TOSHIBA

TX System RISC
TX79 Family
TMPR7901

(Symmetric 2-way superscalar
64-bit CPU)

The information contained herein is subject to change without notice.

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

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

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

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

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Handling Precautions

1. Using Toshiba Semiconductors Safely

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

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

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

2. Safety Precautions

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

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

[Explanation of labels]

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

[Explanation of graphic symbol]

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

2.1 General Precautions regarding Semiconductor Devices

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

2.2 Precautions Specific to Each Product Group

2.2.1 Optical semiconductor devices

⚠DANGER

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

⚠WARNING

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

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

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

2.2.2 Power devices

⚠DANGER

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

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

⚠WARNING

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

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

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

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

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

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

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

⚠CAUTION

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

2.2.3 Bipolar ICs (for use in automobiles)

⚠CAUTION

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

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

3. General Safety Precautions and Usage Considerations

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

3.1 From Incoming to Shipping

3.1.1 Electrostatic discharge (ESD)

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

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

(1) Work environment

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

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

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

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

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

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

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

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

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

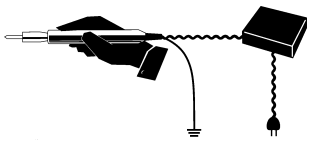
(e) Make sure that sections of the tape carrier which come into contact with installation devices or other electrical machinery are made of a low-resistance material.

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

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

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

(2) Operating environment



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

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

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

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

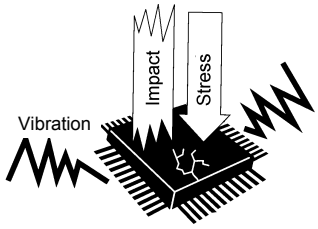
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.

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

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

3.1.2 Vibration, impact and stress

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



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

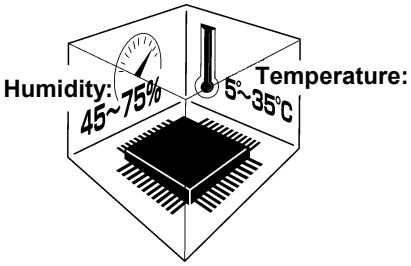
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

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

3.2 Storage

3.2.1 General storage

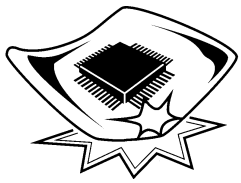
- Avoid storage locations where devices will be exposed to moisture or direct sunlight.
- Follow the instructions printed on the device cartons regarding transportation and storage.
- The storage area temperature should be kept within a temperature range of 5°C to 35°C, and relative humidity should be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



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

3.2.2 Moisture-proof packing

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



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

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

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

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

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

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

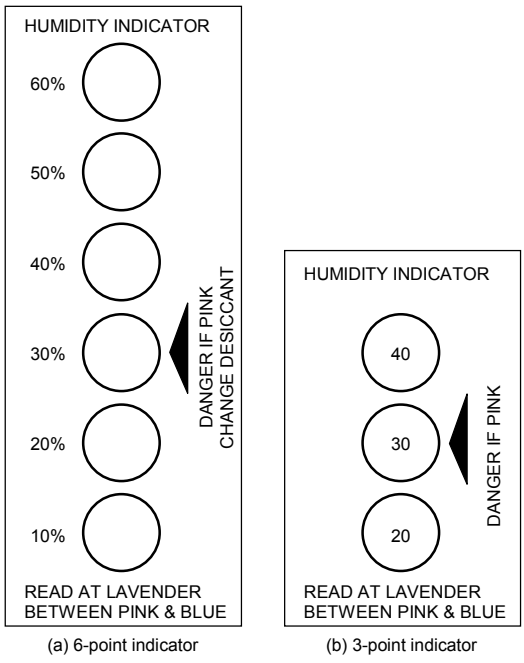


Figure 1 Humidity indicator

3.3 Design

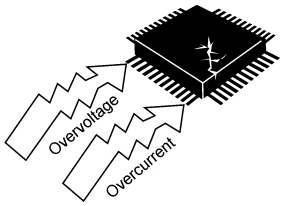
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will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

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

3.3.1 Absolute maximum ratings

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

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



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

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

3.3.2 Recommended operating conditions

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

3.3.3 Derating

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

3.3.4 Unused pins

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

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

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

3.3.5 Latch-up

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

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

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

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

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

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

3.3.6 Input/Output protection

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

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

3.3.7 Load capacitance

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

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

3.3.8 Thermal design

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

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

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

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

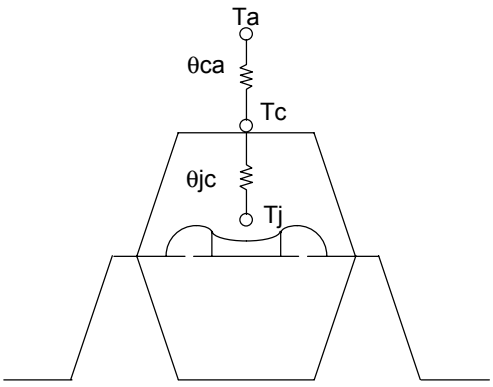


Figure 2 Thermal resistance of package

3.3.9 Interfacing

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

3.3.10 Decoupling

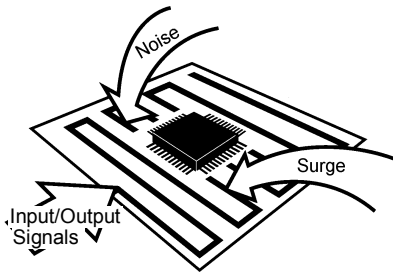
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with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

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

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

3.3.11 External noise

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



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

3.3.12 Electromagnetic interference

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

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

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

the thickness of the power supply wiring patterns on the printed circuit board. One effective method, for example, is to devise several shielding options during design, and then select the most suitable shielding method based on the results of measurements taken after the prototype has been completed.

3.3.13 Peripheral circuits

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

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

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

3.3.14 Safety standards

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

3.3.15 Other precautions

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

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

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

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

3.4 Inspection, Testing and Evaluation

3.4.1 Grounding

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

3.4.2 Inspection Sequence

⚠CAUTION

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

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

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

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

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

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

3.5 Mounting

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

3.5.1 Lead forming

⚠CAUTION

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

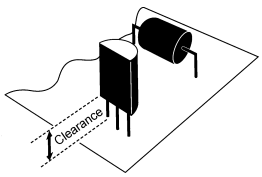
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

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

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

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

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



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

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

3.5.2 Socket mounting

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

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

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

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

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

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

3.5.3 Soldering temperature profile

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

(1) Using a soldering iron

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

(2) Using medium infrared ray reflow

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

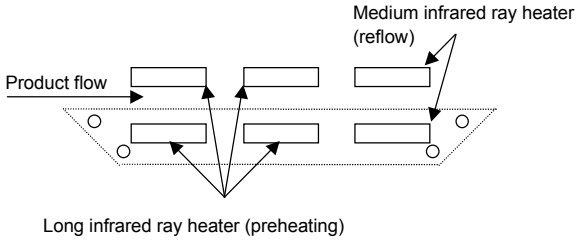


Figure 3 Heating top and bottom with long or medium infrared rays

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

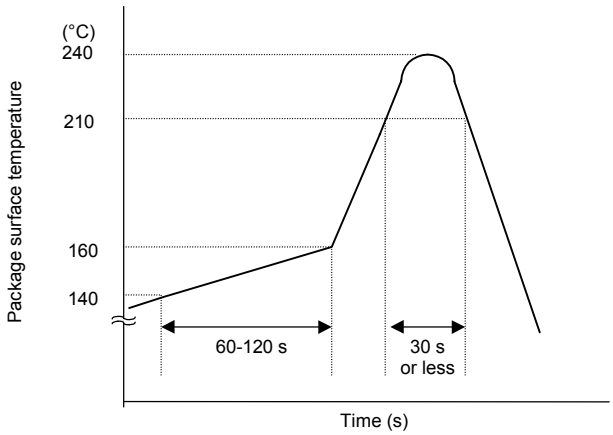


Figure 4 Sample temperature profile for infrared or hot air reflow

(3) Using hot air reflow

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

(4) Using solder flow

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

- For surface-mount packages, complete soldering within 5 seconds at a temperature of 250°C or less in order to prevent thermal stress in the device.
- Figure 5 shows an example of a recommended temperature profile for surface-mount packages using solder flow.

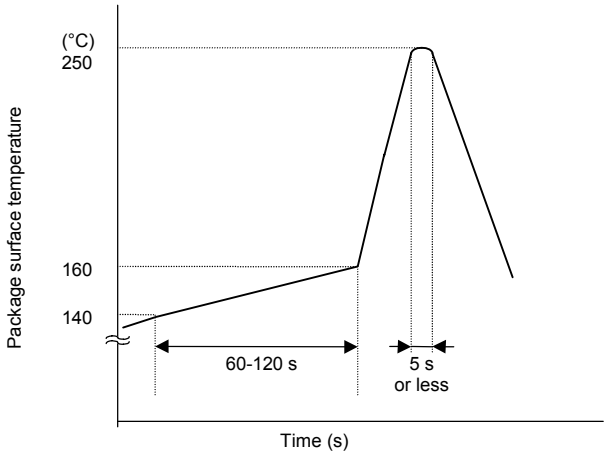


Figure 5 Sample temperature profile for solder flow

3.5.4 Flux cleaning and ultrasonic cleaning

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

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

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

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

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

Frequency: 27 kHz ~ 29 kHz

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

Cleaning time: 30 seconds or less

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

3.5.5 No cleaning

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

3.5.6 Mounting tape carrier packages (TCPs)

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

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

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

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

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

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

3.5.7 Mounting chips

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

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

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

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

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

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

3.5.8 Circuit board coating

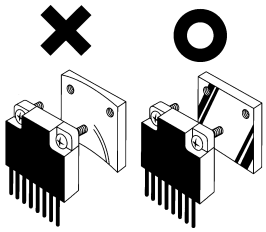
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects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

3.5.9 Heat sinks

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

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

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



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

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

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

3.5.10 Tightening torque

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

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

3.5.11 Repeated device mounting and usage

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

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

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

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

3.6 Protecting Devices in the Field

3.6.1 Temperature

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

3.6.2 Humidity

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

3.6.3 Corrosive gases

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

3.6.4 Radioactive and cosmic rays

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

3.6.5 Strong electrical and magnetic fields

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

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

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

3.6.7 Dust and oil

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

3.6.8 Fire

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

3.7 Disposal of Devices and Packing Materials

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

4. Precautions and Usage Considerations

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

4.1 Microcontrollers

4.1.1 Design

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

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

(2) Undefined functions

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

TX7901 User's Manual

Rev. 6.30T

November, 2001

DOCUMENT NUMBER M–99–00004–07

1. Introduction

1.1 Overview

The TX7901 MIPS RISC microcontroller is a highly integrated solution based on Toshiba's dual-issue super-scalar pipeline Processor Core, the C790 (henceforth referred to as "the C790"). The C790 has a 128-bit internal architecture featuring MIPS ISA support and additional instruction enhancements specially developed for embedded applications.

The TX7901 is a new generation MIPS processor solution offering high performance, high bandwidth and high integration, utilizing Toshiba's Computer-On-Silicon concept. This class of product is targeted for applications that require a high-performance, cost-effective solution such as networking, printers, and set-top boxes.

1.2 Terminology

1.2.1 Abbreviations used

802.3x IEEE 802.3x standard for Ethernet based Local Area Networks

b bits (e.g. 1 Mb = 1 Mega bit)

B Bytes (e.g. 4 MB = 4 Mega Bytes)

Suffix for active low signals

BIU Bus Interface Unit

BTAC Branch Target Address Cache

COP0 Coprocessor 0

C790 High-performance MIPS CPU Core on which the TX7901 is based

C790 Bus Main system bus that connects the C790 CPU to the rest of the system devices such as the memory controller and G-bus bridge.

D$ Data Cache Memory

DIMM DRAM chip module
Set of DRAM chips that are controlled by a chip selector signal. Double-sided DIMM in this case is considered to be two DIMMs.

DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DRAM	Dynamic RAM
DTLB	Data Translation Look-aside Buffer
F/B	Feedback
FIFO	“First In First Out” Buffer
Fiber PMD (EMAC)	Fiber Physical Medium Dependent
FPU	Floating Point Unit
G-Bus	Proprietary Toshiba On-chip Bus Interface intended for IP interfaces
I$	Instruction Cache Memory
IC	Integrated Circuit
IP	Intellectual Property – proprietary circuit implementations from multiple vendors intended to be incorporated into a larger IC design
ISA	Instruction Set Architecture
ITLB	Instruction Translation Look-aside Buffer
JTLB	Joint Translation Look-aside Buffer
MAC	Multiply-Accumulator, CPU such as iMAC (integer MAC), fMAC (floating-point MAC)
	Media Access Controller, Ethernet controller such as eMAC
MBPS	Million(s of) bits per second
MII	Media-Independent Interface
MIPS I, II, III	Instruction set capabilities of successive generations of MIPS series processors
NMI	Non-Maskable Interrupt
PCI	Peripheral Component Interconnect
PHY	Physical Layer
	Chip on which a physical layer is implemented
PLL	Phase-Locked Loop
R/O	Read-Only

R/W	Read/Write
Rx	Receive
S/C	Self-Clearing
TBD	To Be Determined
TP-PMD	Twisted Pair-Physical Medium Dependent
TS	Time Slot
Tx	Transmit
UCAB	Un-cached Accelerated Buffer
UART	Universal Asynchronous Receiver/Transmitter
VAddr	Virtual Address
WBB	Write-Back Buffer
W/O	Write-Only

1.2.2 Other Terminology

To ***Assert*** a signal means to take it to its active level. An active high signal is "1" when asserted, and an active low signal is "0" when asserted.

1.3 Conventions

- **Register values** are expressed in this manual as base 16 (hexadecimal) numbers, and are prefixed by the characters "0x", which are not part of the number, in keeping with Verilog (and C) Language Terminology. In order to improve legibility, the underscore character (_) separates numbers larger than four hex digits in length into four-digit groups.
- **Internal signals** start with a lower case module name, followed by suffixes with capitalized initial letter(s).
- **Active Low internal signals** are indicated by signal names ending with the letter B.
- Signals on **external pins** are always written completely in UPPER CASE, for example, BOOT16.
- **Active Low external signals** are indicated by signal names ending with an asterisk (*).
- A **bit field** within a register is referred to by the terminology: Register[bit_field] when referring to the bit field by name, or by its bit position.

2. Features

■ C790 integrated high-performance RISC processor core with 128-bit internal architecture optimized for high data throughput

- 2-way super-scalar pipeline with 128-bit (2x64-bit) data path
- 200/266 MHz operation
- MIPS I, II, III compatible ISA with selected MIPS IV ISA (Pre-fetch and Move Conditional Instructions)
- Additional multimedia instruction set support to provide SIMD operation
- 32 KB two-way set associative Instruction Cache, and 32 KB two-way set associative Data Cache
- Line lockable Data cache, write back cache (WBB), non-blocking load, and data cache Pre-fetch instruction to enhance performance
- 64-entry fully associative branch target address cache
- 48-entry fully associative JTLB supporting 4 KB –16 MB page size. WinCE profile II recommended.
- IEEE754 Double Precision FPU tightly coupled with the CPU core. FPU is compatible with the TX49.
- Bi-endian (Little Endian and Big Endian) operations supported

■ Dual 10/100 Mbps Ethernet Media Access Controller with scatter-gather DMA bus master capability ("MAC")

- Supports 10 or 100 Mbps MII-based PHY devices including 100BASE-TX, 100BASE-FX & 100BASE-T4
- Media-Independent Interface (MII) Management feature
- Supports ENDEC, TP-PMD & Fiber PMD devices
- Supports Half and Full Duplex operation
- Scatter-Gather DMA bus master capability

■ PCI/G-Bus Bridge ("PGB")

- Fully compliant with PCI Local Bus Specification Rev. 2.1
- 32-bit PCI bus interface
- 33 MHz or 66 MHz PCI operation
- Zero-Latency Back to Back transfers
- Supports on-chip arbitration of up to 5 masters (supports a maximum of 2 - 4 external PCI devices)
- Dual Address cycle
- Supports all PCI specific configuration registers

■ SDRAM Memory Controller ("SDRAMC")

The SDRAM Memory Controller has been designed and integrated into the TX7901 processor to take advantage of high memory bandwidth to external memories. The C790 and the DMA Controller can both access memory to perform Read / Write operations. The SDRAMC supports:

- 1 GB memory with four DIMMs ($8M \times 8B \times 4$) or two double-side DIMMs ($8M \times 8B \times 4 \times 2$)
- PC 100/133 DIMM/SO-DIMM
- Internal PLL for de-skewing clock and data between the TX7901 and SDRAM DIMMs
- Four-bank interleaving for 64/128/256 Mb SDRAMs
- Two-bank interleaving for 16 Mb SDRAMs
- ECC, single-bit error correction, double-bit error detection
- Aligned burst transactions

■ DMA Controller ("DMAC")

Eight independent DMA channels have been implemented for external and internal peripherals that access memory. DMA operation provides the fastest access to memory while the C790 executes in parallel from internal caches. The DMAC supports:

- Eight independent DMA channels for both internal and external DMA requests
- Chaining via linked lists of records
- Block and Slice modes
- Memory to memory, memory to peripheral, and peripheral to memory transactions

■ Interrupt Controller

The Interrupt controller in the TX7901 supports both internal and external interrupts to the C790 core. It contains an interrupt source register to identify up to 22 different interrupt sources. In addition, there is an interrupt register mask that is used to mask interrupt sources to the C790. The supported Interrupts are:

- PCI Controller, MACs, DMAC, UARTs, Timers, SPI Interrupts, Bus errors and external interrupts

■ Non-Maskable Interrupt (NMI)

Non-Maskable Interrupts are used to indicate fatal conditions. ECC uncorrectable errors, the watchdog timer, and external NMI pins are sources of NMIs.

■ Triple Timer-Counters ("TMR")

The three timer counters integrated into the TX7901 each have their own particular 24-bit up counter and control registers for implementing each timer control "channel".

- Timer 0 is an internal timer that uses the internal clock and operates in the Internal Timer Mode, which causes periodic interrupts.
- Timer 1 for Channel 1 operates in the Interval Timer Mode or Pulse Generator Mode, which generates waveforms of arbitrary frequencies and duty ratios.
- Timer 2 for Channel 2 operates in the Pulse Generator Mode and Watchdog Timer mode, monitoring system runaway conditions in the Interval Timer mode.

■ UART

- Dual UARTs
- Modem flow control (CTS/RTS)
- Baud rate generator
- Software-compatible with NSC NS16550A

■ SPI

- Master operation
- Supports Serial Boot ROM and RTC. Flash ROM and EEPROM are supported by the Companion chip.
- Supports Atmel serial ROM as Boot ROM

■ HW JTAG Scan Test logic support

- JTAG external test mode for chip boundary tests facilitates board testing.
- Full SCAN design and direct memory test modes facilitate chip testing.

3. Configuration

The following is the block diagram of the TX7901:

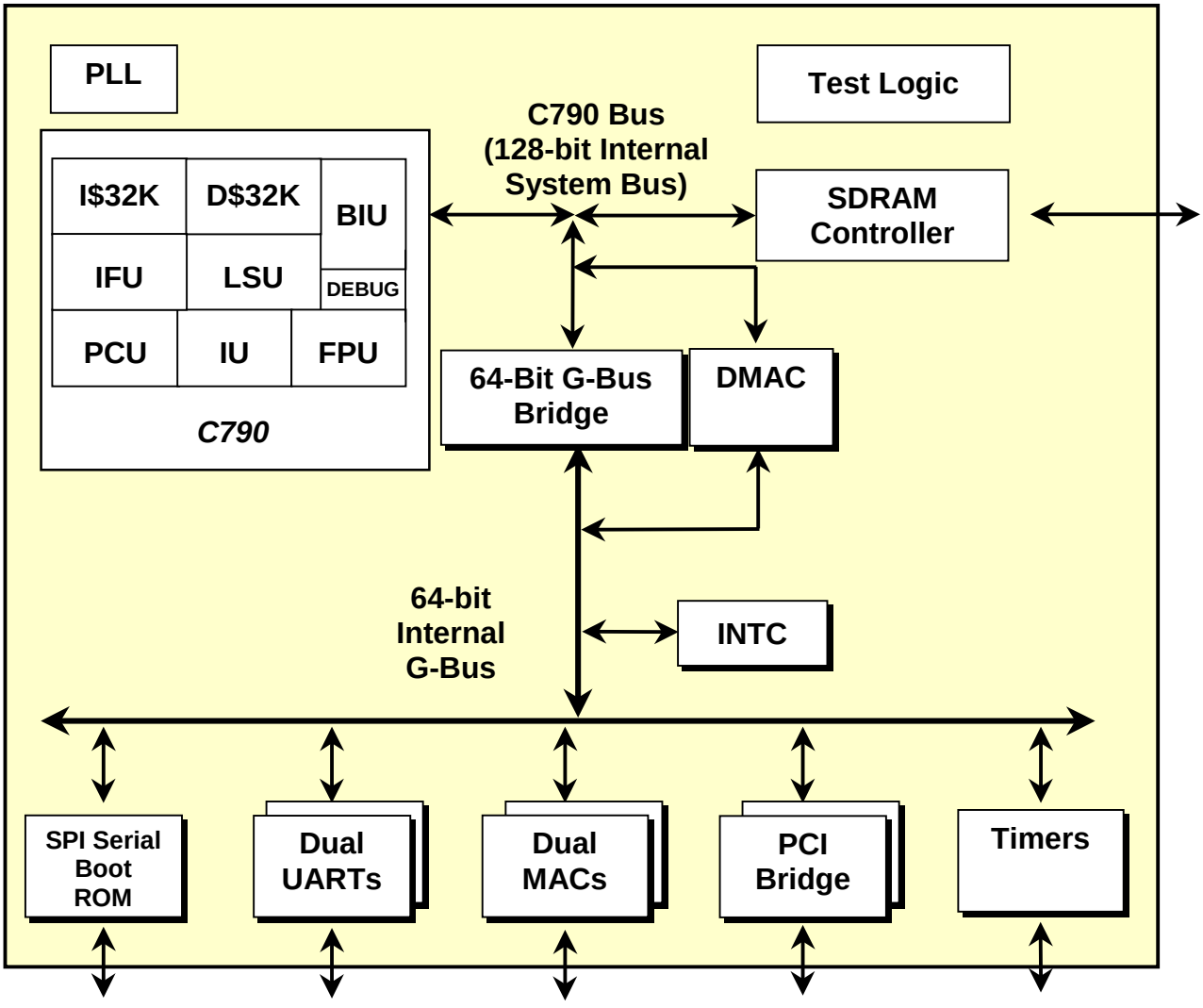


Figure 3-1 TX7901 Block Diagram

■ ***C790***

High performance MIPS RISC processor core with 128-bit internal system bus interface.

■ ***MAC***

Dual 10/100Mbps Ethernet MAC with scatter-gather DMA bus master capability.

■ ***PCI Bridge***

32-bit PCI bus interface compliance with PCI Local Bus Specification Rev. 2.1. PCI0 is 66 MHz/32-bit PCI. PCI1 is 33 MHz/32-bit PCI.

■ ***SDRAMC***

SDRAM memory controller

■ ***DMAC***

8-channel DMA controller

■ ***INTC***

22 internal and external sources of interrupts, and interrupt controller for these interrupt sources.

■ ***Timers***

3-channel 24-bit up counters work as the interval timer, pulse generator, and watchdog timer.

■ ***UART***

2-channel serial I/Os, NS 16550 software compatible
Channel 0 has full function. Channel 1 has two pins (SW, SOUT) only, and is used for the debug monitor.

■ ***SPI***

Serial Peripheral Interface connects the serial Boot ROM and Real-time clock

■ ***64-Bit G-Bus Bridge***

A bridge between the 128-bit internal system bus and the 64-bit G-Bus

■ ***PLL***

Phase-Locked-Loop to generate the TX7901's internal clocks from an external oscillator

■ ***Test Logic***

Supports Scan and JTAG.

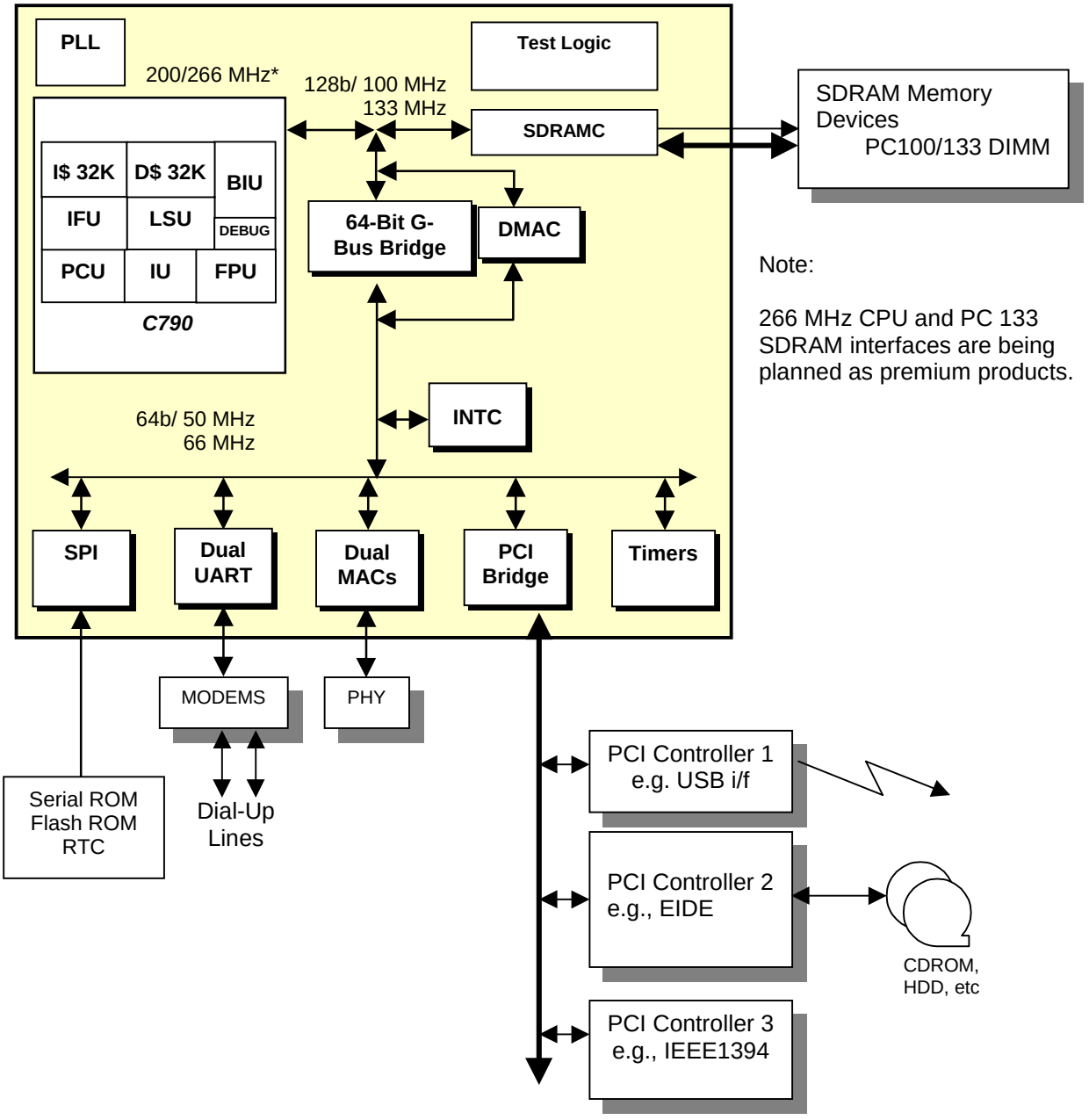


Figure 3-2 A typical system utilizing TX7901

3.1 Reset Configuration

C790: Pins define the endianness.

SPI: 2 MHz (133)/1.56 MHz (100) of the bit rate accesses Boot ROM.

SDRAM: 8 MB per DIMM (chip select) starting at physical address 0x0000_0000.

4.Address Maps

4.1 Memory Map

The physical memory space of the TX7901 is 4 GB. The memory management unit (MMU) of the C790 manages the memory map of the TX7901. The TX7901 virtual and physical addresses are both 32 bits wide.

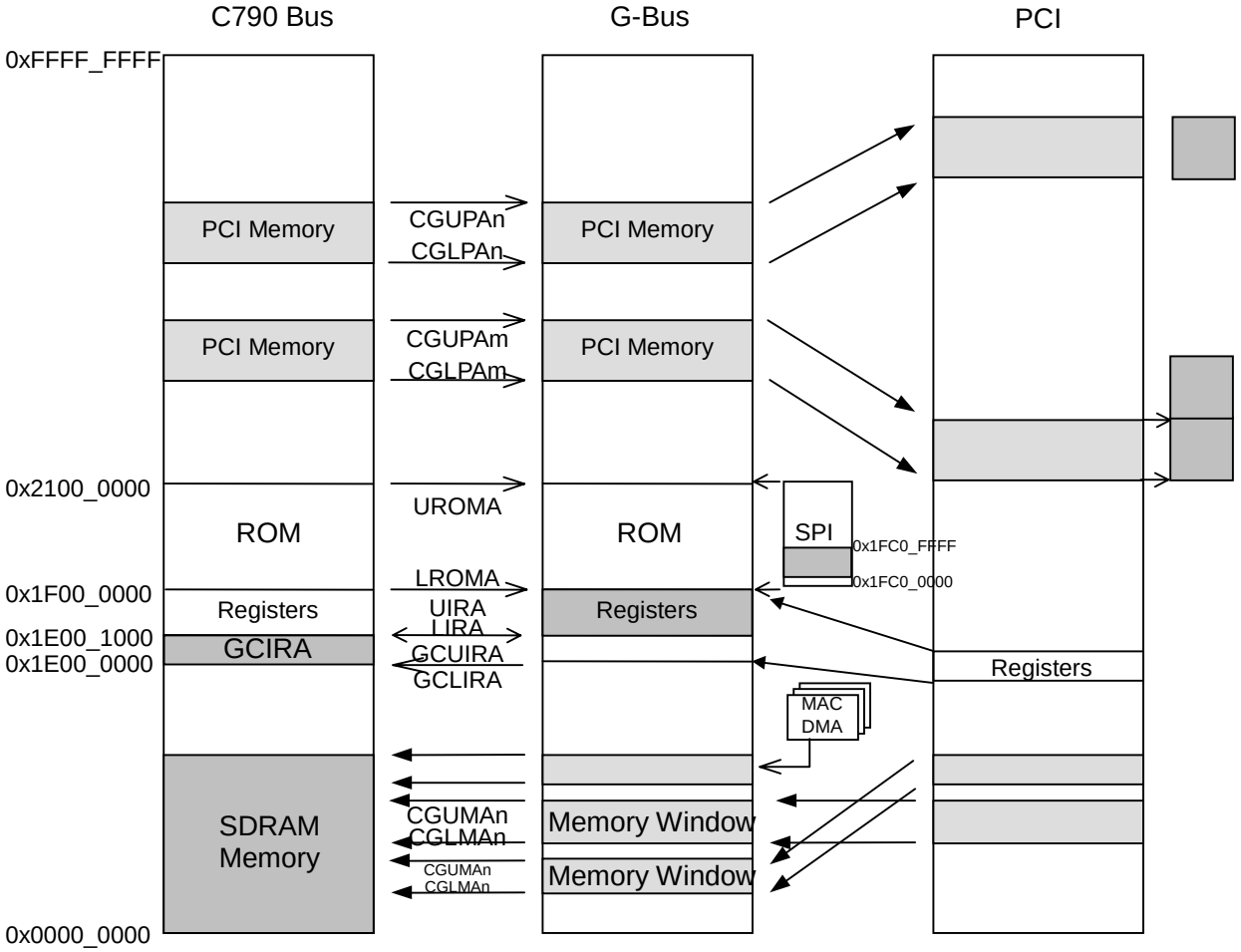


Figure 4-1 Memory Map

TX7901 internal registers are mapped from 0x1E00_0000 to 0x1EFF_FFFF (16 MB). The ROM/SRAM addresses are mapped from 0x1F00_0000 to 0x20FF_FFFF (32 MB).

Main memory space (SDRAM) can be located anywhere except in the internal register range. This memory space is located on the C790 Bus. PCI memory space can also be located anywhere except in that range, and up to four segments (including one I/O space) are allowed. Note that all nine segments must be *non-overlapping* and are programmed by the user. For details, please refer to the chapters on the SDRAM memory controller and PCI controller.

4.2 Register Map

The following is the register map of the TX7901 built-in modules:

0x1E00_0000	-	0x1E00_0FFF	SDRAM Memory Controller on the C790 Bus
0x1E00_1000	-	0x1E00_1FFF	DMA Controller
0x1E00_2000	-	0x1E00_2FFF	G-Bus Bridge and Interrupt Controller
0x1E00_3000	-	0x1E00_3FFF	PCI Bridge (PGB)
0x1E00_4000	-	0x1E00_4FFF	Timer/Counter
0x1E00_5000	-	0x1E00_5FFF	MAC0
0x1E00_6000	-	0x1E00_6FFF	MAC1
0x1E00_7000	-	0x1E00_7FFF	UART0
0x1E00_8000	-	0x1E00_8FFF	UART1
0x1E00_9000	-	0x1E00_9FFF	SPI (TSEI) and GPIO
0x1E00_A000	-	0x1E00_AFFF	PCI Bridge (PGB1): optional
0x1E00_B000	-	0x1E00_BFFF	Reserved
0x1E00_C000	-	0x1E00_FFFF	Reserved
0x1E01_0000	-	0x1EFF_FFFF	Reserved

For more information, please refer to the chapter that pertains to the relevant module.

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Table 4-1 List of 7901 Device Registers

Name	Register Description	Address	R/W	Size(b)
SDRAM Memory Controller, Base Address 0x1E00_0000				
D0PR	DIMM 0 Parameters Register	0x1E00_0000	R/W	128
D1PR	DIMM 1 Parameters Register	0x1E00_0010	R/W	128
D2PR	DIMM 2 Parameters Register	0x1E00_0020	R/W	128
D3PR	DIMM 3 Parameters Register	0x1E00_0030	R/W	128
DOMR	Operation Mode Register	0x1E00_0040	R/W	128
DEMR	ECC Mode Register	0x1E00_0050	R/W	128
DEESR	ECC Error Status Register	0x1E00_0060	R	128
DEEAR	ECC Error Address Register	0x1E00_0070	R	128
–	RESERVED	0x1E00_0080		
DREFRESH	Refresh Register	0x1E00_0090	R/W	128
DDRIVE	SDRAM Interface Output Drive-Strength Control Register	0x1E00_00A0	R/W	128
D0LOW	DIMM 0 LOW Address Decode	0x1E00_0100	R/W	128
D0HIGH	DIMM 0 HIGH Address Decode	0x1E00_0110	R/W	128
D1LOW	DIMM 1 LOW Address Decode	0x1E00_0120	R/W	128
D1HIGH	DIMM 1 HIGH Address Decode	0x1E00_0130	R/W	128
D2LOW	DIMM 2 LOW Address Decode	0x1E00_0140	R/W	128
D2HIGH	DIMM 2 HIGH Address Decode	0x1E00_0150	R/W	128
D3LOW	DIMM 3 LOW Address Decode	0x1E00_0160	R/W	128
D3HIGH	DIMM 3 HIGH Address Decode	0x1E00_0170	R/W	128
DMA Controller, Base Address 0x1E00_1000				
CCR0	Channel 0 Control Register	0x1E00_1000	R/W	64
CSR0	Channel 0 Status Register	0x1E00_1010	R/W	64
SAR0	Channel 0 Source Address Register	0x1E00_1020	R/W	64
DAR0	Channel 0 Destination Address Register	0x1E00_1030	R/W	64
BCR0	Channel 0 Byte Count Register	0x1E00_1040	R/W	64
NRPR0	Channel 0 Next Record Pointer Register	0x1E00_1050	R/W	64
--	RESERVED	0x1E00_1060 - 0x1E00_10F0		
CCR1	Channel 1 Control Register	0x1E00_1100	R/W	64
CSR1	Channel 1 Status Register	0x1E00_1110	R/W	64
SAR1	Channel 1 Source Address Register	0x1E00_1120	R/W	64
DAR1	Channel 1 Destination Address Register	0x1E00_1130	R/W	64
BCR1	Channel 1 Byte Count Register	0x1E00_1140	R/W	64
NRPR1	Channel 1 Next Record Pointer Register	0x1E00_1150	R/W	64
--	RESERVED	0x1E00_1160 - 0x1E00_11F0		
CCR2	Channel 2 Control Register	0x1E00_1200	R/W	64
CSR2	Channel 2 Status Register	0x1E00_1210	R/W	64
SAR2	Channel 2 Source Address Register	0x1E00_1220	R/W	64
DAR2	Channel 2 Destination Address Register	0x1E00_1230	R/W	64
BCR2	Channel 2 Byte Count Register	0x1E00_1240	R/W	64
NRPR2	Channel 2 Next Record Pointer Register	0x1E00_1250	R/W	64
-	RESERVED	0x1E00_1260 - 0x1E00_12F0		
CCR3	Channel 3 Control Register	0x1E00_1300	R/W	64
CSR3	Channel 3 Status Register	0x1E00_1310	R/W	64
SAR3	Channel 3 Source Address Register	0x1E00_1320	R/W	64
DAR3	Channel 3 Destination Address Register	0x1E00_1330	R/W	64
BCR3	Channel 3 Byte Count Register	0x1E00_1340	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
NRPR3	Channel 3 Next Record Pointer Register	0x1E00_1350	R/W	64
--	RESERVED	0x1E00_1360 -0x1E00_13F0		
CCR4	Channel 4 Control Register	0x1E00_1400	R/W	64
CSR4	Channel 4 Status Register	0x1E00_1410	R/W	64
SAR4	Channel 4 Source Address Register	0x1E00_1420	R/W	64
DAR4	Channel 4 Destination Address Register	0x1E00_1430	R/W	64
BCR4	Channel 4 Byte Count Register	0x1E00_1440	R/W	64
NRPR4	Channel 4 Next Record Pointer Register	0x1E00_1450	R/W	64
--	RESERVED	0x1E00_1460 -0x1E00_14F0		
CCR5	Channel 5 Control Register	0x1E00_1500	R/W	64
CSR5	Channel 5 Status Register	0x1E00_1510	R/W	64
SAR5	Channel 5 Source Address Register	0x1E00_1520	R/W	64
DAR5	Channel 5 Destination Address Register	0x1E00_1530	R/W	64
BCR5	Channel 5 Byte Count Register	0x1E00_1540	R/W	64
NRPR5	Channel 5 Next Record Pointer Register	0x1E00_1550	R/W	64
--	RESERVED	0x1E00_1560 -0x1E00_15F0		
CCR6	Channel 6 Control Register	0x1E00_1600	R/W	64
CSR6	Channel 6 Status Register	0x1E00_1610	R/W	64
SAR6	Channel 6 Source Address Register	0x1E00_1620	R/W	64
DAR6	Channel 6 Destination Address Register	0x1E00_1630	R/W	64
BCR6	Channel 6 Byte Count Register	0x1E00_1640	R/W	64
NRPR6	Channel 6 Next Record Pointer Register	0x1E00_1650	R/W	64
--	RESERVED	0x1E00_1660 -0x1E00_16F0		
CCR7	Channel 7 Control Register	0x1E00_1700	R/W	64
CSR7	Channel 7 Status Register	0x1E00_1710	R/W	64
SAR7	Channel 7 Source Address Register	0x1E00_1720	R/W	64
DAR7	Channel 7 Destination Address Register	0x1E00_1730	R/W	64
BCR7	Channel 7 Byte Count Register	0x1E00_1740	R/W	64
NRPR7	Channel 7 Next Record Pointer Register	0x1E00_1750	R/W	64
--	RESERVED	0x1E00_1760 -0x1E00_1FF0		
G–Bus Bridge / Chip Configuration / Interrupt Controller, Base Address 0x1E00_2000				
SCR	System Configuration Register	0x1E00_2000	R	64
BCR	C790 Bus Control Register	0x1E00_2008	R/W	64
BSR	C790 Bus Status Register	0x1E00_2010	R/W	64
BBAR	C790 Bus Bad Address Register	0x1E00_2018	R	64
UIRA	CG Upper Internal Register Address	0x1E00_2020	R	64
LIRA	CG Lower Internal Register Address	0x1E00_2028	R	64
UROMA	CG Upper ROM Address Register	0x1E00_2030	R	64
LROMA	CG Lower ROM Address Register	0x1E00_2038	R	64
CGUPA0	CG Upper PCI Address 0	0x1E00_2040	R/W	64
CGLPA0	CG Lower PCI Address 0	0x1E00_2048	R/W	64
CGUPA1	CG Upper PCI Address 1	0x1E00_2050	R/W	64
CGLPA1	CG Lower PCI Address 1	0x1E00_2058	R/W	64
CGUPA2	CG Upper PCI Address 2	0x1E00_2060	R/W	64
CGLPA2	CG Lower PCI Address 2	0x1E00_2068	R/W	64
CGUPA3	CG Upper PCI Address 3	0x1E00_2070	R/W	64
CGLPA3	CG Lower PCI Address 3	0x1E00_2078	R/W	64
GCUIRA	GC Upper Internal Register Address	0x1E00_2080	R	64
GCLIRA	GC Lower Internal Register Address	0x1E00_2088	R	64
GCUMA0	GC Upper MEM Address 0	0x1E00_2090	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
GCLMA0	GC Lower MEM Address 0	0x1E00_2098	R/W	64
GCUMA1	GC Upper MEM Address 1	0x1E00_20A0	R/W	64
GCLMA1	GC Lower MEM Address 1	0x1E00_20A8	R/W	64
GCUMA2	GC Upper MEM Address 2	0x1E00_20B0	R/W	64
GCLMA2	GC Lower MEM Address 2	0x1E00_20B8	R/W	64
GCUMA3	GC Upper MEM Address 3	0x1E00_20C0	R/W	64
GCLMA3	GC Lower MEM Address 3	0x1E00_20C8	R/W	64
GCUMA4	GC Upper MEM Address 4	0x1E00_20D0	R/W	64
GCLMA4	GC Lower MEM Address 4	0x1E00_20D8	R/W	64
IRSTAT	Interrupt Status Register	0x1E00_20E0	R	64
IRMSK	Interrupt Mask Register	0x1E00_20E8	R/W	64
LT	C790 Bus Latency Timer	0x1E00_20F0	R/W	64
NRSTAT	NMI Status Register	0x1E00_20F8	R	64
GBMLT	G-Bus Master Latency Timer	0x1E00_2100	R/W	64
GBBMLT	G-Bus Broken Master Timer	0x1E00_2108	R/W	64
GBSLT	G-Bus Slave Latency Timer	0x1E00_2110	R/W	64
RTT	G-Bus Retry Timer	0x1E00_2118	R/W	64
GCCR	GC Control register	0x1E00_2120	R/W	64
GBSTAT	G-Bus Status Register	0x1E00_2128	R/W	64
GBBAR	G-Bus Bad Address Register	0x1E00_2130	R	64
GBARSR	G-Bus Arbiter Request Status Register	0x1E00_2138	R	64
GBAGSR	G-Bus Arbiter Granted Status Register	0x1E00_2140	R	64
GBAMSR	G-Bus Arbiter Master Status Register	0x1E00_2148	R/W	64
GBACR	G-Bus Arbiter Control Register	0x1E00_2150	R/W	64
PCI / G-Bus Bridge / PCI Controller, Base Address 0x1E00_3000				
PCI0 Configuration Space	Device & Vendor ID Register	0x1E00_3000	R	32
	Status & Command Register	0x1E00_3004	R/W	32
	Class Code, Revision ID Register	0x1E00_3008	R	32
	BIST, Header Type, Master Latency Timer & Cache line Size	0x1E00_300C	R/W	32
	Memory Base Address[0]	0x1E00_3010	R/W	32
	Memory DAC Base Address[0]	0x1E00_3014	R/W	32
	Memory Base Address[1]	0x1E00_3018	R/W	32
	Memory DAC Base Address[1]	0x1E00_301C	R/W	32
	Memory Base Address[2]	0x1E00_3020	R/W	32
	Memory DAC Base Address[2]	0x1E00_3024	R/W	32
	Reserved	0x1E00_3028		
	Subsystem ID, Subsystem Vendor ID	0x1E00_302C	R	32
	XX,XX,XX,XX	0x1E00_3030	R/W	32
	Reserved, 0xDC	0x1E00_3034	R	32
	Reserved	0x1E00_3038		
	Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line	0x1E00_303C	R	32
	Reserved, Retry Time Value, TRDY Timeout	0x1E00_3040	R/W	32
	I/O Base Address [0]	0x1E00_3044	R/W	32
	Reserved	0x1E00_3048 - 0x1E00_30D8		
	Pre-existing features, 0xE401	0x1E00_30DC	R/W	32
	Pre-existing features	0x1E00_30E0	R/W	32
	Reserved, p2gBase3[35:32], 0x0002	0x1E00_30E4	R/W	32
	p2gBase3[31:0]	0x1E00_30E8	R/W	32
	Reserved	0x1E00_30EC - 0x1E00_30FF		

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
PgbCSR	PGB Control and Status Register	0x1E00_3100	R/W	64
g2pLower0	g2pwindow Lower Address Register 0	0x1E00_3108	R/W	64
g2pUpper0	g2pwindow Upper Address Register 0	0x1E00_3110	R/W	64
g2pLower1	g2pwindow Lower Address Register 1	0x1E00_3118	R/W	64
g2pUpper1	g2pwindow Upper Address Register 1	0x1E00_3120	R/W	64
g2pLower2	g2pwindow Lower Address Register 2	0x1E00_3128	R/W	64
g2pUpper2	g2pwindow Upper Address Register 2	0x1E00_3130	R/W	64
g2pLower3	g2pwindow Lower Address Register 3	0x1E00_3138	R/W	64
g2pUpper3	g2pwindow Upper Address Register 3	0x1E00_3140	R/W	64
g2pBase0	g2pwindow Base Address Register 0	0x1E00_3148	R/W	64
g2pBase1	g2pwindow Base Address Register 1	0x1E00_3150	R/W	64
g2pBase2	g2pwindow Base Address Register 2	0x1E00_3158	R/W	64
g2pBase3	g2pwindow Base Address Register 3	0x1E00_3160	R/W	64
g2pCycleType	g2pwindow Cycle Type Register	0x1E00_3168	R/W	64
p2gBase0	p2gwindow Base Address Register 0	0x1E00_3170	R/W	64
p2gBase1	p2gwindow Base Address Register 1	0x1E00_3178	R/W	64
p2gBase2	p2gwindow Base Address Register 2	0x1E00_3180	R/W	64
p2gBase3	p2gwindow Base Address Register 3	0x1E00_3188	R/W	64
Ia	Failing Transaction Address Register	0x1E00_3190	R/W	64
--	Reserved	0x1E00_3198 - 0x1E00_3FFF		
Programmable Timer / Counters, Base Address 0x1E00_4000				
TMTCR0	Timer Control Register 0	0x1E00_4000	R/W	32
TMTISR0	Timer Interrupt Status Register 0	0x1E00_4004	R/W	32
TMCPRA0	Compare Register A 0	0x1E00_4008	R/W	32
TMCPRB0	Compare Register B 0	0x1E00_400C	R/W	32
TMITMR0	Interval Timer Mode Register 0	0x1E00_4010	R/W	32
TMCCDR0	Divider Register 0	0x1E00_4020	R/W	32
-	Reserved	0x1E00_4030		
-	Reserved	0x1E00_4040		
TMTRR0	Timer Read Register 0	0x1E00_40F0	R/O	32
TMTCR1	Timer Control Register 1	0x1E00_4100	R/W	32
TMTISR1	Timer Interrupt Status Register 1	0x1E00_4104	R/W	32
TMCPRA1	Compare Register A 1	0x1E00_4108	R/W	32
TMCPRB1	Compare Register B 1	0x1E00_410C	R/W	32
TMITMR1	Interval Timer Mode Register 1	0x1E00_4110	R/W	32
TMCCDR1	Divider Register 1	0x1E00_4120	R/W	32
TMPGMR1	Pulse Generator Mode Register 1	0x1E00_4130	R/W	32
-	Reserved	0x1E00_4140	R/O	32
TMTRR1	Timer Read Register 1	0x1E00_41F0	R/W	32
TMTCR2	Timer Control Register 2	0x1E00_4200	R/W	32
TMTISR2	Timer Interrupt Status Register 2	0x1E00_4204	R/W	32
TMCPRA2	Compare Register A 2	0x1E00_4208	R/W	32
TMCPRB2	Compare Register B 2	0x1E00_420C	R/W	32
TMITMR2	Interval Timer Mode Register 2	0x1E00_4210	R/W	32
TMCCDR2	Divider Register 2	0x1E00_4220	R/W	32
TMPGMR2	Pulse Generator Mode Register 2	0x1E00_4230	R/W	32
TMWTMR2	Watch Dog Timer Mode Register 2	0x1E00_4240	R/W	32
TMTRR2	Timer Read Register 2	0x1E00_42F0	R/W	32

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
Dual Ethernet Media Access Controllers, Base Addresses 0x1E00_5000 and 0x1E00_6000 (Note: Counters start at offsets 0x200)				
-	Reserved	0x1E00_5000	R	
CCReg0	Command & Configuration Register 0	0x1E00_5008	R/W	64
TFCReg0	Transmit Frame Configuration 0	0x1E00_5010	R/W	64
RFCReg0	Receive Frame Configuration 0	0x1E00_5018	R/W	64
TSReg0	Transmit Status Register 0	0x1E00_5020	R	64
RSReg0	Receive Status Register 0	0x1E00_5028	R	64
TIMReg0	Transmit Interrupt Mask Register 0	0x1E00_5030	R/W	64
TIReg0	Transmit Interrupt Register 0	0x1E00_5038	R	64
RIMReg0	Receive Interrupt Mask Register0	0x1E00_5040	R/W	64
RIReg0	Receive Interrupt Register0	0x1E00_5048	R	64
-	Reserved	0x1E00_5050		
-	Reserved	0x1E00_5058		
TPFTReg0	Transmit Pause Frame Timer Register0	0x1E00_5060	R/W	64
VLANReg0	VLAN Tag Register 0	0x1E00_5068	R/W	64
TDPReg0	Transmit Frame Descriptor Pointer Register 0	0x1E00_5070	R/W	64
RDPReg0	Receive Frame Descriptor Pointer Register 0	0x1E00_5078	R/W	64
CDPReg0	Current Frame Descriptor Pointer Register0	0x1E00_5080	R	64
BusErrReg0	Bus Error Address Register0	0x1E00_5088	R	64
TCDReg	Transmit Frame Current Descriptor Pointer	0x1E00_5090	R	32
RCDReg	Receive Frame Current Descriptor Pointer	0x1E00_5098	R	32
-	Reserved	0x1E00_50A0 - 0x1E00_50FF		
peMACC0	Internal Test Register 0 (peMACC)	0x1E00_5100	R/W	64
peMACT0	Internal Test Register 0 (peMACT)	0x1E00_5108	R/W	64
IPGReg0	Back-to-Back IPG gap0	0x1E00_5110	R/W	64
NBTBReg0	Non Back-to-Back IPG gap0	0x1E00_5118	R/W	64
peCLRT0	Internal Test Register0 (peCLRT)	0x1E00_5120	R/W	64
peMAXF0	Internal Test Register0 (peMAXF)	0x1E00_5128	R/W	64
pePNCT0	Internal Test Register0 (pePNCT)	0x1E00_5130	R/W	64
peTBCT0	Internal Test Register 0 (peTBCT)	0x1E00_5138	R/W	64
LSAII0	Local Station Addr II	0x1E00_51A8	R/W	64
LSAI0	Local Station Addr I	0x1E00_51B0	R/W	64
peVLTP0	Internal Test Register (peVLTP)	0x1E00_51C8	R/W	64
TBTCnt0	Total Bytes Transmitted Count Register	0x1E00_5200	R/W	64
TGFTCnt0	Total Good Frames Transmitted	0x1E00_5208	R/W	64
MFTCnt0	Multicast Frames Transmitted	0x1E00_5210	R/W	64
BFTCnt0	Broadcast Frames Transmitted	0x1E00_5218	R/W	64
TxFrame64_0	Frames Transmitted (TxFrame64)	0x1E00_5220	R/W	64
TxFrame127_0	Frames Transmitted (TxFrame127)	0x1E00_5228	R/W	64
TxFrame255_0	Frames Transmitted (TxFrame255)	0x1E00_5230	R/W	64
TxFrame511_0	Frames Transmitted (TxFrame511)	0x1E00_5238	R/W	64
TxFrame1K0	Frames Transmitted (TxFrame1K)	0x1E00_5240	R/W	64
TxFrameGt1K0	Frames Transmitted (TxFrameGt1K)	0x1E00_5248	R/W	64
MPFTCnt0	MAC Pause Frames Transmitted	0x1E00_5250	R/W	64
LFTCnt0	Long Frames Transmitted	0x1E00_5258	R/W	64
TCCnt0	Total Collisions	0x1E00_5260	R/W	64
LCCnt0	Late Collision	0x1E00_5268	R/W	64
MCCnt0	Multiple Collision	0x1E00_5270	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
SCCnt0	Single Collision	0x1E00_5278	R/W	64
EDCnt0	Excessive Deferrals	0x1E00_5280	R/W	64
TRECnt0	Transmit Retry Errors	0x1E00_5288	R/W	64
TUECnt0	Transmit Underflow Errors	0x1E00_5290	R/W	64
-	Reserved	0x1E00_5298		
TBRCnt0	Total Bytes Received	0x1E00_52A0	R/W	64
TRFRCnt0	Total Readable Frames Received	0x1E00_52A8	R/W	64
MFRCnt0	Multicast Frames Received	0x1E00_52B0	R/W	64
BFRCnt0	Broadcast Frames Received	0x1E00_52B8	R/W	64
RxFrame64_0	Frames Received (RxFrame64)	0x1E00_52C0	R/W	64
RxFrame127_0	Frames Received (RxFrame127)	0x1E00_52C8	R/W	64
RxFrame255_0	Frames Received (RxFrame255)	0x1E00_52D0	R/W	64
RxFrame511_0	Frames Received (RxFrame511)	0x1E00_52D8	R/W	64
RxFrame1K0	Frames Received (RxFrame1K)	0x1E00_52E0	R/W	64
RxFrameGt1K0	Frames Received (RxFrameGt1K)	0x1E00_52E8	R/W	64
MCFRCnt0	MAC Pause Frames Received	0x1E00_52F0	R/W	64
LFRCnt0	Long Frames Received	0x1E00_52F8	R/W	64
RECnt0	Receive Errors	0x1E00_5300	R/W	64
FRBCCnt0	Frames Received with Bad CRC	0x1E00_5308	R/W	64
MFRCnt0	Misaligned Frames Received	0x1E00_5310	R/W	64
UFCnt0	Undersized Frames	0x1E00_5318	R/W	64
FFCnt0	Fragmented Frames	0x1E00_5320	R/W	64
JFRCnt0	Jabber Frames Received	0x1E00_5328	R/W	64
NRDMFCnt0	No RxDescriptor Missed Frames	0x1E00_5330	R/W	64
NRMFCnt0	No RxFIFO Missed Frames	0x1E00_5338	R/W	64
MIIMCR0	MIIM Control Register	0x1E00_5400	R/W	64
MIIMDR0	MIIM Data Register	0x1E00_5408	R/W	64
PhyAddr0_0	Physical Address 0	0x1E00_5600	R/W	64
PhyAddr1_0	Physical Address 1	0x1E00_5608	R/W	64
PhyAddr2_0	Physical Address 2	0x1E00_5610	R/W	64
PhyAddr3_0	Physical Address 3	0x1E00_5618	R/W	64
PhyAddr4_0	Physical Address 4	0x1E00_5620	R/W	64
PhyAddr5_0	Physical Address 5	0x1E00_5628	R/W	64
PhyAddr6_0	Physical Address 6	0x1E00_5630	R/W	64
PhyAddr7_0	Physical Address 7	0x1E00_5638	R/W	64
PhyAddr8_0	Physical Address 8	0x1E00_5640	R/W	64
PhyAddr9_0	Physical Address 9	0x1E00_5648	R/W	64
PhyAddrA_0	Physical Address A	0x1E00_5650	R/W	64
PhyAddrB_0	Physical Address B	0x1E00_5658	R/W	64
PhyAddrC_0	Physical Address C	0x1E00_5660	R/W	64
PhyAddrD_0	Physical Address D	0x1E00_5668	R/W	64
PhyAddrE_0	Physical Address E	0x1E00_5670	R/W	64
PhyAddrF_0	Physical Address F	0x1E00_5678	R/W	64
-	Reserved	0x1E00_6000		
CCReg1	Command & Configuration Register	0x1E00_6008	R/W	64
TFCReg1	Transmit Frame Configuration	0x1E00_6010	R/W	64
RFCReg1	Receive Frame Configuration	0x1E00_6018	R/W	64
TSReg1	Transmit Status Register	0x1E00_6020	R/W	64
RSReg1	Receive Status Register	0x1E00_6028	R/W	64
TIMReg1	Transmit Interrupt Mask Register	0x1E00_6030	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
TIReg1	Transmit Interrupt Register	0x1E00_6038	R/W	64
RIMReg1	Receive Interrupt Mask Register	0x1E00_6040	R/W	64
RIReg1	Receive Interrupt Register	0x1E00_6048	R/W	64
-	Reserved	0x1E00_6050		
-	Reserved	0x1E00_6058		
TPFTReg1	Transmit Pause Frame Timer Register	0x1E00_6060	R/W	64
VLANReg1	VLAN Tag Register	0x1E00_6068	R/W	64
TDPReg1	Transmit Frame Descriptor Pointer Register	0x1E00_6070	R/W	64
RDPReg1	Receive Frame Descriptor Pointer Register	0x1E00_6078	R/W	64
CDPReg1	Current Frame Descriptor Pointer Register	0x1E00_6080	R/W	64
BusErrReg1	Bus Error Address Register	0x1E00_6088	R/W	64
peMACC1	Internal Test Register (peMACC)	0x1E00_6100	R/W	64
peMACT1	Internal Test Register (peMACT)	0x1E00_6108	R/W	64
IPGReg1	Back-to-Back IPG gap0	0x1E00_6110	R/W	64
NBTBReg1	Non Back-to-Back IPG gap0	0x1E00_6118	R/W	64
peCLRT1	Internal Test Register0 (peCLRT)	0x1E00_6120	R/W	64
peMAXF1	Internal Test Register0 (peMAXF)	0x1E00_6128	R/W	64
pePNCT1	Internal Test Register0(pePNCT)	0x1E00_6130	R/W	64
peTBCT1	Internal Test Register 0 (peTBCT)	0x1E00_6138	R/W	64
LSAII1	Local Station Addr II	0x1E00_61A8	R/W	64
LSAI1	Local Station Addr I	0x1E00_61B0	R/W	64
peVLTP1	Internal Test Register (peVLTP)	0x1E00_61C8	R/W	64
TBTCnt1	Total Bytes Transmitted Count Register	0x1E00_6200	R/W	64
TGFTCnt1	Total Good Frames Transmitted	0x1E00_6208	R/W	64
MFTCnt1	Multicast Frames Transmitted	0x1E00_6210	R/W	64
BFTCnt1	Broadcast Frames Transmitted	0x1E00_6218	R/W	64
TxFrame64_1	Frames Transmitted (TxFrame64)	0x1E00_6220	R/W	64
TxFrame127_1	Frames Transmitted (TxFrame127)	0x1E00_6228	R/W	64
TxFrame255_1	Frames Transmitted (TxFrame255)	0x1E00_6230	R/W	64
TxFrame511_1	Frames Transmitted (TxFrame511)	0x1E00_6238	R/W	64
TxFrame1K1	Frames Transmitted (TxFrame1K)	0x1E00_6240	R/W	64
TxFrameGt1K1	Frames Transmitted (TxFrameGt1K)	0x1E00_6248	R/W	64
MPFTCnt1	MAC Pause Frames Transmitted	0x1E00_6250	R/W	64
LFTCnt1	Long Frames Transmitted	0x1E00_6258	R/W	64
TCCnt1	Total Collisions	0x1E00_6260	R/W	64
LCCnt1	Late Collision	0x1E00_6268	R/W	64
MCCnt1	Multiple Collision	0x1E00_6270	R/W	64
SCCnt1	Single Collision	0x1E00_6278	R/W	64
EDCnt1	Excessive Deferrals	0x1E00_6280	R/W	64
TRECnt1	Transmit Retry Errors	0x1E00_6288	R/W	64
TUECnt1	Transmit Underflow Errors	0x1E00_6290	R/W	64
-	Reserved	0x1E00_6298		
TBRCnt1	Total Bytes Received	0x1E00_62A0	R/W	64
TRFRCnt1	Total Readable Frames Received	0x1E00_62A8	R/W	64
MFRCnt1	Multicast Frames Received	0x1E00_62B0	R/W	64
BFRCnt1	Broadcast Frames Received	0x1E00_62B8	R/W	64
RxFrame64_1	Frames Received (RxFrame64)	0x1E00_62C0	R/W	64
RxFrame127_1	Frames Received (RxFrame127)	0x1E00_62C8	R/W	64
RxFrame255_1	Frames Received (RxFrame255)	0x1E00_62D0	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
RxFrame511_1	Frames Received (RxFrame511)	0x1E00_62D8	R/W	64
RxFrame1K1	Frames Received (RxFrame1K)	0x1E00_62E0	R/W	64
RxFrameGt1K1	Frames Received (RxFrameGt1K)	0x1E00_62E8	R/W	64
MCFRCnt1	MAC Pause Frames Received	0x1E00_62F0	R/W	64
LFRCnt1	Long Frames Received	0x1E00_62F8	R/W	64
RECnt1	Receive Errors	0x1E00_6300	R/W	64
FRBCCnt1	Frames Received with Bad CRC	0x1E00_6308	R/W	64
MFRCnt1	Misaligned Frames Received	0x1E00_6310	R/W	64
UFCnt1	Undersized Frames	0x1E00_6318	R/W	64
FFCnt1	Fragmented Frames	0x1E00_6320	R/W	64
JFRCnt1	Jabber Frames Received	0x1E00_6328	R/W	64
NRDMFCnt1	No RxDescriptor Missed Frames	0x1E00_6330	R/W	64
NRMFCnt1	No RxFIFO Missed Frames	0x1E00_6338	R/W	64
MIIMCR1	MIIM Control Register	0x1E00_6400	R/W	64
MIIMDR1	MIIM Data Register	0x1E00_6408	R/W	64
PhyAddr0_1	Physical Address 0	0x1E00_6600	R/W	64
PhyAddr1_1	Physical Address 1	0x1E00_6608	R/W	64
PhyAddr2_1	Physical Address 2	0x1E00_6610	R/W	64
PhyAddr3_1	Physical Address 3	0x1E00_6618	R/W	64
PhyAddr4_1	Physical Address 4	0x1E00_6620	R/W	64
PhyAddr5_1	Physical Address 5	0x1E00_6628	R/W	64
PhyAddr6_1	Physical Address 6	0x1E00_6630	R/W	64
PhyAddr7_1	Physical Address 7	0x1E00_6638	R/W	64
PhyAddr8_1	Physical Address 8	0x1E00_6640	R/W	64
PhyAddr9_1	Physical Address 9	0x1E00_6648	R/W	64
PhyAddrA_1	Physical Address A	0x1E00_6650	R/W	64
PhyAddrB_1	Physical Address B	0x1E00_6658	R/W	64
PhyAddrC_1	Physical Address C	0x1E00_6660	R/W	64
PhyAddrD_1	Physical Address D	0x1E00_6668	R/W	64
PhyAddrE_1	Physical Address E	0x1E00_6670	R/W	64
PhyAddrF_1	Physical Address F	0x1E00_6678	R/W	64
UART0, UART1, Base Addresses 0x1E00_7000 and 0x1E00_8000				
RBR0	Receive Buffer Register 0	0x1E00_7000	R/O	8
THR0	Transmit Holding Register 0	0x1E00_7000	W/O	8
DLL0	Divisor Latch 0 (LS)	0x1E00_7000	R/W	8
IER0	Interrupt Enable Register 0	0x1E00_7004	R/W	8
DLM0	Divisor Latch 0 (MS)	0x1E00_7004	R/W	8
IIR0	Interrupt Identification Register 0	0x1E00_7008	R/O	8
FCR0	FIFO Control Register 0	0x1E00_7008	W/O	8
LCR0	Line Control Register 0	0x1E00_700C	R/W	8
MCR0	Modem Control Register 0	0x1E00_7010	R/W	8
LSR0	Line Status Register 0	0x1E00_7014	R/O	8
MSR0	Modem Status Register 0	0x1E00_7018	R/W	8
SCR0	Scratch Register 0	0x1E00_701C	R/W	8
PSR0	Pre-scaler Register 0	0x1E00_7020	R/W	8
RBR1	Receive Buffer Register 1	0x1E00_8000	R/O	8
THR1	Transmit Holding Register 1	0x1E00_8000	W/O	8
DLL1	Divisor Latch 1 (LS)	0x1E00_8000	R/W	8
IER1	Interrupt Enable Register 1	0x1E00_8004	R/W	8

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
DLM1	Divisor Latch 1 (MS)	0x1E00_8004	R/W	8
IIR1	Interrupt Identification Register 1	0x1E00_8008	R/O	8
FCR1	FIFO Control Register 1	0x1E00_8008	W/O	8
LCR1	Line Control Register 1	0x1E00_800C	R/W	8
MCR1	Modem Control Register 1	0x1E00_8010	R/W	8
LSR1	Line Status Register 1	0x1E00_8014	R/O	8
MSR1	Modem Status Register 1	0x1E00_8018	R/W	8
SCR1	Scratch Register 1	0x1E00_801C	R/W	8
PSR1	Pre-scaler Register 1	0x1E00_8020	R/W	8
SPI (TSEI) and GPIO Base Address 0x1E00_9000				
GPIO_outreg	General-purpose I/O Register	0x1E00_9000	R/W	8
GPIO_inreg	General-purpose I/O Register	0x1E00_9004	R/O	8
GPIO_outenab	General-purpose I/O Register	0x1E00_9008	R/W	8
-	Reserved	0x1E00_900C		
TSEI_SECR	Control Register	0x1E00_9010	R/W	8
TSEI_SESR	Status Register	0x1E00_9014	R/W	8
TSEI_SEDR	Data Register	0x1E00_9018	R/W	8
TSEI_DDCR	Data Direction Register	0x1E00_901C	R/W	8
PGB1 (Optional) Base Address 0x1E00_A000				
PCI1 Configuration Space	Device & Vendor ID Register	0x1E00_A000	R	32
	Status & Command Register	0x1E00_A004	R/W	32
	Class Code, Revision ID Register	0x1E00_A008	R	32
	BIST, Header Type, Master Latency Timer & Cache line Size	0x1E00_A00C	R/W	32
	Memory Base Address[0]	0x1E00_A010	R/W	32
	Memory DAC Base Address[0]	0x1E00_A014	R/W	32
	Memory Base Address[1]	0x1E00_A018	R/W	32
	Memory DAC Base Address[1]	0x1E00_A01C	R/W	32
	Memory Base Address[2]	0x1E00_A020	R/W	32
	Memory DAC Base Address[2]	0x1E00_A024	R/W	32
	Reserved	0x1E00_A028		
	Subsystem ID, Subsystem Vendor ID	0x1E00_A02C	R	32
	XX,XX,XX,XX	0x1E00_A030	R/W	32
	Reserved, 0xDC	0x1E00_A034	R	32
	Reserved	0x1E00_A038		
	Max_Lat, Min_Gnt, Interrupt Pin, Interrupt Line	0x1E00_A03C	R	32
	Reserved, Retry Time Value, TRDY Timeout	0x1E00_A040	R/W	32
	I/O Base Address [0]	0x1E00_A044	R/W	32
	Reserved	0x1E00_A048 – 0x1E00_A0D8		
	Pre-existing features, 0xE401	0x1E00_A0DC	R/W	32
	Pre-existing features	0x1E00_A0E0	R/W	32
	Reserved, p2gBase3[35:32], 0x0002	0x1E00_A0E4	R/W	32
	p2gBase3[31:0]	0x1E00_A0E8	R/W	32
	Reserved	0x1E00_A0EC - 0x1E00_A0FF		
pgbCSR	PGB Control and Status Register	0x1E00_A100	R/W	64
g2pLower0	g2pwindow Lower Address Register 0	0x1E00_A108	R/W	64
g2pUpper0	g2pwindow Upper Address Register 0	0x1E00_A110	R/W	64
g2pLower1	g2pwindow Lower Address Register 1	0x1E00_A118	R/W	64

The following table is a register map of the individual modules. Please note that this table is still under construction. See the tables in each relevant chapter for more information.

Name	Register Description	Address	R/W	Size(b)
g2pUpper1	g2pwindow Upper Address Register 1	0x1E00_A120	R/W	64
g2pLower2	g2pwindow Lower Address Register 2	0x1E00_A128	R/W	64
g2pUpper2	g2pwindow Upper Address Register 2	0x1E00_A130	R/W	64
g2pLower3	g2pwindow Lower Address Register 3	0x1E00_A138	R/W	64
g2pUpper3	g2pwindow Upper Address Register 3	0x1E00_A140	R/W	64
g2pBase0	g2pwindow Base Address Register 0	0x1E00_A148	R/W	64
g2pBase1	g2pwindow Base Address Register 1	0x1E00_A150	R/W	64
g2pBase2	g2pwindow Base Address Register 2	0x1E00_A158	R/W	64
g2pBase3	g2pwindow Base Address Register 3	0x1E00_A160	R/W	64
g2pCycleType	g2pwindow Cycle Type Register	0x1E00_A168	R/W	64
p2gBase0	p2gwindow Base Address Register 0	0x1E00_A170	R/W	64
p2gBase1	p2gwindow Base Address Register 1	0x1E00_A178	R/W	64
p2gBase2	p2gwindow Base Address Register 2	0x1E00_A180	R/W	64
p2gBase3	p2gwindow Base Address Register 3	0x1E00_A188	R/W	64
Ia	Failing Transaction Address Register	0x1E00_A190	R/W	64
-	RESERVED	0x1E00_A198 - 0x1E00_AFFF		

5. C790 Processor Core

This chapter is an overview of the C790 Processor Core.

5.1 Features

The C790 is an integrated, high-performance, RISC processor core with 128-bit internal data paths optimized for high data throughput.

Some of the C790 features are listed below:

- 2-way super-scalar pipeline with 128-bit (2x64-bit) data path
- 200/266 MHz operation
- MIPS I, II, III compatible ISA, with selected MIPS IV ISA (Pre-fetch and Move Conditional Instruction)
- Additional multimedia instruction set support to provide SIMD operation
- 32 KB, 2-way set associative Instruction Cache and 32 KB, 2-way set associative Data Cache
- Supports the data cache line lock, write back cache, and non-blocking load functions, and the prefetch instruction.
- 64-entry fully associative branch target address cache
- 48-entry fully associative JTLB supporting 4 KB –16 MB page size
- Supports 32-bit physical address
- IEEE-754 compatible, double-precision FPU is coupled with the C790.
- Supports bi-Endian (Little Endian and Big Endian) operation

5.2 Block Diagram and Functional Block Descriptions

This section shows a block diagram of the C790 and summarizes the modules' functionality.

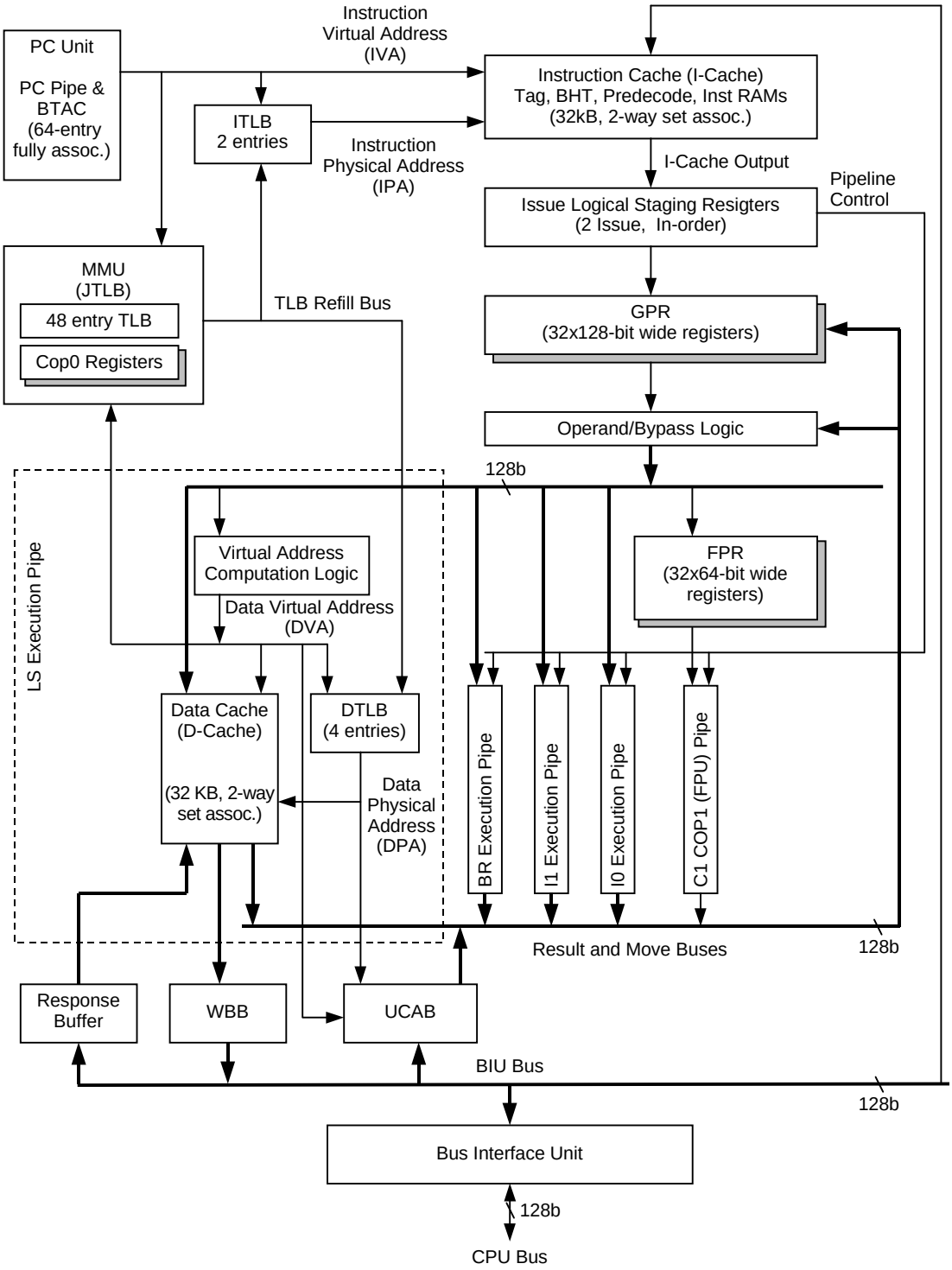


Figure 5-1 C790 Block Diagram

- **PC Unit:** The 32-bit *Program Counter* (*PC*) holds the address of the instruction that is being executed. It also contains a 64-entry **Branch Target Address Cache (BTAC)** which stores branch target addresses used for branch prediction (to eliminate branch penalties).

- **Issue Logic and Staging Registers:** The issue logic controls the transfer of fetched instructions to the appropriate execution pipes. It can issue a maximum of two instructions per cycle, and any instructions remaining that were fetched but could not be issued because of conflicts such as resource conflicts or hazards are held in staging registers until they can be issued.

- **General Purpose Registers (GPRs):** The width of the GPRs is extended from MIPS III's 64 bits, to a width of 128 bits. The upper 64 bits of the GPRs are accessible using the quad word Load/Store instructions, the quad word funnel shift instruction, and the parallel (multimedia) instructions.

- **I0 and I1 Pipes:** The two integer pipelines I0 and I1 each contain a complete 64-bit ALU, Shifter, and Multiply-Accumulate (MAC) unit. The I0 pipeline additionally contains a Shift Amount (SA) register that is used for funnel shift operations, and the I1 pipeline contains a leading zero counter. The two 64-bit data paths can be configured dynamically, on an instruction-by-instruction basis, into a single 128-bit data-path when it is necessary to execute 128-bit wide multimedia, shift, ALU or Multiply-Accumulate instructions. The two 64-bit data paths share a single 128-bit multimedia shifter during 128-bit wide shift operations.

- **Load / Store (LS) Pipe:** The Load/Store (LS) pipe supports a single issue of Load and Store instructions at widths ranging from one byte (8 bits), to one quad-word (128 bits).

- **Memory Management Unit (MMU):** The Memory Management Unit supports the address translation functions of the C790. It contains a 48-entry fully associative JTLB, a 2-entry Instruction Translation Lookaside Buffer (ITLB), and a 4-entry Data Translation Lookaside Buffer (DTLB).

- **Memory Caches:** The C790 includes an Instruction Cache and a Data Cache. For each branch instruction present in the instruction cache, two bits of branch history information are stored in a **Branch History Table (BHT)**.

- **Response and Write-back Buffer:** The Write-back Buffer (WBB) is an 8-entry by 16-byte (one quad-word) FIFO queues up stores prior to accessing the C790 bus. It increases C790 performance by isolating the processor from the latencies of the C790 bus. It is also used during the gathering operation of uncached accelerated stores. Sequential stores less than a quad-word in length are gathered in the WBB, thereby improving bus bandwidth usage.

- **Uncached Accelerated Buffer (UCAB) :** The Uncached Accelerated Buffer (UCAB) is a 2-entry by 4 quad-word buffer. It caches 128 sequential bytes of data during an uncached accelerated load miss. Subsequent loads from the uncached accelerated address space get their data from this buffer if the address hits in the UCAB, thereby eliminating bus latencies and providing higher performance.

