Preface

Thank you for your new or continued patronage of Toshiba semiconductor products. This is the 1998 edition of the user's manual for the TMPR3922 RISC ASSP, a member of the TLCS-R3900 Family of 32-bit RISC microcontrollers.

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

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

September 1998

1 Handling Devices

1.1 Transport

Handle devices and packaging materials with care. To avoid damage to the devices, do not toss or drop packages. Ensure that devices are not subjected to mechanical vibration or shock during transporting. Also avoid getting devices or packaging wet. Moisture can adversely affect packaging by nullifying the effect of the antistatic agent.

1.2 Storage

1.2.1 General Storage

(1) Avoid storage locations where devices will be exposed to moisture or direct sunlight. (Be especially careful during periods of rain or snow.)

(2) Do not place device cartons upside down. Stack cartons on top of one another in an upright position only; do not place cartons on their sides.

(3) The storage area temperature should be kept within a temperature range of 5 to 35°C, and relative humidity should be maintained between 40 and 75%.

(4) Do not store devices in the presence of harmful (especially corrosive) gases, or in dusty conditions.

(5) 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.

(6) When repacking devices, use antistatic containers.

(7) Do not allow external forces or loads to be applied to devices while they are in storage.

(8) If devices have been stored for more than two years, they should be tested for solderability and electrical characteristics before use.

1.2.2 Moisture-proof Packing

(1) Do not drop or toss device packing. The laminated aluminum material in it can be rendered ineffective by rough handling.

(2) Ensure that packing materials are stored in a 30°C, 90% RH environment. Use devices within 12 months; do not store them for longer.

(3) Some devices have silica gel granules in their packing that indicate whether or not moisture penetration has occurred. If, when the pack is opened, the normally blue-colored silica gel granules have become transparent, this means that the laminated aluminum packing material has been damaged and that moisture has been absorbed. In this case, depending on the device and packing types, it may be advisable to bake the devices at high temperature to remove the moisture. See section 1.2.2(5) below.

(4) If the 30% humidity indicator shown in Figure 1.1.1 is pink when the packing is opened, depending on the device and packing types, it may be advisable to bake the devices at high temperature to remove any moisture. See section 1.2.2(5) below. It may also be advisable to bake the devices, whether still packed or not, if the effective usage period of the indicator has expired. After the pack is opened, use the device in a 30°C and 90% RH environment, and within the effective usage period which is listed on each moisture-proof package.

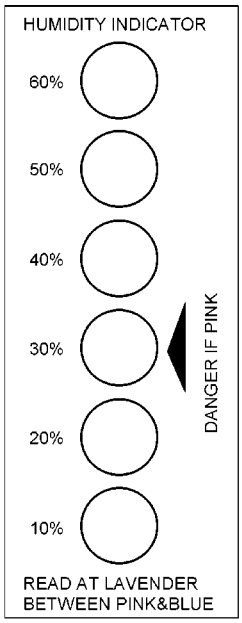
(5) The following describes high-temperature treatments for the various packing types. Contact Toshiba or a Toshiba distributor for more information.

(a) Tray : If the tray is heatproof, bake at 125°C for 20 hours (heatproof trays bear a "Heat Proof" marking). Bake non-heatproof trays at 70°C for 168 hours.

(b) Tube : Tubes are not heatproof. Transfer devices to heatproof trays or aluminum tubes before baking at 125°C for 20 hours.

(c) Tape : Packing that includes adhesive or embossed tape cannot be baked. Devices so packed must be used within their allowable time limits after unpacking, as specified on the packing.

For surface-mount or insertion mounting devices, be careful not to bend the leads when baking.



The indicator shown on the left detects an approximate level of ambient humidity at a standard temperature of 25°C. All humidity indicators are blue when the ambient humidity is below 10%.
If the 30% humidity indicator is found to be entirely pink when unpacked, treat the devices at high temperature as described in section 1.2.2(5) above, to remove moisture before using.

Figure 1.1.1 Humidity Indicator

1.3 Inspection (on Receipt and during Processing)

1.3.1 Ground

(1) When static electricity is stated to be a potential problem for the device, handle it with particular care during inspection, as follows. Ensure that flooring, workbenches, conveyance mechanisms and mats are all well grounded to earth to prevent the build-up of electrostatic charge. Use earth-grounded antistatic mats (100 KΩ to 100 MΩ/cm^2), especially for workbenches or flooring that come into direct contact with devices.

(2) Ground all measuring instruments, jigs, tools and soldering irons to earth.

(3) Ensure that operators wear antistatic clothing. Operators should also wear grounding rings or bands that transfer electrostatic charge from the body to earth via an 0.5 to 1 MΩ resistor.

1.3.2 Electrical Leakage

Electrical leakage in equipment used to inspect devices, or in systems in which devices are installed, is dangerous to personnel and can cause electrical breakdown of the devices. Ensure that there is no such leakage in testers, curve tracers, synchroscopes or other measuring instruments, or in equipment that applies solder to devices. Ensure that all such equipment is grounded to earth.

1.3.3 Inspection Sequence

(1) Before beginning device inspection, make a final check to ensure that all involved equipment is well grounded to earth and that there is no electrical leakage as described above. Apply voltage to the test jig only after inserting the device securely into it. (Do not power the test jig up or down abruptly; always apply or remove power gradually or in steps.)

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

1.3.4 Electrical Shock

A device undergoing electrical measurement poses a danger of electrical shock. Do not touch the device unless you are sure that power to the measuring instrument is off.

1.4 Mounting

Broadly classified, there are two main types of semiconductor device packages: lead insertion and surface mount. During mounting onto circuit boards, devices can become contaminated by flux or damaged by thermal stress from the soldering process. Particularly with surface mount devices, 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 that you should take when mounting devices on circuit boards. Note, however, that the appropriate mounting methods differ depending on the chip size and frame design, even for devices with the same package type. Therefore, please contact Toshiba or a Toshiba distributor for details concerning the best mounting method for each device type.

1.4.1 Temperature Profiles

(1) When Using a Soldering Iron

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

(2) When Using Long or Medium Infrared Ray Reflow

(a) Heating top and bottom with long or medium infrared rays is recommended (see Figure 1.1.2).

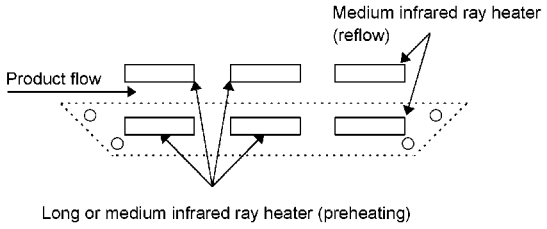


Figure 1.1.2 Heating Top and Bottom with Long or Medium Infrared Rays

(b) Complete the infrared ray reflow process within 30 seconds at a package surface temperature between 210°C and 240°C.

(c) Short infrared ray reflow soldering produces thermal stress equivalent to that of dip soldering, so take care when using this method.

(d) Refer to Figure 1.1.3 for an example of a good temperature profile for infrared or hot air reflow.

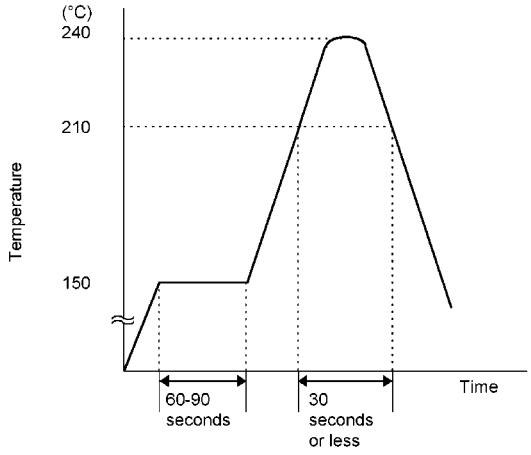


Figure 1.1.3 Example Temperature Profile for Infrared or Hot Air Reflow

(3) When Using Hot Air Reflow

(a) Complete hot air reflow within 30 seconds at a package surface temperature of between 210°C and 240°C.

(b) For an example of a recommended temperature profile, refer to Figure 1.1.3.

(4) When Using Vapor Phase Reflow

(a) The recommended solvent is Fluorinate FC-70 or equivalent.

(b) Complete hot air reflow within 30 seconds at an ambient atmospheric temperature of 215°C, or within 60 seconds at an ambient atmospheric temperature of 200°C.

(c) Refer to Figure 1.1.4 for an example of a good temperature profile for vapor phase reflow soldering.

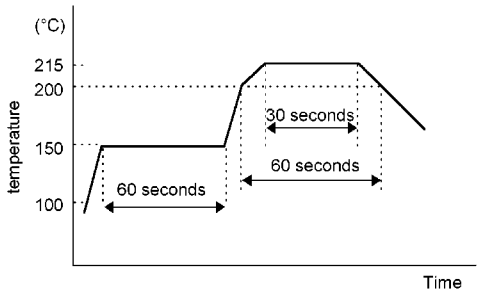


Figure 1.1.4 Example Temperature Profile for Vapor Phase Reflow

(5) When Using Dip Soldering (Recommended Temperature Profiles are Different for each Product)

(a) Apply preheating of 150°C to the circuit board and devices for 60 seconds or more.

(b) Complete soldering within ten seconds at a solder flow temperature of up to 260°C.

1.4.2 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 that can degrade device performance.

Conventional cleaning solvents that contain freon are not recommended due to the danger that they pose to the earth's ozone layer. Alternative products are available on the market. Some alternative cleaning agents that do not contain freon include:

- FRW-1, 17; FRV-100 from Toshiba Corporation
- AK-225AES from Asahi Glass Co., Ltd.
- 750H from Kao Co., Ltd.
- ST-100 from Arakawa Chemical Co., Ltd.

(2) 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.

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

(4) Ultrasonic cleaning can clean circuit boards efficiently in a short period of time. However, it should not be used with hermetically-sealed ceramic packages such as a Leadless Chip Carrier (LCC) or Pin Grid Array (PGA), because their connecting wires can become disconnected due to resonance during the cleaning process.

This is not a problem with plastic packages. However, adherence to the following ultrasonic cleaning conditions is recommended:

Frequency : 27 to 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.

Contact Toshiba or a Toshiba distributor regarding cleaning conditions and other relavant information for each product type.

1.4.3 Lead Forming

Semiconductor devices must undergo a process in which the leads are cut and formed before the devices are mounted on printed circuit boards. If undue stress is applied to device interiors during this process, mechanical breakdown or degraded reliability can result. This is attributable primarily to differences in stress experienced by the device's external and 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 precisely match the device lead pitch.

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

(3) Leave a gap between the device bottom surfaces and the circuit board by inserting spacers or by forming the leads appropriately.

(4) Do not repeatedly bend or stretch device leads.

(5) Some devices have sharp, pointed leads to facilitate mounting. Be careful not ot injure yourself when handling such devices with your bare hands.

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

(a) Use a tool or jig to secure the lead at its base (where the lead meets the device package) while bending.

(b) Maintain a certain distance between the device package and the tool or jig.

(c) When forming a lead by bending it over a jig surface, be careful not to damage the lead on the edge of the jig surface.

(d) Follow other precautions described in the individual standards for each device and package type.

(7) Be careful that debris from the cut leads does not cause damage or inflict injury (for example, by getting into your eyes).

1.4.4 Circuit Board Coating

When devices are to be used in equipment requiring high 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.

1.4.5 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) 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.

(3) 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 nonvolatile compound, as volatile compounds can crack after a time, causing the heat radiation properties to deteriorate.

(4) 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 and G746 from Shinetsu Chemical Industries.

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

1.4.6 Fastening Torque

(1) Do not exceed the specified fastening torque for screw-mounted devices.

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

(3) Observe precautions (1) and (2) above with particular care when mounting devies on circuit boards or heat sinks. While doing so, also observe the precautions concerning grounding (see Section 1.3.1), electrical leakage (Section 1.3.2), electrical shock (Section 1.3.4), soldering (Section 1.4.1) and load forming (Section 1.4.3).

1.5 Electrostatic Discharge

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

1.5.1 Work Environment

(1) 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.

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

(3) 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 $10^8\Omega$/sq and the resistance between surface and ground, 7.5×10^5 to $10^8\Omega$.

(4) Cover the workbench surface also with a conductive mat (with a surface resistivity of 10^4 to $10^8\Omega$ / sq, for a resistance between surface and ground of 7.5×10^5 to $10^8\Omega$). 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.

(5) 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 at 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 that come into contact with device lead terminals, use a material that 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 a tape carrier which come into contact with automatic 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 sustain an electrostatic charge, use an ionizer to neutralize the ions.

(6) 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.

(7) Keep track of charged potential in the working area by taking periodic measurements.

(8) Ensure that work chairs are protected by an antistatic 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 $10^{12}\Omega$/sq.)

(9) Install antistatic mats on storage shelf surfaces. (Suggested surface resistivity is 10^4 to $10^8\Omega$/sq; suggested resistance between surface and ground is 7.5×10^5 to $10^8\ \Omega$/sq.)

(10) For transport and temporary storage of devices, use containers (boxes, jigs, bags) that are made of antistatic materials or of materials that dissipate electrostatic electricity.

(11) Make sure that cart surfaces which come into contact with device packaging are made of materials that will conduct static electricity, and check to verify that they are grounded to the floor surface with a grounding chain. (The suggested resistance between the cart surface and grounding chain is 7.5×10^5 to $10^{10}\ \Omega$/sq.)

(12) Install dedicated grounding wire in static electricity control areas. Use grounding wire of class 3 or above. If possible, use different grounds for machine chassis and equipment which may touch devices.

1.5.2 Operating Environment

(1) Operators must wear antistatic clothing and conductive shoes (or a leg or heel strap).

(2) Operators must wear a wrist strap grounded to earth via a resistor of about 1 MΩ.

(3) Soldering irons must be grounded from iron tip to earth, and must be used only at low voltage (6 V to 24 V).

(4) If the tweezers you use are likely to touch the device terminals, use antistatic 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 antistatic purposes (suggested resistance value: 10^4 to $10^8\Omega$).

(5) Do not place devices or their containers near sources of strong electrical fields (such as above a CRT).

(6) When storing circuit boards that have devices mounted on them, use a board container or bag that is protected against static charge. Keep the boards separate from one other, and do not stack them directly on top of one another, lest static charge or discharge occur due to friction.

(7) Ensure that articles (such as clipboards) that are brought into static electricity control areas are constructed of antistatic materials if possible.

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

(9) Equipment safety covers installed near devices should have resistance ratings of $10^9\ \Omega$ or less.

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

1.6 Disposal Precautions

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

2 Protecting Devices in the Field

2.1 Temperature

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

2.2 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 that 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 electrostatic discharge. Unless damp-proofing measures have been specifically carried out, use devices only in an appropriate ambient moisture environment (that is in a relative humidity range of 40 to 60%).

2.3 Corrosive Gases

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

2.4 Radioactive and Cosmic Rays

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

2.5 Strong Electric and Magnetic Fields

Devices exposed to strong magnetic fields can undergo a polarization phenomenon in plastic material, or within the IC chip, that 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 mounting location has to be changed or the device has to 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.

2.6 Vibration, Shock and Stress

Ceramic package devices, and devices in canister-type packages that have empty spaces internally, are subject to damage from vibration and shock because bonding wires are secured only at the ends. Plastic molded devices, on the other hand, have relatively strong resistance to vibration and mechanical shock because the wires are encased and fixed in resin. However, when any device or package type is installed in equipment, it is to some extent susceptible to wiring disconnections and other damage from vibration, shock and stressed solder junctions. When using devices in vibration- prone equipment therefore, the structural design of the equipment must be considered carefully.

Furthermore, it is generally known that stress applied to a semiconductor chip through the package can change the resistance characteristics within the chip due to piezo effects. In analog circuit design, attention must be paid to the problem of package stress as well as to vibration and shock as described above. If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack.

2.7 Interference from Light

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 devices are protected against incident light from external sources. This problem is not limited only to optical semiconductors and EPROMs. All types of devices can be affected by light.

2.8 Dust and Oil

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

2.9 Fire

Semiconductor devices may be 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 inflammable or combustible materials.

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 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, voltage and current surges, as well as mounting conditions that affect device reliability. This section describes some general precautions that you should observe when designing circuits and mounting devices on circuit boards.
For more detailed information about each product family, refer to the relevant individual databooks, available from Toshiba.

3.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. If absolute maximum ratings are stipulated for two or more parameters, no two can be applied to the device simultaneously. 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, storage and lead 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 soldering temperatures exceed rated values, the package seal can deteriorate or the wire bonds can open due to the differences between the thermal expansion coefficients of the materials of which the device is constructed.

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 data sheet.
If greater reliability is required, derate the device absolute maximum ratings for voltage, current, power and temperature before using the device. (For more information about derating, refer to the specific databook for the device being used.)

3.3 Unused Terminals

Some devices can exhibit unstable input problems if unused pins are left open. Similarly, care must be taken not to connect the output pins of a device to the power supply pin (Vdd) or to other output pins. For details concerning the handling of unused pins, follow the procedures described in the appropriate technical data sheet or databook for the device being used. 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, both P-channel and N-channel transistors may be turned on, allowing unwanted supply current to flow. Therefore, ensure that the unused input gates of a device are connected to the power supply (Vdd) pin or ground (GND) pin of the same device. For details about how to handle the pins of heat sinks, consult Toshiba or a Toshiba distributor.

3.4 Latch-up

Latch-up is an abnormal condition inherent in CMOS devices, in which Vdd 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 on the order of several hundred mA or more to flow between Vdd and GND, eventually causing the device to break down. Latch-up occurs when the input / output voltage exceeds the rated value, causing a large current to flow in the internal chip, or when the voltage on the 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 Vdd and GND. This causes heating and, in extreme cases, gas fumes as well. To prevent this problem, observe the following precautions:

(1) Do not allow voltage levels on the input / output pins either to rise above Vdd or to fall below Vss. Also, be sure to consider the power-on timing, so that power is applied gradually or in steps rather than abruptly.

(2) Allow no abnormal noise signals to be applied to the device.

(3) Fix the voltage levels of unused input pins to Vdd or Vss.

(4) Do not connect outputs to one another.

3.5 Input / Output Protection

Wire-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 Vdd or GND.
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.

3.6 Load Capacitance

Some devices display increased delay times if there is a large load capacitance. 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.

3.7 Thermal Design

The failure rate of semiconductor devices is greatly increased as operating temperatures increase. As shown in Figure 1.3.1, 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, observe the following precautions concerning thermal design to achieve optimum reliability:

(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 maximum ratings to minimize thermal stress from power dissipation.

$\theta ja = \theta jc + \theta ca$
$\theta ja = (Tj - Ta) / W$
$\theta jc = (Tj - Tc) / W$
$\theta ca = (Tc - Ta) / W$

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)
- W = power dissipation (W)

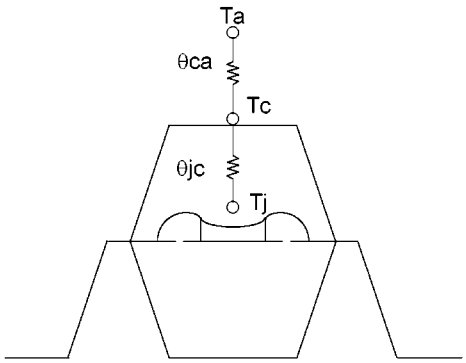


Figure 1.3.1 Thermal Resistance of Package

3.8 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 on how to interface each device, consult the appropriate technical data sheets and databooks. If you have any questions or doubts about interfacing, contact your nearest Toshiba office or distributor.

3.9 Decoupling

Spike currents generated during switching can cause Vdd and GND 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 to 100Ω.) 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 Vdd and GND lines and by installing decoupling capacitors (of approximately 0.01 to 1 μF capacitance) as high-frequency filters between Vdd and GND at strategic locations on the circuit board.

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

An important note about wiring is that, in the case of high-speed logic ICs, noise is caused mainly by reflection and crosstalk, or by power supply common impedance. Reflections cause increased signal delay, ringing, overshoot and undershoot, thereby reducing the device's safety margin 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. Care must be taken, however, because this corrective measure tends to cause crosstalk between wiring components. In practice, there must be a trade-off between these two factors.

3.10 External Noise

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

For details of the appropriate protective measures to be taken for a particular device, consult the databook for that product.

3.11 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 that 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 that allows designers to calculate, at the design stage, the strength of 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 interfering electromagnetic waves.
Yet it is possible during system design to incorporate some measures for the prevention of electromagnetic interference, that can facilitate taking corrective measures once the design is completed. These include installing shields and noise filters, and increasing the thickness of power supply wiring patterns on the circuit board. One effective method, for example, is to devise several shielding alternatives during design, then select the most suitable shielding method based on the results of measurements taken after the prototype has been completed.

3.12 Safety Standards

Each country has safety standards which must be observed. For example, for devices that handle high voltages, it is often required that an appropriate insulation distance be maintained between the device proper and the circuit board conductor pattern. Such requirements must be fully taken into account to ensure that your design conforms to the applicable safety standards.

3.13 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 memory chips and microcomputers, caution is required when powering on or resetting the device. Consult the appropriate databook for each device used, to ensure that your design conforms to specifications.

(4) Some packages for some products (especially power devices) will get hot during operation. Be careful not to touch such devices directly.

(5) Be careful that no conductive material or object (such as a metal pin) drops onto and shorts the leads of a device mounted on the circuit board.

4 Product-Specific Precautions

4.1 Using Resonators not Listed under "Recommended Types"

Resonators recommended for use with Toshiba products in microcontroller oscillator applications are listed in the 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 about the suitability of the device for your application.

4.2 Undefined Functions

Depending on the microcontroller type, some instruction code values do not constitute valid processor instructions. Similarly, the presence of undefined bits in registers is possible. Be careful in your applications not to use undefined instructions or to cause register bits to be undefined.

4.3 Injuries from Probe Tips

Some devices have sharply pointed leads. Be careful not to injure yourself with these types of device.

SECTION 1 TMPR3922 Overview

1.1 Overview

The TMPR3922 is the single-chip, integrated digital ASSP for the Personal Information Communicator (PIC). Figure 1-1 shows a block diagram of the overall PIC system. The TMPR3922 consists of the PIC system support logic, integrated with an embedded TX3920 Processor core designed by TOSHIBA.

The TMPR3922 consists of a TX3920 Processor core with 16KBytes of instruction cache memory and 8KByte of data cache memory, plus integrated functions for interfacing to numerous system components and external I/O modules. The TX3920 Processor core is also augmented with a single-cycle multiply/accumulate module to allow integrated DSP functions, such as a software modem for high-performance standard data and fax protocols. The TMPR3922 also contains multiple DMA channels and a high-performance and flexible Bus Interface Unit (BIU) for providing an efficient means for transferring data between external system memory, cache memory, the Processor core, and external I/O modules. The types of external memory devices supported include dynamic random access memory (DRAM), synchronous dynamic random access memory (SDRAM), static random access memory (SRAM), Flash memory, read-only memory (ROM), and expansion cards (PCMCIA and so on). The TMPR3922 also contains a System Interface Module (SIM) containing integrated functions for interfacing to numerous external I/O modules such as the TC35143F (which handles most of the analog functions of the system, including sound and telecom codecs and touchscreen ADC), ISDN/high-speed serial, infrared, wireless peripherals, etc. Lastly, The TMPR3922 contains support for implementation of power management for PIC system, whereby various TMPR3922 internal modules and external subsystems can be individually (under software control) powered up and down. The TMPR3922 contains the following overall features:

- high level of integration on a single chip, small board space, low pin count, low power, and high performance
  - LQFP208-pin package
  - 32-bit TX3920 Processor core, cache memory, multiply-accumulate module, multi-channel DMA controller, bus interface unit and memory controller, power management, and other peripheral subsystems all on a single integrated chip
  - minimal number of inter-chip connections
  - operation frequency: 166/150/133 MHz

- low power consumption
  - typ. active current at maximum operation frequency (166MHz) = 200 mA
  - typ. standby current=100 μA
  - entire TMPR3922 operation is 3.3V(I/O)/2.5V(Internal)
  - real-time clock based on 32.768 kHz reference
  - Processor core clock stop state for low standby current
  - power-down modes for individual internal peripheral modules

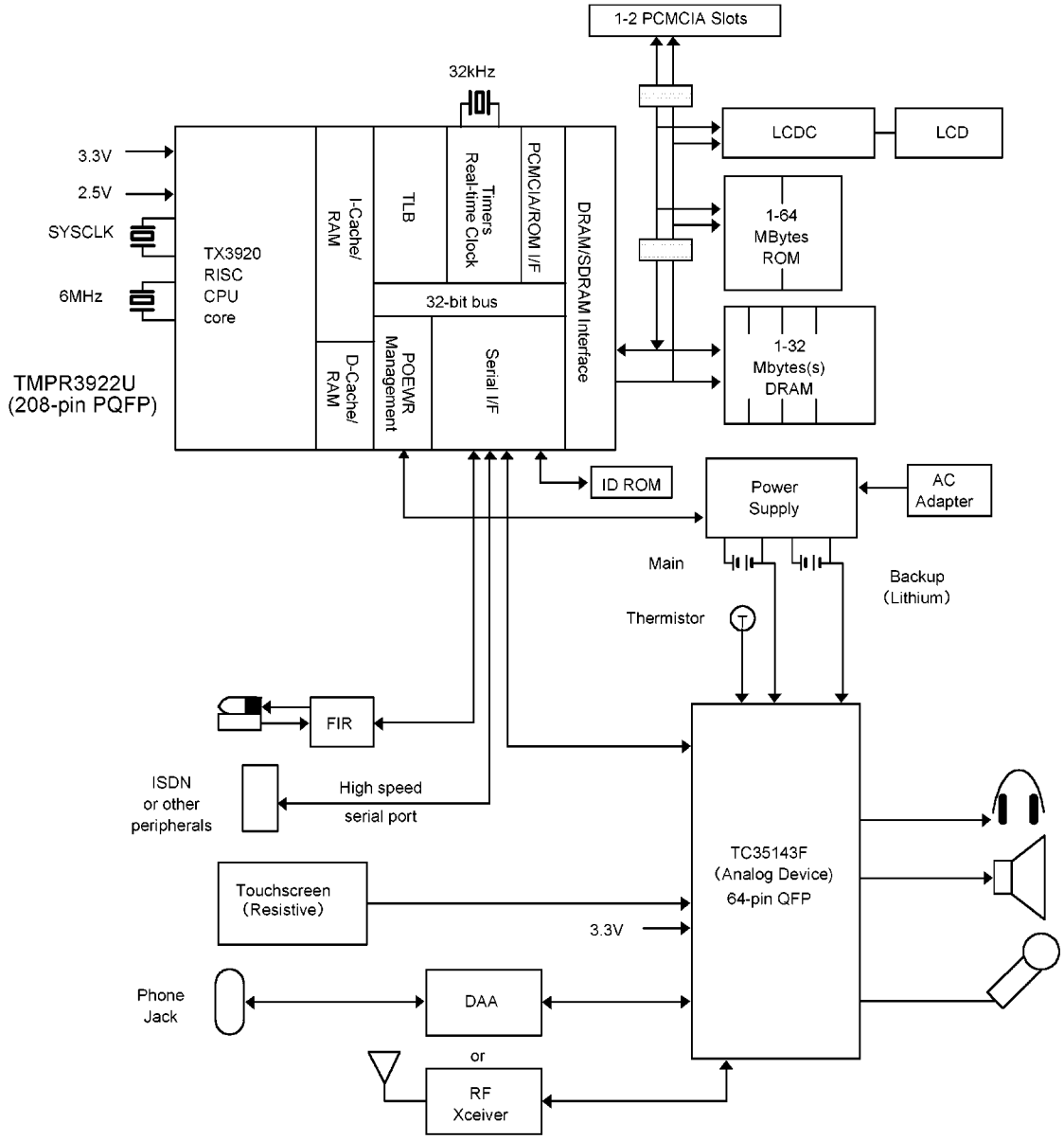


Figure 1-1 PIC Block Diagram

Figure 1-2 shows a block diagram of the TMPR3922. The key functions and features for each TMPR3922 module are itemized below.

Processor core Module

- TX3920 Processor core
  - full 32-bit operation (registers, instructions, addresses, etc.)
  - 32 general purpose 32-bit registers; 32-bit program counter
  - MIPS RISC Instruction Set Architecture (ISA) supported
  - Translation Look-aside Buffer (TLB) (4K/16K/64K/256K/1M/4M Page size, 64 entries)
- on-chip cache
  - 16 KByte instruction cache (I-cache)
    - physical address tag and valid bit per cache line
    - programmable burst size
    - instruction streaming mode supported
    - two-way set associative
  - 8 KByte data cache (D-cache)
    - physical address tag and valid bit per cache line
    - programmable burst size(write-through mode)
    - write-back/write-through
    - two-way set associative
  - cache address snoop mode supported for DMA(write-through mode)
- high-speed multiplier/accumulator
  - on-chip hardware multiplier
  - supports 32 × 32 multiplier operations, with 64-bit accumulator
  - existing multiply instructions are enhanced and new multiply and add instructions are added to the R3000A instruction set to improve the performance of DSP applications
- Processor core interface
  - handles data bus, address bus, and control interface between Processor core and rest of the TMPR3922 logic

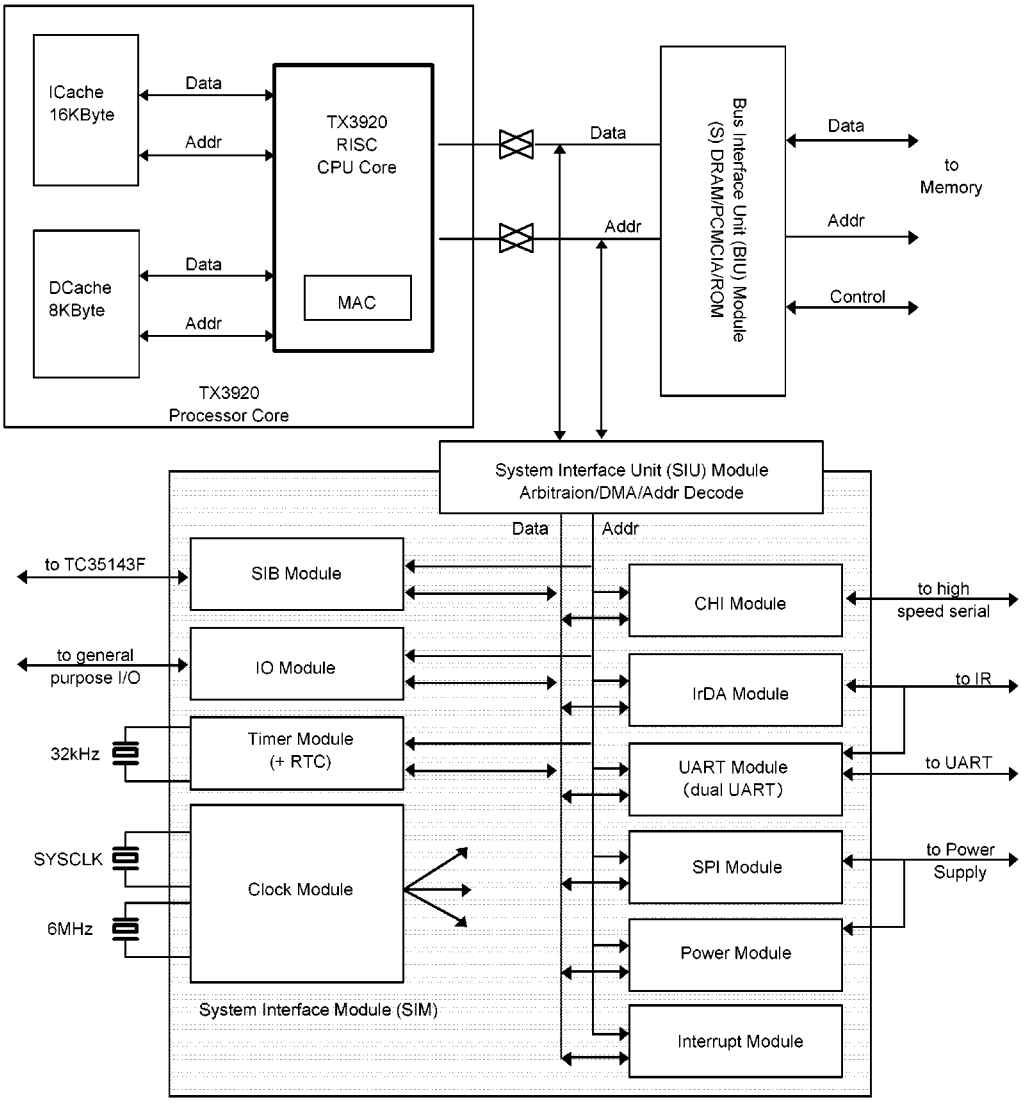


Figure 1-2 TMPR3922 Block Diagram

BIU Module

- system memory and the TMPR3922 Bus Interface Unit (BIU)
  - supports up to 2 banks of physical memory
  - supports self-refreshing DRAM and SDRAM
  - programmable parameters for each bank of DRAM or SDRAM (row/column address configuration, refresh, burst modes, etc.)
- programmable chip select memory access
  - 4 programmable (size, wait states, burst mode control) memory device and general purpose chip selects
    - available for system ROM, SRAM, Flash
    - available for external port expansion registers
  - 2 programmable (size, wait states, burst mode control) general purpose chip selects
    - supports a WAIT input
- supports up to 2 identical full PCMCIA ports
  - the TMPR3922 provide the control signals and accepts the status signals which conform to the PCMCIA version 2.01 standard
  - appropriate connector keying and level-shifting buffers required for 3.3V versus 5V PCMCIA interface implementations

SIU Module

- multi-channel 32-bit DMA controller and System Interface Unit (SIU)
- independent DMA channels for SIB to/from TC35143F audio/telecom codecs, high-speed serial port, IrDA, and general purpose UART
- address decoding for submodules within System Interface Module (SIM)

Clock Module

- the TMPR3922 supports system-wide single crystal configuration, besides the 32 kHz RTC XTAL (reduces cost, power, and board space)
- common crystal rate divided to generate clock for CPU, sound, telecom, UARTs, etc.
- independent enabling or disabling of individual clocks under software control, for power management

CHI Module

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

Interrupt Module

- contains logic for individually enabling, reading, and clearing all TMPR3922 interrupt sources
- interrupts generated from internal TMPR3922 modules or from edge transitions on external signal pins

IO Module

- contains support for reading and writing the 16 bi-directional general purpose IO pins and the 32 bi-directional multi-function IO pins
- each IO port can generate a separate positive and negative edge interrupt
- independently configurable IO ports allow the TMPR3922 to support a flexible and wide range of system applications and configurations

IR Module

- IR consumer mode
  - allows control of consumer electronic devices such as stereos, TVs, VCRs, etc.
  - programmable pulse parameters
  - external analog LED circuitry
- IrDA communication mode
  - IrDA 1.0 mode with filter is supported(BOF and EOF are detected by hardware and the bit pattern which data translation is necessary is detected and translated by hardware.)
  - Also IrDA 1.1 compliance(2.4/9.6/19.2/38.4/57.6/115.2 kbps are available)
  - 1.152 Mbps NRZ supported
  - 4 Mbps 4ppm/single plus supported(512 kbps and 4Mbps with double pulse are not supported)
  - CRC generation/check supported
  - Address filter mode supported
  - Power down mode (Power down register controls FIR clock to reduce power)
  - supported by the UART module within the TMPR3922
  - external analog receiver preamp and LED circuitry
  - data rate = up to 115 kbps at 1 meter
- IR FSK communication mode
  - supported by the UART module within the TMPR3922
  - external analog IR chip(s) perform frequency modulation to generate the desired IR communication mode protocol
  - data rate = up to 36000 bps at 3 meters
- carrier detect state machine
  - periodically enables IR receiver to check if a valid carrier is present

Power Module

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

SIB Module

- the TMPR3922 contains holding and shift registers to support the serial interface to the TC35143F and/or other optional codec devices
- interface compatible with slave mode 3 of the Crystal CS4216 codec
- synchronous, frame-based protocol
- the TMPR3922 always master source of clock and frame frequency and phase ; programmable clock frequency
- each SIB frame consists of 128 clock cycles, further divided into 2 subframes or words of 64 bits each (supports up to 2 devices simultaneously)
- independent DMA support for audio receive and transmit, telecom receive and transmit
- supports 8-bit or 16-bit mono telecom formats
- supports 8-bit or 16-bit mono or stereo audio formats
- independently programmable audio and telecom sample rates
- CPU read/write registers for subframe control and status

SPI Module

- provides interface to SPI peripherals and devices
- full-duplex, synchronous serial data transfers (data in, data out, and clock signals)
- the TMPR3922 supplies dedicated chip select and interrupt for an SPI interface serial power supply
- 8-bit or 16-bit data word lengths for the SPI interface
- programmable SPI baud rate

Timer Module

- Real Time Clock (RTC) and Timer
- 43-bit counter (30.517 μs granularity); maximum uninterrupted time = 3104days
- 43-bit alarm register (30.517 μs granularity)
- 16-bit periodic timer (0.868 μs granularity); maximum timeout = 56.8 ms
- interrupts on alarm, timer, and prior to RTC roll-over

UART Module

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

1.2 References

(1) MIPS RISC Architecture, Gerry Kane and Joe Heinrich, Prentice Hall, 1992

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

(2) PC Card Standard, Release 2.01, Personal Computer Memory Card International Association, November 1992

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

(3) TX39 Family Users Manual, TOSHIBA

SECTION 2 Pin Descriptions

2.1 Overview

The TMPR3922 contains 208 pins consisting of input, output, bi-directional, and power and ground pins. These pins are used to support various functions. The following sections will describe the function of each pin including any special power-down considerations for each pin.

2.2 Pins

The TMPR3922 contains 208 total pins, consisting of 143 signal pins, 31 power pins, and 32 ground pins. Of the 143 signal pins, 32 of them are multi-function and can be independently programmed either as IO ports or for an alternate standard/normal function. As an IO port, any of these pins can be programmed as an input or output port, with the capability of generating a separate positive and negative edge interrupt. See Section 2.3 for a summary of the multi-function IO ports versus their standard functions.

2.2.1 Memory Pins

D[31:0]: INPUT/OUTPUT

These pins are the data bus for the system. 16-bit SDRAMs and DRAMs should be connected to bits 31 : 16. All other 16-bit ports should be connected to bits 15 : 0. Of course, 32-bit ports should be connected to bits 31 : 0. These pins are normally outputs and only become inputs during reads, thus no resistors are required since the bus will only float for a short period of time during bus turn-around.

A[12:0]: OUTPUT

These pins are the address bus for the system. The address lines are multiplexed and can be connected directly to SDRAM and DRAM devices. To generate the full 26-bit address for static devices, an external latch must be used to latch the signals using the ALE signal. For static devices, address bits 25 : 13 are provided by the external latch and address bits 12 : 0 (directly connected from the TMPR3922's address bus) are held afterward by the TMPR3922 for the remainder of the address bus cycle.

ALE: OUTPUT

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

RD*: OUTPUT

This pin is used as the read signal for static devices. This signal is asserted for reads from MCS1-0*, CS3-0*, CARD2CS* and CARD1CS* for memory and attribute space.

WE*: OUTPUT

This pin is used as the write signal for the system. This signal is asserted for writes to MCS1-0*, CS3-0*, CARD2CS* and CARD1CS* for memory and attribute space, and for writes to DRAM and SDRAM.

CAS3* (WE3*): OUTPUT

This pin is used as the CAS signal for SDRAMs, the CAS signal for D[31:24] for DRAMs, and the write enable signal for D[31:24] for static devices. If the ENMCS* BE bit in the Memory Configuration 1 Register is set, this pin is used as the byte enable signal for D[31:24] for the MCS static devices.

CAS2* (WE2*): OUTPUT

This pin is used as the CAS signal for D[23:16] for DRAMs and the write enable signal for D[23:16] for static devices. If the ENMCS* BE bit in the Memory Configuration 1 Register is set, this pin is used as the byte enable signal for D[23:16] for the MCS static devices. And this pin is also used as BS1 signal for 4-Bank SDRAM configuration.

CAS1* (WE1*): OUTPUT

This pin is used as the CAS signal for D[15:8] for DRAMs and the write enable signal for D[15:8] for static devices and the lower data mask for a 32-bit SDRAM configuration. If the ENMCS* BE bit in the Memory Configuration 1 Register is set, this pin is used as the byte enable signal for D[15:8] for the MCS static devices.

CAS0* (WE0*): OUTPUT

This pin is used as the CAS signal for D[7:0] for DRAMs and the write enable signal for D[7:0] for static devices and the upper data mask for a 32-bit SDRAM configuration. If the ENMCS* BE bit in the Memory Configuration 1 Register is set, this pin is used as the byte enable signal for D[7:0] for the MCS static devices.

RAS0*: OUTPUT

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

RAS1* (DCS1*): OUTPUT

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

DCS0*: OUTPUT

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

DCKE : OUTPUT

This pin is used as the clock enable for SDRAMs.

DCLKIN: INPUT

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

DCLKOUT: OUTPUT

This pin is the (nominal) 73.728MHz clock for the SDRAMs.

DQMH: OUTPUT

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

DQML: OUTPUT

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

CS3-0*: OUTPUT

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

MCS1-0*: OUTPUT

These pins are the Chip Select 1 through 0 signals for the external device. They can be configured to support either 32-bit or 16-bit ports.

CARD2CSH*, L*: OUTPUT

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

CARD1CSH*, L*: OUTPUT

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

CARDREG*: OUTPUT

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

CARDIORD*: OUTPUT

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

CARDIOWR*: OUTPUT

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

CARDDIR*: OUTPUT

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

CARD2WAIT*: INPUT

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

CARD1WAIT*: INPUT

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

MCS1WAIT*: INPUT

This pin is the wait signal from the external device 1.

MCS0WAIT*: INPUT

This pin is the wait signal from the external device 0.

2.2.2 Bus Arbitration Pins

DREQ*: INPUT

This pin is used to request external arbitration. If the TESTAIU signal is high and the TESTAIU function has been enabled, then once DGRNT* is asserted, external logic can initiate reads or writes to the TMPR3922 registers by driving the appropriate input signals. If the TESTAIU signal is low or the TESTAIU function has not been enabled, then the TMPR3922 memory transactions are halted and certain memory signals will be tri-stated when DGRNT* is asserted in order to allow an external master to access memory.

DGRNT*: OUTPUT

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

2.2.3 Clock Pins

SYSCLKIN: INPUT

This pin should be connected along with SYSCLKOUT to an external crystal which is the main TMPR3922 clock source.

SYSCLKOUT: OUTPUT

This pin should be connected along with SYSCLKIN to an external crystal which is the main TMPR3922 clock source.

C32KIN: INPUT

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

C32KOUT: OUTPUT

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

C6MIN: INPUT

This pin should be connected along with C6MOUT to an external crystal which is the IrDA clock source inside the TMPR3922.

C6MOUT: OUTPUT

This pin should be connected along with C6MIN to an external crystal which is the IrDA clock source inside the TMPR3922.

C48MOUT: OUTPUT

This pin is a buffered output of the 48MHz clock.

BC32K: OUTPUT

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

BCLK: OUTPUT

This pin is a reference clock for the external devices and is derived by dividing down from the TMPR3922 system clock FREECLK.

2.2.4 CHI Pins

CHIFS: INPUT/OUTPUT

This pin is the CHI frame synchronization signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3922 to be the master CHI sync source. As an input, this pin allows an external peripheral to be the master CHI sync source and the TMPR3922 CHI module will slave to this external sync.

CHICLK: INPUT/OUTPUT

This pin is the CHI clock signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3922 to be the master CHI clock source. As an input, this pin allows an external peripheral to be the master CHI clock source and the TMPR3922 CHI module will slave to this external clock.

CHIDOUT: OUTPUT

This pin is the CHI serial data output signal.

CHIDIN: INPUT

This pin is the CHI serial data input signal.

2.2.5 IO Pins

IO[15:0]: INPUT/OUTPUT

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

2.2.6 Reset Pins

CPURES*: INPUT

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

PON*: INPUT

This pin serves as the Power On Reset signal for the TMPR3922. This signal must remain low when VSTANDBY is asserted until VSTANDBY is stable. Once VSTANDBY is asserted, this signal should never go low unless all power is lost in the system.

2.2.7 Power Supply Pins

ONBUTN: INPUT

This pin is used as the On Button for the system. Asserting this signal will cause PWRCS to set to indicate to the System Power Supply to turn power on to the system. PWRCS will not assert if the PWROK signal is low.

PWRCS: OUTPUT

This pin is used as the chip select for the System Power Supply. When the system is off, the assertion of this signal will cause the System Power Supply to turn VCCDRAM and VCC3 on to power up the system. The Power Supply will latch SPI commands on the falling edge of PWRCS.

PWROK: INPUT

This pin provides a status from the System Power Supply that there is a good source of power in the system. This signal typically will be asserted if there is a Battery Charger supplying current or if the Main Battery is good and the Battery Door is closed. If PWROK is low when the system is powered off, PWRCS will not assert as a result of the user pressing the ONBUTN or an interrupt attempting to wake up the system. If the device is on when the PWROK signal goes low, the software will immediately shut down the system since power is about to be lost. When PWROK goes low, there must be ample warning so that the software can shut down the system before power is actually lost.

PWRINT: INPUT

This pin is used by the System Power Supply to alert the software that some status has changed in the System Power Supply and the software should read the status from the System Power Supply to find out what has changed. These will be low priority events, unlike the PWROK status, which is a high priority emergency case.

VCC3: INPUT

This pin provides the status of the power supply for the ROM, TC35143F, system buffers, and other transient components in the system. This signal will be asserted by the System Power Supply when PWRCS is asserted, and will always be turned off when the system is powered down.

2.2.8 SIB Pins

SIBDIN: INPUT

This pin contains the input data shifted from TC35143F and/or external codec device.

SIBDOUT: OUTPUT

This pin contains the output data shifted to TC35143F and/or external codec device.

SIBSCLK: OUTPUT

This pin is the serial clock sent to TC35143F and/or external codec device. The programmable SIBSCLK rate is derived by dividing down from SIBMCLK.

SIBSYNC: OUTPUT

This pin is the frame synchronization signal sent to TC35143F and/or external codec device. This frame sync is asserted for one clock cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data.

SIBIRQ: INPUT

This pin is a general purpose input port used for the SIB interrupt source from TC35143F. This interrupt source can be configured to generate an interrupt on either a positive and/or negative edge.

SIBMCLK: INPUT/OUTPUT

This pin is the master clock source for the SIB logic. This pin is available for use in one of two modes. First, SIBMCLK can be configured as a high-rate output master clock source required by certain external codec devices. In this mode all SIB clocks are synchronously slaved to the main TMPR3922 system clock FREECLK. Conversely, SIBMCLK can be configured as an input slave clock source. In this mode, all SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to FREECLK. Also, for this mode, SIBMCLK can still be optionally used as a high-rate master clock source required by certain external codec devices.

2.2.9 SPI Pins

SPICLK: INPUT/OUTPUT

This pin is used to clock data in and out of the slave device. This pin is the master clock source for the SPI logic. This pin is available for use in one of two modes. First, SPICLK can be configured as a output master clock source required by certain external devices. In this mode all SPI clocks are synchronously slaved to the main TMPR3922 system clock XHFEE. Conversely, SPICLK can be configured as an input slave clock source. In this mode, all SPI clocks are derived from an external oscillator source, which is asynchronous with respect to XHFREE. .

SPIOUT: OUTPUT

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

SPIIN: INPUT

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

2.2.10 UART and IR Pins

TXD: OUTPUT

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

RXD: INPUT

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

IROUT: OUTPUT

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

IRINA: INPUT

This pin is an input for the receive signal to the IrDA(FIR) module.

IRINB: INPUT

This pin is an input for the receive signal to the IrDA(FIR) module.

RXPWR: OUTPUT

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

CARDET: INPUT

This pin is the UART receive signal to the UARTB module or is the carrier detect input signal from the external communication IR analog circuitry. If Consumer IR mode is enabled.

FIROUT: OUTPUT

This pin is an output for the transmit signal from the IrDA(FIR) module.

2.2.11 Endian Pin

ENDIAN: INPUT

This pin is used to select the endianess of the TMPR3922. The "1" level input sets the endianess to the big endian, while the "0" level input to the little endian.

2.2.12 Test Pins

TESTAIU: INPUT

This pin is used to define if the Boot ROM is 16 or 32 bits wide. If the TESTAIU pin is asserted during reset, the BIU will assume a 32-bit Boot ROM. The TESTAIU pin should remain static (either high or low).

TESTCPU: INPUT

This pin is used for debugging purposes only. Then the TESTCPU should not be asserted.

TESTIN: INPUT

This pin is used for debugging purposes only. Then the TESTIN should not be asserted.

2.2.13 Power Supply Pins

VDDH(15 each): +3.3V

These pins are the power pins for I/O of the TMPR3922.

VDDL(8 each): +2.5V

These pins are the power pins for the internal logic of the TMPR3922.

VDDLS(8 each): +2.5V

These pins are the power pins for the internal logic of the TMPR3922.
In the suspend mode these pins should be 0V.

VSS(32 each): GND

These pins are the ground pins for the TMPR3922.

2.3 Pin Usage Information

This section contains tables summarizing various aspects of the pin usage for the TMPR3922. Table 2-1 lists the standard versus multi-function usage for each TMPR3922 pin, if applicable. Those signal names shown in parentheses are test signals for debugging purposes only. The column showing the multi-function select signal and reset state indicates the internal control signal used to select the multi-function mode, as well as the default configuration of each multi-function pin during reset. The "Bus Arb State" column shows which pins are tri-stated whenever the DGRNT* signal is asserted in response to a DREQ* (external bus arbitration request).

Table 2-1 TMPR3922 Standard and Multi-Function Pin Usage

TMPR3922 pin	standard function (I = input, O = output)	multi-function	multi-function select (reset state: 1 = multi-function mode selected 0 = standard function & mode selected)	Bus Arb State
D[31:0]	D[31:0] (I/O)	X	X	Z
A[12:0]	A[12:0] (I/O)	X	X	Z
ALE	ALE (O)	X	X	Z
RD*	RD* (O)	X	X	Z
WE*	WE* (O)	X	X	Z
CAS0* (WE0*)	CAS0* (O)	X	X	Z
CAS1* (WE1*)	CAS1* (O)	X	X	Z
CAS2* (WE2*)	CAS2* (O)	X	X	Z
CAS3* (WE3*)	CAS3* (O)	X	X	Z
RAS0*	RAS0* (O)	X	X	Z
RAS1* (DCS1)*	RAS1* (O)	X	X	Z
DCS0*	DCS0* (O)	X	X	Z
DCKE	DCKE (O)	X	X	Z
DCLKIN	DCLKIN (I)	X	X	-
DCLKOUT	DCLKOUT (O)	X	X	Z
DQMH	DQMH (O)	X	X	Z
DQML	DQML (O)	X	X	Z
DREQ*	DREQ* (I)	MIO[27]	MIOSEL[27] (0)	-
DGRNT*	DGRNT* (O)	MIO[26]	MIOSEL[26] (0)	-
SYSCLKIN	SYSCLKIN (I)	X	X	-
SYSCLKOUT	SYSCLKOUT (O)	X	X	-
C32KIN	C32KIN (I)	X	X	-
C32KOUT	C32KOUT (O)	X	X	-
C6MIN	C6MIN (I)	X	X	-
C6MOUT	C6MOUT (O)	X	X	-
C48MOUT	C48MOUT(O)	X	X	-
BC32K	BC32K (O)	MIO[25]	MIOSEL[25] (1)	-
BCLK	BCLK (O)	X	X	-
PWRCS	PWRCS (O)	X	X	-
PWRINT	PWRINT (I)	X	X	-

TMPR3922 pin	standard function (I = input, O = output)	multi-function	multi-function select (reset state: 1 = multi-function mode selected 0 = standard function & mode selected)	Bus Arb State
PWROK	PWROK (I)	X	X	-
ONBUTN	ONBUTN (I)	X	X	-
CPURES*	CPURES* (I)	X	X	-
PON*	PON* (I)	X	X	-
TXD	TXD (O)	MIO[24]	MIOSEL[24] (0)	-
RXD	RXD (I)	MIO[23]	MIOSEL[23] (0)	-
CS0*	CS0* (O)	X	X	Z
CS1*	CS1* (O)	MIO[22]	MIOSEL[22] (0)	-
CS2*	CS2* (O)	MIO[21]	MIOSEL[21] (0)	-
CS3*	CS3* (O)	MIO[20]	MIOSEL[20] (0)	
MCS0*	MCS0* (O)	MIO[19]	MIOSEL[19] (0)	-
MCS1*	MCS1* (O)	MIO[18]	MIOSEL[18] (0)	-
MCS0WAIT*	MCS0WAIT * (I)	MIO[0]	MIOSEL[0] (0)	-
MCS1WAIT*	MCS1WAIT * (I)	MIO[1]	MIOSEL[1] (0)	-
CHIFS	CHIFS (I/O)	MIO[31]	MIOSEL[31] (1)	-
CHICLK	CHICLK (I/O)	MIO[30]	MIOSEL[30] (1)	-
CHIDOUT	CHIDOUT (O)	MIO[29]	MIOSEL[29] (1)	-
CHIDIN	CHIDIN (I)	MIO[28]	MIOSEL[28] (1)	-
VCC3	VCC3 (I)	X	X	-
IO15	IO15(I/O)	X	X	-
IO14	IO14(I/O)	X	X	-
IO13	IO13(I/O)	X	X	-
IO12	IO12(I/O)	X	X	-
IO11	IO11(I/O)	X	X	-
IO10	IO10(I/O)	X	X	-
IO9	IO9(I/O)	X	X	-
IO8	IO8(I/O)	X	X	-
IO7	IO7(I/O)	X	X	-
IO6	IO6 (I/O)	X	X	-
IO5	IO5 (I/O)	X	X	-
IO4	IO4 (I/O)	X	X	-
IO3	IO3 (I/O)	X	X	-
IO2	IO2 (I/O)	X	X	-
IO1	IO1 (I/O)	X	X	-
IO0	IO0 (I/O)	X	X	-
SPICLK	SPICLK (I/O)	MIO[15]	MIOSEL[15] (0)	-
SPIOUT	SPIOUT (O)	MIO[14]	MIOSEL[14] (0)	-
SPIIN	SPIIN (I)	MIO[13]	MIOSEL[13] (0)	-
SIBSYNC	SIBSYNC (O)	X	X	-
SIBDOUT	SIBDOUT (O)	X	X	-
SIBDIN	SIBDIN (I)	X	X	-
SIBMCLK	SIBMCLK (I/O)	MIO[12]	MIOSEL[12] (0)	-
SIBSCLK	SIBSCLK(O)	X	X	-
SIBIRQ	SIBIRQ (I)	X	X	-
RXPWR	RXPWR (O)	MIO[17]	MIOSEL[17] (1)	-
IROUT	IROUT(O)	MIO[16]	MIOSEL[16] (1)	
IRINA	IRINA (I)	X	X	-
IRINB	IRINB (I)	X	X	-
FIROUT	FIROUT (O)	X	X	-

TMPR3922 pin	standard function (I = input, O = output)	multi-function	multi-function select (reset state: 1 = multi-function mode selected 0 = standard function & mode selected)	Bus Arb State
CARDET	CARDET (I)	X	X	-
TESTAIU	TESTAIU (I)	X	X	-
TESTCPU	TESTCPU (I)	X	X	-
TESTIN	TESTIN (I)	X	X	-
CARDREG*	CARDREG* (O) (SHOWDINO CS*)	MIO[11]	MIOSEL[11] (1)	-
CARDIOWR*	CARDIOWR* (O)	MIO[10]	MIOSEL[10] (1)	-
CARDIORD*	CARDIORD* (O)	MIO[9]	MIOSEL[9] (1)	-
CARD1CSL*	CARD1CSL* (O)	MIO[8]	MIOSEL[8] (1)	-
CARD1CSH	CARD1CSH* (O)	MIO[7]	MIOSEL[7] (1)	-
CARD2CSL*	CARD2CSL* (O)	MIO[6]	MIOSEL[6] (1)	-
CARD2CSH*	CARD2CSH* (O)	MIO[5]	MIOSEL[5] (1)	-
CARD1WAIT*	CARD1WAIT* (I)	MIO[4]	MIOSEL[4] (1)	-
CARD2WAIT*	CARD2WAIT* (I)	MIO[3]	MIOSEL[3] (1)	-
CARDDIR*	CARDDIR* (O)	MIO[2]	MIOSEL[2] (1)	-
ENDIAN	ENDIAN (I)	X	X	-
VDDH-15 EACH	+3.3V	X	X	-
VDDL-8 EACH	+2.5V	X	X	-
VDDLS-8 EACH	+2.5V	X	X	-
VSS-32EACH	GND	X	X	-

Table 2-2 lists various power-down states and conditions for each TMPR3922 pin. The "Power-Down Control" column shows the conditions which trigger a power-down for each respective pin. This column also shows the reset state for each of these conditions.

