TOSHIBA

64-Bit TX System RISC
TX49 Family
TMPR4925

Rev. 1.0

TOSHIBA CORPORATION

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

The information contained herein is subject to change without notice.

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

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

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

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

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

Preface

Thank you for new or continued patronage of TOSHIBA semiconductor products. This is the 2003 edition of the user's manual for the TMPR4925 64-bit RISC microprocessor.

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

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

February 2003

Table of Contents

Handling precautions

TX4925

Handling Precautions

1. Using Toshiba Semiconductors Safely

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

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

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

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

2. Safety Precautions

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

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

[Explanation of labels]

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

[Explanation of graphic symbol]

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

2.1 General Precautions regarding Semiconductor Devices

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

2.2 Precautions Specific to Each Product Group

2.2.1 Optical semiconductor devices

⚠ DANGER

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

⚠ WARNING

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

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

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

2.2.2 Power devices

⚠ DANGER

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

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

⚠WARNING

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

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

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

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

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

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

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

⚠CAUTION

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

2.2.3 Bipolar ICs (for use in automobiles)

⚠CAUTION

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

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

3. General Safety Precautions and Usage Considerations

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

3.1 From Incoming to Shipping

3.1.1 Electrostatic discharge (ESD)

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

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

(1) Work environment

- When humidity in the working environment decreases, the human body and other insulators can easily become charged with static electricity due to friction. Maintain the recommended humidity of 40% to 60% in the work environment, while also taking into account the fact that moisture-proof-packed products may absorb moisture after unpacking.
- Be sure that all equipment, jigs and tools in the working area are grounded to earth.
- Place a conductive mat over the floor of the work area, or take other appropriate measures, so that the floor surface is protected against static electricity and is grounded to earth. The surface resistivity should be 10^4 to 10^8 Ω/sq and the resistance between surface and ground, 7.5×10^5 to 10^8 Ω
- Cover the workbench surface also with a conductive mat (with a surface resistivity of 10^4 to 10^8 Ω/sq, for a resistance between surface and ground of 7.5×10^5 to 10^8 Ω) . The purpose of this is to disperse static electricity on the surface (through resistive components) and ground it to earth. Workbench surfaces must not be constructed of low-resistance metallic materials that allow rapid static discharge when a charged device touches them directly.
- Pay attention to the following points when using automatic equipment in your workplace:

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

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

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

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

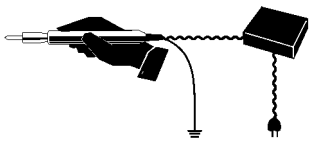
(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

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

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

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

(2) Operating environment



- Operators must wear anti-static clothing and conductive shoes (or a leg or heel strap).
- Operators must wear a wrist strap grounded to earth via a resistor of about 1 MΩ.
- Soldering irons must be grounded from iron tip to earth, and must be used only at low voltages (6 V to 24 V).
- If the tweezers you use are likely to touch the device terminals, use anti-static tweezers and in particular avoid metallic tweezers. If a charged device touches a low-resistance tool, rapid discharge can occur. When using vacuum tweezers, attach a conductive chucking pat to the tip, and connect it to a dedicated ground used especially for anti-static purposes (suggested resistance value: 10^4 to 10^8 Ω).
- Do not place devices or their containers near sources of strong electrical fields (such as above a CRT).

- When storing printed circuit boards which have devices mounted on them, use a board container or bag that is protected against static charge. To avoid the occurrence of static charge or discharge due to friction, keep the boards separate from one other and do not stack them directly on top of one another.

- Ensure, if possible, that any articles (such as clipboards) which are brought to any location where the level of static electricity must be closely controlled are constructed of anti-static materials.

- In cases where the human body comes into direct contact with a device, be sure to wear anti-static finger covers or gloves (suggested resistance value: 10^8 Ω or less).

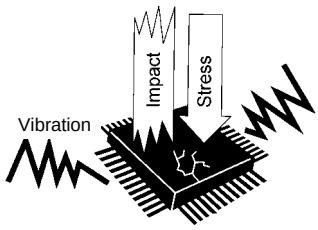
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.

- If a wrist strap cannot be used for some reason, and there is a possibility of imparting friction to devices, use an ionizer.

- The transport film used in TCP products is manufactured from materials in which static charges tend to build up. When using these products, install an ionizer to prevent the film from being charged with static electricity. Also, ensure that no static electricity will be applied to the product's copper foils by taking measures to prevent static occuring in the peripheral equipment.

3.1.2 Vibration, impact and stress

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



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

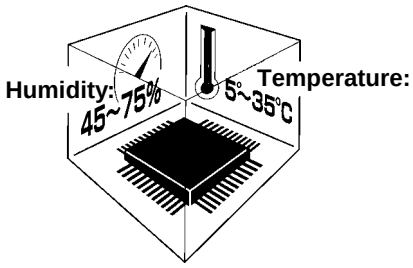
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

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

3.2 Storage

3.2.1 General storage

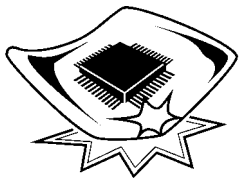
- Avoid storage locations where devices will be exposed to moisture or direct sunlight.
- Follow the instructions printed on the device cartons regarding transportation and storage.
- The storage area temperature should be kept within a temperature range of 5°C to 35°C, and relative humidity should be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



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

3.2.2 Moisture-proof packing

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



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

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

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

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

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

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

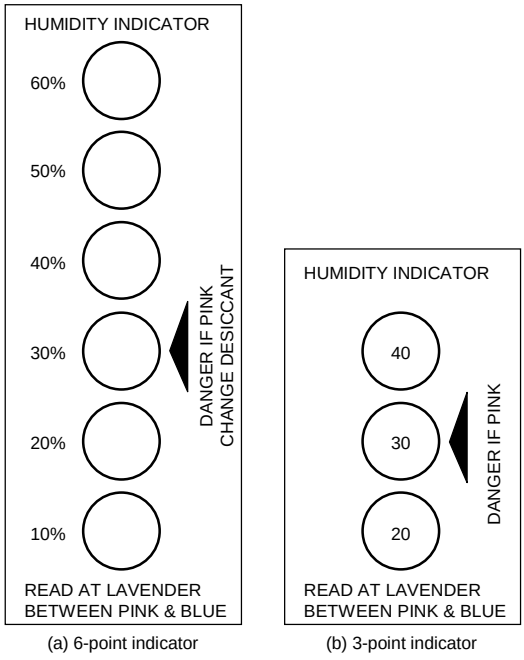


Figure 1 Humidity indicator

3.3 Design

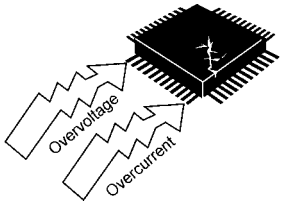
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will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

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

3.3.1 Absolute maximum ratings

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

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



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

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

3.3.2 Recommended operating conditions

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

3.3.3 Derating

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

3.3.4 Unused pins

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

Since the details regarding the handling of unused pins differ from device to device and from pin

to pin, please follow the instructions given in the relevant individual datasheets or databook.

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

3.3.5 Latch-up

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

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

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

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

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

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

3.3.6 Input/Output protection

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

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

3.3.7 Load capacitance

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

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

3.3.8 Thermal design

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

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

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

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

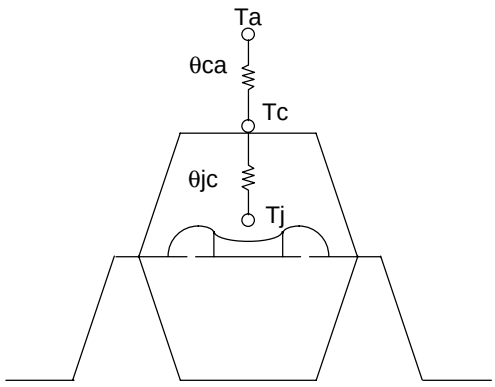


Figure 2 Thermal resistance of package

3.3.9 Interfacing

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

3.3.10 Decoupling

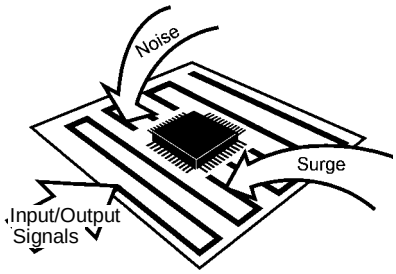
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with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

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

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

3.3.11 External noise

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



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

3.3.12 Electromagnetic interference

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

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

Difficulties in controlling electromagnetic interference derive from the fact that there is no method available which allows designers to calculate, at the design stage, the strength of the electromagnetic waves which will emanate from each component in a piece of equipment. For this reason, it is only after the prototype equipment has been completed that the designer can take measurements using a dedicated instrument to determine the strength of electromagnetic interference waves. Yet it is possible during system design to incorporate some measures for the

prevention of electromagnetic interference, which can facilitate taking corrective measures once the design has been completed. These include installing shields and noise filters, and increasing the thickness of the power supply wiring patterns on the printed circuit board. One effective method, for example, is to devise several shielding options during design, and then select the most suitable shielding method based on the results of measurements taken after the prototype has been completed.

3.3.13 Peripheral circuits

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

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

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

3.3.14 Safety standards

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

3.3.15 Other precautions

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

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

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

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

3.4 Inspection, Testing and Evaluation

3.4.1 Grounding

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

3.4.2 Inspection Sequence

⚠CAUTION

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

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

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

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

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

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

3.5 Mounting

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

3.5.1 Lead forming

⚠CAUTION

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

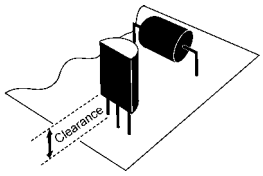
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

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

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

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

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



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

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

3.5.2 Socket mounting

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

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

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

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

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

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

3.5.3 Soldering temperature profile

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

(1) Using a soldering iron

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

(2) Using medium infrared ray reflow

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

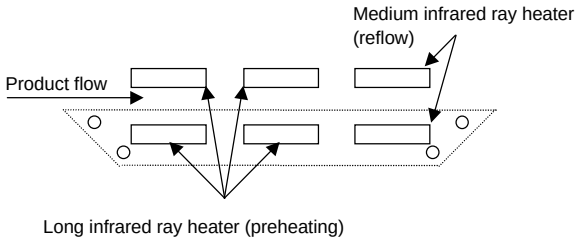


Figure 3 Heating top and bottom with long or medium infrared rays

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

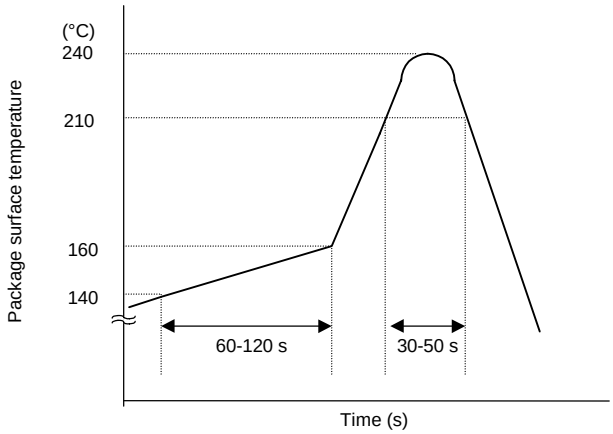


Figure 4 Sample temperature profile for infrared or hot air reflow

(3) Using hot air reflow

- Complete hot air reflow within 30 to 50 seconds at a package surface temperature of between 210°C and 240°C.
- For an example of a recommended temperature profile, refer to Figure 4 above.

(4) Using solder flow

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

less in order to prevent thermal stress in the device.

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

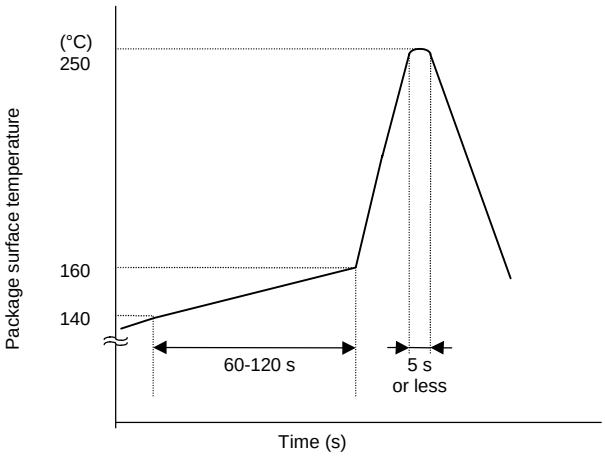


Figure 5 Sample temperature profile for solder flow

3.5.4 Flux cleaning and ultrasonic cleaning

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

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

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

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

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

Frequency: 27 kHz ~ 29 kHz

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

Cleaning time: 30 seconds or less

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

3.5.5 No cleaning

If analog devices or high-speed devices are used without being cleaned, flux residues may cause minute amounts of leakage between pins. Similarly, dew condensation, which occurs in environments containing residual chlorine when power to the device is on, may cause between-lead leakage or migration. Therefore, Toshiba recommends that these devices be cleaned. However, if the flux used contains only a small amount of halogen (0.05W% or less), the devices may be used without cleaning without any problems.

3.5.6 Mounting tape carrier packages (TCPs)

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

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

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

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

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

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

3.5.7 Mounting chips

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

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

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

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

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

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

3.5.8 Circuit board coating

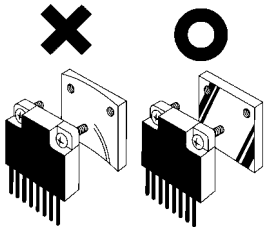
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects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

3.5.9 Heat sinks

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

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

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



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

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

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

3.5.10 Tightening torque

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

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

3.5.11 Repeated device mounting and usage

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

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

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

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

3.6 Protecting Devices in the Field

3.6.1 Temperature

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

3.6.2 Humidity

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

3.6.3 Corrosive gases

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

3.6.4 Radioactive and cosmic rays

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

3.6.5 Strong electrical and magnetic fields

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

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

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

3.6.7 Dust and oil

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

3.6.8 Fire

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

3.7 Disposal of Devices and Packing Materials

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

4. Precautions and Usage Considerations

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

4.1 Microcontrollers

4.1.1 Design

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

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

(2) Undefined functions

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

TMPR4925

Conventions in this Manual

Value Conventions

- Hexadecimal values are expressed as in the following example. (This value is expressed as 42 in the decimal system.)
- KB (kilobyte) = 1,024 Bytes,
  MB (megabyte) = 1,024 × 1,024 = 1,048,576 Bytes,
  GB (gigabyte) = 1,024 × 1,024 × 1,024 = 1,073,741,824 Bytes

Data Conventions

- Byte: 8 bits
- Half-word: 2 consecutive Bytes (16 bits)
- Word: 4 consecutive Bytes (32 bits)
- Double-word: 8 consecutive Bytes (64 bits)

Signal Conventions

- An asterisk ("*") is added to the end of signal names to indicate Low Active signals. (Example: RESET*)
- "Assert" means to move a signal to its Active level. "Deassert" means to move a signal to its Inactive level.

Register Conventions

- Bit operation is expressed as follows.
  Set: Put a bit in the "1" position.
  Clear: Put a bit in the "0" position.
- Properties of each bit in a register are expressed as follows.
  R: Read only. The software cannot change the bit value.
  W: Write only. The value that is read is undefined.
  R/W: Read/Write is possible.
  W1C Write 1 Clear. This corresponding bit is cleared when "1" is written to this bit. "0" is invalid if written.
  R/W1C: Read/Write 1 Clear. These bits can be read from and written to. The corresponding bit is cleared when "1" is written to this bit. "0" is invalid if written.
  R/W0C: Read/Write 0 Clear. These bits can be read from and written to. The corresponding bit is cleared when "0" is written to this bit. "1" is invalid if written.
  R/W1S Read/Write 1 Set. These bits can be read from and written to. The corresponding bit is set when "1" is written to this bit. "0" is invalid if written.
  RS/WC Read Set/Write Clear. These bits can be read from and written to. The bits is set when read, and a write of an arbitrary value to the bit clears it.
  R/L: Property unique to the PCI Controller. This bit can be read. The value of this bit can only be changed by the method described in "10.3.14: Set Configuration Space".
- Registers and the register bit/field name are expressed as "*<register name>.<bit/field name>*".
  Example: CCFG.TOE
  The above example indicates Time Out Bus Error Enable (TOE), a bit field of bit 14 in the Chip Configuration Register (CCFG).

Handling reserved regions

Operation is undefined when a register defined in this document as a reserved region (Reserved) is accessed. If there is a bit or field that was defined as Reserved in a register, write the expressed default value or the specified value ("0" if no particular value is expressed) to these bits. Also, do not use any value read from this bit/field.

Diagnostic function

Any function described as a "diagnostic function" is used to facilitate operation evaluations. The operation of such functions is not guaranteed.

References

64-bit TX System RISC TX49/H2 Core Architecture User's Manual
(http://www.semicon.toshiba.co.jp/eng/index.html)

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

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

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

PCI Local Bus Specification Revision 2.2 (http://www.pcisig.com/)

PCI Bus Power Management Interface Specification Revision 1.1

Audio CODEC '97 (AC '97) Revision 2.1 (http://developer.intel.com/ial/scalableplatforms/audio/)

1. Features

1.1 Outline

The TMPR4925XB, to be referred as TX4925 MIPS RISC micro-controller is a highly integrated ASSP solution based on Toshiba's TX49/H2 processor core, a 64-bit MIPS I, II, III ISA Instruction Set Architecture (ISA) compatible with additional instructions. For information on the architecture of the TX49/H2 core, including the instruction set, refer to the manual 64-bit TX System RISC, TX49/H2 Core Architecture.

The TX4925 is a highly integrated device with integrated peripherals such as SDRAM memory controller, NAND Flash memory controller, PCI controller, AC-Link controller, PIO, SIO, SPI, CHI, PCMCIA I/F and Timer.

This class of product is targeted for applications that require a high performance and cost-effective solution such as networking internet appliance and information terminal.

1.2 Features

- TX49/H2 core with an integrated IEEE 754-compliant FPU for single- and double-precision operations
- 4-channel SDRAM Controller (32 bit/80 MHz) and support SyncFlash® memory
- 6-channel External Bus Controller (including 2-slot PCMCIA Interface)
- NAND Flash Controller
- 32-bit PCI Controller (33 MHz)
- 4-channel Direct Memory Access (DMA) Controller
- 2-channel Serial I/O Port
- Parallel I/O Port (up to 32-bit)
- SPI (Serial Peripheral Interface)
- CHI (high-speed serial Concentration Highway Interface)
- Interrupt Controller
- 3-channel Timer/Counter and 44-bit up-counter RTC
- AC-Link Controller
- PCMCIA Interface (2-slot)
- Supports selection between little endian and big endian modes
- Low power dissipation

  The TX4925 operates with the 1.5 V core and the 3.3 V I/O, while supporting a low-power (Halt) mode.
- CPU maximum operating frequency: 200 MHz
- IEEE1149.1 (JTAG) support: Debug Support Unit (Enhanced JTAG)
- 256-pin PBGA package

1.2.1 TX49/H2 Processor Core Features

The TX49/H2 is a high-performance 64-bit microprocessor core developed by Toshiba.

- 64-bit operation
- 32-, 64-bit integer general purpose registers
- 32-bit physical address space and 64-bit virtual address space
- Optimized 5-stage pipeline
- Instruction Set
  - MIPS I, II , III compatible ISA
  - PREF (Prefetch) and MAC (Multiply/Accumulate) instructions.
- 16k Byte Instruction Cache, and 16k Byte Data Cache
  - 4-way set associative with lock function
- MMU (Memory Management Unit): 48-entry fully associative JTLB
- The on-chip FPU supports both single- and double-precision arithmetic, as specified in IEEE Std 754.
- On-chip 4-deep write buffer
- Enhanced JTAG debug feature
  - Built-in Debug Support Unit (DSU)

1.2.2 TX4925 Peripheral Circuit Features

1.2.2.1 External Bus Controller (EBUSC)

The External Bus Controller generates necessary signals to control external memory and I/O devices.

- 6 channels of chip select signals, enabling control of up to six devices
- Supports access to ROM (including mask ROM, page mode ROM, EPROM and EEPROM), SRAM, flash ROM, and I/O devices
- Supports 32-bit, 16-bit and 8-bit data bus sizing on a per channel basis
- Supports selection among full speed (up to 80 MHz), 1/2 speed (up to 40 MHz), 1/3 speed (up to 27 MHz) and 1/4 speed (up to 20 MHz) on a per channel basis
- Support specification of timing on a per channel basis
- The user can specify setup and hold times for address, chip enable, write enable, and output enable signals
- Supports memory sizes of 1M byte to 1G byte for devices with 32-bit data bus, 1M byte to 512M bytes for devices with 16-bit data bus, and 1M byte to 256M bytes for devices with 8-bit data bus

1.2.2.2 DMA Controller (DMAC)

The TX4925 contains a 4-channel DMA controller that executes DMA transfer to memory and I/O devices.

- 4-channel independently handling internal/external DMA requests

  (Usable 2 channels by external DMA requests)
- Supports DMA transfer with built-in serial I/O controller and AC-link controller based on internal DMA requests
- Supports signal address (fly-by DMA) and dual address transfers in external I/O DMA transfer mode using external DMA requests
- Supports transfer between memory and external I/O devices having 32-/16-/8-bit data bus
- Supports memory-to-memory copy mode, with no address boundary restrictions
- Supports burst transfer of up to 8 words for a single read/write
- Supports memory fill mode, writing word data to specified memory area
- Supports chained DMA transfer

1.2.2.3 SDRAM Controller (SDRAMC)

The SDRAM Controller generates necessary control signals for the SDRAM interface. It has four channels and can handle up to 2G bytes (512 MB/channel) of memory by supporting a variety of memory configurations.

- Memory clock frequency: 80 MHz (divided by 2.5)
- 4 sets of independent memory channels
- Supports 16M-/64M-/128M-/256M-/512M-bit SDRAM with 2/4 bank size availability
- Supports Single Data Rate (SDR) SDRAM and SyncFlash® memory
- Supports use of Registered DIMM
- Supports 32-/16-bit data bus sizing on a per channel basis
- Supports specification of SDRAM timing on a per channel basis
- Supports critical word first access of TX49/H2 core
- Low power mode: selectable between self-refreshing and pre-charge power-down

1.2.2.4 PCI Controller (PCIC)

The TX4925 contains a PCI Controller that complies with PCI Local Bus Specification Revision 2.2.

- Compliance with PCI Local Bus Specification Revision 2.2
  (Partly supports power management as optional function)
- 32-bit PCI interface featuring maximum PCI bus clock frequency of 33 MHz
- Supports both target and initiator functions
- Supports change of address mapping between internal bus and PCI bus
- PCI bus arbiter enables connection of up 4 external bus masters
- Supports booting of TX4925 from memory on PCI bus
- 1 channel of DMA controller dedicated to PCI controller (PDMAC)

1.2.2.5 Serial I/O Controller (SIO)

The TX4925 contains a 2-channels asynchronous serial I/O interface (full duplex UART).

- 2-channel full duplex UART
- Built-in baud rate generator
- FIFOs
  8-bit x 8 transmitter FIFO
  13-bit (8 data bits and 5 status bits) x 16 receiver FIFO
- Supports DMA tranfer

1.2.2.6 Timers/Counters Controller (TMR)

The TX4925 contains 3-channel timer/counters.

- 3-channel 32-bit up-counter
- Supports three modes : interval timer mode, pulse generator mode, and watchdog timer mode
- 2 timer output pins
- 1 count clock input pin

1.2.2.7 Parallel I/O Ports (PIO)

The TX4925 contains 32-bit parallel I/O ports

- Independent selection of direction of pins and output port type (totem-pole or open-drain outputs) on a per bit basis.

1.2.2.8 AC-Link Controller (ACLC)

AC-Link Controller in TX4925 can be connected to audio and/or modem CODECs described in the “Audio CODEC ’97 Revision 2.1” (AC’97) and can operate them.

- AC97 2.1 compliant CODEC register access protocol
- Up to two CODECs are supported
- Support Audio CODEC (Recording/Playback of 16-bit PCM Left/Right channels)
  - Support Playback of 16-bit Surround, Center, and LFE channels
  - Support Variable Rate Audio recording/Playback
- Support Modem CODEC (Line 1 and GPIO slots)
- Support AC-link low-power mode, wake-up, and warm-reset
- Support sample-data I/O via DMA transfer

1.2.2.9 Interrupt Controller (IRC)

Interrupt controller in TX4925 supports both internal and external interrupts to the processor core. The priority or the value of each interrupt source is programmed in interrupt level registers.

It has a 16-bit flag register to generate interrupt requests to external devices or the TX49/H2 core.

- Priority process of 21 internal and up to 8 external interrupt sources.
  - PCIC, DMAC, SIO, Timer Interrupts, RTC, NAND Flash Controller, ACLC, SPI, CHI, PIO and External Interrupts
  - All of external interrupt signals shares other function signals
- Edge/Level selectable trigger interrupt
- Support of Non Mask-able interrupt (NMI)
- 16-bit read/write flag register for interrupt requests, making it possible to issue interrupt requests to external devices and to the TX49/H2 core (IRC interrupts)

1.2.2.10 high-speed serial Concentration Highway Interface (CHI)

The TX4925 has a CHI module.

- Contents logic for interfacing to external full-duplex serial time-division-multiplexed (TDM) communication peripherals
- Supports ISDN line interface chips and other PCM/TDM serial devices
- Programmable CHI Interface (numbers of channels, frame rate, bit rate, etc.)
- supports data rates up to 4.096 Mbps

1.2.2.11 Serial Peripheral Interface (SPI)

The TX4925 has an SPI module.

- full-duplex, synchronous serial data transfers (data in/out, and clock signals)
- 8-bit or 16-bit data word lengths
- Programmable SPI baud rate

1.2.2.12 NAND Flash Controller

The TX4925 has a NAND Flash memory Controller.

- Controlled NAND Flash I/F by Setting Register
- Supports On-chip ECC (Error Correct Circuit) calculating circuits

1.2.2.13 PCMCIA Interface (PCMCIAI/F)

The TX4925 has a 2 identical full PCMCIA ports.

- Provide the control signals and accepts the status signals which conform to the PCMCIA version 2.1 standard
- Appropriate connector keying and level-shifting buffers required for 3.3 V versus 5 V PCMCIA interface implementations

1.2.2.14 Real Time Clock (RTC)

The TX4925 has an RTC module.

- 44-bit up-counter
- Interrupts on alarm, timer, and prior to RTC roll-over
- Date managed by software

1.2.2.15 Power-down Mode

The TX4925 contains support for implementation of power-down mode.

- HALT mode (stopping CPU core clock) for TX49/H2 core block
- Power-down mode (stopping input clock) for individual internal peripheral modules
- RF (Reduced Frequency) Function (1/1, 1/2, 1/4, 1/8)

1.2.2.16 EJTAG Interface

The TX4925 contains an Extended Enhanced Joint Test Action Group (Extended EJTAG) interface, which provides two functions: JTAG boundary scan test that complies with IEEE1149.1 and real-time debugging using a debug support unit (DSU) built into the TX49/H2 core.

- IEEE 1149.1 JTAG Boundary Scan
- Real-time debugging functions using special emulation probe: execution control (execution, break, step, and register/memory access) and PC trace

2. Block Diagram

2.1 TX4925 Block Diagram

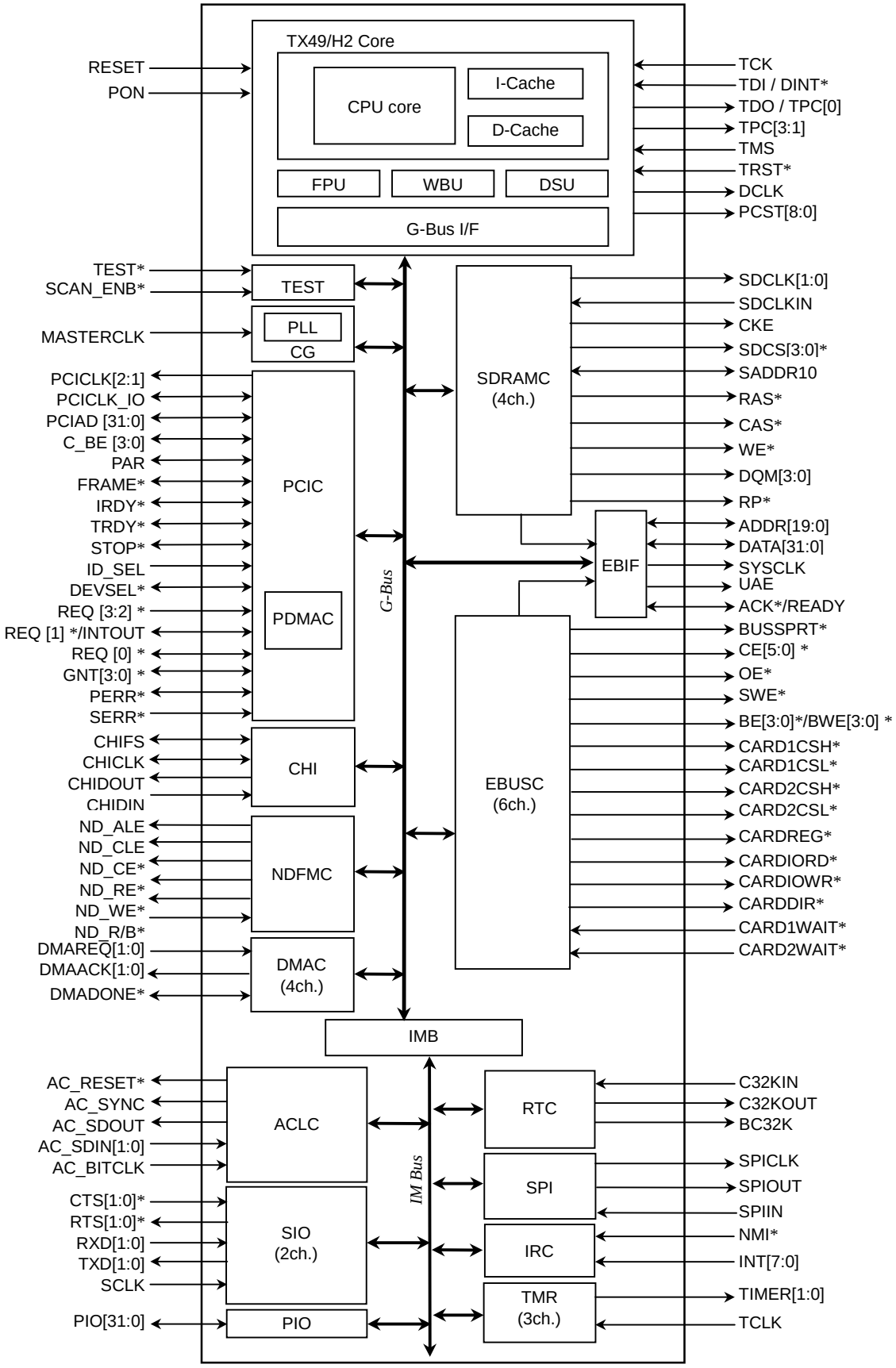


Figure 2.1.1 TX4925 Block Diagram

Figure 2.1.1 shows a diagram of the TX4925. The each block is itemized below.

(1) TX49/H2 core

- It is composed of CPU, System Control Coprocessor (CP0), Instruction Cache, Data Cache, Floating-Point Unit (FPU), Write Buffer (WBU), Debug Support Unit (DSU) and G-Bus I/F.
  - FPU: IEEE754 compatible single and double precision FPU
    It is assigned as one of the coprocessor unit, CP1.
  - I-Cache: Instruction Cache Memory, 16Kbyte, 4-Way set associative
  - D-Cache: Data Cache Memory, 16Kbyte, 4-Way set associative Select write-back mode or write-through(no write allocate/write allocate) mode for cache write policy.
  - DSU: Debug Support Unit, It is used for on chip debugging module.

(2) EBUSC

- External Bus Controller, Provides 6-channels programmable Chip Selects, Supports ROMs (page-mode ROM, mask ROM, EPROM and EEPROM), SRAMs, flash ROMs and I/O devices

(3) DMAC

- Direct Memory Access Controller, Supports transfers to and from both memory and internal/external I/O devices

(4) SDRAMC

- SDRAM Controller, Supports 4-channels/80 MHz bus frequencies/16 or 32 bus width

(5) PCIC

- PCI-bus Controller, Compliance with PCI Local Bus Specification Revision 2.2, Supports 32-bit PCI bus interface and 33 MHz operation

(6) SIO

- Serial I/O, Supports 2-channels for universal asynchronous receiver/transmitters

(7) TMR

- Timer/Counter, Supports 3-channels

(8) PIO

- Parallel I/O, Supports 32-bit of shared PIO pins

(9) ACLC

- AC-Link Controller, Compliance with Audio CODEC ’97 Revision 2.1 (AC ’97)

(10) IRC

- Interrupt Controller

(11) CHI

- High–speed serial Concentration Highway Interface

(12) SPI

- Serial Peripheral Interface, Supports full-duplex synchronous serial data transfers

(13) NDFMC

- NAND Flash memory Controller, Supports ECC (Error Correction Code) function

(14) RTC

- Real Time Clock, 44-bit up-counter

(15) EBIF

- External Bus Interface, Connects between 20-bit external address bus/32-bit external data bus and SDRAMC/EBUSC

(16) CG

- Clock Generator, Incorporates an phase-locked loop (PLL) circuit to drive the multiplied clocks, Generates the clocks for each module

(17) G-Bus

- High-speed bus which is 32-bit bus width within TX4925, Direct connect to TX49/H2 core block

(18) IM-Bus

- Low-speed bus which is 32-bit bus width within TX4925, Connect to G-Bus via IMB

(19) IMB

- G-Bus and IM-Bus bridge

(20) TEST

- Internal diagnostic module

3. Signals

3.1 Pin Signal Description

In the following tables, asterisks at the end of signal names indicate active-low signals.

In the Type column, PU indicates that the pin is equipped with an internal pull-up resister and PD indicates that the pin is equipped with an internal pull-down resister. OD indicates an open-drain pin.

The Initial State column shows the state of the signal when the RESET* signal is asserted and immediately after it is deasserted. Those signals which are selected by a configuration signal upon a reset have the state selected by the configuration signal even when the reset signal is asserted.

3.1.1 Signals Common to SDRAM and External Bus Interfaces

Table 3.1.1 Signals Common to SDRAM and External Bus Interfaces

Signal Name	Type	Description	Initial State
ADDR[19:0]	Input/output PU	Address Address signals. For SDRAM, ADDR[19:16 , 14:5] and SADDR10 are used (refer to Sections “9.3.2.2” and “9.3.2.3 Address Signal Mapping”). When the external bus controller uses these pins, the meaning of each bit varies with the data bus width (refer to Section “7.3.5 Data Bus Size”). The ADDR signals are also used as boot configuration signals (input) during a reset. For details of configuration signals, refer to Section “3.2 Boot Configuration”. The ADDR signals are input signals only when the RESET* signal is asserted and become output signals after the RESET* signal is deasserted.	Input
SADDR10	Input/output PU	Address10 for SDRAM Address signal for SDRAM (refer to Sections “9.3.2.2” and “9.3.2.3 Address Signal Mapping”). This signal is also used as a boot configuration input signal for testing. Because this signal is used for testing, ensure that it will not pulled Low during a reset sequence. For details of configuration signals, refer to Section “3.2 Boot Configuration”. This signal is used as an input signal while the RESET* signal is asserted. It becomes an output signal once the RESET* signal has been deasserted.	Input
DATA[31:0]	Input/output PU	Data 32-bit data bus.	Input
BUSSPRT	Output	Bus Separate Controls the connection and separation of devices controlled by the external bus controller to or from a high-speed device, such as SDRAM (refer to Section “7.6 Flash ROM, SRAM Usage Example”). H: Separate devices other than SDRAM from the data bus. L: Connect devices other than SDRAM to the data bus. Separation and connection are performed using external bidirectional bus buffers (such as the 74xx245).	High

3.1.2 SDRAM Interface Signals

Table 3.1.2 SDRAM Interface Signals

Signal Name	Type	Description	Initial State
SDCLK[1:0]	Output	SDRAM Controller Clock Clock signals used by SDRAM/SyncFlash. The clock frequency is the same as the G-Bus clock (GBUSCLK) frequency. When these clock signals are not used, the pins can be set to L using the SDCLK Enable field of the pin configuration register (PCFG.SDCLKEN[1:0]).	All High
SDCLKIN	Input/output	SDRAM Feedback Clock input Feedback clock signal for SDRAM controller input signals.	Input
CKE	Output	Clock Enable CKE signal for SDRAM/SyncFlash.	High
SDCS[3:0]*	Output	Synchronous Memory Device Chip Select Chip select signals for SDRAM/SyncFlash.	All High
RAS*	Output	Row Address Strobe RAS signal for SDRAM/SyncFlash.	High
CAS*	Output	Column Address Strobe CAS signal for SDRAM/SyncFlash.	High
WE*	Output	Write Enable WR signal for SDRAM/SyncFlash.	High
DQM[3:0]	Output	Data Mask During a write cycle, the DQM signals function as a data mask. During a read cycle, they control the SDRAM output buffers. The bits correspond to the following data bus signals: DQM[3]:DATA[31:24], DQM[2]:DATA[23:16] DQM[1]:DATA[15:8], DQM[0]:DATA[7:0]	All High
RP*	Output	Initialize/Power Down RP* signal for SyncFlash.	Low

3.1.3 External Interface Signals

Table 3.1.3 External Interface Signals (1/2)

Signal Name	Type	Description	Initial State
SYSCLK	Output	System Clock Clock for external I/O devices. Outputs a clock in full speed mode (at the same frequency as the G-Bus clock (GBUSCLK) frequency), half speed mode (at one half the GBUSCLK frequency), third speed mode (at one third the GBUSCLK frequency), or quarter speed mode (at one quarter the GBUSCLK frequency). The boot configuration signals on the ADDR[4:3] pins select which speed mode will be used. When this clock signal is not used, the pin can be set to L using the SYSCLK Enable bit of the configuration register (PCFG.SYSCLKEN).	High
UAE	Output PU	Upper Address Enable Latch enable signal for the high-order address bits of ADDR. This is a Low active signal. This signal is also used as a boot configuration input signal for testing. Because this signal is used for testing, ensure that it will not pulled Low during a reset sequence. For details of configuration signals, refer to Section “3.2 Boot Configuration”. This signal is used as an input signal while the RESET* signal is asserted. It becomes an output signal once the RESET* signal has been deasserted.	Input
CE[5:4]*	Output PU	Chip Enable Chip select signals for ROM, SRAM, and I/O devices. The pins are shared with other functions (refer to Section “3.3 Pin Multiplexing”).	All High
CE[3:0]*	Output	Chip Enable Chip select signals for ROM, SRAM, and I/O devices.	All High
OE*	Output	Output Enable Output enable signal for ROM, SRAM, and I/O devices.	High
SWE*	Output	Write Enable Write enable signal for SRAM and I/O devices.	High
BWE[3:0]* /BE[3:0]*	Output	Byte Enable/Byte Write Enable BE[3:0]* indicate a valid data position on the data bus DATA[31:0] during read and write bus operation. In 16-bit bus mode, only BE[1:0]* are used. In 8-bit bus mode, only BE[0]* is used. BWE[3:0]* indicate a valid data position on the data bus DATA[31:0] during write bus operation. In 16-bit bus mode, only BWE[1:0]* are used. In 8-bit bus mode, only BWE[0]* is used. The following shows the correspondence between BE[3:0]*/BWE[3:0]* and the data bus signals. BE[3]*/BWE[3]*: DATA[31:24] BE[2]*/BWE[2]*: DATA[23:16] BE[1]*/BWE[1]*: DATA[15:8] BE[0]*/BWE[0]*: DATA[7:0] The boot configuration signal on the ADDR[11] pin and the EBCCRn.BC bit of the external bus controller determine whether the signals are used as BE[3:0]* or BWE[3:0]*.	All High
ACK*/ READY	Input/output PU	Data Acknowledge/Ready Flow control signal (refer to Section “7.3.6 Access Modes”).	Input

Table 3.1.3 External Interface Signals (2/2)

Signal Name	Type	Description	Initial State
CARD1CSH* CARD1CSL*	Output PU	PCMCIA card slot 1 chip select Chip select signals for PCMCIA card slot 1. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARD2CSH* CARD2CSL*	Output PU	PCMCIA card slot 2 chip select Chip select signals for PCMCIA card slot 2. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARDREG*	Output PU	PCMCIA card register REG* signal for a PCMCIA card. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARDIORD*	Output PU	PCMCIA card I/O read IORD* signal for a PCMCIA card. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARDIOWR*	Output PU	PCMCIA card I/O write IOWR* signal for a PCMCIA card. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARDDIR*	Output PU	PCMCIA card directory Controls the direction of the bidirectional buffer used for a PCMCIA slot. This signal is asserted during a read transaction when any of CARD2CSH*, CARD2CSL*, CARD1CSH* and CARD1CSL* are asserted. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARD1WAIT*	Input PU	PCMCIA card slot 1 wait Card wait signal from PCMCIA card slot 1. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CARD2WAIT*	Input PU	PCMCIA card slot 2 wait Card wait signal from PCMCIA card slot 2. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.4 DMA Interface Signals

Table 3.1.4 DMA Interface Signals

Signal Name	Type	Description	Initial State
DMAREQ[1:0]	Input PU	DMA Request DMA transfer request signals from an external I/O device. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
DMAACK[1:0]	Output	DMA Acknowledge DMA transfer acknowledge signals to an external I/O device. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
DMADONE*	Input/output PU	DMA Done DMADONE* is either used as an output signal that reports the termination of DMA transfer or as an input signal that causes DMA transfer to terminate. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.5 PCI Interface Signals

Table 3.1.5 PCI Interface Signals (1/2)

Signal Name	Type	Description	Initial State
PCICLK[2:1]	Output	PCI Clock PCI bus clock signals. A boot configuration signal (ADDR[18]) can determine whether the clock internally generated in the TX4925 is used as PCICLK. If the TX4925 internal clock is selected, the clock signals are output from these pins. When these clock signals are not used, the pins can be set to High-Z using the PCICLK Enable field of the pin configuration register (PCFG.PCICLKEN[2:1]).	Selected by ADDR[18] H: High L: L
PCICLKIO	Input/output	PCI Feedback Clock PCI feedback clock input. A boot configuration signal (ADDR[18]) can determine whether the clock internally generated in the TX4925 is used as PCICLK. If the TX4925 internal clock is selected, the clock signals are output and simultaneously fed back to the internal PCI block. When using the PCI block, therefore, do not set the PCICLK Enable field of the pin configuration register (PCFG.PCICLKIOEN) to 0.	Selected by ADDR[18] H: High L: Input
PCIAD[31:0]	Input/output	PCI Address and Data Multiplexed address and data bus.	Input
C_BE[3:0]	Input/output	Command and Byte Enable Command and byte enable signals.	Input
PAR	Input/output	Parity Even parity signal for PCIAD[31:0] and C_BE[3:0]*.	Input
FRAME*	Input/output	Cycle Frame Indicates that bus operation is in progress.	Input
IRDY*	Input/output	Initiator Ready Indicates that the initiator is ready to complete data transfer.	Input
TRDY*	Input/output	Target Ready Indicates that the target is ready to complete data transfer.	Input
STOP*	Input/output	Stop The target sends this signal to the initiator to request termination of data transfer.	Input
ID_SEL	Input	Initialization Device Select Chip select signal used for configuration access.	Input
DEVSEL*	Input/output	Device Select The target asserts this signal in response to access from the initiator.	Input

Table 3.1.5 PCI Interface Signals (2/2)

Signal Name	Type	Description	Initial State
REQ[3:2]*	Input	Request Signals used by the master to request bus mastership. The boot configuration signal on the ADDR[1] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, REQ[3:2]* are PCI bus request input signals. In external arbiter mode, REQ[3:2]* are not used. Because the pins are still placed in the input state, they must be pulled up externally.	Input
REQ[1]* /INTOUT	Input/output/ OD	Request Signal used by the master to request bus mastership. The boot configuration signal on the ADDR[1] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, this signal is a PCI bus request input signal. In external arbiter mode, this signal is an external interrupt output signal (INTOUT). Refer to Section “15.3.7 Interrupt Requests”.	Selected by ADDR[1] H: Input L: High-Z
REQ[0]*	Input/output	Request Signal used by the master to request bus mastership. The boot configuration signal on the ADDR[1] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, this signal is a PCI bus request input signal. In external arbiter mode, this signal is a PCI bus request output signal.	Selected by ADDR[1] H: Input L: High
GNT[3:0]*	Input/output	Grant Indicates that bus mastership has been granted to the PCI bus master. The boot configuration signal on the ADDR[1] pin determines whether the built-in PCI bus arbiter is used. In internal arbiter mode, all of GNT[3:0]* are PCI bus grant output signals. In external arbiter mode, GNT[0]* is a PCI bus grant input signal. Because GNT[3:1]* also become input signals, they must be pulled up externally.	Selected by ADDR[1] H: All High L: Input
PERR*	Input/output	Data Parity Error Indicates a data parity error in a bus cycle other than special cycles.	Input
SERR*	Input/OD	System Error Indicates an address parity error, a data parity error in a special cycle, or a fatal error. In host mode, SERR* is an input signal. In satellite mode, SERR* is an open-drain output signal. The mode is determined by the boot configuration signal on the ADDR[19] pin.	Input

3.1.6 Serial I/O Interface Signals

Table 3.1.6 Serial I/O Interface Signals

Signal Name	Type	Description	Initial State
CTS [1:0]*	Input PU *1	SIO Clear to Send CTS* signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
RTS [1:0]*	Output PU *1	SIO Request to Send RTS* signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
RXD[1:0]	Input PU *1	SIO Receive Data Serial data input signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
TXD[1:0]	3-state Output PU *1	SIO Transmit Data Serial data output signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SCLK	Input PU	External Serial Clock SIO clock input signal. SIO0 and SIO1 share this signal. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

*1: These signals are pulled up for channel 0 only. No pull-up resistor is provided for channel 1.

3.1.7 Timer Interface Signals

Table 3.1.7 Timer Interface Signals

Signal Name	Type	Description	Initial State
TIMER[1:0]	Output PU	Timer Output Timer output signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
TCLK	Input PU	External Timer Clock Timer input clock signal. TMR0, TMR1, and TMR2 share this signal. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.8 Parallel I/O Interface Signals

Table 3.1.8 Parallel I/O Interface Signals

Signal Name	Type	Description	Initial State
PIO[31:20]	Input/output PU	PIO Ports[31:20] Parallel I/O signals. The pins are shared with other functions, including PC trace (refer to Section "3.3 Pin Multiplexing"). The boot configuration signal on the TDO pin determines whether the signals are used for PC trace.	Selected by TDO H: PIO input L: Output (PC trace function)
PIO[19:0]	Input/output PU *1	PIO Ports[19:0] Parallel I/O signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	Input

*1: PIO[17:12] do not have pull-up resistors.

3.1.9 AC-link Interface Signals

Table 3.1.9 AC-link Interface Signals

Signal Name	Type	Description	Initial State
ACRESET*	Output	AC '97 Master H/W Reset The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SYNC	Output	48 kHz Fixed Rate Sample Sync The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SDOUT	Output	Serial, Time Division Multiplexed, AC '97 Output Stream The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SDIN]1]	Input	Serial, Time Division Multiplexed, AC '97 Input Stream The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SDIN[0]	Input	Serial, Time Division Multiplexed, AC '97 Input Stream The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
BITCLK	Input	12.288 MHz Serial Data Clock The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.10 Interrupt Signals

Table 3.1.10 Interrupt Signals

Signal Name	Type	Description	Initial State
NMI*	Input PU	Non-Maskable Interrupt Non-maskable interrupt signal.	Input
INT[7:0]*	Input PU	External Interrupt Requests External interrupt request signals. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.11 CHI Interface Signals

Table 3.1.11 CHI Interface Signals

Signal Name	Type	Description	Initial State
CHIFS	Input/output PU	CHI Frame synchronization CHI frame synchronization signal. This pin can be used in either output or input mode. In output mode, the pin allows the TX4925 to become the master CHI synchronization source. In input mode, the pin allows the external peripheral device to become the master CHI synchronization source. In that case, the TX4925 CHI module becomes a slave for external synchronization. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CHICLK	Input/output PU	CHI Clock CHI clock signal. This pin can be used in either output or input mode. In output mode, the pin allows the TX4925 to become the master CHI clock source. In input mode, the pin allows the external peripheral device to become the master CHI clock source. In that case, the TX4925 CHI module becomes a slave for the external clock. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CHIDOUT	Output PU	CHI Data Output CHI serial data output signal. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
CHIDIN	Input PU	CHI Data Input CHI serial data input signal. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.12 SPI Interface Signals

Table 3.1.12 SPI Interface Signals

Signal Name	Type	Description	Initial State
SPICLK	Output PU	SPI Clock This pin is used for a data clock to or from an SPI slave device. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SPIOUT	Output PU	SPI Data Output This signal contains data to be shifted to an SPI slave device. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
SPIIN	Input PU	SPI Data Input This signal contains data to be shifted from an SPI slave device. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.13 NAND Flash Memory Interface Signals

Table 3.1.13 NAND Flash Memory Interface Signals

Signal Name	Type	Description	Initial State
ND_ALE	Output	NAND Flash Address Latch Enable ALE signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
ND_CLE	Output	NAND Flash Command Latch Enable CLE signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
ND_CE*	Output	NAND Flash Chip Enable CE signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
ND_RE*	Output	NAND Flash Read Enable RE signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
ND_WE*	Output	NAND Flash Write Enable WE signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input
ND_R/B*	Input	NAND Flash Ready/Busy Ready/Busy signal for NAND flash memory. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	PIO input

3.1.14 Extended EJTAG Interface Signals

Table 3.1.14 Extended EJTAG Interface Signals (1/2)

Signal Name	Type	Description	Initial State
TCK	Input PU	JTAG Test Clock Input Clock input signal for JTAG. TCK is used to execute JTAG instructions and input/output data.	Input
TDI/DINT*	Input PU	JTAG Test Data Input/Debug Interrupt When PC trace mode is not selected, this signal is a JTAG data input signal. It is used to input serial data to JTAG data/instruction registers. When PC trace mode is selected, this signal is an interrupt input signal used to cancel PC trace mode for the debug unit.	Input
TDO/TPC[0]	Output	JTAG Test Data Output/PC Trace Output When PC trace mode is not selected, this signal is a JTAG data output signal. Data is output by means of serial scan. When PC trace mode is selected, this signal outputs the value of the noncontiguous program counter in sync with the debug clock (DCLK).	Input

Table 3.1.14 Extended EJTAG Interface Signals (2/2)

Signal Name	Type	Description	Initial State
TPC[3:1]	Output	PC Trace Output TPC[3:1] output the value of the noncontiguous program counter in sync with DCLK. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	Selected by TDO H: PIO input L: All High
TMS	Input PU	JTAG Test Mode Select Input TMS mainly controls state transition in the TAP controller state machine.	Input
TRST*	Input	Test Reset Input Asynchronous reset input for the TAP controller and debug support unit (DSU). When an EJTAG probe is not connected, this pin must be fixed to low. When connecting an EJTAG probe, prevent floating, for example, by connecting a pull-up resistor. When this signal is deasserted, G-Bus timeout detection is disabled (refer to Section “5.1.1 Detecting G-Bus Timeout”).	Input
DCLK	Output	Debug Clock Clock output signal for the real-time debugging system. When PC trace mode is selected, the TPC[3:1] and PCST signals are output synchronously. This clock is the TX49/H2 core operating clock (CPUCLK) divided by 3. The pin is shared with other functions (refer to Section "3.3 Pin Multiplexing").	Selected by TDO H: PIO input L: Low
PCST[8:0]	Output	PC Trace Status Information Outputs PC trace status and other information. The pins are shared with other functions (refer to Section "3.3 Pin Multiplexing").	Selected by TDO H: PIO input (PCST[8:1]) BC32K(PCST[0]) L: All Low

3.1.15 Clock Signals

Table 3.1.15 Clock Signals

Signal Name	Type	Description	Initial State
MASTERCLK	Input	Master Clock Input pin for the TX4925 operating clock. A crystal resonator cannot be connected to this pin because the pin does not contain an oscillator.	Input
C32KIN	Input	32 KHz Crystal Input Connect this pin and C32KOUT to a 32.768 kHz crystal.	Input
C32KOUT	Output	32 KHz Crystal output Connect this pin and C32KIN to a 32.768 kHz crystal.	Output
BC32K	Output PU	Buffer output of 32 KHz Crystal Buffer output for a 32.768 kHz clock.	Selected by TDO H: Output (BC32K) L: Low

3.1.16 Initialization Signals

Table 3.1.16 Initialization Signals

Signal Name	Type	Description	Initial State
RESET*	Input SMT	Reset Reset signal.	Input
PON*	Input SMT	Power On Reset Initializes the CG. For timing, refer to Section "6.3 Power-On Sequence".	Input

3.1.17 Test Signals

Table 3.1.17 Test Signals

Signal Name	Type	Description	Initial State
TEST*	Input	Test Mode Setting Test pin. This pin must be fixed to High.	Input
SCANENB*	Input	Scan Mode Test Control Test pin. This pin must be fixed to High.	Input

3.1.18 Power Supply Pins

Table 3.1.18 Power Supply Pins

Signal Name	Type	Description	Initial State
PLL1VDD_A	—	PLL Power Pins PLL analog power supply pins. PLL1VDD_A = 1.5 V	—
PLL1VSS_A	—	PLL Ground Pins PLL analog ground pins. PLL1VSS_A = 0 V	—
VccInt (VDDC)	—	Internal Power Pins Digital power supply pins for internal logic. VccInt = 1.5 V.	—
VccIO (VDDS)	—	I/O Power Pins Digital power supply pins for input/output pins. VccIO = 3.3 V.	—
Vss	—	Ground Pins Digital ground pins. Vss = 0 V.	—

3.2 Boot Configuration

The ADDR[19:0], UAE, SADDR10 and TDO signals can also function as configuration signals for initially setting various functions upon booting the system. The states of the configuration signals immediately after the RESET* or PON* signal is deasserted are read as initial values for the TX4925 internal registers. A High signal level sets a value of 1 and a Low signal level sets a value of 0. UAE, SADDR10 and ADDR[17,16,10] are used for testing. Ensure that these pins will not set to 0.

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

Table 3.2.1 lists the functions that can be set using configuration signals. Table 3.2.2 and 3.2.3 describe each configuration signal. Note that the functions of some pins vary with the boot memory device used.

Table 3.2.1 Functions that Can be Set Using Configuration Signals

Peripheral Function	Functions that Can be Set	Configuration Signal
PCI controller	PCICLK enable	ADDR[18]
	PCI controller operating mode (satellite or host)	ADDR[15]
	PCI bus arbiter selection (internal or external)	ADDR[1]
	PCI bus master*1	ADDR[11]
External bus controller	Division ratio of SYSCLK to GBUSCLK	ADDR[4:3]
	Boot device selection	ADDR[8:6]
	BE[3:0]*/BWE[3:0]* function selection upon booting	ADDR[11]
	Handling of the ACK signal upon booting (internal or external)	ADDR[5]
	Data bus width for the boot device	ADDR[13:12]
SDRAM controller	Data bus width for the boot device*2	ADDR[13]
	Row address size for boot memory SyncFlash*2	BADDR[5,2]
	Column address size for boot memory SyncFlash*2	ADDR[11,12]
Others	Endian setting	ADDR[14]
	Controlling built-in timer interrupts of the TX49/H2 core	ADDR[0]
	PC trace	TDO

*1: Valid when the boot memory device is PCI and the PCIC operates in satellite mode.

*2: Valid when the boot memory device is SyncFlash.

Table 3.2.2 Boot Configuration Specified with the ADDR[19:0], TDO, UAE and SADDR10 Signals (1/2)

Signal	Description	Corresponding Register Bit	Configuration Determined at
ADDR[19]	Reserved This signal will be set to 0 upon booting.	CCFG bit [12]	RESET* deassert edge
ADDR[18]	PCI Clock Enable Specifies whether the clock generated by the TX4925 internal clock generator is used as a PCI clock. L = Use a PCI clock input from an external device. The PCI clock is input via PCICLKIO. PCICLK[2:1] are placed in High-Z state. H = Use a clock generated by the TX4925 clock generator. The clock is output from PCICLK[2:1] and PCICLKIO. The clock output from PCICLKIO is fed back for use with the internal PCIC.	PCFG. PCICLKEN	PON* deassert edge
ADDR[17:16]	Reserved Used for testing. Because this signal is used for setting a clock frequency, ensure that the signal will not be set to 0 upon booting.	CCFG.bit[18](ADDR[17]) CCFG.bit[27](ADDR[16])	PON* deassert edge
ADDR[15]	PCI Controller Mode Select Specifies the operating mode of the TX4925 PCI controller. L = Satellite H = Host	CCFG. PCIMODE	RESET* deassert edge
ADDR[14]	TX4925 Endian Mode Specifies the TX4925 endian mode. L = Little endian H = Big endian	CCFG. ENDIAN	RESET* deassert edge
ADDR[13:12]	Boot ROM Bus Width Specifies the data bus width when booting from a memory device connected to the external memory controller. If SyncFlash is selected as a boot memory device, refer to Table 3.2.3. LL = Reserved LH = 32 bits HL = 16 bits HH = 8 bits	EBCCR0.BSZ	RESET* deassert edge
ADDR[11]	Boot Byte Enable Type When booting from a memory device connected to the external memory controller, specifies the function of the BE[3:0]*/BWE[3:0]* pins upon booting. Be sure that this bit does not become “0” when booting from PCL. L = BE[3:0]* (Byte Enable) H = BWE[3:0]* (Byte Write Enable) When booting from SyncFlash Refer to Table 3.2.3.	EBCCR0.BC	RESET* deassert edge
ADDR[10]	Reserved Used for testing. Setting this signal to 0 disables TCK, TDI, TMS, TRST* and TDO. Ensure that the signal will not be set to 0 upon booting.	CCFG.bit[31]	RESET* deassert edge
ADDR[9]	Reserved This signal will not be set to 0 upon booting.	—	RESET* deassert edge

Table 3.2.2 Boot Configuration Specified with the ADDR[19:0], TDO, UAE and SADDR10 Signals (2/2)

Signal	Description	Corresponding Register Bit	Configuration Determined at
ADDR[8:6]	Select Boot Memory Selects boot memory. HHH = Device connected to channel 0 of the external bus controller (clock frequency division ratio: 1/1) HHL = Device connected to channel 0 of the external bus controller (clock frequency division ratio: 1/2) HLH = Device connected to channel 0 of the external bus controller (clock frequency division ratio: 1/3) HLL = Reserved LHH = PCI boot LHL = Reserved LLH = Reserved LLL = SyncFlash memory connected to SDRAMC channel 0	EBCCR0.ME	RESET* deassert edge
ADDR[5]	Boot ACK* Input Specifies the access mode for external bus controller channel 0. If SyncFlash is selected as a boot memory device, refer to Table 3.2.3. L = ACK* Input Mode Enable H = ACK* Input Mode Disable	EBCCR0.EACK	RESET* deassert edge
ADDR[4:3]	Select SYSCLK Frequency Specifies the division ratio of the SYSCLK frequency to the G-Bus clock (GBUSCLK) frequency. LL = 4 (SYSCLK frequency = GBUSCLK frequency/4) LH = 3 (SYSCLK frequency = GBUSCLK frequency/3) HL = 2 (SYSCLK frequency = GBUSCLK frequency/2) HH = 1 (SYSCLK frequency = GBUSCLK frequency)	CCFG. SYSSP	PON* deassert edge
ADDR[2]	Reserved This signal will not be set to 0 upon booting. If Syncflash is selected as a boot memory device, refer to Table 3.2.3.	—	RESET*deassert edge
ADDR[1]	PCI Arbiter Select Selects a PCI bus arbiter. H = Built-in PCI bus arbiter. L = External PCI bus arbiter.	CCFG.PCIARB	RESET* deassert edge
ADDR[0]	TX49/H2 Internal Timer Interrupt Disable Specifies whether timer interrupts within the TX49/H2 core are enabled. H = Disable timer interrupts within the TX49/H2 core. L = Enable timer interrupts within the TX49/H2 core.	CCFG.TINTDIS	RESET* deassert edge
TDO	PC Trace Specifies whether PIO[31:20] and BC32K are used as PC trace signals. H = Use as PIO[31:20] and BC32K L = Use as TPC[3:1], PCST[7:0] and DCLK	CCFG.PCTRCE	PON* deassert edge
UAE	Reserved Used for testing. Because this signal is used for setting a clock frequency, ensure that the signal will not be set to 0 upon booting.	CCFG.bit[28]	PON* deassert edge
SADDR10	Reserved Used for testing. Because this signal is used for setting a clock frequency, ensure that the signal will not be set to 0 upon booting.	—	PON* deassert edge

Table 3.2.3 Boot Configuration When SyncFlash is Used as Boot Memory

Signal	Description	Corresponding Register Bit	Configuration Determined at
ADDR[13]	Boot ROM Bus Width Specifies the data bus width when booting from SyncFlash memory. L = 32 bits H = 16 bits	SDCCR0.MW	RESET* deassert edge
ADDR[5,2]	Boot SyncFlash Memory Row Size Specifies the SyncFlash memory row size when booting from SyncFlash memory. LH = Reserved LL = 13 bits HH = 12 bits HL = 11 bits	SDCCR0.RS	RESET* deassert edge
ADDR[12:11]	Boot SyncFlash Memory Column Size Specifies the SyncFlash memory column size when booting from SyncFlash memory. HH = 11 bits HL = 9 bits LH = 10 bits LL = 8 bits	SDCCR0.CS	RESET* deassert edge

3.3 Pin Multiplexing

The TX4925 has 35 multiplexed pins as shown in Table 3.3.1. Each pin is used for different functions depending on the settings of the PCFG control register or the TDO boot configuration signal. Table 3.3.2 to Table 3.3.8 show how to set the function for each pin.

Table 3.3.1 Multiplexed Pins

Pin No.	Signal Name	Shared Functions
B5	PIO[31]	PIO[31]/CARDDIR*/BCLK/TPC[2]
D6	PIO[30]	PIO[30]/CARDREG*/PCST[8]
B6	PIO[29]	PIO[29]/CARD2CSH*/CE5*/INT[7] *1/PCST[6]
C6	PIO[28]	PIO[28]/CARD2CSL*/CE4*/INT[6] *1/PCST[7]
A6	PIO[27]	PIO[27]/CARD2WAIT** 2/CHIOUT/PCST[5]
A4	PIO[26]	PIO[26]/CARD1CSH*/DCLK
C5	PIO[25]	PIO[25]/CARD1CSL*/TPC[3]
A5	PIO[24]	PIO[24]/CARD1WAIT** 2/TPC[1]
C8	PIO[23]	PIO[23]/SPICLK/PCST[2]
A7	PIO[22]	PIO[22]/SPIIN/PCST[3]
B7	PIO[21]	PIO[21]/SPIOUT/PCST[4]
B9	PIO[20]	PIO[20]/TIMER[0]/CHIFS/PCST[1]
A8	PIO[19]	PIO[19]/TIMER[1]/CHICLK
B8	PIO[18]	PIO[18]/TCLK *2/CHIDIN
W15	PIO[17]	PIO[17]/AC_SDIN[0]/ND_WE*/TXD[1]
V16	PIO[16]	PIO[16]/AC_SDOUT/ND_RB*/RXD[1]
W16	PIO[15]	PIO[15]/AC_BITCLK/ND_CLE/RTS[1]/INT[5] *1
Y16	PIO[14]	PIO[14]/AC_SYNC/ND_RE*/CTS[1]/INT[4] *1
V15	PIO[13]	PIO[13]/AC_SDIN[1]/ND_ALE
Y15	PIO[12]	PIO[12]/AC_RST*/ND_CE*
Y13	PIO[11]	PIO[11]/TXD[0]
W13	PIO[10]	PIO[10]/RXD[0]
W14	PIO[9]	PIO[9]/RTS[0]*/INT[3] *1
Y14	PIO[8]	PIO[8]/CTS[0]*/INT[2] *1
U15	PIO[7]	PIO[7]/INT[1] *1
U13	PIO[6]	PIO[6]/INT[0] *1
V13	PIO[5]	PIO[5]/SCLK *3
W11	PIO[4]	PIO[4]/DMAACK[1]
W12	PIO[3]	PIO[3]/DMAREQ[1]
V10	PIO[2]	PIO[2]/DMAACK[0]
V12	PIO[1]	PIO[1]/DMAREQ[0]
U10	PIO[0]	PIO[0]/DMADONE
Y17	BC32K	PCST[0]
V6	BE[3]*/BWE[3]* *4	CARDIORD* *4
U6	BE[2]*/BWE[2]* *4	CARDIOWR* *4

*1: INT[7:0] are directly input to the IRC. When these signals are not used as interrupt signals, do not enable interrupts in the IRC.[m2]

*2: TCLK is directly connected to the TCLK pin of the timer module. When this signal is not used as TCLK, do not enable the external clock (TCLK) in the timer module.

*3: SCLK is directly connected to the SCLK pin of the SIO module. When this signal is not used as SCLK, do not enable the external clock (SCLK) in the SIO module.

4: BE[3]/BWE[3] functions as CARDIORD* when accessing a PCMCIA device and as BE[3]*/BWE[3] when accessing other devices. Similarly, BE[2]*/BWE[2] functions as CARDIOWR* when accessing a PCMCIA device and as BE[2]*/BWE[2] when accessing other devices.

Table 3.3.2 Function Selection for PIO[31:24] (1)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits				
			SELCARD[1]	SELCARD[0]	SELCE[1]	SELCE[0]	SELCHI
PIO[31]	TPC[2]	0	-	-	-	-	-
	CARDDIR*	1	1	-	-	-	-
		1	-	1	-	-	-
	PIO[31]	1	0	0	-	-	-
PIO[30]	PCST[8]	0	-	-	-	-	-
	CARDREG*	1	1	-	-	-	-
		1	-	1	-	-	-
	PIO[30]	1	0	0	-	-	-
PIO[29]	PCST[6]	0	-	-	-	-	-
	CARD2CSH*	1	1	-	-	-	-
	CE[5]*	1	0	-	1	-	-
	PIO[29]	1	0	-	0	-	-
PIO[28]	PCST[7]	0	-	-	-	-	-
	CARD2CSL*	1	1	-	-	-	-
	CE[4]*	1	0	-	-	1	-
	PIO[28]	1	0	-	-	0	-
PIO[27]	PCST[5]	0	-	-	-	-	-
	CHIDOUT	1	-	-	-	-	1
	CARD2WAIT*	1	1	-	-	-	0
	PIO[27]	1	0	-	-	-	0
PIO[26]	DCLK	0	-	-	-	-	-
	CARD1CSH*	1	-	1	-	-	-
	PIO[26]	1	-	0	-	-	-
PIO[25]	TPC[3]	0	-	-	-	-	-
	CARD1CSL*	1	-	1	-	-	-
	PIO[25]	1	-	0	-	-	-
PIO[24]	TPC[1]	0	-	-	-	-	-
	CARD1WAIT*	1	-	1	-	-	-
	PIO[24]	1	-	0	-	-	-

Table 3.3.3 Function Selection for PIO[23:21] (2)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits
			SELSPI
PIO[23]	PCST[2]	0	-
	SPICLK	1	1
	PIO[23]	1	0
PIO[22]	PCST[3]	0	-
	SPIIN	1	1
	PIO[22]	1	0
PIO[21]	PCST[4]	0	-
	SPIOUT	1	1
	PIO[21]	1	0

Table 3.3.4 Function Selection for PIO[20:18] (3)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits		
			SELCHI	SELTMR[1]	SELTMR[0]
PIO[20]	PCST[1]	0	-	-	-
	PIO[20]	1	0	-	0
	TIMER[0]	1	0	-	1
	CHIFS	1	1	-	-
PIO[19]	PIO[19]	-	0	0	-
	TIMER[1]	-	0	1	-
	CHICLK	-	1	-	-
PIO[18]	PIO[18]	-	0	-	-
	CHIDIN	-	1	-	-

Table 3.3.5 Function Selection for PIO[17:12] (4)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits			
			SELACLC	SELNAND	SELSIO[1]	SELSIOC[1]
PIO[17]	PIO[17]	-	0	0	0	-
	TXD[1]	-	0	0	1	-
	ND_WE*	-	0	1	-	-
	AC_SDIN0	-	1	-	-	-
PIO[16]	PIO[16]	-	0	0	0	-
	RXD[1]	-	0	0	1	-
	ND_RB*	-	0	1	-	-
	AC_SDOUT	-	1	-	-	-
PIO[15]	PIO[15]	-	0	0	-	0
	RTS[1]	-	0	0	1	1
	ND_CLE	-	0	1	-	-
	AC_BITCLK	-	1	-	-	-
PIO[14]	PIO[14]	-	0	0	-	0
	CTS[1]	-	0	0	1	1
	ND_RE*	-	0	1	-	-
	AC_SYNC	-	1	-	-	-
PIO[13]	PIO[13]	-	0	0	-	-
	ND_ALE	-	0	1	-	-
	AC_SDIN1	-	1	-	-	-
PIO[12]	PIO[12]	-	0	0	-	-
	ND_CE*	-	0	1	-	-
	AC_RST*	-	1	-	-	-

Table 3.3.6 Function Selection for PIO[11:8] (5)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits	
			SELSIO[0]	SELSIOC[0]
PIO[11]	PIO[11]	-	0	-
	TXD[0]	-	1	-
PIO[10]	PIO[10]	-	0	-
	RXD[0]	-	1	-
PIO[9]	PIO[9]	-	-	0
	RTS[0]	-	1	1
PIO[8]	PIO[8]	-	-	0
	CTS[0]	-	1	1

Table 3.3.7 Function Selection for PIO[7:5] (6)

Pin Name	Function	Boot Signal TDO
PIO[7] (*1)	PIO[7]	-
	INT[1]	-
PIO[6] (*1)	PIO[6]	-
	INT[0]	-
PIO[5] (*1)	PIO[5]	-
	SCLK	-

*1: PIO[7:5] : Refer to “*2” of “Table 3.3.1 Multiplexed Pins”.

Table 3.3.8 Function Selection for PIO[4:0] and BC32K (7)

Pin Name	Function	Boot Signal TDO	PCFG Control Bits		
			SELDMA[1]	SELDMA[0]	SELDONE
PIO[4]	PIO[4]	-	0	-	-
	DMAACK[1]	-	1	-	-
PIO[3]	PIO[3]	-	0	-	-
	DMAREQ[1]	-	1	-	-
PIO[2]	PIO[2]	-	-	0	-
	DMAACK[0]	-	-	1	-
PIO[1]	PIO[1]	-	-	0	-
	DMAREQ[0]	-	-	1	-
PIO[0]	PIO[0]	-	-	-	0
	DMADONE	-	-	-	1
BC32K	PCST[0]	0	-	-	-
	BC32K	1	-	-	-

4. Address Mapping

This chapter explains the physical address map of TX4925.

Please refer to "64-Bit TX System RISC TX49/H2 Core Architecture" about the details of mapping to a physical address from the virtual address of TX49/H2 core.

4.1 TX4925 Physical Address Map

TX4925 supports up to 4G (2^{32}) bytes of physical address.

Following resources are to be allocated in the physical address of the TX4925.

- TX4925 Internal registers (refer to "4.2 Register Map")
- SDRAM (refer to "9.3.2 Address Mapping")
- External Devices such as ROM, I/O Devices (refer to "7.3.3 Address Mapping")
- PCI Bus (refer to "10.3.4 Initiator Access")

Each resource is to be allocated in arbitrary physical addresses by the register setup. Refer to the explanation of each controller for the details of the mapping.

At initialization, only the internal registers and the memory space which stores the TX49/H2 core reset vectors are allocated shown as Figure 4.1.1. Usually ROM connected to the external bus controller channel 0 is used for the memory device that stores the reset vectors. TX4925 also supports using the memories on PCI bus as the memory device stores the reset vectors. Refer to "10.3.12 PCI Boot Configuration" for detail about this.

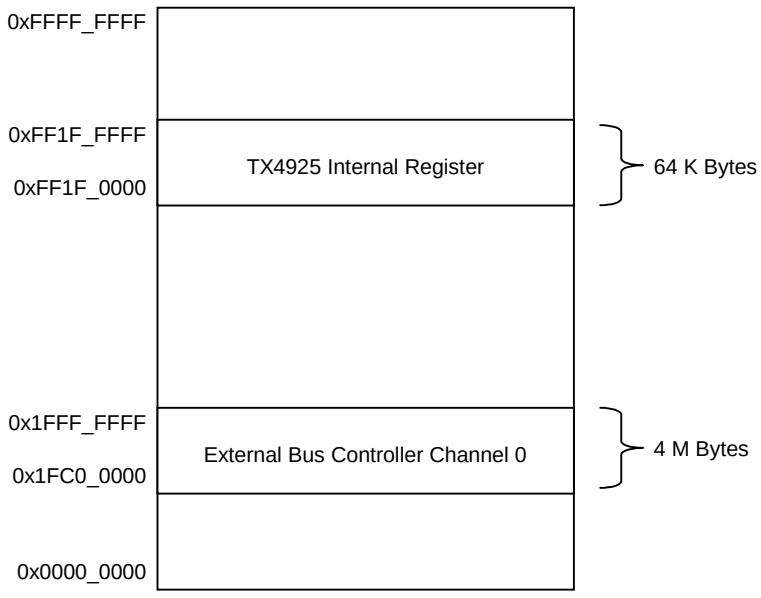


Figure 4.1.1 Physical Address Map at Initializing System

It is possible to access a resource of TX4925 as a PCI target device through PCI bus. About how to allocate resources of TX4925 to the PCI bus address space, refer to "10.3.5 Target Access".

4.2 Register Map

4.2.1 Addressing

TX4925 internal registers are to be accessed through 64 K bytes address space that is based on physical address 0xFF1F_0000 or pointed address by RAMP register (refer to 5.2.11). Figure 4.2.1 shows how to generate internal register address. Physical address 1 and physical address 2 shown Figure 4.2.1 access the same register.

In TX49/H2 Core, the physical address form 0xFF00_0000 to 0xFF3F_FFFF are unchached mapped to the virtual address form 0xFF00_0000 to 0xFF3F_FFFF.

This space includes the region form 0xFF1F_0000 allocated TX4925 internal registers at initialization.

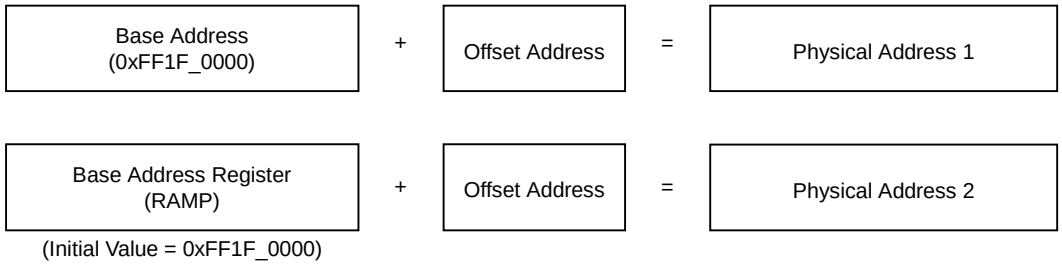


Figure 4.2.1 Generating Physical Address for a Internal Register

4.2.2 Ways to Access to Internal Registers

2 ways to access to the internal registers of TX4925 are supported. First is 32-bit register access. Second is PCI configuration register access in PCI satellite mode.

32-bit register supports 32-bit size access only. Another size access without 32-bit size is undefined.

When the build-in PCI controller works in the satellite mode (refer to “10.3.1 Terminology Explanation”), PCI configuration registers are to be accessed through PCI bus in configuration cycles. It is possible to access to the arbitrary size of PCI configuration register as always Little Endian space regardless the system setup.

4.2.3 Register Map

The outline of the register map allocated built-in controllers is shown in Table 4.2.1, and the table of the internal registers is shown in Table 4.2.2, respectively.

Please refer to “10.5 PCI Configuration Space Register” about PCI configuration register.

Table 4.2.1 Register Map

Offset Address	Peripheral Controller	Detail
0x0000 to 0x7FFF	Reserved	-
0x8000 to 0x8FFF	SDRAMC	Refer to 9.4
0x9000 to 0x9FFF	EBUSC	Refer to 7.4
0xA000 to 0xA7FF	Reserved	-
0xA800 to 0xAFFF	CHI	Refer to 16.4
0xB000 to 0xBFFF	DMAC	Refer to 8.4
0xC000 to 0xCFFF	NDFMC	Refer to 18.4
0xD000 to 0xDFFF	PCIC	Refer to 10.4
0xE000 to 0xEFFF	CONFIG	Refer to 5.2
0xF000 to 0xF0FF	TMR0	Refer to 12.4
0xF100 to 0xF1FF	TMR1	Refer to 12.4
0xF200 to 0xF2FF	TMR2	Refer to 12.4
0xF300 to 0xF3FF	SIO0	Refer to 11.4
0xF400 to 0xF4FF	SIO1	Refer to 11.4
0xF500 to 0xF50F	PIO	Refer to 13.4
0xF510 to 0xF6FF	IRC	Refer to 15.4
0xF700 to 0xF7FF	ACLC	Refer to 14.4
0xF800 to 0xF8FF	SPI	Refer to 17.4
0xF900 to 0xF9FF	RTC	Refer to 19.4
0xFA00 to 0xFFFF	Reserved	-

Table 4.2.2 Internal Registers (1/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
SDRAM Controller (SDRAMC)			
0x8000	32	SDCCR0	SDRAM Channel Control Register 0
0x8004	32	SDCCR1	SDRAM Channel Control Register 1
0x8008	32	SDCCR2	SDRAM Channel Control Register 2
0x800C	32	SDCCR3	SDRAM Channel Control Register 3
0x8020	32	SDCTR	SDRAM Timing Register
0x802C	32	SDCCMD	SDRAM Command Register
0x8030	32	SFCMD	Sync Flash Command Register
External Bus Controller (EBUSC)			
0x9000	32	EBCCR0	EBUS Channel Control Register 0
0x9004	32	EBBAR0	EBUS Base Address Register 0
0x9008	32	EBCCR1	EBUS Channel Control Register 1
0x900C	32	EBBAR1	EBUS Base Address Register 1
0x9010	32	EBCCR2	EBUS Channel Control Register 2
0x9014	32	EBBAR2	EBUS Base Address Register 2
0x9018	32	EBCCR3	EBUS Channel Control Register 3
0x901C	32	EBBAR3	EBUS Base Address Register 3
0x9020	32	EBCCR4	EBUS Channel Control Register 4
0x9024	32	EBBAR4	EBUS Base Address Register 4
0x9028	32	EBCCR5	EBUS Channel Control Register 5
0x902c	32	EBBAR5	EBUS Base Address Register 5
0x9030	32	EBCCR6	EBUS Channel Control Register 6
0x9034	32	EBBAR6	EBUS Base Address Register 6
0x9038	32	EBCCR7	EBUS Channel Control Register 7
0x903c	32	EBBAR7	EBUS Base Address Register 7
CHI Module (CHI)			
0xA800	32	CTRL	CHI Control Register
0xA804	32	PNTREN	CHI Pointer Enable Register
0xA808	32	RXPTRA	CHI Receive Pointer A Register
0xA80C	32	RXPTRB	CHI Receive Pointer B Register
0xA810	32	TXPTRA	CHI Transmit Pointer A Register
0xA814	32	TXPTRB	CHI Transmit Pointer B Register
0xA818	32	CHISIZE	CHI Size Register
0xA81C	32	RXSTRT	CHI RX Start Register
0xA820	32	TXSTRT	CHI TX Start Register
0xA824	32	HOLD	CHI TX/RX Hold Register
0xA828	32	CLOCK	CHI Clock Register
0xA82C	32	CHIINTE	CHI Interrupt Enable Register
0xA830	32	CHIINT	CHI Interrupt Status Register

Table 4.2.2 Internal Registers (2/8)

DMA Controller (DMAC)			
0xB000	32	DMCHAR0	DMA Chain Address Register 0
0xB004	32	DMSAR0	DMA Source Address Register 0
0xB008	32	DMDAR0	DMA Destination Address Register 0
0xB00C	32	DMCNTR0	DMA Count Register 0
0xB010	32	DMSAIR0	DMA Source Address Increment Register 0
0xB014	32	DMDAIR0	DMA Destination Address Increment Register 0
0xB018	32	DMCCR0	DMA Channel Control Register 0
0xB01C	32	DMCSR0	DMA Channel Status Register 0
0xB020	32	DMCHAR1	DMA Chain Address Register 1
0xB024	32	DMSAR1	DMA Source Address Register 1
0xB028	32	DMDAR1	DMA Destination Address Register 1
0xB02C	32	DMCNTR1	DMA Count Register 1
0xB030	32	DMSAIR1	DMA Source Address Increment Register 1
0xB034	32	DMDAIR1	DMA Destination Address Increment Register 1
0xB038	32	DMCCR1	DMA Channel Control Register 1
0xB03C	32	DMCSR1	DMA Channel Status Register 1
0xB040	32	DMCHAR2	DMA Chain Address Register 2
0xB044	32	DMSAR2	DMA Source Address Register 2
0xB048	32	DMDAR2	DMA Destination Address Register 2
0xB04C	32	DMCNTR2	DMA Count Register 2
0xB050	32	DMSAIR2	DMA Source Address Increment Register 2
0xB054	32	DMDAIR2	DMA Destination Address Increment Register 2
0xB058	32	DMCCR2	DMA Channel Control Register 2
0xB05C	32	DMCSR2	DMA Channel Status Register 2
0xB060	32	DMCHAR3	DMA Chain Address Register 3
0xB064	32	DMSAR3	DMA Source Address Register 3
0xB068	32	DMDAR3	DMA Destination Address Register 3
0xB06C	32	DMCNTR3	DMA Count Register 3
0xB070	32	DMSAIR3	DMA Source Address Increment Register 3
0xB074	32	DMDAIR3	DMA Destination Address Increment Register 3
0xB078	32	DMCCR3	DMA Channel Control Register 3
0xB07C	32	DMCSR3	DMA Channel Status Register 3
0xB0A4	32	DMMFDR	DMA Memory Fill Data Register
0xB0A8	32	DMMCR	DMA Master Control Register
NAND Flush Memory Controller (NDFMC)			
0xC000	32	NDFDTR	NAND Flush Memory Data Transmit Register
0xC004	32	NDFMCR	NAND Flush Memory Mode Control Register
0xC008	32	NDFSR	NAND Flush Memory Status Register
0xC010	32	NDFISR	NAND Flush Memory Interrupt Status Register
0xC014	32	NDFIMR	NAND Flush Memory Interrupt Mask Register
0xC018	32	NDFSPR	NAND Flush Memory Strobe Plus width Register
0xC024	32	NDFSTR	NAND Flush Memory Reset Register

Table 4.2.2 Internal Registers (3/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller (PCIC)			
0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
0xD010	32	P2GM0PLBASE	P2G Memory Space 0 PCI Lower Base Address Register (Base Address 0 Lower)
0xD014	32	P2GM1PLBASE	P2G Memory Space 1 PCI Lower Base Address Register (Base Address 1 Lower)
0xD018	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Base Address 2)
0xD01C	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (Base Address 3)
0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
0xD040	32	G2PTOCNT	G2P Timeout Count register (Retry Timeout Value, TRDY Timeout Value)
0xD060	32	G2PCFG	G2P Configuration Register
0xD064	32	G2PSTATUS	G2P Status Register
0xD068	32	G2PMASK	G2P Interrupt Mask Register
0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
0xD08C	32	PCIMASK	PCI Status Interrupt Mask Register
0xD090	32	P2GCFG	P2G Configuration Register
0xD094	32	P2GSTATUS	P2G Status Register
0xD098	32	P2GMASK	P2G Interrupt Mask Register
0xD09C	32	P2GCCMD	P2G Current Command Register
0xD0DC	8	Cap_ID	Capabilities ID Register
0xD0DD	8	Next_Item_Ptr	Next Item Pointer Register
0xD0DE	16	PMC	Power Management Capabilities Register
0xD0E0	16	PMCSR	Power Management Control / Status Register
0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for a diagnosis)
0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for a diagnosis)
0xD11C	32	PBACSTATE	PCI Bus Arbiter Current Status Register (for a diagnosis)

Table 4.2.2 Internal Registers (4/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
PCI Controller (PCIC)			
0xD120	32	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
0xD128	32	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
0xD130	32	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
0xD138	32	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
0xD140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
0xD144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
0xD148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
0xD14C	32	G2PIOMASK	G2P I/O Space Address Mask Register
0xD150	32	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
0xD158	32	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register
0xD160	32	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
0xD168	32	G2PIOPBASE	G2P I/O Space PCI Base Address Register
0xD170	32	PCICCFG	PCI Controller Configuration Register
0xD174	32	PCICSTATUS	PCI Controller Status Register
0xD178	32	PCICMASK	PCI Controller Interrupt Mask register
0xD180	32	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
0xD184	32	P2GM0CTR	P2G Memory Space 0 G-Bus Control Register
0xD188	32	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
0xD18C	32	P2GM1CTR	P2G Memory Space 1 G-Bus Control Register
0xD190	32	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
0xD194	32	P2GM2CTR	P2G Memory Space 2 G-Bus Control Register
0xD198	32	P2GIOGBASE	P2G I/O Space 0 G-Bus Base Address Register
0xD19C	32	P2GIOCTR	P2G I/O Space 0 G-Bus Control Register
0xD1A0	32	G2PCFGADRS	G2P Configuration Address Register
0xD1A4	32	G2PCFGDATA	G2P Configuration Data Register
0xD1B0	32	G2PIDADRS	G2P Indirect Access Address Register
0xD1B4	32	G2PIDDATA	G2P Indirect Access Data Register
0xD1B8	32	G2PIDCMD	G2P Indirect Access Command / Byte Enable Register
0xD1C8	32	G2PINTACK	G2P Interrupt Acknowledge Register
0xD1CC	32	G2PSPC	G2P Special Cycle Data Register
0xD1E0	32	PCICDATA0	PCI Configuration Data 0 Register
0xD1E4	32	PCICDATA1	PCI Configuration Data 1 Register
0xD1E8	32	PCICDATA2	PCI Configuration Data 2 Register
0xD1EC	32	PCICDATA3	PCI Configuration Data 3 Register
0xD200	32	PDMCA	PDMAC Chain Address Register
0xD204	32	PDMGA	PDMAC G-Bus Address Register
0xD208	32	PDMPA	PDMAC PCI Bus Address Register
0xD210	32	PDMCTR	PDMAC Count Register
0xD214	32	PDMCFG	PDMAC Configuration Register
0xD21C	32	PDMSTATUS	PDMAC Status Register

Table 4.2.2 Internal Registers (5/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Configuration			
0xE000	32	CCFG	Chip Configuration Register
0xE004	32	REVID	Chip Revision ID Register
0xE008	32	PCFG	Pin Configuration Register
0xE00C	32	TOEA	Timeout Error Access Address Register
0xE010	32	PDNCTR	Power Down Control Register
0xE014	32	-	-
0xE018	32	GARBC	G-Bus Arbiter Control Register
0xE01C	32	-	-
0xE020	32	TOCNT	Time Out Count Register
0xE024	32	DRQCTR	DMA Request Control Register
0xE028	32	CLKCTR	Clock Control Register
0xE02C	32	GARBC	G-Bus Arbiter Control Register
0xE030	32	RAMP	Register Address Mapping Register
0xE034	32	-	-
Timer (Channel 0)			
0xF000	32	TMTCR0	Timer Control Register 0
0xF004	32	TMTISR0	Timer Interrupt Status Register 0
0xF008	32	TMCPRA0	Compare Address Register A 0
0xF00C	32	TMCPRB0	Compare Address Register B 0
0xF010	32	TMITMR0	Interval Timer Mode Register 0
0xF020	32	TMCCDR0	Divider Register 0
0xF030	32	TMPGMR0	Plus Generator Mode Register 0
0xF0F0	32	TMTRR0	Timer Read Register 0
Timer (Channel 1)			
0xF100	32	TMTCR1	Timer Control Register 1
0xF104	32	TMTISR1	Timer Interrupt Status Register 1
0xF108	32	TMCPRA1	Compare Address Register A 1
0xF10C	32	TMCPRB1	Compare Address Register B 1
0xF110	32	TMITMR1	Interval Timer Mode Register 1
0xF120	32	TMCCDR1	Divider Register 1
0xF130	32	TMPGMR1	Plus Generator Mode Register 1
0xF1F0	32	TMTRR1	Timer Read Register 1
Timer (Channel 2)			
0xF200	32	TMTCR2	Timer Control Register 2
0xF204	32	TMTISR2	Timer Interrupt Status Register 2
0xF208	32	TMCPRA2	Compare Register A 2
0xF210	32	TMITMR2	Interval Timer Mode Register 2
0xF220	32	TMCCDR2	Divider Register 2
0xF240	32	TMWTMR2	Watch Dog Timer Register 2
0xF2F0	32	TMTRR2	Timer Read Register 2

Table 4.2.2 Internal Registers (6/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Serial I/O (Channel 0)			
0xF300	32	SILCR0	Line Control Register 0
0xF304	32	SIDICR0	DMA/Interrupt Control Register 0
0xF308	32	SIDISR0	DMA/ Interrupt Status Register 0
0xF30C	32	SISCISR0	Status Change Interrupt Status Register 0
0xF310	32	SIFCR0	FIFO Control Register 0
0xF314	32	SIFLCR0	Flow Control Register 0
0xF318	32	SIBGR0	Baud Rate Control Register 0
0xF31C	32	SITFIFO0	Transmitter FIFO Register 0
0xF320	32	SIRFIFO0	Receiver FIFO Register 0
Serial I/O (Channel 1)			
0xF400	32	SILCR1	Line Control Register 1
0xF404	32	SIDICR1	DMA/Interrupt Control Register 1
0xF408	32	SIDISR1	DMA/ Interrupt Status Register 1
0xF40C	32	SISCISR1	Status Change Interrupt Status Register 1
0xF410	32	SIFCR1	FIFO Control Register 1
0xF414	32	SIFLCR1	Flow Control Register 1
0xF418	32	SIBGR1	Baud Rate Control Register 1
0xF41C	32	SITFIFO1	Transmitter FIFO Register 1
0xF420	32	SIRFIFO1	Receiver FIFO Register 1
Parallel I/O (PIO)			
0xF500	32	PIODO	Output Data Register
0xF504	32	PIODI	Input Data Register
0xF508	32	PIODIR	Direction Control Register
0xF50C	32	PIOOD	Open Drain Control Register

Table 4.2.2 Internal Registers (7/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
Interrupt Controller (IRC)			
0xF510	32	IRFLAG0	Interrupt Request Flag 0 Register
0xF514	32	IRFLAG1	Interrupt Request Flag 1 Register
0xF518	32	IRPOL	Interrupt Request Polarity Control Register
0xF51C	32	IRRCNT	Interrupt Request Control Register
0xF520	32	IRMASKINT	Internal Interrupt Mask Register
0xF524	32	IRMASKEXT	External Interrupt Mask Register
0xF600	32	IRDEN	Interrupt Detection Enable Register
0xF604	32	IRDM0	Interrupt Detection Mode Register 0
0xF608	32	IRDM1	Interrupt Detection Mode Register 1
0xF610	32	IRLVL0	Interrupt Level Register 0
0xF614	32	IRLVL1	Interrupt Level Register 1
0xF618	32	IRLVL2	Interrupt Level Register 2
0xF61C	32	IRLVL3	Interrupt Level Register 3
0xF620	32	IRLVL4	Interrupt Level Register 4
0xF624	32	IRLVL5	Interrupt Level Register 5
0xF628	32	IRLVL6	Interrupt Level Register 6
0xF62C	32	IRLVL7	Interrupt Level Register 7
0xF640	32	IRMSK	Interrupt Mask Level Register
0xF660	32	IREDC	Interrupt Edge Detection Clear Register
0xF680	32	IRPND	Interrupt Pending Register
0xF6A0	32	IRCS	Interrupt Current Status Register

Table 4.2.2 Internal Registers (8/8)

Offset Address	Register Size (bit)	Register Symbol	Register Name
AC-link Controller (ACLC)			
0xF700	32	ACCTLEN	ACLC Control Enable Register
0xF704	32	ACCTLDIS	ACLC Control Disable Register
0xF708	32	ACREGACC	ACLC CODEC Register Access Register
0xF710	32	ACINTSTS	ACLC Interrupt Status Register
0xF714	32	ACINTMSTS	ACLC Interrupt Masked Status Register
0xF718	32	ACINTEN	ACLC Interrupt Enable Register
0xF71C	32	ACINTDIS	ACLC Interrupt Disable Register
0xF720	32	ACSEMAPH	ACLC Semaphore Register
0xF740	32	ACGPIDAT	ACLC GPI Data Register
0xF744	32	ACGPODAT	ACLC GPO Data Register
0xF748	32	ACSLTEN	ACLC Slot Enable Register
0xF74C	32	ACSLTDIS	ACLC Slot Disable Register
0xF750	32	ACFIFOSTS	ACLC FIFO Status Register
0xF780	32	ACDMASTS	ACLC DMA Request Status Register
0xF784	32	ACDMASEL	ACLC DMA Channel Selection Register
0xF7A0	32	ACAUDODAT	ACLC Audio PCM Output Data Register
0xF7A4	32	ACSURRDAT	ACLC Surround Data Register
0xF7A8	32	ACCENTDAT	ACLC Center Data register
0xF7AC	32	ACLFEDAT	ACLC LFE Data Register
0xF7B0	32	ACAUDIDAT	ACLC Audio PCM Input Data Register
0xF7B8	32	ACMODODAT	ACLC Modem Output Data Register
0xF7BC	32	ACMODIDAT	ACLC Modem Input Data Register
0xF7FC	32	ACREVID	ACLC Revision ID Register
SPI Module (SPI)			
0xF800	32	SPMCR	SPI Master Control Register
0xF804	32	SPCR0	SPI Control Register 0
0xF808	32	SPCR1	SPI Control Register 1
0xF80C	32	SPFS	SPI Inter Frame Space Register
0xF810	32	-	-
0xF814	32	SPSR	SPI Status Register
0xF818	32	SPDR	SPI Data Register
0xF81C	32	-	-
RTC Module (RTC)			
0xF900	32	RTCHI	RTC Register (High)
0xF904	32	RTCLO	RTC Register (Low)
0xF908	32	ALARMHI	Alarm Register (High)
0xF90C	32	ALARMLO	Alarm Register (Low)
0xF910	32	RTCCTRL	RTC Control Register
0xF914	32	RTCINT	RTC Interrupt Status Register

5. Configuration Register

5.1 Outline

The configuration registers set up and control the basic functionality of the entire TX4927. Refer to Section 5.2 for details of each configuration register. Also refer to sections mentioned in the description about each bit field.

5.1.1 Detecting G-Bus Timeout

The G-Bus is an internal bus of the TX4925. Access to each address on the G-Bus is completed upon a bus response from the accessed address. If an attempt is made to access an undefined physical address or if a hardware failure occurs, no bus response is made. If a bus response does not occur, the bus access will not be completed, leading to a system halt. To solve this problem, the TX4925 is provided with a G-Bus timeout detection function. This function forcibly stops bus access if no bus response occurs within the specified time.

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

If a timeout error is detected while the TX49/H2 core, as the bus master, is gaining write access to the G-Bus, the Write-Access Bus Error bit (CCFG.BEOW) is set. Enabling interrupt No. 1 in the interrupt controller makes it possible to post an interrupt to the TX49/H2 core. If a timeout error is detected while the TX49/H2 core is gaining read access to the bus, a bus error exception occurs in the TX49/H2 core.

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

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

5.2 Register

Table 5.2.1 lists the configuration registers.

Table 5.2.1 Configuration Register Map

Reference	Offset Address	Size in Bits	Mnemonic	Register Name
5.2.1	0xE000	32	CCFG	Chip Configuration Register
5.2.2	0xE004	32	REVID	Chip Revision ID Register
5.2.3	0xE008	32	PCFG	Pin Configuration Register
5.2.4	0xE00C	32	TOEA	Timeout Error Access Address Register
5.2.5	0xE010	32	PDNCTR	Power Down Control Register
—	0xE014	32	—	(Reserved)
5.2.6	0xE018	32	GARBP	GBUS Arbiter Priority Register
—	0xE01C	32	—	(Reserved)
5.2.7	0xE020	32	TOCNT	Timeout Count Register
5.2.8	0xE024	32	DRQCTR	DMA Request Control Register
5.2.9	0xE028	32	CLKCTR	Clock Control Register
5.2.10	0xE02C	32	GARBC	GBUS Arbiter Control Register
5.2.11	0xE030	32	RAMP	Register Address Mapping Register
—	0xE034	32	—	(Reserved)

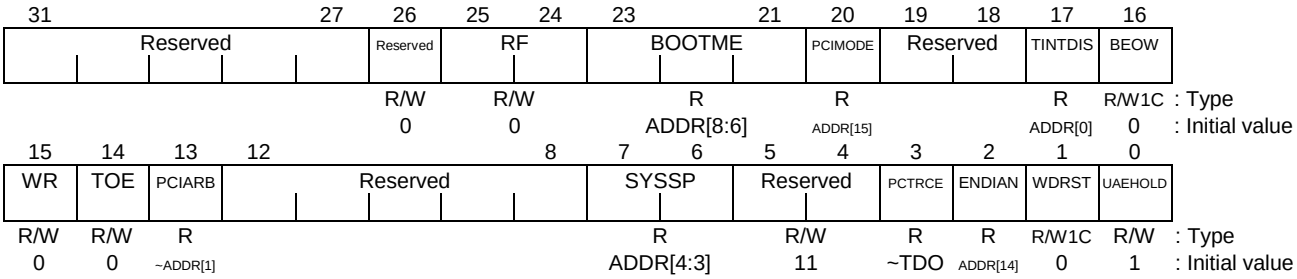
Any address not defined in this table is reserved for future use.

5.2.1 Chip Configuration Register (CCFG) 0xE000

For the bit fields whose initial values are set by boot configuration (refer to Section 3.2), the initial input signal level and the corresponding register value are indicated.

The following bits are Reserved (Read only). An explanation of the type and default was added so the default is reflected in the Boot signal.

bit 31: R ~ADDR[10], bit 28: R UAE, bit 27: R ADDR[16]
bit 18: R ADDR[17], bit 12: R ADDR[19]



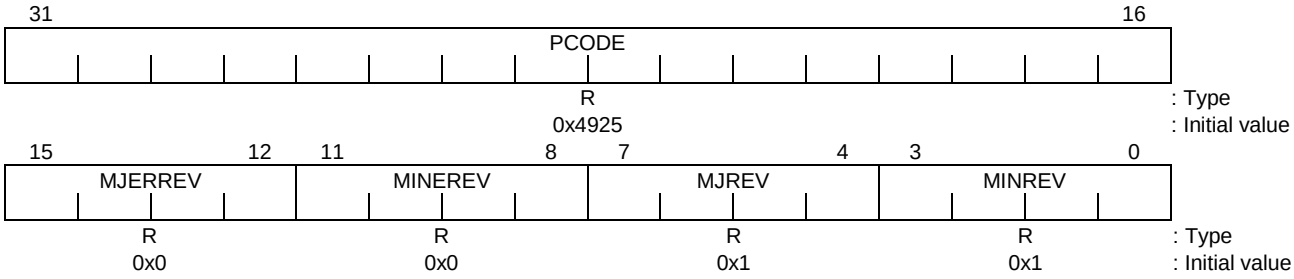
Bits	Mnemonic	Field Name	Description
31:27	—	Reserved	—
26	—	Reserved	Note: This bit is always set to “0”. (Initial value: 0, R/W)
25:24	RF	Reduced Frequency	Reduced Frequency (Initial value: 00, R/W) These bits select the internal bus speed. 00: full speed 01: 1/2 speed 10: 1/4 speed 11: 1/8 speed
23:21	BOOTME	Boot Memory	Boot Memory (Initial value: ADDR[8:6], R) Shows Boot Memory 000: SyncFlash 001: Reserved 010: Reserved 011: PCIC 100: Reserved 101: EBUSC ch0 at third speed 110: EBUSC ch0 at half speed 111: EBUSC ch0 at full speed
20	PCIMODE	PCI Mode	PCI Mode (Initial value: ADDR[15], R) Shows the PCI operation mode. L: Satellite mode H: Host mode
19:18	—	Reserved	—
17	TINTDIS	TX49/H2 core Timer Interrupt Disable	TX49/H2 core Timer Interrupt Disable (Initial value: ADDR[0], R) Shows whether TX49/H2 core Timer Interrupt is enable or disable. L: Enable H: Disable
16	BEOW	Bus Error on Write	Bus Error on Write (Initial value: 0, R/W1C) Indicates that a bus error was generated by a write operation of the TX49/H2 Core. Writing a “1” clears the bit. 0: No error occur 1: Error occurs
15	WR	Watchdog Timer for Reset/NMI	Watchdog Timer for Reset/NMI (Initial value: 0, R/W) Designates the connection of the Watchdog Timer. 0: Watchdog Timer Interrupt is connected to TX4925 internal NMI*. 1: Watchdog Timer Interrupt is connected to TX4925 internal Reset.

Figure 5.2.1 Chip Configuration Register (CCFG) (1/2)

Bits	Mnemonic	Field Name	Description
14	TOE	Timeout Enable for Bus Error	Timeout Enable for Bus Error (Initial value: 0, R/W) Designates the state of the Bus Error time-out function. 0: Disable time out function. 1: Enable time out function.
13	PCIARB	PCI arbiter	PCI arbiter. (Initial value: ADDR[1], R) Select PCI arbiter. Latched from ADDR[1] at RESET. L: 0: External arbiter H: 1: Internal arbiter
12:8	—	Reserved	—
7:6	SYSSP	SYSCLK Speed	SYSCLK Speed (Initial value: ADDR[4:3], R) Shows SYSCLK frequency. LL: 00: SYSCLK speed = GBUSCLK speed/4 LH: 01: SYSCLK speed = GBUSCLK speed/3 HL: 10: SYSCLK speed = GBUSCLK speed/2 HH: 11: SYSCLK speed = GBUSCLK speed
5:4	—	Reserved	Note: These bits are always set to “11” (Initial value: 11, R/W).
3	PCTRCE	PC Trace Enable	PC Trace Enable (Initial value: TD0, R) Shows whether PC Trace signals are enable or disable. 0: Disable 1: Enable
2	ENDIAN	Endian	Current Endian Setting (Initial value: ADDR[14], R) Shows the endian mode. L: 0: LITTLE ENDIAN H: 1: BIG ENDIAN
1	WDRST	Watchdog Reset Status	Watchdog Reset Status (Initial value: 0, R/W1C) Indicates that a watchdog reset was generated. 0: No watchdog reset occur 1: Watchdog reset occurs
0	UAEHOLD	UAE Address Hold	UAE* Address Hold (Initial value: 1, R/W) 0: Address goes away during the same clock as UAE*. 1: Address is held one clock after the rising edge of UAE*.

Figure 5.2.1 Chip Configuration Register (CCFG) (2/2)

5.2.2 Chip Revision ID Register (REVID) 0xE004



Bits	Mnemonic	Field Name	Description
31:16	PCODE	Product Code	Product Code (Initial value: 0x4925, R) This field defines the product number.
15:12	MJERREV	Major Extra Code	Major Extra Code Implementation Revision (Initial value: 0x0, R) This field defines the major extra code.
11:8	MINEREV	Minor Extra Code	Minor Extra Code Implementation Revision (Initial value: 0x0, R) This field defines the minor extra code.
7:4	MJREV	Major Revision	Major Implementation Revision (Initial value: 0x1, R) This field defines a major revision. Contact Toshiba technical staff for an explanation of the revision value.
3:0	MINREV	Minor Revision	Minor Implementation Revision (Initial value: 0x1, R) This field defines a minor revision. Contact Toshiba technical staff for an explanation of the revision value.

Figure 5.2.2 Chip Revision ID Register (REVID)

5.2.3 Pin Configuration Register (PCFG) 0xE008

For the bit fields whose initial values are set by boot configuration (refer to Section 3.2), the initial input signal level and the corresponding register value are indicated.

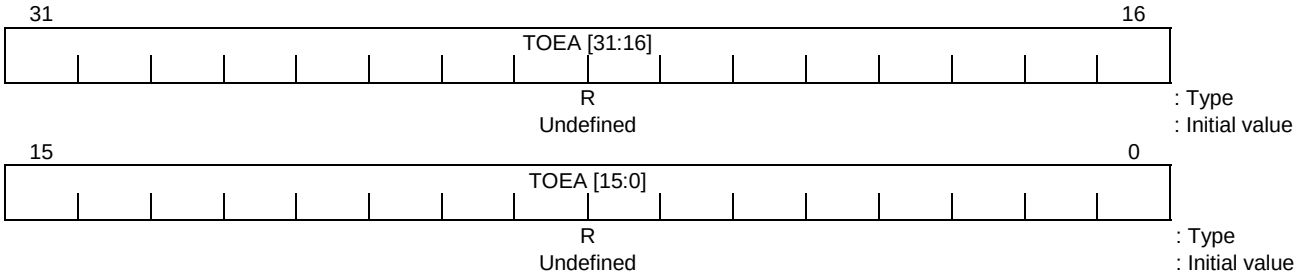
Bits	Mnemonic	Field Name	Description
31	SYSCLKEN	SYSCLK Enable	SYSCLK Enable (Initial value: 1, R/W) Specifies whether to output the SYSCLK. 1: Clock output. 0: L.
30:29	SDRCLKEN [1:0]	SDRAM Clock Enable	SDRAM Clock Enable (Initial value: 11, R/W) Individually specifies whether to output each of SDCLK [1:0]. 1: Clock output. 0: L. Bit 30 = SDCLK [1] Bit 29 = SDCLK [0]
28:27	PCICLKEN [2:0]	PCI Clock Enable	PCI Clock Enable (Initial value: ADDR[18], R/W) Individually specifies whether to output each of PCICLK [2:1]. 1: Clock output. 0: L. Bit 28 = PCICLK [2] Bit 27 = PCICLK [1]
26	PCICLKIOEN	PCI Clock I/O Enable	PCI Clock I/O Enable (Initial value: ADDR[18], R/W) Individually specifies whether to output each of PCICLK [2:1]. 1: Clock output. 0: Clock input.
25:23	—	Reserved	—
22	—	Reserved	Note: This bit is always set to “0” (Initial value: 0, R/W).
21	SELSPL	Select SPI	Select SPI (Initial value: 0, R/W) Select SPI function as PIO[23:21] pin. Please refer to “3.3 Pin Multiplexing” about setting.
20	SELCHI	Select CHI	Select CHI (Initial value: 0, R/W) Select SPI function as PIO[27,20:18] pin. Please refer to “3.3 Pin Multiplexing” about setting.
19:18	SELCARD	Select PCMCIA CARD	Select PCMCIA CARD (Initial value: 00, R/W) Select PCMCIA CARD function as PIO[31:24] pin. Please refer to “3.3 Pin Multiplexing” about setting.
17:16	SELCE	Select CE[5:4]*	Select CE[5:4]* (Initial value: 00, R/W) Select CE[5:4]* function as PIO[29:28] pin. Please refer to “3.3 Pin Multiplexing” about setting.
15:14	SELSIOC	Select SIO Control Pins	Select SIO Control Pins (Initial value: 00, R/W) Select SIO ch1, 0 control signals (RTS, CTS) as PIO[15,14,9,8] pin. Please refer to “3.3 Pin Multiplexing” about setting.
13:12	SELSIO	Select SIO	Select SIO (Initial value: 00, R/W) Select SIO ch1, 0 signals (TXD, RXD) as PIO[17,16,11,10] pin. Please refer to “3.3 Pin Multiplexing” about setting.
11	ACKIN	ACK* input	ACK* input (Initial value: 1, R/W) When this bit is one, ACK* signal is input. 0 : ACK* pin changes from input to output dynamically based on EBUSC channel settings. 1 : ACK* pin is input only.

Figure 5.2.3 Pin Configuration Register (PCFG) (1/2)

Bits	Mnemonic	Field Name	Description
10:9	SELTMR[1:0]	Select TIMER	Select TIMER (Initial value: 00, R/W) Select TIMER Pins as PIO[20:19] pin. Please refer to “3.3 Pin Multiplexing” about setting.
8	SELDONE	Select DMADONE*	Select DMADONE* (Initial value: 0, R/W) Select DMADONE* as PIO[0] pin. Please refer to “3.3 Pin Multiplexing” about setting.
7:4	—	Reserved	—
3	SELACLC	Select ACLC	Select ACLC (Initial value: 0, R/W) Select ACLC function as PIO[17:12] pin. Please refer to “3.3 Pin Multiplexing” about setting.
2	SELNAND	Select NAND Interface	Select NAND Interface (Initial value: 0, R/W) Select NAND Flash Memory function as PIO[17:12] pin. Please refer to “3.3 Pin Multiplexing” about setting.
1:0	SELDMA	Select DMA	Select DMA (Initial value: 00, R/W) Select DMA ch1,0 signals (DMAACK, DMAREQ) function as PIO[4:1] pin. Please refer to “3.3 Pin Multiplexing” about setting.

Figure 5.2.3 Pin Configuration Register (PCFG) (2/2)

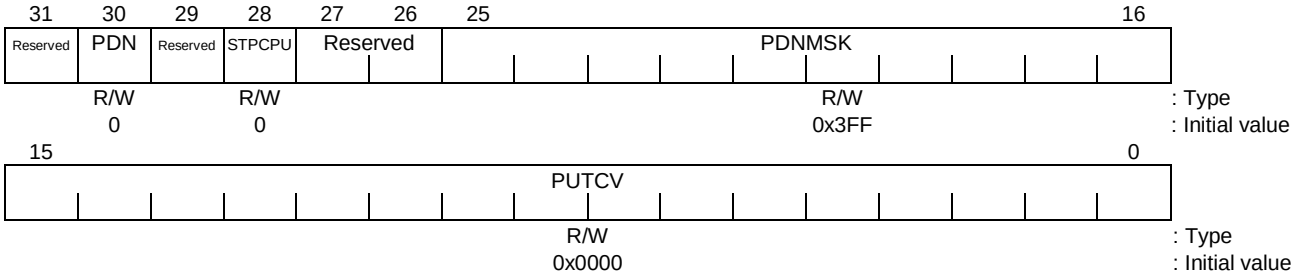
5.2.4 Timeout Error Access Address Register (TOEA) 0xE00C



Bits	Mnemonic	Field Name	Description
31:0	TOEA	Timeout Error Access Address	Timeout Error Access Address (Initial value: Undefined, R) This register latches the address on the bus when bus error occurs.

Figure 5.2.4 Timeout Error Access Address Register (TOEA)

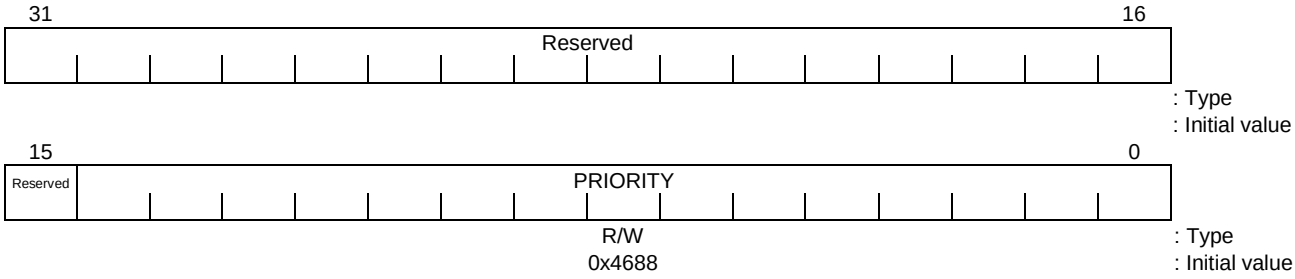
5.2.5 Power Down Control Register (PDNCTR) 0xE010



Bits	Mnemonic	Field Name	Description
31	—	Reserved	—
30	PDN	Power Down Trigger	Power Down Trigger (Initial value: 0, R/W) This bit is a trigger for the Power Down Sequence. It is set to 0 upon reset and a 0 to 1 transition will initiate the power down sequence. This bit is not cleared to 0 automatically by wakeup from power down. Please clear to 0 before next power down.
29	—	Reserved	—
28	STPCPU	Stop CPU Clock	Stop CPU Clock (Initial value: 0, R/W) When this bit is set to 1, the CPU clock stops after TX49/H2 core operates WAIT instruction and becomes the HALT mode. It is set to 0 upon reset. It is not cleared to 0 automatically by wakeup from power down mode. Please clear to 0 and set to 1 before next CPU clock stop.
27:26	—	Reserved	—
25:16	PDNMSK	Power Down Mask	Power Down Mask (Initial value: 0x3FF, R/W) Indicates which external interrupt signals wake from power down mode. A bit is allocated to each interrupt source. Set 1 and enables wakeup. 1: Interrupt Enabled 0: Interrupt Disabled PDNMSK[9] = RTC PDNMSK[8] = NMI PDNMSK[7] = INT[7] PDNMSK[6] = INT[6] PDNMSK[5] = INT[5] PDNMSK[4] = INT[4] PDNMSK[3] = INT[3] PDNMSK[2] = INT[2] PDNMSK[1] = INT[1] PDNMSK[0] = INT[0]
15:0	PUTCV	Power Up Time Counter Value	Power Up Time Counter Value (Initial value: 0x0000, R/W) This field determines the number of input clocks that expire after a wakeup has been initiated from STANDBY mode. The clock signal of this counter is MSTRCLK.

Figure 5.2.5 Power Down Control Register (PDNCTR)

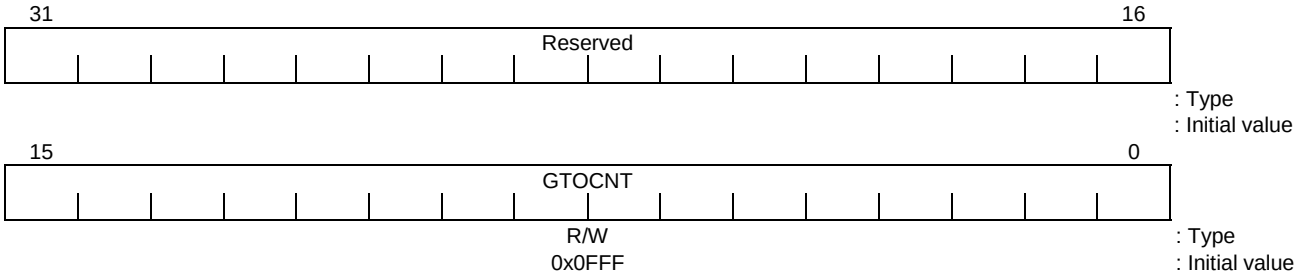
5.2.6 GBUS Arbiter Priority Register (GARBP) 0xE018



Bits	Mnemonic	Field Name	Description
31:15	—	Reserved	—
14:0	PRIORITY	Arbitration Priority	Arbitration Priority (Initial value: 0x4688, R/W) This field determines the priority when GARBC.ARBMD is cleared “0” to choose the fixed priority. Bit[14:12] = Lowest Priority Bus Master Bit[11:9] = Fourth Priority Bus Master Bit[8:6] = Third Priority Bus Master Bit[5:3] = Second Priority Bus Master Bit[2:0] = Highest Priority Bus Master 000: CHI 001: Reserved 010: PDMAC 011: PCIC 100: DMAC The initial priority is the following. CHI > PDMAC > PCIC > DMAC

Figure 5.2.6 GBUS Arbiter Priority Register (GARBP)

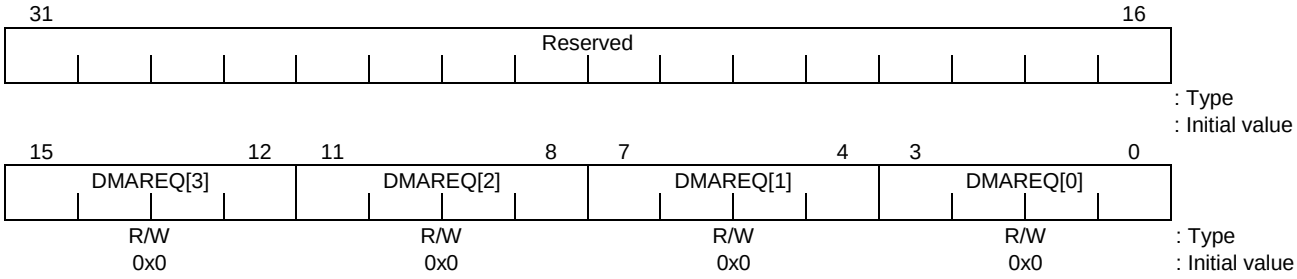
5.2.7 Timeout Count Register (TOCNT) 0xE020



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:0	GTOCNT	GBUS Timeout Count	GBUS Timeout Count (Initial value: 0x0FFF, R/W) This register defined the number of GBUSCLK before a time out occurs.

Figure 5.2.7 Timeout Count Register (TOCNT)

5.2.8 DMA Request Control Register (DRQCTR) 0xE024

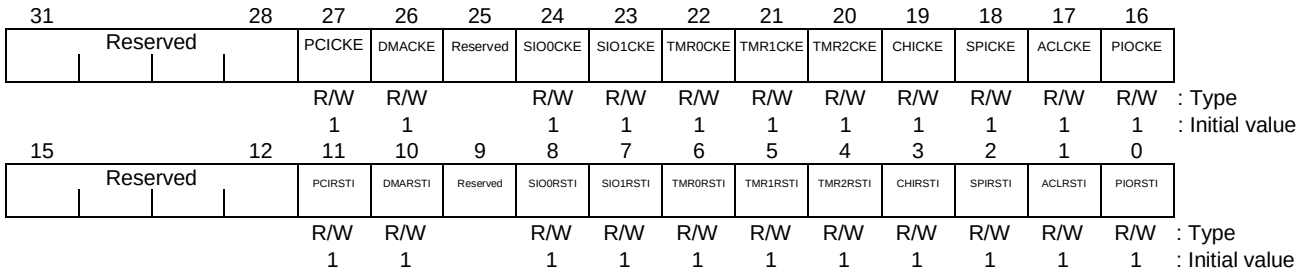


Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:12	DMAREQ[3]	DMA Request 3	DMA Request 3 (Initial value: 0x0, R/W) This field selects the DMA request source of DMAREQ[3]. 0xxx: reserved 1000: SIO ch0. Receive 1001: SIO ch1 Receive 1010: SIO ch0. Transmit 1011: SIO ch1 Transmit 1100: ACLC ch0 1101: ACLC ch1 1110: ACLC ch2 1111: ACLC ch3
11:8	DMAREQ[2]	DMA Request 2	DMA Request 2 (Initial value: 0x0, R/W) This field selects the DMA request source of DMAREQ[2]. 0xxx: reserved 1000: SIO ch0. Receive 1001: SIO ch1 Receive 1010: SIO ch0. Transmit 1011: SIO ch1 Transmit 1100: ACLC ch0 1101: ACLC ch1 1110: ACLC ch2 1111: ACLC ch3
7:4	DMAREQ[1]	DMA Request 1	DMA Request 1 (Initial value: 0x0, R/W) This field selects the DMA request source of DMAREQ[1]. 0xxx: DMAREQ[1] (external signal) 1000: SIO ch0. Receive 1001: SIO ch1 Receive 1010: SIO ch0. Transmit 1011: SIO ch1 Transmit 1100: ACLC ch0 1101: ACLC ch1 1110: ACLC ch2 1111: ACLC ch3
3:0	DMAREQ[0]	DMA Request 0	DMA Request 0 (Initial value: 0x0, R/W) This field selects the DMA request source of DMAREQ[0]. 0xxx: DMAREQ[0] (external signal) 1000: SIO ch0. Receive 1001: SIO ch1 Receive 1010: SIO ch0. Transmit 1011: SIO ch1 Transmit 1100: ACLC ch0 1101: ACLC ch1 1110: ACLC ch2 1111: ACLC ch3

Figure 5.2.8 DMA Request Control Register (DRQCTR)

5.2.9 Clock Control Register (CLKCTR) 0xE028

For the low-order 16 bits of the clock control register, canceling a reset requires that the corresponding reset bit be cleared by software. Before clearing them, wait at least 128 CPU clock cycles after they are set.



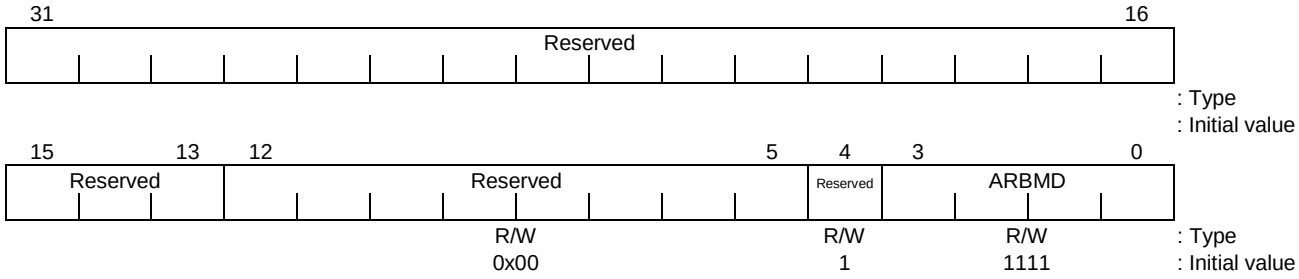
Bits	Mnemonic	Field Name	Description
31:28	—	Reserved	—
27	PCICKE	PCIC Clock Enable	PCIC Clock Enable (Initial value: 1, R/W) This bit controls the PCIC clock. 0: Stop clock 1: Supply clock
26	DMACKE	DMAC Clock Enable	DMAC Clock Enable (Initial value: 1, R/W) This bit controls the DMAC clock. 0: Stop clock 1: Supply clock
25	—	Reserved	—
24	SIO0CKE	SIO0 Clock Enable	SIO0 Clock Enable (Initial value: 1, R/W) This bit controls the SIO0 clock. 0: Stop clock 1: Supply clock
23	SIO1CKE	SIO1 Clock Enable	SIO1 Clock Enable (Initial value: 1, R/W) This bit controls the SIO1 clock. 0: Stop clock 1: Supply clock
22	TMR0CKE	TMR0 Clock Enable	TMR0 Clock Enable (Initial value: 1, R/W) This bit controls the TMR0 clock. 0: Stop clock 1: Supply clock
21	TMR1CKE	TMR1 Clock Enable	TMR1 Clock Enable (Initial value: 1, R/W) This bit controls the TMR1 clock. 0: Stop clock 1: Supply clock
20	TMR2CKE	TMR2 Clock Enable	TMR2 Clock Enable (Initial value: 1, R/W) This bit controls the TMR2 clock. 0: Stop clock 1: Supply clock
19	CHICKE	CHI Clock Enable	CHI Clock Enable (Initial value: 1, R/W) This bit controls the CHI clock. 0: Stop clock 1: Supply clock
18	SPICKE	SPI Clock Enable	SPI Clock Enable (Initial value: 1, R/W) This bit controls the SPI clock. 0: Stop clock 1: Supply clock
17	ACLCKE	ACLC Clock Enable	ACLC Clock Enable (Initial value: 1, R/W) This bit controls the ACLC clock. 0: Stop clock 1: Supply clock

Figure 5.2.9 Clock Control Register (CLKCTR) (1/2)

Bits	Mnemonic	Field Name	Description
16	PIOCKE	PIO Clock Enable	PIO Clock Enable (Initial value: 1, R/W) This bit controls the PIO clock. 0: Stop clock 1: Supply clock
15:12	—	Reserved	—
11	PCIRSTI	PCIC Reset Inactive	PCIC Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, PCIC is reset. 0: Reset 1: normal t
10	DMARSTI	DMAC Reset Inactive	DMAC Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, DMAC is reset. 0: Reset 1: normal
9	—	Reserved	—
8	SIO0RSTI	SIO0 Reset Inactive	SIO0 Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, SIO0 is reset. 0: Reset 1: normal
7	SIO1RSTI	SIO1 Reset Inactive	SIO1 Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, SIO1 is reset. 0: Reset 1: normal
6	TMR0RSTI	TMR0 Reset Inactive	TMR0 Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, TMR0 is reset. 0: Reset 1: normal
5	TMR1RSTI	TMR1 Reset Inactive	TMR1 Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, TMR1 is reset. 0: Reset 1: normal
4	TMR2RSTI	TMR2 Reset Inactive	TMR2 Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, TMR2 is reset. 0: Reset 1: normal
3	CHIRSTI	CHI Reset Inactive	CHI Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, CHI is reset. 0: Reset 1: normal
2	SPIRSTI	SPI Reset Inactive	SPI Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, SPI is reset. 0: Reset 1: normal
1	ACLRSTI	ACLC Reset Inactive	ACLC Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, ACLC is reset. 0: Reset 1: normal
0	PIORSTI	PIO Reset Inactive	PIO Reset Inactive (Initial value: 1, R/W) When this bit is set to “0”, PIO is reset. 0: Reset 1: normal

Figure 5.2.9 Clock Control Register (CLKCTR) (2/2)

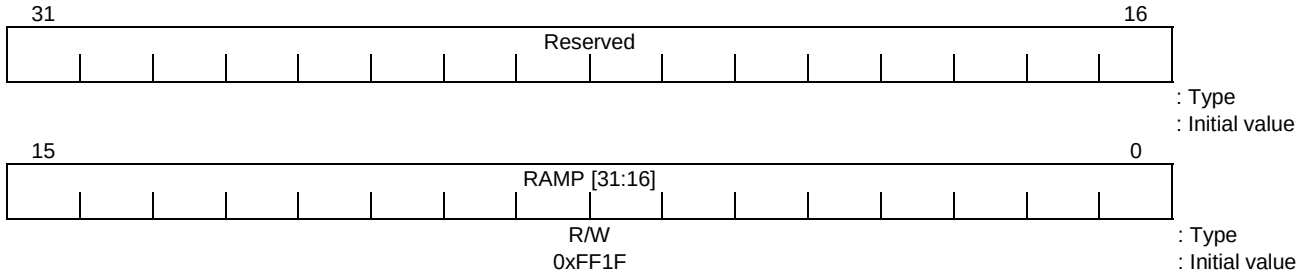
5.2.10 GBUS Arbiter Control Register (GARBC) 0xE02C



Bits	Mnemonic	Field Name	Description
31:13	—	Reserved	—
12:5	—	Reserved	Note: This bit is always set to “0” (Initial value: 0x00, R/W).
4	—	Reserved	Note: This bit is always set to “1” (Initial value: 1, R/W).
3:0	ARBMD	Arbitration Mode	Arbitration Priority (Initial value: 1111, R/W) Specifies how to prioritize G-Bus arbitration. 0000: Fixed priority. The G-Bus arbitration priority conforms to the content of the PRIORITY field (bits [14:0]). 1111: Round-robin (in a round-robin fashion, CHI > PDMAC > RTC > DMAC) Others: Reserved (Don’t set these values.) Note: Before accessing the PCI by DMAC, specify round-robin as the priority mode. If fixed-priority mode is selected, a dead lock is likely to occur in PCI bus access.

Figure 5.2.10 GBUS Arbiter Control Register (GARBC)

5.2.11 Register Address Mapping Register (RAMP) 0xE030



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:0	RAMP[31:16]	Register Address Mapping	Register Address Mapping (Initial value: 0xFF1F, R/W) This is a base address register for the TX4925 built-in registers. It holds the high-order 16 bits of a register address. The default built-in register base address is 0xFFIF_0000. Even after the content of the base address register is changed, the default value can be used to reference the built-in registers. (Refer to "4.2 Register Map".)

Figure 5.2.11 Register Address Mapping Register (RAMP)

6. Clocks

6.1 TX4925 Clock Signals

Figure 6.1.1 shows the configuration of TX4925 blocks and clock signals. Table 6.1.1 describes each clock signal. Table 6.1.2 shows the relationship among different clock signals when the CPU clock frequency is 200 MHz.

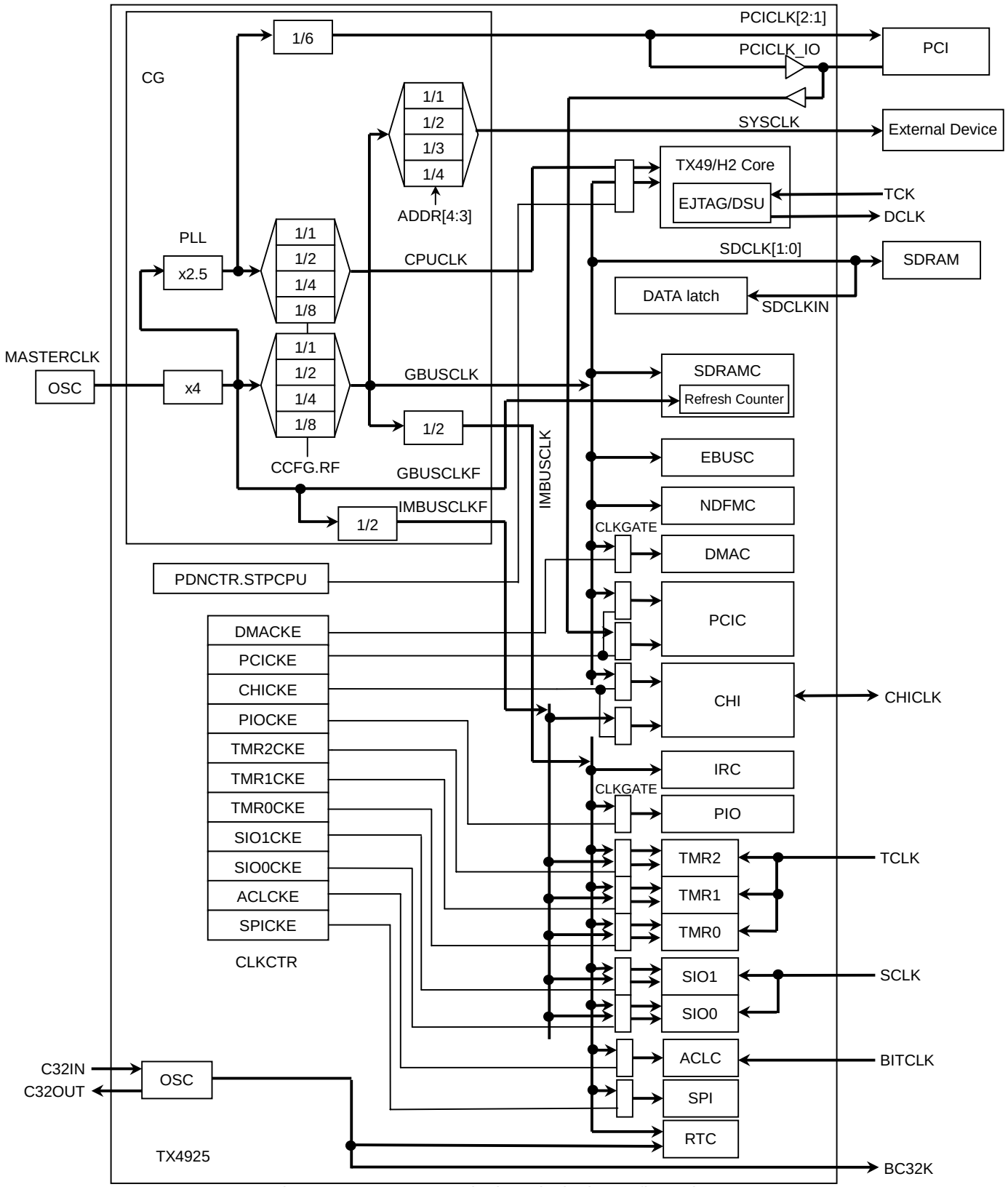


Figure 6.1.1 TX4925 Block and Clock Configuration

Table 6.1.1 TX4925 Clock Signals (1/2)

Clock	Input/ Output	Description	Related Configuration Signals (Refer to Section 3.2)	Related Registers (Refer to Chapters 5 and 10.)
MASTERCLK	Input	Master input clock for the TX4925. The TX4925 internal clock generator multiplies or divides MASTERCLK to generate internal clock pulses.	—	—
CPUCLK	Internal signal	Clock supplied to the TX49/H2 core. The PLL in the TX4925 generates CPUCLK by multiplying MASTERCLK. The value of CCFG.RF can be used to dynamically change the frequency ratio of CPUCLK to MASTERCLK. CCFG.RF[1:0] LL = 10 times MASTERCLK LH = 5 times MASTERCLK HL = 2.5 times MASTERCLK HH = 1.25 times MASTERCLK	—	CCFG.RF [1:0]
GBUSCLK	Internal signal	Clock supplied to peripheral blocks on the G-Bus. The PLL in the TX4925 generates GBUSCLK by multiplying MASTERCLK. The value of CCFG.RF can be used to dynamically change the frequency ratio of GBUSCLK to MASTERCLK. CCFG.RF[1:0] LL = 4 times MASTERCLK LH = 2 times MASTERCLK HL = 1 times MASTERCLK HH = 1/2 times MASTERCLK	—	CCFG.RF [1:0]
GBUSCLKF	Internal signal	Clock supplied to peripheral blocks on the G-Bus. The PLL in the TX4925 generates GBUSCLKF by multiplying MASTERCLK by 4. The frequency of this clock does not vary with the value of CCFG.RG. It is used for SDRAMC refresh counting.	—	—
IMBUSCLK	Internal signal	Clock supplied to peripheral modules on the IM-Bus. The frequency of IMBUSCLK is half that of GBUSCLK. In the same way as with GBUSCLK, the frequency of IMBUSCLK varies with the value of CCFG.RF.	—	CCFG.RF [1:0]
IMBUSCLKF	Internal signal	Clock supplied to peripheral modules on the IM Bus. The frequency of IMBUSCLKF is half that of GBUSCLKF. In the same way as with GBUSCLKF, the frequency of IMBUSCLKF does not vary with the value of CCFG.RF. It is used as a SIO baud rate clock or TMR count clock.	—	—
SYSCLK	Output	System clock output from the TX4925. Used by the devices connected to the external bus controller (EBUSC). Boot configuration signals ADDR[4: 3] can set the frequency ratio of SYSCLK to GBUSCLK. ADDR[4:3] LL: GBUSCLK divided by 4 LH: GBUSCLK divided by 3 HL: GBUSCLK divided by 2 HH: GBUSCLK divided by 1 In the same way as with GBUSCLK, the frequency of SYSCLK varies with the value of CCFG.RF. The SYSCLKEN bit of the PCFG register can disable the output of SYSCLK. Note:To use SYSCLK to access external devices, the SYSCLK rate must match the EBUSC channel operating rate. For details, refer to Section 7.3.8.	ADDR[4: 3]	CCFG.SYSSP PCFG.SYSCLKEN CCFG.RF [1:0]

Table 6.1.1 TX4925 Clock Signals (2/2)

Clock	Input/ Output	Description	Related Configuration Signals (Refer to Section 3.2)	Related Registers (Refer to Chapters 5 and 10.)
SDCLK[1:0]	Output	Clock supplied to SDRAM. The frequency of SDCLK[1:0] is the same as that of GBUSCLK. In the same way as with GBUSCLK, the frequency of SDCLK varies with the value of CCFG.RF. The SDCLKEN[1:0] field of the PCFG register can disable the output of SDCLK[1:0] on a per bit basis.	—	PCFG.SDCLKEN [1:0] CCFG.RF[1:0]
SDCLKIN	Input/output	Reference clock used to latch input data signals from SDRAM. The clock output from SDCLK should be connected to SDCLKIN via a feedback line outside the TX4925.	—	—
PCICLK[2:1]	Output	Clock supplied to devices on the PCI bus. The PLL in the TX4925 generates PCICLK by multiplying MASTERCLK by 5/3. These clock signals are output when the ADDR[18] boot configuration signal is set to use the clock internally generated in the TX4925 as PCICLK. Otherwise, the pins are placed in High-Z state. The PCICLKEN bit of the PCFG register can also disable the output of PCICLK after the reset sequence is completed. Note:PCICLK[2:1] can supply clock pulses at 33 MHz when the MASTERCLK frequency is set to 20 MHz.	ADDR[18]	PCFG.PCICLKEN[2:1]
PCICLKIO	Input/output	PCI bus clock. The built-in PCI controller of the TX4925 operates with this clock. A boot configuration signal (ADDR[18]) can determine whether the clock internally generated in the TX4925 is used as PCICLK. If the TX4925 internal clock is selected, the clock signals are output and simultaneously fed back to the internal PCI block. When using the PCI block, therefore, do not set the PCICLK Enable field of the pin configuration register (PCFG.PCICLKEN[0]) to 0.	ADDR[18]	PCFG.PCICLKEN[0]
SCLK	Input	Input clock for SIO. SCLK is shared by SIO0 and SIO1. The pin is shared with the PIO[5] signal.	—	—
TCLK	Input	Input clock for timers. TCLK is shared by TMR0, TMR1, and TMR2. The pin is shared with the PIO[18] signal.	—	—
BITCLK	Input	Input clock for the AC-link controller. The pin is shared with the PIO[15] signal.	—	—
CHICLK	Input/output	Clock for the CHI module. The pin is shared with the PIO[19] signal. The direction of the clock signal is set using CHICLOCK.	—	CHICLOCK
SPICLK	Output	Output clock for the SPI module. The pin is shared with the PIO[23] signal.	—	—
TCK	Input	Input clock for JTAG.	—	—
DCLK	Output	Clock output for the real-time debugging system.	TDO	—

Table 6.1.2 Relationship Among Different Clock Frequencies

Input Clock		Internal Clock					External Clock (Output)					
MASTERCLK (MHz)	RF Setting	CPUCLK (MHz)	GBUSCLK (MHz)	GBUSCLKF (MHz)	IMBUSCLK (MHz)	IMBUSCLKF (MHz)	SDCLK [1:0] (MHz)	SYSCLK (MHz) Boot Configuration Setting ADDR[4:3]				PCICLK[2:1], PCICLK_IO (MHz)
								11	10	01	00	
20	00	200	80	80	40	40	80	80	40	26.7	20	33
	01	100	40	80	20	40	40	40	20	13.3	10	
	10	50	20	80	10	40	20	20	10	6.7	5	
	11	25	10	80	5	40	10	10	5	3.3	2.5	

6.2 Power-Down Mode

6.2.1 Halt Mode and Doze Mode

The WAIT instruction causes the TX49/H2 core to enter either of the two low-power modes: Halt and Doze. The TX49/H2 can exit from Halt or Doze mode upon an interrupt exception. Ensure, therefore, that the TX49/H2 does not enter Halt or Doze mode when all interrupts are masked in the interrupt controller.

The HALT bit of the TX49/H2 core Config register is used to select Halt or Doze mode. As the TX4925 does not use the snoop function of the TX49/H2 core, the bit should be set to select Halt mode, which achieves greater power reduction than Doze mode.

Setting PDNCTR.STPCPU to 1 and then executing the WAIT instruction can stop the clock input to the TX49/H2 core, thus further reducing power dissipation.

The PDNCTR.STPCPU bit is not automatically cleared to 0 when the device exits from Halt or Doze mode. To subsequently reenter Halt or Doze mode, first clear the bit to 0 in the program.

6.2.2 Power Reduction for Peripheral Modules

When the system does not use the DMA controller, PCI controller, CHI module, serial I/O controller, timers/counters, SPI module, parallel I/O controller, or AC-link controller, it can stop the input clock for that module to reduce power dissipation.

The clock control register (CLKCTR) is used to control whether to turn each clock on or off. The module should be reset before its clock can be turned on or off. This reset is performed using the reset bit for the specific module, provided in the clock control register. The reset also initializes the registers of the module, thus requiring subsequent setup of necessary register values and other configurations. Refer to Section 5.2.9, “Clock Control Register” for detail of the clock control register (CLKCTR).

6.2.3 Power-Down Mode

Setting the PDNCTR.PDN bit to 1 can stop all clocks output from the CG. This also stops the PLL. In this state, only the RTC operates with the 32 kHz clock.

There is some delay between the time when the PDNCTR.PDN bit is set to 1 and the time when the PLL and clock outputs actually stop. To prevent a bus cycle from occurring during power-down or power-up transition, execute the program from cache.

Use an external interrupt or RTC interrupt to exit from power-down mode. PDNCTR.PDNMSK specifies which interrupt will be used. Note that if the device enters power-down mode with all interrupts disabled, you can only restore device operation by resetting it.

The PDNCTR.PDN bit is not automatically cleared to 0 when the device exits from power-down mode. To subsequently reenter power-down mode, first clear the bit to 0 in the program.

6.3 Power-On Sequence

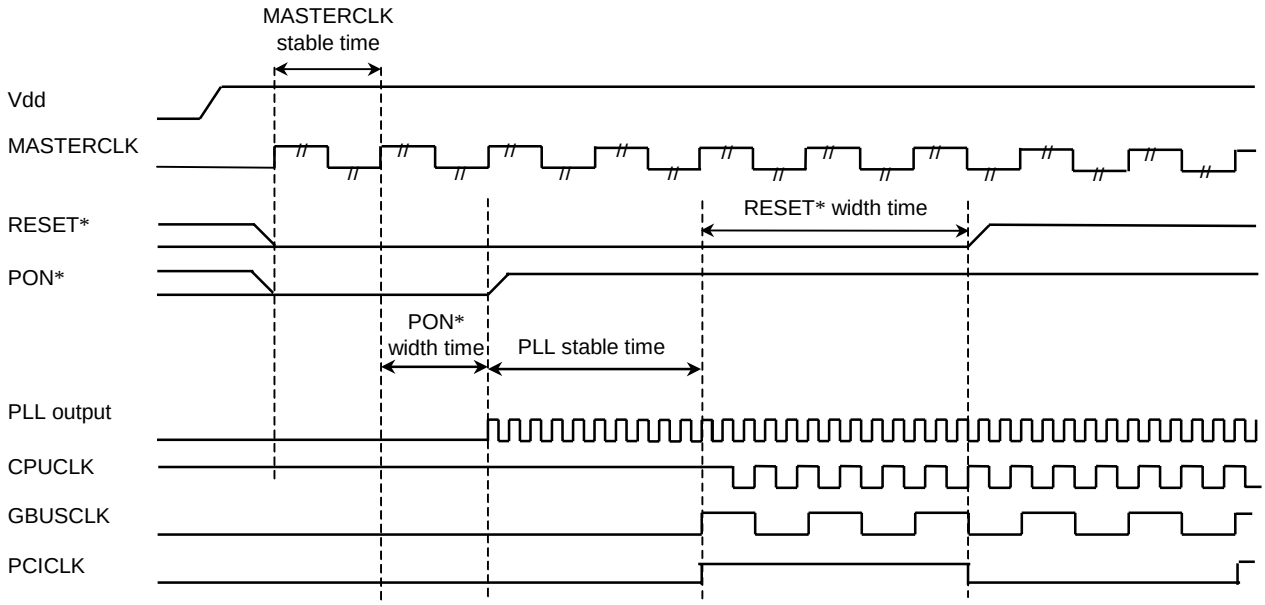


Figure 6.3.1 Power-On Sequence

7. External Bus Controller

7.1 Features

The External Bus Controller is used for accessing ROM, SRAM memory, and I/O peripherals. The features of this bus are described below.

- 8 independent channels (Channel 6 and 7 are used only PCMCIA mode.)
- Supports access to ROM (mask ROM, page mode ROM, EPROM, EEPROM), SRAM, flash memory, and I/O peripherals.
- Selectable data bus width of 8-bit, 16-bit, 32-bit for each channel
- Selectable full-speed, 1/2 speed, 1/3 speed, 1/4 speed for each channel
- Programmable timing for each channel. Programmable setup and hold time of address, chip enable, write enable, and output enable signals.
- Supports memory sizes from 1 MB to 1 GB for devices with a 32-bit data bus. Supports memory sizes from 1 MB to 512 MB for devices with a 16-bit data bus. Supports memory sizes from 1 MB to 256 MB for devices with an 8-bit data bus.
- Supports special DMAC Burst access (address decrement/fixed).
- Supports critical word first access of the TX49/H2 core.
- Supports page mode memory. Supports 4-, 8-, and 16-page size.
- Supports the External Acknowledge Signal (ACK*) and External Ready Signal modes.
- Channel 0 and 7 can be used as Boot memory. Boot settings can be made from the following selections:
  - Data bus width: 8-bit, 16-bit, 32-bit
  - ACK* output or ACK* input
  - BWE pin (byte enable or byte Write enable)
  - Boot channel clock frequency

7.2 Block Diagram

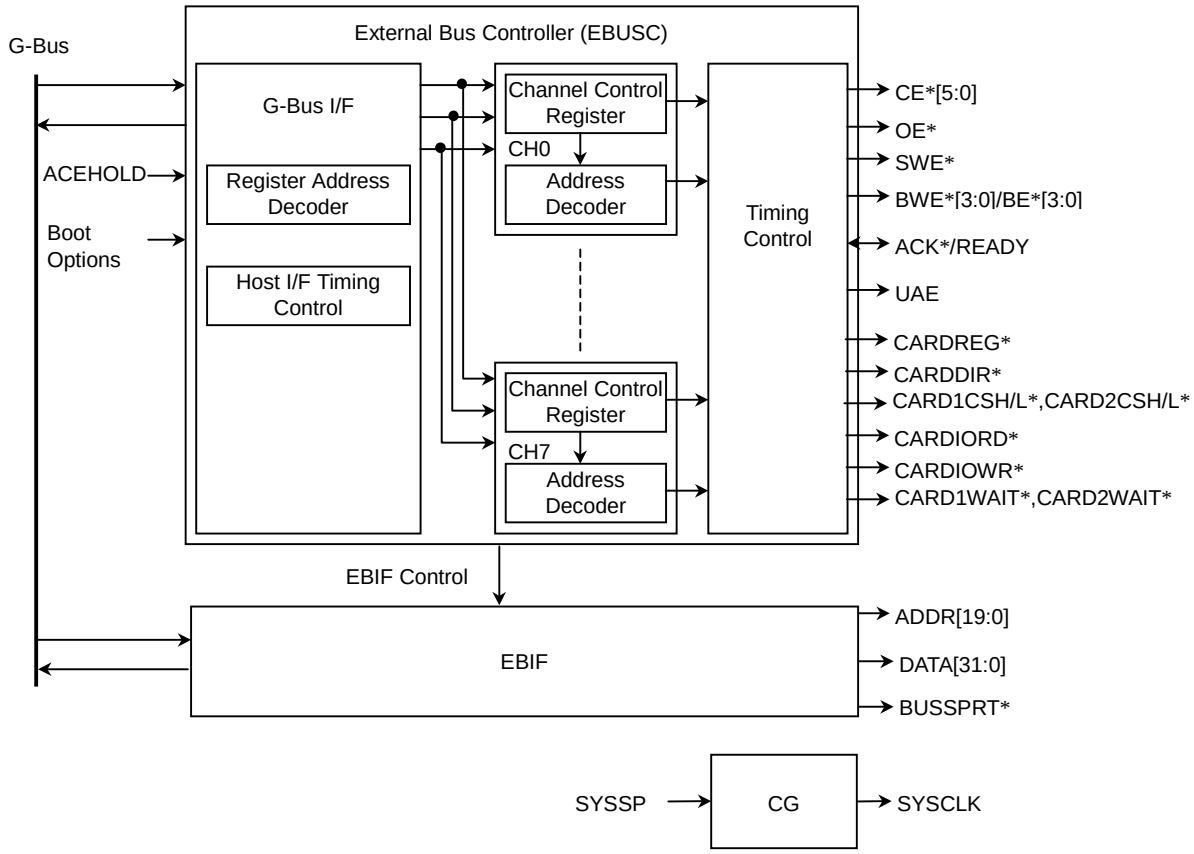


Figure 7.2.1 Block Diagram of External Bus Controller

7.3 Detailed Explanation

7.3.1 External Bus Control Register

The External Bus Controller (EBUSC) has eight channels. This register contains one Channel Control Register (EBCCRn) for each channel, and all settings can be made independently for each channel. However channel 6 and 7 are used only PCMCIA mode because TX4925 hasn't CE[7:6] signals.

Word access is possible for a Control Register. Be sure to make any Enable settings to EBCCRn.ME last. If EBCCRn.ME is enabled before setting the base address, then unintended memory access may result.

7.3.2 Global/Boot-up Options

In addition to the settings made separately for each channel, the Channel Control Registers can also use global options that make settings common to all channels.

External Bus Controller Channel 0 and 7 can be used as a Boot memory channel. Channel 0 and 7 is set by the external pins (Boot pins) during reset.

These settings are summarized below in Table 7.3.1. (Please refer to "3.2 Boot Configuration" and "5.2.1 Chip Configuration Register" for more information.)

Table 7.3.1 Global/Boot-up Options (1/2)

Pin Name	Set Register	Explanation
—	PCFG.ACKIN	Selects the operation mode of the ACK*/READY signal. 0 = ACK*/READY Dynamic mode (Default) 1 = ACK*/READY Static mode
—	CCFG.UAEHOLD	Sets the address hold time relative to the UAE signal. 0: Address changes simultaneous to deassertion of the UAE signal. 1: Address changes 1 clock cycle after deassertion of the UAE signal. (Default)
ADDR[4:3]	CCFG.SYSSP	Specifies the division ratio of the SYSCLK output relative to the internal bus clock (GBUSCLK). 00: 1/4 speed (1/4 the GBUSCLK frequency) 01: 1/3 speed (1/3 the GBUSCLK frequency) 10: 1/2 speed (1/2 the GBUSCLK frequency) 11: Full speed (same frequency as the GBUSCLK frequency)
ADDR[8:6]	EBCCR0.ME	Specifies whether to enable or disable Channel 0. 0: Disable this channel as a Boot channel. (ADDR[8] = 0) 1: Enable this channel as a Boot channel. (ADDR[8] = 1)
	EBCCR0.SP	Specifies the operation speed of Channel 0. 00: 1/4 Speed mode (ADDR[8:6] = 100b) 01: 1/3 Speed mode (ADDR[8:6] = 101b) 10: 1/2 Speed mode (ADDR[8:6] = 110b) 11: Full Speed mode (ADDR[8:6] = 111b)
ADDR[11]	EBCCR0.BC	When accessing Channel 0, specifies whether to use the BWE[3:0] signal as a Byte Enable signal (BE[3:0]) or to use it as a Byte Write Enable signal (BWE[3:0]). 0: Byte Enable mode 1: Byte Write Enable mode
ADDR[5]	EBCCR0.EACK	Specifies the Channel 0 access mode. 0: Disable ACK* Input Mode (ADDR[5] = H) 1: Enable ACK* Input Mode (ADDR[5] = L)
ADDR[13:12]	EBCCR0.BSZ	Specifies the memory bus width of Channel 0. 00: Reserved 01: 32-bit width 10: 16-bit width 11: 8-bit width

7.3.3 Address Mapping

Each of the eight channels can use the Base Address field (EBBAR.BA[31:20]) and the Channel Size field (EBCCRn.CS[3:0]) of the External Bus Channel Control Register to map to any physical address.

A channel is selected when the following equation becomes True.

paddr[31:20] & !Mask[31:20] == BA[31:20] & !Mask[31:20]

In the above equation, paddr represents the accessed physical address, Mask[31:20] represents the address mask value selected from Table 7.3.1 from the Channel Size, the ampersand (&) represents the AND operation, and the exclamation mark (!) represents the Logical NOT for each bit.

Operation is indeterminate when either multiple channels are selected simultaneously, or a channel is selected simultaneously with the SDRAM Controller or PCI Controller.

Table 7.3.2 Address Mask

CS[3:0]	Channel Size	Address Mask[31:20]
0000	1 MB	0000_0000_0000
0001	2 MB	0000_0000_0001
0010	4 MB	0000_0000_0011
0011	8 MB	0000_0000_0111
0100	16 MB	0000_0000_1111
0101	32 MB	0000_0001_1111
0110	64 MB	0000_0011_1111
0111	128 MB	0000_0111_1111
1000	256 MB	0000_1111_1111
1001	512 MB	0001_1111_1111
1010	1 GB	0011_1111_1111
1011	Reserved	Reserved
1100	Reserved	Reserved
1101	Reserved	Reserved
1110	Reserved	Reserved
1111	Reserved	Reserved

7.3.4 External Address Output

The maximum memory space size for each channel is 1 GB (230B). Addresses are output by dividing the 20-bit ADDR[19:0] signal into two parts: the upper address and the lower address. The address bit output to each bit of the ADDR[19:0] signal changes according to the setting of the channel data bus width. (See "7.3.5 Data Bus Size" for more information.)

It is possible for an external device to latch the upper eight address bits using the UAE signal. Either the UAE signal itself can be used as a Latch Enable signal or the upper address can be latched at the rise of SYSCLK when the UAE signal is being asserted.

The ADDR signal output is held for one clock cycle after the UAE signal assertion when the CCFG.UAEHOLD bit is set (default). (See Figure 7.5.1.) The ADDR signal output is not held when the CCFG.UAEHOLD bit is cleared. This hold time setting is applied globally to all channels.

The UAE signal of the upper address is always asserted at the first external bus access cycle after Reset. In all subsequent external bus access cycles, the bit mapping of the upper address output to ADDR[19:12] is compared to the bit mapping of the upper address output to ADDR[19:12] previously. The upper address is output and the UAE signal is asserted only if the compared results do not match.

As indicated below in Table 7.3.3, in the case of channel sizes that do not use the upper address latched by the UAE signal, with the exception of the first cycle after reset, the upper address is not output and the UAE signal is not asserted.

Table 7.3.3 Relationship Between the Upper Address Output and the Channel Size (CS)

CS / Bus Width	1 MB	2 MB	4 MB	8 MB or more
32 bits	—	—	—	√
16 bits	—	—	√	√
8 bits	—	√	√	√

√: The upper address output changes when the upper address changes.
—: The upper address output does not change (with the exception of the first cycle after reset.)

7.3.5 Data Bus Size

The External Bus Controller supports devices with a data bus width of 8 bits, 16 bits, and 32 bits. The data bus width is selected using the BSZ field of the Channel Control Register (EBCCRn). The address bits output to each bit of the ADDR[19:0] signal change according to the mode. When access of a size larger than the data bus width is performed, the dynamic bus sizing function is used to execute multiple bus access cycles in order from the lower address.

7.3.5.1 32-bit Bus Width Mode

DATA[31:0] becomes valid.

Bits [21:2] of the physical address are output to ADDR[19:0]. The internal address bits [29:22], which are the upper address, are multiplexed to external ADDR[19:12]. The maximum memory size is 1 GB.

Table 7.3.4 Address Output Bit Correspondence in the 32-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	29	28	27	26	25	24	23	22												
Lower Address	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

When a Single cycle that accesses 1-Byte/1 half-word/1-word data is executed, 32-bit access is executed only once on the external bus. 32-bit access is executed twice when performing 1-double-word access. When a Burst cycle is executed, one 32-bit cycles are executed for each Burst access when the Bus cycle tries to request a byte combination other than word data.

7.3.5.2 16-bit Bus Width Mode

DATA[15:0] becomes valid.

Bits [20:1] of the physical address are output to ADDR[19:0]. The internal address bits [28:21], which are the upper address, are multiplexed to external ADDR[19:12]. In other words, the address is shifted up one bit relative to the 32-bit bus mode when output. As a result, the maximum memory size of the 16-bit bus mode is 512 MB.

Table 7.3.5 Address Output Bit Correspondence in the 16-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	28	27	26	25	24	23	22	21												
Lower Address	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

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

7.3.5.3 8-bit Bus Width Mode

DATA[7:0] becomes valid.

Bits [19:0] of the physical address are output to ADDR[19:0]. The internal address bits [27:20], which are the upper address, are multiplexed to external ADDR[19:12]. In other words, the address is shifted up two bits or more relative to the 32-bit bus mode when output. As a result, the maximum memory size of the 8-bit bus mode is 256 MB.

Table 7.3.6 Address Output Bit Correspondence in the 8-bit Mode

ADDR Bit	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Address	27	26	25	24	23	22	21	20												
Lower Address	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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

7.3.6 Access Modes

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

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

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

(1) ACK*/READY Dynamic mode (PCFG.ACKIN = 0)
This mode is selected in the initial state.

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

Please refer to the timing diagrams (Figure 7.5.23 and Figure 7.5.24) and be careful to avoid conflicts when switching from output to input.

(2) ACK*/Ready Static mode (PCFG.ACKIN = 1)

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

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

Table 7.3.7 Operation Mode

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

7.3.6.1 Normal Mode

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

Wait cycles are inserted according to the EBCCRn.PWT and EBCCRn.WT value at the access cycle. The Wait cycle count is 0 – 0x3f.

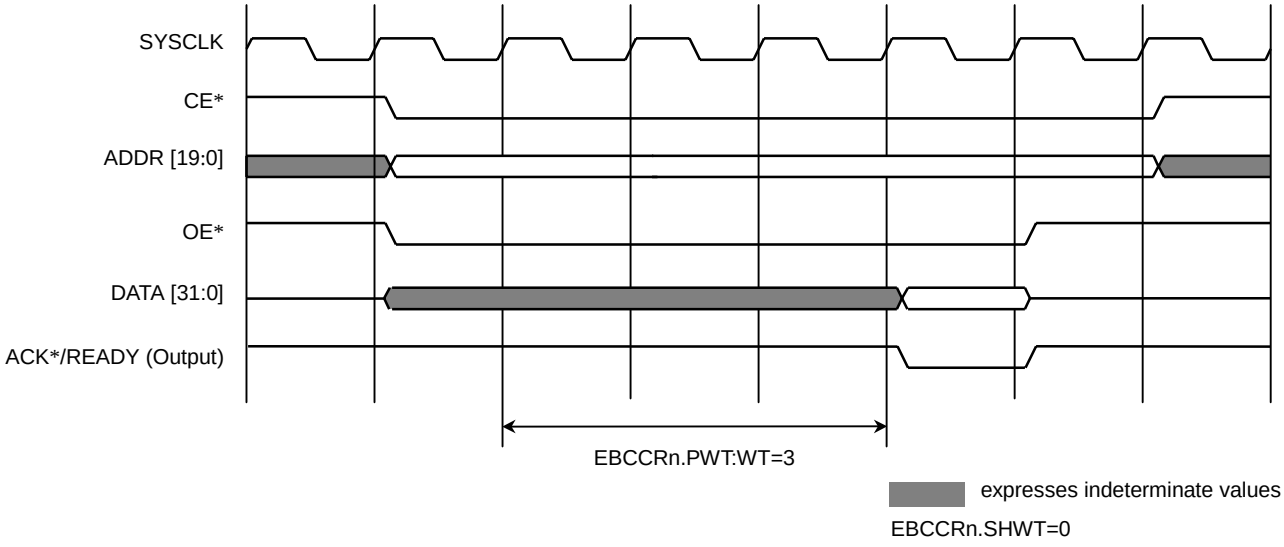


Figure 7.3.1 Normal Mode

7.3.6.2 External ACK Mode

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

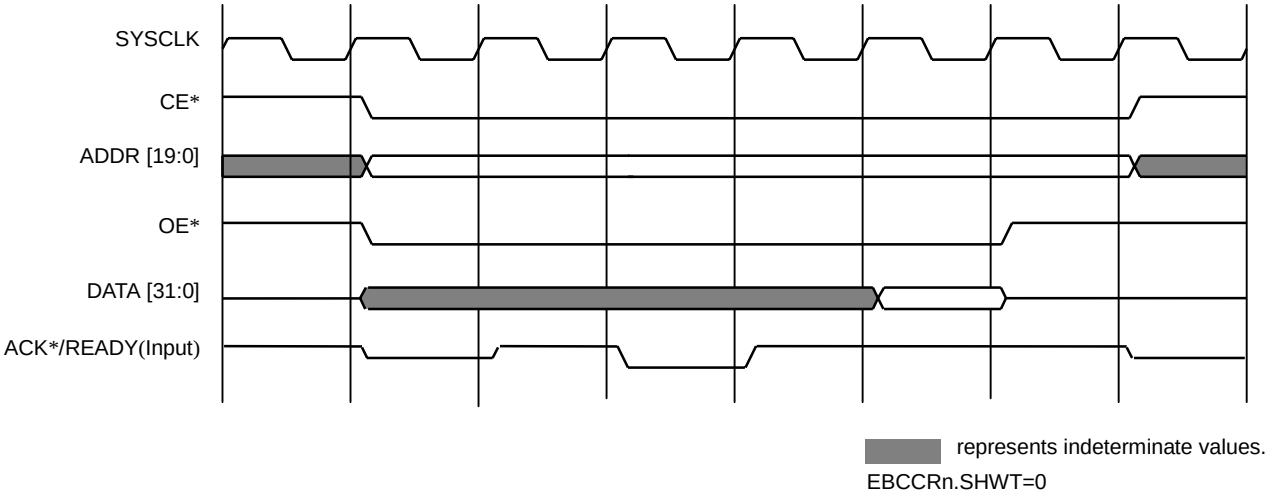


Figure 7.3.2 External ACK Mode

7.3.6.3 Ready Mode

When in this mode, the ACK*/Ready pin becomes Ready input, and the cycle is ended by Ready input from an external device. Ready input is internally initialized. See Section "7.3.7.5 Ready Input Timing" for more information regarding the operation timing.

When the Wait cycle count specified by EBCCREBCCRn.PWT:WT elapses, a check is performed to see whether the Ready signal was asserted.

The Ready mode does not support Burst access by the internal bus.

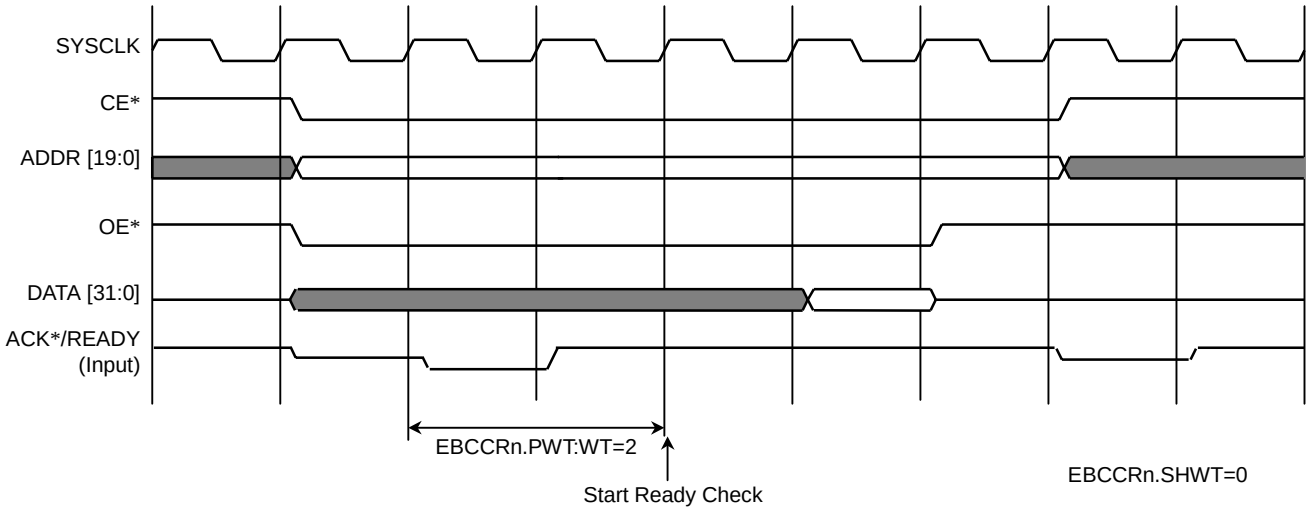


Figure 7.3.3 Ready Mode

7.3.6.4 Page Mode

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

Wait cycles are inserted into the access cycle according to the values of EBCCRn.PWT and EBCCRn.WT. The Wait cycle count in the first access cycle of Single access or Burst access is determined by the EBCCRn.WT value. The Wait cycle count can be set from 0 to 15. The Wait cycle count of subsequent Burst cycles is determined by the EBCCRn.PWT value. The Wait cycle count can be set from 0 to 3.

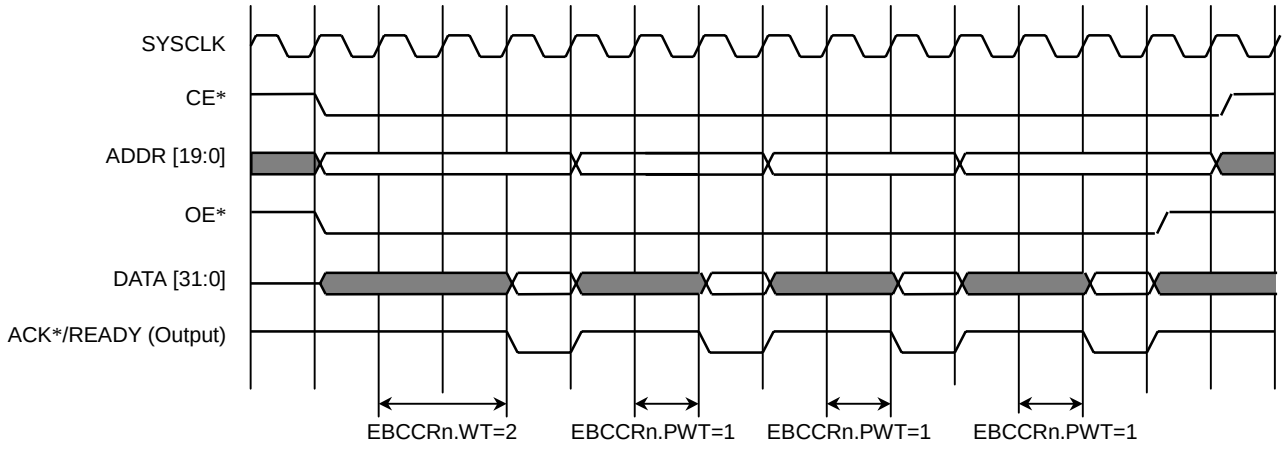


Figure 7.3.4 Page Mode

7.3.7 Access Timing

7.3.7.1 SHWT Option

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

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

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

This option is used for I/O devices that are generally slow. All Setup cycles and Hold cycles will be identical, so each cycle cannot be set individually.

The SHWT mode cannot be used by the Page mode. The SHWT mode can be used by all other modes, but there is one restriction: the internal bus cannot use Burst access.

