) 0000 (N = 16)
0000	Don't use.	Don't use.	Divided by N
0001(K = 1) thru 1111(K = 15)	Don't use.	Divided by N + (16 – K) / 16	

Note 1: **The baud rate generator divisor can not be set to 1 in UART mode if the N + (16 – K) / 16 clock division function is enabled. The divisor should be set to 2 or greater in I/O Interface mode.**

Note 2: **To use the N + (16 – K) / 16 clock division function, the value of K must be programmed in the BR0ADD.BR0K[3:0] field before setting BR0CR.BR0ADDE to 1. However, the N + (16 – K) / 16 clock division function is not usable when BR0CR.BR0S[3:0] = 0000 (N = 16) or 0001 (N = 1).**

Note 3: **The N + (16 – K) / 16 clock division function can only be used in UART mode. In I/O Interface mode, this must be disabled by clearing BR0CR.BR0ADDE to 0.**

Figure 13.17 SIO0 Baud Rate Generator Control Registers (BR0CR and BR0ADD)

BR1CR (0xFFFF_F20B)

	7	6	5	4	3	2	1	0
Name	—	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: φT0 01: φT2 10: φT8 11: φT32		Clock divisor value N			

Clock source for baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BR1ADD (0xFFFF_F20C)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	BR1K3	BR1K2	BR1K1	BR1K0
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	0	0	0	0
Function					Value of K in N+(16–K)/16			

Clock divisor value for baud rate generator

BR1ADD. BR1K[3:0]	BR1CR.BR1ADDE = 1		BR1CR.BR1ADDE = 0
	BR1CR. BR1S[3:0]		
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) thru 1111 (N = 15)	0001 (N = 1) (Only UART) thru 1111 (N = 15) 0000 (N = 16)
0000	Invalid	Invalid	Divided by N
0001(K = 1) thru 1111(K = 15)	Invalid	Divided by N + (16 – K) / 16	

Note 1: **The baud rate generator divisor can not be set to 1 in UART mode if the N + (16 – K) / 16 clock division function is enabled. The divisor should be set to 2 or greater in I/O Interface mode.**

Note 2: **To use the N + (16 – K) / 16 clock division function, the value of K must be programmed in the BR0ADD.BR0K[3:0] field before setting BR0CR.BR0ADDE to 1. However, the N + (16 – K) / 16 clock division function is not usable when BR0CR.BR0S[3:0] = 0000 (N = 16) or 0001 (N = 1).**

Note 3: **The N + (16 – K) / 16 clock division function can only be used in UART mode. In I/O Interface mode, this must be disabled by clearing BR0CR.BR0ADDE to 0.**

Figure 13.18 SIO1 Baud Rate Generator Control Registers (BR1CR and BR1ADD)

BR3CR (0xFFFF_F283)

	7	6	5	4	3	2	1	0
Name	—	BR3ADDE	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: φT0 01: φT2 10: φT8 11: φT32		Clock divisor value N			

Clock source for baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BR3ADD (0xFFFF_F284)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	BR3K3	BR3K2	BR3K1	BR3K0
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	0	0	0	0
Function					Value of K in N+(16–K)/16			

Clock divisor value for baud rate generator

BR3ADD. BR3K[3:0]	BR3CR.BR3ADDE = 1		BR3CR.BR3ADDE = 0
	BR3CR. BR3S[3:0]		
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) thru 1111 (N = 15)	0001 (N = 1) (Only UART) thru 1111 (N = 15) 0000 (N = 16)
0000	Invalid	Invalid	Divided by N
0001(K = 1) thru 1111(K = 15)	Invalid	Divided by N + (16 – K) / 16	

Note 1: **The baud rate generator divisor can not be set to 1 in UART mode if the N + (16 – K) / 16 clock division function is enabled. The divisor should be set to 2 or greater in I/O Interface mode.**

Note 2: **To use the N + (16 – K) / 16 clock division function, the value of K must be programmed in the BR0ADD.BR0K[3:0] field before setting BR0CR.BR0ADDE to 1. However, the N + (16 – K) / 16 clock division function is not usable when BR0CR.BR0S[3:0] = 0000 (N = 16) or 0001 (N = 1).**

Note 3: **The N + (16 – K) / 16 clock division function can only be used in UART mode. In I/O Interface mode, this must be disabled by clearing BR0CR.BR0ADDE to 0.**

Figure 13.19 SIO3 Baud Rate Generator Control Registers (BR3CR and BR3ADD)

BR4CR (0xFFFF_F28B)

	7	6	5	4	3	2	1	0
Name	—	BR4ADDE	BR4CK1	BR4CK0	BR4S3	BR4S2	BR4S1	BR4S0
Read/Write	R/W							
Reset Value	0	0	0	0	0	0	0	0
Function	Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			

Clock source for baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR4ADD (0xFFFF_F28C)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	BR4K3	BR4K2	BR4K1	BR4K0
Read/Write	—	—	—	—	R/W			
Reset Value	—	—	—	—	0	0	0	0
Function					Value of K in N+(16–K)/16			

Clock divisor value for baud rate generator

BR4ADD. BR4K[3:0]	BR4CR.BR4ADDE = 1		BR4CR.BR4ADDE = 0
	BR4CR. BR4S[3:0]		
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) thru 1111 (N = 15)	0001 (N = 1) (Only UART) thru 1111 (N = 15) 0000 (N = 16)
0000	Invalid	Invalid	Divided by N
0001(K = 1) thru 1111(K = 15)	Invalid	Divided by N + (16 – K) / 16	

Note 1: **The baud rate generator divisor can not be set to 1 in UART mode if the N + (16 – K) / 16 clock division function is enabled. The divisor should be set to 2 or greater in I/O Interface mode.**

Note 2: **To use the N + (16 – K) / 16 clock division function, the value of K must be programmed in the BR0ADD.BR0K[3:0] field before setting BR0CR.BR0ADDE to 1. However, the N + (16 – K) / 16 clock division function is not usable when BR0CR.BR0S[3:0] = 0000 (N = 16) or 0001 (N = 1).**

Note 3: **The N + (16 – K) / 16 clock division function can only be used in UART mode. In I/O Interface mode, this must be disabled by clearing BR0CR.BR0ADDE to 0.**

Figure 13.20 SIO4 Baud Rate Generator Control Registers (BR4CR and BR4ADD)

SC0BUF (0xFFFF_F200)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(For tranmit)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(For receive)

Figure 13.21 SIO0 Transmit/Receive Buffer Register (SC0BUF)

SC0MOD1 (0xFFFF_F205)

	7	6	5	4	3	2	1	0
Name	I2S0	FDPX0	—	—	—	—	—	—
Read/Write	R/W	R/W	—	—	—	—	—	—
Reset Value	0	0	—	—	—	—	—	—
Function	SIO operation in IDLE mode 0: Off 1: On	Synchro-nous 0: Half-duplex 1: Full-duplex						

Figure 13.22 SIO0 Mode Register 1 (SC0MOD1)

SC1BUF (0xFFFF_F208)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(For tranmit)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(For receive)

Figure 13.23 SIO1 Transmit/Receive Buffer Register (SC1BUF)

SC1MOD1 (0xFFFF_F20D)

	7	6	5	4	3	2	1	0
Name	I2S0	FDPX0	—	—	—	—	—	—
Read/Write	R/W	R/W	—	—	—	—	—	—
Reset Value	0	0	—	—	—	—	—	—
Function	SIO operation in IDLE mode 0: Off 1: On	Synchro-nous 0: Half-duplex 1: Full-duplex						

Figure 13.24 SIO1 Mode Register 1 (SC1MOD1)

SC3BUF (0xFFFF_F280)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(For tranmit)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(For receive)

Figure 13.25 SIO3 Transmit/Receive Buffer Register (SC3BUF)

SC3MOD1 (0xFFFF_F285)

	7	6	5	4	3	2	1	0
Name	I2S0	—	—	—	—	—	—	—
Read/Write	R/W	—	—	—	—	—	—	—
Reset Value	0	—	—	—	—	—	—	—
Function	SIO operation in IDLE mode 0: Off 1: On							

Figure 13.26 SIO3 Mode Register 1 (SC3MOD1)

SC4BUF (0xFFFF_F288)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(For tranmit)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(For receive)

Figure 13.27 SIO4 Transmit/Receive Buffer Register (SC4BUF)

SC4MOD1 (0xFFFF_F28D)

	7	6	5	4	3	2	1	0
Name	I2S0	—	—	—	—	—	—	—
Read/Write	R/W	—	—	—	—	—	—	—
Reset Value	0	—	—	—	—	—	—	—
Function	SIO operation in IDLE mode 0: Off 1: On							

Figure 13.28 SIO4 Mode Register 1 (SC4MOD1)

13.4 Operating Modes

13.4.1 Mode 0 (I/O Interface Mode)

Mode 0 utilizes a synchronization clock (SCLK), which can be configured for either output mode in which the SCLK clock is driven out from the TMP1940CYAF or input mode in which the SCLK clock is supplied externally.

(1) Transmit Operations

In SCLK Output mode, each time the CPU writes a character to the transmit buffer, the eight bits of the character is shifted out on the TXD0 pin, and the synchronization clock is driven out from the SCLK0 pin. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated.

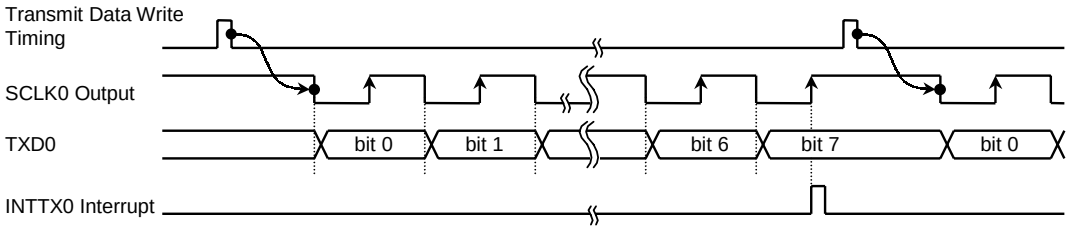


Figure 13.29 Transmit Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK0 Input mode, the CPU must write a character to the transmit buffer before the SCLK0 input is activated. The eight bits of a character in the transmit buffer are shifted out on the TXD0 pin, synchronous to the programmed edge of the SCLK0 input. When all the bits have been shifted out, the transmit-done interrupt (INTTX0) is generated. The CPU must load the next character into the transmit buffer by point A.

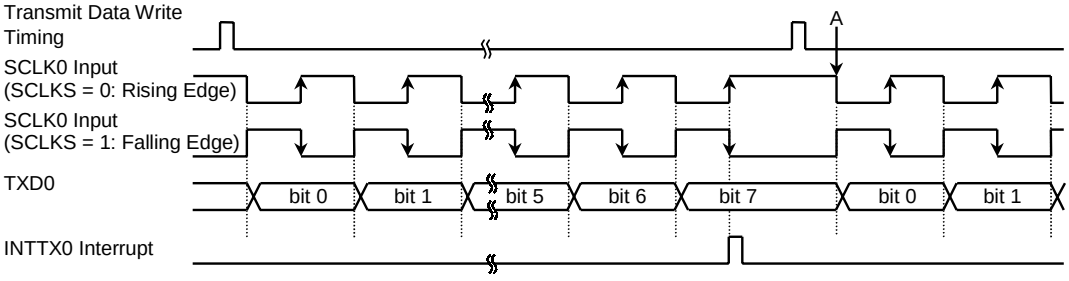


Figure 13.30 Transmit Operation in I/O Interface Mode (SCLK0 Input Mode)

(2) Receive Operations

In SCLK Output mode, each time the CPU picks up the character in Receive Buffer 2, the synchronization clock is driven out from the SCLK0 pin to shift the next character into Receive Buffer 1. When a whole 8-bit character has been loaded into Receive Buffer 1, it is transferred to Receive Buffer 2, and the receive-done interrupt (INTRX0) is generated.

The SCLK output is initiated by setting the SC0MOD0.RXE bit to 1.

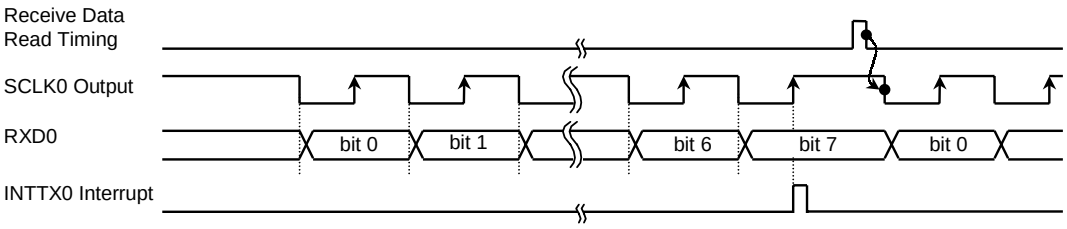


Figure 13.31 Receive Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK Input mode, the CPU must pick up the character in the Receive Buffer 2 before the SCLK0 input is activated to shift the next character into Receive Buffer 1. When a whole 8-bit character has been loaded into Receive Buffer 1, it is transferred to Receive Buffer 2, and the receive-done interrupt (INTRX0) is generated.

The CPU must read the character in Receive Buffer 2 by point A. Until that is done, the receiver is not ready to accept the next character. In case the CPU reads the character in Receiver Buffer 2 after point A, reception of the next character begins at that point, causing the received data to be corrupted. For system applications in which the CPU might not be able to keep pace with incoming data streams, handshaking is required.

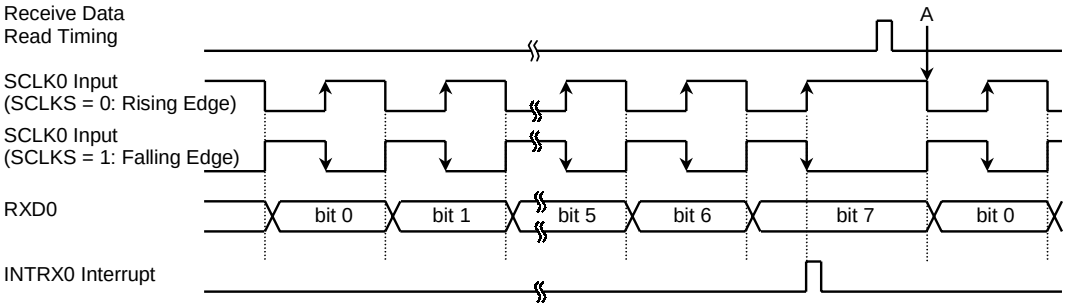


Figure 13.32 Receive Operation in I/O Interface Mode (SCLK0 Input Mode)

Note: Regardless of whether SCLK is in input mode or output mode, the receiver must be enabled by setting the SC0MOD.RXE bit to 1 in order to perform receive operations.

(3) Full-Duplex Transmit/Receive Operations

Setting the SC0MOD1.FDPX0 bit enables full-duplex communication. In this mode of operation, the double-buffering is enabled. When Receive Buffer 1 is filled with an 8-bit character, it is transferred to Receive Buffer 2 (SC0BUF), and the receive-done interrupt (INTRX0) is generated. While an 8-bit character is being received, an 8-bit character can be transmitted from the TXD0 pin simultaneously. When a whole 8-bit character has been shifted out, the transmit-done interrupt (INTTX0) is generated.

In SCLK Output mode, loading the transimit buffer with a character restarts the transmit/receive operation. The CPU must pick up the received character before the next character fills Receive Buffer 1. Otherwise, the latter character is discarded. (The previous character is preserved. Transmission proceeds with no error.)

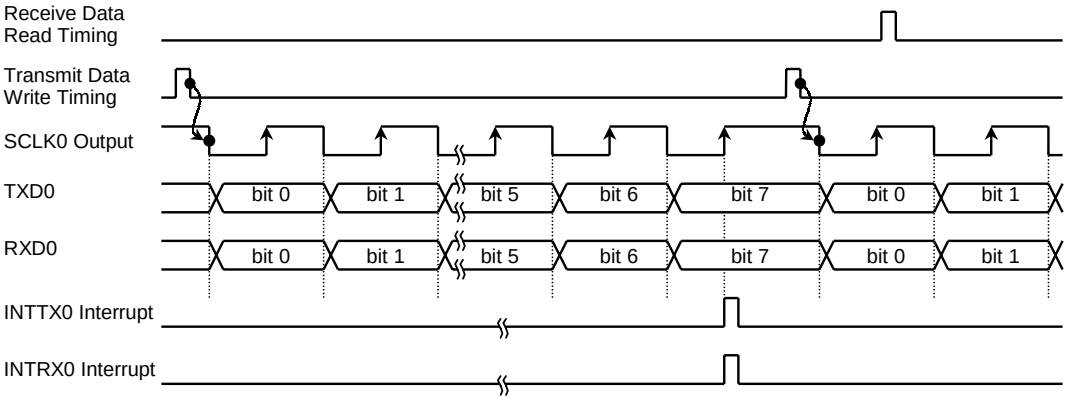


Figure 13.33 Full-Duplex Transmit/Receive Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK Input Mode, the CPU must write a character to be transmitted into the transmit buffer by point A. No transimi/receive operation occurs until the transmit buffer is filled. In case the transmit buffer is loaded after point A, the transmit/receive operation begins at that point, causing the transimit/receive data to be corrupted. For system applications in which transmit underrun conditions could occur, handshaking is required.

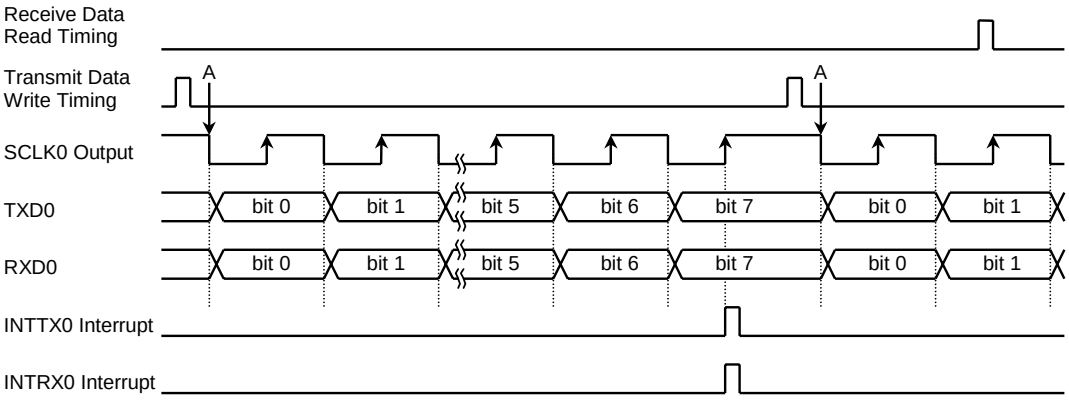
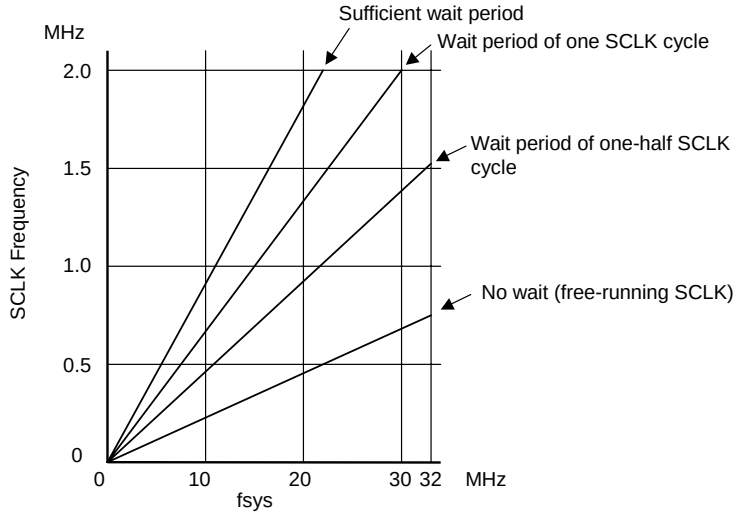


Figure 13.34 Full-Duplex Transmit/Receive Operation in I/O Interface Mode (SCLK0 Input Mode)

- Restrictions on SCLK Configured as an Input

In I/O Interface mode, the CPU may be unable to access the receive or transmit buffer fast enough to support back-to-back transfers. When SCLK is configured as an output, one or more wait cycles are automatically inserted to prolong the SCLK intervals. However, when SCLK is

configured as an input, the SCLK input must be delayed by external hardware so that the CPU can keep pace with the data rate. Generally, the wait period is a function of the fsys frequency and the data rate. The following figure gives some indication of the relationsip between SCLK and fsys frequencies for different wait periods. In reality, processing load during transfers also affect the maximum SCLK frequency.

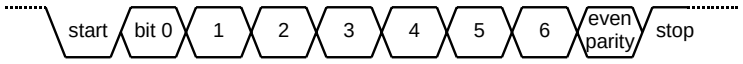


Note: The above figure assumes that the DMAC is utilized for reads of the receive buffer and writes of the transmit buffer.

13.4.2 Mode 1 (7-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 01 puts the SIO0 in 7-bit UART mode. In this mode of operation, the parity bit can be added to the transmitted character, and the receiver can perform a parity check on incoming data. Parity can be enabled and disabled through the programming of the PE bit in the SC0CR. When PE = 1, the SCR0CR.EVEN bit selects even or odd parity.

Example: Transmitting 7-bit UART characters with an even-parity bit



← Goes out first (transfer rate = 2400 bps @fc = 24.576 MHz)

Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	× 1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

Settings in the main routine

		7	6	5	4	3	2	1	0	
P9CR	←	–	–	–	–	–	–	–	1	Configures the P90 pin as TXD0.
P9FC	←	–	–	–	–	–	–	–	1	
SC0MOD	←	X	0	–	X	0	1	0	1	Selects 7-bit UART mode.
SC0CR	←	X	1	1	X	X	X	0	0	Selects even parity.
BR0CR	←	0	0	1	0	1	0	1	0	Sets the transfer rate to 2400 bps.
IMCCLH	←	–	–	1	1	0	1	0	0	Enables the INTTX0 interrupt and sets its priority level to 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Loads the transmit buffer with a character.

Transmit-done interrupt routine

INTCLR	←	X	X	1	1	0	0	0	1	Clears the interrupt request.

Interrupt processing

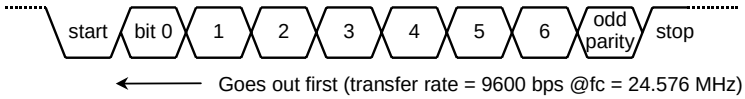
End of interrupt processing

X = Don't care, – = No change

13.4.3 Mode 2 (8-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 10 puts the SIO0 in 8-bit UART mode. In this mode of operation, the parity bit can be added to the transmitted character, and the receiver can perform a parity check on incoming data. Parity can be enabled and disabled through the programming of the PE bit in the SC0CR. When PE = 1, the SCR0CR.EVEN bit selects even or odd parity.

Example: Transmitting 8-bit UART characters with an odd-parity bit



Clocking conditions:

System clock:	High-speed (fc)
High-speed clock gear:	× 1 (fc)
Prescaler clock:	fperiph/4 (fperiph = fsys)

Settings in the main routine

		7	6	5	4	3	2	1	0	
P9CR	←	–	–	–	–	–	–	0	–	Configures P91 (RXD0) to be an input.
SC0MOD	←	–	0	1	X	1	0	0	1	Selects 8-bit UART mode and enables the receiver.
SC0CR	←	X	0	1	X	X	X	0	0	Selects odd parity.
BR0CR	←	0	0	0	1	0	1	0	1	Sets the transfer rate to 9600 bps.
IMCCLL	←	–	–	1	1	0	1	0	0	Enables the INTRX0 interrupt and sets its priority level to 4.

Example of interrupt routine processing

		7	6	5	4	3	2	1	0	
INTCLR	←	X	X	1	1	0	0	0	0	Clears the interrupt request.
Reg.	←	SC0CR AND 0x1C								Checks for errors.
if Reg. ≠ 0 then Error										
Reg.	←	SC0BUF								
End of interrupt processing										

X = Don't care, – = No change

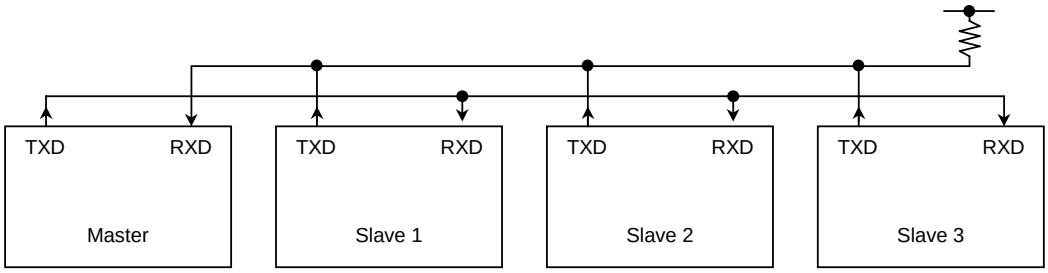
13.4.4 Mode 3 (9-Bit UART Mode)

Setting the SM[1:0] field in the SC0MOD0 to 11 puts the SIO0 in 9-bit UART mode. In this mode, a parity bit cannot be used; thus, parity should be disabled by clearing the SC0CR.PE bit to 0.

For transmit operations, the most-significant bit (9th bit) is stored in the TB8 bit in the SC0MOD0. For receive operations, the most-significant bit is stored in the RB8 bit in SC0CR. Reads and writes of the transmit/receive character must be done with the most-significant bit first, followed by the SC0BUF.

Wake-up Feature

In 9-bit UART mode, the receiver wake-up feature allows the slave station in a multidrop system to wake up whenever an address character is received. Setting the SC0MOD0.WU bit enables the wake-up feature. When the SC0CR.RB8 bit has received an address/data flag bit set to 1, the receiver generates the INTRX0 interrupt.



Note: The slave controller's TXD pin must be configured as an open-drain output by programming the ODE register.

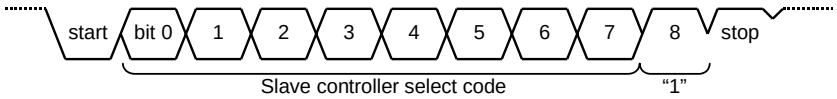
Figure 13.35 Serial Link Using the Wake-Up Function

Protocol

(1) Put all the master and slave controllers in 9-bit UART mode.

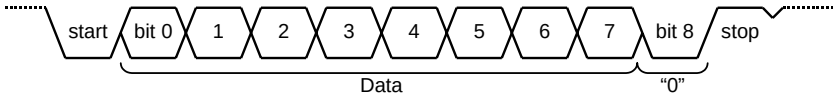
(2) Enables the receiver in each slave controller by setting the SC0MOD0.WU bit to 1.

(3) The master controller transmits an address character (i.e, select code) that identifies a slave controller. The address character has the most-significant bit (bit 8) set to 1.



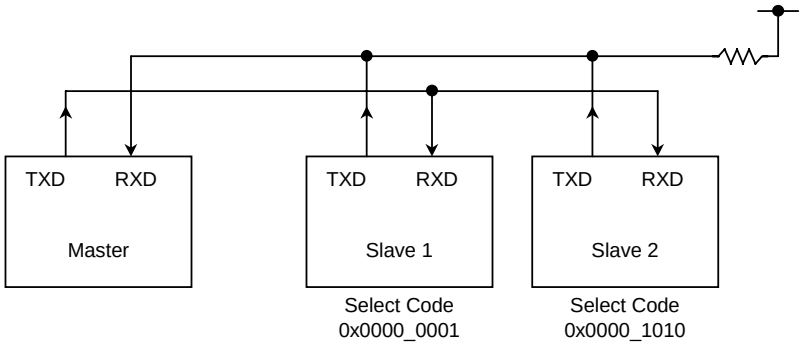
(4) Each slave controller compares the received address to its station address and clears the WU bit if they match.

(5) The master controller transmits data characters or block of data to the selected slave controller (with SC0MOD0.WU bit cleared). Data characters have the most-significant bit (bit 8) cleared to 0.



(6) Slave controllers not addressed continue to monitor the data stream, but discard any characters with the most-significant bit (RB8) cleared, and thus does not generate receive-done interrupts (INTRX0). The addressed slave controller with its WU bit cleared can transmit data to the master controller to notify that it has successfully received the message.

Example: Connecting a master station with two slave stations through a serial link using the fsys/2 clock as a serial clock



- Master controller settings

Main routine

```
                7  6  5  4  3  2  1  0

P9CR      ← –  –  –  –  –  –  0  1
P9FC      ← –  –  –  –  –  –  X  1
IMCCLL    ← –  –  1  1  0  1  0  1
IMCCLH    ← –  –  1  1  0  1  0  0
SC0MOD0   ← 1  0  1  0  1  1  1  0
SC0BUF    ← 0  0  0  0  0  0  0  1
```

Configures the P90 pin as TXD0 and the P91 pin as RXD0

Enables INTRX0 and sets its interrupt level to 5.

Enables INTTX0 and sets its interrupt level to 4.

Selects 9-bit UART mode and selects fsys/2 as a serial clock.

Loads the select code for slave 1.

Interrupt routine (INTTX0)

```
INTCLR    ← X  X  1  1  0  0  0  1
SC0MOD0   ← 0  –  –  –  –  –  –  –
SC0BUF    ← *  *  *  *  *  *  *  *
End of interrupt processing
```

Clears the interrupt request.

Clears the TB0 bit to 0.

Loads the transmit data.

- Slave controller settings

Main routine

```
                7  6  5  4  3  2  1  0

P9CR      ← –  –  –  –  –  –  0  1
P9FC      ← –  –  –  –  –  –  X  1
ODE       ← X  X  –  –  –  –  –  1
IMCCLL    ← –  –  1  1  0  1  1  0
IMCCLH    ← –  –  1  1  0  1  0  1
SC0MOD0   ← 0  0  1  1  1  1  1  0
```

Configures the P90 pin as TXD (open-drain output) and the P91 pin as RXD.

Enables INTTX0 and INTRX0.

Selects 9-bit UART mode, selects fsys/2 as the serial clock and and sets the WU bit to 1.

Interrupt routine (INTRX0)

```
INTCLR    ← X  X  1  1  0  0  0  0
Reg.      ← SC0BUF
if Reg. = Select code
Then
SC0MOD0   ← –  –  –  0  –  –  –  –
```

Clears the interrupt request.

Clears the WU bit to 0.

14. Serial Bus Interface (SBI)

The TMP1940CYAF contains a Serial Bus Interface (SBI) channel, which has the following two operating modes:

- I^2C Bus mode (with multi-master capability)
- Clock-Synchronous 8-Bit SIO mode

In I^2C Bus mode, the SBI is connected to external devices via two pins, PA6 (SDA) and PA7 (SCL). In Clock-Synchronous 8-Bit SIO mode, the SBI is connected to external devices via three pins, PA5 (SCK), PA6 (SO) and PA7 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	ODE.ODEA7 thru ODE.ODEA6	PACR.PA7C thru PACR.PA5C	PAFC.PA7F thru PAFC.PA5F
I^2C Bus Mode	11	11X	110
Clock-Synchronous 8-Bit SIO Mode	XX	011 010	111

X = Don't care

Note: With the TMP1940FDBF with flash memory, the SBI is unusable when the DSU feature is enabled.

14.1 Block Diagram

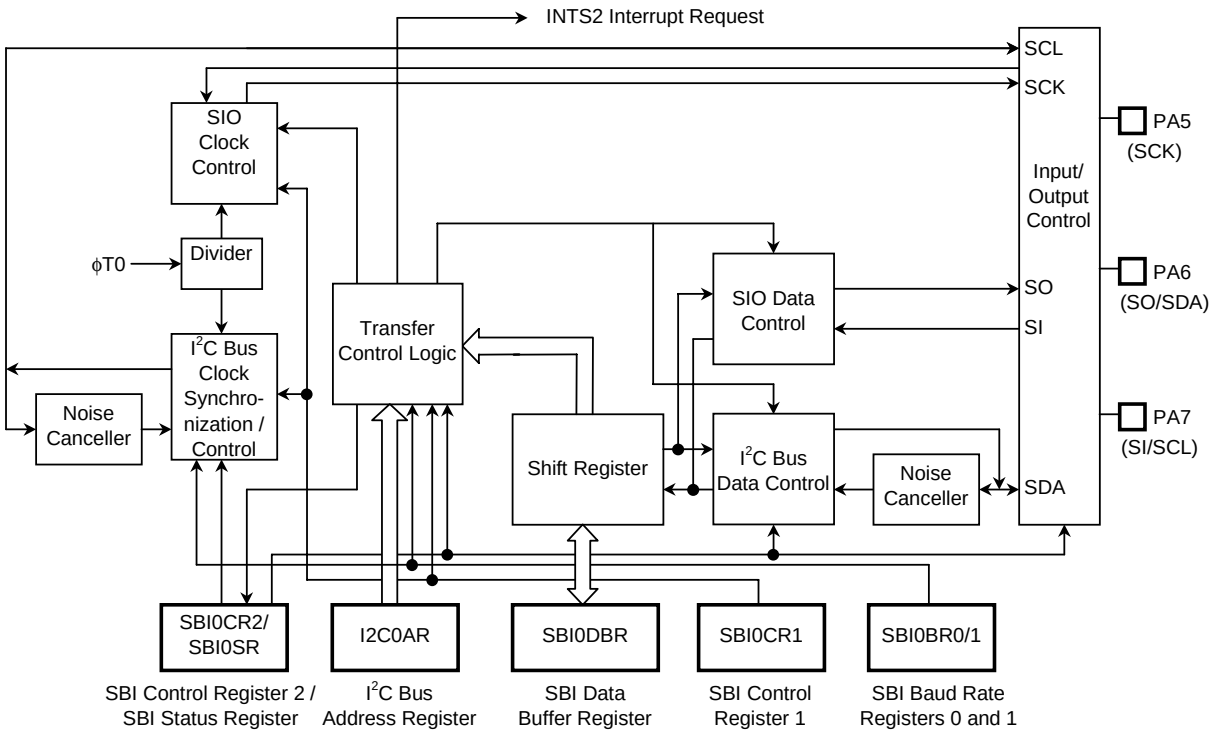


Figure 14.1 SBI Block Diagram

14.2 Registers

A listing of the registers used to control the SBI follows:

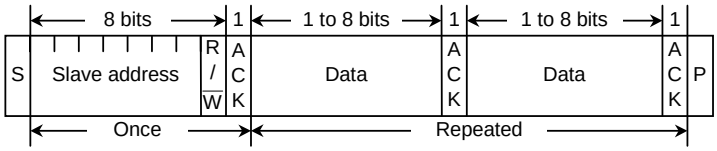
- Serial Bus Interface Control Register 1 (SBI0CR1)
- Serial Bus Interface Control Register 2 (SBI0CR2)
- Serial Bus Interface Data Buffer Register (SBI0DBR)
- I^2C Bus Address Register (I2C0AR)
- Serial Bus Interface Status Register (SBI0SR)
- Serial Bus Interface Baud Rate Register 0 (SBI0BR0)
- Serial Bus Interface Baud Rate Register 1 (SBI0BR1)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to Section 14.5, I^2C Bus Mode Configuration, and Section 14.8, Clock-Synchronous 8-Bit SIO Mode Operation.

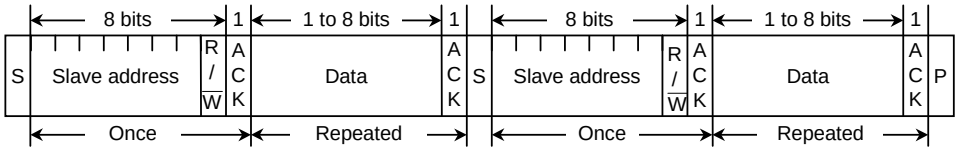
14.3 I^2C Bus Mode Data Formats

Figure 14.2 shows the serial bus interface data formats used in I^2C Bus mode.

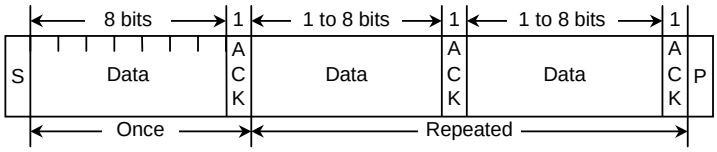
(a) Addressing format



(b) Addressing format (with repeated START condition)



(c) Free data format (master-transmitter to slave-receiver)



S = START condition
R/$\overline{W}$ = Direction bit
ACK = Acknowledge bit
P = STOP condition

Figure 14.2 I^2C-Bus Mode Data Formats

14.4 Description of the Registers Used in I^2C Bus Mode

This section provides a summary of the registers which control I^2C bus operation and provide I^2C bus status information for bus access/monitoring.

Serial Bus Interface Control Register 1

SBI0CR1 (0x FFFF_F240)

	7	6	5	4	3	2	1	0
Name	BC2	BC1	BC0	ACK	—	SCK2	SCK1	SCK0/ SWRMON
Read/Write	W			R/W	—	W		R/W
Reset Value	0	0	0	0	—	0	0	1
Function	Number of bits per transfer (Note 1)			ACK clock pulse 0: No ACK 1: ACK		Internal SCL output clock frequency (Note 2) / Software reset monitor		

On writes: SCK[2:0] = Internal SCL output clock frequency

000	n=4	400 kHz
001	n=5	222 kHz
010	n=6	118 kHz
011	n=7	60.6 kHz
100	n=8	30.8 kHz
101	n=9	15.5 kHz
110	n=10	7.78 kHz
111		Reserved

Assumptions:
System clock: fc (= 32 MHz)
Clock gear: fc/1
$\phi T0$ = fperiph/4 (= 8 MHz)

$$\text{Frequency} = \frac{\phi T0}{2^n + 4} \text{ (Hz)}$$

On reads: SWRMON = Software reset monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

Number of bits per transfer

BC [2:0]	ACK = 0		ACK = 1	
	# of clock cycles	Data length	# of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Note 1: Clear the BC[2:0] field to 000 before switching the operating mode to Clock-Synchronous 8-Bit SIO mode.

Note 2: For details on the SCL bus clock frequency, refer to Section 14.5.3, Serial Clock.

Figure 14.3 I^2C Bus Mode Registers (1)

Serial Bus Interface Control Register 2

SBI0CR2 (0xFFFF_F243)

	7	6	5	4	3	2	1	0
Name	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 1)		W (Note 1)	
Reset Value	0	0	0	1	0	0	0	0
Function	Master/ slave 0: Slave 1: Master	Transmit/ receive 0: Receive 1: Transmit	START / STOP generation 0: STOP condition 1: START condition	INTS2 interrupt clear 0: Don't care 1: Interrupt clear	Operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2C Bus mode 11: Reserved		Software reset A write of 10 followed by a write of 01	

Operating mode (Note 2)

00	Port mode (serial bus interface output disabled)
01	Clock-Synchronous 8-Bit SIO mode
10	I2C Bus mode
11	Reserved

Note 1: Reading this register causes it to function as a status register (SBI0SR). See the next page.

Note 2: Ensure that the bus is free before switching the operating mode to Port mode. Ensure that the port is at logic high before switching from Port mode to I2C Bus or SIO mode.

Figure 14.4 I2C Bus Mode Registers (2)

Table 14.1 Prescalar Output Clock (ϕT0) Resolutions

@fc = 32 MHz

Peripheral Clock Select SYSCR1.FPSEL	Clock Gear Value SYSCR1.GEAR[1:0]	Prescalar Clock Select SYSCR0.PRCK[1:0]	Prescalar Output Clock Resolution ϕT0
0 (fgear)	00 (fc)	00 (fperiph/4)	fc/2^2 (0.125 μs)
		01 (fperiph/2)	—
		10 (fperiph)	—
	01 (fc/2)	00 (fperiph/4)	fc/2^3 (0.25 μs)
		01 (fperiph/2)	—
		10 (fperiph)	—
	10 (fc/4)	00 (fperiph/4)	fc/2^4 (0.5 μs)
		01 (fperiph/2)	—
		10 (fperiph)	—
	11 (fc/8)	00 (fperiph/4)	fc/2^5 (1.0 μs)
		01 (fperiph/2)	—
		10 (fperiph)	—
1 (fc)	00 (fc)	00 (fperiph/4)	fc/2^2 (0.25 μs)
		01 (fperiph/2)	—
		10 (fperiph)	—
	01 (fc/2)	00 (fperiph/4)	—
		01 (fperiph/2)	—
		10 (fperiph)	—
	10 (fc/4)	00 (fperiph/4)	—
		01 (fperiph/2)	—
		10 (fperiph)	—
	11 (fc/8)	00 (fperiph/4)	—
		01 (fperiph/2)	—
		10 (fperiph)	—

Note: The — character means "Don't use."

Serial Bus Interface Status Register

SBI0SR (0xFFFF_F243)

	7	6	5	4	3	2	1	0
Name	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
Reset Value	0	0	0	1	0	0	0	0
Function	Master/ slave 0: Slave 1: Master	Transmit/ receive 0: Receive 1: Transmit	I^2C Bus status 0: Free 1: Busy	INTS2 interrupt status 0: Asserted 1: Not asserted	Arbitration lost 0: — 1: Detected	Addressed as slave 0: — 1: Detected	Address 0 (general call) 0: — 1: Detected	Last received bit 0: 0 1: 1

Last received bit

0	The last bit received was 0.
1	The last bit received was 1.

Addressed as slave

0	—
1	The address on the bus matches the I2COAR or general-call address (slave receiver mode only)

Arbitration lost

0	—
1	Arbitration was lost to another master.

Note: Writing to this register causes it to function as a control register (SBI0CR2). See the previous page.

Figure 14.5 I^2C Bus Mode Registers (3)

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (0xFFFF_F244)

	7	6	5	4	3	2	1	0
Name	—	I2SBI0	—	—	—	—	—	—
Read/Write	—	R/W	—	—	—	—	—	W
Reset Value	—	0	—	—	—	—	—	—
Function		IDLE 0: Off 1: On						Must be written as 0.

SBI on/off in IDLE mode

0	Off
1	On

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0xFFFF_F245)

	7	6	5	4	3	2	1	0
Name	P4EN	—	—	—	—	—	—	—
Read/Write	R/W	—	—	—	—	—	—	—
Reset Value	0	—	—	—	—	—	—	—
Function	Internal clock 0: Off 1: On							

Controls the iternal baud rate generator

0	Off
1	On

Serial Bus Interface Data Buffer Register

SBI0DBR (0xFFFF_F241)

	7	6	5	4	3	2	1	0
Name	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive) / W (transmit)							
Reset Value	Undefined							

Note: In transmitter mode, data must be written to this register, with bit 7 being the most-significant bit (MSB).

I^2C Bus Address Register

I2C0AR (0xFFFF_F242)

	7	6	5	4	3	2	1	0
Name	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
Reset Value	0	0	0	0	0	0	0	0
Function	When the SBI is addressed as a slave, this field specifies a 7-bit I^2C-bus address to which the SBI responds.							Address recognition mode

Address recognition mode

0	Recognizes the slave address.
1	Does not recognize the slave address.

Figure 14.6 I^2C Bus Mode Registers (4)

14.5 I²C Bus Mode Configuration

14.5.1 Acknowledgment Mode

Setting the SBI0CR1.ACK bit selects Acknowledge mode. When operating as a master, the SBI generates a clock pulse for acknowledge automatically after each data. As a transmitter, the SBI releases the SDA line during this acknowledge cycle so that the receiver of the data transfer can drive the SDA line low to acknowledge receipt of the data. As a receiver, the SBI pulls the SDA line low during the acknowledge cycle after each data has been received.

Clearing the SBI0CR1.ACK bit selects Non-Acknowledge mode. When operating as a master, the SBI does not generate acknowledge clock pulses.

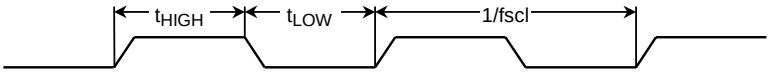
14.5.2 Number of Bits Per Transfer

The SBI0CR1.BC[2:0] field specifies the number of bits of the next data item to be transmitted or received. After a reset, this field is cleared to 000, causing a 7-bit slave address and the data direction ($R/\overline{W}$) bit to be transferred in a packet of eight bits. At other times, the SBI0CR1.BC[2:0] field keeps a previously programmed value.

14.5.3 Serial Clock

(1) I²C Bus Clock Source

The SBI0CR1.SCK[2:0] field controls the maximum frequency of the SCL clock driven out on the SCL pin in master mode, as illustrated below.



$t_{LOW} = 2^{n-1}/\phi T0$
$t_{HIGH} = 2^{n-1}/\phi T0 + 4/\phi T0$
$fscl = 1/(t_{Low} + t_{HIGH})$
$= \frac{\phi T0}{2^n+4}$

SBI0CR1.SCK[2:0]	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 14.7 I²C Bus Clock Source

(2) Clock Synchronization

Clock synchronization is performed using the wired-AND connection of all I^2C-bus components to the bus. If two or more masters try to transfer messages on the I^2C bus, the first to pull its clock line low wins the arbitration, overriding other masters producing a high on their clock lines.

Clock signals of two or more devices on the I^2C-bus are synchronized to ensure correct data transfers. Figure 14.8 shows a depiction of the clock synchronization mechanism for the I^2C bus with two masters.

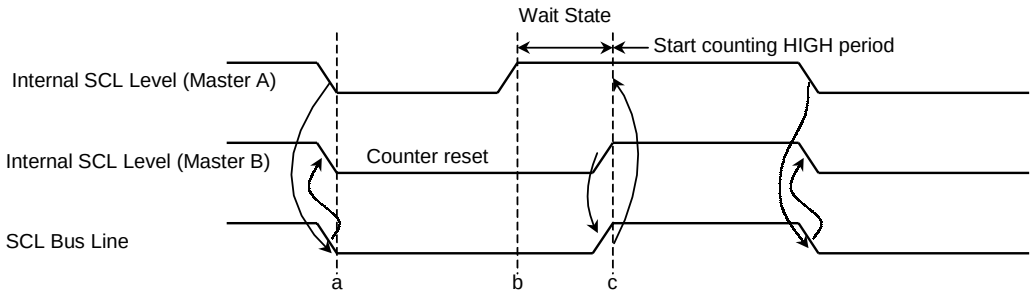


Figure 14.8 Clock Synchronization Example

At point a, Master A pulls its internal SCL level low, bringing the SCL bus line low. The high-to-low transition on the SCL bus line causes Master B to reset its high-level counter and pulls its internal SCL level low.

Master A completes its low period at point b. However, the low-to-high transition on its internal SCL level does not change the state of the SCL bus line if Master B's internal SCL level is still within its low period. Therefore, Master A enters a high wait state, where it does not start counting off its high period.

When Master B has counted off its low period at point c, its internal SCL level goes high, releasing the SCL bus line (high). There will then be no difference between the internal SCL levels and the state of the SCL bus line, and both Master A and Master B start counting off their high periods.

This way, a synchronized SCL clock is generated with its high period determined by the master with the shortest clock high period and its low period determined by the one with the longest clock low period.

14.5.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave, the SA[6:0] field in the I2C0AR must be loaded with the 7-bit I^2C-bus address to which the SBI is to respond. The ALS bit must be cleared for the SBI to recognize the incoming slave address.

14.5.5 Configuring the SBI as a Master or a Slave

Setting the SBI0CR2.MST bit configures the SBI as a master, and clearing it configures the SBI as a slave. This bit is cleared by hardware when a STOP condition has been detected and when arbitration for the I^2C bus has been lost.

14.5.6 Configuring the SBI as a Transmitter or a Receiver

The SBI0CR2.TRX bit is set or cleared by hardware to configure the SBI as a transmitter or a receiver.

As a slave, the SBI is put in either slave-receiver or slave-transmitter mode, depending on the value of the data direction ($R/\overline{W}$) bit transmitted by the master. When the SBI is addressed as a slave, the TRX bit reflects the value of the $R/\overline{W}$ bit. The TRX bit is set or cleared on the following occasions:

- when transferring data using addressing format
- when the received slave address matches the value in I2C0CR
- when a general-call address is received; i.e., the eight bits following the START condition are all zeros.

As a master, the SBI is put in either master-transmitter or a master-receiver mode upon reception of an acknowledge from an addressed slave. The TRX bit changes to the opposite value of the $R/\overline{W}$ bit sent by the SBI. If the SBI does not receive an acknowledge from a slave, the TRX bit retains the previous value.

The TRX bit is cleared by hardware when a STOP condition has been detected and when arbitration for the I^2C bus has been lost.

14.5.7 Generating START and STOP Conditions

When the SBI0SR.BB bit is cleared, the bus is free. At this time, writing 1s to the MST, TRX, BB and PIN bits in the SBI0CR2 causes the SBI to generate a START condition on the bus and shift out 8-bit I^2C-bus data. Before generating a START condition, the ACK bit must be set to 1.

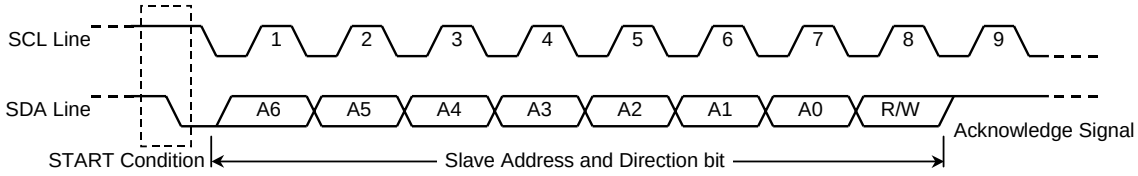


Figure 14.9 Generating a START Condition and a Slave Address

When the SBI0SR.BB bit is set, the bus is busy. When SBI0SR.BB=1, writing 1s to the MST, TRX and PIN bits and a 0 to the BB bit causes the SBI to start a sequence for generating a STOP condition on the bus to abort the transfer. The MST, TRX, BB and PIN bits should not be altered until a STOP condition appears on the bus.

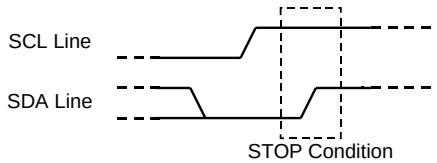


Figure 14.10 Generating a STOP Condition

The BB bit can be read to determine if the I^2C bus is in use. The BB bit is set when a START condition is detected and cleared when a STOP condition is detected.

14.5.8 Asserting and Deasserting Interrupt Requests

When an SBI interrupt (INTS2) is generated, the Pending Interrupt Not (PIN) bit in the SBI0CR2 is cleared to 0. While the PIN bit is 0, the SBI pulls the SCL line low.

After transmission or reception of one data word on the I^2C bus, the PIN bit is automatically cleared. In transmitter mode, the PIN bit is subsequently set to 1 each time the SBI0DBR is written. In receiver mode, the PIN bit is set to 1 each time the SBI0DBR is read.

It takes a period of t_{LOW} for the SCL line to be released after the PIN bit is set.

In Address Recognition mode (ALS=0), the PIN bit is cleared when the SBI is addressed as a slave and the received slave address matches the value in the I2C0CR or is all 0s (i.e., a general call).

A write of 1 by software sets the PIN bit, but a write of 0 has no effect on this bit.

14.5.9 SBI Operating Modes

The SBIM[1:0] field in the SBI0CR2 is used to select an operating mode of the SBI. To configure the SBI for I^2C Bus mode, set the SBIM[1:0] field to 10.

A switch to Port mode should only be attempted when the bus is free.

14.5.10 Lost-Arbitration Detection Monitor

The I^2C bus is a multi-master bus and has an arbitration procedure to ensure correct data transfers.

A master may start a transfer only if the bus is free. A master that attempts to generate a START condition while the bus is busy loses bus arbitration, with no START condition occurring on the SDA and SCL lines.

The I^2C-bus arbitration takes place on the SDA line.

Figure 14.11 shows the arbitration procedure for two masters. Up until point a, the internal data levels of Master A and Master B are the same. At point a Master B's internal data level makes a low-to-high transition while Master A's internal data level remains at logic low. However, the SDA bus line is held low because it is the wired-AND of the two data outputs. When the SCL bus clock goes high at point b, the addressed slave device reads the data transmitted by Master A (i.e., winning master). Master B loses arbitration and switches off its data output stage, releasing its SDA line (high), so that it does not affect the data transfer initiated by the winning master.

In case two competing masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

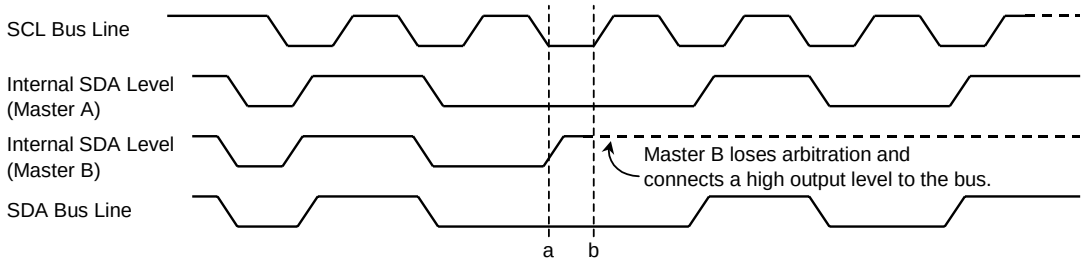


Figure 14.11 Arbitration Procedure of Two Masters

A master compares its internal data level to the actual level on the SDA line at the rising edge of the SCL clock. The master loses arbitration if there is a difference between these two values. The losing master sets the AL bit in the SBI0SR to 1, which causes the MST and TRX bits in the same register to be cleared. That is, the losing master switches to slave-receiver mode.

The AL bit is subsequently cleared when data is written to or read from the SBI0DBR and when the SBI0CR2 is programmed with new parameters.

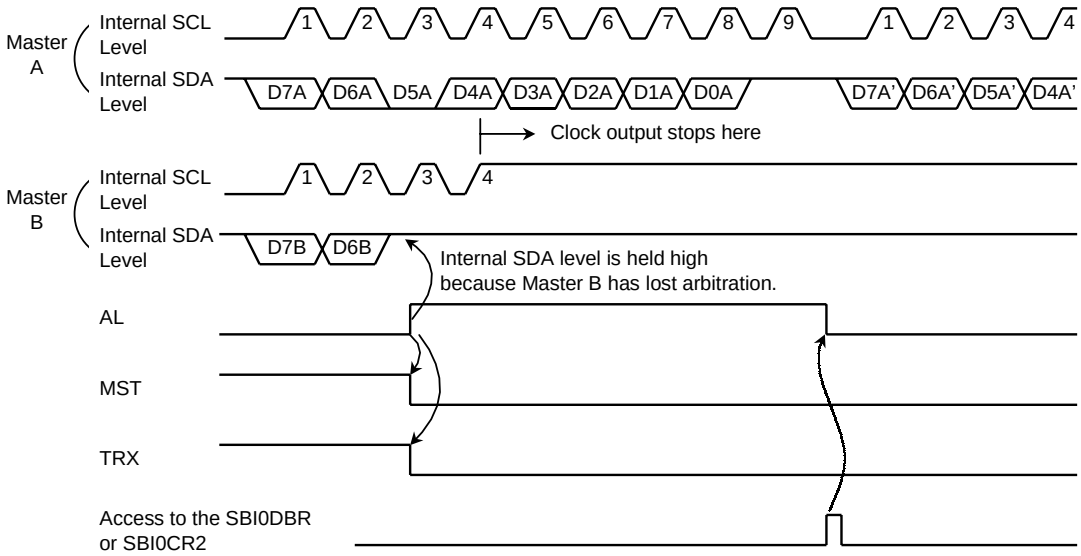


Figure 14.12 Master B Loses Arbitration (D7A – D7B, D6A – D6B)

14.5.11 Slave Address Match Monitor

When acting as a slave-receiver, the ALS bit in the I2C0CR determines whether the SBI recognizes the incoming slave address or not. In Address Recognition mode (i.e., ALS=0), the Addressed-As-Slave (AAS) bit in the SBI0SR is set when an incoming address over the I^2C bus matches the value in the I2C0CR or when the general-call address has been received. When ALS=1, the AAS bit is set when the first data word has been received. The AAS bit is cleared each time the SBI0DBR is read or written.

14.5.12 General-Call Detection Monitor

When acting as a slave receiver, the AD0 bit in the SBI0SR is set when a general-call address has been received. The general-call address is detected when the eight bits following a START condition are all zeros. The AD0 bit is cleared when a START or STOP condition is detected on the bus.

14.5.13 Last Received Bit Monitor

The LRB bit in the SBI0SR holds the value of the last bit received over the SDA line at the rising edge of the SCL clock. In Acknowledge mode, reading this bit immediately after generation of the INTS2 interrupt returns the value of the ACK signal.

14.5.14 Software Reset

The SBI provides a software reset, which permits recovery from system lockups caused by external noise. A software reset is performed by a write of 10 followed by a write of 01 to the SWRST[1:0] field in the SBI0CR2. After a software reset, all control and status register bits are initialized to their reset values. Upon resetting the SBI, the SWRST[1:0] field is automatically cleared to 00.

Note: A software reset causes the SBI operating mode to switch from I^2C Bus mode to Port mode. This does not affect the Port A Function register, however.

14.5.15 Serial Bus Interface Data Buffer Register (SBI0DBR)

The SBI0DBR is a data buffer interfacing to the I^2C bus. All read and write operations to/from the I^2C bus are done via this register.

When the SBI is acting as a master, loading this register with a slave address and a data direction bit causes a START condition to be generated.

14.5.16 I^2C Bus Address Register (I2C0AR)

When the SBI is configured as a slave, the SA[6:0] field in the I2C0AR must be loaded with the 7-bit I^2C-bus address to which the SBI is to respond.

If the ALS bit in the I2C0AR is cleared, the SBI recognizes a slave address transmitted by the master device, interpreting incoming frame structures as per addressing format. If the ALS bit is set, the SBI does not recognize a slave address and interprets all frame structures as per free data format.

14.5.17 Baud Rate Register 1 (SBI0DBR1)

Before the I^2C bus can be used, the P4EN bit in the SBI0BR1 must be set to enable the SBI internal baud rate generation logic.

14.5.18 Baud Rate Register 0 (SBI0BR0)

The I2SBI0 bit in the SBI0BR0 determines whether the SBI is shut down or not when the TMP1940CYAF is put in IDLE standby mode. This register must be programmed before executing an instruction for entering a standby mode.

14.6 Programming Sequences in I^2C Bus Mode

14.6.1 SBI Initialization

First, program the P4EN bit in the SBI0BR1, and the ACK and SCK[2:0] bits in the SBI0CR1. Set the SBI0BR1.P4EN bit to 1 to enable the internal baud rate generation logic. Write 0s to bits 7–5 and bit 3 in the SBI0CR1.

Next, program the I2C0AR. The SA[6:0] field in the I2C0AR defines the chip's slave address, and the ALS bit (bit 0) selects an address recognition mode. (The ALS bit must be cleared when using the addressing format.)

Next, program the SBI0CR2 to initially configure the SBI in slave-receiver mode; i.e., clear the MST, TRX and BB bits to 0, set the PIN bit to 1 and set the SBIM[1:0] field to 10. Write 00 to the SWRST[1:0] field.

```
                7 6 5 4 3 2 1 0
SBI0BR1  ← 1 0 0 0 0 0 0 0   Enable internal baud rate generator.
SBI0CR1  ← 0 0 0 X 0 X X X   Disable generation of ACK and select SCL clock frequency.
I2C0AR   ← X X X X X X X X   Load a slave address and selects address recognition mode.
SBI0CR2  ← 0 0 0 1 1 0 0 0   Configure the SBI in slave-receiver mode.
```

Note: X = Don't care

14.6.2 Generating a START Condition and a Slave Address

(1) Master Mode

In master mode, the following steps are required to generate a START condition and a slave address on the I^2C-bus.

First, ensure that the bus is free (i.e., SBI0CR2.BB = 0).

Next, set the ACK bit in the SBI0CR1 to enable generation of acknowledge clock pulses. Then, loads the SBI0DBR with a slave address and a data direction bit to be transmitted via the I^2C bus.

When BB=0, writing 1s to the MST, TRX, BB and PIN bits in the SBI0CR2 causes a START condition to be generated on the bus. Following a START condition, the SBI generates SCL clock pulses nine times: the SBI shifts out the contents of the SBI0DBR with the first eight SCL clocks, and releases the SDA line during the last (i.e., ninth) SCL clock to receive an acknowledgement signal from the addressed slave.

The INTS2 interrupt request is generated on the falling edge of the ninth SCL clock pulse, and the PIN bit in the SBI0CR2 is cleared to 0. In master mode, the SBI holds the SCL line low while the PIN bit is 0. Upon interrupt, the TRX bit either remains set or is cleared according to the value of the transmitted direction bit, provided an acknowledgement signal has been returned from the slave.

Settings in main routine

```
                7 6 5 4 3 2 1 0
Reg.        ← SBI0SR
Reg.        ← Reg. & 0x20
if Reg.     ≠ 0x00                Ensure that the bus is free.
Then
SBI0CR1  ← X X X 1 0 X X X   Select Acknowledgement mode.
SBI0DBR  ← X X X X X X X X   Load the slave address and a data direction bit.
SBI0CR2  ← 1 1 1 1 1 0 0 0   Generate a START condition.
```

INTS2 interrupt routine

```
INTCLR   ← 0x34               Clear the interrupt request.
Interrupt processing
End of interrupt
```

(2) Slave Mode

In slave mode, the following steps are required to receive a START condition and a slave address via the I^2C bus.

Upon detection of a START condition, the SBI clocks in a 7-bit slave address and a data direction bit transmitted by the master during the first eight SCL clock pulses. If the received slave address matches its own address in the I2C0AR or is equal to the general-call address (00H), the SBI pulls the SDA line low during the last (i.e., ninth) SCL clock for acknowledgement.

The INTS2 interrupt request is generated on the falling edge of the ninth SCL clock pulse, and the PIN bit in the SBI0CR2 is cleared to 0. In slave mode, the SBI holds the SCL line low while the PIN bit is 0.

Note: The user can only use a DMA transfer:
- when there is only one master and only one slave on the I^2C bus; and
- continuous transmission or reception is possible.

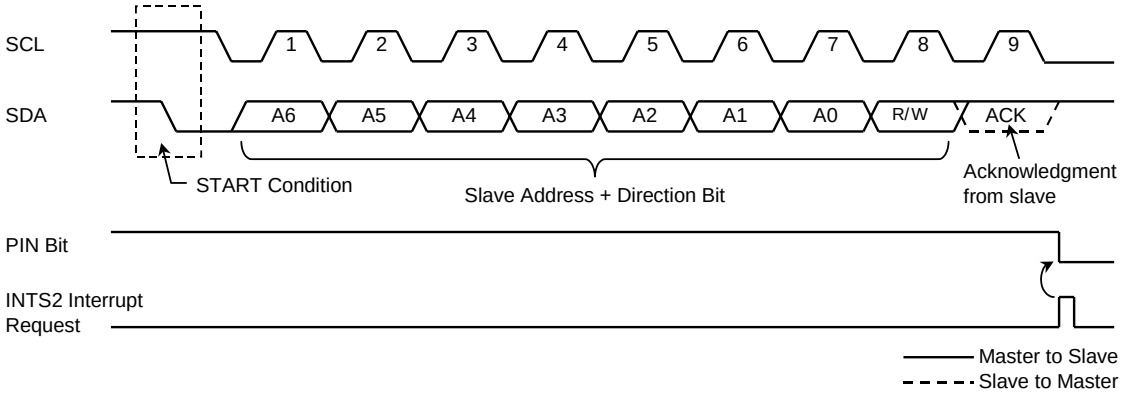


Figure 14.13 Generation of a START Condition and a Slave Address

14.6.3 Transferring a Data Word

Each time a data word has been transmitted or received, the INTS2 interrupt is generated. It is the responsibility of the INTS2 interrupt service routine to test the MST bit in the SBI0CR to determine whether the SBI is in master or slave mode.

(1) Master Mode (SBI0CR2.MST = 1)

If the MST bit in the SBI0CR2 is set, then test the TRX bit in the same register to determine whether the SBI is in master-transmitter or master-receiver mode.