The "PON* state" column defines the state of each pin at power-on reset (PON*). This condition is defined as initial power up of the TMPR3922, whereby the TMPR3922 is initialized and the TMPR3922 pins are reset to the state shown in the table. This state is entered after power is applied for the very first time (VSTANDBY is turned on but VCC3 is still turned off).

The "1st-time power-up state" column defines the state of each pin after power-up mode (RUNNING STATE) is executed for the first time. This mode is defined as VCC3 applied to the entire system and is initiated by the user pressing the ONBUTN while in the power-on reset (PON*) state. Note that the defined state of various pins for 1st-time power-up may depend on the configuration of external devices attached to these pins. After 1st-time power-up, the software could change the state of various pins to be different from those shown in the table. Thereafter, subsequent transitions from SLEEP STATE to RUNNING STATE might result in different states for these pins.

The "power-down state" column defines the state of each pin during power-down mode (SLEEP STATE). This mode is defined as VCC3 turned off to the entire system, except for the TMPR3922 (RTC and interrupts alive) and any persistent memory.

Table 2-2 TMPR3922 Power-Down Pin Usage

TMPR3922 pin	Power-Down Control powerdown = (vccon & vcc3)* (reset state)	PON* state	1st time power-up state	power-down state
D[31:0]	MEMPOWERDOWN	LOW	LOW	LOW
A[12:0]	MEMPOWERDOWN	LOW	LOW	LOW
ALE	X	LOW	LOW	LOW
RD*	POWERDOWN	LOW	HI	LOW
WE*	MEMPOWERDOWN	LOW	LOW	LOW
CAS0* (WE0*)	MEMPOWERDOWN	LOW	LOW	LOW
CAS1* (WE1*)	MEMPOWERDOWN	LOW	LOW	LOW
CAS2* (WE2*)	MEMPOWERDOWN	LOW	LOW	LOW
CAS3* (WE3*)	MEMPOWERDOWN	LOW	LOW	LOW
RAS0*	MEMPOWERDOWN	LOW	LOW	LOW
RAS1* (DCS1*)	MEMPOWERDOWN	LOW	LOW	LOW
DCS0*	MEMPOWERDOWN	LOW	LOW	LOW
DCKE	MEMPOWERDOWN	LOW	LOW	LOW
DCLKIN*	X	X	X	X
DCLKOUT	MEMPOWERDOWN	LOW	LOW	LOW
DQMH	MEMPOWERDOWN	LOW	LOW	LOW
DQML	MEMPOWERDOWN	LOW	LOW	LOW
DREQ*	POWERDOWN & MIOPD[27] (1)	PULL-DOWN	X	SELECTABLE
DGRNT*	POWERDOWN & MIOPD[26] (0)	LOW	HI	SELECTABLE
SYSCLKIN	POWERDOWN	OSC OFF	OSC ON	OSC OFF
SYSCLKOUT	POWERDOWN	OSC OFF	OSC ON	OSC OFF
C32KIN	X	OSC ON	OSC ON	OSC ON
C32KOUT	X	OSC ON	OSC ON	OSC ON
C6MIN	POWERDOWN	OSC OFF	OSC ON	OSC OFF
C6MOUT	POWERDOWN	OSC OFF	OSC ON	OSC OFF
C48MOUT	POWERDOWN	LOW	LOW	LOW
BC32K	POWERDOWN & MIOPD[25] (1)	PULL-DOWN	X	SELECTABLE
BCLK	POWERDOWN	LOW	LOW	LOW

TMPR3922 pin	Power-Down Control powerdown = (vccon & vcc3)* (reset state)	PON* state	1st time power-up state	power-down state
PWRCS	X	LOW	HI	LOW
PWRINT	X	X	X	X
PWROK	X	X	X	X
ONBUTN	X	X	X	X
CPURES*	X	X	X	X
PON*	X	X	X	X
TXD	POWERDOWN & MIOPD[24] (0)	LOW	LOW	SELECTABLE
RXD	POWERDOWN & MIOPD[23] (1)	PULL-DOWN	IN	SELECTABLE
CS0*	POWERDOWN	PULL-DOWN	HI	PULL-DOWN
CS1*	POWERDOWN & MIOPD[22] (1)	PULL-DOWN	HI	SELECTABLE
CS2*	POWERDOWN & MIOPD[21] (1)	PULL-DOWN	HI	SELECTABLE
CS3*	POWERDOWN & MIOPD[20] (1)	PULL-DOWN	HI	SELECTABLE
MCS0*	POWERDOWN & MIOPD[19] (0)	PULL-DOWN	HI	SELECTABLE
MCS1*	POWERDOWN & MIOPD[18] (0)	PULL-DOWN	HI	SELECTABLE
MCSWAIT0*	POWERDOWN & MIOPD[0] (0)	PULL-DOWN	IN	SELECTABLE
MCSWAIT1*	POWERDOWN & MIOPD[1] (0)	PULL-DOWN	IN	SELECTABLE
CHIFS	POWERDOWN & MIOPD[31] (1)	PULL-DOWN	IN	SELECTABLE
CHICLK	POWERDOWN & MIOPD[30] (1)	PULL-DOWN	IN	SELECTABLE
CHIDOUT	POWERDOWN & MIOPD[29] (1)	PULL-DOWN	IN	SELECTABLE
CHIDIN	POWERDOWN & MIOPD[28] (1)	PULL-DOWN	IN	SELECTABLE
VCC3	POWERDOWN	PULL-DOWN	X	PULL-DOWN
IO15	POWERDOWN & IOPD[15] (1)	PULL-DOWN	IN	SELECTABLE
IO14	POWERDOWN & IOPD[14] (1)	PULL-DOWN	IN	SELECTABLE
IO13	POWERDOWN & IOPD[13] (1)	PULL-DOWN	IN	SELECTABLE
IO12	POWERDOWN & IOPD[12] (1)	PULL-DOWN	IN	SELECTABLE
IO11	POWERDOWN & IOPD[11] (1)	PULL-DOWN	IN	SELECTABLE
IO10	POWERDOWN & IOPD[10] (1)	PULL-DOWN	IN	SELECTABLE
IO9	POWERDOWN & IOPD[9] (1)	PULL-DOWN	IN	SELECTABLE
IO8	POWERDOWN & IOPD[8] (1)	PULL-DOWN	IN	SELECTABLE
IO7	POWERDOWN & IOPD[7] (1)	PULL-DOWN	IN	SELECTABLE
IO6	POWERDOWN & IOPD[6] (1)	PULL-DOWN	IN	SELECTABLE
IO5	POWERDOWN & IOPD[5] (1)	PULL-DOWN	IN	SELECTABLE
IO4	POWERDOWN & IOPD[4] (1)	PULL-DOWN	IN	SELECTABLE
IO3	POWERDOWN & IOPD[3] (1)	PULL-DOWN	IN	SELECTABLE
IO2	POWERDOWN & IOPD[2] (1)	PULL-DOWN	IN	SELECTABLE
IO1	POWERDOWN & IOPD[1] (1)	PULL-DOWN	IN	SELECTABLE
IO0	POWERDOWN & IOPD[0] (1)	PULL-DOWN	IN	SELECTABLE
SPICLK	POWERDOWN & MIOPD[15] (0)	LOW	LOW	SELECTABLE
SPIOUT	POWERDOWN & MIOPD[14] (0)	LOW	LOW	SELECTABLE
SPIIN	POWERDOWN & MIOPD[13] (1)	PULL-DOWN	X	SELECTABLE
SIBYNC	POWERDOWN	LOW	LOW	LOW
SIBDOUT	POWERDOWN	LOW	LOW	LOW
SIBDIN	POWERDOWN	PULL-DOWN	X	PULL-DOWN
SIBMCLK	POWERDOWN & MIOPD[12] (1)	PULL-DOWN	IN	SELECTABLE
SIBSCLK	POWERDOWN	LOW	LOW	LOW
SIBIRQ	POWERDOWN	PULL-DOWN	X	PULL-DOWN
RXPWR	POWERDOWN & MIOPD[17] (0)	PULL-DOWN	IN	SELECTABLE
IROUT	POWERDOWN & MIOPD[16] (0)	PULL-DOWN	IN	SELECTABLE
CARDET	POWERDOWN	PULL-DOWN	X	PULL-DOWN

TMPR3922 pin	Power-Down Control powerdown = (vccon & vcc3)* (reset state)	PON* state	1st time power-up state	power-down state
IRINA	POWERDOWN	PULL-DOWN	X	PULL-DOWN
IRINB	POWERDOWN	PULL-DOWN	X	PULL-DOWN
FIROUT	POWERDOWN	LOW	LOW	LOW
TESTAIU	X	X	X	X
TESTCPU	X	X	X	X
TESTIN	X	X	X	X
CARDREG*	POWERDOWN & MIOPD[11] (1)	PULL-DOWN	X	SELECTABLE
CARDIOWR*	POWERDOWN & MIOPD[10] (1)	PULL-DOWN	X	SELECTABLE
CARDIORD*	POWERDOWN & MIOPD[9] (1)	PULL-DOWN	X	SELECTABLE
CARD1CSL*	POWERDOWN & MIOPD[8] (1)	PULL-DOWN	X	SELECTABLE
CARD1CSH*	POWERDOWN & MIOPD[7] (1)	PULL-DOWN	X	SELECTABLE
CARD2CSL*	POWERDOWN & MIOPD[6] (1)	PULL-DOWN	X	SELECTABLE
CARD2CSH*	POWERDOWN & MIOPD[5] (1)	PULL-DOWN	X	SELECTABLE
CARD1WAIT*	POWERDOWN & MIOPD[4] (1)	PULL-DOWN	X	SELECTABLE
CARD2WAIT*	POWERDOWN & MIOPD[3] (1)	PULL-DOWN	X	SELECTABLE
CARDDIR*	POWERDOWN & MIOPD[2] (1)	PULL-DOWN	X	SELECTABLE
ENDIAN	X	X	X	X
VDD-15 EACH	X	X	X	X
VDDL-8 EACH	X	X	X	X
VDDLS-8 EACH	X	X	X	X
VSS-32 EACH	X	X	X	X

SECTION 3 CPU Module

3.1 Overview

The TMPR3922 consists of an embedded TX3920 Processor core along with the System Interface Logic required to support the PIC System. These include a 16KByte instruction cache and an 8KByte data cache used to improve performance, a 32 × 32 multiplier with accumulator used to perform DSP functions to support features such as software modems.

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

3.2 CPU Core

3.2.1 Description

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

- Translation Look-aside Buffer (TLB) (4K/16K/64K/256K/1M/4M Page size 64 Entries)
- 16 KByte instruction cache (I-cache)
  - 16 bytes (4 words) per line (512 lines total)
  - physical address tag per cache line
  - single valid bit per cache line
  - programmable burst size (16 bytes to 128 bytes)
  - instruction streaming mode supported
  - two-way set associative
- 8 KByte data cache (D-cache)
  - 16 bytes (4 word) per line (256 lines total)
  - physical address tag per cache line
  - single valid bit per cache line
  - programmable burst size (16 bytes to 128 bytes); (write-through mode)
  - write-back/write-through
  - two-way set associative
- cache address snoop mode supported for DMA(write-through mode)
- 32 × 32 on-chip hardware multiplier
  - supports 32 × 32 (single-cycle) multiplier operations, with 64-bit accumulator
  - signed or unsigned data formats
  - accumulator supports sign-extension and overflow status
  - existing multiply instructions are enhanced and new multiply and add instructions are added to the R3000A instruction set to improve the performance of DSP applications

3.3 CPU Interface

3.3.1 Block Diagram

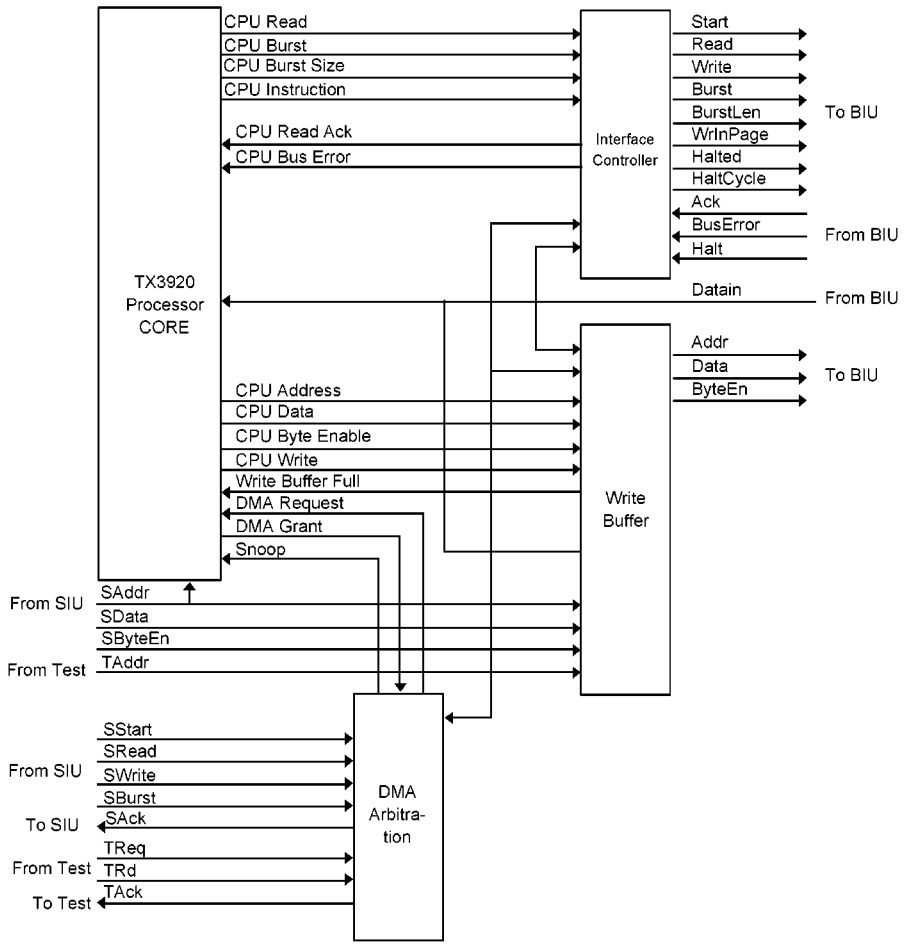


Figure 3-1 CPU Inerface Block Diagram

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

3.3.2 Write Buffer

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

3.3.3 Interface Controller

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

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

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

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

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

3.3.4 DMA Arbitration

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

At the same time that the data is loaded into the Write Buffer, the Snoop signal is asserted to the CPU core. The snoop logic in the CPU core will invalidate the cache locations defined by the SAddr bits when the Snoop signal is asserted. This provides cache coherency during DMA to simplify managing the cache. But this function is only supported in the write-through mode, not write-back mode.

For a DMA read transaction, the SRead signal will be asserted by the SIU. The Write Buffer will always be flushed before allowing the DMA read to begin. Once the Write Buffer is empty, the DMA read will begin and the SAck signal will be asserted to end the DMA read once the BIU asserts the Ack signal. The SIU will receive data input via the DataIn bus.

3.3.5 CPU Stop Mode

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

SECTION 4 BIU Module

4.1 Overview

The BIU provides an interface to the memory devices in the system, including static devices, as well as DRAM, EDO and SDRAM devices. An overall block diagram of the BIU is shown in Figure 4-1. The BIU provides control to support the following memory devices:

- 2 Banks of SDRAM and/or DRAM
  - 16-bit or 32-bit SDRAM configuration
  - 16-bit or 32-bit DRAM(EDO) configuration
  - 4 Mbit, 16 Mbit and 64 Mbit parts supported
  - page mode reads and writes supported
  - independent refresh counters for each bank
  - self refreshing parts supported to retain memory when system is powered down
- 4 general purpose chip selects (CS3*-CS0*)
  - 16-bit or 32-bit ports
  - programmable wait states
  - read page mode
- 2 general purpose chip selects (MCS1*- MCS0*)
  - 16-bit or 32-bit ports
  - programmable wait states
  - read page mode and writes supported
  - WAIT signal supported
- 2 full PCMCIA slots
  - 8-bit or 16-bit ports
  - IORD and IOWR provided to support I/O cards
  - WAIT signal supported

Most of the BIU logic uses the DCIKOUT signal (nominally 73.728 MHz) The DCLKOUT signal is also used to provide more precise timing for generating the control signals for the chip selects, as well as the SDRAM and DRAM control signals. Input data is sampled using the DCLKIN signal which is connected externally to the DCLKOUT signal. The external connection ensures that the input data and DCLKIN will not be skewed, which is necessary to meet the hold time specification on the SDRAM devices. Also, the external looping of DCLKOUT to DCLKIN helps to improve the access time of normal DRAMs, since the input data will be sampled closer to the end of the access.

4.1.1 Block Diagram

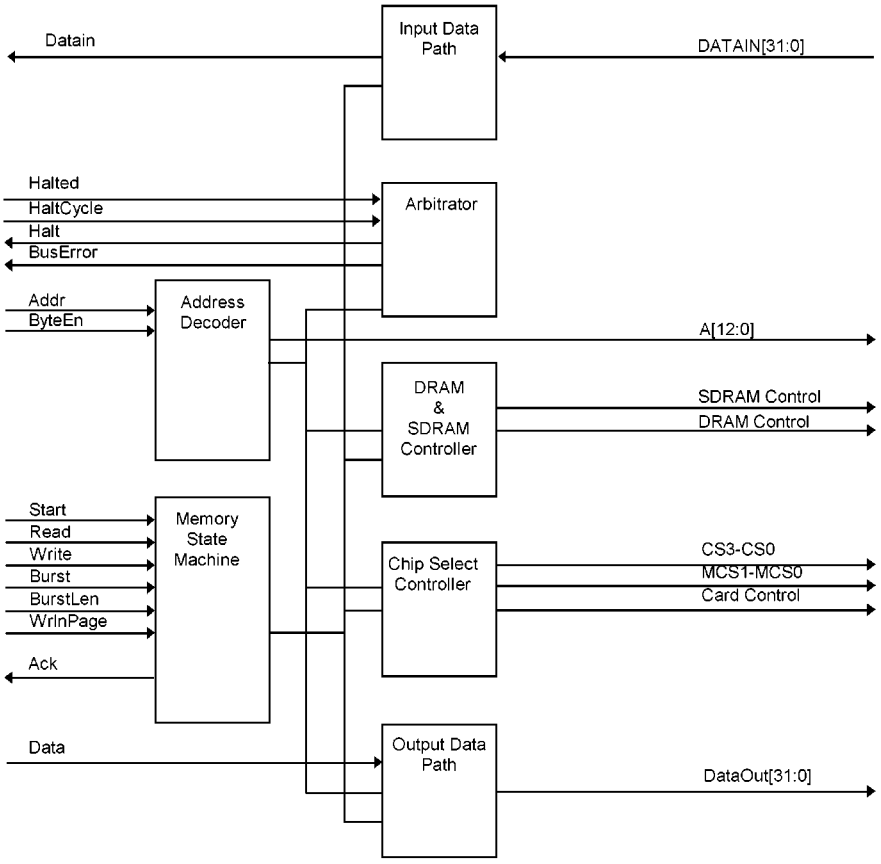


Figure 4-1 BIU Block Diagram

4.2 Address Decoder

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

4.2.1 System Address Map

The System Address Map is shown in Table 4-1. The address space is partitioned to support two banks (Bank 1 and Bank 0) of DRAM and/or SDRAM, four general purpose 16/32-bit chip selects (CS3-CS0), two general purpose 16/32-bit chip selects (MCS1-MCS0), two PCMCIA slots (Card 2 and Card 1), and internal function register access. The address space contains two primary regions: kseg0/kseg1 are only accessible in kernel mode, while kuseg is accessible in either user or kernel mode.

MCS1-MCS0 are available in kuseg space and are thus always available to either user or kernel mode accesses. CS3-CS1 are located in both kseg and kuseg space, but accesses in kuseg space can be independently disabled for each chip select using the ENCS3USER, ENCS2USER, and ENCS1USER control bits in the Memory Configuration 0 Register. CS0 is used as the chip select for the boot ROM in the system. It is mapped starting at address $11000000 in kernel space, which will map into the TX3920 boot vector location at address $1FC00000. CS0 is also mapped in kuseg space to allow access to the ROM by programs operating in user mode.

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

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

A system can be built without DRAM or SDRAM and instead can contain SRAM as the main memory. In this case at least one bank of SRAM should be connected to CS1 and the ENCS1DRAM control bit in the Memory Configuration 0 Register should be set. Setting this bit will cause CS1 to map in place of Bank 1 and Bank 0, as shown in Table 4-1. This is required because exception vectors are mapped into low memory, thus it is required that some memory exist in this region to support exception handling. Accesses to reserved or non-enabled locations will respond with a Bus Error if the Watch Dog Timer is enabled.

Table 4-1 System Address Map

Size	Address	Segment	Devices
16 Mbyte	FF00_0000	reserved	reserved
1 Gbyte	C000_0000	kseg2	reserved
1 Gbyte	8000_0000	kuseg	reserved
64 Mbyte	7C00_0000	kuseg	reserved
128 Mbyte	7400_0000	kuseg	reserved
64 Mbyte	7000_0000	kuseg	MCS1
64 Mbyte	6C00_0000	kuseg	MCS0
64 Mbyte	6800_0000	kuseg	Card 2 (Memory)
64 Mbyte	6400_0000	kuseg	Card 1 (Memory)
64 Mbyte	6000_0000	kuseg	CS3 (if enabled)
64 Mbyte	5C00_0000	kuseg	CS2 (if enabled)
64 Mbyte	5800_0000	kuseg	CS1 (if enabled)
128 Mbyte	5000_0000	kuseg	CS0 (ROM)
128 Mbyte	4800_0000	kuseg	reserved
32 Mbyte	4600_0000	kuseg	DRAM BANK1
32 Mbyte	4400_0000	kuseg	DRAM BANK0
32 Mbyte	4200_0000	kuseg	CS1 (if enabled, else DRAM BANK 1)
32 Mbyte	4000_0000	kuseg	CS1 (if enabled, else DRAM BANK 0)
512 Mbyte	2000_0000	reserved	reserved
240 Mbyte	1100_0000	kseg0, kseg1	CS0 (ROM)
1 Mbyte	10F0_0000	kseg0, kseg1	SDRAM BANK 1 Mode Register
1 Mbyte	10E0_0000	kseg0, kseg1	SDRAM BANK 0 Mode Register
2 Mbyte	10C0_0000	kseg0, kseg1	Internal Function Registers
4 Mbyte	1080_0000	kseg0, kseg1	CS3
4 Mbyte	1040_0000	kseg0, kseg1	CS2
4 Mbyte	1000_0000	kseg0, kseg1	CS1
64 Mbyte	0C00_0000	kseg0, kseg1	Card 2 (I/O or Attribute)
64 Mbyte	0800_0000	kseg0, kseg1	Card 1 (I/O or Attribute)
32 Mbyte	0600_0000	kseg0, kseg1	DRAM BANK1
32 Mbyte	0400_0000	kseg0, kseg1	DRAM BANK 0
32 Mbyte	0200_0000	kseg0, kseg1	CS1 (if enabled, else DRAM BANK 1)
32 Mbyte	0000_0000	kseg0, kseg1	CS1 (if enabled, else DRAM BANK 0)

4.2.2 Address Generation

The Address Decoder logic generates the A[12:0] address bus for the memory devices. Addresses are multiplexed on the A[12:0] address bus. For static devices such as CS3-CS0, MCS1-MCS0, Card 2 and Card 1, the addresses are multiplexed such that Addr[25:13] is provided first-in-time followed by Addr[12:0], as shown in Figure 4-2. Addr[25:13] must be latched with an external latch using the ALE pin. The latched addresses concatenated with A[12:0] will provide a 26-bit address bus for the system, HA[25:0].

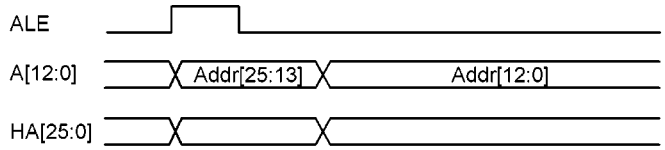


Figure 4-2 ALE Timing

DRAM and SDRAM devices contain multiplexed address buses. Thus, A[12:0] will be directly connected to the DRAM and SDRAM devices. In order to support a wide range of DRAM and SDRAM types and configurations, there are control bits in the Memory Configuration 0 Register that provide soft configuration for the Row and Column address bit positions. These control bits are ROWSEL1[1:0] and COLSEL1[2:0] for Bank 1, and ROWSEL0[1:0] and COLSEL0[2:0] for Bank 0. Table 4-2 shows the position that specific address bits are placed, depending on the configuration of the control bits. This flexibility in address bit positioning makes it possible to support 4-Mbit, 16-Mbit, and 64-Mbit DRAM and SDRAM devices with different bus width configurations. The two sets of control bits make it possible to support different devices in Bank 1 and Bank 0. The "ϕ" character in the table means that this bit position will be a zero and the "X" implies a "don't care" bit position.

Table 4-2 TMPR3922 DRAM and SDRAM Address Mapping

ROW A[12:0]	ROWSEL	COL A[12:0]	COLSEL
18, 17:9	00	22, 20, 18, 8:1	0000
22, 18, 20, 19, 17:9	01	19, 18, 8:2	0001
20, 22, 21, 19, 17:9	10	21, 20, 18, 8:2	0010
22, 23, 21, 19, 17:9	11	23, 22, 20, 18, 8:2	0011
		24, 22, 20, 18, 8:2	0100
		22, 18, ϕ, 24, 23, 21, 8:2	0101
		22, X, ϕ, X,24,21, 8:2	0110
		22, 18, ϕ, 23, 21, 8:1	0111
		22, X, ϕ, 24, 21, 8:1	1000
		24, 23, 21, 8:2	1001

4.2.3 Address Re-Mapper Logic

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

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

Table 4-3 Address Re-Mapper

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

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

4.3 Chip Select Controller

4.3.1 Description

The Chip Select Controller provides the logic required to generate CS3-CS0, MCS1-MCS0, Card 2, and Card 1 control signals. CS3-CS0 and MCS1-MCS0 can each independently be set as either 32-bit or 16-bit ports. Card 2 and Card 1 can each independently be set as either 16-bit or 8-bit ports.

Reads from a 32-bit port will result in a single access with the appropriate chip select being asserted along with the RD* signal. Writes to a 32-bit port will result in a single access with the appropriate chip select being asserted along with the WE* signal, as well as CAS3*-CAS0* being asserted to select the byte to be written. The ByteEn signals from the CPU Interface define the desired byte lane for memory accesses. A Word access will read or write all 32 bits, a Tri access will read or write either the upper 24 bits or lower 24 bits, a Half-Word access will read or write either the upper 16 bits or lower 16 bits, and a Byte access will read or write any of the 4 bytes.

For writes to CS3-CS0 when the ports are configured as 32 bits, the CAS3*-CAS0 signals are used as individual write enables for each byte, with CAS0* used for the lower byte, CAS1* used for the next byte, etc. Word or Tri accesses to 16-bit ports will be split into two consecutive accesses. HA(1) is used to select between the upper or lower 16 bits to be accessed. The CAS0* and CAS1* signals are used as write enables for the byte lanes for accesses to 16-bit ports. Accesses to Card 2 or Card 1 do not use CAS0* and CAS1*, since the card already contains an upper and lower chip select. All 16-bit port memory devices should always be connected to D[15:0].

The access time for each chip select is individually programmed using the control bits in the Memory Configuration 1-3 Registers. Also provided for CS3-CS0 and MCS1-MCS0 is separate access time control for Read Page Mode that can be enabled or disabled independently for each chip select using the ENxxPAGE control bits in the Memory Configuration 3 Register. The Read Page Mode provides burst read capability to support memory devices such as page mode ROMs. There is separate access time control provided for memory accesses to Card 2 or Card 1, or IO/Attribute accesses to Card 2 or Card 1. The address inputs for all 32-bit memory devices should be connected starting with HA(2) and all 16-bit memory devices should be connected starting with HA(1). The PCMCIA cards will use all the address bits HA[25:0].

4.3.2 Access Mapping

Tables 4-5 and 4-6 show how write and read accesses are presented to the memory devices. For 32-bit ports there is only one access required, but Word and Tri accesses to 16-bit ports must be split into two accesses.

Table 4-4 Chip Select Write Map (Little Endian)

Access	Byte En	HA[1:0]	Port Size	D[7:0]	D[15:8]	D[23:16]	D[31:24]
Word	1111	00	32-bit	Data[31:24]	Data[23:16]	Data[15:8]	Data[7:0]
Word	1111	00	16-bit	Data[31:24]	Data[23:16]	X	X
Word	1111	10	16-bit	Data[15:8]	Data[7:0]	X	X
Tri	1110	00	32-bit	Data[31:24]	Data[23:16]	Data[15:8]	X
Tri	1110	00	16-bit	Data[31:24]	Data[23:16]	X	X
Tri	1110	10	16-bit	Data[15:8]	X	X	X
Tri	0111	01	32-bit	X	Data[23:16]	Data[15:8]	Data[7:0]
Tri	0111	01	16-bit	X	Data[23:16]	X	X
Tri	0111	10	16-bit	Data[15:8]	Data[7:0]	X	X
Half-Word	1100	00	32-bit	Data[31:24]	Data[23:16]	X	X
Half-Word	1100	00	16-bit	Data[31:24]	Data[23:16]	X	X
Half-Word	0011	10	32-bit	X	X	Data[15:8]	Data[7:0]
Half-Word	0011	10	16-bit	Data[15:8]	Data[7:0]	X	X
Byte	1000	00	32-bit	Data[31:24]	X	X	X
Byte	1000	00	16-bit	Data[31:24]	X	X	X
Byte	0100	01	32-bit	X	Data[23:16]	X	X
Byte	0100	01	16-bit	X	Data[23:16]	X	X
Byte	0010	10	32-bit	X	X	Data[15:8]	X
Byte	0010	10	16-bit	Data[15:8]	X	X	X
Byte	0001	11	32-bit	X	X	X	Data[7:0]
Byte	0001	11	16-bit	X	Data[7:0]	X	X

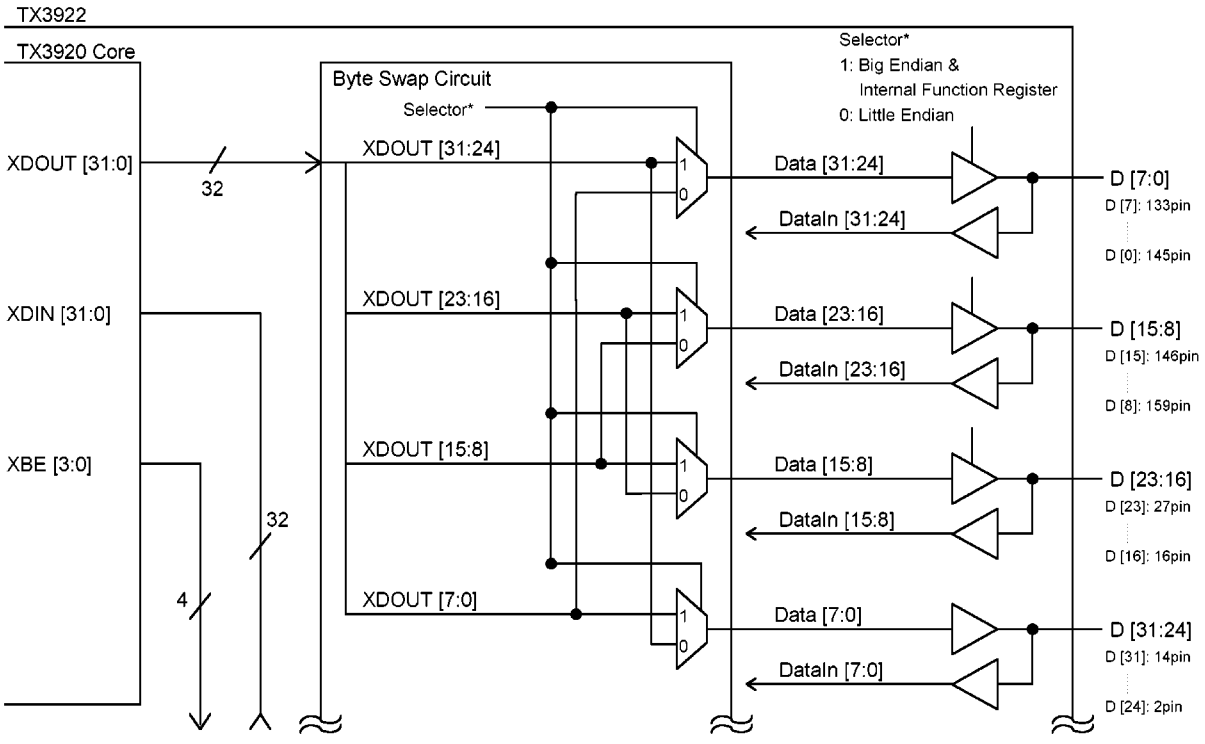


Figure 4-3a TX3922 Byte Swap Circuit (1/2)

Table 4-5 Chip Select Read Map (Little Endian)

Access	Byte En	HA[1:0]	Port Size	DataIn[31:24]	DataIn[23:16]	DataIn[15:8]	DataIn[7:0]
Word	1111	00	32-bit	D[7:0]	D[15:8]	D[23:16]	D[31:24]
Word	1111	00	16-bit	D[7:0]	D[15:8]	X	X
Word	1111	10	16-bit	X	X	D[7:0]	D[15:8]
Tri	1110	00	32-bit	D[7:0]	D[15:8]	D[23:16]	X
Tri	1110	00	16-bit	D[7:0]	D[15:8]	X	X
Tri	1110	10	16-bit	X	X	D[7:0]	X
Tri	0111	01	32-bit	X	D[15:8]	D[23:16]	D[31:24]
Tri	0111	01	16-bit	X	D[15:8]	X	X
Tri	0111	10	16-bit	X	X	D[7:0]	D[15:8]
Half-Word	1100	00	32-bit	D[7:0]	D[15:8]	X	X
Half-Word	1100	00	16-bit	D[7:0]	D[15:8]	X	X
Half-Word	0011	10	32-bit	X	X	D[23:16]	D[31:24]
Half-Word	0011	10	16-bit	X	X	D[7:0]	D[15:8]
Byte	1000	00	32-bit	D[7:0]	X	X	X
Byte	1000	00	16-bit	D[7:0]	X	X	X
Byte	0100	01	32-bit	X	D[15:8]	X	X
Byte	0100	01	16-bit	X	D[15:8]	X	X
Byte	0010	10	32-bit	X	X	D[23:16]	X
Byte	0010	10	16-bit	X	X	D[7:0]	X
Byte	0001	11	32-bit	X	X	X	D[31:24]
Byte	0001	11	16-bit	X	X	X	D[15:8]

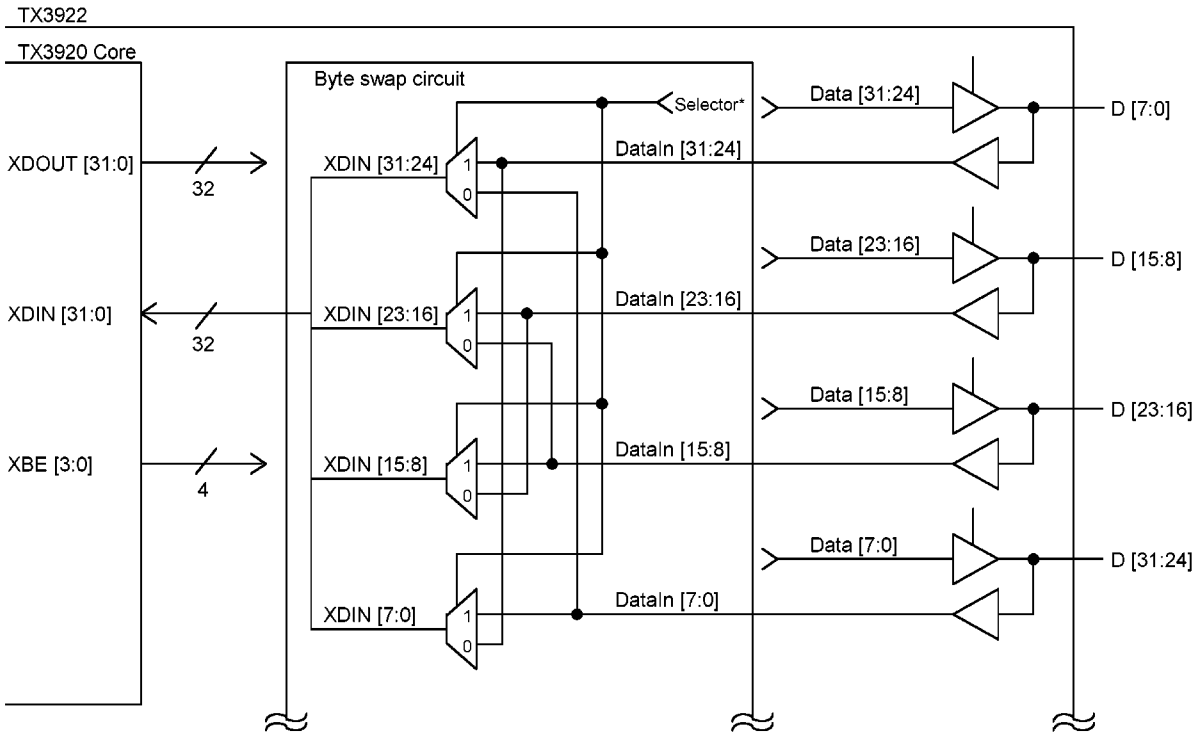


Figure 4-3b TX3922 Byte Swap Circuit (2/2)

4.3.3 CS3-CS0 and MCS1-MCS0 Timing

Figures 4-4 through 4-7 show the timing for CS3-CS0 and MCS1-MCS0. The timing is the same for all these signals, with the only exception that MCS1-MCS0 support a WAIT signal for the external device and has the two kind of timing which is set by software.

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

Figure 4-4 shows the timing for a single Word read and write for a 32-bit memory device. The access time is defined using the ACCVAL1 control bits. There are separate ACCVAL1 control bits for each of the chip selects, such that each chip select has an independently configurable access time. The ACCVAL1 bits should be set to one less than the desired number of CLK times for the chip select.

Figure 4-5 shows the timing for a Word read from a 16-bit port. The access is split into two parts to read the full 32 bits. During the first read HA(1) will be low and during the second read HA(1) will be high. The input data path will latch both accesses and concatenate them together as 32 bits for the processor before asserting the Ack signal to the CPU Interface logic.

Figure 4-6 shows a 4-Word burst read from a 32-bit port. The ALE signal is only asserted at the beginning of the cycle since the upper address bits will not be changed during the burst. The burst will always start on a 16-byte boundary. The address bits will increment as follows: HA[3:2] = 00, HA[3:2] = 01, HA[3:2] = 10, then HA[3:2] = 11.

Figure 4-7 shows the same 4-Word burst read from a 32-bit port, except this time Read Page Mode is enabled. In this case the chip select will remain asserted for the entire access and the addresses will change to access the new address location. The first access uses the ACCVAL1 control bits, while the next three accesses use the ACCVAL2 control bits. If the burst had been longer then 4 Words, the ACCVAL1 control bits would have been used for every fourth access. This type of bursting is compatible with ROM devices that support page mode access. If the 4-Word burst would have been from a 16-bit port, eight accesses would have been required, but the ACCVAL1 bits would still be used for the first and fifth access, while the ACCVAL2 bits would have been used for the rest. If Read Page Mode is enabled for a particular chip select, then the ACCVAL1 and ACCVAL2 control bits must not be set to $0. The “up arrow” markings on DCLKOUT in Figures 4-4 through 4-7 show the points where input data is sampled.

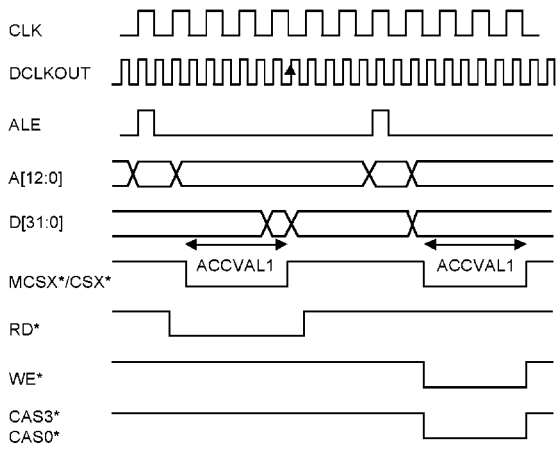


Figure 4-4a Word Read/Write 32-Bit Port (EN MCSXACC bit = 1) (EN MCSXBE bit = 0)

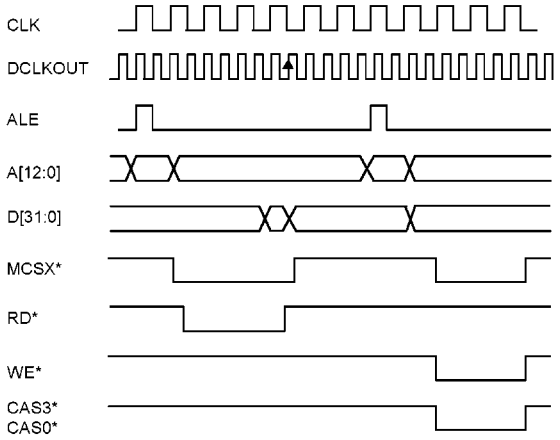


Figure 4-4b Word Read/Write 32-Bit Port (EN MCSXACC bit = 0) (EN MCSXBE bit = 0)

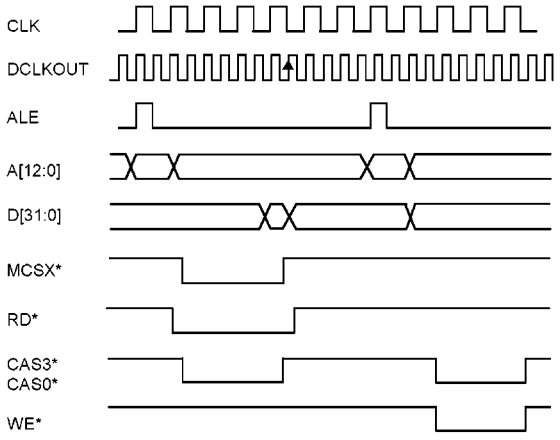


Figure 4-4c Word Read/Write 32-Bit Port (EN MCSXACC bit = 1) (EN MCSXBE bit = 1)

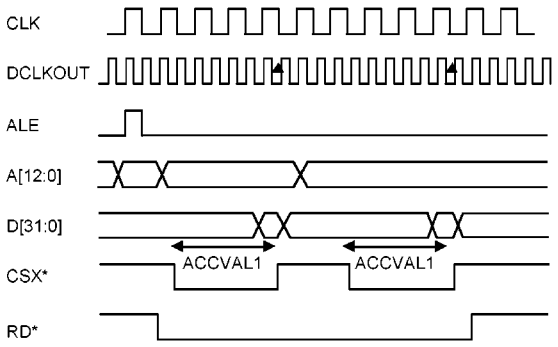


Figure 4-5 Word Read 16-Bit Port

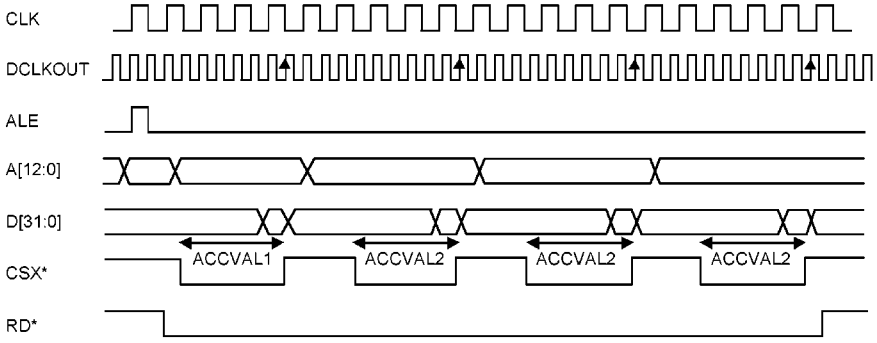


Figure 4-6 4-Word Burst Read 32-Bit Port

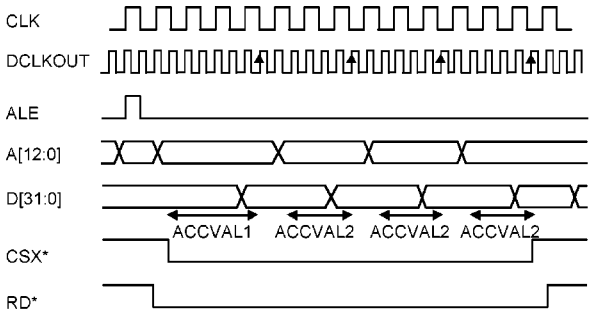


Figure 4-7 4-Word Burst Read 32-Bit Port Read with Page Mode Enabled

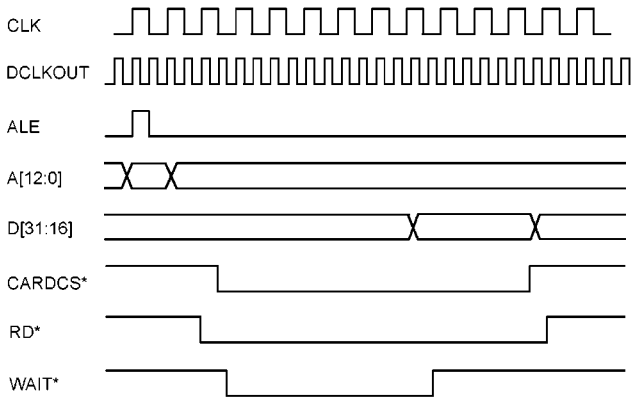


Figure 4-8 Wait Signal "MCS1-0"

4.3.4 Card 2 and Card 1

The chip selects for Card 2 and Card 1 are designed to comply with Revision 2.01 of the PCMCIA specification. Table 4-6 shows the byte lanes on the PCMCIA card. To support this mapping it is required that CD[7:0] be connected to D[7:0] and CD[15:8] be connected to D[15:8] when the TMPR3922 is mapped as Little Endian.

Table 4-6 PCMCIA Byte Lanes

	/ CARDxCSH	/ CARDxCSL	HA (0)	CD[15:8]	CD[7:0]
Standby	1	1	X	High-Z	High-Z
Byte	1	0	0	High-Z	Even-Byte
Byte (8 bit)	1	0	1	High-Z	Odd-Byte
Byte (16 bit)	0	1	X	Odd-Byte	High-z
Word	0	0	X	Odd-Byte	Even-Byte

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

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

There is separate access time control provided for Memory access versus IO Attribute* access. CARDxACCVAL is used to control the access time for memory accesses and CARDxIOACCVAL is used to control the access time for IO or Attribute access. Adding one to either access value control setting will increase the access time by two CLK periods. A value of $0 will result in an access time of 1 CLK period. Thus, the formula for the access time for the card is: Access Time = (ACCVAL * 2 - 1) CLK periods. Figures 4-9 through 4-11 show the different timing for Memory, Attribute and IO accesses. If a Word access is made to the Card, the access will be split into two accesses in the same manner as is done for CS3-CS0 and MCS1-MCS0. Read Page Mode is not supported for the Cards since it is not supported by the PCMCIA specification.

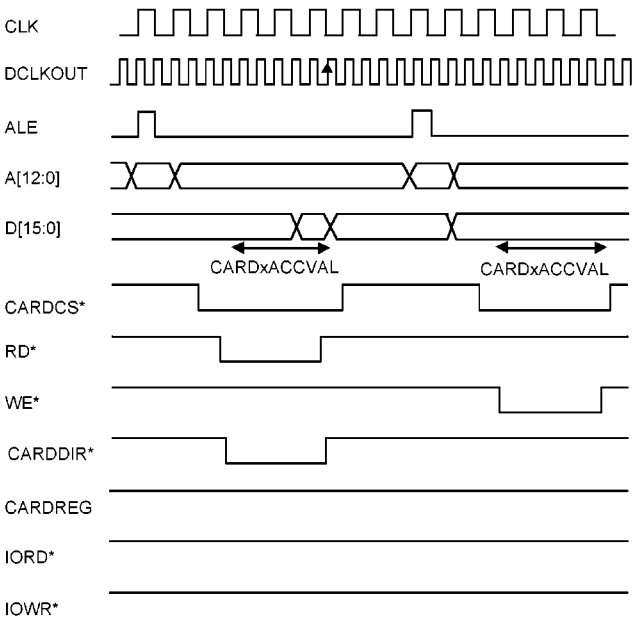


Figure 4-9 Memory Access to Card

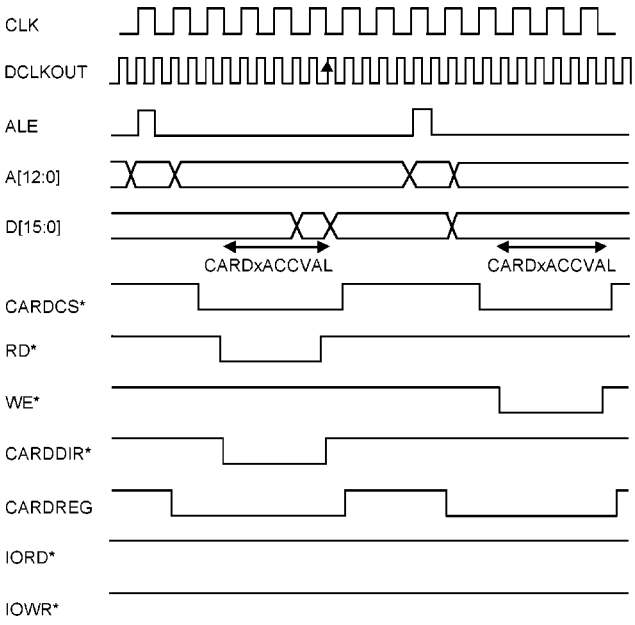


Figure 4-10 Attribute Access to Card

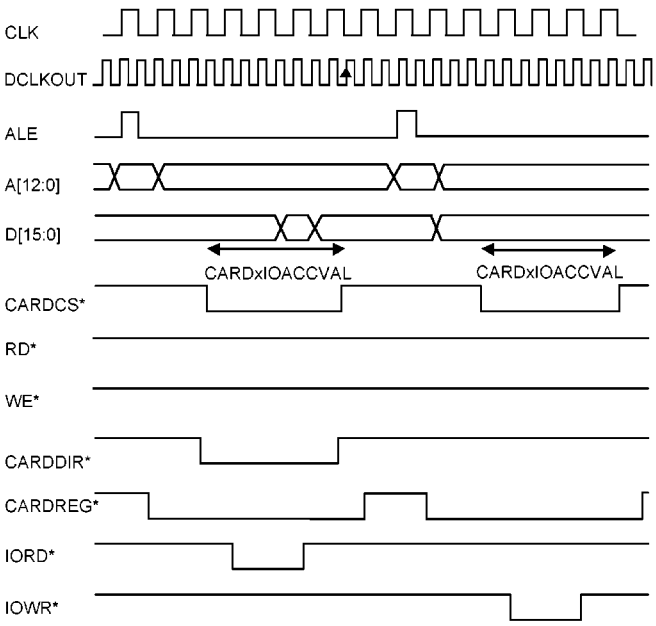


Figure 4-11 IO Access to Card

The PCMCIA specification provides for a WAIT* signal that allows the Card to halt the bus cycle until the WAIT* signal is de-asserted. This allows the Card to control the access time. The WAIT* signal will assert at the beginning of the cycle and the cycle should not terminate until the WAIT* signal becomes de-asserted. There are separate WAIT* signals provided for each Card: WAIT2* and WAIT1*. The WAIT* signals are sampled on two consecutive rising edges of CLK. If the WAIT* signal is low, the access time counter will freeze until the WAIT* signal goes high. This function will only occur if the WAIT function is enabled using the CARDxWAITEN control bit. If the WAIT function is not enabled, the WAIT* signal is ignored. Figure 4-12 shows the timing for Card WAIT accesses.