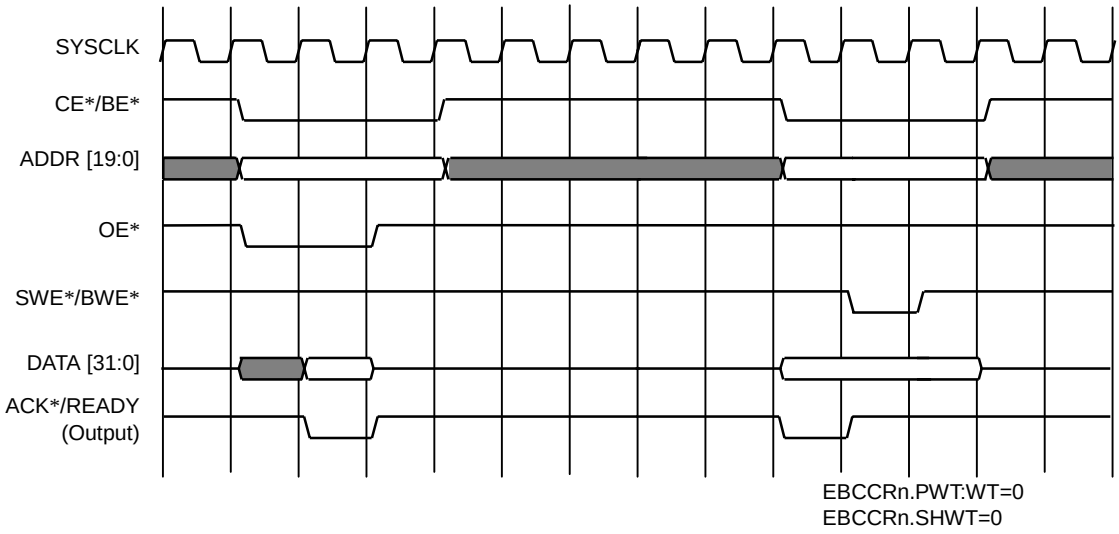


Figure 7.3.5 SWHT Disable (Normal Mode, Single Read/Write Cycle)

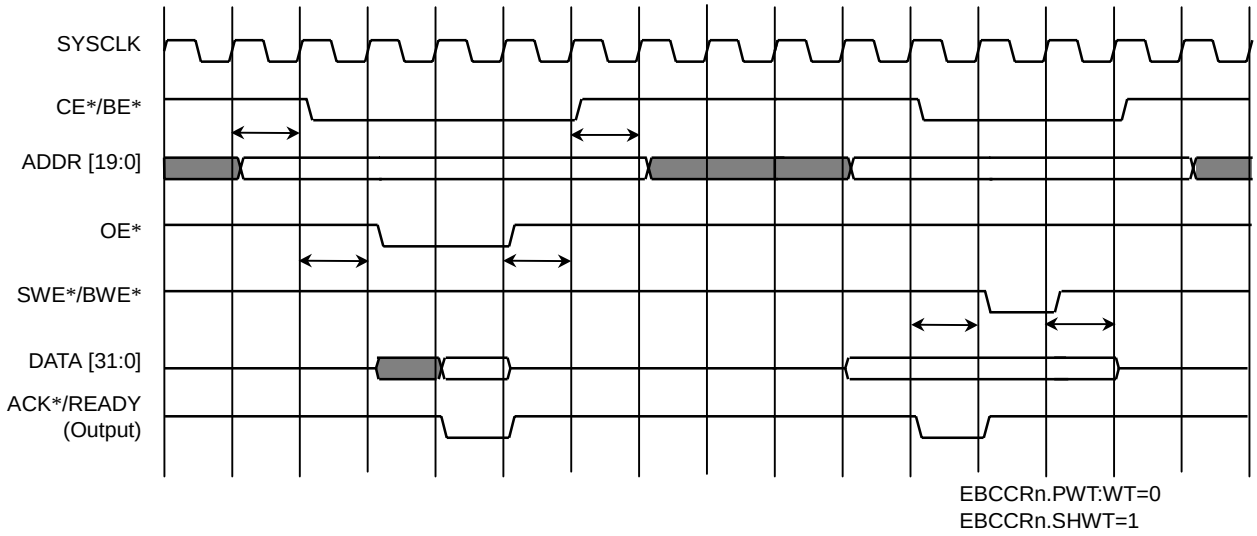


Figure 7.3.6 SHWT 1 Wait (Normal Mode, Single Read/Write Cycle)

7.3.7.2 ACK*/READY Input/Output Switching Timing

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

During External ACK mode or Ready mode access, the ACK* signal becomes High-Z at the cycle where the CE* signal is asserted. At the end of the access cycle, the ACK* signal is output (driven) again one clock cycle after the CE* signal is deasserted (see Figure 7.3.3).

7.3.7.3 ACK* Output Timing (Normal Mode, Page Mode)

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

During the Read cycle, the data is latched at the rise of the next clock cycle after when the ACK* signal is asserted. (See Figure 7.3.7 ACK* Output Timing (Single Read Cycle)).

During the Write cycle, SWE*/BWE* is deasserted at the next clock cycle after when the ACK* signal is deasserted, and the data is held for one more clock cycle after that. (See Figure 7.3.8 ACK* Output Timing (Single Write Cycle)).

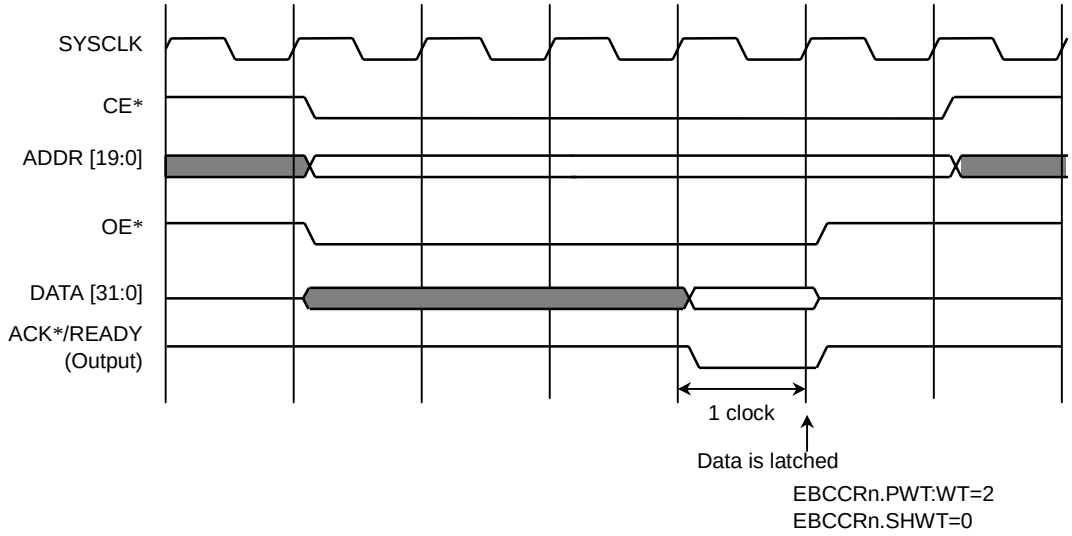


Figure 7.3.7 ACK* Output Timing (Single Read Cycle)

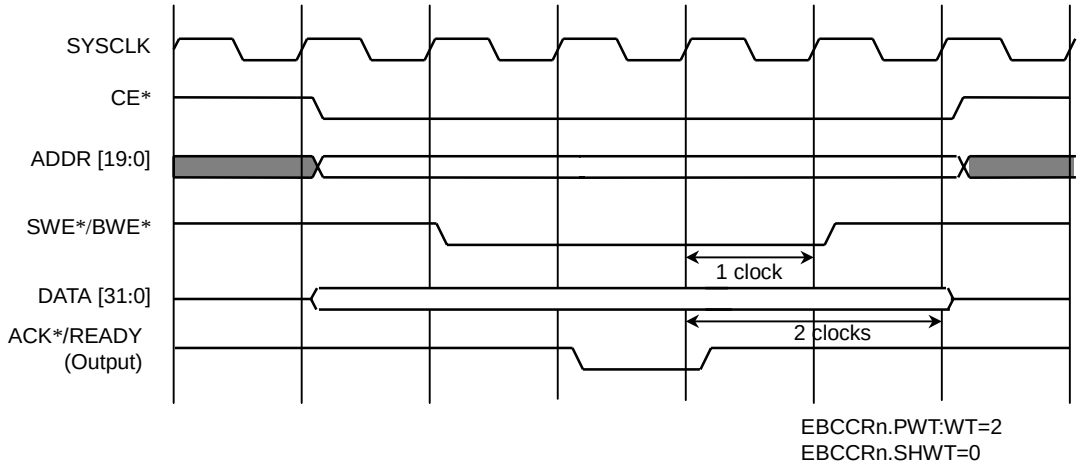


Figure 7.3.8 ACK* Output Timing (Single Write Cycle)

7.3.7.4 ACK* Input Timing (External ACK Mode)

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

During a Read cycle, data latched timing is selectable from two cases by EBCCRn.LDEA bit. When EBCCRn.LDEA is zero, data is latched two clock cycles after assertion of the ACK* signal is acknowledged (Figure 7.3.9 ACK* Input Timing (Single Read Cycle)). When EBCCRn.LDEA is one, data is latched at the assertion of the ACK* signal is acknowledged (Figure 7.3.10 ACK* Input Timing (Single Read Cycle)). During a Write cycle, assertion of the ACK* signal is acknowledged, SWE*/BWE* is deasserted three clock cycles later, then data is held for one clock cycle after that (Figure 7.3.11 ACK* Input Timing (Single Write Cycle).

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

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

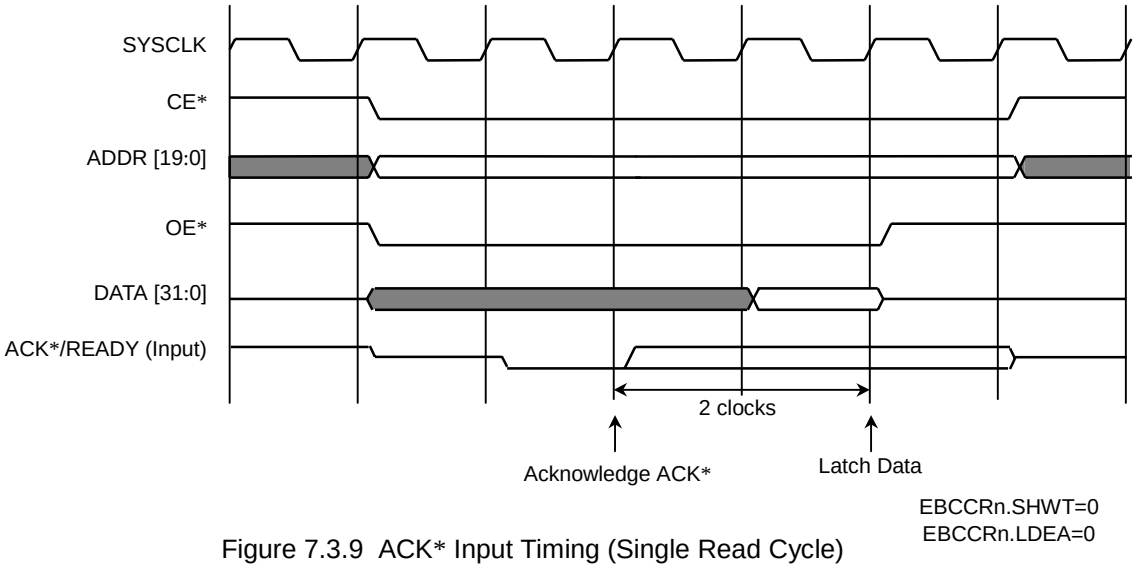


Figure 7.3.9 ACK* Input Timing (Single Read Cycle)

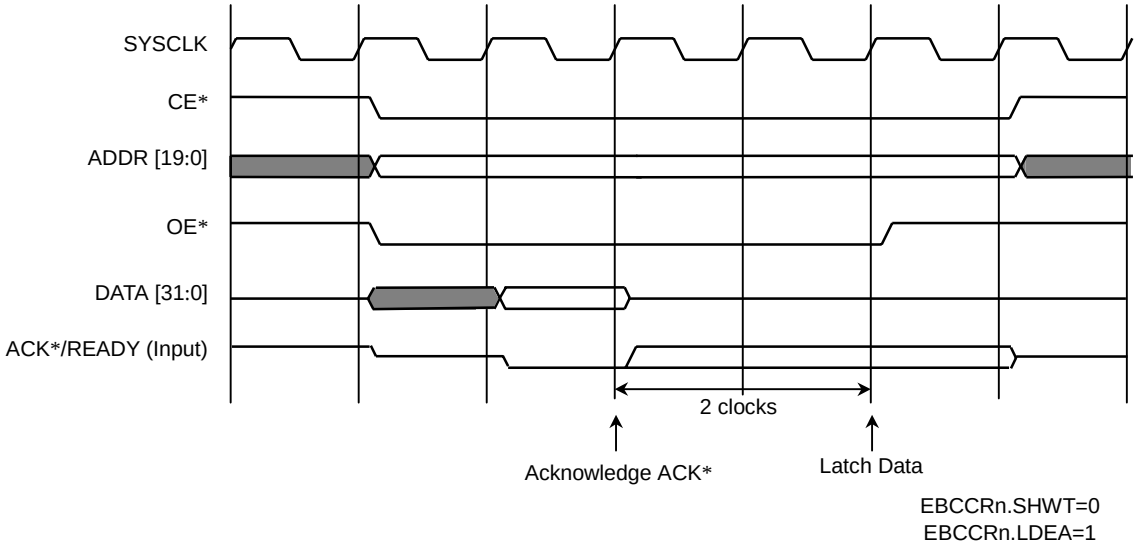


Figure 7.3.10 ACK* Input Timing (Single Read Cycle)

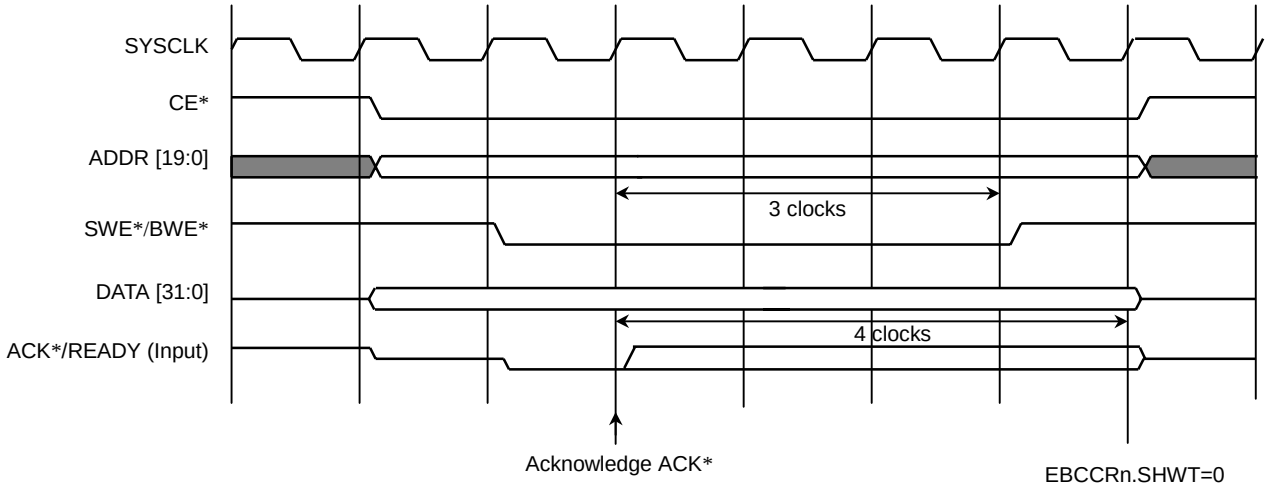


Figure 7.3.11 ACK* Input Timing (Single Write Cycle)

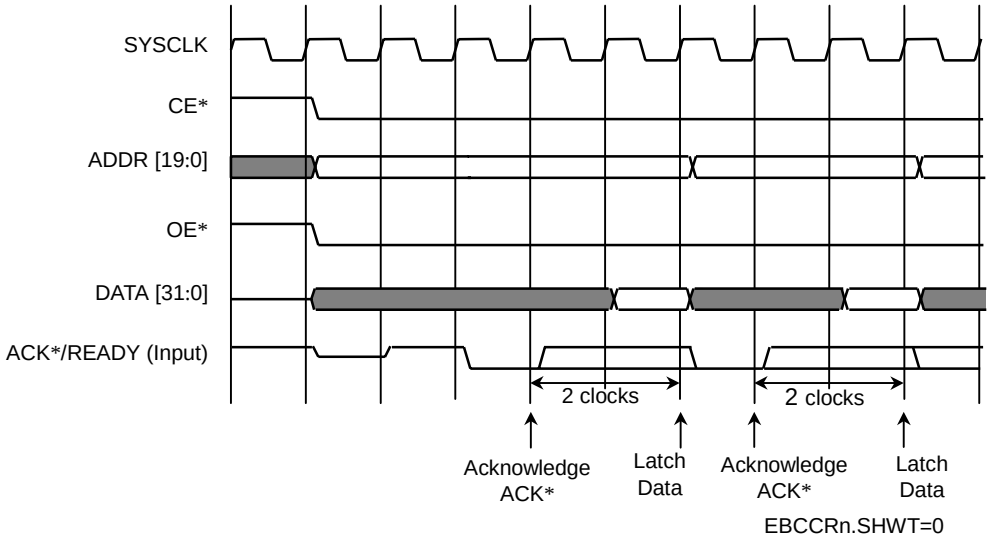


Figure 7.3.12 ACK* Input Timing (Burst Read Cycle)

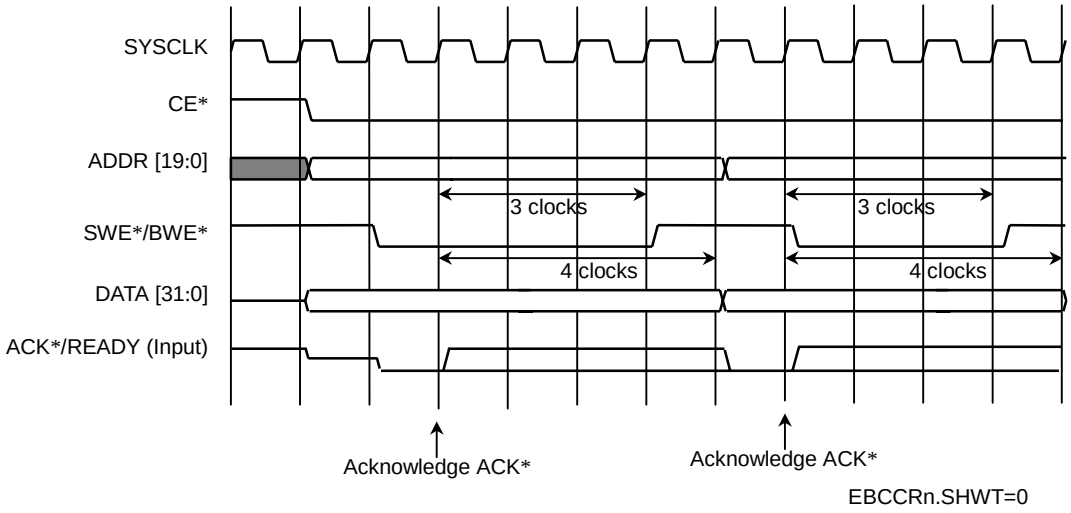


Figure 7.3.13 ACK* Input Timing (Burst Write Cycle)

7.3.7.5 Ready Input Timing

The ACK*/Ready pin is used as a Ready input when in the Ready mode. The Ready input timing is the same as the ACK* input timing explained in "7.3.7.4 ACK* Input Timing (External ACK Mode)" with the two following exceptions.

- Ready must be a High Active signal.
- When in the Ready mode, the Wait cycle count specified by EBCCRn.PWT:WT must be inserted in order to delay the Ready signal check (see "7.3.6.3 Ready Mode").

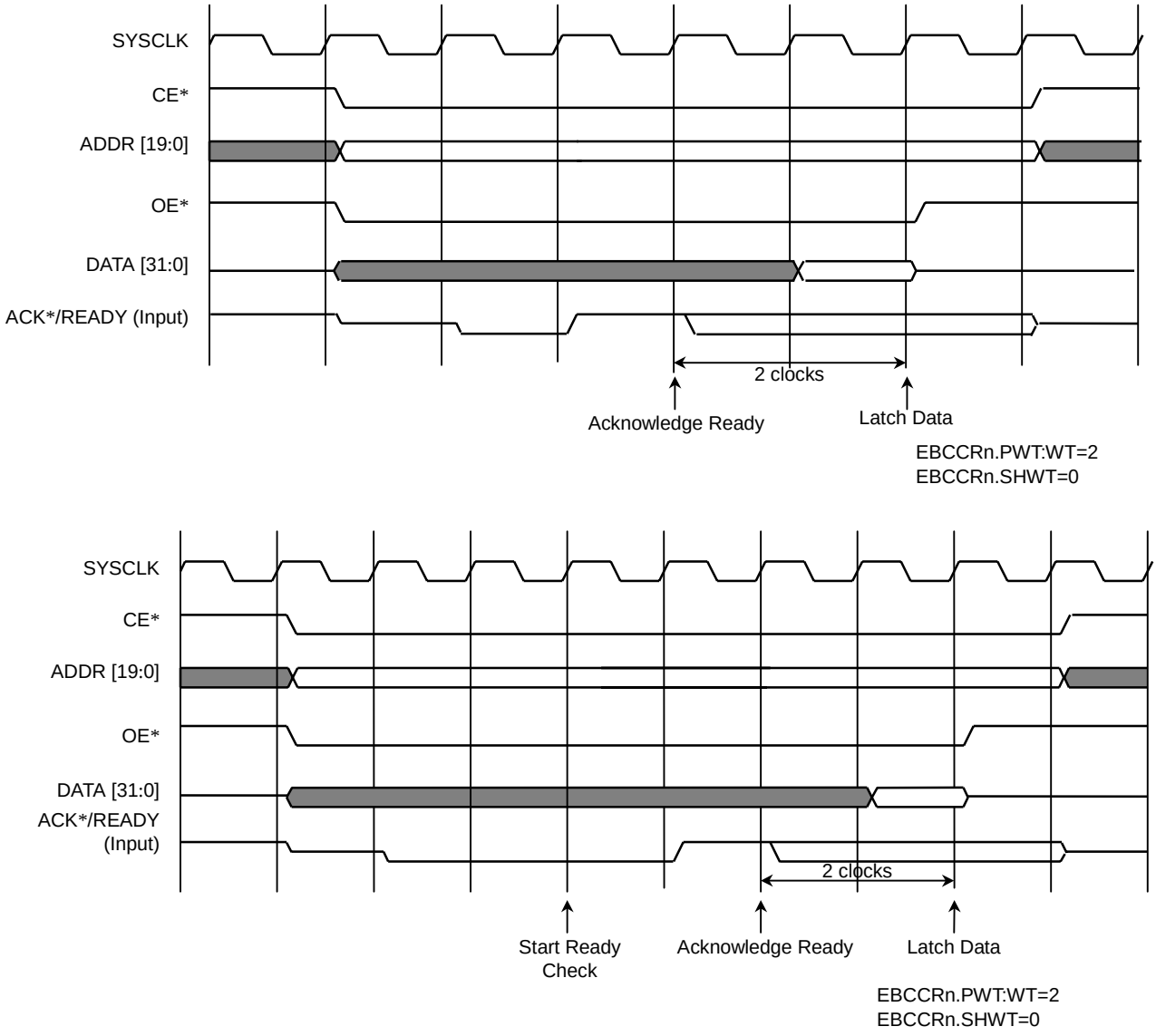


Figure 7.3.14 Ready Input Timing (Read Cycle)

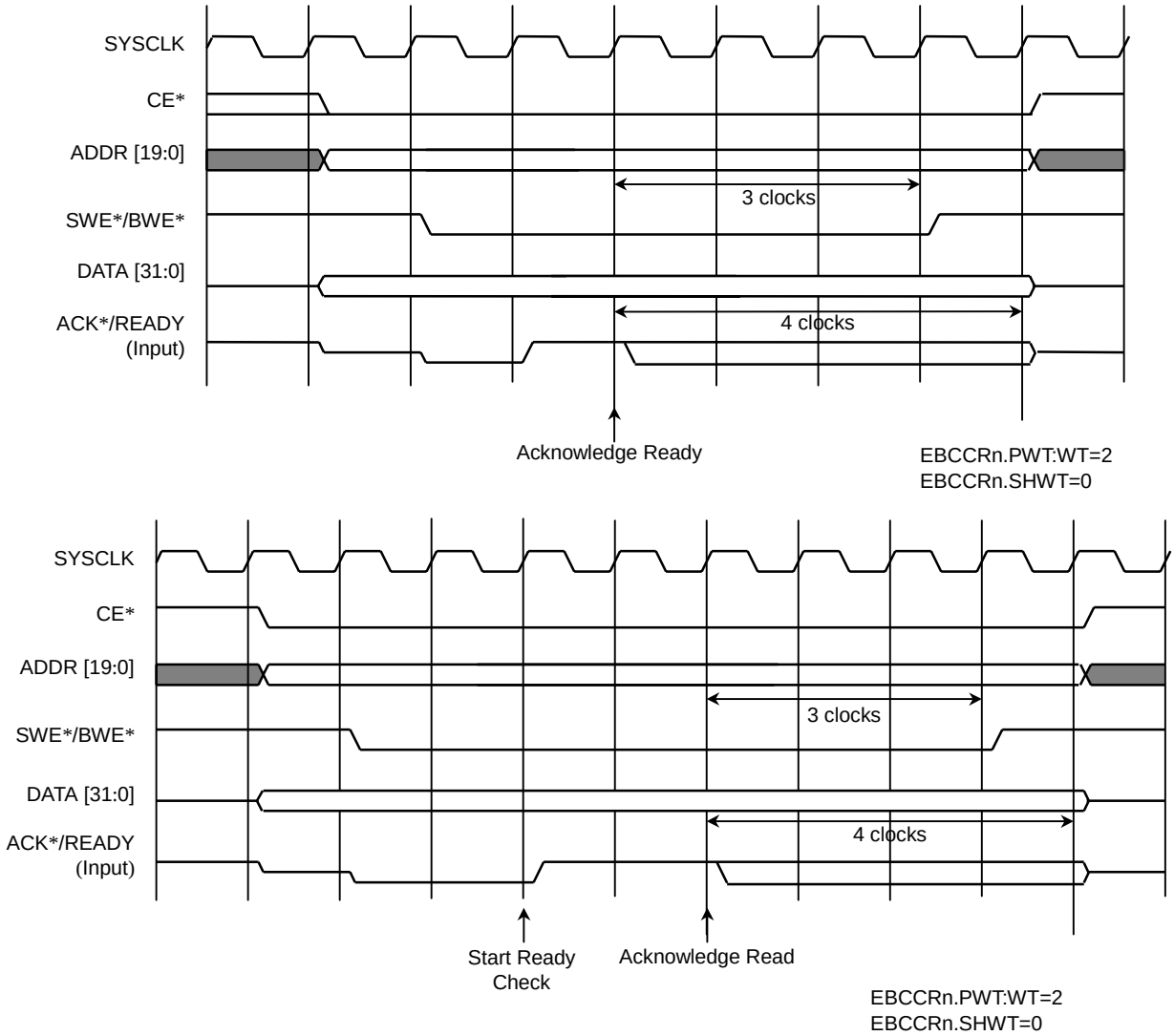


Figure 7.3.15 Ready Input Timing (Write Cycle)

7.3.8 Clock Options

External devices connected to the external bus can use the SYSCLK signal as the clock. The SYSCLK signal clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/1, 1/2, 1/3, 1/4. The ADDR[4:3] signal is used to set this frequency during reset, and the setting is reflected in the SYSCLK Division Ratio field (CCFG.SYSSP) of the Chip Configuration Register.

The operation reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel independent of the SYSCLK signal clock frequency: 1/1, 1/2, 1/3, 1/4. The external signal of the External Bus Controller operates synchronous to this operation clock. The Bus Speed field (EBCCRn.SP) of the External Bus Channel Control Register sets this frequency.

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

7.3.9 PCMCIA mode

The EBUSC Controller supports the following functions of the 16-Bit PC Card interface.

The EBUSC Controller supports 2 PCMCIA slots. Any of the eight EBUSC Controller channels can target the 2 PCMCIA slots. The use of channels 6 and 7 first for PCMCIA is recommended, since their normal CE_ signals are not pinned out and they cannot be used for any other function. For a channel to target a PCMCIA slot, it must first be in a PCMCIA mode. This is done by programming the PCM field to a non zero value. See the PCM field description for more details. Once a channel has been programmed to a PCMCIA mode, the PCS bit determines which slot is accessed via the two pairs of Card Enable signals.

Since pin multiplexing exists on the TX4925 device, the PCMCIA pin selection should be configured appropriately. Please refer to “3.3 Pin Multiplexing” for more information.

It should be noted that a normal dedicated CE_ for a channel used for PCMCIA access will continue to go active when that channel is accessed. It is up to the system designer to make sure that this does not cause any conflict on the external buses. The signals BWE*/BE*, OE*, and WE* also continue to cycle regardless of the PCMCIA mode. This was done for AC timing considerations.

OE* and WE* are forced inactive during IO access.

CARDIORD* and CARDIOWR* are multiplexed on BWE*/BE*[3:2] pins respectively in a time multiplexed manner on the TX4925.

7.3.9.1 PCMCIA Mode Selects

A channel that is allocated to be used as a PCMCIA control channel can be put into one of four access modes. The first is the Attribute Memory Access mode. This mode is typically used at power on only to determine the type of PCMCIA card installed. The second is the Common Memory Access Mode, used for cards that are of the memory type. The third is IO Access mode, used for cards that support IO. A single channel can be used initially for Attribute access to determine the card types and then that channel can be reallocated for one of the cards.

In the case of properties access, memory access, and I/O access is determined by differences in the signals CARDREG*, CARDIORD*, CARDIOWR*, the maximum address space, and width of the access. Consult the PCMCIA 16-Bit PC Card specification for details.

7.3.9.2 PCMCIA Slot Selection

Slot 1 is accessed by programming the PCS bit to 0 and the CARD1CSH*/CARD1CSL* signals are active. If the PCS bit is set to 1 then Slot 2 is accessed and the CARD2CSH*/CARD2CSL* signals are active.

7.3.9.3 PCMCIA Channel Programming Requirements

Because of the multiple modes possible on a given EBUSC Channel the following restrictions apply when a channel is used for PCMCIA access.

Page Mode is not allowed on a PCMCIA enabled Channel. Undetermined results will occur if Page Mode is active.

External Ack Mode is not allowed on a PCMCIA enabled Channel. Undetermined results will occur if External Ack Mode is active.

A channel that is used for PCMCIA access must be programmed to have a bus size of 16 or 8 bits. Addressing will not be correct if using a 32 bit bus size.

To meet the setup and hold timing of the PCMCIA bus the SHWT mode must be used. Depending on the access speed of the card, the number of SHWT states and the speed of the channel will need to be programmed to meet the timing requirements.

PWT and WT can be used to extend the OE*, WE*, CARDIORD*, and CARDIOWR* active timing. See the description of the PWT and WT fields for details.

The EBUSC Controller can be used to implement the WAIT* function of the PCMCIA specification. To do so PCMCIA channel must be programmed with the RDY bit set. See RDY mode description for more details. Wait states will then be inserted externally until the WAIT* signal from the PC card goes inactive and informing the EBUSC controller that it is READY. The EBUSC controller supports a unique WAIT* signal for both slots. See device pin multiplexing for details on support. The PWT/WT counter must be used programmed to account for the delay in WAIT* valid from OE*, WE*, CARDIORD*, CARDIOWR*. Otherwise the high time of WAIT* during this delay may cause the EBUSC controller to terminate the cycle early. The delay value should take into account that the WAIT* signals are synchronized to the full speed clock regardless of the speed of the channel.

7.3.9.4 PCMCIA Addressing and UAE

The PCMCIA specification has a 26 bit address bus. Since the external address bus of the device is only 20 bits, the top 6 bits must be latched using the UAE signal. See the description on UAE for more details.

The PCMCIA specification always uses a byte addressing scheme, where A0 is a “don’t care” for “Word” and “Odd-Byte-Only” accesses. Normally the EBUSC controller outputs an address that allows for addressing bytes, half words, or words depending on the programmed bus size. For accessing data sizes smaller than the programmed bus size the BE*/BWE* signals are used in normal operations. In the case of a channel being used for PCMCIA access the addressing will always be for bytes, regardless of the bus size being 16 or 8 bit.

7.3.9.5 PCMCIA IOIS16* Signal

The IOIS16* signal is not supported. This means that the size of IO accesses needs to be known ahead of time and are not dynamically sized.

7.3.9.6 PCMCIA Cycle Types

Burst Cycles are not allowed on a PCMCIA enabled Channel. This is due to the fact that the EBUSC Controller does not toggle all signals between the individual accesses for burst accesses. The PCMCIA specification requires that all signals toggle per each access. In addition programming requirements for a channel supporting PCMCIA require SHWT mode. Burst accesses are not allowed with SHWT enabled.

Table 7.3.8 Access Mapping (PCMCIA in 16-bit Channel and Little Endian Mode)

Access	Access Address	ADDR[1:0]	Port Size	CARDnCSH*, CARDnCSL*	DATA[15:8]	DATA[7:0]
Word(1/2)	00	00	16-bit	LL	R[15:8]	R[7:0]
Word{2/2}		10	16-bit	LL	R[31:24]	R[23:16]
Triple byte(1/2)	01	00	16-bit	LH	R[15:8]	-
Triple byte(2/2)		10	16-bit	LL	R[31:24]	R[23:16]
Triple byte(1/2)	00	00	16-bit	LL	R[15:8]	R[7:0]
Triple byte(2/2)		10	16-bit	HL	-	R[23:16]
Half-Word	10	10	16-bit	LL	R[15:8]	R[7:0]
Half-Word	00	00	16-bit	LL	R[15:8]	R[7:0]
Byte	11	10	16-bit	LH	R[7:0]	-
Byte	10	10	16-bit	HL	-	R[7:0]
Byte	01	00	16-bit	LH	R[7:0]	-
Byte	00	00	16-bit	HL	-	R[7:0]

Note: Word access and triple byte access yield 2 separate external cycles.

Table 7.3.9 Access Mapping (PCMCIA in 8-bit Channel and Little Endian Mode)

Access	Access Address	ADDR[1:0]	Port Size	CARDnCSH*, CARDnCSL*	DATA[15:8]	DATA[7:0]
Word(1/4)	00	00	8-bit	HL	-	R[7:0]
Word{2/4}		01	8-bit	HL	-	R[15:8]
Word(3/4)		10	8-bit	HL	-	R[23:16]
Word{4/4}		11	8-bit	HL	-	R[31:24]
Triple byte(1/3)	01	01	8-bit	HL	-	R[15:8]
Triple byte(2/3)		10	8-bit	HL	-	R[23:16]
Triple byte(3/3)		11	8-bit	HL	-	R[31:24]
Triple byte(1/3)	00	00	8-bit	HL	-	R[7:0]
Triple byte(2/3)		01	8-bit	HL	-	R[15:8]
Triple byte(3/3)		10	8-bit	HL	-	R[23:16]
Half-Word(1/2)	10	10	8-bit	HL	-	R[7:0]
Half-Word(2/2)		11	8-bit	HL	-	R[15:8]
Half-Word(1/2)	00	00	8-bit	HL	-	R[7:0]
Half-Word(2/2)		01	8-bit	HL	-	R[15:8]
Byte	11	11	8-bit	HL	-	R[7:0]
Byte	10	10	8-bit	HL	-	R[7:0]
Byte	01	01	8-bit	HL	-	R[7:0]
Byte	00	00	8-bit	HL	-	R[7:0]

Note: Word access yields 4 separate external cycles.
Triple byte access yields 4 separate external cycles.
Half-word access yields 4 separate external cycles.

Table 7.3.10 Access Mapping (PCMCIA in 16-bit Channel and Big Endian Mode)

Access	Access Address	ADDR[1:0]	Port Size	CARDnCSH*, CARDnCSL*	DATA[15:8]	DATA[7:0]
Word(1/2)	00	00	16-bit	LL	R[31:24]	R[23:16]
Word{2/2}		10	16-bit	LL	R[15:8]	R[7:0]
Triple byte(1/2)	01	00	16-bit	LL	R[31:24]	R[23:16]
Triple byte(2/2)		10	16-bit	LH	R[15:8]	-
Triple byte(1/2)	00	00	16-bit	HL	-	R[23:16]
Triple byte(2/2)		10	16-bit	LL	R[15:8]	R[7:0]
Half-Word	10	10	16-bit	LL	R[15:8]	R[7:0]
Half-Word	00	00	16-bit	LL	R[15:8]	R[7:0]
Byte	11	10	16-bit	LH	-	R[7:0]
Byte	10	10	16-bit	HL	R[7:0]	-
Byte	01	00	16-bit	LH	-	R[7:0]
Byte	00	00	16-bit	HL	R[7:0]	-

Note: Word access and triple byte access yield 2 separate external cycles.

Table 7.3.11 Access Mapping (PCMCIA in 8-bit Channel and Big Endian Mode)

Access	Access Address	ADDR[1:0]	Port Size	CARDnCSH*, CARDnCSL*	DATA[15:8]	DATA[7:0]
Word(1/4)	00	00	8-bit	HL	-	R[31:24]
Word{2/4}		01	8-bit	HL	-	R[23:16]
Word(3/4)		10	8-bit	HL	-	R[15:8]
Word{4/4}		11	8-bit	HL	-	R[7:0]
Triple byte(1/3)	01	01	8-bit	HL	-	R[23:16]
Triple byte(2/3)		10	8-bit	HL	-	R[15:8]
Triple byte(3/3)		11	8-bit	HL	-	R[7:0]
Triple byte(1/3)	00	00	8-bit	HL	-	R[31:24]
Triple byte(2/3)		01	8-bit	HL	-	R[23:16]
Triple byte(3/3)		10	8-bit	HL	-	R[15:8]
Half-Word(1/2)	10	10	8-bit	HL	-	R[15:8]
Half-Word(2/2)		11	8-bit	HL	-	R[7:0]
Half-Word(1/2)	00	00	8-bit	HL	-	R[15:8]
Half-Word(2/2)		01	8-bit	HL	-	R[7:0]
Byte	11	11	8-bit	HL	-	R[7:0]
Byte	10	10	8-bit	HL	-	R[7:0]
Byte	01	01	8-bit	HL	-	R[7:0]
Byte	00	00	8-bit	HL	-	R[7:0]

Note: Word access yields 4 separate external cycles.
Triple byte access yields 4 separate external cycles.
Half-word access yields 4 separate external cycles.

7.3.9.7 PCMCIA INT# Support

The TX4925 does not have the PCMCIA INT* signal; it is up to the system designer to implement it external to the device, if necessary.

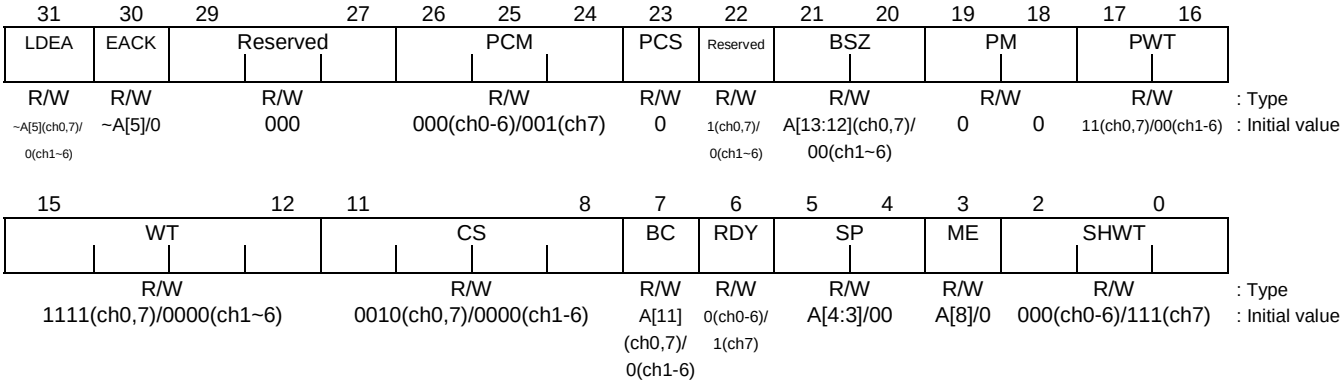
7.4 Register

Table 7.4.1 External Bus Controller (EBUSC) Registers

Reference	Offset Address	Bit Width	Register Symbol	Register Name
7.4.1	0x9000	32	EBCCR0	External Bus Channel Control Register 0
7.4.2	0x9004	32	EBBAR0	External Bus Base Address Register 0
7.4.1	0x9008	32	EBCCR1	External Bus Channel Control Register 1
7.4.2	0x900c	32	EBBAR1	External Bus Base Address Register 1
7.4.1	0x9010	32	EBCCR2	External Bus Channel Control Register 2
7.4.2	0x9014	32	EBBAR2	External Bus Base Address Register 2
7.4.1	0x9018	32	EBCCR3	External Bus Channel Control Register 3
7.4.2	0x901c	32	EBBAR3	External Bus Base Address Register 3
7.4.1	0x9020	32	EBCCR4	External Bus Channel Control Register 4
7.4.2	0x9024	32	EBBAR4	External Bus Base Address Register 4
7.4.1	0x9028	32	EBCCR5	External Bus Channel Control Register 5
7.4.2	0x902c	32	EBBAR5	External Bus Base Address Register 5
7.4.1	0x9030	32	EBCCR6	External Bus Channel Control Register 6
7.4.2	0x9034	32	EBBAR6	External Bus Base Address Register 6
7.4.1	0x9038	32	EBCCR7	External Bus Channel Control Register 7
7.4.2	0x903c	32	EBBAR7	External Bus Base Address Register 7

7.4.1 External Bus Channel Control Register (EBCCRn) 0x9000 (ch. 0), 0x9008 (ch. 1) 0x9010 (ch. 2), 0x9018 (ch. 3) 0x9020 (ch. 4), 0x9028 (ch. 5) 0x9030 (ch. 6), 0x9038 (ch. 7)

Channel 0 and 7 can be used as Boot memory. Therefore, the default is set by the Boot signal (see "7.3.2 Global/Boot-up Options"). Channels 1 - 7 have the same register configuration as Channel 0, but they have different defaults than Channel 0.



Only in the case of Channel 0 is fields with different defaults in the "Channel 0/Other channel" state. D[ ] represents the corresponding Data[ ] signal value when the RESET* signal is deasserted. A[ ] represents the corresponding ADDR[ ] signal value when the RESET* signal is deasserted.

Bits	Mnemonic	Field Name	Description
31	LDEA	Latch Data at External ACK*	Latch Data at External ACK* (Initial value: ~A[5](ch0,7)/0(ch1~6), R/W) Specifies the data latched timing in the external ACK* input mode. 0 : Data is latched for External ACK* Input mode Reads at OE* active. 1 : Data is latched for External ACK* Input mode Reads at ACK* active.
30	EACK	ACK* Input Active	ACK* Input Active (Initial value: ~A[5]/0, R/W) Enable ACK* Input mode. 0 : ACK* Input mode is disabled. 1 : ACK* Input mode is enabled.
29 : 27	—	Reserved	Note: These bits are always set to "0" (Initial value: 000, R/W).

Figure 7.4.1 External Bus Channel Control Register (1/3)

Bits	Mnemonic	Field Name	Description
26:24	PCM	PCMCIA 16-bit PC Card Mode	PCMCIA 16-bit PC Card Mode (Initial value: 000(ch0-6)/001(ch7), R/W) Specifies the PCMCIA mode. 000 : PCMCIA Disabled for Channel. 001 : PCMCIA Common Memory Access Enabled for Channel. 010 : PCMCIA IO Access Enabled for Channel. 011 : PCMCIA Attribute Memory Access Enabled for Channel. 100, 101, 110, 111: Reserved
23	PCS	PCMCIA Slot Selection	PCMCIA Slot Selection (Initial value: 0, R/W) This bit select PCMCIA slot. 0 : Channel PCMCIA access is for Slot1. 1 : Channel PCMCIA access is for Slot2.
22	—	Reserved	Note: This bit is always set “1” (Initial value: 1(ch0,7)/0(ch1-6), R/W). The value of Boot signal ADDR[2] is set in the default value for bit 22 of Channel 0 and Channel 7. However, Boot signal ADDR[2] must be “1”. If the default value for bit 22 of Cannel 0 and Channel 7 is “0”, please confirm that Boot signal ADDR[2] is not “0”.
21:20	BSZ	Bus Width	External Bus Control Bus Size (Initial value: A[13:12](ch0,7)/00(ch1~6), R/W) Specifies the memory bus width. 00: Reserved 10: 16-bit width 01: 32-bit width 11: 8-bit width Note: ADDR[13:12] is set to Channel 0 and 7 as the default.
19:18	PM	Page Mode Page Size	External Bus Control Page Mode Page Size (Initial value: 00, R/W) Specifies the Page mode (Page mode memory support) use and page size. 00: Normal mode 01: 4-page mode 10: 8-page mode 11: 16-page mode
17:16	PWT	Page Mode Wait time	External Bus Control Page Mode Wait Time (Initial value: 11(ch0,7)/00(ch1-6), R/W) Specifies the wait cycle count during Burst access when in the Page mode. 00: 0 wait cycles 10: 2 wait cycles 01: 1 wait cycle 11: 3 wait cycles Specifies a wait cycle count from 0 to 63 that matches WT when in the Normal mode or Ready mode. (See the WT item.)
15:12	WT	Normal Mode Wait Time	External Bus Control Normal Mode Wait Time (Initial value: 1111(ch0,7)/0000(ch1~6), R/W) Specifies the wait cycle count in the first cycle of a Single Cycle or Burst access. Specifies the following wait cycle count when in the Page mode. 0000: 0 wait cycles 0100: 4 wait cycles 1000: 8 wait cycles 1100: 12 wait cycles 0001: 1 wait cycle 0101: 5 wait cycles 1001: 9 wait cycles 1101: 13 wait cycles 0010: 2 wait cycles 0110: 6 wait cycles 1010: 10 wait cycles 1110: 14 wait cycles 0011: 3 wait cycles 0111: 7 wait cycles 1011: 11 wait cycles 1111: 15 wait cycles Specifies a wait cycle count from 0 to 63 that matches PWT when in a mode other than the Page mode. PWT[1:0]: WT[3:0] 000000: 0 wait cycles 010000: 16 wait cycles 110000: 48 wait cycles 000001: 1 wait cycle 010001: 17 wait cycles 110001: 49 wait cycles : : : 001110: 14 wait cycles 011110: 30 wait cycles 111110: 62 wait cycles 001111: 15 wait cycles 011111: 31 wait cycles 111111: 63 wait cycles Note 1: Set the WT wait cycle count to a value greater than the PWT Wait cycle count when in the Page mode.

Figure 7.4.1 External Bus Channel Control Register (2/3)

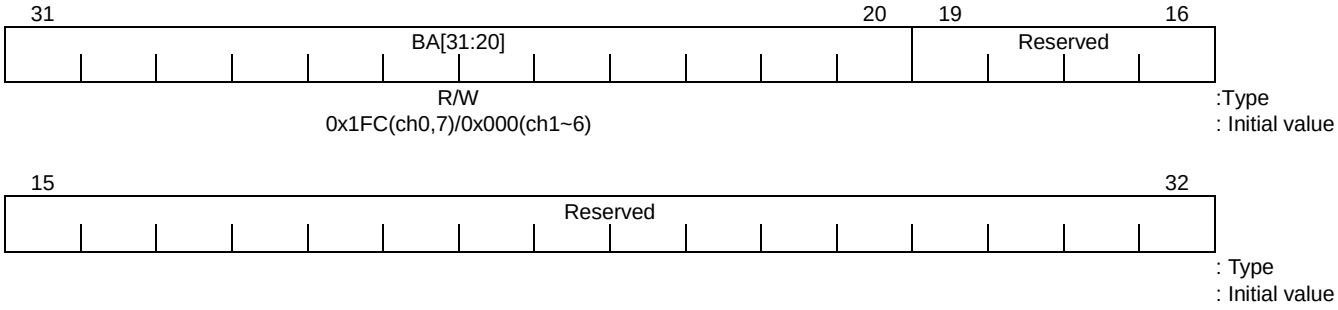
Bits	Mnemonic	Field Name	Description
11:8	CS	Channel Size	External Bus Control Channel Size (Initial value: 0010(ch0,7)/0000(ch1-6), R/W) Specifies the channel memory size. 0000: 1 MB 0101: 32 MB *1010: 1 GB 0001: 2 MB 0110: 64 MB 1011-1111: Reserved 0010: 4 MB 0111: 128 MB 0011: 8 MB 1000: 256 MB 0100: 16 MB *1001: 512 MB * The channel memory size can be set up to 512 MB when the memory bus width is 16 bits, or up to 256 MB when the memory bus width is 8 bits. No size larger than this can be set.
7	BC	Byte Control	External Bus Byte Control (Initial value: A[11](ch0,7)/0(ch1-6), R/W) Specifies whether to use the BWE*[3:0] signal as an asserted Byte Write Enable signal (BWE*[3:0]) only during a Write cycle, or to use it as an asserted Byte Enable signal (BE*[3:0]) that is asserted during both Read and Write cycles. 0: Byte Enable (BE *[3:0]) 1: Byte Write Enable (BWE*[3:0]) Note: ADDR[11] is set to Channel 0 and 7 as the default.
6	RDY	Ready Input Mode	External Bus Control Ready Input Mode (Initial value: 0(ch0-6)/1(ch7), R/W) Specifies whether to use the Ready mode. 0: Disable the Ready mode. 1: Enable the Ready mode. Note: The Ready mode cannot be used when the Page mode is selected.
5:4	SP	Bus Speed	External Bus Control Bus Speed (Initial value: A[4:3]/00, R/W) Specifies the External Bus speed. 00: 1/4 speed (1/4 of the GBUSCLK frequency) 01: 1/3 speed (1/3 of the GBUSCLK frequency) 10: 1/2 speed (1/2 of the GBUSCLK frequency) 11: Full speed (same frequency as GBUSCLK) Note: ADDR[4:3] is set to Channel 0 as the default.
3	ME	Master Enable	External Bus Control Master Enable (Initial value: A[8]/0, R/W) Enables a channel. 0: Disable channel 1: Enable channel Note: EBCCR0.ME bit is set when ADDR[8:6] equal to “1xx” as the default. EBCCR7.ME bit is set when ADDR[8:6] equal to “010” as the default. The default value for the ME bit of Channel 7 is “1” when Boot signal ADDR[8:6] is 010b. The value 010b cannot be used as Boot signal ADDR[8:6]. If the default value of the ME bit for Channel 7 is “1”, please confirm that Boot signal ADDR[8:6] is not 010b.
2:0	SHWT	Set Up/Hold Wait Time	External Bus Control Setup/Hold Wait Time (Initial value: 000(ch0-6)/111(ch7), R/W) Specifies the wait count when switching between the Address and Chip Enable signal, or the Chip Enable Signal and Write Enable/Output Enable signal. * 000: Disable 100: 4 wait cycles 001: 1wait cycle 101: 5 wait cycles 010: 2 wait cycles 110: 6 wait cycles 011: 3 wait cycles 111: 7 wait cycles * Set this bit field to “0” when using it in the Page mode or when performing Burst access.

Figure 7.4.1 External Bus Channel Control Register (3/3)

7.4.2 External Bus Base Address Register (EBBARn)

0x9000 (ch. 0), 0x9008 (ch. 1)
0x9010 (ch. 2), 0x9018 (ch. 3)
0x9020 (ch. 4), 0x9028 (ch. 5)
0x9030 (ch. 6), 0x9038 (ch. 7)

Channel 0 can be used as Boot memory. Therefore, the default is set by the Boot signal (see "7.3.2 Global/Boot-up Options"). Channels 1 - 7 have the same register configuration as Channel 0, but they have different defaults than Channel 0.



Only in the case of Channel 0 are fields with different defaults in the "Channel 0/Other channel" state.

Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Base Address	External Bus Control Base Address (Initial value: 0x1FC/0x000, R/W) A physical address is used to specify the base address. The upper 12 bits [31:20] of the physical address are compared to the value of this field.
19:0	—	Reserved	—

Figure 7.4.2 External Bus Base Address Register

7.5 Timing Diagrams

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

(1) The clock frequency of the SYSCLK signal can be set to one of the following divisions of the internal bus clock (GBUSCLK): 1/1, 1/2, 1/3, or 1/4. Also, the operating reference clock frequency can be set to one of the following divisions of the internal bus clock (GBUSCLK) for each channel: 1/1, 1/2, 1/3, or 1/4. (See 7.3.8.) The timing diagrams indicate the SYSCLK signal clock frequency and channel operating reference clock frequency as being equivalent.

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

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

(4) The timing diagrams display each clock cycle currently being accessed using the symbols described in the following table. (n=1, 2, 3, …)

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

(5) Shaded areas (▒▒▒) in the diagrams are undefined values.

7.5.1 UAE Signal

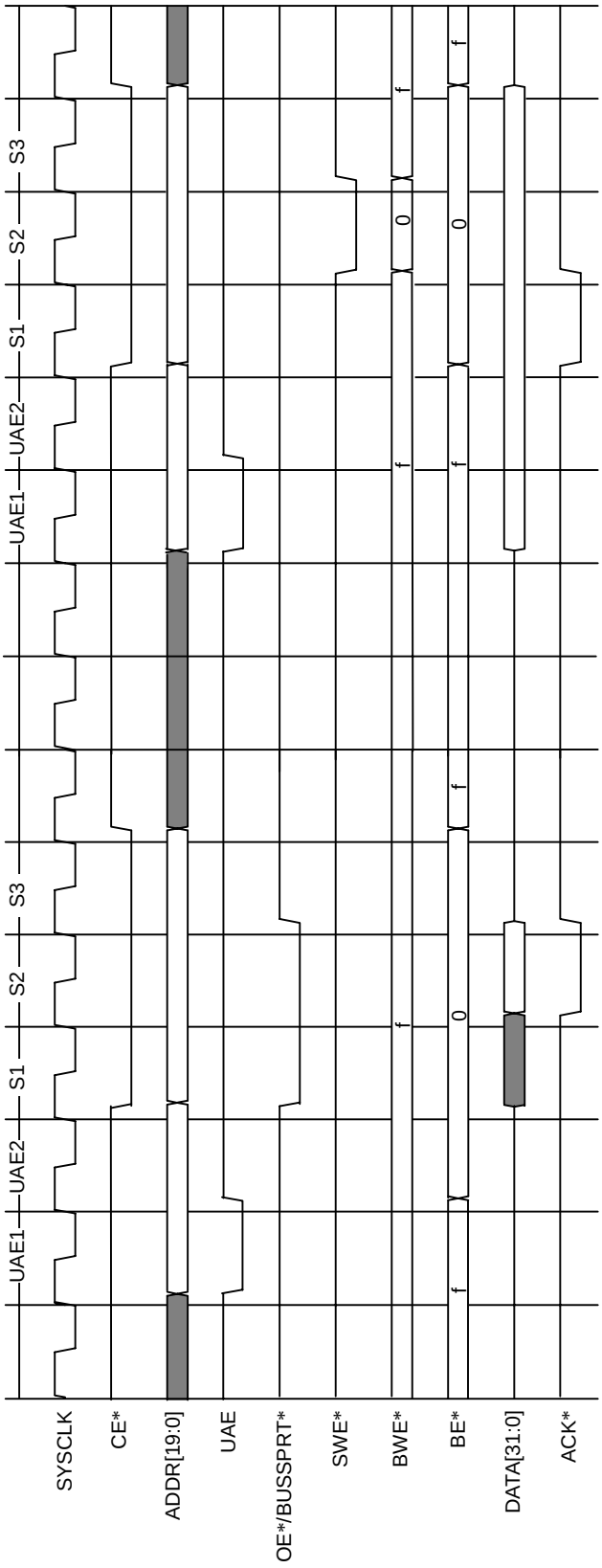


Figure 7.5.1 UAE Signal (CCFG.ACEHOLD=1, PWT: WT=0, SHWT=0, Normal)

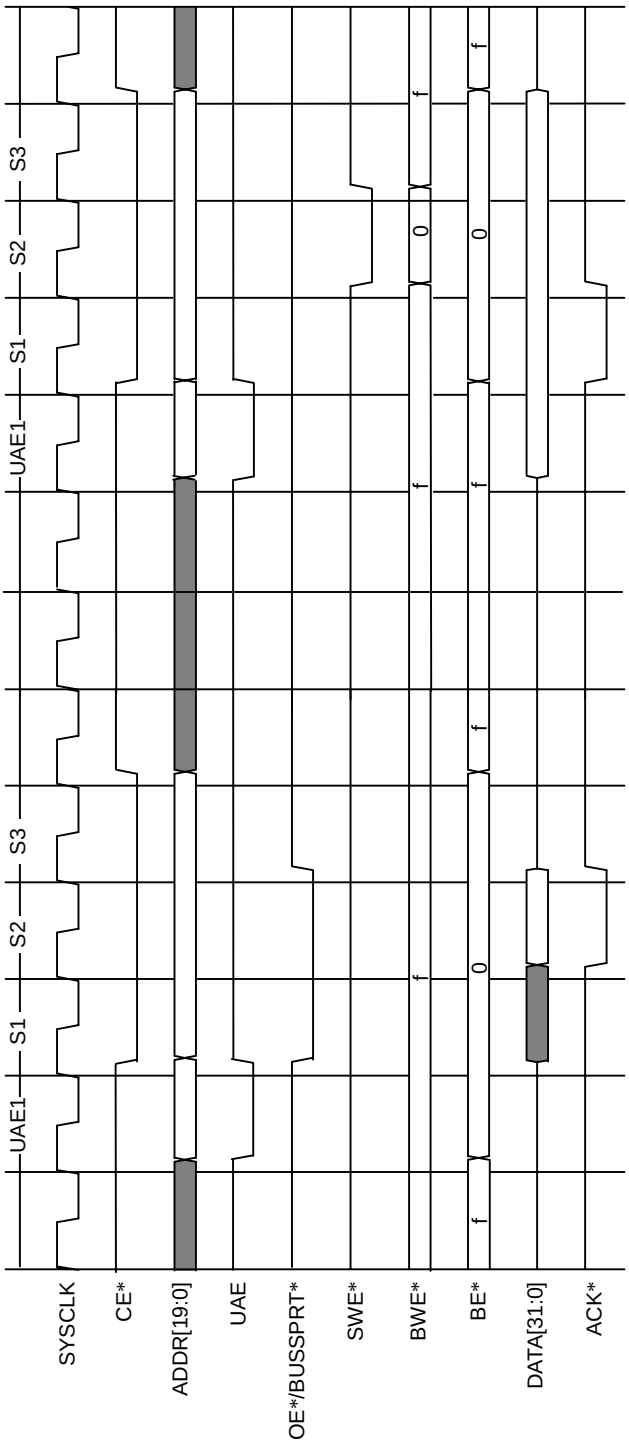


Figure 7.5.2 UAE Signal (CCFG.ACEHOLD=0, PWT: WT=0, SHWT=0, Normal)

7.5.2 Normal Mode Access (Single, 32-bit Bus)

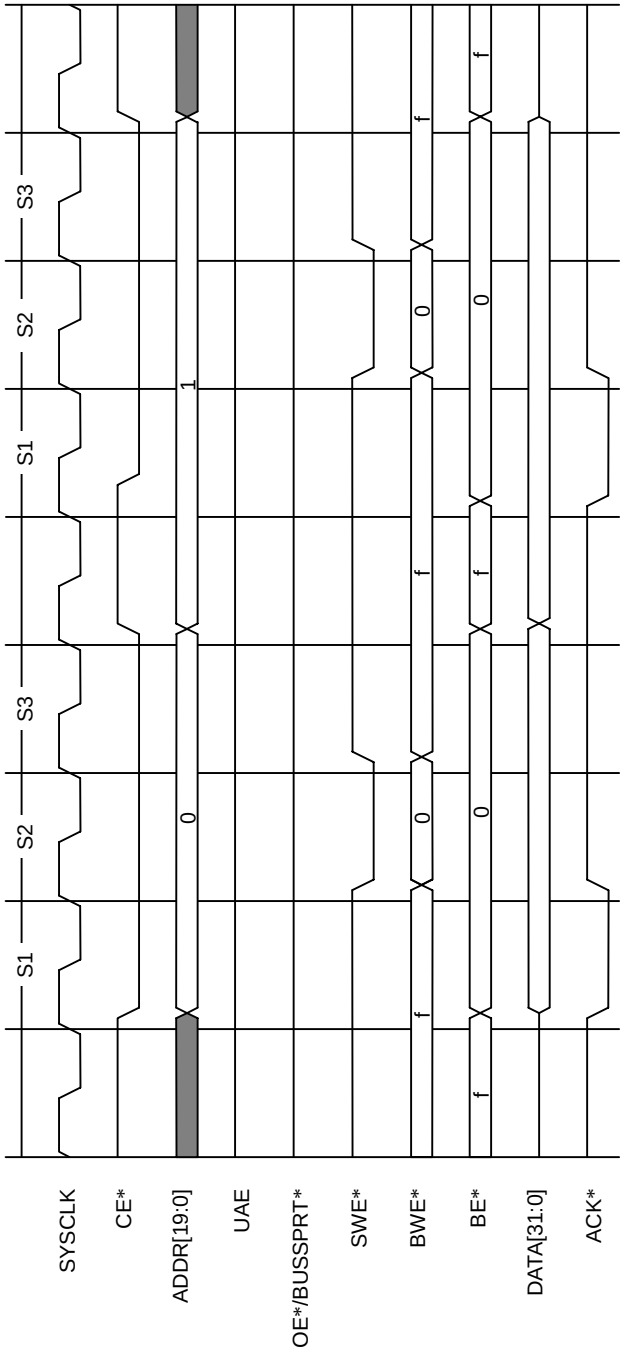


Figure 7.5.3 Double-word Single Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)

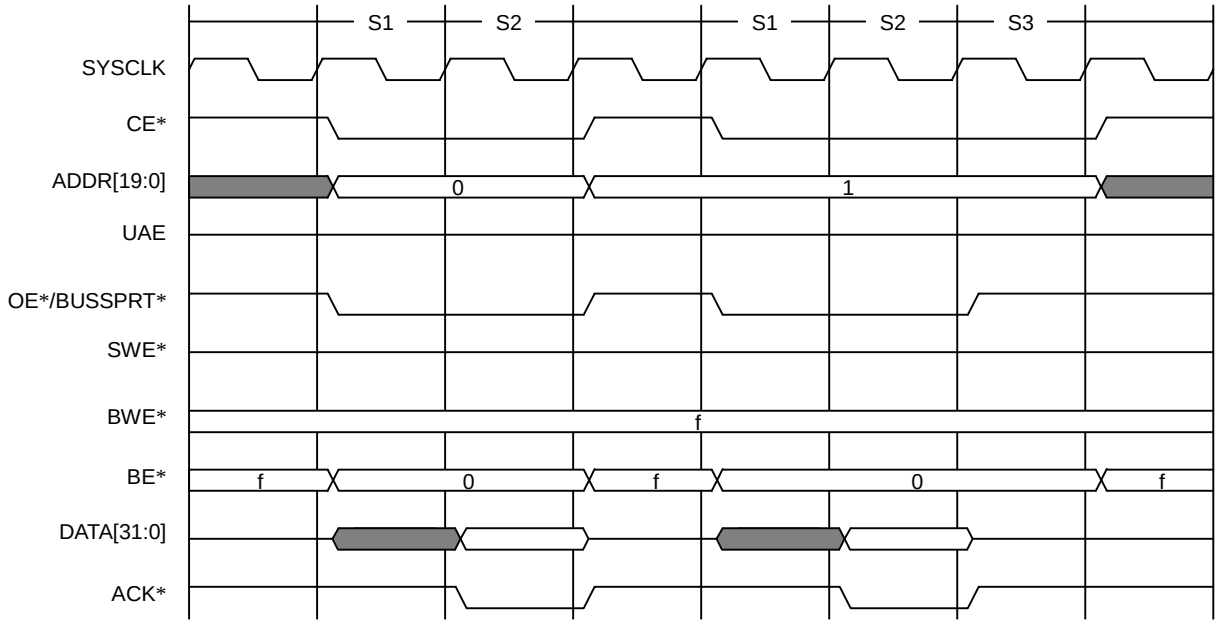


Figure 7.5.4 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

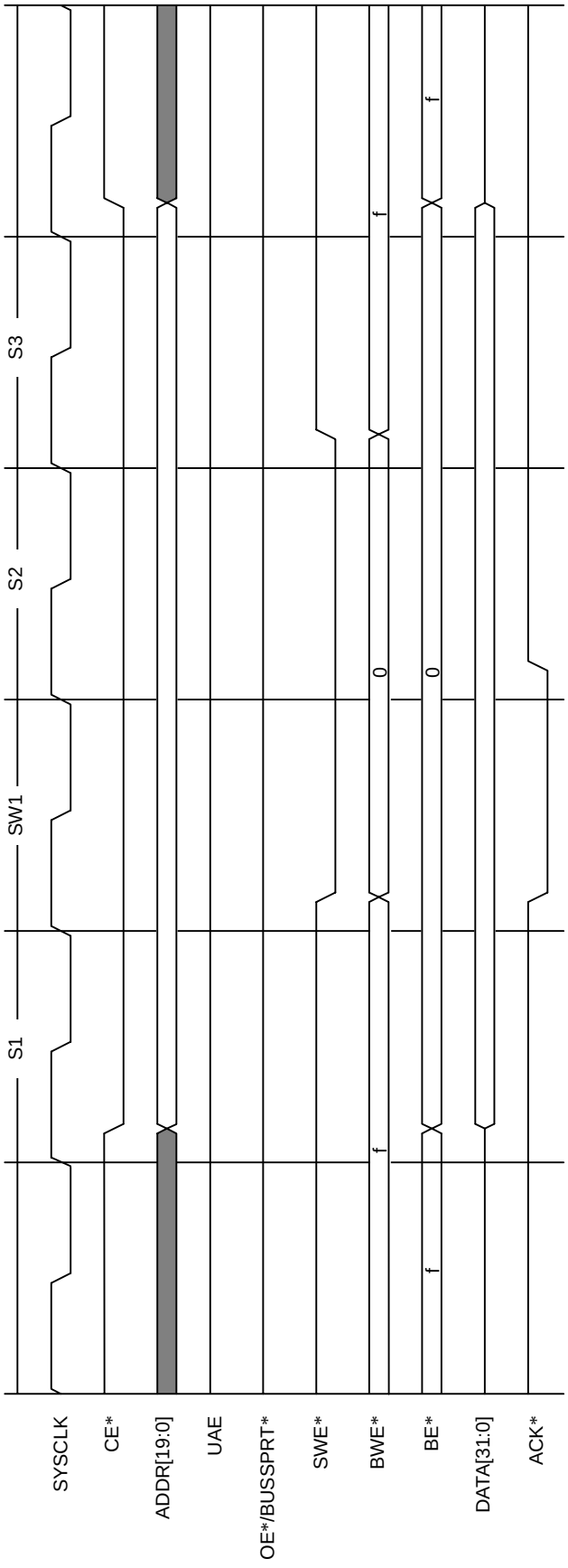


Figure 7.5.5 1-word Single Write (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

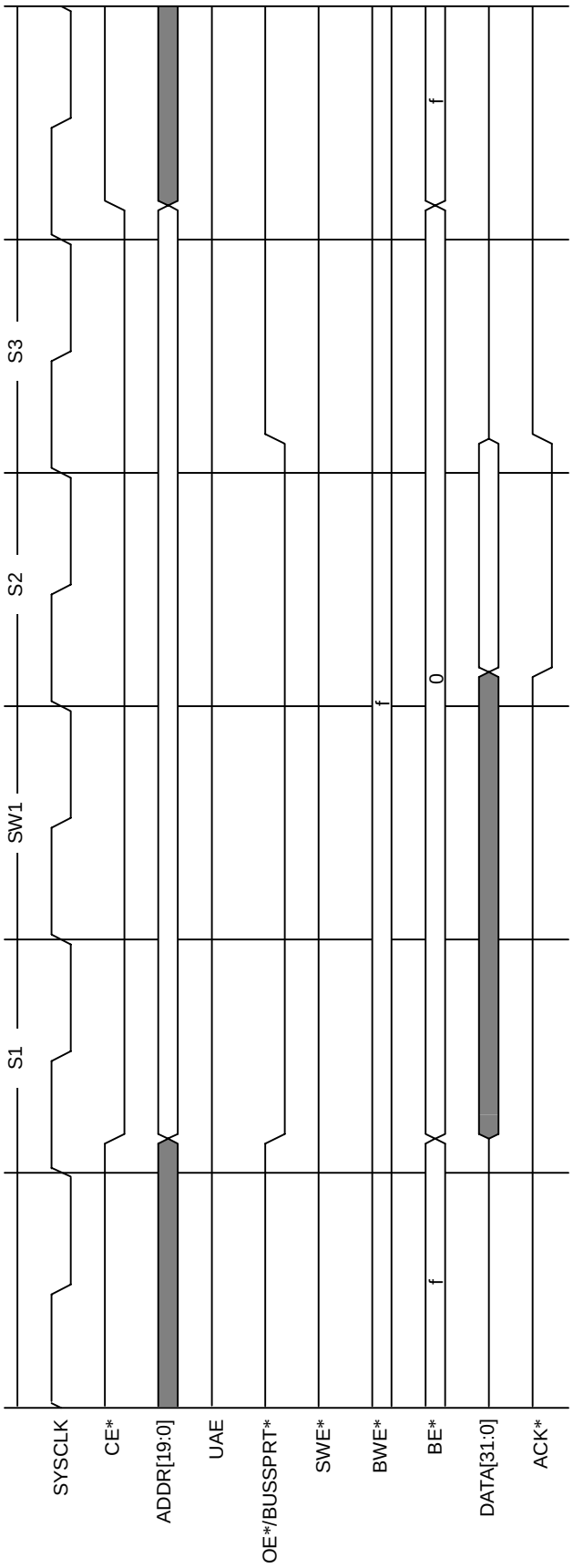


Figure 7.5.6 1-word Single Read (PWT: WT=0, SHWT=0, Normal, 32-bit Bus)

7.5.3 Normal Mode Access (Burst, 32-bit Bus)

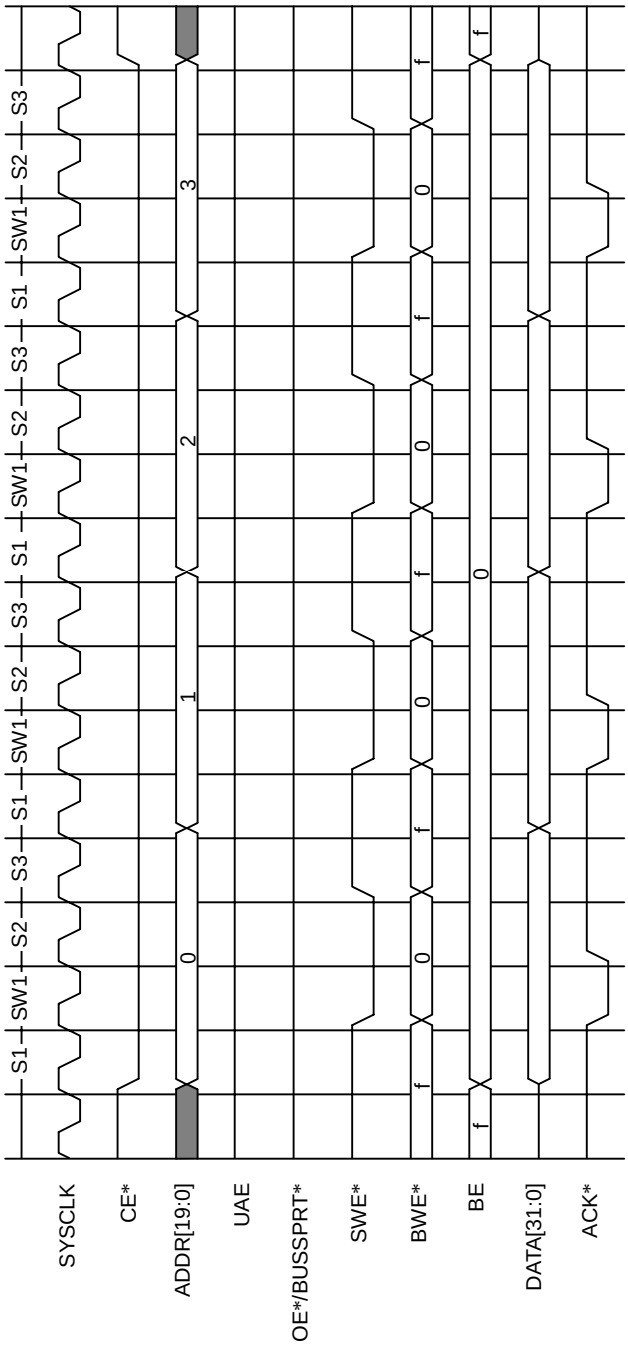


Figure 7.5.7 4-word Burst Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)

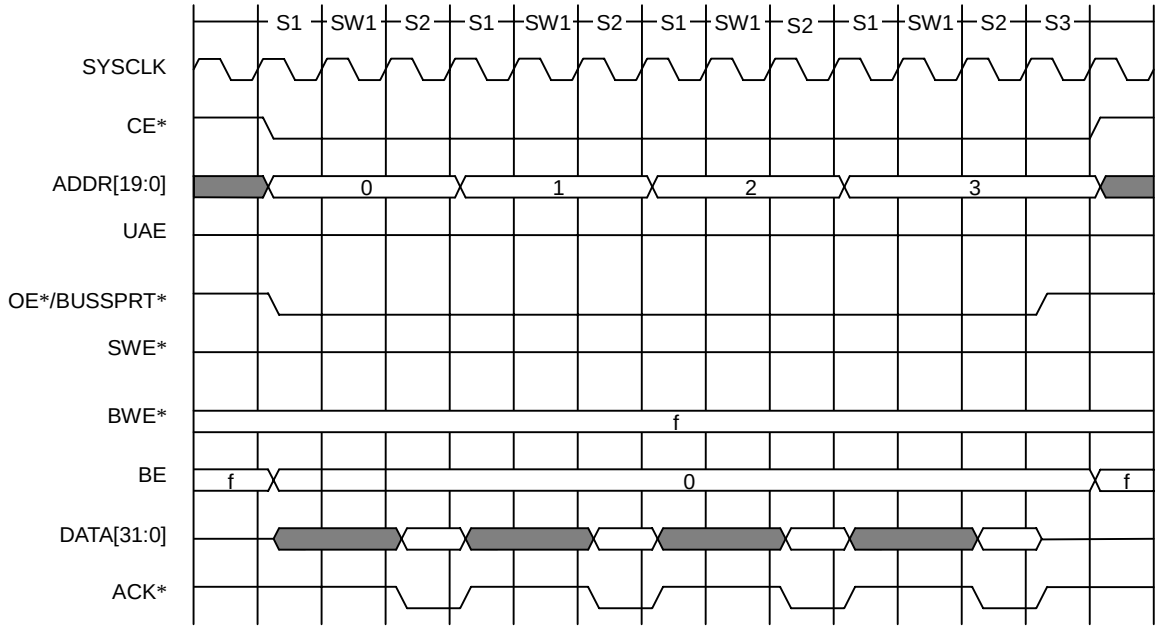


Figure 7.5.8 4-word Burst Write (PWT: WT=1, SHWT=0, Normal, 32-bit Bus)

7.5.4 Normal Mode Access (Single, 16-bit Bus)

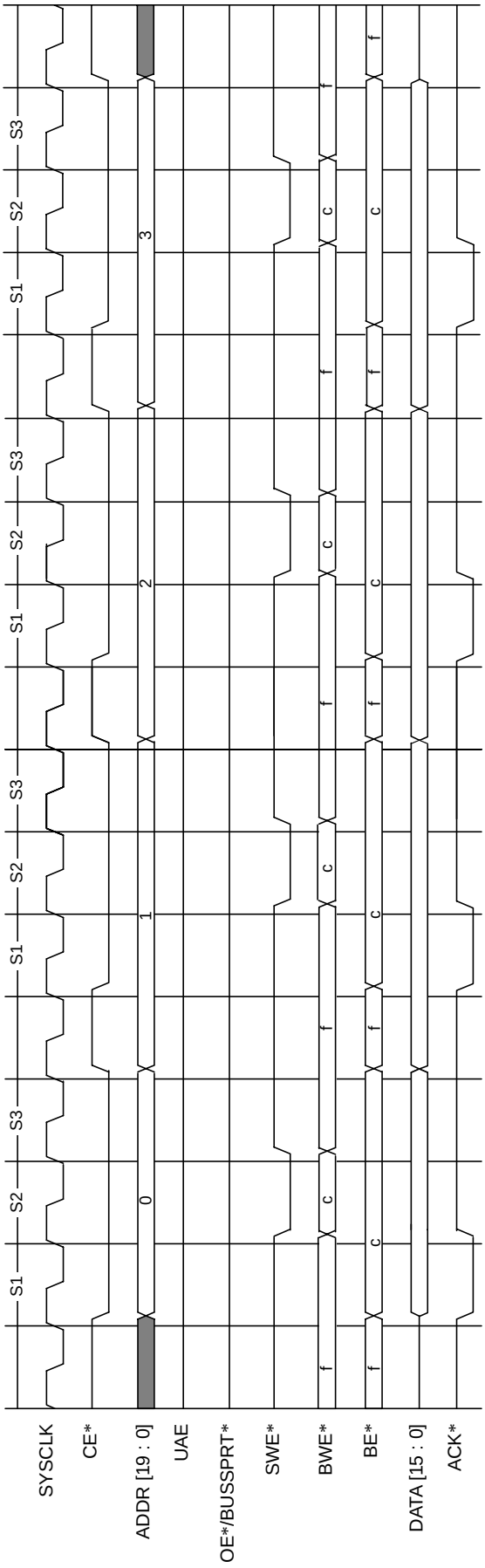


Figure 7.5.9 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

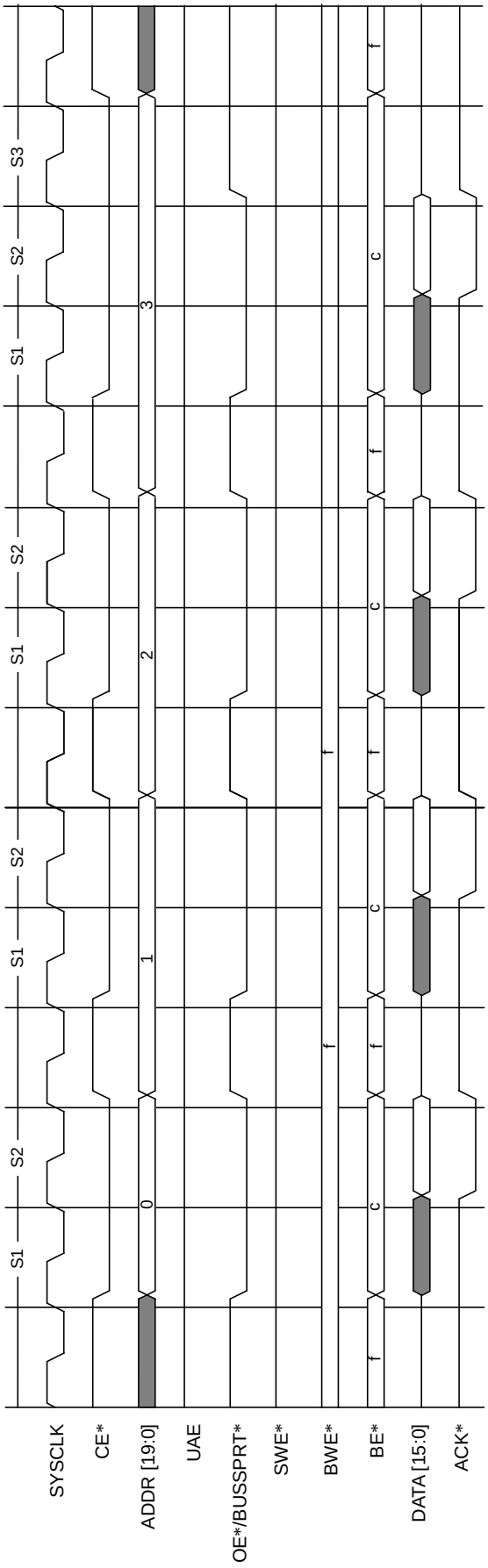


Figure 7.5.10 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

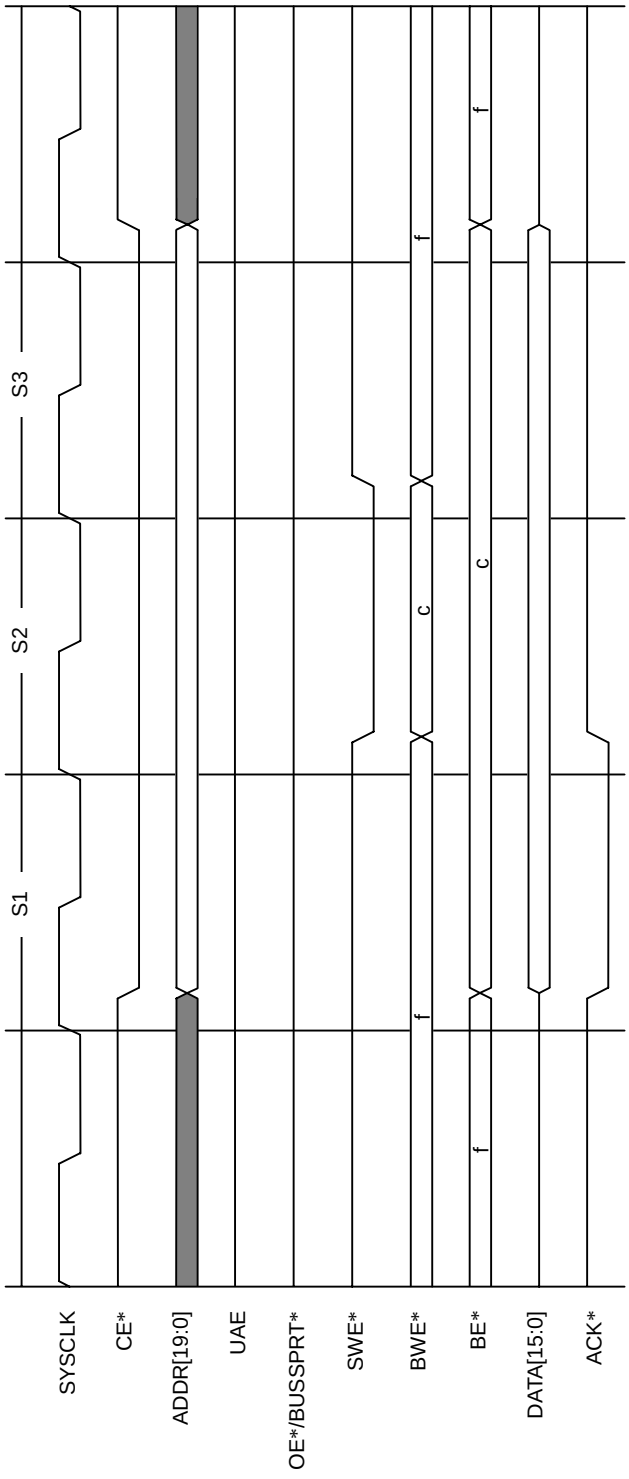


Figure 7.5.11 Half-word Single Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

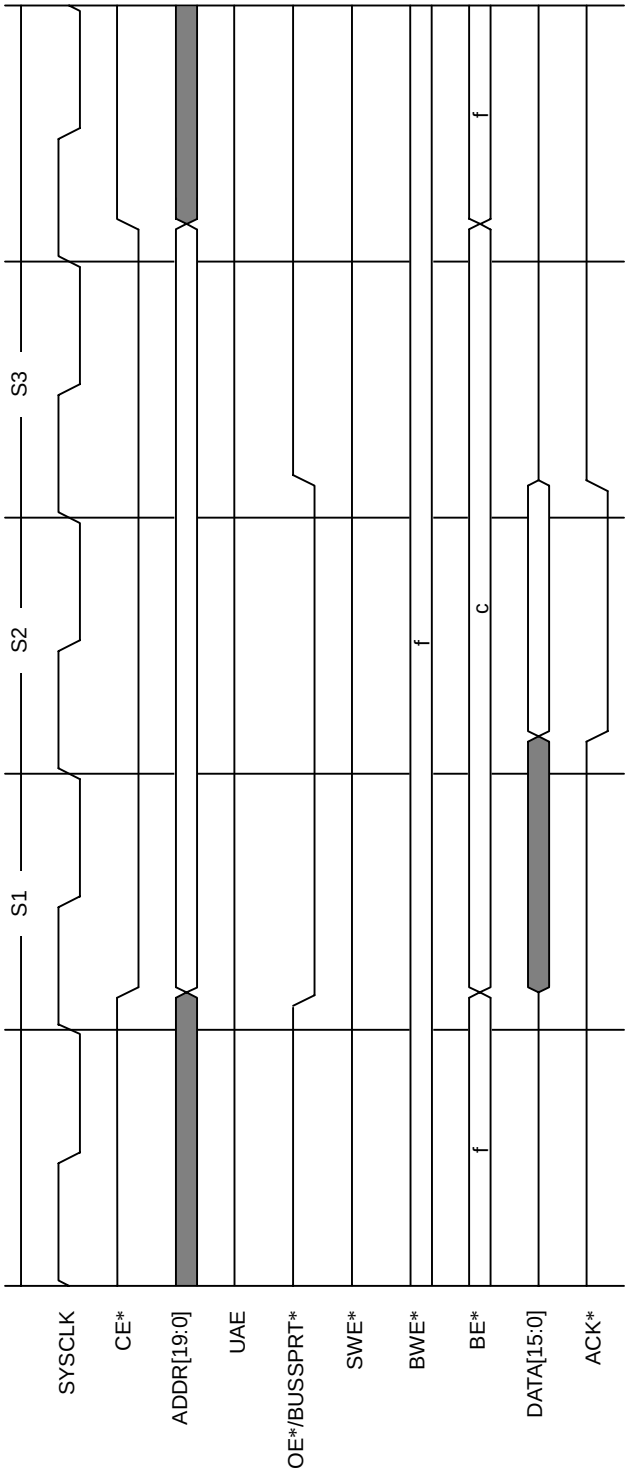


Figure 7.5.12 Half-word Single Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

7.5.5 Normal Mode Access (Burst, 16-bit Bus)

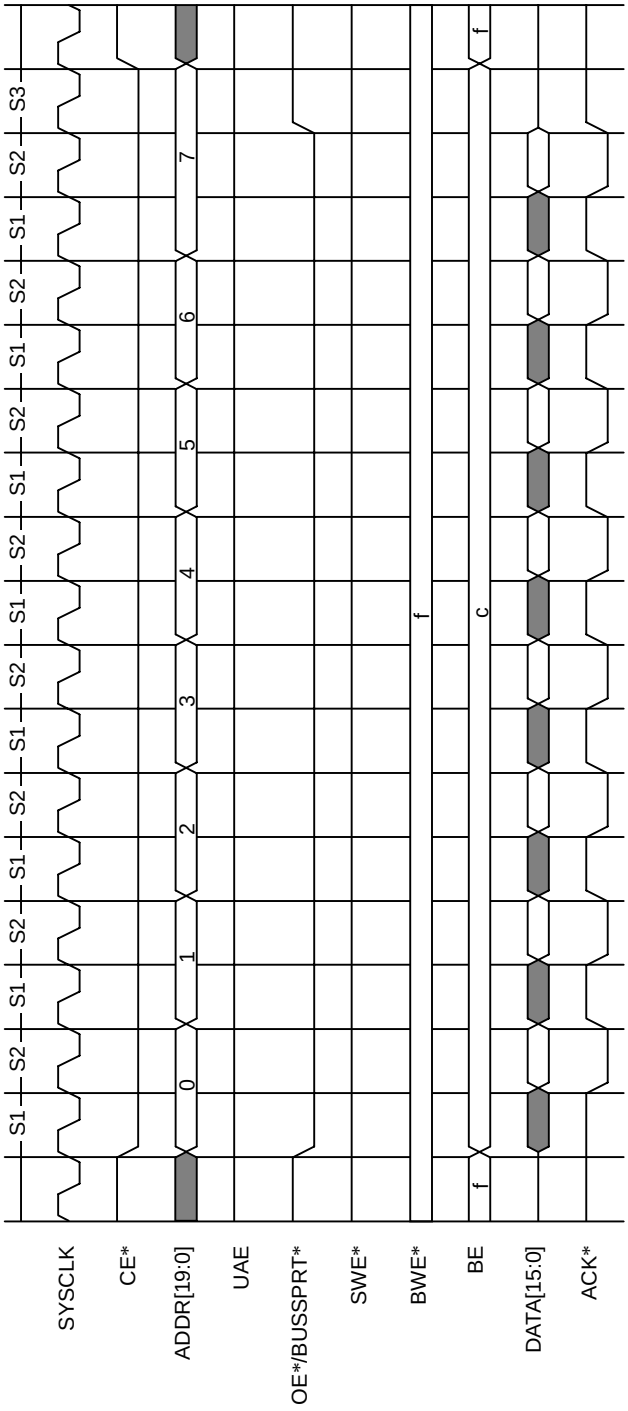


Figure 7.5.13 4-word Burst Read (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

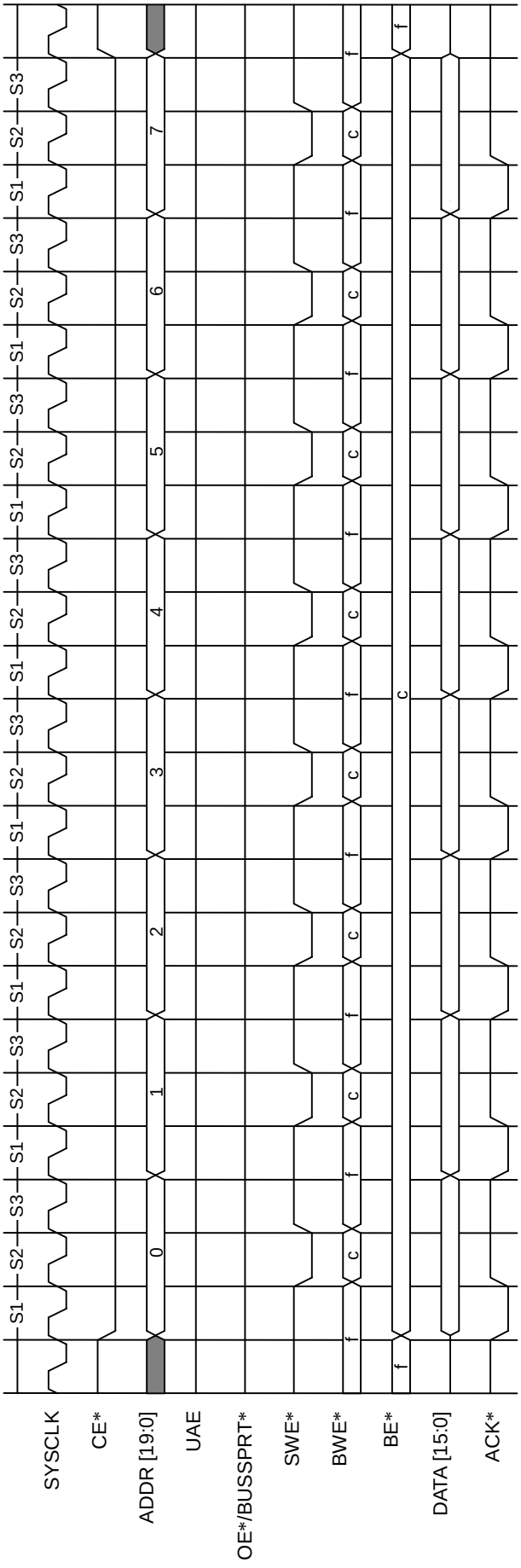


Figure 7.5.14 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 16-bit Bus)

7.5.6 Normal Mode Access (Single, 8-bit Bus)

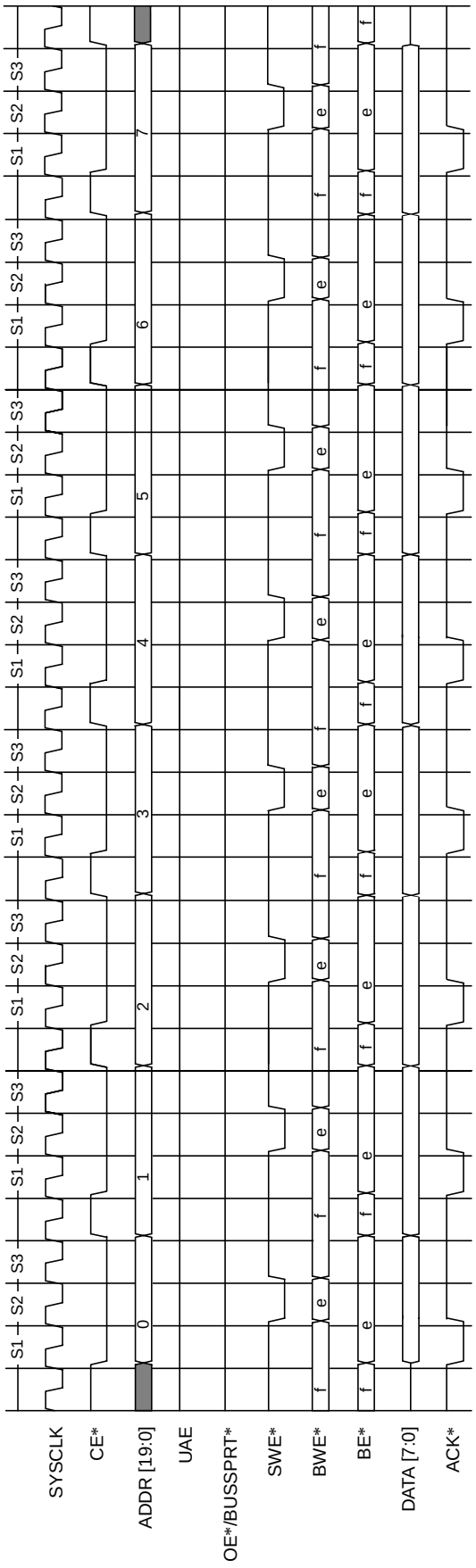


Figure 7.5.15 Double-word Single Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

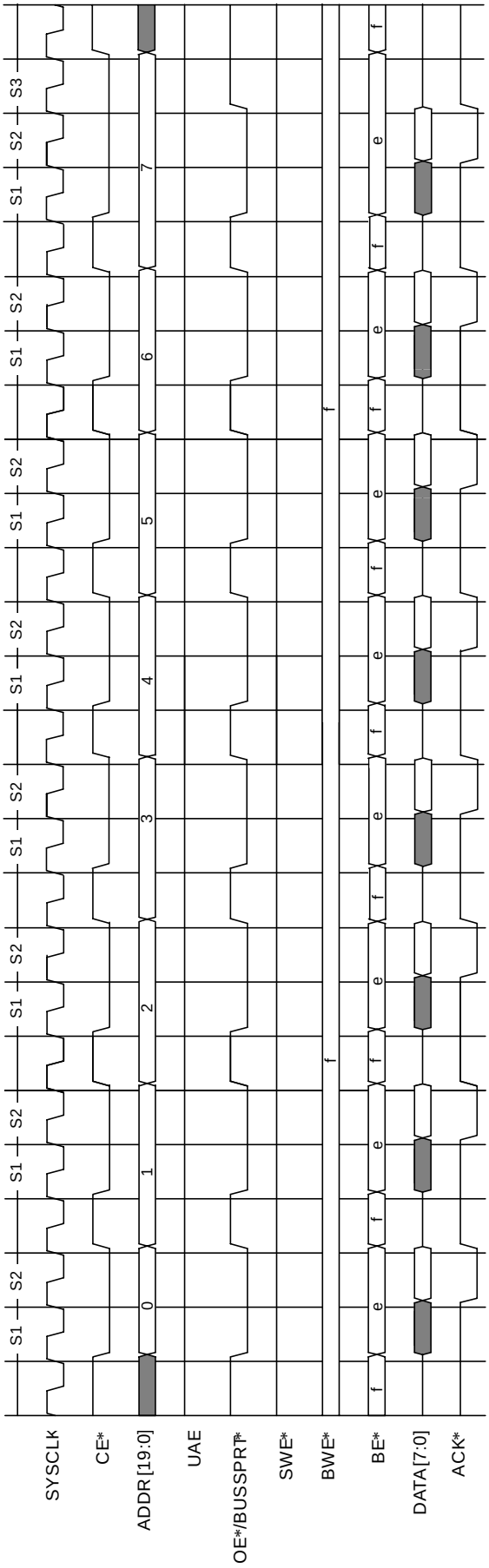


Figure 7.5.16 Double-word Single Read (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

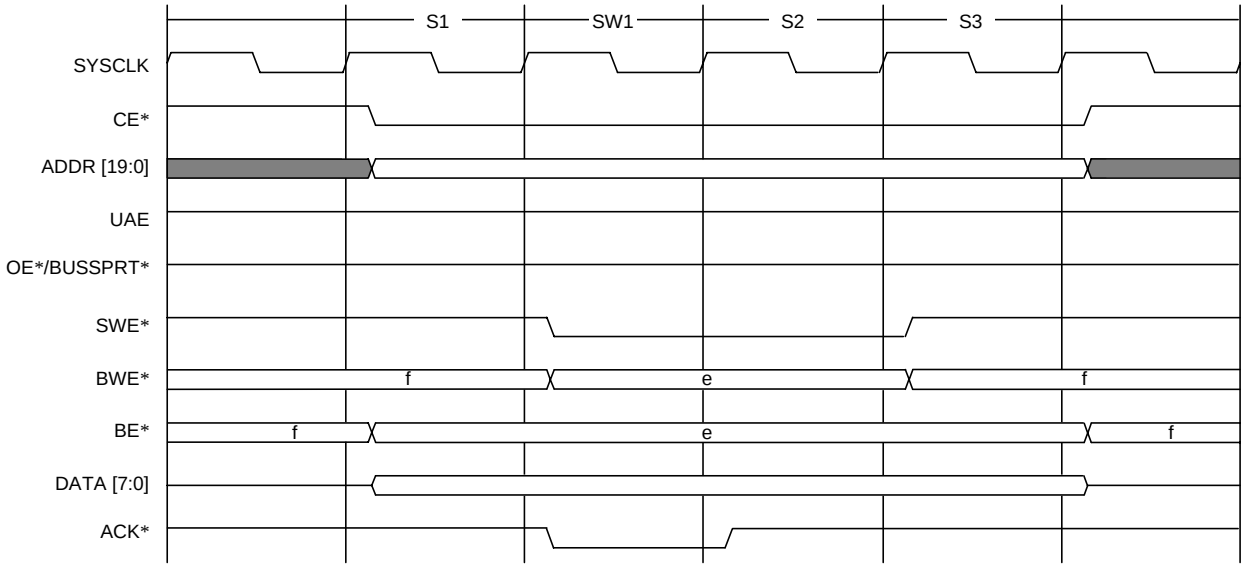


Figure 7.5.17 1-byte Single Write (PWT: WT=1, SHWT=0, Normal, 8-bit Bus)

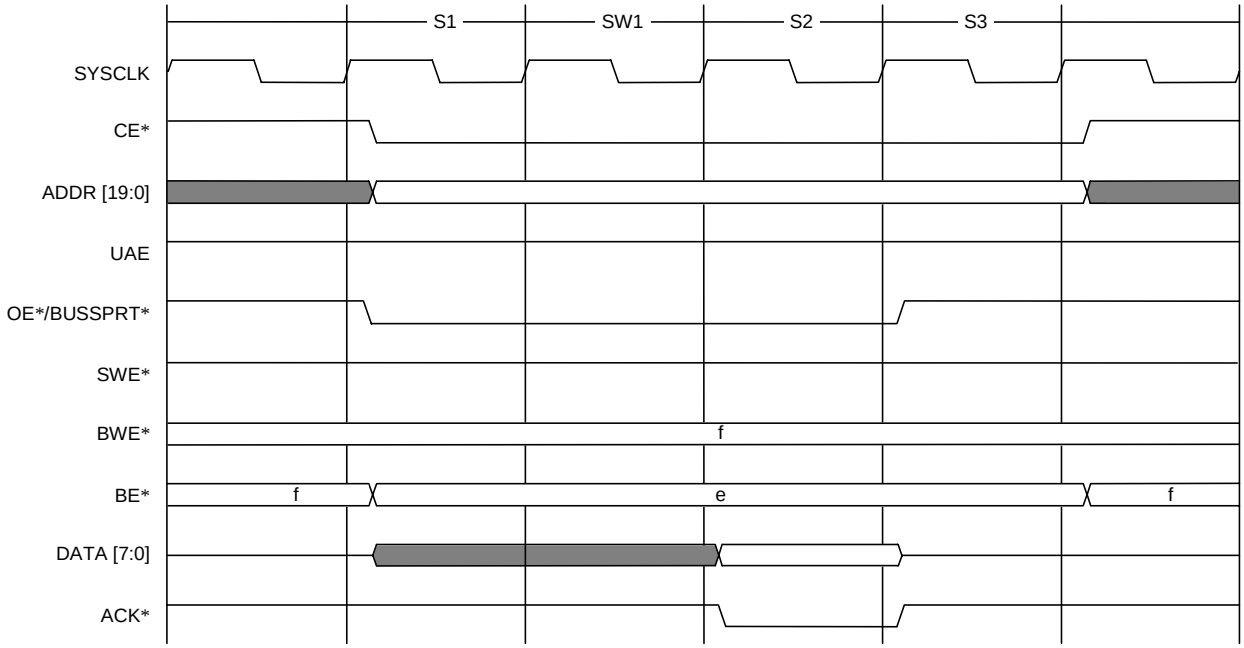


Figure 7.5.18 1-byte Single Read (PWT: WT=1, SHWT=0, Normal, 8-bit Bus)

7.5.7 Normal Mode Access (Burst, 8-bit Bus)

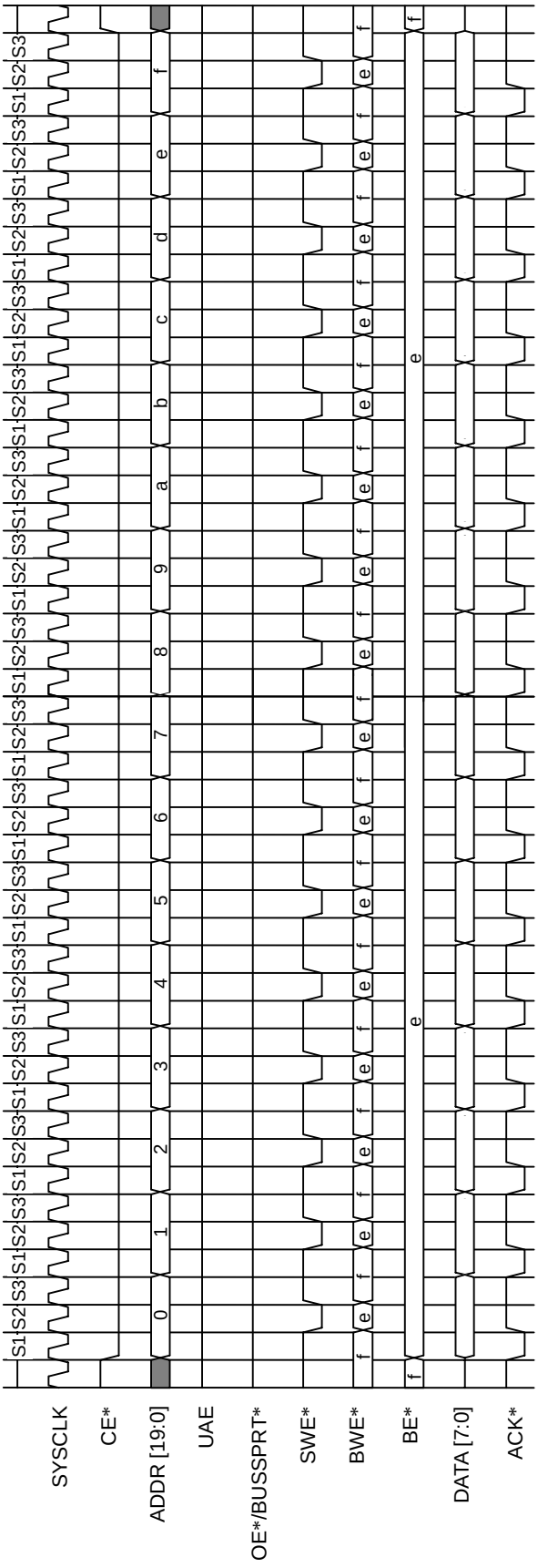


Figure 7.5.19 4-word Burst Write (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

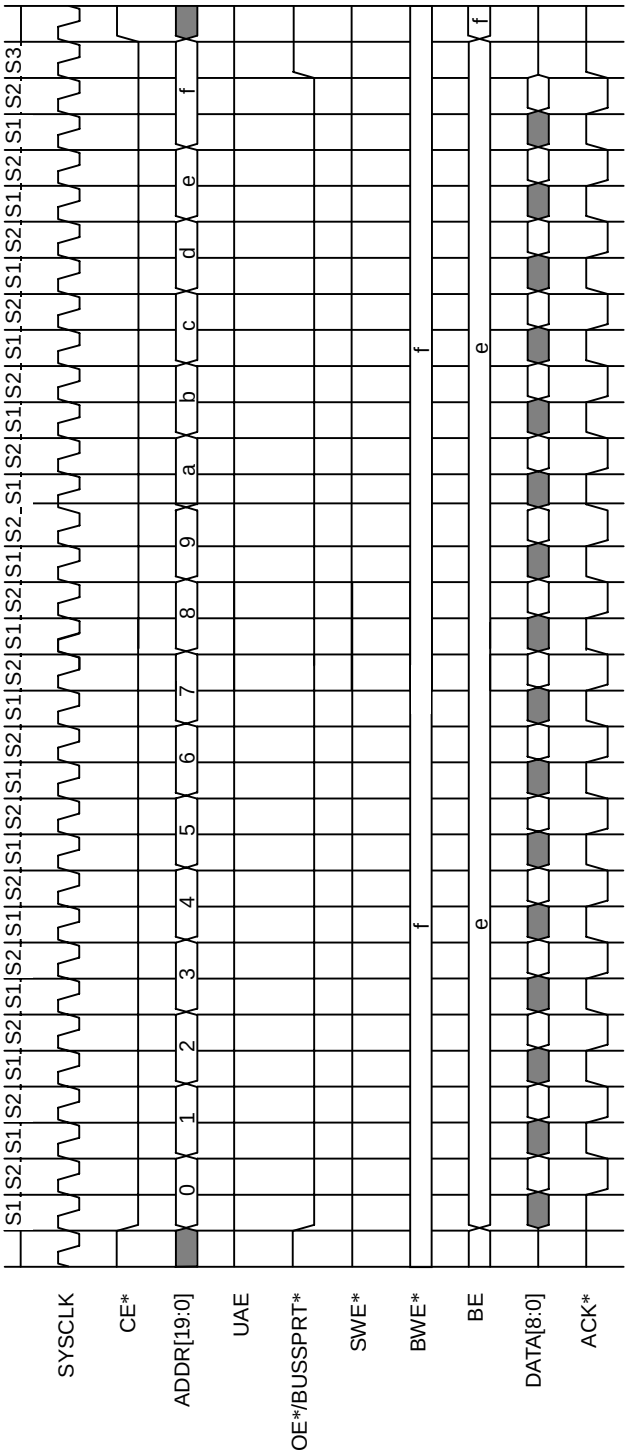


Figure 7.5.20 4-word Burst Read (PWT: WT=0, SHWT=0, Normal, 8-bit Bus)

7.5.8 Page Mode Access (Burst, 32-bit Bus)

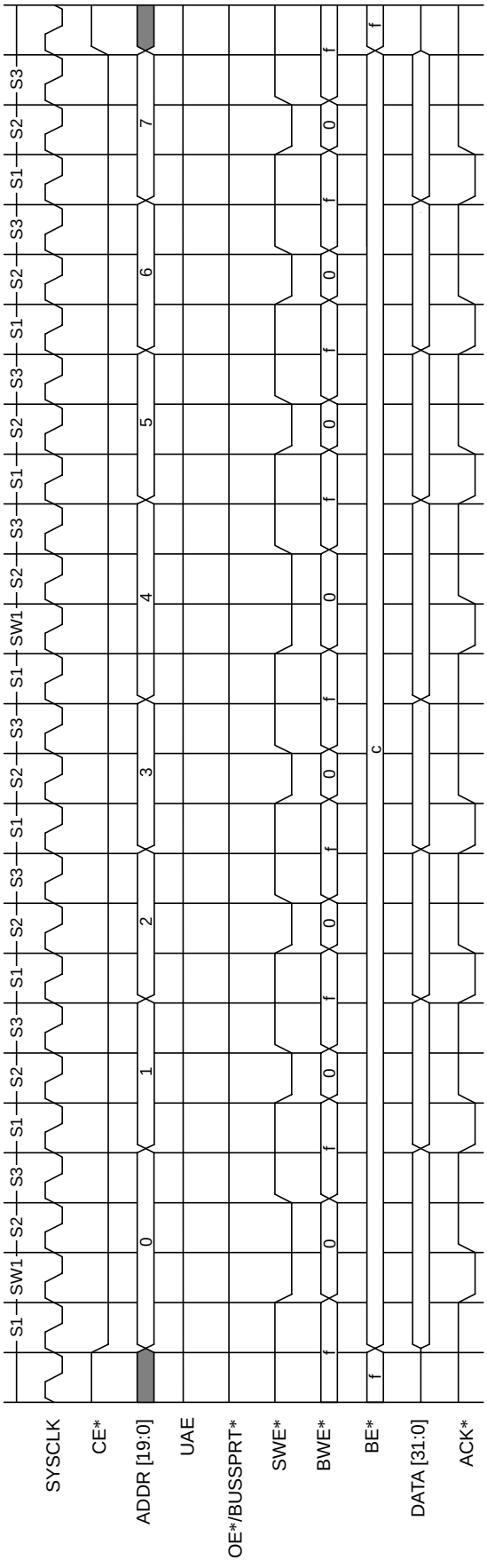


Figure 7.5.21 8-word Burst Write (WT=1, PWT=0, SHWT=0, 4-page, 32-bit Bus)

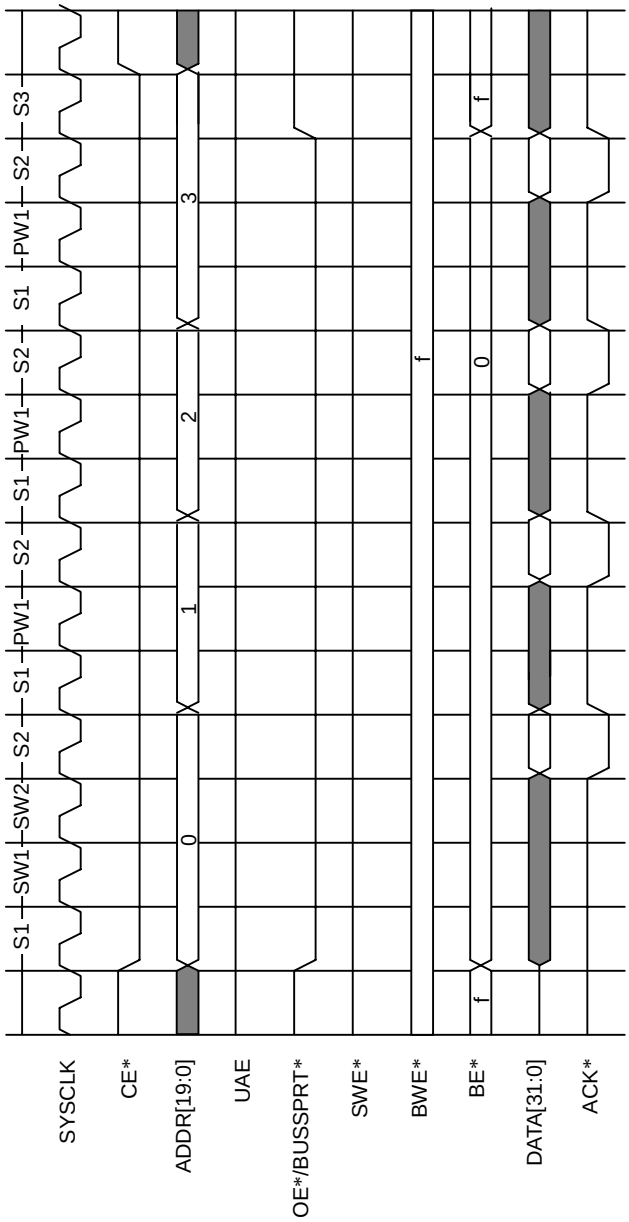
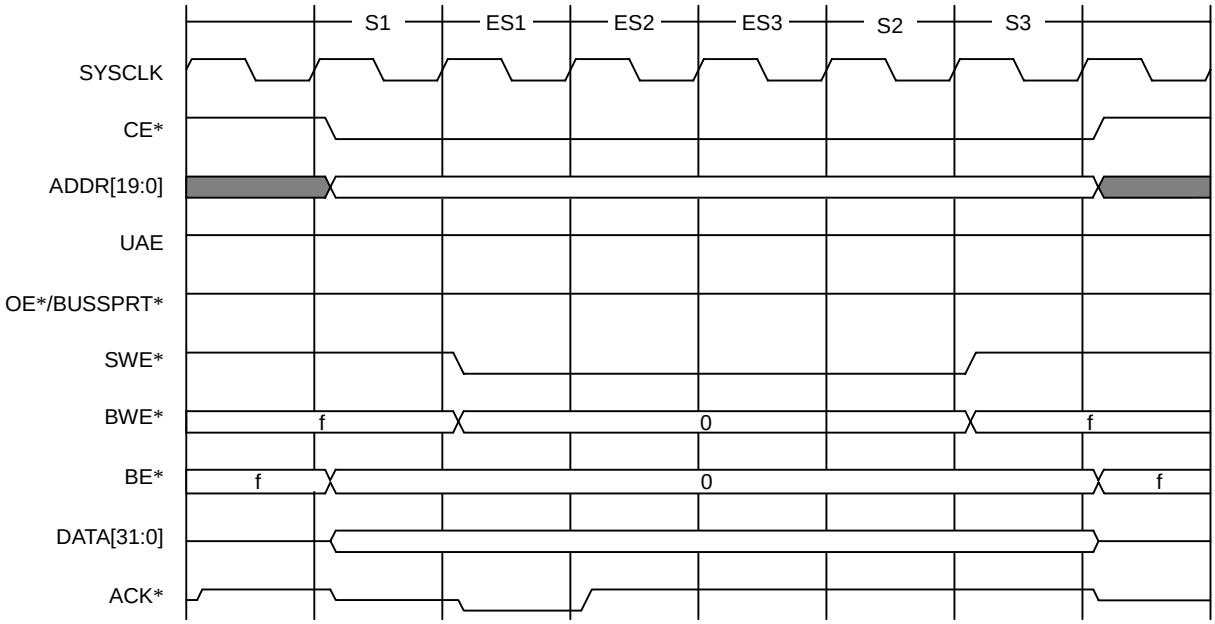


Figure 7.5.22 4-word Burst Read (WT=2, PWT=1, SHWT=0, 4-page, 32-bit Bus)

7.5.9 External ACK Mode Access (32-bit Bus)



Note 1: The TX4925 sets the ACK* signal to High Impedance in the S1 State.

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

Note 3: External devices drive the ACK* signal to High (deassert the signal) in the ES2 State. If an external device is late in asserting ACK*, then the Wait State is inserted for the amount of time the external device is late. If a certain condition is met, it is okay for the ACK* signal to be driven to Low for 1 clock cycle or more. See “7.3.7.4 ACK* Input Timing (External ACK Mode)” for more information.

Figure 7.5.23 1-word Single Write (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

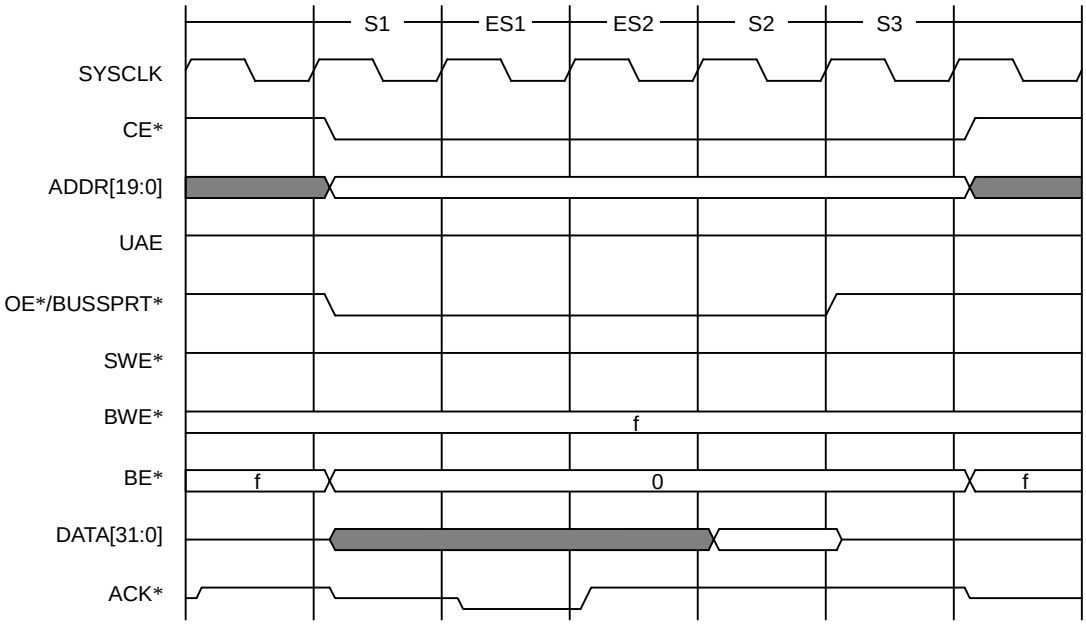


Figure 7.5.24 1-word Single Read (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

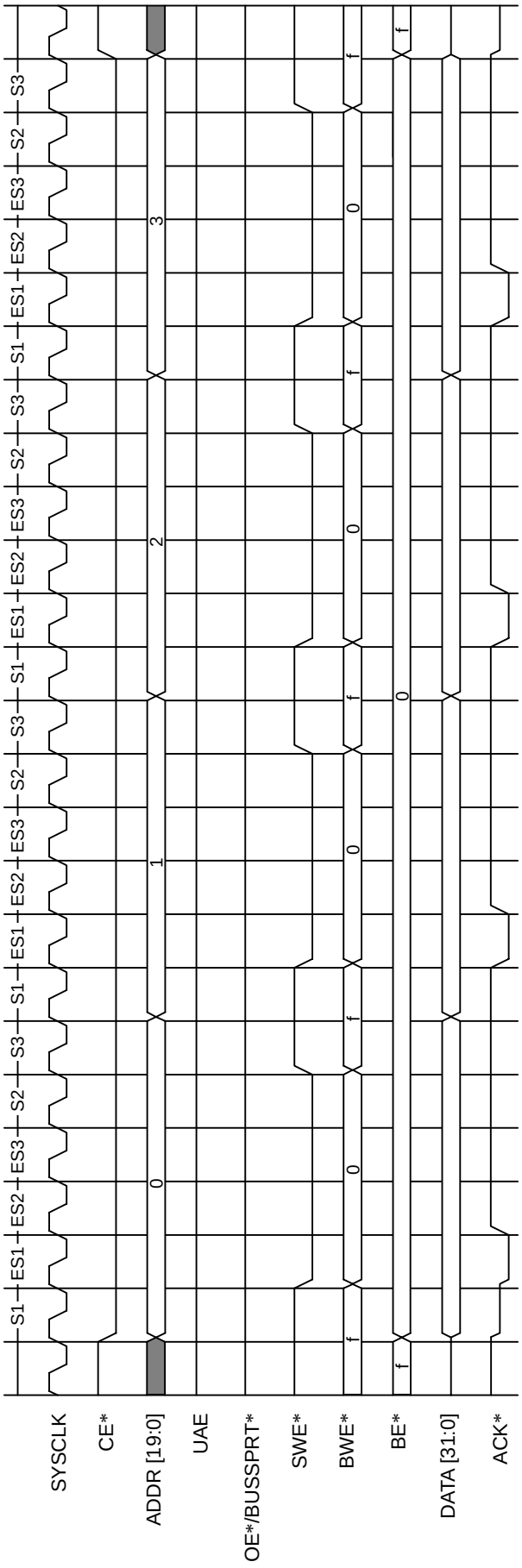


Figure 7.5.25 4-word Burst Write (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

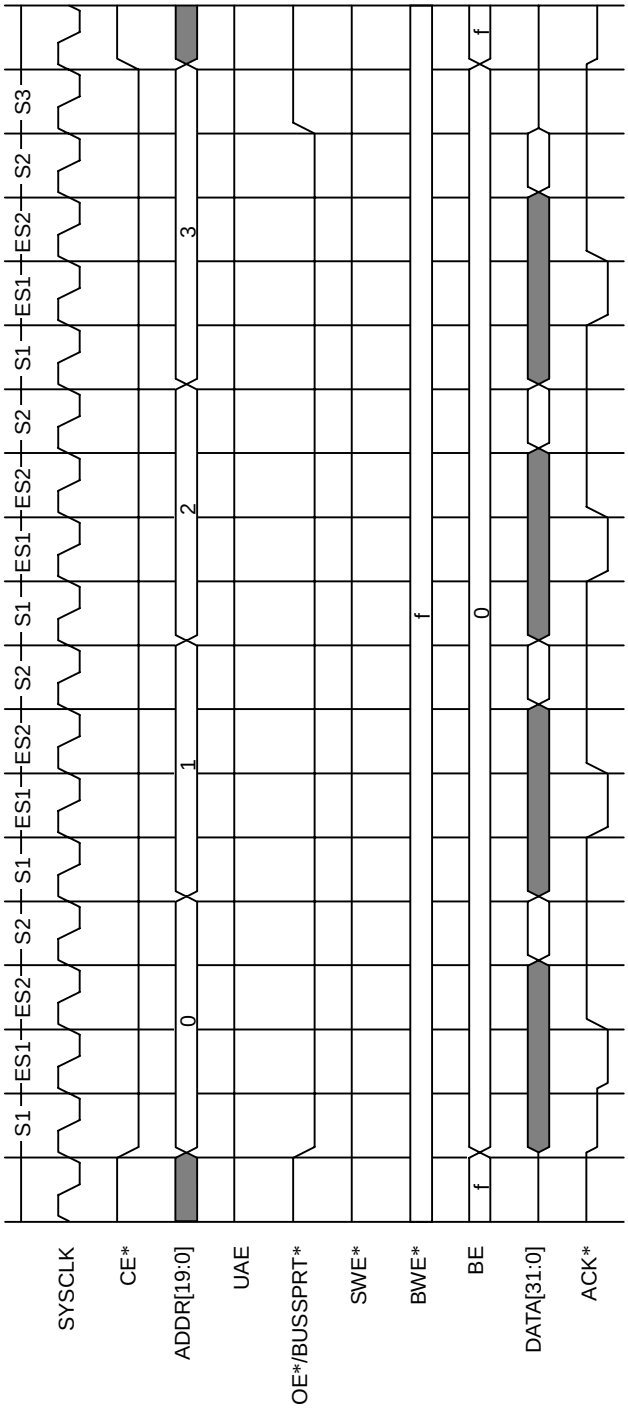


Figure 7.5.26 4-word Burst Read (0 Wait, SHWT=0, External ACK*, 32-bit Bus)

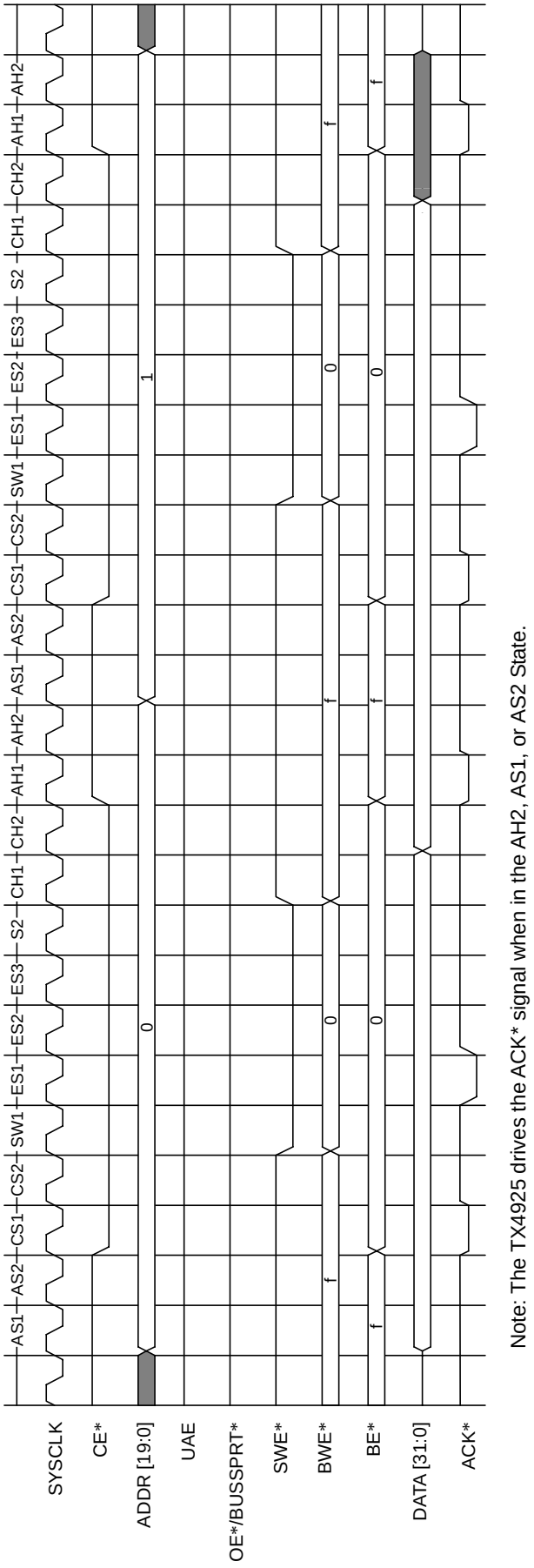


Figure 7.5.27 Double-word Single Write (1 Wait, SHWT=2, External ACK*, 32-bit Bus)

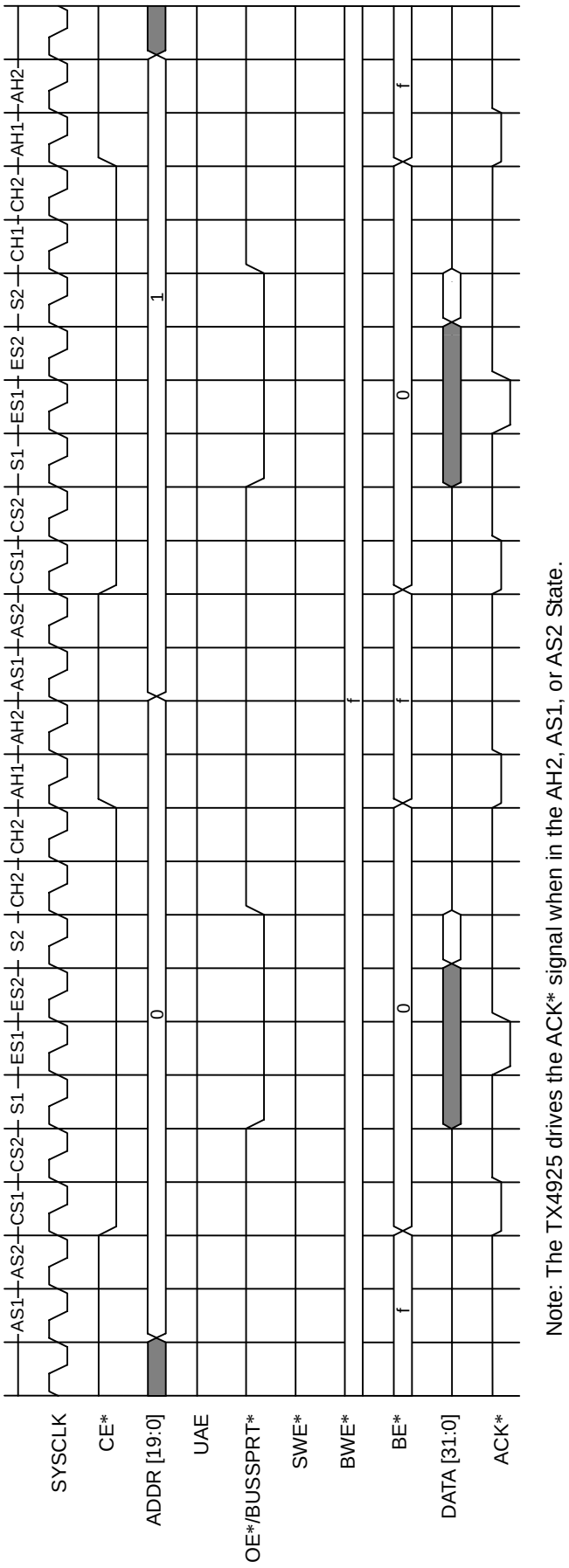


Figure 7.5.28 Double-word Single Read (0 Wait, SHWT=2, External ACK*, 32-bit Bus)

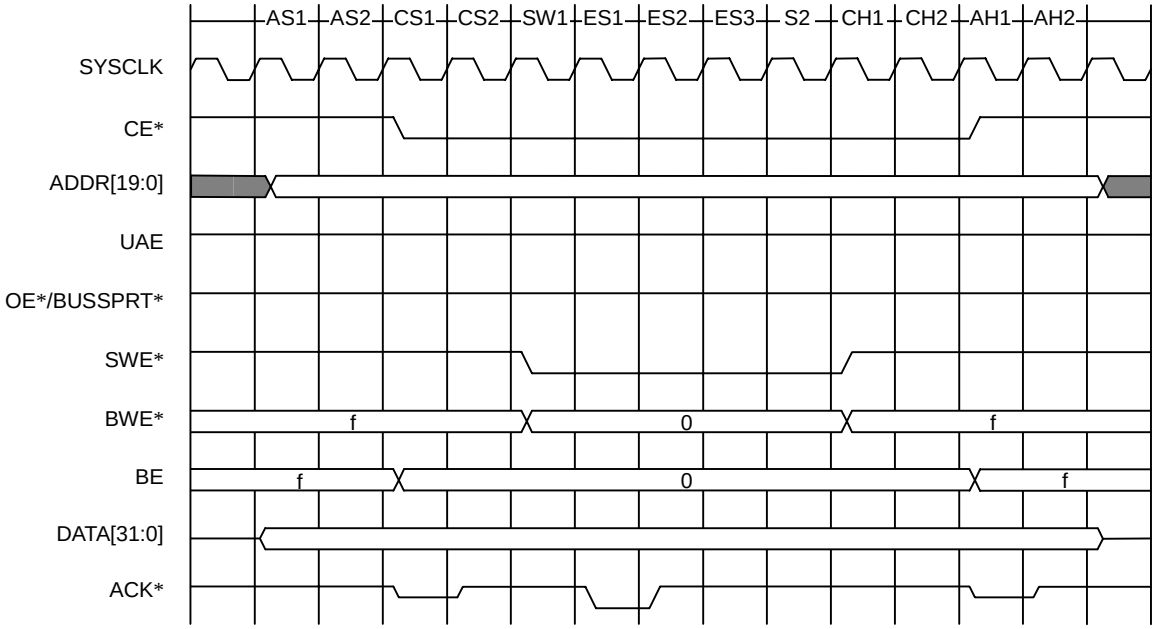


Figure 7.5.29 1-word Single Write (1 Wait, SHWT=2, External ACK*, 32-bit Bus)

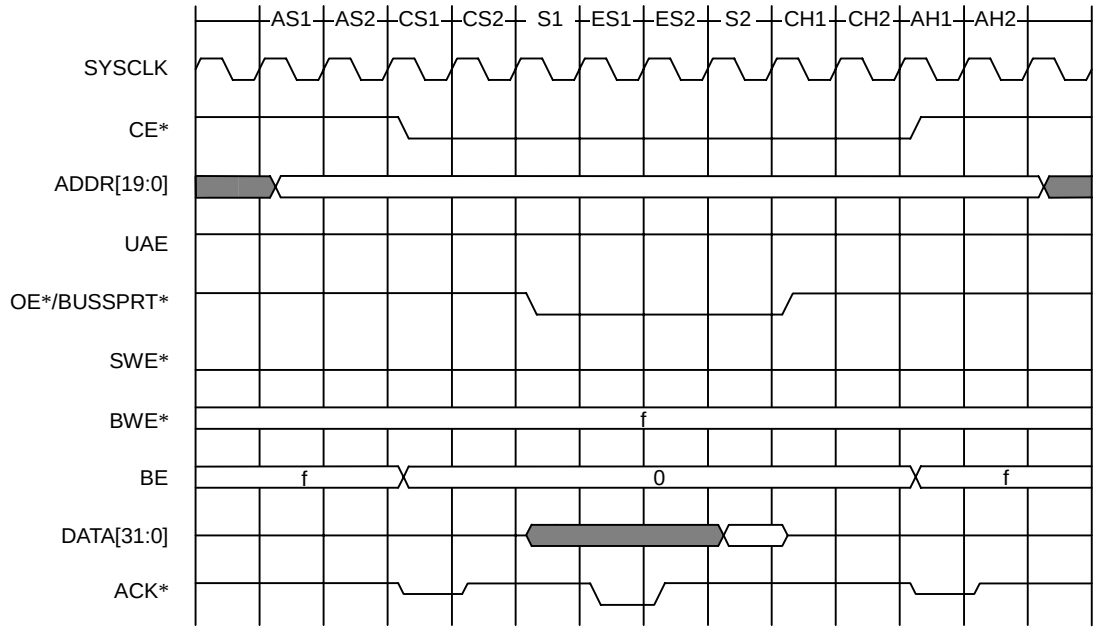


Figure 7.5.30 1-word Single Read (0 Wait, SHWT=2, External ACK*, 32-bit Bus)

7.5.10 READY Mode Access (32-bit Bus)

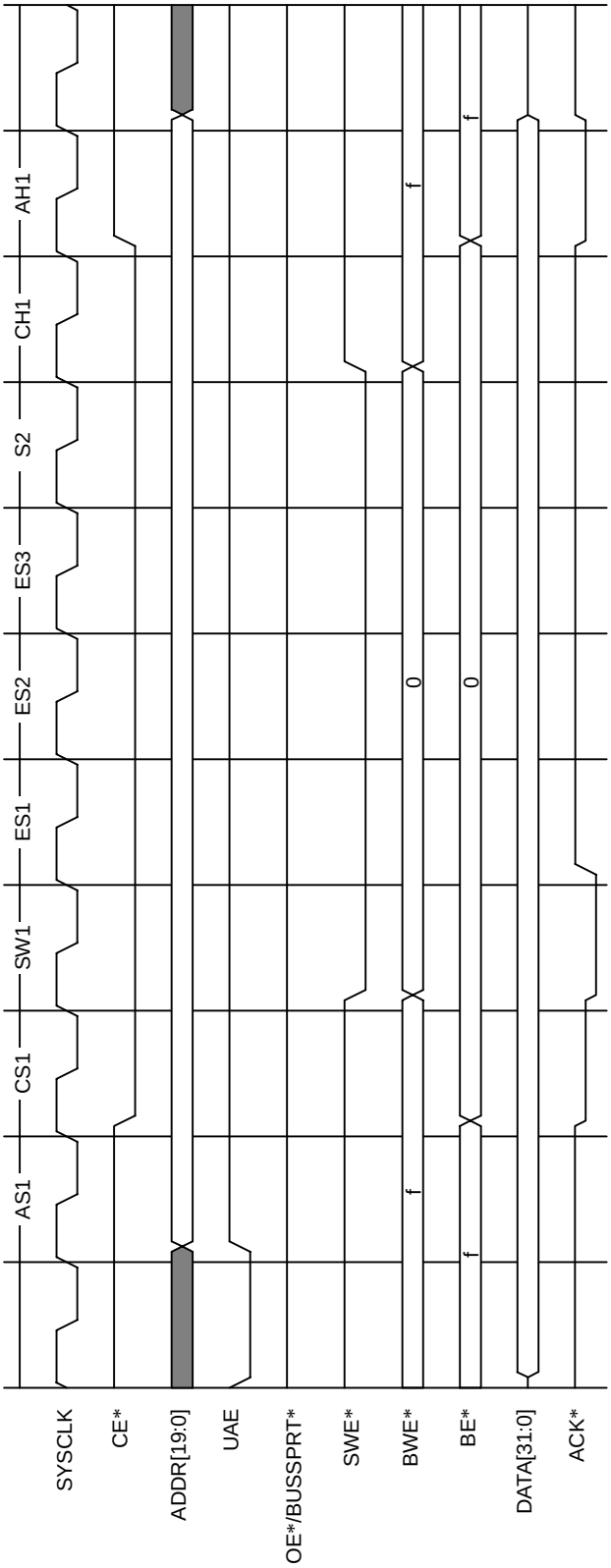


Figure 7.5.31 1-word Single Write (PWT: WT=2, SHWT=1, READY, 32-bit Bus)

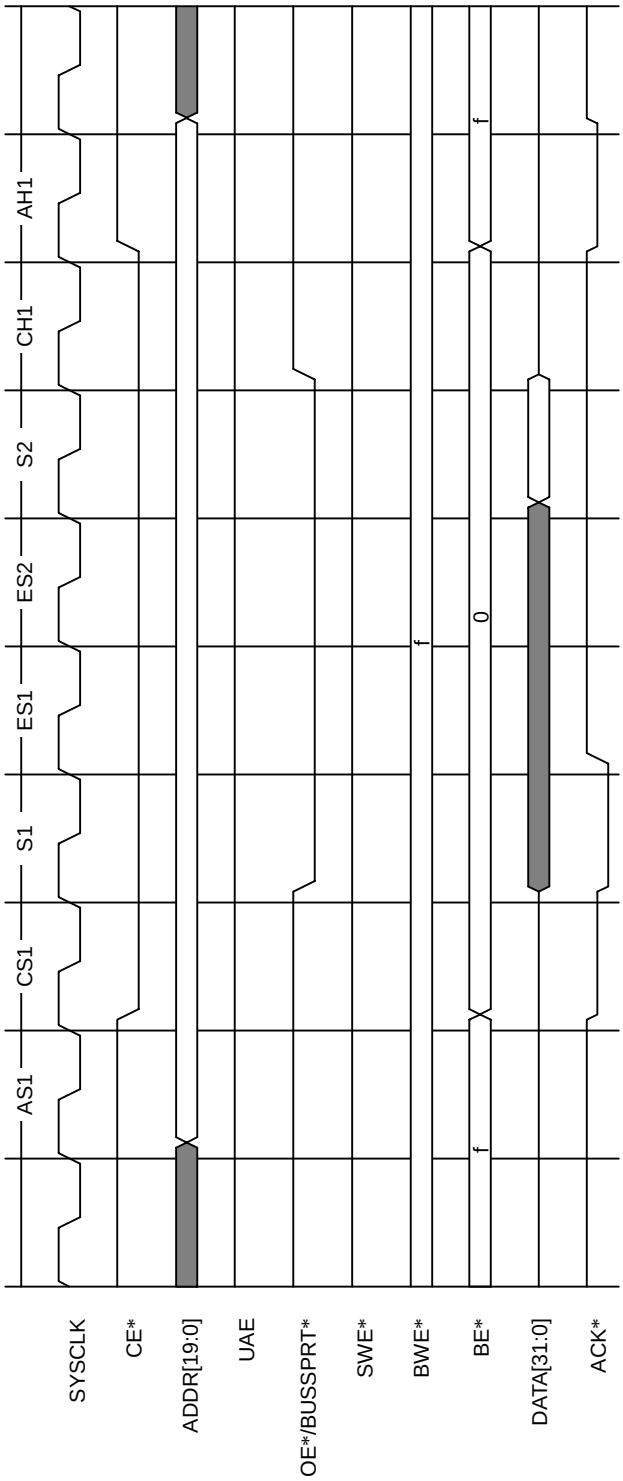


Figure 7.5.32 1-word Single Read (PWT: WT=2, SHWT=1, READY, 32-bit Bus)

7.6 Flash ROM, SRAM Usage Example

Figure 7.6.1 illustrates example Flash ROM connections, and Figure 7.6.2 illustrates example SRAM connections. Also, Figure 7.6.3 illustrates example connections with the SDRAM and the bus separated.

Since connecting multiple memory devices such as SDRAM and ROM onto a single bus increases the load, 100 MHz class high-speed SDRAM access may not be performed normally. As a corrective measure, there is a way of reducing the bus load by connecting a device other than SDRAM via a buffer. If such a method is employed, directional control becomes necessary since the data becomes bidirectional.

The TX4925 prepares the BUSSPRT* signal for performing data directional control (see Figure 7.6.3). BUSSPRT* is asserted when the External Bus Controller channel is active and a Read operation is being performed.

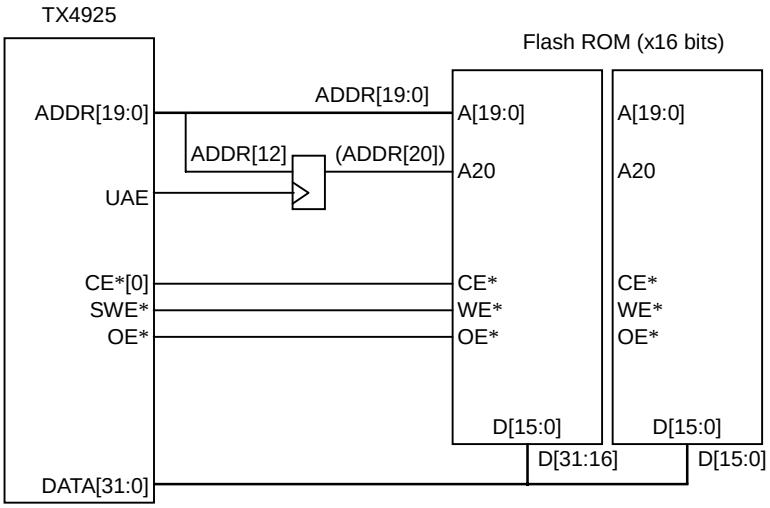


Figure 7.6.1 Flash ROM (x16 Bits) Connection Example (32-bit Data Bus)

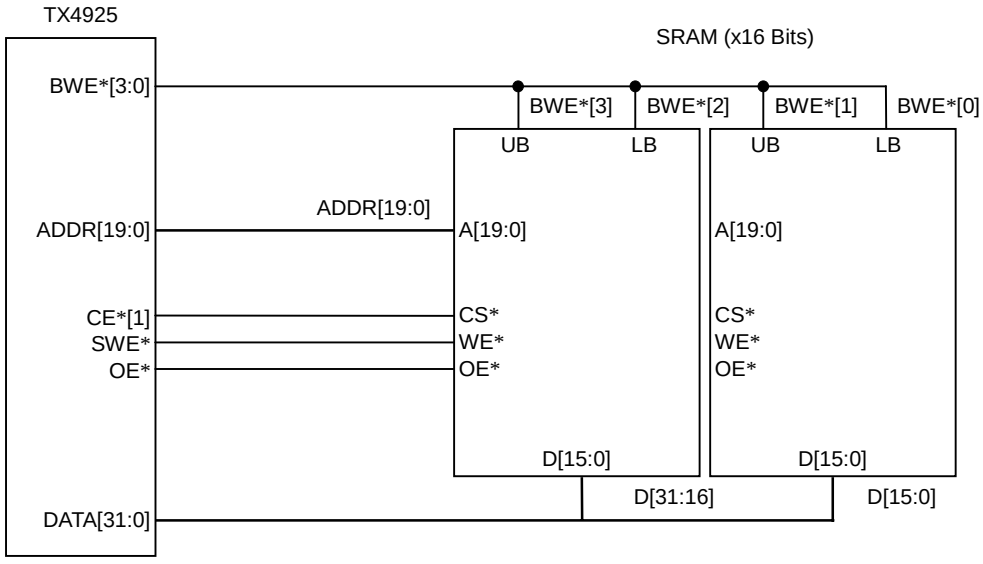


Figure 7.6.2 SRAM (x16 Bits) Connection Example (32-bit Data Bus)

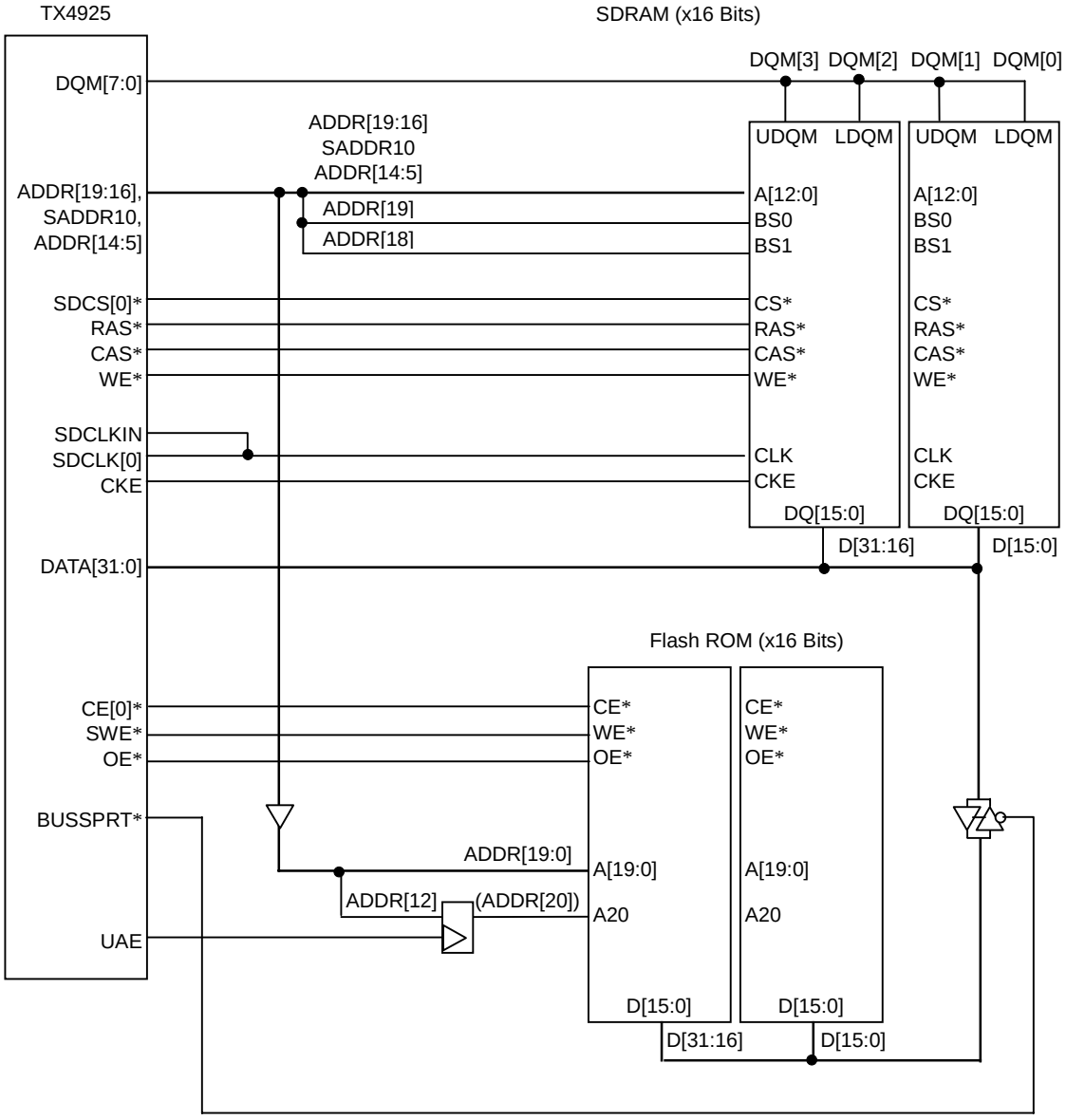


Figure 7.6.3 Connection Example with SDRAM and the Bus Separated

8. DMA Controller

8.1 Features

The TX4925 contains a four-channel DMA Controller (DMAC) that executes DMA (Direct Memory Access) with memory and I/O devices.

The DMA Controller has the following characteristics.

- Has four on-chip DMA channels
- Supports external I/O devices with 8-, 16-, and 32-bit Data Bus widths and transfer between memory devices.
- Supports single address transfer (Fly-by DMA) and dual address transfer when in the external I/O DMA Transfer Mode that is operated by external request signals
- Supports on-chip Serial I/O Controllers and AC-Link Controllers
- Supports Memory-Memory Copy modes that do not have address boundary limitations. Burst transfer of up to eight words is possible for each Read or Write operation.
- Supports Memory Fill mode that writes word data to the specified memory region
- Supports Chained DMA Transfer
- On-chip signed 24-bit address count up registers for both the source address and destination address
- On-chip 26-bit Byte Count Register for each channel
- One of two methods can be selected for determining access priority among multiple channels: Round Robin or Fixed Priority
- Big Endian or Little Endian mode can be set separately for each channel

8.2 Block Diagram

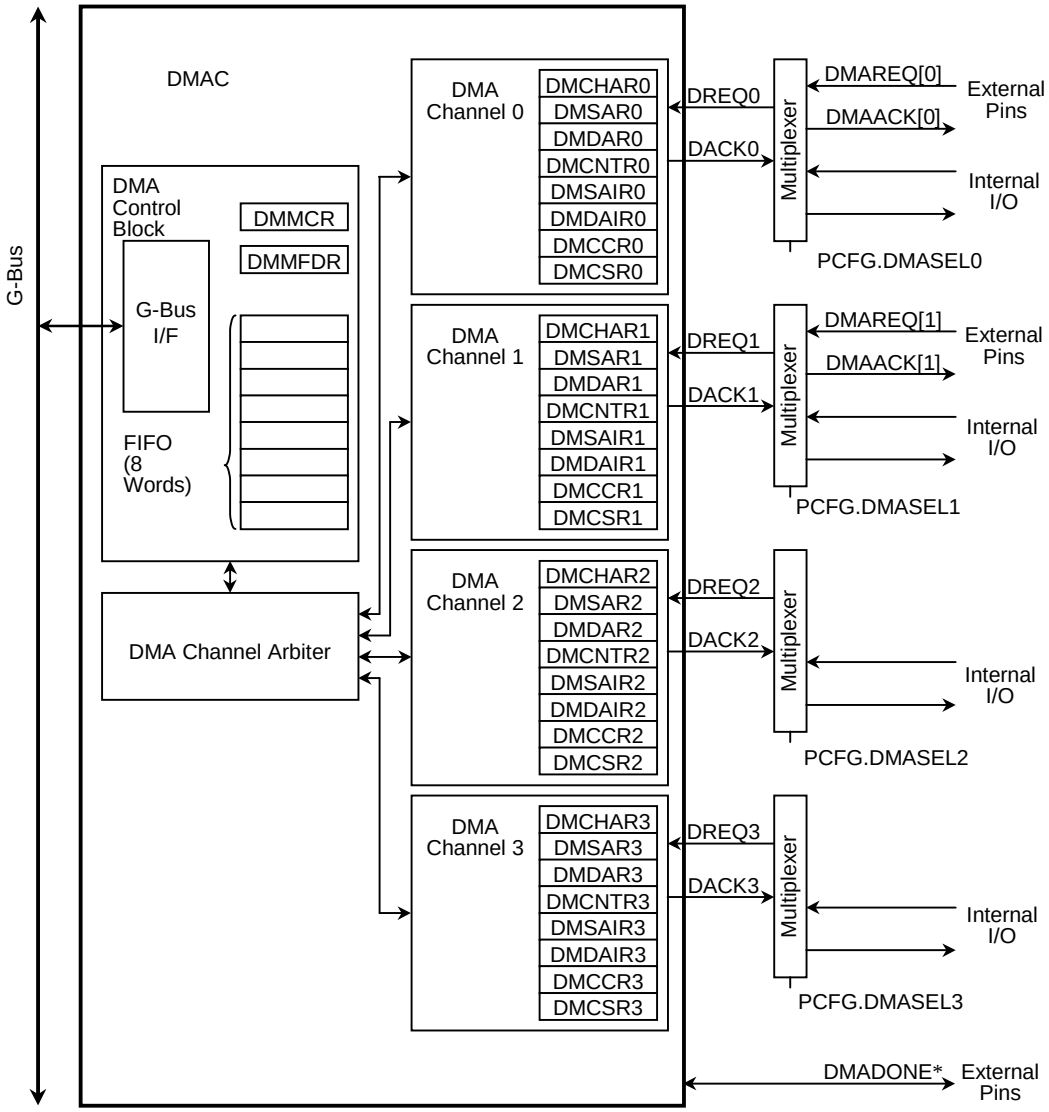


Figure 8.2.1 DMA Controller Block Diagram

8.3 Detailed Explanation

8.3.1 Transfer Mode

The DMA Controller supports five transfer mode types (refer to Table 8.3.1 below). The setting of the External Request bit (DMCCRn.EXTRQ) of the DMA Channel Control Register selects whether transfer with an I/O device is a DMA transfer.

- I/O DMA Transfer Mode (DMCCRn.EXTRQ = "1")
  Perform DMA transfer with either an external device connected to the External Bus Controller or an on-chip I/O device (ACLC or SIO).

- Memory Transfer Mode (DMCCRn.EXTRQ = "0")
  Either copies data between memory devices or fills data in memory.

Table 8.3.1 DMA Controller Transfer Modes

Transfer Mode	DMCCRn EXTREQ	DRQCTR DMAREQn	DMCCRn SNGAD	DMSAR	DMDAR	Ref.
External I/O (Single Address)	1	0xxxx	1	√	-	8.3.3 8.3.7
External I/O (Dual Address)	1	0xxxx	0	√	√	8.3.3 8.3.8
Internal I/O	1	10xx (SIO) 11xx (ACLC)	0	√	√	8.3.4 8.3.8
Memory-Memory Copy	0	-	0	√	√	8.3.4 8.3.8
Fill Memory	0	-	1	√	-	8.3.6 8.3.7

8.3.2 On-chip Registers

The DMA Controller has two shared registers that are shared by four channels. Section 8.4 explains each register in detail.

- Shared Registers
  DMMCR: DMA Master Control Register
  DMMFDR: DMA Memory Fill Data Register

- DMA Channel Register
  DMCHARn: DMA Chained Address Register
  DMSARn: DMA Source Address Register
  DMDARn: DMA Destination Address Register
  DMCNTRn: DMA Count Register
  DMSAIRn: DMA Source Address Increment Register
  DMDAIRn: DMA Destination Address Increment Register
  DMCCRn: DMA Channel Control Register
  DMCSRn: DMA Channel Status Register

8.3.3 External I/O DMA Transfer Mode

The External I/O DMA Transfer Mode performs DMA transfer with external I/O devices that are connected to the External Bus Controller.

8.3.3.1 External Interface

External I/O devices signal DMA requests to the DMA Controller by asserting the DMA Transfer Request Signal (DMAREQ[n]). On the other hand, the DMA Controller accesses external I/O devices by asserting the DMA Transfer Acknowledge Signal (DMAACK[n]).

The DMA Transfer Request signal (DMAREQ[n]) can use the Request Polarity bit (REQPOL) of the DMA Channel Control Register (DMCCRn) to select the signal polarity for each channel, and can use the Edge Request bit (EGREQ) to select either edge detection or level detection for each channel. The DMA Transfer Acknowledge signal (DMAACK[n]) can also use the Acknowledge Polarity bit (ACKPOL) to select the polarity.

Please assert/deassert the DMAREQ[n] signal as follows below.

- When level detection is set (DMCCRn.EGREQ = 0)

  The DMAREQ[n] signal must be continuously asserted until one SYSCLK cycle after the DMAACK[n] signal is asserted. Also, the DMAREQ[n] signal must be asserted before the CE*/CS* signal is deasserted. If this signal is asserted too soon, DMA transfer will not be performed. If this signal is asserted too late, unexpected DMA transfer may result.

  During Dual Address transfer, we recommend detecting assertion of the CE* signal for the external I/O device that is currently asserting DMAACK[n], then deasserting DMAREQ[n].

- When edge detection is set (DMCCRn.EGREQ = 1)

  Please set up assertion of the DMAREQ[n] signal so the DMAREQ[n] signal is asserted after the DMAACK[n] signal corresponding to a previously asserted DMAREQ[n] signal is deasserted. The DMAREQ[n] signal will not be detected even if it is asserted before DMAACK[n] is deasserted.

Figure 8.3.1 is a timing diagram that shows the timing of external DMA access. In this timing diagram, both the DMAREQ[n] signal and the DMAACK[n] signal are set to Low active (DMCCRn.REQPL = 0, DMCCRn.ACKPOL = 0).

The DMAACK[n] and DMADONE signals, which are DMA control signals, are synchronized to SDCLK. When these signals are used by an external I/O device that is synchronous to SYSCLK, it is necessary to take clock skew into account.

The DMAACK[n] signal is asserted either at the SYSCLK cycle, the same as with assertion of the CE*/CS* signal, or before that. In addition, it is deasserted after the last ACK*/READY signal is deasserted.

When the DMADONE* signal (refer to 8.3.3.4) is used as an output signal, it is asserted for at least one SYSCLK cycle while the DMAACK[n] signal is asserted either during the same SYSCLK cycle that the CE*/CS* signal is deasserted or during a subsequent SYSCLK cycle. When the DMADONE* signal is used as an input signal, it must be asserted for one SYSCLK cycle while the DMAACK[n] signal is being asserted.

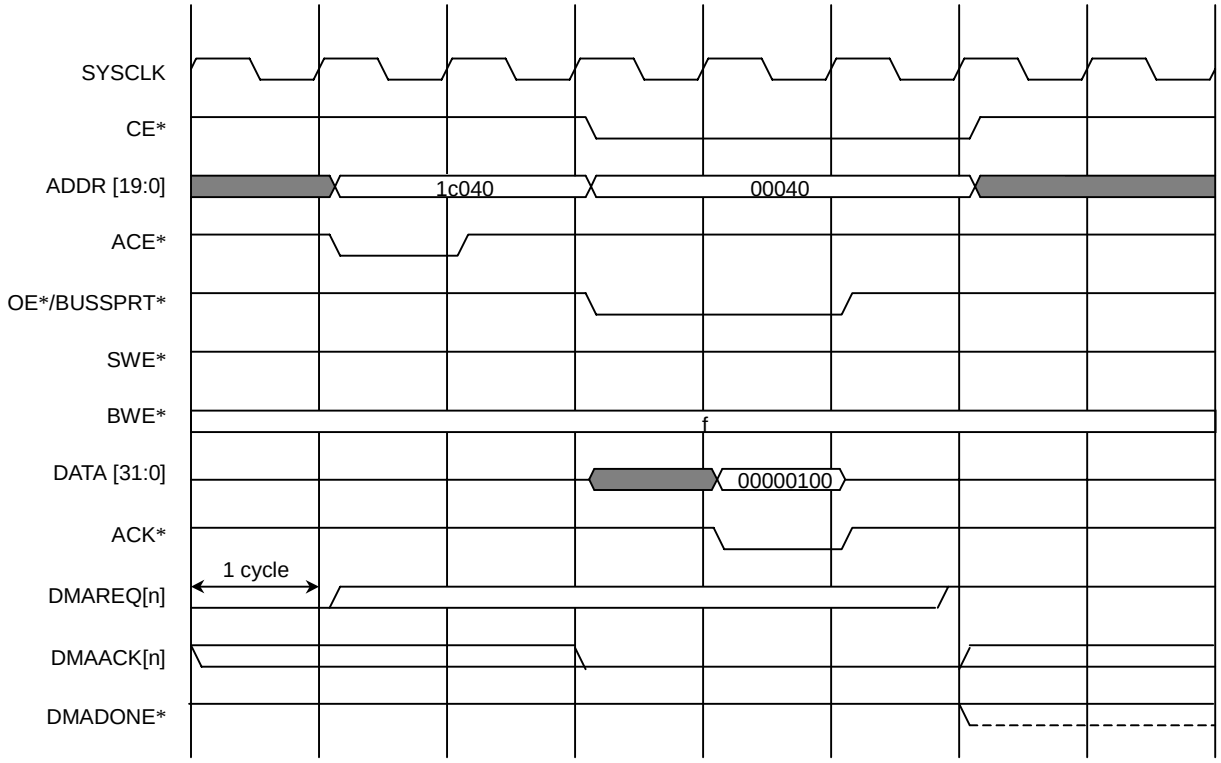


Figure 8.3.1 External I/O DMA Transfer (Single Address, Level Request)

8.3.3.2 Dual Address Transfer

If the Single Address bit (DMCCRn.SNGAD) has been cleared, access to external I/O devices and to external memory is each performed continuously. Each access is the same as normal access except when the DMAACK[n] signal is asserted.

Please refer to “8.3.8 Dual Address Transfer” for information regarding setting the register.

8.3.3.3 Single Address Transfer (Fly-by DMA)

If the Single Address bit (DMCCRn.SNGAD) is set, either data reading from an external I/O device and data writing to external memory or data reading from external memory and data writing to an external I/O device is performed simultaneously. The following conditions must be met in order to perform Single Address transfer.

- The data bus widths of the external I/O device and external memory match
- Data can be input/output to/from the external I/O device and external memory during the same clock cycle

The Transfer Direction bit (MEMIO) of the DMA Channel Control Register (DMCCRn) specifies the transfer direction.

- From memory to an external I/O device (DMCCRn.MEMIO = “1”)

  External memory Read operation to an address specified by the DMA Source Address Register (DMSARn) is performed simultaneously to assertion of the DMAACK[n] signal.

- Single Address transfer from memory to an external I/O device (DMCCRn.MEMIO = “0”)

  External memory Write operation to an address specified by the DMA Source Address Register (DMSARn) is performed simultaneously to assertion of the DMAACK[n] signal. At this time, the external I/O device drives the DATA signal instead of the TX4925.

Special attention must be paid to the timing design when the bus clock frequency is high or when performing Burst transfer. Single Address transfer using Burst transfer with SDRAM is not recommended.

8.3.3.4 DMADONE* Signal

The DMADONE* signal operates as either the DMA stop request input signal or the DMA done signalling output signal, or may operate as both of these signals depending on the setting of the DONE Control Field (DNCTRL) of the DMA Channel Control Register (DMCCRn).

The DMADONE* signal is shared by four channels. The DMADONE* channel is valid for a channel when the DMAACK[n] signal for that channel is asserted.

If the DMADONE* channel is set to be used as an output signal (DMCCRn.DNCTRL = 10/11), it will operate as follows depending on the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn).

When the Chain End bit (CHDN) is set, the DMADONE* signal is only asserted when the DMAACK[n] signal for the last DMA transfer in the Link List Command Chain is asserted.

When the Chain End bit (CHDN) is cleared, the DMADONE* signal is asserted when the DMAACK[n] signal for the last data transfer in a DMA transfer specified by the current DMA Channel Register is asserted. Namely, if the Link List Command chain is used, there is one assertion at the end of each data transfer specified by each Descriptor.

If the DMADONE* signal is set to be used as an input signal (DMCCRn.DNCTRL = 01/11), DMA transfer can be set to end normally when the external device asserts the DMADONE* signal when the DMAACK[n] signal of channel *n* is asserted. DMADONE* is asserted during DMAACK[n] is not asserted, then unexpected operation occurs. When DMA transfer is terminated by the DMADONE* assertion of the external device, the External DONE Assert bit (DMCSRn.EXTDN) of the DMA Channel Status Register is set regardless of the setting of the Chain End bit (CHDN) of the DMA Channel Control Register (DMCCRn). Operation is as follows depending on the setting of the Chain End bit (CHDN).

When the Chain End bit (CHDN) is set, all DMA transfer for that chain is terminated. At this time, the Normal Chain End bit (NCHNC) and the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register are both set and the Transfer Active bit (DMCCRn.XFACT) of the DMA Channel Control Register is cleared.

When the Chain End bit (CHDN) is cleared, only DMA transfer specified by the current DMA Channel Register ends normally, and only the Normal Transfer End bit (NTRNFC) is set. When the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn) is set, chain transfer is executed and DMA transfer continues. When the Chain Enable bit (CHNEN) is cleared, the Transfer Active bit (DMCCRn.XFACT) is cleared and the Normal Chain End bit (NCHNC) is set.

Three clock cycles are required from external assertion of the DMADONE* signal to disabling of new DMA access. Operation will not stop even if the bus operation in progress is a Single transfer or a Burst transfer. For example, if the DMADONE* signal is asserted during Read operation of Dual Address transfer, the corresponding Write bus operation will also be executed.

If the DMADONE* pin is set to become both input and output for channel *n* (DMCCRn.DNCTRL = "11"), the DMADONE* signal becomes an open drain signal when the channel becomes active. When used by this mode, the DMADONE* signal must be pulled up by an external source. When in this mode, the External DONE Assert bit (DMCSRn.EXTDN) is not only set when asserted by an external device, but is also set when asserted by the TX4925.

8.3.4 Internal I/O DMA Transfer Mode

Performs DMA with the on-chip Serial I/O Controller and the AC-link Controller. Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 1: I/O DMA Transfer mode
- DMCCRn.SNGAD = 0: Dual Address Transfer

Refer to "8.3.8 Dual Address Transfer" and "11.3.6 DMA transfer (Serial I/O Controller)" or "14.3.6.4 DMA Operation (AC-link Controller)" for more information.

8.3.5 Memory-Memory Copy Mode

It is possible to copy memory from any particular address to any other particular address when in the Memory-Memory Copy mode.

Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 0: Memory Transfer mode
- DMCCRn.SNGAD = 0: Dual Address mode

Furthermore, when in the Memory-Memory Copy mode it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to "8.3.8 Dual Address Transfer" for information regarding the setting of other registers.

8.3.6 Memory Fill Transfer Mode

When in the Memory Fill Transfer mode, word data set in the DMA Memory Fill Data Register (DMMFDR) is written to the data region specified by the DMA Source Address Register (DMSARn). This data can be used for initializing the memory, etc.

Set the DMA Channel Control Register (DMCCRn) as follows.

- DMCCRn.EXTRQ = 0: Memory transfer mode
- DMCCRn.SNGAD = 1: Single Address Transfer
- DMCCRn.MEMIO = 0: Transfer from I/O to memory

In addition, when in the Memory Fill Transfer mode, it is possible to set the interval for requesting ownership of each bus using the Internal Request Delay field (INTRQD) of the DMA Channel Control Register (DMCCRn).

