TOSHIBA

32-Bit TX System RISC
TX39 Family
TMPR3916

TOSHIBA CORPORATION

MIPS16, application Specific Extensions and R3000A are a trademark of MIPS Technologies, Inc.

The information contained herein is subject to change without notice.

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

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

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

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

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

© 2002 TOSHIBA CORPORATION
All Rights Reserved

Preface

This document describes the basic functions of TMPR3916F "Capricorn2". This document is meant to provide all information, which is needed to program and operate the device from a software developer's point of view.

October 2002

Table of Contents

Handling precautions

TMPR3916

1. The TMPR3916 1-1
   1.1 Applications and References 1-1
   1.2 Features 1-2
   1.3 Differences Between TX3903AF and TMPR3916 1-3
   1.4 Structure of TMPR3916 and a System Example 1-4
   1.5 Address Map 1-6
   1.6 Clocks 1-8
   1.7 Resets 1-9
   1.8 Time-Out-Error Control Unit 1-10
   1.9 Operating Modes of TMPR3916 1-10
   1.10 Chip Configuration Register (CCR) 1-11
2. Memory Controller (MC) 2-1
   2.1 Structure of Memory Controller 2-2
   2.2 Example Memory Configuration 2-3
   2.3 Ports of Memory Controller 2-6
   2.4 Registers 2-6
   2.5 SDRAMC Functions 2-7
   2.6 MEMC Function 2-20
3. Graphics Display Controller (GDC) 3-1
   3.1 GDC Structure 3-2
   3.2 Scrolling 3-9
   3.3 Internal Blockdiagram 3-12
   3.4 Registers 3-13
   3.5 Setting Examples 3-21
4. Interrupt Controller (INTC) 4-1
   4.1 Basic Interrupt Handling 4-1
   4.2 Registers 4-2
   4.3 Non Maskable Interrupt 4-4
5. TIMER 5-1
   5.1 PWM Timer 5-1
   5.2 Periodic Timers 5-3
6. Direct Memory Access Controller (DMAC) 6-1
   6.1 Programming the DMA Controller 6-2
   6.2 Registers 6-5
7. CAN Module (TXCAN) 7-1
   7.1 Block Diagram 7-2
   7.2 TXCAN Registers 7-3
   7.3 TXCAN Interrupt Logic 7-22
   7.4 TXCAN Operation Modes 7-25
   7.5 Handling of Message-Objects 7-30
8. Parallel Interface (PORT) 8-1

9. Synchronous Serial I/O (TXSEI) 9-1
9.1 TXSEI Structure 9-2
9.2 Registers 9-3
9.3 TXSEI Operations 9-15
9.4 Interrupts 9-19
10. Asynchronous Serial Interface (UART) 10-1
10.1 Operations on Serial Interface 10-1
10.2 Registers 10-10
10.3 Timings 10-19
10.4 Flow Charts 10-21
11. Electrical Characteristics 11-1
11.1 DC Characteristics of TMPR3916 11-1
11.2 Power Up Sequence 11-2
11.3 Crystal Oscillator 11-3
11.4 View DAC 11-4
11.5 Standby Mode Timing 11-5
11.6 Boot Device 11-5
11.7 SDRAM Timing 11-6
11.8 ROM / SRAM Timing 11-7
11.9 External Slave 11-8
11.10 External Interrupts and NMI 11-10
11.11 General Purpose I/O (PORT Module) 11-10
11.12 TXSEI Timing 11-11
12. Package Dimension 12-1
12.1 Pin Assignment 12-1
12.2 Pin Functions 12-5
Appendix A. Register Overview of TMPR3916 ..... A-1

Handling Precautions

1. Using Toshiba Semiconductors Safely

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

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

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

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

2. Safety Precautions

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

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

[Explanation of labels]

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

[Explanation of graphic symbol]

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

2.1 General Precautions regarding Semiconductor Devices

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

2.2 Precautions Specific to Each Product Group

2.2.1 Optical semiconductor devices

⚠ DANGER

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

⚠ WARNING

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

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

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

2.2.2 Power devices

⚠ DANGER

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

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

⚠WARNING

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

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

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

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

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

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

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

⚠CAUTION

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

2.2.3 Bipolar ICs (for use in automobiles)

⚠CAUTION

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

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

protective functions fail, the device may break down causing injury to the user.

3. General Safety Precautions and Usage Considerations

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

3.1 From Incoming to Shipping

3.1.1 Electrostatic discharge (ESD)

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

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

(1) Work environment

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

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

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

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

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

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

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

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

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

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

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

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

- Make sure that CRT displays in the working area are protected against static charge, for example by a VDT filter. As much as possible, avoid turning displays on and off. Doing so can cause electrostatic induction in devices.

- Keep track of charged potential in the working area by taking periodic measurements.

- Ensure that work chairs are protected by an anti-static textile cover and are grounded to the floor surface by a grounding chain. (Suggested resistance between the seat surface and grounding chain is 7.5×10^5 to $10^{12}\Omega$.)

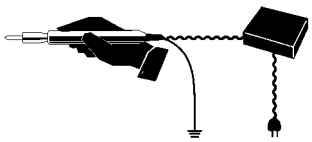
- Install anti-static mats on storage shelf surfaces. (Suggested surface resistivity is 10^4 to 10^8 Ω/sq; suggested resistance between surface and ground is 7.5×10^5 to 10^8 Ω.)

- For transport and temporary storage of devices, use containers (boxes, jigs or bags) that are made of anti-static materials or materials which dissipate electrostatic charge.

- Make sure that cart surfaces which come into contact with device packaging are made of materials which will conduct static electricity, and verify that they are grounded to the floor surface via a grounding chain.

- In any location where the level of static electricity is to be closely controlled, the ground resistance level should be Class 3 or above. Use different ground wires for all items of equipment which may come into physical contact with devices.

(2) Operating environment



- Operators must wear anti-static clothing and conductive shoes (or a leg or heel strap).

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

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

- If the tweezers you use are likely to touch the device terminals, use anti-static tweezers and in particular avoid metallic tweezers. If a charged device touches a low-resistance tool, rapid discharge can occur. When using vacuum tweezers, attach a conductive chucking pat to the tip, and connect it to a dedicated ground used especially for anti-static purposes (suggested resistance value: 10^4 to 10^8 Ω).

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

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

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

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

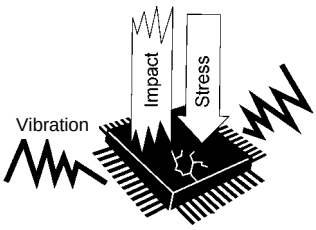
- Equipment safety covers installed near devices should have resistance ratings of 10^9 Ω or less.

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

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

3.1.2 Vibration, impact and stress

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



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

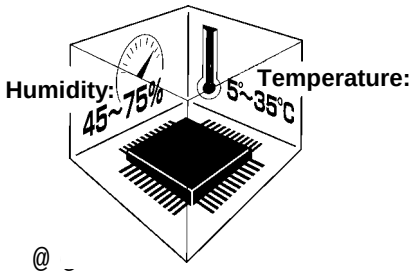
If a device is subjected to especially strong vibration, mechanical shock or stress, the package or the chip itself may crack. In products such as CCDs which incorporate window glass, this could cause surface flaws in the glass or cause the connection between the glass and the ceramic to separate.

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

3.2 Storage

3.2.1 General storage

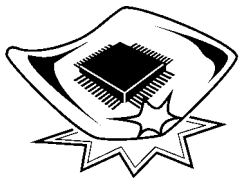
- Avoid storage locations where devices will be exposed to moisture or direct sunlight.
- Follow the instructions printed on the device cartons regarding transportation and storage.
- The storage area temperature should be kept within a temperature range of 5°C to 35°C, and relative humidity should be maintained at between 45% and 75%.
- Do not store devices in the presence of harmful (especially corrosive) gases, or in dusty conditions.



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

3.2.2 Moisture-proof packing

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



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

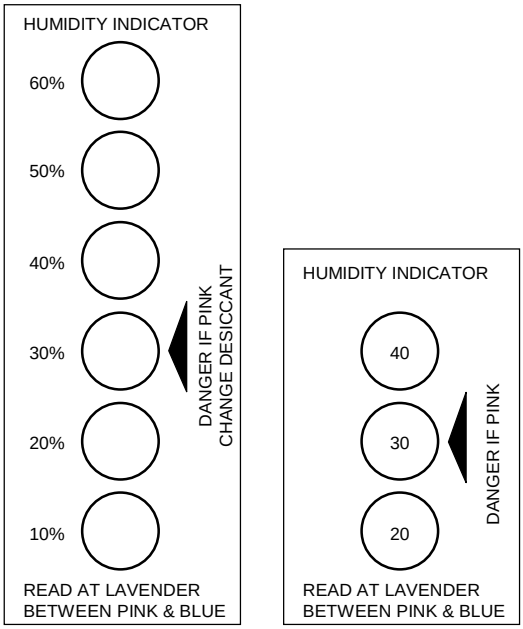
- Do not drop or toss device packing. The laminated aluminum material in it can be rendered ineffective by rough handling.
- The storage area temperature should be kept within a temperature range of 5°C to 30°C, and relative humidity should be maintained at 90% (max). Use devices within 12 months of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

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

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

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

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



(a) 6-point indicator (b) 3-point indicator

Figure 1 Humidity indicator

3.3 Design

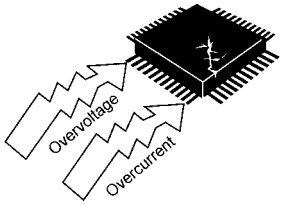
Care must be exercised in the design of electronic equipment to achieve the desired reliability. It is important not only to adhere to specifications concerning absolute maximum ratings and recommended operating conditions, it is also important to consider the overall environment in which equipment will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

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

3.3.1 Absolute maximum ratings

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

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



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

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

3.3.2 Recommended operating conditions

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

3.3.3 Derating

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

3.3.4 Unused pins

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

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

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

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

3.3.5 Latch-up

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

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

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

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

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

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

3.3.6 Input/Output protection

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

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

3.3.7 Load capacitance

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

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

3.3.8 Thermal design

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

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

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

(3) Derate the device's absolute maximum ratings to minimize thermal stress from power dissipation.

$\theta ja = \theta jc + \theta ca$

$\theta ja = (Tj - Ta) / P$

$\theta jc = (Tj - Tc) / P$

$\theta ca = (Tc - Ta) / P$

in which θja = thermal resistance between junction and surrounding air (°C/W)

θjc = thermal resistance between junction and package surface, or internal thermal resistance (°C/W)

θca = thermal resistance between package surface and surrounding air, or external thermal resistance (°C/W)

Tj = junction temperature or chip temperature (°C)

Tc = package surface temperature or case temperature (°C)

Ta = ambient temperature (°C)

P = power dissipation (W)

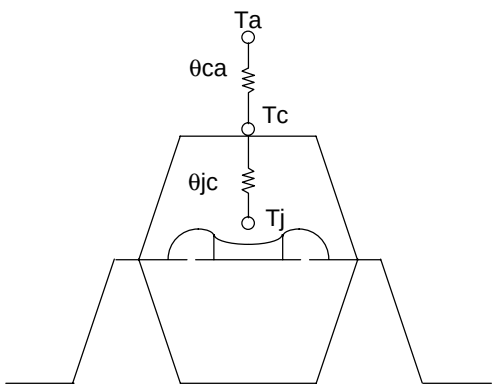


Figure 2 Thermal resistance of package

3.3.9 Interfacing

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

3.3.10 Decoupling

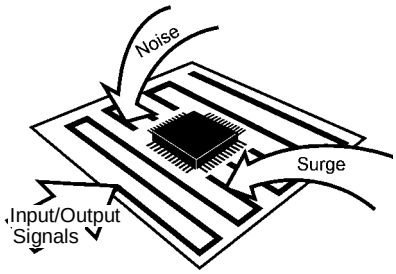
Spike currents generated during switching can cause Vcc (Vdd) and GND (Vss) voltage levels to fluctuate, causing ringing in the output waveform or a delay in response speed. (The power supply and GND wiring impedance is normally 50 Ω to 100 Ω.) For this reason, the impedance of power supply lines with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

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

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

3.3.11 External noise

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



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

3.3.12 Electromagnetic interference

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

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

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

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

3.3.13 Peripheral circuits

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

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

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

3.3.14 Safety standards

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

3.3.15 Other precautions

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

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

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

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

3.4 Inspection, Testing and Evaluation

3.4.1 Grounding

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

3.4.2 Inspection Sequence

⚠CAUTION

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

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

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

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

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

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

3.5 Mounting

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

3.5.1 Lead forming

⚠CAUTION

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

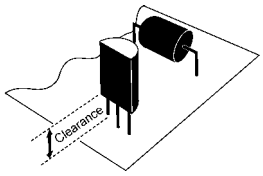
② Do not touch the tips of device leads. Because some types of device have leads with pointed tips, you may prick your finger.

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

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

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

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



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

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

3.5.2 Socket mounting

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

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

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

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

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

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

3.5.3 Soldering temperature profile

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

(1) Using a soldering iron

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

(2) Using medium infrared ray reflow

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

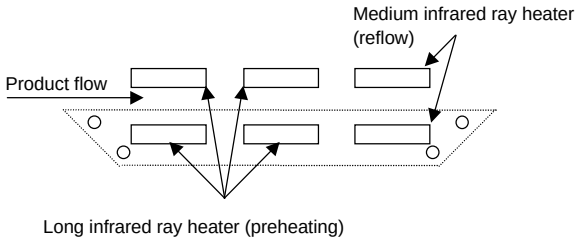


Figure 3 Heating top and bottom with long or medium infrared rays

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

- Refer to Figure 4 for an example of a good temperature profile for infrared or hot air reflow.

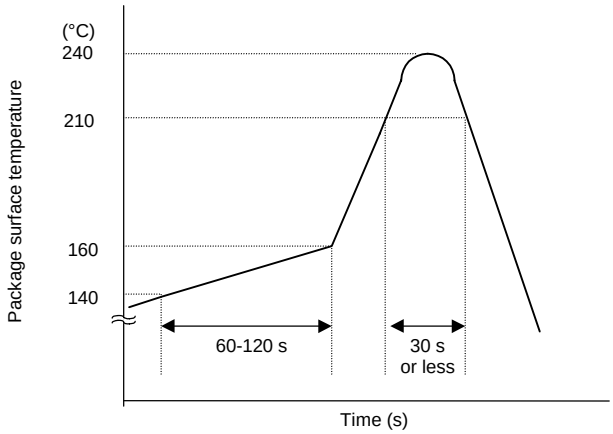


Figure 4 Sample temperature profile for infrared or hot air reflow

(3) Using hot air reflow

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

- For an example of a recommended temperature profile, refer to Figure 4 above.

(4) Using solder flow

- Apply preheating for 60 to 120 seconds at a temperature of 150°C.

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

- For surface-mount packages, complete soldering within 5 seconds at a temperature of 250°C or

less in order to prevent thermal stress in the device.

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

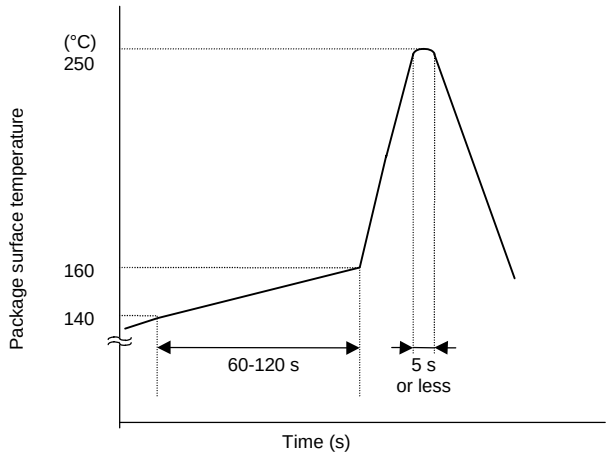


Figure 5 Sample temperature profile for solder flow

3.5.4 Flux cleaning and ultrasonic cleaning

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

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

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

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

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

Frequency: 27 kHz ~ 29 kHz

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

Cleaning time: 30 seconds or less

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

3.5.5 No cleaning

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

3.5.6 Mounting tape carrier packages (TCPs)

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

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

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

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

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

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

3.5.7 Mounting chips

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

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

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

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

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

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

3.5.8 Circuit board coating

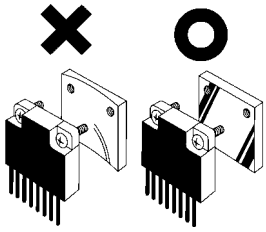
When devices are to be used in equipment requiring a high degree of reliability or in extreme environments (where moisture, corrosive gas or dust is present), circuit boards may be coated for protection. However, before doing so, you must carefully consider the possible stress and contamination effects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

3.5.9 Heat sinks

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

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

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



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

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

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

3.5.10 Tightening torque

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

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

3.5.11 Repeated device mounting and usage

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

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

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

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

3.6 Protecting Devices in the Field

3.6.1 Temperature

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

3.6.2 Humidity

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

3.6.3 Corrosive gases

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

3.6.4 Radioactive and cosmic rays

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

3.6.5 Strong electrical and magnetic fields

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

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

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

3.6.7 Dust and oil

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

3.6.8 Fire

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

3.7 Disposal of Devices and Packing Materials

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

4. Precautions and Usage Considerations

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

4.1 Microcontrollers

4.1.1 Design

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

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

(2) Undefined functions

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

TX3916F

1. The TMPR3916

1.1 Applications and References

The TMPR3916 is a family member of Toshiba's 32-bit system RISC family. As an application-specific standard product (ASSP) it is designed for a wide range of applications such as:

- Car Navigation Systems
- Driver Information Displays
- Personal Digital Assistants (PDAs)
- Musical Instruments
- Electronic Book Players

The TMPR3916 uses a TX39/H core as its CPU. The TX39/H CPU core is a RISC processor developed by Toshiba based on the R3000A architecture of MIPS Technologies Inc. .

In addition to the processor core, this ASSP includes peripheral circuits such as a graphics display controller, a memory controller, a DMA controller, several serial communication interfaces, CAN-bus interfaces, interval timers and general purpose I/Os.

Please refer to the following document for information about the TX39 core architecture, including the instruction set:

32-Bit TX System RISC TX39 Family Architecture (Document Number 44137D)

1.2 Features

Miscellaneous:

- 60 MHz maximum operating frequency:
- 208 pin QFP package (QFP208-P-2828-0.50)
- 3.3 V power supply voltage
- ca. 1200mW Maximum Power Dissipation
- -40°C to 85°C operating ambient temperature
- Built-in clock generator
- 5V tolerant I/Os on UARTs, TXSEI and CAN-bus interface
- Unified memory architecture with a high performance dual bus structure (Video bus + CPU bus)

Graphics Display Controller:

- Four-layer (A-D) overlay hardware processing with transparent color:
- Layer A, B can display 256 out of 64 K colors each
- Layer C, D can display 16 out of 64 K colors each
- Alternatively layer A may be configured in picture mode with 64K colors
- SDRAM and SRAM frame-buffer memory (SRAM recommended only for low resolutions)
- Burst access to frame-buffer memory
- 16 bytes built-in dot buffer
- Built-in color look-up tables (for plane A-D, 544 colors in total)
- Built-in three-channel 6-bit video DAC, alternatively connection to digital displays (digital RGB output)
- Dotclock, horizontal and vertical synchronisation signals can be generated internally or input from external device

Built-in TX39 Core:

- Toshiba-developed TX39H core based on MIPS R3000A architecture
- 4 KB Instruction Cache, 1 KB Data Cache
- Built-in debug support unit for in-system debugging incl. real time PC-tracing
- Big-endian coding

Peripheral Controllers:

- Memory Controller (MEMC), 4 channels for SRAM, ROM, Flash
- SDRAM controller (SDRAMC), 2 channels
- DMA controller (DMAC), 2 channels
- Interrupt controller (INTC), 13 internal interrupts, 3 external interrupts, 1 non-maskable interrupt (NMI)
- Serial I/O : UART 4 channels, TXSEI 1 channel (SPI compatible, with FIFO´s)
- CAN-bus controller (TXCAN), 2 channels, 16 mailboxes each
- 30 Pin General Purpose I/O´s (PORT)

1.3 Differences Between TX3903AF and TMPR3916

In catchwords this section explains changed features of TMPR3916 in comparison to its predecessor TX3903AF. For detailed information please have a deeper look into this document.

- 60 MHz operating frequency
- added dual CAN device
- added TXSEI functionality
- added two channel SDRAM Controller
- separate Video- and CPU-bus to SDRAM in order to increase system performance
- removed EDO-DRAM channels from MEMC
- no more support of pipelined burst SRAM
- graphics display controller: increased number of colors from 16 to 256 in layer A and B
- raised number of 16 general purpose IOs to 30 and added capability of triggering interrupt
- extended timer functionality to PWM-support
- increased number of internal interrupts
- modified UART incl. register structure
- external bus-master functionality is not supported any more

1.4 Structure of TMPR3916 and a System Example

The following picture shows the block diagram of the TMPR3916:

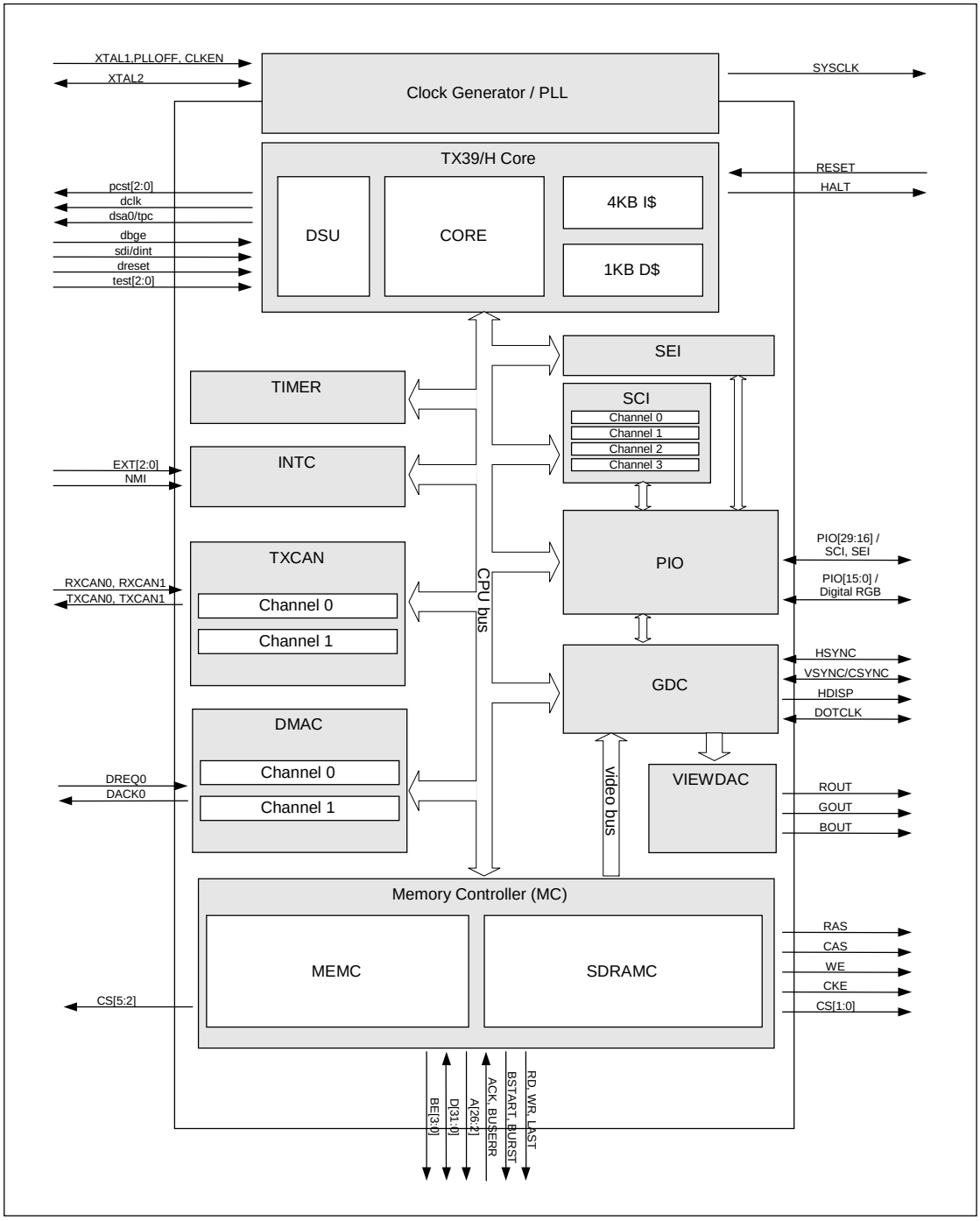


Figure 1.4.1 Block Diagram TMPR3916

The following picture shows a system example with TMPR3916:

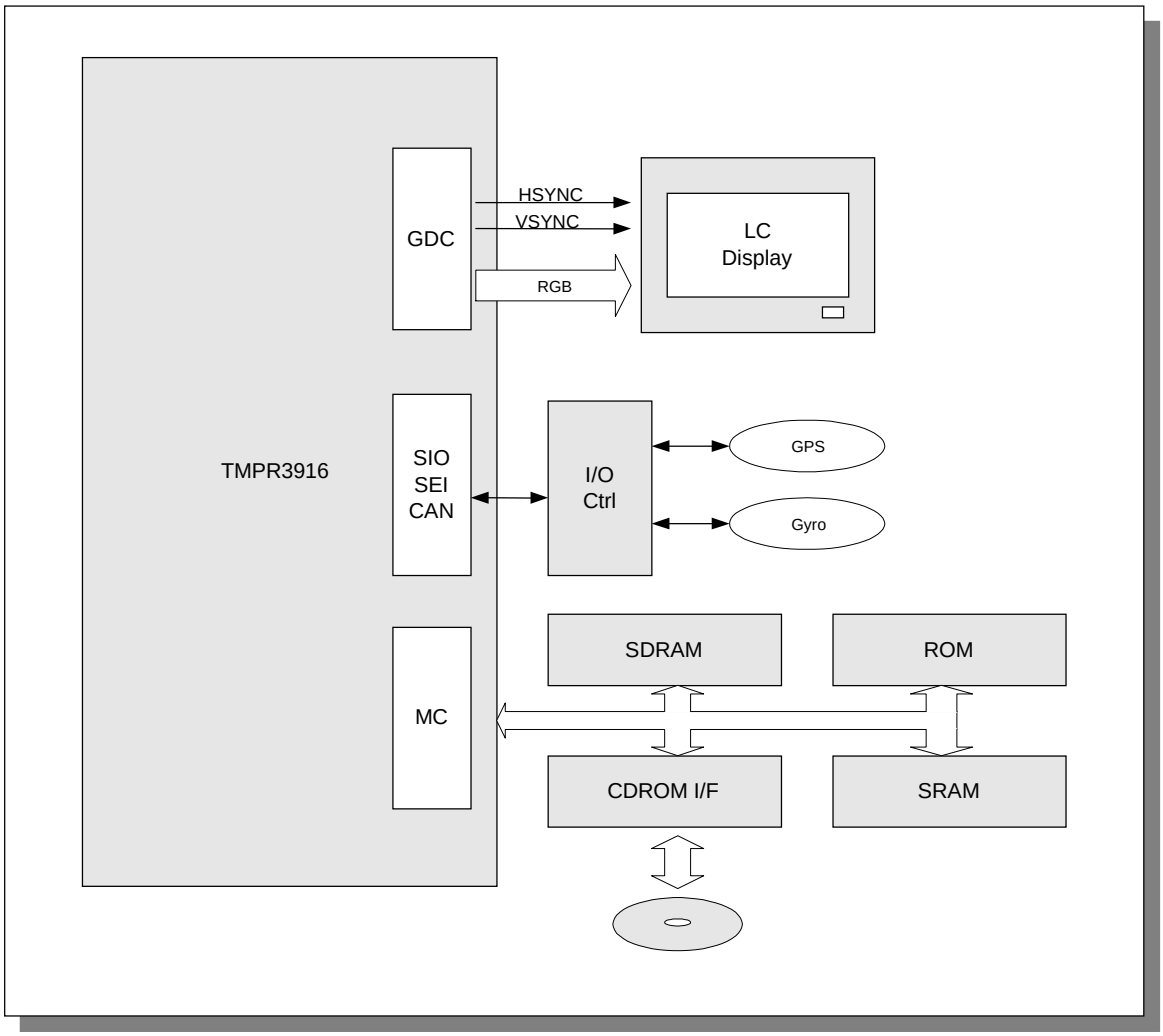


Figure 1.4.2 System Example Using TMPR3916

1.5 Address Map

The following table shows the memory map of TMPR3916.

Memory Area of TX39-CPU	Physical Address	Memory Device	Special Use in TMPR3916
Kernel Uncached/ Cached (kseg0, kseg1)	0x0000 0000	SDRAM, SRAM, ROM *	interrupt vector at 0x0000 0080
	0x1C00 0000	internal register	devices of TMPR3916
	0x1E00 0000	SDRAM, SRAM, ROM *	
	0x1FC0 0000	Boot ROM	start address after reset or NMI
Inaccessible	0x2000 0000		
User / Kernel Cached (kuseg)	0x4000 0000	SDRAM, SRAM, ROM *	
User / Kernel Uncached (kuseg - reserved)	0xBF00 0000		
Kernel Cached (kseg2)	0xC000 0000		
Kernel Uncached (kseg2 - reserved)	0xFF00 0000		

Figure 1.5.1 TMPR3916's Memory Map

* For SDRAM, SRAM or ROM shown in the above table, the software can define the address range of the connected memory devices. For further information see chapter "Memory Controller".

The following table shows the address ranges of the internal devices:

Address Range (physical address)	Address Range (virtual address)	Device
0x1C00_0000 .. 0x1C00_07FF	0xBC00_0000 .. 0xBC00_07FF	Asynchronous Serial Interface (UART)
0x1C00_8000 .. 0x1C00_FFFF	0xBC00_8000 .. 0xBC00_FFFF	Synchronous Serial Interface (TXSEI)
0x1C01_0000 .. 0x1C01_FFFF	0xBC01_0000 .. 0xBC01_FFFF	TIMER
0x1C02_0000 .. 0x1C02_7FFF	0xBC02_0000 .. 0xBC02_7FFF	Memory Controller (MEMC)
0x1C02_8000 .. 0x1C02_FFFF	0xBC02_8000 .. 0xBC02_FFFF	Memory Controller (SDRAMC)
0x1C03_0000 .. 0x1C03_FFFF	0xBC03_0000 .. 0xBC03_FFFF	Parallel Interface (PORT)
0x1C04_0000 .. 0x1C04_FFFF	0xBC04_0000 .. 0xBC04_FFFF	Interrupt Controller (INTC)
0x1C05_0000 .. 0x1C05_FFFF	0xBC05_0000 .. 0xBC05_FFFF	Graphic Display Controller (GDC)
0x1C06_0000 .. 0x1C06_FFFF	0xBC06_0000 .. 0xBC06_FFFF	Direct Memory Access Controller (DMAC)
0x1C07_0000 .. 0x1C07_7FFF	0xBC07_0000 .. 0xBC07_7FFF	CAN Module (TXCAN), channel 0
0x1C07_8000 .. 0x1C07_FFFF	0xBC07_8000 .. 0xBC07_FFFF	CAN Module (TXCAN), channel 1
0x1C08_0000 .. 0x1C08_FFFF	0xBC08_0000 .. 0xBC08_FFFF	Chip Configuration Register (CCR)

Figure 1.5.2 Physical and Virtual Addresses for Internal Devices

Note: Please note that addresses seen on GBUS are physical. Therefore virtual addresses can only be used in program code and will be translated before being output to the bus.

1.6 Clocks

The TMPR3916 incorporates an eight-times PLL **clock generator**. Connect a crystal oscillator with 1/8 the frequency of the processor clock (processor clock = TX39 core input clock frequency). To reduce power dissipation and simplify system design, the TMPR3916 can control the TX39 core operating frequency and the bus operation reference frequency.

Clock Types:

- **Master clock**

  Master clock regulates the TMPR3916 operations. The clock is eight times the frequency of the external crystal oscillator.

- **Processor clock**

  This clock is used for TMPR3916 processor core operations. It has the same frequency as the master clock. (When using this clock, set reduced frequency indicator RF[1:0] of the core configuration register to 00. The processor clock will not operate if RF is set to any other value.)

- **System clock**

  The system clock regulates the TMPR3916 bus operations. It is generated from the processor clock and is of the same frequency and phase. This clock is output to pin SYSCLK.

Setting the CLKEN pin to low **stops all clocks** of the device. The SYSCLK pin is set to high in this state and power-consumption is reduced to a minimum. The processor can resume its function immediately after the CLKEN pin has been asserted. For further information see chapter “Electrical Characteristics”.

1.7 Resets

Setting RESET* = Low resets the TMPR3916.

RESET* should be held Low for at least 10 cycles of system clock (SYSCLK). Because the RESET* signal is synchronized with the TMPR3916 internal clock, the RESET* signal can be set asynchronously to system clock.

At a reset the TMPR3916 will do the following operations:

- Pipeline will be stalled, internal states reset.
- The valid and lock bits of the TX39 cache will be cleared.

During reset period the output signals have the following states:

A[31:2]	=	undefined
D[31:0]	=	undefined
BE[3:0]*	=	“High”
RD*, WR*	=	“High”
BURST*	=	“High”
LAST*	=	“High”
SYSCLK	=	continues outputting clock

1.8 Time-Out-Error Control Unit

This unit is a kind of watchdog unit for the internal CPU-bus.

When a master sends a GBSTART signal, the Time-Out Error Control Unit starts counting cycles. If no reaction has been detected on the bus, an internal acknowledging GACK is generated after 1024 cycles so that the bus is free for interaction again.

1.9 Operating Modes of TMPR3916

In **Normal Mode**, the TX39 core and peripheral circuits operate at maximum frequency.

Halt Mode halts the core operations and reduces power dissipation by stopping the clock in the TX39 core.

To switch to Halt mode, set the Halt bit of the Configuration Register in TX39 core.

In Halt mode, the TX39 core holds the status of the pipeline processing and stops the core operations. The write buffer does not stop. If data remains in the write buffer when Halt mode is selected, write operations continue until the write buffer becomes empty. Also SYSCLK does not stop.

The processor is released from HALT mode by using the NMI* signal, RESET* signal or by any kind of enabled interrupt. The corresponding exception handler is executed after the HALT mode has been released.

Doze Mode halts some TX39 core operations and reduces power dissipation. Unlike Halt mode, only some clocks in the processor core stop, allowing external bus release requests to be received. Also the peripheral blocks continue operating normally in Doze mode.

To switch to Doze mode, set the Doze bit of the configuration register in TX39 core.

Standby Mode halts the clock generator PLL circuit operation and reduces power dissipation.

First, set CLKEN pin to low to stop the clock supply. Then, set PLLOFF* pin to low to halt the PLL circuit operations.

1.10 Chip Configuration Register (CCR)

The configuration register is used to configure chip functions concerning more than one module.

Bit	31	30	29	28	27	26	25	24
Name	—						VIEWDAC	SFB

Bit	23	22	21	20	19	18	17	16
Name	—		CANM			CANDIV		

Bit	15	14	13	12	11	10	9	8
Name	—			SEIMUX	—		TOE	BEOW

Bit	7	6	5	4	3	2	1	0
Name	DMA1CC				DMA0CC			

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25	VIEWDAC	By using this bit it is possible to power down the VIEWDAC. 1 = Enables the VIEWDAC (default) 0 = Disables the VIEWDAC	1	R/W
24	SFB	SRAM Frame Buffer: 0 = Display frame is stored in SDRAM 1 = Display frame is stored in SRAM	0	R/W
23:22	—	Wired to zero	00	R
21:20	CANM	CAN operation mode 00 = Normal mode X1 = Internal test mode 10 = 2 internal CAN on one TX/RX pair, 1 Transceiver	00	R/W
19	—	Wired to zero	0	R
18:16	CANDIV	The CANDIV bits set the clock divider for the CAN modules. The following table shows possible settings and the corresponding divider ratios. 000 = Invalid setting 001 = System clock divided by 2 010 = System clock divided by 3 (default) 011 = System clock divided by 4 100 = System clock divided by 5 101 = System clock divided by 6 110 = System clock divided by 7 111 = System clock divided by 8	010	R/W
15:13	—	Wired to zero	0	R
12	SEIMUX	Determines whether the TXSEI or the UART use pins PIO16 to PIO29. 0 = The UART uses pins 1 = The TXSEI uses pins	1	R/W
11:10	—	Wired to zero	0	R
9	TOE	Time-Out Error Control The time-out error counter aborts bus transactions with exception after 1024 cycles, if they are not responded to. 0 = No time-out on internal bus 1 = Abort not responded access on internal bus	1	R/W

Bit	Name	Function	Reset Value	R/W
8	BEOW	Bus Error on Write This bit determines, if a write transaction on internal bus will aborted, when no device responds after 1024 cycles. 0 = No bus-error on time-out at write 1 = Generate bus-error on time-out at write	1	R/W
7:4	DMA1CC	These bits are used to select devices for DMA transfers of DMA channel 1. 0000 = UART0, Transmission 0001 = UART0, Reception 0010 = UART1, Transmission 0011 = UART1, Reception 0100 = UART2, Transmission 0101 = UART2, Reception 0110 = UART3, Transmission 0111 = UART3, Reception 1000 = TXSEI, Transmission 1001 = TXSEI, Reception 1010 = External Device 1111 = No device selected (reset value) Other settings are invalid	1111	R/W
3:0	DMA0CC	These bits are used to select devices for DMA transfers of DMA channel 0. The settings are similar to DMA1CC.	1111	R/W

2. Memory Controller (MC)

This system's memory controller consists of two modules: the SDRAM Controller (SDRAMC) and MEMC for other types of memory.

Six multi-purpose memory channels can be administrated: While the channels 0 and 1 are assigned to the SDRAM controller, the memory controller's channels are numbered from 2 to 5.

The SDRAM Controller contains the following features:

- uses memory architecture single-data-rate SDRAM
- 2 memory channels with 32 bit width, 16 bit width connectivity is not supported for SDRAM devices
- base address, mask, DRAM size & organization configurable for each channel (same physical address space on both Video and System-Busses)
- true dual G-Bus connectivity
- read bursts 4, 8, 16, 32 words,
- single read accesses, single write accesses
- different, mixed burst sizes on both busses are possible
- fair memory arbitration, predictable latency
- memory arbitrated on a first come first serve basis
- for the case of a simultaneous transaction request, the prioritized G-Bus can be configured
- one word write-back buffer to reduce bus utilization during write operations
- low-power / self-refresh mode supported
- background refresh during MEMC accesses
- built-in power-up logic
- programmable refresh cycle

The MEMC contains the following features:

- 4 separate channels
- Support for ROM, MASK ROM, PAGE MODE ROM, EPROM, EEPROM, SRAM, and FLASH devices
- Support for Page-Mode
- Base Address and size programmable per channel
- external Acknowledge mode for external ASIC slave device connectivity
- Data Bus width of 16-bit/32-bit is selectable by channel
- Supports programmable Setup and Hold Time for Address, Chip Enable, Write Enable Signals
- Channel 5 supports BOOT options

2.1 Structure of Memory Controller

TMPR3916 owns two internal busses, the system bus and the video bus. The system bus can be accessed by the TX39 core and other devices capable of being master on the bus. Additionally, the TMPR3916 provides a second, so called video bus. The video bus is used by the GDC in order to read picture data from the SDRAM frame buffer. The GDC is the only device on that bus and is not able to write data into SDRAM. It is only possible to write data into the frame buffer via the system bus. Due to system performance considerations it is recommended to use SDRAM memory for frame buffer, though it is also possible to locate the frame buffer in memory devices accessible by the MEMC. For this purpose the TMPR3916 provides a bridge between system and video bus. In order to activate this bridge the SFB bit in Chip Configuration Register (CCR) needs to be asserted. It has to be assured that the GDC is not accessing SDRAM mapped memory in this case. In this mode, the system-performance is restricted by the higher utilization of the system bus and the reduced throughput to the frame buffer memory devices.

The following figure shows the structure of the memory controller:

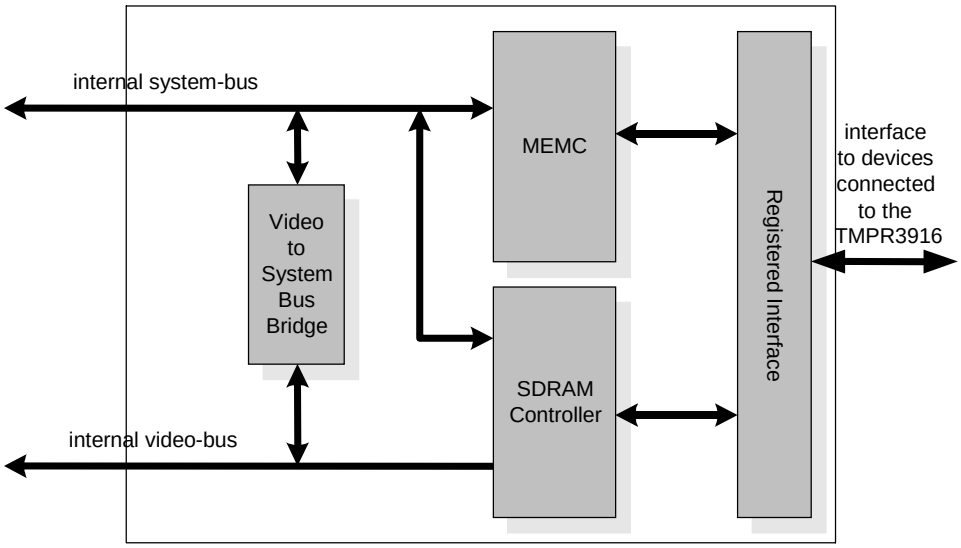


Figure 2.1.1 Data Flow in Memory Controller

2.2 Example Memory Configuration

The following figures show examples how to connect different devices to the TMPR3916. It is possible to have a mixture of different kinds of memories because timing & device configurations are programmable for each channel in the MC.

Keep in mind that additionally a boot device must always be connected to channel 5 of the memory controller!