- **Bus Interface Unit (BIU) :** The Bus Interface Unit (BIU) connects the core's internal bus to the C790 bus. It interfaces the core's internal bus signals to the C790 Bus.

The C790 extends the normal MIPS-compatible register set by extending the width of the general purpose registers (GPRs) from 64 bits to 128 bits. It also incorporates an additional pair of HI/LO registers for the I1 pipe, and the SA register for funnel shift instructions.

5.3 C790 Registers

The C790 has 128-bit wide GPRs. The upper 64 bits of the GPRs are only used by the C790-specific “Quad Load/Store”, and “Multimedia (Parallel)” instructions.

HI1 and LO1, which are the upper 64 bits of each of the 128-bit HI and LO registers, are also used by new multiply and divide instructions such as MULT1, MULTU1, DIV1, DIVU1, MADD1, MADDU1, MFHI1, MFLO1, MTHI1, and MTLO1, which are non-parallel I1 pipeline-specific instructions. They are also used by multimedia (parallel) multiply and divide instructions.

5.4 FPU Registers

The floating-point unit (COP1) has thirty-two 64-bit wide floating-point registers. It also contains two floating-point control registers.

5.5 Memory Management

The C790 provides a memory management unit (MMU) which uses an on-chip translation look-aside buffer (TLB) to translate virtual addresses into physical addresses.

Features

- MIPS III-compatible 32-bit MMU
- Operating Modes: User, Supervisor, and Kernel
- TLB: 48 entries of even/odd page pairs (96 pages)
Fully associative
- Page Size: 4 KB, 16 KB, 64 KB, 256 KB, 1 MB, 4 MB, 16 MB
- ITLB: 2 entries
- DTLB: 4 entries
- Address Sizes: Virtual Address Size = 32 bits
Physical Address Size = 32 bits

5.6 Cache Memory

The C790 contains an instruction cache and a separate data cache.

Features

- Separate Instruction Cache and Data Cache
- Caches are virtually indexed and physically tagged
- Write-back policy for the Data Cache
- Cache Size: Instruction Cache: 32 KB
Data Cache: 32 KB
- Line size: 64 Bytes
- Associativity: 2-way set-associative
- Write Policy: Write-back and write allocate

- Data order for block reads: Sequential ordering
- Data order for block writes: Sequential ordering
- Instruction cache miss restart: After all data are received
- Data cache miss restart: Early restart on first quad-word
- Cache parity: No
- Cache Locking: Data Cache Line Lock
  Controlled by CACHE instruction
- Cache Snooping: No
- Non-blocking load: Yes
- Hit Under Miss: Yes (Supports multiple hits under one miss)
- Data Cache Pre-fetch: Yes

5.7 Floating Point Unit

The floating-point unit implements double-precision and single-precision operations. The unit is IEEE-754 compatible.

Features:

- MIPS III floating point instructions
- High performance double-precision floating point unit tightly coupled to the C790
- Supports double-precision and single-precision formats as defined in the IEEE-754 specifications
- Compatible with the TX49 FPU
- No hardware support for denormalized numbers

5.8 Performance Monitor

The performance monitor provides the means for gathering statistical information about the internal events of the C790 and its pipeline during program execution. The statistics gathered during program execution aid in tuning the performance of hardware and software systems based on the processor core.

The performance monitor consists of one control register and two counters. The control register controls the functions of the monitor while the counters count the number of events specified by the control register.

Features:

- Two performance counter registers
- Can count over twenty different events within the C790
- Counting can be selectively enabled in User, Supervisor, Kernel, and Exception modes

5.9 Debug Functions

The C790 supports ranged hardware break pointers with mask registers. This makes it possible to debug with less observational impact. Note that C790 debugging also supports software debugging using the BREAK instruction as defined in MIPS ISA.

Features:

- One Instruction Address Breakpoint register
- One Instruction Address Breakpoint Mask register
- One Data Address Breakpoint register
- One Data Address Breakpoint Mask register
- One Data Value Breakpoint register
- One Data Value Breakpoint Mask register
- Each breakpoint is individually enabled
- Breakpoint function can be selectively enabled in User, Supervisor, Kernel, and Exception modes
- External Trigger signal can be generated when breakpoint occurs
- 11 signals are used to provide real-time PC tracing functionality

6. SDRAM Memory Controller

6.1 Overview

This SDRAM Controller is used to connect the C790 (128-bit MIPS CPU) to SDRAM. The SDRAM devices that can be connected are 64 Mb, 128 Mb, or 256 Mb with a 4-bank architecture. If 8M x 8 x 4 SDRAM chips are used, the TX7901 can support 1 GB memory with 4 physical memory banks.

6.2 Features

- Directly connected to a 128-bit C790 sysbus operating up to 133 MHz
- Supports PC100/133 DIMM
- Supports 4-bank interleaving for 64/128/256 Mbit SDRAMs, or 2-bank interleaving for 16 Mbit
- ECC, single-bit error correction, double-bit error detection
- Maximum access rate of 133 MHz
- Supports aligned (8 quad-word) burst transfers

6.3 Address Space Decoding

The SDRAM Controller has a fully programmable address map. It uses a two-stage decoding process where major device regions are decoded first, and then the individual devices are sub-decoded. Addresses for regions in 256 MB units are compared by exact matching, and individual device addresses are compared by size comparison. One device (DIMM) therefore cannot span across 256 MB boundaries.

6.4 Two-Stage Decoding Process

Physical space is divided into 16 regions. Each region can have 256 MBytes of address space. Memory space decoding starts with the sysAddr address being compared with the values in the four LOW and HIGH Registers. Device0, device1, device2, and device3 correspond to sdrCSB[0], sdrCSB[1], sdrCSB[2], sdrCSB[3.]

Address decoding is performed as follows:

- sysAddr[31:28] are compared against region field [31:28] in the LOW Decoder

registers. This value must match exactly. This value effectively sets a 256 MB region.

- sysAddr[27:20] are compared against bits [27:20] in the LOW Decoder registers. This value of sysAddr must be greater than or equal to the LOW decode value. This describes the low boundary for the region.
- sysAddr[27:20] are compared against bits [27:20] in the HIGH Decoder registers. The value of sysAddr must be less than or equal to the HIGH decode value. This describes the high boundary for the region.
- If all of the above are true, then the device region is selected and the corresponding chip select signal is activated.

Any device region can be disabled by setting the value of the "LOW" decoder to be higher than that of the "HIGH" decoder.

The LOW and HIGH Decode Registers cannot be programmed in the region from 0x_1E00_0000 to 0x_20FF_FFFF. This is reserved for TX7901 registers and Boot Devices. It is important to note that devices never span across region boundaries. This is detected by reading a HIGH Register after a write. Two devices may be put in one region, but they must not overlap.

Examples of the two-stage decoding process are shown in Figure 6-1 and Figure 6-2.

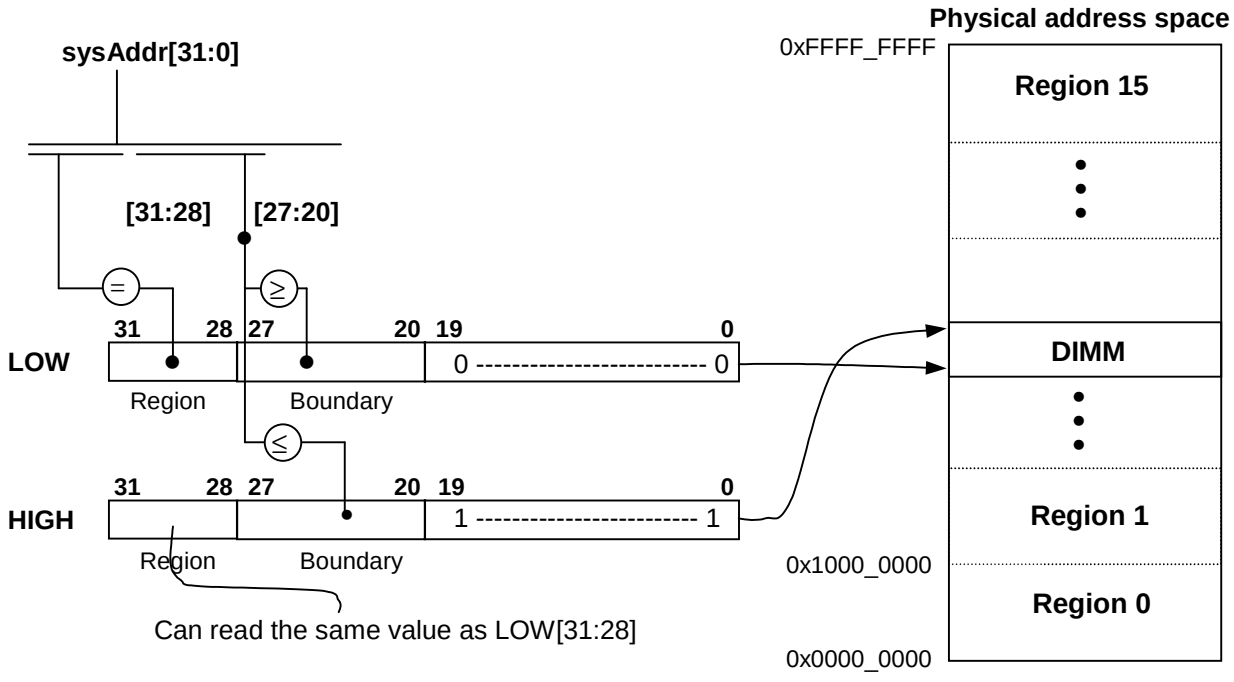


Figure 6-1 Two-stage Decoding

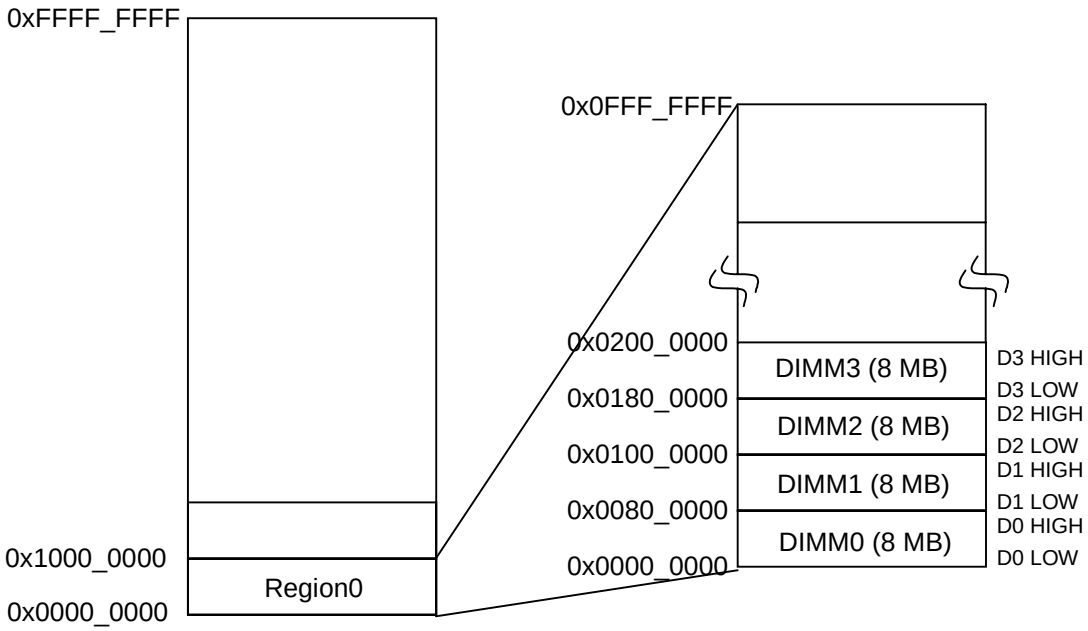


Figure 6-2 Initial Setting after Reset

Programming note

It is okay to place multiple DIMMs into a region. It is okay to place two DIMMs into consecutive physical addresses. However, it is not okay to place a DIMM across two regions. In other words, placing a DIMM across region boundaries is not permitted. This violation can be detected by reading the HIGH register again after LOW and HIGH are written. If the HIGH register is read the same as it was written, then it is okay. If it is different because the HIGH region field is the same as that of LOW, then it is okay to place a DIMM across the 256 MB region boundary.

```
la  r4, <LOW of DIMM0>
la  r5, <HIGH of DIMM1>
la  r7, <SDRAM base address>

sq  r4, D0LOW (r7)
sq  r5, D0HIGH (r7)
lq  r6, D0HIGH (r7)
bne  r5, r6, crossing DIMM
```

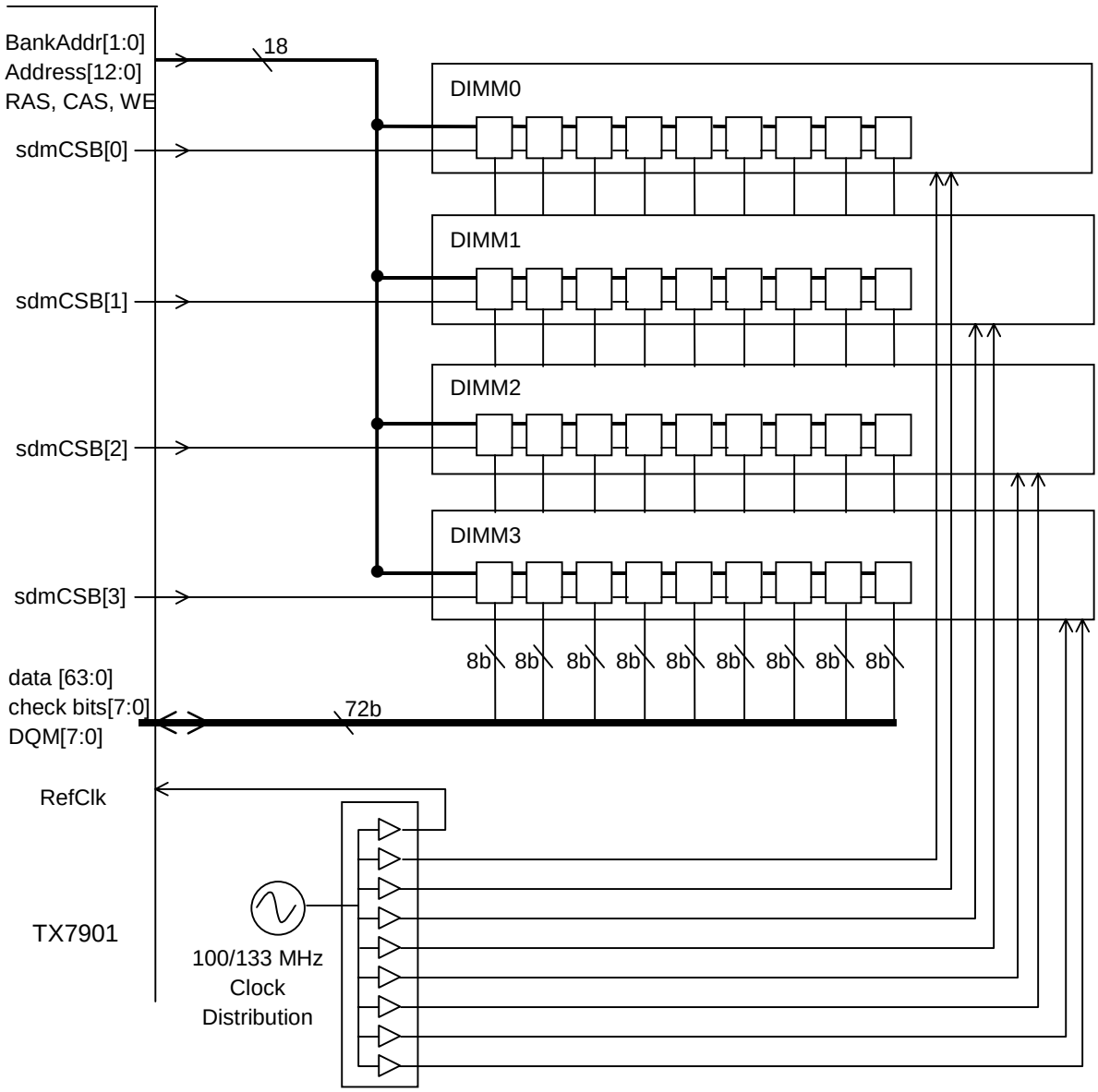


Figure 6-3 Example Connection of Single-sided DIMMs

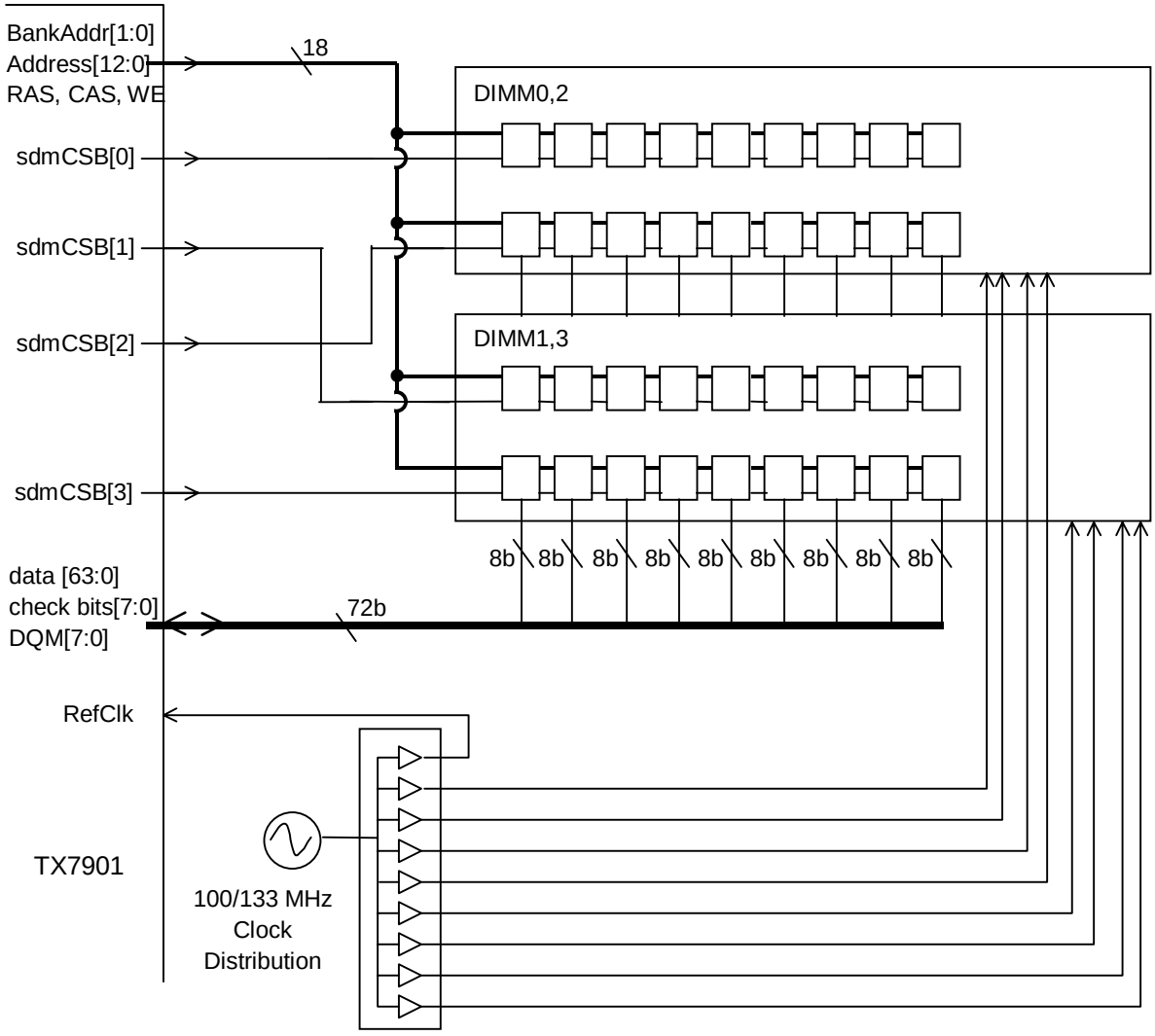


Figure 6-4 Example Connection of Double-sided DIMMs

6.4.1 Default Memory Map

After sysResetB is active, the default memory map is as shown in Table 6-1.

Table 6-1 Initial Values after Reset

Device	CS	LOW		HIGH		Initial Size
DIMM0	sdrCSB[0]	D0LOW	(0x0 000_0000)	D0HIGH	(0x0 07F_FFFF)	8 MB
DIMM1	sdrCSB[1]	D1LOW	(0x0 080_0000)	D1HIGH	(0x0 0FF_FFFF)	8 MB
DIMM2	sdrCSB[2]	D2LOW	(0x0 100_0000)	D2HIGH	(0x0 17F_FFFF)	8 MB
DIMM3	sdrCSB[3]	D3LOW	(0x0 180_0000)	D3HIGH	(0x0 1FF_FFFF)	8 MB

Table 6-2 below shows an example of four 64 MB DIMMs.

Table 6-2 Example Values for Four DIMMs

Device	CS	LOW		HIGH		Initial Size
DIMM0	sdrCSB[0]	D0LOW	(0x0 000_ 0000)	D0HIGH	(0x0 3FF_FFFF)	64 MB
DIMM1	sdrCSB[1]	D1LOW	(0x0 400_0000)	D1HIGH	(0x0 7FF_FFFF)	64 MB
DIMM2	sdrCSB[2]	D2LOW	(0x0 800_0000)	D2HIGH	(0x0 BFF_FFFF)	64 MB
DIMM3	sdrCSB[3]	D3LOW	(0x0 C00_0000)	D3HIGH	(0x0 FFF_FFFF)	64 MB

6.4.2 Example connection of DIMMs

It is possible to connect up to four single-sided DIMMs. See Figure 6-3 above.

It is possible to connect up to two double-sided DIMMs. See Figure 6-4 above.

Double-sided DIMM is considered as two DIMMs since it has two Chip Select inputs. See the Application Notes (to be released) for more information regarding the clocking.

6.5 Registers

The following table is a register map of the SDRAM Memory Controller Module.

Table 6-3 List of SDRAM Memory Controller Registers

Name	Register Description	Address	R/W	Size(b)
SDRAM Memory Controller, Base Address 0x1E00_0000				
D0PR	DIMM 0 Parameters Register	0x1E00_0000	R/W	128
D1PR	DIMM 1 Parameters Register	0x1E00_0010	R/W	128
D2PR	DIMM 2 Parameters Register	0x1E00_0020	R/W	128
D3PR	DIMM 3 Parameters Register	0x1E00_0030	R/W	128
DOMR	Operation Mode Register	0x1E00_0040	R/W	128
DEMR	ECC Mode Register	0x1E00_0050	R/W	128
DEESR	ECC Error Status Register	0x1E00_0060	R	128
DEEAR	ECC Error Address Register	0x1E00_0070	R	128
–	RESERVED	0x1E00_0080		
DREFRESH	Refresh Register	0x1E00_0090	R/W	128
DDRIVE	SDRAM Interface Output Drive- Strength Control Register	0x1E00_00A0	R/W	128
D0LOW	DIMM 0 LOW Address Decode	0x1E00_0100	R/W	128
D0HIGH	DIMM 0 HIGH Address Decode	0x1E00_0110	R/W	128
D1LOW	DIMM 1 LOW Address Decode	0x1E00_0120	R/W	128
D1HIGH	DIMM 1 HIGH Address Decode	0x1E00_0130	R/W	128
D2LOW	DIMM 2 LOW Address Decode	0x1E00_0140	R/W	128
D2HIGH	DIMM 2 HIGH Address Decode	0x1E00_0150	R/W	128
D3LOW	DIMM 3 LOW Address Decode	0x1E00_0160	R/W	128
D3HIGH	DIMM 3 HIGH Address Decode	0x1E00_0170	R/W	128

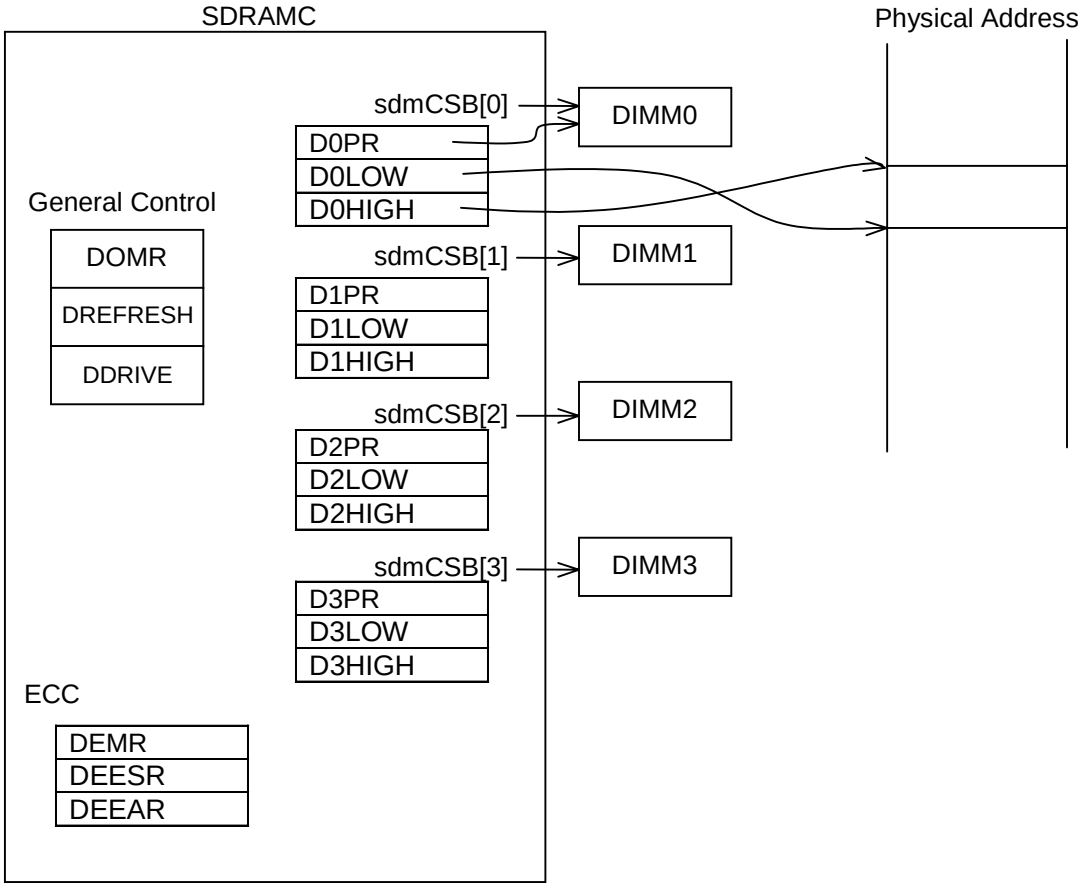
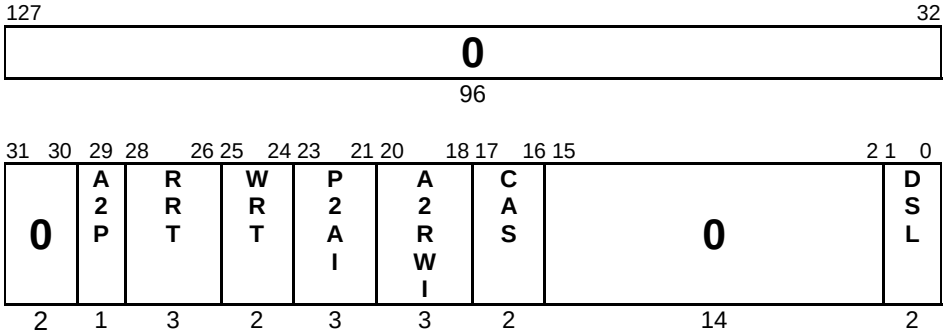


Figure 6-5 SDRAM Registers

All of the following registers are 128 bits wide and are aligned to 16 Byte boundaries. In order to facilitate Bi-Endian programming, it is strongly recommended to use lq/sq to access these registers.

6.5.1 Parameters register



This register contains parameters that are used for each of the physical memory devices. All memory devices share the same timing parameters (bits[29:16]) in the Device Region 0 Parameters Register.

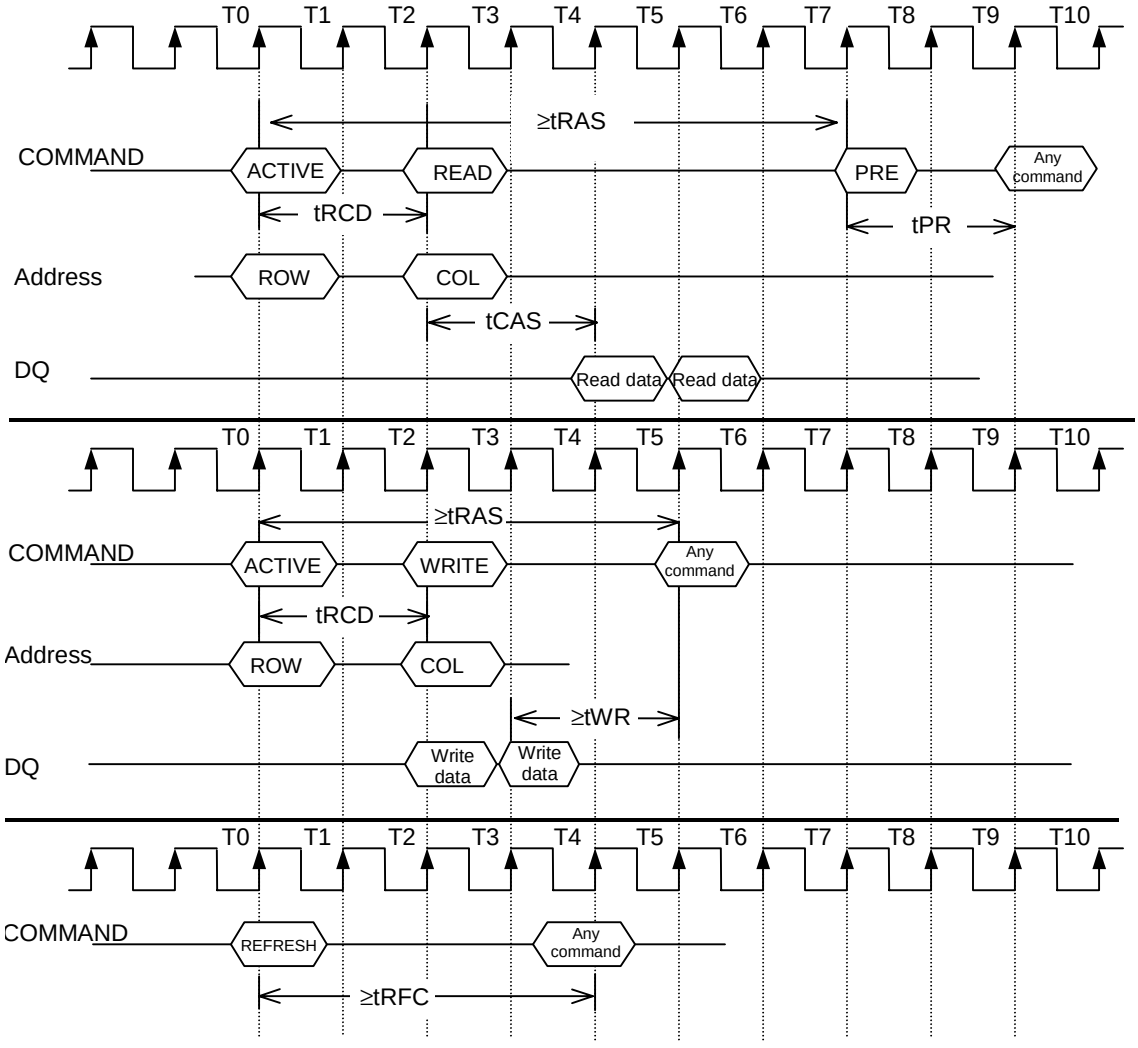
DIMM 0 Parameters Register (0x1E00_0000) R/W

Field	Bit(s)	Description	Timing Symbol
DSL	1:0	Device Select (01) 00: 16 Mb SDRAM 01: 64/128 Mb SDRAM 10: 256 Mb SDRAM 11: Reserved	–
–	15:2	Reserved	–
CAS	17:16	CAS Latency (11) These bits specify the timing for the first read data after SDRAM samples a column address. 00, 01: Reserved 10: 2 clock cycles 11: 3 clock cycles	tCAS
A2RWI	20:18	Active to Read/Write Interval (011) Specifies the earliest timing for a READ/WRITE command after an ACTIVE command. 000, 001 : Reserved 010 : 2 clock cycles 011 : 3 clock cycles 100 - 111 : Reserved	tRCD
P2AI	23:21	Precharge to active interval (011) These bits specify the earliest timing for a new command after a PRECHARGE command. 000, 001 : Reserved 010 : 2 clock cycles 011 : 3 clock cycles 100 - 111 : Reserved	tRP
WRT	25:24	Write recovery time (10) These bits specify the earliest timing for a PRECHARGE command after the last data were written to the SDRAM. 01 : 1 clock cycle 10 : 2 clock cycles 00, 11 : Reserved	tWR

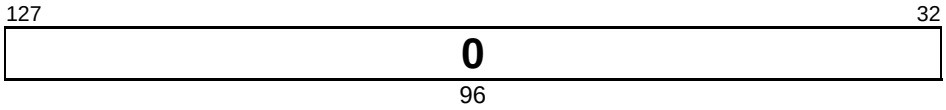
Field	Bit(s)	Description	Timing Symbol
RRT	28:26	Refresh Recovery Time (110) These bits specify the earliest timing for a new command after a Refresh command. 001 : 5 clock cycles 010 : 6 clock cycles 011 : 7 clock cycles 100 : 8 clock cycles 101 : 9 clock cycles 110 : 10 clock cycles 111 : 16 clock cycles 000 : Reserved	tRFC
A2P	29	Activate to Precharge (1) This bit specifies the earliest timing for a PRECHARGE command after an ACTIVE command. 0 : 5 clock cycles 1 : 6 clock cycles	tRAS
–	31:30	Reserved	–

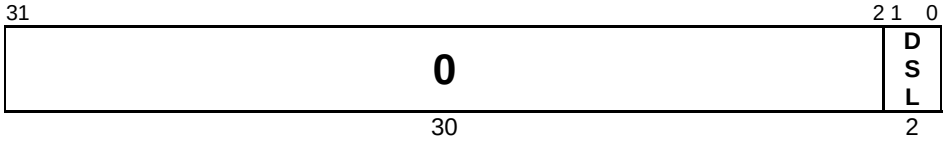
The following figure shows example timing parameters.

Figure 6-6 Example Timing Parameters



DIMM 1 Parameters Register (0x1E00_0010) R/W

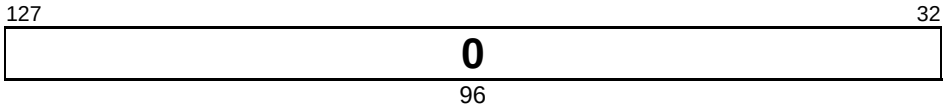


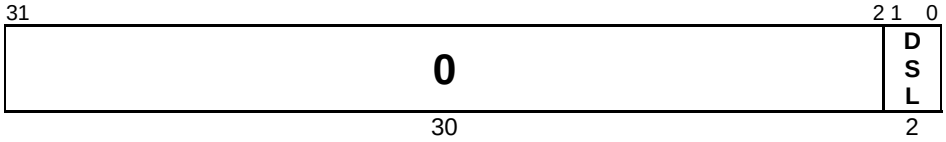


Field	Bit	Description
DSL	1:0	Device Select (01) 00: 16 Mb SDRAM 01: 64/128 Mb SDRAM 10: 256 Mb SDRAM 11: Reserved
–	31:2	Reserved

Other parameters use the same values as DIMM 0.

DIMM 2 Parameters Register (0x1E00_0020) R/W

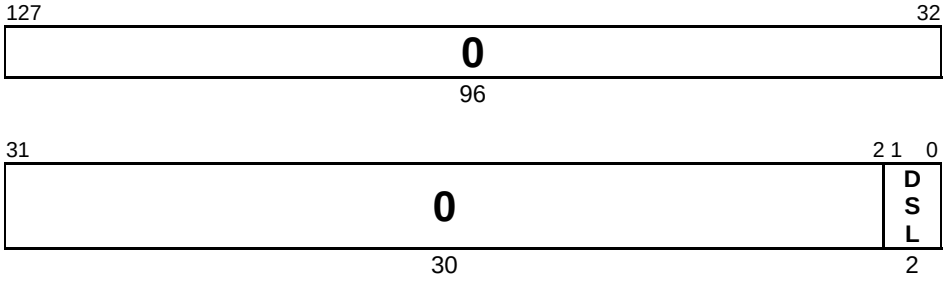




Field	Bit	Description
DSL	1:0	Device Select (01) 00: 16 Mb SDRAM 01: 64/128 Mb SDRAM 10: 256 Mb SDRAM 11: Reserved
–	31:2	Reserved

Other parameters use the same values as DIMM 0.

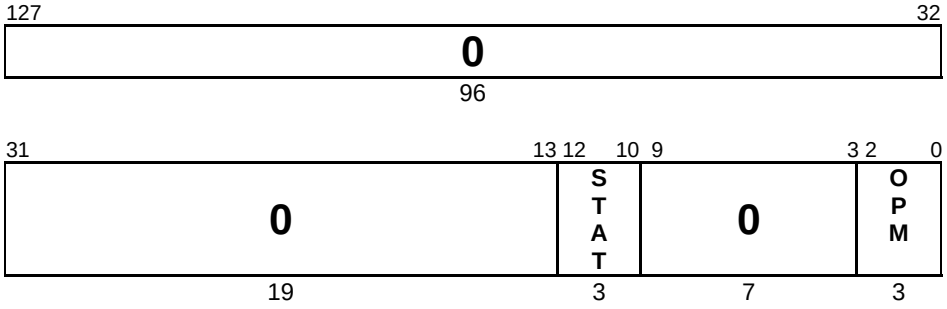
DIMM 3 Parameters Register (0x1E00_0030) R/W



Field	Bit	Description
DSL	1:0	Device Select (01) 00: 16 Mb SDRAM 01: 64/128 Mb SDRAM 10: 256 Mb SDRAM 11: Reserved
–	31:2	Reserved

Other parameters use the same values as DIMM 0.

6.5.2 Operation Mode Register (0x1E00_0040) R/W



This register is used to execute commands other than standard memory reads and writes to the SDRAM. Bits [12:10] are used to check the current state of the SDRAM Controller.

Field	Bit(s)	Description
OPM	2:0	Operation Mode (001) 000 : Normal SDRAM Mode (Read/Write) 001 : NOP Commands 010 : Precharge All Banks 011 : Writing to the SDRAM Mode Register. Each DIMM could have a different Mode. 100 : Force a Refresh Cycle Others : Reserved
–	9:3	Reserved (0)
Stat	12:10	Status (read only) 000 : Idle 001 : Page Activation 010 : Precharge 011 : Write Mode Register 100 : Refresh 101 : Read 110 : Write 111 : Reserved
–	31:13	Reserved (0)

In order to execute one of the above commands on the SDRAM, the following procedure should occur:

1. The corresponding value should be written to the SDRAM Operation Mode Register.

2. OMR Write should be followed by a dummy write to the corresponding SDRAM. For Mode Register Write, the RAS address [12:0] will be put in the Mode Register. To map sysADDR into the RAS address, please see 6.6 Address Mapping.

3. SDRAM and SDRAM Controller initialization should be complete before writing 000 to this register in order to place it back into the Normal SDRAM mode. Normal SDRAM operation can then start.

6.5.2.1 Normal SDRAM Mode

0x0 should be written to the SDRAM Operation Mode Register to enable normal reading and writing to the SDRAM.

Note: SDRAM and SDRAM Controller must be complete before entering this mode.

6.5.2.2 NOP Commands

NOP commands are used to issue NOPs to SDRAM when the DIMM is accessed. This prevents unwanted commands from being registered during idle or wait states of the initialization sequence.

6.5.2.3 Precharge All Banks

The Precharge All Banks command is used to deactivate the open row. The Precharge All Banks command is the first command called after reset. In this mode, any write to a particular DIMM causes the Precharge command to be issued. Once a bank has been precharged, it is in the idle state and must be activated prior to any read or write commands being issued to that bank. This sequence will be performed by the hardware sequencer.

6.5.2.4 Writing to the SDRAM Mode Register

Each DIMM has its own Mode Register. The Mode Register is used to define the specific mode of operation for the SDRAM. This definition includes the selection of a burst length, sdrCAS* latency, burst type, operating mode, etc. (Please see your SDRAM data sheet for more information about this register.) Typically, the Mode Register of each SDRAM is initialized during system boot-up and is kept static.

The parameter that the SDRAM Controller can change is the CAS latency. The burst length must be programmed to 2. The bust type is Sequential. The Write Burst Mode is the Programmed Burst Length. In order to change this parameter in the SDRAM's Mode Register:

1. The DIMM Parameter Registers are updated properly. DIMMs are precharged, deactivated.

2. The SDRAM Operation Mode Register should be written to 0x3 to indicate a Write Command to the SDRAM Mode Register.

3. Store dummy data (32-bit) to a location. The address of this store instruction is saved in the Mode Register as data. The address is shuffled by address mapping. The corresponding address bit may vary according to the SDRAM chip and DIMM connection. SDRAMC uses RA[12:0] of sysAddr as a Mode Register value.

Table 6-4 SDRAM Mode Register Settings

Burst Length	2
Burst Type	Sequential
Write burst mode	Programmed burst length
CAS Latency	2 or 3

6.5.2.5 Force Refresh

The Force Refresh Command is used to execute a refresh cycle. In this mode, any write to a DIMM causes the Auto Refresh command to be called. The lower address and data are ignored because the Auto Refresh command uses a refresh counter internal to the SDRAM chip. At least eight Auto Refresh commands are required for the power on sequence.

6.5.2.6 Initialization sequence

Intel's "PC SDRAM Specification" Rev. 1.7, November 1999 recommends the sequence described below.

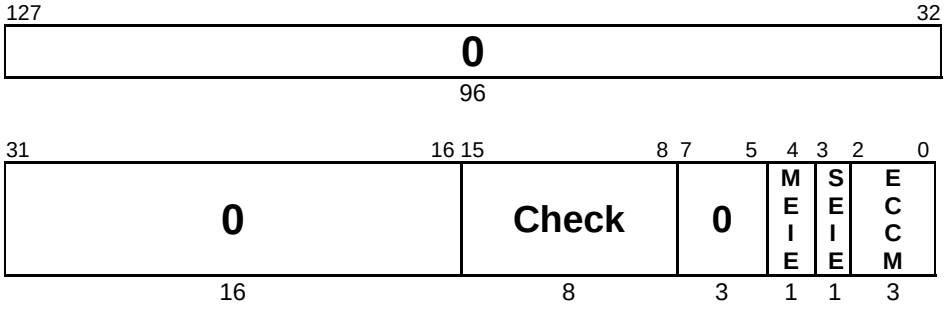
Following the initialization sequence, the device must be ready for full functionality. SDRAM devices are initialized by the following sequence:

1. At least one NOP cycle will be issued after the 1msec device deselect.
2. A minimum pause of 200 μsec will be provided after the NOP.
3. A precharge all command will be issued to the SDRAM.
4. Eight Auto Refresh (CBR) refresh cycles will be provided.
5. A mode register set cycle will be issued to program the SDRAM parameters (e.g., Burst length, CAS# latency, etc.).

6.5.2.7 Important programming note

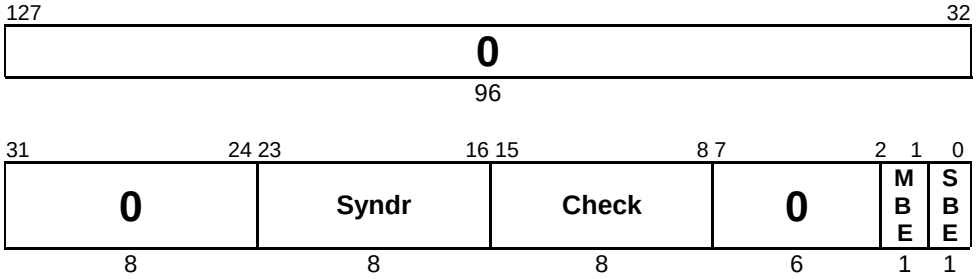
Program codes that change any SDRAM/SDRAMC parameters should be located in a memory device other than SDRAM memory (boot ROM, for example). Otherwise, such program codes may cause a deadlock.

6.5.3 ECC Mode Register (0x1E00_0050) R/W



Field	Bit(s)	Description
ECCM	2:0	Error Correction Check Mode (000) 000 : ECC Disable Mode, no check bit generation 001 : Detect mode. Performs check bit generation during memory writes and error detection only during memory reads. 010 : ECC Enable Mode. Performs check bit generation during memory writes and error detection and correction during memory reads. 011 : Diagnostic Mode for verifying the ECC function. All check bits are forced to check bits in this register during memory writes.
SEIE(3)	3	Single ECC Error Interrupt Enable (0)
MEIE(4)	4	Multiple ECC Error Interrupt Enable (0) Interrupt occurs when an ECC error is detected and this bit is set to “1.”
–	7:5	Reserved (0)
Check	15:8	Check bits [7:0] to write (Diagnostic Mode)
–	31:16	Reserved (0)

6.5.4 ECC Error Status Register (read only) (0x1E00_0060)

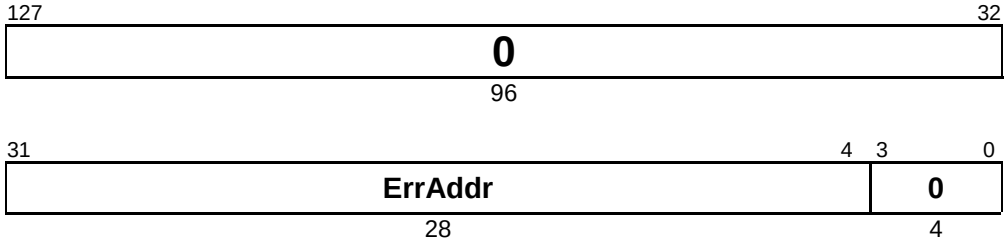


When there is an ECC Error (single or double bit error), the failing status is stored in this register and an interrupt is generated.

After an ECC Error, the ECC Error Status Register and the ECC Error Address Register keep the status and address of the latest error until it has been read. SBE and MBE are cleared after it is read. Regardless of the ECC Interrupt Enable Bit in the ECC Mode Register, these registers are updated when an ECC error is detected.

Field	Bit(s)	Description
SBE	0	Single Bit Error (SBE)
MBE	1	Multiple Bit Error (MBE)
–	7:2	Reserved
Check	15:8	Check bits [7:0]
Syndr	23:16	Syndrome bits [7:0]
–	31:24	Reserved

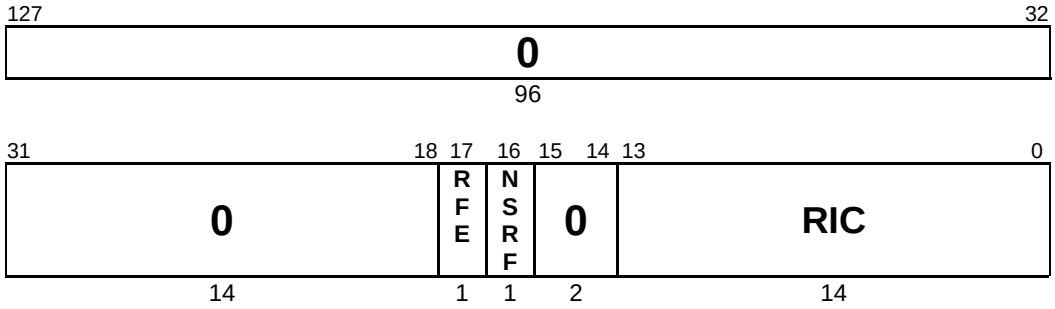
6.5.5 ECC Error Address Register (read only) (0x1E00_0070)



When there is an ECC Error, the failing address is stored in this register. This register keeps the error address of the latest ECC error.

Field	Bits	Description
ErrAddr	31:4	Error Address [31:4]
–	3:0	Reserved

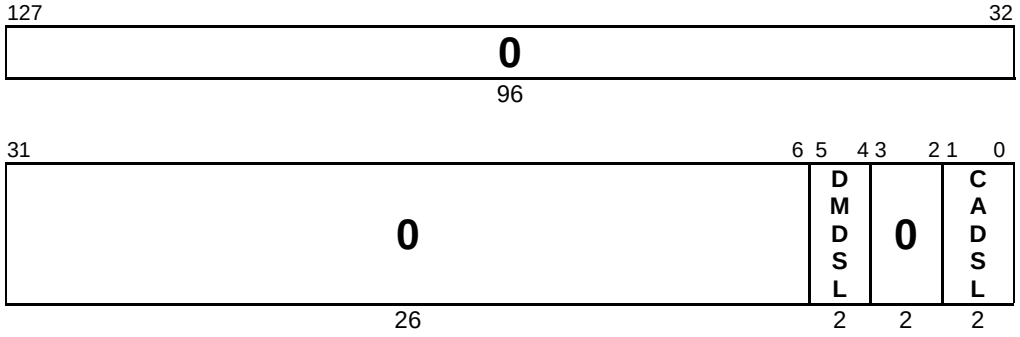
6.5.6 Refresh Register (0x1E00_0090) R/W



This register contains parameters that are used for all SDRAMs with the SDRAM controller.

Field	Bit(s)	Description
RIC	13:0	Refresh Interval Count Value (0x0400) Implements standard CAS before refreshing RAS. Refresh rates for all banks can be programmed in this register. RefIntCnt is a 14-bit counter. If the default value is 0x400 for example and if the clock is 100 MHz, then a refresh sequence will occur every 10 μs. (10 ns × 1024) = 10.24 μs
–	15:14	Reserved (0)
NSRF	16	Non-Staggered Refresh (0) In the non-staggered refresh mode, this bit is set and sdrCSB[3:0] will be active to simultaneously refresh all banks. Staggered refresh is used to refresh banks in order sequentially.
RFE	17	Refresh Enable (0) SDRAM refresh will be enabled when this bit is set, and will be disabled when this bit is reset. This bit defaults to 0 (reset) after chip reset. This bit should be enabled after BIOS finishes the SDRAM initialization and initializes all other control registers in the SDRAM controller.
–	31:18	Reserved (0)

6.5.7 SDRAM Interface Output Drive-Strength Control Register (0x1E00_00A0) R/W

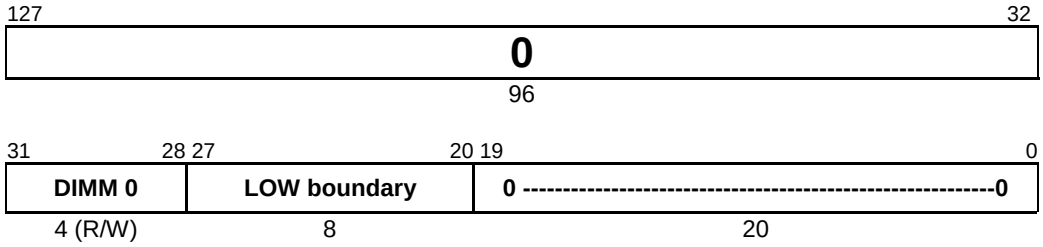


This register contains parameters which are used to select the SDRAM interface control/address (i.e. CSB[3:0], CKE, RASB, CASB, WEB, BA[1:0], and AD[12:0]) and data/data-mask (i.e. CB[7:0], DQ[127:0], and DQM[15:0]) output drive-strength.

Field	Bits	Description
CADSL	1:0	Control/Address output drive-strength select (11) 00 : 8 mA 01 : 16 mA 10 : 24 mA 11 : 32 mA
–	2:3	Reserved (0)
DMDSL*	5:4	Data/Data-mask output drive-strength select (10) 00 : 8 mA 01 : 16 mA 10 : 24 mA 11 : 32 mA
–	31:6	Reserved

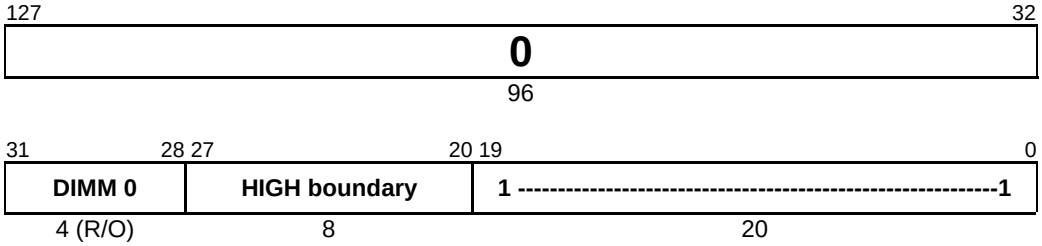
Note: * Currently 16 mA fixed drivers are implemented for Data/Data-mask output due to I/O area limitation. DMDSL doesn't affect drive-strength of Data/Data-mask output drivers.

6.5.8 DIMM 0 LOW Address Decode (0x1E00_0100)



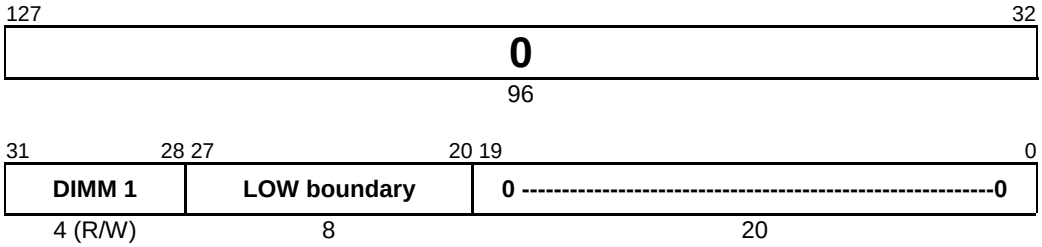
Bits	R/W	Description
31:28	R/W	DIMM 0
27:20	R/W	LOW boundary
19:0	R/O	0

6.5.9 DIMM 0 HIGH Address Decode (0x1E00_0110)



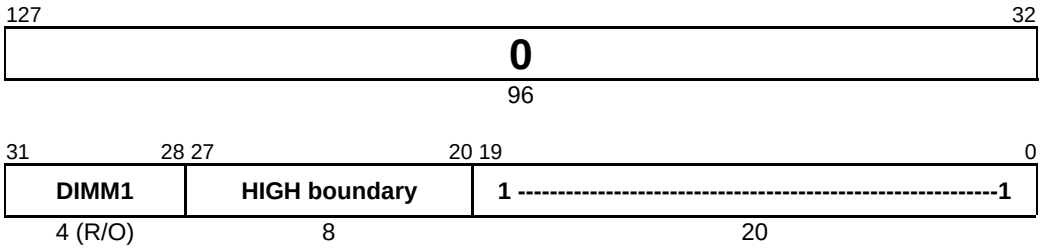
Bits	R/W	Description
31:28	R/O	DIMM 0
27:20	R/W	HIGH Boundary
19:0	R/O	1

6.5.10 DIMM 1 LOW Address Decode (0x1E00_0120)



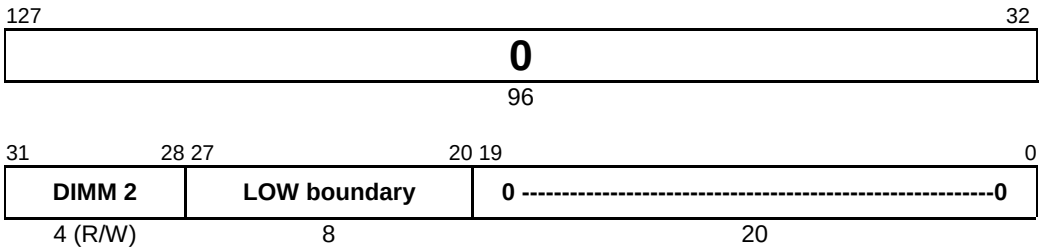
Bits	R/W	Description
31:28	R/W	DIMM1
27:20	R/W	LOW boundary
19:0	R/O	0

6.5.11 DIMM1 HIGH Address Decode (0x1E00_0130)



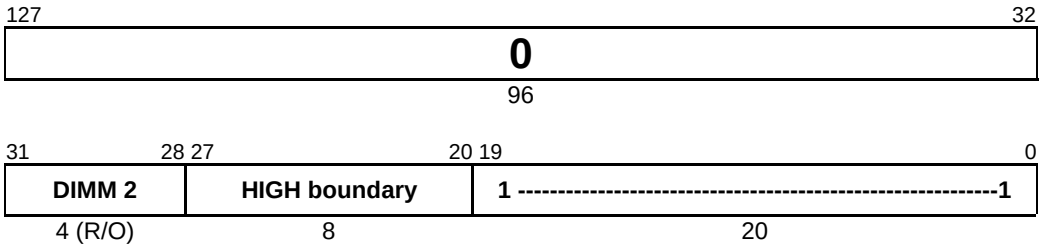
Bits	R/W	Description
31:28	R/O	DIMM1
27:20	R/W	HIGH Boundary
19:0	R/O	1

6.5.12 DIMM 2 LOW Address Decode (0x1E00_0140)



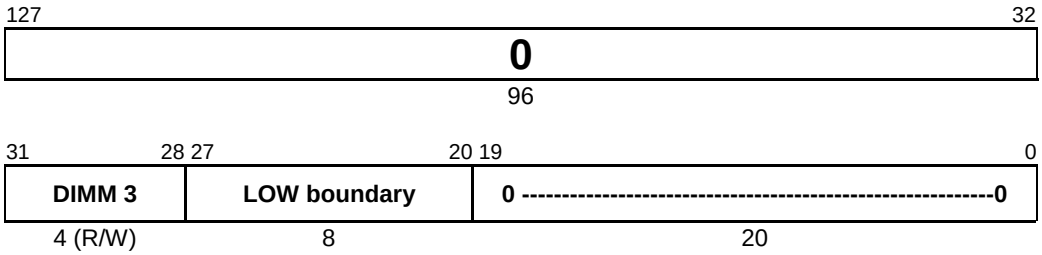
Bits	R/W	Description
31:28	R/W	DIMM 2
27:20	R/W	LOW boundary
19:0	R/O	0

6.5.13 DIMM 2 HIGH Address Decode (0x1E00_0150)



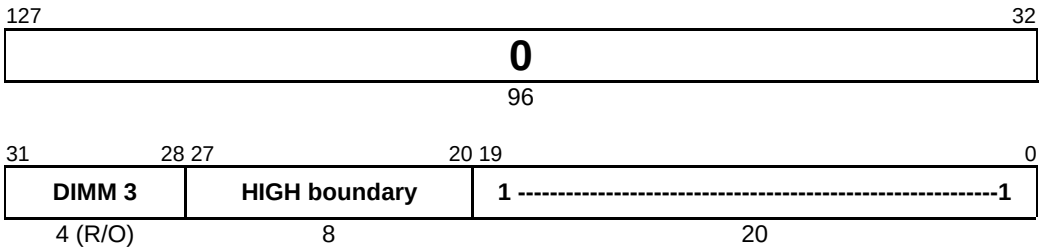
Bits	R/W	Description
31:28	R/O	DIMM 2
27:20	R/W	HIGH Boundary
19:0	R/O	1

6.5.14 DIMM 3 LOW Address Decode (0x1E00_0160)



Bits	R/W	Description
31:28	R/W	DIMM 3
27:20	R/W	LOW boundary
19:0	R/O	0

6.5.15 DIMM 3 HIGH Address Decode (0x1E00_0170)

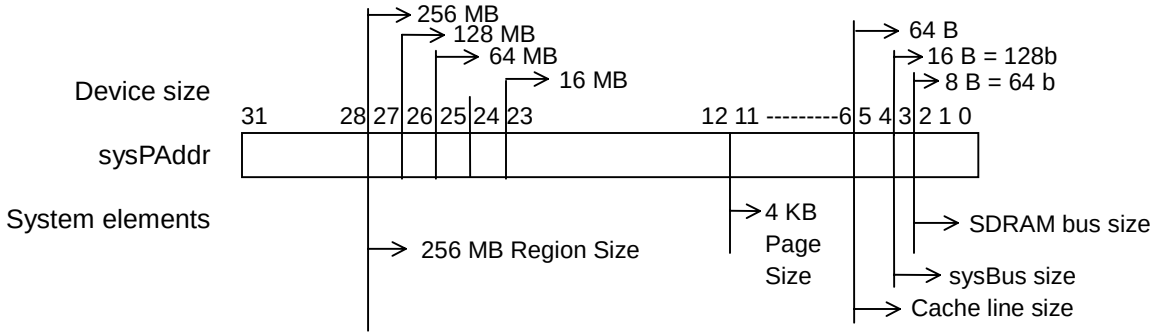


Bits	R/W	Description
31:28	R/O	DIMM3
27:20	R/W	HIGH Boundary
19:0	R/O	1

6.6 Address Mapping

sysPAddr[27:0] is assorted into bank, row, and column addresses. This section describes the mapping of address bits for performance analysis purposes.

Address range



Device Type	Bank address BA[1:0]	Row Address RA[12:0]	Column Address CA[12:0]
	1 0	12 11 10 … 0	12 11 10 9 8 7 6 … 3 2 … 0
16 Mb	– 6	– – 22 …12	– – × × 7 23 11… 8 5 …3
64 Mb	7 6	– 24 22 …12	– × AP × 25 23 11… 8 5 …3
128 Mb	7 6	– 24 22 …12	– × AP 26 25 23 11… 8 5 … 3
256 Mb	7 6	27 24 22 …12	× × AP 26 25 23 11… 8 5 … 3

– : No pins in this device
× : Don't care
AP: Auto Precharge: Assign a constant of 1 to this bit.
Number: bit number of sysPAddr [27:3]

6.7 Data ECC Generation

Each of the 64 data bits and 8 check bits has a unique 8-bit SECDED ECC check code; this check code is generated by taking the even parity of the ECC check code for a selected group of data bits. As Figure 6-3 shows, bit locations are numbered from right to left in ascending order, from data bit 0 (furthest right) to data bit 63 (furthest left). For example, data bit 0, in the far right column of the figure, has an 8-bit check value of 0001_0011_2 (0s are represented in this figure by periods, (.), because they are not used in the calculations).

Figure 6-7 also gives values for the 8 check bits, 7:0. For instance, the 8-bit SECDED ECC code for check bit 6 is in column 6, near the right hand edge of the figure.

Data bit 63 — Nibbles — Check bit 6 — Data bit 0

Check Bit			43		52												70		61	
Data Bit		6666 3210	55 98	5555 7654	55 32	5544 1098	4444 7654	4444 3210	3333 9876	3333 5432	3322 1098	2222 7654	2222 3210	1111 9876	1111 5432	11 10	9876	54	3210
MSB	27	1111	11..	11..	1...	1...	....	1111	1111	....	1...	1...	1...	1...	....	1. 1.	.1..	1...	1...
	27	1111	1...	1...	1...	.1..	....	....	....	1111	.1..	.1..	.1..	.1..	1111	11..	11..	1. 1.	.1..
ECC	27	....	1...	11..	1. 1.	..1.	1111	1111	....	....	..1.	..1.	..1.	..1.	1111	1...	1...	11..	..1.
Code	27	....	1. 1.	.1..	11..	...1	1111	....	1111	1111	...1	...1	...1	...1	....	1...	11..	1...	...1
Bits	27	1...	.1.1	..11	.1..	....	1...	1...	1...	1...	1111	1111	....	1111	1...	11..	...1	.1..	....
	27	.1..	11..	..1.	.1.1	1111	.1..	.1..	.1..	.1..	....	....	1111	1111	.1..	.1..	..11	.1..	....
	27	..1.	.1..	..11	11..	1111	..1.	..1.	..1.	..1.	1111	....	....	....	..1.	.1..	..1.	.1.1	1111
LSB	27	...1	.1..	...1	.1..	...1	...1	...1	...1	...1	....	1111	1111	....	...1	.1.1	..11	11..	1111
Number of 1s in syndrome*		3333	5511	3333	5511	3333	3333	3333	3333	3333	3333	3333	3333	3333	3333	5511	3333	5511	3333

Figure 6-7 Check Matrix for Data ECC Code

Note: * This row indicates the total number of 1s in the generated syndrome for each data bit in an error.

The SDRAM Controller supports ECC detection and correction of 64-bit (72-bit) SDRAMs. If ECC is enabled and there is an ECC error in the read, an interrupt will be asserted to the CPU and the ECC Error Status Register will be set to indicate that bad data were returned.

For 64-bit (72-bit) SDRAMS, if ECC is enabled, check bits will be generated and written to the sdrCB[7:0] lines during the same cycle that the data are written.

ECC will be generated for partial writes via the read-modify-write protocol. Since most SDRAMs come in 8-bit granularities, ECC checking and generation require an additional SDRAM chip to store the ECC information. In order to generate the ECC to this extra device during partial writes, the current ECC bits must first be read and then modified during the partial write. The protocol for the read-modify-write transaction is as follows:

1. Read the existing data. On this read, all sdrDQM lines are asserted LOW. This means that the BE (byte enable) for the ECC byte can be connected to ANY of the

sdrDQM[7:0]* outputs. The ECC data are read on the sdrCB[7:0] inputs.

2. Modify the ECC information based on the data that are to be written. The ECC nibble is modified in the SDRAM Controller.

3. Write the new data and new ECC byte.

Figure 6-8 illustrates the procedure that the SDRAM Controller uses to generate ECC in a partial write to SDRAM.

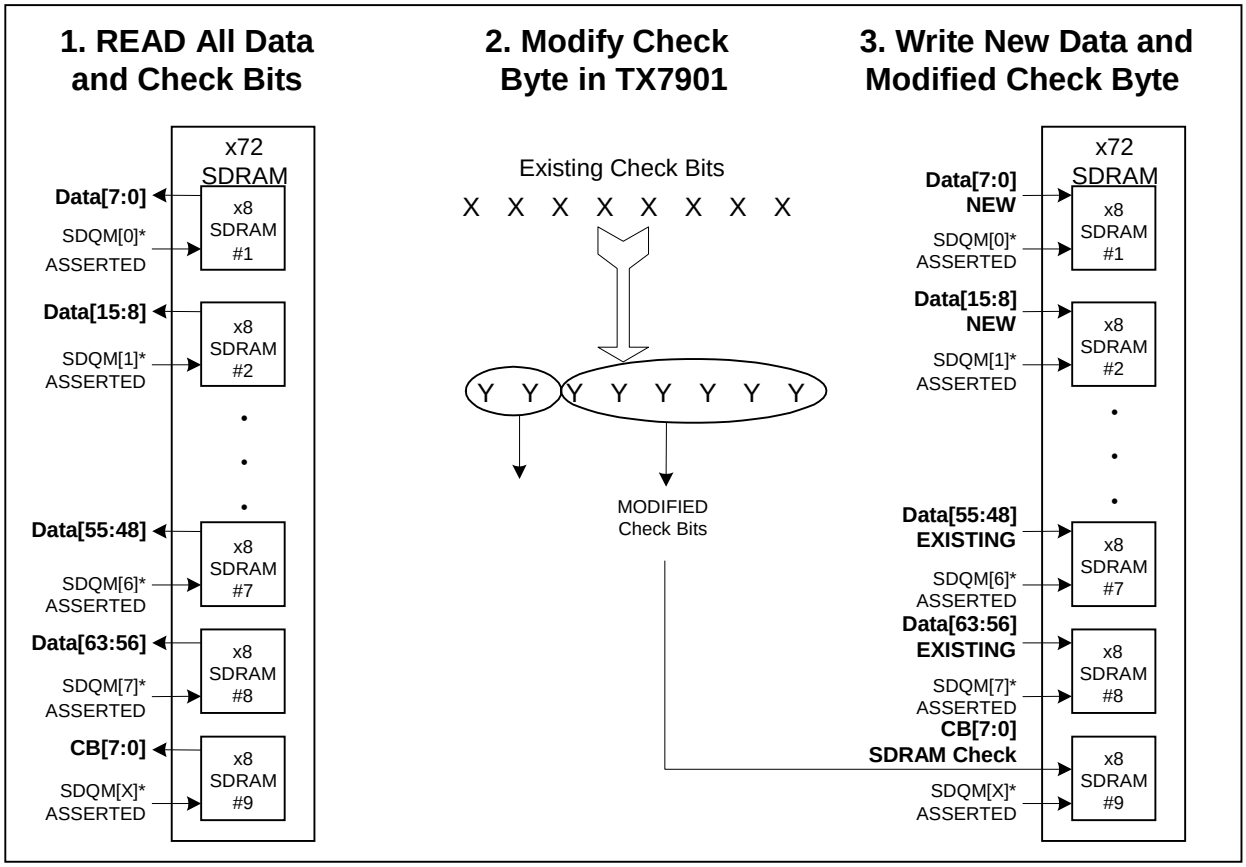


Figure 6-8 Read Modify Write Transaction by the SDRAM Controller

6.8 SDRAM Initialization

Following below is an example of the code used during SDRAM initialization.

```
/* SDRAMC register definition */
#define SDRAMC  0xbe00_0000   /* SDRAMC base address(virtual) */
#define D0PR    0x0000        /* DIMM0 Parameter Register */
#define D1PR    0x0010        /* DIMM1 Parameter Register */
#define D2PR    0x0020        /* DIMM2 Parameter Register */
#define D3PR    0x0030        /* DIMM3 Parameter Register */

#define DOMR    0x0040        /* Operation Mode Register  */
#define DOMR_NORMAL     0     /* Normal Operation Mode    */
#define DOMR_NOP        1     /* NOP mode                 */
#define DOMR_PRECHARGE  2     /* Precharge All Banks      */
#define DOMR_WR_MODEREG 3     /* Write SDRAM mode register*/
#define DOMR_REFRESH    4     /* Force a Refresh Cycle    */
#define DEMR    0x0050        /* ECC Mode Register        */
#define DEESR   0x0060        /* ECC Error Status Register */
#define DEEAR   0x0070        /* ECC Error Address Register */
#define DREFRESH 0x0090       /* Refresh Register         */
#define DDRIVE  0x00a0        /* Output Driver Strength   */

#define D0LOW   0x0100        /* DIMM0 LOW address        */
#define D0HIGH  0x0110        /* DIMM0 HIGH address       */
#define D1LOW   0x0120        /* DIMM1 LOW address        */
#define D1HIGH  0x0130        /* DIMM1 HIGH address       */
#define D2LOW   0x0140        /* DIMM2 LOW address        */
#define D2HIGH  0x0150        /* DIMM2 HIGH address       */
#define D3LOW   0x0160        /* DIMM3 LOW address        */
#define D3HIGH  0x0170        /* DIMM3 HIGH address       */

/* configuration for single 32MB DIMM on DIMM0
 *   consist of five 64Mbit chips with ECC check bit
 *        start       end         size
 * DIMM0  0x0000_0000 0x01ff_ffff 32MB
 * DIMM1  0x0200_0000 0x01ff_ffff  0MB
 * DIMM2  0x0400_0000 0x03ff_ffff  0MB
 * DIMM3  0x0600_0000 0x05ff_ffff  0MB
 */

#define RAMBASE 0x0000_0000   /* start physical address   */
#define RAM32MB 0x0200_0000   /* size of 32MB memory      */
#define DIMM0ADDR 0xa000_000  /* dummy write address(virtual) for DIMM0 */
#define DPR_CL3 \
                  1           /*  1:0  DSL   64/128Mb */\
                | 3 << 16     /* 17:16 CAL   CL=3 cycle */\
                | 3 << 18     /* 20:18 A2RWI Active to Write = 3 cycle */\
                | 3 << 21     /* 23:21 P2AI  Precharge to active = 3 cycle */\
                | 2 << 24     /* 25:24 WRT   Write recovery = 2 cycle */\
                | 6 << 26     /* 28:26 RRT   Refresh Recovery = 10 cycle */\
                | 1 << 29     /* 29    A2P   Active to Precharge = 6 cycle */

#define SDRM_CL3              /* RA  Paddr description                */\
                  1 << 12     /* 2:0 14:12 Burst Length = 2           */\
```

```
        | 0 << 15     /* 3   15    Burst Type = Sequential    */\
        | 3 << 16     /* 6:4 18:16 CAS Latency = 3            */\
        | 0 << 19     /* 8:7 20:19 OpMode = Standard Op       */\
        | 0 << 21     /* 9   21    Write Burst Mode = programmed */\
        | 0xa000_0000 /* UnCached UnMapped area for DIMM0     */

#define SDR_RFSH       /* Refresh Register's value */\
                1500  /* RIC: 64ms/4096 >15,000 ns @ 100 MHz */\
              | 0<<16 /* NSRF: Staggered refresh */\
              | 1<<17 /* RFE: Refresh enable */\

# initialize registers and force NOP mode
la      k1, SDRAMC      # SDRAMC base address
la      t1, DIMM0ADDR   # address of dummy write for DIMM0
li      t0, DOMR_NOP
sq      t0, DOMR(k1)    # force NOP mode to prevent unnecessary access

# specify SDRAM parameter
li      t0, DPR_CL3
sq      t0, D0PR(k1)
sq      t0, D1PR(k1)
sq      t0, D2PR(k1)
sq      t0, D3PR(k1)

# set up SDRAM address windows
li      t0, RAMBASE     # start address of DIMM0
li      t2, RAMBASE + RAM 32MB-1
                        # end address of DIMM0
sq      t0, D0LOW(k1)   # store DIMM0 LOW reg  (starting 0x0000_0000)
sq      t2, D0HIGH(k1)  # store DIMM0 HIGH reg (32MB)

li      t0, RAMBASE + RAM32MB
li      t2, RAMBASE + RAM32MB*2-1
sq      t0, D1LOW(k1)   # store DIMM1 LOW reg  (starting 0x0200_0000)
sq      t2, D1HIGH(k1)  # store DIMM1 HIGH reg (0MB)

li      t0, RAMBASE + RAM32MB*2
li      t2, RAMBASE + RAM32MB*3-1
sq      t0, D2LOW(k1)   # store DIMM2 LOW reg  (starting 0x0400_0000)
sq      t2, D2HIGH(k1)  # store DIMM2 HIGH reg (0MB)

li      t0, RAMBASE + RAM32MB*3
li      t2, RAMBASE + RAM32MB*4-1
sq      t0, D3LOW(k1)   # store DIMM3 LOW reg  (starting 0x0600_0000)
sq      t2, D3HIGH(k1)  # store DIMM3 HIGH reg (0MB)

#Precharge All Bank
li      t0, DOMR_PRECHARGE
sq      t0, DOMR(k1)    # mode set to precharge all bank
sw      $0, 0(t1)       # issue command

# two Auto Refresh cycles
li      t0, DOMR_REFRESH
sq      t0, DOMR(k1)
sw      $0, 0(t1)       # issue command
sw      $0, 0(t1)       # issue command twice
```

```
# write SDRAM chip's Mode Register
li      t0, DOMR_WR_MODEREG
sq      t0, DOMR(k1)    # write SDRAM Mode Register
la      t2, SDRM_CL3
sw      $0, 0(t2)       # write address goes to MODE register
lw      t0, DOMR(k1)    # read DOMR to make sure Write Mode Reg
                        # It takes more than 2 x sysBusClk

# start Refresh
li      t0, SDR_RFSH
sq      t0, DREFRESH (k1)

# return to Normal mode of SDRAMC access
li      t0, DOMR_NORMAL
sq      t0, DOMR(k1)
sync.l
lw      t0, DOMR(k1)    # read DOMR to make sure Normal mode
sync.l

# Now SDRAM is ready for access
```

7.C790 Bus / G-Bus Bridge

7.1 Introduction

The C790 Bus/ G-Bus Bridge provides an efficient interface between the C790 bus (and its attached C790 CPU and Main Memory), and the G-Bus (and its attached peripheral devices.) The bridge supports C790 accesses to devices on the G-Bus, and G-Bus Mastering devices' access to the main memory on the C790 Bus.

Figure 7-1 shows the block diagram of the G-bridge.