Refer to “8.3.7 Single Address Transfer” for information regarding the setting of other registers.

8.3.7 Single Address Transfer

This section explains register settings during Single Address transfer (DMCCRn.SNGAD = 1). This applies to the following DMA Transfer modes.

- External I/O (Single Address) Transfer
- Memory Fill Transfer

8.3.7.1 Channel Register Settings During Single Address Transfer

Table 8.3.2 shows restrictions of the Channel Register settings during Single Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set and DMA transfer is not performed.

For Burst transfer, +4, 0, or –4 can be set to the DMA Source Address Increment Register (DMSAIRn). Setting 0 is only possible during transfer from memory to external I/O. A Configuration Error will result if the value “0” is set during transfer from external I/O to memory or during Memory Fill transfer.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 2 bytes or larger, then a value will be set in the DMA Source Address Register (DMSARn) is as follows below.

- Transfer setting size: 2bytes, (DMSARn) that reflects the lower 1 bits.
- Transfer setting size: 4bytes, (DMSARn) that reflects the lower 2 bits.

During Single Address transfer, the DMA Destination Address Register (DMDARn) and DMA Destination Address Increment Register (DMDAIRn) settings are ignored.

Table 8.3.2 Channel Register Setting Restrictions During Single Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[1:0]		DMSAIRn[1:0]	DMCNTRn[1:0]
	DMSAIRn is "0" or greater	DMSAIRn setting is a negative value		
1 Byte	**	**	**	**
2 Bytes	*0	*1	*0	*0
4 Bytes	00	11	00	00
4 Words	00	11	4/0/-4	00
8 Words				
16 Words				
32 Words				

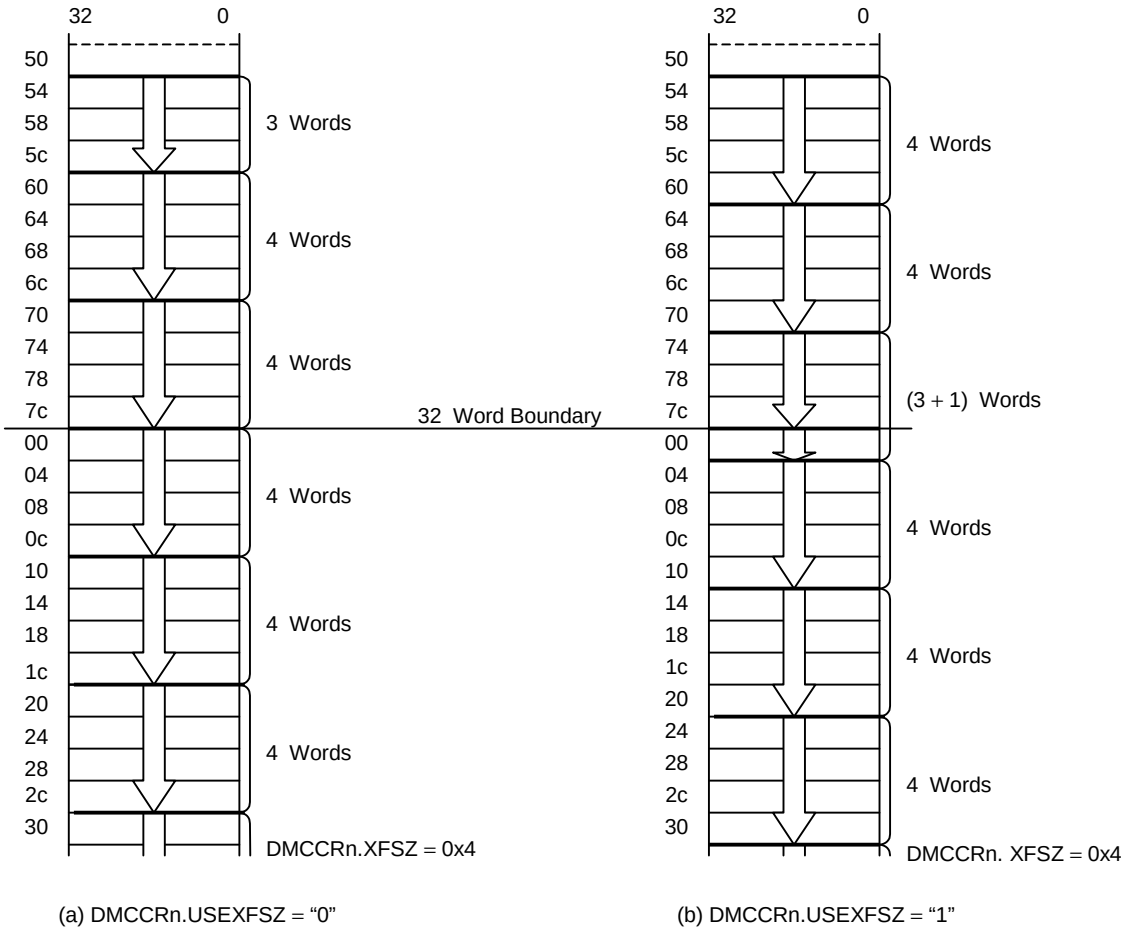
8.3.7.2 Burst Transfer During Single Address Transfer

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32 word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, the DMA Controller executes multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register.

Figure 8.3.2 shows the Single Address Burst transfer status when the lower 7 bits of the Transfer Start address are 0x54 and the transfer setting size (DMCCRn.XFSZ) is set to 4 words.

Panel (a) of this figure shows the situation when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is "0". In this case, first a three word transfer is performed up to the address aligned to the transfer setting size. Then, four word transfer specified by the transfer setting size is repeated. This setting is normally used.

On the other hand, panel (b) shows when the Transfer Size Mode bit (DMCCRn.USEXFSWZ) is "1". in this case, transfer is repeated according to the transfer setting size. Three word transfer and one word transfer is only performed consecutively without releasing bus ownership when transfer spans across a 32 word boundary.



(a) DMCCRn.USEXFSZ = “0” (b) DMCCRn.USEXFSZ = “1”

Figure 8.3.2 Non-aligned Single Address Burst Transfer

8.3.8 Dual Address Transfer

This section explains the register settings for Dual Address transfer (DMCCRn.SNGAD = 0). This applies to the following DMA transfer modes.

- External I/O (Dual Address) transfer
- Internal I/O DMA transfer
- Memory-Memory Copy transfer

8.3.8.1 Channel Register Settings During Dual Address Transfer

Table 8.3.3 shows restrictions of the Channel Register settings during Dual Address transfer. If these restrictions are not met, then a Configuration Error is detected, the Configuration Error bit (CFERR) of the DMA Channel Status Register (DMCSRn) is set, and DMA transfer is not performed.

If the setting of the DMA Source Address Increment Register (DMSAIRn) is negative and the transfer setting size is 4 bytes or larger, then a value will be set in the DMA Source Address Register (DMSARn) that reflects the lower 2 bits. Similarly, if the setting of the DMA Destination Address Increment Register (DMDAIRn) is negative and the transfer setting size is 4 bytes or larger, then a value will be set in the DMA Destination Address Register (DMDARn) that reflects the lower 2 bits.

Example: When the transfer address is 0x0001_0000, the DMA Source Address Register (DMSARn) is as follows below.

- DMSAIRn setting is “0” or greater: 0x0001_0000
- DMSAIRn setting is a negative value: 0x0001_0003

Table 8.3.3 Channel Register Setting Restrictions During Dual Address Transfer

Transfer Setting Size (DMCCRn.XFSZ)	DMSARn[1:0]		DMDARn[1:0]		DMSAIRn	DMDAIRn	DMCNTRn [1:0]	DMCCRn. REVBYTE
	DMSAIRn setting is 0 or greater	DMSAIRn setting is a negative value	DMDAIRn setting is 0 or greater	DMDAIRn setting is a negative value				
1 Byte	**	**	**	**	**	**	**	0
2 Bytes	*0	*1	*0	*1	*0	*0	*0	0
4 Bytes	00	11	00	11	00	00	00	0/1
4 / 8 Wods (DMMCR.FIFUM[n]=0)	00	11	00	11	00	00	00	0/1
4 / 8 Words (DMMCR.FIFUM[n]=1)	00	11	00	11	4/0/-4	4/-4 ‡	00	0/1
	**	-	**	-	4	4	**	0
	-	**	-	**	-4	-4		0
16 Words	Cannot be set (Configuration Error)							
32 Words	Cannot be set (Configuration Error)							

†: 4, 0 or -4 can be specified when Source Burst Inhibit bit (DMCCRn, SBINH) is set.

‡: 4, 0 or -4 can be specified when DestinationBurst Inhibit bit (DMCCRn.DBINH) is set.

8.3.8.2 Burst Transfer During Dual Address Transfer

The DMA Controller has a 32-bit 8-stage FIFO on-chip that is connected to the internal bus (G-Bus) for Burst transfer during Dual Address transfer. Since this FIFO employs a shifter, it is possible to perform transfer of any address or data size. Burst transfer is only performed when 4 Words or 8 Words is set by the Transfer Setting Size field (DMCCRn.XFSZ) and the FIFO Use Enable bit (DMMCRn.FIFUM[n]) of the DMA Master Control Register is set.

According to the SDRAM Controller and External Bus Controller specifications, the DMA Controller cannot perform Burst transfer that spans across 32 word boundaries. Consequently, if the address that starts DMA transfer is not a multiple of the transfer setting size (DMCCRn.XFSZ) (is not aligned), transfer cannot be performed by any of the transfer sizes that were specified by a Burst transfer. Therefore, it is necessary to divide the transfer into multiple Burst transactions of a transfer size smaller than the specified transfer size. This division method changes according to the seting of the Transfer Size Mode bit (DMCCRn.USEXFSZ) of the DMA Channel Control Register and whether or not the address offset relative to the Transfer Setting size (DMCCRn.XFSZ) is equivalent to the source address and destination address combined.

Figure 8.3.3 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to “1”, the lower 7 bits of the Transfer Start address for the transfer source are set to 0x54, the lower 7 bits of the Transfer Start address for the transfer destination are set to 0x1C, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 Words.

Transfer repeats according to the transfer setting size, regardless of the different address offsets. However, transfers that span across 32 word boundaries are divided. Since data remains in the on-chip FIFO when in this mode, it becomes possible to share the on-chip FIFO among multiple DMA channels.

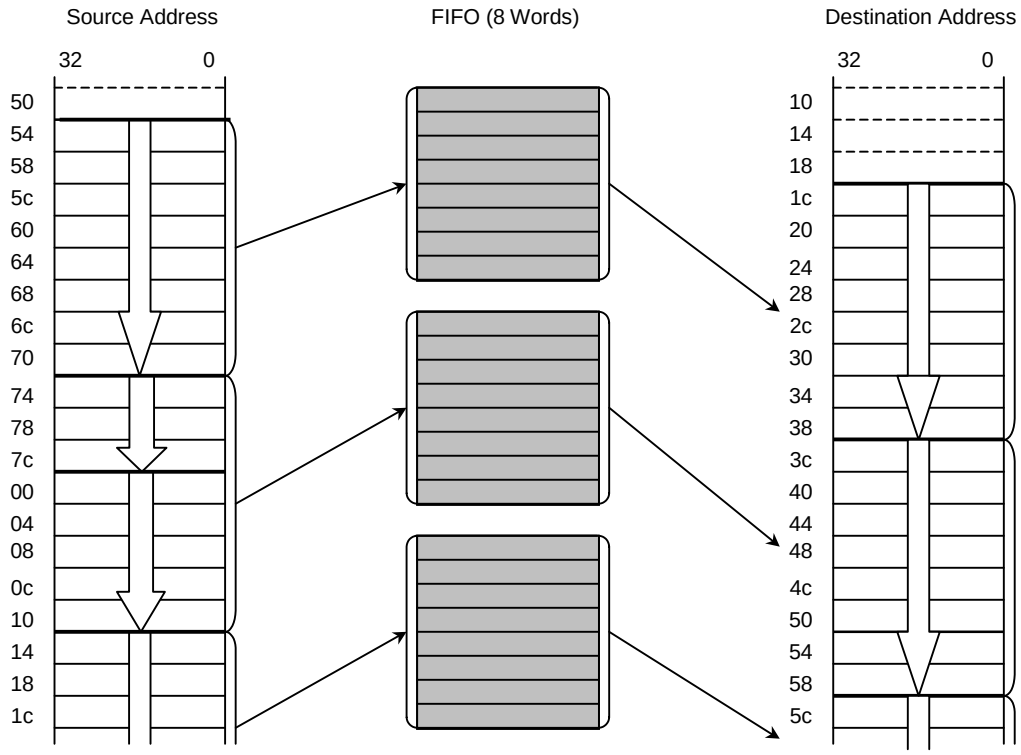


Figure 8.3.3 Dual Address Burst Transfer (DMCCRn.USEXFSZ = 1)

Figure 8.3.4 shows Dual Address Burst transfer when the Transfer Size Mode bit (DMCCRn.USEXFSZ) is set to "0", the lower 7 bits of the Transfer Start address for the transfer source are set to 0x54, the lower 7 bits of the Transfer Start address for the transfer destination are set to (a) 0x14/(b) 0x18, and the Transfer Setting Size (DMCCRn.XFSZ) is set to 8 words.

Panel (a) of this figure shows when the address offset is equivalent. In this case, first transfer of three words is performed up to the address that is aligned with the transfer setting size. Then, transfer of eight words that is specified by the transfer setting size is repeated.

On the other hand, panel (b) show when the address offset is not equivalent. In this case, first only data up to the address that is aligned with the transfer setting size is read to the on-chip FIFO. Then, data is written up to the address that is aligned with the transfer setting size as long as data remains in the on-chip FIFO. Efficiency decreases since the transfer size is divided. Also, since data may remain in the on-chip FIFO, Burst transfer of a Dual Address that uses the on-chip FIFO simultaneously with another channel cannot be performed.

Using the Burst Inhibit bit makes it possible to mix Burst transfer with 8-Word Single transfer. This in turn makes it possible to perform Burst access only for memory access during DMA transfer with external I/O devices that cannot perform Burst transfer.

When the Source Burst Inhibit bit (DMCCRn.SBINH) is set, data read from the Source Address to the on-chip FIFO is divided into multiple 4-byte Single Read transfers, then transfer is executed.

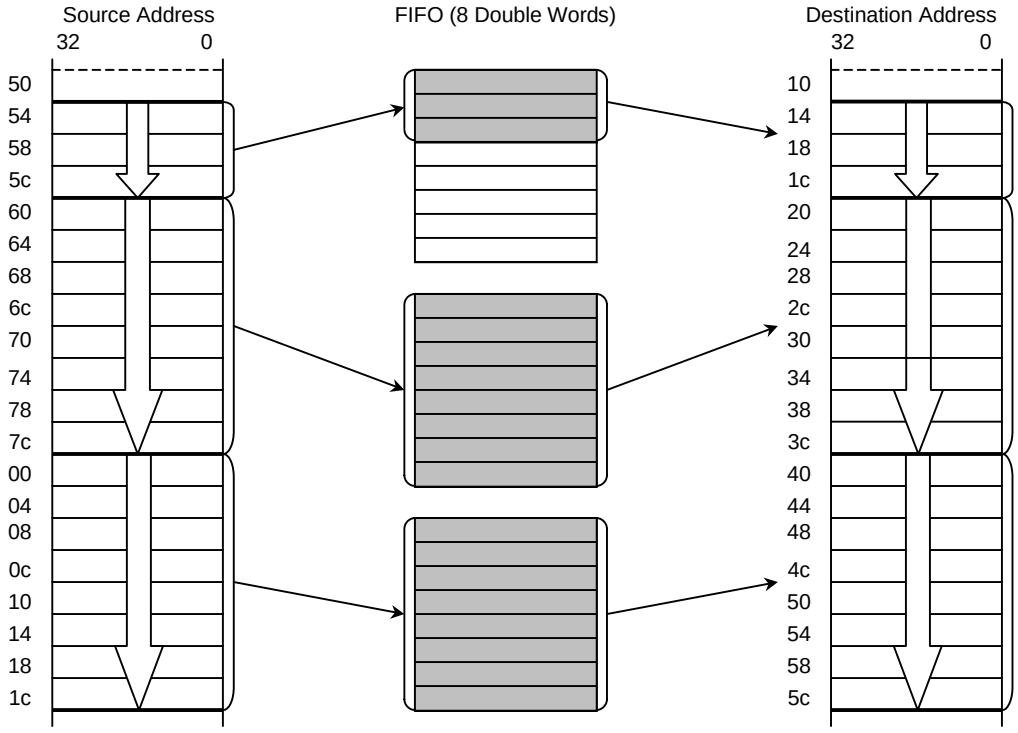
When the Destination Burst Inhibit bit (DMCCRn.DBINH) is set, data written from the FIFO to the Destination Address is divided into multiple 4-byte Single Write transfers, then transfer is executed.

Only 4, 0 and 4 can be set on Burst Inhibit bit during Burst transfer. When the Burst Inhibit bit is set, an any multiples of 4 can be set. (Refer to Table 8.3.3)

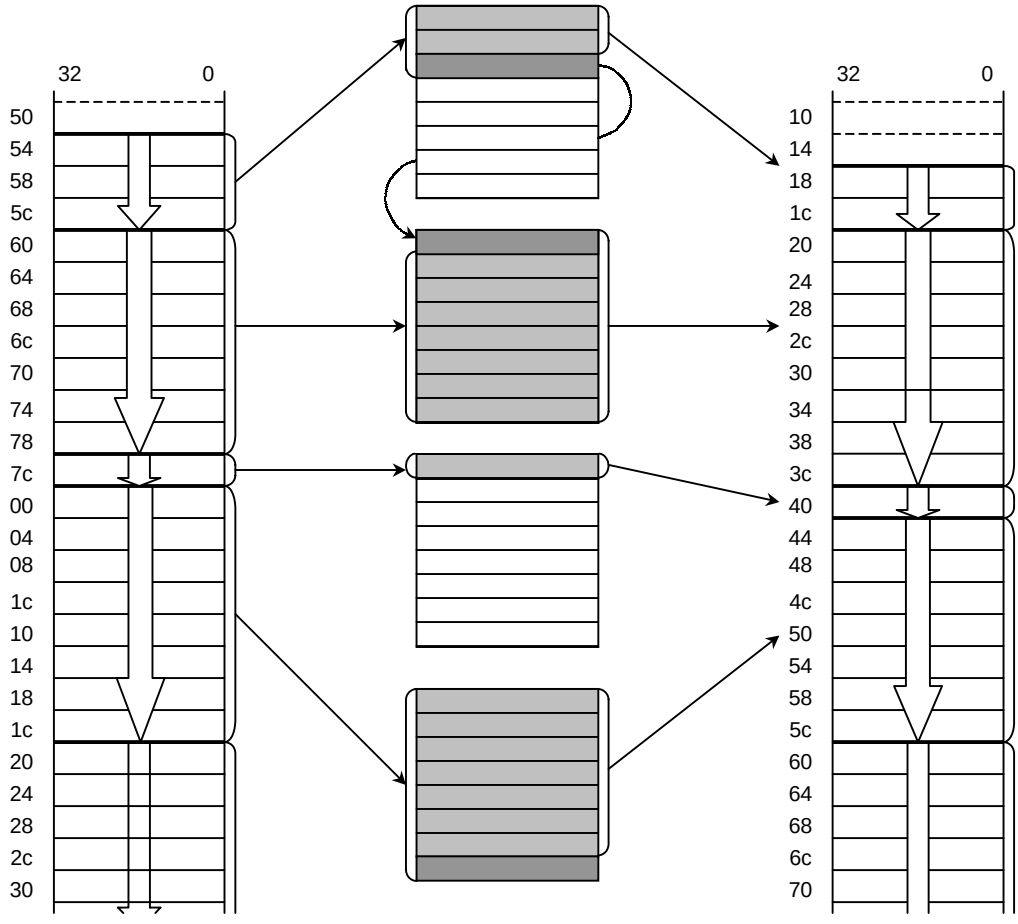
8.3.8.3 Double Word Byte Swapping

When the Reverse Byte bit (REVBYTE) of the DMA Channel Configuration Register (DMCCRn) is set, read word data is written after byte swapping is performed. For example, if the read data is “0x0123_4567”, then the data “0x6745_2301” is written.

The Reverse Byte bit can only be set when the REVBYTE column of Table 8.3.3 is set so “0/1” is indicated.



(a) Address offset is equivalent



(b) Address offset differs

Figure 8.3.4 Dual Address Burst Transfer (DMCCRn.USEXFSZ = 0)

8.3.9 DMA Transfer

The sequence of DMA transfer that uses only the DMA Channel Register is as follows below.

(1) Select DMA request signal
To perform external I/O or internal I/O DMA, set the DMA Request Select field of the DMA Request Control Register (DRQCTR. DMAREQ). For external I/O DMA, also program the function of the shared pin through the DMA Select field of the Pin Configuration Register (PCFG.SELDMA).

(2) Set the Master Enable bit
Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

(3) Set the Address Register and Count Register
Set the five following register values.

- DMA Source Address Register (DMSARn)
- DMA Destination Address Register (DMDARn)
- DMA Count Register (DMCNTRn)
- DMA Source Address Increment Register (DMSAIRn)
- DMA Destination Address Increment Register (DMDAIRn)

(4) Set Chain Address Register
Set “0” to the DMA Chain Address Register (DMCHARn).

(5) Clear the DMA Channel Status Register (DMCSRn)
Clear when status from the previous DMA transfer remains.

(6) Set the DMA Channel Control Register (DMCCRn)

(7) Initiate DMA transfer
DMA transfer is started by setting the Transfer Active bit (XFACT) of the DMA Channel Control Register.

(8) Signal completion
When DMA data transfer ends normally, set the Normal Transfer Complete bit (NTRNFC) of the DMA Channel Status Register (DMCSRn). An interrupt is signalled if the Transfer Complete Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set.

If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status Register and the transfer is interrupted. If the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set, then the interrupt is signaled.

8.3.10 Chain DMA Transfer

Table 8.3.4 shows the data structure in memory that the DMA Command Descriptor has. When the Simple Chain bit (SMPCHN) of the DMA Channel Control Register (DMCCRn) is set, only the initial four words are used. DMSAIRn, DMDAIR, DMCCRn, and DMCSRn use the settings from when DMA started. In addition, all eight words are used when the Simple Chain bit (SMPCHN) is cleared.

Saving the start memory address of another DMA Command Descriptor in the Offset 0 Chain Address field makes it possible to construct a chain list of DMA Command Descriptors (Figure 8.3.5). Set "0" in the Chain Address field of the DMA Command Descriptor at the end of the chain list.

When DMA transfer that is specified by one DMA Command Descriptor ends, the DMA Controller automatically reads the next DMA Command Descriptor indicated by the Chain Address Register (Chain transfer), then continues DMA transfer. Continuous DMA transfer that uses multiple Descriptors connected into such a chain-like structure is called Chain DMA transfer.

Since the DMA Channel Status Register is also overwritten during Chain transfer when the DMA Simple Chain bit (SMPCHN) is cleared, be sure not to unnecessarily clear necessary bits.

Placing DMA Command Descriptors at addresses that do not span across 32 word boundaries in memory is efficient since they are read by one G-Bus Burst Read operation.

Table 8.3.4 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	DMA Chain Address Register (DMCHARn)
0x04	Source Address	DMA Source Address Register (DMSARn)
0x08	Destination Address	DMA Destination Address Register (DMDARn)
0x0c	Count	DMA Count Register (DMCNTRn)
0x10	Source Address Increment	DMA Source Address Increment Register (DMSAIRn)
0x14	Destination Address Increment	DMA Destination Address Increment Register (DMDAIRn)
0x18	Channel Control	DMA Channel Control Register (DMCCRn)
0x1c	Channel Status	DMA Channel Status Register (DMCSRn)

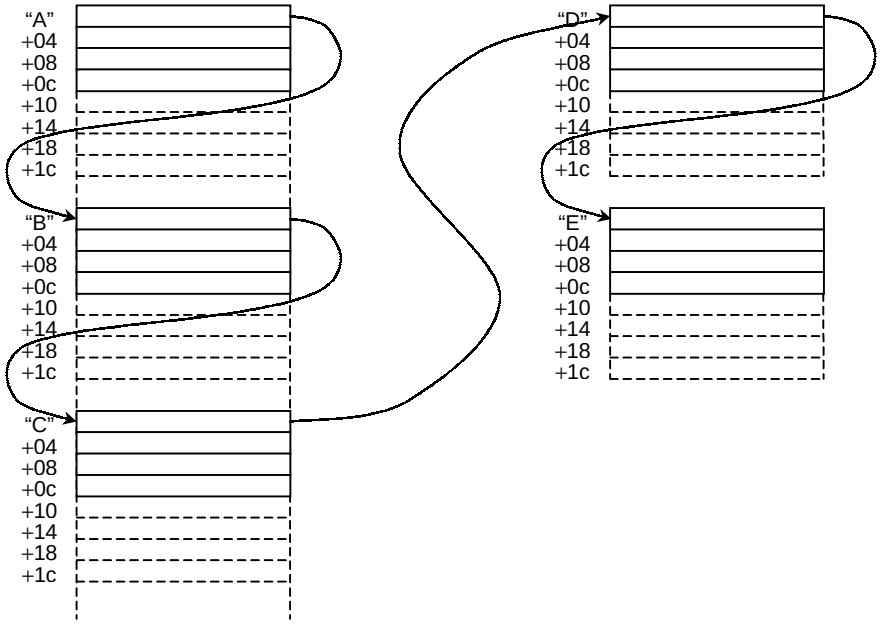


Figure 8.3.5 DMA Command Descriptor Chain

The sequence of Chain DMA transfer is as follows below.

(1) Select DMA request signal
To perform external I/O or internal I/O DMA, set the DMA Request Select field of the DMA Request Control Register (DRQCTR. DMAREQ). For external I/O DMA, also program the function of the shared pin through the DMA Select field of the Pin Configuration Register (PCFG.SELDMA).

(2) Set the Master Enable bit
Set the Master Enable bit (DMMCR.MSTEN) of the DMA Master Control Register.

(3) Structure of the DMA command Descriptor chain
Construct the DMA Command Descriptor Chain in memory.

(4) Set the Count Register
Set "0" to the DMA Count Register (DMCNTRn) .
Sets the DMA Source Address Increment Register (DMSAIRn) and DMA destination Address Increment Register (MMDAIRn).
Never set 0 or less than 0 for the increment value.

(5) Clear the DMA Channel Status Register (DMCSRn)
Clear the status of the previous DMA transfer.

(6) Set the DMA Channel Control Register (DMCCRn).

(7) Initiate DMA transfer
Setting the address of the DMA Command Descriptor at the beginning of the chain list in the DMA Chain Address Register (DMCHARn) automatically initiates DMA transfer. First, the value stored in each field of the DMA Command descriptor at the beginning of the Chain List is read to each corresponding DMA Channel register (Chain transfer), then DMA transfer is performed according to the read value.

When a value other than "0" is stored in the DMA Chain Address Register (DMCHARn), data of the size stored in the DMA Count Register (DMCNTRn) is completely transferred, then the DMA Command Descriptor value of the memory address specified by the DMA Chain Address Register is read.

In addition, if the Chain Address field value read the Descriptor 0, the DMA Chain Address Register value is not updated. All previous values (Data Command Descriptor Addresses with the value "0" in the Chain Address field when the values were read) are held.

(8) Signal completion
Set the Normal Chain End bit (NCHNC) of the DMA Channel Status Register (DMCSRn) when DMA data transfer of all Descriptor Chains is complete. An interrupt is signalled if the Chain End Interrupt Enable bit (INTENC) of the DMA Channel Control Register (DMCCRn) is set at this time.

In addition, the Normal Transfer End bit (NTRNFC) of the DMA Channel Status Register (DMCSRn) is set each time DMA data transfer specified by each DMA Command Descriptor ends normally. An interrupt is signalled if the Transfer End Interrupt Enable bit (INTENT) of the DMA Channel Control Register (DMCCRn) is set at this time.

If an error is detected during DMA transfer, the error cause is recorded in the lower four bits of the DMA Channel Status register and transfer is interrupted. An interrupt is signalled if the Error Interrupt Enable bit (INTENE) of the DMA Channel Control Register is set.

8.3.11 Dynamic Chain Operation

It is possible to add DMA Command Descriptor chains to the DMA Command Descriptor chain while Chain DMA transfer is in progress. This is performed according to the following procedure. This procedure is available only when the value of the last descriptor DMSAIRn/DMDAIRn in the last command descriptor chain is one or more than one byte.

(1) Construct the DMA Command Descriptor chain
Construct the DMA Command Descriptor chain to be added to memory.

(2) Add a DMA Command Descriptor chain
Substitute the address of the Command Descriptor at the beginning of the Descriptor Chain to be added into the Chain Address field of the Descriptor at the end of the DMA Command Descriptor chain that is currently performing DMA transfer.

(3) Check the Chain Enable bit
Read the value of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn). If that value is “0”, then write the Chain Address field value of the DMA Command Descriptor that is indicated by the address stored in the DMA Chain Address Register (DMCHARn).

8.3.12 Interrupts

An interrupt number (10 – 13) of the Interrupt Controller is mapped to each channel. In addition, there are completion interrupts for when transfer ends normally and error interrupts for when transfer ends abnormally for each channel. When an interrupt occurs, then the bit that corresponds to either the Normal Interrupt Status field (DIS[3:0]) or the Error Interrupt Status field (EIS[3:0]) of the DMA Master Control Register (DMMCR) is set.

Figure 8.3.6 shows the relationship between the Status bit and Interrupt Enable bit for each interrupt cause. Refer to the explanation for each Status bit for more information regarding each information cause.

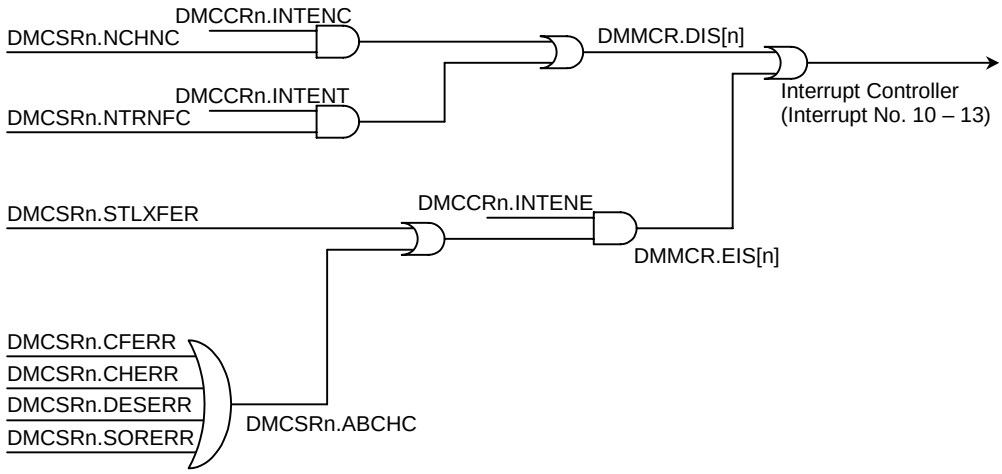


Figure 8.3.6 DMA Controller Interrupt Signal

8.3.13 Transfer Stall Detection Function

If the period from when a certain channel last performs internal bus access to when the next internal bus access is performed exceeds the Transfer Stall Detection Interval field (STLTIME) of the DMA Channel Control Register (DMCCRn), the Transfer Stall Detection bit (STLXFER) of the DMA Channel Status Register (DMCSRn) is set. An error interrupt is signalled if the Error Interrupt Enable bit (DMCCRn.INTENE) is set.

In contrast to other error interrupts, DMA transfer is not stopped. Normal DMA transfer is executed if bus ownership can be obtained. Furthermore, clearing the Transfer Stall Detection (STLXFER) resumes transfer stall detection as well.

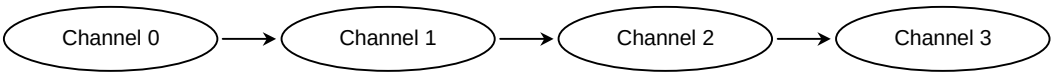
Setting the Transfer Stall Detection Interval field (STLTIME) to “000” disables the Transfer Stall Detection function.

8.3.14 Arbitration Among DMA Channels

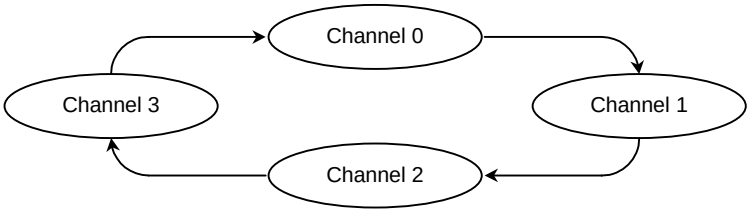
The DMA Controller has an on-chip DMA Channel Arbiter that arbitrates bus ownership among four DMA channels that use the internal bus (G-Bus). There are two methods for determining priority: the round robin method and the fixed priority method. (See Figure 8.3.7.) The Round Robin Priority bit (RRPT) of the DMA Master Control Register (DMMCR) selects the priority method.

- Fixed priority (DMMCR.RRPT = 0)
  As shown below, Channel 0 has the highest priority and Channel 3 has the lowest priority.
  CH0 > CH1 > CH2 > CH3

- Round Robin method (DMMCR.RRPT = 1)
  The last channel to perform DMA transfer has the lowest priority.
  - After CH0 DMA transfer execution: CH1 > CH2 > CH3 > CH0
  - After CH1 DMA transfer execution: CH2 > CH3 > CH0 > CH1
  - After CH2 DMA transfer execution: CH3 > CH0 > CH1 > CH2
  - After CH3 DMA transfer execution: CH0 > CH1 > CH2 > CH3



a) Fixed Priority is selected



b) Round Robin Priority is selected

Figure 8.3.7 DMA Channel Arbitration

8.3.15 Restrictions in Access to PCI Bus

The PCI Controller detects a bus error if the DMA Controller performs one of the following accesses to the PCI Bus.

- Burst transfer exceeding 8 words (PCICSTATUS.TLB)
- Address Increment value –4 Burst transfer (PCICSTATUS.NIB)
- Address Increment Value 0 Burst transfer (PCICSTATUS.ZIB)
- Dual Address Burst transfer when the setting for DMSARn, DMDARn, or DMCNTRn is not a word boundary (PCICSTATUS.IAA)

In addition, Single Address transfers between an external I/O device and the PCI Bus are not supported. Data transfer is not performed, but no error is detected.

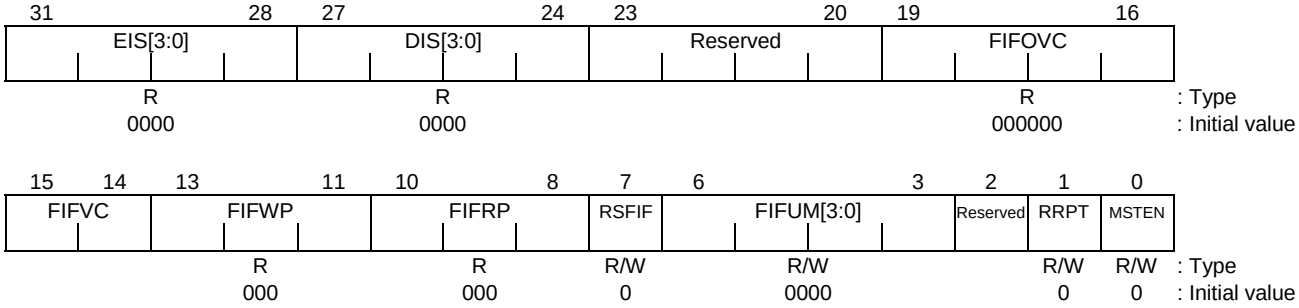
8.4 Registers

Table 8.4.1 DMA Controller Registers

Reference	Offset Address	Bit Width	Mnemonic	Register Name
8.4.6	0xB000	32	DMCHAR0	DMA Chain Address Register 0
8.4.4	0xB004	32	DMSAR0	DMA Source Address Register 0
8.4.5	0xB008	32	DMDAR0	DMA Destination Address Register 0
8.4.9	0xB00C	32	DMCNTR0	DMA Count Register 0
8.4.7	0xB010	32	DMSAIR0	DMA Source Address Increment Register 0
8.4.8	0xB014	32	DMDAIR0	DMA Destination Address Increment Register 0
8.4.2	0xB018	32	DMCCR0	DMA Channel Control Register 0
8.4.3	0xB01C	32	DMCSR0	DMA Channel Status Register 0
8.4.6	0xB020	32	DMCHAR1	DMA Chain Address Register 1
8.4.4	0xB024	32	DMSAR1	DMA Source Address Register 1
8.4.5	0xB028	32	DMDAR1	DMA Destination Address Register 1
8.4.9	0xB02C	32	DMCNTR1	DMA Count Register 1
8.4.7	0xB030	32	DMSAIR1	DMA Source Address Increment Register 1
8.4.8	0xB034	32	DMDAIR1	DMA Destination Address Increment Register 1
8.4.2	0xB038	32	DMCCR1	DMA Channel Control Register 1
8.4.3	0xB03C	32	DMCSR1	DMA Channel Status Register 1
8.4.6	0xB040	32	DMCHAR2	DMA Chain Address Register 2
8.4.4	0xB044	32	DMSAR2	DMA Source Address Register 2
8.4.5	0xB048	32	DMDAR2	DMA Destination Address Register 2
8.4.9	0xB04C	32	DMCNTR2	DMA Count Register 2
8.4.7	0xB050	32	DMSAIR2	DMA Source Address Increment Register 2
8.4.8	0xB054	32	DMDAIR2	DMA Destination Address Increment Register 2
8.4.2	0xB058	32	DMCCR2	DMA Channel Control Register 2
8.4.3	0xB05C	32	DMCSR2	DMA Channel Status Register 2
8.4.6	0xB060	32	DMCHAR3	DMA Chain Address Register 3
8.4.4	0xB064	32	DMSAR3	DMA Source Address Register 3
8.4.5	0xB068	32	DMDAR3	DMA Destination Address Register 3
8.4.9	0xB06C	32	DMCNTR3	DMA Count Register 3
8.4.7	0xB070	32	DMSAIR3	DMA Source Address Increment Register 3
8.4.8	0xB074	32	DMDAIR3	DMA Destination Address Increment Register 3
8.4.2	0xB078	32	DMCCR3	DMA Channel Control Register 3
8.4.3	0xB07C	32	DMCSR3	DMA Channel Status Register 3
8.4.10	0xB0A4	32	DMMFDR	DMA Memory Fill Data Register
8.4.1	0xB0A8	32	DMMCR	DMA Master Control Register

8.4.1 DMA Master Control Register (DMMCR) 0xB0A8

This register controls the entire DMA Controller.



Bits	Mnemonic	Field Name	Description
31:28	EIS[3:0]	Error Interrupt Status	Error Interrupt Status [3:0] (Initial value: 0x0, R) These four bits indicate the error interrupt status of each channel. EIS[n] corresponds to channel *n*. 1: There is an error interrupt in the corresponding channel. 0: There is no error interrupt in the corresponding channel.
27:24	DIS[3:0]	Normal Completion Interrupt Status	Done Interrupt Status [3:0] (Initial value: 0x0, R) These four bits indicate the transfer completion (transfer complete or chain ended) interrupt status of each channel. DIS[n] corresponds to channel *n*. 1: There is a transfer completion interrupt in the corresponding channel. 0: There is no transfer completion interrupt in the corresponding channel.
23:20	—	Reserved	—
19:14	FIFVC	FIFO Valid Entry Count	FIFO Valid Entry Count (Initial value: 000000, R) These read only bits indicate the byte count of data that were written to FIFO but not read out from the FIFO.
13:11	FIFWP	FIFO Write Pointer	FIFO Write Pointer (Initial value: 000, R) These read only bits indicate the next write position in FIFO. This is a diagnostic function.
10:8	FIFRP	FIFO Read Pointer	FIFO Read Pointer (Initial value: 000, R) These read only bits indicate the next read position in FIFO. This is a diagnostic function.
7	RSFIF	Reset FIFO	Reset FIFO (Initial value: 0, R/W) This bit is used for resetting FIFO. When this bit is set to “1”, the FIFO read pointer, FIFO write pointer and FIFO valid entry count are initialized to “0”. If an error occurs during DMA transfer, use this bit when data remains in the FIFO (when the FIFO Valid entry Count Field is not “0”) to initialize the FIFO.

Figure 8.4.1 DMA Master Control Register (1/2)

Bit	Mnemonic	Field Name	Description
6:3	FIFUM[3:0]	FIFO Use Enable [3:0]	FIFO Use Enable [3:0] (Initial value: 0x0, R/W) Each channel specifies whether to use 8 -word FIFO in Dual Address transfer. FIFUM[n] corresponds to channel *n*. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information.
2	—	Reserved	—
1	RRPT	Round Robin Priority	Round Robin Priority (Initial value: 0, R/W) Specifies the method for determining priority among channels. 1: Round Robin method. Priority of the last channel used is the lowest, and the next previous channel has the next lowest priority. Round robin is in the order Channel 0 > Channel 1 > Channel > Channel 3. 0: Fixed Priority. Priority is fixed in the order Channel 0 > Channel 1 > Channel 2 > Channel 3.
0	MSTEN	Master Enable	Master Enable (Initial value: 0, R/W) This bit enables the DMA Controller. 1: Enable 0: Disable Note: If the entire DMA Controller is disabled, then all internal logic including the Bus Interface Logic and State Machine are reset.

Figure 8.4.1 DMA Master Control Register (2/2)

8.4.2 DMA Channel Control Register (DMCCRn) 0xB018 (ch. 0) 0xB038 (ch. 1) 0xB058 (ch. 2) 0xB078 (ch. 3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved		IMMCHN	USEXFSZ	LE	DBINH	SBINH	CHRST	RVBYTE	ACKPOL	REQPL	EGREQ	CHDN	DNCTL		EXTRQ	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	: Type
		0	0	-	0	0	1	0	0	0	0	0	00		0	: Initial value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
STLTIME/INTRQD			INTENE	INTENC	INTENT	CHNEN	XFACT	Reserved		SMPCHN	XFSZ			MEMIO	SNGAD	
R/W			R/W	R/W	R/W	R	R/W	R/W		R/W	R/W			R/W	R/W	: Type
000			0	0	0	0	0	00		0	000			0	0	: Initial value

Bit	Mnemonic	Field Name	Description
31:30	—	Reserved	—
29	IMMCHN	Immediate Chain	Immediate Chain (Initial value: 0, R/W) Selects the control method of bus ownership during chain transfer. 1: When the DMA transfer completes due to the current DMA Channel Register and DMCCRn.CHNEN=1, DMA Command Descriptor of the address set in DMCHARn is loaded to DMA channel Register without bus ownership release. (Chain Transfer) 0: When the DMA transfer completes due to the current DMA Channel Register and DMCCRn.CHNEN=1, DMA controller once releases the bus ownership. After that it gets bus ownership again and starts Chain Transfer. Note: It is not concerned with the setup of this bit but DMA controller releases bus ownership after the Chain Transfer ends.
28	USEXFSZ	Transfer Set Size Mode	Use Transfer Set Size (Initial value: 0, R/W) Selects the DMA channel operation mode during Burst DMA transfer. Refer to “8.3.7.2 Burst Transfer During Single Address Transfer” and “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. 1: The DMA Controller always transfers the amount of data set in DMCCRn.XFSZ for each bus operation. Since alignment to the boundary of the DMCCRn.XFSZ in the address is not forced when in this mode, transfers that exceed 32 -word boundaries are divided into two operations. 0: The DMA Controller calculates the transfer size so the address set in DMSARn and DMDARn (only during Dual Address transfer) can be aligned to the boundary of the size set in DMCCRn.XFSZ, then transfers data according to that size.
27	LE	Little Endian	Little Endian (Default: value that is the opposite of the G-Bus Endian (CCFG.ENDIAN)) This bit sets the Endian of the channel. 1: Channel operates in the Little Endian mode 0: Channel operates in the Big Endian mode

Figure 8.4.2 DMA Channel Control Register (1/4)

Bits	Mnemonic	Field Name	Description
26	DBINH	Destination Burst Inhibit	Destination Burst Inhibit (Initial value: 0, R/W) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Write cycle to the address set from FIFO to DMDARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.
25	SBINH	Source Burst Inhibit	Source Burst Inhibit (Initial value: 0, R/W) During Dual Address transfer, this bit sets whether to perform Burst transfer or Single transfer on a Read cycle to the FIFO from the address set to DMSARn when Burst transfer is set by DMCCRn.XFSZ. Refer to “8.3.8.2 Burst Transfer During Dual Address Transfer” for more information. The settings of this bit have no effect during Single Address transfers. 1: Multiple Single transfers are executed. 0: Burst transfer is executed.
24	CHRST	Channel Reset	Channel Reset (Initial value: 1, R/W) This bit is used for initializing channels. The DMCCRn.XFACT, DMCCRn.CHNEN, and DMCSRn bits are all cleared. In addition, all channel logic and interrupts from channels are cleared and bus ownership requests to the DMA Channel Arbiter are also reset. The software must clear this bit before operating a channel. 1: Reset channel 0: Enable channel
23	REVBYTE	Reverse Byte	Reverse Bytes (Initial value: 0, R/W) This bit specifies whether to reverse the byte order during a Dual Address transfer when the Transfer Setting Size field (DMCCRn.XFSZ) setting is 4 bytes or more. Refer to “0 Double Word Byte Swapping” for more information. 1: Reverses the byte order. 0: Does not reverse the byte order.
22	ACKPOL	Acknowledge Polarity	Acknowledge Polarity (Initial value: 0, R/W) Specifies the polarity of the DMAACK[n] signal. 1: Asserts when the DMAACK[n] signal is High 0: Asserts when the DMAACK[n] signal is Low
21	REQPL	Request Polarity	Request Polarity (Initial value: 0, R/W) Specifies the polarity of the DMAREQ[n] signal. 1: Asserts when the DMAREQ[n] signal is High. 0: Asserts when the DMAREQ[n] signal is Low.
20	EGREQ	Edge Request	Edge Request (Initial value: 0, R/W) Specifies the method for detecting DMA requests by the DMAREQ[n] signal. 1: DMAREQ[n] signal is Edge Detect. 0: DMAREQ[n] signal is Level Detect.
19	CHDN	Chain Complete	Chain Done (Initial value: 0, R/W) Selects control by the DMADONE* signal. See “8.3.3.4 DMA Controller” for more information. 1: Assertion of the DMADONE* signal controls the overall Chain DMA transfer. 0: Assertion of the DMADONE* signal controls DMA transfer according to the DMA Channel Register setting at that time.
18:17	DNCTL	DONE Control	Done Control (Initial value: 00, R/W) Specifies the input/output mode of the DMADONE* signal. Refer to “8.3.3.4 DMADONE* Signal” for more information. 00: DMADONE* signal becomes the input signal, but input is ignored. 01: DMADONE* signal becomes the input signal. 10: DMADONE* signal becomes the output signal. 11: DMADONE* signal becomes the open drain input/output signal.

Figure 8.4.2 DMA Channel Control Register (2/4)

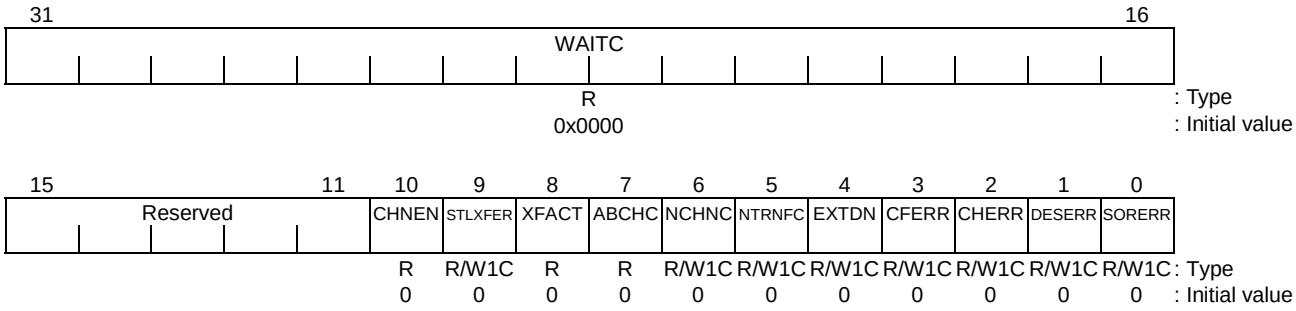
Bits	Mnemonic	Field Name	Description
16	EXTRQ	External Request	External Request (Initial value: 0, R/W) Sets the Request Transfer mode. 1: I/O DMA transfer mode This bit is used by the External I/O DMA Transfer mode and the Internal I/O DMA Transfer mode. A channel requests internal bus ownership when the I/O device asserts the DMA request signal. 0: Memory Transfer mode This bit is used by the Memory-Memory Copy Transfer mode and the Memory Fill Transfer mode. A channel requests internal bus ownership when the value of DMCSRn.WAITC becomes “0”.
15:13	STLTIME / INTRQD	Transfer Stall Detection Interval/Internal Request Delay	• When in the I/O DMA Transfer mode (DMCCRn.EXTRQ is “1”) Stalled Transfer Detect Time (Initial value: 000, R/W) Sets the detection interval for a lack of bus ownership. If this channel *n* releases bus ownership then the interval it does not have ownership exceeds the clock count set by this field, then DMCSRn.STLXFER is set to “1”. Refer to “8.3.13 Transfer Stall Detection Function” for more information. 000: Does not detect stalled transfers. 001: Sets 960 (15 × 64) clocks as the detection interval 010: Sets 4032 (63 × 64) clocks as the detection interval 011: Sets 16320 (255 × 64) clocks as the detection interval 100: Sets 65472 (1023 × 64) clocks as the detection interval 101: Sets 262080 (4095 × 64) clocks as the detection interval 110: Sets 1048512 (16383 × 64) clocks as the detection interval 111: Sets 4194240 (65535 × 64) clocks as the detection interval • When in the Memory Transfer mode (DMCCRn.EXTRQ is “0”) Internal Request Delay (Initial value: 000, R/W) Sets the delay time from when bus ownership is released to the next bus ownership request. Bus ownership is released, the set delay time elapses, then a bus ownership request is generated from the channel. 000: Always requests bus ownership when this channel is active. (Bus ownership is released after bus operation ends) 001: Set 16 clocks as the delay time 010: Set 32 clocks as the delay time 011: Set 64 clocks as the delay time 100: Set 128 clocks as the delay time 101: Set 256 clocks as the delay time 110: Set 512 clocks as the delay time 111: Set 1024 clocks as the delay time
12	INTENE	Error Interrupt Enable	Interrupt Enable on Error (Initial value: 0, R/W) Enables interrupts when the Error End bit (DMCSRn.ABCHC) or the Transfer Stall Detection bit (DMCSRn.STLXFER) is set. 1: Generates interrupts. 0: Does not generate interrupts.
11	INTENC	Chain End Interrupt Enable	Interrupt Enable on Chain Done (Initial value: 0, R/W) This bit enables interrupts when the Chain End bit (DMCSRn.NCHNC) is set. 1: Generate interrupts. 0: Do not generate interrupts.
10	INTENT	Transfer End Interrupt Enable	Interrupt Enable on Transfer Done (Initial value: 0, R/W) This bit enables interrupts when the Transfer End bit (DMCSRn.NTRNFC) is set. 1: Generate interrupts. 0: Do not generate interrupts.

Figure 8.4.2 DMA Channel Control Register (3/4)

Bits	Mnemonic	Field Name	Description
9	CHNEN	Chain Enable	Chain Enable (Initial value: 0, R) This bit indicates whether Chain operation is being performed. Read Only. This bit is cleared when either the Master Enable bit (DMMCR.MSTEN) is cleared or the Channel Reset bit (DMCCRn.CHRST) is set. This bit is set if a value other than "0" is set when the CPU writes to the DMA Chain Address Register (DMCHARn) or when a Chain transfer writes DMA Command Descriptor. This bit is then cleared when "0" is set to the DMA Chain Address Register (DMCHARn). 1: If transfer completes due to the current DMA Channel Register setting, a DMA Command Descriptor is loaded in the DMA Channel Register from the specified DMA Chain Address Register (DMCHARn) address, then DMA transfer continues. 0: Further transfer does not start even if transfer completes due to the current DMA Channel Register setting.
8	XFACT	Transfer Active	Transfer Active (Initial value: 0, R/W) DMA transfer is performed according to the DMA Channel Register setting when this bit is set. This bit is automatically set when a value other than "0" is set in the DMA Chain Address Register (DMCHARn). DMA transfer is then initiated. This bit is automatically cleared either when DMA transfer ends normally it is stopped due to an error. 1: Perform DMA transfer. 0: Do not perform DMA transfer.
7:6	—	Reserved	Please write "00" (Initial value: 00, R/W).
5	SMPCHN	Simple Chain	Simple Chain (Initial value: 0, R/W) This bit selects the DMA Channel Register that loads data from DMA Command Descriptors during Chain DMA transfer. 1: Data is only loaded to the four following DMA Channel Registers: the Chain Address Register (DMCHARn), the Source Address Register (DMSARn), the Destination Address Register (DMDARn), and the Count Register (DMCNTRn). 0: Data is loaded to all eight DMA Channel Registers.
4:2	XFSZ	Transfer Set Size	Transfer Set Size (Initial value: 000, R/W) These bits set the transfer data size of each bus operation in the internal bus. When the transfer set size is set to four words or greater, the data size actually transferred during a single bus operation does not always match the transfer set size. Refer to "8.3.7.2 Burst Transfer During Single Address Transfer" and "8.3.8.2 Burst Transfer During Dual Address Transfer" for more information. 000: 1 byte 001: 2 bytes 010: 4 bytes 011: (Reserved) 100: 4 words 101: 8 words 110: 16 words (Single Address transfer only) 111: 32 words (Single Address transfer only)
1	MEMIO	Memory to I/O	Memory to I/O (Initial value: 0, R/W) This bit specifies the transfer direction during Single Address transfer (DMCCRn.SNGAD = 1). Clear this bit when in the Memory Fill Transfer mode. The setting of this bit is ignored when Dual Address transfer is set (DMCCRn.SNGAD = 0). 1: From memory to I/O 0: From I/O to memory
0	SNGAD	Single Address	Single Address (Initial value: 0, R/W) This bit specifies whether the transfer method is Single Address transfer or Dual Address transfer. 1: Single Address transfer 0: Dual Address transfer

Figure 8.4.2 DMA Channel Control Register (4/4)

8.4.3 DMA Channel Status Register (DMCSRn) 0xB01C (ch. 0) 0xB03C (ch. 1) 0xB05C (ch. 2) 0xB07C (ch. 3)



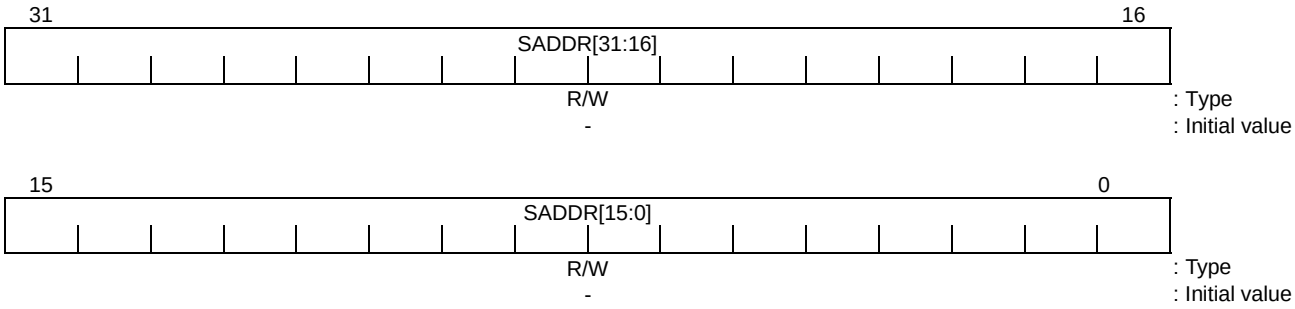
Bits	Mnemonic	Field Name	Description
31:16	WAITC	Wait Counter	Wait Counter (Initial value: 0x0000, R) This is a diagnostic function. I/O DMA transfer mode (DMCCRn.EXTRQ = “1”) This counter is decremented by 1 at each 64 G-Bus cycles. After channel *n* releases bus ownership, this counter sets the default (the value that is the detection interval clock cycle count set by the Transfer Stall Detection Interval field (DMCCRn.STLTIME) divided by 64). The Transfer Stall Detect bit (DMCSRn.STLXFER) is set when the interval during which bus ownership is not held reaches the set clock cycle. The counter is reset to the default and stops counting. Clearing the Transfer Stall Detect bit (DMCSRn.STLXFER) resumes the count and starts stall detection. Memory transfer mode (DMCCRn.EXTRQ = “0”) This counter is decremented by 1 at each G-Bus cycle. After bus ownership is released, the counter is set to the delay clock cycle count set by the Internal Request Delay field (DMCCRn.INTRQD). When the counter reaches “0” the count stops and channel *n* requests bus ownership.
15:11	—	Reserved	—
10	CHNEN	Chain Enable	Chain Enable (Initial value: 0, R) This value is a copy of the Chain Enable bit (CHNEN) of the DMA Channel Control Register (DMCCRn).
9	STLXFER	Transfer Stall Detect	Stalled Transfer Detect (Initial value: 0, R/W1C) This bit indicates whether the interval during which bus ownership is not held exceeds the value set by the Transfer Stall Detect Interval field (DMCCRn.STLTIME) after bus ownership is released when in the I/O DMA transfer mode. 1: Indicates that the interval during which bus ownership was not held exceeds the DMCCRn.STLTIME setting. 0: The interval during which bus ownership was not held did not exceed the setting since this bit was last cleared.

Figure 8.4.3 DMA Channel Status Register (1/2)

Bits	Mnemonic	Field Name	Description
8	XFACT	Transfer Active	Transfer Active (Initial value: 0, R) This value is a copy of the Transfer Active bit (XFACT) of the DMA Channel Control Register (DMCCRn).
7	ABCHC	Error Complete	Error Completion (Initial value: 0, R) This bit indicates whether an error occurred during DMA transfer. This bit indicates the logical sum of the four error bits (CFERR, CHERR, DESERR, SORERR) in DMCSRn[3:0]. 1: DMA transfer ends due to an error. 0: No error occurred since this bit was last cleared.
6	NCHNC	Chain Complete	Normal Chain Completion (Initial value: 0, R/W1C) When performing chain DMA transfer, This bit indicates whether all DMA data transfers in the DMA Descriptor chain are complete. 1: All DMA data transfers in the DMA Descriptor chain ended normally. Or, DMA transfer that did not use a DMA Descriptor chain ended normally. 0: DMA transfer has not ended normally since this bit was last cleared.
5	NTRNFC	Transfer Complete	Normal Transfer Completion (Initial value: 0, R/W1C) This bit indicates whether DMA transfer ended according to the current DMA Channel Register setting. 1: DMA transfer ended normally. 0: DMA transfer has not ended since this bit was last cleared.
4	EXTDN	External DONE Asserted	External Done Asserted (Initial value: 0, R/W1C) This bit indicates whether an external I/O device asserted the DMADONE* signal. When the DMADONE* signal is set to bidirectional, this bit is also set when the TX4925 asserts the DMADONE* signal. 1: DMADONE* signal was asserted. 0: DMADONE* signal was not asserted.
3	CFERR	Configuration Error	Configuration Error (Initial value: 0, R/W1C) Indicates whether an illegal register setting was made. 1: There was a configuration error. 0: There was no configuration error.
2	CHERR	Chain Bus Error	Chain Bus Error (Initial value: 0, R/W1C) This bit indicates whether a bus error occurred while reading a DMA Command Descriptor. 1: Bus error occurred. 0: No bus error occurred.
1	DESERR	Destination Error	Destination Bus Error (Initial value: 0, R/W1C) This bit indicates whether a bus error occurred during a destination bus Write operation (a Write to a set DMDARn address). 1: Bus error occurred. 0: No bus error occurred.
0	SORERR	Source Bus Error	Source Bus Error (Initial value: 0, R/W1C) This bit indicates whether a bus error occurred during either a source bus Read or Write operation (A Read or Write to a set DMSARn address). 1: Bus error occurred. 0: No bus error occurred.

Figure 8.4.3 DMA Channel Status Register (2/2)

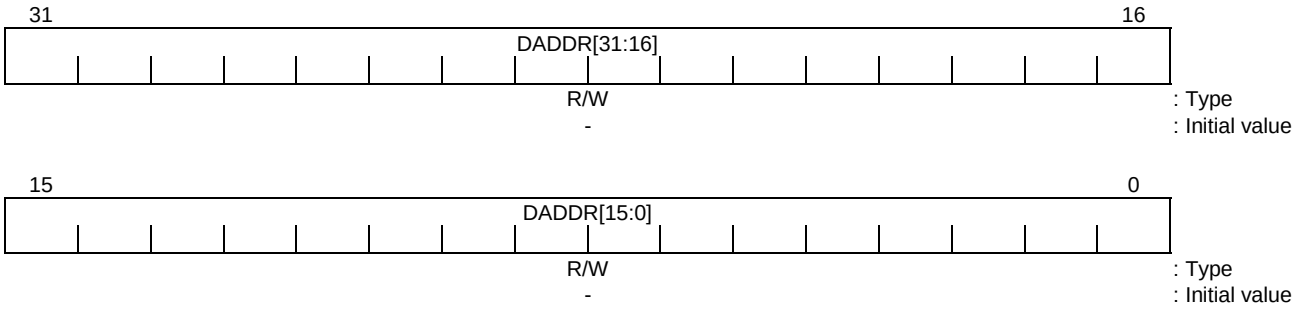
8.4.4 DMA Source Address Register (DMSARn) 0xB004 (ch. 0) 0xB024 (ch. 1) 0xB044 (ch. 2) 0xB064 (ch. 3)



Bits	Mnemonic	Field Name	Description
31:0	SADDR	Source Address	Source Address (Initial value: undefined, R/W) This field sets the physical address of the transfer source during Dual Address transfer. This field sets the physical address of memory access during Single Address transfer. This field is used for either Memory-to-I/O or I/O-to-Memory transfers. Refer to “8.3.7.1 Channel Register Settings During Single Address Transfer” and “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes once for each bus operation only by the size that was transferred. During Single transfer, the value only changes by the value specified by the DMA Source Address Increment Register (DMSAIRn).

Figure 8.4.4 DMA Source Address Register

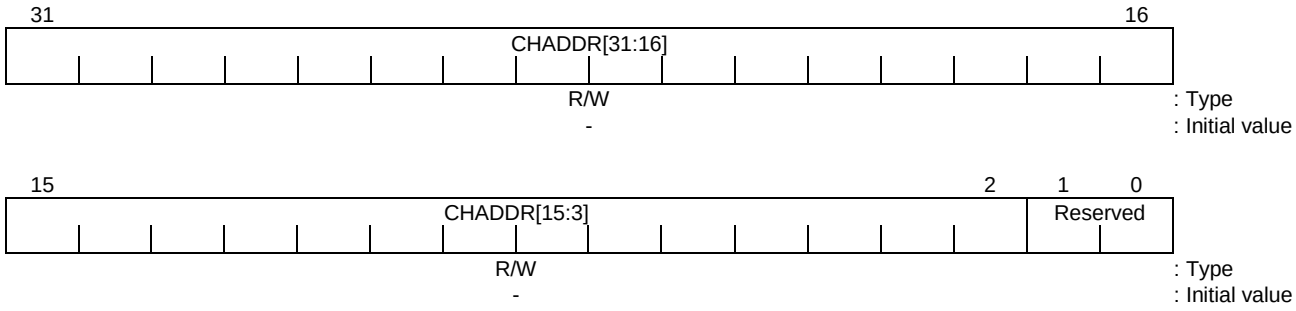
8.4.5 DMA Destination Address Register (DMDARn) 0xB008 (ch. 0) 0xB028 (ch. 1) 0xB048 (ch. 2) 0xB068 (ch. 3)



Bits	Mnemonic	Field Name	Description
31:0	DADDR	Destination Address	Destination Address (Initial value: undefined, R/W) This register sets the physical address of the transfer destination during Dual Address transfer. This register is ignored during Single Address transfer. Refer to “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information. During Burst transfer, the value changes only by the size of data transferred during each single bus operation. During Single transfer, the value only changes by the value specified by the DMA Destination Address Increment Register (DMDAIRn).

Figure 8.4.5 DMA Destination Address Register

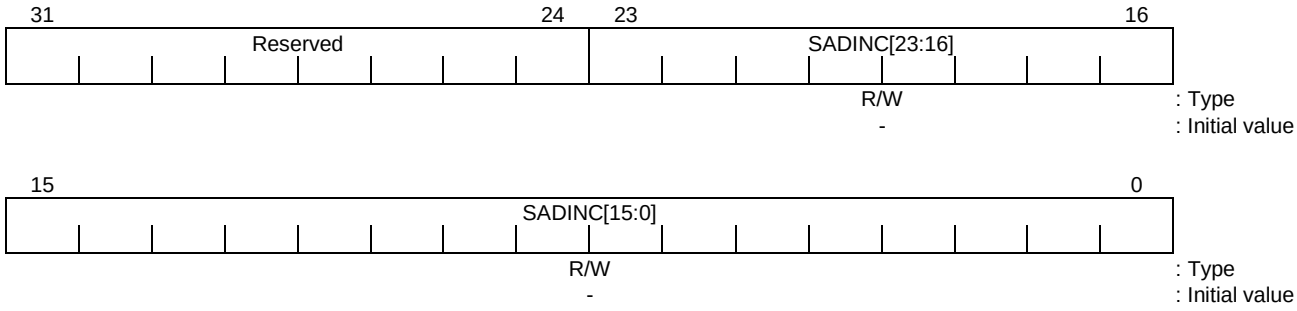
8.4.6 DMA Chain Address Register (DMCHARn) 0xB000 (ch. 0) 0xB020 (ch. 1) 0xB040 (ch. 2) 0xB060 (ch. 3)



Bits	Mnemonic	Field Name	Description
31:2	CHADDR	Chain Address	Chain Address (Initial value: undefined, R/W) When Chain DMA transfer is executed, this register sets the physical address of the next DMA Command Descriptor to be read. If DMA transfer according to the current Channel Register setting ends and the Chain Enable bit (DMCCRn.CHNEN) is set, then the DMA Command Descriptor is loaded in the Channel Register starting from the address indicated by this register. When a value other than “0” is set in this register, the Chain Enable bit (DMCCRn.CHNEN) and the Transfer Active bit (DMCCRn.XFACT) are set. When “0” is set in this register, only the Chain Enable bit (DMCCRn.CHNEN) is cleared. When the Chain Address field value reads a DMA Command Descriptor of 0, the value of this register is not updated and the value before that one (address of the Data Command Descriptor when the value of the Chain Address field being read was “0”) is held.
2:0	—	Reserved	—

Figure 8.4.6 DMA Chain Address Register

8.4.7 DMA Source Address Increment Register (DMSAIRn) 0xB010 (ch. 0) 0xB030 (ch. 1) 0xB050 (ch. 2) 0xB070 (ch. 3)

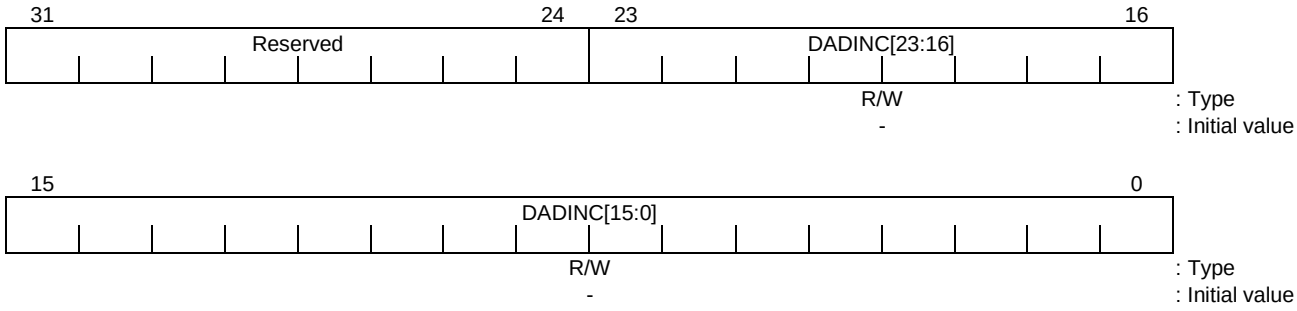


Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23:0	SADINC	Source Address Increment	Source Address Increment (Initial value: undefined, R/W) This field sets the increase/decrease value of the DMA Source Address Register (DMSARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "8.3.7.1 Channel Register Settings During Single Address Transfer" and "8.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.

Figure 8.4.7 DMA Source Address Increment Register

8.4.8 DMA Destination Address Increment Register (DMDAIRn)

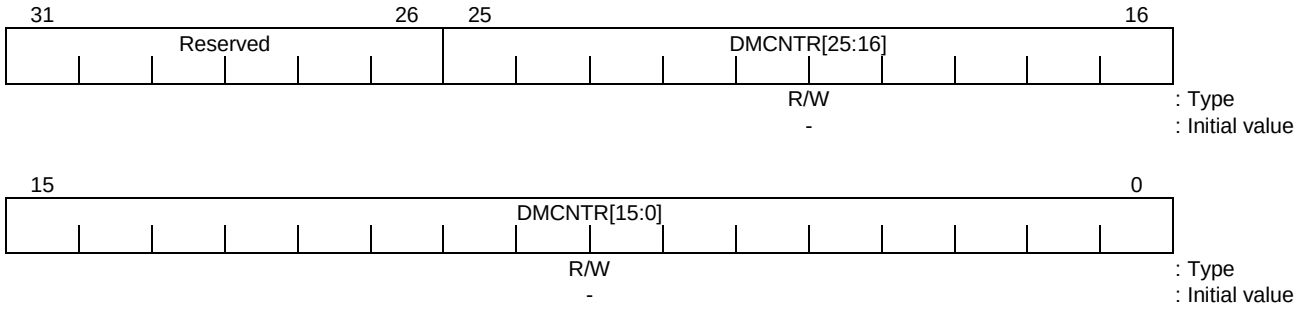
0xB014 (ch. 0) 0xB034 (ch. 1)
0xB054 (ch. 2) 0xB074 (ch. 3)



Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23:0	DADINC	Destination Address Increment	Destination Address Increment (Initial value: undefined, R/W) This field sets the increase/decrease value of the DMA Destination Address Register (DMDARn). This value is a 24-bit two's complement and indicates a byte count. Refer to "8.3.8.1 Channel Register Settings During Dual Address Transfer" for more information.

Figure 8.4.8 DMA Destination Address Increment Register

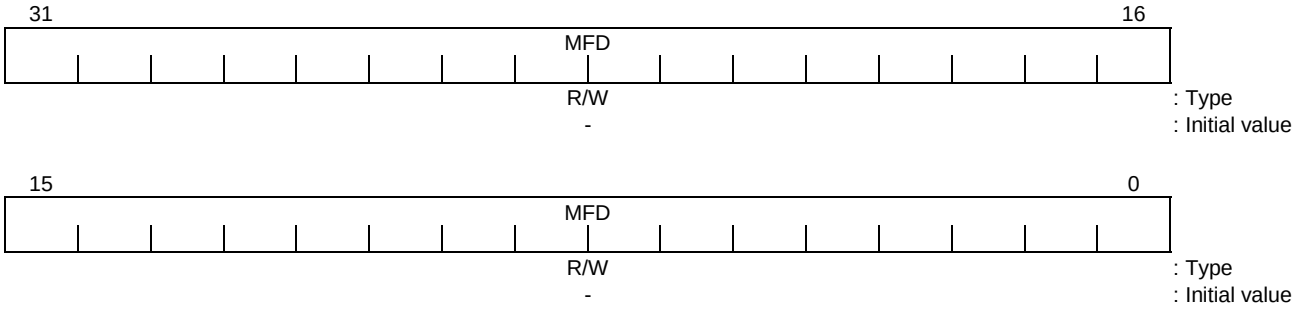
8.4.9 DMA Count Register (DMCNTRn) 0xB00C (ch. 0) 0xB02C (ch. 1) 0xB04C (ch. 2) 0xB06C (ch. 3)



Bits	Mnemonic	Field Name	Description
31:26	—	Reserved	—
25:0	DMCNTR	Count	Count Register (Initial value: undefined, R/W) This register sets the byte count that is transferred by the DMA Channel Register setting. The value is a 26-bit unsigned data that is decremented only by the size of the data transferred during a single bus operation. Refer to “8.3.7.1 Channel Register Settings During Single Address Transfer” and “8.3.8.1 Channel Register Settings During Dual Address Transfer” for more information.

Figure 8.4.9 DMA Count Register

8.4.10 DMA Memory Fill Data Register (DMMFDR) 0xB0A4



Bits	Mnemonic	Field Name	Description
31:0	MFD	Memory Fill Data	Memory Fill Data (Initial value: undefined, R/W) This register, which stores word data written to memory when in the Memory Fill Transfer mode, is shared between all channels.

Figure 8.4.10 DMA Memory Fill Data Register

8.5 Timing Diagrams

This section contains timing diagrams for the external I/O DMA transfer mode. The DMAREQ[n] signals and DMAACK[n] signals in the timing diagrams are set to Low Active.

8.5.1 Single Address Single Transfer from Memory to I/O (32-bit ROM)

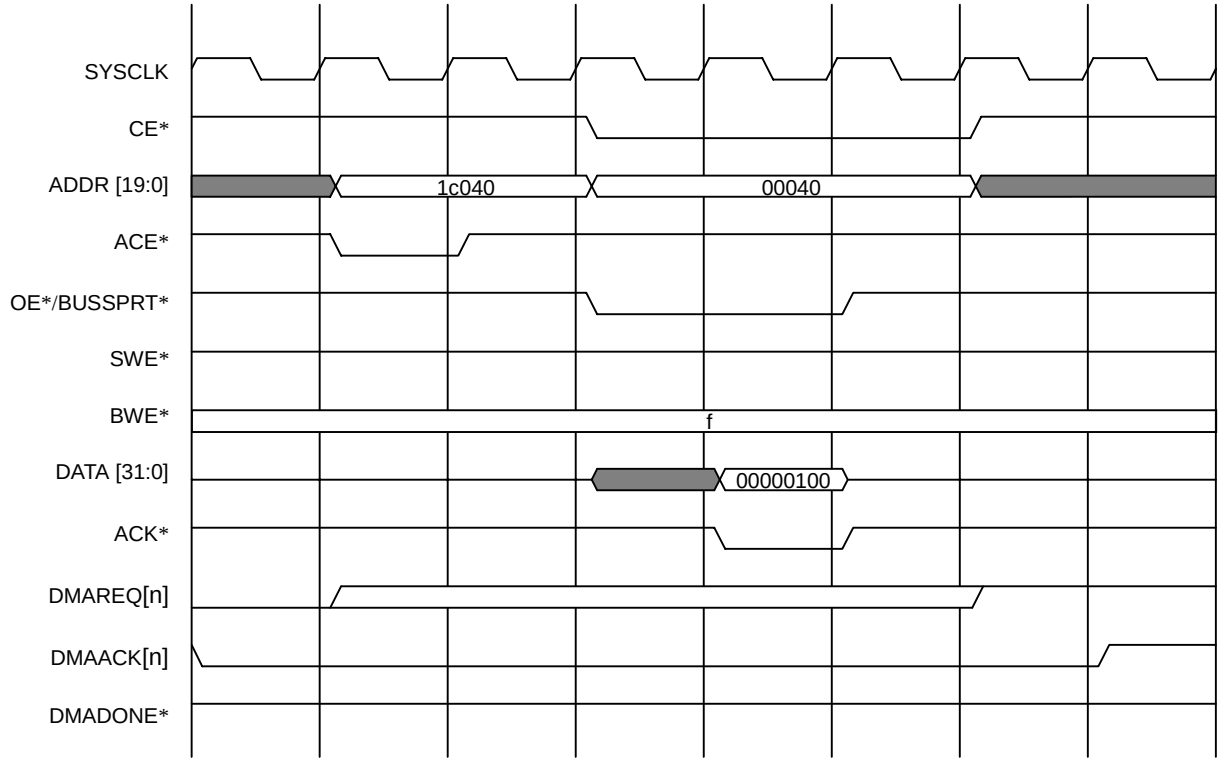


Figure 8.5.1 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit ROM)

8.5.2 Single Address Single Transfer from Memory to I/O (16-bit ROM)

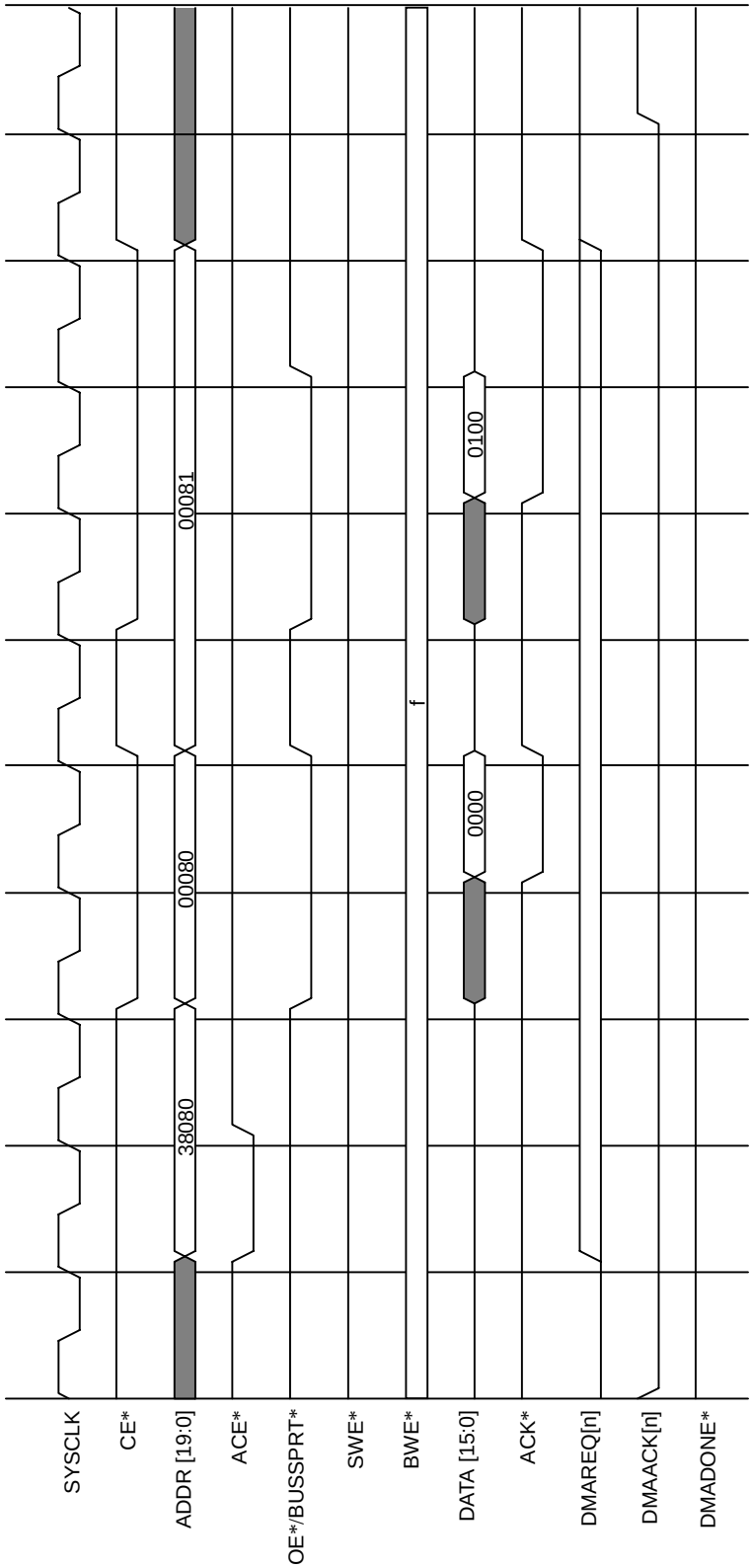


Figure 8.5.2 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 16-bit ROM)

8.5.3 Single Address Single Transfer from I/O to Memory (32-bit SRAM)

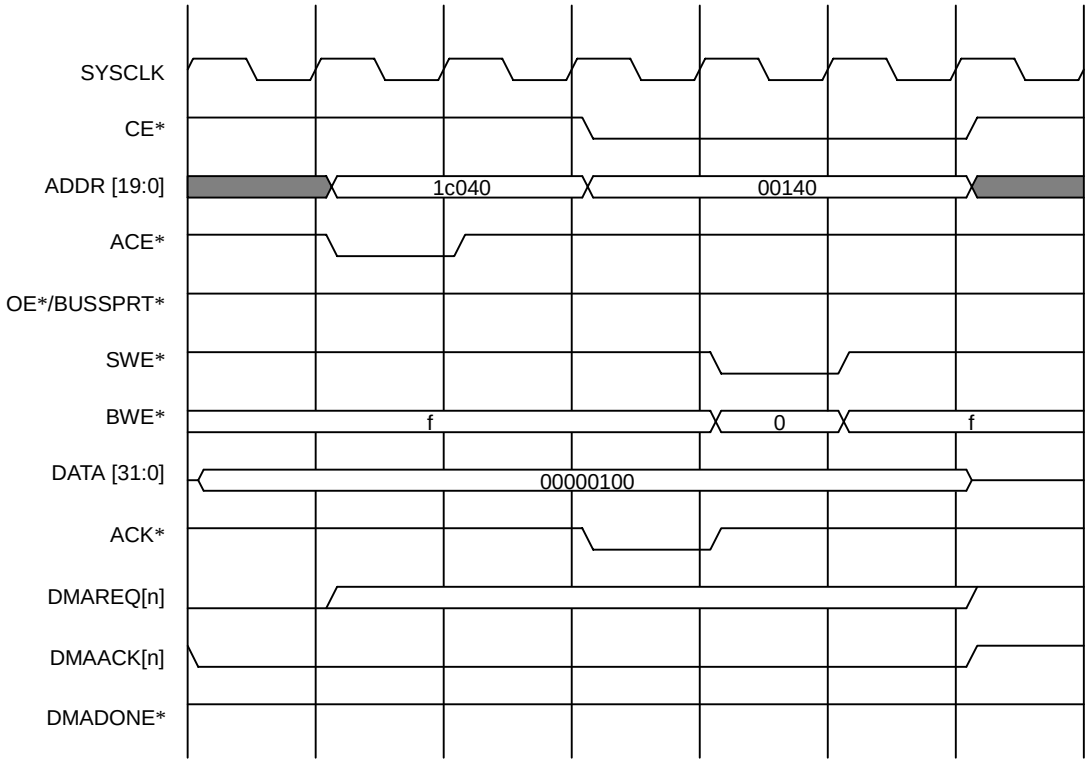


Figure 8.5.3 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit SRAM)

8.5.4 Single Address Burst Transfer from Memory to I/O (32-bit ROM)

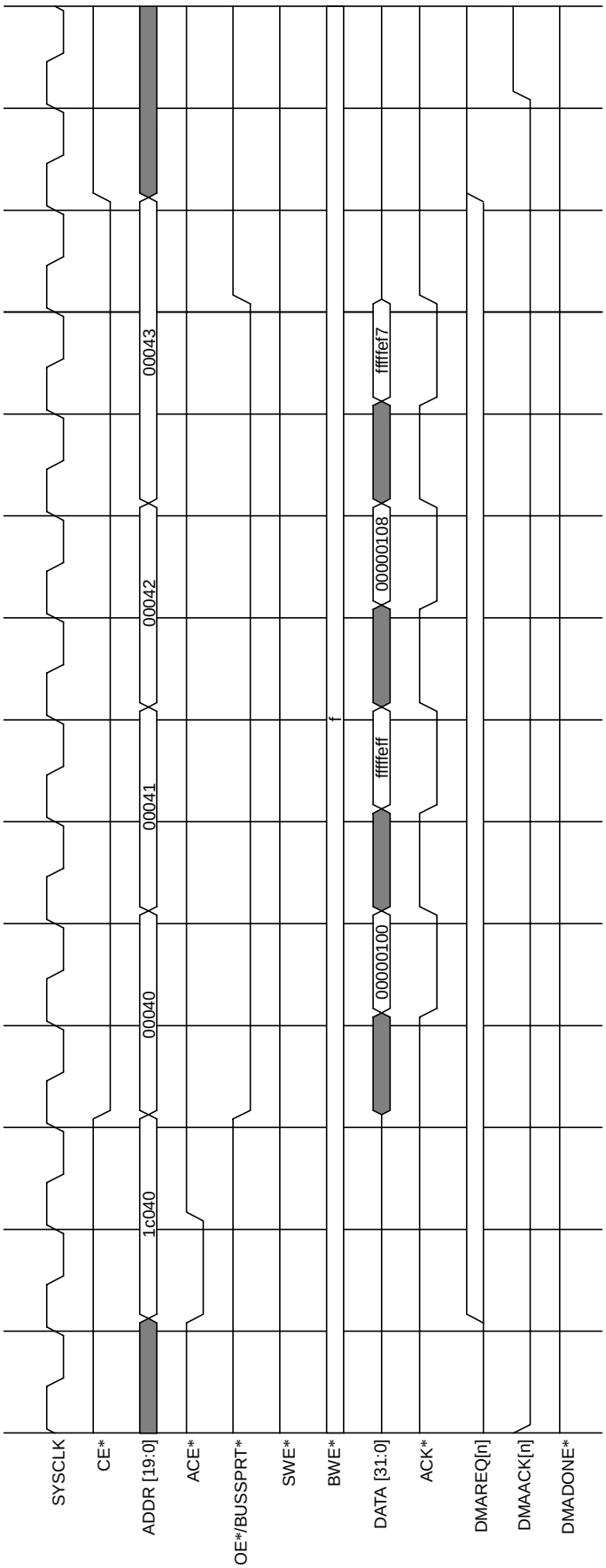


Figure 8.5.4 Single Address Burst Transfer from Memory to I/O
(Burst Read of 4-word Data from 32-bit ROM)

8.5.5 Single Address Burst Transfer from I/O to Memory (32-bit SRAM)

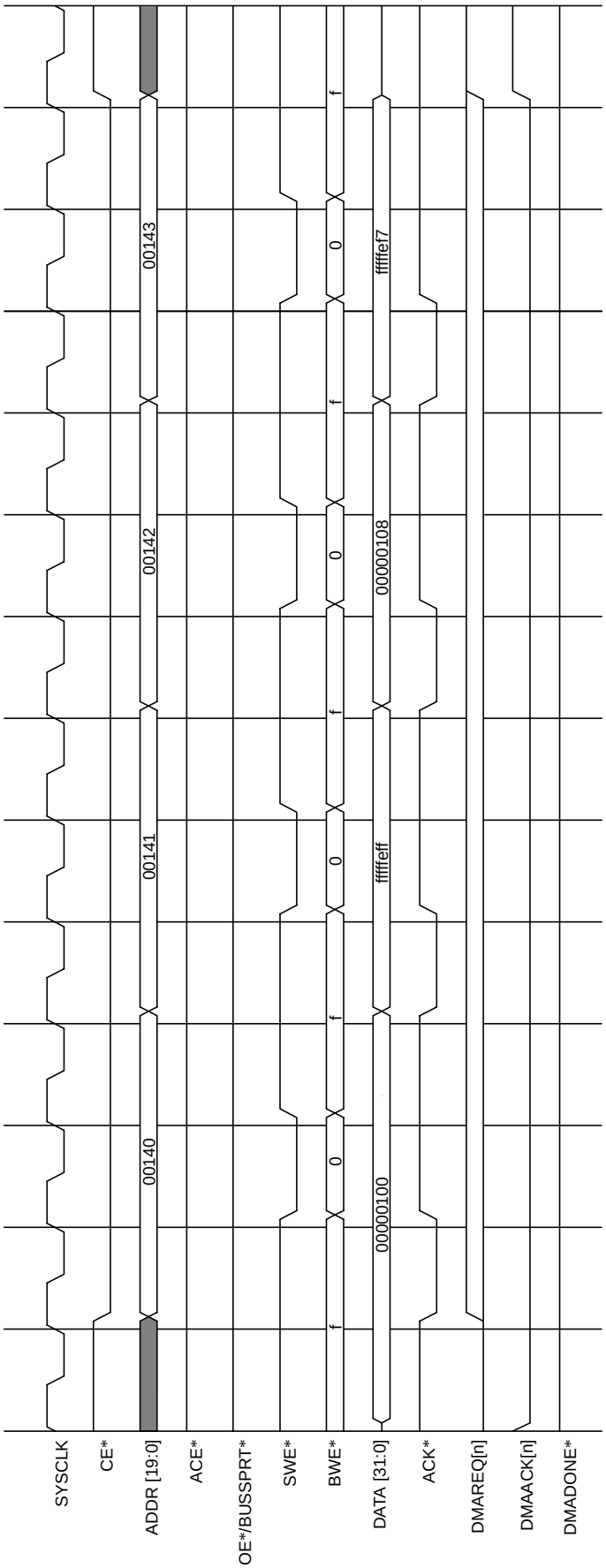


Figure 8.5.5 Single Address Burst Transfer from I/O to Memory
(Burst Write of 4-word Data from 32-bit SRAM)

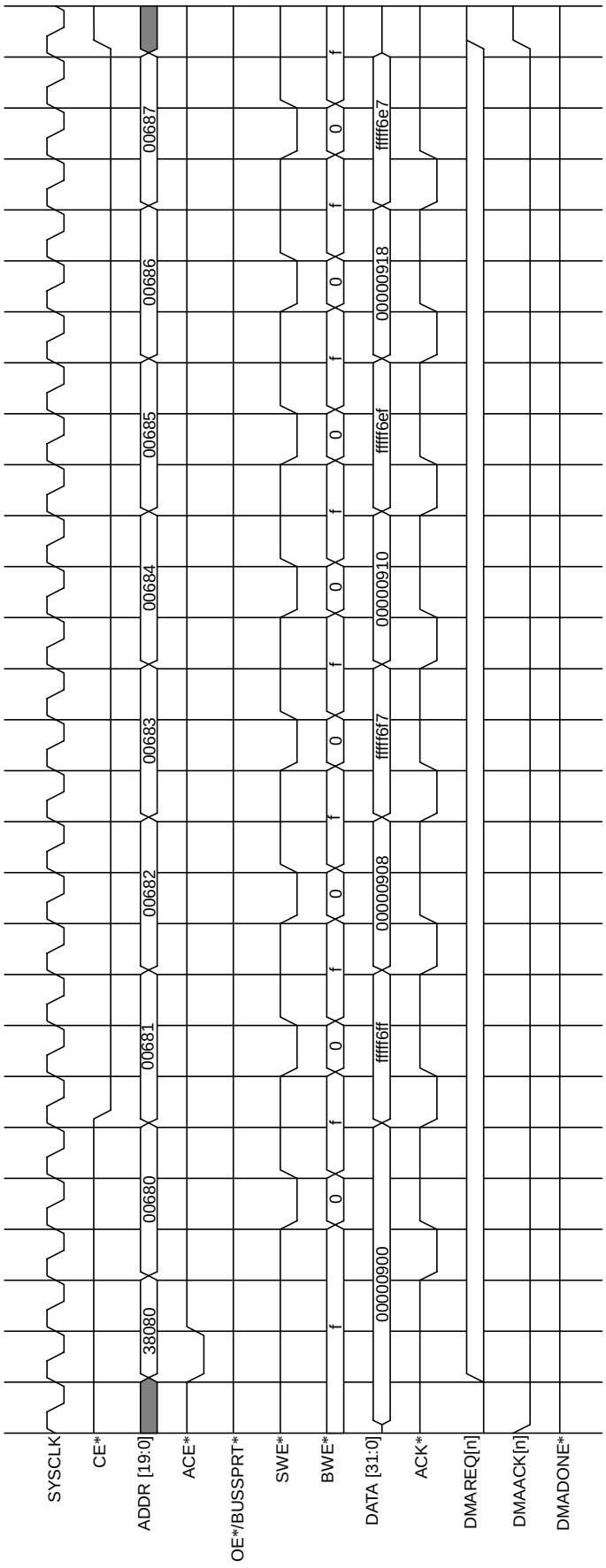


Figure 8.5.6 Single Address Burst Transfer from I/O to Memory
(Burst Write of 8-word Data to 32-bit SRAM)

8.5.6 Single Address Single Transfer from Memory to I/O (16-bit ROM)

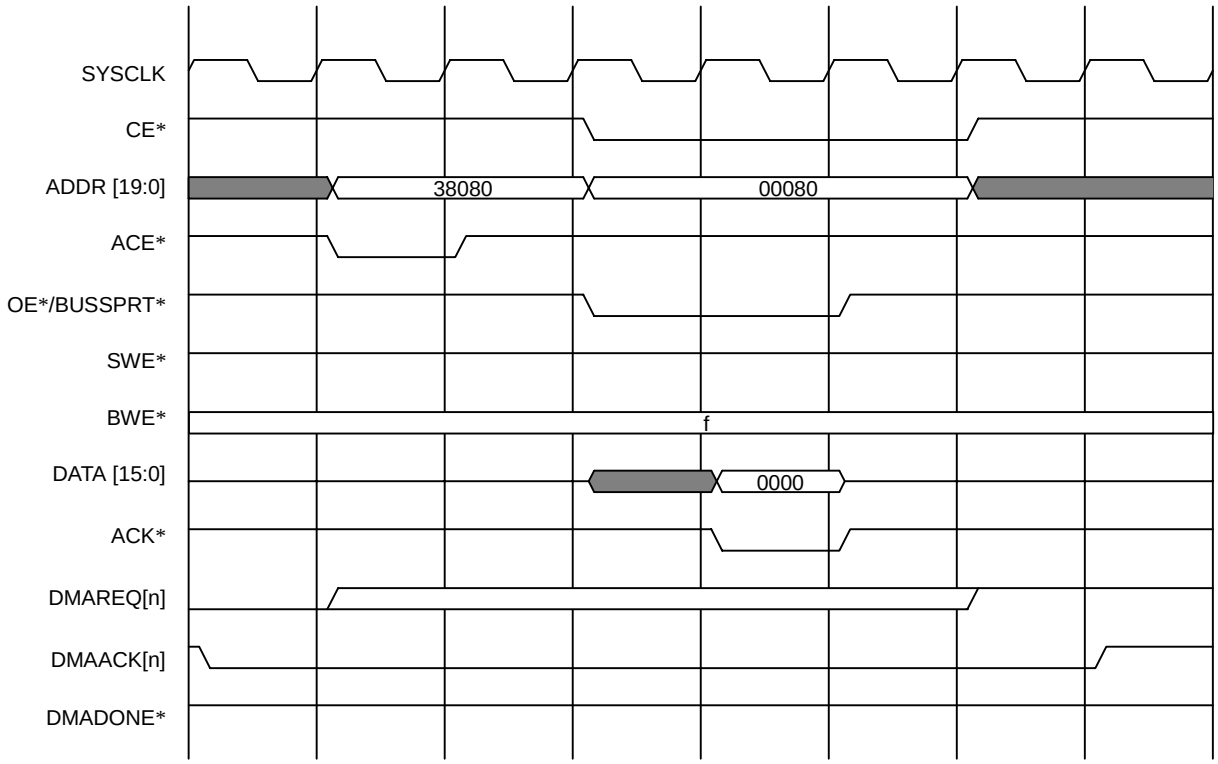


Figure 8.5.7 Single Address Single Transfer from Memory to I/O
(Single Read from 16-bit ROM to 16-bit Data)

8.5.7 Single Address Single Transfer from I/O to Memory (16-bit SRAM)

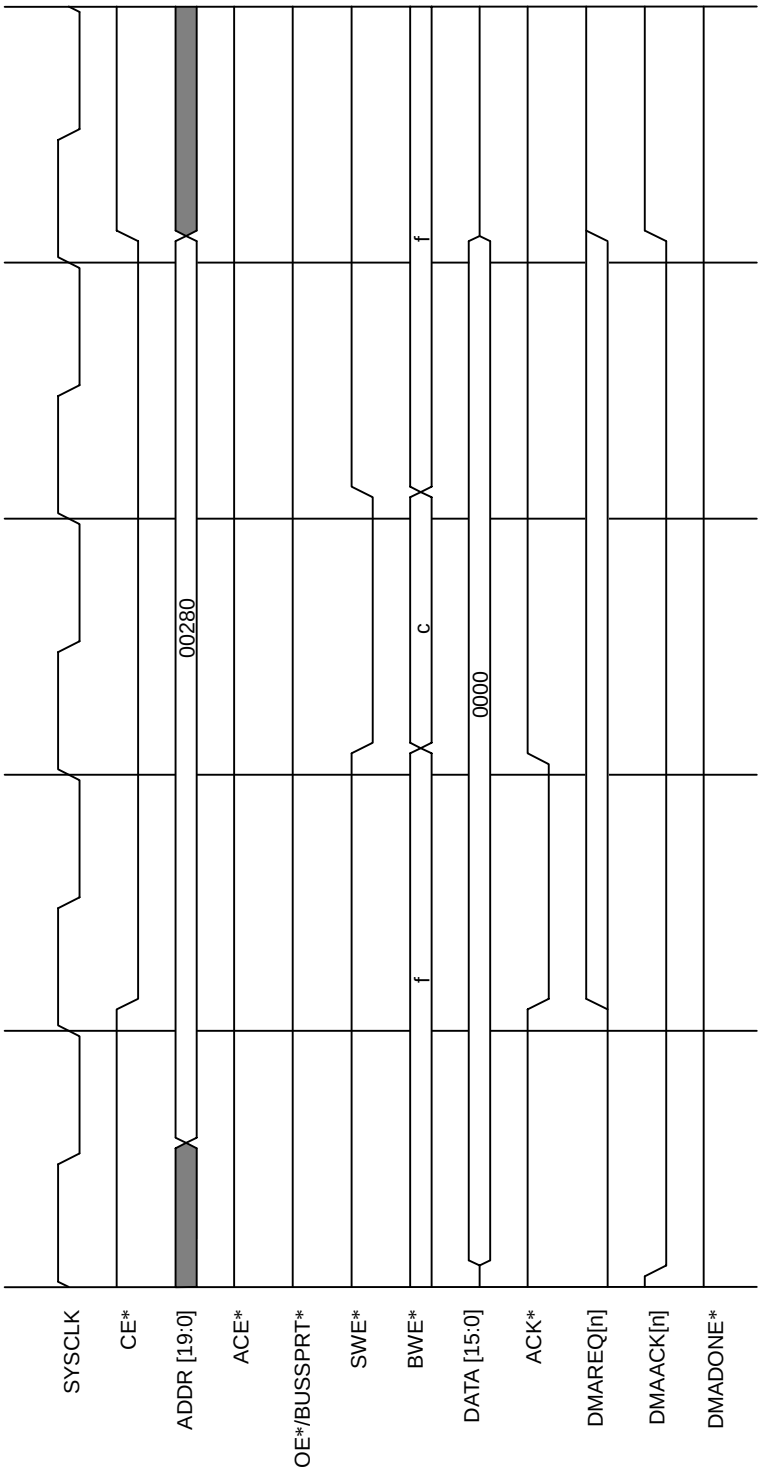


Figure 8.5.8 Single Address Single Transfer from I/O to Memory
(Single Write of 16-bit Data to 16-bit SRAM)

8.5.8 Single Address Single Transfer from Memory to I/O (32-bit Half Speed ROM)

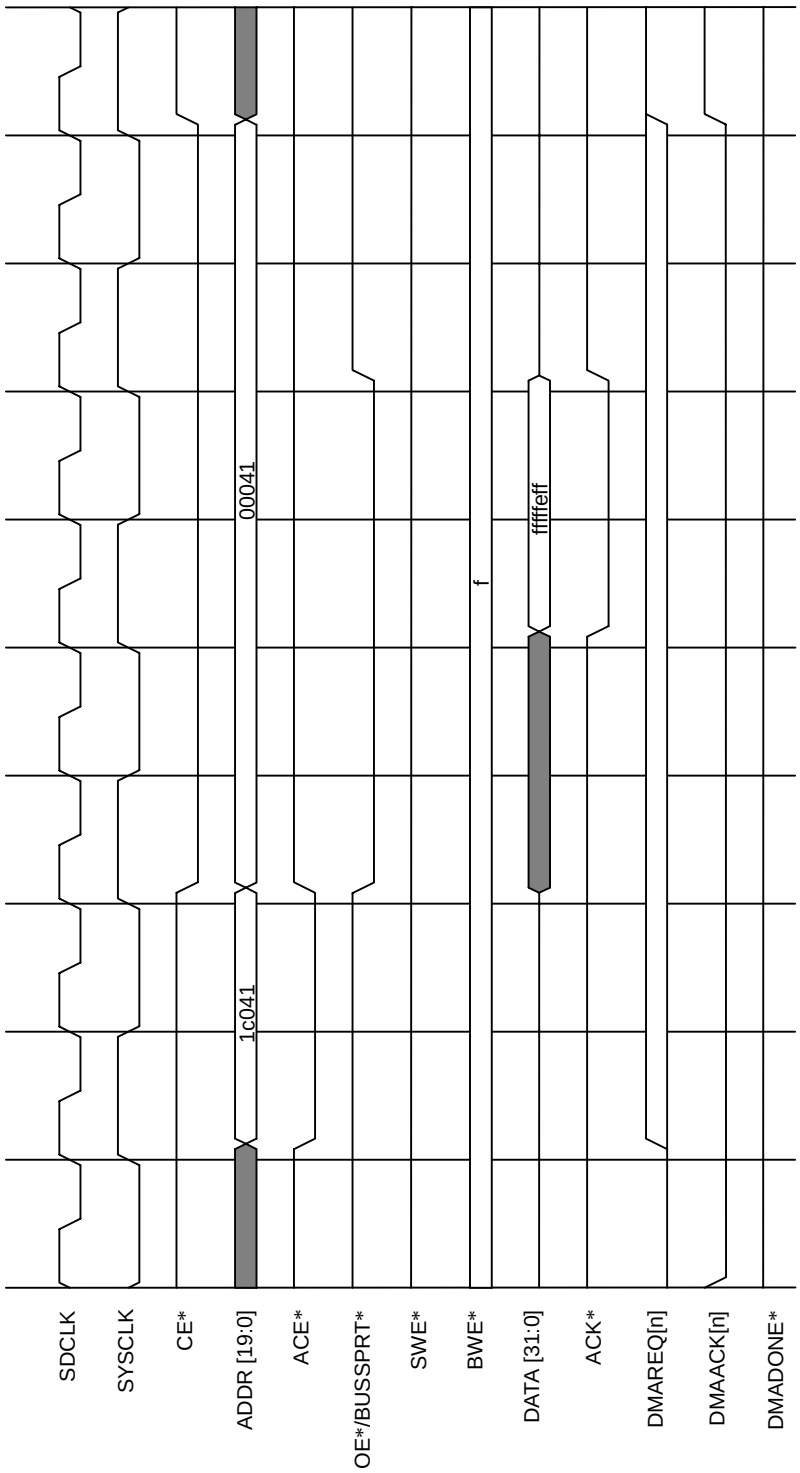


Figure 8.5.9 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit Half Speed ROM)

8.5.9 Single Address Single Transfer from I/O to Memory (32-bit Half Speed SRAM)

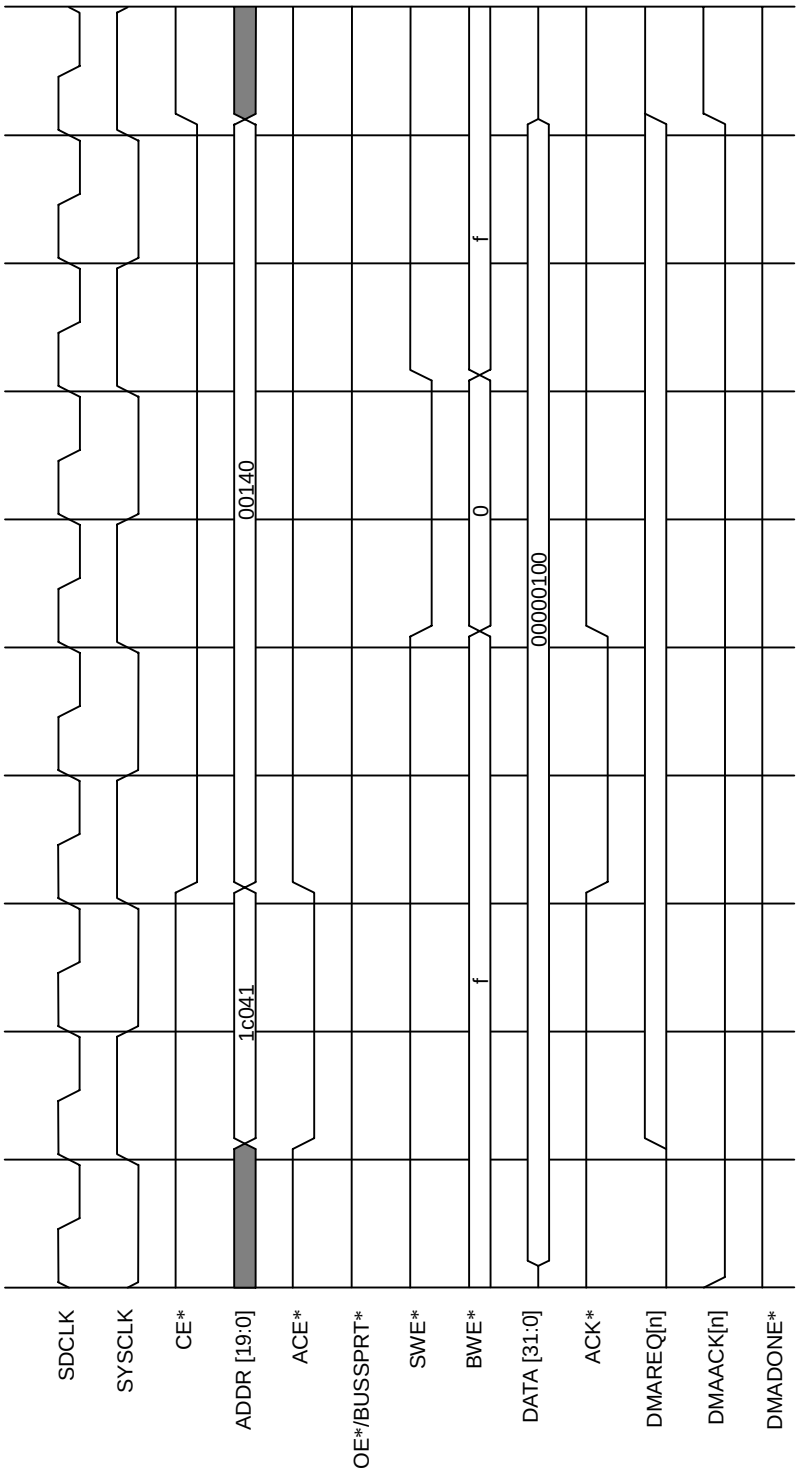


Figure 8.5.10 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit Half Speed SRAM)

8.5.10 Single Address Single Transfer from Memory to I/O (32-bit SRAM)

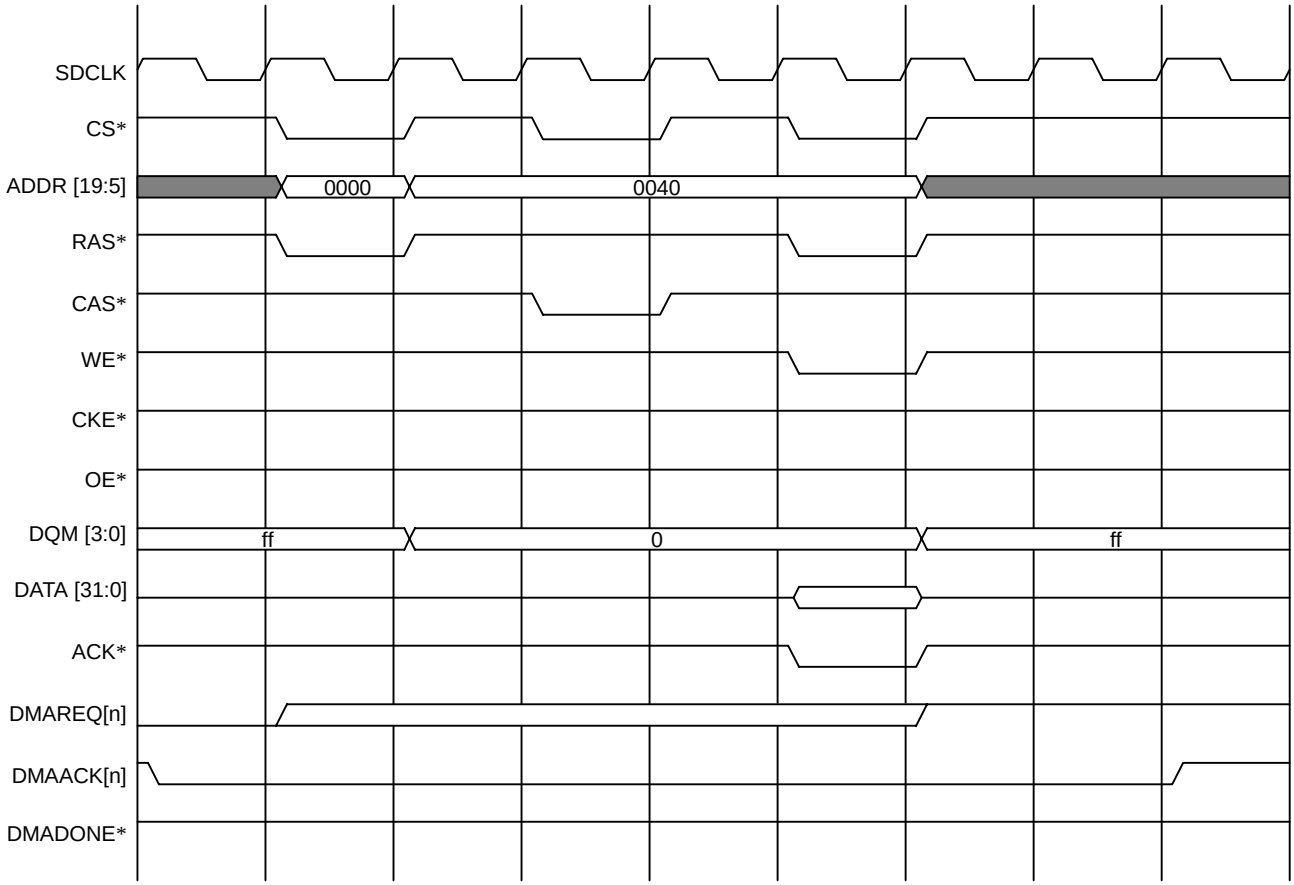


Figure 8.5.11 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit SDRAM)

8.5.11 Single Address Single Transfer from I/O to Memory (32-bit SDRAM)

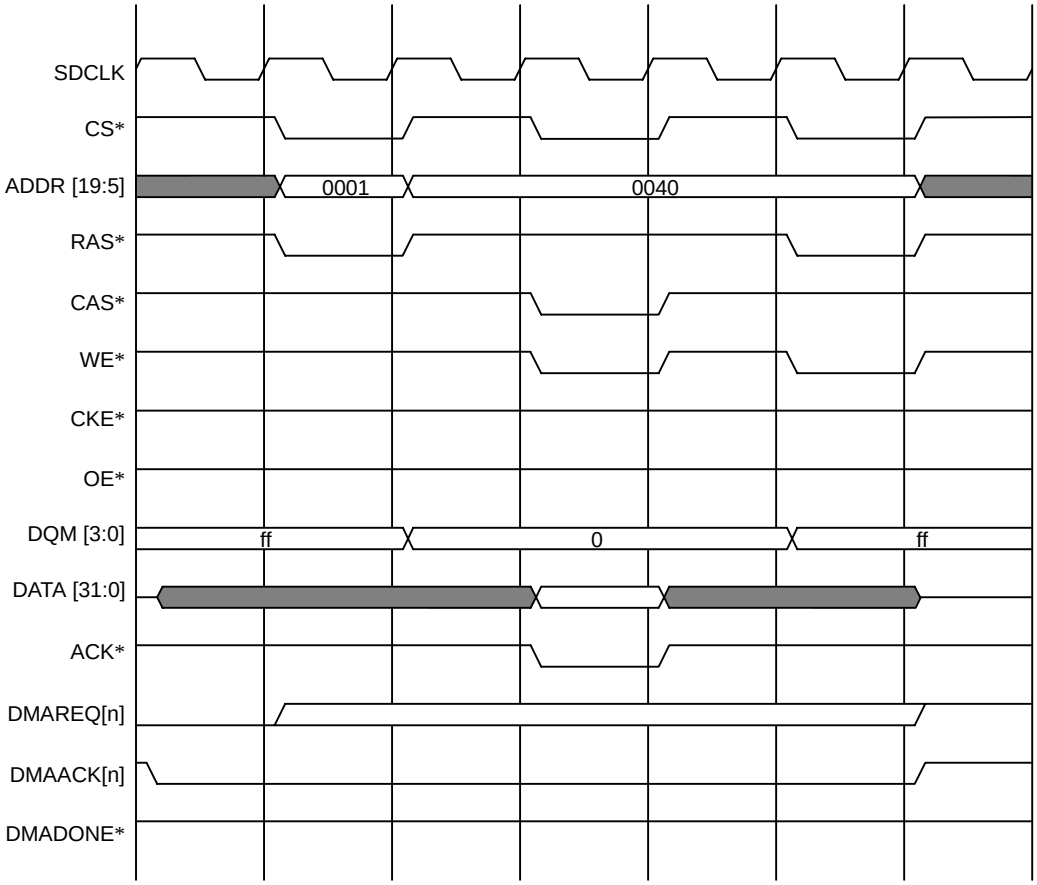


Figure 8.5.12 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit SDRAM)

8.5.12 Single Address Single Transfer from Memory to I/O of Last Cycle when DMADONE* Signal is Set to Output

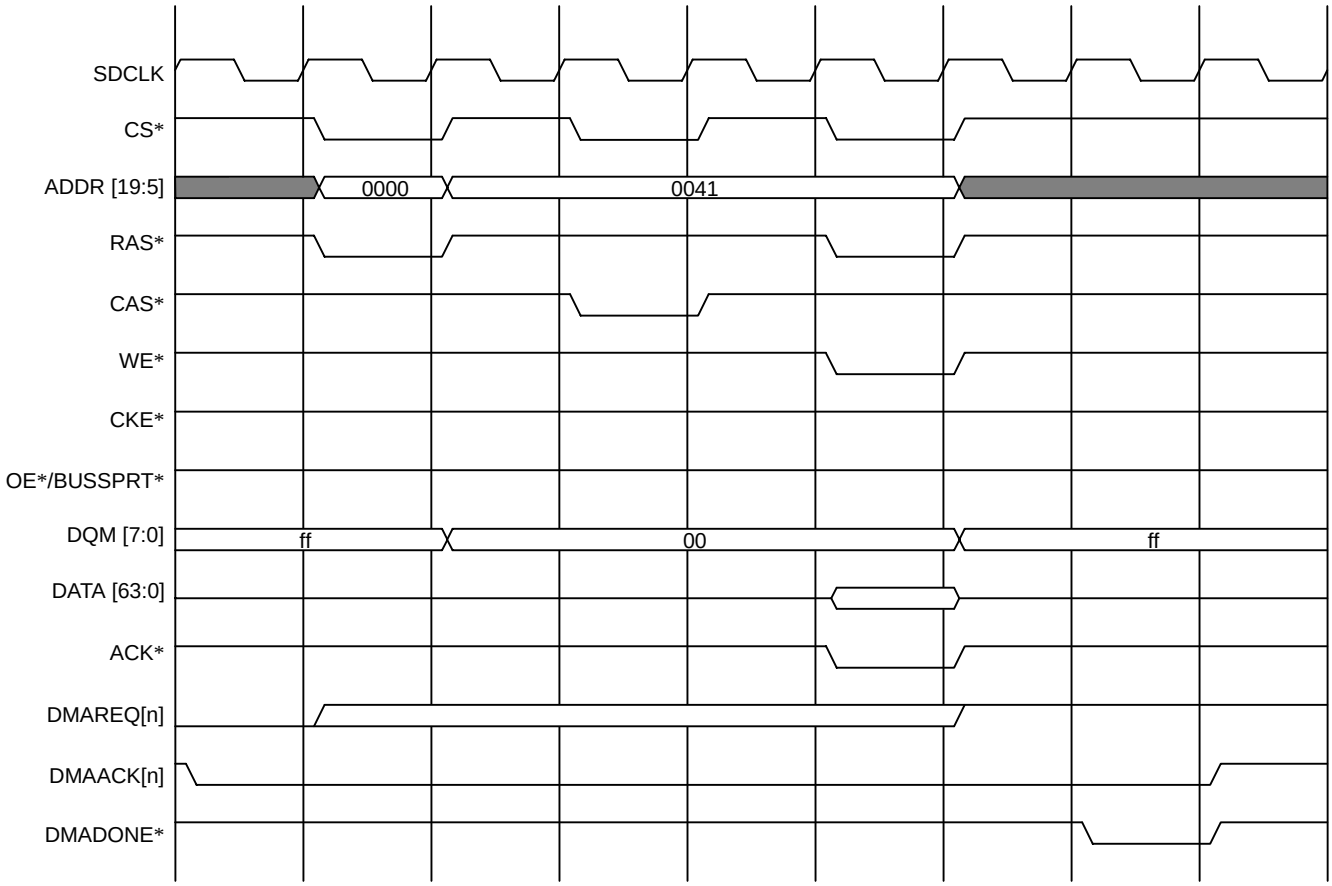


Figure 8.5.13 Single Address Single Transfer from Memory to I/O
(Single Read of 64-bit Data from 64-bit SDRAM)

8.5.13 Single Address Single Transfer from Memory to I/O (32-bit SDRAM)

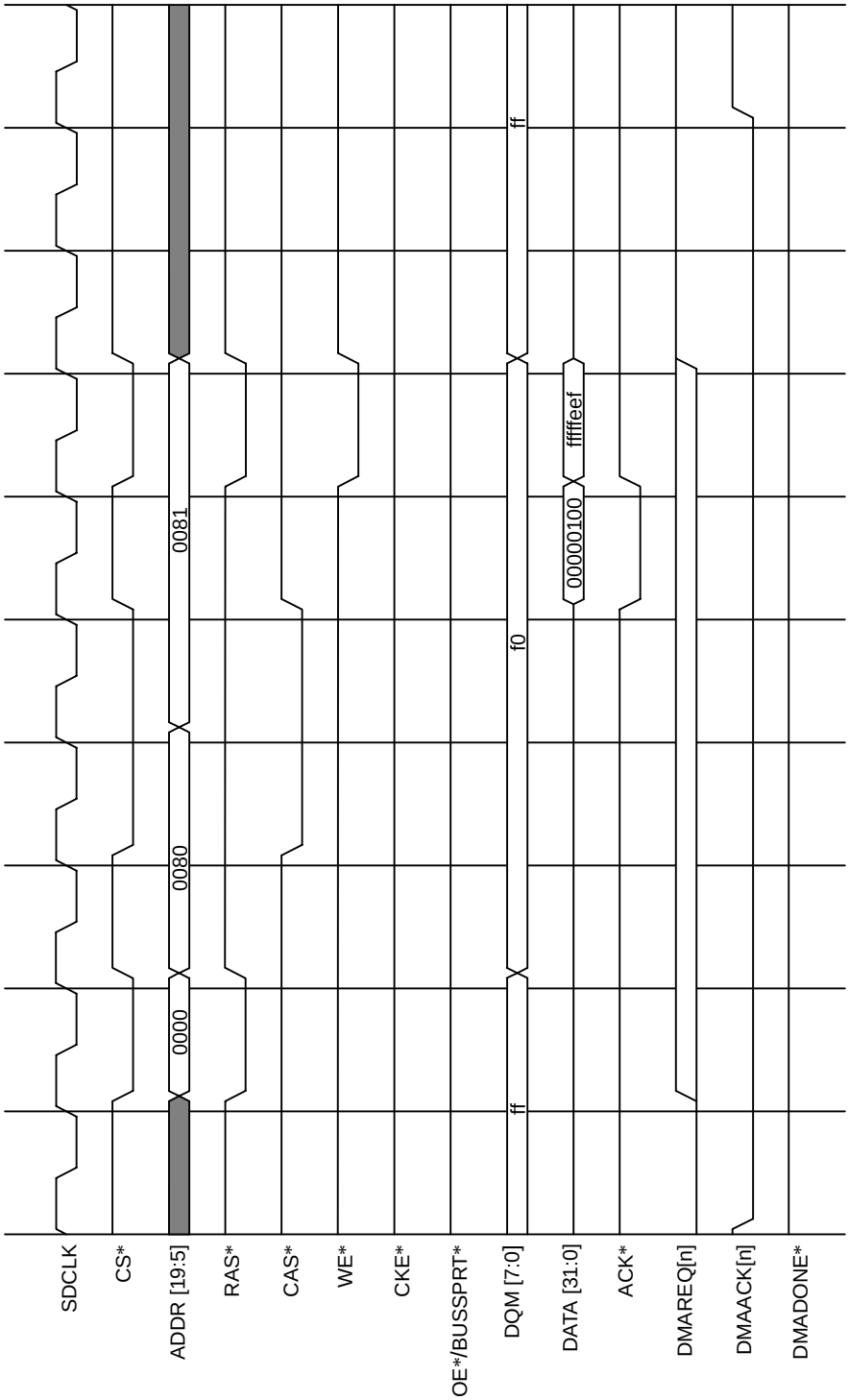


Figure 8.5.14 Single Address Single Transfer from Memory to I/O
(Single Read of 32-bit Data from 32-bit SDRAM)

8.5.14 Single Address Single Transfer from I/O to Memory (32-bit SDRAM)

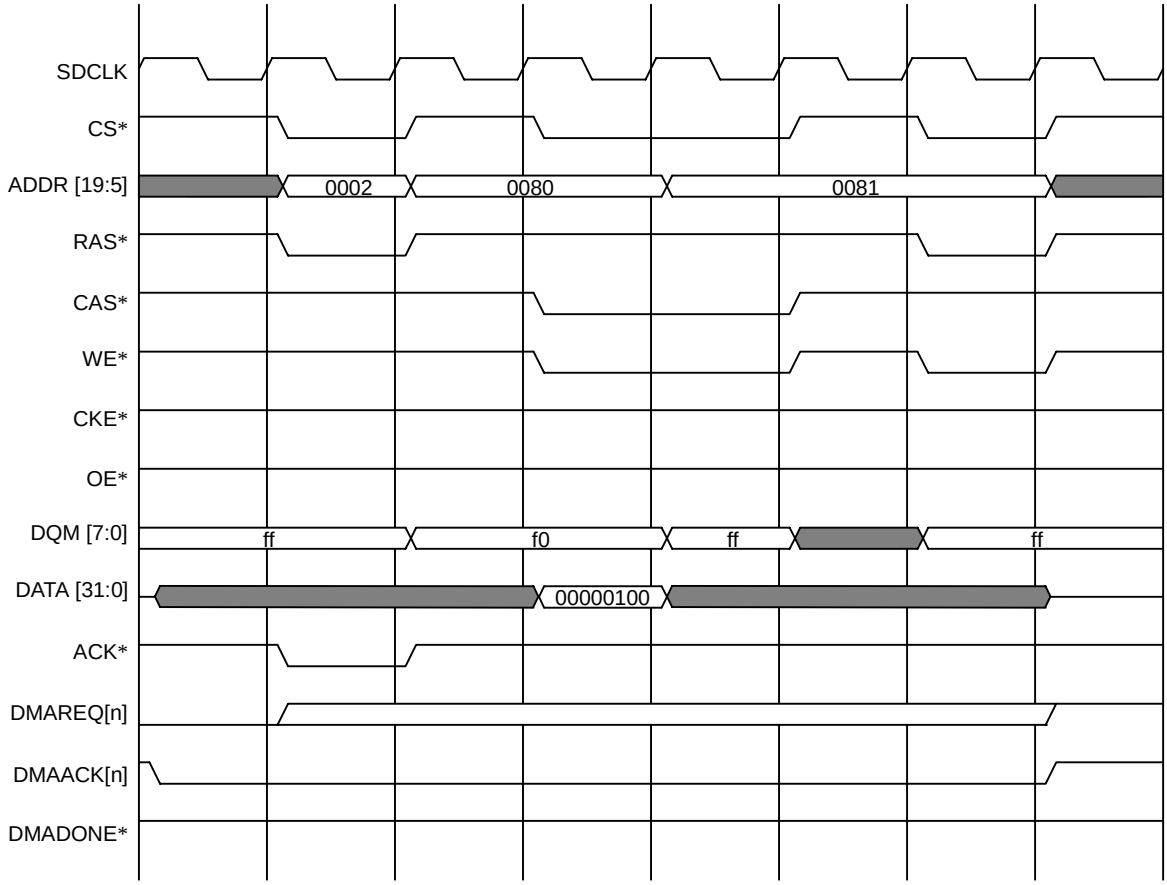


Figure 8.5.15 Single Address Single Transfer from I/O to Memory
(Single Write of 32-bit Data to 32-bit SDRAM)

8.5.15 External I/O Device – SRAM Dual Address Transfer

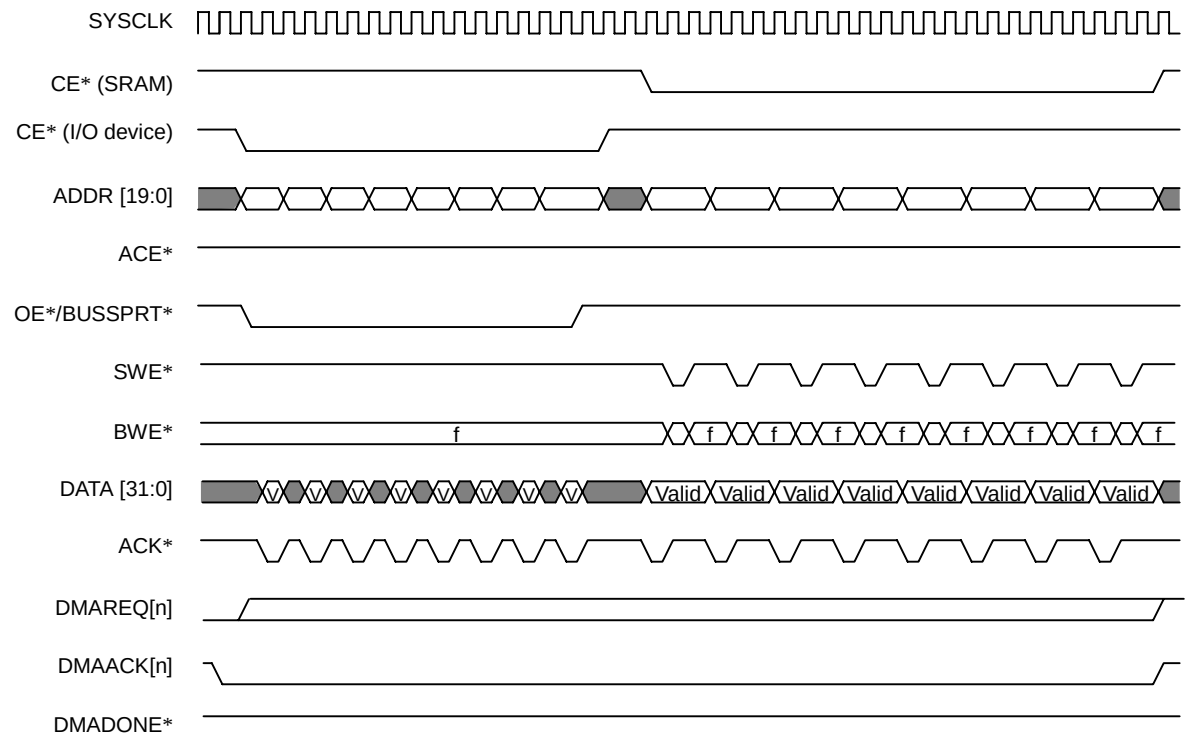


Figure 8.5.16 Dual Address Transfer from External I/O Device to SRAM
(8-word Burst Transfer to 32-bit Bus SRAM)

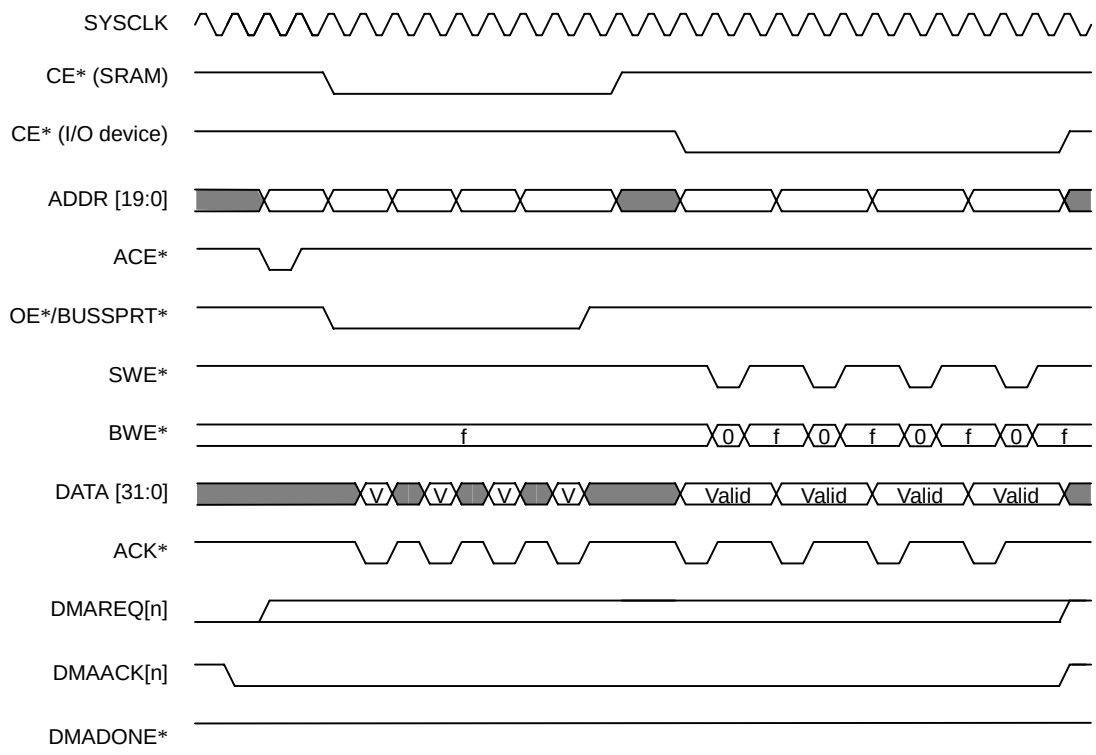


Figure 8.5.17 Dual Address Transfer from SRAM to External I/O Device
(4-word Burst Transfer from 32-bit Bus SRAM)

8.5.16 External I/O Device – SDRAM Dual Address Transfer

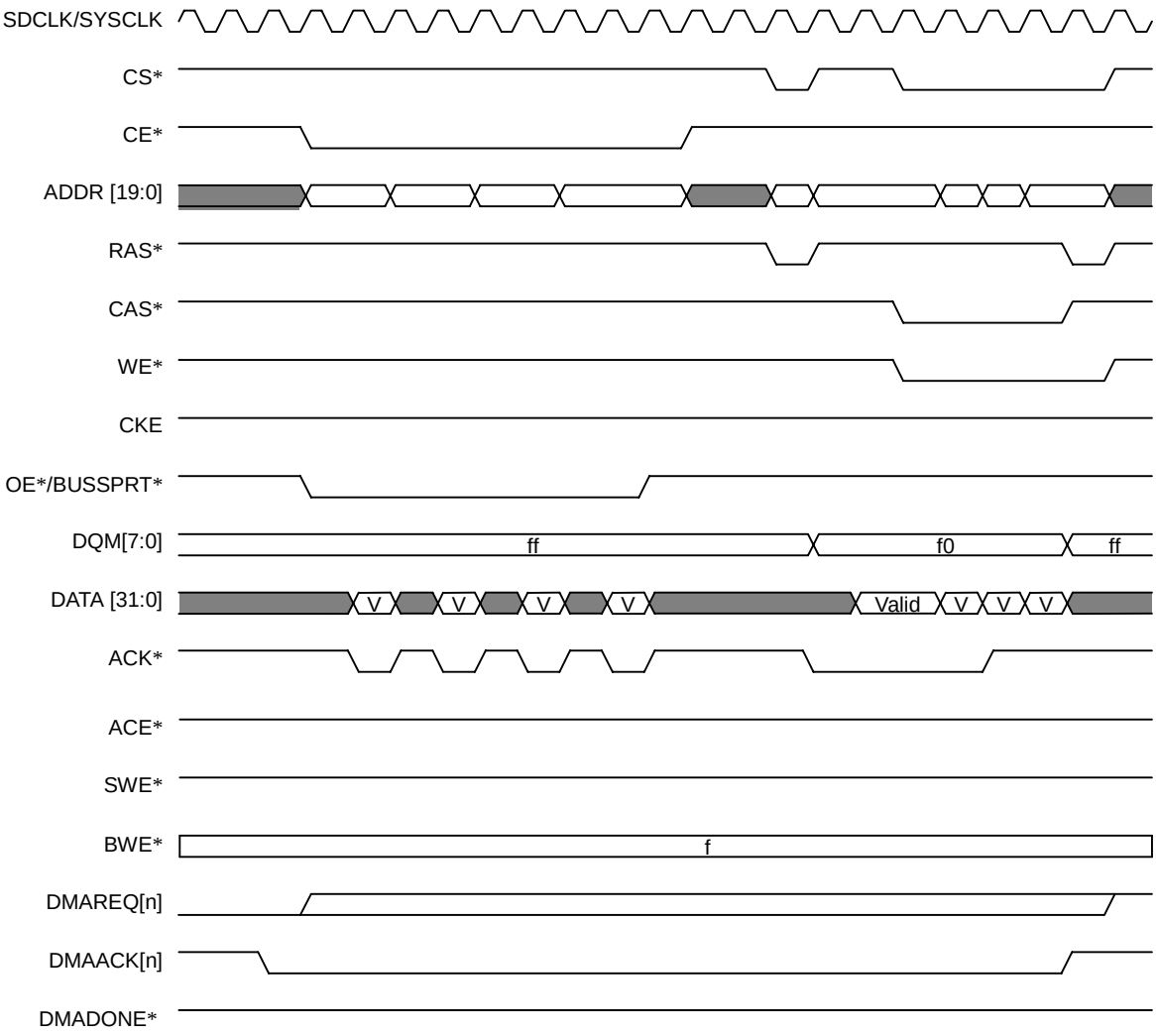


Figure 8.5.18 Dual Address Transfer from External I/O Device to SDRAM
(4-word Burst Transfer to 32-bit SDRAM)

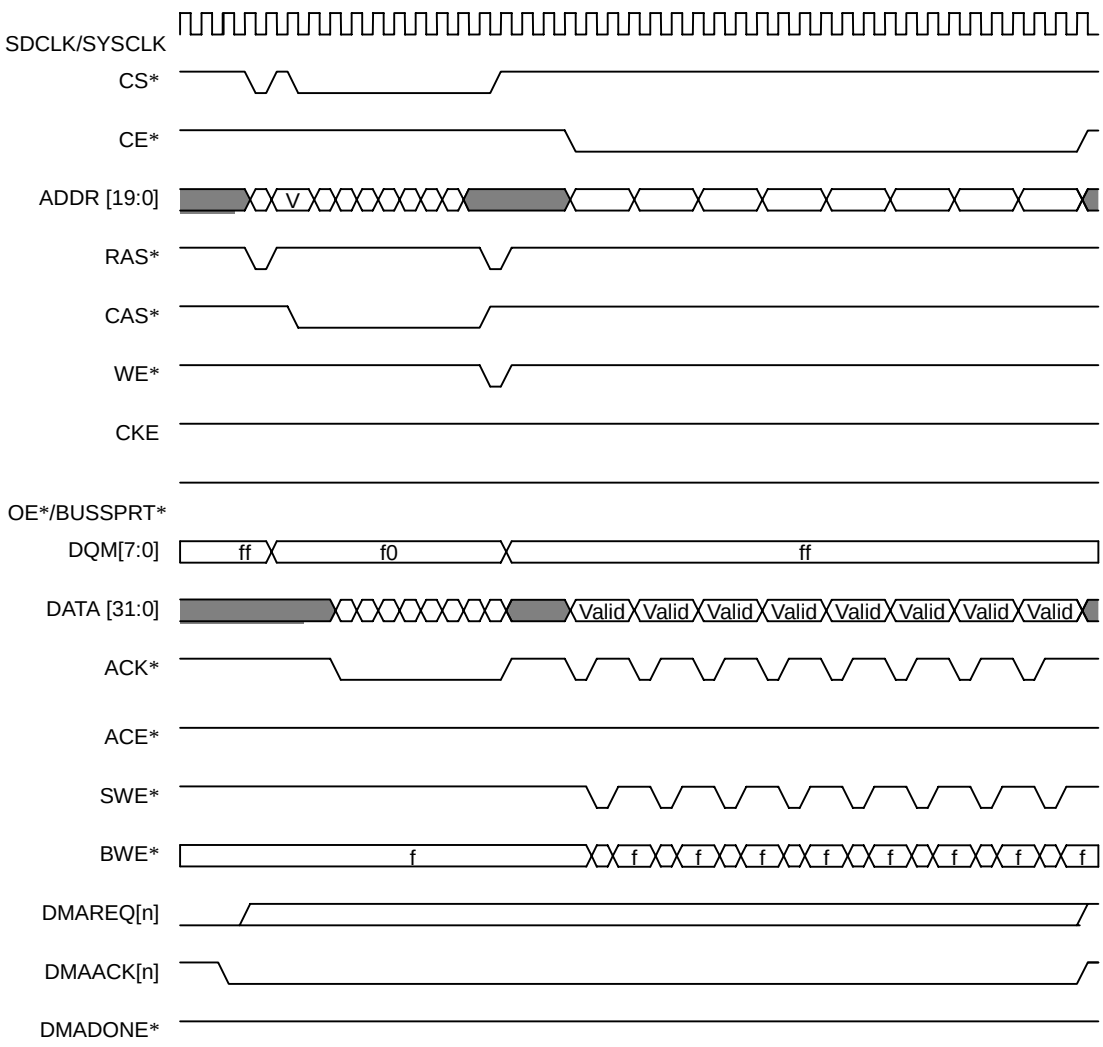


Figure 8.5.19 Dual Address Transfer from SDRAM to External I/O Device
(8-word Burst Transfer from 32-bit SDRAM)

8.5.17 External I/O Device (Non-burst) – SDRAM Dual Address Transfer

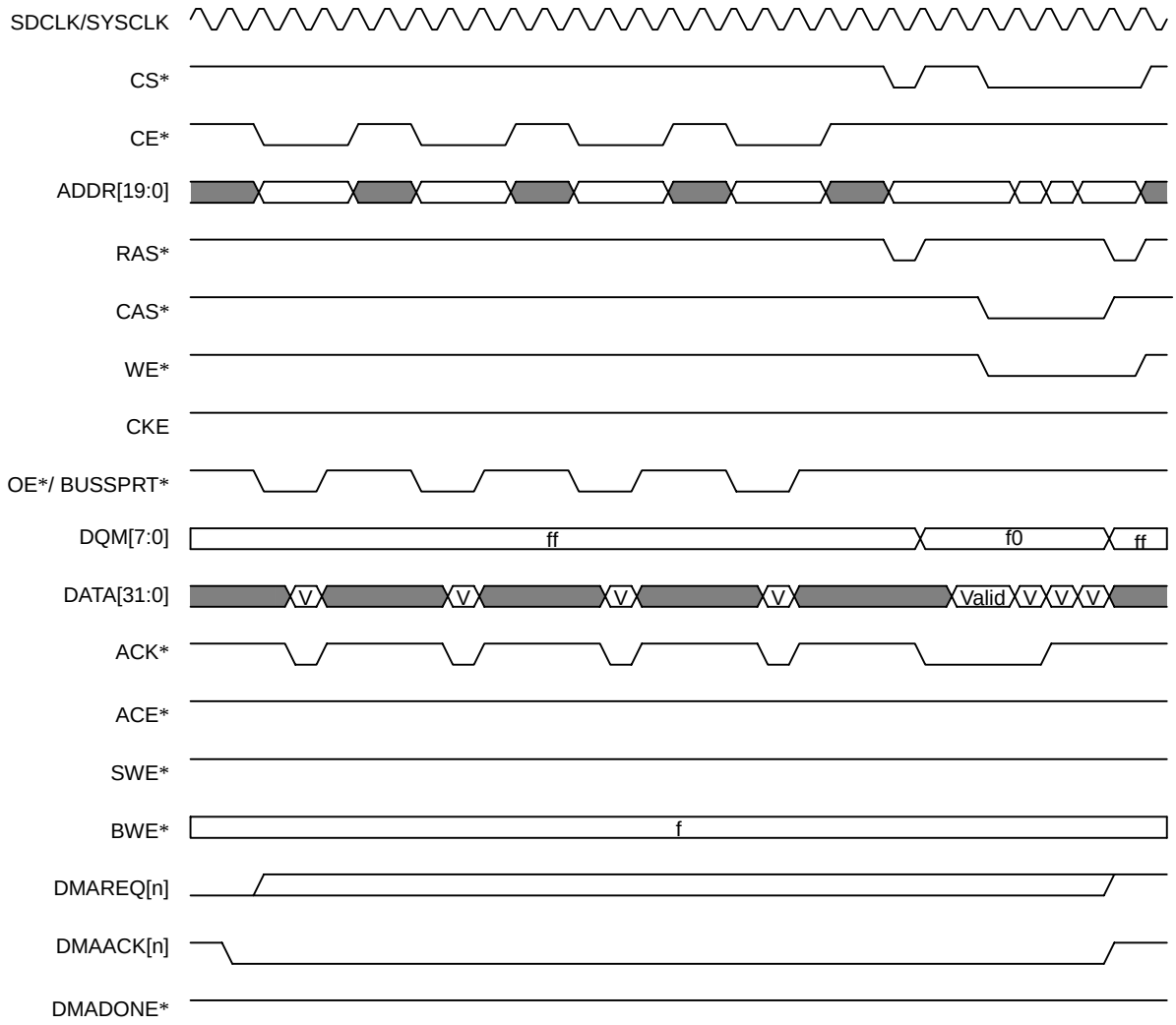


Figure 8.5.20 Dual Address Transfer from External I/O Device (Non-Burst) to SDRAM
(4-word Burst Transfer to 32-bit SDRAM: Set DMCCRn.SBINH to “1”)

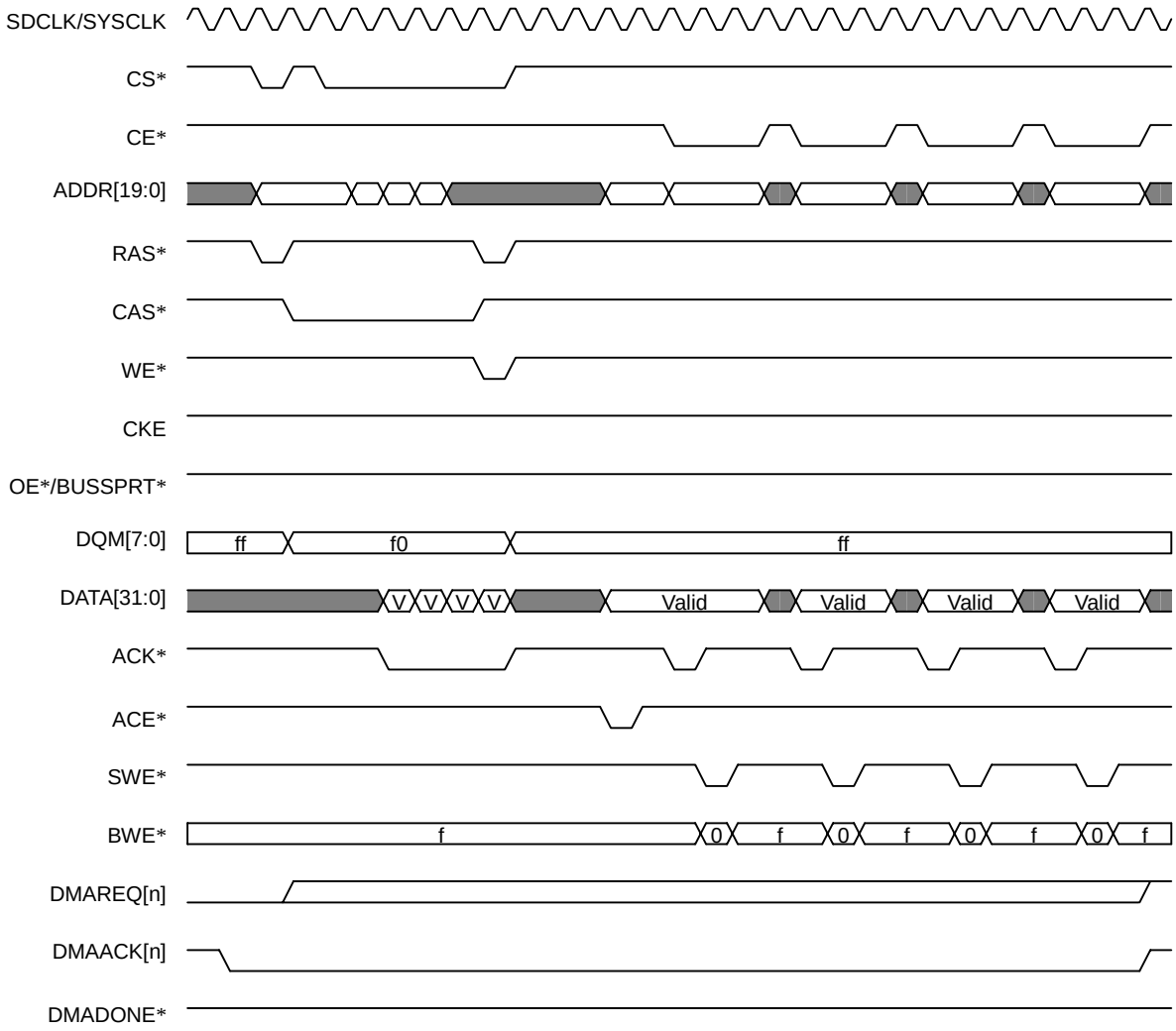


Figure 8.5.21 Dual Address Transfer from SDRAM to External I/O Device
(4-word Burst Transfer from 32-bit SDRAM: Set DMCCRn.DBINH to “1”)

9. SDRAM Controller

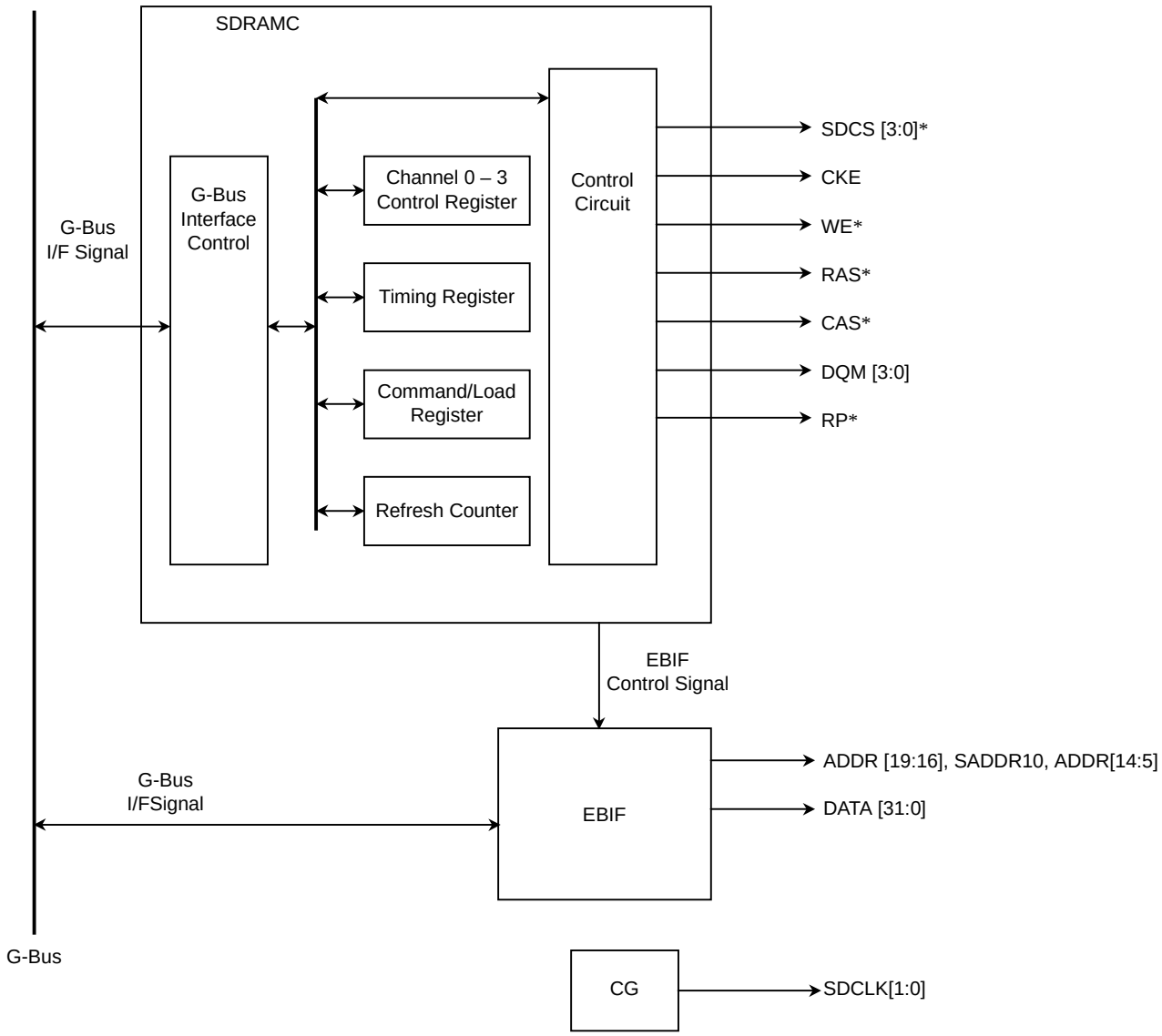
9.1 Characteristics

The SDRAM Controller (SDRAMC) generates the control signals required to interface with the SDRAM. There are a total of four channels, which can each be operated independently. The SDRAM Controller supports various bus configurations and a memory size of up to 2 GB.

The SDRAM has the following characteristics.

- Clock frequency: 80 MHz
- Four independent memory channels
- Support SDRAM and SyncFlash® memory
- Can use registered DIMM
- Selectable data bus width for each channel: 32-bit/16-bit
- Supports critical word first access of the TX49/H2 core
- Supports DMAC special Burst access (address decrement/fix)
- Programmable SDRAM timing latency
  Can set timing to match the clock frequency used and the memory speed. Can realize a system with optimized memory performance.
- Can write to any byte during Single or Burst Write operation. This feature is controlled by the DQM signal.
- Can set the refresh cycle to be programmable.
- SDRAM refresh mode: both auto refresh and self refresh are possible.
- Low power consumption mode: can select between self refresh or pre-charge power down
- SDRAM Burst length: fixed to "2"
- SDRAM addressing mode: Fixed to the Sequential mode
- Supports systems with high fan-out
  Supports two selectable data read-back buses and supports the Slow Write Burst Mode in order to handle data buses with large load. In order to maintain timing consistency during Read operation, it is possible to select whether to use the feedback clock to latch data or to by-pass this latch path. Two clock cycles are used for each Write operation when in the Slow Write Burst Mode.

9.2 Block Diagram



Note: Address signals for SDRAM are ADDR[19:16], SADDR10, and ADDR[14:5]. Don't use ADDR[15] for SDRAM.

Figure 9.2.1 Block Diagram of SDRAMC

9.3 Detailed Explanation

9.3.1 Supported SDRAM Configurations

This controller supports the SDRAM Configurations listed below in Table 9.3.1.

The MW field of the SDRAM Channel Control Register (SDCCRn) can be used to separately set the data bus width for each channel to either 32 bits or 16 bits.

DATA[15:0] and DQM[1:0] are used when using a 16-bit data bus. DQM[3:2] output High. DATA[31:16] output an undefined value when DATA[15:0] become the output, but enter the High-Z state when DATA[15:0] are the input. When in the Big Endian Mode, first external access of the upper half word (bits 31:16) of the internal data bus is performed, then external access of the lower half word (bits 15:0) is performed. When in the Little Endian Mode, first external access of the lower half word (bits 15:0) is performed, then external access of the upper half word (bits 31:16) is performed. When using a 16-bit data bus, two external access will always be performed even when accessing less than 16 bits of data.

The maximum memory capacity per channel when a 32-bit data bus is configured is 512 MBytes when using 8 512-Mbit SDRAMs with a 4-bit data bus. The total maximum memory capacity is 2 GBytes when totaling up the four channels.

Table 9.3.1 Supported SDRAM Configurations

SDRAM Configuration			Row Address (bit)	Column Address (bit)	Remarks
16 Mbit	2-bank	1 M × 16	11	8	
		2 M × 8	11	9	
		4 M × 4	11	10	See Note
64 Mbit	2-bank	2 M × 32	11	9	
		2 M × 32	12	8	
		4 M × 16	11	10	
		4 M × 16	13	8	
		8 M × 8	13	9	
		16 M × 4	13	10	See Note
	4-bank	2 M × 32	11	8	
		4 M × 16	12	8	
		8 M × 8	12	9	
		16 M × 4	12	10	See Note
128 Mbit	4-bank	4 M × 32	12	8	
		8 M × 16	12	9	
		16 M × 8	12	10	
		32 M × 4	12	11	See Note
256 Mbit	4-bank	8 M × 32	13	8	
		16 M × 16	13	9	
		32 M × 8	13	10	
		64 M × 4	13	11	See Note
512 Mbit	4-bank	32 M × 16	13	10	
		64 M × 8	13	11	
		128 M × 4	13	12	See Note

Note1: The SDRAM Controller logic-wise does support these configurations, but please design carefully since the memory bus load will be large.

Note2: With this composition, since the memory size of one channel is 512 Mbyte, if the memory area is mapped from physical address 0, it will overlap with the address area which is used for ROM area of boot vector.

9.3.2 Address Mapping

9.3.2.1 Physical Address Mapping

It is possible to map each of the four channels to an arbitrary physical address using the Base Address field (SDCCRn.BA[31:21]) of the SDRAM Channel Control Register and the Address Mask Field (SDCCRn.AM[31:21]).

The channel that becomes True in the following equation is selected.

paddr[31:21] & !AM[31:21] = BA[31:21] & !AM[31:21]

In the above equation, “paddr” represents the accessed physical address, “&” represents the AND of each bit, and “!” represents the logical NOT of each bit.

Operation is undefined when multiple channels are simultaneously selected, or when external bus controllers or PCI controllers are simultaneously selected.

9.3.2.2 Address Signal Mapping (32-bit Data Bus)

Table 9.3.2 shows the address signal mapping when using a 32-bit data bus. B0 is used in the bank selection in memory with a two-bank configuration. [B1:B0] are used in the bank selection in memory with a four-bank configuration. Bits with the description "L/H" output High when performing auto-precharging, or output Low when not performing auto-precharging.

Table 9.3.2 Address Signal Mapping (32-bit Data Bus) (1/2)

Row address width = 11
Column address width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	21	22	22	21	L/H	22	21	9	8	7	6	5	4	3	2
Row Address	21	22	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 11
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	22	22	21	L/H	22	21	9	8	7	6	5	4	3	2
Row Address	22	22	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 11
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	22	22	21	L/H	22	21	9	8	7	6	5	4	3	2
Row Address	23	22	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12
Column Address Width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	23	22	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	22	23	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	22	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	23	24	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 12
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	22	21	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	24	25	22	21	20	19	18	17	16	15	14	13	12	11	10

Table 9.3.2 Address Signal Mapping (32-bit Data Bus) (2/2)

Row Address Width = 12
Column Address Width = 11

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	22	24	L/H	23	22	9	8	7	6	5	4	3	2
Row Address	25	26	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13
Column Address Width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	23	24	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address = 13
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	24	25	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	22	21	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	25	26	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13
Column Address Width = 11

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	26	27	22	25	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	26	27	22	21	20	19	18	17	16	15	14	13	12	11	10

Row Address Width = 13
Column Address Width = 12

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	27	28	26	25	L/H	24	23	9	8	7	6	5	4	3	2
Row Address	27	28	22	21	20	19	18	17	16	15	14	13	12	11	10

9.3.2.3 Address Signal Mapping (16-bit Data Bus)

Table 9.3.3 shows the address signal mapping when using a 16-bit data bus. B0 is used in the bank selection in memory with a two-bank configuration. [B1:B0] are used in the bank selection in memory with a four-bank configuration. Bits with the description "L/H" output High when performing auto-precharging, or output Low when not performing auto-precharging.

Table 9.3.3 Address Signal Mapping (16-bit Data Bus) (1/2)

Row Address Width = 11
Column Address Width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	20	21	21	20	L/H	21	20	8	7	6	5	4	3	2	1
Row Address	20	21	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 11
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	21	21	21	20	L/H	21	20	8	7	6	5	4	3	2	1
Row Address	21	21	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 11
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	21	21	20	L/H	21	20	8	7	6	5	4	3	2	1
Row Address	22	21	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 12
Column Address Width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	21	22	21	20	L/H	22	21	8	7	6	5	4	3	2	1
Row Address	21	22	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 12
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	23	21	20	L/H	22	21	8	7	6	5	4	3	2	1
Row Address	22	23	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 12
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	21	20	L/H	22	21	8	7	6	5	4	3	2	1
Row Address	23	24	21	20	19	18	17	16	15	14	13	12	11	10	9

Table 9.3.3 Address Signal Mapping (16-bit Data Bus) (2/2)

Row Address Width = 12
Column Address Width = 11

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	21	23	L/H	22	21	8	7	6	5	4	3	2	1
Row Address	24	25	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 13
Column Address Width = 8

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	22	23	21	20	L/H	23	22	8	7	6	5	4	3	2	1
Row Address	22	23	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 13
Column Address Width = 9

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	23	24	21	20	L/H	23	22	8	7	6	5	4	3	2	1
Row Address	23	24	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 13
Column Address Width = 10

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	24	25	21	20	L/H	23	22	8	7	6	5	4	3	2	1
Row Address	24	25	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 13
Column Address Width = 11

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	25	26	21	24	L/H	23	22	8	7	6	5	4	3	2	1
Row Address	25	26	21	20	19	18	17	16	15	14	13	12	11	10	9

Row Address Width = 13
Column Address Width = 12

Address Bit ADDR [19:5]	19 (B0)	18 (B1)	17	16	SAD DR10 (AP)	14	13	12	11	10	9	8	7	6	5
Column Address	26	27	25	24	L/H	23	22	8	7	6	5	4	3	2	1
Row Address	26	27	21	20	19	18	17	16	15	14	13	12	11	10	9

9.3.3 Initialization of SDRAM

The TX4925 Command Register has functions for generating the cycles required for initializing SDRAM and SyncFlash. Using software to set each register makes it possible to execute initial settings at a particular timing. However the Set Mode Register command should be applied separately to the SDRAM and SyncFlash channels.

(1) Set the SDRAM Channel Control Register (SDCCRn).

(2) Set the SDRAM Timing Register (SDCTR). This timing setting is applied to all channels, so please set it to the slowest memory device.

(3) Use the SDRAM Command Register (SDCCMD) to issue the Pre-charge All command.

(4) Issue the Set Mode Register command in the same manner.

(5) Set the refresh count required to initialize SDRAM to the refresh counter (SDCTR.RC)[1] and set the refresh cycle (SDCTR.RP).[2] [3]

(6) Wait until the refresh counter returns to "0."

(7) Set the refresh cycle (SDCTR.RP) to the proper value.

[1] The number of refresh operations can be counted using the refresh counter. With this function, it is no longer necessary to assemble special timing groups in the software when counting refresh operations.

[2] Setting the refresh cycle to a small value makes it possible to expedite completion of the refresh cycle required for SDRAM initialization. As described above, please set normal values after the required number of refresh cycles have been generated.

[3] Refresh requests have priority over all other SDRAM Controller access requests. Please do not set the memory refresh cycle to an unnecessarily short value.

9.3.4 Low Power Consumption Function

9.3.4.1 Power Down Mode, Self-Refresh Mode, Deep Power Down Mode

SDRAM has two low power consumption modes called the Power Down mode and the Self-Refresh mode. Memory data is lost in the case of the Power Down mode since Memory Refresh is not performed, but the amount of power consumed is reduced the most. Memory data is not lost in the case of the Self-Refresh mode.

SDRAM is set to the Power Down mode by using the SDRAM Command Register (SDCCMD) to issue the Power Down Mode command. Similarly, SDRAM is set to the Self-Refresh mode by issuing the Self-Refresh Mode command. The SDRAMC terminates internal refresh circuit operation after one of these commands has been issued. Issuing the Normal Mode command returns operation to normal.

When the Power Down Auto Entry bit (SDCTR.PDAE) of the SDRAM Timing Register is set, SDRAM is automatically set to the Power Down mode when memory access is not being performed. The SDRAMC internal refresh circuit will continue operating, so there will be no loss of memory data.

If either the Memory Access, Memory Refresh, or Memory command is executed while SDRAM is set to the Power Down mode or the Self-Refresh mode, then the Power Down mode and Self-Refresh mode will automatically terminate, and memory access will be performed.

After returning from a low power consumption mode that was set by either the Power Down Mode command or the Self-Refresh Mode command, the next memory access starts after 10 SDCLK cycles pass. This latency sufficiently follows the stipulated time from Power Down to first access of the SDRAM.

If setting the Power Down Auto Entry bit automatically causes memory access to be requested when set in the Power Down mode, then add 1 SDCLK cycle more of access latency than when not in the Power Down mode.

SyncFlash has two low power consumption modes called the Power Down mode and the Deep Power Down mode. Memory data is not lost in both mode. SyncFlash need the device power-up initialize sequence in the case of the Deep Power Down mode, but the amount of power consumed is reduced the most.

9.3.4.2 Advanced CKE

Advanced CKE is a function that speeds up the CKE assertion and deassertion timing by 1 clock cycle. This function is set using the Address CKE bit (SDCTR.ACE) of the SDRAM Timing Register.

Advanced CKE assumes that it will be used in a system where SDRAM data is saved even when the power to the TX4925 itself is cut. Since CKE On/Off becomes 1 cycle faster, it is possible to delay CKE by 1 clock cycle using external power consumption control logic. Please set the SDRAM to the Self-Refresh mode before using this function.

When combining advanced CKE functionality with Power Down Auto Entry functionality and memory access is requested while in the Power Down mode, two more SDCLK cycles of latency are added than would be the case when not in the Power Down mode.

9.3.5 Bus Errors

The SDRAMC detects bus errors in the following situations:

- Bus time-out occurs during Read or Write operation to the SDRAMC

If a bus error occurs when accessing the SDRAMC, then the SDRAMC will immediately abort current operation. Then, the current SDRAM cycle will end, remaining SDRAMC operations will be aborted, a Pre-charge All command will be issued to SDRAM, then the SDRAMC will return to the Idle state.

9.3.6 Memory Read and Memory Write

The RAS* signal, CAS* signal, WE*, signal, ADDR[19:16], SADDR10, and ADDR[14:5] signal are set up 1 cycle before the SDCS* signal is asserted in the case of the Read command, Write command, Pre-charge command, or Mode Register Set command. The same set up time is observed even for active commands if the Active Command Ready bit (SDCTR.DA) of the SDRAM Timing Register is set. Figure 9.5.1 is a timing diagram of Single Read operation when the SDCTR.DA bit is cleared. Figure 9.5.2 is a timing diagram of Single Read operation when the SDCTR.DA bit is set.

Burst or Single Read operation is terminated by the Pre-charge Active Bank command. Burst or Single Write operation is terminated by the Auto Pre-charge Command.

9.3.7 Slow Write Burst

When the Slow Write Burst bit (SDCTR.SWB) of the SDRAM Timing Register is cleared, the data changes at each cycle during Burst Write operation (Figure 9.5.6). When the Slow Write Burst bit is set, the data will change every other cycle (Figure 9.5.7).

When Slow Write Burst bit is set, it always operates as $t_{RCD} = 3t_{CK}$ against all write access in no relation with the value of RAS-CAS Delay bit (SDCTR.RCD) of SDRAM Timing Register. When slow write burst is invalid, the value of RAS-CAS Delay bit is valid. During read access the value of RAS-CAS Delay bit is valid in no relation with Slow Write Burst bit setup.

9.3.8 Clock Feedback

When performing Read access at fast rates like 80 MHz, there may be insufficient set up time if an attempt to directly latch Read data with the internal clock is made. With the TX4925, it is possible to latch data using SDRAM clock SDCLKIN that is input from outside the chip. Please connect SDCLKIN to one of the SDCLK[1:0] pins and the external source.

9.3.9 SyncFlash ®

SDRAMC supports SyncFlash Memory. This memory is drop-in compatible with SDRAM for reads and require a special command sequence for writes and other control operations. The Low power consumption function is some different from SDRAM. Please refer 9.3.4 in detail.

A write to SFCMD register performs a Load Command Register (LCR) Cycle that is the first cycle of SyncFlash memory command sequences to the SyncFlash device. The SyncFlash are fully supported in 16-bit mode for reads and writes (programming) with one exception. In order to read the “Device ID” or “Device protect bit” using the Read Device Configuration LCR operation, the Memory Width field for SyncFlash channel should be briefly set to 32-bit mode (SDCCRn.MW = 0). The relevant data will be on the lower data bus, Data[15:0]. This allows the SDRAMC to correctly generate the odd address necessary to read these SyncFlash configuration registers. All other LCR commands and reads and

writes function normally in 16-bit mode.