The ACCVAL for the card must be set to a minimum value such that the cycle will not end before the WAIT* signal has been sampled for two clock periods. Support for the WAIT* signal has been included to comply with the PCMCIA specification, but enabling and usage of this function is not recommended in a PIC system. The reason is that the WAIT* signal can assert for up to 12 μs, which will cause the CPU and DMA to not be able to access memory for this amount of time. This could cause major problems in the system since the DMA will typically need to access memory more often than 12 μs and certain programs cannot be held off for so long. Support of the WAIT* signal should be used cautiously depending on the maximum WAIT* time for a particular card and the other demands of the system.

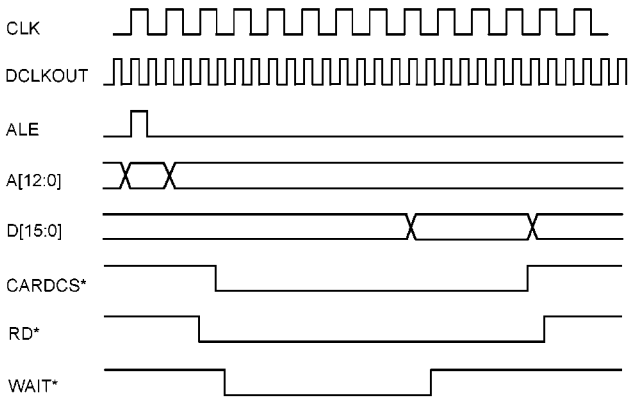


Figure 4-12 Card Wait Timing

4.3.5 CS0 Size Configuration

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

4.4 DRAM and SDRAM Controller

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

4.4.1 Memory Chips Supported

Table 4-7 shows the different DRAM and SDRAM chips that are supported by the DRAM and SDRAM Controller. Also shown is the Row and Column configuration for the supported chips. SDRAM chips also contain two / four banks that are selectable through the address space.

Table 4-7 Supported DRAM and SDRAM Chips

Bank Select	ROW	COL	Memory Size	Memory Type
—	9	9	256K x 16	4-Mbit DRAM
—	10	8	256K x 16	4-Mbit DRAM
—	10	9	512K x 8	4-Mbit DRAM
—	10	10	1M x 16	16-Mbit DRAM
—	11	9	1M x 16	16-Mbit DRAM
—	12	8	1M x 16	16-Mbit DRAM
—	11	10	2M x 8	16-Mbit DRAM
—	12	9	2M x 8	16-Mbit DRAM
1	11	8	1M x 16	16-Mbit SDRAM
—	11	11	4M x16	64-Mbit DRAM
—	12	10	4M x16	64-Mbit DRAM
—	12	11	8M x 8	64-Mbit DRAM
2	11	9	4M x 16	64-Mbit SDRAM

4.4.2 DRAM and SDRAM Configurations

Table 4-8 shows the various combinations of DRAM and SDRAM chips that are supported. Bank 1 and Bank 0 are independently programmable so each can contain any of the combinations. The only restriction is that Bank 1 cannot contain SDRAM if Bank 0 contains DRAM.

Table 4-8 Supported DRAM and SDRAM Configurations

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

4.4.3 DRAM

DRAM in Bank 1 is controlled using RAS1* and CAS3*-CAS0*, while DRAM in Bank 0 is controlled using RAS0* and CAS3*-CAS0*. The Memory Configuration 0 Register provides control bits that allow Bank 1 or Bank 0 to be independently set as either 32-bit or 16-bit DRAM configurations. The 32-bit DRAM configurations are connected to all 32 data bus bits D[31:0], while 16-bit DRAM configurations are connected to D[31:16]. The 16-bit DRAM configurations will only use CAS3* and CAS2* for control, while 32-bit configurations will use all four CAS* lines. Word and Tri accesses to 16-bit configured DRAMs will be split into two accesses in the same manner as was done for the chip selects, except that the data is connected to D[31:16] instead of D[15:0]. Connecting DRAM devices to the lower data bus and chip select memory devices to the upper data bus helps to balance the capacitive loading on the data bus.

The DRAM (FPM) timing is fixed to support 70 ns devices or faster. The only programmability is the ENBANK1OPT and ENBANK0OPT control bits in the Memory Configuration 4 Register that allow an extra clock of time for the first access to the DRAM. This option is included to support 80 ns DRAM devices with a 73.728 MHz clock for the BIU. DRAMs faster than 70 ns will not need the ENBANKxOPT bit to be set. And the EDO timing is fixed to support 50 ns devices or faster.

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

Refresh is inserted using CAS* before RAS* refresh. The Refresh Control is generated by the Arbitration Logic. Figures 4-13 through 4-17 show various timing examples, with refresh timing shown in Figure 4-15. The timing is the same for 32-bit and 16-bit ports, except there are twice as many accesses to the 16-bit port. For example, a Word read from a 16-bit port will result in a burst of two 16-bit reads to fetch the 32-bits of data.

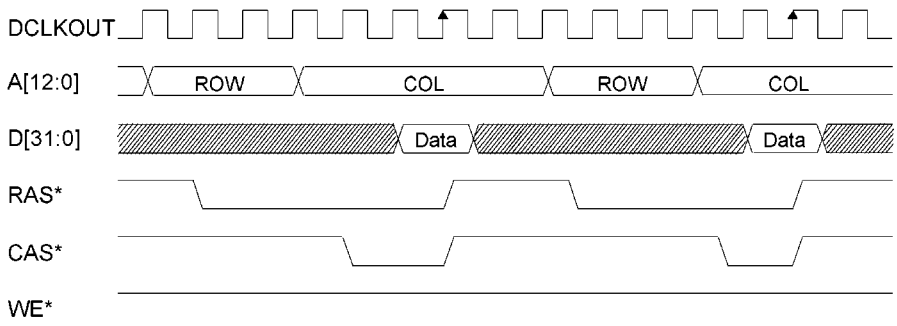


Figure 4-13a DRAM Read (Word)

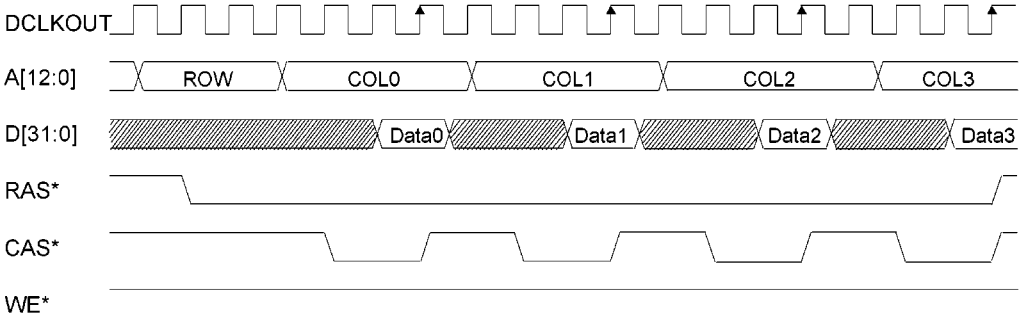


Figure 4-13b Burst Read (4Word)

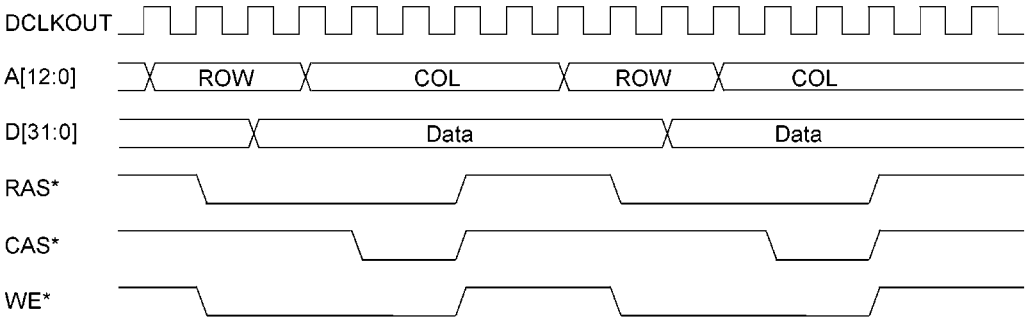


Figure 4-14a Write (Word)

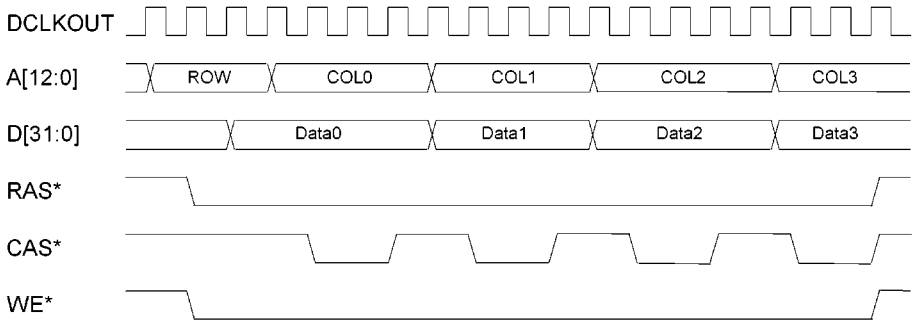


Figure 4-14b In-Page Write

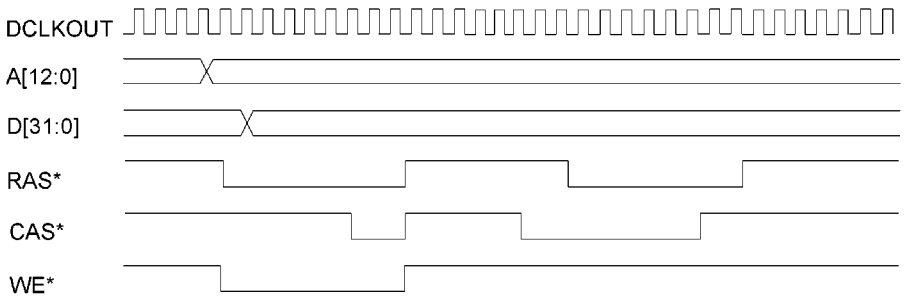


Figure 4-15 Write with ENBANKOPT Set followed by Refresh Cycle

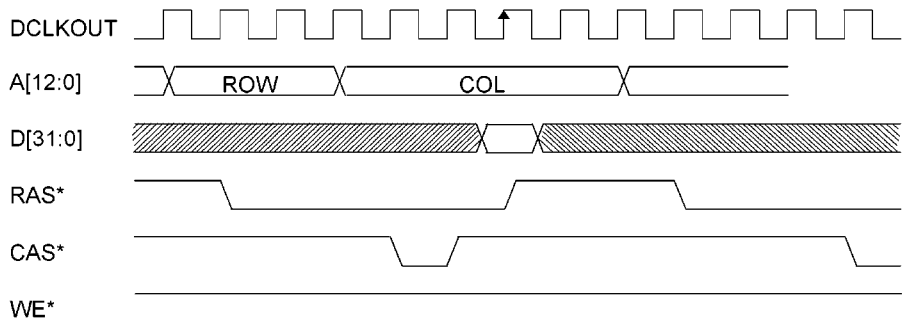


Figure 4-16a EDO READ

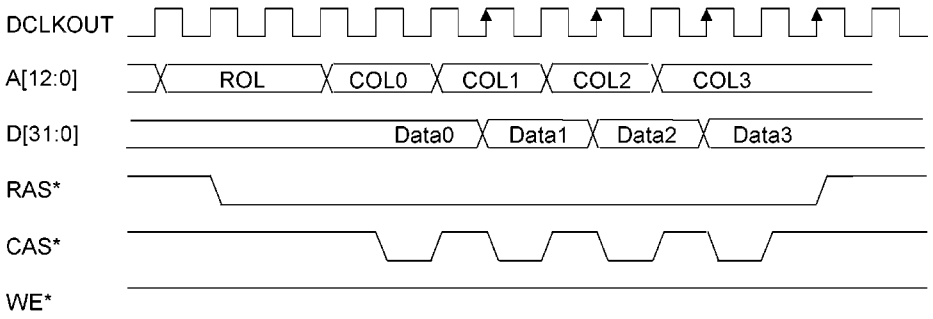


Figure 4-16b EDO Burst Read

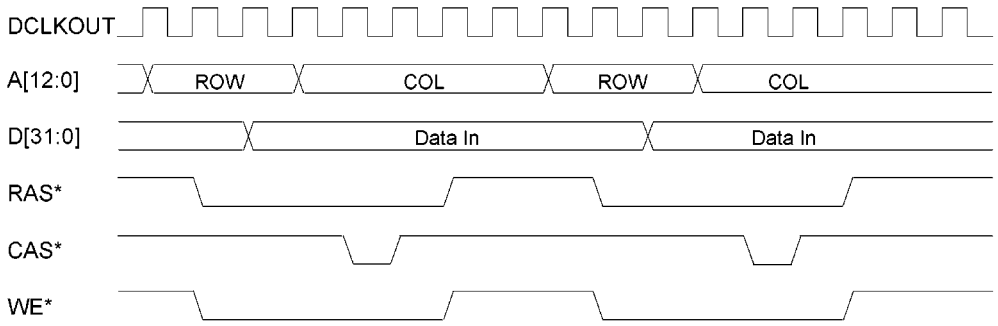


Figure 4-17a EDO Write (1word)

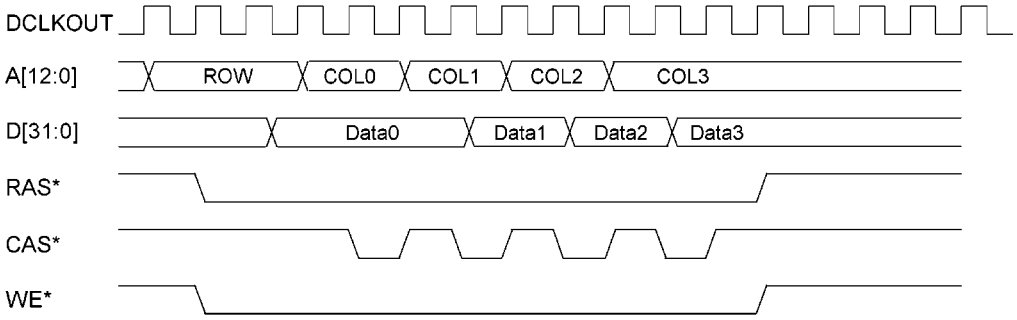


Figure 4-17b EDO In-Page Write

4.4.4 DRAM Initialization

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

4.4.5 SDRAM

SDRAM devices are supported in Bank 1 and Bank 0. Each Bank can independently be configured to support either 16-bit or 32-bit SDRAMs. Bank 1 uses RAS1* as the chip select for the SDRAM and Bank 0 uses DCS0*. All other control signals are shared between the two banks.

The clock for the SDRAM devices is the DCLKOUT signal which nominally runs at 73.728 MHz. Reads from the SDRAMs occur on every DCLKOUT cycle, such that the reading of all 32 bits for 32-bit SDRAMs can happen within a single DCLKOUT cycle. A 32-bit SDRAM configuration provides the maximum memory bandwidth and once a read cycle begins, the CPU will not stall since 32 bits are read on each cycle. All 16-bit SDRAMs require 2 DCLKOUT cycles to read 32 bits of data. During a burst read the SDRAM devices are kept in page mode for the entire read cycle.

The SDRAM devices are also kept in page mode during in-page write cycles. All 32-bit SDRAMs can keep up at 32 bits per DCLKOUT cycle, which can provide a long write block without stalling the CPU. Tri, Half-Word and Byte writes are supported to the SDRAM devices using the DQMH and DQML signals in the 16-bit SDRAM configuration and the DQMH, the DQML, the CAS0* and CAS1* signals in the 32-bit SDRAM configuration. The Table 4-9 shows the Mask signal configuration. These signals provide a data mask so that only the desired bytes will be written. The data mask signals are required because the SDRAM is setup to always read or write 32 bits of data at a time. The data mask signals will "mask out" the bytes that should not be written during partial Word stores.

Table 4-9 Mask signal configuration (Little Endian)

	16 bit	32 bit
DQMH	D[31:24]	D[31:24]
DQML	D[23:16]	D[23:16]
CAS1*		D[15:8]
CAS0*		D[7:0]

4.4.6 SDRAM Initialization

Prior to accessing an SDRAM device, an initialization procedure must be followed. First, all Memory Configuration Registers must be properly initialized for the chosen SDRAM and bus configuration. Next, after a pause of at least 100 μs after power is applied to the SDRAM, a command must be sent to precharge the SDRAM banks. The command is sent by writing to the SDRAM Bank 1 and/or Bank 0 Mode Register address with Data[7] set high. Setting this bit will cause a Precharge All command to be sent to the SDRAM instead of writing to the Mode Register. Next, the Mode Register must be properly initialized. Finally, refresh should be enabled and at least 2 refresh cycles must occur prior to reading or writing from or to the SDRAM.

4.4.7 SDRAM Mode Register

An SDRAM contains a write-only Mode Register that defines the parameters for accessing the device. The mode register must be initialized prior to reading or writing from the SDRAM. They can be written via the SDRAM Bank 1 and SDRAM Bank 0 addresses as shown in the System Address Map. When writing to the mode register, the data from the CPU Interface on Data[12:0] is placed on A[12:0], and Data[31] must be a zero. This is done because the Mode Register receives it's data on the address bus instead of the data bus. The SDRAM should be configured for the following settings: Burst Length = 1 (for 32-bit SDRAM) or Burst Length = 2 (for 16-bit SDRAM), CAS Latency = 3, Addressing Mode = Sequential, Single Write Mode = Burst read and Burst write.

4.4.8 SDRAM Power Down Mode

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

4.4.9 SDRAM Timing

Figures 4-18 through 4-22 show a few examples of the SDRAM timing. The first chip select (assertion of CS*) is used to latch the ROW address. Subsequent chip selects will either read or write data, and the final chip select is used to precharge the bank. Note from Table 4-8 that address bit 10 for 16/64-Mbit parts is always a zero during the column address. This is done to ensure that auto precharge is never latched during a read or write command and also will cause only the bank that is being accessed to be precharged during the precharge command. Since the burst size of the SDRAMs is always setup to 32 bits, reads will always be 32 bits without regard to the actual number of bytes requested by the CPU. Writes are also 32 bits, but the data mask signals DQMH, DQML, CAS0* and CAS1* (32-bit) will mask out the bytes that should not be written, thus allowing for Tri, Half-Word and Byte writes.

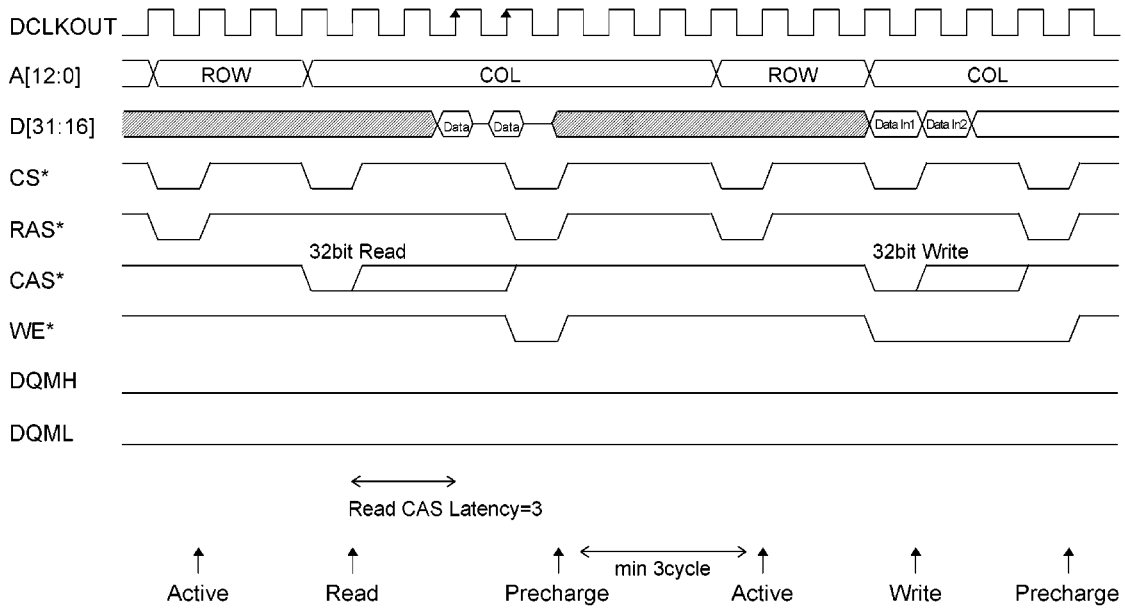


Figure 4-18 Word Read and Write to 16-Bit SDRAM

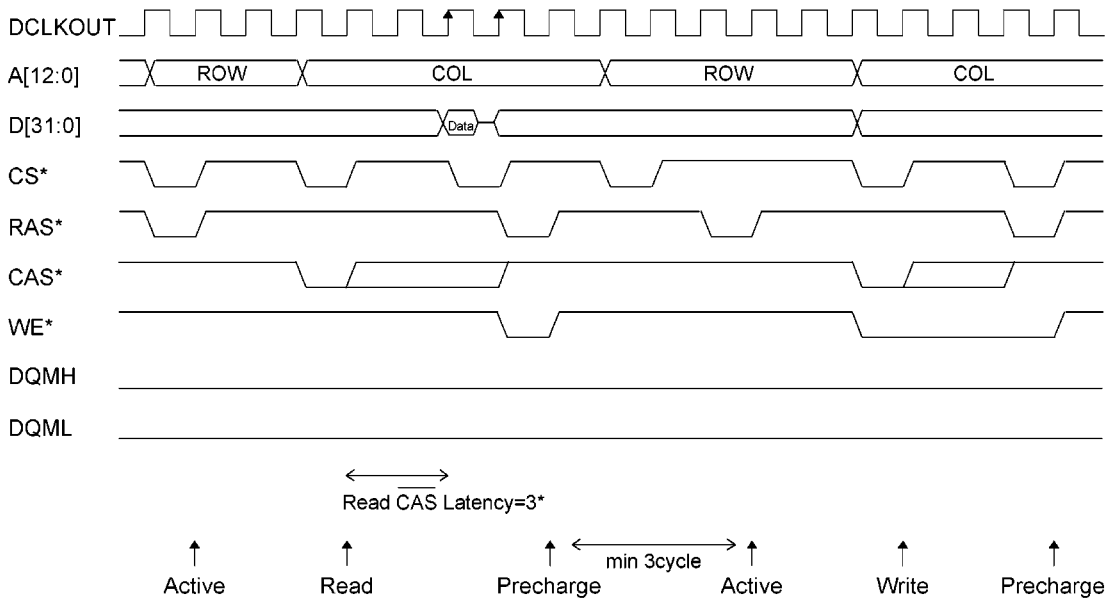


Figure 4-19 Word Read and Write to 32-bit

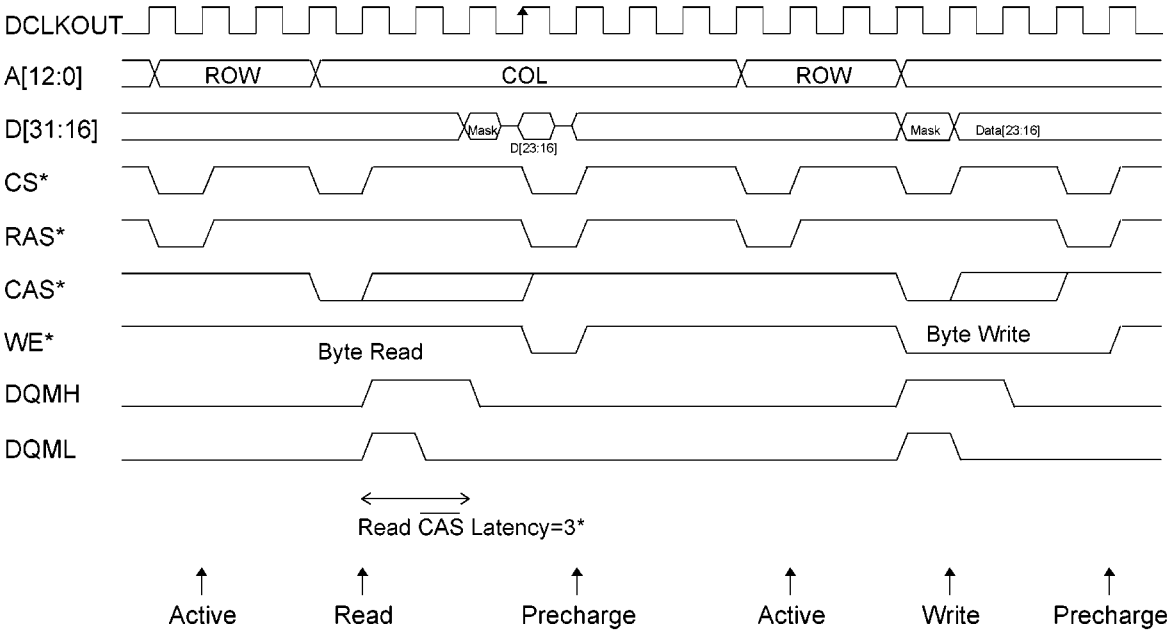


Figure 4-20 Byte Read and Byte Write (16-bit SDRAM)

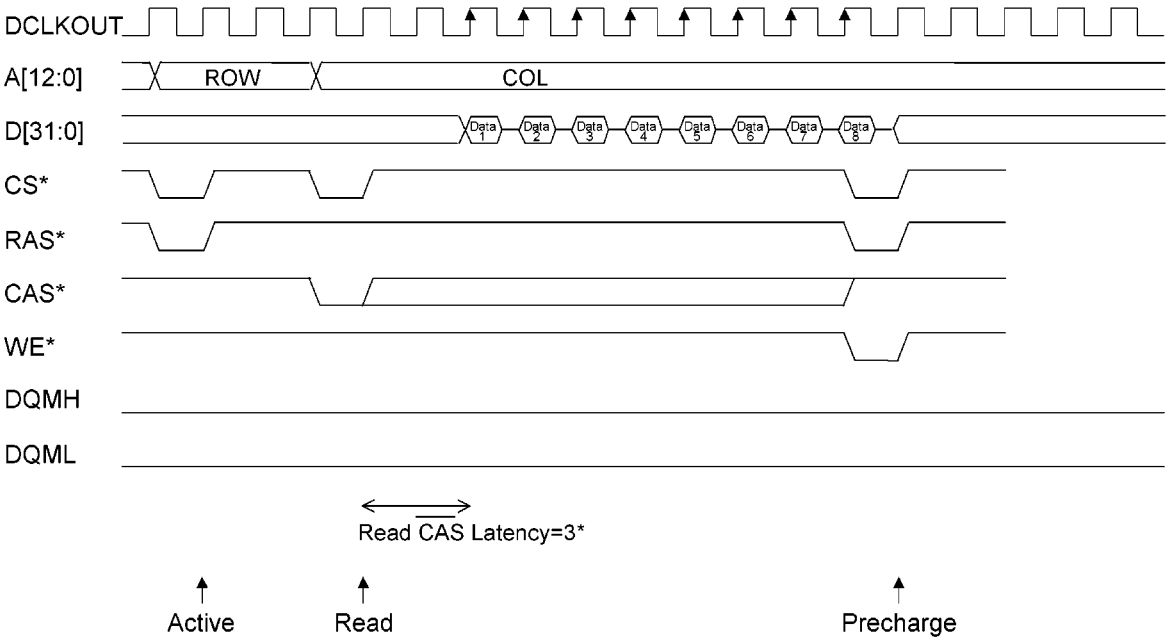


Figure 4-21 8 Word Burst Read to 32-bit SDRAM

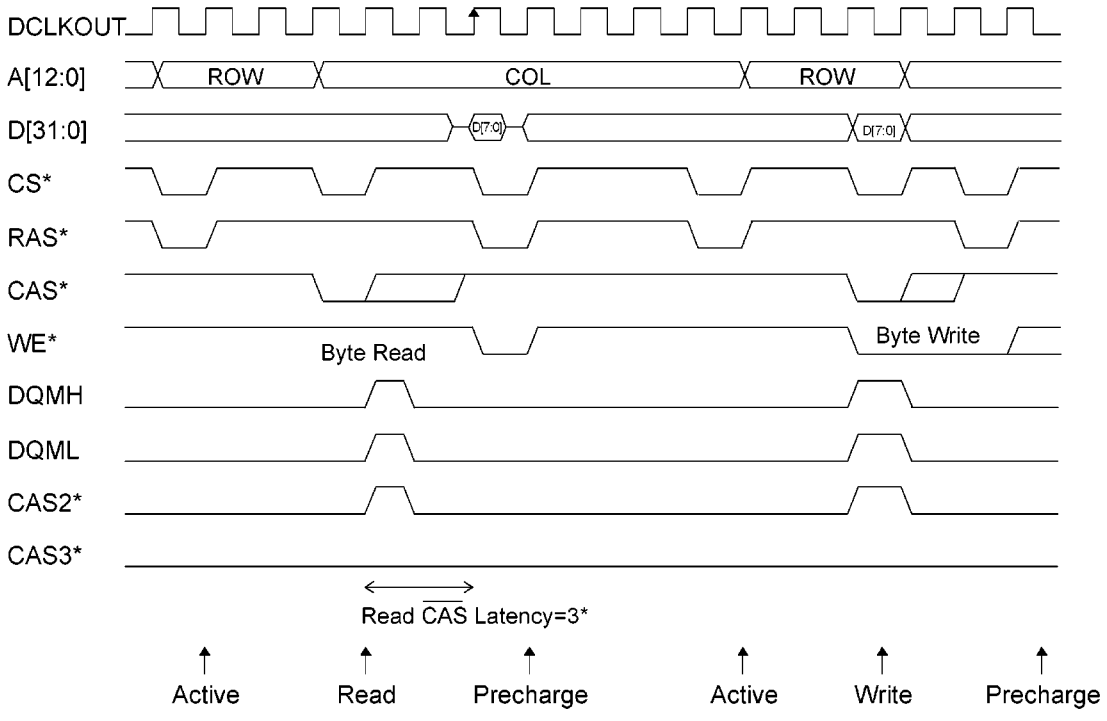


Figure 4-22 Byte Read and Byte Write (32-bit SDRAM)

4.5 Arbitration

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

4.5.1 Refresh Controller

The Refresh Controller logic contains separate refresh counters for Bank 1 and Bank 0. Each counter is a 6-bit counter that uses the 32 kHz clock for the counter. The refresh rate is set using the RFSHVAL1[5:0] and RFSHVAL2[5:0] control bits in the Memory Configuration 4 Register. These values are loaded into the respective refresh counter. The refresh counter will down count and when it reaches a count of zero, refresh will be inserted. If the RFSHVALx control bits are set to $00, the counter is not used and instead refresh is generated based on both the rising and falling edge of the 32 kHz clock. This is required to generate a 15.26 μs refresh rate required by some DRAMs or SDRAMs. RFSHVALx control bits set to $01 will cause a refresh rate of 61.04 μs and additional values will increase the refresh rate by increments of 30.52 μs. When it is time for the Refresh Controller to insert refresh, the Halt signal will be asserted to the CPU Interface. Once the CPU Interface asserts the Halted signal, the Refresh Controller will generate the appropriate control signals that will cause the DRAM & SDRAM control logic to provide refresh for the memory devices. Once the Refresh Controller has completed inserting refresh, the Halt signal will be de-asserted so that the CPU Interface can continue running.

4.5.2 External Bus Master

The External Bus Master logic allows an external device to take control of the memory signals so that memory can be shared with an external device. The external device will request the bus by asserting the DREQ* signal. Once the DREQ* signal is asserted, the Halt signal will be asserted to the CPU Interface logic. Once the CPU Interface logic asserts the Halted signal, the External Bus Master logic will then assert the DGRNT* signal to inform the external device that it now controls the memory bus. At the same time that DGRNT* is asserted, the following signals are tri-stated: D[31:0], A[12:0], ALE, RD*, WE*, CAS3*-CAS0*, RAS1*, RAS0*, DCS0*, DCKE, DQMH, DQML and CS0*. DCLKOUT will be tri-stated if the ENDCLKOUTTRI control bit is set in the Memory Configuration 0 Register. Otherwise, DCLKOUT will continue to run. Once the external device de-asserts DREQ*, DGRNT* will be de-asserted, the memory signals will be driven by the TMPR3922, and the Halt signal will be de-asserted so the CPU Interface can begin launching memory transactions. The external device should not attempt to take over the memory signals for a long period of time, otherwise TMPR3922 DMA events and refresh will be interrupted. The External Bus Master logic will not respond with a DGRNT* if the memory interface is powered down.

4.5.3 Watch Dog Timer

The Watch Dog Timer prevents accesses to reserved memory locations from hanging the BIU. The Watch Dog Timer consists of a 10-bit down-counter whose initial value is determined by the WATCHTIMEVAL[3:0] control bits in the Memory Configuration 4 Register. The WATCHTIMEVAL[3:0] control bits are loaded into the upper 4 bits of the 10-bit counter and the lower 6 bits are loaded with $3F. The counter uses the CLK signal (nominally 36.864 MHz) to count. This will imply a maximum count of 28μs if the WATCHTIMEVAL[3:0] control bits are set to $F. The Watch Dog Timer rate is determined by the following equation:

$$\text{Watch Dog Timer} = \frac{(\text{WATCHTIMEVAL[3:0]} + 1)}{36.864}$$

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

4.5.4 Memory Power Down

The memory interface is powered down whenever the MEMPOWERDOWN control bit is set in the Memory Configuration 4 Register. When this bit is set, the Halt signal will be asserted to the CPU Interface, the DRAM and/or SDRAM will be placed into self refresh mode, and all memory interface signals will be driven low. The memory interface will wake up whenever the CPU Interface asserts the HaltCycle signal and the address defined by the Addr[31:2] bits correspond to an address other than internal function registers. When the memory interface wakes up, the DRAM and/or SDRAM are taken out of self refresh mode and the Halt signal is de-asserted to allow the CPU Interface to launch memory transactions. Figures 4-23 and 4-24 show timing for the wake-up and power-down of the memory interface.

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

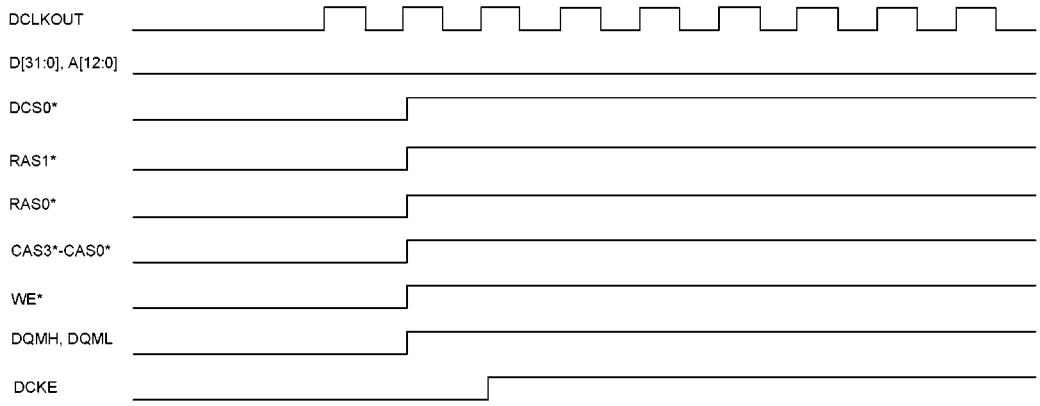


Figure 4-23 Wake-up

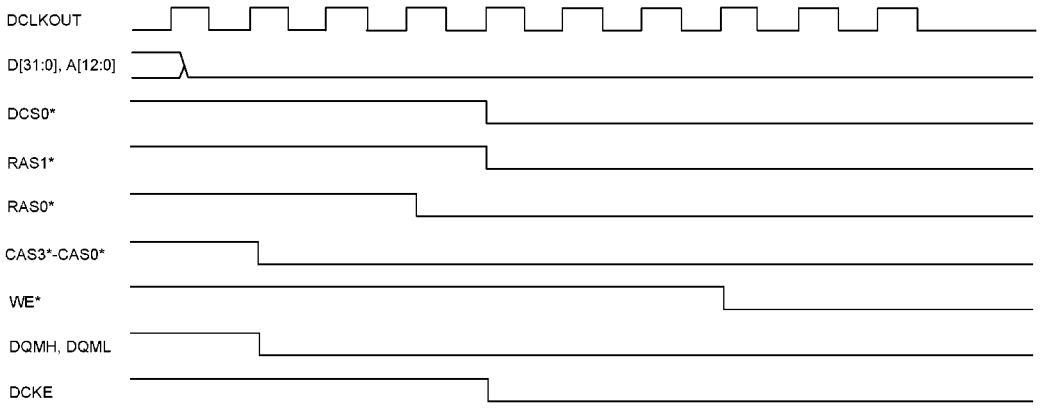


Figure 4-24 Power down

4.6 Memory Connections

Figures 4-25 through 4-28 illustrate the various signal connections between the TMPR3922 and external SDRAM, DRAM, and static memory devices. These figures show the connections for various bus width configurations (e.g., 16-bit, 32-bit).

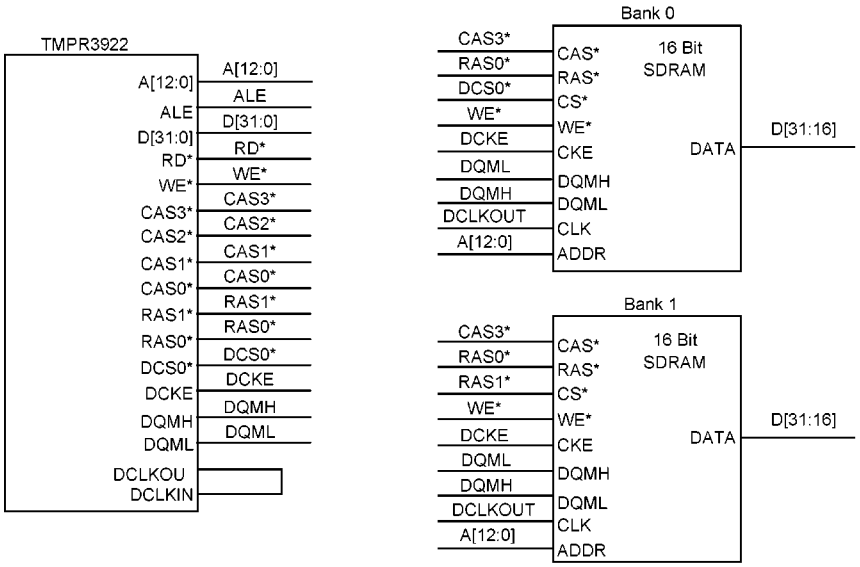


Figure 4-25 SDRAM Memory Connection

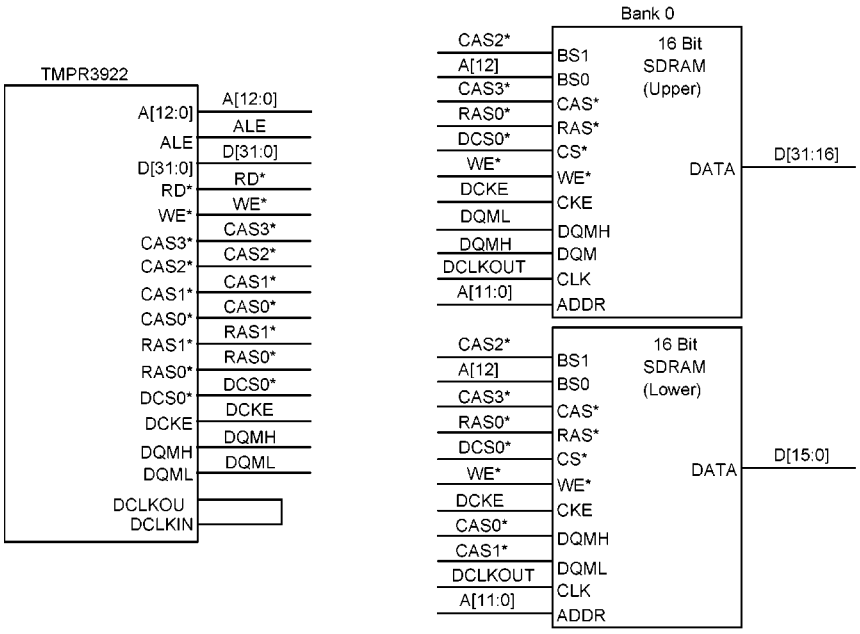


Figure 4-26 32bit-SDRAM Memory Connection

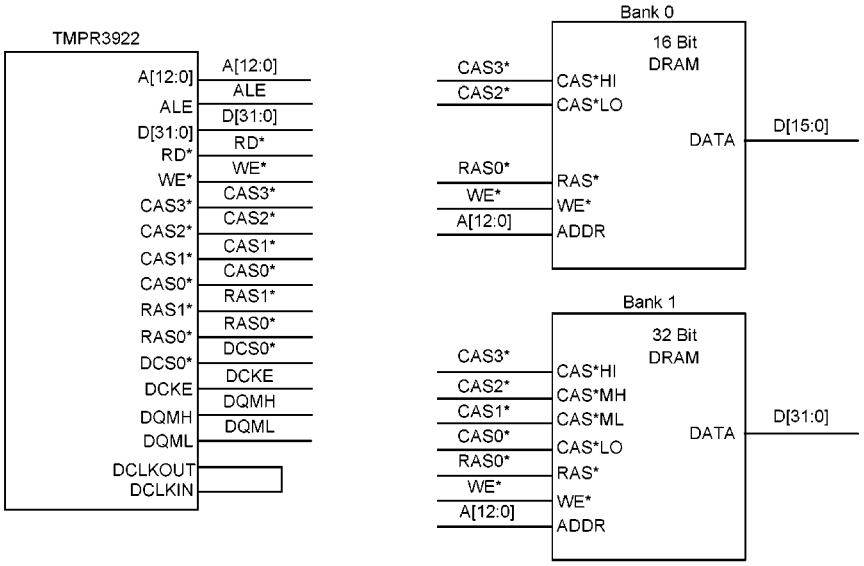


Figure 4-27 DRAM Memory Connection

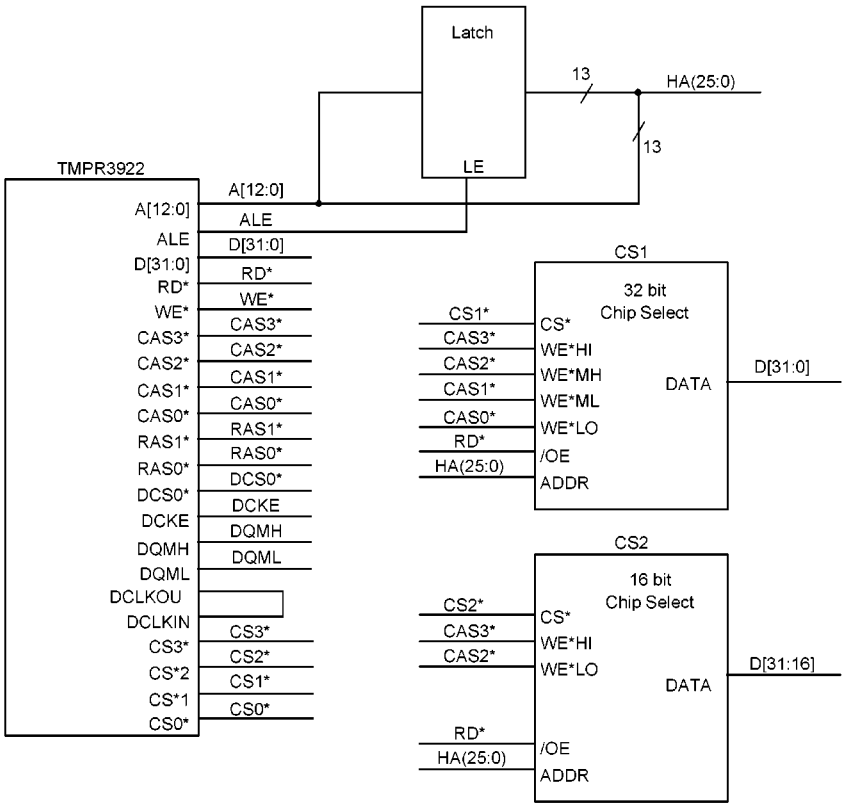


Figure 4-28 Chip Select Memory Connection

4.7 BIU Registers

4.7.1 Memory Configuration 0 Register

OFFSET=$000:

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

ENDCLKOUTTRI:

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

DISDQMINIT:

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

ENSDRAMPD:

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

SHOWDINO:

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

ENRMAP2:

Setting this bit will enable Address Re-Mapper 2.

ENRMAP1:

Setting this bit will enable Address Re-Mapper 1.

ENWRINPAGE:

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

ENCS3USER:

Setting this bit will enable CS3 in the kuseg Space.

ENCS2USER:

Setting this bit will enable CS2 in the kuseg Space.

ENCS1USER:

Setting this bit will enable CS1 in the kuseg Space.

ENCS1DRAM:

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

BANK1CONF[1:0]:

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

11 32-bit SDRAM
10 16-bit SDRAM
01 32-bit DRAM/EDO
00 16-bit DRAM/EDO

BANK0CONF[1:0]:

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

11 32-bit SDRAM
10 16-bit SDRAM
01 32-bit DRAM/EDO
00 16-bit DRAM/EDO

ROWSEL1[1:0]:
ROWSEL0[1:0]:
COLSEL1[3:0]:
COLSEL0[3:0]:

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

ROW	ROWSEL	COL	COLSEL
18, 17:9	00	22, 20, 18, 8:1	0000
22, 18, 20, 19, 17:9	01	19, 18, 8:2	0001
20, 22, 21, 19, 17:9	10	21, 20, 18, 8:2	0010
22, 23, 21, 19, 17:9	11	23, 22, 20, 18, 8:2	0011
—	—	24, 22, 20, 18, 8:2	0100
—	—	22, 18, ϕ, 24, 23, 21, 8:2	0101
—	—	22, X, ϕ, X, 24, 21, 8:2	0110
—	—	22, 18, ϕ, 23, 21, 8:1	0111
		22, X, ϕ, 24, 21, 8:1	1000
—	—	24, 23, 21, 8:2	1001
25:13	—	12:0	NOT DRAM

CS3SIZE:

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

CS2SIZE:

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

CS1SIZE:

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

CS0SIZE:

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

4.7.2 Memory Configuration 1 Register

OFFSET=$004

Bit	Label	RESET	Read/Write
31	Reserved	0	R/W
30	C48MPLLON	0	R/W
29	ENMCS1BE	0	R/W
28	ENMCS0BE	0	R/W
27	ENMCS1ACC	0	R/W
26	ENMCS0ACC	0	R/W
25-23	BCLKDIV[2:0]	0	R/W
22	ENBCLK	0	R/W
21	ENMCS1PAGE	0	R/W
20	ENMCS0PAGE	0	R/W
19	ENMCS1WAIT	0	R/W
18	ENMCS0WAIT	0	R/W
17	MCS1_32	0	R/W
16	MCS0_32	0	R/W
15-12	MCS1ACCVAL1[3:0]	X	R/W
11-8	MCS1ACCVAL2[3:0]	X	R/W
7-4	MCS0ACCVAL1[3:0]	X	R/W
3-0	MCS0ACCVAL2[3:0]	X	R/W

C48MPLLON:

Setting this bit will enable the PLL for 48 MHz.

ENMCS1BE:

Setting this bit will enable the Byte enable signals (CAS3-0*) during the MCS1 chip select device access.

ENMCS0BE:

Setting this bit will enable the Byte enable signals (CAS3-0*) during the MCS0 chip select device access.

ENMCS1ACC:

Setting this bit select the MCS1 access timing. Please refer to Figure 4-3.

ENMCS0ACC:

Setting this bit select the MCS0 access timing. Please refer to Figure 4-3.

BCLKDIV[2:0]:

These bits select the start count value and the stop count value for the 3-bit programmable counter used to generate BCLK; these clocks are derived by dividing down FREECLK. Since the MSB of the counter output is used for these clocks, the start count and stop count values are chosen to provide (as close as possible) a 50% duty cycle for these clocks. The table used to compute these counter start and stop values are as follows:

BCLKDIV	start value	stop value	divide-modulus
0	3	4	2
1	2	4	3
2	2	5	4
3	1	5	5
4	1	6	6
5	0	6	7
6	0	7	8
7	—	—	1

ENBCLK:

Setting this bit will enable BCLK as the reference clock for the external device..

ENMCS1PAGE:

Setting this bit will enable Read Page mode and Write Burst mode for MCS1 .

ENMCS0PAGE:

Setting this bit will enable Read Page mode and Write Burst mode for MCS0 .

ENMCS1WAIT:

Setting this bit will enable the MCS1WAIT* signal .

ENMCS0WAIT:

Setting this bit will enable the MCS0WAIT* signal.

MCS1_32:

Setting this bit defines MCS1 to be a 32-bit port. Otherwise, a 16-bit port is assumed.

MCS0_32:

Setting this bit defines MCS0 to be a 32-bit port. Otherwise, a 16-bit port is assumed.

MCS1ACCVAL1[3:0]:

These bits define the access time for MCS1. If ENMCS1BURST is set, these bits will be used for the first access and MCS1ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS1BURST is not set, these bits and MCS1ACCVAL2[3:0] should be set with the same value. If ENMCS1BURST is set, these bits must be set to at least $1.

MCS1ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS1BURST is set. If ENMCS1BURST is not set, these bits and MCS1ACCVAL1[3:0] should be set with the same value. If ENMCS1BURST is set, these bits must be set to at least $1.

MCS0ACCVAL1[3:0]:

These bits define the access time for MCS0. If ENMCS0BURST is set, these bits will be used for the first access and MCS0ACCVAL2[3:0] will be used for the next three accesses for reads only. For writes, only these bits are used. If ENMCS0BURST is not set, these bits and MCS0ACCVAL2[3:0] should be set with the same value. If ENMCS0BURST is set, these bits must be set to at least $1.

MCS0ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENMCS0BURST is set. If ENMCS0BURST is not set, these bits and MCS0ACCVAL1[3:0] should be set with the same value. If ENMCS0BURST is set, these bits must be set to at least $1.

4.7.3 Memory Configuration 2 Register

OFFSET=$008

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

CS3ACCVAL1[3:0]:

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

CS3ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS3PAGE is set. If ENCS3PAGE is not set, these bits and CS3ACCVAL1[3:0] should be set with the same value. If ENCS3PAGE is set, these bits must be set to at least $1.

CS2ACCVAL1[3:0]:

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

CS2ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS2PAGE is set. If ENCS2PAGE is not set, these bits and CS2ACCVAL1[3:0] should be set with the same value. If ENCS2PAGE is set, these bits must be set to at least $1.

CS1ACCVAL1[3:0]:

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

CS1ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS1PAGE is set. If ENCS1PAGE is not set, these bits and CS1ACCVAL1[3:0] should be set with the same value. If ENCS1PAGE is set, these bits must be set to at least $1.

CS0ACCVAL1[3:0]:

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

CS0ACCVAL2[3:0]:

These bits define the access time for the next three reads in a burst sequence if ENCS0PAGE is set. If ENCS0PAGE is not set, these bits and CS0ACCVAL1[3:0] should be set with the same value. If ENCS0PAGE is set, these bits must be set to at least $1.

4.7.4 Memory Configuration 3 Register

OFFSET=$00C

Bit	Label	RESET	Read/Write
31-28	CARD2ACCVAL[3:0]	X	R/W
27-24	CARD1ACCVAL[3:0]	X	R/W
23-20	CARD2IOACCVAL[3:0]	X	R/W
19-16	CARD1IOACCVAL[3:0]	X	R/W
15	Reserved		
14	Reserved		
13	Reserved		
12	Reserved		
11	ENCS3PAGE	X	R/W
10	ENCS2PAGE	X	R/W
9	ENCS1PAGE	X	R/W
8	ENCS0PAGE	0	R/W
7	CARD2WAITEN	X	R/W
6	CARD1WAITEN	X	R/W
5	CARD2IOEN	X	R/W
4	CARD1IOEN	X	R/W
3	CARD2_8SEL	0	R/W
2	CARD1_8SEL	0	R/W
1-0	Reserved		

CARD2ACCVAL[3:0]:

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

CARD1ACCVAL[3:0]:

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

CARD2IOACCVAL[3:0]:

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

CARD1IOACCVAL[3:0]:

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

ENCS3PAGE:

Setting this bit will enable Read Page Mode for CS3.

ENCS2PAGE:

Setting this bit will enable Read Page Mode for CS2.

ENCS1PAGE:

Setting this bit will enable Read Page Mode for CS1.

ENCS0PAGE:

Setting this bit will enable Read Page Mode for CS0.

CARD2WAITEN:

Setting this bit will enable the CARD2WAIT* signal.

CARD1WAITEN:

Setting this bit will enable the CARD1WAIT* signal.

CARD2IOEN:

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

CARD1IOEN:

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

CARD2_8SEL:

This bit defines Card 2 port size.