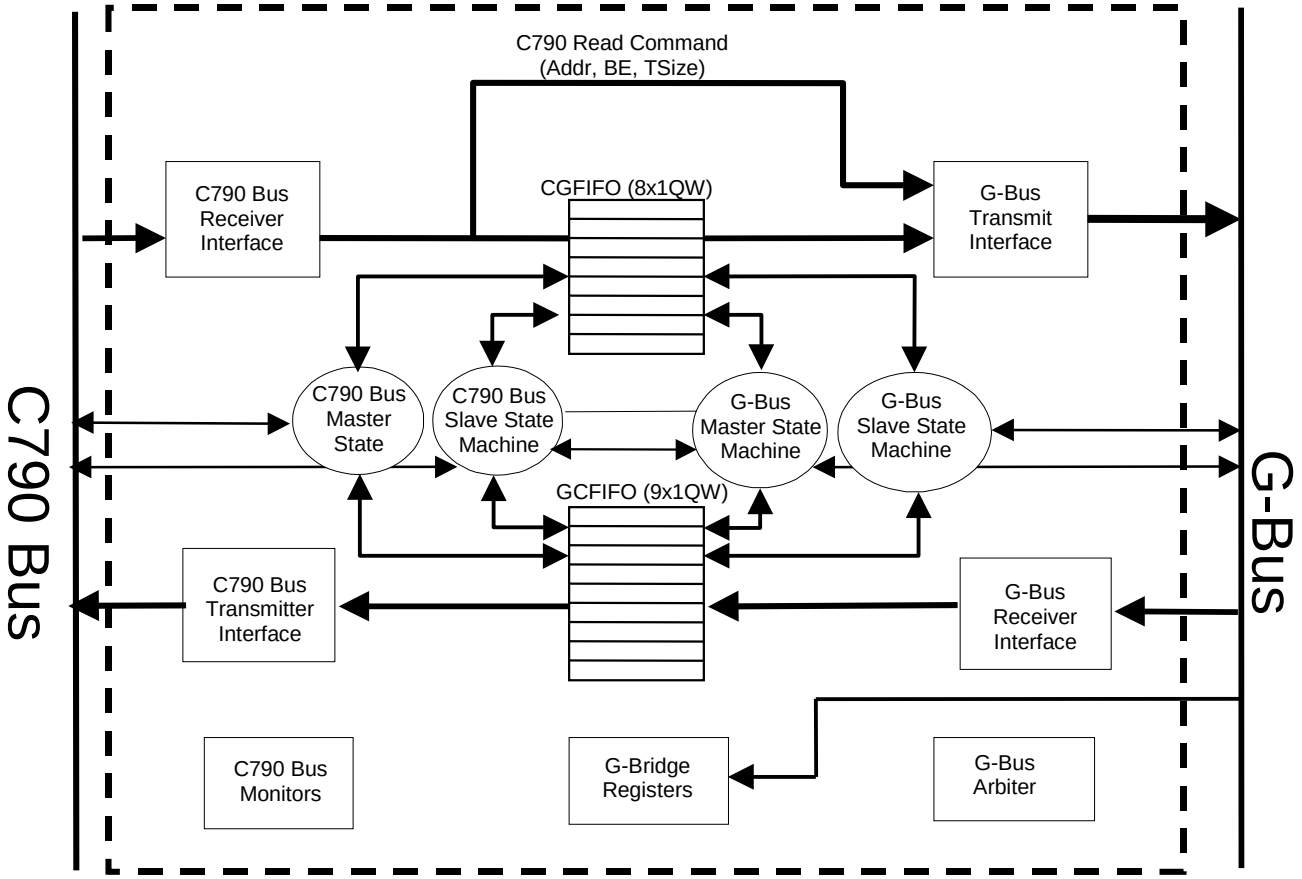


Figure 7-1 C790 Bus / G-Bus Bridge Block Diagram

7.2 Address Space Decode and Translation

The C790 accesses G-Bus devices through one CG internal register window, one ROM window and four PCI windows. The upper 28-bit address pAddr[31:4] seen on the C790 bus is copied directly onto the G-Bus while the lower 2–bit address is derived from the byte enable bits. G-Bus Masters access the main memory through the memory controller on the C790 Bus via one GC internal register window and five memory windows. The upper 28-bit address seen on the G-Bus is copied directly into the C790 Bus while the lower 2 address bits on the G-Bus are used to specify the location of the word/double-word in a quad-word. (See Figure 7-2.)

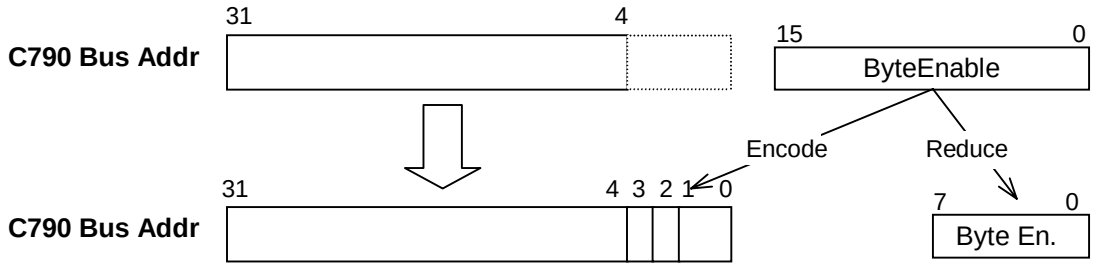


Figure 7-2 G-Bridge Address Translation

7.3 Bus Transactions

The C790 can issue either a single quad-word operation or 4 quad-word burst operations on the C790 bus. When the C790 initiates a bus cycle to a G-Bus device, the bridge detects the address and posts the written data into the CGFIFO if it is a write transaction, or initiates a G-Bus read transaction. Such transactions are called CG (C790 to G-Bus) transactions.

The masters on the G-Bus can initiate a single double-word operation or a burst operation of up to 8 quad-words (16 double words). When G-Bus masters initiate a bus cycle to main memory, the bridge detects the address and posts the written data into the GCFIFO if it is a write transaction, or initiates a C790 Bus read transaction. Such transactions are called GC (G-Bus to C790) transactions.

The C790 Bus supports the Wrap-Around addressing mode while the G-Bus supports the linearly-incrementing addressing mode. 4-quad-word burst operation on the C790 Bus is translated into 4-quad-word aligned burst transaction on the G-Bus.

The C790 and G-Bus Masters can issue their transactions independently and the bridge can handle and arbitrate their concurrent transactions correctly.

7.4 Endianness

The C790 Bus and G-Bus support both Little and Big Endian byte ordering, while the PCI Bus only supports fixed Little Endian byte ordering. When the C790 bus is configured for the Big Endian mode via the Endianness signal, the G-Bus is configured for Big Endian as well. Bus narrowing is performed by the G-Bridge. As a result of dynamic bus signing, the G-Bus retries the transaction to locate data in the proper word lane.

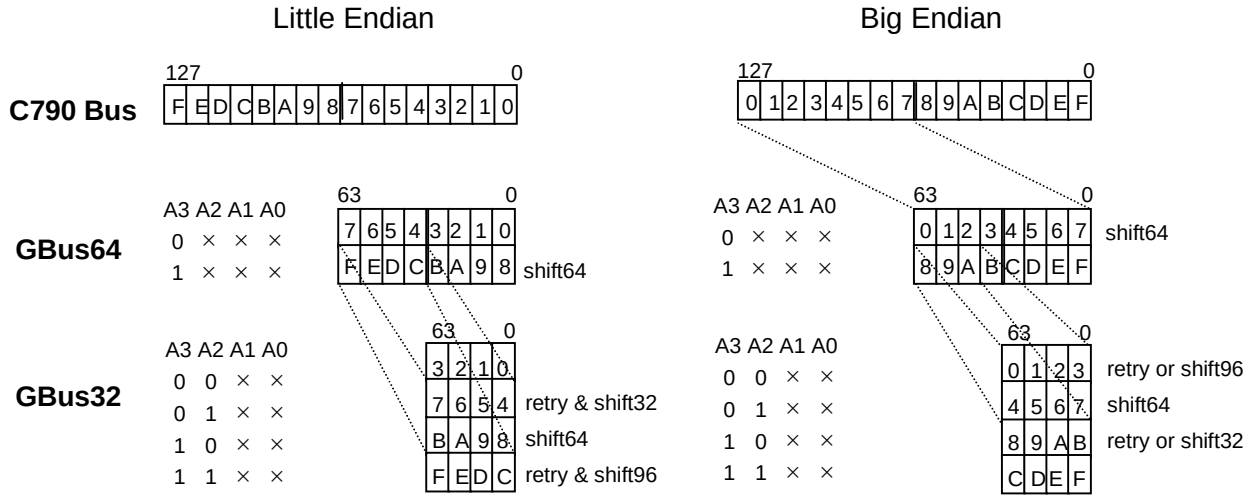


Figure 7-3 Bi-Endian Support

Note that all internal registers except those for the SDRAMC are located on the G-Bus. See Chapter 4 “Address Maps” for more information.

7.5 Bus Errors

Bus errors occur during a bus transaction when no target device responds to the initiator within a given number of bus cycles. When a bus error occurs, the initiator terminates the transaction, signals an interrupt to the C790 and records the bad address. The target device returns to the idle state after a bus error has been detected.

There are two kinds of bus errors. One is a C790 Bus error, and the other is a G-Bus error. Each bus error has a counter associated with it. When the initiator initiates a transaction, the counter begins to count down. When the counter counts to zero and there is no response from any target, a bus error is signaled. Since the C790 Bus Clock is twice as fast as the G-Bus Clock, we can state the following constraint:

(C790 Bus Counter) < G-Bus Counter $\times$ 2 – n, where *n* is the maximum latency of G $\rightarrow$ C transfer

This implies that the C790 bus countdown always expires **sooner** than the G-Bus countdown if there is any transaction going across the G-Bus bridge (called a “crossing transaction”.) When there is a crossing transaction, the bridge always passes any C790 bus error to the G-Bus. The initiator or target handles it in the same way a G-Bus error would be handled.

When a write transaction has posted its data into the FIFO in the bridge, the transaction is considered “finished”. A write transaction is never a crossing transaction.

7.6 Registers

The registers in the G-Bus bridge are memory-mapped into the C790 address space ranging from address 0x1E00_2000 to address 0x1E00_2FFF.

Table 7-1 List of G-Bus Bridge Registers (Base Address = 0x1E00_2000)

Register Name	Offset	R/W	Size	Initial Value
System Configuration Register	0x00	R	64	0x0000_0000
C790 Bus Control Register	0x08	R/W	64	0x0000_0043
C790 Bus Status Register	0x10	R/W	64	0x0000_0000
C790 Bus Bad Address Register	0x18	R	64	Undefined
CG Upper Internal Register Address	0x20	R	64	0x1EFF_FFFF
CG Lower Internal Register Address	0x28	R	64	0x1E00_1000
CG Upper ROM Address Register	0x30	R	64	0x20FF_FFFF
CG Lower ROM Address Register	0x38	R	64	0x1F00_0000
CG Upper PCI Address 0	0x40	R/W	64	Undefined
CG Lower PCI Address 0	0x48	R/W	64	Undefined
CG Upper PCI Address 1	0x50	R/W	64	Undefined
CG Lower PCI Address 1	0x58	R/W	64	Undefined
CG Upper PCI Address 2	0x60	R/W	64	Undefined
CG Lower PCI Address 2	0x68	R/W	64	Undefined
CG Upper PCI Address 3	0x70	R/W	64	Undefined
CG Lower PCI Address 3	0x78	R/W	64	Undefined
GC Upper Internal Register Address	0x80	R	64	0x1E00_0FFF
GC Lower Internal Register Address	0x88	R	64	0x1E00_0000
GC Upper MEM Address 0	0x90	R/W	64	Undefined
GC Lower MEM Address 0	0x98	R/W	64	Undefined
GC Upper MEM Address 1	0xA0	R/W	64	Undefined
GC Lower MEM Address 1	0xA8	R/W	64	Undefined
GC Upper MEM Address 2	0xB0	R/W	64	Undefined
GC Lower MEM Address 2	0xB8	R/W	64	Undefined
GC Upper MEM Address 3	0xC0	R/W	64	Undefined
GC Lower MEM Address 3	0xC8	R/W	64	Undefined
GC Upper MEM Address 4	0xD0	R/W	64	Undefined
GC Lower MEM Address 4	0xD8	R/W	64	Undefined
Interrupt Status Register	0xE0	R	64	Undefined
Interrupt Mask Register	0xE8	R/W	64	0x0030_0000
C790 Bus Latency Timer	0xF0	R/W	64	0xFFFF_FFFF
NMI Status Register	0xF8	R	64	Undefined
G-Bus Master Latency Timer	0x100	R/W	64	0xFFFF
G-Bus Broken Master Timer	0x108	R/W	64	0xFF
G-Bus Slave Latency Timer	0x110	R/W	64	0xFFFF
G-Bus Retry Timer	0x118	R/W	64	0x02
GC Control register	0x120	R/W	64	0x0000_0001
G-Bus Status Register	0x128	R/W	64	0x0000_0000
G-Bus Bad Address Register	0x130	R	64	Undefined
G-Bus Arbiter Request Status Register	0x138	R	64	Undefined
G-Bus Arbiter Granted Status Register	0x140	R	64	0x0000_FFFE
G-Bus Arbiter Master Status Register	0x148	R/W	64	0x0000_0000
G-Bus Arbiter Control Register	0x150	R/W	64	0x0000_FFFF

7.6.1 System Configuration Register

The System Configuration Register specifies configurations for the system. Its fields are detailed below and in Table 7-2.

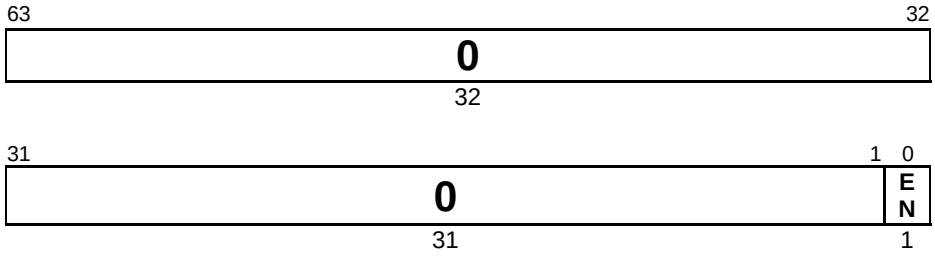


Table 7-2 System Configuration Register Fields

Bit(s)	Name	Description	Type	Initial Value
63:1	–	Reserved. Must be written as zeroes, and returns zeroes when read.	Read only	0
0	EN	Endianness 1: Big Endian 0: Little Endian	Read	Pin defined

7.6.2 C790 Bus Control Register

The Bus Control Register enables various functions on the C790 Bus. The fields of the Bus Control Register are detailed below and in Table 7-3.

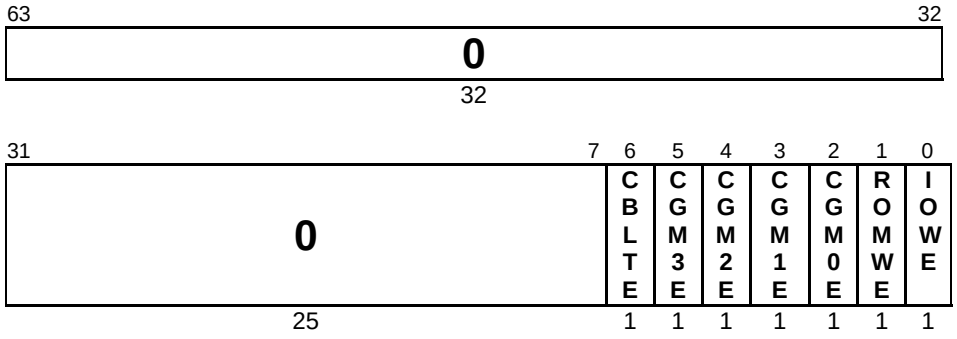


Table 7-3 C790 Bus Control Register Fields

Bit(s)	Field	R/W	Description	Initial Value
63:7	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
6	CBLTE	R/W	0: C790 Bus Latency Timer Enable 1: Enable	1
5	CGM3E	R/W	CG memory Mapping Window 3 Enable 1: Enable 0: Disable	0
4	CGM2E	R/W	CG memory Mapping Window 2 Enable 1: Enable 0: Disable	0
3	CGM1E	R/W	CG memory Mapping Window 1 Enable 1: Enable 0: Disable	0
2	CGM0E	R/W	CG memory Mapping Window 0 Enable 1: Enable 0: Disable	0
1	ROMWE	R/O	CG ROM Mapping Window Enable 1: Enable 0: Disable	1
0	IOWE	R/O	CG Internal Register Mapping Window Enable 1: Enable 0: Disable	1

7.6.3 C790 Bus Status Register

The Bus Status Register reports the status of the C790 Bus. The ERR bit is set when the bridge is writing to main memory, then a C790 bus error occurs. The C790 can clear this bit by writing a “0”. Writing a “1” is ignored. The fields of the Bus Status Register are detailed below and in Table 7-4. Note that read errors are detected by the transaction originator.

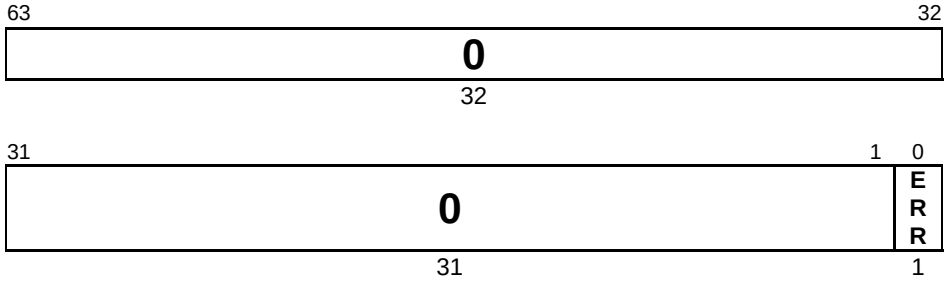


Table 7-4 C790 Bus Status Register Fields

Bit(s)	Field	R/W	Description	Initial Value
63:1	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
0	ERR	R/W	C790 Bus Error 1: Bus Error occurred 0: No bus error Error is generated when ERR = 1. Writes "0" when cleared. Writing "1" to this bit has no effect. This bit is not set during reads. Notification is just sent to the Master.	0

7.6.4 C790 Bus Bad Address Register

The C790 Bus Bad Address Register reports the C790 address when there is a bus error on the C790 Bus and the bridge is writing to the main memory. This register can only be set when the ERR bit in the Status register is “0”. Note that the first error is held in case multiple errors occur.

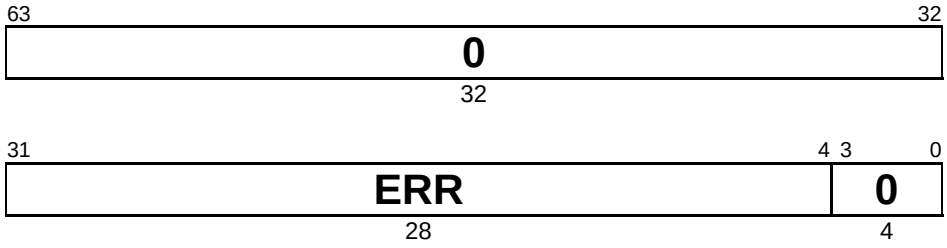


Table 7-5 C790 Bus Bad Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:4	ERR	R/O	Bad C790 Address	Undefined
3:0	–	R/O	Reserved	0x0

7.6.5 CG Upper Internal Register Address (UIRA)

The Upper Internal Register Address defines the upper internal register address for devices on the G-Bus.

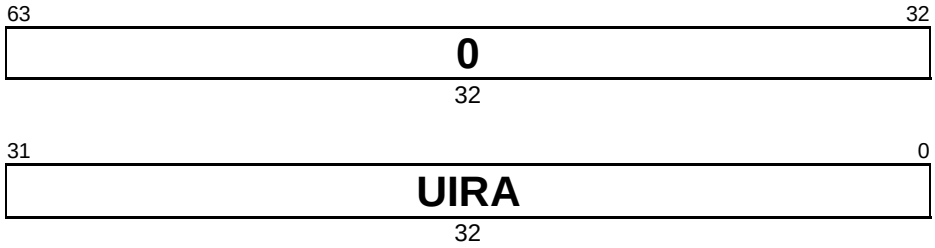


Table 7-6 CG Upper I/O Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	UIRA	R/O	Upper Internal Register Address	0x1EFF_FFFF

7.6.6 CG Lower Internal Register Address (LIRA)

The Lower Internal Register Address Register defines the lower internal register address for devices on the G-Bus.

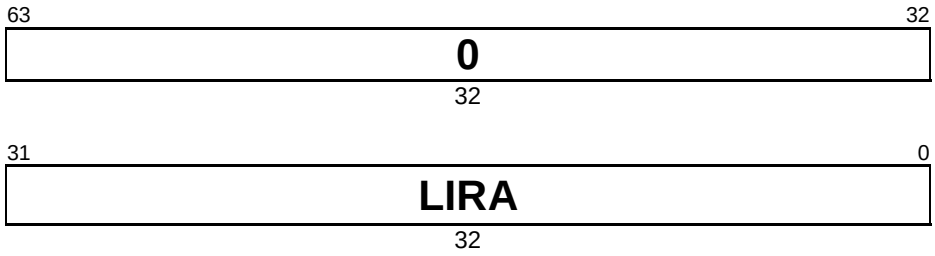


Table 7-7 CG Lower Internal Register Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	LIRA	R/O	Lower Internal Register Address	0x1E00_1000

7.6.7 CG Upper ROM Address Register (UROMA)

The Upper ROM Address register defines the upper address for ROM / SRAM / external I/O devices on the G-Bus.

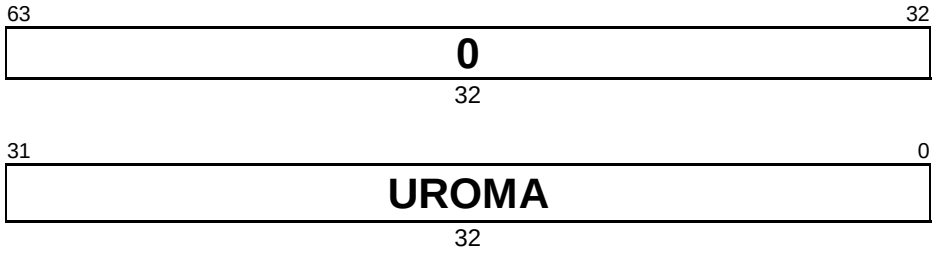


Table 7-8 CG Upper ROM Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	UROMA	R/O	Upper ROM Address	0x20FF_FFFF

7.6.8 CG Lower ROM Address Register (LROMA)

The Lower ROM Address Register defines the lower address for ROM / SRAM / external I/O devices on the G-Bus.

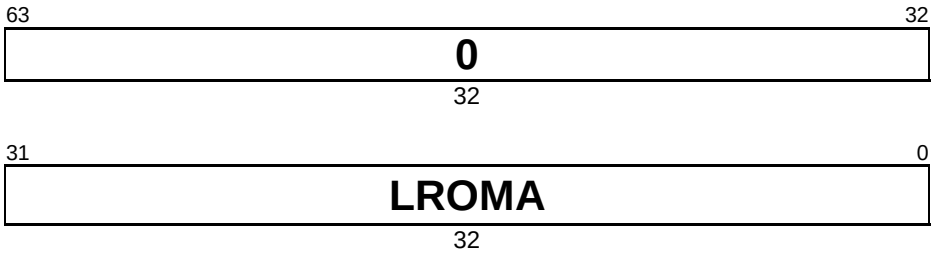


Table 7-9 CG Lower ROM Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	LROMA	R/O	Lower ROM Address	0x1F00_0000

7.6.9 CG Upper PCI Address Register (CGUPA0, CGUPA1, CGUPA2, CGUPA3)

The CG Upper PCI Address Register defines the upper address of the CG mapping window (CGUPA0, CGUPA1, CGUPA2, CGUPA3). These addresses should be aligned to the last byte of the word boundary.

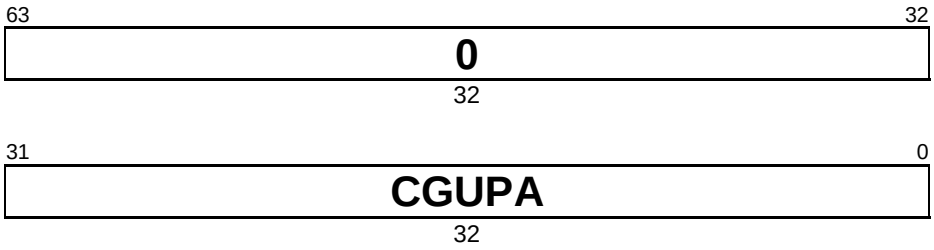


Table 7-10 CG Upper PCI Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	CGUPA	R/W	Upper Address	Undefined

7.6.10 CG Lower PCI Address Register (CGLPA0, CGLPA1, CGLPA2, CGLPA3)

The CG Lower PCI Address Register defines the lower address of the CG mapping window (CGLPA0, CGLPA1, CGLPA2, CGLPA3). These addresses should be aligned to the word boundary.

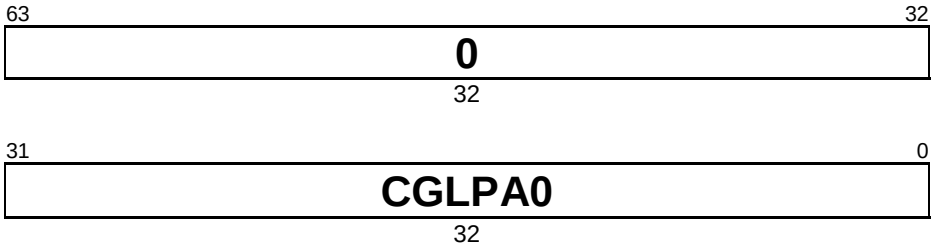


Table 7-11 CG Upper Lower PCI Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	CGLPA0	R/W	Upper Address	Undefined

7.6.11 GC Upper Internal Register Address Register (GCUIRA)

The GC Upper Internal Register Address register defines the upper bound register address of the C790 Bus that any G-Bus master can access.

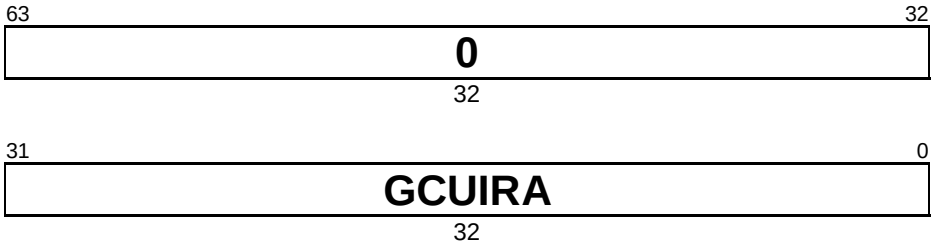


Table 7-12 GC Upper Internal Register Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	GCUIRA	R/O	Upper C790 Bus Upper Internal Register Address	0x1E00_0FFF

7.6.12 GC Lower Internal Register Address Register (GCLIRA)

The Lower Internal Register Address Register defines the lower bound register address of the C790 Bus that any G-Bus master can access.

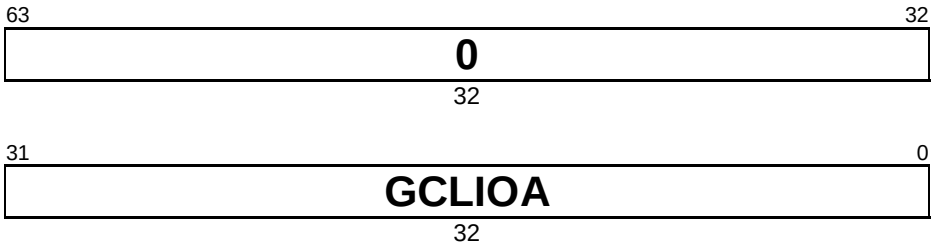


Table 7-13 GC Lower Internal Register Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	GCLIOA	R/O	Lower I/O Address	0x1E00_0000

7.6.13 GC Upper Memory Address Register (*GCUMAx*)

The GC Upper Memory Address Register defines the upper address of the GC memory mapping window. These addresses should be aligned to the quad-word (QW) boundary.

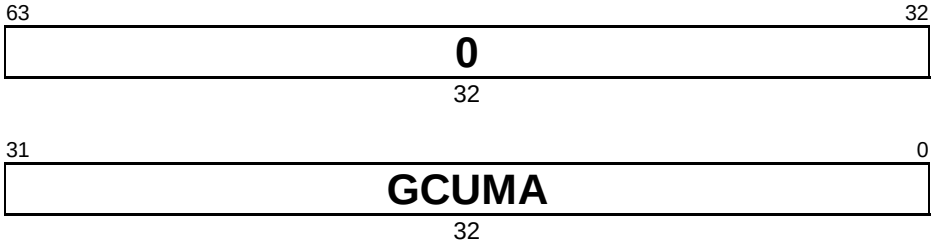


Table 7-14 GC Upper Memory Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	GCUMA	R/W	GC Upper Memory Address	Undefined

7.6.14 GC Lower Memory Address Register (GCLMA*x*)

The GC Lower Memory Address Register defines the lower address of the GC memory-mapping window. These addresses should be aligned to the quad-word boundary.

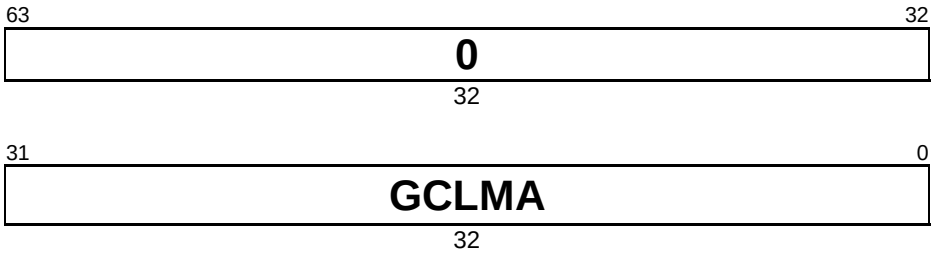


Table 7-15 GC Lower Memory Address Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	GCLMA	R/W	GC Lower Memory Address	Undefined

7.6.15 Interrupt Status Register (IRSTAT)

The Interrupt Register reports the status of the internal and external interrupt requests. An interrupt is generated if a bit in the register is set to “1” and its corresponding mask bit in the Mask register is set to “1.”

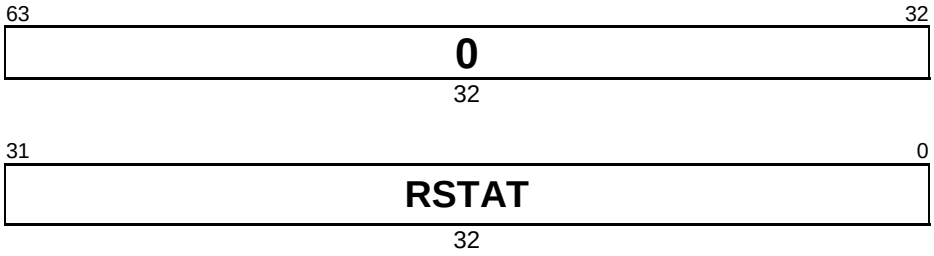


Table 7-16 C790 Interrupt Status Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	RSTAT	R/O	Interrupt Status 1: Interrupt request exists. 0: Interrupt request doesn’t exist.	Undefined

Table 7-17 G-Bus Interrupt Source Table

Event	Status	Mask	Mask Init. Value	R/W
[31:22]	Reserved	[31:22]	0	R/O
IRSTAT[21]	PCI1 Reset	IRMSK[21]	1	R/O
IRSTAT[20]	PCI0 Reset	IRMSK[20]	1	R/O
IRSTAT[19]	External Interrupt 4	IRMSK[19]	0	R/W
IRSTAT[18]	External Interrupt 3	IRMSK[18]	0	R/W
IRSTAT[17]	External Interrupt 2	IRMSK[17]	0	R/W
IRSTAT[16]	External Interrupt 1	IRMSK[16]	0	R/W
IRSTAT[15]	External Interrupt 0	IRMSK[15]	0	R/W
IRSTAT[14]	External Interrupt Mac 1	IRMSK[14]	0	R/W
IRSTAT[13]	External Interrupt Mac 0	IRMSK[13]	0	R/W
IRSTAT[12]	PCI – G-Bus Bridge 1	IRMSK[12]	0	R/W
IRSTAT[11]	SPI	IRMSK[11]	0	R/W
IRSTAT[10]	UART1	IRMSK[10]	0	R/W
IRSTAT[9]	UART0	IRMSK[9]	0	R/W
IRSTAT[8]	Timer 2	IRMSK[8]	0	R/W
IRSTAT[7]	Timer 1	IRMSK[7]	0	R/W
IRSTAT[6]	Timer 0	IRMSK[6]	0	R/W
IRSTAT[5]	DMA	IRMSK[5]	0	R/W
IRSTAT[4]	Reserved	IRMSK[4]	0	R/W
IRSTAT[3]	eMAC1	IRMSK[3]	0	R/W
IRSTAT[2]	eMAC0	IRMSK[2]	0	R/W
IRSTAT[1]	PCI – G-Bus Bridge 0	IRMSK[1]	0	R/W
IRSTAT[0]	G-Bridge	IRMSK[0]	0	R/W

7.6.16 Interrupt Mask Register (IRMSK)

The Interrupt Mask Register enables/disables interrupt generation.

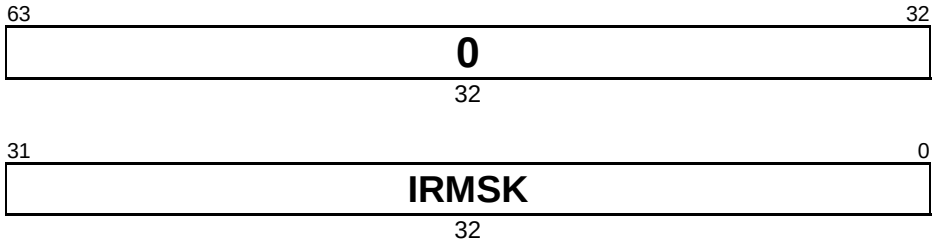


Table 7-18 C790 Interrupt Mask Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	IRMSK	R/W	Interrupt Mask 0: Mask the interrupt request 1: Enable the interrupt request	0

7.6.17 C790 Bus Latency Timer (LT)

The latency timer specifies the maximum period in which the slave has to acknowledge the master. The counter starts counting down automatically when the master asserts the SYSASTARTB signal. The counter is decremented at every C790 bus clock cycle. If the counter is down to zero and the slave device does not respond, a C790 Bus error is generated and the master generates an interrupt signal to inform the C790 core.

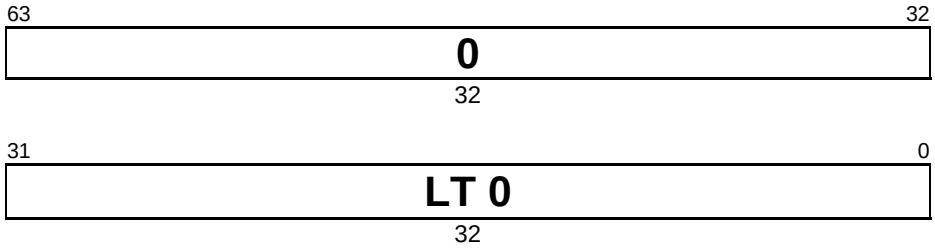


Table 7-19 C790 Bus Latency Timer

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	LT 0	R/W	Latency Timer	0xFFFFFFFF

7.6.18 NMI Status Register (NRSTAT)

The NMI Status Register reports the status of the Non-Maskable interrupt requests. An interrupt is generated if a bit in the register is set to “1.” The NMI handler analyzes the cause of the NMI and serves the request. After service is finished, the handler must reset the cause of the NMI. Then, the handler returns from the NMI, if possible.

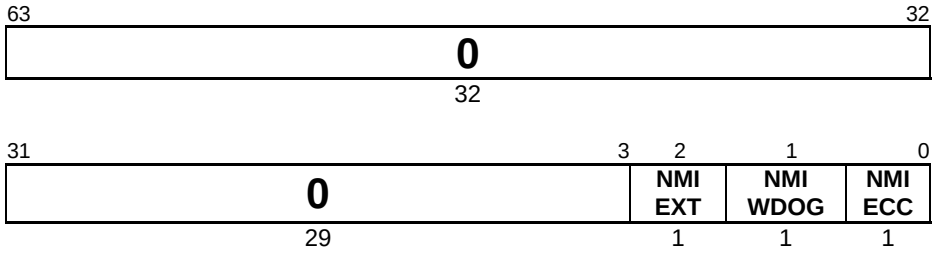


Table 7-20 NMI Status Register

Bit(s)	Field	R/W	Description	Initial Value
63:3	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
2	NMI EXT	R/O	External NMI input	1
1	NMI WDOG	R/O	Timer Watchdog at Time2	1
0	NMI ECC	R/O	ECC uncorrectable Error	1

7.6.19 G-Bus Master Latency Timer

The latency timer specifies the maximum period in which the master can hold the G-Bus when other masters are requesting the G-Bus. The counter is decremented at every G-Bus clock cycle. When the counter counts down to zero and there is a pending G-Bus request, the gbsgRelB signal is asserted to request the current master to release the G-Bus.

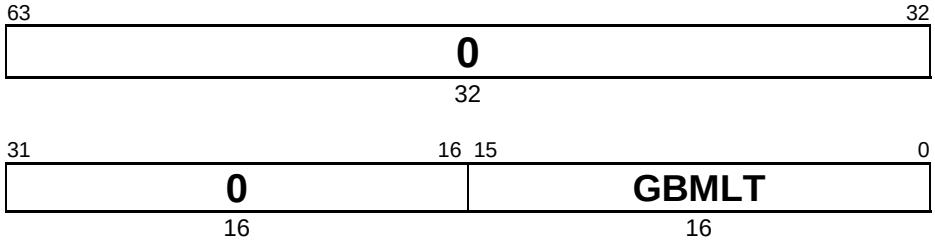


Table 7-21 G-Bus Master Latency Timer

Bits	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
15:0	GBMLT	R/W	Latency Timer. Is counted down by the G-Bus clock	0xFFFF

7.6.20 G- Bus Broken Master Latency Timer

The latency timer specifies the maximum period in which the master has to claim the G-bus after the G-bus is granted. The counter is decremented at every B-Bus clock cycle. The counter starts counting down when the arbiter asserts the grant signal. If the counter is down to zero and the master device does not respond, an interrupt is signaled to the C790, and the corresponding master status bit is set.

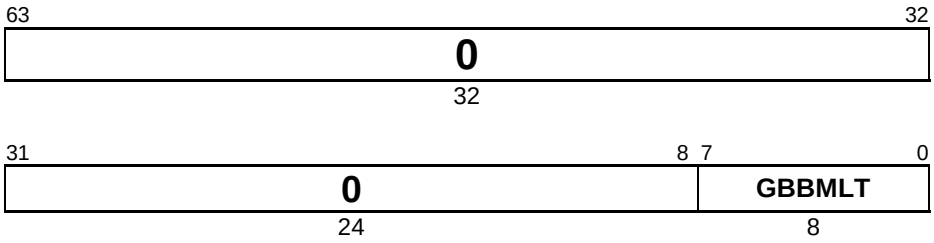


Table 7-22 G-Bus Broken Master Latency Timer

Bits	Field	R/W	Description	Initial Value
63:8	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
7:0	GBBMLT	R/W	G-Bus Broken Master Latency Timer. Is counted down by the G-Bus clock.	0xFF

7.6.21 G- Bus Slave Latency Timer

The slave latency timer specifies the maximum period in which the slave must acknowledge the master. The counter is decremented at every G-Bus clock. The counter starts counting down when the master asserts the STARTB signal. If the counter is down to zero and the slave device does not respond, a G-Bus error is generated and the master generates an interrupt signal to inform the C790.

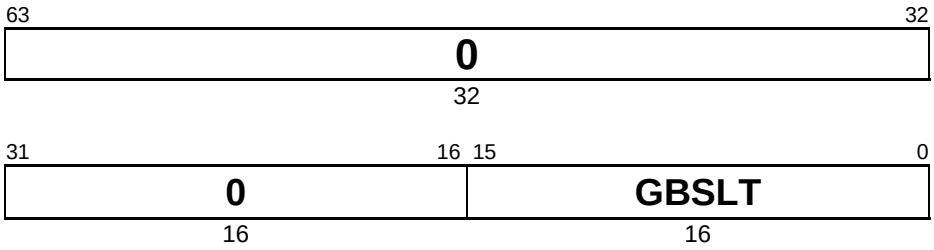


Table 7-23 G-Bus Slave Latency Timer

Bits	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
15:0	GBSLT	R/W	Latency Timer. Is counted down by the G-Bus clock.	0xFFFF

7.6.22 G-Bus Retry Timer

The Retry timer specifies the minimum period in which the bridge can re-assert the G-Bus request signal after it receives the Retry signal and releases the G-Bus. The minimum value is 2. This timer is counted down by the G-Bus clock.

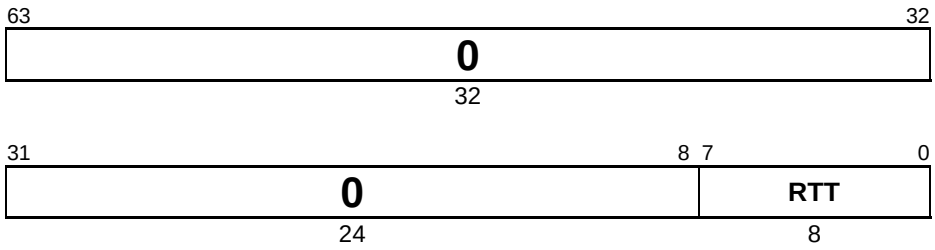


Table 7-24 G-Bridge Retry Timer Fields

Bits	Field	R/W	Description	Initial Value
63:8	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
7:0	RTT	R/W	Retry Timer	0x02

7.6.23 GC Control Register

The Control register enables the mapping from the G-Bus to the C790 bus.

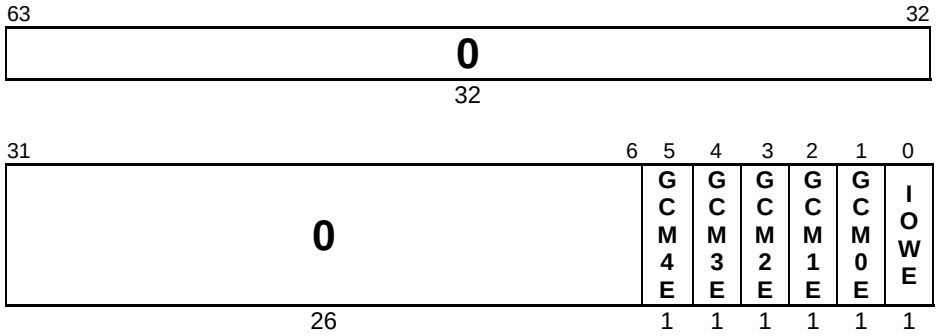


Table 7-25 The GC Control Register Fields

Bit(s)	Field	R/W	Description	Initial Value
63:6	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
5	GCM4E	R/W	GC memory Mapping Window 4 Enable 1: Enable 0: Disable	0
4	GCM3E	R/W	GC memory Mapping Window 3 Enable 1: Enable 0: Disable	0
3	GCM2E	R/W	GC memory Mapping Window 2 Enable 1: Enable 0: Disable	0
2	GCM1E	R/W	GC memory Mapping Window 1 Enable 1: Enable 0: Disable	0
1	GCM0E	R/W	GC memory Mapping Window 0 Enable 1: Enable 0: Disable	0
0	IOWE	R/O	GC Internal Register Mapping Window Enable 1: Enable 0: Disable Note: Writing a “0” to this bit resets it to “0”. Writing a “1” to this bit has no effect.	1

7.6.24 G-Bus Status Register

The Status register reports the status of the G-Bridge. The ERR bit is set when the bridge is writing to main memory, then a G-Bus error occurs. The G-Bus can clear this bit by writing a “0”. Writing a “1” is ignored. The fields of the Bus Status Register are detailed below and in Table 7-26. Note that read errors are detected by the transaction originator.

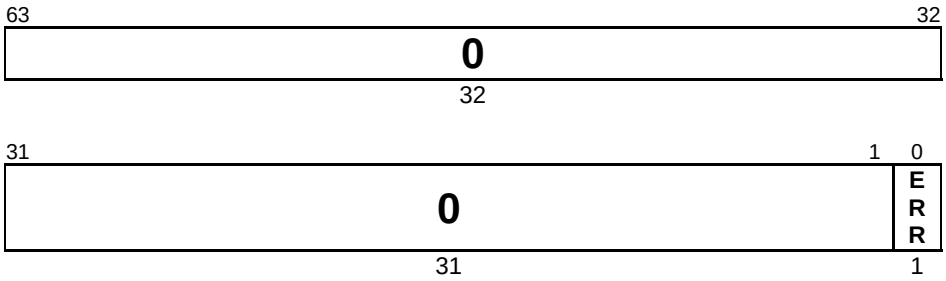


Table 7-26 G-Bridge Status Register Fields

Bit(s)	Field	R/W	Description	Initial Value
63:1	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
0	ERR	R/W	G- Bus Error when the G-bridge is the G-Bus master 1: Bus Error occurred 0: No bus error Error is generated when ERR = 1. Program must write “0” to clear the error. Writing "1" to this bit has no effect. This bit is not set during reads. Notification is just sent to the Master.	0

7.6.25 G-Bus Bad Address Register

The Bad Address register reports the G-Bus address when there is bus error on the G-Bus and the G-Bus bridge is writing to the device on the G-bus. This register can only be set when the ERR bit in the Status register is “0”. Note that the first error is held in case multiple errors occur.

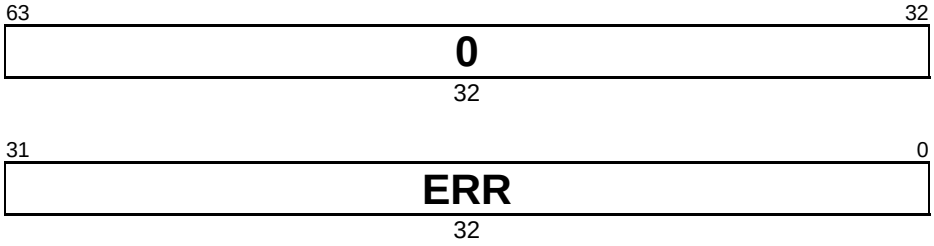


Table 7-27 C790 Bus Status Register Fields

Bits	Field	R/W	Description	Initial Value
63:32	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
31:0	ERR	R/O	Bad G-Bus Address	Undefined

7.6.26 G-Bus Arbiter Request Status Register

This register indicates which G-master is requesting the G-Bus.

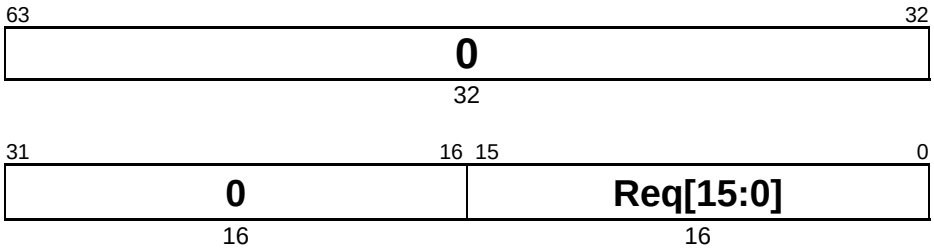


Table 7-28 G-Bus Arbiter Request Status Register Fields

Bits	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0x0000
15:0	Req[15:0]	R/O	G-Bus Request 0: Request 1: No request	Undefined

Table 7-29 G-Bus Arbitration Request Table

Req [15:7]	Reserved
Req 6	PCI-G-Bus Bridge 1
Req 5	DMAC
Req 4	Reserved
Req 3	eMAC0
Req 2	eMAC1
Req 1	PCI-G-Bus Bridge 0
Req 0	G-Bridge

7.6.27 G-Bus Arbiter Granted Status Register

This register indicates which G-Bus master the G-Bus is granted to.

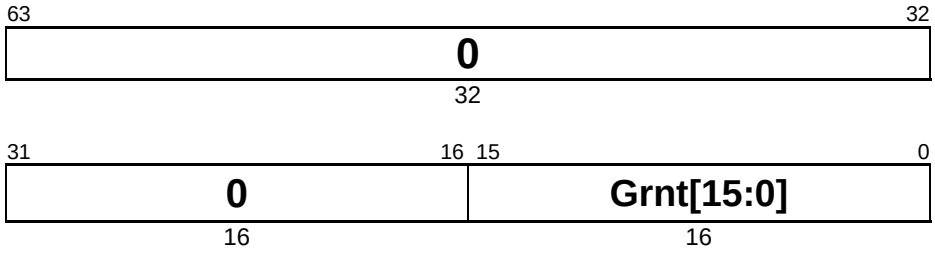


Table 7-30 G-Bus Arbiter Granted Register Fields

Bits	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
15:0	Grnt[15:0]	R/O	G-Bus Granted 0: Granted 1: Not granted	0xFFFE

7.6.28 G-Bus Arbiter Master Status Register

This register indicates the status of the current G-Bus Master. When a master is granted bus ownership, the Bus Broken Timer starts to count down. If the counter counts down to zero before the arbiter receives the gbsgHaveIt signal, the corresponding bit is set and an interrupt is generated to the C790. These bits are only cleared by the Reset signal.

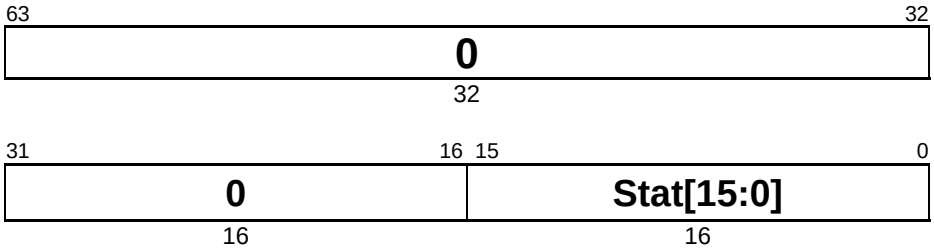


Table 7-31 G-Bus Arbiter Master Status Register Fields

Bits	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0x0000
15:0	Stat[15:0]	R/O	G-Bus Master Status 0: Normal 1: Broken	0x0000

7.6.29 G-Bus Arbiter Control Register

The Control Register enables/disables granting the ownership of G-Bus to G-Bus masters.

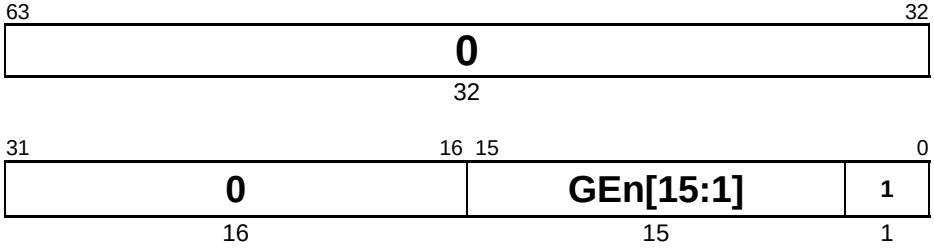


Table 7-32 G-Bus Arbiter Granted Register Fields

Bit(s)	Field	R/W	Description	Initial Value
63:16	–	R/O	Reserved. Must be written as zeroes, and returns zeroes when read.	0
15:1	GEn[15:1]	R/W	G-Bus Grant Enable 1: Enable Grant. 0: Do not enable Grant.	15'b All "1"
0	GEn[0]	R/O	G-Bus Grant Enable of the G-Bridge 1: Enable Grant. 0: Do not enable Grant.	1

8.PCI/G-Bus Bridge

8.1 PCI/G-Bus Bridge (“PGB”)

This section describes the TX7901’s PCI to G-Bus Bridge implementation.

8.1.1 Overview

The PCI bus is an industry-standard computer bus and the G-Bus is an on-chip bus proprietary to Toshiba. The PGB maps transactions between PCI and the G-Bus. The PGB is fully compliant with the PCI 2.1 specifications with minor exceptions, which are described later. It implements delayed transactions and posted writes to improve performance. The PGB also integrates a standard PCI arbiter for five PCI devices.

8.1.1.1 General Specifications

The G-Bus is a Toshiba-proprietary on-chip bus that has the following important characteristics:

- G-Bus operates at up to 66 MHz.
- G-Bus has non-multiplexed 32-bit addressing and 32-/64-bit data transfers.
- G-Bus can support multiple masters.
- G-Bus has a “Retry” mechanism to support delayed transactions.

The PGB is designed with the following features:

- Fully compliant with Revision 2.1 of the PCI bus specification.
  - No support for VGA palette snooping.
- Supports 32-bit PCI bus.
- PCI bus can run up to 66 MHz.
- Forwards memory and I/O cycles from PCI to the G-Bus.
- Forwards memory, I/O cycles, and configuration cycles from the G-Bus to the PCI bus.
- Provides concurrent operation on both the PCI and G-Bus to isolate traffic between the two buses.

- Implements up to eight posted write transactions for PCI memory write commands.
- Implements delayed read transactions for all PCI Master I/O and memory read commands – only one transaction at a time.
- Implements “conditional” delayed read transactions for all GBUS master configuration, I/O, and memory read commands to PCI – only one transaction at a time.
- Implements posted write transactions for all G-Bus master configuration, I/O, and memory write commands to PCI – up to eight transactions at a time.
- Implements mapping functions between G-Bus and PCI addresses:
  - 32-bit G-Bus to 32-bit PCI I/O mapping
  - No support for the ISA-aware mode
- On-chip PCI arbiter for up to five PCI master devices
  - Allows use of external arbiter

The PGB is based on a 66MHz PCI core. The bridge logic interfaces to the PCI core on one side and to the G-Bus on the other side as shown in Figure 8-1.

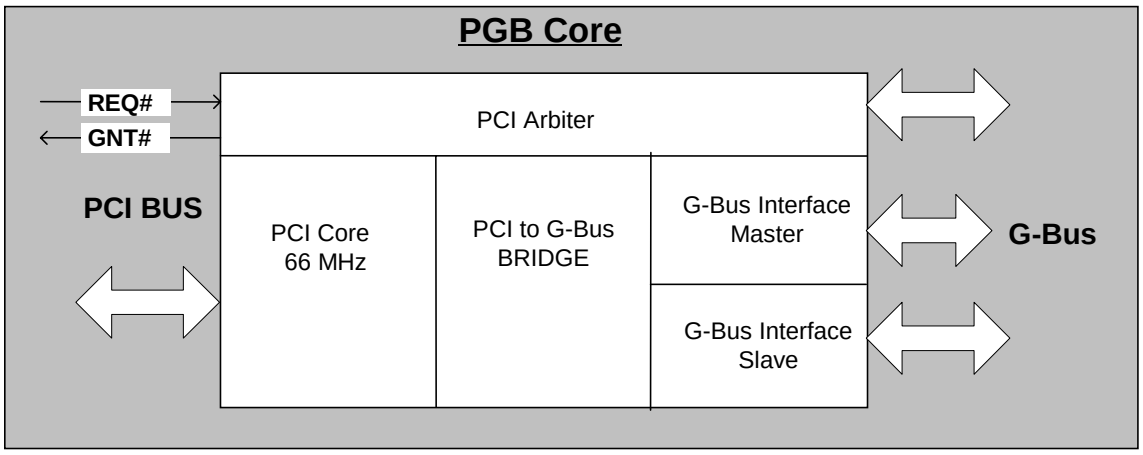


Figure 8-1 Top level Block Diagram

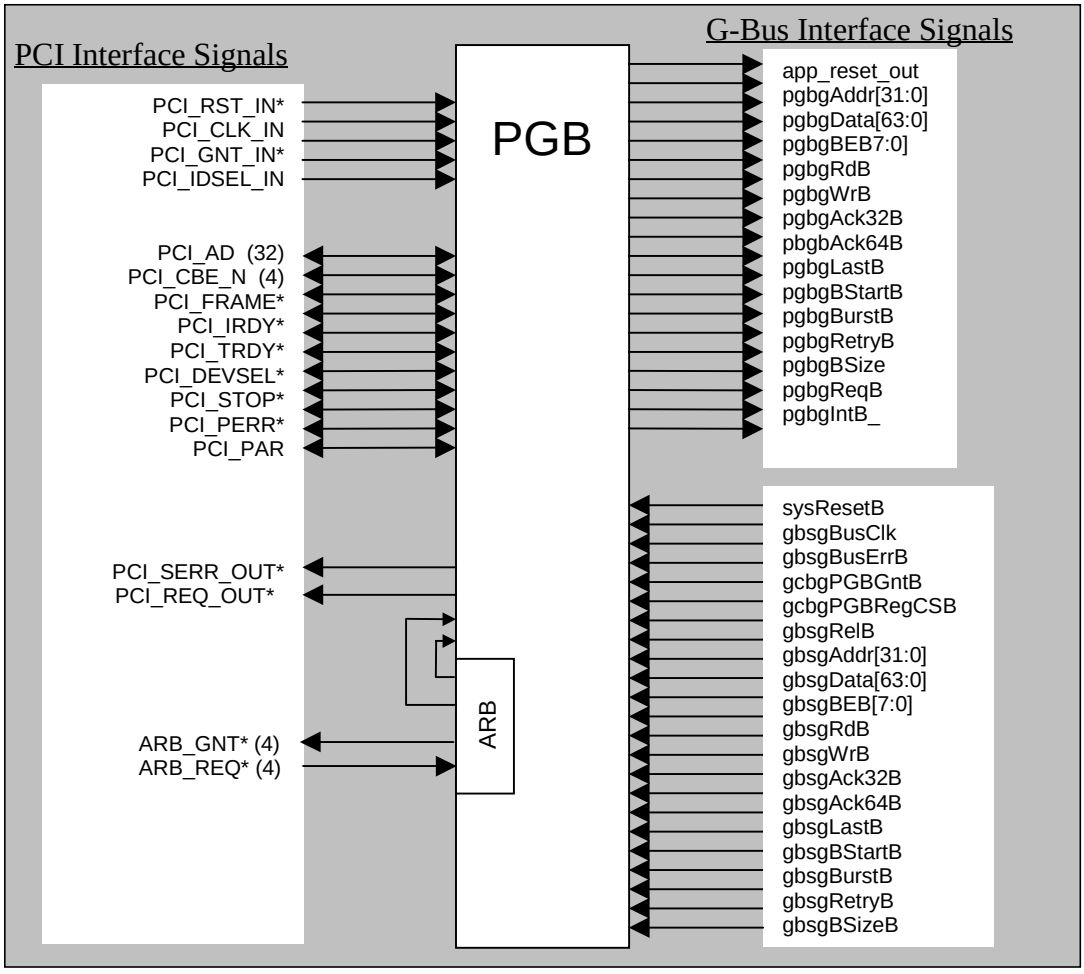


Figure 8-2 PGB Signals

8.1.2 PCI / G-Bus Bridge Interface Signals

Table 8-1 Signal Description

Signal Name	Signal Description
PCI Interface Signals	
PCI_RST_IN*	Reset
PCI_CLK_IN	PCI Clock
PCI_GNT_IN*	Bus Grant
PCI_IDSEL_IN	IDSEL
PCI_AD (32)	Address/Data
PCI_CBE_N (4)	Command/Byte Enable
PCI_FRAME*	Frame
PCI_IRDY*	Initiator Ready
PCI_TRDY*	Target Ready
PCI_DEVSEL*	Device Select
PCI_STOP*	Stop
PCI_PERR*	Parity Error
PCI_PAR	Parity
PCI_SERR_OUT*	System Error
PCI_REQ_OUT*	Request
ARB_GNT* (4)	PGB Arbiter Grant
ARB_REQ* (4)	PGB Arbiter Request
G-Bus Output Signals	
app_reset_out	PCI reset signal fed out to GBUS
pgbgAddr[31:0]	G-Bus Address Out
pgbgData[63:0]	G-Bus Data Out
pgbgBEB[7:0]	G-Bus Byte Enables Out
pgbgRdB	G-Bus Read Out
pgbgWrB	G-Bus Write Out
pgbgAck32B	G-Bus 32 bit Ack Out
pbgbAck64B	G-Bus 64 bit Ack Out
pgbgLastB	G-Bus Last signal Out
pgbgBStartB	G-Bus Start Out
pgbgBurstB	G-Bus Burst Out
pgbgRetryB	G-Bus Retry Out
pgbgBSize	G-Bus Quad-Word Count Out
pgbgReqB	G-Bus Request Out
pgbgIntB	G-Bus Interrupt Out
G-Bus Input Signals	
sysResetB	G-Bus Reset In
gbsgBusClk	G-Bus Clock In
gbsgBusErrB	G-Bus Error In
gcbgPGBGntB	G-Bus Grant In
gcbgPGBRegCSB	G-Bus Chip Select In
gbsgRelB	G-Bus Release In
gbsgAddr[31:0]	G-Bus Address In
gbsgData[63:0]	G-Bus Data In
gbsgBEB[7:0]	G-Bus Byte Enables In
gbsgRdB	G-Bus Read In
gbsgWrB	G-Bus Write In
gbsgAck32B	G-Bus 32 bit Ack In
gbsgAck64B	G-Bus 64 bit Ack In

Signal Name	Signal Description
gbsgLastB	G-Bus Last signal In
gbsgBStartB	G-Bus Start In
gbsgBurstB	G-Bus Burst In
gbsgRetryB	G-Bus Retry In
gbsgBSizeB	G-Bus Quad-Word Count In

8.2 Theory of Operation

The PGB handles read and write transactions between PCI and the G-Bus. This section describes four transaction cases.

Any read crossing the PGB causes writes posted in the same direction to be completed before any read is done. Posted writes through both sides of the PGB can happen concurrently. Posted writes and delayed reads can happen concurrently. The PGB will retry the G-Bus Master if a G-Bus Master and a PCI Master initiate reads simultaneously. Writes can be posted on the PCI side and reads can be delayed on the G-Bus side concurrently. The reverse can also occur.

8.2.1 G-Bus Write to PCI (Bridge Master Write)

To improve performance on the G-Bus, the following Bridge master write strategy is used:

> Receive write data in the PCI write FIFO if there is enough space to hold the entire data burst. Retry the transaction if this is not the case.

Up to eight separate write transactions can be posted at any one time. Maximum burst size supported is 16 words of 64 bits each. When an error occurs, an interrupt is posted to the G-Bus and the address of the corrupted data is latched for use by the error handler. Subsequent errors are ignored during the interrupt pending period. Posted writes from the G-Bus to the PCI bus can happen in parallel with posted writes from the PCI bus to the G-Bus.

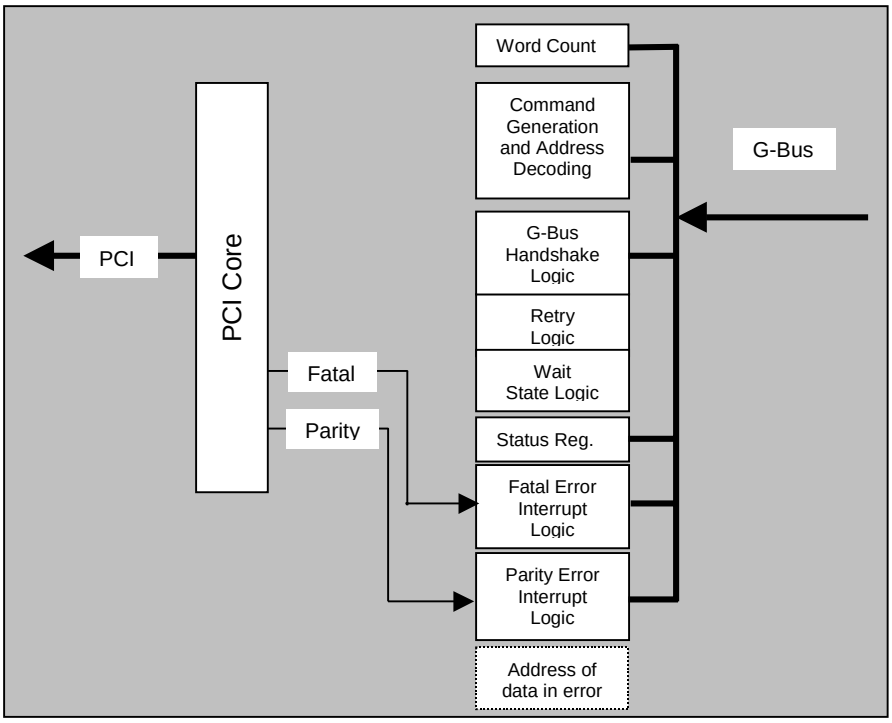


Figure 8-3 Write to PCI from G-Bus

8.2.2 G-Bus Master Reading from PCI (Bridge Master Read)

G-Bus Master reads to PCI targets can pass through one or two phases as described in the following subsections. The state transition diagram for Bridge Master read cycles is shown in Figure 8-5 below. The PGB will retry the G-Bus Master if a G-Bus Master and a PCI Master initiate reads simultaneously.

8.2.2.1 "Wait States" Phase

The PGB initially makes the G-Bus Master wait for PCI read completion. If the read is not completed in (Lt) clock cycles, then the "Wait State" Phase is terminated and the next phase, the "Retry" Phase is started. The value of Lt is programmable at pgbCSR[23:16] and will be set to a default value of "16" at Reset. The PGB causes a G-Bus timeout on all PCI errors (parity or fatal) that happen during this phase. To do this, the PGB stops responding, and causes a timeout (bus error on the G_Bus) to occur.

For proper operation of the G-Bus, the value of Lt should be programmed to be less than the G-Bus slave latency value programmed in the G-Bridge registers.

8.2.2.2 "Retry" Phase

The PGB records the address of the current cycle along with the word count and goes into a delayed read state. The Bridge ignores new read or write transactions from the G-Bus side by issuing Retries. When the read is finished on the PCI side, the PGB waits for the G-Bus Master to re-issue the Delayed Read transaction on the G-Bus (Address and word count compare) and completes the transaction at that time. An "Orphaned Read" results if the G-Bus Master that initiated the Delayed Read does not return to complete the read. The PGB reports orphaned reads by posting an interrupt after a timeout period equal to 2^{16} clocks.

The PGB causes a G-Bus timeout on all PCI errors (parity or fatal) that happen during this phase. To do this, the PGB issues a retry to all cycles with any address not equal to the failed address, and then causes a timeout for the next request to the failed address.

The PGB will flush any posted writes from the G-Bus to PCI before performing any read from a PCI slave. This is performed by first posting a retry to G-Bus Master read requests and then completing the posted write.

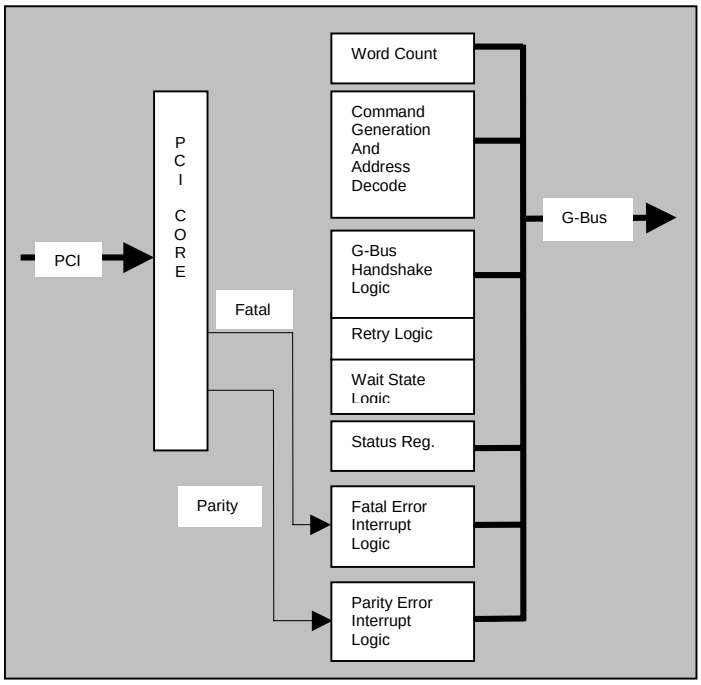


Figure 8-4 G-Bus Master Read from PCI

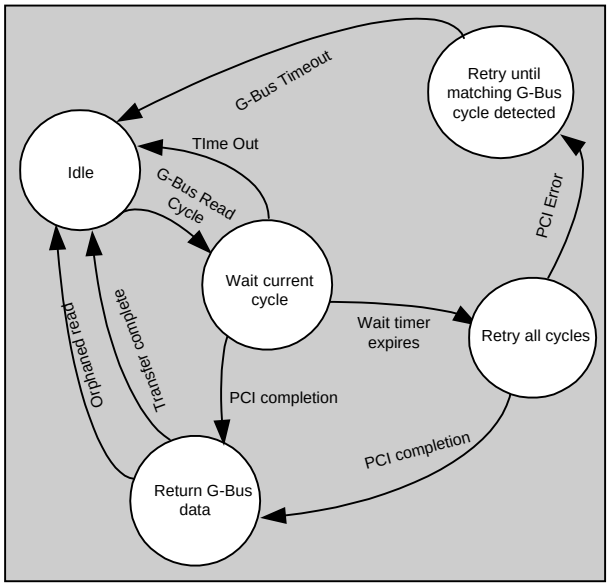


Figure 8-5 State Diagram for G-Bus Master Read from PCI

8.2.3 PCI Master Writing to G-Bus Slave (Bridge Target Write)

The PGB performs posted writes for all memory and I/O write transactions to the G-Bus. PCI configuration transactions to the PGB are not forwarded to the G-Bus.

Although PCI bursts may be of arbitrary length, burst transactions on the G-Bus can only have "packet" sizes as shown in the table below. Because of this, a Bridge target write strategy is implemented as described below.

Table 8-2 G-Bus Burst Sizes

GBTZ	# of Quad-word Equivalent	# of Transfers for 64-bit Device	# of Transfers for 32-bit Devices
000	1	2	4
001	2	4	8

Transactions on the G-Bus, whether for memory or I/O, can also be non-burst 32-bit or 64-bit single data transactions.

8.2.3.1 Posted Write

The PGB breaks large PCI bursts of data into smaller bursts that fit into the PCI Bridge write FIFO. The PGB write FIFO can hold a burst of size two quad-wards. The core provides the word count of the current burst to the G-Bus. Any burst transaction that either comes in with a size not conforming to the sizes of Table 8-2 or has some bytes disabled is broken down into a minimum number of smaller size bursts that conform to the sizes in Table 8-2. Posted writes from the PCI bus to the G-Bus can happen in parallel with posted writes from the G-Bus to the PCI bus. Bursts transmitted onto the G-Bus have all Byte Enables active. When the target is a 32-bit data target, the PGB drives the upper 32 data bits to zero. For such a transaction, the PGB sets the upper four Byte Enable bits to ones. (One means "disabled".)

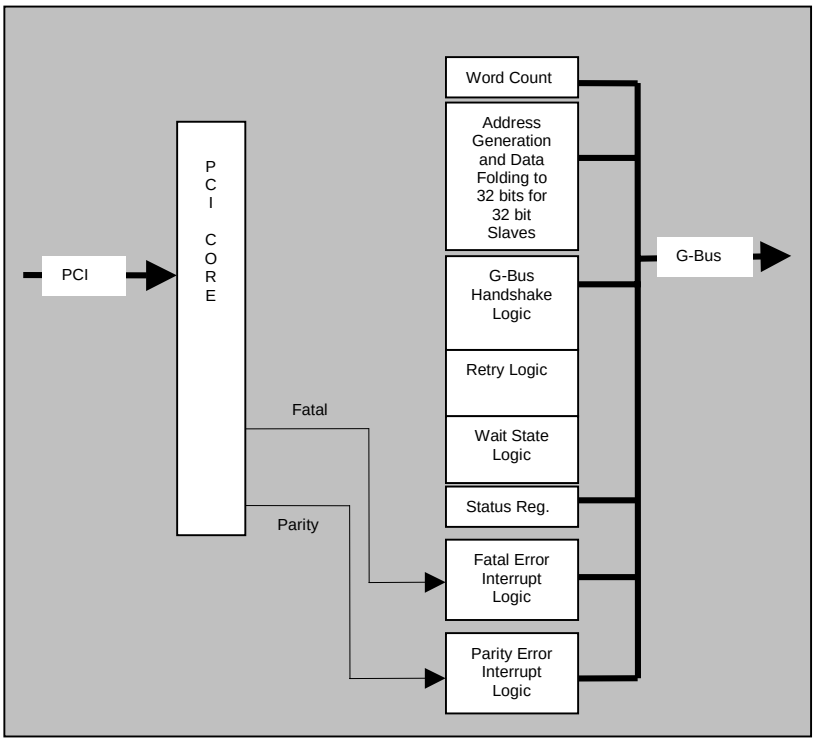


Figure 8-6 PCI Master Writing to the G-Bus

8.2.4 PCI Master Reading from G-Bus Slave (Bridge Target Read)

The core implements a delayed read strategy as described below. The PGB will retry the G-Bus Master if a G-Bus Master and a PCI Master initiate reads simultaneously.

8.2.4.1 Memory Read and I/O Read

A Master on the PCI Bus initiates a read to the PGB. If the data are not returned within 16 clock cycles, the core issues a Delayed Read to the PCI Master but continues fetching the data from the G-Bus side of the core. For both burst and single Memory space transactions, the PGB performs G-Bus pre-fetching until the PCI core indicates “last-word” fetched. When the PCI core indicates “last-word”, the FIFO is flushed. The PGB uses a burst size of four 64-bit words for the speculative reads.

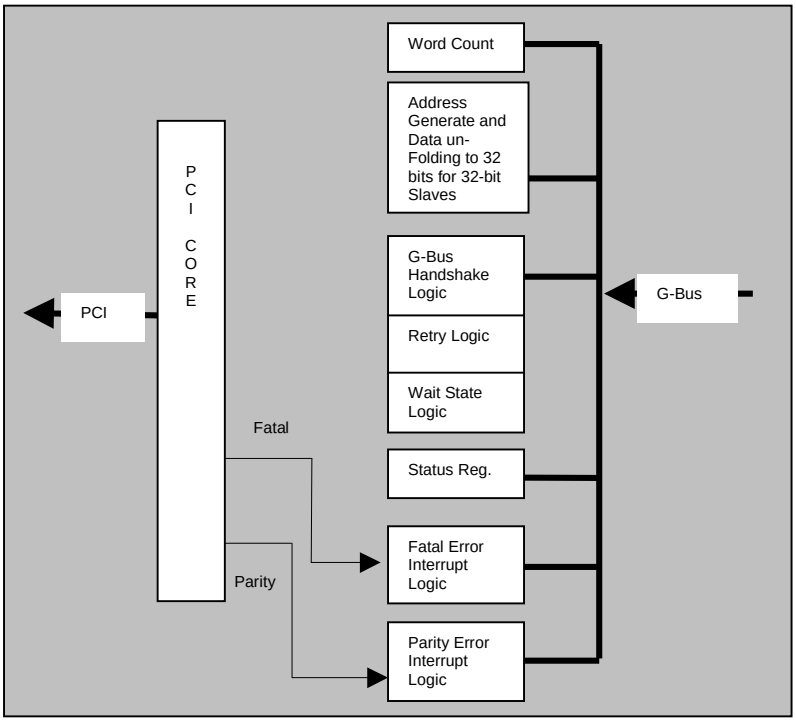


Figure 8-7 PCI Master Reading from G-Bus

8.2.5 Doorbell Feature

The PGB G-Bus Command and Status register has an interrupt bit that may be set by a PCI Master requiring attention from resources on the G-Bus.

8.2.6 PCI Transaction Commands Supported.

The PGB maps G-Bus transactions to any of the 16 possible PCI transactions using a combination of the G-Bus gbsgRdB signal and a 3-bit field in the g2pCycleType register. The PCI core supports a subset of the possible PCI transactions. Table 8-3 shows the supported PCI transaction types in each direction.

Table 8-3 Supported PCI transaction types

C/BE#	Transaction Type	PCI to G-Bus	G-Bus to PCI
0000	Interrupt Acknowledge	No	Yes
0001	Special Cycle	No	Yes
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	No	Yes
0101	Reserved	No	Yes
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	No	Yes

C/BE#	Transaction Type	PCI to G-Bus	G-Bus to PCI
1001	Reserved	No	Yes
1010	Configuration Read	Yes	Yes
1011	Configuration Write	Yes	Yes
1100	Memory Read Multiple	Yes	Yes*
1101	Dual-Address Cycle	Yes	No*
1110	Memory Read Line	Yes	Yes*
1111	Memory Write and Invalidate	Yes	Yes*

*g2pCycleType.type[n] may only be set to the value 0x6 to perform a “memory read multiple” cycle through G2pWindow[n]. A write through G2pWindow[n] with g2pCycleType.type[n] set to 0x6 will result in the PGB issuing a single DAC cycle with no data phase. This is an unsupported PCI transaction and may produce unexpected results. Value 0b111 in type[n] is for cache operation, which requires careful consideration of the cache configuration over the entire system.

8.2.7 Lower Address Bits

For RAM access, the G-Bus address bits gbsgAddr[1:0] are always forced to 0, and during RAM burst transactions, linear sequential addressing is always performed. For RAM transactions, PGB ensures that these lower bits are transmitted as “0” in both directions.

8.2.8 G-Bus to PCI Address Mapping, Addition method

The PGB performs G-Bus to PCI address mapping using the mechanism shown below. The g2pLower and g2pUpper registers hold the lower and upper limits of the windows. These limits lie on double-word boundaries. For example, if the limits of one window are addresses 0x1000 – 0x1400, the lower limit of the next window can be as close as 0x1400. The first window will pass data up to byte 0x13FF inclusive.

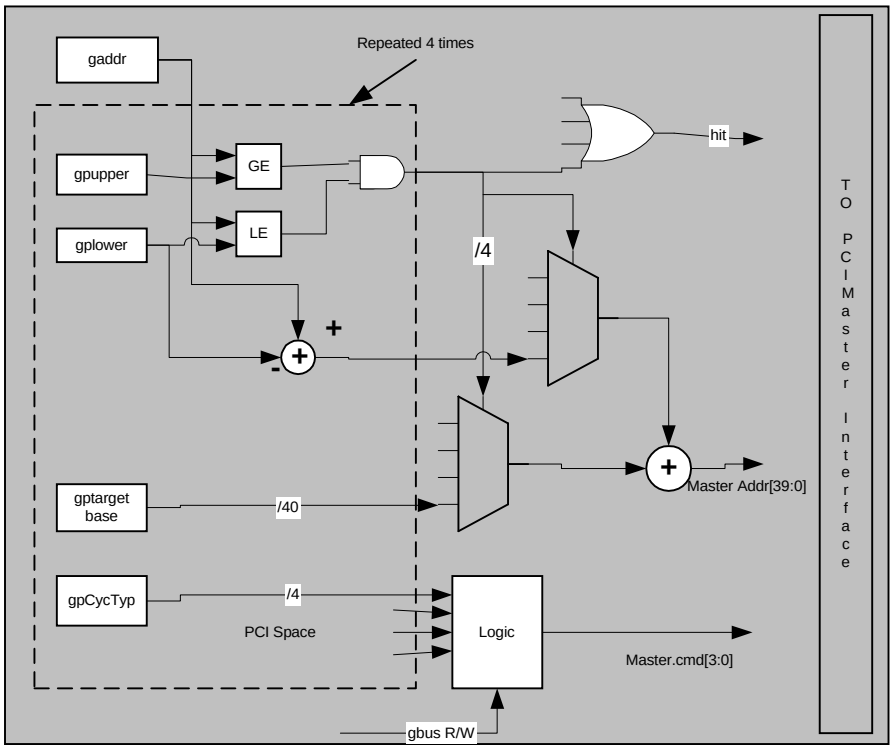


Figure 8-8 G-Bus to PCI Address Mapping, Addition method

8.2.9 PCI to G-Bus Address Mapping

The PGB performs PCI to G-Bus address mapping using the mechanism shown in the figure below.