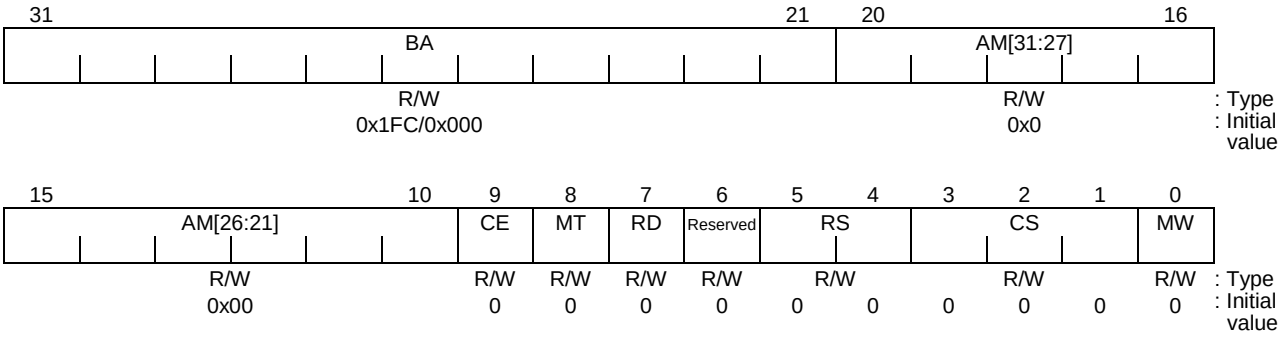
This is caused by two external accesses always being performed for each data access of 16 bits or less when the bus width is 16 bits. These two bus cycles do not output odd addresses in order to execute Burst transfer. Therefore, device IDs with an odd address and Device Protect bits cannot be accessed.

9.4 Registers

Table 9.4.1 SDRAM Control Register

Reference	Offset Address	Bit Width	Register Symbol	Register Name
9.4.1	0x8000	32	SDCCR0	SDRAM Channel Control Register 0
9.4.1	0x8004	32	SDCCR1	SDRAM Channel Control Register 1
9.4.1	0x8008	32	SDCCR2	SDRAM Channel Control Register 2
9.4.1	0x800C	32	SDCCR3	SDRAM Channel Control Register 3
9.4.2	0x8020	32	SDCTR	SDRAM Timing Register
9.4.3	0x802C	32	SDCCMD	SDRAM Command Register
9.4.4	0x8030	32	SFCMD	SyncFlash Command Register

9.4.1 SDRAM Channel Control Register (SDCCR0) 0x8000 (ch. 0)
(SDCCR1) 0x8004 (ch. 1)
(SDCCR2) 0x8008 (ch. 2)
(SDCCR3) 0x800C (ch. 3)



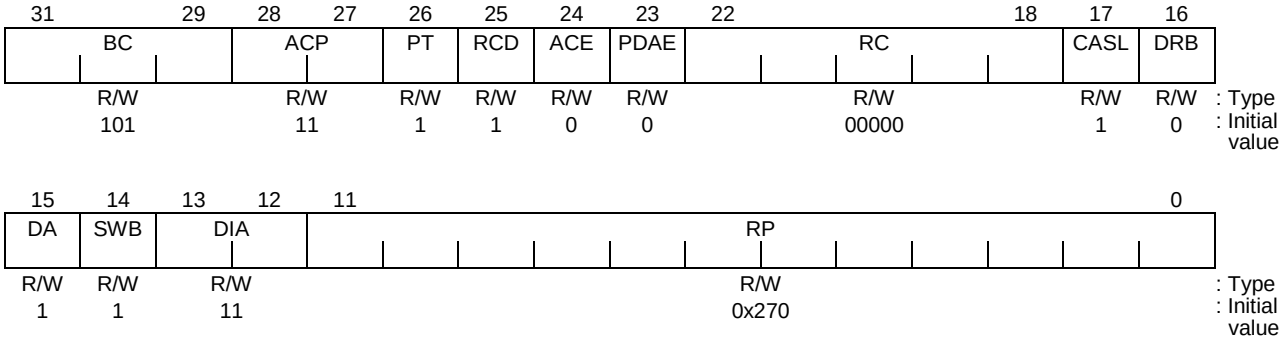
Bits	Mnemonic	Field Name	Description
31:21	BA[31:21]	Base Address	Base Address (Initial value: (0x1FC & (ADDR[8:6]==000)/0x000, R/W) Specifies the base address. The upper 11 bits [31:21] of the physical address are compared to the value of this field. (Note) When ADDR[8:6] is 000b: 0x1FC (select SyncFlash as the Boot memory) For all other ADDR[8:6] values: 0x000
20:10	AM[31:21]	Address Mask	Address Mask (Initial value: 0x000, R/W) Sets the valid bits for address comparison according to the base address. 0: Bits of the corresponding BA field are compared. 1: Bits of the corresponding BA field are not compared.
9	CE	Channel Enable	Enable (Initial value: (ADDR[8:6]==000)/0, R/W) Specifies whether to enable a channel. If ADDR[8:6] are 000 during the boot sequence to select SyncFlash as the boot memory, the value in channel 0 become 1. 0: Disable 1: Enable
8	MT	Memory Type	Memory Type (Initial value: (ADDR[8:6]==000)/0, R/W) Specifies whether SDRAM or SynvFlash. If ADDR[8:6] are 000 during the boot sequence to select SyncFlash as the boot memory, the value in channel 0 become 1. 0: SDRAM 1: SyncFlash
7	RD	Registered DIMM	Registered DIMM (Initial value: 0, R/W) Specifies whether the SDRAM connected to the channel is Registered memory. 0: Disable Registered memory 1: Enable Registered memory
6	—	Reserved	Note: this bit is always set to “0” (Initial value: 0, R/W)
5:4	RS	Row Size	Row Size (Initial value: ((-ADDR[5], ADDR[2]) & (ADDR[8:6]==000))/00, R/W) Specifies the row size. If ADDR[8:6] are 000 during the boot sequence to select SyncFlash as the boot memory, the value of ADDR[5,2] during the boot sequence are set in these bits. 00: 2048 Rows (11 bits) 01: 4096 Rows (12 bits) 10: 8192 Rows (13 bits) 11: Reserved (Note) When ADDR[8:6] is 000b: Up to ADDR[5], ADDR[2] (select SyncFlash as the Boot memory) For all other ADDR[8:6] values: 00

Figure 9.4.1 SDRAM Channel Control Register (1/2)

Bits	Mnemonic	Field Name	Description
3:1	CS	Column Size	Column Size (Initial value: {0,(ADDR[11,12] & (ADDR[8:6]==000))}/000, R/W) Specifies the column size. If ADDR[8:6] are 000 during the boot sequence to select SyncFlash as the boot memory, the value of ADDR[11,12] during the boot sequence are set in CS[1:0]. CS[2] is zero. 000: 256 words (8 bits) 001: 512 words (9 bits) 010: 1024 words (10 bits) 011: 2048 words (11 bits) 100: 4096 words (12 bits) 101 – 111: Reserved
0	MW	Memory Width	Memory Width (Initial value: (ADDR[13] & (ADDR[8:6]==000))/0, R/W) Specifies the bus width. If ADDR[8:6] are 000 during the boot sequence to select SyncFlash as the boot memory, the value of ADDR[13] during the boot sequence is set in this bit. 0: 32 bits 1: 16 bits

Figure 9.4.1 SDRAM Channel Control Register (2/2)

9.4.2 SDRAM Timing Register (SDCTR) 0x8020



Bits	Mnemonic	Field Name	Description
31:29	BC	Bank Cycle Time	Bank Cycle Time (t_{RC}) (Initial value: 101, R/W) Specifies the bank cycle time.(*2) 000: 5 t_{CK}(*1) 100: 9 t_{CK} 001: 6 t_{CK} 101: 10 t_{CK} 010: 7 t_{CK} 110: Reserved 011: 8 t_{CK} 111: Reserved
28:27	ACP	Active Command Time	Active Command Period (tRAS) (Initial value: 11, R/W) Specifies the active command time. 00: 3 t_{CK} 01: 4 t_{CK} 10: 5 t_{CK} 11: 6 t_{CK}
26	PT	Precharge Time	Precharge Time (tRP) (Initial value: 1, R/W) Specifies the precharge time. 0: 2 t_{CK} 1: 3 t_{CK}
25	RCD	RAS-CAS Delay	RAS to CAS Delay (tRCD) (Initial value: 1, R/W) Specifies the RAS - CAS delay. 0: 2 t_{CK} 1: 3 t_{CK}
24	ACE	Advanced CKE	Advanced CKE enable (Initial value: 0, R/W) Enabling this function makes the timing at which CKE changes one cycle earlier. 0: Disable 1: Enable
23	PDAE	Power Down Auto Entry	Power Down Auto Entry Enable (Initial value: 0, R/W) Enabling this function makes CKE become “L” while the SDRAMC is in the Idle state. When refresh, memory access, or command execution is performed, CKE automatically becomes “H”, the requested operation is performed, then CKE returns to “L” when the operation is complete. 0: Disable 1: Enable

Figure 9.4.2 SDRAM Timing Register (1/2)

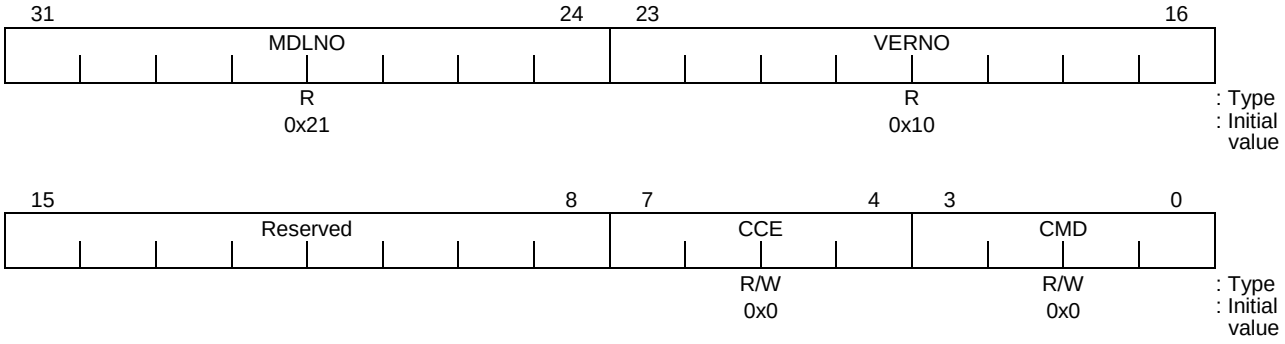
Bits	Mnemonic	Field Name	Description
22:18	RC	Refresh Counter	Refresh Counter (Initial value: 00000, R/W) This counter is decremented at each refresh. If the refresh circuit is activated and a value other than “0” is loaded, this field becomes a down counter that stops at “0”. A value other than “0” must be reloaded to start the countdown again. This is used during memory initialization.
17	CASL	CAS Latency	CAS Latency (t_{CASL}) (Initial value: 1, R/W) Specifies the CAS latency. 0: 2 t_{CK} 1: 3 t_{CK}
16	DRB	Data Read Bypass	Data Read Bypass (Initial value: 0, R/W) Selects the Data Read path used. 0: Data Read latches to the register using the feedback clock. 1: Data Read bypasses the feedback clock latch.
15	DA	Active Command Delay	Delay Activate (t_{DA}) (Initial value: 1, R/W) Specifies the delay from the row address to the bank active command. Setting this bit to “1” sets up the row address two cycles before the active command is executed. 0: 0 t_{CK} 1: 1 t_{CK}
14	SWB	Slow Write Burst	Slow Write Burst (t_{SWB}) (Initial value: 1, R/W) Specifies whether to perform Slow Write Burst. 0: Burst Write occurs at each 1 t_{CK} 1: Burst Write occurs at each 2 t_{CK}
13:12	DIA	Write Active Period	Data In to Active(t_{DAL}) (Initial value: 11, R/W) Specifies the period from the last Write data to the Active command. 00: Reserved 01: 4 t_{CK} 10: 5 t_{CK} 11: 6 t_{CK}
11:0	RP	Refresh Period	Refresh Period (Initial value: 0x270, R/W) Specifies the clock cycle count that generates the refresh cycle. Refresh is only enabled when at least one SDRAM channel is enabled. Please program the Timing Register before an arbitrary channel is enabled. Default is 0x270. A refresh cycle occurs for each 7.8 µs@80 MHz in this situation. The Refresh Cycle count clock is GBUSCLKF. The CCFG.RF value does not change the frequency of GBUSCLKF. Therefore, it is not necessary to change this register value even when CCFG.RF lowers the chip operation clock frequency.

Figure 9.4.2 SDRAM Timing Register (2/2)

*1: t_{CK} = Clock cycle

*2: t_{RC} is used during (i) refresh cycle time, (ii) single Read, (iii) two transfer burst Reads. The bank cycle time is $t_{RAS} + t_{RP} + 1t_{CK}$ if $t_{RAS} + t_{RP} < t_{RC}$ in the case of (ii) (iii).

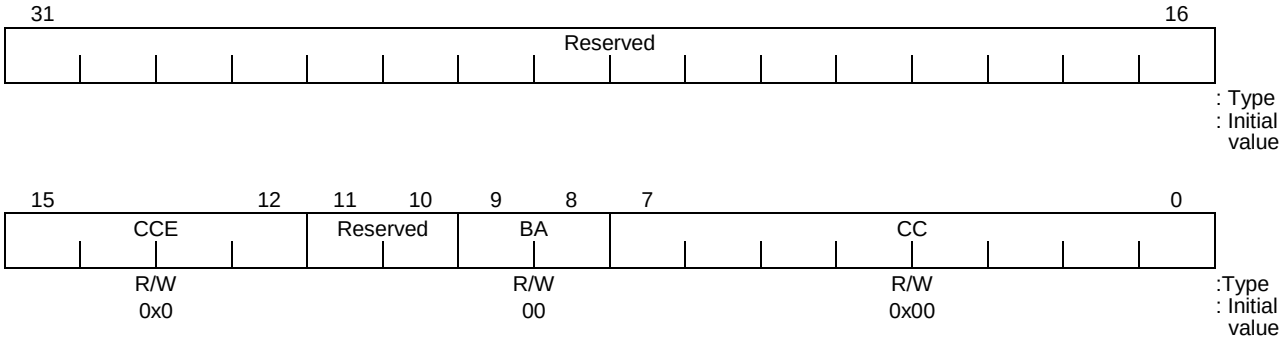
9.4.3 SDRAM Command Register (SDCCMD) 0x802C



Bits	Mnemonic	Field Name	Description
31:24	MDLNO	Model Number	Model Number (Initial value: 0x21, R) Indicates the model number. The default value is 0x30 for the TX4925. This field is Read Only.
23:16	VERNO	Version Number	Version Number (Initial value: 0x10, R) Indicates the version number. The default value is 0x10 for the TX4925. This field is Read Only.
15:8	—	Reserved	—
7:4	CCE	Command Channel Enable	Command Channel Enable (Initial value: 0x0, R/W) Setting one of these bits to “1” enables the command of the corresponding channel. This command is simultaneously executed on all channels that are enabled. Bit 7: Channel 3 Bit 6: Channel 2 Bit 5: Channel 1 Bit 4: Channel 0
3:0	CMD	Command	Command (Initial value: 0x0, R/W) Specifies a command that is performed on memory. 0x0: NOP command 0x1: Set Mode Register command Set SDRAM Mode Register from SDCTR value 0x2: Reserved 0x3: Precharge All command Precharge All SDRAM Banks 0x4: Self-Refresh Mode command Sets SDRAM to the Self-Refresh Mode 0x5: Power Down Mode Command Set SDRAM/SyncFlash to the Power Down Mode 0x6: Normal Mode Command Cancel Self-Refresh/Power Down Mode 0x7: Reserved 0x8: Deep Power Down Mode command for SyncFlash Sets SyncFlash to the Deep Power Down Mode 0x9: Exit Deep Power Down Mode command for SyncFlash Cancel Deep Power Down Mode command for SyncFlash 0xa-0xf: Reserved

Figure 9.4.3 SDRAM Command Register

9.4.4 SyncFlash Command Register (SFCMD) 0x8030



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:12	CCE	Command Channel Enable	Command Channel Enable (Initial value: 0x0, R/W) Setting one of these bits to “1” enables the command of the corresponding channel. This command is simultaneously executed on all channels that are enabled. Bit 15: Channel 3 Bit 14: Channel 2 Bit 13: Channel 1 Bit 12: Channel 0
11:10	—	Reserved	—
9:8	BA	Bank Address	Bank Address (Initial value: 00, R/W) The value is used as Bank Address in the LCR cycle of the SyncFlash command sequence. The value in this field is outputs to ADDR[19:18]
7:0	CMD	Command	Command (Initial value: 0x00, R/W) The value is used as command in the LCR cycle of the SyncFlash command sequence. The value in this field is outputs to ADDR[12:5]

Figure 9.4.4 SyncFlash Command Register

9.5 Timing Diagrams

Please note the following when referring to the timing diagrams in this section: the shaded area in each diagram expresses values that have yet to be determined.

9.5.1 Single Read (32-bit Bus)

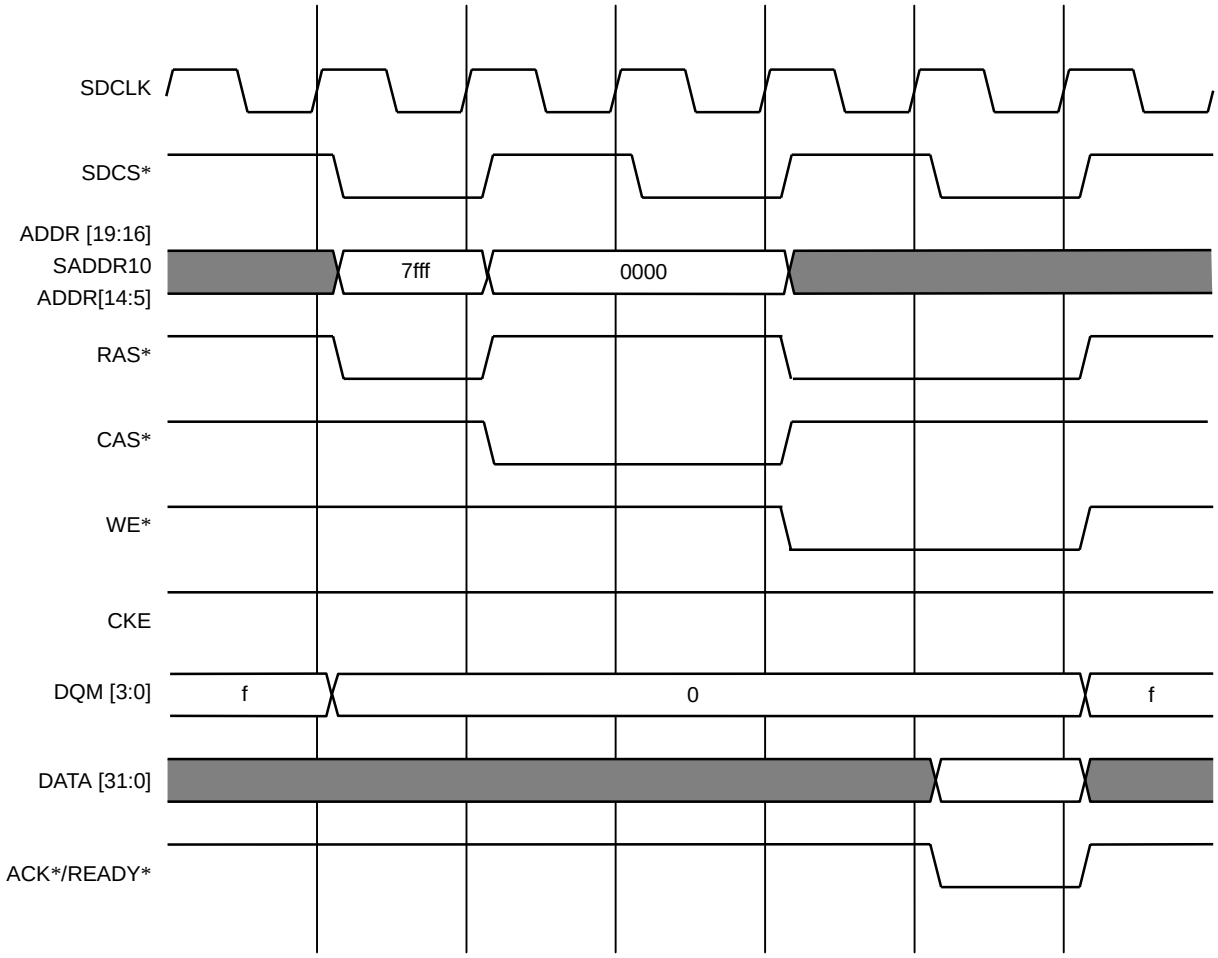


Figure 9.5.1 Single Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 32-bit Bus)

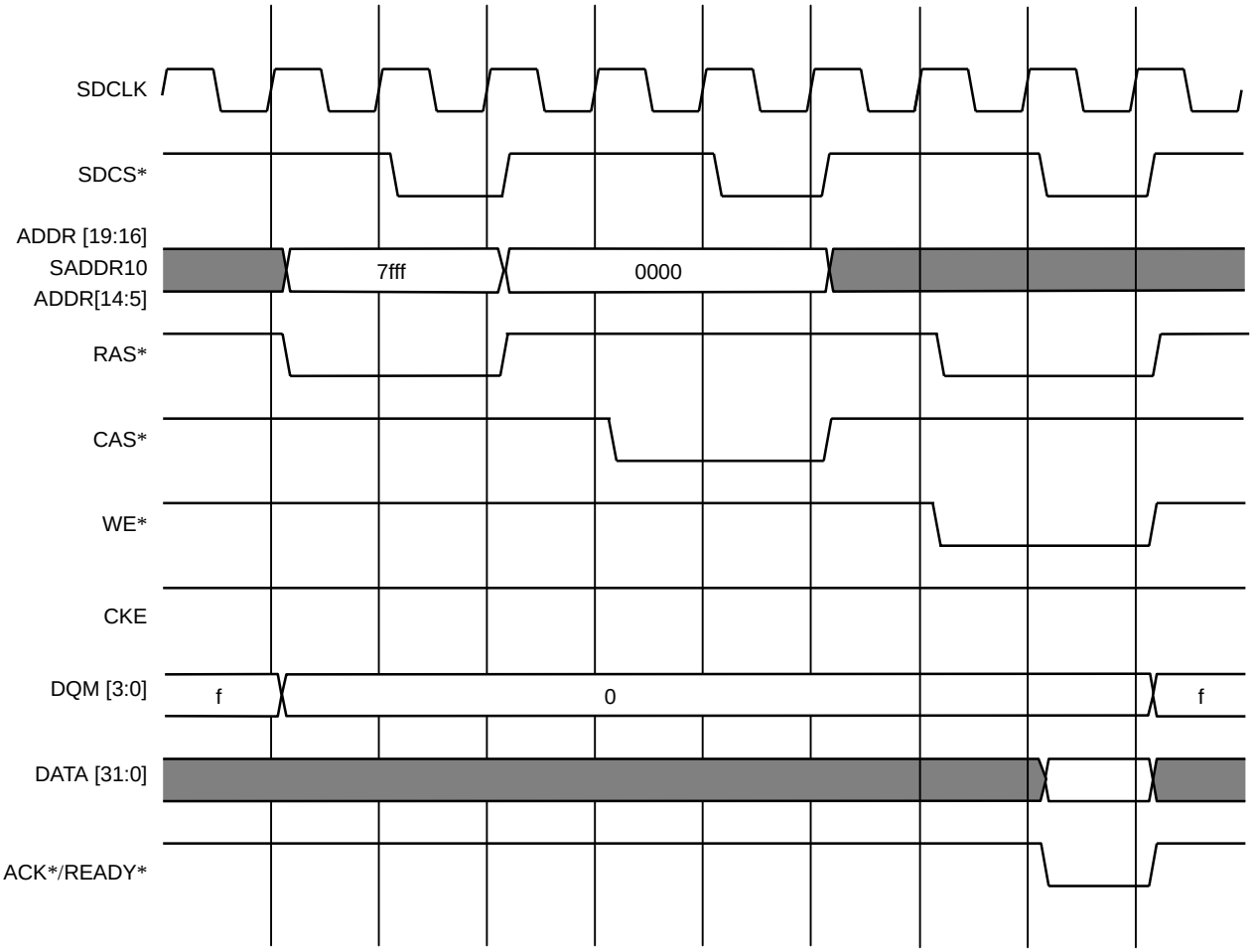


Figure 9.5.2 Single Read (t_{RCD} = 3, t_{CASL} = 3, t_{DA} = 1, 32-bit Bus)

9.5.2 Single Write (32-bit Bus)

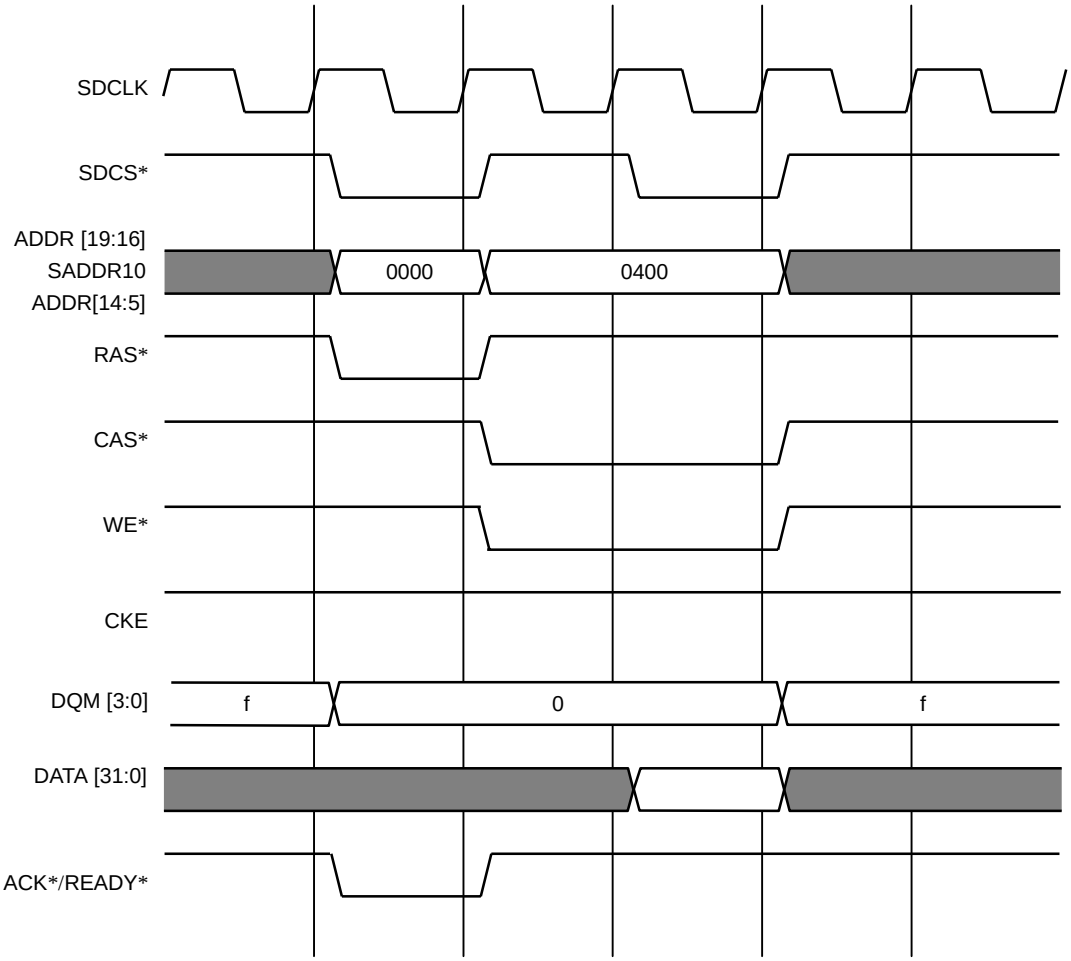


Figure 9.5.3 One-Word Single Write (t_{RCD} = 2, t_{DA} = 0, 32-bit Bus)

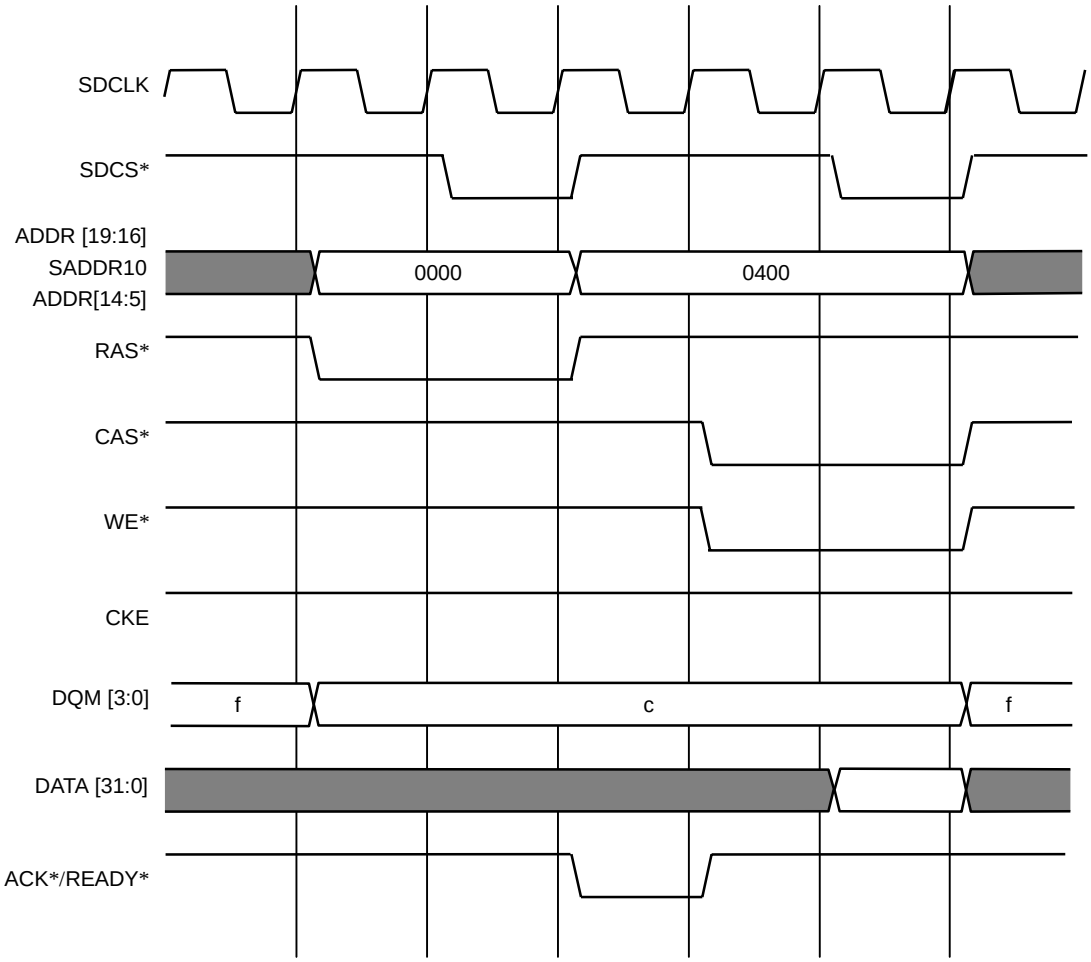


Figure 9.5.4 Half-Word Single Write (t_{RCD} = 3, t_{DA} = 1, 32-bit Bus)

9.5.3 Burst Read (32-bit Bus)

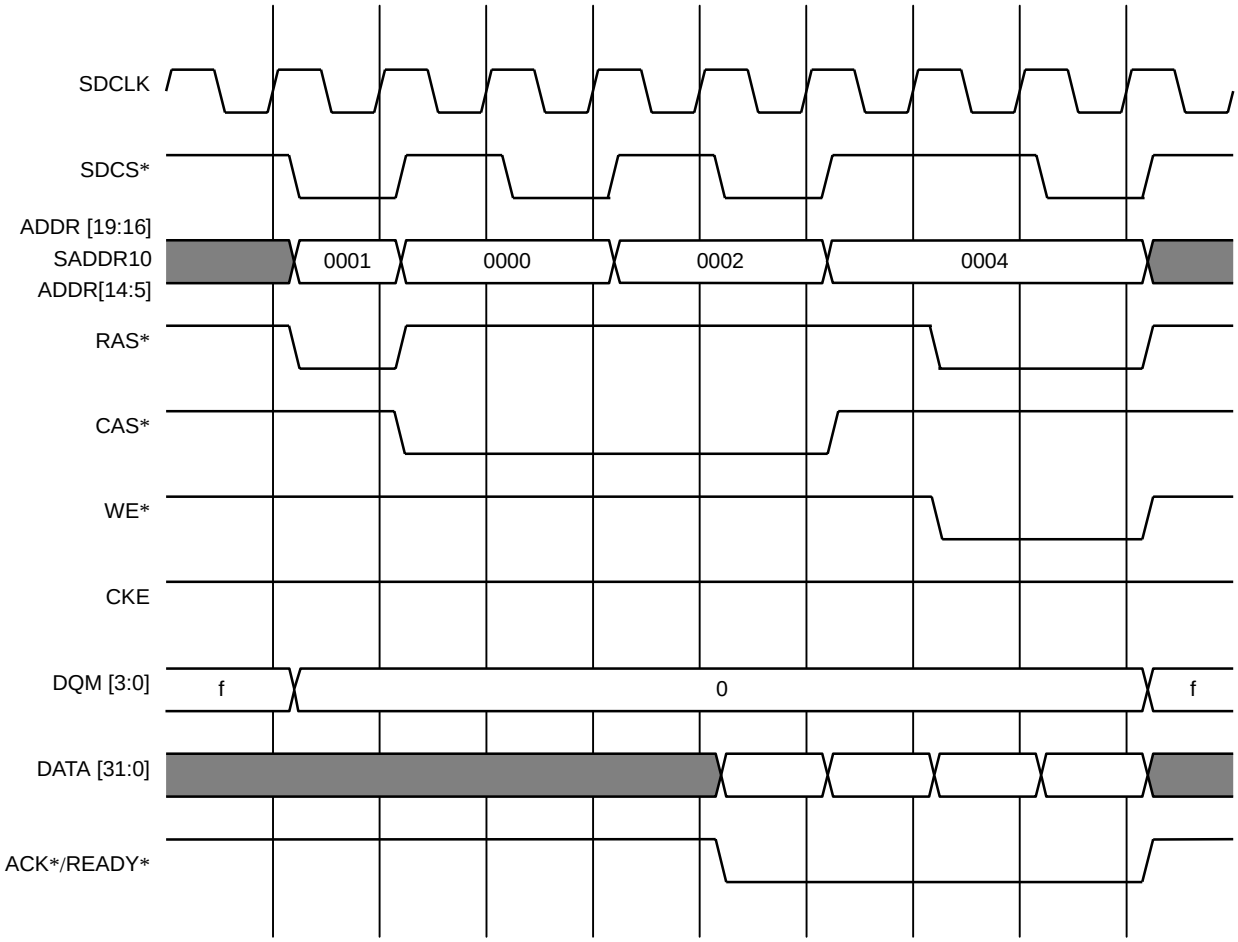


Figure 9.5.5 Four-Word Burst Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 32-bit Bus)

9.5.4 Burst Write (32-bit Bus)

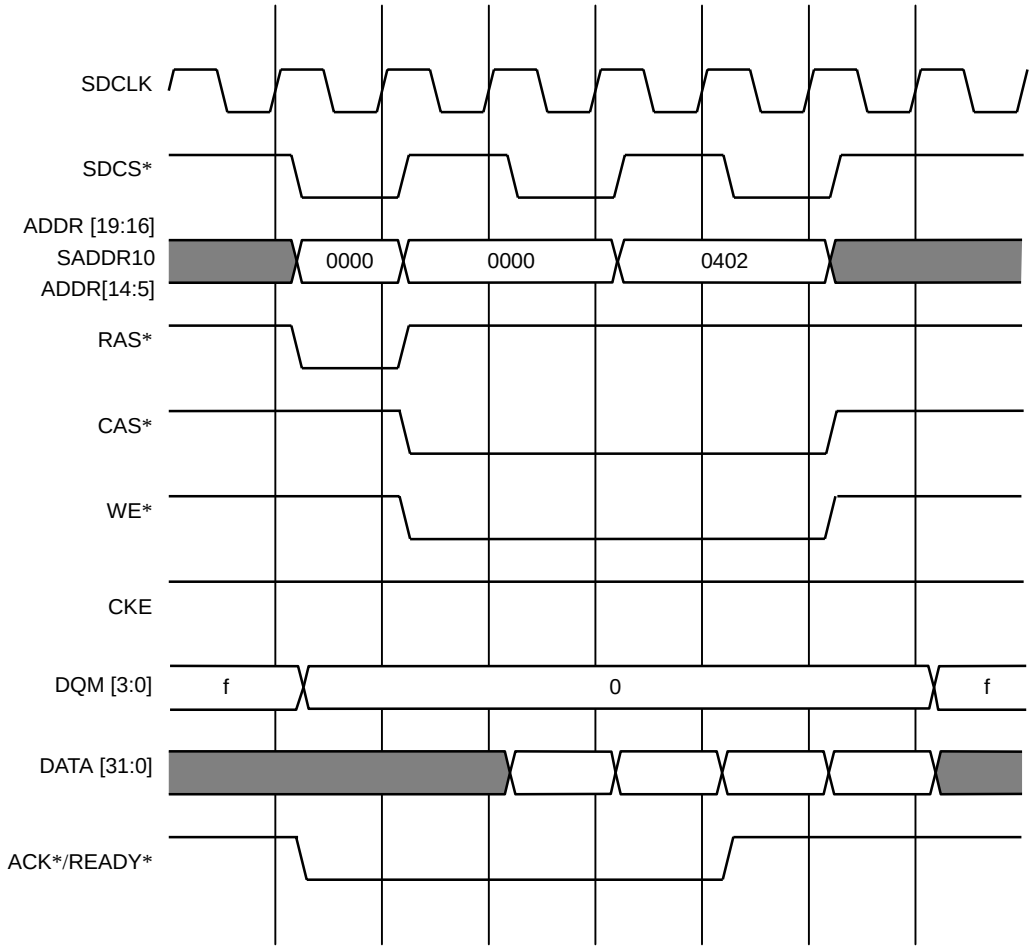


Figure 9.5.6 Four-Word Burst Write (t_{RCD} = 2, t_{DA} = 0, 32-bit Bus)

9.5.5 Burst Write (32-bit Bus, Slow Write Burst)

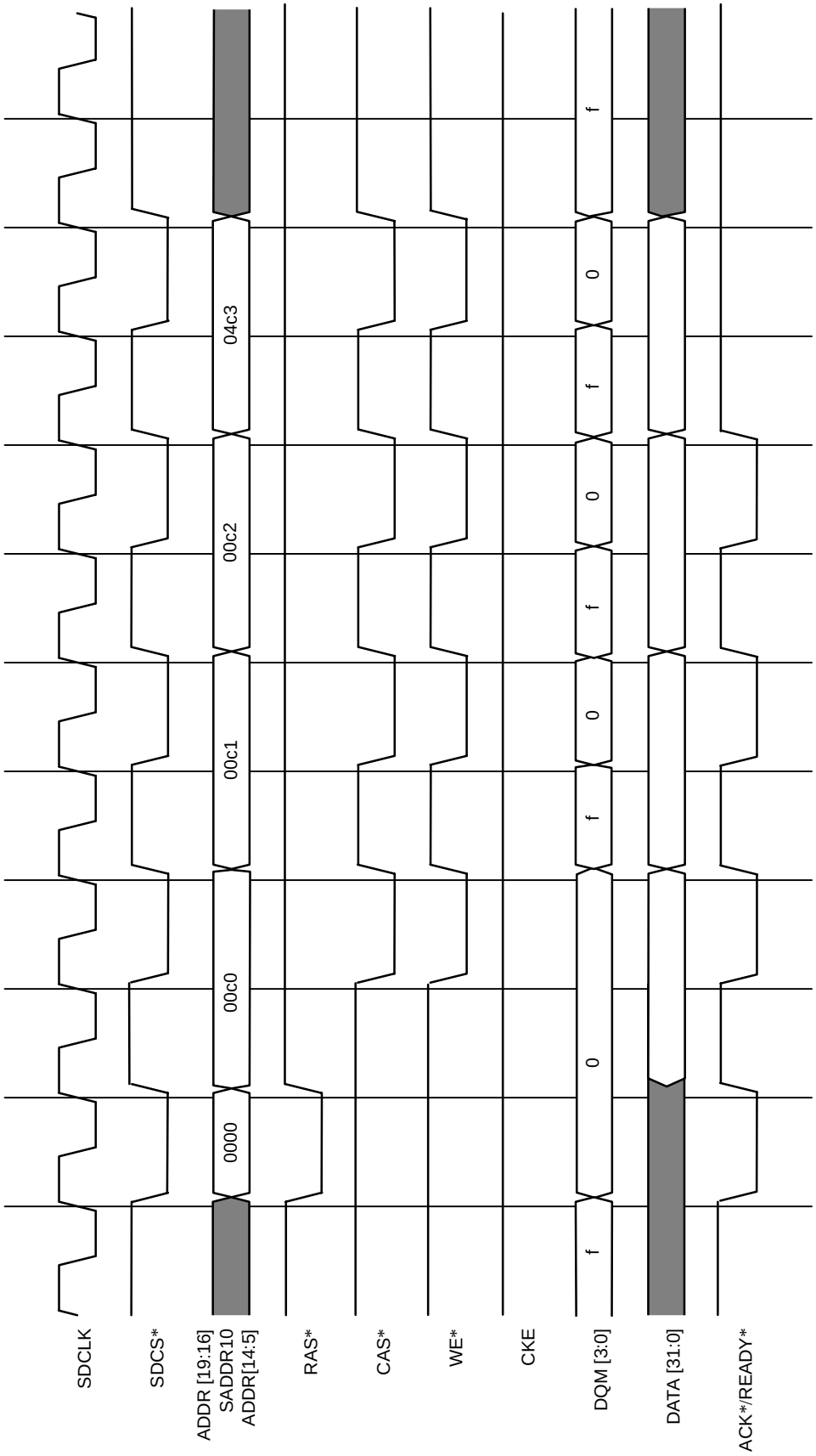


Figure 9.5.7 Four-Word Burst Write (t_{RCD} = 2, t_{DA} = 0, 32-bit Bus, Slow Write Burst)

9.5.6 Single Read (16-bit Bus)

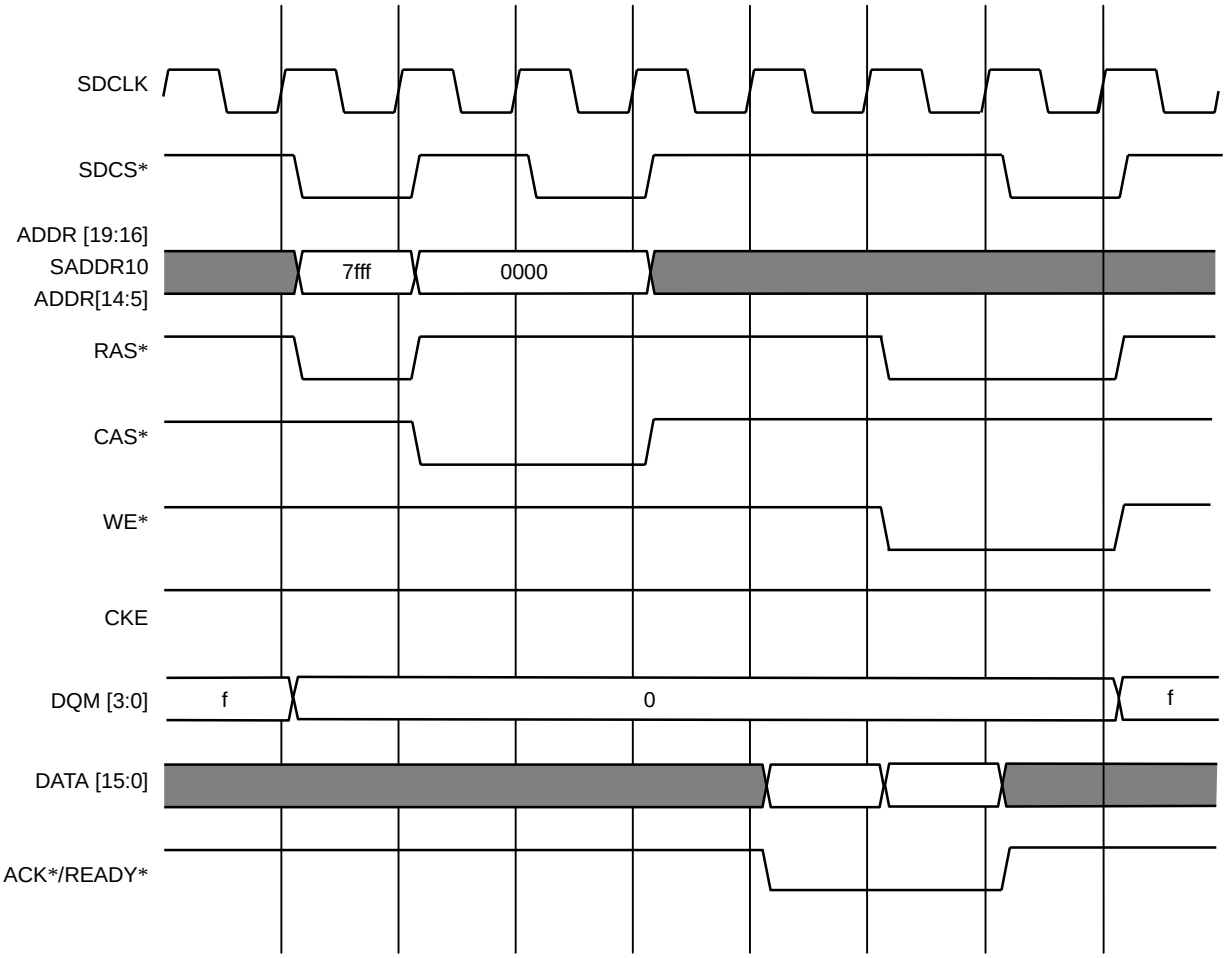


Figure 9.5.8 One Word Single Read (t_{RCD} = 2, t_{CASL} = 2, t_{DA} = 0, 16-bit Bus)

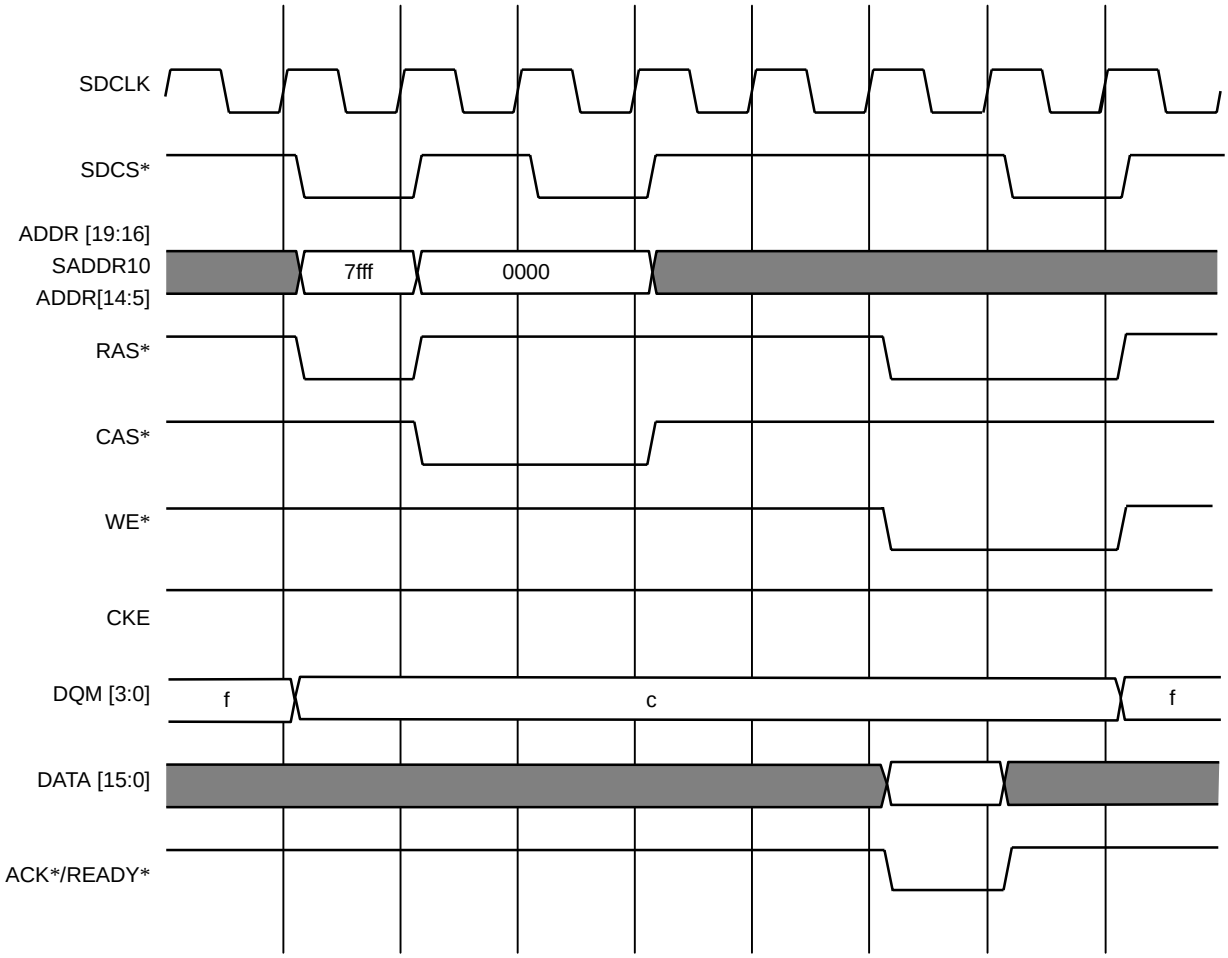


Figure 9.5.9 Half-Word Single Read (t_{RCD} = 2, t_{CASL} = 3, t_{DA} = 0, 16-bit Bus)

9.5.7 Single Write (16-bit Bus)

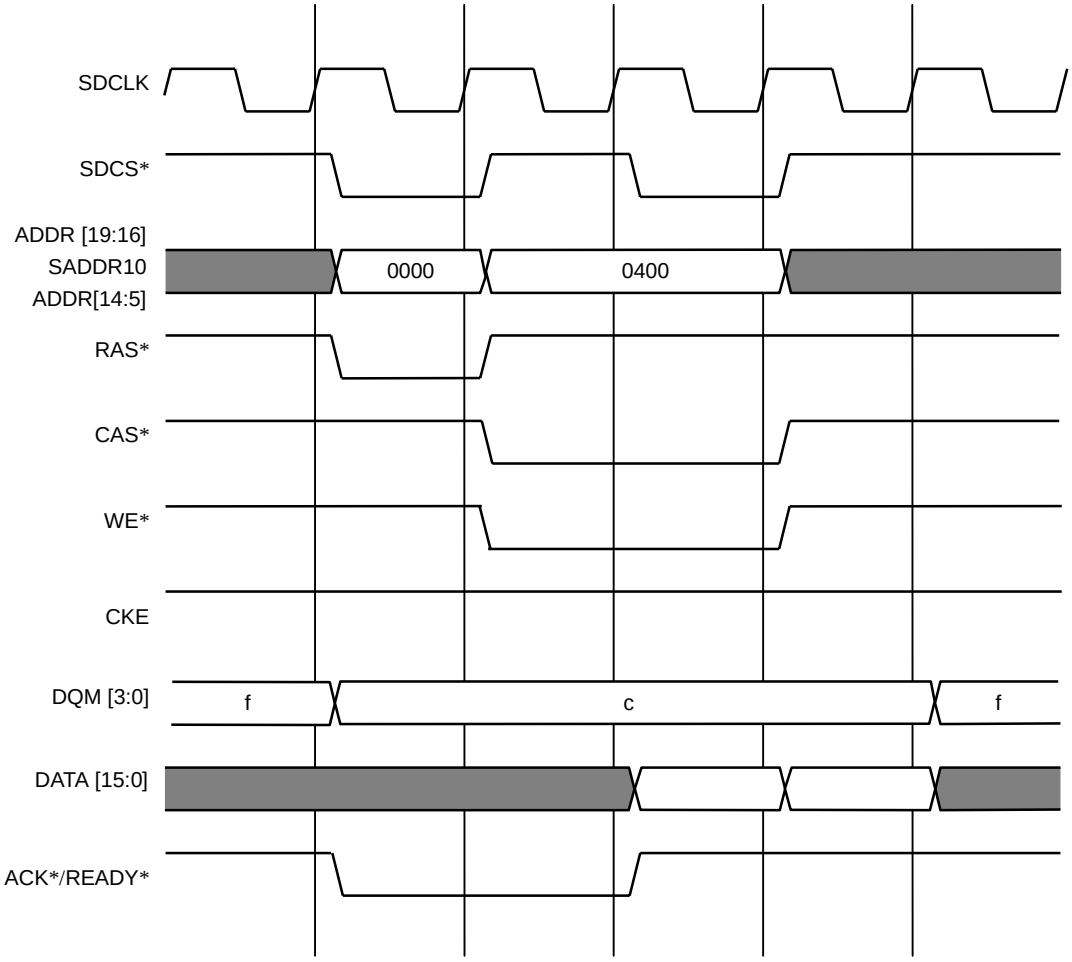


Figure 9.5.10 One-Word Single Write (t_{RCD} = 2, t_{DA} = 0, 16-bit Bus)

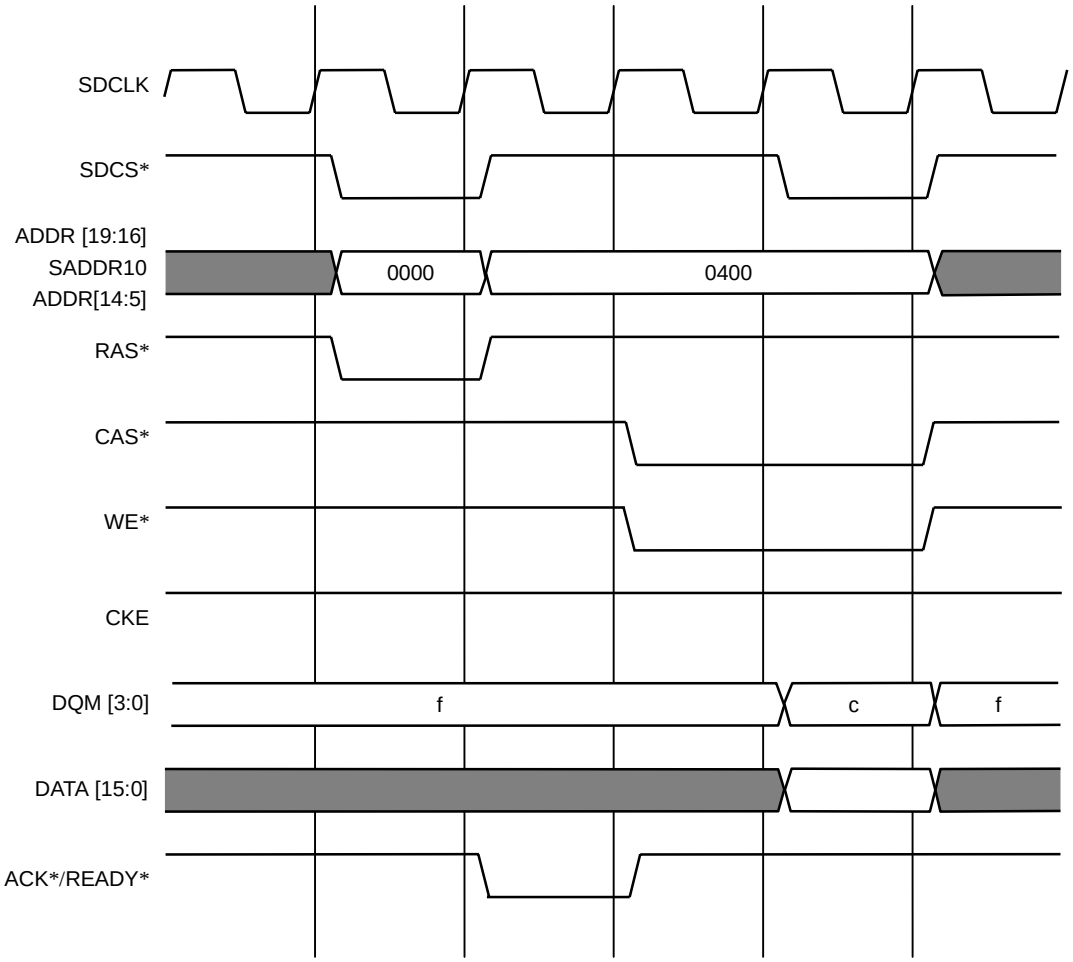


Figure 9.5.11 Half-Word Single Write (t_{RCD} = 3, t_{DA} = 0, 16-bit Bus)

9.5.8 Low Power Consumption and Power Down Mode

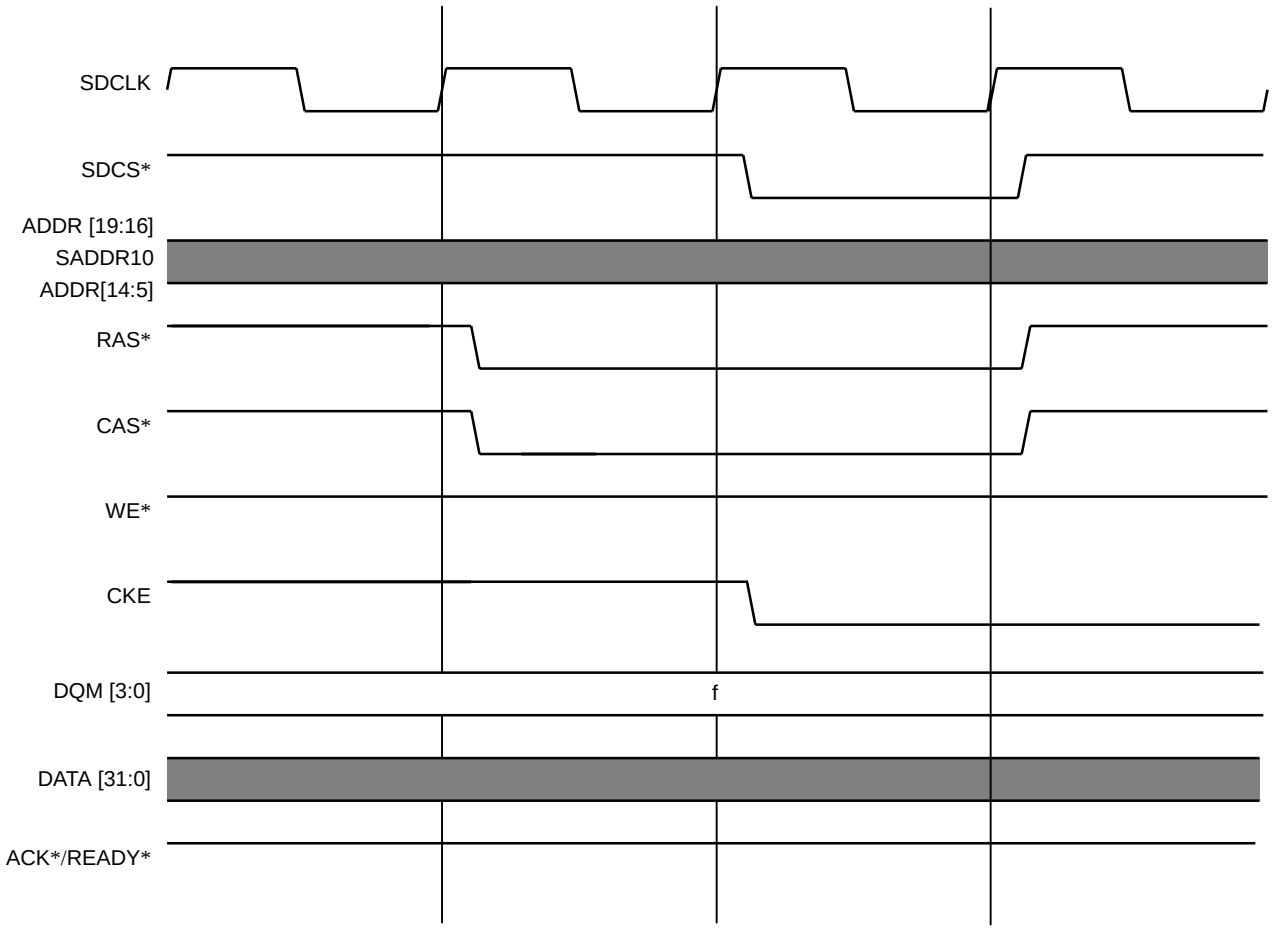


Figure 9.5.12 Transition to Low Power Consumption Mode (SDCTR.ACE = 0)

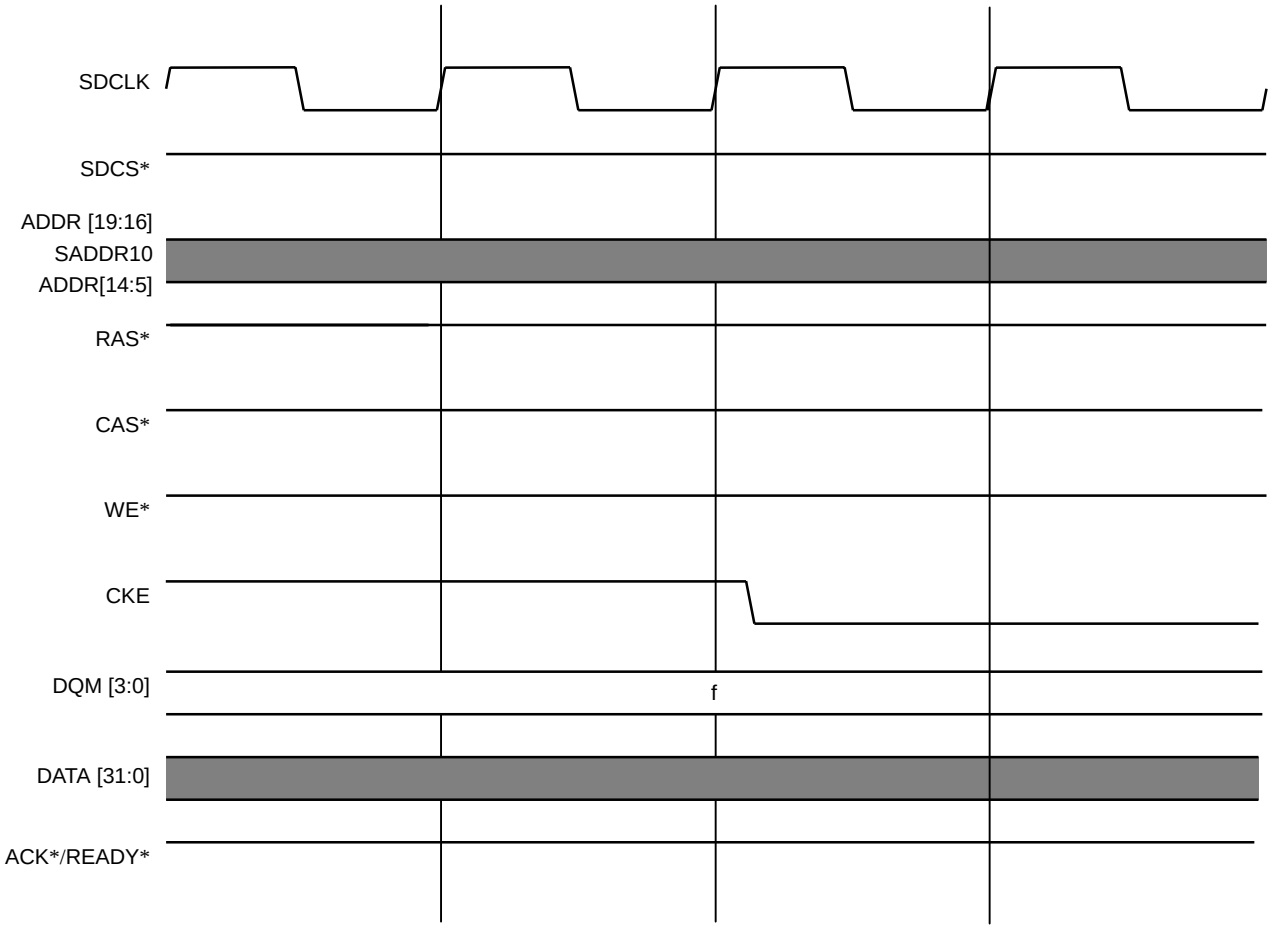


Figure 9.5.13 Transition to Power Down Mode

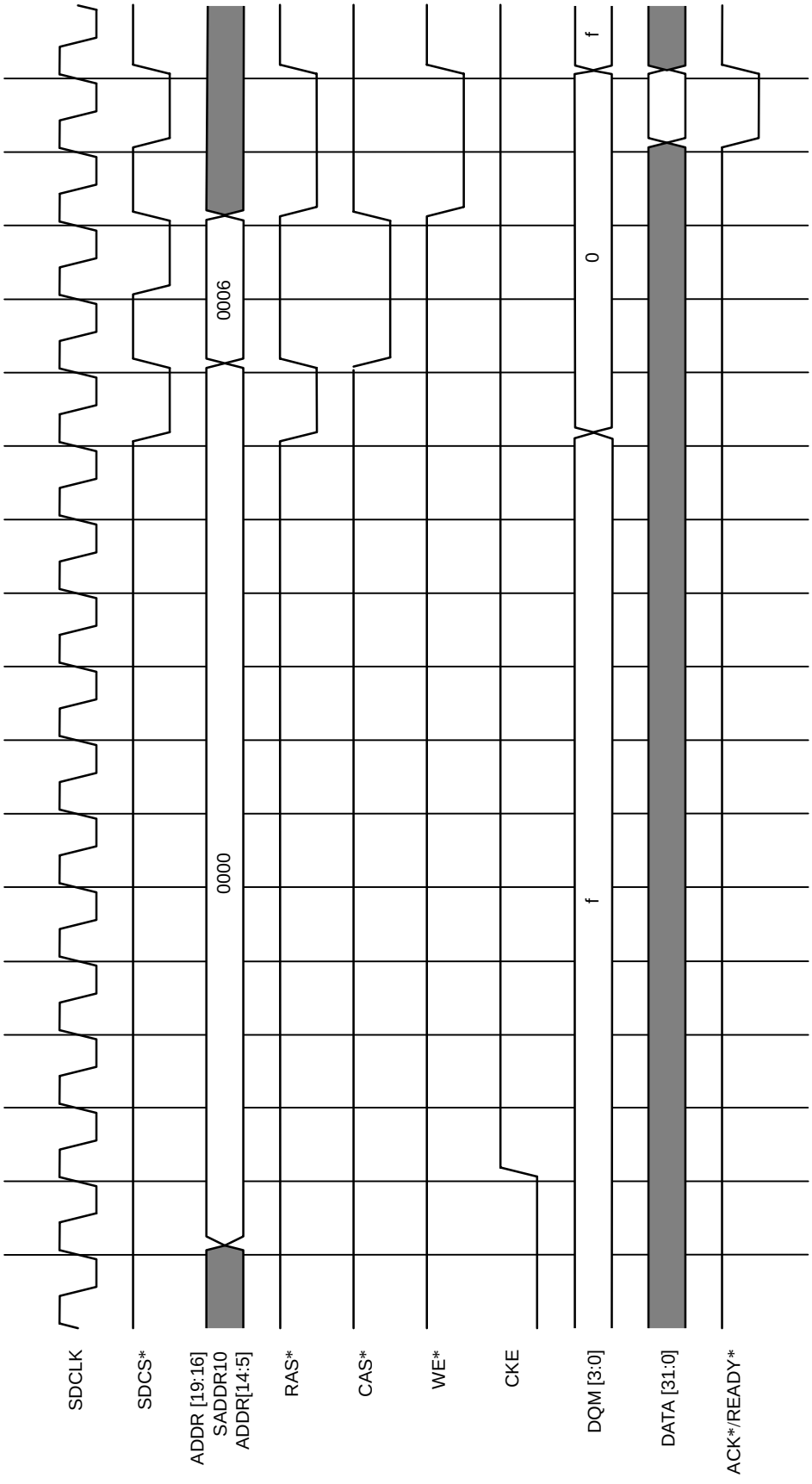


Figure 9.5.14 Return From Low Power Consumption/Power Down Mode (SDCTR.PDAE=0, SDCTR.ACE=0)

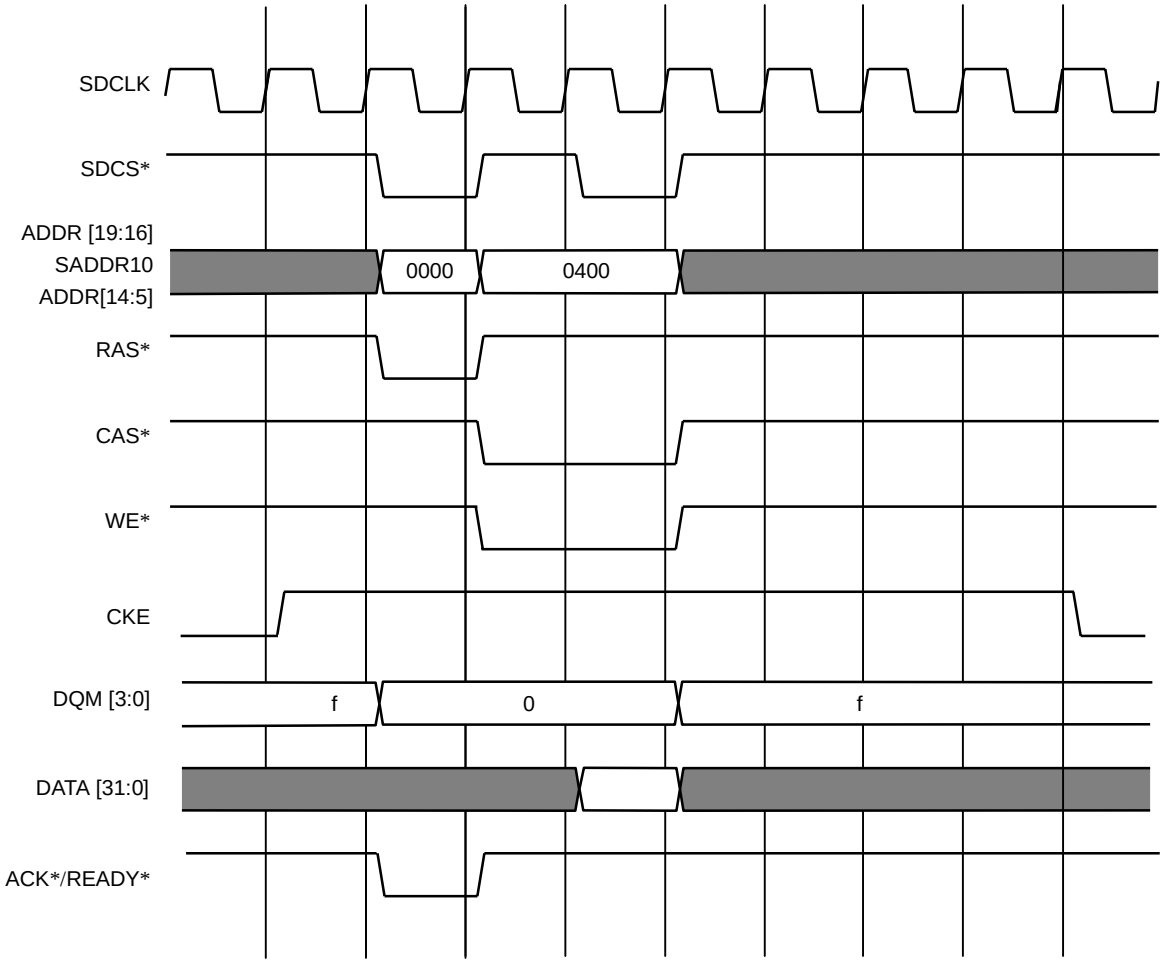


Figure 9.5.15 Power Down Auto Entry (SDCTR.PDAE=1, SDCTR.ACE=0)

9.6 SDRAM Usage Example

Figure 9.6.1 illustrates an example SDRAM connection.

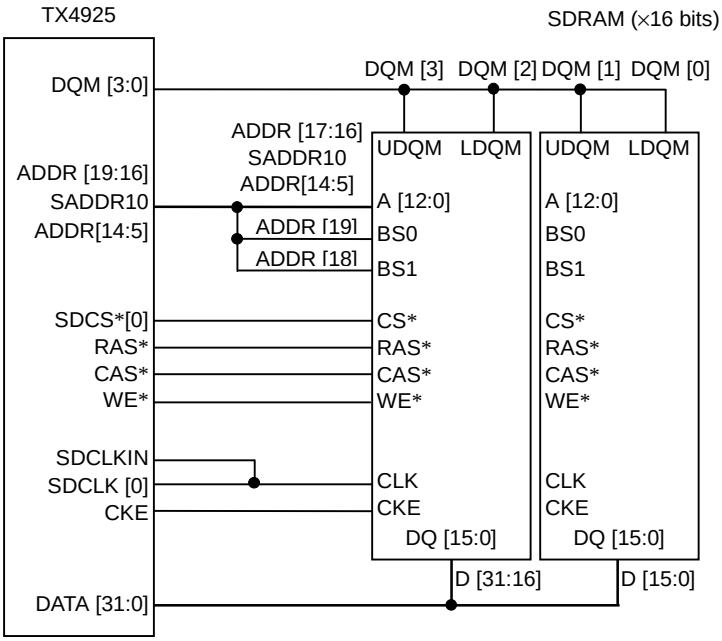


Figure 9.6.1 SDRAM (×16 bits) Connection Example

10. PCI Controller

10.1 Features

The TX4925 PCI Controller functions as a bus bridge between the TX4925 External PCI and the internal bus (G-Bus).

10.1.1 Overall

- Compliant to “PCI Local Bus Specification Revision 2.2”
- PCI Bus: 32-bit data bus; Internal Bus: 32-bit data bus
- Maximum PCI bus clock operating frequency: 33 MHz
- Supports both the Initiator and Target functions
- Supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1 (Partially unsupported).
- On-chip PCI Bus Arbiter, can connect to a maximum of four external bus masters
- 1-channel on-chip DMA Controller (PDMAC) dedicated to the PCI Controller
- Supports PCI clock input mode/output mode
- The Internal Bus clock and PCI Bus clock are asynchronous and can be set independently
- Includes function for booting the TX4925 from memory on the PCI Bus
- Can set configuration data by software at the initialize routine
- Mounted a retry function on the Internal Bus side also in order to avoid deadlock on the PCI Bus.

10.1.2 Initiator Function

- Single and Burst transfer from the Internal Bus to the PCI Bus
- Initiator function Supports memory, I/O, configuration, special cycle, and interrupt acknowledge transactions.
- Address mapping between the Internal Bus and the PCI Bus can be modified
- Mounted 8-stage 32-bit data one FIFO each for Read and Write
- Indirect Read and Write function enables quick termination of single transactions by the G-Bus without waiting for completion on the PCI Bus.
- Endian switching function

10.1.3 Target Function

- Single and Burst transfer from the PCI Bus to the Internal Bus
- Supports memory, I/O, and configuration cycles
- Supports high-speed back-to-back transactions on the PCI Bus
- Address mapping between the PCI Bus and the Internal bus can be modified
- Mounted 32-stage 32-bit data FIFO on each PCI Channel for Read
- Mounted 16-stage 32-bit data FIFO for Write
- Post Write function enables quick termination of a maximum of eight Write transactions by the PCI Bus without waiting for completion on the G-Bus.
- Read Burst length (pre-fetch data size) on the Internal Bus when reading a pre-fetchable space can be made programmable
- Endian switching function

10.1.4 PCI Arbiter

- Supports four external PCI bus masters
- Uses the Programmable Fairness algorithm (two levels with different priorities for four round-robin request/grant pairs)
- Supports bus parking
- Bus master uses the Most Recently Used algorithm
- Unused slots and broken masters can be automatically disabled after Power On reset
- On-chip arbitration function can be disabled and external arbiter can be used

10.1.5 PDMAC (PCI DMA Controller)

- Direct Memory Access (DMA) Controller dedicated to 1-channel PCI
- Is possible to transfer data using minimal G-Bus bandwidth
- Data can be transferred bidirectionally between the G-Bus and the PCI Bus
- Specifying a physical address on the PCI Bus and an address on the G-Bus makes it possible to automatically transfer data between the PCI Bus and the G-Bus
- Supports the Chain DMA mode, in which a Descriptor containing chain-shaped addresses and a transfer size is automatically read from memory while DMA transfer continuous
- On-chip 16-stage 32-bit data buffer

10.2 Block Diagram

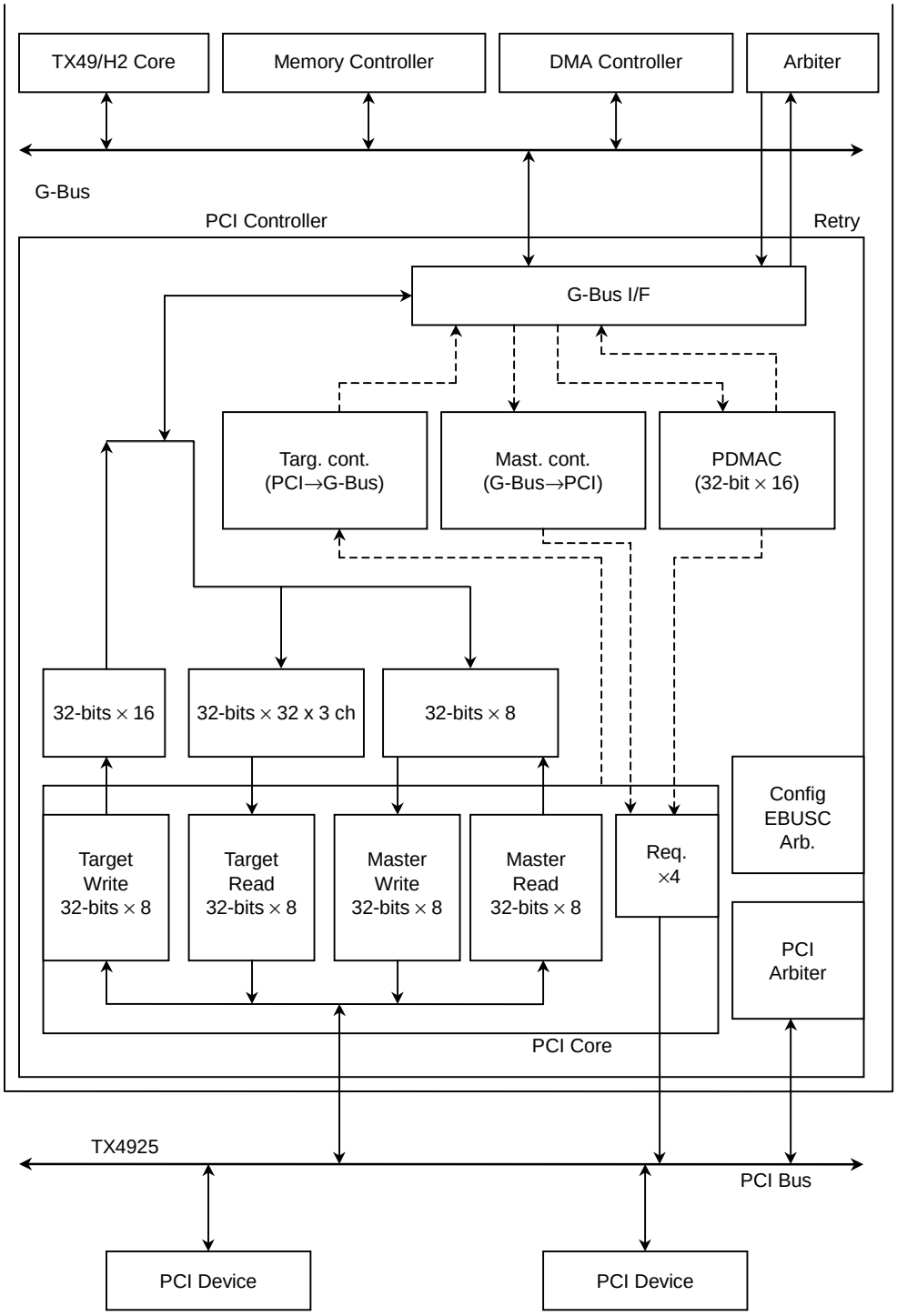


Figure 10.2.1 PCI Controller Block Diagram

10.3 Detailed Explanation

10.3.1 Terminology Explanation

The following terms are used in this chapter.

- Initiator

  Means the bus Master of the PCI Bus. The TX4925 operates as the initiator when it obtains the PCI Bus and issues PCI access.

- Target

  Means the bus Slave of the PCI Bus. The TX4925 operates as the target when an external PCI device on the PCI Bus executes PCI access to the TX4925.

- Host mode

  One PCI Host device exists for one PCI Bus. The PCI Host device uses a PCI configuration space to perform PCI configuration on other PCI devices on the PCI Bus.

  The TX4925 is set to the Host mode if the ADDR[15] signal is High when the RESET* signal is being deasserted.

- Satellite mode

  A PCI device other than the PCI Host device accepts configuration from the PCI Host device. This state is referred to as the Satellite mode.

  The TX4925 is set to the Satellite mode if the ADDR[15] signal is Low when the RESET* signal is being deasserted.

- DWORD, QWORD

  DWORD expresses 32-bit words, and QWORD expresses 64-bit words. According to conventions observed regarding MIPS architecture, this manual uses the following expressions:

  Byte: 8-bit
  Half-word: 16-bit
  Word: 32-bit
  Double-word: 64-bit

10.3.2 On-Chip Register

The PCI Controller on-chip register contains the PCI Configuration Space Register and the PCI Controller Control Register. The registers that can be accessed vary according to whether the current mode is the Host mode or the Satellite mode.

An external PCI Host device only accesses the PCI Configuration Space Register when in the Satellite mode. This register is defined in the PCI Bus Specifications. A PCI configuration cycle is used to access this register. This register cannot be accessed when in the Host mode. Section 10.5 “PCI Configuration Space Register” explains each register in detail.

The PCI Controller Control Register is only accessed by the TX49 core and cannot be accessed from the PCI Bus.

Registers in the PCI Controller Control Register that include an offset address in the range from 0xD000 to 0xD07F can only be accessed when in the Host mode and cannot be accessed when in the Satellite mode. These registers correspond to PCI Configuration Space Registers that an external PCI Host device accesses when in the Satellite mode. Section 10.4 “PCI Controller Control Register” explains each register in detail.

Figure 10.3.1illustrates the register map when in the Host mode. Figure 10.3.2 illustrates the register map when in the Satellite mode.

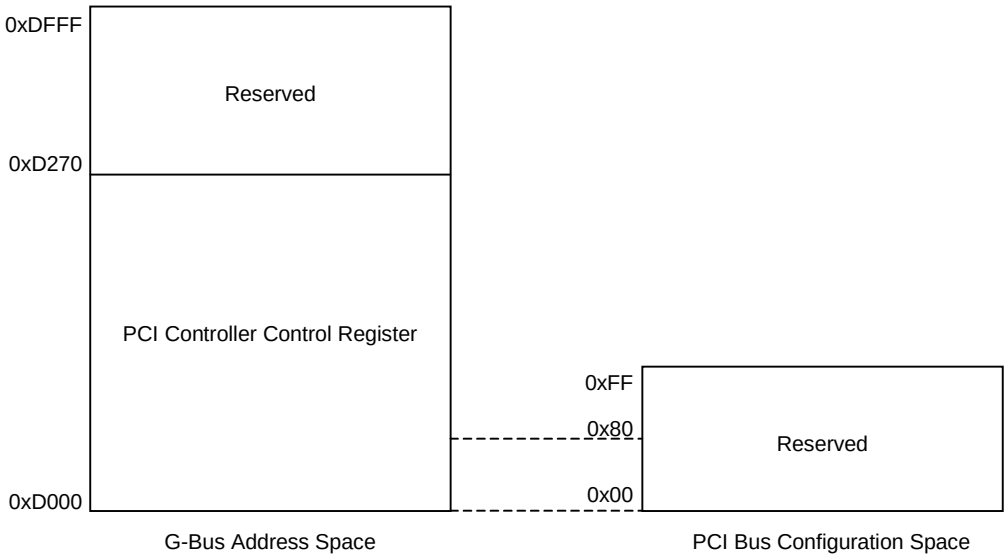


Figure 10.3.1 Register Map in the Host Mode

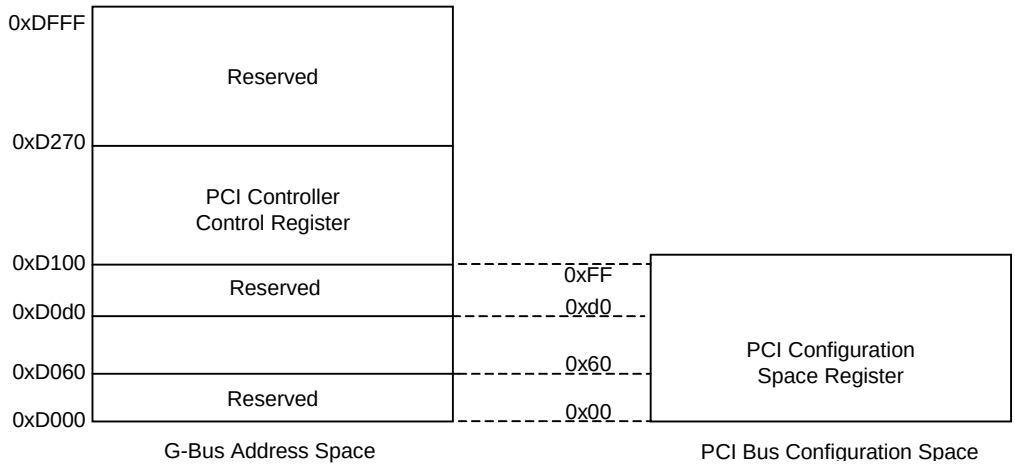


Figure 10.3.2 Register Map in the Satellite Mode

10.3.3 Supported PCI Bus Commands

Table 10.3.1 shows the PCI Bus commands that the PCI Controller supports.

Table 10.3.1 Supported PCI Bus Commands

C/BE Value	PCI Command	As Initiator	As Target
0000	Interrupt Acknowledge	†	—
0001	Special Cycle	†	—
0010	I/O Read	√	√
0011	I/O Write	√	√
0100	(Reserved)	—	—
0101	(Reserved)	—	—
0110	Memory Read	√	√
0111	Memory Write	√	√
1000	(Reserved)	—	—
1001	(Reserved)	—	—
1010	Configuration Read	√	‡
1011	Configuration Write	√	‡
1100	Memory Read Multiple	√	√
1101	Dual Address Cycle	—	—
1110	Memory Read Line	√	√
1111	Memory Write and Invalidate	√	√

√ : Supported when in both the Host mode and the Satellite mode
† : Supported only when in the Host mode
‡ : Supported only when in the Satellite mode
— : Not supported

- I/O Read, I/O Write, Memory Read, Memory Write

  This command executes Read/Write access to the address mapped on the G-Bus and PCI Bus.

- Memory Read Multiple, Memory Read Line

  The Memory Read Multiple command is issued if all of the following conditions are met when the Initiator function is operating and Burst Read access is issued from the G-Bus to the PCI Bus.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.
  (2) The Read data word count is larger than the value set in the Cache Line Size Field.

  Also, the Read Memory Line command is issued when all of the following conditions are met.

  (1) A value other than “0” is set to the Cache Line Size Field (PCICFG1.CLS) of the PCI Configuration 1 Register.
  (2) The Read data word count is larger than the value set in the Cache Line Size Field.

  The Memory Read command is issued if these conditions are not met, namely, if “0” is set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register. In the case of the target, a normal G-Bus cycle is issued to the address mapped from the PCI Bus to the G-Bus.

- Memory Write and Invalidate

  When the TX4925 operates as the initiator, the PCI Controller issue the Memory Write and Invalidate command if all of the following conditions are met when write access from the G-Bus to the PCI Bus occurs.

  (1) The Memory Write and Invalidate Enable bit (PCISTATUS.MWIEN) of the PCI Status Command Register is set.

  (2) A value other than "0" was set to the Cache Line Size field (PCICFG1.CLS) of the PCI Configuration 1 Register.

  (3) The word count of the Write data is larger than the value set in the Cache Line Size field.

  The Memory Write command is issued in these conditions are not met.

  When the TX4925 operates as the target, the Memory Write and Invalidate command is converted into G-Bus Write access. Note that the TX4925 does not support the cache memory Snoop function.

- Configuration Read, Configuration Write

  The corresponding configuration cycles are issued on the PCI Bus. This is done by either reading or writing from/to the G2P Configuration Data Register (G2PCFGDATA) after writing the configuration space address to the G2P Configuration Address Register. The TX4925 supports both "Type 0" and "Type 1" configuration transactions.

  On systems that have PCI card slots, the PCI Host device checks each PCI card slot during system initialization to see if PCI device exist, then sets the Configuration Space Register of the devices that do exist. If a PCI Configuration Read operation is performed for devices that do not exist, then by default a Bus Error exception will be generated since there is no PCI Bus response. Clearing the Bus Error Response During Initiator Read bit (G2PCFG.IRBER) of the PCI Controller Configuration Register makes it possible to execute a Read transaction without causing a Bus Error. All bits of the data read at this time will be set to "1".

  Configuration cycles will be accepted as the target only when in the Satellite mode. After reset, Retry response to PCI Configuration access will continue until the software sets the Target Configuration Access Ready Bit (PCICFG.TCAR) of the PCI Controller Configuration Register. Please use the software to set this bit after the software initialization process ends and the software is ready to accept PCI configuration.

- Interrupt Acknowledge

  This command issues interrupt acknowledge cycles as an initiator only when in the Host mode. Interrupt acknowledge cycles are executed on the PCI Bus when the G2P Interrupt Acknowledge Data Register (G2PINTACK) is read. The value returned by this Read becomes the interrupt acknowledge cycle data.
  The TX4925 does not support interrupt acknowledge cycles as the target.

- Special Cycle

  This command issues specially cycles as the initiator only when in the Host mode. This command issues special cycles on the PCI Bus when writing to the G2P Special Cycle Data Register (G2PSPC). The written value is output as the special cycle data.

  The TX4925 does not support special cycles as the target.

10.3.4 Initiator Access (G-Bus → PCI Bus address conversion)

During PCI initiator access, the G-Bus address of the Burst transaction issued by the G-Bus that was converted into the PCI Bus address is used to issue a Burst transaction on the PCI Bus. 32-bit physical address (G-Bus addresses) are used on the G-Bus. Also, 32-bit PCI Bus addresses are used on the PCI Bus.

Three memory access windows and one I/O access window can be set in the G-Bus space (Figure 10.3.3). The size of each window is variable from 256 bytes to 512 Mbytes. When Burst transactions are issued to these access windows on the G-Bus, then that G-Bus address is converted into a PCI Bus address that is used to issue a Burst transaction to the PCI Bus as the initiator. PCI memory access is issued when the access window is the memory access window. PCI I/O access is issued when the access window is the I/O access window.

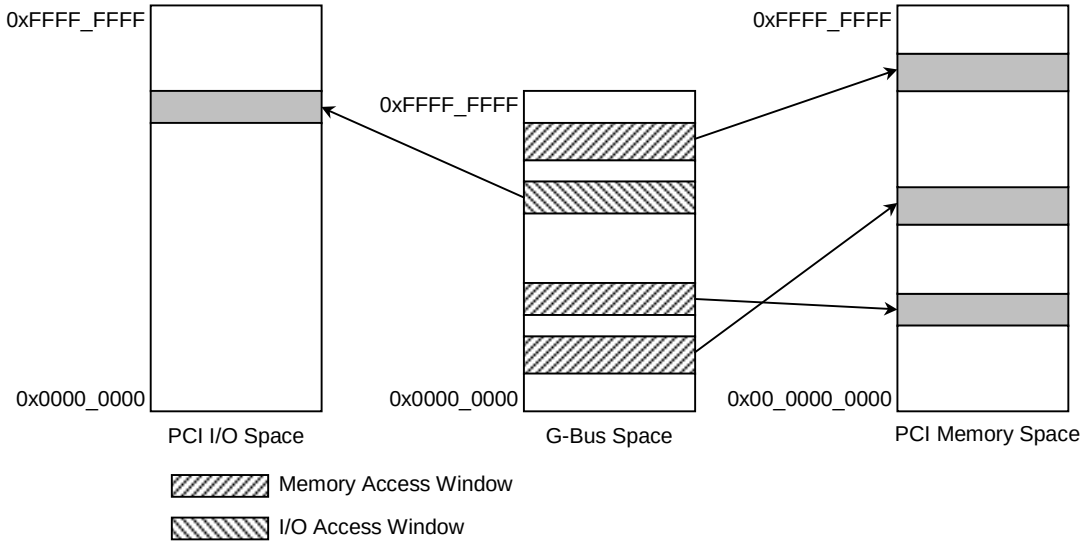


Figure 10.3.3 Initiator Access Memory Window

When expressed as a formula, conversion of a G-Bus address (GBusAddr[31:0]) into a PCI Bus Address (PCIAddr[31:0]) is as follows below. GBASE[31:8], PBASE[31:8], and AM[28:8] each represent the setting register of the corresponding access window indicated below in Table 10.3.2. The "&" symbol indicates a logical AND for each bit, "||" indicates a logical OR for each bit, "!" indicates logical NOT, and "|" indicates bit linking.

If ((GBusAddr[31:29] | (GBusAddr[28:8] & ! AM[28:8] == GBASE[31:29] | (GBASE[28:8] & ! AM[28:8]) then
PCIAddr[31:0] = PBASE[31:29] | ((PBASE[28:8] & ! AM[28:8]) || (GBusAddr[28:8] & AM[28:8]))
| GBusAddr[7:0];

Table 10.3.2 Initiator Access Space Address Mapping Register

	G-Bus Base Address GBASE[31:8]	PCI Bus Base Address PBASE[31:8]	Address Mask AM[28:8]
Memory Space 0	G2PM0GBASE.BA[31:8]	G2PM0PBASE.BA[31:8]	G2PM0MASK.AM[28:8]
Memory Space 1	G2PM1GBASE.BA[31:8]	G2PM1PBASE.BA[31:8]	G2PM1MASK.AM[28:8]
Memory Space 2	G2PM2GBASE.BA[31:8]	G2PM2PBASE.BA[31:8]	G2PM2MASK.AM[28:8]
I/O Space	G2PIOGBASE.BA[31:8]	G2PIOPBASE.BA[31:8]	G2PIOMASK.AM[28:8]

Figure 10.3.4 illustrates this address conversion.

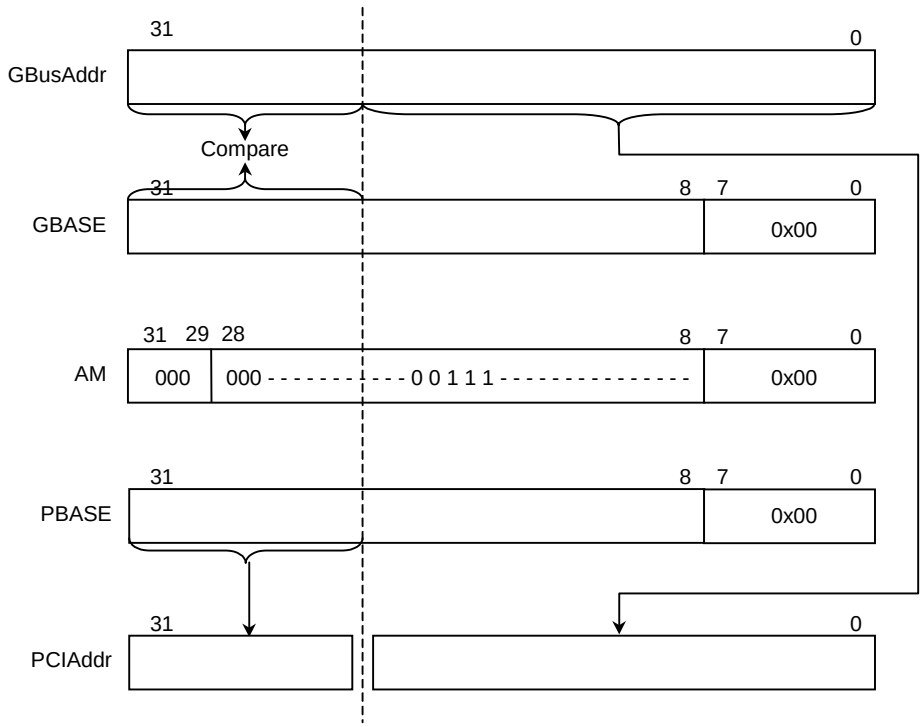


Figure 10.3.4 Address Conversion For Initiator (G-Bus → PCI Bus Address Conversion)

It is possible to set each space to valid/invalid or to perform Word Swap (see "10.3.7 Endian Switching Function"). Table 10.3.3 shows the settings registers for these properties.

Also, operation is not guaranteed if resources in the PCI space were made cacheable and were then accessed when the Critical Word First function of the TX49/H2 core was enabled.

Table 10.3.3 Initiator Access Space Properties Register

	Enable	Word Swap
Memory Space 0	*BusMasterEnable* & G2PCFG.G2PM0EN	G2PCFG.BSWAPM0
Memory Space 1	*BusMasterEnable* & G2PCFG.G2PM1EN	G2PCFG.BSWAPM1
Memory Space 2	*BusMasterEnable* & G2PCFG.G2PM2EN	G2PCFG.BSWAPM2
I/O Space	*BusMasterEnable* & G2PCFG.G2PIOEN	G2PCFG.BSWAPIO

BusMasterEnable:
Host mode: PCI State Command Register Bus Master Bit (PCISTATUS.BM)
Satellite mode: Command Register Bus Master bit

10.3.5 Target Access (PCI Bus → G-Bus Address Conversion)

During PCI target access, the PCI Bus address of the Bus transaction issued by the PCI Bus is converted into a G-Bus address and is used to issue a Bus transaction on the G-Bus. 32-bit PCI Bus addresses are used on the PCI Bus. Also, 32-bit physical addresses are used on the G-Bus.

Three memory access windows and one I/O access window can be set in the PCI bus space (Figure 10.3.5). The size of each memory window is variable from 1 MByte to 512 MBytes. The size of the I/O windo is variable from 256 Bytes to 64 Kbytes. When Bus transactions to these access windows is issued on the PCI Bus, these Bus transactions are accepted as PCI target devices. The PCI Bus Address is converted into G-Bus addresses, then Bus transactions are issued to the G-Bus.

The memory space window responds to the PCI memory space access command. The I/O space window responds to the PCI I/O space access command.

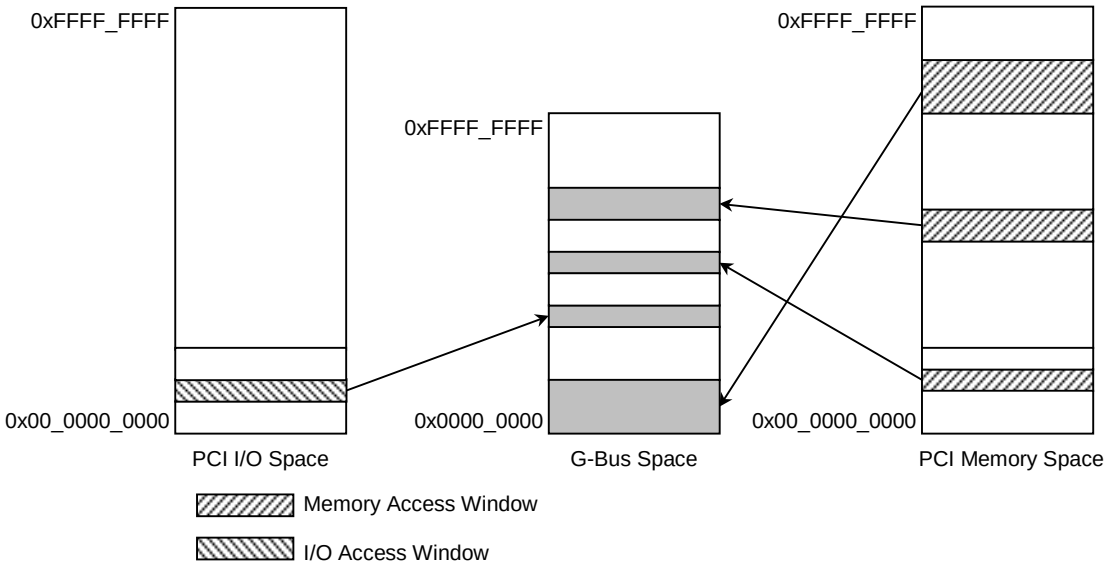


Figure 10.3.5 Target Access Memory Window

When expressed as a formula, conversion of a PCI Bus Address (PCIAddr[31:0]) into a G-Bus address (GBusAddr[31:0]) is as follows below. GBASE[31:8], PBASE[31:8], and AM[28:20]/AM[15:8] each represent the setting register of the corresponding access window indicated below in Table 10.3.4. The "&" symbol indicates a logical AND for each bit, and "|" indicates bit linking.

Memory space 0

```
If (PCIAddr[31:29] | ( PCIAddr[28:20] & !AM[28:20] ) ==  PBASE[31:29] |
    ( PBASE[28:20] & !AM[28:20])) then
    GBusAddr[31:0] = GBASE[31:29] | ((GBASE[28:20] & !AM[28:20]) ||
    (PCIAddr[28:20] & AM[28:20])) | PCIAddr[19:0];
```

Memory space 1

```
If (PCIAddr[31:29] | ( PCIAddr[28:20] & !AM[28:20] ) ==  PBASE[31:29] |
    ( PBASE[28:20] & !AM[28:20])) then
    GBusAddr[31:0] = GBASE[31:29] | ((GBASE[28:20] & !AM[28:20]) ||
    (PCIAddr[28:20] & AM[28:20])) | PCIAddr[19:0];
```

Memory space 2

```
If (PCIAddr[31:29] | ( PCIAddr[28:20] & !AM[28:20] ) ==  PBASE[31:29] |
    ( PBASE[28:20] & !AM[28:20])) then
    GBusAddr[31:0] = GBASE[31:29] | ((GBASE[28:20] & !AM[28:20]) ||
    (PCIAddr[28:20] & AM[28:20])) | PCIAddr[19:0];
```

I/O space

```
If (PCIAddr[31:8]  == P2GIOPBASE.BA[31:8]) then
    GBusAddr[31:0] = P2GIOGBASE[31:8] | PCIAddr[7:0];
```

Table 10.3.4 Target Access Space Address Mapping Register

	PCI Bus Base Address PBASE	G-Bus Base Address GBASE	Address Mask AM
Memory Space 0	P2GM0PLBASE.BA[31:20]	P2GM0GBASE.BA[31:20]	P2GM0CTR.AM[28:20]
Memory Space 1	P2GM1PLBASE.BA[31:20]	P2GM1GBASE.BA[31:20]	P2GM1CTR.AM[28:20]
Memory Space 2	P2GM2PBASE.BA[31:20]	P2GM2GBASE.BA[31:20]	P2GM2CTR.AM[28:20]
I/O Space	P2GIOPBASE.BA[31:8]	P2GIOGBASE.BA[31:8]	P2GIOCTR.AM[15:8]

Figure 10.3.6 and Figure 10.3.7 illustrate this address conversion.

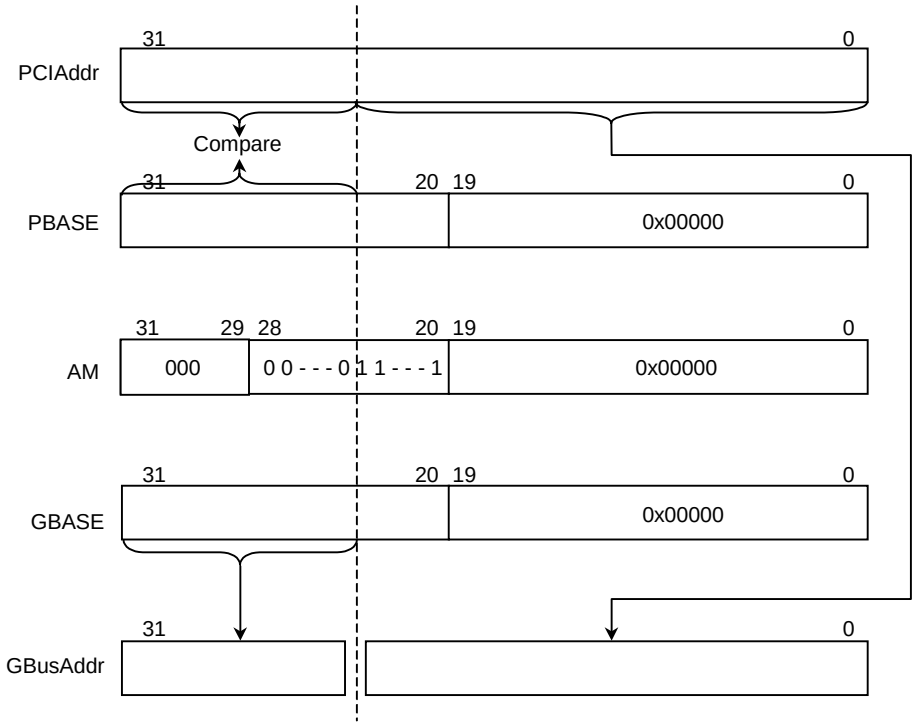


Figure 10.3.6 Memory Address Conversion for Target (PCI Bus → G-Bus Address Conversion)

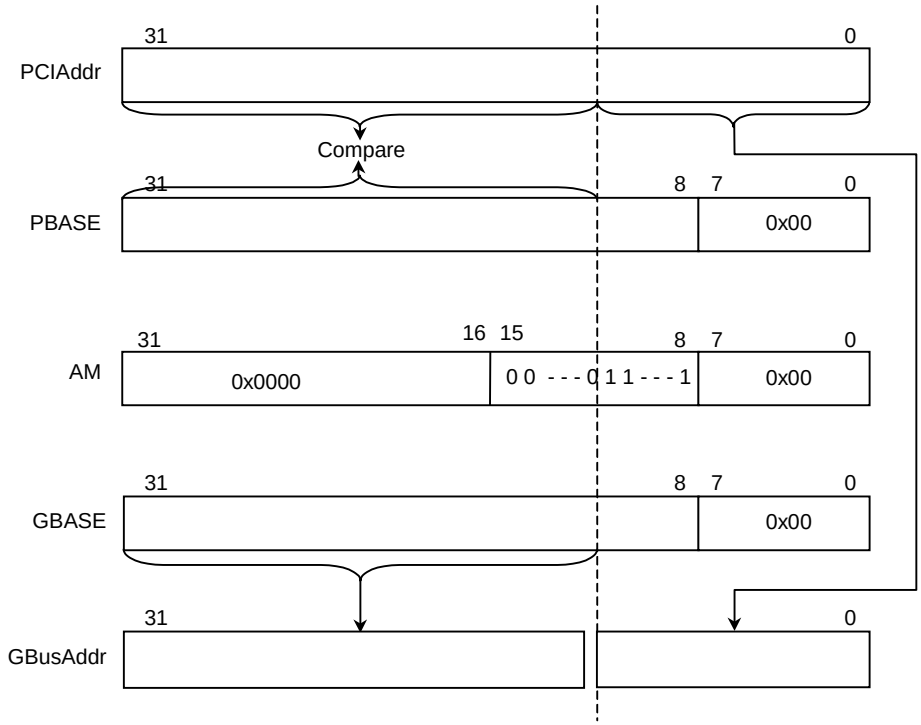


Figure 10.3.7 I/O Address Conversion for Target (PCI Bus → G-Bus Address Conversion)

It is possible to set each space to valid/invalid, pre-fetch Read to valid/invalid, or to perform Word Swap (see 10.3.7). Table 10.3.5 shows the settings registers for these properties.

When pre-fetch Reads are set to valid, data transfer is performed on the G-Bus according to the size set by the Target Pre-fetch Read Burst Length Field (P2GMnCTR.TPRBL) of the P2G Memory Space n Control Register during a PCI target Read transaction. This is performed using accesses to resources that will not be affected even if a pre-read such as memory is performed. Also, PCI Burst Reads to memory spaces that were set to I/O space and pre-fetch disable are not supported.

Table 10.3.5 Target Access Space Properties Register

	Enable	Pre-fetch (Initial State)	Word Swap
Memory Space 0	PCICCFG.TCAR & *MemEnable* & P2GM0CTR.P2GM0EN	P2GM0CTR.MEM0PD (valid)	P2GM0CTR.BSWAP
Memory Space 1	PCICCFG.TCAR & *MemEnable* & P2GM1CTR.P2GM1EN	P2GM1CTR.MEM1PD (valid)	P2GM1CTR.BSWAP
Memory Space 2	PCICCFG.TCAR & *MemEnable* & P2GM2CTR.P2GM2EN	P2GM2CTR.MEM2PD (invalid)	P2GM2CTR.BSWAP
I/O Space	PCICCFG.TCAR & *IOEnable* & P2GIOCTR.P2GIOEN	Always invalid	P2GIOCTR.BSWAP

MemEnable:
Host mode: PCI State Command Register Memory Space bit (PCISTATUS.MEMSP)
Satellite mode: Command Register Memory Space bit

IOEnable:
Host mode: PCI State Command Register I/O Space bit (PCISTATUS.IOSP)
Satellite mode: Command Register I/O Space bit

10.3.6 Post Write Function

The Post Write function improves system performance by completing the original bus Write transaction without waiting for the other bus to complete its transaction when the first bus issues a Write transaction. Initiator Write can Post Write a maximum of four Write transactions, and Target Write can Post Write a maximum of nine Write transactions.

Due to compatibility issues with old PC software in the PCI specifications, performing Post Writes with Initiator Configuration Write and Target I/O Write is not recognized. However, the TX4925 PCI Controller can even perform Post Writes to these functions. In order to guarantee that these Writes are completed by the target device, please execute Reads to the device that performed the Write, then either refer to the read value (so the TX49/H2 core can support non-blocking load) or execute the SYNC instruction.

10.3.7 Endian Switching Function

The TX4925 supports both the Little Endian mode and the Bit Endian mode. On the other hand, the PCI Bus is only defined in Little Endian logic. Therefore, when the TX4925 is in the Big Endian mode, either the software or the hardware must perform some kind of conversion when exchanging data larger than 2 B in size with the PCI Bus.

The PCI Controller can specify the endian switching function that reverses the byte arrangement of the DWORD (32-bit) data for each access window.

Initial state operation matches the correspondence between the address and byte data regardless of the endian mode (operation is address consistent). For example, if WORD (16-bit) data is written to address 0 of the PCI Bus when the TX4925 is in the Big Endian mode, the upper byte (address 0 in Big Endian) is written to PCI Bus address 0 and the lower byte (address 1 in Big Endian) is written to address 1 of the PCI Bus. For Little Endian PCI devices, this means that the byte order is reversed.

When in the Big Endian mode and a particular access window Endian switching mechanism is validated, data is transferred so the byte order does not change in DWORD (32-bit) access to that access window.

Endian switching during initiator access is specified by the Byte Swap bit (BSWAPMn, BSWAPIO) of the G2P Configuration Register (G2PCFG) of the access window for each initiator access (see Table 10.3.3).

Ending switching during target access is specified by the Byte Swap bit (BSWAP) of the G2P Memory Space n Control Register (P2GMnCTR) or G2P I/O Space Control Register (P2GIOCTR) of the access window for each target access.

10.3.8 Power Management

The TX4925 PCI Controller supports power management functions that are compliant to PCI Bus Power Management Interface Specifications Version 1.1 (Partially unsupported).

The PCI Host device controls the system status by reporting the power management state to the PCI Satellite device.

10.3.8.1 Power Management State

In the case of the PCI Bus Power Management Interface Specifications, four power management states are defined from State D0 to State D3. The TX4925 supports states D0 through D3. Figure 10.3.8 illustrates the power management state transition.

After Power On Reset, or when transitioning from the $D3_{HOT}$ state to the D0 state, the power management state becomes uninitialized D0. If initialized by the system software at this point, the state transitions to D0 Active.

If an external PCI Host device writes 11b ($D3_{HOT}$) to the PowerState field of the Power Management Control Status Register (PMCSR) of the Configuration space when in the Satellite mode, then the Power Management State Change bit (P2GSTATUS.PMSC) of the P2G Status Register is set and transitions to the $D3_{HOT}$ state. It then becomes possible to report Power State Change interrupts. The PowerState field value can be read from the PowerState field (PCISSTATUS.PS) of the Satellite Mode PCI Status Register.

The TX4925 uses software to change the system status after a status change is detected.

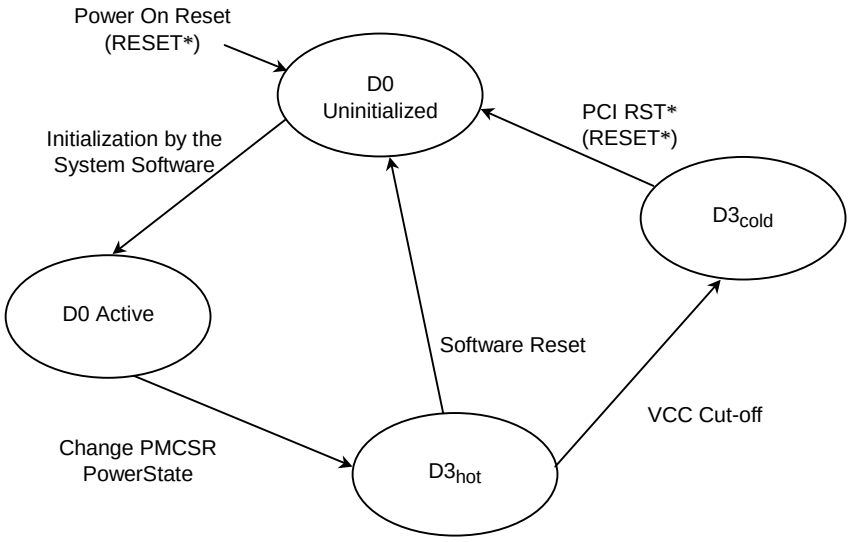


Figure 10.3.8 Transition of the Power Management States

10.3.9 PDMAC (PCI DMA Controller)

The PCI DMA Controller (PDMAC) is a one-channel PCI Director Memory Access (DMA) controller. Data can be transferred bidirectionally between the G-Bus and the PCI Bus.

10.3.9.1 DMA Transfer

The following DMA transfer procedure does not use the Chain DMA mode.

(1) Address Register and Count Register Setting
Sets values for the three following registers.

- PDMAC G-Bus Address Register (PDMGA)
- PDMAC PCI Bus Address Register (PDMPA)
- PDMAC Count Register (PDMCTR)

(2) Chain Address Register Setting
Sets “0” to the PDMAC Chain Address Register (PDMCA).

(3) PDMAC Status Register (PDMSTATUS) Clearing
Clears any remaining status from a previous DMA transfer.

(4) PDMAC Control register (PDMCFG) Setting
Clears the Channel Reset bit (CHRST), and makes settings such as the data transfer direction (XFRDIRC), and the burst mode (BRSTMD).

(5) DMA Transfer Initiation
Setting the Transfer Active bit (XFRACT) of the PDMAC Control Register initiates DMA transfer.

(6) Termination Report

When the DMA data transfer terminates normally, the Normal Data Transfer Complete bit (NTCMP) of the PDMAC Status Register (PDMSTATUS) is set. An interrupt is then reported if the Normal Data Transfer Complete Interrupt Enable bit (NDCMPIE) of the PDMAC Control Register is set.

If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Control register is set.

10.3.9.2 Chain DMA

DMA Command Descriptors are 4 DWORD (16-Byte) data structures indicated in Table 10.3.6 that are placed in memory.

Storing the starting memory address of another DMA Command Descriptor in the Offset 0 Chain Address Field makes it possible to configure a chain list for the DMA command Descriptor. Set “0” in the Chain Address field of the DMA Command Descriptor at the end of the chain list.

When the DMA transfer specified by one DMA Command Descriptor ends, the PDMAC reads the next DMA Command Descriptor that the Chain Address field automatically points to, then continues the DMA transfer. Such continuous DMA transfer that uses multiple descriptors in a chain format is referred to as the Chain DMA mode.

When a DMA Command Descriptor is placed to an address that does not extend across a 32 DWORD boundary in memory, this transfer method is more efficient since data can be read by a single G-Bus Burst Read transaction.

Table 10.3.6 DMA Command Descriptors

Offset Address	Field Name	Transfer Destination Register
0x00	Chain Address	PDMAC Chain Address Register (PDMCA)
0x04	G-Bus Address	PDMAC G-Bus Address Register (PDMGA)
0x08	PCI Bus Address	PDMAC PCI Bus Address Register (PDMPA)
0x0c	Count	PDMAC Count Register (PDMCTR)

The DMA transfer procedure is as follows when in the Chain DMA mode.

(1) Count Register Setting

Sets “0” to the PDMAC Count Register (PMDCTR).

(2) DMA Command Descriptor Chain Construction

Constructs the DMA Command Descriptor Chain in memory.

(3) PDMAC Status Register (PDMSTATUS) Clearing

Clears any remaining status from a previous DMA transfer.

(4) PDMAC Control Register (PDMCFG) Setting

Clears the Channel Reset bit (CHRST) and makes settings such as the data transfer direction (XFRDIRC) and the burst mode (BRSTMD).

(5) DMA Transfer Initiation

Setting the address of the DMA Command descriptor that is at the beginning of the Chain List in the PDMAC Chain Address Register (PDMCA) automatically initiates DMA transfer.

First, the values stored in each field of the DMA Command Descriptor that is at the beginning of the Chain List are read to each corresponding PDMAC Register, then DMA transfer is performed according to the read values.

If a value other than "0" is stored in the PDMAC Chain Address Register (PDMCA), data transfer of the size stored in the PDMAC Count Register is complete, then the DMA Command Descriptor value for the memory address specified by the PDMAC Chain Address Register is read.

When the Chain Address field value reads a descriptor of "0", the PDMAC Chain Address Register value is not updated and the previous value (address of the Data Command Descriptor at which the Chain Address field value is "0" when read) is held.

(6) Termination Report

When DMA data transfer of all descriptor chains terminates normally, the Normal Chain Complete bit (NCCMP) of the PDMAC Status Register is set. An interrupt is reported if the Chain Termination Interrupt Enable bit (MCCMPIE) of the PDMAC Control register (PDMCFG) is set.

Also, the Normal Data Transfer Complete bit (NTCMP) of the DPMAC Status Register is set each time the DMA data transfer specified by a DMA Command Descriptor terminates normally. An interrupt is reported if the Normal Data Transfer Complete Interrupt Enable bit (NTCMPIE) of the PDMAC Control Register (PDMCFG) is set.

If an error is detected during DMA transfer, the error cause is recorded in the lower 5 bits of the PDMAC Status Register and the transfer is aborted. An interrupt is then reported if the Error Detection Interrupt Enable bit (ERRIE) of the PDMAC Control register is set.

10.3.9.3 Dynamic Chain Operation

It is possible to dynamically add other DMA Command Descriptor Chains to a DMA Command Descriptor Chain that is currently being processed when executing DMA data transfer. This is done according to the following procedure.

(1) DMA Command Descriptor Chain Construction

Constructs a DMA Command Descriptor Chain in memory.

(2) Addition of DMA Command Descriptor Chains

Substitutes the address of the command descriptor that is at the beginning of the descriptor chain to be added into the Descriptor Chain Address field at the end of the DMA Command Descriptor Chain that is currently performing DMA transfer.

(3) Chain Enable bit checking

Reads the value of the Chain Enable bit (CHNEN) in the PDMAC Control Register (PDMCFG). If the read value is "0", then the Chain Address field value of the DMA Command Descriptor indicated by the address stored in the PDMAC Chain Address Register (PDMCA) is written to the PDMAC Chain Address Register (PDMCA).

10.3.9.4 Data Transfer Mode

The Transfer Mode field in the PDMAC Configuration register (PDMCFG.XFRMODE) selects a data transfer mode for a DMA transaction over the G-Bus. Transfer data size and when a transfer is started differ mode by mode.

Table 10.3.7 shows the available data transfer modes. Mode 00 performs a single-beat transfer; Mode 01 performs a burst transfer. In either mode, the PDMAC reads data from the source address, and after the read cycles are complete, writes the data to the destination address. Source read and destination write cycles do not overlap.

Table 10.3.7 Data Transfer Modes

G-Bus to the PCI bus

PDMCFG.XFRMODE	Free FIFO Space Required for G-Bus Read Accesses (DWORDs)	Number of DWORDs Read from the G-Bus	Number of DWORDs required in FIFO for PCI Bus Write Accesses	Number of DWORDs Written to the PCI Bus	Overlaps of PCI Bus and G-Bus Cycles
00	1	1	1	1	None
01	16	16 (Burst) *1	16	16 (Burst)	None

PCI Bus to the G-Bus

PDMCFG.XFRMODE	Free FIFO Space Required for G-Bus Write Accesses (DWORDs)	Number of DWORDs Written to the G-Bus	Number of DWORDs Required in FIFO for PCI Bus Read Accesses	Number of DWORDs Read from the PCI Bus	Overlaps of PCI Bus and G-Bus Cycles
00	1	1	1	1	None
01	16	16 (Burst) *1	16	16 (Burst)	None

*1: The last DMA transfer consists of less than 16 DWORDs if the data to be transferred is not a multiple of 16 DWORDs.

Note: The amount of data transferred varies, depending on the number of DWORDs present in the FIFO.

10.3.10 Error Detection, Interrupt Reporting

The PCI Controller reports the four following types of interrupts to the Interrupt Controller (IRC).

- Normal Operation Interrupt (Interrupt Number: 20, PCIC)
- PDMAC Interrupt (Interrupt Number: 19, PDMAC)
- Power Management Interrupt (Interrupt Number: 30, PCIPME)
- Error Detection Interrupt (Interrupt Number: 29, PCIERR)

When each cause is detected, an interrupt is reported if the corresponding Status bit is set, and the corresponding Interrupt Enable Bit is set. The following tables list the name of each interrupt cause, the Status bit, and the Interrupt Enable bit. Please refer to the explanation of each Status bit for more information regarding each interrupt cause.

10.3.10.1 Normal Operation Interrupt

Name	Status Bit		Interrupt Enable Bit	
Master Abort Reception	PCISTATUS / PCISSTATUS	RMA	PCIMASK	RMAIE
Target Abort Reception		RTA		RTAIE
Target Abort Signal		STA		STAIE

10.3.10.2 PDMAC Interrupts

Name	Status Bit		Interrupt Enable Bit	
Normal Chain Completion	PDMSTATUS	NCCMP	PDMCFG	NCCMPIE
Normal Data Transfer Completion		NTCMP		NTCMPIE
PCI Parity Error		PCIPERR		ERRIE
PCI System Error		PCISERR		
PCI Fatal Error		PCIERR		
G-Bus Chain Error		CHNERR		
G-Bus Data Error		DATAERR		

10.3.10.3 Power Management Interrupts

Name	Status Bit		Interrupt Enable Bit	
PM Status Change Detection	P2GSTATUS	PMSC	P2GMASK	PMSCIE

10.3.10.4 Error Detection Interrupts

Name	Status Bit		Interrupt Enable Bit	
Bus Error Detection	PCISTATUS / PCISSTATUS	DPE	PCIMASK	DPEIE
System Error Signal		SSE		SSEIE
Master Data Parity Error		MDPE		MDPEIE
Master Direct Fatal Error	G2PSTATUS	MDFE	G2PMASK	MDFEIE
Master Direct Parity Error		MPRE		MPREIE
TRDY Timeout Error		IDTTOE		IDTTOEIE
Retry Timeout Error		IDRTOE		IDRTOEIE
Broken Master Detection	PBASTATUS	BMD	PBAMASK	BMDIE
Target PERR* Detection	P2GSTATUUS	PERR	P2GMASK	PERRIE
Target G-bus Error Detection		GBE		GBEIE
SERR* Detection	PCICSTATUS	SERR	PCICMASK	SERRIE

10.3.11 PCI Bus Arbiter

Configuration settings (ADDR[1] signal = “1”) during boot up selects whether to use the on-chip PCI Bus arbiter (Internal PCI Bus Arbiter mode) or to use the External PCI Bus arbiter (External PCI Bus Arbiter mode).

When in the Internal PCI Bus Abiter mode, setting the PCI Bus Arbiter Enable bit (PBACFG.PBAEN) of the PCI Bus Arbiter Configuration Register starts operation.

The on-chip PCI Bus arbiter can arbitrate eight sets of PCI Bus usage requests from the Bus Master. Five ports are used: one for the PCI Controller bus master and four for External Bus masters. The three remaining ports are reserved for future expanded features.

10.3.11.1 Request Signal, Grant Signal

The four external Bus Masters are connected to the REQ[3:0] signal and the GNT[3:0]* signal.

Also, when in the External PCI Bus Arbiter mode (Satellite Mode), the REQ[0]* signal becomes the PCI Bus Request Output signal and the GNT[0]* signal becomes the Bus Usage Permission Input Signal. Furthermore, the REQ[1]* signal can be used as an interrupt output signal to the external devices (see 14.3.7 for more information).

10.3.11.2 Priority Control

As illustrated below in Figure 10.3.9, a combination of two round-robin sequences is used as the arbitration algorithm that determines the priority of Internal PCI Bus arbiter bus requests. The round-robin with the lower priority (Level 2) consists of Masters W - Z, and the round-robin with the high priority (Level 1), consists of Master A - D and Level 2 Masters. The PCI Bus Arbiter Request Port Register (PBAREQPORT) specifies whether to allocate the PCI Controller and the four External Bus Masters to Masters A-D or W - Z.

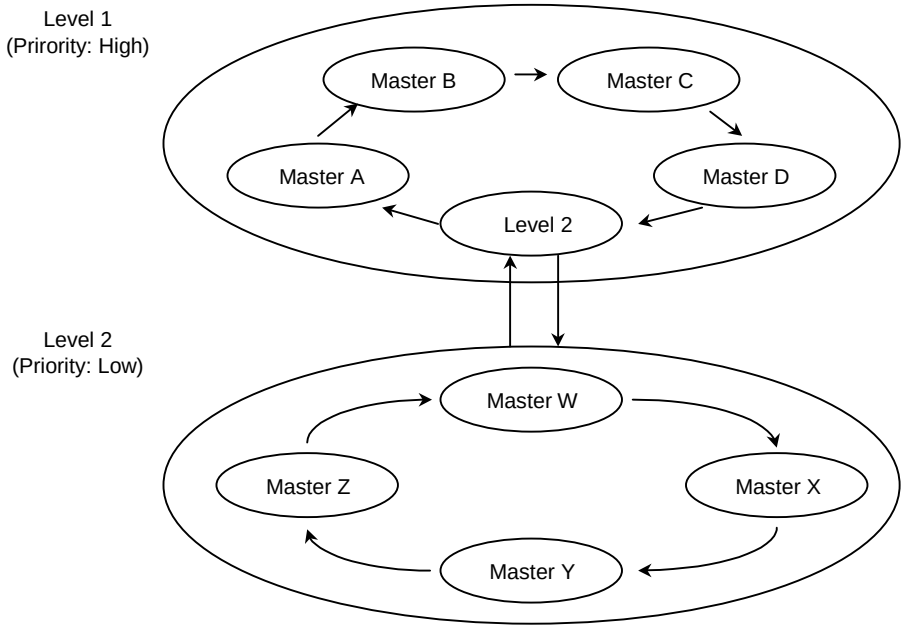


Figure 10.3.9 PCI Bus Arbitration Priority

The Bus Master priority is determined based on the Level 1 round-robin sequence. However, when Level 2 is used inside Level 1, the Level 2 Bus Master priority is determined based on the Level 2 round-robin sequence.

All 8 Bus Masters cannot be used on the TX4925. However, the Bus Master priority would be as follows if we assume there is a hypothetical device that can use all 8 Bus Masters and all 8 Bus Masters (Masters A – D, W – Z) simultaneously requested the bus.

A → B → C → D → W
→ A → B → C → D → X
→ A → B → C → D → Y
→ A → B → C → D → Z
→ A (returns to the beginning)

Since the priority can only transition in the order indicated by the above arrows (or the arrows in Figure 10.3.9, if we assume that the three Bus Masters A, B, and W exist, then Master B will obtain the bus first. If A and W then simultaneously request the bus, then PCI Bus ownership will transition in the order B → W → A.

10.3.11.3 Bus Parking

The On-chip PCI Bus Arbiter supports bus parking.

The last PCI Bus Master is made the Park Master when the Fix Park Master bit (FIXPM) of the PCI Bus Arbiter Configuration Register (PBACFG) is cleared (in the default state). When this bit is set, the Internal PCI Bus Arbiter Request A Port (Master A) becomes the Park Master.

10.3.11.4 Broken Master Detect

The TX4925 On-chip PCI Bus Arbiter has a function for automatically detecting broken masters.

If the PCI Bus Master requests and is granted the bus when the PCI Bus is in the Idle state, this master must assert the FRAME* signal within 16 PCI Clock cycles and start a transaction. The PCI Bus Arbiter recognizes any device that breaks this rule as a broken bus master and removes that device from the bus arbitration sequence.

This detection function is enabled when the Broken Master Check Enable bit (BMCEN) of the PCI Bus Arbiter Configuration Register (PBACFG) is set. When a broken master is detected, the Broken Master Detection bit (PBSTATUS.BMD) of the PCI Bus Arbiter Status Register is set and the bit in the PCI Bus Arbiter Broken Master Register (PBABM) that corresponds to that master is set. Then it also becomes possible to report an interrupt.

10.3.12 PCI Boot

Setting the configuration during boot up (ADDR[8:6] = 0x011) makes it possible to set the reset exception vector address of the TX49/H2 core to PCI Bus address 0xBFC0_0000.

Two windows of the memory space from the G-Bus to the PCI Bus space are used when in the PCI Boot mode. The defaults of several registers are changed as indicated below.

- G-Bus base address (G2GBASE): 0x1FC0_0000
- Space size (G2PM2MASK): 4 MB
- PCI Bus base address (G2PM2PBASE): 0xBFC0_0000
- Initiator Memory Space 2 Enable (PCICCFG.G2PM2EN): 1
- Bus Master bit (PCISTATUS.BM) [Only when in the Host mode] 1
- Target Configuration Access Ready (PCICSTAUTS.TCAR) [Only when in the Satellite mode] 1

Also, the on-chip PCI Bus Arbiter cannot be used when the PCI Boot mode is being used while in the Satellite mode.

10.3.13 Set Configuration Space

In Table 10.5.1, the values for the registers inside the PCI Configuration Space Register that have a gray background can be rewritten using one of the following method.

10.3.13.1 Set the Configuration Space Using Software Reset

By using the following procedure, it is possible to use the software to set the configuration space.

(1) Set the value to be loaded in the Configuration Data 0 Register (PCICDATA0), the Configuration Data 1 Register (PCICDATA1), the Configuration Data 2 Register (PCICDATA2), and the Configuration Data 3 Register (PCICDATA3).
(2) Clear the Load Configuration Data Register bit (LCFG) of the PCI Controller Configuration Register (PCICCFG).

After these processes are complete, please set the Target Configuration Access Ready bit (PCICCFG.TCAR) of the PCI Controller Configuration Register to be able to accept access to the PCI Configuration space.

10.3.14 PCI Clock Signal

The configuration setting via ADDR[18] during boot-up determines whether or not the clock from the on-chip PLL is driven out from the PCI Clock outputs, PCICLK[2:1] and PCICLKIO. When ADDR[18] is High, PCICLK[2:1] and PCICLKIO are configured as outputs. When ADDR[18] is Low, PCICLK[2:1] are forced to the High-Z state and PCICLKIO is configured as an input.

When PCICLK[2:1] and PCICLKIO are configured as outputs, the PCFG.PCICLKIOEN bit must not be cleared. When configured as an output, the PCIC internally feeds back PCICLKIO as a PCI bus clock to adjust the phase of the clock. However, if the PCFG.PCICLKEN[2:1] and PCFG.PCICLKIOEN bits are cleared, PCICLK[2:1] and PCICLKIO are held Low respectively. (see Figure 6.1.1).

10.4 PCI Controller Control Register

Table 10.4.1 lists the registers contained in the PCI Controller Control Register. Parentheses in the register names indicate the corresponding PCI Configuration Space Register.

Table 10.4.1 PCI Controller Control Register (1/2)

Section	Address	Bit Width	Mnemonic	Register Name
10.4.1	0xD000	32	PCIID	ID Register (Device ID, Vendor ID)
10.4.2	0xD004	32	PCISTATUS	PCI Status, Command Register (Status, Command)
10.4.3	0xD008	32	PCICCREV	Class Code, Revision ID Register (Class Code, Revision ID)
10.4.4	0xD00C	32	PCICFG1	PCI Configuration 1 Register (BIST, Header Type, Latency Timer, Cache Line Size)
10.4.5	0xD010	32	P2GM0PBASE	P2G Memory Space 0 PCI Base Address Register (Memory Space 0 Base Address)
10.4.6	0xD014	32	P2GM1PBASE	P2G Memory Space 1 PCI Base Address Register (Memory Space 1 Base Address)
10.4.7	0xD018	32	P2GM2PBASE	P2G Memory Space 2 PCI Base Address Register (Memory Space 2 Base Address)
10.4.8	0xD01C	32	P2GIOPBASE	P2G I/O Space PCI Base Address Register (I/O Space Base Address)
10.4.9	0xD02C	32	PCISID	Subsystem ID Register (Subsystem ID, Subsystem Vendor ID)
10.4.10	0xD034	32	PCICAPPTR	Capabilities Pointer Register (Capabilities Pointer)
10.4.11	0xD03C	32	PCICFG2	PCI Configuration 2 Register (Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line)
10.4.12	0xD040	32	G2PTOCNT	G2P Timeout Count Register (Retry Timeout Value, TRDY Timeout Value)
10.4.13	0xD060	32	G2PCFG	G2P Configuration Register
10.4.14	0xD064	32	G2PSTATUS	G2P Status Register
10.4.15	0xD068	32	G2PMASK	G2P Interrupt Mask Register
—	0xD06C	—	—	(Reserved)
10.4.16	0xD088	32	PCISSTATUS	Satellite Mode PCI Status Register (Status, PMCSR)
10.4.17	0xD08C	32	PCIMASK	PCI Status Interrupt Mask Register
10.4.18	0xD090	32	P2GCFG	P2G Configuration Register
10.4.19	0xD094	32	P2GSTATUS	P2G Status Register
10.4.20	0xD098	32	P2GMASK	P2G Interrupt Mask Register
10.4.21	0xD09C	32	P2GCCMD	P2G Current Command Register
10.5.1	0xD0DC	8	Cap_ID	Capability ID Register
10.5.2	0xD0DD	8	Next_Item_Ptr	Next Item Pointer Register
10.5.3	0xD0DE	16	PMC	Power Management Capability Register
10.5.4	0xD0E0	16	PMCSR	Power Management Control/Status Register
10.4.22	0xD100	32	PBAREQPORT	PCI Bus Arbiter Request Port Register
10.4.23	0xD104	32	PBACFG	PCI Bus Arbiter Configuration Register
10.4.24	0xD108	32	PBASTATUS	PCI Bus Arbiter Status Register
10.4.25	0xD10C	32	PBAMASK	PCI Bus Arbiter Interrupt Mask Register
10.4.26	0xD110	32	PBABM	PCI Bus Arbiter Broken Master Register
10.4.27	0xD114	32	PBACREQ	PCI Bus Arbiter Current Request Register (for diagnostics)
10.4.28	0xD118	32	PBACGNT	PCI Bus Arbiter Current Grant Register (for diagnostics)
10.4.29	0xD11C	32	PBACSTATE	PCI Bus Arbiter Current State Register (for diagnostics)
10.4.30	0xD120	32	G2PM0GBASE	G2P Memory Space 0 G-Bus Base Address Register
10.4.31	0xD128	32	G2PM1GBASE	G2P Memory Space 1 G-Bus Base Address Register
10.4.32	0xD130	32	G2PM2GBASE	G2P Memory Space 2 G-Bus Base Address Register
10.4.33	0xD138	32	G2PIOGBASE	G2P I/O Space G-Bus Base Address Register
10.4.34	0xD140	32	G2PM0MASK	G2P Memory Space 0 Address Mask Register
10.4.35	0xD144	32	G2PM1MASK	G2P Memory Space 1 Address Mask Register
10.4.36	0xD148	32	G2PM2MASK	G2P Memory Space 2 Address Mask Register
10.4.37	0xD14C	32	G2PIOMASK	G2P I/O Space Address Mask Register
10.4.38	0xD150	32	G2PM0PBASE	G2P Memory Space 0 PCI Base Address Register
10.4.39	0xD158	32	G2PM1PBASE	G2P Memory Space 1 PCI Base Address Register

Table 10.4.1 PCI Controller Control Register (2/2)

Section	Address	Bit Width	Mnemonic	Register Name
10.4.40	0xD160	32	G2PM2PBASE	G2P Memory Space 2 PCI Base Address Register
10.4.41	0xD168	32	G2PIOPBASE	G2P I/O Space PCI Base Address Register
10.4.42	0xD170	32	PCICCFG	PCI Controller Configuration Register
10.4.43	0xD174	32	PCICSTATUS	PCI Controller Status Register
10.4.44	0xD178	32	PCICMASK	PCI Controller Interrupt Mask Register
10.4.45	0xD180	32	P2GM0GBASE	P2G Memory Space 0 G-Bus Base Address Register
10.4.46	0xD184	32	P2GM0CTR	P2G Memory Space 0 Control Register
10.4.47	0xD188	32	P2GM1GBASE	P2G Memory Space 1 G-Bus Base Address Register
10.4.48	0xD18C	32	P2GM1CTR	P2G Memory Space 1 Control Register
10.4.49	0xD190	32	P2GM2GBASE	P2G Memory Space 2 G-Bus Base Address Register
10.4.50	0xD194	32	P2GM2CTR	P2G Memory Space 2 Control Register
10.4.51	0xD198	32	P2GIOGBASE	P2G I/O Space G-Bus Base Address Register
10.4.52	0xD19C	32	P2GIOCTR	P2G IO Space Control Register
10.4.53	0xD1A0	32	G2PCFGADRS	G2P Configuration Address Register
10.4.54	0xD1A4	32	G2PCFGDATA	G2P Configuration Data Register
—	0xD1B0	—	—	(Reserved)
—	0xD1B8	—	—	(Reserved)
—	0xD1C0	—	—	(Reserved)
10.4.55	0xD1C8	32	G2PINTACK	G2P Interrupt Acknowledge Data Register
10.4.56	0xD1CC	32	G2PSPC	G2P Special Cycle Data Register
10.4.57	0xD1E0	32	PCICDATA0	Configuration Data 0 Register
10.4.58	0xD1E4	32	PCICDATA1	Configuration Data 1 Register
10.4.59	0xD1E8	32	PCICDATA2	Configuration Data 2 Register
10.4.60	0xD1EC	32	PCICDATA3	Configuration Data 3 Register
10.4.61	0xD200	32	PDMCA	PDMAC Chain Address Register
10.4.62	0xD204	32	PDMGA	PDMAC G-Bus Address Register
10.4.63	0xD208	32	PDMPA	PDMAC PCI Bus Address Register
10.4.64	0xD20C	32	PDMCTR	PDMAC Count Register
10.4.65	0xD210	32	PDMCFG	PDMAC Control Register
10.4.66	0xD214	32	PDMSTATUS	PDMAC Status Register

10.4.1 ID Register (PCIID) 0xD000

The Device ID field corresponds to the Device ID Register in the PCI Configuration Space, and the Vendor ID field corresponds to the Vendor ID register of the PCI Configuration Space. These two fields can be modified by software only when PCICCFG.ConfigBusy=1 after Reset. A write to Register PCICDATA0 when PCICCFG.ConfigBusy=1 will modify the contents of this register. Otherwise this register is Read Only.

This register cannot be accessed when in the Satellite mode.

Bits	Mnemonic	Field Name	Description
31:16	DID	Device ID	Device ID (Initial value: 0x0181, R/L) This register indicates the ID that is allocated to a device. The ID can be changed.
15:0	VID	Vendor ID	Vendor ID (Initial value: 0x102f, R/L) This register indicates the device product that is allocated by PCI SIG. The product allocation can be changed.

Figure 10.4.1 ID Registers

10.4.2 PCI Status, Command Register (PCISTATUS) 0xD004

The upper 16 bits correspond to the Status Register in the PCI Configuration Space, and the lower 16 bits correspond to the Command Register in the PCI Configuration Space.

This register cannot be accessed when in the Satellite mode. However, it is possible to read some values of the upper 16 bits from the Satellite Mode PCI Status Register (PCISSTATUS).

31	30	29	28	27	26	25	24	23	22	21	20	19				16	
DPE	SSE	RMA	RTA	STA	DT		MDPE	FBBCP	Reserved	66MCP	CL	Reserved					
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R		R/W1C	R		R	R						: Type
0	0	0	0	0	01		0	1		0	1						: Initial value

15						10	9	8	7	6	5	4	3	2	1	0	
Reserved							FBBEN	SEREN	STPC	PEREN	VPS	MWIEN	SC	BM	MEMSP	IOSP	
							R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	: Type
							0	0	0	0	0	0	0	0/1	0	0	: Default

Bits	Mnemonic	Field Name	Explanation
31	DPE	Detected Parity Error	Detected Parity Error (Initial value: 0, R/W1C) Indicates that a parity error was detected. A parity error is detected in the three following situations: • Detected a data parity error as the Read command PCI initiator. • Detected a data parity error as the Write command PCI target. • Detected an address parity error. This bit is set regardless of the setting of the Parity Error Response bit (PCISTATUS.PEREN) of the PCI Status, Command Register. 1: Detected a parity error. 0: Did not detect a parity error.
30	SSE	Signaled System Error	Signaled System Error (Initial value: 0, R/W1C) Detects either an address parity error or a special cycle data parity error. This bit is set when the SERR* signal is asserted. 1: Asserted the SERR* signal 0: Did not assert the SERR* signal.
29	RMA	Received Master Abort	Received Master Abort (Initial value: 0, R/W1C) This bit is set when a Master Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator (except for special cycles). 1: Transaction was aborted by a Master Abort. 0: Transaction was not aborted by a Master Abort.
28	RTA	Received Target Abort	Received Target Abort (Initial value: 0, R/W1C) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI initiator. 1: Transaction was aborted by a Target Abort. 0: Transaction was not aborted by a Target Abort.
27	STA	Signaled Target Abort	Signaled Target Abort (Initial value: 0, R/W1C) This bit is set when a Target Abort aborts a PCI Bus Transaction when the PCI Controller operates as the PCI target. 1: Bus transaction was aborted by a Target Abort. 0: Bus transaction was not aborted by a Target Abort.
26:25	DT	DEVSEL Timing	DEVSEL Timing (Fixed value: 01, R) Three DEVSEL assert timings are defined in the PCI 2.2 Specifications: 00b = Fast; 01b = Medium; 10b = Slow; 11b = Reserved). With the exception of Read Configuration and Write Configuration, when the PCI Controller is the PCI target, the DEVSEL signal is asserted to a certain bus command and indicates the slowest speed for responding to the PCI Bus Master.

Figure 10.4.2 PCI Status, Command Register (1/2)

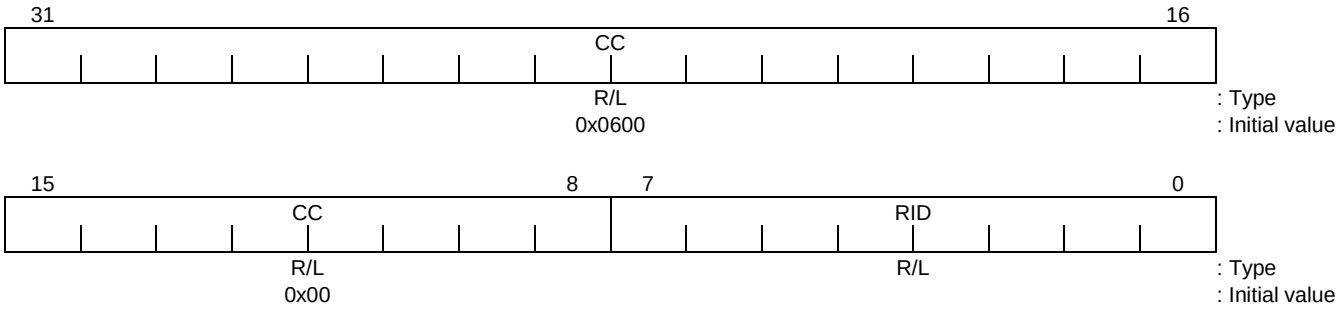
Bits	Mnemonic	Field Name	Explanation
24	MDPE	Master Data Parity Error	Master Data Parity Error (Initial value: 0, R/W1C) Indicates the a parity error occurred when the PCI Controller is the PCI initiator. This bit is not set when the PCI Controller is the target. This bit is set when all of the three following conditions are met. • It has been detected that the PERR* signal was set either directly or indirectly. • The PCI Controller is the Bus Master for a PCI Bus transaction during which an error occurred. • The Parity Error Response bit of the PCI Status Command Register (PCISTATUS.PEREN) has been set.
23	FBBCP	Fast Back-to-Back Capable	Fast Back-to-Back Capable (Fixed value: 1, R) Indicates whether target access of a fast back-to-back transaction can be accepted. Is fixed to “1”.
22	—	Reserved	—
21	66MCP	66 MHz Capable	66 MHz Capable (Fixed value: 0, R) Indicates the 66 MHz operation is impossible. Is fixed to “0”.
20	CL	Capabilities List	Capabilities List (Fixed value: 1, R) Indicates that the capabilities list is being implemented. Is fixed to “1”.
19:10	—	Reserved	—
9	FBBEN	Fast Back-to-Back Enable	Fast Back-to-Back Enable (Initial value: 0, R/W) Indicates that issuing of fast back-to-back transactions has been enabled. 1: Enable 0: Disable
8	SEREN	SERR* Enable	SERR* Enable (Initial value: 0, R/W) Enables/Disables the SERR* signal. The SERR* signal reports that either a PCI Bus address parity error or a special cycle data parity error was detected. The SERR* signal is only asserted when the Parity Error Response bit is set and this bit is set. 1: Enable 0: Disable
7	STPC	Stepping Control	Stepping Control (Fixed value: 0, R) Indicates that stepping control is not being supported.
6	PEREN	Parity Error Response	Parity Error Response (Initial value: 0, R/W) Sets operation when a PCI address/data parity error is detected. A parity error response (either when the Parity Error Response bit (PCISTATUS.PEREN) of the PERR* Signal Assert or PCI Status, Command Register is set, or the SERR* signal is asserted) is performed only when this bit is set. When this bit is cleared, the PCI Controller ignores all parity errors and continues the transaction process as if the parity of that transaction was correct. 1: Parity error response is performed. 0: Parity error response is not performed.
5	VPS	VGA Palette Snoop	VGA Palette Snoop (Fixed value: 0, R) Indicates that the VGA palette snoop function is not supported.
4	MWIEN	Memory Write and Invalidate Enable	Memory Write and Invalidate Enable (Initial value: 0, R/W) Controls whether to use the Memory Write and Invalidate command instead of the Memory Write command when the PCI Controller is the initiator.
3	SC	Special Cycles	Special Cycles (Fixed value: 0, R) Indicates that special cycles will not be accepted as PCI targets.
2	BM	Bus Master	Bus Master (Initial value: 0/1, R/W) The default is only “1” when in the PCI Boot mode and in the Host mode. 1: Operates as the Bus Master. 0: Does not operate as the Bus Master.
1	MEMSP	Memory Space	Memory Space (Initial value: 0, R/W) 1: Respond to PCI memory access. 0: Do not respond to PCI memory access.
0	IOSP	I/O Space	I/O Space (Initial value: 0, R/W) 1: Respond to PCI I/O access. 0: Do not respond to PCI I/O access.

Figure 10.4.2 PCI Status, Command Register (2/2)

10.4.3 Class Code, Revision ID Register (PCICCREV) 0xD008

The Class Code field corresponds to the Class Code Register of the PCI Configuration Space, and the Revision ID field corresponds to the Revision ID Register of the PCI Configuration Space. These two fields can be modified by software only when PCICCFG.ConfigBusy=1 after Reset. A write to Register PCICDATA1 when PCICCFG.ConfigBusy=1 will modify the contents of this register. Otherwise this register is Read Only.

This register cannot be accessed when in the Satellite mode.



Bits	Mnemonic	Field Name	Explanation
31:8	CC	Class Code	Class Code (Initial value: 0x060000, R/L) Classifies the device types. The default is 060000h, which defines the PCI Controller as a Host bridge device. It is possible to change the device type by software.
7:0	RID	Revision ID	Revision ID (Initial value: *****, R/L) Indicates the device revision ID. Please contact our Engineering Department for the exact value. It is possible to change the revision IDsoftware.

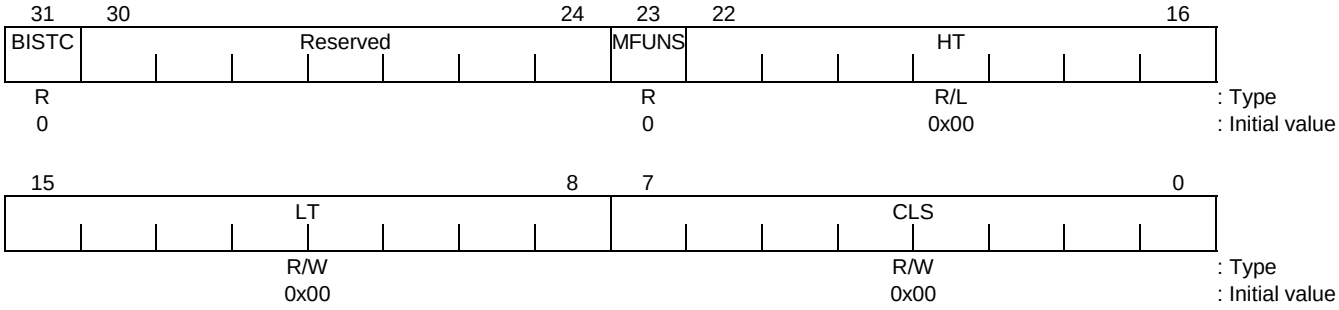
Figure 10.4.3 Class Code, Revision ID Register

10.4.4 PCI Configuration 1 Register (PCICFG1) 0xD00C

The following fields correspond to the following registers.

BIST field → BIST Register of the PCI Configuration Space
Header Type field → Header Type Register in the PCI Configuration Space
Latency Timer field → Latency Timer Register of the PCI Configuration Space
Cache Line Size field → Cache Line Size Register of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



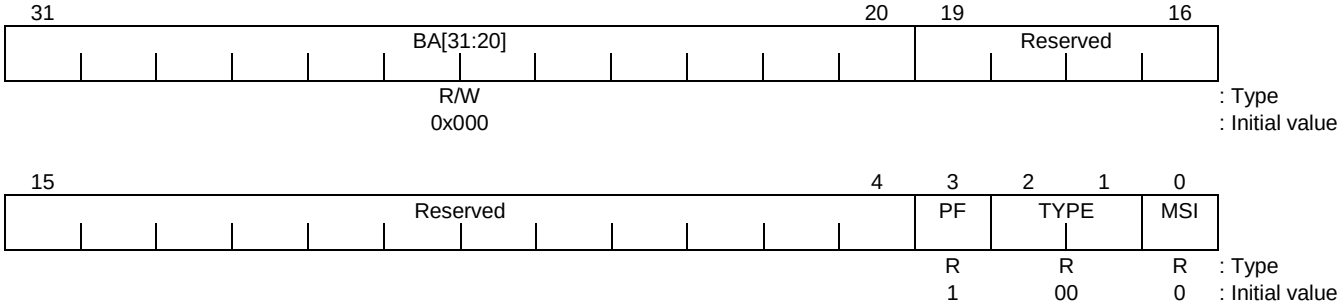
Bits	Mnemonic	Field Name	Description
31	BISTC	BIST Capable	BIST Capable (Fixed value: 0, R) Indicates that the BIST function is not being supported.
30:24	—	Reserved	—
23	MFUNS	Multi-Function	Multi-Function (Fixed value: 0, R) 0: Indicates that the device is a single-function device.
22:16	HT	Header Type	Header Type (Initial value: 0x00, R/L) Indicates the Header type. 0000000: Header Type 0 It is possible to change to the value that was written to the PCICDATA3 Register when PCICCFG.LCFG is “1”.
15:8	LT	Latency Timer	Latency Timer (Initial value: 0x00, R/W) Sets the latency timer value. Specifies the PCI Bus clock count during which to abort access when the GNT* signal is deasserted during PCI access. Since the lower two bits are fixed to “0”, cycle counts can only be specified in multiples of 4.
7:0	CLS	Cache Line Size	Cache Line Size (Initial value: 0x00, R/W) Is used to select the PCI Bus command during a Burst Read transaction. See “10.3.3 Supported PCI Bus Commands)” for more information.

Figure 10.4.4 PCI Configuration 1 Register

10.4.5 P2G Memory Space 0 PCI Base Address Register (P2GM0PBASE) 0xD010

This register corresponds to the Memory Space 0 Base Address Register at offset address 0x10 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



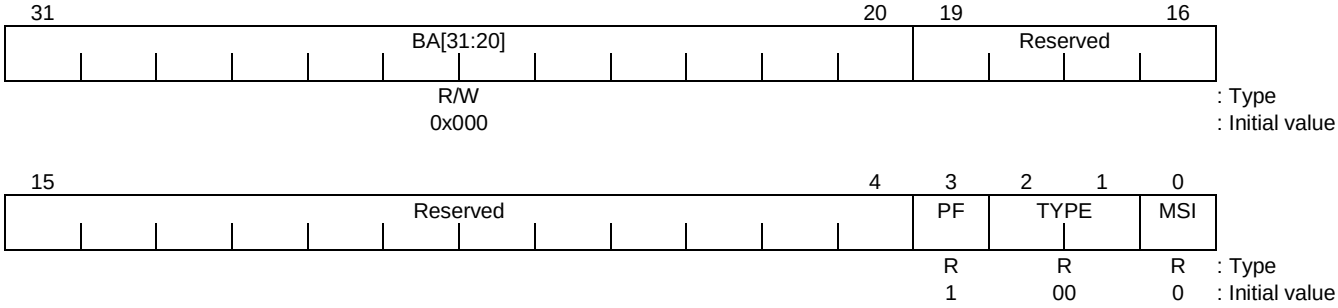
Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Base Address	Base Address (Initial value: 0x000, R/W) Sets the PCI base address in Target Access Memory Space 0. The size of Memory Space 0 is selected from 1MB to 512MB using TC0[28:20].
19:4	—	Reserved	—
3	PF	Prefetchable	Prefetchable (Fixed value: 1, R) 1: Indicates that memory is prefetchable.
2:1	TYPE	Type	Type (Initial value: 00, R) 10: Indicates that an address is within a 64-bit address region.
0	MSI	Memory Space	Memory Space Indicator (Fixed value: 0, R) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.

Figure 10.4.5 P2G Memory Space 0 PCI Base Address Register

10.4.6 P2G Memory Space 1 PCI Base Address Register (P2GM1PBASE) 0xD014

This register corresponds to the Memory Space 1 Base Address Register at offset address 0x14 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



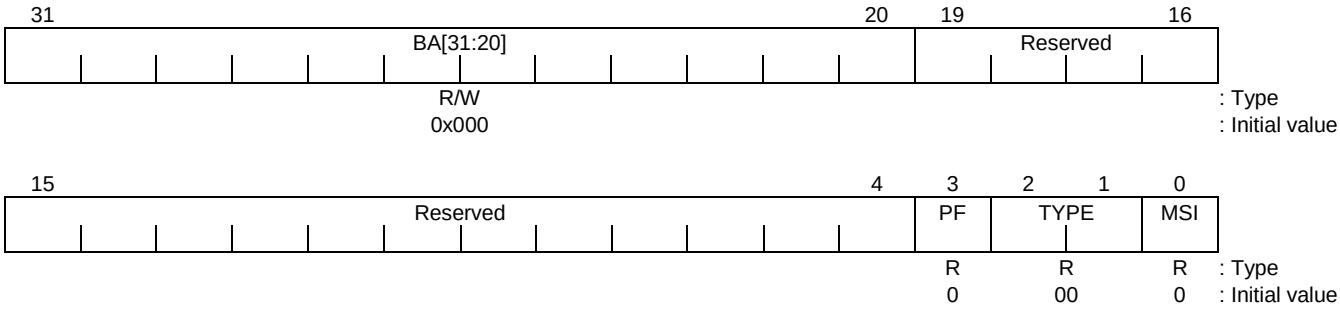
Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Base Address	Base Address (Initial value: 0x000, R/W) Sets the PCI base address in Target Access Memory Space 0. The size of Memory Space 0 is selected from 1MB to 512MB using TC0[28:20].
19:4	—	Reserved	—
3	PF	Prefetchable	Prefetchable (Fixed value: 1, R) 1: Indicates that memory is prefetchable.
2:1	TYPE	Type	Memory Type (Fixed value: 00, R) 00: Indicates that an address is within a 32-bit address region.
0	MSI	Memory Space	Memory Space Indicator (Fixed value: 0, R) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.

Figure 10.4.6 P2G Memory Space 1 PCI Base Address Register

10.4.7 P2G Memory Space 2 PCI Base Address Register (P2GM2PBASE) 0xD018

This register corresponds to the Memory Space 2 Base Address Register at offset address 0x18 of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



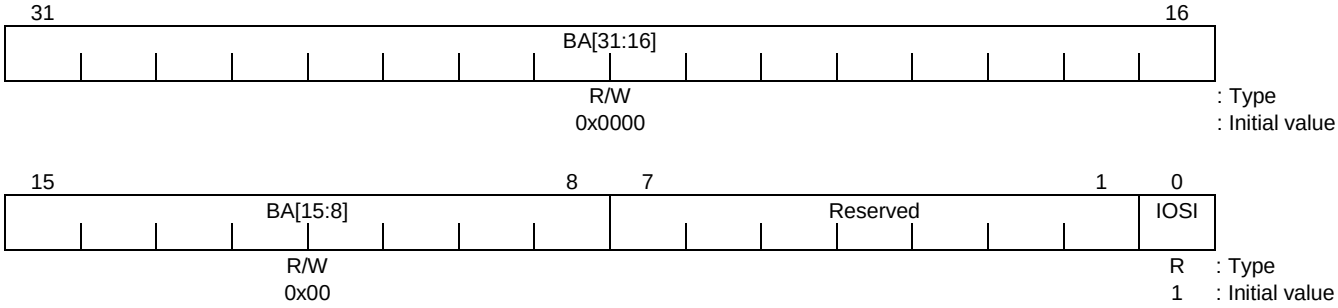
Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Base Address	Base Address (Initial value: 0x000, R/W) Sets the PCI base address in Target Access Memory Space 0. The size of Memory Space 0 is selected from 1MB to 512MB using TC0[28:20].
19:4	—	Reserved	—
3	PF	Prefetchable	Prefetchable (Fixed value: 0, R) 1: Indicates that memory is prefetchable.
2:1	TYPE	Type	Memory Type (Fixed value: 00, R) 00: Indicates that an address is within a 32-bit address region.
0	MSI	Memory Space	Memory Space Indicator (Fixed value: 0, R) 0: Indicates that this Base Address Register is for use by the PCI Memory Space.

Figure 10.4.7 P2G Memory Space 2 PCI Base Address Register

10.4.8 P2G I/O Space PCI Base Address Register (P2GIOPBASE) 0xD01C

This register corresponds to the I/O Space Base Address at offset address 0x1C of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



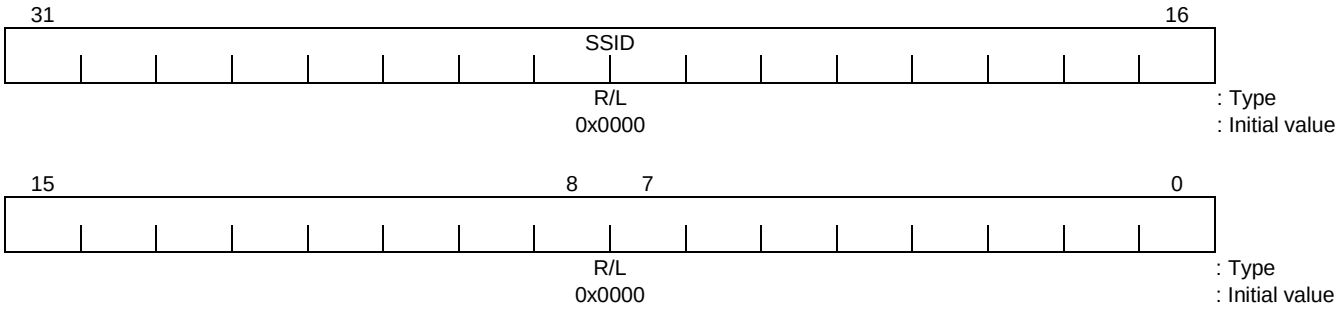
Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x00, R/W) Sets the PCI base address of the Target Access I/O Space. The size of this I/O space is fixed at 256 Bytes selected from 256B to 32KB using TC3[15:8].
7:1	—	Reserved	—
0	IOSI	I/O Space	I/O Space Indicator (Fixed value: 1, R) 1: Indicates that this Base Address Register is for use by the PCI I/O Space.

Figure 10.4.8 P2G I/O Space PCI Base Address Register

10.4.9 Subsystem ID Register (PCISID) 0xD02C

The Subsystem ID field corresponds to the Subsystem ID Register of the PCI Configuration Space, and the Subsystem Vendor ID field corresponds to the Subsystem Vendor ID Register of the PCI Configuration Space. These two fields can be modified by software only when PCICCFG.LCFG=1 after Reset. A write to Register PCICDATA2 when PCICCFG.LCFG=1 will modify the contents of this register. Otherwise this register is Read Only.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



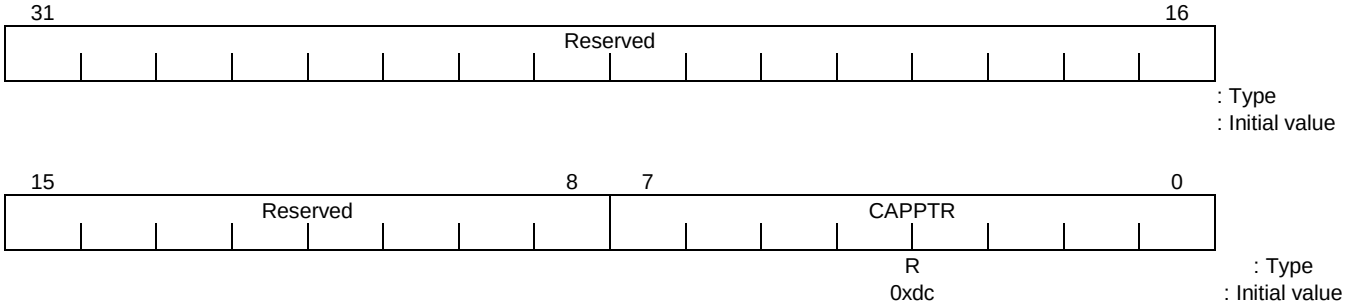
Bits	Mnemonic	Field Name	Description
31:16	SSID	Subsystem ID	Subsystem ID (Initial value: 0x0000, R/L) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board. It is possible to change the Subsystem ID bysoftware.
15:0	SSVID	Subsystem Vendor ID	Subsystem Vendor ID (Initial value: 0x0000, R/L) This register is used to acknowledge either a subsystem that has a PCI device or an add-in board. It is possible to change the Subsystem ID bysoftware.

Figure 10.4.9 Subsystem ID Register

10.4.10 Capabilities Pointer Register (PCICAPPTR) 0xD034

The Capabilities Pointer field corresponds to the Capabilities Pointer Register of the PCI Configuration Space.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	CAPPTR	Capabilities Pointer	Capabilities Pointer (Fixed value: 0xdc, R) Indicates as an offset value the starting address of the capabilities list that indicates extended functions.

Figure 10.4.10 Capabilities Pointer Register

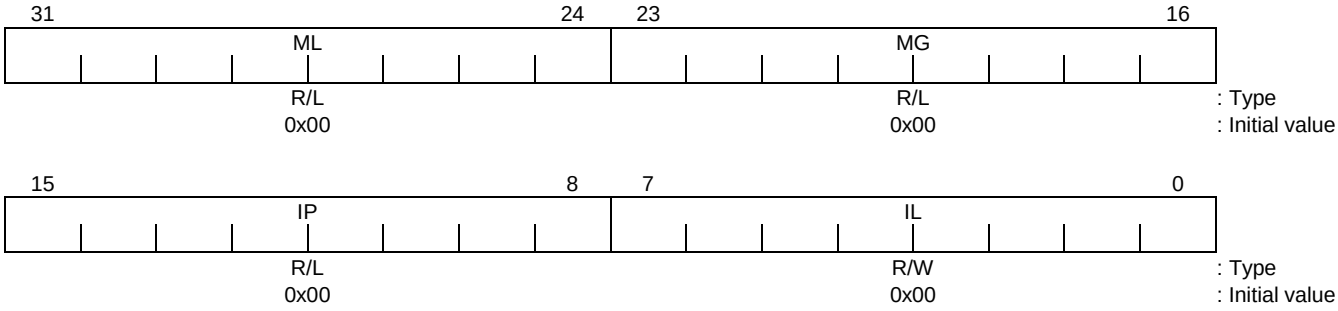
10.4.11 PCI Configuration 2 Register (PCICFG2) 0xD03C

The following fields correspond to the following registers:

Max. Latency field → Max_Lat Register of the PCI Configuration Space
Min. Grant field → Min_Gnt Register of the PCI Configuration Space
Interrupt Pin field → Interrupt Pin Register of the PCI Configuration Space
Interrupt Line field → Interrupt Line Register of the PCI Configuration Space.

A write to Register PCICDATA3 when PCICCFG.LCFG=1 will modify the contents of this register. Otherwise this register is Read Only.

This register cannot be accessed when the PCI Controller is in the Satellite mode.