Master-Transmitter Mode (SBI0CR2.TRX = 1)

Test the LRB bit in the SBI0SR. If the LRB bit is set, that means the slave-receiver requires no further data to be sent from the master-transmitter. The master-transmitter must then generate a STOP condition as described later to stop transmission.

If the LRB bit is cleared, that means the slave-receiver requires further data. If the number of bits per transfer is 8, then write the transmit data into the SBI0DBR. When using other data length, program the BC[2:0] and ACK bits in the SBI0CR1, and then write the transmit data into the SBI0DBR. When the SBI0DBR is loaded, the PIN bit in the SBI0SR is set to 1, and the transmit data is shifted out from the SDA pin, clocked by the SCL clock. Once the transfer is complete, the INTS2 interrupt is generated, the PIN bit is cleared, and the SCL line is pulled low. To transmit further data, test the LRB bit again and repeat the above procedure.

INTS2 interrupt

if MST = 0
Then go to slave-mode processing
if TRX = 0
Then go to receiver-mode processing
if LRB = 0
Then go to processing for generating a STOP condition

SBI0DBR ← X X X X X X X X	Set number of bits to be transmitted and specify whether ACK is required.
SBI0DBR	Load the transmit data.

End of interrupt processing

X = Don't care

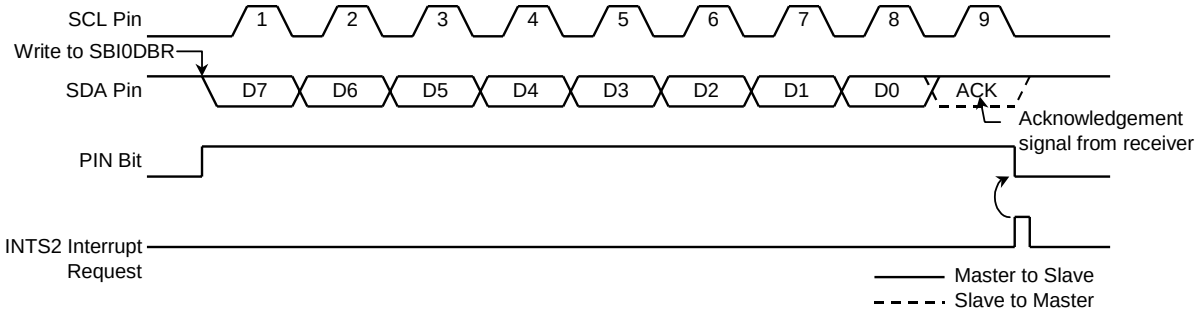


Figure 14.14 SBI0CR1.BC[2:0] = 000 and SBI0CR1.ACK = 1 (Master-Transmitter Mode)

Master-Receiver Mode (SBI0CR2.TRX = 0)

If the number of bits per transfer is 8, read the SBI0DBR. When using other data length, program the BC[2:0] and ACK bits in the SBI0CR1, and then read the SBI0DBR. The first read of the SBI0DBR is a dummy read because data has not yet been received. A dummy read returns an undefined value. Upon this read, the SCL line is released, the PIN bit in the SBI0SR is set, and the SCL clock is driven out to receive a data word into the SBI0DBR. The master-transmitter generates an acknowledgement signal (i.e., a low level) on the SDA line following the last received bit.

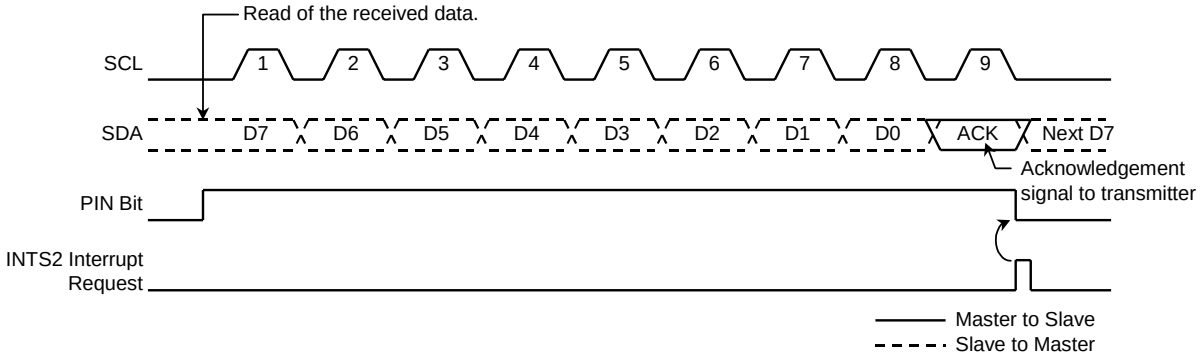


Figure 14.15 SBI0CR1.BC[2:0] = 000 and SBI0CR1.ACK = 1 (Master-Receiver Mode)

To prepare to terminate the data transfer, the master-receiver must clear the ACK bit in the SBI0CR1 immediately before the read of the second to last data word. This causes an acknowledge clock pulse not to be generated on the last data word.

When the transfer is complete, the INTS2 interrupt is generated. After interrupt processing, the INTS2 interrupt handler must set the BC[2:0] field in the SBI0CR1 to 001 and read the SBI0DBR,

so that a clock is generated on the SCL line once. With the ACK bit cleared, the master-receiver holds the SDA line high, which signals the end of transfer to the slave-transmitter.

Then, the SBI generates the INTS2 interrupt again, whereupon the INTS2 interrupt service routine must generate a STOP condition to stop communication via the I^2C bus.

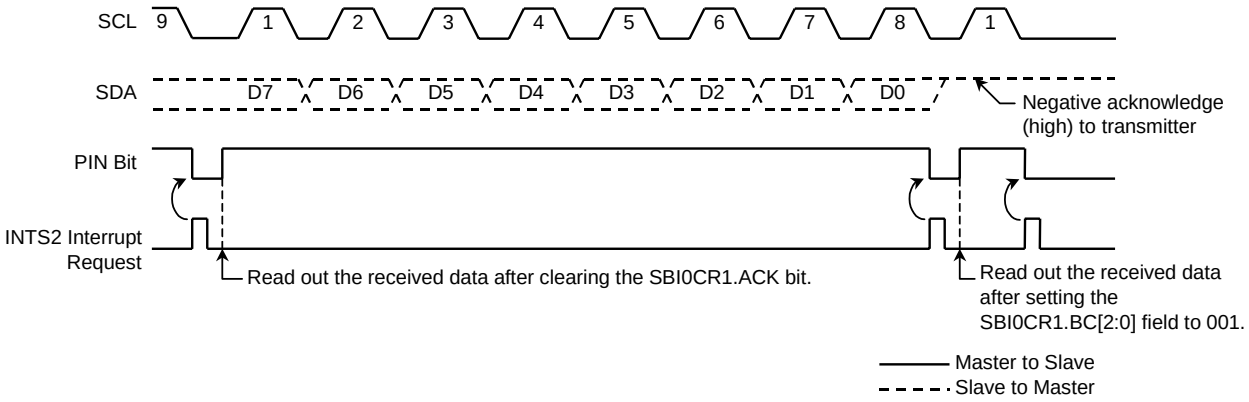


Figure 14.16 Terminating Data Transmission in Master-Receiver Mode

Example: When receiving N data words

INTS2 interrupt (after data transmission)

```
            7 6 5 4 3 2 1 0
SBI0CR1 ← X X X X 0 X X X     Set the number of bits to be received and specify whether ACK is required.
Reg.    ← SBI0DBR             Dummy read
End of interrupt
```

INTS2 interrupt (first to (N-2)th data reception)

```
            7 6 5 4 3 2 1 0
Reg.    ← SBI0DBR             Read the first to (N-2)th data words.
End of interrupt
```

INTS2 interrupt ((N-1)th data reception)

```
            7 6 5 4 3 2 1 0
SBI0CR1 ← X X X 0 0 X X X     Disable generation of acknowledgement clock.
Reg.    ← SBI0DBR             Read the (N-1)th data word.
End of interrupt
```

INTS2 interrupt (Nth data reception)

```
            7 6 5 4 3 2 1 0
SBI0CR1 ← 0 0 1 0 0 X X X     Generate a clock once.
Reg.    ← SBI0DBR             Read the Nth data word.
End of interrupt
```

INTS2 interrupt (after completing data reception)

```
            7 6 5 4 3 2 1 0
SBI0CR1 ← 0 0 1 0 0 X X X     Generate a clock once.
Reg.    ← SBI0DBR             Read the Nth data word.
End of interrupt
```

X = Don't care

(2) Slave Mode (SBI0CR2.MST = 0)

If the MST bit in the SBI0CR2 is cleared, the SBI is in slave mode. In slave mode, the SBI generates the INTS2 interrupt on four occasions: 1) when the SBI has received any slave address; 2) when the SBI has received a general-call address; 3) when the received slave address matches its own address in the I2C0AR; and 4) when a data transfer has been completed in response to a general-call.

Also, if the SBI, as a master, loses arbitration for the I^2C bus, it switches to slave mode. If arbitration is lost during a data transfer, SCL continues to be generated until the data word is complete; then the INTS2 interrupt is generated.

When the INTS2 interrupt occurs, the PIN bit in the SBI0SR is cleared, and the SCL line is pulled low. When the SBI0DBR is read or written or when the PIN bit is set back to 1, the SCL line is released after a period of t_{LOW}.

Processing to be done in slave mode varies, depending on whether or not the SBI has switched over to slave mode as a result of lost arbitration.

Test the AL, TRX, AAS and AD0 bits in the SBI0SR to determine the processing required, as summarized in Table 14.2.

Example: When the received slave address matches the SBI's own address and the data direction ($R/\overline{W}$) bit is 1

INTS2 interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBI0CR1 ← X X X 1 0 X X X      Set the number of bits to be transmitted.
SBI0DBR ← X X X X 0 X X X      Load the transmit data.
```

X = Don't care

Table 14.2 Processing in Slave Mode

TRX	AL	AAS	AD0	State	Processing
1	1	1	0	Arbitration was lost while the slave address was being transmitted, and the SBI received a slave address with the direction bit set transmitted by another master.	Set the SBI0CR1.BC[2:0] field to the number of bits in a data word and write the transmit data into the SBI0DBR.
	0	1	0	In slave-receiver mode, the SBI received a slave address with the direction bit set transmitted by the master.	
		0	0	In slave-transmitter mode, the SBI has completed a transmission of one data word.	Test the SBI0SR.LRB bit. If the LRB bit is set, that means the master-receiver does not require further data. Set the SBI0CR2.PIN bit to 1 and clear the TRX bit to 0 to release the bus. If the LRB bit is cleared, that means the master-receiver requires further data. Set the SBI0CR1.BC[2:0] field to the number of bits in the data word and write the transmit data to the SBI0DBR.
0	1	1	1/0	Arbitration was lost while a slave address was being transmitted, and received either a slave address with the direction bit cleared or a general-call address transmitted by another master.	Read the SBI0DBR (a dummy read) to set the SBI0CR2.PIN bit to 1, or write a 1 to this bit.
		0	0	Arbitration was lost while a slave address or a data word was being transmitted, and the transfer terminated.	
	0	1	1/0	In slave-receiver mode, the SBI received either a slave address with the direction bit cleared or a general-call address transmitted by the master.	
		0	1/0	In slave-receiver mode, the SBI has completed a reception of a data word.	Set the SBI0CR1.BC[2:0] field to the number of bits in the data word and read the received data from the SBI0DBR.

14.6.4 Generating a STOP Condition

When the SBI0SR.BB bit is set, setting the MST, TRX and PIN bits in the SBI0CR2 to 1 and clearing the BB bit in the same register causes the SBI to start a sequence for generating a STOP condition on the I^2C bus. Do not alter the contents of these bits until the STOP condition is present on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released (high) again; when SCL is high, the SBI drives the SDA pin high to generate a STOP condition.

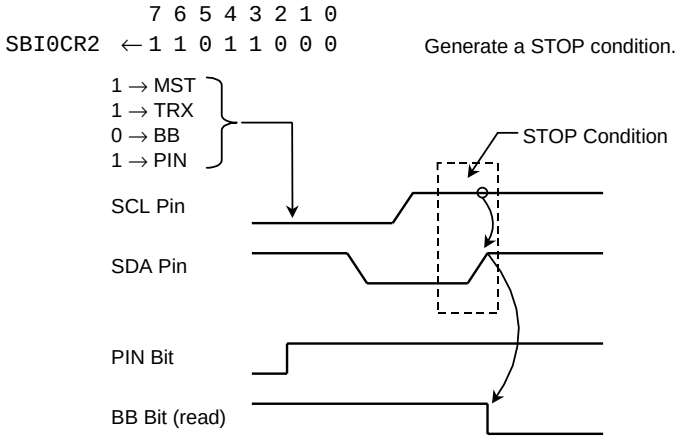


Figure 14.17 Generating a STOP Condition

14.6.5 Repeated START Condition

A data transfer is always terminated by a STOP condition. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition and address another slave or change the data direction without first generating a STOP condition. The following describes the steps required to generate a repeated START condition.

First, clear the MST, TRX and BB bits in the SBI0CR2 and set the PIN bit in the same register to release the bus. This causes the SDA pin to be held high and the SCL pin to be released. Because no STOP condition is generated on the bus, other devices think that the bus is busy.

Then, poll the SBI0SR.BB bit until it is cleared to ensure that the SCL pin is released. Next, poll the LRB bit until it is set to ensure that no other device is pulling the SCL bus line low. Once the bus is determined to be free this way, use the steps described in Section 14.6.2 to generate a START condition.

To satisfy the minimum setup time of the START condition, in Standard-mode, at least 4.7-μs wait period must be created by software after the bus becomes free.

```
                  7 6 5 4 3 2 1 0
SBI0CR2  ←0 0 0 1 1 0 0 0   Release the bus.
if SBI0SR<BB> ≠ 0           Check that the SCL pin is released.
Then
if SBI0SR<LRB> ≠ 1          Check that no other device is pulling the SCL line low.
Then
4.7-μs Wait
SBI0CR1 ← X X X 1 0 X X X   Select Acknowledge mode.
SBI0DBR ← X X X X X X X X   Load a slave address and the direction bit.
SBI0CR2 ← 1 1 1 1 1 0 0 0   Generate a START condition.
```

X = Don't care

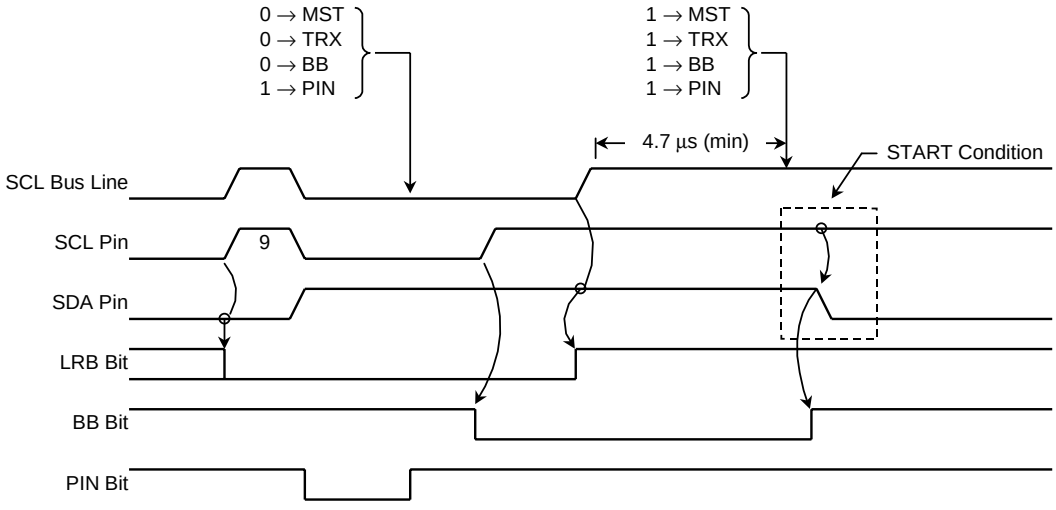


Figure 14.18 Repeated START Condition

14.7 Description of Registers Used in Clock-Synchronous 8-Bit SIO Mode

This section provides a summary of the registers which control clock-synchronous 8-bit SIO operation and provides its status information for monitoring.

Serial Bus Interface Control Register 1

SBI0CR1 (0xFFFF_F240)

	7	6	5	4	3	2	1	0
Name	SIOS	SIOINH	SIOM1	SIOM0	—	SCK2	SCK1	SCK0
Read/Write	W				—	W		R/W
Reset Value	0	0	0	0	—	0	0	1
Function	Start transfer 0: Stop 1: Start	Abort transfer 0: Continue 1: Abort	Transfer mode 00: Transmit mode 01: Reserved 10: Transmit/Receive mode 11: Receive mode			Serial clock frequency / Software reset monitor		

On writes: SCK[2:0] = Serial clock frequency

000	n = 3	1 MHz
001	n = 4	500 kHz
010	n = 5	250 kHz
011	n = 6	125 kHz
100	n = 7	62.5 kHz
101	n = 8	31.25 kHz
110	n = 9	15.63 kHz
111	—	External clock

Assumptions:
System clock: fc (= 32 MHz)
Clock gear: fc/1
ϕT0 = fperiph/4 (= 8 MHz)

Frequency = $\frac{\phi T0}{2^n}$ (Hz)

Note: Clear the SIOS bit and set the SIOINH bit before programming the transfer mode and serial clock frequency bits.

Serial Bus Interface Data Buffer Register

SBI0DBR (0xFFFF_F241)

	7	6	5	4	3	2	1	0
Name	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive)/ W (transmit)							
Reset Value	Undefined							

Figure 14.19 SIO Mode Registers (1)

Serial Bus Interface Control Register 2

SBI0CR2 (0xFFFF_F243)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	SBIM1	SBIM0	—	—
Read/Write	—	—	—	—	W		—	—
Reset Value	—	—	—	—	0	0	—	—
Function					SBI operating mode 00: Port mode 01: Clock-Synchronous 8-Bit SIO mode 10: I^2C Bus mode 11: Reserved			

Serial Bus Interface Register

SBI0SR (0xFFFF_F243)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	SIOF	SEF	—	—
Read/Write	—	—	—	—	R		—	—
Reset Value	—	—	—	—	0	0	—	—
Function					Serial transfer status 0: Terminated 1: In progress	Shift operation status		

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (0xFFFF_F244)

	7	6	5	4	3	2	1	0
Name	—	I2SBI0	—	—	—	—	—	—
Read/Write	—	R/W	—	—	—	—	—	W
Reset Value	—	0	—	—	—	—	—	
Function		IDLE 0: Off 1: On						Must be written as 0.

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0xFFFF_F245)

	7	6	5	4	3	2	1	0
Name	P4EN	—	—	—	—	—	—	—
Read/Write	R/W	—	—	—	—	—	—	—
Reset Value	0	—	—	—	—	—	—	—
Function	Internal clock 0: Off 1: On							Must be written as 0.

Figure 14.20 SIO Mode Registers (2)

14.8 Clock-Synchronous 8-Bit SIO Mode Operation

14.8.1 Serial Clock

(1) Clock Source

The clock source for the SIO mode can be selected from internal and external clocks through the programming of the SCK[2:0] field in the SBI0CR1.

- Internal clocks

  One of the seven internal clocks can be used as a serial clock, which is driven onto the SCK pin. At the beginning of a transfer, the SCK clock will start out at logic high.

  If software is slow and the reading of the received data or the writing of the transmit data can not keep up with the serial clock rate, the SBI automatically inserts a wait period, as shown below. During this period, the serial clock is temporarily stopped to suspend a shift operation.

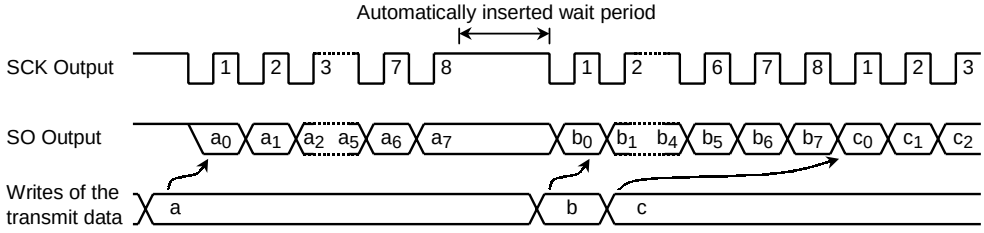


Figure 14.21 Automatic Wait Insertion

- External clock (SBI0CR1.SCK[2:0] = 111)

  If the SCK[2:0] field in the SBI0CR1 contains 111, the SBI uses an external clock supplied from the SCK pin as a serial clock. For proper shift operations, the clock high width and the clock low width must satisfy the following relationship.

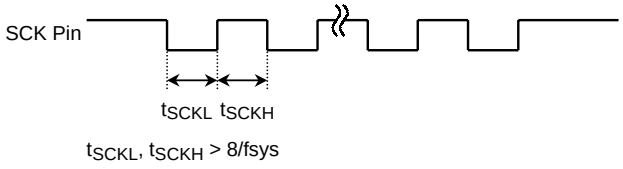


Figure 14.22 Maximum External Clock Frequency

(2) Shift Edge Types

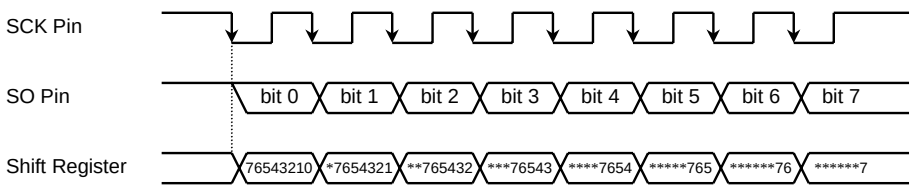
In transmit mode, leading-edge shift is used. In receive mode, trailing-edge shift is used.

- Leading-edge shift

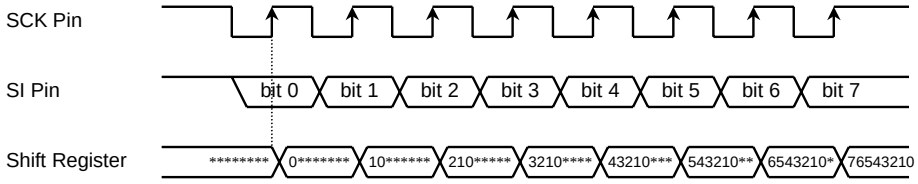
  Every bit of SIO data is shifted by the leading edge of the serial clock (falling edge of SCK).

- Trailing-edge shift

  Every bit of SIO data is shifted by the trailing edge of the serial clock (rising edge of SCK).



(a) Leading-Edge Shift



(b) Trailing-Edge Shift

Figure 14.23 Shift Edge Types

14.8.2 SIO Transfer Modes

The SBI supports three SIO transfer modes: receive mode, transmit mode and transmit/receive mode. The SIOM[1:0] field in the SBI0CR1 is used to select a transfer mode.

(1) 8-Bit Transmit Mode

Configure the SIO interface in transmit mode and write the transmit data into the SBI0DBR. Then setting the SIOS bit in the SBI0CR1 initiates a transmission. The contents of the SBI0DBR is moved to an internal shift register and then shifted out on the SO pin, with the least-significant bit (LSB) first, synchronous to the serial clock. Once the transmit data is transferred to the shift register, the SBI0DBR becomes empty, and the buffer-empty interrupt (INTS2) is generated.

In internal clock mode, the SIO interface will be in wait state (SCK will stop) until the INTS2 interrupt service routine provides the next transmit data to the SBI0DBR. Once the SBI0DBR is loaded, the SIO interface will automatically get out of the wait state.

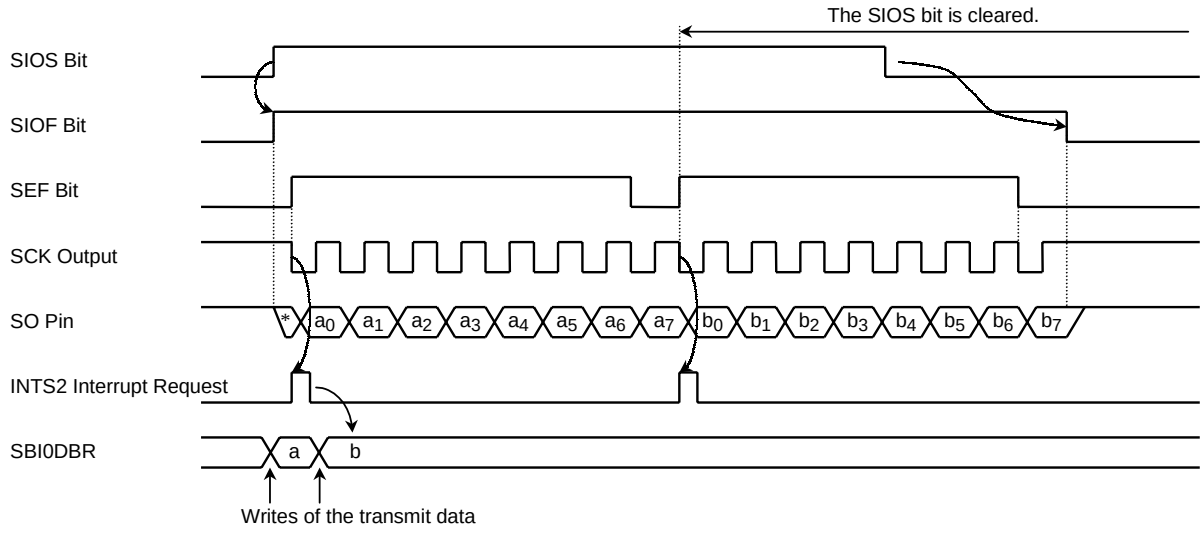
In external clock mode, the INTS2 interrupt service routine must provide the next transmit data to the SBI0DBR before the previous transmit data has been shifted out. Therefore, the data rate is a function of the maximum latency between when the INTS2 interrupt is generated and when the SBI0DBR is loaded by the interrupt service routine.

At the beginning of a transmission, the value of the last bit of the previously transmitted byte appears on the SO pin between when the SBI0SR.SIOF bit is set and when SCK subsequently goes low.

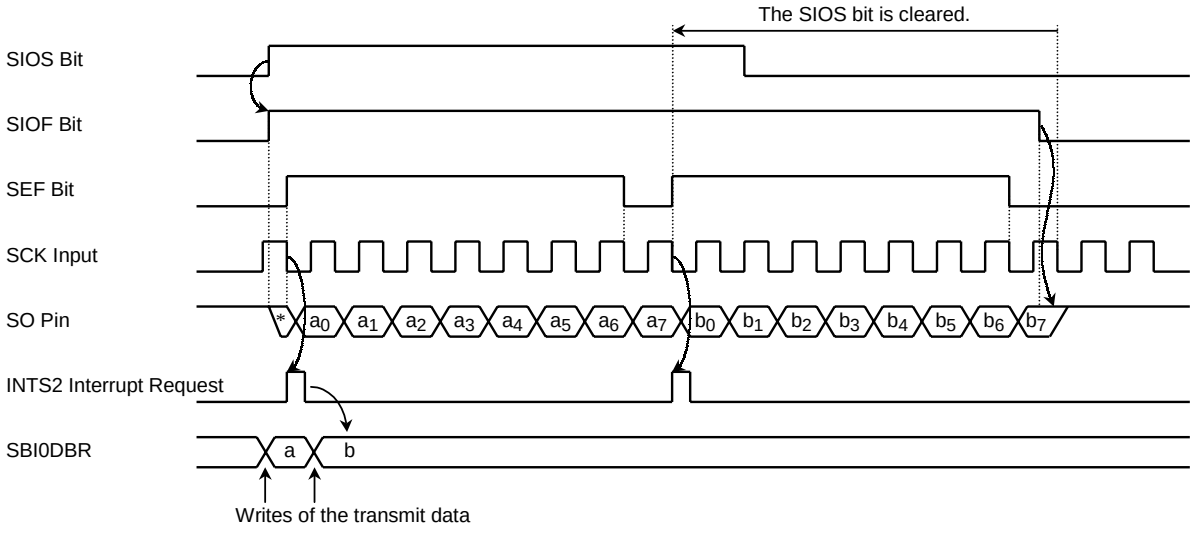
Transmission can be terminated by the INTS2 interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, the remaining bits in the SBI0DBR continue to be shifted out before transmission ends. In this case, software can check the SBI0SR.SIOF bit to determine whether transmission has come to an end (0 = end-of-transmission). If the SIOINH bit is set, the ongoing transmission is aborted immediately, and the SIOF bit is cleared at that point.

In external clock mode, the SIOS bit must be cleared before the SIO interface begins shifting out the next transmit data. Otherwise, the SIO will stop after sending out dummy data.

```
                7 6 5 4 3 2 1 0
SBI0CR1  ←  0 1 0 0 0 X X X      Select transmit mode.

SBI0DBR  ←  X X X X X X X X      Write the transmit data.
SBI0CR1  ←  1 0 0 0 0 X X X      Start transmission.

INTS2 interrupt

SBI0DBR  ←  X X X X X X X X      Write the next transmit data.
```



(a) Internal Clock Mode



(b) External Clock Mode

Figure 14.24 Transmit Mode

Example: MIP16 code to terminate transmission by SIOS (external clock mode)

```
          ADDIU r3, r0, 0x04
STEST1  : LB    r2,(SBI0SR)          ; If SBI0SR.SEF = 1 then loop
          AND   r2, r3
          BNEZ  r2, STEST1
          ADDIU r3, r0, 0x20
STEST2  : LB    r2, (PA)             ; If SCK = 0 then loop
          AND   r2, r3
          BEQZ  r2, STEST2
          ADDIU r3, r0, 0x00000111
          STB   r3, (SBI0CR1)        ; SIOS ← 0
```

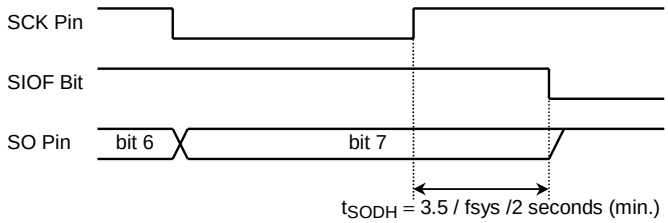


Figure 14.25 Retention Time of the Last Transmitted Bit

(2) 8-Bit Receive Mode

Configure the SIO interface in receive mode. Then setting the SIOS bit in the SBI0CR1 enables reception. The receive data is clocked into the internal shift register via the SI pin, synchronous to the serial clock. Once the shift register is fully loaded, the received byte is transferred to the SBI0DBR, and the buffer-full interrupt (INTS2) is generated. The INTS2 interrupt service routine must then pick up the received data from the SBI0DBR.

In internal clock mode, the SIO interface will be in wait state (SCK will stop) until the INTS2 interrupt service routine reads the data from the SBI0DBR.

In external clock mode, shift operations continue, synchronous to the external clock. In this mode, the maximum data rate is a function of the maximum latency between when the INTS2 interrupt is generated and when the SBI0DBR is read by the interrupt service routine.

Reception can be terminated by the INTS2 interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, reception continues until the shift register is fully loaded and transferred to the SBI0DBR. In this case, software can check the SBI0SR.SIOF bit to determine whether reception has come to an end (0 = end-of-reception). If the SIOINH bit is set, the ongoing reception is aborted immediately, and the SIOF bit is cleared at that point. (The received data becomes invalid; there is no need to read it out.)

Note: The contents of the SBI0DBR is not preserved after changing the transfer mode. Before changing the transfer mode, clear the SIOS bit to complete the ongoing reception and have the INTS2 interrupt service routine pick up the last received data.

```
                7 6 5 4 3 2 1 0
SBI0CR1 ← 0 1 1 1 0 X X X      Select receive mode.

SBI0CR1 ← 1 0 1 1 0 0 0 0      Start reception.

INTS2 interrupt

Reg.    ← SBI0DBR              Read the received data.
```

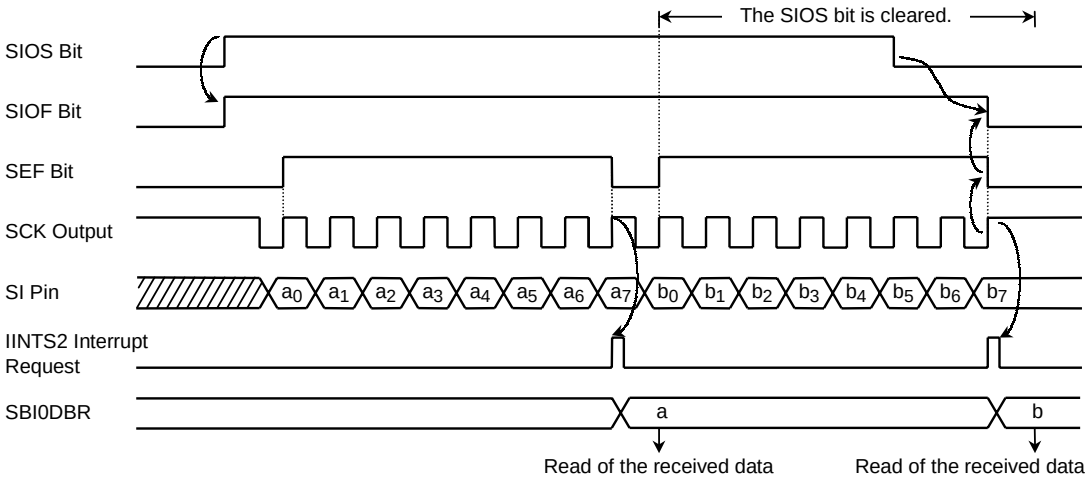


Figure 14.26 Receive Mode (Internal Clock Mode)

(3) 8-Bit Transmit/Receive Mode

Configure the SIO interface in transmit/receive mode and write the transmit data into the SBI0DBR. Then setting the SIOS bit in the SBI0CR1 initiates transmission and reception. The transmit data is shifted out through the SO pin, with the least-significant bit (LSB) first, with the falling edge of the serial clock, while at the same time the receive data is shifted in through the SI pin with the rising edge of the serial clock. Once the shift register is fully loaded with eight bits of the received data, it is transferred to the SBI0DBR, and the INTS2 interrupt is generated. The INTS2 interrupt service routine must then pick up the received data from the SBI0DBR and writes the next transmit data into the SBI0DBR. Because the SBI0DBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In internal clock mode, the SIO interface will be in wait state (SCK will stop) after a read of the received data until a write of the transmit data.

In external clock mode, shift operations continue, synchronous to the external clock. Therefore, software must read the received data and write the transmit data before the next shift operation begins. In this mode, the maximum data rate is a function of the maximum latency between when the INTS2 interrupt is generated and when the interrupt service routine reads the received data and writes the transmit data.

At the beginning of a transmission, the value of the last bit of the previously transmitted byte appears on the SO pin between when the SBI0SR.SIOF bit is set and when SCK subsequently goes low.

Transmission/reception can be terminated by the INTS2 interrupt service routine clearing the SIOS bit to 0 or setting the SIOINH bit to 1. If the SIOS bit is cleared, reception continues until the shift register is fully loaded and transferred to the SBI0DBR. In this case, software can check the SBI0SR.SIOF bit to determine whether transmission/reception has come to an end (0 = end-of-reception/transmission). If the SIOINH bit is set, the ongoing transmission/reception is aborted immediately, and the SIOF bit is cleared at that point.

Note:	**The contents of the SBI0DBR is not preserved after changing the transfer mode. Before changing the transfer mode, clear the SIOS bit to complete the ongoing transmission/reception and have the INTS2 interrupt service routine pick up the last received data.**

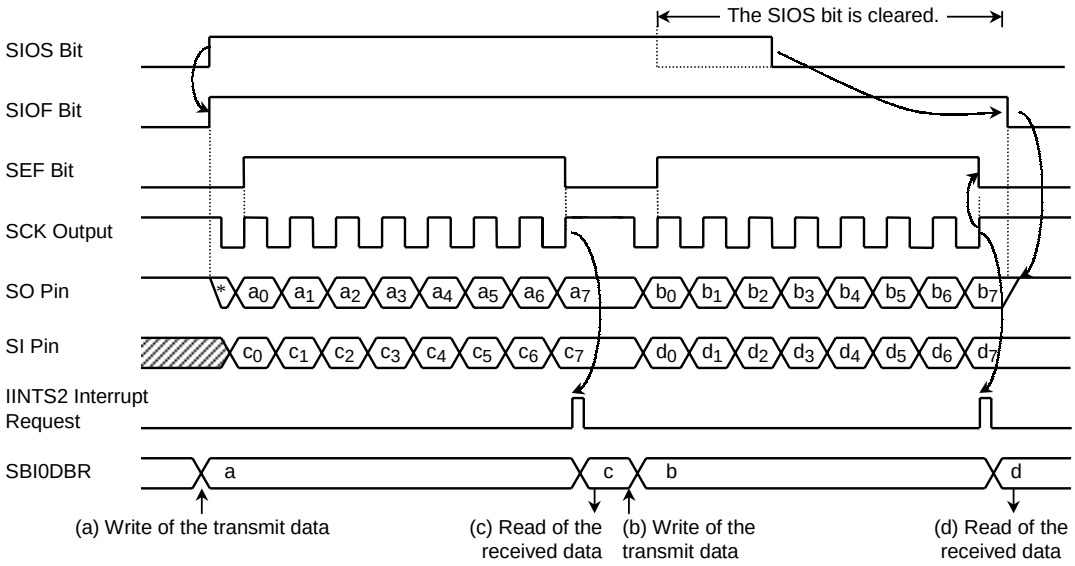


Figure 14.27 Receive/Transmit Mode (Internal Clock Mode)

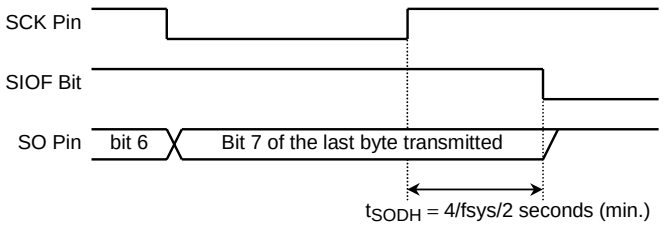


Figure 14.28 Retention Time of the Transmit Data in Receive/Transmit Mode

```
                7 6 5 4 3 2 1 0
SBI0CR1  ← 0 1 1 0 0 X X X      Select receive/transmit mode.

SBI0DBR  ← X X X X X X X X      Write the transmit data.
SBI0CR1  ← 1 0 1 0 0 X X X      Start reception/transmission.

INTS2 interrupt

Reg.     ← SBI0DBR              Read the received data.
SBI0DBR  ← X X X X X X X X      Write the transmit data.
```

15. Analog-to-Digital Converter (ADC)

The TMP1940CYAF has a 8-channel, multiplexed-input, 10-bit successive-approximation ananlog-to-digital converter (ADC).

Figure 15.1 shows a block diagram of the ADC. The eight analog input channels (AN0–AN7) can be used as general-purpose digital inputs (Port 5) if not needed as analog channels.

Note: Ensure that the ADC has halted before executing an insturction to place the TMP1940CYAF in IDLE, SLEEP or STOP mode to reduce power supply current. Otherwise, the TMP1940CYAF might go into a standby mode while the internal analog comparator is still active. In SLOW mode, the ADC must be disabled.

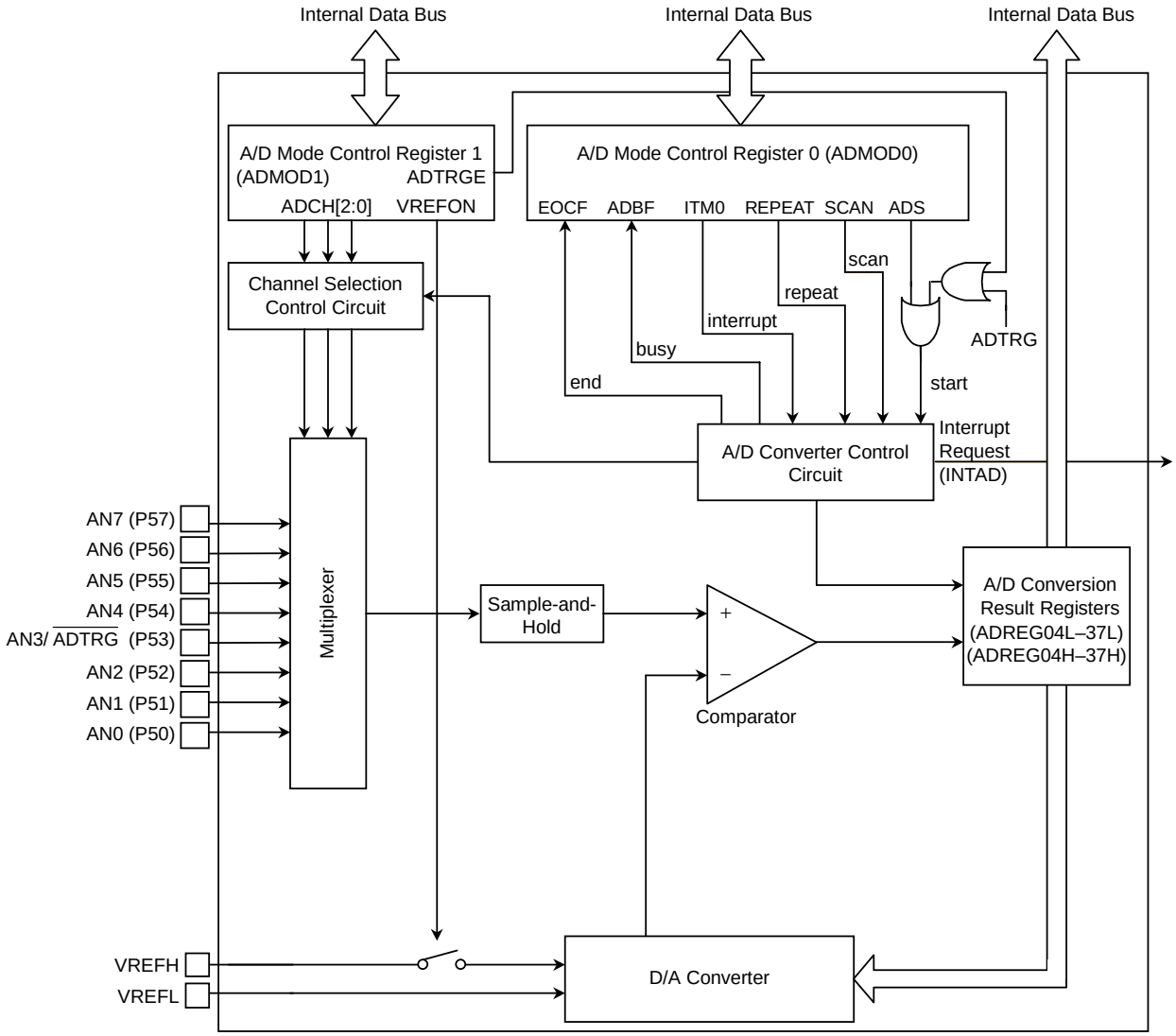


Figure 15.1 ADC Block Diagram

15.1 Register Description

The ADC has two mode control registers (ADMOD0 and ADMOD1), four conversion result high/low register pairs (ADREG04H/L, ADREG15H/L, ADREG26H/L, ADREG37H/L) and a clock select register (ADCCLK). The conversion result registers contain the digital values of completed conversions. The clock select register selects an A/D conversion clock.

Figure 15.2 to Figure 15.6 show the registers available in the ADC.

A/D Mode Control Register 0

ADMOD0 (0xFFFF_F310)

	7	6	5	4	3	2	1	0
Name	EOCF	ADBF	—	—	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R/W					
Reset Value	0	0	0	0	0	0	0	0
Function	End-of-conversion flag 0: Before or during conversion 1: Completed	A/D conversion busy flag 0: Idle 1: During conversion	Must be written as 0.	Must be written as 0.	Interrupt See below.	Continuous conversion mode 0: Single 1: Continuous	Channel scan mode 0: Fixed-channel 1: Channel scan	A/D conversion start 0: Don't care 1: Start This bit is always read as 0.

Interrupt in fixed-channel continuous conversion mode

	Fixed-Channel Continuous Conversion Mode SCAN = 0, REPEAT = 1
0	Generates INTAD interrupt when a single conversion has been completed.
1	Generates INTAD interrupt when a sequence of four conversions has been completed.

Note: The EOCF bit is cleared when read.

Figure 15.2 A/D Mode Control Register 0 (ADMOD0)

A/D Mode Control Register 1

ADMOD1 (0xFFFF_F311)

	7	6	5	4	3	2	1	0
Name	VREFON	I2AD	—	—	ADTRGE	ADCH2	ADCH1	ADCH0
Read/Write	R/W	R/W	—	—	R/W			
Reset Value	0	0	—	—	0	0	0	0
Function	VREF control 0: Off 1: On	ADC operation in IDLE mode 0: Off 1: On			External conversion trigger 0: Disable 1: Enable	Analog input channel select		

Analog Input Channel Select

ADCH[2:0]	SCAN	
	0 (Fixed-Channel Mode)	1 (Channel Scan Mode)
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011 (Note)	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

A/D external conversion trigger ($\overline{\text{ADTRG}}$ input)

0	Disable
1	Enable

Note 1: Set the VREFON bit to 1 before setting the ADS bit in the ADMOD0 to start a conversion.

Note 2: The AN3 pin is shared with the $\overline{\text{ADTRG}}$ pin. Therefore, when the external conversion trigger input ($\overline{\text{ADTRG}}$) is enabled (i.e., when ADMOD1.ADTRGE = 1), the ADCH[2:0] field must not be programmed to 011.

Figure 15.3 A/D Mode Control Register (ADMOD1)

A/D Conversion Result Low Register 0/4

ADREG04L (0xFFFF_F300)

	7	6	5	4	3	2	1	0
Name	ADR01	ADR00	—	—	—	—	—	ADR0RF
Read/Write	R		—	—	—	—	—	R
Reset Value	Undefined		—	—	—	—	—	0
Function	Lower 2 bits of an A/D conversion result							Conversion result store flag 1: Stored

A/D Conversion Result High Register 0/4

ADREG04H (0xFFFF_F301)

	7	6	5	4	3	2	1	0
Name	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 1/5

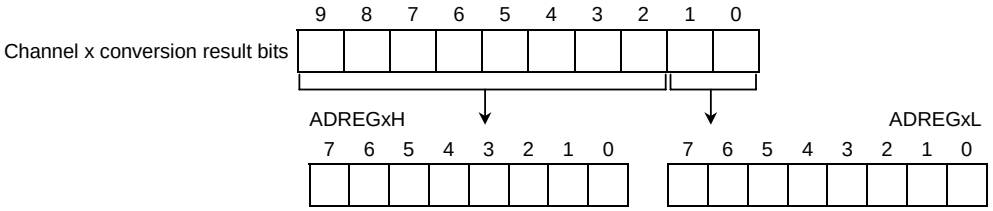
ADREG15L (0xFFFF_F302)

	7	6	5	4	3	2	1	0
Name	ADR11	ADR10	—	—	—	—	—	ADR1RF
Read/Write	R		—	—	—	—	—	R
Reset Value	Undefined		—	—	—	—	—	0
Function	Lower 2 bits of an A/D conversion result							Conversion result store flag 1: Stored

A/D Conversion Result High Register 1/5

ADREG15H (0xFFFF_F303)

	7	6	5	4	3	2	1	0
Name	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							



Note 1: Bits 5–1 are always read as 1s.

Note 2: Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when either the ADREGxH or the ADREGxL is read.

Figure 15.4 A/D Convesion Result High/Low Registers (1)

A/D Conversion Result Low Register 2/6

ADREG26L (0xFFFF_F304)

	7	6	5	4	3	2	1	0
Name	ADR21	ADR20	—	—	—	—	—	ADR2RF
Read/Write	R		—	—	—	—	—	R
Reset Value	Undefined		—	—	—	—	—	0
Function	Lower 2 bits of an A/D conversion result							Conversion result store flag 1: Stored

A/D Conversion Result High Register 2/6

ADREG26H (0xFFFF_F305)

	7	6	5	4	3	2	1	0
Name	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of an A/D conversion result							

A/D Conversion Result Low Register 3/7

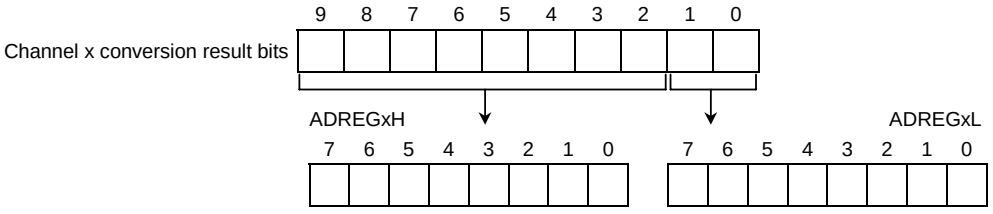
ADREG37L (0xFFFF_F306)

	7	6	5	4	3	2	1	0
Name	ADR31	ADR30	—	—	—	—	—	ADR3RF
Read/Write	R		—	—	—	—	—	R
Reset Value	Undefined		—	—	—	—	—	0
Function	Lower 2 bits of an AD conversion result							Conversion result store flag 1: Stored

A/D Conversion Result High Register 3/7

ADREG37H (0xFFFF_F307)

	7	6	5	4	3	2	1	0
Name	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
Reset Value	Undefined							
Function	Upper 8 bits of A/D conversion result							



Note 1 **Bits 5–1 are always read as 1s.**

Note 2 **Bit 0 (ADRxRF), when set, indicates that the conversion result has been stored in the ADREGxH/L register pair. This bit is cleared when either the ADREGxH or the ADREGxL is read.**

Figure 15.5 A/D Conversion Result High/Low Registers (2)

A/D Conversion Clock Select Register

ADCCLK (0xFFFF_EE04)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	ADCCK1	ADCCK0
Read/Write	—	—	—	—	—	—	R/W	R/W
Reset Value	—	—	—	—	—	—	0	0
Function							A/D conversion clock (fadc) select 00: fsys/2 01: fsys/4 10: fsys/8 11: Reserved	

Note 1: The ADC operates off the selected A/D conversion clock, which must be selected from Table 15.3, *Conversion Time*, to assure conversion accuracy.

Note 2: Programming the ADCCLK register should only be attempted when an A/D conversion is not in progress.

Figure 15.6 A/D Conversion Clock Select Register (ADCCLK)

15.2 Operation

15.2.1 Analog Reference Voltages

The VREFH and VREFL pins provide the reference voltages for the ADC. These pins estabilish the full-scale range for the internal resistor string, which divides the range into 1024 steps. The digital result of the conversion is derived by comparing the sampled analog input voltage to the resistor string voltages.

Clearing the VREFON bit in the ADMOD1 turns off the switch between VREFH and VREFL. Once the VREFON bit is cleared, the internal reference voltage requires a recovery time of 3 μs to stabilize after the VREFON bit is again set to 1. This recovery time is independent of the system clock frequency. The ADS bit in the ADMOD0 must then be set to initiate an conversion.

15.2.2 Selecting an Analog Input Channel (s)

There are two basic conversion modes: fixed-channel mode and channel scan mode. The SCAN bit in the ADMOD0 affects the conversion channel(s) that will be selected as follows.

- Fixed-channel mode (ADMOD0.SCAN = 0)

  When the SCAN bit in the ADMOD0 is cleared, the ADC runs conversions on a single input channel selected from AN0–AN7 via the ADCH[2:0] field in the ADMOD1.

- Channel scan mode (ADMOD0.SCAN = 1)

  When the SCAN bit in the ADMOD0 is set, the ADC runs conversions on sequential channels in a specific group selected via the ADCH[2:0] field in the ADMOD1.

Refer to Table 15.1. After a reset, the ADMOD0.SCAN bit defaults to 0, and the ADMOD1.ADCH[2:0] field defaults to 000. Thus, the AN0 pin is selected as the conversion channel. The AN0–AN7 pins can be used as general-purpose input ports if not used as analog input channels.

Table 15.1 Analog Input Channel Selection

ADMOD1.ADCH[2:0]	Fixed-Channel Mode ADMOD1.SCAN = 0	Channel Scan Mode ADMOD0.SCAN = 1
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

15.2.3 Starting an A/D Conversion

The ADC initiates a conversion or a sequence of conversions when the ADS bit in the ADMOD0 is set, or when a falling edge is applied to the $\overline{\text{ADTRG}}$ pin if the ADTRGE bit in the ADMOD1 is set. When a conversion starts, the Busy flag (ADMOD0.ADBF) is set.

Writing a 1 to the ADS bit causes the ADC to abort any ongoing conversion and start sampling the selected channel to begin a new conversion. The Conversion Result Store flag (ADREGxL.ADRxRF) indicates whether the result register contains a valid digital result at that point.

In external conversion trigger mode, a falling edge on the $\overline{\text{ADTRG}}$ pin is ignored while a conversion is in progress.

15.2.4 Conversion Modes and Conversion-Done Interrupts

The ADC supports the following four conversion modes:

- Fixed-channel single conversion mode
- Channel scan single conversion mode
- Fixed-channel continuous conversion mode
- Channel scan continuous conversion mode

The REPEAT and SCAN bits in the ADMOD1 select the conversion mode.

The ADC generates the INTAD interrupt and sets the EOCF bit in the ADMOD0 at the end of the conversion process.

- Fixed-Channel Single Conversion Mode

  This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 00. In this mode, the ADC performs a single conversion on a single selected channel. When a conversion is completed, the ADC sets the ADMOD0.EOCF bit, clears the ADMOD0.ADBF bit and generates the INTAD interrupt.

- Channel Scan Single Conversion Mode

  This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 01. In this mode, the ADC performs a single conversion on each of a selected group of channels. When a single conversion sequence is completed, the ADC sets the ADMOD0.EOCF bit, clears the ADMOD0.ADBF bit and generates the INTAD interrupt.

- Fixed-Channel Continuous Conversion Mode

  This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 10. In this mode, the ADC repeatedly converts a single selected channel. When a conversion process is completed, the ADC sets the ADMOD.EOCF bit. The ADMOD0.ADBF bit remains set.

  The ITM0 bit in the ADMOD0 controls interrupt generation in this mode. If the ITM0 bit is cleared, the ADC generates an interrupt after each conversion. If the ITM0 bit is set, the ADC generates an interrupt after every four conversions.

- Channel Scan Continuous Conversion Mode

  This mode is selected by programming the REPEAT and SCAN bits in the ADMOD0 to 11. In this mode, the ADC repeatedly converts the selected group of channels. When a single conversion sequence is completed, the ADC sets the ADMOD0.EOCF bit and generates the INTAD interrupt. The ADMOD0.ADBF bit remains set.

In continuous conversion modes, clearing the ADMOD0.REPEAT bit stops the conversion sequence after the ongoing conversion process is completed.

If the I2AD bit in the ADMOD1 is cleared, putting the TMP1940CYAF in any standby mode (IDLE, SLEEP or STOP) causes the ADC to be immediately disabled, even if a conversion is in progress. Once the TMP1940CYAF exits the standby mode, the ADC restarts a conversion sequence when in a continuous conversion mode, but remains inactive when in a single conversion mode.

Table 15.2 summarizes interrupt request generation in each of the conversion modes.

Table 15.2 Interrupt Request Generation in Each AD Conversion Mode

Mode	Interrupt Request Generation	ADMOD0		
		ITM0	REPEAT	SCAN
Fixed-Channel Single Conversion Mode	After a conversion	X	0	0
Channel Scan Single Conversion Mode	After a scan conversion sequence	X	0	1
Fixed-Channel Continuous Conversion Mode	After each conversion	0	1	0
	After every four conversions	1		
Channel Scan Continuous Conversion Mode	After each scan conversion sequence	X	1	1

X = Don't care

15.2.5 Conversion Time

The conversion process requires 86 conversion clocks per channel. For example, this results in a conversion time of 10.75 μs with 8-MHz fadc. The A/D conversion clock can be selected from fsys/2, fsys/4 and fsys/8 through the programming of the ADCCK[1:0] field in the ADCCLK register. To assure conversion accuracy, conversion time must be no shorter than 8.6 μs.

Table 15.3 Conversion Time

fsys	Conversion Clock		
	fsys/2	fsys/4	fsys/8
32 MHz	Don't use.	10.75 μs	21.5 μs
20 MHz	8.6 μs	17.2 μs	34.4 μs
16 MHz	10.75 μs	21.5 μs	43.0 μs
10 MHz	17.2 μs	34.4 μs	68.8 μs
8 MHz	21.5 μs	43.0 μs	86.0 μs

15.2.6 Storing and Reading the A/D Conversion Result

Conversion results are loaded into conversion result high/low register pairs (ADREG04H/L to ADREG37H/L). These registers are read-only.

In fixed-channel continuous conversion mode, conversion data goes into the ADREG04H/L to the ADREG37H/L sequentially. In other modes, channels AN0 and AN4 share the ADREG04H/L; channels AN1 and AN5 share the ADREG15H/L; channels AN2 and AN6 share the ADREG26H/L; and channels AN3 and AN7 share the ADREG37H/L.

Table 15.4 shows the relationships between the analog input channels and the A/D conversion result registers.

Table 15.4 Relationships Between Analog Input Channels and A/D Conversion Result Registers

Analog Input Channel (Port 5)	A/D Conversion Result Register	
	Fixed-Channel Continuous Conversion Mode (for each sequence of four conversions)	Other Modes
AN0	ADREG04H/L → ADREG15H/L → ADREG26H/L → ADREG37H/L → ADREG04H/L	ADREG04H/L
AN1		ADREG15H/L
AN2		ADREG26H/L
AN3		ADREG37H/L
AN4		ADREG04H/L
AN5		ADREG15H/L
AN6		ADREG26H/L
AN7		ADREG37H/L

Bit 0 (ADRxRF) in each ADREGxL register indicates whether the conversion result has been read. This bit is set when the conversion result is loaded into the ADREGxH/L pair, and cleared when either the ADREGxH or ADREGxL is read.

Reading the conversion result clears the End-of-Conversion flag (ADMOD0.EOCF).

15.3 Programming Examples

- Converting the analog input voltage on the AN3 pin to a digital value and storing the converted value in a memory location (0xFFFF_B800) using an A/D interrupt (INTAD) handler routine

Settings in the main routine

```
                  7 6 5 4 3 2 1 0
IMCEHH      ← X X 0 1 0 1 0 0   Enables INTAD and sets its priority level to 4.
ADMOD1      ← 1 X X X 0 0 1 1   Selects AN3 as the analog input channel.
ADMOD0      ← X X 0 0 0 0 0 1   Starts conversion in fixed-channel single conversion mode.
```

Interrupt routine processing example

```
r4          ←  ADREG37          Loads the conversion result into general-purpose register r4 from
                                ADREG37L and ADREG37H.
r4          > > 6               Shifts the contents of r4 six bits to the right, padding 0s to the
                                vacated MSB bits.
(FFFFB800H) ←  r4               Stores the contents of r4 to address 0xFFFF_B800.
```

- Converting the analog input voltages on AN0–AN2 sequentially in channel scan continuous conversion mode

```
IMCEHH      ← X X 0 1 0 0 0 0   Disables INTAD.
ADMOD1      ← 1 X X X 0 0 1 1   Selects AN0–AN2 as analog input channels.
ADMOD0      ← X X 0 0 0 0 0 1   Starts conversion in channel scan continuous conversion mode.
```

X = Don't care

Notes: The ADC supports both polled and interrupt-driven operation. The CPU can perform polling operation to detect completion of a conversion.

- **Don't poll the ADRxRF bit in the ADREGxxL register.**
- **In single conversion modes, poll the ADBF bit in the ADMOD0.**
- **In any conversion modes, the EOCF bit in the ADMOD0 can be polled. After the EOCF bit is set, one or two fadc clocks are required as shown below before the ADREGxH/L can be read.**

Conversion Mode	Time Required Before Reading the ADREGxx
Fixed-channel single conversion mode	1 fadc clock
Fixed-channel continuous conversion mode	1 fadc clock
Channel scan single scan conversion mode	2 fadc clocks
Channel scan continuous conversion mode	2 fadc clocks

fadc: A/D conversion clock selected by the ADCCLK register

16. Watchdog Timer (WDT)

The TMP1940CYAF contains a watchdog timer (WDT). The WDT is used to regain control of the system in the event of software or system lockups due to spurious noises, etc. When a watchdog timer time-out occurs, the WDT generates a nonmaskable interrupt to the CPU.

Also, the time-out event can be programmed for system reset generation, which is accomplished by routing the time-out signal to the internal reset pin.

16.1 Implementation

Figure 16.1 shows a block diagram of the WDT.

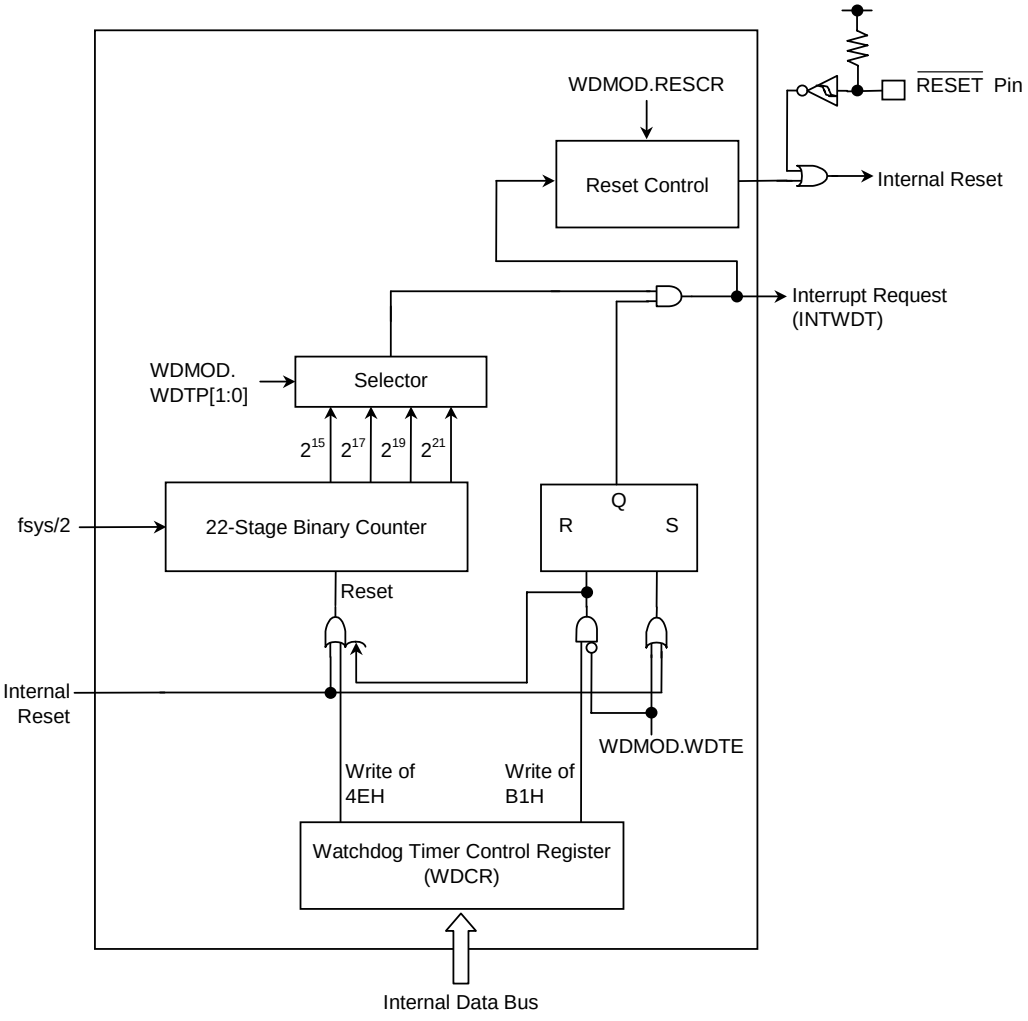


Figure 16.1 WDT Block Diagram

The WDT contains a 22-stage binary counter clocked by the fsys/2 clock. This binary counter provides 2^{15}, 2^{17}, 2^{19} or 2^{21} as a counter overflow signal, as programmed into the WDTP[1:0] field in the WDMOD. When a counter overflow occurs, the WDT generates a WDT interrupt, as shown below.

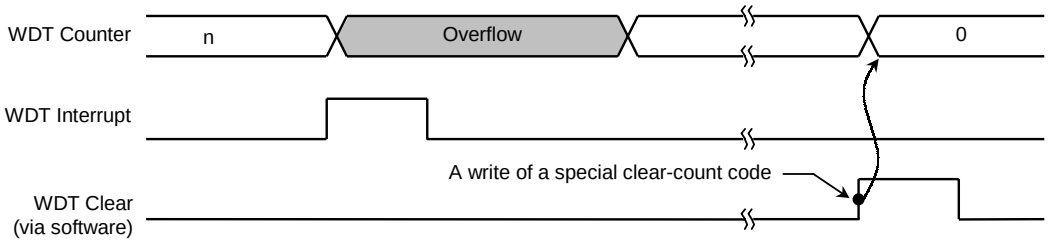
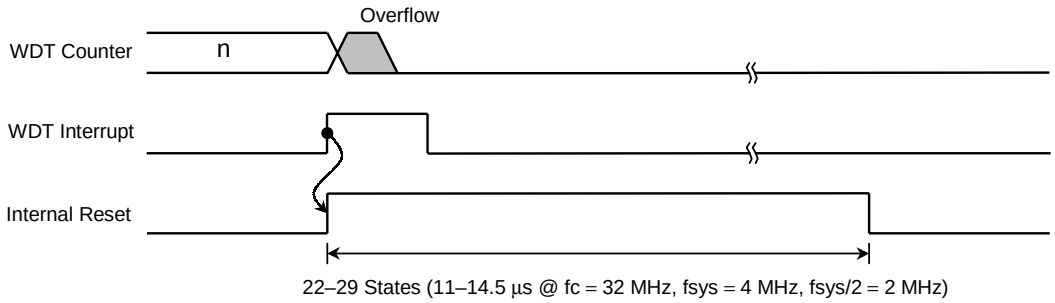


Figure 16.2 Default Operation

Also, the counter overflow can be programmed to cause a system reset as the time-out action. If so programmed, a counter overflow causes the WDT to assert the internal reset signal for a 22- to 29-state time. After a reset, the fsys clock is, by default, generated by dividing the high-speed oscillator clock (fc) by eight through the clock gear function; the WDT clock source (fsys/2) is derived from this fsys clock.



Note: The TMP1940CYAF continues sampling the $\overline{\text{PLLOFF}}$ pin during a reset operation caused by the WDT. Therefore, the $\overline{\text{PLLOFF}}$ pin must be tied to either logic high or logic low.

Figure 16.3 Reset Operation

16.2 Register Description

The WDT is controlled by two registers called WDMOD and WDCR.

16.2.1 Watchdog Timer Mode Register (WDMOD)

- Time-out Period (WDMOD.WDTP[1:0])

  This 2-bit field determines the duration of the WDT time-out interval. Upon reset, the WDTP[1:0] field defaults to 00. Figure 16.5 shows possible time-out periods.

- WDT Enable (WDMOD.WDTE)

  Upon reset, the WDTE bit is set to 1, enabling the WDT. To disable the WDT, the clearing of the WDTE bit must be followed by a write of a special key code (B1H) to the WDCR register. This prevents a “lost” program from disabling the WDT operation. The WDT can be re-enabled only by setting the WDTE bit.

- System Reset (WDMOD.RESCR)

  This bit is used to program the WDT to generate a system reset on a time-out. Upon reset, this bit is cleared; thus the time-out does not cause a system reset.

16.2.2 Watchdog Timer Control Register (WDCR)

This register is used to disable the WDT and to clear the WDT binary counter.

- Disabling the WDT

  The WDT can be disabled by clearing the WDMOD.WDTE to 0 and then writing the special disable code (B1H) to the WDCR register.