0: 16 bit port access
1: 8 bit port access

CARD1_8SEL:

This bit defines Card 1 port size.

0: 16 bit port access
1: 8 bit port access

4.7.5 Memory Configuration 4 Register

OFFSET=$010

Bit	Label	RESET	Read/Write
31	ENBANK1EDO	0	R/W
30	ENBANK0EDO	0	R/W
29	ENARB	0	R/W
28	DISSNOOP	0	R/W
27	CLRWRBUSERRINT	0	R/W
26	ENBANK1OPT	0	R/W
25	ENBANK0OPT	0	R/W
24	ENWATCH	0	R/W
23-20	WATCHTIMEVAL[3:0]	X	R/W
19-17	Reserved		
16	MEMPOWERDOWN	1	R/W
15	ENRFSH1	0	R/W
14	ENRFSH0	0	R/W
13-8	RFSHVAL1[5:0]	X	R/W
7-6	Reserved		
5-0	RFSHVAL0[5:0]	X	R/W

ENBANK1EDO:

Setting this bit will enable the TMPR3922 to support EDO devices in Memory Bank 1.
The BANK1CONF bits in the Memory Configuration 0 Register will configure the TMPR3922 to support either a 16-bit EDO configuration or a 32-bit EDO configuration, in the same manner that standard DRAM is setup.

ENBANK0EDO:

Setting this bit will enable the TMPR3922 to support EDO devices in Memory Bank 0.
The BANK0CONF bits in the Memory Configuration 0 Register will configure the TMPR3922 to support either a 16-bit EDO configuration or a 32-bit EDO configuration, in the same manner that standard DRAM is setup.

ENARB:

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

DISSNOOP:

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

CLRWRBUSERRINT:

WRBUSERRINT will occur whenever a write is timed out by the Watch Dog Timer.
This interrupt is connected directly to INT[0] on the CPU and is completely independent of the rest of the TMPR3922 interrupt logic. When this interrupt is set, it will remain set until the CLRWRBUSERRINT bit is asserted. The bit must be set and subsequently cleared to clear the interrupt.

ENBANK1OPT:

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

ENBANK0OPT:

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

ENWATCH:

Setting this bit will enable the Watch Dog Timer.

WATCHTIMEVAL[3:0]:

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

$$\text{Watch Dog Rate} = \frac{(\text{WATCHTIME}[3:0] + 1) * 64}{36.864\ \text{MHz}}$$

MEMPOWERDOWN:

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

ENRFSH1:

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

ENRFSH0:

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

RFSHVAL1[5:0]:

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

$0 15.26 µs
$1 30.52 µs
$2 61.04 µs
$3 91.55 µs
$4 122.07 µs

RFSHVAL0[5:0]:

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

$0 15.26 µs
$1 30.52 µs
$2 61.04 µs
$3 91.55 µs
$4 122.07 µs
:

4.7.6 Memory Configuration 5 Register

OFFSET=$014: write-only

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

STARTVAL2[31:9]: write-only

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

MASK2[3:0]: write-only

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

4.7.7 Memory Configuration 6 Register

OFFSET=$018: write-only

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

STARTVAL1[31:9]: write-only

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

MASK1[3:0]: write-only

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

4.7.8 Memory Configuration 7 Register

OFFSET=$01C: write-only

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

RMAPADD2[31:9]: write-only

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

4.7.9 Memory Configuration 8 Register

OFFSET=$020: write-only

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

RMAPADD1[31:9]: write-only

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

SECTION 5 SIU Module

5.1 Overview

The SIU Module within the TMPR3922 provides the multi-channel 32-bit DMA controller used to select and arbitrate the DMA channels from each SIM DMA source. Independent DMA channels are provided for IrDA, sound, telecom, CHI, UARTA, and UARTB sources.

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

5.2 Implementation

5.2.1 Block Diagram

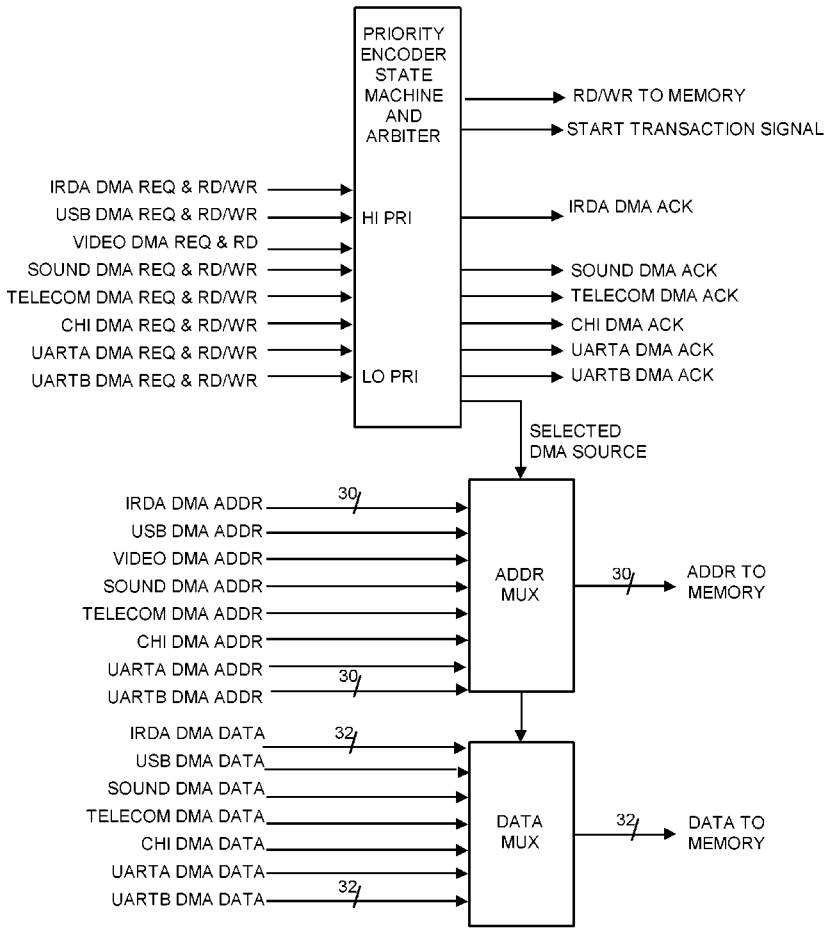


Figure 5-1 SIU DMA Controller Block Diagram

5.2.2 DMA Controller Description

Figure 5-1 shows a block diagram of the SIU DMA Controller. The DMA Controller receives the DMA requests and read/write status from all of the 6 possible DMA sources (IrDA, sound, telecom, CHI, UARTA, and UARTB) and uses a Priority Encoder to arbitrate and select the highest priority request. If a new request is received and the DMA controller is currently busy, that request will remain pending and will be processed as soon as all other higher priority pending requests have been processed. The priority of the DMA sources and the DMA direction (read or write) supported are shown in Table 5-1.

Table 5-1 DMA Sources

DMA Source	priority	DMA read (from memory)	DMA write (to memory)
sound	Highest	yes	yes
telecom		yes	yes
IrDA		yes	yes
CHI		yes	yes
UART-A (1 byte)		yes	yes
UART-B (1 byte)	lowest	yes	yes

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

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

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

5.2.3 Address Decoder Description

Figure 5-2 shows a block diagram of the SIM Address Decoder. The Address Decoder generates the read and write enable pulses for each of the internal TMPR3922 control and status registers. The lower 8 bits of the internal address bus are decoded, along with the SIM chip select and read/write status signal, to generate any 1 of 256 possible write enable pulses or any 1 of 256 possible read enable pulses. These enable pulses are used by the circuit/module targeted by the respective read or write pulse to either latch the data from the CPU (for a write transaction) or to enable the data to the CPU (for a read transaction). See the next section for a full listing of all the internal function Registers, including their address offset and read versus write support. The address offset for function Registers is the offset from the base address ($10C00000) of the function Registers in the System Address Map.

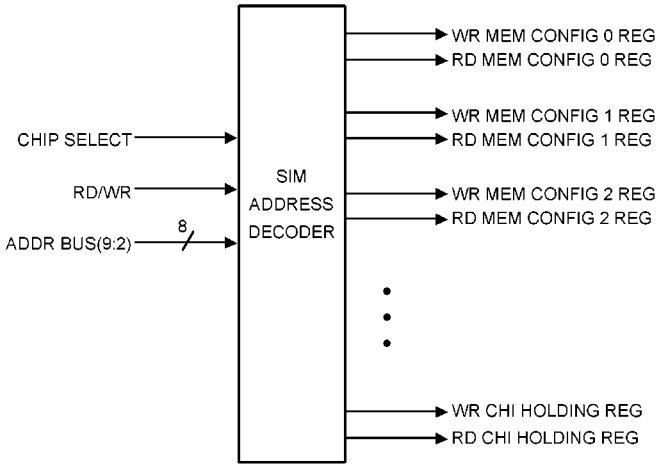


Figure 5-2 SIM Address Decoder Block Diagram

5.2.4 Internal Function Registers

Offset	Read/Write	Register
000	R/W	Memory Configuration 0
004	R/W	Memory Configuration 1
008	R/W	Memory Configuration 2
00C	R/W	Memory Configuration 3
010	R/W	Memory Configuration 4
014	W	Memory Configuration 5
018	W	Memory Configuration 6
01C	W	Memory Configuration 7
020	W	Memory Configuration 8
028	R	Infrared Ring Pointer Readback
02C	R/W	Infrared Ring Base Address
030	R/W	Infrared Ring Size
034	W	Infrared Ring Prompt
038	R/W	Infrared Config 0
03C	R/W	Infrared Config 1
040	R	Infrared Sir Flags
044	R/W	Infrared Enable Register
048	R	Config to Phy Register
04C	R/W	Infrared Phy Config
050	R/W	Maximum Packet Length
054	R	Receive Byte Count
058	R/W	Unicast Filtering Address 1-4
05C	R	Infrared Status
060	W	SIB Size
064	W	SIB Sound Receive Start
068	W	SIB Sound Transmit Start
06C	W	SIB Tel Receive Start
070	W	SIB Tel Transmit Start
074	R/W	SIB Control
078	R/W	SIB Sound Holding Register
07C	R/W	SIB Tel Holding Register
080	R/W	SIB SF0 Control
084	R/W	SIB SF1 Control
088	R	SIB SF0 Status
08C	R	SIB SF1 Status
090	R/W	SIB DMA Control
0A0	R/W	IR Control 1
0A4	W	IR Control 2
0A8	W	IR Holding Register
0B0	R/W	UARTA Control 1
0B4	W	UARTA Control 2
0B8	W	UARTA DMA Control 1
0BC	W	UARTA DMA Control 2

Offset	Read/Write	Register
0C0	R	UARTA DMA Count
0C4	R/W	UARTA Holding Register
0C8	R/W	UARTB Control 1
0CC	W	UARTB Control 2
0D0	W	UARTB DMA Control 1
0D4	W	UARTB DMA Control 2
0D8	R	UARTB DMA Count
0DC	R/W	UARTB Holding Register
100	W	Clear Interrupt 1
100	R	Interrupt Status 1
104	W	Clear Interrupt 2
104	R	Interrupt Status 2
108	W	Clear Interrupt 3
108	R	Interrupt Status 3
10C	W	Clear Interrupt 4
10C	R	Interrupt Status 4
110	W	Clear Interrupt 5
110	R	Interrupt Status 5
114	R	Interrupt Status 6
118	R/W	Enable Interrupt 1
11C	R/W	Enable Interrupt 2
120	R/W	Enable Interrupt 3
124	R/W	Enable Interrupt 4
128	R/W	Enable Interrupt 5
12C	R/W	Enable Interrupt 6
130	W	Clear Interrupt 7
130	R	Interrupt Status 7
134	R/W	Enable Interrupt 7
138	W	Clear Interrupt 8
138	R	Interrupt Status 8
13C	R/W	Enable Interrupt 8
140	R	RTC High
144	R	RTC Low
148	R/W	Alarm High
14C	R/W	Alarm Low
150	R/W	Timer Control
154	R/W	Periodic Timer
160	R/W	SPI Control
164	R/W	SPI Holding Register
180	R/W	IO Control
184	R/W	Multi-function IO Data Out
188	R/W	Multi-function IO Direction
18C	R	Multi-function IO Data In
190	R/W	Multi-function IO Select

Offset	Read/Write	Register
194	R/W	IO Power Down
198	R/W	Multi-function IO Power Down
19C	R/W	IO Data In/Out
1C0	R/W	Clock Control0
1C4	R/W	Power Control
1C8	R/W	SIU Test
1D8	R/W	CHI Control
1DC	R/W	CHI Pointer Enable
1E0	W	CHI Receive Pointer A
1E4	W	CHI Receive Pointer B
1E8	W	CHI Transmit Pointer A
1EC	W	CHI Transmit Pointer B
1F0	R/W	CHI Size
1F4	W	CHI Receive Start
1F8	W	CHI Transmit Start
IFC	R/W	CHI Holding Register
208	R	TMPR3922 revision

5.3 SIU Registers

5.3.1 SIU Test Register

OFFSET=$1C8:

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

ENDMATEST:

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

ENNOTIMETEST:

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

DMATESTWR[5:0]:

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

5.3.2 TMPR3922 Revision

OFFSET=$208

Bit	Label	RESET	Read/Write
31-8	Reserved	—	
7-0	Revision [7:0]	$10	R

SECTION 6 Clock Module

6.1 Overview

The Clock Module within the TMPR3922 contains logic for generating the clocks for all other internal TMPR3922 modules, as well as for generating certain externally driven TMPR3922 output clocks. The Clock Module contains dividers for generating the correct rates for each clock and also contains logic for independently enabling or disabling individual clocks under software control.

6.1.1 Related Pins

SYSCLKIN: INPUT

This pin should be connected along with SYSCLKOUT to an external crystal which is the main TMPR3922 clock source.

SYSCLKOUT: OUTPUT

This pin should be connected along with SYSCLKIN to an external crystal which is the main TMPR3922 clock source.

C6MIN: INPUT

This pin should be connected along with C6MOUT to an external crystal which is the IrDA/USB clock source.

C6MOUT: OUTPUT

This pin should be connected along with C6MIN to an external crystal which is the IrDA/USB clock source.

C48MOUT: OUTPUT

This pin is a buffered output of the 48MHz clock.

6.2 Implementation

6.2.1 Block Diagram

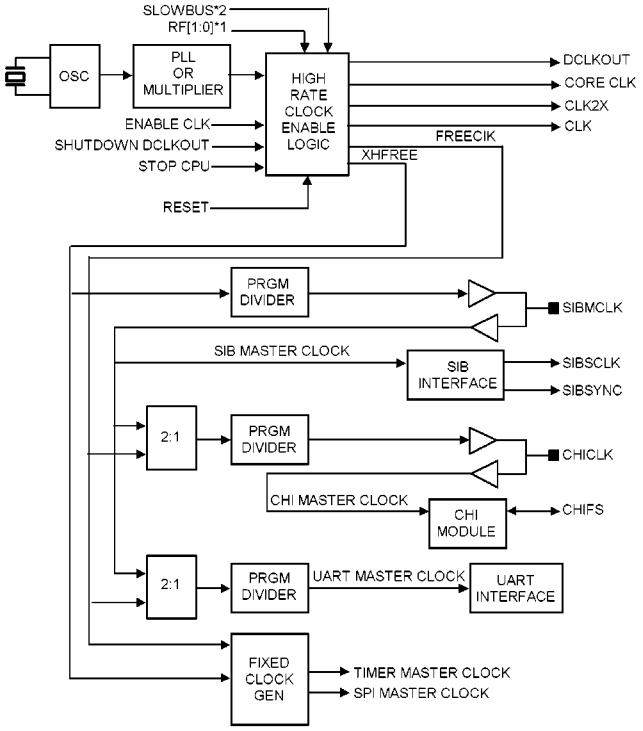


Figure 6-1b Clock Module Block Diagram

RF[1:0]*1	SLOWBUS*2	CORECLK	DCLKOUT	CLK2X	FREECLK	CLK	XHFREE
00	0	F	F/2	F/2	F/2	F/4	F/4
01	0	F/2	F/4	F/4	F/2	F/8	F/4
10	0	F/4	F/8	F/8	F/2	F/16	F/4
11	0	F/8	F/16	F/16	F/2	F/32	F/4
00	1	F/2	F/4	F/4	F/2	F/8	F/4
01	1	F/4	F/8	F/8	F/2	F/16	F/4
10	1	F/8	F/16	F/16	F/2	F/32	F/4
11	1	F/16	F/32	F/32	F/2	F/64	F/4

*1 RF[1:0] is defined in the Config Register in TX3920 Core.
*2 SLOWBUS is defined in the Power Control Register.

6.2.2 Clock Module Description

Figure 6-1 shows a block diagram of the Clock Module. The main TMPR3922 clock source is from an external crystal, connected to the TMPR3922 via the SYSCLKIN/SYSCLKOUT pins. The TMPR3922 contains an oscillator which is internally connected to these pins. The output of the oscillator feeds a phase-locked-loop (or multiplier) circuit which takes the lower rate oscillator output and generates the higher-rate clocks required by the TMPR3922. For example, if the external SYSCLK crystal rate is 9.216 MHz, the PLL multiplier circuit would perform a 16x rate multiplication to generate a high-speed clock rate of 147.456 MHz.

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

- DCLKOUT – one-half the rate of CORECLK (highest rate clock) ; used as the master clock for many of the TMPR3922 modules
- CORECLK – highest rate clock and 2X the rate of DCLKOUT; used as master clock for CPU core.
- FREECLK – same rate as DCLKOUT; used to generate the clocks for all other TMPR3922 modules and independent of the reduced frequency mode
- XHFREE – one-half the rate of FREECLK; used to generate the clocks for all other TMPR3922 modules and independent of the reduced frequency mode

The FREECLK and XHFREE signals are used to generate the clocks for all other TMPR3922 function modules.

The SIBMCLK pin can be configured as an output, for which the programmable rate is generated by dividing down from FREECLK. The SIBMCLK pin can also be configured as an input. In this mode, all SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to FREECLK. This mode allows optional decoupling of the SIB (as well as the CHI and UART) rates from the CPU core clock rate. The selected SIBMCLK source is then used as the SIB Master Clock for the SIB Module circuits, and is also used to generate the SIBSCLK and SIBSYNC externally driven TMPR3922 output signals.

The CHICLK pin can be configured as an output, for which the programmable rate is generated by dividing down from either XHREE or the SIB Master Clock. The CHICLK pin can also be configured as an input. In this mode, all CHI clocks are derived from an external peripheral source and the CHI Module will slave to this external clock. The selected CHICLK source is used as the CHI Master Clock for the CHI Module circuits, and is also used to generate the CHIFS externally driven TMPR3922 output signal.

The programmable UART Master Clock is generated by dividing down from either FREECLK or the SIB Master Clock, and is used as the master clock for the baud generator circuit within each UART Module.

The Clock Module also contains several fixed dividers for generating the master clocks for RTC timer and SPI circuits. These clocks are divided down from either FREECLK or XHFREE.

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

6.3 Clock Registers

6.3.1 Clock Control Register

OFFSET=$1C0:

Bit	Label	RESET	Read/Write
31-24	CHICLKDIV[7:0]	0	R/W
23	ENCLKTEST	0	R/W
22	CLKTESTSELSIB	0	R/W
21	CHIMCLKSEL	0	R/W
20	CHICLKDIR	0	R/W
19	ENCHIMCLK	0	R/W
18	SPICLKDIR	1	R/W
17	ENIRDACLK	0	R/W
16	ENSPICLK	0	R/W
15	ENTIMERCLK	0	R/W
14	ENFASTTIMERCLK	0	R/W
13	SIBMCLKDIR	0	R/W
12	ENC48MOUT	0	R/W
11	ENSIBMCLK	0	R/W
10-8	SIBMCLKDIV[2:0]	0	R/W
7	CSERSEL	1	R/W
6-4	CSERDIV[2:0]	0	R/W
3	ENCSERCLK	0	R/W
2	ENIRCLK	0	R/W
1	ENUARTACLK	0	R/W
0	ENUARTBCLK	0	R/W

CHICLKDIV[7:0]:

These bits select the start count value and the stop count value for the 8-bit programmable counter used to generate CHICLK, which is derived by dividing down from either SIBMCLK or CLK; CHICLK is the 1X or 2X bit clock used for the CHI Module and is generated by TMPR3922 whenever the CHI is in master mode. Since the MSB of the counter output is used for CHICLK, the start count and stop count values are chosen to provide (as close as possible) a 50% duty cycle CHICLK. The counter divide modulus is equal to (CHICLKDIV + 2). The equations used to compute these counter start and stop values are as follows:

start count value = 127 – ((CHICLKDIV + 1) >> 1)
stop count value = 128 + (CHICLKDIV >> 1)

For example, if CHICLKDIV = 7, then the start count value = 123 and the stop count value = 131, resulting in a divide-by-9 counter.

ENCLKTEST:

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

CLKTESTSELSIB:

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

CHIMCLKSEL:

Setting this bit to a logic "1" selects the external SIBMCLK source to be the CHI master clock. Clearing this bit to a logic "0" selects an internal clock source (of rate equal to XHFREE) to be the CHI master clock.

CHICLKDIR:

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

ENCHIMCLK:

This bit is used to enable or disable the CHICLK counter and CHICLK clock generation. Setting this bit to a logic "1" enables the CHICLK counter and CHICLK generation. Clearing this bit to a logic "0" disables the CHICLK counter and CHICLK generation, halting the clock to the CHI Module in order to reduce power consumption. The CHICLK counter is a programmable 8-bit divider.

SPICLKDIR:

This bit controls the direction of the SPICLK pin. Setting this bit to a logic "1" configures SPICLK to be an output (SPI master mode). Clearing this bit to a logic "0" configures SPICLK to be an input (SPI slave mode).

ENIRDACLK:

This bit is used to enable or disable the IrDA(FIR) master clock. Setting this bit to a logic "1" enables the IRDACLK. Clearing this bit to a logic "0" disables the IRDACLK, halting the clock to the IrDA(FIR) Module in order to reduce power consumption.

ENSPICLK:

This bit is used to enable or disable the SPICLK counter and SPICLK clock generation. Setting this bit to a logic "1" enables the SPICLK counter and SPICLK clock generation. Clearing this bit to a logic "0" disables the SPICLK counter and SPICLK clock generation, halting the clock to the SPI Module in order to reduce power consumption. The SPICLK counter is a fixed divide-by-5 of XHFREE.

ENTIMERCLK:

This bit is used to enable or disable the RTC periodic timer clock counter. Setting this bit to a logic "1" enables the timer clock counter. Clearing this bit to a logic "0" disables the timer clock counter. The timer clock counter is a fixed divide-by-32 of XHFREE.

ENFASTTIMERCLK:

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

SIBMCLKDIR:

This bit controls the direction of the SIBMCLK pin. Setting this bit to a logic "1" configures SIBMCLK to be an output. In this SIB master mode, the SIB clocks are slaved to FREECLK. Clearing this bit to a logic "0" configures SIBMCLK to be an input. In this SIB slave mode, the SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to FREECLK.

ENC48MOUT:

This bit is used to enable or disable the buffered output of the 48MHz clock.

ENSIBMCLK:

This bit is used to enable or disable the SIBMCLK counter and SIBMCLK clock generation. Setting this bit to a logic "1" enables the SIBMCLK counter and SIBMCLK generation. Clearing this bit to a logic "0" disables the SIBMCLK counter and SIBMCLK generation, halting the clock to the SIB Module in order to reduce power consumption. The SIBMCLK counter is a programmable 3-bit divider.

SIBMCLKDIV[2:0]:

These bits select the start count value and the stop count value for the 3-bit programmable counter used to generate SIBMCLK, which is derived by dividing down FREECLK; SIBMCLK is the master clock used for the SIB Module and is generated by TMPR3922 whenever the SIB is in SIB master mode. Since the MSB of the counter output is used for SIBMCLK, the start count and stop count values are chosen to provide (as close as possible) a 50% duty cycle SIBMCLK. The table used to compute these counter start and stop values are as follows:

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

CSERSEL:

Clearing this bit to a logic "0" selects the external SIBMCLK source to be the CSERCLK master clock. setting this bit to a logic "1" selects an internal clock source (of rate equal to FREECLK) to be the CSERCLK master clock.

CSERDIV[2:0]:

These bits select the start count value and the stop count value for the 4-bit programmable counter used to generate IRCLK, UARTACLK, and UARTBCLK; these clocks are derived by dividing down CSERCLK. Since the MSB of the counter output is used for these clocks, the start count and stop count values are chosen to provide (as close as possible) a 50% duty cycle for these clocks. The table used to compute these counter start and stop values are as follows:

CSERDIV	start value	stop value	divide-modulus
0	7	8	2
1	6	8	3
2	6	9	4
3	5	9	5
4	5	a	6
5	4	a	7
6	4	b	8
7	3	b	9

ENCSERCLK:

This bit is used to enable or disable the CSERCLK counter. Setting this bit to a logic "1" enables the CSERCLK counter. Clearing this bit to a logic "0" disables the CSERCLK counter. The CSERCLK counter is a programmable 4-bit divider.

ENIRCLK:

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

ENUARTACLK :

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

ENUARTBCLK:

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

SECTION 7 CHI Module

7.1 Overview

The CHI Module within the TMPR3922 contains holding registers, shift registers, DMA support, and other logic to support interfacing to external full-duplex serial TDM communication peripherals, including ISDN communication devices and PCM / TDM serial highways. The TMPR3922 implementation of the CHI Module is based on the Concentration Highway Interface standard specified by Intel and AT&T, which is intended to allow glueless interface to various TDM highways used by numerous commercial products. The TMPR3922 CHI Module can also be used to support an intra-system high-speed serial DMA channel within a PIC system.

The TMPR3922 CHI Module utilizes a 4-signal interface consisting of clock, sync, transmit serial data, and receive serial data. The data is organized as a TDM format, with up to 64 timeslots (nominally 8 bits each) per frame, with a nominal frame rate of 8 kHz (8 bits × 8 kHz = 64 kbps nominal data rate per channel). The number of timeslots and the data rate (up to 4.096 Mbps) and frame rate are programmable, providing flexibility for supporting various TDM communication peripherals. These timeslots are commonly used to carry voice, data, or control and status information.

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

7.1.1 Related Pins

CHIFS: INPUT / OUTPUT

This pin is the CHI frame synchronization signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3922 to be the master CHI sync source. As an input, this pin allows an external peripheral to be the master CHI sync source and the TMPR3922 CHI Module will slave to this external sync.

CHICLK: INPUT / OUTPUT

This pin is the CHI clock signal. This pin is available for use in one of two modes. As an output, this pin allows the TMPR3922 to be the master CHI clock source. As an input, this pin allows an external peripheral to be the master CHI clock source and the TMPR3922 CHI Module will slave to this external clock.

CHIDOUT: OUTPUT

This pin is the CHI serial data output signal.

CHIDIN: INPUT

This pin is the CHI serial data input signal.

7.2 Interface Requirements

7.2.1 Frame Structure and Serial Timing

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

The TMPR3922 CHI Module supports a master or slave mode for both the clock (CHICLK) and sync (CHIFS). For the master mode, the TMPR3922 contains programmable dividers for generating the clock and/or sync signal, synchronously dividing down from the main clock (XHFREE). For the slave mode, TMPR3922 accepts external clock and/or sync signals and utilizes "digital-PLL" type circuitry to stayed "locked" to the external source.

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

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

The TMPR3922 CHI Module allows for a programmable bit offset for both CHIDIN and CHIDOUT, which is related to the number of clock cycles between the start of timeslot 0 and CHIFS. This flexibility allows the TMPR3922 CHI Module to support a wide variety of interface clock and sync timing formats. The control bits for controlling the CHIDIN bit offset are CHIRXBOFF[3:0], while the control bits for controlling the CHIDOUT bit offset are CHITXBOFF[3:0].

Table 7-1 shows a summary matrix for the values of CERX and CETX for all possible settings of CHIRXBOFF and CHITXBOFF, respectively. These values are shown for various configurations of CHICLK mode (1x versus 2x), CHIFSEDGE, CHIRXEDGE, and CHITXEDGE. The CHIFSEDGE settings determine whether to use the rising edge (CHIFSEDGE = 1) or falling edge (CHIFSEDGE = 0) of CHICLK to sample CHIFS. The CHIRXEDGE settings determine whether to use the rising edge (CHIRXEDGE = 1) or falling edge (CHIRXEDGE = 0) of CHICLK to sample the CHIDIN input. The CHITXEDGE settings determine whether to use the rising edge (CHITXEDGE = 1) or falling edge (CHITXEDGE = 0) of CHICLK to push the CHIDOUT output. CERX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDIN is sampled. CETX is defined as the number of CHICLK clock edges (rising and falling) between the edge where CHIFS is sampled and the edge where CHIDOUT is pushed. The CHI frame structure and bit offsets are shown in Figure 7-1 for various clock and sync configurations.

Table 7-1a CERX Values for CHIRXBOFF Versus Clock and Edge Configurations

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

Table 7-1b CETX Values for CHITXBOFF Versus Clock and Edge Configurations

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

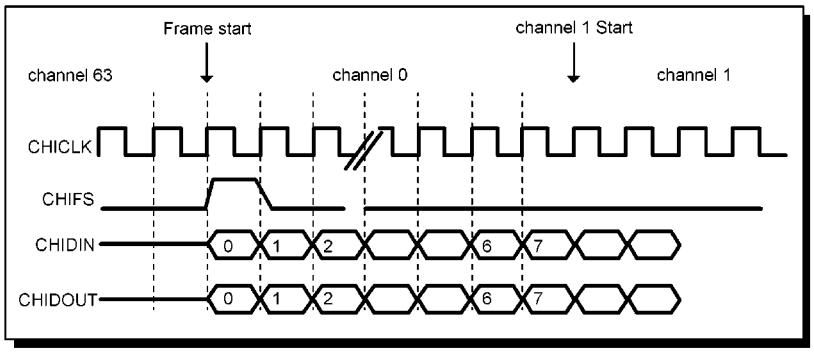


Figure 7-1a CHI Frame Structure Example

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

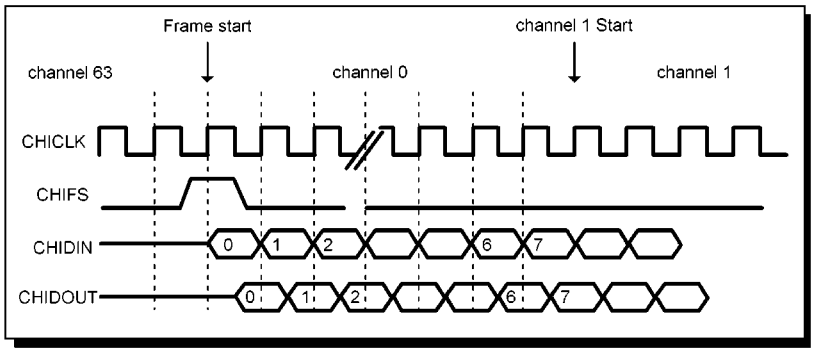


Figure 7-1b CHI Frame Structure Example

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

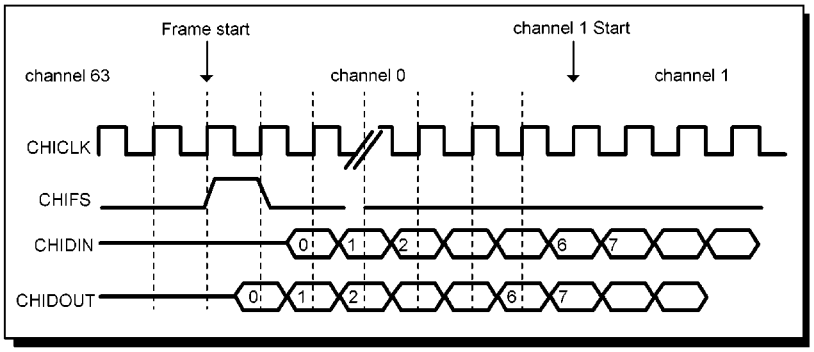


Figure 7-1c CHI Frame Structure Example

CHICLK 1X mode
CHIFS sampled on falling edge
CHIDIN sampled on rising edge; RXBOFF = 4; CERX = 3
CHIDOUT pushed on falling edge; TXBOFF = 2; CETX = 0

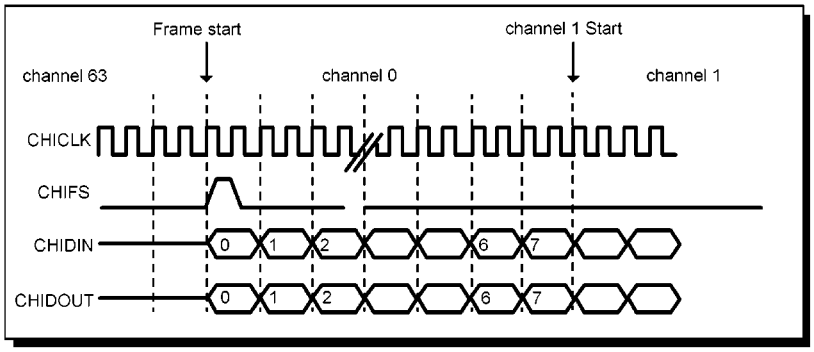


Figure 7-1d CHI Frame Structure Example

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

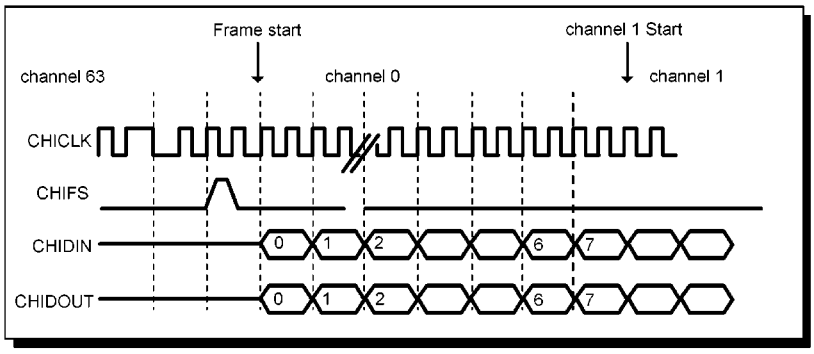


Figure 7-1e CHI Frame Structure Example

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

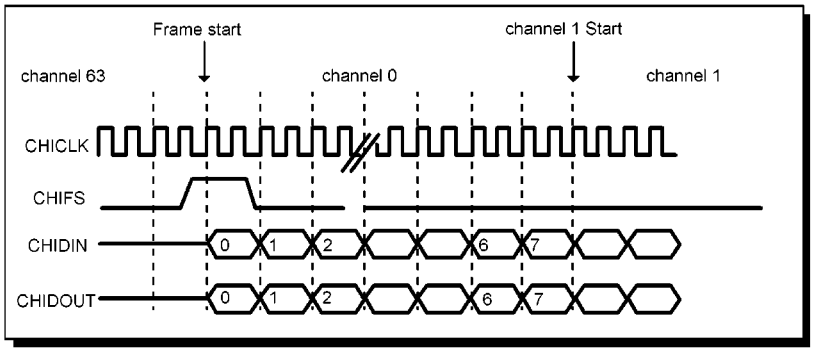


Figure 7-1f CHI Frame Structure Example

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

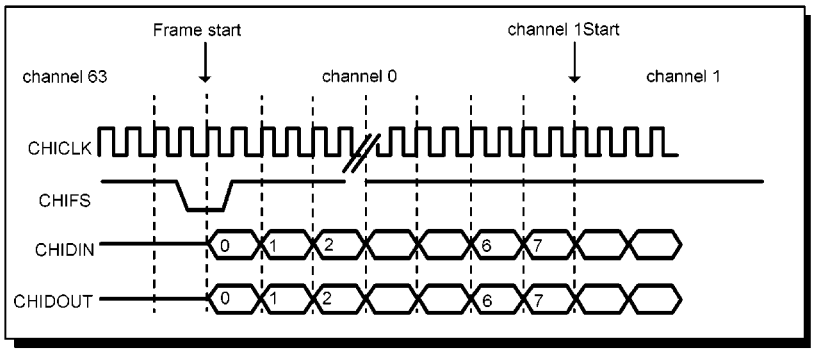


Figure 7-1g CHI Frame Structure Example

CHICLK 2X mode
CHIFS sampled on falling edge
CHIRXFSPOL = 1 (negative polarity)
CHIDIN sampled on rising edge; RXBOFF = 1; CERX = 3
CHIDOUT pushed on rising edge; TXBOFF = 1; CETX = 1

7.3 Implementation

7.3.1 Block Diagram

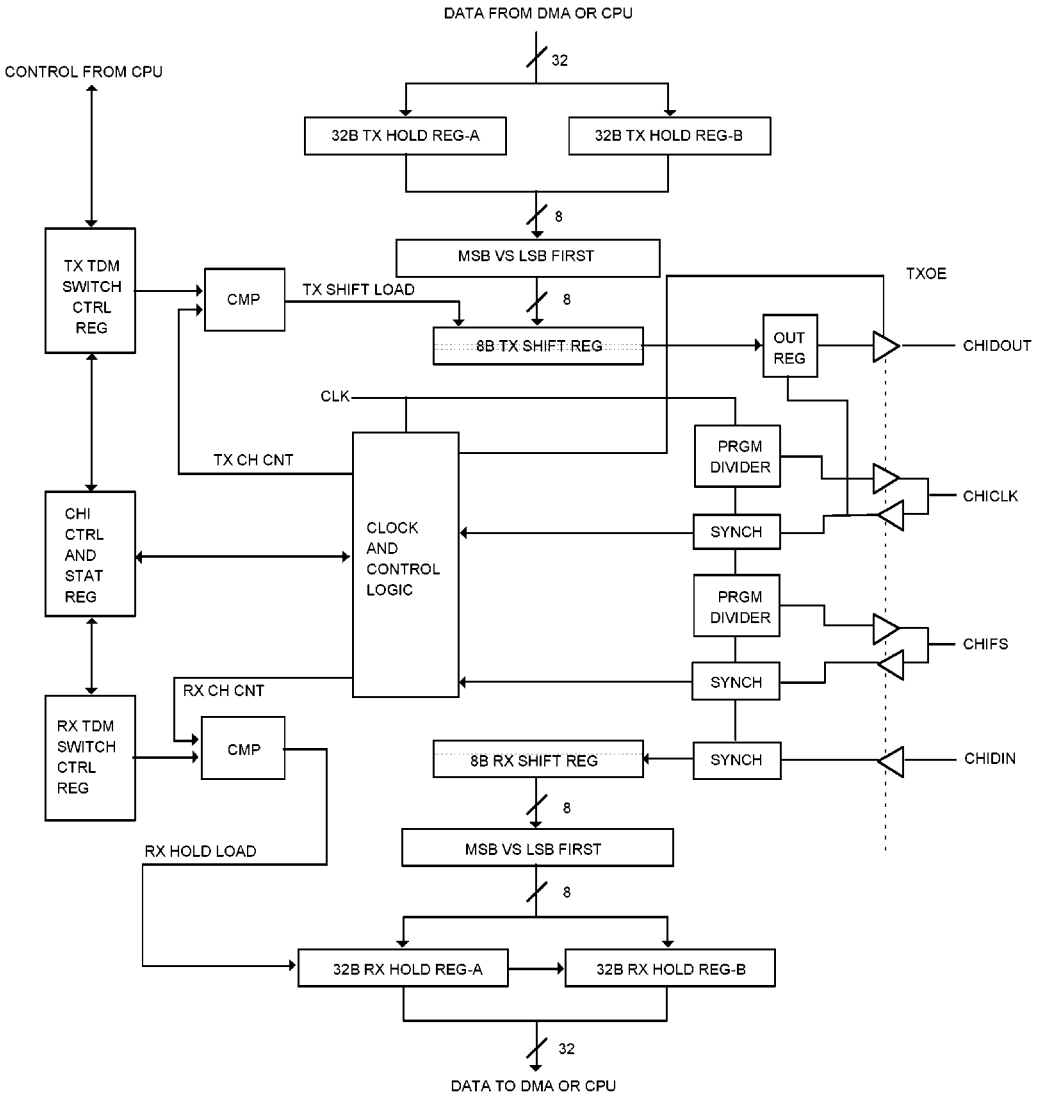


Figure 7-2 CHI Module Block Diagram

The CHI Module consists of holding registers (both transmit and receive), shift registers (both transmit and receive), DMA support, and other logic to support interfacing to various types of TDM highways. See Figure 7-2 for a block diagram of the CHI Module.

7.3.2 Transmitter

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

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

7.3.3 Receiver

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

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

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

7.3.4 Clock and Control Generation

The CHI Module contains several programmable counters which are used to generate the various CHI internal and external control signals and clocks. See Figure 7-3 for a block diagram of the CHI clock and control generation circuit. As mentioned previously, CHICLK can be configured as either an output (master mode) or input (slave mode). As an output, CHICLK is derived by dividing down from CLK. In this mode, all CHI clocks are then synchronously locked to the main TMPR3922 system clock. As an input, CHICLK is generated from an external clock source, which is asynchronous with respect to CLK. The TMPR3922 CHI Module utilizes a digital-PLL circuit to stayed "locked" to the external source, while still operating internally using CLK. CHIDIN and CHIDOUT are also synchronized between CLK and the externally-supplied CHICLK.

CHIFS can also be configured as either an output (master mode) or input (slave mode). As an output, CHIFS is derived by dividing down from CHICLK. For this mode, the CHIFS pulse width and polarity is also programmable. As an input, CHIFS is generated from an external sync source. The TMPR3922 CHI Module utilizes a digital-PLL circuit to stayed "locked" to the external sync source, while still operating internally using CLK.

The programmable receive and transmit sync delay counters shown in Figure 7-3 are used to implement the bit offset feature described earlier. The bit offset control bits determine the number of clock cycles between the start of timeslot 0 and CHIFS. The receive and transmit sync delay counters are independent from each other, such that the receive and transmit serial data streams can have different bit offsets.

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

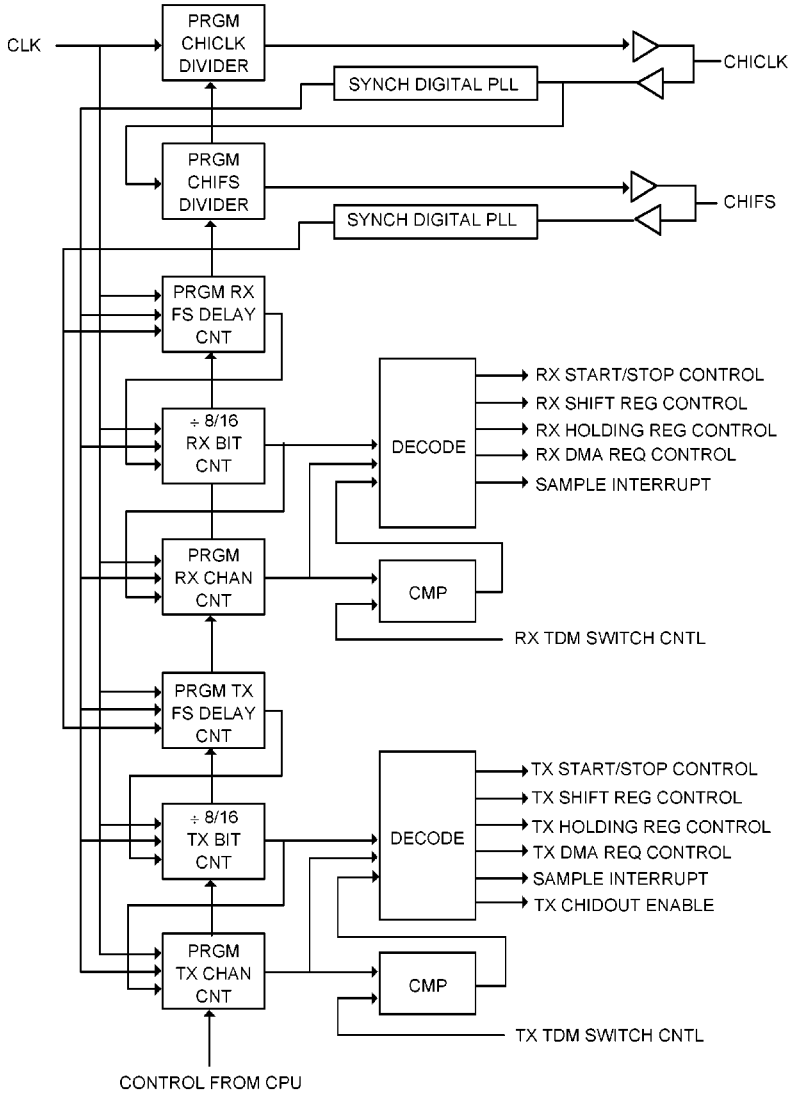


Figure 7-3 CHI Clock and Control Generation

7.3.5 DMA Address Generation

The CHI Module provides support for 2 independent DMA channels: receive and transmit. The circuit used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts is shown in Figure 7-4.

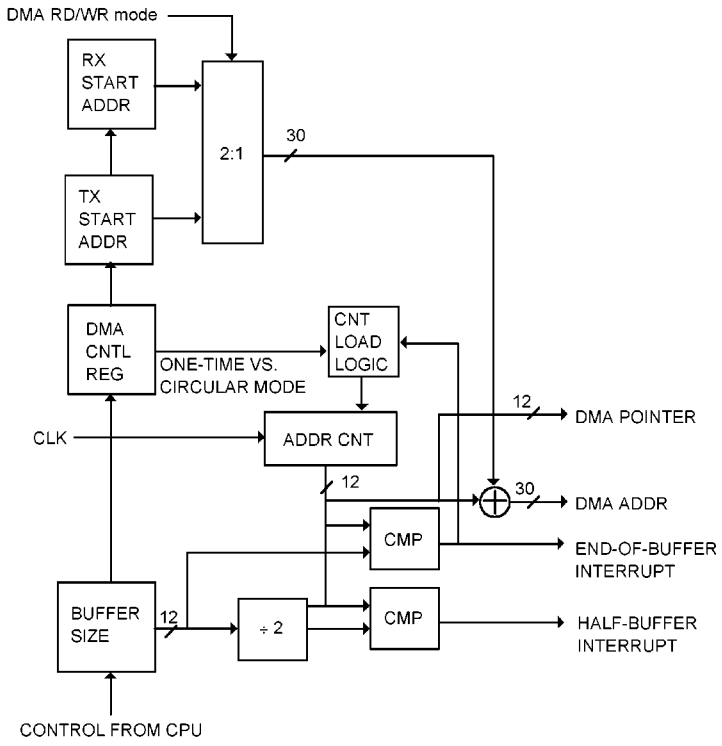


Figure 7-4 CHI DMA Adress Generation

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

The DMA buffers can be configured in a circular buffer mode or a one-time buffer mode. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TMPR3922's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

Because the CHI Module reads and writes a byte at a time between the shift registers and the longword holding registers, the software must pack and unpack these bytes to and from the longwords in memory in order to multiplex and demultiplex each channel for processing. Table 7-3 shows the format and organization of the CHI channels within memory for DMA mode. Consecutive byte samples for a given channel reside in memory every 8th byte.

Table 7-3 CHI DMA Memory Organization

(relative) memory address	bits 31:24	bits 23:16	bits 15:8	bits 7:0
0 × 0	buffA, ch3 sample 0	buffA, ch2 sample 0	buffA, ch1 sample 0	buffA, ch0 sample 0
0 × 4	buffB, ch3 sample 0	buffB, ch2 sample 0	buffB, ch1 sample 0	buffB, ch0 sample 0
0 × 8	buffA, ch3 sample 1	buffA, ch2 sample 1	buffA, ch1 sample 1	buffA, ch0 sample 1
0 × C	buffB, ch3 sample 1	buffB, ch2 sample 1	buffB, ch1 sample 1	buffB, ch0 sample 1
0 × 10	buffA, ch3 sample 2	buffA, ch2 sample 2	buffA, ch1 sample 2	buffA, ch0 sample 2
etc.				

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

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

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

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

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

7.3.6 Related Interrupts

CHI0_5INT:

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

CHI1_0INT:

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

CHIDMACNTINT:

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

CHIININTA:

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

CHIININTB:

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

CHIACTINT:

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

CHIERRINT:

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

7.4 CHI Registers

7.4.1 CHI Control Register

OFFSET=$1D8:

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

CHILOOP:

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

CHIENTEST:

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

CHIFSDIR:

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

CHIFSWIDTH[1:0]:

These bits are used to select pulse width for the CHIFS signal, relevant whenever the CHI Module is configured as master mode. The available CHIFS pulse widths are as follows:

CHIFSWIDTH	CHIFS pulse width
0	1 bit wide
1	2 bits wide
2	1 byte wide
3	(CHINCHAN ÷ 2) channels wide

CHINCHAN[4:0]:

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

CHITXBOFF[3:0]:

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

CHIRXBOFF[3:0]:

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

TXMSBFIRST:

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

RXMSBFIRST:

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

CHIRXFSPOL:

This bit selects between positive (active high) or negative (active low) polarity for the received CHIFS signal pulse, relevant whenever the CHI Module is configured as slave mode. Setting this bit to a logic "1" selects negative polarity for the CHIFS pulse. In this case, the falling edge of the CHIFS pulse is used to trigger the start of the CHI frame period. Clearing this bit to a logic "0" selects positive polarity for the CHIFS pulse. In this case, the rising edge of the CHIFS pulse is used to trigger the start of the CHI frame period.

CHITXFSPOL:

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

CHIRXEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive data CHIDIN. Setting this bit to a logic "1" selects rising edge. Clearing this bit to a logic "0" selects falling edge.

CHITXEDGE:

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

CHIFSEDGE:

This bit selects whether to use either the rising edge or falling edge of CHICLK to sample the receive frame sync CHIFS. Setting this bit to a logic "1" selects rising edge. Clearing this bit to a logic "0" selects falling edge.

CHITXFSEDGE:

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

CHICLK2XMODE:

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

CHIRXEN:

This bit is used to enable/disable CHI receive processing. Setting this bit to a logic "1" enables CHI receive processing. Clearing this bit to a logic "0" disables CHI receive processing, causing all received data to not be processed by the CHI module. This bit should not be set until after the CHI Module is setup, then ENCHI asserted.

CHITXEN:

This bit is used to enable/disable CHI transmit processing. Setting this bit to a logic "1" enables CHI transmit processing. Clearing this bit to a logic "0" disables CHI transmit processing, causing the CHI serial transmitted data to be tri-stated. This bit should not be set until after the CHI Module is setup, then ENCHI asserted.

ENCHI:

This bit is used to enable/disable the CHI module. Setting this bit to a logic "1" enables the CHI module. Clearing this bit to a logic "0" disables the CHI Module and keeps the module in a reset state.

7.4.2 CHI Pointer Enable Register

OFFSET=$1DC:

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

CHITXPTRB3EN:

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

CHITXPTRB2EN:

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

CHITXPTRB1EN:

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

CHITXPTRB0EN:

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

CHITXPTRA3EN:

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

CHITXPTRA2EN:

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

CHITXPTRA1EN:

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

CHITXPTRA0EN:

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

CHIRXPTRB3EN:

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

CHIRXPTRB2EN:

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

CHIRXPTRB1EN:

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

CHIRXPTRB0EN:

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

CHIRXPTRA3EN:

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

CHIRXPTRA2EN:

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

CHIRXPTRA1EN:

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

CHIRXPTRA0EN:

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

7.4.3 CHI Receive Pointer A Register

OFFSET=$1E0: write-only

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

CHIRXPTRA3[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 3 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHIRXPTRA2[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 2 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHIRXPTRA1[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 1 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHIRXPTRA0[4:0]: write-only

These bits represent the TDM switch pointer which defines the receive channel timeslot for byte 0 of the CHI receive holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

7.4.4 CHI Receive Pointer B Register

OFFSET=$1E4: write-only

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

CHIRXPTRB3[4:0]: write-only

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

CHIRXPTRB2[4:0]: write-only

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

CHIRXPTRB1[4:0]: write-only

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

CHIRXPTRB0[4:0]: write-only

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

7.4.5 CHI Transmit Pointer A Register

OFFSET=$1E8: write-only

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

CHITXPTRA3[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 3 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHITXPTRA2[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 2 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHITXPTRA1[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 1 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

CHITXPTRA0[4:0]: write-only

These bits represent the TDM switch pointer which defines the transmit channel timeslot for byte 0 of the CHI transmit holding register A; register A handles all timeslots from channel 0 to channel (CHINCHAN ÷ 2) –1.

7.4.7 CHI Size Register

OFFSET=$1F0:

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

CHIDMAPTR[13:2]: read-only

These bits provide the status of the CHI DMA counter.

CHIBUFF1TIME:

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

CHIDMALOOP:

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

CHISIZE(13:2): write-only

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

ENDMARXCHI:

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

ENDMATXCHI:

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

7.4.8 CHI RX Start Register

OFFSET=$1F4: write-only

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

CHIRXSTART[31:2]: write-only

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

7.4.9 CHI TX Start Register

OFFSET=$1F8: write-only

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

CHITXSTART[31:2]: write-only

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

7.4.10 CHI TX Holding Register

OFFSET=$1FC: write-only

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

CHITXHOLD[31:0] : write-only

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

7.4.11 CHI RX Holding Register

OFFSET=$1FC: read-only

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

CHIRXHOLD[31:0]: read-only

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

SECTION 8 Interrupt Module

8.1 Overview

The Interrupt Module within the TMPR3922 contains logic for reading, enabling, and clearing all of the interrupt sources. These interrupts are either generated from internal TMPR3922 modules or from edge transitions on external signal pins. All of these latter interrupt types generate a separate positive and negative edge interrupt.