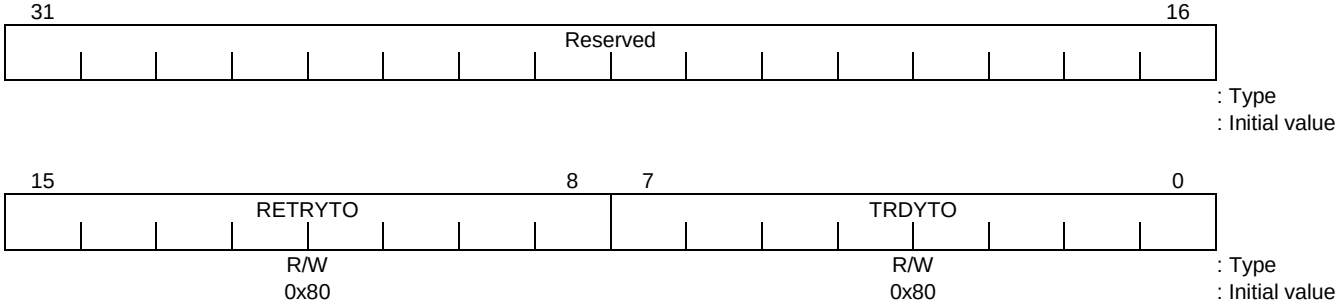
Bits	Mnemonic	Field Name	Description
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Initial value: 0x00, R/L) 00h: Does not use this register to determine PCI Bus priority. 01h-FFh: Specifies the time interval for requesting bus ownership. In units of 250 ns, assuming the PCICLK is 33 MHz. It is possible to change the maximum latency by software.
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Initial value: 0x00, R/L) 00h: Is not used to calculate the latency timer value. 01h-FFh: Sets the time required for Burst transfer. In units of 250 ns, assuming the PCICLK is 33 MHz. It is possible to change this valuesoftware.
15:8	IP	Interrupt Pin	Interrupt Pin (Initial value: 0x00, R/L) Valid values: 00 - 04h 00h: Do not use interrupt signals. 01h: Use Interrupt signal INTA* 02h: Use Interrupt signal INTB* 03h: Use Interrupt signal INTC* 04h: Use Interrupt signal INTD* 05h - FFh: Reserved It is possible to change this value by software. When using either the REQ[2]* signal or the PIO signal to report an interrupt to an external device as the PCI device, please use EEPROM to set the connection with that device.
7:0	IL	Interrupt Line	Interrupt Line (Initial value: 0x00, R/L) This is a readable/writable 8-bit register. The software uses this register to indicate information such as the interrupt signal connection information. Operation of the TX4925 is not affected.

Figure 10.4.11 PCI Configuration 2 Register

10.4.12 G2P Timeout Count Register (G2PTOCNT) 0xD040

The Retry Timeout field corresponds to the Retry Timeout Value Register of the PCI Configuration Space, and the TRDY Timeout field corresponds to the TRDY Timeout Value Register of the PCI Configuration Space.

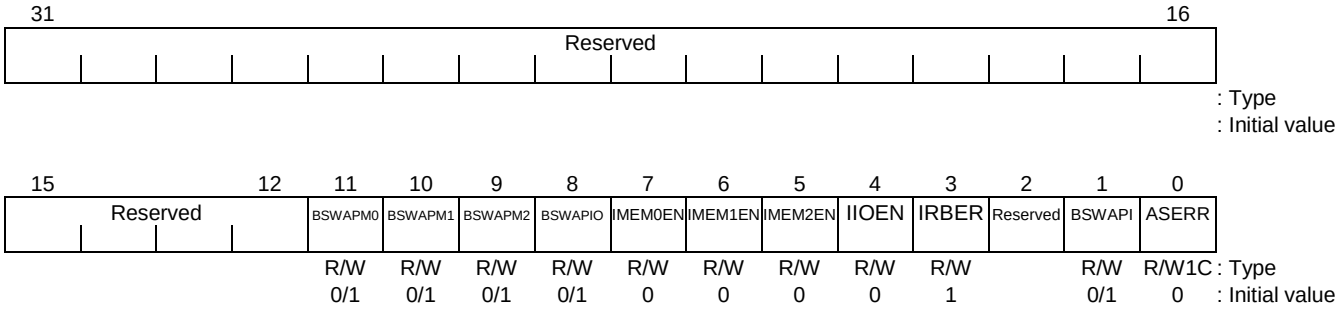
This register cannot be accessed when the PCI Controller is in the Satellite mode.



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:8	RETRYTO	Retry Timeout	Retry Time Out (Initial value: 0x80, R/W) Sets the maximum number of retries to accept when operating as the initiator on the PCI Bus. Ends with an error when receiving more retry terminations than the set maximum number. Setting a “0” disables this timeout function.
7:0	TRDYTO	TRDY Timeout	TRDY Time Out (Initial value: 0x80, R/W) Sets the maximum value of the time to wait for assertion of the TRDY* signal when operating as the initiator on the PCI Bus. Setting a “0” disables this timeout function.

Figure 10.4.12 G2P Timeout Count Register

10.4.13 G2P Configuration Register (G2PCFG) 0xD060



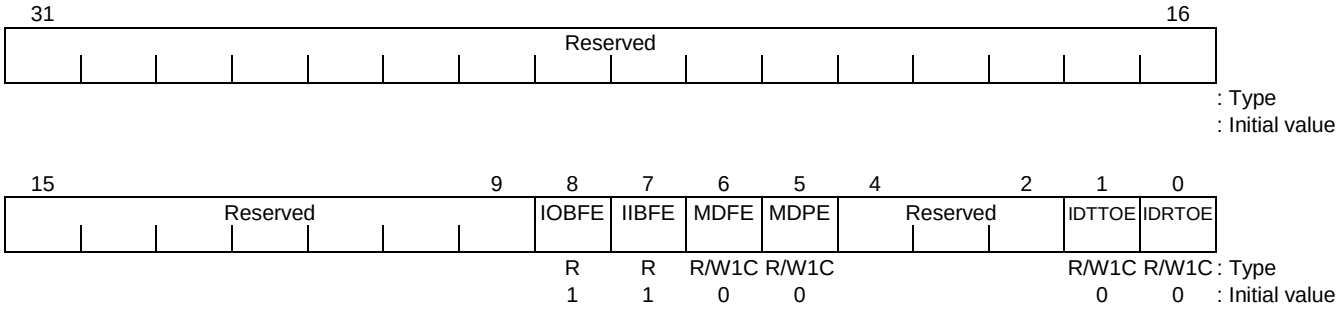
Bits	Mnemonic	Field Name	Description
31:12	—	Reserved	—
11	BSWAPM0	Byte Swap for Memory Space 0	Byte Swap Disable for Memory Space 0 (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 0. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “0” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.
10	BSWAPM1	Byte Swap for Memory Space 1	Byte Swap Disable for Memory Space 1 (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 1. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "0” when in the Big Endian Mode, the byte order of transfer to Memory Space 1 through DWORD (32-bit) access will not change.
9	BSWAPM2	Byte Swap for Memory Space 2	Byte Swap Disable for Memory Space 2 (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 2. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “0” when in the Big Endian Mode, the byte order of transfer to Memory Space 2 through DWORD (32-bit) access will not change.
8	BSWAPIO	Byte Swap for I/O Space	Byte Swap Disable for I/O Space (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of I/O Space. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “0” when in the Big Endian Mode, the byte order of transfer to I/O Space through DWORD (32-bit) access will not change.
7	G2PM0EN	Initiator Memory Space 0 Enable	Initiator Memory Space 0 Enable (Initial value: 0, R/W) Controls PCI initiator access to Memory Space 0. 1: Memory Space 0 is valid. 0: Memory Space 0 is invalid.
6	G2PM1EN	Initiator Memory Space 1 Enable	Initiator Memory Space 1 Enable (Initial value: 0, R/W) Controls PCI initiator access to Memory Space 1. 1: Memory Space 1 is valid. 0: Memory Space 1 is invalid.

Figure 10.4.13 G2P Configuration Register (1/2)

Bits	Mnemonic	Field Name	Description
5	G2PM2EN	Initiator Memory Space 2 Enable	Initiator Memory Space 2 Enable (Initial value: Normal Mode: 0; PCI Boot Mode: 1, R/W) Controls PCI initiator access to Memory Space 2. 1: Memory Space 2 is valid. 0: Memory Space 2 is invalid.
4	G2PIOEN	Initiator I/O Space Enable	Initiator I/O Space Enable (Initial value: 0, R/W) Controls PCI initiator access to the I/O Space.. 1: I/O Space is valid. 0: I/O Space is invalid.
3	IRBER	Bus Error Response Setting During Initiator Access	Initiator Access Bus Error Response (Initial value: 1, R/W) Bus error responses on the G-Bus are controlled when the following phenomena indicated by the PCI Status, Command Register (PICSTATUS) and the G2P Status Register (G2PSTATUS) occur during initiator Read access. Detected Fatal Error (G2PSTATUS.MDFE) Detected Parity Error (G2PSTATUS.MDPE) Received Master Abort (PCISTATUS.RMA) Received Target Abort (PCISTATUS.RTA) Initiator Detected TRDY Time Out Error (G2PSTATUS.IDTTOE) Initiator Detected Retry Time Out Error (G2PSTATUS.IDRTOE) 1: Responds with a Bus error on the G-Bus. 0: Does not respond with a Bus error on the G-Bus. (Normally terminates the transaction on the G-Bus. Read data is invalid.)
2	—	Reserved	—
1	BSWAPI	Byte Swap for Indirect, Config, IACK, and Special cycle	Byte Swap for Indirect, Config, IACK, and Special cycle (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Indirect, Config, IACK, and Special cycle. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “0” when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.
0	ASERR	Assert SERR	Assert SERR (Initial value: 0, R/W1C) A write of 1 will assert SERR for 1 PCI clock. Always read 0.

Figure 10.4.13 G2P Configuration Register (2/2)

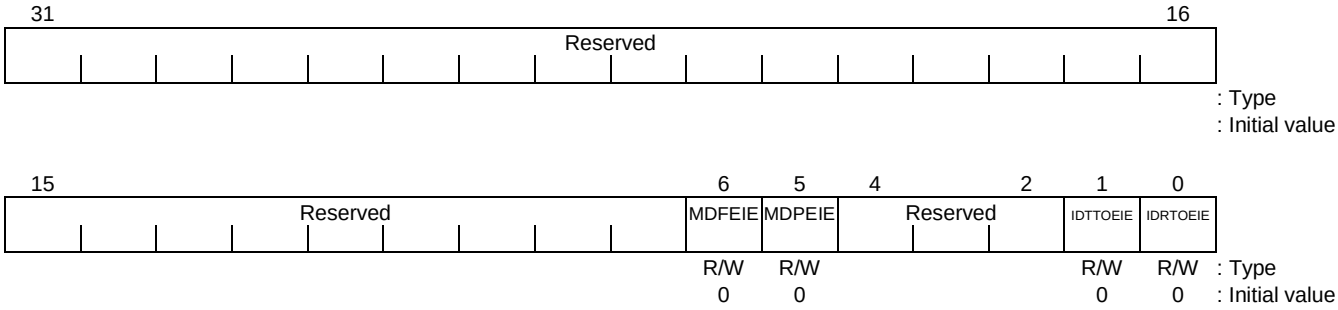
10.4.14 G2P Status Register (G2PSTATUS) 0xD064



Bits	Mnemonic	Field Name	Description
31:9	—	Reserved	—
8	IOBFE	Initiator Out-Bound FIFO Empty	Initiator Out-Bound FIFO Empty (Initial value: 1, R) 1: Indicates that the Initiator Out-Bound FIFO is empty. 0: Indicates that the Initiator Out-Bound FIFO is not empty. This is a diagnostic function.
7	IIBFE	Initiator In-Bound FIFO Empty	Initiator In-Bound FIFO Empty (Initial value: 1, R) 1: Indicates that the Initiator In-Bound FIFO is empty. 0: Indicates that the Initiator In-Bound FIFO is not empty. This is a diagnostic function.
6	MDFE	Master Direct Fatal Error	Master Direct Fatal Error (Initial value: 0, R/W1C) This bit is set when the initiator detects a fatal error in master direct cycle. A fatal error is an event such as one of the following: • Master abort • Target abort • Trdy timeout • Retry timeout The G2PSTATUS.MDFE bit is set if one of the above events occurs.
5	MDPE	Master Direct Parity Error	Master Direct Parity Error (Initial value: 0, R/W1C) This bit is set when the initiator detects a parity error in master direct cycle.
4:2	—	Reserved	—
1	IDTTOE	TRDY Timeout Error	Initiator Detected TRDY Time Out Error (Initial value: 0, R/W1C) This bit is set when the initiator detects a TRDY timeout.
0	IDRTOE	Retry Timeout Error	Initiator Detected Retry Time Out Error (Initial value: 0, R/W1C) This bit is set when the initiator detects a Retry timeout.

Figure 10.4.14 G2P Status Register

10.4.15 G2P Interrupt Mask Register (G2PMASK) 0xD068

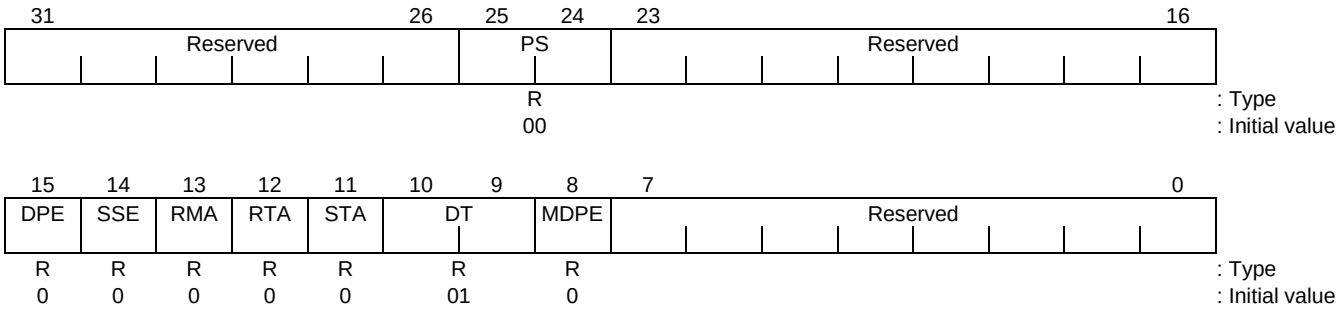


Bits	Mnemonic	Field Name	Description
31:7	—	Reserved	—
6	MDFEIE	Master Direct Fatal Error Interrupt Enable	Master Direct Fatal Error Interrupt Enable (Initial value: 0, R/W) The initiator generates an interrupt when it detects a fatal error in a direct cycle. 1: Generates an interrupt. 0: Does not generate an interrupt.
5	MDPEIE	Master Direct Parity Error Interrupt Enable	Master Direct Parity Error Interrupt Enable (Initial value: 0, R/W) The initiator generates an interrupt when it detects a parity error in a direct cycle. 1: Generates an interrupt. 0: Does not generate an interrupt.
4:2	—	Reserved	Please write “0”.
1	IDTTOEIE	TRDY Timeout Error Interrupt Enable	Initiator Detected TRDY Time Out Interrupt Enable (Initial value: 0, R/W) The initiator generates an interrupt when it detects a TRDY timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.
0	IDRTOEIE	Retry Timeout Error Interrupt Enable	Initiator Detected Retry Time Out Interrupt Enable (Initial value: 0x0, R/W) The initiator generates an interrupt when it detects a Retry timeout. 1: Generates an interrupt. 0: Does not generate an interrupt.

Figure 10.4.15 G2P Interrupt Mask Register

10.4.16 Satellite Mode PCI Status Register (PCISSTATUS) 0xD088

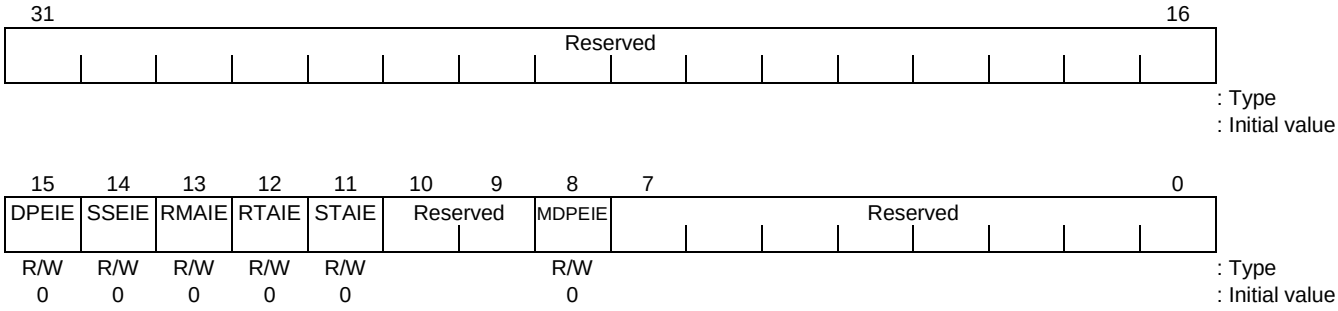
The PCI Status, Command Register (PCISTATUS) or the PMCSR Register of the Configuration Space cannot be accessed when the PCI Controller is in the Satellite mode. It is possible however to read values from either of these registers.



Bits	Mnemonic	Field Name	Description
31:26	—	Reserved	—
25:24	PS	Power State	PowerState (Initial value: 00, R) This is a shadow register of the PowerState field in the PMCSR Register.
23:16	—	Reserved	—
15	DPE	Detected Parity Error	Detected Parity Error (Initial value: 0, R) This is a shadow register of the PCISTATUS.DPE bit.
14	SSE	Signaled System Error	Signaled System Error (Initial value: 0, R) This is a shadow register of the PCISTATUS.SSE bit.
13	RMA	Received Master Abort	Received Master Abort (Initial value: 0, R) This is a shadow register of the PCISTATUS.RMA bit.
12	RTA	Received Target Abort	Received Target Abort (Initial value: 0, R) This is a shadow register of the PCISTATUS.RTA bit.
11	STA	Signaled Target Abort	Signaled Target Abort (Initial value: 0, R) This is a shadow register of the PCISTATUS.STA bit.
10:9	DT	Set DEVSEL Timing	DEVSEL Timing (Fixed value: 01, R) This is a shadow register of the PCISTATUS.DT field.
8	MDPE	Data Parity Detected	Master Data Parity Error Detected (Initial value: 0, R) This is a shadow register of the PCISTATUS.MDPE bit.
7:0	—	Reserved	—

Figure 10.4.16 Satellite Mode PCI Status Register

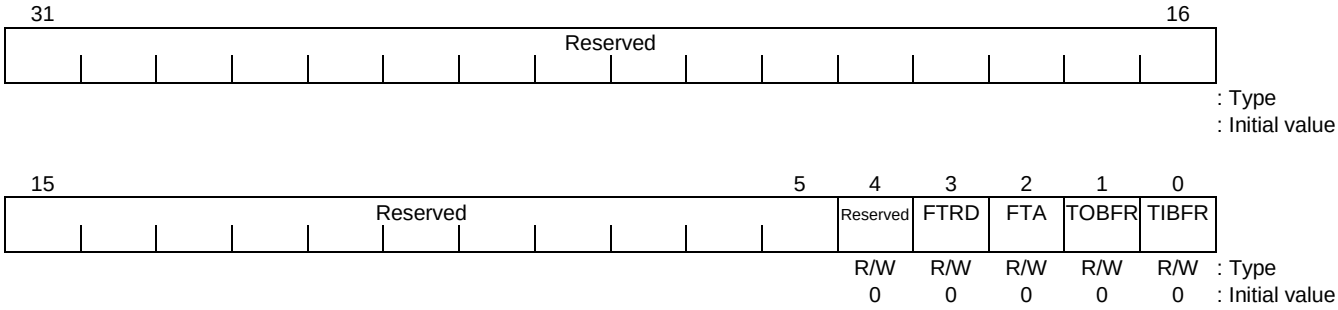
10.4.17 PCI Status Interrupt Mask Register (PCIMASK) 0xD08C



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	DPEIE	Detected Parity Error Interrupt Enable	Detected Parity Error Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a parity error is detected. Usually, this interrupt is masked and a Master Data Parity error signals the error to the system. 1: Generates an interrupt. 0: Does not generate an interrupt.
14	SSEIE	Signaled System Error Interrupt Enable	Signaled System Error Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a system error is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.
13	RMAIE	Received Master Abort Interrupt Enable	Received Master Abort Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a Master Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.
12	RTAIE	Received Target Abort Interrupt Enable	Received Target Abort Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a Target Abort is received. 1: Generates an interrupt. 0: Does not generate an interrupt.
11	STAIE	Signaled Target Abort Interrupt Enable	Signaled Target Abort Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a Target Abort is signaled. 1: Generates an interrupt. 0: Does not generate an interrupt.
10:9	—	Reserved	—
8	MDPEIE	Master Data Parity Detected Interrupt Enable	Master Data Parity Detected Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when data parity is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.
7:0	—	Reserved	—

Figure 10.4.17 PCI Status Interrupt Mask Register

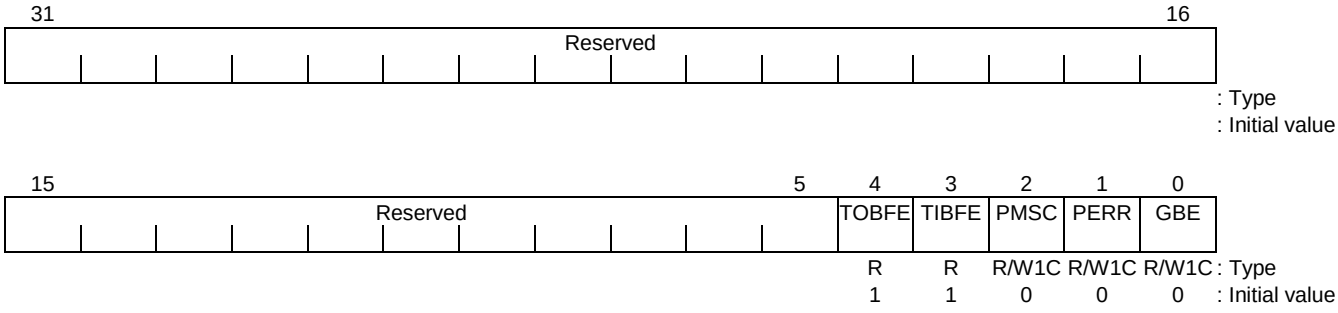
10.4.18 P2G Configuration Register (P2GCFG) 0xD090



Bits	Mnemonic	Field Name	Description
31:5	—	Reserved	—
4	—	Reserved	Don't write one in this bit (Initial value: 0, R/W).
3	FTRD	Force Target Retry/Disconnect	Force Target Retry/Disconnect (Initial value: 0, R/W) The PCI Controller executes Retry Termination on a PCI Read access transaction if this bit is set to "1". This is a diagnostic function.
2	FTA	Force Target Abort	Force Target Abort (Initial value: 0, R/W) The PCI Controller executes a Target Abort on a PCI Read access transaction if this bit is set to "1". This is a diagnostic function.
1	TOBFR	Target read FIFO Reset	Target read FIFO Reset (Initial value: 0, R/W) The PCI Controller flushes the CORE internal Target Out-Bound FIFO when "1" is written to this bit. This bit always reads out "0" when it is read. This is a diagnostic function.
0	TIBFR	Target write FIFO Reset	Target write FIFO Reset (Initial value: 0, R/W) The PCI Controller flushes the CORE internal Target In-Bound FIFO when "1" is written to this bit. This bit always read out "0" when it is read. This is a diagnostic function.

Figure 10.4.18 P2G Configuration Register

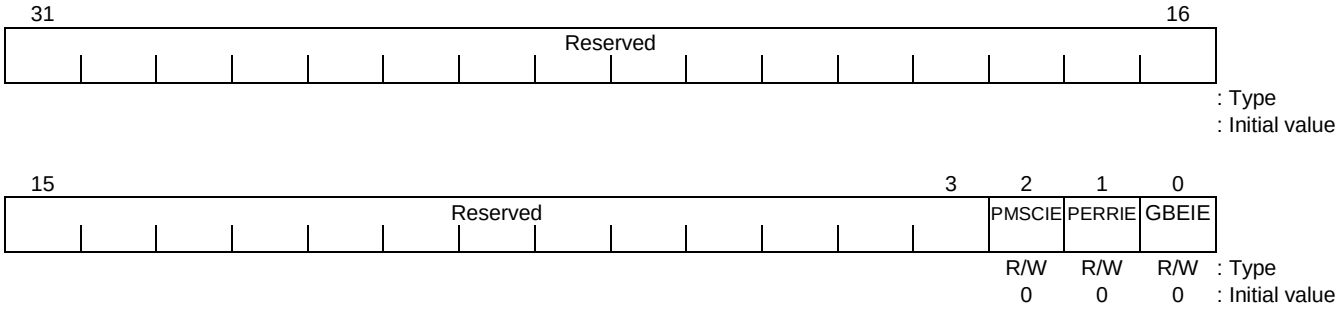
10.4.19 P2G Status Register (P2GSTATUS) 0xD094



Bits	Mnemonic	Field Name	Description
31:5	—	Reserved	—
4	TOBFE	Target Out-Bound FIFO Empty	Target Out-Bound FIFO Empty (Initial value: 1, R) 1: Indicates that the Target Out-Bound FIFO is empty. 0: Indicates that the Target Out-Bound FIFO is not empty. This is a diagnostic function.
3	TIBFE	Target In-Bound FIFO Empty	Target In-Bound FIFO Empty (Initial value: 1, R) 1: Indicates that the Target In-Bound FIFO is empty. 0: Indicates that the Target In-Bound FIFO is not empty. This is a diagnostic function.
2	PMSC	PM State Change Detected	Power Management State Change (Initial value: 0, R/W1C) “1” is set to this bit when the PowerState field of the Power Management Register (PMCSR) is rewritten. This bit is cleared to “0” when a “1” is written to it. This bit is only valid when the PCI Controller is in the Satellite mode.
1	PERR	PERR* Detected	PERR* Occurred (Initial value: 0, R/W1C) Indicates that a Parity error occurred during Target access. 1: Indicates that a Parity error occurred. 0: Indicates that no Parity error has occurred..
0	TGBE	Target G-Bus Error Detect	Target G-Bus Error Detect (Initial value: 0, R/W1C) Indicates that a G-Bus Error occurred when the G-Bus was Target of the PCI cycle. This error is indicated when a timeout occurs on the G-Bus. This bit is only set during PCI Target cycle Bus Errors. 1: Indicates that a G-Bus Error was detected. 0: Indicates that no G-Bus Error was detected.

Figure 10.4.19 P2G Status Register

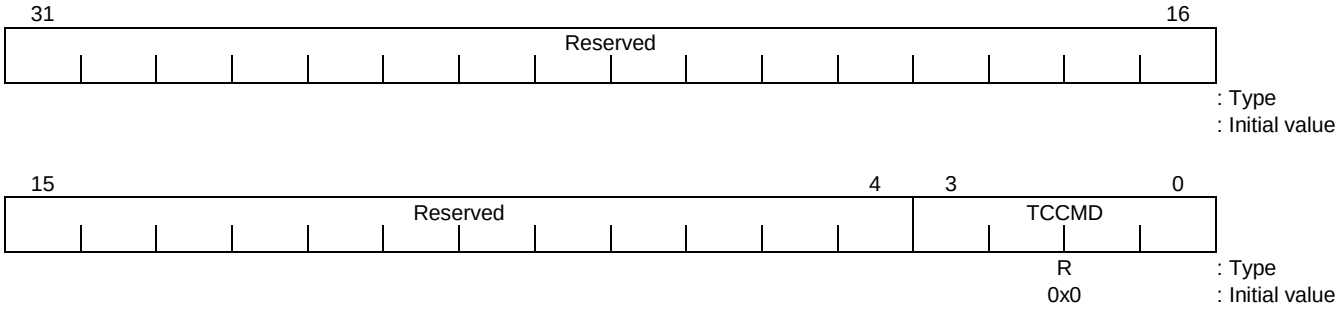
10.4.20 P2G Interrupt Mask Register (P2GMASK) 0xD098



Bits	Mnemonic	Field Name	Description
31:3	—	Reserved	—
2	PMSCIE	Power Management State Change Interrupt Enable	Power Management State Change Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when the PowerState field of the Power Management Register (PMCSR) is rewritten. 1: Generates an interrupt. 0: Does not generate an interrupt.
1	PERRIE	PERR* Detect Interrupt Enable	PERR* Interrupt Enable (Initial value: 0, R/W) This bit generates an interrupt when the Parity Error signal (PERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.
0	GBEIE	G-Bus Bus Error Detect Interrupt Enable	G-Bus Bus Error Interrupt Enable (Initial value: 0, R/W) This bit generates an interrupt when a Bus Error is asserted while the PCI Controller is the G-Bus Master. (Target cycle to G-Bus) 1: Generates an interrupt. 0: Does not generate an interrupt.

Figure 10.4.20 P2G Interrupt Mask Register

10.4.21 P2G Current Command Register (P2GCCMD) 0xD09C



Bits	Mnemonic	Field Name	Description
31:4	—	Reserved	—
3:0	TCCMD	Target Current Command Register	Target Current Command (Initial value: 0x0, R) Indicates the PCI command within the target access process that is currently in progress. This is a diagnostic function.

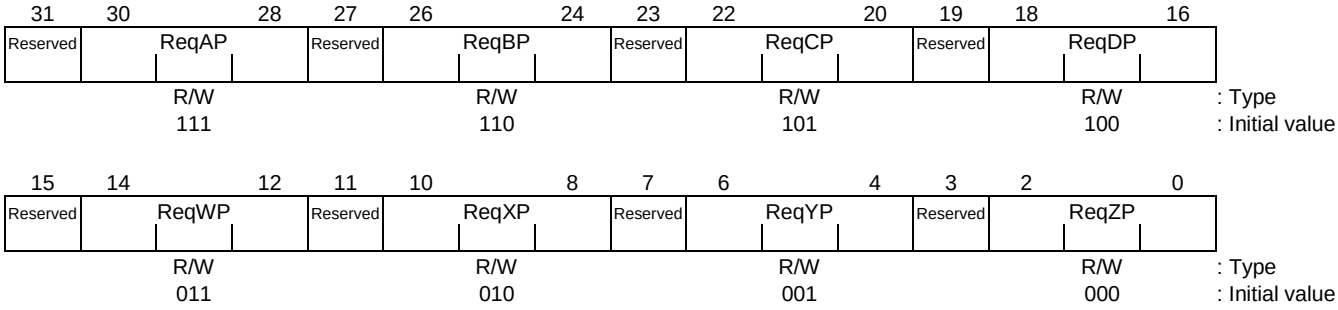
Figure 10.4.21 P2G Current Command Register

10.4.22 PCI Bus Arbiter Request Port Register (PBAREQPORT) 0xD100

This register sets the correlation between each PCI Bus request source (PCI Controller and REQ[3:0]) and each Internal PCI Bus Arbiter Request port (Master A - D, W - Z) (see Figure 10.3.9).

When changing these settings, each of the eight field values must always be set to different values. After changing this register, the Broken Master Register (BM) value becomes invalid since the bit mapping changes.

This register is only valid when using the on-chip PCI Bus Arbiter.



Bits	Mnemonic	Field Name	Description
31	—	Reserved	—
30:28	ReqAP	Request A Port	Request A Port (Initial value: 111, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request A Port (Master A). 111: Makes the PCI Controller Master A. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master A. 010: Makes REQ*[2] Master A. 001: Makes REQ*[1] Master A. 000: Makes REQ*[0] Master A.
27	—	Reserved	—
26:24	ReqBP	Request B Port	Request B Port (Initial value: 110, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request B Port (Master B). 111: Makes the PCI Controller Master B. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master B. 010: Makes REQ*[2] Master B. 001: Makes REQ*[1] Master B. 000: Makes REQ*[0] Master B.
23	—	Reserved	—
22:20	ReqCP	Request C Port	Request C Port (Initial value: 101, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request C Port (Master C). 111: Makes the PCI Controller Master C. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master C. 010: Makes REQ*[2] Master C. 001: Makes REQ*[1] Master C. 000: Makes REQ*[0] Master C.
19	—	Reserved	—

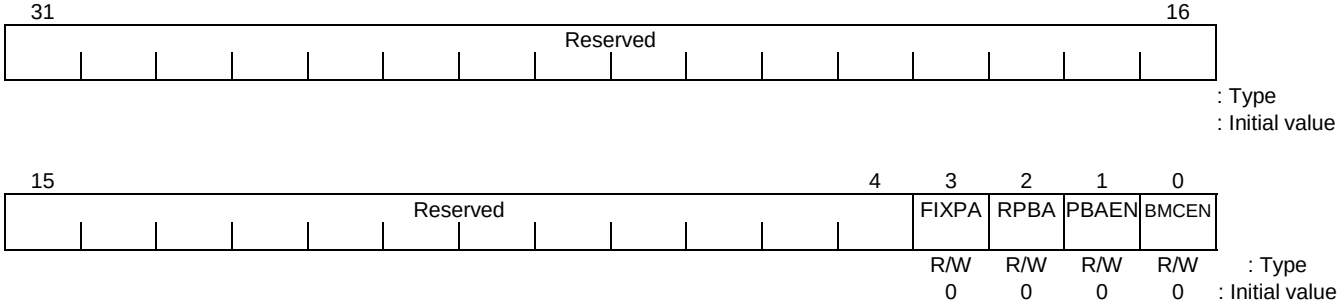
Figure 10.4.22 PCI Bus Arbiter Request Port Register (1/2)

Bits	Mnemonic	Field Name	Description
18:16	ReqDP	Request D Port	Request D Port (Initial value: 100, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request D Port (Master D). 111: Makes the PCI Controller Master D. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master D. 010: Makes REQ*[2] Master D. 001: Makes REQ*[1] Master D. 000: Makes REQ*[0] Master D.
15	—	Reserved	—
14:12	ReqWP	Request W Port	Request W Port (Initial value: 011, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request W Port (Master W). 111: Makes the PCI Controller Master W. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master W. 010: Makes REQ*[2] Master W. 001: Makes REQ*[1] Master W. 000: Makes REQ*[0] Master W.
11	—	Reserved	—
10:8	ReqXP	Request X Port	Request X Port (Initial value: 010, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request X Port (Port X). 111: Makes the PCI Controller Master X. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master X. 010: Makes REQ*[2] Master X. 001: Makes REQ*[1] Master X. 000: Makes REQ*[0] Master X.
7	—	Reserved	—
6:4	ReqYP	Request Y Port	Request Y Port (Initial value: 001, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Y Port (Port Y). 111: Makes the PCI Controller Master Y. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master Y. 010: Makes REQ*[2] Master Y. 001: Makes REQ*[1] Master Y. 000: Makes REQ*[0] Master Y.
3	—	Reserved	—
2:0	ReqZP	Request Z Port	Request Z Port (Initial value: 000, R/W) Sets the PCI Bus Master that connects to the Internal PCI Bus Arbiter Request Z Port (Port Z). 111: Makes the PCI Controller Master Z. 110: Reserved 101: Reserved 100: Reserved 011: Makes REQ*[3] Master Z. 010: Makes REQ*[2] Master Z. 001: Makes REQ*[1] Master Z. 000: Makes REQ*[0] Master Z.

Figure 10.4.22 PCI Bus Arbiter Request Port Register (2/2)

10.4.23 PCI Bus Arbiter Configuration Register (PBACFG) 0xD104

This register is only valid when using the on-chip PCI Bus Arbiter.

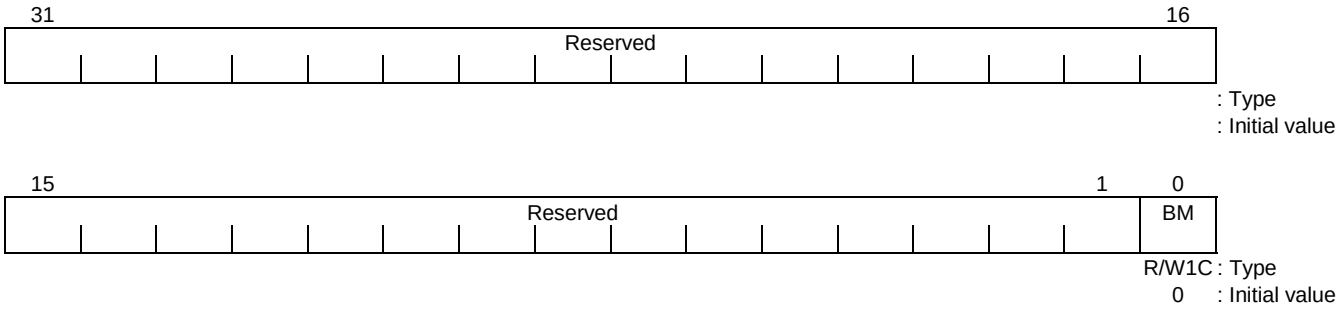


Bits	Mnemonic	Field Name	Description
31:4	—	Reserved	—
3	FIXPA	Fixed Park Master	Fixed Park Master (Initial value: 0, R/W) Selects the method for determining the Park Master. 0: The last Bus Master becomes the Park Master. 1: Internal PCI Bus Arbiter Request Port A is the Park Master.
2	RPBA	Reset PCI Bus Arbiter	Reset PCI Bus Arbiter (Initial value: 0, R/W) Resets the PCI Bus Arbiter. However, the PCI Bus Arbiter Register settings are saved. Please use the software to clear this bit. 1: The PCI Bus Arbiter is currently being reset. 0: The PCI Bus Arbiter is not currently being reset.
1	PBAEN	PCI Bus Arbiter Enable	PCI Bus Arbiter Enable (Initial value: 0, R/W) This is the Bus Arbiter Enable bit. After Reset, External PCI Bus requests to the PCI Arbiter cannot be accepted until this bit is set to “1”. The PCI Controller is the default Parking Master after Reset. 1: Enables the PCI Bus Arbiter. 0: Disables the PCI Bus Arbiter.
0	BMCEN	Broken Master Check Enable	Broken Master Check Enable (Initial value: 0, R/W) Controls Broken Master detection. 1: Enables the Broken PCI Bus Master check. 0: Disables the Broken PCI Bus Master check.

Figure 10.4.23 PCI Bus Arbiter Configuration Register

10.4.24 PCI Bus Arbiter Status Register (PBASTATUS) 0xD108

This register is only valid when using the on-chip PCI Bus Arbiter.

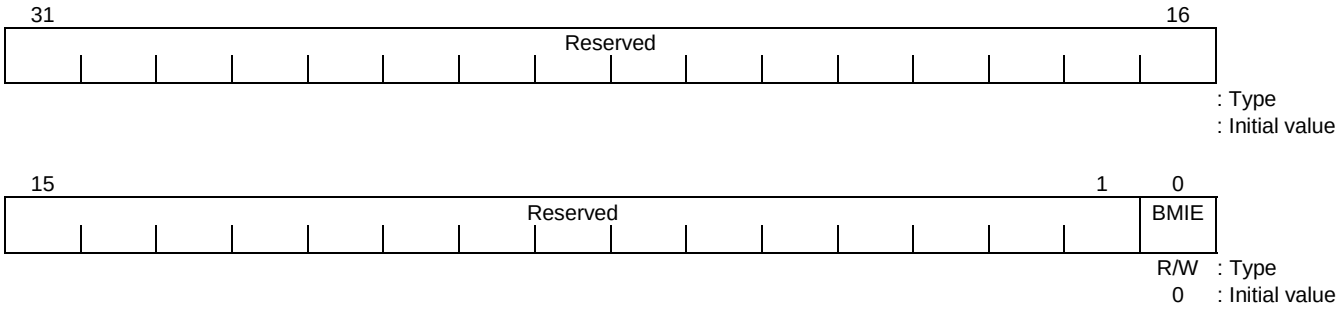


Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	BM	Broken Master Detected	Broken Master Detected (Initial value: 0, R/W1C) This bit indicates that a Broken Master was detected. This bit is set to “1” if even one of the bits in the PCI Bus Arbiter Broken Master Register (PBABM) is “1”. 1: Indicates that a Broken Master was detected. 0: Indicates that no Broken Master has been detected.

Figure 10.4.24 PCI Bus Arbiter Status Register

10.4.25 PCI Bus Arbiter Interrupt Mask Register (PBAMASK) 0xD10C

This register is only valid when using the on-chip PCI Bus Arbiter.



Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	BMIE	Broken Master Detected Interrupt Enable	Broken Master Detected Interrupt Enable (Initial value: 0, R/W) Generates an interrupt when a Broken Master is detected. 1: Generates an interrupt. 0: Does not generate an interrupt.

Figure 10.4.25 PCI Bus Arbiter Interrupt Mask Register

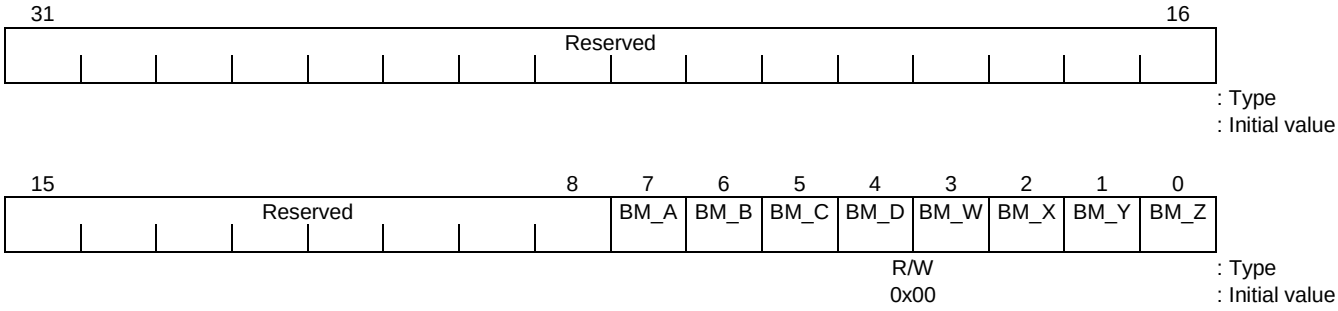
10.4.26 PCI Bus Arbiter Broken Master Register (PBABM) 0xD110

This register indicates the acknowledged Broken Master. This register sets the bit that corresponds to the PCI Master device that was acknowledged as the Broken Master when the Broken Master Check Enable bit (BMCEN) in the PCI Bus Arbiter Configuration Register (PBACFG) is set.

Regardless of the value of the Broken Master Check Enable bit, a PCI Master device is removed from the arbitration scheme when "1" is written to the corresponding BM bit.

This register must be cleared to "0" since bit mapping changes, making this register value invalid when the PCI Bus Arbiter Request Port Register (PBAREQPORT) is changed.

This register is only valid when using the on-chip PCI Bus Arbiter.



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	BM_A	Broken Master	Broken Master A (Initial value: 0, R/W) Indicates whether PCI Bus Master A is a Broken Master. 1: PCI Bus Master A was acknowledged as a Broken Master. 0: PCI Bus Master A was not acknowledged as a Broken Master.
6	BM_B	Broken Master	Broken Master B (Initial value: 0, R/W) Indicates whether PCI Bus Master B is a Broken Master. 1: PCI Bus Master B was acknowledged as a Broken Master. 0: PCI Bus Master B was not acknowledged as a Broken Master.
5	BM_C	Broken Master	Broken Master C (Initial value: 0, R/W) Indicates whether PCI Bus Master C is a Broken Master. 1: PCI Bus Master C was acknowledged as a Broken Master. 0: PCI Bus Master C was not acknowledged as a Broken Master.
4	BM_D	Broken Master	Broken Master D (Initial value: 0, R/W) Indicates whether PCI Bus Master D is a Broken Master. 1: PCI Bus Master D was acknowledged as a Broken Master. 0: PCI Bus Master D was not acknowledged as a Broken Master.
3	BM_W	Broken Master	Broken Master W (Initial value: 0, R/W) Indicates whether PCI Bus Master W is a Broken Master. 1: PCI Bus Master W was acknowledged as a Broken Master. 0: PCI Bus Master W was not acknowledged as a Broken Master.
2	BM_X	Broken Master	Broken Master X (Initial value: 0, R/W) Indicates whether PCI Bus Master X is a Broken Master. 1: PCI Bus Master X was acknowledged as a Broken Master. 0: PCI Bus Master X was not acknowledged as a Broken Master.

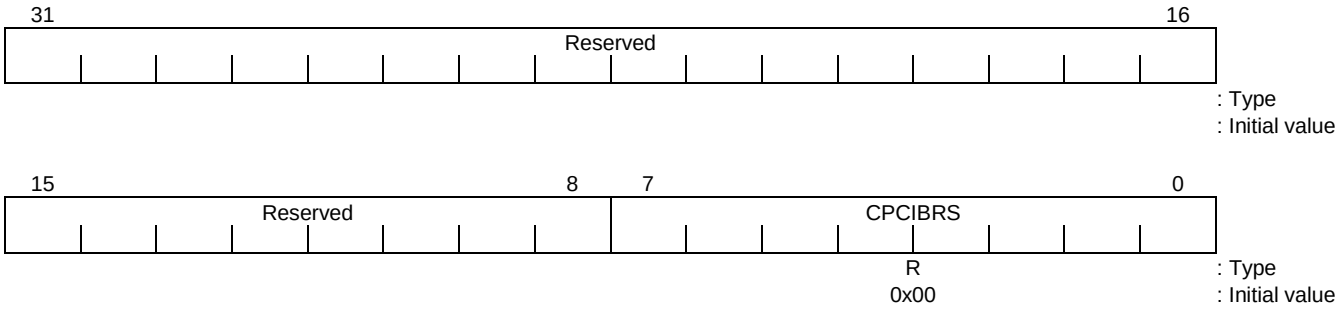
Figure 10.4.26 PCI Bus Arbiter Broken Master Register (1/2)

Bits	Mnemonic	Field Name	Description
1	BM_Y	Broken Master	Broken Master Y (Initial value: 0, R/W) Indicates whether PCI Bus Master Y is a Broken Master. 1: PCI Bus Master Y was acknowledged as a Broken Master. 0: PCI Bus Master Y was not acknowledged as a Broken Master.
0	BM_Z	Broken Master	Broken Master Z (Initial value: 0, R/W) Indicates whether PCI Bus Master Z is a Broken Master. 1: PCI Bus Master Z was acknowledged as a Broken Master. 0: PCI Bus Master Z was not acknowledged as a Broken Master.

Figure 10.4.26 PCI Bus Arbiter Broken Master Register (2/2)

10.4.27 PCI Bus Arbiter Current Request Register (PBACREQ) 0xD114

This register is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

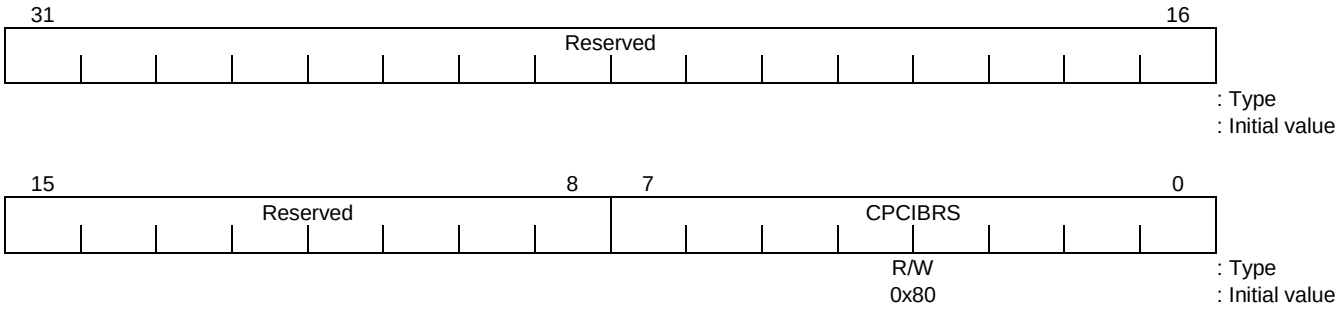


Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	CPCIBRS	Current PCI Bus Request Status	Current PCI Bus Request Status (Initial value: 0x00, R) This register indicates the status of the current PCI Bus Request Input Signal (PCI Controller and REQ*[3:0]). CPCIBRS[7] corresponds to the PCI Controller and CPCIBRS[3:0] correspond to REQ*[3:0].

Figure 10.4.27 PCI Bus Arbiter Current Request Register

10.4.28 PCI Bus Arbiter Current Grant Register (PBACGNT) 0xD118

This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.

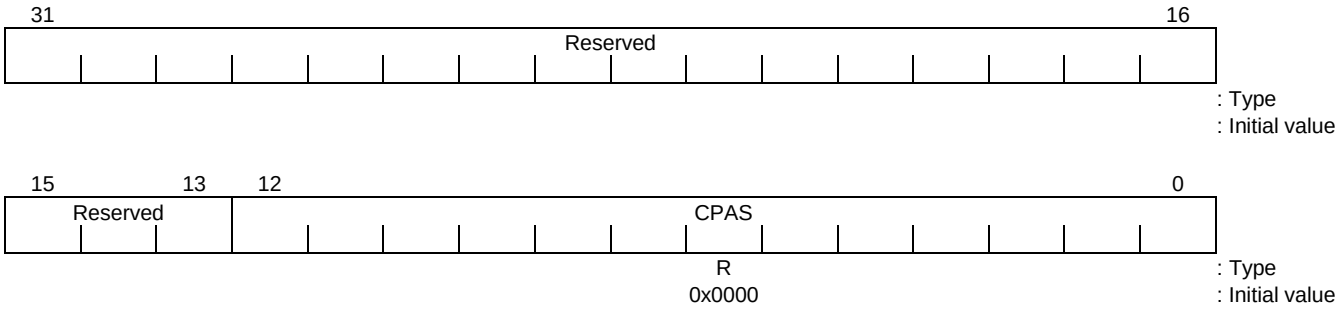


Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	CPCIBGS	Current PCI Grant Status	Current PCI Bus Grant Status (Initial value: 0x80, R/W) This register indicates the current PCI Bus Grant output signal (PCI Controller and GNT*[3:0]). CPCIBGS[7] corresponds to the PCI Controller, and CPCIBGS[3:0] correspond to GNT*[3:0].

Figure 10.4.28 PCI Bus Arbiter Current Grant Register

10.4.29 PCI Bus Arbiter Current State Register (PBACSTATE) 0xD11C

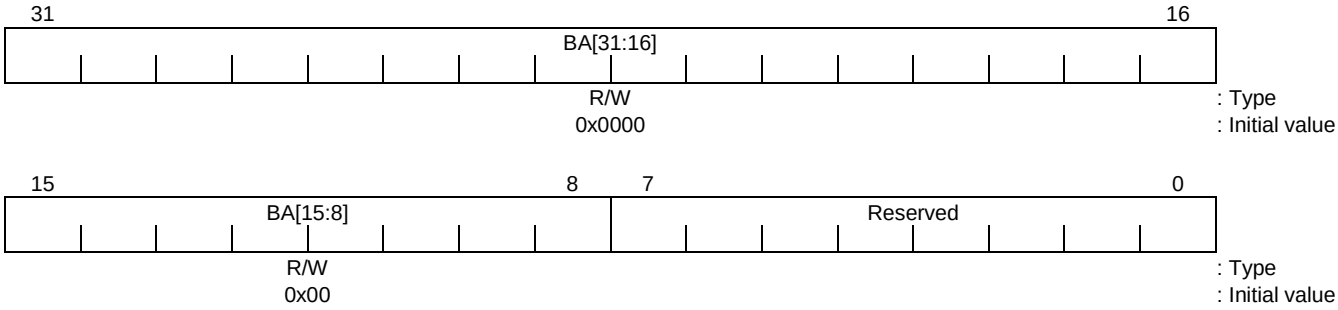
This is a diagnostic register that is only valid when using the on-chip PCI Bus Arbiter.



Bits	Mnemonic	Field Name	Description
31:13	—	Reserved	—
12:0	CPAS	Current PCI bus Arbiter State	Current State of the Arbiter State machine (Initial value: 0x0000, R) Indicates the statemachine of PCI Bus Arbiter. This register is used for debug.

Figure 10.4.29 PCI Bus Arbiter Current State Register

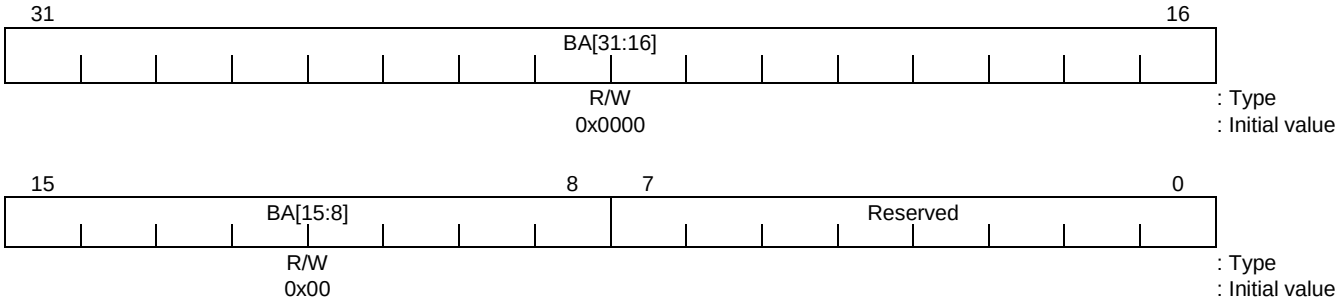
10.4.30 G2P Memory Space 0 G-Bus Base Address Register (G2PM0GBASE) 0xD120



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00, R/W) Sets the G-Bus base bus address of Memory Space 0 for initiator access. Can set the base address in 256-byte units.
7:0	—	Reserved	—

Figure 10.4.30 G2P Memory Space 0 G-Bus Base Address Register

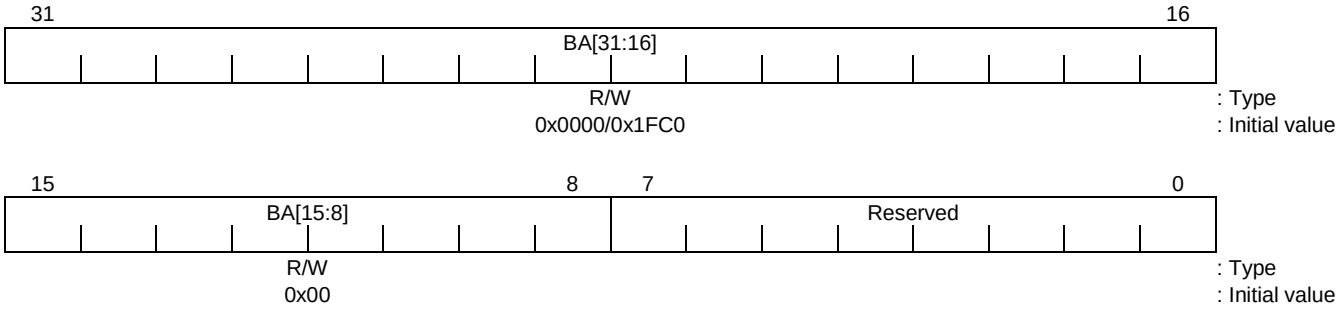
10.4.31 G2P Memory Space 1 G-Bus Base Address Register (G2PM1GBASE) 0xD128



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Memory Space Base Address 1	Base Address (Initial value: 0x0000_00, R/W) Sets the G-Bus base bus address of Memory Space 1 for initiator access. Can set the base address in 256-byte units.
7:0	—	Reserved	—

Figure 10.4.31 G2P Memory Space 1 G-Bus Base Address Register

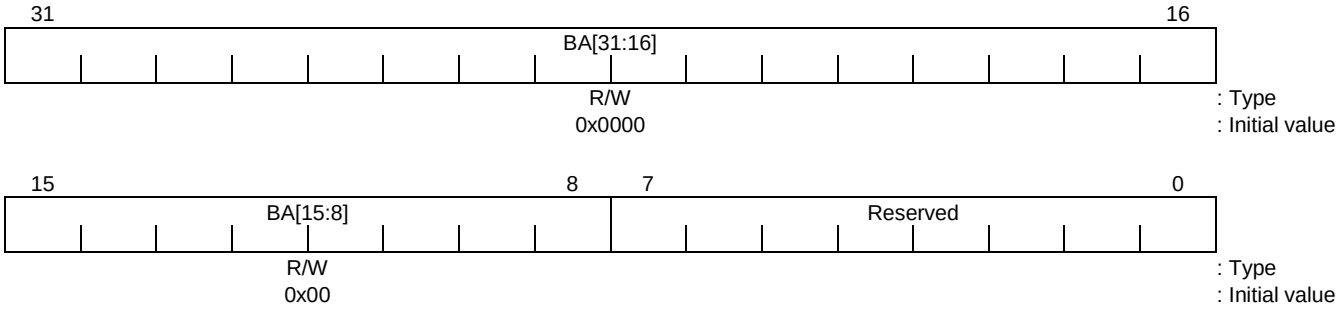
10.4.32 G2P Memory Space 2 G-Bus Base Address Register (G2PM2GBASE) 0xD130



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: Normal Mode: 0x0000_00; PCI Boot Mode: 0x1FC0_00, R/W) Sets the G-Bus base bus address of Memory Space 2 for initiator access. Can set the base address in 256-byte units.
7:0	—	Reserved	—

Figure 10.4.32 G2P Memory Space 2 G-Bus Base Address Register

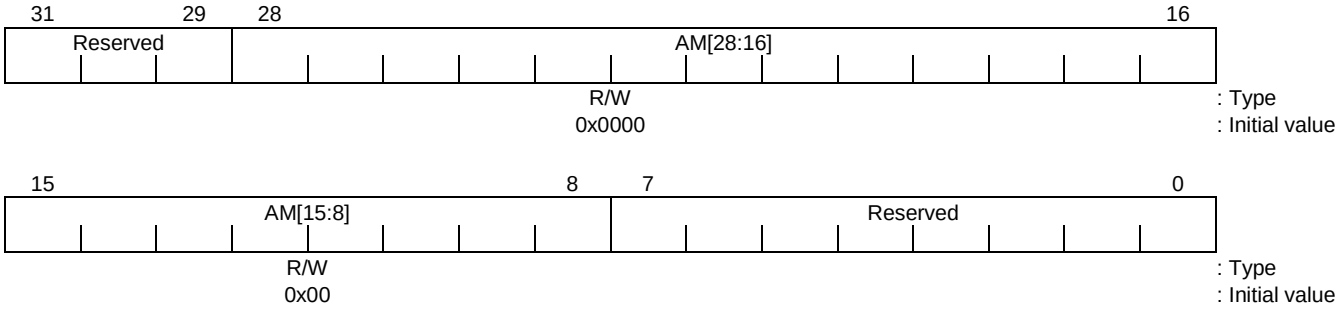
10.4.33 G2P I/O Space G-Bus Base Address Register (G2PIOGBASE) 0xD138



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00, R/W) Sets the G-Bus base bus address of the I/O Memory Space for initiator access. Can set the base address in 256-byte units.
7:0	—	Reserved	—

Figure 10.4.33 G2P I/O Space G-Bus Address Register

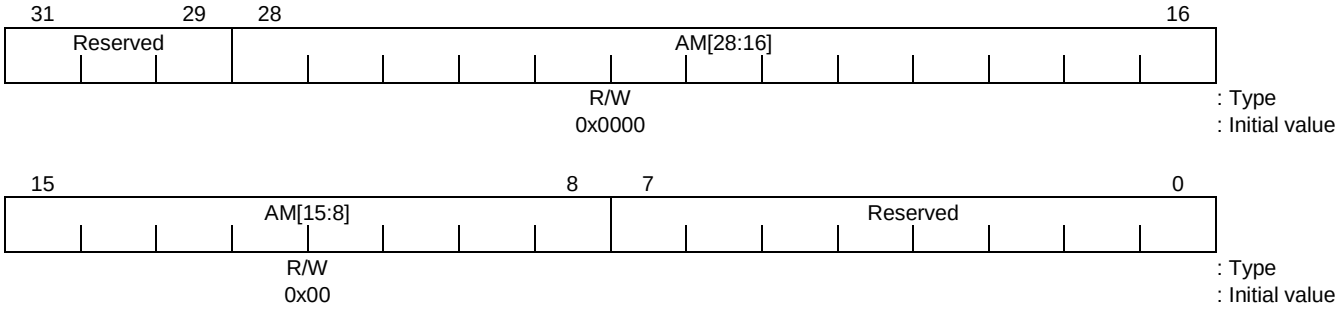
10.4.34 G2P Memory Space 0 Address Mask Register (G2PM0MASK) 0xD140



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:8	AM[28:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Initial value: 0x0000_00, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF_FF00.
7:0	—	Reserved	—

Figure 10.4.34 G2P Memory Space 0 Address Mask Register

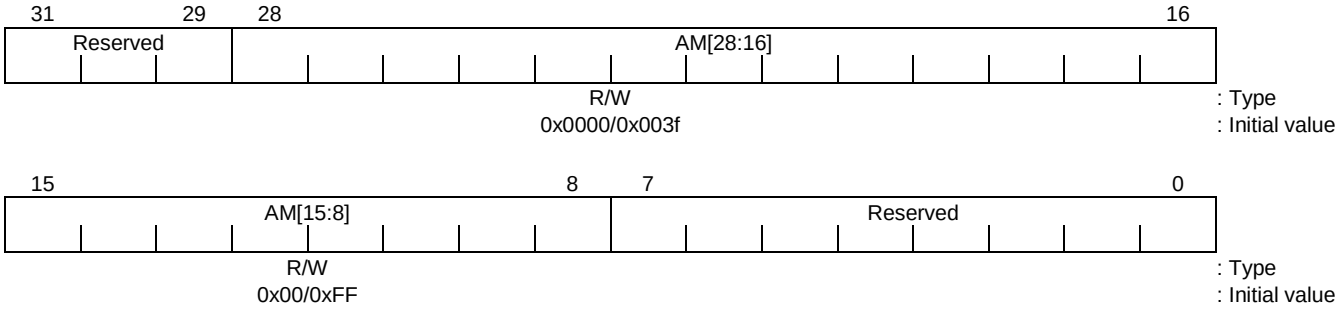
10.4.35 G2P Memory Space 1 Address Mask Register (G2PM1MASK) 0xD144



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:8	AM[31:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Initial value: 0x0000_00, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF_FF00.
7:0	—	Reserved	—

Figure 10.4.35 G2P Memory Space 1 Address Mask Register

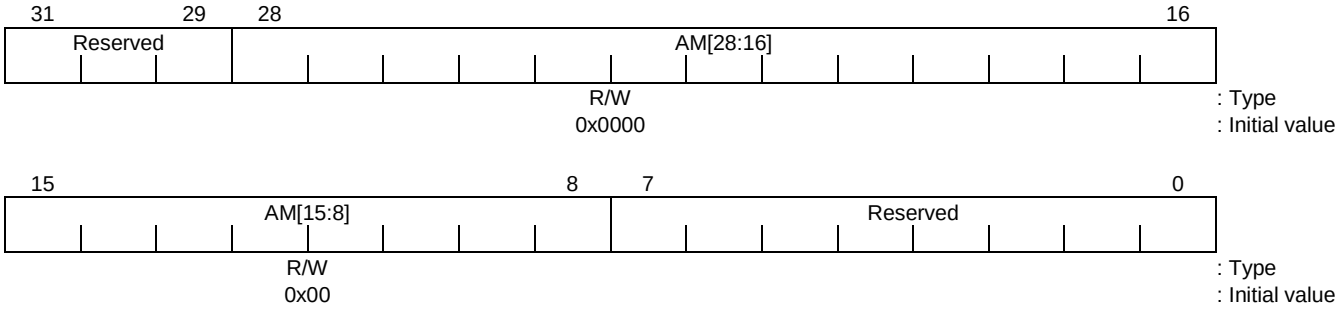
10.4.36 G2P Memory Space 2 Address Mask Register (G2PM2MASK) 0xD148



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:8	AM[31:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Initial value: 0x0000_00, R/W) (Initial value: Normal Mode: 0x0000_00; PCI Boot Mode: 0x003F_FF, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF_FF00.
7:0	—	Reserved	—

Figure 10.4.36 G2P Memory Space 2 Address Mask Register

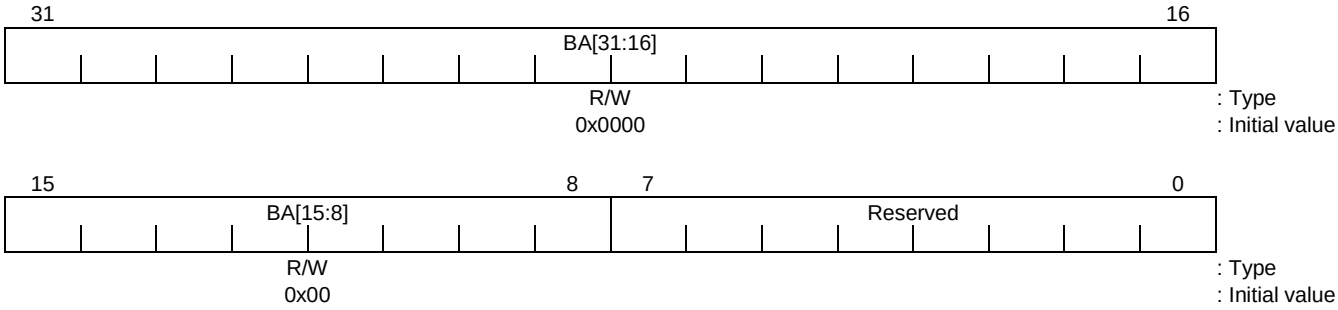
10.4.37 G2P I/O Space Address Mask Register (G2PIOMASK) 0xD14C



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:8	AM[28:8]	Address Mask	G-Bus to PCI-Bus Address Mask (Initial value: 0x0000_00, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 B (0x0000_0100) for example, the value becomes 0x0000_0000.
7:0	—	Reserved	—

Figure 10.4.37 G2P I/O Space Address Mask Register

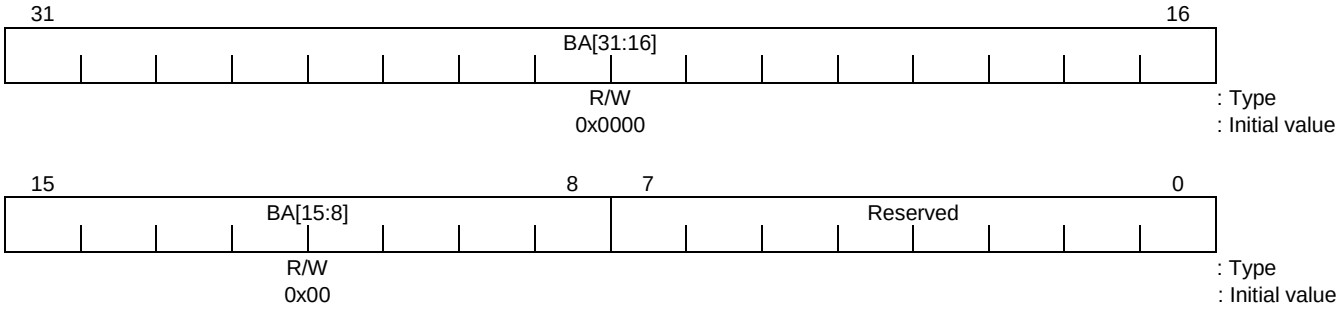
10.4.38 G2P Memory Space 0 PCI Base Address Register (G2PM0PBASE) 0xD150



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00, R/W) Sets the PCI Base address of Memory Space 0 for initiator access. Can set the base address in 256-Byte units.
7:0	—	Reserved	—

Figure 10.4.38 G2P Memory Space 0 PCI Base Address Register

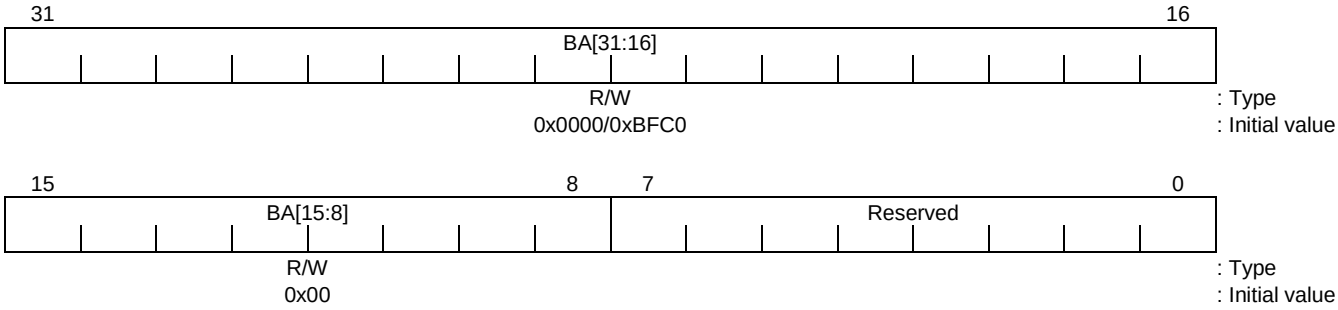
10.4.39 G2P Memory Space 1 PCI Base Address Register (G2PM1PBASE) 0xD158



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00, R/W) Sets the PCI Base address of Memory Space 1 for initiator access. Can set the base address in 256-Byte units.
7:0	—	Reserved	—

Figure 10.4.39 G2P Memory Space 1 PCI Base Address Register

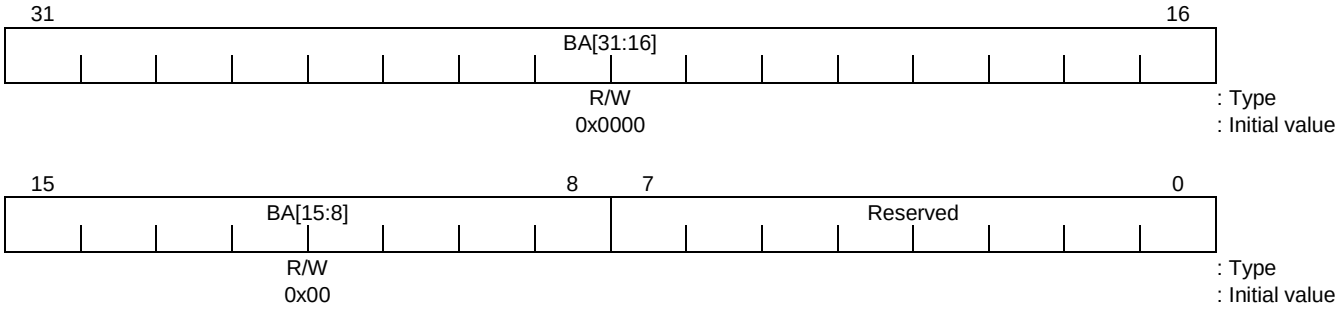
10.4.40 G2P Memory Space 2 PCI Base Address Register (G2PM2PBASE) 0xD160



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00/0xBFC0_00(if PCI boot), R/W) Sets the PCI Base address of Memory Space 2 for initiator access. Can set the base address in 256-Byte units.
7:0	—	Reserved	—

Figure 10.4.40 G2P Memory Space 2 PCI Base Address Register

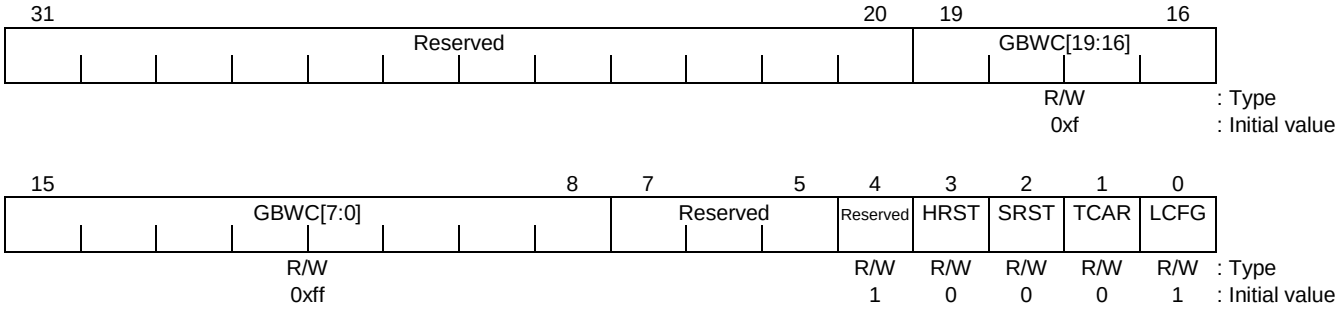
10.4.41 G2P I/O Space PCI Base Address Register (G2PIOPBASE) 0xD168



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Base Address	Base Address (Initial value: 0x0000_00, R/W) Sets the PCI Base address of the I/O Space for initiator access. Can set the base address in 256-Byte units.
7:0	—	Reserved	—

Figure 10.4.41 G2P I/O Space PCI Base Address Register

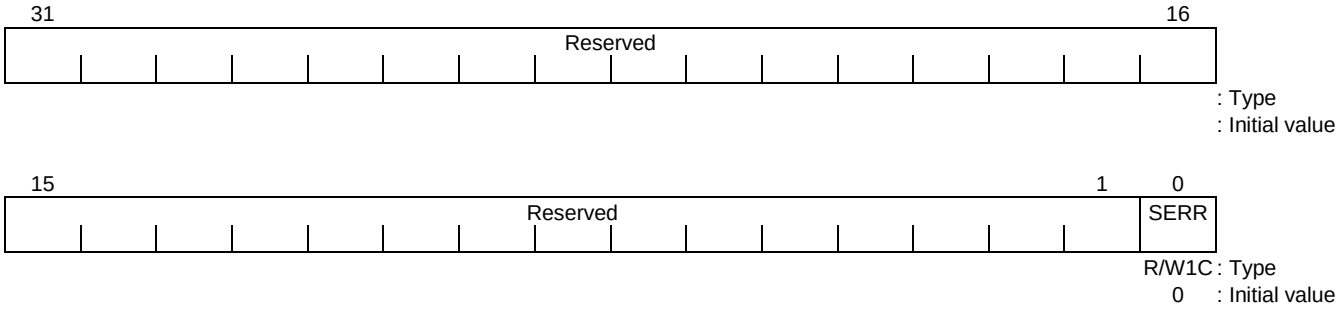
10.4.42 PCI Controller Configuration Register (PCICCFG) 0xD170



Bits	Mnemonic	Field Name	Description
31:20	—	Reserved	—
19:8	GBWC	G-Bus Wait Counter Setting	G Bus Wait Counter (Initial value: 0xfff, R/W) Sets the Retry response counter at the G-Bus during a PCI initiator Read transaction. When the initiator Read access cycle exceeds the setting of this counter, a Retry response is sent to the G-Bus and the G-Bus is released. PCI Read operation continues. This counter uses the G-Bus clock (GBUSCLK) when operating. When 0x000 is set, a Retry response is not sent to the G-Bus by a long response cycle count.
7:5	—	Reserved	—
4	—	Reserved	Note: This bit is always set to “1”. (Initial value: 1, R/W)
3	HRST	Hardware Reset	Hard Reset (Initial value: 0, R/W) Performs PCI Controller hardware reset control. This bit is automatically cleared when Reset ends. This is a diagnostic function. The PCI Controller cannot be accessed for 32 G-Bus clock cycles after this bit is set. 1: Perform a hardware reset on the PCI Controller. 0: Do not perform a hardware reset on the PCI Controller.
2	SRST	Software Reset	Soft Reset (Initial value: 0, R/W) Performs PCI Controller software reset control. Also, please use the software to clear this bit at least four PCI Bus Clock cycles after Reset. Other registers of the PCI Controller cannot be accessed while this bit is set. This bit differs from the Hardware Reset bit (HRST) in that the G-Bus Ack State Machine is not affected. Should be able to R/W any registers. G2P Status Register
1	TCAR	Target Configuration Access Ready	Target Configuration Access Ready (Initial value: 0/1, R/W) Specifies whether to accept PCI access as a target. During PCI boot, configuration access can be received from the PCI Bus after all initialization has completed. This bit becomes “1” only when in the PCI Boot Mode and the Satellite Mode. Operation when this bit is set to “1” then reset to “0” is not defined. 1: Responds to PCI target access. 0: Performs a Retry response to PCI target access.
0	LCFG	Load Configuration Data Register	Load PCI Configuration Data Register (Initial value: 1, R/W) This bit is set to 1 after reset including Hard or Soft reset in PCIC. It can be cleared only by software. When this bit is “1”, the value written to the Configuration Data 0/1/2/3 Register is also written to the Configuration Space Register. This bit must be cleared by software after load because no PCI config cycles will be possible until it is cleared. 1: Load from the Configuration Data 0/1/2/3 Register. 0: No Load.

Figure 10.4.42 PCI Controller Configuration Register

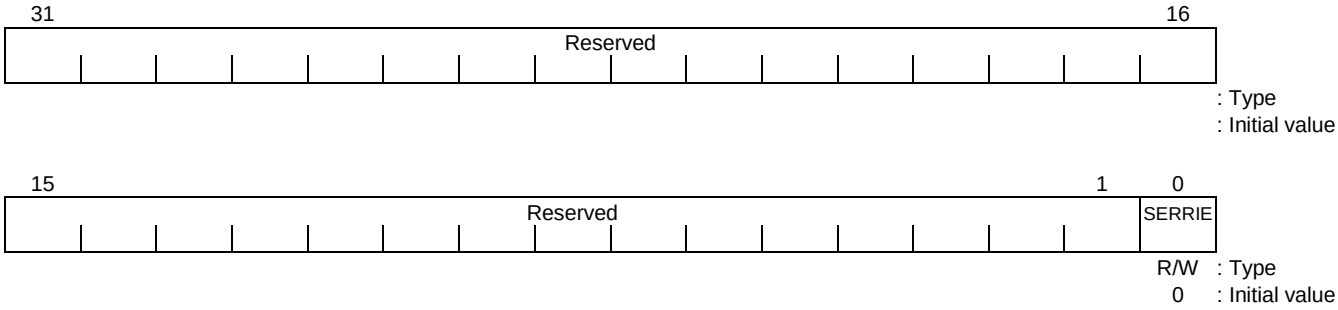
10.4.43 PCI Controller Status Register (PCICSTATUS) 0xD174



Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	SERR	SERR* Detected	SERR* Occurred (Initial value: 0, R/W1C) Indicates that the System Error signal (SERR*) was asserted. This bit is a monitor status bit that records assertion of the SERR* signal even if the TX4925 is not accessing PCI. 1: Indicates that the SERR* signal was asserted. 0: Indicates that the SERR* signal was not asserted.

Figure 10.4.43 PCI Controller Status Register

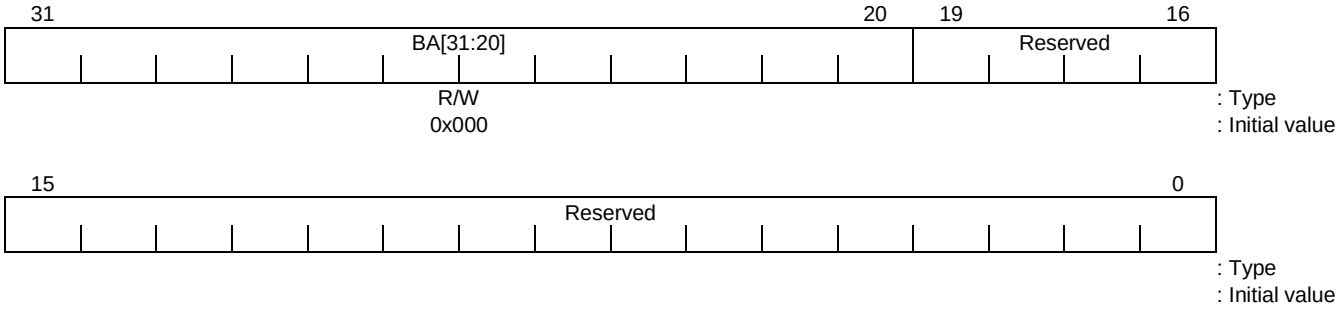
10.4.44 PCI Controller Interrupt Mask Register (PCICMASK) 0xD178



Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	SERRIE	SERR* Detect Interrupt Enable	SERR* Interrupt Enable (Initial value: 0, R/W) This bit generates an interrupt when the System Error signal (SERR*) is asserted. 1: Generates an interrupt. 0: Does not generate an interrupt.

Figure 10.4.44 PCI Controller Interrupt Mask Register

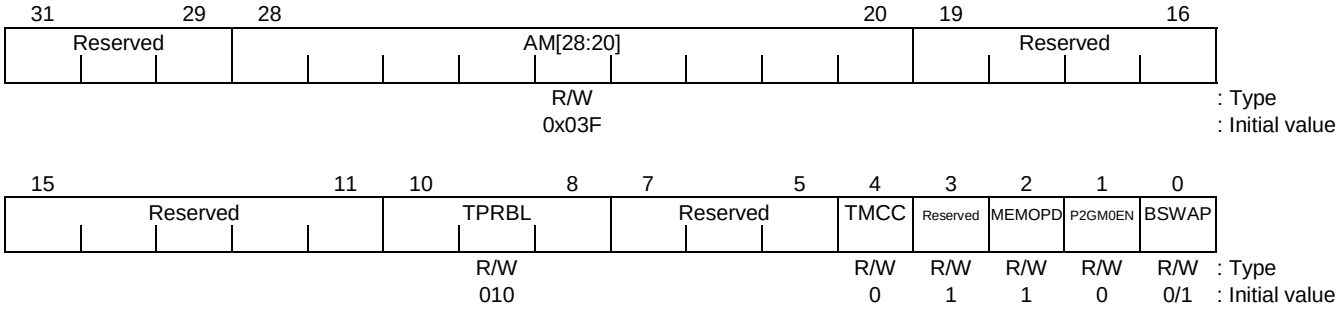
10.4.45 P2G Memory Space 0 G-Bus Base Address Register (P2GM0GBASE) 0xD180



Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Base Address	Base Address 0 (Initial value: 0x000, R/W) Sets the G-Bus base bus address of Memory Space 0 for target access. Can set the base address from 1MB to 512-MB.
19:0	—	Reserved	—

Figure 10.4.45 P2G Memory Space 0 G-Bus Base Address Register

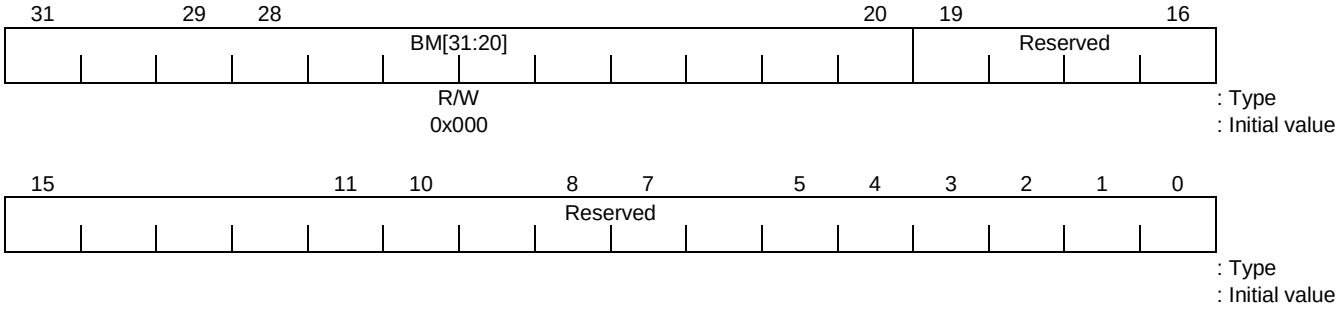
10.4.46 P2G Memory Space 0 Control Register (P2GM0CTR) 0xD184



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:20	AM[28:20]	Address Mask	PCI-Bus to G-Bus Address Mask (Initial value: 0x03F, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF.
19:11	—	Reserved	—
10:8	TPRBL	Target Prefetch Read Burst Length	Target Prefetch Read Burst Length (Initial value: 0x3, R/W) These bits set the G-Bus Burst Size (in DWORDS, (32-bit words)) to be read into the data FIFO during a target memory Read operation. 0x000: Access and transfer 1DWORD (NO BURST) 0x001: Access and transfer 4 DWORDs of data to the target read FIFO. 0x010: Access and transfer 8 DWORDs of data to the target read FIFO. 0x011: Access and transfer 16 DWORDs of data to the target read FIFO. 0x1xx: Access and transfer 32 DWORDs of data to the target read FIFO.
7:5	—	Reserved	—
4	TMCC	Target Memory space 0 Cache Clear	Target Memory space 0 Cache Clear (Initial value: 0, R/W) A write of 1 will flush the Target Memory Cache 0. This bit is cleared automatically. 1: Cache Clear 0: Don't care
3	—	Reserved	Note: This bit is always set to "0". (Initial value: 1, R/W)
2	MEM0PE	Memory 0 Window Prefetch Enable	Memory 0 Window Prefetch Enable (Initial value: 1, R/W) If this bit is set, Prefetching of G-Bus data will occur on Target Memory Reads. If this bit is cleared, 1 Burst of length TPRBL will be done on the G-Bus. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.
1	P2GM0EN	Memory Space 0 Enable	Target Memory Space 0 Enable (Initial value: 0, R/W) Controls whether Memory Space 0 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 0 Base Address Register of the PCI Configuration Register become invalid. Also, "0" is returned to Reads as a response. 1: Validates Memory Space 0 for target access. 0: Invalidates Memory Space 0 for target access.
0	BSWAP	Byte Swap	Byte Swap Enable (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 0 for target access.. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "0" when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.

Figure 10.4.46 P2G Memory Space 0 Control Register

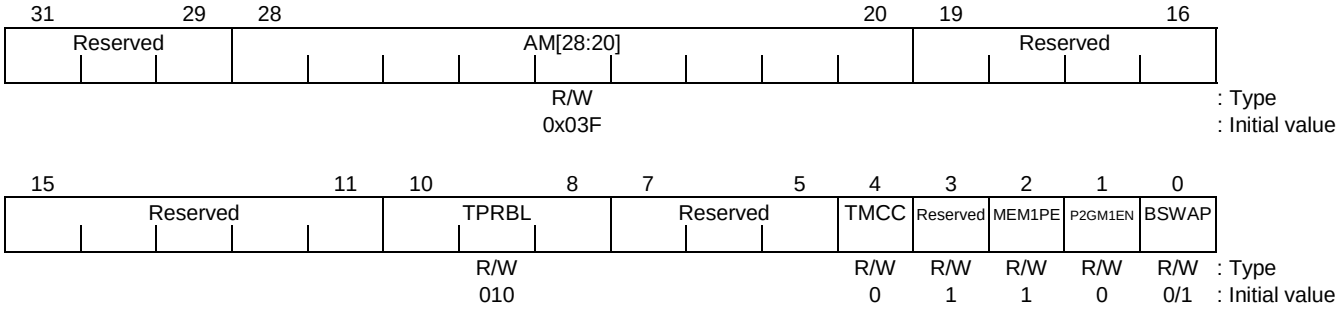
10.4.47 P2G Memory Space 1 G-Bus Base Address Register (P2GM1GBASE) 0xD188



Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Memory Space Base Address 1	Base Address 0 (Initial value: 0x000, R/W) Sets the G-Bus base bus address of Memory Space 1 for target access. Can set the base address from 1 MB to 512 MB.
19:0	—	Reserved	—

Figure 10.4.47 P2G Memory Space 1 G-Bus Base Address Register

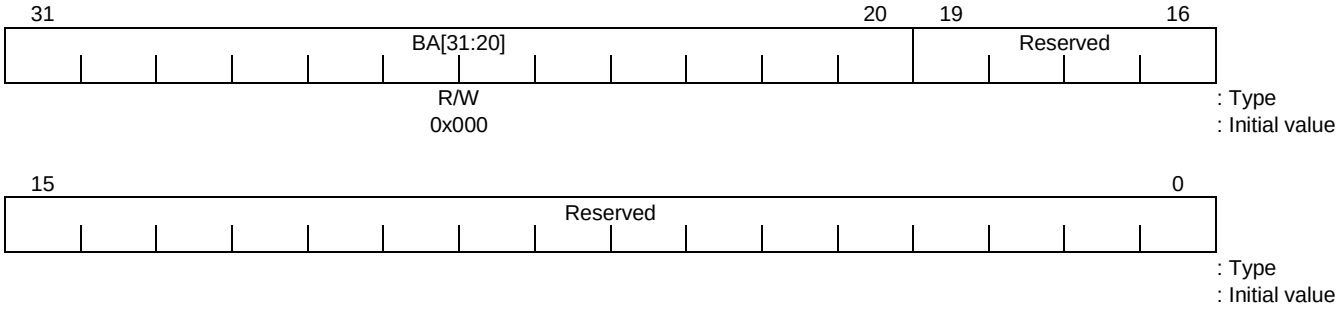
10.4.48 P2G Memory Space 1 Control Register (P2GM1CTR) 0xD18C



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:20	AM[28:20]	Address Mask	PCI-Bus to G-Bus Address Mask (Initial value: 0x03F, R/W) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF.
19:11	—	Reserved	—
10:8	TPRBL	Target Prefetch Read Burst Length	Target Prefetch Read Burst Length (Initial value: 0x3, R/W) These bits set the G-Bus Burst Size (in DWORDS, (32-bit words)) to be read into the data FIFO during a target memory Read operation. 0x000: Access and transfer 1DWORD (NO BURST) 0x001: Access and transfer 4 DWORDs of data to the target read FIFO. 0x010: Access and transfer 8 DWORDs of data to the target read FIFO. 0x011: Access and transfer 16 DWORDs of data to the target read FIFO. 0x1xx: Access and transfer 32 DWORDs of data to the target read FIFO.
7:5	—	Reserved	—
4	TMCC	Target Memory space 1 Cache Clear	Target Memory space 1 Cache Clear (Initial value: 0, R/W) A write of 1 will flush the Target Memory Cache 1. This bit is cleared automatically. 1: Cache Clear 0: Don't care
3	—	Reserved	Note: This bit is always set to "0". (Initial value: 1, R/W)
2	MEM1PE	Memory 1 Window Prefetch Enable	Memory 1 Window Prefetch Enable (Initial value: 1, R/W) If this bit is set, Prefetching of G-Bus data will occur on Target Memory Reads. If this bit is cleared, 1 Burst of length TPRBL will be done on the G-Bus. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.
1	P2GM1EN	Memory Space 1 Enable	Target Memory Space 1 Enable (Initial value: 0, R/W) Controls whether Memory Space 1 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 1 Lower Base Address Register or the Memory Space 1 Upper Base Address Register of the PCI Configuration Register become invalid. Also, "1" is returned to Reads as a response. 1: Validates Memory Space 1 for target access. 0: Invalidates Memory Space 1 for target access.
0	BSWAP	Byte Swap	Byte Swap Disable (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 1 for target access. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "0" when in the Big Endian Mode, the byte order of transfer to Memory Space 0 through DWORD (32-bit) access will not change.

Figure 10.4.48 P2G Memory Space 1 Control Register

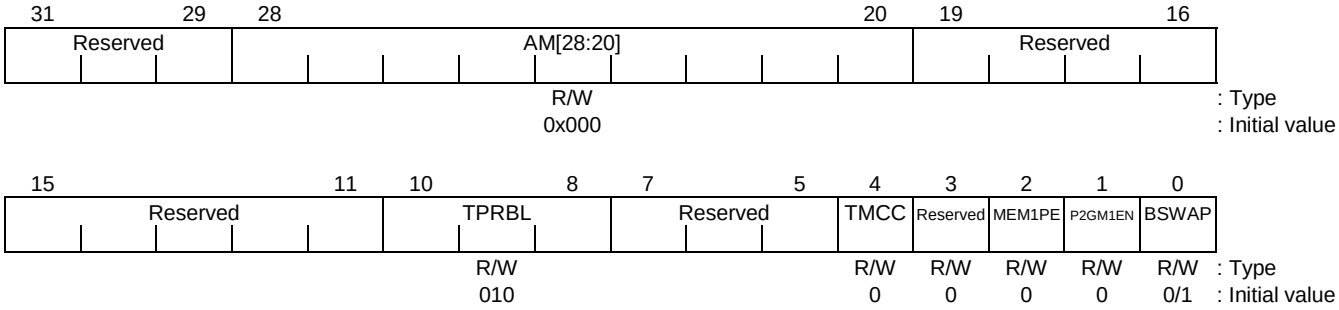
10.4.49 P2G Memory Space 2 G-Bus Base Address Register (P2GM2GBASE) 0xD190



Bits	Mnemonic	Field Name	Description
31:20	BA[31:20]	Memory Space Base Address 2	Base Address 2 (Initial value: 0x000, R/W) Sets the G-Bus base bus address of Memory Space 2 for target access. Can set the base address from 1 MB to 512 MB.
19:0	—	Reserved	—

Figure 10.4.49 P2G Memory Space 2 G-Bus Base Address Register

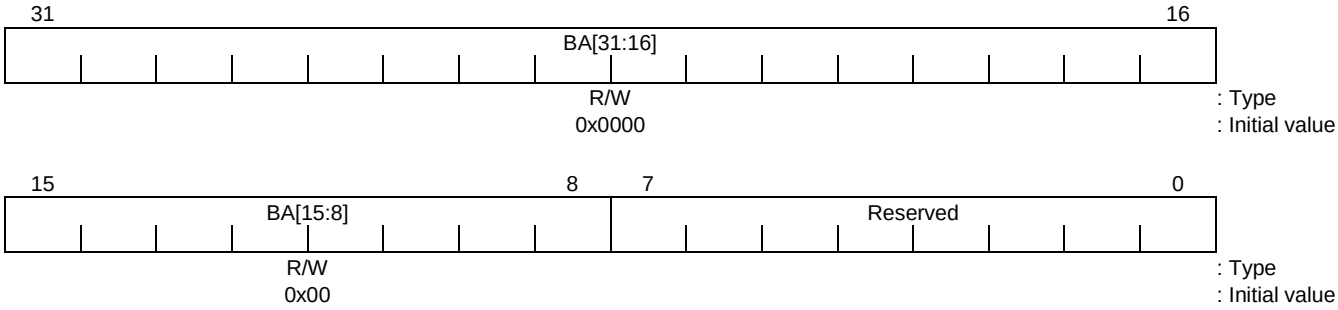
10.4.50 P2G Memory Space 2 Control Register (P2GM2CTR) 0xD194



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:20	AM[28:20]	Address Mask	PCI-Bus to G-Bus Address Mask (Initial value: 0x000, R) Sets the bits to be subject to address comparison. See 10.3.4 for more information. When setting a memory space size of 256 MB (0x1000_0000) for example, the value becomes 0x0FFF.
19:11	—	Reserved	—
10:8	TPRBL	Target Prefetch Read Burst Length	Target Prefetch Read Burst Length (Initial value: 0x3, R/W) These bits set the G-Bus Burst Size (in DWORDS, (32-bit words)) to be read into the data FIFO during a target memory Read operation. 0x000: Access and transfer 1DWORD (NO BURST) 0x001: Access and transfer 4 DWORDs of data to the target read FIFO. 0x010: Access and transfer 8 DWORDs of data to the target read FIFO. 0x011: Access and transfer 16 DWORDs of data to the target read FIFO. 0x1xx: Access and transfer 32 DWORDs of data to the target read FIFO.
7:5	—	Reserved	—
4	TMCC	Target Memory space 2 Cache Clear	Target Memory space 1 Cache Clear (Initial value: 0, R/W) A write of 1 will flush the Target Memory Cache 2. This bit is cleared automatically. 1: Cache Clear 0: Don't care
3	—	Reserved	Note: This bit is always set to "0". (Initial value: 0, R/W)
2	MEM2PE	Memory 2 Window Space Prefetch Enable	Memory 2 Window Prefetch Enable (Initial value: 0, R/W) If this bit is set, Prefetching of G-Bus data will occur on Target Memory Reads. If this bit is cleared, 1 Burst of length TPRBL will be done on the G-Bus. Even if the setting of this bit is changed, prefetchable bits in the Base Address Register of the PCI Configuration Space will not reflect this change. We recommend using the default setting when the PCI Controller is in the Satellite mode.
1	P2GM1EN	Memory Space 2 Enable	Target Memory Space 2 Enable (Initial value: 0, R/W) Controls whether Memory Space 2 for target access is valid or invalid. When this bit is set to invalid, Writes to the Memory Space 2 Lower Base Address Register or the Memory Space 2 Upper Base Address Register of the PCI Configuration Register become invalid. Also, "0" is returned to Reads as a response. 1: Validates Memory Space 2 for target access. 0: Invalidates Memory Space 2 for target access.
0	BSWAP	Byte Swap	Byte Swap Disable (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of Memory Space 2 for target access. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to "0" when in the Big Endian Mode, the byte order of transfer to Memory Space 2 through DWORD (32-bit) access will not change.

Figure 10.4.50 P2G Memory Space 2 Control Register

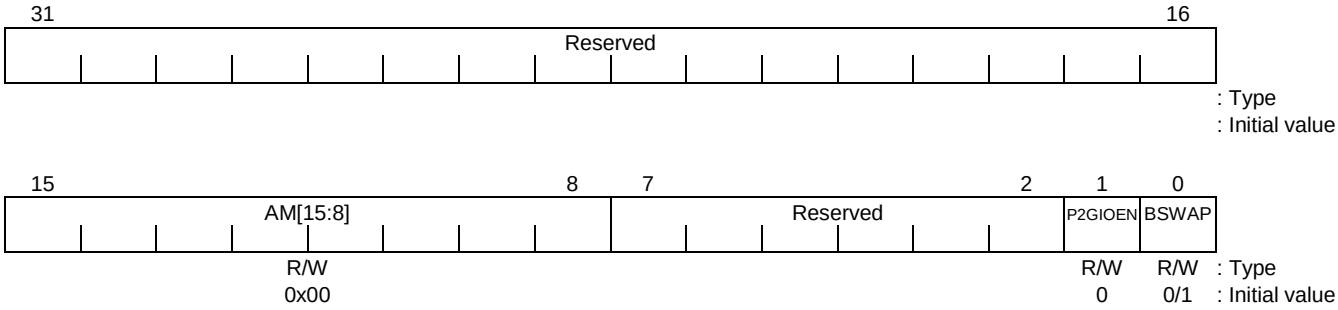
10.4.51 P2G I/O Space G-Bus Base Address Register (P2GIOGBASE) 0xD198



Bits	Mnemonic	Field Name	Description
31:8	BA[31:8]	Memory Space Base Address 2	Base Address 2 (Initial value: 0x0000_00, R/W) Sets the G-Bus base bus address of the I/O Space for target access. Can set the base address from 256 B to 64 KB.
7:0	—	Reserved	—

Figure 10.4.51 P2G I/O Space G-Bus Base Address Register

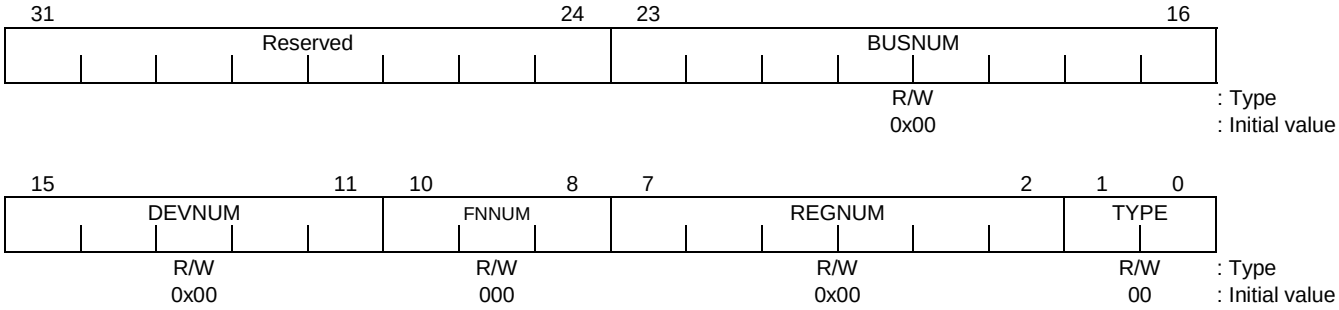
10.4.52 P2G I/O Space Control Register (P2GIOCTR) 0xD19C



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:8	AM[15:8]	Address Mask	PCI-Bus to G-Bus Address Mask (Initial value: 0x00, R/W) Sets the bits to be subject to address comparison. See 10.3.5 for more information. When setting a I/O space size of 256 B (0x0000_0100) for example, the value becomes 0x00.
7:2	—	Reserved	—
1	P2GIOEN	I/O Space Enable	Target I/O Space Enable (Initial value: 0, R/W) Controls whether the I/O Space for target access is valid or invalid. When this bit is set to invalid, Writes to the I/O Space Base Address Register of the PCI Configuration Register become invalid. Also, “0” is returned to Reads as a response. 1: Validates I/O Space for target access. 0: Invalidates I/O Space for target access.
0	BSWAP	Byte Swap	Byte Swap Disable (Initial value: Little Endian Mode: 0; Big Endian Mode: 1, R/W) Sets the byte swapping of the I/O Space for target access. 0: Do not perform byte swapping. 1: Perform byte swapping. Please use the default state in most situations. If this bit is changed to “0” when in the Big Endian Mode, the byte order of transfer to the I/O Space through DWORD (32-bit) access will not change.

Figure 10.4.52 P2G I/O Space Control Register

10.4.53 G2P Configuration Address Register(G2PCFGADRS) 0xD1A0[m3]



Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23:16	BUSNUM	Bus Number	Bus Number (Initial value: 0x00, R/W) Indicates the target PCI Bus Number (one of 256).
15:11	DEVNUM	Device Number	Device Number (Initial value: 0x00, R/W) This field is used to identify the target physical device number. (This is one number out of 32 devices. 21 of these 32 devices are used.) When in the address phase of Type 0 configuration access, AD[31:11] of the upper 21 address lines are used as the IDSEL signal. 0x00: Use AD [11] as IDSEL. 0x01: Use AD [12] as IDSEL. 0x02: Use AD [13] as IDSEL. : : 0x13: Use AD [30] as IDSEL. 0x14: Use AD [31] as IDSEL. 0x15 - 0x1F: Reserved
10:8	FNNUM	Function Number	Function Number (Initial value: 000, R/W) This field is used to identify the target logic function number (one out of 8).
7:2	REGNUM	Register Number	Register Number (Initial value: 0x00, R/W) This field is used to identify the DWORD (one out of 64) inside the Configuration Space of the target function
1:0	TYPE	Type	Type (Initial value: 00, R/W) This field is used to identify the address type in the address phase of the target function configuration cycle. 0x0: Type 0 configuration (Use the AD[31:11] signal as the IDSEL signal.) 0x1: Type 1 configuration (Output all bits unchanged as the address to the AD[ ] signal.)

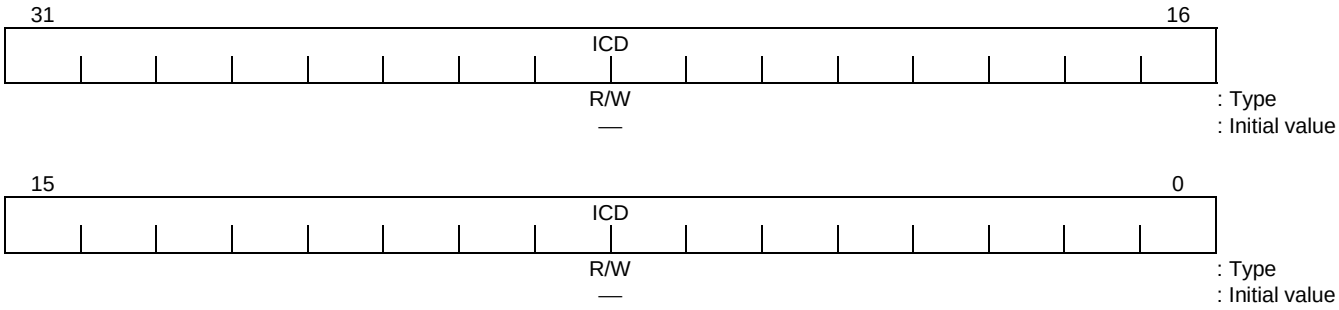
Figure 10.4.53 G2P Configuration Address Register

10.4.54 G2P Configuration Data Register (G2PCFGDATA) 0xD1A4

This is the only register that supports Byte access and 16-bit Word access. The upper address bit of the PCI Configuration Space is specified by the G2P Configuration Address Register (G2PCFGADRS). The lower two bits of the address are specified by the lower two bits of the offset address in this register as shown in Table 10.4.2. The TX4925 also has the ability to swap Byte Enables for Configuration Cycles by setting G2PCFG.BSWAPI.

Table 10.4.2 PCI Configuration Space Access Address

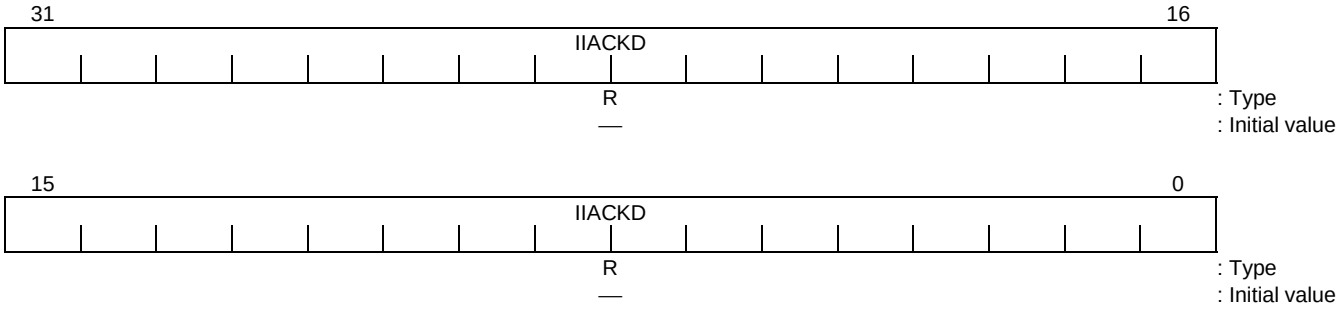
Access Size	Configuration Space Address [1:0]	Offset Address	
		Little Endian Mode	Big Endian Mode
32-bit	00	0xD1A4	0xD1A4
16-bit	00	0xD1A4	0xD1A6
	10	0xD1A6	0xD1A4
8-bit	00	0xD1A4	0xD1A7
	01	0xD1A5	0xD1A6
	10	0xD1A6	0xD1A5
	11	0xD1A7	0xD1A4



Bits	Mnemonic	Field Name	Description
31:0	ICD	Initiator Configuration Data	Initiator Configuration Data Register (Initial value: R/W) This is a data port that is used when performing initiator PCI configuration access. PCI configuration Read or Write transactions are issued when this register is read to or written from.

Figure 10.4.54 G2P Configuration Data Register

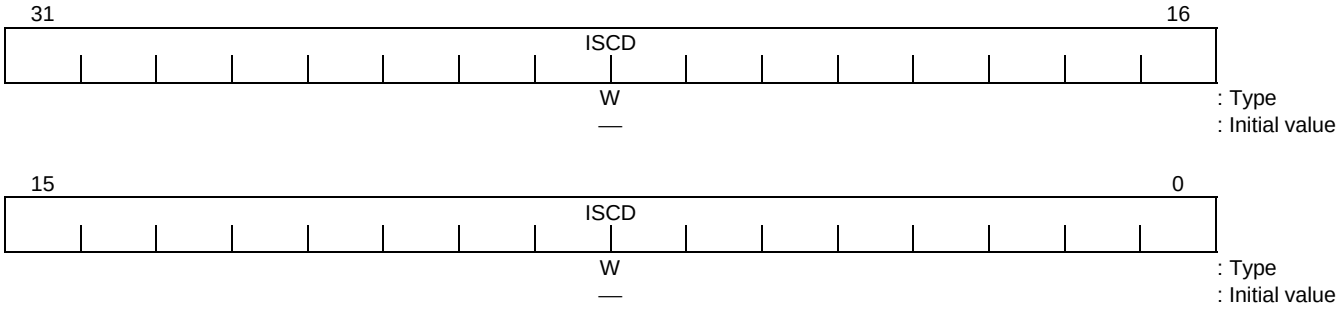
10.4.55 G2P Interrupt Acknowledge Data Register (G2PINTACK) 0xD1C8



Bits	Mnemonic	Field Name	Description
31:0	IIACKD	Initiator Interrupt Acknowledge Address Port	Initiator Interrupt Acknowledge Address Port (Initial value: –, R) An Interrupt Acknowledge cycle is generated on the PCI Bus when this register is read. The data that is returned by this Read transaction becomes the Interrupt Acknowledge data.

Figure 10.4.55 G2P Interrupt Acknowledge Data Register

10.4.56 G2P Special Cycle Data Register (G2PSPC) 0xD1CC

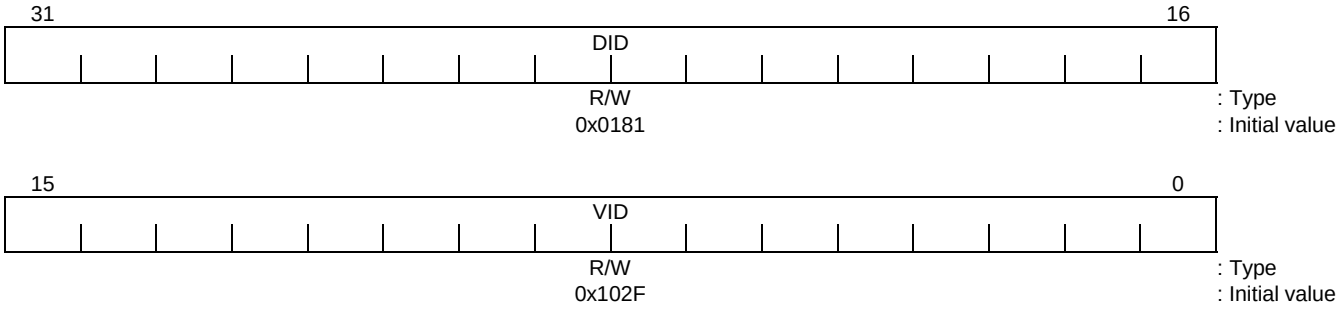


Bits	Mnemonic	Field Name	Description
31:0	ISCD	Initiator Special Cycle Data Port	Initiator Special Cycle Data Port (Initial value: –, W) When this register is written to, Special Cycles are generated on the PCI Bus depending on the data that is written.

Figure 10.4.56 G2P Special Cycle Data Register

10.4.57 Configuration Data 0 Register (PCICDATA0) 0xD1E0

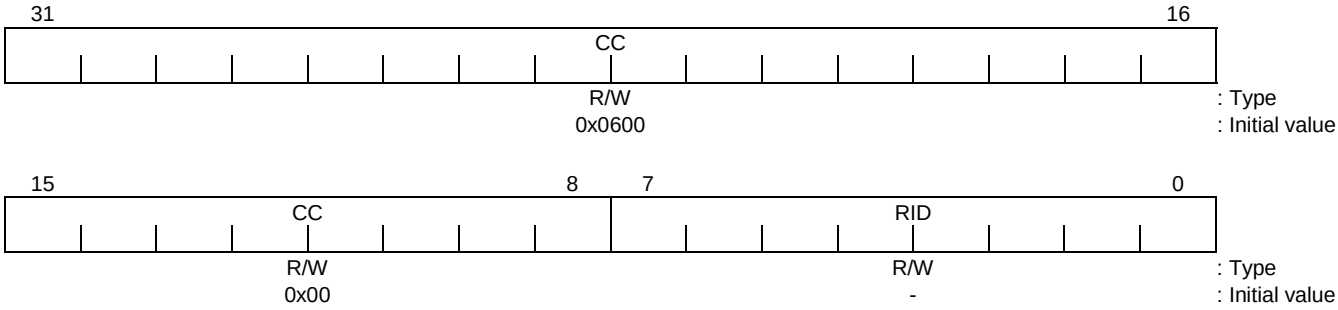
If PCICCFG.LCFG is set, a write to PCICDATA0, 1, 2 & 3 will modify the contents of the associated Configuration registers in the PCI Core.



Bits	Mnemonic	Field Name	Description
31:16	DID	Device ID	Device ID (Initial value: 0x0181, R/W) This is the data loaded in the Device ID Register of the PCI Configuration Space.
15:0	VID	Vendor ID	Vendor ID (Initial value: 0x102F, R/W) This is the data loaded in the Vendor ID Register of the PCI Configuration Space.

Figure 10.4.57 Configuration Data 0 Register

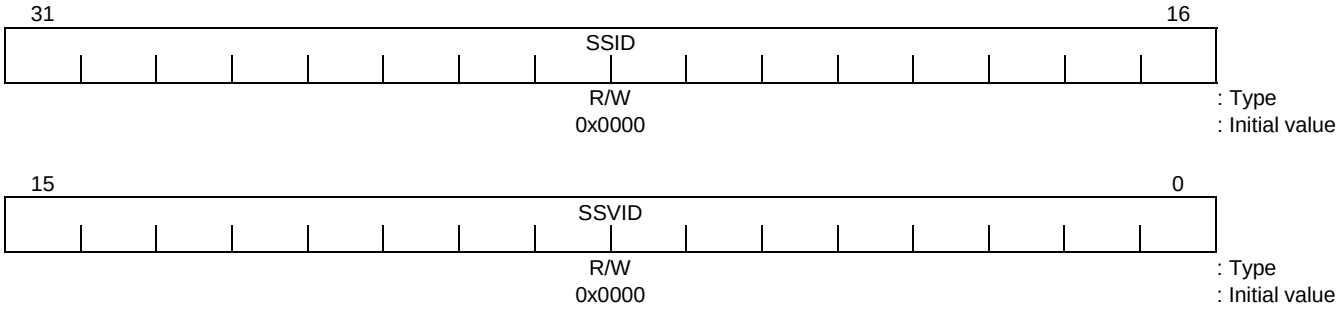
10.4.58 Configuration Data 1 Register (PCICDATA1) 0xD1E4



Bits	Mnemonic	Field Name	Description
31:8	CC	Class Code	Class Code (Initial value: 0x0600_00, R/W) This is the data loaded in the Class Code Register of the PCI Configuration Space.
7:0	RID	Revision ID	Revision ID (Initial value: –, R/W) This is the data loaded in the Revision ID Register of the PCI Configuration Space.

Figure 10.4.58 Configuration Data 1 Register

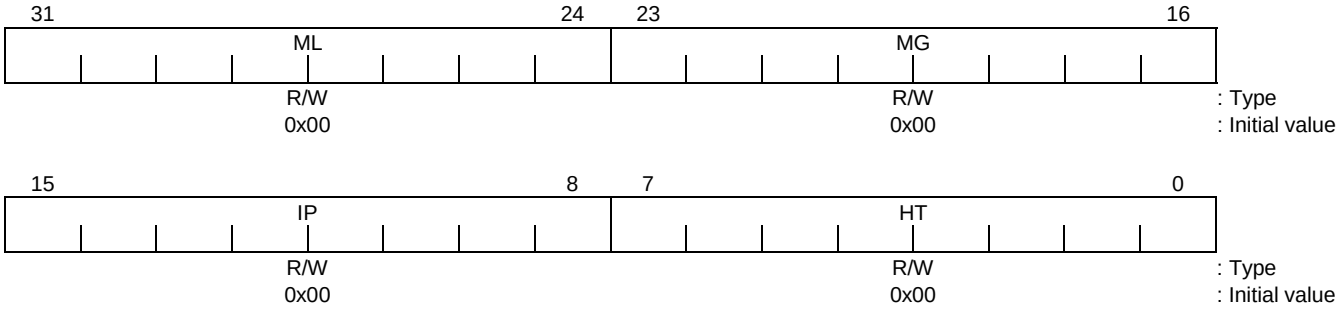
10.4.59 Configuration Data 2 Register (PCICDATA2) 0xD1E8



Bits	Mnemonic	Field Name	Description
31:16	SSID	Sub System ID	Subsystem ID (Initial value: 0x0000, R/W) This is the data loaded in the Sub System ID Register of the PCI Configuration space.
15:0	SSVID	Sub System Vendor ID	Subsystem Vendor ID (Initial value: 0x0000, R/W) This is the data loaded in the Sub System Vendor ID Register of the PCI Configuration space.

Figure 10.4.59 Configuration Data 2 Register

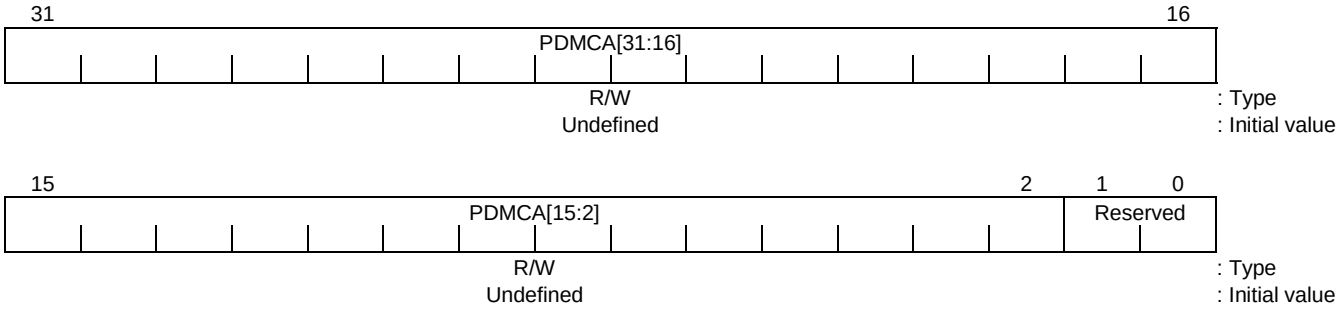
10.4.60 Configuration Data 3 Register (PCICDATA3) 0xD1EC



Bits	Mnemonic	Field Name	Description
31:24	ML	Maximum Latency	Max_Lat (Maximum Latency) (Initial value: 0x00, R/W) This is the data loaded in the Max_Lat Register of the PCI Configuration Space.
23:16	MG	Minimum Grant	Min_Gnt (Minimum Grant) (Initial value: 0x00, R/W) This is the data loaded in the Min_Gnt Register of the PCI Configuration Space.
15:8	IP	Interrupt Pin	Interrupt Pin (Initial value: 0x00, R/W) This is the data loaded in the Interrupt Pin Register of the PCI Configuration Space.
7:0	HT	Header Type	Header Type (Initial value: 0x00, R/W) This is the data loaded in the Header Type Register of the PCI Configuration Space.

Figure 10.4.60 Configuration Data 3 Register

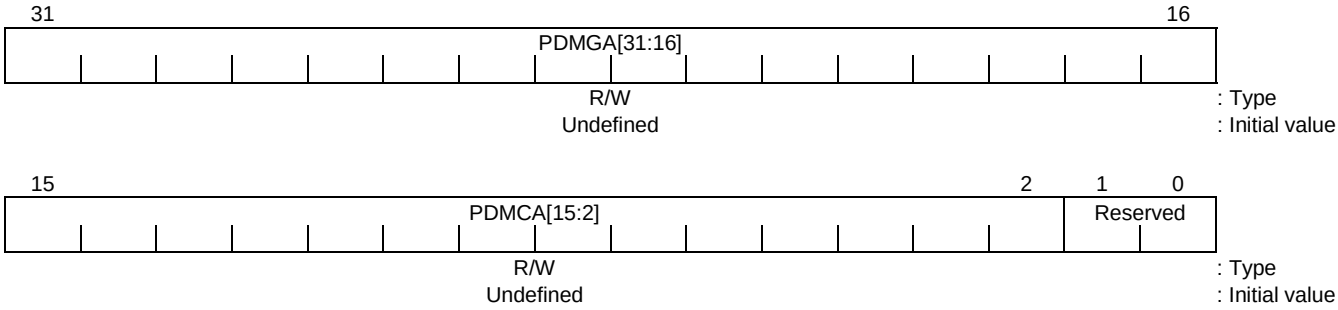
10.4.61 PDMAC Chain Address Register (PDMCA) 0xD200



Bits	Mnemonic	Field Name	Description
31:2	PDMCA	Chain Address	PDMAC Chain Address (Initial value: undefined, R/W) The address of the next PDMAC Data Command Descriptor to be read is specified by a G-Bus physical address on a 32-bit address boundary. This register value is held without being affected by a Reset. DMA transfer is automatically initiated if a non-zero value is written to this register.
1:0	—	Reserved	—

Figure 10.4.61 PDMAC Chain Address Register

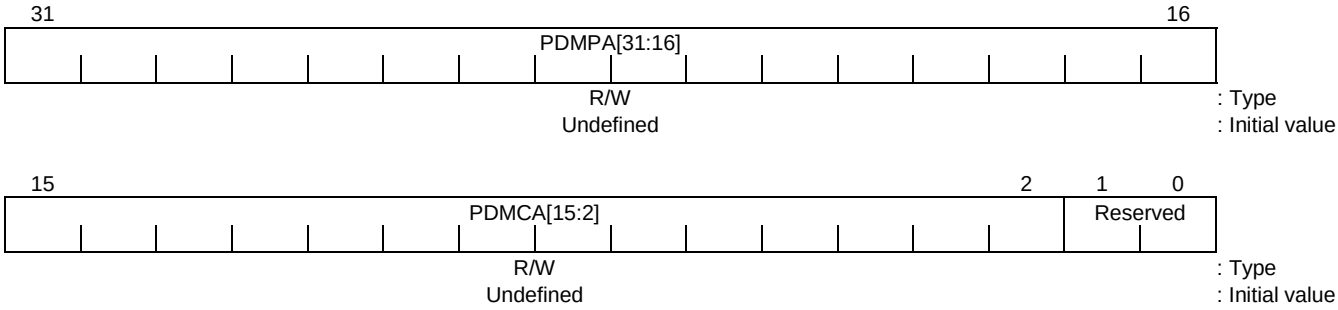
10.4.62 PDMAC G-Bus Address Register (PDMGA) 0xD204



Bits	Mnemonic	Field Name	Description
31:2	PDMGA	G-Bus Address	PDMAC G-Bus Address (Initial value: undefined, R/W) The G-Bus DMA transfer address is specified by a G-Bus physical address on a 32-bit address boundary. This register value is used for G-Bus Read access during DMA transfer from the G-Bus to the PCI Bus, or it is used for G-Bus Write access during DMA transfer from the PCI Bus to the G-Bus. This register value is held without being affected by a Reset.
1:0	—	Reserved	—

Figure 10.4.62 PDMAC G-Bus Address Register

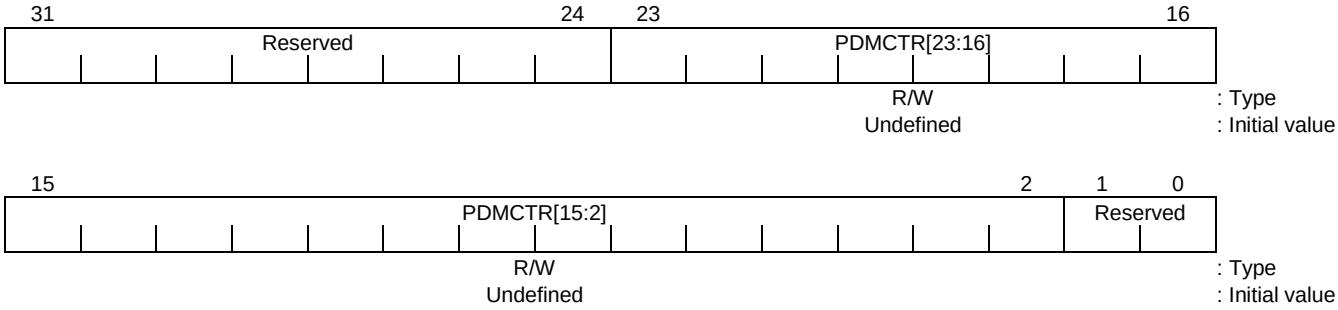
10.4.63 PDMAC PCI Bus Address Register (PDMPA) 0xD208



Bits	Mnemonic	Field Name	Description
31:2	PDMPA	PCI Bus Address	PDMAC PCI-Bus Address (Initial value: undefined, R/W) The PCI Bus DMA transfer address is specified by a PCI Bus physical address on a 32-bit address boundary. This register value is held without being affected by a Reset. Note: This register value is used for PCI Bus Write access during DMA transfer from the G-Bus to the PCI Bus, or it is used for PCI Bus Read access during DMA transfer from the PCI Bus to the G-Bus.
1:0	—	Reserved	—

Figure 10.4.63 PDMAC PCI Bus Address Register

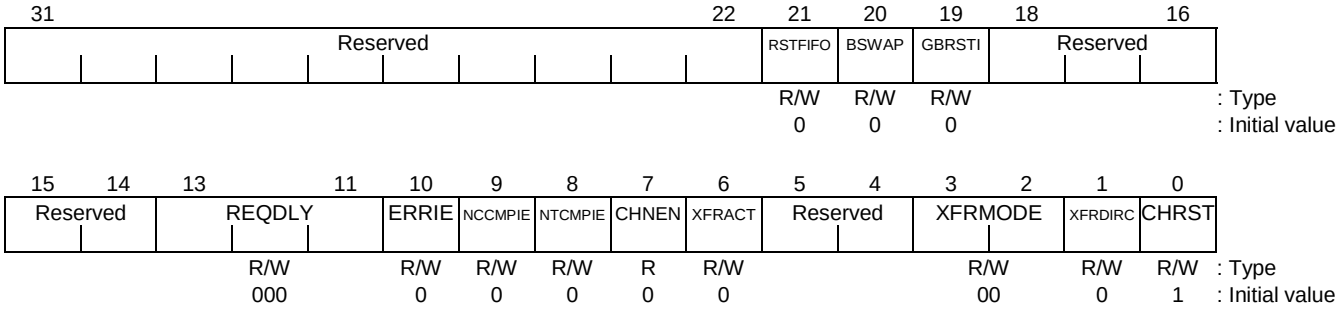
10.4.64 PDMAC Count Register (PDMCTR) 0xD20C



Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23:2	PDMCTR	Transfer Byte Count	PDMAC Transfer Count (Initial value: undefined, R/W) Sets an uncoded 24-bit transfer byte count in 32-bit word units. Also, the setting of this register must always be a multiple of the transfer size specified inside the PDMAC Control Register. No data transfer is performed if a count of “0” is set. This byte count value is calculated from the transferred byte size as the PDMAC performs a DMA transfer. This register value is held without being affected by a Reset.
1:0	—	Reserved	—

Figure 10.4.64 PDMAC Count Register

10.4.65 PDMAC Configuration Register (PDMCFG) 0xD210



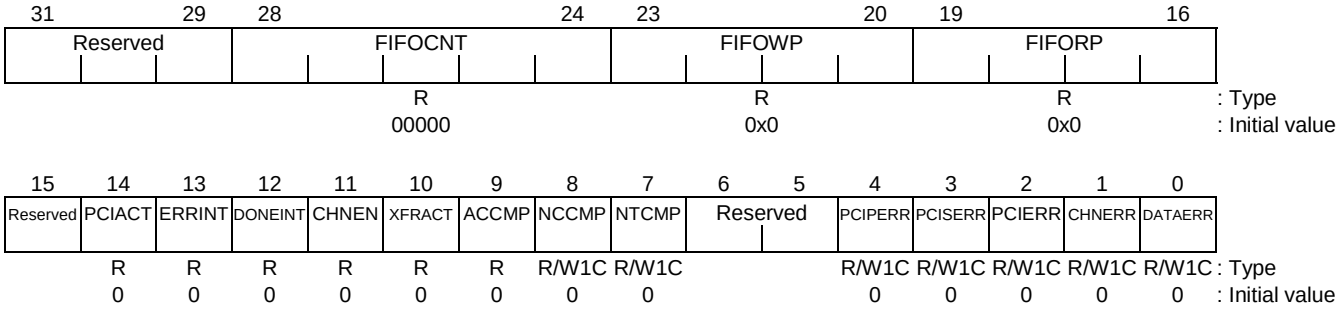
Bits	Mnemonic	Field Name	Description
31:22	—	Reserved	—
21	RSTFIFO	Reset FIFO	Reset FIFO (Initial value: 0, R/W) Initializes the Read pointer and Write pointer to the FIFO in the PDMAC, and sets the FIFO hold count to “0”. Please use the software to clear this bit when it is set. 1: Performs FIFO reset. 0: Does not perform FIFO reset.
20	BSWAP	Byte Swap Within DWORD	Swap Bytes in DWORD (Initial value: 0, R/W) Specifies whether to perform 32-bit data byte swapping. Please leave this bit at “0” for normal usage. Setting this bit when in the Big Endian mode executes data transfer so the byte order of the 32-bit data on the PCI Bus (which is Little Endian) does not change. 1: Transfer without swapping the byte order of each 32-bit DWORD data. 0: Swap the byte order of each 32-bit DWORD data, then transfer.
19	GBRSTI	G-Bus Burst Inhibit	G-Bus Burst Inhibit (Initial value: 0, R/W) 1: Do not use burst operations on G-Bus even if the burst mode field specifies burst. This allows devices that can’t burst to transfer data using bursts on the PCI bus. This bit applies only to G-Bus data transfers; it has no affect on chain operations. 0: The MDA will use burst operations for G-Bus data transfers when programmed to do so and if alignment, count, etc. permit.
18:14	—	Reserved	—
13:11	REQDLY	Request Delay Time	Request Delay (Initial value: 0x0, R/W) G-Bus transactions for DMA transfer must be performed separated at least by the interval this field specifies. 000: Continuously try to perform G-Bus transfer. 001: 16 G-Bus clocks 010: 32 G-Bus clocks 011: 64 G-Bus clocks 100: 128 G-Bus clocks 101: 256 G-Bus clocks 110: 512 G-Bus clocks 111: 1024 G-Bus clocks
10	ERRIE	Error Detect Interrupt Enable	Interrupt Enable on Error (Initial value: 0, R/W) 1: PDMAC generates an error during error detection. 0: PDMAC does not generate an error during error detection.
9	NCCMPIE	Normal Chain Complete Interrupt Enable	Interrupt Enable on Chain Done (Initial value: 0, R/W) 1: PDMAC generates an interrupt when the current chain is complete. 0: PDMAC does not generate an interrupt when the current chain is complete.
8	NTCMPIE	Normal Data Transfer Complete Interrupt Enable	Interrupt Enable on Transfer Done (Initial value: 0, R/W) 1: PDMAC generates an interrupt when the current data transfer is complete. 0: PDMAC does not generate an interrupt when the current data transfer is complete.

Figure 10.4.65 PDMAC Configuration Register (1/2)

Bits	Mnemonic	Field Name	Description
7	CHNEN	Chain Enable	Chain Enable (Initial value: 0) (Read Only) When the current data transfer is complete, this field reads the next data command Descriptor from the address indicated by the PDMAC Chain Address Register then indicates whether to continue the transfer or not. This bit is only set to “1” when either a CPU Write process or a Descriptor Read process sets a value other than “0” in the PDMAC Chain Address Register. This bit is cleared to “0” if either the Channel Reset bit is set, or “0” is set in the PDMAC Chain Address Register by a CPU Write or a Descriptor Read process. 1: Reads the next data command Descriptor. 0: Does not read the next data command Descriptor.
6	XFRACT	Transfer Active	Transfer Active (Initial value: 0, R/W) Specifies whether to perform DMA transfer or not. Setting this bit after setting the appropriate value in the register group initiates DMA data transfer. This bit is not set if the PDMAC Count Register value is “0” and the Chain Enable bit is cleared when “1” is written to this bit. Even when a value other than “0” is written to the Chain Address Register, “1” is set to this bit and DMA transfer automatically starts. Data transfer will be stopped after a short delay if this bit is cleared while the data transfer is in progress. This bit is automatically cleared to “0” either when data transfer ends normally or is stopped by an error. 1: Perform data transfer. 0: Do not perform data transfer.
5:4	—	Reserved	—
3:2	XFRMODE	Transfer Mode	Transfer Size (Initial value: 0, R/W) Specifies the transfer type (see detailed description in 10.3.9.4). 00: 1 DWORD (32-bit) with no overlap of G-Bus and PCI operation. 01: up to 16 DWORDs with no overlap of G-Bus and PCI operation. 10: Reserved 11: Reserved
1	XFRDIRC	Transfer Direction	Transfer Direction (Initial value: 0, R/W) Specifies the DMA data transfer direction. 1: Transfers data from the G-Bus to the PCI Bus. 0: Transfers data from the PCI Bus to the G-Bus.
0	CHRST	Channel Reset	Channel Reset (Initial value: 1, R/W) Resets the DMA channel. This bit must be cleared by the software in advance so the channel can start the data transfer. 1: All logic and State Machines are reset. 0: The channel becomes valid.

Figure 10.4.65 PDMAC Configuration Register (2/2)

10.4.66 PDMAC Status Register (PDMSTATUS) 0xD214



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:24	FIFOCNT	FIFO Hold Count	FIFO Valid Entry Count (Initial value: 00000, R) This field indicates the number of DWORDs written to the FIFO but not yet read. This is a diagnostic function.
23:20	FIFOWP	FIFO Write Pointer	FIFO Write Pointer (Initial value: 0x0, R) This field indicates the next Write position in the FIFO. This is a diagnostic function.
19:16	FIFORP	FIFO Read Pointer	FIFO Read Pointer (Initial value: 0x0, R) This field indicates the next Read position in the FIFO. This is a dianostic function.
15	—	Reserved	—
14	PCIACT	PCI Active	PCI Active (Initial value: 0, R) 1: The PDMAC is requesting to transfer data to or from the PCI bus of it is currently transferring data. 0: There is no active request or transfer from the PDMAC for the PCI bus.
13	ERRINT	Error Interrupt Status	Error Interrupt Status (Initial value: 0, R) Indicates whether to signal an error interrupt. 1: An error interrupt request exists. 0: No error interrupt request exists.
12	DONEINT	Normal Transfer Complete Interrupt Status	Normal Transfer Complete Interrupt Status (Initial value: 0, R) Indicates whether a Normal Transfer Complete Interrupt is signaled. This bit becomes “1” when either the Normal Chain Complete bit (NCCMP) is set and the Normal Chain Complete Interrupt Enable bit (NCCMPIE) is set, or when the Normal Transfer Complete bit (NCCMP) is set and the Normal Transfer Complete Interrupt Enable bit (NDCMPIE) is set. 1: A Normal Transfer Complete Interrupt request exists. 0: No Normal Transfer Complete Interrupt request exists.
11	CHNEN	Chain Enable	Chain Enable (Initial value: 0, R) This bit is a copy of the Chain Enable bit in the PDMAC Control Register.
10	XFRACT	Transfer Active	Transfer Active (Initial value: 0, R) This bit is a copy of the Transfer Active bit in the PDMAC Control Register.
9	ACCMP	Abnormal Chain Completion	Abnormal Chain Complete (Initial value: 0, R) 1: Indicates that the Chain transfer ended in an error state. In other words, this reflects an OR operation of the PDMAC Status Register bits [4:0]. 0: Indicates that no error has occurred in the Chain transfer since the previous error bit was cleared. Note: Bits [4:0] of the PDMAC Status Register must be cleared in order to clear this bit.
8	NCCMP	Normal Chain Completion	Normal Chain Complete (Initial value: 0, R/W1C) 1: Indicates that the Chain transfer ended in the Normal state. 0: Indicates that Chain transfer has not ended since this bit was previously cleared.
7	NTCMP	Normal Data Transfer Complete	Normal Data Transfer Complete (Initial value: 0, R/W1C) 1: Indicates that the data transfer specified by the PDMAC Register ended in the Normal state. 0: Indicates that data transfer has not ended since this bit was previously cleared.

Figure 10.4.66 PDMAC Status Register (1/2)

Bits	Mnemonic	Field Name	Description
6:5	—	Reserved	—
4	PCIPERR	PCI Parity Error	PCI Parity Error (Initial value: 0, R/W1C) 1: Indicates that there was a parity error on a PCI transaction performed on behalf of the PDMAC. 0: Indicates that no parity error has been detected on PDMAC PCI transfer since this bit was cleared.
3	PCISERR	PCI System Error	PCI System Error (Initial value: 0, R/W1C) 1: Indicates that a PCI bus module asserted SERR* during a PCI operation initiated by the PDMAC. 0: Indicates that SERR* has not been asserted during a PDMAC-initiated PCI cycle since this bit was last cleared.
2	PCIERR	PCI Fatal Error	PCI Fatal Error (Initial value: 0, R/W1C) 1: Indicates that an error was signaled on the PCI Bus during a PCI operation initiated by the PDMAC. 0: Indicates that no error has been signaled on the PCI Bus since this bit was previously cleared.
1	CHNERR	G-Bus Chain Error	G-Bus Chain Bus Error (Initial value: 0, R/W1C) 1: Indicates that a G-Bus error occurred during the Chain process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the Chain process since this bit was cleared.
0	DATAERR	G-Bus Data Error	G-Bus Data Bus Error (Initial value: 0, R/W1C) 1: Indicates that a G-Bus error occurred during the data transfer process. DMA transfer stops. 0: Indicates that no G-Bus error has occurred during the data transfer process since this bit was cleared.

Figure 10.4.66 PDMAC Status Register (2/2)

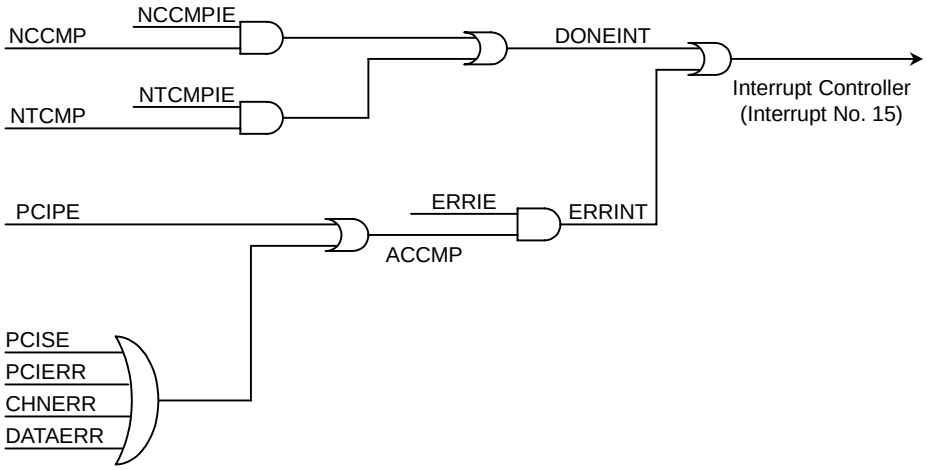


Figure 10.4.67 PDMAC Interrupt Signaling

10.5 PCI Configuration Space Register

The PCI Configuration Space Register is accessed using PCI Configuration cycles by way of an external PCI host device only when in the Satellite mode. Table 10.5.1 lists registers contained within the PCI Configuration Space Register. The registers in the table with a shaded background are those whose values can be rewritten by software. (See 10.3.13.)

Registers at addresses 0x00 through 0x41 can use the corresponding PCI Controller Control Register to access from the TX49/H2 core when in the Host mode. Please refer to the explanation of the corresponding PCI Controller Control registers for more information about these registers. This section only describes the registers that are accessed from the PCI Configuration Space.

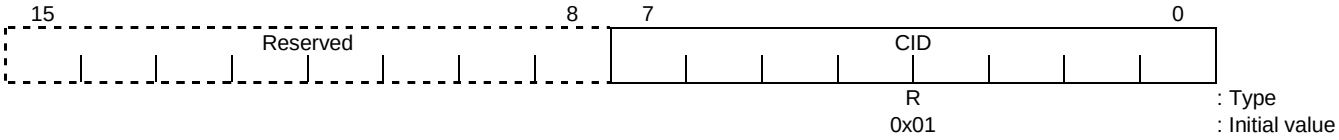
Also, it is possible to read some of the fields in the Status Register and PMCSR register from the Satellite Mode PCI Status Register.

Please refer to the PCI Bus Specifications for more information on the PCI Configuration Register.

Table 10.5.1 PCI Configuration Space Register

Address	31 16		15 0		Corresponding Register
00h	Device ID		Vendor ID		PCIID
04h	Status		Command		PCISTATUS
08h	Class Code			Revision ID	PCICCREV
0Ch	BIST	Header Type	Latency Timer	Cache Line Size	PCICFG1
10h	Memory Space 0 Base Address				P2GM0PBASE
14h	Memory Space 1 Base Address				P2GM1PBASE
18h	Memory Space 2 Base Address				P2GM2PBASE
1Ch	I/O Space Base Address				P2GIOPBASE
20h	Reserved				—
24h	Reserved				—
28h	Reserved				—
2Ch	Subsystem ID		Subsystem Vendor ID		PCISID
30h	Reserved				—
34h	Reserved			Capabilities Pointer (Cap_Ptr)	PCICAPPTR
38h	Reserved				—
3Ch	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	PCICFG2
40h	Reserved		Retry Timeout Value	TRDY Timeout Value	G2PTOCNT
44h-DBh	Reserved				—
DCh	Power Management Capabilities (PMC)		Next Item Ptr (Next_Item_Ptr)	Capability ID (Cap_ID)	—
E0h	Reserved	Reserved	Power Management Control/Status Register (PMCSR)		—
E4h-FFh	Reserved				—

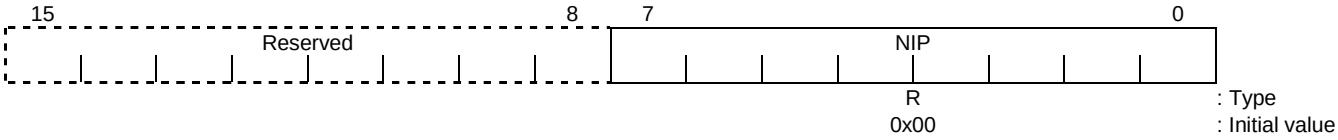
10.5.1 Capability ID Register (Cap_ID) 0xDC



Bits	Mnemonic	Field Name	Description
15:8	—	Reserved	—
7:0	CID	Capability ID	Capability ID (Initial value: 0x01, R) Indicates that a list is the link list of the Power Management Register.

Figure 10.5.1 Capability ID Register

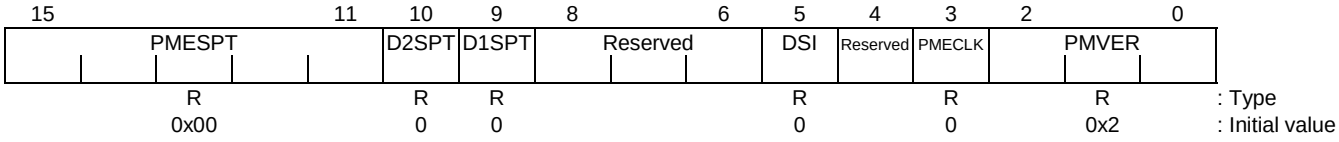
10.5.2 Next Item Pointer Register (Next_Item_Ptr) 0xDD



Bits	Mnemonic	Field Name	Description
15:8	—	Reserved	—
7:0	NIP	Next Item Pointer	Next Item Pointer (Initial value: 0x00, R) This is the Next Item pointer. Indicates the end of a list.

Figure 10.5.2 Next Item Pointer Register

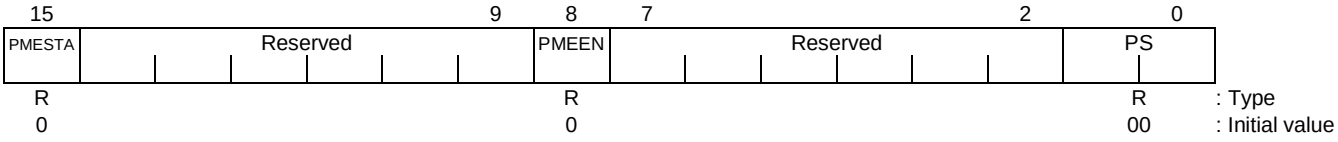
10.5.3 Power Management Capability Register (PMC) 0xDE



Bits	Mnemonic	Field Name	Description
15:11	PMESPT	PME Output Support	PME_ Support (Fixed value: 0x00, R) On TX4925 the function is not supported.
10	D2SPT	D2 Support	D2_Support (Fixed value: 0, R) 0: Indicates that the D2 state is not supported.
9	D1SPT	D1 Support	D1_Support (Fixed value: 0, R) 0: Indicates that the D1 state is not supported.
8:6	—	Reserved	—
5	DSI	DSI	DSI (Fixed value: 0, R) 1: Indicates that Device Specific Initialization is required.
4	—	Reserved	—
3	PMECLK	PME Clock	PME Clock (Fixed value: 0, R) 0: Indicates that the PCI Clock is not required to assert the PME* signal.
2:0	PMVER	Power Management I/F Version	Version (Fixed value: 0x2, R) 2: Indicates compliance with “PCI Power Management Interface Specification” Version 1.1.

Figure 10.5.3 Power Management Capability Register

10.5.4 Power Management Control/Status Register (PMCSR) 0xE0



Bits	Mnemonic	Field Name	Description
15	PMESTA	PME Status	PME_Status (Initial value: 0, R) On TX4925 the function is not supported.
14:9	—	Reserved	—
8	PMEEN	PME Enable	PME_En (Initial value: 0, R) On TX4925 the function is not supported.
7:2	—	Reserved	—
1:0	PS	Power State	PowerState (Initial value: 00, R) Sets the Power Management state. The Power Management State Change bit (P2GSTATUS.PMSC) of the P2G Status Register is set when the value of this field is changed. It also becomes possible to generate a Power State Change Interrupt at this time. The TX4925 can read the value of this field from the PowerState field (PCISSTATUS.PS) of the Satellite Mode PCI Status Register. 00b: D0 (no change) 01b: D1 :Reserved 10b: D2 :Reserved 11b: D3hot

Figure 10.5.4 Power Management Control/Status Register

11. Serial I/O Port

11.1 Features

The TX4925 asynchronous Serial I/O (SIO) interface has two full duplex UART channels (SIO0 and SIO1). SIO has the following features.

- Full duplex transmission (simultaneous transmission and reception)
- On-chip baud rate generator
- Modem flow control (CTS/RTS)
- FIFO
  - Transmit FIFO: 8 bits × 8 stages
  - Reception FIFO: 13 bits × 16 stages (data: 8 bits, status: 5 bits)
- Supports DMA transfer
- Supports multi-controller systems
  - Supports Master/Slave operation

11.2 Block Diagram

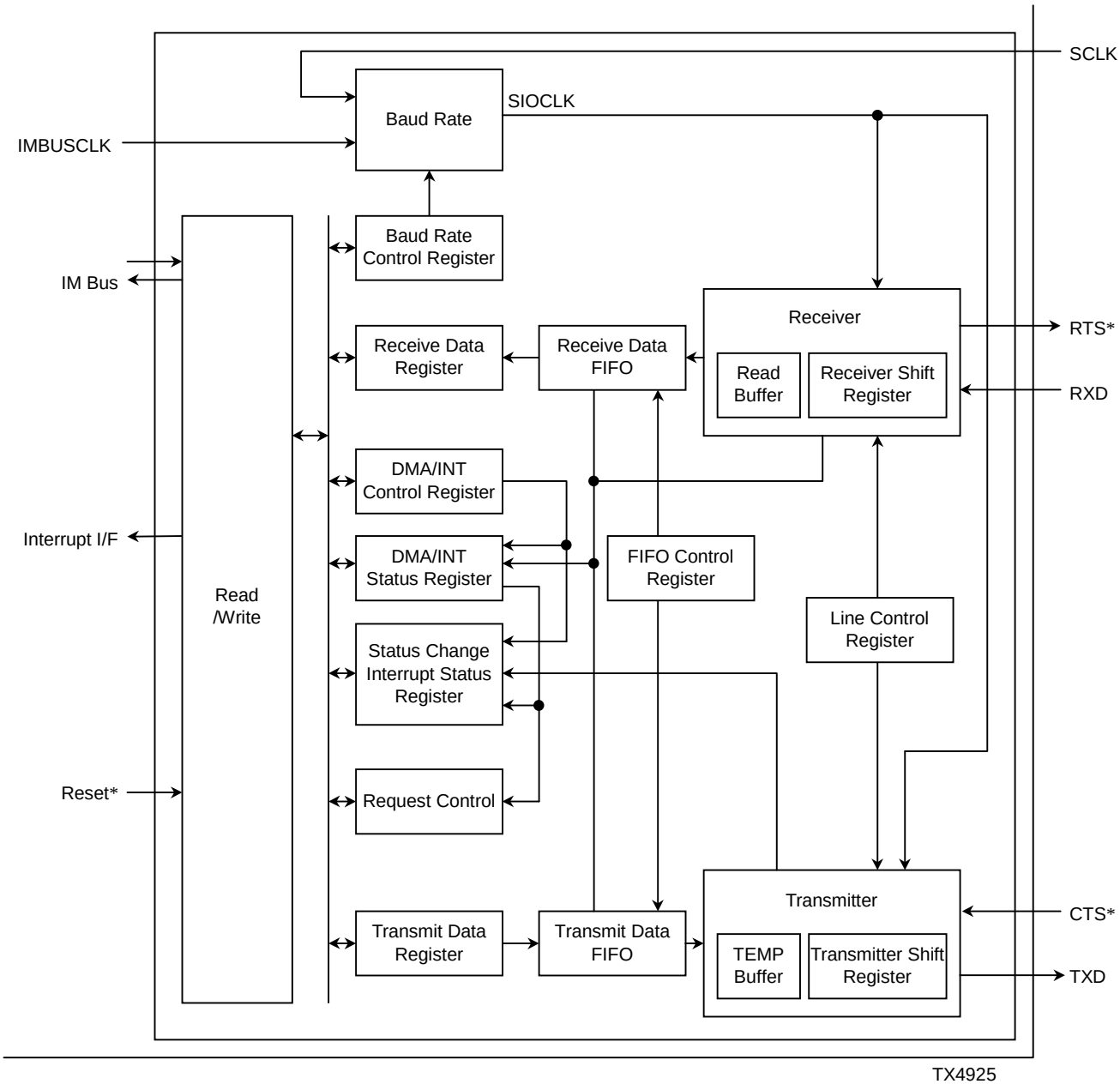


Figure 11.2.1 SIO Internal Block Diagram

11.3 Detailed Explanation

11.3.1 Overview

During reception, serial data that are input as an RXD signal from an external source are converted into parallel data, then are stored in the Receive FIFO buffer. Parallel data stored in the FIFO buffer are fetched by either CPU or DMA transfer.

During transmission, parallel data written to the Transmit FIFO buffer by CPU or DMA transfer are converted into serial data, then are output as a TXD signal.

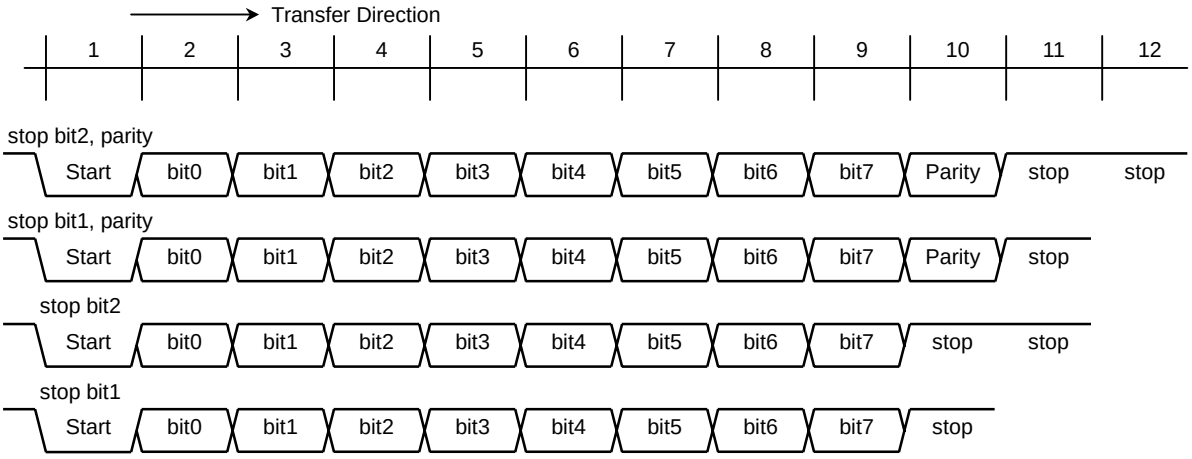
11.3.2 Data Format

The TX4925 SIO can use the following data formats.

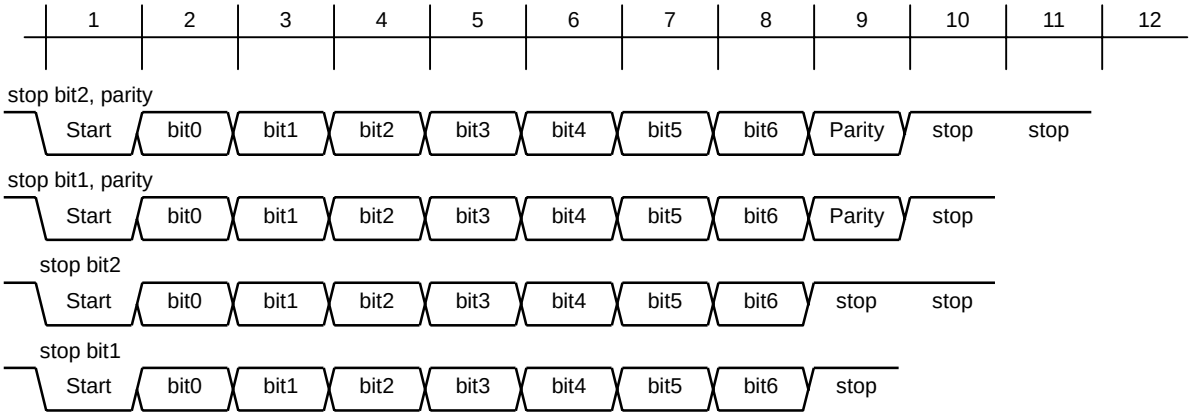
Data Length	: 8/7 bits
Stop Bit	: 1/2 bits
Parity Bit	: Yes/No
Parity Format	: Even/Odd
Start Bit	: Fixed to 1 bit

Figure 11.3.1 illustrates the data frame when making each setting.

8-bit Data

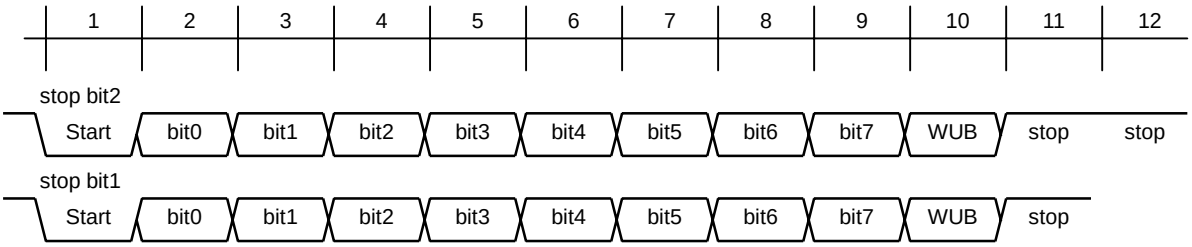


7-bit Data



8-bit Data Multi-Control System

WUB = Wake Up bit
1: Address (ID) Frame
0: Data Frame



7-bit Data Multi-Control System

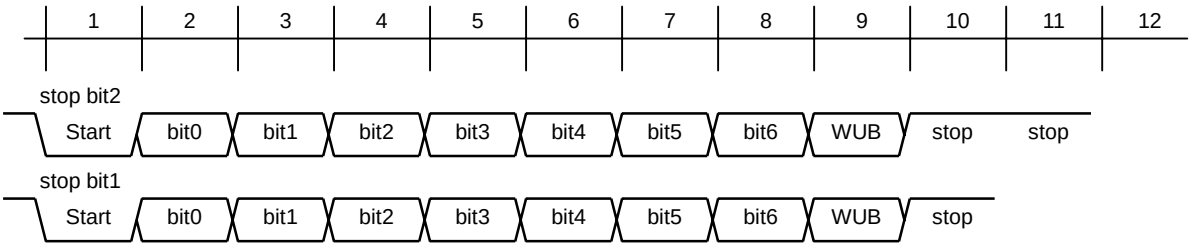


Figure 11.3.1 Data Frame Configuration

11.3.3 Serial Clock Generator

Generates the Serial Clock (SIOCLK). SIOCLK determines the serial transfer rate and has a frequency that is 16× the baud rate. One of the following can be selected as the source for the Serial Clock (SIOCLK).

- Internal System Clock (IMBUSCLKF)
- External Clock Input (SCLK)
- Baud rate generator circuit output

The IMBUSCLKF frequency can be selected from frequencies that are 1/5 the frequency of the CPU clock. The maximum frequency tolerance of the external clock input (SCLK) is 45% the frequency of IMBUSCLKF. For example, if IMBUSCLKF = 40 MHz, then set SCLK to 18 MHz or less.

The baud rate generator is a circuit that divides these clock signals according to the following formula.

$$\text{Baud Rate} = \frac{fc}{\text{Prescalar} \times \text{Divisor} \times 16}$$

- fc: Clock frequency of IMBUSCLKF or an external clock input (SCLK)
- Prescalar Value: 2, 8, 32, 128
- Divide Value: 1, 2, 3,…255

Table 11.3.1 shows example settings of divide values relative to representative baud rates.

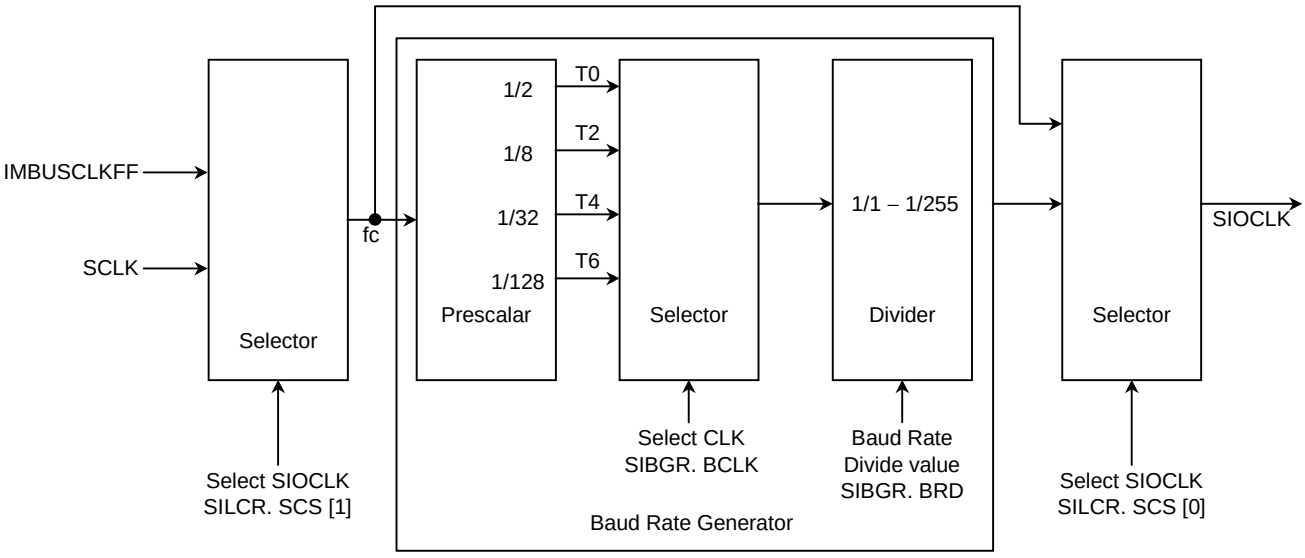


Figure 11.3.2 Baud Rate Generator and SIOCLK Generator

It is possible to correctly receive data if the error of the baud rate set by this controller is within 3.12 % of the target baud rate (communication baud rate).

Table 11.3.1 Example Divide Value Settings (and error [%] from target baud rate value)

fc [MHz]		kbps	Prescalar Value (SIBGR.BLCK) and Divide Value (SIBGR.BRD)			
			2	8	32	128
IMBUS CLKF	40	0.11				178 (–0.25 %)
		0.15				130 (–0.16 %)
		0.30				65 (0.16 %)
		0.60			130 (0.16 %)	33 (–1.36 %)
		1.20			65 (0.16 %)	16 (1.73 %)
		2.40		130 (–0.16 %)	33 (–1.36 %)	
		4.80		65 (0.16 %)	16 (1.73 %)	
		9.60	130 (0.16 %)	33 (–1.36 %)		
		14.40	87 (–0.22 %)	22 (–1.36 %)		
		19.20	65 (0.16 %)	16 (1.73 %)		
		28.80	43 (0.94 %)	11 (–1.36 %)		
		38.40	33 (–1.36 %)			
		57.60	22 (–1.36 %)			
		76.80	16 (1.73 %)			
		115.20	11 (–1.36 %)			
	36.864	0.11				164 (–0.22%)
		0.15				120 (0.00%)
		0.30				60 (0.00%)
		0.60			120 (0.00%)	30 (0.00%)
		1.20			60 (0.00%)	15 (0.00%)
		2.40		120 (0.00%)	30 (0.00%)	
		4.80		80 (0.00%)	15 (0.00%)	
		9.60	120 (0.00%)	60 (0.00%)		
		14.40	80 (0.00%)	40 (0.00%)	5 (0.00%)	
		19.20	60 (0.00%)	30 (0.00%)		
		28.80	40 (0.00%)	20 (0.00%)		
		38.40	30 (0.00%)			
		57.60	20 (0.00%)	5 (0.00%)		
		76.80	15 (0.00%)			
		115.20	10 (0.00%)			
SCLK	7.3728	0.11			131 (–0.07 %)	33 (–0.83 %)
		0.15			96 (0.00 %)	24 (0.00 %)
		0.30			48 (0.00 %)	12 (0.00 %)
		0.60		96 (0.00 %)	24 (0.00 %)	6 (0.00 %)
		1.20		48 (0.00 %)	12 (0.00 %)	3 (0.00 %)
		2.40	96 (0.00 %)	24 (0.00 %)	6 (0.00 %)	
		4.80	48 (0.00 %)	12 (0.00 %)	3 (0.00 %)	
		9.60	24 (0.00 %)	6 (0.00 %)		
		14.40	16 (0.00 %)	4 (0.00 %)	1 (0.00 %)	
		19.20	12 (0.00 %)	3 (0.00 %)		
		28.80	8 (0.00 %)	2 (0.00 %)		
		38.40	6 (0.00 %)			
		57.60	4 (0.00 %)	1 (0.00 %)		
		76.80	3 (0.00 %)			
		115.20	2 (0.00 %)			

11.3.4 Data Reception

When the Serial Data Reception Disable bit (RSDE) of the Flow Control Register (SIFLCRn) is set to “0”, reception operation starts after the RXD signal start bit is detected. Start bits are detected when the RXD signal transitions from the High state to the Low state. Therefore, the RXD signal is not interpreted as a start bit if it is Low when the Serial Data Reception Disable bit is set to “0”.

The received data are stored in the Receive FIFO. The Reception Data Full bit (RDIS) of the DMA/Interrupt Status Register (SIDISRn) is set if the byte count of the stored reception data exceeds the value set by the Receive FIFO Request Trigger Level field (RDIL) of the FIFO Control Register (SIFCRn).

An interrupt is signaled when the Reception Data Interrupt Enable bit (RIE) of the DMA/Interrupt Control Register (SIDICRn) is set. The received data can be read from the Receive FIFO Data Register (SIRFIFOn).

In addition, DMA transfer is initiated when the Reception Data DMA Enable bit (RDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

11.3.5 Data Transmission

Data stored in the Transmission Data FIFO are transmitted when the Serial Data Transmission Disable bit (TSDE) of the Flow Control Register (SIFLCRn) is set to “0”.

If the available space in the Transmit FIFO is greater than the byte count set by the Transmit FIFO Request Trigger Level (TDIL) of the Control Register (SIFCRn), the transmission data empty bit (TDIS) of the DMA/Interrupt Status Register (SIDISRn) is set.

An interrupt is signaled when the Transmission Data Interrupt Enable bit (TIE) of the DMA/Interrupt Control Register (SIDICRn) is set.

In addition, DMA transfer is initiated when the Transmission Data DMA Enable bit (TDE) of the DMA/Interrupt Control Register (SIDICRn) is set.

11.3.6 DMA Transfer

The DMA Request Control Register (DRQCTR)of the DMA Request Select field (DMAREQ[3:0]) can be used to allocate DMA channels for each reception and transmission channel in the following manner.

SIO Channel 1 Reception	DMA Channel 0
SIO Channel 1 Transmission	DMA Channel 1
SIO Channel 0 Reception	DMA Channel 2
SIO Channel 0 Transmission	DMA Channel 3

Set the DMA Channel Control Register of the DMA Controller as described below.

DMA Request Polarity	Low Active	DMCCRn.ACKPOL = 0
DMA Acknowledge Polarity	Low Active	DMCCRn.REQPOL = 0
Request Detection	Level Detection	DMCCRn.EGREQ = 0
Transfer Size	1 Byte	DMCCRn.XFSZ = 000b
Transfer Address Mode	Dual	DMCCRn.SNGAD = 0

In the case of transmission channels, the address of the Transmit FIFO Register (SITFIFOn) is set in the DMAC Destination Address Register (DMDARn). In the case of reception channels, the address of the Receive FIFO Register (SIRFIFOn) is set in the DMAC Source Address Register (DMSARn). Please set the addresses specified in "11.4.8 Transmit FIFO Register" and "11.4.9 Receive FIFO Register" since the set address differs depending on the Endian mode.

11.3.7 Flow Control

SIO supports hardware flow control that uses the RTS*/CTS* signal.

The CTS* (Clear to Send) input signal indicates that data can be received from the reception side when it is Low. Setting the Transmission Enable Select bit (TES) of the Flow Control Register (SIFLCRn) makes transmission flow control that uses the CTS* signal more effective.

It is also possible to generate status change interrupts by changing the state of the CTS* signal. The conditions in which interrupts are generated can be selected by the CTSS Active Condition field of the DMA/Interrupt Control Register (SIDICRn).

Setting the RTS* (Request to Send) output signal to High requests the transmission side to pause transmission. Transmission resumes when the reception side becomes ready and the RTS* signal is set to Low.

Setting the Reception Enable Select bit (RCS) of the flow Control Register (SIFLCRn) makes reception flow control that uses the RTS* signal more effective. The RTS* signal pin status becomes High when data of the byte count set by the RTS Active Trigger Level field (RTSTL) of the Flow Control Register (SIFLCRn) accumulates in the Receive FIFO. The RTS* signal can also be made High by setting the RTS Software Control bit (RTSSC) of the Flow Control Register (SIFLCRn). Setting this bit requests the transmission side to pause transmission.

11.3.8 Reception Data Status

Status data such as the following is also stored in the Receive FIFO.

- Overrun error

  An overrun error is generated if all 16-stage Receive FIFO buffers become full and more data is transferred to the Reception Read buffer. When this occurs, the Overrun Status bit is set by the last stage of the Receive FIFO.

- Parity error

  A parity error is generated when a parity error is detected in the reception data.

- Framing error

  A framing error is generated when "0" is detected at the first stop bit of the reception data.