```
WDMOD   ← 0 – – – – – – –    Clears the WDTE bit to 0.
WDCR    ← 1 0 1 1 0 0 0 1    Writes the disable code (B1H) to the WDCR.
```

- Enabling the WDT

  The WDT can be enabled only by setting the WDTE bit in the WDMOD to 1.

- Clearing the WDT counter

  Writing the special clear-count code (4EH) to the WDCR resets the binary counter to zero. The counting process begins again.

```
WDCR    ← 0 1 0 0 1 1 1 0    Writes the clear-count code (4EH) to the WDCR.
```

Note: Writing the disable code (B1H) to the WDCR causes the binary counter to be reset to zero.

WDMOD (0xFFFF_F090)

	7	6	5	4	3	2	1	0
Name	WDTE	WDTP1	WDTP0	—	—	I2WDT	RESCR	—
Read/Write	R/W	R/W		—	—	R/W		R/W
Reset Value	1	0	0	—	—	0	0	0
Function	WDT enable 1: Enable	Time-out period 00: 2^{16}/fsys 01: 2^{18}/ fsys 10: 2^{20}/ fsys 11: 2^{22}/ fsys				IDLE 0: Off 1: On	1: System reset	Must be written as 0.

System reset

0	—
1	Internally routes the WDT time-out signal to the system reset

Time-out peiord (@ fc = 32 MHz, fs = 32.768 kHz)

System Clock Select SYSCR1.SYSCK	Clock Gear Value SYSCR1.GEAR[1:0]	Watchdog Timer Time-out Period WDMOD.WDTP[1:0] 00	01	10	11
1 (fs)	xxx	2.0 s	8.0 s	32.0 s	128.0 s
0 (fgear)	00 (fc)	2.048 ms	8.192 ms	32.768 ms	131.072 ms
	01 ($fc/_2$)	4.096 ms	16.384 ms	65.536 ms	262.144 ms
	10 ($fc/_4$)	8.192 ms	32.768 ms	131.072 ms	524.288 ms
	11 ($fc/_8$)	16.384 ms	65.536 ms	262.144 ms	1048.576 ms

WDT enable

0	Disable
1	Enable

Figure 16.4 Watchdog Timer Mode Register (WDMOD)

WDCR (0xFFFF_F091)

	7	6	5	4	3	2	1	0
Name	—							
Read/Write	W							
Reset Value	—							
Function	B1H: WDT disable code 4EH : WDT clear-count code							

Special code

B1H	WDT disable code
4EH	WDT clear-count code
Other values	Don't care

Figure 16.5 Watchdog Timer Control Register (WDCR)

16.3 Operation

The watchdog timer is a kind of timer that generates an interrupt request if it times out. The WDT of the TMP1940CYAF allows the user to program the time-out period in the WDTP[1:0] field in the WDMOD. While enabled, the software can reset the counter to zero at any time by writing a special clear-count code. If the software is unable to reset the counter before it reaches the time-out count, the WDT generates the INTWDT interrupt. In response to the interrupt, the CPU jumps to a system recovery routine to regain control of the system.

The WDT begins counting immediately after reset.

When the TMP1940CYAF goes into SLEEP or STOP mode, the WDT counter is reset to zero automatically and stops counting. The WDT continues counting while an off-chip peripheral has mastership of the bus (i.e., $\overline{\text{BUSAK}}$ = 0).

In IDLE mode, the I2WDT bit in the WDMOD determines whether or not to disable the WDT. The I2WDT bit can be programmed before putting the TMP1940CYAF in IDLE mode.

Examples:

- Clearing the WDT binary counter

```
                7 6 5 4 3 2 1 0
WDCR     ← 0 1 0 0 1 1 1 0      Writes the clear-count code (4EH) to the WDCR.
```

- Programming the time-out interval to 2^{18}/fsys

```
                7 6 5 4 3 2 1 0
WDMOD    ← 1 0 1 – – – – –
```

- Disabling the watchdog timer

```
                7 6 5 4 3 2 1 0
WDMOD    ← 0 – – – – – – –      Clears the WDTE bit to 0.
WDCR     ← 1 0 1 1 0 0 0 1      Writes the disable code (B1H) to the WDCR.
```

17. Real-Time Clock (RTC)

The TMP1940CYAF contains a real-time clock (RTC). Clocked by a 32.768-kHz clock, the RTC provides a periodic interrupt at a programmed interval: 0.0625 seconds, 0.125 seconds, 0.25 seconds or 0.50 seconds.

The RTC can continue operating in any standby modes in which the low-speed oscillator is active.

The RTC interrupt (INTRTC) can be used as a wake-up signal to exit a standby mode (except STOP mode). The IMCGB3 register located within the CG must be programmed to use the INTRTC interrupt.

17.1 Implemention

Figure 17.1 shows a block diagram of the RTC.

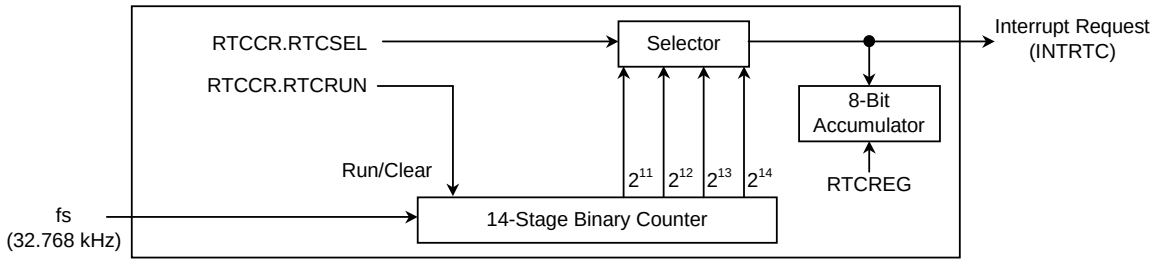


Figure 17.1 RTC Block Diagram

The RTC Control Register (RTCCR) provides control over the RTC. The organization of the RTCCR is shown below.

RTCCR (0xFFFF_F0A0)

	7	6	5	4	3	2	1	0
Name	—				RTCRCLR	RTCSEL1	RTCSEL0	RTCRUN
Read/Write	R/W				R/W	R/W		R/W
Reset Value	0				0	0	0	0
Function	Must be written as 0.				Accumu-lator clear 0: Clear RTCREG 1: Don't care	00: 2^{14}/fs 01: 2^{13}/fs 10: 2^{12}/fs 11: 2^{11}/fs		0: Stop and clear the counter. 1: Begin counting.

Interrupt interval (fs = 32.768 kHz)

00	0.50 seconds
01	0.25 seconds
10	0.125 seconds
11	0.0625 seconds

Figure 17.2 RTC Control Register (RTCCR)

The RTC provides an 8-bit read-only accumulator (RTCREG) that counts the number of INTRTC interrupts that have occurred. The accumulator allows the user to keep track of time up to 127.5 seconds if the interrupt interval is programmed to 0.5 seconds.

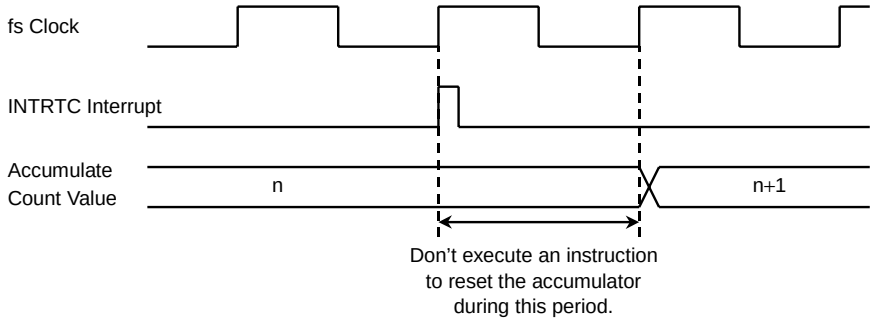
Accumulator

RTCREG (0xFFFF_F0A4)

	7	6	5	4	3	2	1	0
Name	RUI7	RUI6	RUI5	RUI4	RUI3	RUI2	RUI1	RUI0
Read/Write	R							
Reset Value	0	0	0	0	0	0	0	0
Function	Accumulate count value							

Figure 17.3 RTC Accumulator Register (RTCREG)

The RTCREG is incremented with a delay of one fs clock after the INTRTC interrupt is generated. Reads of the RTCREG must be performed in SLOW mode. The resetting of the RTCREG is inhibited for one fs clock cycle after the INTRTC interrupt is generated. The RTCREG can be reset to zero by executing the accumulator-clear command twice in SLOW mode.



Example 1: Clearing the accumulator

```
                7 6 5 4 3 2 1 0
SYSCR1  ← X X 1 – – X – –          Puts the TMP1940CYAF in SLOW mode.
RTCCR   ← 0 X X X 0 – – 1 }
RTCCR   ← 0 X X X 0 – – 1 }        Executes accumulator-clear command twice.
SYSCR1  ← X X 0 – – X – –          Puts the TMP1940CYAF back in NORMAL Mode.
```

Example 2: Programming the RTC interrupt interval

Initialization

```
                7 6 5 4 3 2 1 0
IMCGB3  ← 0 0 1 1 0 0 0 1
IMCEHL  ← 0 0 0 1 0 X X X          Sets the interrupt level.
EICRCG  ← 0 0 0 0 0 1 1 1          Clears the interrupt request via the CG block.
INTCLR  ← 0 0 1 1 1 0 1 0          Clears the interrupt request via the INTC block.
RTCCR   ← 0 0 0 0 1 X X 1          Starts counting.
```

INTRTC interrupt