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

8.2 Implementation

8.2.1 Block Diagram

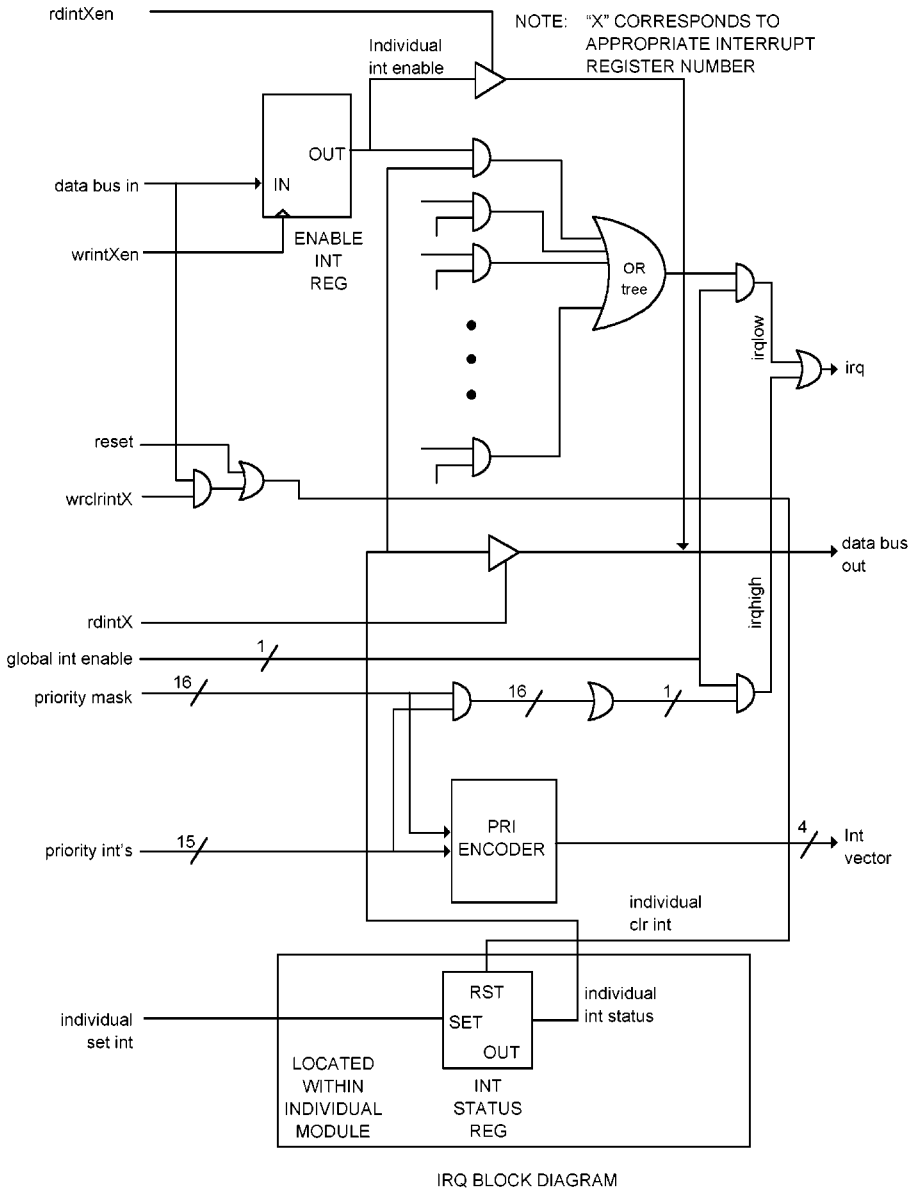


Figure 8-1 Interrupt Circuit Block Diagram

8.2.2 Interrupt Logic Description

Figure 8-1 shows a block diagram of the interrupt circuit for a single interrupt source. Every other interrupt source contains an identical circuit.

The overall Interrupt Module contains 7 main sets of registers, with a 6th register allocated for setting the global interrupt enable and the 16-bit Priority Mask, and for reading the Priority Interrupt Vector and IRQHIGH and IRQLOW status bits. Each bit within each of the 7 main sets then consists of 3 registers: Status, Enable, and Clear.

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

A given Interrupt Status Register is set by the corresponding individual interrupt event. For example, this event could be a positive (or negative) edge transition on an external TMPR3922 pin. Another example might be an event triggered by an internal TMPR3922 module, such as a particular DMA buffer reaching the end-of-buffer point.

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

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

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

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

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

The IRQHIGH signal is connected to interrupt bit 4 on the TX3920 Processor Core and the IRQLOW signal is connected to interrupt bit 2. The IRQHIGH signal is also connected to CPU Co-Processor Condition bit 3.

8.3 Interrupt Registers

8.3.1 Interrupt Status 1 Register

OFFSET=$100: read-only

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

CHI0_5INT:

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

CHI1_0INT:

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

CHIDMACNTINT:

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

CHIININTA:

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

CHIININTB:

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

CHIACTINT:

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

CHIERRINT:

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

SND0_5INT:

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

SND1_0INT:

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

TEL0_5INT:

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

TEL1_0INT:

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

SNDDMACNTINT:

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

TELDMACNTINT:

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

LSNDCLIPINT:

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

RSNDCLIPINT:

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

VALSNDPOSINT:

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

VALSNDNEGINT:

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

VALTELPOSINT:

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

VALTELNEGINT:

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

SNDININT:

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

TELININT:

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

SIBSF0INT:

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

SIBSF1INT:

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

SIBIRQPOSINT:

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

SIBIRQNEGINT:

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

8.3.2 Interrupt Status 2 Register

OFFSET=$104: read-only

Bit	Label	RESET	Read/Write
31	UARTARXINT	—	R
30	UARTARXOVERRUNINT	—	R
29	UARTAFRAMEERRINT	—	R
28	UARTABREAKINT	—	R
27	UARTAPARITYERRINT	—	R
26	UARTATXINT	—	R
25	UARTATXOVERRUNINT	—	R
24	UARTAEMPTYINT	—	R
23	UARTADMAFULLINT	—	R
22	UARTADMAHALFINT	—	R
21	UARTBRXINT	—	R
20	UARTBRXOVERRUNINT	—	R
19	UARTBFRAMEERRINT	—	R
18	UARTBBREAKINT	—	R
17	UARTBPARITYERRINT	—	R
16	UARTBTXINT	—	R
15	UARTBTXOVERRUNINT	—	R
14	UARTBEMPTYINT	—	R
13	UARTBDMAFULLINT	—	R
12	UARTBDMAHALFINT	—	R
11-0	Reserved	—	R

UARTARXINT:

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

UARTARXOVERRUNINT:

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

UARTAFRAMEERRINT:

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

UARTABREAKINT:

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

UARTAPARITYERRINT:

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

UARTATXINT:

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

UARTATXOVERRUNINT:

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

UARTAEMPTYINT:

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

UARTADMAFULLINT:

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

UARTADMAHALFINT:

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

UARTBRXINT:

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

UARTBRXOVERRUNINT:

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

UARTBFRAMEERRINT:

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

UARTBBREAKINT:

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

UARTBPARITYERRINT:

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

UARTBTXINT:

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

UARTBTXOVERRUNINT:

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

UARTBEMPTYINT:

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

UARTBDMAFULLINT:

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

UARTBDMAHALFINT:

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

8.3.3 Interrupt Status 3 Register

OFFSET=$108 : read-only

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

MFIOPOSINT[31:0]:

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

8.3.4 Interrupt Status 4 Register

OFFSET=$10C: read-only

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

MFIONEGINT[31:0] :

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

8.3.5 Interrupt Status 5 Register

OFFSET=$110: read-only

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

RTCINT:

This interrupt is set whenever all 43 bits of the RTC counter reach a value of "$7FFFFFFFFF" to alert the software that the counter is "rolling over".

ALARMINT:

This interrupt is set whenever the RTC counter reaches a count that is equal to the value of the ALARM[42:0] bits set in the Alarm Register.

PERINT:

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

STPTIMERINT:

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

STPTIMERINT:

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

POSPWRINT:

This interrupt is set when the PWRINT pin transitions from a logic "0" to a logic "1".

NEGPWRINT:

This interrupt is set when the PWRINT pin transitions from a logic "1" to a logic "0".

POSPWROKINT:

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

NEGPWROKINT:

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

POSONBUTNINT:

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

NEGONBUTNINT:

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

SPIBUFAVAILINT:

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

SPIERRINT:

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

SPIRCVINT:

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

SPIEMPTYINT:

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

IRCONSMINT:

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

CARSTINT:

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

POSCARINT:

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

NEGCARINT:

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

8.3.6 Interrupt Status 6 Register

OFFSET=$114: read-only

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

IRQHIGH:

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

IRQLOW:

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

INTVECT[3:0]:

These 4 status bits are a vector pointing to the highest priority pending interrupt from the set of 13 possible high priority interrupt events. The set of high priority interrupts is listed below, where level 15 is the highest priority and level 1 is the lowest priority. Level 0 corresponds to the standard interrupt handler.

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

8.3.7 Interrupt Status 7 Register

OFFSET=$130: read-only

Bit	Label	RESET	Read/Write
31-21	Reserved		
20	IRTXCINT	—	R
19	IRRXCINT	—	R
18	IRTXEINT	—	R
17	IRRXEINT	—	R
16	IRSIRPXINT	—	R
15-0	Reserved		

IRTXCINT:

Issues an interrupt if the IrDA Packet Transmit is completed when the REQEST_TO_CLEAR_ENTX bit is set..

IRRXCINT:

Issues an interrupt whenever one Packet data is received correctly.

IRTXEINT:

Issues an interrupt if the IrDA detects the underrun in the Transmit FIFO.

IRRXEINT:

Issues an interrupt if the IrDA detects the underrun in the Receive FIFO.

IRSIRPXINT:

Issues an interrupt if the IrDA module detects Serial Interaction Pulse(SIP).

8.3.8 Interrupt Status 8 Register

OFFSET=$138 : read-only

Bit	Label	RESET	Read/Write
31-16	IOPOSINT[15:0]	—	R
15-0	IONEGINT[15:0]	—	R

IOPOSINT[15:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic "0" to a logic "1". There are a total of 16 general purpose I/O pins (15 thru 0), each of which corresponds to the respective bit within this Status Register. Each general purpose I/O pin can independently trigger an interrupt.

IONEGINT[15:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic "1" to a logic "0". There are a total of 16 general purpose I/O pins (15 thru 0), each of which corresponds to the respective bit within this Status Register. Each general purpose I/O pin can independently trigger an interrupt.

8.3.9 Clear Interrupt 1 Register

OFFSET = $100: write-only

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

8.3.10 Clear Interrupt 2 Register

OFFSET = $104: write-only

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

8.3.11 Clear Interrupt 3 Register

OFFSET = $108: write-only

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

8.3.12 Clear Interrupt 4 Register

OFFSET = $10C: write-only

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

8.3.13 Clear Interrupt 5 Register

OFFSET = $110: write-only

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

8.3.14 Clear Interrupt 7 Register

OFFSET = $130: write-only

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

8.3.15 Clear Interrupt 8 Register

OFFSET = $138: write-only

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

8.3.16 Enable Interrupt 1 Register

OFFSET = $118: read/write

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

8.3.17 Enable Interrupt 2 Register

OFFSET = $11C: read/write

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

8.3.18 Enable Interrupt 3 Register

OFFSET = $120: read/write

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

8.3.19 Enable Interrupt 4 Register

OFFSET = $124: read/write

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

8.3.20 Enable Interrupt 5 Register

OFFSET = $128: read/write

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

8.3.21 Enable Interrupt 6 Register

OFFSET = $12C: read/write

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

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

GLOBALEN:

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

IRQPRITEST:

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

IRQTEST:

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

PRIORITYMASK[15:0]:

These bits are an enable mask for the 13 possible high priority interrupts, where bit 15 is the highest priority. A logic "1" written to a specific bit location will enable the corresponding high priority interrupt.

8.3.22 Enable Interrupt 7 Register

OFFSET = $134: read/write

The Enable Interrupt 7 Register bit locations are mapped on a one to one basis with the Interrupt Status 7 Register. A logic "1" written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

8.3.23 Enable Interrupt 8 Register

OFFSET = $13C: read/write

The Enable Interrupt 8 Register bit locations are mapped on a one to one basis with the Interrupt Status 8 Register. A logic "1" written to a specific bit location will enable the corresponding interrupt in the Status Register. This register is not cleared upon reset; however, the GLOBALEN global interrupt enable bit is cleared upon reset, therefore disabling all interrupts.

SECTION 9 IO Module

9.1 Overview

The IO Module within the TMPR3922 contains support for reading and writing the 16 bi-directional general purpose IO pins and the 32 bi-directional multi-function IO pins.

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

Of the 141 signal pins found on the TMPR3922, 32 of them are multi-function and can be independently programmed either as IO ports or for an alternate standard / normal function. This allows the TMPR3922 to support a flexible and wide range of system applications and configurations. As an IO port, any of these pins can be programmed as an input or output port, with the capability of generating a separate positive and negative edge interrupt.

9.1.1 Related Pins

IO[15:0]: INPUT/OUTPUT

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

MFIO[31:0]: INPUT/OUTPUT

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

9.2 Implementation

9.2.1 Block Diagram

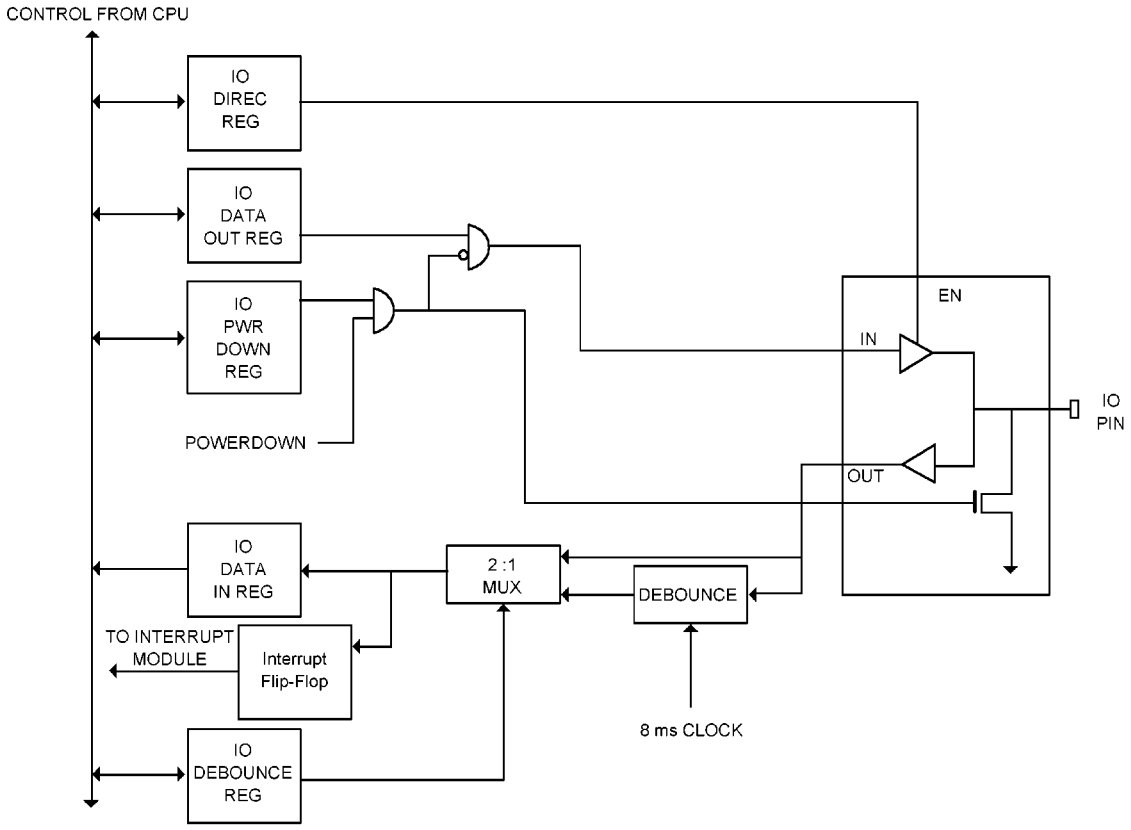


Figure 9-1 General Purpose IO Port Block Diagram

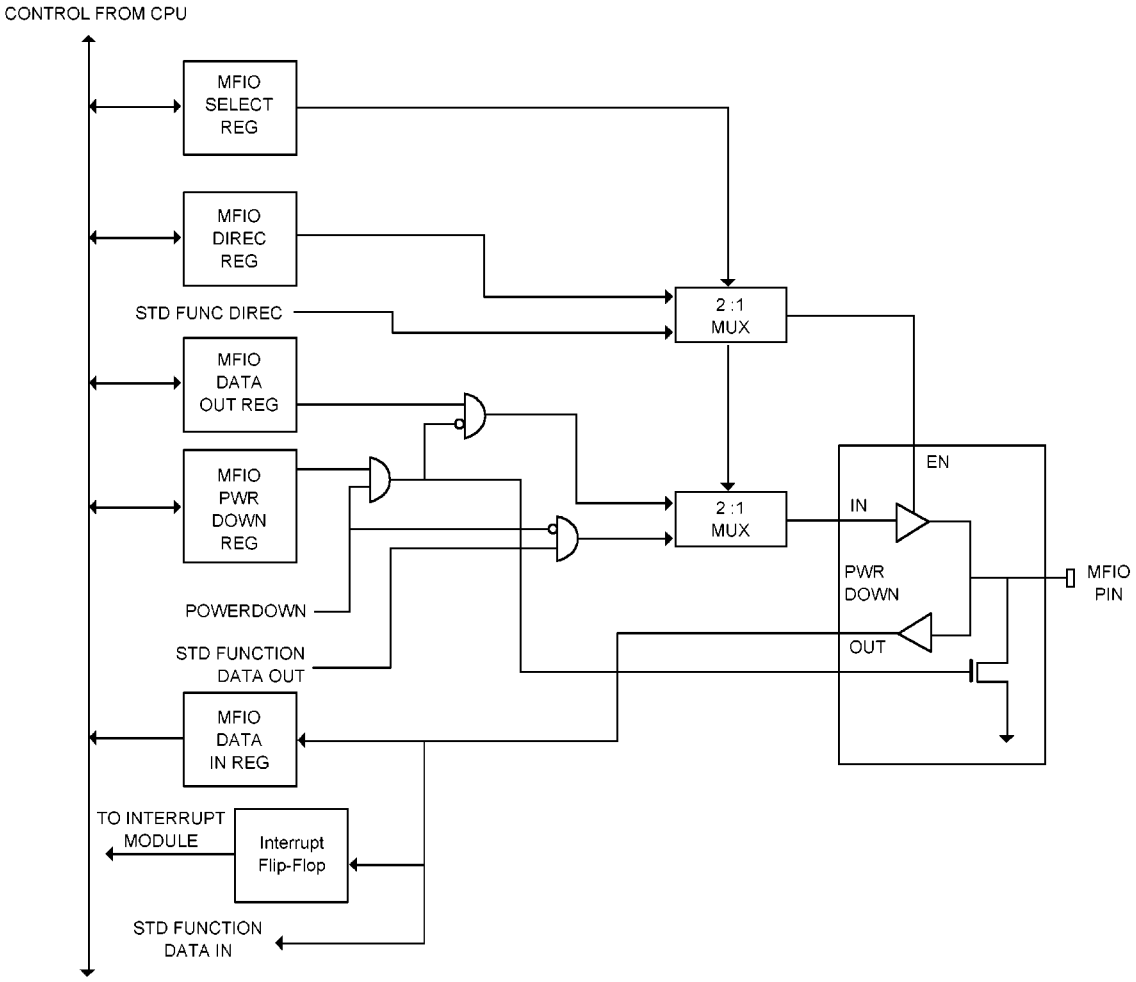


Figure 9-2 Multi-Function IO Port Block Diagram

9.2.2 General Purpose IO Ports

Each of the 16 general purpose IO ports can be independently programmed as an input or output port. Each port can generate a separate positive and negative edge interrupt. Figure 9-1 shows a block diagram of a general purpose IO port.

Each port consists of a bi-directional buffer connected to the appropriate TMPR3922 pin. For the input direction, the output signal from the input buffer is routed directly to a debounce circuit. This circuit performs a 16 to 24 ms debounce of the input signal. The debounce select control signal from the IO Debounce Select Register is used to select between the debouncer output and the direct signal from the port input buffer, which bypasses the debouncer. This selected signal is then routed to the IO Data Input Register and to the Interrupt Flip-Flop. Reading a specific bit location within the IO Data Input Register returns the logic state of the respective general purpose IO pin (either direct or debounced), regardless of whether that pin is configured as an output or input. If the pin is configured as an input, the value read is the logic state of the pin as driven by an external source. If the pin is configured as an output, the value read is the logic state of the pin as driven by the TMPR3922.

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

The IO Power-Down Register provides independent control bits for controlling the power-down state for each of the 12 general purpose IO ports. If a particular IO port is configured to power down, then the output will be gated low and a Pull-Down resistor will be enabled whenever the device powers down. Device power down occurs when either VCC3 or VCCON become de-asserted. The Pull-Down will prevent the input from floating if the general purpose IO port is configured as an input.

9.2.3 Multi-function IO Ports

Each of the 32 multi-function IO ports can be independently programmed as an input or output port or can be programmed for an alternate standard / normal function. This allows the TMPR3922 to support a flexible and wide range of system applications and configurations. As an IO port, any of these pins can generate a separate positive and negative edge interrupt. Figure 9-2 shows a block diagram of a multi-function IO port.

Each port consists of a bi-directional buffer connected to the appropriate TMPR3922 pin. For the input direction, the output signal from the input buffer is routed directly to the MFIO Data Input Register, to the Interrupt Flip-Flop, and to the appropriate module corresponding to the standard / normal function of the particular port, if this standard use is as an input signal. Reading a specific bit location within the MFIO Data Input Register returns the logic state of the respective multi-function IO pin, regardless of whether that pin is configured as an output or input. If the pin is configured as an input, the value read is the logic state of the pin as driven by an external source. If the pin is configured as an output, the value read is the logic state of the pin as driven by the TMPR3922.

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

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

As mentioned previously, each of the multi-function IO ports can be independently programmed for routing of signals for a standard / normal function to or from any given multi-function IO port. Table 9-1 lists each of these standard functions for each multi-function IO port. Note that depending on the reset state for the respective MFIO Select Register bits, the pin function is configured either as a multi-function port or as a standard function port.

Table 9-1 Multi-Function IO Ports Versus Standard Functions

TMPR3922 pin	standard function (I = input, O = output)	multi-function IO port	multi-function select (reset state)	Power Down Control Powerdown = / (vccon & vcc3) (reset state)
CHIFS	CHIFS (I/O)	MIO[31]	MIOSEL[31] (1)	POWERDOWN & MIOPD[31] (1)
CHICLK	CHICLK (I/O)	MIO[30]	MIOSEL[30] (1)	POWERDOWN & MIOPD[30] (1)
CHIDOUT	CHIDOUT (O)	MIO[29]	MIOSEL[29] (1)	POWERDOWN & MIOPD[29] (1)
CHIDIN	CHIDIN (I)	MIO[28]	MIOSEL[28] (1)	POWERDOWN & MIOPD[28] (1)
DREQ*	DREQ* (I)	MIO[27]	MIOSEL[27] (0)	POWERDOWN & MIOPD[27] (1)
DGRNT*	DGRNT* (O)	MIO[26]	MIOSEL[26] (0)	POWERDOWN & MIOPD[26] (0)
BC32K	BC32K (O)	MIO[25]	MIOSEL[25] (1)	POWERDOWN & MIOPD[25] (1)
TXD	TXD (O)	MIO[24]	MIOSEL[24] (0)	POWERDOWN & MIOPD[24] (0)
RXD	RXD (I)	MIO[23]	MIOSEL[23] (0)	POWERDOWN & MIOPD[23] (1)
CS1*	CS1* (O)	MIO[22]	MIOSEL[22] (0)	POWERDOWN & MIOPD[22] (1)
CS2*	CS2* (O)	MIO[21]	MIOSEL[21] (0)	POWERDOWN & MIOPD[21] (1)
CS3*	CS3* (O)	MIO[20]	MIOSEL[20] (0)	POWERDOWN & MIOPD[20] (1)
MCS0*	MCS0* (O)	MIO[19]	MIOSEL[19] (0)	POWERDOWN & MIOPD[19] (0)
MCS1*	MCS1* (O)	MIO[18]	MIOSEL[18] (0)	POWERDOWN & MIOPD[18] (0)
RXPWR	RXPWR (O)	MIO[17]	MIOSEL[17] (1)	POWERDOWN & MIOPD[17] (0)
IROUT	IROUT (O)	MIO[16]	MIOSEL[16] (1)	POWERDOWN & MIOPD[16] (0)
SPICLK	SPICLK (I/O)	MIO[15]	MIOSEL[15] (0)	POWERDOWN & MIOPD[15] (0)
SPIOUT	SPIOUT (O)	MIO[14]	MIOSEL[14] (0)	POWERDOWN & MIOPD[14] (0)
SPIIN	SPIIN (I)	MIO[13]	MIOSEL[13] (0)	POWERDOWN & MIOPD[13] (1)
SIBMCLK	SIBMCLK (I/O)	MIO[12]	MIOSEL[12] (0)	POWERDOWN & MIOPD[12] (1)
CARDREG*	CARDREG* (O)	MIO[11]	MIOSEL[11] (1)	POWERDOWN & MIOPD[11] (1)
CARDIOWR*	CARDIOWR* (O)	MIO[10]	MIOSEL[10] (1)	POWERDOWN & MIOPD[10] (1)
CARDIORD*	CARDIORD* (O)	MIO[9]	MIOSEL[9] (1)	POWERDOWN & MIOPD[9] (1)
CARD1CSL*	CARD1CSL* (O)	MIO[8]	MIOSEL[8] (1)	POWERDOWN & MIOPD[8] (1)
CARD1CSH*	CARD1CSH* (O)	MIO[7]	MIOSEL[7] (1)	POWERDOWN & MIOPD[7] (1)
CARD2CSL*	CARD2CSL* (O)	MIO[6]	MIOSEL[6] (1)	POWERDOWN & MIOPD[6] (1)
CARD2CSH*	CARD2CSH* (O)	MIO[5]	MIOSEL[5] (1)	POWERDOWN & MIOPD[5] (1)
CARD1WAIT*	CARD1WAIT* (I)	MIO[4]	MIOSEL[4] (1)	POWERDOWN & MIOPD[4] (1)
CARD2WAIT*	CARD2WAIT* (I)	MIO[3]	MIOSEL[3] (1)	POWERDOWN & MIOPD[3] (1)
CARDDIR*	CARDDIR* (O)	MIO[2]	MIOSEL[2] (1)	POWERDOWN & MIOPD[2] (1)
MCS1WAIT*	MCS1WAIT* (I)	MIO[1]	MIOSEL[1] (0)	POWERDOWN & MIOPD[1] (0)
MCS0WAIT*	MCS0WAIT* (I)	MIO[0]	MIOSEL[0] (0)	POWERDOWN & MIOPD[0] (0)

9.2.4 Related Interrupts

MFIOPOSINT[31:0]:

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

MFIONEGINT[31:0]:

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

IOPOSINT[15:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic "0" to a logic "1". There are a total of 16 general purpose I/O pins (15 thru 0), each of which corresponds to the respective bit within the Interrupt Status 8 Register. Each general purpose I/O pin can independently trigger an interrupt.

IONEGINT[15:0]:

Issues an interrupt whenever any of the general purpose I/O pin(s) transition from a logic "1" to a logic "0". There are a total of 16 general purpose I/O pins (15 thru 0), each of which corresponds to the respective bit within the Interrupt Status 8 Register. Each general purpose I/O pin can independently trigger an interrupt.

9.3 IO Registers

9.3.1 IO Control Register

OFFSET = $180:

Bit	Label	RESET	Read/Write
31-16	IODEBSEL[15:0]	X	R/W
15-0	IODIREC[15:0]	0	R/W

IODEBSEL[15:0]:

These bits select the debounce mode for the 16 general purpose IO pins. Setting a specific bit location to a logic "1" causes the respective general purpose IO port signal to be filtered by a debounce circuit for the input signal direction. Clearing a specific bit location to a logic "0" causes the respective general purpose IO port signal to bypass the input debounce circuit.

IODIREC[15:0]:

These bits control the direction of the 16 general purpose IO pins. Setting a specific bit location to a logic "1" configures the respective general purpose IO pin to be an output. Clearing a specific bit location to a logic "0" configures the respective general purpose IO pin to be an input.

9.3.2 IO Data In/Out Register

OFFSET = $19C:

Bit	Label	RESET	Read/Write
31-16	IODOUT[15:0]	X	R/W
15-0	IODIN[15:0]	—	R

IODOUT[15:0]:

These bits correspond to the data output values for the 16 general purpose IO pins. Setting or clearing a specific bit location controls the logic state of the respective general purpose IO pin, if that pin is configured as an output.

IODIN[15:0]: read-only

These bits correspond to the data input values for the 16 general purpose IO pins. Reading a specific bit location returns the logic state of the respective general purpose IO pin, regardless of whether that pin is configured as an output or input.

9.3.3 MFIO Data Output Register

OFFSET = $184:

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

MFIODOUT[31:0]:

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

9.3.4 MFIO Direction Register

OFFSET = $188:

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

MFIODIREC[31:0]:

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

9.3.5 MFIO Data Input Register

OFFSET = $18C: read-only

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

MFIODIN[31:0]: read-only

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

9.3.6 MFIO Select Register

OFFSET = $190:

Bit	Label	RESET	Read/Write
31-0	MFIOSEL[31:0]	$F2030FFC	R/W

MFIOSEL[31:0]:

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

9.3.7 IO Power-Down Register

OFFSET = $194:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	IOPD[15:0]	$FFF	R/W

IOPD[15:0]:

These bits control the power-down state for the 16 general purpose IO pins. Setting a specific bit location to a logic "1" configures the respective general purpose IO pin to power down when the TMPR3922 powers down. Clearing a specific bit location to a logic "0" configures the respective general purpose IO pin to be in an active state at all times.

9.3.8 MFIO Power-Down Register

OFFSET = $198:

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

MFIOPD[31:0]:

These bits control the power-down state for the 32 multi-function IO ports. Setting a specific bit location to a logic "1" configures the respective multi-function IO port to power down when the TMPR3922 powers down. Clearing a specific bit location to a logic "0" configures the respective multi-function IO port to be in an active state at all times.

SECTION 10 IR Module

10.1 Overview

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

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

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

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

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

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

10.1.1 Related Pins

IROUT: OUTPUT

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

IRIN: INPUT

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

RXPWR: OUTPUT

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

CARDET: INPUT

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

10.2 Consumer IR

10.2.1 Requirements

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

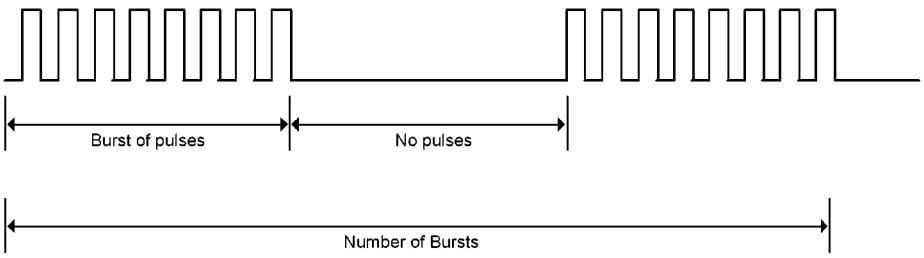


Figure 10-1 Consumer IR Pulses

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

10.2.2 Implementation

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

10.2.3 Block Diagram

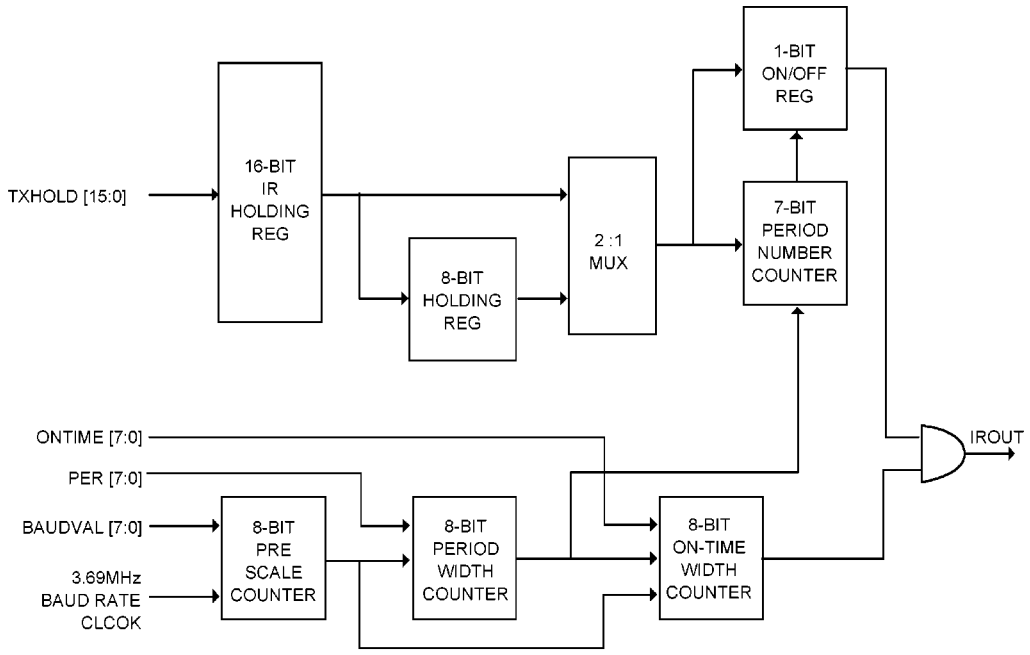


Figure 10-2 Consumer IR Block Diagram

10.2.4 Functional Description

The Consumer IR logic remains idle until the CPU activates the IR logic by writing to the IR Control 1 Register to enable the Consumer mode (ENCONSM = "1"). Prior to enabling the Consumer mode, the CPU must configure the Baud Value (BAUDVAL), Period Value (PER), and OnTime Value (ONTIME) fields located in the IR Control 2 Register. BAUDVAL[8:0] is used to pre-scale the 9.216 MHz baud rate clock since typical consumer IR data rates are in the kHz range instead of the MHz range. PER[7:0] is used to define the period of the pulses for a burst, representing the number of pre-scaled counts per period. ONTIME[7:0] defines the amount of time that the pulse is on for the given period, representing the number of pre-scaled counts for which the pulse is on. ONTIME should always be less than PER or the IROUT signal will never transition low.

Transmit data is written by the CPU into a 16-bit holding register. When the 7-bit Period Number Counter (down-counter) reaches zero it will load the lower 7 bits of the holding register's upper byte into the Period Number Counter and the upper bit into the 1-bit ON/OFF Register.
If the upper bit is set to a "1" then the IROUT signal will be a "1" until the OnTime Width Counter reaches it's terminal count. The OnTime Width Counter freezes once it reaches it's terminal count in order to prevent the signal from transitioning until the counter is loaded again. The IROUT signal will continue to transition according to the Period Value and OnTime Value until the 7-bit Period Number Counter reaches zero. At this time the lower 7 bits of the holding register's lower byte is loaded into the Period Number Counter and the upper bit of the lower byte is loaded into the 1-bit ON/OFF Register. If the upper bit is set to a "0" this time, then the IROUT signal will remain low for as many period counts as are defined by the value loaded into the 7-bit Period Number Counter. Thus the CPU controls the pulse generation by defining a number of pulses to burst (upper bit set to "1", lower 7-bit data defines the number of pulses) or an amount of time when there are no pulses (upper bit set to "0", lower 7-bit data defines the number of pulse times when IROUT is "0") on a byte by byte basis. The OnTime Value and Period Value timing relationships are shown in Figure 10-3.

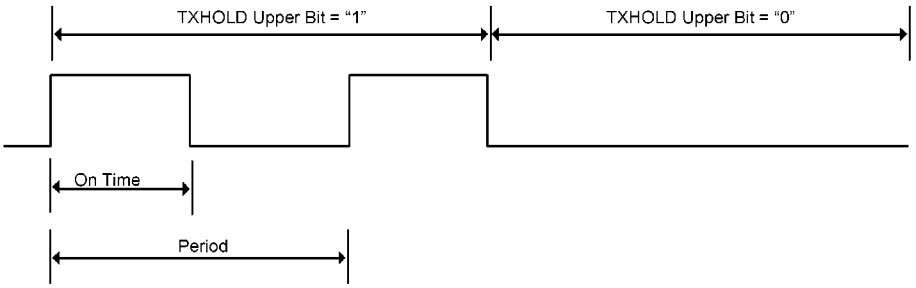


Figure 10-3 Consumer IR Signal Generation

10.2.5 Related Interrupts

IRCONSMINT:

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

10.3 Two-Way Communication Via IR

10.3.1 Requirements

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

For the FSK mode transmit direction, external communication IR analog circuitry encodes the UARTB output data using an FSK modulation scheme, which modulates at two different frequencies depending on whether the data is a "1" or a "0". The carrier frequency is 1375 kHz and the mark and space frequencies are 1325 kHz and 1425 kHz, respectively. For the receive direction, external communication IR analog circuitry amplifies the received photo-diode signal and demodulates and decodes the FSK signal before feeding the UARTB input. The transmitter and receiver portions of the external communication IR analog circuitry contain individual power-down control for power management purposes. The FSK communication mode operates at a data rate of 2400 to 36000 bps at 3 meters.

For the IRDA mode transmit direction, the UARTB output data directly drives the external analog LED circuit. For the receive direction, the received photo-diode signal is externally amplified before feeding the UARTB input. Additional control bits in the UART Module also allow for various narrow pulse options to support the IRDA standard. When configured for these options, the UART transmit output pulses are narrower than normal and the UART receiver circuit also expects the same narrow pulse format for the input data. The IRDA communication mode operates at a data rate up to 115 kbps at 1 meter.

10.3.2 Block Diagram

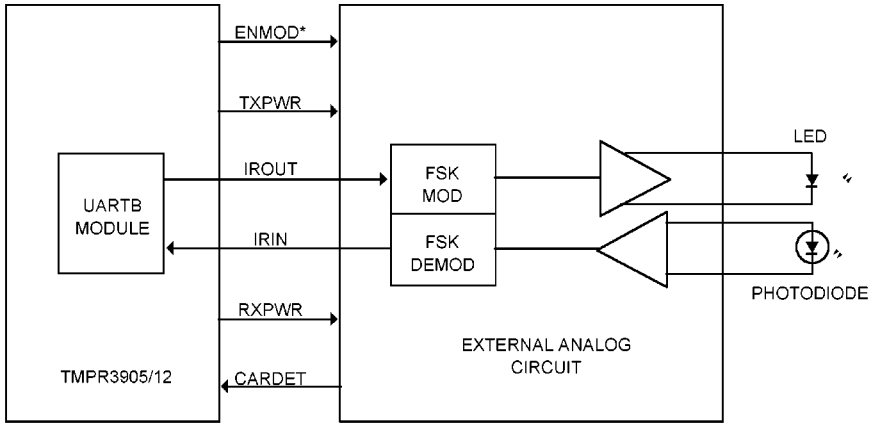


Figure 10-4 FSK Communication Circuit

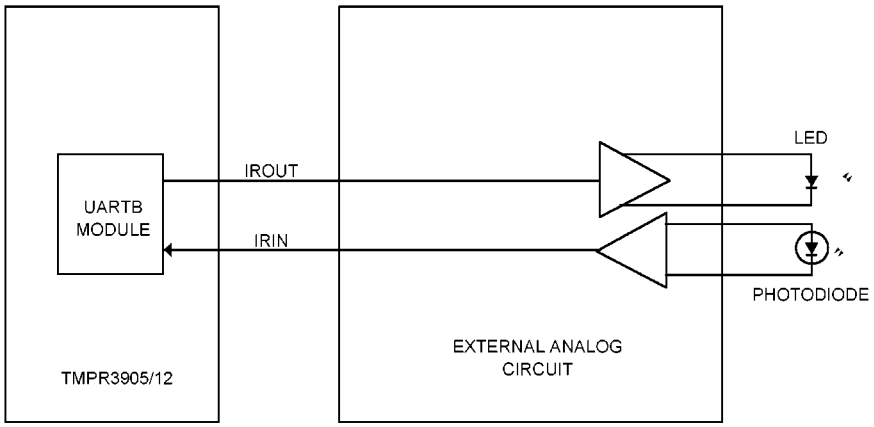


Figure 10-5 IRDA Communication Circuit

10.4 Carrier Detect State Machine

10.4.1 Requirements

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

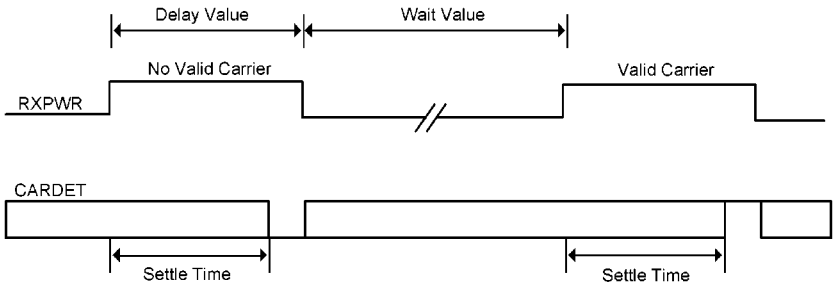


Figure 10-6 Carrier Detect State Timing

10.4.2 Block Diagram

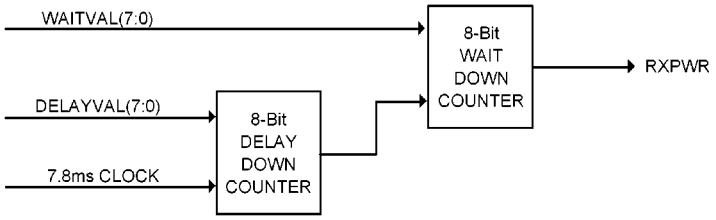


Figure 10-7 Carrier Detect Block Diagram

10.4.3 Functional Description

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

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

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

(DELAYVAL + 1) *7.8 ms

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

10.4.4 Related Interrupts

CARSTINT:

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

POSCARINT:

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

NEGCARINT:

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

10.5 IR Registers

10.5.1 IR Control 1 Register

OFFSET = $0A0:

Bit	Label	RESET	Read/Write
31-25	Reserved		
24	CARDET	—	R
23-15	BAUDVAL[8:0]	X	R/W
14-5	Reserved		
4	TESTIR	0	R/W
3	DTINVERT	0	R/W
2	RXPWR	0	R/W
1	ENSTATE	0	R/W
0	ENCONSM	0	R/W

CARDET: read-only

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

BAUDVAL[8:0]:

These bits are used to pre-scale the 9.216 MHz baud rate clock. The bits are loaded into a 9-bit down-counter that counts at the 9.216 MHz clock rate. The resulting baud clock is used to clock the Period Width Counter and OnTime Width Counter.

TESTIR:

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

DTINVERT:

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

RXPWR:

This bit is connected to the RXPWR pin.

ENSTATE:

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

ENCONSM:

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

10.5.2 IR Control 2 Register

OFFSET = $0A4: write-only

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

PER[7:0]: write-only

These bits are used to define the period of the Consumer IR pulses. The bits are loaded into an 8-bit down-counter that counts at the rate defined by the Baud Value. The resulting period is given by the following equation.

Period = (PER + 1) * (BAUDVAL + 1) * (1 / 9.216 MHz)

ONTIME[7:0]: write-only

These bits are used to define the On Time of the Consumer IR pulses. These bits are loaded into an 8-bit down-counter that counts at the rate defined by the Baud Value. The resulting On Time is given by the following equation.

On Time = ONTIME * (BAUDVAL + 1) * (1 / 9.216 MHz)

DELAYVAL[7:0]: write-only

These bits are used to define the amount of time that the RXPWR pin will be a logic "1" when the Carrier State Machine logic is enabled. The bits are loaded into an 8-bit down-counter that counts at a 7.8 ms rate. Thus the time is programmable from 7.8 ms to 2.0 seconds.

Delay Time = (DELAYVAL + 1) *7.8 ms

WAITVAL[7:0]: write-only

These bits are used to define the amount of time to wait before asserting the RXPWR pin to a logic "1" when the Carrier State Machine logic is enabled. The bits are loaded into an 8-bit down-counter that counts whenever the DELAYVAL counter reaches a count of zero. Thus the time is programmable from 7.8 ms to 511 seconds.

Wait Time = (DELAYVAL + 1) * (WAITVAL + 1) *7.8 ms

10.5.3 IR Holding Register

OFFSET = $0A8: write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	TXHOLD[15:0]	X	W

TXHOLD[15:0]: write-only

These bits are used to define the number of pulses in a burst or the amount of time to keep the IROUT pin low. The lower seven bits of the upper byte and the lower seven bits of the lower byte are used as values to load into a 7-bit counter that counts the number of periods as defined by the Period Value. The upper bit in each byte if set to a logic "1" causes pulses on the IROUT pin and if set to a logic "0" causes the IROUT pin to remain low for a given count.

Before setting the ENCONSM bit the TXHOLD register must be pre-loaded with the first valid word of data.

SECTION 11 IrDA Module

11.1 Overview

- IrDA 1.1 Conformance
  - 2.4 Kbps, 9.6 Kbps, 19.2 Kbps, 38.4 Kbps, 57.6 Kbps, 115.2 Kbps (IrDA 1.0)
  - 1.152 Mbps NRZ
  - 4.0 Mbps 4ppm/single pulse
  - Does not support 0.576 Mbps or 4 Mbps/double pulse
- Generate and check CRC by hardware (transfer rate 1.152 Mbps or higher)
- Filtering based on IrDA 1.0 standard
  - Use hardware to detect and convert BOF, EOF, and bit patterns requiring data conversion (filtering can be turned off).
- Unicast mode (address filtering) supported: 1.152 Mbps, 4 Mbps
  - Can filter an incoming packet by checking the address field in the packet.
  - Up to four addresses are settable. Whether the broadcast is received or not can be set.

11.1.1 Related Pins

IRINA: INPUT

This pin is an input for the receive signal to the IrDA(FIR) module.

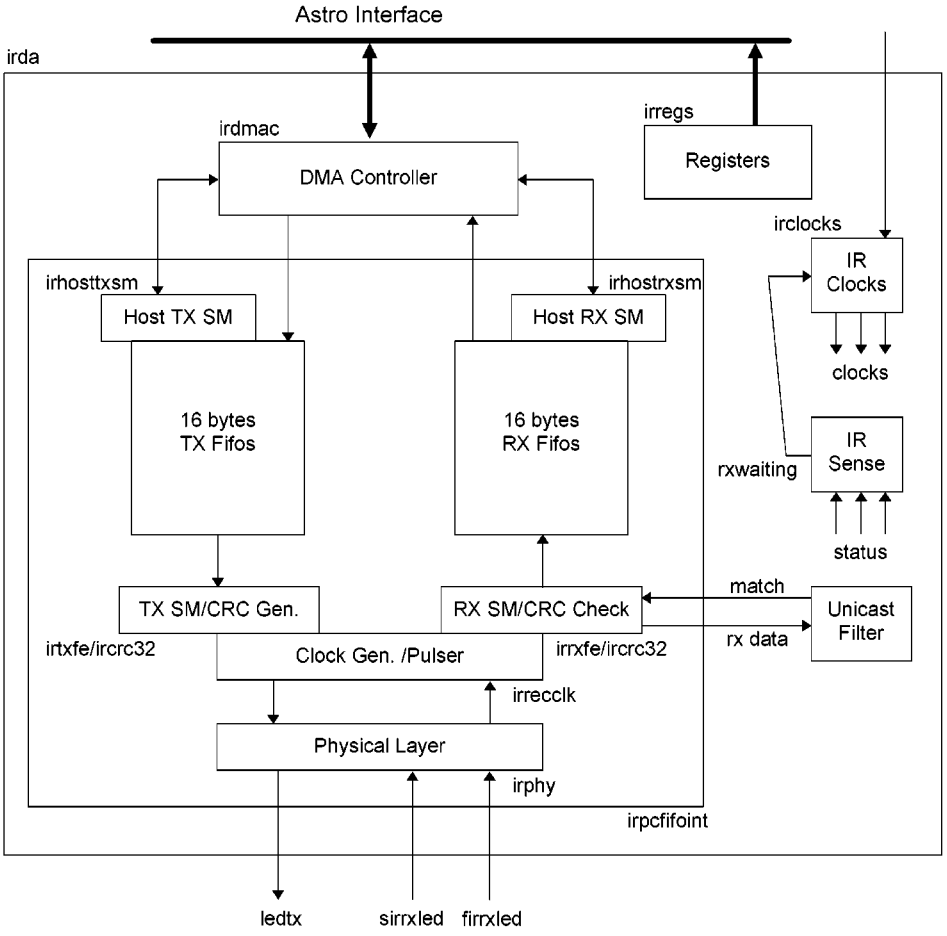
IRINB: INPUT

This pin is an input for the receive signal to the IrDA(FIR) module.

FIROUT: OUTPUT

This pin is an output for the transmit signal from the IrDA(FIR) module.

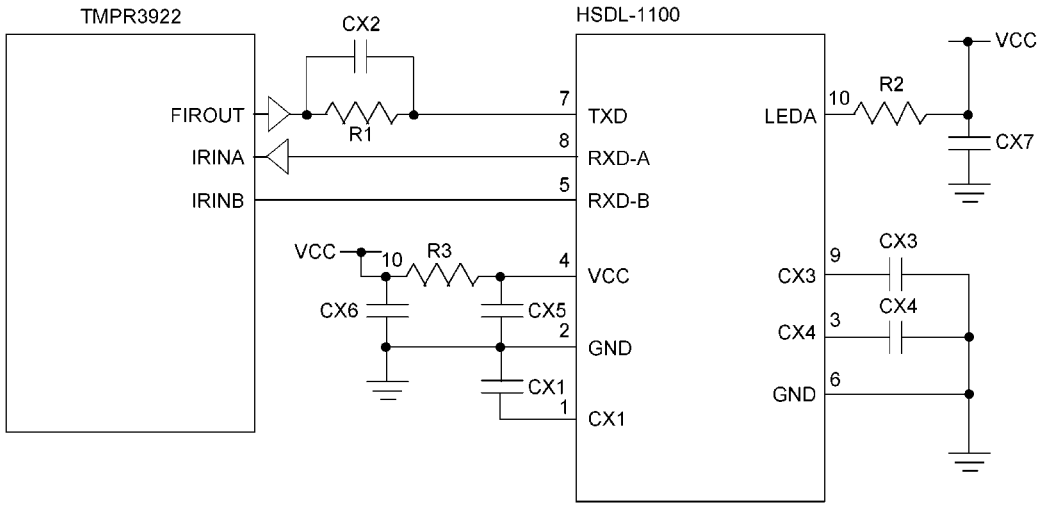
11.1.2 Block Diagram



11.1.3 Connection Example

FIR Connection Example (1/2)

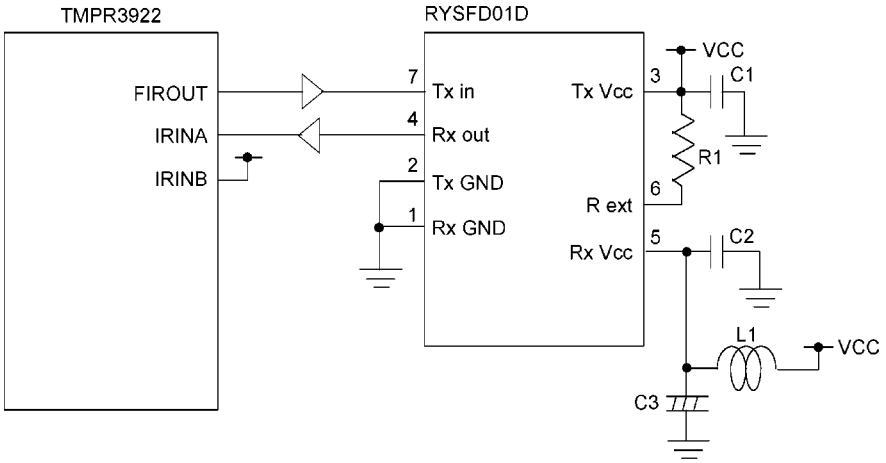
(1) HP HSDL-1100



Parts	
R1	560Ω , ±5% , 0.125Watt
R2	4.7Ω , ±5% , 0.5Watt
R3	10Ω , ±5% , 0.125Watt
CX1	0.47μF , ±10%
CX2	220 pF , ±10%
CX3	4700 pF , ±10%
CX4	0.01μF , ±10%
CX5	0.47μF , ±20%
CX6	6.8μF
CX7	0.47μF , ±20%

FIR Connection Example (2/2)

(2) Sharp RY5FD01D



Parts	
R1	10Ω , ±5% , 0.5Watt
C1	2.2 μF
C2	0.1 μF
C3	100 μF
L1	33 μH

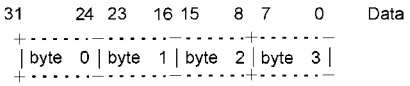
11.2 DMA

DMA

- Up to 64 ring descriptors for transmitting or receiving (8 bytes per ring)
- At the beginning of communication, the transfer speed is 9,600 bps (fixed) and the maximum data size is 64 bytes. After connection is established, the communication speed is selected.
- Transmits/receives one packet per descriptor.
- One packet size is 2 KB max. (based on IrLAP standard)
- Up to seven packets can be transmitted consecutively (based on IrDA standard).
- Incorporate 16-byte FIFO registers for transmitting or receiving.
- DMA transfers to/from memory use a 4-byte read and a 1-, 2-, or 4-byte write.
- The DMA interface operating frequency is 8 MHz.

The DMA transfer is controlled by a DMA descriptor, known as a ring, in memory.
The IrDA controller: starts DMA read for the ring after the write access to the Infrared Ring Prompt register (IrPrompt) or the completion of DMA transfer. The IrDA controller also records such information as DMA completion status to the ring.
The processor: can detect the completion of DMA transfer or an error in DMA transfer either by reading the ring status information shown in the Infrared Ring Pointer Readback register (IrRingPointer) or by interrupt.
One ring consists of eight bytes. Up to 64 rings can be set for transmitting or receiving.
The Infrared Ring Base Address register (IrRingBase) sets the receive ring start address. The transmit ring start address is fixed to IrRingBase + 512 bytes.