- Break reception

  A break is detected when a framing error occurs in the reception data and all data in a single frame are "0". When this occurs, 2 frames (2 Bytes) of 0x00 data are stored in the Receive FIFO.

The Reception Error Interrupt bit (SIDISR.ERI) of the DMA/Interrupt Status Register (SIDISRn) is set when one of the following errors is detected: an overrun error, a parity error, or a framing error. An interrupt is signaled if the Reception Error Interrupt Enable bit of the DMA/Interrupt Control Register (SIDICRn) is set.

The Receive Break bit (RBRKD) and the Receiving Break bit (RBRKD) of the Status Change Interrupt Status Register (SISCISR) is set when a break is detected. The Receive Break bit (RBRKD) remains set until it is cleared by the software. The Receiving Break bit (RBRKD) is automatically cleared when a frame is received that is not a break.

The status of the next reception data to be read is set to the Overrun Error bit (UOER), Parity Error bit (UPER), Framing Error bit (UFER), and the Receive Break bit (RBRKD). Each of these statuses is updated when reception data is read from the Receive FIFO Register (SIRFIFOn).

During DMA transfer, an error is signaled and DMA transfer stops with error data remaining in the Receive FIFO if either an error (Framing Error, Parity Error, or Overrun Error) or a Reception time out (TOUT) is detected. If a Reception Error occurs during DMA transfer, use the Receive FIFO Reset bit (RFRST) of the FIFO Control Register (SIFCRn) to clear the Receive FIFO. However, a software reset will be required if a reception overrun error has occurred. Refer to “11.3.10 Software Reset” for more information.

11.3.9 Reception Time Out

A Reception time out is detected and the Reception Time Out bit (TOUT) of the DMA/Interrupt Status Register (SIDISR) is set under the following conditions.

- Non-DMA transfer mode (SIDICRn.RDE = 0):

  When at least 1 Byte of reception data exists in the Receive FIFO and the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed

- DMA transfer mode (SIDICRn.RDE = 1):

  When the data reception time for the 2 frames (2 Bytes) after the last reception has elapsed regardless of whether reception data exists in the Receive FIFO

11.3.10 Software Reset

It is necessary to reset the FIFO and perform a software reset in the following situations.

(1) After transmission data is set in FIFO, etc., transmission started but stopped before its completion

(2) An overrun occurred during data reception

Software reset is performed by setting the Software Reset bit (SWRST) of the FIFO Control Register (SIFCR). This bit automatically returns to “0” after initialization is complete. This bit must be set again since all SIO registers are initialized by software resets.

11.3.11 Error Detection/Interrupt Signaling

An interrupt is signaled if an error or an interrupt cause is detected, the corresponding status bit is set and the corresponding Interrupt Enable bit is set.

Figure 11.3.3 shows the relationship between the status bit for each interrupt cause and each interrupt enable bit. Please refer to the explanation for each status bit for more information about each interrupt cause.

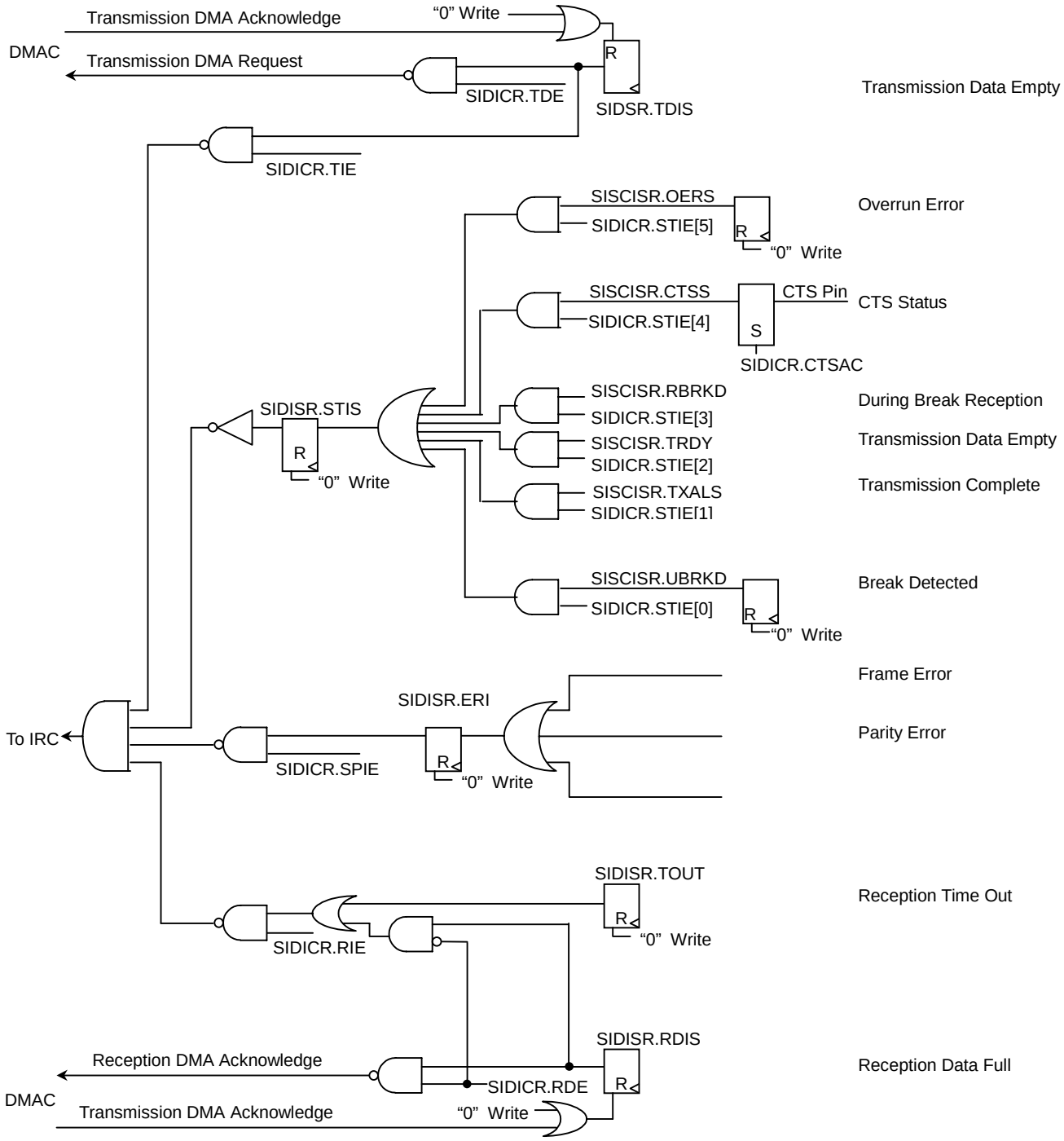


Figure 11.3.3 Relationship Between Interrupt Status Bits and Interrupt Signals

11.3.12 Multi-Controller System

The Multi-Controller System consists of one Master Controller, and multiple Slave Controllers as shown below in Figure 11.3.4.

In the case of the Multi-Controller System, the Master Controller transmits an address (ID) frame to all Slave Controllers, then transmits and receives data with the selected Slave Controller. Slave Controllers that were not selected will ignore this data.

Data frames whose data frame Wake Up bits (WUB) are “1” are handled as address (ID) frames. Data frames whose Wake Up bit (WUB) is “0” are handled as data frames.

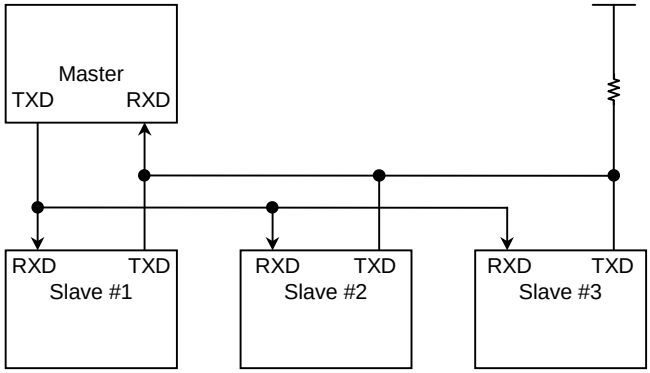


Figure 11.3.4 Example Configuration of Multi-Controller System

The data transfer procedure for the Multi-Controller System is as follows.

(1) The Master and Slave Controllers set the Mode field (UMODE) of the Line Control Register (SILCR) to “10” or “11” to set the Multi-Controller System mode. Also, the Slave Controller sets the open drain enable bit (UODE) of the Line Control Register (SILCR), setting the TXD output signal to open drain output.

(2) The Slave Controller sets the Reception Wake Up bit (RWUB) of the Line Control Register (SILCR), making it possible to receive address (ID) frames from the Master Controller.

(3) The Master Controller sets the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), and transmits the address (ID) of the selected Slave Controller. This causes the address (ID) frame to be transmitted. The Reception after Address Transmission Wake Up bit (RWUB) is cleared, enabling reception of data frames.

(4) Since the Reception Wake Up bit (RWUB) is set, the Slave Controller generates an interrupt to the CPU by receiving an address (ID) frame. The CPU compares its own address (ID) and the received data together. If they match, the Reception Wake Up bit (RWUB) is cleared, making data frame reception possible.

(5) The Master Controller and the selected Slave Controller clear the Transmission Wake Up bit (TWUB) of the Line Control Register (SILCR), then set the mode that transmits data frames.

(6) Transmit/Receive data between the Master Controller and the selected Slave Controller. Then, Slave Controllers that were not selected ignore data frames since the Reception Wake Up bit (RWUB) is still set.

11.4 Registers

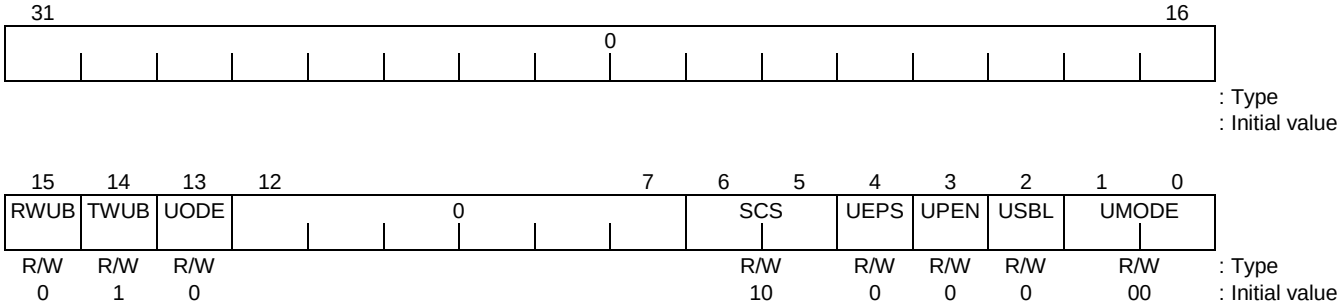
With the exception of DMA access to the Transmit FIFO Register or the Receive FIFO Register, please use Word access when accessing register in the Serial I/O Port.

Table 11.4.1 SIO Registers

Reference	Offset Address	Mnemonic	Register Name
SIO0 (Channel 0)			
11.4.1	0xF300	SILCR0	Line Control Register 0
11.4.2	0xF304	SIDICR0	DMA/Interrupt Control Register 0
11.4.3	0xF308	SIDISR0	DMA/Interrupt Status Register 0
11.4.4	0xF30C	SISCISR0	Status Change Interrupt Status Register 0
11.4.5	0xF310	SIFCR0	FIFO Control Register 0
11.4.6	0xF314	SIFLCR0	Flow Control Register 0
11.4.7	0xF318	SIBGR0	Baud Rate Control Register 0
11.4.8	0xF31C	SITFIFO0	Transmit FIFO Register 0
11.4.9	0xF320	SIRFIFO0	Receive FIFO Register 0
SIO1 (Channel 1)			
11.4.1	0xF400	SILCR1	Line Control Register 1
11.4.2	0xF404	SIDICR1	DMA/Interrupt Control Register 1
11.4.3	0xF408	SIDISR1	DMA/Interrupt Status Register 1
11.4.4	0xF40C	SISCISR1	Status Change Interrupt Status Register 1
11.4.5	0xF410	SIFCR1	FIFO Control Register 1
11.4.6	0xF414	SIFLCR1	Flow Control Register 1
11.4.7	0xF418	SIBGR1	Baud Rate Control Register 1
11.4.8	0xF41C	SITFIFO1	Transmit FIFO Register 1
11.4.9	0xF420	SIRFIFO1	Receive FIFO Register 1

11.4.1 Line Control Register 0 (SILCR0) 0xF300 (Ch. 0)
Line Control Register 1 (SILCR1) 0xF400 (Ch. 1)

These registers specify the format of asynchronous transmission/reception data.

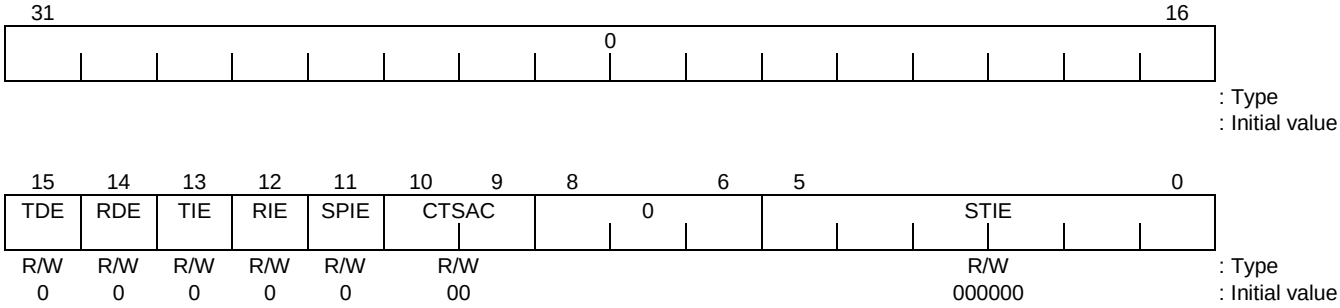


Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	RWUB	Receive Wake Up Bit	Wake Up Bit for Receive (Initial value: 0, R/W) When in the Multi-Controller System mode, this field selects whether to receive address (ID) frames whose Wake Up bits (WUB) are “1” or to receive data frames whose Wake Up bits (WUB) are “0”. This value is undefined when not in the Multi-Controller System mode. 0: Receive data frames. 1: Receive address (ID) frames.
14	TWUB	Transmit Wake Up Bit	Wake Up Bit for Transmit (Initial value: 1, R/W) When in the Multi-Controller System mode, this field specifies the Wake Up bit (WUB). This value is undefined when not in the Multi-Controller System mode. 0: Data frame transfer (WUB = 0) 1: Address (ID) frame transfer (WUB = 1)
13	UODE	Open Drain Enable	TXD Open Drain Enable (Initial value: 0, R/W) This field selects the output mode of the TXD signal. When in the Multi-Controller System mode, the Slave Controller must set the TXD signal to Open Drain. 0: Totem pole output 1: Open drain output
12:7	—	Reserved	—
6:5	SCS	Clock Select	SIO Clock Select (Initial value: 10, R/W) This field selects the serial transfer clock. The clock frequency that is the serial transfer clock divided by 16 becomes the baud rate (bps). 00: Internal clock (IMBUSCLKF) 01: Baud rate generator output that divided IMBUSCLKF 10: External clock (SCLK) 11: Baud rate generator output that divided SCLK
4	UEPS	Even Parity Select	UART Even Parity Select (Initial value: 0, R/W) This field selects the parity mode. 0: Odd parity 1: Even parity
3	UPEN	Parity Check Enable	UART Parity Enable (Initial value: 0, R/W) This field selects whether to perform the parity check. 0: Disable the parity check 1: Enable the parity check
2	USBL	Stop Bit Length	UART Stop Bit Length (Initial value: 0, R/W) This field specifies the stop bit length. 0: 1 bit 1: 2 bits
1:0	UMODE	Mode	UART Mode (Initial value: 00, R/W) This field sets the data frame mode. 00: 8-bit data length 01: 7-bit data length 10: Multi-Controller 8-bit data length 11: Multi-Controller 7-bit data length

Figure 11.4.1 Line Control Register

11.4.2 DMA/Interrupt Control Register 0 (SIDICR0) 0xF304 (Ch. 0)
DMA/Interrupt Control Register 1 (SIDICR1) 0xF404 (Ch. 1)

These registers use either DMA or interrupts to execute the Host Interface.



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	TDE	Transmit DMA Transfer Enable	Transmit DMA Enable (Initial value: 0, R/W) This field sets whether to use DMA in the method for writing transmission data to the Transmit FIFO. 0: Do not use DMA. 1: Use DMA.
14	RDE	Receive DMA Transfer Enable	Receive DMA Enable (Initial value: 0, R/W) This field sets whether to use DMA in the method for reading reception data from the Receive FIFO. 0: Do not use DMA. 1: Use DMA.
13	TIE	Transmit Data Empty Interrupt Enable	Transmit Data Empty Interrupt Enable (Initial value: 0, R/W) When there is open space in the Transmit FIFO, this field sets whether to signal an interrupt. Set “0” when in the DMA Transmit mode (TDE = 1). 0: Do not signal an interrupt when there is open space in the Transmit FIFO. 1: Signal an interrupt when there is open space in the Transmit FIFO.
12	RIE	Reception Data Full Interrupt Enable	Receive Data Full Interrupt Enable (Initial value: 0, R/W) This field sets whether to signal interrupts when reception data is full (SIDISRn.RDIS = 1) or a reception time out (SIDISRn.TOUT = 1) occurs. Set to “0” when in the DMA Receive mode (RDE = 1). 0: Do not signal interrupts when reception data is full/reception time out occurred. 1: Signal interrupts when reception data is full/reception time out occurred.
11	SPIE	Reception Error Interrupt Enable	Receive Data Error Interrupt Enable (Initial value: 0, R/W) This field sets whether to signal interrupts when a reception error (Frame Error, Parity Error, Overrun Error) occurs (SIDISR.ERI = 1). 0: Do not signal reception error interrupts. 1: Signal reception error interrupts.
10:9	CTSAC	CTSS Active Condition	CTSS Active Condition (Initial value: 00, R/W) This field specifies status change interrupt request conditions using the CTS Status (CTSS) of the Status Change Interrupt Status Register. 00: Do not detect CTS signal changes. 01: Rising edge of the CTS pin 10: Falling edge of the CTS pin 11: Both edges of the CTS pin
8:6	—	Reserved	—

Figure 11.4.2 DMA/Interrupt Control Register (1/2)

Bits	Mnemonic	Field Name	Description
5:0	STIE	Status Change Interrupt Enable	Status Change Interrupt Enable (Initial value: 000000, R/W) This field sets the set conditions of the Status Change bit (STIS) of the DMA/Interrupt Status Register (SIDISR). The condition is selected depending on which bit of the Status Change Interrupt Status Register (SISCISR) is set. (Multiple selections are possible.) An SIO interrupt is asserted when STIC is “1”. 000000: Do not detect status changes. 1*****: Set “1” to STIS when the Overrun bit (OERS) is “1”. *1****: Set “1” to STIS when a change occurs in a condition set by the CTSS Active Condition field (CTSAC) in the CTS Status bit (CTSS). **1***: Set “1” to STIS when the Break bit (RBRKD) becomes “1”. ***1**: Set “1” to STIS when the Transmit Data Empty bit (TRDY) becomes “1”. ****1*: Set “1” to STIS when the Transmission Complete bit (TXALS) becomes “1”. *****1: Set “1” to STIS when the Break Detection bit (UBRKD) becomes “1”.

Figure 11.4.2 DMA/Interrupt Control Register (2/2)

11.4.3 DMA/Interrupt Status Register 0 (SIDISR0) 0xF308 (Ch. 0)
DMA/Interrupt Status Register 1 (SIDISR1) 0xF408 (Ch. 1)

These registers indicate the DMA or interrupt status information.

31															16
								0							

: Type
: Initial value

15	14	13	12	11	10	9	8	7	6	5	4				0
UBRK	UVALID	UFER	UPER	UOER	ERI	TOUT	TDIS	RDIS	STIS	0	RFDN				
R	R	R	R	R	R/W0C	R/W0C	R/W0C	R/W0C	R/W0C				R		: Type
0	1	0	0	0	0	0	1	0	0				00000		: Initial value

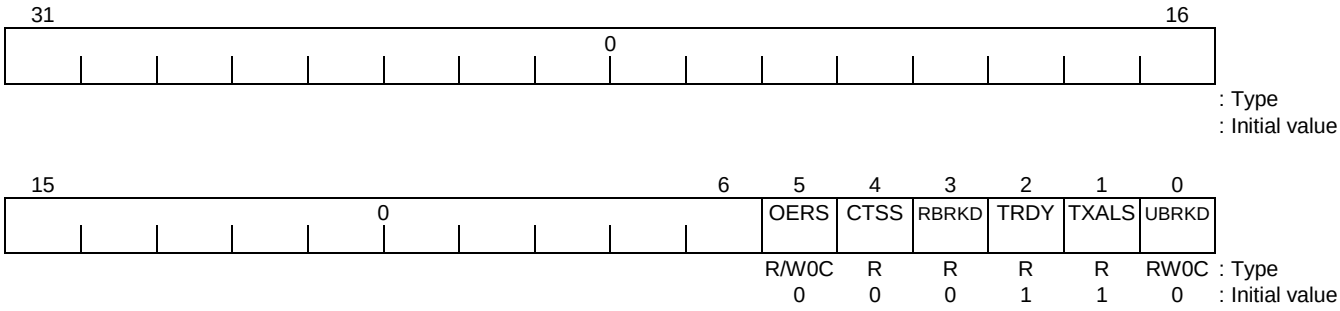
Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	UBRK	Receive Break	UART Break (Initial value: 0, R) This field indicates the break reception status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: No breaks 1: Detect breaks
14	UVALID	Receive FIFO Available Status	UART Available Data (Initial value: 1, R) This field indicates whether or not data exists in the Receive FIFO (SIRFIFO). 0: Data exists in the Receive FIFO. 1: No data exists in the Receive FIFO.
13	UFER	Frame Error	UART Frame Error (Initial value: 0, R) This field indicates the frame error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no frame errors. 1: There are frame errors.
12	UPER	Parity Error	UART Parity Error (Initial value: 0, R) This field indicates the parity error status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no parity errors. 1: There are parity errors.
11	UOER	Overrun Error	UART Overrun Error (Initial value: 0, R) This register indicates the overrun status of the next data in the Receive FIFO to be read. Reading the Receive FIFO Register (SIRFIFO) updates the status. 0: There are no overrun errors. 1: There are overrun errors.
10	ERI	Reception Error Interrupt	Receive Data Error Interrupt (Initial value: 0, R/W0C) This bit is immediately set to “1” when a reception error (Frame Error, Parity Error, or Overrun Error) is detected.
9	TOUT	Reception Time Out	Time Out (Initial value: 0, R/W0C) This bit is set to “1” when a reception time out occurs.
8	TDIS	Transmission Data Empty	Transmit DMA/Interrupt Status (Initial value: 1, R/W0C) This bit is set when available space of the amount set by the Transmit FIFO Request Trigger Level (TDIL) of the FIFO Control Register (SIFCR) exists in the Transmit FIFO.
7	RDIS	Reception Data Full	Receive DMA/Interrupt Status (Initial value: 0, R/W0C) This bit is set when valid data of the amount set by the Receive FIFO Request Trigger Level (RDIL) of the FIFO Control register (SIFCR) is stored in the Receive FIFO.

Figure 11.4.3 DMA/Interrupt Status Register (1/2)

Bits	Mnemonic	Field Name	Description
6	STIS	Status Change	Status Change Interrupt Status (Initial value: 0, R/W0C) This bit is set when at least one of the interrupt statuses selected by the Status Change Interrupt Condition field (STIE) of the DMA/Interrupt Control Register (SIDICR) becomes “1”.
5	—	Reserved	—
4:0	RFDN	Reception Data Stage Status	Receive FIFO Data Number (Initial value: 00000, R) This field indicates how many stages of reception data remain in the Receive FIFO (0 – 16 stages).

Figure 11.4.3 DMA/Interrupt Status Register (2/2)

11.4.4 Status Change Interrupt Status Register 0 (SISCISR0) 0xF30C (Ch. 0)
Status Change Interrupt Status Register 1 (SISCISR1) 0xF40C (Ch. 1)

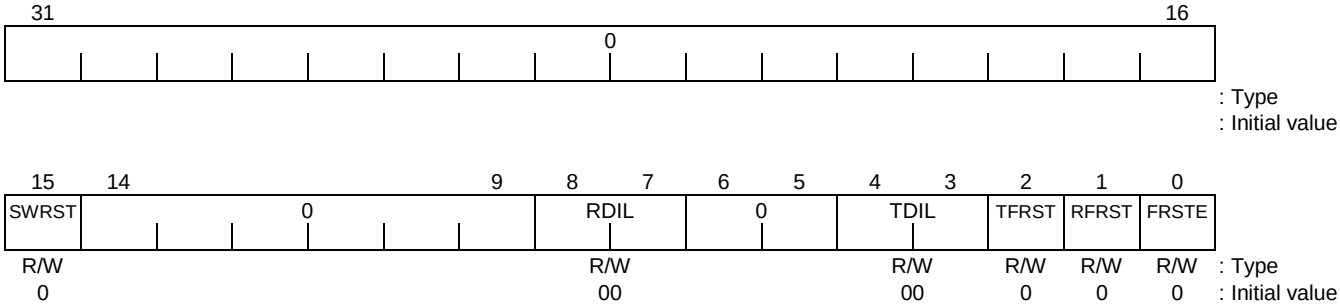


Bits	Mnemonic	Field Name	Description
31:6	—	Reserved	—
5	OERS	Overrun Error	Overrun Error Status (Initial value: 0, R/W0C) This bit is immediately set to “1” when an overrun error is detected. This bit is cleared when a “0” is written.
4	CTSS	CTS Status	CTS Terminal Status (Initial value: 0, R) This field indicates the status of the CTS signal. 1: The CTS signal is High. 0: The CTS signal is Low.
3	RBRKD	Receiving Break	Receive Break (Initial value: 0, R) This bit is set when a break is detected. This bit is automatically cleared when a frame that is not a break is received. 1: Current status is Break. 0: Current status is not Break.
2	TRDY	Transmission Data Empty	Transmit Ready (Initial value: 1, R) This bit is set to “1” if at least one stage in the Transmit FIFO is free.
1	TXALS	Transmission Complete	Transmit All Sent (Initial value: 1, R) This bit is set to “1” if the Transmit FIFO and all transmission shift registers are empty.
0	UBRKD	Break Detected	UART Break Detect (Initial value: 0, R/W0C) This bit is set when a break is detected. Once set, this bit remains set until cleared by writing a “0” to it.

Figure 11.4.4 Status Change Interrupt Status Register

11.4.5 FIFO Control Register 0 (SIFCR0) 0xF310 (Ch. 0)
FIFO Control Register 1 (SIFCR1) 0xF410 (Ch. 1)

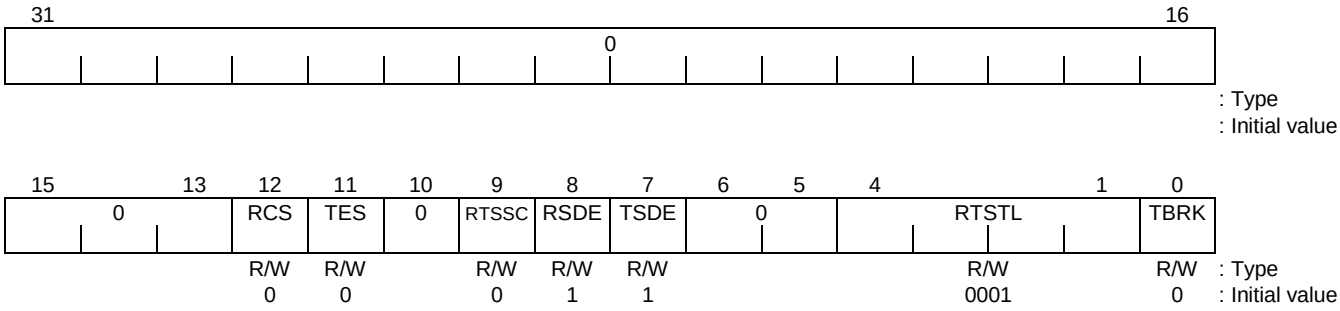
These registers set control of the Transmit/Receive FIFO buffer.



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	SWRST	Software Reset	Software Reset (Initial value: 0, R/W) This field performs SIO resets except for the FIFOs. Setting this bit to “1” initiates the reset. Set registers are also initialized. This bit returns to “0” when initialization is complete. 0: Normal operation 1: SIO software reset
14:9	—	Reserved	—
8:7	RDIL	Receive FIFO Request Trigger Level	Receive FIFO DMA/Interrupt Trigger Level (Initial value: 00, R/W) This register sets the level for reception data transfer from the Receive FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: 12 Bytes
6:5	—	Reserved	—
4:3	TDIL	Transmit FIFO Request Trigger Level	Transmit FIFO DMA/Interrupt Trigger Level (Initial value: 00, R/W) This register sets the level for transmission data transfer to the Transmit FIFO. 00: 1 Byte 01: 4 Bytes 10: 8 Bytes 11: Setting disabled
2	TFRST	Transmit FIFO Reset	Transmit FIFO Reset (Initial value: 0, R/W) The Transmit FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Cancel reset by using the software to clear this bit. 0: During operation 1: Reset Transmit FIFO
1	RFRST	Receive FIFO Reset	Receive FIFO Reset (Initial value: 0, R/W) The Receive FIFO buffer is reset when this bit is set. This bit is valid when the FIFO Reset Enable bit (FRSTE) is set. Cancel reset by using the software to clear this bit. 0: During operation 1: Reset Receive FIFO
0	FRSTE	FIFO Reset Enable	FIFO Reset Enable (Initial value: 0, R/W) This field is the Reset Enable for the Transmit/Receive FIFO buffer. The FIFO is reset by combining the Transmit FIFO Reset bit (TFRST) and Receive FIFO Reset bit (RRST). 0: During operation 1: Reset Enable

Figure 11.4.5 FIFO Control Register

11.4.6 Flow Control Register 0 (SIFLCR0) 0xF314 (Ch. 0)
Flow Control Register 1 (SIFLCR1) 0xF414 (Ch. 1)

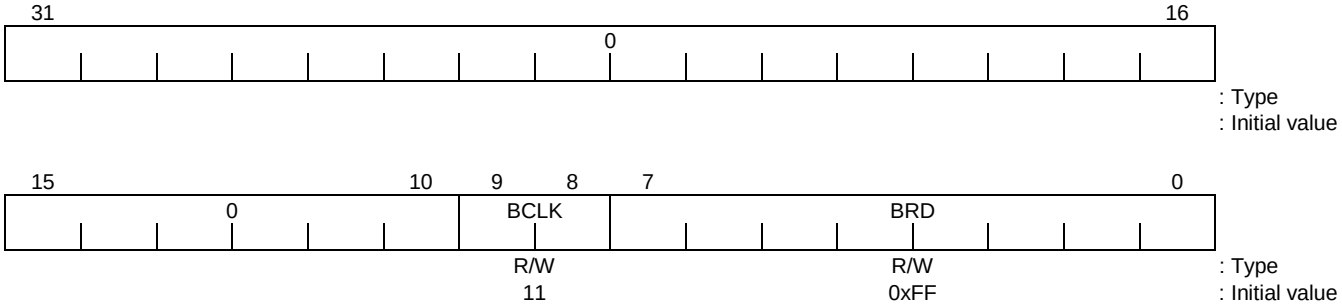


Bits	Mnemonic	Field Name	Description
31:13	—	Reserved	—
12	RCS	RTS Signal Control Select	RTS Control Select (Initial value: 0, R/W) This field sets the reception flow control using RTS output signals. 0: Disable flow control using RTS signals. 1: Enable flow control using RTS signals.
11	TES	CTS Signal Control Select	CTS Control Select (Initial value: 0, R/W) This field sets the transmission flow control using CTS input signals. 0: Disable flow control using CTS signals. 1: Enable flow control using CTS signals.
10	—	Reserved	—
9	RTSSC	RTS Software Control	RTS Software Control (Initial value: 0, R/W) This register is used for software control of RTS output signals. 0: Set the RTS signal to Low (can receive data). 1: Sets the RTS signal to High (transmission pause request)
8	RSDE	Serial Data Reception Enable	Receive Serial Data Enable (Initial value: 1, R/W) This is the Serial Data Enable bit. When this bit is cleared, data reception starts after the start bit is detected. The RTS signal will not become High even if this bit is cleared. 0: Enable (can receive data) 1: Disable (halt reception)
7	TSDE	Serial Data Transmit Enable	Transmit Serial Data Enable (Initial value: 1, R/W) This is the Serial Data Transmission Enable bit. When this bit is cleared, data transmission starts. When set, transmission stops after completing transmission of the current frame. 0: Enable (can transmit data) 1: Disable (halt transmission)
6:5	—	Reserved	—
4:1	RTSTL	RTS Active Trigger Level	RTS Trigger Level (Initial value: 0001, R/W) The RTS hardware control assert level is set by the reception data stage count of the Receive FIFO. 0000: Disable setting 0001: 1 : 1111: 15
0	TBRK	Break Transmission	Break Transmit (Initial value: 0, R/W) Transmits a break. The TXD signal is Low while TBRK is set to “1”. 0: Disable (clear break) 1: Enable (transmit break)

Figure 11.4.6 Flow Control Register

11.4.7 Baud Rate Control Register 0 (SIBGR0) 0xF318 (Ch. 0)
Baud Rate Control Register 1 (SIBGR1) 0xF418 (Ch. 1)

These registers select the clock that is provided to the baud rate generator and set the divide value.



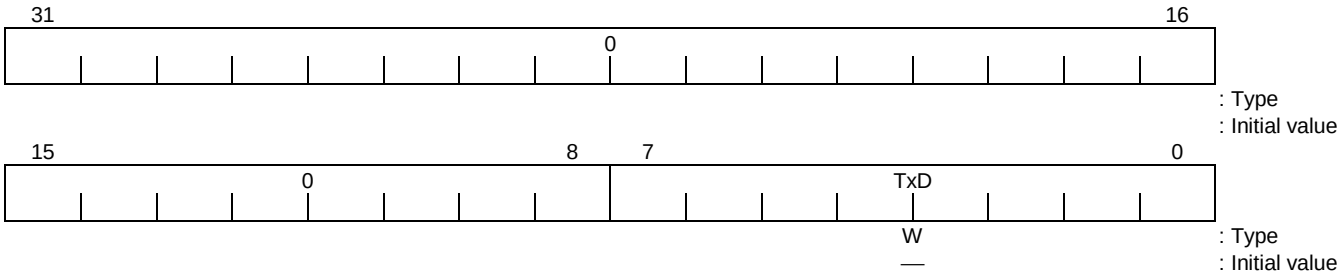
Bits	Mnemonic	Field Name	Description
31:10	—	Reserved	—
9:8	BCLK	Baud Rate Generator Clock	Baud Rate Generator Clock (Initial value: 11, R/W) This field sets the input clock for the baud rate generator. 00: Select prescalar output T0 (IMBUSCLKF/2) 01: Select prescalar output T2 (IMBUSCLKF/8) 10: Select prescalar output T4 (IMBUSCLKF/32) 11: Select prescalar output T6 (IMBUSCLKF/128)
7:0	BRD	Baud Rate Divide Value	Baud Rate Divide Value (Initial value: 0xFF, R/W) This field set divide value BRG of the baud rate generator. This value is expressed as a binary value.

Figure 11.4.7 Baud Rate Control Register

11.4.8 Transmit FIFO Register 0 (SITFIFO0) 0xF31C (Ch. 0)
Transmit FIFO Register 1 (SITFIFO1) 0xF41C (Ch. 1)

When using the DMA Controller to perform DMA transmission, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

- Little Endian: 0xF31C (Ch.0), 0xF41C (Ch.1)
- Big Endian: 0xF31F (Ch.0), 0xF41F (Ch.1)



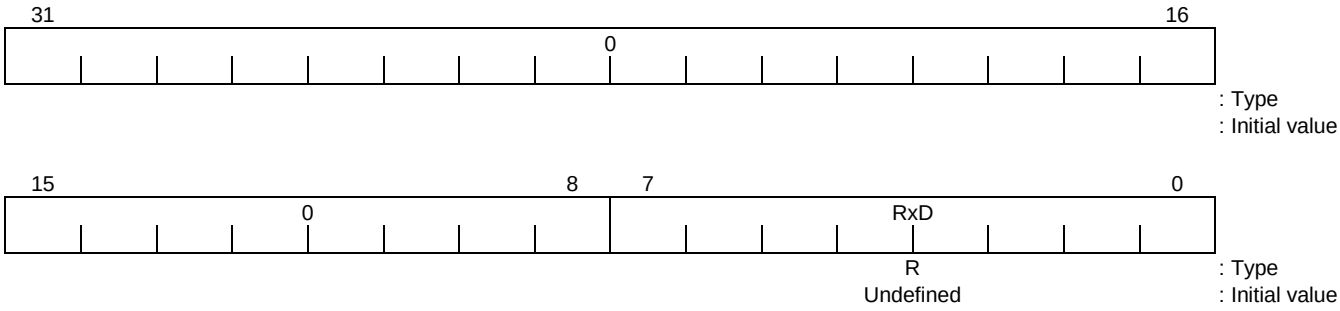
Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	TxD	Transmission Data	Transmit Data (Initial value: –, W) Data written to this register are written to the Transmit FIFO.

Figure 11.4.8 Transmit FIFO Register

11.4.9 Receive FIFO Register 0 (SIRFIFO0) 0xF320 (Ch. 0)
Receive FIFO Register 1 (SIRFIFO1) 0xF420 (Ch. 1)

When using the DMA Controller to perform DMA transmission, set the following addresses in the Destination Address Register (DMDARn) of the DMA Controller according to the Endian Mode bit (DMCCRn.LE) setting of the DMA Controller.

- Little Endian: 0xF320 (Ch.0), 0xF420 (Ch.1)
- Big Endian: 0xF323 (Ch.0), 0xF423 (Ch.1)



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	RxD	Reception Data	Receive Data (Initial value: undefined, R) This field reads reception data from the Receive FIFO. Reading this register updates the Reception Data Status.

Figure 11.4.9 Receive FIFO Register

12. Timer/Counter

12.1 Features

The TX4925 has an on-chip 3-channel timer/counter.

- 32-bit Up Counter: 3 Channels
- Interval Timer Mode (Channel 0, 1, 2)
- Pulse Generator Mode (Channel 0, 1)
- Watchdog Timer Mode (Channel 2)
- Timer Output Signal (TIMER[1:0]) × 2
- Counter Input Signal (TCLK): × 1

12.2 Block Diagram

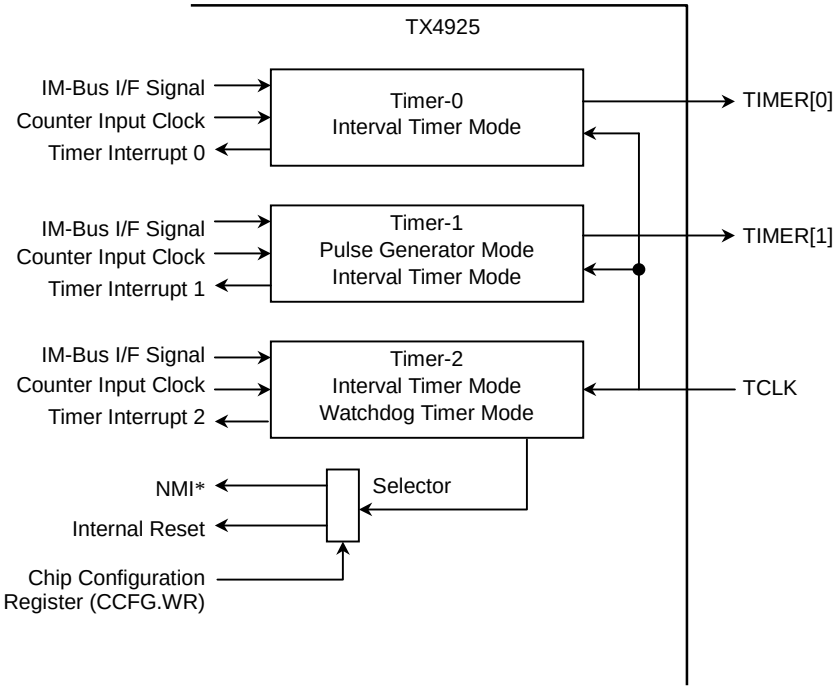


Figure 12.2.1 Connecting Timer Module Inside the TX4925

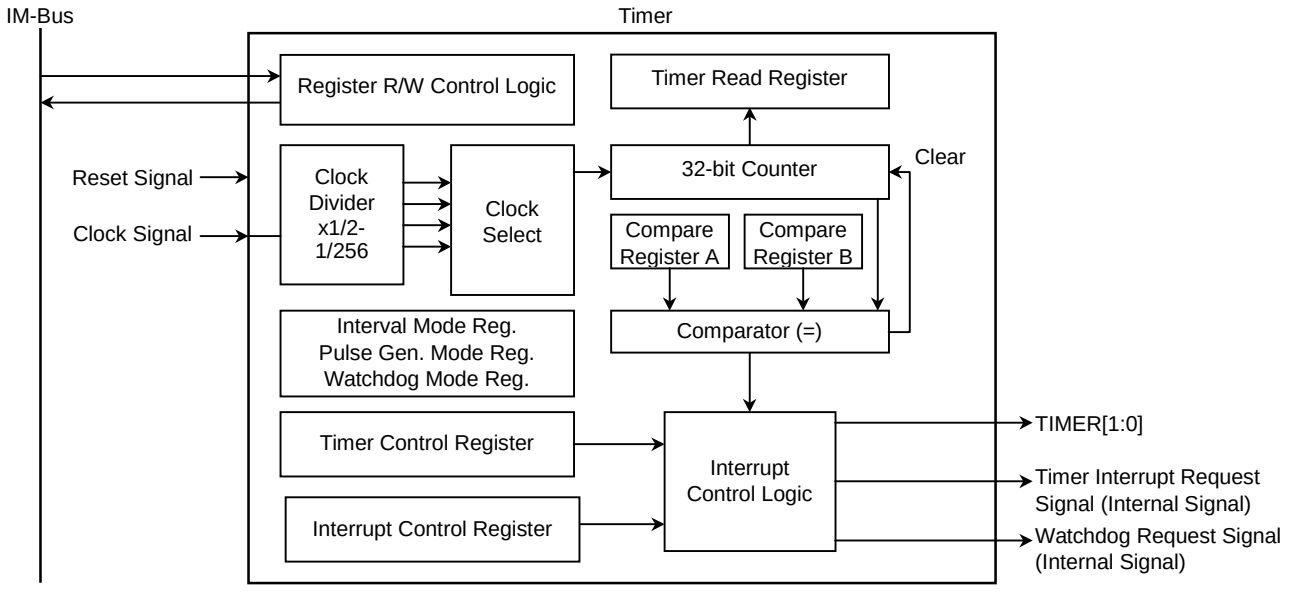


Figure 12.2.2 Timer Internal Block Diagram

12.3 Detailed Explanation

12.3.1 Overview

The TX4925 has an on-chip 3-channel 32-bit timer/counter. Each channel supports the following modes.

(1) Interval Timer Mode (Timer 0, 1, 2)

This mode periodically generates interrupts.

(2) Pulse Generator Mode (Timer 0, 1)

This is the pulse signal output mode.

(3) Watchdog Timer Mode (Timer 2)

This mode is used to monitor system abnormalities.

12.3.2 Counter Clock

The clock used for counting can be set to a frequency that is 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of the internal clock (IMBUSCLK) frequency, or can be selected from nine counter input signal (TCLK) types. Divide Register *n* (TMCCDRn) and the Counter Clock Select bit (TMTCRn.CCS) are used to select the counter clock. In this situation, IMBUSCLK is the internal clock signal which is the G-Bus clock divided by 2. See “Chapter 6 Clocks” for more information.

The counter input signal (TCLK) is used by three channels. Using TCLK makes it possible to count external events. The External Clock Edge bit (TMTCRn.ECES) can be used to select the clock rising/falling count.

Set the TCLK clock frequency to 45% or less of IMBUSCLK (TCLK = 18 MHz or less when IMBUSCLK = 40 MHz). The following table shows example count times when using 40 MHz IMBUSCLK.

Table 12.3.1 Divide Value and Count (IMBUSCLK = 40 MHz)

Divide Rate	TMCCDRn. CCD	Counter Clock Frequency (Hz)	Resolution (ns)	Max. Set Time (sec.)	TMCPRAn Value for 1 sec.
2	000	20.0 M	50.00	214.75	20000000
4	001	10.0 M	100.00	429.50	10000000
8	010	5.0 M	200.00	858.99	5000000
16	011	2.5 M	400.00	1717.99	2500000
32	100	1.25 M	800.00	3435.97	1250000
64	101	625.0 K	1600.00	6871.95	625000
128	110	312.5 K	3200.00	13743.90	312500
256	111	156.25 K	6400.00	27487.79	156250

12.3.3 Counter

Each channel has an independent 32-bit counter. Set the Timer Count Enable bit (TMTCRn.TCE) and the 32-bit counter will start counting.

Clear the Timer Count Enable bit to stop the counter. If the Counter Reset Enable bit (TMTCRn.CRE) is set, then the counter will be cleared also. The Watchdog Timer Disable bit (TMWTRM2.WDIS) must be set in order to stop and clear this counter when in the Watch Dog Timer mode.

Also, reading the Timer Read Register (TMTRR) makes it possible to fetch the counter value.

12.3.4 Interval Timer Mode

The Interval Timer mode is used to periodically generate interrupts. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to “00” sets the timer to the Interval Timer mode. This mode can be used by all timers.

When the count value matches the value of Compare Register A (TMCPRAn), the Interval Timer TMCPRA Status bit (TMTISRn.TIIS) of the Timer Interrupt Status Register is set. When the Interval Timer Interrupt Enable bit (TMITMRn.TIIE) of the Interval Timer Mode Register is set, timer interrupts occur. When a “0” is written to the Interval Timer TMCPRA Status bit (TMTISRn.TIIS), TIIS is cleared and timer interrupts stop.

If the Timer Zero Clear Enable bit (TMITMRn.TZCE) is set, the counter is cleared to 0 if the count value matches the Compare Register A (TMCPRAn) value. Count operation stops when the Timer Zero Clear Enable bit (TMITMRn.TZCE) is cleared.

The level of the TIMER[1:0] output signal stays in the High level in this mode. Figure 12.3.1 shows an outline of the count operation and generation of interrupts when in the Interval Timer mode and Figure 12.3.2 shows the operation when using an external input clock.

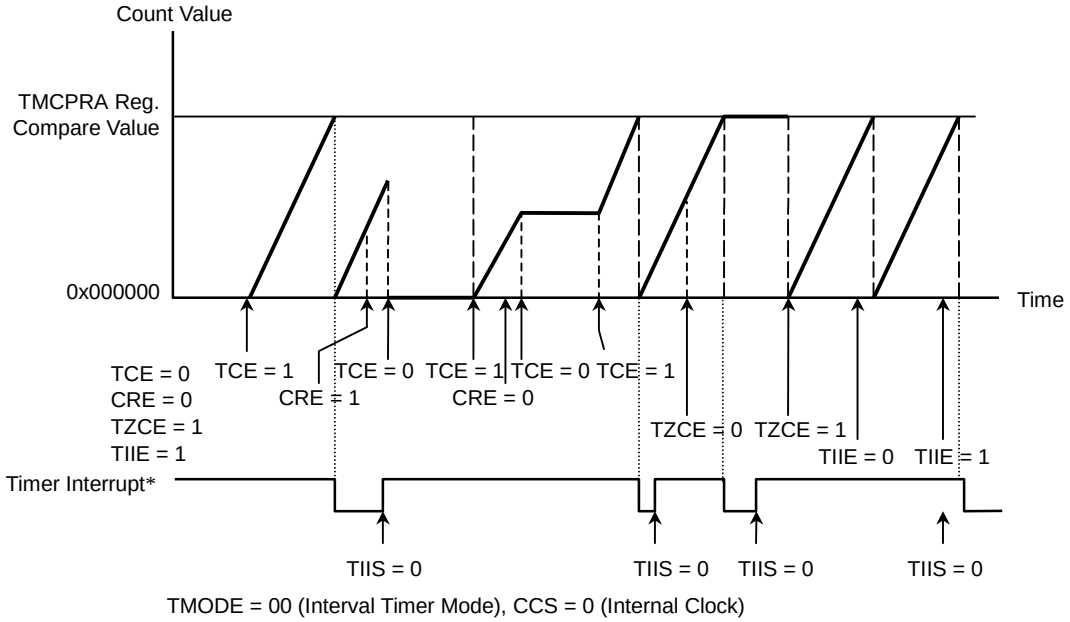


Figure 12.3.1 Operation Example of Interval Timer (Using Internal Clock)

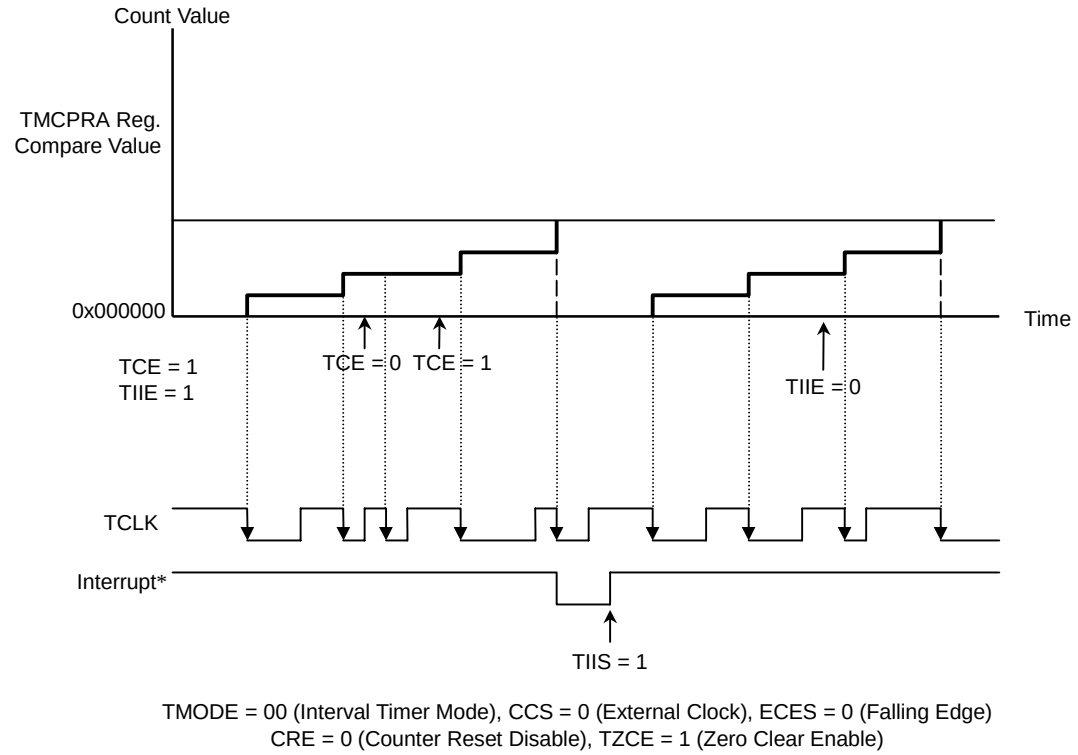


Figure 12.3.2 Operation Example of the Interval Timer (External Input Clock: Rising Edge Operation)

12.3.5 Pulse Generator Mode

When in the Pulse Generator mode, use Compare Register A (TMCPRAn) and Compare Register B (TMCPRBn) to output a particular period and particular duty square wave to the TIMER[n] signal. Setting the Timer Mode field (TMTCRn.TMODE) of the Timer Control Register to “01” sets the timer to the Pulse Generator mode. Timer 0 and Timer 1 can be used, but Timer 2 cannot.

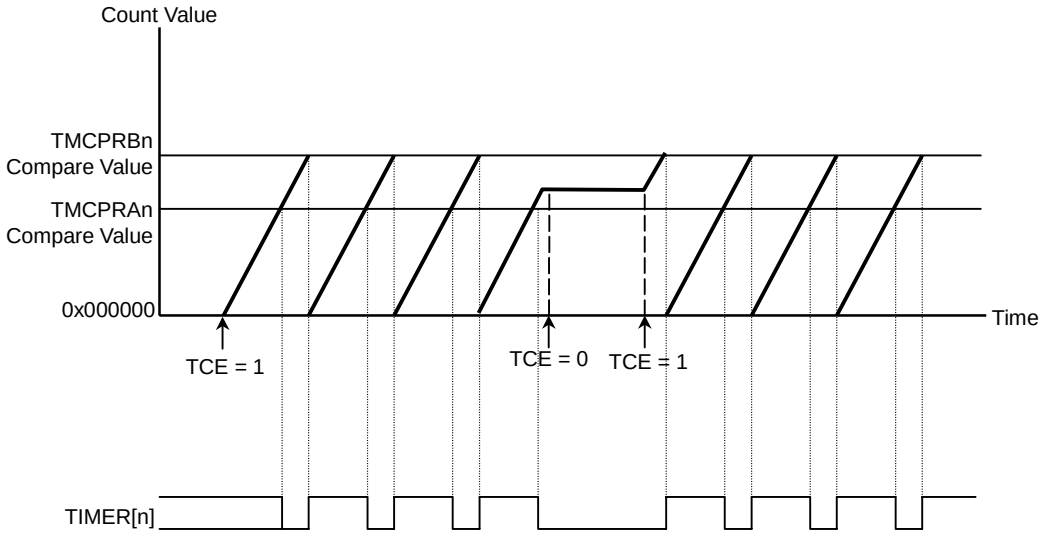
The initial state of the TIMER[n] signal can be set by the Flip Flop Default bit (TMPGMRn.FFI) of the Pulse Generator Mode Register.

The TIMER[n] output signal reverses when the counter value matches the value set in Compare Register A (TMCPRAn). The TIMER[n] output signal reverse again, clearing the counter when the counter continues counting and the value set in Compare Register B (TMCPRBn) and the counter value match. Consequently, a value greater than that in Compare Register A (TMCPRAn) must not be set in Compare Register B (TMCPRBn).

Interrupts can be generated in the Pulse Generator mode as well. However, this is not standard practice.

The Pulse Generator TMCPRA Status bit (TMTISRn.TPIAS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register A (TMCPRAn). Timer interrupts are generated when the TMCPRA Interrupt Enable bit (TMPGMRn.TPIAE) of the Pulse Generator Mode Register is set.

Similarly, the Pulse Generator TMCPRB Status bit (TMTISRn.TPIBS) of the Timer Interrupt Status Register is set when the count value matches the value of Compare Register B (TMCPRBn). Timer interrupts are generated when the TMCPRB Interrupt Enable bit (TMPGMRn.TPIBE) of the Pulse Generator Mode Register is set.



TMODE1 = 01 (Pulse Generator Mode), CCS= 0 (Internal Clock)
FFI = 1 (Initial High), CRE = 0 (Counter Reset Disable)

Figure 12.3.3 Operation Example of the Pulse Generator Mode

12.3.6 Watchdog Timer Mode

The Watchdog Timer mode is used to monitor system anomalies. The software periodically clears the counter and judges an anomaly to exist if the counter is not cleared within a specified period of time. Then, either the TX4925 is internally reset or an NMI is signaled to the TX49/H2 core. Set the Timer mode field (TMTCR2.TMODE) of the Timer Control Register to “10” to set the timer to the Watchdog Timer mode. This mode can only be used by Timer 2.

Use the Watchdog Reset bit (WR) of the Chip Configuration Register (CCFG) to select whether to perform an internal reset or signal an NMI. Set this bit to “1” to select Watchdog Reset, or set it to “0” to select NMI Signaling.

When the timer count reaches the value programmed in Compare Register A (TMCPRA2), the Watchdog Timer TMCPRA Match Status bit in the Timer Interrupt Status Register (TMTISR2.TWIS) is set. Either the watchdog timer reset or NMI is issued if the Timer Watchdog Enable bit in the Watchdog Timer Mode Register (TMWTMR2.TWIE) is set.

When the watchdog timer reset is selected, the Watchdog Reset Status bit in the Chip Configuration Register (CCFG.WDRST) is set. If the Watchdog Reset External Output bit in the Chip Configuration Register (CCFG.WDREXEN) is cleared, the entire TX4927 is initialized but the configuration registers. If the CCFG.WDREXEN bit is set, the WDRST* signal is asserted and remains asserted until the RESET* signal is asserted.

There are three ways of stopping NMI signaling from being performed.

(1) Clear the Watchdog Timer Interrupt Status bit (TMTISR2.TWIS) of the timer Interrupt Status Register.
(2) Clear the counter by writing “1” to the Watchdog Timer Clear bit (TMWTMR2.TWC) of the Watchdog Timer Mode Register.
(3) Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE) while the Watchdog Timer Disable bit (TMWTMR2.WDIS) is still set.

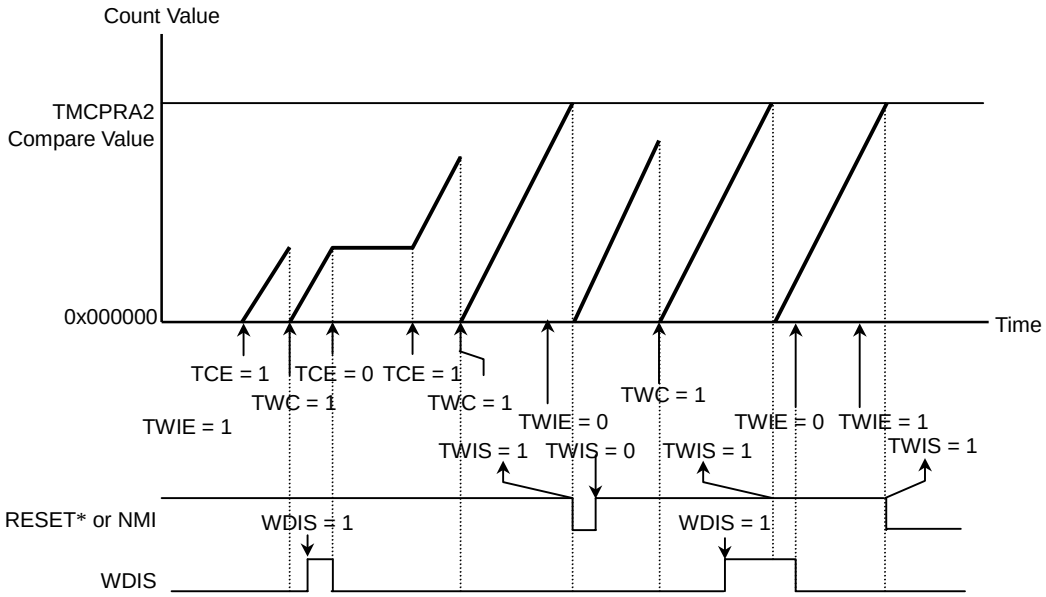
It is possible to stop the counter when in the Watchdog Timer mode by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register while the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set to “1”.

It is also possible to stop the counter by clearing the Counter Clock Divide Cycle Enable bit (TMTCR2.CCDE) of the Timer Control Register when the internal clock is being used as the counter clock.

It is not possible to directly write “0” to the Watchdog Timer Disable bit (TMWTMR2.WDIS). There are two ways to clear this bit.

(1) Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.WDIS)
(2) Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register

The level of the TIMER[1:0] output signal when in this mode remains in the default state (Low). Output is undefined when the mode is changed from the Pulse Generator mode to this mode.



TMODE = 10 (Watch Dog Timer Mode), CRE = 0 (Counter Reset Disable)

Figure 12.3.4 Operation Example of the Watchdog Timer Mode

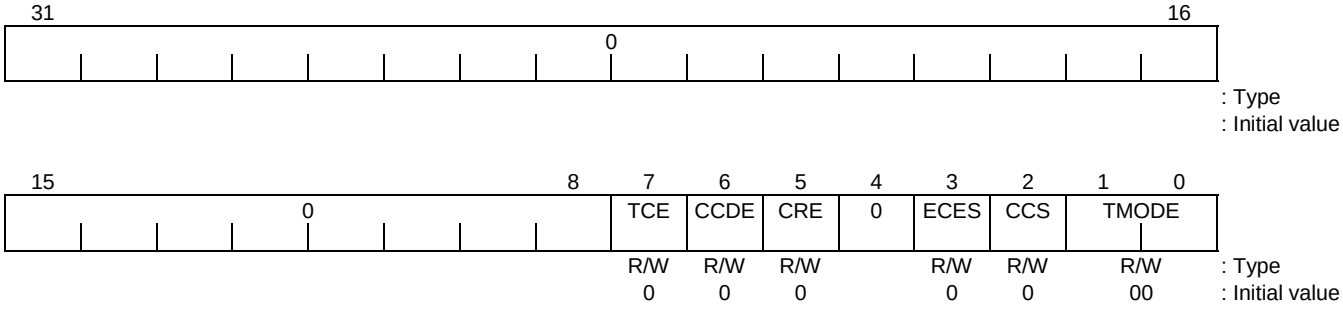
12.4 Registers

Table 12.4.1 Timer Register List

Reference	Offset Address	Bit Width	Register Symbol	Register Name
Time 0 (TMR0)				
12.4.1	0xF000	32	TMTCR0	Timer Control Register 0
12.4.2	0xF004	32	TMTISR0	Timer Interrupt Status Register 0
12.4.3	0xF008	32	TMCPRA0	Compare Register A 0
12.4.4	0xF00C	32	TMCPRB0	Compare Register B 0
12.4.5	0xF010	32	TMITMR0	Interval Timer Mode Register 0
12.4.6	0xF020	32	TMCCDR0	Divide Cycle Register 0
12.4.7	0xF030	32	TMPGMR0	Pulse Generator Mode Register 0
12.4.8	0xF040	32	TMWTMR0	(Reserved)
12.4.9	0xF0F0	32	TMTRR0	Timer Read Register 0
Timer 1 (TMR1)				
12.4.1	0xF100	32	TMTCR1	Timer Control Register 1
12.4.2	0xF104	32	TMTISR1	Timer Interrupt Status Register 1
12.4.3	0xF108	32	TMCPRA1	Compare Register A 1
12.4.4	0xF10C	32	TMCPRB1	Compare Register B 1
12.4.5	0xF110	32	TMITMR1	Interval Timer Mode Register 1
12.4.6	0xF120	32	TMCCDR1	Divide Cycle Register 1
12.4.7	0xF130	32	TMPGMR1	Pulse Generator Mode Register 1
12.4.8	0xF140	32	TMWTMR1	(Reserved)
12.4.9	0xF1F0	32	TMTRR1	Timer Read Register 1
Timer 2 (TMR2)				
12.4.1	0xF200	32	TMTCR2	Timer Control Register 2
12.4.2	0xF204	32	TMTISR2	Timer Interrupt Status Register 2
12.4.3	0xF208	32	TMCPRA2	Compare Register A 2
12.4.4	0xF20C	32	TMCPRB2	(Reserved)
12.4.5	0xF210	32	TMITMR2	Interval Timer Mode Register 2
12.4.6	0xF220	32	TMCCDR2	Divide Cycle Register 2
12.4.7	0xF230	32	TMPGMR2	(Reserved)
12.4.8	0xF240	32	TMWTMR2	Watchdog Timer Mode Register 2
12.4.9	0xF2F0	32	TMTRR2	Timer Read Register 2

12.4.1 Timer Control Register *n* (TMTCRn)

TMTCR0 0xF000
TMTCR1 0xF100
TMTCR2 0xF200

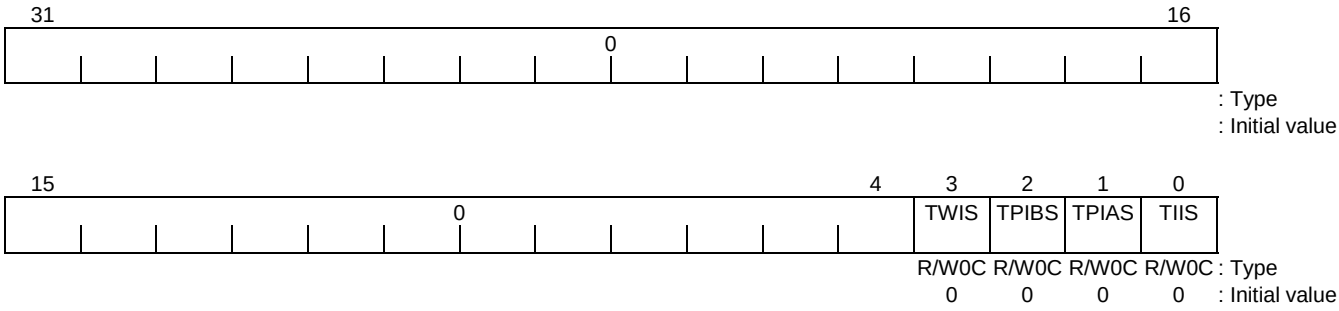


Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	TCE	Timer Counter Enable	Timer Count Enable (Initial value: 0, R/W) This field controls whether the counter runs or stops. When in the Watchdog mode, counter operation only stops when the Watchdog Timer Disable bit (TMWTMR2.WDIS) of the Watchdog Timer Mode Register is set. When the Watchdog Timer Disable bit is cleared, the value of this Timer Count Enable bit becomes “0”, but the count continues. 0: Stop counter (the counter is also cleared to “0” when CRE = 1) 1: Counter operation
6	CCDE	Counter Clock Divider Enable	Counter Clock Divide Enable (Initial value: 0, R/W) This bit enables the divide operation of the internal clock (IMBUSCLK). The counter stops if this bit is set to “0” when the internal bus clock is in use. 0: Disable 1: Enable
5	CRE	Counter Reset Enable	Counter Reset Enable (Initial value: 0, R/W) This bit controls the counter reset when the TCE bit was used to stop the counter. 1: Stop and reset the counter to “0” when the TCE bit is cleared to “0”. 0: Only stop the counter when the TCE bit is cleared to “0”.
4	—	Reserved	—
3	ECES	External Clock Edge Select	External Clock Edge Select (Initial value: 0, R/W) This bit specifies the counter operation edge when using the counter input signal (TCLK). 0: Falling edge of the counter input signal (TCLK) 1: Rising edge of the counter input signal (TCLK)
2	CCS	Counter Clock Select	Counter Clock Select (Initial value: 0, R/W) This bit specifies the timer clock. 0: Internal clock (IMBUSCLK) 1: External input clock (TCLK)
1:0	TMODE	Timer Mode	Timer Mode (Initial value: 00, R/W) This bit specifies the timer operation mode. 11: Reserved 10: Watchdog Timer mode (Timer 2), Reserved (Timer 0, 1) 01: Pulse Generator mode (Timer 0, 1), Reserved (Timer 2) 00: Interval Timer mode

Figure 12.4.1 Timer Control Register

12.4.2 Timer Interrupt Status Register *n* (TMTISRn)

TMTISR0	0xF004
TMTISR1	0xF104
TMTISR2	0xF204

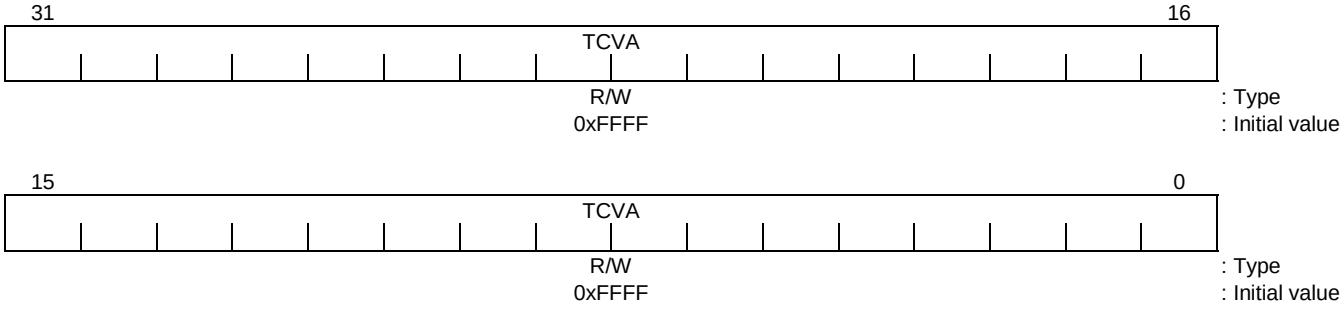


Bits	Mnemonic	Field Name	Description
31:4	—	Reserved	—
3	TWIS	Watchdog Timer Status	Watchdog Timer TMCPRA Match Status (Initial value: 0, R/W0C) (This bit is Reserved in the case of the TMTISR0 Register and the TMTISR1 Register.) When in the Watchdog Timer mode, this bit is set when the counter value matches Compare Register 2 (TMCPRA2). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Negate interrupt 1: Invalid
2	TPIBS	Pulse Generator TMCPRB Status	Pulse Generator TMCPRB Match Status (Initial value: 0, R/W0C) (This bit is Reserved in the case of the TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register Bn (TMCPRBn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid
1	TPIAS	Pulse Generator TMCPRA Status	Pulse Generator TMCPRA Match Status (Initial value: 0, R/W0C) (This bit is Reserved in the case of the TMTISR2 Register.) When in the Pulse Generator mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid
0	TIIS	Interval Timer TMCPRA Status	Interval Timer TMCPRA Match Status (Initial value: 0, R/W0C) When in the Interval Timer mode, this bit is set when the counter value matches Compare Register A n (TMCPRAn). This bit is cleared by writing a “0” to it. During Read 0: Did not match the Compare Register 1: Matched the Compare Register During Write 0: Clear 1: Invalid

Figure 12.4.2 Timer Interrupt Status Register

12.4.3 Compare Register An (TMCPRAn)

TMCPRA0	0xF008
TMCPRA1	0xF108
TMCPRA2	0xF208

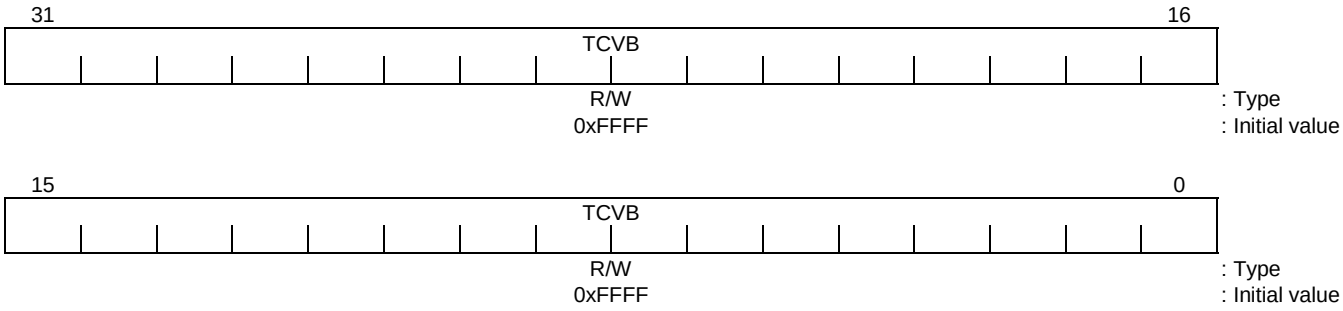


Bits	Mnemonic	Field Name	Description
31:0	TCVA	Timer Compare Register A	Timer Compare Value A (Initial value: 0xFFFF_FFFF, R/W) Sets the timer compare value as a 32-bit value. This register can be used in all modes.

Figure 12.4.3 Compare Register A

12.4.4 Compare Register Bn (TMCPRBn)

TMCPRB0 0xF00C
TMCPRB1 0xF10C

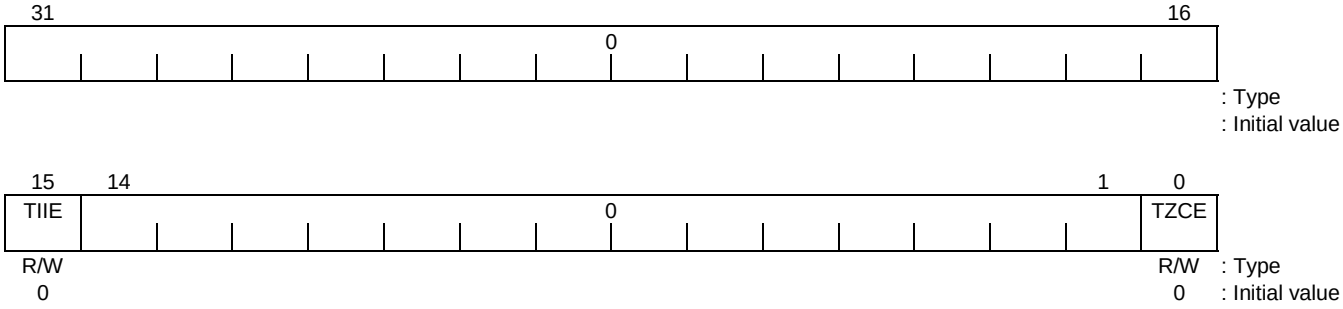


Bits	Mnemonic	Field Name	Description
31:0	TCVB	Timer Compare Value B	Timer Compare Value B (Initial value: 0xFFFF_FFFF, R/W) Sets the timer compare value as a 32-bit value. This register can only be used when in the Pulse Generator mode. Please set a value greater than that in Compare Register A.

Figure 12.4.4 Compare Register B

12.4.5 Interval Timer Mode Register *n* (TMITMRn)

TMITMR0	0xF010
TMITMR1	0xF110
TMITMR2	0xF210

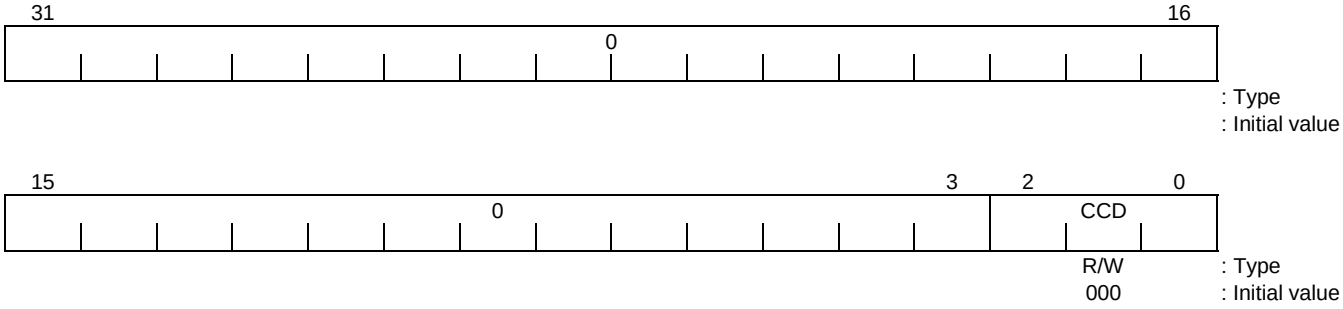


Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	TIIE	Interval Timer Interrupt Enable	Timer Interval Interrupt Enable (Initial value: 0, R/W) Sets Interval Timer TMCPRA Interrupt Enable/Disable. 0: Disable (mask) 1: Enable
14:1	—	Reserved	—
0	TZCE	Interval Timer Clear Enable	Interval Timer Zero Clear Enable (Initial value: 0, R/W) This bit specifies whether or not to clear the counter to “0” after the count value matches Compare Register A. Count stops at this value if it is not cleared. 0: Do not clear 1: Clear

Figure 12.4.5 Interval Timer Mode Register

12.4.6 Divide Register *n* (TMCCDRn)

TMCCDR0	0xF020
TMCCDR1	0xF120
TMCCDR2	0xF220

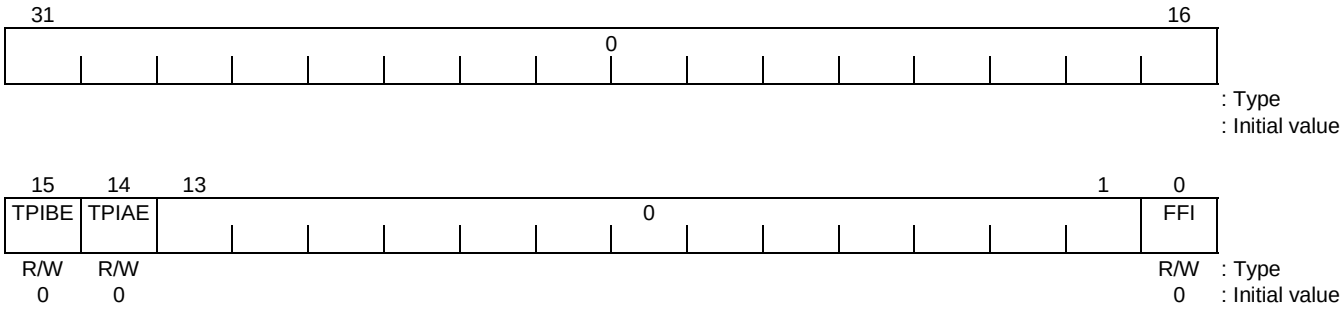


Bits	Mnemonic	Field Name	Description
31:3	—	Reserved	—
2:0	CCD	Counter Clock Divide Value	Counter Clock Divide (Initial value: 000, R/W) These bits specify the divide value when using the internal clock (IMBUSCLK) as the counter input clock source. The binary value n is divided by 2^{n+1}. 000: Divide by 2^1 (f/2) 001: Divide by 2^2 (f/4) 010: Divide by 2^3 (f/8) 011: Divide by 2^4 (f/16) 100: Divide by 2^5 (f/32) 101: Divide by 2^6 (f/64) 110: Divide by 2^7 (f/128) 111: Divide by 2^8 (f/256)

Figure 12.4.6 Divide Register

12.4.7 Pulse Generator Mode Register n (TMPGMRn)

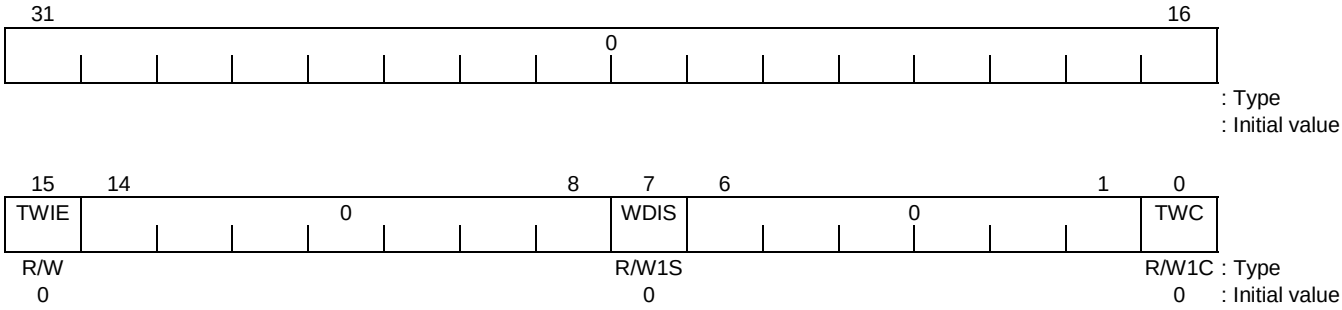
TMPGMR0 0xF030
TMPGMR1 0xF130



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	TPIBE	TMCPRB Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRB Enable (Initial value: 0, R/W) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRB and the counter value match. 0: Mask 1: Do not mask
14	TPIAE	TMCPRA Interrupt Enable	Timer Pulse Generator Interrupt by TMCPRA Enable (Initial value: 0, R/W) When in the Pulse Generator mode, this bit sets Interrupt Enable/Disable for when TMCPRA and the counter value match. 0: Mask 1: Do not mask
13:1	—	Reserved	—
0	FFI	Flip Flop Default	Initial TIMER Output Level (Initial value: 0, R/W) This bit specifies the TIMER[n] signal default when in the Pulse Generator mode. 0: Low 1: High

Figure 12.4.7 Pulse Generator Mode Register

12.4.8 Watchdog Timer Mode Register *n* (TMWTMRn) TMWTMR2 0xF240

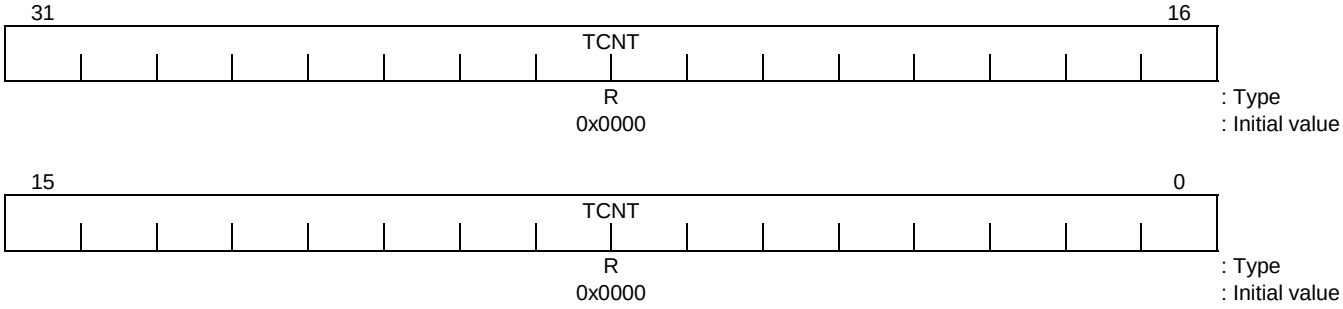


Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	TWIE	Watchdog Timer Signaling Enable	Timer Watchdog Enable (Initial value: 0, R/W) This bit sets NMI signaling enable/disable either when in the Watchdog Timer mode or during a reset. This bit cannot be cleared when the Watchdog Timer Disable bit (WDIS) is “0”. 0: Disable (mask) 1: Enable
14:8	—	Reserved	—
7	WDIS	Watchdog Timer Disable	Watchdog Timer Disable (Initial value: 0, R/W1S) Only when this bit is set can the counter be stopped by clearing the Watchdog Timer Signaling Enable bit (TWIE) or by clearing the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register. Writing “0” to this bit is not valid. This bit can be cleared in either of the following ways. Clear the Watchdog Timer Interrupt Enable bit (TMWTMR2.TWIE). Clear the Timer Counter Enable bit (TMTCR2.TCE) of the Timer Control Register.
6:1	—	Reserved	—
0	TWC	Watchdog Timer Clear	Watchdog Timer Clear (Initial value: 0, R/W1C) Setting this bit to “1” clears the counter. Writing “0” to this bit is not valid. This bit is always read as “0”.

Figure 12.4.8 Watchdog Timer Mode Register

12.4.9 Timer Read Register *n* (TMTRRn) 0xF0F0

TMTRR0	0xF0F0
TMTRR1	0xF1F0
TMTRR2	0xF2F0



Bits	Mnemonic	Field Name	Description
31:0	TCNT	Timer Counter	Timer Counter (Initial value: 0x0000_0000, R) This Read Only register is a 32-bit counter. Operation when this register is written to is undefined.

Figure 12.4.9 Timer Read Register 0

13. Parallel I/O Port

13.1 Characteristics

The TX4925 on-chip Parallel I/O port (PIO) is a 16-bit general-purpose parallel port. The input/output direction and the port type during output (totem pole output/open drain output) can be set for each bit.

13.2 Block Diagram

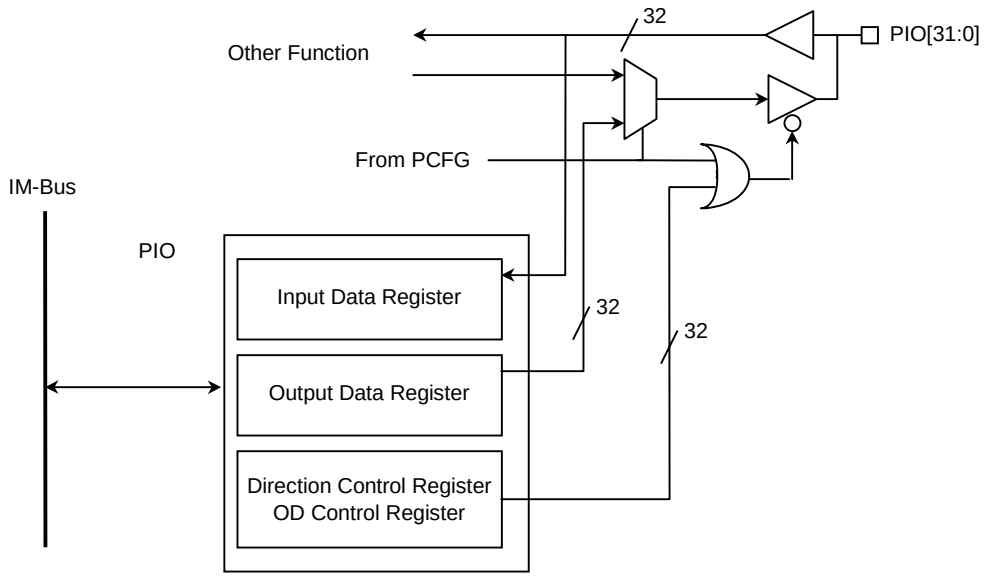


Figure 13.2.1 Parallel I/O Block Diagram

13.3 Detailed Description

13.3.1 Selecting PIO Pins

All of the 32-bit PIO signals are shared with other functions. The boot configuration signal (TDO) and pin configuration register (PCFG) determine which functions will be used. See Sections “3.2 Boot Configuration”, “3.3 Pin Multiplexing” and “5.2.3 Pin Configuration Register” for more information.

13.3.2 General-purpose Parallel Port

The four following registers are used to control the PIO port.

- PIO Output Data Register (PIODO)
- PIO Input Data Register (PIODI)
- PIO Direction Control Register (PIODIR)
- PIO Open Drain Control Register (PIOOD)

PIO signals can be selected by the PIO Direction Control Register (PIODIR) for each bit as either input or output.

Signals selected as output signals output the values written into the PIO Data Output Register (PIODO). The PIO Open Drain Control Register (PIOOD) can select whether each bit is either an open drain output or a totem pole output.

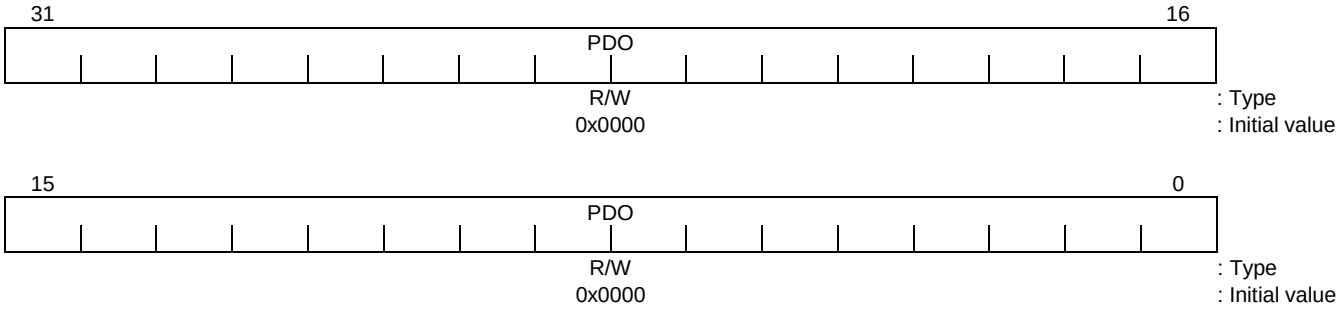
PIO signal status is indicated by the PIO Data Input Register. This register can be read out at any time regardless of the pin direction settings.

13.4 Registers

Table 13.4.1 PIO Register Map

Reference	Offset Address	Bit Width	Mnemonic	Register Name
13.4.1	0xF500	32	PIODO	PIO Output Data Register
13.4.2	0xF504	32	PIODI	PIO Input Data Register
13.4.3	0xF508	32	PIODIR	PIO Direction Control Register
13.4.4	0xF50C	32	PIOOD	PIO Open Drain Control Register

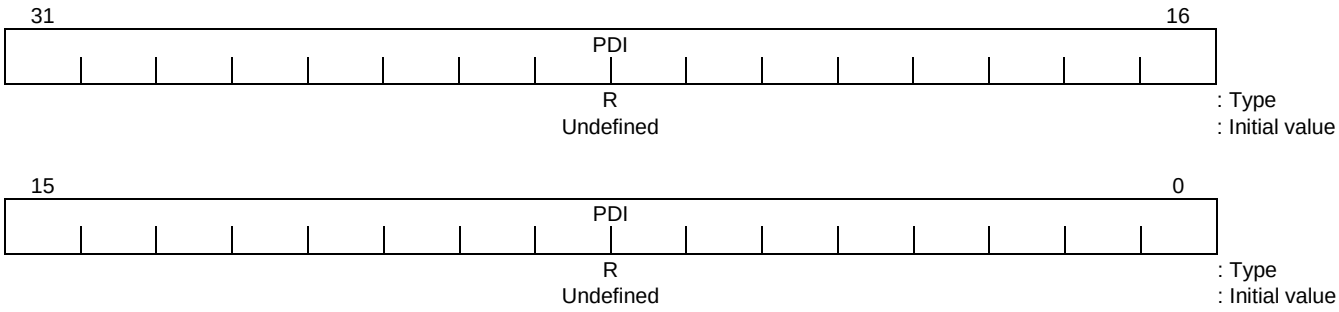
13.4.1 PIO Output Data Register (PIODO) 0xF500



Bits	Mnemonic	Field Name	Description
31:0	PDO [31:0]	Data Out	Port Data Output [31:0] (Initial value: 0x0000_0000, R/W) Data that is output to the PIO pin (PIO [31:0]).

Figure 13.4.1 PIO Output Data Register

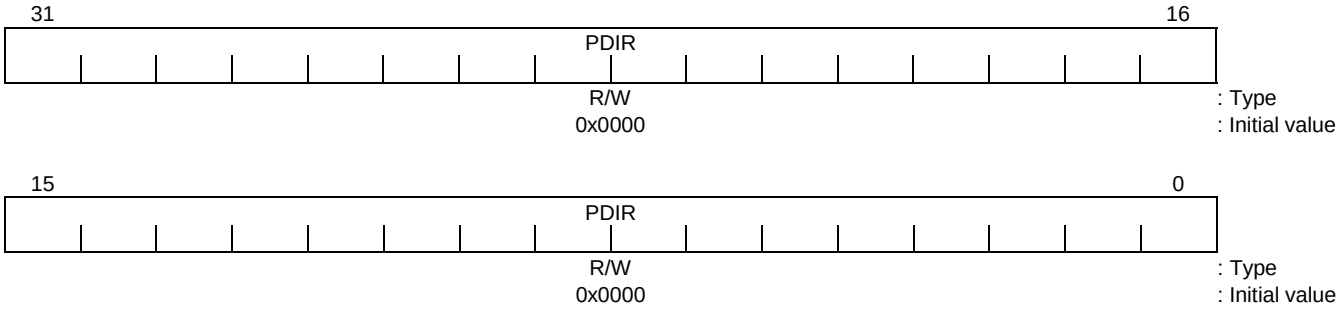
13.4.2 PIO Input Data Register (PIODI) 0xF504



Bits	Mnemonic	Field Name	Description
31:0	PDI [31:0]	Data In	Port Data Input [31:0] (Initial value: undefined, R) Data that is input to the PIO pin (PIO [31:0]).

Figure 13.4.2 PIO Input Data Register

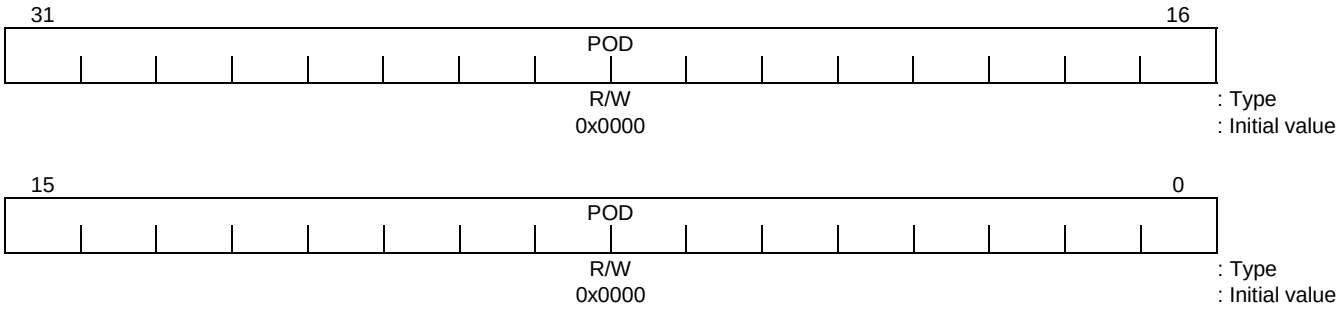
13.4.3 PIO Direction Control Register (PIODIR) 0xF508



Bits	Mnemonic	Field Name	Description
31 :0	PDIR [31:0]	Direction Control	Port Direction Control [31:0] (Initial value: 0x0000_0000, R/W) Sets the I/O direction of the PIO pin (PIO [31:0]). 0: Input (Reset) 1: Output

Figure 13.4.3 PIO Direction Control Register

13.4.4 PIO Open Drain Control Register (PIOOD) 0xF50C



Bits	Mnemonic	Field Name	Description
31 :0	POD [31:0]	Open Drain Control	Port Open Drain Control [31:0] (Initial value: 0x0000_0000, R/W) Sets whether to use the PIO pin (PIO [31:0]) as an open drain. 0: Open drain (Reset) 1: Totem pole

Figure 13.4.4 PIO Open Drain Control Register

14. AC-link Controller

14.1 Features

ACLC, AC-link controller module can be connected to audio and/or modem CODECs described in the "Audio CODEC '97 Revision 2.1" (AC'97) defined by Intel and can operate them. Refer to the following Web site for more information regarding the AC'97 specification.

http://developer.intel.com/ial/scalableplatforms/audio/

Its features are summarized as follows.

- Up to two CODECs are supported.
- AC'97 compliant CODEC register access protocol is supported.
- CODEC register access completion is recognized by polling or interrupt.
- Recording and playback of 16-bit PCM Left&Right channels are supported.
- Recording can be selected from PCM L&R or Mic.
- Playback of 16-bit Surround, Center, and LFE channels is supported.
- Variable Rate Audio recording is supported.
- Variable Rate Audio playback is supported.
- Line 1 and GPIO slots for Modem CODEC are supported.
- AC-link low-power mode, wake-up, and warm-reset are supported
- Sample-data I/O via DMA transfer is supported.

14.2 Configuration

Figure 14.2.1 illustrates the ACLC configuration.

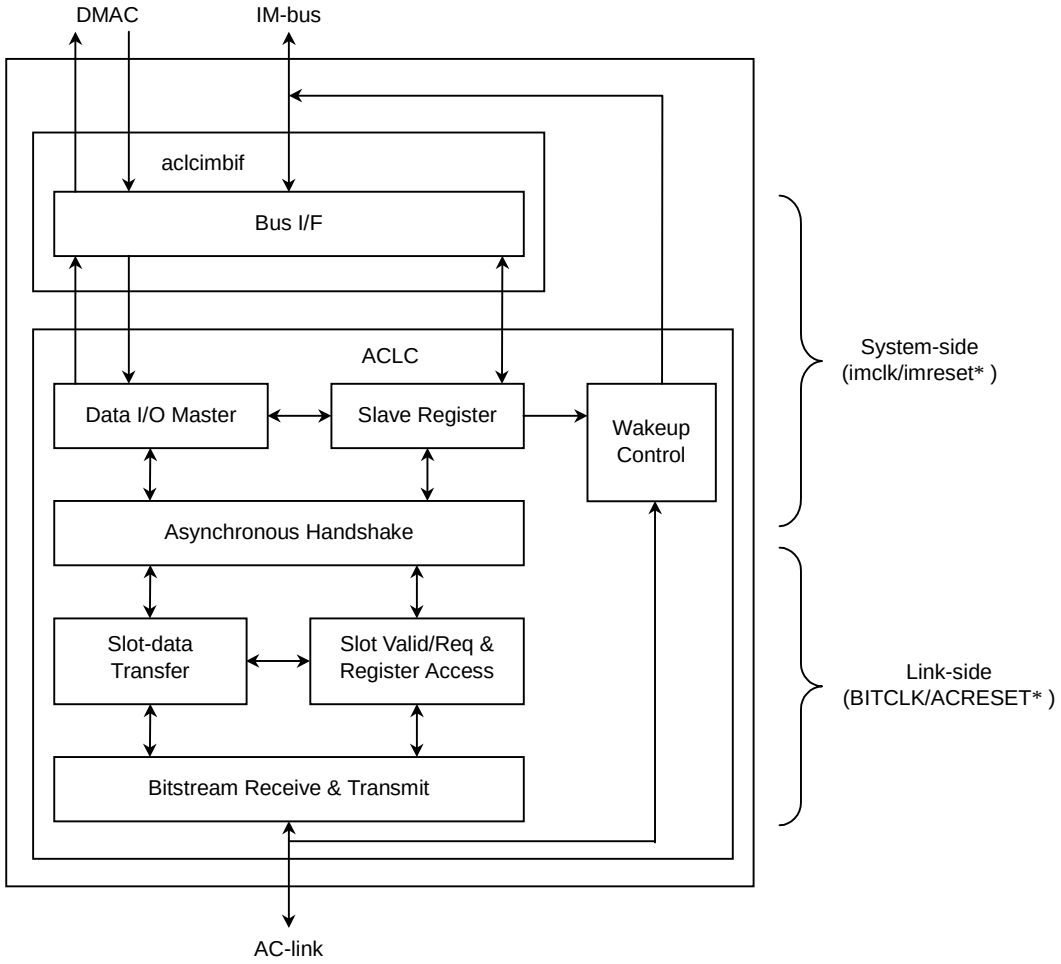


Figure 14.2.1 ACLC Module Configuration

14.3 Functional Description

ACLC provides four mechanisms to operate AC'97-compliant CODEC(s):

- AC-link status control (start-up and low-power mode)
- CODEC register access
- Sample-data transmission and reception
- GPIO operation

This section first describes the CODEC connection, chip configuration, and overall usage-flow. Then AC-link start-up sequence and the other mechanisms will be described. Using low-power mode comes last.

14.3.1 CODEC Connection

The ACLC module has two SDIN (named as SDATA_IN in the AC'97 specification) signals and supports up to two CODECs to be connected. This section shows some system configuration diagrams for typical usages. Note that the diagrams shown here is intended to provide conceptual understanding and some components may be necessary on the actual circuit board to ensure proper electrical signals. The diagrams assume CODECs compliant with the CODEC ID strapping recommendation described in the section D.5.2 of the AC'97 revision 2.1 specification.

14.3.1.1 Stereo Audio and Optional Modem Connection

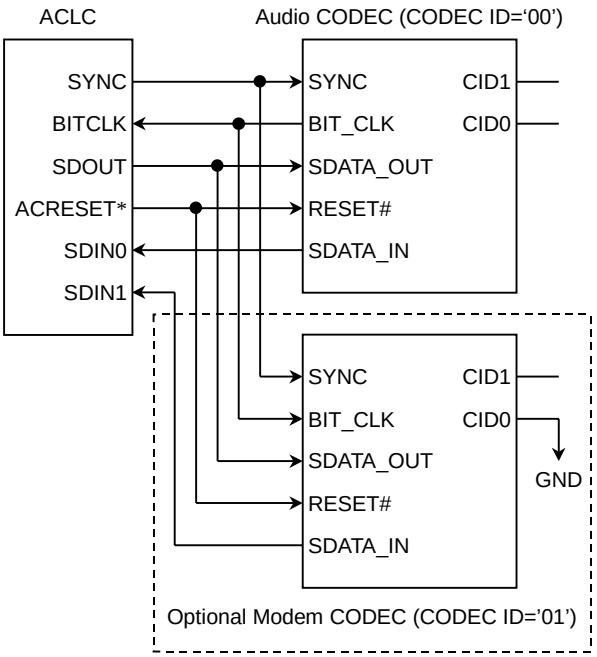


Figure 14.3.1 Stereo Audio and Optional Modem Connection Diagram

14.3.1.2 5.1 Channel Audio Connection

This sample assumes one CODEC with four DACs mapped to stereo front (3&4) and stereo rear (7&8) slots, and another CODEC with two DACs mapped to center (6) and LFE (9) slots.

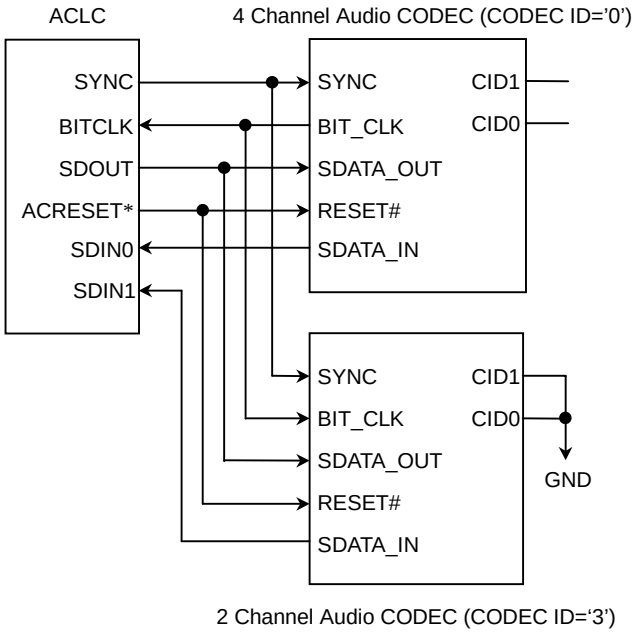


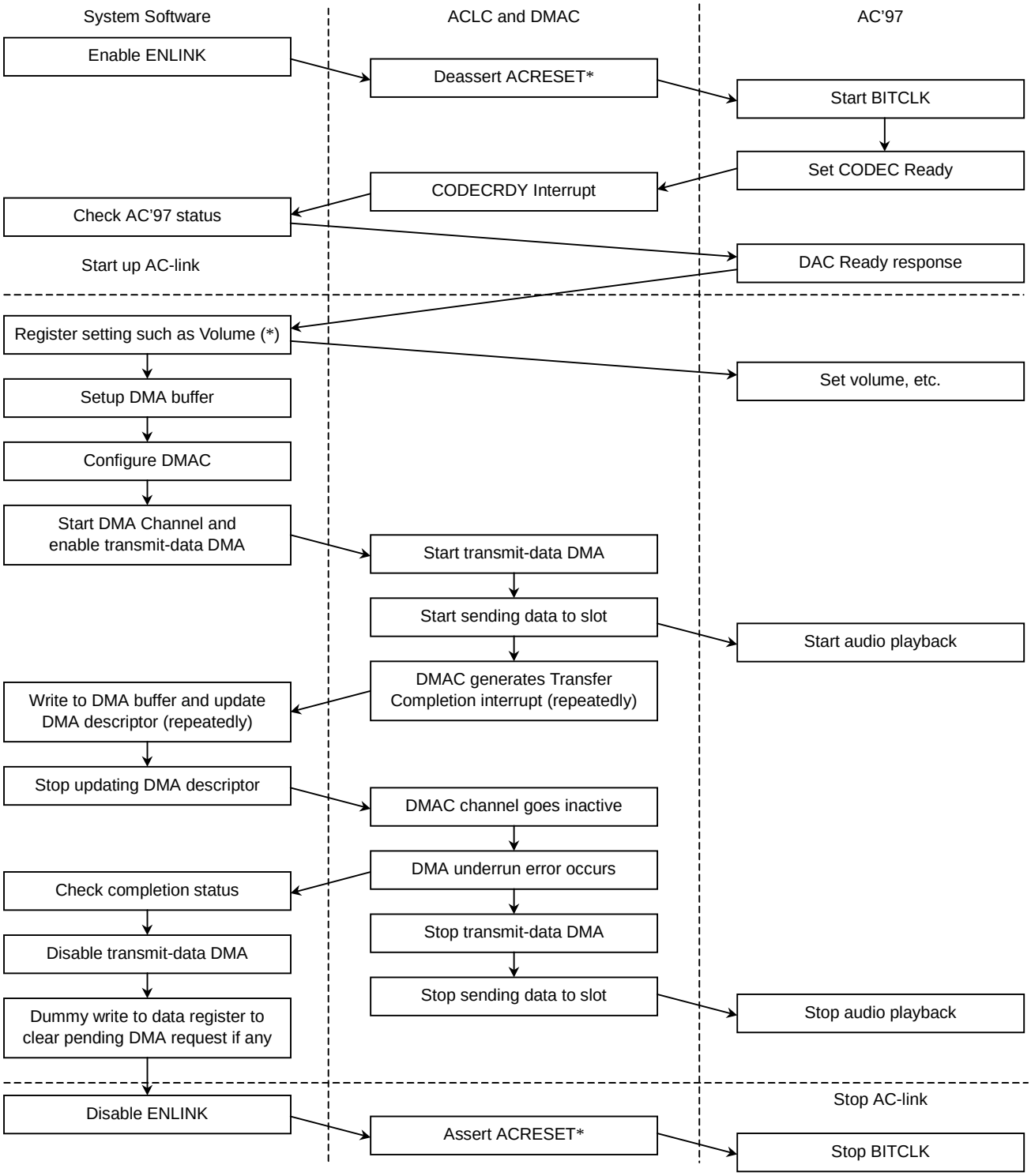
Figure 14.3.2 5.1 Channel Audio Connection Diagram

14.3.2 Pin Configuration

To utilize ACLC, the Select ACLC (SELACLC) bit in Pin Configuration Register (PCFG) must be set to 1. Refer to the Sections 3.3 and 5.2.3 for the detail of the pin configuration.

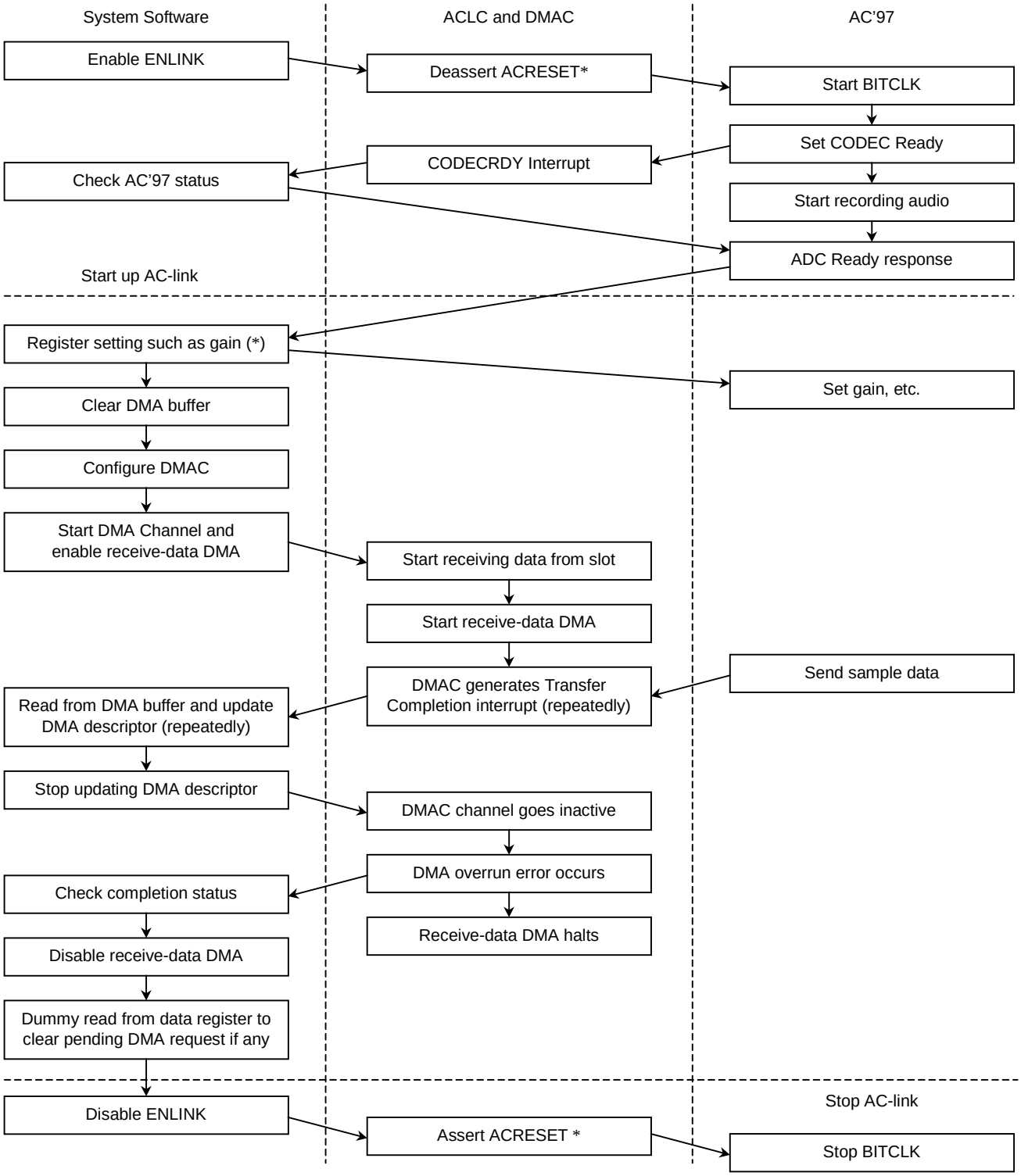
14.3.3 Usage Flow

This section outlines a process flow when using the AC'97 connected to ACLC. Refer to the subsequent sections for the details of each operation performed in this process flow. The diagrams below describe the audio playback and recording processes. The modem transmission and reception can be done in a similar way.



(*) Register settings such as volume can be made during data playback.

Figure 14.3.3 Audio Playback Process Flow



(*) Register settings such as gain can be made during data recording

Figure 14.3.4 Audio Recording Process Flow

14.3.4 AC-link Start Up

Figure 14.3.5 shows the conceptual sequence of AC-link start-up.

The ACLC Control Enable Register's Enable AC-link bit is used to deassert/assert the ACRESET* signal to the link side (including AC-link). This bit defaults to '0', so the CPU asserts the ACRESET* signal when it boots up.

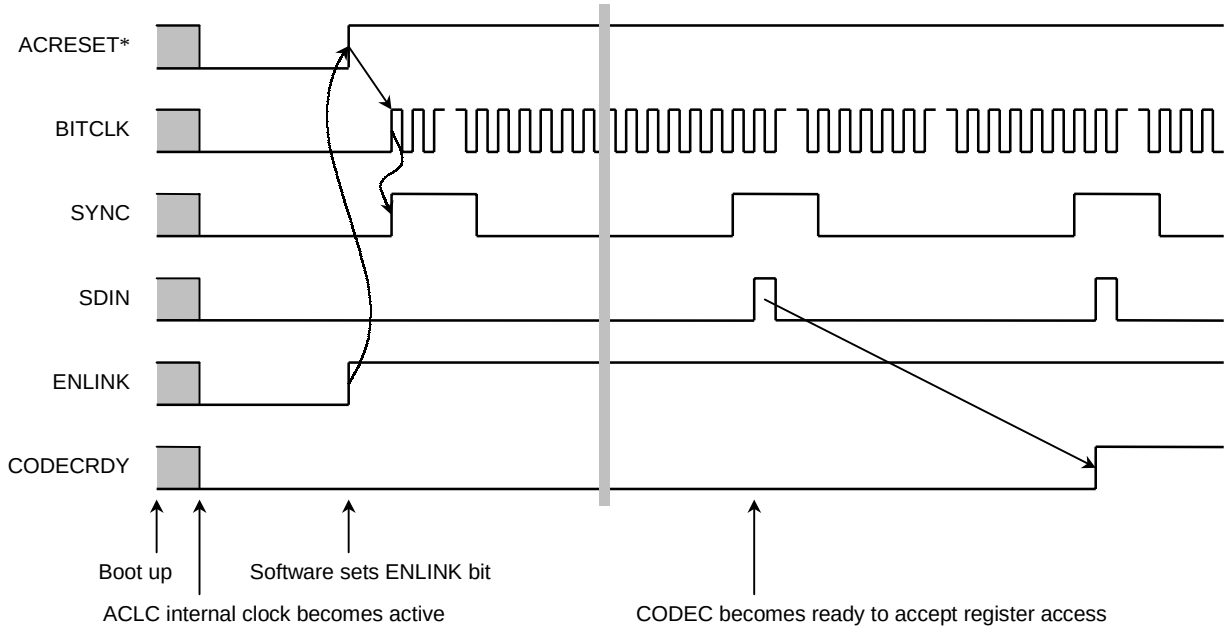
The AC'97 specification requires that the reset assertion period is 1μs or longer. The software is responsible for controlling the length of this period.

The AC'97 specification also requires that the primary CODEC stops the AC-link clock (BITCLK) signal during the period from ACRESET* signal assertion to 162.8ns after ACRESET* signal deassertion. ACLC assumes the primary CODEC meet this requirement.

Deasserting the link-side reset makes the primary CODEC start driving the BITCLK signal. When the BITCLK signal is provided, ACLC starts the SYNC signal output, which indicates the start of the AC-link frame, and starts the frame-length counting.

When a CODEC becomes ready to receive access to its own register, the CODEC sets the "CODEC Ready" bit of the Tag slot. When ACLC detects that this bit has been set, the ACLC Interrupt Status Register (ACINTSTS)'s CODEC[1:0] Ready (CODEC[1:0]RDY) bit is set. The system software is able to recognize the readiness of the CODEC(s) by detecting this event by way of either polling or interrupt.

In case of 5.1 channel audio connection example (Figure 14.3.2), because the secondary CODEC is connected to the SDIN1 signal of ACLC, the software must watch ACINTSTS.CODEC1RDY bit to determine the CODEC's readiness for the register access.



Note: The number of BITCLK cycles relative to other signals is not to scale.

Figure 14.3.5 Cold Reset and CODEC Ready Recognition

14.3.5 CODEC Register Access

By accessing registers in the CODEC, the system software is able to detect or control the CODEC state. This section describes how to read and write CODEC registers via ACLC. For details about AC'97 register set and proper sequence to operate CODEC, refer to the AC'97 specification and target CODEC datasheet.

It takes several frame periods for a read or write access to complete. Taking this into account, ACLC is equipped with a function for reporting CODEC register access completion as status-change or interrupt.

In order to read an AC'97 register, write the access destination CODEC ID and register address in ACLC CODEC Register Access Register (ACREGACC) with its CODECRD bit set to "1". After the ACLC Interrupt Status Register (ACINTSTS)'s REGACC Ready (REGACCRDY) bit is set, the software is able to get the data returned from the AC'97 by reading the ACREGACC register and issue another access.

In order to write to an AC'97 register, write the access destination CODEC ID, register address, and the data in ACLC's ACREGACC register with ACREGACC.CODECRD bit set to "0". After the ACINTSTS.REGACCRDY bit has been set, the software is able to issue another access.

In case of 5.1 channel audio connection example (Figure 14.3.2), because the secondary CODEC has CODEC ID of '3', the software must write '3' into ACREGACC.CODECID field when it issues secondary CODEC register access.

14.3.6 Sample-data Transmission and Reception

This section describes the mechanism for transmission and reception of PCM audio and modem wave-data. An overview is described first. The DMA (Direct Memory Access) operation, error detection and recovery procedure follow. A special case using slot activation control is described last.

14.3.6.1 Overview

Figure 14.3.6 and Figure 14.3.7 show conceptual views of the sample-data transmission and reception mechanisms.

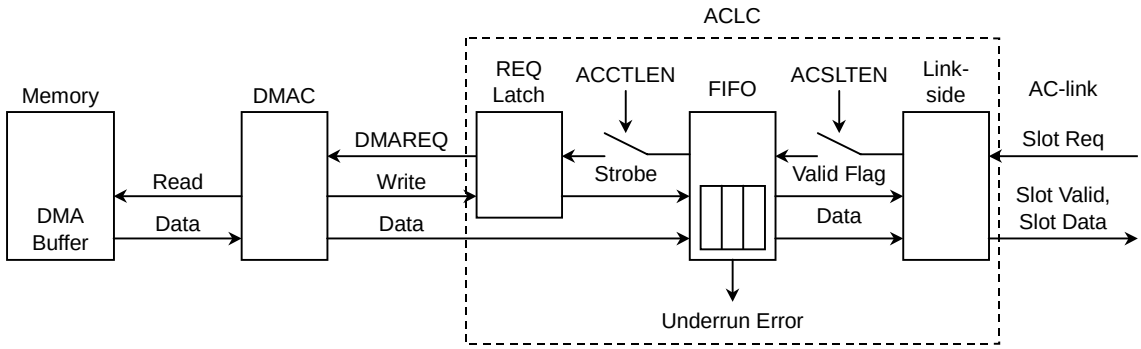


Figure 14.3.6 Sample-data Transmission Mechanism

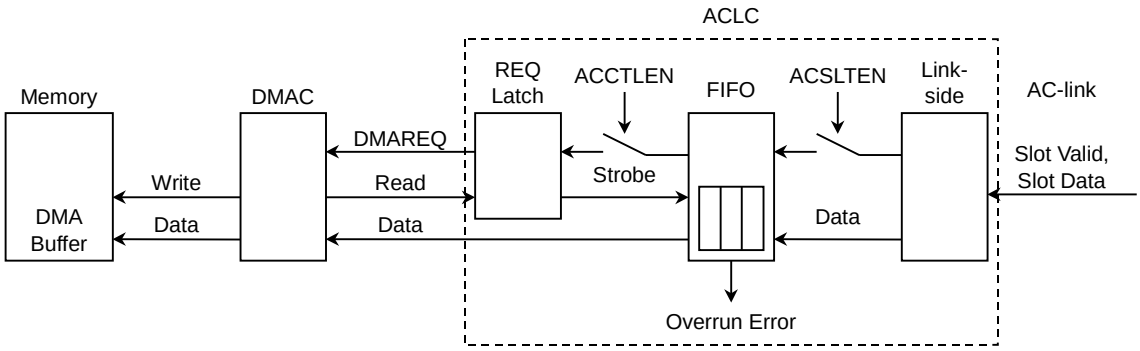


Figure 14.3.7 Sample-data Reception Mechanism

The CODEC requests ACLC to transmit and receive sample-data via ‘slot-request’ and ‘slot-valid’ bit-fields on the SDIN signal of AC-link.

For transmission, ACLC transmits the data with ‘slot-valid’ tag set. For reception, ACLC captures the slot-data.

Transmission or reception through each stream can be independently activated or deactivated under control of ACLC Slot Enable Register (ACSLTEN).

ACLC is equipped with a separate FIFO for each data-stream. The data to transmit is prefetched from memory to FIFO through DMA. The received data is buffered on FIFO and then stored to memory through DMA. In this stage, each DMA is independently activated or deactivated under control of ACLC Control Enable Register (ACCTLEN).

14.3.6.2 DMA Channel Mapping

ACLC uses four DMA request channels. These DMA channels are allocated to four out of seven data-streams, or slots, on the AC-link frame, according to ACLC DMA Channel Selection Register (ACDMASEL) setting as shown in Table 14.3.1. The pin configuration register allocates these DMA channels of ACLC to the DMAC (DMA controller) channels according to DMA Request Control Register (DRQCTR)'s DMA Request Selection (DMAREQ0-3) bits as described in section 5.2.8.

Table 14.3.1 DMA Channel Mapping Modes

AC-link Slot Number	ACDMASEL			
	0	1	2	3
PCM L&R out (3&4)	ACLC ch0	ACLC ch0	ACLC ch0	ACLC ch0
Surround L&R out (7&8)		ACLC ch1	ACLC ch1	ACLC ch1
Center out (6)			ACLC ch2	
LFE out (9)			ACLC ch3	ACLC ch3
PCM L&R in (3&4) or Mic in (6)	ACLC ch1			ACLC ch2
Modem Line1 out (5)	ACLC ch2	ACLC ch2		
Modem Line 1 in (5)	ACLC ch3	ACLC ch3		

14.3.6.3 Sample-data Format

ACLC transmits/receives 16 bits per sample for each data slot shown in Table 14.3.1. The data resides on the first 16 bits of the 20 bits assigned to each slot on AC-link. Each sample-data register allows access by word (32-bit) unit only. Therefore the DMA count must be a multiple of word. Note that the transmit-data DMA count also must be the FIFO depth (refer to Table 14.3.8) or more for a reason described later.

For audio PCM front and surround streams, every data-word is loaded with a couple of left and right samples. For audio MIC stream, valid data is loaded in the same field as the left sample while the other field is filled with '0'. For audio center, LFE, and modem line 1 streams, two consecutive samples are packed into every word.

The data format at the sample-data register is arranged so that the data format on the DMA buffer follows the rules below.

- Each sample data is put in the byte order in which the CPU operates (big- or little-endian).
- Samples are put in the time-sequential order at increasing addresses on memory.
- For a DMA channel which couples left and right samples, each left sample precedes the corresponding right sample.

Refer to the sections 14.4.16 and later for the register format.

Figures below show the format of DMA buffer for each type of DMA channel. #0, #1, … means the sample's sequential number for the AC-link slot. Subscript 'L' means lower 8-bit of each sample and subscript 'H' means upper 8-bit.

Table 14.3.2 Front and Surround DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_L	Left#0_H	Right#0_L	Right#0_H
+4	Left#1_L	Left#1_H	Right#1_L	Right#1_H
+8	Left#2_L	Left#2_H	Right#2_L	Right#2_H
:	:	:	:	:

Table 14.3.3 Center, LFE, and Modem DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	#1_L	#1_H
+4	#2_L	#2_H	#3_L	#3_H
+8	#4_L	#4_H	#5_L	#5_H
:	:	:	:	:

Table 14.3.4 Mic DMA Buffer Format in Little-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_L	#0_H	0	0
+4	#1_L	#1_H	0	0
+8	#2_L	#2_H	0	0
:	:	:	:	:

Table 14.3.5 Front and Surround DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	Left#0_H	Left#0_L	Right#0_H	Right#0_L
+4	Left#1_H	Left#1_L	Right#1_H	Right#1_L
+8	Left#2_H	Left#2_L	Right#2_H	Right#2_L
:	:	:	:	:

Table 14.3.6 Center, LFE, and Modem DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
+0	#0_H	#0_L	#1_H	#1_L
+4	#2_H	#2_L	#3_H	#3_L
+8	#4_H	#4_L	#5_H	#5_L
:	:	:	:	:

Table 14.3.7 Mic DMA Buffer Format in Big-endian Mode

Address offset	+0	+1	+2	+3
z	#0_H	#0_L	0	0
+4	#1_H	#1_L	0	0
+8	#2_H	#2_L	0	0
:	:	:	:	:

14.3.6.4 DMA Operation

When ACLC's REQ latch (refer to Figure 14.3.6 and Figure 14.3.7) needs to read or write sample-data, it issues a DMA request. When DMAC acknowledges the request by performing write- or read-access to the ACLC sample-data register, ACLC deasserts the request. Therefore, the software must properly set up DMAC so that the source or destination points to the corresponding sample-data register for the DMA channel.

Setup the DMA Channel Control Registers (DMCCRn) in DMAC as follows.

Immediate chain	Enable	DMCCRn.IMMCHN = 1 [Note]
DMA request polarity	Low-active	DMCCRn.REQPOL = 0
DMA acknowledge polarity	Low-active	DMCCRn.ACKPOL = 0
Request sense	Level-sensitive	DMCCRn.EGREQ = 0
Simple chain	Enable	DMCCRn.SMPCHN = 1
Transfer size	1 word	DMCCRn.XFSZ = 010b
Transfer address mode	Dual	DMCCRn.SNGAD = 0

Note: Use this setting when DMA chain operation is utilized

For a transmission channel, assign the address of ACLC Audio PCM Output/Surround/Center/LFE/Modem Output Register (ACAUDO/SURR/CENT/LFE/MODODAT) to the DMAC destination address register (DMDARn). For a reception channel, assign the address of ACLC Audio input/Modem Input Register (ACAUDI/MODIDAT) to the DMAC source address register (DMSARn).

When any DMA request is pending, the REQ latch will not deasserted the request until the corresponding sample-data register is accessed. Just unsetting ACLC Control Enable Register (ACCTLEN)'s DMA Enable (xxxxDMA) bit corresponding to the DMA will not clear the REQ latch.

The procedure to continuously push or pull the sample-data stream through the chain DMA operation follows the DMAC specification. Refer to section 8.3.10 for this respect.

14.3.6.5 Sample-data FIFO

For a transmission stream, as long as ACLC Control Enable Register (ACCTLEN) allows that transmission and the FIFO has any room to fill data in, the FIFO issues a request via the REQ latch. On the other side, when a transmission FIFO receives a data-request from the link-side, it provides data with valid-flag set if it has any valid data. If it has no valid data, it responds with valid-flag unset and an underrun error bit is set.

At the transmit-data DMA start-up, until the FIFO becomes full, it responds to the link-side with valid-flag unset, in order to maximize the buffering effect. Therefore, the DMA size must be the FIFO depth or more.

Table 14.3.8 Transmission FIFO Depth

Data-stream	FIFO Depth (Word)
PCM L&R out	3
Surround L&R out	3
Center out	2
LFE out	2
Modem Line 1 out	1

The link-side drives the slot-valid bit and slot-data on AC-link. When underrun occurs, these bits are driven to all '0'.

For a reception stream, as long as the FIFO has any valid data, the FIFO issues a request via the REQ latch. On the other side, when ACCTLEN allows that reception and the link-side issues a data strobe, the FIFO stores the valid data. If the FIFO is full when it receives a data strobe, the data is discarded and an overrun error bit is set.

14.3.6.6 Error Detection and Recovery

In most usages, since the CODEC continuously requests sample-data transmission and reception, after DMA is finished, underrun and overrun will occur. The procedure described below allows the software to determine whether an error has occurred during DMA operation.

The software sets ACLC Control Enable Register (ACCTLEN)'s Error Halt Enable (xxxxEHLT) bit before it starts a DMA channel. After it starts the DMA channel, it waits until ACLC Interrupt Status Register (ACINTSTS)'s Underrun or Overrun Error (xxxxERR) bit is set. When the event is detected, the software checks DMA Channel Control Register (DMCCRn)'s Transfer Active (XFACT) bit and ACLC DMA Request Status Register (ACDMASTS)'s Request (xxxxDMA) bit and determines the DMA completion status as follows.

Table 14.3.9 DMA Completion Status Determination

DMCCRn.XFACT	ACDMASTS.xxxxDMA	Completion Status
Inactive	Pending	No Error during DMA
Inactive	Not Pending	Underrun or Overrun
Active	*	Underrun or Overrun

To recover from error, disable and enable the stream via ACCTLEN, and restart the DMA.

14.3.6.7 Slot Activation Control

In case ACLC is required to begin transmission or reception of multiple streams at the same time, slot activation control will be useful. To use this feature, the software must deactivate the relevant streams first, enable ACLC Control Enable Register (ACCTLEN), make sure the transmission FIFO becomes full by checking ACLC FIFO Status Register (ACFIFOSTS)'s Full (xxxxFULL) bit, and finally enable ACLC Slot Enable Register (ACSLTEN). This procedure assures that all the reception streams are activated at a frame and all the transmission streams begin to respond to the slot-request bits of that frame.

Note that access to ACSLTEN and ACLC Slot Disable Register (ACSLTDIS) needs special care to synchronize with the link-side. Refer to the register description for detail.

Since operating ACCTLEN register and DMAC without touching ACSLTEN is sufficient for most usages, the initial ACSLTEN value enables all the transmission and reception through the slots by default.

14.3.6.8 Variable Rate Limitation

To improve compatibility with existing AC'97 CODECs and controllers on the market, ACLC combines sample-data for the slots 3 and 4 into one DMA channel, and similarly for the slots 7 and 8. This feature effectively considers that the slot request bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame, and similarly for the slots 7 and 8. ACLC also considers that the slot valid bit from the CODEC for slot 4 shall be always same (in tandem) as for slot 3 for each frame.

14.3.7 GPIO Operation

ACLC supports the slot 12 for the MC'97 (Modem Codec) GPIO.

The slot 12 is shadowed in the ACLC GPI Data Register (ACGPIDAT) and ACLC GPO Data Register (ACGPODAT) in the following way:

- ACLC copies the slot 12 input data into the ACGPIDAT register, if the slot 12 input is marked by the CODEC as valid in the AC-link frame period.
- ACLC generates the slot 12 output data from the ACGPODAT register and mark it as valid, if the slot 12 is required from the CODEC in the previous AC-link frame.

This shadowing function is enabled as long as ACSLTEN allows.

The bit 0 of the slot 12 is defined as 'GPIO_INT' and can cause ACLC to request an interrupt.

14.3.8 Interrupt

ACLC generate two kinds of interrupt to the interrupt controller as below.

- ACLC Interrupt

  Logical OR of all the valid bits of ACLC Interrupt Masked Status Register (ACINTMSTS) is connected. Refer to section 14.4.5.

- ACLCPME Interrupt

  This interrupt shows the wake-up from CODEC in AC-link low-power mode.

  Refer to the description for ACLC Control Enable Register (ACCTLEN)'s Wake-up Enable (WAKEUP) bit in section 14.4.1.

14.3.9 AC-link Low-power Mode

The AC'97 specification makes provision for saving power during system suspension by powering-down both the controller and CODEC except the minimum circuit to detect modem RING/Caller-ID event and wake up the system. AC'97 CODEC is required to go into the low-power mode when they receive a special register-write access. In this mode, the AC-link controller must drive all output signals to low level to allow the CODEC digital I/O power cut.

ACLC provides 'AC-link low-power mode' setting. When this mode is enabled by ACLC Control Enable Register (ACCTLEN)'s Enable AC-link Low-power Mode (LOWPWR) bit, all the output signals except the ACRESET* signal to the AC-link are forced to low level.

The AC-link will be reactivated out of the low-power mode when the SYNC signal is driven high for 1 μs or longer by the AC-link controller while the BITCLK signal is inactive. The software is responsible for controlling the length of this period.

ACLC also provides the 'wake-up' function. While this function is enabled by ACCTLEN Register's Enable Wake-up (WAKEUP) bit, high-level input at any SDIN[x] signal will force ACLCPME interrupt assertion.

When ACLCPME interrupt is recognized, the software must disable the low-power mode and assert warm reset to the AC-link via ACCTLEN Register's Enable Warm Reset (WRESET) bit. After the warm reset is deasserted, the CODEC will start providing the BITCLK signal, and then ACLC will generate the SYNC signal for usual AC-link frames.

Refer to section B.5.1 of AC'97 specification revision 2.1 for the power-down and wake-up sequence in AC-link power-down mode.

14.4 Registers

The base address for the ACLC registers is described in section 5.1.7. Only word (32-bit) accesses are allowed. These registers return to their initial values when the module gets reset by power-on or configuration-register operation. The 'Disable AC-link' operation initializes the ACREGACC, ACGPIDAT, ACGPODAT, and ACSLTEN registers while keeping the other registers.

Do not access any location which is not mentioned in this section.

All the register bits marked as 'Reserved' are reserved. The value of the reserved bit when read is undefined. When any register is written, write to the reserved bit(s) the same value as the previous value read.

Table 14.4.1 ACLC Registers

Reference	Offset Address	Bit Width	Mnemonic	Register Name
14.4.1	0xF700	32	ACCTLEN	ACLC Control Enable Register
14.4.2	0xF704	32	ACCTLDIS	ACLC Control Disable Register
14.4.3	0xF708	32	ACREGACC	ACLC CODEC Register Access Register
14.4.4	0xF710	32	ACINTSTS	ACLC Interrupt Status Register
14.4.5	0xF714	32	ACINTMSTS	ACLC Interrupt Masked Status Register
14.4.6	0xF718	32	ACINTEN	ACLC Interrupt Enable Register
14.4.7	0xF71C	32	ACINTDIS	ACLC Interrupt Disable Register
14.4.8	0xF720	32	ACSEMAPH	ACLC Semaphore Register
14.4.9	0xF740	32	ACGPIDAT	ACLC GPI Data Register
14.4.10	0xF744	32	ACGPODAT	ACLC GPO Data Register
14.4.11	0xF748	32	ACSLTEN	ACLC Slot Enable Register
14.4.12	0xF74C	32	ACSLTDIS	ACLC Slot Disable Register
14.4.13	0xF750	32	ACFIFOSTS	ACLC FIFO Status Register
14.4.14	0xF780	32	ACDMASTS	ACLC DMA Request Status Register
14.4.15	0xF784	32	ACDMASEL	ACLC DMA Channel Selection Register
14.4.16	0xF7A0	32	ACAUDODAT	ACLC Audio PCM Output Data Register
14.4.16	0xF7A4	32	ACSURRDAT	ACLC Surround Data Register
14.4.17	0xF7A8	32	ACCENTDAT	ACLC Center Data Register
14.4.17	0xF7AC	32	ACLFEDAT	ACLC LFE Data Register
14.4.18	0xF7B0	32	ACAUDIDAT	ACLC Audio PCM Input Data Register
14.4.17	0xF7B8	32	ACMODODAT	ACLC Modem Output Data Register
14.4.19	0xF7BC	32	ACMODIDAT	ACLC Modem Input Data Register
14.4.20	0xF7FC	32	ACREVID	ACLC Revision ID Register

14.4.1 ACLC Control Enable Register (ACCTLEN) 0xF700

This register is used to check the setting of various ACLC features and to enable them.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								MODIEHLT	MODOEHLT	Reserved	AUDIEHLT	LFEEHLT	CENTEHLT	SURREHLT	AUDOEHLT
								R/W1S	R/W1S		R/W1S	R/W1S	R/W1S	R/W1S	R/W1S : Type
								0	0		0	0	0	0	0 : Initial value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODIDMA	MODODMA	Reserved	AUDIDMA	LFEDMA	CENTDMA	SURRDMA	AUDODMA	Reserved		RDYCLR	MICSEL	WRESET	WAKEUP	LOWPWR	ENLINK
R/W1S	R/W1S		R/W1S	R/W1S	R/W1S	R/W1S	R/W1S			W1C	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S : Type
0	0		0	0	0	0	0			—	0	0	0	0	0 : Initial value

Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23	MODIEHLT	Enable Modem Receive-data DMA Error Halt	Enable Modem Receive-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that MODIDMA error halt is disabled. 1: Indicates that MODIDMA error halt is enabled. W1S 0: No effect 1: Enables MODIDMA error halt. When MODIDMA overrun occurs, subsequent DMA will not be issued.
22	MODOEHLT	Enable Modem Transmit-data DMA Error Halt	Enable Modem Transmit-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that MODODMA error halt is disabled. 1: Indicates that MODODMA error halt is enabled. W1S 0: No effect 1: Enables MODODMA error halt. When MODODMA underrun occurs, subsequent DMA will not be issued.
21	—	Reserved	—
20	AUDIEHLT	Enable Audio Receive-data DMA Error Halt	Enable Audio Receive-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that AUDIDMA error halt is disabled. 1: Indicates that AUDIDMA error halt is enabled. W1S 0: No effect 1: Enables AUDIDMA error halt. When AUDIDMA overrun occurs, subsequent DMA request will not be issued.
19	LFEEHLT	Enable Audio LFE Transmit-data DMA Error Halt	Enable Audio LFE Transmit-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that LFEDMA error halt is disabled. 1: Indicates that LFEDMA error halt is enabled. W1S 0: No effect 1: Enables LFEDMA error halt. When LFEDMA underrun occurs, subsequent DMA request will not be issued.
18	CENTEHLT	Enable Audio Center Transmit-data DMA Error Halt	Enable Audio Center Transmit-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that CENTDMA error halt is disabled. 1: Indicates that CENTDMA error halt is enabled. W1S 0: No effect 1: Enables CENTDMA error halt. When CENTDMA underrun occurs, subsequent DMA request will not be issued.

Figure 14.4.1 ACCTLEN Register (1/3)

Bits	Mnemonic	Field Name	Description
17	SURREHLT	Enable Audio Surround L&R Transmit-data DMA Error Halt	Enable Audio Surround L&R Transmit-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that SURRDMA error halt is disabled. 1: Indicates that SURRDMA error halt is enabled. W1S 0: No effect 1: Enables SURRDMA error halt. When SURRDMA underrun occurs, subsequent DMA request will not be issued.
16	AUDOEHLT	Enable Audio PCM L&R Transmit-data DMA Error Halt	Enable Audio PCM L&R Transmit-data DMA Error Halt. (Initial value: 0, R/W1S) R 0: Indicates that AUDODMA error halt is disabled. 1: Indicates that AUDODMA error halt is enabled. W1S 0: No effect 1: Enables AUDODMA error halt. When AUDODMA underrun occurs, subsequent DMA request will not be issued.
15	MODIDMA	Enable Modem Receive-data DMA	Enable Modem Receive-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that modem receive-data DMA is disabled. 1: Indicates that modem receive-data DMA is enabled. W1S 0: No effect 1: Enables modem receive-data DMA.
14	MODODMA	Enable Modem Transmit-data DMA	Enable Modem Transmit-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that modem transmit-data DMA is disabled. 1: Indicates that modem transmit-data DMA is enabled. W1S 0: No effect 1: Enables modem transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]
13	—	Reserved	—
12	AUDIDMA	Enable Audio Receive-data DMA	Enable Audio Receive-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that audio receive-data DMA is disabled. 1: Indicates that audio receive-data DMA is enabled. W1S 0: No effect 1: Enables audio receive-data DMA.
11	LFEDMA	Enable Audio LFE Transmit-data DMA	Enable Audio LFE Transmit-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that audio LFE transmit-data DMA is disabled. 1: Indicates that audio LFE transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio LFE transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]
10	CENTDMA	Enable Audio Center Transmit- data DMA	Enable Audio Center Transmit-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that audio Center transmit-data DMA is disabled. 1: Indicates that audio Center transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Center transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]
9	SURRDMA	Enable Audio Surround L&R Transmit-data DMA	Enable Audio Surround L&R Transmit-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that audio Surround L&R transmit-data DMA is disabled. 1: Indicates that audio Surround L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio Surround L&R transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]
8	AUDODMA	Enable Audio PCM L&R Transmit-data DMA	Enable Audio PCM L&R Transmit-data DMA. (Initial value: 0, R/W1S) R 0: Indicates that audio PCM L&R transmit-data DMA is disabled. 1: Indicates that audio PCM L&R transmit-data DMA is enabled. W1S 0: No effect 1: Enables audio PCM L&R transmit-data DMA. [Note: DMA size must be internal FIFO depth or more.]

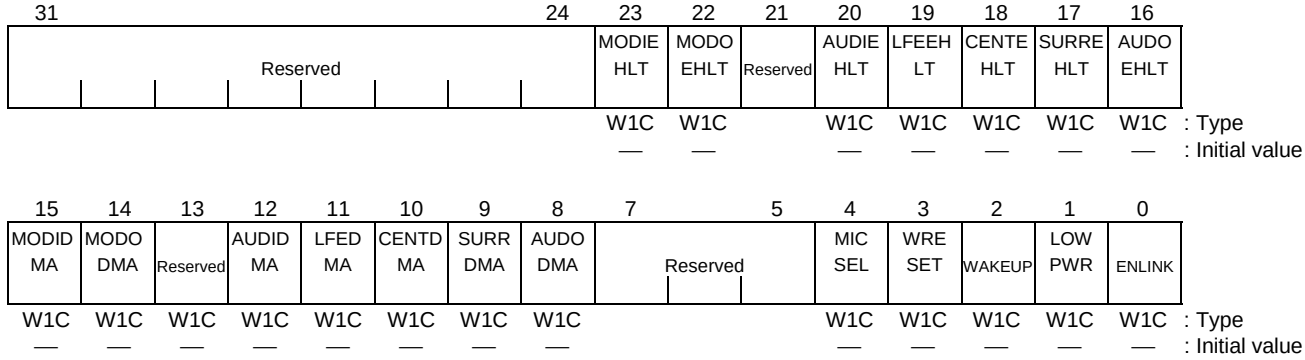
Figure 14.4.1 ACCTLEN Register (2/3)

Bits	Mnemonic	Field Name	Description
7:6	—	Reserved	—
5	RDYCLR	Clear CODEC Ready Bit	Clear CODEC Ready Bit (Initial value: 0, W1S) W1C 0: No effect 1: Clear CODEC[1:0] ready bits Note: This bit should only be written to reevaluate the CODEC ready status after power-down command is sent to CODEC.
4	MICSEL	MIC Selection	MIC Selection. (Initial value: 0, R/W1S) R 0: Indicates that PCM L&R (Slot 3&4) is selected for audio reception. 1: Indicates that MIC (Slot 6) is selected for audio reception. W1S 0: No effect 1: Selects MIC (Slot 6) for audio reception.
3	WRESET	Assert Warm Reset	Assert Warm Reset. (Initial value: 0, R/W1S) R 0: Indicates that warm reset is not asserted. 1: Indicates that warm reset is asserted. W1S 0: No effect 1: Asserts warm reset. Note 1: Do not assert warm reset during normal operation. Note 2: The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).
2	WAKEUP	Enable Wake-up	Enable Wake-up. (Initial value: 0, R/W1S) R 0: Indicates that wake-up from low-power mode is disabled. 1: Indicates that wake-up from low-power mode is enabled. While any SDIN signal is driven high, ACLC asserts ACLCPME interrupt request to the interrupt controller. W1S 0: No effect 1: Enables wake-up from low-power mode. Note: Do not enable wake-up during normal operation.
1	LOWPWR	Enable AC-link low-power mode	Enable AC-link Low-power Mode. (Initial value: 0, R/W1S) R 0: SYNC and SDOUT signals are not forced to low. 1: SYNC and SDOUT signals are forced to low. W1S 0: No effect 1: Forces SYNC and SDOUT signals low. Note: Do not enable AC-link low-power mode during normal operation.
0	ENLINK	Enable AC-link	Enable AC-link. (Initial value: 0, R/W1S) R 0: Indicates that the ACRESET* signal to AC-link is asserted. 1: Indicates that the ACRESET* signal to AC-link is not asserted. W1S 0: No effect 1: Deasserts the ACRESET* signal to AC-link Note: The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).

Figure 14.4.1 ACCTLEN Register (3/3)

14.4.2 ACLC Control Disable Register (ACCTLDIS) 0xF704

This register is used to disable various ACLC features.



Bits	Mnemonic	Field Name	Description
31:24	—	Reserved	—
23	MODIEHLT	Disable Modem Receive-data DMA Error Halt	Disable Modem Receive-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables MODIDMA error halt. MODIDMA request(s) will continue to be issued even after MODIDMA overrun occurs.
22	MODOEHLT	Disable Modem Transmit-data DMA Error Halt	Disable Modem Transmit-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables MODODMA error halt. MODODMA request(s) will continue to be issued even after MODODMA underrun occurs.
21	—	Reserved	—
20	AUDIEHLT	Disable Audio Receive-data DMA Error Halt	Disable Audio Receive-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables AUDIDMA error halt. AUDIDMA request(s) will continue to be issued even after AUDIDMA overrun occurs.
19	LFEEHLT	Disable Audio LFE Transmit-data DMA Error Halt	Disable Audio LFE Transmit-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables LFEDMA error halt. LFEDMA request(s) will continue to be issued even after LFEDMA underrun occurs.
18	CENTEHLT	Disable Audio Center Transmit-data DMA Error Halt	Disable Audio Center Transmit-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables CENTDMA error halt. CENTDMA request(s) will continue to be issued even after CENTDMA underrun occurs.
17	SURREHLT	Disable Audio Surround L&R Transmit-data DMA Error Halt	Disable Audio Surround L&R Transmit-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables SURRDMA error halt. SURRDMA request(s) will continue to be issued even after SURRDMA underrun occurs.
16	AUDOEHLT	Disable Audio PCM L&R Transmit-data DMA Error Halt	Disable Audio PCM L&R Transmit-data DMA Error Halt. (Initial value: –, W1C) W1C 0: No effect 1: Disables AUDODMA error halt. AUDODMA request(s) will continue to be issued even after AUDODMA underrun occurs.
15	MODIDMA	Disable Modem Receive-data DMA	Disable Modem Receive-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables modem receive-data DMA.
14	MODODMA	Disable Modem Transmit-data DMA	Disable Modem Transmit-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables modem transmit-data DMA.

Figure 14.4.2 ACCTLDIS Register (1/2)

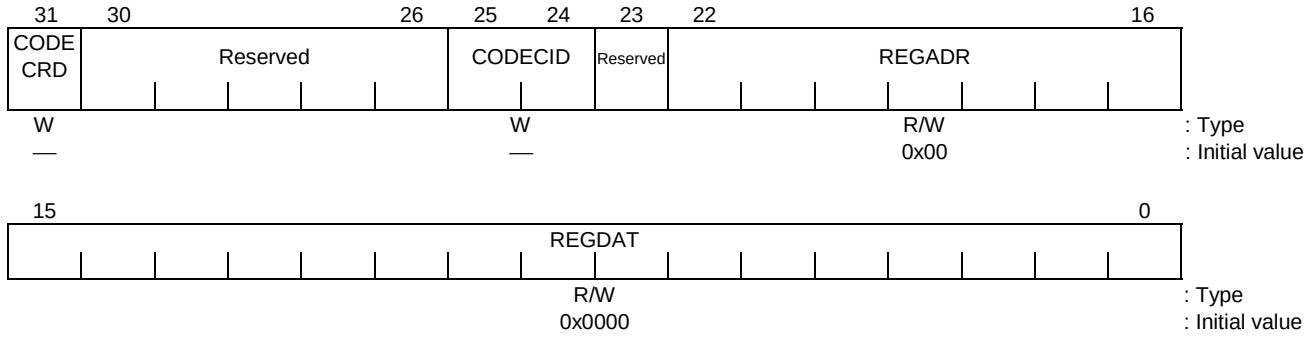
Bits	Mnemonic	Field Name	Description
13	—	Reserved	—
12	AUDIDMA	Disable Audio Receive-data DMA	Disable Audio Receive-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio receive-data DMA.
11	LFEDMA	Disable Audio LFE Transmit-data DMA	Disable Audio LFE Transmit-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio LFE transmit-data DMA.
10	CENTDMA	Disable Audio Center Transmit-data DMA	Disable Audio Center Transmit-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio Center transmit-data DMA.
9	SURRDMA	Disable Audio Surround L&R Transmit-data DMA	Disable Audio Surround L&R Transmit-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio Surround L&R transmit-data DMA.
8	AUDODMA	Disable Audio PCM L&R Transmit-data DMA	Disable Audio PCM L&R Transmit-data DMA. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio PCM L&R transmit-data DMA.
7:5	—	Reserved	—
4	MICSEL	MIC Selection	MIC Selection (Initial value: –, W1C) W1C 0: No effect 1: Selects PCM L&R (Slot 3&4) for audio reception
3	WRESET	Deassert Warm Reset	Deassert Warm Reset. (Initial value: –, W1C) W1C 0: No effect 1: Deasserts warm reset. Note: The software must guarantee the warm reset assertion time meets the AC'97 specification (1.0 μs or more).
2	WAKEUP	Disable Wake-up.	Disable Wake-up. (Initial value: –, W1C) W1C 0: No effect 1: Disables wake-up from low-power mode.
1	LOWPWR	Disable AC-link Low-power Mode.	Disable AC-link Low-power Mode. (Initial value: –, W1C) W1C 0: No effect 1: Releases SYNC and SDOUT signals from low.
0	ENLINK	Disable AC-link	Disable AC-link. (Initial value: –, W1C) W1C 0: No effect 1: Asserts the ACRESET* signal to AC-link. Note: The software must guarantee the ACRESET* signal assertion time meets the AC'97 specification (1.0 μs or more).

Figure 14.4.2 ACCTLDIS Register (2/2)

Clear xxxxDMA bits in ACCTLEN to "0" by using this register to disable transmit/receive-data DMA and to stop transmission/reception by the AC-link. Note that if these bits are cleared while output-slot data is flowing in the FIFO, ACLC may output a wrong data as the last sample. This behavior will not occur if the software waits for data-flow completion by detecting underrun before it disables the corresponding slot.

14.4.3 ACLC CODEC Register Access Register (ACREGACC) 0xF708

CODEC registers can be accessed through this register.



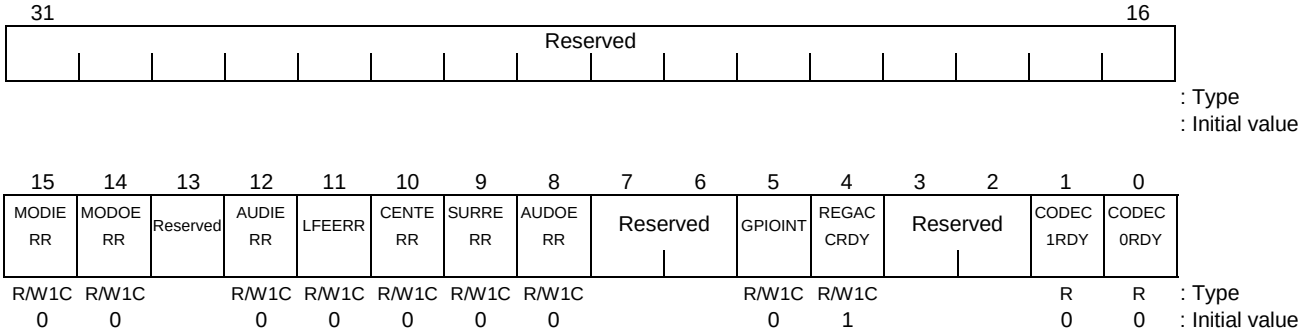
Bits	Mnemonic	Field Name	Description
31	CODECRD	AC'97 register read access	AC'97 register read access (Initial value: –, W) W 0: Indicates a write access. 1: Indicates a read access.
30:26	—	Reserved	—
25:24	CODECID	AC'97 CODEC ID	AC'97 CODEC ID (Initial value: –, W) W Specifies the CODEC ID of the read/write access destination. The values "0" through "3" can be specified as the CODEC ID, but the number of CODECs actually supported depends on the configuration.
23	—	Reserved	—
22:16	REGADR	AC'97 register address	AC'97 register address (Initial value: 0x00, R/W) R Read address. Valid address can be read after read access is complete. W Specifies the read/write access destination address.
15:0	REGDAT	AC'97 register data	AC'97 register data (Initial value: 0x0000, R/W) R Read data. Valid data can be read after read access is complete. W Write data.

Figure 14.4.3 ACREGACC

This register must not be read from or written to until access completion is reported through the ACINTSTS register.

14.4.4 ACLC Interrupt Status Register (ACINTSTS) 0xF710

This register shows various kinds of AC-link and ACLC status.



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15	MODIERR	Modem Receive-data DMA Overrun	Modem Receive-data DMA Overrun (Initial value: 0, R/W1C) R 1: Indicates that the modem receive-data DMA overran. W1C This bit is cleared when “1” is written to it.
14	MODOERR	Modem Transmit-data DMA Underrun	Modem Transmit-data DMA Underrun (Initial value: 0, R/W1C) R 1: Indicates that the modem transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.
13	—	Reserved	—
12	AUDIERR	Audio Receive-data DMA Overrun	Audio Receive-data DMA Overrun (Initial value: 0, R/W1C) R 1: Indicates that the audio receive-data DMA overran. W1C This bit is cleared when “1” is written to it.
11	LFEERR	Audio LFE Transmit-data DMA Underrun	Audio LFE Transmit-data DMA Underrun (Initial value: 0, R/W1C) R 1: Indicates that the audio LFE transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.
10	CENTERR	Audio Center Transmit-data DMA Underrun	Audio Center Transmit-data DMA Underrun (Initial value: 0, R/W1C) R 1: Indicates that the audio center transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.
9	SURRERR	Audio Surround L&R Transmit-data DMA Underrun	Audio Surround L&R Transmit-data DMA Underrun (Initial value: 0, R/W1C) R 1: Indicates that the audio surround L&R transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.
8	AUDOERR	Audio PCM L&R Transmit-data DMA Underrun	Audio PCM L&R Transmit-data DMA Underrun (Initial value: 0, R/W1C) R 1: Indicates that the audio PCM L&R transmit-data DMA underran. W1C This bit is cleared when “1” is written to it.
7:6	—	Reserved	—
5	GPIOINT	GPIO Interrupt	GPIO Interrupt (Initial value: 0, R/W1C) R 1: Indicates that the incoming slot 12 bit[0] is ‘1’ (the modem CODEC GPIO interrupt). W1C This bit is cleared if “1” is written to it while the incoming slot 12 bit[0] is ‘0’.

Figure 14.4.4 ACINTSTS Register (1/2)

Bits	Mnemonic	Field Name	Description
4	REGACCRDY	ACREGACC Ready	ACREGACC Ready (Initial value: 1, R/W1C) R 1: Indicates that the ACREGACC register is ready to get the value (in case the previous operation was a read access) and to initiate another R/W access to an AC'97 register. The result of reading or writing to the ACREGACC register before the completion notification is undefined. W1C This bit is cleared if "1" is written to it. This bit automatically becomes '0' when the ACREGACC register is written.
3:2	—	Reserved	—
1	CODEC1RDY	CODEC1 Ready	CODEC1 Ready (Initial value: 0, R) R 1: Indicates that the CODEC Ready bit of SDIN1 Slot0 is set.
0	CODEC0RDY	CODEC0 Ready	CODEC0 Ready (Initial value: 0, R) R 1: Indicates that the CODEC Ready bit of SDIN0 Slot0 is set.

Figure 14.4.4 ACINTSTS Register (2/2)

14.4.5 ACLC Interrupt Masked Status Register (ACINTMSTS) 0xF714

Every bit in this register is configured as follows:

ACINTMSTS = ACINTSTS & ACINTEN

Bit placement is the same as for the ACINTSTS register. The logical OR of all bits in this register is used as ACLC interrupt request to the interrupt controller.

14.4.6 ACLC Interrupt Enable Register (ACINTEN) 0xF718

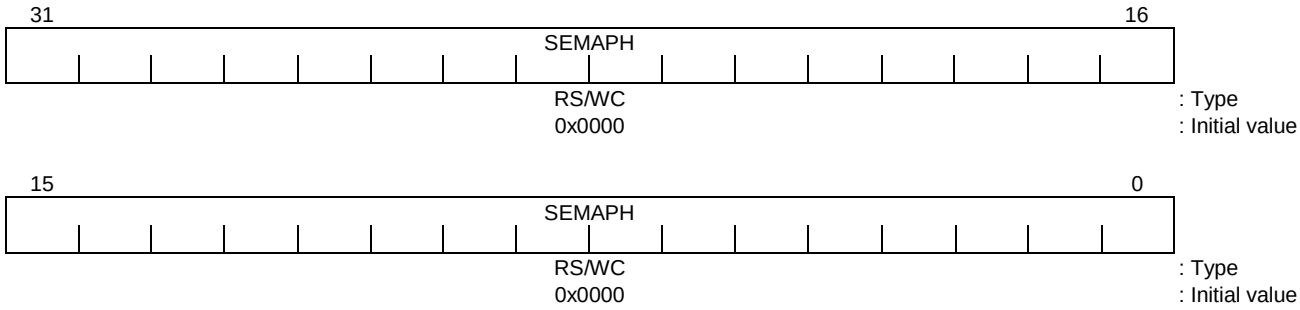
Interrupt request enable (R/W1S). Bit placement is the same as for the ACINTSTS register. Its initial value is all '0'. When a value is written to this register, the bit in the position where "1" was written is set to "1."

14.4.7 ACLC Interrupt Disable Register (ACINTDIS) 0xF71C

Interrupt request enable clear (W1C). Bit placement is the same as for the ACINTSTS register. When a value is written to this register, the ACINTEN register bit in the position where a "1" was written is cleared to "0."

14.4.8 ACLC Semaphore Register (ACSEMAPH) 0xF720

This register is used for mutual exclusion control for resource.



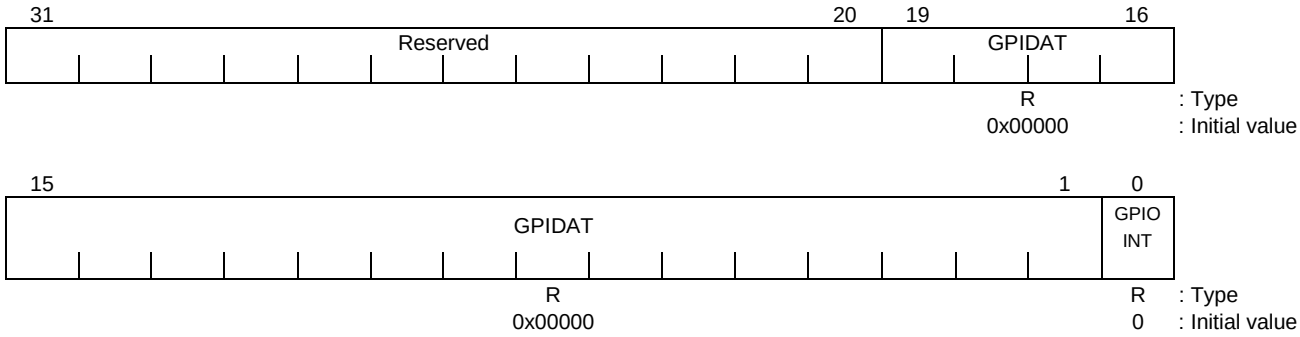
Bits	Mnemonic	Field Name	Description
31:0	SEMAPH	Semaphore flag	Semaphore flag. (Initial value: 0x0000_0000, RS/WC) RS 0: Indicates that the semaphore is unlocked. The read operation to this register will atomically set the bit[0] to lock the semaphore. 1: Indicates that the semaphore is locked. WC x: Writing any value to this register clears the bit[0] to release the semaphore.

Figure 14.4.5 ACSEMAPH Register

This register is provided primarily for the mutual exclusion between the audio and modem drivers to share the common resources of ACLC, such as the ACREGACC register and the link-control bits in the ACCTLEN/DIS register.

14.4.9 ACLC GPI Data Register (ACGPIDAT) 0xF740

This register shows GPIO (slot 12) input data.

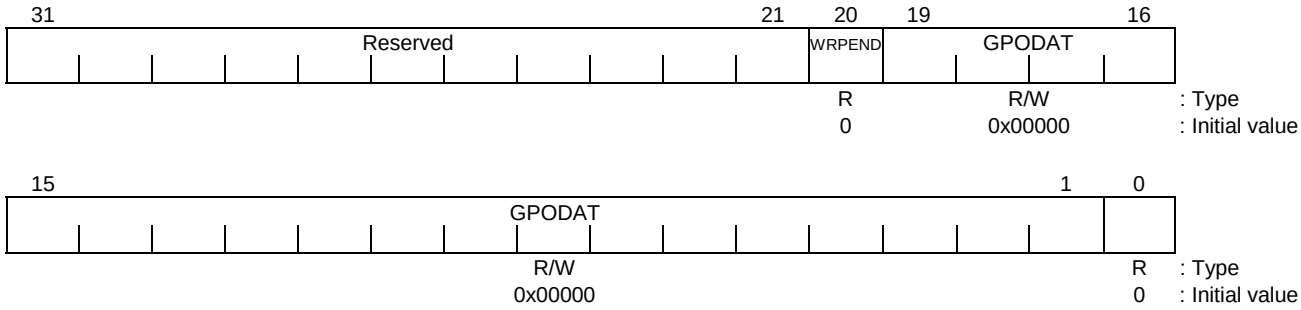


Bits	Mnemonic	Field Name	Description
31:20	—	Reserved	—
19:1	GPIDAT	GPIO-In data	GPIO-In data (Initial value: 0x0_0000, R) R Read data. The incoming slot 12 bits[19:1] are shadowed here.
0	GPIOINT	GPIO Interrupt Indication	GPIO Interrupt Indication (Initial value: 0, R) R GPIO Interrupt. The incoming slot 12 bit[0] is shadowed here.

Figure 14.4.6 ACGPIDAT Register

14.4.10 ACLC GPO Data Register (ACGPODAT) 0xF744

This register specifies GPIO (slot 12) output data.



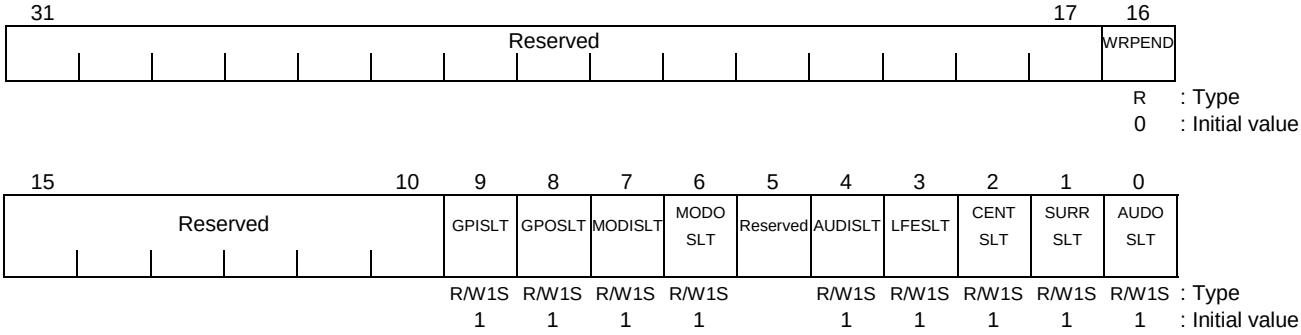
Bits	Mnemonic	Field Name	Description
31:20	—	Reserved	—
20	WRPEND	Write Pending	Write Pending (Initial value: 0, R) R 0: Indicates that the previous write operation is complete and the ACGPODAT register is ready to be written. 1: Indicates that the previous write operation is not complete and the ACGPODAT register is not yet ready to be written.
19:1	GPODAT	GPIO-Out data	GPIO-Out data (Initial value: 0x0_0000, R/W) R Reads back the value previously written to this field. W Writes data to the outgoing slot 12 bits[19:1].
0	—	—	R Reads always '0'.

Figure 14.4.7 ACGPODAT Register

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACGPODAT.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.11 ACLC Slot Enable Register (ACSLTEN) 0xF748

This register enables independently the AC-link slot data streams.



Bits	Mnemonic	Field Name	Description
31:17	—	Reserved	—
16	WRPEND	Write Pending	Write Pending (Initial value: 0, R) R 0: Indicates that the previous write operation is complete and the ACSLTEN and ACSLTDIS registers are ready to be accessed. 1: Indicates that the previous write operation is not complete and the ACSLTEN and ACSLTEDIS registers are not yet ready to be accessed.
15:10	—	Reserved	—
9	GPISLT	Enable GPI slot reception	Enable GPI slot reception. (Initial value: 1 R/W1S) R 0: Indicates that GPI slot reception is disabled. 1: Indicates that GPI slot reception is enabled. W1S 0: No effect 1: Enables GPI slot reception.
8	GPOSLT	Enable GPO Slot transmission	Enable GPO Slot transmission. (Initial value: 1 R/W1S) R 0: Indicates that GPO slot transmission is disabled. 1: Indicates that GPO slot transmission is enabled. W1S 0: No effect 1: Enables GPO slot transmission.
7	MODISLT	Enable Modem slot reception	Enable Modem slot reception. (Initial value: 1 R/W1S) R 0: Indicates that modem slot reception is disabled. 1: Indicates that modem slot reception is enabled. W1S 0: No effect 1: Enables modem slot reception.
6	MODOSLT	Enable Modem slot transmission	Enable Modem slot transmission. (Initial value: 1 R/W1S) R 0: Indicates that modem slot transmission is disabled. 1: Indicates that modem slot transmission is enabled. W1S 0: No effect 1: Enables modem slot transmission.
5	—	Reserved	—

Figure 14.4.8 ACSLTEN Register (1/2)

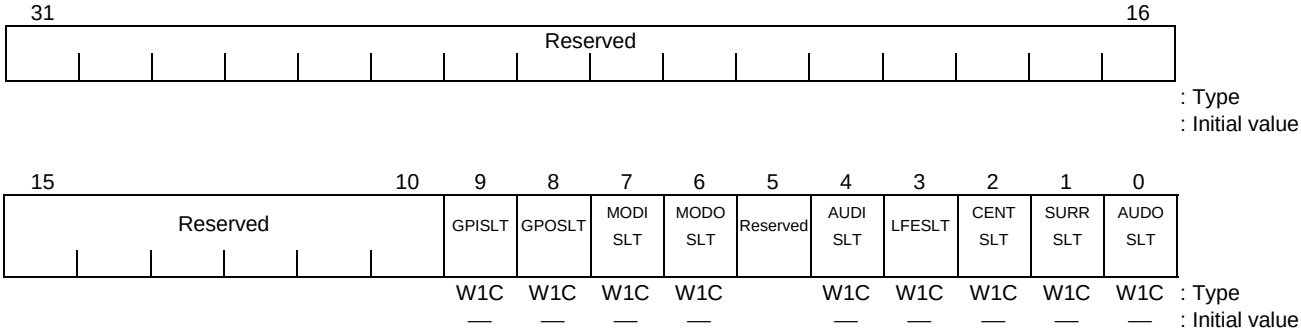
Bits	Mnemonic	Field Name	Description
4	AUDISLT	Enable Audio slot reception	Enable Audio slot reception. (Initial value: 1, R/W1S) R 0: Indicates that audio slot reception is disabled. 1: Indicates that audio slot reception is enabled. W1S 0: No effect 1: Enables audio slot reception.
3	LFESLT	Enable Audio LFE slot transmission	Enable Audio LFE slot transmission. (Initial value: 1, R/W1S) R 0: Indicates that audio LFE slot transmission is disabled. 1: Indicates that audio LFE slot transmission is enabled. W1S 0: No effect 1: Enables audio LFE slot transmission.
2	CENTSLT	Enable Audio Center slot transmission	Enable Audio Center slot transmission. (Initial value: 1, R/W1S) R 0: Indicates that audio Center slot transmission is disabled. 1: Indicates that audio Center slot transmission is enabled. W1S 0: No effect 1: Enables audio Center slot transmission.
1	SURRSLT	Enable Audio Surround L&R slot transmission	Enable Audio Surround L&R slot transmission. (Initial value: 1, R/W1S) R 0: Indicates that audio Surround L&R slot transmission is disabled. 1: Indicates that audio Surround L&R slot transmission is enabled. W1S 0: No effect 1: Enables audio Surround L&R slot transmission.
0	AUDOSLT	Enable Audio PCM L&R slot transmission	Enable Audio PCM L&R slot transmission. (Initial value: 1, R/W1S) R 0: Indicates that audio PCM L&R Slot transmission is disabled. 1: Indicates that audio PCM L&R Slot transmission is enabled. W1S 0: No effect 1: Enables audio PCM L&R slot transmission.