Asynchronous SRAM connected to the TMPR3916 (16 bit data width):

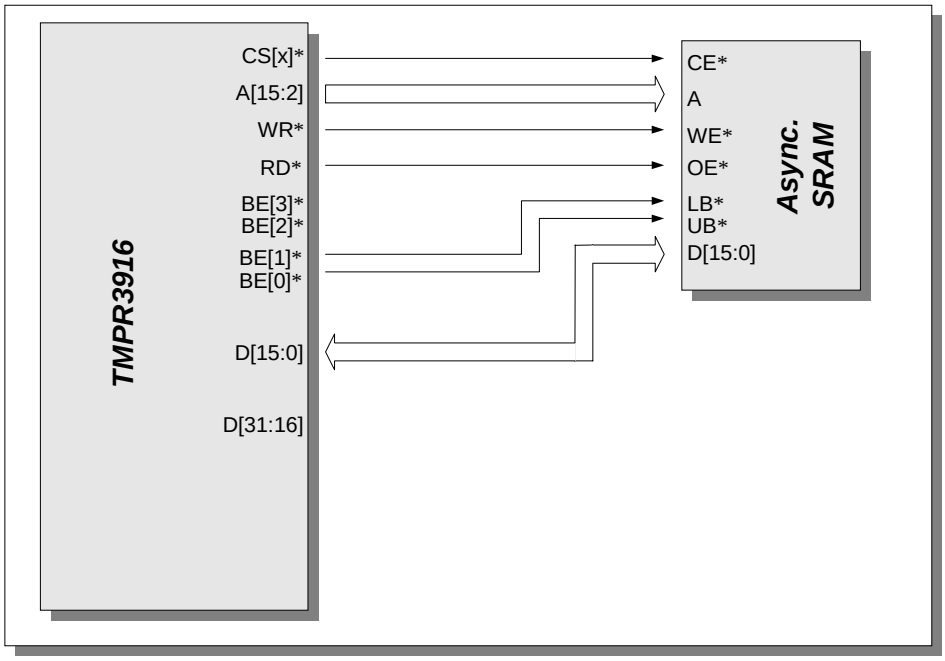


Figure 2.2.1 16-Bit Asynchronous SRAM Connected to TMPR3916

Two asynchronous SRAMs connected to one channel of the TMPR3916 :

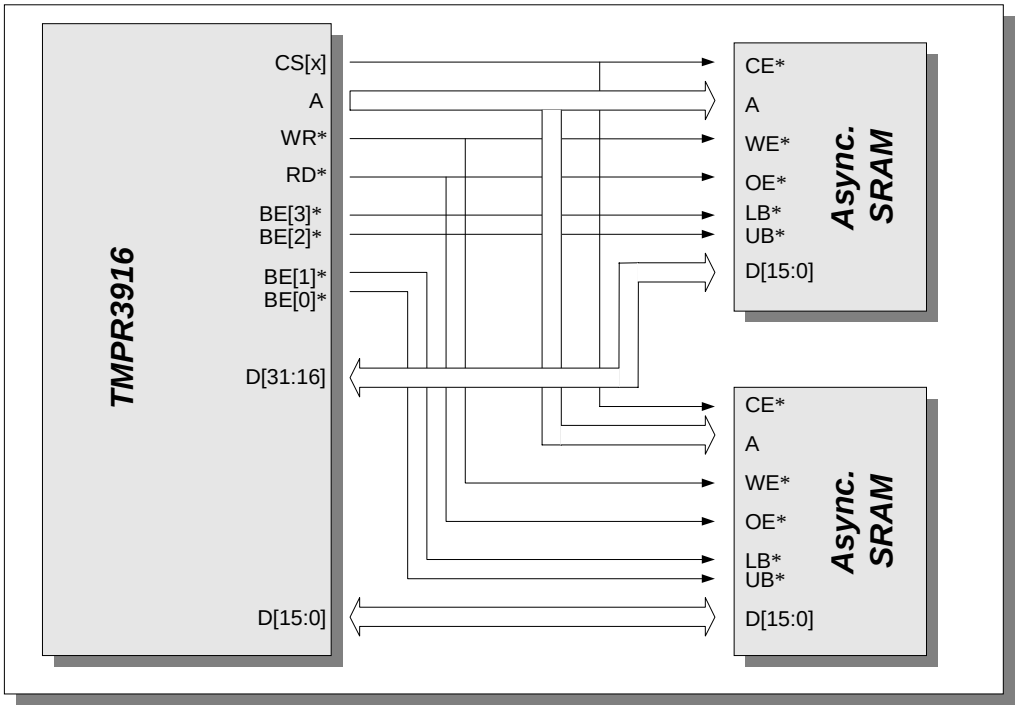


Figure 2.2.2 2 × 16-Bit Asynchronous SRAM Connected to One 32-Bit Memory Channel

Connection of an external slave device with a data width of 16 bit :

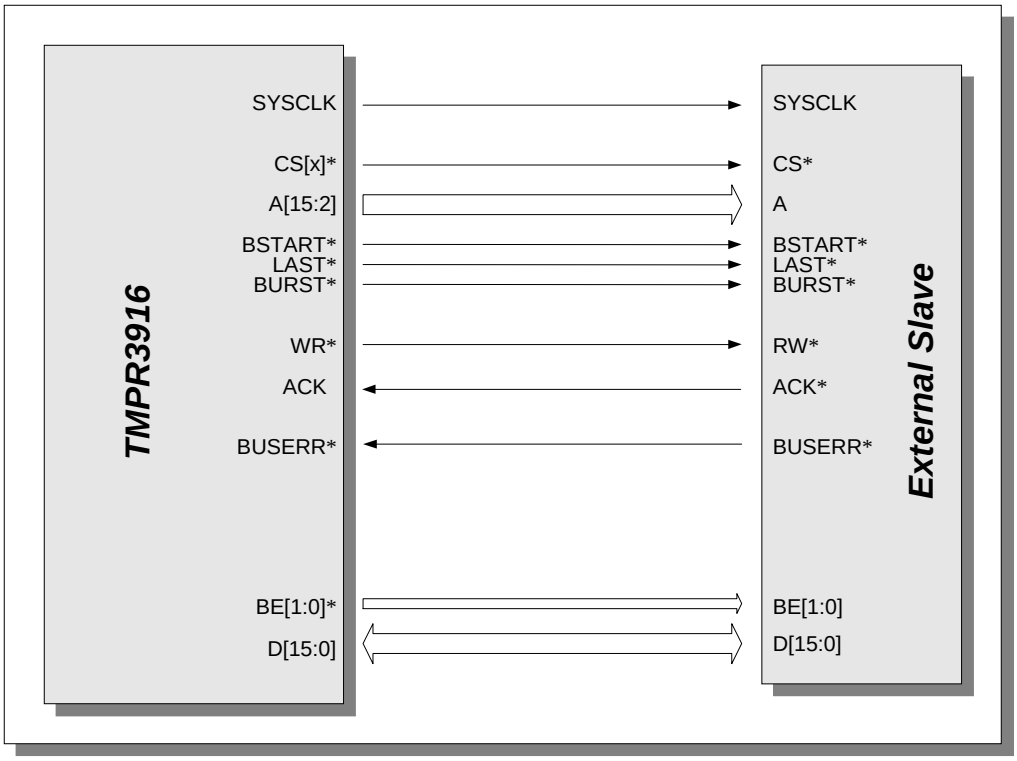


Figure 2.2.3 16-Bit External Slave Device Connected to Memory Channel

Connection of an external slave device with a data width of 32 bit :

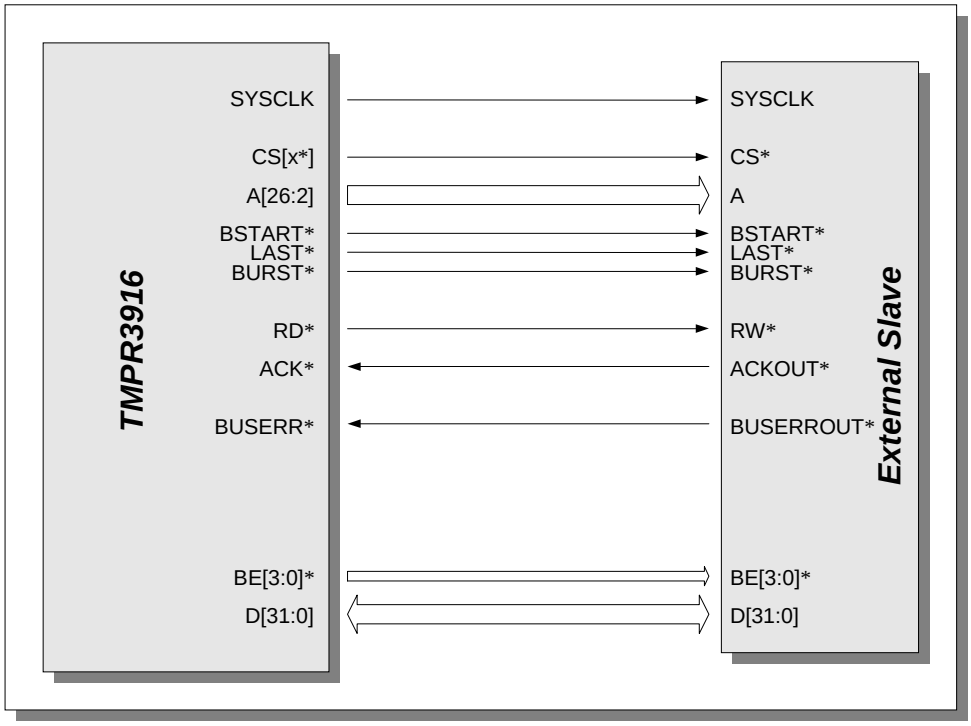


Figure 2.2.4 32-Bit External Slave Device Connected to Memory Channel

The following figure shows the connection of a 32 bit width 16Mbit SDRAM device to the TMPR3916. As can be seen on the chip select signal connectivity (CS0) the device is accessed via channel 0 of the SDRAM Controller.

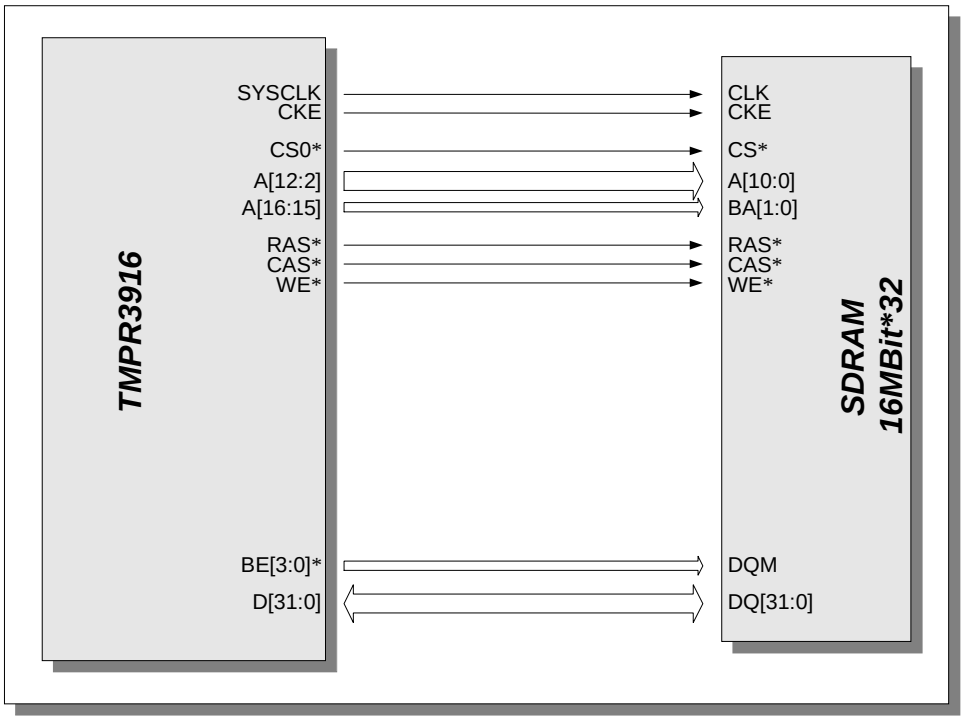


Figure 2.2.5 16 Mbit × 32 SDRAM Device Connected to Memory Channel

2.3 Ports of Memory Controller

The SDRAM controller externally connects up to two channels of SDRAM memory. Each of the two SDRAM controller channels has its own organization register and can support different device sizes and organization. The timing settings are shared for both channels and must be the same for both channels. Refresh is conducted on both channels in parallel.

The memory channels share the memory data, address and control busses with the exception of the chip-select signal, which is wired individually for each channel. Each channel must be connected with a width of 32 bits. Devices with organizations of 4, 8, 16 and 32 bit width can be used as long as the connected devices jointly form a 32 bit channel. The following table summarizes the externally connected memory signals.

Signal	Type	Description
SYSCLK	OUT	DRAM Clock
A[14:2]	OUT	Row/Column Address Bus, connect to A at the SDRAM device
A[16:15]	OUT	Bank Address Bus, connect to BA at the SDRAM device
RAS*	OUT	Row Access Strobe signal
CAS*	OUT	Column Access Strobe signal
WE*	OUT	Write Enable Signal
CKE	OUT	Clock Enable Signal for SDRAM
CS*[1:0]	OUT	Chip Select Signal, one for each SDRAM channel CS[0] => channel X CS[1] => channel Y
D[31:0]	IN/OUT	Data Bus, connect to DQ at the SDRAM device
BE[3:0]	OUT	Output Mask, connect to DQM at the SDRAM device

2.4 Registers

The following registers are used for configuration and operation of the TMPR3916 memory channels 2 to 5 via MEMC. For channels 0 and 1 (SDRAM) refer to section 2.5.4.

Device	Register (short name)	*Physical Address (hex)*	Function
MEMC	RCCR2	*1C02 0008H*	ROM Control Register Channel 2
	RCCR3	*1C02 000CH*	ROM Control Register Channel 3
	RCCR4	*1C02 0010H*	ROM Control Register Channel 4
	RCCR5	*1C02 0014H*	ROM Control Register Channel 5
SDRAM	DCCR	*1C02 8000H*	SDRAM Configuration Register
	DCBA	*1C02 8004H*	SDRAM Base Address Register
	DCAM	*1C02 8008H*	SDRAM Address Mask Register
	DCTR	*1C02 800CH*	SDRAM Timing Register

2.5 SDRAMC Functions

2.5.1 Address Translation

The table below shows the different memory organizations that have been considered during the controller design and that can be used for each of the two memory channels.

Channel Size (for one channel)	Mapped G-Bus Address Bits	Toshiba Device Number	Organizations (Devices × Banks × Rows × Columns × Bits)
8 Mbyte	[22:2]	TC59S6432	1 × 4 × 2 K × 256 × 32 (SDRAM, 64 Mbit)
16 Mbyte	[23:2]	TC59S6416	2 × 4 × 4 K × 256 × 16 (SDRAM, 64 Mbit)
32 Mbyte	[24:2]	TC59S6408 TC59SM716	4 × 4 × 4 K × 512 × 8 (SDRAM, 64 Mbit) 2 × 4 × 4 K × 512 × 16 (SDRAM, 128 Mbit)
64 Mbyte	[25:2]	TC59S6404 TC59SM708 TC59SM816	8 × 4 × 4 K × 1 K × 4 (SDRAM, 64 Mbit) 4 × 4 × 4 K × 1 K × 8 (SDRAM, 128 Mbit) 2 × 4 × 8 K × 512 × 16 (SDRAM, 256 Mbit)
128 Mbyte	[26:2]	TC59SM704 TC59SM808	8 × 4 × 4 K × 2 K × 4 (SDRAM, 128 Mbit) 4 × 4 × 8 K × 1 K × 8 (SDRAM, 256 Mbit) 2 × 4 × 8 K × 1 K × 16 (SDRAM, 512 Mbit)
256 Mbyte	[27:2]	TC59SM804	8 × 4 × 8 K × 2 K × 4 (SDRAM, 256 Mbit) 4 × 4 × 8 K × 2 K × 8 (SDRAM, 512 Mbit)

Therefore, SDRAMC supports the following memory organizations:

Banks: 2 or 4
Rows: 2 K, 4 K, 8 K
Columns: 256, 512, 1 K, 2 K

The following table shows address translation of the G-Bus address for supported memory configurations.

Organization Column Size	Organization Row Size	Organization Bank Size	Address Mapping ColAddr	Address Mapping RowAddr	Address Mapping BankAddr
256	2 K	2 4	GAO[9:2]	GAO[20:10]	GAO[21] GAO[22:21]
	4 K	2 4	GAO[9:2]	GAO[21:10]	GAO[22] GAO[23:22]
	8 K	2 4	GAO[9:2]	GAO[22:10]	GAO[23] GAO[24:23]
512	2 K	2 4	GAO[10:2]	GAO[21:11]	GAO[22] GAO[23:22]
	4 K	2 4	GAO[10:2]	GAO[22:11]	GAO[23] GAO[24:23]
	8 K	2 4	GAO[10:2]	GAO[23:11]	GAO[24] GAO[25:24]
1024	2 K	2 4	GAO[11:2]	GAO[22:12]	GAO[23] GAO[24:23]
	4 K	2 4	GAO[11:2]	GAO[23:12]	GAO[24] GAO[25:24]
	8 K	2 4	GAO[11:2]	GAO[24:12]	GAO[25] GAO[26:25]
2048	2 K	2 4	GAO[12:2]	GAO[23:13]	GAO[24] GAO[25:24]
	4 K	2 4	GAO[12:2]	GAO[24:13]	GAO[25] GAO[26:25]
	8 K	2 4	GAO[12:2]	GAO[25:13]	GAO[26] GAO[27:26]

Note: GAO (G-Bus Address Output) refers to the physical address of the internal system bus.

2.5.2 SDRAM Controller Function / Bank Interleaving

The SDRAMC module functions as an advanced DRAM controller for synchronous DRAM. The TMPR3916 SDRAMC is “advanced” compared to regular SDRAM controllers in the way that it is capable of handling two different MCU busses at the same time.

SDRAM memory devices are usually organized in 4 different banks. All of these banks contain a high-speed static RAM memory buffer, which allows fast sequential access once a DRAM memory row has been sensed into these buffers. The sensing of the DRAM memory row and the rewriting of the sensed row are major contributors to the total duration of each memory access.

The following picture shows the benefits of the dual-bus structure:

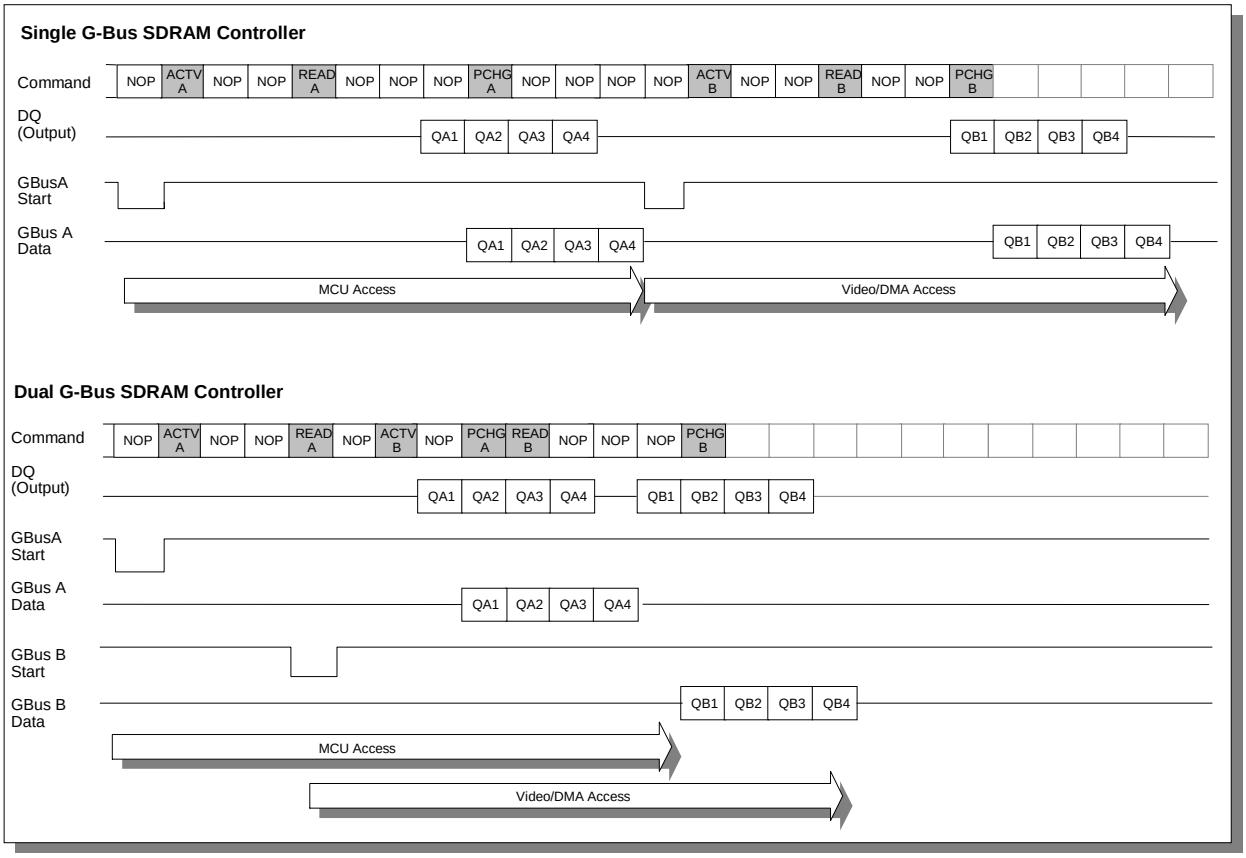


Figure 2.5.1 Benefits of Dual-Bus Structure

A regular SDRAM memory controller with only one MCU-bus interface will connect both the MCU and the video-controller on one bus. By transferring the bus-ownership both components access the external DRAM memory. Typically, the SDRAM controller is able to work with only one address at the same time. It will not know the address for the next access before the previous transfer has been completed.

A dual G-Bus memory controller always knows the state and address of both busses. Therefore, it is possible to work with both addresses at the same time.

The dual G-Bus structure can only be effectively used, if the following condition is met:

The two busses should utilize either different banks of the same channel or two different channels.

This means for example that the frame-buffer for the video-controller and the program memory accessed by the MCU should be located in different memory banks. If the condition is not met for two consecutive addresses, the controller needs to precharge the bank first and it has to wait for the precharge-to-active-latency before it can start with the access on the second MCU-bus. A memory throughput gain of up to 40 % can be achieved, if software allows the simultaneous utilization of several SDRAM banks.

2.5.3 Self-Refresh Mode

The SDRAM controller also offers support for the SDRAM self-refresh-mode. In self-refresh mode, the connected SDRAM devices maintain data retention without clock-supply and without the requirement of auto refresh cycles. The self-refresh mode significantly lowers the power consumption of the connected memory devices.

Self-refresh mode is entered with the assertion of the SRM bit in the DCCR register. The connected SDRAM channels and the SDRAM controller need to be enabled during this operation.

Upon assertion of this flag, SDRAMC will enter the self-refresh-mode by issuing a self-refresh-entry command. SDRAMC will remain in self-refresh mode until the deassertion of the SRM flag. The SDRAMC logic guarantees a minimum time of APL latency cycles in self-refresh mode.

The SRM flag can be cleared in two different ways, by either writing the bit to “0” or by conducting an access to mapped memory space.

After clearing the SRM flag, the CKE signal is asserted to exit the self-refresh mode. The SDRAM controller issues NOP commands for the number of cycles specified by the ARL setting. After this, a regular auto refresh cycle will be conducted to finalize the sequence.

2.5.4 Configuration Registers

Overview

SDRAMC is configured using a total of four configuration registers. These configuration registers reside in the SDRAMC configuration area. The configuration registers can be accessed using byte, half-word and word type accesses. Following, the two memory channels are numbered X (CS0) and Y (CS1). Each channel can be set up using its own memory configuration. The registers RSX, CSX and BSX are used to set up the configuration for memory channel X. The registers RSY, CSY and BSY are used to setup the configuration for memory channel Y.

Control Register (DCCR)

Bit	31	30	29	28	27	26	25	24
Name								

Bit	23	22	21	20	19	18	17	16
Name	RSX		CSX		BSX	—	ACCD	SRM

Bit	15	14	13	12	11	10	9	8
Name	RSY		CSY		BSY	—		

Bit	7	6	5	4	3	2	1	0
Name	—		CEN		PRIO	—	ENA	PWR

Bit	Name	Function	Reset Value	R/W
31:24	—	Wired to zero	0	R
23:22	RSX	Row Size for channel X The RSX, CSX and BSX fields are used to specify the organization of memory channel X. 00 = 2 K 01 = 4 K 10 = 8 K 11 = Invalid setting	00	R/W
21:20	CSX	Column Size for channel X Number of Columns 00 = 256 01 = 512 10 = 1 K 11 = 2 K	00	R/W
19	BSX	Bank Size for channel X Number of Banks 0 = 2 banks 1 = 4 banks	0	R/W
18	—	Wired to zero	0	R
17	ACCD	Acceleration Disable Disabling dual G-Bus accelerations significantly lowers system performance. Therefore, this mode is meant to be used as fault mode only. 0 = All Accelerations Enabled 1 = Prevents Dual G-Bus Accelerations	0	R/W

Bit	Name	Function	Reset Value	R/W
16	SRM	Self Refresh Mode This register is used to enter and exit the SDRAM self-refresh mode. Self-Refresh mode will automatically be entered, by writing this register to 1. Self-Refresh mode is exited, by writing SRM with a value of 0 or by conducting a read or write access to the SDRAM mapped memory region. CEN or ENA may not be deasserted during self-refresh mode. If self-refresh mode is not utilized, initialize SRM to 0.	0	R/W
15:14	RSY	Row Size for channel Y The RSY, CSY and BSY fields are used to specify the organization of memory channel Y. 00 = 2 K 01 = 4 K 10 = 8 K 11 = Invalid setting	00	R/W
13:12	CSY	Column Size for channel Y Number of Columns 00 = 256 01 = 512 10 = 1 K 11 = 2 K	00	R/W
11	BSY	Bank Size for channel Y Number of Banks 0 = 2 banks 1 = 4 banks	0	R/W
10:6	—	Wired to zero	0	R
5	CEN1	Memory Channel Enable Channel Y 1 = Enable address decoding for this channel 0 = Disable address decoding for this channel This setting enables and disables the address decoding for the memory channel. A memory channel cannot be accessed, if it is disabled using this bit and G-Bus transactions within the channel's address range will not be answered.	0	R/W
4	CEN0	Memory Channel Enable Channel X 1 = Enable address decoding for this channel 0 = Disable address decoding for this channel This setting enables and disables the address decoding for the memory channel. A memory channel cannot be accessed, if it is disabled using this bit and G-Bus transactions within the channel's address range will not be answered.		R/W
3	PRIO	G-Bus Priority Setting This bit can be used to prioritize the access of one G-Bus. However, this bit has only a minor impact. In the scenario, the memory controller is in idle state when both G-Busses simultaneously start a G-Bus transaction: 1 = G-Bus X will be serviced with priority 0 = G-Bus Y will be serviced with priority A real prioritization of one G-Bus is not efficient in terms of performance, since it is not possible to utilize the dual G-Bus structure under these circumstances. Additionally, the latency of one G-Bus would become unpredictable.	0	R/W
2	—	Wired to zero	0	R

Bit	Name	Function	Reset Value	R/W
1	ENA	ENA – Module Enable 1 = Enable Module 0 = Disable Module If SDRAMC is disabled the state-machine of the SDRAM controller is suspended and the controller will remain in the idle state. This also means that connected SDRAMs will not be refreshed and that their contents will be lost. The module should never be disabled during its normal operation. It has to be ensured, that no SDRAM accesses are performed on one of the G-Busses when the module is disabled. For example, SDRAMC should never be disabled in a situation, where read accesses are still performed on the video bus. It is mandatory to disable the GDC before SDRAMC is being disabled. Software should also ensure, that the internal write-buffer has been written to SDRAM memory. Violations to these rules might result in bus hang-ups.	0	R/W
0	PWR	PWR – Power-Up Sequence Setting this bit will run the power-up sequence for all connected and enabled memory devices. The power-up sequence is described more closely in the next chapter. This flag can be read to determine the end of the power-up procedure. At the end of the power-up sequence, this bit is automatically cleared by the SDRAM controller state-machine.	0	R/W

Base Address Register (DCBA)

Bit	31							22	21			16
Name	BAX								—			

Bit	15							6	5			0
Name	BAY								—			

Bit	Name	Function	Reset Value	R/W
31:22	BAX	Base Address Memory Channel X This register is used to define the base address for memory channel X. This address is compared to the upper 10 bits of the physical G-Bus address.	0	R/W
21:16	—	Wired to zero	0	R
15:6	BAY	Base Address Memory Channels Y The same as for the memory channel X.	0	R/W
5:0	—	Wired to zero	0	R

Address Mask Register (DCAM)

Bit	31							22	21			16
Name	AMX								—			

Bit	15							6	5			0
Name	AMY								—			

Bit	Name	Function	Reset Value	R/W
31:22	AMX	Address Mask Memory Channel X This field is used to mask the base address for memory bank X. 1 = Bit is taken into account during comparison 0 = Bit is don't care for the comparison The base address mask is used to select the memory part of an appropriate size. It is also possible to mirror DRAM memory parts or to protect memory parts using this register.	0	R/W
21:16	—	Wired to zero	0	R
15:6	AMY	Address Mask Memory Channels Y The same as for memory channels X.	0	R/W
5:0	—	Wired to zero	0	R

Timing Register (DCTR)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RFC														WRL	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	APL			RASL			CASL			PAL			ARL			

Bit	Name	Function	Reset Value	R/W
31:18	RFC	Refresh Counter Reload Value This field provides the reload value for the internal refresh counter. The refresh counter is implemented as a 14 bit down counter, which reloads and issues a refresh request, every time a value of 0 is reached. This Settings depends on the specified refresh cycle time and number of rows of the connected memory devices: $Register\ Value = \frac{Refresh\ Time \quad Operation\ Frequency}{Number\ of\ Rows}$ (Example settings at 60 MHz, counter values / settings in Hex) Time 2 K Row 4 K Row 8 K Row 32 ms 0x03A9 0x01D4 0x00EA 64 ms 0x0752 0x03A9 0x01D4 128 ms 0x0EA4 0x0752 0x03A9	0x1000	R/W

Bit	Name	Function	Reset Value	R/W
17:16	WRL	Write Recovery Latency This setting defines the minimum number of cycles, from the point where the last word has been written until the “precharge” command is issued. This setting has to match the write recovery time (t_{WR}) of the selected SDRAM. 00 = 1 cycle 01 = 2 cycles 10 = 3 cycles 11 = 4 cycles	11	R/W
15:13	APL	Active to Precharge Latency This field and the associated timer is used to ensure the Active to Precharge Latency (t_{RAS}). The register value is used as reload value for an internal down counter. PCHG Latency 000 ..010 = Invalid setting 011 = 3 cycles 100 = 4 cycles 101 = 5 cycles 110 = 6 cycles 111 = 7 cycles	111	R/W
12:10	RASL	RAS Latency This timeframe has to safely match the t_{RCD} time of the selected SDRAM device. The setting defines the number of cycles between providing the row address and the column address. This setting is used as reload value for a 3 bit down counter. 010 = 2 cycles 011 = 3 cycles 100 = 4 cycles 101 = 5 cycles Others = Invalid setting	011	R/W
9:7	CASL	CAS Latency This setting defines the number of clock cycles from the time that the column address is provided (READ, RDA / WRITE, WRA) until the first data is taken/output from/on the SDRAM data bus. During the power-up sequence, this value is programmed into the mode register as it is. For the SDRAM controller this setting is used as a reload value for an internal down counter. CAS latencies of less than 2 cycles are not supported! 010 = 2 cycles 011 = 3 cycles Others = Invalid settings	011	R/W
6:4	PAL	Same Bank Precharge -> Active/Refresh Latency This setting defines the minimum number of cycles from the point where the bank is precharged until it is reused (Activate, Refresh). This timeframe is defined by the t_{RP} time of the selected memory device. This time will not be taken, if a different bank or a different memory device is accessed. 010 = 2 cycles 011 = 3 cycles 100 = 4 cycles 101 = 5 cycles 110 = 6 cycles 111 = 7 cycles Others = Invalid settings	111	R/W

Bit	Name	Function	Reset Value	R/W
3:0	ARL	Auto Refresh To Next Command Latency This setting defines the number of clocks, which are taken after an Auto Refresh Command has been issued. This setting is used as a reload value for a down counter. The counter will prevent any active SDRAM commands until it is expired. 0000 = Invalid setting 0001 = 3 cycles 0010 = 4 cycles 0011 = 5 cycles … … 1111 = 17 cycles	111	R/W

2.5.5 Software Power-Up Sequence

The picture below shows the software flow that is necessary to initialize the module.

The SDRAM memory typically requires to be held in reset or disabled state until the internal circuits have stabilized. Software has to ensure that this state is applied for the specified amount of time by not enabling the SDRAM controller during this timeframe. After the configuration registers have been sequentially configured, a power-up cycle needs to be scheduled by writing a “1” value to the PWR bit in the DCCR register. Finally, the SDRAM controller and the utilized channels needs to be enabled to start the initialization. The PWR bit can be read to determine the end of the Power-Up sequence. During the power-up procedure, this flag will be read as “1”. It is cleared, when the power-up procedure completes.

Software Flow of Initialization:

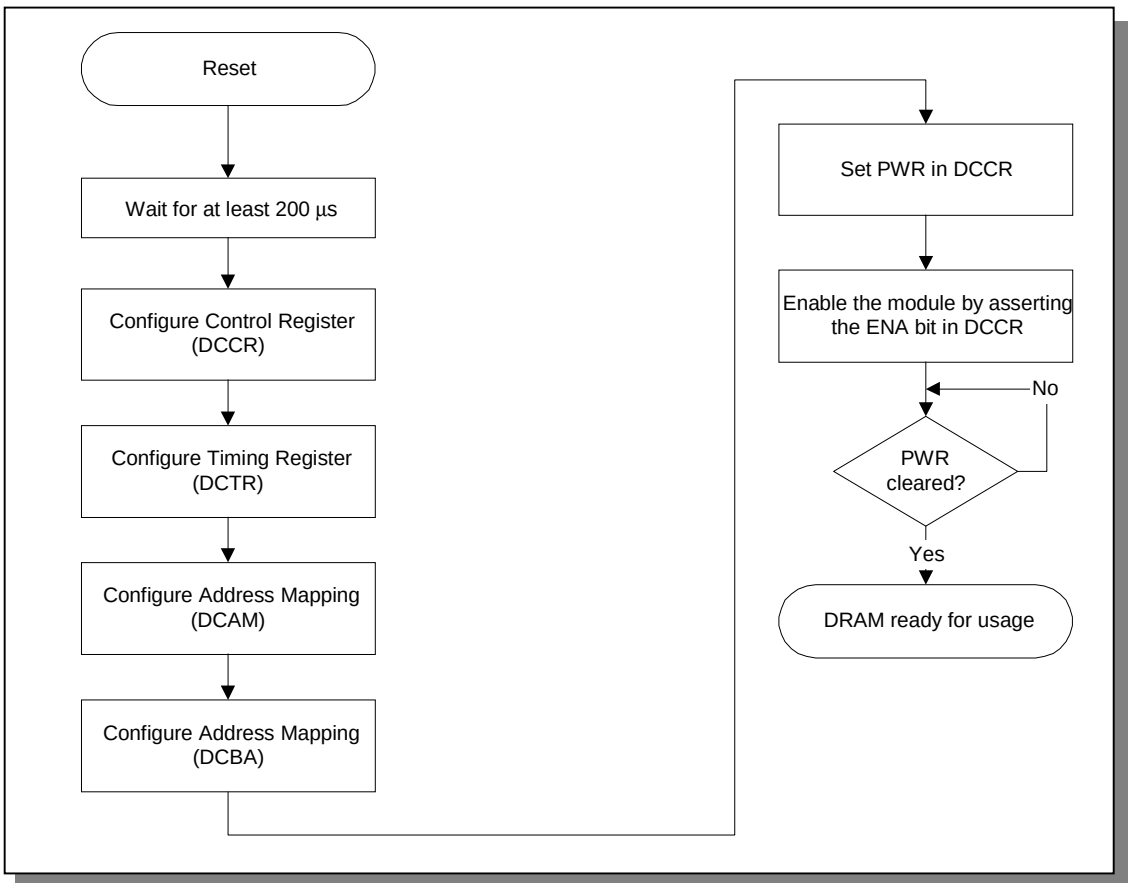


Figure 2.5.2 SDRAM Power-up Software Sequence Flow Chart

2.5.6 Address Mask Configuration

The following example provides a step by step approach that shows how to configure base-address and address-masks for a memory channel.

For the example it is assumed that two 128 MBit SDRAM devices with 16 Bit connectivity are connected to one of the SDRAM Controller channels.

To configure the address mask proceed in the following order:

1. Select a base-address for the memory area to be mapped.
   In this example, we choose 0x5000_0000 as the physical base address for the memory area to be mapped.
2. Determine how much memory is connected to the memory channel:
   In this example 2 devices with 16 MBytes each => 32 MByte
3. Determine how many address bits are required to address 33.554.432 Bytes
   log2 (33554432) = 25
   This means that 25 address bits are required for the channel

Therefore, the correct address mask is 0xfe00.

In this case, the memory area from 0x5000_0000 to 0x51ff_ffff will be mapped to the SDRAM channel. The SDRAM controller will respond only to bus accesses that refer to an address within this area.

Starting from this setup also mirror areas and protected areas can be generated:

<u>Example for mirroring:</u>

An address mask 0x7e00 will map the same physical memory twice to the addresses:

0x5000_0000 - 0x51ff_ffff and

0xD000_0000 - 0xD1ff_ffff

<u>Example for protection:</u>

An address mask 0xff00 will leave the memory area

0x5000_0000 - 0x50ff_ffff accessible, while the second half

0x5100_0000 - 0x51ff_ffff is protected.

Please note that the SDRAM always refers to physical address space, which might differ from the virtual address space used within a computer program.

Timing Diagrams

Basic Read Timing:

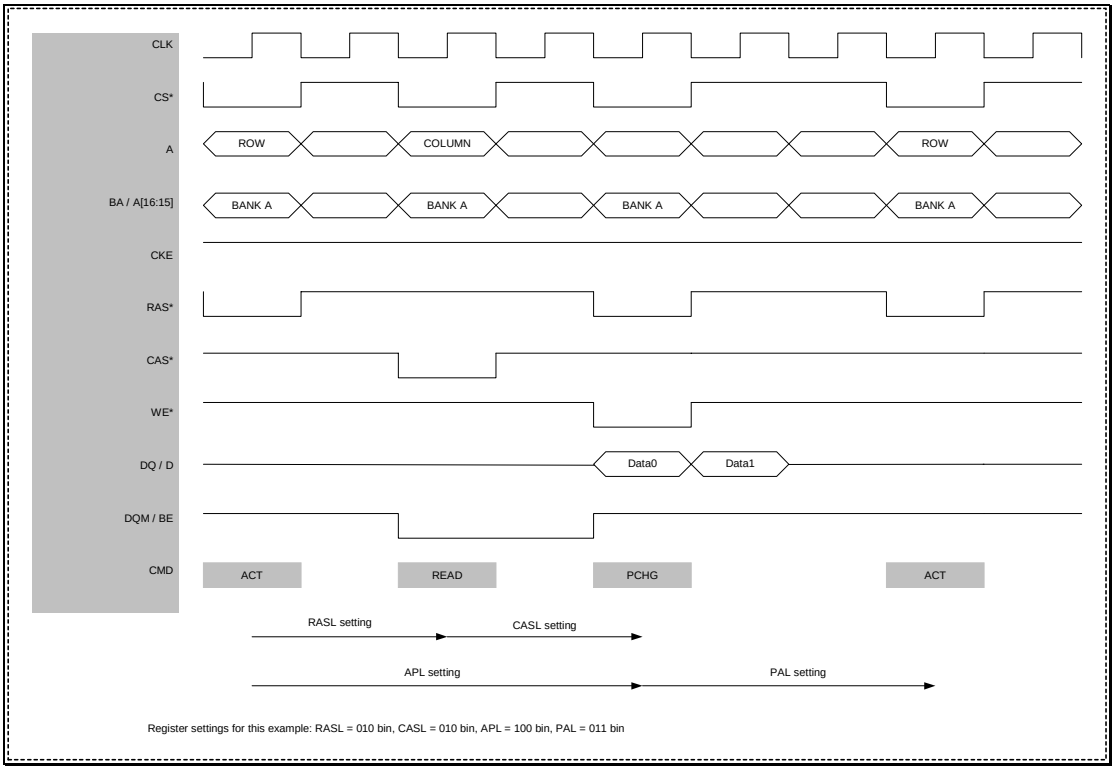


Figure 2.5.3 SDRAM Read Access Timing Diagram

Basic Write Timing:

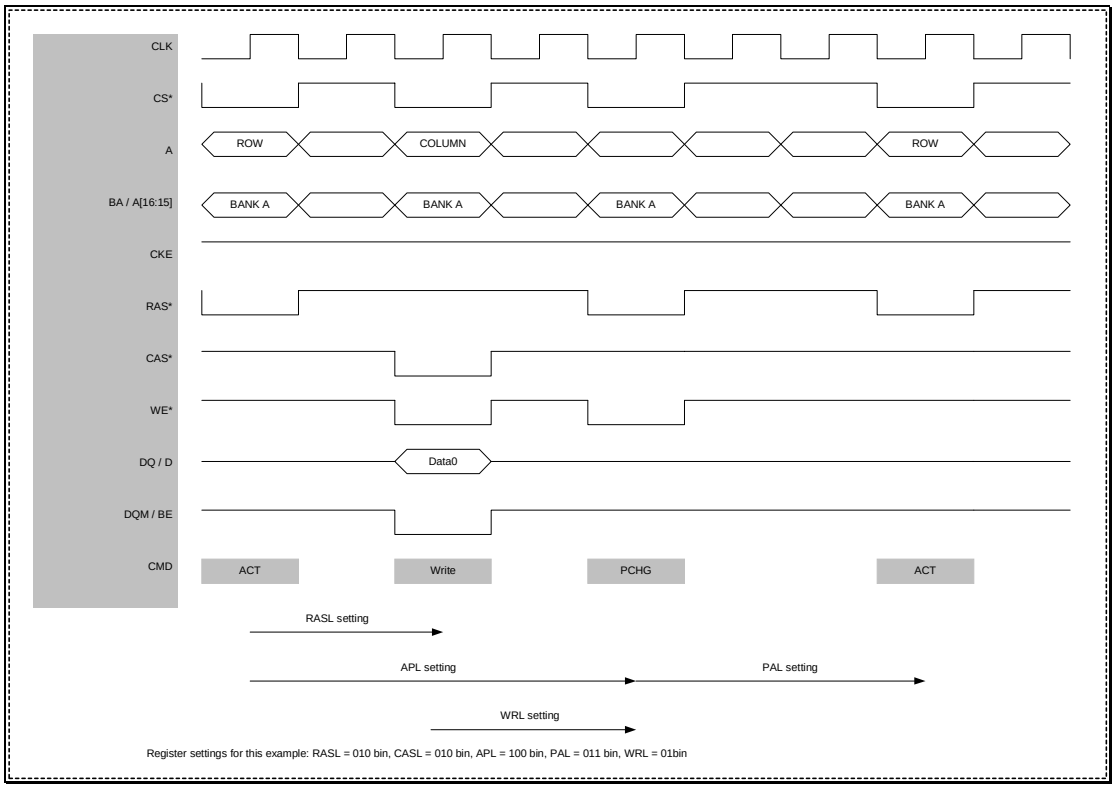


Figure 2.5.4 SDRAM Write Access Timing Diagram

Bank Interleaved Read Timing:

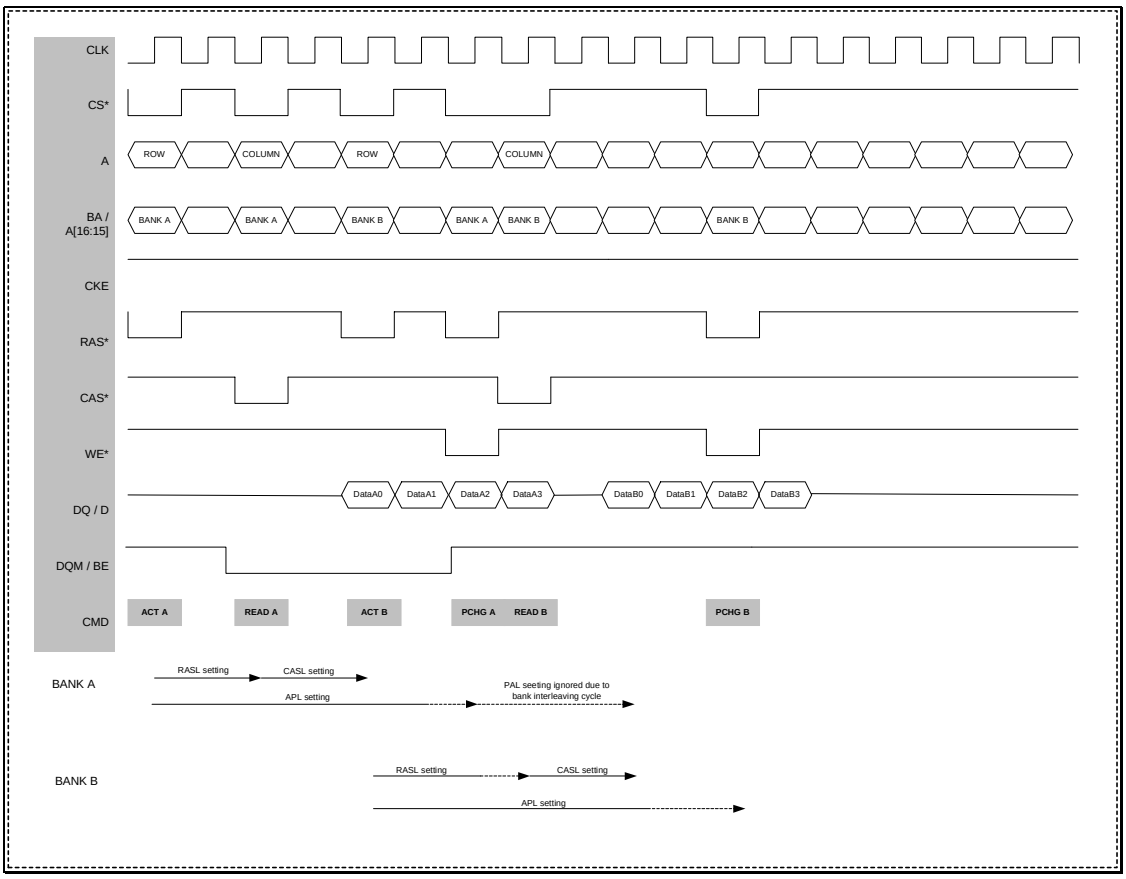


Figure 2.5.5 Read Access Using Bank Interleaving

Auto-Refresh Timing:

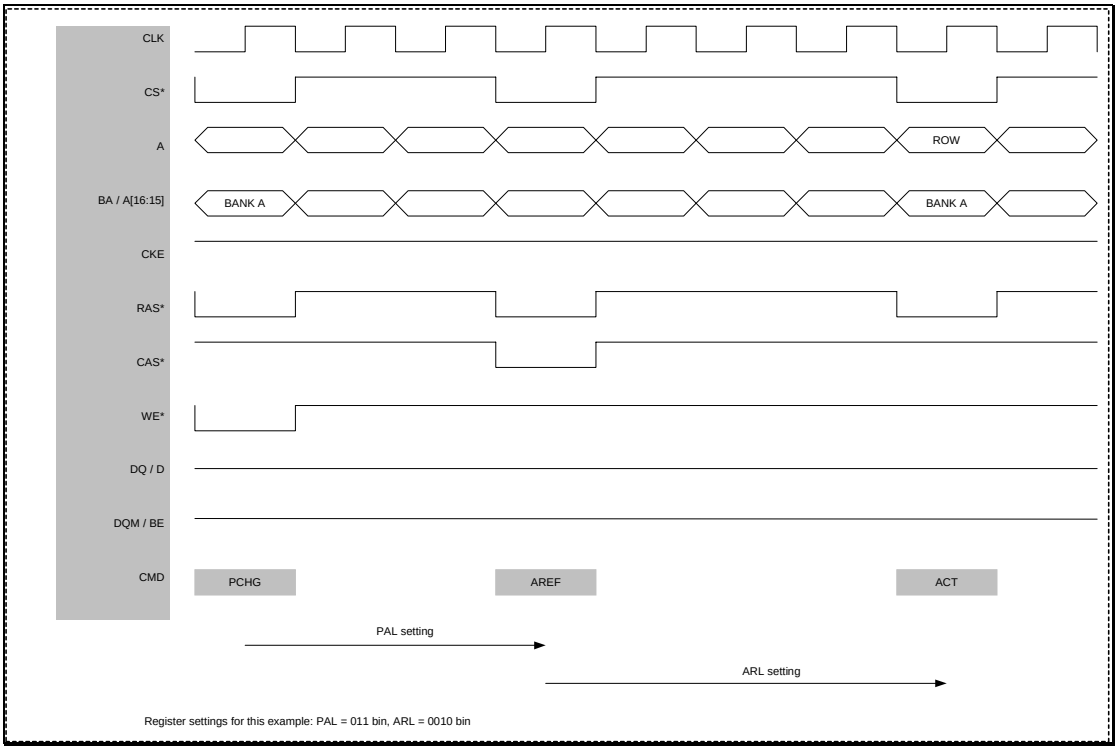


Figure 2.5.6 Waveform for Auto-Refresh Timing

Self Refresh Mode Timing:

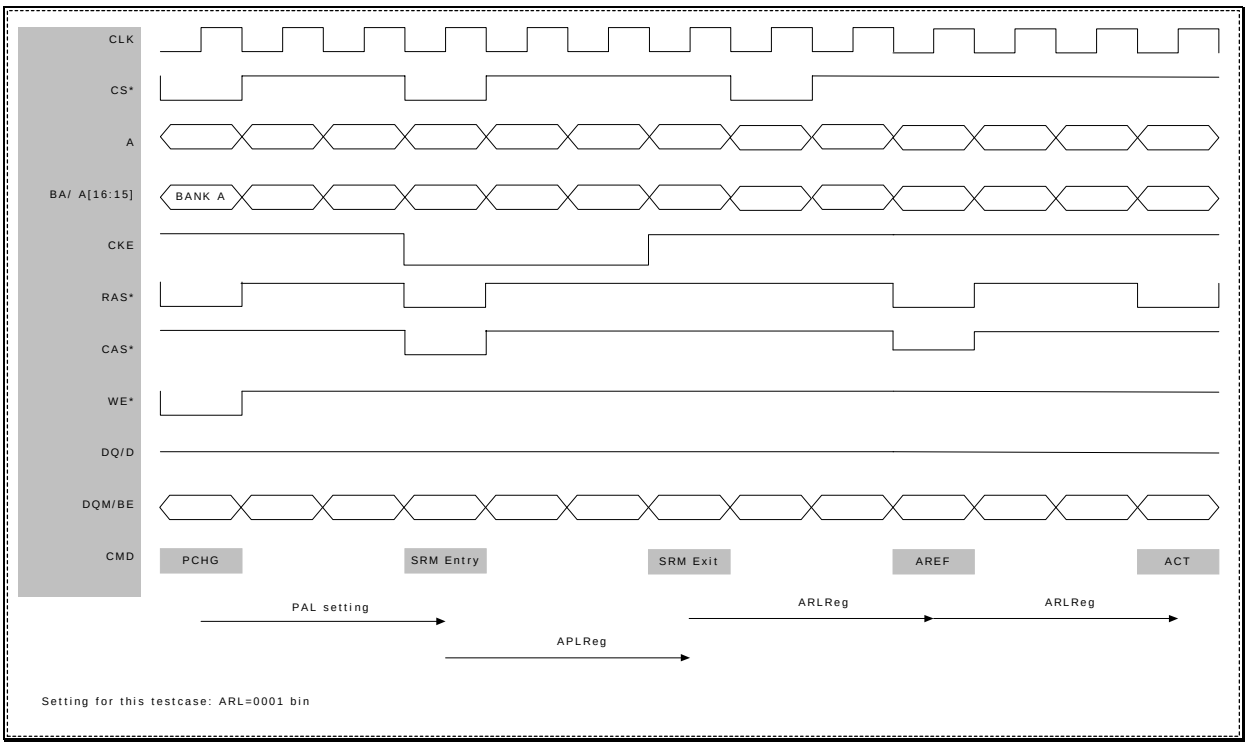


Figure 2.5.7 Timing Diagram for Self Refresh Mode

2.6 MEMC Function

Channel functionality is specified by applying special values to Channel Control Registers. They must be accessed using 32-bit cycles.

2.6.1 Channel Assignment

TMPR3916 contains the total of four multi-purpose memory controller channels.

The following chip-select signals are assigned to these channels:

Chip Enable Signal	Assigned to Channel
CS2	MEMC Channel 2
CS3	MEMC Channel 3
CS4	MEMC Channel 4
CS5	MEMC Channel 5 (Boot Channel)

Channel 5 is a special function channel and is used to boot the microprocessor from external devices like SRAM, ROM or FLASH memories.

2.6.2 Channel 5 Boot Function

For the system boot procedure, it can be chosen, whether the system shall be booted from a 16 or 32 bit device.

This choice is made by connecting a pull-up or pull-down resistor to the A26/ BOOT16 pin. The value of this pin is latched once during system startup on the rising edge of the reset signal, when the EBIF data bus is tri-stated.

Value of A26 / BOOT16 Pin on Rising Edge of Reset	Boot-function
0	Channel 5 is 32-bits wide at boot time
1	Channel 5 is 16-bits wide at boot time

2.6.3 Operational Modes

We can distinguish two operational modes depending on the types of memory connected to the channels 2 to 5. These two operational modes lead to two different settings: Flash/SRAM setting and Page Mode ROM setting.

2.6.3.1 Flash/SRAM setting

The SRAM and Flash memories cannot be accessed in page mode. Therefore some parameters in the RCCRx (x = 2, 3, 4, 5) control registers assume a particular meaning.

The setting of bits RPM is always 00 (“Not configured for page mode”).

The RPWT is no longer used as a wait state parameter for second and following accesses in burst mode. In non-page mode RPWT and RWT define together the wait time in the access to the Flash or SRAM. The RWT represents the lower significant bits of the wait time parameter, while RPWT represents the most significant bits.

When RWT = 0xF and RPWT = 0x3, the ACK* pin becomes an input and the data transaction is completed upon assertion of the ACK* signal by the accessed memory.

2.6.3.2 Page Mode ROM setting

Please note the following before setting the registers to drive memories in burst mode.

- Normal Sub-mode

  A channel enters this mode when the following conditions exist: RPM = 00 and (RPWT:RWT[3:0])! = 0x3Fh.

  In this mode the ACK*/READY pin is an ACK* output and the cycle is terminated based on a 6-bit wait counter. The 6-bit wait counter is the concatenation of the RPWT and RWT fields. Access time can be programmed to allow for 0 to 62 wait states. (Note: (RPWT:RWT[3:0] = 0x3Fh indicates external ACK* sub-mode.)

- External ACK* Sub-mode

  A channel enters this mode when the following conditions exist: RPM =00 and (RPWT:RWT[3:0] = 0x3Fh.

  In this mode the ACK*/READY pin is an ACK* input and the cycle is terminated by the external device. The ACK* input is synchronized before it is fed to the internal state machine. See section ACK* input timing for more details.

- Page Sub-mode

  A channel enters this mode when the following conditions exist: RPM! = 00.

  In this mode the ACK*/READY pin is an ACK* output and the cycle is terminated based on either the RPWT or RWT wait counter. The mode specifically targets Page Mode ROMs. During single cycle access, or the first word of a burst access, the 4-bit RWT field determines the access time. The access time can be programmed to allow for 0 to 15 wait states. During subsequent burst cycle accesses, the 2-bit RPWT field determines the access time and can be programmed to allow for 0 to 3 wait states.

  There are 3 different Page Mode Burst size settings allowed in the RPM field. When the Page Mode burst size is less than the CPU burst size, the channel will break access such that the 4-bit RWT field is always used on the programmed Page Mode boundary. In Page Mode the RWT time must be greater than or equal to the RPWT time or undetermined results may occur.

- 16-bit Bus Operation

  In the case of 16-bit mode if a single cycle is run from the G-Bus that requires a single byte or half word that is contained within one 16-bit word then only a single 16-bit access is run on the external bus. Otherwise two 16-bit accesses are executed externally. In the case of 16-bit mode, if a burst cycle is run from the G-Bus, two 16-bit cycles will be run for each of the burst accesses regardless of the fact that the internal byte enable signal is requesting a byte, half word or any other combination of internal byte enable signal that is not a full 32-bit word.

  In 16-bit mode the maximum channel size is 512 Mbytes.

- RSHT Option

  The RSHT option is entered when the RSHT field is non-zero. This option adds the capability of adding setup and hold time between the following signals:

  Setup — ADDR to CE, CE to OE, CE to BE.

  Hold — CE to ADDR, OE to CE, BE to CE.

  Typically it is used with slow I/O peripherals. All setup and hold times are the same for a given value and are not individually programmable.

  RSHT mode cannot be used in conjunction with Page Mode. All other modes can incorporate the RSHT mode but are restricted such that burst accesses are not allowed.

ROM Channel Control Register 2-4 (RCCR2-RCCR4)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RBA												RPM		RPWT	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RWT				RCS				RBS	—	RBC	—	RME	RSHT		

Bit	Name	Function	Reset Value	R/W
31:20	RBA	Base Address Designates the physical base address.	0x000	R/W
19:18	RPM	Page Mode, Page Size Designates the page size of channel word burst page mode. 00 = Not configured for page mode 01 = 4-word burst page mode 10 = 8-word burst page mode 11 = 16-word burst page mode	00	R/W
17:16	RPWT	Page Mode Wait Time Designates a 2-bit wait state counter in Page Mode for consecutive burst accesses. 00 = 0 wait cycles 01 = 1 wait cycle 10 = 2 wait cycles 11 = 3 wait cycles Designates the upper 2 bits of a 6-bit wait state counter for all other modes except External ACK mode. External ACK mode is entered when all bits of RPWT and RWT are set to 1. (Refer to “RWT”)	00	R/W
15:12	RWT	Normal Mode Wait Time Designates a 4-bit wait state counter in Page Mode for single cycles or initial burst cycle. 0000 = 0 wait cycles 0001 = 1 wait cycle 0010 = 2 wait cycles : : 1111 = 15 wait cycles Designates the lower 4-bits of a 6-bit wait state counter in all other modes except External ACK mode. External ACK mode is entered when all bits of RPWT and RWT are set to 1. RPWT [1:0] : RWT [3:0] 000000 = 0 wait cycles 000001 = 1 wait cycle 000010 = 2 wait cycles : : 011110 = 30 wait cycles 011111 = 31 wait cycles : : 111110 = 62 wait cycles 111111: External ACK* mode Note: When PM=00, if setting RPWT: RWT = 0x3f, the wait number doesn’t become the longest in ACK* output mode but the mode becomes ACK* input	0000	R/W

Bit	Name	Function	Reset Value	R/W
11:8	RCS	Channel Size Designates the memory size to be assigned. 0000 = 1 M byte 0110: 64 M bytes 0001 = 2 M bytes 0111: 128 M bytes 0010 = 4 M bytes 1000: 256 M bytes 0011 = 8 M bytes 1001: 512 M bytes 0100 = 16 M bytes 1010: 1 G bytes 0101 = 32 M bytes 1011-1111: Reserved In 16-bit mode the maximum channel size allowed is 512M Bytes.	0000	R/W
7	RBS	Bus Size Sets up the memory bus width of Channel 2. 0 = 32-bit bus size 1 = 16-bit bus size	0	R/W
6	—	Wired to zero	0	R
5	RBC	Byte Enable Control This bit determines whether the Byte Enable signals (BE[3:0]) are asserted on read and write accessed or only on write accesses. 0 = Byte Enables active on read and write accesses 1 = Byte Enables active only on write accesses	0	R/W
4	—	Wired to zero	0	R
3	RME	Master Enable Enables channel. 0 = Channel is disabled 1 = Channel is enabled	0	R/W
2:0	RSHT	Setup/Hold Wait Time Selects the number of wait states between address and chip enable signal, chip select signal and write enable/output enable signal. 000 = Disabled 001 = 1 wait 010 = 2 wait 011 = 3 wait : : 111 = 7 wait Burst access and Page Mode are not allowed if this bit field is non-zero.	000	R/W

ROM Channel Control Register 5 (RCCR5)

The channel 5 is used as boot channel. Note that the pin A26 / Boot16 has an impact on the bus width setting of the connected boot device.

The RCCR5 has the same bit functions like RCCR2 to RCCR4. Only reset-values differ. For detailed description of function, please see RCCR2-RCCR4.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RBA												RPM		RPWT	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RWT				RCS				RBS	—	RBC	—	RME	RSHT		

Bit	Name	Reset Configuration	Reset Value	R/W
31:20	RBA	Base Address (default: 0x1FC0 0000)	0x1FC	R/W
19:18	RPM	Page Mode, Page Size (default: no page mode)	00	R/W
17:16	RPWT	The device is not configured for page mode (RPM bits). This means that RPWT and RWT are combined to one counter value for the wait state generation. The wait state counter is set to 64 wait states, that can be reduced in the boot routine to a value that fits to the connected device.	11	R/W
15:12	RWT	Normal Mode Wait Time (default: 14 wait cycles)	1110	R/W
11:8	RCS	Channel Size (default: 4 Mbyte)	0010	R/W
7	RBS	Bus Size (configuration is latched from A26/BOOT16 pin)	BOOT16 Pin	R/W
6	—	Wired to zero	0	R
5	RBC	Byte Control (default: byte enables only active on write access)	0	R/W
4	—	Wired to zero	0	R
3	RME	Master Enable (default: enabled)	1	R/W
2:0	RSHT	Setup/Hold Wait Time (default: disabled)	000	R/W

2.6.4 Timing Diagrams

Single Read Access:

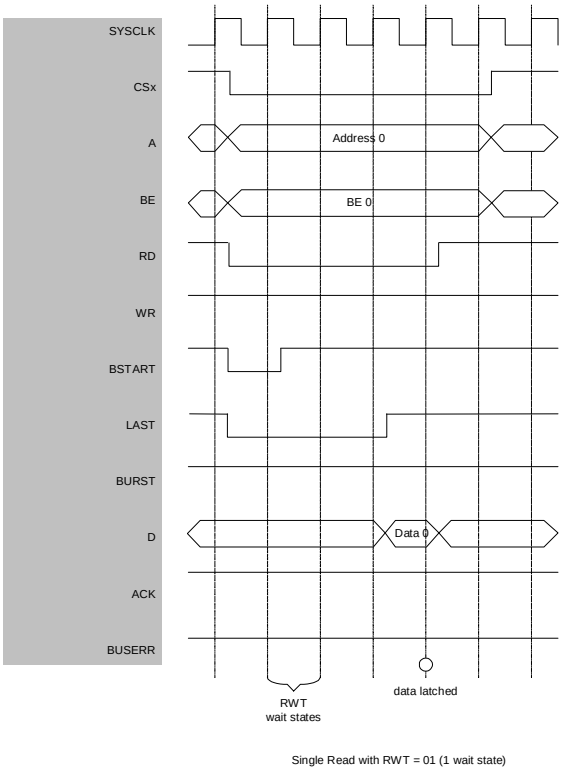


Figure 2.6.1 SRAM/ROM/Flash Single Read Access Timing

Single Write Access:

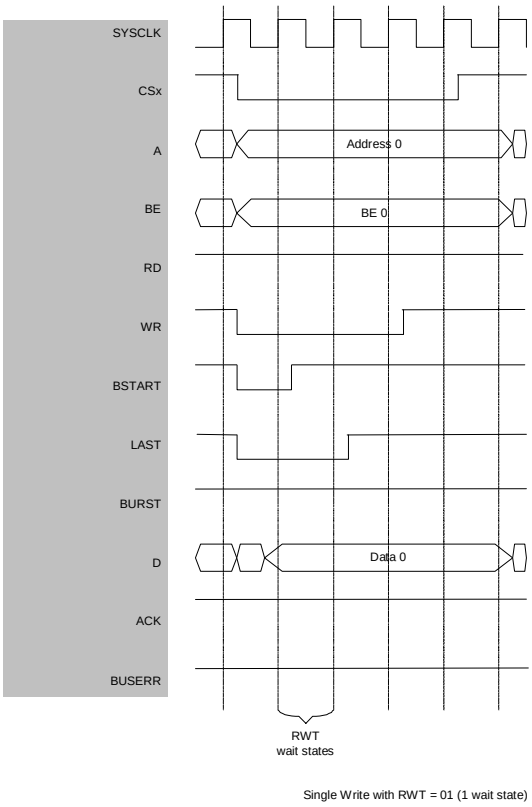


Figure 2.6.2 SRAM/ROM/Flash Single Write Access Timing

Page Mode Read Access:

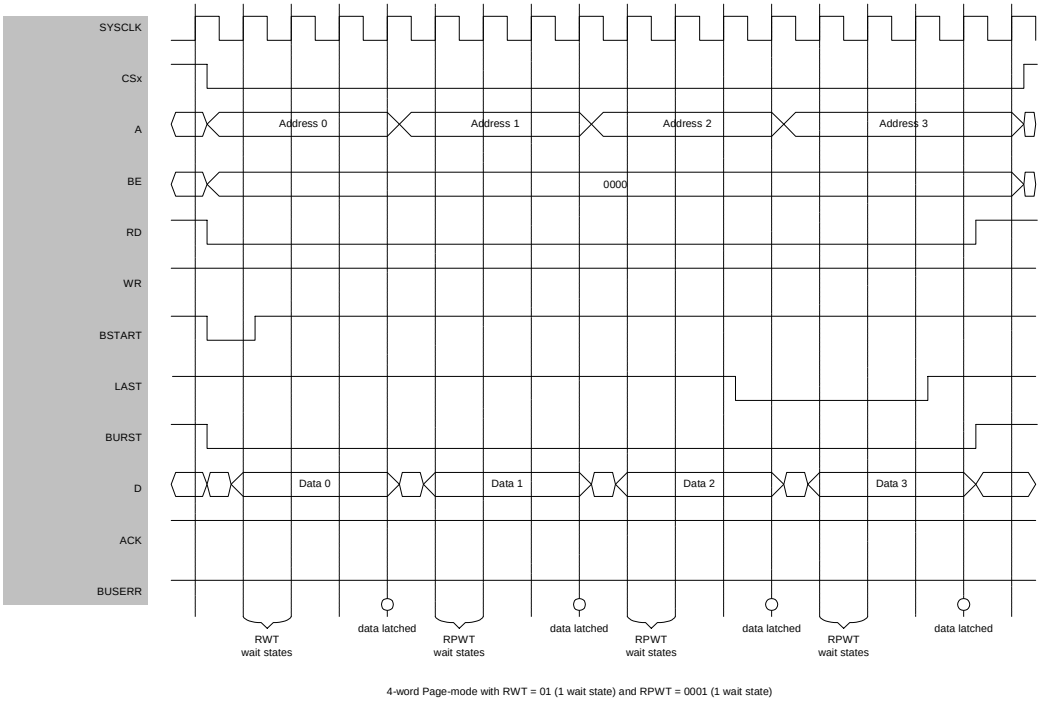


Figure 2.6.3 Timing Diagram for SRAM/ROM/Flash Read Access Using Page Mode

External Acknowledge Mode Read Access:

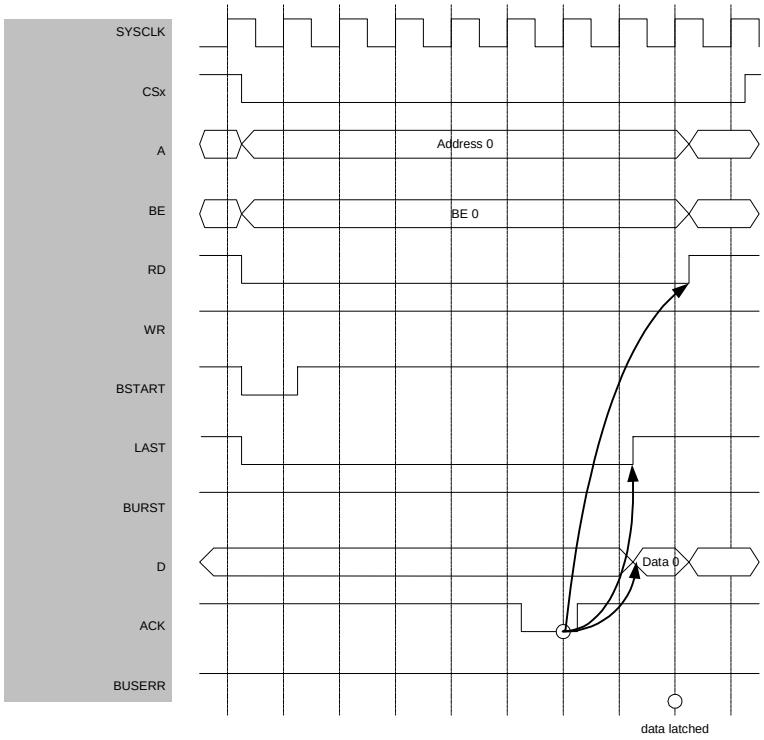


Figure 2.6.4 Waveform for External Acknowledge Mode Read Access Timing

External Acknowledge Mode Write Access:

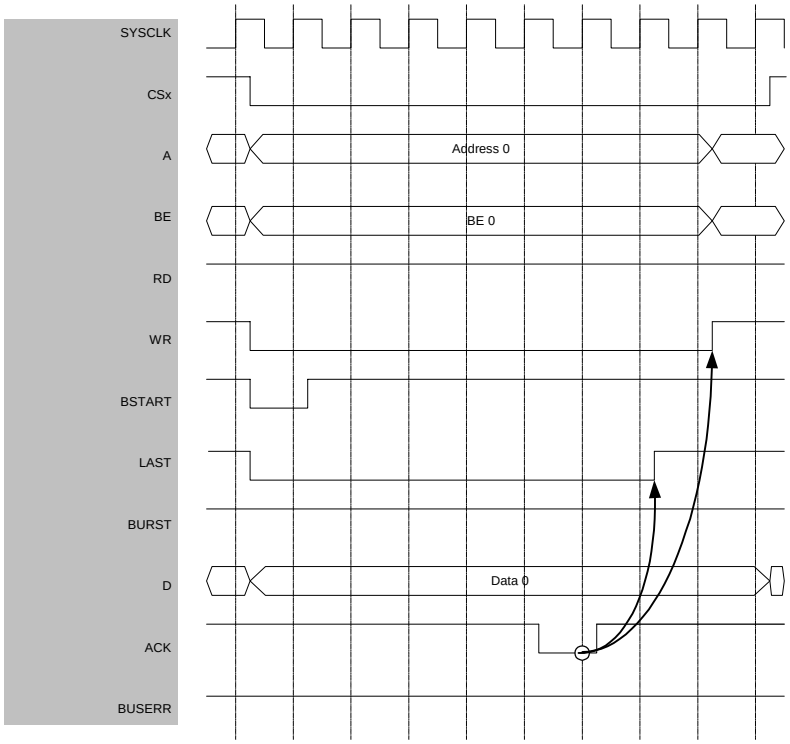


Figure 2.6.5 Waveform for External Acknowledge Mode Write Access Timing

External Acknowledge Mode Bus Error:

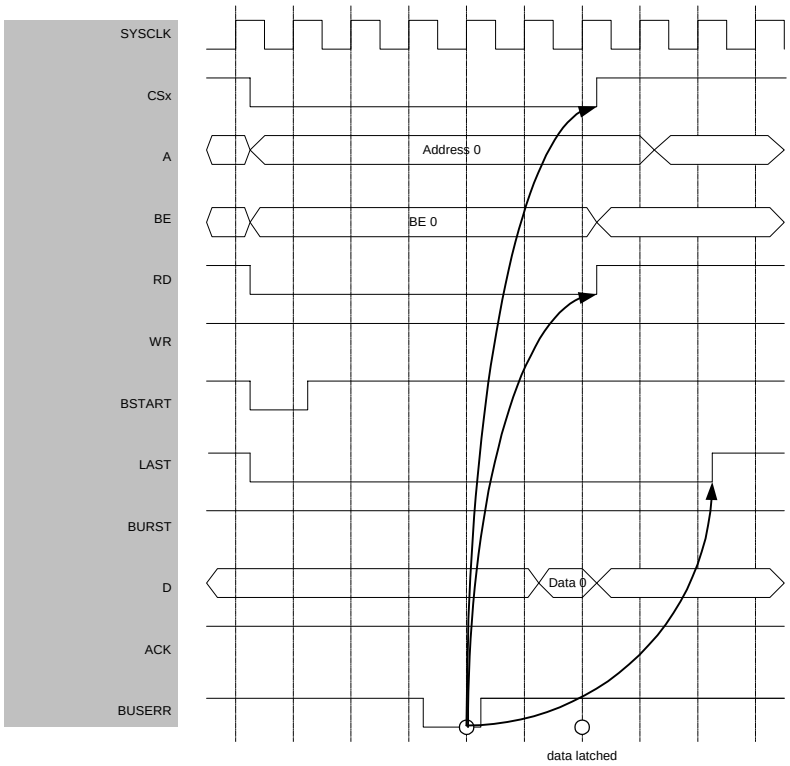


Figure 2.6.6 Bus Error During External Acknowledge Mode Timing

16 bit Read Access:

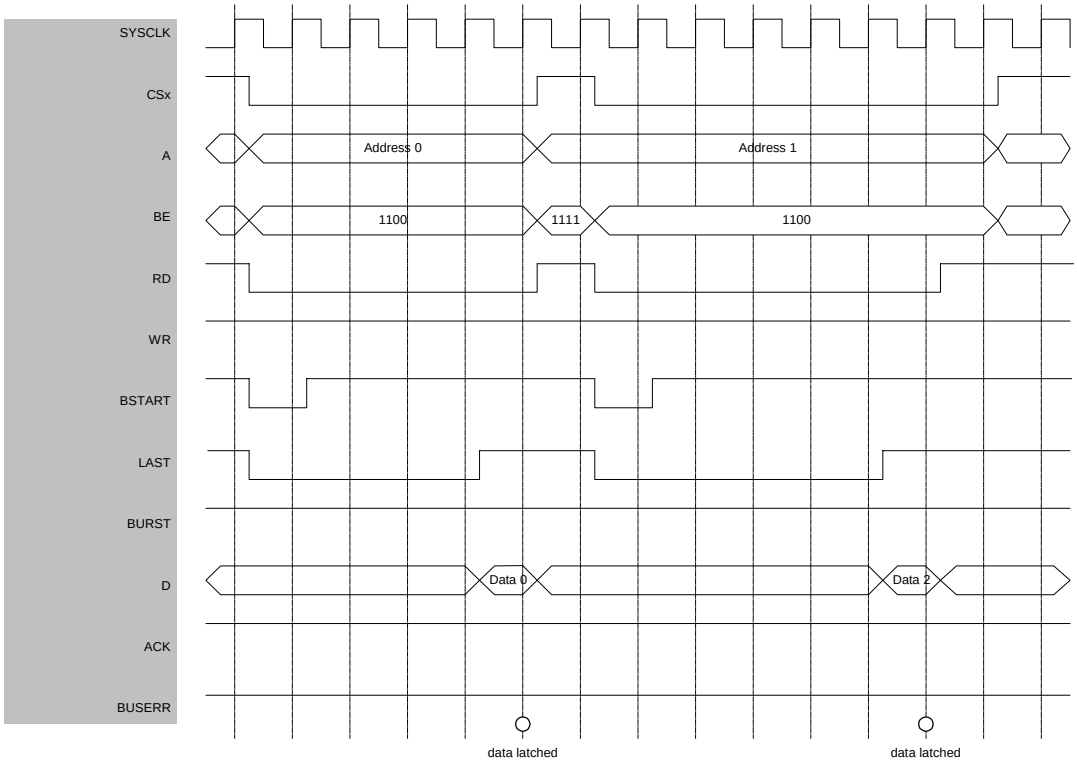


Figure 2.6.7 SDRAM Timing Diagram for 16-Bit Read Access

16 bit Write Access:

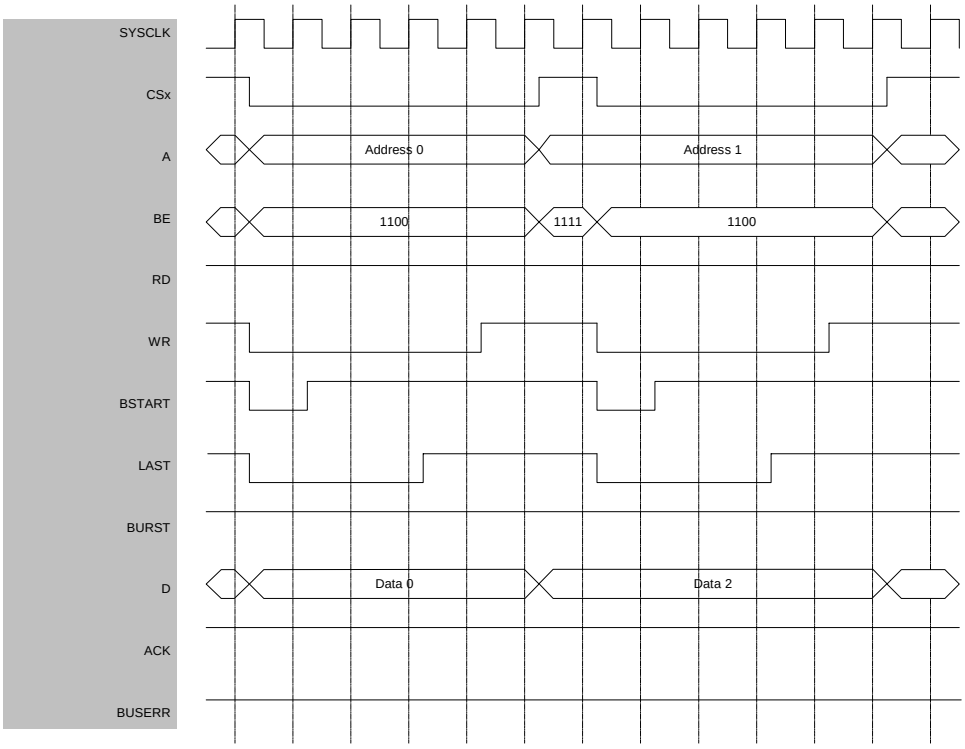


Figure 2.6.8 SDRAM Timing Diagram for 16-Bit Write Access

3. Graphics Display Controller (GDC)

The graphics display controller contains the following characteristics:

- Supports SDRAM and SRAM frame buffer
  → Burst mode for reading on SDRAM
- Supports four-layer overlay display function using hardware processing

Layer A/E:	Map mode (256 colors) – referred to as layer A,
	Picture mode (65,536 colors) – referred to as layer E
Layer B:	Map mode (one of the 256 colors is transparent)
Layer C:	Map mode (one of the 16 colors is transparent)
Layer D:	Map mode (one of the 16 colors is transparent)

  → Displaying layers A and E together is not possible.
  → Layer D screen size can be set independently of the other layer sizes.
  → Incorporates a 544 entries color palette (256 colors × two layers and 16 colors × two layers).
- Display control:
  → Non-interlaced scanning
  → Smooth scrolling (vertical and horizontal) for layers A, B, C and D
  → Generates and outputs synchronization signals (HSYNC, VSYNC / CSYNC), also supports control by external synchronization signal input.
  → Both digital and analog RGB signal output
- Dot clock:
  → Supports internal and external dot clock

3.1 GDC Structure

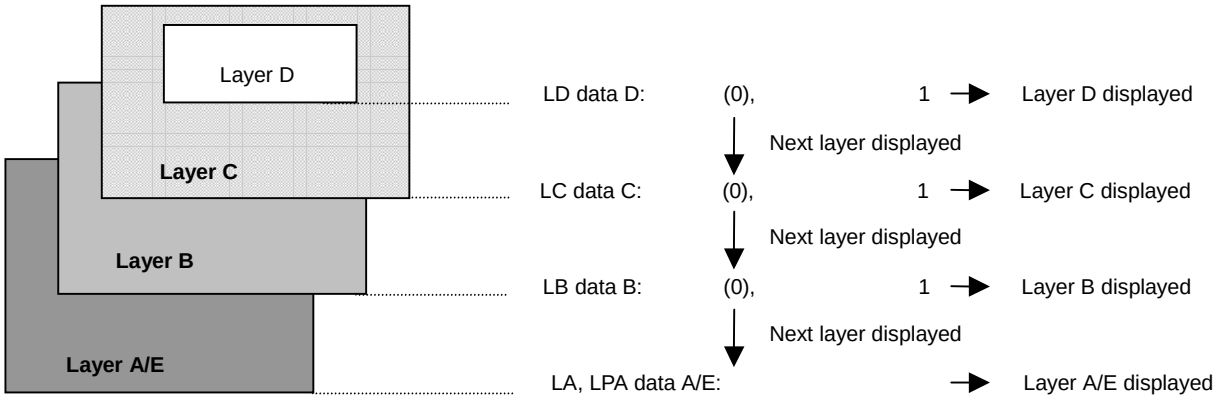
3.1.1 Display Screen

The TMPR3916 can display four layers, overlaid in the following order: A/E, B, C and D. Top layer is D (see Figure 3.1.1). Displaying layers A and E together is not possible because these layers use the same resources except for color palette and dot buffer.

The blank signal BLK enables the display data output. The layer active signals LA, LB, LC and LD activate each layer.

Layer D screen size can be set independently of the other layer sizes.

Layer A can be switched between Map mode and Picture mode (then called layer E). Use the LPA bit of the display control register (DCR) to switch the modes.



Lx data x = 1	: Display
Lx data x = 0	: No display (transparent)
Lx	: Bit in display control register
data x	: Layer x display data
x	: A/E, B, C, D

Figure 3.1.1 Layer Arrangement

As can be seen in the figure above layer A/E will always be displayed if all other activated layers contain transparent dots at the current location. Therefore it is recommended to always use one of these two layers to prevent the system from reading dots in an undefined background.

For display dimensions the following restrictions apply: Layers A/E, B and C shall be configured at least 64 dots wide horizontally, layer D should have 16 dots minimum. In vertical direction no restriction exists.

3.1.2 Frame Buffer

The TMPR3916 supports SDRAM and SRAM as frame buffer.

The GDC uses separated address generators for each of its four layers A/E, B, C, and D. It can specify the A/E, B, C, or D layer display data address area by writing the data to the start address register (SARx, where x stands for: A/E, B, C, D), the memory width register (MWRx), the horizontal display start register (HDSR) and the horizontal display end register (HDER). The HDSR and HDER are stored as one value in the horizontal display start end register (HDSER).

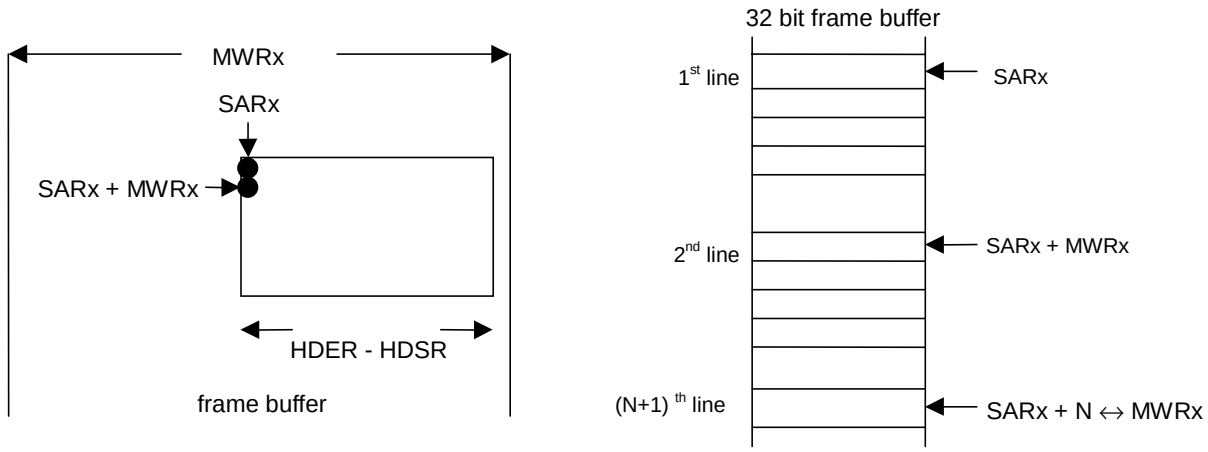


Figure 3.1.2 Frame Buffer Data

The SARx stores the address of the upper-left corner's dot belonging to layer x. The MWRx stores the offset value to be added to the contents of SARx to get the address of the left-most pixel of the following line. It is not allowed to store the last dot of one line and the first dot of the next line within one memory word as shown below.

The TMPR3916 dot data structure in the frame buffer is as follows: In map mode these values are addresses for color palette. In picture mode these values represent the displayed colors.

	Data Length Per Dot	Active Level	# Colors
Map mode C/D	4 bits	Layers C, D	16
Map mode A/B	8 bits	Layers A, B	256
Picture mode	16 bits	Layer E	64K

Accordingly, the data structure for each word is as follows.

Bits	31:28	27:24	23:20	19:16	15:12	11: 8	7:4	3:0
Map mode C/D	Dot 1	Dot 2	Dot 3	Dot 4	Dot 5	Dot 6	Dot 7	Dot 8
Map mode A/B	Dot 1		Dot 2		Dot 3		Dot 4	
Picture mode	Dot 1				Dot 2			

The minimum value to be stored in MWRx is equal to the number of dots per line in the frame buffer divided by 4 (in case of layers A and B, 8 for C and D and 2 for layer E) and rounded to the next greater integer value (MWRx stores numbers of 32 bit words). In practice, it is advisable to have a certain space between two adjacent lines. Therefore the practical memory size is larger than the one requested to cover just one layer. The frame buffer contents represent the address of the color palette which contains the displayed colors for the respective layer:

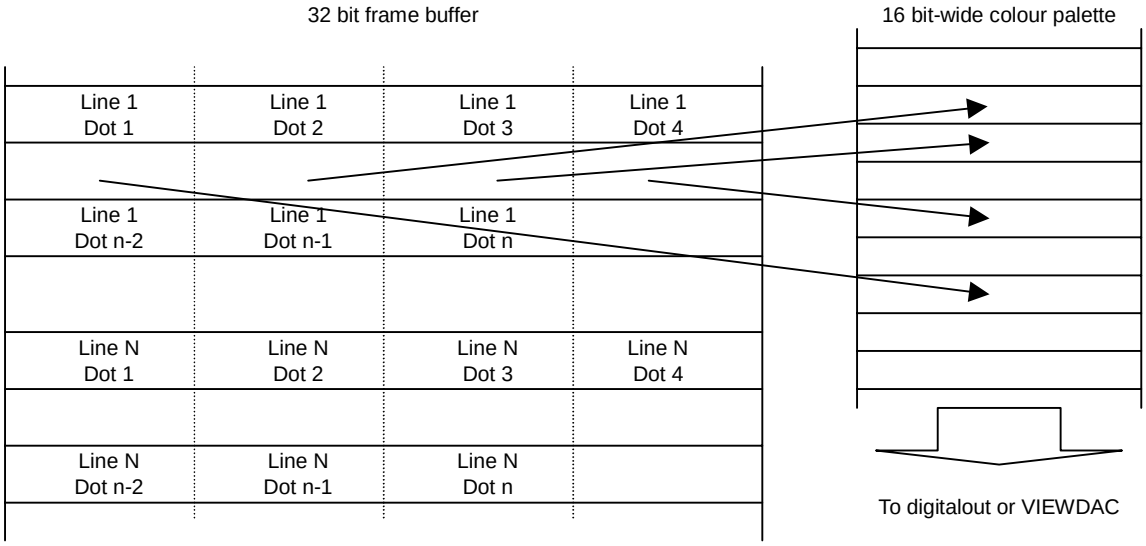


Figure 3.1.3 Frame Buffer and Color Palette in Case of Layers A or B

In case of layers C and D, the contents of register MWRx must be adjusted taking into consideration that there are 8 dots in a 32 bit word. Therefore the minimum value is given by the number of dots in a line divided by 8.