Memory Order



The address consists of 30 bits (irdadmaaout [31:2]). The lower two bits are fixed to 00. The valid byte position can only be determined by a byte enable bit setting (irda_be).

irda_be	byte_0	byte_1	byte_2	byte_3
1000	○	×	×	×
1100	○	○	×	×
1110	○	○	○	×
1111	○	○	○	○
0111	×	○	○	○
0011	×	×	○	○
0001	×	×	×	○

The IrDA controller transmits data from byte 0.
The controller stores the received data to memory from byte 0.
The IrDA controller can perform 1-word (4-byte) reads and 1-, 2-, and 4-byte writes only.

- Transmit ring format

byte 0	15:12	unused, TxCount [11:8]	byte 1	7:0	TxCount [7:0]
byte 2	15:8	TxStatus [7:0]	byte 3	7:0	unused
byte 4	15:8	Address [31:24]	byte 5	7:0	Address [23:16]
byte 6	15:8	Address [15:8]	byte 7	7:0	Address [7:0]

TxCount = 12 bits = Transmit data size (2 KB max.)
Address = 32 bits = Transmit data start address

- TxStatus[7:0]

Bit 7 R/W: HW_OWNS

Bit 7 = 1 indicates the ring is valid. If valid, the controller transmits the data to the IR link. When transmission is complete, this bit is cleared by hardware.

Bit 6 R/W: DISTX_CRC

Disables the transmit CRC. When set to 1, a CRC cannot be generated or transmitted by hardware. In SIR mode this bit is ignored. When transmission is complete, this bit is cleared by hardware.

Bit 5 R/W: Bad_CRC

When set to 1, the controller transmits an erroneous CRC. The CRC is used for testing the receiving device CRC check. When transmission is complete, this bit is cleared by hardware.

Bit 4 R/W: Need_Pulse

When set to 1, the controller transmits a serial interaction pulse (SIP) after a packet transmission (2 μ s). When transmission is complete, this bit is cleared by hardware. This bit is valid in other than SIR mode.

Bit 3 R/W: Force_Underrun

When set to 1, the controller forces to generate an underrun (for diagnosis). To generate an underrun, TxCount must be set to at least 18. When transmission is complete, this bit is cleared by hardware.

Bit 2 R/W: Request_To_Clear_ENTX

When set to 1, the controller clears the transmit enable bit (IrConfig0.ENTX) of the Infrared Config 0 register (IrConfig0) at the completion of transmission. To issue a transmission complete interrupt, this bit must be set to 1.

Bit 1: Unused

Bit 0 Read-only: Underrun

This bit is set by hardware. Bit 0 = 1 indicates that an underrun occurred during a packet transmission.

- Receive ring format

byte 0	15:12	unused, RxCount [11:8]	byte 1	7:0	RxCount [7:0]
byte 2	15:8	RxStatus [7:0]	byte 3	7:0	unused
byte 4	15:8	Address [31:24]	byte 5	7:0	Address [23:16]
byte 6	15:8	Address [15:8]	byte 7	7:0	Address [7:0]

RxCount = 12 bits = Receive data size (2 KB max.)
Address = 32 bits = Start address for storing the receive data

- RxStatus[7:0]

Bit 7 R/W: HW_OWNS

Bit 7 = 1 indicates the ring is valid and data were received from the IR link and stored in memory. When reception is complete, this bit is cleared by hardware.

Bit 6 Read-only: Phy_Error

This bit is set by hardware. Bit 6 = 1 indicates detection of a reception decode error.
In 1.152 Mbps transfer mode, incorrect bitstreams are received. Incorrect bitstreams include such cases as data that are not 4ppm or when 1 is successively repeated seven or more times. At that time, the CRC_Error bit must be set to 1.
If Phy_Error is detected before the four bytes are fully received, the packet is ignored for reception (the ring is not updated).

Bit 5 Read-only: CRC_Error

This bit is set by hardware. Bit 5 = 1 indicates detection of a CRC error (only in transfer modes of 1.152 Mbps or higher).

Bit 4 Read-only: Length

This bit is set by hardware. Bit 4 = 1 indicates that the packet being received reached the maximum packet length.
The Rx field of the Max Packet Length register (IrMaxLength) specifies the maximum packet length.
In IrAD 1.0 mode (or during filtering), the next ring is opened and any excess packet data are saved to the memory area.
In other modes, the next ring is not opened until detection of the end of the packet. As a result, any excess packet data are discarded.

Bit 3 Read-only: Overrun

This bit is set by hardware. Bit 3 = 1 indicates that the reception FIFO register has overflowed. Overflow occurs when DMA transfer to the memory is slower than reception of the packet.

Bit 2 Read-only: SIRBAD

When IrDA 1.0 filtering is enabled (IrConfig0.SIR_DEC_ENB), this bit is set by hardware when a second BOF is detected before detection of an EOF (end of frame) after detection of a BOF (beginning of frame) or when an abort (ESC[7d hex], ABORT [e1 hex]) is received.

Bit 1 Read-only: RXEOF

This bit is set when a packet is received normally.
In IrDA 1.0 filtering mode, after the maximum packet length is reached, the RXEOF bit of the next ring is set to 1 if the end of the packet (EOF (c1 hex) is detected without any further data being received.

Bit 0 unused

- Description of Operation

(1) Set the receive ring start address in the Infrared Base Address register (IrRingBase).
(2) Clear the PhyAndClock bit of the Infrared Enable register (IrEnable).
(3) Set the Infrared Config 0 register (IrConfig0).
(4) Set the Infrared Enable register (IrEnable).
(5) Set the Infrared Phy Config register (IrPhyConfig).
(6) Set the transmit/receive ring.
    Transmit ring: IrRingBase + 512 + IrRingPoint.Tx
    Receive ring: IrRingBase + IrRingPoint.Rx
    First, set the address, then the status, in that order.
(7) Write to the Infrared Prompt register (IrPrompt).

When the IrPrompt register is written, the IrDA controller reads the receive ring, then the transmit ring.

When not transmitting or receiving, the HW_OWNS bit is set to 0.

However, when transmitting, even if not receiving, the HW_OWNS bit of the receive ring descriptor is set to 1. Or, the IrConfig1.RxAlways bit is set to 1.

When the transmission or reception of one packet (ring) is complete, the controller increments IrRingPoint.Rx/Tx and reads the following ring. If the HW_OWNS bit is set to 1, the controller transmits/receives the next packet. If the bit is set to 0, the controller does not transmit/receive the next packet.

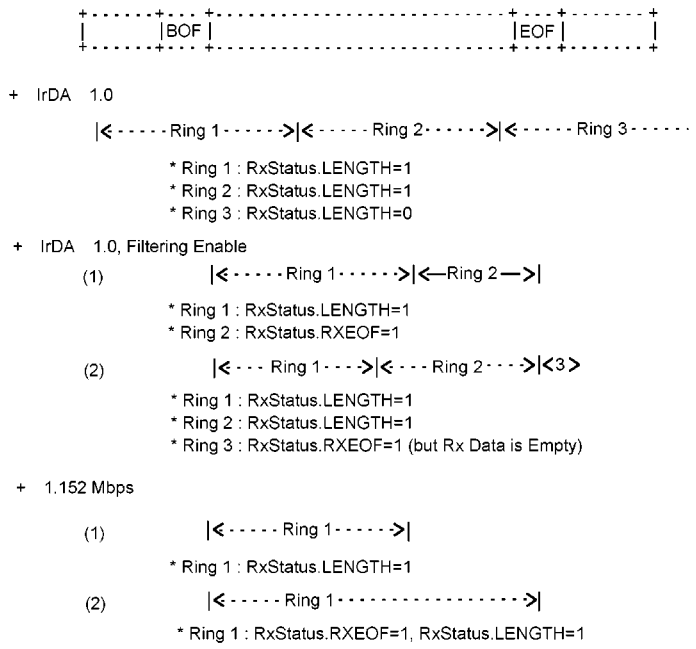
When setting the ring, be sure to set the address, then the HW_OWNS bit, in that order. This is necessary because the IrDA controller may be reading the ring. The IrDA controller first reads the status

field (HW_OWNS), then the address.

Normally this is not a problem, as sufficient time elapses between the transmit/receive complete interrupt and the setting of the next ring.

The CRC is also stored in memory area.

Relationship between reception of the packet and the ring



11.3 Power Management

Power Management

(1) Turn off the power supply of the optical transmit/receive module (assert the XCVROFF signal).
(2) Stop the internal clock (clk40mhz) (data receive disabled).
(3) Provide a low power dissipation wait-to-receive mode.

In (1) above, the XCVROFF signal is used with RXPWR in the IR modul.
Use the clock module for (2) above.

- Low power dissipation wait-to-receive mode
  - Not transmitting.
  - Remain in wait-to-receive mode.
  - Ring is not open.
  - Not receiving infrared.

When all the above conditions are satisfied, the Irda clock automatically stops. If the above conditions are not satisfied, the Irda clock resumes.

When infrared signals are received, the clock is supplied for some time while the controller determines whether the packet has started (STA).

The time to detect the start of the packet and the time to supply the clock are as follows.

4Mbps: 2.0us > 875ns = 125ns * 7 (Pre :1/8 light_on, firrxbsy)
1Mbps: 8.0us > 6us = 1.0us * 6 (STA:1/7 light_on, mirrxbsy)
IrDA-1.0 (2.4k) : 3.0ms > 1.25ms = 416.7us * 3 (sync+STA+modirrxbsy)

11.4 Reset

Reset

- Asynchronous reset
- Software reset

Setting the EnCore bit of the Infrared Config 1 register (IrConfig1) to 0 resets the IrDA controller.
Setting IrConfig1.EnCore = 0 does not reset the Infrared Config 1 register (IrConfig1).
When using the clock stop function in the clock module, first set the IrConfig1.EnCore bit must be set to 0.

11. 5 Programming Guide

(1) IrDA-1.0 mode without filter

1. Clear PhyAndClock BIT in the IrEnable REGISTER.
2. Define IrMaxLength. Rx to correct value.
3. Set the following bits in IrConfig0
   Receive = "0000 0e10" hex
   Transmit = "0000 1610" hex
   + Either EnTx or EnRx. (if both are set, EnTx blocks EnRx).
   + EnDMAC
   + RecvAny
   + EnSir
4. Set Infrared Phy register Baud Rate and Pulse width to correct value.
5. Set PhyAndClock BIT in the IrEnable REGISTER.
6. Read IrEnable register, should
   Receive = '0000 8a00 hex.
   Transmit = '0000 8c00 hex.
7. Write to the IrPrompt location once rings are set.

(2) IrDA-1.0 mode with filter

1. Clear PhyAndClock BIT in the IrEnable REGISTER.
2. Define IrMaxLength. Rx to correct value.
3. Set the following bits in IrConfig0
   Receive = "0000 0e18" hex
   Transmit = "0000 1618" hex
   + Either EnTx or EnRx. (if both are set, EnTx blocks EnRx).
   + EnDMAC
   + RecvAny
   + EnSir
   + EnSirFilter
4. Set Infrared Phy register Baud Rate and Pulse width to correct value.
5. Set PhyAndClock BIT in the IrEnable REGISTER.
6. Read IrEnable register, should
   Receive = "0000 8a00 hex.
   Transmit = "0000 8c00 hex.
7. Write to the IrPrompt location once rings are set.

(3) 1.152 Mbps mode

1. Clear PhyAndClock BIT in the IrEnable REGISTER.
2. Define IrMaxLength. Rx to correct value.
3. Set the following bits in IrConfig0
   Receive = "0000 0ca0" hex
   Transmit = "0000 14a0" hex
   + Either EnTx or EnRx. (if both are set, EnTx blocks EnRx).
   + EnDMAC
   + CRC16n32
   + EnMir

4. Set Infrared Phy register Baud Rate and Preamble width to correct value.
5. Set PhyAndClock BIT in the IrEnable REGISTER.
6. Read IrEnable register, should
   Receive = '0000 9300 hex.
   Transmit = '0000 9500 hex.
7. Write to the IrPrompt location once rings are set.

(4) 4 Mbps mode

1. Clear PhyAndClock BIT in the IrEnable REGISTER.
2. Define IrMaxLength. Rx to correct value.
3. Set the following bits in IrConfig0
   Receive = "0000 0c40" hex
   Transmit = "0000 1440" hex
   + Either EnTx or EnRx. (if both are set, EnTx blocks EnRx).
   + EnDMAC
   + EnFir
4. Set Infrared Phy register Baud Rate and Preamble width to correct value.
5. Set PhyAndClock BIT in the IrEnable REGISTER.
6. Read IrEnable register, should
   Receive = '0000 a200 hex.
   Transmit = '0000 9400 hex.
7. Write to the IrPrompt location once rings are set.

11.6 Interrupt

Interrupt

- Transmission complete interrupt/clear
- Reception complete interrupt/clear
- Transmission error interrupt/clear
- Reception error interrupt/clear
- SIP interrupt/clear

The IrDA controller outputs the interrupt signals and inputs the interrupt clear signals. The Interrupt Enable/Status register is located in the interrupt module.

(1) Transmission complete interrupt (IR TXCINT)

When the Request_To_Clear_Entx bit of the transmit ring is set to 1, the transmission of the packet completes normally and an interrupt occurs after the HW_OWN bit of the transmit ring is cleared to 0. A transmission error interrupt occurs on detection of a transmission error.

(2) Reception complete interrupt (IR RXCINT)

Whenever a packet is received normally, the HW_OWN bit of the receive ring is cleared to 0. Then, after the RXEOF bit is set, an interrupt occurs.

In IrDA 1.0 mode (or during filtering), an interrupt also occurs when the reception data size reaches the maximum packet length. In other reception modes, a reception error interrupt occurs when the reception data size exceeds the maximum packet length.

(3) Transmission error interrupt (IR TXEINT)

A transmission error interrupt occurs on detection of a transmission FIFO register underrun.

Detection of an underrun transmits the bad CRC and terminates the transfer of the packet.

Underruns occur even in IrDA 1.0 mode.

The bad CRC is also transmitted in SIR mode. Abnormal data are transmittedeven with the DISTX_CRC bit set.

The interrupt is not issued immediately on detection of an error.The IrDA controller first finishes counting the transfer size, then generates the interrupt at the close of the ring.

(4) Reception error interrupt (IR RXEINT)

This interrupt occurs on detection of a physical layer error or a reception FIFO register overrun.

A physical layer detects the following errors.

- An abort (7del) is received in IrDA 1.0 filtering mode.Or, after reception of a BOF (c0), a second BOF is received before reception of an EOF. At that time, the SirBad bit of the Reception Ring Status register is set.
- A CRC error is detected in transfer modes of 1.152 Mbps or higher. At that time, the CRC_Error bit of the Reception Ring Status register is set.
- An illegal bitstream pattern is detected during data reception. At that time, the Phy_Error bit of the
- Reception Ring Status register is set.
- In 1.152 Mbps transfer mode, when the reception data size exceeds the maximum packet length, an interrupt occurs. The controller detection the end of the packet and issues the interrupt after closing the ring. At that time, the Length bit of the Reception Ring Status register is set.

After the reception FIFO register is full, the reception of further data issues an overrun and the excess data are discarded. Subsequent reception data are stored to memory area. Detection of the end of a packet issues the close of the ring. After the status is written to the ring, an interrupt occurs.

(5) SIP interrupt (IR SIRPXINT)

An interrupt occurs when a serial interaction pulse (SIP) is received.

When IrEnable.PhyAndClock, IrEnable.EnRx, and IrEnable.EnSir = 1 and IrEnable.EnTx = 0, if IrConfig1.SipIntr = 0, an interrupt occurs when a pulse of 1.4 μs or longer is received. If IrConfig1.SipIntr = 1, an interrupt occurs if a pulse is not detected for 500 ms or longer.

11.7 IrDA Registers

11.7.1 Infrared Ring Pointer

OFFSET=$28:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15	Reserved		
14	Reserved		
13-8	Tx[5:0]	X	R
7	Reserved		
6	Reserved		
5-0	Rx[5:0]	X	R

Tx[5:0]:

These bits provide the index of current Transmit pointer.

Rx [5:0]:

These bits provide the index of current Receive Pointer

11.7.2 Infrared Base Address

OFFSET=$2C:

Bit	Label	RESET	Read/Write
31-10	Addr[31:10]	0	R/W
9-0	Reserved		

Addr [31:10]:

These bits define the ring base address.

Receive Ring [31:0] = {Addr[31:10], 00_0000_0000}
Transmit Ring [31:0]= {Addr[31:10], 10_0000_0000}

11.7.3 Infrared Ring Size

OFFSET=$30:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-12	TX[3:0]	0	R/W
11-8	Rx[3:0]	0	R/W
7-0	Reserved		

Tx [3:0]:
These bits define the number of Tx ring entries.

Rx[3:0]:
These bits define the number of Rx ring entries.

Tx/Rx	RingSize
0000	4
0001	8
0011	16
0111	32
1111	24

11.7.4 Infrared Ring prompt

OFFSET=$34:

Bit	Label	RESET	Read/Write
31-1	Reserved		
0	Ir Promt	X	W

Infrared Ring Prompt register (IrPrompt), write-only Starts transmit/receive operations.Writing to this register triggers a read of the current transmit/receive ring. If HW_ONWS is set to 1, DMA begins.

11.7.5 Infrared Config 0

OFFSET=$38:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15	TX On Loop	0	R
14	Loop Back	0	R/W
13	Reserved		
12	EnTx	0	R/W
11	EnRx	0	R/W
10	EnDMAC(EN_MEMSCHD)	0	R/W
9	RecvAny (RECV_ANY)	0	R/W
8	Reserved	0	R/W
7	CRC16n32	0	R/W
6	EnFir	0	R/W
5	EnMir	0	R/W
4	EnSir	0	R/W
3	EnSirFilter	0	R/W
2	SirTest	0	R/W
1	InvertTx	0	R/W
0	InvertRx	0	R/W

Tx On Loop:

Normally, does not output ledtx in LoopBack mode. Setting to 1 outputs ledtx even in LoopBack mode. However, note that the pulse width is not guaranteed.

Loop Back:

This bit enables internal loopback. See programming section when set to 1.

EnTx:

Setting to 1 enables transmission at the physical layer. Setting to 0 resets the transmission encoding circuit.

EnRx:

Setting to 1 enables reception at the physical layer. (Reception is disabled if EnTx is active but LoopBack is inactive.) Setting to 0 resets the reception encoding circuit.

EnDMAC (EN_MEMSCHD):

This bit enables the memory scheduler. Allows either DRAM access or RAM by the infrared controller when this bit is set to 1. Setting to 0 resets the SIR reception decoding circuit and the DMA controller and re-set the ring pointer.

RecvAny (RECV_ANY):

Enable Reception of small/runt packets (< 4 bytes). See programming section when set to 1. Setting this bit performs DMA writes in single-byte units.

CRC16n32:

Setting to 0 is for 32 bit CRC (4 Mbps), setting to 1 is for a 16 bit CRC(1.152 Mbps).

EnFir:

Setting to 1 is for 4 Mbps mode.

EnMir:

Setting to 1 is for 1.152 Mbps mode.

EnSir:

Setting to 1 is for IrDA-1.0 mode.

EnSirFilter:

Setting to 1 is to enable IrDA-1.0 Byte FILTER on the receiver when the SIR mode bit is bit is set. Detect and delete the BOF or EOF. Only received data between the BOF and EOF are stored to memory area. The escape character also is converted.

SirTest:

Setting to 1 allows the SIR FILTER to be used in no SIR mode. In a physical layer, packets are received in 1.152 Mbps/4 Mbps transfer modes. After excluding such data as STA, STO, and PRE, IrDA 1.0 filtering is performed.

InvertTx:

Setting to 1 Inverts TX LED line on way out.

InvertRx:

Setting to 1 Inverts Rx LED line on way in.

11.7.6 Infrared Config 1

OFFSET=$3C:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15	EnCore	0	R/W
14-12	Reserved		
11	XCVRON	0	R/W
10	Reserved		
9	CanRx	0	R/W
8	CanSirRx	0	R/W
7	SirIntr	0	R/W
6	RxInput	0	R/W
5	RxAlways	0	R/W
4-0	Reserved		

EnCore:

Setting to 0 resets the IrDA controller. After enabling the clock to the IrDA controller located in the clock module, set this bit to 1.

XCVRON:

The power is supplied to the optical transmit/receive module when this bit is set to 1.

CanRx:

The irda model enters into the low power dissipation mode for wait-to-receive status when this bit is set to 1.

CanSirRx:

The automatic Irda clock stops at SIR wait-to-receive only when this bit is set to 1.

SirIntr:

When set to 0, an interrupt occurs when a SIP pulse (1.4 μ s or longer) is received. When set to 1, an interrupt occurs if nothing is received for 500 ms or longer.
When set to 0, an interrupt occurs whenever one byte is received.
When set to 1, an interrupt occurs at every packet, assuming that packets use the IrLAP standard.

RxInput:

This bit defines the number of signals input from the photodetector module.
Setting to 0 specifies two signals.
- firrxled 1.152 Mbps/4 Mbps reception
- sirrxled IrDA 1.0 reception

Setting to 1 specifies one signal.
- firrxled fixed to 1 or 0
- sirrxled reception

RxAlways:

This bit ensures the HW_OWN bit of the receive ring is always read as 1.
This bit eliminates the need for setting the HW_OWN bit of the receive ring each time by software.
However, when data are actually received, this bit must be set to 0.

11.7.7 Infrared SIR flag

OFFSET=$40:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-8	EOF [7:0]	$C1	R
7-0	BOF [7:0]	$C0	R

EOF[7:0]:

When these bits are read, $C1 is returned.

BOF[7:0]:

When these bits are read, $C0 is returned.

11.7.8 Infrared Enable

OFFSET=$44:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15	PhyAndClock	0	R/W
14	ConfigError	X	R
13	FinOn	X	R
12	MirOn	X	R
11	SirOn	X	R
10	EnTx	X	R
9	EnRx	X	R
8	CRC16n32	X	R
7-1	Reserved		
0	Broadcast	0	R/W

PhyAndClock:

To validate the config registers (IrConfig0, IrPhyConfig), PhyAndClock must be set first to 0 , then to 1. This copies the settings to the IrEnable and IrConfigPhy registers. To check whether the configurations are now valid, read the IrEnable register and check that the PhyAndClock bit of the IrEnable register is set to 1.
When a config error occurs, the PhyAndClock bit is set to 0 and the ConfigError bit is set to 1.

ConfigError:

This bit is set to 1 when a config error occurs.

FirOn:

The FIR configuration is valid when this bit is "1".

MirOn:

The MIR configuration is valid when this bit is "1".

SirOn:

The Sir configuration is valid when this bit is "1".

EnTx:

The TX BIT is enable when this bit is "1".

EnRx:

The RX to frontend and filtered version is enable when this bit is "1".

CRC16n32:

This bit provide the CRC size.
0:32bit , 1:16bit

Broadcast:

Setting to 1 receives the packet with the broadcast address (8b1111_111x) unconditionally (default) when unicast filtering is performed.

11.7.9 Infrared Config to PHY

OFFSET=$48:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-10	BaudRate [5:0]	X	R
9-5	PulseWidth [4:0]	X	R
4-0	Preamble [4:0]	X	R

BaudRate [5:0]:

These bits provide the status of the BaudRate of Data. (see programming section)

PulseWidth [4:0]:

These bits provide the status of the Width of pulse for SIR signal and also indication pulse.
(see programming section)

Preamble [4:0]:

These bits provide the status of the Number of Preamble bytes to send in MIR and FIR mode. For MIR this is the number of start flags plus 1, for FIR it is the number of preamble bytes plus 1.
(ie 0 means 1 byte) (see programming section)

11.7.10 Infrared Phy Config

OFFSET=$4C:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-10	Baud Rate [5:0]	0	R/W
9-5	Pulse Width [4:0]	0	R/W
4-0	Preamble[4:0]	0	R/W

	Baud Rate	Pulse Width Min	Nom	Max	Preamble	Period
2400bps	47	0	25		don't care	416.66us
9600bps	11	0	25		don't care	104.16us
19200bps	5	0	25		don't care	52.08us
38400bps	2	0	25		don't care	26.04us
57600bps	1	0	25		don't care	17.36us
115200bps	0	0	25		don't care	8.68us
1.152Mbps	0	8			1	868.06ns
4Mbps	0	dont care			14	250.00ns

Setting the Pulse Width bit to 0 specifies 3/16 pulse width.
Setting Pulse Width to 12 specifies 1/16 MHz x (25 + 1) = 1.625 μ s.

Baud Rate [5:0]:

These bits define the Baud Rate of Data. (See programming section)

Pulse Width [4:0]:

These bits define the Width of pulse for SIR signal and also indication pulse. (See programming section)

Preamble [4:0]:

These bits define the Number of Preamble bytes to send in MIR and FIR mode. (See programming section)

11.7.11 Maximum packet length

OFFSET=$50:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	Rx [15:0]	0	R/W

Rx [15:0]:

The lower 12 bits are used to indicate the maximum length in bytes of a receive packet. (See programming section)

11.7.12 Receive Byte Count

OFFSET=$54:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-12	Reserved		
11-0	Rx Count [11:0]	X	R

When SIR filtering is enabled, the BOF and EOF are not counted. This register is reset before the start of the next packet reception.

Rx Count [11:0]:

These bits provide the status of the 12 bit receive counter.

11.7.13 Unicast Filtering, Address Register 1-4

OFFSET=$58:

Bit	Label	RESET	Read/Write
31-24	Addr4 [7:0]	0	R/W
23-16	Addr3 [7:0]	0	R/W
15-8	Addr2 [7:0]	0	R/W
7-0	Addr1 [7:0]	0	R/W

The target of the address comparison between the Unicast Filtering Address Setting register and the Broadcast Packet Through Function Setting register is the seven bits of Addr*[7 - 1].
The lowest bit of Addr*[0] indicates whether the settings of those registers are valid (1) or invalid (0).
If all Addr1 to 4 are the invalid, the unicast filtering is disabled (all packets are received as normal).
When performing unicast filtering, the Broadcast bit (bit 0) of the IrEnable register specifies whether or not to unconditionally receive packets with the broadcast address (1111111x). Setting the bit to 0 specifies no unconditional reception (default). Setting the bit to 1 specifies unconditional reception of broadcast packets.

Valid in other than IrDA 1.0 mode.

11.7.14 Infrared Status Register

OFFSET=$5C:

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-11	Reserved		
11	RxMaxCount	1	R
10	TxUnderrun	X	R
9	RxOverrun	X	R
8	Sleep	X	R
7	unused		
6-4	RxOffTime [2:0]	X	R
3	RxBusy	X	R
2	FirRx	X	R
1	MirRx	X	R
0	SirRx	X	R

RxMaxCount:

Indicates the result of comparison of the reception data size with the maximum packet length in 1.152 Mbps/4 Mbps transfer modes.

TxUnderrun:

Indicates transmission FIFO register underrun.

RxOverrun:

Indicates reception FIFO register overrun.

Sleep:

Indicates automatic low power dissipation mode.

RxOffTime [2:0]:

Indicates the time no light is detected from the photodetector port.

[2:0]		[2:0]	
001	less than 10ms	100	200ms or longer
001	10ms or longer	101	300ms or longer
010	50ms or longer	110	400ms or longer
011	100ms or longer	111	500ms or longer

Valid only when the IrEnable.PhyAndClock and IrEnable.EnRx bits are set to 1, and the IrEnable.EnTx bit is set to 0.

RxBusy:

Indicates that a packet is being received. This bit is set to 1 when the write request to the memory occurs after the data are received. The bit is cleared at a write to a ring for a packet being received.

FirRx:

In 4 Mbps transfer mode, this bit is set to 1 at reception of a PA (preamble) and cleared by reception of STO.

MirRx:

In 1.152 Mbps transfer mode, this bit is set to 1 at reception of STA and cleared by reception of STO.

SirRx:

In IrDA 1.0 with filtering mode, this bit is set to 1 by reception of BOF and cleared by reception of EOF.

SECTION 12 Power Module

12.1 Overview

The System Power Supply must provide several power supplies to various parts of the system. The software uses the SPI serial control interface, along with the PWRCS signal, to control turning on or off the various supplies. The software will only turn on the supplies that are required to perform a particular task. The TMPR3922 must always be provided power as long as there is a good Main or Backup battery in the system, or if a Battery Charger is plugged in. Similarly, there may be other components in the system that must always be powered. These components, along with the TMPR3922, should be supplied power from the VSTANDBY signal.

DRAM or SDRAM should be supplied power from the VCCDRAM signal. The VCCDRAM signal should remain low when VSTANDBY is first asserted, until PWRCS is asserted to turn on the system for the first time. Once the system boots for the first time, the software can decide not to turn off VCCDRAM when the system is subsequently turned off, in order to preserve the contents of memory.

Much of the rest of the system, including ROM, TC35143F, and system buffers are powered by VCC3. The VCC3 signal is only on when the system is on. Other switchable power supplies are provided for components such as the LCD, Magicbus, PCMCIA cards, etc. Each of these supplies will normally be turned off by the software when powering down the system. When the system is on, these supplies will only be turned on if the particular module is being used.

In addition to the System Power Supply control, the Power Module also contains logic to support the TMPR3922 initialization, Clock Enable function, CPU Stop Mode, Stop Timer, and power failure logic. Each of these are described in more detail in the following sub-sections.

12.1.1 Power Signals

VSTANDBY_3.3:

This signal provides power for the TMPR3922 and other components in the system that must never lose power. This signal should always be asserted if there is either a good Main Battery or Backup Battery, or if a Battery Charger is plugged in.

VSTANDBY_2.5:

This signal provides 2.5V power for the TMPR3922 that must never lose power. This signal should always be asserted if there is either a good Main Battery or Backup Battery, or if a Battery Charger is plugged in.

VSTANDBY_SUSPEND

This signal provides 2.5V power for the TMPR3922 in the active mode and 0V power in the suspend mode.

VCCDRAM:

This signal provides power for the DRAM and/or SDRAM. This supply must be off when VSTANDBY is first asserted, and remain off until the system is powered up by the assertion of PWRCS. When the software subsequently powers down the system it may choose to keep this supply on to preserve the contents of memory.

VCC3:

This signal provides power for the ROM, TC35143F, system buffers, and other transient components in the system. This signal will be asserted by the System Power Supply when PWRCS is asserted and will always be turned off when the system is powered down.

In addition to these supplies, the System Power Supply will generate individual switchable power signals for the LCD, Magicbus, PCMCIA cards, and other devices used in the system. These different supplies will vary according to particular system implementations.

12.1.2 Related Pins

PON*: INPUT

This pin serves as the Power On Reset signal for the TMPR3922. This signal must remain low when VSTANDBY is asserted until VSTANDBY is stable. Once VSTANDBY is asserted, this signal should never go low unless all power is lost in the system.

ONBUTN: INPUT

This pin is used as the On Button for the system. Asserting this signal will cause PWRCS to set to indicate to the System Power Supply to turn power on to the system. PWRCS will not assert if the PWROK signal is low.

PWRCS: OUTPUT

This pin is used as the chip select for the System Power Supply. When the system is off, the assertion of this signal will cause the System Power Supply to turn VCCDRAM and VCC3 on to power up the system. The Power Supply will latch SPI commands on the falling edge of PWRCS.

PWROK: INPUT

This pin provides a status from the System Power Supply that there is a good source of power in the system. This signal typically will be asserted if there is a Battery Charger supplying current or if the Main Battery is good and the Battery Door is closed. If PWROK is low when the system is powered off, PWRCS will not assert as a result of the user pressing the ONBUTN or an interrupt attempting to wake up the system. If the device is on when the PWROK signal goes low, the software will immediately shut down the system since power is about to be lost. When PWROK goes low, there must be ample warning so that the software can shut down the system before power is actually lost.

PWRINT: INPUT

This pin is used by the System Power Supply to alert the software that some status has changed in the System Power Supply and the software should read the status from the System Power Supply to find out what has changed. These will be low priority events, unlike the PWROK status, which is a high priority emergency case.

VCC3: INPUT

This pin provides the status of the power supply for the ROM, TC35143F, system buffers, and other transient components in the system. This signal will be asserted by the System Power Supply when PWRCS is asserted, and will always be turned off when the system is powered down.

12.2.2 Power Up

The Power Up sequence timing is shown in Figure 12-2. When the ONBUTN is pressed, the PWRCS signal will be set if the PWROK signal is high. Once the PWRCS signal is set, the System Power Supply must assert VCCDRAM and VCC3 in order to power up the system. Once VCC3 is asserted, the high speed oscillator and PLL that generates the clock signals to the CPU and BIU is asserted. When the oscillator is first enabled, the oscillation can vary wildly until the oscillator and PLL become stable. The stabilization time can vary between 1 ms and 10 ms, depending on the oscillator and PLL of a particular TMPR3922 implementation. In order to prevent the TMPR3922 from using the clock before the clock becomes stable, a circuit in the Power Module will provide a signal ENSYSCLK that will disable the clock until the oscillator and PLL become stable. The circuit uses the 32kHz signal as its reference, since this is known to be good because of the long Power On Reset time. ENSYSCLK is asserted (16-24 ms +512µs after VCC3 is asserted or 4-6 ms +512µs after VCC3 is asserted, depending on the state of the SELC2MS control bit in the Power Control Register.

Once the ENSYSCLK signal asserts, the CPU will start up. The PWRCS signal is then de-asserted to de-select and latch the command into the Power Supply. If the PWROK signal is low when the ONBUTN is pressed, the PWRCS signal will not be asserted. The PWRCS signal will also be asserted when the system is off if an enabled interrupt is set and the PWROK signal is asserted. If an enabled interrupt is set when the system is off, but the PWROK signal is low, then the system will power up once the PWROK signal is asserted since the interrupt is latched. On the other hand, the ONBUTN is not a latched event unless the POSONBUTNINT or NEGONBUTNINT interrupts are enabled. Therefore, the ONBUTN will not latch and wake up the system once PWROK asserts.

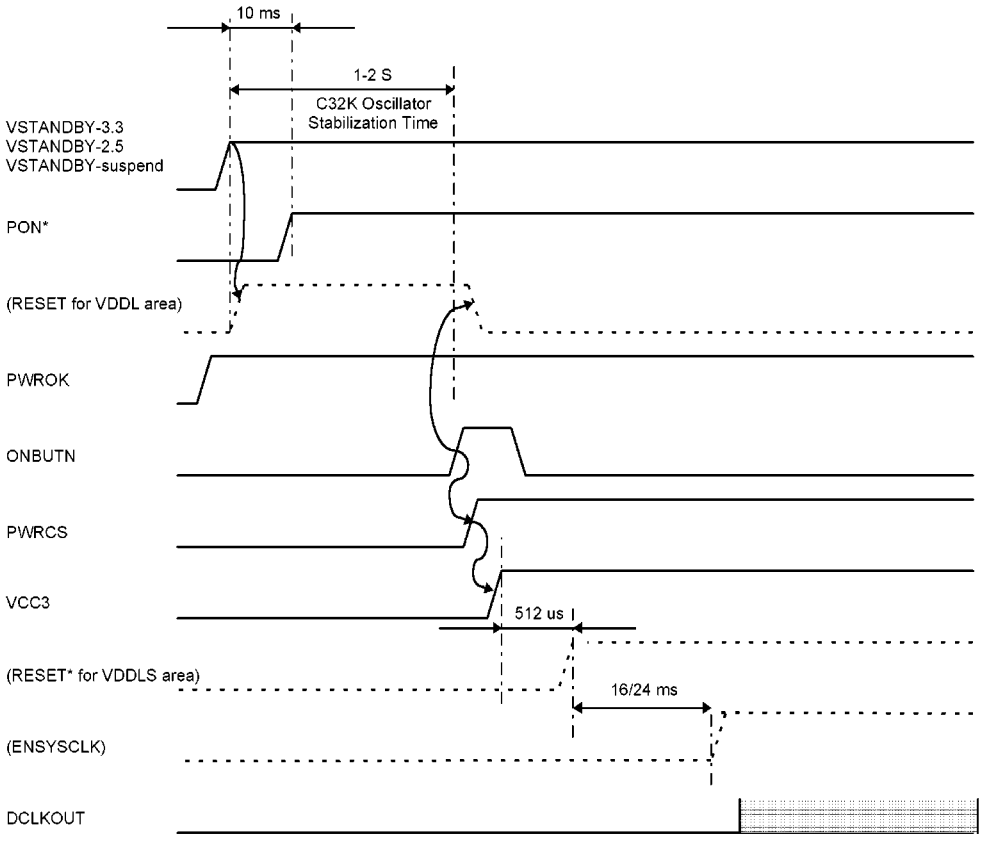


Figure 12-2 Power Up

12.2.3 Power Down Sequence

The system is powered down by following sequence.

1. All modules in the TMPR3922 should be disabled and all clocks in the Clock Module should be turned off.
2. The interrupt for the wake-up events such as Power SW, alarm(RTC), serial DCD are enabled and other interrupts should be disabled.
   Followings are the wake-up interrupt events.
   *Interrupt Status 3 Register (OFFSET = $108)
   MFIOPOSINT [31:0]
   *Interrupt Status 4 Register (OFFSET = $10C)
   MFIONEGINT [31:0]
   *Interrupt Status 5 Register (OFFSET = $110)
   RTCINT
   ALARMINT
   PERINT
   STPTIMERINT
   POSPWRINT
   NEGPWRINT
   POSPWROKINT
   NEGPWROKINT
   POSONBUTNINT
   NEGONBUTNINT
   *Interrupt Status 8 Register (OFFSET = $138)
   IOPOSINT [15:0]
   IONEGINT [15:0]
3. The context such as general purpose register and Program counter inside TX3920 processor core and the special register inside TMPR3922 should be saved in the external memory. Because the power (VSTANDBY-SUSPEND) to the TX3920 core and the most part of the TMPR3922 is lost in order to reduce power consumption in the suspend mode.
4. The software must then set the MEMPOWERDOWN bit in the Memory Configuration 4 Register in order to place the DRAM and / or SDRAM into self refresh mode and power down the memory interface.
5. The WARMSTART bit in the Power Control Register which shows that the TMPR3922 entered into the suspend mode should be set by software.
6. Next, the PWRCS and VCCON bits in the Power Control Register should be set low and inform the TMPR3922 that the system is powering off after making sure that the TMPR3922 enters into the DRAM self refresh mode by checking the BUS_busy bit in the TMPR3922 status register.
7. After the PWRCS is de-asserted, the VCC3 is de-asserted and the VSTANDBY_SUSPEND should be dropped down to 0V.

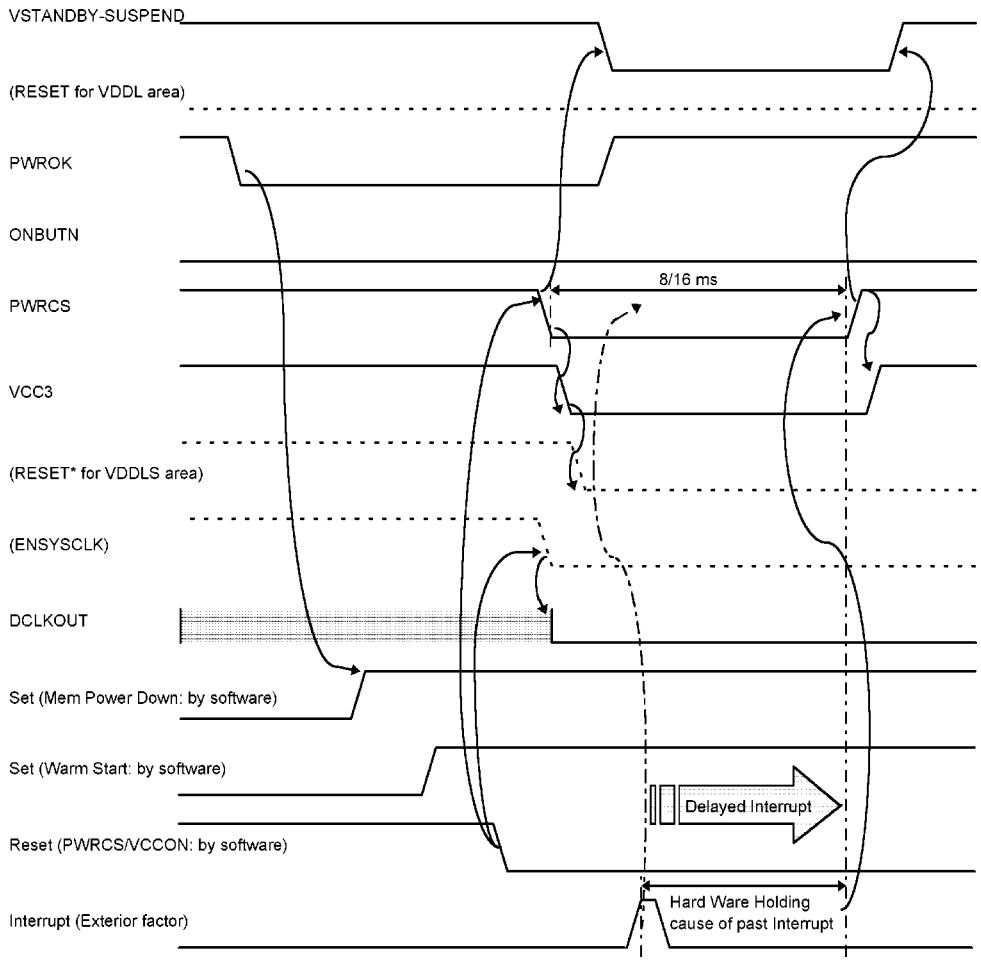


Figure 12-3 Power down

12.2.4 Wake Up Sequence

The system is waked up by following sequence.

1. The enabled interrupt for the wake-up is set.
2. After receiving the interrupt, TMPR3922 is provided 2.5V power and TX3920 processor core jumps to Reset Vector(0xBFC0_0000) and the reset exception handler makes sure which resource causes this reset by checking the WARMSTART bit in the Power Control Register.
3. If software causes this reset, the resume procedure initializes. The resume procedure includes that.
   * Setup the configuration register inside TMPR3922 as is done in the Power On Reset sequence
   * Clear the WARMSTART bit in the Power Control Register
   * Reset I/O
   * Enable the interrupt
   * Jump to the exception handler for the wake-up interrupt

12.2.5 Force Shut Down

The Power Module provides a feature called Force Shut Down that will force the system to power down if the PWROK signal goes low prior to the software clearing a bit. The timing for Force Shut Down is shown in Figure 12-4. This feature can be enabled or disabled by setting the ENFORCESHUTDWN bit in the Power Control Register. This feature is useful if a particular battery will provide a PWROK indication when there is no load on the battery, but will quickly de-assert PWROK once VCC3 is asserted.

A signal called FORCESHUTDWN is asserted whenever PWRCS is asserted as a result of the ONBUTN being pressed or an enabled interrupt being set when the device is off. If the PWROK signal goes low prior to the hardware clearing the FORCESHUTDWN bit, the Power Module will simply deassert the PWRCS signal and the System Power Supply will turn off the VCC3 power supply. The hardware clearing the FORCESHUTDWN bit by assertion of the ENSYSCLK is executed prior to bringing the memory interface out of self refresh mode.

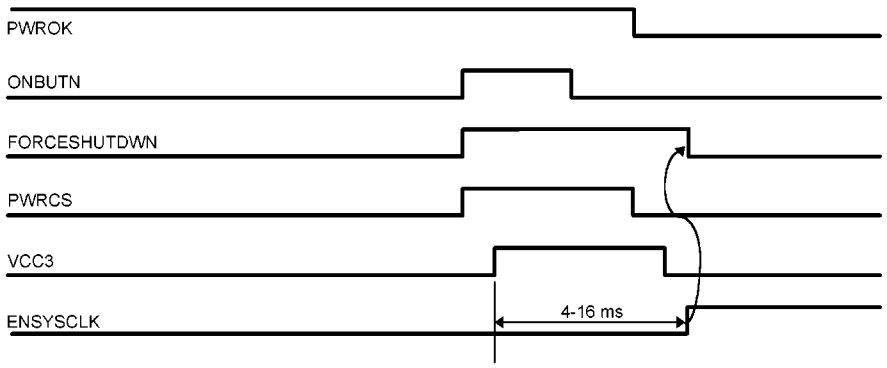


Figure 12-4 Force Shut Down

12.2.6 Software Requirement

As the Cache cell inside TX3920 processor core is not provided power, the dirty data cache line should be flushed.
As the only method for entering into the suspend mode is software, the NMI interrupt resource can be used for the PWROK transaction such as sudden power dropping down by setting the PWRONMI bit in the Power Control Register.

12.2.7 Stop Mode

In order to reduce the power consumption in the TMPR3922, it is desirable to stop the clock to the CPU when the CPU is inactive. This is possible since the CPU core is fully static and will simply re-start from where it left off when the clock is re-enabled. The software can stop the clock to the CPU by asserting the STOPCPU bit in the Power Control Register. Once this bit is set, the clock to the CPU will be stopped and remain that way until an enabled interrupt clears the STOPCPU bit. This bit should not be set when powering down the system because the ENSYSCLK signal is a more dominant clock shut down for the entire TMPR3922 chip.

12.2.8 Stop Timer

The Stop Timer is used as a coarse timer to bring the CPU out of Stop Mode. The Stop Timer consists of a 4-bit up counter and a comparator to generate the STPTIMERINT interrupt. The Stop Timer is enabled using the ENSTPTIMER control bit in the Power Control Register. When the ENSTPTIMER control bit is not set the counter will be reset to zero. Once the ENSTPTIMER bit is set the counter will count up using an 8 ms clock as the input. Once the counter reaches a value that is equal to the STPTIMERVAL[3:0] control bits, the STPTIMERINT interrupt is set. The Stop Timer will provide a maximum length of 120 ms, in steps of 8 ms.

12.2.9 Power Module Interrupts

STPTIMERINT:

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

POSPWRINT:

This interrupt is set when the PWRINT pin transitions from a logic "0" to a logic "1".

NEGPWRINT:

This interrupt is set when the PWRINT pin transitions from a logic "1" to a logic "0".

POSPWROKINT:

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

NEGPWROKINT:

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

POSONBUTNINT:

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

NEGONBUTNINT:

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

12.3 Power Registers

12.3.1 Power Control Register

OFFSET =$ 1C4:

Bit	Label	RESET	Read/Write
31	ONBUTN	—	R
30	PWRINT	—	R
29	PWROK	—	R
28	PWROKNMI	0	R/W
27	Reserved		
26	SLOWBUS	0	R/W
25-16	Reserved		
15-12	STPTIMERVAL[3:0]	X	R/W
11	ENSTPTIMER	0	R/W
10	ENFORCESHUTDWN	0	R/W
9	FORCESHUTDWN	0	R/W
8	FORCESHUTDWNOCC	0	R/W
7	SELC2MS	0	R/W
6	WARMSTART	0	R/W
5	BPDBVCC3	0	R/W
4	STOPCPU	0	R/W
3	DBNCONBUTN	0	R/W
2	COLDSTART	1	R/W
1	PWRCS	0	R/W
0	VCCON	0	R/W

ONBUTN: read-only

This bit provides the status of the ONBUTN signal.

PWRINT: read-only

This bit provides the status of the PWRINT signal.

PWROK: read-only

This bit provides the status of the PWROK signal.

PWROKNMI:

When this bit is set to 1, the TX3920 processor NMI input is asserted whenever the negedge interrupt of PWROK occurs.

STPTIMERVAL[3:0]:

When the Stop Timer is enabled the STPTIMERINT interrupt will be set when the counter is equal to the value set by these bits.

ENSTPTIMER:

This bit is used to enable the Stop Timer.

SLOWBUS:

Setting this bit will cause the internal clocks except FREECLK and XHFEE to be divided by 2.

ENFORCESHUTDWN:

The FORCESHUTDWN signal is set whenever VCCON and PWRCS are asserted by either the ONBUTN being pressed or an enabled interrupt being set. If the PWROK signal goes low prior to the processor clearing the FORCESHUTDWN signal, then VCCON and PWRCS will go low if the ENFORCESHUTDWN bit is set. If the ENFORCESHUTDWN is not set, then the FORCESHUTDWN signal will have no affect.

FORCESHUTDWN:

If the ENFORCESHUTDWN bit is set, then this bit must be cleared after the initial boot, but prior to the memory interface waking up. This can only be guaranteed if the processor executes out of cache until this bit is cleared. This is not necessary for first time Power On Reset, since the cache has not been initialized.

FORCESHUTDWNOCC:

This bit is set if a force shut down occurs due to the PWROK signal going low prior to the processor clearing the FORCESHUTDWN bit. This bit provides status to the processor to indicate that the event has occurred. The bit should be cleared once the status has been checked, so it can be used again for the next power up.

SELC2MS:

The assertion of PWRCS will cause the System Power Supply to assert the VCC3 signal. Once VCC3 is asserted, the Oscillator and PLL inside of the TMPR3922 will wake up and generate the main system clock, but for a certain period of time the value of the clock will be unstable while the Oscillator and PLL lock to the correct frequency. The Power Module will prevent the clock from driving any logic until the clock is stable. The reference for the debounce delay time is provided by the C2MS (2 ms clock) and C8MS (8 ms clock) signals generated by the RTC from the 32 kHz Oscillator. If SELC2MS is set to a logic "1", the clock will be deasserted for a minimum of 4 ms and a maximum of 6 ms after VCC3 is turned on. If SELC2MS is set to a logic "0", the clock will be deasserted for a minimum of 16 ms and a maximum of 24 ms after VCC3 is turned on.

WARMSTART:

This bit is set when the TMPR3922 enters into the suspend mode and provides status to the processor that the power supply to the processor has dropped down to 0V.

BPDBVCC3:

If this bit is set, the clock will be asserted immediately following the assertion of VCC3, instead of waiting for the delay mentioned in the SELC2MS bit description.

STOPCPU:

Setting this bit will cause the clock to the CPU core to be disabled for low power operation. The bit is cleared whenever an enabled interrupt is set. This bit should not be set when powering down the system. It is only used when the device is powered up, but the processor is not in use.

DBNCONBUTN:

If this bit is set, then the POSONBUTNINT and NEGONBUTNINT interrupts and the ONBUTN wake up function will only occur after a minimum of 16 ms and a maximum of 24 ms of debounce of the ONBUTN signal has taken place. If this bit is low, then the interrupts and power up will occur immediately upon transition of the ONBUTN signal.

COLDSTART:

This bit is set by RESET and provides status to the processor that a Power On Reset has occurred.

PWRCS:

This signal is set whenever the ONBUTN is pressed or an enabled interrupt occurs if the PWROK signal is high. The software clears this signal to latch commands into the System Power Supply. When powering down the system, this bit must be cleared simultaneously with the VCCON bit. All enabled interrupts must be cleared prior to powering down the system, or the PWRCS and VCCON signals will not go low.

VCCON:

This signal is asserted whenever the ONBUTN is pressed or an enabled interrupt occurs if the PWROK signal is high. When powering down the system, this bit must be cleared simultaneously with the PWRCS bit. All enabled interrupts must be cleared prior to powering down the system, or the PWRCS and VCCON signals will not go low.

SECTION 13 SIB Module

13.1 Overview

The SIB Module within the TMPR3922 contains holding registers, shift registers, and other logic to support interfacing to the TC35143F ASIC and / or other optional external codec devices. The definition of the SIB is compatible with the Serial Slave Mode 3 of the CS4216 codec manufactured by Crystal Semiconductor Corporation. This allows a Crystal codec to be used to provide CD-quality stereo audio instead of the slightly lower performance audio within TC35143F.

The overall sound subsystem allows playing and recording of sounds, and consists of a single-channel 12-bit codec within TC35143F, TC35143F interface circuits for direct connection to an external microphone (MIC) and speaker, and DMA support within the TMPR3922.

Similarly, the overall telecom subsystem allows support of high-performance modems (up to V.32bis data rates or possibly even faster in future versions), and consists of a single-channel 14-bit codec within TC35143F, which also includes optional echo cancellation and interface circuits for connection to an external Data Access Arrangement (DAA), as well as an auxiliary input / output port for supporting future wireless interfaces. The DAA provides the front-end interface circuitry needed between the analog telephone network and the telecom codec for wireline configurations. The front-end circuit can also include ring detect, off-hook detect, and connect detect functions.

Audio and telecom data and control / status information (such as sampling rate, gain control, muting, clip detect, etc.) is passed between the TMPR3922 and TC35143F (and / or other external codec devices) via the SIB. Within the TMPR3922, the SIB Module logic provides DMA support and control / status registers for data transfers between external system memory, the CPU core, and the SIB. SIB data transfer is always synchronous and is frame-based, with the TMPR3922 side always the master source of the clock and frame frequency and phase.

For the sound subsystem, TC35143F allows the system to digitize (using the sound codec's ADC) the MIC input, as well as to convert (using the sound codec's DAC) sampled sounds (either pre-stored, synthesized, or previously recorded) stored in system memory to an analog audio output, for routing to either a speaker and / or headphone output. For the telecom subsystem, TC35143F allows the system to digitize (using the telecom codec's ADC) the telecom analog baseband receive input, and the subsequent modem baseband signal processing can be implemented using entirely a software-based approach. TC35143F then allows the system to convert (using the telecom codec's DAC) the processed baseband transmit output to an analog telecom output for routing back out the wireline or wireless port.

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

13.1.1 Related Pins

SIBDIN: INPUT

This pin contains the input data shifted from TC35143F and / or external codec device.

SIBDOUT: OUTPUT

This pin contains the output data shifted to TC35143F and / or external codec device.

SIBSCLK: OUTPUT

This pin is the serial clock sent to TC35143F and / or external codec device. The programmable SIBSCLK rate is derived by dividing down from SIBMCLK.

SIBSYNC: OUTPUT

This pin is the frame synchronization signal sent to TC35143F and / or external codec device. This frame sync is asserted for one clock cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data.