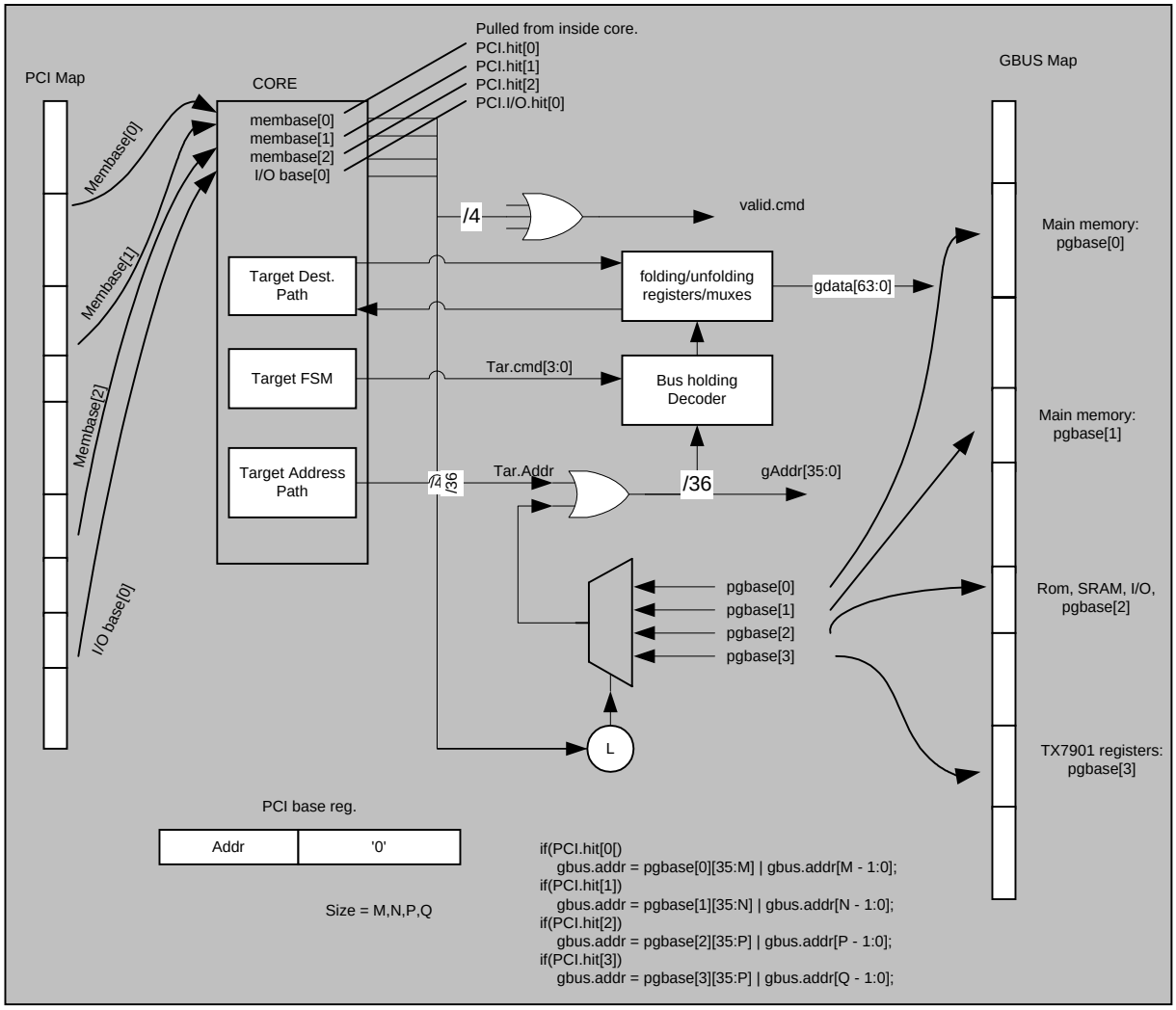


Figure 8-9 PCI to G-Bus Address Mapping

8.2.10 Bus Error Handling Policies

The PCI core supports PCI bus parity and system error capabilities. These errors are forwarded to the application side of the core. The PGB performs the actions described below when errors are detected.

8.2.10.1 Transactions with a PCI Master

PCI parity errors detected by the PGB for transactions with a PCI Master are reported by the core to the PCI Master using the PERR signal. The PGB completes the transaction without reporting errors to the G-Bus. If a PCI to G-Bus write cycle encounters a PCI error, the G-Bus transfer (single or burst) is completed with undefined data. G-Bus errors are not

reported to the PCI Master for PCI Master write operations. G-Bus errors are reported to the PCI Master for PCI Master read operations by an abort. G-Bus errors will be reported on the PCI Bus as a Target Abort regardless of whether this transaction has appeared on the G-Bus or not.

8.2.10.2 Transactions with a G-Bus Master

During PCI transactions that have a G-Bus Master, the PCI bus may respond with a parity error or a fatal error such as a Target Abort, or Non-response. When any of these PCI errors occurs during a G-Bus Master write, the PGB posts an interrupt to the G-Bus, flushes any pending requests, and preserves the G-Bus address of the failed transaction in a 1-deep queue. The PGB makes this failed address available to the G-Bus interrupt handler. During the interrupt pending period, errors subsequent to the first error are ignored.

When any of the above PCI errors occur during a G-Bus Master read, the PGB forces a retry to the current cycle, issues a retry to all other cycles, and then causes a timeout for the next request to the failed address.

An attempt by a G-Bus Master to cross a g2pWindow boundary is an error condition. The PGB posts an interrupt to the G-Bus and the PGB goes through the G-Bus motion for the transaction. The transaction is suppressed at the PGB and does not get to the PCI interface. If this error occurs when the G-Bus Master is performing a read, undefined data are passed to the G-Bus Master. The G-Bus address of the failed transaction is preserved in a 1-deep queue and made available to the G-Bus interrupt handler.

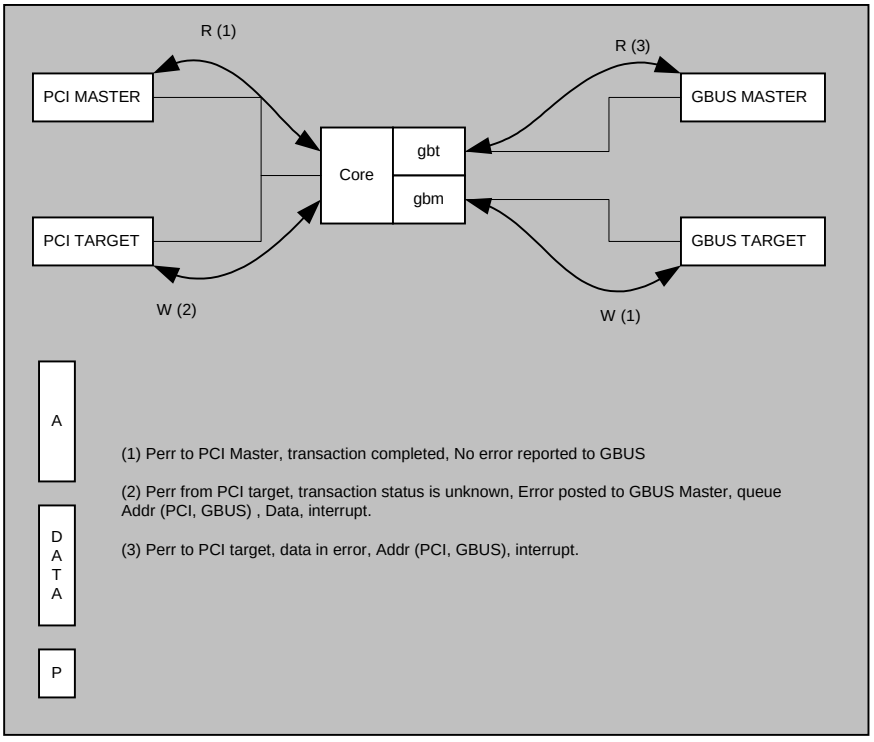


Figure 8-10 Transactions with a G-Bus Master

8.2.11 PCI Bus Arbiter

The PCI Bus Arbiter supports up to five Masters on the PCI Bus, including the PGB. The Arbiter may be enabled or disabled through the G-Bus to allow an external arbiter to be used.

8.2.11.1 PCI Bus Priority

All masters have the same round robin priority.

8.2.11.2 Arbiter Bus Parking

The PCI Bus will be parked at the last Master to have a successful data phase.

8.2.11.3 Retried Masters

When a Master on the PCI Bus is retried, the Arbiter gives ownership of the bus to another Master with active ARB_REQ* signal.

8.2.11.4 Failed Master Lockout

If a Master does not use the PCI Bus for at least 16 PCI clocks after it is granted the bus, it is assumed broken and the Arbiter stops granting it the bus.

8.2.11.5 PCI Arbiter Implementation

The TX7901 allows both internal and external arbiters.

In the case of internal arbiters, PGB requests are connected to Port0 of the arbiter, so there is no need to make a connection to the outside of the chip. In contrast, it is required that Gnt0, a grant from arbiter, is connected to PGB's Gnt outside of the chip. In the single PCI configuration, PCI0 has five ports of arbiter pins outside the chip. In the Dual PCI configuration, PCI0 and PCI1 both have three ports. These arbiter pins are shared in order to save package pins.

In the case of external arbiters, the user may connect Req and Gnt to the user's own arbiters. All the Request signals of internal arbiter, should be connected to 0.

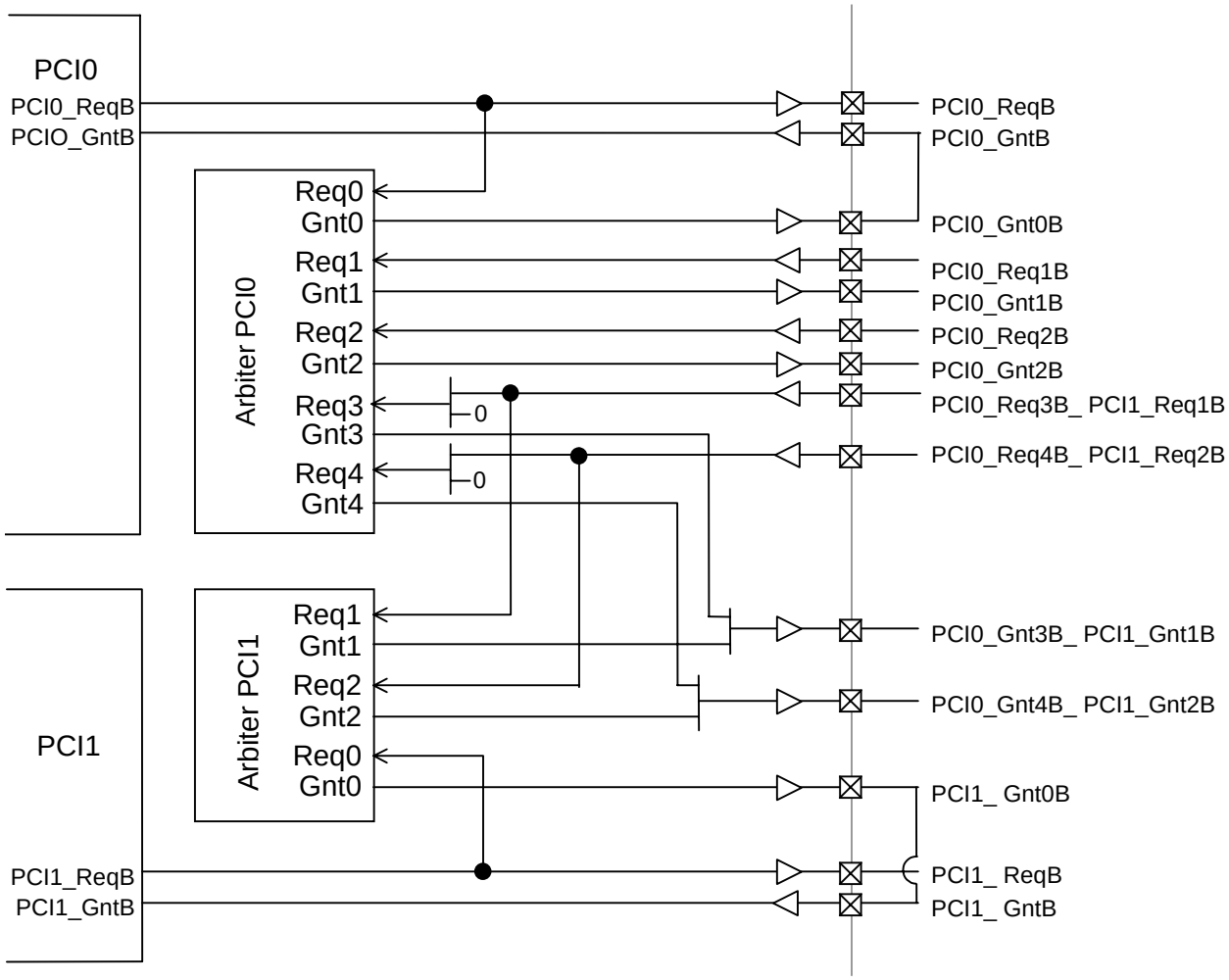


Figure 8-11 PCI Arbiter Implementation

8.2.12 Reset

When a PCI Reset occurs, it is latched as an NMI in the pgbCSR and is reported to the CPU. The actions taken during Reset are described below.

8.2.12.1 PCI Interface

During reset, all bits of the Command Register in the PCI Configuration Space are initialized, and the PCI core target interface tri-states the PCI Bus address/data and parity.

The Hst bit in the pgbCSR register decides whether the PGB is in the “Host” Mode or in the “Satellite” Mode. During PCI Reset, the application should set this bit to “1” to configure the core to perform Host functions, or set this bit to “0” to configure the core to behave as a

Satellite. This bit should not be changed dynamically by the application, which could cause the system to read the wrong PCI config space and disable certain operations.

8.2.12.1.1 Satellite Mode

If the PGB is operating in the “Satellite Mode”, the PCI target Interface is disabled after reset except to Configuration Space access cycles on the PCI Bus. The PCI interface is enabled for target operation when one of bits [1:0] in the PCI Command Register is set by a Configuration Write.

8.2.12.1.2 Host Mode

If the PGB is operating in the “Host Mode”, the PCI target interface is disabled after reset until it is enabled through the Application side, i.e. the G-Bus side. The PCI interface is enabled for target operation when one of bits [1:0] in the PCI Command Register is set by the application.

8.2.12.2 The PCI Arbiter

At Reset, the PCI Arbiter is parked with the ARB_GNT[0]* signal in the active state. The Arbiter remains in this state until RESET is deasserted and a PCI request is received..

8.2.12.3 The G-Bus Interface

At Reset, all G-Bus PCI interface registers are set to initial conditions. Pending interrupt requests are cleared. All transactions and operations in progress are cleared. Bridge transaction processing is disabled. PCI Configuration Space and G-Bus register interface cycles remain enabled.

8.2.13 Retry requests

The PGB performs retry requests on the PCI bus and on the G-Bus for the following operations:

1. The Core performs PCI retries on the PCI bus compliant with the PCI 2.1 standard.
2. A G-Bus Master is retried during the “Retry Phase” of its own read operation. See Section 8.2.2.2.
3. A G-Bus Master is retried at a read attempt during the “Retry Phase” belonging to another G-Bus Master. See Section 10.2.2.2.
4. A G-Bus Master is retried when the G-Bus Master and a PCI Master both request a read simultaneously. See Sections 8.2 and 8.2.2.
5. Retry to G-Bus Master read requests before completing the posted write. See Sections 8.2.2.2
6. The PGB posts an interrupt to the G-Bus when a transaction with a G-Bus Master sees an error on the PCI Bus. All other G-Bus to PCI transactions will be retried.

8.3 PGB Memory Map

The PGB uses a number of memory mapped regions on the G-Bus to provide access to PCI resources. These are controlled and configured through a number of PGB registers mapped into the G-Bus address space. This section describes the registers used to control and configure the PGB.

8.3.1 PCI Configuration Registers

The PCI configuration registers are accessed at G-Bus addresses [0x1E00_3000] through [0x1E00_30FF]. The PGB appears as a normal PCI device (as opposed to a PCI bridge), so the PCI configuration register map reflects a standard PCI target device. The PCI configuration registers are outlined in Table 8-4 below. Details of the PCI configuration bit assignments are given in Section 8.7.

The PGB PCI configuration register map is shown in Table 8-4 below.

Table 8-4 PGB PCI Configuration Register Map

31 16	15 0			
(R/O) Device ID(*)		(R/O) Vendor ID (102Fh)		00h
Status		Command		04h
(R/O) Class Code (000000h)			(R/O) Revision ID (00h)	08h
(R/O) Reserved	(R/O) Header Type (00h)	Master Latency Timer (0)	Cache-line Size (0)	0Ch
Memory Base Address[0]				10h
Memory DAC Base Address[0]				14h
Memory Base Address[1]				18h
Memory DAC Base Address[1]				1Ch
Memory Base Address[2]				20h
Memory DAC Base Address[2]				24h
Reserved				–
(R/O) Subsystem ID (0000h)		(R/O) Subsystem Vendor ID (0000h)		2Ch
Reserved				30h – 3Bh
(R/O) Max_Lat (00h)	(R/O) Min_Gnt (00h)	(R/O) Interrupt Pin (00h)	Interrupt Line	3Ch
Reserved		Retry Time Value	TRDY Timeout	40h
I/O Base Address [0]				44h
Reserved				48h-E7h
p2gBase3[31:0]				E8h
Reserved				ECh - FFh

Note: For TX7901 – PBG0, Device ID = 0040h
For TX7901 – PGB1, Device ID = 0041h

8.3.1.1 PCI Base Address Block Sizes

Table 8-5 shows the PCI window sizes.

Table 8-5 PCI Window Sizes

PCI Configuration Address		Type	Size (bytes)	Corresponding G-Bus Window
44h	IO Base Address[0]	I/O	256	p2gwindow3
20h, 24h	Memory Base Address[2], Memory DAC Base Address[2]	Memory	16M	p2gwindow2
18h, 1Ch	Memory Base Address[1], Memory DAC Base Address[1]	Memory	16M	p2gwindow1
14h, 10h	Memory Base Address[0], Memory DAC Base Address[0]	Memory	16M	p2gwindow0

8.3.1.2 PCI Configuration Register Accessibility

Two Configuration access modes are supported, based on the hostMode bit in the G-Bus control and status register:

- When hostMode is False (the Satellite Mode), Configuration Register accesses by the CPU are not allowed through the G-Bus. Configuration read and write accesses by the CPU are not allowed even through PCI Configuration cycles using a g2p window. Configuration Register read and write accesses are allowed through PCI Bus configuration cycles.
- When hostMode is True, Configuration Register read and write accesses by the CPU are allowed through the G-Bus. Configuration Register accesses by an external PCI Master are not allowed through PCI Bus configuration cycles. However, the external PCI Master can access PCI Configuration Registers through the G-Bus by using a p2g window.

8.3.2 PGB G-Bus Registers

The control and configuration registers for the PGB are memory mapped into the G-Bus address space through a 4 KB range located at [0x1E00_3000] through [0x1E00_3FFF]. A chip select signal selects the Configuration register space during address decoding of G-Bus target cycles to the Configuration registers.

The first 256 bytes are used to map PCI configuration registers onto the G-Bus. The remaining bytes are used to map other PGB control and status registers onto the G-Bus. Table 8-6 shows an address map for PGB Registers.

Table 8-6 PGB Register Address Map

Register Name	G-Bus Address
Timer/Counter	0x1E00_4FFF
	0x1E00_4000
Reserved for PGB	0x1E00_3FFF
	0x1E00_31b0
regSwapCtrl	0x1E00_31a8
p2gSwapCtrl	0x1E00_31a0
g2pSwapCtrl	0x1E00_3198
Ia	0x1E00_3190
p2gBase3	0x1E00_3188
p2gBase2	0x1E00_3180
p2gBase1	0x1E00_3178
p2gBase0	0x1E00_3170
g2pCycleType	0x1E00_3168
g2pBase3	0x1E00_3160
g2pBase2	0x1E00_3158
g2pBase1	0x1E00_3150
g2pBase0	0x1E00_3148
g2pUpper3	0x1E00_3140
g2pLower3	0x1E00_3138
g2pUpper2	0x1E00_3130
g2pLower2	0x1E00_3128
g2pUpper1	0x1E00_3120
g2pLower1	0x1E00_3118
g2pUpper0	0x1E00_3110
g2pLower0	0x1E00_3108
pgbCSR	0x1E00_3100
PCI Configuration and Status Space	0x1E00_30FF
	0x1E00_3000
Chip Configuration and Interrupt Controller	0x1E00_2FFF
	0x1E00_2000

8.3.2.1 PGB Control and Status Register (pgbCSR)

The pgbCSR Register provides overall control and status for the PGB. Ten bits provide control and status. Bits[23:16] are the latency timer for G-Bus delayed reads. G-Bus Access to this register is allowed using the G-Bus single cycle mode.

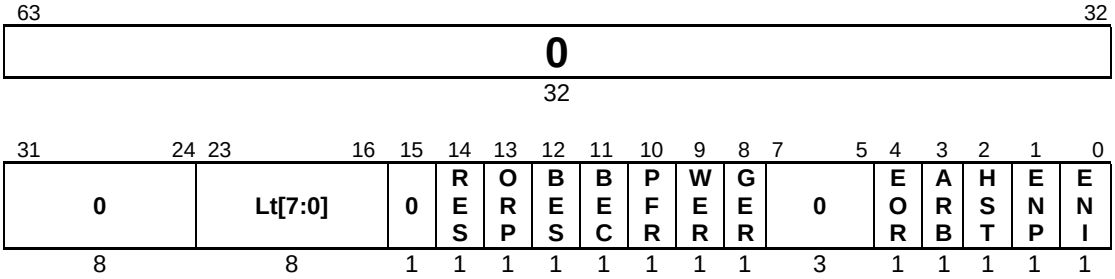


Table 8-7 PGB Control and Status Register Field Descriptions

Bit(s)	Field	R/W	Description
63:24	–	R/O	Reserved. Returns “0” when read. (0)
23:16	Lt[7:0]	R/W	Latency Timer: 8-bit latency timer for G-Bus delayed reads to PCI. (8’h30)
15	–	R/O	Reserved. Returns "0" when read. (0)
14	Res	R/W	PCI Reset: This bit is set when a PCI Reset occurs and the incoming PCI RST input causes a reset. This bit is cleared by a system reset (G-Bus reset) or by writing a “1” to it. (0)
13	Orp	R/W	Orphan: This bit is set when a G-Bus Master does not return to complete a delayed read, causing an orphaned read interrupt. Cleared during reset or by writing a “1” to it. (0)
12	Bes	R/O	Bell Set: writing a “1” to this bit sets the Bec status bit and generates an interrupt. This bit always reads as “0”. (0)
11	Bec	R/W	Bell Clear: Set when a one is written to the Bes bit. Cleared during reset or by writing a “1” to it. (0)
10	Pfr	R/W	PCI Error: Set when a PCI Parity or Fatal error takes place during a PCI transaction with a G-Bus Master. PGB interrupt is generated when an error occurs. Cleared during reset or by writing a “1” to it. (0)
9	Wer	R/W	Window Error: Set when G-Bus Master attempts to cross a g2pWindow. Cleared during reset or by writing a “1” to it. (0)
8	Ger	R/W	G-Bus error: Set by PGB Master on GBUSERR#. Cleared during reset or by writing a “1” to it. (0)
7:5	–	R/O	Reserved. Returns "0" when read. (0)
4	Eor	R/W	Enables Orp provided that Eni does not override Eor. Enables the Orphan interrupt when set. Eni, when cleared, overrides this bit and Orp is disabled regardless of Eor status. Set to "1" during reset. (0)
3	Arb	R/W	Arbiter: Enables the Arbiter. The Arbiter is unaffected by the condition of the other bits. Set to "1" during reset. The contents of this bit should only be changed during PCI Reset. (0)
2	Hst	R/W	Host Mode: Set this bit to “1” for Host Mode of operation or clear it for Satellite Mode operation. Cleared during G-Bus reset. During PCI Reset, the application should set this bit to “1” to configure the core to perform Host functions, or set this bit to “0” to configure the core to behave as a Satellite. This bit should not be changed dynamically by the application, which could cause the system to read the wrong PCI configuration space and disable certain operations. (0)
1	Enp	R/W	Enable PGB: Enables G-Bus transactions involving g2pWindows. Reading this command is equivalent to reading pciCommand[2]. Cleared during reset. (0)
0	Eni	R/W	Enable Interrupts: Enables PGB interrupts. Overrides the Eor bit when cleared. Set to "1" during reset. (0)

8.3.2.2 G-Bus to PCI Memory Address Window Registers

The G-Bus access PCI locations through G-Bus memory address windows called g2pwindows. Each g2pwindow is defined by four registers: g2pUpper, g2pLower, g2pBase, and g2pCycleType. The PGB provides four G2Pwindows.

The four g2pUpper and g2pLower register pairs are compared to the current G-Bus address on each Gbstart cycle. For each pair, if the G-Bus address is greater than or equal to g2pLower and the G-Bus address + (burstSize-1) is less than g2pUpper, the address is judged to be within that G2Pwindow and a PCI cycle is initiated.

When a PCI cycle is initiated, g2pLower is subtracted from the G-Bus address and g2pBase is added to the remainder to produce the effective PCI address. G-Bus access to these registers is allowed using the G-Bus single cycle mode.

Note: The behavior of overlapping windows is undefined.

8.3.2.2.1 g2pUpper Address Registers (g2pUpper0, g2pUpper1, g2pUpper2, g2pUpper3)

The functionality of these registers is described in Section 8.3.2.2 above. The fields of these registers are further detailed in the following figure and Table 8-8.

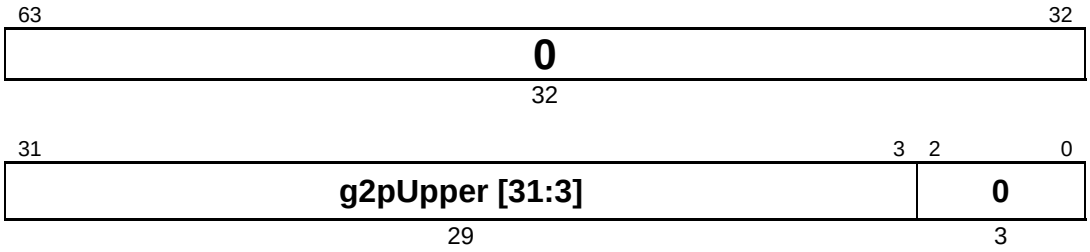


Table 8-8 g2pUpper Address Register Field Definitions

Bit(s)	Field	R/W	Description
63:32	–	R/O	Reserved (0)
31:3	g2pUpper	R/W	Upper boundary of G-Bus Address. Cleared during reset. (0)
2:0	–	R/O	Reserved (0)

8.3.2.2.2 g2pLower Address Registers (g2pLower0, g2pLower1, g2pLower2, g2pLower3)

The functionality of these registers is described in Section 8.3.2.2 above. The fields of these registers are further detailed in the following figure and Table 8-9.

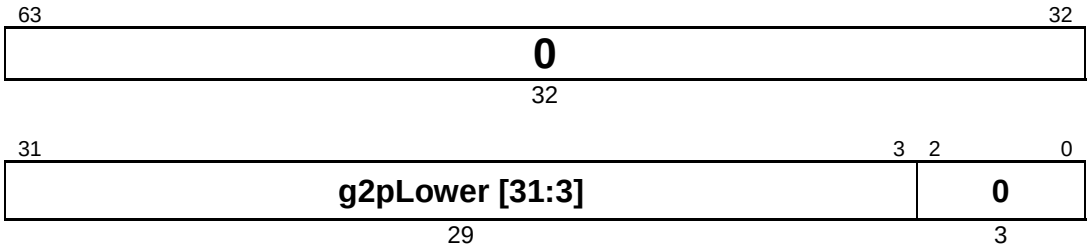


Table 8-9 g2pLower Address Register Field Descriptions

Bit(s)	Field	R/W	Description
63:32	–	R/O	Reserved (0)
31:3	g2pLower	R/W	Lower boundary of the G-Bus Address. Cleared during reset. (0)
2:0	–	R/O	Reserved (0)

8.3.2.2.3 g2pBase Address Registers (g2pBase0, g2pBase1, g2pBase2, g2pBase3)

The functionality of these registers is described in Section 8.3.2.2 above. The fields of these registers are further detailed in the following figure and Table 8-10.

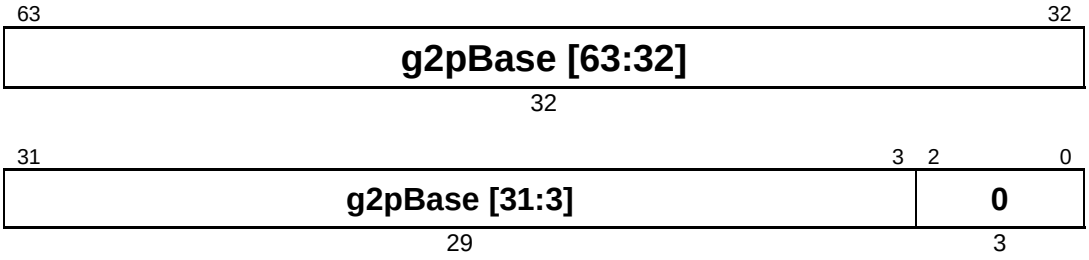


Table 8-10 g2pBase Address Register Field Descriptions

Bit(s)	Field	R/W	Description
63:3	g2pBase	R/W	Address used for base of PCI transaction. Cleared during reset. (0)
2:0	–	R/O	Reserved (0)

8.3.2.2.4 g2pCycleType Register (g2pCycleType)

When a PCI cycle is initiated, g2pCycleType is used to define which of the 15 possible PCI cycle types are performed. G-Bus access to this register is allowed using the G-Bus single cycle mode.

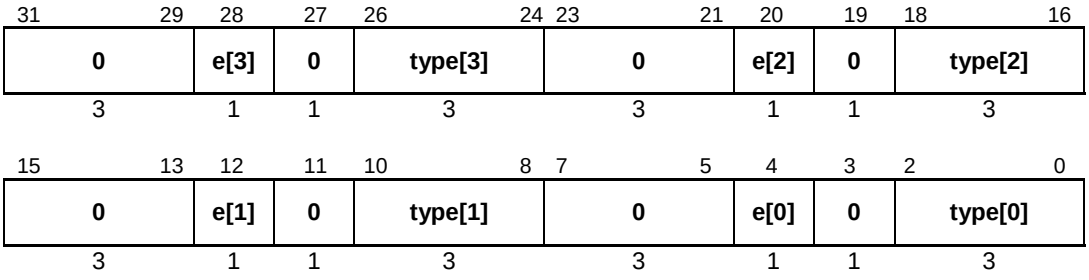


Table 8-11 g2pCycleType Register Field Definitions

Bit(s)	Field	R/W	Description
31:29	–	R/O	Reserved. Read back as “0”. (0)
28	e[3]	R/W	Enable for g2pWindow[3]. Cleared during reset. (0)
27	–	R/O	Reserved (0)
26:24	type[3]	R/W	Assigned to PCI C/BE[3:1]* during PCI address phase through g2pWindow[3]. Cleared during reset. (0)
23:21	–	R/O	Reserved. Read back as ”0”. (0)
20	e[2]	R/W	Enable for g2pWindow[2]. Cleared during reset. (0)
19	0	R/O	Reserved (0)
18:16	type[2]	R/W	Assigned to PCI C/BE[3:1]* during PCI address through g2pWindow[2]. Cleared during reset. (0)
15:13	–	R/O	Reserved. Read back as “0”. (0)

Bit(s)	Field	R/W	Description
12	e[1]	R/W	Enable for g2pWindow[1]. Cleared during reset. (0)
11	–	R/O	Reserved (0)
10:8	type[1]	R/W	Assigned to PCI C/BE[3:1]* during the PCI address phase through g2pWindow[1]. Cleared during reset. (0)
7:5	–	R/O	Reserved. Read back as "0". (0)
4	e[0]	R/W	Enable for g2pWindow[0]. Cleared during reset. (0)
3	–	R/W	Reserved
2:0	type[0]	R/W	Assigned to PCI C/BE[3:1]* during the PCI address phase through g2pWindow[0]. Cleared during reset. (0)

* C/BE[0] is defined as 0: Read, 1: Write.

Note: Table 8-3 shows the g2p transactions supported by the PGB. In order to perform a particular g2p transaction (shown in Table 8-3), set the "type" field in this register to the corresponding C/BE#[3:1]. It is also important to note however that only a Read is allowed when the "type" field is set to 3'b110, and a Write would cause a Dual Address Cycle to be initiated. The PGB does not support Dual Address Cycles, however. Furthermore, the cache configuration over the entire system should be given careful consideration before using cache coherency commands where the type field is 3b111.

8.3.2.2.5 Ia Address Register (Ia)

This register holds the addresses of failing G-Bus Master writes or the addresses of transactions that cause WER errors (Window Crossover Errors). G-Bus Read access to this register is allowed using the G-Bus single cycle mode. The fields of this register are detailed below and in Table 8-12.

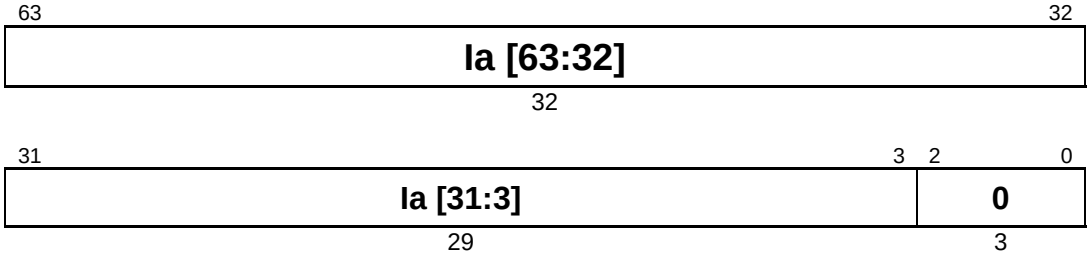


Table 8-12 IA Address Register Field Definitions

Bits	Field	R/W	Description
63:3	Ia	R/W	Address [63:0] for failing transaction. (0)
2:0	–	R/O	Reserved (0)

8.3.2.3 PCI to G-Bus Memory Windows

The PCI accesses G-Bus locations through PCI memory windows called p2gwindows. Each p2gwindow is defined by the normal PCI base register mechanism. The PGB provides four PCI base registers; three DAC memory base pairs, and a single I/O base. These PCI base registers allow for four individual p2gwindows. An additional register called p2gBase provides the base address of the G-Bus transaction performed.

8.3.2.3.1 p2g Base Address Registers (p2gBase)

When a G-Bus cycle is initiated, p2gBase is added to the PCI offset address to produce the effective G-Bus address. G-Bus access to these registers is allowed using the G-Bus single cycle mode.

The p2g Base Address Register fields are detailed below and in Table 8-13.

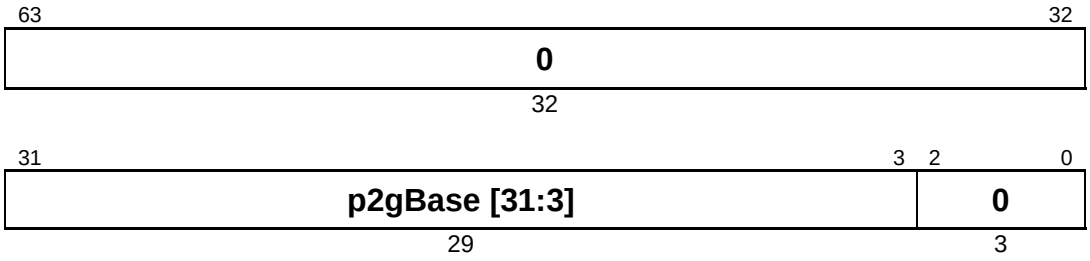


Table 8-13 p2gBase Address Register Field Descriptions

Bits	Field	R/W	Description
63:32	–	R/O	Reserved. Read back at “0”. (0)
31:3	p2gBase	R/W	Address used for base of G-Bus transaction. Cleared on reset. (0)
2:0	–	R/O	Reserved. Read back as “0”. (0)

8.3.2.4 Bi-Endian support

The following registers support the Bi-Endian feature.

8.3.2.4.1 g2pSwapCtrl

The g2pSwapCtrl Register controls the Byte Swapper in the data path from the G-Bus to the PCI Bus.

The PCI Bus is always Little Endian. The Byte Swapper aligns the byte stream when the CPU is in the Big Endian mode.

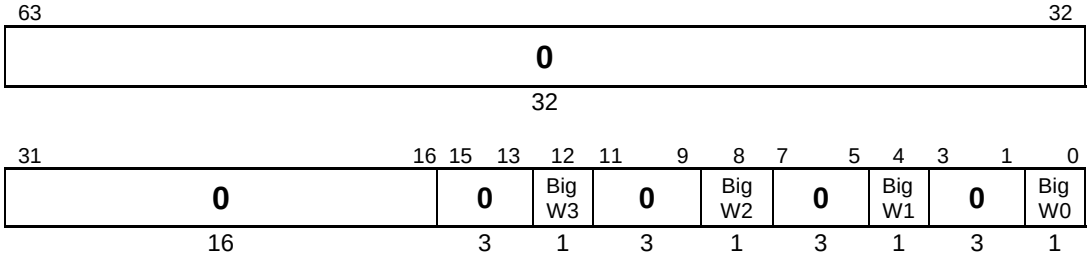


Table 8-14 g2pSwapCtrl Register Field Descriptions

Bit(s)	Field	R/W	Description
63:16	-	R/O	Reserved. 0 for Read operations. Is ignored by Write operations.
15:13	-	R/O	Reserved. 0 for Read operations. Is ignored by Write operations.
12	BigW3	R/W	Window 3 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
11:9	-	R/O	Reserved
8	BigW2	R/W	Window 2 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
7:5	-	R/O	Reserved
4	BigW1	R/W	Window 1 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
3:1	-	R/O	Reserved
0	BigW0	R/W	Window 0 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight

8.3.2.4.2 p2gSwapCtrl

The p2gSwapCtrl Register controls the Byte Swapper in the data path from the PCI Bus to the G-Bus.

The PCI Bus is always Little Endian. The Byte Swapper aligns the byte stream when the CPU is in the Big Endian mode.

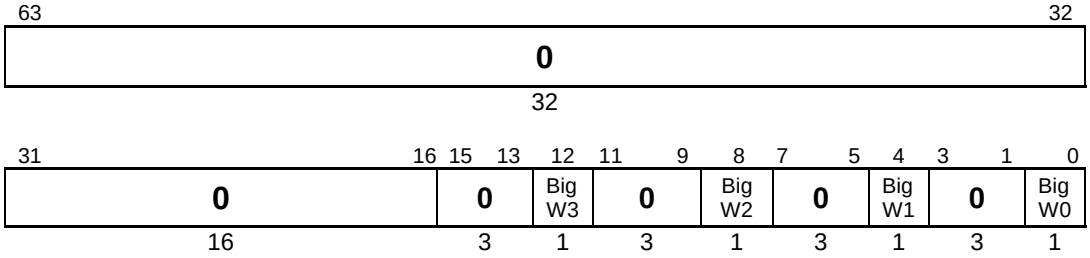


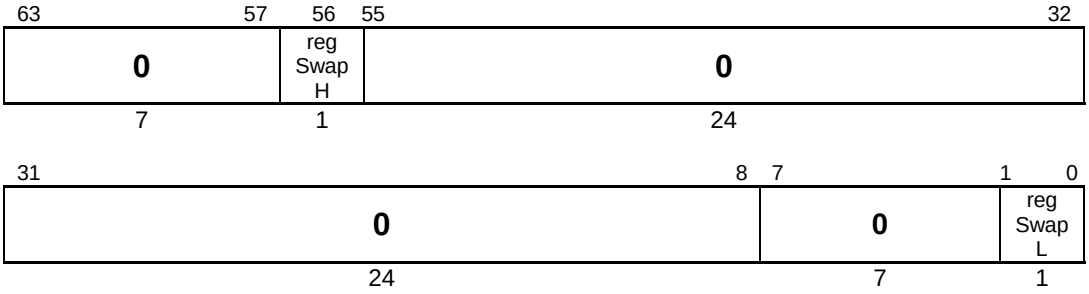
Table 8-15 p2gSwapCtrl Register Field Descriptions

Bit(s)	Field	R/W	Description
63:16	-	R/O	Reserved. 0 for Read operations. Is ignored by Write operations.
15:13	-	R/O	Reserved. 0 for Read operations. Is ignored by Write operations.
12	BigW3	R/W	Window 3 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
11:9	-	R/O	Reserved
8	BigW2	R/W	Window 2 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
7:5	-	R/O	Reserved
4	BigW1	R/W	Window 1 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight
3:1	-	R/O	Reserved
0	BigW0	R/W	Window 0 Swapper. (Default after reset is “1” for Big Endian, “0” for Little Endian) 1: Swap 0: Straight

8.3.2.4.3 regSwapCtrl

The regSwapCtrl Register controls the Byte Swapper in the data path from the G-Bus to the PGB registers. The regSwapper cancels the effect of the p2gSwapper when the CPU is in the Big Endian mode and the p2gSwapper is enabled. Access from the external master on the PCI Bus to the PGB register is always non-swapped.

Note: When in the Big Endian mode, the regSwapper causes PGB registers to be accessed by byte swapped data.



Note: This register is located on the PGB registers. The user must input the same value into regSwapH and regSwapL. Regardless of whether ByteSwapping is performed or not, one of these values is written to the regSwap bit.

Table 8-16 regSwapCtrl Register Field Descriptions

Bit(s)	Field	R/W	Description
63:57	-	R/O	Reserved
56	regSwapH	R/W	Same contents as bit 0. (Default is 0 if Little Endian (ie. Big Endian == 0), 1 if Big Endian == 1.)
55:32	-	R/O	Reserved
31:8	-	R/O	Reserved
7:1	-	R/O	Reserved
0	regSwapL		Swapper in from of the PGB Registers. (Default is 0 if Little Endian (ie. Big Endian == 0), 1 if Big Endian == 1.)

Note: These bits are the same register, but the position of these bits is changed by the swapper. If the same value is written to both bit 0 and bit 56, then this same value will be written regardless of the swapper's direction.

8.4 Register Dual-Porting

The configuration registers of the PGB consist of two groups; namely, the PCI configuration register group and the G-Bus configuration register group. The G-Bus configuration register group is used to configure the G-Bus interface of the PGB and is accessible at all times from the G-Bus. The PCI configuration register group is used to configure the PCI interface of the PGB as defined in the PCI 2.1 standard, but can only be written from the G-Bus side when pgbCSR[2] (PCI Master enable) is set. Two registers are dual-ported between the G-Bus configuration register group and the PCI configuration register group as described in this section.

8.4.1 pgbCSR[1] Dual Porting

When pgbCSR[2] is set (i.e. the PGB is in the “HostMode”), the PCI configuration registers can be written to from the G-Bus, which allows the G-Bus master to take complete control of the PGB PCI configuration interface.

When pgbCSR[2] is cleared (i.e. the PGB is in the “SatelliteMode”), the PCI configuration registers cannot be written to from the G-Bus. If pgbCSR[2] and pciCommand[2] are both cleared, the G-Bus master will not be able to initiate PCI master transactions.

This is why pciCommand[2] is dual ported onto G-Bus register pgbCSR[1]. By dual porting pciCommand[2] onto G-Bus register pgbCSR[1], the PGB can be in the SatelliteMode and a G-Bus master can still become the PCI master.

8.4.2 p2gBase[3] Dual Porting

The p2gBase[3] register is dual ported into PCI configuration register [E8h], making it directly accessible from both sides of the PGB. This allows a PCI master to control the G-Bus target address of a PCI access to the p2gWindow through the PGB IO base address range. By initializing the p2gBase[3] register to point to the base of the G-Bus registers, all other PGB registers can be initialized from the PCI side of the PGB.

8.4.3 Protection Strategy

The PGB can be set to various protected modes of operation as follows by using different combinations of pgbCSR[2:1] and bits 1 (Memory Access Enable) and 0 (I/O Access Enable) of the PCI Command configuration register:

Table 8-17 Protection Levels

Mode	PgbCSR[2:1]	PCI_Command[1:0]	Protection level
Run	0x1	0x1 , 0x2, 0x3	C790 and PCI can access the G-Bus.
Protected	0x2	0x0	PCI cannot access the G-Bus,. C790 cannot access PCI.

8.5 PCI Core

8.5.1 Overview

This section describes the TX7901's PCI core with FIFOs. It covers the 66 MHz Asynchronous Host Bridge and Satellite cores.

8.5.2 Features

The TX7901's PCI Host Bridge and Satellite core provide the following major features:

- 32-bit PCI bus path
- 32- or 64-bit G-Bus data path
- PCI 2.1 compliant
- Master operations and Target operations
- Eight-location FIFO in each data path
- Full Bandwidth Burst Mode
- Memory Write and Invalidate support
- Memory Read Multiple support
- Dual Address Cycle support
- Loadable Configuration Space
- Fast Back-to-Back Target cycles support
- Automatic or Manual retry for most efficient use of PCI bus

8.6 Architecture

8.6.1 Major Internal Modules

This section identifies the main blocks comprising the PCI core. These blocks are listed below and shown in Figure 8-12. Each core provides eight-location dual-port FIFOs to buffer all data paths between the G-Bus and the PCI bus.

Master Write FIFO	Stores data from the G-Bus for Master Write cycles. Dual-ported, 16 x 64b.
Target Read FIFO	Stores data from the G-Bus for Target Read cycles. Dual-ported, 16 x 64b.
Multiplexer Registers	Used by the core to pre-load data from the G-Bus prior to the PCI transfer. The multiplexer registers reduce data transmission latency by storing pre-fetched data.
Output Multiplexer	Multiplexes addresses and data to the output registers.
Output Registers	Drive the PCI I/O buffers with addresses and data.
PCI ADOUT Register	Registers data and addresses from the PCI bus.
Master Read FIFO	Stores data from the PCI bus for Master Read cycles. Dual-ported, 16 x 64b.
Target Write FIFO	Stores data from the PCI bus for Target Write cycles. Dual-ported, 16 x 64b.
Master State Machine	Controls Master transfers, which are initiated by the G-Bus.
Target State Machine	Monitors Target transfers directed to the G-Bus by other devices on the PCI bus.
Parity	Generates and checks parity on incoming and outgoing addresses and data.
Command Decode	Decodes incoming commands from the PCI bus.
Address Compare	Compares incoming addresses with the base address register's contents.
Configuration Registers	Stores configuration information.
PCI Bus Registers	Used for Address translation.

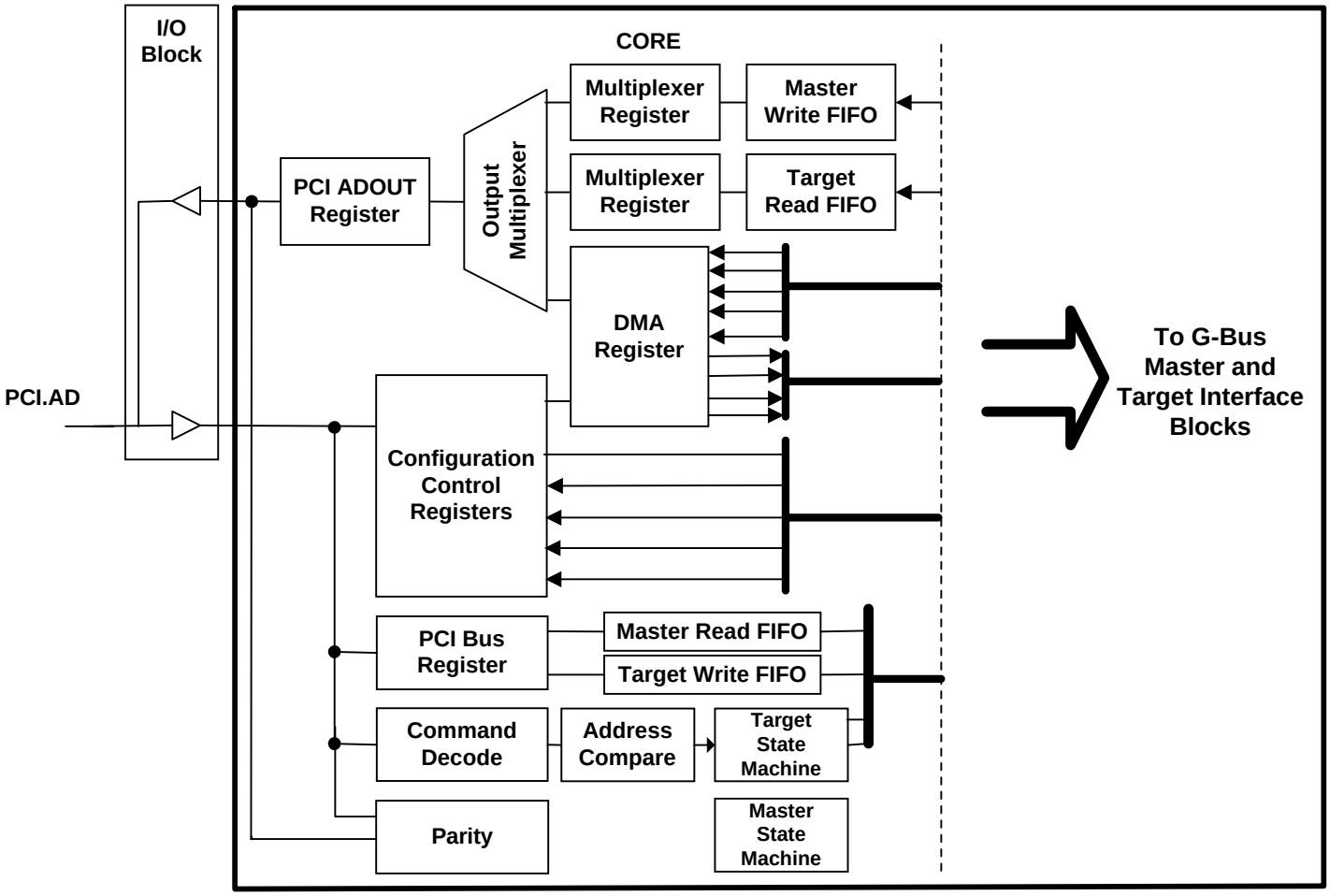


Figure 8-12 High Level Architecture of PCI core

8.6.2 I/O Signals for PCI Core

This section identifies the input and output signals for the PCI core with FIFOs. In Figure 8-13, PCI bus interface signals are shown on the left, and G-Bus interface signals are on the right. I/O cells are not included in this figure. The core provides data_in, data_out, and output_enable signals for connecting to the PCI bus pad ring.

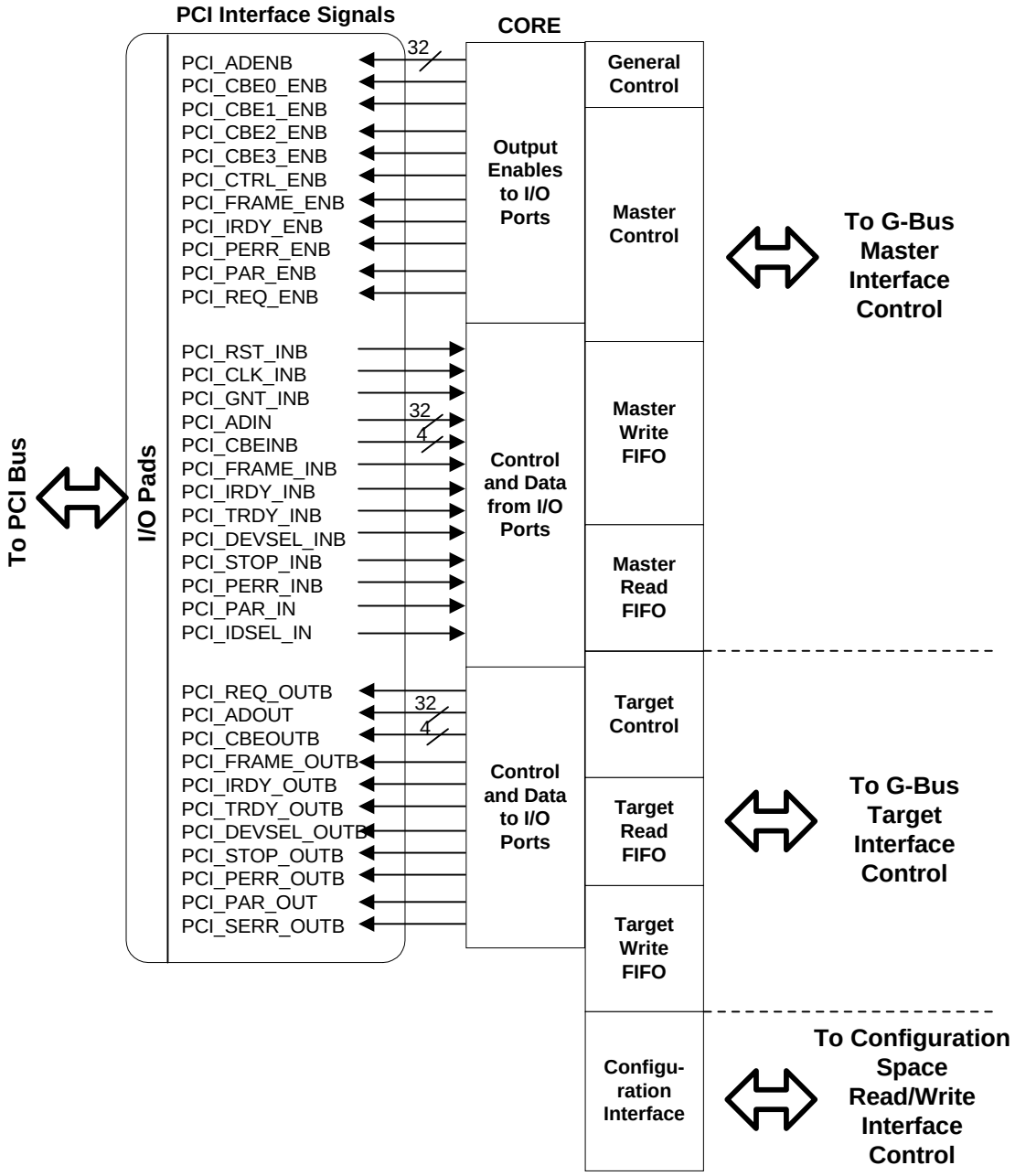


Figure 8-13 PCI and Application Signals for PCI Core

8.6.2.1 PCI Bus Interface Signal List

The tables in this section describe the PCI bus interface signals for the 32-bit PCI to 64-bit application core. Signal names ending with "*" are Active Low.

Table 8-18 Enables from PGB Core to I/O Pads, Listed Alphabetically

Signal	Dir	Width	Description
PCI_ADEN_N	OUT	32	Address/Data Bus Enable for bits 0 to 31
PCI_CBE0_ENB	OUT	1	Command/Byte Enable output enable for byte 0
PCI_CBE1_ENB	OUT	1	Command/Byte Enable output enable for byte 1
PCI_CBE2_ENB	OUT	1	Command/Byte Enable output enable for byte 2
PCI_CBE3_ENB	OUT	1	Command/Byte Enable output enable for byte 3
PCI_CTRL_ENB	OUT	1	PCI_DEVSEL_OUTB, PCI_STOP_OUTB, PCI_TRDY_OUTB, output enable
PCI_FRAME_ENB	OUT	1	PCI_FRAME_OUTB output enable
PCI_IRDY_ENB	OUT	1	PCI_IRDY_OUTB output enable
PCI_PAR_ENB	OUT	1	PCI_PAR_OUTB output enable
PCI_PERR_ENB	OUT	1	PCI_PERR_OUTB output enable
PCI_REQ_ENB	OUT	1	PCI_REQ_OUTB output enable

Table 8-19 Control and Data From I/O Pads to Core, Listed Alphabetically

Signal	Dir	Width	Description
PCI_ADIN	IN	32	Address/Data
PCI_CBEINB	IN	4	Command/Byte Enable
PCI_CLK_IN	IN	1	PCI Clock
PCI_DEVSEL_INB	IN	1	Device Select
PCI_GNT_INB	IN	1	Bus Grant
PCI_FRAME_INB	IN	1	Frame
PCI_ IDSEL_IN	IN	1	IDSEL
PCI_ IRDY _INB	IN	1	Initiator Ready
PCI_PAR_IN	IN	1	Parity
PCI_PERR_IN_EN	IN	1	Parity Error
PCI_RST_INB	IN	1	Reset
PCI_STOP_INB	IN	1	Stop
PCI_TRDY_INB	IN	1	Target Ready

Table 8-20 Control & Data Signals From Core to I/O Pads, Listed Alphabetically

Signal	Dir	Width	Description
PCI_ADOUT	OUT	32	Address/Data
PCI_CBEOUTB	OUT	4	Command/Byte Enable
PCI_DEVSEL_OUTB	OUT	1	Device Select
PCI_FRAME_OUTB	OUT	1	Frame
PCI_IRDY_OUTB	OUT	1	Initiator Ready
PCI_PAR_OUT	OUT	1	Parity
PCI_PERR_OUTB	OUT	1	Parity Error
PCI_REQ_OUTB	OUT	1	Request
PCI_SERR_OUTB	OUT	1	System Error
PCI_STOP_OUTB	OUT	1	Stop
PCI_ TRDY_OUTB	OUT	1	Target Ready

8.6.3 TRDY_TIMEOUT

Lock up could occur if the requested PCI Target responds with a **PCI_DEVSEL*** signal, but does not follow with a **PCI_TRDY** or **PCI_STOP** signal to allow the cycle to complete. To prevent this, the core provides the programmable **TRDY_TIMEOUT** timer to determine the point at which the Master will abandon the cycle. This register, which is at configuration address 0x40, contains a value that is the number of PCI clocks to allow before generating a timeout. The default value for the register is 0x80, which is well in excess of the PCI 2.1 requirement for new devices. A write of "0" to the **TRDY_TIMEOUT** register disables this function so it can be used with any non-compliant legacy devices that may require more time to return a **PCI_TRDY** or **PCI_STOP** signal.

8.6.4 RETRY_TIMEOUT

The second case for which the core includes a timeout register is the case for which the PCI Target device retries beyond the retry count. To prevent this, the core provides the programmable **RETRY_TIMEOUT** register whose value sets the number of retries that the core will perform as a Master before abandoning a cycle. This register is at configuration address 0x41. The default value for the register is 0x80, which is well in excess of the PCI 2.1 requirement for new devices. A write of "0" to the **RETRY_TIMEOUT** register disables this function, so it can be used with any non-compliant legacy devices that may execute more retries.

8.6.5 PCI Target Delayed Read Handling

The core considers a Target Read to be a Delayed Read if the application has not returned data within 16 PCI clock cycles. In this case, the core will issue a PCI bus retry and automatically lock out all other Target Read requests until the application has returned the data and the original requestor has returned to read it. If after 2^{15} PCI clocks the original requestor has not returned to read the data, the core aborts the lockout of Target Read requests and discards the Delayed Read cycle.

Target Writes are still allowed during a Delayed Read.

8.7 Configuration Register Descriptions

8.7.1 PCI Vendor ID Register

Address: 00h

Bits Used: Bits 15:0 are used at this address.

Access: Read-Only

Table 8-21 Configuration PCI Vendor ID Register

Bit(s)	Description	Reset
15:0	Manufacturer ID	0x102F

8.7.2 PCI Device ID Register

Address: 00h

Bits Used: Bits 31:16 are used at this address.

Access: Read-Only

Table 8-22 Configuration PCI Device ID Register

Bit(s)	Description		Reset
15:0	Device ID	PGB0	0x0040
		PGB1	0x0041

8.7.3 PCI Command Register

Address: 04h

Bits Used: Bits 15:0 are used at this address.

Access: Read/Write

Table 8-23 Configuration PCI Command Register

Bit(s)	Description	Reset
15:10	Reserved	0x0
9	Fast Back-to-Back Master Enable	0
8	System Error Enable	0
7	Reserved	
6	Parity Error Enable	0
5	Reserved	
4	Memory Write and Invalidate Enable(MWINV)	0
3	Reserved	
2	Bus Master Enable	0
1	Memory Access Enable	0
0	I/O Access Enable	0

8.7.4 PCI Status Register

Address: 04h

Bits Used: Bits 31:16 are used at this address.

Access: Read Only, Status (Status bits: see PCI 2.1 Specifications for usage)

Reports the status of operations on the PCI bus. Also indicates the **PCI_DEVSEL*** timing that has been selected.

Table 8-24 Configuration PCI Status Register

Bit(s)	Description	Reset	Type
15	Detect Parity Error	0	Status
14	Signaled System Error	0	Status
13	Received Master Abort Status. Set when PCI Master terminates a Host-to-PCI transaction with a Master Abort.	0	Status
12	Received Target Abort Status. Set when the core initiates a PCI transaction and is terminated by the Target.	0	Status
11	Signaled Target Abort Status	0	Status
10:9	Device Select Timing. Indicates timing of **PCI_DEVSEL*** when the core responds to a PCI transaction as a Target.	01	R/O
8	Data Parity Detected	0	Status
7	Fast Back-to-Back Capable Status Flag	1	R/O
6	RESERVED	0	R/O
5	66 MHz-Capable Status Flag	1	Status
4:0	Reserved	0x10 *1	R/O

*1: The PCI Specifications stipulate that this field must be ignored. Please do not rely on this value since it is an implementation dependent value.

8.7.5 Device Revision Identification Register

Address: 08h

Bits Used: Bits 7:0 are used at this address.

Access: Read Only

Table 8-25 Configuration Device Revision Identification Register

Bits	Description	Reset
7:0	Revision Identification Number (Hardwired)	00h

8.7.6 Class Code Register

Address: 08h

Bits Used: Bits 31:8 are used at this address.

Access: Read Only

The Class Code register contains a code value identifying the generic function of this device.

Table 8-26 Configuration Class Code Register

Bits	Description	Reset
23:0	Class Code value	000000h

8.7.7 Cache-Line Size Register

Address: 0Ch

Bits Used: Bits 7:0 are used at this address.

Access: Read/Write

For better performance, use values up to 32 (20h). If this register is “00h”, then “Memory Write and Invalidate” commands will be converted into “Memory Write” commands. Also, “Memory Read Line” and “Memory Read Multiple” commands will be converted into “Memory Read” commands. Refer to sections 3.1.1, 3.1.2 and 6.2.4 of the PCI Local Bus Specifications, Ver. 2.15 for further information.

Table 8-27 Configuration Cache-Line Size Register

Bits	Description	Reset
7:0	Cache-line size (in terms of 32-bit words)	00h

8.7.8 Master Latency Timer Register

Address: 0Ch

Bits Used: Bits 15:8 are used at this address.

Access: Read/Write

The Master Latency Time Register is an 8-bit register controlling the amount of time that the core, as a bus Master, can perform burst transfers if another Master requests the bus. The two least significant bits are hardwired to “0”, allowing interval changes in increments of four clocks.

Table 8-28 Configuration Master Latency Timer Register

Bits	Description	Reset
7:2	Master Latency Timer Count Value: This register sets the minimum number of PCI clock cycles that the core will be guaranteed access to the PCI bus. After this count has expired, the core will surrender the PCI bus as soon as the Arbiter grants the bus to other PCI Master devices.	00h
1:0	Reserved: Hardwired to “0”.	0h

8.7.9 Header Type

Address: 0Ch

Bits Used: Bits 23:16 are used at this address.

Access: Read/Write

Header Type is defined in Section 6.2.1 of the PCI 2.1 Specifications.

Table 8-29 Header Type Register

Bits	Description	Reset
7:0	Header Type	00h

8.7.10 Subsystem Vendor ID

Address: 2Ch

Bits Used: Bits 15:0 are used at this address.

Access: Read-Only.

The Subsystem Vendor ID is defined in section 6.2.4 of the PCI 2.1 Specifications.

Table 8-30 Subsystem Vendor ID Register

Bits	Description	Reset
15:0	Subsystem Vendor ID	0000h

8.7.11 Subsystem ID Register

Address: 2Ch

Bits Used: 31:16 are used at this address.

Access: Read-Only

Subsystem ID is defined in section 6.2.4 of the PCI 2.1 Specifications.

Table 8-31 Subsystem ID Register

Bits	Description	Reset
15:0	Subsystem ID	0000h

8.7.12 Interrupt Line Register

Address: 3Ch

Bits Used: Bits 7:0 are used at this address.

Access: Read/Write

Interrupts are not implemented, so Writes to this register should not be performed. The Interrupt Line is defined in section 6.2.4 of the PCI 2.1 Specifications.

Table 8-32 Interrupt Line Register

Bits	Description	Reset
7:0	Interrupt Line Routing	00h

8.7.13 Interrupt Pin Register

Address: 3Ch

Bits Used: Bits 15:8 are used at this address.

Access: Read-Only

The Interrupt Pin is defined in section 6.2.4 of the PCI 2.1 Specifications.

Table 8-33 Interrupt Pin Register

Bits	Description	Reset
7:0	Interrupt Pin Used	00h

8.7.14 MIN_GNT Register

Address: 3Ch

Bits Used: Bits 23:16 are used at this address.

Access: Read/Write

Table 8-34 MIN_GNT Register

Bits	Description	Reset
7:0	Identifies length of burst period, assuming a 33 MHz clock. Is in units of 0.25 µS.	00h

8.7.15 MAX_LAT Register

Address: 3Ch

Bits Used: Bits 31:24 are used at this address.

Access: Read/Write

Table 8-35 MAX_LAT Register

Bits	Description	Reset
7:0	Sets value of MAX_LAT. See PCI 2.1 Specifications Section 6.2.4 for details. Is in units of 0.25 µS.	00h

8.7.16 TRDY Timeout Value

Address: 40h

Bits Used: Bits 7:0 are used at this address.

Access: Read/Write

Table 8-36 Configuration TRDY Timeout Value

Bits	Description	Reset
7:0	Sets the number of PCI clocks that the core will wait for **TRDY** as the Master.	80h

8.7.17 Retry Timeout Value

Address: 40h

Bits Used: Bits 15:8 are used at this address.

Access: Read/Write

Table 8-37 Configuration Retry Timeout Value

Bits	Description	Reset
7:0	Sets number of retries that the core will perform as Master.	80h

9.DMA Controller

The Direct Memory Access Controller (DMAC) employed in the TX7901 is a flexible direct memory access engine that optimizes the data transfers between the C790 bus and the G-Bus without significant intervention of the core CPU. Instead of having the CPU read data from one source and write it to another, the DMA controllers can be programmed to automatically transfer data independent of the CPU. This frees up the CPU and allows it to continue executing other instructions simultaneous to the transfer of DMA data.

The following is a summary of the DMAC features in the TX7901:

- Eight independent channels of direct memory access (DMA)
- Chaining via link lists of records
- Fixed/round-robin priority arbitration
- 128-bit C790 bus support, capable of transferring 1.6/2.1 GB/second
- Burst transfer of 1 to 8 quad-words on the C790 bus
- Bi-endian support on the C790 bus
- 64-bit G-Bus support, capable of transferring 400/533 MB/second
- 32-/64-bit I/O device support on the G-Bus
- Burst transfer of 1 to 8 quad-words on the G-Bus
- Burst I/O support
- Block & slice support
- Memory byte alignment support on both the C790 bus and the G-Bus
- FIFO queue size of 16 quad-words

Each DMA channel can move data from:

- C790 bus memory to C790 bus memory
- C790 bus memory to G-Bus I/O
- G-Bus I/O to C790 bus memory
- C790 bus memory to G-Bus memory
- G-Bus memory to C790 bus memory
- G-Bus memory to G-Bus memory

9.1 Modes of Operation

The DMAC has eight independent channels. As channels become active, the Arbiter grants control to the highest priority channel. The DMAC then begins reading the source data from the source address and puts the data into the FIFO queue. When the data are ready in the FIFO queue, the DMAC transfers the data out to the destination address. Figure 9-6 shows the DMAC operation.

The data FIFO unit contains a FIFO and an aligner. The FIFO can accommodate 16 quad-words of data. The FIFO gets data from the source device, and the aligner shifts data into the correct destination alignment depending on the destination address. A request is generated to the destination device once one of the following results: a non-aligned write data condition, a FIFO size of more than 8 quad-words, or a “no more read data” condition.

9.1.1 DMA Channel Priority

The DMAC has two types of priority schemes for the DMA channels depending on the setting of the FRP. If FRP=0, a fixed priority scheme is used. If FRP=1, a round-robin priority scheme is used.

When fixed priority is programmed, the DMA channel priority corresponds to the FPL[3:0] field in the CCR registers. FPL[3:0]=0000 is the highest priority whereas FPL[3:0] = 1111 is the lowest priority. If two DMA channels have the same priority level, the DMA channel number determines the priority. The DMA channel with the lower channel number is assigned the higher priority. For example, if channel 1 and channel 5 have the same priority level 3, channel 1 is assigned a higher priority.

If FRP=1, a round-robin priority scheme is used: the next sequential DMA channel after the DMA channel that has the token is assigned the highest priority. Upon power-up, DMA channel 0 has the token. Each time a channel is granted DMA access, the token is transferred to that channel and the next sequential channel is assigned the highest priority. For example, if DMA channel 1 has the token originally, then DMA channel 2 has the highest priority, and DMA channel 3 has the second highest priority. At this time, if DMA channel 4 submits a request and it is granted, the token is transferred from DMA channel 1 to DMA channel 4. Once the token is transferred, DMA channel 5 has the highest priority, and DMA channel 6 has the second highest priority. Figure 9-1 illustrates this round-robin priority scheme described above.

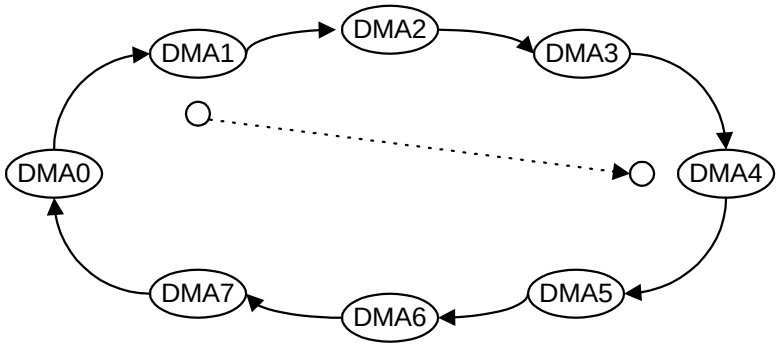


Figure 9-1 Round-Robin Priority Scheme

9.1.2 Source and Destination

The DMAC conducts data transfers within memory or between a memory and an I/O device. The device at the data transfer origin is called a **source** device, and the device at the data transfer destination is called a **destination** device. Memory and I/O devices can be designated as a source device or a destination device. The DMAC can transfer data from the C790 bus memory to any other memory or I/O device. The difference between a memory and an I/O device is the access method to the device. When the DMAC accesses an I/O device, it always uses the same fixed address. On the other hand, when the DMAC accesses memory, it either increments or decrements the address after each access.

9.1.3 Block Transfers

In the block mode, each hardware or software DMA request sends a request to the DMAC to transfer a block of data. The size of the block of data in bytes is in the current byte count register (BCR0 – BCR7).

When a block mode DMA channel is granted access, the actual transfer takes place in two steps.

1. The DMAC arbitrates for the source bus, transfers the data from the source device to the FIFO when granted the source bus, and aligns the data before they are written to the FIFO.

2. The DMAC now arbitrates for bus ownership on the destination bus. When the DMAC is granted the destination bus, it transfers the block of data from the FIFO to the destination device.

During the block mode transfer, the DMAC ignores other DMA requests, and there is no interference from other DMA channels. At the end of the block transfer, the DMAC generates an interrupt to the C790.

The transfer can be from memory to an I/O device, from an I/O device to memory, or from memory to memory.

Table 9-1 shows the types of transfer that can be performed in block mode.

Table 9-1 Block and Slice Transfer Types

Block Mode	Slice Mode
• Memory to I/O • I/O to Memory • Memory to Memory	• Memory to I/O • I/O to Memory • Memory to Memory

9.1.4 Slice Transfers

The slice mode is designed for I/O to memory and memory to I/O transfers. When the I/O device needs more than one slice of data to transfer to/from memory, the I/O device sends a request to the DMAC to perform a slice transfer.

In the slice mode, each hardware or software DMA request sends a request to the DMAC to transfer one slice of data. The size of the slice (SLS[2:0]) of data is selected in the corresponding DMA channel control register (one of CCR0 – CCR7).

When slice mode DMA channel access is granted, the DMAC arbitrates for the source bus to become a bus master. When the DMAC is granted the source bus, it transfers only one slice of data from the source device to the FIFO. The DMAC now arbitrates for the destination bus, and when the bus is granted, it transfers the slice of data from the FIFO to the destination device. Once the slice of data is transferred, the DMAC is ready to serve other DMA channels. Essentially, in the slice mode, the entire DMA transfer is broken down into a number of slice transfers. At the end of the very last slice transfer, the DMAC generates an interrupt to the C790.

The target memory address should be quad-word aligned for efficient data transfer. Otherwise, the number of read/write cycles required for the DMA transfer would increase by one per slice transfer. For example, if the whole DMA transfer requires 300 slice transfers from or to the C790 bus memory, the DMA transfer will increase by 300 C790 bus clock cycles.

9.1.5 C790 Cycle Stealing

If cycle stealing is enabled, the C790 is allowed to steal some cycles from the C790 bus in the middle of a DMA block or slice transfer. Otherwise, during the DMA block or slice transfer, the C790 is idle. This feature is for data synchronization between the DMAC and the C790. When C790 cycle stealing is disabled, the C790 cannot use any system resources until the DMA transfer is complete.

For example, Figure 9-2 illustrates a situation in which DMA channel 5 is programmed in the slice mode, and it requires two slice transfers or five DMA bus cycles to finish the data transfer.

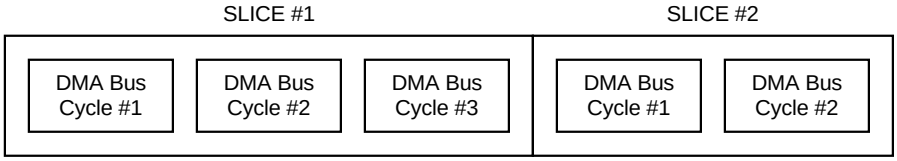


Figure 9-2 DMA Channel 5

When DMA channel 5 is active and cycle stealing is enabled, the C790 can use itself in the middle of a DMA slice cycle. Figure 9-3 shows the sequence of operations on the C790 bus for this scenario. The completion interrupt is generated after Slice #2 DMA Bus Cycle #2 finishes.

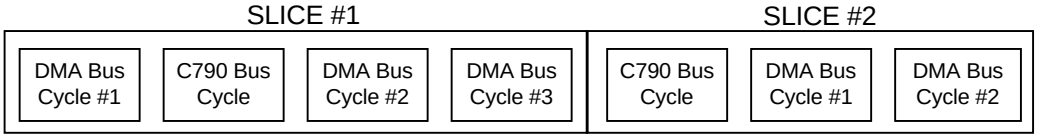


Figure 9-3 C790 Bus Operations With Cycle Stealing

When DMA channel 5 is active and cycle stealing is disabled, the C790 can use itself only between the slice transfers. Figure 9-4 shows the sequence of operations on the C790 bus for this scenario. The completion interrupt is generated after Slice #2 DMA Bus Cycle #2 finishes.

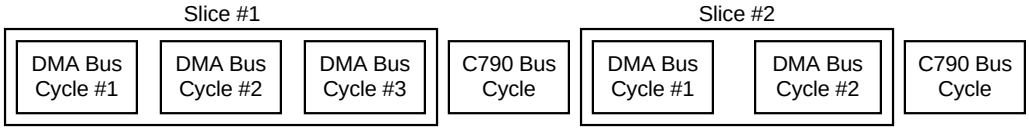


Figure 9-4 C790 Bus Operations Without Cycle Stealing

9.1.6 Chain Mode

In the Chain mode, the DMA Descriptor list located in the local main memory contains all the necessary information for each DMA transfer. Each Descriptor consists of a Source Address, Destination Address, Byte Count, and Next Record Pointer. The Descriptor must be aligned to a 16-byte boundary. All four words in the Descriptor are required and must be kept in the right order. The Descriptor is only located in the local main memory.

A Descriptor list must be constructed before the chained DMA is started. The Next Record Pointer of the first Descriptor points to the address of the second Descriptor and so on until the last Descriptor, in which the Next Record Pointer points to NULL. A Descriptor list is shown in Figure 9-6.

After creating the Descriptor list, the CPU must set the corresponding channel's Next Record Pointer in the DMAC with the address of the first Descriptor. Then, the CPU sets up the corresponding fields in the channel control register. Chained DMAs only work with the Block Transfer mode. Finally, the CPU can start the DMA by setting the EN and STRT bits in the Control Register. The last step can be at the same time as setting up the other fields of the Control Register.

When a Chained DMA is started, the DMAC first fetched the Descriptor from local main memory and writes into the corresponding control registers in the DMAC. Then, the DMAC tries to fetch the data from the source memory pointed to by the source address register and puts it into the destination memory pointed to by the destination address register. After completion of the data transfer, the DMAC fetches the subsequent Descriptor if the Next Record Pointer does not contain the NULL value. The DMAC will repeat the operation until the Next Record Pointer contains the NULL value. A Completion interrupt will then be issued at that time.