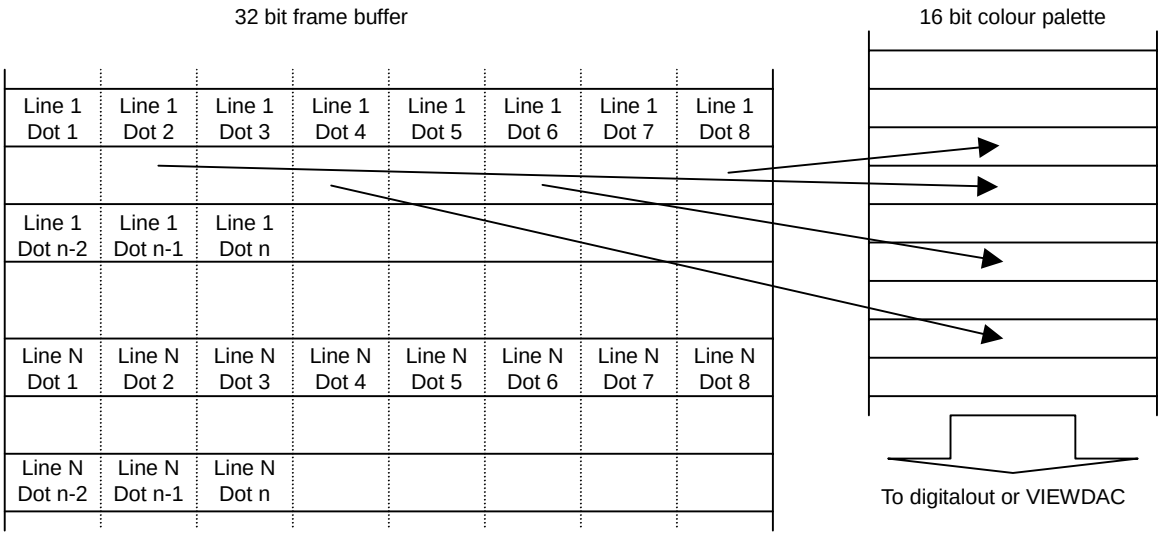


Figure 3.1.4 Frame Buffer and Color Palette in Case of Layers C or D

In case of layer E, each 32 bit word is storing the data of 2 dots. So the minimum number to store in MWRA (E layer uses A layer registers) is equal to the number of dots in a line divided by 2. No color palette is needed for layer E.

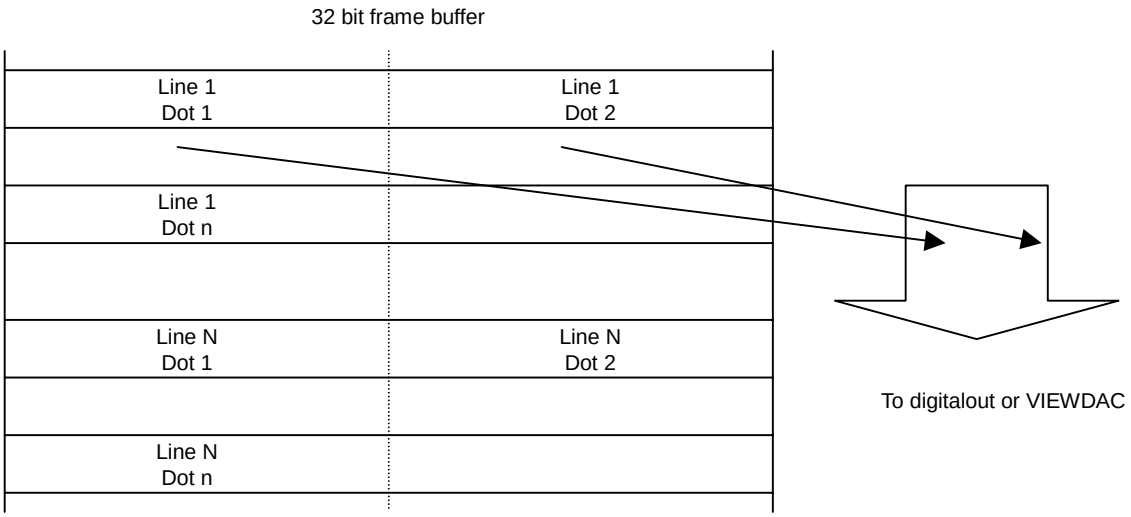


Figure 3.1.5 Frame Buffer in Case of Layer E

The number in MWRx is always rounded to the higher integer.

Some display devices do colour calibrations during synchronization periods. That is why the graphics display controller outputs black level as analog output from VIEWDAC while HDISP=0. For digital output this value is stored in register PA (see PORT module) with respect to the output mode (dot or pixel). As a result the PA register has to be initialized with the correct value (reset value: 0x00h).

3.1.3 Display Control Signal

The GDC block generates and outputs the synchronization signal (HSYNC, VSYNC, and CSYNC). The display uses non-interlaced scanning.

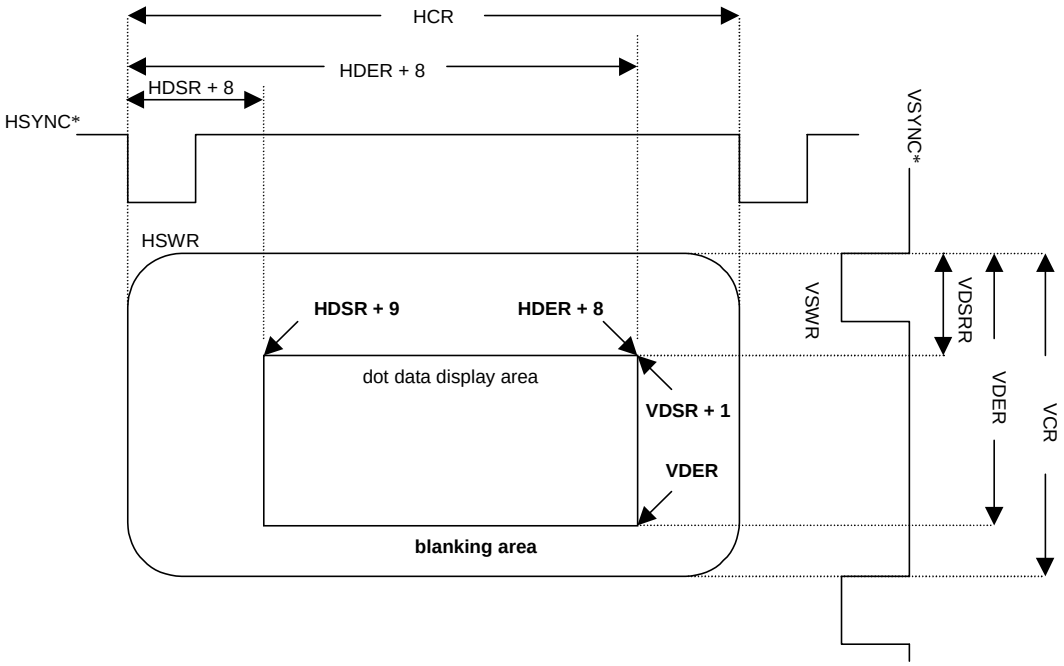


Figure 3.1.6 Display Timing

The GDC block can select and output a separated synchronization (HSYNC*, VSYNC*) or a composite synchronization (CSYNC*) signal. The composite CSYNC* is the EX-NOR (exclusive nor) signal of HSYNC* and VSYNC*.

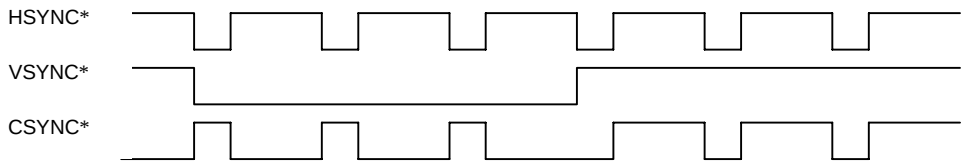


Figure 3.1.7 CSYNC* Timing

In positive mode the composite CSYNC is the EX-OR (exclusive or) signal of HSYNC and VSYNC.

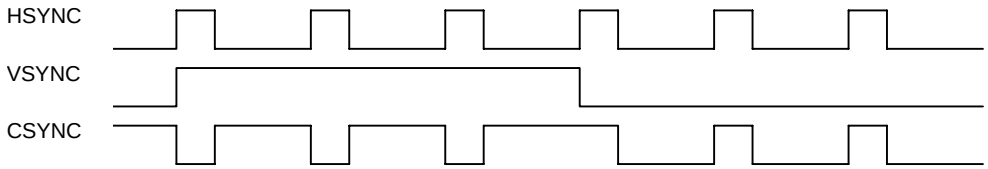


Figure 3.1.8 CSYNC Timing When HSYNC and VSYNC are in Positive Mode

The DCR bit CSEL enables composite synchronization signal output. The bit NSYNC enables using synchronization signals in positive mode.

Externally separated synchronization signals can also be used for input. Set using the ESYNC bit of the DCR.

Note that the use of external synchronization signals needs processing time. Therefore the input signals HSYNC and VSYNC are delayed internally. To compensate this delay, set the horizontal display-related registers -3, the vertical display-related registers need not be changed. For positive mode set the horizontal display-related register –6 and the vertical display-related register -2.

When using external HSYNC and VSYNC signals, the horizontal synchronization pulse width (HSWR), vertical synchronization pulse width (VSWR), horizontal cycle register (HCR) and vertical cycle register (VCR) settings are invalid.

External composite synchronization signals cannot be processed by the TMPR3916.

Please note also that due to internal data structures HDS and HDE are automatically incremented by 8 so that the effective values are HDS+8 and HDE+8.

3.1.4 Dots and Pixels

The GDC is able to support both, dot and pixel on digital output. The 16-bit data for one dot is divided into R (red), G (green), and B (blue). A pixel represents one of these colors in a 6-bit value. In consequence one dot represents three pixels as shown in the following figure:

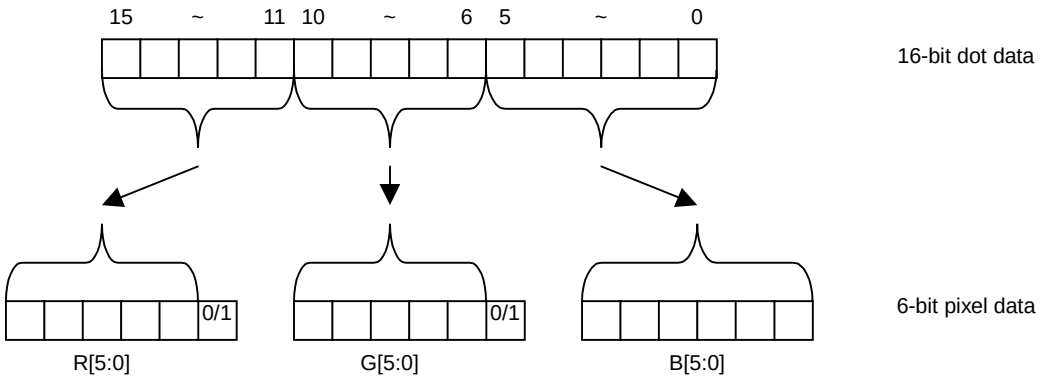


Figure 3.1.9 Partitioning RGB Data into Dot and Pixel Data

Each six bits of RGB data are input to the 6-bit DAC (three channels) in the GDC. The LSB of R and G transmitted to the DAC data are always fixed to zero.

In dot mode (DCR[3] = 0) the GDC supports dots on the digital RGB port (PIO[15:0]) and dot clock on the DOTCLK port. In pixel mode (DCR[3] = 1) the GDC supports pixels on the lower 6 ports of the digital RGB port (PIO[5:0]) and pixel clock on the DOTCLK port. Within the pixel mode the LSB of R and G are equal to bit 4 of the DCR register (DCR[4]).

3.1.4.1 Dot Clock

The dot clock is the reference clock for graphics data output. It is also used to determine the screen size, for example, of each layer. Use the following formula to select the dot clock:

$$\text{Dot clock speed (MHz)} = \frac{\text{number of display dots per line (including non visible dots)}}{\text{horizontal display period } (\mu s)}$$

Note that the internal circuit imposes the following restriction:

dot clock speed (MHz) < ½ SYSCLK.

The dot clock can be provided internally as a derivative of the system clock. The divisor is defined by DCKPS (see Display Control Register (DCR)) and can be set to 4, 6, 8, …, 24. Therefore frequencies of 2.5 MHz up to 15 MHz can be reached when using the internal dot clock. Default setting for DCKPS is 0x4h for a dot clock of 6 MHz.

Alternatively the dot clock may be input via the DOTCLK pin.

3.1.5 Color Palette

In Map mode, each dot can be set for 16 colors using layer C or D and 256 colors using layer A or B. By using the color palette, any 16 or 256 colors respectively can be selected from among the 65,536 colors. The color palette can be set independently for each layer (A, B, C or D). One of the defined colors is transparent for layers B, C and D. It must not be used if no background layer is activated.

In Picture mode (layer E), the TMPR3916 can display up to 65,536 colors. The dot data are directly defined at the frame buffer and therefore no color palette is necessary.

When programming values to CLUT entries the upper 16 bits of the bus represent the color value, the lower 16 data bits are ignored.

Example: For programming entry #7 in layer A's color look-up table to a color with a red part of R = 20, a green pixel G = 23 and a blue one of B = 48, load 0xA5F00000 into memory location 0x1C05081C.

The following table shows the color palette structure:

Layer	*Palette Name*	Number of Colors	Color Palette No.	*Address*	Color Specification
A	*CPLTA*	256 colors	0 1 : 255	*1C05 0800* *1C05 0804* *:* *1C05 0BFC*	free defined color 0 free defined color 1 : free defined color 255
B	*CPLTB*	255 colors	0 1 : 255	*1C05 0C00* *1C05 0C04* *:* *1C05 0FFC*	transparent free defined color 0 : free defined color 254
C	*CPLTC*	15 colors	0 1 : 15	*1C05 0180* *1C05 0184* *:* *1C05 01BC*	transparent free defined color 0 : free defined color 14
D	*CPLTD*	15 colors	0 1 : 15	*1C05 01C0* *1C05 01C4* *:* *1C05 01FC*	transparent free defined color 0 : free defined color 14

3.2 Scrolling

There are two different ways of scrolling when using the TMPR3916's graphics display controller: **Smooth scrolling** and some kind of **register scrolling**.

Smooth scrolling is done by changing the start address of graphics data in memory (SARx). The layers in TMPR3916 have different resolutions in colour depth. That is why the number of dots per word changes from layer to layer. Therefore the address of the next dot for display differs. This implies different handling of smooth scrolling in the different kinds of layers.

The easiest implementation can be reached using layers C and D. It is also the type of implementation this document starts with. Following is the description for layers A and B.

Register scrolling is another special type of scrolling a layer by changing the values for horizontal and vertical display start and end. As layers A, B, C and E have common settings for display size and are normally fixed to the display's maximum settings, this scrolling type only applies to layer D. The related registers are HDSERD, VDSRD and VDERD.

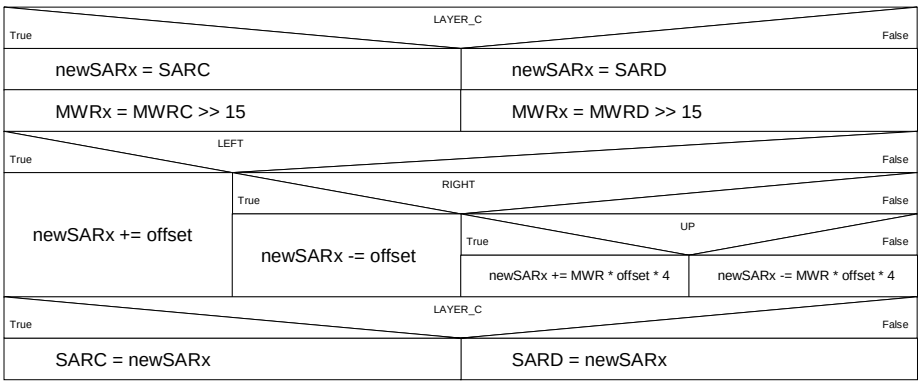
The algorithms following are based on C syntax for documentation. They also imply just one step at a time in one direction. For moving in several directions at the same step the related algorithm has to be executed for each direction. For example, if scrolling to the left and up is wanted, two runs of the algorithm are needed – one for "LEFT", one for "UP".

It is assumed that only four directions have been defined: LEFT, RIGHT, UP and DOWN. This is why the "UP" decision automatically branches into "DOWN" case if false.

3.2.1 Smooth Scrolling

3.2.1.1 Scrolling Layers C and D

For smooth scrolling three bits are reserved in start address register (SARx). As layer C and D both consist of data with eight dots per word and this is the maximum amount of differing addresses that can be generated using three bits for coding, no problem occurs for these layers: Just add (subtract) to (from) SARx the amount of dots you want to use as offset for smooth scrolling to the left (right) or respectively up (down). The related algorithm is shown in the following diagram:



Example:

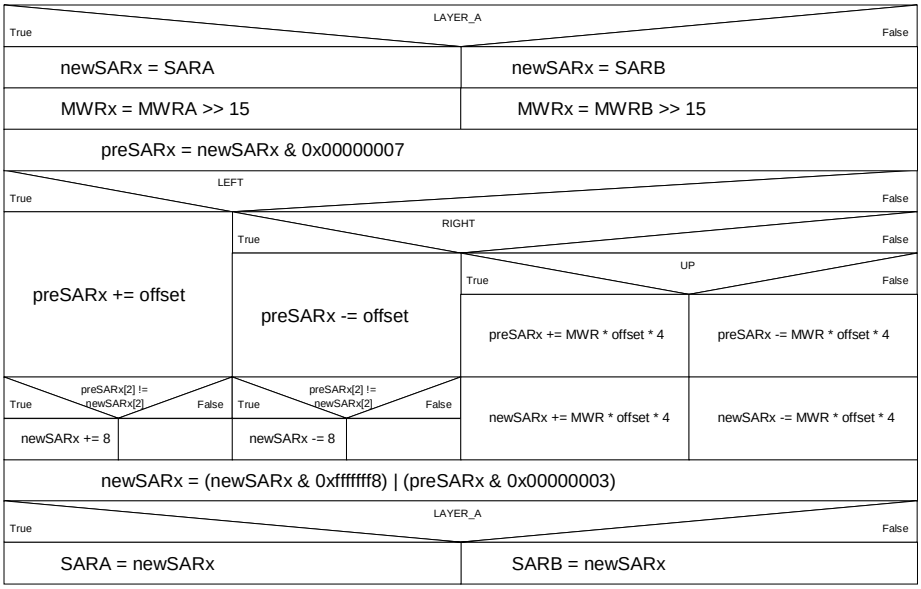
For scrolling a picture located at 0x50010024 (SARx: 0xA0020048) to the left by one dot simply add 0x01 to the latter value: 0xA0020049 is the correct start address register value for the next picture (physical: 0x50010024). Note that the least significant bits in this start adress register

are not related to the address counter part (AC) but to this smooth scrolling feature. This leads to a dot offset of 1 in the SS register part, so that the dot read as next is located at 0x50010025.

3.2.1.2 Scrolling Layers A and B

For working on layers A and B the application of smooth scrolling becomes a little more difficult: One word contains data of four dots, of which the number can be coded by using just two of the three bits reserved in SARx.

The algorithm shown in the following diagram is to be used:



Example:

For scrolling a picture located at 0x50010024 (SARx: 0xA0020048) to the right by four dots do the following:

1. newSARx = 0xA0020048
2. MWRx = 0x-------- (not needed for this example)
3. preSARx = 0x00000000
4. preSARx = 0xFFFFFFFC
5. newSARx = 0xA0020044
6. newSARx = 0xA0020044
7. SARA = 0xA0020044

This new entry for start address register results in a physical bus address of 0x50010020, which is the memory address located before the one read first in the last step. The smooth scrolling bits show an offset of 0 so in the end the first dot read is located at 0x50010020, too.

Please be aware of the relationship of adding or subtracting the extra offset 0x08 to the scrolling direction.

3.2.2 Register Scrolling

This scrolling type is implemented easily: Just change the values of the start and end registers in horizontal and vertical direction using the projected dot offset:

HDSERD = HDSERD + ((horoffset << 16) + horoffset)

VDSRD = VDSRD + (vertoffset << 16)

VDERD = VDERD + (vertoffset << 16)

Note: For register scrolling the offset handling is kind of reversed in comparison to smooth scrolling. For scrolling a layer to the right the coordinate of the first visible dot has to be incremented. For scrolling a picture in a layer to the right the start address has to be decremented.

Example:

Start values:

- Horizontal start: dot 34
- Horizontal end: dot 50
- Vertical start: dot 42
- Vertical end: dot 58

For moving layer D up by 2 lines and right by 3 dots the following calculation applies:

HDSERD = 0x00220032 + 0x00030003 = 0x00250035

VDSRD = 0x002A0000 – 0x00020000 = 0x00280000

VDERD = 0x003A0000 – 0x00020000 = 0x00380000

3.3 Internal Blockdiagram

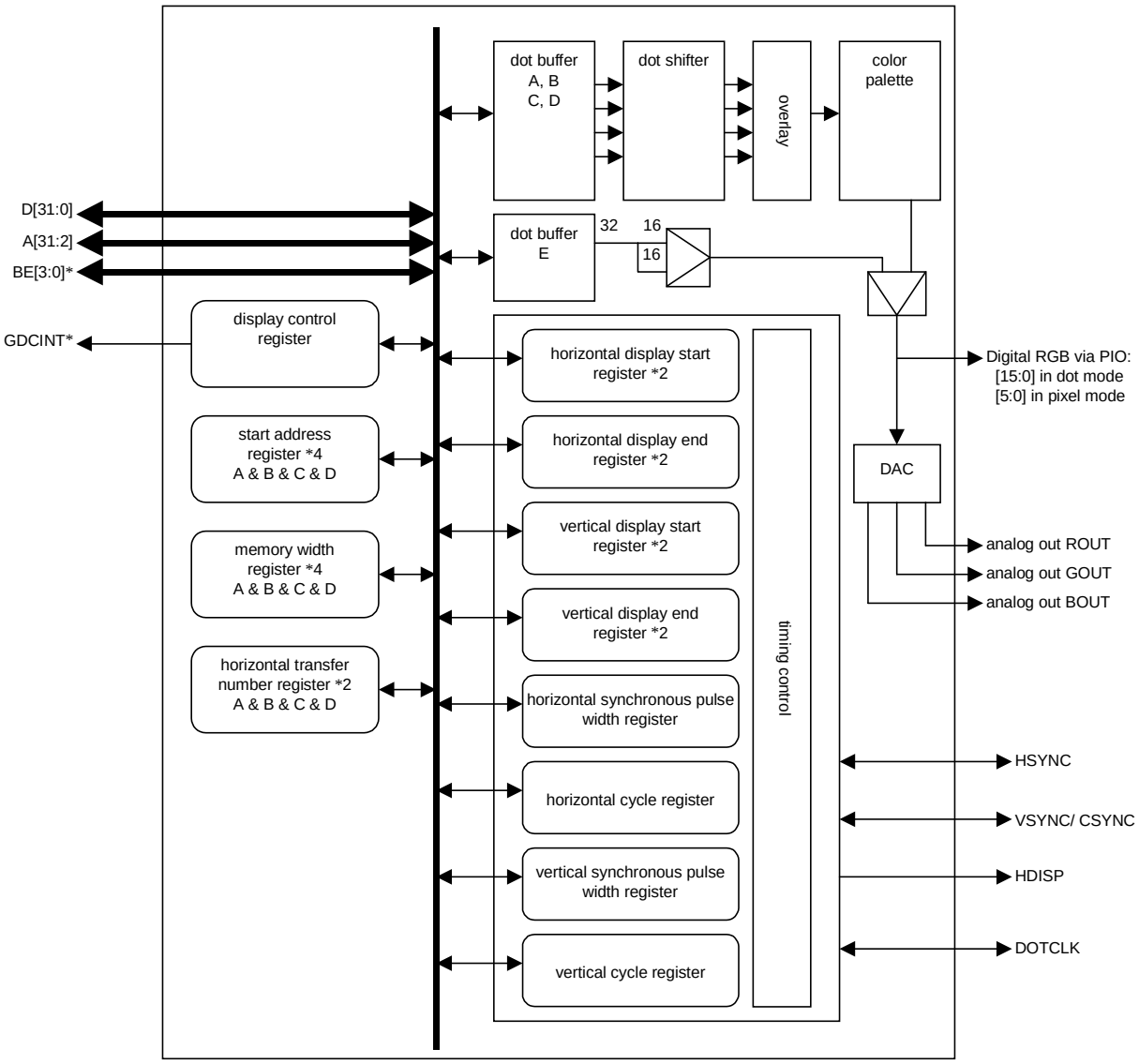


Figure 3.3.1 GDC Block Diagram

3.4 Registers

Overview

Short Name	Address	*Name*	Function
DCR	1C050000	*display control register*	Sets the GDC operations.
SARA/E	1C050010	*Layer A/E's start address register*	Specifies the start address of layer A/E on the frame buffer.
SARB	1C050014	*Layer B's start address register*	Specifies the start address of layer B on the frame buffer.
SARC	1C050018	*Layer C's start address register*	Specifies the start address of layer C on the frame buffer.
SARD	1C05001C	*Layer D's start address register*	Specifies the start address of layer D on the frame buffer.
MWRA/E	1C050020	*Layer A/E's Memory Width register*	Specifies the line width of layer A/E on the frame buffer.
MWRB	1C050024	*Layer B's Memory Width register*	Specifies the line width of layer B on the frame buffer.
MWRC	1C050028	*Layer C's Memory Width register*	Specifies the line width of layer C on the frame buffer.
MWRD	1C05002C	*Layer D's Memory Width register*	Specifies the line width of layer D on the frame buffer.
HTN	1C050030	*Layer A/E, B, C's horizontal transfer number register*	Sets the number of display dots for one line divided by 32. For layers A/E, B, and C.
HTND	1C050034	*Layer D's horizontal transfer number register*	Sets the number of display dots for one line divided by 32. For layer D.
HDSER	1C050038	*Layer A/E, B, C's horizontal display start / end register*	Sets the layer's display position and the number of display dots for layers A/E, B, and C.
HDSERD	1C05003C	*Layer D's horizontal display start / end register*	Sets the horizontal display position and the number of display dots for layer D.
HCR	1C050040	*horizontal cycle register*	Specifies the total number of dots within a horizontal cycle.
HSWR	1C050044	*horizontal synchronization pulse width register*	Specifies the horizontal sync signal pulse width using the number of dot clocks.
VCR	1C050048	*vertical cycle register*	Specifies the total number of lines within a vertical cycle.
VSWR	1C05004C	*vertical synchronization pulse width register*	Specifies the vertical sync signal pulse width using the number of lines.
VDSR	1C050050	*Layer A/E, B, C's vertical display start register*	Sets the vertical display start position for layers A/E, B, and C.
VDSRD	1C050054	*Layer D's vertical display start register*	Sets the vertical display start position for layer D.
VDER	1C050058	*Layer A/E, B, C's vertical display end register*	Sets the vertical display end position for layers A/E, B, and C. The difference VDER – VDSR defines the number of display dots for layers A/E, B, and C.
VDERD	1C05005C	*Layer D's vertical display end register*	Sets the vertical display end position for layer D. The different VDERD – VDSRD defines the number of display dots for layer D.

Display Control Register (DCR)

Bit	31	30	29	28	27	26	25	24
Name	BLK	DCK	ADS	LPA	LD	LC	LB	LA

Bit	23	22	21	20	19	18	17	16
Name	DCKPS				HV	BUSERR	VSYNC	HSYNC

Bit	15	14	13	12	11	10	9	8
Name	—							

Bit	7	6	5	4	3	2	1	0
Name	—			LSB	MODE	NSYNC	CSEL	ESYNC

Bit	Name	Function	Reset Value	R/W
31	BLK	0 Blank screen (analog output) 1 Display frame buffer Note: In order to blank the digital outputs the PORT module has to be set to drive the corresponding PIOs. This can be achieved by setting the lower bits of the PAMUX register to logic zero. The data values specified in the corresponding bits of the PA register are output to the PIOs.	0	R/W
30	DCK	0 Selects external dot clock 1 Selects internal dot clock (divides SYSCLK)	0	R/W
29	—	Wired to zero.	0	R
28	LPA	0 Map mode (layer A, 8 bits per dot) 1 Picture mode (layer E, 16 bits per dot)	0	R/W
27	LD	0 Do not display layer D 1 Displays layer D	0	R/W
26	LC	0 Do not display layer C 1 Displays layer C	0	R/W
25	LB	0 Do not display layer B 1 Displays layer B	0	R/W
24	LA	0 Do not display layer A/E 1 Displays layer A/E	0	R/W
23:20	DCKPS	Dot Clock: 0_{hex} Not allowed 1_{hex} 1/4 system frequency 2_{hex} 1/6 system frequency 3_{hex} 1/8 system frequency 4_{hex} 1/10 system frequency 5_{hex} 1/12 system frequency 6_{hex} 1/14 system frequency 7_{hex} 1/16 system frequency 8_{hex} 1/18 system frequency 9_{hex} 1/20 system frequency A_{hex} 1/22 system frequency B_{hex} 1/24 system frequency Others 1/10 system frequency These bits are relevant only when DCK = 1.	0x4	R/W
19	HV	0 Interrupt at VSYNC* 1 Interrupt at HSYNC*	0	R/W
18	BUSERR	0 No bus error occurred since last write access to DCR 1 Bus error on video bus during GDC display data transfer When a bus error occurs on video bus this bit is set to 1. It can only be reset by a write access to DCR (any value for BUSERR bit).	0	R/W

Bit	Name	Function	Reset Value	R/W
17	VSYNC	0 Vertical scanning synchronization pulse period 1 Vertical scanning display period Note: In positive mode the synchronization period takes place when VSYNC equals logic 1.	0	R
16	HSYNC	0 Horizontal scanning synchronization pulse period 1 Horizontal scanning display period Note: In positive mode the synchronization period takes place when HSYNC equals logic 1.	0	R
15:5	—	Wired to zero.	0	R
4	LSB	0 Outputs 0 on LSB of R/G in pixel mode 1 Outputs 1 on LSB of R/G in pixel mode	0	R/W
3	MODE	0 Dot mode: Outputs dots and dot clock 1 Pixel mode: Outputs pixels serially and pixel clock	0	R/W
2	NSYNC	0 Normal mode: Outputs sync signals low active 1 Positive mode: Outputs sync signals high active	0	R/W
1	CSEL	0 Outputs VSYNC signal from VSYNC pin 1 Outputs CSYNC signal from VSYNC pin	0	R/W
0	ESYNC	0 Sets HSYNC and VSYNC pins to output mode 1 Sets HSYNC and VSYNC pins to input mode	1	R/W

Note: When outputting a digital RGB signal, you must set bits [15:0] of PAMUX register in module PORT to 1.

Start Address Register A/E, B, C, D (SARA/E, SARB, SARC, SARD)

Bit	31 21	20 3	2	1	0
Name	SA	AC	SS		

Bit	Name	Function	Reset Value	R/W
31:21	SA	Segment Address These bits specify the segment start address of each layer's frame buffer.	0	R/W
20:3	AC	Address Counter These bits specify the start address within the segment defined by SA of each layer's frame buffer.	0	R/W
2:0	SS	Dot Position These bits specify the first visible dot within the first word addressed by AC. Note that SS width differs for layer A/B and C/D and does not matter for layer E. The SS[2] bit is not relevant for layers A and B. Display Start Position Layer C/D Layer A/B 000 from dot 1 from dot 1 001 from dot 2 from dot 2 010 from dot 3 from dot 3 011 from dot 4 from dot 4 100 from dot 5 from dot 1 101 from dot 6 from dot 2 110 from dot 7 from dot 3 111 from dot 8 from dot 4 AC & SS specify the display start position for each layer in units of dots. Counting up AC & SS scrolls screen to the left. Counting down AC & SS scrolls screen to the right.	0	R/W

The specified start address SARx is not the same address as the address on the video bus. The lowest bits (SS bits) are used internally only (!). The differences are described in figure 12:

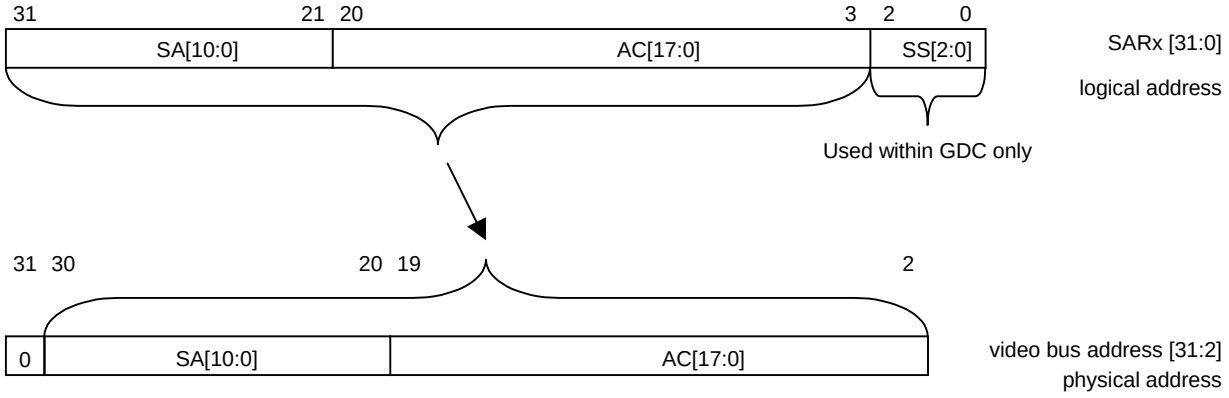


Figure 3.4.1 Logical Address and Physical Bus Address

Note: Bit 31 of video bus address is tied to 0.

Memory Width Register A/E, B, C, D (MWRA/E, MWRB, MWRC, MWRD)

Bit	31 26	25 16	15 0
Name	—	MW	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	MW	Specifies the line widths of each layer on the frame buffer in one-word units. For example, where the line width is 320 dots: MW = 28_{hex} for layer C and D MW = 50_{hex} for layer A and B MW = $A0_{hex}$ for layer E	0	R/W
15:0	—	Wired to zero	0	R

Horizontal Transfer Number Register (HTN) for layers A/E, B and C

Bit	31 21	20 16	15 0
Name	—	HTN	—

Bit	Name	Function	Reset Value	R/W
31:21	—	Wired to zero	0	R
20:16	HTN	Line Size of Layer A/E, B and C These specify the number of data transfers for one line. Sets the number of display dots for one line divided by 32.	0	R/W
15:0	—	Wired to zero	0	R

Note: HTN is always rounded to the higher integer value.

Horizontal Transfer Number Register (HTND) for layer D

Bit	31 21	20 16	15 0
Name	—	HTND	—

Bit	Name	Function	Reset Value	R/W
31:21	—	Wired to zero	0	R
20:16	HTND	Line Size of Layer D These specify the number of data transfers for one line. Sets the number of display dots for one line divided by 32.	0	R/W
15:0	—	Wired to zero	0	R

Note: HTND is always rounded to the higher integer value. It is for layer D only as layer D may have a different size.

Horizontal Display Start/ End Register (HDSER) for layers A/E, B and C

Bit	31 24	23 16	15 10	9 0
Name	—	HDS	—	HDE

Bit	Name	Function	Reset Value	R/W
31:24	—	Wired to zero	0	R
23:16	HDS	These specify the horizontal display start position using the number of dot clocks, starting from the HSYNC* signal falling edge.	0x2D	R/W
15:10	—	Wired to zero	0	R
9:0	HDE	These specify the horizontal display end position using the number of dot clocks, starting from the HSYNC* signal falling edge. Number of dots displayed during one period = HDE – HDS	0x15D	R/W

Note: The lowest value allowed for HDS and HDE is 2_{hex}.
Due to internal data structures effective values are HDS+9 and HDE+8.

Horizontal Display Start/ End Register (HDSERD) for layer D

Bit	31 24	23 16	15 10	9 0
Name	—	HDSD	—	HDED

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	HDSD	These specify the horizontal display start position using the number of dot clocks, starting from the HSYNC* signal falling edge.	0x2D	R/W
15:10	—	Wired to zero	0	R
9:0	HDED	These specify the horizontal display end position using the number of dot clocks, starting from the HSYNC* signal falling edge. Number of dots displayed during one period = HDED - HDSD	0x15D	R/W

Note: The lowest value allowed for HDSD and HDED is 2_{hex}.
Due to internal data structures effective values are HDS+9 and HDE+8.

Horizontal Cycle Register (HCR)

Bit	31 26	25 16	15 0
Name	—	HC	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	HC	Specifies the total number of dots and dot clocks within one horizontal cycle. HC = horizontal cycle time (µs) × dot clock frequency (MHz)	0x189	R/W
15:0	—	Wired to zero	0	R

Vertical Cycle Register (VCR)

Bit	31 26	25 16	15 0
Name	—	VC	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	VC	Specifies total number of lines within one vertical cycle. VC = vertical cycle time (ms) / horizontal cycle time (ms)	0x107	R/W
15:0	—	Wired to zero	0	R

Horizontal Synchronous Pulse Width Register (HSWR)

Bit	31 23	22 16	15 0
Name	—	HSW	—

Bit	Name	Function	Reset Value	R/W
31:23	—	Wired to zero	0	R
22:16	HSW	Specifies the horizontal sync signal pulse width using the number of dot clocks.	0x11	R/W
15:0	—	Wired to zero	0	R

Vertical Synchronous Pulse Width Register (VSWR)

Bit	31 21	20 16	15 0
Name	—	VSW	—

Bit	Name	Function	Reset Value	R/W
31:21	—	Wired to zero	0	R
20:16	VSW	Specifies the vertical sync signal pulse width using the number of lines.	0x03	R/W
15:0	—	Wired to zero	0	R

Vertical Display Start Register (VDSR) for layers A/E, B and C

Bit	31 22	21 16	15 0
Name	—	VDS	—

Bit	Name	Function	Reset Value	R/W
31:21	—	Wired to zero	0	R
21:16	VDS	These bits specify the vertical display start position using the number of lines, starting from the VSYNC* signal falling edge.	0x04	R/W
15:0	—	Wired to zero	0	R

Note: The lowest value allowed for VDS is 2_{hex}.

Vertical Display Start Register (VDSRD) for layer D

Bit	31 26	25 16	15 0
Name	—	VDSD	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	VDSD	These bits specify the vertical display start position using the number of lines, starting from the VSYNC* signal falling edge.	0x0F6	R/W
15:0	—	Wired to zero	0	R

Note: The lowest value allowed for VDSD is 2_{hex}.

Vertical Display End Register (VDER) for layers A/E, B and C

Bit	31 26	25 16	15 0
Name	—	VDE	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	VDE	These specify the vertical display end position using the number of lines, starting from the VSYNC* signal falling edge. Number of display period lines = VDE - VDS	0x0F6	R/W
15:0	—	Wired to zero	0	R

Note: The lowest value allowed for VDE is 2_{hex}.

Vertical Display End Register (VDERD) for layer D

Bit	31 26	25 16	15 0
Name	—	VDED	—

Bit	Name	Function	Reset Value	R/W
31:26	—	Wired to zero	0	R
25:16	VDED	These specify the vertical display end position using the number of lines, starting from the VSYNC* signal falling edge. Number of display period lines = VDED – VDSD	0x0F6	R/W
15:0	—	Wired to zero	0	R

Note: The lowest value allowed for VDED is 2_{hex}.

3.5 Setting Examples

Display characters: Horizontal synchronous cycle is 64 µsec/line,
Vertical synchronous cycle is 16.678 msec/screen.

If DOTCLK = 3 MHz ⇒ DCKPS = 0x9,
HC = 64 µs × 3 MHz = 192 dots,
VC = 16.768 ms × 1/64 µs = 262 lines.

When displaying a 160-dot × 225-line (32 dots × 32 lines for layer D) picture out of a 368 × 270-frame, and the frame buffer is SDRAM, then

SARA=0xA0000000, (physical: 0x50000000)
SARB=0xA0400000, (physical: 0x50200000)
SARC=0xA0800000, (physical: 0x50400000)
SARD=0xA0C00000, (physical: 0x50600000)
MWRA=0x5C,
MWRB=0x5C,
MWRC=0x2E,
MWRD=0x0A.

Please note that the defined start addresses are logical addresses. The physical start address is defined as the right shifted logical address and a zero as MSB (see chapter 3.4 'Start address register' Figure 3.4.1).

The following are example register settings while internal synchronization signals HSYNC and VSYNC are used:

HTN = 0x05,
HTND = 0x01,
HDS = 0x17,
HDE = 0xB7,
HDSD = 0x37,
HDED = 0x57,
HC = 0xC0,
VC = 0x106,
HSW = 0x0F,
VSW = 0x04,
VDS = 0x1B,
VDE = 0xFC,
VDSD = 0x3B,
VDED = 0x5B.