Figure 14.4.8 ACSLTEN Register (2/2)

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.12 ACLC Slot Disable Register (ACSLTDIS) 0xF74C

This register disables independently the AC-link slot data streams.



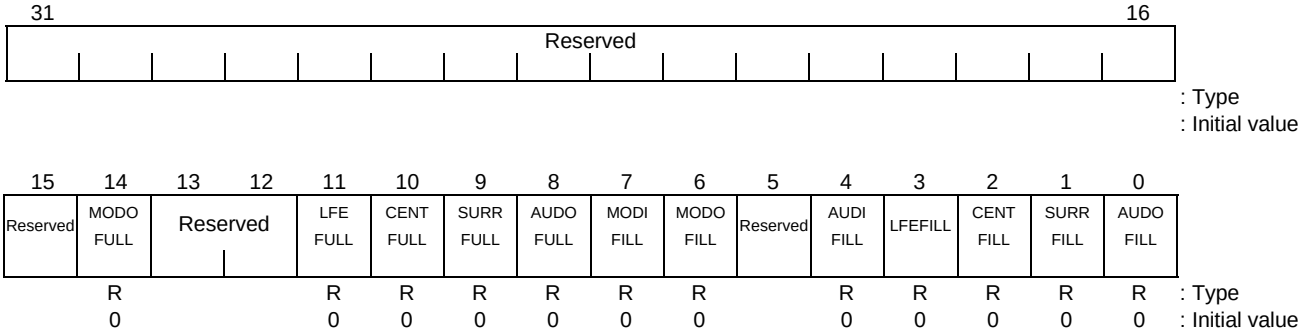
Bits	Mnemonic	Field Name	Description
31:10	—	Reserved	—
9	GPISLT	Disable GPI slot reception	Disable GPI slot reception. (Initial value: –, W1C) W1C 0: No effect 1: Disables GPI slot reception.
8	GPOSLT	Disable GPO Slot transmission	Disable GPO Slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables GPO slot transmission.
7	MODISLT	Disable Modem slot reception	Disable Modem slot reception. (Initial value: –, W1C) W1C 0: No effect 1: Disables modem slot reception.
6	MODOSLT	Disable Modem slot transmission	Disable Modem slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables modem slot transmission.
5	—	Reserved	—
4	AUDISLT	Disable Audio slot reception	Disable Audio slot reception. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio slot reception.
3	LFESLT	Disable Audio LFE slot transmission	Disable Audio LFE slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio LFE slot transmission.
2	CENTSLT	Disable Audio Center slot transmission	Disable Audio Center slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio Center slot transmission.
1	SURRSLT	Disable Audio Surround L&R slot transmission	Disable Audio Surround L&R slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio Surround L&R slot transmission.
0	AUDOSLT	Disable Audio PCM L&R slot transmission	Disable Audio PCM L&R slot transmission. (Initial value: –, W1C) W1C 0: No effect 1: Disables audio PCM L&R slot transmission.

Figure 14.4.9 ACSLTDIS Register

Writing a value into this register needs several BITCLK cycles to take effect. The software must guarantee that no write access be executed until the previous write access takes effect (completes), by reading the ACSLTEN.WRPEND bit prior to writing this register. If it is set for a long time, the BITCLK signal on the AC-link is probably inactive for whatever reason.

14.4.13 ACLC FIFO Status Register (ACFIFOSTS) 0xF750

This register indicates the AC-link slot data FIFO status.



Bits	Mnemonic	Field Name	Description
31:15	—	Reserved	—
14	MODOFULL	Modem Transmit-data Full	Modem Transmit-data Full. (Initial value: 0, R) R 0: Indicates modem transmit-data FIFO is not full. 1: Indicates modem transmit-data FIFO is full.
13:12	Reserved	Reserved	—
11	LFEFULL	Audio LFE Transmit-data Full	Audio LFE Transmit-data Full. (Initial value: 0, R) R 0: Indicates audio LFE transmit-data FIFO is not full. 1: Indicates audio LFE transmit-data FIFO is full.
10	CENTFULL	Audio Center Transmit-data Full	Audio Center Transmit-data Full. (Initial value: 0, R) R 0: Indicates audio Center transmit-data FIFO is not full 1: Indicates audio Center transmit-data FIFO is full.
9	SURRFULL	Audio Surround L&R Transmit-data Full	Audio Surround L&R Transmit-data Full. (Initial value: 0, R) R 0: Indicates audio Surround L&R transmit-data FIFO is not full. 1: Indicates audio Surround L&R transmit-data FIFO is full.
8	AUDOFULL	Audio PCM L&R Transmit-data Full	Audio PCM L&R Transmit-data Full. (Initial value: 0, R) R 0: Indicates audio PCM L&R transmit-data FIFO is not full. 1: Indicates audio PCM L&R transmit-data FIFO is full.
7	MODIFILL	Modem Receive-data Filled	Modem Receive-data Filled. (Initial value: 0, R) R 0: Indicates modem receive-data FIFO is empty. 1: Indicates modem receive-data FIFO is not empty.
6	MODOFILL	Modem Transmit-data Filled	Modem Transmit-data Filled. (Initial value: 0, R) R 0: Indicates modem transmit-data FIFO is empty. 1: Indicates modem transmit-data FIFO is not empty.
5	—	Reserved	—
4	AUDIFILL	Audio Receive-data Filled	Audio Receive-data Filled. (Initial value: 0, R) R 0: Indicates audio receive-data FIFO is empty. 1: Indicates audio receive-data FIFO is not empty.
3	LFEFILL	Audio LFE Transmit-data Filled	Audio LFE Transmit-data Filled. (Initial value: 0, R) R 0: Indicates audio LFE transmit-data FIFO is empty. 1: Indicates audio LFE transmit-data FIFO is not empty.

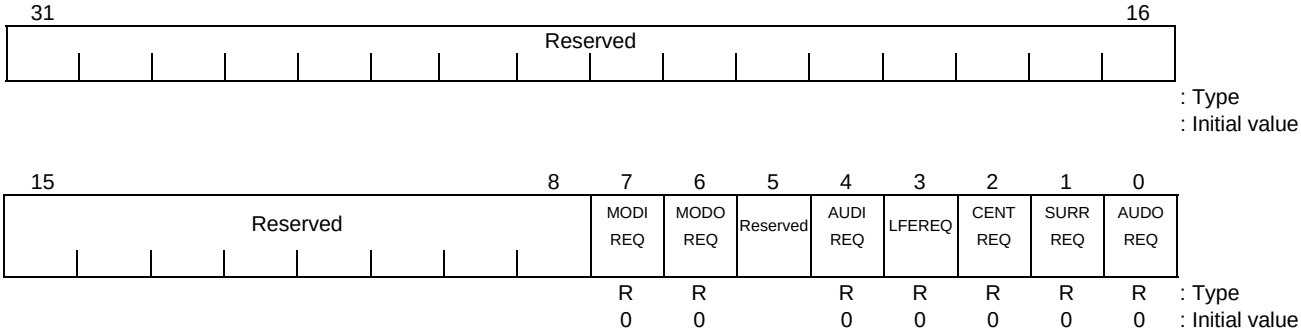
Figure 14.4.10 ACFIFOSTS Register (1/2)

Bits	Mnemonic	Field Name	Description
2	CENTFILL	Audio Center Transmit-data Filled	Audio Center Transmit-data Filled. (Initial value: 0, R) R 0: Indicates audio Center transmit-data FIFO is empty. 1: Indicates audio Center transmit-data FIFO is not empty.
1	SURRFILL	Audio Surround L&R Transmit-data Filled	Audio Surround L&R Transmit-data Filled. (Initial value: 0, R) R 0: Indicates audio Surround L&R transmit-data FIFO is empty. 1: Indicates audio Surround L&R transmit-data FIFO is not empty.
0	AUDOFILL	Audio PCM L&R Transmit-data Filled	Audio PCM L&R Transmit-data Filled. (Initial value: 0, R) R 0: Indicates audio PCM L&R transmit-data FIFO is empty. 1: Indicates audio PCM L&R transmit-data FIFO is not empty

Figure 14.4.10 ACFIFOSTS Register (2/2)

14.4.14 ACLC DMA Request Status Register (ACDMASTS) 0xF780

This register indicates the AC-link slot data DMA request status.



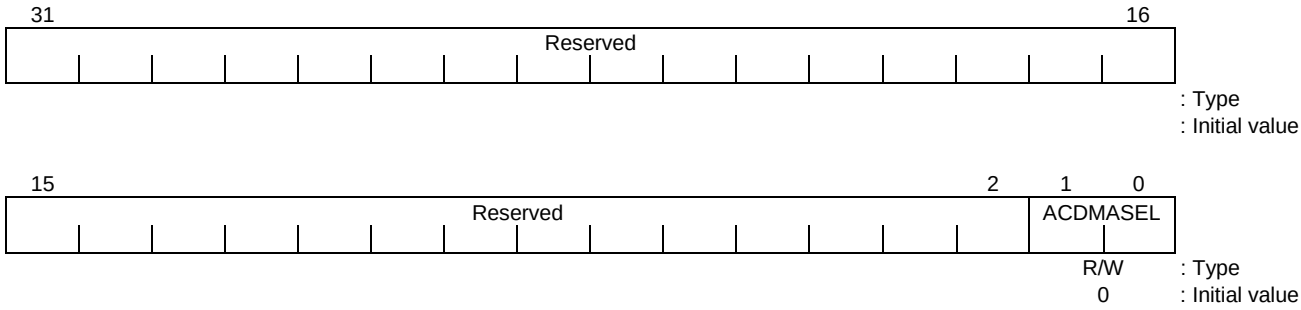
Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	MODIREQ	Modem Data Reception Request	Modem Data Reception Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
6	MODOREQ	Modem Data Transmission Request	Modem Data Transmission Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
5	—	Reserved	—
4	AUDIREQ	Audio Data Reception Request	Audio Data Reception Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
3	LFEREQ	Audio LFE Data Transmission Request	Audio LFE Data Transmission Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
2	CENTREQ	Audio Center Data Transmission Request	Audio Center Data Transmission Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
1	SURRREQ	Audio Surround L&R Data Transmission Request	Audio Surround L&R Data Transmission Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.
0	AUDOREQ	Audio PCM L&R Data Transmission Request	Audio PCM L&R Data Transmission Request (Initial value: 0, R) R 0: No request is pending. 1: Request is pending.

Figure 14.4.11 ACDMASTS Register

This read-only register shows if any DMA request is pending for each data I/O channel. A DMA request can be pending after the software deactivates the DMAC channel or disables DMA by ACCTLDIS register bit to complete DMA operation. In this case, write or read the sample data register (ACAUDODAT and others) to clear the DMA request.

14.4.15 ACLC DMA Channel Selection Register (ACDMASEL) 0xF784

This register is used to select and check the channel allocation for AC-link slot data DMA.



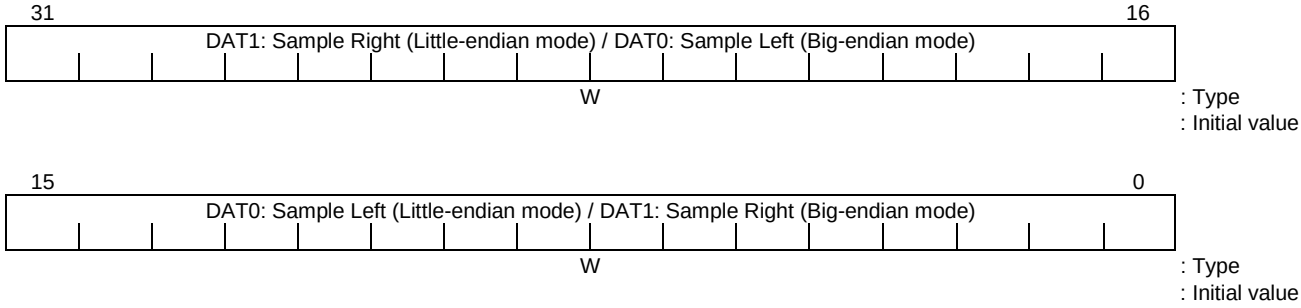
Bits	Mnemonic	Field Name	Description
31:2	—	Reserved	—
1:0	ACDMASEL	DMA Channel Selection	DMA Channel Selection (Initial value: 0, R/W) W ACDMASEL: DMA Channel Selection 0: PCM L&R out, Audio in, and Modem out&in. 1: PCM L&R out, Surround L&R out, and Modem out&in. 2: PCM L&R out, Surround L&R out, Center out, and LFE out. 3: PCM L&R out, Surround L&R out, Center out, and Audio in.

Figure 14.4.12 ACDMASEL Register

This register selects DMA channel mapping mode. The software is recommended to make sure no DMA request is pending before changing this register value.

14.4.16 ACLC Surround Data Register (ACAUDODAT) 0xF7A0
ACLC Audio PCM Output Data Register (ACSURRDAT) 0xF7A4

These registers are used to write audio PCM L&R and surround L&R output data.

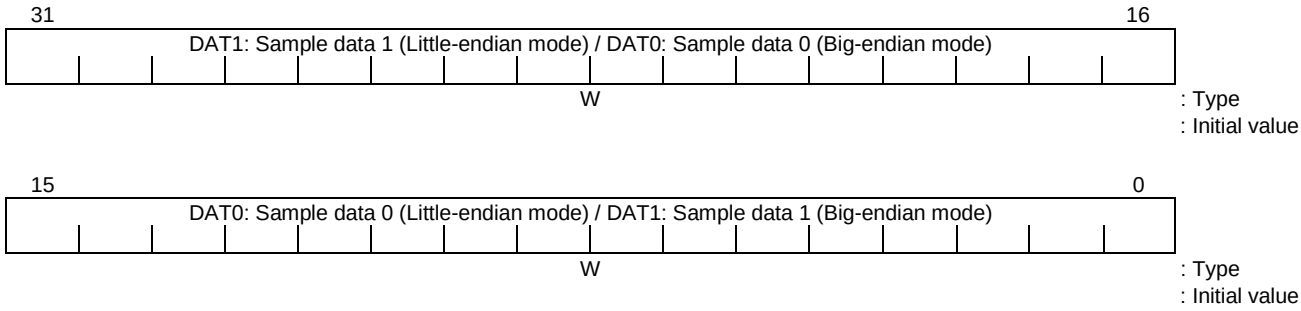


Bits	Mnemonic	Field Name	Description		
31:16	—	—	W	DAT1: Sample Right	DAT0: Sample Left
15:0	—	—	W	DAT0: Sample Left	Left DAT1: Sample Right

Figure 14.4.13 ACAUDODAT/ACSURRDAT Register

14.4.17 ACLC Center Data Register (ACCENTDAT) 0xF7A8
ACLC LFE Data Register (ACLFEDAT) 0xF7AC
ACLC Audio PCM Input Data Register (ACMODODAT) 0xF7B8

This registers are used to write audio center, LFE, and modem output data.

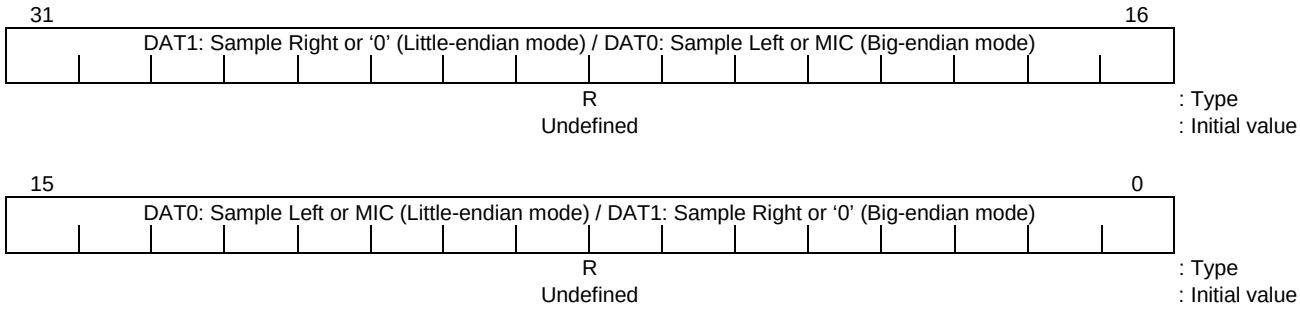


Bits	Mnemonic	Field Name	Description		
31:16	—	—	W	DAT1: Sample data 1	DAT0: Sample data 0
15:0	—	—	W	DAT0: Sample data 0	DAT1: Sample data 1

Figure 14.4.14 ACCENDAT/ACLFEDAT/ACMODODAT Register

14.4.18 ACLC Modem Output Data Register (ACAUDIDAT) 0xF7B0

This register is used to read audio PCM L&R input data.

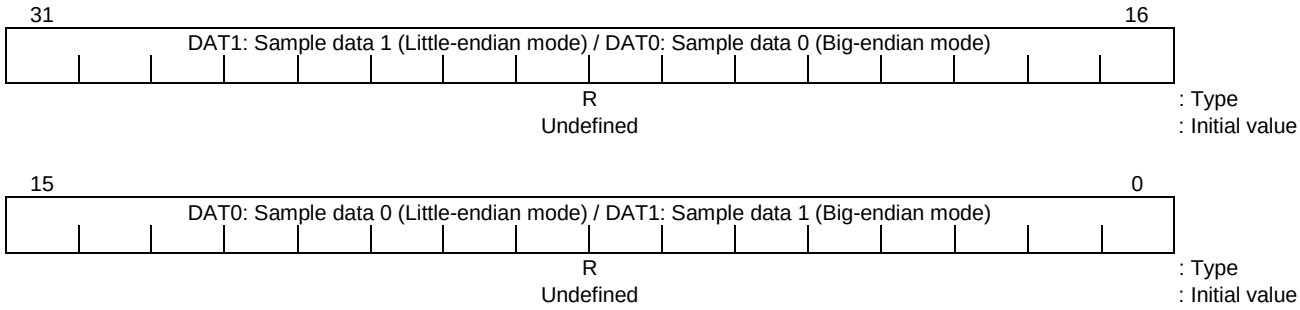


Bits	Mnemonic	Field Name	Description		
31:16	—	—	R	DAT1: Sample Right or '0'	DAT0: Sample Left or MIC
15:0	—	—	R	DAT0: Sample Left or MIC	DAT1: Sample Right or '0'

Figure 14.4.15 ACAUDIDAT Register

14.4.19 ACLC Modem Input Data Register (ACMODIDAT) 0xF7BC

This register is used to read modem input data.

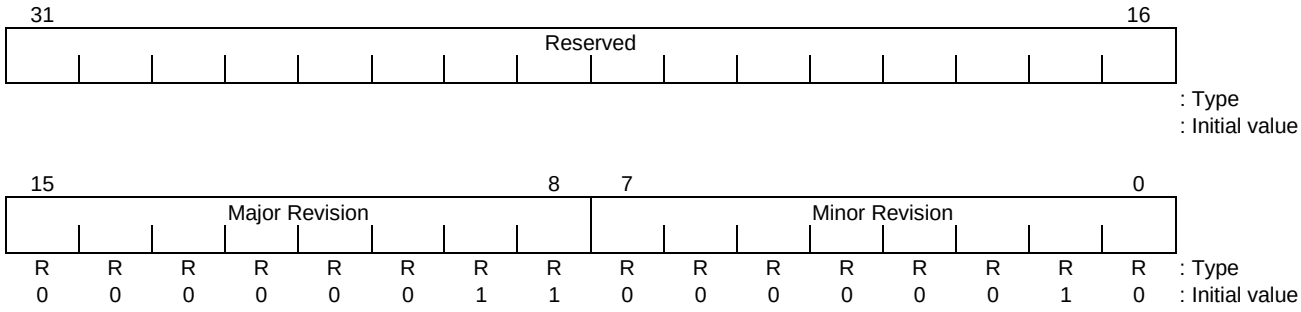


Bits	Mnemonic	Field Name	Description		
31:16	—	—	R	DAT1: Sample data 1	DAT0: Sample data 0
15:0	—	—	R	DAT0: Sample data 0	DAT1: Sample data 1

Figure 14.4.16 ACMODIDAT Register

14.4.20 ACLC Revision ID Register (ACREVID) 0xF7FC

This register is used to read ACLC module's revision ID.



Bits	Mnemonic	Field Name	Description
31:16	—	Reserved	—
15:8	—	—	R Major Revision Contact Toshiba technical staff for an explanation of the revision value.
7:0	—	—	R Minor Revision Contact Toshiba technical staff for an explanation of the revision value.

Figure 14.4.17 ACREVID Register

This read-only register shows the revision of ACLC module. Note that this number is not related to the AC'97 specification revision.

15. Interrupt Controller

15.1 Characteristics

The TX4925 on-chip Interrupt Controller (IRC) receives interrupt requests from the TX4925 on-chip peripheral circuitry as well as external interrupt requests then generates interrupt requests to the TX49/H2 processor core.

Also, the Interrupt Controller has a 32-bit flag register that generates interrupt requests to either external devices or to the TX49/H2 core.

The Interrupt Controller has the following characteristics.

- Supports interrupts from 21 types of on-chip peripheral circuits and a maximum of 8 external interrupt signal inputs
- Sets 8 priority interrupt levels for each interrupt input
- Can select either edge detection or level detection for each external interrupt when in the interrupt detection mode
- As a flag register used for interrupt requests, the Interrupt Controller contains a 32-bit readable/writeable register and can issue interrupt requests to external devices as well as to the TX49/H2 core (IRC interrupt).

15.2 Block Diagram

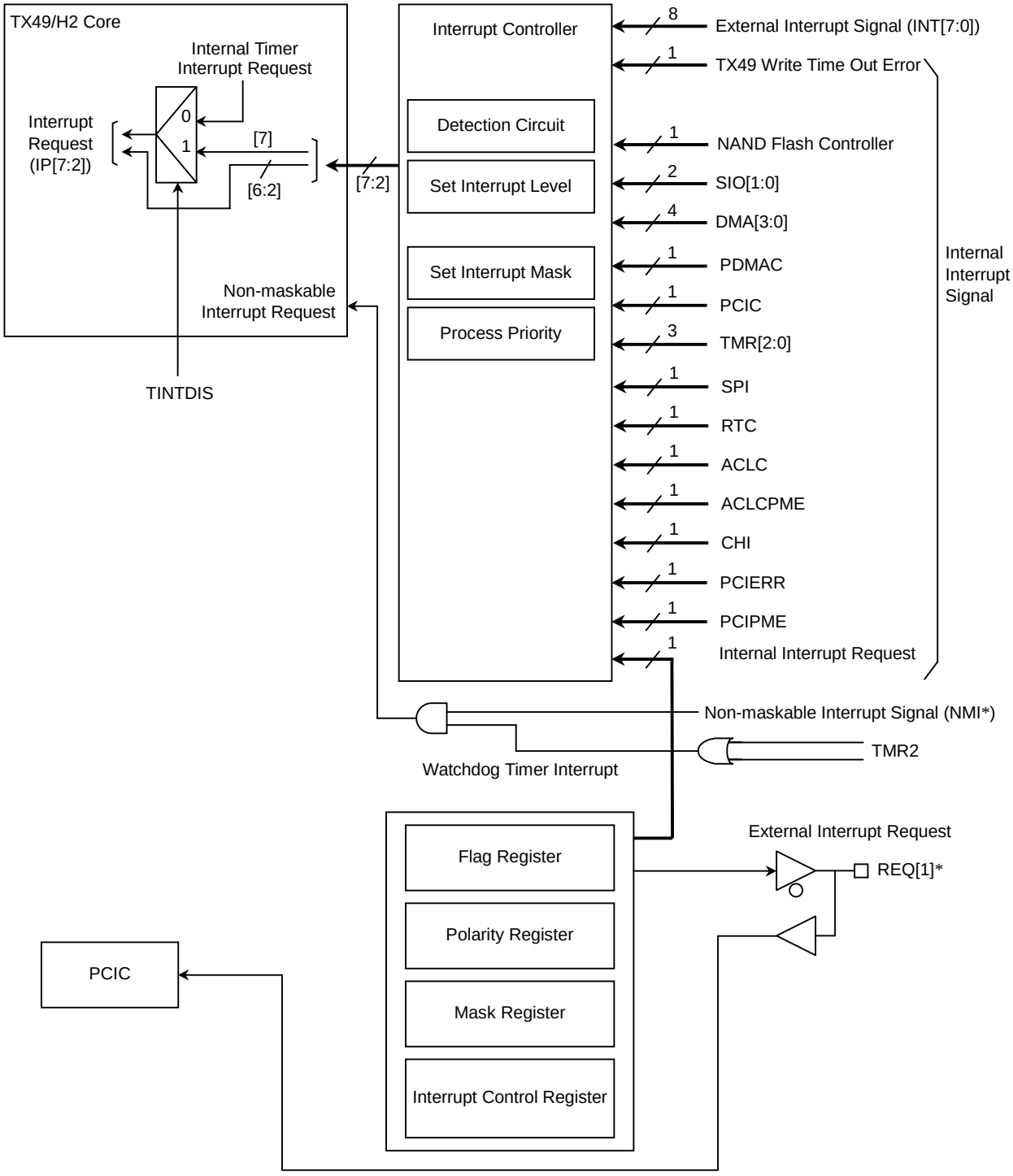


Figure 15.2.1 Interrupt Controller Outline

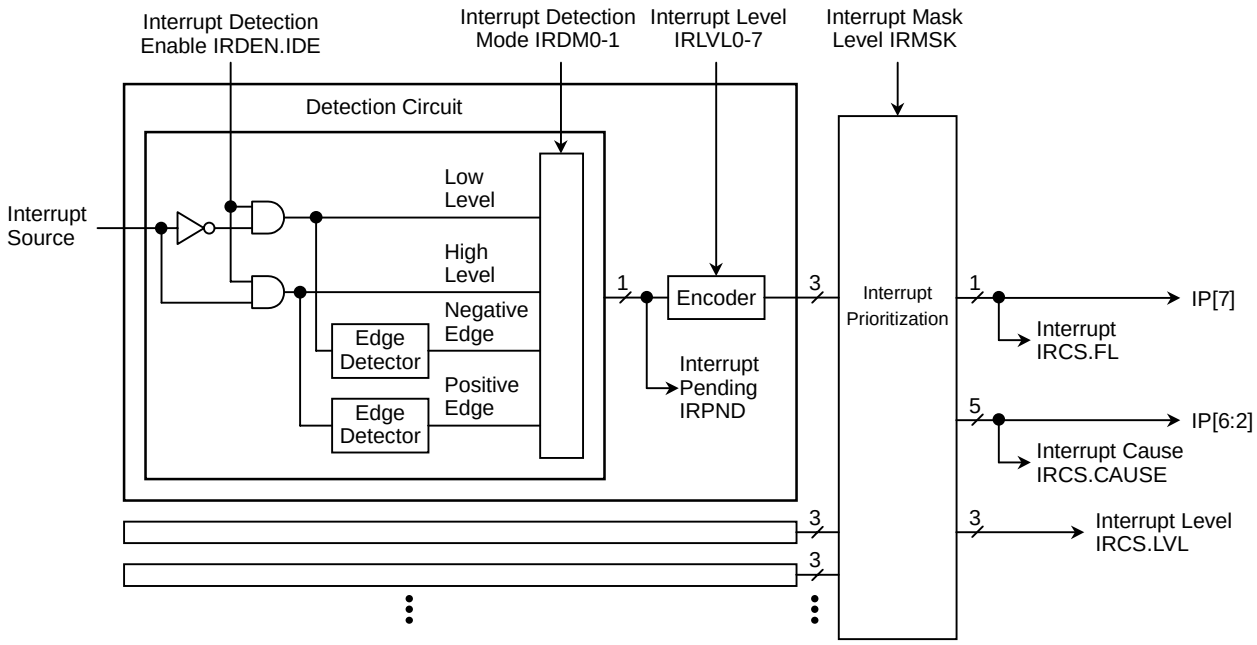


Figure 15.2.2 Internal Block Diagram of Interrupt Controller

15.3 Detailed Explanation

15.3.1 Interrupt Sources

The TX4925 has as interrupt sources interrupts from 21 types of on-chip peripheral circuits and 8 external interrupt signals.

Table 15.3.1 lists the interrupt sources. Signals with the lower interrupt number have the higher priority. Refer to section “15.3.4 Interrupt Priority Assigning” for the priorities.

Table 15.3.1 Interrupt Sources and Priorities

Priority	Interrupt Number	Interrupt Source
High	0	(Reserved)
	1	TX49 Write Timeout Error (Internal)
	2	INT[0] (External)
	3	INT[1] (External)
	4	INT[2] (External)
	5	INT[3] (External)
	6	INT[4] (External)
	7	INT[5] (External)
	8	INT[6] (External)
	9	INT[7] (External)
	10	(Reserved)
	11	NAND Flash Controller (internal)
	12	SIO0 (Internal)
	13	SIO1 (Internal)
	14	DMA0 (Internal)
	15	DMA1 (Internal)
	16	DMA2 (Internal)
	17	DMA3 (Internal)
	18	IRC (Internal)
	19	PDMAC (Internal)
	20	PCIC (Internal)
	21	TMR0 (Internal)
	22	TMR1 (Internal)
	23	TMR2 (Internal)
	24	SPI (Internal)
	25	RTC (Internal)
	26	ACLC (Internal)
	27	ACLCPME (Internal)
	28	CHI (Internal)
	29	PCIERR (Internal)
	30	PCIPME (Internal)
Low	31	(Reserved)

In addition to the above, the TX49/H2 core has a TX49/H2 core internal timer interrupt and two software interrupts, but these interrupts are directly reported to the TX49/H2 core independently of this Interrupt Controller. Please refer to the “64-Bit TX System RISC TX49/H2 Core Architecture” for more information.

15.3.2 Interrupt Request Detection

In order to perform interrupt detection, each register of the Interrupt Controller is initialized, then the IDE bit of the Interrupt Detection Enable Register (IRDEN) is set to "1." All interrupts detected by the Interrupt Controller are masked when this bit is cleared.

It is possible to set each interrupt factor detection mode using Interrupt Detection Mode Register 0 (IRDM0) and Interrupt Detection Mode Register 1 (IRDM1). There are four detection modes: Low level, High level, falling edge, and rising edge.

The detected interrupt factors can be read out from the Interrupt Pending Register (IRPND).

15.3.3 Interrupt Level Assigning

Interrupt levels from 0 to 7 are assigned to each detected interrupt using the Interrupt Level Register (IRLVL0-7). Interrupt level 7 is the highest priority and interrupt level1 is the lowest priority. Level 0 interrupts will be masked. (Table 15.3.2).

The priorities set by these interrupt levels will be given higher priority than the priorities provided for each interrupt source indicated in Table 15.3.1.

Table 15.3.2 Interrupt Levels

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

15.3.4 Interrupt Priority Assigning

When multiple interrupt requests exist, the Interrupt Controller selects the interrupt with the highest priority according to the priority level and interrupt number. Interrupt factors with an interrupt level lower than the interrupt level specified by the Interrupt Mask Level Register (IRMSK) will be excluded (masked).

When the interrupt with the highest priority is selected, then the interrupt number of that interrupt is set in the interrupt factor field (CAUSE) of the Interrupt Current Status Register (IRCS), the interrupt level is set in the Interrupt Level field (LVL), and the Interrupt Flag bit (IF) is set.

Priorities are assigned as follows.

- When interrupt levels differ, the interrupt with the higher interrupt level has priority (Table 15.3.2)
- When multiple interrupts with the same interrupt level are simultaneously detected, the interrupt with the smaller interrupt number has priority (Table 15.3.1).

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

- When an interrupt request with a higher interrupt level than that of the currently selected interrupt is detected. However, when the interrupt levels are equal, the Interrupt Level field (LVL) does not change even if the interrupt number is small.
- When the interrupt level (IRLVLn.ILm) of the currently selected interrupt changes to a value smaller than the current setting.
- When the currently selected interrupt is cleared (refer to "15.3.6 Clearing Interrupt Requests").

15.3.5 Interrupt Notification

When the interrupt with the highest priority is selected, then the interrupt factor is reported to the Interrupt Current Status Register (IRCS) and an interrupt is reported to the TX49/H2 core.

The TX49/H2 core distinguishes interrupt factors using the IP field (IP[7:2]) of the Cause Register. The interrupt notification from the Interrupt Controller is reflected in the IP[2] bit. The Interrupt Handler uses the IP[2] bit to judge whether or not there are interrupts from this Interrupt Controller and uses the Interrupt Current Status Register (IRCS) to determine the interrupt cause.

The Interrupt Factor field (IRCS.CAUSE) value is reflected in the remaining bits of the IP field. Since bit IP[7] is also being used for notification of TX49/H2 CPU core internal timer interrupts, the contents specified by IP[7] differ according to whether internal timer interrupts are set to valid (TINTDIS=0) or invalid (TINTDIS=1), as indicated Table 15.3.3.

The Interrupt Factor field (IRCS.CAUSE) value is reflected in the remaining bits of the IP field.

TINTDIS is the value that is set from ADDR[0] at the timing when the RESET* signal is deasserted. See the explanation "3.3 Pin Multiplexing" for more information.

Table 15.3.3 Interrupt Notification to IP[7:2] of the CP0 Cause Register

TINTDIS	IP[7]	IP[6:3]	IP[2]
0 (Internal Timer Interrupts: Valid)	Internal Timer Interrupt Notification	IRCS.CAUSE[3:0]	IRCS.IF
1 (Internal Timer Interrupts: Invalid)	IRCS.CAUSE[4:0]		IRCS.IF

15.3.6 Clearing Interrupt Requests

Interrupt requests are cleared according to the following process.

- When the detection mode is set to the High level or Low level:
  Operation is performed to deassert the request of a source that is asserting an interrupt request.
- When the detection mode is set to Rising edge or Falling edge
  Edge detection requests are cleared by first specifying the interrupt source of the interrupt request to be cleared in the Edge Detection Clear Source field (EDCS0 or EDCS1) of the Interrupt Edge Detection Clear Register (IREDC) then writing the resulting value when the corresponding Edge Detection Clear Enable bit (EDCE0 or EDCE1) is set to "1."

15.3.7 Interrupt Requests

It is possible to make interrupt requests to external devices and interrupt requests (IRC interrupts) to the TX49/H2 core by using a 32-bit interrupt request flag register. REQ[1]* signals are used as interrupt output signals. Consequently, external interrupt requests can only be used when in the PCI External Arbiter mode. Also, internal interrupt requests are assigned to interrupt number 13 of the Interrupt Controller (IRC).

The following six registers set the interrupts.

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

The following formulas derive the interrupt generation conditions:

Internal interrupt request =

(|((IRFLAG[31:0] ^ IRPOL[31:0]) & IRMASKINT[31:0]))^ IRRCNT.INTPOL

External interrupt request =

(|((IRFLAG[31:0] ^ IRPOL[31:0]) & IRMASKEXT[31:0]))^ IRRCNT.EXTPOL

In the above formulas, "^" indicates Exclusive OR operations and "|" indicates reduction operators that perform an OR operation on all bits.

Also, the External Interrupt OD Control bit (IRRCNT.OD) of the Interrupt Request Control Register can select whether the external interrupt supply signal is open drain output or totem pole output.

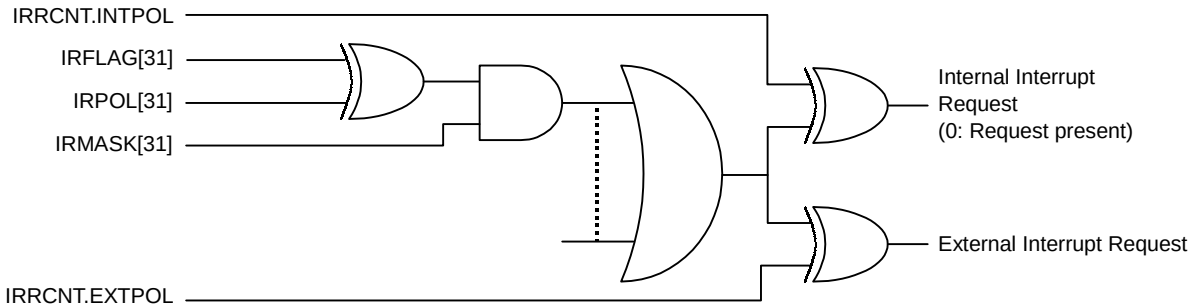


Figure 15.3.1 External Interrupt Request Logic

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

Either “0” or “1” can be written to Flag Register 0

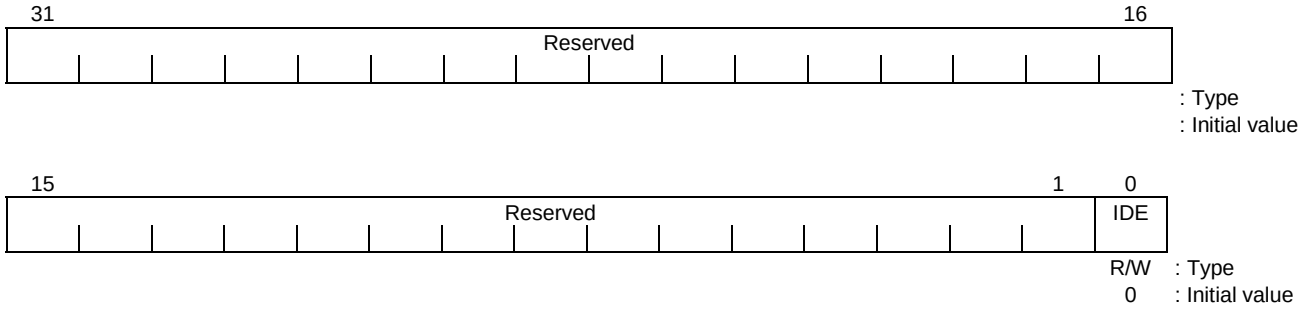
In the case of Flag Register 1 however, “1” can be written from the TX49/H2 core, but “0” cannot be written. On the other hand, bits that wrote “1” are cleared to “0” in the case of access from a device other than the TX49/H2 core (access from an external PCI device for example). The bit value at this time will not change even if “0” is written. This register sends interrupt notification from the TX49/H2 core to external devices. External devices can be used in applications that clear these interrupt notifications.

15.4 Registers

Table 15.4.1 Interrupt Control Registers

Reference	Address	Bit Width	Register	Register Name
15.4.1	0xF600	32	IRDEN	Interrupt Detection Enable Register
15.4.2	0xF604	32	IRDM0	Interrupt Detection Mode Register 0
15.4.3	0xF608	32	IRDM1	Interrupt Detection Mode Register 1
15.4.4	0xF610	32	IRLVL0	Interrupt Level Register 0
15.4.5	0xF614	32	IRLVL1	Interrupt Level Register 1
15.4.6	0xF618	32	IRLVL2	Interrupt Level Register 2
15.4.7	0xF61C	32	IRLVL3	Interrupt Level Register 3
15.4.8	0xF620	32	IRLVL4	Interrupt Level Register 4
15.4.9	0xF624	32	IRLVL5	Interrupt Level Register 5
15.4.10	0xF628	32	IRLVL6	Interrupt Level Register 6
15.4.11	0xF62C	32	IRLVL7	Interrupt Level Register 7
15.4.12	0xF640	32	IRMSK	Interrupt Mask Register
15.4.13	0xF660	32	IREDC	Interrupt Edge Detection Clear Register
15.4.14	0xF680	32	IRPND	Interrupt Pending Register
15.4.15	0xF6A0	32	IRCS	Interrupt Current Status Register
15.4.16	0xF510	32	IRFLAG0	Interrupt Request Flag Register 0
15.4.17	0xF514	32	IRFLAG1	Interrupt Request Flag Register 1
15.4.18	0xF518	32	IRPOL	Interrupt Request Polarity Control Register
15.4.19	0xF51C	32	IRRCNT	Interrupt Request Control Register
15.4.20	0xF520	32	IRMASKINT	Interrupt Request Internal Interrupt Mask Register
15.4.21	0xF524	32	IRMASKEXT	Interrupt Request External Interrupt Mask Register

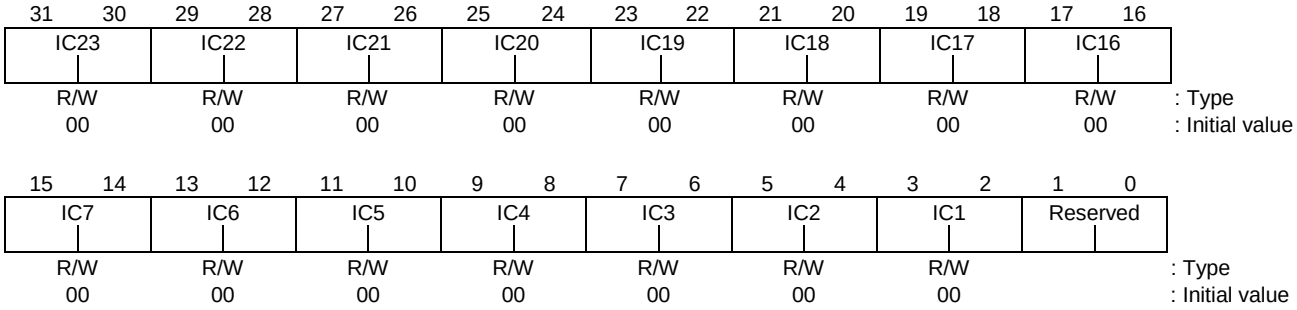
15.4.1 Interrupt Detection Enable Register (IRDEN) 0xF600



Bits	Mnemonic	Field Name	Explanation
31:1	—	Reserved	—
0	IDE	Interrupt Detection Enable	Interrupt Detection Enable (Initial value: 0, R/W) Enables interrupt detection. 0: Stop interrupt detection. 1: Start interrupt detection

Figure 15.4.1 Interrupt Detection Enable Register

15.4.2 Interrupt Detection Mode Register 0 (IRDM0) 0xF604



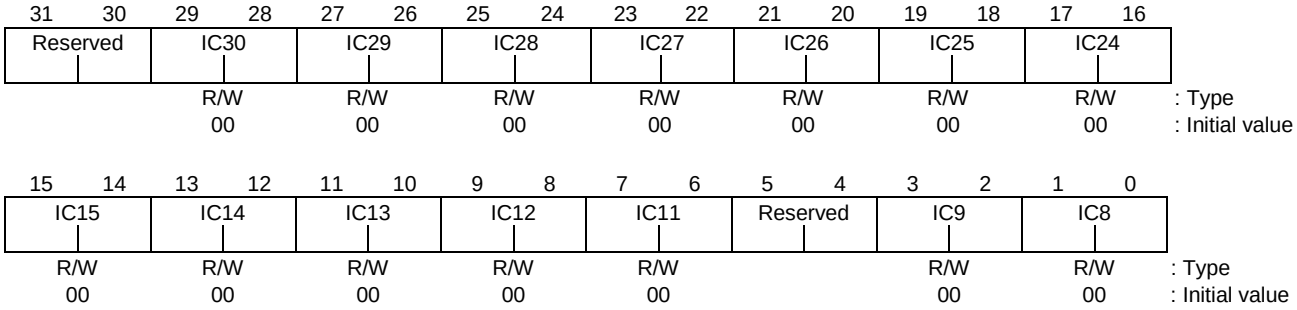
Bits	Mnemonic	Field Name	Explanation
31:30	IC23	Interrupt Source Control 23	Interrupt Source Control 23 (Initial value: 00, R/W) These bits specify the active state of TMR[2] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
29:28	IC22	Interrupt Source Control 22	Interrupt Source Control 22 (Initial value: 00, R/W) These bits specify the active state of TMR[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
27:26	IC21	Interrupt Source Control 21	Interrupt Source Control 21 (Initial value: 00, R/W) These bits specify the active state of TMR[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
25:24	IC20	Interrupt Source Control 20	Interrupt Source Control 20 (Initial value: 00, R/W) These bits specify the active state of PCIC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
23:22	IC19	Interrupt Source Control 19	Interrupt Source Control 19 (Initial value: 00, R/W) These bits specify the active state of PDMAC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
21:20	IC18	Interrupt Source Control 18	Interrupt Source Control 18 (Initial value: 00, R/W) These bits specify the active state of IRC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
19:18	IC17	Interrupt Source Control 17	Interrupt Source Control 17 (Initial value: 00, R/W) These bits specify the active state of DMA[3] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable

Figure 15.4.2 Interrupt Detection Mode Register 0 (1/2)

Bits	Mnemonic	Field Name	Explanation
17:16	IC16	Interrupt Source Control 16	Interrupt Source Control 16 (Initial value: 00, R/W) These bits specify the active state of DMA[2] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
15:14	IC7	Interrupt Source Control 7	Interrupt Source Control 7 (Initial value: 00, R/W) These bits specify the active state of external INT[5] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
13:12	IC6	Interrupt Source Control 6	Interrupt Source Control 6 (Initial value: 00, R/W) These bits specify the active state of external INT[4] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
11:10	IC5	Interrupt Source Control 5	Interrupt Source Control 5 (Initial value: 00, R/W) These bits specify the active state of external INT[3] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
9:8	IC4	Interrupt Source Control 4	Interrupt Source Control 4 (Initial value: 00, R/W) These bits specify the active state of external INT[2] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
7:6	IC3	Interrupt Source Control 3	Interrupt Source Control 3 (Initial value: 00, R/W) These bits specify the active state of external INT[1] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
5:4	IC2	Interrupt Source Control 2	Interrupt Source Control 2 (Initial value: 00, R/W) These bits specify the active state of external INT[0] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
3:2	IC1	Interrupt Source Control 1	Interrupt Source Control 1 (Initial value: 00, R/W) These bits specify the active state of TX49 Write Timeout Error interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
1:0	—	Reserved	—

Figure 15.4.2 Interrupt Detection Mode Register 0 (2/2)

15.4.3 Interrupt Detection Mode Register 1 (IRDM1) 0xF608



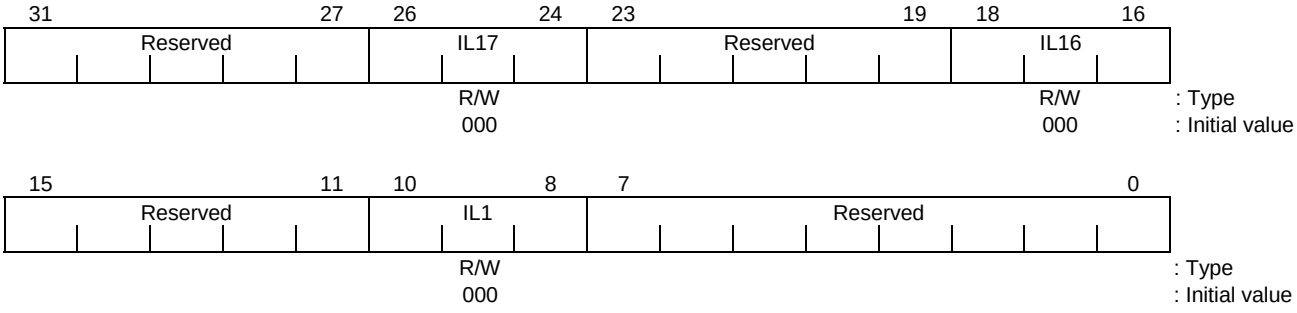
Bits	Mnemonic	Field Name	Explanation
31:30	—	Reserved	—
29:28	IC30	Interrupt Source Control 30	Interrupt Source Control 30 (Initial value: 00, R/W) These bits specify the active state of PCIPME interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
27:26	IC29	Interrupt Source Control 29	Interrupt Source Control 29 (Initial value: 00, R/W) These bits specify the active state of PCIERR interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
25:24	IC28	Interrupt Source Control 28	Interrupt Source Control 28 (Initial value: 00, R/W) These bits specify the active state of CHI interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
23:22	IC27	Interrupt Source Control 27	Interrupt Source Control 27 (Initial value: 00, R/W) These bits specify the active state of ACLCPME interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
21:20	IC26	Interrupt Source Control 26	Interrupt Source Control 26 (Initial value: 00, R/W) These bits specify the active state of ACLC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
19:18	IC25	Interrupt Source Control 25	Interrupt Source Control 25 (Initial value: 00, R/W) These bits specify the active state of RTC interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
17:16	IC24	Interrupt Source Control 24	Interrupt Source Control 26 (Initial value: 00, R/W) These bits specify the active state of SPI interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable

Figure 15.4.3 Interrupt Detection Mode Register 1 (1/2)

Bits	Mnemonic	Field Name	Explanation
15:14	IC15	Interrupt Source Control 15	Interrupt Source Control 15 (Initial value: 00, R/W) These bits specify the active state of DMA[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
13:12	IC14	Interrupt Source Control 14	Interrupt Source Control 14 (Initial value: 00, R/W) These bits specify the active state of DMA[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
11:10	IC13	Interrupt Source Control 13	Interrupt Source Control 13 (Initial value: 00, R/W) These bits specify the active state of SIO[1] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
9:8	IC12	Interrupt Source Control 12	Interrupt Source Control 12 (Initial value: 00, R/W) These bits specify the active state of SIO[0] interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
7:6	IC11	Interrupt Source Control 11	Interrupt Source Control 11 (Initial value: 00, R/W) These bits specify the active state of NAND Flash Controller interrupts. 00: Low level active 01: Disable 10: Disable 11: Disable
5:4	—	Reserved	—
3:2	IC9	Interrupt Source Control 9	Interrupt Source Control 9 (Initial value: 00, R/W) These bits specify the active state of external INT[7] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active
1:0	IC8	Interrupt Source Control 8	Interrupt Source Control 8 (Initial value: 00, R/W) These bits specify the active state of external INT[6] interrupts. 00: Low level active 01: High level active 10: Falling edge active 11: Rising edge active

Figure 15.4.3 Interrupt Detection Mode Register 1 (2/2)

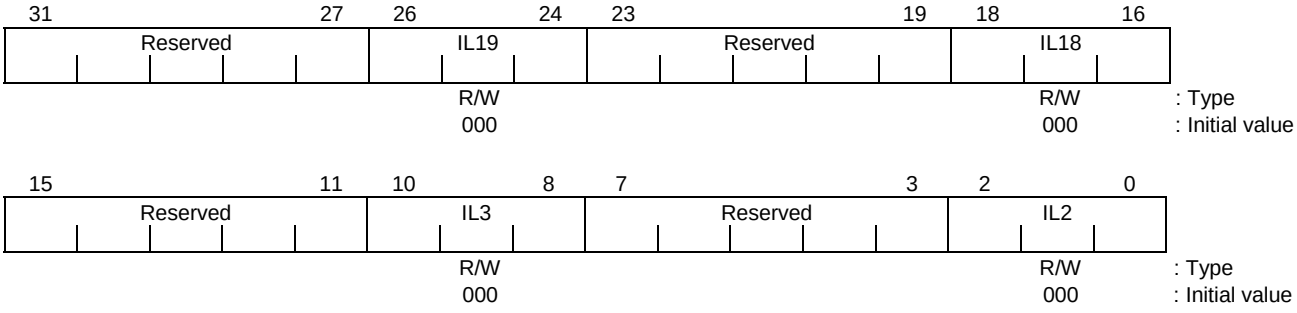
15.4.4 Interrupt Level Register 0 (IRLVL0) 0xF610



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL17	Interrupt Level 17	Interrupt Level of INT[17] (Initial value: 000, R/W) These bits specify the interrupt level of DMA[3] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL16	Interrupt Level 16	Interrupt Level of INT[16] (Initial value: 000, R/W) These bits specify the interrupt level of DMA[2] interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL1	Interrupt Level 1	Interrupt Level of INT[1] (Initial value: 000, R/W) These bits specify the interrupt level for TX49 Write Timeout Error interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:0	—	Reserved	—

Figure 15.4.4 Interrupt Level Register 0

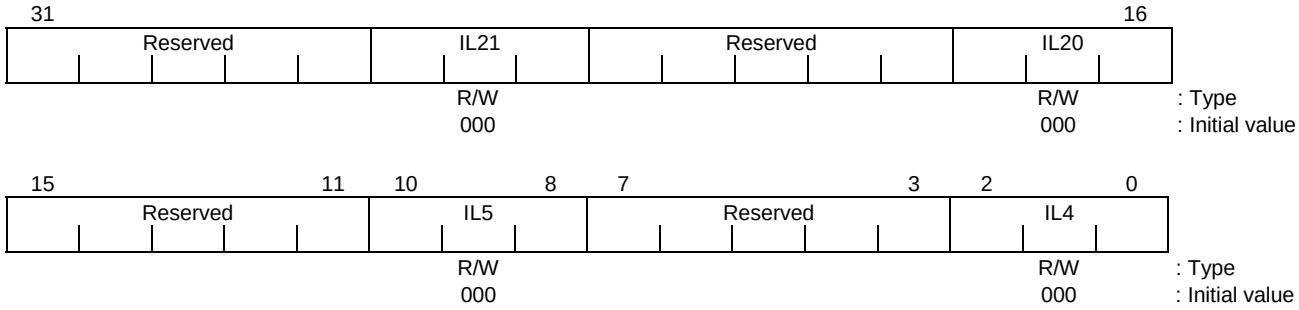
15.4.5 Interrupt Level Register 1 (IRLVL1) 0xF614



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL19	Interrupt Level 19	Interrupt Level of INT[19] (Initial value: 000, R/W) These bits specify the interrupt level of PDMAC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL18	Interrupt Level 18	Interrupt Level of INT[18] (Initial value: 000, R/W) These bits specify the interrupt level of IRC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL3	Interrupt Level 3	Interrupt Level of INT[3] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[1]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL2	Interrupt Level 2	Interrupt Level of INT[2] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[0]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.5 Interrupt Level Register 1

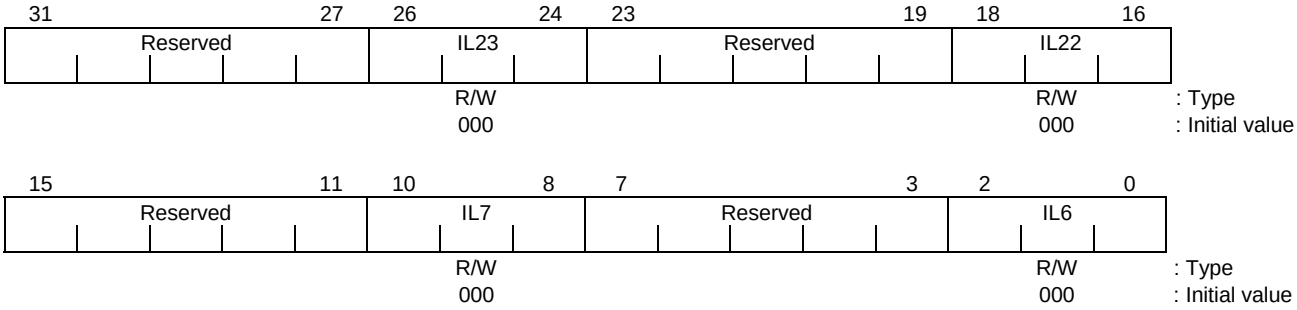
15.4.6 Interrupt Level Register 2 (IRLVL2) 0xF618



Bits	Mnemonic	Field Name	Explanation
31:11	—	Reserved	—
26:24	IL21	Interrupt Level 21	Interrupt Level of INT[21] (Initial value: 000, R/W) These bits specify the interrupt level of TMR[0]. 000: Interrupt Level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL20	Interrupt Level 20	Interrupt Level of INT[20] (Initial value: 000, R/W) These bits specify the interrupt level of PCIC interrupts. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL5	Interrupt Level 5	Interrupt Level of INT[5] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[3]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL4	Interrupt Level 4	Interrupt Level of INT[4] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[2]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.6 Interrupt Level Register 2

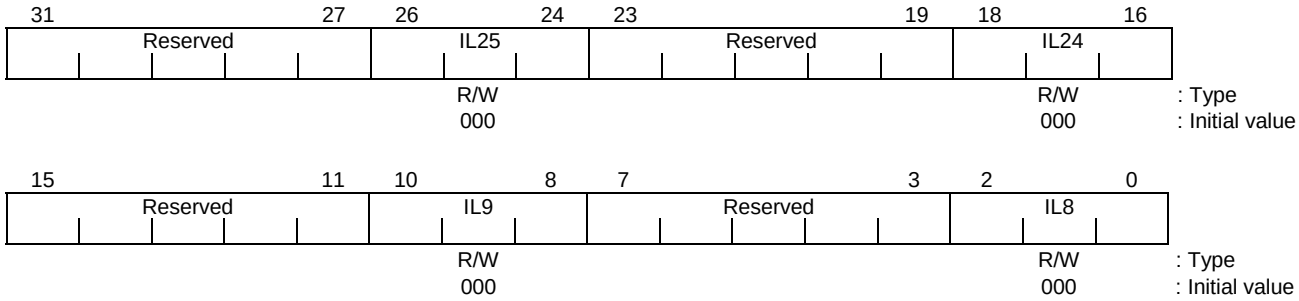
15.4.7 Interrupt Level Register 3 (IRLVL3) 0xF61C



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL23	Interrupt Level 23	Interrupt Level of INT[23] (Initial value: 000, R/W) These bits specify the interrupt level of TMR[2]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL22	Interrupt Level 22	Interrupt Level of INT[22] (Initial value: 000, R/W) These bits specify the interrupt level of TMR[1]. 000: Interrupt level 0 (Interrupt disable) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL7	Interrupt level 7	Interrupt Level of INT[7] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[5] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL6	Interrupt level 6	Interrupt Level of INT[6] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[4] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.7 Interrupt Level Register 3

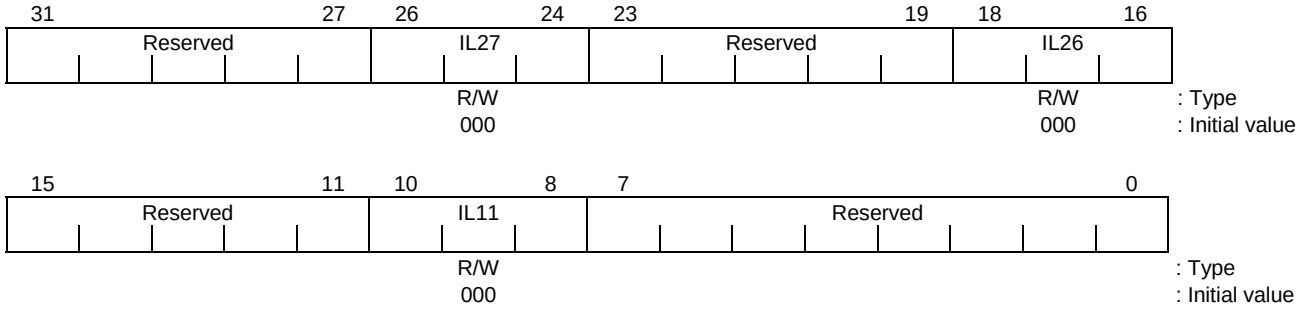
15.4.8 Interrupt Level Register 4 (IRLVL4) 0xF620



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL25	Interrupt level 25	Interrupt Level of INT[25] (Initial value: 000, R/W) These bits specify the interrupt level of RTC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL24	Interrupt level 24	Interrupt Level of INT[24] (Initial value: 000, R/W) These bits specify the interrupt level of SPI interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL9	Interrupt level 9	Interrupt Level of INT[9] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[7] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL8	Interrupt level 8	Interrupt Level of INT[8] (Initial value: 000, R/W) These bits specify the interrupt level of external INT[6] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.8 Interrupt Level Register 4

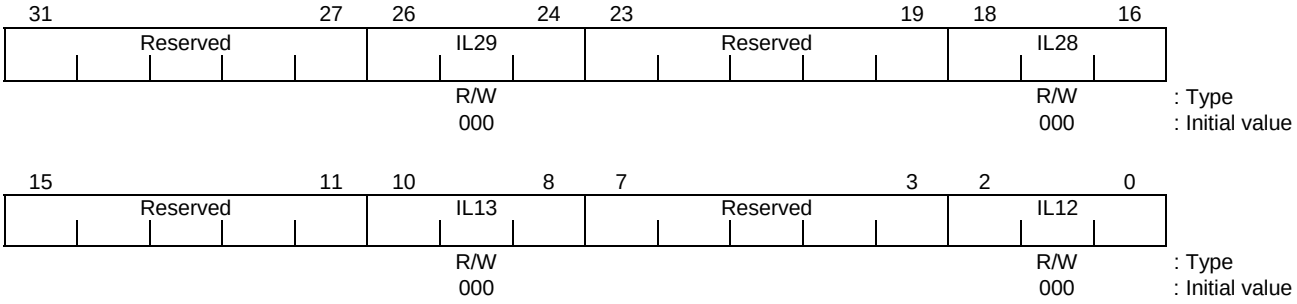
15.4.9 Interrupt Level Register 5 (IRLVL5) 0xF624



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL27	Interrupt Level 27	Interrupt Level of INT[27] (Initial value: 000, R/W) These bits specify the interrupt level of ACLCPME interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL26	Interrupt Level 26	Interrupt Level of INT[26] (Initial value: 000, R/W) These bits specify the interrupt level of ACLC interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL11	Interrupt level 11	Interrupt Level of INT[11] (Initial value: 000, R/W) These bits specify the interrupt level of NAND Flash Controller interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:0	—	Reserved	—

Figure 15.4.9 Interrupt Level Register 5

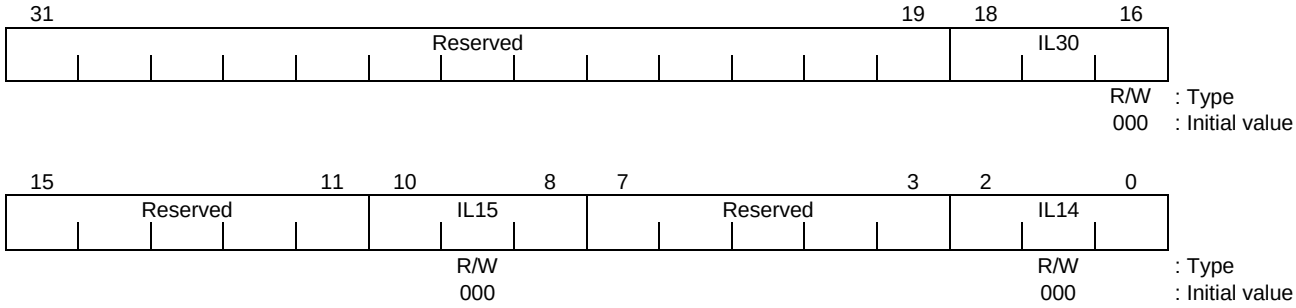
15.4.10 Interrupt Level Register 6 (IRLVL6) 0xF628



Bits	Mnemonic	Field Name	Explanation
31:27	—	Reserved	—
26:24	IL29	Interrupt Level 29	Interrupt Level of INT[29] (Initial value: 000, R/W) These bits specify the interrupt level of PCIERR interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
23:19	—	Reserved	—
18:16	IL28	Interrupt Level 28	Interrupt Level of INT[28] (Initial value: 000, R/W) These bits specify the interrupt level of CHI interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL13	Interrupt level 13	Interrupt Level of INT[13] (Initial value: 000, R/W) These bits specify the interrupt level of SIO[1] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL12	Interrupt level 12	Interrupt Level of INT[12] (Initial value: 000, R/W) These bits specify the interrupt level of SIO[0] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.10 Interrupt Level Register 6

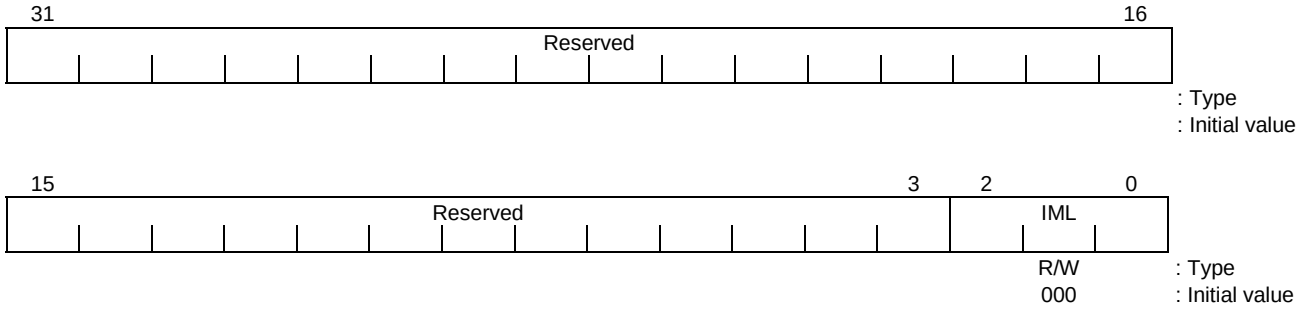
15.4.11 Interrupt Level Register 7 (IRLVL7) 0xF62C



Bits	Mnemonic	Field Name	Explanation
31:19	—	Reserved	—
18:16	IL30	Interrupt Level 30	Interrupt Level of INT[30] (Initial value: 000, R/W) These bits specify the interrupt level of PCIPME interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
15:11	—	Reserved	—
10:8	IL15	Interrupt level 15	Interrupt Level of INT[15] (Initial value: 000, R/W) These bits specify the interrupt level of DMA[1] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7
7:3	—	Reserved	—
2:0	IL14	Interrupt Level 14	Interrupt Level of INT[14] (Initial value: 000, R/W) These bits specify the interrupt level of DMA[0] interrupts. 000: Interrupt level 0 (Interrupt disabled) 001: Interrupt level 1 010: Interrupt level 2 011: Interrupt level 3 100: Interrupt level 4 101: Interrupt level 5 110: Interrupt level 6 111: Interrupt level 7

Figure 15.4.11 Interrupt Level Register 7

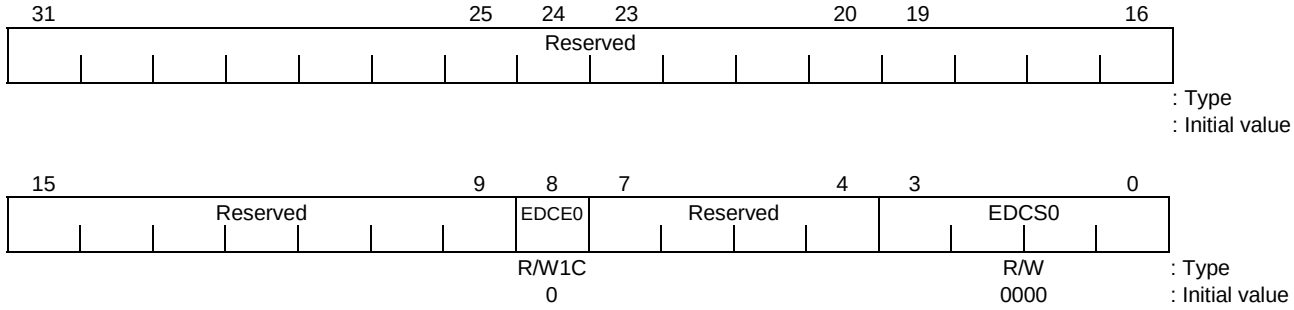
15.4.12 Interrupt Mask Level Register (IRMSK) 0xF640



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

Figure 15.4.12 Interrupt Mask Register

15.4.13 Interrupt Edge Detection Clear Register (IREDC) 0xF660



Bits	Mnemonic	Field Name	Explanation
31:9	—	Reserved	—
8	EDCE0	Edge Detection Clear Enable 0	Edge Detection Clear Enable 0 (Initial value: 0, R/W1C) Clears edge detection of interrupts specified by the EDCS0 field. 0: Does not clear. 1: Clears. Value always becomes “0” when this bit is read.
7:4	—	Reserved	—
3:0	EDCS0	Edge Detection Clear Source 0	Edge Detection Clear Source 0 (Initial value: 0x0, R/W1C) These bits specify the interrupt source to be cleared. 1111: (Reserved) 1110: (Reserved) 1101: (Reserved) 1100: (Reserved) 1011: (Reserved) 1010: (Reserved) 1001: External INT[7] interrupt 1000: External INT[6] interrupt 0111: External INT[5] interrupt 0110: External INT[4] interrupt 0101: External INT[3] interrupt 0100: External INT[2] interrupt 0011: External INT[1] interrupt 0010: External INT[0] interrupt 0001: (Reserved) 0000: (Reserved)

Figure 15.4.13 Interrupt Status Control Register

15.4.14 Interrupt Pending Register (IRPND) 0xF680

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16	
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	: Type
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	: Initial value

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

Bits	Mnemonic	Field Name	Explanation
31	—	Reserved	—
30	IS30	Interrupt Status 30	IRINTREQ[30] status (Initial value: 0, R) This bit indicates the PCIPME interrupt status. 1: Interrupt requests 0: No interrupt requests
29	IS29	Interrupt Status 29	IRINTREQ[29] status (Initial value: 0, R) This bit indicates the PCIERR error status. 1: Interrupt requests 0: No interrupt requests
28	IS28	Interrupt Status 28	IRINTREQ[28] status (Initial value: 0, R) This bit indicates the CHI interrupt status. 1: Interrupt requests 0: No interrupt requests
27	IS27	Interrupt Status 27	IRINTREQ[27] status (Initial value: 0, R) This bit indicates the ACLCPME error status. 1: Interrupt requests 0: No interrupt requests
26	IS26	Interrupt Status 26	IRINTREQ[26] status (Initial value: 0, R) This bit indicates the ACLC error status. 1: Interrupt requests 0: No interrupt requests
25	IS25	Interrupt Status 25	IRINTREQ[27] status (Initial value: 0, R) This bit indicates the RTC error status. 1: Interrupt requests 0: No interrupt requests
24	IS24	Interrupt Status 24	IRINTREQ[26] status (Initial value: 0, R) This bit indicates the SPI error status. 1: Interrupt requests 0: No interrupt requests
23	IS23	Interrupt Status 23	IRINTREQ[23] status (Initial value: 0, R) This bit indicates the TMR[2] interrupt status. 1: Interrupt requests 0: No interrupt requests
22	IS22	Interrupt Status 22	IRINTREQ[22] status (Initial value: 0, R) This bit indicates the TMR[1] interrupt status. 1: Interrupt requests 0: No interrupt requests
21	IS21	Interrupt Status 21	IRINTREQ[21] status (Initial value: 0, R) This bit indicates the TMR[0] interrupt status. 1: Interrupt requests 0: No interrupt requests

Figure 15.4.14 Interrupt Pending Status Register (1/3)

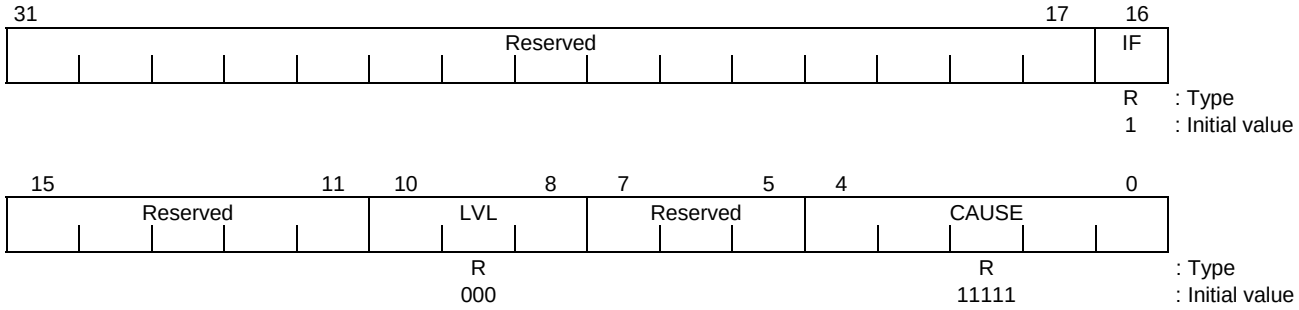
Bits	Mnemonic	Field Name	Explanation
20	IS20	Interrupt Status 20	IRINTREQ[20] status (Initial value: 0, R) This bit indicates the PCIC interrupt status. 1: Interrupt requests 0: No interrupt requests
19	IS19	Interrupt Status 19	IRINTREQ[19] status (Initial value: 0, R) This bit indicates the PDMAC interrupt status. 1: Interrupt requests 0: No interrupt requests
18	IS18	Interrupt Status 18	IRINTREQ[18] status (Initial value: 0, R) This bit indicates the IRC interrupt status. 1: Interrupt requests 0: No interrupt requests
17	IS17	Interrupt Status 17	IRINTREQ[17] status (Initial value: 0, R) This bit indicates the DMA[3] interrupt status. 1: Interrupt requests 0: No interrupt requests
16	IS16	Interrupt Status 16	IRINTREQ[16] status (Initial value: 0, R) This bit indicates the status of DMA[2] interrupts. 1: Interrupt requests 0: No interrupt requests
15	IS15	Interrupt Status 15	IRINTREQ[15] status (Initial value: 0, R) This bit indicates the status of DMA[1] interrupts. 1: Interrupt requests 0: No interrupts requests
14	IS14	Interrupt Status 14	IRINTREQ[14] status (Initial value: 0, R) This bit indicates the status of DMA[0] interrupts. 1: Interrupt requests 0: No interrupt requests
13	IS13	Interrupt Status 13	IRINTREQ[13] status (Initial value: 0, R) This bit indicates the status of SIO[1] interrupts. 1: Interrupt requests 0: No interrupt requests
12	IS12	Interrupt Status 12	IRINTREQ[12] status (Initial value: 0, R) This bit indicates the status of SIO[0] interrupts. 1: Interrupt requests 0: No interrupt requests
11	IS11	Interrupt Status 11	IRINTREQ[11] status (Initial value: 0, R) This bit indicates the status of NAND Flash Controller interrupts. 1: Interrupt requests 0: No interrupts requests
10	—	Reserved	—
9	IS9	Interrupt Status 9	IRINTREQ[9] status (Initial value: 0, R) This bit indicates the status of external INT[7] interrupts. 1: Interrupt requests 0: No interrupt requests
8	IS8	Interrupt Status 8	IRINTREQ[8] status (Initial value: 0, R) This bit indicates the status of external INT[6] interrupts. 1: Interrupt requests 0: No interrupt requests
7	IS7	Interrupt Status 7	IRINTREQ[7] status (Initial value: 0, R) This bit indicates the status of external INT[5] interrupts. 1: Interrupt requests 0: No interrupt requests
6	IS6	Interrupt Status 6	IRINTREQ[6] status (Initial value: 0, R) This bit indicates the status of external INT[4] interrupts. 1: Interrupt requests 0: No interrupt requests

Figure 15.4.14 Interrupt Pending Status Register (2/3)

Bits	Mnemonic	Field Name	Explanation
5	IS5	Interrupt Status 5	IRINTREQ[5] status (Initial value: 0, R) This bit indicates the status of external INT[3] interrupts. 1: Interrupt requests 0: No interrupt requests
4	IS4	Interrupt Status 4	IRINTREQ[4] status (Initial value: 0, R) This bit indicates the status of external INT[2] interrupts. 1: Interrupt requests 0: No interrupt requests
3	IS3	Interrupt Status 3	IRINTREQ[3] status (Initial value: 0, R) This bit indicates the status of external INT[1] interrupts. 1: Interrupt requests 0: No interrupt requests
2	IS2	Interrupt Status 2	IRINTREQ[2] status (Initial value: 0, R) This bit indicates the status of external INT[0] interrupts. 1: Interrupt requests 0: No interrupt requests
1	IS1	Interrupt Status 1	IRINTREQ[1] status (Initial value: 0, R) This bit indicates the status of TX49 Write Timeout Error interrupts. 1: Interrupt requests 0: No interrupt requests
0	—	Reserved	—

Figure 15.4.14 Interrupt Pending Status Register (3/3)

15.4.15 Interrupt Current Status Register (IRCS) 0xF6A0



Bits	Mnemonic	Field Name	Explanation
31:17	—	Reserved	—
16	IF	Interrupt Flag	Interrupt Flag (Initial value: 1, R) This bit indicates the interrupt generation status. 0: Interrupt requests have been generated. 1: Interrupt requests have not been generated
15:11	—	Reserved	—
10:8	LVL	Interrupt Level	Interrupt Level (Initial value: 000, R) These bits specify the level of the interrupt request that was reported to the TX49/H2 core. The values of these bits are undefined when there is no interrupt request. 000: Interrupt level 0 001: Interrupt level 1 : : 111: Interrupt level 7
7:5	—	Reserved	—

Figure 15.4.15 Interrupt Current Status Register (1/2)

Bits	Mnemonic	Field Name	Explanation
4:0	CAUSE	Interrupt Cause	Interrupt Cause (Initial value: 0x1F, R) These bits specify the interrupt cause that was reported to the TX49/H2 core. The values of these bits are undefined when there is no interrupt request. 00000: (Reserved) 00001: TX49 Write Timeout Error 00010: External INT[0] interrupt 00011: External INT[1] interrupt 00100: External INT[2] interrupt 00101: External INT[3] interrupt 00110: External INT[4] interrupt 00111: External INT[5] interrupt 01000: External INT[6] interrupt 01001: External INT[7] interrupt 01010: (Reserved) 01011: NAND Flash Controller interrupt 01100: SIO[0] interrupt 01101: SIO[1] interrupt 01110: DMA[0] interrupt 01111: DMA[1] interrupt 10000: DMA[2] interrupt 10001: DMA[3] interrupt 10010: IRC interrupt 10011: PDMAC interrupt 10100: PCIC interrupt 10101: TMR[0] interrupt 10110: TMR[1] interrupt 10111: TMR[2] interrupt 11000: SPI interrupt 11001: RTC interrupt 11010: ACLC interrupt 11011: ACLCPME interrupt 11100: CHI interrupt 11101: PCIERR interrupt 11110: PCIPME interrupt 11111: (Reserved)

Figure 15.4.15 Interrupt Current Status Register (2/2)

15.4.16 Interrupt Request Flag Register 0 (IRFLAG0) 0xF510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PF0 [31]	PF0 [30]	PF0 [29]	PF0 [28]	PF0 [27]	PF0 [26]	PF0 [25]	PF0 [24]	PF0 [23]	PF0 [22]	PF0 [21]	PF0 [20]	PF0 [19]	PF0 [18]	PF0 [17]	PF0 [16]
R/W															: Type
0x0000															: Initial value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PF0 [15]	PF0 [14]	PF0 [13]	PF0 [12]	PF0 [11]	PF0 [10]	PF0 [9]	PF0 [8]	PF0 [7]	PF0 [6]	PF0 [5]	PF0 [4]	PF0 [3]	PF0 [2]	PF0 [1]	PF0 [0]
R/W															: Type
0x0000															: Initial value

Bits	Mnemonic	Field Name	Explanation
31:0	PF0[31:0]	Flag 0	Interrupt Request Flag 0[31:0] (Initial value: 0x0000_0000, R/W) Changes made to this register are reflected in Flag Register 1 also since they are the same registers. Both “0” and “1” can be written to Flag Register 0.

Figure 15.4.16 Interrupt Request Flag Register 0

15.4.17 Interrupt Request Flag Register 1 (IRFLAG1) 0xF514

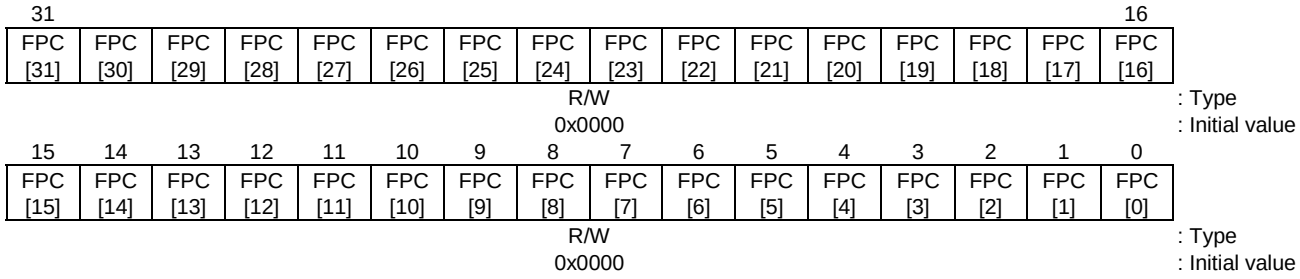
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PF1 [31]	PF1 [30]	PF1 [29]	PF1 [28]	PF1 [27]	PF1 [26]	PF1 [25]	PF1 [24]	PF1 [23]	PF1 [22]	PF1 [21]	PF1 [20]	PF1 [19]	PF1 [18]	PF1 [17]	PF1 [16]
R/W															: Type
0x0000															: Initial value

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PF1 [15]	PF1 [14]	PF1 [13]	PF1 [12]	PF1 [11]	PF1 [10]	PF1 [9]	PF1 [8]	PF1 [7]	PF1 [6]	PF1 [5]	PF1 [4]	PF1 [3]	PF1 [2]	PF1 [1]	PF1 [0]
R/W															: Type
0x0000															: Initial value

Bits	Mnemonic	Field Name	Explanation
31:0	PF1[31:0]	Flag 1	Interrupt Request Flag 1[31:0] (Initial value: 0x0000_0000, R/W) Changes made to this register are reflected in Flag Register 0 also since they are the same registers. Writes to Flag Register 1 operate as follows: Write Write from the TX49/H2 core 1: Set the flag bit 0: No change Write from other devices (DMAC, PCIC) 1: Clear the flag bit 0: No change Read: Read the flag bit

Figure 15.4.17 Interrupt Request Flag Register 1

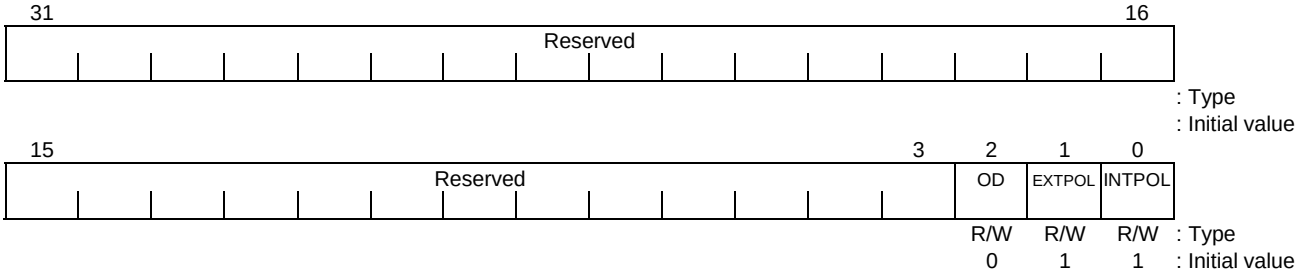
15.4.18 Interrupt Request Polarity Control Register (IRPOL) 0xF518



Bits	Mnemonic	Field Name	Explanation
31:0	FPC[31:0]	Flag Polarity Control	Flag Polarity Control[31:0] (Initial value: 0x0000_0000, R/W) These bits specify the polarity of the flag bit that generated the interrupt. An interrupt request is generated when the XOR of the FPC bit and the flag bit is “1.” Flag bit (PF) FPC bit Interrupt request 0 0 No 0 1 Yes 1 0 Yes 1 1 No

Figure 15.4.18 Interrupt Request Polarity Control Register

15.4.19 Interrupt Request Control Register (IRRCNT) 0xF51C



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

Figure 15.4.19 Interrupt Request Control Register

15.4.20 Interrupt Request Internal Interrupt Mask Register (IRMASKINT) 0xF520

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MINT [31]	MINT [30]	MINT [29]	MINT [28]	MINT [27]	MINT [26]	MINT [25]	MINT [24]	MINT [23]	MINT [22]	MINT [21]	MINT [20]	MINT [19]	MINT [18]	MINT [17]	MINT [16]	
R/W																: Type
0x0000																: Initial value
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MINT [15]	MINT [14]	MINT [13]	MINT [12]	MINT [11]	MINT [10]	MINT [9]	MINT [8]	MINT [7]	MINT [6]	MINT [5]	MINT [4]	MINT [3]	MINT [2]	MINT [1]	MINT [0]	
R/W																: Type
0x0000																: Initial value

Bits	Mnemonic	Field Name	Explanation
31:0	MINT[31:0]	Internal Request Mask	Internal Interrupt Mask (Initial value: 0x0000_0000, R/W) These bits specify whether to use the corresponding flag bit as an internal interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (Reset) 1: Do not mask

Figure 15.4.20 Interrupt Request Internal Interrupt Mask Register

15.4.21 Interrupt Request External Interrupt Mask Register (IRMASKEXT) 0xF524

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MEXT [31]	MEXT [30]	MEXT [29]	MEXT [28]	MEXT [27]	MEXT [26]	MEXT [25]	MEXT [24]	MEXT [23]	MEXT [22]	MEXT [21]	MEXT [20]	MEXT [19]	MEXT [18]	MEXT [17]	MEXT [16]	
R/W																: Type
0x0000																: Initial value
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MEXT [15]	MEXT [14]	MEXT [13]	MEXT [12]	MEXT [11]	MEXT [10]	MEXT [9]	MEXT [8]	MEXT [7]	MEXT [6]	MEXT [5]	MEXT [4]	MEXT [3]	MEXT [2]	MEXT [1]	MEXT [0]	
R/W																: Type
0x0000																: Initial value

Bits	Mnemonic	Field Name	Explanation
31:0	MEXT[31:0]	External Request Mask	External Interrupt Mask (Initial value: 0x0000_0000, R/W) These bits specify whether to use the corresponding flag bit as an external interrupt cause. Interrupt causes are masked when this bit is “0.” 0: Mask (reset) 1: Do not mask

Figure 15.4.21 Interrupt Request External Interrupt Mask Register

16. CHI Module

16.1 Characteristics

The CHI Module within the TX4925 contains holding registers, shift registers, DMA support, and other logic to support interfacing to external full-duplex serial TDM communication peripherals, including ISDN communication devices and PCM/TDM serial highways. The TX4925 implementation of the CHI Module is based on the Concentration Highway Interface standard specified by Intel and AT&T, which is intended to allow glueless interface to various TDM highways used by numerous commercial products. The TX4925 CHI Module can also be used to support an intra-system high-speed serial DMA channel within a PDA system.

The TX4925 CHI Module utilizes a 4-signal interface consisting of clock, sync, transmit serial data, and receive serial data. The data is organized as a TDM format, with up to 64 timeslots (nominally 8 bits each) per frame, with a nominal frame rate of 8 kHz (8 bits × 8 kHz = 64 kbps nominal data rate per channel). Up to four input timeslots and up to four output timeslots can be independently selected in each half-frame. The CHI Module does not handle the timeslots which are not selected for input or output. The number of timeslots and the data rate (up to 4.096 Mbps) and frame rate (up to 64 kHz, depending on the system configuration) are programmable, providing flexibility for supporting various TDM communication peripherals. These timeslots are commonly used to carry voice, data, or control and status information.

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

16.2 Block Diagram

See Figure 16.2.1 for a block diagram of the CHI Module.

The CHI Module consists of holding registers (both transmit and receive), shift registers (both transmit and receive), DMA support, and other logic to support interfacing to various types of TDM highways.

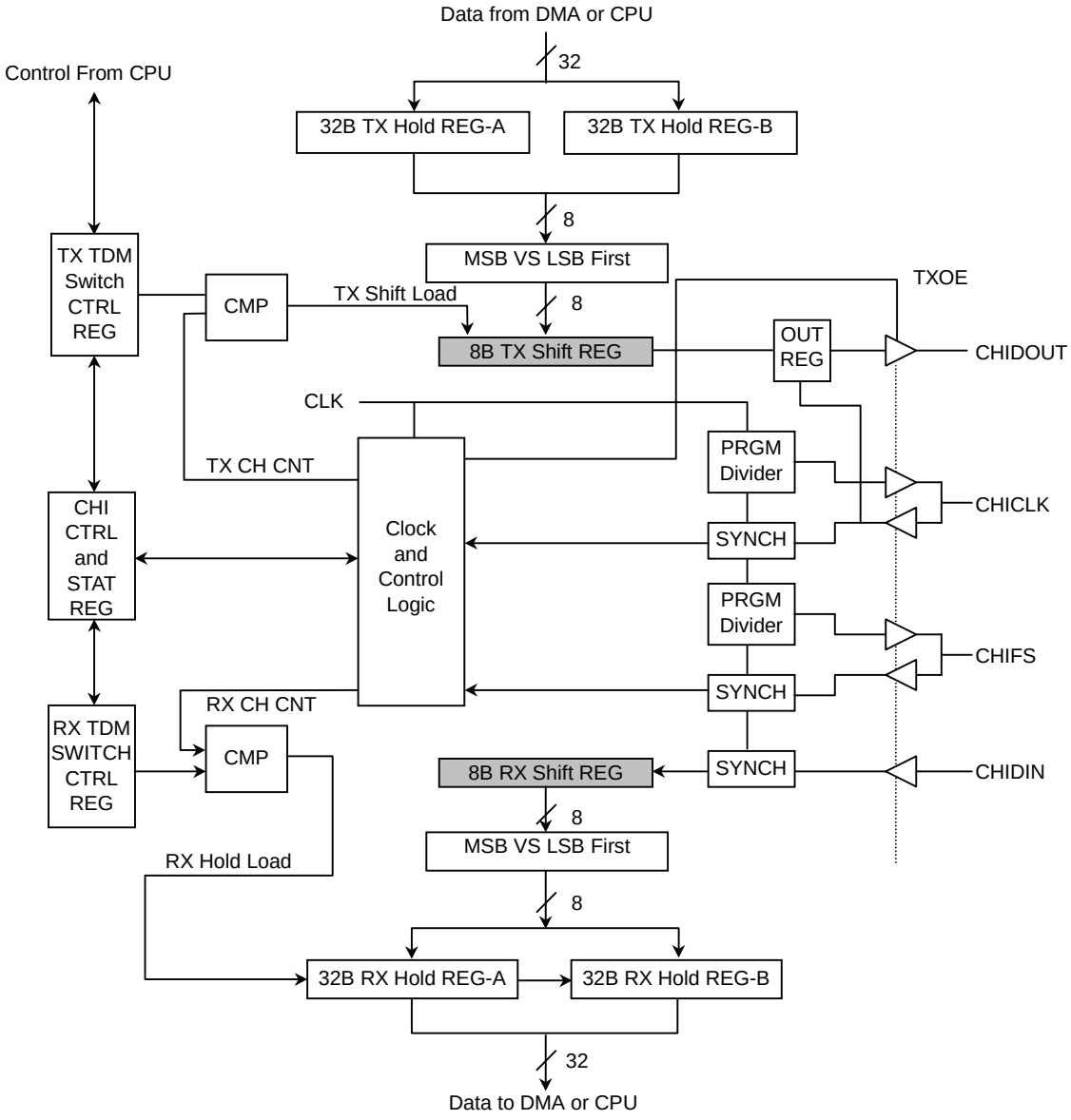


Figure 16.2.1 CHI Module Block Diagram

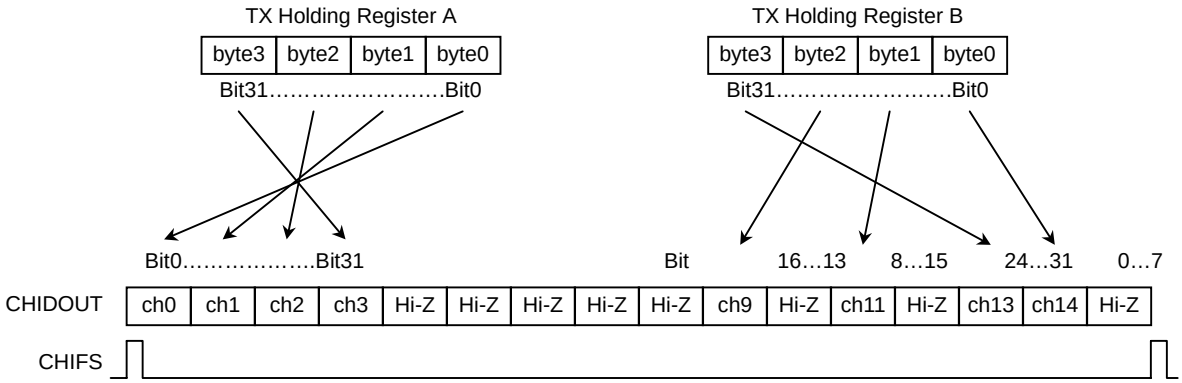
16.3 Detailed Explanation

16.3.1 Transmitter

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

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

Figure 16.3.1 shows how the transmit TDM switch works, for example, with 16 total timeslots. Please refer to “16.4 Registers”.



CHINCHAN[4:0]	0x07
TXMSBFIRST	0
CHI Pointer Enable	0xFFxx_0000
CHI Transmit Pointer A	0x0302_0100
CHI Transmit Pointer B	0x0501_0306

Figure 16.3.1 Transmit TDM Switch Example

16.3.2 Receiver

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

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

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

16.3.3 Clock and Control Generation

The CHI Module contains several programmable counters which are used to generate the various CHI internal and external control signals and clocks. See Figure 16.3.3 for a block diagram of the CHI clock and control generation circuit. As mentioned previously, CHICLK can be configured as either an output (master mode) or input (slave mode). As an output, CHICLK is derived by dividing down from IMBUSCLKF. In this mode, all CHI clocks are then synchronously locked to the main TX4925 system clock. As an input, CHICLK is generated from an external clock source, which is asynchronous with respect to IMBUSCLKF. The TX4925 CHI Module utilizes a digital-PLL circuit to stay “locked” to the external source, while still operating internally using IMBUSCLKF CHIDIN and CHIDOUT are also synchronized between IMBUSCLKF and the externally-supplied CHICLK.

CHIFS can also be configured as either an output (master mode) or input (slave mode). As an output, CHIFS is derived by dividing down from CHICLK. For this mode, the CHIFS pulse width and polarity is also programmable. As an input, CHIFS is generated from an external sync source. The TX4925 CHI Module utilizes a digital-PLL circuit to stay “locked” to the external sync source, while still operating internally using IMBUSCLKF.

The programmable receive and transmit sync delay counters shown in See Figure 16.3.2 are used to implement the bit offset feature described earlier. The bit offset control bits determine the number of clock cycles between the start of timeslot 0 and CHIFS. The receive and transmit sync delay counters are independent from each other, such that the receive and transmit serial data streams can have different bit offsets.

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

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

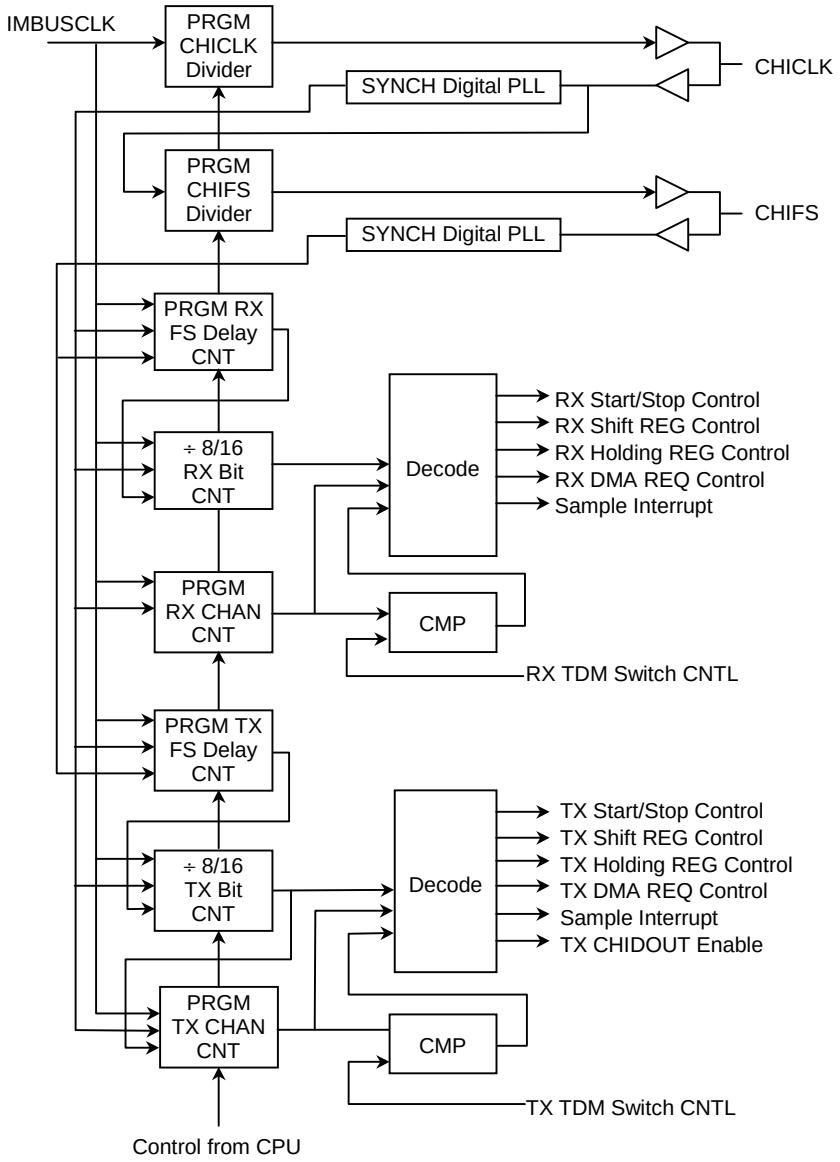


Figure 16.3.2 CHI Clock and Control Generation

16.3.4 DMA Address Generation

The CHI Module provides support for 2 full-duplex DMA channels: receive and transmit. The circuit used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts is shown in Figure 16.3.3.

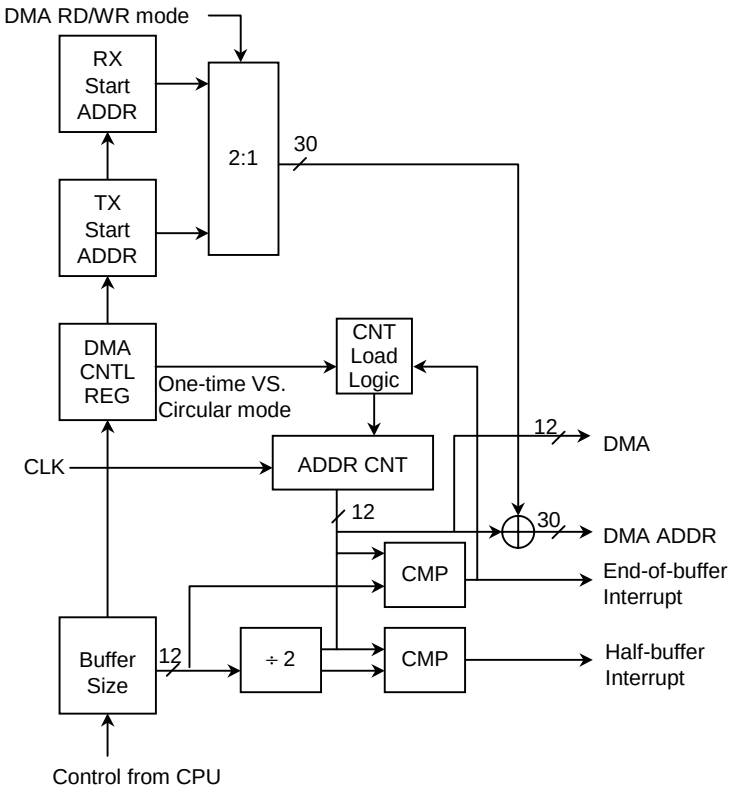


Figure 16.3.3 CHI DMA Address Generation

The DMA buffer size is programmable (from a minimum of 16bytes up to a maximum of 16 Kbytes) and the receive and transmit buffer start addresses are also programmable (anywhere over the full 32-bit address space). Because there are separate start addresses, the receive and transmit buffers can be configured to either reside in different memory spaces or share the same memory space. But in the loopback mode, it can only share the same memory space and this ordering allows a receive-to-transmit immediate via the DMA buffer. Thus, received samples are written to the DMA buffer location immediately after transmit samples were read from that same location (which then became immediately available). This ordering allows a single circular DMA buffer to be used for both transmit and receive samples.

The DMA buffers can be configured in a circular buffer mode or a one-time buffer mode. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TX4925's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

Because the CHI Module reads and writes a byte at a time between the shift registers and the 32 bit holding registers, the software must pack and unpack these bytes to and from the 32 bit words in memory in order to multiplex and demultiplex each channel for processing. Table 16.3.1 shows the format and organization of the CHI channels within memory for DMA mode. Consecutive byte samples for a given channel reside in memory every 8th byte.

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

Table 16.3.1 CHI DMA Memory Organization

(Relative) Memory Address	+0	+1	+2	+3
+0x0	buffA, byte3 sample 0	buffA, byte2 sample 0	buffA, byte1 sample 0	buffA, byte0 sample 0
+0x4	buffB, byte3 sample 0	buffB, byte2 sample 0	buffB, byte1 sample 0	buffB, byte0 sample 0
+0x8	buffA, byte3 sample 1	buffA, byte2 sample 1	buffA, byte1 sample 1	buffA, byte0 sample 1
+0xC	buffB, byte3 sample 1	buffB, byte2 sample 1	buffB, byte1 sample 1	buffB, byte0 sample 1
+0x10	buffA, byte3 sample 2	buffA, byte2 sample 2	buffA, byte1 sample 2	buffA, byte0 sample 2
etc.				

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

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

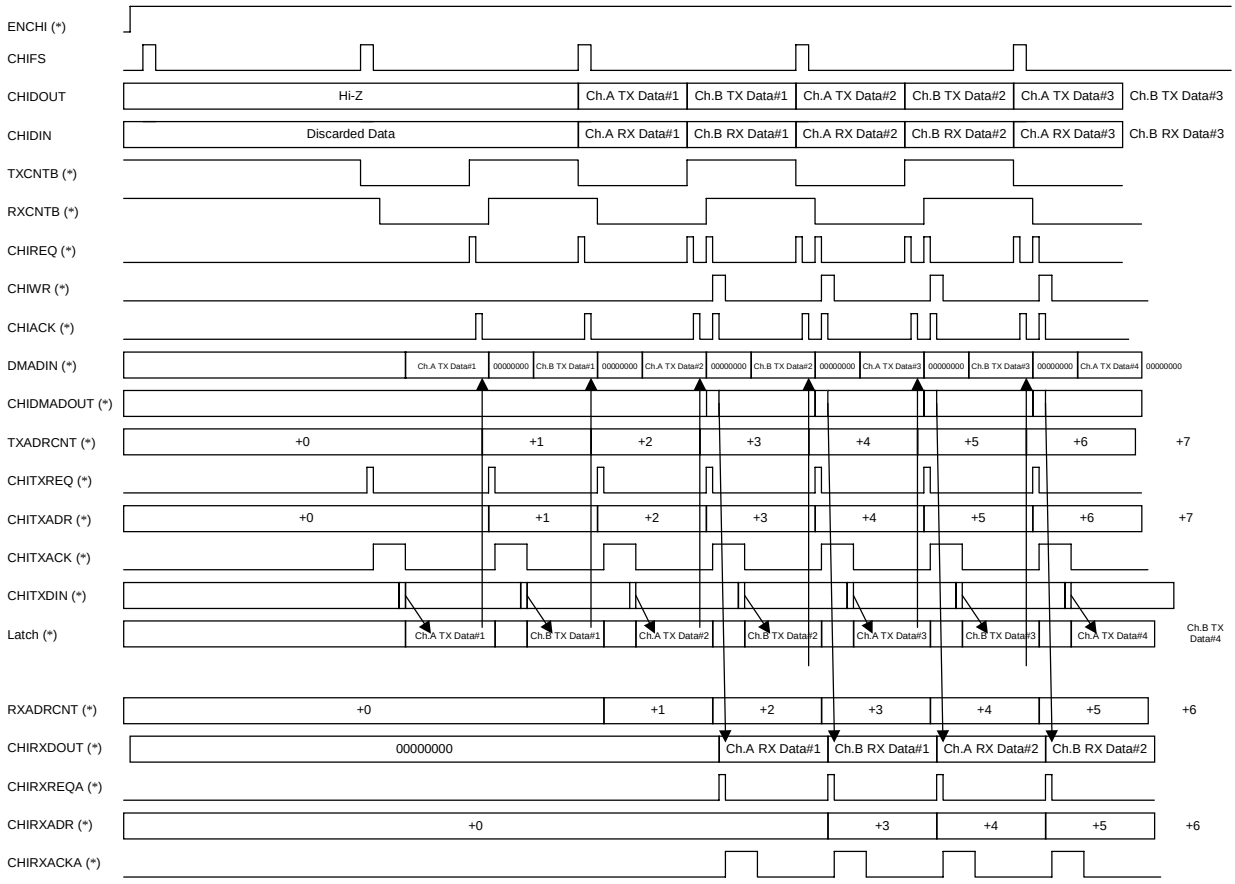
Half-buffer and end-of-buffer DMA address counter interrupts are available, allowing the CPU to minimize overhead and utilize the DMA buffer in a ping-pong fashion. For transmit mode, the CPU can use these interrupts to fill or write one half of the buffer while the other half is being emptied by the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

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

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

16.3.5 Timing Diagram

Figure 16.3.4 shows timing diagram for DMA transfer.



(*): Internal signal

Figure 16.3.4 Timing Diagram for DMA Transfer.

16.3.6 Interrupts

The CHI module has eight types interrupt sources. OR signal of them connects to the internal Interrupt Controller (IRC). Please check CHI Interrupt Status Register (CHIINT) to know which type of interrupt occurred.

Type	Status Bits	Mask-able Bit
CHIBUSERROR	BUSI	BUSIE
CHI0_5	05I	05IE
CHI1_0	10I	10IE
CHIDMACNT	DCI	DCIE
CHIININTA	INAI	INAIE
CHIININTB	INBI	INBIE
CHIACT	ACTI	ACTIE
CHIERR	ERRI	ERRIE

CHIBUSERRORINT:

Issues an interrupt whenever the CHI DMA has bus error.

CHI0_5INT:

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

CHI1_0INT:

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

CHIDMACNTINT:

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

CHIININTA:

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

CHIININTB:

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

CHIACTINT:

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

CHIERRINT:

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

16.3.7 Frame Structure and Serial Timing

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

The TX4925 CHI Module supports a master or slave mode for both the clock (CHICLK) and sync (CHIFS). For the master mode, the TX4925 contains programmable dividers for generating the clock and/or sync signal, synchronously dividing down from the main core clock (CLK). For the slave mode, TX4925 accepts external clock and/or sync signals and utilizes “digital-PLL” type circuitry to stayed “locked” to the external source.

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

1x versus 2x clock modes for CHICLK (2x clock mode uses two CHICLK periods per data bit)

MSB-first versus LSB-first serial formats for transmit and receive

rising versus falling edge (polarity) used for frame sync triggering

CHIFS signal can be sampled on either rising or falling edge of CHICLK

CHIDIN receive data can be sampled on either rising or falling edge of CHICLK

CHIDOUT transmit data can be pushed on either rising or falling edge of CHICLK

CHIDIN receive data can have programmable bit offset (timeslot 0 offset from CHIFS)

CHIDOUT transmit data can have programmable bit offset (timeslot 0 offset from CHIFS)

CHIDOUT transmit data (tri-state) output buffer enable is dynamically asserted for only active timeslots; for sleep mode CHIDOUT is always tri-stated

for CHIFS master mode, the CHIFS pulse width and polarity is programmable

The TX4925 CHI Module allows for a programmable bit offset for both CHIDIN and CHIDOUT, which is related to the number of clock cycles between the start of timeslot 0 and CHIFS. This flexibility allows the TX4925 CHI Module to support a wide variety of interface clock and sync timing formats. The control bits for controlling the CHIDIN bit offset are CHIRXBOFF[3:0], while the control bits for controlling the CHIDOUT bit offset are CHITXBOFF[3:0].

Table 16.3.2 and Table 16.3.3 shows a summary matrix for the values of CERX and CETX for all possible settings of CHIRXBOFF and CHITXBOFF, respectively. These values are shown for various configurations of CHICLK mode (1x versus 2x), CHIFSEDGE, CHIRXEDGE, and CHITXEDGE. The CHIFSEDGE settings determine whether to use the rising edge (CHIFSEDGE = 1) or falling edge (CHIFSEDGE = 0) of CHICLK to sample CHIFS. The CHIRXEDGE settings determine whether to use the rising edge (CHIRXEDGE = 1) or falling edge (CHIRXEDGE = 0) of CHICLK to sample the CHIDIN input. The CHITXEDGE settings determine whether to use the rising edge (CHITXEDGE = 1) or falling edge (CHITXEDGE = 0) of CHICLK to push the CHIDOUT output. CERX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDIN is sampled. CETX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDOUT is pushed. The CHI frame structure and bit offsets are shown in Figure 16.3.5 to 16.3.11 for various clock and sync configurations.

Table 16.3.2 CERX Values for CHIRXBOFF Versus Clock and Edge Configurations

CHICLK Mode	CHIFS-Edge	CHIRX-Edge	CHIRXBOFF															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	0	1	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
1x	1	0	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
2x	1	1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
2x	0	1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
2x	1	0	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31

Table 16.3.3 CETX Values for CHITXBOFF Versus Clock and Edge Configurations

CHICLK Mode	CHIFS-Edge	CHIRX-Edge	CHITXBOFF															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1x	0	1	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
1x	1	0	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	0	0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
2x	1	1	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
2x	0	1	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
2x	1	0	–1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29

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

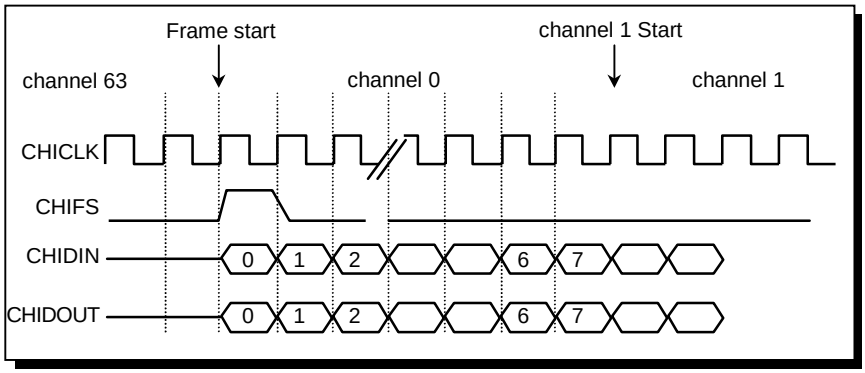


Figure 16.3.5 CHI Frame Structure Example

CHICLK 1X mode

CHIFS sampled on falling edge

CHIDIN sampled on falling edge; RXBOFF = 0; CERX = 0

CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = 1

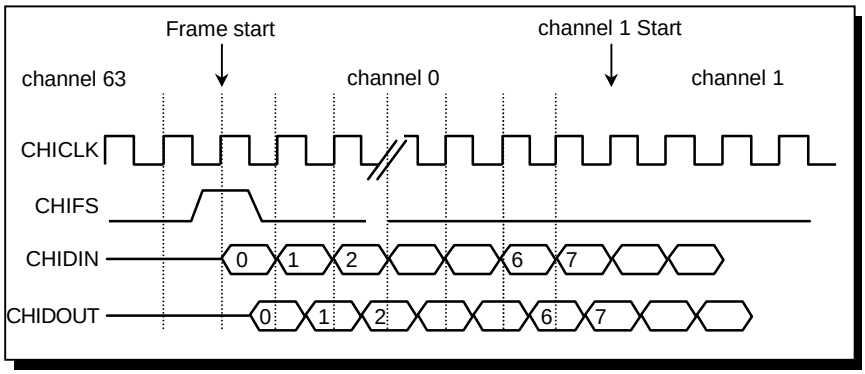


Figure 16.3.6 CHI Frame Structure Example

CHICLK 1X mode

CHIFS sampled on rising edge

CHIDIN sampled on falling edge; RXBOFF = 1; CERX = 1

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

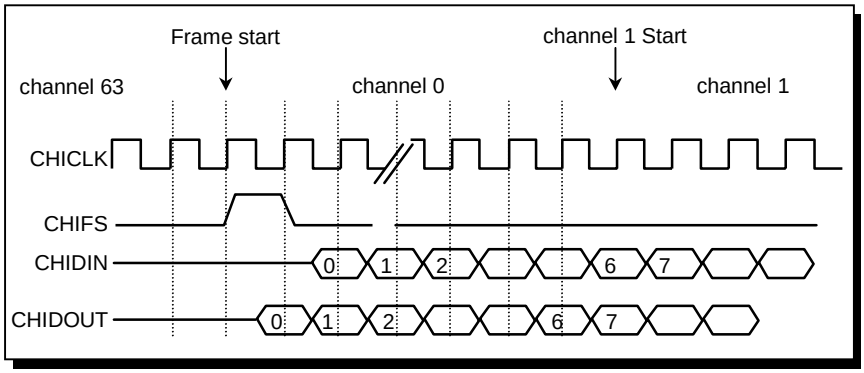


Figure 16.3.7 CHI Frame Structure Example

CHICLK 1X mode

CHIFS sampled on falling edge

CHIDIN sampled on rising edge; RXBOFF = 2; CERX = 3

CHIDOUT pushed on falling edge; TXBOFF = 0; CETX = 0

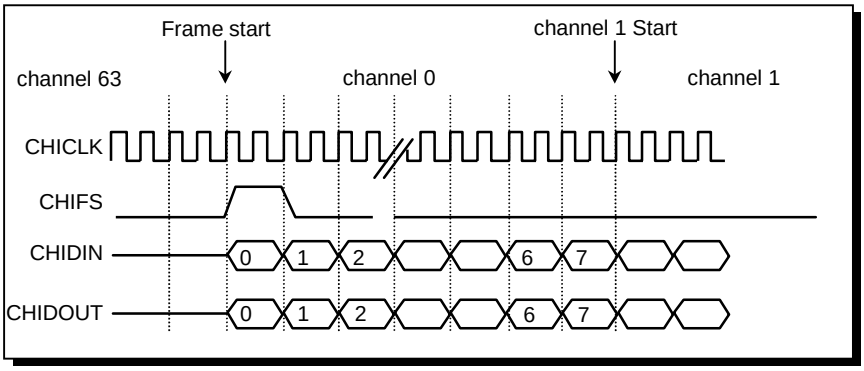


Figure 16.3.8 CHI Frame Structure Example

CHICLK 2X mode

CHIFS sampled on falling edge

CHIDIN sampled on rising edge; RXBOFF = 0; CERX = 1

CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = 1

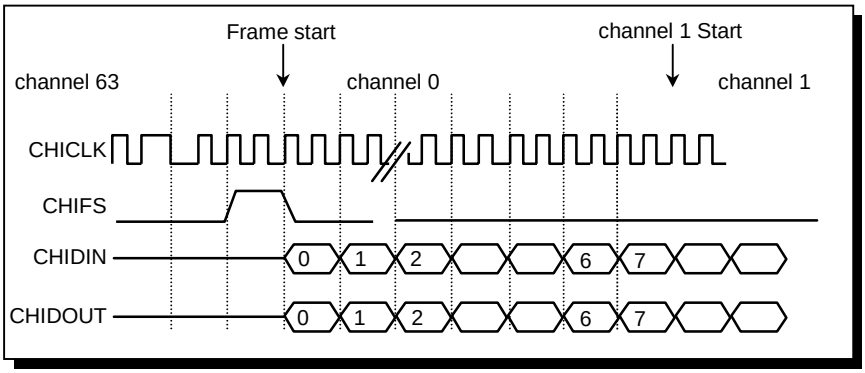


Figure 16.3.9 CHI Frame Structure Example

CHICLK 2X mode

CHIFS sampled on falling edge

CHIDIN sampled on rising edge; RXBOFF = 2; CERX = 5

CHIDOUT pushed on rising edge; TXBOFF = 2; CETX = 3

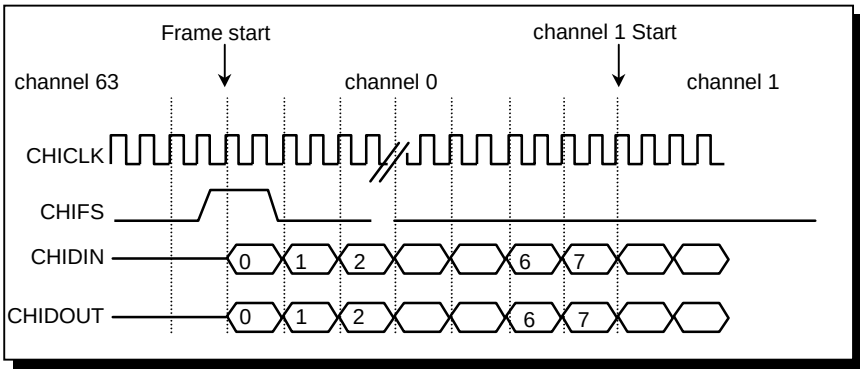


Figure 16.3.10 CHI Frame Structure Example

CHICLK 2X mode

CHIFS sampled on rising edge

CHIDIN sampled on rising edge; RXBOFF = 0; CERX = 2

CHIDOUT pushed on rising edge; TXBOFF = 0; CETX = 0

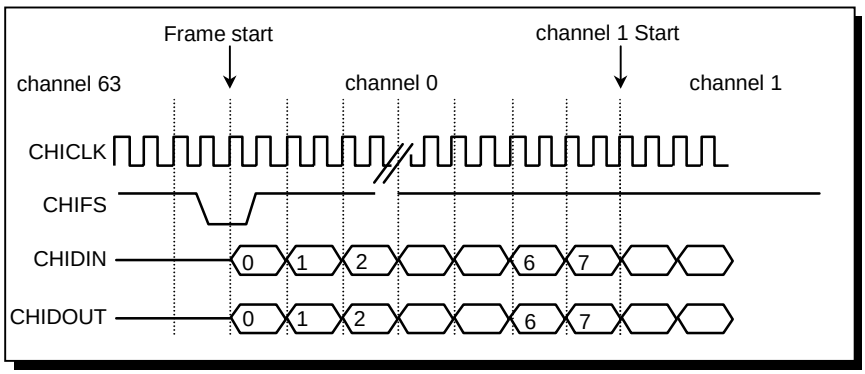


Figure 16.3.11 CHI Frame Structure Example

CHICLK 2X mode

CHIFS sampled on falling edge

CHIRXFSPOL = 1 (negative polarity)

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

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

16.3.8 Configurations

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

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

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

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

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

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

Table 16.3.4 shows a summary matrix of several example CHI configurations and their associated parameters.

Table 16.3.4 Example CHI Configurations (Table Values Based on IMBUSCLKF = 32.256 MHz)

CHICLK Divide	Time Slots	CHICLK Mode	fs	CHICLK Rate	Data Rate	Comments
4	64	2x	8 kHz	8.064 MHz	4.032 Mbps	slave mode only
6	48	2x	8 kHz	5.376 MHz	2.688 Mbps	GCI format
8	64	1x	8 kHz	4.032 MHz	4.032 Mbps	CHI format
8	8	1x	64 kHz	4.032 MHz	4.032 Mbps	hi-speed mode
9	64	1x	7.2 kHz	3.584 MHz	3.584 Mbps	
11	24	2x	8 kHz	2.932 MHz	1.466 Mbps	
11	48	1x	8 kHz	2.932 MHz	2.932 Mbps	
16	32	1x	8 kHz	2.016 MHz	2.016 Mbps	CEPT PCM format
21	12	2x	8 kHz	1.536 MHz	768 kbps	IOM-2 format
63	8	1x	8 kHz	512 kHz	512 kbps	K2, SLD formats

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

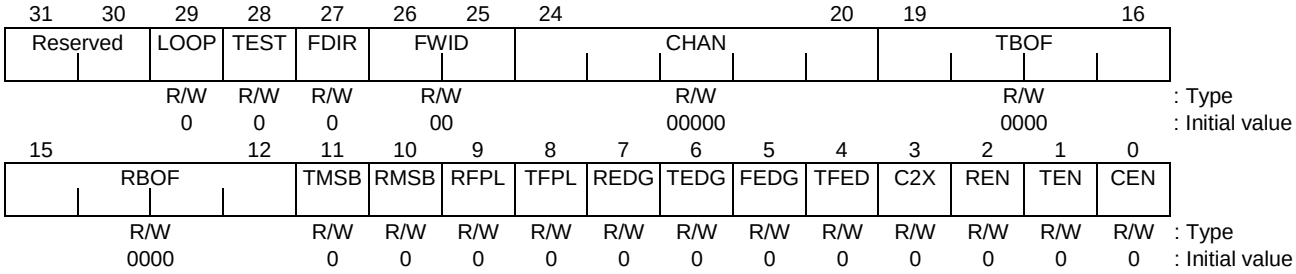
16.4 Registers

All registers should be accessed only as full word (32-bit) accesses. Any other type of access produces an undefined result. Please write “0” to the undefined bit.

Table 16.4.1 CHI Module Registers

Reference	Offset Address	Bit Width	Register Symbol	Register Name
16.4.1	0xA800	32 bit	CTRL	CHI Control Register
16.4.2	0xA804	32 bit	PNTREN	CHI Pointer Enable Register
16.4.3	0xA808	32 bit	RXPTRA	CHI Receive Pointer A Register
16.4.4	0xA80C	32 bit	RXPTRB	CHI Receive Pointer B Register
16.4.5	0xA810	32 bit	TXPTRA	CHI Transmit Pointer A Register
16.4.6	0xA814	32 bit	TXPTRB	CHI Transmit Pointer B Register
16.4.7	0xA818	32 bit	CHISIZE	CHI SIZE Register
16.4.8	0xA81C	32 bit	RXSTRT	CHI RX Start Register
16.4.9	0xA820	32 bit	TXSTRT	CHI TX Start Register
16.4.10	0xA824	32 bit	HOLD	CHI TX Hold Register (write only)
16.4.10	0xA824	32 bit	HOLD	CHI RX Hold Register (read only)
16.4.12	0xA828	32 bit	CLOCK	CHI Clock Register
16.4.13	0xA82C	32 bit	CHIINTE	CHI Interrupt Enable Register.
16.4.14	0xA830	32 bit	CHIINT	CHI Interrupt Status Register

16.4.1 CHI Control Register (CTRL) 0xA800



Bits	Mnemonic	Field Name	Description
31:30	—	Reserved	—
29	LOOP	CHILOOP	CHILOOP bit (Initial value: 0, R/W) This bit is used for IC testing and should not be set. Setting this bit to a logic “1” will cause the CHI serial transmitted data to be internally looped back to the CHI serial receive data path. The data is inverted when this mode is selected. Clearing this bit to a logic “0” selects the normal CHIDIN pin as the CHI serial receive data source.
28	TEST	CHIENTEST	CHIENTEST bit (Initial value: 0, R/W) This bit is used for IC testing and should not be set.
27	FDIR	CHIFSDIR	CHIFSDIR bit (Initial value: 0, R/W) This bit controls the direction of the CHIFS pin. 0: CHIFS to be an input.(CHI sync slave mode) Note* 1: CHIFS to be an output.(CHI sync master mode) Note that CHIRXFSPOL and CHIFSEDGE bits must be set to proper values even when CHIFSDIR is set to a logic “1” (CHI sync master mode).
26:25	FWID[1:0]	CHIFSWIDTH	CHIFSWIDTH bits (Initial value: 00, R/W) These bits are used to select pulse width for the CHIFS signal, relevant whenever the CHI Module is configured as master mode. The pulse width is counted by data bit width. (In clk2x mode, two CHICLKs correspond to one data bit.) The available CHIFS pulse widths are as follows 00: 1 bit wide 01: 2 bits wides 10: 1 byte wide 11: half-frame wide
24:20	CHAN[4:0]	CHINCHAN	CHINCHAN bits (Initial value: 00000, R/W) These bits are used to program the number of 8-bit channel timeslots per half-frame, up to 32 total per half-frame. The value loaded for CHINCHAN is the desired number of channels-1.
19:16	TBOF[3:0]	CHTXBOFF	CHITXBOFF bits (Initial value: 0000, R/W) These bits select the transmit data programmable bit offset, which is related to the number of clocks from the start of timeslot 0 (1st timeslot) transmit data to the CHIFS edge used to trigger the start of each CHI frame. The value loaded for CHITXBOFF must be chosen from Table 16.3.3 according to the configuration.
15:12	RBOF[3:0]	CHIRXBOFF	CHIRXBOFF bits (Initial value: 0000, R/W) These bits select the receive data programmable bit offset, which is related to the number of clocks from the start of timeslot 0 (1st timeslot) receive data to the CHIFS edge used to trigger the start of each CHI frame. The value loaded for CHIRXBOFF must be chosen from Table 16.3.2 according to the configuration.
11	TMSB	TXMSBFIRST	TXMSBFIRST bit (Initial value: 0, R/W) This bit selects between MSB-first and LSB-first serial data formats for each byte of the CHI transmit data. 0: LSB-first 1: MSB-first

Figure 16.4.1 Control Register (CTRLREG) (1/3)

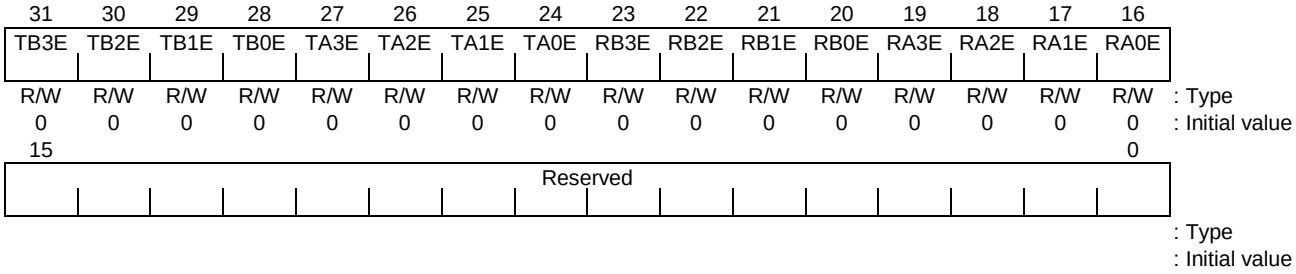
Bits	Mnemonic	Field Name	Description
10	RMSB	RXMSBFIRST	RXMSBFIRST bit (Initial value: 0, R/W) This bit selects between MSB-first and LSB-first serial data formats for each byte of the CHI receive data. 0: LSB-first 1: MSB-first
9	RFPL	CHIRXFSPOL	CHIRXFSPOL bit (Initial value: 0, R/W) This bit selects between positive (active high) or negative (active low) polarity for the received CHIFS signal pulse. This bit must be properly set in CHI sync master mode, as well as in CHI sync slave mode; In CHI sync master mode, CHIRXFSPOL must be equal to CHITXFSPOL. 0: positive polarity for the CHIFS pulse. (the rising edge of the CHIFS pulse is used to trigger the start of the CHI frame period.) 1: negative polarity for the CHIFS pulse. (the falling edge of the CHIFS pulse is used to trigger the start of the CHI frame period.)
8	TFPL	CHITXFSPOL	CHITXFSPOL bit (Initial value: 0, R/W) This bit selects between positive (active high) or negative (active low) polarity for the transmitted CHIFS signal pulse, relevant whenever the CHI Module is configured as master mode. 0: positive polarity for the CHIFS pulse 1: negative polarity for the CHIFS pulse
7	REDG	CHIRXEDGE	CHIRXEDGE bit (Initial value: 0, R/W) This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive data CHIDIN. CHIDIN must be stable before and after the specified CHICLK edge. 0: falling edge 1: rising edge
6	TEDG	CHITXEDGE	CHITXEDGE bit (Initial value: 0, R/W) This bit selects whether to use either the rising edge or falling edge of CHICLK to clock out the transmit data CHIDOUT. 0: falling edge 1: rising edge
5	FEDG	CHIFSEDGE	CHIFSEDGE bit (Initial value: 0, R/W) This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive frame sync CHIFS. This bit must be properly set in CHI sync master mode, as well as in CHI sync slave mode. CHIFS must be stable before and after the specified CHICLK edge; In CHI sync master mode, CHIFSEDGE must be equal to inverted CHITXFSEDGE. 0: falling edge 1: rising edge
4	TFED	CHITXFSEDGE	CHITXFSEDGE (Initial value: 0, R/W) This bit selects whether to use either the rising edge or falling edge of CHICLK to clock out the transmit frame sync CHIFS, relevant whenever the CHI Module is configured as master mode. 0: falling edge 1: rising edge
3	C2X	CHICLK2XMODE	CHICLK2XMODE bit (Initial value: 0, R/W) This bit selects between 1x and 2x clock modes. 0: 1x clock mode (CHICLK frequency equals the serial data bit rate) 1: 2x clock mode (CHICLK frequency equals twice the serial data bit rate)
2	REN	CHIRXEN	CHIRXEN bit (Initial value: 0, R/W) This bit is used to enable/disable CHI receive processing in the direct CPU read/write mode, where the CPU reads the received data through the CHI RX holding register. This bit has no effect when RX DMA is enabled. 0: Disable (all received data to not be processed by the CHI module) 1: Enable

Figure 16.4.1 Control Register (CTRLREG) (2/3)

Bits	Mnemonic	Field Name	Description
1	TEN	CHITXEN	CHITXEN bit (Initial value: 0, R/W) This bit is used to enable/disable CHI transmit processing in the direct CPU read/write mode, where the CPU writes the data to be transmitted through the CHI TX holding register. This bit has no effect when TX DMA is enabled. 0: Disable (the CHI serial transmitted data to be tri-stated) 1: Enable
0	CEN	ENCHI	ENCHI bit (Initial value: 0, R/W) This bit is used to enable/disable the CHI module. Setting this bit to a logic “1” enables the CHI module. Clearing this bit to a logic “0” disables the CHI Module and keeps the module in a reset state but gives no effect on the CHI Control Register. 0: Disable (CHI Module and keeps the module in a reset state but gives no effect on the CHI Control Register) 1: Enable To begin the CHI operation, follow the procedure below. (1) Set up all the configuration registers except CHIRXEN, CHITXEN, ENDMARXCHI, ENDMATXCHI, and ENCHI bits. (2) Set either CHIRXEN or ENDMARXCHI, and/or, either CHITXEN or ENDMATXCHI, to a logic “1”. (3) Set ENCHI to a logic “1”. To finish the CHI operation. (1) Clear ENCHI to a logic “0”. (2) Clear ENDMARXCHI and/or ENDMATXCHI to a logic “0”, if they were previously set.

Figure 16.4.1 Control Register (CTRLREG) (3/3)

16.4.2 CHI Pointer Enable Register (PNTREN) 0xA804



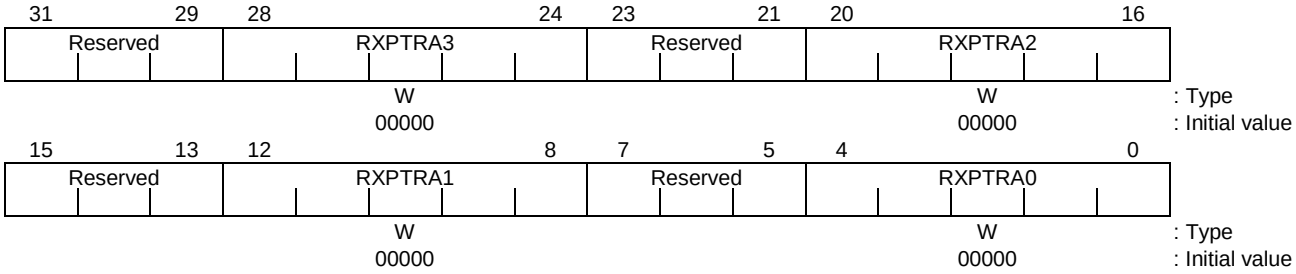
Bits	Mnemonic	Field Name	Description
31	TB3E	CHITXPTRB3EN	CHITXPTRB3EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB3. 0: Disable 1: Enable
30	TB2E	CHITXPTRB2EN	CHITXPTRB2EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB2. 0: Disable 1: Enable
29	TB1E	CHITXPTRB1EN	CHITXPTRB1EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB1. 0: Disable 1: Enable
28	TB0E	CHITXPTRB0EN	CHITXPTRB0EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRB0. 0: Disable 1: Enable
27	TA3E	CHITXPTRA3EN	CHITXPTRA3EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA3. 0: Disable 1: Enable
26	TA2E	CHITXPTRA2EN	CHITXPTRA2EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA2. 0: Disable 1: Enable
25	TA1E	CHITXPTRA1EN	CHITXPTRA1EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTRA1. 0: Disable 1: Enable
24	TA0E	CHITXPTRA0EN	CHITXPTRA0EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the transmit channel pointed to by the TDM switch pointer CHITXPTAB0. 0: Disable 1: Enable

Figure 16.4.2 CHI Pointer Enable Register (PNTREN) (1/2)

Bits	Mnemonic	Field Name	Description
23	RB3E	CHIRXPTRB3EN	CHIRXPTRB3EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB3 0: Disable 1: Enable
22	RB2E	CHIRXPTRB2EN	CHIRXPTRB2EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB2 0: Disable 1: Enable
21	RB1E	CHIRXPTRB1EN	CHIRXPTRB1EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB1 0: Disable 1: Enable
20	RB0E	CHIRXPTRB0EN	CHIRXPTRB0EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRB0 0: Disable 1: Enable
19	RA3E	CHIRXPTRA3EN	CHIRXPTRA3EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA3 0: Disable 1: Enable
18	RA2E	CHIRXPTRA2EN	CHIRXPTRA2EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA2 0: Disable 1: Enable
17	RA1E	CHIRXPTRA1EN	CHIRXPTRA1EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA1 0: Disable 1: Enable
16	RA0E	CHIRXPTRA0EN	CHIRXPTRA0EN bit (Initial value: 0, R/W) This bit is used to enable/disable the timeslot for the receive channel pointed to by the TDM switch pointer CHIRXPTRA0 0: Disable 1: Enable
15:0	—	Reserved	—

Figure 16.4.2 CHI Pointer Enable Register (PNTREN) (2/2)

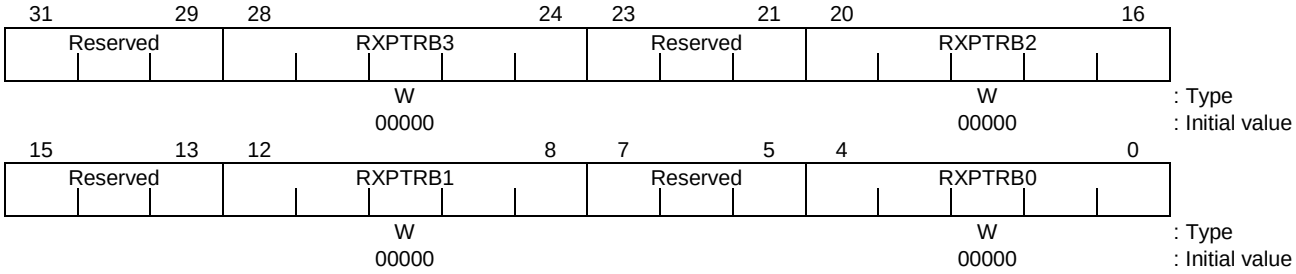
16.4.3 CHI Receive Pointer A Register (RXPTRA) 0xA808



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:24	RXPTRA3[4:0]	CHIRXPTRA3	CHIRXPTRA3 bits (Initial value: 00000, R/W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 3 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
23:21	—	Reserved	—
20:16	RXPTRA2[4:0]	CHIRXPTRA2	CHIRXPTRA2 bits (Initial value: 00000, R/W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 2 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
15:13	—	Reserved	—
12:8	RXPTRA1[4:0]	CHIRXPTRA1	CHIRXPTRA1 bits (Initial value: 00000, R/W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 1 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
7:5	—	Reserved	—
4:0	RXPTRA0[4:0]	CHIRXPTRA0	CHIRXPTRA0 bits (Initial value: 00000, R/W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 0 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

Figure 16.4.3 CHI Receive Pointer A Register (RXPTRA)

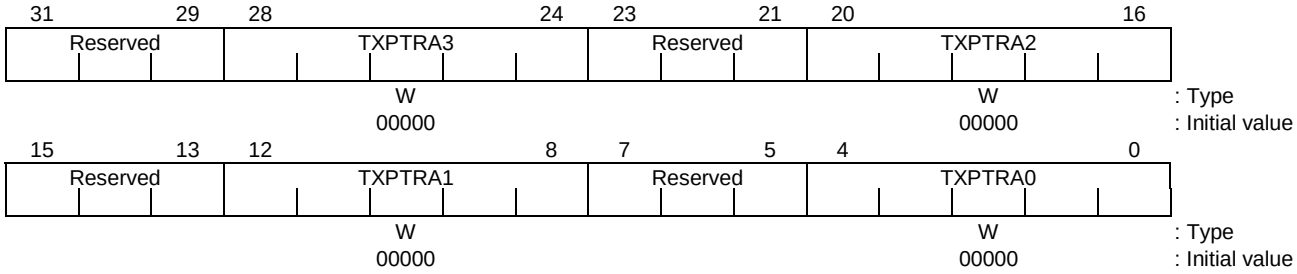
16.4.4 CHI Receive Pointer B Register (RXPTRB) 0xA80C



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:24	RXPTRB3[4:0]	CHIRXPTRB3	CHIRXPTRB3 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 3 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
23:21	—	Reserved	—
20:16	RXPTRB2[4:0]	CHIRXPTRB2	CHIRXPTRB2 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 2 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
15:13	—	Reserved	—
12:8	RXPTRB1[4:0]	CHIRXPTRB1	CHIRXPTRB1 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 1 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
7:5	—	Reserved	—
4:0	RXPTRB0[4:0]	CHIRXPTRB0	CHIRXPTRB0 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 0 of the CHI receive holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

Figure 16.4.4 CHI Receive Pointer B Register (RXPTRB)

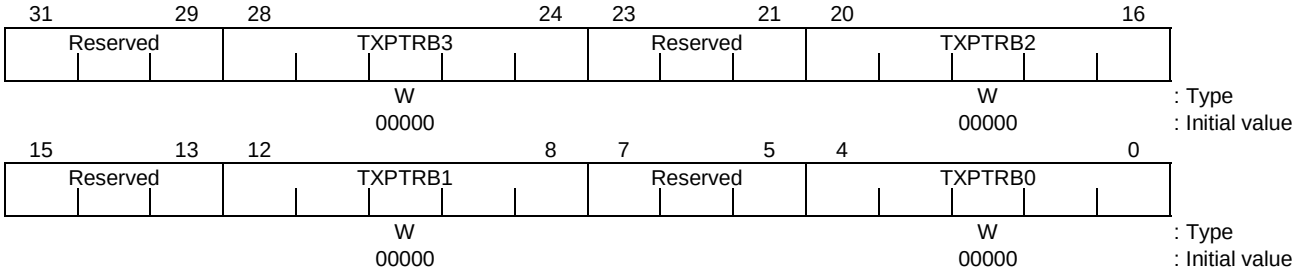
16.4.5 CHI Transmit Pointer A Register (TXPTRA) 0xA810



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:24	TXPTRA3[4:0]	CHITXPTRA3	CHITXPTRA3 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 3 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
23:21	—	Reserved	—
20:16	TXPTRA2[4:0]	CHITXPTRA2	CHITXPTRA2 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 2 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
15:13	—	Reserved	—
12:8	TXPTRA1[4:0]	CHITXPTRA1	CHITXPTRA1 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 1 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.
7:5	—	Reserved	—
4:0	TXPTRA0[4:0]	CHITXPTRA0	CHITXPTRA0 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 0 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel CHINCHAN.

Figure 16.4.5 CHI Transmit Pointer A Register (TXPTRA)

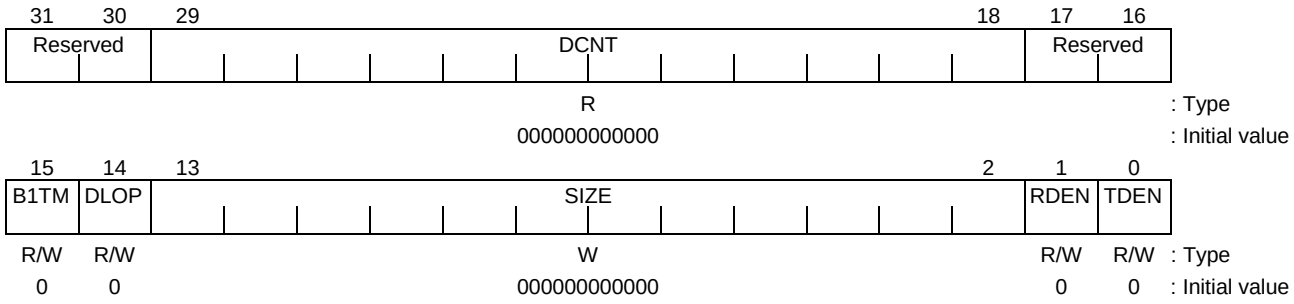
16.4.6 CHI Transmit Pointer B Register (TXPTRB) 0xA814



Bits	Mnemonic	Field Name	Description
31:29	—	Reserved	—
28:24	TXPTRB3[4:0]	CHITXPTRB3	CHITXPTRB3 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 3 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
23:21	—	Reserved	—
20:16	TXPTRB2[4:0]	CHITXPTRB2	CHITXPTRB2 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 2 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
15:13	—	Reserved	—
12:8	TXPTRB1[4:0]	CHITXPTRB1	CHITXPTRB1 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 1 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).
7:5	—	Reserved	—
4:0	TXPTRB0[4:0]	CHITXPTRB0	CHITXPTRB0 bits (Initial value: 00000, W) These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 0 of the CHI transmit holding register B; register B handles all timeslots from channel (CHINCHAN + 1) to channel (CHINCHAN × 2 + 1). The value loaded for this TDM switch pointer is the desired timeslot number minus (CHINCHAN + 1).

Figure 16.4.6 CHI Transmit Pointer B Register (TXPTRB)

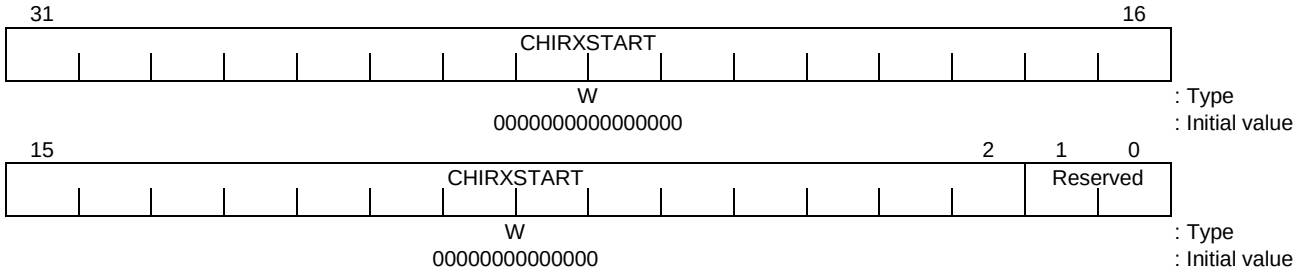
16.4.7 CHI SIZE Register (CHISIZE) 0xA818



Bits	Mnemonic	Field Name	Description
31:30	—	Reserved	—
29:18	DCNT[13:2]	CHIDMACNT	CHIDMACNT bits (Initial value: 0000_0000_0000, R) These bits provide the status of the CHI DMA counter.
17:16	—	Reserved	—
15	B1TM	CHIBUF1TIME	CHIBUF1TIME bit (Initial value: 0, R/W) The CHI DMA controller supports two buffer addressing modes depending on the state of this bit. 0: the CHI DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner. 1: the CHI DMA controller will stop executing when it reaches the end of the DMA buffer.
14	DLOP	CHIDMALOOP	CHIDMALOOP bit (Initial value: 0, R/W) This bit selects loopback modes depending on the state of this bit. 0: not loopback mode. 1: loopback mode(This ordering allows an RX-to-TX immediate loopback via the DMA buffer. Please refer to the chapter of 16.3.4
13:2	SIZE[13:2]	CHISIZE	CHISIZE bits (Initial value: 0000_0000_0000, W) These bits define the size of the CHI DMA buffers (16 bytes minimum/16 Kbytes maximum). Both the CHI RX buffer and the CHI TX buffer are the same size. The last address in the CHI RX DMA buffer is given by CHIRXSTART[31:2] · CHISIZE[13:2]. The last address in the CHI TX DMA buffer is given by CHITXSTART[31:2] · CHISIZE[13:2]. The value loaded into CHISIZE should be equal to the desired buffer length-1.
1	RDEN	CHIRXDMAEN	CHIRXDMAEN bit (Initial value: 0, R/W) This bit enables the CHI DMA receive function. This bit should not be set until the CHIRXSTART, CHITXSTART, and CHISIZE registers are setup. To enable CHI DMA receive function, this bit must be set before the CHI Module is enabled (ENCHI asserted). Either CHIRXDMAEN or CHITXDMAEN or both can be set at a time since the CHI DMA controller can support full duplex operation. 0: Disable 1: Enable
0	TDEN	CHITXDMAEN	CHITXDMAEN bit (Initial value: 0, R/W) This bit enables the CHI DMA transmit function. This bit should not be set until the CHIRXSTART, CHITXSTART, and CHISIZE registers are setup. To enable CHI DMA transmit function, this bit must be set before the CHI Module is enabled (ENCHI asserted). Either CHIRXDMAEN or CHITXDMAEN or both can be set at a time since the CHI DMA controller can support full duplex operation. 0: Disable 1: Enable

Figure 16.4.7 CHI SIZE Register (CHISIZE)

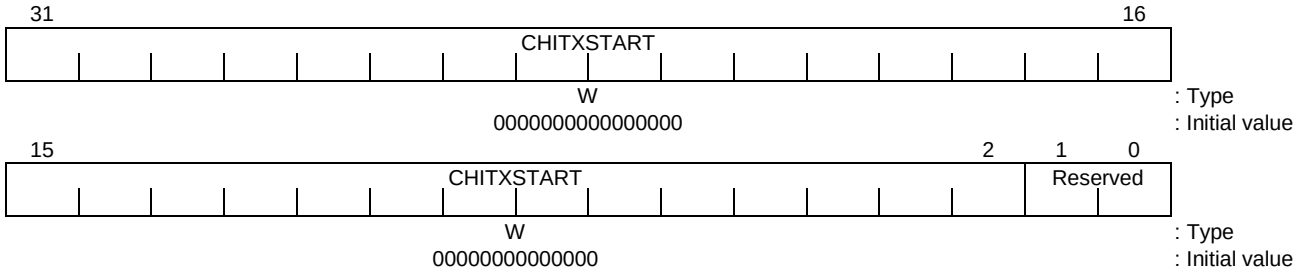
16.4.8 CHI RX Start Register (RXSTRT) 0xA81C



Bits	Mnemonic	Field Name	Description
31:2	RxStrt[31:2]	CHIRXSTART	CHIRXSTART bits (Initial value: 30'b0, W) These bits define the start address for the CHI RX DMA buffer. The CHI RX buffer and CHI TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).
1:0	—	Reserved	—

Figure 16.4.8 CHI RX Start Register (RXSTRT)

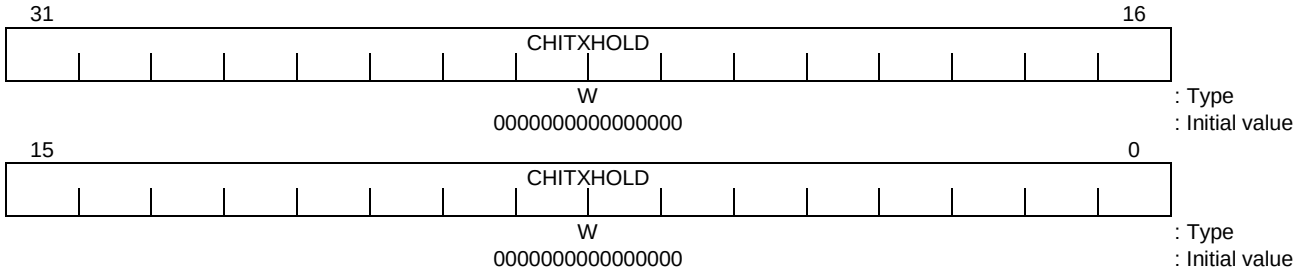
16.4.9 CHI TX Start Register (TXSTRT) 0xA820



Bits	Mnemonic	Field Name	Description
31:2	TxStrt[31:2]	CHITXSTART	CHITXSTART bits (Initial value: 30'b0, W) These bits define the start address for the CHI TX DMA buffer. The CHI RX buffer and CHI TX buffer can be configured to either reside in different memory spaces or share the same memory space (overlapping buffers for loopback purposes or for optimum memory allocation).
1:0	—	Reserved	—

Figure 16.4.9 CHI TX Start Register (TXSTRT)

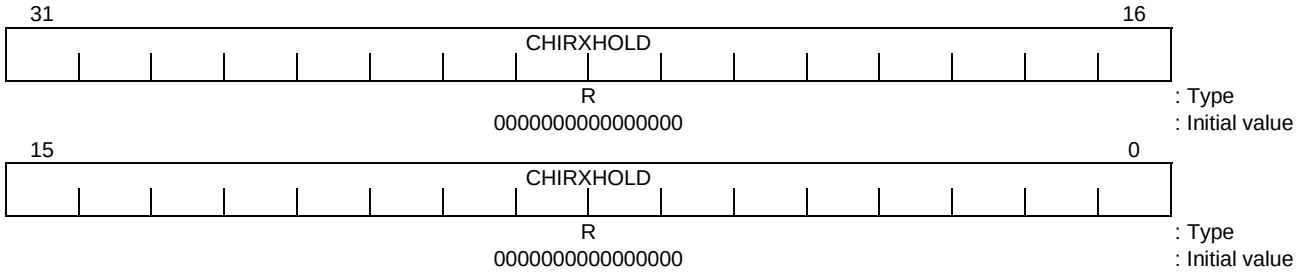
16.4.10 CHI TX Holding Register (CHIHOLD) 0xA824



Bits	Mnemonic	Field Name	Description
31:0	CHIHOLD[31:0]	CHITXHOLD	CHITXHOLD bits (Initial value: 32'b0, W) These bits represent the CHI data to be transmitted. CHI data can be either written directly to this register by the CPU or transparently read from the CHI TX DMA buffer to this register. This register should only be loaded by the CPU after the CHIININTA or CHIININTB interrupt is asserted. The write immediately after CHIININTA updates the internal TX holding register A and the write immediately after CHIININTB updates the internal TX holding register B. Transmit data for bytes 3, 2, 1, and 0 are loaded into the 32-bit CHITXHOLD at locations [31:24], [23:16], [15:8], and [7:0], respectively. These data bytes correspond to the CHI timeslots as defined by the values in the TXPTRA and TXPTRB TDM switch registers.

Figure 16.4.10 CHI TX Holding Register (CHIHOLD)

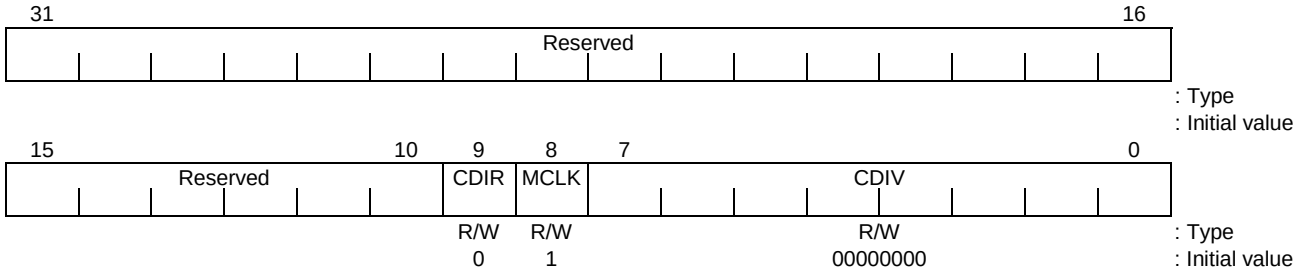
16.4.11 CHI RX Holding Register (CHIHOLD) 0xA824



Bits	Mnemonic	Field Name	Description
31:0	CHIHOLD[31:0]	CHIRXHOLD	CHIRXHOLD bits (Initial value: 32'b0, R) These bits represent the CHI data to be received. CHI data can be either read directly from this register by the CPU or transparently written to the CHI RX DMA buffer from this register. This register should only be read by the CPU after the CHIININTA or CHIININTB interrupt is asserted. The read immediately after CHIININTA sees the internal RX holding register A and the read immediately after CHIININTB sees the internal RX holding register B. Receive data for bytes 3, 2, 1, and 0 are stored into the 32-bit CHIRXHOLD at locations [31:24], [23:16], [15:8], and [7:0], respectively. These data bytes correspond to the CHI timeslots as defined by the values in the RXPTRA and RXPTRB TDM switch registers.

Figure 16.4.11 CHI RX Holding Register (CHIHOLD)

16.4.12 CHI Clock Register (CHICLOCK) 0xA828



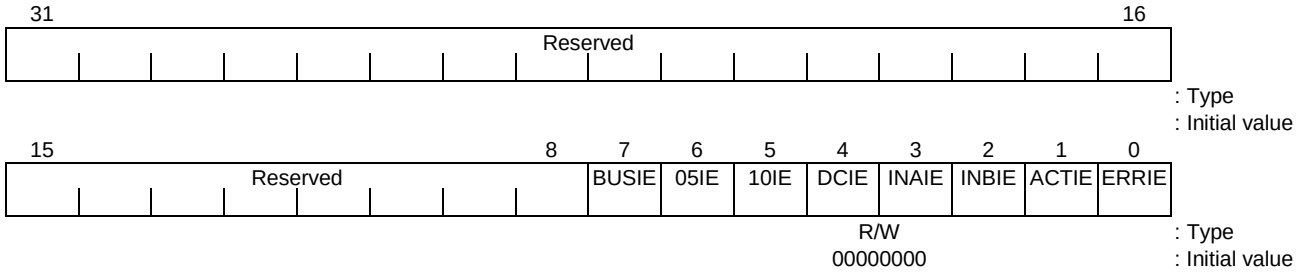
Bits	Mnemonic	Field Name	Description
31:10	—	Reserved	—
9	CDIR	CHICLKDIR	CHICLKDIR bit (Initial value: 0, R/W) This bit controls the direction of the CHICLK pin. 0: input (CHI slave mode) 1: output (CHI master mode) Note: Please set the same value CHICLKDIR and CHIFSDIR. Each set the difference value (CHIFSDIR = 1, CHICLKDIR = 0 or CHIFSDIR = 0, CHICLKDIR = 1) can't recommend
8	MCLK	CHIMCLKEN	CHIMCLKEN bit (Initial value: 1, R/W) This bit is used to enable or disable the CHICLK counter and CHICLK clock generation. This bit controls the direction of the CHICLK pin. 0: disable (halting the clock to the CHI Module in order to reduce power consumption) 1: enable
7:0	CDIV[7:0]	CHICLKDIV	CHICLKDIV bit (Initial value: 0000_0000, R/W) These bits define the divide-modulus for the programmable divider used to generate CHICLK. The divide-modulus is equal to (CHICLKDIV + 2). (See the following Table 16.4.2.) Note: CLKDIV less than 1 can't recommend

Figure 16.4.12 CHI Clock Register (CHICLOCK)

Table 16.4.2 CHI Clock Divide

CHICLKDIV[7:0]	Divide-modulus
2	4
:	:
N	N+2
:	:
253	255
254	256

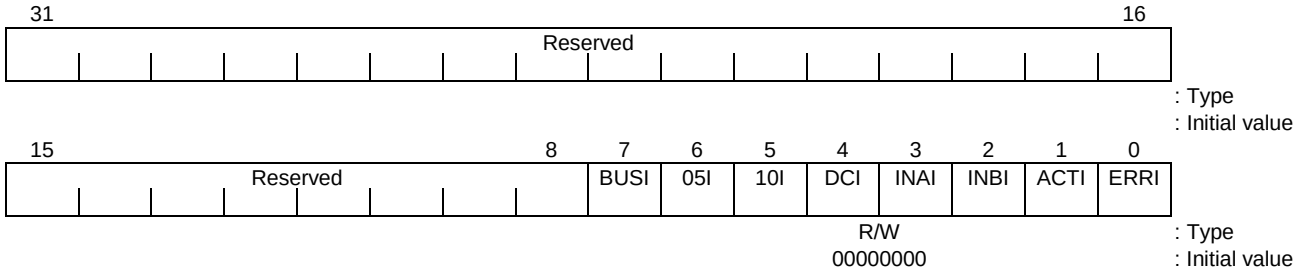
16.4.13 HI Interrupt Enable Register (CHIINTE) 0xA82C



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	BUSIE	CHIBUSERROR Interrupt Enable	CHIBUSERROR Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIBUSERROR Interrupt. 0: Disable 1: Enable
6	05IE	CHI0_5 Interrupt Enable	CHI0_5 Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHI0_5 Interrupt. 0: Disable 1: Enable
5	10IE	CHI1_0 Interrupt Enable	CHI1_0 Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHI1_0 Interrupt. 0: Disable 1: Enable
4	DCIE	CHIDMACNT Interrupt Enable	CHIDMACNT Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIDMACNT Interrupt. 0: Disable 1: Enable
3	INAIE	CHIININTA Interrupt Enable	CHIININTA Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIININTA Interrupt. 0: Disable 1: Enable
2	INBIE	CHIININTB Interrupt Enable	CHIININTB Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIININTB Interrupt. 0: Disable 1: Enable
1	ACTIE	CHIACTINT Interrupt Enable	CHIACTINT Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIACTINT Interrupt. 0: Disable 1: Enable
0	ERRIE	CHIERRINT Interrupt Enable	CHIERRINT Interrupt Enable bits (Initial value: 0, R/W) This bit is used to enable or disable the CHIERRINT Interrupt. 0: Disable 1: Enable

Figure 16.4.13 CHI Interrupt Enable Register (CHIINTE)

16.4.14 CHI Interrupt Status Register (CHIINT) 0xA830



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	BUSI	CHIBUSERROR Interrupt Status	CHIBUSERROR Interrupt status bit (Initial value: 0, R/W) This bit shows the CHIBUSERROR Interrupt Status. This bit is cleared by written “0”. 0: No interrupt 1: Interrupt occurs
6	05I	CHI0_5 Interrupt Status	CHI0_5 Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHI0_5 Interrupt Status. This bit is cleared by written “0”. 0: No interrupt 1: Interrupt occurs
5	10I	CHI1_0 Interrupt Status	CHI1_0 Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHI1_0 Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs
4	DCI	CHIDMACNT Interrupt Status	CHIDMACNT Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHIDMACNT Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs
3	INAI	CHIININTA Interrupt Status	CHIININTA Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHIININTA Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs
2	INBI	CHIININTB Interrupt Status	CHIININTB Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHIININTB Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs
1	ACTI	CHIACTINT Interrupt Status	CHIACTINT Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHIACTINT Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs
0	ERRI	CHIERRINT Interrupt Status	CHIERRINT Interrupt Status bit (Initial value: 0, R/W) This bit shows the CHIERRINT Interrupt Status. This bit is cleared by written “0” 0: No interrupt 1: Interrupt occurs

Figure 16.4.14 CHI Interrupt Status Register (CHIINT)

17. Serial Peripheral Interface

17.1 Characteristics

The SPI is a serial interface consisting of clock, data out, and data in. The SPI is used to interface to devices such as serial power supplies, serial A/D converters, and other devices that contain simple serial clock and data interface. The TX4925 only supports the master mode and generates the SPI clock to the slaves. Multiple slave devices can share the SPI by using a unique chip select for each slave device. The chip select can be generated using one of the general purpose I/O ports on TX4925, or using some other output port available in the system. When a device is selected by asserting it's chip select, the device will shift data in using the SPICLK and SPIOUT signals and the device will shift data out using the SPIIN signal. When a device is not selected then the data output connected to SPIIN must be tri-stated so other devices can share the SPIIN signal. The SPI module contains registers which provide programmability for SPICLK rate, MSB first versus LSB first, clock polarity, data phase polarity, and byte mode versus word mode operation.

The SPI has the following characteristics.

- Phase and Polarity Selection
- Transfer size of 8 or 16 bits
- 4 frame transmit, 4 frame receive buffers
- Master operation
- Inter Frame Space Delay Feature
- MSB/LSB first transfer

17.2 Block Diagram

The SPI Module primarily consists of a 16-bit SPI Data Register (SPDR), a 16-bit Shift Register, a 16-bit Transmitter Buffer, a 16-bit Receiver Buffer, a Baud Rate Generator, an Inter Frame Time Counter, and Interrupt Logic. A block diagram of the SPI Module is shown in Figure 17.2.1.

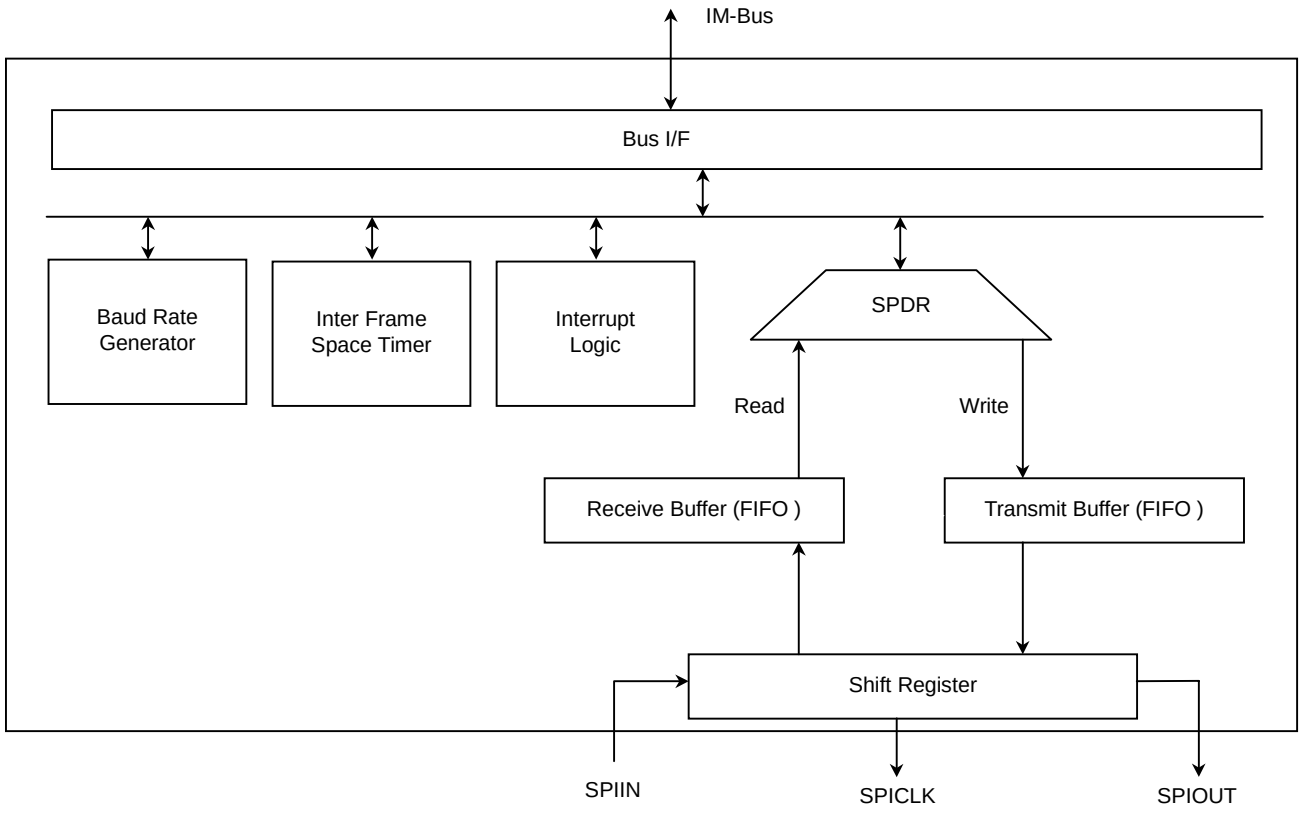


Figure 17.2.1 SPI Block Diagram

17.3 Detailed Explanation

17.3.1 Operation mode

There are 2 operation modes possible:

- Configuration mode (OPMODE = '01') :

  Only in this mode it is possible to change the setting of the low byte (bit 7 to 0) in the SPI Control Register 0 (SPCR0) and all bits in the SPI Control Register 1 (SPCR1). In this mode the SPSTP bit and also the receive and transmit FIFO will be cleared. The SPI module will be kept in reset. Running transfer are immediately aborted, even within the current frame.

- Active mode (OPMODE = '10') :

  This is the normal operation mode. A transfer occurs in this mode.

17.3.2 Transmitter/Receiver

The SPI module is kept in a reset state in the Configuration mode. Before setting the Active mode in OPMODE bits, the low byte (bit 7 to 0) in the SPI Control Register 0 (SPCR0) and all bits in the SPI Control Register 1 (SPCR1) should be set to the desired value. Once the OPMODE bits is set to the Active mode, the SPI module can run a transfer. The SPI logic will then wait until the software writes to the SPI Data Register (SPDR).

Once the software writes to the SPI Data Register (SPDR) then the contents will be transferred to the Shift Register and shifted out to the slave device. While data is shifting out to the slave device using the SPIOUT signal, data will shift in using the SPIIN signal. Once the data has finished shifting, the contents of the Shift Register will be loaded into the Receive Buffer and the SRRDY bit in SPI Status Register (SPSR) will be asserted to indicate that there is valid receive data in the Receive Buffer. The RBSI bit in the SPI Status Register (SPSR) is set and the interrupt occurs when the Receive Buffer level selected RXIFL bits in the SPI Control Register 0 (SPCR0) is filled.

Once the contents of the Transmit Buffer are transferred to the Shift Register, the STRDY bit in SPI Status Register (SPSR) will be asserted to indicate that the Transmit Buffer is once again ready to receive new data. The TBSI bit in the SPI Status Register (SPSR) is set and the interrupt occurs when the Transmit Buffer level selected TXIFL bits in the SPI Control Register 0 (SPCR0) is filled. Therefore the software is supposed to do the following procedure every time it attempts to write data into the Transmit Buffer.

(1) Check if STRDY or TBSI is a logic "1". If not, do nothing.
(2) Write data into the SPI Data Register (SPDR).

Thus, as long as the software can keep the Transmit Buffer serviced before the data shifts out of the Shift Register, the SPI can maintain seamless data transfer. If the software fails to keep up with the transfer rate, then the SPI will simply wait until the next data is written to the SPI Data Register (SPDR).

At the end of every series of transmission the software is supposed to negate the chip select signal for the target device by the following procedure.

(1) Check if SRRDY or RBSI is a logic “1”. If not, do nothing.
(2) Check if SIDLE is a logic “1”. If not, do nothing.
(3) Negate the chip select signal.

The SPI supports either 8-bit per character or 16-bit per character operation, as defined by the SSZ bits in the SPI Control Register 1 (SPCR1). The software can also select whether the MSB or LSB should shift first using the SBOS bit in the SPI Control Register 0 (SPCR0). Another set of SPHA and SPOL bits bit in the SPI Control Register 0 (SPCR0) select the transfer format. Please see to “17.3.4 Transfer Format”.