```
                7 6 5 4 3 2 1 0
EICRCG  ← 0 0 0 0 0 1 1 1          Clears the interrupt request via the CG block.
INTCLR  ← 0 0 1 1 1 0 1 0          Clears interrupt request via the INTC block.
Interrupt processing
End of interrupt
```

X = Don't care

Note: To disable interrupts, program the IMCEHL and then the IMCGB3 in this order.

18. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK=0) and a clock gear factor of 1/fc (SYSCR1.GEAR[1:0]=00).

18.1 Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC}	−0.5 to 4.0	V
Input voltage		V_{IN}	−0.5 to V_{CC} + 0.5	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	80	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	−80	
Power dissipation (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−65 to 150	°C
Operating temperature		T_{OPR}	−40 to 85	°C

$V_{CC} = DV_{CC} = AV_{CC}$; $V_{SS} = DV_{SS} = AV_{SS}$

Note: Maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no maximum rating value is exceeded with respect to current, voltage, power dissipation, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

18.2 DC Electrical Characteristics (1/2)

Ta = –40 to 85°C

Parameter		Symbol	Condition		Min	Typ (Note 1)	Max	Unit
Supply voltage $AV_{CC} = V_{CC}$ $AV_{SS} = V_{SS} = 0$ V		V_{CC}	PLLON	fosc = 5 to 8 MHz fsys = 2.5 to 32 MHz fs = 30 to 34 kHz	3.0		3.6	V
				fosc = 5 to 6.5 MHz fsys = 2.5 to 26 MHz fs = 30 to 34 kHz	2.7			
			PLLOFF (Crystal)	fosc = 16 to 20 MHz fsys = 1 to 20 MHz fs = 30 to34 kHz	2.7			
			PLLOFF (External clock)	fosc = 16 to 20 MHz fsys = 1 to 20 MHz fs = 30 to 34 kHz	2.7			
				fosc = 20 to 32 MHz fsys = 1.25 to 16 MHz fs = 30 to 34 kHz (SYSCR1.DFOSC = 0) (Note 2)				
Low-level input voltage	P00–P17 (AD0–15)	V_{IL}	$V_{CC} \geq 2.7$ V		–0.3		0.6	V
	P20–PA7 (except P77)	V_{IL1}					$0.3V_{CC}$	
	$\overline{PLLOFF}$, BW0, BW1, $\overline{RESET}$, $\overline{NMI}$, P77 (INT0)	V_{IL2}					$0.25V_{CC}$	
	X1	V_{IL4}					$0.2V_{CC}$	
High-level input voltage	P00–P17 (AD0–15)	V_{IH}			2.0		$V_{CC} + 0.3$	
	P20–PA7 (except P77)	V_{IH1}			$0.7V_{CC}$			
	$\overline{PLLOFF}$, BW0, BW1, $\overline{RESET}$, $\overline{NMI}$, P77 (INT0)	V_{IH2}			$0.80V_{CC}$			
	X1	V_{IH4}			$0.8V_{CC}$			
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	$V_{CC} \geq 2.7$ V			0.45	V
High-level output voltage		V_{OH}	I_{OH} = –400 μA		2.4			

Note 1: V_{CC} = 3.3 V, Ta = 25°C, unless otherwise noted.

Note 2: The DFOSC bit in the SYSCR1 register must be cleared to 0.

18.3 DC Electrical Characteristics (2/2)

Ta = –40 to 85°C

Parameter	Symbol	Condition	Min	Typ (Note 1)	Max	Unit
Input leakage current	I_{LI}	$0.0 \le V_{IN} \le V_{CC}$		0.02	± 5	μA
Output leakage current	I_{LO}	$0.2 \le V_{IN} \le V_{CC} - 0.2$		0.05	± 10	
Power-down voltage (STOP mode, RAM backup)	V_{STOP}	$V_{IL2} = 0.2V_{CC}$, $V_{IH2} = 0.8V_{CC}$	2.2		3.6	V
Pull-up resistor at Reset	RRST	V_{CC} = 3.3 V ± 0.3 V	100		550	kΩ
Pin capacitance (except power/ground pins)	C_{IO}	fc = 1 MHz			10	pF
Schmitt hysteresis $\overline{\text{PLLOFF}}$, BW0, BW1, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0	V_{TH}	$V_{CC} \ge 2.7$ V	0.4			V
Programmable pull-up resistor	PKH	V_{CC} = 3.3 V ± 0.3 V	100		550	kΩ
NORMAL (Note 2); Gear = 1/1	I_{CC}	V_{CC} = 3.3V ± 0.3 V fsys = 32 MHz (fosc= 8 MHz, PLLON) INTLV = H		48	58	mA
NORMAL (Note 2); Gear = 1/8				10	13	
IDLE (Doze)				18.5	22.5	
IDLE (Halt)				16	19.5	
NORMAL (Note 2); Gear = 1/1		V_{CC} = 3.3 V ± 0.3 V fsys = 20 MHz (fosc = 20 MHz, PLLOFF) INTLV = L		32	38	mA
NORMAL (Note 2); Gear = 1/8				6	8	
IDLE (Doze)				11.5	15.3	
IDLE (Halt)				10	12.4	
SLOW (Note 3)		V_{CC} = 3.3 V ± 0.3 V fs = 32.768 kHz SYSCR2.DRVOSCL = 1		47	90	μA
SLLEP (Note 3)		V_{CC} = 3.3 V ± 0.3 V fs = 32.768 kHz SYSCR2.DRVOSCL = 1		3	35	μA
STOP		V_{CC} = 2.7 to 3.6 V		0.5	15	μA

Note 1: **V_{CC} = 3.3 V, Ta = 25°C, unless otherwise noted.**

Note 2: **Measured with the CPU operating; two TMRAs, one TMRB and a DMAC channel on; and input pin levels held at fixed logic levels. IREF excluded.**

Note3: **Measured with RTC on and low-speed oscillator drive capability reduced to low (SYSCR2.DRVOSCL=1).**

18.4 AC Electrical Characteristics

(1) V_{CC} = 3.0 to 3.6 V, Ta = 0 to 70°C, ALE width = 0.5 clock cycle (recommended when t_{SYS} is 50 ns or longer)

No.	Parameter	Symbol	Equation		fsys = 20 MHz *		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	31.25	33333	50		ns
2	A0–A15 valid to ALE low	t_{AL}	0.4x – 12		8		ns
3	A0–A15 hold after ALE low	t_{LA}	0.4x – 8		12		ns
4	ALE pulse width high	t_{LL}	0.4x – 6		14		ns
5	ALE low to $\overline{RD}$ or $\overline{WR}$ asserted	t_{LC}	0.4x – 8		12		ns
6	$\overline{RD}$ or $\overline{WR}$ negated to ALE high	t_{CL}	x – 15		35		ns
7	A0–A15 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACL}	x – 20		30		ns
8	A0–A23 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACH}	x – 20		30		ns
9	A0–A23 hold after $\overline{RD}$ or $\overline{WR}$ negated	t_{CAR}	x – 15		35		ns
10	A0–A15 valid to D0–D15 data in	t_{ADL}		x (2 + W) – 42		58	ns
11	A0–A23 valid to D0–D15 data in	t_{ADH}		x (2 + W) – 42		58	ns
12	$\overline{RD}$ asserted to D0–D15 data in	t_{RD}		x (1 + W) – 28		22	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 10		40		ns
14	D0–D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0–A15 output	t_{RAE}	x – 15		35		ns
16	$\overline{WR}$ width low	t_{WW}	x (1 + W) – 10		40		ns
17	D0–D15 valid to $\overline{WR}$ negated	t_{DW}	x (1 + W) – 18		32		ns
18	D0–D15 hold after $\overline{WR}$ negated	t_{WD}	x – 15		35		ns
19	A0–A23 valid to $\overline{WAIT}$ input	t_{AWH}		1.5x – 30		45	ns
20	A0–A15 valid to $\overline{WAIT}$ input	t_{AWL}		1.5x – 30		45	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$ or $\overline{WR}$ asserted	t_{CW}	(0.5 + N – 1) x + 2	(0.5 + N) x – 17	27	58	ns

* W = 0

W: Number of wait-state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (1 + N) wait insertion

AC measurement conditions:

- Output levels: High = 2.4 V, Low = 0.45 V, CL = 30 pF
- Input levels: High = 2 V, Low = 0.6 V

(2) V_{CC} = 3.0 to 3.6 V, Ta = 0 to 70°C, ALE width = 1.5 clock cycles

No.	Parameter	Symbol	Equation		fsys = 32 MHz*		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	31.25	33333			ns
2	A0–A15 valid to ALE low	t_{AL}	1.4x – 12		31		ns
3	A0–A15 hold after ALE low	t_{LA}	0.4x – 8		4		ns
4	ALE pulse width high	t_{LL}	1.4x – 6		37		ns
5	ALE low to $\overline{RD}$ or $\overline{WR}$ asserted	t_{LC}	0.4x – 8		4		ns
6	$\overline{RD}$ or $\overline{WR}$ negated to ALE high	t_{CL}	x – 15		16		ns
7	A0–A15 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACL}	2x – 20		42		ns
8	A0–A23 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACH}	2x – 20		42		ns
9	A0–A23 hold after $\overline{RD}$ or $\overline{WR}$ negated	t_{CA}	x – 15		16		ns
10	A0–A15 valid to D0–D15 data in	t_{ADL}		x (3 + W) – 42		51	ns
11	A0–A23 valid to D0–D15 data in	t_{ADH}		x (3 + W) – 42		51	ns
12	$\overline{RD}$ asserted to D0–D15 data in	t_{RD}		x (1 + W) – 28		3	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 10		21		ns
14	D0–D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0–A15 output	t_{RAE}	x – 15		16		ns
16	$\overline{WR}$ width low	t_{WW}	x (1 + W) – 10		21		ns
17	D0–D15 valid to $\overline{WR}$ negated	t_{DW}	x (1 + W) – 18		13		ns
18	D0–D15 hold after $\overline{WR}$ negated	t_{WD}	x – 15		16		ns
19	A0–A23 valid to $\overline{WAIT}$ input	t_{AWH}		2.5x – 30		48	ns
20	A0–A15 valid to $\overline{WAIT}$ input	t_{AWL}		2.5x – 30		48	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$ or $\overline{WR}$ asserted	t_{CW}	(0.5 + N – 1) x + 2	(0.5 + N) x – 17	18	29	ns

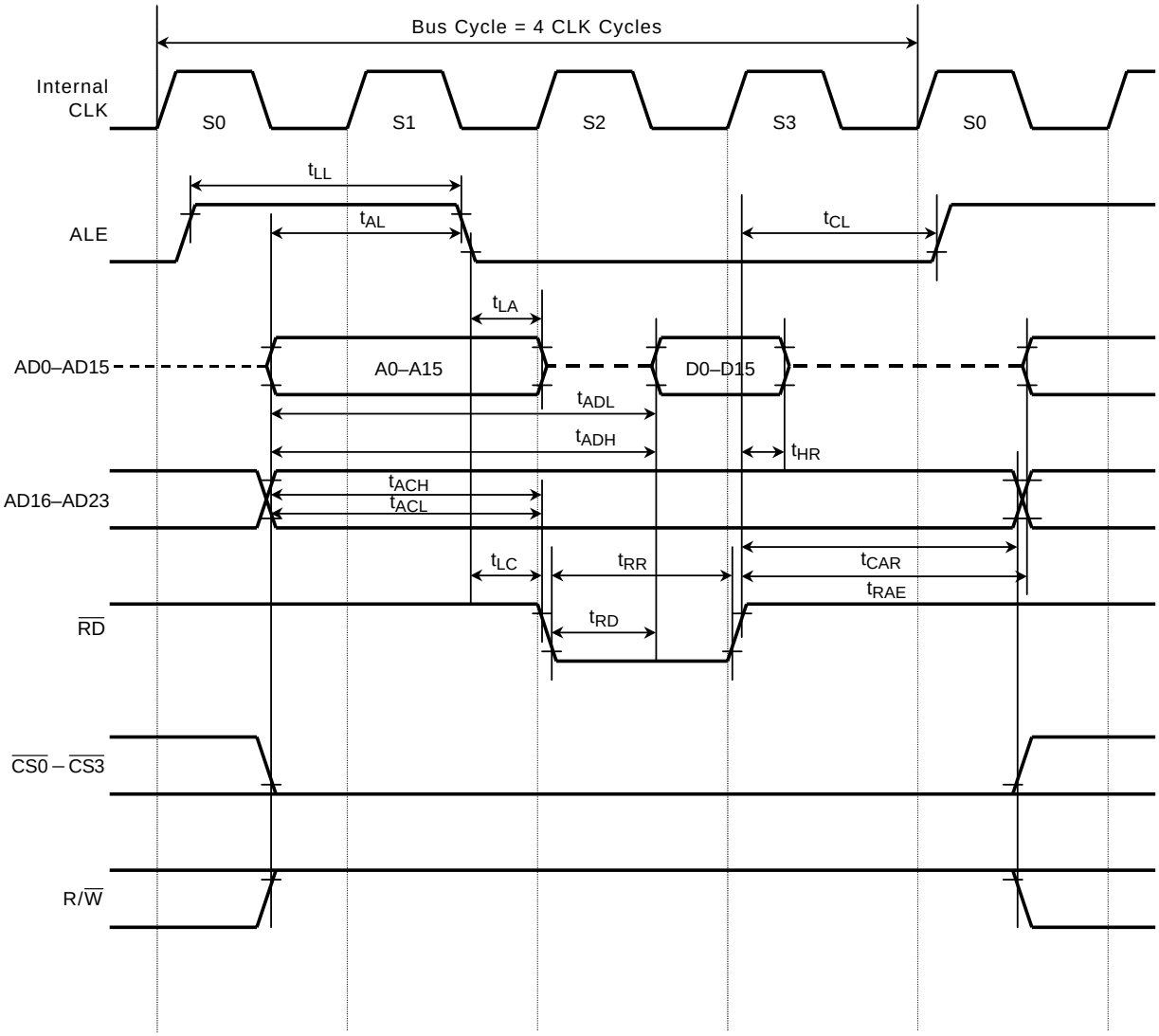
* W = 0

W: Number of wait-state cycles inserted (0 to 7 for programmed wait insertion)

N: Value of N for (1 + N) wait insertion

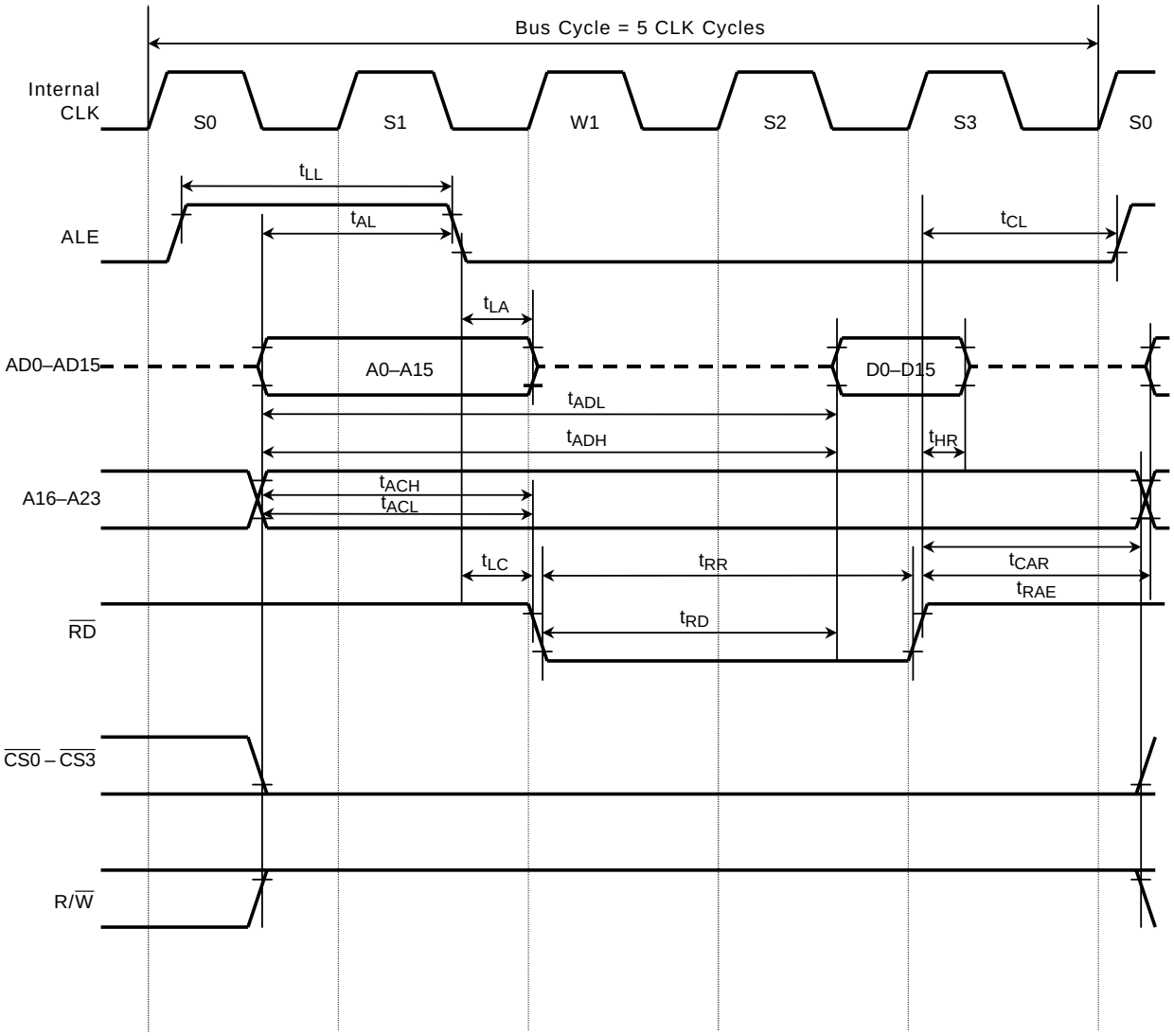
AC measurement conditions:

- Output levels: High = 2.4 V, Low = 0.45 V, CL = 30 pF
- Input levels: High = 2 V, Low = 0.6 V



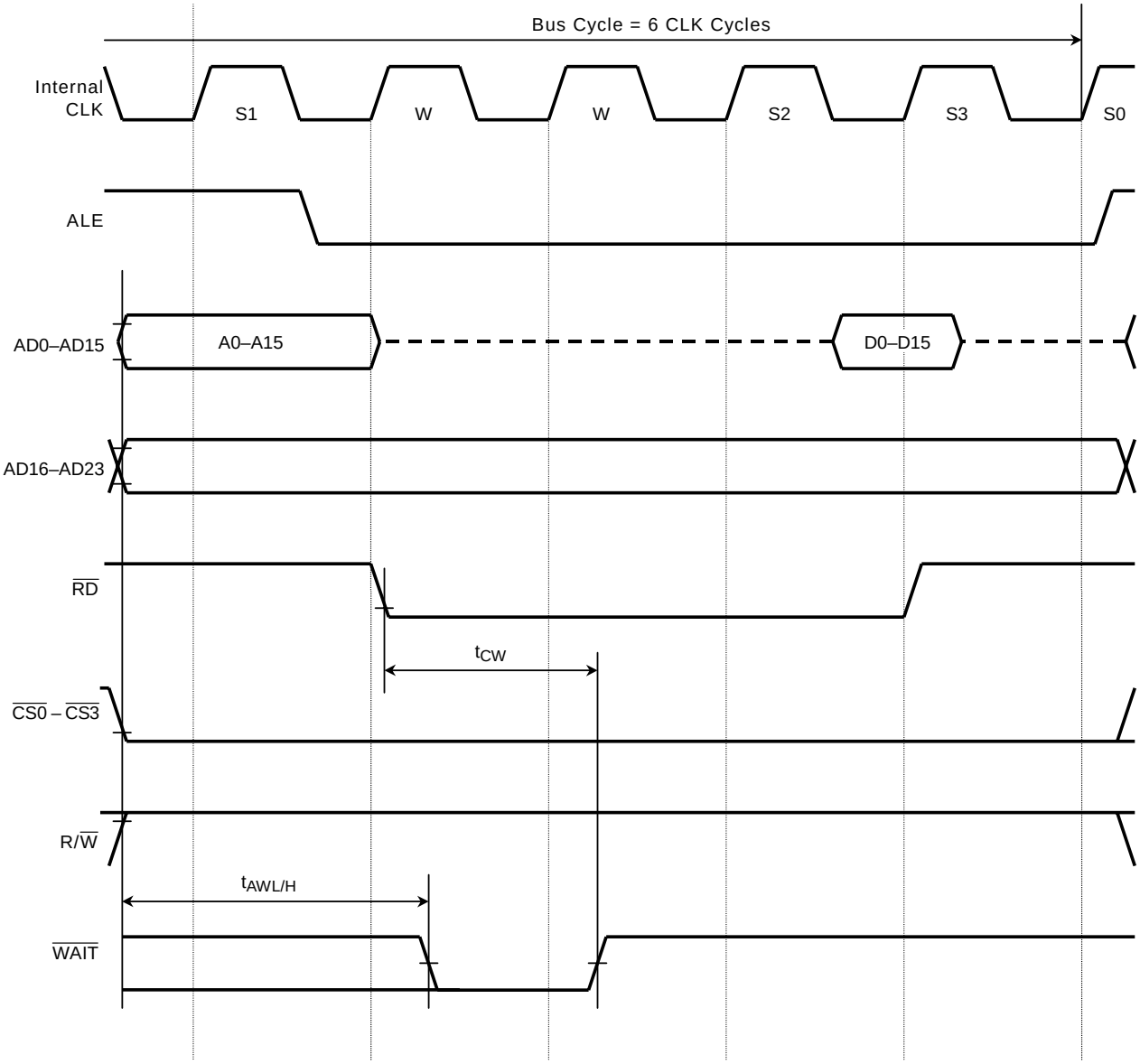
Note: The internal CLK is not the system clock driven out from the SCOUT pin.

Figure 18.1 Read Cycle Timing (ALE = 1.5, Zero Wait State)



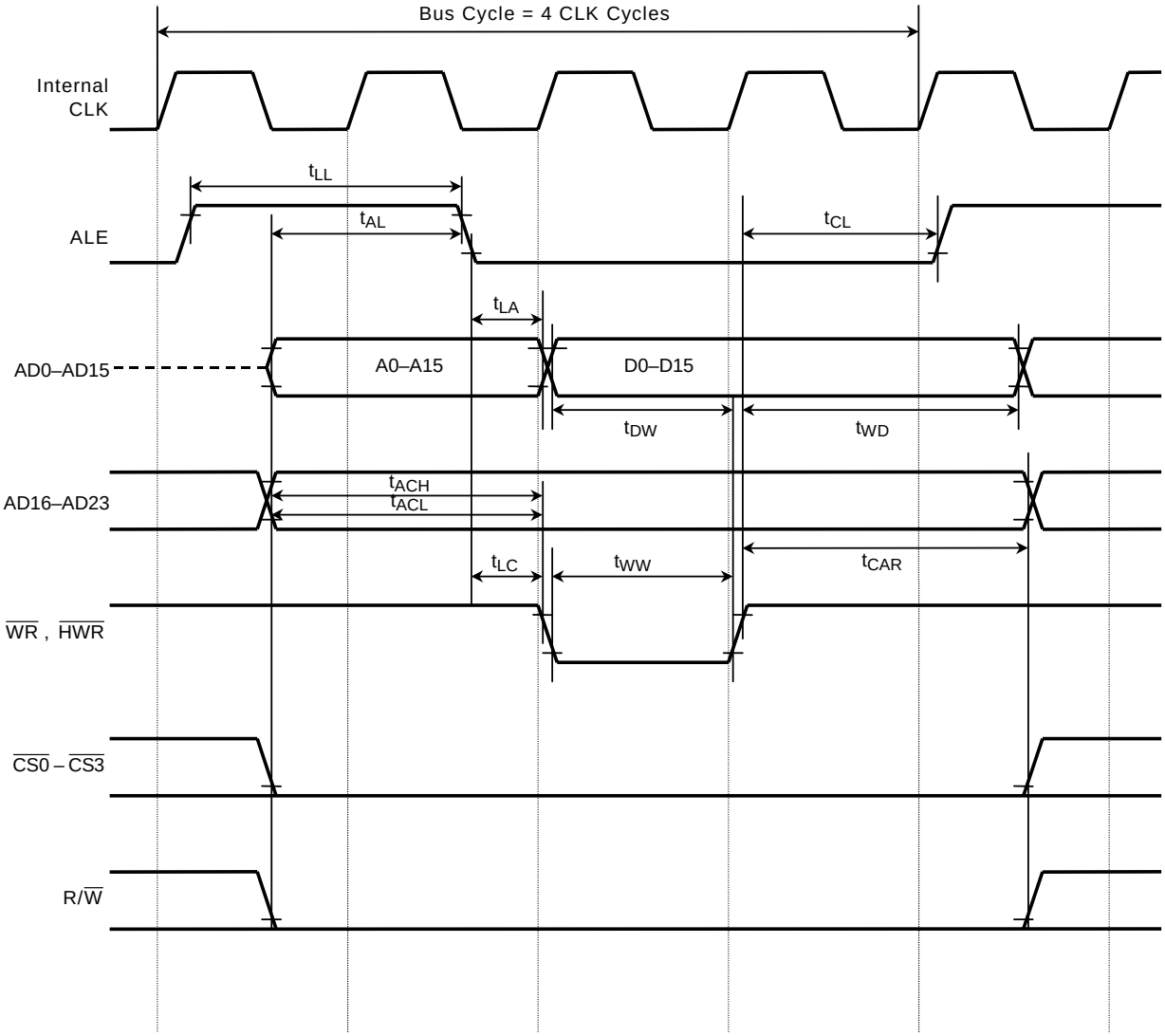
Note: The internal CLK is not the system clock driven out from the SCOUT pin.

Figure 18.2 Read Cycle Timing (ALE = 1.5, 1 Programmed Wait State)



Note1:	**If t_{AWH} and/or t_{AWL} cannot be satisified, a bus cycle must be initiated with the $\overline{WAIT}$ pin asserted.**
Note2:	**The internal CLK is not the system clock driven out from the SCOUT pin.**

Figure 18.3 Read Cycle Timing (ALE = 1.5, 2 Externally Generated Wait States with N=1)



Note: The internal CLK is not the system clock driven out from the SCOUT pin.

Figure 18.4 Write Cycle Timing (ALE = 1, Zero Wait State)

18.5 ADC Electrical Characteristics

$AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$

Parameter		Symbol	Condition	Min	Typ	Max	Unit
Analog reference voltage (+)		VREFH	V_{CC} = 3.3 ± 0.3 V	V_{CC} – 0.2 V	V_{CC}	V_{CC}	V
Analog reference voltage (–)		VREFL	V_{CC} = 3.3 ± 0.3 V	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
Analog input voltage		VAIN		VREFL		VREFH	
Analog supply current	ADMOD1.VREFON = 1	IREF (VREFL = VSS) (VREFH = VCC)	V_{CC} = 3.3V ± 0.3 V		0.8	1.2	mA
	ADMOD1.VREFON = 0		V_{CC} = 2.7 to 3.6 V		0.02	5.0	μA
Total error (not including quantization error)		—	V_{CC} = 3.3 V ± 0.3 V		± 1	± 3	LSB

Note 1: 1 LSB = (VREFH – VREFL) / 1024 (V)

Note 2: The A/D converter must be stopped when operating the TMP1940CYAF with the low-speed clock (fs).

Note 3: The supply current flowing through the AVCC pin is included in the digital supply current parameter (ICC).

18.6 SIO Timing

18.6.1 I/O Interface Mode

In the tables below, the letter x represents the fsys cycle period, which varies, depending on the programming of the clock gear function.

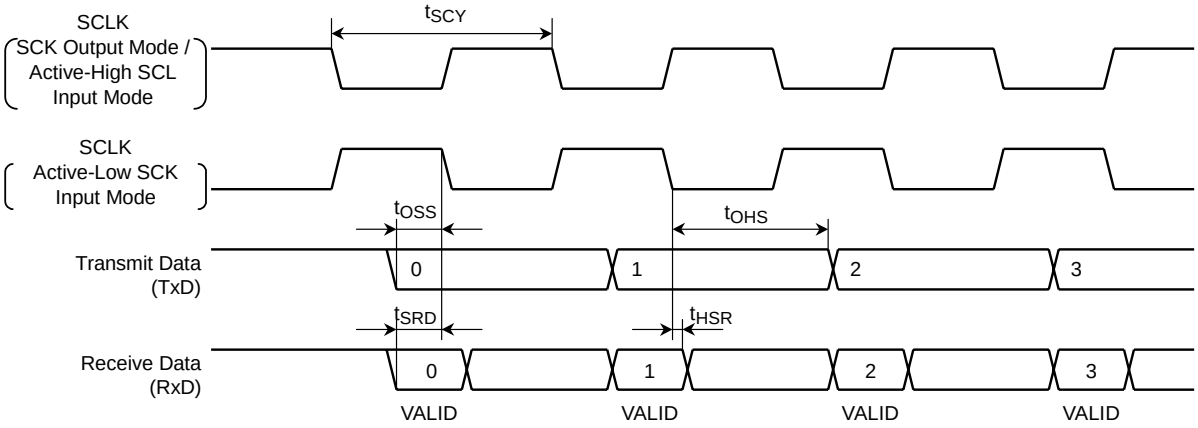
(1) SCLK Input Mode

Parameter	Symbol	Equation		20 MHz		32 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period	t_{SCY}	16x		800		500		ns
TxD data to SCLK rise or fall*	t_{OSS}	$(t_{SCY}/2) - 5x - 23$		127		71		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	$(t_{SCY}/2) + 3x$		550		343		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	2x + 8		108		71		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	0		0		0		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

(2) SCLK Output Mode

Parameter	Symbol	Equation		20 MHz		32 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period (programmable)	t_{SCY}	16x		800		500		ns
TxD data to SCLK rise	t_{OSS}	(tSCY/2) – 15		385		235		ns
TxD data hold after SCLK rise	t_{OHS}	(tSCY/2) – 15		385		235		ns
RxD data valid to SCK rise	t_{SRD}	x + 23		73		54		ns
RxD data hold after SCK rise	t_{HSR}	0		0		0		ns



18.7 SBI Timing

18.7.1 I^2C Mode

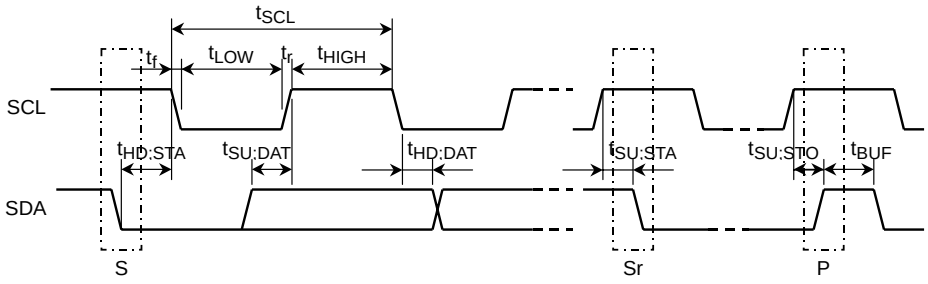
In the table below, the letters x and T represent the fsys and ϕT0 cycle periods, respectively. The letter n denotes the value of n programmed into the SCK[2:0] (SCL output frequency select) field in the SBI0CR1.

Parameter		Symbol	Equation		Standard Mode fsys = 8 MHz, n = 4		Fast Mode fsys = 32 MHz, n = 4		Unit
			Min	Max	Min	Max	Min	Max	
SCL clock frequency		t_{SC}	0		0	100	0	400	kHz
Hold time for START condition		$t_{HD:STA}$			4.0		0.6		μs
Low period of the SCL clock	Input	t_{LOW}			4.7		1.3		μs
	Output		$2^{(n-1)}$ T		4 (Note 1)		1 (Note 1)		μs
SCL clock high width	Input	t_{HIGH}			4.0		0.6		μs
	Output		$(2^{(n-1)} + 4)$ T		6		1.5		μs
Setup time for a repeated START condition		$t_{SU;STA}$	Software-dependent		4.7		0.6		μs
Data hold time		$t_{HD;DAT}$			0		0		μs
Data setup time		$t_{SU;DAT}$			250		100		ns
Setup time for STOP condition		$t_{SU;STO}$			4.0		0.6		μs
Bus free time between STOP and START conditions		t_{BUF}	Software-dependent		4.7		1.3		μs

Note 1: Different from the Philips I^2C-bus specification.

Note 2: The ouptut data hold time is equal to 12x.

Note 3: The Philips I^2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the fall edge of SCL. However, the TMP1940CYAF SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.



S: START condition
Sr: Repeated START condition
P: STOP condition

Note 4: To operate the SBI in I^2C Fast mode, the fsys frequency must be no less than 20 MHz. To operate the SBI in I^2C Standard mode, the fsys frequency must be no less than 4 MHz.

Note 5: Although *THE I^2C BUS SPECIFICATION* from Philips states that I/O pins of Fast-mode devices must not obstruct the SDA and SCL lines if V_{DD} is switched off, the TMP1940CYAF does not comply with this requirement.

18.7.2 Clock-Synchronous 8-Bit SIO Mode

In the tables below, the letters x and T represent the fsys and ϕT0 cycle periods, respectively. The letter n denotes the value of n programmed into the SCK[2:0] (SCL output frequency select) field in the SBI0CR1.

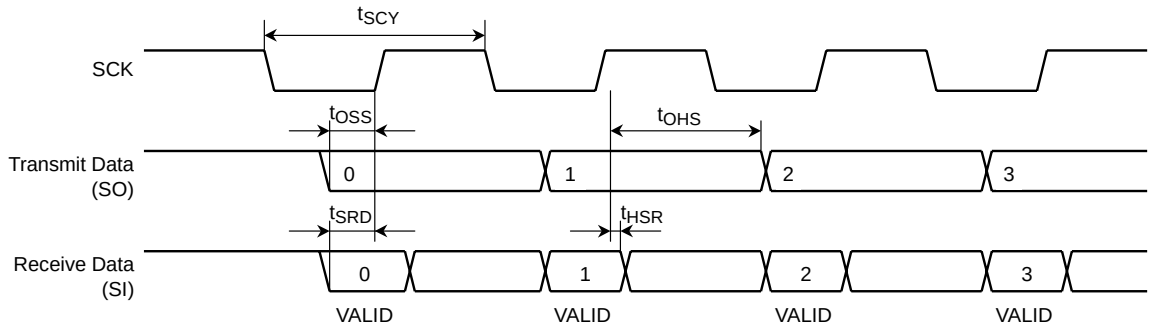
The electrical specifications below are for an SCK signal with a 50% duty cycle.

(1) SCK Input Mode

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
SCK period	t_{SCY}	16x		500		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - (6x + 30)$		32		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) + 4x$		375		ns
SI data valid to SCK rise	t_{SRD}	0		0		ns
SI data hold after SCK rise	t_{HSR}	4x + 10		135		ns

(2) SCK Output Mode

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
SCK period (programmable)	t_{SCY}	$2^n \times T$		1000		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - 20$		480		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) - 20$		480		ns
SI data valid to SCK rise	t_{SRD}	2x + 30		93		ns
SI data hold after SCK rise	t_{HSR}	0		0		ns



18.8 Event Counters (TA0IN, TA2IN, TB0IN0, TB0IN1, TB2IN0)

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100		163		ns
Clock high pulse width	t_{VCKH}	2x + 100		163		ns

18.9 Timer Capture (TB0IN0, TB0IN1, TB1IN0, TB1IN1, TB2IN0, TB2IN1)

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100		163		ns
High pulse width	t_{CPH}	2x + 100		163		ns

18.10 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0–INTA	t_{INTAL}	x + 100		132		ns
High pulse width for INT0–INTA	t_{INTAH}	x + 100		132		ns

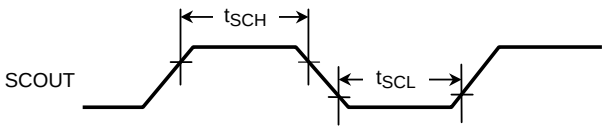
18.11 $\overline{\text{NMI}}$ and STOP/SLEEP Wake-up Interrupts

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for $\overline{\text{NMI}}$ and INT0–INT4	t_{INTBL}	100		100		ns
High pulse width for INT0–INT4	t_{INTBH}	100		100		ns

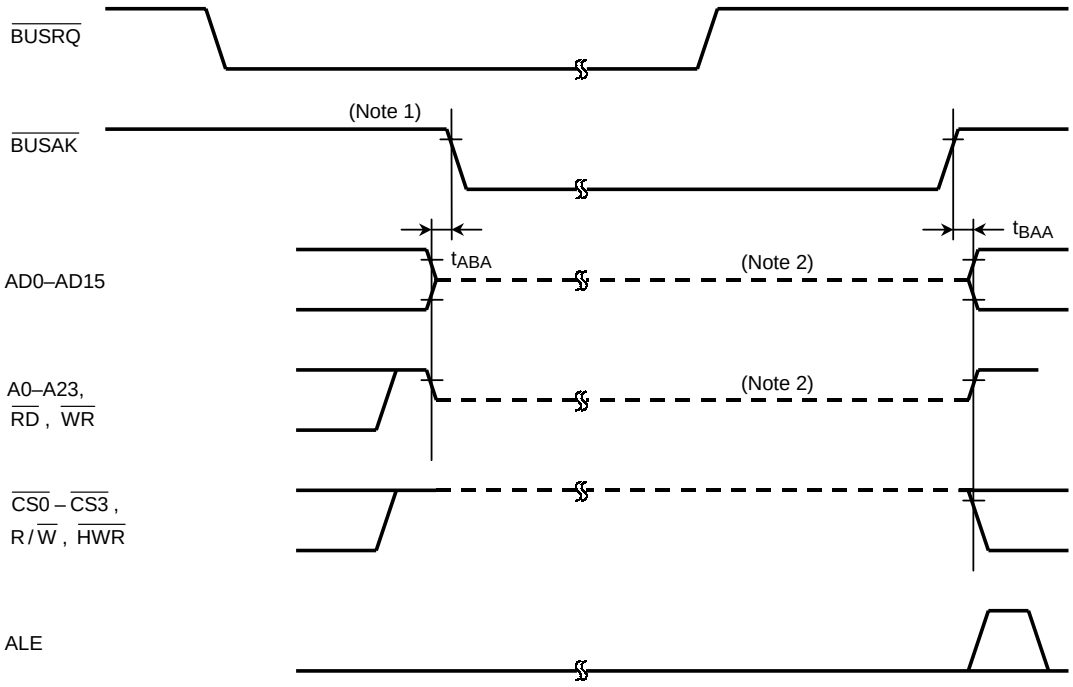
18.12 SCOUT Pin

In the table below, the letter T represents the cycle period of the SCOUT output clock.

Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{SCH}	0.5T – 5		10.6		ns
Clock high pulse width	t_{SCL}	0.5T – 5		10.6		ns



18.13 Bus Request and Bus Acknowledge Signals



Parameter	Symbol	Equation		32 MHz		Unit
		Min	Max	Min	Max	
Bus float to $\overline{\text{BUSAK}}$ asserted	t_{ABA}	0	80	0	80	ns
Bus float after $\overline{\text{BUSAK}}$ negated	t_{BAA}	0	80	0	80	ns

Note 1: **If the current bus cycle has not terminated due to wait-state insertion, the TMP1940CYAF does not respond to $\overline{\text{BUSRQ}}$ until the wait state ends.**

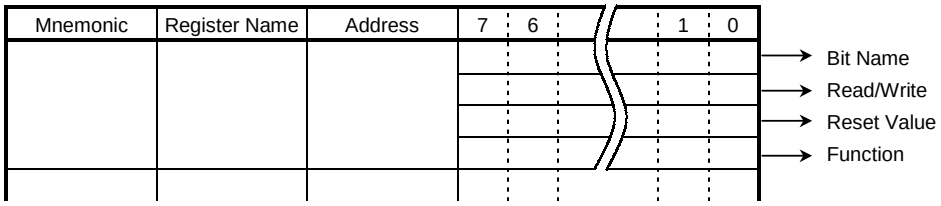
Note 2: **This broken lines indicate that output buffers are disabled, not that the signals are at indeterminate states. The pin holds the last logic value present at that pin before the bus is relinquished. This is dynamically accomplished through external load capacitances. The equipment manufacturer may maintain the bus at a predefined state by means of off-chip resistors, but he or she should design, considering the time (determined by the CR constant) it takes for a signal to reach a desired state. The on-chip, integrated programmable pullup/pulldown resistors remain active, depending on internal signal states.**

19. I/O Register Summary

The internal I/O registers configure and access the I/O ports, and control on-chip functions. These registers occupy 8-kbyte addresses from 0xFFFF_E000 through 0xFFFF_FFFF.

1. I/O ports
2. Watchdog Timer (WDT)
3. Real-Time Clock (RTC)
4. 8-Bit Timers (TMRAs)
5. 16-Bit Timer/Event Counters (TMRBs)
6. Serial I/O (SIO0 and SIO1)
7. Serial Bus Interface (SBI)
8. Serial I/O (SIO3 and SIO4)
9. A/D Converter (ADC)
10. Interrupt Controller (INTC)
11. DMA Controller (DMAC)
12. Chip Select/Wait Controller
13. Clock Generator (CG)
14. Flash control/status (TMP1940FDBF only)

Table Organization



Access

R/W: Read/write. The user can read and write the register bit.

R: Read only.

W: Write only.

W*: The user can read and write the register bit, but a read always returns a value of 1.

1. I/O Ports

Address	Mnemonic
0xFFFF_F000	P0
1	P1
2	P0CR
3	
4	P1CR
5	P1FC
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_F010	
1	
2	P2
3	
4	P2CR
5	P2FC
6	
7	
8	P3
9	
A	P3CR
B	P3FC
C	
D	
E	P4
F	

Address	Mnemonic
0xFFFF_F020	P4CR
1	P4FC
2	
3	
4	
5	P5
6	
7	
8	
9	
A	
B	P7
C	
D	
E	P7CR
F	P7FC

Address	Mnemonic
0xFFFF_F030	P8
1	P9
2	P8CR
3	P8FC
4	P9CR
5	P9FC
6	PA
7	
8	PACR
9	PAFC
A	
B	
C	
D	
E	
F	

2. WDT

Address	Mnemonic
0xFFFF_F050	ODE
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_F090	WDMOD
1	WDCR
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

3. RTC

Address	Mnemonic
0xFFFF_F0A0	RTCCR
1	
2	
3	
4	RTCREG
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

4. 8-Bit Timers

Address	Mnemonic
0xFFFF_F100	TA01RUN
1	
2	TA0REG
3	TA1REG
4	TA01MOD
5	TA1FFCR
6	
7	
8	TA23RUN
9	
A	TA2REG
B	TA3REG
C	TA23MOD
D	TA3FFCR
E	
F	

5. 16-Bit Timer/Event Counters

Address	Mnemonic
0xFFFF_F180	TB0RUN
1	
2	TB0MOD
3	TB0FFCR
4	
5	
6	
7	
8	TB0RG0L
9	TB0RG0H
A	TB0RG1L
B	TB0RG1H
C	TB0CP0L
D	TB0CP0H
E	TB0CP1L
F	TB0CP1H

Address	Mnemonic
0xFFFF_F190	TB1RUN
1	
2	TB1MOD
3	TB1FFCR
4	
5	
6	
7	
8	TB1RG0L
9	TB1RG0H
A	TB1RG1L
B	TB1RG1H
C	TB1CP0L
D	TB1CP0H
E	TB1CP1L
F	TB1CP1H

Address	Mnemonic
0xFFFF_F1A0	TB2RUN
1	
2	TB2MOD
3	TB2FFCR
4	
5	
6	
7	
8	TB2RG0L
9	TB2RG0H
A	TB2RG1L
B	TB2RG1H
C	TB2CP0L
D	TB2CP0H
E	TB2CP1L
F	TB2CP1H

Address	Mnemonic
0xFFFF_F1B0	TB3RUN
1	
2	TB3MOD
3	TB3FFCR
4	
5	
6	
7	
8	TB3RG0L
9	TB3RG0H
A	TB3RG1L
B	TB3RG1H
C	TB3CP0L
D	TB3CP0H
E	TB3CP1L
F	TB3CP1H

Figure 19.1 I/O Register Address Map (1/5)

6. SIO0 and SIO1

Address	Mnemonic
0xFFFF_F200	SC0BUF
1	SC0CR
2	SC0MOD0
3	BR0CR
4	BR0ADD
5	SC0MOD1
6	
7	
8	SC1BUF
9	SC1CR
A	SC1MOD0
B	BR1CR
C	BR1ADD
D	SC1MOD1
E	
F	

7. SBI

Address	Mnemonic
0xFFFF_F240	SBI0CR1
1	SBI0DBR
2	I2C0AR
3	SBI0CR2/SR
4	SBI0BR0
5	SBI0BR1
6	
7	

8. SIO3 and SIO4

Address	Mnemonic
0xFFFF_F280	SC3BUF
1	SC3CR
2	SC3MOD0
3	BR3CR
4	BR3ADD
5	SC3MOD1
6	
7	
8	SC4BUF
9	SC4CR
A	SC4MOD0
B	BR4CR
C	BR4ADD
D	SC4MOD1
E	
F	

9. ADC

Address	Mnemonic
0xFFFF_F300	ADREG04L
1	ADREG04H
2	ADREG15L
3	ADREG15H
4	ADREG26L
5	ADREG26H
6	ADREG37L
7	ADREG37H
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_F310	ADMOD0
1	ADMOD1
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Figure 19.1 I/O Register Address Map (2/5)

10. INTC

Address	Mnemonic
0xFFFF_E000	IMC0L
1	
2	IMC0H
3	
4	IMC1L
5	
6	
7	
8	
9	
A	IMC2H
B	
C	IMC3L
D	
E	IMC3H
F	

Address	Mnemonic
0xFFFF_E010	
1	
2	
3	
4	IMC5L
5	
6	IMC5H
7	
8	
9	
A	
B	
C	IMC7L
D	
E	IMC7H
F	

Address	Mnemonic
0xFFFF_E020	IMC8L
1	
2	IMC8H
3	
4	
5	
6	
7	
8	IMCAL
9	
A	IMCAH
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E030	IMCCL
1	
2	IMCCH
3	
4	IMCDL
5	
6	IMCDH
7	
8	IMCEL
9	
A	IMCEH
B	
C	IMCFL
D	
E	IMCFH
F	

Address	Mnemonic
0xFFFF_E040	IVR
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E050	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E060	INTCLR
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E070	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Note: **Any attempt to access an address in the shaded areas causes a bus error to be signaled to the TX19 core processor. Any attempt to access an address in the ranges 0xFFFF_E080 through 0xFFFF_E0FF and 0xFFFF_E10C through 0xFFFF_E1FF also causes a bus error.**

Figure 19.1 I/O Register Address Map (3/5)

11. DMAC

Address	Mnemonic
0xFFFF_E200	CCR0
1	
2	
3	
4	CSR0
5	
6	
7	
8	SAR0
9	
A	
B	
C	DAR0
D	
E	
F	

Address	Mnemonic
0xFFFF_E210	BCR0
1	
2	
3	
4	
5	
6	
7	
8	DTCR0
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E220	CCR1
1	
2	
3	
4	CSR1
5	
6	
7	
8	SAR1
9	
A	
B	
C	DAR1
D	
E	
F	

Address	Mnemonic
0xFFFF_E230	BCR1
1	
2	
3	
4	
5	
6	
7	
8	DTCR1
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E240	CCR2
1	
2	
3	
4	CSR2
5	
6	
7	
8	SAR2
9	
A	
B	
C	DAR2
D	
E	
F	

Address	Mnemonic
0xFFFF_E250	BCR2
1	
2	
3	
4	
5	
6	
7	
8	DTCR2
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E260	CCR3
1	
2	
3	
4	CSR3
5	
6	
7	
8	SAR3
9	
A	
B	
C	DAR3
D	
E	
F	

Address	Mnemonic
0xFFFF_E270	BCR3
1	
2	
3	
4	
5	
6	
7	
8	DTCR3
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E280	DCR
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	DHR
D	
E	
F	

Address	Mnemonic
0xFFFF_E290	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E2A0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E2B0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Note: **Any attempt to access an address in the shaded areas causes a bus error to be signaled to the TX19 core processor. Any attempt to access an address in the range 0xFFFF_E2C0 through 0xFFFF_E2FF also causes a bus error. Any attempt to access an address in the range 0xFFFF_E300 through 0xFFFF_E3FF is disallowed.**

Figure 19.1 I/O Register Address Map (4/5)

12. CS/Wait Controller

Address	Mnemonic
0xFFFF_E400	BMA0
1	
2	
3	
4	BMA1
5	
6	
7	
8	BMA2
9	
A	
B	
C	BMA3
D	
E	
F	

Address	Mnemonic
0xFFFF_E410	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E480	B01CS
1	
2	
3	
4	B23CS
5	
6	
7	
8	BEXCS
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E490	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

13. CG

Address	Mnemonic
0xFFFF_EE00	SYSCR0
1	SYSCR1
2	SYSCR2
3	SYSCR3
4	ADCCLK
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_EE10	IMCGA0
1	IMCGA1
2	IMCGA2
3	IMCGA3
4	IMCGB0
5	
6	
7	IMCGB3
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_EE20	EICRCG
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

14. Flash Control/Status

Address	Mnemonic
0xFFFF_E510	SEQMOD
1	
2	
3	
4	SEQCNT
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Address	Mnemonic
0xFFFF_E520	FLCS
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

Note: **Any attempt to access an address in the shaded areas causes a bus error to be signaled to the TX19 core processor. Any attempt to access an address in the following ranges also cause a bus error.**
0xFFFF_E420 thru 0xFFFF_E47F
0xFFFF_E450 thru 0xFFFF_E4FF
0xFFFF_E700 thru 0xFFFF_EDFF
0xFFFF_EE30 thru 0xFFFF_EEFF

An attempt to access an address in the following ranges also cause a bus error.
0xFFFF_F040 thru 0xFFFF_F04F
0xFFFF_F060 thru 0xFFFF_F08F
0xFFFF_F0B0 thru 0xFFFF_F0FF
0xFFFF_F110 thru 0xFFFF_F17F
0xFFFF_F1C0 thru 0xFFFF_F1FF
0xFFFF_F210 thru 0xFFFF_F23F
0xFFFF_F248 thru 0xFFFF_F27F
0xFFFF_F290 thru 0xFFFF_F2FF
0xFFFF_F320 thru 0xFFFF_FFFF

Figure 19.1 I/O Register Address Map (5/5)

19.1 I/O Ports

I/O Port Data Registers

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0 Register	FFFF F000H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Undefined							
			Input mode							
P1	Port 1 Register	FFFF F001H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode							
P2	Port 2 Register	FFFF F012H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P3	Port 3 Register	FFFF F018H	P37	P36	P35	P34	P33	P32	P31	P30
			R/W							
			1	1	1	1	1	1	1	1
			Input mode						Output mode	
P4	Port 4 Register	FFFF F01EH	—	—	—	P44	P43	P42	P41	P40
			—	—	—	R/W				
			—	—	—	1	1	1	1	1
			—	—	—	Input mode				
P5	Port 5 Register	FFFF F025H	P57	P56	P55	P54	P53	P52	P51	P50
			R							
			Input mode							
P7	Port 7 Register	FFFF F02BH	P77	P76	P75	P74	P73	P72	P71	P70
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P8	Port 8 Register	FFFF F030H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P9	Port 9 Register	FFFF F031H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W							
			1	1	1	1	1	1	1	1
			Output mode			Input mode				
PA	Port A Register	FFFF F036H	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							

I/O Port Control and Function Registers (1 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
P0CR	Port 0 Control Register	FFFF F002H	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: IN, 1: OUT							
P1CR	Port 1 Control Register	FFFF F004H	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			Refer to P1FC.							
P1FC	Port 1 Function Register	FFFF F005H	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00: Input port, 01: Output port, 10: AD15–AD8, 11: A15–A8							
P2CR	Port 2 Control Register	FFFF F014H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			Refer to P2FC.							
P2FC	Port 2 Function Register	FFFF F015H	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00: Input port, 01: Output port, 10: A0–A7, 11: A23–A16							
P3CR	Port 3 Control Register	FFFF F01AH	P37C	P36C	P35C	P34C	P33C	P32C	—	—
			W							
			0	0	0	0	0	0		
			0: IN, 1: OUT							
P3FC	Port 3 Function Register	FFFF F01BH	—	P36F	P35F	P34F	—	P32F	P31F	P30F
				W				W		
				0	0	0		0	0	0
				0: Port 1: R/ $\overline{W}$ output	0: Port 1: $\overline{BUSAK}$ output	0: Port 1: $\overline{BUSRQ}$ input		0: Port 1: $\overline{HWR}$ output	0: Port 1: $\overline{WR}$ output	0: Port 1: $\overline{RD}$ output
P4CR	Port 4 Control Register	FFFF F020H	—	—	—	P44C	P43C	P42C	P41C	P40C
						W				
						0	0	0	0	0
						0: IN, 1: OUT				
P4FC	Port 4 Function Register	FFFF F021H	—	—	—	P44F	P43F	P42F	P41F	P40F
						W				
						0	0	0	0	0
						0: Port 1: SCOUT output	0: Port 1: $\overline{CS3}$ output	0: Port 1: $\overline{CS2}$ output	0: Port 1: $\overline{CS1}$ output	0: Port 1: $\overline{CS0}$ output
P7CR	Port 7 Control Register	FFFF F02EH	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P7FC	Port 7 Function Register	FFFF F02FH	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
			0	0	0	0	0	0	0	0
			0: Port 1: Wake-up INT0 input	0: Port 1: TB0OUT output	0: Port 1: TB0IN1 input	0: Port 1: TB0IN0 input	0: Port 1: TA3OUT output / RXD4 input	0: Port 1: TA2IN input / TXD4 output	0: Port 1: TA1OUT output / RXD3 input	0: Port 1: TA0IN input / TXD3 output

Note: P77F must be set to 1 when INT0 is used to exit STOP mode with SYSCR2.DRVE cleared.

I/O Port Control and Function Registers (2 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
P8CR Port 8	Port 8 Control Register	FFFF F032H	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0: IN, 1: OUT							
P8FC Port 8	Port 8 Function Register	FFFF F033H	—	P86F	P85F	P84F	P83F	P82F	P81F	P80F
			W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	0: Port 1: TB3OUT output	0: Port 1: TB2OUT output	0: Port 1: TB2IN1 input	0: Port 1: TB2IN0 input	0: Port 1: TB1OUT output	0: Port 1: TB1IN1 input	0: Port 1: TB1IN0 input
P9CR Port 9	Port 9 Control Register	FFFF F034H	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W							
			0	0	0	0	0	0	0	0
			0: IN, 1: OUT							
P9FC Port 9	Port 9 Function Register	FFFF F035H	—	—	P95F	—	P93F	P92F	—	P90F
					W		W			W
					0		0	0		0
					0: Port 1: SCLK1 output or $\overline{CTS1}$ / SCLK1 input		0: Port 1: TXD1 output	0: Port 1: SCLK0 output or $\overline{CTS0}$ / SCLK0 input		0: Port 1: TXD0 output
PACR Port A	Port A Control Register	FFFF F038H	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0	0	0	0	0	0	0	0
			0: IN, 1: OUT							
PAFC Port A	Port A Function Register	FFFF F039H	PA7F	PA6F	PA5F	—	PA3F	PA2F	PA1F	PA0F
			W							
			0	0	0	0	0	0	0	0
			0: Port 1: SCL output	0: Port 1: SDA/SO output	0: Port 1: SCK output	Must be written as 0.	0: Port 1: Wake-up INT4 input	0: Port 1: Wake-up INT3 input	0: Port 1: Wake-up INT2 input	0: Port 1: Wake-up INT1 input

Note: PA0F–PA3F must be set to 1 when INT1–INT4 are used to exit STOP mode with SYSCR2.DRVE cleared.

Open-Drain Enable Register

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
ODE	Open-Drain Enable Register	FFFF F050H	—	—	ODE72	ODE70	ODEA7	ODEA6	ODE93	ODE90
			—	—	R/W					
			—	—	0	0	0	0	0	0
					P72 0: Push-pull 1: Open-drain	P70 0: Push-pull 1: Open-drain	PA7 0: Push-pull 1: Open-drain	PA6 0: Push-pull 1: Open-drain	P93 0: Push-pull 1: Open-drain	P90 0: Push-pull 1: Open-drain

19.2 Interrupt Controller

Interrupt Controller (1 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC0L	Interrupt Mode Control Register 0L	FFFF E000H	—	—	EIM01	EIM00	DM0	IL02	IL01	IL00
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level Must be written as 00.		DMA trigger 0: Disable 1: Enable	When DM0 = 0 Interrupt Number 0 (Software Set) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM0 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM11	EIM10	DM1	IL12	IL11	IL10
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM1 = 0 Interrupt Number 1 (INT0 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM1 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC0H	Interrupt Mode Control Register 0H	FFFF E002H	23	22	21	20	19	18	17	16
			—	—	EIM21	EIM20	DM2	IL22	IL21	IL20
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM2 = 0 Interrupt Number 2 (INT1 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM2 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM31	EIM30	DM3	IL32	IL31	IL30
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM3 = 0 Interrupt Number 3 (INT2 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Note1: When using INT0–INT4 to exit STOP and SLEEP modes, program their sensitivity in the IMCGA0 to IMCGA3 and IMCGB0 located within the CG. In this case, the EIMx[1:0] fields in the IMC0L, IMC0H and IMC1L have no effect, but must always be set to "high-level" (i.e., 01).

Note 2: Interrupt sensitivity must be programmed before interrupt priority levels are programmed into the ILx[2:0] field.

Interrupt Controller (2 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC1L	Interrupt Mode Control Register 1L	FFFF E004H	—	—	EIM41	EIM40	DM4	IL42	IL41	IL40
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM4 = 0 Interrupt Number 4 (INT3 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM4 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM51	EIM50	DM5	IL52	IL51	IL50
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DM5 = 0 Interrupt Number 5 (INT4 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM5 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC2H	Interrupt Mode Control Register 2H	FFFF E00AH	23	22	21	20	19	18	17	16
			—	—	EIMA1	EIMA0	DMA	ILA2	ILA1	ILA0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DMA = 0 Interrupt Number 10 (INT5 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DMA = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIMB1	EIMB0	DMB	ILB2	ILB1	ILB0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DMB = 0 Interrupt Number 11 (INT6 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DMB = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (3 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC3L	Interrupt Mode Control Register 3L	FFFF E00CH	—	—	EIMC1	EIMC0	DMC	ILC2	ILC1	ILC0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DMC = 0 Interrupt Number 12 (INT7 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DMC = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIMD1	EIMD0	DMD	ILD2	ILD1	ILD0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DMD = 0 Interrupt Number 13 (INT8 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DMD = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC3H	Interrupt Mode Control Register 3H	FFFF E00EH	23	22	21	20	19	18	17	16
			—	—	EIME1	EIME0	DME	ILE2	ILE1	ILE0
			—	—	R/W					
			—	—	1	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DME = 0 Interrupt Number 14 (INT9 pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DME = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIMF1	EIMF0	DMF	ILF2	ILF1	ILF0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Interrupt sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge		DMA trigger 0: Disable 1: Enable	When DMF = 0 Interrupt Number 15 (INTA pin) 000: Interrupt disabled. 001–111: Priority level (1–7) When DMF = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (4 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC5L	Interrupt Mode Control Register 5L	FFFF E014H	—	—	EIM141	EIM140	DM14	IL142	IL141	IL140
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM14 = 0 Interrupt Number 20 (INTTA0) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM14 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM151	EIM150	DM15	IL152	IL151	IL150
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM15 = 0 Interrupt Number 21 (INTTA1) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM15 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC5H	Interrupt Mode Control Register 5H	FFFF E016H	23	22	21	20	19	18	17	16
			—	—	EIM161	EIM160	DM16	IL162	IL161	IL160
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM16 = 0 Interrupt Number 22 (INTTA2) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM16 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM171	EIM170	DM17	IL172	IL171	IL170
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM17 = 0 Interrupt Number 23 (INTTA3) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM17 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (5 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC7L	Interrupt Mode Control Register 7L	FFFF E01CH	—	—	EIM1C1	EIM1C0	DM1C	IL1C2	IL1C1	IL1C0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM1C = 0 Interrupt Number 28 (INTTB00) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM1C = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM1D1	EIM1D0	DM1D	IL1D2	IL1D1	IL1D0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM1D = 0 Interrupt Number 29 (INTTB01) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM1D = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC7H	Interrupt Mode Control Register 7H	FFFF E01EH	23	22	21	20	19	18	17	16
			—	—	EIM1E1	EIM1E0	DM1E	IL1E2	IL1E1	IL1E0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM1E = 0 Interrupt Number 30 (INTTB10) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM1E = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM1F1	EIM1F0	DM1F	IL1F2	IL1F1	IL1F0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM1F = 0 Interrupt Number 31 (INTTB11) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM1F = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (6 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMC8L	Interrupt Mode Control Register 8L	FFFF E020H	—	—	EIM201	EIM200	DM20	IL202	IL201	IL200
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM20 = 0 Interrupt Number 32 (INTTB20) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM20 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM211	EIM210	DM21	IL212	IL211	IL210
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM21 = 0 Interrupt Number 33 (INTTB21) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM21 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMC8H	Interrupt Mode Control Register 8H	FFFF E022H	23	22	21	20	19	18	17	16
			—	—	EIM221	EIM220	DM22	IL222	IL221	IL220
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM22 = 0 Interrupt Number 34 (INTTB30) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM22 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM231	EIM230	DM23	IL232	IL231	IL230
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM23 = 0 Interrupt Number 35 (INTTB31) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM23 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (7 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCAL	Interrupt Mode Control Register AL	FFFF E028H	—	—	EIM281	EIM280	DM28	IL282	IL281	IL280
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM28 = 0 Interrupt Number 40 (INTTBOF0) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM28 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM291	EIM290	DM29	IL292	IL291	IL290
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM29 = 0 Interrupt Number 41 (INTTBOF1) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM29 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMCAH	Interrupt Mode Control Register AH	FFFF E02AH	23	22	21	20	19	18	17	16
			—	—	EIM2A1	EIM2A0	DM2A	IL2A2	IL2A1	IL2A0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM2A = 0 Interrupt Number 42 (INTTBOF2) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM2A = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM2B1	EIM2B0	DM2B	IL2B2	IL2B1	IL2B0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM2B = 0 Interrupt Number 43 (INTTBOF3) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM2B = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (8 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCCL	Interrupt Mode Control Register CL	FFFF E030H	—	—	EIM301	EIM300	DM30	IL302	IL301	IL300
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM30 = 0 Interrupt Number 48 (INTRX0) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM30 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM311	EIM310	DM31	IL312	IL311	IL310
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM31= 0 Interrupt Number 49 (INTTX0) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM31= 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMCCH	Interrupt Mode Control Register CH	FFFF E032H	23	22	21	20	19	18	17	16
			—	—	EIM321	EIM320	DM32	IL322	IL321	IL320
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM32 = 0 Interrupt Number 50 (INTRX1) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM32 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM331	EIM330	DM33	IL332	IL331	IL330
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM33 = 0 Interrupt Number 51 (INTTX1) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM33= 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (9 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCDL	Interrupt Mode Control Register DL	FFFF E034H	—	—	EIM341	EIM340	DM34	IL342	IL341	IL340
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM34 = 0 Interrupt Number 52 (INTS2) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM34 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—						
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 00.		Must be written as 0.	Must be written as 000.		
IMCDH	Interrupt Mode Control Register DH	FFFF E036H	23	22	21	20	19	18	17	16
			—	—	EIM361	EIM360	DM36	IL362	IL361	IL360
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM36 = 0 Interrupt Number 54 (INTRX3) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM36 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM371	EIM370	DM37	IL372	IL371	IL370
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM37 = 0 Interrupt Number 55 (INTTX3) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM37 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (10 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCEL	Interrupt Mode Control Register EL	FFFF E038H	—	—	EIM381	EIM380	DM38	IL382	IL381	IL380
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM38 = 0 Interrupt Number 56 (INTRX4) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM38 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM391	EIM390	DM39	IL392	IL391	IL390
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM39 = 0 Interrupt Number 57 (INTTX4) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM39 = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMCEH	Interrupt Mode Control Register EH	FFFF E03AH	23	22	21	20	19	18	17	16
			—	—	EIM3A1	EIM3A0	DM3A	IL3A2	IL3A1	IL3A0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 01.		DMA trigger 0: Disable 1: Enable	When DM3A = 0 Interrupt Number 58 (INTRTC) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3A = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM3B1	EIM3B0	DM3B	IL3B2	IL3B1	IL3B0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 11.		DMA trigger 0: Disable 1: Enable	When DM3B = 0 Interrupt Number 59 (INTAD) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3B = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (11 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCFL	Interrupt Mode Control Register FL	FFFF E03CH	—	—	EIM3C1	EIM3C0	DM3C	IL3C2	IL3C1	IL3C0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 10.		DMA trigger 0: Disable 1: Enable	When DM3C = 0 Interrupt Number 60 (INTDMA0) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3C = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			15	14	13	12	11	10	9	8
			—	—	EIM3D1	EIM3D0	DM3D	IL3D2	IL3D1	IL3D0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 10.		DMA trigger 0: Disable 1: Enable	When DM3D = 0 Interrupt Number 61 (INTDMA1) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3D = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
IMCFH	Interrupt Mode Control Register FH	FFFF E03EH	23	22	21	20	19	18	17	16
			—	—	EIM3E1	EIM3E0	DM3E	IL3E2	IL3E1	IL3E0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 10.		DMA trigger 0: Disable 1: Enable	When DM3E = 0 Interrupt Number 62 (INTDMA2) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3E = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		
			31	30	29	28	27	26	25	24
			—	—	EIM3F1	EIM3F0	DM3F	IL3F2	IL3F1	IL3F0
			—	—	R/W					
			—	—	0	0	0	0	0	0
					Must be written as 10.		DMA trigger 0: Disable 1: Enable	When DM3F = 0 Interrupt Number 63 (INTDMA3) 000: Interrupt disabled. 001–111: Priority level (1–7) When DM3F = 1 DMAC channel select 000–011: Ch. number (0–3) 100–111: Don't use.		

Interrupt Controller (12 of 12)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IVR	Interrupt Vector Register	FFFF E040H	IVRL				—	—	—	—
			R							
			0	0	0	0	0	0	0	0
			Interrupt vector for the source of the current interrupt							
			15	14	13	12	11	10	9	8
			IVRH						IVRL	
			R/W						R	
			0	0	0	0	0	0	0	0
									Interrupt vector for the source of the current interrupt	
			23	22	21	20	19	18	15	16
			IVRH							
			R/W							
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			IVRH							
			R/W							
			0	0	0	0	0	0	0	0
INTCLR	Interrupt Request Clear Register	FFFF E060H	7	6	5	4	3	2	1	0
			—	—	EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
			—	—	W					
			—	—	—	—	—	—	—	—
			IVRL[9:4] value for an interrupt to be cleared							

19.3 Chip Select/Wait Controller

Chip Select/Wait Controller (1 of 4)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
BMA0	Base/ Mask Address Register	FFFF E400H	MA0							
			R/W							
			1	1	1	1	1	1	1	1
			Bits 9–0 specify the address bits (A23–A14) to be masked. 0: The corresponding address bit is not masked. 1: The corresponding address bit is masked.							
			15	14	13	12	11	10	9	8
			MA0							
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Address mask 0: Not masked 1: Masked	
			23	22	21	20	19	18	17	16
			BA0							
			R/W							
			0	0	0	0	0	0	0	0
			A23–A16 of the starting address for CS0							
			31	30	29	28	27	26	25	24
			BA0							
			R/W							
			0	0	0	0	0	0	0	0
			A31–A24 of the starting address for CS0							
BMA1	Base/ Mask Address Register	FFFF E404H	7	6	5	4	3	2	1	0
			MA1							
			R/W							
			1	1	1	1	1	1	1	1
			Bits 9–0 specify the address bits (A23–A14) to be masked. 0: The corresponding address bit is not masked. 1: The corresponding address bit is masked.							
			15	14	13	12	11	10	9	8
			MA1							
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Address mask 0: Not masked 1: Masked	
			23	22	21	20	19	18	17	16
			BA1							
			R/W							
			0	0	0	0	0	0	0	0
			A23–A16 of the starting address for CS1							
			31	30	29	28	27	26	25	24
			BA1							
			R/W							
			0	0	0	0	0	0	0	0
			A31–A24 of the starting address for CS1							

Chip Select/Wait Controller (2 of 4)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
BMA2	Base/ Mask Address Register	FFFF E408H	MA2							