9.1.7 Appending to The End of a Chain

It is possible for the application software to append a chain Descriptor when a DMA channel is active. The software first has to construct a new Descriptor in the local main memory. Then, it changes the Next Record Pointer of the current last Descriptor in the Descriptor list to point to the next Descriptor. After that, the software has to check the ACT bit of the Channel Status Register. If it is reset (not active), the software can set the Next Record Pointer and enable the DMA. If this bit is set (Active) and the Next Record Pointer is NULL, the software has to wait for the completion interrupt signals and then set the Next Record Pointer and enable the DMA.

9.1.8 Bus Error

When a bus error occurs on the C790 bus or the G-Bus, the DMAC will record the bus error and generate an interrupt. It then tries to finish the pending DMA read/write cycles and abort any further cycles. After this is done, the DMAC waits for the interrupt routine to service the bus error condition.

9.1.9 32-/64-bit G-Bus I/O

The DMAC supports 32-/64-bit G-Bus I/O using dynamic bus sizing. When the DMAC initiates a G-Bus cycle, the I/O device communicates its device size through gAck32B and gAck64B.

For 32-bit devices, the DMAC reads or writes data on the lower 32 bits of the G-Bus only.

9.1.10 Memory Byte Alignment Support

If the C790 bus memory start address is not quad-word aligned, the DMAC divides the memory access into C790 bus Single operations followed by C790 bus Burst operations. If the C790 bus memory end address is also not quad-word aligned, the DMAC finishes with a single C790 bus Single operation. Figure 9-5 is an example of the cycle division in little endian memory. In this example, the memory transfer is divided into a Single cycle, a Burst cycle of 2 quad-words, two burst cycles of 8 quad-words, and then two single cycles.

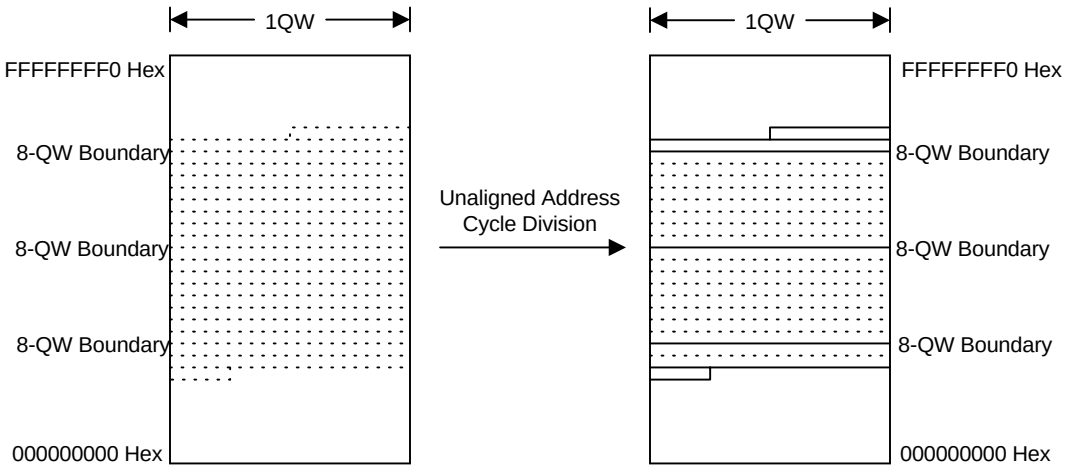


Figure 9-5 C790 Bus Unaligned Address Cycle Break Down

The same occurs in the case of G-Bus memory access. If the G-Bus memory start address is not quad-word aligned, the DMAC divides the memory access into G-Bus Single operations followed by G-Bus Burst operations. If the G-Bus memory end address is also not quad-word aligned, the DMAC finishes with a G-Bus Single operation.

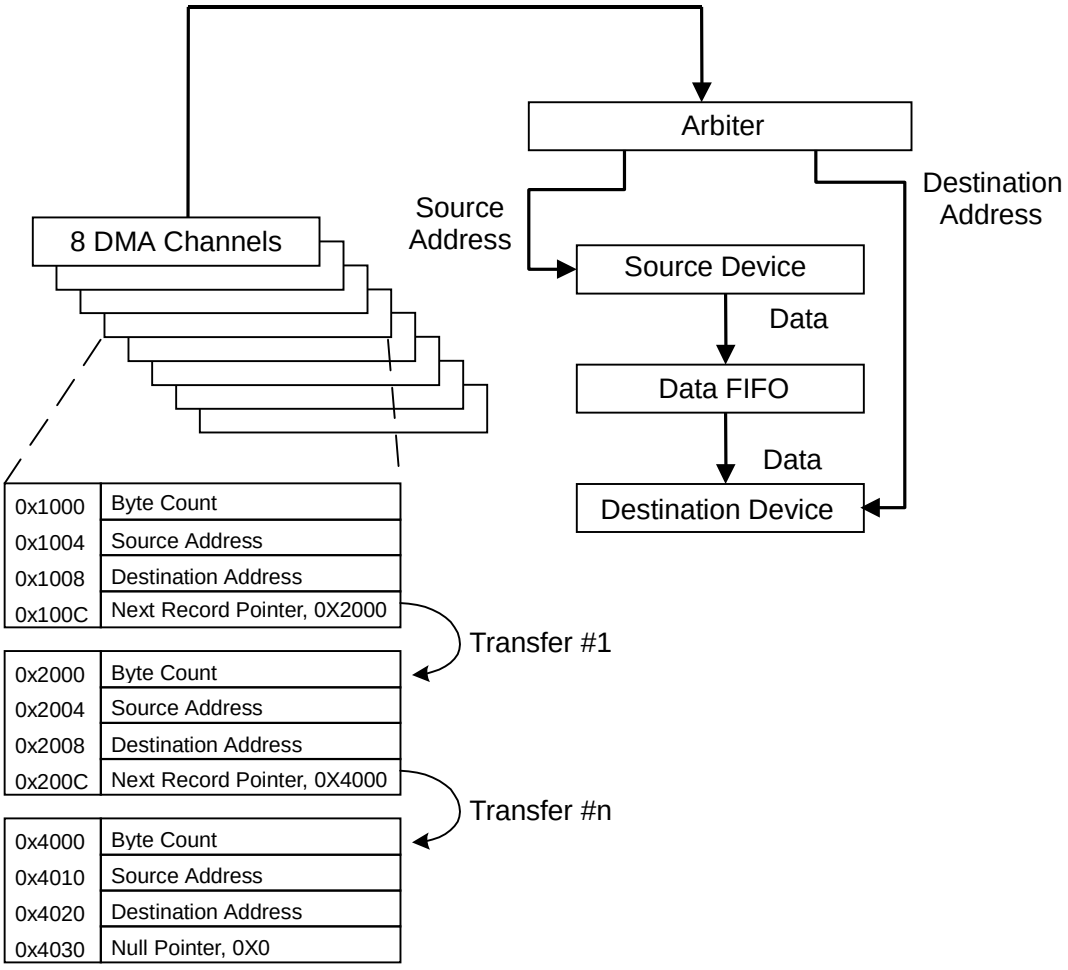


Figure 9-6 DMAC Operation

9.1.11 Restarting a Disabled Channel

(Text Later)

9.1.12 Reprogramming an Active Channel

(Text Later)

9.1.13 Restrictions

(Text Later)

9.2 Registers

Table 9-2 is a summary of all the DMAC registers. The DMAC registers reside on the G-Bus interface. Any G-Bus master can program the DMAC registers using G-Bus single read or write cycles. All DMAC registers are 64 bits wide and byte addressable.

Table 9-2 DMAC Registers

Address	Register Symbol	Register Name
0x1E00_1000	CCR0	Channel 0 Control Register
0x1E00_1010	CSR0	Channel 0 Status Register
0x1E00_1020	SAR0	Channel 0 Source Address Register
0x1E00_1030	DAR0	Channel 0 Destination Address Register
0x1E00_1040	BCR0	Channel 0 Byte Count Register
0x1E00_1050	NRPR0	Channel 0 Next Record Pointer Register
0x1E00_1060 : 0x1E00_10F0	--	RESERVED
0x1E00_1100	CCR1	Channel 1 Control Register
0x1E00_1110	CSR1	Channel 1 Status Register
0x1E00_1120	SAR1	Channel 1 Source Address Register
0x1E00_1130	DAR1	Channel 1 Destination Address Register
0x1E00_1140	BCR1	Channel 1 Byte Count Register
0x1E00_1150	NRPR1	Channel 1 Next Record Pointer Register
0x1E00_1160 : 0x1E00_11F0	--	RESERVED
0x1E00_1200	CCR2	Channel 2 Control Register
0x1E00_1210	CSR2	Channel 2 Status Register
0x1E00_1220	SAR2	Channel 2 Source Address Register
0x1E00_1230	DAR2	Channel 2 Destination Address Register
0x1E00_1240	BCR2	Channel 2 Byte Count Register
0x1E00_1250	NRPR2	Channel 2 Next Record Pointer Register
0x1E00_1260 : 0x1E00_12F0	--	RESERVED
0x1E00_1300	CCR3	Channel 3 Control Register
0x1E00_1310	CSR3	Channel 3 Status Register
0x1E00_1320	SAR3	Channel 3 Source Address Register
0x1E00_1330	DAR3	Channel 3 Destination Address Register
0x1E00_1340	BCR3	Channel 3 Byte Count Register
0x1E00_1350	NRPR3	Channel 3 Next Record Pointer Register
0x1E00_1360 : 0x1E00_13F0	--	RESERVED
0x1E00_1400	CCR4	Channel 4 Control Register
0x1E00_1410	CSR4	Channel 4 Status Register
0x1E00_1420	SAR4	Channel 4 Source Address Register
0x1E00_1430	DAR4	Channel 4 Destination Address Register
0x1E00_1440	BCR4	Channel 4 Byte Count Register
0x1E00_1450	NRPR4	Channel 4 Next Record Pointer Register
0x1E00_1460 : 0x1E00_14F0	--	RESERVED

Address	Register Symbol	Register Name
0x1E00_1500	CCR5	Channel 5 Control Register
0x1E00_1510	CSR5	Channel 5 Status Register
0x1E00_1520	SAR5	Channel 5 Source Address Register
0x1E00_1530	DAR5	Channel 5 Destination Address Register
0x1E00_1540	BCR5	Channel 5 Byte Count Register
0x1E00_1550	NRPR5	Channel 5 Next Record Pointer Register
0x1E00_1560 : 0x1E00_15F0	--	RESERVED
0x1E00_1600	CCR6	Channel 6 Control Register
0x1E00_1610	CSR6	Channel 6 Status Register
0x1E00_1620	SAR6	Channel 6 Source Address Register
0x1E00_1630	DAR6	Channel 6 Destination Address Register
0x1E00_1640	BCR6	Channel 6 Byte Count Register
0x1E00_1650	NRPR6	Channel 6 Next Record Pointer Register
0x1E00_1660 : 0x1E00_16F0	--	RESERVED
0x1E00_1700	CCR7	Channel 7 Control Register
0x1E00_1710	CSR7	Channel 7 Status Register
0x1E00_1720	SAR7	Channel 7 Source Address Register
0x1E00_1730	DAR7	Channel 7 Destination Address Register
0x1E00_1740	BCR7	Channel 7 Byte Count Register
0x1E00_1750	NRPR7	Channel 7 Next Record Pointer Register
0x1E00_1760 : 0x1E00_17F0	--	RESERVED
0x1E00_1800	GCSR	Global Control and Status Register
0x1E00_1810	CBEA	CPU bus Error Address Register
0x1E00_1820	GBEA	G-Bus Error Address Register

Note: Any register address marked “Reserved” reads back as ones, while writes to it are ignored.

9.2.1 Channel Control Registers (CCR0 - CCR7)

The eight CCR registers provide global control for each of the eight channels.

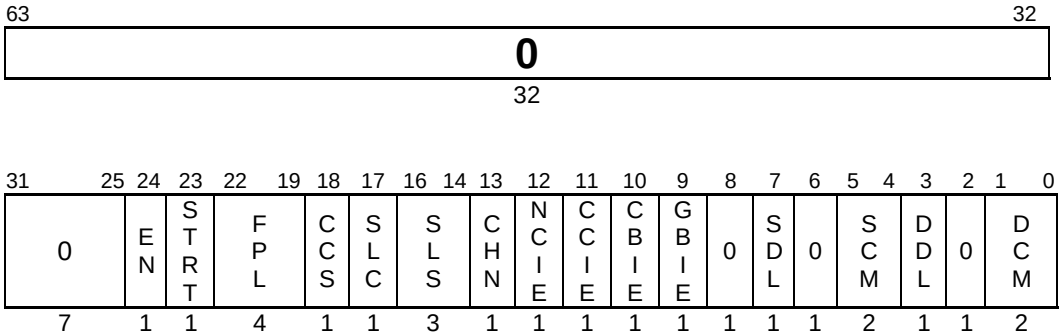


Table 9-3 Channel Control Register Field Descriptions

Bit(s)	Field	R/W	Default	Description
63:25	–	R/O	0	Reserved (0)
24	EN	R/W	0	DMA Channel Enable (0) This bit enables this DMA channel. 0 - Disable the DMA channel. 1 - Enable the DMA channel.
23	STRT	R/W	0	DMA Channel Start (0) This bit is reset automatically once the DMA channel is granted in normal mode or upon the transfer completion of the entire block of data in the chain mode. Writing a “1” sets this bit. 0s written to this bit are ignored. 0 - Ignored 1 - Software DMA request
22:19	FPL	R/W	0	Fixed Priority Level (0000) The fixed priority level is used when the DMA channel is programmed to work with fixed priority GCSR[FRP]=0. Level 0 is the highest priority whereas level 15 is the lowest. If several channels have the same priority, the channel with the lowest physical request number will be granted access. For example, if both DMA Channel 1 and DMA Channel 5 request access at the same time and are at the same priority level, DMA Channel 1 will be granted access. 0000 - Level 0 0001 - Level 1 0010 - Level 2 0011 - Level 3 0100 - Level 4 0101 - Level 5 0110 - Level 6 0111 - Level 7 1000 - Level 8 1001 - Level 9 1010 - Level 10 1011 - Level 11 1100 - Level 12 1101 - Level 13 1110 - Level 14 1111 - Level 15
18	CCS	R/W	0	C790 Cycle Stealing (0) Setting this bit enables the DMAC to allow the C790 to perform cycle stealing. This feature is used to synchronize the data between the DMAC and the C790. Other bus masters are always allowed to perform cycle stealing to minimize the master request-to-grant latency while the DMAC is the master on the C790 bus or the G-Bus. 0 - No cycle stealing 1 - Cycle stealing
17	SLC	R/W	0	Block/Slice Transfer (0) If this bit is set, the DMA channel will transfer a slice of data for a DMA hardware request (dmaReqB) or a DMA software request (STRT). If this bit is reset, the DMA channel will transfer the entire block of data for a DMA hardware or software request. 0 - Block transfer 1 - Slice transfer
16:14	SLS	R/W	0	Slice Size (000) The slice size is used only when SLC is set. 000 - 1 quad-word 001 - 2 quad-words 010 - 3 quad-words 011 - 4 quad-words 100 - 5 quad-words

Bit(s)	Field	R/W	Default	Description
				101 - 6 quad-words 110 - 7 quad-words 111 - 8 quad-words
13	CHN	R/W	0	Chain Mode (0) This bit indicates the Chain mode when it is set. The Chain mode only works in conjunction with the Block mode. This bit indicates the Normal mode when it is reset. 0 - Normal mode 1 - Chain mode
12	NCIE	R/W	0	Normal Completion Interrupt Enable (0) When this bit is set, it enables normal completion interrupts upon the transfer completion of the entire block of data. 0 - Disable normal completion interrupts. 1 - Enable normal completion interrupts.
11	CCIE	R/W	0	Chain Mode Completion Interrupt Enable (0) When this bit is set, it enables chain mode completion interrupts upon the transfer completion of the entire block of data. 0 - Disable chain mode completion interrupt. 1 - Enable chain mode completion interrupt.
10	CBIE	R/W	0	C790 Bus Error Interrupt Enable (0) When this bit is set, it enables C790 bus error interrupts. 0 - Disable C790 bus error interrupts 1 - Enable C790 bus error interrupts
9	GBIE	R/W	0	G-Bus Error Interrupt Enable (0) When this bit is set, it enables G-Bus error interrupts. 0 - Disable G-Bus error interrupts. 1 - Enable G-Bus error interrupts.
8	–	R/O	0	Reserved (0)
7	SDL	R/W	0	Source Device Location (0) This bit indicates whether the source device is located on the C790 bus or the G-Bus. 0 - Source device is on the C790 bus. 1 - Source device is on the G-Bus.
6	–	R/O	0	Reserved (0)
5:4	SCM	R/W	00	Source Device Counting Mode (00) These bits indicate the way the source address is changed after each transfer. The memory device can reside on the C790 bus or the G-Bus, whereas the I/O device resides on the G-Bus only. 00 - No change, no burst transaction (maximum transfer = bus size) (normal I/O) 01 - Reserved 10 - Increment (memory device) 11 - Decrement (memory device) Note: If SCM = 10, then DCM must not be 11. If SCM = 11, then DCM must not be 10.
3	DDL	R/W	0	Destination Device Location (0) This bit indicates whether the destination device is located on the C790 bus or the G-Bus. 0 - Destination device is on the C790 bus. 1 - Destination device is on the G-Bus.
2	–	R/O	0	Reserved (0)
1:0	DCM	R/W	00	Destination Device Counting Mode (00) These bits indicate the way the destination address is changed after each transfer. The memory device can reside on the C790 bus or the G-Bus, whereas the I/O device resides on the G-Bus only. 00 - No change, no burst transaction (maximum transfer = bus size) (normal I/O)

Bit(s)	Field	R/W	Default	Description
				01 - Reserved 10 - Increment (memory device) 11 - Decrement (memory device) Note: If SCM = 10, then DCM must not be 11. If SCM = 11, then DCM must not be 10.

9.2.2 Channel Status Register (CSR0 – CSR7)

These eight registers indicate the status of each of the DMA channels.

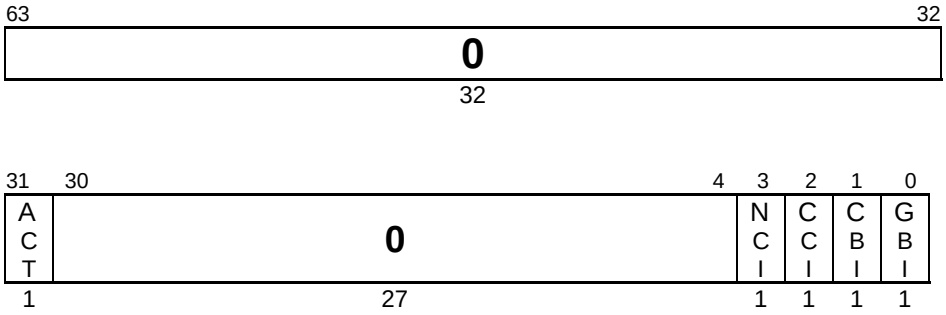


Table 9-4 Channel Status Register Field Descriptions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31	ACT	R/O	0	Active This bit indicates whether the channel is active. 0: Not active 1: Active
30:4	–	R/O	0	Reserved
3	NCI	R/W	0	Normal Completion Interrupt When reading, this bit means the following: 0: No normal completion interrupt. 1: Normal completion interrupt pending. When writing, this bit means the following: 0: The interrupt bit is cleared. 1: The interrupt bit is ignored.
2	CCI	R/W	0	Chain Mode Completion Interrupt When reading, this bit means the following: 0: No chain mode completion interrupt. 1: Chain completion interrupt pending. When writing, this bit means the following: 0: The interrupt bit is cleared. 1: The interrupt bit is ignored.
1	CBI	R/W	0	C790 Bus Error Interrupt If there is a C790 bus error interrupt, the DMAC will finish the pending DMA transfer and stay idle until the C790 bus error interrupt is cleared. In this state, the DMAC will stay idle even if the C790 bus error interrupt is masked (CBIE=0). When reading, this bit means the following: 0: No C790 bus error interrupt. 1: C790 bus error interrupt pending. When writing, this bit means the following:

Bit(s)	Field	R/W	Default	Description
				0: The interrupt bit is cleared. 1: The interrupt bit is ignored.
0	GBI	R/W	0	G-Bus Error Interrupt If there is a G-Bus error interrupt, the DMAC will finish the pending DMA transfer and stay idle until the G-Bus error interrupt is cleared. In this state, the DMAC will stay idle even if the G-Bus error interrupt is masked (GBIE=0). When reading, this bit means the following: 0: No G-Bus error interrupt. 1: G-Bus error interrupt pending. When writing, this bit means the following: 0: The interrupt bit is cleared. 1: The interrupt bit is ignored.

9.2.3 Source Address Registers (SAR0 - SAR7)

These eight registers contain the source addresses of the DMA operation in progress for each of the eight DMA channels.

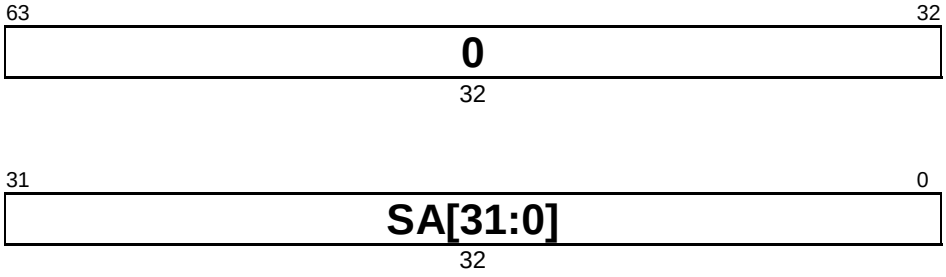


Table 9-5 Source Address Register Field Definitions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31:0	SA	R/W	0	Source Address - SA[31:0] For each DMA read cycle from the source device, the source address is updated depending on the Source Device Counting Mode (SCM). If the channel is programmed in the Chain Mode (CHN = 1), the source address is loaded from the address pointed to by the Next Record Pointer Register when a whole block of data has been completely transferred.

9.2.4 Destination Address Register (DAR0 - DAR7)

These eight registers contain the destination address of the DMA operation in progress for each of the eight DMA channels.

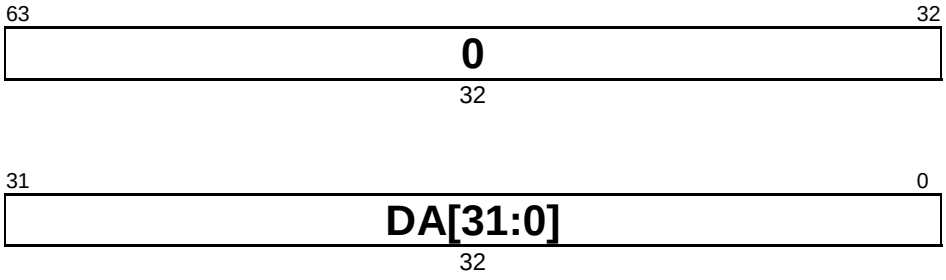


Table 9-6 Destination Address Register Field Definitions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31:0	DA	R/W	0	Destination Address - CDA[31:0] For each DMA write cycle to the destination device, the destination address is updated depending on the Destination Device Counting Mode (DCM). If the channel is programmed in the Chain Mode (CHN=1), the destination address is loaded from the address pointed to by the Next Record Pointer Register when a whole block of data has been completely transferred.

9.2.5 Byte Count Register (BCR0 – BCR7)

These eight registers contain the byte counts of the DMA operation in progress for each of the eight DMA Channels.

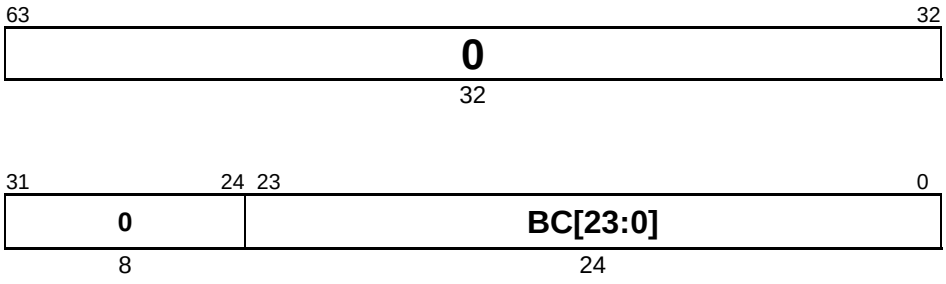


Table 9-7 Current Byte Count Register Field Definitions

Bit(s)	Field	R/W	Default	Description
63:24	–	R/O	0	Reserved
23:0	BC	R/W	0	Byte Count For each DMA write cycle performed, the byte count is decremented. If the channel is programmed in the Chain Mode (CHN=1), the byte count register is loaded from the address pointed to by the Next Record Pointer Register when a whole block of data has been completely transferred.

9.2.6 Next Record Pointer Registers (NRPR0 – NRPR7)

These eight registers contain the address of the next record in the Descriptor list. These registers are only used when the DMA channel is configured in the Chained mode. A Null value for a register indicates the last Descriptor in the list.

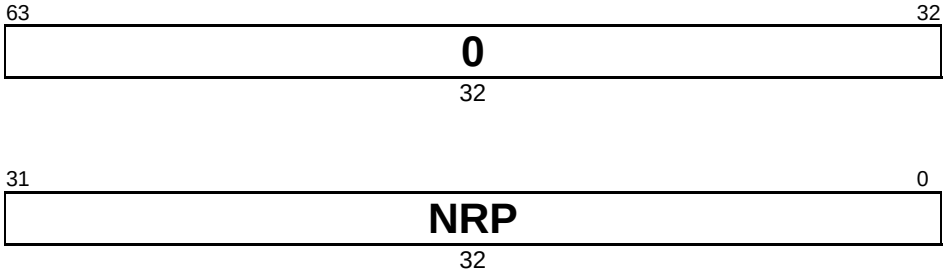


Table 9-8 Next Record Pointer Register Field Definitions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31:0	NRP	R/W	0	Next Record Pointer Points to the next record. A "Null" value indicates the end of the list.

9.2.7 Global Control and Status Register (GCSR)

This register controls the global behavior of the DMAC.

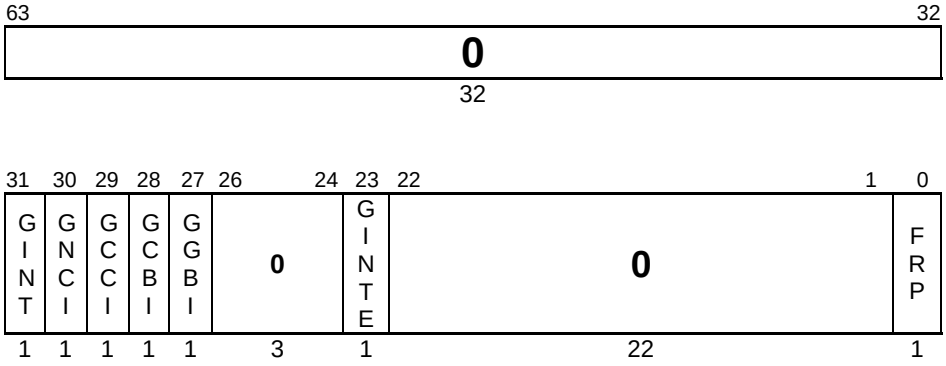


Table 9-9 Global Control and Status Register Field Descriptions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31	GINT	R/O	0	DMA Global Interrupt Active 0: No interrupt pending 1: Interrupt pending
30	GNCI	R/O	0	DMA Global Normal Completion Interrupt 0: No normal completion interrupt in any DMA channel 1: Normal completion interrupt in some DMA channels

Bit(s)	Field	R/W	Default	Description
29	GCCI	R/O	0	DMA Global Chain Mode Completion Interrupt 0: No Chain Mode completion interrupt in any DMA channel 1: Chain Mode completion interrupt in some DMA channels
28	GCBI	R/O	0	DMA Global C790 Bus Error Interrupt 0: No C790 bus error interrupt in any DMA channel 1: C790 bus error interrupt in some DMA channels
27	GGBI	R/O	0	DMA Global G-Bus Error Interrupt 0: No G-Bus error interrupt in any DMA channel 1: G-Bus error interrupt in some DMA channels
26:24	–	R/O	0	Reserved
23	GINTE	R/W	0	DMA Global Interrupt Enable 0: Disable All DMAC Interrupts. 1: Enable All DMAC Interrupts.
22:1	–	R/O	0	Reserved
0	FRP	R/W	0	Fixed/Round-robin Priority 0: Fixed 1: Round-robin

9.2.8 C790 Bus Error Address Register (CBEADDR)

Table 9-10 lists the fields of the C790 Bus Error Address Register.

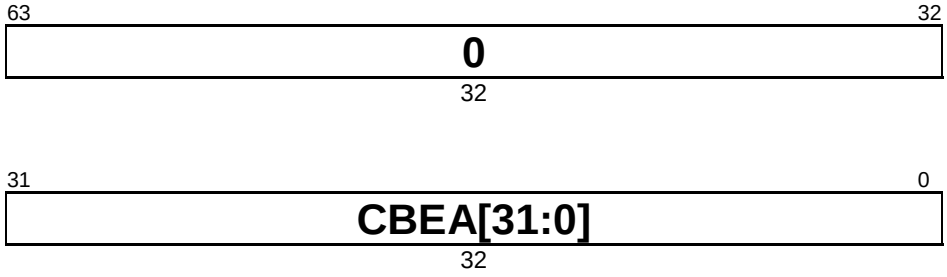


Table 9-10 C790 Bus Error Address Register Field Descriptions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31:0	CBEA	R/O	0	C790 Bus Error Address When the DMAC is the master on the C790 bus and encounters a bus error, the first bus error address is recorded in this register. Further bus error addresses are ignored until the C790 bus error interrupt (CBI) is cleared.

9.2.9 G-Bus Error Address Register (GBEADDR)

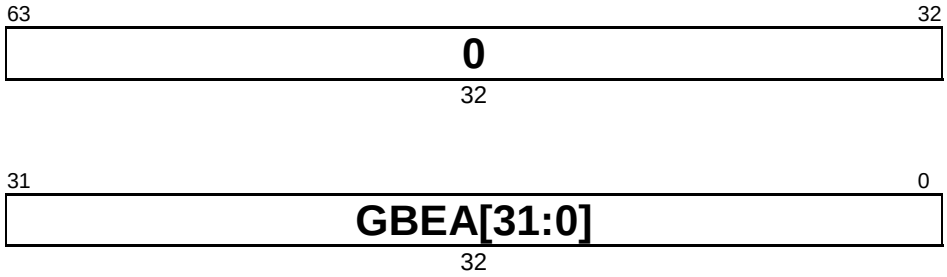


Table 9-11 lists the fields of the G-Bus Error Address Register.

Table 9-11 G-Bus Error Address Register Field Descriptions

Bit(s)	Field	R/W	Default	Description
63:32	–	R/O	0	Reserved
31:0	GBEA	R/O	0	G-Bus Error Address When the DMAC is the master on the G-Bus and encounters a bus error, the first bus error address is recorded in this register. Further bus error addresses are ignored until the G-Bus error interrupt (GBI) is cleared.

10. Programmable Timer/Counters

10.1 Features

The TX7901 Programmable Timer/Counters consist of three timer "channels", Timer 0, Timer 1, and Timer 2. Timer 0 operates only in the Interval Timer Mode. Timer 1 is capable of operating only in the Interval Timer and Pulse Generation Modes. Timer 2 is capable of operating in the Interval Timer, Pulse Generator, and Watchdog Timer Modes. All three modes of operation utilize a 24-bit up counter. There is one counter per timer channel. The three counter modes operate as described below.

10.1.1 Interval Timer Mode

In this mode, the timer generates periodic interrupts and can be configured to use internal or external clock inputs to trigger the counter. The external clock input can be configured to trigger the counter on its rising or falling edge.

10.1.2 Pulse Generator Mode

In this mode, the Timer output can be configured to provide waveforms of specific frequency and duty ratio.

10.1.3 Watchdog Timer Mode

In this mode, the timer monitors and times out system runaway conditions.

10.1.4 Internal or External Clock Division

The internal or external clock source selected for a particular counter may be individually scaled down to 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of its input frequency.

Note: When using an external clock, its frequency should be less than or equal to one half that of the internal G-Bus clock (i.e. less than or equal to 33/25 MHz).

Table 10-1 Timer Modes and Channels

Mode	Description	Channel		
		0	1	2
Interval Timer	Periodic interrupt	✓	✓	✓
Pulse Generator	Provides waveform of specified frequency and duty rate.		✓	✓
Watchdog Timer	Detects runaway condition. Causes NMI or sysReset.			✓

10.2 Block Diagrams

Figure 10-1 shows the block diagram for the programmable Timer/Counters and their connections inside the TX7901. This is followed in Figure 10-2 by a block diagram of the internal connections within a maximally equipped timer such as Timer 2. The internal connections within Timer 0 and Timer 1 are similar to Timer 2 except for each of them lacking some features present in Timer 2.

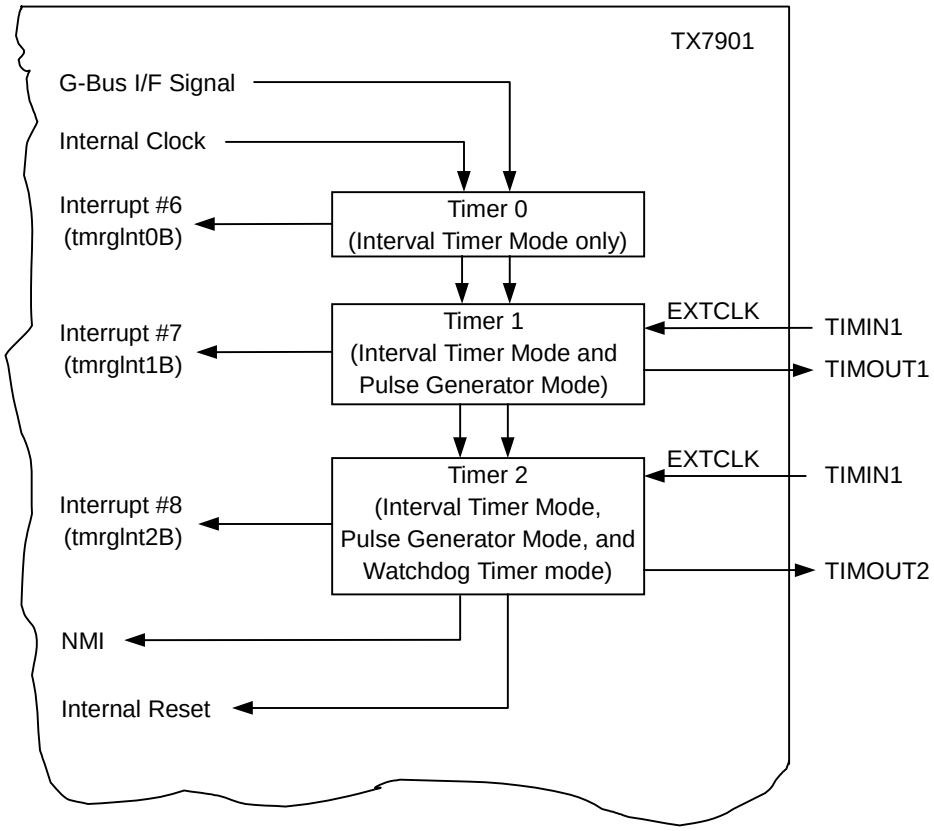


Figure 10-1 Timer Module Connections inside the TX7901

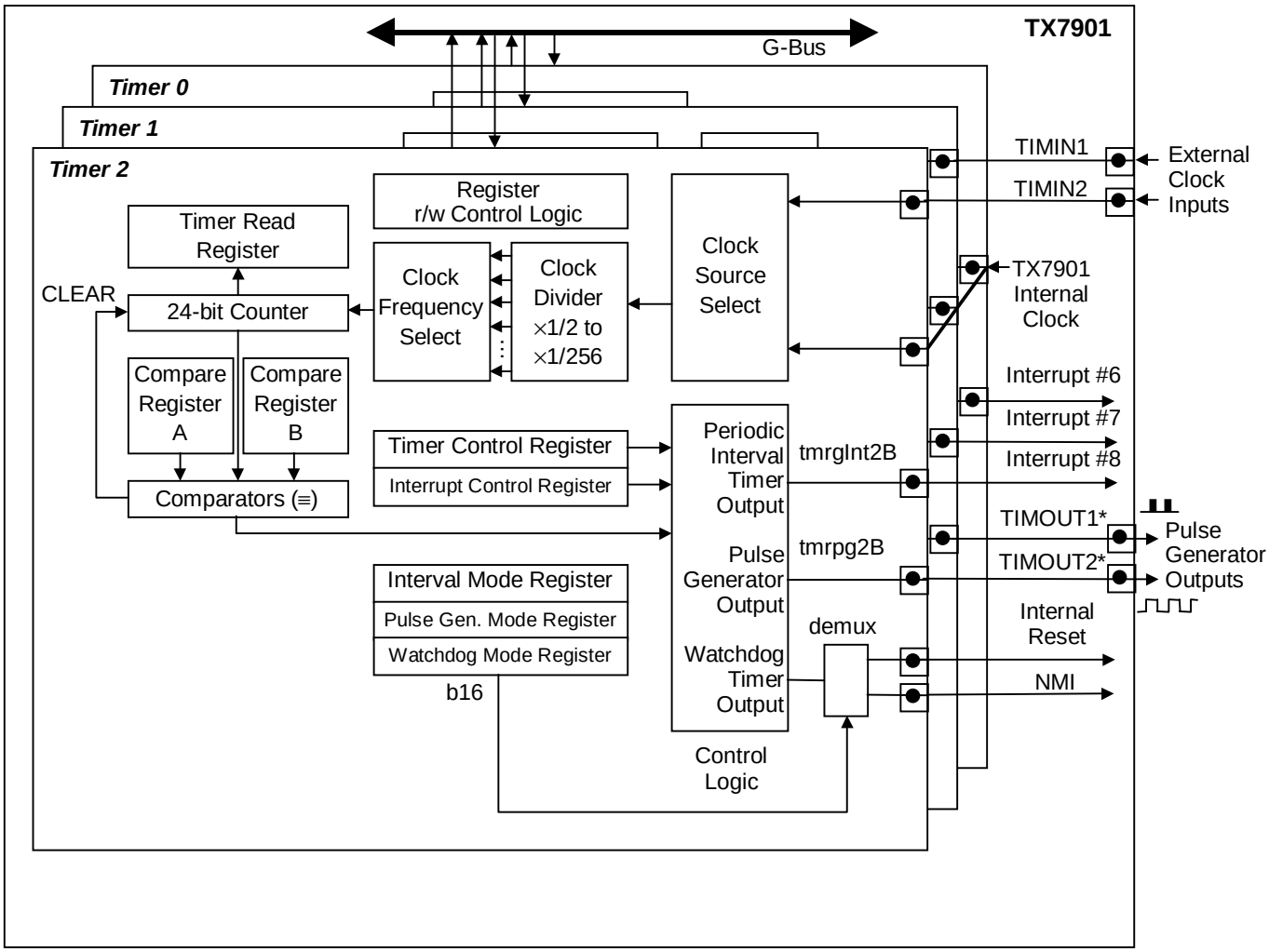


Figure 10-2 Timer 0, Timer 1, and Timer 2 Connections

10.3 Signals

Table 10-2 lists the signals that implement the interface between the Timer/Counter and the C790. Direct inputs and outputs to the outside of the TX7901 are in the upper case.

Table 10-2 TX7901 Programmable Timer/Counter Signals

Signal Name	I/O	Description
gbsBusClk	I	Internal Clock from G-Bus, currently 66/50 MHz.
gbsgData[64:0]	I	64-bit Data Bus input from the G-Bus to the Timers. Only the lower 24 bits [23:0] are used by the timer for writing to its internal registers.
gbsgAddr[31:2]	I	32-bit Address Bus input from the G-Bus to the Timers. Is used internally for correctly addressing the device registers inside each timer.
gcbgTMRRegCSB	I	Global Chip Select input signal from the G-Bus to the Timer. The G-Bus generates to the timers a single chip select, which is decoded with the help of address information that is used for selecting each timer. Active Low
gbsgRdB	I	Read input signal from the G-Bus to the Timer. This Active Low signal indicates when a master wants to read from the Timer's internal device registers. This signal remains asserted during the entire read cycle.
gbsgWrB	I	Write input signal from the G-Bus to the Timer. This Active Low signal indicates when a master wants to write to the Timer's internal device registers. This signal remains asserted during the entire write cycle.
gbsgBStartB	I	Input Start signal from the G-Bus to the Timer. This Active Low signal indicates that the G-Bus is starting either a read or a write cycle. This signal is latched internally by the timer unit for one cycle, since the gcbgTMRRegCSB signal arrives one clock cycle later and both signals are needed together to start a read or write cycle on any of the three timers.
gbsBEB[7:0]	I	Byte Enable bus inputs from the G-Bus to the Timer. A value of 0xF0 on this bus indicates that the lower 32 bits [31:0] on the Data Bus are valid and that the upper 32 bits [63:32] are invalid. The Timer only writes to its internal registers if this value is 0xF0. Otherwise, it generates a dummy Acknowledge signal to terminate the cycle. This was implemented to accommodate the G-Bus specifications.
sysResetB	I	System Reset input signal from the TX7901 to the Timer. This Active Low signal, when asserted, resets all three timer units simultaneously.
tgdisB	I	Disable Signal input from the TX7901 to the Timer. This Active Low signal, when asserted, allows the timers to be tested in the stand-by and multiplex modes. This signal is currently hardwired to always be deasserted.
ttest	I	Test Signal input from the TX7901 to the Timer. This Active High signal, when asserted, allows a preset count data value to be written to the TMTRR register. Currently, this signal is hardwired to always be deasserted.
TIMIN1	I	External Clock input for Timer 1 from a source external to the TX7901. This clock may be used instead of the internal clock for counting or for pulse generation, and may be divided internally before use.
TIMIN2	I	External Clock input for Timer 2 from a source external to the TX7901. This clock may be used instead of the internal clock for counting or for pulse generation, and may be divided internally before use.
tscanB	I	Scan Test input from the TX7901 to the Timer. When asserted, this Active Low signal causes the timers to be in the scan test mode. When the scan test mode is not used, this signal must be deasserted. This signal is currently hardwired to always be deasserted.
tmrgAck32B	O	Active Low acknowledge signal from the Timer. This signal is connected to the gout04Ack32* signal of the G-Bus to indicate that a read or write operation from/to the timer is complete. When the G-Bus master initiates a read or a write cycle to the timer, the timer responds by asserting this signal to indicate that the operation is complete. In the event that the master sends a read or write request with Byte Enables other than 0xF0, the timer will assert this signal (dummy acknowledgement) to terminate the cycle, and will not read or write from/to its registers.
tmrgInt0B	O	Active Low output for Timer 0 when it is configured as a Periodic Interval Timer. Timer 0 is configured as a Periodic Interval Timer by writing to its configuration register. This

Signal Name	I/O	Description
		output is connected to Interrupt 6 of the Interrupt Controller. This output will be asserted when the Max Count is reached, and will remain asserted until the proper registers are written to (to de-assert it), or until the C790 is Reset.
tmrgInt1B	O	Active Low output for Timer 1 when it is configured as a Periodic Interval Timer. Timer 1 is configured as a Periodic Interval Timer by writing to its configuration register. This output is connected to Interrupt 7 of the Interrupt Controller. This output will be asserted when the Max Count is reached, and will remain asserted until the proper registers are written to (to de-assert it), or until the C790 is Reset.
tmrgInt2B	O	Active Low output for Timer 2 when it is configured as a Periodic Interval Timer. Timer 2 is configured as a Periodic Interval Timer by writing to its configuration register. This output is connected to Interrupt 8 of the Interrupt Controller. This output will be asserted when the Max Count is reached, and will remain asserted until the proper registers are written to (to de-assert it), or until the C790 is Reset.
TIMOUT1*	O	Active Low output of Timer 1 when it is configured as a Pulse Generator. This output is exclusively used by devices outside of the TX7901. It is connected to an I/O pad output buffer for this purpose. Both the duty cycle and the frequency of the pulse generated on this output are programmable by writing appropriate values to the respective timer's configuration registers.
TIMOUT2*	O	Active low output of Timer 2 when it is configured as a Pulse Generator. This output is exclusively used by devices outside of the TX7901. It is connected to an I/O pad output buffer for this purpose. Both the duty cycle and the frequency of the pulse train generated from this output are programmable by writing appropriate values to the respective timer's configuration registers.
tmrwdtreq2B	O	Active Low output of Timer 2 when configured as a Watchdog Timer. This output is demultiplexed onto either the NMI input or the Master Reset input of the TX7901, and the demultiplexer is controlled by bit 16 of the Watchdog Mode Register.
tmrgdata[63:0]	O	Data bus output from the Timer to the G-Bus. The Timer drives this bus to 0xFFFF_FFFF when there are no data to output to the G-Bus. In other cases, the Timer only drives bits [31:0] of this bus.

10.4 Configuration Registers

Table 10-3 lists the addresses of the registers that configure the operation of the three TX7901 Timer/Counters when written to. Detailed descriptions of the fields of each of these registers appear in the following sub-sections. Register functionality and operation is discussed in Section 10.5 in addition to discussing Timer/Counter Operation.

Table 10-3 Timer/Counter Configuration Registers

Register Name	R/W	Timer 0		Timer 1		Timer 2	
		Address	Field Name	Address	Field Name	Address	Field Name
Timer Control Register	R/W	0x1E00_4000	TMTCR0	0x1E00_4100	TMTCR1	0x1E00_4200	TMTCR2
Timer Interrupt Status Register	R/W	0x1E00_4004	TMTISR0	0x1E00_4104	TMTISR1	0x1E00_4204	TMTISR2
Compare Register A	R/W	0x1E00_4008	TMCPRA0	0x1E00_4108	TMCPRA1	0x1E00_4208	TMCPRA2
Compare Register B	R/W	0x1E00_400C	TMCPRB0	0x1E00_410C	TMCPRB1	0x1E00_420C	TMCPRB2
Interval Timer Mode Register	R/W	0x1E00_4010	TMITMR0	0x1E00_4110	TMITMR1	0x1E00_4210	TMITMR2
Divider Register	R/W	0x1E00_4020	TMCCDR0	0x1E00_4120	TMCCDR1	0x1E00_4220	TMCCDR2
Pulse Generator Mode Register	R/W	Reserved	Reserved	0x1E00_4130	TMPGMR1	0x1E00_4230	TMPGMR2
Watch Dog Timer Mode Register	R/W	Reserved	Reserved	Reserved	Reserved	0x1E00_4240	TMWTMR2
Timer Read Register	R/O	0x1E00_40F0	TMTRR0	0x1E00_41F0	TMTRR1	0x1E00_42F0	TMTRR2

Note: Accessing registers that are marked "Reserved" will result in unknown behavior. All registers must be accessed as 32-bit quantities.

10.4.1 Timer Control Registers TMTCR0, TMTCR1, TMTCR2

The following figure and Table 10-4 detail the fields of Timer Control Registers TMTCR0, TMTCR1, and TMTCR2.

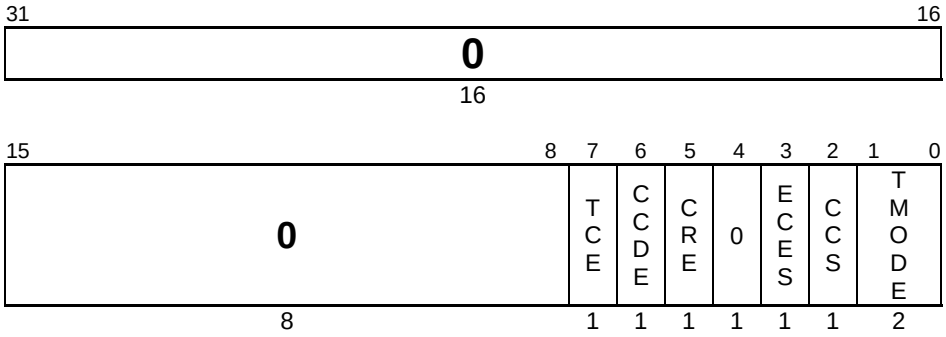


Table 10-4 Field Descriptions for Timer Control Registers TMTCRx

Bit(s)	Field	Field Name	R/W	Description
31:8	–	–	R/O	Reserved
7	TCE	Timer Counter Enable	R/W	0: When CRE = 0, Counter is only stopped. When CRE = 1, Counter is stopped and reset. (0) 1: Counter is enabled (i.e. in operation).
6	CCDE	Counter Clock Divide Enable	R/W	0: Disables frequency division of the internal clock. (0) 1: Enables the frequency division of the internal clock.
5	CRE	Counter "Reset-When-Stopped" Enable	R/W	When the Counter is stopped (by clearing the TCE bit to "0") this bit determines whether the Counter should also be reset at that time. (0) 0: Holds the count value (i.e. does not clear it) when the counter is stopped by the TCE bit. 1: Clears the count value when the counter is stopped by the TCE bit.
4	–	–	R/O	Reserved
3	ECES	External Clock (Trigger-) Edge Select	R/W	Selects the clock edge used for externally triggering counter. (0) 0: Falling edge of external clock input triggers counter 1: Rising edge of external clock triggers counter
2	CCS	Counter Clock (Source) Select	R/W	0: Select internal system clock as clock source gbusClk = 1/2 RefClk (0) 1: Select external clock input as clock source
1:0	TMODE	Timer Mode	R/W	Programs the Timer/Counter's Mode of Operation. (00) 00: Interval Timer Mode 01: Pulse Generator Mode 10: Watchdog Timer Mode 11: Unused (reserved)

Note: The register fields marked "Reserved" are read back as zeroes, and are ignored when written to.

10.4.2 Interval Timer Mode Registers TMITMR0, TMITMR1, TMITMR2

The following figure and Table 10-5 detail the fields for the Interval Timer Registers TMITMR0, TMITMR1, and TMITMR2.

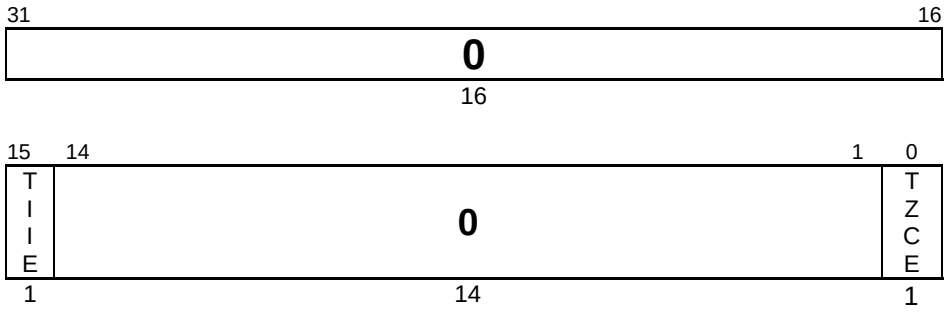


Table 10-5 Fields Descriptions of Interval Timer Mode Registers TMITMRx

Bit(s)	Field	Field Name	R/W	Description
31:16	–	–	R/O	Reserved
15	TIIE	Timer's Interval Timer Interrupt Enable	R/W	Sets up interrupt enable/disable in the interval timer mode. (0) 0: Disable (mask) 1: Enable
14:1	–	–	R/O	Reserved
0	TZCE	Interval Timer Zero Clear Enable	R/W	Determines whether to clear the counter to "0" after the count value matches compare register A. If not cleared, the counter is halted at that value. If the timer interrupt is terminated when TZCE = 0 while the counter is halted, the interrupt will not recur. (0) 0: Disable 1: Enable

Note: The register fields marked "Reserved" are read back as zeroes, and are ignored when written to.

10.4.3 Divider Registers TMCCDR0, TMCCDR1, TMCCDR2

The following figure and Table 10-6 detail the fields of the Divider Registers, TMCCDR0, TMCCDR1 and TMCCDR2.

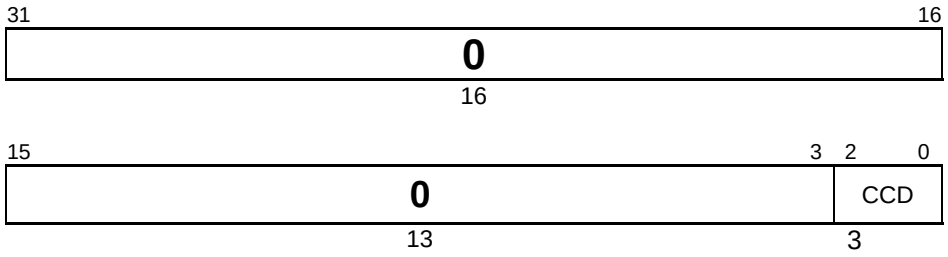


Table 10-6 Field Descriptions for Divider Registers TMCCDR0, TMCCDR1, TMCCDR2

Bit(s)	Field	Field Name	R/W	Description
31:3	–	–	R/O	Reserved
2:0	CCD	Counter Clock Divide value	R/W	Specifies the divisor value of the selected clock. The value used is 2 raised to the power (***n***+1) when a binary value ***n*** is used in this field. (000) 000:2 001:4 010:8 011:16 100:32 101:64 110:128 111:256

Note: The register fields marked “Reserved” are read back as zeroes, and are ignored when written to.

10.4.4 Pulse Generator Mode Registers TMPGMR1, TMPGMR2

The following figure and Table 10-7 detail the fields of the Pulse Generator Mode Registers for Timers 1 and 2, TMPGMR1 and TMPGMR2.

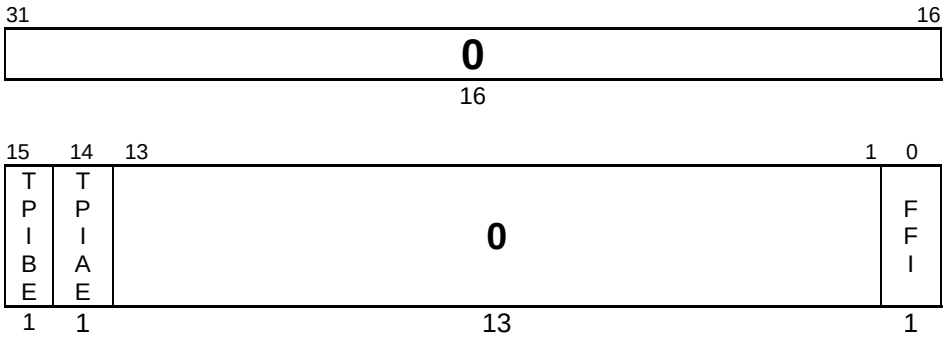


Table 10-7 Field Descriptions for Pulse Generator Mode Registers TMPGMRx

Bit	Field	Field Name	R/W	Description
31:16	–	–	R/O	Reserved
15	TPIBE	TMCPRB Interrupt Enable	R/W	Timer/Pulse Generator Interrupt by TMCPRB Enable; sets up enable/ disable for the timer interrupt that occurs in the Pulse Generator Mode when the TMCPRB and the counter value match. (0) 0: Disable (mask) 1: Enable
14	TPIAE	TMCPRA Interrupt Enable	R/W	Timer/Pulse Generator Interrupt by TMCPRA Enable; sets up enable/ disable for the timer interrupt that occurs in the Pulse Generator Mode when the TMCPRA and the counter value match. (0) 0: Disable (mask) 1: Enable
13:1	–	–	R/O	Reserved
0	FFI	Flip-flop initial value	R/W	Specifies the initial value of the output flip-flop. (0) 0: Low 1: High

Note: The register fields marked “Reserved” are read back as zeroes, and are ignored when written to.

10.4.5 Watchdog Timer Mode Register (TMWTMR) Fields

The following figure and Table 10-8 detail the fields of the Watchdog Timer Mode Register, TMWTMR.

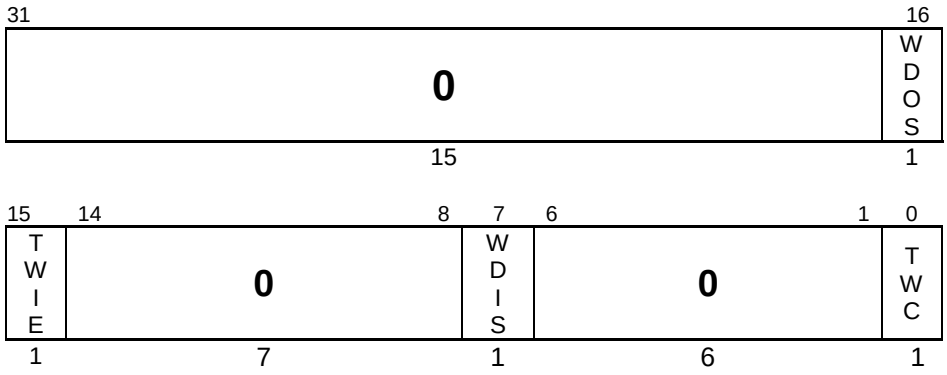


Table 10-8 Watchdog Timer Mode Register (TMWTMR) Field Descriptions

Bit(s)	Field	Field Name	R/W	Description
31:17	–	–	R/O	Reserved
16	WDOS	Watchdog Output Select	R/W	Selects the type of exception to issue when the Watchdog Timer expires. (0) 0 : Causes an NMI. 1 : Causes a Master Reset (The default value is "0.")
15	TWIE	Watchdog Timer Interrupt Enable	R/W	Sets interrupt enable/disable in the Watchdog Timer Mode. (0) 0: Disable (mask) 1: Enable
14:8	–	–	R/O	Reserved
7	WDIS	Watchdog Timer Disable	R/W	Disables the Watchdog Timer Mode when WDIS is set to “1” and then sets the TCE bit of the timer control register to “0”. (0) Note: The WDIS bit will automatically be cleared to “0” after the watchdog timer is disabled. Writing 0 to the WDIS bit has no effect
6:1	–	–	R/W	Reserved
0	TWC	Watchdog Timer Clear	R/W	1: Clears the Watchdog Timer’s counter. (0) Note: The TWC bit will be automatically cleared to “0” after the counter is cleared. Writing 0 to the TWC bit has no effect.

Note: The register fields marked “Reserved” are read back as zeroes, and are ignored when written to.

10.4.6 Timer Interrupt Status Registers TMTISR0, TMTISR1, TMTISR2

The following figure and Table 10-9 detail the fields of the Timer Interrupt Status Registers, TMTISR0, TMTISR1, and TMTISR2. Note that Timers 0 and 1 use only some of these fields.

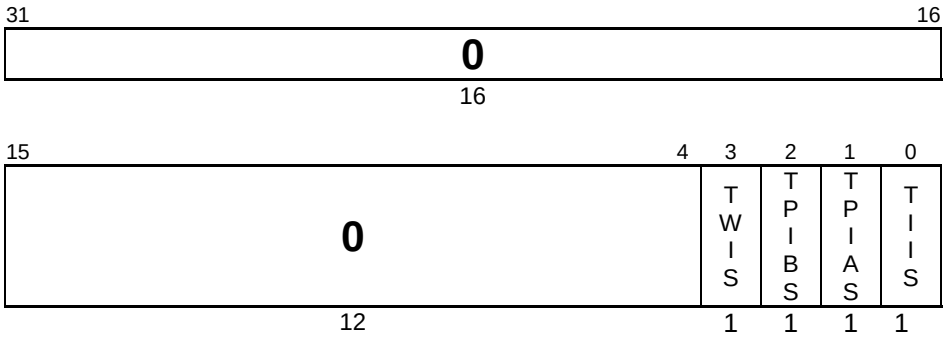


Table 10-9 Field Descriptions for Timer Interrupt Status Registers TMTISRx

Bit	Field	Field Name	Timer(s)	R/W	Description
31:4	–	–	–	R/O	Reserved
3	TWIS	Watchdog Timer Interrupt Status	2	R/W	When the watchdog timer interrupt is enabled by setting the TWIE bit and the counter value matches the compare register TMCPRA value during counting, the TWIS bit is set, asserting the WDTINTREQ* line Low. (0) By clearing the TWIS bit, the WDTINTREQ* line is de-asserted High, clearing the interrupt request. "0" read: No interrupt request is being generated. "1" read: Counter has matched the compare register and an interrupt request is being generated. "0" written: Resets the timer interrupt request. Note: Writing a "1" to the TWIS bit has no effect functionally.
2	TPIBS	Timer/Pulse Generator B (TMCPRB) interrupt status	1, 2	R/W	When the Timer/Pulse Generator interrupt is enabled by setting the TPIBE bit and the counter value matches the compare register TMCPRB value during counting, the TPIBS bit is set, asserting the TMINTREQ* line Low. By clearing the TPIBS bit, the TMINTREQ* line is de-asserted High, clearing the interrupt request. (0) "0" read: No interrupt request is being generated. "1" read: Counter has matched compare register B, and an interrupt request is being generated. "0" written: Resets the pulse generator interrupt request for TMCPRB. Note: Writing a "1" to the TPIBS bit has no effect functionally.
1	TPIAS	Timer/Pulse Generator A (TMCPRA) interrupt status	1, 2	R/W	When the Timer/Pulse Generator Interrupt is enabled by setting the TPIAE bit and the counter value matches the compare register TMCPRA value during counting, the TPIAS bit is set, asserting the TMINTREQ* line Low. By clearing the TPIAS bit, the TMINTREQ* line is de-asserted High, resetting the interrupt request. (0) "0" read: No interrupt request is being generated. "1" read: Counter has matched compare register A, and an interrupt request is being generated. "0" written: Resets the pulse generator interrupt request for TMCPRA. Note: Writing a "1" to the TPIAS bit has no effect functionally.

Bit	Field	Field Name	Timer(s)	R/W	Description
0	TIIS	Timer's Interval Timer Interrupt Status bit	0, 1, 2	R/W	When the Interval Timer Interrupt is enabled by setting the TIIE bit in the Interval Timer Mode Register (TMITMRx) and the counter value matches the compare register TMCPRA value during counting, the TIIS bit is set, asserting the TMINTREQ* line Low. By clearing the TIIS bit, the TMINTREQ* line is de-asserted High, resetting the interrupt request. (0) "0" read: No interrupt request is being generated. "1" read: Counter has matched compare register A, and an interrupt request is being generated for the Interval Timer. "0" written: Resets the Interval Timer interrupt request for TMCPRA. Note: Writing a "1" to the TIIS bit has no effect functionally.

10.4.7 Fields for Timer Compare Registers A (TMCPRAx) and B (TMCPRBx)

The following figure and Table 10-10 detail the fields of the Timer Compare Registers TMCPRAx and TMCPRBx.

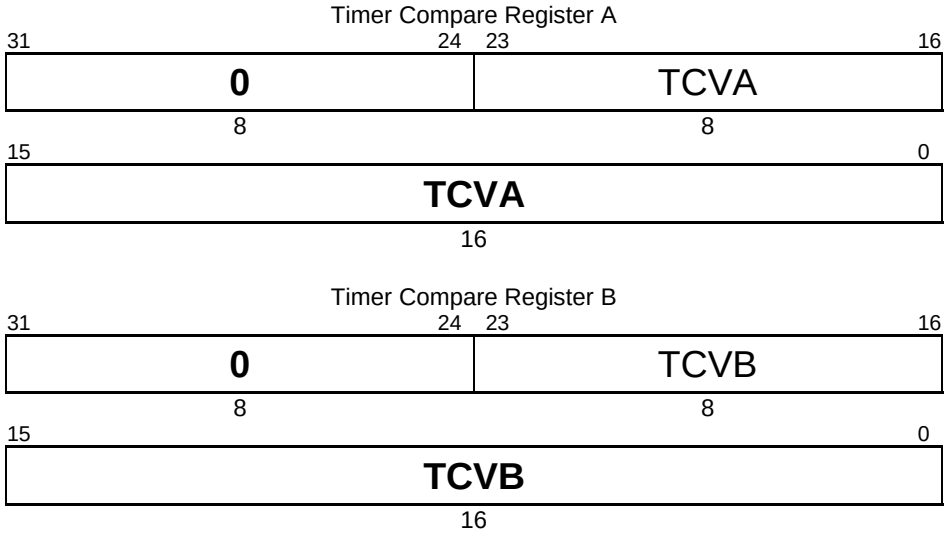


Table 10-10 Field Descriptions for Time Compare Registers TMCPRAx, TMCPRBx

Compare Register Field	Field Name	R/W	Description
TCVA [23:0]	Timer Compare Value A[23:0]	R/W	Contains the 24-bit compare value A for the timer. Used in all modes. (0xFFFFFF)
TCVB [23:0]	Timer Compare Value B[23:0]	R/W	Contains the 24-bit compare value B for the timer. Used in the Pulse Generator Mode. (0xFFFFFF)

Note: Compare Register B should be set up with a 24-bit value greater than that in Compare Register A.

10.4.8 Timer Read Registers (TMTRR0, TMTRR1, TMTRR2)

The following figure and Table 10-11 detail the fields of the Timer Read Registers TMTRR0, TMTRR1 and TMTRR2.

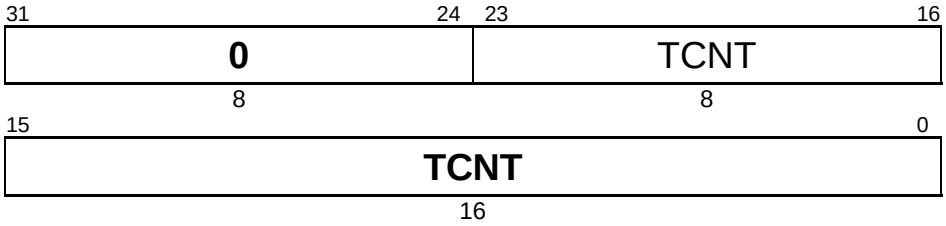


Table 10-11 Field Descriptions of Timer Read Registers TMTRRx

Bit(s)	Field	Field Name	R/W	Description
31:24	–	–	R/O	Reserved
23:0	TCNT	Timer Counter Value	R/O	The 24-bit value is copied to this register. The count value may be verified by reading this register. In the Preset Mode (TMTEST = 1) an arbitrary value may be written to this register. (0x000000)

Note: The register fields marked “Reserved” are read back as zeroes, and are ignored when written to.

10.5 Programmable Timer/Counter Operation

This section describes the operation of the Programmable Timers/Counters in each of the three modes in which they can operate. Note that not all of the operational modes are available in every timer/counter. (See Table 10-1.)

10.5.1 Interval Timer Mode Operation

This mode is set up by setting the timer mode (TMODE) field of the timer control register (TMTCR) to 0b00 (binary). The Counter Clock Select (CCS) bit of the TMTCR register determines whether the internal system clock or the external clock input is selected.

Dividing down the selected (internal or external) clock by the value set up for the divisor generates the actual clock frequency. This value is set up in the Counter Clock Divide (CCD) field of the divider register (TMCCDR) and is activated by setting the Counter Clock Divide Enable (CCDE) bit of the TMTCR register to “1”. The clock can be divided down by powers of 2, ranging from 2 to 256, depending on the value in the 3-bit CCD field. (If the external clock input is to be used, the clock edge at which the count happens is selectable by setting up the External Clock Edge Select (ECES) field of the TMTCR register.)

When the Timer Count Enable (TCE) field of the timer control register TMTCR is set to “1”, the 24-bit counter begins counting. When the count value reaches (i.e., matches) the value of compare register A (TMCPRA), the Timer Interval Interrupt Status (TIIS) bit of the Timer Interrupt status register (TMTISR) is set to “1”. At this point the interrupt control logic passes the inverted TIIS bit as the timer interrupt request signal TMINTREQ*, driving it Low (since TIIS* = 0), if the Timer Interval Interrupt Enable (TIIE) bit of the Interval Timer Mode Register

(TMITMR) is set to “1”. When the TIIE bit is subsequently cleared to “0”, this causes the interrupt logic to mask the inverted TIIS bit (and to not transmit it), effectively disabling the interrupt request signal TMINTREQ*.

When the Timer Interval Interrupt status (TIIS) bit of the status register (TMTISR) is cleared to “0” (by writing a “0” to it while servicing the interrupt), the timer interrupt request signal TMINTREQ* is de-asserted High, effectively terminating the interrupt. Writing “1” to the TIIS bit is ignored.

When the Timer Zero Clear Enable (TZCE) bit of the TMITMR Register is set to “1”, the 24-bit counter is cleared to “0” and counting restarts, asserting the interrupt request signal when the count value reaches the value in TMCPRA.

When the TZCE bit is “0”, counter operation is stopped when the count matches the value of TMCPRA. This is summarized in the table below:

Table 10-12 Interrupt control with the TIIE and TZCE bits

TIIE	TZCE	Action or inaction when counter has reached the set up value
0	X	Timer Interval Interrupt Requests are disabled (masked out), and do not occur.
1	0	An Interrupt Request occurs. If the Timer Zero Clear Enable bit is cleared to “0” when an interrupt is terminated (by clearing the TIIS bit), no further interrupts occur. When the interrupt is terminated after TZCE is set to “1”, a timer interrupt recurs. This is the same as when TIIE = 1 and TZCE = 1.
1	1	Timer Interval Interrupt Requests are enabled, and do occur periodically.

A copy of the count value in the 24-bit counter can be read from the Timer Read Register (TMTRR).

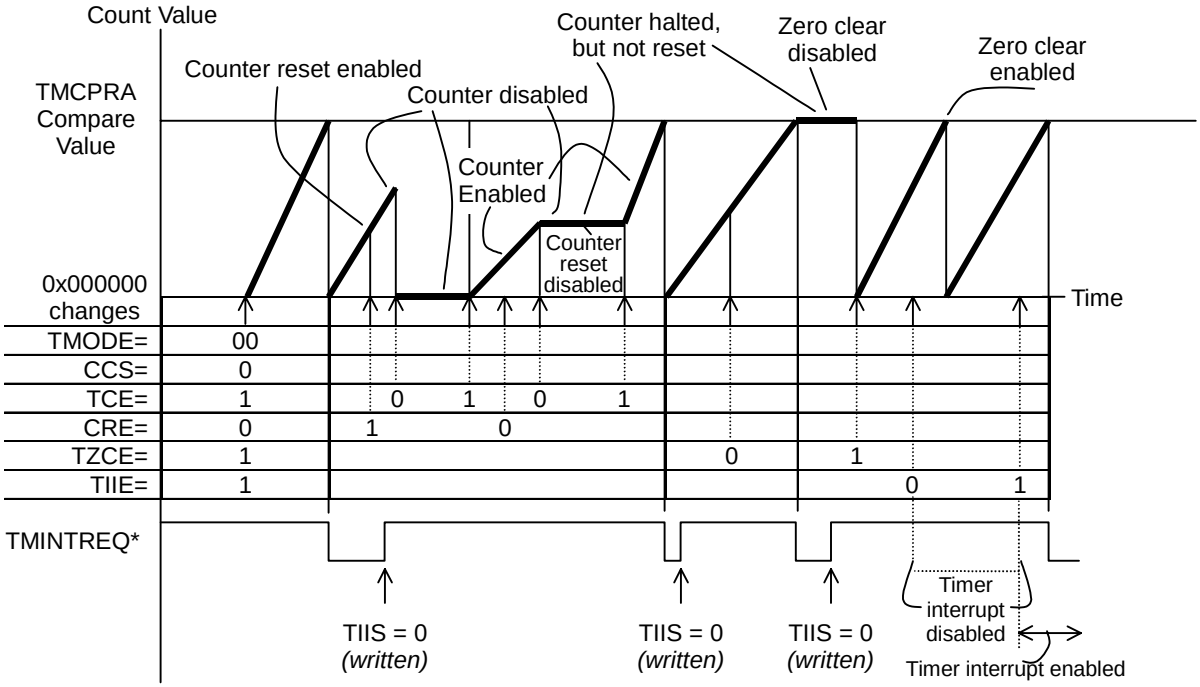


Figure 10-3 Interval Timer Operation using Internal Clock

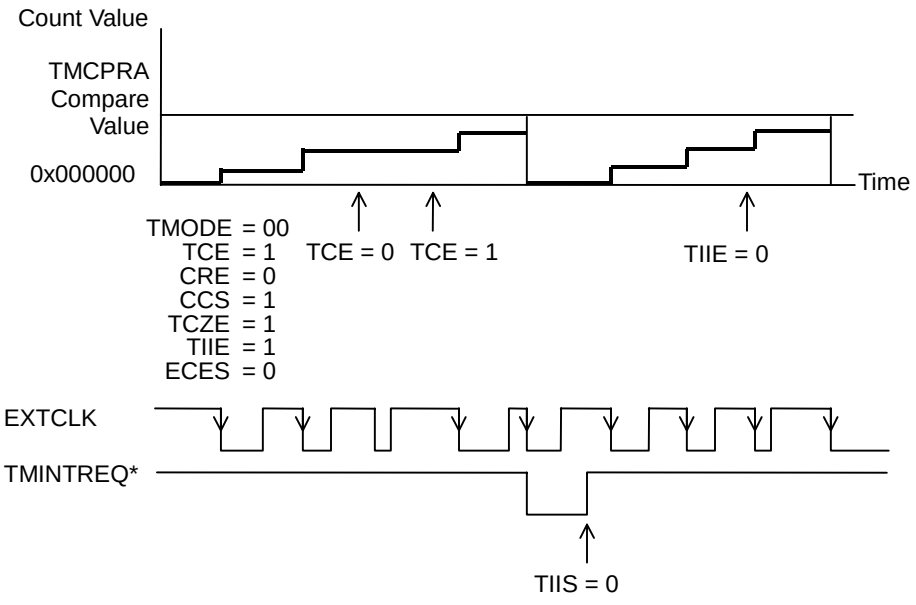


Figure 10-4 Interval Timer Operation using External Clock

10.5.1.1 Divisors For Counting

Table 10-13 shows the counter frequencies that result from selecting and dividing the internal 66/50 MHz clock by the decimal Divider values given (using these values in the Divider Register). These frequencies are used in all three counter modes.

Table 10-13 Divider values and Counter Frequencies Generated

Divider		gbusClk = 66.7 MHz				
Decimal	CCDR <CCD>	Freq. Unit	Resolution Unit	24bit Max Time(s)	Generate 60Hz interval	Counter value for 1 second
2	0b000	33.33 MHz	30 ns	0.50	555,556	33,333,333
4	0b001	16.67 MHz	60 ns	1.01	277,778	16,666,667
8	0b010	8.33 MHz	120 ns	2.01	138,889	8,333,333
16	0b011	4.17 MHz	240 ns	4.03	69,444	4,166,667
32	0b100	2.08 MHz	480 ns	8.05	34,722	2,083,333
64	0b101	1.04 MHz	960 ns	16.11	17,361	1,041,667
128	0b110	520.83 kHz	1.92 us	32.21	8,681	520,833
256	0b111	260.42 kHz	3.84 us	64.42	4,340	260,417

Divider		gbusClk = 50.0 MHz				
Decimal	CCDR <CCD>	Freq. Unit	Resolution Unit	24bit Max Time(s)	Generate 60Hz interval	Counter value for 1 second
2	0b000	25.00 MHz	40 ns	0.67	416,667	25,000,000
4	0b001	12.50 MHz	80 ns	1.34	208,333	12,500,000
8	0b010	6.25 MHz	160 ns	2.68	104,167	6,250,000
16	0b011	3.13 MHz	320 ns	5.37	52,083	3,125,000
32	0b100	1.56 MHz	640 ns	10.74	26,042	1,562,500
64	0b101	781.25 kHz	1.28 us	21.47	13,021	781,250
128	0b110	390.63 kHz	2.56 us	42.95	6,510	390,625
256	0b111	195.31 kHz	5.12 us	85.90	3,255	195,313

10.5.2 Pulse Generator Mode Operation

This mode is set up by setting the timer mode (TMODE) field of the Timer Control Register (TMTCR) to 0b01 (binary). In this mode, rectangular pulses of specific frequency and duty-cycle can be generated with the help of the two compare registers TMCPRA and TMCPRB. The contents of TMCPRA and TMCPRB should be set up with the value in TMCPRA being less than the value in TMCPRB.

When the Timer Counter Enable (TCE) bit of the Timer Control Register is set to “1”, the 24-bit counter begins counting. When the counter reaches the value set in compare register TMCPRA, the Timer Flip-Flop toggles, inverting its output, which is made available on the TMFFOUT output of the timer/counter. The counter continues counting until the counter reaches the (larger) value set in compare register TMCPRB. The Timer Flip-Flop toggles again, and the counter is cleared. Setting the required state in the Flip-Flop Initialize (FFI) field of the Pulse Generator Mode Register (TMPGMR) configures the initial state of the Timer Flip-Flop.

When the count value reaches the TMCPRA compare register value, the Pulse Generator Interrupt Status bit for Compare Register A (TPIAS) in the Timer Interrupt Status Register TMTISR is set to “1”. The interrupt control logic asserts TMINTREQ* when the TPIAS bit is set and also enabled (by having the TPIAE bit of the TMPGMR register set as well). If the TPIAE bit is cleared, it masks the TPIAS bit from being forwarded as an interrupt request and masks assertion of TMINTREQ* Low. Similarly, the interrupt control logic asserts TMINTREQ* when the TPIBS bit is set as well as enabled (by having the TPIBE bit of the TMPGMR register set as well). If the TPIBE bit is cleared, it masks the TPIBS bit from being forwarded as an interrupt request and masks assertion of TMINTREQ* Low. These interrupt requests may be respectively reset (cleared) by clearing the TPIAS or TPIBS bits as appropriate, causing TMINTREQ* to be de-asserted as well.

The counter can be clocked by either the internal system clock or the external clock. The CCS bit of the Timer Counter Register is used to select the clock that controls the timer.

The selected clock is frequency divided before being used to clock the counters. When the CCDE bit of the TMTCR register is a “1”, the three CCD bits of the divider register scale the clock down to between 1/2 and 1/256 of the selected clock frequency. When using the internal clock, the counter is triggered by the rising edge of the clock. When using the external clock, it may be configured to trigger on either edge by writing a “0” or a “1” for bit ECES of the Timer Counter Register.