SIBIRQ: INPUT

This pin is a general purpose input port used for the SIB interrupt source from TC35143F. This interrupt source can be configured to generate an interrupt on either a positive and / or negative edge.

SIBMCLK: INPUT/OUTPUT

This pin is the master clock source for the SIB logic. This pin is available for use in one of two modes. First, SIBMCLK can be configured as a high-rate output master clock source required by certain external codec devices. In this mode all SIB clocks are synchronously slaved to the main TMPR3922 system clock FREECLK. Conversely, SIBMCLK can be configured as an input slave clock source. In this mode, all SIB clocks are derived from an external SIBMCLK oscillator source, which is asynchronous with respect to FREECLK. Also, for this mode, SIBMCLK can still be optionally used as a high-rate master clock source required by certain external codec devices.

13.2 Interface Requirements

13.2.1 Frame Structure

Each SIB frame consists of 128 clock cycles, further divided into 2 subframes or words of 64 bits each. One word is allocated for each of up to 2 devices residing on the SIB, with TC35143F hard-coded to be the first word. The second word can support optional external audio or telecom codec devices. On such example might be the CS4216 CD-quality stereo audio codec. The SIB frame structure is shown in Figure 13-1.

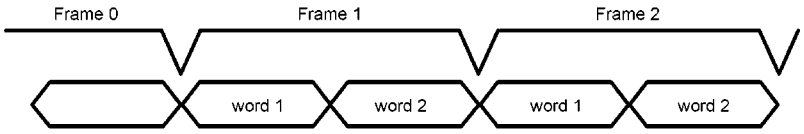


Figure 13-1 SIB Framing Structure

13.2.2 Timing Requirements

The TMPR3922 always samples receive data (SIBDIN) on the falling edge of SIBSCLK, whereas this data transmitted from TC35143F is always pushed on the rising edge of SIBSCLK. Similarly, the TMPR3922 always pushes transmit data (SIBDOUT) on the rising edge of SIBSCLK, whereas this data received by TC35143F is always sampled on the falling edge of SIBSCLK.

The SIBSYNC frame sync signal is asserted for one clock cycle immediately before each frame starts and all devices connected to the SIB monitor SIBSYNC to determine when they should transmit or receive data. SIBSYNC is always sampled by TC35143F on the falling edge of SIBSCLK. The timing relationships for the SIB clock, sync, and data signals are shown in Figure 13-2.

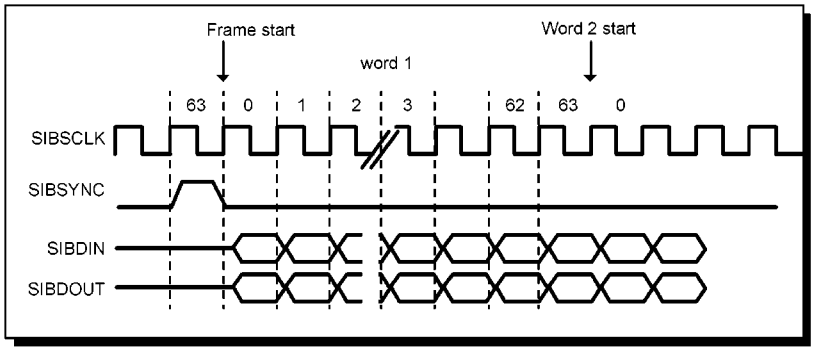


Figure 13-2 SIB Timing Relationships

13.2.3 Configurations

Many configuration options exist for the SIB. The SIB Module control bits SELSNDSF1 and SELTELSF1 are used to determine the desired SIB configuration. See Table 13-1 for a summary matrix of these possible configurations. The lowest cost configuration utilizes TC35143F only, assigned to subframe 0 of the SIB, while subframe 1 is not used. For this configuration, the system utilizes the audio and telecom codecs within TC35143F. The SIBSCLK rate is fixed (at nominally 9.216MHz) and the independent sound and telecom sample rates are controlled via TMPR3922 and TC35143F programmable counters.

A second possible configuration consists of utilizing the telecom codec within TC35143F (assigned to subframe 0) and utilizing a Crystal CS4216 CD-quality stereo audio codec (assigned to subframe 1). This higher cost configuration provides higher fidelity audio (16-bit instead of 12-bit codec), as well as stereo audio instead of mono.

Table 13-1 SIB Configuration Matrix

(Note: numbers in square brackets refer to notes/comments in the next page)

config	sound codec	telecom codec	TMPR3922 clk2x [1]	sibsclk	sibmclk out or sound OSC in [2]	SIB frame rate [3]	comments
1	TC35143F; Fs = 8k	TC35143F; Fs = 7.2k, 8k, or 9.6k	73.728M	9.216M (÷8)	18.432M out (÷4)	72k	TC35143F only; lowest cost; sibsclk = fixed; sound & telecom Fs control; via TC35143F prgm [4]; SIB subframe0 = TC35143F; SIB subframe1 = not used; SELSNDSF1 = 0; SELTELSF1 = 0;
2	CS4216; Fs = 43.2k, & submult (2% error vs. 44.1k)	TC35143F; Fs = 7.2k, 7.85k, 9.6k [5]	77.4144M	5.5296M & submult	11.0592M out (÷7) [6]	43.2k & submult	TC35143F + CS4216 (audio codec); higher cost; hi-fi and/or stereo audio; SIB subframe0 = TC35143F; SIB subframe1 = CS4216; SELSNDSF1 = 1; SELTELSF1 = 0;

NOTES :

[1] TMPR3922 freeclk is internal high-rate system-wide clock (which is one-half rate of the CPU clock rate)

[2] optional external sound OSC can be connected to TMPR3922 sibmclk (to allow decoupling of SIB clocks from freeclk ; else this pin can be configured as buffered sibmclk output which is synchronously divided down from freeclk (sibmclk allows potential support of codecs requiring a high-rate fixed master clock + a sample-rate-dependent serial clock); sibsclk serial clock is always synchronously divided down from sibmclk

[3] SIB frame rate=sibsclk ÷ 128 ; ratio for average number of SIB-frames per valid-data-frame is dependent upon sound & telecom Fs

[4] TC35143F audio / telecom Fs divider programmability: SIB-frame / valid-data-frame RATIO

(sibsclk × 2) ÷ (40 × 64)=7.2k	5
(sibsclk × 2) ÷ (36 × 64)=8k	4.5
(sibsclk × 2) ÷ (30 × 64)=9.6k	3.75

[5] TC35143F telecom Fs divider programmability :

(sibsclk × 2) ÷ (24 × 64)=7.2k		3
(sibsclk × 2) ÷ (22 × 64)=7.85k	(1.8% error vs. 8K)	2.75
(sibsclk × 2) ÷ (18 × 64)=9.6k		2.25

[6] the ÷ 7 generated clocks will have a 4-to-3 ratio duty cycle

13.2.4 Sample Rates

The SNDFSDIV and TELFSDIV control bits are used to independently configure the sampling rate for the audio and telecom codecs, respectively. These control bits set the modulus of the audio and telecom sample rate counters within the SIB Module. TC35143F also contains its own set of audio and telecom sample rate counters for its internal use and these counters must be both frequency and phase synchronized to the sample rate counters within the TMPR3922 in order to guarantee correct operation. The synchronization of these counters is triggered from the codec enable command transmitted by the TMPR3922 and received by TC35143F, as shown in Figure 13-3.

The sample rate is calculated using the following (note that the FSDIV value loaded into the counter is the desired count modulus –1):

$$Fs = (SIBSCLK \times 2) \div ((FSDIV+1) \times 64)$$

For example, if SIBSCLK=9.216 MHz and FSDIV=39 (modulo 40), then:

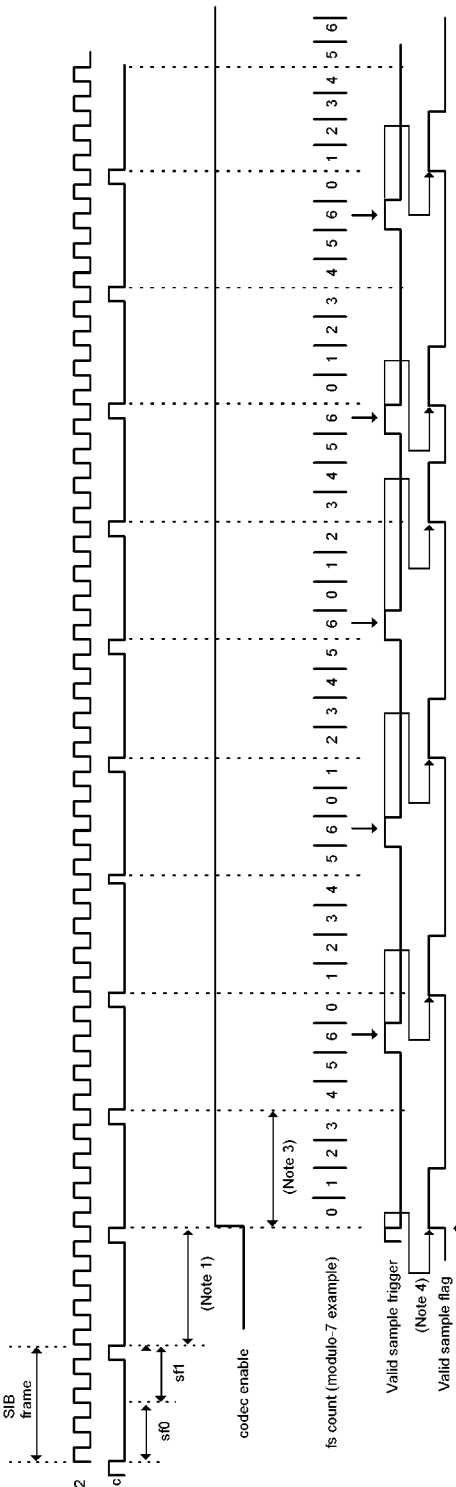
$$Fs = (9.216\ MHz \times 2) \div ((39+1) \times 64)$$
$$= 7.2\ kHz$$

For those SIB configurations using the CS4216 codec (see configuration 2 in Table 13-1), the sample rate of this codec is controlled by the SIBMCLK and SIBSCLK rates and ratio (the SIBSCLK rate is derived by dividing down from SIBMCLK). For this configuration, the CS4216 sample rate is always equal to the SIB frame rate, and lower sampling rates are obtained by lowering the SIBSCLK rate (thus also lowering the overall SIB frame rate, too). SIBMCLK and SIBSCLK are connected to CLKIN and SIBSCLK, respectively, on the CS4216. For Serial Slave Mode 3 of the CS4216, the sample rate is calculated using the following:

$$Fs = SIBMCLK \div (128 \times (SIBMCLK \div SIBSCLK))$$

For example, if SIBMCLK = 11.0592 MHz and SIBSCLK = 2.7648 MHz, then:

$$Fs = 11.0592\ MHz \div (128 \times (11.0592\ MHz \div 2.7648\ MHz))$$
$$= 21.6\ kHz$$



Note 1: during this frame, audio (and/or telecom) codec input or output enable command is transmitted by TMPR3922 and receive by TC35143F

Note 2: at start of first frame after codec is enabled, audio (and/or telecom) counters are reset; this ensures that TMPR3922 and TC35143F fs counters are immediately phase-synchronized

Note 3: during this frame, first valid DAC sample is transmitted by TMPR3922 (received by TC35143F) or first valid ADC sample is transmitted by TC35143F (received by TMPR3922); valid DAC and ADC samples are always transmitted during the same subframe (full duplex)

Note 4: audio (and/or telecom) vflag bits are asserted to TC35143F for one subframe each time a new DAC sample is being transmitted (TMPR3922) will also latch a new ADC sample from TC35143F during the same subframe); since the audio (and/or telecom) programmable sample rate is derived from sibsclk/32 (= 4x frame rate), valid samples are synchronized from the periodic fs count rollover to the next available SIB subframe

Above timing diagram shows fs count and valid sample flag generation for example configuration where frequency divisor = 6 (modulo-7 divide).

Figure 13-3 SIB Sample Rate Setup Timing

13.2.5 Enable/Disable Sequencing

As mentioned in the previous section, the synchronization of the sample rate counters within TMPR3922 and TC35143F is triggered from the codec enable command transmitted by TMPR3922 and received by TC35143F. The following steps illustrate the full sequence of events needed to enable and disable the audio and/or telecom processing paths, while ensuring synchronization of these sample rate counters (the example below shows a configuration with TC35143F for sound and telecom, using subframe 0 only):

INITIAL ENABLE SEQUENCE:

(1) SIB is initially disabled (ENSIB=0)
(2) setup SIB control registers to desired settings (SIB clock rates, sample rate dividers, data formats (8-bit vs. 16-bit, etc.), DMA settings, subframe 0 enable=ENSF0
(3) write initial data and control values to transmit holding registers (SNDTXHOLD, TELTXHOLD, SF0AUX); SF0AUX control register sets up TC35143F audio and telecom codecs, including sample rate dividers (to match TMPR3922 sample rates), gains, etc. codecs are not enabled at this point
(4) enable SIB (assert ENSIB=1)
(5) SIB then begins transmitting and receiving data; TC35143F control registers get configured with desired settings; codecs still not enabled at this point

SND (or TEL) ENABLE SEQUENCE:

(6) setup software to trigger from SIBSF0INT
(7) immediately after SIBSF0INT trigger, software must write to SF0AUX to assert Audio (or Telecom) Codec Enable, and during same subframe (i.e., before SIBSF1INT event occurs), software must assert ENSND (or ENTEL)
(8) after next SIBSF0INT and during the subframe 0 period, the Codec Enable Command is transmitted from TMPR3922 and received by TC35143F
(9) the Audio (or Telecom) Codec within TC35143F is then enabled at the start of the next frame after the Enable Command is sent, at which time the sample rate counters within both TMPR3922 and TC35143F begin counting in a synchronized fashion
(10) during this first frame after codec is enabled, the first DAC sample is transmitted by TMPR3922 (received by TC35143F) and/or the first ADC sample is transmitted by TC35143F (received by TMPR3922) ; DAC and ADC samples are always transmitted during the same subframe in a full-duplex manner (as mentioned in the previous section, DAC and ADC samples are synchronized from the periodic sample rate counter rollover to the next available SIB subframe)
(11) check valid status from TC35143F (indicates whether ADC's are settled, all turn-on delays have been met, etc.) which indicates valid samples

SND (or TEL) DISABLE SEQUENCE:

(12) software writes to SF0AUX to de-assert Audio (or Telecom) Codec Enable, in order to disable codec
(13) software then de-asserts ENSND (or ENTEL) to disable sound (or telecom) processing on TMPR3922 side

13.2.6 Data Formats

The TMPR3922's SIB Module logic contains transmit and receive holding registers which are used by the DMA circuit or CPU to write and read sound and telecom data and SIB control and status register data. The SNDTXHOLD and SNDRXHOLD registers are used for transmit and receive sound data corresponding to the appropriate subframe (0 or 1), according to the configuration as determined by the SELSNDSF1 control bit. Similarly, the TELTXHOLD and TELRXHOLD registers are used for transmit and receive telecom data corresponding to the appropriate subframe (0 or 1), according to the configuration as determined by the SELTELSF1 control bit. All of these holding registers are 32 bits wide, and the appropriate shift register fields for each subframe are written to or read from based on the programmed sound and telecom samples rates, mono versus stereo mode, and 8-bit versus 16-bit modes.

The SF0AUX and SF1AUX control holding registers and SF0STAT and SF1STAT status holding registers are also 32bits wide. Each of these control holding registers updates the control fields for the respective subframe once per frame. These transmit holding registers load the same control data into the shift register until the CPU updates the contents of the holding register. Similarly, the status holding registers are updated from the received status data in the shift register once per frame. The contents of the receive holding registers are available to be read by the CPU at any time.

The SIB data format for transfers to and from TC35143F is shown in Figure 13-4. Each word consists of fields containing audio data, telecom data, and control register address and data. During each frame, data is transferred bi-directionally (using SIBDIN and SIBDOUT) for all of these fields. For instance, the audio data field is written once every frame (from the TMPR3922 to TC35143F) via SIBDOUT and also returns audio data (from TC35143F to the TMPR3922) at the same time via SIBDIN.

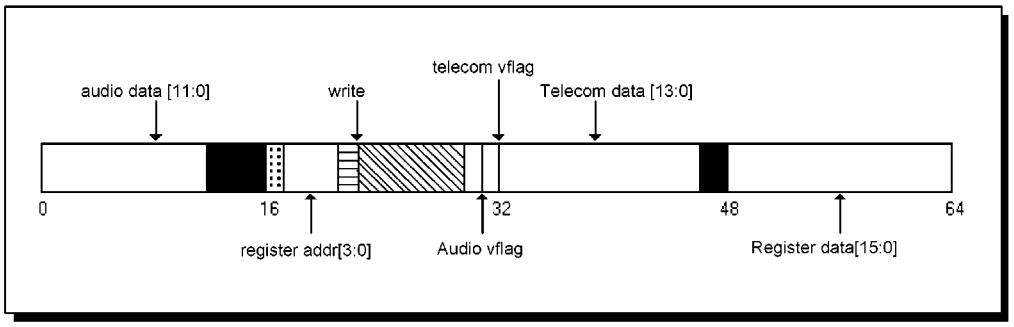


Figure 13-4 Data Format for TC35143F Word Interface

The SF0AUX control holding register and SF0STAT status holding register map directly to bits 16-31 and 48-63 of the subframe 0 word format. See Table 13-2 which shows the mapping of SF0AUX and SF0STAT bit positions to the respective TC35143F fields. For TC35143F, the audio and telecom "vflag" bits (valid flags) in the transmit direction are dynamically asserted to TC35143F for one subframe each time a new sample is being transmitted. The audio and telecom "vflag" bits in the receive direction are statically asserted from TC35143F to the TMPR3922 only after the ADC's are settled, all turn-on delays have been met, etc. The TMPR3922 then uses these valid flags to determine when the receive data from TC35143F is truly valid.

Table 13-2 SIB Subframe 0 Control/Status Bits and Fields

SF0AUX/SF0STAT bits	TC35143F word bits	field definition
- - -	0-15	audio data
31	16	reserved for register extension; must write to 0 if not used
30-27	17-20	register address [3:0]
26	21	write bit
25-18	22-29	reserved, must write to 0
17	30	audio valid flag
16	31	telecom valid flag
- - -	32-47	telecom data
15-0	48-63	register data [15:0]

The last 16 bits of the TC35143F word is made up of control register data. The exact contents and definition of this field is defined by the register address field and the "write" bit. For a read cycle ("write" bit=0), the address bits determine which TC35143F register to read and this read data is sent by TC35143F within the control register data field of SIBDIN during the same frame as the read request occurred. The separation between the address and data fields should make this possible. In addition, during a read cycle, the control register data field of SIBDOUT is ignored by TC35143F. For a write cycle ("write" bit=1), the control register data contents of SIBDOUT are written to the TC35143F register pointed to by the register address field.

The SIB data format for transfers to and from a Crystal CS4216 codec is shown in Figure 13-5. Each word consists of fields containing audio data, telecom data, and control register data. During each frame, data is transferred bi-directionally (using SIBDIN and SIBDOUT) for all of these fields. For instance, the audio data field is written once every frame (from TMPR3922 to TC35143F) via SIBDOUT and also returns audio data (from TC35143F to the TMPR3922) at the same time via SIBDIN.

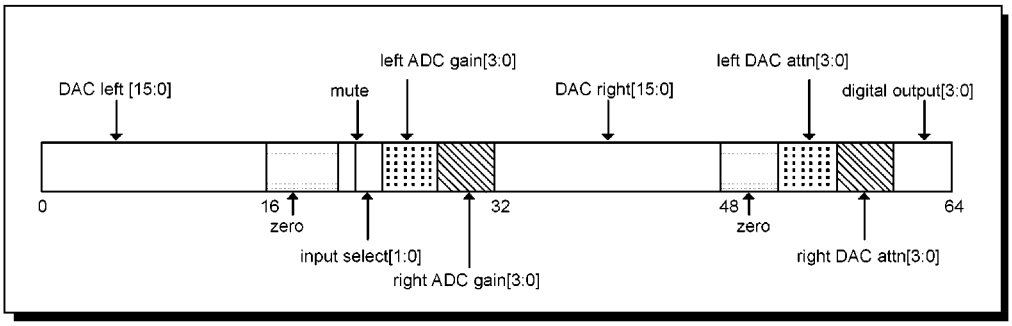


Figure 13-5a Data Format for CS4216 Input Word Interface

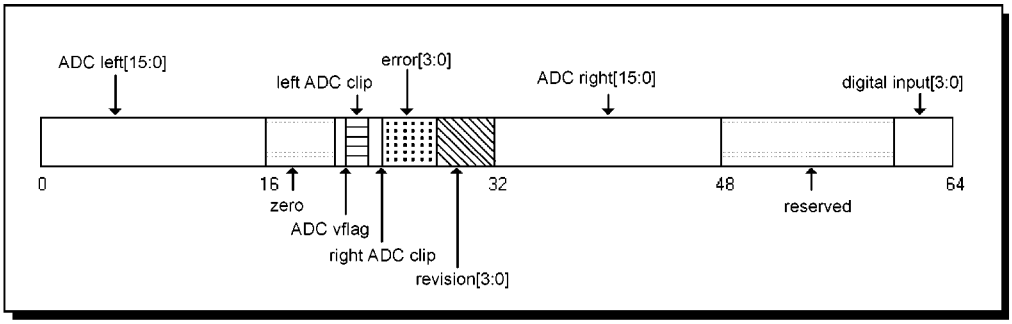


Figure 13-5b Data Format for CS4216 Output Word Interface

The SF1AUX control holding register and SF1STAT status holding register map directly to bits 16-31 and 48-63 of the subframe 1 word format. See Table 13-3 which shows the mapping of SF1AUX and SF1STAT bit positions to the respective CS4216 fields.

Table 13-3a SIB Subframe 1 Control Bits and Fields

SF1AUX bits	CS4216 word bits	field definition
- - -	0-15	DAC left audio data
31-27	16-20	zero
26	21	mute bit
25	22	left input mux select
24	23	right input mux select
23-20	24-27	left ADC gain[3:0]
19-16	28-31	right ADC gain[3:0]
- - -	32-47	DAC right audio data
15-12	48-51	zero
11-8	52-55	left DAC attn[3:0]
7-4	56-59	right DAC attn[3:0]
3-0	60-63	digital outputs[3:0]

Table 13 – 3b SIB Subframe 1 Status Bits and Fields

SF1AUX bits	CS4216 word bits	field definition
- - -	0-15	ADC left audio data
31-27	16-20	reserved (zero)
26	21	ADC valid flag
25	22	left ADC clip flag
24	23	right ADC clip flag
23-20	24-27	error[3:0]
19-16	28-31	revision[3:0]
- - -	32-47	ADC right audio data
15-4	48-59	reserved
3-0	60-63	digital inputs[3:0]

13.3 Implementation

13.3.1 Block Diagram

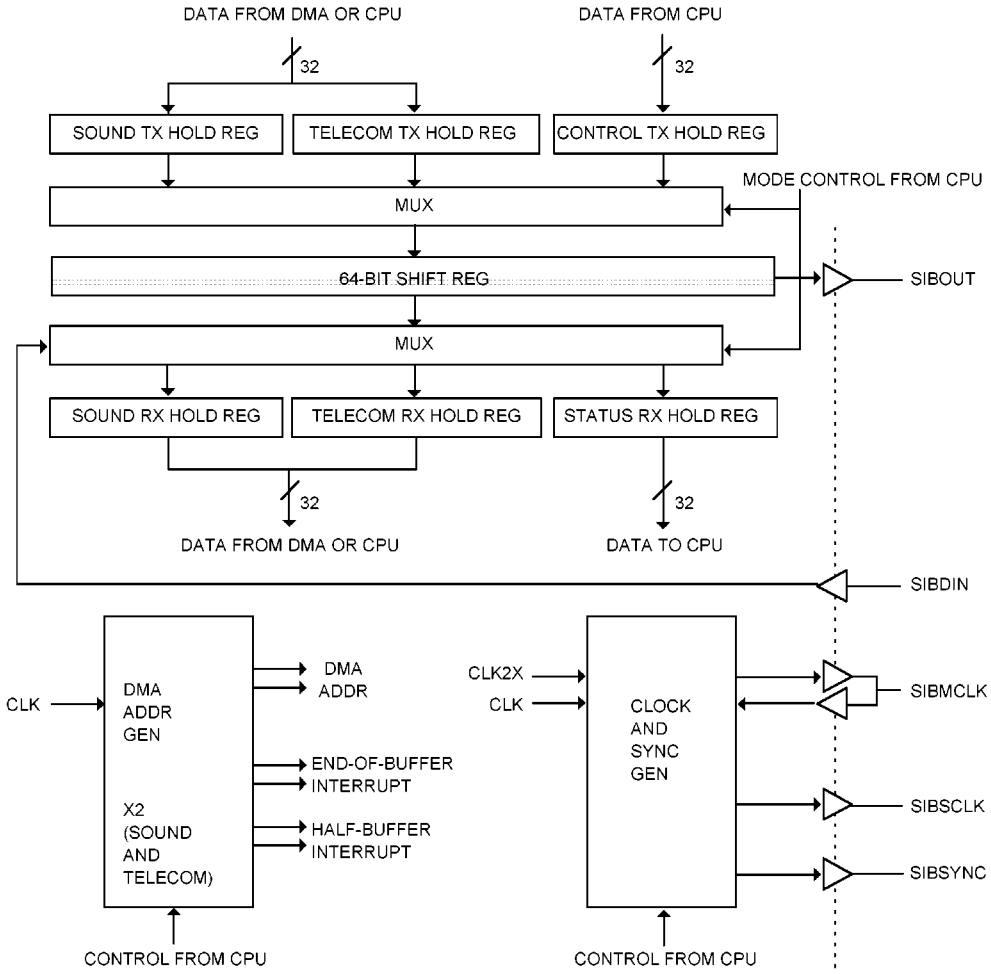


Figure 13-6 SIB Module Block Diagram

13.3.2 Holding and Shift Registers

The SIB Module logic consists of holding registers (both transmit and receive) and a serial shift register to support the transfer of sound and telecom data and control / status information between the TMPR3922 and TC35143F (and / or other external codec devices) via the SIB (see Figure 13-6 for a block diagram of the SIB Module).

For the SIB transmit direction, the sound and telecom transmit holding registers are written either from the DMA circuit or directly from the CPU, while the subframe 0 and subframe 1 control holding registers are always written directly from the CPU. These holding registers are processed by an array of multiplexers which select which fields of the various 32-bit transmit holding registers are loaded into the 64-bit shift register, such that an entire subframe is loaded in parallel and then shifted out the serial output SIBDOUT. The sequence of loading the shift register from the various holding register fields is determined by the programmed sound and telecom sample rates, mono versus stereo mode, and 8-bit versus 16-bit modes.

Conversely, for the SIB receive direction, the sound and telecom receive holding registers are read either by the DMA circuit or directly by the CPU, while the subframe 0 and subframe 1 status holding registers are always read directly by the CPU. The 64-bit shift register containing an entire subframe of received data from the serial input SIBDIN is processed by an array of multiplexers which select which fields of the various 32-bit receive holding registers are loaded from the shift register. The sequence of loading the various fields of the receive holding registers from the shift register is determined by the programmed sound and telecom sample rates, mono versus stereo mode, left versus right mono source, and 8-bit versus 16-bit modes.

13.3.3 Subframe Formats

SIB subframe 0 is nominally reserved for interfacing to TC35143F. Four possible sample-size modes are supported for sound and telecom data (these are all mono sound and telecom formats):

sound	telecom
16-bit	16-bit
16-bit	8-bit
8-bit	16-bit
8-bit	8-bit

SIB subframe 1 is nominally reserved for interfacing to a CS4216 audio codec or to a Daaphny telecom codec (for Universal Line Interface applications).

For the CS4216 audio codec, four possible sample-size modes are supported :

sound
16-bit stereo
16-bit mono
8-bit stereo
8-bit mono

Since the CS4216 can support either mono or stereo formats, the holding register format can be defined in several ways. For stereo 16-bit samples, the upper 16 bits of the holding register contains the left audio data and the lower 16 bits of the holding register contains the right audio data. For stereo 8-bit samples, the uppermost 8 bits of the holding register contains the left audio data and the middle-upper 8 bits of the holding register contains the right audio data for a given sample. Then the middle-lower 8 bits of the holding register contains the left audio data and the lowermost 8bits of the holding register contains the right audio data for the next sample. For mono samples in the transmit direction, the mono data contained in the transmit holding register is replicated for both the left and right channel output fields. For mono samples in the receive direction, based on the RMONOSNDIN control bit, either the left or right audio input sample is loaded from the shift register into the receive holding register.
For the Daaphny audio codec, two possible sample-size modes are supported (these are both mono telecom formats):

telecom
16-bit
8-bit

For the 16-bit modes, it requires 2 sample frames to read or write the 16-bit sound or telecom data samples from or to the 32-bit longword holding register. Similarly, for the 8-bit modes, it requires 4 sample frames to read or write the 8-bit sound or telecom data samples from or to the 32-bit longword holding register. See Table 13-4 for a summary matrix of all the subframe 0 and subframe 1 holding register and shift register formats.

Table 13-4a SIB Holding and Shift Register Formats (Subframe 0)

Subframe 0 Sound 16 Bit, Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
1	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]
2	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
3	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]

Subframe 0 Sound 16 Bit, Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[31:24], 8b0	sf0aux[15:0]
1	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[23:16], 8b0	sf0aux[15:0]
2	sndtxhold[31:16]	sf0aux[31:18], VS, VT	teltxhold[15:8], 8b0	sf0aux[15:0]
3	sndtxhold[15:0]	sf0aux[31:18], VS, VT	teltxhold[7:0], 8b0	sf0aux[15:0]

Subframe 0 Sound 8 Bit, Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
1	sndtxhold[23:16], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]
2	sndtxhold[15:8], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:16]	sf0aux[15:0]
3	sndtxhold[7:0], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:0]	sf0aux[15:0]

Subframe 0 Sound 8 Bit, Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf0aux[31:18], VS, VT	teltxhold[31:24], 8b0	sf0aux[15:0]
1	sndtxhold[23:16], 8b0	sf0aux[31:18], VS, VT	teltxhold[23:16], 8b0	sf0aux[15:0]
2	sndtxhold[15:8], 8b0	sf0aux[31:18], VS, VT	teltxhold[15:8], 8b0	sf0aux[15:0]
3	sndtxhold[7:0], 8b0	sf0aux[31:18], VS, VT	teltxhold[7:0], 8b0	sf0aux[15:0]

NOTE: "VS" = valid sound flag bit; "VT" = valid telecom flag bit
"8b0" = 8-bit field of zero values
"SR" = Shift Register

Table 13-4b SIB Holding and Shift Register Formats (Subframe 1 Sound)

Subframe 1 Sound 16 Bit Stereo:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
1	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
2	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
3	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]

Subframe 1 Sound 16 Bit Mono:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[31:16]	sf1aux[15:0]
1	sndtxhold[15:0]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]
2	sndtxhold[31:16]	sf1aux[31:16]	sndtxhold[31:16]	sf1aux[15:0]
3	sndtxhold[15:0]	sf1aux[31:16]	sndtxhold[15:0]	sf1aux[15:0]

Subframe 1 Sound 8 Bit Stereo:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
1	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]
2	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
3	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]

Subframe 1 Sound 8 Bit Mono:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	sndtxhold[31:24], 8b0	sf1aux[31:16]	sndtxhold[31:24], 8b0	sf1aux[15:0]
1	sndtxhold[23:16], 8b0	sf1aux[31:16]	sndtxhold[23:16], 8b0	sf1aux[15:0]
2	sndtxhold[15:8], 8b0	sf1aux[31:16]	sndtxhold[15:8], 8b0	sf1aux[15:0]
3	sndtxhold[7:0], 8b0	sf1aux[31:16]	sndtxhold[7:0], 8b0	sf1aux[15:0]

NOTE: "8b0" = 8-bit field of zero values
"SR" = Shift Register

Table 13-4c SIB Holding and Shift Register Formats (Subframe 1 Telecom)

Subframe 1 Telecom 16 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	16b0	sf1aux[31:17], VT	teltxhold[31:16]	sf1aux[15:0]
1	16b0	sf1aux[31:17], VT	teltxhold[15:8]	sf1aux[15:0]
2	16b0	sf1aux[31:17], VT	teltxhold[31:16]	sf1aux[15:0]
3	16b0	sf1aux[31:17], VT	teltxhold[15:8]	sf1aux[15:0]

Subframe 1 Telecom 8 Bit:

Frame	SR[64:49]	SR[48:33]	SR[32:17]	SR[16:1]
0	16b0	sf1aux[31:17], VT	teltxhold[31:24], 8b0	sf1aux[15:0]
1	16b0	sf1aux[31:17], VT	teltxhold[23:16], 8b0	sf1aux[15:0]
2	16b0	sf1aux[31:17], VT	teltxhold[15:8], 8b0	sf1aux[15:0]
3	16b0	sf1aux[31:17], VT	teltxhold[7:0], 8b0	sf1aux[15:0]

NOTE: "VS" = valid sound flag bit; "VT" = valid telecom flag bit
"8b0" = 8-bit field of zero values; "16b0" = 16-bit field of zero values
"SR" = Shift Register

13.3.4 Clock and Sync Generation

The SIB Module logic contains several programmable counters which are used to generate the various SIB internal and external control signals and clocks. See Figure 13-7 for a block diagram of the SIB clock and sync generation circuit. As mentioned previously, SIBMCLK can be configured as either an input or output. As an output, SIBMCLK is derived by dividing down from CLK2X. In this mode, all SIB clocks are then synchronously slaved to the main TMPR3922 system clock and only one main XTAL source is required for the entire system (besides the 32 kHz RTC XTAL), resulting in the lowest cost system configuration with regards to clock sources. As an input, SIBMCLK is generated from an external oscillator source, which is asynchronous with respect to CLK2X. This more expensive configuration allows the SIB (and also potentially the CHI and dual-UART circuits) to operate independent of the frequency used for the CPU core. This allows flexible system design options with respect to the CPU operating frequency, along with the ability to generate the required or desired UART baud rates and audio / telecom sampling frequencies.

The programmable SIBSCLK rate is derived by dividing down from SIBMCLK. Nominal rates for a TC35143F-only configuration are SIBMCLK=18.432 MHz (equals CLK2X ÷ 4) and SIBSCLK=9.216MHz (equals SIBMCLK ÷ 2). The SIBSCLK rate is then used to enable a set of counters (word counter and subframe / frame counter) to generate the SIBSYNC frame rate of 72kHz (equals SIBSCLK ÷ 128), as well as various internal SIB clocks and control signals.

The subframe rate is used to enable independent sets of programmable sound and telecom sample rate counters which generate various control signals based on the desired sample rates. For instance, this logic correctly generates the sound and telecom valid flag bits which are asserted within the SIBDOUT serial output stream every time a valid sample is being transmitted.

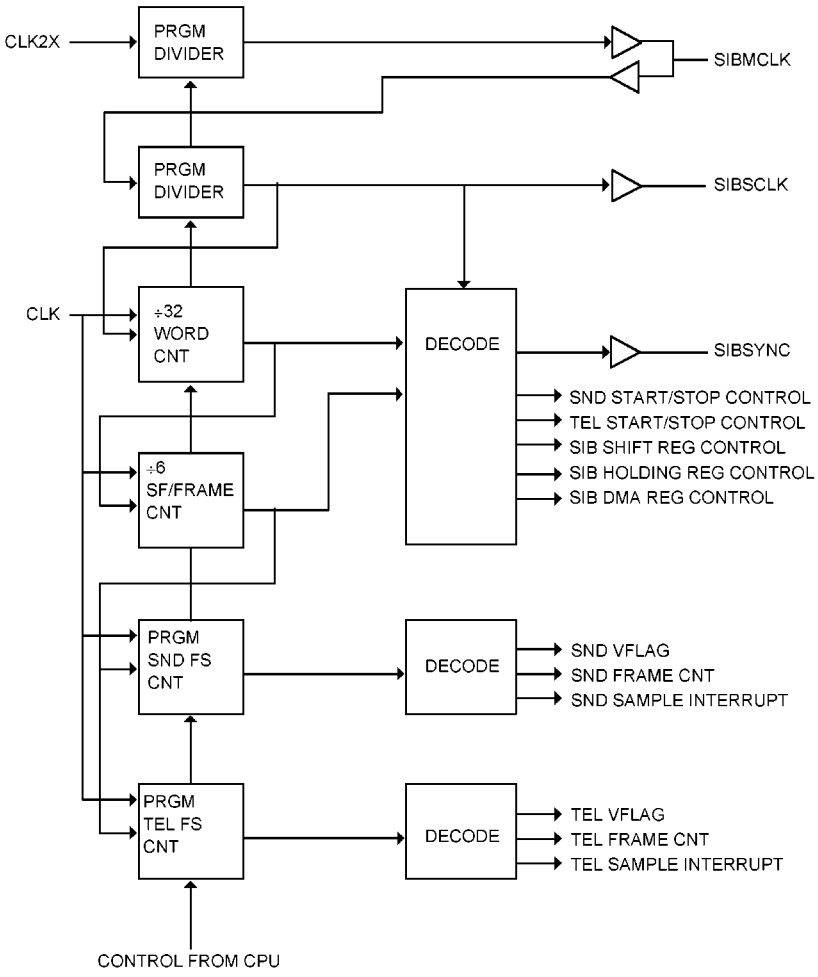


Figure 13-7 SIB Clock and Sync Generation

13.3.5 DMA Address Generation

The SIB Module provides support for four independent DMA channels: sound receive and transmit and telecom receive and transmit. Two identical circuits (one for sound DMA and one for telecom DMA) are used to generate the DMA address, as well as half-buffer and end-of-buffer interrupts (see Figure 13-8).

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

The DMA buffers can be configured in a circular buffer mode or a one-time buffer mode. For the circular mode, the DMA address is continuously incremented (each time a DMA acknowledge is received from the TMPR3922's central DMA controller) and rolls over back to the start address after the end-of-buffer is reached and will continue operating in a continuous and circular manner. For the one-time mode, the DMA logic will stop executing whenever the end-of-buffer is reached.

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

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

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

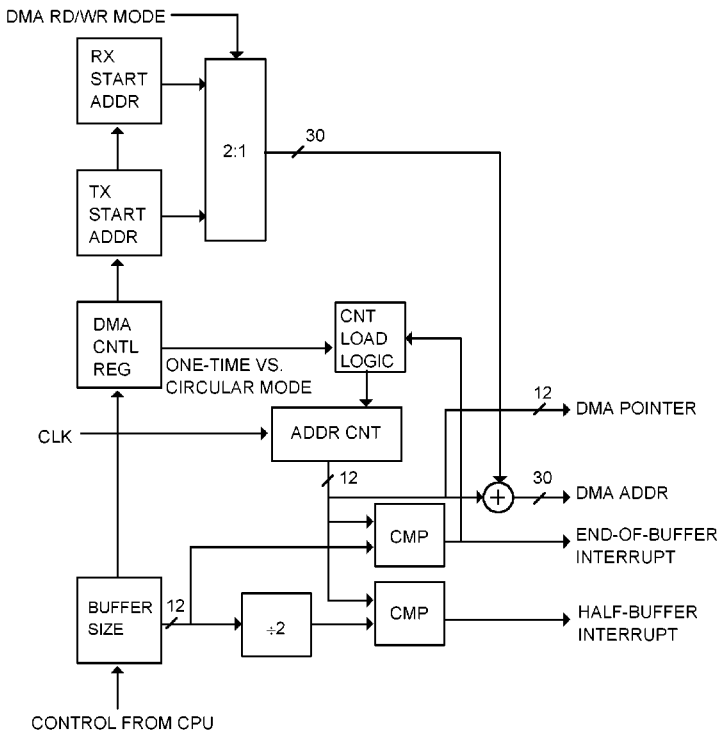


Figure 13-8 SIB DMA Address Generation

13.3.6 Related Interrupts

SND0_5INT:

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

SND1_0INT:

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

TEL0_5INT:

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

TEL1_0INT:

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

SNDDMACNTINT:

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

TELDMACNTINT:

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

LSNDCLIPINT:

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

RSNDCLIPINT:

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

VALSNDPOSINT:

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

VALSNDNEGINT:

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

VALTELPOSINT:

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

VALTELNEGINT:

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

SNDININT:

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

TELININT:

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

SIBSF0INT:

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

SIBSF1INT:

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

SIBIRQPOSINT:

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

SIBIRQNEGINT:

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

13.4 SIB Registers

13.4.1 SIB Size Register

OFFSET=$060: write-only

Bit	Label	RESET	Read/Write
31-30	Reserved		
29-18	SNDSIZE[13:2]	X	W
17-14	Reserved		
13-2	TELSIZE[13:2]	X	W
1-0	Reserved		

SNDSIZE[13:2]: write-only

These bits define the size of the sound DMA buffers (16 KBytes maximum). Both the sound RX buffer and the sound TX buffer are the same size. The last address in the sound RX DMA buffer is given by SNDRXSTART[31:2] + SNDSIZE[13:2]. The last address in the sound TX DMA buffer is given by SNDTXSTART[31:2] + SNDSIZE[13:2]. The value loaded into SNDSIZE should be equal to the desired buffer length -1.

TELSIZE[13:2]: write-only

These bits define the size of the telecom DMA buffers (16 KBytes maximum). Both the telecom RX buffer and the telecom TX buffer are the same size. The last address in the telecom RX DMA buffer is given by TELRXSTART[31:2] + TELSIZE[13:2]. The last address in the telecom TX DMA buffer is given by TELTXSTART[31:2] + TELSIZE[13:2]. The value loaded into TELSIZE should be equal to the desired buffer length -1.

13.4.2 SIB Sound RX Start Register

OFFSET=$064: write-only

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

SNDRXSTART[31:2] : write-only

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

13.4.3 SIB Sound TX Start Register

OFFSET=$068: write-only

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

SNDTXSTART[31:2] : write-only

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

13.4.4 SIB Telecom RX Start Register

OFFSET=$06C: write-only

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

TELRXSTART[31:2]: write-only

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

13.4.5 SIB Telecom TX Start Register

OFFSET=$070: write-only

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

TELTXSTART[31:2]: write-only

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

13.4.6 SIB Control Register

OFFSET=$074:

Bit	Label	RESET	Read/Write
31	SIBIRQ	—	R
30	ENCNTTEST	0	R/W
29	ENDMATEST	0	R/W
28	SNDMONO	X	R/W
27	RMONOSNDIN	X	R/W
26-24	SIBSCLKDIV[2:0]	X	R/W
23	TEL16	X	R/W
22-16	TELFSDIV[6:0]	X	R/W
15	SND16	X	R/W
14-8	SNDFSDIV[6:0]	X	R/W
7	SELTELSF1	X	R/W
6	SELSNDSF1	X	R/W
5	ENTEL	0	R/W
4	ENSND	0	R/W
3	SIBLOOP	0	R/W
2	ENSF1	0	R/W
1	ENSF0	0	R/W
0	ENSIB	0	R/W

SIBIRQ: read-only

This bit provides the logic state of the SIBIRQ input pin. The SIBIRQ pin is active-high from TC35143F.

ENCNTTEST:

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

ENDMATEST:

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

SNDMONO:

This bit is used to configure the sound input and output format for SIB subframe 1 as mono versus stereo. Setting this bit to a logic "1" selects mono mode. Clearing this bit to a logic "0" selects stereo mode.

RMONOSNDIN:

This bit is used to select left versus right channel as the mono sound source for SIB subframe 1, whenever the sound is configured as mono mode. Setting this bit to a logic "1" selects the right channel as the mono sound source. Clearing this bit to a logic "0" selects the left channel as the mono sound source.

SIBSCLKDIV[2:0]:

These bits select the start count value and the stop count value for the 4-bit programmable counter used to generate SIBSCLK, which is derived by dividing down SIBMCLK; SIBSCLK is the serial bit clock used for the SIB. Since the MSB of the counter output is used for SIBSCLK, the start count and stop count values are chosen to provide (as close as possible) a 50 % duty cycle SIBSCLK. The table used to compute these counter start and stop values are as follows:

SIBSCLKDIV	start value	stop value	divide-modulus
0	7	8	2
1	6	8	3
2	6	9	4
3	5	9	5
4	5	10	6
5	4	11	8
6	3	12	10
7	(2)	(2)	1

TEL16:

This bit is used to configure the SIB telecom input and output format as 8-bit versus 16-bit mode. Setting this bit to a logic "1" selects 16-bit mode. Clearing this bit to a logic "0" selects 8-bit mode.

TELFSDIV[6:0]:

These bits select the divider modulus for the 7-bit programmable counter used to generate the telecom sample rate clock, which is derived by dividing down an internal SIB clock of rate equal to the 4 times the SIB frame rate or 2 times the SIB subframe rate (where each frame consists of 128 serial data bits). This programmable divider consists of a down-counter which counts down from TELFSDIV to zero, so the value loaded for TELFSDIV should be the desired divider modulus -1.

For example, if SIBMCLK = 18.432 MHz and SIBSCLK = 9.216 MHz (= SIBMCLK ÷ 2), in order to generate a telecom sample rate of 7.2 kHz, the value loaded for TELFSDIV should = 39, since 9.216 MHz ÷ (40 × 32) = 7.2 kHz.

SND16:

This bit is used to configure the SIB sound input and output format as 8-bit versus 16-bit mode. Setting this bit to a logic "1" selects 16-bit mode. Clearing this bit to a logic "0" selects 8-bit mode.

SNDFSDIV[6:0]:

These bits select the divider modulus for the 7-bit programmable counter used to generate the sound sample rate clock, which is derived by dividing down an internal SIB clock of rate equal to the 4 times the SIB frame rate or 2 times the SIB subframe rate (where each frame consists of 128 serial data bits). This programmable divider consists of a down-counter which counts down from SNDFSDIV to zero, so the value loaded for SNDFSDIV should be the desired divider modulus -1.

For example, if SIBMCLK = 18.432 MHz and SIBSCLK = 9.216 MHz (= SIBMCLK ÷ 2), in order to generate a sound sample rate of 24kHz, the value loaded for SNDFSDIV should = 11, since 9.216 MHz ÷ (12 × 32) = 24 kHz.

SELTELSF1:

This bit is used to select between SIB subframe 0 and subframe 1 for the telecom data source and destination. Setting this bit to a logic "1" selects SIB subframe 1 as the telecom source. Clearing this bit to a logic "0" selects SIB subframe 0 as the telecom source.

SELSNDSF1:

This bit is used to select between SIB subframe 0 and subframe 1 for the sound data source and destination. Setting this bit to a logic "1" selects SIB subframe 1 as the sound source. Clearing this bit to a logic "0" selects SIB subframe 0 as the sound source.

ENTEL:

This bit is used to enable / disable telecom processing for the SIB module. Setting this bit to a logic "1" enables telecom processing. Clearing this bit to a logic "0" disables telecom processing. This bit should not be set until after the SIB module is setup, then ENSIB asserted.

Special timing restrictions are required whenever an enable / disable command is sent to TC35143F and / or other external codec device (see Figure 14-3). First, after SIBSF0INT (if SELTELSF1 = "0") or SIBSF1INT (if SELTELSF1 = "1") is asserted, the CPU then writes a telecom codec enable command to TC35143F via the SF0AUX (if SELTELSF1 = "0") or SF1AUX (if SELTELSF1 = "1") register. The CPU then asserts ENTEL. Both of these CPU transactions must occur within the same SIB frame period, which is before the next SIBSF0INT (if SELTELSF1 = "0") or SIBSF1INT (if SELTELSF1 = "0"). This ensures that the sample rate counters within the TMPR3922 and TC35143F are fully phase-synchronized.

ENSND:

This bit is used to enable / disable sound processing for this SIB module. Setting this bit to a logic "1" enables sound processing. Clearing this bit to a logic "0" disables sound processing. This bit should not be set until after the SIB module is setup, then ENSIB asserted.

Special timing restrictions are required whenever an enable / disable command is sent to TC35143F and / or other external codec device (see Figure 14-3). First, after SIBSF0INT (if SELSNDSF1 = "0") or SIBSF1INT (if SELSNDSF1 = "1") is asserted, the CPU then writes a sound codec enable command to TC35143F via the SF0AUX (if SELSNDSF1 = "0") or SF1AUX (if SELSNDSF1 = "1") register. The CPU then asserts ENSND. Both of these CPU transactions must occur within the same SIB frame period, which is before the next SIBSF0INT (if SELSNDSF1 = "0") or SIBSF1INT (if SELSNDSF1 = "0"). This ensures that the sample rate counters within the TMPR3922 and TC35143F are fully phase-synchronized.

SIBLOOP:

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

ENSF1:

This bit is used to enable / disable SIB subframe 1 processing. Setting this bit to a logic "1" enables SIB subframe 1. Clearing this bit to a logic "0" disables SIB subframe 1, causing the SIB serial transmitted data during this subframe to be zeroed and all received data during this subframe to not be processed by the SIB module. This bit should be set before ENSIB is asserted.

ENSF0:

This bit is used to enable / disable SIB subframe 0 processing. Setting this bit to a logic "1" enables SIB subframe 0. Clearing this bit to a logic "0" disables SIB subframe 0, causing the SIB serial transmitted data during this subframe to be zeroed and all received data during this subframe to not be processed by the SIB module. This bit should be set before ENSIB is asserted.

ENSIB:

This bit is used to enable / disable the SIB module. Setting this bit to a logic "1" enables the SIB module. Clearing this bit to a logic "0" disables the SIB module and keeps the module in a reset state.

13.4.7 SIB Sound TX Holding Register

OFFSET=$078: write-only

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

SNDTXHOLD[31:0]: write-only

These bits represent the sound data to be transmitted. Sound data can be either written directly to this register by the CPU or transparently read from the sound TX DMA buffer to this register. This register should only be loaded by the CPU after the SNDININT interrupt is asserted. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 14-4 for a summary of these formats.

13.4.8 SIB Sound RX Holding Register

OFFSET=$078: read-only

Bit	Label	RESET	Read/Write
31-0	SNDRXHOLD[31:0]	—	R

SNDRXHOLD[31:0]: read-only

These bits represent the sound data to be received. Sound data can be either read directly from this register by the CPU or transparently written to the sound RX DMA buffer from this register. This register should only be read by the CPU after the SNDININT interrupt is set. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 14-4 for a summary of these formats.

13.4.9 SIB Telecom TX Holding Register

OFFSET=$07C: write-only

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

TELTXHOLD[31:0]: write-only

These bits represent the telecom data to be transmitted. Telecom data can be either written directly to this register by the CPU or transparently read from the telecom TX DMA buffer to this register. This register should only be loaded by the CPU after the TELININT interrupt is asserted. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 14-4 for a summary of these formats.

13.4.10 SIB Telecom RX Holding Register

OFFSET=$07C: read-only

Bit	Label	RESET	Read/Write
31-0	TELRXHOLD[31:0]	—	R

TELRXHOLD[31:0]: read-only

These bits represent the telecom data to be received. Telecom data can be either read directly from this register by the CPU or transparently written to the telecom RX DMA buffer from this register. This register should only be read by the CPU after the TELININT interrupt is set. The sound and telecom data processing can be configured from among several possible formats (mono versus stereo and 8-bit versus 16-bit data formats); see Table 14-4 for a summary of these formats.

13.4.11 SIB Subframe 0 Control Register

OFFSET=$080:

Bit	Label	RESET	Read/Write
31-0	SF0AUX[31:0]	X	R/W

SF0AUX[31:0]:

These bits represent the control data to be transmitted during SIB subframe 0. This register can be loaded by the CPU asynchronously with respect to the SIB frame timing (although special timing restrictions are required whenever an enable / disable command is sent to TC35143F and / or other external codec device). Whenever this register is not updated, the previous contents are transmitted during consecutive SIB frames.

13.4.12 SIB Subframe 1 Control Register

OFFSET=$084:

Bit	Label	RESET	Read/Write
31-0	SF1AUX[31:0]	X	R/W

SF1AUX[31:0]:

These bits represent the control data to be transmitted during SIB subframe 1. This register can be loaded by the CPU asynchronously with respect to the SIB frame timing (although special timing restrictions are required whenever an enable / disable command is sent to TC35143F and / or other external codec device). Whenever this register is not updated, the previous contents are transmitted during consecutive SIB frames.

13.4.13 SIB Subframe 0 Status Register

OFFSET=$088: read-only

Bit	Label	RESET	Read/Write
31-0	SF0STAT[31:0]	—	R

SF0STAT[31:0]: read-only

These bits represent the status data to be read during SIB subframe 0. This register can be read by the CPU asynchronously with respect to the SIB frame timing. Whenever this register is not updated from TC35143F and / or other external codec device, the previous contents are transmitted during consecutive SIB frames.

13.4.14 SIB Subframe 1 Status Register

OFFSET=$08C: read-only

Bit	Label	RESET	Read/Write
31-0	SF1STAT[31:0]	—	R

SF1STAT[31:0]: read-only

These bits represent the status data to be read during SIB subframe 1. This register can be read by the CPU asynchronously with respect to the SIB frame timing. Whenever this register is not updated from TC35143F and / or other external codec device, the previous contents are transmitted during consecutive SIB frames.

13.4.15 SIB DMA Control Register

OFFSET=$090:

Bit	Label	RESET	Read/Write
31	SNDBUFF1TIME	0	R/W
30	SNDDMALOOP	0	R/W
29-18	SNDDMAPTR[13:2]	—	R
17	ENDMARXSND	0	R/W
16	ENDMATXSND	0	R/W
15	TELBUFF1TIME	0	R/W
14	TELDMALOOP	0	R/W
13-2	TELDMALOOP[13:2]	—	R
1	ENDMARXTEL	0	R/W
0	ENDMATXTEL	0	R/W

SNDBUFF1TIME:

The sound DMA controller supports two buffer addressing modes depending on the state of this bit. When SNDBUFF1TIME is set to a logic "1", the sound DMA controller will stop executing when it reaches the end of the DMA buffer. When SNDBUFF1TIME is cleared to a logic "0", the sound DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner.