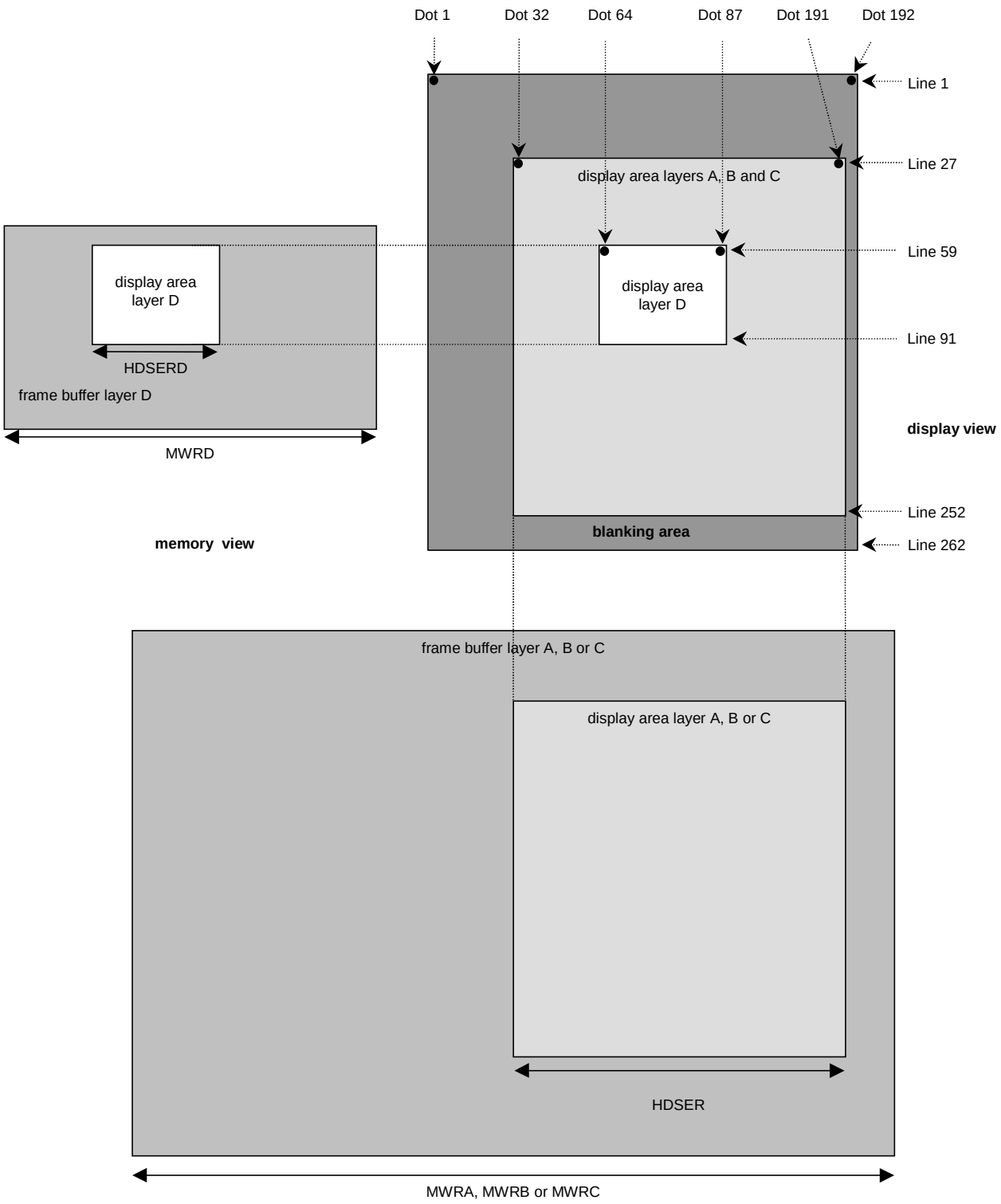


Figure 3.5.1 Picture Composition from Frame Buffer to Display Using Example Register Settings

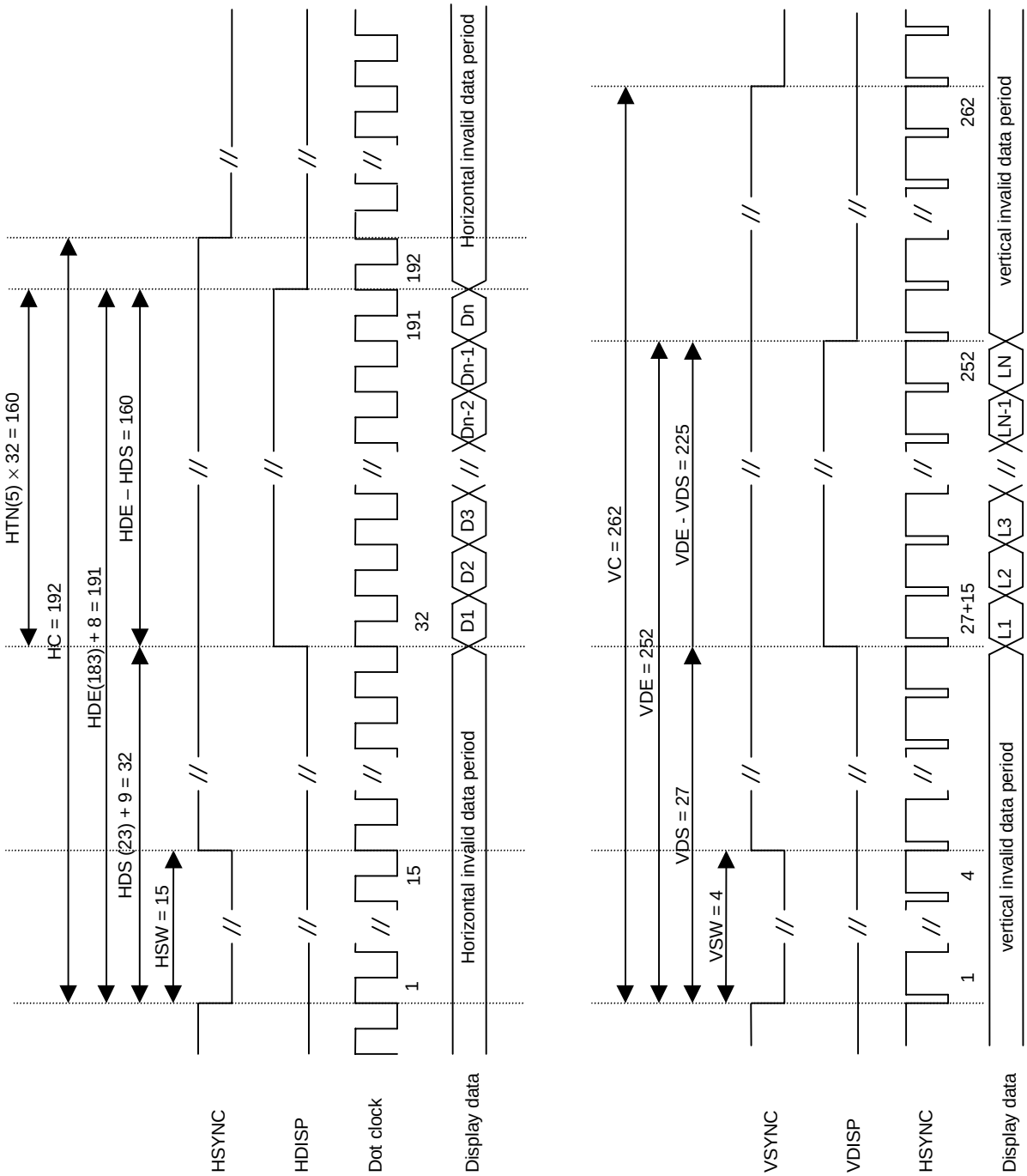


Figure 3.5.2 Display Signals Using Internal Dot Clock and Internal Sync Signals; signals HDISP and VDISP show active display area in horizontal and vertical direction respectively (Note that VDISP is an internal signal only)

The following are example register settings while external synchronization signals HSYNC and VSYNC are used:

Same as internal synchronization signals:

HTN = 0x05,
HTND = 0x01,
VDS = 0x1B,
VDE = 0xFC,
VDSD = 0x3B,
VDED = 0x5B.

Different from settings using internal synchronization signals:

HDS = 0x14,
HDE = 0xB4,
HDSD = 0x34,
HDED = 0x54.

Not relevant registers:

HC,
VC,
HSW,
VSW.

The display shows the same picture with the selected external synchronization signals HSYNC and VSYNC as in case of internal synchronization signal described before.

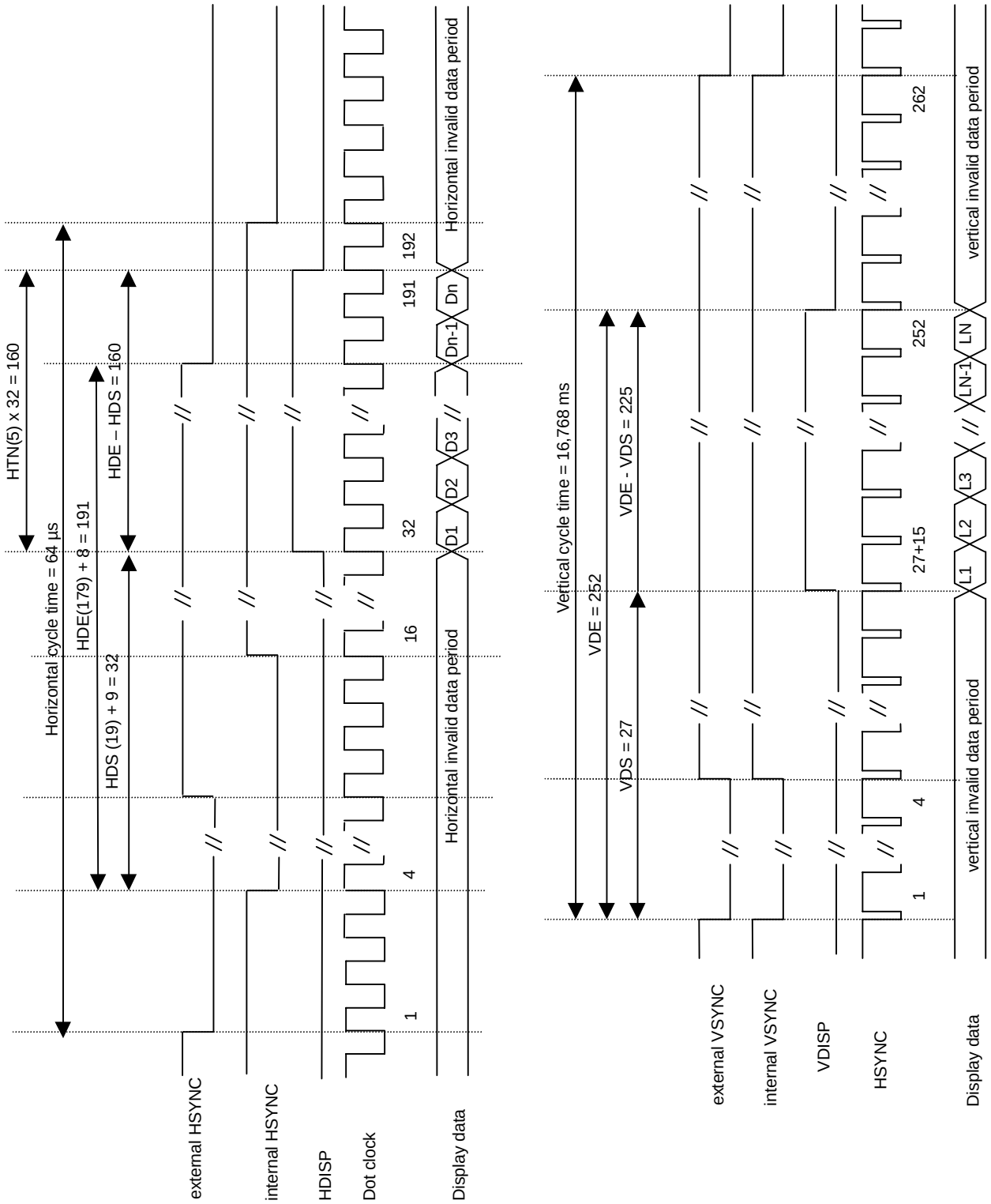


Figure 3.5.3 Display Signals Using Internal Dot Clock and External Sync Signals
(Note that VDISP is an internal signal only)

4. Interrupt Controller (INTC)

The Interrupt Controller has the following purpose:

- show the cause of an interrupt
- make it possible for software to mask all interrupts, except the NMI
- handle 3 external interrupt pins
- handle non-maskable-interrupt (NMI)

4.1 Basic Interrupt Handling

The following list shows the basic steps of an interrupt handler:

1. Change Status Register of CPU to inhibit interrupts with equal or lesser priority.
2. Read Cause Register of CPU to get cause of interrupt. (see also table below)
3. Read IRQR of Interrupt Controller to get more information about interrupt source.
4. Run interrupt routine.
5. Reset interrupt in source.
6. Reset bit of interrupt source in IRQR of Interrupt Controller.
7. Restore Status Register of CPU and jump back to program.

The following table shows all interrupt sources and the corresponding interrupt pin on the CPU:

Interrupt Source	Interrupt Pin of CPU
External interrupt 0	INT2
External interrupt 1	INT3
External interrupt 2	INT3
PWM timer	INT1
Periodic timer 0	INT0
Periodic timer 1	INT1

Interrupt Source	Interrupt Pin of CPU
DMAC	INT2
GDC	INT2
PORT	INT4
TXCAN	INT4
TXSEI	INT4
UART	INT5

4.2 Registers

Register Overview

Name	*Phys. Address (hex)*	Function
IRQR	*1C04 0000*	Indicates interrupt sources
IMASKR	*1C04 0004*	Enables/ disables interrupts
ILEXT	*1C04 0008*	Edge/ level detection of external interrupts

Interrupt Request Register (IRQR), Interrupt Mask Register (IMASKR)

IRQR: If an interrupt occurs, the corresponding bit is set to 1.
Writing 0 to a bit resets the contents.
Writing 1 to a bit does not change the contents.

IMASKR: 1 = enables interrupt
0 = disables interrupt (an incoming interrupt will be stored in IRQR, but no interrupt request will be sent to CPU)

Bit	31	30	29	28	27	26	25	24
Name	EXT2	EXT1	EXT0	GDC	DMAC1	DMAC0	T1	T0

Bit	23	22	21	20	19	18	17	16
Name	MPWM (1)	SEIEXC	CAN1EXC	CAN0EXC	SIO3EXC	SIO2EXC	SIO1EXC	SIO0EXC

Bit	15	14	13	12	11	10	9	8
Name	PWM	PORT	SEITX	SEIRX	CAN1TX	CAN1RX	CAN0TX	CAN0RX

Bit	7	6	5	4	3	2	1	0
Name	SIO3TX	SIO3RX	SIO2TX	SIO2RX	SIO1TX	SIO1RX	SIO0TX	SIO0RX

(1) Only in IMASKR, in IRQR this bit is wired to zero

Bit	Name	Cause of Interrupt	Reset Value	R/W
31	EXT2	External interrupt 2	0	R/W
30	EXT1	External interrupt 1	0	R/W
29	EXT0	External interrupt 0	0	R/W
28	GDC	Vertical or horizontal sync. on GDC	0	R/W
27	DMAC1	DMA on channel 1 finished or error on channel 1	0	R/W
26	DMAC0	DMA on channel 0 finished or error on channel 0	0	R/W
25	T1	Periodic timer 1	0	R/W
24	T0	Periodic timer 0	0	R/W
23	MPWM	Overflow on PWM timer enable (only in IMASKR, in IRQR this bit is wired to zero)	0	R/W
22	SEI EXC	Exception during TXSEI transfer	0	R/W
21	CAN1 EXC	Status change in TXCAN1	0	R/W
20	CAN0 EXC	Status change in TXCAN0	0	R/W
19	SIO3 EXC	Exception during serial I/O on UART channel 3	0	R/W
18	SIO2 EXC	Exception during serial I/O on UART channel 2	0	R/W
17	SIO1 EXC	Exception during serial I/O on UART channel 1	0	R/W
16	SIO0 EXC	Exception during serial I/O on UART channel 0	0	R/W

Bit	Name	Cause of Interrupt	Reset Value	R/W
15	PWM	PWM counter reached compare value or overflow of PWM counter if enabled by bit 23 of IMASKR	0	R/W
14	PORT	Interrupt from PORT module	0	R/W
13	SEI TX	Transmission on TXSEI finished	0	R/W
12	SEI RX	Reception on TXSEI finished	0	R/W
11	CAN1 TX	Transmission on TXCAN1 finished	0	R/W
10	CAN1 RX	Reception on TXCAN1 finished	0	R/W
9	CAN0 TX	Transmission on TXCAN0 finished	0	R/W
8	CAN0 RX	Reception on TXCAN0 finished	0	R/W
7	SIO3 TX	Transmission on UART channel 3 finished	0	R/W
6	SIO3 RX	Reception on UART channel 3 finished	0	R/W
5	SIO2 TX	Transmission on UART channel 2 finished	0	R/W
4	SIO2 RX	Reception on UART channel 2 finished	0	R/W
3	SIO1 TX	Transmission on UART channel 1 finished	0	R/W
2	SIO1 RX	Reception on UART channel 1 finished	0	R/W
1	SIO0 TX	Transmission on UART channel 0 finished	0	R/W
0	SIO0 RX	Reception on UART channel 0 finished	0	R/W

ILEXT

The register ILEXT controls edge or level detection of all external interrupt pins. When edge detection is enabled, an interrupt is caused on falling edge.

When level detection is enabled, an interrupt is caused on low level. On level detection the bits in IRQR show the current level of the external interrupt pin. Before you can clear the interrupt, the external interrupt signal must be set to 1.

Bit	31	30	29	28	27	0
Name	—	LEXT2	LEXT1	LEXT0	—	

Bit	Name	Function	Reset Value	R/W
31	—	Wired to zero	0	R
30	LEXT2	0 = Edge detection on external interrupt 2 (falling edge) 1 = Level detection on external interrupt 2 (low level)	0	R/W
29	LEXT1	0 = Edge detection on external interrupt 1 (falling edge) 1 = Level detection on external interrupt 1 (low level)	0	R/W
28	LEXT0	0 = Edge detection on external interrupt 0 (falling edge) 1 = Level detection on external interrupt 0 (low level)	0	R/W
27:0	—	Wired to zero	0	R

4.3 Non Maskable Interrupt

The TMPR3916 generates a non-maskable interrupt exception of the transition from high to low of the NMI* signal. To generate the next non-maskable interrupt exception, the NMI* signal must be set to high again and then to low.

The TX39 core completes the current bus operation, before it acknowledges the non-maskable interrupt exception. When the TX39 is not owner of the bus at the moment, the non-maskable interrupt occurs, the non-maskable interrupt must wait until TX39 regains the busmastership.

5. TIMER

The TIMER module contains the following features:

- Two periodic timers with variable intervals
- A PWM timer with variable interval / pulse width (PWM = pulse width modulation)

5.1 PWM Timer

The PWM timer contains a 16 bit counter. The TIMER module sends an interrupt, when the counter reaches a programmable compare value. In addition an interrupt can be sent on the overflow of the counter, if the bit 23 (MPWM) of IMASKR in the Interrupt Controller is set. The PWM counter starts after the bit 15 (PWM) of IMASKR in the Interrupt Controller is set. Setting bit 15 of IMASKR to 0 will clear the PWM Counter.

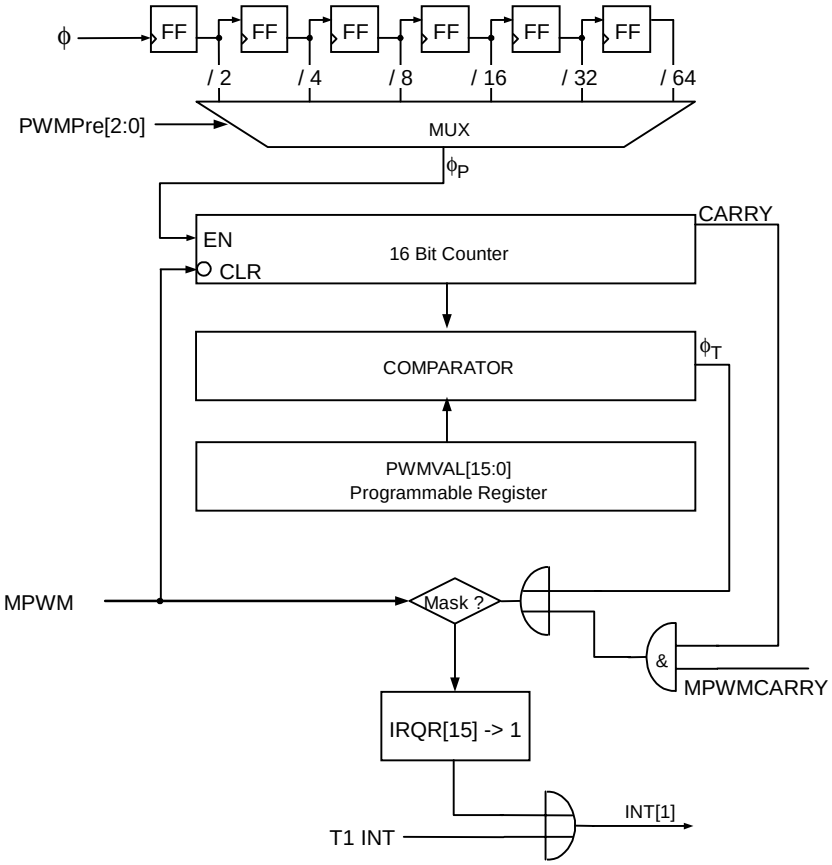


Figure 5.1.1 Block Diagram for Interrupt Generation

Use the following formula to calculate the time between start of the PWM counter and send of PWM interrupt (pulse width):

PWMVAL = compare value of PWM Timer (see register description of PWMVAL)

$$\text{Time to PWM Interrupt} = \frac{\text{PWMVAL}}{\text{System frequency}} \times 1 / \text{Prescaler}$$

Use the following formula to calculate the time between start and overflow of the PWM counter:

$$\text{Time to PWM Interrupt} = \frac{65536}{\text{System frequency}} \times 1 / \text{Prescaler}$$

Register Overview

Name	*Phys. Address (hex)*	Function
PWMVAL	*1C01 0008*	Compare value for 16 bit PWM counter

PWM Value Register (PWMVAL)

Bit	31 16	15 3	2	1	0
Name	PWMVAL	—	PWMPRE		

Bit	Name	Function	Reset Value	R/W
31:16	PWMVAL	Compare Value of PWM Counter If the PWM counter reaches the value of this register an interrupt is generated.	0x007F	R/W
15:3	—	Wired to zero	0	R
2:0	PWMPRE	Prescaler of PWM Counter The following clock is used to provide PWM counter: 000 = 1/2 system clock 001 = 1/4 system clock 010 = 1/8 system clock 011 = 1/16 system clock 100 = 1/32 system clock 101,110,111 = 1/64 system clock	101	R/W

5.2 Periodic Timers

Both periodic timers are running all the time. Every time they reach the end of the interval, they cause an interrupt. The interrupt is maskable in the Interrupt Controller (INTC). Both periodic timers use the same prescaler.

Use the following formula to calculate the interval:

T0INT = interval settings of periodic timer 0 (see register description of TITR)
T1INT = interval settings of periodic timer 1 (see register description of TITR)

$$\text{Interval of Timer 0} = \frac{2^{(T0INT + 8)}}{\text{System frequency}} \times 1 / \text{Prescaler}$$

$$\text{Interval of Timer 1} = \frac{2^{(T1INT + 9)}}{\text{System frequency}} \times 1 / \text{Prescaler}$$

Note that at the same settings timer 0 has twice the speed of timer 1.

Example:

T0INT = interval setting of Periodic Timer 0 = 010 (bin) = 2 (dec)
TPRE = prescaler setting = 100 (bin) ⇒ Prescaler = 1 / 32 (dec)

$$\text{Interval of Timer 0} = \frac{2^{(2 + 8)}}{60\ \text{MHz}} \times 32 = \frac{32768}{60\ \text{MHz}} = 0.55\ \text{msec}$$

Register Overview

Name	*Phys. Address (hex)*	Function
TIMER	*1C01 0000*	16 bit free-running counter
TITR	*1C01 0004*	Settings for periodic timer

TIMER Register (TIMER)

Bit	31 16	15 0
Name	TVAL	—

Bit	Name	Function	Reset Value	R/W
31:16	TVAL	Free running counter The free running counter provides the periodic timers. The counter operates continuously on output clock of the prescaler. Write this register only using full word access.	0x0000	R/W
15:0	-	Wired to zero	0x0000	R

TITR

Bit	31 27	26	25	24	23 19	18	17	16
Name	—	T1INT			—	T0INT		

Bit	15 3	2	1	0
Name	—	TPRE		

Bit	Name	Function	Reset Value	R/W
31:27	—	Wired to zero	0	R
26:24	T1INT	Interval of Periodic Timer 1 For interval calculation see above formula	000	R/W
23:19	—	Wired to zero	0	R
18:16	T0INT	Interval of Periodic Timer 0 For interval calculation see above formula	000	R/W
15:3	—	Wired to zero	0	R
2:0	TPRE	Prescaler for Timer 0 and Timer 1 The following clock is used to provide the periodic timer: 000 = 1/2 system clock 001 = 1/4 system clock 010 = 1/8 system clock 011 = 1/16 system clock 100 = 1/32 system clock 101, 110, 111 = 1/64 system clock The prescaler bits read back as zero.	101	W

6. Direct Memory Access Controller (DMAC)

The purpose of the Direct Memory Access Controller (DMAC) is to accelerate system speed and to make it easier for software to transfer data between memory and peripheral devices. The DMA Controller contains the following **features**:

- The DMAC consists of two independent channels: channel 0 and channel 1. Channel 0 has higher priority than channel 1 on GBUS transfers.
- DMA transactions are only possible from **device to memory** or from **memory to device**. Devices are for example UART and TXSEI.
- The DMA transfer will start after request of a peripheral device. That's why only data transfer will happen, when the device needs data. The UARTs, the TXSEI or an external device can send a request.
- At the end of a transaction or at the occurrence of an error an interrupt will be caused. The interrupt is only maskable in the Interrupt Controller (INTC).

The following **sequence** shows the principle use of the DMA Controller to transfer data between memory and a peripheral device:

1. Configure the DMA and the peripheral device. Set the number of Bytes to be transferred.
2. The DMA Controller waits for a request of the peripheral device.
3. The DMA Controller transfers data between memory and the peripheral device.
4. The DMA Controller sends an acknowledge to the peripheral device.
5. If there are more bytes left to transfer go to point 2.
6. The DMA Controller causes an interrupt.

6.1 Programming the DMA Controller

The programming of both channels are done in the same way. All register values are given in hexadecimal format.

6.1.1 Start DMA Transaction Between Memory and TXSEI/ UART

The DMA transaction between memory and TXSEI/UART is the main application of the DMA Controller. These transactions are only possible as **32-bit word transfers**. One data piece fills the lower bits of the memory word.

For example: The bytes 0x6E and 0x37 shall be transmitted over the UART. First these bytes have to be stored in bits [7:0] of the words in the memory. The first word in the memory is then 0x0000006E and the second word is 0x00000037.

In case of a receive, the data will be stored in the memory in the same way. The rest of the word (higher bits) will be filled with a random value.

Execute the following steps to start a DMA transaction:

1. Write the table value in
   - bits [3:0] of the Chip Configuration Register (CCR) for channel 0 :
   - bits [7:4] of the Chip Configuration Register (CCR) for channel 1 :

Device	Memory >> Device (transmit)	Device >> Memory (receive)
SIO 0	0x0	0x1
SIO 1	0x2	0x3
SIO 2	0x4	0x5
SIO 3	0x6	0x7
TXSEI	0x8	0x9

2. memory >> device: Write the start pointer of your sending data into the Source Address Register (SAR).

   Write the address of the device data register into the Destination Address Register (DAR).

   device >> memory: Write the address of the device data register into the Source Address Register (SAR).

   Write the start pointer of your receiving data into the Destination Address Register (DAR).

   Note: You must write the physical addresses into the registers SAR and DAR.
   All addresses have to be word aligned.

3. Write the number of bytes you want to transfer into the Byte Count Register (BCR).

   Note: The DMA Controller transfers a 32-bit word in every step. That's why the byte count must be at least 4. In addition, it must be possible to divide the contents of the Byte Count Register by 4.

4. memory >> device: Write the value 0xA9 into the Operation Definition Register (ODR).
   device >> memory: Write the value 0x69 into the Operation Definition Register (ODR).
   Write the value 0x11 into the Channel Control Register (CCR).

   Now the DMA Controller waits for a request of the external device to start the first transfer.

5. Configure and start the devices TXSEI or UART.

   After DMA transaction has finished, the DMA Controller sets the OPC bit (bit 0) in the Channel Status Register (CSR) and generates an interrupt. Each channel has its own interrupt signal. The interrupts are only maskable in the Interrupt Controller.

To terminate a running DMA transaction, set bit 22 (ABT) in the Channel Control Register (CCR).

6.1.2 Start DMA with External Device

The TMPR3916 has one pin for an external DMA request signal. For instance the external request makes sense together with a PORT module. The procedure is similar to a UART/ TXSEI transaction. For details of the procedure see section “Start DMA transaction between memory and TXSEI/UART” and the register description.

The following steps are executed to start a DMA transaction:

1. Write the hexadecimal-value 0xA in
   - bits [3:0] of the Chip Configuration Register (CCR) for channel 0
   - bits [7:4] of the Chip Configuration Register (CCR) for channel 1

2. Define source and destination address in registers SAR and DAR.

3. Write the number of bytes you want to transfer into BCR.

4. Write the following value into ODR:

Application	Memory >> Device (transmit)	Device >> Memory (receive)
Edge detection on external request	0xA8	0x68
Level detection on external request	0xA9	0x69

Write the following value into CCR:

Application	Memory >> Device (transmit)	Device >> Memory (receive)
Positive logic on external request	0x19	0x19
Negative logic on external request	0x11	0x11

Now the DMA Controller waits for a request of the external device to start the first transfer.

5. Configure and start external device.

   When the DMA Controller starts its first transfer, it sets the external acknowledge signal to 0. After finishing the whole DMA transaction, the DMA Controller sets the external acknowledge signal to 1.

6.1.3 Potential Problems

One of the following reasons can be responsible for an error:

- The software has made invalid settings in the Operation Definition Register (ODR) or in the Channel Control Register (CCR).
- The software has defined an address in the Source or Destination Address Register (SAR or DAR) that does not exist. The access to this address has caused a bus error.
- Either the byte count, the source address or the destination address is not correctly aligned.

The DMA Controller sends an interrupt, when an error occurs.

These steps need to be executed in case of an error:

1. Read the contents of the Channel Error Register (CER) to get the cause of the error.
2. For details see the register description.
3. Write 0x80 into the Channel Control Register (CCR).
4. Write 0x00 into the Channel Status Register (CSR).
5. Start the whole DMA programming procedure from the first point.

6.2 Registers

Overview:

Register (short name)	Physical Address (hex)	*Name*	Function
ODR0 ODR1	1C060000 1C060010	*Operation Definition Register*	Basic channel settings
CCR0 CCR1	1C060001 1C060011	*Channel Control Register*	Controls channel operations
CER0 CER1	1C060002 1C060012	*Channel Error Register*	Cause of an error or interrupt
CSR0 CSR1	1C060003 1C060013	*Channel Status Resister*	Channel operating status
SAR0 SAR1	1C060004 1C060014	*Source Address Register*	Specifies the address, where to get data from
DAR0 DAR1	1C060008 1C060018	*Destination Address Register*	Specifies the address, where to write data to
BCR0 BCR1	1C06000C 1C06001C	*Byte Count Register*	Specifies the number of transfer data in byte units

Operation Definition Register (ODR)

Bit	31	30	29	28	27	26	25	24
Name	SAC	DAC	PSIZ		OSIZ		MSIZ	BST

Bit	Name	Function	Reset Value	R/W
31	SAC	Specifies, if the Source Address Register should count up. 0 = Source address does not count up (device to memory transfer) 1 = Source address counts up on each transfer (memory to device transfer) This bit should be set to 0, when the DAC bit is set to 1.	0	R/W
30	DAC	Specifies, if the Destination Address Register should count up. 0 = Destination address does not count up (memory to device transfer) 1 = Source address counts up on each transfer (device to memory transfer) This bit should be set to 0, when the SAC bit is set to 1.	0	R/W
29:28	PSIZ	Specifies the size of the device data register. 00 = 8 bits, register bits [31:24] 01 = 16 bits, register bits [31:16] 10 = 32 bits, register bits [31:0] 11 = Setting prohibited	00	R/W
27:26	OSIZ	Specifies the width of DMA transfer. 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = Setting prohibited	00	R/W
25	MSIZ	Specifies the bus width of the memory. 0 = 32 bits 1 = 16 bits (there is no use for this setting)	0	R/W
24	BST	Specifies the detection mode for the request signal from the device. 0 = Edge detection 1 = Level detection For UART and TXSEI request signals use level detection.	0	R/W

Channel Control Register (CCR)

Bit	23	22	21	20	19	18	17	16
Name	RST	ABT	—	CEN	RPL	—	—	ETE

Bit	Name	Function	Reset Value	R/W
23	RST	Performs a software reset on all of the channel registers. 0 = Normal operation 1 = Executes software reset	0	R/W
22	ABT	Termination of channel operations regardless of the operating status. 0 = Normal operation 1 = Termination of current DMA transaction	0	R/W
21	—	Wired to zero.	0	R/W
20	CEN	Sets the channel to operating mode. 0 = Channel is inactive 1 = Start DMA transaction	0	R/W
19	RPL	Specifies the polarity of the device request signal. 0 = Negative logic (low level / falling edge) 1 = Positive logic (high level / rising edge) UART and TXSEI use negative logic.	0	R/W
18:17	—	Wired to zero	00	R
16	ETE	Decides if the request signal from a device is provided to the DMAC. 0 = No request signal is provided to the DMAC 1 = Request signal is provided to the DMAC Without setting this bit to 1, the DMA controller does not start a transaction.	0	R/W

Channel Error Register (CER)

Bit	15	14	13	12	11	10	9	8
Name	SWA	BER	ACE	CONF	SBE	DBE	BCE	AER

Bit	Name	Function	Reset Value	R/W
15	SWA	Software Abort: If this bit is set to 1, the software has terminated the current DMA transaction	0	R
14	BER	Bus Error: If this bit is set to 1, a bus error has occurred on DMA transfer.	0	R
13	ACE	Address or Count Error: (ACE = BCE + AER) If this bit is set to 1, a byte count error (BCE, bit 9) or an address boundary error (AER, bit 8) has occurred.	0	R
12	CONF	Configuration Error: If this bit is set to 1, the software has made invalid settings in the Operation Definition Register (ODR) or in the Channel Control Register (CCR).	0	R
11	SBE	Bus Error on Source Address Access: If this bit is set to 1, a bus error has occurred during read from a source.	0	R
10	DBE	Bus Error on Destination Address Access: If this bit is set to 1, a bus error has occurred during write to a destination.	0	R

Bit	Name	Function	Reset Value	R/W
9	BCE	Byte Count Error: If this bit is set to 1, the software has stored a value in the Byte Count Register, which is not correctly aligned. For example: When the DMA Controller transfers 32 bit words, it must be possible to divide the contents of the Byte Count Register by 4.	0	R
8	AER	Address Error: If this bit is set to 1, the software has stored a value in the Source- or Destination Address Register, which is not correctly aligned. For example: When the DMA Controller transfers 32 bit words, the lower two bits of the Source- and the Destination Address Register must be zero.	0	R

Channel Status Register (CSR)

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	ACT	EXC	OPC

Bit	Name	Function	Reset Value	R/W
7:3	—	Wired to zero	0	R
2	ACT	Channel Operation Status If this bit is set to 1, the channel is executing a DMA.	0	R
1	EXC	Exception If this bit is set to 1, an error has occurred during DMA transfer. The cause of error is shown in the Channel Error Register (CER). You can delete the exception by writing 0 to this bit. Writing 1 to this bit shows no effect.	0	R/W
0	OPC	Operation Successfully Completed If this bit is set to 1, the DMA transaction has finished without any errors.	1	R

Source Address Register (SAR)

Bit	31 24	23 0
Name	SAL	SAC

Bit	Name	Function	Reset Value	R/W
31:24	SAL	Base Address This part of the source address will not count up.	0	R/W
23:0	SAC	Address Offset This part of the source address will count up on every read, when the DMA Controller transfers data from memory to a peripheral device.	0	R/W

Destination Address Register (DAR)

Bit	31 24	23 0
Name	DAL	DAC

Bit	Name	Function	Reset Value	R/W
31:24	DAL	Base Address This part of the destination address will not count up.	0	R/W
23:0	DAC	Address Offset This part of the destination address will count up on every write, when the DMA Controller transfers data from a peripheral device to the memory.	0	R/W

Byte Count Register (BCR)

Bit	31 24	23 0
Name	—	BC

Bit	Name	Function	Reset Value	R/W
31:24	—	Unused	0	R
23:0	BC	These bits contain the number of bytes left to transfer. The DMA Controller decrements the contents of this register after every transfer.	0	R/W

7. CAN Module (TXCAN)

Outline and Features of TXCAN:

- 2.0 B active
- Standard identifier and remote frames
- Extended identifier and remote frames
- Full-CAN Controller
- 16 Mailboxes (15 Receive & Transmit + 1 Receive-only)
- Baud rate up to 1 MBit/sec on the CAN bus at minimum 8 MHz system clock
- Extended prescaler
- Bit Timing Parameter like Intel 82527™
- selectable mechanism for internal arbitration of transmit messages
- Time-Stamp for receive and transmit messages
- Readable Error Counters
- Warning Level IRQ, Error passive IRQ, Bus-off IRQ
- Local Loop Back Test Mode (Self Acknowledge)
- Programmable global mask for mailboxes 0-14
- Programmable local mask for mailbox 15
- Acceptance mask register for identifier extension bit
- Flexible interrupt structure
- Flexible status interface
- Sleep Mode
- Wake-up on CAN-bus activity or MCU access

7.1 Block Diagram

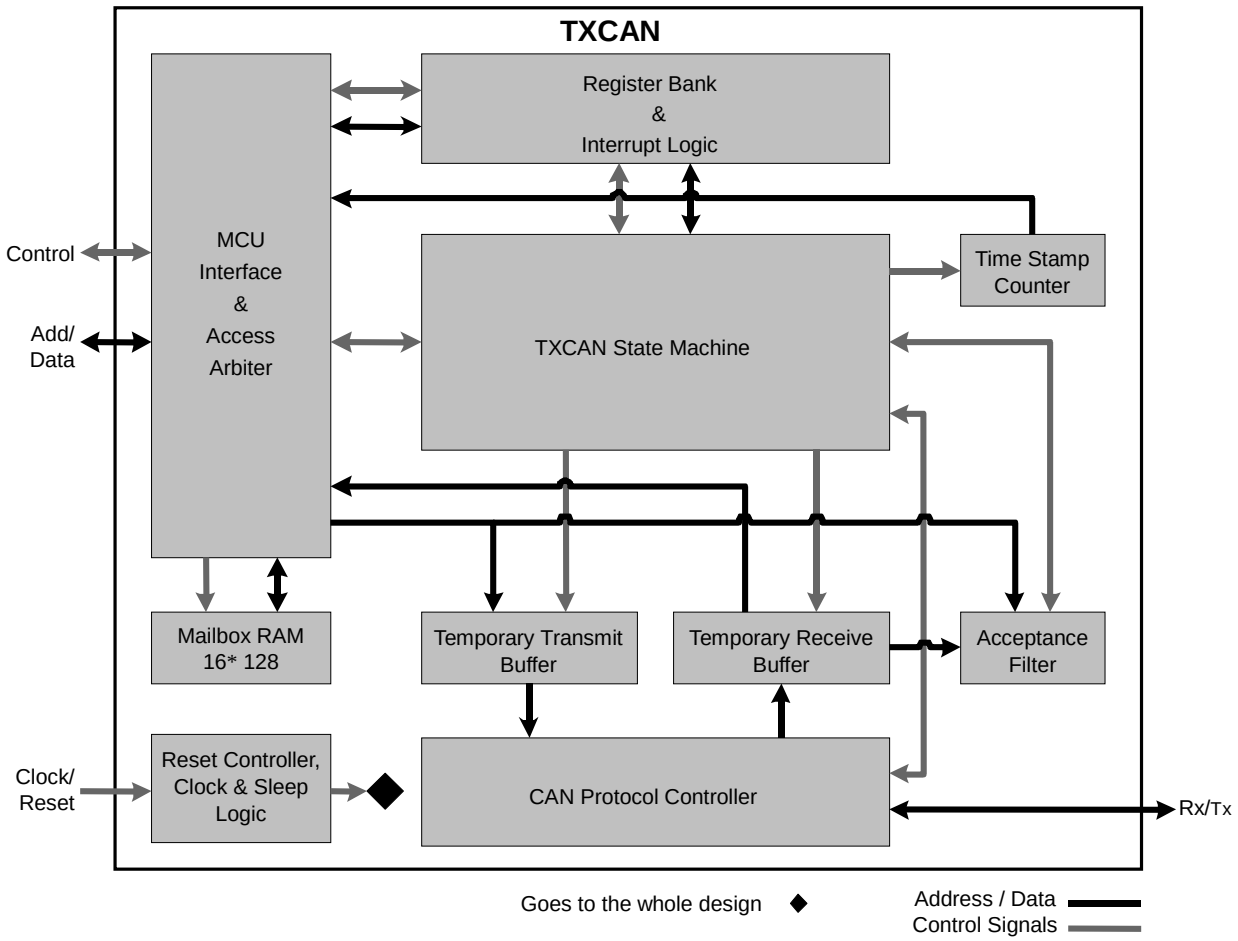


Figure 7.1.1 Block Diagram CAN Module

7.1.1 Message Buffers

The message storage is implemented in a single-port RAM, which can be addressed by the inner CAN core and the MCU. The MCU controls the CAN controller by modifying the various mailboxes in the RAM or the configuration registers.

In order to initiate a transfer, the transmission request bit has to be set in the corresponding register. Afterwards the entire transmission procedure and possible error handling is done without any MCU involvement. If a mailbox has been configured as receive the MCU reads the mailbox data using MCU read instructions. The mailbox can be configured to interrupt the MCU after every successful message transmission or reception.

The mailbox module provides 16 mailboxes of 8-byte data length, 29 bit identifier and several control bits. Each mailbox can be configured as either transmit or receive, except for mailbox 15. This mailbox is a receive-only buffer with a special acceptance mask designed to select groups of message identifiers to be received.

The mailbox area is implemented in a single-port-RAM.

7.1.2 Electrical CAN-Interface

The interface to the CAN bus is a simple two-wire line consisting of an input pin Rx and an output pin Tx. The pins are thought to operate with CAN bus transceivers according to ISO/DIS 11989 (e.g. Philips PCA 82C252, Bosch CF150 or Siliconix SI 9200).

7.2 TXCAN Registers

TXCAN Local Memory Map

physical base address channel 1 = 1C07 0000 (hex)

physical base address channel 2 = 1C07 8000 (hex)

Offset Address	*Name*	Description
0x000	*DPRAM*	Mailbox RAM (mailbox 0)
:		
0x0F0	*DPRAM*	Mailbox RAM (mailbox 15)
0x100	*MC*	Mailbox Configuration Register
0x104	*MD*	Mailbox Direction Register
0x108	*TRS*	Transmit Request Set Register
0x10C	*TRR*	Transmit Request Reset Register
0x110	*TA*	Transmission Acknowledge Register
0x114	*AA*	Abort Acknowledge Register
0x118	*RMP*	Receive Message Pending Register
0x11C	*RML*	Receive Message Lost Register
0x120	*LAM*	Local Acceptance Mask Register
0x124	*GAM*	Global Acceptance Mask Register
0x128	*MCR*	Master Control Register
0x12C	*GSR*	Global Status Register
0x130	*BCR1*	Bit Configuration Register 1
0x134	*BCR2*	Bit Configuration Register 2
0x138	*GIF*	Global Interrupt Flag Register
0x13C	*GIM*	Global Interrupt Mask Register
0x140	*MBTIF*	Mailbox Transmit Interrupt Flag Register
0x144	*MBRIF*	Mailbox Receive Interrupt Flag Register
0x148	*MBIM*	Mailbox Interrupt Mask Register
0x14C	*CDR*	Change Data Request
0x150	*RFP*	Remote Frame Pending Register
0x154	*CEC*	CAN Error Counter Register
0x158	*TSP*	Time Stamp Counter Prescaler
0x15C	*TSC*	Time Stamp Counter

7.2.1 Mailbox Structure

The following picture shows the structure of the mailbox RAM:

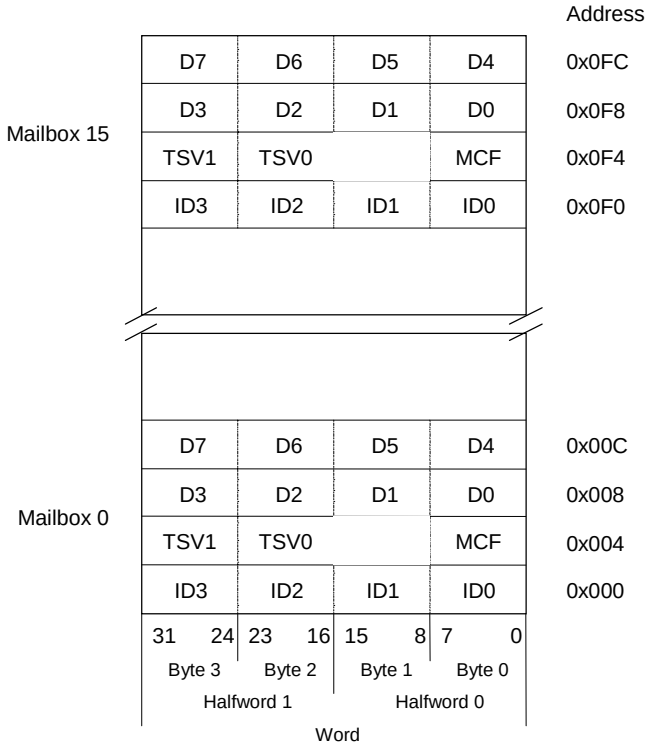


Figure 7.2.1 Mailbox RAM Structure

Each mailbox consists of 16 bytes. The first 4 bytes ID0 to ID3 contain the identifier. Byte 4 (MCF) contains the message control field and byte 5 is unused. Byte 6 and 7 are reserved for the time stamp value TSV of an implemented free running counter that indicates when a message was received or transmitted. The data field consists of the bytes D0 to D7.