			R/W							
			1	1	1	1	1	1	1	1
			Bits 8–0 specify the address bits (A23–A15) to be masked 0: The corresponding address bit is not masked. 1: The corresponding address bit is masked.							
			15	14	13	12	11	10	9	8
			MA2							
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Address mask 0: Not masked 1: Masked
			23	22	21	20	19	18	17	16
			BA2							
			R/W							
			0	0	0	0	0	0	0	0
			A23–A16 of the starting address for CS2							
			31	30	29	28	27	26	25	24
			BA2							
			R/W							
			0	0	0	0	0	0	0	0
			A31–A24 of the starting address for CS2							
BMA3	Base/ Mask Address Register	FFFF E40CH	7	6	5	4	3	2	1	0
			MA3							
			R/W							
			1	1	1	1	1	1	1	1
			Bits 8–0 specify the address bits (A23–A15) to be masked 0: The corresponding address bit is not masked. 1: The corresponding address bit is masked.							
			15	14	13	12	11	10	9	8
			MA3							
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Address mask 0: Not masked 1: Masked
			23	22	21	20	19	18	17	16
			BA3							
			R/W							
			0	0	0	0	0	0	0	0
			A23–A16 of the starting address for CS3							
			31	30	29	28	27	26	25	24
			BA3							
			R/W							
			0	0	0	0	0	0	0	0
			A31–A24 of the starting address for CS3							

Chip Select/Wait Controller (3 of 4)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
B01CS	Chip Select/ Wait Control Register	FFFF E480H	B0OM		—	B0BUS	B0W			
			W		—	W				
			0	0	—	0	0	1	0	1
			Chip select output waveform 00: ROM/SRAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
			15	14	13	12	11	10	9	8
			—	—	—	—	B0E	—	B0RCV	
			—	—	—	—	W	—	W	
			—	—	—	—	0	—	0	0
							CS0 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	
			23	22	21	20	19	18	17	16
			B1OM		—	B1BUS	B1W			
			W		—	W				
			0	0	—	0	0	1	0	1
			Chip select output waveform 00: ROM/SRAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
			31	30	29	28	27	26	25	24
			—	—	—	—	B1E	—	B1RCV	
			—	—	—	—	W	—	W	
			—	—	—	—	0	—	0	0
							CS1 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

Chip Select/Wait Controller (4 of 4)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
B23CS	Chip Select/ Wait Control Register	FFFF E484H	B2OM		—	B2BUS	B2W			
			W		—	W				
			0	0	—	0	0	1	0	1
			Chip select output waveform 00: ROM/SRAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
			15	14	13	12	11	10	9	8
			—	—	—	—	B2E	B2M	B2RCV	
			—	—	—	—	W	W	W	
			—	—	—	—	1	0	0	0
							CS2 enable 0: Disable 1: Enable	CS2 space select 0: Whole 4-Gbyte space 1: CS space	Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	
			23	22	21	20	19	18	17	16
			B3OM	—	—	B3BUS	B3W			
			W	—	—	W				
			0	0	—	0	0	1	0	1
			Chip select output waveform 00: ROM/SRAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Number of wait-state cycles 0000: No wait state, 0001: 1 wait state 0010: 2 wait states, 0011: 3 wait states 0100: 4 wait states, 0101: 5 wait states 0110: 6 wait states, 0111: 7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
			31	30	29	28	27	26	25	24
			—	—	—	—	B3E	—	B3RCV	
			—	—	—	—	W	—	W	
			—	—	—	—	0	—	0	0
							CS3 enable 0: Disable 1: Enable		Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	
BEXCS	Chip Select/ Wait Control Register	FFFF E488H	7	6	5	4	3	2	1	0
			BEXOM		—	BEXBUS	BEXW			
			W		—	W				
			0	0	—	0	0	1	0	1
			Chip select output waveform 00: ROM/SRAM Don't use any other value.			Data bus width 0: 16-bit 1: 8-bit	Sets the number of wait cycles 0000–0111: 0–7 wait states 1111: (1+N) wait states, as determined by the $\overline{WAIT}$ pin Don't use any other value.			
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	BEXRCV	
			—	—	—	—	—	—	W	
			—	—	—	—	—	—	0	0
									Number of dummy cycles (Read recovery time) 00: 2 dummy cycles 01: 1 dummy cycle 10: No dummy cycle 11: Don't use.	

19.4 Clock Generator (CG)

Clock Generator (1 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System Clock Control Register 0	FFFF EE00H	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			High-speed oscillator 0: Disable 1: Enable	Low-speed oscillator 0: Disable 1: Enable	High-speed oscillator after exiting STOP mode 0: Disable 1: Enable	Low-speed oscillator after exiting STOP mode 0: Disable 1: Enable	Clock select after exiting STOP mode 0: High-speed 1: Low-speed	Oscillator warm-up period (WUP) timer On writes: 0: Don't-care 1: Start WUP On reads: 0: Expired 1: Not expired	Prescaler clock select 00: fperiph/4 01: fperiph/2 10: fperiph 11: Reserved	
SYSCR1	System Clock Control Register 1	FFFF EE01H	—	—	SYSCK	FPSEL	DFOSC	—	GEAR1	GEAR0
			—	—	R/W			—	R/W	
			—	—	0	0	0	—	1	1
					System clock (fsys) select 0: High-speed (fgear) 1: Low-speed (fs)	fperiph select 0: fgear 1: fc	High-speed oscillator frequency divide factor 0: Divide-by-2 1: Divide-by-1		High-speed clock (fc) gear select 00: fc 01: fc/2 10: fc/4 11: fc/8	
SYSCR2	System Clock Control Register 2	FFFF EE02H	DRVOSCH	DRVOSCL	WUPT1	WUPT0	STBY1	STBY0	—	DRVE
			R/W	R/W	R/W	R/W	R/W	R/W	—	R/W
			0	0	1	0	1	1	—	0
			High-speed oscillator drive capability 0: High 1: Low	Low-speed oscillator drive capability 0: High 1: Low	Oscillator warm-up time 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		Standby mode select 00: Reserved 01: STOP mode 10: SLEEP mode 11: IDLE mode			1: Pins are driven in STOP mode. 0: Pins are not driven in STOP mode.
SYSCR3	System Clock Control Register 3	FFFF EE03H	—	SCOSEL	—	ALESEL	—	—	LUPFG	LUPTM
			—	R/W	—	R/W	—	—	R/W	
			—	0	—	1	—	—	0	0
				SCOUT output select 0: fs 1: fsys		ALE output width select 0: fsys × 0.5 1: fsys × 1.5			PLL lock 0: Locked 1: Unlocked	PLL lock time select 0: 2^{16}/input frequency 1: 2^{12}/input frequency
ADCCLK	ADC Conversion Clock Register	FFFF EE04H	—	—	—	—	—	—	ADCCK1	ADCCK0
			—	—	—	—	—	—	R/W	R/W
			—	—	—	—	—	—	0	0
									ADC conversion clock (fadc) select 00: fsys/2 01: fsys/4 10: fsys/8 11: Don't use.	

Clock Generator (2 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
IMCGA0	Interrupt CG Control Register A0	FFFF EE10H	—	—	EMCG01	EMCG00	—	—	—	INT0EN
			—	—	R/W		—	—	—	R/W
			—	—	1	0	—	—	—	0
					Wake-up INT0 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT0 enable 0: Disable 1: Enable
IMCGA1	Interrupt CG Control Register A1	FFFF EE11H	—	—	EMCG11	EMCG10	—	—	—	INT1EN
			—	—	R/W		—	—	—	R/W
			—	—	1	0	—	—	—	0
					Wake-up INT1 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT1 enable 0: Disable 1: Enable
IMCGA2	Interrupt CG Control Register A2	FFFF EE12H	—	—	EMCG21	EMCG20	—	—	—	INT2EN
			—	—	R/W		—	—	—	R/W
			—	—	1	0	—	—	—	0
					Wake-up INT2 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT2 enable 0: Disable 1: Enable
IMCGA3	Interrupt CG Control Register A3	FFFF EE13H	—	—	EMCG31	EMCG30	—	—	—	—
			—	—	—	—	—	—	—	—
			—	—	1	0	—	—	—	0
					Wake-up INT3 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT3 enable 0: Disable 1: Enable
IMCGB0	Interrupt CG Control Register B0	FFFF EE14H	—	—	EMCG41	EMCG40	—	—	—	—
			—	—	—	—	—	—	—	—
			—	—	1	0	—	—	—	0
					Wake-up INT4 sensitivity 00: Low level 01: High level 10: Falling edge 11: Rising edge					INT4 enable 0: Disable 1: Enable
IMCGB1	Interrupt CG Control Register B1	FFFF EE15H	—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			—	—	1	0	—	—	—	0
					Must be written as 10.					Must be written as 0.
IMCGB2	Interrupt CG Control Register B2	FFFF EE16H	—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			—	—	1	0	—	—	—	0
					Must be written as 10.					Must be written as 0.
IMCGB3	Interrupt CG Control Register B3	FFFF EE17H	—	—	EMCG71	EMCG72	—	—	—	INTRTCEN
			—	—	R/W		—	—	—	R/W
			—	—	1	0	—	—	—	0
					Must be written as 11.					INTRTC enable 0:Disable 1: Enable
EICRCG	Interrupt Request Clear Register	FFFF EE20H	—	—	—	—	—	ICRCG2	ICRCG1	ICRCG0
			—	—	—		—	—	W	—
			—	—	—	—	—	0	0	0
								Clear interrupt request (Only when relevant interrupts are programmed to be used to exit STOP/SLEEP mode.) 000: INT0 100: INT4 001: INT1 101: Reserved 010: INT2 110: Reserved 011: INT3 111: INTRTC		

19.5 DMA Controller (DMAC)

DMA Controller (1 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CCR0	DMA Channel Control Register 0	FFFF E200H	SAC0	DIO	DAC1	DAC0	TrSiz1	TrSiz0	DPS1	DPS0
			R/W							
			0	0	0	0	0	0	0	0
			Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed	Destination (I/O) 0: Memory 1: I/O	Destination address count 00: Incremented 01: Decremented 1x: Fixed		Transfer size 0x: 32 bits 10: 16 bits 11: 8 bits		Device port size 0x: 32 bits 10: 16 bits 11: 8 bits	
			15	14	13	12	11	10	9	8
			—	ExR	PosE	Lev	SReq	RelEn	SIO	SAC1
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	External request mode 1: External transfer request 0: Internal transfer request	Must be written as 0.	Must be written as 1.	Snoop request 0: Disabled 1: Enabled	Bus release request enable 0: Disabled 1: Enabled	Source (I/O) 0: Memory 1: I/O	Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed
			23	22	21	20	19	18	17	16
			NIEn	AbIEn	—	—	—	—	Big	—
			R/W							
			1	1	1	0	0	0	1	0
			Normal completion interrupt enable 0: Disabled 1: Enabled	Abnormal termination interrupt enable 0: Disabled 1: Enabled	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.
			31	30	29	28	27	26	25	24
			Str	—	—	—	—	—	—	—
			W	—	—	—	—	—	—	W
			0	0	0	0	0	0	0	0
			1: Channel 0 start							Must be written as 0.

DMA Controller (2 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CSR0	DMA Channel Status Register 0	FFFF E204H	—	—	—	—	—	—	—	—
			—	—	—	—	—	R/W		
			0	0	0	0	0	0	0	0
								Must be written as 0.	Must be written as 0.	Must be written as 0.
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			—	—	—	—	—	—	—	—
			23	22	21	20	19	18	17	16
			NC	AbC	—	BES	BED	Conf	—	—
			R/W			R			—	—
			0	0	0	0	0	0	0	0
			1: Normal completion status flag	1: Abnormal termination status flag	Must be written as 0.	1: Bus error (source)	1: Bus error (destination)	1: Configuration error		
			31	30	29	28	27	26	25	24
			Act	—	—	—	—	—	—	—
			R	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			1: Channel 0 active							

DMA Controller (3 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SAR0	DMA Source Address Register 0	FFFF E208H	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			SAddr23	SAddr22	SAddr21	SAddr20	SAddr19	SAddr18	SAddr17	SAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			SAddr31	SAddr30	SAddr29	SAddr28	SAddr27	SAddr26	SAddr25	SAddr24
			R/W							
			Undefined							
DAR0	DMA Destination Address Register 0	FFFF E20CH	7	6	5	4	3	2	1	0
			DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			DAddr23	DAddr22	DAddr21	DAddr20	DAddr19	DAddr18	DAddr17	DAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			DAddr31	DAddr30	DAddr29	DAddr28	DAddr27	DAddr26	DAddr25	DAddr24
			R/W							
			Undefined							
BCR0	DMA Byte Count Register 0	FFFF E210H	7	6	5	4	3	2	1	0
			BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			BC23	BC22	BC21	BC20	BC19	BC18	BC17	BC16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (4 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
DTCR0	DMA Transfer Control Register 0	FFFF E218H	—	—	DACM2	DACM1	DACM0	SACM2	SACM1	SACM0
			—	—	R/W					
			0	0	0	0	0	0	0	0
					Bit position at which destination addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved			Bit position at which source addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved		
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0

DMA Controller (5 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CCR1	DMA Channel Control Register 1	FFFF E220H	SAC0	DIO	DAC1	DAC0	TrSiz1	TrSiz0	DPS1	DPS0
			R/W							
			0	0	0	0	0	0	0	0
			Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed	Destination (I/O) 0: Memory 1: I/O	Destination address count 00: Incremented 01: Decremented 1x: Fixed		Transfer size 0x: 32 bits 10: 16 bits 11: 8 bits		Device port size 0x: 32 bits 10: 16 bits 11: 8 bits	
			15	14	13	12	11	10	9	8
			—	ExR	PosE	Lev	SReq	RelEn	SIO	SAC1
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	External request mode 1: External 0: Internal	Must be written as 0.	Must be written as 1.	Snoop request 0: Disabled 1: Enabled	Bus release request enable 0: Disabled 1: Enabled	Source (I/O) 0: Memory 1: I/O	Source address count (bits 8 & 7) 00: Inc 01: Dec 1x: Fixed
			23	22	21	20	19	18	17	16
			NIEn	AbIEn	—	—	—	—	Big	—
			R/W							
			1	1	1	0	0	0	1	0
			Normal completion interrupt enable 0: Disabled 1: Enabled	Abnormal termination interrupt enable 0: Disabled 1: Enabled	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.
			31	30	29	28	27	26	25	24
			Str	—	—	—	—	—	—	—
			W	—	—	—	—	—	—	W
			0	0	0	0	0	0	0	0
			1: Channel 1 start							Must be written as 0.

DMA Controller (6 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CSR1	DMA Channel Status Register 1	FFFF E224H	—	—	—	—	—	—	—	—
			—	—	—	—	—	R/W		
			0	0	0	0	0	0	0	0
								Must be written as 0.	Must be written as 0.	Must be written as 0.
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			NC	AbC	—	BES	BED	Conf	—	—
			R/W			R			—	—
			0	0	0	0	0	0	0	0
			1: Normal termination status flag	1: Abnormal termination status flag	Must be written as 0.	1: Bus error (source)	1: Bus error (destination)	1: Configuration error		
			31	30	29	28	27	26	25	24
			Act	—	—	—	—	—	—	—
			R	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			1: Channel 1 active							

DMA Controller (7 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SAR1	DMA Source Address Register 1	FFFF E228H	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			SAddr23	SAddr22	SAddr21	SAddr20	SAddr19	SAddr18	SAddr17	SAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			SAddr31	SAddr30	SAddr29	SAddr28	SAddr27	SAddr26	SAddr25	SAddr24
			R/W							
			Undefined							
DAR1	DMA Destination Address Register 1	FFFF E22CH	7	6	5	4	3	2	1	0
			DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			DAddr23	DAddr22	DAddr21	DAddr20	DAddr19	DAddr18	DAddr17	DAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			DAddr31	DAddr30	DAddr29	DAddr28	DAddr27	DAddr26	DAddr25	DAddr24
			R/W							
			Undefined							
BCR1	DMA Byte Count Register 1	FFFF E230H	7	6	5	4	3	2	1	0
			BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			BC23	BC22	BC21	BC20	BC19	BC18	BC17	BC16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (8 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
DTCR1	DMA Transfer Control Register 1	FFFF E238H	—	—	DACM2	DACM1	DACM0	SACM2	SACM1	SACM0
			—	—	R/W					
			0	0	0	0	0	0	0	0
					Bit position at which destination addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved			Bit position at which source addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved		
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (9 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CCR2	DMA Channel Control Register 2	FFFF E240H	SAC0	DIO	DAC1	DAC0	TrSiz1	TrSiz0	DPS1	DPS0
			R/W							
			0	0	0	0	0	0	0	0
			Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed	Destination (I/O) 0: Memory 1: I/O	Destination address count 00: Incremented 01: Decremented 1x: Fixed		Transfer size 0x: 32 bits 10: 16 bits 11: 8 bits		Device port size 0x: 32 bits 10: 16 bits 11: 8 bits	
			15	14	13	12	11	10	9	8
			—	ExR	PosE	Lev	SReq	RelEn	SIO	SAC1
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	External request mode 1: External transfer request 0: Internal transfer request	Must be written as 0.	Must be written as 1.	Snoop request 0: Disabled 1: Enabled	Bus release request enable 0: Disabled 1: Enabled	Source (I/O) 0: Memory 1: I/O	Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed
			23	22	21	20	19	18	17	16
			NIEn	AbIEn	—	—	—	—	Big	—
			R/W							
			1	1	1	0	0	0	1	0
			Normal completion interrupt enable 0: Disabled 1: Enabled	Abnormal termination interrupt enable 0: Disabled 1: Enabled	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.
			31	30	29	28	27	26	25	24
			Str	—	—	—	—	—	—	—
			W	—	—	—	—	—	—	W
			0	0	0	0	0	0	0	0
			1: Channel 2 start							Must be written as 0.

DMA Controller (10 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CSR2	DMA Channel Status Register 2	FFFF E244H	—	—	—	—	—	—	—	—
			—	—	—	—	—	R/W		
			0	0	0	0	0	0	0	0
								Must be written as 0.	Must be written as 0.	Must be written as 0.
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			NC	AbC	—	BES	BED	Conf	—	—
			R/W			R			—	—
			0	0	0	0	0	0	0	0
			1: Normal completion status flag	1: Abnormal termination status flag	Must be written as 0.	1: Bus error (source)	1: Bus error (destination)	1: Configuration error		
			31	30	29	28	27	26	25	24
			Act	—	—	—	—	—	—	—
			R	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			1: Channel 2 active							

DMA Controller (11 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SAR2	DMA Source Address Register 2	FFFF E248H	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			SAddr23	SAddr22	SAddr21	SAddr20	SAddr19	SAddr18	SAddr17	SAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			SAddr31	SAddr30	SAddr29	SAddr28	SAddr27	SAddr26	SAddr25	SAddr24
			R/W							
			Undefined							
DAR2	DMA Destination Address Register 2	FFFF E24CH	7	6	5	4	3	2	1	0
			DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			DAddr23	DAddr22	DAddr21	DAddr20	DAddr19	DAddr18	DAddr17	DAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			DAddr31	DAddr30	DAddr29	DAddr28	DAddr27	DAddr26	DAddr25	DAddr24
			R/W							
			Undefined							
BCR2	DMA Byte Count Register 2	FFFF E250H	7	6	5	4	3	2	1	0
			BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			BC23	BC22	BC21	BC20	BC19	BC18	BC17	BC16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (12 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
DTCR2	DMA Transfer Control Register 2	FFFF E258H	—	—	DACM2	DACM1	DACM0	SACM2	SACM1	SACM0
			—	—	R/W					
			0	0	0	0	0	0	0	0
					Bit position at which destination addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved			Bit position at which source addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved		
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (13 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CCR3	DMA Channel Control Register 3	FFFF E260H	SAC0	DIO	DAC1	DAC0	TrSiz1	TrSiz0	DPS1	DPS0
			R/W							
			0	0	0	0	0	0	0	0
			Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed	Destination (I/O) 0: Memory 1: I/O	Destination address count 00: Incremented 01: Decremented 1x: Fixed		Transfer size 0x: 32 bits 10: 16 bits 11: 8 bits		Device port size 0x: 32 bits 10: 16 bits 11: 8 bits	
			15	14	13	12	11	10	9	8
			—	ExR	PosE	Lev	SReq	RelEn	SIO	SAC1
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	External request mode 1: External transfer request 0: Internal transfer request	Must be written as 0.	Must be written as 1.	Snoop request 0: Disabled 1: Enabled	Bus release request enable 0: Disabled 1: Enabled	Source (I/O) 0: Memory 1: I/O	Source address count (bits 8 & 7) 00: Incre-mented 01: Decre-mented 1x: Fixed
			23	22	21	20	19	18	17	16
			NIEn	AbIEn	—	—	—	—	Big	—
			R/W							
			1	1	1	0	0	0	1	0
			Normal completion interrupt enable 0: Disabled 1: Enabled	Abnormal termination interrupt enable 0: Disabled 1: Enabled	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.
			31	30	29	28	27	26	25	24
			Str	—	—	—	—	—	—	—
			W	—	—	—	—	—	—	W
			0	0	0	0	0	0	0	0
			1: Channel 3 start							Must be written as 0.

DMA Controller (14 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
CSR3	DMA Channel Status Register 3	FFFF E264H	—	—	—	—	—	—	—	—
			—	—	—	—	—	R/W		
			0	0	0	0	0	0	0	0
								Must be written as 0.	Must be written as 0.	Must be written as 0.
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			NC	AbC	—	BES	BED	Conf	—	—
			R/W			R			—	—
			0	0	0	0	0	0	0	0
			1: Normal termination status flag	1: Abnormal termination status flag	Must be written as 0.	1: Bus error (source)	1: Bus error (destination)	1: Configuration error		
			31	30	29	28	27	26	25	24
			Act	—	—	—	—	—	—	—
			R	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			1: Channel 3 active							

DMA Controller (15 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SAR3	DMA Source Address Register 3	FFFF E268H	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			SAddr23	SAddr22	SAddr21	SAddr20	SAddr19	SAddr18	SAddr17	SAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			SAddr31	SAddr30	SAddr29	SAddr28	SAddr27	SAddr26	SAddr25	SAddr24
			R/W							
			Undefined							
DAR3	DMA Destination Address Register 3	FFFF E26CH	7	6	5	4	3	2	1	0
			DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			DAddr23	DAddr22	DAddr21	DAddr20	DAddr19	DAddr18	DAddr17	DAddr16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			DAddr31	DAddr30	DAddr29	DAddr28	DAddr27	DAddr26	DAddr25	DAddr24
			R/W							
			Undefined							
BCR3	DMA Byte Count Register 3	FFFF E270H	7	6	5	4	3	2	1	0
			BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			BC23	BC22	BC21	BC20	BC19	BC18	BC17	BC16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0

DMA Controller (16 of 16)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
DTCR3	DMA Transfer Control Register 3	FFFF E278H	—	—	DACM2	DACM1	DACM0	SACM2	SACM1	SACM0
			—	—	R/W					
			0	0	0	0	0	0	0	0
					Bit position at which destination addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved			Bit position at which source addresses are counted 000: Bit 0 001: Bit 4 010: Bit 8 011: Bit 12 100: Bit 16 101: Reserved 110: Reserved 111: Reserved		
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			—	—	—	—	—	—	—	—
			—							
			0	0	0	0	0	0	0	0
DCR	DMA Control Register	FFFF E280H	7	6	5	4	3	2	1	0
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			15	14	13	12	11	10	9	8
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			23	22	21	20	19	18	17	16
			—	—	—	—	—	—	—	—
			—	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			31	30	29	28	27	26	25	24
			Rst	—	—	—	—	—	—	—
			W	—	—	—	—	—	—	—
			0	0	0	0	0	0	0	0
			1: DMAC software reset							
DHR	DMA Data Holding Register	FFFF E28CH	7	6	5	4	3	2	1	0
			DOT7	DOT6	DOT5	DOT4	DOT3	DOT2	DOT1	DOT0
			R/W							
			Undefined							
			15	14	13	12	11	10	9	8
			DOT15	DOT14	DOT13	DOT12	DOT11	DOT10	DOT9	DOT8
			R/W							
			Undefined							
			23	22	21	20	19	18	17	16
			DOT23	DOT22	DOT21	DOT20	DOT19	DOT18	DOT17	DOT16
			R/W							
			Undefined							
			31	30	29	28	27	26	25	24
			DOT31	DOT30	DOT29	DOT28	DOT27	DOT26	DOT25	DOT24
			R/W							
			Undefined							

19.6 8-Bit Timers (TMRAs)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
TA01RUN	TMRA01 Run Register	FFFF F100H	TA0RDE	—	—	—	I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W	—	—	—	R/W			
			0	—	—	—	0	0	0	0
			Double Buffering 0: Disable 1: Enable				IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run	Run/Stop Control 0: Stop & clear 1: Run	
TA23RUN	TMRA23 Run Register	FFFF F108H	TA2RDE	—	—	—	I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W	—	—	—	R/W			
			0	—	—	—	0	0	0	0
			Double Buffering 0: Disable 1: Enable				IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run	Run/Stop Control 0: Stop & clear 1: Run	
TA01MOD	TMRA01 Mode Register	FFFF F104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		TMRA1 clock source 00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA0 clock source 00: TA0IN input 01: ϕT1 10: ϕT4 11: ϕT16	
TA23MOD	TMRA23 Mode Register	FFFF F10CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operating mode 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM period 00: Reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		TMRA3 clock source 00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA2 clock source 00: TA2IN input pin 01: ϕT1 10: ϕT4 11: ϕT16	
TA1FFCR	TMRA01 Timer Flip-Flop Control Register	FFFF F105H	—	—	—	—	TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS
			—	—	—	—	R/W			
			—	—	—	—	1	1	0	0
							00: Toggles TA1FF. (software toggle) 01: Sets TA1FF to 1. 10: Clears TA1FF to 0. 11: Don't-care This field is always read as 11.		TA1FF toggle enable 0: Disable 1: Enable	TA1FF toggle trigger 0: TMRA0 1: TMRA1
TA3FFCR	TMRA23 Timer Flip-Flop Control Register	FFFF F10DH	—	—	—	—	TAFF3C1	TAFF3C0	TAFF3IE	TAFF3IS
			—	—	—	—	R/W			
			—	—	—	—	1	1	0	0
							00: Toggles TA3FF (software toggle). 01: Sets TA3FF to 1 10: Clears TA3FF to 0 11: Don't care This field is always read as 11.		TA3FF toggle enable 0: Disable 1: Enable	TA3FF trigger 0: TMRA2 1: TMRA3

19.7 16-Bit Timer/Event Counters (TMRBs)

16-Bit Timer Control (1 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
TB0RUN	TMRB0 Run Register	FFFF F180H	TB0RDE	—	—	—	I2TB0	TB0PRUN	—	TB0RUN
			R/W		—	—	R/W		—	R/W
			0	0	—	—	0	0	—	0
			Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run
TB1RUN	TMRB1 Run Register	FFFF F190H	TB1RDE	—	—	—	I2TB1	TB1PRUN	—	TB1RUN
			R/W		—	—	R/W		—	R/W
			0	0	—	—	0	0	—	0
			Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run
TB2RUN	TMRB2 Run Register	FFFF F1A0H	TB2RDE	—	—	—	I2TB2	TB2PRUN	—	TB2RUN
			R/W		—	—	R/W		—	R/W
			0	0	—	—	0	0	—	0
			Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run
TB3RUN	TMRB3 Run Register	FFFF F1B0H	TB3RDE	—	—	—	I2TB3	TB3PRUN	—	TB3RUN
			R/W		—	—	R/W		—	R/W
			0	0	—	—	0	0	—	0
			Double Buffering 0: Disable 1: Enable	Must be written as 0.			IDLE 0: Off 1: On	Prescalar Run/Stop Control 0: Stop 1: Run		Run/Stop Control 0: Stop & clear 1: Run
TB0MOD	TMRB0 Mode Register	FFFF F182H	—	—	TB0CP0	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB0IN0↑TB0IN1↑ 10: TB0IN0↑TB0IN0↓ 11: TA1OUT↑TA1OUT↓		UC0 clear control 0: Disable 1: Enable	TMRB0 clock source 00: TB0IN0 input 01: φT1 10: φT4 11: φT16	
TB1MOD	TMRB1 Mode Register	FFFF F192H	—	—	TB1CP0	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB1IN0↑TB1IN1↑ 10: TB1IN0↑TB1IN0↓ 11: TA1OUT↑TA1OUT↓		UC1 clear control 0: Disable 1: Enable	TMRB1 clock source 00: TB1IN0 input 01: φT1 10: φT4 11: φT16	
TB2MOD	TMRB2 Mode Register	FFFF F1A2H	—	—	TB2CP0	TB2CPM1	TB2CPM0	TB2CLE	TB2CLK1	TB2CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: TB2IN0↑TB2IN1↑ 10: TB2IN0↑TB2IN0↓ 11: TA1OUT↑TA1OUT↓		UC2 clear control 0: Disable 1: Enable	TMRB2 clock source 00: TB2IN0 input 01: φT1 10: φT4 11: φT16	
TB3MOD	TMRB3 Mode Register	FFFF F1B2H	—	—	TB3CP0	TB3CPM1	TB3CPM0	TB3CLE	TB3CLK1	TB3CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			Must be written as 00.		Software capture 0: Capture 1: Don't care	Capture triggers 00: Disabled 01: Disabled 10: Disabled 11: TA1OUT↑TA1OUT↓		UC3 clear control 0: Disable 1: Enable	TMRB3 clock source 00: TB3IN0 input 01: φT1 10: φT4 11: φT16	

16-Bit Timer Control (2 of 2)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
TB0FFCR	TMRB0 Timer Flip-Flop Control Register	FFFF F183H	—	—	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Must be written as 11. * This field is always read as 11.		TB0FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB0FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
					UC0 →TB0CP1	UC0 → TB0CP0	UC0 = TB0RG1	UC0 = TB0RG0		
TB1FFCR	TMRB1 Timer Flip-Flop Control Register	FFFF F193H	—	—	TB1C1T1	TB1C0T1	TB1E1T1	TB1E0T1	TB1FF0C1	TB1FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Must be written as 11. * This field is always read as 11.		TB1FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB1FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
					UC1 → TB1CP1	UC1 → TB1CP0	UC1 = TB1RG1	UC1 = TB1RG0		
TB2FFCR	TMRB2 Timer Flip-Flop Control Register	FFFF F1A3H	—	—	TB2C1T1	TB2C0T1	TB2E1T1	TB2E0T1	TB2FF0C1	TB2FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Must be written as 11. * This field is always read as 11.		TB2FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB2FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
					UC2 → TB2CP1	UC2 → TB2CP0	UC2 = TB2RG1	UC2 = TB2RG0		
TB3FFCR	TMRB3 Timer Flip-Flop Control Register	FFFF F1B3H	—	—	TB3C1T1	TB3C0T1	TB3E1T1	TB3E0T1	TB3FF0C1	TB3FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			Must be written as 11. * This field is always read as 11.		TB3FF0 toggle-trigger 0: Trigger disabled 1: Trigger enabled				TB3FF0 control 00: Toggle 01: Set 10: Clear 11: Don't care * This field is always read as 11.	
					UC3 → TB3CP1	UC3 → TB3CP0	UC3 = TB3RG1	UC3 = TB3RG0		

19.8 Serial I/O (SIO)

SIO0

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SC0CR	Serial Channel 0 Control Register	FFFF F201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared when read)			R/W	
			0	0	0	0	0	0	0	0
			Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	0:SCLK0↑ 1:SCLK0↓	0: Baud rate generator 1: SCLK0 input
SC0MOD0	Serial Channel 0 Mode Register 0	FFFF F202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Bit 8 of a transmitted character	Handshake control 0: Disables CTS operation 1: Enables CTS operation	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: I/O Interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: External clock (SCLK0 input)	
BR0CR	Baud Rate Generator 0 Control Register	FFFF F203H	—	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			
BR0ADD	Baud Rate Generator 0 Control Register	FFFF F204H	—	—	—	—	BR0K3	BR0K2	BR0K1	BR0K0
			—	—	—	—	R/W			
			—	—	—	—	0	0	0	0
							Value of K in N+(16–K)/16			
SC0MOD1	Serial Channel 0 Mode Register 1	FFFF F205H	I2S0	FDPX0	—	—	—	—	—	—
			R/W	R/W	—	—	—	—	—	—
			0	0	—	—	—	—	—	—
			IDLE 0: Off 1: On	Synchro-nous 0: Half-duplex 1: Full-duplex						

SIO1

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SC1CR	Serial Channel 1 Control Register	FFFF F209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared when read)			R/W	
			0	0	0	0	0	0	0	0
			Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	0: SCLK1↑ 1: SCLK1↓	0: Baud rate generator 1: SCLK1 input
SC1MOD0	Serial Channel 1 Mode Register 0	FFFF F20AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Bit 8 of a transmitted character	Handshake control 0: Disables CTS operation 1: Enables CTS operation	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: I/O Interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: External clock (SCLK1 input)	
BR1CR	Baud Rate Generator 1 Control Register	FFFF F20BH	—	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			
BR1ADD	Baud Rate Generator 1 Control Register	FFFF F20CH	—	—	—	—	BRK1K3	BRK1K2	BRK1K1	BRK1K0
			—	—	—	—	R/W			
			—	—	—	—	0	0	0	0
							Value of K in N+(16–K)/16			
SC1MOD1	Serial Channel 1 Mode Register 1	FFFF F20DH	I2S0	FDPX0	—	—	—	—	—	—
			R/W		—	—	—	—	—	—
			0	0	—	—	—	—	—	—
			IDLE 0: Off 1: On	Synchro-nous 0: Half-duplex 1: Full-duplex						

SIO3

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SC3CR	Serial Channel 3 Control Register	FFFF F281H	RB8	EVEN	PE	OERR	PERR	FERR	—	—
			R	R/W		R (Cleared when read)			R/W	
			0	0	0	0	0	0	0	0
			Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	Must be written as 00.	
SC3MOD0	Serial Channel 3 Mode Register 0	FFFF F282H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Bit 8 of a transmitted character	Must be written as 0.	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: Reserved 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: Don't care	
BR3CR	Baud Rate Generator 3 Control Register	FFFF F283H	—	BR3ADDE	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			
BR3ADD	Baud Rate Generator 3 Control Register	FFFF F284H	—	—	—	—	BR3K3	BR3K2	BR3K1	BR3K0
			—	—	—	—	R/W			
			—	—	—	—	0	0	0	0
			—	—	—	—	Value of K in N+(16–K)/16			
SC3MOD1	Serial Channel 3 Mode Register 1	FFFF F285H	I2S0	—	—	—	—	—	—	—
			R/W	—	—	—	—	—	—	—
			0	—	—	—	—	—	—	—
			IDLE 0: Off 1: On							

SIO4

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SC4CR	Serial Channel 4 Control Register	FFFF F289H	RB8	EVEN	PE	OERR	PERR	FERR	—	—
			R	R/W		R (Cleared when read)			R/W	
			0	0	0	0	0	0	0	0
			Bit 8 of a received character	Parity type 0: Odd 1: Even	Parity 0: Disabled 1: Enabled	1: Error has occurred. Overrun	Parity	Framing	Must be written as 00.	
SC4MOD0	Serial Channel 4 Mode Register 0	FFFF F28AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Bit 8 of a transmitted character	Must be written as 0.	Receive control 0: Disables receiver 1: Enables receiver	Wake-up function 0: Disabled 1: Enabled	Serial transfer mode 00: Reserved 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial clock (for UART) 00: TA0TRG (timer) 01: Baud rate generator 10: Internal fsys/2 clock 11: Don't care	
BR4CR	Baud Rate Generator 4 Control Register	FFFF F28BH	—	BR4ADDE	BR4CK1	BR4CK0	BR4S3	BR4S2	BR4S1	BR4S0
			R/W							
			0	0	0	0	0	0	0	0
			Must be written as 0.	N + (16–K)/16 function 0: Disabled 1: Enabled	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Clock divisor value N			
BR4ADD	Baud Rate Generator 4 Control Register	FFFF F28CH	—	—	—	—	BR4K3	BR4K2	BR4K1	BR4K0
			—	—	—	—	R/W			
			—	—	—	—	0	0	0	0
							Value of K in N+(16–K)/16			
SC4MOD1	Serial Channel 4 Mode Register 1	FFFF F28DH	I2S0	—	—	—	—	—	—	—
			R/W	—	—	—	—	—	—	—
			0	—	—	—	—	—	—	—
			IDLE 0: Off 1: On							

19.9 Serial Bus Interface (SBI)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0/
SBI0CR1	Serial Bus Interface Control Register 1	FFFF F240H (I^2C Bus Mode)	BC2	BC1	BC0	ACK	—	SCK2	SCK1	SCK0 SWRMON
			W			R/W	—	W	W	R/W
			0	0	0	0	—	0	0	1
			Number of bits per transfer (when ACK = 0) 000: 8, 001: 1, 010: 2 011: 3, 100: 4, 101: 5 110: 6, 111: 7			ACK clock pulse 0: No ACK 1: ACK		Internal SCL output clock frequency (on writes) / Software reset monitor 000: 4, 001: 5, 010: 6 011: 7, 100: 8, 101: 9 110: 10, 111: Reserved		
		FFFF F240H (SIO Mode)	SIOS	SIOINH	SIOM1	SIOM0	—	SCK2	SCK1	SCK0
			W				—	W		R/W
			0	0	0	0	—	0	0	1
			Start transfer 0: Stop 1: Start	Abort transfer 0: Continue 1: Abort	Transfer mode 00: Transmit mode 01: Reserved 10: Transmit/Receive mode 11: Receive mode			Serial clock frequency (on writes) / Software reset monitor 000: 3, 001: 4, 010: 5 011: 6, 100: 7, 101: 8 110: 9, 111: External clock		
SBI0DBR	SBI Data Buffer Register	FFFF F241H	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
			R (receive) / W (transmit)							
			Undefined							
I2C0AR	I^2C bus Address Register	FFFF F242H	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W							
			0	0	0	0	0	0	0	0
			When the SBI is addressed as a slave, this field specifies a 7-bit I^2C-bus address to which the SBI responds.							Address recognition 0: Recognize 1: Does not recognize
SBI0CR2 on writes SBI0SR on reads	Serial Bus Interface Control 2 /Status Register	FFFF F243H (I^2C Bus Mode)	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
			W							
			0	0	0	1	0	0	0	0
			Master/ slave	Transmit/ receive	START/ STOP generation	INTSBI interrupt clear	Operating mode 00: Port mode 01: SIO mode 10: I^2C Bus mode 11: Reserved		Software reset A write of 10 followed by a write of 01	
			MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			R							
			0	0	0	1	0	0	0	0
			Master/ slave	Transmit/ receive	I^2C Bus status	INTS2 interrupt status	Arbitration lost 0: — 1: Detected	Addressed as slave 0: — 1: Detected	Address 0 (general call) 0: — 1: Detected	Last received bit 0: 0 1: 1
		FFFF F243H (SIO Mode)	—	—	—	—	SIOF	SEF	—	—
			—	—	—	—	R		—	—
			—	—	—	—	0	0	—	—
							Serial transfer status 0: Terminated 1: In progress	Shift operation status 0: Terminated 1: In progress		
SBI0BR0	Serial Bus Interface Control Register 0	FFFF F244H	—	I^2CSBI0	—	—	—	—	—	—
			—	R/W	—	—	—	—	—	W
			—	0	—	—	—	—	—	0
				IDLE 0: Off 1: On						Must be written as 0.
SBI0BR1	Serial Bus Interface Control Register 1	FFFF F245H	P4EN	—	—	—	—	—	—	—
			R/W	—	—	—	—	—	—	—
			0	—	—	—	—	—	—	—
			Internal clock 0: Off 1: On							

19.10 A/D Converter (ADC)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	A/D Mode Control Register 0	FFFF F310H	EOCF	ADBF	—	—	ITM0	REPEAT	SCAN	ADS
			R		R/W					
			0	0	0	0	0	0	0	0
			End-of-conversion flag 0: Before conversion or conversion in progress 1: Conversion completed	A/D conversion busy flag 0: Idle 1: Conversion in progress	Must be written as 0.	Must be written as 0.	Interrupt timing in fixed-channel continuous conversion mode	1: Continuous conversion	1: Channel scan conversion	1: A/D conversion start
ADMOD1	A/D Mode Control Register 1	FFFF F311H	VREFON	I2AD	—	—	ADTRGE	ADCH2	ADCH1	ADCH0
			R/W		—	—	R/W			
			0	0	—	—	0	0	0	0
			VREF control 0: Off 1: On	IDLE 0: Off 1: On			External conversion trigger 0: Disable 1: Enable	Analog input channel select		
ADREG04L	A/D Conversion Result Reg 0/4 Low	FFFF F300H	ADR01	ADR00	—	—	—	—	—	ADR0RF
			R		—	—	—	—	—	R
			Undefined		—	—	—	—	—	0
ADREG04H	A/D Conversion Result Reg 0/4 High	FFFF F301H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG15L	A/D Conversion Result Reg 1/5 Low	FFFF F302H	ADR11	ADR10	—	—	—	—	—	ADR1RF
			R		—	—	—	—	—	R
			Undefined		—	—	—	—	—	0
ADREG15H	A/D Conversion Result Reg 1/5 High	FFFF F303H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG26L	A/D Conversion Result Reg 2/6 Low	FFFF F304H	ADR21	ADR20	—	—	—	—	—	ADR2RF
			R		—	—	—	—	—	R
			Undefined		—	—	—	—	—	0
ADREG26H	A/D Conversion Result Reg 2/6 High	FFFF F305H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG37L	AD Result Reg 3/7 low	FFFF F306H	ADR31	ADR30	—	—	—	—	—	ADR3RF
			R		—	—	—	—	—	R
			Undefined		—	—	—	—	—	0
ADREG37H	A/D Conversion Result Reg 3/7 High	FFFF F307H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
ADCCLK	A/D Conversion Clock Select Register	FFFF EE04H	—	—	—	—	—	—	ADCCK1	ADCCK0
			—	—	—	—	—	—	R/W	
			—	—	—	—	—	—	0	0
									A/D conversion clock 00: fsys/2 01: fsys/4 10: fsys/8 11: Reserved	

Analog input channel select

	SCAN=0	SCAN=1
000	AN0	AN0
001	AN1	AN0 →AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN6 → AN7

19.11 Watchdog Timer (WDT)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT Mode Register	FFFF F090H	WDTE	WDTP1	WDTP0	—	—	I2WDT	RESCR	—
			R/W	R/W		—	—	R/W		
			1	0	0	—	—	0	0	0
			1: WDT enable	00: 2^{16}/fsys 01: 2^{18}/ fsys 10: 2^{20}/ fsys 11: 2^{22}/ fsys				IDLE 0: Off 1: On	1: System reset	Must be written as 0.
WDCR	WDT Control Register	FFFF F091H	—							
			W							
			—							
			B1H: WDT disable code; 4EH: WDT clear-count code							

19.12 Real-Time Clock (RTC)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
RTCCR	RTC Control Register	FFFF F0A0H	—	—	—	—	RTCRCLR	RTCSEL1	RTCSEL0	RTCRUN
			R/W	—	—	—	R/W	R/W		R/W
			0	—	—	—	0	0	0	0
			Must be written as 0.				0: Clears Accumulator.	00: 2^{14}/fs 01: 2^{13}/fs 10: 2^{12}/fs 11: 2^{11}/fs		0: Stop and clear the counter. 1: Begin counting.
RTCREG	RTC Accumu-lator Register	FFFF F0A4H	RUI7	RUI6	RUI5	RUI4	RUI3	RUI2	RUI1	RUI0
			R							
			0	0	0	0	0	0	0	0

19.13 Flash Control/Status (TMP1940FDBF Only)

Mnemonic	Name	Address	7	6	5	4	3	2	1	0
SEQMOD	Security Mode Register	FFFF E510	—	—	—	—	—	—	—	SEQON
			—	—	—	—	—	—	—	R/W
			—	—	—	—	—	—	—	1
										1: Security on 0: Security off
SEQCNT	Security Control Register	FFFF E514	—	—	—	—	—	—	—	—
			W							
			—	—	—	—	—	—	—	—
			Must be written as 0x0000_00C5.							
			—	—	—	—	—	—	—	—
			W							
			—	—	—	—	—	—	—	—
			Must be written as 0x0000_00C5.							
			—	—	—	—	—	—	—	—
			W							
			—	—	—	—	—	—	—	—
			Must be written as 0x0000_00C5.							
			—	—	—	—	—	—	—	—
			W							
			—	—	—	—	—	—	—	—
			Must be written as 0x0000_00C5.							
FLCS	Flash Control/ Status Register	FFFF E520H	—	—	—	—	—	RDY_BSY	—	FSE
			—	—	—	—	R/W	R	R/W	R/W
			—	—	—	—	0	1	0	0
							Must be written as 0.	0: Busy 1: Ready	Must be written as 0.	0: Access main logic. 1: Access security logic.

Note: The SEQMOD and FLCS registers are 32-bit registers and must be accessed as a 32-bit quantity.

20. I/O Port Equivalent-Circuit Diagrams

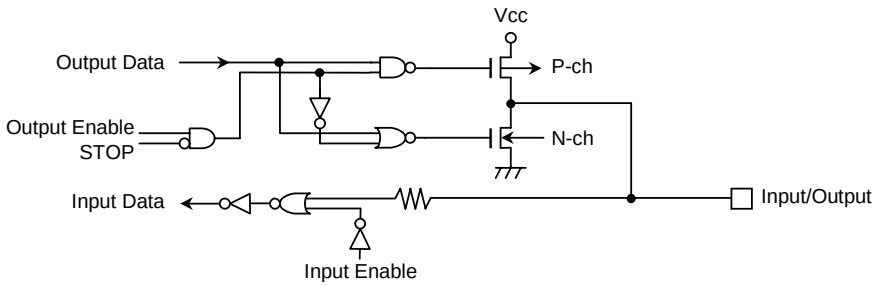
- How to read circuit diagrams

  The circuit diagrams in this chapter are drawn using the same gate symbols as for the 74HCxx Series standard CMOS logic ICs.

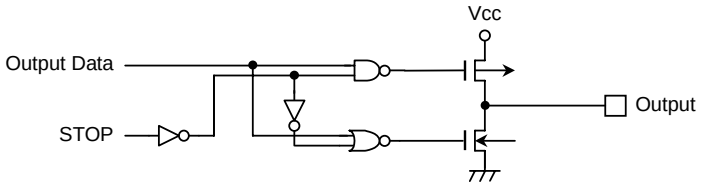
  The signal named STOP has a unique function. This signal goes active-high if the CPU sets the HALT bit when the STBY[1:0] field in the SYSCR2 register is programmed to 01 (i.e., STOP mode) and the Drive Enable (DRVE) bit in the same register is cleared. If the DRVE bit is set, the STOP signal remains inactive (at logic 0).

- The input protection circuit has a resistor in the range of several tens to several hundreds of ohms.

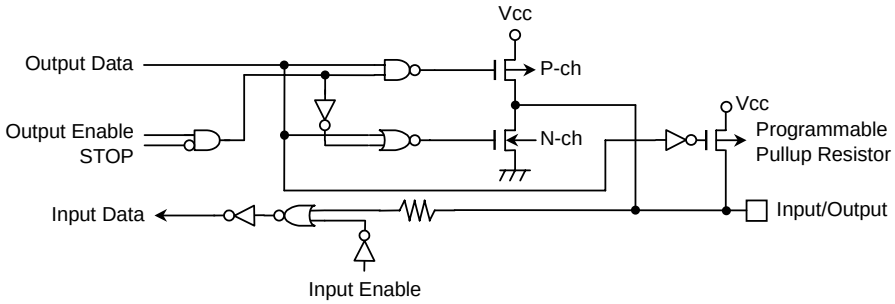
■ Port 0 (AD0–AD7), Port 1 (AD8–AD15, A8–A15), Port 2 (A16–A23, A0–A7), P44, P71, P73–P76, P80–P87, P91–P92, P94–P95, PA0–PA5



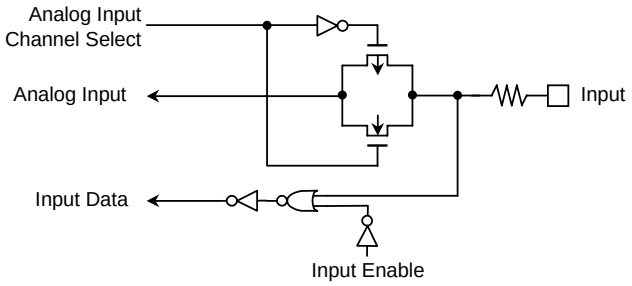
■ P30 ($\overline{RD}$), P31 ($\overline{WR}$)



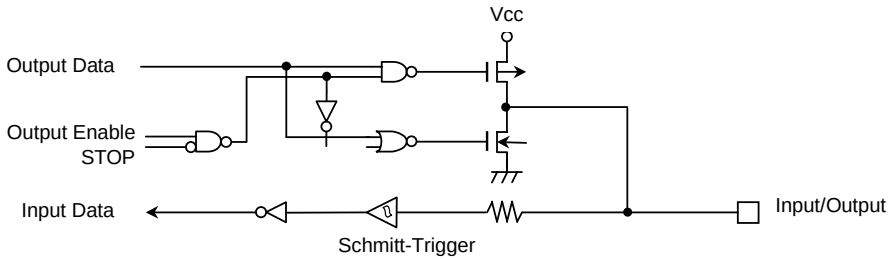
■ P32–P37, P40–P43



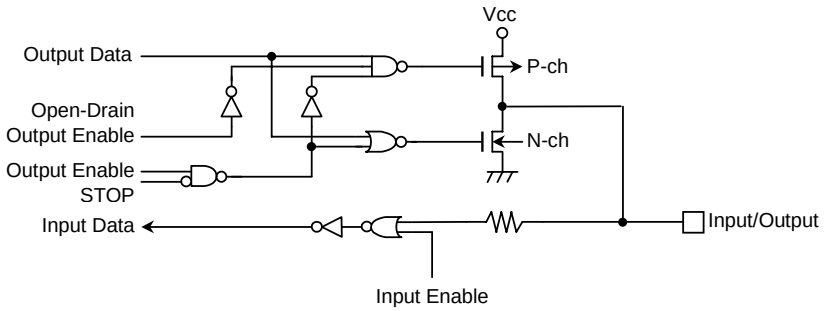
■ Port 5 (AN0–AN7)



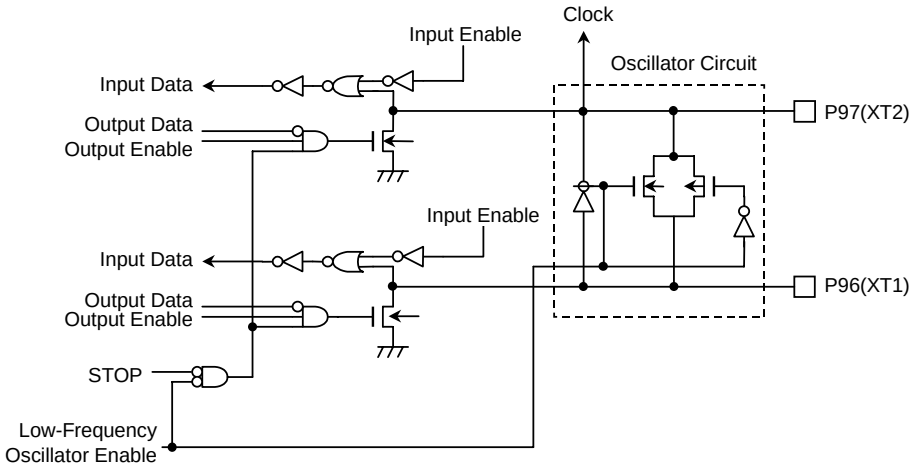
■ P77 (INT0)



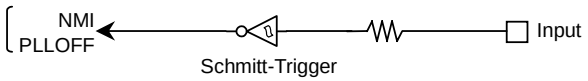
■ P70, P72, P90, P93, PA6–PA7



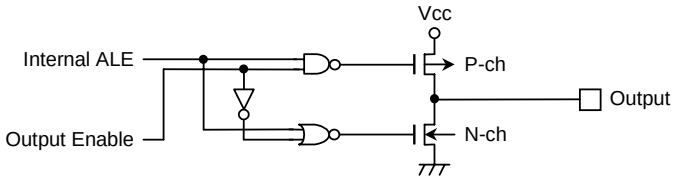
■ P96 (XT1), P97 (XT2)



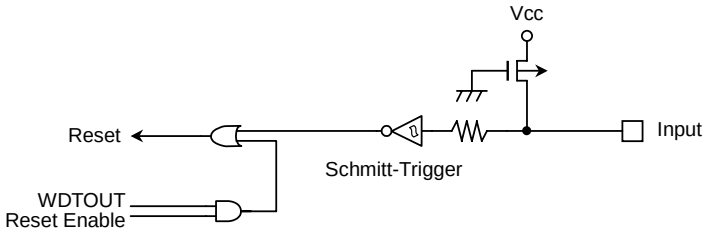
■ $\overline{\text{NMI}}$, BW0–BW1, $\overline{\text{PLLOFF}}$



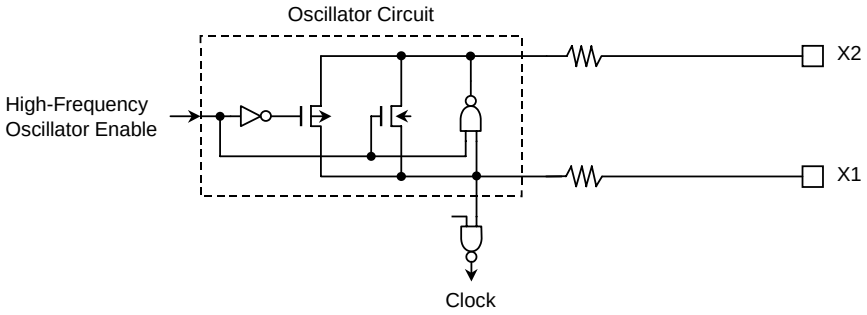
■ ALE



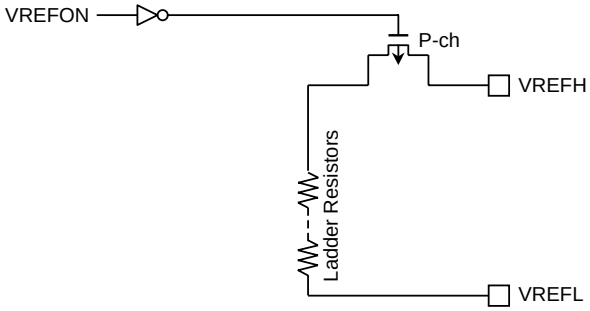
■ $\overline{\text{RESET}}$



■ X1, X2



■ VREFH, VREFL



21. Notations, Precautions and Restrictions

21.1 Notations and Terms

(1) I/O register fields are often referred to as <*register_mnemonic*>.<*field_name*> for the interest of brevity. For example, TA01RUN.TA0RUN means the TA0RUN bit in the TA01RUN register.

(2) fc, fs, fsys, state

fosc: Clock supplied from the X1 and X2 pins

fpll: Clock generated by the on-chip PLL

fc: Clock selected by the $\overline{\text{PLLOFF}}$ pin

fs: Clock supplied from the XT1 and XT2 pins

fgear: Clock selected by the SYSCR1.GEAR[1:0] bits

fsys: Clock selected by the SYSCR1.SYSCK bit

The fsys cycle is referred to as a state.

In addition, the clock selected by the SYSCR1.FPSEL bit and the prescalar clock source selected by the SYSCR0.PRCK[1:0] bits are referred to as fperiph and ϕT0 respectively.

21.2 Precautions and Restrictions

(1) Processor Revision Identifier

The Process Revision Identifier (PRId) register in the TX19 core of the TMP1940CYAF contains 0x0000_2C91.

(2) BW0–BW1 Pins

The BW0 and BW1 pins must be connected to the DVcc pin to ensure that their signal levels do not fluctuate during chip operation.

(3) Oscillator Warm-Up Counter

If an external crystal is utilized, an interrupt signal programmed to bring the TMP1940CYAF out of STOP mode triggers the on-chip warm-up counter. The system clock is not supplied to the on-chip logic until the warm-up counter expires.

(4) Programmable Pullup Resistors

When port pins are configured as input ports, the integrated pullup resistors can be enabled and disabled under software control. The pullup resistors are not programmable when port pins are configured as output ports.

The relevant port registers must be programmed by using store instructions.

(5) External Bus Mastership

The pin states while the bus is granted to an external device are described in Chapter 7, *I/O Ports*.

(6) Watchdog Timer (WDT)

Upon reset, the WDT is enabled. If the watchdog timer function is not required, it must be disabled after reset. When relevant pins are configured as bus arbitration signals, the I/O peripherals including the WDT can operate during external bus mastership.

(7) A/D Converter (ADC)

The ladder resistor network between the VREFH and VREFL pins can be disconnected under software control. This helps to reduce power dissipation, for example, in STOP mode.

(8) Undefined Bits in I/O Registers

Undefined I/O register bits are read as undefined states. Therefore, software must be coded without relying on the states of any undefined bits.

32-Bit RISC Microprocessor TX19 Family

TMP1940FDBF

1. Features

The TX19 is a family of high-performance 32-bit microprocessors that offers the speed of a 32-bit RISC solution with the added advantage of a significantly reduced code size of a 16-bit architecture. The instruction set of the TX19 includes as a subset the 32-bit instructions of the TX39, which is based on the MIPS R3000A™ architecture. Additionally, the TX19 supports the MIPS16 Application-Specific Extensions (ASE) for improved code density.