17.3.3 Baud Rate Generator

The rate of the SPICLK signal is determined by the value of the SER[7:0] bits in the SPI Control Register 1 (SPCR1). The SER[7:0] bits are used by the Baud Rate Generator to devide the SPI master clock generated by the Clock Generator (CG). The frequency of the SPI master clock is 40 MHz when MASTERCLK input is 80 MHz. The SPICLK rate is shown in the table below in this time.

Table 17.3.1 SPICLK Rate when MASTERCLK is 80 MHz

SER[7:0]	SPI Clock Rate
00000001b	10 MHz
00000010b	6.667 MHz
00000011b	5 MHz
00000100b	4 MHz
00000101b	3.33 MHz
…	
00001001b	2 MHz
…	
00010011b	1 MHz
…	
11111111b	78.125 KHz

17.3.4 Transfer Format

During a SPI transfer, data is simultaneously transmitted (shifted out serially) and received serially (shifted in serially). The serial clock synchronizes shifting and sampling of the information on the two serial data lines.

The transfer format depends on the setting of the SPHA and SPOL bits in the SPI Control Register 0 (SPCR0). SPHA switches between two fundamentally different protocols, which are described below.

17.3.4.1 SPHA Equals 0 Format

Figure 17.3.1 shows the transfer format for a SPHA=0 transfer.

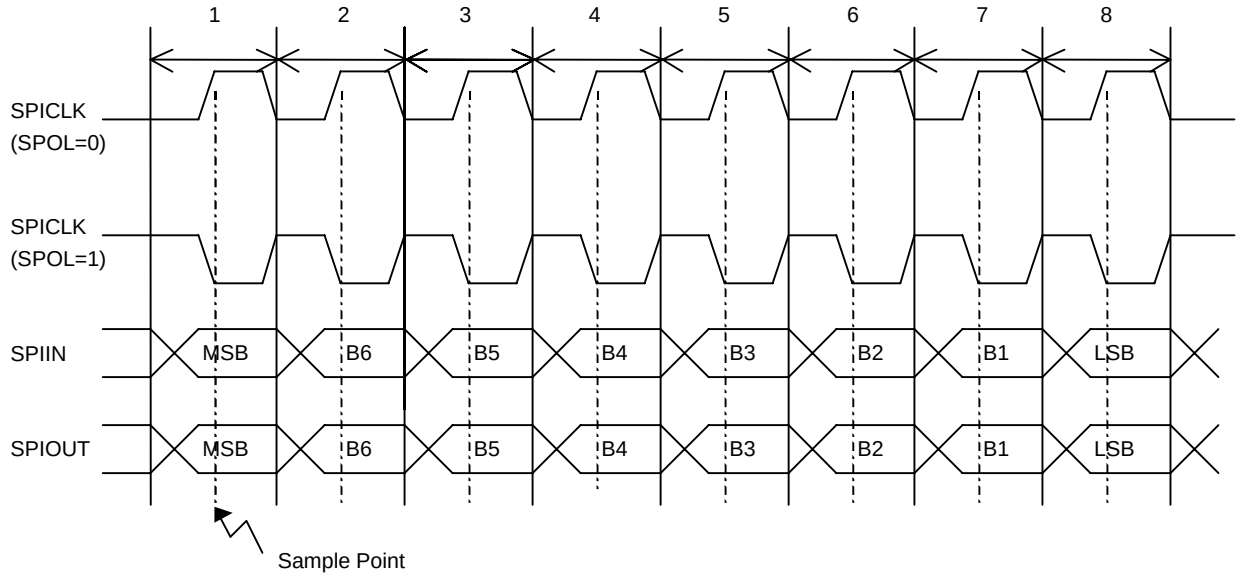


Figure 17.3.1 Transfer format when SPHA is “0”.

In this transfer format, the bit value is captured on the first clock edge. This will be on a rising edge when SPOL bit equals zero and on a falling edge when SPOL equals one. The value on the SPIIN and SPIOUT signals changes with the second clock edge on SPICLK. This clock edge will be a falling edge when SPOL equals zero and a rising edge, when SPOL equals one. With SPOL equal to zero, the shift clock will be idle low. With SPOL equals 1 it will idle high.

17.3.4.2 SPHA Equals 1 Format

Figure 17.3.2 shows the transfer format for a SPHA=1 transfer.

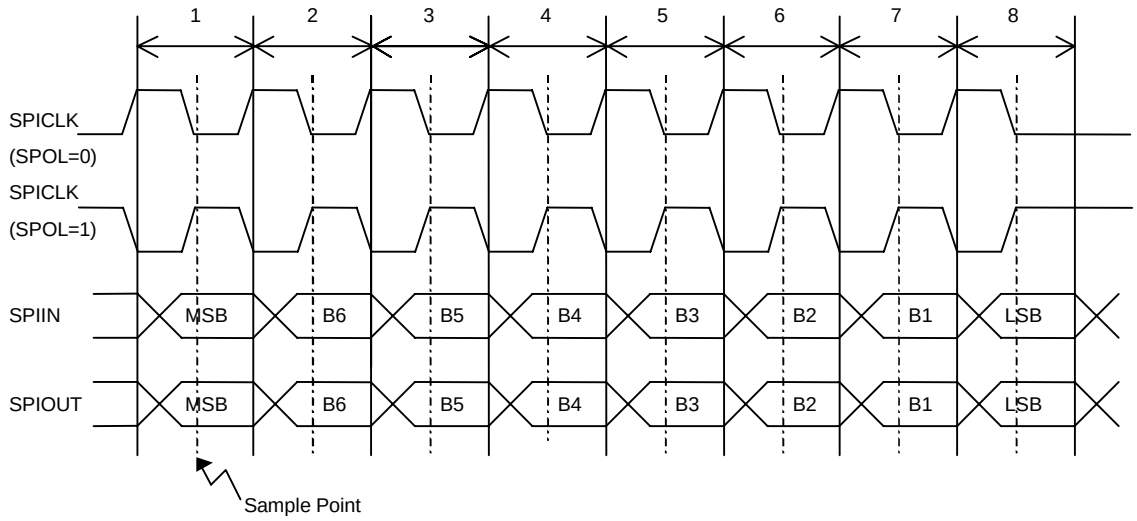


Figure 17.3.2 Transfer format when SPHA is “1”.

In this transfer format, the value on the SPIIN and SPIOUT signals changes with the second clock edge on SPICLK. This clock edge will be a rising edge when SPOL equals zero and a falling edge, when SPOL equals one. The bit value is shifted in on the second clock edge. This will be on a falling edge when SPOL bit equals zero and on a rising edge when SPOL equals one. With SPOL equal to zero, the shift clock will be idle low. With SPOL equals 1 it will idle high.

17.3.5 Inter Frame Space Counter

Sometimes it is desirable to guarantee a minimum time between groups of data. The Inter Frame Space Counter is used to provide delay between groups of data. If 16-bit data size is selected in the SPI Control Register 1 (SPCR1), delay will be inserted after 16 bits of data are shifted. If 8-bit data size is selected, delay will be inserted after 8 bits of data are shifted, as shown in Figure 17.4.3. Inter Frame delay is added by setting the IFS[7:0] bits to a value other than 0. The number stored in these bits will directly correspond to the number of the four times of SPI Master clock of delay that will be inserted between frames. A zero value for these bits will imply seamless operation and the SPI will shift data and provide clocks continuously as long as the software keeps up with the transmitter rate.

17.3.6 SPI Buffer Structure

The SPI has both transmit and receive buffer. The buffers are implemented as FIFO and are able to store four frames each.

When a new SPI transfer is started by writing the data register, the transfer value is first stored in SPI's transmit buffer. From there the value will be fetched by the shift register immediately, if the module is idle or after the currently running transfer has completed.

A receive value from the shift register is stored in the receive buffer every time a transfer completes.

The SPI is able to generate interrupts depending on the fill-level of these buffers. Therefore, it is possible to refill the buffers with several values within one interrupt service routine, if desired.

17.3.7 SPI System Errors

SPI is able to detect the following system error during the transfer.

17.3.7.1 Overrun Error (SPOE)

An Overrun Error will be generated, when the transmit buffer is completely filled, while a new value has been written on the SPI Transmit FIFO. In this case the already written data in the transmit buffer is not changed and the new value is abandoned. Then the SPOE flag in the SPSR register is set.

17.3.8 Interrupts

The SPI has three types interrupt sources. OR signal of them connects to the internal Interrupt Controller (IRC). Please check SPI Status Register (SPSR) to know which type interrupt occurred.

Type	Status Bits	Mask-able Bit
System error or idle	SPOE, SIDLE	SPOIE, SILIE
Receive Buffer Fill	RBSI	RBFIE
Transmit Buffer Fill	TBSI	TBFIE

The System error or idle interrupt is used for error detection purpose and idle state interrupt. The receive buffer fill interrupt and the transmit buffer fill interrupt are used to fetch and setup new data in an interrupt service routine.

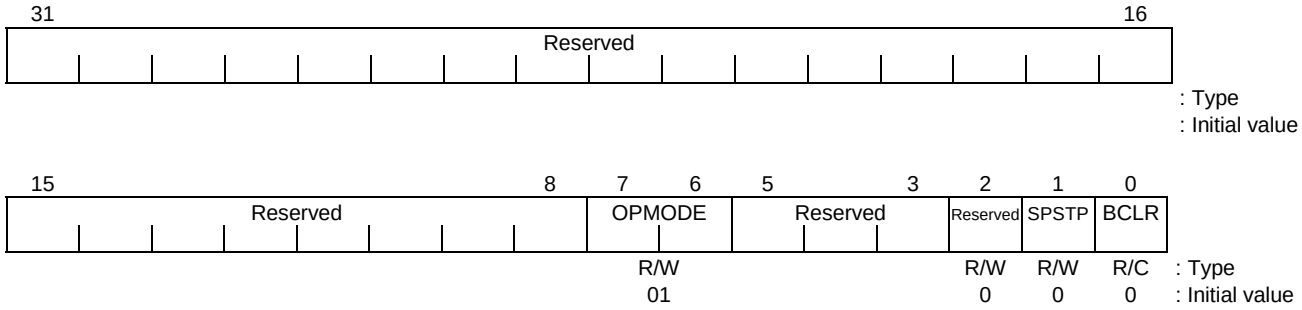
17.4 Registers

All registers in the SPI Module should be accessed only as full word (32-bit) accesses. Any other type of access produces an undefined result. Please write “0” to the undefined bit.

Table 17.4.1 SPI Module Registers

Reference	Address	Bit Width	Register	Register Name
17.4.1	0xF800	32	SPMCR	SPI Master Control Register
17.4.2	0xF804	32	SPCR0	SPI Control Register 0
17.4.3	0xF808	32	SPCR1	SPI Control Register 1
17.4.4	0xF80C	32	SPFS	SPI Inter Frame Space Register
17.4.5	0xF810	32	—	(Reserved)
17.4.5	0xF814	32	SPSR	SPI Status Register
17.4.6	0xF818	32	SPDR	SPI Data Register
	0xF81C	32	—	(Reserved)

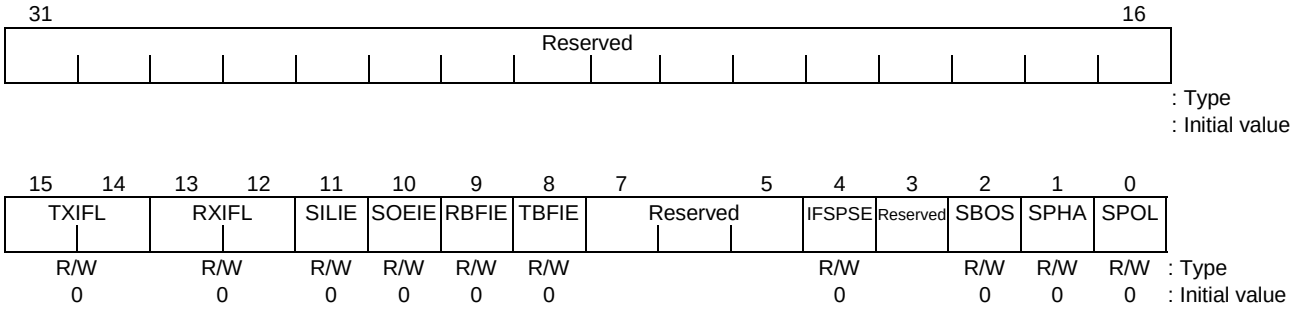
17.4.1 SPI Master Control Register (SPMCR) 0xF800



Bits	Mnemonic	Field Name	Explanation
31 : 8	—	Reserved	—
7:6	OPMODE	Operation Mode	Operation Mode (Initial value: 01, R/W) Set operation mode 00: Don't care. Writing this value to the OPMODE bits doesn't change any thing. 01: Configuration mode 10: Active mode (normal operation mode) 11: Reserved
5:3	—	Reserved	—
2	—	—	This bit is reserved. Don't write "1" to this bit (Initial value: 0, R/W).
1	SPSTP	SPI Stop	SPI Stop (Initial value: 0, R/W) If this flag is asserted, the module will stop the transferring after the current frame has been completed. This bit could be set only when the SPI is in active mode. Setting the SPI in configuration mode will clear this bit. 0: Normal operation 1: Stop after completion of the current transfer
0	BCLR	SPI Buffer Clear	SPI Buffer Clear (Initial value: 0, R/C) This flag is used to clear the receive and transmit FIFO. The FIFO logic can be reset by writing a "1" value to this bit. Please wait until the SPI module is idle (SIDLE = 1) before activating the BCLR bit. This register will always be read as "0". Write: 0: Don't care 1: FIFO clear

Figure 17.4.1 SPI Master Control Register

17.4.2 SPI Control Register 0 (SPCR0) 0xF804



Bits	Mnemonic	Field Name	Explanation
31:16	—	Reserved	—
15:14	TXIFL	Transmit Interrupt Fill Level	Transmit Interrupt Fill Level (Initial value: 00, R/W) Select the interrupt fill level of the transmit FIFO. 00: Interrupt, if one or more Tx values can be stored 01: Interrupt, if two or more Tx values can be stored 10: Interrupt, if three or more Tx values can be stored 11: Interrupt, if four Tx values can be stored
13:12	RXIFL	Receive Interrupt Fill Level	Receive Interrupt Fill Level (Initial value: 00, R/W) Select the interrupt fill level of the receive FIFO. 00: Interrupt, if one or more Rx values are stored 01: Interrupt, if two or more Rx values are stored 10: Interrupt, if three or more Rx values are stored 11: Interrupt, if four Rx values are stored
11	SILIE	SPI IDLE Interrupt Enable	SPI IDLE Interrupt Enable (Initial value: 0, R/W) Enable the SPI IDLE Interrupt to the interrupt controller of TX4925. 0: Disable 1: Enable
10	SOEIE	SPI Overrun Interrupt Enable	SPI IDLE Overrun Enable (Initial value: 0, R/W) Enable the SPI Overrun Interrupt to the interrupt controller of TX4925. 0: Disable 1: Enable
9	RBSIE	Receive Buffer Fill Interrupt Enable	Receive Buffer Fill Interrupt Enable (Initial value: 0, R/W) Enable the Receive Buffer Fill Interrupt to the interrupt controller of TX4925. 0: Disable 1: Enable
8	TBSIE	Transmit Buffer Fill Interrupt Enable	Transmit Buffer Fill Interrupt Enable (Initial value: 0, R/W) Enable the Transmit Buffer Fill Interrupt to the interrupt controller of TX4925. 0: Disable 1: Enable
7:5	—	Reserved	—

Note 1: Bit 5 to Bit 0 could only be written, when the SPI module is in configuration mode.

Note 2: SPOL and SPHA bits determine the idle phase of SPICLK and the valid clock edge for sampling data. Refer to “17.3.4 Transfer Format”.

Figure 17.4.2 SPI Control Register 0 (SPCR0) (1/2)

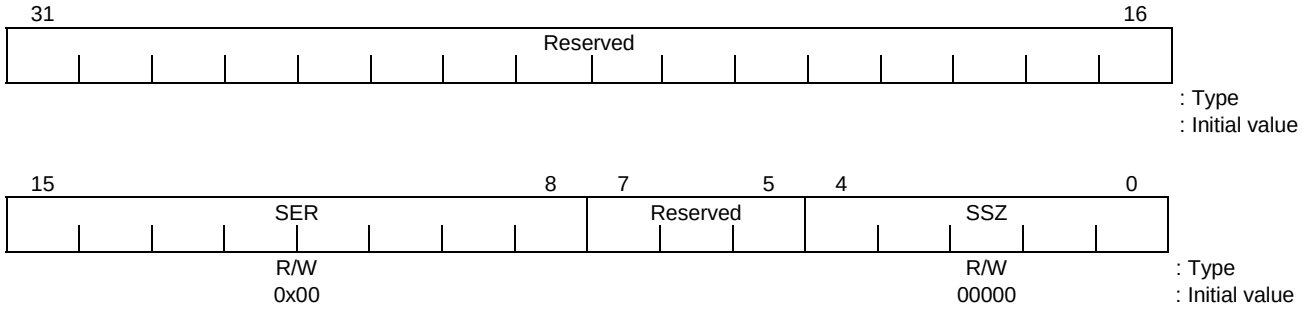
Bits	Mnemonic	Field Name	Explanation
4	IFSPSE	Inter Frame Space prescaler enable	Inter Frame Space prescaler Enable (Initial value: 0, R/W) Enable the Inter Frame Space prescaler. 0: Disable 1: Enable
3	—	Reserved	—
2	SBOS	SPI Bit Order Select	SPI Bit Order Select (Initial value: 0, R/W) Select bit order of transfer data. 0: LSB first operation, the least significant bit is shifted first. 1: MSB first operation, the most significant bit is shifted first.
1	SPOL	SPI Polarity	SPI clock Polarity (Initial value: 0, R/W) Select the SPICLK polarity. 0: Active High Clocks selected; SPICLK idles low 1: Active Low Clocks selected; SPICLK idles high
0	SPHA	SPI Phase	SPI clock Phase (Initial value: 0, R/W) Selects one of two fundamentally different transfer format. 0: Sampling on the first edge, Shift on the second edge. 1: Shift on the first edge, Sampling on the second edge.

Note 1: Bit 5 to Bit 0 could only be written, when the SPI module is in configuration mode.

Note 2: SPOL and SPHA bits determine the idle phase of SPICLK and the valid clock edge for sampling data. Refer to “17.3.4 Transfer Format”.

Figure 17.4.2 SPI Control Register 0 (SPCR0) (2/2)

17.4.3 SPI Control Register 1 (SPCR1) 0xF808



Bits	Mnemonic	Field Name	Explanation
31:16	—	Reserved	—
15:8	SER	SPI Data Rate	SPI Data Rate (Initial value: 0x00, R/W) Control the bit-rate for the transmission. The clock-rate on the SPI bus can be calculated using the following formula: $f_{BR} = f_{SPI}/2\ (n + 1)$ But n = 0 is not available. Refer to Note 2.
7:5	—	Reserved	—
4:0	SSZ	SPI Transfer Size	SPI Transfer size (Initial value: 00000b, R/W) Select the number of bits to shift. 0x08: 8 bits 0x10: 16 bits others: Reserved. Don't set these values.

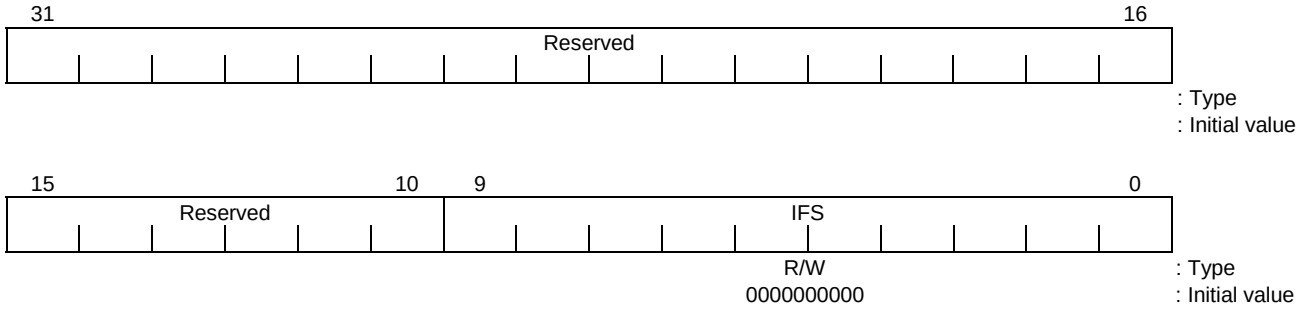
Note 1: This register could only be written, when the SPI module is in configuration mode.

Note 2: The SPICLK rate is shown in the table below in this time when MASTERCLK input is 40 MHz.

SER[7:0]	SPI Clock Rate
00000001b	10 MHz
00000010b	6.667 MHz
00000011b	5 MHz
00000100b	4 MHz
00000101b	3.33 MHz
…	
00001001b	2 MHz
…	
00010011b	1 MHz
…	
11111111b	78.125 KHz

Figure 17.4.3 SPI Control Register 1 (SPCR1)

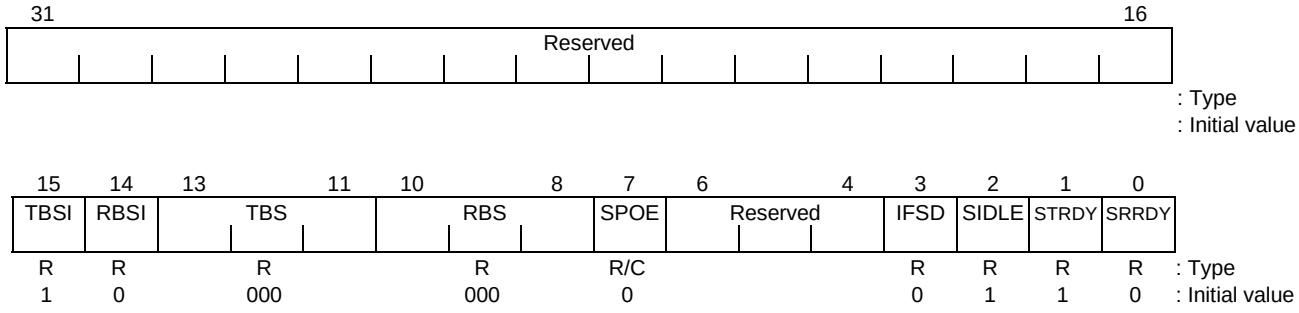
17.4.4 SPI Inter Frame Space Register (SPFS) 0xF80C



Bits	Mnemonic	Field Name	Explanation
31:10	—	Reserved	—
9:0	IFS	Inter Frame Space	Inter Frame Space (Initial value: 0000000000b, R/W) Configure the amount of time, which is inserted between two consecutive frames. When setting this register to 0, two consecutive transfers will be send using only minimum amount of time required to load the buffers between consecutive frames. This minimum amount of time is not zero. When the prescaler is not used (IFSPSE bit in SPCR0 register is “0”), the inter frame space can be calculated using the following formula: $f_{IFS} = n/f_{SEI}$ (range is 25 ns up to 25.6 μs when MASTERCLK frequency is 80 MHz.) When using the prescaler (IFSPSE bit in SPCR0 register is “1”), the inter frame space can be calculated using the following formula: $f_{IFS} = 32 \times n/f_{SEI}$ (range is 800 ns up to 819.2 μs when MASTERCLK frequency is 80 MHz.) Please write to this register when SIDLE bit is “0”.

Figure 17.4.4 SPI Inter Frame Space Register (SPFS)

17.4.5 SPI Status Register (SPSR) 0xF814



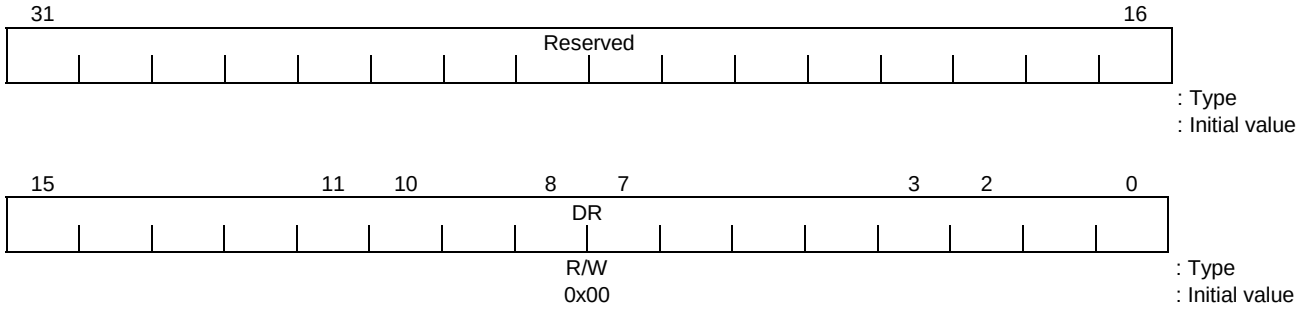
Bits	Mnemonic	Field Name	Explanation
31:16	—	Reserved	—
15	TBSI	Transmit Buffer Status Indicator	Transmit Buffer Status Indicator (Initial value: 1, R) This bit indicates a transmit fill level interrupt. 0: Interrupt have been generated 1: Interrupt have not been generated
14	RBSI	Receive Buffer Status Indicator	Receive Buffer Status Indicator (Initial value: 0, R) This bit indicates a receive fill level interrupt. 0: Interrupt have been generated 1: Interrupt have not been generated
13:11	TBS	Transmit Buffer Status	Transmit Buffer Status (Initial value: 000, R) The field shows the status of the transmit buffer. 000: Transmit Buffer Empty 001: 1 transfer stored 010: 2 transfers stored 011: 3 transfers stored 100: 4 transfers stored, Buffer full 101 – 111: Not Available
10:8	RBS	Receive Buffer Status	Receive Buffer Status (Initial value: 000, R) The field shows the status of the receive buffer. 000: Receive Buffer Empty 001: 1 transfer stored 010: 2 transfers stored 011: 3 transfers stored 100: 4 transfers stored, Buffer full 101 – 111: Not Available
7	SPOE	SPI Overrun Error	SPI Overrun Error (Initial value: 0, R/C) This flag indicates that a value in the transmit buffer has been overwritten, before it could be sent. It can be cleared by writing a “1” value to it. This flag will be cleared by setting the module in configuration mode. Read: 0: no error 1: Overrun error occurred Write: 0: Don’t care 1: Clear
6:4	—	Reserved	—

Figure 17.4.5 SPI Status Register (SPSR) (1/2)

Bits	Mnemonic	Field Name	Explanation
3	IFSD	SPI Inter Frame Space Delay Indicator	SPI Inter Frame Space Delay Indicator (Initial value: 0, R) This flag is asserted during the time, where one frame has been processed and the next frame is being delayed by the inter-frame-space timer. 0: no inter frame cycle 1: inter frame cycle
2	SIDLE	SPI Idle Indicator	SPI Idle Indicator (Initial value: 0, R) This flag is asserted, if no transfer is in progress and if the transmit buffer is empty or the stop mode (SESTP = 1) is activated. 0: run 1: Idle
1	STRDY	SPI Transmit Ready	SPI Transmit Ready (Initial value: 0, R) This flag indicates, that the transmit buffer is ready to receive new data. The flag is cleared, if the transmit buffer is full. 0: transmit buffer full 1: one or more space in transmit buffer
0	SRRDY	SPI Receive Ready	SPI Receive Ready (Initial value: 0, R) This flag indicates, that there is valid data stored in the receive buffer. This flag is cleared by reading the SPDR or SPRS register until the receive buffer is empty. 0: Receive buffer empty 1: one or more data in receive buffer

Figure 17.4.5 SPI Status Register (SPSR) (2/2)

17.4.6 SPI Data Register (SPDR) 0xF818



Bits	Mnemonic	Field Name	Explanation
31:16	—	Reserved	—
15:0	DR	SPI Data Register	SPI Data Register (Initial value: 0x00, R/W) A write to the SPDR register writes the value to the transmit buffer. From there, the data will be transferred to the shift register as soon as SPI module is ready for the next transfer. Reading the SPDR register delivers the current value from the receive FIFO and increments the receive FIFO pointer, if there are other values stored in the FIFO. For eight bit transfers, only the lower eight bits of the SPDR register are used. The upper byte, bit 15 to 8, are read “0x0” for eight bit transfers. For 16 bit transfers all 16 bits of SPDR are used.

Figure 17.4.6 SPI Data Register (SPDR)

18. NAND Flash Memory Controller

18.1 Characteristics

The TX4925 on-chip NAND Flash Memory Controller (NDFMC) generates the control signals required to interface with the NAND Flash Memory. It has also the ECC calculating circuits.

The NAND Flash Memory Controller has the following characteristics.

- Controlled NAND Flash memory interface by setting Registers.
- On-chip ECC calculating circuits

18.2 Block Diagram

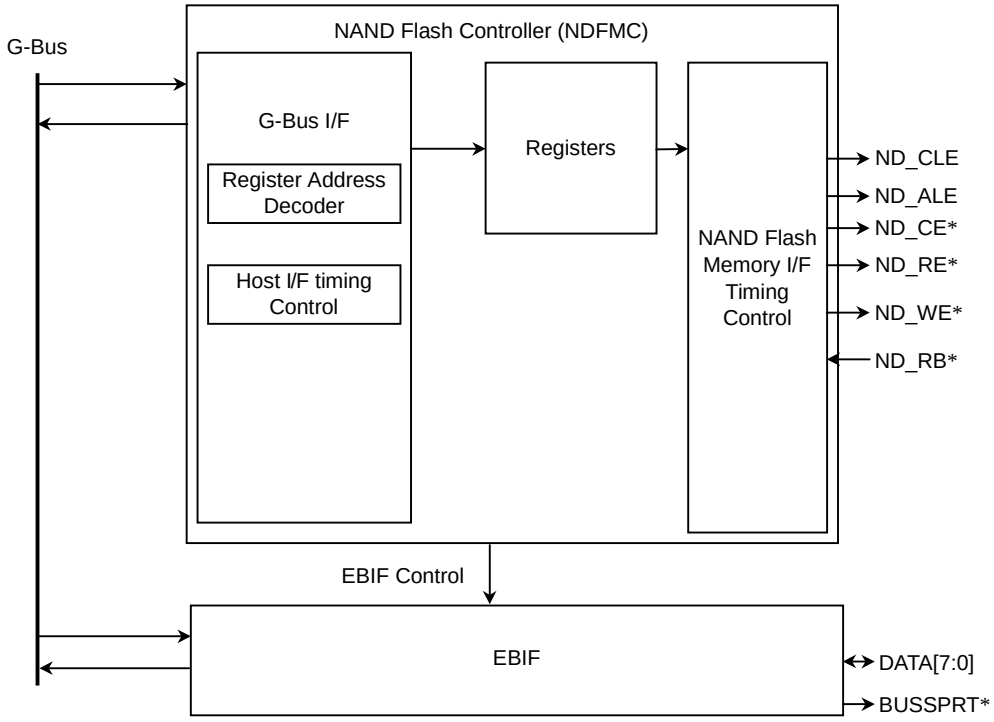


Figure 18.2.1 NAND Flash Memory Controller Block Diagram

18.3 Detailed Explanation

18.3.1 Access to NAND Flash Memory

The TX4925 NDFMC supports the interface between the NAND Flash Memory using register indirect sequence. It has the ECC calculating circuits. Please see 18.3.2 in detail of the ECC. This section describes the procedure to access to NAND Flash Memory.

Basically, set the command in NDFMCR at first and then read or write in NDFDTR. The read cycle for NDFDTR is finished after the external read cycle for the NAND Flash Memory is finished. Equally, the write cycle for NDFDTR is finished after the external write cycle for the NAND Flash Memory is finished.

18.3.1.1 Initialize

The initialize sequence is below.

(1) NDFSPR (0xC014): Set the Low pulse width.
(2) NDFIMR (0xC010): Set 0x81 if need enable interrupt.

18.3.1.2 Write

The write sequence is below.

(1) NDFMCR (0xC004): Set 0x70 to reset ECC data.
(2) NDFMC hasn't WP* signal. It must be high using other logic for example PIO.
(3) Write 512 bytes
- NDFMCR (0xC004): Set 0x91 to assert ND_CLE* signal and do the command mode.
- NDFDTR (0xC000): Set 0x80 to write the Serial Data Input command.
- NDFMCR (0xC004): Set 0x92 to assert ND_ALE* signal and do the address mode.
- NDFDTR (0xC000): Set A[7:0], A[16:9], and A[24:17]. If need, set A[25].
- NDFMCR (0xC004): Set 0xb0 to do the data mode.
- NDFDTR (0xC000): Write 512 bytes data.

(4) Read ECC data
- NDFMCR (0xC004): Set 0xd0 to do the ECC data read mode.
- NDFDTR (0xC000): Read 6 bytes ECC data.

First data:	LPR[7:0]
Second data:	LPR[15:8]
Third data:	CPR[5:0], 2'b11
Fourth data:	LPR[23:16]
Fifth data:	LPR[31:24]
Sixth data:	CPR[11:6], 2'b11

(5) Write 16 bytes redundant data
- NDFMCR (0xC004): Set 0x90 to do the data mode without ECC.
- NDFDTR (0xC000): Write 16 bytes redundant data.

  D520: LPR[23:16]

  D521: LPR[31:24]

  D522: CPR[11:6], 2'b11

  D525: LPR[7:0]

  D526: LPR[15:8]

  D527: CPR[5:0], 2'b11

(6) Run Page Program
- NDFMCR (0xC004): Set 0x91 to assert ND_CLE* signal and do the command mode.
- NDFDTR (0xC000): Set 0x10 to write the Page Program command.
- NDFMCR (0xC004): Set 0x10 to deassert ND_ALE* signal.
- NDFSR (0xC008): Check BUSY flag. If it's zero, go to the next. If it's one, wait till it will become zero.

(7) Read Status
- NDFMCR (0xC004): Set 0x11 to assert ND_CLE* signal and do the command mode.
- NDFDTR (0xC000): Set 0x70 to write the Read Status command.
- NDFMCR (0xC004): Set 0x10 to deassert ND_CLE* signal.
- NDFDTR (0xC000): Read the Status data from the NAND Flash memory.

(8) Continue from (1) to (7) for all other pages if need.

18.3.1.3 Read

The read sequence is below.

(1) NDFMCR (0xC004): Set 0x70 to reset ECC data.

(2) Read 512 bytes
- NDFMCR (0xC004): Set 0x11 to assert ND_CLE* signal and do the command mode. (2-1)
- NDFDTR (0xC000): Set 0x00 to write the Read command. (2-2)
- NDFMCR (0xC004): Set 0x12 to assert ND_ALE* signal and do the address mode. (2-3)
- NDFDTR (0xC000): Set A[7:0], A[16:9], and A[24:17]. If need, set A[25]. (2-4)
- NDFMCR (0xC004): Set 0x10 to deassert ND_ALE* signal. (2-5)
- NDFSR (0xC008): Check BUSY flag. If it's zero, go to the next. If it's one, wait till it will become zero. (2-6)
- NDFMCR (0xC004): Set 0x30 to do the data mode with ECC calculation. (2-7)
- NDFDTR (0xC000): Read 512 bytes data. (2-8)
- NDFMCR (0xC004): Set 0x10 to do the data mode without ECC calculation. (2-9)
- NDFDTR (0xC000): Read 16 bytes redundant data. (2-10)

(3) Read ECC data
- NDFMCR (0xC004): Set 0x50 to do the ECC data read mode.
- NDFDTR (0xC000): Read 6 bytes ECC data.

  First data: LPR[7:0]
  Second data: LPR[15:8]
  Third data: CPR[5:0], 2'b11
  Fourth data: LPR[23:16]
  Fifth data: LPR[31:24]
  Sixth data: CPR[11:6], 2'b11

(4) Compare ECC data and run the error routine if error occurs by Software.

(5) Read other pages
- NDFMCR (0xC004): Set 0x10.
- NDFSR (0xC008): Check BUSY flag. If it's zero, go to the next. If it's one, wait till it will become zero.
- Continue from (1) to (4) but (2-1) to (2-5) can be skipped when Sequential Read.

18.3.1.4 ID Read

The ID read sequence is below.

(1) NDFMCR (0xC004): Set 0x11 to assert ND_CLE* signal and do the command mode.
(2) NDFDTR (0xC000): Set 0x90 to write the ID Read command.
(3) NDFMCR (0xC004): Set 0x12 to assert ND_ALE* signal and do the address mode.
(4) NDFDTR (0xC000): Set 0x00.
(5) NDFMCR (0xC004): Set 0x10 to do the data mode without ECC calculation.
(6) NDFDTR (0xC000): Read Maker Code.
(7) NDFDTR (0xC000): Read Device Code.

18.3.2 ECC Control

NDFMC has the ECC calculating circuits. The circuits are controlled by NDFMCR. The Software compares the ECC data and checks if error or not.

The calculated ECC data can be read from NDFDTR register when NDFMCR is 0xD0 (Write mode) or 0x50 (Read mode). It is six bytes and six read operations for NDFDTR are needed. The order of the data is following as.

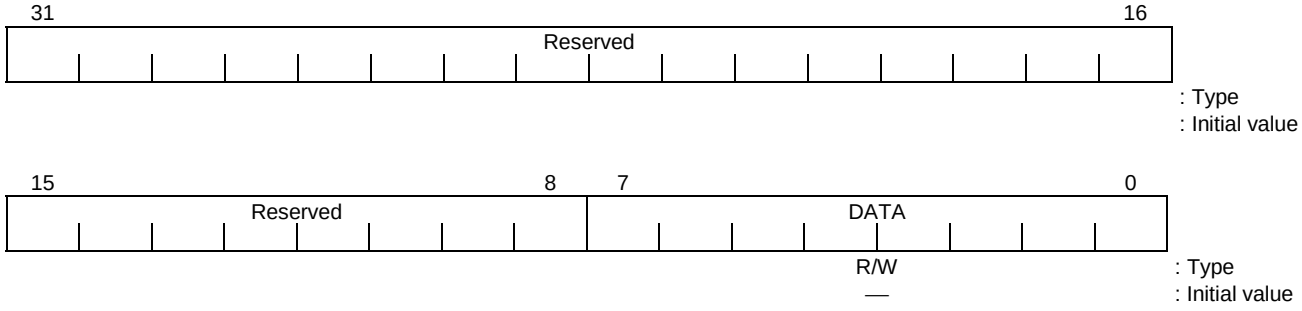
First data: LPR[7:0]
Second data: LPR[15:8]
Third data: CPR[5:0], 2'b11
Fourth data: LPR[23:16]
Fifth data: LPR[31:24]
Sixth data: CPR[11:6], 2'b11

18.4 Registers

Table 18.4.1 NAND Flash Memory Control Registers

Reference	Address	Bit Width	Register	Register Name
18.4.1	0xC000	32	NDFDTR	NAND Flash memory Data transfer Register
18.4.2	0xC004	32	NDFMCR	NAND Flash memory Mode Control Register
18.4.3	0xC008	32	NDFSR	NAND Flash memory Status Register
18.4.4	0xC00C	32	NDFISR	NAND Flash memory Interrupt Status Register
18.4.5	0xC010	32	NDFIMR	NAND Flash memory Interrupt Mask Register
18.4.6	0xC014	32	NDFSPR	NAND Flash memory Strobe pulse width Register
18.4.7	0xC018	32	NDFRSTR	NAND Flash memory Reset Register

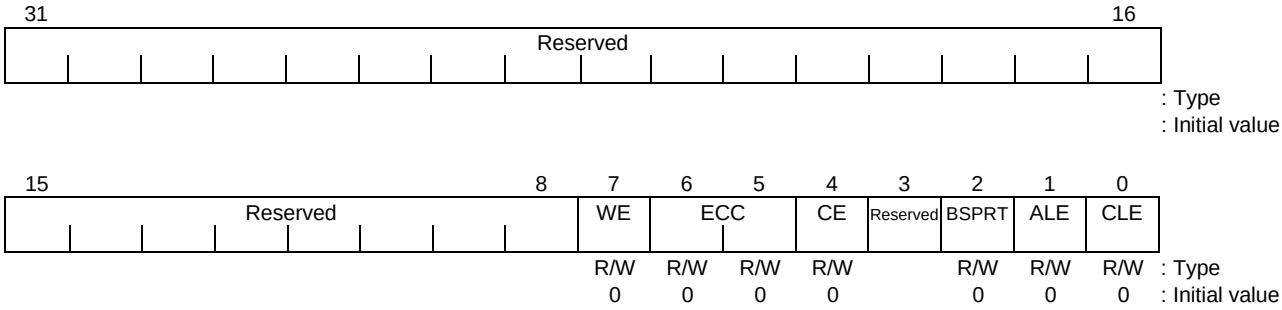
18.4.1 NAND Flash Memory Data Transfer Register (NDFDTR) 0xC000



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:0	DATA	DATA	Interrupt Detection Enable (Initial value: undefined, R/W) NAND Flash memory data. Read: The data is read from the NAND Flash memory. Note: Be able to read the data that was read from the NAND Flash memory at the previous read cycle for the NAND Flash memory. At the same time the NANDFC runs the next read cycle and stores the data in the buffer. Please take care to read the first byte data and the last byte data. Write: The data is written to the NAND Flash memory.

Figure 18.4.1 NAND Flash Memory Data Transfer Register (NDFDTR)

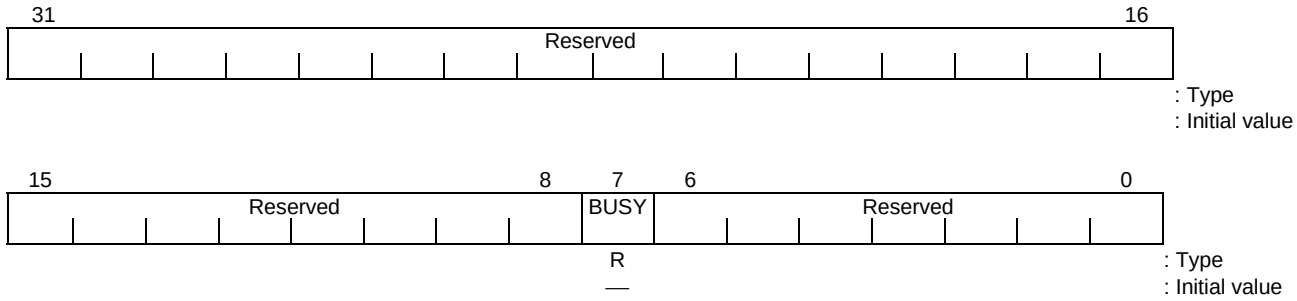
18.4.2 NAND Flash Memory Mode Control Register (NDFMCR) 0xC004



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	WE	Write Enable	Write Enable (Initial value: 0, R/W) This bit enables the data write operation. When you write the data in the NAND flash memory, this bit must be set one. 0: Inhibit write operation 1: Enable write operation
6:5	ECC	ECC Control	ECC Control 19 (Initial value: 00, R/W) These bits control the ECC calculating circuits. ECC 11: Reset ECC circuits. 00: ECC circuits is disable. 01: ECC circuits is enable. 10: Read ECC data calculated by NDFMC.
4	CE	Chip Enable	Chip Enable (Initial value: 0, R/W) Enable NAND Flash access. This bit must be set one when access to the NAND Flash memory. 0: Disable (ND_CE* is high.) 1: Enable (ND_CE* is low.)
3	—	Reserved	—
2	BSPRT	Bus Separate	Bus Separate (Initial value: 0, R/W) This bit enables the BUSSPRT* signal during NAND Flash memory access. 0: Disable 1: Enable
1	ALE	Address Latch Enable	Address Latch Enable (Initial value: 0, R/W) This bit specifies the value of ND_ALE signal. 0: Low 1: High
0	CLE	Command Latch Enable	Command Latch Enable (Initial value: 0, R/W) This bit specifies the value of ND_CLE signal. 0: Low 1: High

Figure 18.4.2 NAND Flash Memory Mode Control Register (NDFMCR)

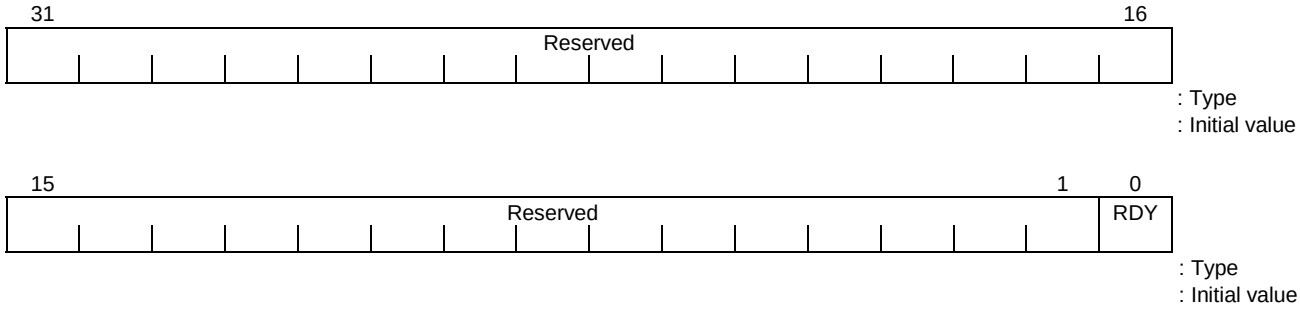
18.4.3 NAND Flash Memory Status Register (NDFSR) 0xC008



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	BUSY	BUSY	BUSY (Initial value: undefined, R) This bit shows the status of NAND flash memory. 0: Ready 1: Busy
6:0	—	Reserved	—

Figure 18.4.3 NAND Flash Memory Status Register (NDFSR)

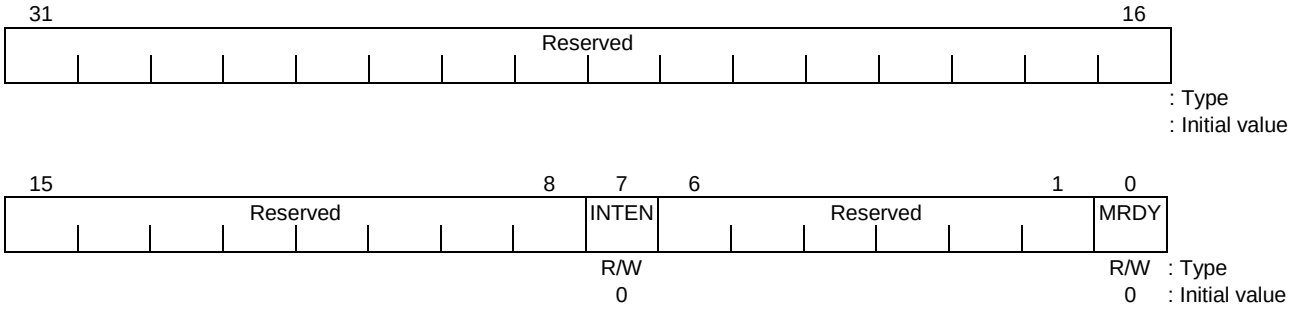
18.4.4 NAND Flash Memory Interrupt Status Register (NDFISR) 0xC00C



Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	RDY	Ready	Ready (Initial value: 0) This bit is set when ND_RB* signal changes from Low to High if MRDY in NDFIMR is one. Writing “0” clears this bit to zero. Read: 0: None 1: Change ND_RB* signal from Busy to Ready. Write: 0: No change 1: Clear to zero

Figure 18.4.4 NAND Flash Memory Interrupt Status Register (NDFISR)

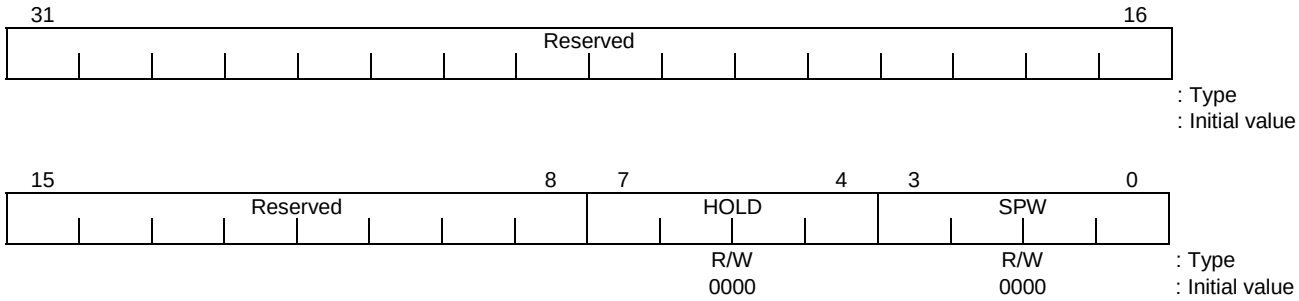
18.4.5 NAND Flash Memory Interrupt Mask Register (NDFIMR) 0xC010



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7	INTEN	Interrupt Enable	Interrupt Enable (Initial value: 0, R/W) Enable Interrupt. When this bit and MRDY bit in this register are set one and RDY bit in NDFISR becomes one, the interrupt occurs. 0: Disable 1: Enable
6:1	—	Reserved	—
0	MRDY	Mask RDY interrupt	Mask Ready Interrupt (Initial value: 0, R/W) This bit masks the RDY bit in NDFISR. If this bit one, RDY in NDFISR set when ND_RB* signal changes from Low to High. 0: Disable RDY in NDFISR 1: Enable RDY in NDFISR

Figure 18.4.5 NAND Flash Memory Interrupt Mask Register (NDFIMR)

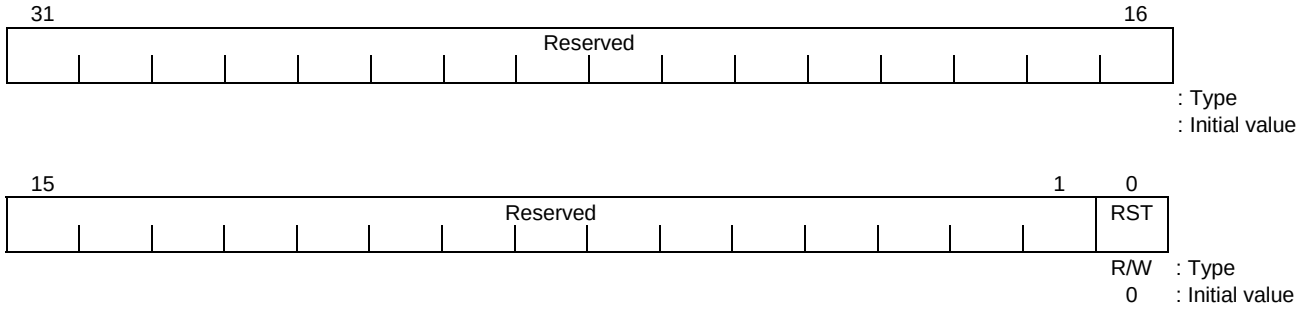
18.4.6 NAND Flash Memory Strobe Pulse Width Register (NDFSPR) 0xC014



Bits	Mnemonic	Field Name	Description
31:8	—	Reserved	—
7:4	HOLD	Hold Time	Hold Time (Initial value: 0000, R/W) These bits specify the Hold time from the rising edge of the ND_RE* and ND_WE* signals to the end of the bus cycle. The BUSSPRT* signal is deasserted after this hold timing if the NDFMCR.BSPRT is one. 0000: Zero 0001: One GBUSCLK 0010: Two GBUSCLKs 0011: Three GBUSCLKs 0100: Four GBUSCLKs 0101: Five GBUSCLKs 0110: Six GBUSCLKs 0111: Seven GBUSCLKs 1000: Eight GBUSCLKs 1001: Nine GBUSCLKs 1010: Ten GBUSCLKs 1011: Eleven GBUSCLKs 1100: Twelve GBUSCLKs 1101: Thirteen GBUSCLKs 1110: Fourteen GBUSCLKs 1111: Fifteen GBUSCLKs
3:0	SPW	Strobe Pulse Width	Strobe Pulse Width (Initial value: 0000, R/W) These bits specify the Low pulse width of the ND_RE* and ND_WE* signals. The low pulse width is the value of this field plus one. 0000: One GBUSCLK 0001: Two GBUSCLKs 0010: Three GBUSCLKs 0011: Four GBUSCLKs 0100: Five GBUSCLKs 0101: Six GBUSCLKs 0110: Seven GBUSCLKs 0111: Eight GBUSCLKs 1000: Nine GBUSCLKs 1001: Ten GBUSCLKs 1010: Eleven GBUSCLKs 1011: Twelve GBUSCLKs 1100: Thirteen GBUSCLKs 1101: Fourteen GBUSCLKs 1110: Fifteen GBUSCLKs 1111: Sixteen GBUSCLKs

Figure 18.4.6 NAND Flash Memory Strobe Pulse Width Register (NDFSPR)

18.4.7 NAND Flash Memory Reset Register (NDFRSTR) 0xC018



Bits	Mnemonic	Field Name	Description
31:1	—	Reserved	—
0	RST	Reset	Reset (Initial value: 0, R/W) Setting this bit reset the NANDFC. After reset, this bit is cleared automatically. 0: Don't care 1: Reset

Figure 18.4.7 NAND Flash Memory Reset Register (NDFRSTR)

18.5 Timing Diagrams

18.5.1 Command and Address Cycle

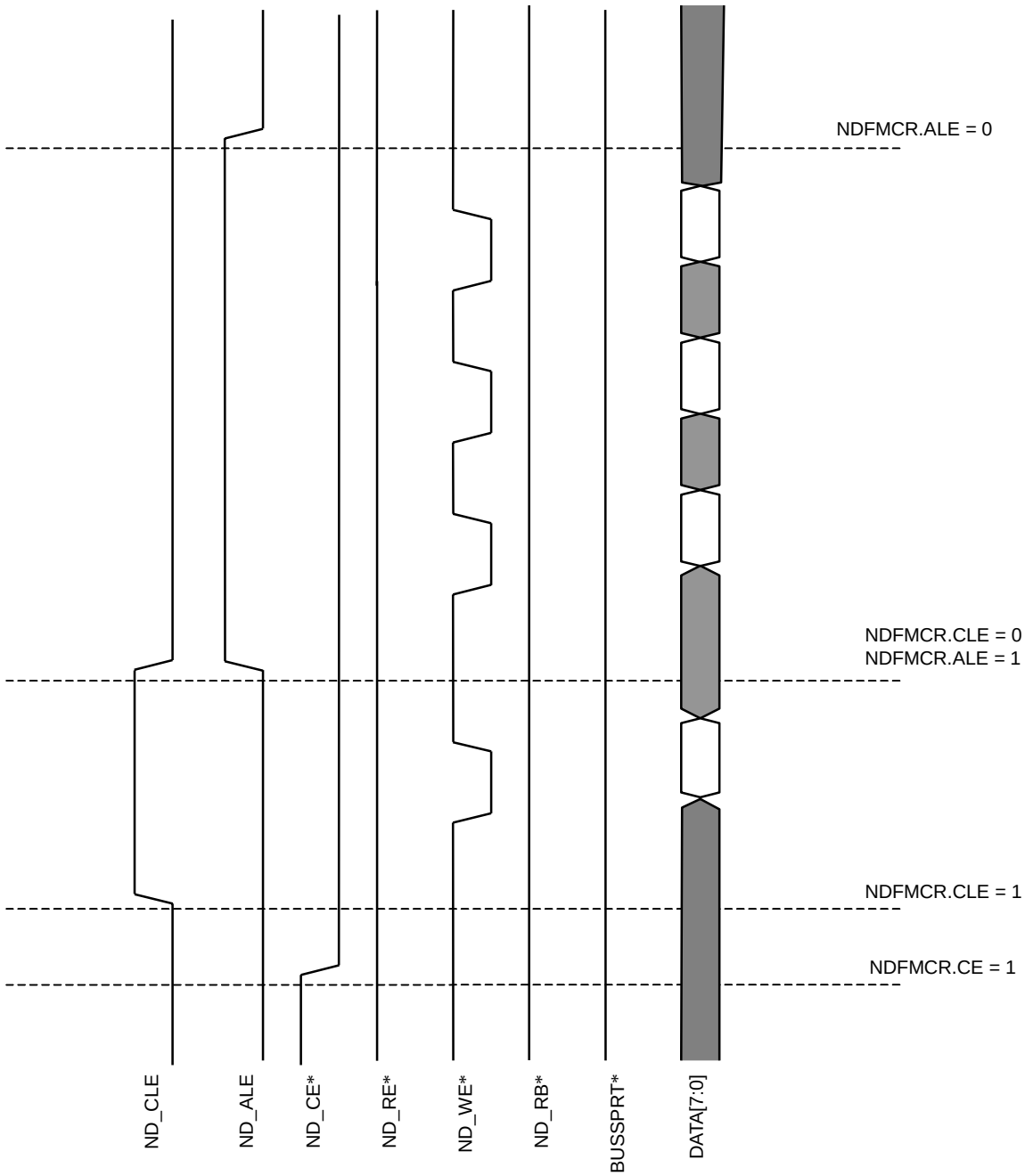


Figure 18.5.1 Command and Address Cycle (NDFMCR.BSPRT = 0)

18.5.2 Data Read Cycle

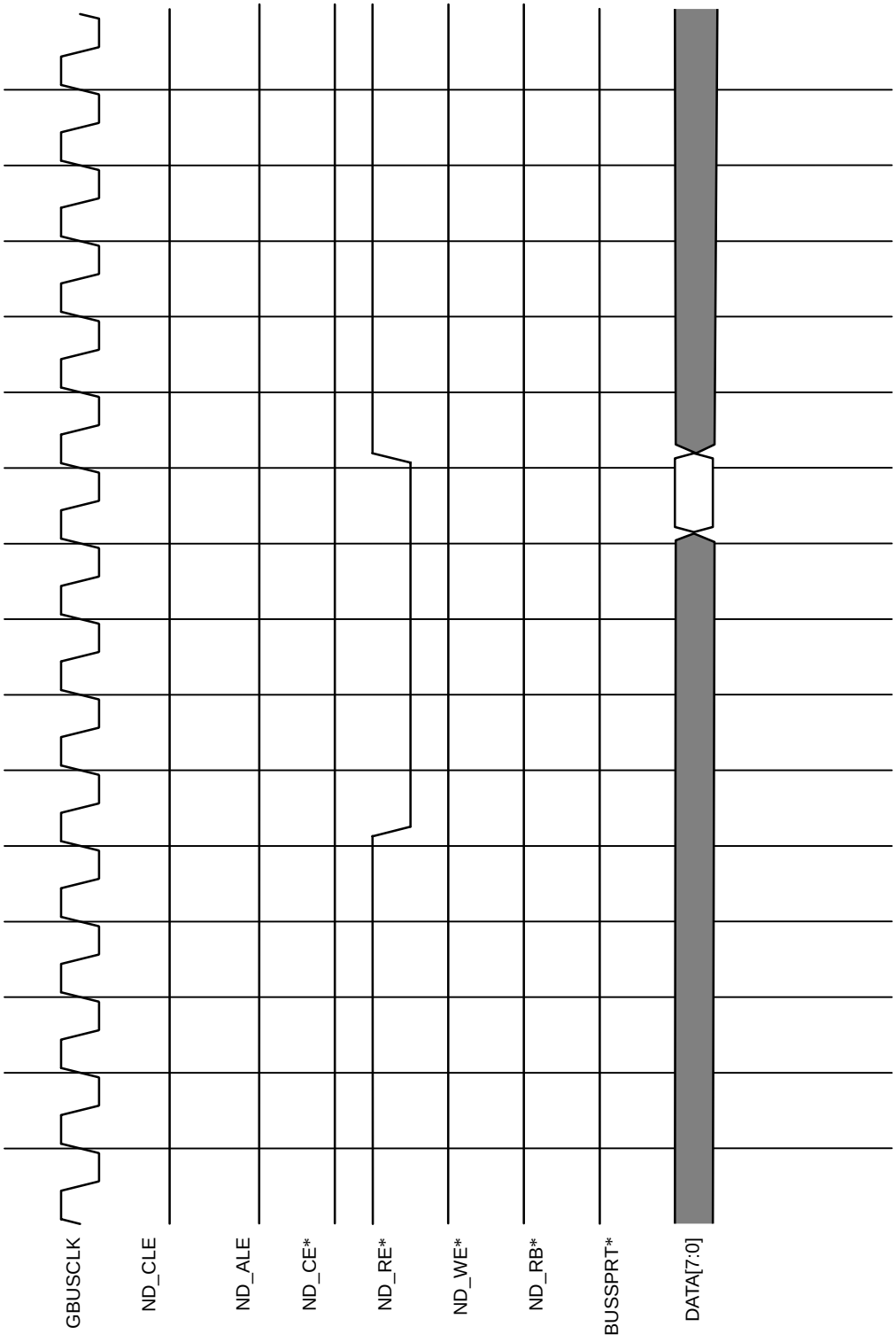


Figure 18.5.2 Data Read Cycle (NDFMCR.BSPRT = 0, NDFSPR = 0x34)

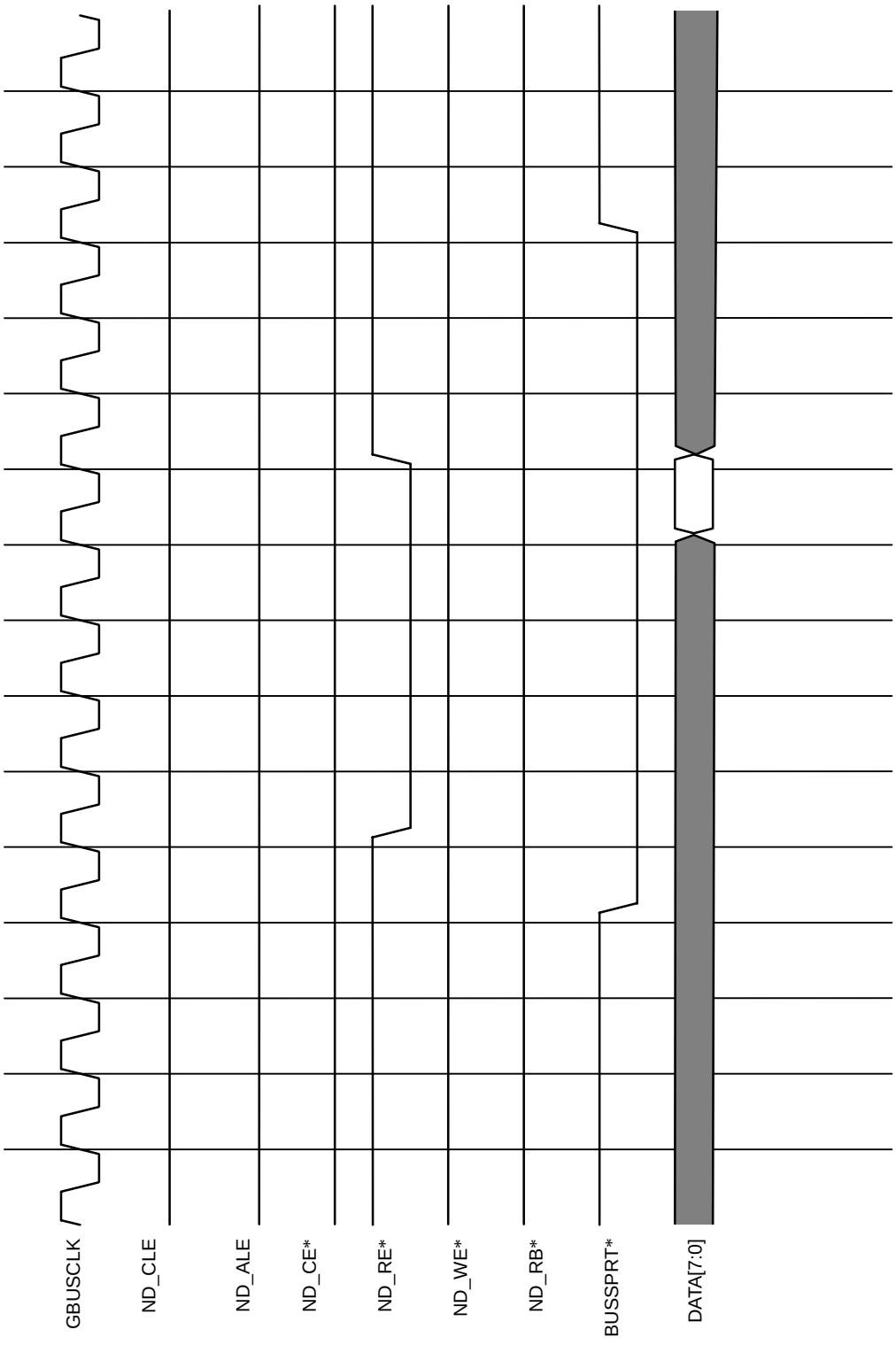


Figure 18.5.3 Data Read Cycle (NDFMCR.BSPRT = 1, NDFSPR = 0x34)

18.5.3 Data Write Cycle

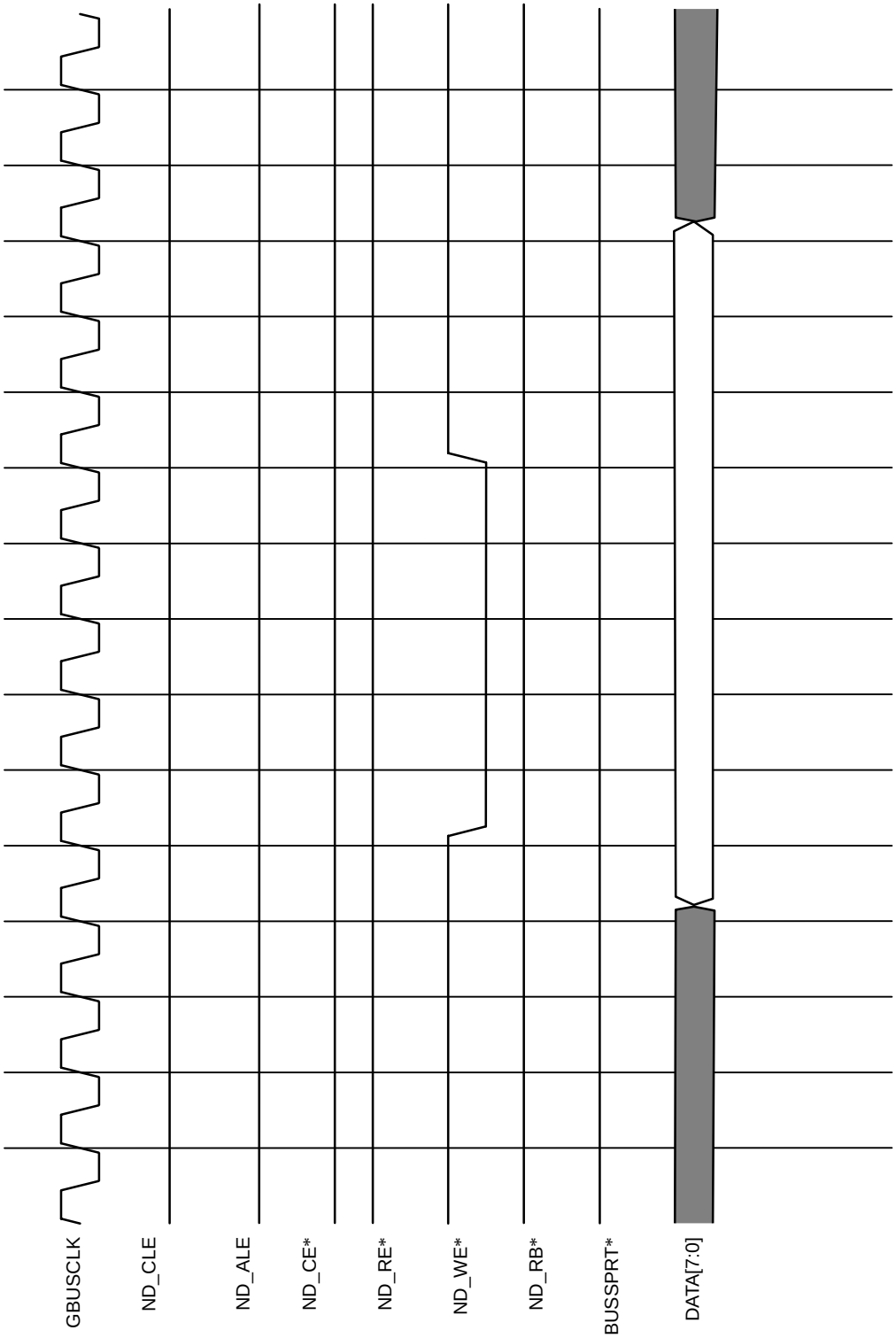


Figure 18.5.4 Data Write Cycle (NDFMCR.BSPRT = 0, NDFSPR = 0x34 or Use 74xx245 type, NDFMCR.BSPRT = 1, NDFSPR = 0x34)

18.6 Example of Using NAND Flash Memory

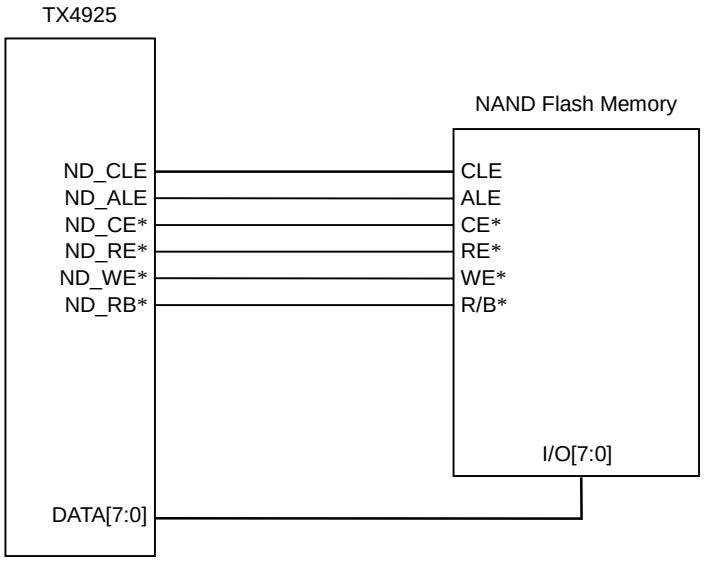


Figure 18.6.1 Example of Using NAND Flash Memory

19. Real Time Clock (RTC)

19.1 Features

Real Time Clock (RTC) is a 44-bit counter that uses a 32.768 kHz clock. The counter will provide a maximum count of 6213 days. Also includes is a 44-bit alarm register for the RTC that allows the software to set an alarm at any desired count of the RTC counter. The RTC will generate two interrupts for the CPU. The first is the ALARMINT that will generate an interrupt whenever the RTC reaches the value set by the alarm. The second is the RTCINT that will generate an interrupt whenever the RTC counter "rolls over" after reaching a count of 6213 days.

19.2 Block Diagrams

Figure 19.2.1 shows the RTC Block Diagram.

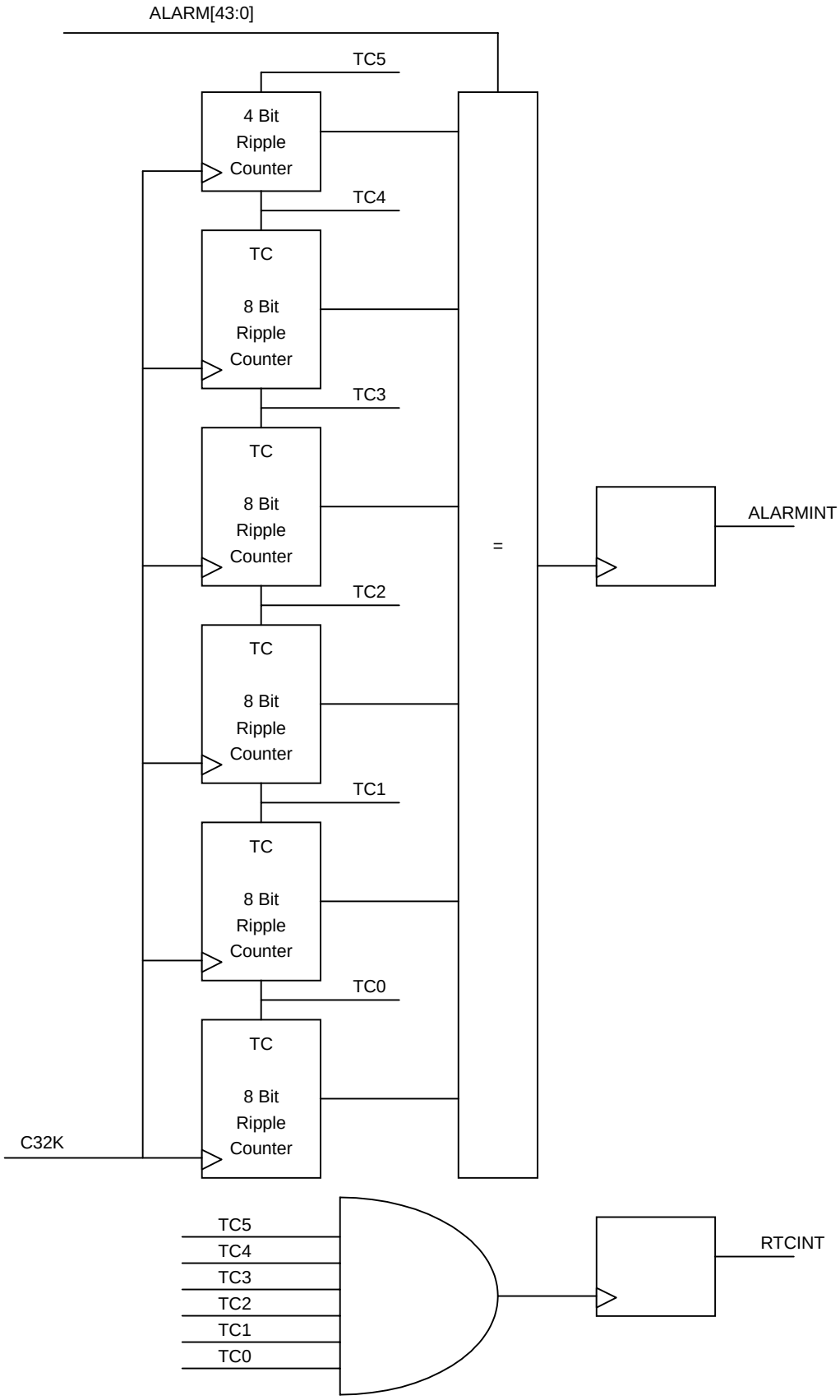


Figure 19.2.1 RTC Block Diagram

19.3 Operations

19.3.1 Operation

The RTC contains five 8-bit ripple counters connected in series. The first counter counts on each C32K clock, while each successive counter only counts when the previous count stage has reached a count of "0xFF". Once the counters reach a count of "0xFFFFFFFFFFF", the RTCINT interrupt will assert to indicate that the counter is "rolling over". Given a 44-bit counter for the RTC and an input clock C32K of 32.768 kHz, the time until RTCINT interrupt will assert is 6213 days.

The software can generate an alarm interrupt(ALARMINT) by setting the ALARM[43:0] bits in the Alarm Register. Whenever the RTC becomes equal to the value set in the Alarm Register, the ALARMINT will be triggered. The value of the RTC counter can be read via the RTC Register.

19.3.2 Interrupt

The RTC has two types interrupt sources. OR signal of them connects to the internal Interrupt Controller (IRC). Please check RTC Interrupt Status Register (RTCINT) to know which type of interrupt occurred.

Type	Status Bits	Mask-able Bit
RTCINT	RTCINT in RTCINT	DSRTCINT in RTCCTRL
ALARMINT	ALARMINT in RTCINT	DSALINT in RTCCTRL

RTCINT:

This interrupt is set 44 bits of the RTC counter reach a value of "0xFFFFFFFFFFF" to alert the software that the counter is "rolling over".

ALARMINT:

This interrupt is set whenever the RTC counter reaches a count that is equal to the value of the ALARM[43:0] bits set in the ALAMHI and ALARMLO Registers.

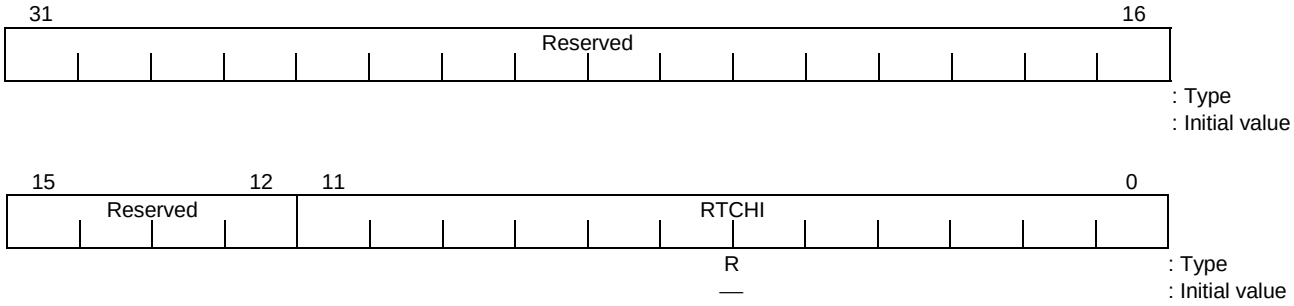
19.4 Registers

All registers should be accessed only as full word (32-bit) accesses. Any other type of access produces an undefined result. Please write "0" to the undefined bit.

Table 19.4.1 RTC Module Registers

Reference	Address	Bit Width	Register Symbol	Register Name
19.4.1	0xF900	32-bit	RTCHI	RTC Register (High)
19.4.2	0xF904	32-bit	RTCLO	RTC Register (Low)
19.4.3	0xF908	32-bit	ALARMHI	Alarm Register (High)
19.4.4	0xF90C	32-bit	ALARMLO	Alarm Register (Low)
19.4.5	0xF910	32-bit	RTCCTRL	RTC Control Register
19.4.6	0xF914	32-bit	RTCINT	RTC Interrupt Status Register

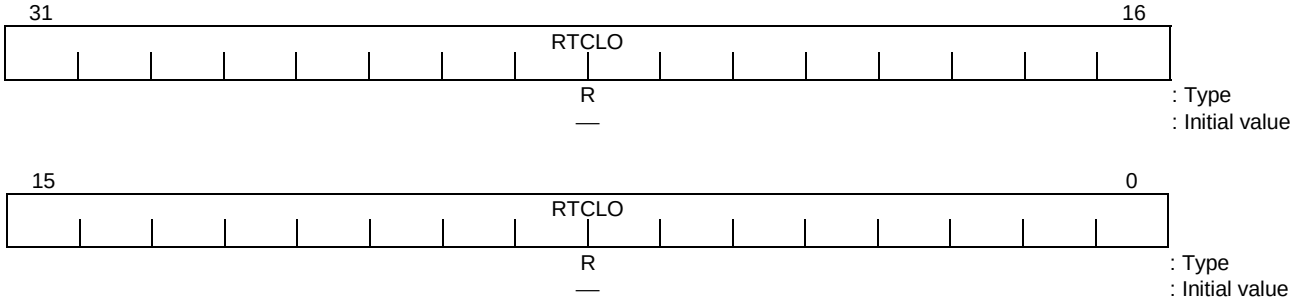
19.4.1 RTC Register (High) (RTCHI) 0xF900



Bits	Mnemonic	Field Name	Explanation
31:12	—	Reserved	—
11:0	RTCHI	RTC Register (High)	RTC Register (High) (Initial value: undefined, R) These bits provide the status of the bit 43 to 32 of RTC counter. The software must read these bits twice and compare the values to ensure that the counter is not read while the counter is counting since the CPU clock is not synchronous with the RTC counter clock. If the two reads do not compare, the software must read the register again to read the correct counter value.

Figure 19.4.1 RTC Register (High) (RTCHI)

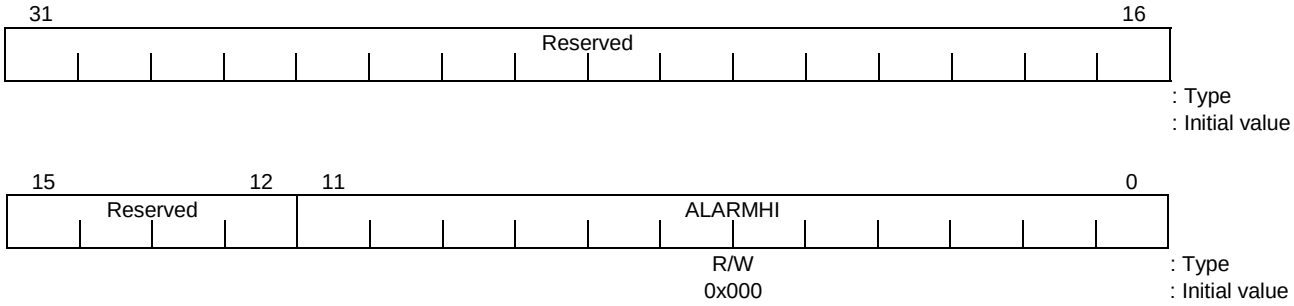
19.4.2 RTC Register (Low) (RTCLO) 0xF904



Bits	Mnemonic	Field Name	Explanation
31:0	RTCLO	RTC Register (Low)	RTC Register (Low) (Initial value: undefined, R) These bits provide the status of the bit 31 to 0 of RTC counter. The software must read these bits twice and compare the values to ensure that the counter is not read while the counter is counting since the CPU clock is not synchronous with the RTC counter clock. If the two reads do not compare, the software must read the register again to read the correct counter value.

Figure 19.4.2 RTC Register (Low) (RTCLO)

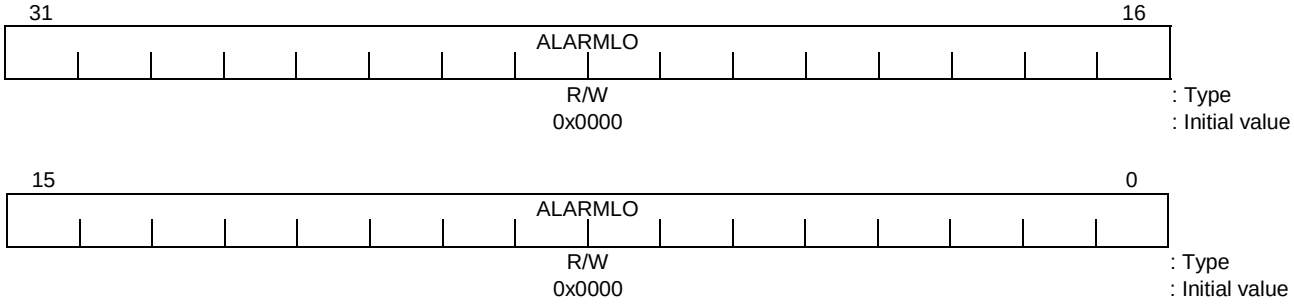
19.4.3 Alarm Register (High) (ALARMHI) 0xF908



Bits	Mnemonic	Field Name	Explanation
31:12	—	Reserved	—
11:0	ALARMHI	Alarm Register (High)	Alarm Register (High) (Initial value: 0x000, R/W) These bits provide the status of the bit 43 to 32 of Alarm counter. Whenever the RTC counter reaches a count that is equal to the value of join to these bits to Alarm Register (Low), the ALARMINT interrupt will be set.

Figure 19.4.3 Alarm Register (High) (ALARMHI)

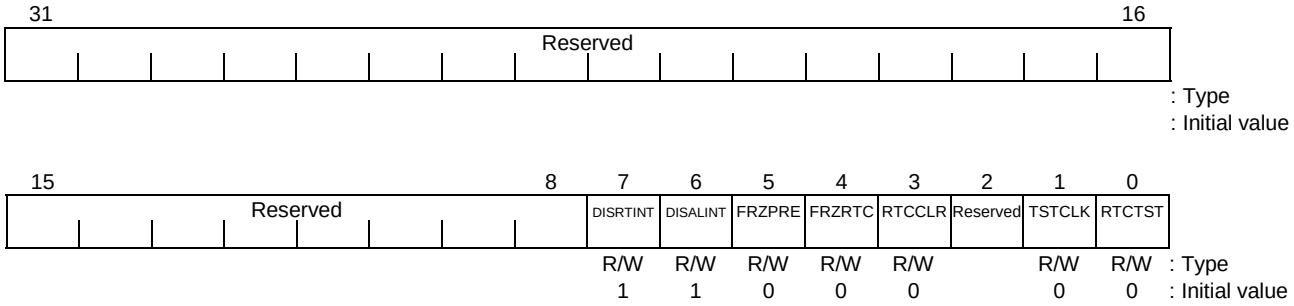
19.4.4 Alarm Register (Low) (ALARMLO) 0xF90C



Bits	Mnemonic	Field Name	Explanation
31:0	ALARMLO	Alarm Register (Low)	Alarm Register (Low) (Initial value: 0x0000_0000, R/W) These bits provide the status of the bit 31 to 0 of Alarm counter. Whenever the RTC counter reaches a count that is equal to the value of join to Alarm Register (High) to these bits, the ALARMINT interrupt will be set.

Figure 19.4.4 Alarm Register (Low) (ALARMLO)

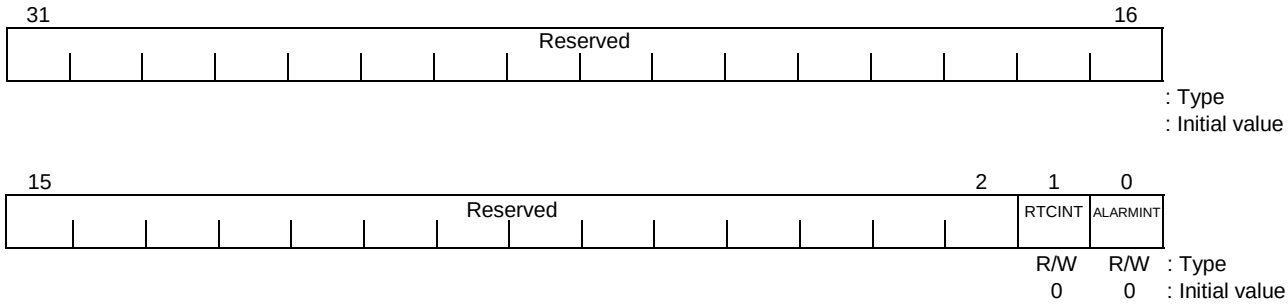
19.4.5 RTC Control Register (RTCCTRL) 0xF910



Bits	Mnemonic	Field Name	Explanation
31:8	—	Reserved	—
7	DISRTINT	Disable RTC Interrupt	Disable RTC Interrupt (Initial value: 1, R/W) Disable RTC Interrupt. If clear, the interrupt occurs when RTC counter reaches a value of “0xfffffffff”. 1: Disable 0: Enable
6	DISALINT	Disable Alarm Interrupt	Disable Alarm Interrupt (Initial value: 1, R/W) Disable Alarm Interrupt. If clear, the interrupt occurs when RTC counter reaches a value that is equal to the value of Alarm count. 1: Disable 0: Enable
5	FRZPRE	Freeze Prescaler	Freeze Prescaler (Initial value: 0, R/W) Setting this bit will cause the lower 8 bits of the RTC counter to freeze. This bit is a test bit and customers can’t use it. 1: Freeze 0: Run
4	FRZRTC	Freeze RTC	Freeze RTC (Initial value: 0, R/W) Setting this bit will cause the upper 36 bits of the RTC counter to freeze. This bit is a test bit and customers can’t use it. 1: Freeze 0: Run
3	RTCCLR	RTC Clear	RTC Clear (Initial value: 0, R/W) Setting this bit to a logic “1” will cause all 44 bits of the RTC counter to initialize to “0x0000000_0000”. The RTC counter will stay cleared and the counter will not start counting until this bit is cleared back to a logic “0”. 1: Stop 0: Run
2	—	Reserved	—
1	TSTCLK	Enable Test Clock	Enable Test Clock (Initial value: 0, R/W) Setting this bit will cause the 32 kHz input clock for the RTC counter to be driven by the IMBUSCLK instead of the 32 kHz input pin. This bit is a test bit and customers can’t use it. 0: 32 KHz input 1: IMBUSCLK
0	RTCTST	Enable RTC Test	Enable RTC Test (Initial value: 0, R/W) Setting this bit will cause all five of the 8-bit counters and the 4-bit counter that comprise the RTC counter to count together. This bit is a test bit and customers can’t use it. 0: Normal run 1: Test run

Figure 19.4.5 RTC Control Register (RTCCTRL)

19.4.6 RTC Interrupt Status Register (RTCINT) 0xF914



Bits	Mnemonic	Field Name	Explanation
31:2	—	Reserved	—
1	RTCINT	RTC Interrupt Status	RTC Interrupt Status (Initial value: 0, R/W) This bit shows the RTC Interrupt Status. This bit is cleared by written “0”. 0: No interrupt 1: Interrupt occurs
0	ALARMINT	Alarm Interrupt Status	Alarm Interrupt Status (Initial value: 0, R/W) This bit shows the Alarm Interrupt Status. This bit is cleared by written “0”. 0: No interrupt 1: Interrupt occurs

Figure 19.4.6 RTC Interrupt Status Register (RTCINT)

20. Removed

21. Extended EJTAG Interface

21.1 Extended EJTAG Interface

The TX4925 Extended EJTAG (Enhanced Joint Test Action Group) Interface provides two real-time debugging functions. One is the IEEE1149.1 standard compliant JTAG Boundary Scan Test, and the other is the Debugging Support Unit (DSU) that is built into the TX49/H2 core.

JTAG Boundary Scan Test

- IEEE1149.1 compatible TAP Controller
- Supports the following five instructions: EXTEST, SAMPLE/PRELOAD, IDCODE, BYPASS, HIGHZ

Real-time Debugging

- Real-time debugging using an emulation probe (made by Corelis or YDC)
- Execution control (run, break, step, register/memory access)
- Real-time PC tracing

Please contact your local Toshiba Sales representative for more information regarding how to connect the emulation probe.

The two functions of the Extended EJTAG Interface operate in one of two modes.

PC Trace Mode

- Execution control (fun, pause, access single steps, access internal register/system memory)
- JTAG Boundary Scan Test

Real-time Mode

- Real-time PC tracing

Refer to Section 3.1.14 for more information regarding signals used with the Extended EJTAG Interface.

Table 21.1.1 EJTAG Interface Function and Operation Code

PC Tracing Mode	Off	On
JTAG Boundary Scan	Boundary Scan Test	—
Real-time Debugging	Execution Control	Real-time PC Tracing

21.2 JTAG Boundary Scan Test

21.2.1 JTAG Controller and Register

The Extended EJTAG Interface contains a JTAG Controller (TAP Controller) and a Control Register. This section explains only those portions that are unique to the TX4925. Please refer to the TX49/H2 Core Architecture Manual for all other portion not covered here. Please contact your local Toshiba Sales representative for more information regarding the required BSDL files when performing the JTAG Boundary Scan Test.

- Instruction Register (Refer to 21.2.2)
- Data Register
  - Boundary Scan Register (Refer to 21.2.3)
  - Bypass Register
  - Device ID Register (Refer to 21.2.4)
  - JTAG Address Register
  - JTAG Data Register
  - JTAG Control Register
  - EJTAG Mount Register
- Test Access Port Controller (TAP Controller) (Refer to 21.3)

21.2.2 Instruction Register

The JTAG Instruction Register consists of an 8-bit shift register. This register is used for selecting either one or both of the test to be performed and the Test Data Register to be accessed. The Data Register is selected according to the instruction code in Table 21.2.1. Refer to the "64-Bit TX System RISC TX49/H2 Core Architecture" for more information regarding each instruction.

Table 21.2.1 Bit Configuration of JTAG Instruction Register

Instruction Code MSB → LSB	Instruction	Selected Data Register
00000000 (0x00)	EXTEST	Boundary Scan Register
00000001 (0x01)	SAMPLE/PRELOAD	Boundary Scan Register
00000010 (0x02)	Reserved	Reserved
00000011 (0x03)	IDCODE	Device ID Register
00000100 - 00001111	Reserved	Reserved
00010000 (0x10)	HIGHZ	Bypass Register
00010001 - 01111111	Reserved	Reserved
10000000 - 11111110	Refer to the TX49/H2 Core Architecture Manual	
11111111 (0xFF)	BYPASS	Bypass Register

Figure 21.2.1 shows the format of the Instruction Register.

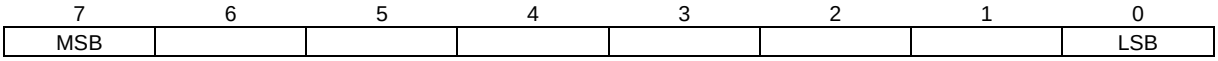


Figure 21.2.1 Instruction Register

The instruction code is shifted to the Instruction Register starting from the Least Significant Bit.

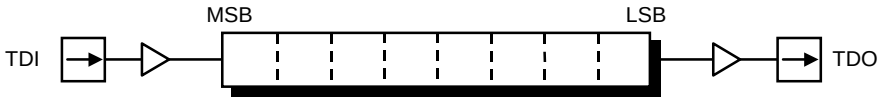


Figure 21.2.2 Shift Direction of the Instruction Register

21.2.3 Boundary Scan Register

The Boundary Scan Register contains a single 178-bit shift register to which all TX4925 I/O signals except for power supply, TDI, TCK, TDO, TMS, TRST* are connected. TEST* and SCANENB* cannot be tested, but it is possible for the Shift Register to sample the input. Figure 21.2.3. shows the bits of the Boundary Scan Register.

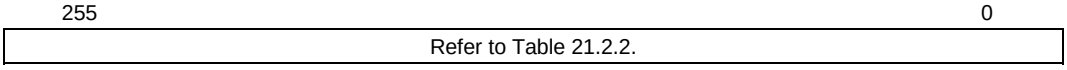


Figure 21.2.3 Boundary Scan Register

Table 21.2.2 shows the scan sequence of 178 signals starting from TDI and ending with TDO.

TDI input is fetched to the Least Significant Bit (LSB) of the Boundary Scan Register and the Most Significant Bit (MSB) of the Boundary Scan Register is sent from the TDO output.

Table 21.2.2 shows the boundary scan sequence relative to the processor signals.

Table 21.2.2 TX4925 Processor JTAG Scan Sequence (1/2)

JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name
	TDI	43	PCIAD[14]	86	PIO[8]
1	GNT[0]*	44	PCIAD[9]	87	PIO[9]
2	PCICLKIO	45	PCIAD[10]	88	PIO[12]
3	PCICLK[1]	46	PCIAD[11]	89	PIO[17]
4	PCICLK[2]	47	C_BE[0]	90	PIO[13]
5	REQ[0]*	48	PCIAD[8]	91	PIO[7]
6	GNT[1]*	49	PCIAD[4]	92	PIO[14]
7	REQ[1]*	50	PCIAD[0]	93	PIO[15]
8	GNT[2]*	51	PCIAD[5]	94	PIO[16]
9	REQ[2]*	52	PCIAD[1]	95	BC32K
10	GNT[3]*	53	PCIAD[6]	96	NMI*
11	REQ[3]*	54	PCIAD[2]	97	TEST*
12	PCIAD[31]	55	PCIAD[7]	98	MASTERCLK
13	PCIAD[30]	56	PCIAD[3]	99	DATA[0]
14	PCIAD[29]	57	BWE[0]*	100	DATA[16]
15	PCIAD[28]	58	BWE[1]*	101	DATA[1]
16	PCIAD[27]	59	SYSCLK	102	DATA[17]
17	PCIAD[26]	60	BWE[2]*	103	DATA[2]
18	PCIAD[25]	61	BWE[3]*	104	DATA[18]
19	PCIAD[24]	62	UAE	105	DATA[3]
20	C_BE[3]	63	SWE*	106	DATA[19]
21	ID_SEL	64	ADDR[0]	107	DATA[4]
22	PCIAD[23]	65	ADDR[1]	108	DATA[20]
23	PCIAD[22]	66	ADDR[2]	109	DATA[5]
24	PCIAD[21]	67	ADDR[3]	110	DATA[21]
25	PCIAD[20]	68	ADDR[4]	111	DATA[6]
26	PCIAD[19]	69	CE[3]*	112	DATA[22]
27	PCIAD[17]	70	CE[2]*	113	DATA[23]
28	PCIAD[18]	71	ADDR[15]	114	DATA[8]
29	FRAME*	72	OE*	115	DATA[7]
30	C_BE[2]	73	PIO[0]	116	DATA[24]
31	PCIAD[16]	74	PIO[2]	117	DATA[9]
32	STOP*	75	CE[1]*	118	DATA[25]
33	DEVSEL*	76	CE[0]*	119	DATA[26]
34	TRDY*	77	BUSSPRT	120	DATA[10]
35	IRDY*	78	PIO[4]	121	DATA[27]
36	SERR*	79	ACK*	122	DATA[11]
37	PERR*	80	PIO[3]	123	DATA[28]
38	PCIAD[15]	81	PIO[1]	124	DATA[12]
39	C_BE[1]	82	PIO[11]	125	DATA[29]
40	PAR	83	PIO[10]	126	DATA[13]
41	PCIAD[12]	84	PIO[5]	127	DATA[15]
42	PCIAD[13]	85	PIO[6]	128	DATA[30]

Table 21.2.2 TX4925 Processor JTAG Scan Sequence (2/2)

JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name	JTAG Scan Sequence	Signal Name
129	DATA[14]	146	ADDR[10]	163	SDCLKIN
130	RP*	147	ADDR[11]	164	PIO[20]
131	DATA[31]	148	ADDR[12]	165	PON*
132	DQM[0]	149	ADDR[13]	166	PIO[19]
133	CAS*	150	ADDR[14]	167	PIO[18]
134	WE*	151	SADDR10	168	PIO[23]
135	DQM[1]	152	ADDR[16]	169	PIO[22]
136	SDCS[0]*	153	ADDR[19]	170	PIO[21]
137	DQM[2]	154	ADDR[18]	171	PIO[27]
138	DQM[3]	155	ADDR[17]	172	PIO[29]
139	ADDR[5]	156	CKE	173	PIO[28]
140	RAS*	157	SDCS[2]*	174	PIO[30]
141	SDCS[1]*	158	SCANENB*	175	PIO[24]
142	ADDR[6]	159	SDCS[3]*	176	PIO[31]
143	ADDR[7]	160	SDCLK[0]	177	PIO[25]
144	ADDR[8]	161	SDCLK[1]	178	PIO[26]
145	ADDR[9]	162	RESET*		TDO

21.2.4 Device ID Register

The Device ID Register is a 32-bit shift register. This register is used for reading the ID code that expresses the IC manufacturer, part number, and version from the IC and sending it to a serial device. The following figure shows the configuration of the Device ID Register.

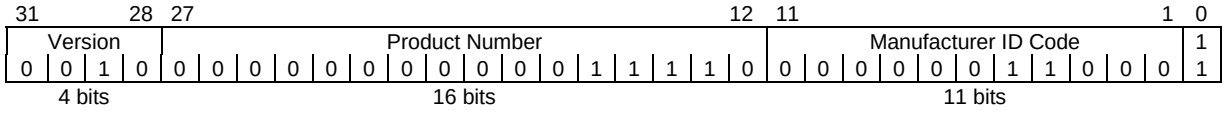


Figure 21.2.4 Device ID Register

The device ID code for the TX4925 is 0x0001_E031. However, the four top bits of the Version field may be changed. The device ID code is shifted out from the Least Significant Bit.

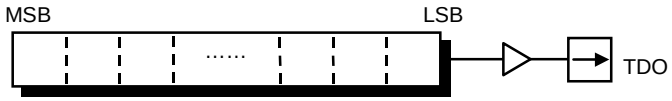


Figure 21.2.5 Shift Direction of the Device ID Register

21.3 Initializing the Extended EJTAG Interface

The Extended EJTAG Interface is not reset by asserting the RESET* signal. Operation of the TX49/H2 core is not guaranteed if the Extended EJTAG Interface is not reset. This interface is initialized by either of the following methods.

- Assert the TRST* signal.
- After clearing the processor reset, set the TMS input to High for five consecutive rising edges of the TCK input. The reset state is maintained if TMS is able to maintain the High state.

The above methods must be performed while the MASTERCLK signal is being input. Also, externally fix the TRST* signal to GND when not using an emulation probe. The G-Bus Time Out Detection function is disabled when the TRST* signal is deasserted. (Refer to Section 5.1.1.)

22. Electrical Characteristics

22.1 Absolute Maximum Rating (*1)

Parameter	Symbol	Rating	Unit
Supply voltage (for I/O)	$V_{CCIOMax}$	–0.3 ~ 3.9	V
Supply voltage (for internal)	$V_{CCIntMax}$	–0.3 ~ 3.0	V
Input voltage (*2)	V_{IN}	–0.3 ~ V_{CCIO} + 0.3	V
Storage Temperature	T_{STG}	–40 ~ +125	°C

*1: If LSI is used above the maximum ratings, permanent destruction of LSI can result. In addition, it is desirable to use LSI for normal operation under the recommended condition. If these conditions are exceeded, reliability of LSI may be adversely affected.

*2: The maximum rated VCCIOMax voltage must not be exceeded even at VCCIO + 0.3 volts.

22.2 Recommended Operating Conditions (*3)

Parameter		Symbol	Condition	Min.	Max.	Unit
Supply voltage	I/O	V_{CCIO}		3.0	3.6	V
	Internal	V_{CCINT}		1.4	1.6	V
Operating Case Temperature		T_C		0	70	°C

*3: Functional operation should be restricted to the recommended operating conditions. Those are the limits under which proper device operation is guaranteed. Therefore, the end product must be designed within the recommended voltage and temperature ranges indicated.

22.3 DC Characteristics

22.3.1 DC Characteristics Except for PCI Interface

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	SYM	Conditions	Min.	Max.	Unit
Low-level input voltage	V_{IL1}	(1)	-0.3	0.8	V
High-level input voltage	V_{IH1}	(1)	2.0	V_{CCIO} + 0.3	V
Low-level output current	I_{OL1}	(2)V_{OL} = 0.4 V	16	—	mA
	I_{OL2}	(3)V_{OL} = 0.4 V	8	—	mA
High-level output current	I_{OH1}	(2)V_{OH} = 2.4 V	—	−16	mA
	I_{OH2}	(3)V_{OH} = 2.4 V	—	−8	mA
Low-level input	I_{IL1}	(4)$V_{IN}=V_{SS}$	-10	10	μA
Leakage current	I_{IL2}	(5)$V_{IN}=V_{SS}$	-200	10	μA
High-level input Leakage current	I_{IH1}	(6)$V_{IN}=V_{CCIO}$	-10	10	μA
Operating current	I_{DDDInt}	Input 200 MHz, GBUSCLK 2.5 RF	—	600	mA
	I_{DDDIO}		—	150	mA

(1): All input and input-mode bidirectional pins except PCI interface signals.

(2): PIO[4,2,0], SYSCLK, BUSSPRT*, RP*, DQM[3:0], CAS*, WE*, SDCS[3:0], RAS*, CKE, SDCLK[1:0], ACK*, DATA[31:0], ADDR[19:16,14,5], SADDR10

(3): PIO[31:5,3,1], BWE[3:0]*, SWE*, CE[3:0]*, OE*, SDCLKIN, UAE, ADDR[15,4:0], BC32K, TDO

(4): PIO[17:12], TRST*, SDCLKIN, RESET*, PON*, TEST*, MASTERCLK, SCANENB*

(5): PIO[31:18,11:0], ACK*, DATA[31:0], ADDR[19:0], SADDR10, TCK, TDI, TMS, NMI*

(6): (4), (5) Signals

22.3.2 DC Characteristics Except for PCI Interface

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

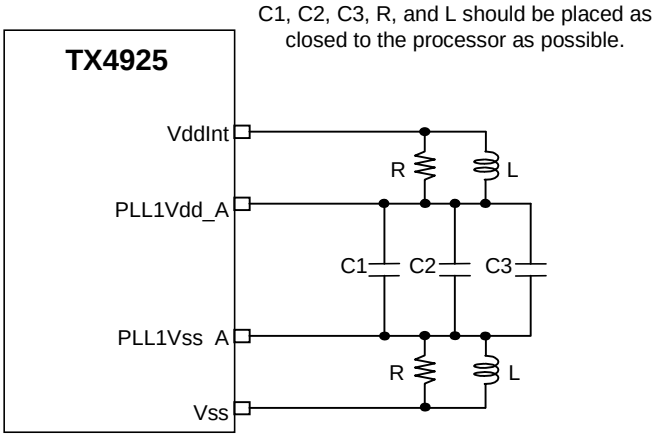
Parameter	SYM	Conditions	Min.	Max.	Unit
Low-level input voltage	V_{ILPCI}	(1)	-0.5	0.9	V
High-level input voltage	V_{IHPCI}	(1)	1.8	V_{CCIO} + 0.3	V
High-level output voltage	V_{OHPCI}	(2) I_{OUT} = –500 μA	$V_{CCIO} \times 0.9$	—	V
Low-level output voltage	V_{OLPCI}	(2) I_{OUT} = 1500 μA	—	$V_{CCIO} \times 0.1$	V
Input leakage current	I_{IHPCI} I_{ILPCI}	(1) $0 < V_{IN} < V_{CCIO}$	-10 -10	10 10	μA μA

(1): PCICLKIO, PCIAD[31:0], C_BE[3:0], PAR, FRAME*, IRDY*, TRDY*, STOP*, ID_SEL, DEVSEL*, REQ[3:0]*, GNT[3:0]*, PERR*, SERR

(2): All PCI interface except ID SEL.

22.4 Power Circuit for PLL

22.4.1 Recommended Circuit for PLL



Parameter	Symbol	AS a Reference Value	Unit
Resistor	R	5.6	Ω
Inductance	L	2.2	μH
Capacitor	C1 C2 C3	1 82 10	nF nF μF
VddInt / VddPLL		1.5 ± 0.1	V

Note: Reference

Figure 22.4.1 Power Circuit for PLL

22.5 AC Characteristics

22.5.1 MASTERCLK AC Characteristics

(Tc = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.3 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Parameter	SYM	Condition	Min.	Max.	Unit
MASTERCLK Period	t_{MCP}		50	—	ns
MASTERCLK Frequency (*1)	f_{MCK}		—	20	MHz
MASTERCLK High	t_{MCH}		3	—	ns
MASTERCLK Low	t_{MCL}		3	—	ns
Internal Operating Frequency	f_{CPU}		25	200	MHz
MASTERCLK Rise Time	t_{MCR}		—	2	ns
MASTERCLK Fall Time	t_{MCF}		—	2	ns

*1: Proper circuit operation of the TX4925 is guaranteed only when power supply to it is stable and the on-chip PLL is enabled.

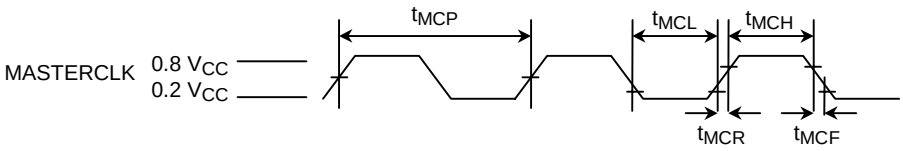


Figure 22.5.1 Timing Diagrams: MASTERCLK

22.5.2 Power On AC Characteristics

(Tc = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.3 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Parameter	SYM	Condition	Min.	Max.	Unit
PON* width time	t_{MCO_PLL}		1	—	ms
PLL stable time	t_{MCP_PLL}		10	—	ms
RESET* width time	t_{MCH_PLL}		1	—	ms

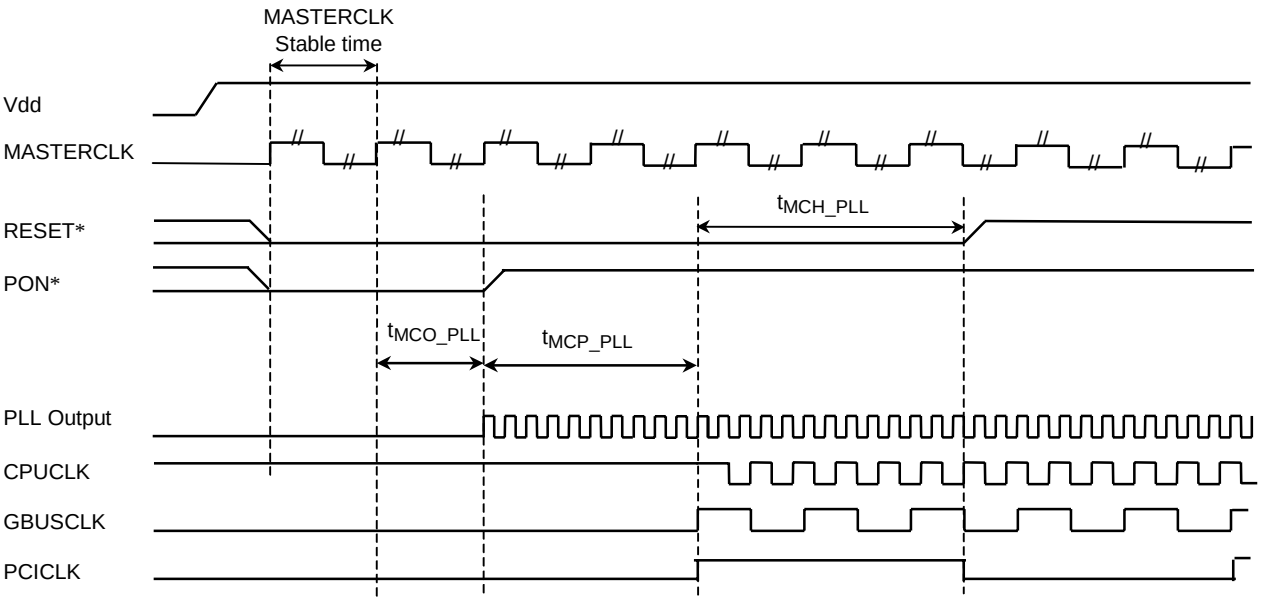


Figure 22.5.2 Timing Diagrams: POWER ON RESET

22.5.3 SDRAM Interface AC Characteristics

(Tc = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.3 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V, CL = 50 pF)

Parameter	Symbol	Rating	Min.	Max.	Unit
SDCLK[3:0] Cycle Time	t_{CYC_SDCLK}		12.5	—	ns
SDCLK[3:0] High Time	t_{HIGH_SDCLK}		3	—	ns
SDCLK[3:0] Low Time	t_{LOW_SDCLK}		3	—	ns
SDCLKIN Skew	t_{BP}	Non bypass mode	0	4.0	ns
ADDR[19:16,14:5],SADDR10 Output delay	t_{VAL_ADDR1}	(*1)	1.5	7.5	ns
SDCS[3:0]* Output delay	t_{VAL_SDCS}		1.5	7.5	ns
RAS* Output delay	t_{VAL_RAS}	(*1)	1.5	7.5	ns
CAS* Output delay	t_{VAL_CAS}	(*3)	1.5	7.5	ns
WE* Output delay	t_{VAL_WE}	(*3)	1.5	7.5	ns
CKE Output delay	t_{VAL_CKE}		1.5	7.5	ns
DQM[3:0] Output delay	t_{VAL_DQM}	(*2)	1.5	7.5	ns
DATA[31:0] Output delay (H->L, L->H)	t_{VAL_DATA1}	(*2)	1.5	7.5	ns
DATA[31:0] Output delay (High-Z->Valid)	$t_{VAL\_DATA1ZV}$	(*2)	1.5	7.5	ns
DATA[31:0] Output delay (Valid->High-Z)	$t_{VAL\_DATA1VZ}$	(*2)	1.5	7.5	ns
DATA[31:0] Input set-up time	t_{SU_DATA1B}	Bypass mode	6.0	—	ns
DATA[31:0] input hold time	t_{HO_DATA1B}	Bypass mode	0.5	—	ns
DATA[31:0] Input set-up time	$t_{SU\_DATA1NB}$	Non bypass mode	1.0	—	ns
DATA[31:0] input hold time	$t_{HO\_DATA1NB}$	Non bypass mode	2.0	—	ns

*1: An SDRAM bus transaction can complete in no more than two clock cycles when the SDCTR.DA is set to 1.

*2: An SDRAM bus transaction can complete in no more than two clock cycles when the SDCTR.SWB is set to 1.

*3: 2 cycle signals.

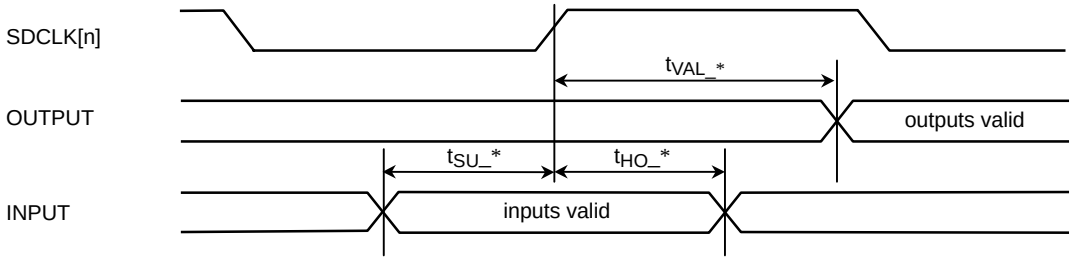


Figure 22.5.3 Timing Diagrams: Output Signals and When Bypass Mode Input Signals (SDCLK Basis)

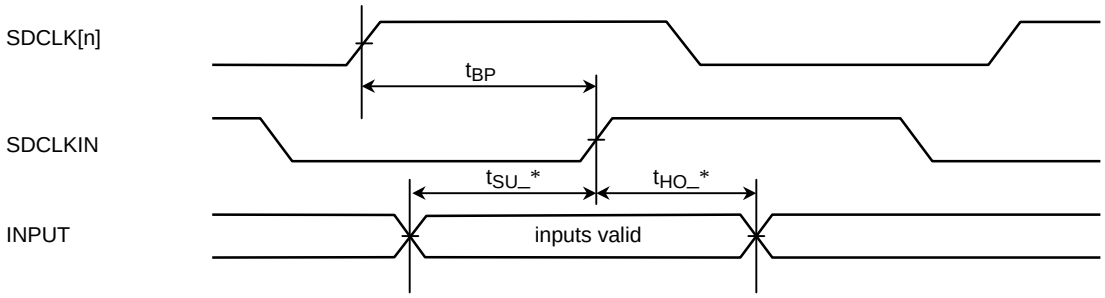


Figure 22.5.4 Timing Diagrams: When Non Bypass Mode Input Signals (SDCLK Basis)

22.5.4 External Bus Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
SYSCLK Cycle Time	t_{CYC_SYSCLK}		12.5	—	ns
SYSCLK HighTime	t_{HIGH_SYSCLK}		4	—	ns
SYSCLK LowTime	t_{LOW_SYSCLK}		4	—	ns
ADDR[19:5] Output delay	t_{VAL_ADDR2}		1.5	9.5	ns
CE[5:0]* Output delay	t_{VAL_CE}		1.5	9.0	ns
OE* Output delay	t_{VAL_OE}		1.5	9.0	ns
SWE* Output delay	t_{VAL_SWE}		1.5	9.0	ns
BWE[3:0]* Output delay	t_{VAL_BWE}		1.5	9.0	ns
UAE Output delay	t_{VAL_UAE}		1.5	9.0	ns
BUSSPRT* Output delay	t_{VAL_DQM}		1.5	9.0	ns
DATA[31:0] Output delay (H->L, L->H)	t_{VAL_BUS}		1.5	9.0	ns
DATA[31:0] Output delay (High-Z->Valid)	$t_{VAL\_DATA2ZV}$		1.5	9.0	ns
DATA[31:0] Output delay (Valid->High-Z)	$t_{VAL\_DATA2VZ}$		1.5	9.0	ns
DATA[31:0] Input set-up time	t_{SU_DATA2}		6.0	—	ns
DATA[31:0] Input set-up time	t_{HO_DATA2}		1.0	—	ns
ACK* Output delay (H->L, L->H)	t_{VAL_ACK}		1.5	9.0	ns
ACK* Output delay (High-Z->Valid)	t_{VAL_ACKZV}		1.5	9.0	ns
ACK* Output delay (Valid->High-Z)	t_{VAL_ACKVZ}		1.5	9.0	ns
ACK* Input set-up time	t_{SU_ACK}		6.0	—	ns
ACK* Input hold time	t_{HO_ACK}		0.5	—	ns

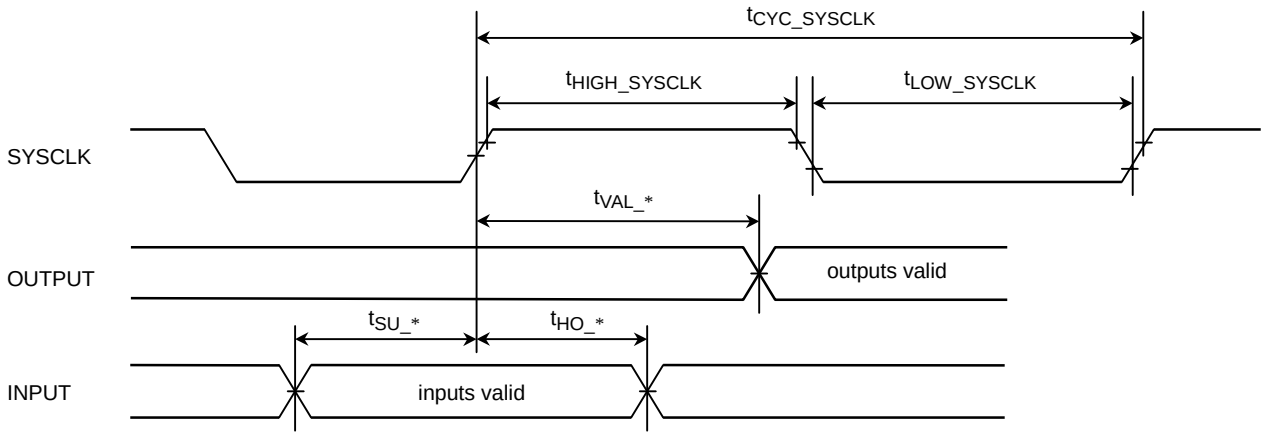


Figure 22.5.5 Timing Diagrams: External Bus Interface

22.5.5 PCI Interface AC Characteristics (33 MHz)

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
PCICLKIO Cycle time (33 MHz)	t_{CYC33}		30	40	ns
PCICLKIO High time (33 MHz)	t_{HIGH33}		11	—	ns
PCICLKIO Low time (33 MHz)	t_{LOW33}		11	—	ns
PCICLKIO Through rate (33 MHz)	t_{SLEW33}		1	4	V/ns
PCICLK[2:1],PCICLKIO Cycle time (33 MHz)	t_{CYC33}	C_L=70 pF	30	40	ns
PCICLK[2:1],PCICLKIO High time (33 MHz)	t_{HIGH33}	C_L=70 pF	11	—	ns
PCICLK[2:1],PCICLKIO Low time (33 MHz)	t_{LOW33}	C_L=70 pF	11	—	ns
PCICLK[2:1],PCICLKIO Skew (33 MHz)	t_{SKEW}	C_L=70 pF, point to point connection	0	T.B.D	ns
PCI output signal (*1) Output delay	t_{VAL33}	C_L=70 pF	2	11	ns
PCI input signal (*2) Input set-up time	t_{SU33}		8	—	ns
PCI input signal (*2) Input set-up time	t_{HO33}		0.5	—	ns
ID_SEL, REQ[0]*, GNT[3:0]* Output delay	$t_{VALPP33}$	C_L=70 pF, point to point connection	2	12	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input set-up time	t_{SUPP33}	point to point connection	10	—	ns
ID_SEL, REQ[3:0]*, GNT[0]* Input hold time	t_{HOPP33}	point to point connection	0	—	ns

1: PCIAD[31:0],C_BE[3:0],PAR,FRAME,IRDY*,TRDY*,STOP*,DEVSEL*,PERR*,SERR*,
2: PCIAD[31:0],C_BE[3:0],PAR,FRAME,IRDY*,TRDY*,STOP*,DEVSEL*,PERR*,SERR*,ID_SEL

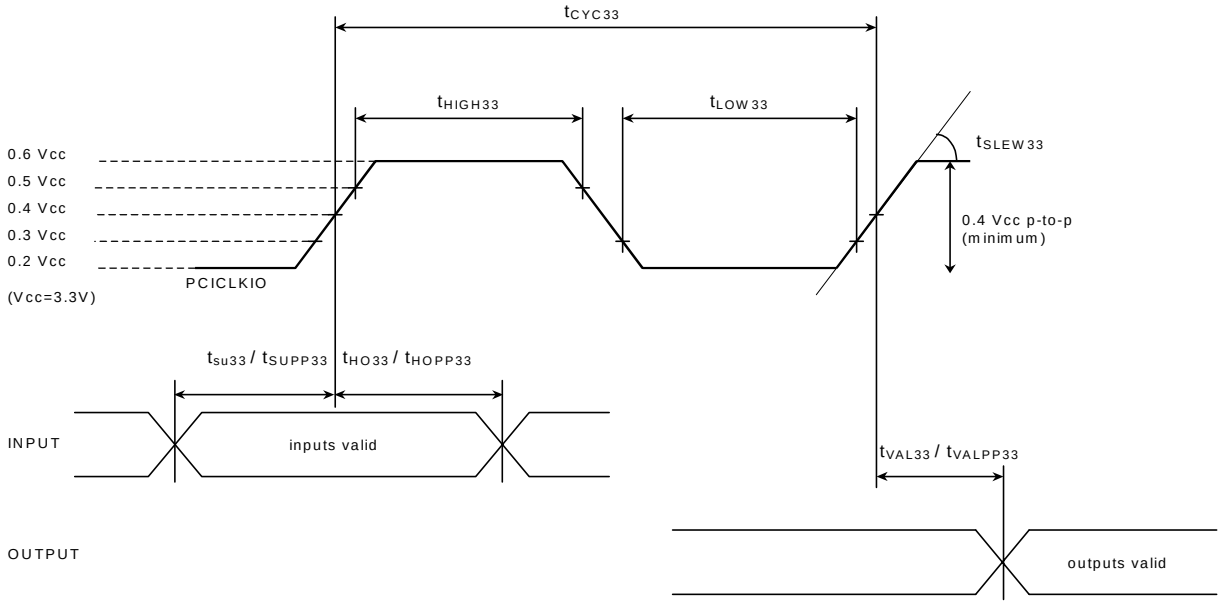


Figure 22.5.6 Timing Diagrams: PCI Interface (3.3V)

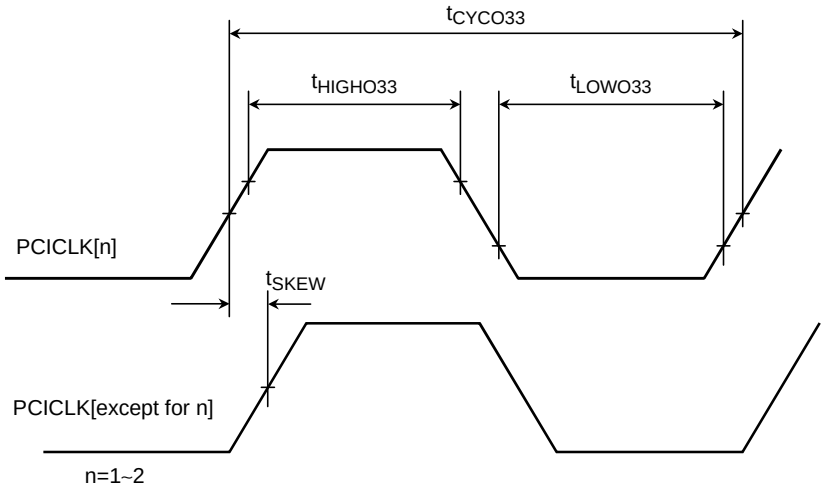


Figure 22.5.7 Timing Diagrams: PCI Clock Skew

22.5.6 DMA Interface AC Characteristics

(Tc = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.3 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Parameter	Symbol	Rating	Min.	Max.	Unit
DMADONE* Delay	t_{VAL_DONE}	C_L=50 pF SYSCLK (C_L=50 pF) Standard	—	5.5	ns
DMADONE* Input pulse width time	t_{PW_DONE}		1/4 × t_{MCP} × 1.1	—	ns

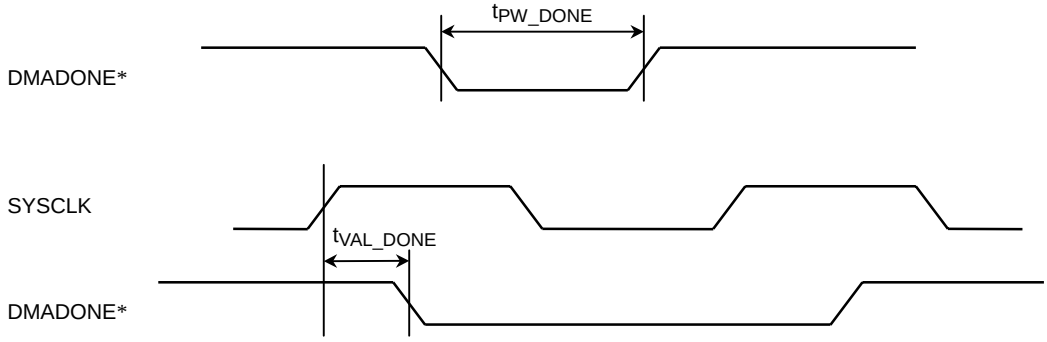


Figure 22.5.8 Timing Diagrams: DMA Interface

22.5.7 Interrupt Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
INT Input pulse width time	t_{PW_INT}		$1/2 \times t_{MCP} \times 1.1$	—	ns
NMI Input pulse width time	t_{PW_NMI}		$1/2 \times t_{MCP} \times 1.1$	—	ns

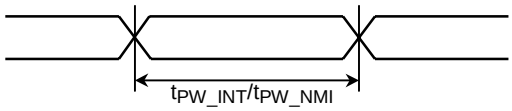


Figure 22.5.9 Timing Diagrams: INT/NMI Interface

22.5.8 SIO Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
SCLK Cycle time	f_{CYC_SCLK}		$t_{MCP} \times 1.1$	—	ns
SCLK Frequency	f_{SCLK}		—	$2 \times f_{MCK} \times 0.45$	MHz
SCLK High time	t_{HIGH_SCLK}		$1/2 \times t_{MCP} \times 1.1$	—	ns
SCLK Low time	t_{LOW_SCLK}		$1/2 \times t_{MCP} \times 1.1$	—	ns

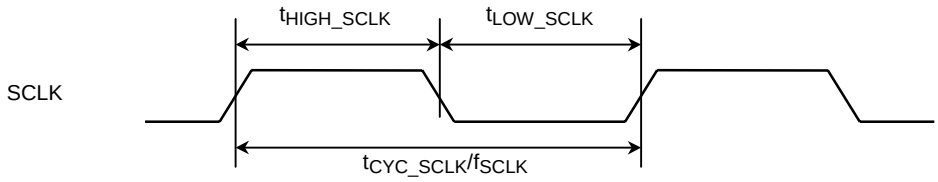


Figure 22.5.10 Timing Diagrams: SIO Interface

22.5.9 Timer Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
TCLK Cycle time	f_{CYC_TCLK}		$t_{MCP} \times 1.1$	—	ns
TCLK Frequency	f_{TCLK}		—	$2 \times f_{MCK} \times 0.45$	MHz
TCLK High time	t_{HIGH_TCLK}		$1/2 \times t_{MCP} \times 1.1$	—	ns
TCLK Low time	t_{LOW_TCLK}		$1/2 \times t_{MCP} \times 1.1$	—	ns

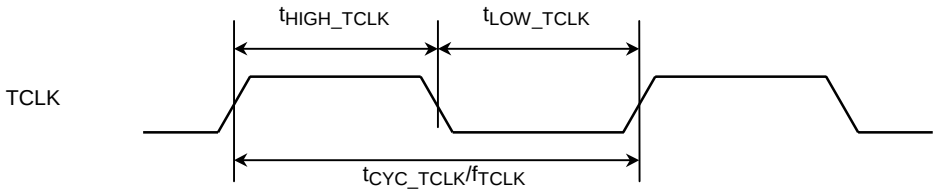


Figure 22.5.11 Timing Diagrams: Timer Interface

22.5.10 PIO Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
PIO[31:0] Output Delay time	t_{VAL_PIO}	IMBUSCLK Standard (C_L=50 pF)	—	9.5	ns
PIO[31:0] Input Setup time	t_{SU_PIO}	IMBUSCLK Standard	8.5	—	ns
PIO[31:0] Input Hold time	t_{HO_PIO}	IMBUSCLK Standard	0	—	ns

Note: The IMBUSCLK is an internal signal. For details, please refer to “Chapter 6 Clocks” in the TMPR4925 Data book.

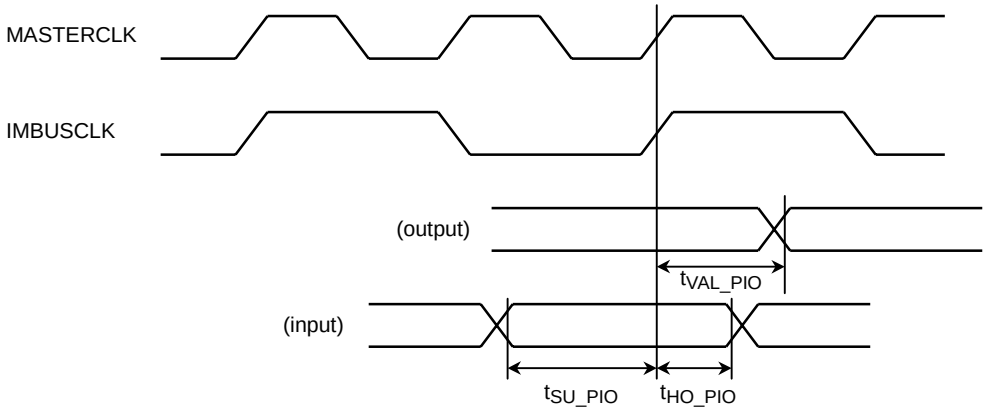


Figure 22.5.12 Timing Diagrams: PIO Interface

22.5.11 AC-link Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
BITCLK High time	t_{HIGH_BCLK}		36	45	ns
BITCLK Low time	t_{LOW_BCLK}		36	45	ns
SYNC Output Delay time	t_{VAL_SYNC}	BITCLK Standard, CL=55 pF	—	15	ns
SDOUT Output Delay time	t_{VAL_SDOUT}	BITCLK Standard, CL=55 pF	—	15	ns
SDIN[1:0] Input Setup time	t_{SU_SDIN}	BITCLK Standard	10	—	ns
SDIN[1:0] Input Hold time	t_{HO_SDIN}	BITCLK Standard	10	—	ns

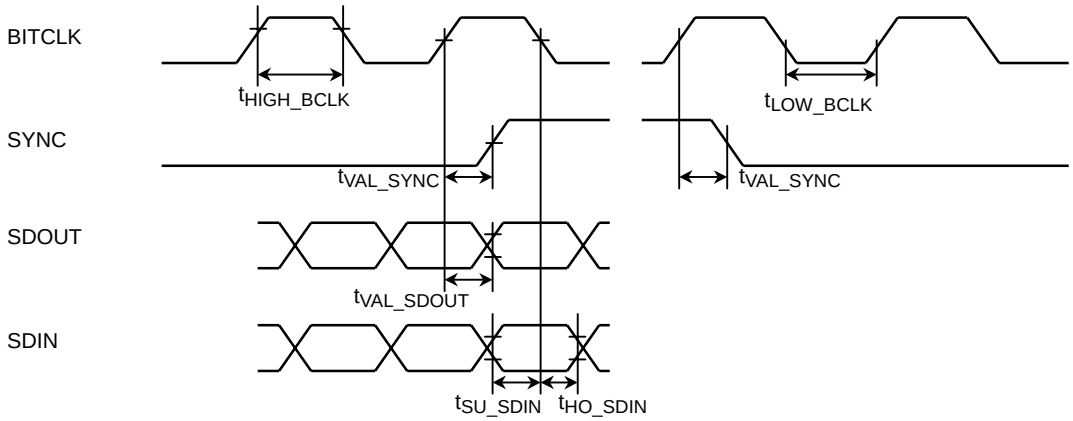


Figure 22.5.13 Timing Diagrams: AC-link Interface

22.5.12 NAND Flash Memory Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
ND_ALE Output Delay time	t_{VAL_NDALE}		—	9.5	ns
ND_CLE Output Delay time	t_{VAL_NDCLE}		—	9.5	ns
ND_CE* Output Delay time	t_{VAL_NDCE}		—	9.5	ns
ND_RE* Output Delay time	t_{VAL_NDRE}		—	9.5	ns
ND_WE* Output Delay time	t_{VAL_NDWE}		—	9.5	ns
DATA[7:0] Read Setup time	t_{SU_RDATA}		8.5	—	ns
DATA[7:0] Read Hold time	t_{HO_RDATA}		1.0	—	ns
DATA[7:0] Write Delay time	t_{VAL_WDATA}		—	9.5	ns
DATA[7:0] Write Hold time	t_{HO_WDATA}		1.0	—	ns

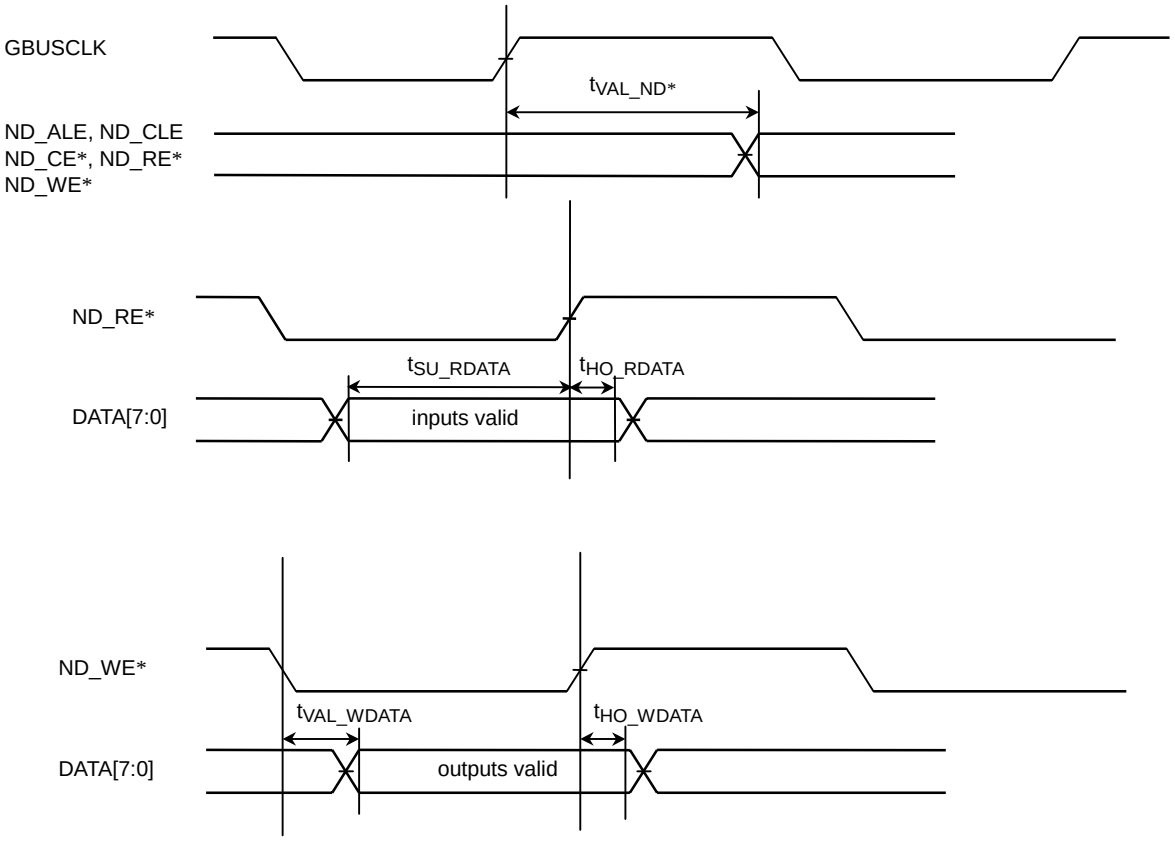


Figure 22.5.14 Timing Diagrams: NAND Flash Memory Interface

22.5.13 CHI Interface AC Characteristics

(Tc = 0 ~ 70°C, V_{CCIO} = 3.3 V ± 0.3 V, V_{CCInt} = 1.5 V ± 0.1 V, V_{SS} = 0 V)

Parameter	Symbol	Rating	Min.	Max.	Unit
CHICLK Cycle time	f_{CYC_CHICLK}		225	—	ns
CHICLK High time	t_{HIGH_CHICLK}		100	—	ns
CHICLK Low time	t_{LOW_CHICLK}		100	—	ns

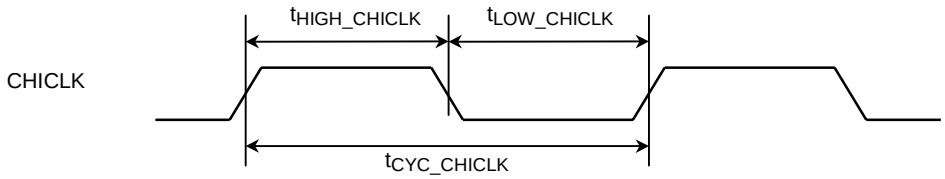


Figure 22.5.15 Timing Diagrams: CHI Interface

22.5.14 SPI Interface AC Characteristics

($Tc = 0 \sim 70°C$, $V_{CCIO} = 3.3\ V \pm 0.3\ V$, $V_{CCInt} = 1.5\ V \pm 0.1\ V$, $V_{SS} = 0\ V$)

Parameter	Symbol	Rating	Min.	Max.	Unit
SPICLK Cycle time	f_{CYC_SPICLK}		$2\times t_{MCP}$	—	ns
SPICLK Frequency	f_{SPICLK}		—	$1/2 \times f_{MCK}$	MHz
SPICLK High time	t_{HIGH_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPICLK Low time	t_{LOW_SPICLK}		$t_{MCP} \times 0.9$	—	ns
SPIOUT Output Delay time	t_{VAL_SPIOUT}		—	9.5	ns
SPIIN Input Setup time	t_{SU_SPIIN}		8.5	—	ns
SPIIN Input Hold time	t_{HO_SPIIN}		1.0	—	ns

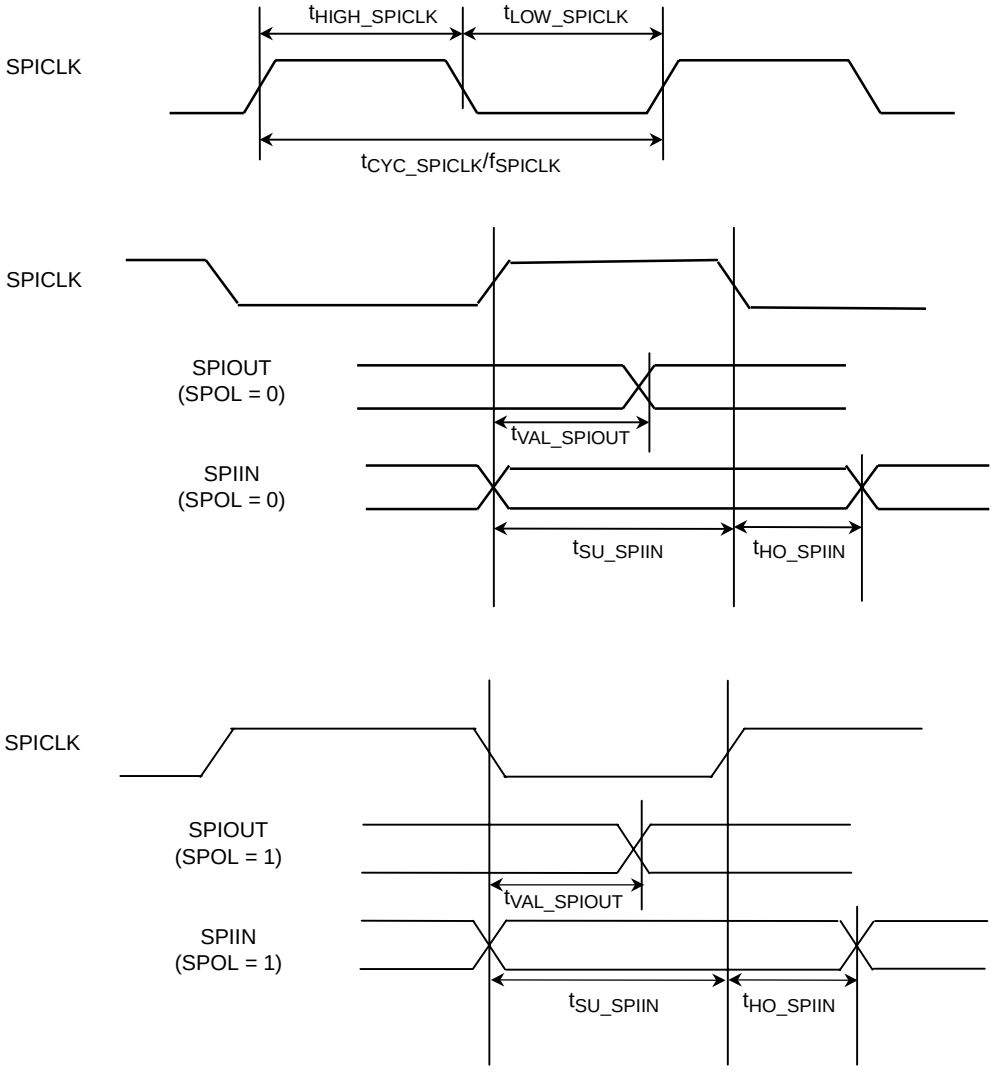


Figure 22.5.16 Timing Diagrams: SPI Interface

23. Pin Layout, Package

23.1 Pin Layout

Table 23.1.1 shows pin layout, Table 23.1.2 shows pin designations (Coordinates), Table 23.1.3 shows pin designations (Signal name).

	A	B	C	D	E	F	G	H	J	K
20	RAS*	SDCS[0]*	DQM[1]	DQM[0]	RP*	DATA[15]	DATA[29]	DATA[28]	DATA[27]	DATA[26]
19	SDCS[1]*	DQM[3]	DQM[2]	CAS*	DATA[31]	DATA[30]	DATA[13]	DATA[12]	DATA[11]	DATA[10]
18	ADDR[6]	ADDR[5]	VDDS	WE*	VDDS	DATA[14]	VDDC	VDDS	VDDC	VDDC
17	ADDR[8]	ADDR[7]	Vss	Vss	Vss	Vss	Vss	Vss	Vss	Vss
16	ADDR[10]	ADDR[9]	VDDC	VDDS						
15	ADDR[13]	ADDR[12]	ADDR[11]	Vss						
14	SADDR10	ADDR[14]	VDDC	Vss						
13	ADDR[18]	ADDR[19]	ADDR[16]	VDDS						
12	SDCS[2]*	CKE	ADDR[17]	Vss						
11	SDCLK[0]	SDCS[3]*	SCANENB*	VDDS						
10	SDCLK[1]	RESET*	VDDC	Vss						
9	SDCLKIN	PIO[20]	TRST*	PON*						
8	PIO[19]	PIO[18]	PIO[23]	VDDS						
7	PIO[22]	PIO[21]	VDDC	Vss						
6	PIO[27]	PIO[29]	PIO[28]	PIO[30]						
5	PIO[24]	PIO[31]	PIO[25]	VDDC	TOP View					
4	PIO[26]	TDO	VDDC	Vss	REQ[2]*	VDDS	Vss	VDDS	Vss	VDDS
3	TDI	TCK	VDDS	GNT[1]*	GNT[3]*	PCIAD[30]	VDDC	PCIAD[25]	ID_SEL	PCIAD[21]
2	TMS	GNT[0]*	REQ[0]*	REQ[1]*	REQ[3]*	PCIAD[29]	PCIAD[27]	PCIAD[24]	PCIAD[23]	PCIAD[20]
1	PCICLKIO	PCICLK[1]	PCICLK[2]	GNT[2]*	PCIAD[31]	PCIAD[28]	PCIAD[26]	C_BE[3]	PCIAD[22]	PCIAD[19]

Table 23.1.1 Pin layout (1/2)

L	M	N	P	R	T	U	V	W	Y	
DATA[25]	DATA[8]	DATA[22]	DATA[21]	DATA[20]	DATA[3]	DATA[17]	Vss	MASTERCLK	Vss	20
DATA[9]	DATA[23]	DATA[6]	DATA[5]	DATA[4]	DATA[18]	DATA[1]	DATA[0]	PLLVDD	PLLVSS	19
DATA[24]	DATA[7]	VDDS	VDDC	DATA[19]	DATA[2]	DATA[16]	VDDS	C32KOUT	C32KIN	18
VDDS	Vss	Vss	Vss	Vss	VDDS	Vss	TEST*	NMI*	BC32K	17
						Vss	PIO[16]	PIO[15]	PIO[14]	16
						PIO[7]	PIO[13]	PIO[17]	PIO[12]	15
						Vss	VDDC	PIO[9]	PIO[8]	14
						PIO[6]	PIO[5]	PIO[10]	PIO[11]	13
						VDDS	PIO[1]	PIO[3]	ACK*	12
						Vss	VDDC	PIO[4]	BUSSPRT	11
						PIO[0]	PIO[2]	CE[1]*	CE[0]*	10
						CE[3]*	CE[2]*	ADDR[15]	OE*	9
						VDDS	ADDR[2]	ADDR[3]	ADDR[4]	8
						Vss	VDDC	ADDR[0]	ADDR[1]	7
						BWE [2]*	BWE [3]*	UAE	SWE*	6
					TOP View	VDDS	BWE [0]*	BWE [1]*	SYSCLK	5
Vss	VDDS	IRDY*	Vss	VDDC	VDDS	Vss	VDDC	PCIAD[7]	PCIAD[3]	4
VDDC	PCIAD[16]	TRDY*	VDDC	PAR	PCIAD[14]	PCIAD[11]	VDDS	PCIAD[6]	PCIAD[2]	3
PCIAD[18]	C_BE[2]	DEVSEL*	PERR*	C_BE[1]	PCIAD[13]	PCIAD[10]	PCIAD[8]	PCIAD[5]	PCIAD[1]	2
PCIAD[17]	FRAME*	STOP*	SERR*	PCIAD[15]	PCIAD[12]	PCIAD[9]	C_BE[0]	PCIAD[4]	PCIAD[0]	1

Table 23.1.1 Pin layout (2/2)

Table 23.1.2 Pin Designations (Coordinates)

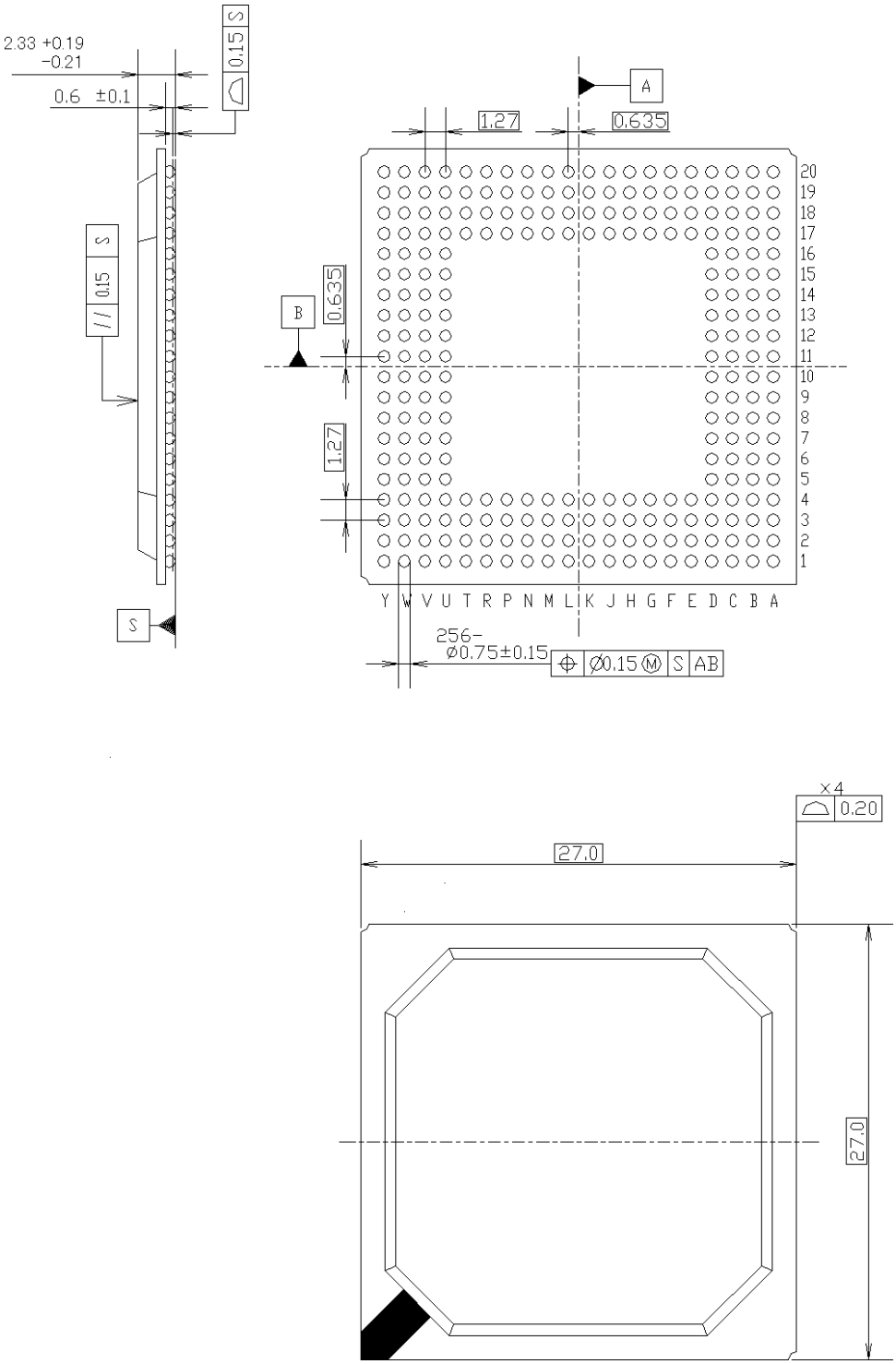
A1	PCICLKIO	C13	ADDR[16]	H1	C_BE[3]	P17	Vss	V13	PIO[5]
A2	TMS	C14	VDDC	H2	PCIAD[24]	P18	VDDC	V14	VDDC
A3	TDI	C15	ADDR[11]	H3	PCIAD[25]	P19	DATA[5]	V15	PIO[13]
A4	PIO[26]	C16	VDDC	H4	VDDS	P20	DATA[21]	V16	PIO[16]
A5	PIO[24]	C17	Vss	H17	Vss	R1	PCIAD[15]	V17	TEST*
A6	PIO[27]	C18	VDDS	H18	VDDS	R2	C_BE[1]	V18	VDDS
A7	PIO[22]	C19	DQM[2]	H19	DATA[12]	R3	PAR	V19	DATA[0]
A8	PIO[19]	C20	DQM[1]	H20	DATA[28]	R4	VDDC	V20	Vss
A9	SDCLKIN	D1	GNT[2]*	J1	PCIAD[22]	R17	Vss	W1	PCIAD[4]
A10	SDCLK[1]	D2	REQ[1]*	J2	PCIAD[23]	R18	DATA[19]	W2	PCIAD[5]
A11	SDCLK[0]	D3	GNT[1]*	J3	ID_SEL	R19	DATA[4]	W3	PCIAD[6]
A12	SDCS[2]*	D4	Vss	J4	Vss	R20	DATA[20]	W4	PCIAD[7]
A13	ADDR[18]	D5	VDDC	J17	Vss	T1	PCIAD[12]	W5	BWE[1]*
A14	SADDR10	D6	PIO[30]	J18	VDDC	T2	PCIAD[13]	W6	UAE
A15	ADDR[13]	D7	Vss	J19	DATA[11]	T3	PCIAD[14]	W7	ADDR[0]
A16	ADDR[10]	D8	VDDS	J20	DATA[27]	T4	VDDS	W8	ADDR[3]
A17	ADDR[8]	D9	PON*	K1	PCIAD[19]	T17	VDDS	W9	ADDR[15]
A18	ADDR[6]	D10	Vss	K2	PCIAD[20]	T18	DATA[2]	W10	CE[1]*
A19	SDCS[1]*	D11	VDDS	K3	PCIAD[21]	T19	DATA[18]	W11	PIO[4]
A20	RAS*	D12	Vss	K4	VDDS	T20	DATA[3]	W12	PIO[3]
B1	PCICLK[1]	D13	VDDS	K17	Vss	U1	PCIAD[9]	W13	PIO[10]
B2	GNT[0]*	D14	Vss	K18	VDDC	U2	PCIAD[10]	W14	PIO[9]
B3	TCK	D15	Vss	K19	DATA[10]	U3	PCIAD[11]	W15	PIO[17]
B4	TDO	D16	VDDS	K20	DATA[26]	U4	Vss	W16	PIO[15]
B5	PIO[31]	D17	Vss	L1	PCIAD[17]	U5	VDDS	W17	NMI*
B6	PIO[29]	D18	WE*	L2	PCIAD[18]	U6	BWE[2]*	W18	C32KOUT
B7	PIO[21]	D19	CAS*	L3	VDDC	U7	Vss	W19	PLLVDD
B8	PIO[18]	D20	DQM[0]	L4	Vss	U8	VDDS	W20	MASTERCLK
B9	PIO[20]	E1	PCIAD[31]	L17	VDDS	U9	CE[3]*	Y1	PCIAD[0]
B10	RESET*	E2	REQ[3]*	L18	DATA[24]	U10	PIO[0]	Y2	PCIAD[1]
B11	SDCS[3]*	E3	GNT[3]*	L19	DATA[9]	U11	Vss	Y3	PCIAD[2]
B12	CKE	E4	REQ[2]*	L20	DATA[25]	U12	VDDS	Y4	PCIAD[3]
B13	ADDR[19]	E17	Vss	M1	FRAME*	U13	PIO[6]	Y5	SYSCLK
B14	ADDR[14]	E18	VDDS	M2	C_BE[2]	U14	Vss	Y6	SWE*
B15	ADDR[12]	E19	DATA[31]	M3	PCIAD[16]	U15	PIO[7]	Y7	ADDR[1]
B16	ADDR[9]	E20	RP*	M4	VDDS	U16	Vss	Y8	ADDR[4]
B17	ADDR[7]	F1	PCIAD[28]	M17	Vss	U17	Vss	Y9	OE*
B18	ADDR[5]	F2	PCIAD[29]	M18	DATA[7]	U18	DATA[16]	Y10	CE[0]*
B19	DQM[3]	F3	PCIAD[30]	M19	DATA[23]	U19	DATA[1]	Y11	BUSSPRT
B20	SDCS[0]*	F4	VDDS	M20	DATA[8]	U20	DATA[17]	Y12	ACK*
C1	PCICLK[2]	F17	Vss	N1	STOP*	V1	C_BE[0]	Y13	PIO[11]
C2	REQ[0]*	F18	DATA[14]	N2	DEVSEL*	V2	PCIAD[8]	Y14	PIO[8]
C3	VDDS	F19	DATA[30]	N3	TRDY*	V3	VDDS	Y15	PIO[12]
C4	VDDC	F20	DATA[15]	N4	IRDY*	V4	VDDC	Y16	PIO[14]
C5	PIO[25]	G1	PCIAD[26]	N17	Vss	V5	BWE[0]*	Y17	BC32K
C6	PIO[28]	G2	PCIAD[27]	N18	VDDS	V6	BWE[3]*	Y18	C32KIN
C7	VDDC	G3	VDDC	N19	DATA[6]	V7	VDDC	Y19	PLLVSS
C8	PIO[23]	G4	Vss	N20	DATA[22]	V8	ADDR[2]	Y20	Vss
C9	TRST*	G17	Vss	P1	SERR*	V9	CE[2]*		
C10	VDDC	G18	VDDC	P2	PERR*	V10	PIO[2]		
C11	SCANENB*	G19	DATA[13]	P3	VDDC	V11	VDDC		
C12	ADDR [17]	G20	DATA[29]	P4	Vss	V12	PIO[1]		

Table 23.1.3 Pin Designations (Signal name)

Y12	ACK*	G19	DATA[13]	L1	PCIAD[17]	Y19	PLLVSS	D13	VDDS
W7	ADDR[0]	F18	DATA[14]	L2	PCIAD[18]	D9	PON*	D16	VDDS
Y7	ADDR[1]	F20	DATA[15]	K1	PCIAD[19]	A20	RAS*	E18	VDDS
V8	ADDR[2]	U18	DATA[16]	K2	PCIAD[20]	C2	REQ[0]*	F4	VDDS
W8	ADDR[3]	U20	DATA[17]	K3	PCIAD[21]	D2	REQ[1]*	H4	VDDS
Y8	ADDR[4]	T19	DATA[18]	J1	PCIAD[22]	E4	REQ[2]*	H18	VDDS
B18	ADDR[5]	R18	DATA[19]	J2	PCIAD[23]	E2	REQ[3]*	K4	VDDS
A18	ADDR[6]	R20	DATA[20]	H2	PCIAD[24]	B10	RESET*	L17	VDDS
B17	ADDR[7]	P20	DATA[21]	H3	PCIAD[25]	E20	RP*	M4	VDDS
A17	ADDR[8]	N20	DATA[22]	G1	PCIAD[26]	A14	SADDR10	N18	VDDS
B16	ADDR[9]	M19	DATA[23]	G2	PCIAD[27]	C11	SCANENB*	T4	VDDS
A16	ADDR[10]	L18	DATA[24]	F1	PCIAD[28]	A11	SDCLK[0]	T17	VDDS
C15	ADDR[11]	L20	DATA[25]	F2	PCIAD[29]	A10	SDCLK[1]	U5	VDDS
B15	ADDR[12]	K20	DATA[26]	F3	PCIAD[30]	A9	SDCLKIN	U8	VDDS
A15	ADDR[13]	J20	DATA[27]	E1	PCIAD[31]	B20	SDCS[0]*	U12	VDDS
B14	ADDR[14]	H20	DATA[28]	B1	PCICLK[1]	A19	SDCS[1]*	V3	VDDS
W9	ADDR[15]	G20	DATA[29]	C1	PCICLK[2]	A12	SDCS[2]*	V18	VDDS
C13	ADDR[16]	F19	DATA[30]	A1	PCICLKIO	B11	SDCS[3]*	C17	Vss
C12	ADDR[17]	E19	DATA[31]	P2	PERR*	P1	SERR*	D4	Vss
A13	ADDR[18]	N2	DEVSEL*	U10	PIO[0]	N1	STOP*	D7	Vss
B13	ADDR[19]	D20	DQM[0]	V12	PIO[1]	Y6	SWE*	D10	Vss
Y17	BC32K	C20	DQM[1]	V10	PIO[2]	Y5	SYSCLK	D12	Vss
Y11	BUSSPRT	C19	DQM[2]	W12	PIO[3]	B3	TCK	D14	Vss
V5	BWE[0]*	B19	DQM[3]	W11	PIO[4]	A3	TDI	D15	Vss
W5	BWE[1]*	M1	FRAME*	V13	PIO[5]	B4	TDO	D17	Vss
U6	BWE[2]*	B2	GNT[0]*	U13	PIO[6]	V17	TEST*	E17	Vss
V6	BWE[3]*	D3	GNT[1]*	U15	PIO[7]	A2	TMS	F17	Vss
D19	CAS*	D1	GNT[2]*	Y14	PIO[8]	N3	TRDY*	G4	Vss
Y10	CE[0]*	E3	GNT[3]*	W14	PIO[9]	C9	TRST*	G17	Vss
W10	CE[1]*	J3	ID_SEL	W13	PIO[10]	W6	UAE	H17	Vss
V9	CE[2]*	N4	IRDY*	Y13	PIO[11]	C4	VDDC	J4	Vss
U9	CE[3]*	W20	MASTERCLK	Y15	PIO[12]	C7	VDDC	J17	Vss
B12	CKE	W17	NMI*	V15	PIO[13]	C10	VDDC	K17	Vss
Y18	C32KIN	Y9	OE*	Y16	PIO[14]	C14	VDDC	L4	Vss
W18	C32KOUT	R3	PAR	W16	PIO[15]	C16	VDDC	M17	Vss
V1	C_BE[0]	Y1	PCIAD[0]	V16	PIO[16]	D5	VDDC	N17	Vss
R2	C_BE[1]	Y2	PCIAD[1]	W15	PIO[17]	G3	VDDC	P4	Vss
M2	C_BE[2]	Y3	PCIAD[2]	B8	PIO[18]	G18	VDDC	P17	Vss
H1	C_BE[3]	Y4	PCIAD[3]	A8	PIO[19]	J18	VDDC	R17	Vss
V19	DATA[0]	W1	PCIAD[4]	B9	PIO[20]	K18	VDDC	U4	Vss
U19	DATA[1]	W2	PCIAD[5]	B7	PIO[21]	L3	VDDC	U7	Vss
T18	DATA[2]	W3	PCIAD[6]	A7	PIO[22]	P3	VDDC	U11	Vss
T20	DATA[3]	W4	PCIAD[7]	C8	PIO[23]	P18	VDDC	U14	Vss
R19	DATA[4]	V2	PCIAD[8]	A5	PIO[24]	R4	VDDC	U16	Vss
P19	DATA[5]	U1	PCIAD[9]	C5	PIO[25]	V4	VDDC	U17	Vss
N19	DATA[6]	U2	PCIAD[10]	A4	PIO[26]	V7	VDDC	V20	Vss
M18	DATA[7]	U3	PCIAD[11]	A6	PIO[27]	V11	VDDC	Y20	Vss
M20	DATA[8]	T1	PCIAD[12]	C6	PIO[28]	V14	VDDC	D18	Vss
L19	DATA[9]	T2	PCIAD[13]	B6	PIO[29]	C3	VDDS		
K19	DATA[10]	T3	PCIAD[14]	D6	PIO[30]	C18	VDDS		
J19	DATA[11]	R1	PCIAD[15]	B5	PIO[31]	D8	VDDS		
H19	DATA[12]	M3	PCIAD[16]	W19	PLLVDD	D11	VDDS		

23.2 Package

Package Type (Package Code) : 256-pin PBGA / PBGA[4L] (P-BGA256-2727-1.27A4)



24. Usage Notes

24.1 Limitation on DMA Data Chaining

- Overview

  The DMA Controller works incorrectly if the DMCCRn.IMMCHN bit is cleared and the address increment value (DMSAIRn/DMDAIRn) is negative. The DMA Controller might also work incorrectly regardless of the setting of the DMCCRn.IMMCHN bit if a dynamic or typical DMA chaining is performed after completion of a transfer with a negative address increment.

- Symptom

  The DMA Controller might read wrong data or write data to wrong addresses.

- Workaround Measure

  Keep the DMCCRn.IMMCHN bit set. Make sure the DMSAIRn and DMDAIRn values for the last descriptor are equal to or greater than 0 before adding the next command descriptor chain for dynamic DMA chaining. For DMA data chaining with a negative address increment, write 0 or a positive number once, then the desired negative increment value to the DMSAIRn/DMDAIRn.

24.2 Limitation on a Register Read After an SIO Software Reset

- Overview

  Immediately reading a register after a software reset (seting “1” to bit 15 of the SIFCRx Register) results in the SIO not responding and the bus locking up.

- Symptom

  After a software reset (setting “1” to bit 15 of the SIFCRx Register), IMBUSCLK resets SIO for four clock cycles. During this reset, the SIO does not recognize that it has been accessed even when it has been accessed. The bus locks up since the device that accesses the SIO waits for a response from the SIO. A bus error will then be issued if timeout errors have been enabled.

- Workaround Measure

  After the software reset, perform a dummy read operation to the register of a different module on the IM-Bus, then access the SIO register. Performing the dummy access on the IM-Bus requires eight IMBUSCLK cycles to complete, making it possible to avoid the problem since this is longer than the amount of time the SIO reset requires.

24.3 Other Precautions

- Always set bit 4 in the G-Bus Arbiter Control register (GARBC) to 1. Clearing this bit might cause arbitration deadlocks.
- There are restrictions for address increment values regarding DMA burst transfers. For a detailed description, see Section 8.3.7, “Single Address Transfer,” and Section 8.3.8, “Dual Address Transfer.”
- The four target address spaces (MEM0, MEM1, MEM2 and IO) of the PCI Controller may not overlap. See Section 10.3.5, “Target Access.”
- It is recommended to set the G2P Timeout register (GPTOCNT) to 0. See Section 10.4.12, “G2P Timeout Count Register.”

Appendix A. TX49/H2 Core Supplement

This section explains items that are unique to the TX4925 of the TX49/H2 Core. Please refer to the “64-bit TX System RISC TX49/H2 Core Architecture” for more information regarding the TX49/H2 Core.

A.1 Processor ID

PRId Register values of the TX4925 TX49/H2 Core are as follows.

Processor Revision Identifier Register: 0x0000_2D23

FPU Implementation/Revision Register (FCR0): 0x0000_2D21

These values may be changed at a later date. Please contact the Toshiba Engineering Department for the most recent information.

A.2 Interrupts

Interrupt signalling of the on-chip interrupt controller is reflected in bit IP[2] of the Cause Register in the TX49/H2 Core. In addition, interrupt causes are reflected in other bits of the IP field. Please refer to Section “15.3.5 Interrupt Notification” for more information.

A.3 Bus Snoop

The Bus Snoop function is not used with the TX4925 due to restrictions of the Bus Snoop specification.

A.4 Halt/Doze Mode

The Doze mode is not necessary when the Bus Snoop function is not used. Please use the Halt mode, which further reduces power consumption. Clearing the HALT bit of the Config Register makes it possible to shift to the Halt mode by executing the WAIT instruction.

A.5 Memory Access Order

The TX49/H2 Core has a 4-stage Write buffer, the PCI Bus Bridge (PCI Controller) has 4 stages for initiator access, and has a 2-stage Post Write buffer (Write buffer) for target access.

When data enters the Write buffer of the TX49/H2 Core, Cache Refill Read operations that do not match the address of that data after the Write is issued may be issued to the internal bus (G-Bus) before the Write. Other accesses are issued in order.

Executing the SYNC instruction guarantees that bus access invoked by a load/store instruction previously executed will be complete on the internal bus.

The PCI Bus Bridge is issued by the issue destination bus in the order all bus accesses are issued on the issue source bus. Please refer to “10.3.6 Post Write Function” for more information regarding methods for guaranteeing the completion of Write transactions of the Post Write Buffer.