SNDDMALOOP:

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

SNDDMAPTR[13:2]: read-only

These bits provide the status of the sound DMA counter.

ENDMARXSND:

This bit enables the sound DMA receive function. Setting this bit to a logic "1" enables the DMA mode. Clearing this bit to a logic "0" disables the DMA mode. This bit should not be set until the SNDRXSTART, SNDTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted). Either ENDMARXSND or ENDMATXSND or both can be set at a time since the sound DMA controller can support full duplex operation.

ENDMATXSND:

This bit enables the sound DMA transmit function. Setting this bit to a logic "1" enables the DMA mode. Clearing this bit to a logic "0" disables the DMA mode. This bit should not be set until the SNDRXSTART, SNDTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted). Either ENDMARXSND or ENDMATXSND or both can be set at a time since the sound DMA controller can support full duplex operation.

TELBUFF1TIME:

The telecom DMA controller supports two buffer addressing modes depending on the state of this bit. When TELBUFF1TIME is set to a logic "1", the telecom DMA controller will stop executing when it reaches the end of the DMA buffer. When TELBUFF1TIME is cleared to a logic "0", the telecom DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating in a continuous and circular manner.

TELDMALOOP:

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

TELDMAPTR[13:2]: read-only

These bits provide the status of the telecom DMA counter.

ENDMARXTEL:

This bit enables the telecom DMA receive function. Setting this bit to a logic "1" enables the DMA mode. Clearing this bit to a logic "0" disables the DMA mode. This bit should not be set until the TELRXSTART, TELTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted). Either ENDMARXTEL or ENDMATXTEL or both can be set at a time since the telecom DMA controller can support full duplex operation.

ENDMATXTEL:

This bit enables the telecom DMA transmit function. Setting this bit to a logic "1" enables the DMA mode. Clearing this bit to a logic "0" disables the DMA mode. This bit should not be set until the TELRXSTART, TELTXSTART, and SIBSIZE registers are setup and the SIB module is enabled (ENSIB asserted). Either ENDMARXTEL or ENDMATXTEL or both can be set at a time since the telecom DMA controller can support full duplex operation.

SECTION 14 SPI Module

14.1 Overview

The SPI is a serial interface consisting of clock, data out, and data in. The SPI is used to interface to devices such as serial power supplies, serial A/D converters, and other devices that contain simple serial clock and data interfaces. The TMPR3922 is always the master and generates the clock. Multiple slave devices can share the SPI by using a unique chip select for each slave device. The chip select can be generated using one of the general purpose I/O ports on the TMPR3922 or TC35143F, or using some other output port available in the system. When a device is selected by asserting it's chip select, the device will shift data in using the SPICLK and SPIOUT signals and the device will shift data out using the SPIIN signal. When a device is not selected then the data output connected to SPIIN must be tri-stated so other devices can share the SPIIN signal. The SPI Module contains registers which provide programmability for SPICLK rate, MSB first versus LSB first, clock polarity, data phase polarity, and byte mode versus word mode operation.

14.1.1 Related Pins

SPICLK: INPUT/OUTPUT

This pin is used to clock data in and out of the slave device. This pin is the master clock source for the SPI logic. This pin is available for use in one of two modes. First, SPICLK can be configured as a output master clock source required by certain external devices. In this mode all SPI clocks are synchronously slaved to the main TMPR3922 system clock XHFEE. Conversely, SPICLK can be configured as an input slave clock source. In this mode, all SPI clocks are derived from an external oscillator source, which is asynchronous with respect to XHFREE. .

SPIOUT: OUTPUT

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

SPIIN: INPUT

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

14.2 Description

14.2.1 Block Diagram

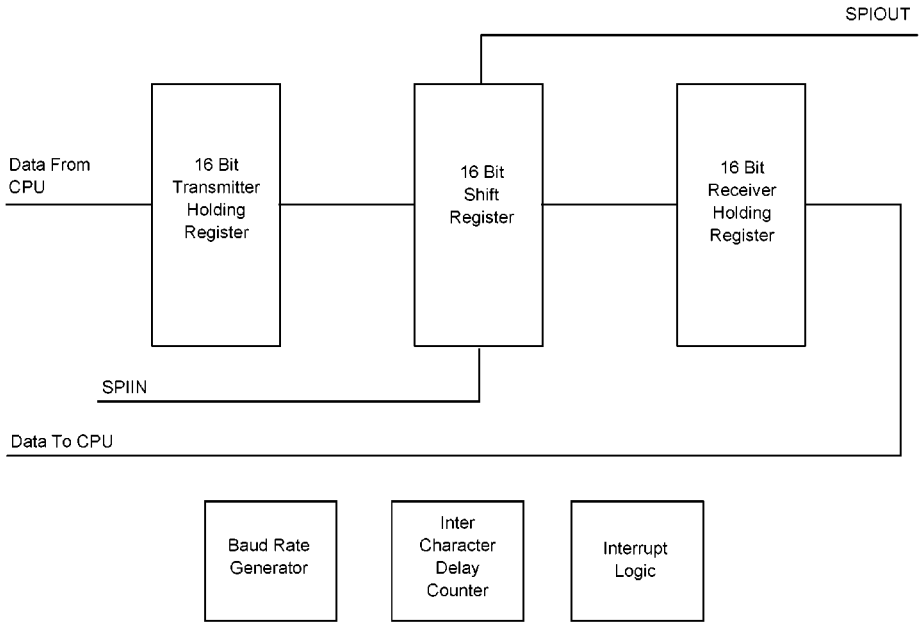


Figure 14-1 SPI Block Diagram

The SPI Module primarily consists of a 16-bit Shift Register, a 16-bit Transmitter Holding Register, a 16-bit Receiver Holding Register, a Baud Rate Generator, an Inter Character Delay Counter, and Interrupt Logic. A block diagram of the SPI Module is shown in Figure 14-1.

14.2.2 Baud Rate Generator

The rate of the SPICLK signal is determined by the value of the BAUDRATE[3:0] bits in the SPI Control Register. The BAUDRATE[3:0] bits are used by the Baud Rate Generator to divide the Master SPICLK generated by the Clock Module logic. The Master SPICLK is typically set to 7.3728 MHz when the main system clock is 73.728 MHz. The BAUDRATE[3:0] bits allows the SPICLK to vary from 3.6864 MHz down to 230 kHz.

14.2.3 Transmitter / Receiver

The SPI Module is kept in a reset state until the ENSPI bit is set in the SPI Control Register. Before setting the ENSPI bit, all other control bits in the SPI Control Register should be set to the desired values. Once the ENSPI bit is set, the SPIBUFAVAILINT interrupt will be asserted to indicate that the SPI Transmitter Holding Register is available. The SPI logic will then wait until the software writes to the SPI Transmitter Holding Register.

Once the software writes to the Transmitter Holding Register then the contents will be transferred to the Shift Register and shifted out to the slave device. While data is shifting out to the slave device using the SPIOUT signal, data will shift in using the SPIIN signal. Once the data has finished shifting, the contents of the Shift Register will be loaded into the Receiver Holding Register and the SPIRCVINT Interrupt will be asserted to indicate that there is valid receive data in the Receiver Holding Register. Once the contents of the Transmitter Holding Register are transferred to the Shift Register, the SPIBUFAVAILINT interrupt is again asserted to indicate that the Transmitter Holding register is once again available. Thus, as long as the software can keep the Transmitter Holding Register serviced before the data shifts out of the Shift Register, the SPI can maintain seamless data transfer. If the software fails to keep up with the transfer rate, then the SPI will simply wait until the next data is written to the Transmitter Holding Register.

The SPI supports either 8-bit per character or 16-bit per character operation, as defined by the WORD bit in the SPI Control Register. The software can also select whether the MSB or LSB should shift first using the LSB control bit in the SPI Control Register. Another set of control bits (CLKPOL and PHAPOL) select the polarity of the SPICLK and determine whether data should be sampled on the rising or falling edge of the SPICLK. SPI timing is shown in Figure 14-2.

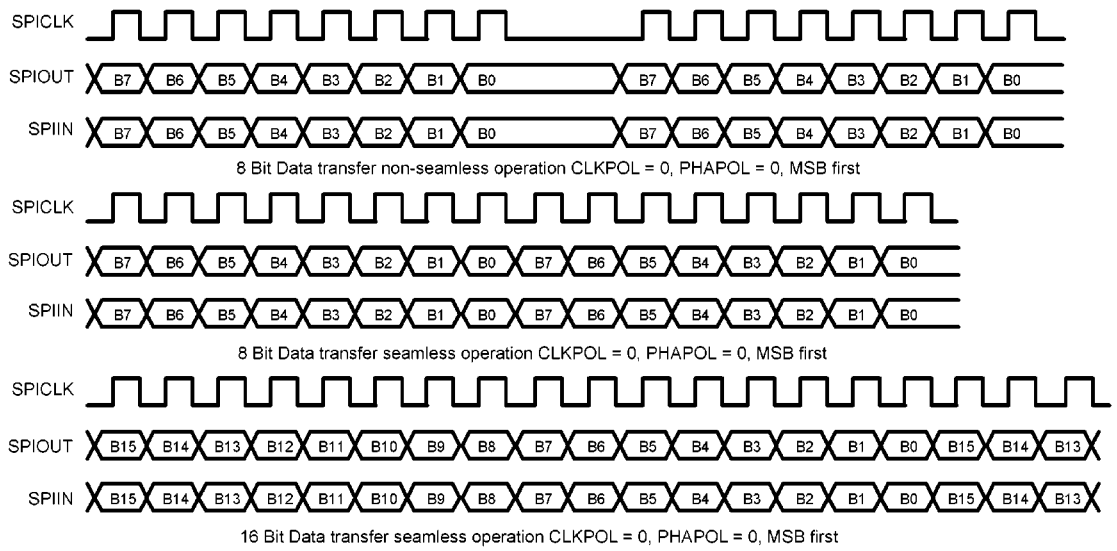


Figure 14-2 SPI Timing

14.2.4 CLKPOL / PHAPOL

The CLKPOL and PHAPOL bits in the SPI Control Register determine the idle phase of SPICLK and the valid clock edge for sampling data. Figure 14-3 shows the four possible combinations.

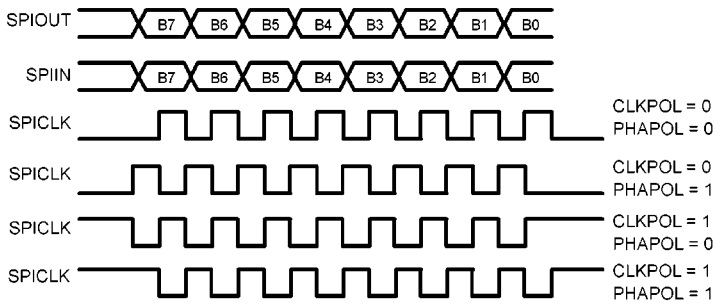


Figure 14-3 CLKPOL/PHAPOL Timing

14.2.5 Inter Character Delay Counter

Sometimes it is desirable to guarantee a minimum time between groups of data. The Inter Character Delay Counter is used to provide delay between groups of data. If WORD mode is selected in the SPI Control Register, delay will be inserted after 16 bits of data are shifted. If WORD mode is not selected, delay will be inserted after 8 bits of data are shifted, as shown in Figure 14-4. Inter character delay is added by setting the DELAYVAL[3:0] bits to a value other than $0. The number stored in these bits will directly correspond to the number of SPICLK periods of delay that will be inserted between characters. A zero value for these bits will imply seamless operation and the SPI will shift data and provide clocks continuously as long as the software keeps up with the transmitter rate.

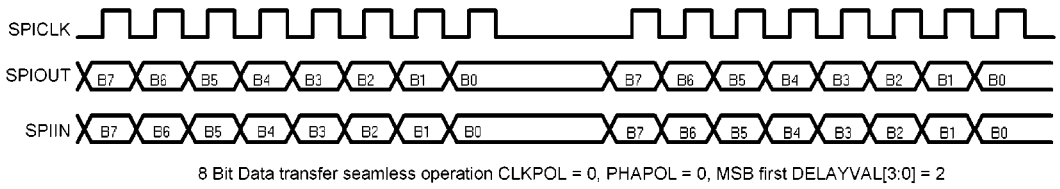


Figure 14-4 Inter Character Delay Timing

14.2.6 SPI Interrupts

SPIBUFAVAILINT:

This interrupt is set when the ENSPI bit is first asserted and subsequently when the contents of the SPI Transmitter Holding Register are transferred to the SPI Shift Register. This interrupt is used to indicate that the SPI Transmitter Holding Register is available to be written by the software. See Section 8 for information on how to read and clear interrupts.

SPIERRINT:

This interrupt is set whenever the SPI Transmitter Holding Register is written, but the SPIBUFAVAILINT has not set to indicate that the register is available. This interrupt serves as a overrun indication for the software. See Section 8 for information on how to read and clear interrupts.

SPIRCVINT:

This interrupt is whenever the contents of the SPI Shift Register are transferred to the SPI Receiver Holding Register. This interrupt is used to indicate that there is valid data in the SPI Receiver Holding Register to be read by the software. See Section 8 for information on how to read and clear interrupts.

SPIEMPTYINT:

This interrupt is set whenever the both the SPI Shift Register and the SPI Transmitter Holding Register are empty. This interrupt can be used by the software to determine when the SPI is idle. See Section 8 for information on how to read and clear interrupts.

14.3 SPI Registers

14.3.1 SPI Control Register

OFFSET=$160:

Bit	Label	RESET	Read/Write
31-18	Reserved		
17	SPION	0	R
16	EMPTY	1	R
15-12	DELAYVAL[3:0]	X	R/W
11-8	BAUDRATE[3:0]	X	R/W
7-6	Reserved		
5	PHAPOL	0	R/W
4	CLKPOL	0	R/W
3	Reserved		
2	WORD	0	R/W
1	LSB	0	R/W
0	ENSPI	0	R/W

SPION: read-only

When the ENSPI bit is disabled, the SPI will not shut down until the Transmitter Holding Register and Shift Register are both empty, in order to make sure that any data still in the SPI is shifted out. This status bit allows the software to know when the module has shut down as a result of clearing the ENSPI control bit.

EMPTY: read-only

This bit is asserted if both the Transmitter Holding Register and Shift Register are empty and the Inter Character Delay Counter is finished inserting delay. This status bit allows the software to know when the SPI is idle.

DELAYVAL[3:0]:

These bits define the number SPICLK periods of delay to insert between 16-bit or 8-bit characters, depending on whether WORD mode is selected or not. The value of these bits directly corresponds to the number of SPICLK periods of delay. Thus, a value of $0 will provide seamless operation with no inter character delay.

BAUDRATE[3:0]:

These bits define the rate of the SPICLK. These bits divide the Master SPICLK of 7.3728 MHz to generate the desired SPICLK rate, as given by the following equation:

$$\text{SPICLK Rate} = \frac{7.3728\ \text{MHz}}{\text{BAUDRATE[3:0]} * 2 + 2}$$

PHAPOL:

Setting this bit will cause the transmitter to clock data out on the rising edge of SPICLK, to be sampled by the peripherals on the falling edge of SPICLK.

CLKPOL:

Setting this bit will cause the SPICLK signal to idle high instead of low.

WORD:

Setting this bit will cause the SPI to shift 16bits of data in and out. If this bit is cleared, 8bits of data are shifted.

LSB:

Setting this bit will cause the data to be shifted out to the slave devices and in from the slave devices LSB-first instead of MSB-first.

ENSPI:

Setting this bit will enable the SPI Module. This bit should only be set after all other control bits have been set to the desired values. When this bit is cleared, the SPI will remain active until the Shift Register and Transmitter Holding Register are both empty. Once both registers are empty, the SPI will shut down and all the logic will be held in a reset state.

14.3.2 SPI Transmitter Holding Register

OFFSET=$164: write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	TXDATA[15:0]	X	W

TXDATA[15:0]: write-only

These bits are the data that is loaded into the Transmitter Holding Register. This register should only be loaded after the SPIBUFAVAILINT interrupt is asserted. If WORD mode is not set, only bits 7:0 are valid.

14.3.3 SPI Receiver Holding Register

OFFSET=$164: read-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	RXDATA[15:0]	X	R

RXDATA[15:0]: read-only

These bits are the receive data in the Receiver Holding Register. The bits are only valid after the SPIBUFAVAILINT interrupt is asserted. If WORD mode is not set, only bits 7:0 are valid.

SECTION 15 Timer Module

15.1 Overview

The Timer Module consists of two portions. The first is a 43-bit Real Time Clock (RTC) counter that uses a 32.768 kHz clock. The counter will provide a maximum count of 388 days. Also included is a 40-bit alarm register for the RTC that allows the software to set an alarm at any desired count of the RTC counter. The RTC will generate two interrupts for the CPU. The first is the ALARMINT that will generate an interrupt whenever the RTC reaches the value set by the alarm. The second is the RTCINT that will generate an interrupt whenever the RTC counter "rolls over" after reaching a count of 3104 days.

The second portion of the Timer Module is the Periodic Timer used by the software to generate periodic interrupts for monitoring system events. The Periodic Timer uses the TIMERCLK generated by the Clock Module, which is normally set to 1.15 MHz. The Periodic Timer contains a programmable 16-bit counter. When enabled, the counter will count down and generate an interrupt (PERINT) whenever reaching a count of zero.

15.1.1 Related Pins

C32KIN: INPUT

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

C32KOUT: OUTPUT

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

BC32K: OUTPUT

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

15.2 RTC

15.2.1 RTC Block Diagram

The RTC consists of five 8-bit ripple counters, a 43-bit equality comparator, roll over detect logic and two interrupt flip-flops, as shown in Figure 15-1.

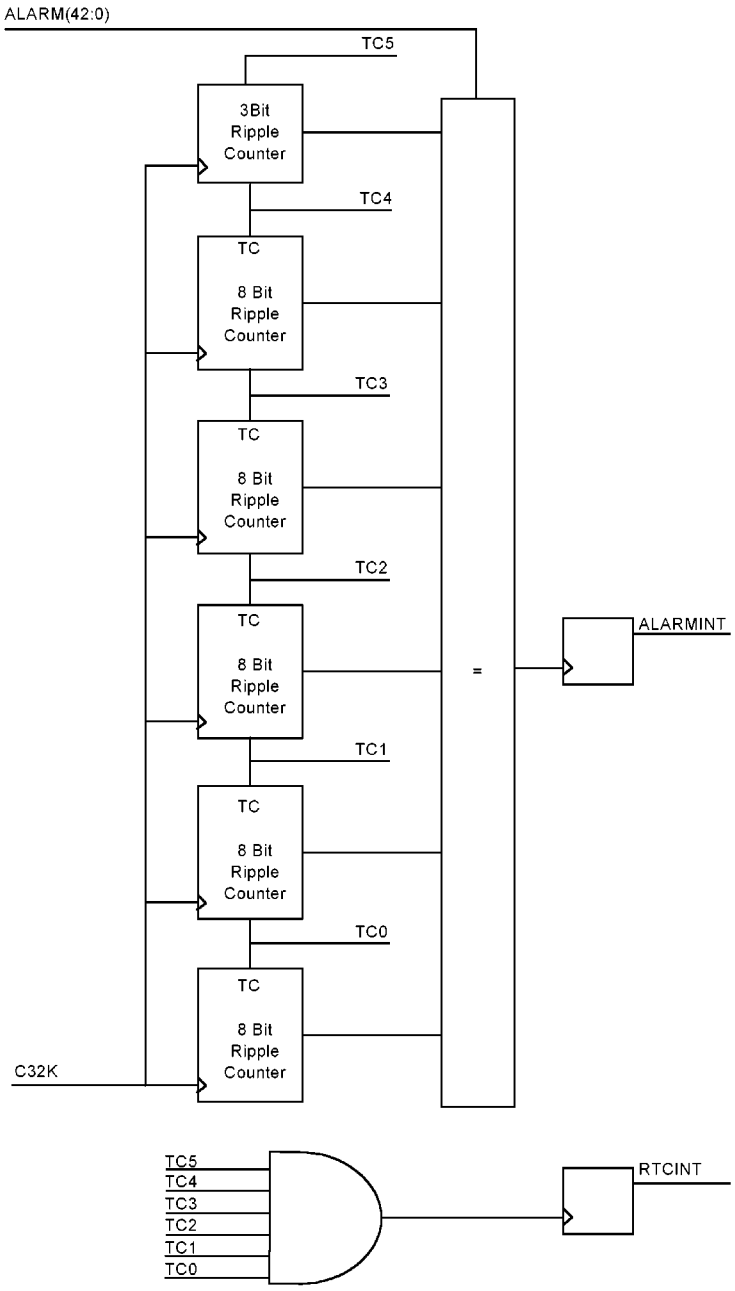


Figure 15-1 RTC Block Diagram

15.2.2 RTC Description

The RTC contains five 8-bit ripple counters connected in series. The first counter counts on each C32K clock, while each successive counter only counts when the previous count stage has reached a count of "$FF". Once all the counters reach a count of "$7FFFFFFFFF", the RTCINT interrupt will assert to indicate that the counter is "rolling over". Given a 43-bit counter for the RTC and an input clock C32K of 32.768kHz, the time until the RTCINT interrupt will assert is 3104 days.

The software can generate an alarm interrupt (ALARMINT) by setting the ALARM[42:0] bits in the Alarm Register. Whenever the RTC becomes equal to the value set in the Alarm Register, the ALARMINT will be triggered. The value of the RTC counter can be read via the RTC Register. The RTC counter is split into groups of five 8-bit counters to simplify IC testing.

15.2.3 RTC Interrupts

RTCINT:

This interrupt is set whenever all 43bits of the RTC counter reach a value of "$7FFFFFFFFF" to alert the software that the counter is "rolling over". See Section 8 for information on how to read and clear interrupts.

ALARMINT:

This interrupt is set whenever the RTC counter reaches a count that is equal to the value of the ALARM[42:0] bits set in the Alarm Register. See Section 8 for information on how to read and clear interrupts.

15.3 Periodic Timer

15.3.1 Periodic Timer Block Diagram

The Periodic Timer consists of a 16-bit down counter, along with a zero detect and an interrupt flip-flop, as shown in Figure 15-2.

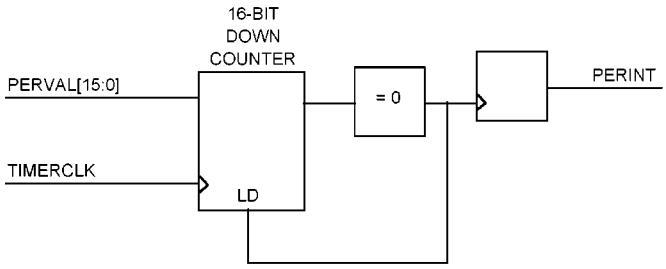


Figure 15-2 Periodic Timer Block Diagram

15.3.2 Periodic Timer Description

The Periodic Timer is used by the software to generate periodic interrupts needed to monitor system events. The Periodic Timer contains a 16-bit down counter whose initial value is programmed by the software via the PERVAL[15:0] control bits in the Periodic Timer Register. The counter uses the TIMERCLK signal generated by the Clock Module as the clock for the counter. With a system clock of 36.864 MHz, the TIMERCLK will normally be set to 1.15 MHz. This implies a maximum Periodic Timer count duration of 56.89 ms with a granularity of 0.87 μs. The Periodic Timer is enabled by asserting the ENPERTIMER control bit in the Timer Control Register. Once enabled, the counter will load the PERVAL[15:0] control bits and down count to zero. Once the count of zero is reached, the PERINT interrupt is asserted and the counter re-loads the PERVAL[15:0] control bits. The value of the Periodic Timer counter can be read by the software via the PERCNT[15:0] status bits in the Periodic Timer Register.

15.3.3 Periodic Timer Interrupts

PERINT:

This interrupt is set whenever the Periodic Timer is enabled and the Periodic Timer counter reaches a count of zero. See Section 8 for information on how to read and clear interrupts.

15.4 Timer Registers

15.4.1 RTC Register

OFFSET=$140: read-only

Bit	Label	RESET	Read/Write
31-11	Reserved		
10-0	RTC[42:32]	X	R

OFFSET=$144: read-only

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

RTC[42:0]: read-only

These bits provide the status of the 43-bit RTC counter. The software must read these bits twice and compare the values to ensure that the counter is not read while the counter is counting since the CPU clock is not synchronous with the RTC counter clock. If the two reads do not compare, the software must read the register again to read the correct count value.

15.4.2 Alarm Register

OFFSET=$148:

Bit	Label	RESET	Read/Write
31-11	Reserved		
10-0	ARARM[42:32]	X	R/W

OFFSET=$14C:

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

ALARM[42:0]:

Whenever the RTC counter reaches a count that is equal to these bits, the ALARMINT interrupt will be set.

15.4.3 Timer Control Register

OFFSET=$150:

Bit	Label	RESET	Read/Write
31-8	Reserved		
7	FREEZEPRE	0	R/W
6	FREEZERTC	0	R/W
5	FREEZETIMER	0	R/W
4	ENPERTIMER	0	R/W
3	RTCCLR	0	R/W
2	TESTC8MS	0	R/W
1	ENTESTCLK	0	R/W
0	ENRTCTST	0	R/W

FREEZEPRE:

Setting this bit will cause the lower 8 bits of the RTC counter to freeze. The lower 8 bits of the RTC counter are also used to generate an 8 ms reference signal for the IR Carrier Detect State Machine and the debouncers in the Power Module and IO Module. Thus, setting this bit will also cause these modules to lose their 8 ms reference signal. This bit can be used by the software to freeze the RTC counter during software debugging.

FREEZERTC:

Setting this bit will cause the upper 32 bits of the RTC counter to freeze. This bit can be used by the software to freeze the RTC counter during software debugging.

FREEZETIMER:

Setting this bit will cause the Periodic Timer counter to freeze. This bit can be used by the software to freeze the Periodic Timer counter during software debugging.

ENPERTIMER:

Setting this bit will cause the Periodic Timer to load the PERVAL[15:0] control bits and begin down counting. The PERINT interrupt will only be asserted if this bit is enabled.

RTCCLR:

Setting this bit to a logic "1" will cause all 40 bits of the RTC counter to initialize to "$0000000000". The RTC counter will stay cleared and the counter will not start counting until this bit is cleared back to a logic "0".

TESTC8MS:

Setting this bit will cause the 8 ms reference pulse used by the IR logic and the debouncer circuits to run at 61μs instead of 8 ms. This bit is used for IC testing to speed up the counters and should normally never be set by the software.

ENTESTCLK:

Setting this bit will cause the 32 kHz input clock for the RTC counter to be driven by the TIMERCLK instead of the 32 kHz input pin. This bit is used for IC testing to speed up the amount of time needed to test the RTC counter and should normally never be set by the software.

ENRTCTST:

Setting this bit will cause all five of the 8-bit counters that comprise the RTC counter to count together. This provides a mechanism for fully exercising all the counters in a timely fashion for IC testing. The software may wish to set this bit in order to test the RTCINT interrupt since normally it would take 388 days to generate this interrupt.

15.4.4 Periodic Timer Register

OFFSET=$154:

Bit	Label	RESET	Read/Write
31-16	PERCNT[15:0]	X	R
15-0	PERVAL[15:0]	X	R/W

PERCNT[15:0]: read-only

These bits provide the status of the Periodic Counter. The software should continually read this register until two consecutive reads provide the same value, in order to ensure that the register is not read while the counter is counting, since the TIMERCLK is not synchronous with the CPU clock.

PERVAL[15:0]:

These bits are loaded into the Periodic Timer counter when the counter is enabled or when the counter reaches a count of zero. The TIMERCLK is normally set to 1.15 MHz, which implies the following equation for the time between the PERINT interrupts:

$$\text{Interrupt Rate} = \frac{\text{PERVAL[15:0]} + 1}{1.15\ \text{MHz}}$$

SECTION 16 UART Module

16.1 Overview

The UART Module contains two identical, fully independent, full duplex UARTs: UARTA and UARTB. Each UART contains the following features:

- adjustable baud rate counter from 230 kHz down to 225 Hz
- single buffered transmit register
- double buffered receive register
- DMA for either transmit or receive
- full duplex
- even, odd or no parity
- 7 or 8 bits per character
- one or two stop bits per character
- pulse option mode to support IRDA Infrared protocol

16.1.1 Related Pins

TXD: OUTPUT

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

RXD: INPUT

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

IROUT: OUTPUT

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

IRIN: INPUT

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

16.2 Overall Operation

The operation of each UART is controlled through six registers: Control 1 Register, Control 2 Register, DMA Control 1 Register, DMA Control 2 Register, DMA Count Register, and Transmit/Receive Holding Register.

16.2.1 Power On/Off

Each UART is powered on and off with the ENUART bit of the Control 1 Register. When the ENUART bit is cleared, the UART will power off only after all data in its transmit shift and holding registers has been clocked out. The on/off status of the UART can be monitored via the UARTON bit of the Control 1 Register. While powered off, the UART is completely disabled.

In addition to being powered on, each UART must also have its clock enabled to operate. The UART clock enables, as well as the UART clock frequency, is controlled by the Clock Module (see Section 6).

16.2.2 Baud Rate and Communication Parameters

The baud rate, bits per symbol (7 or 8), parity type (none, even, or odd), and line polarity (normal or inverted) apply to both the transmitter and receiver and may not be independently selected for each direction. All of these except the baud rate are selected with individual bits of the Control 1 Register (see Register Descriptions in Section 16.5).

The baud rate for each UART is determined by the BAUDRATE[10:0] value in the Control 2 Register and the UART clock frequency (nominally 9.216 MHz). The following equation determines the baud rate:

Baud Rate = 9.216 MHz ÷ ((BAUDRATE[10:0] + 1) * 16)

The UART clock frequency is determined by the Clock Module configuration (see Section 6). Assuming the intended frequency of 9.216 MHz, Table 16-1 shows the value that should be written to the BAUDRATE[10:0] control bits to generate the standard baud rates.

Table 16-1 Standard UART Baud Rates

Baud rate	BAUDRATE[10:0]
38400	14
19200	29
9600	59
4800	119
2400	239
1200	479
600	959
300	1919

16.2.3 Interrupt Operation

Each UART may be configured to generate CPU interrupts for a number of events. When any of these events occur, a corresponding status bit will be set in the Interrupt Status 2 Register (see Section 8 for details on the Interrupt Module). Table 16-2 lists the UART events that can generate interrupts. The entries are in terms of the mnemonic used in this document and the actual event. In the mnemonic the lower-case 'n' is replaced with 'A' for UARTA or 'B' for UARTB.

Table 16-2 UART Interrupt Events

mnemonic	interrupting event
UARTnRXINT	receiver holding register becomes full
UARTnRXOVERRUNINT	receiver shift register becomes full when both holding register and PRXHOLD are full
UARTnFRAMEERRINT	receiver does not detect stop bit-character not 0x0
UARTnBREAKINT	receiver detects BREAK condition-character is 0x0 and no stop bit
UARTnPARITYERRINT	receiver detects parity bit error
UARTnTXINT	transmit holding register becomes empty
UARTnTXOVERRUNINT	transmit holding register written to when full
UARTnEMPTYINT	transmit shift register becomes empty when holding register is also empty
UARTnDMAFULLINT	DMA controller reaches end of specified buffer
UARTnDMAHALFINT	DMA controller reaches half-way point in specified buffer

While these events cause a bit to be set in Interrupt Status 2 Register, they will not cause an interrupt unless they are configured to do so. Interrupts are enabled through the Enable Interrupt 2 Register. Interrupt status bits are cleared through the Clear Interrupt 2 Register. Note that interrupt status bits are set and need to be cleared regardless of whether or not the interrupt is enabled. See Section 8 for more general information about the TMPR3922 interrupts.

16.2.3.1 Responding to Interrupt Status Bits

When responding to an event (whether via interrupt or polling), it is important to clear the interrupt status bit before taking the action that will re-enable the event. This will ensure that an event will not be missed.

For example, when the Receive Holding Register becomes full, the UARTnRXINT bit will get set. If the Receive Holding Register is read before the UARTnRXINT bit is cleared, it is possible that sometime after the Receive Holding Register becomes empty and before the UARTnRXINT bit is cleared, a second character will be transferred to the Receive Holding Register. This transfer will also cause the UARTnRXINT bit to be set. Obviously, if the CPU (still reacting to the first event) then clears the UARTnRXINT bit, the second event will be missed. As a result, the CPU will not be "aware" that the Receive Holding Register is full and an overrun interrupt will eventually occur.

16.2.3.2 Related Interrupts

UARTARXINT:

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

UARTARXOVERRUNINT:

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

UARTAFRAMEERRINT:

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

UARTABREAKINT:

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

UARTAPARITYERRINT:

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

UARTATXINT:

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

UARTATXOVERRUNINT:

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

UARTAEMPTYINT:

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

UARTADMAFULLINT:

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

UARTADMAHALFINT:

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

UARTBRXINT:

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

UARTBRXOVERRUNINT:

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

UARTBFRAMEERRINT:

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

UARTBBREAKINT:

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

UARTBPARITYERRINT:

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

UARTBTXINT:

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

UARTBTXOVERRUNINT:

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

UARTBEMPTYINT:

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

UARTBDMAFULLINT:

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

UARTBDMAHALFINT:

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

16.2.4 DMA Operation

Each UART has one DMA channel which may be used with either the transmitter or the receiver. DMA operation involves specifying a memory buffer and enabling DMA. Once enabled, characters will flow either from the memory buffer to the transmitter, or from the receiver to the memory buffer.

The memory buffer is specified in terms of its start address in physical address space, and its length in bytes. Note that the CPU must know the mapping between virtual and physical addresses. The start address must be written to the DMA Control 1 Register. The two LSB's are ignored, thereby forcing the buffer to begin on a long word boundary. The buffer length minus one must be written to the DMA Control 2 Register.

Once the buffer has been specified, DMA may be enabled. This is done with either ENDMARX or ENDMATX in the Control 1 Register. Only one of these two bits should be set at any one time. Operation is undefined if both are set. Once enabled, UART DMA will begin. For the transmitter, this means that when the Transmit Holding Register becomes empty, the DMA controller will transfer a byte from the specified buffer to the Transmit Holding Register, and update its internal buffer pointer. For the receiver, this means that when the Receive Holding Register becomes full, the DMA controller will transfer a byte from the Receive Holding Register to the specified buffer, and update its internal buffer pointer.

The ENDMALOOP control bit in the Control 1 Register determines when DMA will stop. If this bit is cleared, DMA will stop when the end of the buffer is reached. If this bit is set, the DMA controller will run continuously, looping back to the start of the buffer when the buffer end is reached. In either case, DMA can be explicitly halted by clearing the ENDMARX or ENDMATX bit.

The DMA Count Register is a read-only register which contains the UART DMA controller's buffer counter status. The counter counts down so that the value in this status register indicates how much of the buffer remains to be accessed (on this pass) by the DMA controller.

There are two interrupt status bits associated with the DMA controller: UARTnDMAFULLINT and UARTnDMAHALFINT. These are set as the DMA controller buffer pointer reaches the end and the half-way point of the buffer.

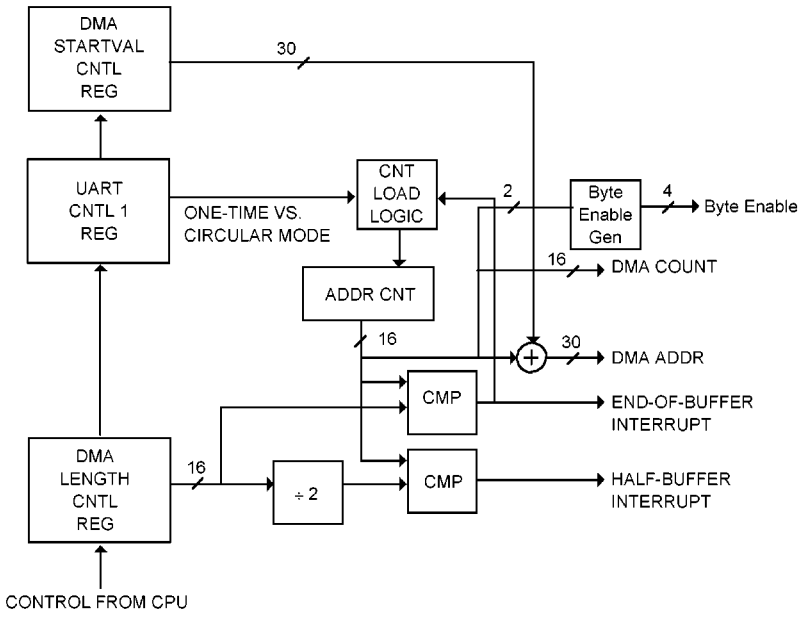


Figure 16-1 UART DMA

16.2.5 Internal Loopback

The LOOPBACK bit of the Control 1 Register, when set, places the UART in an internal loopback mode. When in this mode, the transmit and receive lines of the UART are internally tied together, the external transmit line is driven high, and the external receive line is ignored. This loopback mode is for testing purposes only.

16.3 Transmitter Operation

Figure 16-2 shows a block diagram of the transmitter portion of each UART. Characters to be transmitted are written by the CPU to the Transmit Holding Register. This should only be done when the Transmit Holding Register is empty which is indicated by either the EMPTY bit of the Control 1 Register, or the UARTnTXINT interrupt. Writing a character to the Transmit Holding Register when it is not empty results in a transmitter overrun.

The character in the Transmit Holding Register is transferred into the Transmit Shift Register when the Transmit Shift Register becomes empty. Transferred with the character is 1 start bit, 1 or 2 stop bits (depending on the setting of TWOSTOP in the Control 1 Register), and 0 or 1 parity bits (depending on the setting of ENPARITY and EVENPARITY in the Control 1 Register). Beginning at the next baud period following the transfer, the character is shifted out to the transmit line. The start bit is shifted first, followed by the LSB through MSB of the (7-bit or 8-bit) character, followed by the parity bit (if any), and ending with the stop bit(s). As the last stop bit is shifted out, the transmitter will load the next character from the Transmit Holding Register if one has been written. If the Transmit Holding Register is empty, the Transmit Shift Register will continue to shift out stop bits (logic level "1") until a character is written by the CPU into the Transmit Holding Register.

Depending on the configuration, as few as nine bits (7-bit characters, no parity, 1 stop bit), and as many as twelve bits (8-bit characters, parity enabled, 2 stop bits) are transmitted for every character.

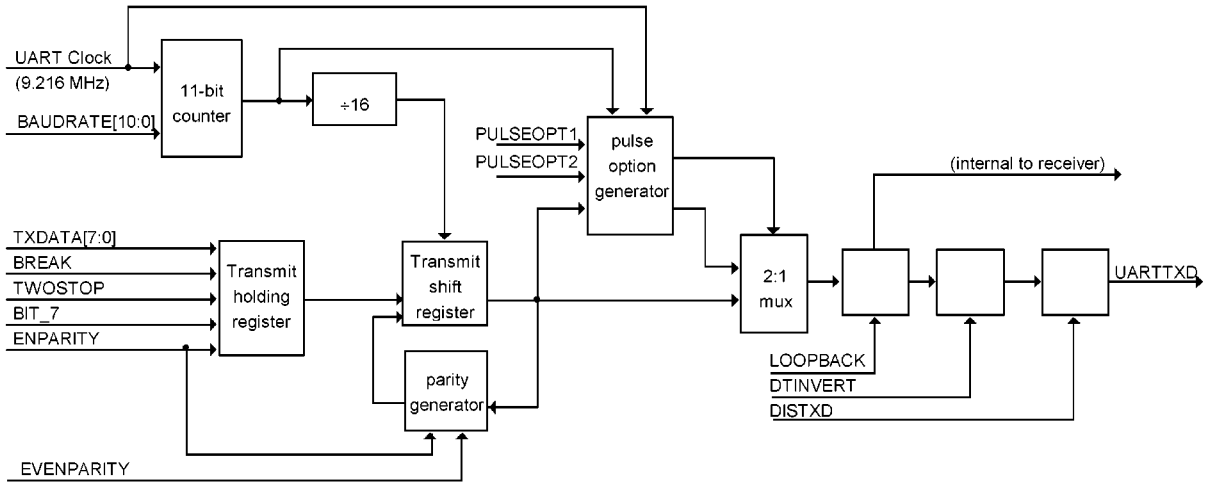


Figure 16-2 UART Transmitter

16.3.1 Transmitter Pulse Output Operation

In normal operation (neither pulse option selected), the transmit line is tied directly to the Transmit Shift Register output. Each transmitted bit causes the transmit line to go high or low for a full baud period.

To support the IRDA Infrared protocol, there are two pulse options which result in a modified signal on the output line. When either of these options are selected, transmitted zeros will cause the transmit line to go low for only a fraction of the baud period. Ones are transmitted normally. The difference between the two pulse option modes is the duration that the line will stay low for each transmitted zero.

Pulse Option 1 (selected with PULSEOPT1 of the Control 1 Register) results in a duration of six UART clock cycles. The UART clock frequency is controlled by the Clock Module (see Section 6), and is normally set to be 9.216 MHz. For Pulse Option 1, this clock frequency would result in a low pulse of 1.63 μs (= 15 ÷ 9.216). Note that this duration is irrespective of the baud rate.

Pulse Option 2 (selected with PULSEOPT2 of the Control 1 Register) results in a duration of 3/16 of the selected baud period.

The mode of operation with both pulse options selected (both PULSEOPT1 and PULSEOPT2 asserted) is reserved for a future implementation.

16.3.2 Transmitter Disable Operation

The transmit line may be disabled with the DISTXD bit of the Control 1 Register. In this mode the transmit line is held at logic level "0". Otherwise, the transmitter operates normally as described above.

16.3.3 Transmitter BREAK Operation

A BREAK may be transmitted by writing the value 0x100 to the Transmit Holding Register. The BREAK is transmitted after the character (if any) in the Transmit Shift Register is sent. The BREAK condition continues until explicitly terminated by the CPU. The CPU may continue to write the value 0x100 to the Transmit Holding Register in order to precisely control the duration of the BREAK. The BREAK condition is terminated when the CPU writes the value 0x00 to the Transmit Holding Register.

16.3.4 Transmitter Overrun

The transmit overrun condition occurs if the CPU writes a character to the Transmit Holding Register when it is not empty. This condition is indicated by the UARTnTXOVERRUNINT interrupt status bit. The character in the Transmit Holding Register resulting from an overrun is undefined.

16.4 Receiver Operation

Figure 16-3 shows a block diagram of the receiver portion of each UART. In normal operation, the receiver waits for a start bit on the receive line. One baud cycle after receiving the start bit, the next 7 or 8 bits (depending on BIT_7 in the Control 1 Register) are clocked into the Receive Shift Register. If parity is enabled (depending on ENPARITY and EVENPARITY in the Control 1 Register), the next bit is compared with the expected parity. As the stop bit is clocked in and checked, the assembled character is transferred from the Receive Shift Register to the PRXHOLD Register.

The PRXHOLD Register is a buffer between the Receive Shift Register and the Receive Holding Register. The PRXHOLDFULL bit of the Control 1 Register, when asserted, indicates that the PRXHOLD Register is full.

On the first UART clock period (nominally 9.216 MHz) that the Receive Holding Register is empty and the PRXHOLD Register is full, the character is transferred to the Receive Holding Register. This transfer results in the RXHOLDFULL bit of the Control 1 Register and the UARTnRXINT interrupt status bit both asserting.

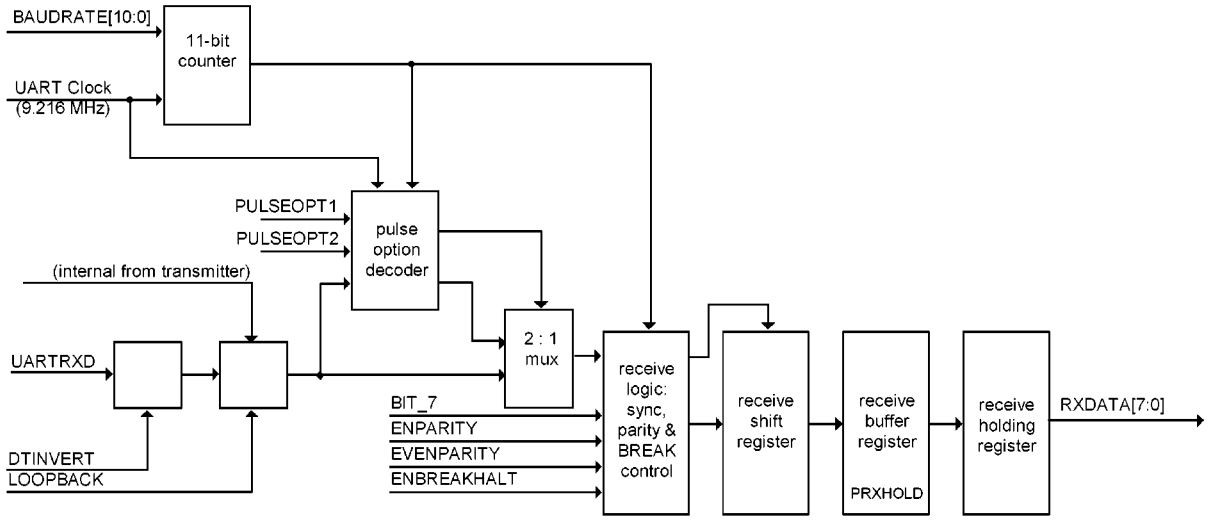


Table 16-3 UART Receiver

16.4.1 Receiver BREAK Operation

The receiver detects a BREAK condition when the received character is 0x00 and there is no stop bit. When this occurs the receiver generates a UARTnBREAKINT interrupt. The interrupt does not assert until the BREAK character is transferred to the Receive Holding Register. In other words, the interrupt is held off until all characters preceding the BREAK are read by the CPU.

Additionally, the operation of the receiver in response to a BREAK is affected by the ENBREAKHALT bit of the Control 1 Register. Normally (with the ENBREAKHALT bit cleared), the receiver will start receiving characters again as soon as a stop bit is detected on the line. However, if the ENBREAKHALT bit is set, the receiver shuts down until the BREAK character is read by the CPU from the Receive Holding Register. While shut down, the receiver ignores all data on the receive line.

16.4.2 Receiver Frame Error Condition

A receiver frame error occurs when the receiver does not detect a stop bit following the character. This condition results in a UARTnFRAMEERRINT interrupt being generated. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the frame error is transferred to the Receive Holding Register.

16.4.3 Receiver Overrun Condition

A receiver overrun occurs when a character is received and both the PRXHOLD Register and Receive Holding Register are full. This condition occurs at the point that the stop bit of the new character is clocked in and results in a UARTnRXOVERRUNINT interrupt. The new character will overwrite the character in the PRXHOLD Register. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the overrun error is transferred to the Receive Holding Register.

16.4.4 Receiver Parity Error Condition

A receiver parity error occurs when parity is enabled and the receiver detects the wrong polarity of the bit following the character and preceding the stop bit. This condition results in a UARTnPARITYERRINT interrupt being generated. Like the UARTnBREAKINT interrupt, the interrupt is not asserted until the character associated with the parity error is transferred to the Receive Holding Register.

16.4.5 Receiver Pulse Operation

In normal operation the received signal corresponds to the polarity of the data for the entire baud period. When the received signal is from a pulsed transmitter, zeros are low for only a fraction of the baud period. If either pulse option is selected (with PULSEOPT1 or PULSEOPT2 of the Control 1 Register) the receiver reconstructs the data stream by extending low pulses to the full baud period.

16.5 UART Registers

16.5.1 UART Control 1 Register

OFFSET=$0B0 : UARTA
OFFSET=$0C8 : UARTB

Bit	Label	RESET	Read/Write
31	UARTON	0	R
30	EMPTY	1	R
29	PRXHOLDFULL	0	R
28	RXHOLDFULL	0	R
27-16	Reserved		
15	ENDMARX	0	R/W
14	ENDMATX	0	R/W
13	TESTMODE	0	R/W
12	ENBREAKHALT	0	R/W
11	ENDMATEST	0	R/W
10	ENDMALOOP	0	R/W
9	PULSEOPT2	0	R/W
8	PULSEOPT1	0	R/W
7	DTINVERT	0	R/W
6	DISTXD	0	R/W
5	TWOSTOP	0	R/W
4	LOOPBACK	0	R/W
3	BIT_7	0	R/W
2	EVENPARITY	0	R/W
1	ENPARITY	0	R/W
0	ENUART	0	R/W

UARTON: read-only

When the ENUART bit is disabled, the module will not shut down until the Transmit Holding Register and Transmit Shift Register are empty plus a couple of clocks. This bit provides the status as to whether the module is still enabled or not.

EMPTY: read-only

This bit is high if the Transmit Holding Register and Transmit Shift Register are both empty.

PRXHOLDFULL: read-only

The receive data path consists of an 8-bit shift register plus two 8-bit holding registers. Whenever the receiver finishes receiving a character, it will transfer the contents into the PRXHOLD Register. The contents of this register will then be loaded into the RXHOLD Register whenever this RXHOLD Register is empty. The RXHOLD Register is emptied by reading a byte from the Receive Holding Register. This PRXHOLDFULL bit provides status as to whether there is a valid byte of data in the PRXHOLD Register.

RXHOLDFULL: read-only

This bit provides the status of the RXHOLD Register.

ENDMARX:

This bit enables the DMA receive function. This bit should not be set until the UARTDMACNTL1 and UARTDMACNTL2 Registers are setup and the module is enabled. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

ENDMATX:

This bit enables the DMA transmit function. This bit should not be set until the UARTDMACNTL1 and UARTDMACNTL2 registers are setup, the module is enabled and the empty flag is set. Only one of ENDMARX or ENDMATX can be set at a time since there is only one DMA channel.

TESTMODE:

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

ENBREAKHALT:

Setting this bit will cause the receiver to halt after receiving a break, until the Receive Holding Register is emptied and the UARTRXD signal goes to the marking state. Otherwise, the receiver will halt until the UARTRXD signal goes to the marking state without regard to the status of the Receive Holding Register.

ENDMATEST:

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

ENDMALOOP:

The DMA controller supports two modes depending on the state of this bit. When ENDMALOOP is low, the DMA controller will stop executing when it reaches the end of the DMA buffer. When ENDMALOOP is high, the DMA controller will loop back to the start of the DMA buffer when the end of the DMA buffer is reached and will continue operating.

PULSEOPT2:

Setting this bit will cause the transmitted data to pulse low for three baud clocks instead of the normal 16 baud clocks. Setting either this bit or PULSEOPT1 will cause the receiver to expect the data to be a pulsed input.

PULSEOPT1:

Setting this bit will cause the transmitted data to pulse low for a fixed six 9.216MHz clocks instead of the normal 16 baud clocks.

DTINVERT:

Setting this bit will cause the UARTTXD and UARTRXD signals to be inverted.

DISTXD:

Setting this bit will cause the UARTTXD signal to go low.

TWOSTOP:

Setting this bit will cause the transmitter to transmit two stop bits instead of the normal one stop bit.

LOOPBACK:

Setting this bit will cause the transmitted data to internally loop back to the receive data. The UARTTXD pin is held high when this bit is set.

BIT_7:

Setting this bit selects 7-bit per character mode instead of 8-bit per character mode.

EVENPARITY:

Setting this bit selects even parity instead of odd parity, if the ENPARITY bit is set.

ENPARITY:

Setting this bit will cause parity to be generated and received.

ENUART:

Setting this bit will enable the UART Module. When this bit is cleared the module is kept in a reset state. This bit should not be set until all other control bits are setup.

16.5.2 UART Control 2 Register

OFFSET=$0B4: UARTA write-only
OFFSET=$0CC: UARTB write-only

Bit	Label	RESET	Read/Write
31-11	Reserved		
10-0	BAUDRATE[10:0]	X	W

BAUDRATE[10:0]: write-only

These bits define the baud rate of the transmitter and receiver. The following equation determines the baud rate :

Baud Rate = 9.216 MHz ÷ ((BAUDRATE[10:0] + 1) * 16)

16.5.3 UART DMA Control 1 Register

OFFSET=$0B8: UARTA write-only
OFFSET=$0D0: UARTB write-only

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

DMASTARTVAL[31:2]: write-only

These bits define the start address for the DMA buffer.

16.5.4 UART DMA Control 2 Register

OFFSET=$0BC: UARTA write-only
OFFSET=$0D4: UARTB write-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	DMALENGTH[15:0]	X	W

DMALENGTH[15:0]: write-only

These bits define the length of the DMA buffer. The last address in the DMA buffer is given by DMASTARTVAL[31:2] + DMALENGTH[15:0].

16.5.5 UART DMA Count

OFFSET=$0C0: UARTA read-only
OFFSET=$0D8: UARTB read-only

Bit	Label	RESET	Read/Write
31-16	Reserved		
15-0	DMACNT[15:0]	X	W

DMACNT[15:0]: read-only

These bits provide the status of the DMA counter.

16.5.6 UART Transmit Holding Register

OFFSET=$0C4: UARTA write-only
OFFSET=$0DC: UARTB write-only

Bit	Label	RESET	Read/Write
31-9	Reserved		
8	BREAK	X	W
7-0	TXDATA[7:0]	X	W

BREAK: write-only

Setting this bit, along with writing $00 to TXDATA[7:0], will cause a break to be generated. The break will continue until this bit is cleared, along with TXDATA[7:0] maintained at $00. This register should only be loaded after the UARTTXINT interrupt is set. The break will flow through the Transmit Holding Register, thus it will not start the break until the current character that is being transmitted from the Transmit Shift Register is finished.

TXDATA[7:0]: write-only

These bits are the data to be transmitted. This register should only be loaded after the UARTTXINT interrupt is set. For 7-bit mode, only bits 6:0 are valid.

16.5.7 UART Receiver Holding Register

OFFSET=$0C4: UARTA read-only
OFFSET=$0DC: UARTB read-only

Bit	Label	RESET	Read/Write
31-8	Reserved		
7-0	RXDATA[7:0]	X	R

RXDATA[7:0]: read-only

These bits are the receive data. The bits are valid after the UARTRXINT interrupt is set or when the RXHOLD flag is asserted. For 7-bit mode, only bits 6:0 are valid.