The TMP1940 is built on a TX19L core processor and a selection of intelligent peripherals. The TMP1940 is suitable for low-voltage, low-power applications.

Features of the TMP1940 include the following:

(1) TX19L core processor

1) Two instruction set architecture (ISA) modes: 16-bit ISA for code density and 32-bit ISA for speed
   - The 16-bit ISA is object-code compatible with the code-efficient MIPS16 ASE.
   - The 32-bit ISA is object-code compatible with the high-performance TX39 family.

2) Combines high performance with low power consumption.

— High performance
   - Single clock cycle execution for most instructions
   - 3-operand computational instructions for high instruction throughput
   - 5-stage pipeline
   - On-chip high-speed memory
   - DSP function: Executes 32-bit x 32-bit multiplier operations with a 64-bit accumulation in a single clock cycle.

— Low power consumption
   - Optimized design using a low-power cell library
   - Programmable standby modes in which processor clocks are stopped

3) Fast interrupt response suitable for real-time control
   - Distinct starting locations for each interrupt service routine
   - Automatically generated vectors for each interrupt source
   - Automatic updates of the interrupt mask level

(2) 16-Kbyte on-chip RAM
512-Kbyte on-chip flash

(3) External memory expansion
- 16-Mbyte off-chip address space for code and data
- External bus interface with dynamic bus sizing for 8-bit and 16-bit data ports

(4) 4-channel DMA controller
- Interrupt- or software-triggered

(5) 4-channel 8-bit timer

(6) 4-channel 16-bit timer

(7) 1-channel real-time counter (RTC)

(8) 4-channel general-purpose serial interface
Two channels support both UART and synchronous transfer modes and the other two channels are solely for UART.

(9) 1-channel serial bus interface
Either I^2C bus mode or clock-synchronous mode can be selected.

(10) 8-channel 10-bit A/D converter (with internal sample/hold)
Conversion time: 10.75 μs @32 MHz

(11) Watchdog timer

(12) 4-channel chip select/wait controller

(13) Interrupt sources
- 4 CPU interrupts: software interrupt instruction
- 32 internal interrupts: 7 priority levels, with the exception of the watchdog timer interrupt
- 11 external interrupts: 7 priority levels, with the exception of the NMI interrupt

(14) 77-pin input/output ports

(15) Four standby modes
- IDLE (HALT, DOZE), SLEEP, STOP

(16) Dual clocks
- Clock for low-power operation: Low-speed clock (32.768 kHz)
- RTC clock: Low-speed clock (32.768 kHz)

(17) Clock generator
- On-chip PLL (x4)
- Clock gear: Divides the operating speed of the CPU by 1/2, 1/4 or 1/8

(18) Little-endian

Higher address	31 24	23 16	15 8	7 0	Word address
↑	11	10	9	8	8
	7	6	5	4	4
Lower address	3	2	1	0	0

- Byte 0 is the lowest-order byte (bits 7-0).
- The address of a word data item is the address of its lowest-order byte (byte 0).

(19) Operating voltage range: 2.7 to 3.6 V

(20) Operating frequency

- 32 MHz (Vcc ≥ 3.0 V), with the flash memory in Interleave mode
- 26 MHz (Vcc ≥ 2.7 V), with the flash memory in Interleave mode

(21) Package

- 100-pin QFP (14 x 14 x 1.4 (t) mm, 0.5-mm pitch)

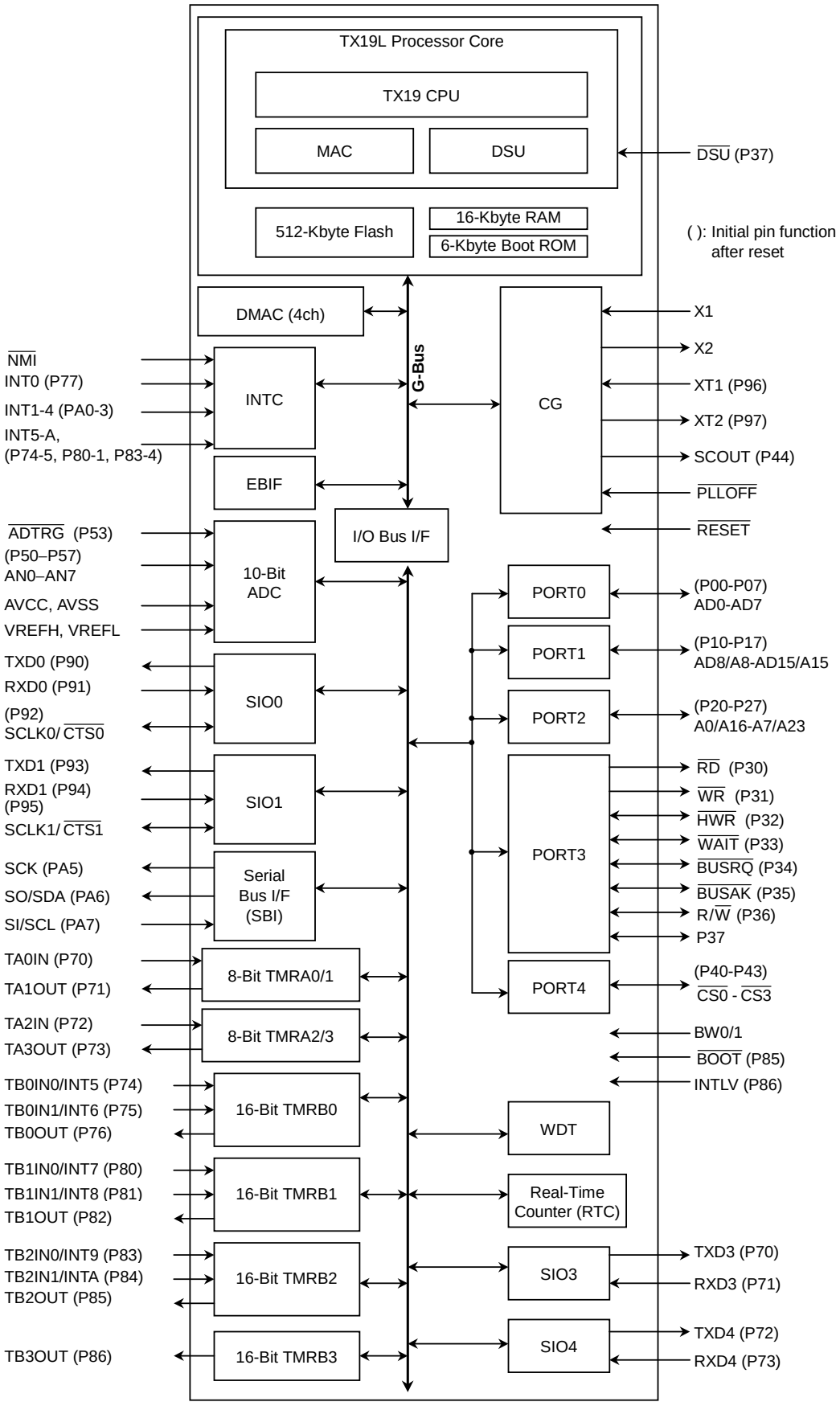


Figure 1.1 TMP1940FDBF Block Diagram

2. Signal Descriptions

This section contains pin assignments for the TMP1940FDBF as well as brief descriptions of the TMP1940FDBF input and output signals.

2.1 Pin Assignment

The following illustrates the TMP1940FDBF pin assignment.

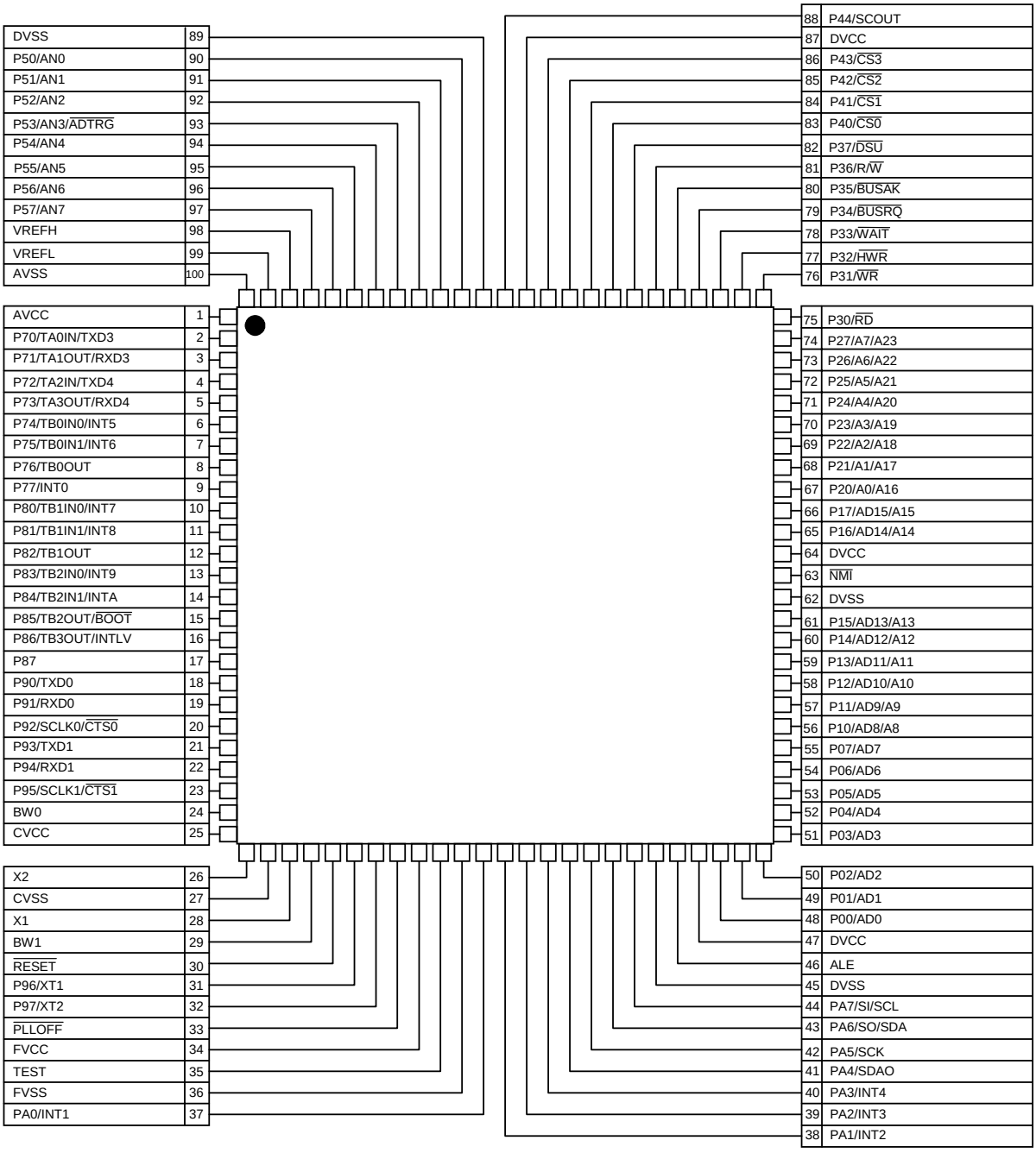


Figure 2.1 100-Pin LQFP Pin Assignment

2.2 Pin Usage Information

Table 2.1 lists the input and output pins of the TMP1940FDBF, including alternate pin names and functions for multi-function pins.

Table 2.1 Pin Names and Functions

Pin Name	# of Pins	Type	Function
P00–P07	8	Input/output	Port 0: Individually programmable as input or output
AD0–AD7		Input/output	Address (Lower): Bits 0-7 of the address/data bus
P10–P17	8	Input/output	Port 1: Individually programmable as input or output
AD8–AD15		Input/output	Address/Data (Upper): Bits 8-15 of the address/data bus
A8–A15		Output	Address: Bits 8-15 of the address bus
P20–P27	8	Input/output	Port 2: Individually programmable as input or output
A0–A7		Output	Address: Bits 0-7 of the address bus
A16–A23		Output	Address: Bits 16-23 of the address bus
P30	1	Output	Port 30: Output-only
$\overline{RD}$		Output	Read Strobe: Asserted during a read operation from an external memory device
P31	1	Output	Port 31: Output-only
$\overline{WR}$		Output	Write Strobe: Asserted during a write operation on D0-D7
P32	1	Input/output	Port 32: Programmable as input or output (with internal pull-up resister)
$\overline{HWR}$		Output	Higher Write Strobe: Asserted during a write operation on D8-D15
P33	1	Input/output	Port 33: Programmable as input or output (with internal pull-up resister)
$\overline{WAIT}$		Input	Wait: Causes the CPU to suspend external bus activity
P34	1	Input/output	Port 34: Programmable as input or output (with internal pull-up resister)
$\overline{BUSRQ}$		Input	Bus Request: Asserted by an external bus master to request bus mastership
P35	1	Input/output	Port 35: Programmable as input or output (with internal pull-up resister)
$\overline{BUSAK}$		Output	Bus Acknowledge: Indicates that the CPU has relinquished the bus in response to $\overline{BUSRQ}$.
P36	1	Input/output	Port 36: Programmable as input or output (with internal pull-up resister)
R / $\overline{W}$		Output	Read/Write: Indicates the direction of data transfer on the bus: 1 = read or dummy cycle, 0 = write cycle
P37	1	Input/output	Port 37: Programmable as input or output (with internal pull-up resister)
$\overline{DSU}$		Input	DSU Enable: If this pin is sampled low at the rising edge of $\overline{RESET}$, the TMP1940FDBF enters DSU mode for software debugging using an external real-time debug system. If this pin is sampled as high at the rising edge of $\overline{RESET}$, the TMP1940FDBF enters NORMAL mode.
P40	1	Input/output	Port 40: Programmable as input or output (with internal pull-up resister)
$\overline{CS0}$		Output	Chip Select 0: Asserted low to enable external devices at programmed addresses
P41	1	Input/output	Port 41: Programmable as input or output (with internal pull-up resister)
$\overline{CS1}$		Output	Chip Select 1: Asserted low to enable external devices at programmed addresses
P42	1	Input/output	Port 42: Programmable as input or output (with internal pull-up resister)
$\overline{CS2}$		Output	Chip Select 2: Asserted low to enable external devices at programmed addresses
P43	1	Input/output	Port 43: Programmable as input or output (with internal pull-up resister)
$\overline{CS3}$		Output	Chip Select 3: Asserted low to enable external devices at programmed addresses
P44	1	Input/output	Port 44: Programmable as input or output
SCOUT		Output	System Clock Output: Drives out a clock signal at the same frequency as the CPU clock (high-speed or low-speed)
P50–P57	8	Input	Port 5: Input-only
AN0–AN7		Input	Analog Input: Input to the on-chip A/D Converter
$\overline{ADTRG}$		Input	A/D Trigger: Starts an A/D conversion (multiplexed with P53)
P70	1	Input/output	Port 70: Programmable as input or output
TA0IN		Input	8-Bit Timer 0 Input: Input to Timer 0
TXD3		Output	Serial Transmit Data 3: Programmable as a push-pull or open-drain output
P71	1	Input/output	Port 71: Programmable as input or output
TA1OUT		Output	8-Bit Timer 1 Output: Output from either Timer 0 or Timer 1
RXD3		Input	Serial Receive Data 3

Pin Name	# of Pins	Type	Function
P72 TA2IN TXD4	1	Input/output Input Output	Port 72: Programmable as input or output 8-Bit Timer 2 Input: Input to Timer 2 Serial Transmit Data 4: Programmable as a push-pull or open-drain output
P73 TA3OUT RXD4	1	Input/output Output Input	Port 73: Programmable as input or output 8-Bit Timer 3 Output: Output from either Timer 2 or Timer 3 Serial Receive Data 4
P74 TB0IN0 INT5	1	Input/output Input Input	Port 74: Programmable as input or output 16-Bit Timer 0 Input 0: Count/capture trigger input to 16-bit Timer 0 Interrupt Request 5: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P75 TB0IN1 INT6	1	Input/output Input Input	Port 75: Programmable as input or output 16-Bit Timer 0 Input 1: Capture trigger input to 16-bit Timer 0 Interrupt Request 6: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P76 TB0OUT	1	Input/output Output	Port 76: Programmable as input or output 16-Bit Timer 0 Output: Output from 16-bit Timer 0
P77 INT0	1	Input/output Input	Port 77: Programmable as input or output Interrupt Request 0: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P80 TB1IN0 INT7	1	Input/output Input Input	Port 80: Programmable as input or output 16-Bit Timer 1 Input 0: Count/capture trigger input to 16-bit Timer 1 Interrupt Request 7: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P81 TB1IN1 INT8	1	Input/output Input Input	Port 81: Programmable as input or output 16-Bit Timer 1 Input 1: Capture trigger input to 16-bit Timer 1 Interrupt Request 8: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P82 TB1OUT	1	Input/output Output	Port 82: Programmable as input or output 16-Bit Timer 1 Output: Output from 16-bit Timer 1
P83 TB2IN0 INT9	1	Input/output Input Input	Port 83: Programmable as input or output 16-Bit Timer 2 Input 0: Count/capture trigger input to 16-bit Timer 2 Interrupt Request 9: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P84 TB2IN1 INTA	1	Input/output Input Input	Port 84: Programmable as input or output 16-Bit Timer 2 Input 1: Capture trigger input to 16-bit Timer 2 Interrupt Request A: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P85 TB2OUT $\overline{\text{BOOT}}$	1	Input/output Output Input	Port 85: Programmable as input or output 16-Bit Timer 2 Output: Output from 16-bit Timer 2 Single Boot Mode: If this pin is sampled low at the rising edge of $\overline{\text{RESET}}$, the TMP1940FDBF enters Single Boot mode for re-programming of the on-chip flash. If this pin is sampled high at the rising edge of $\overline{\text{RESET}}$, the TMP1940FDBF enters NORMAL mode.
P86 TB3OUT INTLV	1	Input/output Output Input	Port 86: Programmable as input or output 16-Bit Timer 3 Output: Output from 16-bit Timer 3 Interleave Mode: The TMP1940FDBF enters Interleave mode when this pin is sampled high at the rising edge of $\overline{\text{RESET}}$. During a reset sequence, this pin should be pulled up to a logic 1 when Interleave mode is used and pulled down to a logic 0 otherwise.
P87	1	Input/output	Port 87: Programmable as input or output This pin is used to select the operating mode during reset. This pin should be pulled down to a logic 0 during a reset sequence.
P90 TXD0	1	Input/output Output	Port 90: Programmable as input or output Serial Transmit Data 0: Programmable as a push-pull or open-drain output
P91 RXD0	1	Input/output Input	Port 91: Programmable as input or output Serial Receive Data 0

Pin Name	# of Pins	Type	Function
P92 SCLK0 $\overline{CTS0}$	1	Input/output Input/output Input	Port 92: Programmable as input or output Serial Clock Input/Output 0 Serial Clear-to-Send 0
P93 TXD1	1	Input/output Output	Port 93: Programmable as input or output Start Serial Transmit Data 1: Programmable as a push-pull or open-drain output
P94 RXD1	1	Input/output Input	Port 94: Programmable as input or output Serial Receive Data 1
P95 SCLK1 $\overline{CTS1}$	1	Input/output Input/output Input	Port 95: Programmable as input or output Serial Clock Input/Output 1 Serial Clear-to-Send 1
P96 XT1	1	Input/output Input	Port 96: Programmable as input or open-drain output Connection pin for a low-speed crystal
P97 XT2	1	Input/output Output	Port 97: Programmable as input or open-drain output Connection pin for a low-speed crystal
PA0–PA3 INT1–INT4	4	Input/output Input	Ports A0–A3: Individually programmable as input or output Interrupt Request 1–4: Individually programmable to be high-level, low-level, rising-edge or falling-edge sensitive
PA4	1	Input/output	Port A4: Programmable as input or output
PA5 SCK	1	Input/output Input/output	Port A5: Programmable as input or output Clock input/output pin when the Serial Bus Interface is in SIO mode
PA6 SO SDA	1	Input/output Output Input/output	Port A6: Programmable as input or output Data transmit pin when the Serial Bus Interface is in SIO mode Data transmit/receive pin when the Serial Bus Interface is in I^2C mode; programmable as a push-pull or open-drain output
PA7 SI SCL	1	Input/output Input Input/output	Port A7: Programmable as input or output Data receive pin when the Serial Bus Interface is in SIO mode Clock input/output pin when the Serial Bus Interface is in I^2C mode; as an output, programmable as a push-pull or open-drain output
ALE	1	Output	Address Latch Enable (This signal is driven out only when external memory is accessed.)
$\overline{NMI}$	1	Input	Nonmaskable Interrupt Request: Causes an NMI interrupt on the falling edge
BW0–1	2	Input	Both BW0 and BW1 should be tied to logic 1.
TEST	1	—	Test pin. This pin should be left open or tied to ground.
$\overline{PLLOFF}$	1	Input	This pin should be tied to logic 1 when the frequency multiplied clock from the PLL is used; otherwise, it should be tied to logic 0.
$\overline{RESET}$	1	Input	Reset (with internal pull-up resister): Initializes the whole TMP1940FDBF.
VREFH	1	Input	Input pin for high reference voltage for the A/D Converter. This pin should be connected to the AVCC pin when the A/D Converter is not used.
VREFL	1	Input	Input pin for low reference voltage for the A/D Converter. This pin should be connected to the AVSS pin when the A/D Converter is not used.
AVCC	1	—	Power supply pin for the A/D Converter. This pin should always be connected to power supply even when the A/D Converter is not used.
AVSS	1	—	Ground pin for the A/D Converter. This pin should always be connected to ground even when the A/D Converter is not used.
X1/X2	2	Input/output	Connection pins for a high-speed crystal
DVCC, CVCC, FVCC	5	—	Power supply pins
DVSS, CVSS, FVSS	5	—	Ground pins (0 V)

Note 1: When the Debug Support Unit (DSU) is enabled, all Port A pins function as DSU interface signals, regardless of the settings of the Port A Function Register (PAFC) and the Port A Control Register (PACR). Consequently, the Port A pins can not be configured as INT1–INT4 or Serial Bus Interface (SBI) pins.

Note 2: P37, P85, P86 and P87 should be held at the prescribed logic states for one system clock cycle before and after the rising edge of $\overline{RESET}$, with the $\overline{RESET}$ signal being stable in either logic state.

The following shows the DSU interface signals.

Figure 2.2 DSU Interface Signals

DSU Debug Interface If the $\overline{\text{DSU}}$ pin is sampled low at the rising edge of $\overline{\text{RESET}}$, the Port A pins are configured as interface signals for an external real-time debug system. The $\overline{\text{DSU}}$ pin has an internal pullup resistor.		
$\overline{\text{DRESET}}$ (PA7)	I	Debug Reset $\overline{\text{DRESET}}$ signal for an external real-time debug system
DCLK (PA0)	O	Debug Clock DCLK signal for an external real-time debug system
$\overline{\text{DBGE}}$ (PA5)	I	Debugger Enable $\overline{\text{DBGE}}$ signal for an external real-time debug system
PCST[2] (PA1)	O	PC Trace Status [2] PCTS[2] signal for an external real-time debug system
PCST[1] (PA2)	O	PC Trace Status [1] PCST[1] signal for an external real-time debug system
PCST[0] (PA3)	O	PC Trace Status [0] PCTS[0] signal for an external real-time debug system
SDI/ $\overline{\text{DINT}}$ (PA6)	I	Serial Data Input / Debug Interrupt SDI/ $\overline{\text{DINT}}$ signal for an external real-time debug system
SDAO/TPC (PA4)	O	Serial Data and Address Output / Target PC SDAO/TPC signal for an external real-time debug system

3. Flash Memory

This chapter describes the flash memory of the TMP1940FDBF, a flash version of the TMP1940CYAF. The TMP1940FDBF contains a 512-Kbyte flash EEPROM and 16-Kbyte RAM whereas the TMP1940CYAF contains a 256-Kbyte ROM and a 10-Kbyte RAM. In other respects, the hardware configuration and the functionality of the TMP1940FDBF are identical to those of the TMP1940CYAF. For descriptions of the on-chip I/O peripherals, refer to the TMP1940CYAF datasheet.

3.1 Features

(1) Organization

The TMP1940FDBF contains 4 Mbits (512 Kbytes) of flash memory, which is divided into a total of 19 blocks (fifteen 32-Kbyte, one 16-Kbyte, one 8-Kbyte and two 4-Kbyte blocks) to allow for independent protection from program and erase for each block. While the CPU can access information in the flash through a full 32-bit data bus, an external flash programmer can only perform 16-bit data bus writes to the flash.

(2) Access Types

The flash memory of the TMP1940FDBF provides two selectable access types: one-clock access and interleaved access.

(3) Program/Erase Time

- Chip programming time: 6 seconds (typ.), including program verify operations (20 μs per word)
- Chip erase time: 30 seconds (typ.), including erase verify operations

Note: These program and erase times are typical values and do not include data transfer overhead. The actual chip program and erase times depend on the programming method used.

(4) Programming Modes

Several options exist to program the TMP1940FDBF flash memory. On-Board Programming modes allow for re-programming of the flash memory while the chip is soldered on a printed circuit board. Programmer mode utilizes an EPROM programmer to perform code updates.

- On-Board Programming modes
  1) User Boot mode

     Supports use of a user-written programming algorithm.
  2) Single Boot mode

     Downloads new program code using a Toshiba-defined serial interface protocol.
- Programmer Mode

  Supports use of a general-purpose EPROM programmer.

  Toshiba recommends EPROM programmers from Minato Electronics, Inc. For questions pertaining to Minato's products, contact the following:

  Phone: +81-045-591-5605

  Fax: +81-045-592-2854

  URL: http://www.minato.co.jp/

(5) Re-programming

The TMP1940FDBF flash memory is compatible with the JEDEC standards, except a few unique functions. Thus, it is easy to migrate from a discrete flash memory device to the on-chip flash memory of the TMP1940FDBF. The TMP1940FDBF contains hardware to perform programming and erase

operations automatically. This eliminates the need for the user to code complex program and erase sequences.

The security feature of the TMP1940FDBF flash memory prevents the stored data from being read while it is being re-programmed with programming equipment. The TMP1940FDBF also allows the user to protect individual blocks of the flash memory against program or erase through software commands; however, 12-V VPP programming does not support data protection on a block-by-block basis.

JEDEC Standard	Changes and Enhancements
Auto Program Auto Chip Erase Auto Block Erase Auto Multi-Block Erase DATA Polling / Toggle Bit	Added feature: Security Auto Program Changed feature: Block protection is available only under software control. Removed feature: Erase Resume/Suspend mode

3.2 Block Diagram

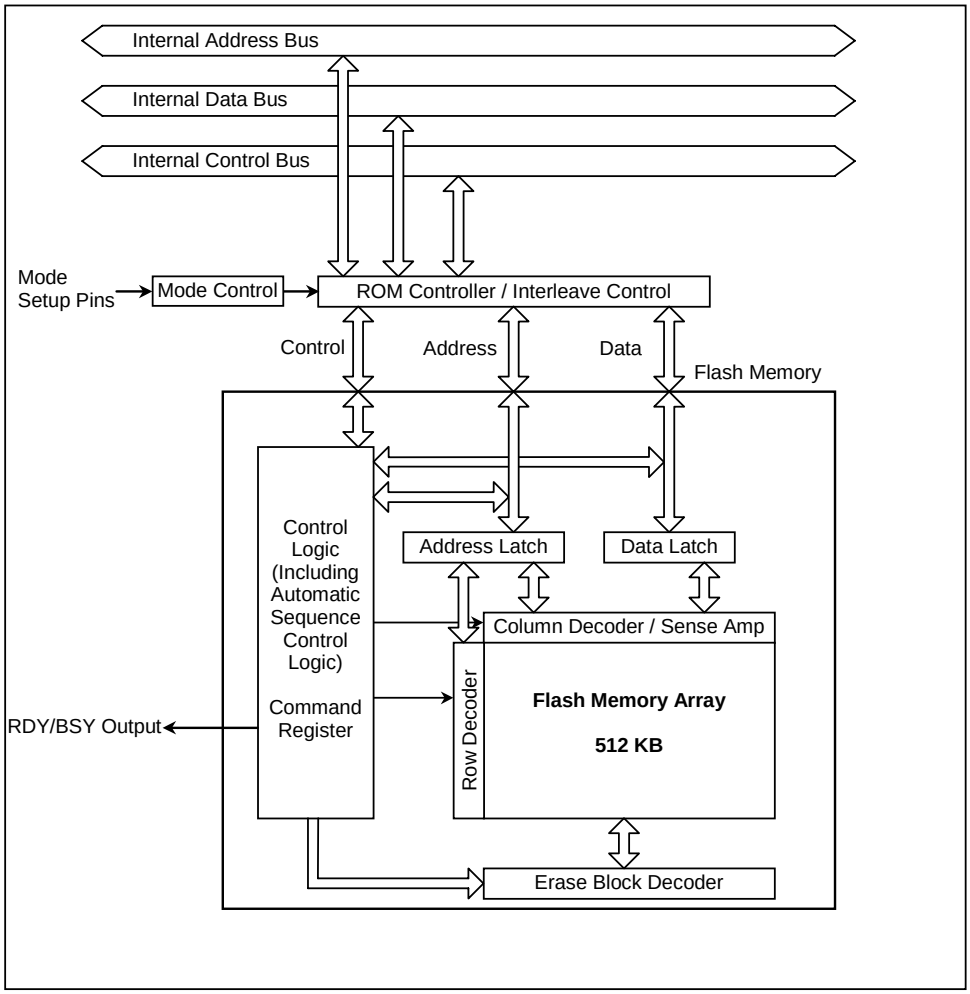


Figure 3.1 Flash Memory Block Diagram

3.3 Operating Modes

3.3.1 Overview

The TMP1940FDBF offers a total of five operating modes, including the one in which the flash memory is unused.

Table 3.1 Operating Modes

Operating Mode	Description
Single-Chip Mode	After a reset, the TX19 core processor executes out of the on-chip flash memory. Either fast (one-clock) or interleave mode operation is selected through the INTLV (P86) pin when $\overline{\text{RESET}}$ is released.
Normal Mode	Single-Chip mode is further divided into Normal mode in which the user application executes and User Boot mode which allows for re-programming of the flash memory while the TMP1940FDBF is installed on a printed circuit board.
User Boot Mode	The user can freely define how to switch between Normal mode and User Boot mode. For example, the logic state on, say, Port 00, can be used to determine whether to put the flash memory in Normal mode or User Boot mode. The user must include a routine in the application program to test the state of that port.
Single Boot Mode	After a reset, the TX19 core processor executes out of the on-chip boot ROM (which is a mask ROM). The boot ROM contains a routine to aid users in performing on-board programming of the flash memory via a serial port of the TMP1940FDBF. The serial port is connected to an external host which transfers new data according to a prescribed protocol.
Programmer Mode	This mode allows for re-programming of the flash memory with a general-purpose EPROM programmer. Use the programmer and programming adaptor recommended by Toshiba.

The on-chip flash memory can be re-programmed in one of the following three modes: User Boot mode, Single Boot mode and Programmer mode. Of these modes, User Boot mode and Single Boot mode are collectively referred to as on-board programming modes.

On-board programming modes allow for re-programming of the flash memory while the TMP1940FDBF is soldered on a printed circuit board. In Single Boot mode, new data comes from a serial port under control of a Toshiba-provided routine in the boot ROM. User Boot mode allows you to create an algorithm of your own for flash memory erase and program operations.

The TMP1940FDBF flash memory provides a security feature to prevent intrusive access to the flash memory while in Programmer mode. This security feature can be enabled upon completion of on-board programming to reduce the potential risk of software leaks to third parties.

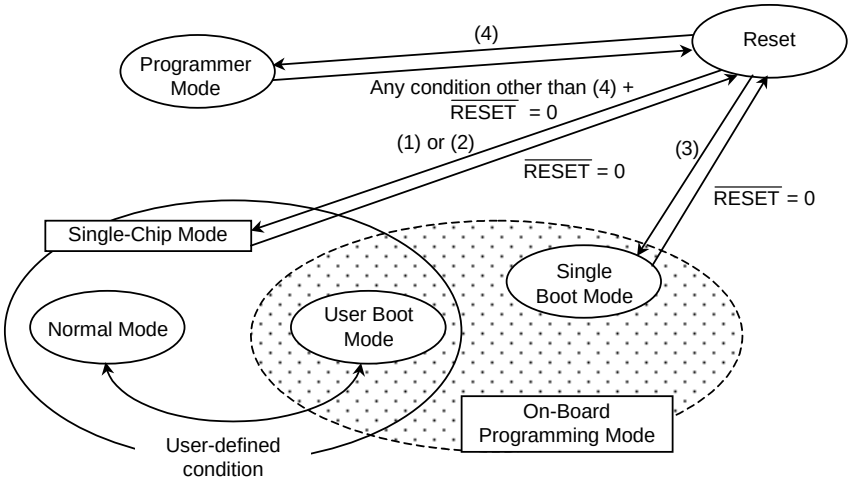
The logic states on the BW0, BW1, $\overline{\text{BOOT}}$ (P85) and INTLV (P86) pins during a reset sequence determine the mode of operation for the flash memory, as shown in Table 3.2. After $\overline{\text{RESET}}$ is released, P85 ($\overline{\text{BOOT}}$) and P86 (INTLV) can be configured as either general-purpose I/O pins or timer output pins.

After a reset, the CPU operates in compliance with the selected mode, except for Programmer mode. When Programmer mode is selected, $\overline{\text{RESET}}$ must be held at logic 0. The input pins listed in Table 3.2 must remain stable once the flash memory is put in a given mode of operation.

Table 3.2 Modes of Operation

#	Operating Mode	Input Pins				
		$\overline{\text{RESET}}$	BW0	BW1	$\overline{\text{RESET}}$	INTLV
(1)	Single-Chip Mode (Interleave)	0 → 1	1	1	1	1
(2)	Single-Chip Mode (Single-Clock)	0 → 1	1	1	1	0
(3)	Single Boot Mode	0 → 1	1	1	0	Note 1
(4)	Programmer Mode (Note 2)	0	0	1	Note 1	Note 1

Note 1: Don't care. The pins must be held at 1 or 0, however.
Note 2: Hold P40 at logic 1, and P41 and P42 at logic 0. 3.7.3 Pin Functions and Settings for a description of how other pins must be maintained in Programmer mode.



Parenthesized numbers indicate that the relevant pins are at the logic states shown in Table 3.2.

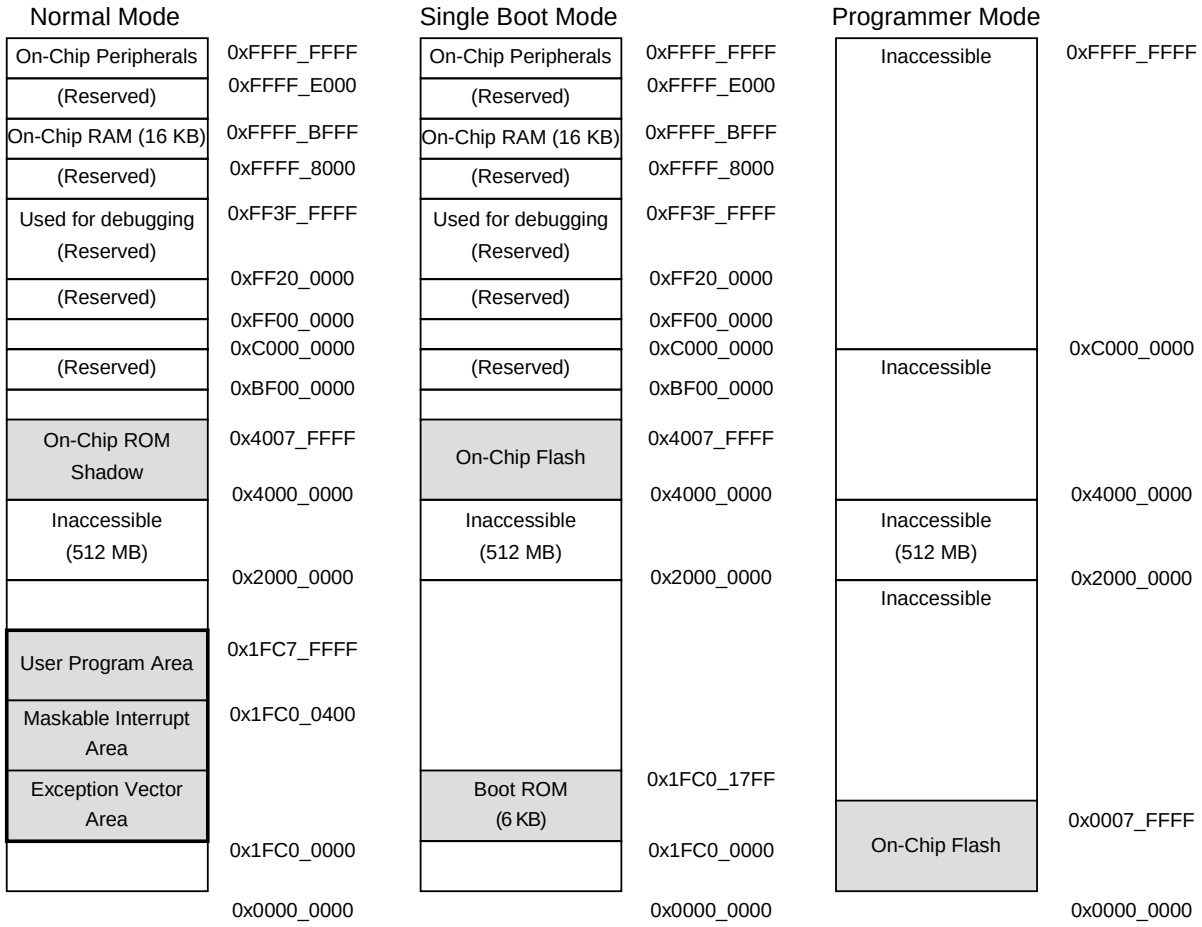
Figure 3.2 Mode Transitions

3.3.2 Reset Operation

To reset the TMP1940FDBF, $\overline{\text{RESET}}$ must be asserted for at least 12 system clock periods after the power supply voltage and the internal high-frequency oscillator have stabilized. This time is typically 3 μs at 32 MHz when the on-chip PLL is utilized. For a detailed description, see Section 3.1.1, *Reset Operation*, in the TMP1940CYAF datasheet.

3.3.3 Memory Maps

The memory map for the TMP1940FDBF varies according to the mode of operation selected for the on-chip flash memory. Following are the memory maps in each operating mode.



Note: The addresses shown above are physical addresses.

Figure 3.3 TMP1940FDBF Memory Maps

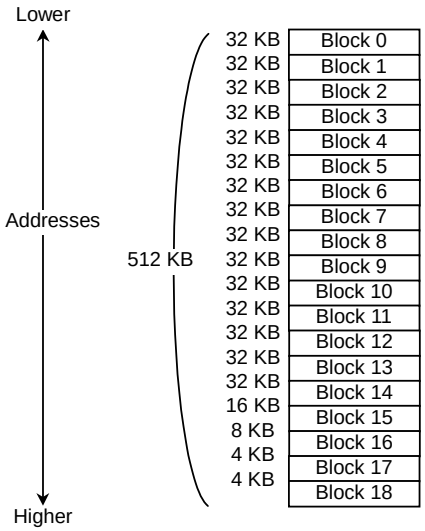


Figure 3.4 Flash Memory Block Architecture

Table 3.3 Block Addresses

	User Boot Mode	Single Boot Mode	Programmer Mode
Block 0	0x1FC0_0000 – 0x1FC0_7FFF (or 0x4000_0000 – 0x4000_7FFF)	0x1FC0_0000 – 0x1FC0_7FFF	0x0000_0000 – 0x0000_7FFF
Block 1	0x1FC0_8000 – 0x1FC0_FFFF (or 0x4000_8000 – 0x4000_FFFF)	0x1FC0_8000 – 0x1FC0_FFFF	0x0000_8000 – 0x0000_FFFF
Block 2	0x1FC1_0000 – 0x1FC1_7FFF (or 0x40010000 – 0x4001_7FFF)	0x1FC1_0000 – 0x1FC1_7FFF	0x0001_0000 – 0x0001_7FFF
Block 3	0x1FC1_8000 – 0x1FC1_FFFF (or 0x4001_8000 – 0x4001_FFFF)	0x1FC1_8000 – 0x1FC1_FFFF	0x0001_8000 – 0x0001_FFFF
Block 4	0x1FC2_0000 – 0x1FC2_7FFF (or 0x4002_0000 – 0x4002_7FFF)	0x1FC2_0000 – 0x1FC2_7FFF	0x0002_0000 – 0x0002_7FFF
Block 5	0x1FC2_8000 – 0x1FC2_FFFF (or 0x4002_8000 – 0x4002_FFFF)	0x1FC2_8000 – 0x1FC2_FFFF	0x0002_8000 – 0x0002_FFFF
Block 6	0x1FC3_0000 – 0x1FC3_7FFF (or 0x4003_0000 – 0x4003_7FFF)	0x1FC3_0000 – 0x1FC3_7FFF	0x0003_0000 – 0x0003_7FFF
Block 7	0x1FC3_8000 – 0x1FC3_FFFF (or 0x4003_8000 – 0x4003_FFFF)	0x1FC3_8000 – 0x1FC3_FFFF	0x0003_8000 – 0x0003_FFFF
Block 8	0x1FC4_0000 – 0x1FC4_7FFF (or 0x4004_0000 – 0x4004_7FFF)	0x1FC4_0000 – 0x1FC4_7FFF	0x0004_0000 – 0x0004_7FFF
Block 9	0x1FC4_8000 – 0x1FC4_FFFF (or 0x4004_8000 – 0x4004_FFFF)	0x1FC4_8000 – 0x1FC4_FFFF	0x0004_8000 – 0x000_4FFFF
Block 10	0x1FC5_0000 – 0x1FC5_7FFF (or 0x4005_0000 – 0x4005_7FFF)	0x1FC5_0000 – 0x1FC5_7FFF	0x0005_0000 – 0x0005_7FFF
Block 11	0x1FC5_8000 – 0x1FC5_FFFF (or 0x4005_8000 – 0x4005_FFFF)	0x1FC5_8000 – 0x1FC5_FFFF	0x0005_8000 – 0x0005_FFFF
Block 12	0x1FC6_0000 – 0x1FC6_7FFF (or 0x4006_0000 – 0x4006_7FFF)	0x1FC6_0000 – 0x1FC6_7FFF	0x0006_0000 – 0x0006_7FFF
Block 13	0x1FC6_8000 – 0x1FC6_FFFF (or 0x4006_8000 – 0x4006_FFFF)	0x1FC6_8000 – 0x1FC6_FFFF	0x0006_8000 – 0x0006_FFFF
Block 14	0x1FC7_0000 – 0x1FC7_7FFF (or 0x4007_0000 – 0x4007_7FFF)	0x1FC7_0000 – 0x1FC7_7FFF	0x0007_0000 – 0x0007_7FFF
Block 15	0x1FC7_8000 – 0x1FC7_BFFF (or 0x4007_8000 – 0x4007_BFFF)	0x1FC7_8000 – 0x1FC7_BFFF	0x0007_8000 – 0x0007_BFFF
Block 16	0x1FC7_C000 – 0x1FC7_DFFF (or 0x4007_C000 – 0x4007_DFFF)	0x1FC7_C000 – 0x1FC7_DFFF	0x0007_C000 – 0x0007_DFFF
Block 17	0x1FC7_E000 – 0x1FC7_EFFF (or 0x4007_E000 – 0x4007_EFFF)	0x1FC7_E000 – 0x1FC7_EFFF	0x0007_E000 – 0x0007_EFFF
Block 18	0x1FC7_F000 – 0x1FC7_FFFF (or 0x4007_F000 – 0x4007_FFFF)	0x1FC7_F000 – 0x1FC7_FFFF	0x0007_F000 – 0x0007_FFFF

3.3.4 Interleave Mode

If P86 is sampled high at the rising edge of $\overline{\text{RESET}}$, the flash memory enters Interleave mode. When the system clock (fsys) operates at 20 MHz or faster, the flash memory must be configured into Interleave mode.

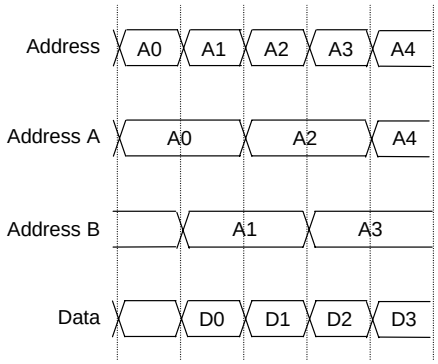


Figure 3.5 Interleave Mode

3.3.5 Block Protection

The TMP1940FDBF flash memory is organized into a total of 18 blocks (32 KB × 15, 16 KB × 1, 8 KB × 1, 4KB × 2). To protect stored data from any program and erase operations, each block has a protect bit, which can be set by executing the Block Protect command sequence. Blocks in protection mode are protected from even the Chip Erase and Multi-Block Erase commands; these commands erase only unprotected blocks. Since protection status is stored in flash memory cells, it is retained if the chip is powered off.

3.3.6 DSU-ICE Interface

If P37 is sampled low at the rising edge of $\overline{\text{RESET}}$, the TMP1940FDBF enters DSU mode, which is used for software debugging using an external DSU-ICE unit. In DSU mode, Port A serves as an interface to the DSU-ICE, and can not be used as general-purpose port, INT1–INT4 or Serial Bus Interface (SBI) pins. Consult the DSU-ICE operation manual for a description of debugging using the DSU-ICE. When the TMP1940FDBF is in DSU mode, the on-chip flash memory provides a security feature.

(1) Flash security feature

The TMP1940FDBF supports on-board debugging while it is installed on a printed circuit board. The TMP1940FDBF provides a security feature to prevent intrusive access to the flash memory. When the flash memory is in the secure state, a DSU-ICE is denied access to the entirety of the flash memory.

(2) Securing the flash (Disabling debugging with a DSU-ICE)

Once program debug is completed, set the FSE bit in the Flash Control/Status (FLCS) register (see section 3.6.14) and write the Auto Security On command. This turns on the flash security feature. While the flash memory is in the secure state, a DSU-ICE can not read its contents. When the chip is powered off and powered on again, the SEQON bit in the SEQMOD register is automatically set, which disables debugging using a DSU-ICE until the flash memory is unsecured.

(3) Unsecuring the flash (Enabling debugging with a DSU-ICE)

The flash memory may only be unsecured by clearing the SEQON bit in the SEQMOD register and then writing a special code (0x0000_00C5) to the Security Control (SEQCNT) register. This prevents runaway software from inadvertently turning off the security feature. Unsecuring the flash memory enables the DSU interface. The flash memory can be secured again by setting the SEQON bit in the SEQMOD and writing 0x0000_00C5 to the SEQCNT while the chip is powered.

SEQMOD (0xFFFF_E510)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	SEQON
Read/Write	—	—	—	—	—	—	—	R/W
Reset Value	—	—	—	—	—	—	—	1
Function								1: Security on 0: Security off

Note: This register must be read as a 32-bit quantity. Bits 1 to 31 are read as 0s.

SEQCNT (0xFFFF_E514)

	7	6	5	4	3	2	1	0
Name								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	15	14	13	12	11	10	9	8
Name								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	23	22	21	20	19	18	17	16
Name								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							
	31	30	29	28	27	26	25	24
Name								
Read/Write	W							
Reset Value								
Function	Must be written as 0x0000_00C5.							

Note: The security feature of the TMP1940FDBF flash memory is not intended to guarantee rigid security protection. In cases where security protection is of utmost importance, use the TMP1940CYAF that contains mask ROM.

(4) Application example

The following flowchart exemplifies how to use the security feature with a DSU-ICE.

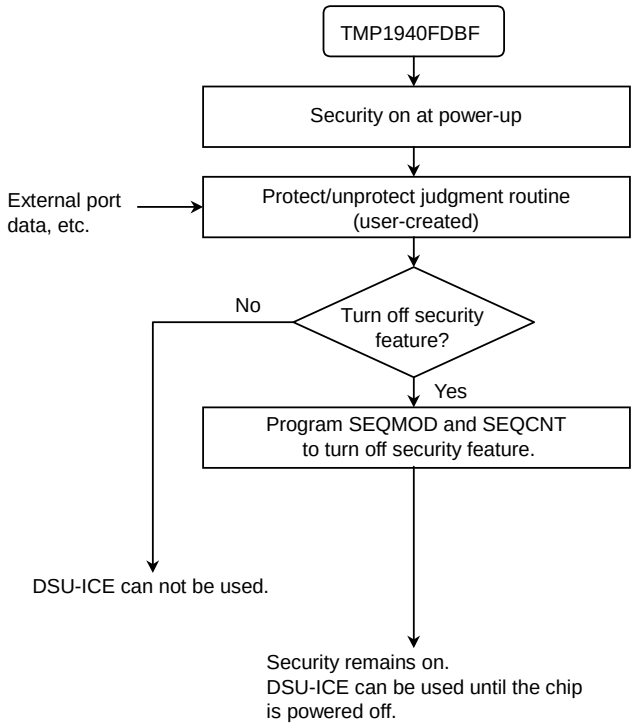


Figure 3.6 Using the Security Feature

3.4 User Boot Mode (Single-Chip Mode)

User Boot mode allows you to create a programming algorithm of your own. This mode supports situations where the flash memory is to be re-programmed via a bus other than serial I/O. User Boot mode is one of the two submodes in Single-Chip mode; the other submode is Normal mode in which the CPU executes the user application. To re-program the flash memory, the mode of operation must be switched from Normal mode to User Boot mode. The user application code must include a mode judgment routine as part of the reset procedure.

The user must define the conditions for mode switching, based on the logic states on I/O ports of the TMP1940FDBF. Additionally, the user must incorporate a programming algorithm into the user application code that is to be executed after User Boot mode is entered.

It is not possible to read from the flash memory while it is being erased or programmed; therefore, the programming algorithm must be placed and executed outside of the flash memory.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption.

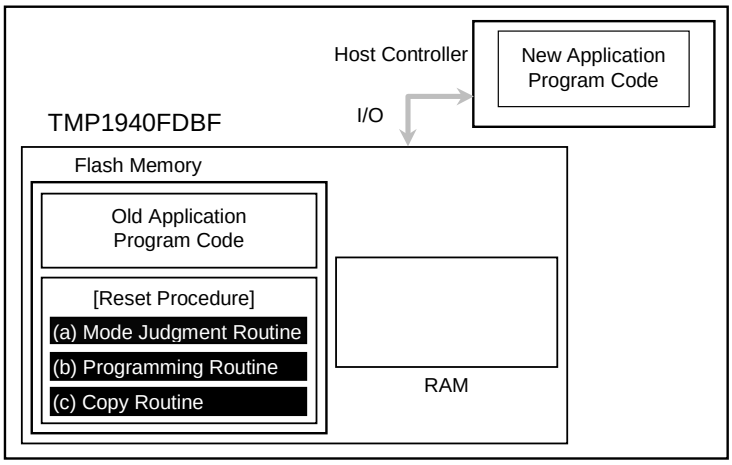
All interrupts including the nonmaskable (NMI) interrupt must be globally disabled while the flash memory is being erased or programmed.

The pages that follow describe the general procedures for two cases where the programming routine is: a) stored within the TMP1940FDBF flash memory, and b) loaded from an external controller. For a detailed description of the erase and program sequence, refer to Section 3.6, *On-Board Programming and Erasure.*

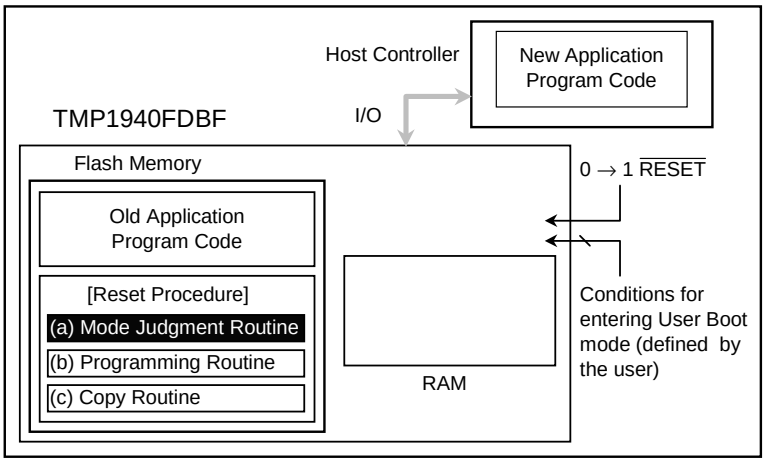
3.4.1 Method 1: Storing a Programming Routine in the Flash Memory

(1) Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP1940FDBF on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

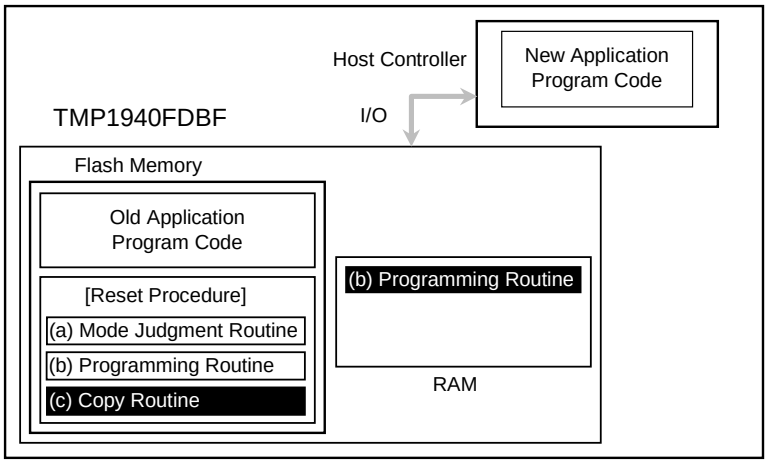
- Mode judgment routine: Code to determine whether or not to switch to User Boot mode
- Programming routine: Code to download new program code from a host controller and re-program the flash memory
- Copy routine: Code to copy the flash programming routine from the TMP1940FDBF flash memory to either the TMP1940FDBF on-chip RAM or external memory device.



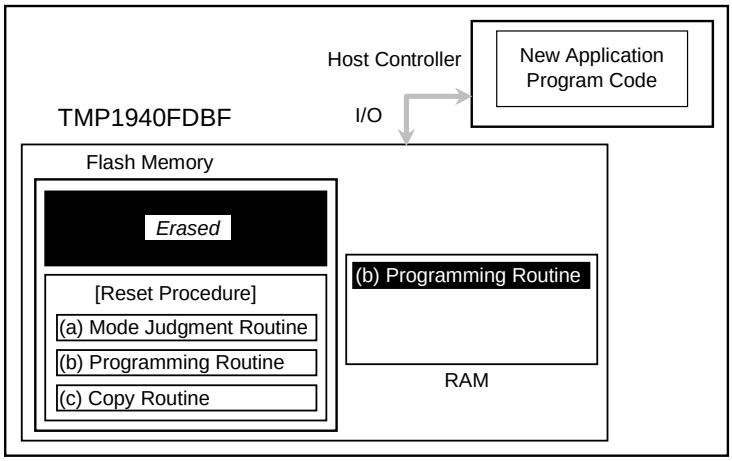
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP1940FDBF flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)



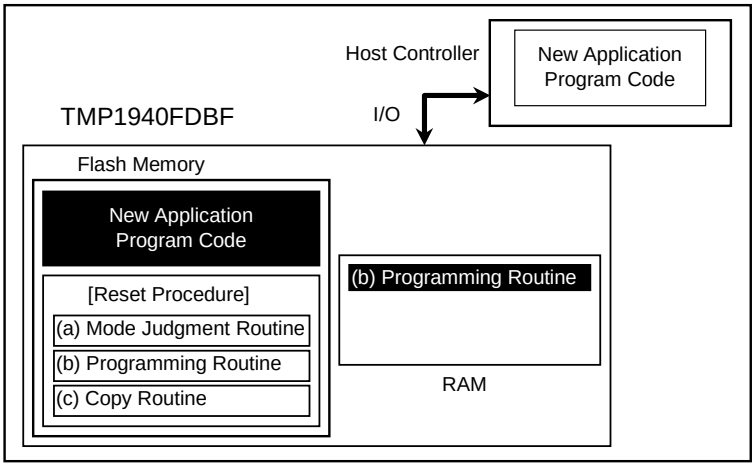
(3) Once User Boot mode is entered, execute the copy routine to copy the flash programming routine to either the TMP1940FDBF on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)



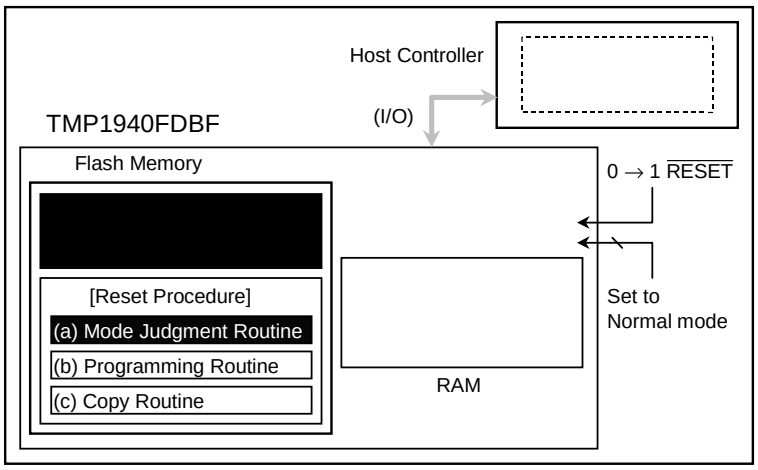
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(6) Drive $\overline{\text{RESET}}$ low to reset the TMP1940FDBF. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



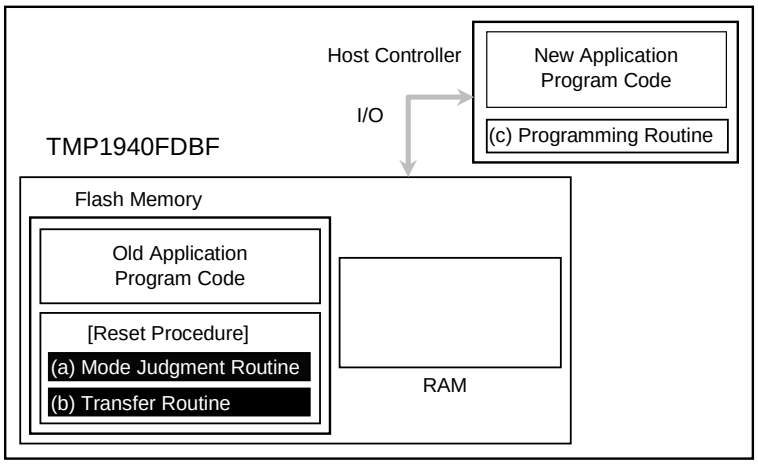
3.4.2 Method 2: Transferring a Programming Routine from an External Host

(1) Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP1940FDBF on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

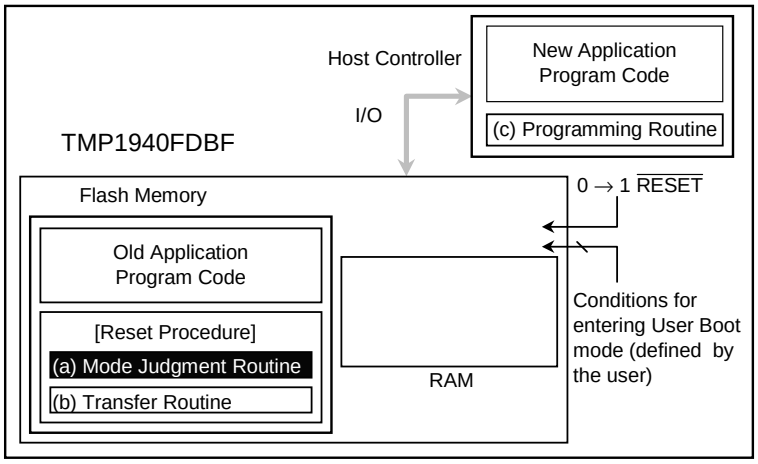
- Mode judgment routine: Code to determine whether or not to switch to User Boot mode
- Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine on the host controller:

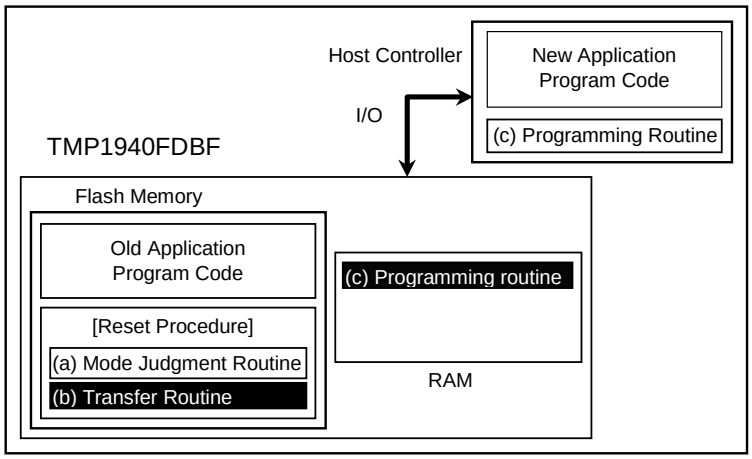
- Programming routine: Code to download new program code from an external host controller and re-program the flash memory



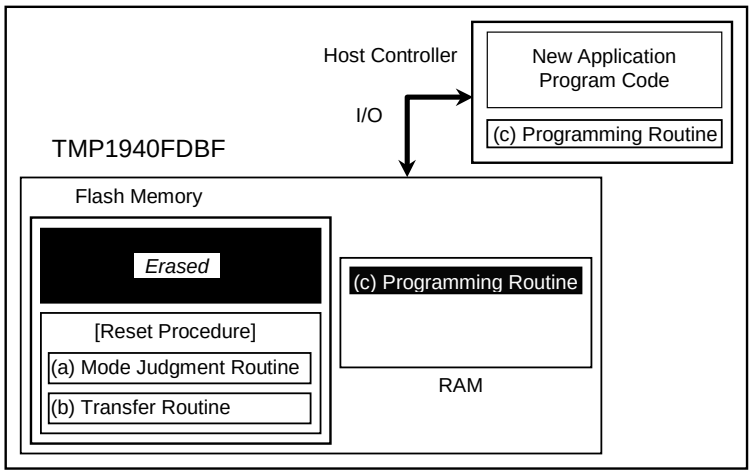
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP1940FDBF flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)



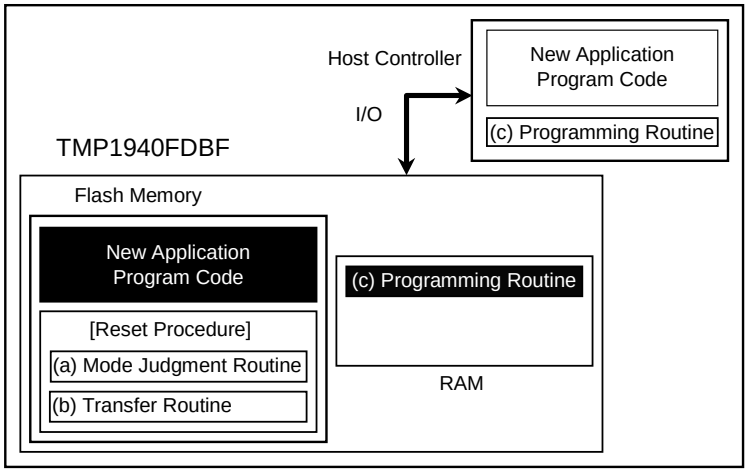
(3) Once User Boot mode is entered, execute the transfer routine to download the flash programming routine from the host controller to either the TMP1940FDBF on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)



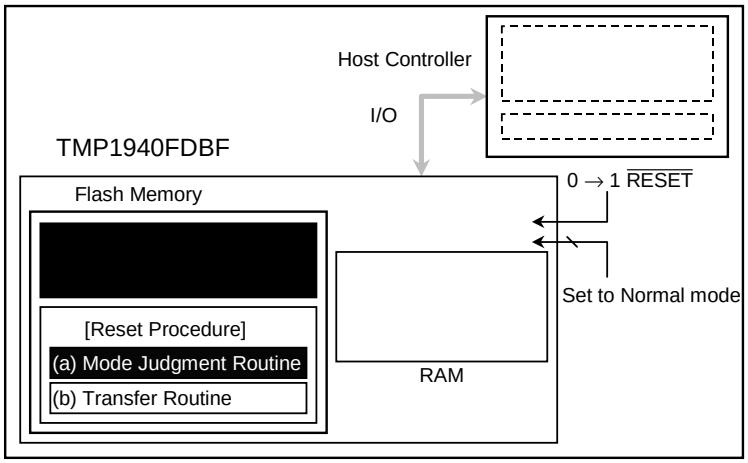
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(6) Drive $\overline{\text{RESET}}$ low to reset the TMP1940FDBF. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



3.5 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMP1940FDBF on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it (see Figure 3.3 on page 14).

Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMP1940FDBF is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMP1940FDBF on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory.

Communications between the SIO0 and the host must follow the prescribed protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

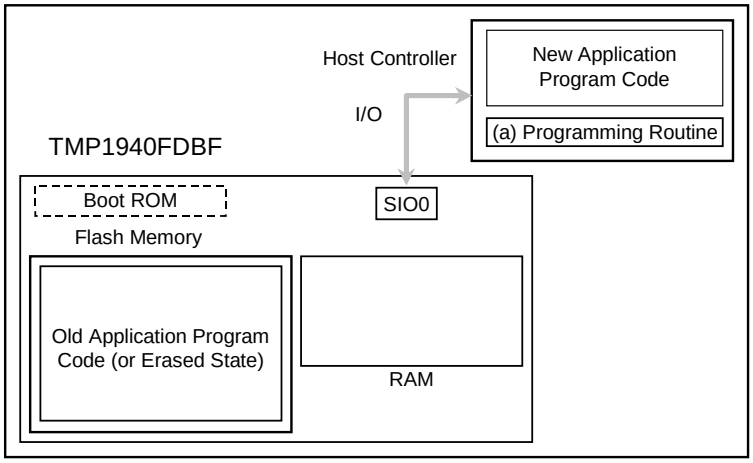
When any on-chip peripherals are utilized in Single Boot mode (such as the SIO), all interrupts must be globally disabled. Even in that case, occurrences of otherwise interrupt-causing events are recorded in the Interrupt Vector Register (IVR). For example, the SIO receive/transmit status can be checked via the IVR. The NMI interrupt must also be disabled.

Note: In Single Boot mode, the boot-ROM programs are executed in Normal mode. Don't change the mode in the programming routine.

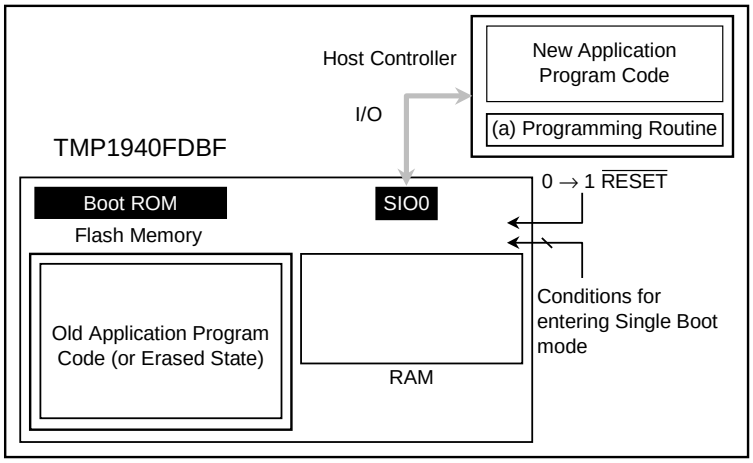
Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. For a detailed description of the erase and program sequence, refer to Section *On-Board Programming and Erasure.*

3.5.1 General Procedure: Using the Program in the On-Chip Boot ROM

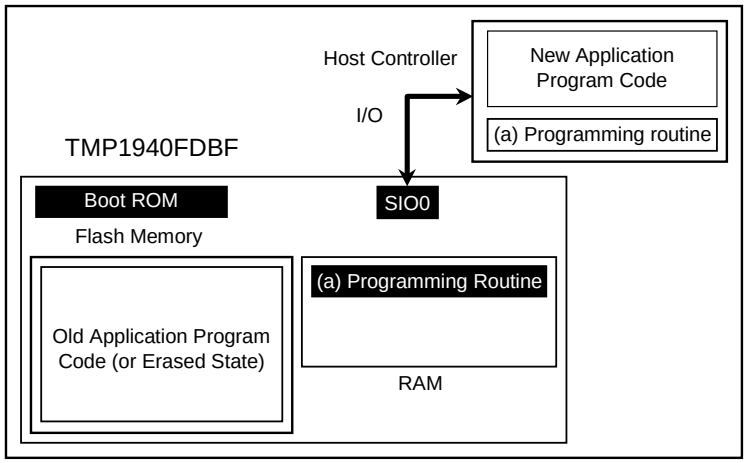
(1) The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO0, the SIO0 must be connected to a host controller. Prepare a programming routine on the host controller.



(2) Reset the TMP1940FDBF with the mode setting pins held at appropriate logic values, so that the CPU re-boots from the on-chip boot ROM. The 12-byte password transferred from the host controller is first compared to the contents of special flash memory locations. (If the flash block has already been erased, the password is 0xFFFF.)

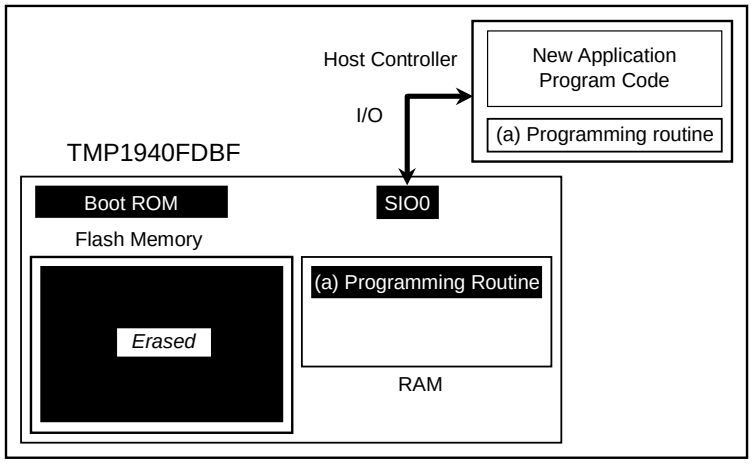


(3) If the password was correct, the boot program downloads, via the SIO0, the programming routine from the host controller into the on-chip RAM of the TMP1940FDBF. The programming routine must be stored in the address range 0xFFFF_8000 – 0xFFFF_8FFF.



Note: At this point, r29 (sp) points to address 0xFFFF_9100.

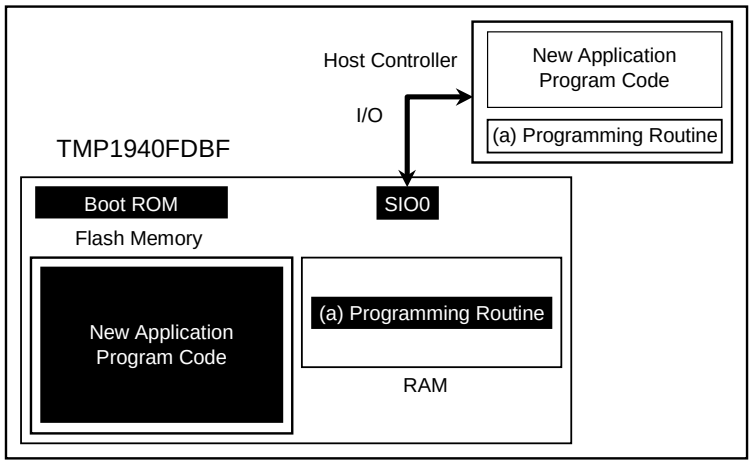
(4) The CPU jumps to the programming routine in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



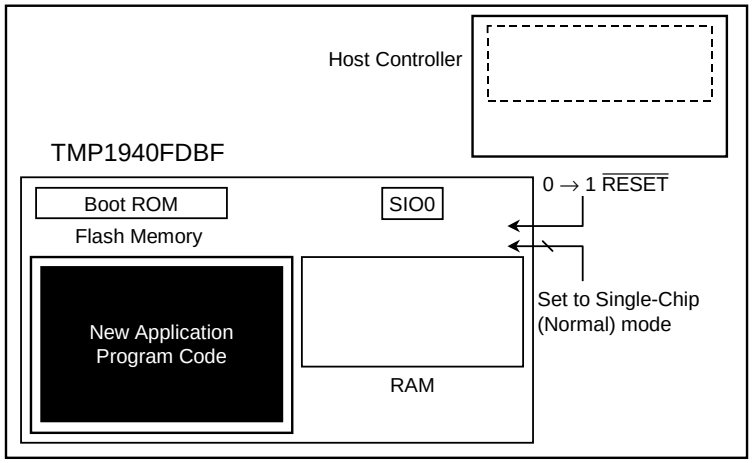
(5) Next, the programming routine downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on.

It is not allowed to move program control from the programming routine back to the boot ROM.

In the example below, new program code comes from the same host controller via the same SIO channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMP1940FDBF re-boots in Single-Chip (Normal) mode to execute the new program.



3.5.2 Host-to-Target Connection Examples

In Single Boot mode, serial transfer is used to re-program the flash memory while the TMP1940FDBF is installed on the board. In this mode, channel 0 of the SIO (SIO0) of the TMP1940FDBF is connected to a host controller, which is to issue commands to the target board. Figure 3.7 and Figure 3.8 show examples of host-to-target connections.

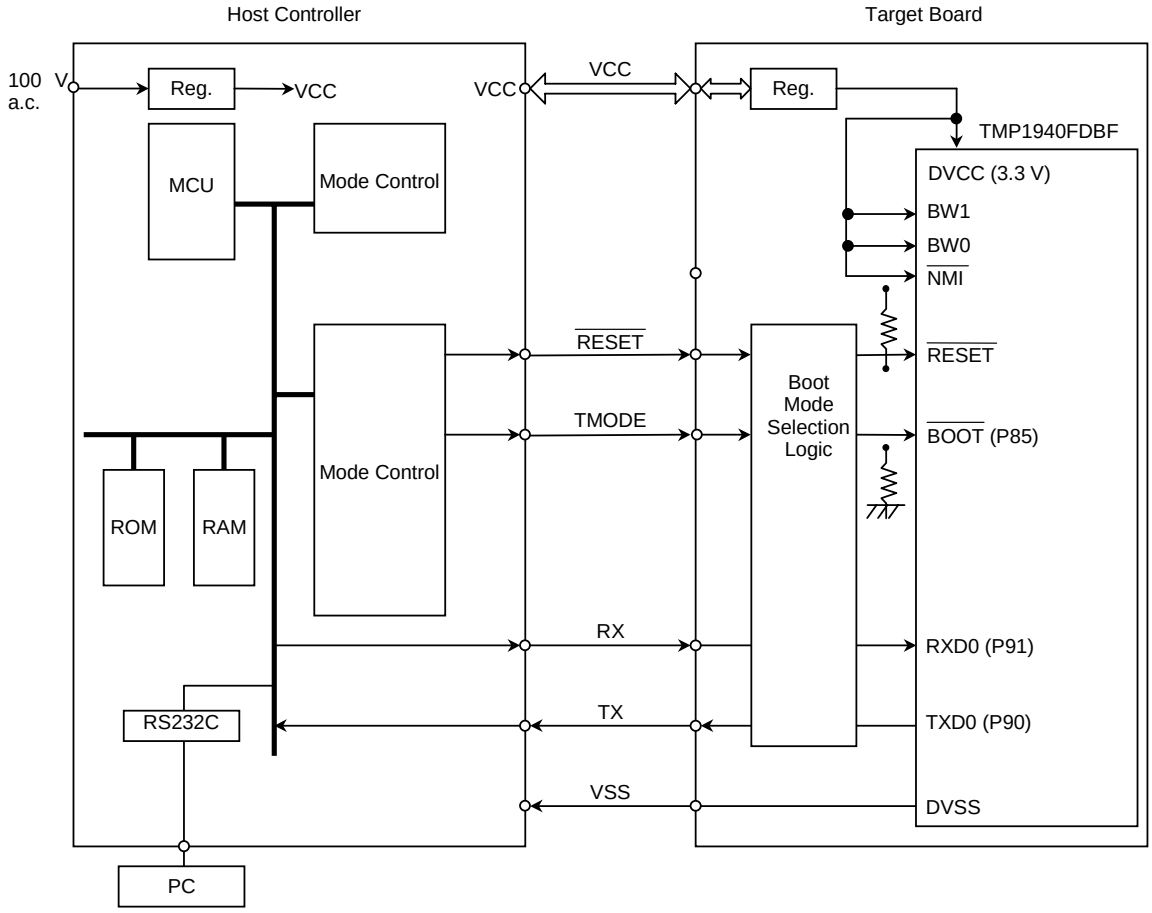


Figure 3.7 Example of a Connection Between a Host Controller and a Target Board (When the SIO0 is Configured for UART Mode)

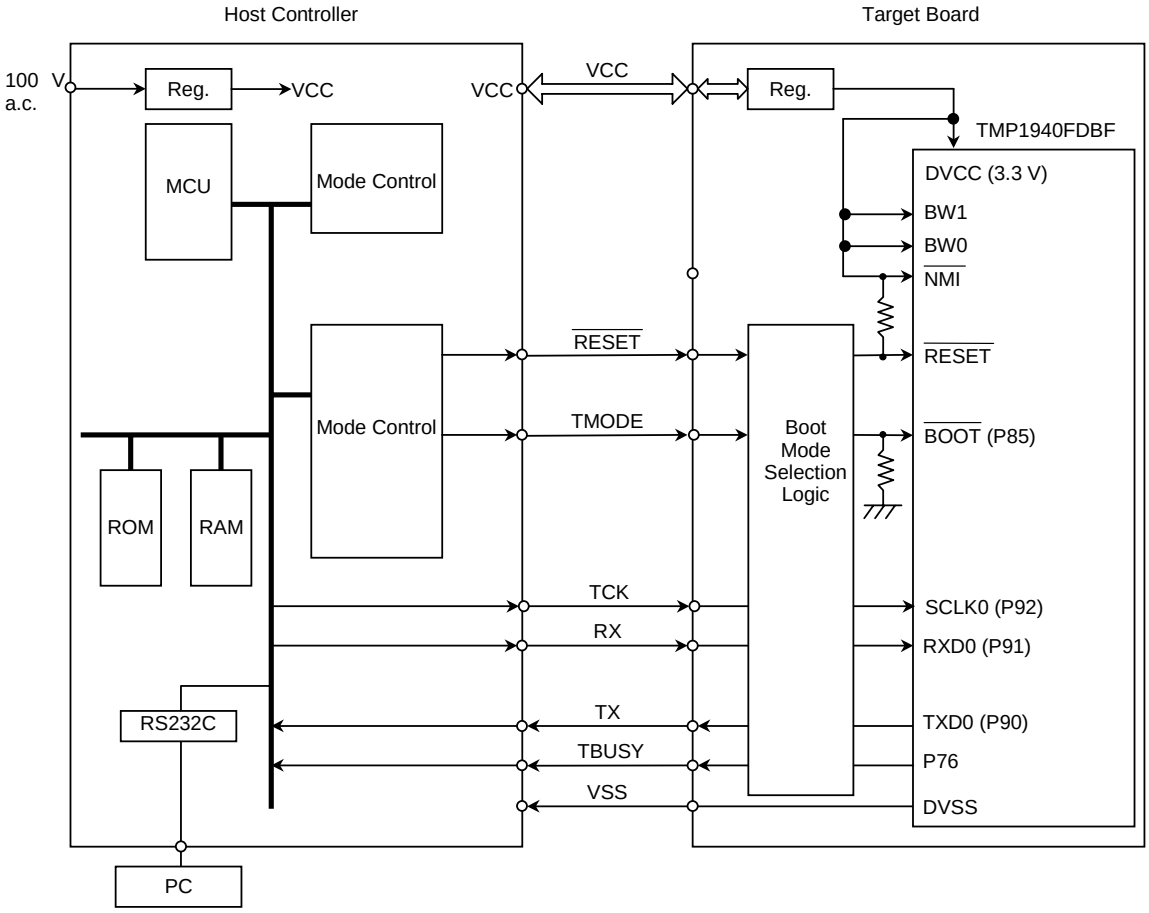


Figure 3.8 Example of a Connection Between a Host Controller and a Target Board (When the SIO0 is Configured for I/O Interface Mode)

The NET IMPRESS controller from Yokogawa Digital Computer Corporation is supported. For a detailed description, consult the manual that accompanies NET IMPRESS.

Note: When using NET IMPRESS, the $\overline{\text{RESET}}$ pin of the TMP1940FDBF must be pulled high with a resistor of 10 kΩ.

Contact: Yokogawa Digital Computer Corporation
Instruments Business Division
Phone: +81-42-333-6224
Fax: +81-42-352-6107
URL: http://www.ydc.co.jp/micom/

3.5.3 Configuring for Single Boot Mode

For on-board programming, boot the TMP1940FDBF in Single Boot mode, as follows:

BW0 = 1

BW1 = 1

$\overline{\text{BOOT}}$ (P85) = 0

$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input at logic 0, and the BW0, BW1 and $\overline{\text{BOOT}}$ (P85) inputs at the logic values shown above, and then release $\overline{\text{RESET}}$ (high).

3.5.4 Memory Map

Figure 3.9 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the on-chip flash memory is mapped to physical addresses 0x4000_0000 through 0x4007_FFFF, and the on-chip boot ROM is mapped to physical addresses 0x1FC0_0000 through 0x1FC0_17FF.

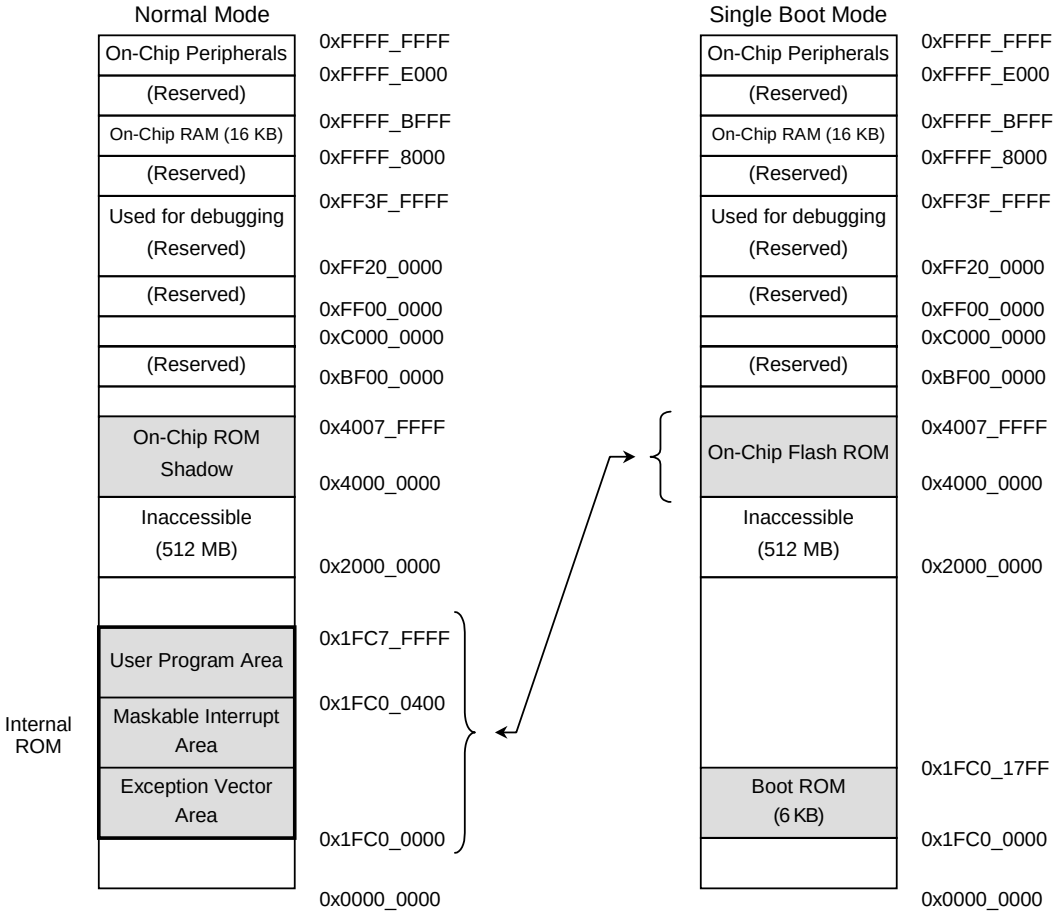


Figure 3.9 Memory Maps for Normal and Single Boot Modes (Physical Addresses)

3.5.5 Interface Specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below. In the subsections that follow, virtual addresses are indicated, unless otherwise noted.

- UART mode

  Communications channel: SIO Channel 0 (SIO0)
  Transfer mode: UART (asynchronous) mode, full-duplex
  Data length: 8 bits
  Parity bits: None
  STOP bits: 1
  Baud rate: See Table 3.13 on page 50.

- I/O Interface mode

  Communications channel: SIO Channel 0 (SIO0)
  Transfer mode: I/O Interface mode, half-duplex
  Synchronization clock (SCLK0): Input
  Handshaking signal: P76 configured as an output
  Baud rate: See Table 3.13 on page 50.

Table 3.4 Required Pin Connections

Pin		Interface	
		UART Mode	I/O Interface Mode
Power Supply Pins	DVCC (3.3 V)	Required	Required
	DVSS	Required	Required
Mode-Setting Pin	$\overline{\text{BOOT}}$	Required	Required
Reset Pin	$\overline{\text{RESET}}$	Required	Required
Communications Pins	TXD0	Required	Required
	RXD0	Required	Required
	SCLK0	Not Required	Required (Input Mode)
	P76	Not Required	Required (Input Mode)

I/O Interface mode uses a simple handshaking protocol, which is shown in Figure 3.10. The boot program clears the RXE bit in the SC0MOD0 register, disabling data reception via the SIO0.

The host controller must communicate with the TMP1940FDBF, using the P76 pin for handshaking. The following enumerates the steps for the TMP1940FDBF to receive and transmit data from/to a host controller.

For receive:

(1) As shown in Figure 3.10, set the RXE bit in the SC0MOD0 register to enable reception and bring the P76 pin high to inform the controller that the TMP1940FDBF is ready to communicate. Then, wait for the SCLK0 signal to come from the controller.

(2) When the SIO0 has received a byte of data, the SC0MOD0.RXE bit is automatically cleared to disable reception until the data is picked up by the CPU and the receive interrupt request is cleared. At this time, bring P76 low to indicate to the controller that the TMP1940FDBF is not ready to receive or transmit the next byte. When the TMP1940FDBF is ready and if the next action of the boot program is again a reception, set the SC0MOD0.RXE bit, bring P76 high and wait for an active SCLK0 edge to come from the controller.

(3) The controller must perform the next action after a high-to-low transition occurs on P76.

Note: The wait period required until P76 is allowed to go low after the seventh rising edge of SCLK0 differs, depending on the operating frequency and the baud rate.

For transmit:

(1) Load the SC0BUF register with the transmit data, bring P76 high and wait for the SCLK0 signal to come from the controller.

(2) When the TMP1940FDBF has sent out a byte of data and generated a transmit-done interrupt request, bring P76 low to indicate to the controller that it is not ready to transmit or receive the next data. When the transmit-done interrupt is cleared and if the next action of the boot program is again a transmission, load the SC0BUF register with the next data and bring P76 high to inform the controller that the TMP1940FDBF is now ready to transmit the next data. Then, wait for the SCLK0 signal to come from the controller. If the next action of the boot program is a reception, follow the steps described above.

(3) The controller must perform the next action after a high-to-low transition occurs on P76.

Note: The wait period required until P76 is allowed to go low after the seventh rising edge of SCLK0 differs, depending on the operating frequency and the baud rate.

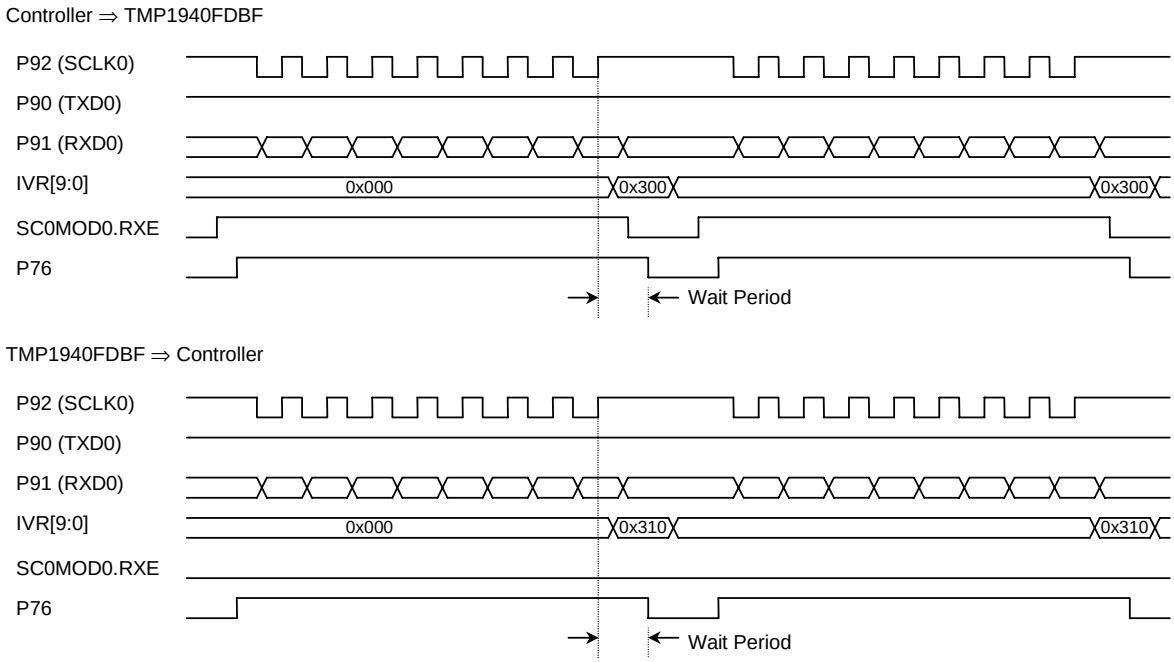


Figure 3.10 Handshake Protocol in I/O Interface Mode

3.5.6 Data Transfer Format

The host controller is to issue one of the commands listed in Table 3.5 to the target board. Table 3.6 to Table 3.8 illustrate the sequence of two-way communications that should occur in response to each command.

Table 3.5 Single Boot Mode Commands

Code	Command
10H	RAM Transfer
20H	Show Flash Memory Sum
30H	Show Product Information
40H	Reserved

Table 3.6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMP1940FDBF	Baud Rate	Data Transferred from the TMP1940FDBF to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate is can not be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (10H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 10H Negative acknowledge x1H Communication error x8H
	5th byte thru 16th byte	Password sequence (12 bytes) (0x0000_03F4 thru 0x0000_03FF)		—
	17th byte	Checksum value for bytes 5–16		—
	18th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	19th byte	RAM storage start address (bits 31–24)		—
	20th byte	RAM storage start address (bits 23–16)		—
	21st byte	RAM storage start address (bits 15–8)		—
	22nd byte	RAM storage start address (bits 7–0)		—
	23rd byte	RAM storage byte count (bits 15–8)		—
	24th byte	RAM storage byte count (bits 7–0)		—
	25th byte	Checksum value for bytes 19–24		—