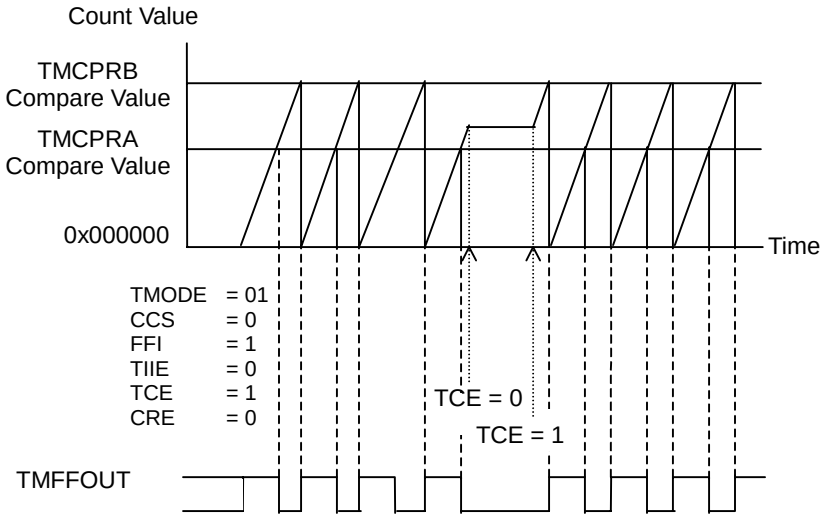


Figure 10-5 Pulse Generator Mode Operation

10.5.3 Watchdog Timer Mode Operation

This mode is set up by setting the timer mode (TMODE) field of the Timer Control Register (TMTCR) to 0b10 (binary). In this mode, when the Timer/Counter Enable (TCE) bit of the TMTCR register is set to “1”, the 24-bit counter begins counting. When the count value reaches the value in register TMCPRA, the comparator sets the Watchdog Timer Interrupt Status bit (TWIS) of the Timer Interrupt Status Register (TMTISR) to “1”. This also causes the interrupt control logic to assert the Watchdog Timer Interrupt Request signal WDTINTREQ*, if the Watchdog Timer Interrupt Enable bit (TWIE) in the Watchdog Timer Mode Register (TMWTMR) is set to “1”. If the TWIE bit is “0”, WDTINTREQ* is masked, and is thus not asserted. Writing a “1” to the TWIS bit is ignored.

To clear the 24-bit counter, the Watchdog Timer Clear (TWC) bit of the Watchdog Timer Mode Register (TMWTMR) is set to “1”. The TWC bit is automatically reset to “0” after the 24-bit counter is cleared.

When the Timer/Counter Enable bit (TCE) is cleared and the Watchdog Timer Disable (WDIS) bit of the TMWTMR register is set to “1”, the Watchdog Timer Counter is disabled. However, when the WDIS bit is “0”, count operation cannot be stopped by clearing the TCE bit to “0”. In addition, the TWIE bit cannot be cleared while the WDIS bit is “0”. Another way to disable the watchdog timer (by masking the interrupt) is to clear the Watchdog Timer Enable bit (TWIE) while WDIS bit is set to “1”. The WDIS bit is automatically cleared when the Watchdog Timer becomes disabled by one of the methods described above.

The count value for the Watchdog Timer can be read from the TMTRR register.

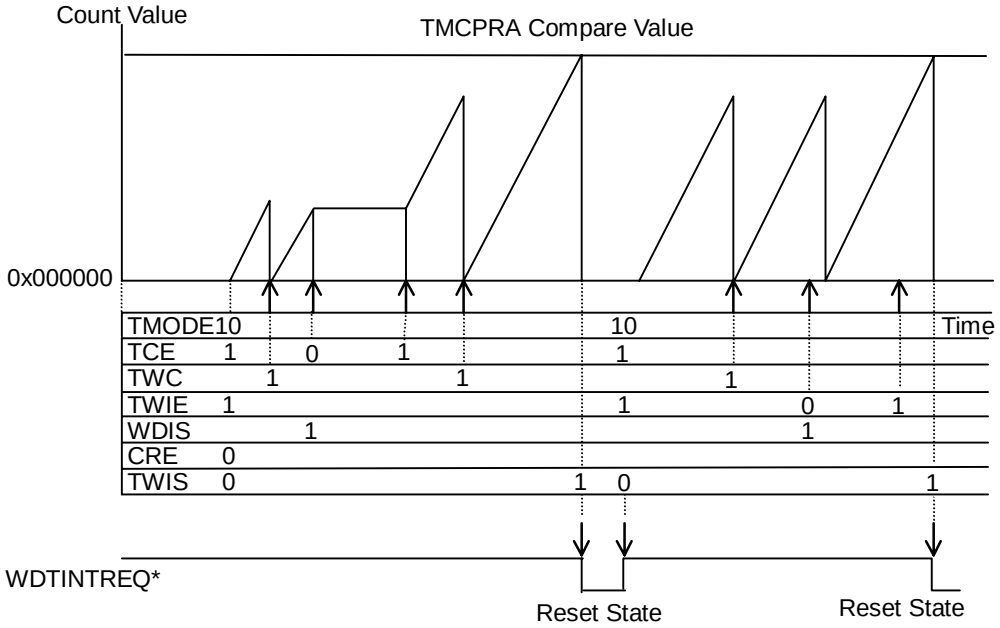


Figure 10-6 Watchdog Timer Mode Operation of Timer/Counter

10.5.4 Examples of Timer/Counter Timing

These examples illustrate the different modes of operation of the various timers. (Refer to Section 10.1 at the beginning of the chapter for details.)

10.5.4.1 Interval Timer Mode Interrupt Timing

Figure 10-7 illustrates a case in which Timer Compare Value A (TCVA) is “3”, and the timer input clock is the 50 MHz internal G-Bus Clock. The Timer Interval Interrupt Status Bit (TIIS) of the Interrupt Status Register TMTISR is set at the rising edge of the G-Bus Clock Input Clock gbsBusClk when a match has been detected. The interrupt signal TMINTREQ* is asserted synchronously with the internal system clock and the counter is simultaneously cleared to “0” (assuming in this example that bit TZCE of the Interval Timer Mode Register TMITMR, is set).

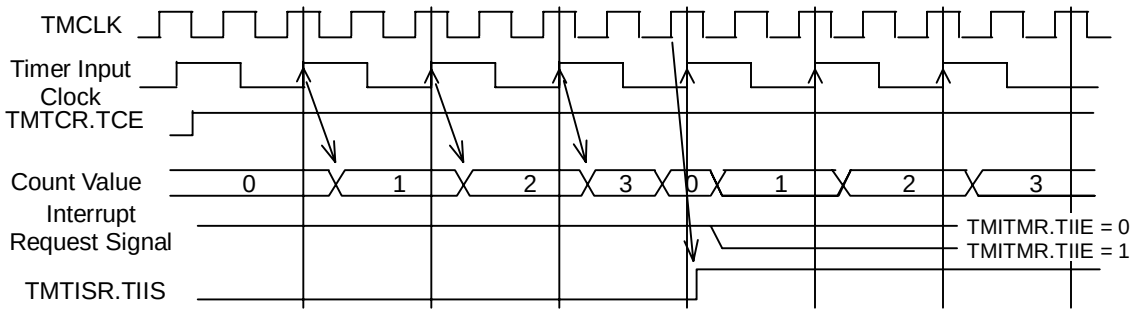


Figure 10-7 Interval Timing Example Using Internal Clock

The above figure illustrates a case in which Timer Compare Value A (TCVA) is “3”, and the counter is triggered by the external clock input. The Timer’s Interval Timer Interrupt Status bit TIIS of the Interrupt Status Register TMTISR is set at the rising edge of the External Input Clock when a match has been detected and TMINTREQ* is asserted Low. Simultaneously, the counter is cleared (assuming in this example that bit TZCE of the Interval Timer Mode Register TMITMR is set).

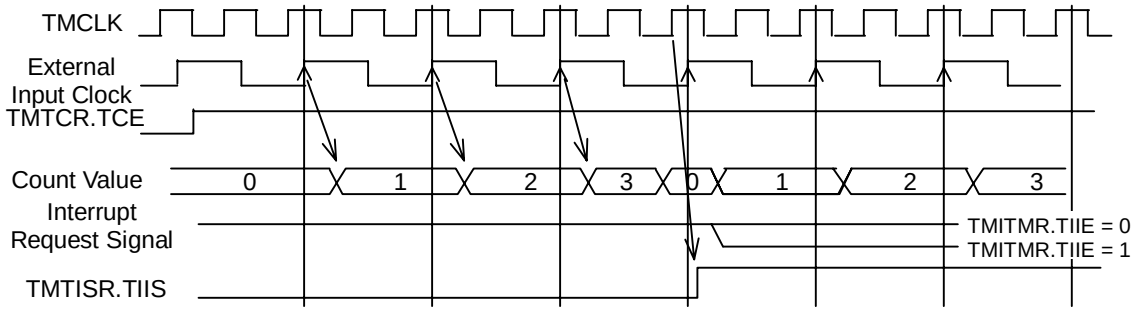


Figure 10-8 Interval Timing Example Using External Input Clock

Pulse Generator Mode Flip-Flop Output Timing

Figure 10-9 illustrates a case in which Timer Compare Value A (TCVA) is “1” and Timer Compare Value B (TCVB) is “3”, in the Pulse Generator Timing Mode. The Initial value for the Timer Flip-flop is 0, and the Timer Flip-Flop is initialized simultaneously with TMPGMR being written to.

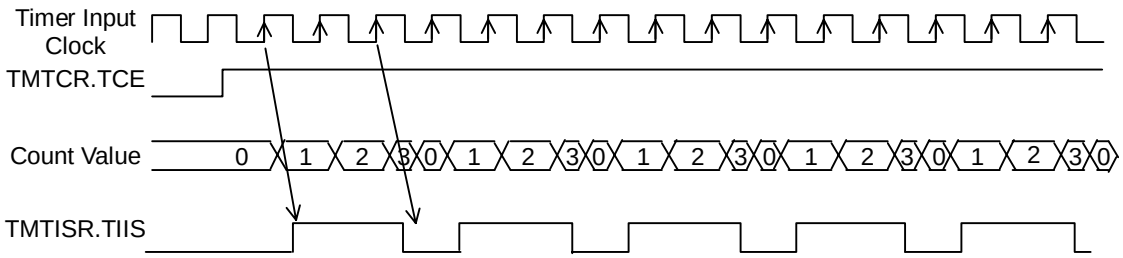


Figure 10-9 Pulse Generator Mode Timing Example

10.5.4.2 Watchdog Timer Mode Interrupt Timing

Figure 10-10 illustrates the use of the Watchdog Timer. This mode is the same as the Internal Timer Mode except that the Watchdog Timer issues an NMI or Master Reset instead of an interrupt.

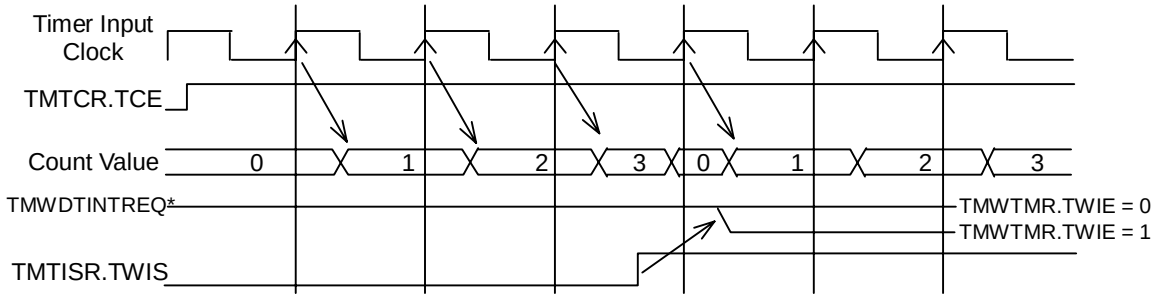


Figure 10-10 Watchdog Timer Timing Example

11. Interrupt Controller

11.1 Introduction

The Interrupt Controller arbitrates all the interrupt requests from internal and external devices and sends the interrupt request that was granted access to the processor. The interrupt controller features are:

- 32 internal and external interrupt requests
- Interrupt mask
- Level trigger only

11.2 Operation

The interrupt controller contains an interrupt status register to identify up to 16 different interrupt sources. There is a corresponding 16-bit interrupt mask register. When the corresponding mask bit is cleared, the interrupt request is disabled. Once an interrupt request is detected and its corresponding mask bit is set, an interrupt request is asserted to the C790.

Table 11-1 lists all the maskable interrupt sources.

Table 11-1 Maskable Interrupt Sources

Interrupt Number	Interrupt Source
0	G-Bridge
1	PCI-0
2	MAC0
3	MAC1
4	Reserved
5	DMAC
6	Timer0
7	Timer1
8	Timer2
9	UART0
10	UART1
11	SPI
12	PCI-1
13	External Interrupt Mac0
14	External Interrupt Mac1
15	External Interrupt 0

Interrupt Number	Interrupt Source
16	External Interrupt 1
17	External Interrupt 2
18	External Interrupt 3
19	External Interrupt 4
20	PCI0 Reset
21	PCI1 Reset
22:31	Reserved

Once the C790 detects the interrupt request, it reads from the interrupt status register and mask register and determines which interrupt it should service. Once an interrupt is asserted, it should not be deasserted until it gets serviced.

11.3 Registers

The two registers associated with the interrupt controller are the Interrupt Status Register and the Interrupt Mask Register.

Table 11-2 Interrupt Controller Registers

Address	Field	Register Name
1E00_20E0	IRSTAT	Interrupt Status Register
1E00_20E8	IRMASK	Interrupt Mask Register

11.3.1 Interrupt Status Register (IRSTAT)

The Interrupt Status Register is periodically updated with the state (asserted or unasserted) of each interrupt. The following figure and Table 11-3 detail the fields of the Interrupt Status Register.

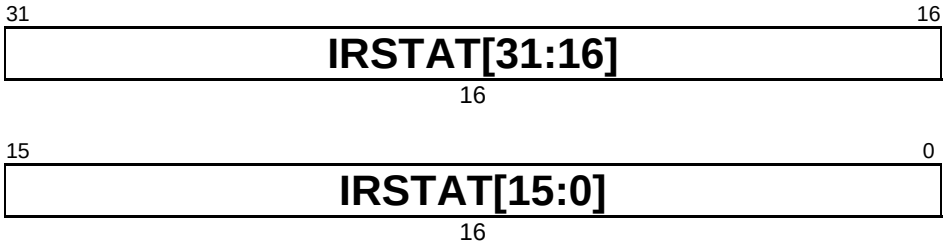


***Table 11-3* Interrupt Status Register Field Description**

Bits	Field	R/W	Description
31:0	IRSTAT	R/O	Interrupt Status 0: Interrupt request does not exist. 1: Interrupt request exists.

11.3.2 Interrupt Mask Register (IRMASK)

The Interrupt Mask Register enables interrupts to be selectively masked from causing an interrupt to the C790. The following figure and Table 11-4 detail the fields of the Interrupt Mask Register.

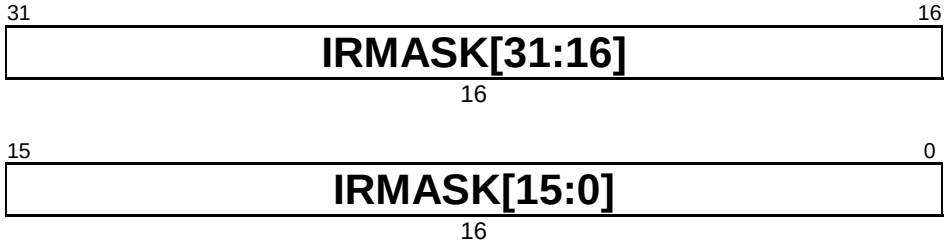


Table 11-4 Interrupt Mask Register Field Description

Bits	Field	R/W	Description
31:0	IRMASK	R/W	Interrupt Mask 0: Mask the interrupt request. 1: Enable the interrupt request.

12. 10/100 IEEE802.3 Media Access Controller

12.1 Overview

This chapter describes a 64-bit G-Bus Media Access Controller (MAC) for the TX7901. It operates at data transfer rates of 100 Mbps or 10 Mbps. In the half-duplex mode, the controller implements the IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. In the full-duplex mode, the controller implements the IEEE 802.3x MAC Control Layer and the PAUSE Operation for flow control. On the host side, it supports direct connection to the 64-bit G-Bus. The MAC can be either a G-Bus master or slave device. An internal DMA controller manages the transfer of data blocks between memory and the MAC's internal FIFOs, relieving the C790 of many tasks associated with network control.

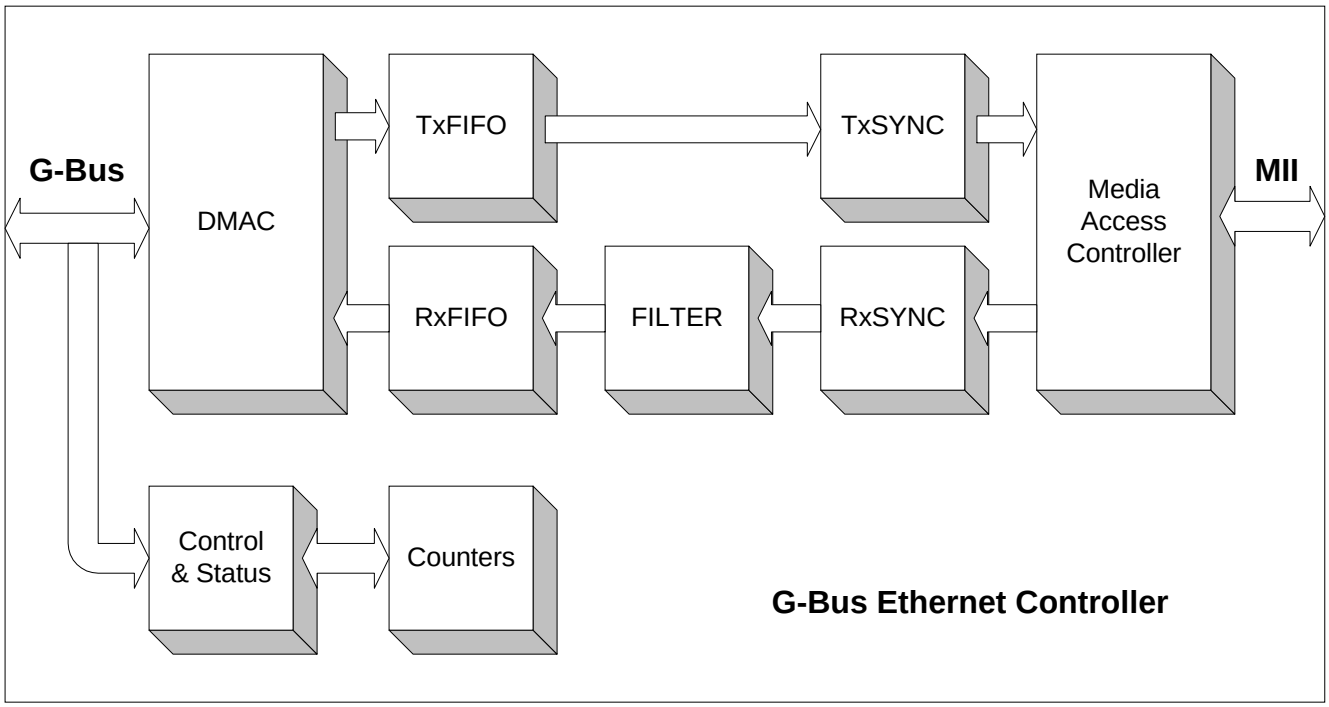


Figure 12-1 TX7901 10/100 MAC Block Diagram

12.1.1 C790 and MAC DMA

The MAC provides a powerful host system interface through its own DMA. It manages the shared memory structures automatically through the frame Descriptors and buffers.

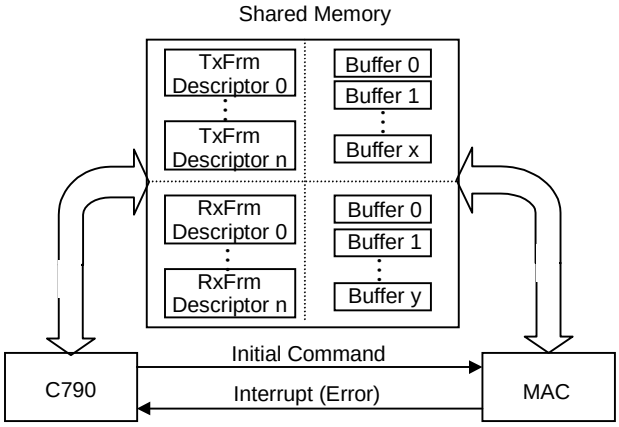


Figure 12-2 MAC Memory Sharing with C790

The MAC's DMA has one channel to transfer data between memory and the MAC. There are four tasks for the DMA:

1. Get a frame Descriptor
2. Update the status in a frame Descriptor
3. Transfer transmit frame data from memory to the Transmit FIFO (TxFIFO)
4. Transfer received frame data from the Receive FIFO (RxFIFO) to memory.

The priority for each task depends on whether the RxFIFO is close to being full or the TxFIFO is close to empty. The software can also select the priority.

The DMA provides logic for controlling the G-Bus master read and write operations. The functions controlled include:

- Burst size control
- Transmit/Receive threshold control
- Automatic Frame Descriptor polling
- Frame Descriptor status update

12.1.2 MAC and MII

The MAC consists of a transmit block, a receive block, a flow control block, a set of control and status registers, counters and a serial controller for the MII (Media Independent Interface) station management interface.

The transmit block buffers the outgoing data in the TxFIFO, encapsulates it, then passes it on to the MII. The transmit block has logic for generating preamble and jam bytes, pad bytes, and the CRC value. It also has logic and a timer for back-off delay after a collision, and a timer for the inter-packet gap after a transmission.

The receive block de-encapsulates the received packet from the MII and stores it in the RxFIFO. The receive block has circuits for checking the CRC value and packet lengths. It also has an address table for accepting or rejecting of a packet based on its destination address.

The TxFIFO and RxFIFO are both 1 KB in size.

Controls for the following network operations are included in the MAC:

- Controls to enable and disable transmit and receive circuits
- Controls to enable and disable interrupts for individual conditions
- Address recognition controls, for up to 16 individual addresses
- Counters and status bits for collecting network management data
- Synchronizing the transmit signals to the transmit clock
- Synchronizing the receive signals to the bus clock
- Controls to enable and disable the PAUSE operation
- Controls for reading and writing data in the TxFIFO and RxFIFO in the diagnostic mode
- Loop-back and other controls to aid in diagnosing problems

12.2 MII (Medium Independent Interface)

Table 12-1 lists the Medium Independent Interface Signals.

Table 12-1 MII Interface Signals

Signal	Input / Output	Description
macxTxClk	Input	Transmit Clock. Provides the timing reference for the transfer of the macxTxEn, macxTxD, and macxTxEr to the PHY.
macxTxEn	Output	Transmit Enable. When High, indicates the presence of nibbles on the MII for transmission. Asserted with the first nibble of the preamble, and remains asserted until all nibbles have been transmitted to the MII. Synchronous with the rising edge of macxTxClk.
macxTxD[3:0]	Output	Transmit Nibble Data. Synchronous with the rising edge of macxTxClk.
macxTxEr	Output	Transmit Coding Error. When macxTxEr is asserted for one or more macxTxClk periods while macxTxEn is also asserted, this signal indicates that one or more symbols that are not part of the valid data in the frame are being transmitted. Synchronous with the rising edge of macxTxClk.
macxRxClk	Input	Receive clock. Provides the timing reference for the transfer of the macxRxDv, macxRxD and macxRxEr from the PHY.
macxRxDV	Input	Receive Data Valid. When asserted, indicates that the PHY is presenting recovered and decoded nibbles on macxRxD[3:0]. Remains asserted continuously from the first nibble through the final nibble of the frame. Synchronous with the rising edge of macxRxClk.
macxRxD[3:0]	Input	Receive Nibble Data. Synchronous with the rising edge of macxRxClk.
macxRxEr	Input	Receive Error. When asserted for one or more macxRxClk periods, indicates that an error was detected somewhere in the frame presently being transferred from the PHY to the MII. Synchronous with the rising edge of macxRxClk.
macxCRS	Input	Carrier Sense. macxCRS should be asserted when either the transmit or receive medium is active, and de-asserted when both are idle. Asynchronous to macxRxClk
macxCOL	Input	Collision Detected. The PHY should assert macxCOL upon detection of a collision on the medium, and remain asserted while the collision condition persists. Asynchronous
macxMDC	Output	MII Management Data Clock. Provides timing reference for transfer of macxMDIO.
macxMDIO	Inout	MII management Data Input/Output. Synchronous with the rising edge of macxMDC.
macxHwFDupSel	Input	Full-Duplex Select. Selects the duplex mode when the HwFDupSelEn bit is set in the Transmit Frame Configuration Register.

Note: In Table 12-1 “Input” refers to signals that go from G-Bus or MII to MAC, while “Output” refers to signals that go from MAC to G-Bus or MII.

12.3 MAC Registers and Counters

Each of the two MACs available in the TX7901 occupies a 4 KB block of address space on the G-Bus for its internal structures. The base addresses of the corresponding blocks for MAC 0 and MAC 1 are 0x1E00_5000 and 0x1E00_6000 respectively.

Thus, to access a given structure (such as a register or a counter) within the appropriate MAC, the offset of that structure must be added to the base address of that MAC to obtain the G-Bus Address.

The address space for each MAC is divided into six regions, corresponding to the following offsets and sizes:

Registers 0x000 ~ 0x1FF (512)

Counters 0x200 ~ 0x3FF (512)

MIIM 0x400 ~ 0x5FF (512)

Address Filtering Table 0x600 ~ 0x7FF (512)

TxFIFO Port Address 0x800 ~ 0xBFF (1024)

RxFIFO Port Address 0xC00 ~ 0xFFF (1024)

The FIFO addresses work in the FIFO Diagnostic Mode to let the C790 read or write a specific location within the appropriate FIFO.

Table 12-2 MAC Configuration Registers

Offset	Register	Width	R / W	Description	Notes
0x000	-	[63:0]	R	Reserved	1,2
0x008	CCReg	[63:0]	R/W	Command & Configuration Register	1
0x010	TFCReg	[63:0]	R/W	Transmit Frame Configuration	1
0x018	RFCReg	[63:0]	R/W	Receive Frame Configuration	1
0x020	TSReg	[63:0]	R	Transmit Status Register	1
0x028	RSReg	[63:0]	R	Receive Status Register	1
0x030	TIMReg	[63:0]	R/W	Transmit Interrupt Mask Register	1
0x038	TIReg	[63:0]	R	Transmit Interrupt Register	1
0x040	RIMReg	[63:0]	R/W	Receive Interrupt Mask Register	1
0x048	RIReg	[63:0]	R	Receive Interrupt Register	1
0x050	-	[63:0]	R/W	Reserved	1,2
0x058	-	[63:0]	R/W	Reserved	1,2
0x060	TPFTReg	[63:0]	R/W	Transmit Pause Frame Timer Register	1
0x068	VLANReg	[63:0]	R/W	VLAN Tag Register	1
0x070	TDPReg	[63:0]	R/W	Transmit Frame Descriptor Pointer Register	1
0x078	RDPReg	[63:0]	R/W	Receive Frame Descriptor Pointer Register	1
0x080	CDPReg	[63:0]	R	Current Frame Descriptor Pointer Register	1
0x088	BusErrReg	[63:0]	R	Bus Error Address Register	1
0x090	TCDReg	[63:0]	R	Transmit frame Current Descriptor Pointer Register	1
0x098	RCDReg	[63:0]	R	Receive frame Current Descriptor Pointer Register	1
0x100	peMACC	[63:0]	R/W	Internal Test Register (peMACC)	1,3
0x108	peMACT	[63:0]	R/W	Internal Test Register (peMACT)	1,3
0x110	IPGReg	[63:0]	R/W	Back-to-Back IPG gap	1

Offset	Register	Width	R / W	Description	Notes
0x118	NBTBReg	[63:0]	R/W	Non Back-to-Back IPG gap	1
0x120	peCLRT	[63:0]	R/W	Internal Test Register (peCLRT)	1,3
0x128	peMAXF	[63:0]	R/W	Internal Test Register (peMAXF)	1,3
0x130	pePNCT	[63:0]	R/W	Internal Test Register (pePNCT)	1,3
0x138	peTBCT	[63:0]	R/W	Internal Test Register (peTBCT)	1,3
0x1A8	LSAII	[63:0]	R/W	Local Station Addr II	1
0x1B0	LASI	[63:0]	R/W	Local Station Addr I	1
0x1C8	peVLTP	[63:0]	R/W	Internal Test Register (peVLTP)	1,3

Table 12-3 MAC Counters

Offset	Counter	Size	R/W	Description	Notes
0x200	TBTCnt	[63:0]	R/W	Total Bytes Transmitted	1
0x208	TGFTCnt	[63:0]	R/W	Total Good Frames Transmitted	1
0x210	MFTCnt	[63:0]	R/W	Multicast Frames Transmitted	1
0x218	BFTCnt	[63:0]	R/W	Broadcast Frames Transmitted	1
0x220	TxFrame64	[63:0]	R/W	Frames Transmitted (TxFrame64)	1
0x228	TxFrame127	[63:0]	R/W	Frames Transmitted (TxFrame127)	1
0x230	TxFrame255	[63:0]	R/W	Frames Transmitted (TxFrame255)	1
0x238	TxFrame511	[63:0]	R/W	Frames Transmitted (TxFrame511)	1
0x240	TxFrame1K	[63:0]	R/W	Frames Transmitted (TxFrame1K)	1
0x248	TxFrameGt1K	[63:0]	R/W	Frames Transmitted (TxFrameGt1K)	1
0x250	MPFTCnt	[63:0]	R/W	MAC Pause Frames Transmitted	1
0x258	LFTCnt	[63:0]	R/W	Long Frames Transmitted	1
0x260	TCCnt	[63:0]	R/W	Total Collisions	1
0x268	LCCnt	[63:0]	R/W	Late Collision	1
0x270	MCCnt	[63:0]	R/W	Multiple Collision	1
0x278	SCCnt	[63:0]	R/W	Single Collision	1
0x280	EDCnt	[63:0]	R/W	Excessive Deferrals	1
0x288	TRECnt	[63:0]	R/W	Transmit Retry Errors	1
0x290	TUECnt	[63:0]	R/W	Transmit Underflow Errors	1
0x298	–	[63:0]	R/W	Reserved	1,2
0x2A0	TBRCnt	[63:0]	R/W	Total Bytes Received	1
0x2A8	TRFRCnt	[63:0]	R/W	Total Readable Frames Received	1
0x2B0	MFRCnt	[63:0]	R/W	Multicast Frames Received	1
0x2B8	BFRCnt	[63:0]	R/W	Broadcast Frames Received	1
0x2C0	RxFrame64	[63:0]	R/W	Frames Received (RxFrame64)	1
0x2C8	RxFrame127	[63:0]	R/W	Frames Received (RxFrame127)	1
0x2D0	RxFrame255	[63:0]	R/W	Frames Received (RxFrame255)	1
0x2D8	RxFrame511	[63:0]	R/W	Frames Received (RxFrame511)	1
0x2E0	RxFrame1K	[63:0]	R/W	Frames Received (RxFrame1K)	1
0x2E8	RxFrameGt1K	[63:0]	R/W	Frames Received (RxFrameGt1K)	1
0x2F0	MCFRCnt	[63:0]	R/W	MAC Pause Frames Received	1
0x2F8	LFRCnt	[63:0]	R/W	Long Frames Received	1
0x300	RECnt	[63:0]	R/W	Receive Errors	1
0x308	FRBCCnt	[63:0]	R/W	Frames Received with Bad CRC	1
0x310	MFRCnt	[63:0]	R/W	Misaligned Frames Received	1
0x318	UFCnt	[63:0]	R/W	Undersized Frames	1
0x320	FFCnt	[63:0]	R/W	Fragmented Frames	1
0x328	JFRCnt	[63:0]	R/W	Jabber Frames Received	1
0x330	NRDMFCnt	[63:0]	R/W	No RxDescriptor Missed Frames	1
0x338	NRMFCnt	[63:0]	R/W	No RxFIFO Missed Frames	1

Table 12-4 MIIM (Media Independent Interface Management) Registers

Offset	Width	R / W	Register	Note
0x400H	[63:0]	R/W	MIIM Control Register	1
0x 408H	[63:0]	R/W	MIIM Data Register	1

Table 12-5 MAC "Perfect Table" Values

Offset	Width	R / W	Register	Note
0x600	[63:0]	R/W	Physical Address 0	4
0x608	[63:0]	R/W	Physical Address 1	4
0x610	[63:0]	R/W	Physical Address 2	4
0x618	[63:0]	R/W	Physical Address 3	4
0x620	[63:0]	R/W	Physical Address 4	4
0x628	[63:0]	R/W	Physical Address 5	4
0x630	[63:0]	R/W	Physical Address 6	4
0x638	[63:0]	R/W	Physical Address 7	4
0x640	[63:0]	R/W	Physical Address 8	4
0x648	[63:0]	R/W	Physical Address 9	4
0x650	[63:0]	R/W	Physical Address A	4
0x658	[63:0]	R/W	Physical Address B	4
0x660	[63:0]	R/W	Physical Address C	4
0x668	[63:0]	R/W	Physical Address D	4
0x670	[63:0]	R/W	Physical Address E	4
0x678	[63:0]	R/W	Physical Address F	4

Notes:

1. In the tables above, registers must be accessed as 64-bit registers, even when their widths are smaller. For such registers, the remaining bits are "Reserved", are ignored when written to, and are read back as zeroes.
2. All registers that are marked as "Reserved" are ignored when written to, and are read back as ones.
3. The contents of Internal Test Registers must be preserved to ensure correct operation.
4. The actual values saved in each Physical Address value are a 48-bit MAC Address in the lower 48 bits and a valid bit at bit 63. The remaining bits (48 through 62) of these registers are also "Reserved" as explained in Note 1.

Table 12-6 MAC Hash Table values

Offset	Width	R / W	Contents
0x600	[63:0]	R/W	Hash Filter (Byte [7:0])
0x608	[63:0]	R/W	Hash Filter (Byte [15:8])
0x610	[63:0]	R/W	Hash Filter (Byte [23:16])
0x618	[63:0]	R/W	Hash Filter (Byte [31:24])
0x620	[63:0]	R/W	Hash Filter (Byte [39:32])
0x628	[63:0]	R/W	Hash Filter (Byte [47:40])
0x630	[63:0]	R/W	Hash Filter (Byte [55:48])
0x638	[63:0]	R/W	Hash Filter (Byte [63:56])
0x640	[63:0]	R/W	Physical Address (Byte [5:0])

12.3.1 Register Functionality and Field Descriptions

MAC Registers are used primarily for configuration and error notification. The MAC also has diagnostic registers that are typically used for system diagnostic testing. Configuration registers are only written to at system start-up, or whenever the chip is reset. They should not be updated while the MAC is transmitting or receiving frames.

A software reset has no effect on the MAC configuration registers. A hardware reset sets the configuration registers to their default values (shown as binary values in parentheses under "Description", if present, in the tables that follow.)

12.3.1.1 Command and Configuration Register (CCReg)

Upon the completion of reset, this register's default value is 0x0080_0160.

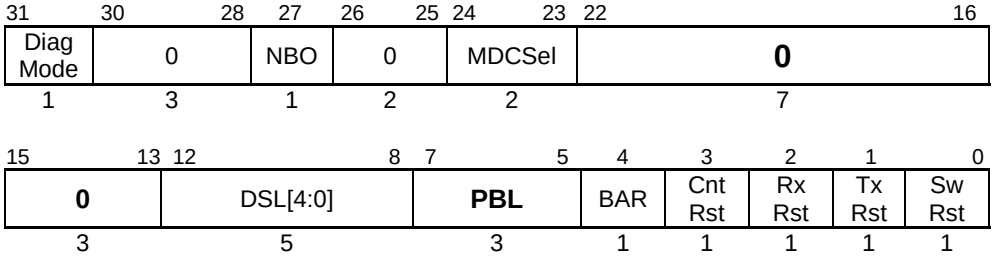


Table 12-7 CCReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31	DiagMode	R/W	When set, the MAC is in the diagnostic mode — all FIFO units are accessible to the C790 and all counters are able to be written to. The read or write is 64-bit aligned for the FIFOs. (0)
30:28	–	R/O	Reserved
27	NBO	R/W	Network Byte Order (0) When set, the most significant byte in the word is stored into the younger address. When cleared, the least significant byte is stored into the younger address.
26:25	–	R/O	Reserved
24:23	MDCSel	R/W	This value determines the frequency of MDC. (01) MDC 00 G-Bus Clock / 24 01 G-Bus Clock / 32 10 G-Bus Clock / 48 11 G-Bus Clock / 64
22:13	–	R/O	Reserved
12:8	DSL[4:0]	R/W	Descriptor Skip Length Specifies the number of 16-byte values to skip between two unchained Descriptors. (00001)

Bit(s)	Field	R/W	Description
7:5	PBL	R/W	Programmable Burst Length (011) Indicates the maximum number of 8-byte values to be transferred in one DMA transaction. X00 2 (16 bytes) X01 4 (32 bytes) X10 8 (64 bytes) X11 16 (128 bytes) It is also a kind of count threshold, and has different definitions in the TxFIFO and the RxFIFO. This threshold guarantees that a PBL space of at least 8 bytes is either free to write to the TxFIFO or to contain data to read from the Rx_FIFO.
4	BAR	R/W	Bus Arbitration Selects the internal bus arbitration between the receive and transmit processes. When set, a round-robin arbitration scheme is applied resulting in equal sharing between processes. When reset, the receive process has priority over the transmit process. (0)
3	CntRst	R/W	Counter Reset. This bit provides counter reset only. It will be kept active for at least 40 gbsBusClk periods. This bit is self-clearing. (0)
2	RxRst	R/W	Receive Port Reset. This bit provides Receive Port reset only. This bit is self-clearing. (0)
1	TxRst	RW	Transmit Port Reset. This bit provides Transmit Port reset only. This bit is self-clearing. (0)
0	SwRst	RW	Software Reset. Reset is immediate and it is equivalent to resetting the counters, the transmitters, the receivers, and the MII management block. It does not affect registers, including interrupt status or diagnostic registers. This bit is self-clearing. (0) Note: The Busy bit in the MII control register must be 0 before this bit is set.

12.3.1.2 Transmit Frame Configuration Register (TFCReg)

The Transmit Frame Configuration Register defines the transmission rules for the MAC. These can be changed to accommodate different options. Upon the completion of reset, this register's default value is 0x2008_0100.

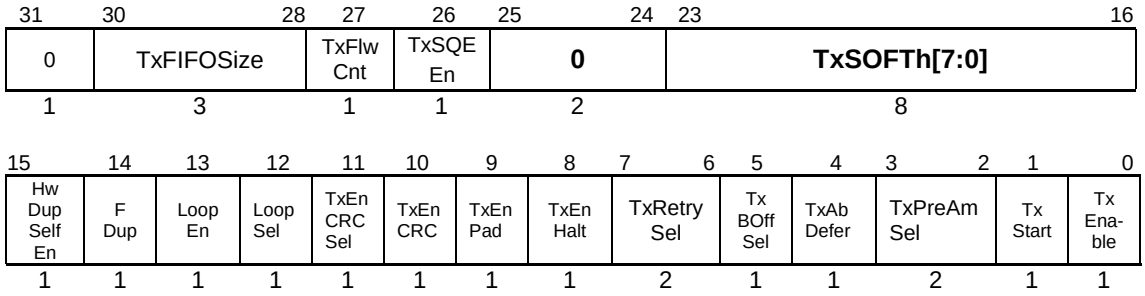


Table 12-8 TFCReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31	–	R/O	Reserved
30:28	TxFifoSize [2:0]	R/W	TxFIFO Size Select (010) 010 : Size = 1 KB (128 x 64 bits) 000, 001, 011-111 : Reserved. Try to avoid changing these values.
27	TxFlwCnt	R/W	Transmit Flow Control This bit must remain cleared. Do not try to set this bit.
26	TxSQEEn	R/W	SQE Test Enable (0) When TxSQE is 0, the transmitter does not report the status of the SQE test performed by

Bit(s)	Field	R/W	Description
			the PHY. When TxSQE is 1, the transmitter reports the status of SQE test. This bit should be 0 if the 100Base-X or 100Base-T PHY being used does not perform the SQE test.
25:24	–	R/O	Reserved
23:16	TxSOFTh [7:0]	R/W	Transmit Start of Frame Threshold (0000_1000) This threshold indicates how many 8-byte units should be written to the Tx_FIFO before a transmit attempt is initiated. This is the minimum requirement to start the frame transmission. If the frame size is less than the TxSFTH, the transmission starts after EOF.
15	HwDupSelEn	R/W	Hardware Full Duplex Select Enable (0) When set, the macxHwFDupSel input pin overrides the setting of the FDup register bit. When reset, the macxHwFDupSel pin is ignored.
14	FDup	R/W	Full-Duplex (0) When set, the port is in the full-duplex mode so that frames can be transmitted and received at the same time. In the full-duplex mode, the transmitter ignores receive carrier and collision indications before and during transmission. When reset, the port works in the half-duplex mode.
13	LoopEn	RW	Loopback Enable (0) The MAC can be configured for loop-back diagnostic operation by setting this bit.
12	LoopSel	R/W	Loop-back Select (0) When set, Loop-back allows transmitted data to be looped back through the TxFIFO and RxFIFO to test DMA and FIFO circuitry. If it is zero, the loop-back goes deeper, including the MAC block. In both situations, data are not transmitted on the MII. Loop-back is only possible in the full-duplex mode.
11	TxEnCRCSel	R/W	Transmit CRC Enable Select (0) When set, TxEnCRC bit is overridden by TxEnCRCDes.
10	TxEnCRC	R/W	Transmit CRC Enable (0) While CRC is low, the CRC check sum is neither calculated nor appended to each transmitted frame.
9	TxEnPad	R/W	Pad Transmit Data (0) While this bit is set, the MAC will automatically append data to a frame less than 60 bytes in length. The TxEnCRC bit should be set also, so a valid CRC is appended to the frame. Otherwise, the frame will be transmitted with a bad CRC. If this bit is reset and a frame less than 60 bytes in length is written to the TxFIFO, this frame is transmitted unchanged.
8	TxEnHalt	R/W	Halt on Error Condition (1) When this bit is set, the TxFIFO will halt after error conditions such as TxFIFO overflow, late collision, excessive deferral, or excessive collision. Resetting the TxStart bit in this register restarts the TxFIFO.
7:6	TxRetrySel [1:0]	R/W	Retry Attempt Select (00) These bits set the number of attempts the transmitter makes to transmit a frame. After this number of attempts, the transmitter aborts the transmission of the frame, resulting in an excessive collision error. Retry[1:0] Transmit Attempts 00 16 (default) 01 8 10 4 11 1
5	TxBOffSel	R/W	Back-off Select (0) When this bit is set, the transmitter will only wait 96 bit times before re-transmitting a frame after a collision. When it is 0, a random backoff time is calculated. The random number of slot times is calculated based on the following algorithm: $T = 512 \times R$ (bit times) $0 \le R \le 2$ ^ (min (n,10)) and n is the number of collisions that have occurred for the current transmit attempt.
4	TxAbDefer	R/W	Abort After Max Deferral (0) When this bit is set, the transmitter aborts a transmission if it is deferring for more than 24,288 bit times.
3:2	TxPreAmSel [1:0]	R/W	Transmit Preamble Length Select (00) These bits must remain zero in order to append a 7-byte preamble and a 1-byte start of frame delimiter to the frame data. Values other than zero are reserved, and should not try

Bit(s)	Field	R/W	Description
			to change.
1	TxStart	R/W	Transmit Start (0) This bit works with TxEnable. If TxEnable is 0, TxStart is ignored. When set, the MAC enters a running state. It will poll the Descriptor first, and then transmit frames. When cleared, it will stop the transmission. If this bit is cleared during the transmission of a frame, the frame is completely transmitted before stopping. This bit is self-clearing when transmission is stopped (error occurred and TxEnhalt is set) or suspended (Descriptor is not available). To resume the transmission process, the driver should set this bit again.
0	TxEnable	R/W	Transmit Port Enable (0) When set, transmission is enabled.

12.3.1.3 Receive Frame Configuration Register (RFCReg)

The Receive Frame Configuration register defines the rules for frame reception on MAC. These can be changed to accommodate different options. Upon the completion of reset, this register's default value is 0x2008_0000.

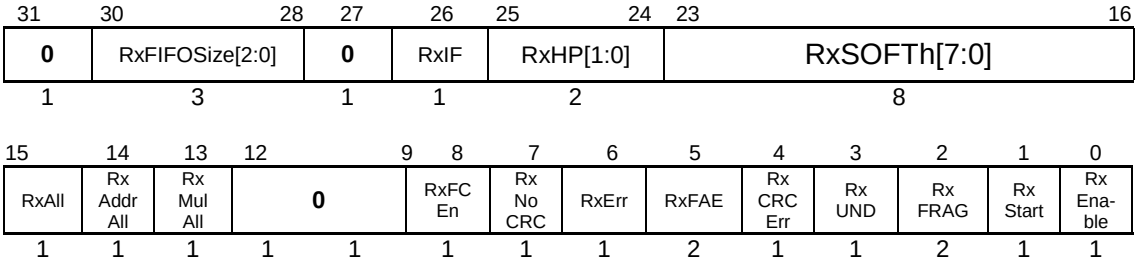


Table 12-9 RFCReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31	–	R/O	Reserved
30:28	RxFifoSize[2:0]	R/W	RxFIFO Size Select (010) 010 : Size = 1 KB (128 x 64 bits) 000, 001, 011, 1xx : Reserved. Try to avoid changing these values.
27	–	R/O	Reserved
26	RxIF	R/W	Inverse Filtering (0) When set, the MAC operates in an inverse filtering mode. This is valid only when in the perfect filtering mode.
25:24	RxHP[1:0]	R/W	Hash/Perfect Receive Filtering Mode (00) 1x : MAC performs imperfect address filtering of all incoming frames according to the hash table. 01: MAC performs imperfect address filtering of multicast frames and perfect address filtering of physical frames 00: MAC performs perfect address filtering of all incoming frames according to the perfect table.
23:16	RxSOFTh [7:0]	R/W	Receive Start of Frame Threshold (0000_1000) This threshold indicates how many 8-byte units of a frame must be received in the RxFIFO before the first DMA request. If the frame size is less than RxSOFTh, the transmission starts after EOF. If this threshold is set higher than 8 (64 bytes), the undersized or fragment frames can be rejected.
15	RxAll	R/W	Receive All (0) When set, all incoming frames are received, regardless of the destination address, and the MAC works in promiscuous mode. When reset, the address match is checked according to Table 12-10.
14	RxAddrAll	R/W	Address Passed All Frames (0) When set, indicates that all incoming frames that passed the address filtering were received, including runt or collided frames and truncated frames caused by FIFO overflow. If any bad received frames are required, promiscuous mode should be set to "1."
13	RxMulAll	R/W	Pass All Multicast (0) When set, indicates that all the incoming frames with a multicast destination address were received.
12:9	–	R/O	Reserved
8	RxFCEn	R/W	Flow Control Enable (0) When set, the MAC will initiate a flow control algorithm when the RxFIFO is overflowing. If the port is configured for full-duplex, a PAUSE command is transmitted. When this bit

Bit(s)	Field	R/W	Description
			is reset, the MAC will not self-initiate any flow control algorithms.
7	RxNoCRC	R/W	Strip CRC (0) When set, the MAC strips the CRC (the last 4 bytes) from all frames being received.
6	RxErr	R/W	Reject Frame MII Receive Error (0) See note below. Event counters are updated regardless of the state of this bit.
5	RxFAE	R/W	Reject Frame Alignment Errors (0) See note. Event counters are updated regardless of the state of this bit.
4	RxCRCErr	R/W	Reject CRC Error Frames (0) See note. Event counters are updated regardless of the state of this bit.
3	RxUND	R/W	Reject Undersized Frames (0) See note. Event counters are updated regardless of the state of this bit.
2	RxFRAG	R/W	Reject Fragments (0) See note. Event counters are updated regardless of the state of this bit.
1	RxStart	R/W	Receive Start (0) This bit works with RxEnable. If the RxEnable is “0,” RxStart is ignored. When set, the MAC enters a running reception state. It will poll the Descriptor first, then transfer the incoming frame data to memory. When cleared, this bit will stop reception. If this bit is cleared during the reception of a frame, the frame is completely received before the port is stopped. This bit is self-clearing when the reception is stopped (error occurred). To resume the transmission process, the driver should set this bit again. If the reception is suspended (Descriptor is not available), it will resume automatically when a new frame arrives.
0	RxEnable	R/W	Receive Port Enable (0) The reception event counters continue to update, regardless of the state of this bit.

Note : If the Reject bits (RFCReg[6:2]) are set, the MAC discards the frame if the RxSOFTh threshold has not been met. If the RxSOFTh threshold has been met, the MAC sends the frame to memory, and updates the Rx status.

Table 12-10 Receive Modes when RxAll is 1 or 0

RxAll	RxMulAll	RxIF	RxHP[1]	RxHP[0]	Filtering Mode
1	x	x	x	x	Promiscuous Mode
0	1	x	x	x	Pass all Multicast
0	0	0	0	0	16 perfect filtering
0	0	1	0	0	Inverse filtering
0	0	0	0	1	512-bit hash for multicast, and one perfect filtering for physical
0	0	0	1	x	512-bit hash only

12.3.1.4 Transmit Status Register (TSReg)

The Transmit Status register is updated after a frame is fully transmitted or the transmission of a frame is aborted due to an error. The register can be read to determine if the frame was successfully transmitted or to determine what errors occurred. Upon the completion of reset, this register's value is 0x0000_0000.

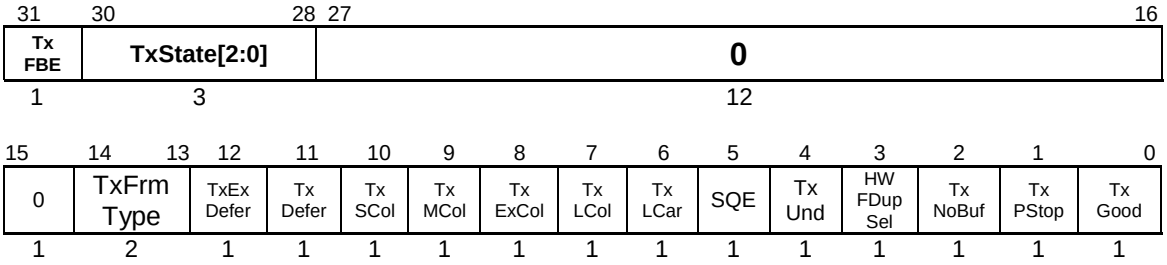


Table 12-11 TSReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31	TxFBE	R/W	Fatal Bus Error (0) When set, indicates that a bus error occurred, and the MAC disables all of its bus access operations.
30:28	TxState [2:0]	R/W	Transmission Process State (000) 000 : Idle, TxEnable is 0 001 : Waiting, FIFO is empty or data is less than TxSOFTh (in TCReg) 010 : Waiting, Descriptor is not available 011 : Stopped, TxStart is reset to zero or Transmit Error occurred and TxEnHalt is set 100 : Suspended, Descriptor is not available 101 : Running, waiting for end of transmission 110 : Running, waiting for end of transmission and next Descriptor is not available 111 : Reserved
27:15	–	R/O	Reserved (0x000, 0)
14:13	TxFrmType	R/W	Transmit Frame Type (00) 00 Ethernet (RFC 894 encapsulation) 01 IEEE802.2 (RFC 1042 encapsulation) 10 VLAN I 11 VLAN II
12	TxExDefer	R/W	Excessive Deferral (0) When set, indicates that the transmission was aborted because of an excessive deferral as defined by the Defer bit in the transmit frame configuration register. This bit is not valid for full-duplex.
11	TxDefer	R/W	Deferred (0) When set, indicates the frame transmission was delayed because of a deferral. This bit is set when a packet is transmitted with a collision and the standard back-off is selected in the configuration register. This bit is not valid while the port is configured for full-duplex.
10	TxSCol	R/W	Single Collision (0) When set, indicates that the packet being transmitted collided only once and was then transmitted successfully on the Ethernet.
9	TxMCol	R/W	Multiple Collisions (0) When set, indicates that the packet being transmitted collided more than once and was then transmitted successfully on the Ethernet.
8	TxExCol	R/W	Collision Error (0) When set, indicates that the packet transmission was aborted because of too many collisions. The number of collision retries allowed is specified in the transmit frame configuration register.
7	TxLCol	R/W	Late Collision (0)

Bit(s)	Field	R/W	Description
			When set, indicates that a transmission was aborted due to a collision occurring later than 512 bit times.
6	TxLCar	R/W	Loss of Carrier (0) When set, indicates the macxCRS input was low during the transmission of a frame.
5	SQE	R/W	Signal Quality Error Missed (0) When set, indicates that the SQE test on the macxCOL signal line was not detected at the end of a transmission.
4	TxUndf	R/W	Transmit Underflow (0) When set, indicates that the TxFIFO had an underflow condition during the frame transmission. Transmission process has been suspended.
3	HwFDupSel	R/W	Pin Status for Hardware Full Duplex Select (0)
2	TxNoBuf	R/W	Transmit Buffer Unavailable (0) When set, indicates that the next Descriptor in the transmit list is owned by the host and cannot be acquired by the MAC. Transmission process is suspended.
1	TxPStop	R/W	Transmit Process Stopped (0) When set, indicates that the transmission process has been suspended.
0	TxGood	R/W	Good Frame (0) When set, indicates that a frame transmission was completed without error. This bit will be set regardless of the state of SQE, TxSCol, TxMCol and TxDefer. This bit will not be set in the case that TxLCol, TxExDefer, TxExCol or TxUndf is reported.

12.3.1.5 Receive Status Register (RSReg)

The Receive Status Register is updated after a frame is received. This register is also copied into the Descriptor in main memory. The error bits are set regardless of the state of the RFCReg (Receive Frame Configuration Register). Upon the completion of reset, this register's default value is 0x0000_0000.

31	30 28	27 19	18	17	16
Rx FBE	**RxState[2:0]**	**0**	**RxD Type**	**RxFF**	**Rx Frm Type [1]**
1	3	9	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxFrm Type (0)	RxBuf Err	RxP Stop	Rx NoBuf	RxErr	RxJab	RxFA Err	Rx CRC Err	Rx Unds	Rx Frag	Rx Long Err	RxPA	RxMA	RXBA	Rx FOver F	Rx Good
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 12-12 RSReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31	RxFBE	R/O	Fatal Bus Error (0) When set, indicates that a bus error occurred, then the MAC disables all of its bus access operations.
30:28	RxState [2:0]	R/O	Receive Process State (000) 000 : Idle, RxEnable is 0 001 : Waiting, there are no frame data in RxFIFO, or data is less than the RxSOFTh (in RCReg) 010 : Waiting, there are no data in RxFIFO, or data is less than the RxSOFTh and there is no Descriptor 011 : Stopped, RxStart is reset to zero or error occurred during reception 100 : Suspended, receive Descriptor is unavailable 101 : Running, queuing the reception frame from the RxFIFO into the receive buffer 110 : Running, queuing the reception frame from the RxFIFO into the receive buffer and the next Descriptor is unavailable 111 : Reserved
27:19	–	R/O	Reserved (0x000)
18	RxDType	R/O	Receive Data Type (0) 0: External Frame 1: Internal Loop-back Frame
17	RxFF	R/O	Receive Frame Filter Fail (0) When set, indicates that the frame failed address recognition filtering. This bit can be set only when the MAC is in promiscuous mode (RxAll bit is set in RFCReg)
16:15	RxFrmType [1:0]	R/O	Receive Frame Type[1:0] (0, 0) 0 0 Ethernet (RFC 894 encapsulation) 0 1 IEEE 802.2 (RFC 1042 encapsulation) 1 0 VLAN I 1 1 VLAN II
14	RxBufErr	R/O	Descriptor Error (0) When set, indicates that frame truncation caused by a frame that does not fit within the current Descriptor buffers occurred, and that the MAC does not own the next Descriptor. The frame is truncated.
13	RxPStop	R/O	Receive Process Stopped (0) When set, indicates that the reception process was suspended.
12	RxNoBuf	R/O	Receive Buffer Unavailable (0) When set, indicates that the next Descriptor in the reception list is owned by the host and cannot be acquired by the MAC. The reception process is suspended.
11	RxErr	R/O	Receive Error (0) When set, indicates that the macxRxEr signal on the MII is asserted while macxRxDv is High. The presence of macxRxEr will also cause the CRC bit to be set and the Frames Received with Bad CRC counter to be incremented.
10	RxJab	R/O	Jabber on Reception (0) When set, indicates that the frame received is greater than the maximum size, with an incorrect CRC.
9	RxFAErr	R/O	Frame Alignment Error (0) When set, indicates that the frame received has a frame alignment error. The CRC is invalid, and the byte count is greater than or equal to 64 bytes.
8	RxCRCErr	R/O	CRC Error (0) When set, indicates that the frame received is greater than or equal to 64 bytes, and had a incorrect CRC.
7	RxUnds	R/O	Undersized Frame (0) When set, indicates that the frame received is less than 64 bytes, and had a correct CRC.
6	RxFrag	R/O	Fragment (0) When set, indicates that the frame received is less than 64 bytes, and had an incorrect CRC.
5	RxLongErr	R/O	Frame Long Error (0) When set, indicates that the frame received is greater than the maximum size with a correct CRC.
4	RxPA	R/O	Physical Address (0)

Bit(s)	Field	R/W	Description
			When set, indicates that the destination address of the received frame is a physical address.
3	RxMA	R/O	Multicast Address (0) When set, indicates that the destination address of the received frame is a multicast address.
2	RxBA	R/O	Broadcast Address (0) When set, indicates that the destination address of the received frame is a broadcast address.
1	RxFOverf	R/O	FIFO Overflow (0) When set, indicates that the frame is not able to be stored in the RxFIFO.
0	RxGood	R/O	Received a Good Frame (0) When set, indicates that reception of a frame has completed. Specific frame status information has been posted in the Descriptor.

12.3.1.6 Transmit Interrupt Mask Register (TIMReg)

The Transmit Interrupt Mask Register enables the corresponding interrupt bits from the Transmit Interrupt Register to drive the macgINTB signal Low. When a transmit event occurs or a counter overflows, it sets a bit in the Transmit Interrupt Register, and the the macgINTB signal becomes active if the corresponding enable bit in this register is set to 1. When the bit in this register is 0, the macgINTB signal is not driven when a bit is set to 1 in the Transmit Interrupt Register. Upon the completion of reset, this register defaults to 0x0000_0000.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						TxFBErrM	TxStopM	TxGoodM	TxLColM	TxExDefM	TxExColM	TxFUndfM	TxCUndfM	TxCRtyM	ExCDefMTx
6						1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCMColM	TxCSColM	TxCLColM	TxCColM	TxCLongM	TxCPauseM	TxCGt1KM	TxC1KM	TxC511M	TxC255M	TxC127M	TxC64M	TxCBCM	TxCMCM	TxCFrmM	TxCByteM
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 12-13 TIMReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:26	–	R/O	Reserved (0x00)
25	TxFBErrM	R/W	Transmission Fatal Bus Error Mask (0)
24	TxStopM	R/W	Frame Transmission Stopped Mask (0)
23	TxGoodM	R/W	A Good Frame Transmitted Mask (0)
22	TxLColM	R/W	Late Collision Mask (0)
21	TxExDefM	R/W	Excessive Deferral Mask (0)
20	TxExColM	R/W	Collision Error Mask (0)
19	TxFUndfM	R/W	Transmit FIFO Underflow Error Mask (0)
18	TxCUndfM	R/W	Transmit Underflows Counter Overflow Mask (0)
17	TxCRtyM	R/W	Transmit Retry Errors Counter Overflow Mask (0)
16	ExCDefMTx	R/W	Excessive Deferrals Counter Overflow (0)
15	TxCMColM	R/W	Multiple Collision Counter Overflow Mask (0)
14	TxCSColM	R/W	Single Collision Counter Overflow (0)
13	TxCLColM	R/W	Late Collisions Counter Overflow Mask (0)
12	TxCColM	R/W	Total Collision Counter Overflow Mask (0)
11	TxCLongM	R/W	Long Frames Transmitted Counter Overflow Mask (0)

Bit(s)	Field	R/W	Description
10	TxCPauseM	R/W	MAC Pause Frames Transmitted Counter Overflow Mask (0)
9	TxCGt1KM	R/W	Frames Transmitted (1024~max byte) Counter Overflow Mask (0)
8	TxC1KM	R/W	Frames Transmitted (512~1023 byte) Counter Overflow Mask (0)
7	TxC511M	R/W	Frames Transmitted (256~511 byte) Counter Overflow Mask (0)
6	TxC255M	R/W	Frames Transmitted (128~255 byte) Counter Overflow Mask (0)
5	TxC127M	R/W	Frames Transmitted (65~127 byte) Counter Overflow Mask (0)
4	TxC64M	R/W	Frames Transmitted (64 byte) Counter Overflow Mask (0)
3	TxCBCM	R/W	Broadcast Frames Transmitted Counter Overflow Mask (0)
2	TxCMCM	R/W	Multicast Frames Transmitted Counter Overflow Mask (0)
1	TxCFrmM	R/W	Total Good Frames Transmitted Counter Overflow Mask (0)
0	TxCByteM	R/W	Total Bytes Transmitted Counter Overflow Mask (0)

12.3.1.7 Transmit Interrupt Register (TIReg)

The Transmit Interrupt Register has the interrupt events. If an event occurs whose interrupt is enabled by setting the corresponding mask bit in the Transmit Interrupt Mask Register, it causes an interrupt. This register is automatically cleared to zero after being read. Upon the completion of reset, this register's value is 0x0000_0000.

31						26	25	24	23	22	21	20	19	18	17	16
0							TxFB Err	Tx Stop	Tx Good	Tx LCol	TxEx Def	TxEx Col	TxF Undf	TxC Undf	TxC Rty	ExC Def
6							1	1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCL Col	TxCM Col	TxCS Col	TxC Col	TxC Long	TxC Pause	TxC Gt1K	TxC 1K	TxC 511	TxC 255	TxC 127	TxC 64	TxC BC	TxC MC	TxC Frm	TxC Byte
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 12-14 TIReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:26	–	R/O	Reserved (0x00)
25	TxFBErr	R/W	Transmit Fatal Bus Error (0)
24	TxStop	R/W	Frame Transmission Stopped or suspended (0)
23	TxGood	R/W	A Good Frame Transmitted (0)
22	TxLCol	R/W	Late Collision (0)
21	TxExDef	R/W	Excessive Deferral (0)
20	TxExCol	R/W	Collision Error (0)
19	TxFUndf	R/W	Transmit FIFO Underflow Error (0)
18	TxCUndf	R/W	Transmit Underflows Counter Overflow (0)
17	TxCRty	R/W	Transmit Retry Errors Counter Overflow (0)
16	ExCDef	R/W	Excessive Deferrals Counter Overflow (0)
15	TxCLCol	R/W	Late Collisions Counter Overflow (0)
14	TxCMCol	R/W	Multiple Collision Counter Overflow (0)
13	TxCSCol	R/W	Single Collision Counter Overflow (0)
12	TxCCol	R/W	Total Collision Counter Overflow (0)
11	TxCLong	R/W	Long Frames Transmitted Counter Overflow (0)
10	TxCPause	R/W	MAC Pause Frames Transmitted Counter Overflow (0)
9	TxCGt1K	R/W	Frames Transmitted (1024~max byte) Counter Overflow (0)
8	TxC1K	R/W	Frames Transmitted (512~1023 byte) Counter Overflow (0)

Bit(s)	Field	R/W	Description
7	TxC511	R/W	Frames Transmitted (256~511 byte) Counter Overflow (0)
6	TxC255	R/W	Frames Transmitted (128~255 byte) Counter Overflow (0)
5	TxC127	R/W	Frames Transmitted (65~127 byte) Counter Overflow (0)
4	TxC64	R/W	Frames Transmitted (64 byte) Counter Overflow (0)
3	TxCBC	R/W	Broadcast Frames Transmitted Counter Overflow (0)
2	TxCMC	R/W	Multicast Frames Transmitted Counter Overflow (0)
1	TxCFrm	R/W	Total Good Frames Transmitted Counter Overflow (0)
0	TxCByte	R/W	Total Bytes Transmitted Counter Overflow (0)

12.3.1.8 Receive Interrupt Mask Register (RIMReg)

The bits of the Receive Interrupt Mask Register enable the corresponding interrupt bits of the Receive Interrupt Register to cause an interrupt. When a receive event occurs or a counter overflows, it sets one of the bits in the Receive Interrupt Register, and the macgIntB signal becomes active if the corresponding enable bit in this register is set to 1. When any bit in this register is 0 and the corresponding bit is set to 1 in the Receive Interrupt Register, it is masked (hidden), preventing it from driving the macgIntB signal and causing an interrupt. Upon the completion of reset, this register‘s value is 0x0000_0000.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							RxFB ErrM	Rx Stop M	Rx Read FrmM	Rx Buf ErrM	RxF Overf M	RxC NoFi FM	RxC No DesM	RxC JabM	ExC Frag M
7							1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxC Unds M	RxC NoAln M	RxC CRC M	RxC ErrM	RxC Long M	RxC Pause M	RxC Gt1K M	RxC 1KM	RxC 511M	RxC 255M	RxC 127M	RxC 64M	RxC BCM	RxC MCM	RxC FrmM	RxC Byte M
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 12-15 RIMReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:25	–	R/O	Reserved (0x00)
24	RxFBErrM	R/W	Receive Fatal Bus Error Mask (0)
23	RxStopM	R/W	Receive Stopped Mask (0)
22	RxReadFrmM	R/W	A Readable Frame received Mask (0)
21	RxBufErrM	R/W	Truncated Frame due to no more Descriptor Mask (0)
20	RxFOverfM	R/W	Receive FIFO Overflow Error Mask (0)
19	RxCNoFiFM	R/W	No RxFIFO Missed Frames Counter Overflow Mask (0)
18	RxCNoDesM	R/W	No RxDescriptor Missed Frame Counter Overflow Mask (0)
17	RxCJabM	R/W	Jabber Frames Received Counter Overflow Mask (0)
16	ExCFragM	R/W	Fragments Received Counter Overflow Mask (0)
15	RxCUndsM	R/W	Undersized Frames Counter Overflow Mask (0)
14	RxCNoAlnM	R/W	Misaligned Frames Counter Overflow Mask (0)
13	RxCCRCM	R/W	Frames Received with Bad CRC Counter Overflow Mask (0)
12	RxCErrM	R/W	Receive Errors Counter Overflow Mask (0)
11	RxCLongM	R/W	Long Frames Received Counter Overflow Mask (0)
10	RxCPauseM	R/W	MAC Pause Frames Received Counter Overflow Mask (0)
9	RxCGt1KM	R/W	Frames Received (1024~max byte) Counter Overflow Mask (0)
8	RxC1KM	R/W	Frames Received (512~1023 byte) Counter Overflow Mask (0)

Bit(s)	Field	R/W	Description
7	RxC511M	R/W	Frames Received (256~511 byte) Counter Overflow Mask (0)
6	RxC255M	R/W	Frames Received (128~255 byte) Counter Overflow Mask (0)
5	RxC127M	R/W	Frames Received (65~127 byte) Counter Overflow Mask (0)
4	RxC64M	R/W	Frames Received (64 byte) Counter Overflow Mask (0)
3	RxCBCM	R/W	Broadcast Frames Received Counter Overflow Mask (0)
2	RxCMCM	R/W	Multicast Frames Received Counter Overflow Mask (0)
1	RxCFrmM	R/W	Readable Frames Received Counter Overflow Mask (0)
0	RxCByteM	R/W	Total Byte Received Counter Overflow Mask (0)

12.3.1.9 Receive Interrupt Register (RIReg)

The Receive Interrupt Register stores interrupt events. If an event occurs whose interrupt is enabled by setting the corresponding bit in the Receive Interrupt Mask Register, it causes an interrupt. This register is automatically cleared to zero after being read. Upon the completion of reset, this register's value is 0x0000_0000.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							RxFB Err	Rx Stop	Rx Read Frm	Rx Buf Err	RxF Overf	RxC NoFi F	RxC No Des	RxC Jab	ExC Frag
7							1	1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxC Unds	RxC NoAln	RxC CRC	RxC Err	RxC Long	RxC Pause	RxC Gt1K	RxC 1K	RxC 511	RxC 255	RxC 127	RxC 64	RxC BC	RxC MC	RxC Frm	RxC Byte
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 12-16 RIReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:25	-	R/O	Reserved (0x00)
24	RxFBErr	R/W	Receive Fatal Bus Error (0)
23	RxStop	R/W	Reception Stopped or Suspended (0)
22	RxReadFrm	R/W	Received a Readable Frame (0)
21	RxBufErr	R/W	Truncated Frame due to no more Descriptor (0)
20	RxFOverf	R/W	Receive FIFO Overflow Error (0)
19	RxCNoFiF	R/W	No RxFIFO Missed Frames Counter Overflow (0)
18	RxCNoDes	R/W	No RxDescriptor Missed Frame Counter Overflow (0)
17	RxCJab	R/W	Jabber Frames Received Counter Overflow (0)
16	ExCFrag	R/W	Fragments Received Counter Overflow (0)
15	RxCUnds	R/W	Undersized Frames Counter Overflow (0)
14	RxCNoAln	R/W	Misaligned Frames Counter Overflow (0)
13	RxCCRC	R/W	Frames Received with Bad CRC Counter Overflow (0)
12	RxCErr	R/W	Receive Errors Counter Overflow (0)
11	RxCLong	R/W	Long Frames Received Counter Overflow (0)
10	RxCPause	R/W	MAC Pause Frames Received Counter Overflow (0)
9	RxCGt1K	R/W	Frames Received (1024~max byte) Counter Overflow (0)
8	RxC1K	R/W	Frames Received (512~1023 byte) Counter Overflow (0)
7	RxC511	R/W	Frames Received (256~511 byte) Counter Overflow (0)
6	RxC255	R/W	Frames Received (128~255 byte) Counter Overflow (0)
5	RxC127	R/W	Frames Received (65~127 byte) Counter Overflow (0)
4	RxC64	R/W	Frames Received (64 byte) Counter Overflow (0)

Bit(s)	Field	R/W	Description
3	RxCBC	R/W	Broadcast Frames Received Counter Overflow (0)
2	RxCMC	R/W	Multicast Frames Received Counter Overflow (0)
1	RxCFrm	R/W	Readable Frames Received Counter Overflow (0)
0	RxCByte	R/W	Total Byte Received Counter Overflow (0)

12.3.1.10 LSAReg I, II, Local Station Address I & II Registers

These address registers are used by the MAC when transmitting MAC control frames. The data in this register should be written by the host during initialization. There are two Station Address registers, named the Station Address I and II Registers. For the MAC Station Address SA5:SA4:SA3:SA2:SA1:SA0 the registers should be written as follows:

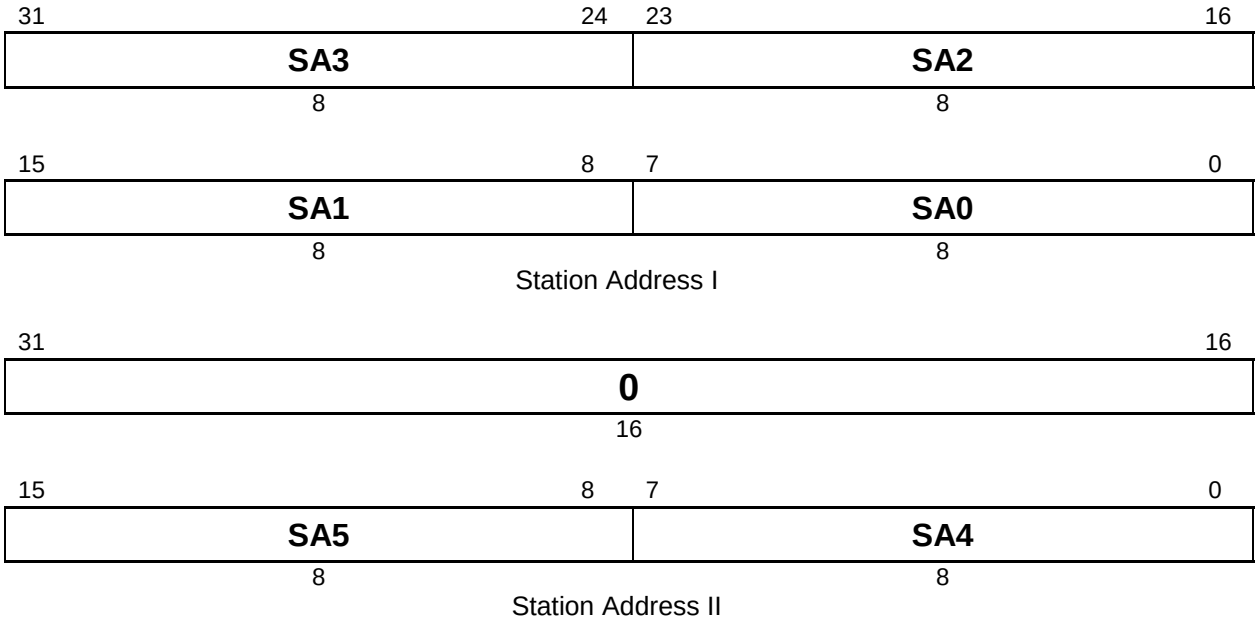


Table 12-17 Station Address I Register Field Descriptions

Bits	Field	R/W	Description
31:24	SA3	R/W	MAC Station Address Bits for SA3 (0x00)
23:16	SA2	R/W	MAC Station Address Bits for SA2 (0x00)
15:8	SA1	R/W	MAC Station Address Bits for SA1 (0x00)
7:0	SA0	R/W	MAC Station Address Bits for SA0 (0x00)

Table 12-18 Station Address II Register Field Descriptions

Bits	Field	R/W	Description
31:24	–	R/O	Reserved
15:8	SA5	R/W	MAC Station Address Bits for SA5 (0x00)
7:0	SA4	R/W	MAC Station Address Bits for SA4 (0x00)

12.3.1.11 Bus Error Address Register (BusErrReg)

The Bus Error Address Register saves the G-Bus Address when a Bus Error occurs while the MAC is the G-Bus Master. Upon the completion of reset, this register's default value is 0x0000_0000.

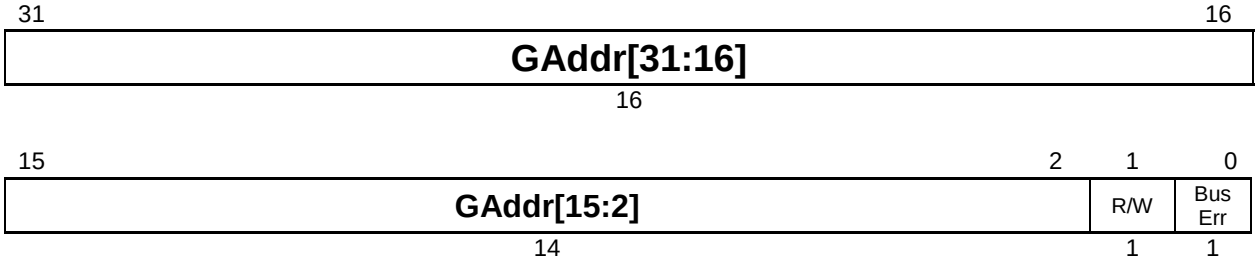


Table 12-19 Bus Error Address Register Field Descriptions

Bit(s)	Field	R/W	Description
31:2	GAddr[31:2]	R/W	Error Address (0x0000)
1	R/W	R/W	Read/Write Operation (0) When set, indicates a read operation
0	BusErr	R/W	When set, indicates the address is valid, i.e. Bus Error has been captured. (0)

12.3.1.12 Transmit Pause Frame Timer Register (TPFTReg)

The Transmit Pause Frame Timer Register is used by the MAC control sub-layer when the MAC is configured for full-duplex operation and flow control is requested by the host or an RxFIFO underflow. The contents of this register are included in the MAC control frame as the n-slots variable, where n-slots represents the number of slot times (512 bit times/slot) for which the receiving station shuts off its transmitter. The number of valid slot times can range from 0 to 65535. Therefore, all bits in the transmit pause frame timer register are valid. Upon the completion of reset, this register's default value is 0x0000_0000.

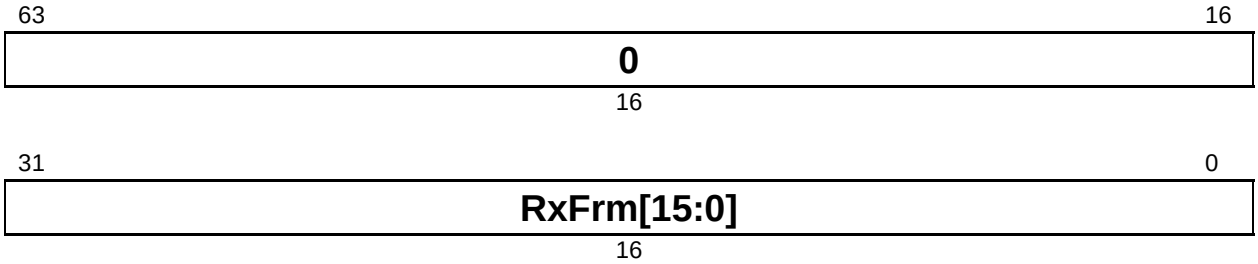


Table 12-20 TPFTReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:16	–	R/O	Reserved (0x0000)
15:0	RxFrm	R/W	Data Bits (0x0000)

12.3.1.13 VLAN Tag Register (VLANReg)

When frames are transmitted or received, the 13th and 14th byte in the frame are compared to this register to determine if the frame is tagged with a one-level VLAN ID or a two-level VLAN ID. Upon the completion of reset, this register's default value is 0x0000_0000.