One mailbox includes the following data:

- 29 bit identifier, 11 bit base ID and 18 bit extended ID (ID0-ID3)
- identifier extension bit (IDE) (ID3, bit 7)
- global (local) acceptance mask enable bit GAME (LAME) (ID3, bit 6)
- remote frame handling bit RFH (ID3, bit 5)
- remote transmission request bit (RTR) (MCF, bit 4)
- data length code (DLC) (MCF, bit 0-3)
- up to eight bytes for the data field (D0-D7)
- two bytes for the time stamp value (TSV)

Message Identifier (ID0 .. ID3)

Bit	31	30	29	28 18	17 0
Name	IDE	GAME	RFH	ID	

Bit	Name	Function
31	IDE	Kind of frame (length of identifier) 0 = Standard frames (CAN 2.0A), 11 bit identifier 1 = Extended frames (CAN 2.0B), 29 bit identifier
30	GAME	Use of global acceptance mask (mailbox 0 to 14) 0 = The received message will only be stored, when the received identifier is identical to that in the mailbox. 1 = The global acceptance mask will be used for acceptance filtering. The bit 30 of mailbox 15 is called LAME and determines, if the local acceptance mask will be used for acceptance filtering. Always set GAME to “0” for transmit mailboxes.
29	RFH	Remote Frame Handling (only for transmit mailboxes) 0 = Software must handle remote frames 1 = The mailbox will automatically respond to remote frames
28:18	ID	Identifier Contains standard identifier or first bits of extended identifier
17:0	ID	Identifier Contains the last bits of extended identifier

Time Stamp Value (TSV)/Message Control Field (MCF)

Bit	31 16
Name	TSV

Bit	15 5	4	3	2	1	0
Name	—	RTR	DLC			

Bit	Name	Function
31:16	TSV	Time-Stamp Counter Value TXCAN contains a 16 bit time-stamp-counter. The value of the time-stamp-counter is stored after the reception and after the transmission of a message. Please refer to chapter “7.2.8 Time Stamp Feature” for a full description of the time-stamp counter’s function.
15:5	—	No function
4	RTR	Remote Frame Transmission Request 0 = Normal frame 1 = Remote frame
3:0	DLC	Data Length These bits contain the number of data bytes transferred by the frame. Only the values 0000(bin) to 1000(bin) are allowed. If these bits are set to 1001(bin) or more, the TXCAN will send 8 data bytes.

7.2.2 Control Registers

Mailbox Configuration Register (MC)

Bit	15	0
Name	MC	

Bit	Name	Function	Reset Value	Mode
15:0	MC	Mailbox Enable 0 = The corresponding mailbox MBn is disabled for the CAN module and the write access to the identifier field of the mailbox is possible. 1 = The mailbox is enabled for the TXCAN state machine. Write access to the identifier field of an enabled mailbox is denied. Write access to the data field and control field of a mailbox is always possible. After power-up, all bits in MC are cleared and all mailboxes are disabled.	0	R/W

Special care is required during reprogramming of MC in operation:

For a receive mailbox it needs to be ensured that the mailbox is not being disabled while a reception for this mailbox is ongoing. If a mailbox is disabled during an ongoing reception, the user must be aware that the current frame might be received, even though the mailbox has been disabled or reconfigured during the reception of the incoming CAN frame.

A transmit mailbox may never be disabled, when a transmit request is pending.

Mailbox Direction Register (MD)

Bit	15	14	0
Name	MD15	MD14 .. MD0	

Bit	Name	Function	Reset Value	Mode
15	MD15	Mailbox Direction of Mailbox 15 Mailbox 15 is receive-only. This bit is always 1 and cannot be changed.	1	R
14:0	MD14 to MD0	Mailbox Direction of Mailboxes 14 to 0 Each mailbox can be configured as transmit mailbox or receive mailbox. 0 = Transmit mailbox 1 = Receive mailbox	0	R/W

The direction of mailboxes may not be changed in operation. A mailbox needs to be disabled before its direction may be changed.

7.2.3 Message Transmission

The transmission control consists of two registers. One register for setting (TRS) and one for resetting (TRR) the transmission request. In this manner it is possible to clear the transmission request without generating a conflict in the handling of the transmit mailboxes in the state-machine. This mechanism also prevents the clearing of the transmission request of a mailbox which transmission is already in progress.

The data to be transmitted will be stored in a mailbox configured as transmit mailbox (MDn = 0). After writing the data and the identifier into the mailbox RAM, the message will be sent if the corresponding TRS bit has been set and the mailbox is enabled (MCn = 1).

If there is more than one mailbox configured as transmission mailbox and more than one corresponding TRS bit is set, then the messages will be sent in the selected order. The order of transmission can be selected in the Master Control Register (MCR).

If MTOS is set to “0”, the mailbox with the lower number has the higher priority. For example: if the mailboxes MB0, MB2 and MB5 are configured for transmission and the corresponding TRS bits are set, then the messages will be transmitted in the following order: MB0, MB2 and MB5. If a new transmission request is set for MB0 during the processing of MB2 then in the next internal arbitration-run MB0 will be selected for the next transmission. This will also happen, when the TXCAN loses arbitration while transmitting MB2. In this case, MB0 will be sent at the next opportunity instead of MB2.

If MTOS is set to “1”, the priority of the identifier stored in the mailbox will determine the sending order. The mailbox with the higher priority identifier will be sent first.

In case of a lost arbitration on the CAN bus line a new internal arbitration run will be started and the message with the highest priority will be sent at the next possible time.

Transmission Request Set Register (TRS)

Bit	15	14	0
Name	—	TRS	

Bit	Name	Function	Reset Value	Mode
15	—	Wired to zero	0	R
14:0	TRS	Setting TRSn causes the particular message “n” to be transmitted. Several bits can be set simultaneously. The messages will be sent one after the other in the selected transmission order. The transmission order can be selected by the MTOS bit in the Master Control Register (MCR). The bits in TRS will be set by writing “1” at the corresponding bit position from the MCU. Writing a “0” has no effect. After power-up, all bits are cleared.	0	R/S

The TRS bits can only be set by the MCU and will be reset by internal logic in case of a successful transmission or an aborted transmission (if requested by setting the corresponding TRR bit) or a hard/software reset. Bit 15 is not implemented because the mailbox 15 is the receive-only mailbox. If a mailbox is configured as receive the corresponding bit in TRS can not be set by MCU.

Transmission Request Reset Register (TRR)

Bit	15	14	0
Name	—	TRR	

Bit	Name	Function	Reset Value	Mode
15	—	Wired to zero	0	R
14:0	TRR	Setting TRRn causes a transmission request to be cancelled that was initiated by the corresponding bit TRSn, provided that the transmission of this mailbox is not currently in process. If the corresponding message is currently processed the bit will be reset in the following cases: a successful transmission (normal operation), an aborted transmission in case of a lost arbitration or an error condition detected on the CAN bus line. In case of an aborted transmission, the corresponding status bit AAn will be set and in case of a successful transmission, the status bit TAn will be set. The bits in TRR will be set by writing a “1” from the MCU. Writing a “0” has no effect. After power-up, all bits are cleared.	0	R/S

These bits can only be set by the MCU and reset by the internal logic. They will be reset by internal logic in case of a successful transmission or an aborted transmission. Bit 15 is not implemented because the mailbox 15 is the receive-only mailbox. If TRRn is set the write access to the corresponding mailbox is denied. If a mailbox is configured as receive the corresponding bit in TRR cannot be set by MCU.

Note: When TRSn is set, after setting TRRn to “1”:

- A transmission request of a message, which is not currently in process, will be cleared immediately (TRSn → 0, TRRn → 0, AAn → 1).
- A transmission request of a message which is currently processed will be cleared in case of a lost arbitration or an error condition on the CAN bus (TRSn → 0, TRR → 0, AAn → 1).
- A transmission request of a message which is currently processed will not be cleared if there is no lost arbitration and no error condition on the CAN bus (TRSn → 0, TRRn → 0, TAn → 1).

Transmission Acknowledge Register (TA)

Bit	15	14	0
Name	—	TA	

Bit	Name	Function	Reset Value	Mode
15	—	Wired to zero	0	R
14:0	TA	If the message of mailbox “n” has been transmitted successfully, the bit “n” of this register will be set and a transmission successful interrupt is generated, if it is enabled. The bits in TA will be reset by writing a “1” from the MCU to TA or TRS. Writing a “0” has no effect. After power-up, all bits are cleared.	0	R/C

Abort Acknowledge Register (AA)

Bit	15	14	0
Name	—	AA	

Bit	Name	Function	Reset Value	Mode
15	—	Wired to zero	0	R
14:0	AA	If the transmission of the message in mailbox “n” has been aborted, the bit “n” of this register will be set and a transmission abort interrupt is generated, if it is enabled. The bits in AA will be reset by writing a “1” from the MCU to AA or TRS. Writing a “0” has no effect. After power-up, all bits are cleared.	0	R/C

Change Data Request (CDR)

Bit	15	14	0
Name	—	CDR	

Bit	Name	Function	Reset Value	Mode
15	—	Wired to zero	0	R
14:0	CDR	If the CDR bit of a transmit mailbox is set, a transmission request for this mailbox will be ignored. That means, that a mailbox with TRS and CDR set will not be considered in the internal arbitration-run: the mailbox is locked for transmission. The processing of this mailbox in the arbitration-run will be considered again after clearing the CDR bit. After power-up, all bits are cleared.	0	R/W

CDR is useful for dealing with remote frames. It is intended for updating the data field of a transmit mailbox, which is configured for automatic reply to remote frames (RFH bit set). By using the CDR bit, the user can update the data field without a need of taking additional care of the data consistency.

See also section “7.5 Handling of Message-Objects”.

7.2.4 Message Reception

The identifier of each incoming message is compared to the identifiers held in the receive mailboxes. The comparison of the identifiers depends on the value of the global/local acceptance mask enable bit (GAME/LAME) stored in the mailbox and the data held in the global/local acceptance mask (GAM/LAM). When a matching identifier is detected, the received identifier, the control bits and the data bytes are written into the matching RAM location. At the same time the corresponding receive message pending bit RMPn is set and a receive-interrupt is generated, if it is enabled. After finding a matching identifier, no further compare will be done. If no match is detected, the message is rejected. The RMP bit has to be reset by the MCU after reading the data. If a second message has been received for this mailbox and the RMP bit is already set, the corresponding message lost bit (RML) is set. In this case, the stored message will be overwritten with the new data.

Only if an incoming message does not match to one of the mailboxes 0 to 14, this message will be stored in the receive-only mailbox in case of a matching identifier (acceptance filter).

The following figure shows the timing of the flags and the write to the mailbox during message reception:

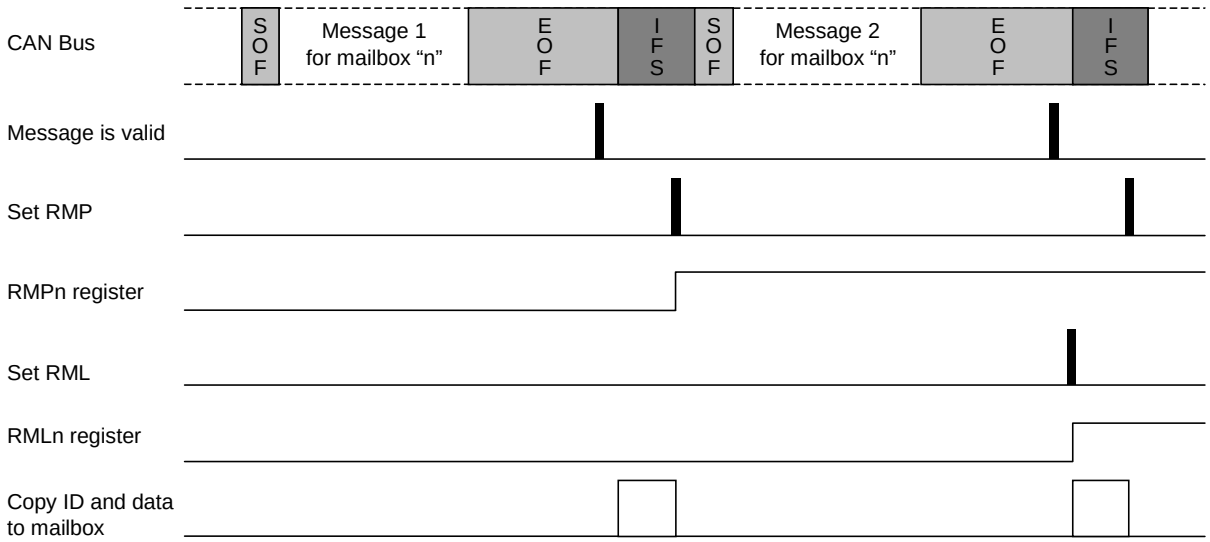


Figure 7.2.2 Timing for Writing Received Message to Mailbox, Including Flags

Receive Message Pending Register (RMP)

Bit	15	0
Name	RMP	

Bit	Name	Function	Reset Value	Mode
15:0	RMP	If mailbox “n” contains a received message, bit RMPn of this register will be set. These bits can only be reset by the MCU, and set by the internal logic. A new incoming message will overwrite the stored one. In this case, the corresponding status bit RMLn will be set before overwriting begins. The bits in RMP and RML can be cleared by a write access to the register RMP with a “1” at the corresponding bit location. After power-up, all bits are cleared.	0	R/C

Receive Message Lost Register (RML)

Bit	15	0
Name	RML	

Bit	Name	Function	Reset Value	Mode
15:0	RML	If there is an overload condition for mailbox “n”, bit RMLn of this register will be set. These bits can only be reset by the MCU, and set by the internal logic. The bits can be cleared by a write access to the register RMP with a “1” at the corresponding bit location. After power-up, all bits are cleared.	0	R/C

See also section “7.5 Handling of Message-Objects”.

7.2.5 Remote Frame Handling

If a remote frame has been received, the internal FSM will compare the identifier to all identifiers of the mailboxes. The comparison of the identifiers depends on the value of the bit global/local acceptance mask enable (GAME/LAME) stored in the mailbox and the data held in the global/local acceptance mask (GAM/LAM).

If there is a matching identifier and the RFH bit in this mailbox is set and this mailbox is configured as transmit, this message object will be marked as “to be sent” (TRS will be set).

If there is a matching identifier and the mailbox is configured as receive, this message will be handled like a data frame and the corresponding bit in RMP and RFP will be set.

After finding a matching identifier, no further compare will be done.

Remote Frame Pending Register (RFP)

Bit	15	0
Name	RFP	

Bit	Name	Function	Reset Value	Mode
15:0	RFP	If a remote frame is received in a mailbox configured as receive mailbox, the corresponding bits in RFPn and RMPn are set. The bits in RFP can be cleared by writing a “1” to the corresponding bit position in RMP. Writing a “0” has no effect. If a remote frame in the mailbox is overwritten by a data frame, the corresponding bit in RFP is cleared. After power-up, all bits are cleared.	0	R/W

See also section “7.5 Handling of Message-Objects”.

7.2.6 Acceptance Filtering

For the mailboxes 0 to 14, the global acceptance mask (GAM) will be used if the bit GAME in the mailbox is set. An incoming message will be stored in the first mailbox with a matching identifier. Only if there is no matching identifier in the mailboxes 0 to 14, the incoming message will be compared to the receive-only mailbox (mailbox 15). If the LAME bit in mailbox 15 is set, the local acceptance mask (LAM) will be used. The acceptance code in the figure below is the content of the identifier words of the current mailbox.

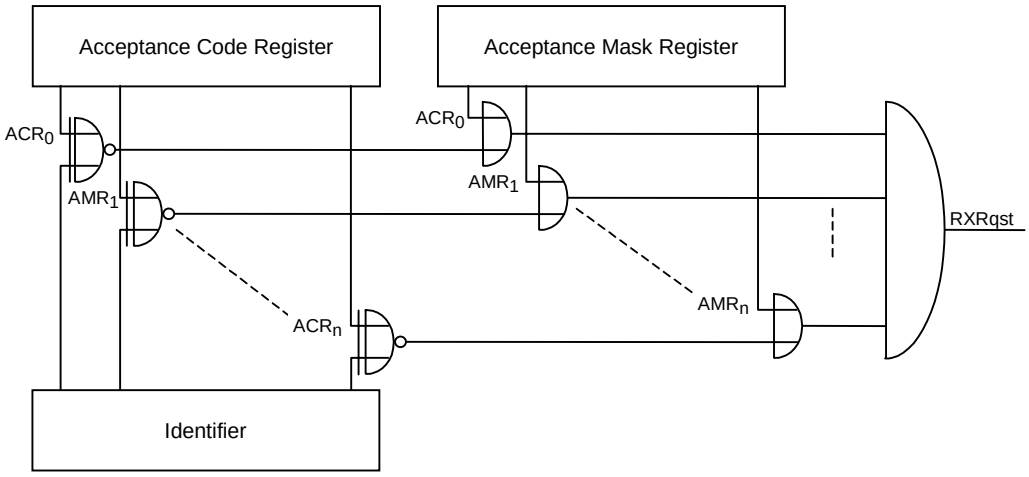


Figure 7.2.3 Acceptance Filter Logic

Local Acceptance Mask (LAM)

The local acceptance mask register will only be used for filtering messages for mailbox 15. This feature allows the user to locally mask, or “don’t care”, any identifier bits of the incoming message for mailbox 15.

Bit	31	30	29	28	0
Name	LAMI	—		LAM	

Bit	Name	Function	Reset Value	Mode
31	LAMI	0 = The identifier extension bit stored in the mailbox determines which messages shall be received. 1 = Don’t care: standard and extended frames can be received. In case of an extended frame all 29 bits of the identifier stored in the mailbox and all 29 bits of the local acceptance mask register will be used for the filter. In case of a standard frame, only the first eleven bits (bit 28 to 18) of the identifier and the local acceptance mask will be used.	0	R/W
30:29	—	Wired to zero	0	R
28:0	LAM	Incoming messages are first checked for an acceptance match in mailbox 0 to 14 before passing through the mailbox 15. A “1” value means, “don’t care” or accept a “0” or “1” for that bit position. A “0” value means that the incoming bit value must match identically to the corresponding bit in the message identifier. The global mask has no effect for mailbox 15. After power-up, all bits are cleared.	0	R/W

For messages in extended format the identifier extension bit and the whole 29 bits of the identifier will be compared and for messages in standard format only the first 11 bits and the identifier extension bit will be compared.

The local acceptance mask will only be used for mailbox 15 (receive-only mailbox).

Global Acceptance Mask (GAM)

Bit	31	30	29	28	0
Name	GAMI	—		GAM	

Bit	Name	Function	Reset Value	Mode
31	GAMI	0 = The identifier extension bit stored in the mailbox determines which messages shall be received. 1 = Don't care, standard and extended frames can be received. In case of an extended frame all 29 bits of the identifier stored in the mailbox and all 29 bits of the global acceptance mask register will be used for the filter. In case of a standard frame, only the first eleven bits (bit 28 to 18) of the identifier and the global acceptance mask will be used.	0	R/W
30:29	—	Wired to zero	0	R
28:0	GAM	For each incoming message, the global acceptance mask will be used if the bit GAME is set. A received message will only be stored in the first mailbox with a matching identifier.	0	R/W

The global acceptance mask will only be used for the mailboxes 0 to 14. After power-up, all bits are cleared.

Master Control Register (MCR)

Bit	15	14	13	12	11	10	9	8
Name	—	—	—	—	SUR	INTLB	TSTLB	TSTERR

Bit	7	6	5	4	3	2	1	0
Name	CCR	SMR	TSBTEST	WUBA	MTOS	—	TSCC	SRES

Bit	Name	Function	Reset Value	Mode
15:12	—	Wired to zero	0	R
11	SUR	Suspend Mode Request 0 = Normal operation requested 1 = Suspend mode is requested	0	R/W
10	INTLB	Internal Loop Back Enable 0 = Internal loop back is disabled in test mode 1 = Internal loop back is enabled in test mode	0	R/W
9	TSTLB	Test Loop back 0 = Normal operation requested 1 = Test loop back mode is requested. This mode supports stand-alone operation.	0	R/W
8	TSTERR	Test Error 0 = Normal operation requested 1 = Test error mode is requested. In this mode, it is possible to write the error counters (CEC).	0	R/W
7	CCR	Change Configuration Request. 0 = Normal operation requested 1 = Write access to the configuration registers (BCR1 and BCR2) requested.	1	R/W
6	SMR	Sleep Mode Request 0 = The sleep mode is not requested (normal operation). 1 = The sleep mode is requested.	0	R/W

Bit	Name	Function	Reset Value	Mode
5	TSBTEST	Toshiba Internal Test Mode Always write this register as “0”.	0	R/W
4	WUBA	Wake Up on Bus Activity 0 = The module leaves the sleep mode only by detecting a write access to MCR 1 = The module leaves the sleep mode by detecting any bus activity or by detecting a write access to MCR.	0	R/W
3	MTOS	Mailbox Transmission Order Select 0 = Mailbox transmission order by mailbox number. The mailbox with the lower number will be sent first. 1 = Mailbox transmission order by identifier priority. The mailbox with the higher priority identifier will be sent first.	0	R/W
2	—	Wired to zero	0	R
1	TSCC	Time Stamp Counter Clear 0 = No effect 1 = The time stamp counter will be cleared. This bit can only be written and will always be read as zero.	0	R/W
0	SRES	Software Reset 0 = No effect 1 = A write access to this register causes a software reset of the module (All parameters will be reset to their initial values). This bit can only be written and will always be read as zero.	0	R/W

7.2.7 Bit Configuration Registers

Bit Configuration Register 1 (BCR1)

Bit	15 8	7 0
Name	—	BRP

Bit	Name	Function	Reset Value	Mode
15:8	—	Wired to zero	0	R
7:0	BRP	BRP is the value of the baud rate prescaler.	0	R/W

Bit Configuration Register 2 (BCR2)

Bit	15	14	13	12	11	10	9	8
Name	—						SJW	

Bit	7	6	5	4	3	2	1	0
Name	SAM	TSEG2			TSEG1			

Bit	Name	Function	Reset Value	Mode
15:10	—	Wired to zero	0	R
9:8	SJW	Indicates by how many units of time-quantums a bit is allowed to be lengthened or shortened when re-synchronising. 00 = 1 time quantum 01 = 2 time quantums 10 = 3 time quantums 11 = 4 time quantums	0	R/W
7	SAM	Sample point setting (see below)	0	R/W
6:4	TSEG2	Timing setting for sampling point (see below)	0	R/W
3:0	TSEG1	Timing setting for sampling point (see below)	0	R/W

The length of a bit is determined by the parameters TSEG1, TSEG2 and BRP. All controllers on the CAN bus must have the same baud rate and bit length. At different clock frequencies of the individual controllers, the baud rate has to be adjusted by the mentioned parameters. In the bit timing logic, the conversion of the parameters to the required bit timing is realized. The configuration registers (BCR1, BCR2) contain the data about the bit timing. Its definition corresponds to the CAN specification 2 (like Intel 82527). The register content is zero after a reset.

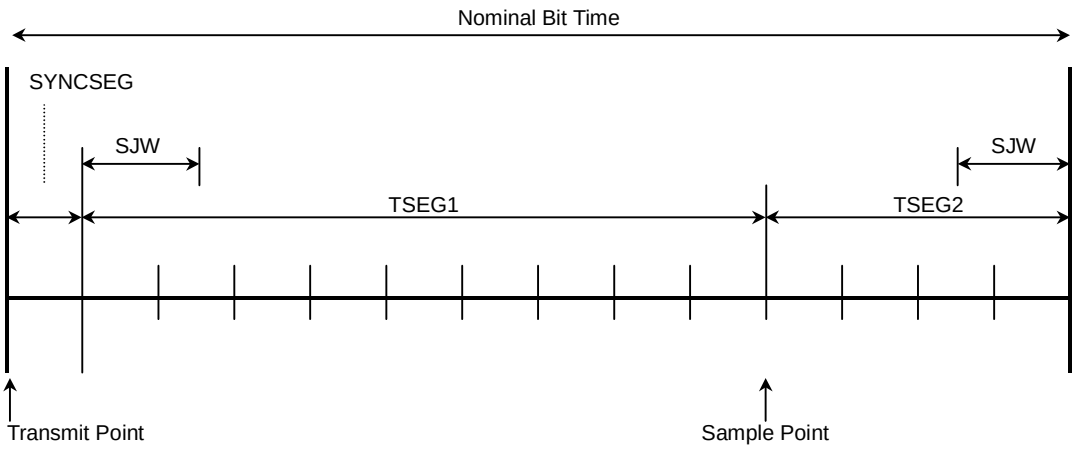


Figure 7.2.4 Required Timing Parameters for CAN Transmission

The length of T_{SCL} (CAN Bus System Clock) is defined by:

$$T_{SCL} = \frac{BRP + 1}{f_{OSC}}$$

$$1 * T_{SCL} = 1 * T_Q \qquad (T_Q = \text{time quantum})$$

f_{OSC} is the TXCAN clock frequency (input clock of the TXCAN module). This clock is the output of a prescaler with a divider ratio from 2 up to 8. This ratio can be set in the Chip Configuration Register (CCR). The default setting is 3.

The synchronization segment SYNCSEG has always the length of one T_{SCL}.

The baud rate is defined by:

$$BR = \frac{1}{((TSEG1 + 1) + (TSEG2 + 1) + 1) * T_{SCL}}$$

IPT (information processing time) is the time segment starting with the sample point reserved for processing of the sampled bit level. The information processing time is equal to 3 TXCAN system clock cycles.

The parameter SJW (2 bits) indicates, by how many units of T_Q a bit is allowed to be lengthened or shortened when re-synchronizing. Values between “1” (SJW = 00b) and “4” (SJW = 11b) are adjustable. The bus line is sampled and a synchronization is performed at each falling edge of the bus signal within a bit grid.

With the corresponding bit timing, it is possible to reach a multiple sampling of the bus line at the sample point by setting SAM. The level determined by the CAN bus then corresponds to the result from the majority decision of the last three values. The sample points are at the rising edges of the external SamPoint signal and twice before with a distance of one TXCAN system clock cycle.

This leads to the following restrictions:

Restrictions for TSEG2

BRP	T_Q Length (TXCAN clock cycles)	IPT Length (TXCAN clock cycles)	TSEG2 Minimum Length (in T_Q)
0	1	3	3
1	2	3	2
>1	BRP+1	3	2

Restrictions for TSEG1

The length of TSEG1 should be equal or greater than the length of TSEG2:

$TSEG1 \geq TSEG2$

Restrictions for SJW

The maximum length of the synchronization jump width is equal to the length of TSEG2:

$SJW \geq TSEG2$

Restrictions for SAM

The three-time sampling is not allowed for BRP<4. For BRP<4 always a one-time sampling will be performed regardless of the value of SAM.

Example:

A transmission rate of 1 MBit/s will be adjusted, i.e. a bit has a length of 1 µs. The clock frequency f_{OSC} is 10 MHz. The baud rate prescaler is set to "0". That means a bit for this data transmission rate has to be programmed with a length of $10*T_Q$. According to the above formula, the values to be set are always by one smaller than the calculated values.

E.g. BRP = 1 (BRP_reg = 0);

TSEG1 = 5 (TSEG1_reg = 4), TSEG2 = 4 (TSEG2_reg = 3).

With this setting a threefold sampling of the bus is not possible (BRP<4), thus SAM = 0 should be set. SJW is not allowed to be greater than TSEG2, so the maximum value could be set to 4 units (SJW = 3).

7.2.8 Time Stamp Feature

There is a free-running 16-bit timer implemented in the module to get an indication of the time of reception or transmission of messages. The content of the timer is written into the time stamp register of the corresponding mailbox (TSV) when a received message has been stored or a message has been transmitted.

The counter is driven from the bit clock of the CAN bus line. When the TXCAN is in configuration mode or in sleep mode, the timer will be stopped. After power-up reset the free running counter can be cleared by writing a value to the time stamp counter prescaler. The counter can be written and read by the MCU in configuration mode and in normal operation mode.

Time Stamp Counter Register

Bit	15	0
Name	TSC	

Overflow of the counter can be detected by the time stamp counter overflow interrupt flag of the global interrupt flag register GIF and the status flag TSO in GSR. Both flags can be cleared by writing a "1" to the corresponding bit location in GIF.

There is a 4-bit prescaler for the time stamp counter. After power-up the time stamp counter is driven directly from the bit clock (TSP = 0). The period T_{TSC} for the time stamp counter will be calculated with the following formula:

$$T_{TSC} = T_{BIT} * (TSP + 1)$$

Time Stamp Counter Prescaler Register

Bit	15 4	3 2 1 0
Name	—	TSP

To be sure, that the value of the counter will not change during the write cycle to the mailbox RAM, there is a hold register implemented. The value of the counter will be copied to this register if a message has been received or transmitted successfully. The reception is successful for the receiver, if there is no error until the last but one bit of End-of-frame. The transmission is successful for the transmitter, if there is no error until the last bit of End-of-frame. (Refer to the CAN specification 2.0B)

The following figure shows the structure of the time stamp counter:

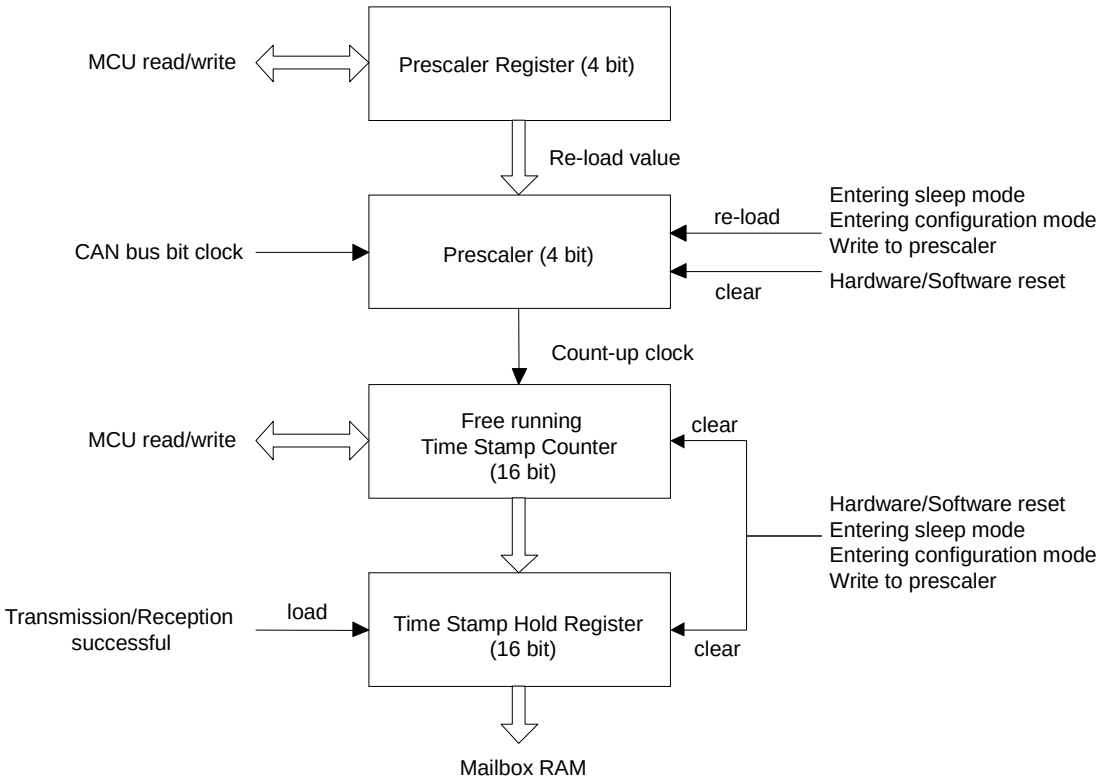


Figure 7.2.5 Time Stamp Counter

The free running time stamp counter and the time stamp hold register will be cleared in the following cases:

- After reset (power-up reset or software reset)
- When the module enters configuration mode
- When the module enters sleep mode
- When a write access to the time stamp prescale register is performed

7.2.9 Status Registers

Global Status Register (GSR)

Bit	15	14	13	12	11	10	9	8
Name	MsgInSlot				RM	TM	—	SUA

Bit	7	6	5	4	3	2	1	0
Name	CCE	SMA	—	—	TSO	BO	EP	EW

Bit	Name	Function	Reset Value	Mode
15:12	Msg InSlot	Message In Slot 1111 = No transmit message in slot 0000 = Message 0 is in the transmission slot ... 1110 = Message 14 is in the transmission slot	1111	R
11	RM	Receive Mode 1 = TXCAN is receiving a message. That means TXCAN is not the transmitter of the message and the bus is not idle. 0 = The CAN module is not receiving a message	0	R
10	TM	Transmit Mode 1 = TXCAN is transmitting a message. The module stays transmitter until the bus is idle or it loses arbitration. 0 = The CAN module is not transmitting a message	0	R
9	—	Wired to zero	0	R
8	SUA	Suspend Mode Acknowledge 1 = TXCAN is in suspend mode 0 = TXCAN is not in suspend mode	0	R
7	CCE	Change Configuration Enable 1 = The MCU is allowed to do write accesses to the configuration registers. 0 = Write accesses to the configuration registers are denied.	1	R
6	SMA	Sleep Mode Acknowledge 1 = TXCAN has entered the sleep mode. 0 = Normal operation	0	R
5	—	Wired to zero	0	R
4	—	Wired to zero	0	R
3	TSO	Time Stamp Overflow Flag 1 = There was at least one overflow of the time stamp counter since this bit has been cleared. To clear this bit, clear the TSOIF bit in the GIF register. 0 = There was no overflow of the time stamp counter	0	R

Bit	Name	Function	Reset Value	Mode
2	BO	Bus Off status 1 = There is an abnormal rate of occurrences of errors on the CAN bus. This condition occurs when the transmit error counter TEC has reached the limit of 256. During “bus off”, no messages can be received or transmitted. The CAN module will go to “bus on” automatically after the “bus off recovery sequence”. After entering “bus off”, the error counters are undefined. 0 = Normal operation	0	R
1	EP	Error Passive status 1 = The CAN module is in the error passive mode. 0 = The CAN module is in the error active mode.	0	R
0	EW	Warning status 1 = At least one of the error counters has reached the warning level of 97. 0 = Both values of the error counters are less than 97.	0	R

CAN Error Counter Register (CEC)

Bit	15 8	7 0
Name	TEC	REC

Bit	Name	Function	Reset Value	Mode
15:8	TEC	Transmit error counter	0	R
7:0	REC	Receive error counter	0	R

The CAN module contains two error counters: receive error counter (REC) and transmit error counter (TEC). The values of both counters can be read via the MCU interface. These counters are incremented or decremented according to the CAN specification version 2.0B. A write access to the error counters is only possible in the test error mode (TSTERR bit in MCR is set). In this mode both the TEC and REC counters take over the value written to the lower byte of the register.

The receive error counter is not increased after exceeding the error passive limit (128). After the correct reception of a message, the counter is set to a value between 119 and 127 (see CAN specification). After reaching the “bus off” status, the error counters are undefined.

If the status “bus off” is reached, the receive error counter is incremented after 11 consecutive recessive bits on the bus. These 11 bits correspond to the gap between two telegrams on the bus. If the counter reaches the count 128, the module changes automatically to the status error active. All internal flags are reset and the error counters are deleted. The configuration registers keep the programmed values. The values of the error counters are undefined during “bus off” status.

When TXCAN enters configuration mode (see paragraph “7.4.1 Configuration Mode”) the error counters will be cleared.

7.3 TXCAN Interrupt Logic

The TXCAN has the following interrupt sources:

- Transmit interrupt: a message has been transmitted successfully
- Receive interrupt: a message has been received successfully
- Warning level interrupt: at least one of the two error counters is greater than or equal to 97
- Error passive interrupt: TXCAN enters the error passive mode
- Bus off interrupt: TXCAN enters the bus off mode
- Time Stamp Overflow Interrupt
- Transmission abort interrupt
- Receive message lost interrupt
- Wake-up interrupt: after wake-up from sleep mode this interrupt will be generated
- Remote frame pending interrupt

These interrupt sources are divided in three groups: transmit interrupts, receive interrupts and global interrupts. There is one interrupt output line for each group. CANRX is dedicated for receive interrupts, CANTX is dedicated for transmit interrupts and CANEXC for the global interrupts.

Global Interrupt Flag Register (GIF)

The interrupt flag bits will be set if the corresponding interrupt condition has occurred. If the corresponding interrupt mask bit is set in the GIM register, the interrupt line IRQ2 will go active high. As long as an interrupt flag in the GIF register is set and the corresponding mask bit is also set, the interrupt line IRQ2 will stay active high (“1”).

Bit	15 8	7	6	5	4	3	2	1	0
Name	—	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF

Bit	Name	Function	Reset Value	Mode
15:8	—	Wired to zero	0	R
7	RFPF	Remote Frame Pending Flag 1 = A remote frame has been received (in a receive-mailbox). This bit will not be set if the identifier of the remote frame matches to a transmit-mailbox with RFH set. 0 = No remote frame has been received.	0	R/C
6	WUIF	Wake-Up Interrupt Flag 1 = The module has left the sleep mode. 0 = The module is still in sleep mode or normal operation.	0	R/C
5	RMLIF	Receive Message Lost Interrupt Flag 1 = At least for one of the mailboxes, configured as receive, an overload condition has been occurred. 0 = No message has been lost.	0	R/C
4	TRMABF	Transmission Abort Flag 1 = Transmission aborted interrupt-flag. At least one of the bits in the AA register is set. 0 = No transmission has been aborted.	0	R/C

Bit	Name	Function	Reset Value	Mode
3	TSOIF	Time Stamp Counter Overflow Interrupt Flag 1 = There was at least one overflow of the time stamp counter since this bit has been cleared. 0 = There was no overflow of the time stamp counter since this bit has been cleared.	0	R/C
2	BOIF	Bus Off Interrupt Flag 1 = The CAN has entered the bus off mode. 0 = The CAN module is still in bus on mode.	0	R/C
1	EPIF	Error Passive Interrupt Flag 1 = The CAN module has entered the error passive mode. 0 = The CAN module is still in error active mode.	0	R/C
0	WLIF	Warning Level Interrupt Flag 1 = At least one of the error counters has reached the warning level. 0 = None of the error counters has reached the warning level.	0	R/C

Note: All interrupt flags in GIF are independent of the interrupt mask bits. The interrupt flags in GIF can be cleared by writing a “1” to the corresponding bit position. Writing a “0” has no effect.

Global Interrupt Mask Register (GIM)

Bit	15			8	7	6	5	4	3	2	1	0
Name	—				RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF

The attachment of bits in GIM to the interrupt conditions is equal to that in GIF. Each interrupt flag bit in GIF is masked by the corresponding mask bit in GIM. After power-up, all bits are cleared.

7.3.1 Mailbox Interrupts

There are two separate interrupt output lines for the mailboxes. One interrupt output for mailboxes, which are configured as transmit and one for mailboxes which are configured as receive.

There are two interrupt flag registers and one interrupt mask register. One interrupt flag register is for receive mailboxes and one for transmit mailboxes. The interrupt mask register is used for transmit and receive mailboxes.

Mailbox Interrupt Mask Register (MBIM)

The settings in MBIM determine, for which mailbox the interrupt generation is enabled or disabled. If a bit in MBIM is “0”, the interrupt generation for the corresponding mailbox is disabled and if it is “1”, the interrupt generation is enabled. Reset value of MBIM is 0.

Bit	15	0
Name	MBIM	

Mailbox Interrupt Flag Registers (MBTIF/MBRIF)

Bit	15	14	0
Name	—	MBTIF	

Bit	15	0
Name	MBRIF	

There are two interrupt flag registers. One for receive mailboxes and one for transmit mailboxes. If a mailbox is configured as receive, the corresponding bits in the transmit interrupt flag register MBTIF will always be read as “0”. In MBTIF, bit 15 is not implemented, because mailbox 15 is the receive-only mailbox. Bit 15 of MBTIF will always be read as “0”. If a mailbox is configured as transmit, the corresponding bits in MBRIF will always be read as “0”.

If a message has been received for mailbox “n” and the mask bit is set to “1” the corresponding interrupt flag “n” of MBRIF will be set to “1” and the interrupt line IRQ0 goes active high (“1”).

If a message has been transmitted from mailbox “n” and the mask bit is set to “1” the corresponding interrupt flag “n” of MBTIF will be set to “1” and the interrupt line IRQ1 goes active high (“1”).

If the mask bit in MBIM is set to “0”, the interrupt flag in MBRIF or MBTIF will not be set and no interrupt will be generated. The information about a successful transmission or reception could be read from the TA or RMP register respectively.

The interrupt output lines IRQ0 and IRQ1 will stay at “1” as long as one of the interrupt flags in MBRIF or MBTIF are “1” respectively and the corresponding bits in MBIM are set to “1”.

The interrupt flags in MBTIF will be cleared by writing a “1” from the MCU to MBTIF and the interrupt flags in MBRIF will be cleared by writing a “1” to MBRIF. Writing a “0” has no effect. The corresponding status flags in TA or RMP have to be cleared separately.

After power-up, all interrupt flags are cleared.

7.4 TXCAN Operation Modes

7.4.1 Configuration Mode

The TXCAN has to be initialized before activation. The bit timing parameters can only be modified when the module is in configuration mode. After reset, the configuration mode is active and the CCR bit of MCR and the CCE bit of GSR are set to “1”. The TXCAN could be set to normal operation mode by writing a “0” to CCR. After leaving configuration mode, the CCE bit will be set to “0” and the power-up sequence will start. The power-up sequence consists of detecting eleven consecutive recessive bits on the CAN bus line. After the power-up sequence, TXCAN is bus on and ready for operation.

To enter configuration mode from normal operation mode the change configuration request bit (CCR) has to be set to “1”. After the TXCAN has entered configuration mode, the change configuration enable bit (CCE) will be set to “1”. See also the following flowchart.

When the TXCAN enters configuration mode the error counters, the time stamp counter and the time stamp hold register will be cleared.