	26th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	27th byte thru mth byte	RAM storage data		—
	(m + 1)th byte	Checksum value for bytes 27–m		—
	(m + 2)th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Non-acknowledge 11H Communications error 18H
RAM	(m + 3)th byte	—		Jump to RAM storage start address

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Note 3: The 19th to 25th bytes must be within the RAM address range 0xFFFF_8000–0xFFFF_8FFF.

Table 3.7 Transfer Format for the Show Flash Memory Sum Command

	Byte	Data Transferred from the Controller to the TMP1940FDBF	Baud Rate	Data Transferred from the TMP1940FDBF to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate is can not be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (20H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 20H Negative acknowledge x1H Communication error x8H
	5th byte	—		SUM (upper byte)
	6th byte	—		SUM (lower byte)
	7th byte	—		Checksum value for bytes 5 and 6
	8th byte	(Wait for the next command code.)		—

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Table 3.8 Transfer Format for the Show Product Information Command (1 of 2)

	Byte	Data Transferred from the Controller to the TMP1940FDBF	Baud Rate	Data Transferred from the TMP1940FDBF to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate is can not be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (30H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 30H Negative acknowledge x1H Communication error x8H
	5th byte	—		Flash memory data (at address 0x0000_03F0H)
	6th byte	—		Flash memory data (at address 0x0000_03F1H)
	7th byte	—		Flash memory data (at address 0x0000_03F2H)
	8th byte	—		Flash memory data (at address 0x0000_03F3H)
	9th byte thru 20th byte	—		Product name (12-byte ASCII code) “TX1940FDAF_ _” from the 9th byte
	21st byte thru 24th byte	—		Password comparison start address (4 bytes) F4H, 03H, 00H and 00H from the 21st byte
	25th byte thru 28th byte	—		RAM start address (4 bytes) 00H, 80H, FFH and FFH from the 25th byte
	29th byte thru 32nd byte	—		Dummy data (4 bytes) FFH, 8FH, FFH and FFH from the 29th byte
	33rd byte thru 36th byte	—		RAM end address (4 bytes) FFH, BFH, FFH and FFH from the 33rd byte
	37th byte thru 40th byte	—		Dummy data (4 bytes) 00H, 91H, FFH and FFH from the 37th byte
	41st byte thru 44th byte	—		Dummy data (4 bytes) FFH, AFH, FFH and FFH from the 41st byte
	45th byte thru 46th byte	—		Fuse information (2 bytes) 00H and 00H from the 45th byte
	47th byte thru 50th byte	—		Flash memory start address (4 bytes) 00H, 00H, 00H and 00H from the 47th byte
	51st byte thru 54th byte	—		Flash memory end address (4 bytes) FFH, FFH, 07H and 00H from the 51st byte
	55th byte thru 56th byte	—		Flash memory block count (2 bytes) 13H and 00H from at the 55th byte

Transfer Format for the Show Product Information Command (2 of 2)

	Byte	Data Transferred from the Controller to the TMP1940FDBF	Baud Rate	Data Transferred from the TMP1940FDBF to the Controller
Boot ROM	57th byte thru 60th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, 00H, 00H and 00H from the 57th byte
	61st byte thru 64th byte	—		Size (in halfwords) of the same-size flash blocks (4 bytes) 00H, 40H, 00H and 00H from the 61st byte
	65th byte	—		Number of flash blocks of the same size (1 byte) 0FH
	66th byte thru 69th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, 80H, 07H and 00H from the 66th byte
	70th byte thru 73rd byte	—		Size (in halfwords) of the same-size flash blocks (4 bytes) 00H, 20H, 00H and 00H from the 70th byte
	74th byte	—		Number of flash blocks of the same size (1 byte) 01H
	75th byte thru 78th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, C0H, 07H and 00H from the 75th byte
	79th byte thru 82nd byte	—		Size (in halfwords) of the same-size flash blocks (4 bytes) 00H, 10H, 00H and 00H from the 79th byte
	83rd byte	—		Number of flash blocks of the same size (1 byte) 01H
	84th byte thru 87th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, E0H, 07H and 00H from the 84th byte
	88th byte thru 91st byte	—		Size (in halfwords) of the same-size flash blocks (4 bytes) 00H, 08H, 00H and 00H from the 88th byte
	92nd byte	—		Number of the flash blocks of the same size (1 byte) 02H
	93rd byte	—		Checksum value for bytes 5 to 92
	94th byte	(Wait for the next command code.)		—

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

3.5.7 Overview of the Boot Program Commands

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these three commands, the details of which are provided on the following subsections.

- RAM Transfer command

  The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The maximum program size is 4 Kbytes. The RAM storage start address must be within the range 0xFFFF_8000–0xFFFF_8FFF.

  The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 3.6.16.

Before initiating a transfer, the RAM Transfer command checks a password sequence coming from the controller against that stored in the flash memory. If they do not match, the RAM Transfer command aborts.

Once the RAM Transfer command is complete, the whole on-chip RAM is accessible.

- Show Flash Memory Sum command

  The Show Flash Memory Sum command adds the contents of the 512 Kbytes of the flash memory together. The boot program does not provide a command to read out the contents of the flash memory. Instead, the Flash Memory Sum command can be used for software revision management.

- Show Product Information command

  The Show Product Information command provides the product name, on-chip memory configuration and the like. This command also reads out the contents of the flash memory locations at addresses 0x0000_03F0 through 0x0000_03F3. In addition to the Show Flash Memory Sum command, these locations can be used for software revision management.

3.5.8 RAM Transfer Command

See Table 3.6.

(1) The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see Section 3.5.12. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD0 register is cleared.

- To communicate in UART mode

  Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

- To communicate in I/O Interface mode

  Send, from the controller to the target board, 30H in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

  In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is low, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate.

  When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (x8H).

(2) The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte : 86H for UART mode and 30H for I/O Interface mode.

- UART mode

  If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the BR0CR and BR0ADD registers of the SIO0 and sends back 86H to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication.

  Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 86H within the alloted time-out period, the controller should give up the communication.

  The boot program sets the RXE bit in the SC0MOD0 register to enable reception before loading the SIO transmit buffer with 86H.

- I/O Interface mode

  The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 30H to the SC0BUF and drives P76 high. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller must wait for P76 to go high before sending the SCLK0 clock to the target board. This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 30H, then the controller should take it as a go-ahead. The controller must then delivers the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the SC0MOD0.RXE bit to 1 before P76 goes high (before the target board is to receive the third byte).

(3) The 3rd byte, which the target board receives from the controller, is a command. The code for the RAM Transfer command is 10H.

(4) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

  If the 3rd byte is equal to any of the command codes listed in Table 3.5 on page 33, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 10H and then branches to the RAM Transfer routine. Once this branch is taken, a password check is done. Password checking is detailed in Section 3.5.13.

  If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command.

(5) The 5th to 16th bytes, which the target board receives from the controller, are a 12-byte password. The 5th byte is compared to the contents of address 0x0000_03F4 in the flash memory; the 6th byte is compared to the contents of address 0x0000_03F5 in the flash memory; likewise, the 16th byte is compared to the contents of address 0x0000_03FF in the flash memory. If the password checking fails, the RAM Transfer routine sets the password error flag.

(6) The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section 3.5.15.

(7) The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes.

First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 17th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password check. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison, all of the 12 bytes of a password in the flash memory are the same value other than FFH.
- Not all of the password bytes transmitted from the controller matched those contained in the flash memory.

When all the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(8) The 19th to 22nd bytes, which the target board receives from the controller, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31–24 of the address, and the 22nd byte corresponds to bits 7–0 of the address.

(9) The 23rd and 24th bytes, which the target board receives from the controller, indicate the number of bytes that will be transferred from the controller to be stored in the RAM.

(10) The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section 3.5.15.

(11) The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data.

First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- The RAM storage start address must be within the range 0xFFFF_8000–0xFFFF_8FFF.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

(12) The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMP1940FDBF. Storage begins at the address specified by the 19th–22nd bytes and continues for the number of bytes specified by the 23rd–24th bytes.

(13) The (m+1)th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section 3.5.15.

(14) The (m+2)th byte is a acknowledge response to the 27th to (m+1)th bytes.

First, the RAM Transfer routine checks for a receive error in the 27th to (m+1)th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1)th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

(15) If the (m+2)th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes in 32-bit ISA mode.

Note: At this point, r29 (sp) points to address 0xFFFF_9100. Program control must not be transferred from the RAM back to the boot ROM.

3.5.9 Show Flash Memory Sum Command

See Table 3.7.

(1) The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

(2) The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Flash Memory Sum command is 20H.

(3) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.5 on page 33, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was

received, the boot program echoes back a value of 20H and then branches to the Show Flash Memory Sum routine.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command.

(4) The Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total sum, respectively. For details on sum calculation, see Section 3.5.14.

(5) The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes together, drop the carry and take the two's complement of the sum. Transmit this checksum value from the controller to the target board.

(6) The 8th byte is the next command code.

3.5.10 Show Product Information Command

See Table 3.8.

(1) The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

(2) The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 30H.

(3) The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 3.5 on page 33, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 30H and then branches to the Show Flash Memory Sum routine.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command.

(4) The 5th to 8th bytes, transmitted from the target board to the controller, are the data read from addresses 0x0000_03F0–0x0000_03F3 in the flash memory. Software version management is possible by storing a software id in these locations.

(5) The 9th to 20th bytes, transmitted from the target board to the controller, indicate the product name, which is TMP1940FDBF_ in ASCII code (where _ is a space).

(6) The 21st to 24th bytes, transmitted from the target board to the controller, indicate the start address of the flash memory area containing the password, i.e., F4H, 03H, 00H, 00H.

(7) The 25th to 28th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip RAM, i.e., 00H, 80H, FFH, FFH.

(8) The 29th to 32nd bytes, transmitted from the target board to the controller, are dummy data (FFH, 8FH, FFH, FFH).

(9) The 33rd to 36th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip RAM, i.e., FFH, BFH, FFH, FFH.

(10) The 37th to 44th bytes, transmitted from the target board to the controller, are dummy data.

(11) The 45th and 46th bytes, transmitted from the target board to the controller, indicate the presence or absence of the security and protect bits and whether the flash memory is divided into blocks. Bit 0 indicates the presence or absence of the security bit; it is 0 if the security bit is available. Bit 1 indicates the presence or absence of the protect bits; it is 0 if the protect bits are available. If bit 2 is 0, it indicates that the flash memory is divided into blocks. The remaining bits are undefined. The 45th and 46th bytes are both 00H.

(12) The 47th to 50th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip flash memory, i.e., 00H, 00H, 00H, 00H.

(13) The 51st to 54th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip flash memory, i.e., FFH, FFH, 07H, 00H.

(14) The 55th and 56th bytes, transmitted from the target board to the controller, indicate the number of flash blocks available.

(15) The 57th to 92nd bytes, transmitted from the target board to the controller, contain information about the flash blocks.

Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in halfwords) and the number of the blocks in that group.

The 57th to 65th bytes are the information about the 32-Kbyte blocks (Block 0 to Block 14); the 66th to 74th bytes are the information about the 16-Kbyte block (Block 15); the 75th to 83rd bytes are the information about the 8-Kbyte block (Block 16); and the 84th to 92nd bytes are the information about the 4-Kbyte blocks (Blocks 17 and 18). See Table 3.8 on page 36 for the values of the bytes transmitted.

(16) The 93rd byte, transmitted from the target board to the controller, is a checksum value for the 5th to 92nd bytes. The checksum value is calculated by adding all these bytes together, dropping the carry and taking the two's complement of the total sum.

(17) The 94th byte is the next command code.

3.5.11 Acknowledge Responses

The boot program represents processing states with specific codes. Table 3.9 to Table 3.11 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 3.9 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
86H	The SIO can be configured to operate in UART mode. (See Note)
30H	The SIO can be configured to operate in I/O Interface mode.

Note: If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 3.10 ACK Response to the Command Byte

Return Value	Meaning
x8H (See Note)	A receive error occurred while getting a command code.
x1H (See Note)	An undefined command code was received. (Reception was completed normally.)
10H	The RAM Transfer command was received.
20H	The Show Flash Memory Sum command was received.
30H	The Show Product Information command was received.

Note: The four high-order bits of the ACK response are the same as those of the previous command code.

Table 3.11 ACK Response to the Checksum Byte

Return Value	Meaning
18H	A receive error occurred.
11H	A checksum or password error occurred.
10H	The checksum was correct.

3.5.12 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 86H at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 30H at 1/16 the desire baud rate. Figure 3.11 shows the waveforms for the first byte.

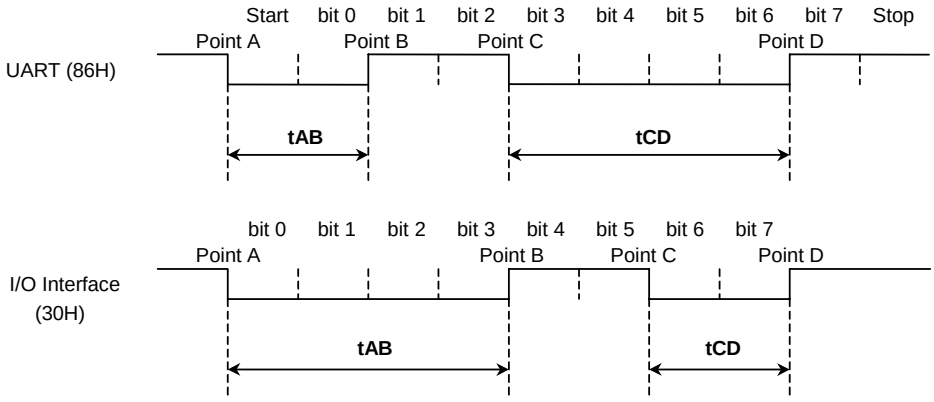


Figure 3.11 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 3.12 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Figure 3.13 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the legnth of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (86H) from the target board. The controller should give up the communication if it fails to get that echo-back within the alloted time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 30H, the controller should give up further communications.

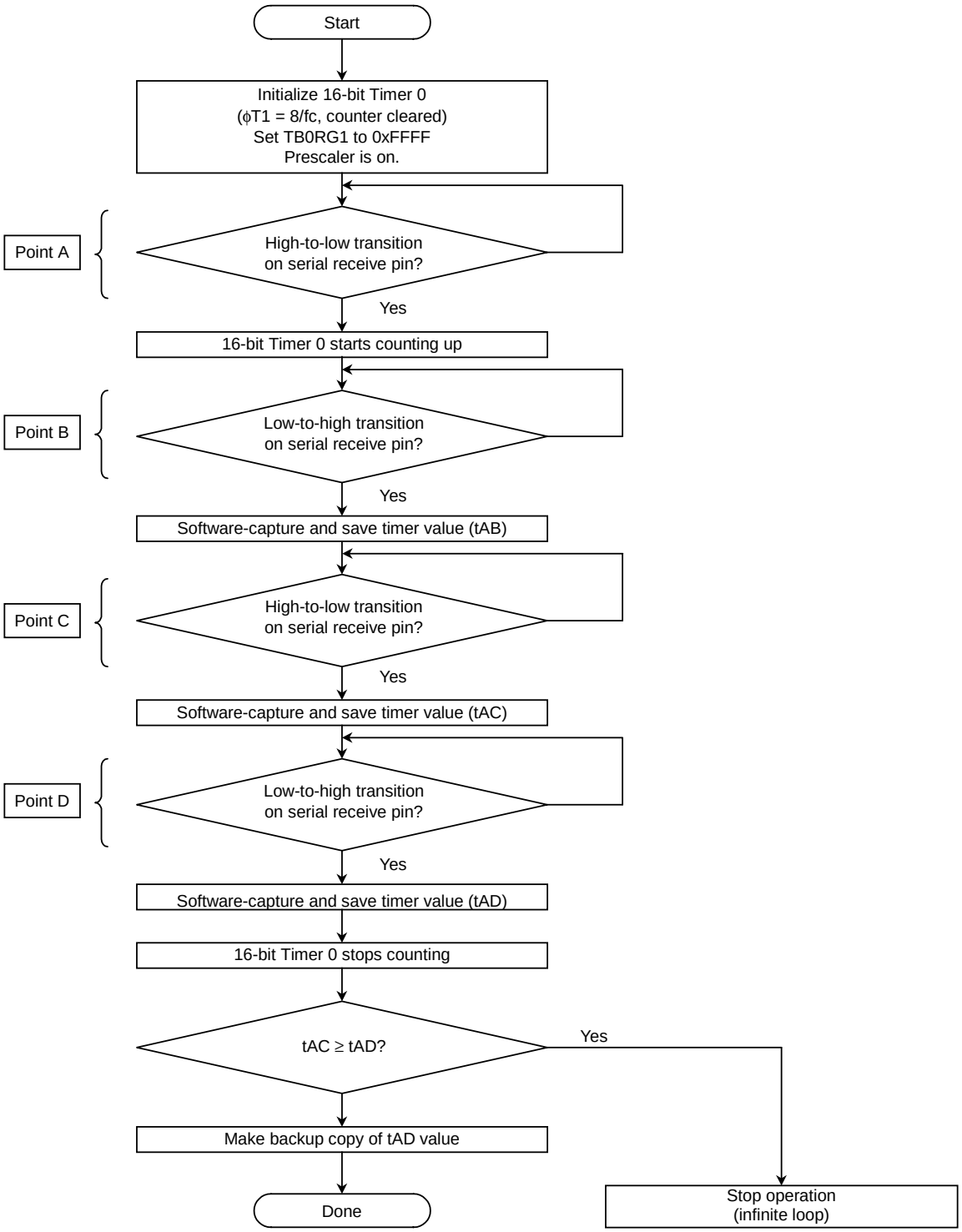


Figure 3.12 Serial Operation Mode Byte Reception Flow

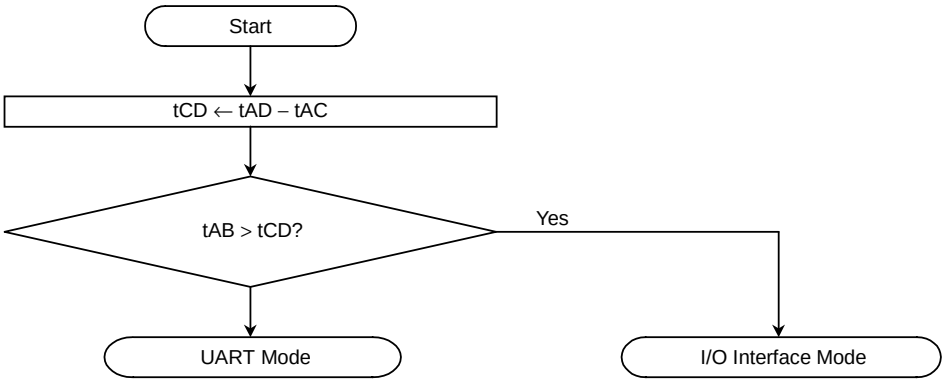


Figure 3.13 Serial Operation Mode Determination Flow

3.5.13 Password

The RAM Transfer command (10H) causes the boot program to perform a password check. Following an echo-back of the command code, the boot program checks the contents of the 12-byte password area (0x0000_03F4 to 0x_0000_03FF) within the flash memory. If all these address locations contain the same bytes of data other than FFH, a password area error occurs. In this case, the boot program returns an error acknowledge (11H) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all FFHs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. Table 3.12 shows how they are compared byte-by-byte. All of the 12 bytes must match to pass the password check. Otherwise, a password error occurs, which causes the boot program to return an error acknowledge in response to the checksum byte (the 17th byte).

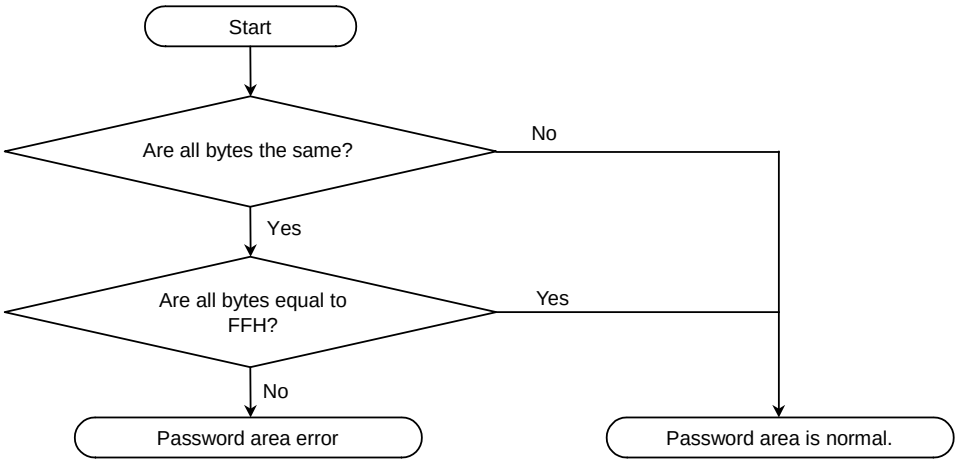


Figure 3.14 Password Area Check Flow

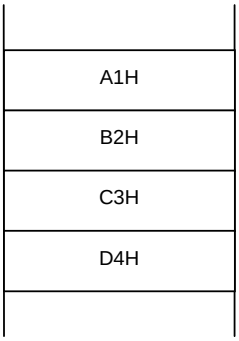
Table 3.12 Relationship between Received Bytes and Flash Memory Locations

Received Byte	Compared Flash Memory Data
5th byte	Address 0x0000_03F4
6th byte	Address 0x0000_03F5
7th byte	Address 0x0000_03F6
8th byte	Address 0x0000_03F7
9th byte	Address 0x0000_03F8
10th byte	Address 0x0000_03F9
11th byte	Address 0x0000_03FA
12th byte	Address 0x0000_03FB
13th byte	Address 0x0000_03FC
14th byte	Address 0x0000_03FD
15th byte	Address 0x0000_03FE
16th byte	Address 0x0000_03FF

3.5.14 Calculation of the Show Flash Memory Sum Command

The Show Flash Memory Sum command adds all 512 Kbytes of the flash memory together and provides the total sum as a halfword quantity. The sum is sent to the controller, with the upper eight bits first, followed by the lower eight bits.

Example:



For the interest of simplicity, assume the depth of the flash memory is four locations. Then the sum of the four bytes is calcualted as:

A1H + B2H + C3H + D4H = 02EAH

Hence, 02H is first sent to the controller, followed by EAH.

3.5.15 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two's complement of the total sum. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation in transmitting checksum bytes.

Example:

Assume the Show Flash Memory Sum command provides the high- and low-order bytes of the sum as E5H and F6H. To calculate the checksum for a series of E5H and F6H:

(1) Add the bytes together.

E5H + F6H = 1DBH

(2) Drop the carry.

(3) Take the two's complement of the sum, and that is the checksum byte.

0 – DBH = 25H

3.5.16 General Boot Program Flowchart

Figure 3.15 shows an overall flowchart of the boot program.

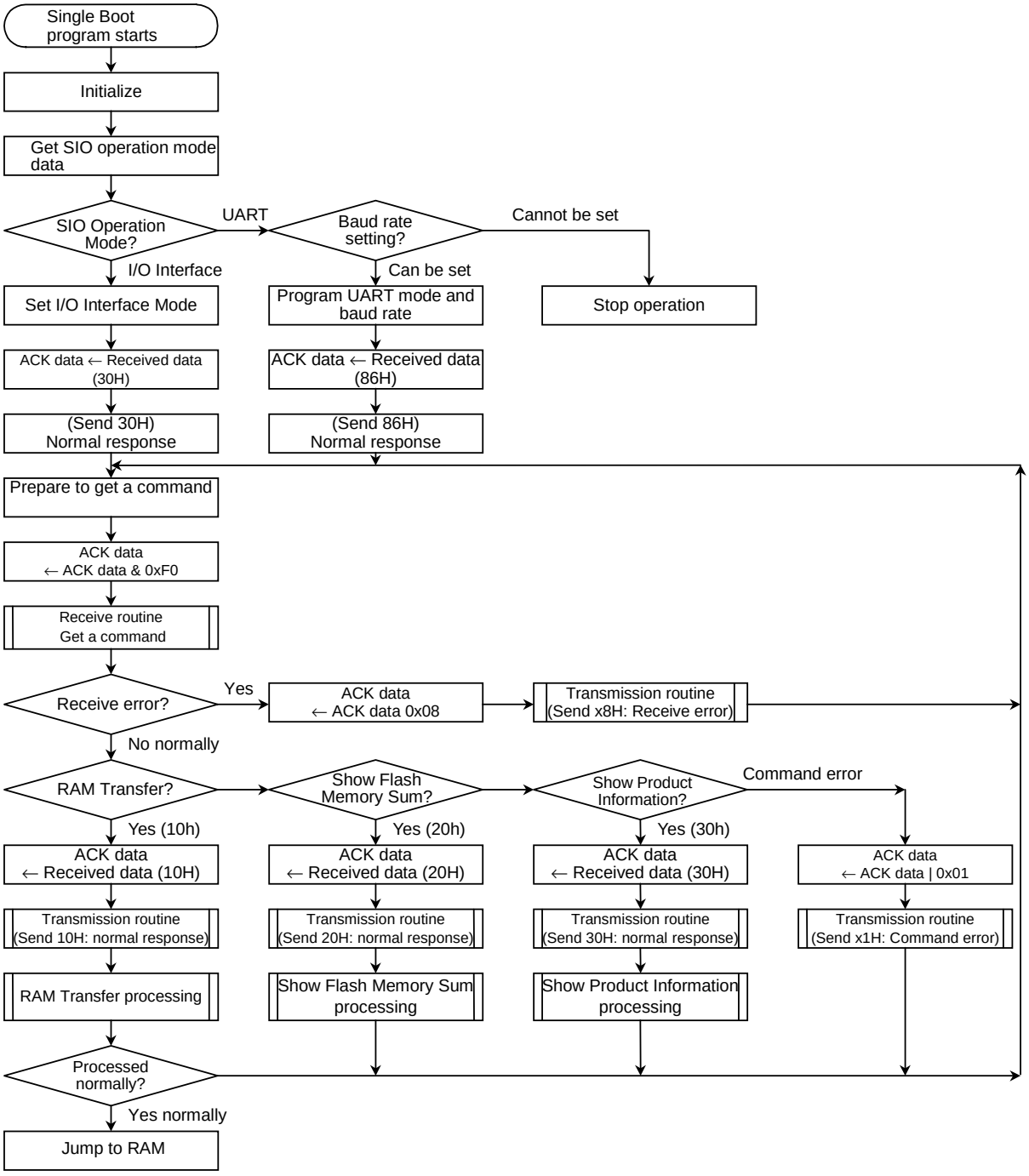


Figure 3.15 Overall Boot Program Flow

3.5.17 Relationships Between Baud Rates and Operating Frequencies

Use the following table as a guide when determining the operating frequency and the baud rate.

Table 3.13 Relationships Between SIO Baud Rates and Frequencies Recommended in Single Boot Mode

UART Mode			Baud Rate (bps)						
External Clock Frequency	PLL	Operating Frequency	76800	57600	38400	19200	9600	4800	2400
8 MHz	On	32 MHz	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7 MHz	On	28 MHz	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6 MHz	On	24 MHz	Yes	Yes	Yes	Yes	Yes	Yes	Yes
20 MHz	Off	20 MHz	Yes	Yes	Yes	Yes	Yes	Yes	Yes
16 MHz	Off	16 MHz	Yes	Yes	Yes	Yes	Yes	Yes	Yes

I/O Interface Mode			Baud Rate (bps)					
External Clock Frequency	PLL	Operating Frequency	1.25 M	850 M	500 K	250 K	125 K	62.5 K
8 MHz	On	32 MHz	Yes	Yes	Yes	Yes	Yes	Yes
7 MHz	On	28 MHz	Yes	Yes	Yes	Yes	Yes	Yes
6 MHz	On	24 MHz	Yes	Yes	Yes	Yes	Yes	Yes
20 MHz	Off	20 MHz	Yes	Yes	Yes	Yes	Yes	Yes
16 MHz	Off	16 MHz	Yes	Yes	Yes	Yes	Yes	Yes

3.6 On-Board Programming and Erasure

The TMP1940FDBF flash memory is command set compatible with the JEDEC EEPROM standard, with a few exceptions. In User Boot mode and Single Boot mode (the RAM Transfer command), the flash memory can be programmed and erased by the CPU executing software commands. It is the user's responsibility to create a program/erase routine. Because the flash memory can not be read while it is being programmed or erased, the program/erase routine must be executed out of the on-chip RAM or an external memory device.

3.6.1 Key Features

The TMP1940FDBF flash memory commands are in principle compatible with the standard JEDEC commands. For program/erase operations, the system can issue a command sequence to the flash memory by using CPU instructions such as LD. After the command sequence is written, the flash memory does not require the system to provide further controls or timings. The flash memory initiates the embedded program or erase algorithm automatically. The entire flash memory or one or more flash blocks can be erased at a time.

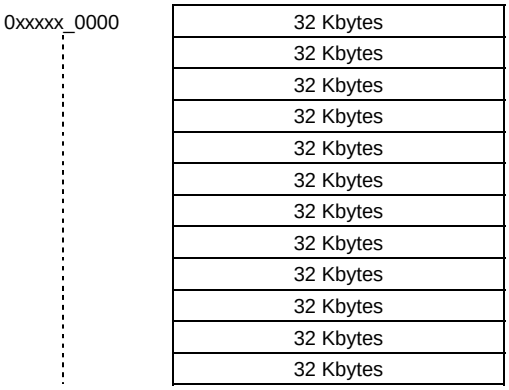
Table 3.14 Flash Memory Features

Feature	Description
Auto Program	Programs and verifies the desired addresses word by word automatically.
Auto Chip Erase	Erases and verifies the entire memory array automatically.
Auto Block Erase	Erases and verifies all memory locations in the selected block automatically.
Auto Multi-Block Erase	Erases and verifies all memory locations in multiple selected blocks automatically.
Write operation status	Provides several status bits such as the Data Polling bit, which can be used to determine whether a program or erase operation is complete or in progress.
Security feature	Prevents intrusive access to the flash memory while in Programmer mode. When the security feature is turned off, the entire memory array is erased and verified automatically, regardless of whether a given block is protected or not.
Block protection	Disables both program and erase operations in any block.

Bear in mind that, due to the on-chip CPU interface, the TMP1940FDBF uses addresses different from those of the standard flash command sequences. Unless otherwise noted, programming is done word by word; thus the word load instruction should be used to write to the flash array. The byte load instruction can be used to issue commands to the flash memory.

The program/erase operations in Programmer mode are very similar to those of the on-board programming modes, with a few exceptions such as the data bus width. Refer to Section 3.7 for a description of the program and erase operations in Programmer mode.

3.6.2 Block Architecture



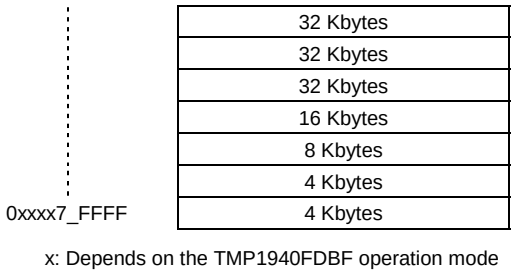


Figure 3.16 Flash Memory Block Architecture

3.6.3 CPU-to-Flash Interface

Figure 3.17 illustrates the internal interface between the CPU and the flash memory in on-board programming modes. The diagram does not show the actual logic network; instead it is only a conceptual depiction of the CPU-to-flash interface.

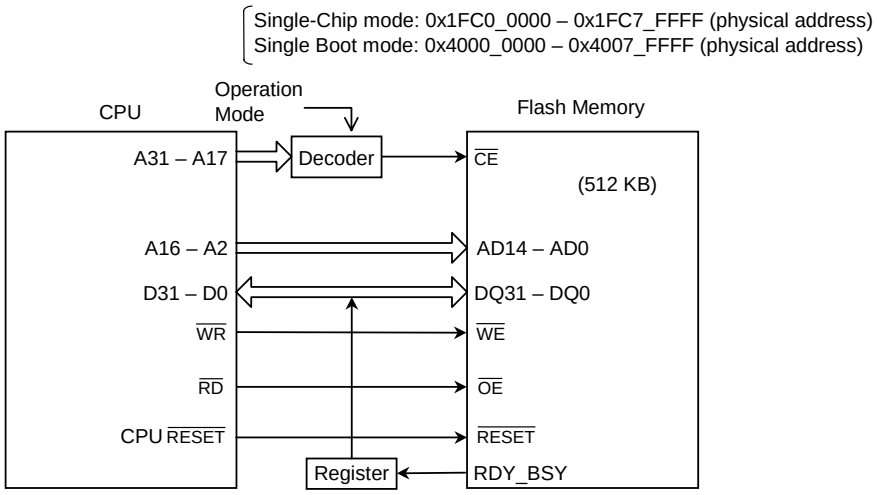


Figure 3.17 Internal CPU-to-Flash Interface

3.6.4 Read Mode and Embedded Operation Mode

The flash memory of the TMP1940FDBF has the following two modes of operation:

- Read mode in which array data is read
- Embedded Operation mode in which the flash array is programmed or erased

The flash memory enters Embedded Operation mode when a valid command sequence is executed in Read mode. In Embedded Operation mode, array data can not be read.

3.6.5 Reading Array Data

The flash memory is automatically set to reading array data upon CPU reset after device power-up and after an embedded operation is successfully completed.

3.6.6 Writing Commands

The operations of the flash memory are selected by commands or command sequences written into the internal command register. This uses the same mechanism as for JEDEC-standard EEPROMs. Commands are made up of data sequences written at specific addresses via the command register. See Table 3.16 on page 60 for the list of command sequences.

Commands are written via DQ0–DQ7 except the fourth (read) cycle in the Read/Reset command sequence, the fourth (write) cycle in the Auto Program command sequence and the fourth (write) cycle in the Verify Block Protect command sequence. Thus commands can be provided byte by byte.

The command sequence being written can be canceled by issuing the Read/Reset command between sequence cycles. The Read/Reset command clears the command register and resets the flash memory to Read mode. Invalid command sequences also cause the flash memory to clear the command register and return to Read mode.

3.6.7 Reset

- Read/Reset command (software reset)

  The flash memory does not return to Read mode if an embedded operation terminated abnormally. In this case, the Read/Reset command must be issued to put the flash memory back in Read mode. The Read/Reset command may also be written between sequence cycles of the command being written to clear the command register.

- Hardware reset ($\overline{\text{RESET}}$ input)

  As shown in Figure 3.17, the flash memory has a reset pin, which is connected to the reset signal of the CPU. When the system drives the $\overline{\text{RESET}}$ pin to V_{IL} or when certain events such as a watchdog timer time-out causes a CPU reset, the flash memory immediately terminates any operation in progress and is reset to Read mode.

  The Read/Reset command is also tied to the $\overline{\text{RESET}}$ pin to reset the flash memory to Read mode. The embedded operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

  For a description of the hardware reset operation, see Section 3.3.2, *Reset Operation*. When a valid reset is achieved, the CPU reads the Reset exception vector from the flash memory and services the Reset exception.

3.6.8 Auto Program Command

A bit must be programmed to change its state from a 1 to a 0. A bit can not be programmed from a 0 back to a 1. Only an erase operation can change a 0 back to a 1.

In User Boot mode and the RAM Transfer command of Single Boot mode, the Auto Program command programs the desired addresses word by word. The Auto Program command requires four bus cycles; the program address and data are written in the fourth cycle, upon completion of which the program operation will commence. As programming is performed on a word-by-word basis, the program address must be a multiple of four.

Writing data shorter than a 32-bit word requires special considerations for the bits that are not to be altered. The word in the memory does not need to be in the erased state prior to programming. If the word is in the erased state, a 32-bit write must be performed, with all the bits not to be altered set to 1. If the word is not in the erased state, it must be loaded into the CPU first to modify necessary bits, and the modified word must be written to the flash memory.

Examples:

- When a word location is in the erased state

  To program the least-significant byte of that word to 55H, 0xFFFF_FF55 must be written to the word address.

- When a word location is not in the erased state and contains 0x8888_88FF

  To program the least-significant byte of that word to AAH, 0x8888_88AA must be written to the word address.

The Auto Program command executes a sequence of internally timed events to program the desired bits of the addressed memory word and verify that the desired bits are sufficiently programmed. The system can determine the status of the programming operation by using write status flags (see Table 3.19 on page 62).

Any commands written during the programming operation are ignored. A hardware reset immediately terminates the programming operation. The programming operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables programming operations in any block. If an attempt is made to program a protected block, the Auto Program command does nothing; the flash memory returns to Read mode in approximately 3 μm after the completion of the fourth bus cycle of the command sequence.

When the embedded Auto Program algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the programming operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of a programming failure, it is recommended to replace the chip or discontinue the use of the failing flash block.

3.6.9 Auto Chip Erase Command

The Auto Chip Erase command requires six bus cycles. After completion of the sixth bus cycle, the Auto Chip Erase operation will commence immediately. The embedded Auto Chip Erase algorithm automatically preprograms the entire memory for an all-0 data pattern prior to the erase; then it

automatically erases and verifies the entire memory for an all-1 data pattern. The system can determine the status of the chip erase operation by using write status flags (see Table 3.19 on page 62).

Any commands written during the chip erase operation are ignored. A hardware reset immediately terminates the chip erase operation. The chip erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Chip Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the blocks are protected, the Auto Chip Erase command does nothing; the flash memory returns to Read mode in approximately 100 μm after the completion of the sixth bus cycle of the command sequence.

When the embedded Auto Chip Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. The failing block can be identified by means of the Block Erase command.

3.6.10 Auto Block Erase and Auto Multi-Block Erase Commands

The Auto Block Erase command requires six bus cycles. A time-out begins from the completion of the command sequence. After a time-out, the erase operation will commence. The embedded Auto Block Erase algorithm automatically preprograms the selected block for an all-0 data pattern, and then erases and verifies that block for an all-1 data pattern.

During the time-out period, additional block addresses and Auto Block Erase commands may be written. For more on this, see Figure 3.20.

Any command other than Auto Block Erase during the time-out period resets the flash memory to Read mode. The block erase time-out period is 50 μm. The system may read DQ3 to determine whether the time-out period has expired. The block erase timer begins counting upon completion of the sixth bus cycle of the Auto Block Erase command sequence. The system can determine the status of the erase operation by using write status flags (see Table 3.19 on page 62).

Any commands written during the block erase operation are ignored. A hardware reset immediately terminates the block erase operation. The block erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Block Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the selected blocks are protected, the Auto Block Erase algorithm does nothing; the flash memory returns to Read mode in approximately 100 μm after the final bus cycle of the command sequence. When the embedded Auto Block Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. If any failure occurred during the multi-block erase operation, the failing block can be identified by running Auto Block Erase on each of the blocks selected for multi-block erasure.

3.6.11 Block Protect Command

The block protection feature disables both program and erase operations in any block. The effects of the program and erase commands on the protected blocks are summarized below.

Table 3.15 Effects of the Program and Erase Commands on the Protected Blocks

Command	Operation
Program command on a protected block	No programming operation is performed, and the flash memory automatically returns to Read mode.
Block Erase command on a protected block	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when all the blocks are protected	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.
Multi-Block Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.

The Block Protect command requires 10 bus cycles. The address of the block to be protected is internally latched in the seventh cycle. Then, allow an interval of 4 µm to elapse before providing data for the eighth cycle, which enables writing to the protection control circuitry. Next, allow an interval of at least 100 µm to elapse before providing data for the ninth cycle. This terminates writing to the protection control circuitry. Finally, allow an interval of 8 µm to elapse and provide data for the tenth cycle to complete the command.

Note that the block protect operation is not verified automatically. The Verify Block Protect command must be written to verify the protect status after executing Block Protect. If the desired block is not in the protected state, the Block Protect command sequence must be re-initiated. Figure 3.22 illustrates the algorithm for the Block Protect command.

Any commands written during the Block Protect algorithm are ignored. A hardware reset immediately terminates the block protect operation. The Block Protect command that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence.

3.6.12 Verify Block Protect Command

The Verify Block Protect command is used to verify the protect status of a block. Verify Block Protect is a four-bus-cycle operation. The address of the block to be verified is given in the fourth cycle. Any address within the block range will suffice, provided A0 = A1 = A2 = A3 = 0, A4 = 1 and A6 = 0. To get correct data, a 32-bit read must be performed at least twice. Use the last read as valid data. If the selected block is protected, a value of 0x0000_0001 is returned. If the selected block is not protected, a value of 0x0000_0000 is returned. Following the fourth bus cycle, an additional block address may be read.

The Verify Block Protect command does not return the flash memory to Read mode. Either the Read/Reset command or a hardware reset is required to reset the flash memory to Read mode or to write the next command.

3.6.13 Write Operation Status

As shown in Table 3.19, the flash memory provides several flag bits to determine the status of an embedded operation: DQ7, DQ5 and DQ3. These status bits can be read during an embedded operation using the same timing as for Read mode. The flash memory automatically returns to Read mode when an embedded operation completes.

During the embedded program operation, the system must provide the program address (with A0 = 0 and A1 = 0) to read valid status information. During the embedded erase operation, the system must provide an address (with A0 = 0 and A1 = 0) within any of the blocks selected for erasure to read valid status information.

- DQ7 (Data Polling)

  The Data Polling bit, DQ7, indicates to the host system the status of the embedded operation. Data Polling is valid after the final bus write cycle of an embedded command sequence.

  When the embedded Program algorithm is in progress, an attempt to read the flash memory will produce the complement of the data last written to DQ7. Upon completion of the embedded Program algorithm, an attempt to read the flash memory will produce the true data last written to DQ7. Therefore, the system can use DQ7 to determine whether the embedded Program algorithm is in progress or complete.

  When the embedded Erase algorithm is in progress, an attempt to read the flash memory will produce a 0 at the DQ7 output. Upon completion of the embedded Erase algorithm, the flash memory will produce a 1 at the DQ7 output.

  If there is a failure during an embedded operation, DQ7 continues to output the same value. Thus, DQ7 must always be polled in conjunction with the Exceeded Timing Limits (DQ5) flag. Figure 3.21 shows the DQ7 polling algorithm.

  The flash memory disables address latching when an embedded operation is complete. Data polling must be performed with a valid programmed address or an address within any of the non-protected blocks selected for erasure.

- DQ5 (Exceeded Timing Limits)

  DQ5 produces a 0 while the program or erase operation is in progress normally. DQ5 produces a 1 to indicate that the program or erase time has exceeded the specified internal limit. This is a failure condition that indicates the program or erase cycle was not successfully completed.

  The DQ5 failure condition also appears if the system tries to program a 1 to a location that was previously programmed to a 0. Only an erase operation can change a 0 back to a 1. In this case, the embedded Program algorithm halts the operation. Once the operation has exceeded the timing limits, DQ5 will indicate a 1. Note that this is not a device failure condition since the flash memory was used incorrectly.

  Under both these conditions, the flash memory remains locked in Embedded Operation mode. The system must issue the Read/Reset command to return the flash memory to Read mode.

- DQ3 (Block Erase Timer)

  After the completion of the sixth bus cycle of the Auto Block Erase command sequence, the block erase time-out window of 50 μm begins. The erase operation will begin after the time-out has expired. When the time-out is complete and the erase operation has begun, DQ3 switches from 0 to 1. If DQ3 is 0, the flash memory will accept additional Auto Block Erase commands. Each time an Auto Block Erase command is written, the time-out window is reset. To ensure that the command has been accepted, the system should check DQ3 prior to and following each Auto Block Erase command. If DQ3 is 1 on the second status check, the command might not have been accepted.

3.6.14 Flash Control/Status Register

This is an 8-bit register that indicates the Ready/Busy status of an embedded algorithm and controls the security feature.

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	RDY_BSY	—	FSE
Read/Write	—	—	—	—	R/W	R	R/W	R/W
Reset Value	—	—	—	—	0	1	0	0
Function					Must be written as 0.	Ready/Busy 0: Embedded algorithm is in progress. 1: Embedded algorithm is complete.	Must be written as 0.	Flash security enable 0: Access flash memory array 1: Access security control logic

Figure 3.18 Flash Control/Status Register

- Bit 2: Ready/Busy Flag (RDY_BSY)

  In Programmer mode, the ALE pin functions as the RDY/$\overline{\text{BSY}}$ pin. The host system can monitor the state of this pin to determine whether an embedded algorithm is in progress or complete. The CPU can poll the RDY_BSY bit in the FLCS register for the same purpose. The RDY_BSY bit is cleared to 0 when the flash memory is actively erasing or programming. The RDY_BSY bit is set to 1 when an embedded operation has completed and the flash memory is ready to accept the next command. If any failure occurs during the program or erase operation, this bit remains cleared. A hardware reset sets this bit.

  The RDY_BSY bit is cleared upon completion of the final bus write cycle of an embedded operation command, with one exception. In the case of the Auto Block Erase command, this bit is cleared after the time-out has expired. Any command is ignored while the RDY_BSY bit is cleared.

- Bit 0: Flash Security Enable (FSE)

  The FSE bit is used to enable and disable the security feature. After a reset, this bit is cleared. Under this condition, the program and erase commands access the memory array. To turn on the security feature, set the FSE bit and write the Auto Security On command. Thereafter, the FSE bit must be cleared to enable access to the memory array. To turn off the security feature, set the FSE bit and write the Auto Security Off command.

Note: The Flash Control/Status register must be accessed as a 32-bit quantity.

3.6.15 Flash Security

The TMP1940FDBF flash memory supports not only on-board programming but also programming using a general-purpose programmer. Therefore, the TMP1940FDBF flash memory provides a security feature to prevent intrusive access to the flash memory while in Programmer mode.

The TMP1940FDBF has a security bit apart from the flash array. Programming this security bit disables access to the flash array. The paragraphs that follow describe the methods to secure and unsecure the flash memory. As is the case with a flash programming routine, the security control routine must also be placed and executed outside of the flash memory — either the on-chip RAM or an external memory device.

- Securing the flash (Disabling read accesses)

  Securing the flash memory disables a general-purpose programmer to read its contents. To turn on the security feature, once programming is complete, set the FSE bit in the FLCS register and write the Auto Security On command. After the completion of the fourth bus cycle of that command sequence, the embedded Security On algorithm automatically programs and verifies the security bit.

  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. The FSE bit must not be altered throughout the embedded operation.

  When the embedded algorithm completes, the flash memory automatically returns to Read mode. In on-board operating modes, the CPU can read the flash memory even if the security is on; clear the FSE bit to 0 to enable access to the flash array.

  If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. Note that this is a security bit failure. If the flash memory needs to be secured, the chip should be replaced. When the security is on, any reads by programming equipment will always return a halfword-length value of 0x0098.

- Unsecuring the flash (Enabling read accesses)

  The security feature is designed to disable reads of the flash memory by programming equipment. While the TMP1940FDBF is soldered on a board, the CPU can always read the flash memory, regardless of whether or not the security is on. Since the flash memory is placed under control of a user's application program in on-board operating modes, it is not easy for third parties to perform intrusive access to the flash memory. Therefore, within the confines of a board, the flash memory does not need to be secured.

  To turn off the security feature, set the FSE bit in the FLCS register and write the Auto Security Off command. After the completion of the sixth bus cycle of that command sequence, the embedded Security Off algorithm automatically erases and verifies the entire flash array, and then erases and verifies the security bit.

  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. In this case, if any erase operation is in progress, data may be corrupted. The FSE bit must not be altered throughout the embedded operation.

  When an embedded algorithm completes, the flash memory automatically returns to Read mode. If any on-board operation is subsequently required, clear the FSE bit to 0 to enable access to the flash array.

  If any failure occurs during an embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. If a failure occurs in the memory array, the security bit is not erased. In this case, the security is left on. The chip should be replaced if a memory array or security bit failure occurs.