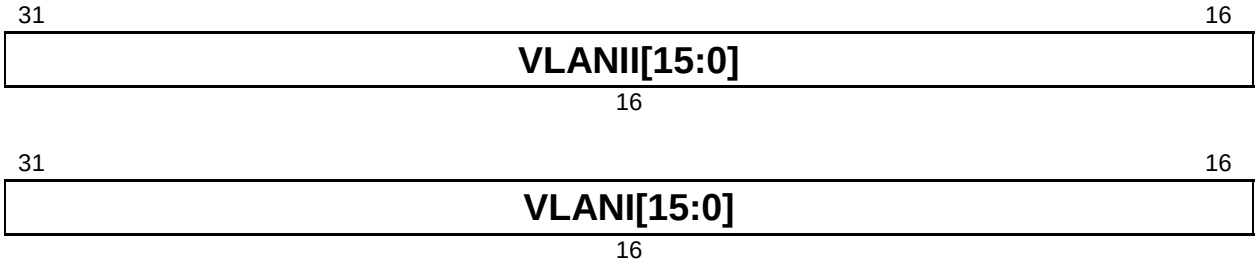


Table 12-21 VLAN Tag Register Field Descriptions

Bit(s)	Field	R/W	Description
31:16	VLANII[15:0]	R/W	Two-Level VLAN tag ID (0x0000)
15:11	VLANI[15:0]	R/W	One-Level VLAN tag ID (0x0000)

12.3.1.14 Transmit Frame Descriptor Pointer Register (TDPReg)

The Transmit Descriptor Pointer Register contains the address of the first Descriptor of the frame to be transmitted. The system must set this register to a properly initialized frame Descriptor before it enables transmission. A valid address must be aligned on a 16-byte boundary (i.e. only using G-Bus Address Bits [31:4].) Upon the completion of reset, this register's value is 0x0000_0000.

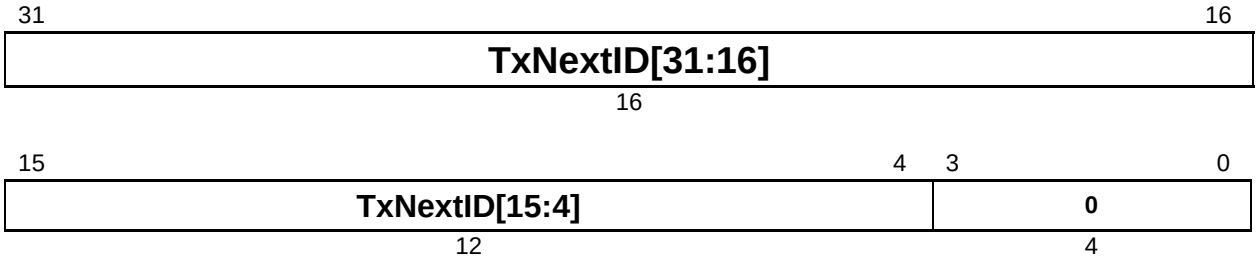


Table 12-22 TDPReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:4	TxNextD	R/O	First Transmit Descriptor Address [31:4] (0x0000)
3:0	–	R/O	Reserved

12.3.1.15 Receive Frame Descriptor Pointer Register (RDPReg)

The Receive Frame Descriptor Pointer Register contains the address of the first frame Descriptor for reception. The system must set this register to a properly initialized frame Descriptor before enabling reception. A valid address must be aligned to a 16-byte boundary. Upon the completion of reset, this register's default value is 0x0000_0000.

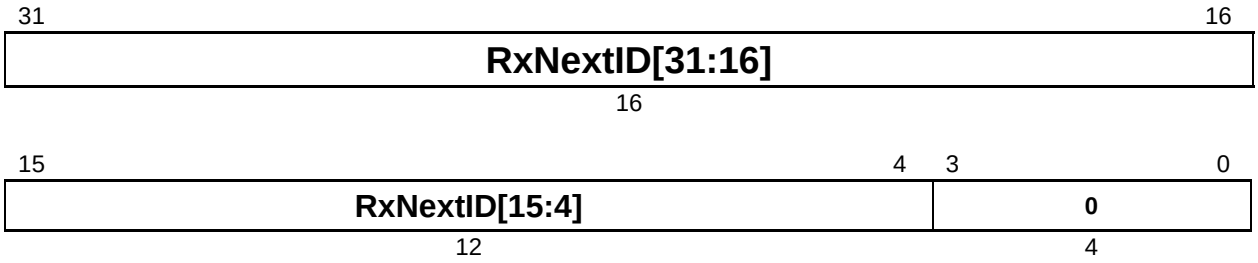


Table 12-23 RDPReg Register Field Descriptions

Bits	Field	R/W	Description
31:4	RxNextD	R/O	First Receive Descriptor Address [31:4] (0x0000)
3:0	–	R/O	Reserved

12.3.1.16 Current Descriptor Pointer Register (CDPReg)

The Current Descriptor Pointer Register contains the address of the last Frame Descriptor which the MAC DMA accessed. Upon the completion of reset, this register's default value is 0x0000_0000.

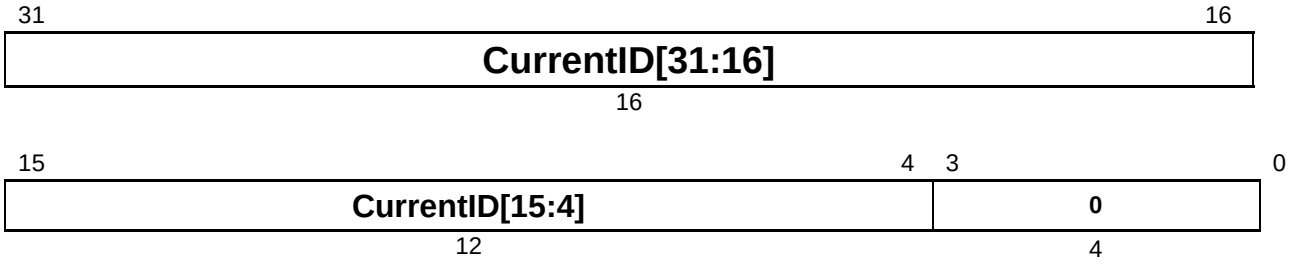


Table 12-24 CDPReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:4	CurrentD	R/W	Descriptor Address [31:4] (0x0000)
3:0	–	R/O	Reserved (0000)

12.3.1.17 Transmit frame Current Descriptor Pointer Register (TCDReg)

The Transmit frame Current Descriptor Pointer Register contains the address of the current Descriptor used for transmission. When the MAC Tx is stopped or suspended (i.e. TFCReg[TxStart] = 0), this register shows the Descriptor where the error occurred or where something is not available.

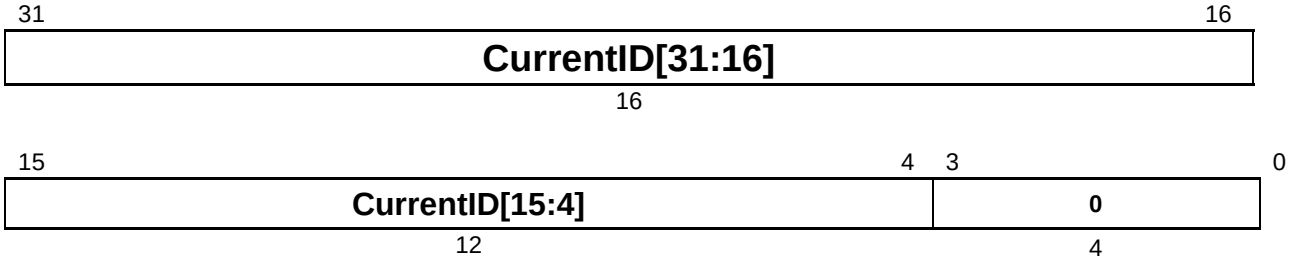


Table 12-25 TCDReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:4	CurrentD	R/W	Descriptor Address [31:4] (0x0000)
3:0	–	R/O	Reserved(0000)

12.3.1.18 Receive frame Current Descriptor Pointer Register (RCDReg)

The Receive frame Current Descriptor Pointer Register contains the address of the current Descriptor used for reception. When the MAC Rx is stopped of is suspended (i.e. RFCReg[RxStart = 0]), this register shows the Descriptor where the error occurred or where something was not available.

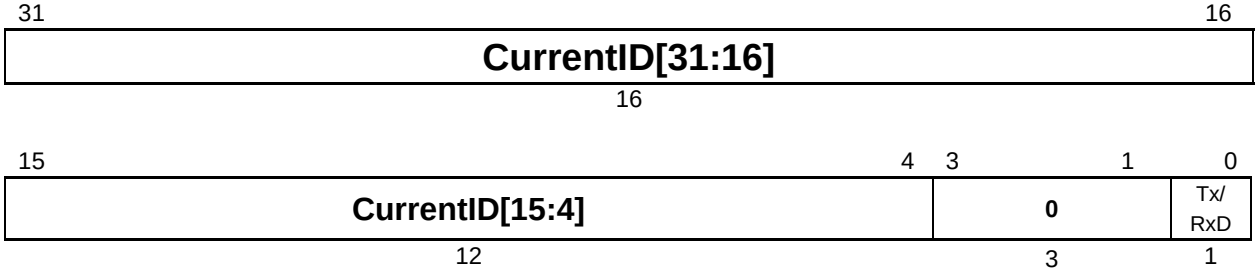


Table 12-26 RCDReg Register Field Descriptions

Bit(s)	Field	R/W	Description
31:4	CurrentD	R/W	Descriptor Address [31:4] (0x0000)
3:1	–	R/O	Reserved (000)
0	Tx/RxD	R/W	When Set, indicates the Transmit Frame Descriptor (0)

12.3.1.19 Back to Back IPG Register (IPGReg)

This Register contains the first bus error address.

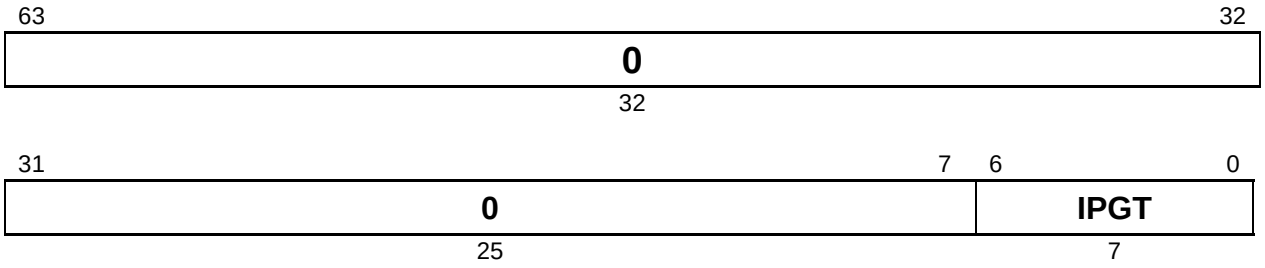


Table 12-27 IPGReg Register Field Descriptions

Bits	Field	R/W	Description
63:7	–	R/W	Reserved (0x000)
6:0	IPGT	R/W	Back-To-Back IPG length. Default is 0x15.

Inter-Packet Gap (IPG) is the measurement between the last nibble of CRC and the first nibble of the preamble of the next packet.

In 100 Mb Ethernet, the IPG is defined as 0.96 microseconds. In 10 Mb Ethernet, the IPG is defined as 9.6 microseconds.

The IPG Time (IPGT) is used to space back-to-back transmit packets. The transmit state machine has an intrinsic delay of 6 clock cycles (macxTxClk) between packets. In other words, with IPGT set to 0, the resultant IPG will be 6 clock cycles. Thus the equation is:

$$IPG = (6 + IPGT) \times T$$

where T is the period of macxTxClk.

Here is a simple chart of IPGT values for 100 Mb and 10 Mb Ethernet:

Table 12-28 Inter Packet Gap T values for 100 Mb / 10 Mb

IPGT	100 Mb	10 Mb
0x01	0.28µs	2.8µs
0x05	0.44µs	4.4µs
0x09	0.60µs	6.0µs
0x0D	0.76µs	7.6µs
0x12*	0.96µs	9.6µs
0x1D	1.40µs	14.0µs

Where T = 40µs for 100 Mb and T = 400 µs for 10 Mb.

*:IEEE 802.3 standard recommended (0.96µs/9.6µs)== default

12.3.1.20 Non Back-To-Back IPG Register (NBTBReg)

This Register contains the programmable transmit IPG for non back-to-back transmits. This register default value is 0x0012.

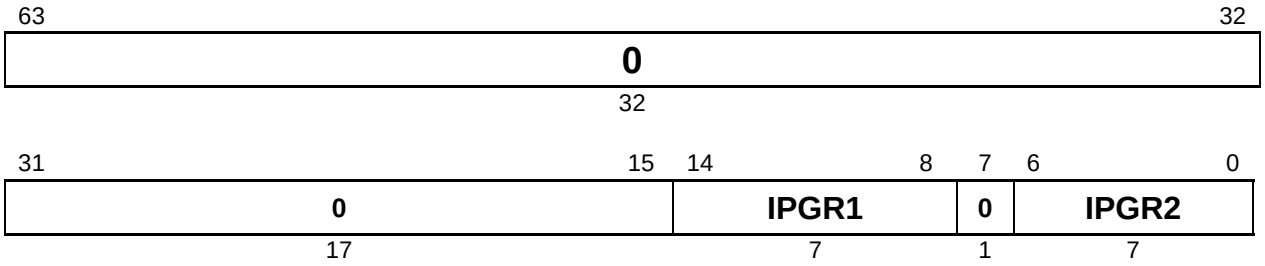


Table 12-29 NBTBReg Register Field Descriptions

Bit(s)	Field	R/W	Description
63:15	–	R/O	Reserved (0)
14:8	IPGR1	R/W	Non Back-To-Back IPG part 1 (0x00)
7	–	R/O	Reserved (0)
6:0	IPGR2	R/W	Non Back-To-Back IPG part 2 (0x12)

The IPGT is used to space non back-to-back transmit packets. The equation is:

$$IPG = (6 + IPGR1 + IGPR2) \times T$$

where T is the period of macxTxClk.

Following is a simple chart of IPGT values for 100 Mb and 10 Mb Ethernet:

Table 12-30 IPGT values for 100 Mb and 10 Mb

IPGR 1+2	100 Mb	10 Mb
00H	0.24us	2.4us
12H	0.96us	9.6us

12.3.2 Counters

The MAC provides an extensive list of network event counters. The counters are cleared to zero upon a hardware reset or software reset (CntRst – see Table 12-7). The counters will count *all* events *even when* the port is disabled or the RxFIFO overflows. When a counter reaches FFFFH, an interrupt bit is set, and the counter rolls over and continues counting from 0.

The following section defines the conditions that increment each of the counters. Counters can also be written to through the G-Bus interface when in diagnostic mode (see Table 12-7). However they should *not* be written during normal operation.

12.3.2.1 Total Bytes Transmitted

This count is the exact number of bytes transmitted on the media by this port. This counter does not include the Preamble and SFD. It does include the bytes before a collision or TxFIFO underflow occurs.

12.3.2.2 Total Good Frame Transmitted

This counter does not count those frames that are transmitted with errors (i.e. collision fragments, partial frames due to FIFO underflow, and frame transmissions that are aborted due to excessive deferral, excessive collisions, or late collisions).

12.3.2.3 Multicast Frames Transmitted

This counter counts the good multicast frames that are transmitted.

12.3.2.4 Broadcast Frames Transmitted

This counter counts the good Broadcast frames that are transmitted.

12.3.2.5 Frames Transmitted (TxFrame64)

This 32-bit counter counts the successfully transmitted frames that are 64 bytes in length (including the CRC).

12.3.2.6 Frames Transmitted (TxFrame127)

This 32-bit counter counts the successfully transmitted frames that are between 65 and 127 bytes (both inclusive) in length (including the CRC).

12.3.2.7 Frames Transmitted (TxFrame255)

This 32-bit counter counts the successfully transmitted frames that are between 128 and 255 bytes (both inclusive) in length (including the CRC).

12.3.2.8 Frames Transmitted (TxFrame511)

This 32-bit counter counts the successfully transmitted frames that are between 256 and 511 bytes

12.3.2.9 Frames Transmitted (TxFrame1K)

This 32-bit counter counts the successfully transmitted frames that are between 512 and 1023 bytes (both inclusive) in length (including the CRC).

12.3.2.10 Frames Transmitted (TxFrameGt1K)

This 32-bit counter counts the successfully transmitted frames that are greater than or equal to 1024 bytes and less than or equal to the maximum size. The upper valid frame size is adjusted and can be set to 1518, 1522, or 1538.

12.3.2.11 MAC Pause Frames Transmitted

This 32-bit counter counts the Pause frames transmitted successfully.

12.3.2.12 Long Frames Transmitted

This 32-bit counter counts the transmitted frames that are greater than the maximum size and do not have any errors.

12.3.2.13 Total Collision

This counter counts the number of collisions during transmission attempts, and includes late collisions as well as all legal collisions.

12.3.2.14 Late Collision

This 32-bit counter counts the number of late collisions.

12.3.2.15 Single Collision

This 32-bit counter counts the number of times a frame is transmitted successfully with one collision. This count does not include frames that are transmitted with errors or collisions that are forced during half-duplex flow control.

12.3.2.16 Multiple Collision

This 32-bit counter counts the number of times a frame is transmitted successfully with more than one collision. This count does not include frames that are transmitted with errors or collisions that are forced during half-duplex flow control.

12.3.2.17 Excessive Deferrals

This 32-bit counter counts the transmissions that failed because the transmitter deferred past 24,288 bit times and the transmission never started. The deferral timer is started at the initiation of each transmission attempt, regardless of the number of retry attempts.

12.3.2.18 Transmit Retry Errors

This 32-bit counter counts the transmissions that failed because the number of collision retry attempts was greater than the threshold defined in the Transmit Frame Configuration Register.

12.3.2.19 Transmit Underflow Errors

This 32-bit counter counts the number of TxFIFOs that underflowed.

12.3.2.20 Total Bytes Received

This 32-bit counter counts all received bytes. This includes CRC bytes and bytes from erroneous frames.

12.3.2.21 Total Readable Frames Received

This 32-bit counter counts the good frames received. It includes the frames that are greater than 63 bytes and less than or equal to the maximum size bytes, and are otherwise well-formed (i.e., correct CRC, and RXER is not asserted when receiving the frame.)

12.3.2.22 Multicast Frames Received

This 32-bit counter counts the multicast frames received successfully. This does not include the broadcast frames.

12.3.2.23 Broadcast Frames Received

This 32-bit counter counts the broadcast frames received successfully.

12.3.2.24 Frames Received (RxFrame64)

This 32-bit counter counts the frames received successfully that are 64 bytes in length (including CRC).

12.3.2.25 Frames Received (RxFrame127)

This 32-bit counter counts the frames received successfully that are between 65 and 127 bytes (both inclusive) in length (including CRC).

12.3.2.26 Frames Received (RxFrame255)

This 32-bit counter counts the frames received successfully that are between 128 and 255 bytes (both inclusive) in length (including CRC).

12.3.2.27 Frames Received (RxFrame511)

This 32-bit counter counts the successfully received frames that are between 256 and 511 bytes (both inclusive) in length (including CRC).

12.3.2.28 Frames Received (RxFrame1K)

This 32-bit counter counts the successfully received frames that are between 512 and 1023 bytes (both inclusive) in length (including CRC).

12.3.2.29 Frames Received (RxFrameGt1K)

This 32-bit counter counts the successfully received frames that are greater than or equal to 1024 bytes and less than or equal to the maximum size. The upper valid frame size can be set to 1518, 1522, or 1538 bytes.

12.3.2.30 MAC Pause Frames Received

This 32-bit counter counts the good MAC Pause frames that are received.

12.3.2.31 Long Frames Received

This 32-bit counter counts the received frames that are greater than the maximum size and do not have any errors.

12.3.2.32 Receive Errors

This 32-bit counter counts the received frames with the RXER signal asserted. The RXER signal indicates that the PHY detected an error that the MAC may not be able to detect.

12.3.2.33 Frames Received with Bad CRC

This 32-bit counter counts the received frames with an incorrect CRC that are aligned on an 8-bit boundary. The frames must be greater than 63 bytes and less than or equal to the maximum size bytes.

If the RXER signal is asserted by the PHY during the reception of a frame, this counter will be incremented in addition to the Receive Error Counter.

12.3.2.34 Misaligned Frames Received

This 32-bit counter counts the received frames with an incorrect CRC that are not aligned on an 8-bit boundary. The frames must be greater than 63 bytes and less than or equal to the maximum size bytes. The RXER signal must not have been asserted at any time during frame reception.

12.3.2.35 Fragmented Frames

This 32-bit counter counts the received frames that are shorter than 64 bytes and have an incorrect CRC.

12.3.2.36 Undersized Frames Received

This 32-bit counter counts the received frames that are shorter than 64 bytes and have the correct CRC.

12.3.2.37 Jabber Frames Received

This 32-bit counter counts the received frames that are greater than the maximum byte size, and have an incorrect CRC.

12.3.2.38 No RxFIFO Missed Frames

This 32-bit counter counts the frames that are not able to be stored in the RxFIFO, which has overflowed.

12.3.2.39 No RxDescriptor Missed Frames

This 32-bit counter counts the frames that are not able to be stored in the RxFIFO since there is no receive Descriptor.

12.3.3 MIIM (Media Independent Interface Management)

12.3.3.1 MIIM Control Register

The MIIM control register allows the C790 to read and write any one of the PHYs connected to the MAC. This register provides bits to address a particular PHY, to address a register, to set the read or write direction, and to indicate that the read or write is still in progress. Upon the completion of reset, this register's value is 0x0000_0000.

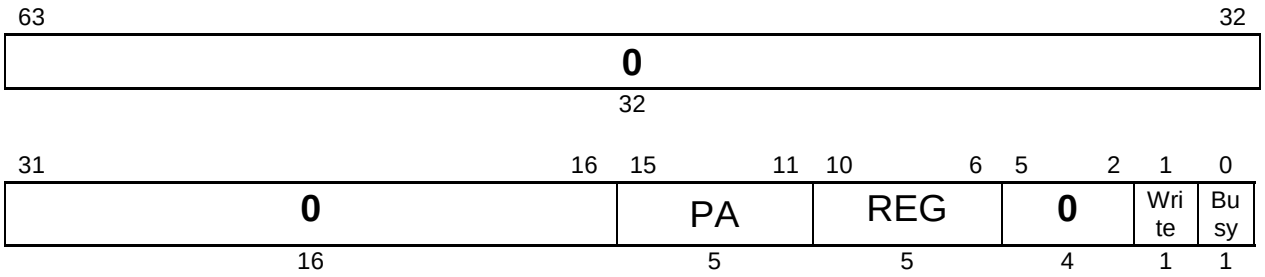


Figure 12-3 Fields of MIIM Control Register

Table 12-31 MIIM Control Register Field Descriptions

Bit(s)	Field	Description
15:11	PA[4:0]	PHY Address The 5-bit address of the PHY being read or written
10:6	REG[4:0]	Register Address The 5-bit address of the MIIM register being read or written
5:2	-	Reserved
1	Write	MIIM Write When set, indicates that this is a write operation on the PHY with the data in the MIIM data register
0	Busy (Read)	MIIM Busy When set, indicates a MIIM management read or write operation. If the Busy bit is set, writes to the MIIM Control or Data Register are ignored. The Busy bit should be 0 when writing to these registers. (Note : The MIIM data register must contain valid transmit data prior to writing to this MIIM Control Register)

12.3.3.2 MIIM Data Register

The MIIM data register is used in conjunction with the MIIM control register. When reading a PHY register, data are written to this register by the PHY. When writing to a PHY register, data from this register are written to the PHY. Upon the completion of reset, this register's value is 0x0000_0000.

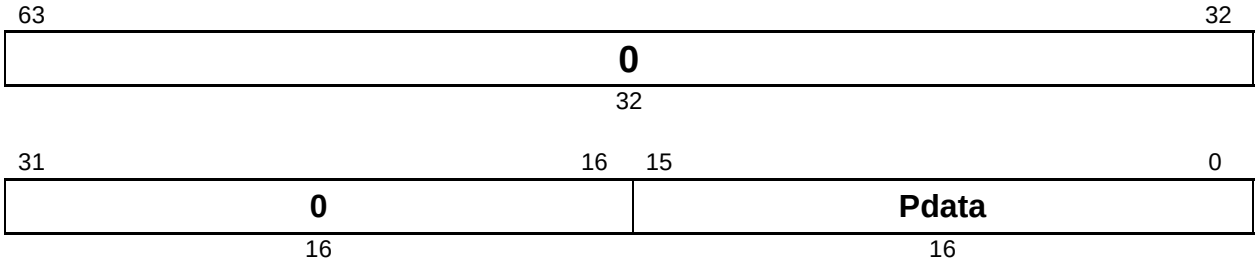


Figure 12-4 Fields of MIIM Data Register

Table 12-32 MIIM Data Register Field Descriptions

Bit(s)	Field	Description
15:0	Pdata [15:0]	PHY Data The 16-bit data read from the PHY after an MIIM read. The 16-bit data to be written to the PHY before an MIIM write. This register should not be written to if the Busy bit in the MIIM Control Register is 1

12.3.4 Address Filtering

The MAC supports 48-bit addresses in two separate address tables: the Perfect Table and the Hash Table. Only one of these tables is active at any time. Depending on the bit settings in RFCReg (see Table 12-9 and Table 12-10), the MAC could work in one of six address filtering modes.

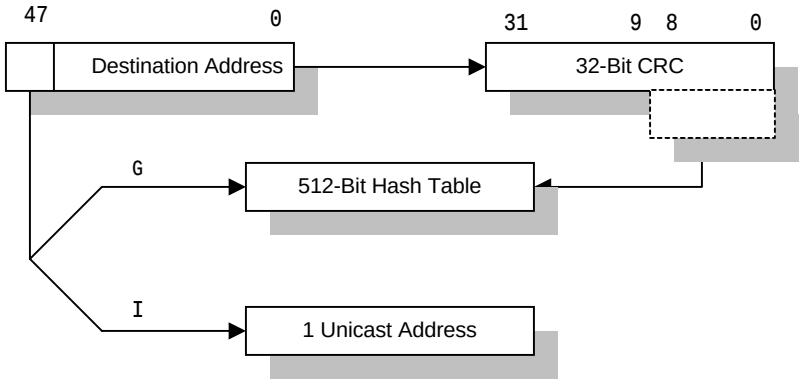


Figure 12-5 Imperfect Filtering of Incoming Frames

12.3.4.1 Perfect Table

The Perfect Table holds 16 destination addresses (full 48-bit MAC addresses). The MAC compares the addresses of any incoming frame to these addresses, and also checks the status of the Receive Frame Configuration Register. It rejects addresses that:

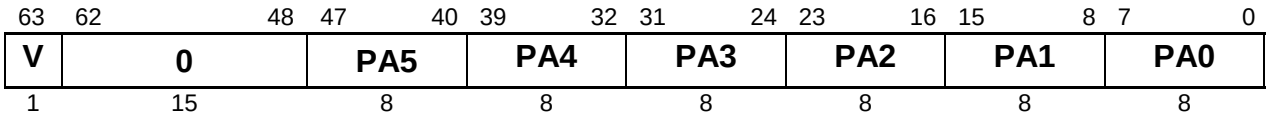
- Do not match if inverse filtering is reset.
- Match if inverse filtering is set.

This table should be filled with valid MAC addresses. Any mix of physical and multicast addresses can be used. Unused addresses should reset the PAValid bit, and reset bit 63 to zero. If none of the PAValid bits is set, the MAC works in the promiscuous mode. ***Note:*** These tables can only be written to when reception is disabled.

Addresses are contained in the lower 48 bits, as per the following table:

Table 12-33 Perfect Table Field Descriptions

Bit(s)	Field	Description
63	PAValid	When set, indicates that this physical address is valid. It is used for address filtering
62:48	–	Reserved
47:0	PA	Physical Address (PA5:PA4:PA3:PA2:PA1:PA0)



12.3.4.2 Hash Table

The Hash Table process is used in individual and group hash filtering. It stores 512 bits that serve as hash bucket heads, and one physical 48-bit MAC address. This process can be used in two different ways:

1. Incoming frames with multicast destination addresses are subjected to imperfect filtering. Frames with physical addresses are checked against the single physical address (see Figure 12-5.)

For any incoming frame with a multicast destination address, the MAC applies the standard Ethernet cyclic redundancy check (CRC) function to the first 6 bytes containing the destination address, then uses the most significant 9 bits of the result as a bit index in the table. If the indexed bit is set, the frame is accepted. If the bit is cleared, the frame is rejected.

2. All incoming frame destination addresses go to the Hash Table. The MAC uses the same method as above to get the indexed bit. If the indexed bit is set, the frame is accepted. If the bit is cleared, the frame is rejected.

12.3.5 FIFO Addresses

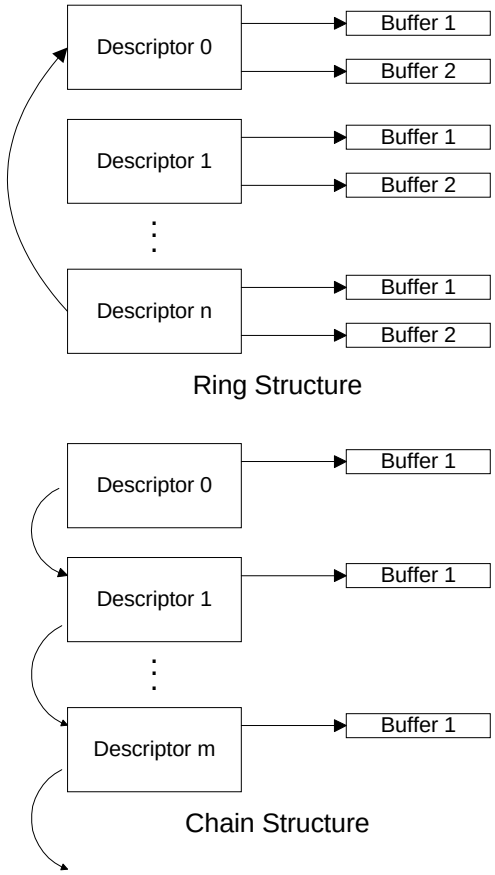
The FIFO address space is for debugging purposes. It allows the C790 to read or write to the FIFO. This may be done only in the FIFO diagnostic mode.

12.4 Memory Organization (Frame Descriptor)

This section describes the organization of the MAC Frame Descriptor in memory.

12.4.1 Descriptor Lists and Data Buffers

The MAC transfers frame data between the C790 memory and the FIFO using a Descriptor list. The Descriptors that reside in the host memory act as pointers to locate memory buffers.



The Descriptor may be implemented in two different ways: ring or chain.

Figure 12-6 Descriptor Ring and Chain Structure

There are two Descriptor lists, one for receive and one for transmit. The base address of each list is written to two registers: the Transmit Descriptor Pointer Register and the Receive Descriptor Pointer Register. The Descriptor lists reside in C790 memory. Each Descriptor can point to a maximum of two buffers. A data buffer consists of either an entire frame or part of a frame, but it cannot exceed one frame. Buffers contain only data. Buffer status is maintained in the Descriptor.

The last Descriptor can point back to the first entry when creating a ring structure.

The buffer address must be 8-byte aligned while the Descriptor address must be 16-byte aligned.

12.4.2 Receive Descriptors

The format of the Receive Descriptors is shown below. It is made of two double-words, and the fields are described in more detail in Table 12-34.

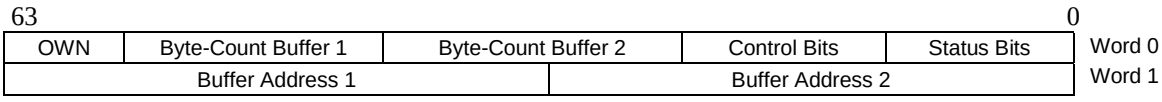


Figure 12-7 Receive Descriptor Format

12.4.2.1 Receive Descriptor

The Receive Descriptor contains the receive frame status, the frame length, and the Descriptor ownership information in the first word (Word 0). The 32 bits of two Receive Buffer Addresses is in the second word (Word 1). Bits [31:0] in Word 0 are read only. Bits [30:0] are valid only when RxEOF is set.

Table 12-34 Receive Descriptor Field Descriptions

Word 0 Fields		
Bit(s)	**Field**	**Description**
63	OWN	When set, indicates that the Descriptor is owned by the MAC. When reset, indicates that the Descriptor is owned by the C790. MAC clears this bit either when it completes the frame reception or when the buffers that are associated with this Descriptor are full.
62:59	-	Must be set to “0”.
58:48	Buffer1 Size	Indicates the size in bytes of the first data buffer. If this field is zero, the MAC ignores this buffer and only uses buffer 2. The buffer size should be a multiple of 8. If it is not, the least three bits are ignored.
47:37	Buffer2 Size	Indicates the size in bytes of the second data buffer. If RxChain is set (Address Chained), the MAC ignores this buffer and fetches the next Descriptor.
36	RxERing	Receive End of Ring When set, indicates that the Descriptor pointer has reached its final Descriptor, and the MAC returns to the base address of the list, creating a Descriptor ring.
35	RxChain	Second Address Chained When set, indicates that the second address in the Descriptor is the next Descriptor address rather than the second buffer address. If RxERing is set, then RxChain is ignored.
34	RxSOF	Start of Frame When set, indicates that this Descriptor contains the first byte of a frame.
33	RxEOF	End of Frame When set, indicates that this Descriptor contains the last byte of a frame. If this bit is not set, all frame status bits are invalid except the OWN bit.
32:31	–	Reserved
30:20	RxFrmLen	Receive Frame Length
19	RxNoCRC	Strip CRC
18	RxDType	Receive Data Type 0 : External frame 1 : Internal loop-back frame
17	RxFF	Receive Frame Filter Fail When set, indicates that the frame failed the address recognition filtering. This bit can be set only when the MAC is in the promiscuous mode (RxAll bit is set in RFCReg).
16:15	RxFrmType	Receive Frame Type

Word 0 Fields		
Bit(s)	**Field**	**Description**
	[1:0]	00 Ethernet type 01 IEEE type 10 VLAN I 11 VLAN II
14	RxBufErr	Descriptor Error When set, indicates that a frame truncation was caused by a frame that does not fit within the current Descriptor buffers, and that the MAC does not own the next Descriptor. The frame is truncated.
13	RxPStop	Receive Process Stopped When set, indicates that the reception process has been stopped.
12	RxNoBuf	Receive Buffer Unavailable When set, indicates that the next Descriptor in the reception list is owned by the host and cannot be acquired by the MAC. The reception process is suspended.
11	RxErr	Receive Error When set, indicates that the macxRxEr signal on the MII is asserted while macxRxDV is High. The presence of macxRxEr will also cause the CRC bit to be set and the Frames Received with Bad CRC counter to be incremented.
10	RxJab	Jabber on Reception When set, indicates that the frame received is greater than the maximum size with an incorrect CRC.
9	RxFAErr	Frame Alignment Error When set, indicates that the frame received has a frame alignment error. The CRC is invalid, and the byte count is greater than or equal to 64 bytes.
8	RxCRCErr	CRC Error When set, indicates that the frame received is greater than or equal to 64 bytes, and has a correct CRC.
7	RxUnds	Undersized Frame When set, indicates that the frame received is less than 64 bytes, and has a correct CRC.
6	RxFrag	Fragment When set, indicates that the frame received is less than 64 bytes, and has an incorrect CRC.
5	RxLongErr	Frame Long Error When set, indicates that the frame received is greater than the maximum size with a correct CRC.
4	RxPA	Physical Address When set, indicates that the destination address of the received frame is a physical address.
3	RxMA	Multicast Address When set, indicates that the destination address of the received frame is a multicast address.
2	RxBA	Broadcast Address When set, indicates that the destination address of the received frame is a broadcast address.
1	RxFOverf	FIFO Overflow When set, indicates that the frame is not able to be stored in the RxFIFO.
0	RxGood	Received a Good Frame When set, indicates that reception of a frame is complete. Specific frame status information has been posted in the Descriptor
Word 1 Fields		
Bit(s)	**Field**	**Description**
63:32	Buffer Address 1	This address points to buffer 1 locations. This address must be 8-byte aligned.
31:0	Buffer Address 2	This address points to buffer 2 locations. This address must be 8-byte aligned.

12.4.3 Transmit Descriptors

The format of the Transmit Descriptors is shown below. It is made of two doublewords, and the fields are described in more detail in Table 12-35.

OWN	Byte-Count Buffer 1	Byte-Count Buffer 2	Control Bits	Status Bits
Buffer Address 1		Buffer Address 2		

Figure 12-8 Transmit Descriptor Format

12.4.3.1 Transmit Descriptor

The Transmit Descriptor contains the transmit frame status, the frame length, and the Descriptor ownership information in the first word (Word 0). The 32 bits of two Transmit Buffer Addresses is in the second word (Word 1). Bits [31:0] in Word 0 are read only. Bits [30:0] are valid only when TxEOF is set and OWN is clear.

Table 12-35 Transmit Descriptor Field Descriptions

Word 0 Fields		
Bit(s)	**Field**	**Description**
63	OWN	When set, indicates that the Descriptor is owned by the MAC. When reset, indicates that the Descriptor is owned by the C790. MAC clears this bit either when it completes the frame transmission or when the buffers that are allocated in the Descriptor are empty. The Ownership bit of the first Descriptor of the frame should be set after all subsequent Descriptors belonging to the same frame have been set. This avoids a possible race condition between the MAC fetching a Descriptor and the driver setting an ownership bit.
62:59	-	Must be set to “0”.
58:48	Buffer1 Size (2K)	Indicates the size, in bytes, of the first data buffer. If this field is 0, the MAC ignores the buffer and uses buffer2.
47:37	Buffer2 Size (2K)	Indicates the size, in bytes, of the second data buffer. If TxChain is set (Address Chained), the MAC ignores this buffer and fetches the next Descriptor.
36	TxERing	Transmit End of Ring When set, indicates that the Descriptor pointer has reached its final Descriptor. The MAC returns to the base address of the list, creating a Descriptor ring.
35	TxChain	Second Address Chained When set, indicates that the second address in the Descriptor is the next Descriptor address, rather than the second buffer address. If TxERing is set, TxChain is ignored.
34	TxSOF	Start of Frame When set, indicates that the buffer contains the first byte of a frame.
33	TxEOF	End of Frame When set, indicates that the buffer contains the last byte of a frame.
32	TxEnCRCDes	CRC Enable When TxDesSelEn is set, this bit overrides the TxEnCRC bit in the TFCReg.
31	–	Reserved
30:20	TxFrmLen[10:0]	Transmission Frame Length
19:15	–	Reserved
14:13	TxFrmType[1:0]	Transmit Frame Type 00 Ethernet 01 IEEE 10 VLAN I 11 VLAN II
12	TxExDefer	Excessive Deferral When set, indicates that the transmission was aborted because of an excessive deferral as defined by the defer bit in the transmit frame configuration register. This bit is not valid for

Word 0 Fields		
Bit(s)	**Field**	**Description**
		full-duplex.
11	TxDefer	Deferred When set, indicates the frame transmission was delayed because of a deferral. This bit is set when a frame is transmitted with a collision and the standard backoff is selected in the configuration register. This bit is not valid while the port is configured for full-duplex.
10	TxSCol	Single Collision When set, indicates that the frame being transmitted collided only once and was then transmitted successfully on the Ethernet.
9	TxMCol	Multiple Collisions When set, indicates that the frame being transmitted collided more than once and was then transmitted successfully on the Ethernet.
8	TxExCol	Collision Error When set, indicates that the frame transmission was aborted because of too many collisions. The number of collision retries allowed is specified in the transmit frame configuration register.
7	TxLCol	Late Collision When set, indicates that a transmission is aborted due to a collision occurring later than 512 bit times.
6	TxLCar	Loss of Carrier When set, indicates the CRS input is Low during the transmission of a frame.
5	SQE	Signal Quality Error Missed When set, indicates that the SQE test on the macxCOL signal line is not detected at the end of a transmission.
4	TxUndf	Transmit Underflow When set, indicates that the TxFIFO had an underflow condition during the frame transmission. The transmission process is suspended.
3	–	Reserved
2	TxNoBuf	Transmit Buffer Unavailable When set, indicates that the next Descriptor in the transmit list is owned by the host and cannot be acquired by the MAC. The transmission process is suspended.
1	TxPStop	Transmit Process Stopped When set, indicates that the transmit process has stopped.
0	TxGOOD	Good Frame When set, indicates that a frame transmission was completed without error. This bit will be set regardless of the state of SQE, TxSCol, TxMCol and TxDefer. This bit will not be set when TxLCol, TxExDefer, TxExCol or TxUndf

Word 1 Fields		
Bit(s)	**Field**	**Description**
63:32	Buffer Address 1	This address points to buffer 1 locations. This address must be 8-byte aligned.
31:0	Buffer Address 2	This address points to buffer 2 locations. This address must be 8-byte aligned.

12.5 Functional Description

12.5.1 DMA

The MAC provides a master DMA interface that is capable of reading or writing data at high speed. When the MAC wants to transfer data to/from memory, it follows the 64-bit G-Bus conventions. All internal control signals are synchronous to the gbsBusClk. Each frame, whether transmit or receive, must be delineated by two signals, SOF (Start of Frame) and EOF(End of Frame).

12.5.1.1 DMA Arbitration

The DMA arbitration scheme depends on the BAR bit setting in the CCReg. When it is reset, the MAC grants precedence to the receive process instead of the transmit process. When BAR is set or the MAC is in the full-duplex mode, a round-robin arbitration scheme is applied.

There are 4 different DMA requests.

The DMA requests possible from the transmit process are:

1. TxDesReq : This is used for DMA to fetch or to close a Descriptor
2. TxFrmReq : This is used for DMA to transfer data from memory to a TxFIFO

The DMA requests possible from the receive process are:

1. RxDesReq : This is used for DMA to fetch or to close a Descriptor
2. RxFrmReq : This is used for DMA to transfer data to memory from a RxFIFO.

12.5.1.2 Transmit

In the running state, the transmit process polls the transmit Descriptor list for frames requiring transmission. After polling starts, it continues in either the sequential Descriptor ring order or in the chained order. When frame transmission is complete, status information is written into the Transmit Descriptor and the Descriptor is closed.

If either the MAC detects a Descriptor flagged as owned by the C790 or an error condition occurs, the transmit process is suspended and an interrupt bit is asserted.

While in the running state, the transmit process can simultaneously acquire two frames. As the transmit process completes copying the first frame, it immediately polls the transmit Descriptor list for the second frame. If the second frame is valid, the transmit process copies the frame without waiting for the status information of the first frame.

12.5.1.2.1 Transmit Frame Process

Frames can be data-chained and span several buffers. Frames must be delimited by SOF (Start of Frame) and EOF (End of Frame) in the frame Descriptor.

SOF must be set before the transmit process is initiated. When this occurs, frame data are transferred from the host buffer to the TxFIFO. Concurrently, if the current frame has the

EOF cleared, it indicates an intermediary buffer, and the transmit process attempts to acquire the next Descriptor. If the EOF is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted to MII, the MAC writes back the final status information to the transmit Descriptor. At this time, if any interrupt condition is met, the transmit interrupt is set, the next Descriptor is fetched, and the process repeats.

Actual frame transmission begins after the TxFIFO has reached either a programmable threshold or a full frame is contained in the TxFIFO.

12.5.1.2.2 Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The MAC detects a Descriptor owned by the C790.
- A frame transmission is aborted when a locally induced error or G-Bus error is detected.

If either of the previous two conditions occur, an interrupt is set and the information is written to the frame Descriptor, causing the suspension.

The MAC does not automatically poll the transmit Descriptor list; the driver must issue a transmit poll demand command after rectifying the cause of the suspension.

12.5.1.2.3 Transmit Frame DMA request

The TxFIFO uses a two-signal handshake for request and acknowledge synchronization. The first signal TxFrmReq is active when the TxFIFO is capable of having data written to it. The amount of data is a count threshold (PBL) in the CCReg (Command and Configuration Register). If the PBL is equal to 001 and there are at least 32 bytes empty in the TxFIFO.

The second signal, TxFrmRdy, is active to start a burst of the number of bytes represented by PBL.

12.5.1.3 Receive

In the running state, the reception process polls the reception Descriptor list in the memory. Incoming frames are processed and placed in acquired Descriptor data buffers. Status information is written to the last received Descriptor of the frame.

12.5.1.3.1 Poll Descriptor

The MAC always attempts to acquire an extra Descriptor in anticipation of incoming frames. Descriptor acquisition is attempted if any of the following conditions are satisfied:

- When Start Receive gets set in RFCReg immediately after being placed in the running state.
- When the MAC begins writing frame data to a data buffer, and the buffer ends

before the frame ends.

- When the MAC completes the reception of a frame and the current received Descriptor has been closed

- When the receive process is suspended because of a host-owned buffer, and a new frame is received.

12.5.1.3.2 Receive Frame Processing

As incoming frames arrive, the MAC recovers the incoming data and clock pulses, and then sends them to the reception engine. The reception engine strips the preamble bits and stores the frame data in the RxFIFO. Concurrently, the reception section performs address filtering depending on the perfect table or hash table. If the frame fails the address filtering, it is ignored and flushed from the RxFIFO. Frames that are shorter than 64 bytes, due to collision or premature termination, are also ignored and flushed from the RxFIFO.

The RxFIFO requires a similar handshake algorithm to that of the TxFIFO. When a new frame is received, the first RxFrmReq is controlled by the reception frame threshold (RxSOFTh). This RxSOFTh is set in the RFCReg (Receive Frame Configuration Register), and determines the minimum frame required in the RxFIFO to initiate the DMA operation. Once this threshold is met, the entire frame needs to be read out of the RxFIFO, regardless of any error conditions and bits set in the RFCReg. If the MAC detects any errors before the threshold is met, the frame will be discarded depending on the state of the appropriate configuration bit in the RFCReg.

Once the threshold is met, each successive assertion of RxDMAReq is based on the PBL, which may or may not be the same as the threshold.

If a frame ends before either PBL or threshold RxSOFTh is met, RxDMAReq will automatically be asserted, regardless the PBL or the threshold.

12.5.1.3.3 Receive Frame Processing Suspended

If a receive frame arrives while the reception process is suspended, the MAC re-fetches the current Descriptor in memory. If the Descriptor is now owned by the MAC, the reception process reverts back to the running state and starts the frame reception. If the Descriptor is still owned by the host, the MAC discards the current frame in the reception FIFO and increments the Missed Frames counter.

12.5.2 FIFO Operation

FIFOs are used internally to buffer frames before they are transmitted on the network or before they are put to memory. The TxFIFO (1 KB) is deep enough to support the retransmission of a frame if a collision occurs within the first 512 bit times of transmission. The RxFIFO (1 KB) is deep enough to filter undersized and fragmented frames without having to interrupt the host.

12.5.2.1 Threshold

Both the TxFIFO and RxFIFO have their own thresholds. The number is between 256 and 0. But it is highly recommended to use a multiple of 16, such as 32, 64, 128 for the byte size. DMA could take the burst transaction instead of the single transaction.

TxSOFTh and PBL are used to monitor the TxFIFO operation. TxSOFTh is used to start a transmission after this number of data is written to the TxFIFO. PBL is used to monitor how many free spaces are left in the TxFIFO. If there is no such free space, TxFrmReq will be placed on hold. Therefore, there is no overflow in the TxFIFO.

In general, TxSOFTh should be more than 64 bytes, multiple of PBL and meet

(TxSOFTh + PBL) ≤ TxFIFO

(TxFIFO - PBL) > PBL

RxSOFTh and PBL are used to monitor the RxFIFO operation. RxSOFTh specifies when the first DMA request is asserted after a frame is received. PBL is used to warn the MAC that only this number of free spaces is left in the RxFIFO. For each frame, after reaching RxSOFTh, subsequent DMA requests are governed by the PBL. This allows the first burst of the frame to be different than the typical burst capability of the host. If there is not enough data to be transferred, RxFrmReq will be placed on hold. This avoids underflowing in the RxFIFO.

In general, the more RxSOFTh, the better. It should be more than 64 bytes, a multiple of PBL and meet the following condition:

(RxFIFO - PBL) > PBL

12.5.2.2 Loopback

MAC supports two kinds of internal loop back operations for diagnostic purposes.

FIFO Level Loop Back

Data are written to the TxFIFO and transmitted to the RxFIFO internally. No data are sent across the MII. The three threshold values are maintained in addition to the Tx and Rx FIFO handshake signals. This loop back allows the host to perform a DMA in order to executed a FIFO self-check. Status bits and event counters are not updated.

MII Level Loop Back

This time the loop goes deeper than the previous case, including the MAC block. Status bits and counters are all active.

To do a loop back test, the MAC should be set in the full-duplex mode.

12.5.2.3 TxFIFO Specific Function

The TxFIFO provides the mechanism for sending frame data through the MAC and onto the network. FIFO input is controlled by the DMA.

Transmission of a frame starts automatically either when a certain number of words is written to the TxFIFO or when EOF is written. This threshold (TxSOFTh) is in the TFCReg.

If the FIFO runs out of data before EOF is detected, the partial frame is transmitted with an incorrect CRC, regardless of the state of the CRC bit in the transmit frame configuration register. If an Underflow occurs before 64 bytes have been transmitted, a fragment is transmitted. When an Underflow occurs while writing data to TxFIFO, the host must still delineate the frame by asserting EOF. Data written to the TxFIFO after the Underflow and before EOF will be ignored.

12.5.2.3.1 Collision Retransmission

The TxFIFO does not overwrite the first 64 bytes of a frame until they are transmitted successfully. In the event of a collision, the FIFO pointers are reset to the beginning of the frame. These bytes are retransmitted after a collision back-off period has elapsed.

12.5.2.3.2 Half-Duplex Flow Control

While in the half-duplex mode, the host system can have the transmitter force a collision by asserting TxEn every time a frame is received. MAC will transmit the preamble and 32 bits of jam bytes then release TxEn. This allows all stations on the network to see the collision and backoff.

MAC can also be configured to assert flow control when the RxFIFO overflows.

12.5.2.3.3 Full-Duplex Flow Control

If the RxFCEn bit in the reception frame configuration register is set, the reception FIFO overflows, and the port is configured for full-duplex, the MAC will generate then transmit a pause frame. Only one pause frame is sent when the FIFO overflows.

The MAC also responds to received flow control frames.

12.5.2.3.4 CRC Generation

The MAC has the capability of calculating and appending the cyclic redundancy check (CRC) to each transmitted frame. This is selectable by the bit in the TFCReg or Descriptor.

If the CRC bit is set, every frame has the CRC calculated and appended. Otherwise, no CRC is calculated and appended.

12.5.2.3.5 Error Conditions

The TxFIFO has the capability of stopping its operation when an error occurs. This option can be enabled by setting the TxEnHalt bit in the transmit frame configuration register. The following conditions cause the TxFIFO to stop: excessive deferral, excessive collisions, late collision, or TxFIFO overrun.

When the FIFO is halted, the TxStart bit in the TFCReg is cleared and TxFrmReq is deasserted until the condition is cleared by setting the TxStart bit in the TFCReg.

12.5.2.3.6 Enabling

The TxFIFO must be enabled before transmission starts, but after a Stop condition. This is done by setting the Enable bit in the TFCReg.

12.5.2.3.7 Short Frame Padding

The MAC can transmit a frame of any length. If the PAD bit and CRC bit in the TFCReg are set, MAC will automatically pad short frames to 64 bytes in length, and include CRC.

12.5.2.3.8 Multiple Frames

The TxFIFO is capable of holding multiple frames. Frame boundaries in the FIFO are recognized by the SOF and EOF. If there is no SOF before EOF, the data are ignored.

12.5.2.4 RxFIFO Specific Functions

The RxFIFO buffers frames as they are received from the network.

12.5.2.4.1 Multiple Frames

The RxFIFO has the ability to hold multiple frames at any given time. If several short frames are received before the MAC gets the G-Bus, then all of those frames will be stored.

12.5.2.4.2 Start and End of Frame

The MAC will mark the SOF and EOF in the frame Descriptor after the frame data have been read from the RxFIFO. The frame byte length is 16 bits. If a frame is larger than 64 KB, the length is represented as FFFFH.

12.5.2.4.3 Stripping CRC from Receive Frames

The RxNoCRC bit in the RFCReg determines whether or not the CRC is removed from each incoming frame. If RxNoCRC is low, the CRC is not stripped.

12.5.2.4.4 Receive Statistics

The status bits are recorded in two places as frames are received. First, a Receive Diagnostic Register (RDReg) is updated at the end of every receive frame. It is not practical for the system to read this register after each frame is read. Instead, the contents of this

register are put in the frame Descriptor after each frame datum has been transferred into memory.

Second, event counters are updated based on the status at the end of a receive frame. These counters can be read at any time.

12.5.2.4.5 Undersized and Fragment Rejection

A frame is received that is less than 64 bytes and has a good CRC (undersized), has a bad CRC, or frame alignment error (fragment). This kind of frame can be filtered if RxSOFTh is set to 64 bytes or more.

12.5.2.4.6 Error Conditions

Error conditions can be found in the Frame Interrupt Register.

12.5.2.4.7 Overflow

When the RxFIFO overflows, the interrupt bit RFIFO is set. The receiver will ignore all subsequent data from that frame. If the flow control enable bit (RxFCEn) is set, the MAC will send out a flow control frame while configured for full-duplex. It will force a collision each time CRS is asserted while configured for half-duplex.

12.5.2.4.8 Enabling

The port must be enabled before the MAC receives any frame. If the port is disabled by clearing the enable bit in RFCReg and a frame is currently being received, the port is disabled after the frame is fully received.

12.5.3 MII Interface

The MII interface is described in this section.

12.5.3.1 macxTxClk

TXCLK is the transmit clock used to provide the timing reference for the transfer of the macxTxEn, macxTxD[3:0] and macTxEr to the PHY. The MAC handles internal synchronization between gbsBusClk and these transmit signals.

12.5.3.2 macxRxClk

macxRxClk is the receive clock used to provide the timing reference for the transfer of the macxRxDv, macxRxD[3:0] and macxRxEr from the PHY. The MAC handles the synchronization between gbsBusClk and these receive signals.

12.5.3.3 MII Management Interface

The MAC has two 16-bit registers that are used to read and write the physical layer device. To initiate a read of one of the MII registers, the user must write to the MII control register. The MII control register requires a valid PHY address, a valid opcode and a valid register address. When this register is written, MAC will initiate the read by diving MDC with a valid clock and MDIO with the proper preamble, start code, and data from the MII control register. The Busy bit in the MII control register is set for the duration of the operation. Data read from the PHY are stored in the MII data register and are available for the host to read when the Busy bit is zero. Writing to an MII register is similar. The host must never write to the MII data register or to the MII control register while the Busy bit is set.

12.5.4 Interrupt

The MAC supports two kinds of interrupt event. One is the frame interrupt FRMINT, and the other is the counter interrupt CNTINT. FRMINT is used mainly for notifying the CPU that a transmission or reception error has occurred which may cause the port to stop operating correctly. CNTINT is active when a counter overflows. CNTINT and FRMINT both share one interrupt signal macgIntB.

12.5.5 Reset

There are two kinds of reset: Hardware Reset and Software Reset.

12.5.5.1 Hardware Reset

When SysResetB is driven Low:

- Registers return to their default values.
- State machines return to their idle state.
- Counters are reset to zero.

- FIFO control is returned to its idle configuration.

Any frame being transmitted or received at the time of reset is lost. To clear all counters gresetB should keep at least 40 gbsBusClk.

12.5.5.2 Software Reset

There are four different reset bits in the CCReg (Command and Configuration Register).

1. Software reset
2. Counter reset
3. Transmit reset
4. Receive reset

The transmit and receive reset bits reset the transmit and receive FIFO as well as the other logic associated with the data path. These resets do not have any effect on the counter values, although during transmission or reception reset, the associated counters are not updated. The counter reset will reset the event counters only. The software reset bit resets the transmit and receive data paths and the MII management logic. It will not reset the event counters and registers. The MAC clears all of the reset bits when the reset is complete.

13. Removed

14. UARTS WITH FIFOS

14.1 Overview

The TX7901 has two individual UARTs (hereinafter referred to as simply “The UARTs”), high-performance universal asynchronous receiver/transmitters each having two 16-byte FIFOs – one for transmit and one for receive. Each of the UARTs also includes a 16-byte programmable baud rate generator, an 8-bit scratch register, and eight modem control lines.

The UARTs are fully programmable through their G-Bus interface. They support word lengths from five to eight bits, an optional parity bit and one or two stop bits. If enabled, the parity can be odd, even, or forced to a defined state. Interrupts can be generated from any of ten sources.

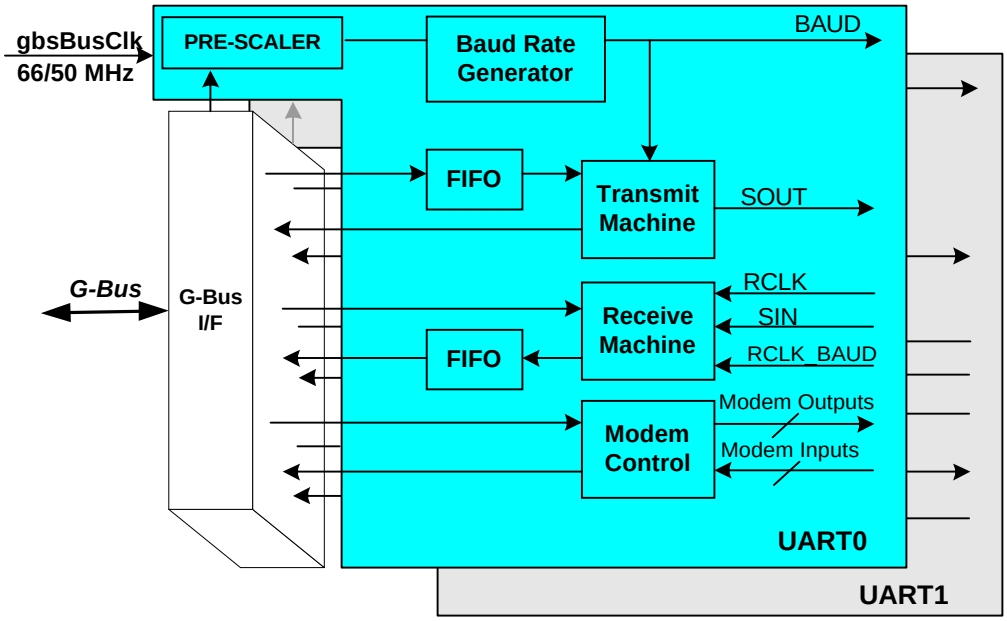


Figure 14-1 UART Block Diagram

Note: If RCLK_BAUD is tied high, RCLK is connected to BAUD; if RCLK_BAUD is tied low, RCLK is taken from an external Receive clock (fed in through the RCLK pin.) If an external Receive clock is used, the Receive clock period must be more than four times the pre-scaled clock period (i.e. more than 480 ns/2.08 MHz).

14.1.1 Key Features

- Software-compatible with NSC NS16550A
- Programmable word length, stop bits, and parity
- Programmable baud rate generator
- Interrupt generator
- Diagnostic loop-back mode
- Scratch register
- Two 16-byte FIFOs
- Scan test ready

14.1.2 Introduction

The UARTs are universal asynchronous receiver/transmitters that are fully programmable through the G-Bus Interface. They support word lengths of five to eight bits, an optional parity bit, and one or two stop bits. If enabled, the parity can be odd, even or forced to a defined state.

A 16-bit programmable baud rate generator and an 8-bit scratch register are included in each UART, together with two 16-byte FIFOs: one for transmit and one for receive. Eight modem control lines and a diagnostic loop-back mode are provided in each UART.

An interrupt can be generated from any one of 10 sources.

Note: The UART has been designed so that most internal operations are synchronized by the gbsBusClk signal. This results in minor timing differences between the UART and the original device, which means that the core is not clock-for-clock identical with the original device.

14.2 Functional Description

14.2.1 Transmit Operation

Transmission is initiated by writing the data to be sent to the TX Holding Register. The data will then be transferred to the TX Shift Register together with the start bit, parity and stop bits as determined by the Line Control Register. The bits to be transmitted are then shifted out of the TX Shift Register using the output from the Baud Rate Generator as the clock.

If enabled, an interrupt will be generated when the TX Holding Register or TxFIFO becomes empty.

When FIFOs are enabled, (i.e. Bit 0 of the FIFO Control Register is set), each UART can store up to 16 bytes of data at a time for transmission. Transmission will continue until the particular Transmit FIFO is empty. The FIFO's readiness to accept more data is indicated by TXRDY or, if the transfer is interrupt driven, by an interrupt request (IRQ).

14.2.2 Receive Operation

Data are sampled into the RX Shift Register using RCLK. A filter is used to remove spurious inputs that last for less than two clock periods.

When the complete word has been clocked into the receiver, the data bits are transferred to the RX Buffer Register or to the RX FIFO (if enabled) to be read by the G-Bus. The receiver also checks for a stop bit and for correct parity as determined by the Line Control Register.

If enabled, an interrupt will be generated when the data have been transferred to the RX Buffer Register. Interrupts can also be generated when there is incorrect parity or a missing stop bit (frame error).

When the FIFOs are enabled, (i.e. Bit 0 of the FIFO Control Register is set), the UART can store up to 16 bytes of received data at a time. Depending on the selected mode, either RXRDY or IRQ will go active when the Receive FIFO contains 1, 4, 8, or 14 bytes of data – to indicate that data are available.

14.2.3 Modem Control Lines

The output Modem Control lines RTS*, DTR*, OUT1*, and OUT2* can be set or cleared by writing to the Modem Control Register.

The current status of the input Modem Control Lines DCD*, RI*, DSR*, and CTS* can be read from the Modem Status Register. Bit 2 of this register will be set if the NRI line has changed from Low to High since the register was last read.

If enabled, an interrupt will be generated when any of the DSR*, CTS*, RI*, or DCD* signals is asserted.

Note that there are no modem control lines for UART1 due to pin constraints. UART1 is assumed to use software flow control.

14.3 Interface Signals

Table 14-1 lists the various interface signals for the UARTs.

Table 14-1 Serial I/O Signal Descriptions

SIGNAL	TYPE	DESCRIPTION
G-Bus Interface		
gbsgBusClk	Input	G-Bus Clock
sysResetB	Input	G-Bus Reset
gbsgAddr[31:2]	Input	G-Bus Address Bus
gbsgData[63:0]	Input	G-Bus Data Bus
gcbgURTRegCSB	Input	Chip select signal which is generated by the G-Bus Bridge for the UART
gbsgBEB[7:0]	Input	G-Bus byte enable
gbsgBSize[2:0]	Input	G-Bus burst transfer size
gbsgRdB	Input	G-Bus read signal, active Low
gbsgWrB	Input	G-Bus write signal, active Low
gbsgBurstB	Input	G-Bus Burst Transaction
gbsgBStartB	Input	G-Bus transaction start. The G-Bus Master asserts this signal to indicate the start of a transaction.

SIGNAL	TYPE	DESCRIPTION
gbsgLastB	Input	The G-Bus Master asserts this signal to indicate the last transaction.
urtgAck32B	Output	G-Bus Acknowledge. UART asserts this signal to acknowledge a 32-bit width read/write transfer.
urtgData[63:0]	Output	64-bit read data from UART to G-Bus
Serial Interface (Two sets)		
BAUD	Output	Receive/Transmit clock, derived from CLK. Divided by the value in the divisor latch DLL & DMM.
RCLK	Input	Receive Clock
RCLK_BAUD	Input	RCLK Select. When tied hHigh, RCLK is connected internally to BAUD; when tied Low, the RCLK pin is used as the Receive clock.
SIN	Input	Serial Input. Data are clocked in using RCLK/16.
SOUT	Output	Serial Output. Data are clocked out using the output from the Baud Rate Generator, then divided by 16.
DCD*	Input	Data Carrier Detect, MSR[7] status bit. Active Low
RI*	Input	Ring Indicator, MSR[6] status bit. Active Low
DSR*	Input	Data Set Ready, MSR[5] status bit. Active Low.
CTS*	Input	Clear To Send, MSR[4] status bit. Active Low
OUT2*	Output	General Control, MSR[3] control bit. Active Low
OUT1*	Output	General Control, MSR[2] control bit. Active Low
RTS*	Output	Request To Send, MSR[1] control bit. Active Low
DTR*	Output	Data Terminal Ready, MSR[0] control bit. Active Low
DMAC Interface		
urtgRxRdyB	Output	DMA Handshake. Goes Low when RX FIFO contains data.
urtgTxRdyB	Output	DMA Handshake. Goes Low when TX FIFO is empty.
G-Bus Bridge Interface		
urtgIntB	Output	Interrupt Request. Goes Low whenever one of the enabled interrupts becomes valid. This signal goes to the interrupt controller inside the G-Bus Bridge.

Note: Active Low external signals are indicated by a suffix of "*" (asterisk), or "B" (Capital B) for Active Low internal TX7901 signals.

14.4 UART Device Register Description

This section describes the block of device registers that control each UART. There are two of each of the device registers described in this section, one for each of the UARTs. Each type of register is located at the same offset position in both UART channels' blocks of device registers.

14.4.1 UART Device Register Addressing

Most of the UART device registers are directly addressed through the address lines. One of the following four register types is accessed: the Divisor Latch Registers, the Receive Buffer Register, the Transmit Holding Register, or the Interrupt Enable Register. The register to be accessed is selected according to the following: the setting of bit 7 (the Divisor Latch Address Bit, DLAB) of the respective Line Control Register, and whether one is reading from or writing to the accessed register. (Please refer to Table 14-2 and Table 14-3).

Table 14-2 Device Register Addressing for UARTs 0 & 1: Little Endian Mode

	G-Bus		DLAB	Bank Offset	Register		Notes
Offset	Address [7:2]	Byte Enables			Acronym	Name	
00	0000_00	1110	0	0	RBR	Receive Buffer Register	R/O. See 14.4.2.
00	0000_00	1110	0	0	THR	Transmit Holding Register	W/O. See 14.4.3.
00	0000_00	1110	1	0	DLL	Divisor Latch (LS)	R/W. See 14.4.13.
04	0000_01	1110	0	1	IER	Interrupt Enable Register	R/W. See 14.4.8.
04	0000_01	1110	1	1	DLM	Divisor Latch (MS)	R/W. See 14.4.13.
08	0000_10	1110	X	2	IIR	Interrupt ID Register	R/O. See 14.4.7.
08	0000_10	1110	X	2	FCR	FIFO Control Register	W/O. See 14.4.6.
0C	0000_11	1110	X	3	LCR	Line Control Register	R/W. See 14.4.4.
10	0001_00	1110	X	4	MCR	Modem Control Register	R/W. See 14.4.9.
14	0001_01	1110	X	5	LSR	Line Status Register	R/O. See 14.4.5.
18	0001_10	1110	X	6	MSR	Modem Status Register	R/W. See 14.4.10.
1C	0001_11	1110	X	7	SCR	Scratch Register	R/W. See 14.4.11.
20	0010_00	1110	X	8	PSR	Pre-scaler Register	R/W. See 14.4.12

Note:
*1 X = don't care, either 0 or 1, R/O = Read Only, W/O = Write Only.
*2 There are no RXFIFO or TXFIFO registers.
*3 The base address for UART0 is 0x1E00_7000. The base address for UART1 is 0x1E00_8000.

Table 14-3 Device Register Addressing for UARTs 0 & 1: Big Endian Mode

	G-Bus		DLAB	Bank Offset	Register		Notes
Offset	Address [7:2]	Byte Enables			Acronym	Name	
00	0000_00	0111	0	0	RBR	Receive Buffer Register	R/O. See 14.4.2.
00	0000_00	0111	0	0	THR	Transmit Holding Register	W/O. See 14.4.3.
00	0000_00	0111	1	0	DLL	Divisor Latch (LS)	R/W. See 14.4.13.
04	0000_01	0111	0	1	IER	Interrupt Enable Register	R/W. See 14.4.8.
04	0000_01	0111	1	1	DLM	Divisor Latch (MS)	R/W. See 14.4.13.
08	0000_10	0111	X	2	IIR	Interrupt ID Register	R/O. See 14.4.7.
08	0000_10	0111	X	2	FCR	FIFO Control Register	W/O. See 14.4.6.
0C	0000_11	0111	X	3	LCR	Line Control Register	R/W. See 14.4.4.
10	0001_00	0111	X	4	MCR	Modem Control Register	R/W. See 14.4.9.
14	0001_01	0111	X	5	LSR	Line Status Register	R/O. See 14.4.5.
18	0001_10	0111	X	6	MSR	Modem Status Register	R/W. See 14.4.10.
1C	0001_11	0111	X	7	SCR	Scratch Register	R/W. See 14.4.11.
20	0010_00	0111	X	8	PSR	Pre-scaler Register	R/W. See 14.4.12

Notes:
*1 X = don't care, either 0 or 1, R/O = Read Only, W/O = Write Only.
*2 There are no RXFIFO or TXFIFO registers.
*3 The base address for UART0 is 0x1E00_7000. The base address for UART1 is 0x1E00_8000.

14.4.2 Receive Buffer Register (RBR0, RBR1)

This register is updated from the RX Shift Register at the end of a receive sequence. If the FIFOs are disabled, this register is undefined after reset. If the FIFOs are enabled, this register will return "0" after reset if the RX FIFO is empty. This register receives data from the FIFOs when the FIFO mode is enabled.

14.4.3 Transmit Holding Registers (THR0, THR1)

Data are held in this register until transferred to the TX Shift Register when in a non-FIFO mode. Data are sent to FIFOs when the FIFO mode is enabled.

14.4.4 Line Control Registers (LCR0, LCR1)

Table 14-4 lists the fields of the Line Control Registers. This register specifies line control parameters and contains the DLAB bit which makes the Divisor Latch addresses accessible.

Table 14-4 Line Control Register Field Descriptions

Bit	Read/ Write	Comments
0	WLS0	Word Length Select
1	WLS1	Word Length Select
2	STB	Number of Stop Bits
3	PEN	Parity Enabled
4	EPS	Even Parity Select
5	SP	Stick Parity
6	SB	Set Break
7	DLAB	**Divisor Latch Access Bit**. Controls access to alternate registers at Addresses 0 and 1

After reset, bits 0 through 7 are all zero.

14.4.4.1 WLS0, 1 – Word Length Select

Transmitted and Received character size is defined as follows:

Table 14-5 Transmit and Receive Character Size

WLS1	WLS0	Character Size
0	0	5-bit
0	1	6-bit
1	0	7-bit
1	1	8-bit

14.4.4.2 STB – Number of stop bits

When set (“1”), two STOP bits are added after each character is sent, unless the character length is 5 when 1½ STOP bits are added. When cleared (“0”), one STOP bit is always added.

Note: Only the Transmit STOP bits are programmable. The Receive stage only expects one STOP bit, irrespective of the value of STB.

14.4.4.3 PEN – Parity Enabled

When set (“1”), the parity is transmitted and checked. The Parity bit is added after the data field and before the STOP bits.

When cleared (“0”), the parity is neither transmitted nor checked.

14.4.4.4 EPS – Even Parity Select

When EPS = “1” and PEN = “1”, an even number of ones is sent and checked. When EPS = “0” and PEN = “1”, an odd number of ones is sent and checked.

14.4.4.5 SP – Stick Parity

When set (“1”), the Parity bit is forced into a defined state, dependent upon the values of EPS and PEN:

If EPS = “1” & PEN = “1”, the Parity bit is transmitted and checked as “0”.

If EPS = “0” & PEN = “1”, the Parity bit is transmitted and checked as “1”.

14.4.4.6 SB – Set Break

When set (“1”), a break condition is transmitted to the receiving UART, and the serial output (SOUT) is forced to “0.”

14.4.4.7 DLAB – Divisor Latch Address Bit

Controls access to alternate registers at addresses 0 and 1.

When cleared (“0”), the RX and TX registers, and the IER register are read/written to.

When set (“1”), LS and MS of the Divisor Latch are read/written to.

14.4.5 Line Status Registers (LSR0,LSR1)

Table 14-6 lists the fields of the Line Status Registers.

Table 14-6 Line Status Register Fields

Bit	Read	Comment
0	DR	Data Ready
1	OE	Overrun Error
2	PE	Parity Error
3	FE	Framing Error
4	BI	Break Interrupt
5	THRE	TX Holding Register Empty
6	TEMT	Transmitter Empty
7	FIFOERR	RX Data Error in FIFO

After reset, bits 0 to 4 and bit 7 are all "0," while bits 5 and 6 are set to "1."

This register is read only.

Note: This register is reset asynchronously.

14.4.5.1 DR – Data Ready

This bit is set either by the RX Buffer becoming full or by a byte being transferred into the FIFO. It is cleared by the G-Bus reading the RX Buffer or by reading all of the FIFO bytes.

This bit is also cleared whenever the FIFO enable bit is changed.

14.4.5.2 OE – Overrun Error

If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the G-Bus before new data from the RX Shift Register overwrote the previous contents. OE is cleared when the G-Bus reads this register.

If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but is not transferred to the FIFO.

14.4.5.3 PE – Parity Error

If the FIFOs are disabled, this bit is set if the received data do not have a valid parity bit. This bit is reset when the G-Bus reads this register.

If the FIFOs are enabled, the state of this bit is revealed to the G-Bus when the byte it refers to is at the top of the FIFO.

14.4.5.4 FE – Framing Error

If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. This bit is reset when the G-Bus reads this register.

If the FIFOs are enabled, the state of this bit is revealed to the G-Bus when the byte it refers to is at the top of the FIFO.

14.4.5.5 BI – Break Interrupt

If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). It is reset by the G-Bus that reads this register.

If the FIFOs are enabled, this error is associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: the next character transfer is enabled when SIN goes into the marking state and it receives the next valid start bit.

14.4.5.6 THRE – TX Holding Register Empty

If the FIFOs are disabled, this bit is set to “1” whenever the TX Holding Register is empty and ready to accept new data and it is cleared when the data are transferred to the TX Shift Register.

If the FIFOs are enabled, this bit is set to “1” whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.

14.4.5.7 TEMT – Transmitter Empty

If the FIFOs are disabled, this bit is set to “1” whenever the TX Holding Register and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.

14.4.5.8 FIFOERR – RX-Data Error in FIFO

If the FIFOs are disabled, this bit is always set to “0”. If the FIFOs are enabled, this bit is set to “1” when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register provided there are no subsequent errors in the FIFO.

14.4.6 FIFO Control Registers (FCR0, FCR1)

Table 14-7 lists the fields of the FIFO Control Registers. These fields control the clearing and enabling of the FIFOs as well as the Receive FIFO Trigger level and DMA mode 1.

Table 14-7 FIFO Control Register Field Descriptions

Bit(s)	Write	Comment
0	FIFOE	Enable FIFOs
1	CLRR	Clear RX FIFO
2	CLRT	Clear TX FIFO
3	DMA1	Set DMA mode 1
4,5	"X"	Reserved for future use
6	RFTL0	RX FIFO Trigger Level Bit 0
7	RFTL1	RX FIFO Trigger Level Bit 1

14.4.6.1 FIFOE – FIFO Enabled

Writing a "1" to this bit enables both the RX and TX FIFOs. When the FIFOs are either enabled or disabled, both the RX and the TX FIFOs are reset.

This bit must be a "1" for any of the other bits in the register to have any effect.

14.4.6.2 CLRR – Clear Receive FIFO

Writing a "1" to this bit clears all the bytes in the RX FIFO and resets its counter logic. The RX Shift Register is not affected. This bit is self-clearing.

14.4.6.3 CLRT – Clear Transmit FIFO

Writing a "1" to this bit clears all the bytes in the TX FIFO and resets its counter logic. The TX Shift Register is not affected. This bit is self-clearing.

14.4.6.4 DMA1 – DMA Mode 1

This bit determines the DMA mode which the TXRDY and RXRDY pins support.

On reset, or when this bit is cleared, the device operates in DMA Mode 0. When this bit is set, the device operates in DMA Mode 1. This bit has no effect unless the FIFOE bit is set as well.

TXRDY – Mode 0: Goes active (Low) when the TX FIFO or the TX Holding Register is empty. Becomes inactive when a byte is written to the Transmit channel.

TXRDY – Mode 1: Goes active (Low) when there is at least one unfilled position in the TX FIFO. Becomes inactive when the TX FIFO is full.

RXRDY – Mode 0: Becomes active (Low) when there is at least one character in the RX FIFO or the RX Holding Register is full. It become inactive when there are no more characters in the RX FIFO or the RX Holding Register.

RXRDY – Mode 1: Becomes active (Low) when the RX FIFO Trigger Level or Timeout occurs. Becomes inactive when the RX FIFO is empty.

14.4.6.5 RTFL0, 1 – Receive FIFO Trigger Level

Table 14-8 lists the programmable trigger levels at which the amount of received data in the Receive FIFO will trigger an interrupt to service the FIFO.

Table 14-8 Receive FIFO Trigger Levels

RFTL1	RFTL0	RX FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

14.4.7 Interrupt Identification Registers (IIR0, IIR1)

Table 14-9 lists the fields of the Interrupt Identification Registers, and Table 14-10 lists the interpretation of the fields of these registers.