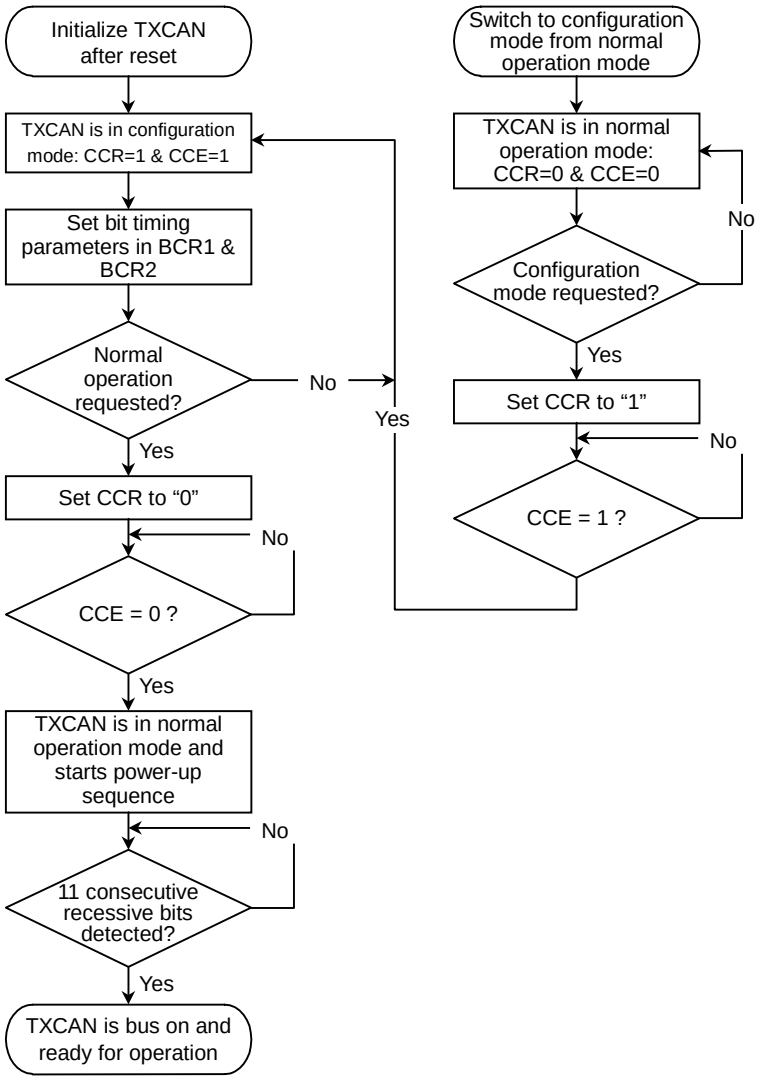


Figure 7.4.1 Configuration Flow Chart for TXCAN

7.4.2 Sleep Mode

The sleep mode will be requested by writing a “1” to SMR (MCR register). When the module enters the sleep mode, the status bit “sleep mode acknowledge” (SMA, GSR register) will be set.

During sleep mode, the internal clock of TXCAN is switched off. Only the wake up logic will be active. The read value of the GSR will be f040h, this means, there is no message in slot and the sleep mode is active (SMA is set). Read accesses to all other registers will deliver the value 0000h. Write accesses to all registers but the MCR will be denied.

The module leaves the sleep mode if a write access to MCR has been detected or there is any bus activity detected on the CAN bus line (if the wake-up on bus activity is enabled).

The automatic “wake up on bus activity” can be enabled/disabled with the configuration bit WUBA in MCR.

If there is a write access to MCR or any activity on the CAN bus line (with WUBA = 1), the module begins its power-up sequence. The module waits until detecting 11 consecutive recessive bits on the RX input line, afterwards it goes to bus active. The first message that initiates the bus activity cannot be received.

In sleep mode, the CAN error counters and all “transmission requests” (TRS) and “transmission reset requests” (TRR) will be cleared. After leaving the sleep mode, SMR and SMA will be cleared.

If the sleep mode is requested while TXCAN is transmitting a message, the module will not switch to the sleep mode immediately. It will continue until a successful transmission or after losing the arbitration, until

- a successful transmission or
- after loosing the arbitration a successful reception

occurs.

7.4.3 Suspend Mode

The suspend mode will be requested by writing a “1” to SUR (MCR register). When the module enters the suspend mode the status bit SUA (GSR register) will be set to “1”. If the CAN bus line is not idle, the current transmission/reception of the message will be finished before the suspend mode will be activated.

In suspend mode the TXCAN is not active on the CAN bus line. That means error flags and acknowledge flags will not be sent. The error counters and the error passive flag will not be cleared in the suspend mode.

If the suspend mode is requested during the bus off recovery sequence, the module stops after the bus off recovery sequence was finished. The module remains inactive until suspend mode request SUR is deactivated. The suspend mode acknowledge flag is not activated, although the SUR bit is “1” and the module is inactive. To restart the module, the SUR bit has to be programmed to “0”. After leaving the bus off state or the inactive state, the module will restart its power-up sequence.

TXCAN leaves the suspend mode by writing a “0” to SUR.

7.4.4 Test Loop Back Mode

In this mode TXCAN can receive its own transmitted message and will generate its own acknowledge bit. No other CAN node is necessary for the operation.

When the INTLB bit of the MCR register is “0”, the internal loop back is disabled. The supposition that TXCAN receives its own messages is that the RX and TX lines must be connected to a CAN bus transceiver or directly together.

When the INTLB bit of the MCR register is set to “1”, the internal loop back is enabled. In this case, there is no need to connect the RX and TX lines together or to a CAN bus transceiver to make the TXCAN able to receive its own messages.

The “test loop back mode” shall only be enabled or disabled when TXCAN is in suspend mode. The following figure shows the set-up procedure.

In “test loop back mode” TXCAN can transmit a message from one mailbox and receive it in another mailbox. The set-up for the mailboxes is the same as in normal operation mode.

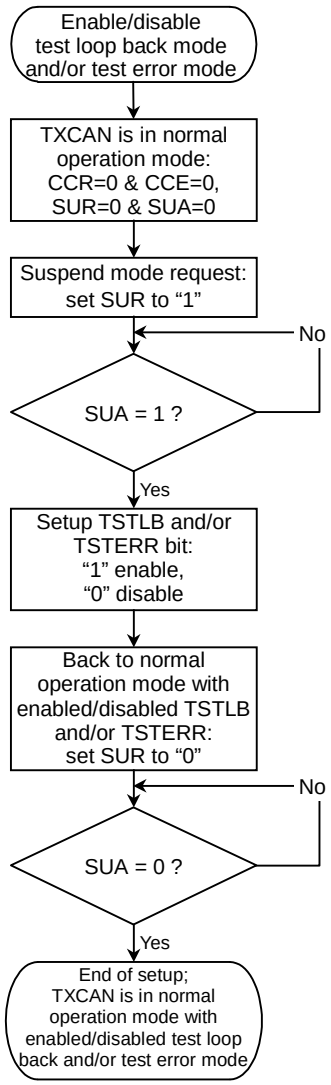


Figure 7.4.2 Internal Test Flow Chart

7.4.5 Test Error Mode

The error counters can only be written when TXCAN is in test error mode.

The "test error mode" shall only be enabled or disabled when TXCAN is in suspend mode. Figure 7.4.2 shows the set-up procedure.

When TXCAN is in "test error mode" both error counters will be written at the same time with the same value. The maximum value that can be written into the error counters is 255. Thus, the error counter value of 256 which forces TXCAN into bus off mode can not be written into the error counters.

7.4.6 Special Modes for Dual Channel CAN

- Modifications due to adding a second CAN Channel
- Bits are required from the CCR register.
- SingleChannelEmu and CanTstInternal. Setting the CanTstInternal to "1" switches to an internal testmode, which is a internal connection independent from any external pins or transceivers.

Internal Test Mode

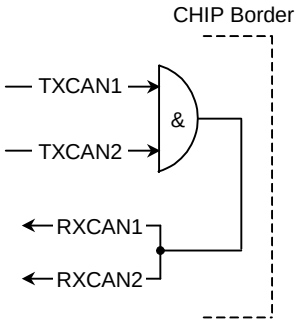


Figure 7.4.3 Internal Test Mode

- The Signal "SingleChannelEmu" connects 2 channels to be able to work on only 1 transceiver.

Single Transceiver Mode

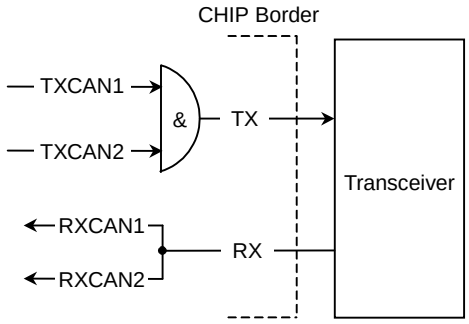


Figure 7.4.4 Single Transceiver Mode

The following table shows the bit function of CANM in the Chip Configuration Register (CCR):

CANM	CAN Mode
00	Normal operation
X1	Internal Test mode
10	1 Transceiver - Mode

7.5 Handling of Message-Objects

In the following sections, there are suggestions how to handle message objects.

7.5.1 Receiving Messages

The following flowchart shows the handling of receive objects using receive interrupt IRQ0.

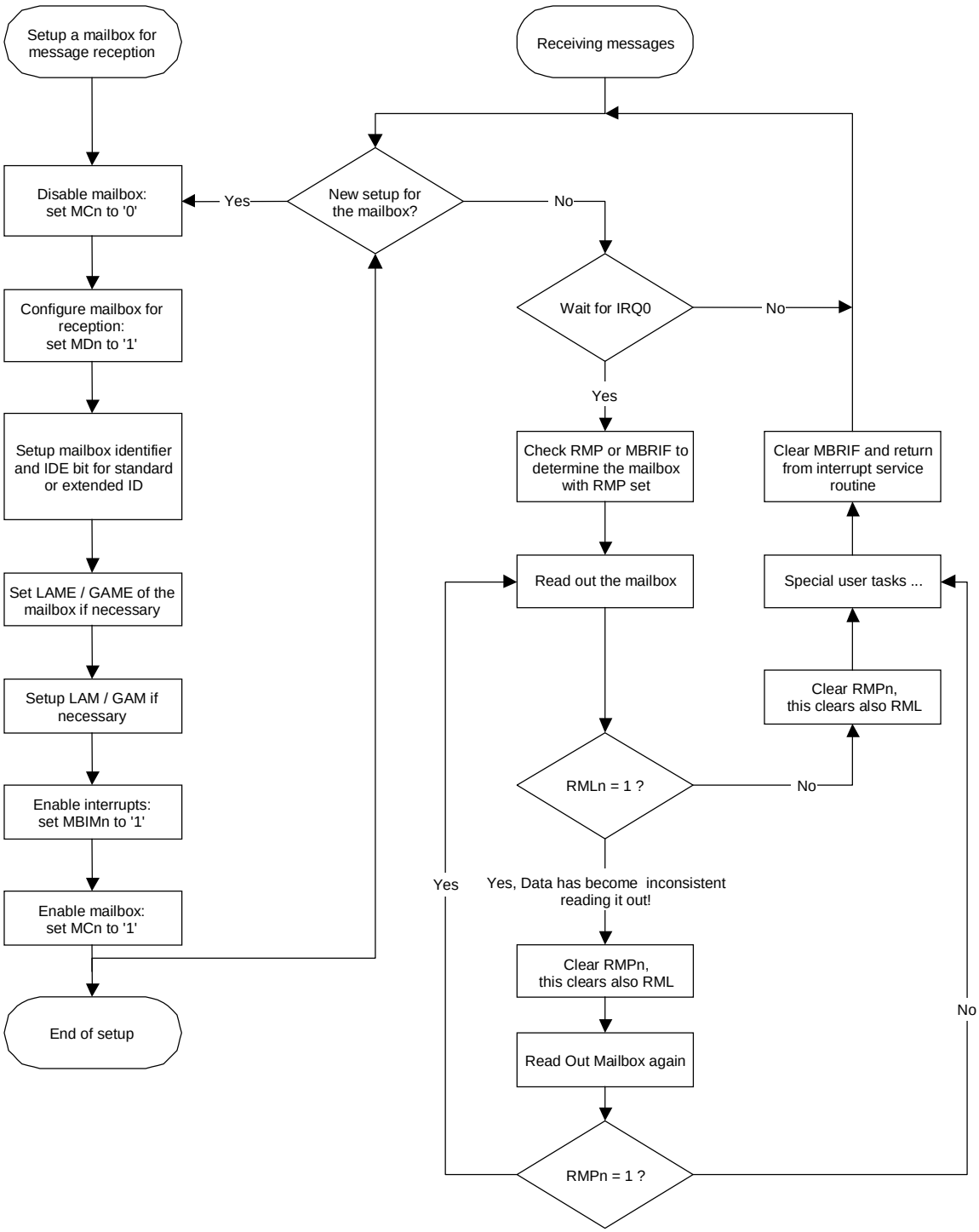


Figure 7.5.1 Receiving Objects Using IRQ0

It is also possible to use polling. In this case, the “waiting for IRQ0” in above flowchart must be replaced by polling RMP. Enabling interrupts and clearing MBRIF must be removed from the flow.

7.5.2 Transmitting Messages

The following flowchart shows the handling of transmit objects by using the transmit interrupt IRQ1.

It is also possible to use polling. In this case, the “waiting for IRQ1” in the flowchart must be replaced by polling TA. Enabling interrupts and clearing MBTIF must be removed from the flow.

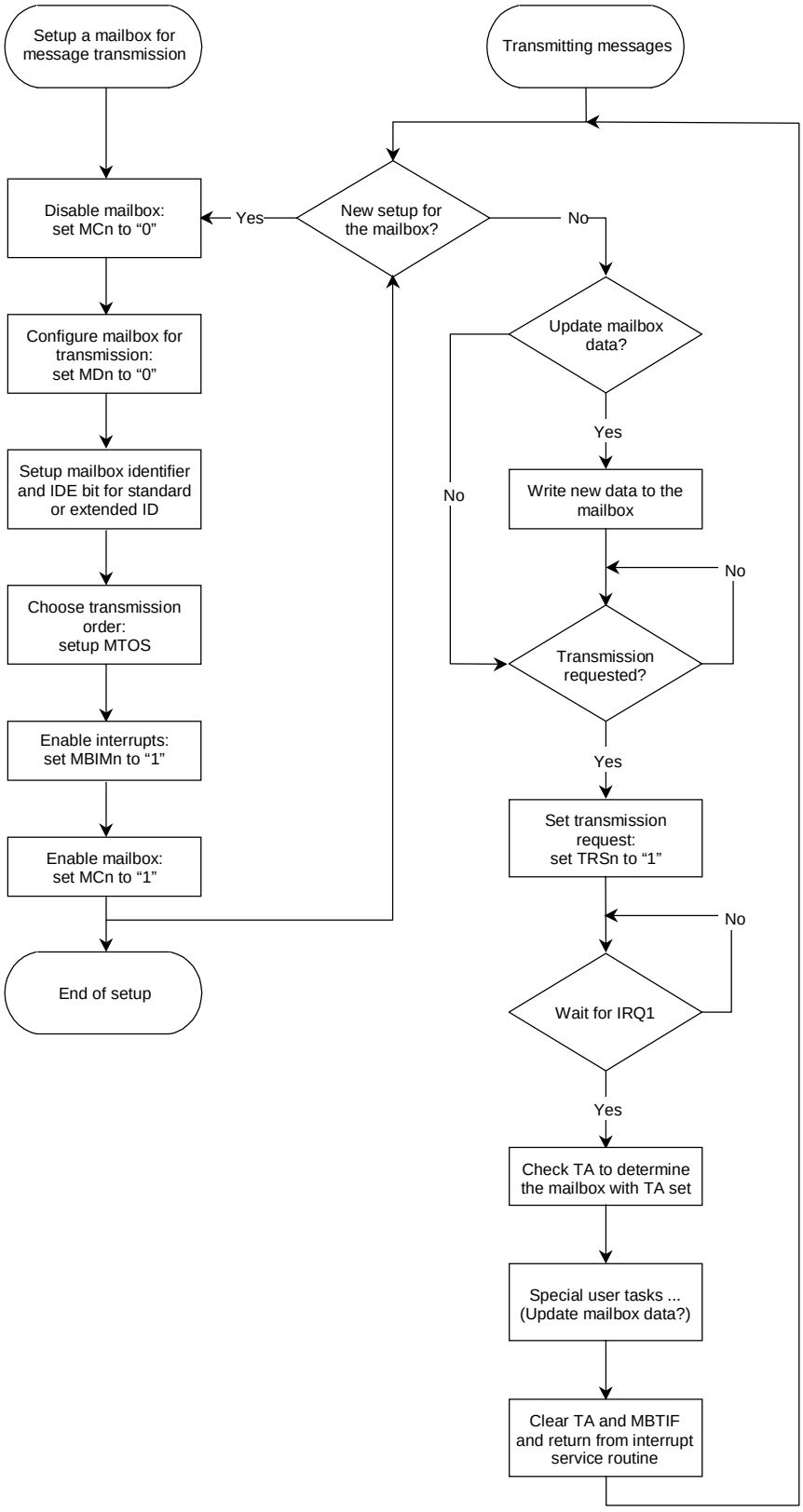


Figure 7.5.2 Transmitting Objects Using IRQ1

7.5.3 Remote Frame Handling

The following flowchart shows the handling of remote frames by using the automatic reply feature. This feature is available when the RFH bit of a mailbox, which is configured for transmission, is set. To avoid data inconsistency problems when updating the mailbox data the CDR register is used.

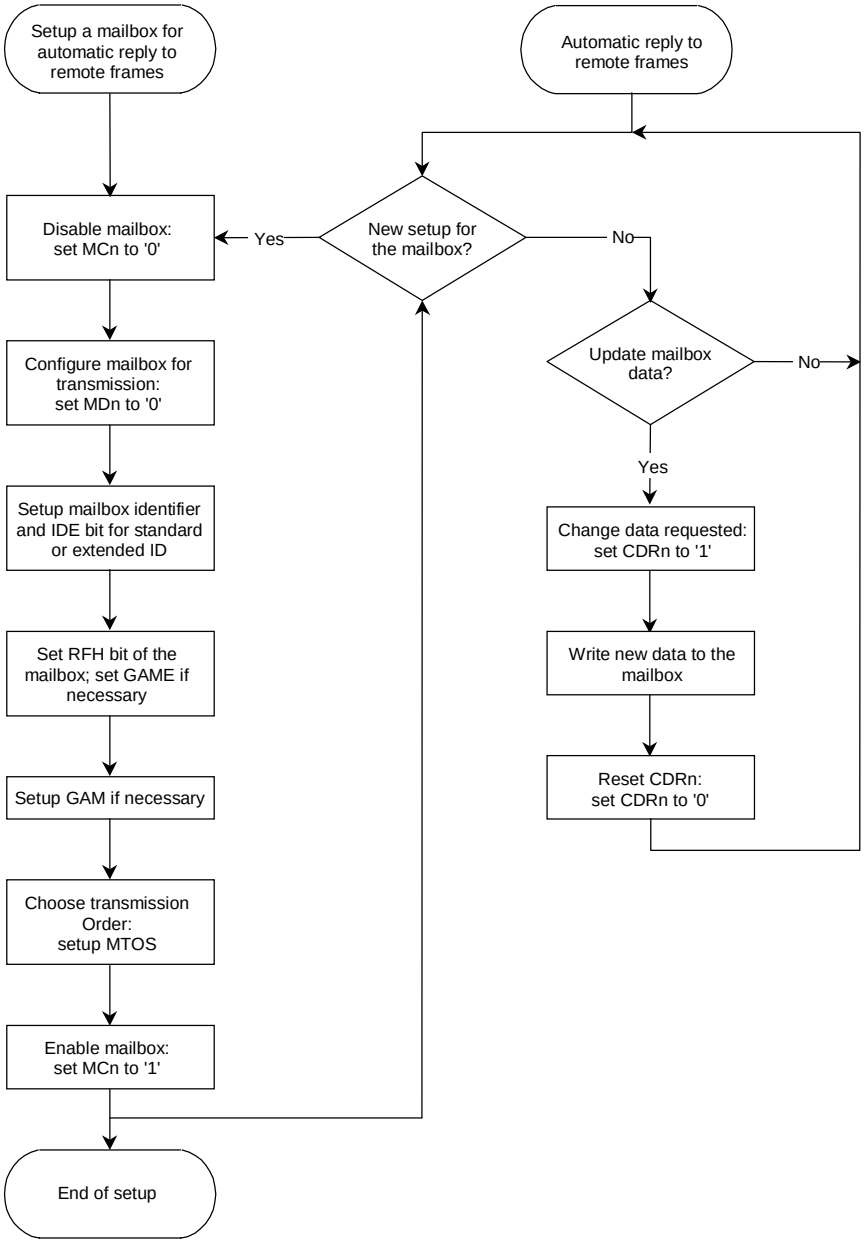


Figure 7.5.3 Automatic Reply Generation for Remote Frames

8. Parallel Interface (PORT)

The PORT-module is a general-purpose parallel interface. The PORT-Module contains the following features:

- 30 pins
- each pin can be configured independently as input or output
- each pin can generate an interrupt on rising or falling edge of input-signal

The PORT module shares its pins with modules GDC, TXSEI and UART.

Bit assignment of all PORT registers:

Bit	31	30	29	28	27	26	25	24
Pin	wired to zero		PIO29	PIO28	PIO27	PIO26	PIO25	PIO24
Access	RO		R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17	16
Pin	PIO23	PIO22	PIO21	PIO20	PIO19	PIO18	PIO17	PIO16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8
Pin	PIO15	PIO14	PIO13	PIO12	PIO11	PIO10	PIO9	PIO8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Pin	PIO7	PIO6	PIO5	PIO4	PIO3	PIO2	PIO1	PIO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Function:

Each bit of the following registers is assigned to the corresponding pin. The table below describes the function of each register for one pin.

Register	Physical Address (hex)	Function of One Bit	Reset Value
PAMUX	1C03 0014	Determines which module uses the pin. Each bit in this register represents the setting of one PIO. The upper both bits (31 and 30) are not relevant, because the TMPR3916F provides 30 PIOs. 0 = PORT module uses the pin 1 = GDC, TXSEI or UART (depending on the pin) Further descriptions see section “Pin Assignment” on next side.	0
PA	1C03 0000	This register contains data read from PORT pins or written to PORT pins. When a pin is used as input, then the corresponding bit in this register is read only.	0
PACR	1C03 0004	Direction of pin 0 = Input 1 = Output The contents of this bit shows no effect, when PAMUX = 1.	0
PAMSK	1C03 0010	Interrupt enable 0 = Disable interrupt 1 = Enable interrupt Whether the interrupt caused on falling or rising edge of input signal, depends on the contents of PALMX register. When the pin is used by another resource or the pin is used as output, the interrupt is inhibited.	0

Register	Physical Address (hex)	Function of One Bit	Reset Value
PALMX	1C03 000C	Controls edge detection for interrupt generation 0 = Cause an interrupt on falling edge 1 = Cause an interrupt on rising edge The contents of this register show no effect, if PAMSK is set to 0.	0
PAL	1C03 0008	Interrupt flag 0 = No interrupt has occurred on pin 1 = Interrupt has occurred on pin Bits in this register are set, when a rising edge or a falling edge (depending on PALMX setting) on the corresponding PIO pin is detected and the PIO was enabled for interrupts (PAMSK). Flags are only set if the PIO is configured to be an input (PACR) and is used by the PORT module (PAMUX). Each flag can be reset to 0 by writing a 0 to it. Before you can reset a PORT-interrupt in the interrupt controller, you must reset the interrupt flag in PAL register. Writing 1 to this register has no effect.	0

Pin Assignment

The PAMUX register in the PORT module and the SEIMUX bits in the Chip Configuration Register (CCR) determine the use of the PIO pins. The following table shows the pin use and the corresponding bit settings:

Register Settings	PIO0 .. PIO15	PIO16 .. PIO29
PAMUX = 0	PORT	PORT
PAMUX = 1 and SEIMUX = 0	GDC	UART
PAMUX = 1 and SEIMUX = 1	GDC	TXSEI

Example for register configuration:

Task		Solution
pins 0 to 15 are used by GDC, pins 16 to 29 are used by PORT	=>	PAMUX = 0x0000FFFF
pins 16 to 20 are used as outputs, pins 21 to 29 are used as inputs	=>	PACR = 0x000F0000
following pins should cause an interrupt - a signal change from low to high on pin 24 - a signal change from high to low on pin 25	=>	PAMSK = 0x03000000 PALMX = 0x01000000

9. Synchronous Serial I/O (TXSEI)

The Toshiba TX Serial Expansion Interface is a synchronous communication unit and compatible to peripheral devices, which can be connected to an SPI/SEI type interface.

TXSEI's flexible clock control logic allows the selection of clock polarity and phase for the transfer protocol. When TXSEI is configured as a master, a large number of different bit rates with up to 15 MHz clock rate in the master mode can be selected. In slave mode transmissions up to 7.25 MHz are possible (assuming the TMPR3916 is operating with 60 MHz)

The built-in error detection logic allows the detection of various error situations, which can occur during SEI transfers.

TXSEI also offers DMA support for automated data transfers to its shift registers. By the use of DMA a larger number of transfers can be scheduled at once. In particular, the usage of DMA allows a more cost-effective implementation than usual large queue or buffer structures. TXSEI is able to perform seamless transfers of consecutive frames. Alternatively, the minimum delay between two consecutive transfers is programmable for master mode.

Feature Overview:

- Phase and Polarity Selection
- Transfer sizes of 5 to 16 bits
- DMA operation: Full-Duplex 2 channels, Half-Duplex 1 channel
- Built-in Error Detection Logic
- 4 frame transmit, 4 frame receive buffers
- Compatible with SPI type interfaces
- Master and Slave operation
- 15 Mbps data-rate when operated with 60 MHz clock rate.
- Inter Frame Space Delay Feature
- seamless transfer of large values without delay between consecutive frames
- integrated MSB / LSB first reordering
- Stop and Flush Buffer functionality for fast event response
- programmable buffer-fill-level dependent receive / transmit interrupts

9.1 TXSEI Structure

The following figure roughly shows the internal structure of the TXSEI and the connectivity to the outside of the chip via the PORT-Multiplexer:

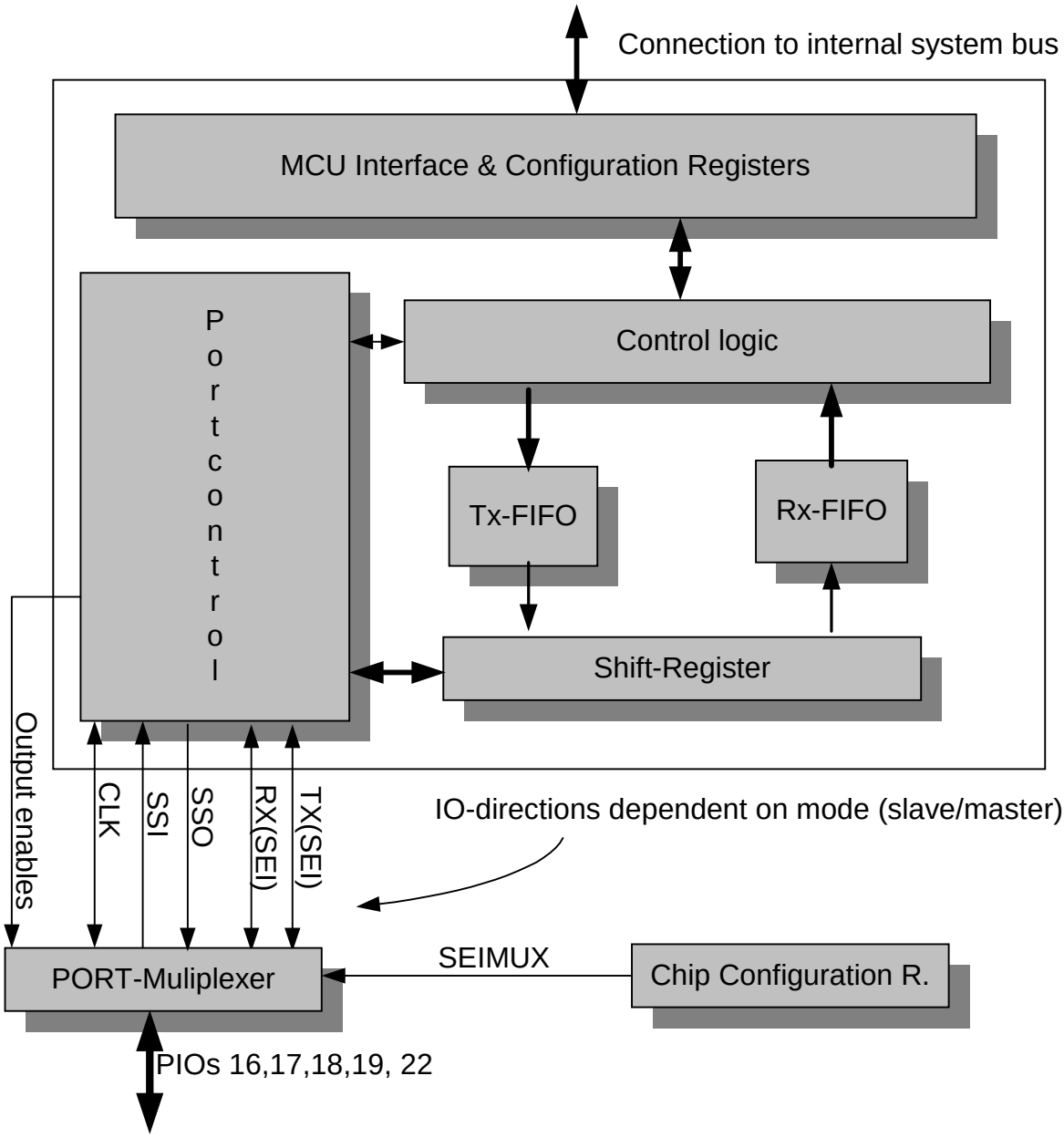


Figure 9.1.1 Internal Structure of TXSEI

9.2 Registers

The following table shows a map of the TXSEI I/O-Space:

Register (short name)	Physical Address (hex)	*Name*	Function
SEMCR	1C00 8000	*Master Control Register*	Mode settings
SECR0	1C00 8004	*SEI Control Register 0*	General settings
SECR1	1C00 8008	*SEI Control Register 1*	Definition of bit-rate and transfer-size
SEFS	1C00 800C	*SEI Inter Frame Space Register*	Definition of space between frames
SESS	1C00 8010	*SEI Slave Select Space Register*	Slave select timer settings
SESR	1C00 8014	*SEI Status Register*	Status information
SEDR	1C00 8018	*SEI Data Register*	Transmit and receive data
SERS	1C00 801C	*SEI Read Start Register*	Alternative register to read received data

TXSEI's registers can be accessed using Byte, Half-Word and Word instructions. Bits [31:16] are unused in all registers. These bits are wired to zero and read-only.

SEI Master Control Register (SEMCR)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	—							

Bit	7	6	5	4	3	2	1	0
Name	OPMODE		—			LOOP	SESTP	BCLR

Bit	Name	Function	Reset Value	R/W
31:8	—	Unused	0x000000	R
7:6	OPMODE	Operation Mode 00 = Don't care. Writing this value to the OPMODE bits doesn't change anything 01 = Configuration Mode: Use this mode to change the settings of the bits MSTR, SBOS, SPOL and SPHA in SECR0 and also the SECR1 register. 10 = Active Mode: normal operation mode 11 = Reserved. Do not use this setting In Configuration Mode the SESTP and LOOP bit and also the receive and transmit FIFO will be cleared. The master and slave control modules will be kept in reset. Running transfers are immediately aborted, even within the current frame.	01	R/W
5:3	—	Wired to zero	000	R
2	LOOP	Loop Enable: If TXSEI is configured as a Master, this bit can be used to switch a loop-back from the TX to the RX pin for diagnostic purpose. It could be set only when the TXSEI is in active mode and configured as a master. Setting the TXSEI in configuration mode will clear this bit. 0 = Loop disabled, normal operation 1 = Loop enabled	0	R/W
1	SESTP	SEI Stop This bit is used only during master mode. If this flag is asserted, the module will stop the transfer after the current frame has been completed. This bit could be set only when the TXSEI is in active mode and configured as a master. Setting the TXSEI in configuration mode will clear this bit. 0 = Normal operation 1 = Module will stop after completion of the current transfer	0	R/W
0	BCLR	SEI Buffer Clear This flag is used to clear the receive and transmit FIFO and can only be used in master mode. The internal buffers can only be cleared, if the module is already in stop mode. In this case, the FIFO logic can be reset by writing a "1" value to this bit. The module can be taken out of the stop mode in the same access. A stop of TXSEI and clearance of the buffers might become necessary to guarantee a fast response to events. It is recommended to wait until the TXSEI module is idle (SIDLE=1) before activating BCLR. This bit will always be read as "0".	0	R/W

SEI Control Register 0 (SECR0)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	TXIFL		RXIFL		SILIE	SOEIE	SUEIE	STFIE

Bit	7	6	5	4	3	2	1	0
Name	—		SSIVAL	IFSPSE	MSTR	SBOS	SPHA	SPOL

Bit	Name	Function	Reset Value	R/W
31:16	—	Unused	0x0000	R
15:14	TXIFL	Transmit Interrupt Fill Level (SEITX): 00 = Interrupt, if one or more Tx values can be stored 01 = Interrupt, if two or more Tx values can be stored 10 = Interrupt, if three or more Tx values can be stored 11 = Interrupt, if four or more Tx values can be stored The first setting (TXIFL = 00) should be used, if the DMA Controller is used for transferring values for TXSEI transmissions.	00	R/W
13:12	RXIFL	Receive Interrupt Fill Level (SEIRX): 00 = Interrupt, if one or more Rx values are stored 01 = Interrupt, if two or more Rx values are stored 10 = Interrupt, if three or more Rx values are stored 11 = Interrupt, if four or more Rx values are stored The first setting (RXIFL = 00) should be used, if the DMA Controller is used for transferring values from TXSEI receptions.	00	R/W
11	SILIE	SEI IDLE Interrupt Enable: 0 = Disable SIDLE as an interrupt source for SEIEXC 1 = Enable SIDLE as an interrupt source for SEIEXC	0	R/W
10	SOEIE	SEI Overflow Error Interrupt Enable: 0 = Disable SEOE as an interrupt source for SEIEXC 1 = Enables SEOE as an interrupt source for SEIEXC	0	R/W
9	SUEIE	SEI Underflow Error Interrupt Enable: 0 = Disable SEUE as an interrupt source for SEIEXC 1 = Enables SEUE as an interrupt source for SEIEXC	0	R/W
8	STFIE	SEI Transfer Format Error Interrupt Enable: 0 = Disable SETF as an interrupt source for SEIEXC 1 = Enable SETF as an interrupt source for SEIEXC	0	R/W
7:6	—	Wired to zero	0	R
5	SSIVAL	SSI valid Determines if the Slave Select input signal is valid in master mode or not. If valid, the SSI signal will be observed in master mode to generate a transfer format error. 0 = SSI not valid in master mode 1 = SSI valid in master mode	0	R/W
4	IFSPSE	Inter Frame Space Prescaler Enable (valid only in master mode). 0 = IFS prescaler disabled 1 = IFS prescaler enabled	0	R/W
3	MSTR	Master / Slave Mode Select 0 = TXSEI is configured as slave 1 = TXSEI is configured as master	0	R/W

Bit	Name	Function	Reset Value	R/W
2	SBOS	SEI Bit Order Select 0 = LSB first operation, the least significant bit is shifted first 1 = MSB first operation, the most significant bit is shifted first	0	R/W
1	SPHA	SEI Phase This flag selects one of two fundamentally different transfer formats. 0 = Sample on 1st edge, Shift on 2nd edge 1 = Shift on 1st edge, Sample on 2nd edge.	0	R/W
0	SPOL	SEI Polarity 0 = Active High Clocks selected; SCLK idles low 1 = Active Low Clocks selected; SCLK idles high	0	R/W

Note: Bits 0 to 5 of this register can only be changed in configuration mode.

SEI Control Register 1 (SECR1)

The number of bits per frame is configured using this register. This register could only be written, when the module is in configuration mode.

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	SER							

Bit	7	6	5	4	3	2	1	0
Name	—			SSZ				

Bit	Name	Function	Reset Value	R/W
31:16	—	Unused	0x0000	R
15:8	SER	In master-mode, this setting controls the bit-rate for transmission. The internal clock rate generator is implemented as a down counter. The SER setting specifies the reload value for this counter.	0x01	R/W
7:5	—	wired to zero	0	R
4:0	SSZ	Transfer Size 0x05 = 5 bits 0x06 = 6 bits ... ... 0x10 = 16 bits others = invalid setting ***Note:*** If SSZ has an invalid setting, the TXSEI will not work properly.	0	R/W

This register can only be written, if TXSEI is in configuration mode.

The clock-rate on the SEI bus can be calculated using the following formula:

$$f_{SEI} = \frac{f_{system}}{2 \cdot (SER + 1)}$$

As an example, some common settings for f_{SYS} = 60 MHz are shown in the table below:

SER Setting	SEI Clock Rate (f_{SEI})	Sustained Peak Data Rate half-duplex	Sustained Peak Data Rate full-duplex
0x00	Invalid setting	Invalid setting	Invalid setting
0x01	15 MHz	15 Mbps	30 Mbps
0x02	10 MHz	10 Mbps	20 Mbps
0x03	7.5 MHz	7.5 Mbps	15 Mbps
0x04	6 MHz	6 Mbps	12 Mbps
0x05	5 MHz	5 Mbps	10 Mbps
0x09	3 MHz	3 Mbps	6 Mbps
0x13	1.5 MHz	1.5 Mbps	3 Mbps
0xFF	117.1875 kHz	117.1875 kbps	234.375 kbps

In slave mode, the setting is ignored and the clock is derived from the clock on the SEI bus.

Note: Due to the internal over-sampling, if the module is operated in slave mode, the input baud-rate must be slightly less than 1/8 of the input system clock to the module. (e.g. 60 MHz system input clock => SPI slave baud rate max. 7.25 Mbps)

SEI Inter Frame Space Register (SEFS)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	—						IFS[9:8]	

Bit	7	6	5	4	3	2	1	0
Name	IFS[7:0]							

Bit	Name	Function	Reset Value	R/W
31:10	—	Unused	0x000000	R
9:0	IFS	Inter frame space: Time between two consecutive transmission frames	0	R/W

This register is used to configure the amount of time, which is inserted between two consecutive frames. The time is guaranteed by an internal 10-bit down counter. The counter can be operated with or without prescaler. The IFSPSE bit of the SECR0 register determines whether the prescaler should be used or not. When operating without prescaler, the counter runs on SEI system clock. When operating with prescaler, the counter runs on 1/32 of SEI system clock.

This counter is implemented as a down counter. It is reloaded each time a transfer is completed. When another transfer is buffered, the new transfer value will be loaded to the shift buffer after the timer has expired and the transmission will start.

The Inter Frame Space Timer can be disabled by setting this register to “0” and two consecutive transfers will be sent using only the minimum amount of time required to load the buffers between consecutive frames (seamless transfer). When the counter reload value in the IFS register is “0” the inter frame space will be one system clock cycle (16.67 ns at 60 MHz system clock).

When the prescaler is not used, the inter frame space can be calculated using the following formula:

$$t_{IFS} = \frac{IFS+1}{f_{SYS}}$$ (range: 16.67 ns up to 17.07 μs at 60 MHz system clock)

When using the prescaler, the inter frame space can be calculated using the following formula:

$$t_{IFS} = \frac{32\times(IFS+1)}{f_{SYS}}$$ (range: 533.33 ns up to 546.13 μs at 60 MHz system clock)

The IFS register can be written in configuration mode and in active mode. Writing to the IFS register always clears the Inter Frame Space counter. Therefore, if the shift buffer contains a message, which is waiting to be transferred, this message will be sent immediately, as soon as the IFS register is being written. This will also be the case, if the old value is rewritten to the register. If this behavior is not intended, it is possible to wait for SIDLE flag becoming “0”, before writing to the register.

The IFSD flag in the SESR register is asserted for the time the transfer is delayed by the IFS mechanism.

SEI Slave Select Space Register (SESS)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	—							

Bit	7	6	5	4	3	2	1	0
Name	SESS							

Bit	Name	Function	Reset Value	R/W
31:8	—	Unused	0x000000	R
7:0	SESS	Slave Select Space: Time between assertion of slave select signal and transmission start	0	R/W

The contents of this register is the reload value of the Slave Select Timer. Write accesses to this register are possible in configuration mode and in active mode. Writing to this register clears the slave select counter.

This register is used to configure the amount of time, which is inserted between activating the slave select output signal in master mode and starting the transfer and between the transfer end and deactivating the slave select output signal. The time is guaranteed by an internal 8-bit down counter. The counter runs on SEI

system clock.

When writing to the SESS register while the shift buffer contains a message, which is waiting to be transferred, this message will be sent immediately, since the counter is cleared to “0”.

The slave select space can be calculated using the following formula:

Pre-transfer time:

$$t_{SSC_PRE} = \frac{2+SESS}{f_{SYS}}$$ (range: 33.33 ns up to 4.28 μs at 60 MHz system clock)

Post-transfer time:

$$t_{SSC_POST} = \frac{3+SESS}{f_{SYS}}$$ (range: 50 ns up to 4.3 μs at 60 MHz system clock)

The Slave Select Space Timer can be disabled by setting this register to “0”. The minimum time between setting the slave select signal and starting the transfer is 2 system clock cycles and the minimum time between the transfer end and deactivating the slave select signal is 3 system clock cycles. This is the case when SESS is set to “0”.

The minimum time between two consecutive transfers is the sum of the minimum values of t_{SCC_PRE}, t_{IFS} and t_{SCC_POST}: this is 6 system clock cycles.

SEI Status Register (SESR)

In the SEI Status Register, the status flags can only be read, while error flags are cleared by writing a “1” value to the respective bit position. Writing a “0” to the Error Flags has no effect.

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	TBSI	RBSI	TBS			RBS		

Bit	7	6	5	4	3	2	1	0
Name	SEOE	SEUE	SETF	—	IFSD	SIDLE	STRDY	SRRDY

Bit	Name	Function	Reset Value	R/W
31:16	—	Unused	0x0000	R
15	TBSI	Transmit Buffer Status Indicator This register indicates a transmit fill level interrupt	1	R
14	RBSI	Receive Buffer Status Indicator This register indicates a receive fill level interrupt	0	R
13:11	TBS	Transmit Buffer Status This register shows the status of the transmit buffer. 000 = Transmit Buffer Empty 001 = 1 transfer stored 010 = 2 transfers stored 011 = 3 transfers stored 100 = 4 transfers stored, Buffer full	000	R
10:8	RBS	Receive Buffer Status This register shows the status of the receive buffer. 000 = Receive Buffer Empty 001 = 1 transfer stored 010 = 2 transfers stored 011 = 3 transfers stored 100 = 4 transfers stored, Buffer full	000	R
7	SEOE	SEI Overflow Error: This flag indicates that a value in the receive buffer has been overwritten, before it could be read. This flag always reads “0” in master mode. In slave mode, it can be cleared by writing a “1” value to it. This flag will be cleared by setting the module in configuration mode.	0	R/C
6	SEUE	SEI Underflow Error: This flag indicates that an external master tried to shift the shift register, while no new output values were specified by writing to the data register. This flag always reads “0” in master mode. In slave mode it is cleared by writing a “1” to it. This flag will be cleared by setting the module in configuration mode.	0	R/C
5	SETF	SEI Transfer Format Error: This flag indicates a violation of the transfer format. See paragraph "Transfer Format Error". It can be cleared by writing a “1” to it. This flag will be cleared by setting the module in configuration mode.	0	R/C
4	—	Unused	0	R

Bit	Name	Function	Reset Value	R/W
3	IFSD	SEI Inter Frame Space Delay Indicator: This Flag is asserted during the time, where one frame has been processed and the next frame is being delayed by the inter-frame-space timer.	0	R
2	SIDLE	SEI Idle Indicator: This flag is asserted, if no transfer is in progress and if the transmit buffer is empty or the stop mode (SESTP=1) is activated in master mode.	1	R
1	STRDY	SEI Transmit Ready: This flag indicates, that the transmit buffer is ready to receive new data. The flag is cleared, if the transmit buffer is full.	1	R
0	SRRDY	SEI Receive Ready: This flag indicates, that there is valid data stored in the receive buffer. This flag is cleared when emptying the receive buffer while reading SEDR or SERS register	0	R

SEI Data Register (SEDR)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	DR[15:8]							

Bit	7							0
Name	DR[7:0]							

Bit	Name	Function	Reset Value	R/W
31:16	—	Unused	0x0000	R
15:0	DR	Data register for transmission and reception	0	R/W

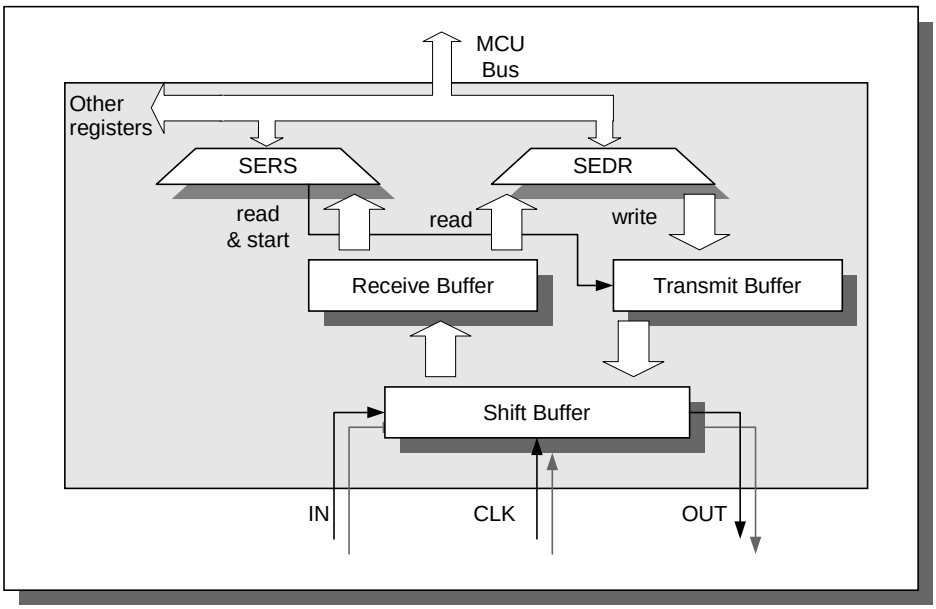


Figure 9.2.1 Data Paths in TXSEI

The actual shift register is buffered for both transmission and reception. The receive and transmit buffers are implemented as FIFO with a depth of four frames. A write to SEDR register writes the value to the transmit buffer. From there, the data will be transferred to the shift register as soon as TXSEI is ready for the next transfer. Reading the SEDR register delivers the current value from the receive FIFO and increments the receive FIFO pointer, if there are other values stored in the FIFO.