  The Auto Security Off command erases the flash array prior to turning off the security feature. Even if a given block is protected, it is unconditionally erased, but the protect status of that block remains unchanged. The Auto Security Off and Auto Chip Erase command sequences are the same. The only difference is that the Auto Security Off command requires the FSE bit to be

set to 1 before the command is written. The Auto Block Erase command can not turn off the security feature even when the FSE bit is set. If the Auto Block Erase command is written when the security is on, no block will be erased and the operation is immediately terminated.

3.6.16 Command Definitions

Table 3.16 On-Board Programming Mode Command Definitions

Command Sequence	Cycles Required	Bus Cycles									
		1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/Write)		5th Cycle (Read/Write)	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read/Reset	1	0xXXX0	0xF0								
Read/Reset	3	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0xF0	RA	RD		
Auto Program	4	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0xA0	PA	PD		
Auto Chip Erase	6	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0x80	0xAAA8	0xAA	0x5554	0x55
Auto Block Erase	6	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0x80	0xAAA8	0xAA	0x5554	0x55
Block Protect	10	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0x9A	0xAAA8	0xAA	0x5554	0x55
Verify Block Protect	4	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0x90	BPA	BD	BPA	BD
Auto Security On (Note 1)	4	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0xA0	0x0000	0x98		
Auto Security Off (Note 1)	6	0xAAA8	0xAA	0x5554	0x55	0xAAA8	0x80	0xAAA8	0xAA	0x5554	0x55

(Continued from above)

Command Sequence	Cycles Required	Bus Cycles									
		6th Cycle (Write)		7th Cycle (Write)		8th Cycle (Write)		9th Cycle (Write)		10th Cycle (Write)	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read/Reset	1										
Read/Reset	3										
Auto Program	4										
Auto Chip Erase	6	0xAAA8	0x10								
Auto Block Erase	6	BA	0x30								
Block Protect	10	0xAAA8	0x9A	BA	0x00	0xXXX0	0x00	0xXXX0	0x00	0xXXX0	0x00
Verify Block Protect	4										
Auto Security On (Note 1)	4										
Auto Security Off (Note 1)	6	0xAAA8	0x10								

Note 1: Before executing the command sequence, set the FSE bit in the Flash Control/Status (FLCS) register to enable access to the security bit.
Note 2: There must be an interval of at least two instructions between each bus cycle.

The addresses to be provided by the CPU are shown below.

Table 3.17 Addresses Provided by the CPU

Command Address	CPU Addresses: A23–A0																
	A23–A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0xXXX0	Flash memory block	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0
0x0000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xAAA8		1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
0x5554		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0

- F0H, AAH, 55H, A0H, 80H, 10H, 30H:

  Command data. Write command data as a byte quantity.

- RA: Read Address

  RD: Read Data

- PA: Program Address

  PD: Program Data

  The address must be a multiple of four. Write data on a word-by-word basis.

- BA: Block Address (BA0–BA18)

  Refer to Table 3.18.

- BPA: Verify Block Protect Address

  BD: Block Protect Data

  Refer to Table 3.18. The address of the block to be verified can be any of the addresses within the block, with A6 = 0, A4 = 1, A3 = 0, A1 = 0 and A0 = 0. If a block is protected, a value of 0x0000_0001 will be returned. If a block is not protected, a value of 0x0000_0000 will be returned.

Table 3.18 Block Erase Addresses

Block	Address Range		Size
	User Boot Mode	Single Boot Mode	
BA0	0x1FC0_0000 thru 0x1FC0_7FFF (or 0x4000_0000 thru 0x4000_7FFF)	0x1FC0_0000 thru 0x1FC0_7FFF	32 Kbytes
BA1	0x1FC0_8000 thru 0x1FC0_FFFF (or 0x4000_8000 thru 0x4000_FFFF)	0x1FC0_8000 thru 0x1FC0_FFFF	32 Kbytes
BA2	0x1FC1_0000 thru 0x1FC1_7FFF (or 0x4001_0000 thru 0x4001_7FFF)	0x1FC1_0000 thru 0x1FC1_7FFF	32 Kbytes
BA3	0x1FC1_8000 thru 0x1FC1_FFFF (or 0x4001_8000 thru 0x4001_FFFF)	0x1FC1_8000 thru 0x1FC1_FFFF	32 Kbytes
BA4	0x1FC2_0000 thru 0x1FC2_7FFF (or 0x4002_0000 thru 0x4002_7FFF)	0x1FC2_0000 thru 0x1FC2_7FFF	32 Kbytes
BA5	0x1FC2_8000 thru 0x1FC2_FFFF (or 0x4002_8000 thru 0x4002_FFFF)	0x1FC2_8000 thru 0x1FC2_FFFF	32 Kbytes
BA6	0x1FC3_0000 thru 0x1FC3_7FFF (or 0x4003_0000 thru 0x4003_7FFF)	0x1FC3_0000 thru 0x1FC3_7FFF	32 Kbytes
BA7	0x1FC3_8000 thru 0x1FC3_FFFF (or 0x4003_8000 thru 0x4003_FFFF)	0x1FC3_8000 thru 0x1FC3_FFFF	32 Kbytes
BA8	0x1FC4_0000 thru 0x1FC4_7FFF (or 0x4004_0000 thru 0x4004_7FFF)	0x1FC4_0000 thru 0x1FC4_7FFF	32 Kbytes
BA9	0x1FC4_8000 thru 0x1FC4_FFFF (or 0x4004_8000 thru 0x4004_FFFF)	0x1FC4_8000 thru 0x1FC4_FFFF	32 Kbytes
BA10	0x1FC5_0000 thru 0x1FC5_7FFF (or 0x4005_0000 thru 0x4005_7FFF)	0x1FC5_0000 thru 0x1FC5_7FFF	32 Kbytes
BA11	0x1FC5_8000 thru 0x1FC5_FFFF (or 0x4005_8000 thru 0x4005_FFFF)	0x1FC5_8000 thru 0x1FC5_FFFF	32 Kbytes
BA12	0x1FC6_0000 thru 0x1FC6_7FFF (or 0x4006_0000 thru 0x4006_7FFF)	0x1FC6_0000 thru 0x1FC6_7FFF	32 Kbytes
BA13	0x1FC6_8000 thru 0x1FC6_FFFF (or 0x4006_8000 thru 0x4006_FFFF)	0x1FC6_8000 thru 0x1FC6_FFFF	32 Kbytes
BA14	0x1FC7_0000 thru 0x1FC7_7FFF (or 0x4007_0000 thru 0x4007_7FFF)	0x1FC7_0000 thru 0x1FC7_7FFF	32 Kbytes
BA15	0x1FC7_8000 thru 0x1FC7_BFFF (or 0x4007_8000 thru 0x4007_BFFF)	0x1FC7_8000 thru 0x1FC7_BFFF	16 Kbytes
BA16	0x1FC7_C000 thru 0x1FC7_DFFF (or 0x4007_C000 thru 0x4007_DFFF)	0x1FC7_C000 thru 0x1FC7_DFFF	8 Kbytes
BA17	0x1FC7_E000 thru 0x1FC7_EFFF (or 0x4007_E000 thru 0x4007_EFFF)	0x1FC7_E000 thru 0x1FC7_EFFF	4 Kbytes
BA18	0x1FC7_F000 thru 0x1FC7_FFFF (or 0x4007_F000 thru 0x4007_FFFF)	0x1FC7_F000 thru 0x1FC7_FFFF	4 Kbytes

The address of the block to be erased can be any of the addresses within that block with A0 = 0 and A1 = 0. For example, to select BA0 in User Boot mode, provide any address in the range between 0x1FC0_0000 and 0x1FC0_7FFF.

Table 3.19 Write Status Flags

Status		D7 (DQ7)	D5 (DQ5)	D3 (DQ3)
Embedded operation in progress	Auto Program	$\overline{DQ7}$	0	0
	Auto Erase (during the time-out window)	0	0	0
	Auto Erase	0	0	1
Time-out in embedded operation	Auto Program	$\overline{DQ7}$	1	1
	Auto Erase	0	1	1

Note: D31–D8, D4 and D2–D0 are don't-cares.

3.6.17 Embedded Algorithms

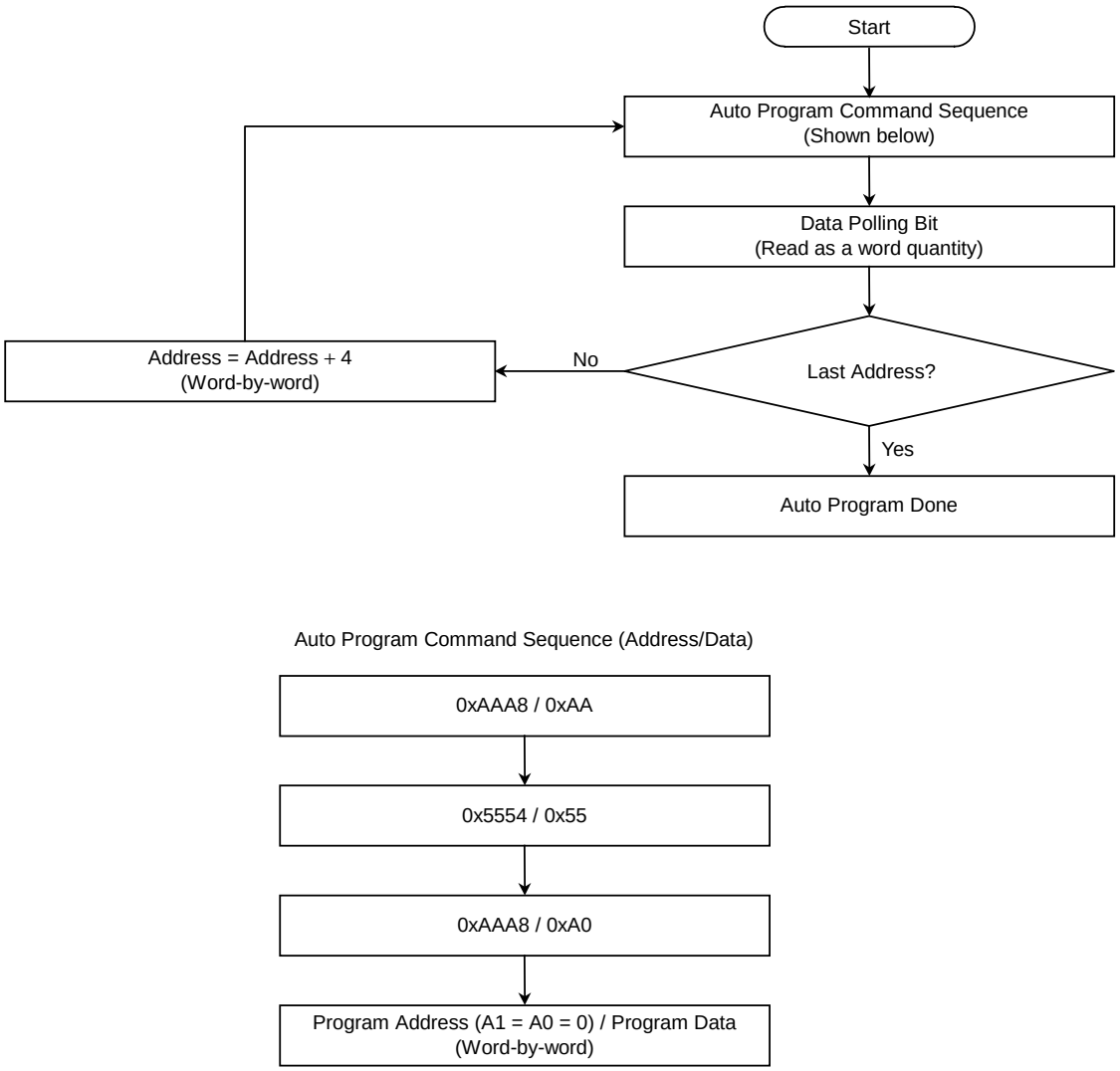


Figure 3.19 Auto Program Operation

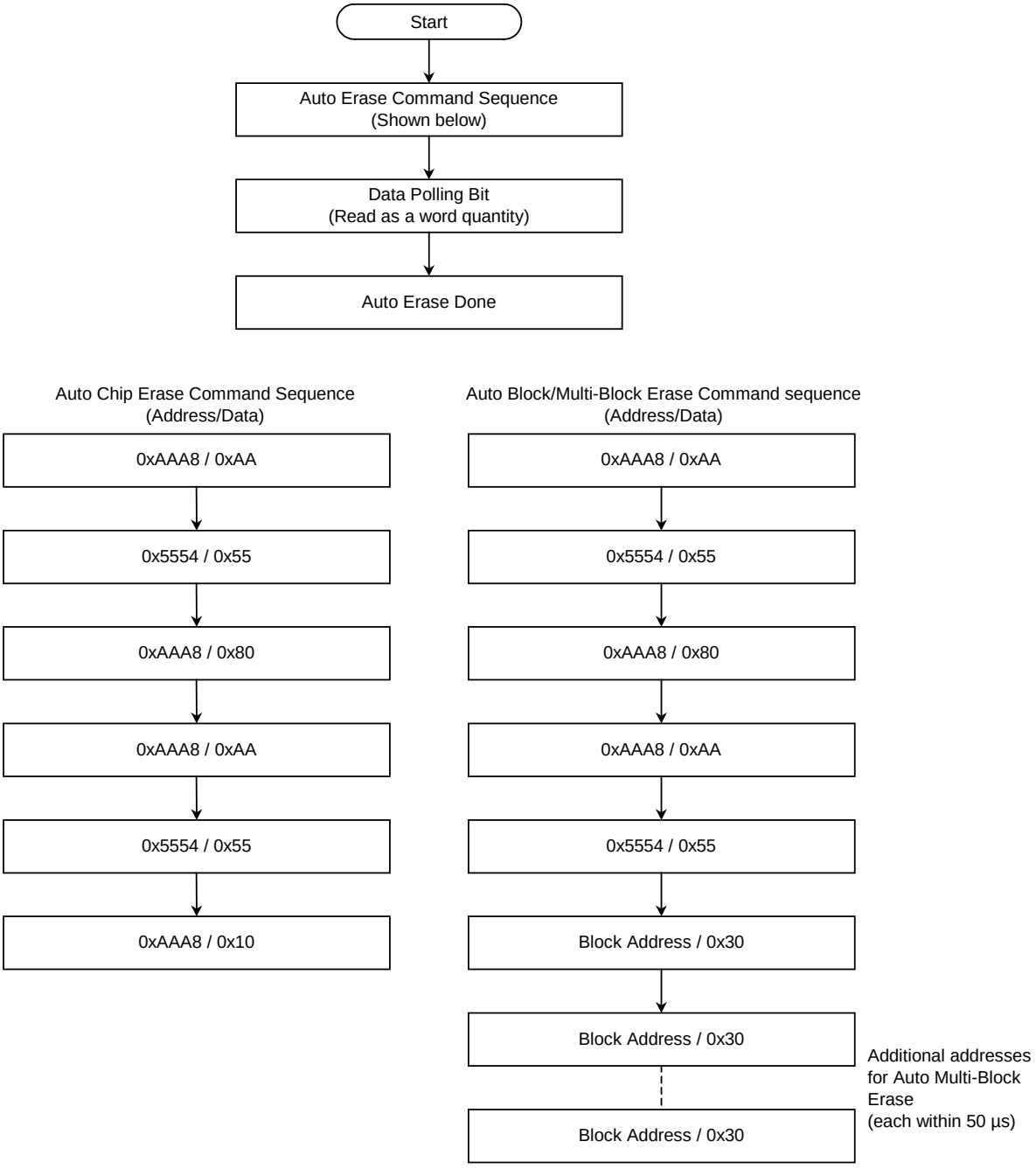


Figure 3.20 Auto Erase Operations

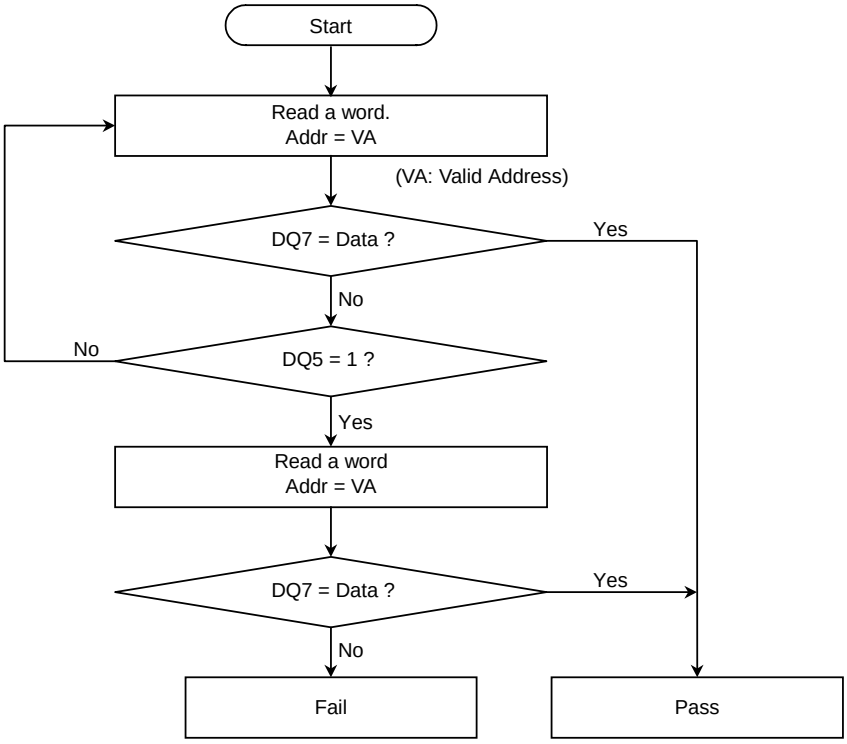


Figure 3.21 Data Polling (DQ7) Algorithm

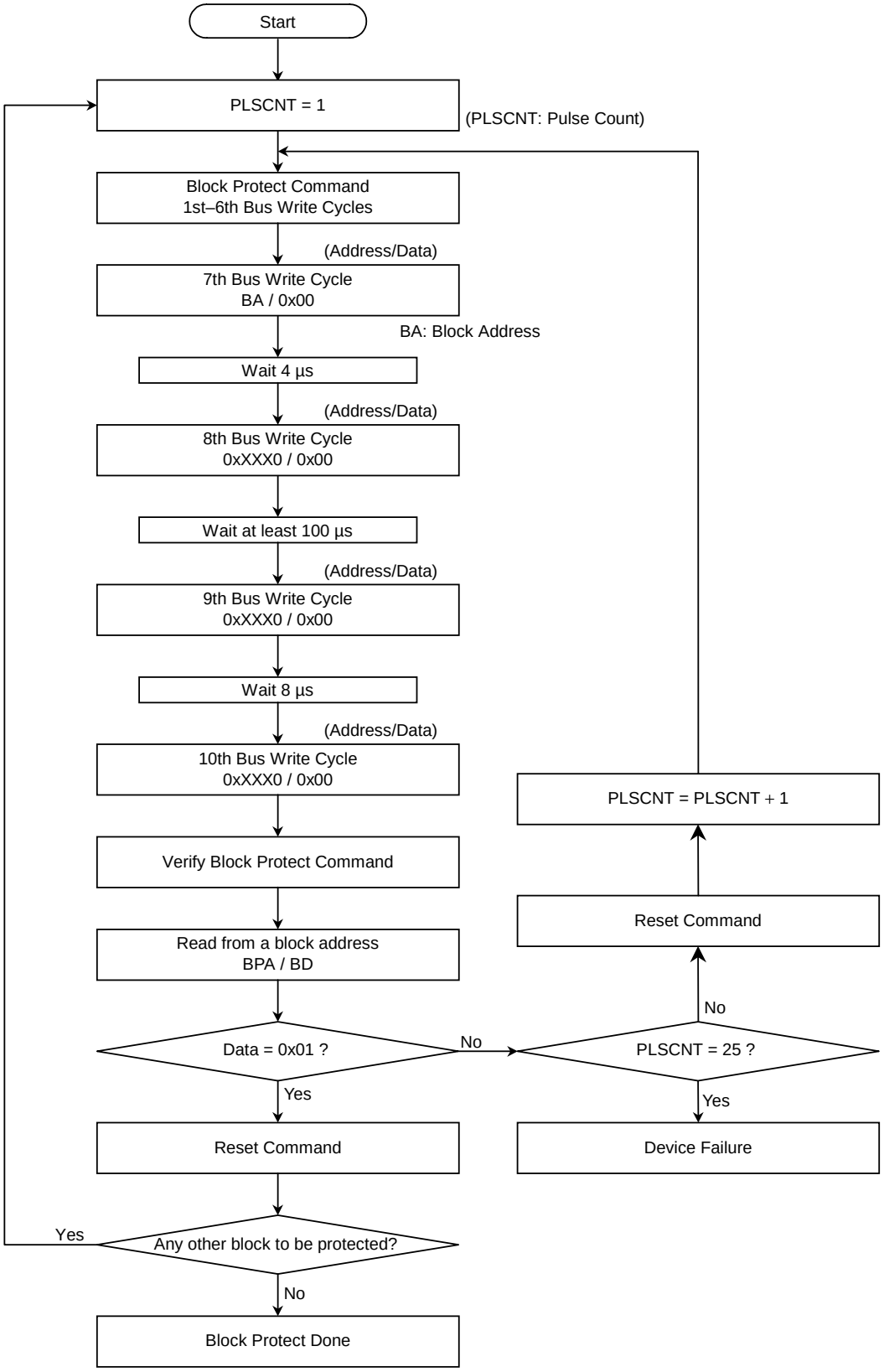


Figure 3.22 Block Protect Operation

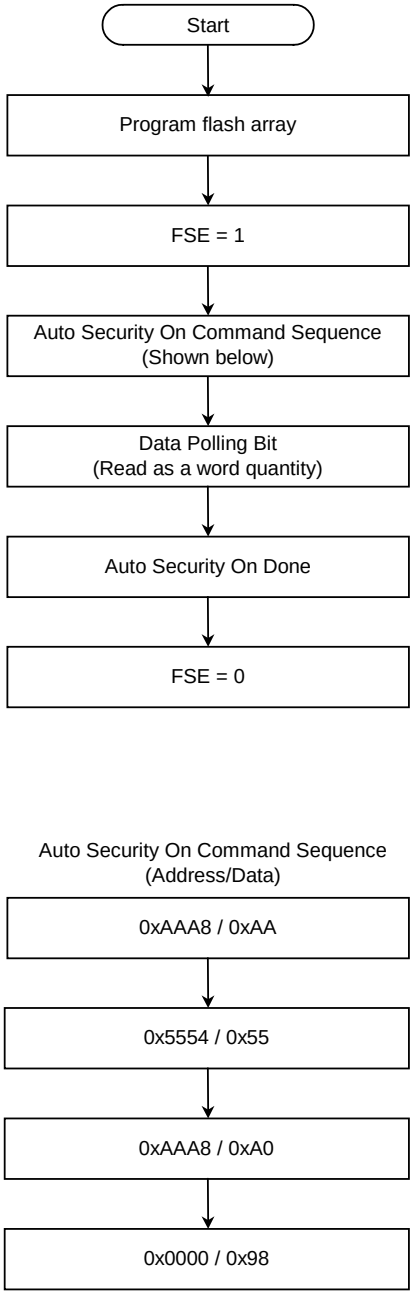


Figure 3.23 Auto Security On Operation

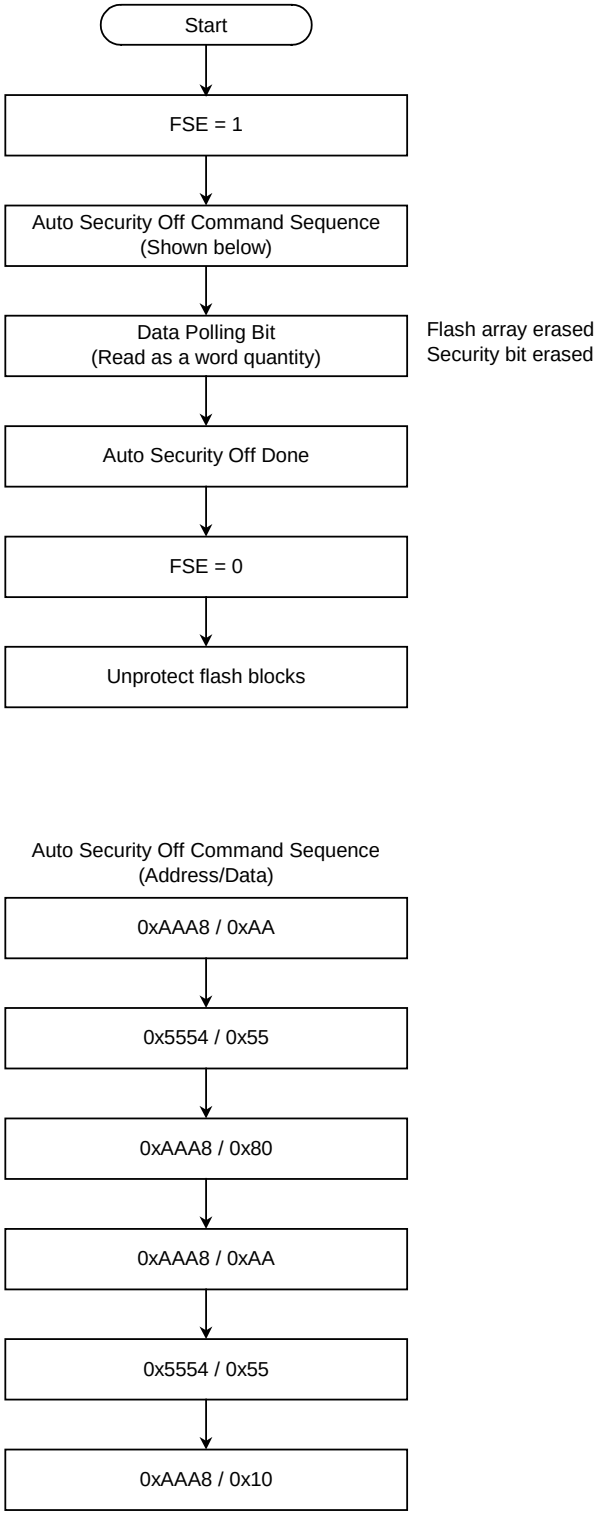


Figure 3.24 Auto Security Off Operation

3.7 Programmer Mode

3.7.1 Mode Setting

The TMP1940FDBF is placed in Programmer mode by holding the $\overline{\text{RESET}}$, BW0, P41 and P42 pins at logic 0 and the BW1 and P40 pins at logic 1. In Programmer mode, the flash memory can be read, erased and programmed using a general-purpose EPROM programmer. For instructions about the settings of the remaining pins, see Section 3.7.3, *Pin Functions and Settings*. Figure 3.25 below shows the pin settings for Programmer mode.

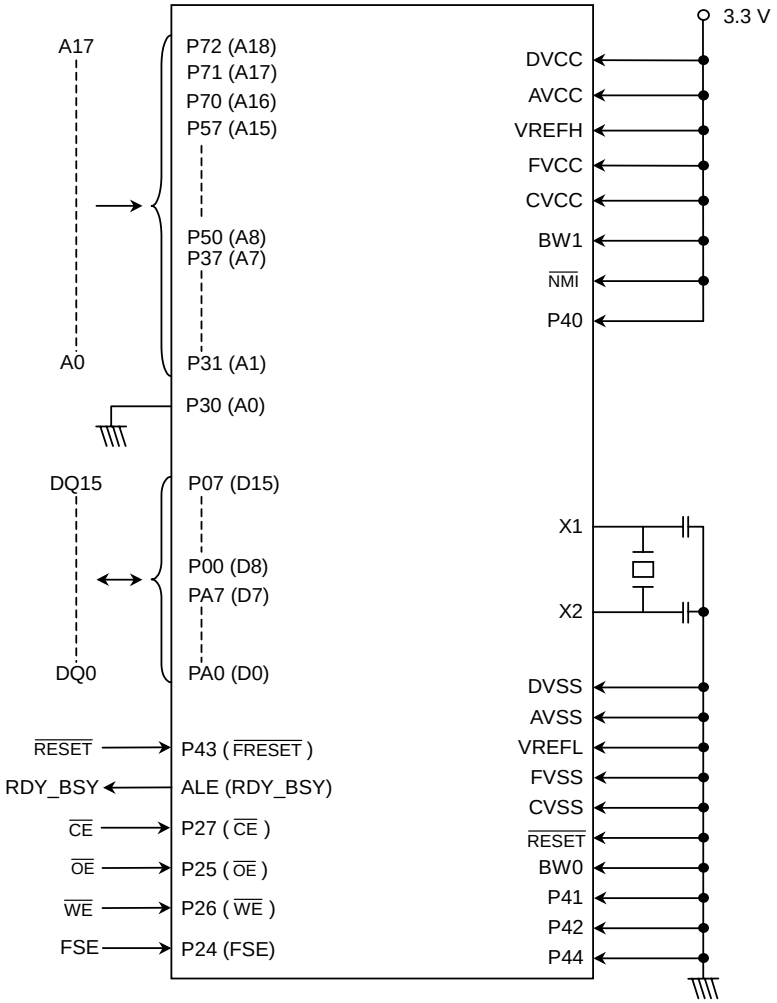


Figure 3.25 Pin Settings for Programmer Mode

3.7.2 Memory Maps

Figure 3.26 shows a comparison of memory maps in Single-Chip (Normal) and Programmer modes. In Programmer mode, the on-chip flash memory is mapped to physical addresses 0x0000_0000 through 0x0007_FFFF. In Programmer mode, all reads and writes use 16-bit halfword accesses aligned on an even-byte boundary.

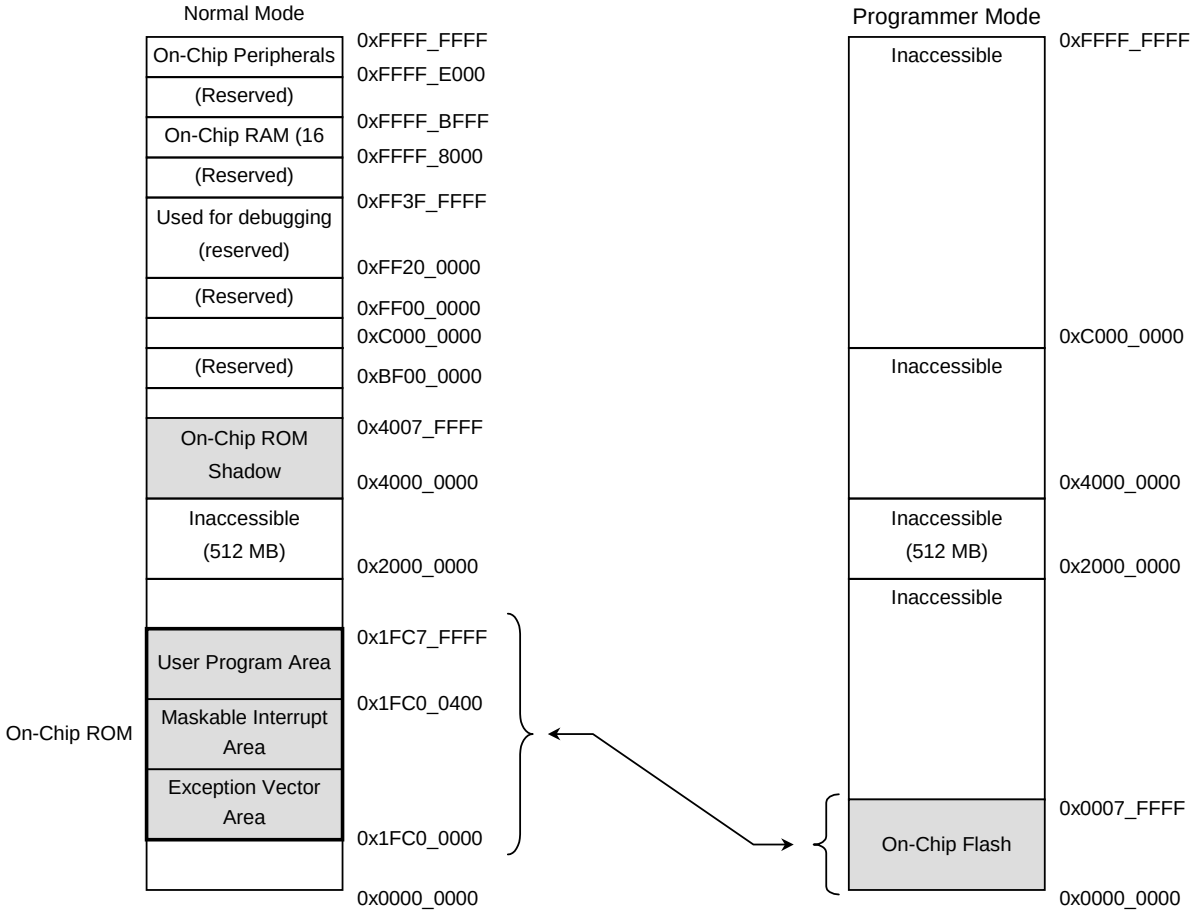


Figure 3.26 Memory Maps in Normal and Programmer Modes

3.7.3 Pin Functions and Settings

Table 3.20 EPROM Programmer Connections

EPROM Programmer	TMP1940F DBF	Function	EPROM Programmer	TMP1940F DBF	Function
GND	P30	Address bus (Input)	DQ0	PA0	Data bus (Input/output)
A0	P31		DQ1	PA1	
A1	P32		DQ2	PA2	
A2	P33		DQ3	PA3	
A3	P34		DQ4	PA4	
A4	P35		DQ5	PA5	
A5	P36		DQ6	PA6	
A6	P37		DQ7	PA7	
A7	P50		DQ8	P00	
A8	P51		DQ9	P01	
A9	P52		DQ10	P02	
A10	P53		DQ11	P03	
A11	P54		DQ12	P04	
A12	P55		DQ13	P05	
A13	P56		DQ14	P06	
A14	P57		DQ15	P07	
A15	P70		$\overline{CE}$	P27	Chip Enable input
A16	P71		$\overline{WE}$	P26	Write Enable input
A17	P72		$\overline{OE}$	P25	Output Enable input
$\overline{RESET}$	P43	Hardware reset input	FSE	P24	Flash Security Enable input
RDY/BSY	ALE	Ready/Busy output			
GND	FVSS	Ground	VCC (3.3 V)	FVCC (3.3 V)	Power supply (+3.3 V)

Table 3.21 Settings of the Other Pins

Pin Name	# of Pins	Type	Setting
$\overline{\text{RESET}}$	1	Input	Tie to logic 0 (0 V). (Programmer mode setting)
BW0	1	Input	Tie to logic 0 (0 V). (Programmer mode setting)
BW1	1	Input	Tie to logic 1 (3.3 V). (Programmer mode setting)
P40	1	Input	Tie to logic 1 (3.3 V). (Programmer mode setting)
P41, P42	2	Input	Tie to logic 0 (0 V). (Programmer mode setting)
$\overline{\text{NMI}}$	1	Input	Tie to logic 1 (3.3 V).
X1	1	Input	Connect a 20-MHz crystal for self-oscillation.
X2	1	Output	
P44	1	Input	Tie to logic 0 (0 V).
P10–P17	8	Input	Tie to logic 1 (3.3 V).
P20–P23	4	Input	Tie to logic 1 (3.3 V).
P73	1	Input	Tie to logic 0 (0 V).
P74	1	Output	Leave unconnected.
P75–P77	3	Input	Tie to logic 1 (3.3 V).
P80–P87	8	Input	Tie to logic 1 (3.3 V).
P90–P95	6	Input	Tie to logic 1 (3.3 V).
P96, P97	2	Input	Tie to logic 0 (0 V).
$\overline{\text{PLLOFF}}$	1	Input	Tie to logic 0 (0 V).
TEST	1	Input	Tie to logic 0 (0 V).
DVCC	3	Input	Tie to logic 1 (3.3 V).
DVSS	3	Input	0 V
CVCC	1	Input	Tie to logic 1 (3.3 V).
CVSS	1	Input	0 V
AVCC	1	Input	Tie to logic 1 (3.3 V).
AVSS	1	Input	0 V
VREFH	1	Input	Tie to logic 1 (3.3 V).
VREFL	1	Input	0 V

3.7.4 Key Features

The TMP1940FDBF flash memory commands are in principle compatible with the standard JEDEC commands. After a command sequence is written, the flash memory does not require the system to provide further controls or timings. The flash memory initiates the embedded program or erase algorithm automatically. The entire flash memory or one or more flash blocks can be erased at a time.

Table 3.22 Flash Memory Features

Feature	Description
Auto Program	Programs and verifies the desired addressed halfword by halfword automatically.
Auto Chip Erase	Erases and verifies the entire memory array automatically.
Auto Block Erase	Erases and verifies all memory locations in the selected block automatically.
Auto Multi-Block Erase	Erases and verifies all memory locations in multiple selected blocks automatically.
Write operation status	Provides several status bits such as the Data Polling bit, which can be used to determine whether a program or erase operation is complete or in progress.
Security feature	Prevents intrusive access to the flash memory while in Programmer mode. When the security feature is turned off, the entire memory array is erased and verified automatically, regardless of whether a given block is protected or not.
Block-protection	Disables both program and erase operations in any block.

All accesses to the flash memory are performed halfword by halfword, including the writing of commands. Unless otherwise noted, the subsections that follow indicate addresses as seen from the programmer.

The program/erase operations of on-board programming modes are very similar to those of Programmer mode, with a few exceptions such as the data bus width. Refer to Section 3.6 for a description of the program and erase operations in on-board programming modes.

3.7.5 Block Architecture

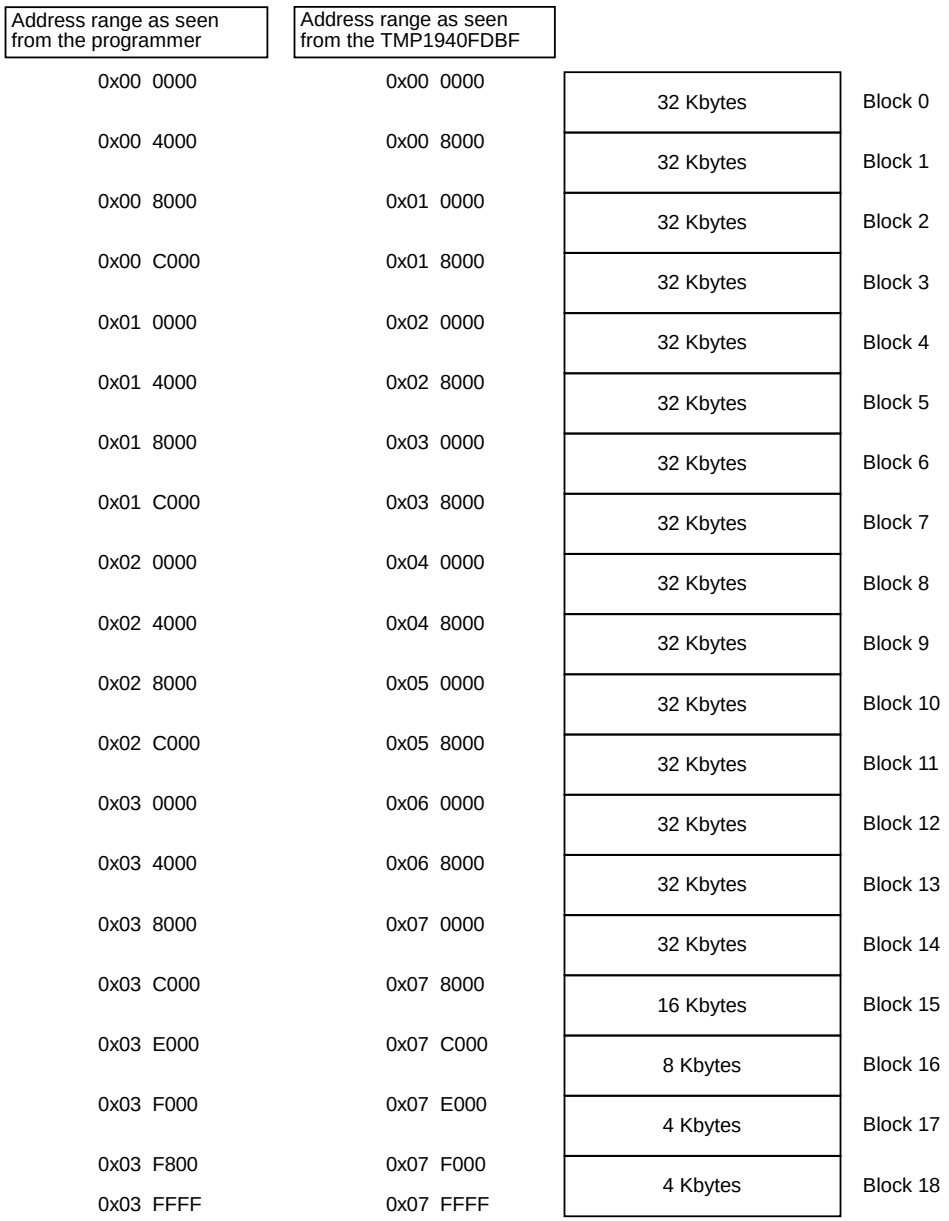


Figure 3.27 Flash Memory Block Architecture and Address Ranges in Programmer Mode

3.7.6 Read Mode and Embedded Operation Mode

The flash memory of the TMP1940FDBF has the following two modes of operation:

- Read mode in which array data is read
- Embedded Operation mode in which the flash memory is programmed or erased

The flash memory enters Embedded Operation mode when a valid command sequence is executed in Read mode. In Embedded Operation mode, array data can not be read. In Programmer mode, all bus cycles such as the writing of commands and the reading of data are performed as a 16-bit halfword quantity.

The flash memory has a security bit apart from the flash array. The reading of the flash array can be disabled in Programmer mode by programming this bit. In Programmer mode, the FSE pin is used for this purpose. For a detailed description, see Section 3.7.17. In Normal operation mode, the FSE pin must be held at the V_{IL} level to access the flash array. During any operation, the FSE pin must remain stable.

3.7.7 Reading Array Data

The flash memory is automatically set to reading array data upon CPU reset after device power-up and after an embedded operation is successfully completed.

3.7.8 Writing commands

The operations of the flash memory are selected by commands or command sequences written into the internal command register. This uses the same mechanism as for JEDEC-standard EEPROMs. Commands are made up of data sequences written at specific addresses via the command register. See Table 3.25 on page 81 for the list of command sequences.

The command sequence being written can be canceled by issuing the Read/Reset command between sequence cycles. The Read/Reset command clears the command register and resets the flash memory to Read mode. Invalid command sequences also cause the flash memory to clear the command register and returns to Read mode.

3.7.9 Reset

- Read/Reset command (software reset)

  The flash memory does not return to Read mode if an embedded operation terminated abnormally. In this case, the Read/Reset command must be issued to put the flash memory back in Read mode. The Read/Reset command may also be written between sequence cycles of the command being written to clear the command register.

- Hardware reset

  The $\overline{\text{RESET}}$ pin provides a hardware method of terminating an embedded operation or clearing the internal command register being written. A reset is performed when the $\overline{\text{RESET}}$ pin is set to V_{IL} and kept low at least 500 ns. It takes 20 μm for a reset to complete and put flash memory in Read mode. An embedded operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

After a reset, the flash memory is set to Read mode if $\overline{RESET}$ is at the V_{IH} level and to Standby mode if $\overline{RESET}$ is at the V_{IL} level. While $\overline{RESET}$ is at the V_{IL} level, D0 to D15 are held at the high-impedance state. Any command sequence must be written after the flash memory is put back in Read mode.

3.7.10 Auto Program Command

In Programmer mode, the programming of the flash array is performed on a halfword-by-halfword basis. In the fourth bus cycle of the Auto Program command sequence, the program address is latched on the falling edge of $\overline{WE}$, and data is latched on the rising edge of $\overline{WE}$. The latching of the program data initiates the embedded Auto Program algorithm. The Auto Program command executes a sequence of internally timed events to program the desired bits of the addressed memory location and verify that the desired bits are sufficiently programmed. The system can determine the status of the programming operation by using write status flags (see Table 3.28 on page 82).

Any commands written during the programming operation are ignored. A hardware reset immediately terminates the programming operation. The programming operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables programming operations in any block. If an attempt is made to program a protected block, the Auto Program command does nothing; the flash memory returns to Read mode in approximately 3 μm after the rising edge of $\overline{WE}$ in the fourth bus cycle of the command sequence.

A bit must be programmed to change its state from a 1 to a 0. A bit can not be programmed from a 0 back to a 1. Only an erase operation can change a 0 back to a 1. A programming failure condition is indicated if the system tries to program a 1 to a location that was previously programmed to a 0. Note that this is not a device failure condition since the flash memory was used incorrectly.

When the embedded Auto Program algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the programming operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of a programming failure, it is recommended to replace the chip or discontinue the use of the failing flash block.

3.7.11 Auto Chip Erase Command

The embedded Auto Chip Erase algorithm is initiated on the rising edge of $\overline{WE}$ in the sixth bus cycle of the command sequence. The embedded Auto Chip Erase algorithm automatically preprograms the entire memory for an all-0 data pattern prior to the erase; then, it automatically erases and verifies the entire memory for an all-1 data pattern. The system can determine the status of the chip erase operation by using write status flags (see Table 3.28 on page 82).

Any commands written during the chip erase operation are ignored. A hardware reset immediately terminates the chip erase operation. The chip erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Chip Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the blocks are protected, the Auto Chip Erase command does nothing; the flash memory returns to Read mode in approximately 100 μm after the rising edge of $\overline{WE}$ in the sixth bus cycle of the command sequence.

When the embedded Auto Chip Erase algorithm is complete, the flash memory returns to Read mode.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. The failing block can be identified by means of the Auto Block Erase command.

3.7.12 Auto Block Erase and Auto Multi-Block Erase Commands

The address of the block to be erased is latched on the falling edge of $\overline{WE}$ in the sixth bus cycle of the command sequence. A time-out begins from the rising edge of that $\overline{WE}$ pulse. After a time-out, the erase operation will commence. The embedded Auto Block Erase algorithm automatically preprograms the selected block for an all-0 data pattern, and then erases and verifies that block for an all-1 data pattern.

During the time-out period, additional block addresses and Auto Block Erase commands may be written. For more on this, see Figure 3.29.

Any command other than Auto Block Erase during the time-out period resets the flash memory to Read mode. The block erase time-out period is 50 µm. The time-out window is reset on each rising edge of $\overline{WE}$. The system can determine the status of the erase operation by using write status flags (see Table 3.28 on page 82).

Any commands written during the block erase operation are ignored. A hardware reset immediately terminates the block erase operation. The block erase operation that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence because data may be corrupted.

The block protection feature disables erase operations in any block. The Auto Block Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the selected blocks are protected, the Auto Block Erase algorithm does nothing; the flash memory returns to Read in approximately 100 µm after the rising edge of $\overline{WE}$ in the final bus cycle of the command sequence.

If any failure occurs during the erase operation, the flash memory remains locked in Embedded Operation mode. The system can determine this status by using write status flags. To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. If any failure occurred during the multi-block erase operation, the failing block can be identified by running Auto Block Erase on each of the blocks selected for multi-block erasure.

3.7.13 Block Protect Command

The block protection feature disables both program and erase operations in any block. The effects of the program and erase commands on the protected blocks are summarized below.

Table 3.23 Effects of the Program and Erase Commands on the Protected Blocks

Command	Operation
Program command on a protected block	No programming operation is performed, and the flash memory automatically returns to Read mode.
Erase command on a protected block	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when all the blocks are protected	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.
Multi-Block Erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.

After the command sequence is complete, writing to the protect control logic is performed by pulsing $\overline{WE}$ for t_{PPLH} while $\overline{CE}$ is set to V_{IL} and the block address is placed on P70 (A16) to P54 (A12).

Any commands written during the Block Protect algorithm are ignored. A hardware reset immediately terminates the block protect operation. The Block Protect command that was interrupted should be re-initiated once the flash memory is ready to accept another command sequence (see Figure 3.31).

Note that the block protect operation is not verified automatically. The Verify Block Protect command must be written to verify the protect status after executing Block Protect. If the desired block is not in the protected state, the Block Protect command sequence must be re-initiated.

3.7.14 Block Unprotect Command

The Block Unprotect command unprotects all blocks simultaneously. All blocks must be protected before executing the Block Unprotect command. After the Block Unprotect command sequence is complete, block uprotection is performed by pulsing $\overline{WE}$ for t_{PULH} with $\overline{CE}$ set to V_{IL}.

Any commands written during the Block Unprotect algorithm are ignored. A hardware reset immediately terminates the block unprotect operation. The Block Unprotect command that was interrupted should be re-initiated from protecting all blocks. The Verify Block Protect command must be written to verify the protect status after executing Block Unprotect.

3.7.15 Verify Block Protect Command

The Verify Block Protect command is used to verify the protect status of a block. Verify Block Protect is a four-bus-cycle operation. The address of the block to be verified is given in the fourth cycle. Any address within the block range will suffice, provided A0 = 1 and A5 = 0. Data must be read as a 16-bit halfword. If the selected block is protected, a value of 0x0001 is returned. If the selected block is not protected, a value of 0x0000 is returned. Following the fourth bus cycle, an additional block address may be provided.

The Verify Block Protect command does not return the flash memory to Read mode. Either the Read/Reset command or a hardware reset is required to reset the flash memory to Read mode or to write the next command.

3.7.16 Write Operation Status

As shown in Table 3.28, the flash memory provides several flag bits to determine the status of an embedded operation: DQ7, DQ5, DQ3 and RDY_BSY. These status bits can be read during an embedded operation using the same timing as for Read mode by setting $\overline{CE}$ and $\overline{OE}$ to V_{IL}. The RDY_BSY status is valid after the rising edge of the final $\overline{WE}$ pulse in the command sequence,

regardless of $\overline{CE}$ and $\overline{WE}$. The flash memory automatically returns to Read mode when an embedded operation completes.

- DQ7 (Data Polling)

  The Data Polling bit, DQ7, indicates to the host system the status of the embedded operation. Data Polling is valid after the rising edge of the final $\overline{WE}$ pulse in the command sequence.

  When the embedded Program algorithm is in progress, an attempt to read the flash memory will produce the complement of the data last written to DQ7. Upon completion of the embedded Program algorithm, an attempt to read the flash memory will produce the true data last written to DQ7. Therefore, the system can use DQ7 to determine whether the embedded Program algorithm is in progress or completed.

  When the embedded Erase algorithm is in progress, an attempt to read the flash memory will produce a 0 at the DQ7 output. Upon completion of the embedded Erase algorithm, the flash memory will produce a 1 at the DQ7 output.

  If there is a failure during an embedded operation, DQ7 continues to output the same value. Thus, DQ7 must always be polled in conjunction with the Exceeded Timing Limits (DQ5) flag. Figure 3.30 shows the DQ7 polling algorithm.

  The flash memory disables address latching when an embedded operation is complete. Data polling must be performed with a valid programmed address or an address within any of the non-protected blocks selected for erasure. DQ7 may change asynchronously while $\overline{OE}$ is asserted low.

- DQ5 (Exceeded Timing Limits)

  DQ5 produces a 0 while the program or erase operation is in progress normally. DQ5 produces a 1 to indicate that the program or erase time has exceeded the specified internal limit. This is a failure condition that indicates the program or erase cycle was not successfully completed.

  The DQ5 failure condition also appears if the system tries to program a 1 to a location that was previously programmed to a 0. Only an erase operation can change a 0 back to a 1. In this case, the embedded Program algorithm halts the operation. Once the operation has exceeded the timing limits, DQ5 will indicate a 1. Note that this is not a device failure condition since the flash memory was used incorrectly.

  Under both these conditions, the flash memory remains locked in Embedded Operation mode. The system must issue the Read/Reset command to return the flash memory to Read mode.

- DQ3 (Block Erase Timer)

  The block erase time-out window begins from the rising edge of the $\overline{WE}$ pulse in the sixth bus cycle of the command sequence. The erase operation will begin after the time-out has expired. When the time-out is complete and the erase operation has begun, DQ3 switches from 0 to 1. If DQ3 is 0, the flash memory will accept additional Auto Block Erase commands. Each time an Auto Block Erase command is written, the time-out window is reset. To ensure that the command has been accepted, the system should check DQ3 prior to and following each Auto Block Erase command. If DQ3 is 1 on the second status check, the command might not have been accepted.

- RDY_BSY (Ready/Busy)

  In Programmer mode, the ALE pin functions as the RDY_BSY pin. The programming equipment can monitor the state of this pin to determine whether an embedded algorithm is in progress or complete. RDY_BSY produces a 0 when the flash memory is actively erasing or programming. RDY_BSY produces a 1 when an embedded operation has completed and the

flash memory is ready to accept the next command. If any failure occurs during the program or erase operation, this flag remains at the 0 logic state. Any command is ignored while RDY_BSY is at the 0 logic state. RDY_BSY is not a open-drain output pin, but a normal CMOS output pin.

3.7.17 Flash Security

The TMP1940FDBF flash memory has a security bit apart from the flash array. Programming this security bit disables access to the flash array. This prevents intrusive access to the flash memory by third parties while in Programmer mode.

- Securing the flash (Disabling read accesses)

  Securing the flash memory disables programming equipment to read its contents. To turn on the security feature, once programming is complete, write the Auto Security On command, with the FSE pin set to V_{IH}. In the fourth bus cycle of the command sequence, program 0x0098 at address 0x0000. After the rising edge of $\overline{WE}$ in the fourth bus cycle, the embedded Security On algorithm automatically programs and verifies the security bit.

  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. The FSE pin must be held stable throughout the embedded operation.

  When the embedded algorithm completes, the flash memory automatically returns to Read mode.

  If any failure occurs during the embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. Note that this is a security bit failure. If the flash memory needs to be secured, the chip should be replaced. When the security is on, any reads by programming equipment will always return a halfword-length value of 0x0098.

- Unsecuring the flash (Enabling read accesses)

  To turn off the security feature, write the Auto Security Off command, with the FSE pin set to V_{IH}. After the rising edge of $\overline{WE}$ in the sixth bus cycle of the command sequence, the embedded Security Off algorithm automatically erases and verifies the entire flash array, and then erases and verifies the security bit.

  Any commands written during the embedded operation are ignored. A hardware reset immediately terminates the embedded operation. In this case, if any erase operation is progress, data may be corrupted. The FSE pin must be held stable througout the embedded operation.

  When an embedded algorithm completes, the flash memory automatically returns to Read mode.

  If any failure occurs during an embedded operation, the flash memory remains locked in Embedded Operation mode and does not return to Read mode. The system can determine the status of the embedded operation by using write status flags. If a failure occurs in the memory array, the security bit is not erased. In this case, the security bit is left on. The chip should be replaced if a memory array or security bit failure occurs.

  The Auto Security Off command erases the flash array prior to turning off the security feature. Even if a given block is protected, it is unconditionally erased, but the protect status of that block remains unchanged. The Auto Security Off and Auto Chip Erase command sequences are the same. The only difference is that the Auto Security Off command requires the FSE pin to be

set to the V_{IH} level before the command is written. The Auto Block Erase command does not turn off the security feature even when the FSE pin is set to V_{IH}. If the Auto Block Erase command is written with the FSE input pin se to V_{IH}, no block will be erased and the operation is immediately terminated.

3.7.18 Command Definitions

Table 3.24 Basic Operation Modes (with Addresses as Seen from the Programmer)

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A5	A0	$\overline{RESET}$	DQ0 to DQ15
Read	0	0	1	A5	A0	1	Dout
Standby	1	X	X	X	X	1	Hi-Z
Output Disable	X	1	1	X	X	X	Hi-Z
Write	0	1	0	A5	A0	1	Din
Hardware Reset / Standby	X	X	X	X	X	0	Hi-Z

Table 3.25 Programmer Mode Command Definitions (with Addresses as Seen from the Programmer)

Command Sequence	Bus Cycles	1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/Write)		5th Cycle (Write)		6th Cycle (Write)	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read/Reset	1	0xXXXX	0xF0										
Read/Reset	3	0x5555	0xAA	0xAAAA	0x55	0x5555	0xF0	RA	RD				
Auto Program	4	0x5555	0xAA	0xAAAA	0x55	0x5555	0xA0	PA	PD				
Auto Chip Erase	6	0x5555	0xAA	0xAAAA	0x55	0x5555	0x80	0x5555	0xAA	0xAAAA	0x55	0x5555	0x10
Auto Block Erase	6	0x5555	0xAA	0xAAAA	0x55	0x5555	0x80	0x5555	0xAA	0xAAAA	0x55	BA	0x30
Block Protect	6	0x5555	0xAA	0xAAAA	0x55	0x5555	0x9A	0x5555	0xAA	0xAAAA	0x55	0x5555	0x9A
Verify Block Protect	4	0x5555	0xAA	0xAAAA	0x55	0x5555	0x90	BPA	BD				
Auto Security On (Note)	4	0x5555	0xAA	0xAAAA	0x55	0x5555	0xA0	0x0000	0x98				
Auto Security Off (Note)	6	0x5555	0xAA	0xAAAA	0x55	0x5555	0x80	0x5555	0xAA	0xAAAA	0x55	0x5555	0x10

Note: Write the command sequence with the FSE input pin set to V_{IH}. This enables access to the security bit. Write the other command sequences with the FSE input pin set to V_{IL}.

- 0xF0, 0xAA, 0x55, 0xA0, 0x80, 0x10, 0x30:
  Command data. Write command data as a half quantity, padding the upper byte with 0x00.
- RA: Read Address
  RD: Read Data
- PA: Program Address
  PD: Program Data
  Write data on a halfword-by-halfword basis.
- BA: Block Address (BA0–BA6)
  Refer to Table 3.27.
- BPA: Verify Block Protect Address
  BD: Block Protect Data

Refer to Table 3.27. The address of the block to be verified can be any of the addresses within the block, with A5 = 0 and A0 = 1. If a block is protected, a value of 0x0001 will be returned. If a block is not protected, a value of 0x0000 will be returned.

Table 3.26 below shows the relationships between the addresses as seen from the programmer and the TMP1940FDBF.

Table 3.26 Relationship between addresses

	Command Address																				
Programmer	—	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	—
TMP1940FDBF	P73 ~ P77	P72 / A18	P71 / A17	P70 / A16	P57 / A15	P56 / A14	P55 / A13	P54 / A12	P53 / A11	P52 / A10	P51 / A9	P50 / A8	P37 / A7	P36 / A6	P35 / A5	P34 / A4	P33 / A3	P32 / A2	P31 / A1	P30 / A0	
0xXXXX	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	
0x0000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xAAAA		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	
0x5555		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	

Table 3.27 Block Erase Addresses in Programmer Mode (as Seen from the Programmer)

Block	Address Range	Size
BA0	0x0000 thru 0x3FFF	32 Kbytes
BA1	0x4000 thru 0x7FFF	32 Kbytes
BA2	0x8000 thru 0xBFFF	32 Kbytes
BA3	0xC000 thru 0xFFFF	32 Kbytes
BA4	0x1000 thru 0x13FFF	32 Kbytes
BA5	0x14000 thru 0x17FFF	32 Kbytes
BA6	0x18000 thru 0x1BFFF	
BA7		
BA8		
BA9		
BA10		
BA11		
BA12		
BA13		
BA14	0x38000 thru 0x3BFFF	32 Kbytes
BA15	0x3C000 thru 0x3DFFF	16 Kbytes
BA16	0x3E000 thru 0x3EFFF	8 Kbytes
BA17	0x3F000 thru 0x3F7FF	4 Kbytes
BA18	0x3F800 thru 0x3FFFF	4 Kbytes

The address of the block to be erased can be any of the addresses within that block. For example, to select BA0, provide any address in the range between 0x0000 and 0x3FFF.

Table 3.28 Write Status Flags

Status		D7 (DQ7)	D5 (DQ5)	D3 (DQ3)
Embedded operation in progress	Auto Program	$\overline{D7}$	0	0
	Auto Erase (during the time-out window)	0	0	0
	Auto Erase	0	0	1
Time-out in embedded operation	Auto Program	$\overline{D7}$	1	1
	Auto Erase	0	1	1

Note: D4 and D2–D0 are don't-cares.

3.7.19 Embedded Algorithms

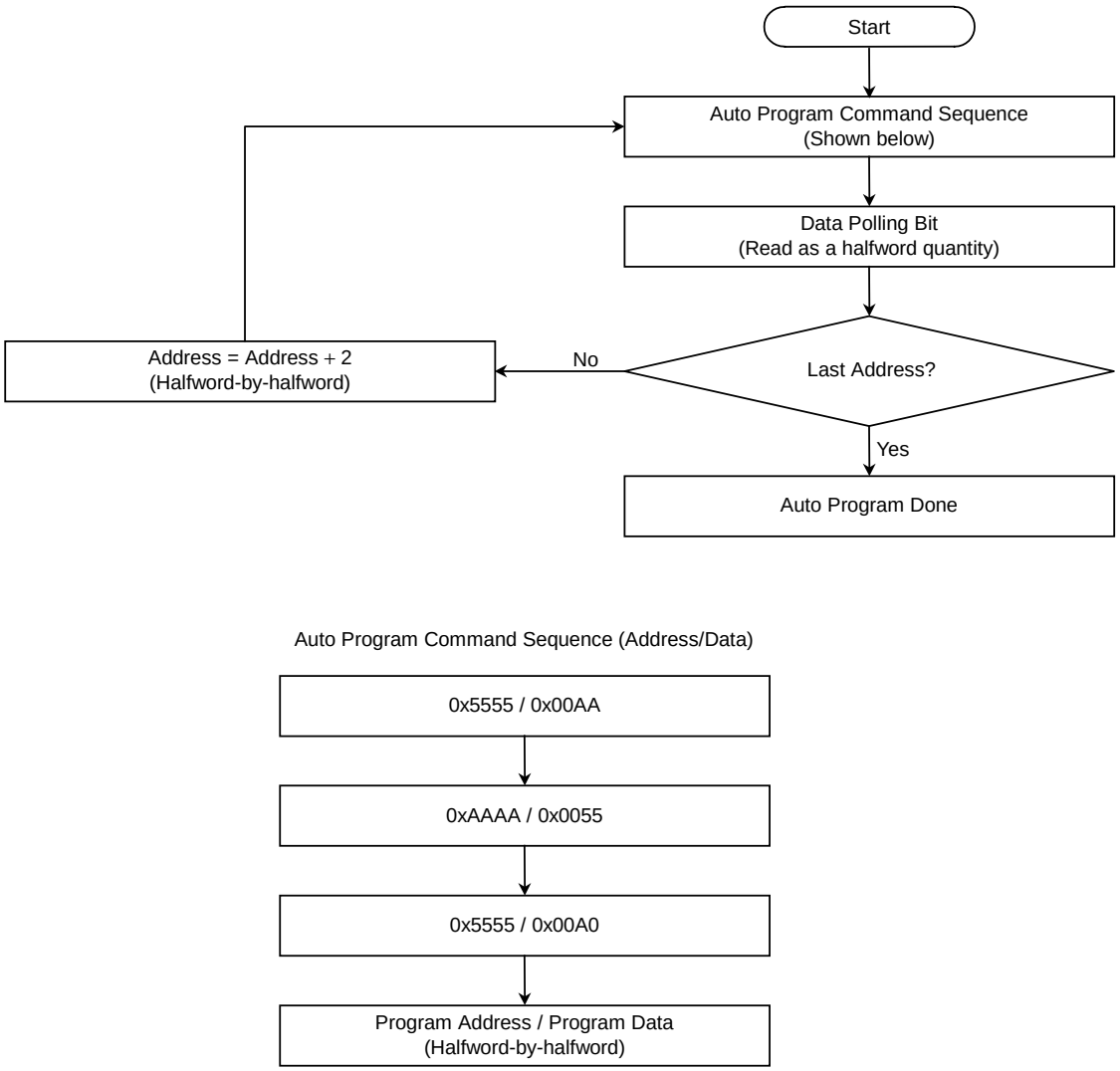


Figure 3.28 Auto Program Operation

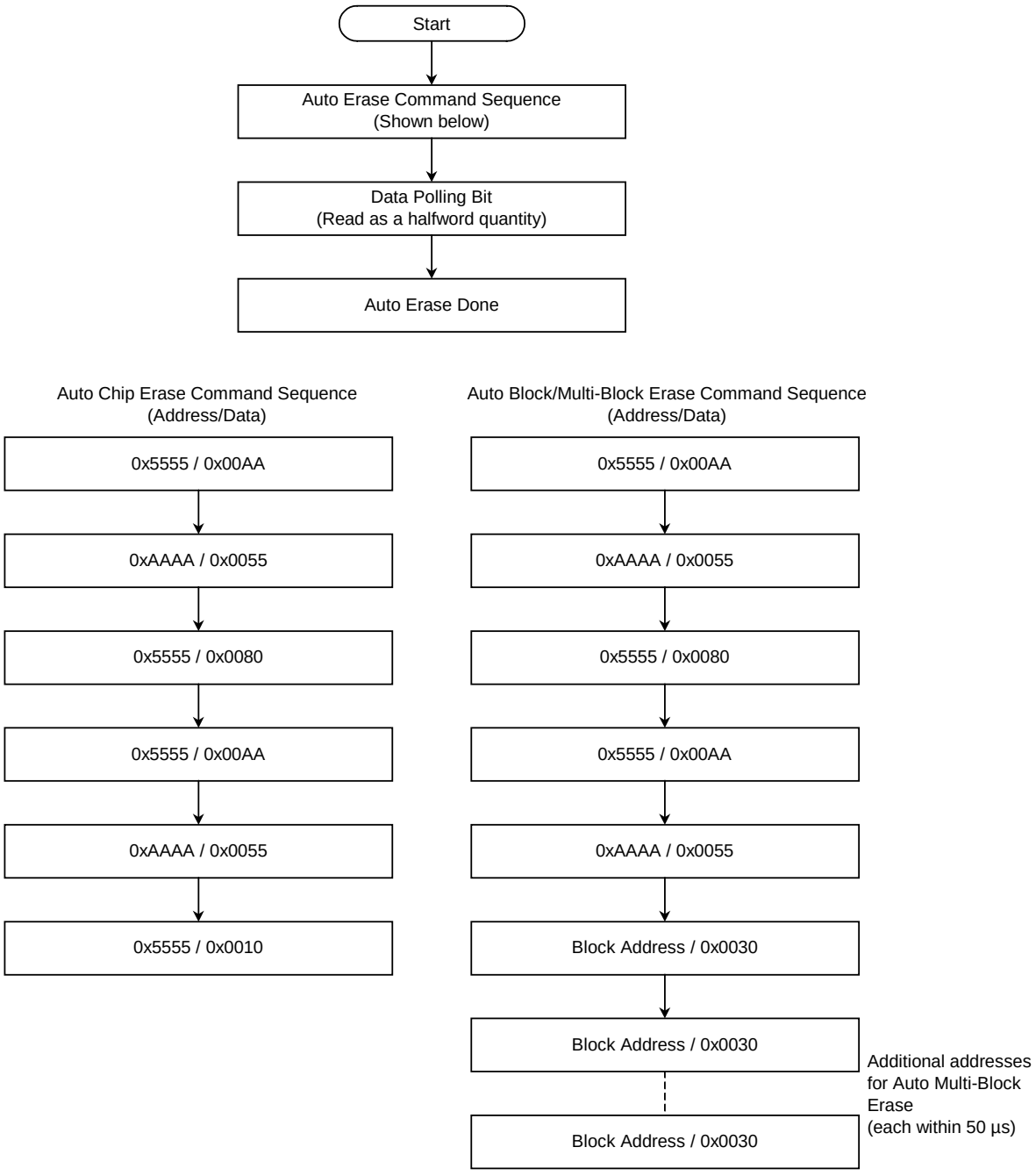


Figure 3.29 Auto Erase Operations

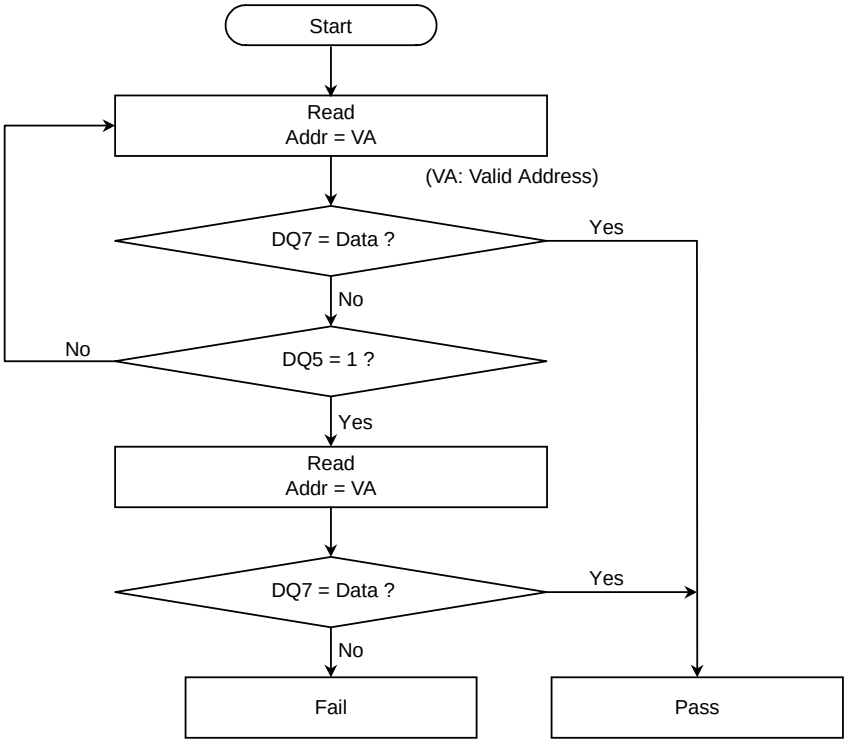


Figure 3.30 Data Polling (DQ7) Algorithm

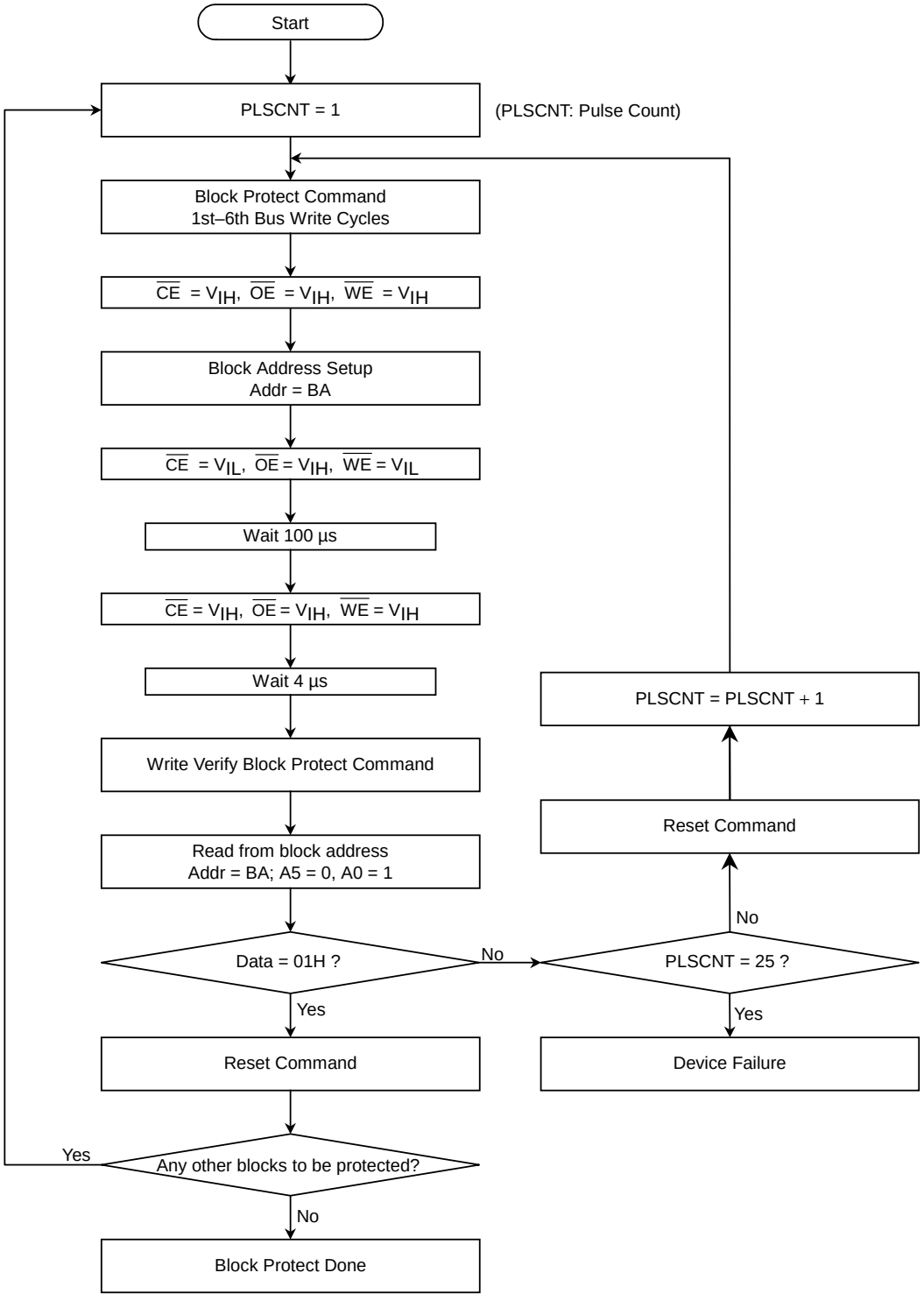


Figure 3.31 Block Protect Operation

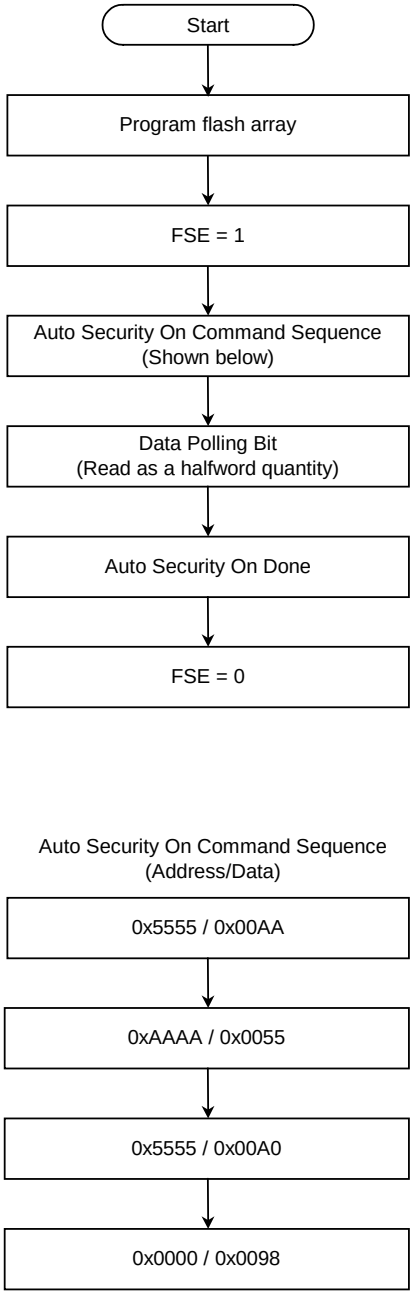
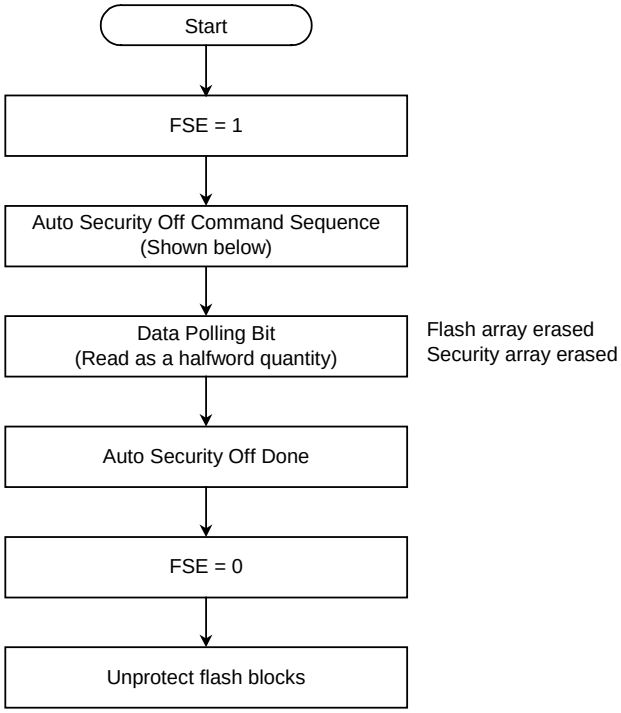


Figure 3.32 Auto Security On Operation



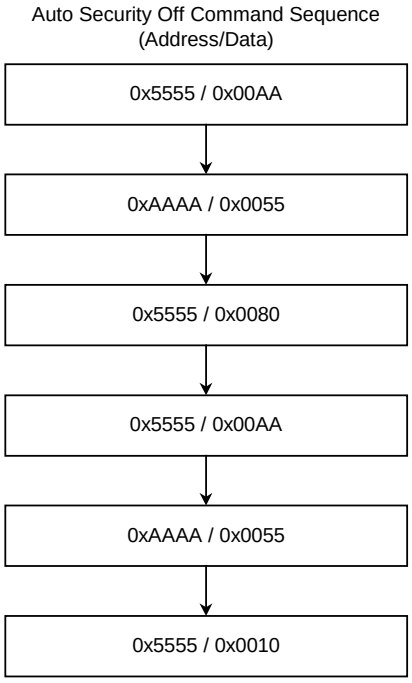


Figure 3.33 Auto Security Off Operation

4. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[1:0] = 00).

4.1 Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC}	–0.5 to 4.0	V
Input voltage		V_{IN}	–0.5 to V_{CC} + 0.5	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	80	
High-level output current	Per pin	I_{OH}	–5	
	Total	ΣI_{OH}	–80	
Power dissipation (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	–65 to 150	°C
Operating temperature	Except during flash W/E	T_{OPR}	–40 to 85	°C
	During flash W/E		0 to 70	
Write/erase cycles		N_{EW}	100	Cycles

Note: Maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no maximum rating value is exceeded with respect to current, voltage, power dissipation, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

4.2 DC Electrical Characteristics (1/3)

Ta = –40 to 85°C

Parameter		Symbol	Conditions		Min	Typ (Note 1)	Max	Unit
Supply voltage $AV_{CC} = V_{CC}$ $AV_{SS} = V_{SS} = 0$ V		V_{CC}	PLLON	fosc = 5 to 8 MHz fsys = 2.5 to 32 MHz fs = 30 to 34 kHz	3.0		3.6	V
				fosc = 5 to 6.5 MHz fsys = 2.5 to 26 MHz fs = 30 to 34 kHz	2.7			
			PLLOFF (Crystal)	fosc = 16 to 20 MHz fsys = 1 to 20 MHz fs = 30 to 34 kHz	2.7			
			PLLOFF (External clock)	fosc = 16 to 20 MHz fsys = 1 to 20 MHz fs = 30 to 34 kHz	2.7			
				fosc = 20 to 32 MHz fsys = 1.25 to 16 MHz fs = 30 to 34 kHz (SYSCR1.DFOSC = 0) (Note 2)				
Low-level input voltage	P00–P17 (AD0– AD15)	V_{IL}	$V_{CC} \geq 2.7$ V		–0.3		0.6	V
	P20–PA7 (except P77)	V_{IL1}					$0.3V_{CC}$	
	$\overline{PLLOFF}$, BW0, BW1, $\overline{RESET}$, $\overline{NMI}$, P77 (INT0)	V_{IL2}					$0.25V_{CC}$	
	X1	V_{IL4}					$0.2V_{CC}$	
High-level input voltage	P00–P17 (AD0– AD15)	V_{IH}			2.0		$V_{CC} + 0.3$	
	P20–PA7 (except P77)	V_{IH1}			$0.7V_{CC}$			
	$\overline{PLLOFF}$, BW0, BW1, $\overline{RESET}$, $\overline{NMI}$, P77 (INT0)	V_{IH2}			$0.80V_{CC}$			
	X1	V_{IH4}			$0.8V_{CC}$			
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	$V_{CC} \geq 2.7$ V			0.45	V
High-level output voltage		V_{OH}	I_{OH} = –400 μA		2.4			

Note 1: V_{CC} = 3.3 V, Ta = 25°C, unless otherwise noted.
Note 2: The DFOSC bit in the SYSCR1 register must be cleared to 0.
Note 3: Tie INTLV high (Interleave mode) when fsys is greater than 20 MHz.
When INTLV is low (i.e., non-interleaved mode), the following conditions must be satisfied:
16 MHz < fsys ≤ 20 MHz at 3.0–3.6 V
fsys ≤ 16 MHz at 2.7–3.6 V

4.3 DC Electrical Characteristics (2/3)

Ta = –40 to 85°C

Parameter	Symbol	Conditions	Min	Typ. (Note 1)	Max	Unit
Input leakage current	I_{LI}	$0.0 \le V_{IN} \le V_{CC}$		0.02	± 5	μA
Output leakage current	I_{LO}	$0.2 \le V_{IN} \le V_{CC} - 0.2$		0.05	± 10	
Power-down voltage (while RAM is being backed up in STOP Mode)	V_{STOP}	$V_{IL2} = 0.2V_{CC}$, $V_{IH2} = 0.8V_{CC}$	2.2		3.6	V
Reset pull-up resistor	RRST	V_{CC} = 3.3 V ± 0.3 V	100		550	kΩ
Pin capacitance (except power supply pins)	C_{IO}	fc = 1 MHz			10	pF
Schmitt Width $\overline{\text{PLLOFF}}$, BW0, BW1, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0	V_{TH}	$V_{CC} \ge 2.7$ V	0.4			V
Programmable pull-up resistor	PKH	V_{CC} = 3.3 V ± 0.3 V	100		550	kΩ
NORMAL (Note 2) when gear ratio is 1/1	I_{CC}	V_{CC} = 3.3V ± 0.3 V, f_{sys} = 32 MHz (f_{osc} = 8 MHz, PLLON), INTLV = H		85	100	mA
IDLE (Doze)				35	50	
IDLE (Halt)				32	42	
NORMAL (Note 2) when gear ratio is 1/1		V_{CC} = 3.3 V ± 0.3 V, f_{sys} = 20 MHz (f_{osc} = 20 MHz, PLLOFF), INTLV = H		65	78	mA
IDLE (Doze)				28	40	
IDLE (Halt)				25	35	
SLOW (Note 3)		V_{CC} = 3.3 V ± 0.3 V, fs = 32.768 kHz		23	30	mA
SLEEP (Note 3)		V_{CC} = 3.3 V ± 0.3 V, fs = 32.768 kHz		4	85	μA
STOP		V_{CC} = 2.7 ~ 3.6 V		0.5	60	μA

Note 1: V_{CC} = 3.3 V, Ta = 25°C, unless otherwise noted.
Note 2: Measured with the CPU operating; two TMRAs, one TMRB and DMAC channel on; and input pin levels held at fixed logic levels. IREF excluded.
Note 3: Measured with RTC on and low-speed oscillator drive capability reduced to low (SYSCR2.DRVOSCL = 1).

4.4 DC Electrical Characteristics (3/3)

4.4.1 DC Electrical Characteristics in Modes Except Programmer Mode

Ta = –40 to 85°C (0 to 70°C during program and erase of the flash memory), V_{CC} = 2.7 to 3.6 V)

Symbol	Parameter	Condition	Min	Max	Unit
I_{DDO1}	Active write current	f_{sys} = 32 MHz	—	150	mA

4.4.2 DC Electrical Characteristics in Programmer Mode

(Ta = 25 ± 5°C, VCC = 2.7 to 3.6 V)

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	High-level input voltage	—	0.7 × V_{CC}	V_{CC} + 0.5	V
V_{IL}	Low-level input voltage	—	–0.3	0.8	V
I_{LI}	Input leakage current	0 V ≤ V_{IN} ≤ V_{CC}	—	±1	μA
I_{LO}	Output leakage current	0 V ≤ V_{OUT} ≤ V_{CC}	—	±1	μA
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA	V_{CC} – 0.4	—	V
		I_{OH} = –2.5 mA	0.85 × V_{CC}	—	
V_{OL}	Low-level output voltage	I_{OL} = 4.0 mA	—	0.4	V
I_{DDO1}	Active write current	t_{CYC} = t_{RC} (min)	—	50	mA

4.5 Precautions for Programming and Erasing the Flash Memory

- In on-board programming modes (Single Boot mode and User Boot mode), the flash program and erase operations must be given the highest priority. All interrupts including NMI must be disabled.
- An auto erase operation is required before performing an auto program operation on addresses that have already been programmed.
- It is recommended to perform an auto erase operation followed by an auto program operation when re-programming the flash memory using programming equipment once it has been programmed or erased in an on-board programming mode.

4.6 AC Characteristics in Programmer Mode

Symbol	Parameter	Min	Max	Unit
t_{RC}	Read cycle time	120	—	ns
t_{ACC}	Address access time	—	120	ns
t_{CE}	$\overline{CE}$ access time	—	120	ns
t_{OE}	$\overline{OE}$ access time	—	50	ns
t_{CEE}	$\overline{CE}$ to output low-Z	0	—	ns
t_{OEE}	$\overline{OE}$ to output low-Z	0	—	ns
t_{OEH}	$\overline{OE}$ hold time (read)	0	—	ns
t_{OH}	Output hold time	0	—	ns
t_{DF1}	$\overline{CE}$ to output high-Z	—	30	ns
t_{DF2}	$\overline{OE}$ to output high-Z	—	30	ns
t_{CMD}	Command cycle time	120	—	ns
t_{AS}	Address setup time	0	—	ns
t_{AH}	Address hold time	50	—	ns
t_{DS}	Data setup time	60	—	ns
t_{DH}	Data hold time	0	—	ns
t_{WELH}	$\overline{WE}$ pulse width	50	—	ns
t_{WEHH}	$\overline{WE}$ pulse width high	20	—	ns
t_{CES}	$\overline{CE}$ setup time	0	—	ns
t_{CEH}	$\overline{CE}$ hold time	0	—	ns
t_{OES}	$\overline{OE}$ setup time	0	—	ns
t_{OEHP}	$\overline{OE}$ hold time (data polling and toggle)	10	—	ns
t_{OEHT}	$\overline{OE}$ pulse width high (toggle)	20	—	ns
t_{PPW}	Auto Program time	16 (Note1)	—	µs
t_{PCEW}	Auto Chip Erase time	30 (Note1)	—	s
t_{PBEW}	Auto Block Erase time	3 (Note1)	—	s
t_{VDS}	DVCC (3.3 V) setup time	500	—	µs
t_{BUSY}	Program/erase valid to RDY_BSY delay	20	—	ns
t_{RP}	$\overline{RESET}$ pulse width	6	—	µs
t_{READY}	$\overline{RESET}$ low to Read mode	—	20	µs
t_{RB}	RDY/BSY recovery time	0	—	ns
t_{RH}	$\overline{RESET}$ recovery time	500	—	ns
t_{PPLH}	$\overline{WE}$ pulse width (Block Protect)	100	—	µs
t_{PAS}	Protect address setup time	0	—	ns
t_{PAH}	Protect address hold time	0	—	ns
t_{CESP}	$\overline{CE}$ setup time (Block Protect)	4	—	µs
t_{CEHP}	$\overline{CE}$ hold time (Block Protect)	8	—	µs

Note 1: Typical values
Note 2: AC measurement conditions are:
- **Input pulse levels: 2.4–0.4 V**
- **Input pulse rise and fall times (10 to 90%): 5 ns**
- **Input timing measurement reference levels: 1.5 V**
- **Output timing measurement reference levels: 1.5 V**
- **Output load capacitance (CL): 100 pF**

Note 3: Other AC characteristics are the same as for the TMP1940CYAF.

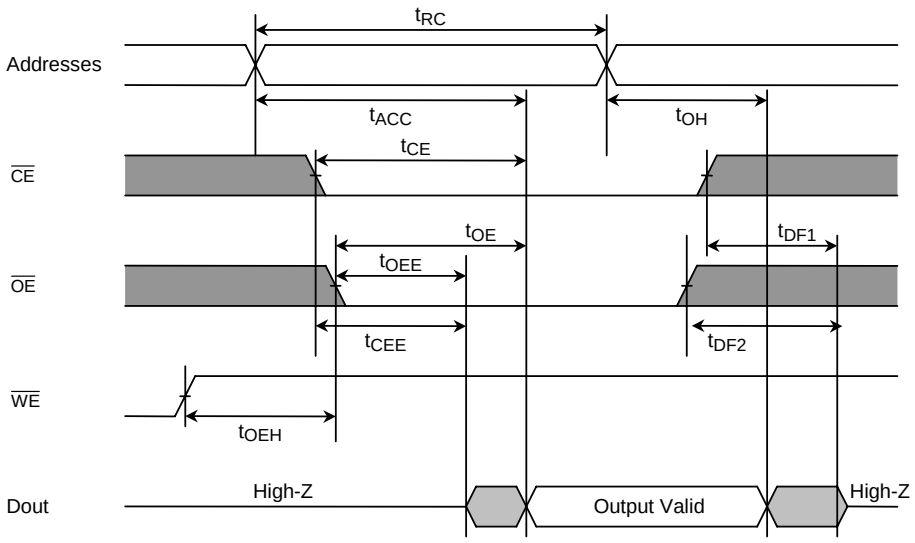
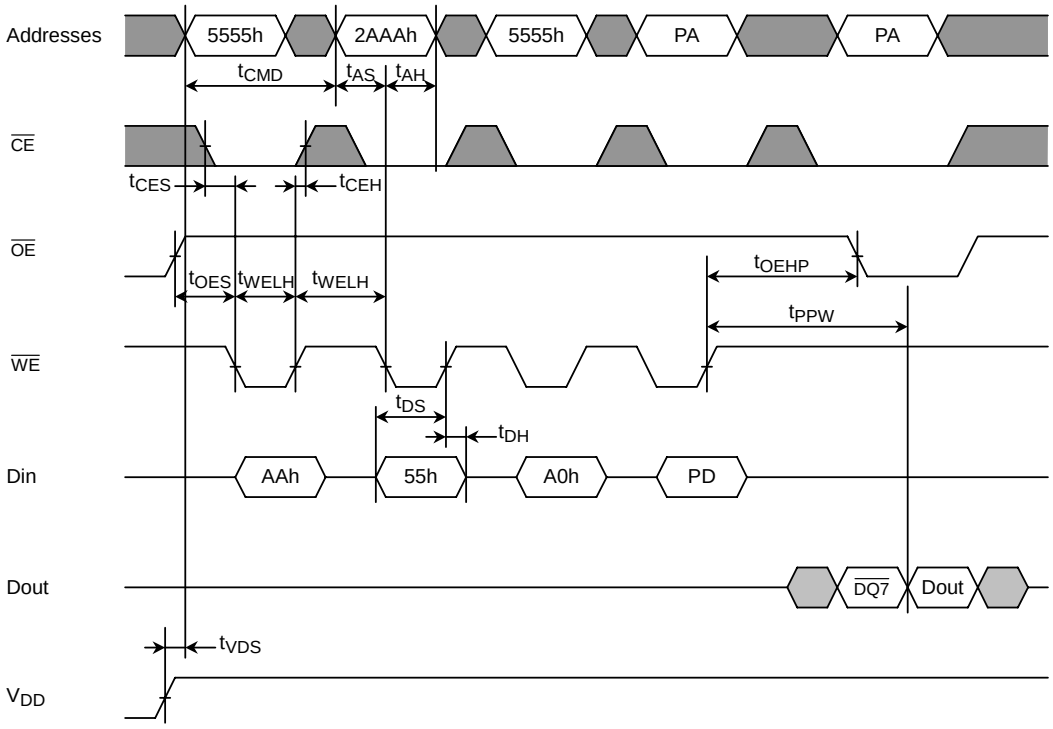
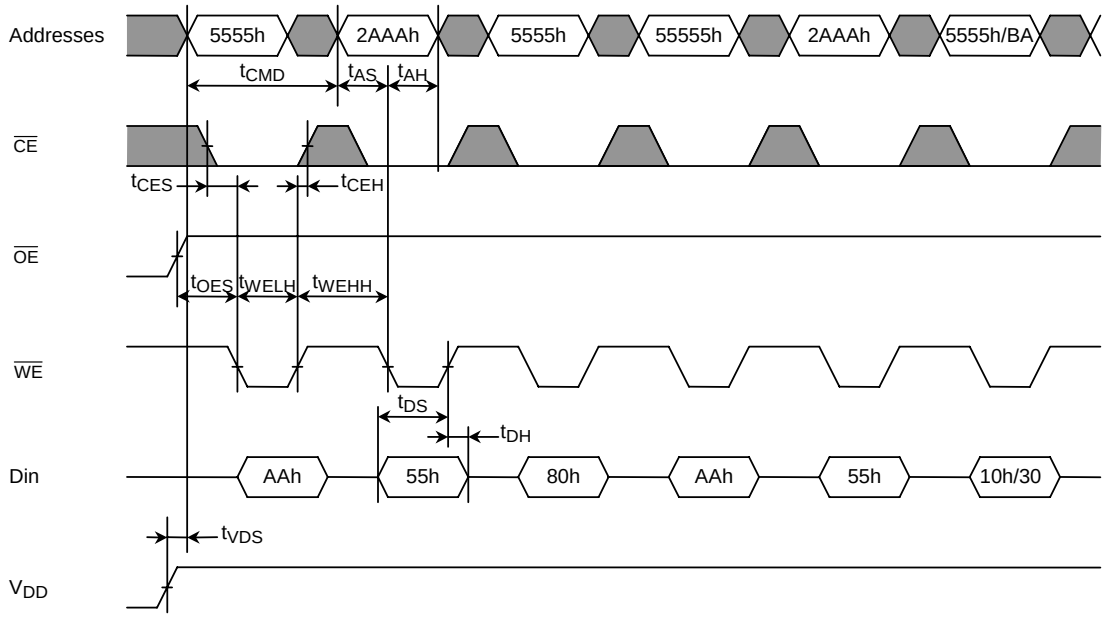


Figure 4.1 Read Operation Timings



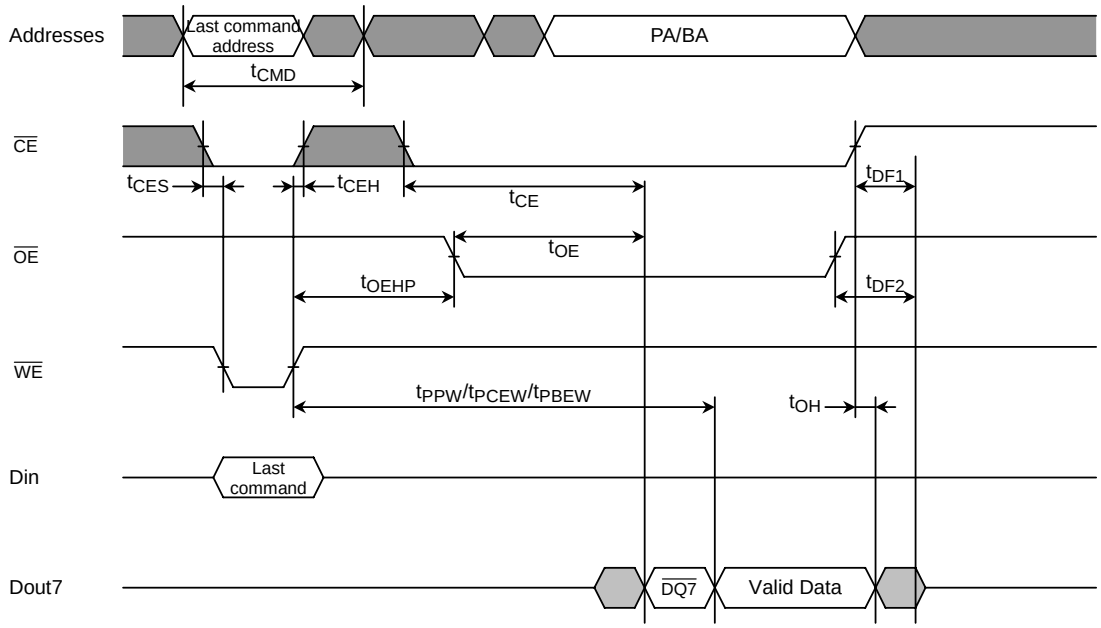
PA = Program address, PD = Program data

Figure 4.2 Auto Program Operation Timings



BA = Block address for Auto Block Erase

Figure 4.3 Auto Chip/Block Erase Operation Timings



PA = Program address, BA = Block address

Figure 4.4 Data Polling Timings During Embedded Algorithms

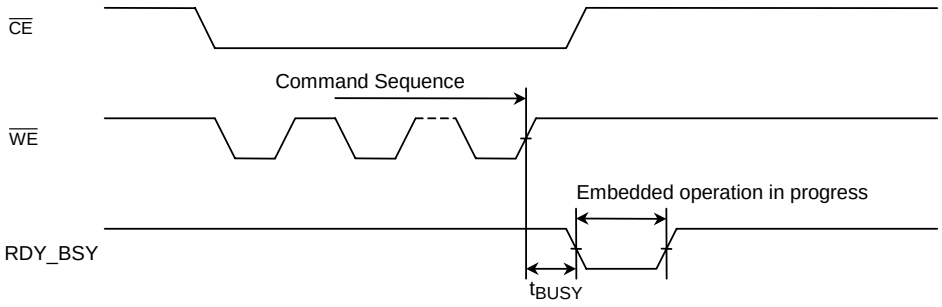


Figure 4.5 RDY_BSY Status Timings During Embedded Operations

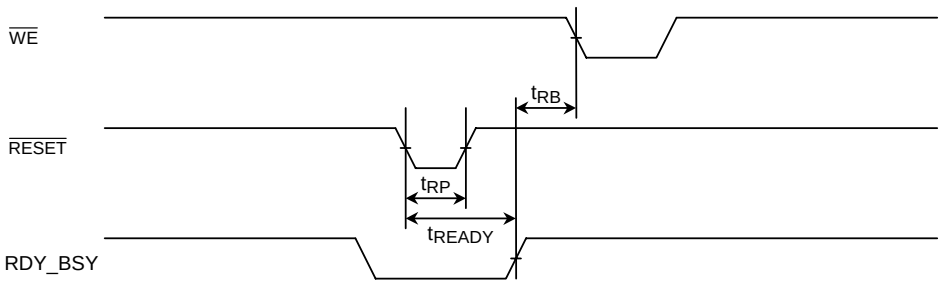


Figure 4.6 Hardware Reset Timings

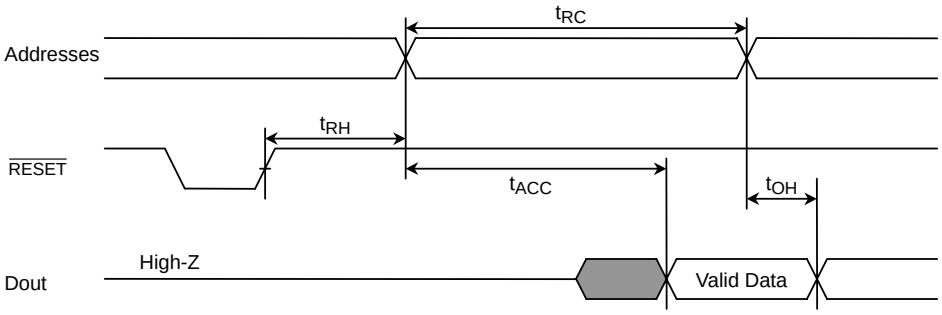


Figure 4.7 Read Timings After $\overline{\text{RESET}}$

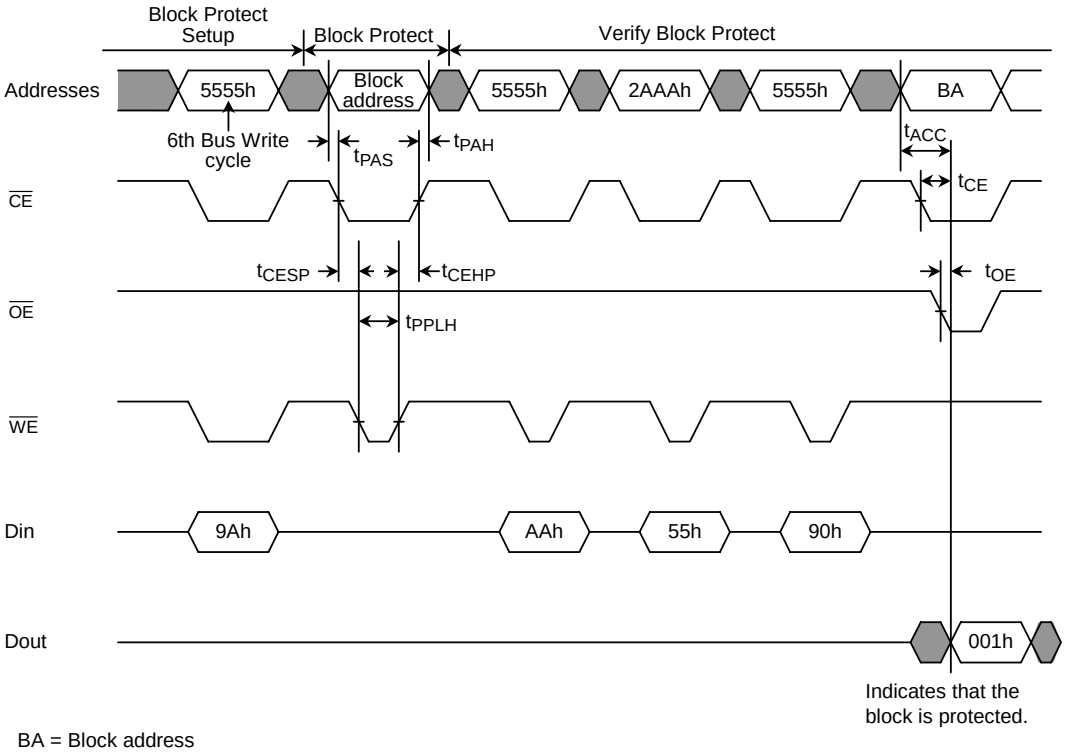


Figure 4.8 Block Protect Timings

TOSHIBA

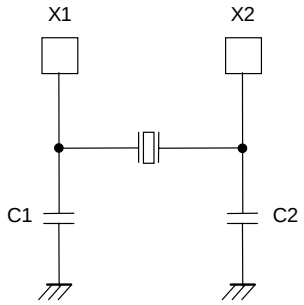
Part 2 Applications

TOSHIBA CORPORATION

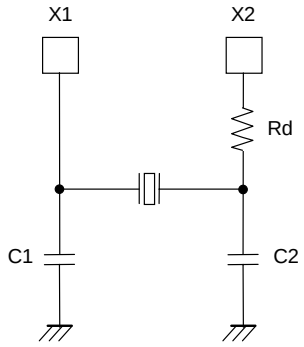
Clock and Reset Circuitry

(1) Sample Crystal Circuit

The TMP1940 series has an on-chip oscillation circuitry. An external crystal connected between the X1 and X2 pins can be used as a reference frequency source.

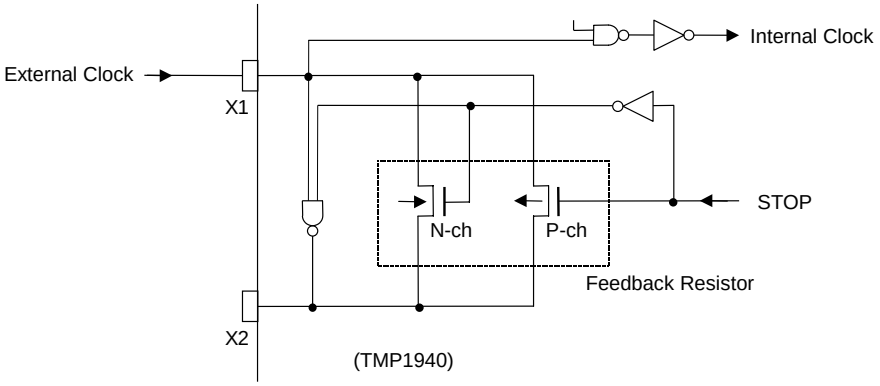


(2) Ceramic Resonator Circuit



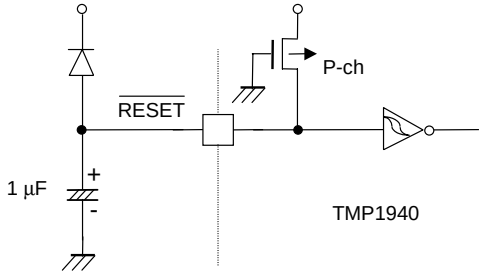
Consult with the manufacturer for specific information about the C1, C2 and Rd components.

(3) Recommended External Clock Connection



To operate the TMP1940 from an external clock, connect the clock source to the X1 pin, as shown above.

(4) Power-On Reset Circuit



TOSHIBA

Part 3 Packaging Information

TOSHIBA CORPORATION

100-Pin LQFP: TMP1940CYAF/TMP1940FDBF

Package Code: LQFP100-P-1414-0.50C

Unit: mm