Table 14-9 Fields of Interrupt Identification Registers

Bit(s)	Read	Comment
6, 7	FIFOE	Returns “1” if FIFOs enabled, otherwise “0.”
4, 5	0	Always returns “0.”
3	ID2	Interrupt ID Bit 2. Returns “0” if FIFOs disabled.
2	ID1	Interrupt ID Bit 1
1	ID0	Interrupt ID Bit 0
0	NINT	No interrupt pending

Table 14-10 Interpretation of the fields of the Interrupt Identification Registers

ID2	ID1	ID0	NINT	Priority	Comment
0	0	0	1	–	No Interrupt Pending
0	1	1	0	1	Rx Line Status
0	1	0	0	2	Receive Data Available or RX FIFO Trigger
1	1	0	0	2	Character Timeout Indication
0	0	1	0	3	TX Holding Register Empty
0	0	0	0	4	Modem Status

Pending Interrupts are cleared by the following actions:

Priority 1) Reading Line Status Register

Priority 2) Reading RX Buffer Register

Priority 3) Reading the IIR register if an event is a priority 3 interrupt, OR writing to the TX Holding Register

Priority 4) Reading the Modem Status Register

When multiple interrupts are pending, the interrupt line pulses Low after each service.

After reset, D0 = “1”, D1 – D7 = “0”.

14.4.7.1 Receive Timeout Interrupt

ID2 = “1” indicates an RX FIFO Character Timeout.

A RX FIFO Character Timeout occurs if **all** of the following apply:

1. There is at least one character in the FIFO.
2. The most recent character was received longer than four character periods ago (inclusive of all start, parity, and stop bits).
3. The most recent G-Bus read of the FIFO was longer than four character periods ago.

The Character Timeout is dependent on the RX clock.

The Timeout Interrupt is cleared by a G-Bus read from the RX FIFO.

The Timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a G-Bus read from the RX FIFO.

14.4.7.2 TX FIFO Interrupt

The TX Holding Register Interrupt occurs when the TX FIFO is empty. It is cleared by writing to the TX Holding Register or by reading from the IIR.

The TX FIFO Empty interrupt will be delayed one character period minus the last stop bit period whenever THRE = 1 and are not at least two bytes in the TX FIFO at the same time. If the TX interrupt is enabled, setting bit 0 of the FCR will generate an immediate interrupt.

14.4.7.3 FIFO Polled Operation

If the FIFOs are enabled and at least one of the active bits in the IER is disabled, the UART will operate in the FIFO Polled mode. Since the Transmit and Receive paths are controlled separately, either one or both can be in the Polled mode. The application software should check Transmit and Receive status using the LSR.

14.4.8 Interrupt Enable Registers (IER0, IER1)

Table 14-11 details the functionality of the Interrupt Enable Register bit fields.

Table 14-11 Fields of Interrupt Enable Registers

Bit(s)	Field Name	Description
7:4	–	Reserved
3	EDSSI	Enable Modem Status Interrupt. When set (“1”), an interrupt is generated if D0, D1, D2, or D3 of the Modem Status Register have been set.
2	ELSI	Enable RX Status Interrupt. When set (“1”), an interrupt is generated if D1, D2, D3 or D4 of the Line Status Register have been set.
1	ETBEI	Enable TX Holding Register Empty Interrupt. When set (“1”), an interrupt is generated if either THRE=1 or the TX Holding Register is empty.
0	ERBFI	Enable RX Buffer Register. When set (“1”), an interrupt is generated if the RX Buffer contains data.

Note: After reset, bits 0 through 7 are all “0”.

14.4.9 Modem Control Registers (MCR0, MCR1)

Table 14-12 Modem Control Register Fields

Bit(s)	Write	Read	Comment
7:5	X	0	
4	Loop	Loop	Loop-back mode
3	OUT2	OUT2	Control Signal
2	OUT1	OUT1	Control Signal
1	RTS	RTS	Control Signal
0	DTR	DTR	Control Signal

After reset, bits 0 through 7 are all zero.

14.4.9.1 LOOP

When set (“1”), the following conditions are implemented:

1. SOUT is forced to “1.”
2. SIN is disconnected from the RX Shift Register input.
3. RX Shift Register input is connected to TX Shift Register output.
4. The Modem status signals (CTS*, DSR*, DCD*, and RI*) are disconnected.
5. The Modem control signals are connected to the modem status inputs (RTS* to CTS*, DTR* to DSR*, OUT1* to RI*, OUT2* to DCD.)

When clear (“0”), the Modem control/status signals and SIN/SOUT are as normal.

14.4.9.2 OUT2, OUT1, RTS and DTR

These signals control the state of their corresponding outputs (OUT2*, OUT1*, RTS*, and DTR*) even in the Loop Mode.

DTR* = “1” when DTR = “0”. (Same for OUT2, OUT1, & RTS)

DTR* = “0” when DTR = “1”. (Same for OUT2, OUT1, & RTS)

14.4.10 Modem Status Registers (MSR0, MSR1)

Table 14-13 lists the fields of the Modem Status Registers that contain the status bits for the modem.

Table 14-13 Modem Status Register Fields

Bit	Read	Comment
7	DCD	Data Carry Detect
6	RI	Ring Indicator
5	DSR	Data Set Ready
4	CTS	Clear To Send
3	DDCD	Delta Data Carry Detect
2	TERI	Trailing Edge Ring Indicator
1	DDSR	Delta Data Set Ready
0	DCTS	Delta Clear to Send

Note: Reading the Modem Status Register clears the Delta bits.

14.4.10.1 DCD – Data Carry Detect

When Loop = “0”, this is the complement of the DCD* input signal.

When Loop = “1”, this is equal to the OUT2 bit in the Modem Control Register.

14.4.10.2 RI – Ring Indicator

When Loop = “0”, this is the complement of the RI* input signal.

When Loop = “1”, this is equal to the OUT1 bit in the Modem Control Register.

14.4.10.3 DSR – Data Set Ready

When Loop = “0”, this is the complement of the DSR* input signal.

When Loop = “1”, this is equal to the DTR bit in the Modem Control Register.

14.4.10.4 CTS – Clear To Send

When Loop = “0”, this is the complement of the CTS* input signal.

When Loop = “1”, this is equal to the RTS bit in the Modem Control Register.

14.4.10.5 DDCD – Delta Data Carry Detect

This bit is set (“1”) if the state of DCD has changed since the Modem Status Register was last read.

14.4.10.6 TERI – Trailing Edge Ring Indicator

This bit is set if the RI* input has changed from “0” to “1” since this register was last read.

14.4.10.7 DDSR – Delta Data Set Ready

This bit is set (“1”) if the state of DSR has changed since this register was last read.

14.4.10.8 DCTS – Delta Clear To Send

This bit is set (“1”) if the state of CTS has changed since this register was last read.

14.4.10.9 MSR RESET and SAMPLE TIMING

After reset bits D0 through D3 are “0” and can be written to, and bits D4 through D7 are inputs.

A modem status interrupt can be cleared by writing “0,” or set by writing “1” to this register.

A change in any of the modem status input signal levels will be sampled twice by gbsBusClk before there is any change to the Modem Status Register value.

14.4.11 Scratch Registers (SCR0, SCR1)

This is a general-purpose read/write register. After reset, the content of this register is undefined.

14.4.12 Pre-scalar Register

The incoming clock gbsBusClk (provided by the 66/50 MHz TX7901 clock) is divided by the value held in the pre-scalar registers to produce the prescaler output signal, which is then passed onto the Baud Rate Generator.

14.4.13 Divisor Latch LS and MS Registers (DLL, DLM)

The table below shows the divisor needed to generate a prescaler output of approximately 8 MHz. The effective Clock Enable generated is 16x the required baud rate.

Table 14-14 Prescaler output and divide values for various CPU & G-Bus Clocks

RefClk (MHz)	G Bus Clock (MHz)	Divide Value	Prescaler Output (MHz)
133	66.6	8	8.333
100	50.0	6	8.333
66	33.3	4	8.333

Table 14-15 Clock Frequency and Percent Error

Pre-scaled Clock	8.3333 MHz	
Baud Rate	Divisor for 16x clock	Error %
50	10417	-0.00320
75	6944	0.00640
110	4735	-0.00320
135	3872	0.00947
150	3472	0.00640
300	1736	0.00640
600	868	0.00640
1200	434	0.00640
1800	289	0.12175
2000	260	0.16026
2400	217	0.00640
3600	145	-0.22350
4800	109	-0.45234
7200	72	0.46939
9600	54	0.46939
19200	27	0.46939
38400	14	-3.11880
56000	9	3.33995
128000	4	1.72526
256000	2	1.72526

14.5 Special Features

This section discusses how and where the TX7901 UARTs differ from the original NS16550A device.

14.5.1 Transmit Machine Timing

TXM (Transmit Machine) starts after 2 – 3 baud clock cycles from the time the TX Holding Register is written. SOUT goes Low 7 – 8 baud clock cycles from the time the TX Holding Register is written.

14.5.2 THR Empty Interrupt Timing

A TX Holding Register Empty interrupt will be generated 17 – 18 clocks after data have been written to the TX Holding Register, providing Transmit Machine Was Idle when the data were written.

If the TX Holding Register is empty when the TX Holding Register Empty interrupt is enabled, an interrupt will be generated immediately. In the original device, this interrupt is reset on the falling edge of WR* (Transmit register). In the UART, it is reset on the rising edge of this signal.

14.5.3 FIFO Reset Timing

When using bits 0 – 3 of the FIFO Control Register to reset the FIFOs, the following timing restrictions apply:

FCR0 – Both FIFOs are reset by the master reset (MR), and are held unless FCR0 is set to 1.

FCR1 – The RX FIFO clear requires at least one CLK period to complete the reset and clear itself.

FCR2 – The TX FIFO clear requires at least one CLK period to complete the reset and clear itself.

14.5.4 Rx Line Status Interrupt Timing

In the original device, the Rx Line Status Interrupt timing is reset on the rising edge of RD* (Receive register). In the UART, it is reset on the falling edge of this signal.

14.5.5 Timeout interrupt timing

In the original device, the Timeout Interrupt is cleared on the falling edge of RD* (Receive register). In the UART, it is cleared one half-clock cycle after the rising edge of this signal.

14.6 Implemented restrictions

14.6.1 Package pins

For UART1, only SIN and SOUT arrive at the package pins. Other input signals are tied internally, and other output signals remain open.

Inputs	Level
RCLK	L
RCLK_BAUD	H
DCD	L
RI	L
DSR	L
CTS	L

Outputs Remaining Open
BAUD
OUT1
OUT2
RTS
DTR

Note that all signals arrive at the package pins for UART0.

15. Serial Port Interface

15.1 Overview

Boot SPI consists of the TSEI serial port interface and Boot Memory sequencer for Word access (BM/W). TSEI is a Toshiba extended version of a serial peripheral interface (SPI) communication unit, which accesses serial ROM, serial RTC, etc.

The Boot Memory sequencer for Word access (BM/W) is a G-Bus interface which accepts G-Bus word access and translates the word transaction into byte access to the SPI devices.

Table 15-1 SPI Register Addresses (for TSEI)

Address	Register
0x1E00_9000	GPIO_outreg
0x1E00_9004	GPIO_inreg
0x1E00_9008	GPIO_outenab
0x1E00_900C	Reserved
0x1E00_9010	TSEI_SECR
0x1E00_9014	TSEI_SESR
0x1E00_9018	TSEI_SEDR
0x1E00_901C	TSEI_DDCR

SPI Memory Map
(TSEI registers, GPIO registers, and Boot address area)

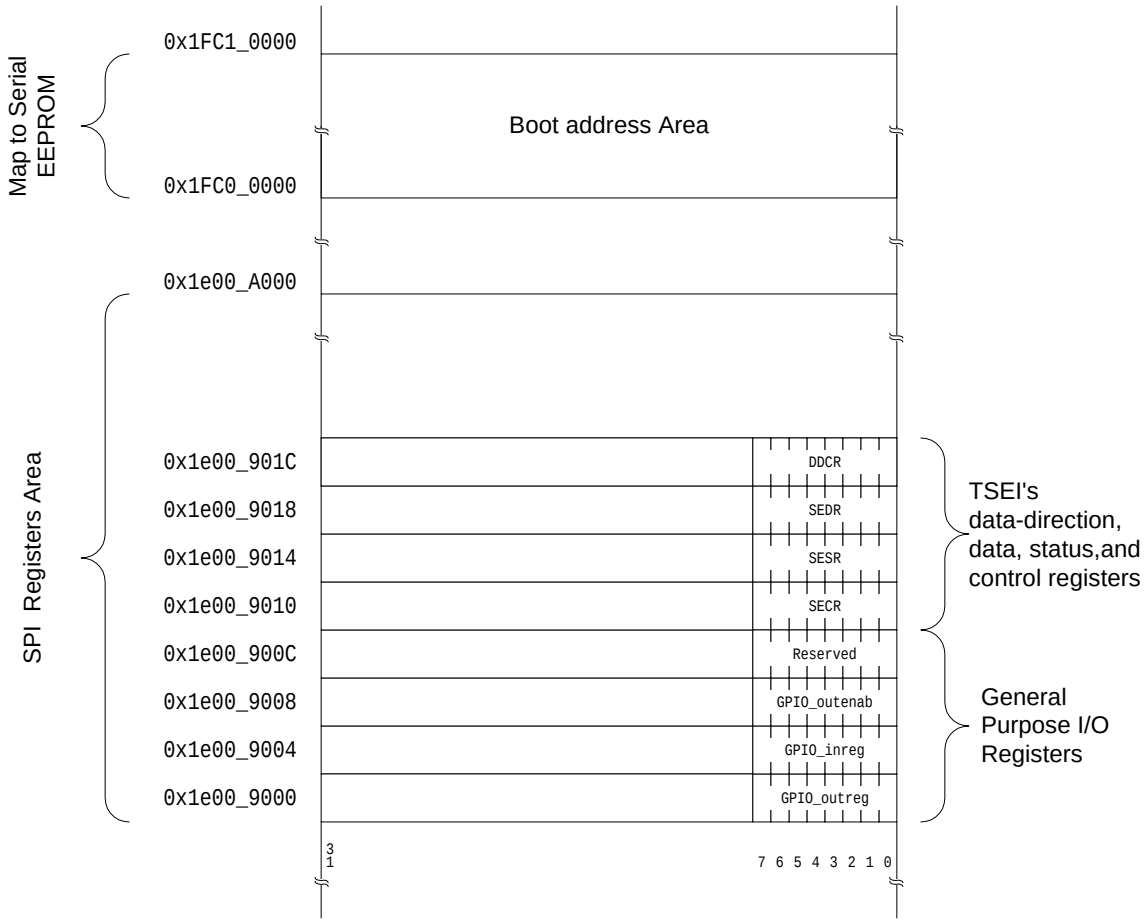


Figure 15-1 SPI Memory Map

Boot SPI is a 32-bit G-Bus slave device that returns GACK32. The 64-bit G-Bus master handles Boot SPI with dynamic bus sizing.

Programming Note: Make sure that TSEI is OFF before changing the clock divisor. Otherwise, this program will no longer be able to run on boot ROM.

15.2 Boot Memory Sequencer for Word Access (BM/W)

When the Boot ROM address is on the G-Bus, BM/W detects the address and starts to access the SPI ROM on Port0. First, BM/W sets some parameters to TSEI.

Table 15-2 TSEI Specifications

Control Register (SECR)

7	6	5	4	3	2	1	0	
SEIE	**SEE**	**BOS**	**MSTR**	**CPOL**	**CPHA**	**SER1**	**SER0**	2'b00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Access
0	0	0	0	0	1	0	0	Reset
0	1	0	1	0	0	1	0	Boot
0	0/1	0/1	1	0	0	1	0	Normal

bit	Name	**val**	**Mode**	**Description**
7	**SEIE**	0	Boot mode	No use
		0	Normal mode	TSEI interrupts are disabled. Polling is used to sense the Transfer Complete flag (SESR[SEF]).
6	SEE	1	Boot mode	TSEI system is on.
		0 1	Normal mode	TSEI system is off TSEI system is on.
5	BOS	0	Boot mode	MSB bit of the SEDR register (data7) will be transmitted first. (compatible to Atmel AT25256 SPI Serial EEPROM)
		0 1	Normal mode	Same as Boot mode. LSB bit of the SEDR register (data0) will be transmitted first.
4	MSTR	1	Boot mode	TSEI is configured as master (generate clock).
		1	Normal mode	Same as Boot mode.
3	CPOL	0	Boot mode	Active high clock selected; SCLK idles Low.
		0	Normal mode	Same as Boot mode.
2	CPHA	0	Boot mode	Bit stream is shifted in on first rising edge of clock (CPOL = 0).
		0	Normal mode	Same as Boot mode.
1 0	SER1 **SER0**		Boot mode	SER0 SER1 | TSEI Clock = GBUS CLOCK Divided by 1 0 | 32 (EEPROM: 2-5MHz) (Assume GBUSCLK = 50 MHz)
			Normal mode	SER0 SER1 | TSEI Clock = GBUS CLOCK Divided by 0 0 | 8 0 1 | 16 1 0 | 32 1 1 | 128

Status Register (SESR)

7	6	5	4	3	2	1	0	
SEF	**WCOL**	**SOVF**	**MODF**	**TSRC**	**TSTC**	**TASM**	**TMSE**	2'b01
R	R	R	R	-	-	R/W	R/W	Access
0	0	0	0	0	0	0	0	Reset
x	x	x	x	x	x	1	1	Boot
x	x	x	x	x	x	1/0	1/0	Normal

bit	Name	val	Mode	Description
1	**TASM**	1	Boot mode	Automatically shift the input data from MISO pin to Data Register when performing a read to the Data Register.
		0 1	Normal mode	Compatible mode: don't care Toshiba mode: 0 Disable automatic mode 1 Enable automatic mode
0	TMSE	1	Boot mode	Toshiba mode
		0 1	Normal mode	Compatible mode Toshiba mode

Data Register (SEDR)

7	6	5	4	3	2	1	0	
Data7	**Data6**	**Data5**	**Data4**	**Data3**	**Data2**	**Data1**	**Data0**	2'b10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Access
0	0	0	0	0	0	0	0	Reset
When the TSEI system is configured as a master, transfers are started by a software write to the SEDR reg.								

Data Direction Register (DDCR)

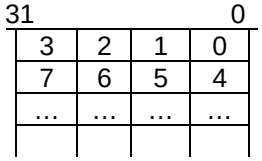
7	6	5	4	3	2	1	0	
-	**-**	**TSS_n**	**TSCLK**	**MOSI**	**MISO**	**-**	**-**	2'b11
-	-	R/W	R/W	R/W	R/W	-	-	Access
0	0	0	0	0	0	0	0	Reset
-	-	1	1	1	x	-	-	Boot
-	-	1	1	1	x	-	-	Normal

bit	Name	val	Mode	Description
5	**TSS_n**	1	Boot mode	This bit is tied to TSS_n pin which is tied High (Vdd).
		1	Normal mode	Same as Boot mode.
4	TSCLK	1	Boot mode	Serial clock output
		1	Normal mode	Same as Boot mode.
3	MOSI	1	Boot mode	Enable master serial output.
		1	Normal mode	Same as Boot mode.
2	MISO	x	Boot mode	When the TSEI system is enabled as a master, the MISO pin acts as the master serial data input, regardless of the state of this bit.
		x	Normal mode	Same as Boot mode.

This is compatible with the Atmel* AT25HP256/512. The bit rate is 2.08 MHz (fGBus = 66 MHz) or 1.56 MHz (fGBus = 50 MHz). = fGBus/4/SER.

(See Figure 15-1 above.)

BM/W locates the first byte on the LSB D[7:0], the second byte on D[15:8], third byte on D[23:16], and the fourth byte on D[31:24], regardless of the CPU endianness. The same ROM may be used in both the Big and Little Endian modes.



15.2.1 Boot ROM (64 KB)

The address range for 64 KB Boot ROM is from 0x1FC0_0000 to 0x1FC0_FFFF.

The acceptable data size of the Boot ROM area is as follows:

Word	(32-bit/4-Byte)	MIPS int. LW/SW
Double Word	(64-bit/8-Byte)	MIPS inst. LD/SD
Quad Word	(128-bit/16 Byte)	C790 Bus Width (instruction fetch)
4 QW	(512-bit/64-Byte)	C790 cache line size (cache refill)
8 QW	(1024-bit/128-Byte)	G-Bus maximum burst size

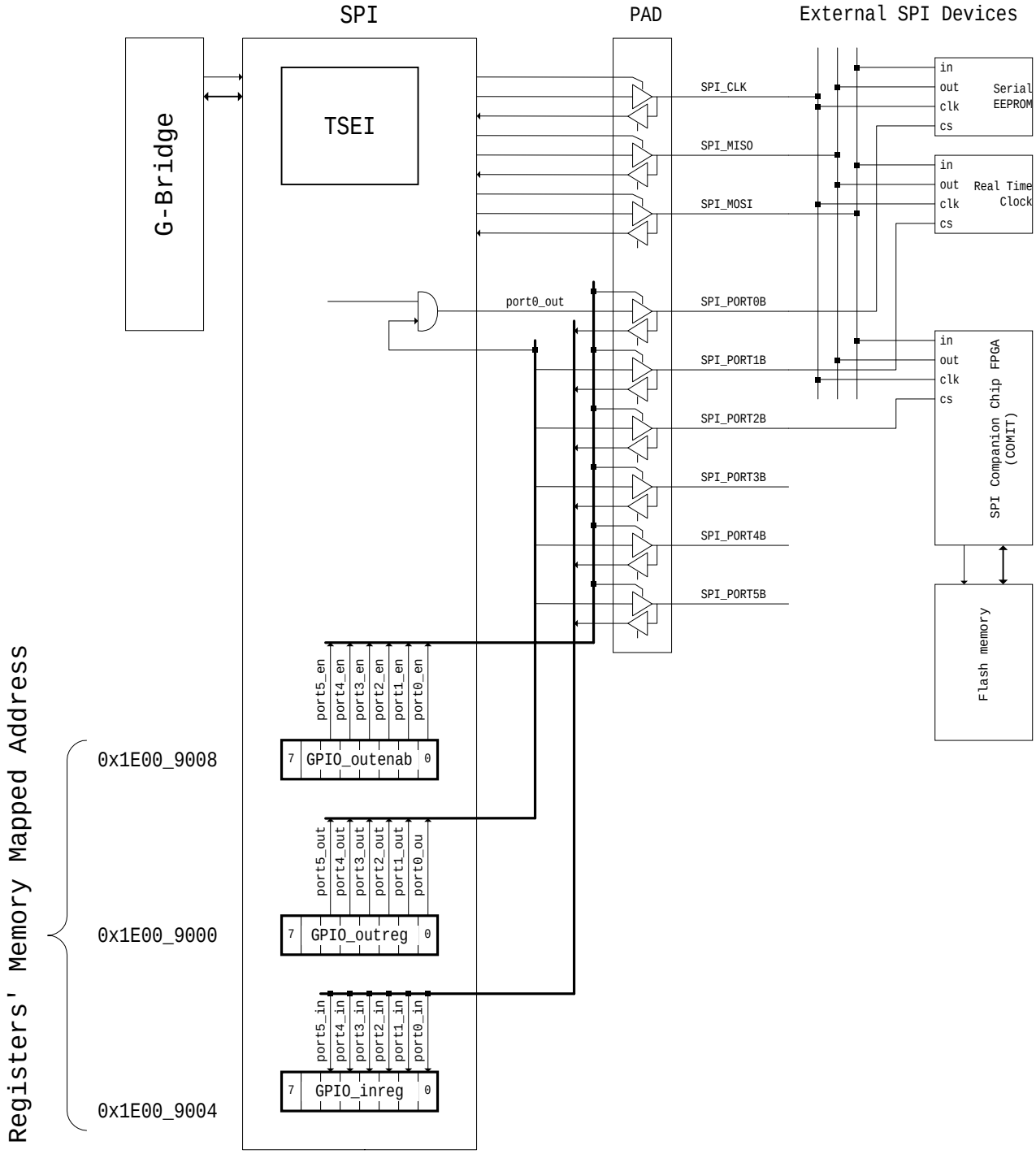
When a read transaction in this address range is on the G-Bus, BM/W initializes TSEI to the Boot configuration (see Figure 15-2).

First, BM/W drives CE0 (SPI_Port0B) to Low, the BM/W issues a read command (0x03) to serial ROM via TSEI. Then, 2 Bytes of the address is sent to Serial ROM. BM/W receives the requested number of data bytes from Serial ROM via TSEI. Finally, BM/W drives CE0 (SPI_Port0B) to High. BM/W then returns the data on the G-Bus.

Important programming note: Do not attempt to access any TSEI registers while the boot ROM area is accessed by both instruction fetch and data read/write. BM/W initializes TSEI every time the boot ROM area is accessed and this breaks any settings made by software.

* Atmel is a registered trademark of Atmel Corp.

SPI External Interface & General Purpose I/O



15.3 TSEI Overview

The Toshiba Serial Expansion Interface (TSEI) is a synchronous communication unit that is compatible with the timing diagrams of peripheral devices, and can be connected to a Serial Port Interface.

Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols: *master* or *slave*. When the TSEI module is configured as a master, software selects one of four different bit rates for the serial clock.

Error-detection logic is included to support inter-processor communications. A write-collision detector indicates when an attempt is made to write data to the serial shift register while a transfer is in progress. A multiple master mode fault detector automatically disables TSEI output drivers if more than one MCU simultaneously attempts to become bus master.

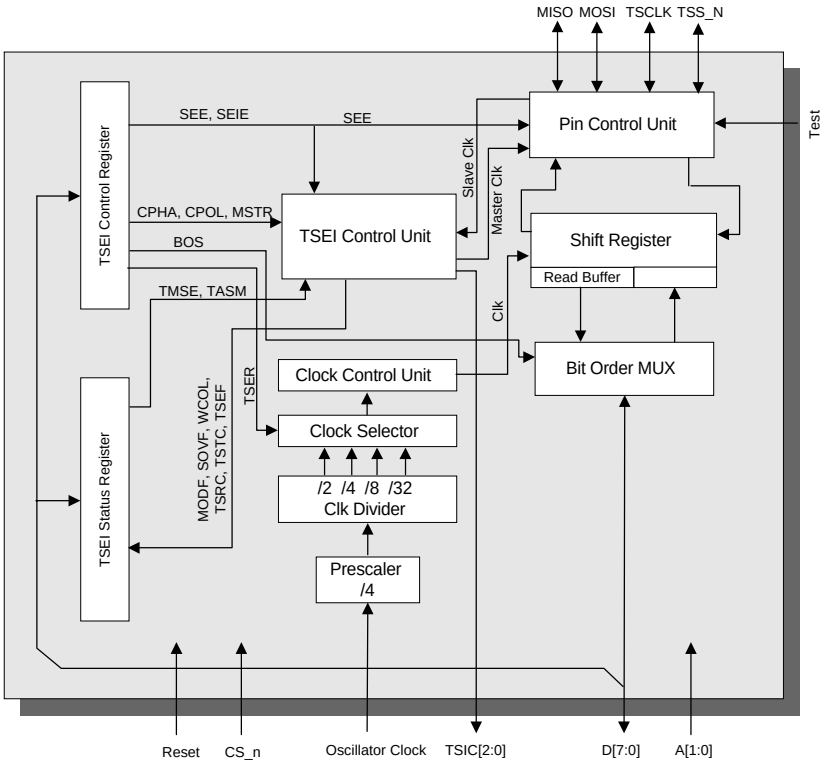


Figure 15-2 TSEI Block Diagram

A special Toshiba operation mode allows the use of MicroDMAs with Toshiba's line of 900/H CPUs, allowing automated data transfer of larger data-blocks without CPU utilization.

15.4 TSEI Transfers

During a TSEI transfer, data are simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line synchronizes the shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave TSEI device; slave devices that are not selected do not interfere with TSEI bus activities. On a master TSEI device, the slave select line can optionally be used to indicate a multiple-master bus connection.

15.4.1 TSEI Clock Phase and Polarity Controls

Software can select any of four combinations of serial clock (SCLK) phase and polarity, using two bits in the TSEI control register (SECR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two fundamentally different transfer formats. The clock phase and polarity should be identical for the master TSEI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

15.4.2 TSEI Data and Clock Timing

The adjustable data clock timings of the TSEI module are compatible with SPI standard data clock timings. The SPI interface allows connection with almost any synchronous serial peripheral device.

Please see Figure 8-1 and Chapter 8 for a detailed description of the transfer format.

15.5 TSEI Signals and Pins

There are four I/O pin signals associated with TSEI transfers. Each signal is dependent on the mode (master/slave) of the TSEI device.

15.5.1 SCLK Pin

In the Master Mode, the SCLK pin is an output when the TSEI is configured as a master and is an input when the TSEI is configured as a slave. When the TSEI is configured as a master, the SCLK signal is derived from the internal TSEI clock, depending on the TSEI interface mode. When the master initiates a transfer, eight clock cycles are automatically generated on the SCLK pin. When the TSEI is configured as a slave, the SCLK pin is an input, and the clock signal from the master synchronizes the data transfer between the master and slave devices. Slave devices ignore the SCLK signal unless the slave select pin is Active Low. In both the master and slave TSEI devices, data are shifted on one edge of the SCLK signal and are sampled on the opposite edge where data are stable. Edge polarity is determined by the TSEI transfer protocol.

15.5.2 SDMISO and SDMOSI

The SDMISO and SDMOSI data pins are used for transmitting and receiving serial data. When the TSEI is configured as a master, SDMISO is the data input line, and SDMOSI is the master data output line. When the TSEI is configured as a slave, these pins reverse roles. In a multiple master system, all SCLK pins are tied together, all SDMOSI pins are tied together, and all SDMISO pins are tied together. A single TSEI device is configured as a master and all other TSEI devices on the TSEI bus are configured as slaves. The single master drives data out from its SCLK and SDMOSI pins to the SCLK and SDMOSI pins of the slaves. One selected slave device optionally drives data out from its SDMISO pin to the SDMISO master pin.

15.5.3 SS_N

The SS_n pin behaves differently on master and slave devices. On a slave device, this pin enables the TSEI slave for a transfer. If the SS_n pin of a slave is inactive (high), the device ignores TSCLK clocks and keeps the SDMISO output pin in the high-impedance state. On a master device, the SS_n pin serves as an error-detection input for the TSEI. If the SS_n pin goes low while the TSEI is a master, it indicates that some other device on the TSEI bus is attempting to be a master. This attempt causes the master device sensing the error to immediately exit the TSEI bus to avoid potentially damaging driver contentions. Such error detection is called “mode fault.”

15.6 TSEI Transfer Formats

The CPHA and CPOL registers in the SECR register set the transfer format. CPHA switches between fundamentally different transfer protocols.

15.6.1 CPHA Equals 0 Format

Figure 15-3 shows the transfer format for a CPHA=0 transfer.

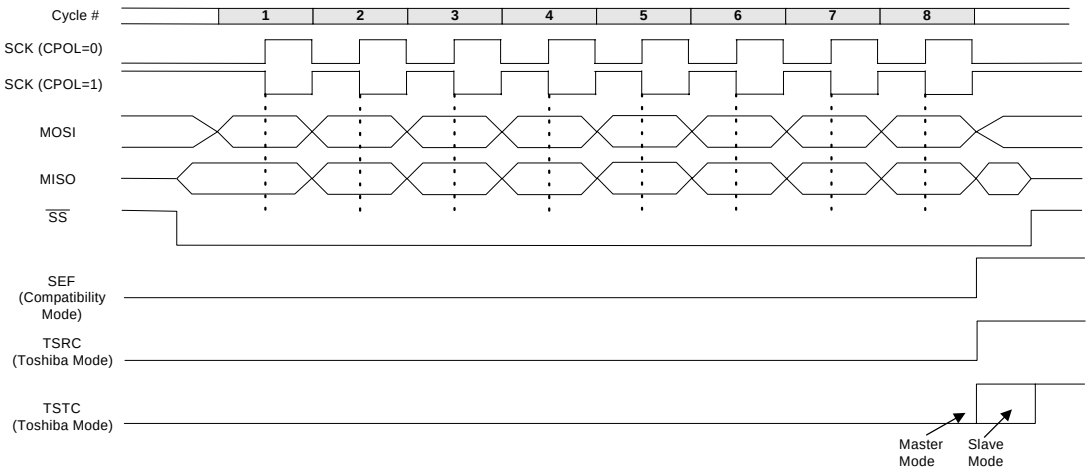


Figure 15-3 CPHA Equals 0 Transfer Format

In this transfer format, the first bit is shifted in on the first clock edge. This will be on a rising edge when CPOL equals 0 and on a falling edge when CPOL equals 1. When CPOL equals

0, the shift clock will idle Low. When CPOL equals 1, it will idle High.

In the Master Mode, when a transfer is initiated by writing a new value to the SEDR register, the new data are placed on the MOSI signal for half a clock cycle before the shift clock starts to operate. The BOS flag determines whether the value will be shifted out in an MSB or LSB order. After the last shift cycle, the SEF flag (in Compatibility Mode) or the TSRC flag and TSTC flags (in Toshiba Mode) will be asserted.

In the Slave Mode, the SEDR register is not allowed to be written to when the SS_n signal is low. A write attempt in this state will result in a write collision and the WCOL bit will be asserted in the SESR register. Therefore, even if the transfer has been completed and the SEF or TSRC bit has been asserted, the software has to wait until the SS signal goes High again before writing a new value to SEDR. To allow the use of a MicroDMA to transfer values to the SEDR register in the Slave Mode, the TSTC signal is delayed until SS_n goes High.

15.6.2 CPHA EQUALS 1 FORMAT

Figure 15-4 shows the transfer format for a CPHA=1 transfer.

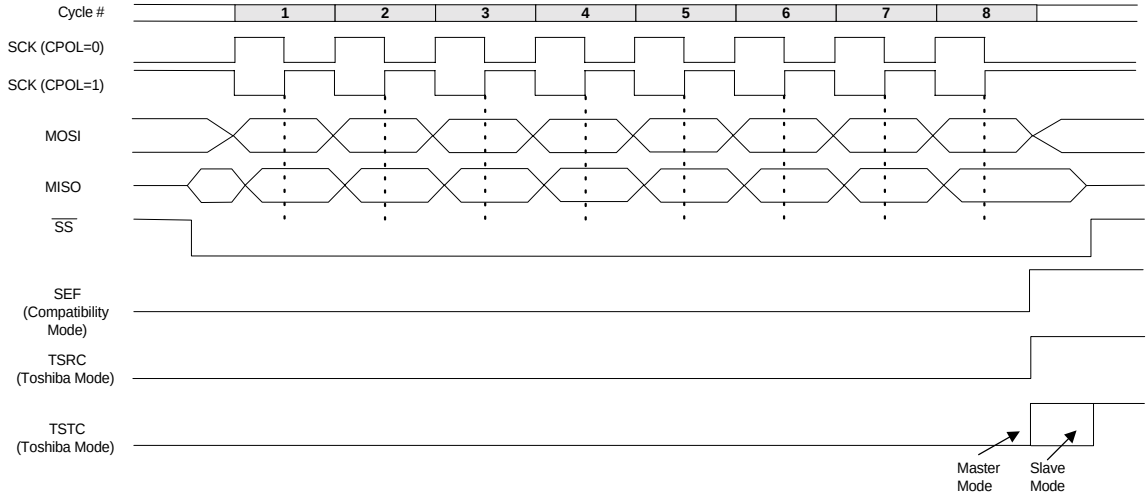


Figure 15-4 CPHA Equals 1 Transfer Format

In this transfer format, the first bit is shifted in on the second clock edge. This will be on a falling edge when CPOL equals 0 and on a rising edge when CPOL equals 1. If CPOL equals 0, the shift clock will idle low. When CPOL equals 1, it will idle high.

In the Master Mode, when a transfer is initiated by writing a new value to the SEDR register, the new data are placed on the MOSI signal with the first edge of the shift clock. Again, the first bit to be transferred will be determined by the BOS bit in the SECR register.

Unlike the CPHA equals 0 format, SEDR is allowed to be written to in the Slave Mode even if the SS_n signal is low.

In both the Master Mode and the Slave Mode, the SEF flag (in the Compatibility Mode) or the TSRC and TSTC flags (in the Toshiba Mode) will be asserted simultaneously after the last shift cycle completes. Any attempt to write to this register while the data shifting is still in progress will result in a write collision.

15.7 MCU Interface

Figure 15-5 shows the transactions for the TSEI MCU Interface. In particular, it shows a read access followed by a write access.

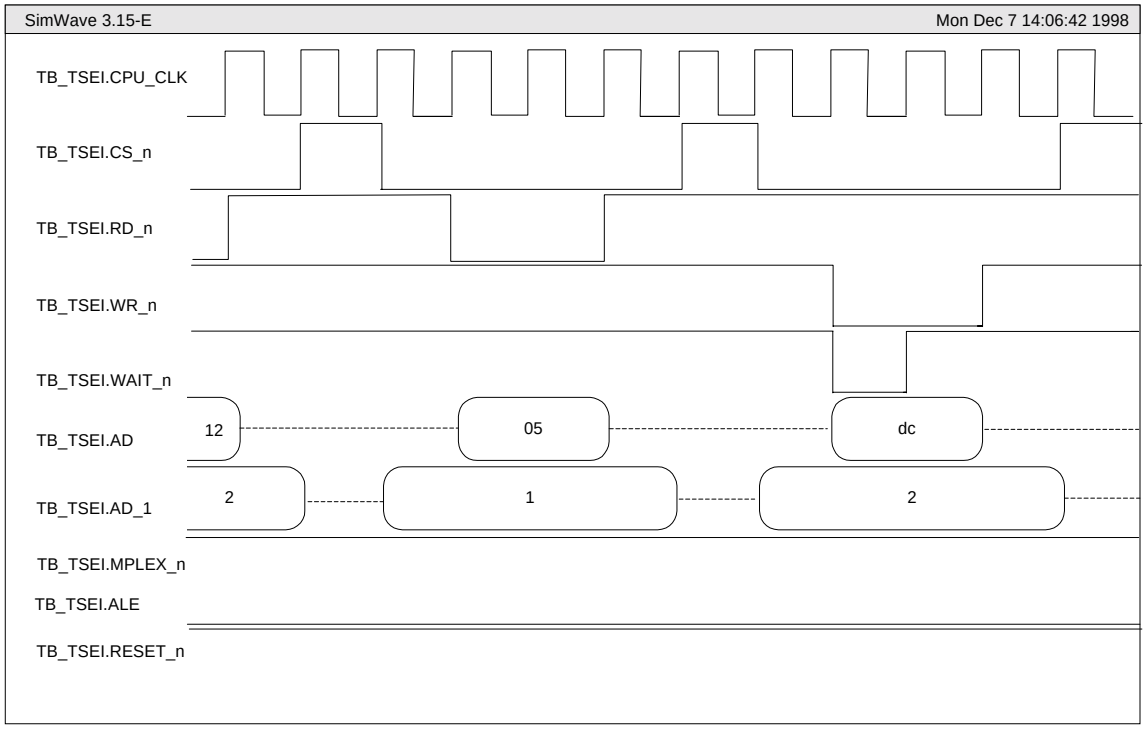


Figure 15-5 MCU Interface Signaling

15.7.1 Read Access

The read access starts by placing the address of the register to be read on the address bus A[1:0]. The Chip Select signal is asserted at the same time. These values are sampled on the following positive clock edge and the output data is prepared during the next clock cycle. Here the RD_n signal is also asserted, enabling the output on the data bus AD[7:0]. The correct time to sample the value from the data bus is on the next positive clock edge of the CPU clock. At the next positive clock edge, the RD_n signal is deasserting, tri-stating the output drivers for the data bus. CS_n is deasserted on the following clock edge.

15.7.2 Write Access

Similar to Read Access above, Write Access begins by asserting the CS_n signal. The address of the value to be written is placed on the data bus at the same time. On the next positive clock edge, the value to be written is placed on the data bus and the WR_n signal is asserted simultaneously. The Wait signal is also asserted and remains asserted for one clock cycle. The WR_n signal remains asserted for two clock cycles. The deassertion of the CS_n signal ends the transfer on the following clock edge.

15.8 TSEI REGISTERS

The TSEI Control Register (SECR), TSEI status register (SESR), and TSEI data register (SEDR) are software-accessible registers used to configure and operate the TSEI system.

15.8.1 Control Register (SECR)

This register, which may be read or written at any time, is used to configure the TSEI system.

	7	6	5	4	3	2	1	0	
	SEIE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0	0h
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset:	0	0	0	0	0	1	0	0	

SEIE: TSEI Interrupt Enable
Compatibility Mode:

0: TSEI interrupts are disabled. Polling is used to sense the SEIF and MODF flags.

1: A TSEI interrupt is requested if SEF or MODF is being asserted.

Toshiba Mode:
This flag is obsolete in the Toshiba mode. Only the Interrupt Controller registers are used to enable or disable interrupts.

SEE: TSEI System Enable

0: TSEI system is off.

It is necessary to disable the TSEI system to switch between the Toshiba operating mode and the compatibility mode.

1: TSEI system is on.

BOS: Bit Order Select
This bit determines the data bit which will be transmitted first.

0: The MSB bit of the SEDR register (data7) will be transmitted first

1: The LSB bit of the SEDR register (data0) will be transmitted first

MSTR: Master/Slave Mode Select

0: TSEI is configured as slave.

1: TSEI is configured as master.

The function of the TASM bit in the SESR register depends on the setting of this bit.

CPOL: Clock Polarity Select

0: Active high clocks selected, SCLK idles Low.

1: Active low clocks selected; SCLK idles High.

CPHA: Clock Phase Select
This control bit selects one of two fundamentally different transfer formats.

SER1, SER0: TSEI Bit Rate Select
The following table shows the relationship between the SER1 and SER0 control bits and the bit rate for transfers when the TSEI is operating as a master. When the TSEI is operating as a slave, the serial clock is input from the master; therefore, the SER1 and SER0 control bits have no meaning.

SER1	SER0	TSEI Clock Divided By
0	0	**2**
0	1	4
17	0	**8**
1	1	**32**

15.8.2 TSEI STATUS REGISTER (SESR)

This register, which may be read or written to at any time, is used to indicate the current state of the TSEI system and is used to switch between the TSEI operation modes. The status flags can be cleared only in the Toshiba Mode by writing a “1” to them. Writing a “0” value to these flags has no effect.

	7	6	5	4	3	2	1	0	
	SEF	WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE	1h
Comp. Mode	R	R	R	R	–	–	R/W	R/W	
Toshiba Mode	–	R/C	R/C	R/C	R/C	R/C	R/W	R/W	
Reset:	0	0	0	0	0	0	0	0	

SEF: TSEI Transfer Complete Flag

Compatibility Mode:

This flag is automatically set to “1” at the end of a TSEI transfer. SEF is automatically cleared by reading the SESR register with the SEF bit set.

Toshiba Mode:

Switching to the Toshiba Mode clears this flag. This flag always reads as “0,” and writes to this flag have no effect.

WCOL: Write Collision Error Flag

Compatibility Mode:

This flag is automatically asserted if the SEDR register is written to while a transfer is in progress. The write itself has no effect on the running transmission. The WCOL flag is automatically cleared by reading the SESR register with the WCOL bit set, then accessing the SEDR register. No interrupt will be generated when asserting this flag.

Toshiba Mode:

The flag can only be reset by writing a “1” value to it. Writing a “0” has no effect. An interrupt will be generated on TSIC0 on a transition from “0” to “1” if the module is configured as a slave and TASM is equal to “0.”

SOVF: Slave Mode Overflow Error

Compatibility Mode:

This flag is used to detect a state where the TSEI module is configured as a slave and the module was unable to follow the transmission of the busmaster on the SEI bus. It is asserted when the TSEI module is configured as a slave, a new byte has been completely received, and the TSEF flag is still asserted. No interrupt will be generated when asserting this flag. It is cleared by reading SESR with the SOVF bit set, then accessing SEDR. Switching to the Master Mode will also clear the flag.

Toshiba Mode:

The TSRC flag is used instead of the TSEF flag to determine whether the Data Register has been read out. An interrupt will be generated on TSIC0 if the TASM bit in the SESR register is asserted.

This flag can only be cleared by writing a “1” to it. Writing a “0” to it has no effect.

MODF: Mode-Fault Error Flag

Compatibility Mode:

This flag is set if the SS_n signal goes to active low while the TSEI is configured as a master (MSTR=1). In this case:

1. The SEI output pin drivers are disabled and the output pins are placed in the high impedance state.
2. The MSTR bit in the SECR register is cleared.
3. The SEE bit is forcibly cleared to disable the SEI system.
4. An interrupt is generated if the SEIE bit is set.

The MODF flag is automatically cleared by reading the SESR register with MODF bit set, then writing to the SECR register.

Toshiba Mode:

This flag can only be cleared by writing a “1” to it. Writing a “0” to this flag has no effect. The SEIE bit is ignored and an interrupt will always be generated on a transition from “0” to “1”.

TSRC: TSEI Receive Completion

Compatibility Mode:

This flag always reads as “0,” and writes to this register have no effect.

Toshiba Mode:

This flag is set when a transfer has been completed — when eight cycles are shifted on the SCK signal. It is cleared by performing a read operation on the TSEI data register, by switching to the Compatibility Mode, or by writing a “1” to this flag. Writing a “0” to this flag has no effect. An interrupt will be generated on TSIC1 when asserting the flag.

TSTC: TSEI Transmit Completion

Compatibility Mode:

This flag always reads as “0,” and writes to this register have no effect.

Toshiba Mode:

This flag is set when one byte of data has been completely shifted out and when new data are allowed to be written to the SEDR register. It is cleared by performing a write operation on the TSEI data register, by switching to the Compatibility Mode, or by writing a “1” to this flag. Writing a “0” to this flag has no effect. An interrupt will be generated on TSIC2 when asserting this flag.

TASM: TSEI Automated Shift Mode

In the Master Mode:

Compatibility Mode:

This flag always reads as “0,” and writes to this register have no effect.

Toshiba Mode:

0: Disables the Automated Shift Mode

1: Enables the Automated Shift Mode

In this Mode, a read access to the TSEI data register (SEDR) will perform the following functions:

- The TSEI data register will be cleared to 00h after it has been read.
- A new transfer will be initiated, thus in the Master Mode 8 low bits will be shifted out, and 8 new bits will be shifted in.

The automated shift mode also works when it is combined with a MicroDMA. It has no effect when TSEI is in the Slave Mode.

In the Slave Mode:

This bit functions as a mask for generating interrupts on the SOVF and WCOL flags.

0: An interrupt will be generated on WCOL, but not on SOVF.

1: An interrupt will be generated on SOVF, but not on WCOL.

TMSE: TSEI Mode Select
Selects a special Toshiba mode that also allows MicroDMA transfers to be performed with Toshiba’s line of 900/H CPUs.

It is necessary to disable the TSEI system before switching to the Toshiba operating mode. After switching to the Compatibility Mode, the TSRC and the TSTC flags are ignored.

15.8.3 TSEI Data Register (SEDR)

This register is the data register of the TSEI system.

	7	6	5	4	3	2	1	0	
	data7	data6	data5	data4	data3	data2	data1	data0	2h
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset:	0	0	0	0	0	0	0	0	

When the TSEI system is configured as a master, transfers are started by a software write to the SEDR register.

15.8.4 TSEI Data Direction Register (DDCR)

This register may be read or written to at any time and is used to enable or disable the input and output pins of the TSEI system.

	7	6	5	4	3	2	1	0	
	-	-	TSS_n	TSCLK	MOSI	MISO	-	-	3h
	-	-	R/W	R/W	R/W	R/W	-	-	
Reset:	0	0	0	0	0	0	0	0	

TSS_n: Slave Select (low active)

When TSEI is enabled as a slave, the TSS_n pin is the slave select input, regardless of the value of the DDCR register.

When TSEI is enabled as a master, the function of the TSS_n pin depends on the value of DDCR bit 5.

0: The TSS_n pin is used as an input to detect mode fault errors. A low TSS_n pin indicates that some other device in a multiple master environment is acting as a master and is trying to select this device as a slave. To prevent harmful contention between output drivers, a mode fault is generated which will disable the module and tri-state all output pins.

1: No mode fault will be generated.

TSCLK: Serial Clock Signal

When TSEI is enabled as a slave, the TSCLK pin will function as an input, regardless of the value in the DDCR register.

When TSEI is acting as a master, bit 4 in the DDCR register must be asserted to enable the TSCLK output.

TSDMOSI: Master Output, Slave Input

When the TSEI system is enabled as a slave, the TSDMOSI pin acts as the slave serial data input regardless of the state of DDCR3.

When the TSEI system is enabled as a master, bit 3 in the DDCR register must be set to “1” to enable the master serial data output. If a master device needs to initiate a TSEI transfer to receive a byte of data from a slave without transmitting a byte, it might purposely leave the TSDMOSI output disabled.

TSEI systems that tie TSDMOSI and TSDMISO together to form a single bi-directional data line also need to selectively disable the TSDMOSI output.

TSDMISO: Master Input, Slave Output

When the TSEI system is enabled as a slave, bit 2 in the DDCR register must be set to “1” to enable the slave serial data output. A master TSEI device can simultaneously broadcast a message to several slaves as long as no more than one slave tries to drive the TSDMISO line. TSEI systems that tie TSDMOSI and TSDMISO together to form a single bi-directional data line also need to selectively disable the TSDMISO output.

When the TSEI system is enabled as a master, the TSDMISO pin acts as the master serial data input, regardless of the state of this bit.

15.9 TSEI System Errors

TSEI is only used in the Slave Mode, and therefore does not report any errors.

15.10 Interrupt Generation

Interrupt processing differs for the two TSEI operating modes, which can be selected using the TMSE bit in the SESR register. The TSEI module is connected to three interrupt channels named TSIC0, TSIC1, and TSIC2.

15.10.1 Compatibility Mode

In the Compatibility Mode, only TSIC Channel 0 is used. This channel generates an interrupt if either the SEF flag or the MODF flag in the SESR register shows a transition from “0” to “1.” The SEIE bit is used as a global interrupt enable/disable.

TSEI Interrupt Channel 0 (TSIC0)	**Interrupt on MODF or SEF**
TSEI Interrupt Channel 1 (TSIC1)	**Inactive**
TSEI Interrupt Channel 2 (TSIC2)	**Inactive**

15.10.2 Toshiba Mode

In the Toshiba Mode, all three interrupt channels are used to allow MicroDMA transfers to, and from, the TSEI data register. Interrupt Channel 0 generates an interrupt on three different sources. The first type of interrupt is generated on a transition of the MODF flag from “0” to “1.” The WCOL interrupt is caused if the module is in the Slave Mode with TASM equal to “0.” Finally, the SOVF flag causes an interrupt only if the module is configured as a slave and TASM is asserted.

Both the TSRC and the TSTC flags in the SESR register are asserted simultaneously after a transfer is complete. Both flags trigger their own interrupt. The TSRC flag generates an interrupt on channel 1 on a transition from “0” to “1”. This flag can be cleared by either reading to the SEDR register or by writing a “1” value to this flag.

The TSTC flag generates an interrupt on channel 2 on a transition from “0” to “1.” This flag is cleared by either writing the SEDR register or by writing a “1” value to this flag.

When using with the MicroDMA, one of the interrupts can be used to trigger a MicroDMA that reads the data from the TSEI register. The other interrupt can be used to trigger a MicroDMA that writes a new value to the data register, thus initiating a new transfer.

TSEI Interrupt Channel 0 (TSIC0)	**Interrupt on MODF or WCOL [1] or SOVF [2]**
TSEI Interrupt Channel 1 (TSIC1)	**Interrupt on TSRC**
TSEI Interrupt Channel 2 (TSIC2)	**Interrupt on TSTC**

The SEIE bit is obsolete in the Toshiba Mode. The Interrupts are individually disabled at the interrupt controller.

15.10.3 Interrupt Generation on TSIC0

If a flag in the SESR register that causes an interrupt shows a transition from “0” to “1,” an interrupt will be generated if no other interrupt flag is already pending.

If an interrupt pulse has been generated by an interrupt source and the interrupt flag for this source has not been cleared, another occurrence of the same source will not generate a new interrupt pulse.

When the flag of a source that caused an interrupt is being cleared, a new interrupt can occur on a different flag. This will happen immediately if another interrupt flag in the SESR register is asserted and was not cleared together with the flag that caused the previous interrupt.

[1] Only if module is in the Slave Mode and TASM in the SESR register is not asserted.
[2] Only if module is in the Slave Mode and TASM in the SESR register is asserted.

Example:

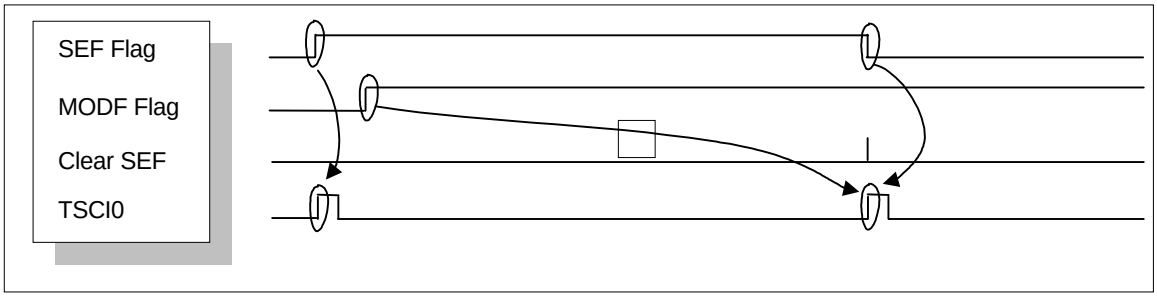


Figure 15-6 TSIC0 behavior, Compatibility Mode

In the example above, the SEF flag is asserted after a completed transfer. On the transition of this flag from “0” to “1,” an IRQ signal is caused on TSIC0. This will be a pulse that is one clock cycle in length.

After this, a mode fault causes a transition of the MODF flag from “0” to “1.” This will not cause another IRQ pulse, since SEF is already asserted. When SEF is cleared and MODF is still asserted, a second IRQ pulse is generated on the MODF flag.

16. Clocks

16.1 Overview

Clocks are one of the fundamental elements of the TX7901, which has three main clock domains, which are depicted in Figure 16-1 and summarized in Table 16-1. By manipulating the internal clock in different ways, the TX7901 can be controlled to run in different modes, and at different clock speeds. Clock control is used for various I/O clock ratios and silicon debug modes.

The TX7901 uses the following clocks as summarized below in Table 16-1.

Table 16-1 TX7901 Clocks

Clock	Description
CPU Clock	Basic clock supplied to the C790 core, data and instruction caches, and MMUs
System Bus Clock	Supplied to the C790 bus and all the devices attached to it, SDRAMC, G-bridge, and DMAC
G-bus Clock	Supplied to all IPs (PCI, MAC, UARTS, SPI and Timer) connected to the G-bus

The TX7901 clock system provides many different clocking options for all on-chip and external devices. The Phase-Locked Loop circuitry can be used to provide a high-frequency system clock from a low-frequency external source. Also, the TX7901 provides various frequency dividers for a variety of clock domains on the chip. Figure 16-2 illustrates internal clock source and distribution that includes the phased-locked loop, clock dividers, drivers, and a generic IP.

16.2 Features

The main features of the TX7901 clocks are as follows:

- f/2, f/3, f/4 dividers are available for the CPU:SysBus clocks
- Fixed f/2 divider for sysBus:gbusClock
- PLL and divider bypass for scan and memory test modes

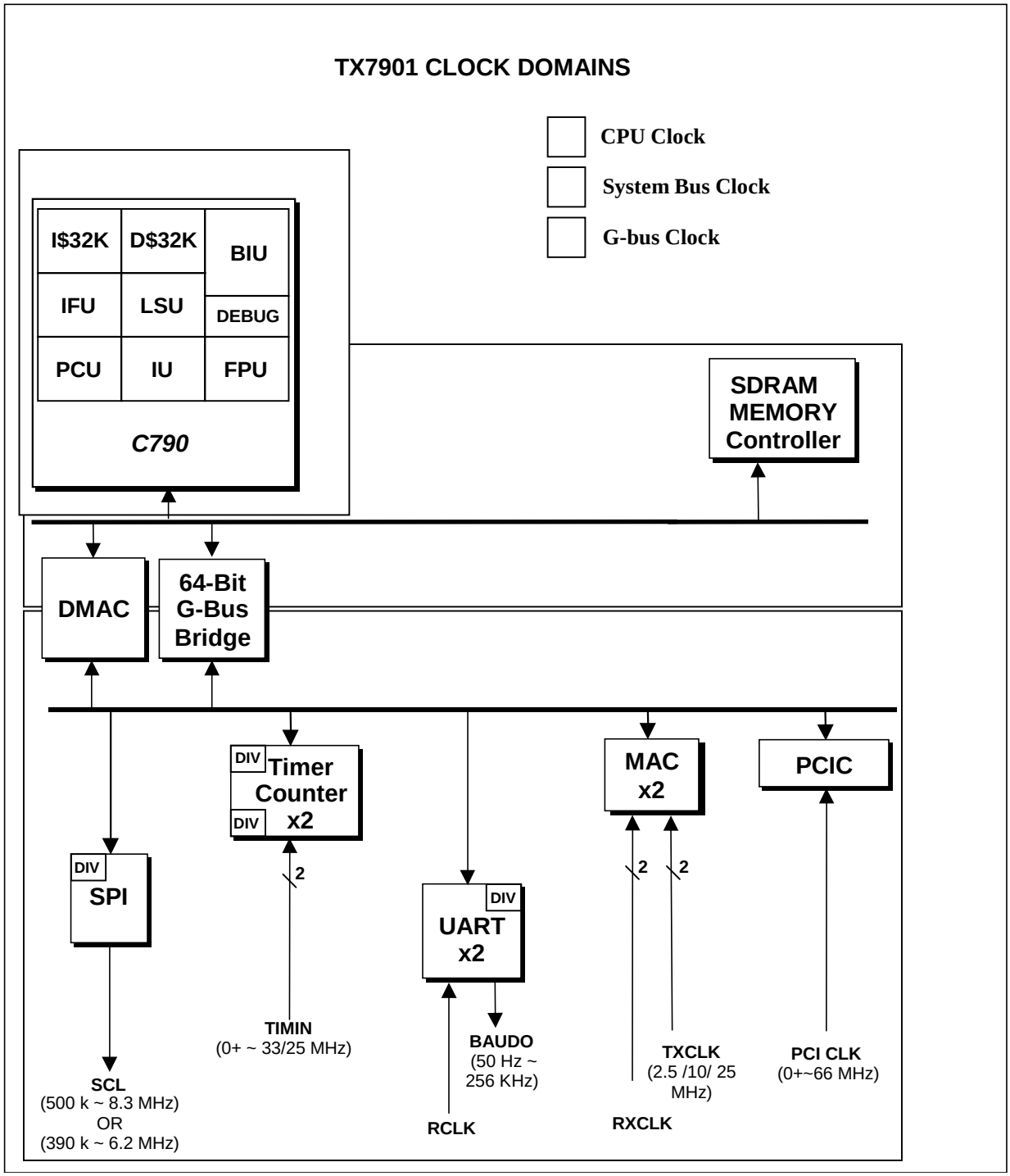


Figure 16-1 TX7901 Clock Domain Diagram

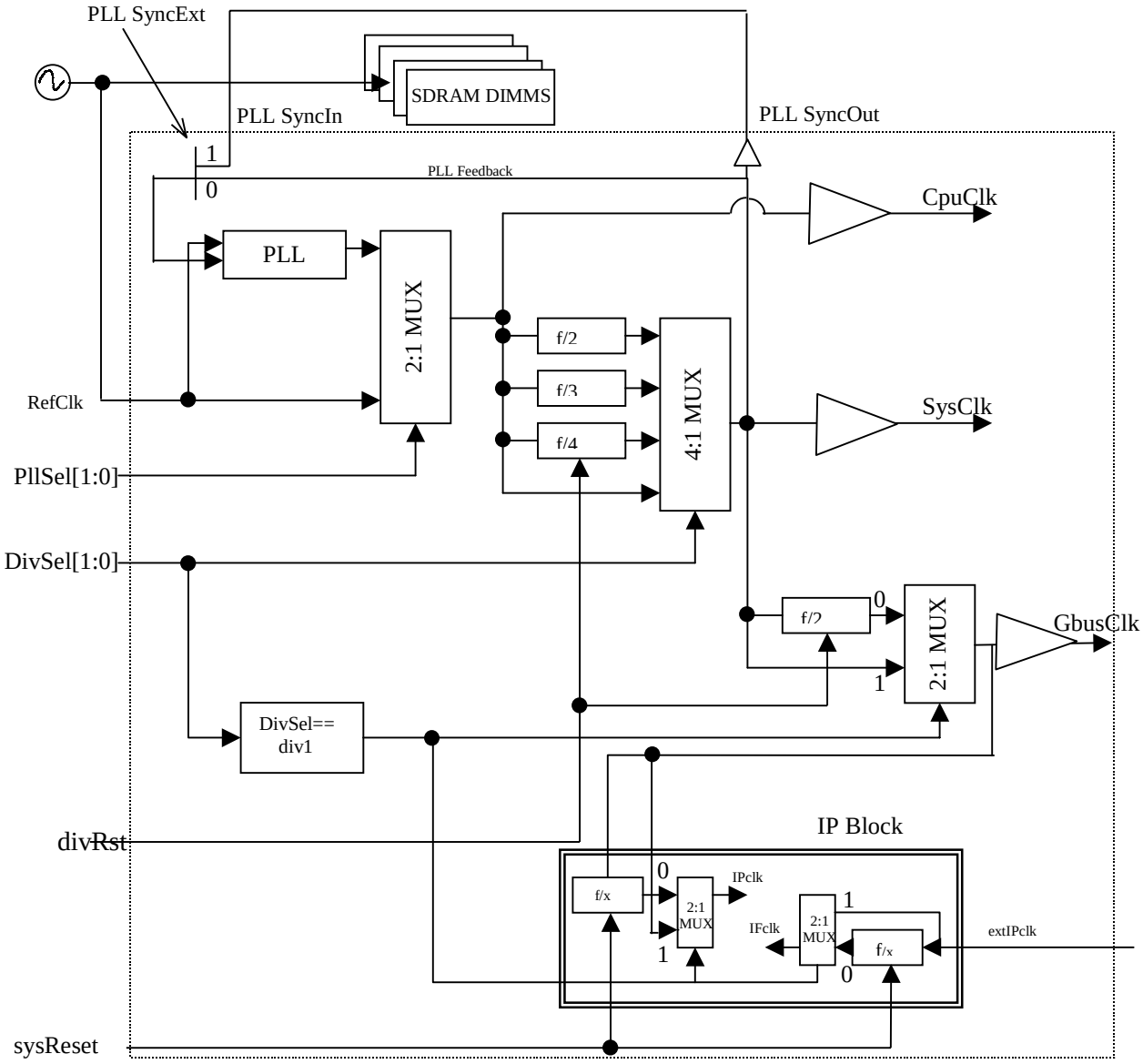


Figure 16-2 TX7901 Clock Distribution Diagram

16.3 Operation

The TX7901 relies on one external reference clock source (refClk) and on-chip phased-locked loop for generating various clocks, as shown in the above Figure. pllSel[1:0]) signals are used to select either PLL or bypass PLL.

Table 16-2 PLL Selection

PllSel[1:0]	Clock Source
00	PLL
01	Reserved
10	Reserved
11	Bypass PLL (test mode)

The PLL generates the overall system operating frequency in integer multiples of the input clock frequency. For special purposes (e.g., test mode), it is possible to disable the PLL, in which case the external reference clock is used as the system operating clock. This mode must not be used as a normal operating mode. Its only valid use is for low-frequency silicon debugging.

The PLL also functions as a multiplier to the cpuClk:sysClk ratios in addition to de-skewing the SDRAMC clock and the external SDRAM clock. During normal operation, refClk is always in sync with sysClk so that the external SDRAM and TX7901 SDRAM controller are running at the same system bus frequency.

The TX7901 also provides a following of clock dividers for cpuClk:sysClk gear ratios. These values of the divider selector (divSel[1:0]) I/O pins are sampled at Reset only.

Table 16-3 Clock Gear Ratio

DivSel[1:0]	CpuClk:sysClk Gear Ratio
00	Divide by 1 (test mode)
01	Divide by 2
10	Divide by 3
11	Divide by 4

For sysClk:gbusClk, only the gear ratio of 2:1 is supported for normal operations. In the test mode all clock domains in the chip run at the same frequency.

Table 16-4 Example Frequency Relationship

RefClk	CpuClk			SysClk	gbusClk
	2	3	4		
133 MHz	266*[1]	400*[2]	533*[2]	133 MHz	66 MHz
100 MHz	200	300*[2]	400*[2]	100 MHz	50 MHz
66 MHz	133	200	266*[1]	66 MHz	33 MHz
50 MHz	100	150	200	50 MHz	25 MHz

*1: Premium products only.
*2: Future product. Not guaranteed at this time.

16.4 Peripheral Module Clock

This section describes the clock requirements of the individual peripheral modules.

16.4.1 MAC Clock

Each 10/100 Ethernet module in the TX7901 is composed of a Tx clock domain and an Rx clock domain. The Rx clock is reconstructed from the received data stream running at 10 MHz or 100 MHz. The reconstruction is done by an external 802.3 Physical Layer device. The 2.5 / 10 MHz Rx clock is used for 10Base Ethernet configuration while the 25 MHz Rx clock is used for 100Base Ethernet configuration.

The Tx clock is generated from an external oscillator. The external oscillator will generate a reference clock to a PHY device. The PHY then constructs a very accurate 2.5 MHz / 10 MHz / 25 MHz TX clock which drives the TX7901 MAC module.

The FIFO structure inside the MAC module synchronizes data transfer between the MAC clock and G-Bus Clock domain.

16.4.2 UART Clock

The UART module has a Transmit and a Receive clock domain. The Receive clock is an input signal from the UART Receive channel. The Transmit clock is generated by the Baud Rate Generator logic in the UART module. The Baud Rate Generation logic is a digital frequency divider and is a sub-module of the UART.

The digital frequency divider takes two steps to generate a Transmit Clock. First, the G-Bus Clock is divided down by a pre-scaler to generate the 8 MHz Clock. Then the 8 MHz clock is divided down to the designated frequency. The Transmit and Receive frequencies' range is between 50 Hz and 256 KHz. Both pre-scaler and Baud Rate dividers are programmable. See Table 14-13 and Table 14-14 for more specific information.

The UART uses FIFO structures to synchronize between Transmit / Receive Logic and the G-Bus.

16.4.3 SPI Clock

The SPI module operates in the Master Mode. When in the Master Mode, SPI needs to drive the clock to the external slaves.

The SPI and G-Bus interface both synchronize Read and Write Control signals between the G-Bus Clock and the SPI clock in order to perform data register access.

The SPI Clock frequency is 1/8, 1/16, 1/32, or 1/128 of the G-Bus clock (gbusClk) frequency.

Table 16-5 SPI Clock Frequency

gbusClk	SPI Clock			
	f/8	f/16	f/32	f/128
66 MHz	8,333 kHz	4,166 kHz	2,083 kHz	520 kHz
50 MHz	6,250 kHz	3,125 kHz	1,562 kHz	390 kHz
33 MHz	4,166 kHz	2,083 kHz	1,041 kHz	260 kHz
25 MHz	3,125 kHz	1,562 kHz	781 kHz	195 kHz

16.4.4 PCI Clock

The PCI module may operate in either the 33 MHz or 66 MHz mode. When the PCI module operates in the 33 MHz mode, it will support PCI input clocks from 0+ MHz to 33 MHz. When it operates in the 66 MHz mode, it will support PCI input clocks from 33 MHz to 66 MHz. The PCI module uses an externally generated clock.

The PCI module uses FIFO structures to synchronize data transfer between the PCI Clock domain and the G-Bus Clock domain.

Note that the PCI Clock frequency must be less than or equal to the G-Bus clock frequency.

16.4.5 Timer Clock

The Timer has counter logic that is referenced to the internal or external clocks. The internal or external clocks will operate in a frequency range of 0+ to 37.5 MHz.

Note that the external Timer Input clock frequency must be less than or equal to one half of the G-Bus clock frequency.

17. Pins

This chapter details the physical pins of the TX7901. Table 17-1 describes the functionality of each group of pins, while Table 17-2 specifies the allocation of pin positions for each individual signal.

Table 17-1 TX7901 Pin Functionality

Rev.2.0 pinout

Name of Signal	I/O	Function
System Bus Interface		
MasterReset	I	Master Reset
BigEndian	I	Input signal to select for Big-Endian or Little Endian at power up.
DPCI_SPCI_SEL	I	To select between different options of TX7901 1: Select Dual PCI 0: Select Single PCI
Interrupt Interface		
NMIB	I	Non-Maskable Interrupt
INT_MAC0B	I	External Interrupt from ether MAC PHY0
INT_MAC1B	I	External Interrupt from ether MAC PHY1
INT_0B	I	External Interrupt Request 0
INT_1B	I	External Interrupt Request 1
INT_2B	I	External Interrupt Request 2
INT_3B	I	External Interrupt Request 3
INT_4B	I	External Interrupt Request 4
Clock Interface		
PLL_DIV0, PLL_DIV1	I	To set up divide Ratio between CPU clock to internal System Bus Clock, 00: Divide By 1 (Test only) 01: Divide by 2 10: Divide by 3 11: Divide by 4
PLL_TestOut	O	PLL Test
PLL_RefCLK	I	Reference clock source to drive CPU clock and other internal clocks of TX7901.
PLL_SyncIn	I	External PLL Synch In signal
PLL_Sel0	I	PLL Select
PLL_Sel1	I	PLL Select
PLL_SyncOut	O	External PLL Synch Out signal
PLL_SyncExt	I	Select external synchronization mode
divRSTB	I	Reset Clock Divisor at Power up (Test only)
SDRAM Memory Interface		
SdmData[63:0]	I/O	SDRAM Memory Data Bus
SdmCB[7:0]	I/O	SDRAM Memory Check bits for ECC
SdmDQM[7:0]	O	SDRAM Data Mask
SdmAddr[12:0]	O	SDRAM Address Bus

Name of Signal	I/O	Function
SdmBAddr[1:0]	O	SDRAM Bank Address bits
SdmCSB[3:0]	O	SDRAM Chip select for each DIMM
sdmCASB	O	SDRAM CAS signal
sdmRASB	O	SDRAM RAS signal
sdmWrB	O	SDRAM Write Enable
sdmCKE	O	SDRAM Clock enable
UART0 Interface		
UA0_BAUD	O	Receive/Transmit clock derived from CLK divided by the value in the divisor latch DLL & DLM.
UA0_RCLK	I	Receive clock
UA0_RCLK_BAUD	I	RCLK Select. When tied high, RCLK is connected internally to BAUD; when tied to low, the RCLK pin is used as the Receive clock.
UA0_SIN	I	Serial Input. Data is clocked in using RCLK/16.
UA0_SOUT	O	Serial Output. Data is clocked out using the output from Baud Rate Generator, divided by 16.
UA0_DCDB	I	Data Carrier Detector, Modem Control register [7] status bit.
UA0_RIB	I	Ring Indicator, Modem Control register [6] status bit.
UA0_DSRB	I	Data Set Ready, Modem Control register [5] status bit.
UA0_CTSB	I	Clear To Send. Modem Control register [4] status bit.
UA0_OUT2B	O	General Control, Modem Control register [3] status bit.
UA0_OUT1B	O	General Control, Modem Control register [2] status bit.
UA0_RTSB	O	Request to Send, Modem Control register [1] status bit.
UA0_DTRB	O	Data Terminal Ready, Modem Control register [0] status bit.
UART1 Interface		
UA1_SIN	I	Serial Input. Data are clocked in using internal RCLK.
UA1_SOUT	O	Serial Output. Data are clocked out using the output from Baud Rate Generator, divided by 16.
PHY0 (MII Interface)		
MAC0_TXCLK	I	Transmit Nibble Clock Input
MAC0_TXD[3:0]	O	Transmit Nibble Data
MAC0_TXEN	O	Transmit Enable
MAC0_TXER	O	Transmit Error
MAC0_CRS	I	Carrier Sense
MAC0_COL	I	Collision Detected
MAC0_RXCLK	I	Receive Nibble Clock
MAC0_RXDV	I	Receive Data Valid
MAC0_RXER	I	Receive Error
MAC0_RXD[3:0]	I	Receive Nibble Data
MAC0_MDC	O	MII Management Clock
MAC0_MDIO	I/O	MII Management Data Input/Output
MAC0_HwFDupSel	I	Full Duplex Select
PHY1 (MII Interface)		
MAC1_TXCLK	I	Transmit Nibble Clock Input
MAC1_TXD[3:0]	O	Transmit Nibble Data
MAC1_TXEN	O	Transmit Enable
MAC1_TXER	O	Transmit Error
MAC1_CRS	I	Carrier Sense
MAC1_COL	I	Collision Detected
MAC1_RXCLK	I	Receive Nibble Clock
MAC1_RXDV	I	Receive Data Valid
MAC1_RXER	I	Receive Error
MAC1_RXD[3:0]	I	Receive Nibble Data

Name of Signal	I/O	Function
MAC1_MDC	O	MII Management Clock
MAC1_MDIO	I/O	MII Management Data Input/Output
MAC1_HwFDupSel	I	Full Duplex Select
Timer/Counter Interface		
TIMOUT1	O	Timer 1 Output
TIMIN1	I	External Clock Input for Timer 1
TIMOUT2	O	Timer 2 Output
TIMIN2	I	External Clock Input for Timer 2
Serial Peripheral Interface		
SPI_MISO	I	Serial Data Input
SPI_MOSI	O	Serial Data Output
SPI_Clk	O	SPI Data Clock
SPI_Port0B	O	SPI Chip select for SPI device 0
SPI_Port1B	O	SPI Chip select for SPI device 1
SPI_Port2B	O	SPI Chip select for SPI device 2
SPI_Port3B	O	SPI Chip select for SPI device 3
SPI_Port4B	O	SPI Chip select for SPI device 4
SPI_Port5B	O	SPI Chip select for SPI device 5
PCI0 Interface		
PCI0_AD[31:0]	I/O	The 64 Bit Address and Data Buses are multiplexed on the same PCI pins.
PCI0_CBEB[3:0]	I/O	Command and Byte Enable
PCI0_PAR	I/O	Parity for PCI_AD[31:0] and PCI_CBE[3:0]B Even Parity
PCI0_FRAMEB	I/O	Indicates beginning and duration of a transaction.
PCI0_TRDYB	I/O	Target Ready
PCI0_IRDYB	I/O	Initiator Ready
PCI0_STOPB	I/O	PCI_STOPB Indicates that the current Target is requesting Initiator to stop the current transaction.
PCI0_DEVSELB	I/O	Device select; it indicates that the current driving device has decoded its address as the target of the current access.
PCI0_CLK	I	PCI Clock Input
PCI0_PERRB	I/O	Data Parity Error Reporting parity error on all transactions except Special Cycle command.
PCI0_IDSEL	I	Initialization Device Select It is used as a chip select during configuration read/write transaction on PCI bus.
PCI0_SERRB	O	System Error Reporting errors for all the address parity errors and data parity error on Special Cycle commands, and may optionally be used on any other non-parity or system errors.
PCI0_REQB	O	PCI0 request signal to PCI arbiter
PCI0_GNTB	I	PCI0 bus Grant form PCI arbiter
PCI0_GNT0B	I	PCI0 bus Grant 0B
PCI0_REQ1B	O	PCI0 Request 1B
PCI0_GNT1B	I	PCI0 bus Grant 1B
PCI0_REQ2B	O	PCI0 Request 2B
PCI0_GNT2B	I	PCI0 bus grant 2B
PCI1_REQ1B_PCI0_REQ3B	O	PCI1 Request 1B Multiplex with PCI10 Request 3B
PCI1_REQ2B_PCI0_REQ4B	O	PCI1 Request 2B Multiplex with PCI10 Request 4B
PCI0_RSTB	I	PCI0 Reset signal
PCI1 Interface		
PCI1_AD[31:0]	I/O	The 64 Bit Address and Data Buses are multiplexed on the same PCI pins.
PCI1_CBEB[3:0]	I/O	Command and Byte Enable
PCI1_PAR	I/O	Parity for PCI_AD[31:0] and PCI_CBEB[3:0] Even Parity
PCI1_FRAMEB	I/O	Indicates beginning and duration of a transaction.

Name of Signal	I/O	Function
PCI1_TRDYB	I/O	Target Ready
PCI1_IRDYB	I/O	Initiator Ready
PCI1_STOPB	I/O	PCI_STOPB Indicates that the current Target is requesting Initiator to stop the current transaction.
PCI1_DEVSELB	I/O	Device select; it indicates that the current driving device has decoded its address as the target of the current access.
PCI1_CLK	I	PCI Clock Input
PCI1_PERRB	I/O	Data Parity Error Reporting parity error on all transactions except Special Cycle command.
PCI1_IDSEL	I	Initialization Device Select It is used as a chip select during configuration read/write transaction on PCI bus.
PCI1_SERRB	O	System Error Reporting errors for all the address parity errors and data parity error on Special Cycle commands, and may optionally be used on any other non-parity or system errors.
PCI1_GNTB	I	PCI1 Grant
PCI1_REQB	O	PCI1 Request
PCI1_GNT0B	I	PCI1 bus Grant 0B
PCI1_GNT1B_PCI0_GNT3B	O	PCI1 bus Grant 1B Multiplex with PCI0 bus Request 3B
PCI1_GNT2B_PCI0_GNT4B	O	PCI1 bus Grant 2B Multiplex with PCI0 bus Request 4B
PCI1_CBEB [3:0]	I/O	Command and Byte Enable
PCI1_CLK	I	PCI1 Clock Input
PCI1_RSTB	I	PCI1 Reset signal
Test/ Debug		
JTDI	I	JTAG Data In
JTCLK	I	JTAG Clock
JTDO	O	JTAG Data Out
JTMS	I	JTAG Mode select
JTRST	I	JTAG Reset
Power and Ground Signals		
PLL_AVSS	I	PLL Ground Pin
PLL_AVDD	I	PLL Voltage Pin
VDD-IO	I	VDD Voltage Pins for I/O signals
VDD-Core	I	VDD Voltage Pins for Internal core
VSS	I	Ground

Table 17-2 TX7901 Pin Positions

TBD

17.1 JTAG Boundary Scan external test chain configuration

Please contact Toshiba to request a BSDL file based on IEEE1149.1b. This file contains the required information.