The shift-buffer is the physical register, which is used during SEI transfers for shifting in/out the data. Besides SEDR, the SERS register offers a second method to access the transfer values.

Data in both the SEDR and the SERS register are stored right aligned. E.g.: For eight bit transfers, only the lower eight bits of the SEDR register are used. For 16 bit transfers the lower 16 bits of SEDR are used. When reading this register, unused bits, except the upper 16 bits, are undefined. The upper 16 bits always read to zero.

SEI Read Start Register (SERS)

Bit	31	30	29	28	27	26	25	24
Name	—							

Bit	23	22	21	20	19	18	17	16
Name	—							

Bit	15	8
Name	RS[15:8]	

Bit	7	0
Name	RS[7:0]	

Bit	Name	Function	Reset Value	R/W
31:16	—	Unused	0x0000	R
15:0	RS	Read Start: Reads received data and immediately starts a transmission.	0	R/W

The SERS register offers a second method to fetch values from the receive buffer.

Reading this register returns the value from the receive buffer. Just like a read from the data register would. In contrast to reading the data register, the read from the SERS register counts for two register accesses: a read from the data register and a write of value 0xFFFF to the data register.

Therefore, in master mode a read access to this register will not only return the value from the receive buffer, but will also start a new transfer.

In slave mode, the received data will be delivered and the data 0xFFFF will be written to the data register, but the transfer will start when the master activates the slave select signal and switches on the SCLK clock. In order not to have TX buffer underruns the user should initially write some data via the SEDR into the TX buffer.

The register is useful during half-duplex operations, where data is read from SPI, while “don’t care data” is shifted out. It can be specified as DMA source address to save a valuable DMA channel during half-duplex transfers.

The register can only be read. Do not write to this register.

9.3 TXSEI Operations

There are five signals associated with SEI transfers multiplexed on the PIO pins. The use of each signal depends on the mode (master/slave) of the SEI device. Because the SEI signals are on a shared pin it is necessary to deactivate PIOs 16, 17, 18, 19 and 22 by writing a zero to bits 16, 17, 18, 19 and 22 in the PMUX register. Furthermore the selection whether TXSEI or UART0 functionality is mapped to the corresponding pins has to be done via the Chip Configuration Register (CCR).

A typical configuration consists out of one master device, which controls several slave devices. Only the master and one slave device are active at once. The master selects one slave for communication using the PORT pins to select each slave separately. Only the selected slave enables its port driver for the RX signal.

TXSEI offers a dedicated Slave Select input, which allows it to act on busses with multiple master devices. The dedicated Slave Select input guarantees a fast response to master's device selection on the bus.

PIO16/**CLK(SEI)**/CLK(SIO0) pin:

In master mode the CLK pin is used as an output, in slave mode it functions as input. When TXSEI is configured as master, the CLK signal is derived from the internal TXSEI clock generator depending on the SEI polarity and clock rate settings. When the master initiates a transfer, a programmable number of 5 to 16 clock cycles are automatically generated on the CLK pin. When TXSEI is configured as a slave, the CLK pin synchronizes data output and input to and from the external master. In both the master and slave SEI device, data is shifted on one edge of the CLK signal and is sampled on the opposite edge where data is stable. The edge polarity is determined by the SEI transfer protocol.

PIO18/**TX(SEI)**/TX(SIO0) and PIO17/**RX(SEI)**/RX(SIO1)

The RX and TX data pins are used for receiving and transmitting serial data. When the SEI is configured as a master, RX is the data input line, and TX is the master data output line. When the SEI is configured as a slave, these pins reverse roles.

PIO19/**SSI(SEI)**/CLK(SIO1) pin

The Slave Select Input port is used in Slave mode. The Slave Select Input signal is active low. If TXSEI's Slave Select Input is inactive, TXSEI will not follow the transmissions on the SEI bus.

If the Slave Select signal goes inactive during a running transfer and there are still other bits of the current transfer expected to receive, a Transfer Format Error will be signaled. The current value of the shift buffer will be transferred to the receive buffer despite of this error.

When TXSEI is configured to be the master and the SSI pin is asserted a transmission error will be recognized. This function can be disabled with the SSIVAL bit in the SECR0 register.

PIO22/**SSO(SEI)**/RTS(SIO1) pin

PIO22 is the dedicated slave select output signal and is asserted during transfer in master mode by the TXSEI device. In the case that the protocol of the connected device expects that the SS signal idles low for longer than 16 bits this signal must be generated using the PORT module.

9.3.1 TXSEI Transfer Format

During an SEI transfer, data is simultaneously transmitted (shifted out serially) and received serially (shifted in serially). The serial clock synchronizes shifting and sampling of the information on the two serial data lines.

The transfer format depends on the settings of the SPHA and SPOL registers in the SECR0 register. SPHA switches between two fundamentally different transfer protocols, which are described below.

9.3.1.1 SPHA Equals 0 Format

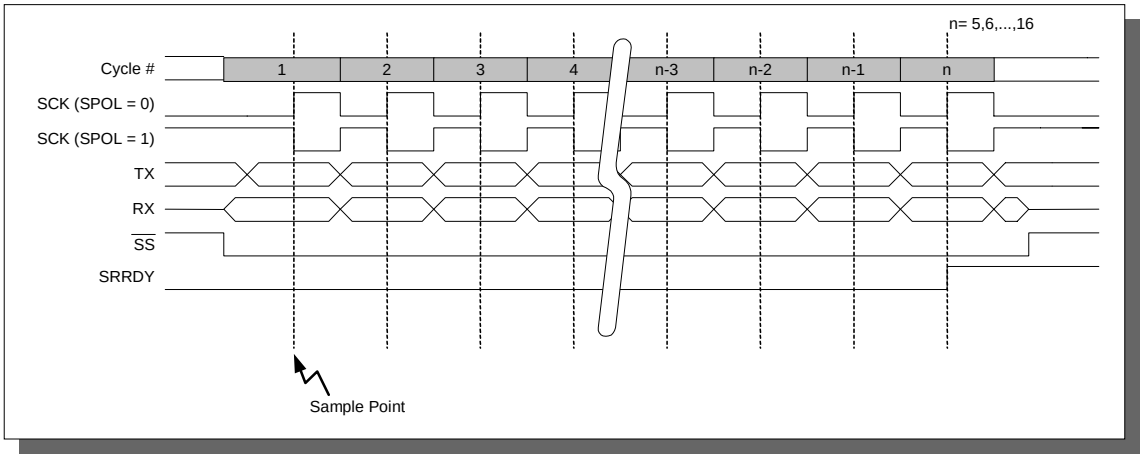


Figure 9.3.1 Protocol Timing for SPHA=0

In this transfer format, the bit value is captured on the first clock edge. This will be on a rising edge when SPOL equals zero and on a falling edge when SPOL equals one. The levels on the TX and RX signals change with the second clock edge on SCK. This clock edge will be a falling edge when SPOL equals zero and a rising edge, when SPOL equals one. With SPOL equal to zero, the shift clock will idle low. With SPOL equals 1 it will idle high.

In master mode, when a transfer is initiated by writing a new value to the SEDR register the new data is placed on the TX signal for half a clock cycle before the shift clock starts to operate. After the last shift cycle, the STRDY and SRRDY flags will be asserted.

In this format the SS signal has to be deasserted and reasserted between each successive frame. If the TXSEI is configured as a slave, SS has to be deasserted at least for the period of one TXSEI bit time (at least 8 system clocks).

9.3.1.2 SPHA Equals 1 Format

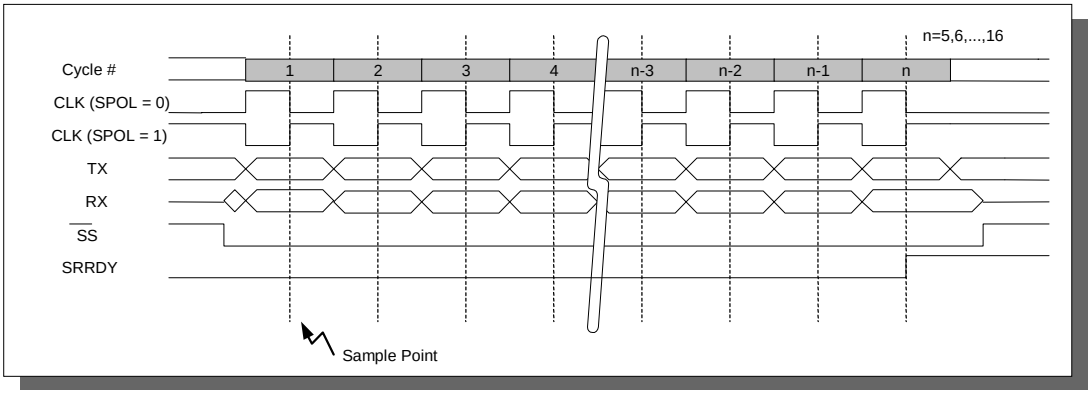


Figure 9.3.2 Protocol Timing for SPHA=1

In this transfer format, the first bit is shifted in on the second clock edge. This will be on a falling edge when SPOL equals 0 and on a rising edge when SPOL equals 1. If SPOL equals 0, the shift clock will idle low; with SPOL equals 1 it will idle high.

In master mode, when a transfer is initiated by writing a new value to the SEDR register the new data is placed on the TX signal with the first edge of the shift clock.

9.3.1.3 Inter-Frame Space Delay Mechanism

Due to its DMA support and its buffered shift register, TXSEI is able to sustain high data rates, with only a minimum amount of space between two consecutive frames.

However, between consecutive transfers it still has to be ensured that the slave device can keep up with the transfer rate of the SEI master. If TXSEI is configured as a master, the slave device typically has to write new values to its transmit buffer, before the next transfer can be started. To allow this, usually a minimum inter-frame space is specified considering interrupt response and data fetch time of the slave device.

TXSEI eases the implementation of this inter-frame space by offering an automated mechanism to guarantee inter-frame delays between consecutive frames.

The inter-frame space counter is implemented as a 10 bit down counter. The counter is reloaded with the value from the SEFS register after each transfer. The next transfer will not start before the IFS counter reaches a value of zero. The internal IFS counter is reset every time the SEFS register is written. Therefore, if the module is in the inter-frame space the next transfer will start immediately, if the register is written, even if the same value is rewritten to the register.

The following figure shows the function of the inter-frame space timer:

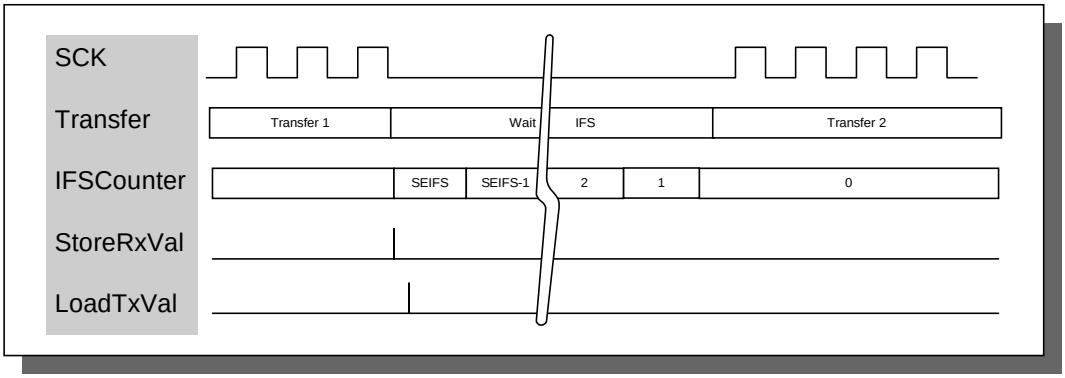


Figure 9.3.3 Waveform While Using Inter-Frame Space Timer

9.3.2 TXSEI Buffer Structure

TXSEI has both a transmit and a receive buffer. The buffers are implemented as FIFO and are able to store four frames each (one frame has a 16-bit length).

When a new TXSEI transfer is started by writing the data register, the transfer value is first stored in TXSEI's transmit buffer. From there the value will be fetched by the shift register immediately, if the module is idle or after the currently running transfer has completed.

A receive value from the shift register is stored in the receive buffer every time a transfer completes.

TXSEI is able to generate interrupts depending on the fill-level of these buffers. Therefore, it is possible to refill the buffers with several values within one interrupt service routine, if desired.

9.3.2.1 TXSEI System Errors

TXSEI is able to detect the following system errors during transfer:

9.3.2.2 SEOE – Overflow Error

An *Overflow Error* will be generated, when the receive buffer is completely filled, while a new value has been completely received on the SEI bus. In this case the data of the last transfer in the receive buffer is overwritten with the new value and the SEOE flag in the SESR register is asserted.

The SEOE register gives the programmer an indication, that data consistency during the transfer was lost.

9.3.2.3 SEUE – Underflow Error

An *Underflow Error* is generated, if the module is in slave mode and the bus master performs a shift, when no output value has been specified by writing to the data register.

9.3.2.4 SETF – Transfer Format Error

This error is generated, if the transfer format is violated.

There are two different scenarios, in which a Transfer Format Error could occur.

In slave mode, a transfer format error will be signaled, if:

- the slave select pin is used for TXSEI purpose (Configured in the port register) and
- TXSEI is in the middle of a transfer and
- the number of bits received yet is smaller than the number specified in the SSZ (SECR1) register

and if the slave select signal is set inactive at this point. It signals the user, that the master ended the transfer before the expected end of the transfer. A possible cause for this error could be different transfer length settings for master and slave devices. The partially received value will not be stored in the receive buffer since it is not complete.

In master mode, a transfer format error will be signaled, if:

- the slave select input signal is enabled in master mode (SSIVAL bit in SECR0 register) and
- both the master bit is set to one (MSTR bit in SECR0) and the system is in active mode (OPMODE=“10” in SEMCR) and
- the Slave Select signal is asserted.

9.4 Interrupts

TXSEI connects to three interrupt signals.

- SEIEXC: System Error Flags SEOE, SEUE, SETF, SIDLE (separately maskable)
- SEIRX Rx Buffer Fill Level Interrupt, Flag RBSI (not maskable)
- SEITX Tx Buffer Fill Level Interrupt, Flag TBSI (not maskable)

Interrupt SEIEXC is used for error detection purpose (SEOE, SEUE, SETF) and idle state interrupt (SIDLE). The interrupts SEIRX and SEITX are used to fetch and setup new data in an interrupt service routine for transferring data.

All the interrupts will occur one system clock cycle later than the internal flags, which are visible in the status register.

10. Asynchronous Serial Interface (UART)

The Asynchronous Serial Interface (UART) has the following features:

- Four channels
- Full-duplex transfer
- Baud rate generator
- Modem flow control (RTS/CTS), available on channels 1 and 2 only
- Transmit and receive FIFO, each of size 2 entries
- Multi controller system support (master/slave* operation capable)

10.1 Operations on Serial Interface

10.1.1 Outline

The UART is used to convert parallel data, retrieved from memory, to a serial stream with control and error detection bits. The same applies for data receive: only that a serial stream of data is converted to parallel by means of a shift register and then put into a FIFO. To retrieve data from memory or store data into memory one of the following policies is available:

- polling the Interrupt Status Register
- by serving an interrupt
- by DMA transfer.

The rate at which data is transferred (Baud Rate) can be programmed in the Baud Rate Control Register.

Care has to be taken in the configuration of the TMPR3916 to enable the I/Os of the UART.

* Please read carefully the documentation if the device has to be used for slave operation, since the TMPR3916 does not have open drain output pins.

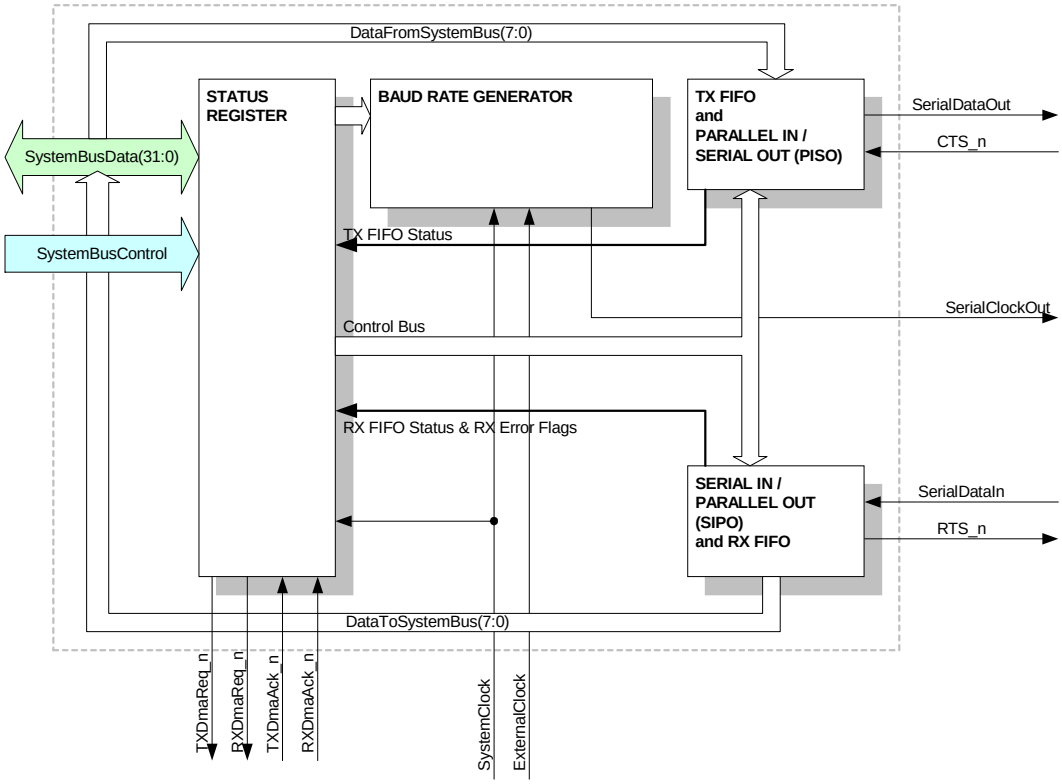


Figure 10.1.1 UART Block Diagram

10.1.2 Data Format

There are several data formats, which can be applied for serial I/O, these are summarized here:

Data length	7 – 8 – 9 bits (9-bit data is practicable for a multi controller system)
Stop bit	1 – 2 bits
Parity bit	provided/not provided
Parity system	even/odd
Start bit	1-bit fixed
data format	MSB/LSB first (switchable by register settings)

Please note that sending a parity bit is not allowed for address transmission in a multi-controller system, Figure 10.1.2 and Figure 10.1.3 show examples for these data formats:

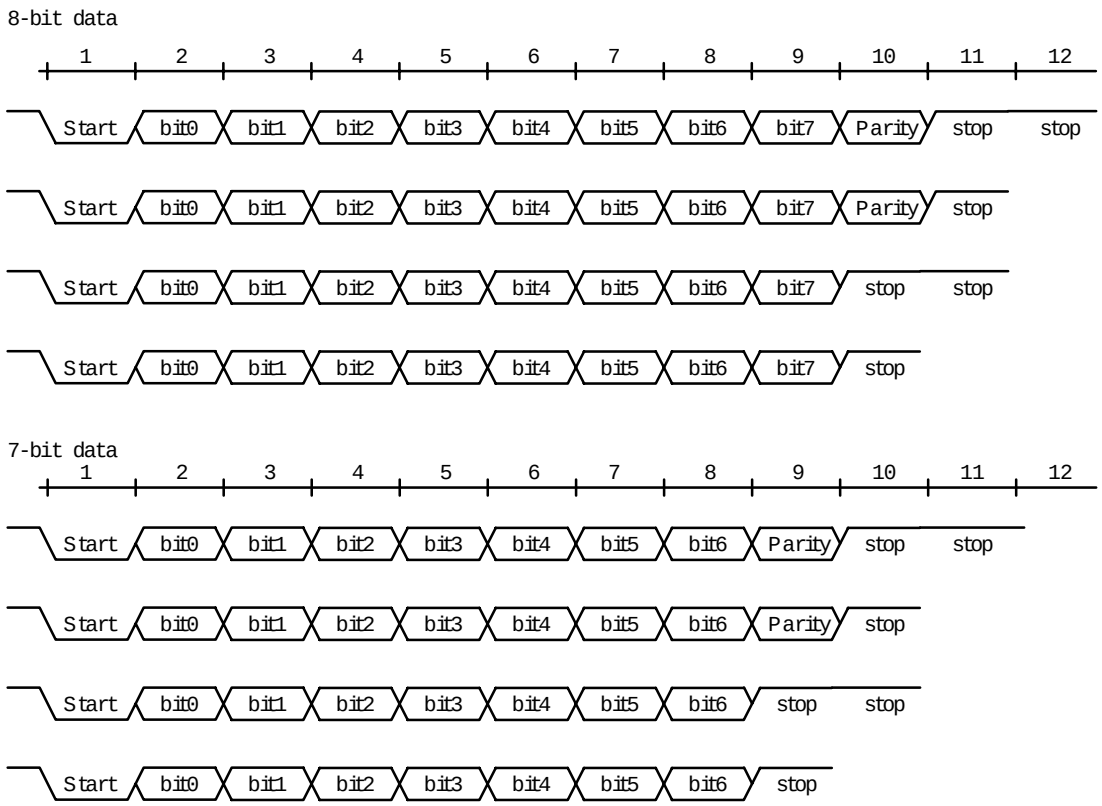


Figure 10.1.2 8 Bit and 7 Bit Data Format for Single Controller Mode, Data Formats are Shown for 1 and 2 Stop Bits, with and without Parity

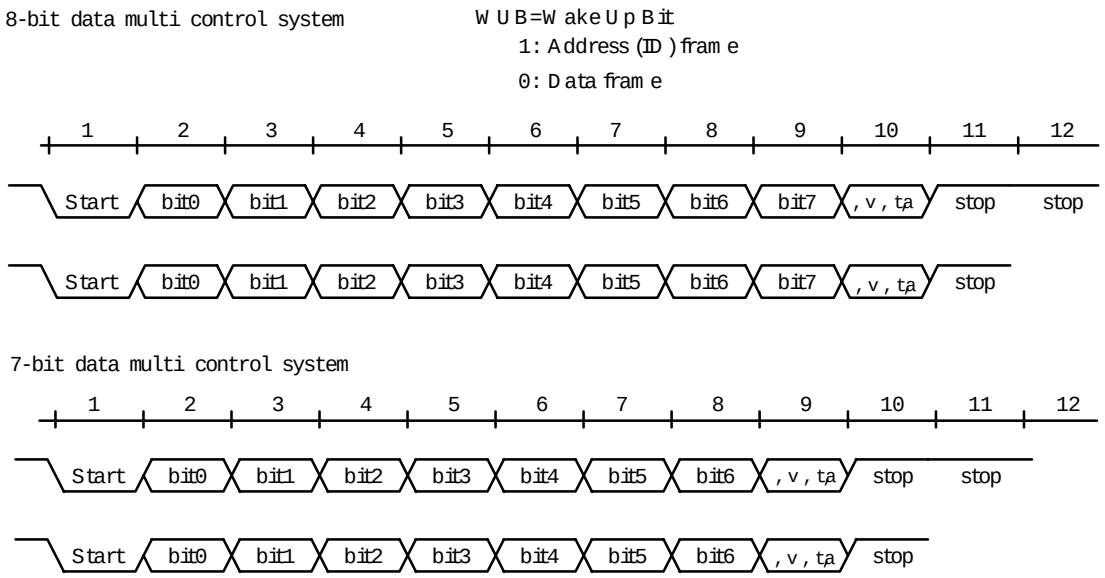


Figure 10.1.3 Multi Controller Mode Transmission, 7 and 8 Bit Data Formats are Shown with 1 or 2 Stop Bits, Parity Check is not Available in Multi Controller Mode

10.1.3 Serial Clock Generator

The data rate of serial transmissions depends on the Serial Clock. To generate the Serial Clock it is possible to use either an external clock or the system clock of the device. Please note that once an external clock is connected to the device's I/O pin the user **must** use this clock, otherwise a conflict will occur.

Once that the clock source is fixed, this is used to generate the Serial Clock by means of a prescaler (optional) and a Baud Rate Divisor.

Figure 10.1.4 shows a block diagram of the Serial Clock Generator:

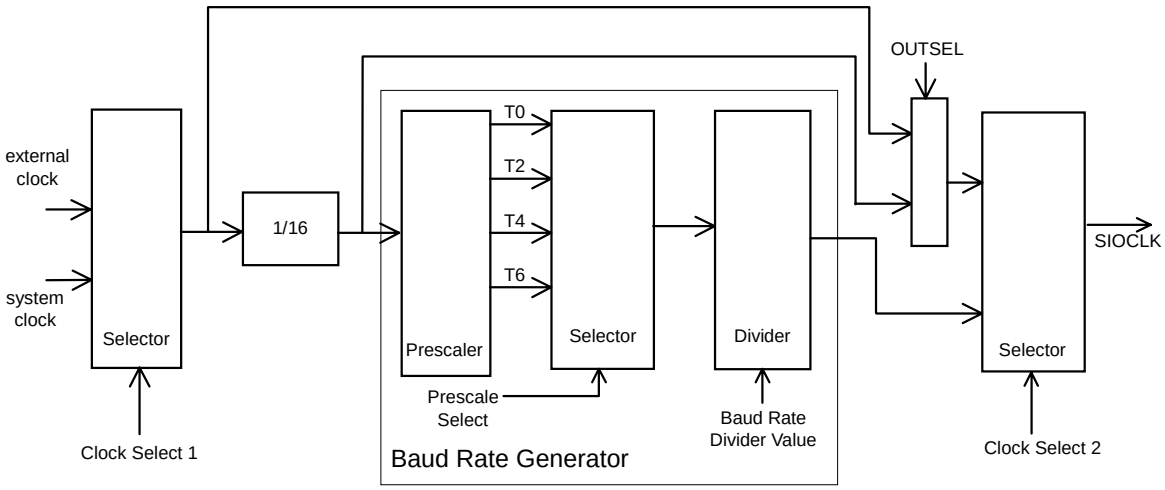


Figure 10.1.4 Serial Clock Generator Structure

The Clock Select 1 and 2 control lines are bits 6 and 5 respectively of the Line Control Register and allow the user to choose between an internal or external clock, and if the Baud Rate Generator has to be bypassed. The output clock – if the BRG is bypassed – can be of the same frequency as the input clock or divided by 16, depending on the value of OUTSEL (Line Control Register).

The SIOCLK is not used as system clock for FIFOs etc, but to synchronize operations. The SIOCLK is output as Serial Clock Out. The OUTSEL bit will affect the frequency of the Serial Clock Out but not the baud rate of the UART.

The **baud rate generator** creates the transmit/receive clock, which regulates the transfer rate for the serial interface. The baud rate can be calculated by the following formula:

$$\text{baud rate} = \frac{\text{input frequency} \times \text{prescaler}}{(\text{divisor of baud rate generator} + 1)} \div 16$$

The input clock (either system or external clock) may be prescaled by the values shown in the SIBGR registers and then divided according to the BRD field of the same register. Pay attention to the fact that the clock is actually divided by a value of (BRD+1). The BRD is an 8-bit register, so the divider ranges from 1 to 256.

Table 10.1.1 shows the output frequency of the baud rate generator (SIOCLK), depending on the baud rate divisor and the prescaler:

Desired Baud Rate	Baud Rate Input	Prescaler	Baud Rate Divisor	Actual Baud Rate
9600	60 MHz	½ (000)	194 (0xC2)	9615
57600	60 MHz	½ (000)	32 (0x20)	56818
115200	60 MHz	½ (000)	15 (0x0F)	117187

Table 10.1.1 Baud Rate Register Settings for Common Baud Rates

Other Baud Rates may be easily calculated. When choosing the value of the Prescaler and the Baud Rate Divisor we recommend that the actual baud rate should not differ more than 4.0% from the nominal value. This is because a certain drift is allowed but if the 4.0% figure is exceeded then the correct transmission of a frame is not guaranteed anymore.

10.1.4 Transmitter Control

Once that the parameters have been set, writing data into the Transmit FIFO is sufficient for transmission. During transmission also the parity bit will be calculated and added to the data stream if the option has been enabled. The data transmission rate will be the one set in the Baud Rate Register. Make sure that both receiver and transmitter have the same settings for Baud Rate, data bits, parity, and stop bits.

The transmitter shift register is an 8-bit shift register, that gets its data from the transmitter FIFO. Bit 0 of the shift register will be sent first.

10.1.5 Receiver Control

After that the receiver is enabled, the controller will be looking for the start bit on the serial input line (RX). A “0” on serial input only will be recognized as start bit, if a “1” was detected in the bit before. When the receive control detects a start bit, the receiving operation will start.

The output of the baud rate generator (SIOCLK) is 16 times the frequency of the data transfer rate on serial interface. The serial data input (RX) will be sampled on 7th, 8th and 9th clock of SIOCLK. A majority logic determines the input value.

The **receiver shift register** consists of an 8-bit shift register. At the end of a transmission, bit 0 of the shift register contains the bit which has been received first.

10.1.6 Host Interface

The data transfer to the transmitter FIFO can be handled via interrupt processing or via DMA transfer. If the transmitter FIFO has as many free entries as set in the transmit DMA interrupt trigger level (TDIL in FIFO control register), an interrupt or DMA request is generated. Afterwards the DMA Controller or the software fetches data from memory and writes it to the transmitter FIFO.

The data transfer from the receiver FIFO can be handled via polling, interrupt processing or via DMA transfer. If the receiver FIFO has as much free space as set in the receive DMA interrupt trigger level (RDIL in FIFO control register), an interrupt or DMA request is generated. Afterwards the DMA Controller or the software fetches data from the receive FIFO and writes it to memory.

The following settings of TDR, RDR, TIR and RIR of Interrupt Control Register (SIDICR) are allowed:

TDR	RDR	TIR	RIR	Transmit	Receive
0	0	0	0	TDIS polling	RDIS polling
0	0	0	1	TDIS polling	Interrupt
0	0	1	0	Interrupt	RDIS polling
0	0	1	1	Interrupt	Interrupt
0	1	0	0	TDIS polling	DMA
0	1	1	0	Interrupt	DMA
1	0	0	0	DMA	RDIS polling
1	0	0	1	DMA	Interrupt
1	1	0	0	DMA	DMA

Table 10.1.2 DMA/Interrupt Control Register Settings for Host Interface Operation

10.1.7 Flow Control

Transmission enable can be set either

- via software control by a transmit serial data request of the MPU (TSDR).
- via hardware control.

When the transmit enable becomes inactive, the transmission will be suspended after the completion of the current frame transmission.

Reception is enabled either

- via RTS software control of the MPU (RTSSC) or
- via hardware control.

For hardware control, the flow control offers the possibility of a DMA transfer or an interrupt request. This choice can be made by configuring the Interrupt Control Register (SIDICR). Remember that RTS/CTS flow control has to be enabled by Flow Control Register (SIFLCR).

During transfers, the receiver can ask for a temporary suspension by setting to 1 the RTS signal. The transmission is resumed by setting to 0 the RTS signal when the receiver is ready to accept new data.

Frame by frame data transfer is available by setting the transmitter to hardware control (TES=1) and the receiver RTSTR to 1 (handshaking).

10.1.8 Parity Control

During **transmission**, parity information is generated when the data is written to the transmitter shift register. The information is stored

- in bit 7 of the transmitter shift register for 7-bit data length or
- in TWUB of the line control register for 8-bit data length.

During **reception**, the parity check is executed when the data is written from receiver shift register to the receiver buffer. A parity error occurs when a difference between received and calculated parity bit is found. The check information is stored

- in bit 7 of the read buffer for 7-bit data length or
- in RWUB of the line control register for 8-bit data length.

10.1.9 Interrupts

The UART can send three interrupt signals to the Interrupt Controller of the TMPR3916:

- Transmit Interrupt
- Receive Interrupt
- Exception Interrupt

A brief explanation of the meaning of each of these interrupts now follows, for further details refer to the Registers section of the document. It is possible to mask the interrupts by programming the SIDICR register. The cause of the interrupt, however, is latched and available for polling.

Transmit Interrupt

A Transmit Interrupt may occur when:

- There is free space in the TX FIFO, hence new data may be written in the FIFO.

Receive Interrupt

A Receive Interrupt may occur when:

- There is valid data in the RX FIFO, which has not been retrieved yet.
- There is a Time Out on the Receiver and the Time Out is mapped on the RX interrupt (programming RIR bit of Interrupt Control Register)
- There is an error (parity, frame or overrun) detected in the RX FIFO.

Exception Interrupt

A Status Interrupt may occur when:

- There is a BREAK in the UART transfer (see section “10.1.11 Breaks”).
- All data in transmit FIFO has been sent.
- There is free space in TX FIFO.
- A CTS active has been detected by the transmitter.
- There is an Overrun Error.
- There is a Frame error.
- There is a Parity error.

10.1.10 Error Flags

The following error flags can be risen:

Overrun error

Occurs when there is an overflow of the transmit or of the receive buffer.

Parity error

Occurs when the received parity bit and the calculated parity information are not the same.

Frame error

Occurs when 0 is detected at the stop bit position during receive.

10.1.11 Breaks

A break is detected by the receiver when all bits in a frame are 0 and the stop bit is also 0. When a break is detected the received data will not be stored and the UART will assert an interrupt to signal the break condition.

To transmit a break, the TBRK bit of the flow control register must be set to 1. To resume normal operation the TBRK bit must be set to 0 again.

10.1.12 Receiver Time Out

A receiver time out occurs when the receiver FIFO has at least 1 entry stored and an equivalent of a 2-byte receive time is elapsed from the previous reception. That sets the receiver Time Out bit (TOUT) in the DMA/Interrupt Status register(SIDISR). Please note that if all data has been fetched from the FIFO, the Time Out will not be set!

It is recommended not to use the TOUT bit for flow control of the receiver.

10.1.13 Handling of received data and receiver FIFO status bits

The following status information is stored in the receiver FIFO along with the received data:

- UART receiver break (UBRK)
- UART available status (UVALID)
- UART frame error (UFER)
- UART parity error (UPER)
- UART overrun error (UOER)

The software can read the status in the DMA/interrupt status register (SIDISR). The status of this register is updated always after data has been read from the FIFO: if there are 2 entries in the FIFO, and the second one contains a parity error, this will be signaled in the SIDISR only after the first entry has been read.

Data that has been received without errors can be read from the receiver FIFO. A Receive Interrupt Request will indicate the presence of new entries in receiver FIFO, if this is enabled. Errors during reception will be indicated by an exception interrupt request.

Only data that has been received without errors is transferable via DMA. When an error (UFER, UPER, UOER) or a receive time-out (TOUT) occurs, the receive error is notified and the received data transfer request is not asserted.

10.1.14 Multi Controller Systems

When UMODE in the Line Control Register (SILCR) is 0x2 or 0x3, the UART changes into the multi controller system mode. In a multi controller system, the master controller sends data to the selected slave controllers. The slaves will be selected by sending an address ID before sending the data. Non-selected slave controllers will ignore the data. Setting the WUB of the transmit frame to 1 indicates the transmission of an address ID. Data frames, instead, will have the WUB set to 0. It is up to the software to compare the addresses and eventually enable the reception of Data Frames.

Protocol of multi controller system:

(1) Master and slave controllers must have the UMODE field 10 or 11 in the Line Control Register to operate in multi controller mode.

(2) Software sets the RWUB in the Line Control Register to 1 – for each slave controller – to be ready to receive the address ID frame from the master controller.

(3) Software sets the WUB of the transmit frame to 1 (Line Control Register WUB=1) – in the master controller – to send the slave controller address ID (7- or 8-bit length).

(4) An interrupt is generated in a slave controller when RWUB in Line Control Register is 1 and WUB of the received data frame is 1 (receive data is an address frame). The software compares its address ID with the received address ID and sets RWUB to 0 when both match. The last operation enables the selected slave to receive data frames.

(5) Then software must set the WUB of the Line Control Register to 0 to enable the transmission of data frames for the master controller. If this operation is not performed the WUB in the transmitted frames will be 1, and the slaves will understand the incoming frames as address frames.

When the selected slave controller (i.e. the controller whose address matched with the one sent by the master) receives data, an interrupt is generated.

A non selected slave controller will have the RWUB still set to 1, incoming data frames will be ignored.

The slave controllers can send data only to the master controller.

An **example** of the multi controller system configuration is shown in Figure 10.1.5

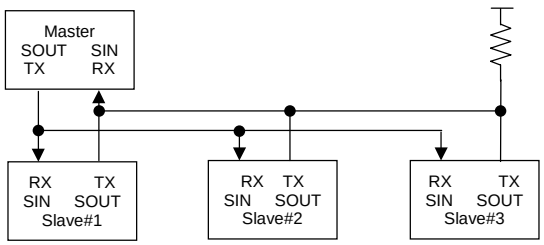


Figure 10.1.5 Example Configuration for Multi Controller System

IMPORTANT: The TMPR3916 does not have Open Drain capability on its output pins. A configuration as the one depicted in Figure 10.1.5 is possible only if external logic is added on the board.

10.2 Registers

10.2.1 Overview

Register (short name)	Physical Address (hex)	*Name*	Function
SILCR	1C00 0000	*Line Control Register*	Specify data format
SIDICR	1C00 0004	*DMA/Interrupt Control Register*	Controls settings about interrupt and DMA requests
SIDISR	1C00 0008	*DMA/Interrupt Status Register*	Shows status information about interrupt and DMA requests
SISCISR	1C00 000C	*Status Change Register*	Shows status information of UART transfer
SIFCR	1C00 0010	*FIFO Control Register*	Controls settings of transmit/ receive FIFO
SIFLCR	1C00 0014	*Flow Control Register*	Controls running transmission
SIBGR	1C00 0018	*Baud Rate Control Register*	Contains baud rate settings
SITFIFO	1C00 001C	*Transmitter FIFO Register*	Transmit data
SIRFIFO	1C00 0020	*Receiver FIFO Register*	Received data

This table includes the addresses of UART channel 0.

Channel 1 uses the addresses 1C00 0040 to 1C00 0060.

Channel 2 uses the addresses 1C00 0080 to 1C00 00A0.

Channel 3 uses the addresses 1C00 00C0 to 1C00 00E0.

10.2.2 Line Control Register (SILCR)

Bit	31							16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	RWUB	TWUB	—	—		OUTSEL	IRDA	LSBF

Bit	7	6	5	4	3	2	1	0
Name	—	SCS		UEPS	UPEN	USBL	UMODE	

Bit	Name	Function	Reset Value	R/W
31:16	—	Wired to zero	0	R
15	RWUB	Wake Up Bit for Receive 0 = The UART does not wait for a wake-up-bit 1 = The UART is looking for a wake-up-bit Used only in multi controller mode.	0	R/W
14	TWUB	Wake Up Bit for Transmit 0 = Next frame contains data (wake-up-bit = 0) 1 = Next frame contains address (wake-up-bit = 1) The contents of this bit make only sense in multi controller mode.	1	R/W
13	—	Not used / Always write to “0”	0	R/W
12:11	—	Wired to zero	00	R
10	OUTSEL	Clock Output Select 0 = Clock frequency is the same as in Baud Rate Register (SIBGR) specified 1 = Clock frequency is 16 times the value specified in Baud Rate Register (SIBGR)	0	R/W
9	IRDA	IrDA Clock 0 = No output of IrDA clock 1 = Output of IrDA clock It has no meaning when OUTSEL = 1.	0	R/W
8	LSBF	LSB First 0 = Reads or sends the MSB first 1 = Reads or sends the LSB first	0	R/W
7	—	Wired to zero	0	R
6:5	SCS	SIO Clock Select 00 = Internal system clock 01 = Baud rate generator provided with internal clock 10 = External clock 11 = Baud rate generator provided with external clock	10	R/W
4	UEPS	UART Parity Select 0 = Odd Parity 1 = Even Parity	0	R/W
3	UPEN	UART Parity Enable 0 = Disable Parity Check 1 = Enable Parity Check The bit should be 0 in the multi controller system mode (UMODE = 10, 11).	0	R/W
2	USBL	UART Stop Bit Length 0 = 1 bit 1 = 2 bit	0	R/W

Bit	Name	Function	Reset Value	R/W
1:0	UMODE	UART Mode For the SIO Mode setting. 00 = 8-bit data length 01 = 7-bit data length 10 = Multi controller 8-bit data length 11 = Multi controller 7-bit data length	00	R/W

The reception of a 1-bit stop while in a 2-bit stop length setting does not generate a frame error.

When an external clock is connected to the Serial Clock In pin of the TMPR3916, bit 6 of the SILCR must be set to 1 to avoid conflicts. It is not possible to connect an external clock and decide to use the system clock for UART operation.

The OUTSEL bit will have effect only on the frequency of the Serial Clock Out. The Baud rate of the UART will not depend on the settings of the OUTSEL bit.

The Wake Up Bit for Receive has to be set to 1 if the receiver has to be enabled for receiving an address frame prior to the data frames. The Wake Up Bit for Transmit has to be set to 1 if an address frame is being sent prior to sending data frames (an address frame may be followed by several data frames to the same receiver). Obviously these two bits make sense only in multi controller mode. In a point-to-point transmission there's no need to address the receiver.

10.2.3 DMA/Interrupt Control Register (SIDICR)

Bit	31							16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	TDR	RDR	TIR	RIR	SPIR	CTSAC		—

Bit	7	6	5	4	3	2	1	0
Name	—		SIOE	SICTS	SIBRK	SITR	SIAS	SIUB

Bit	Name	Function	Reset Value	R/W
31:16	—	Wired to zero	0	R
15	TDR	Transmit DMA Request 0 = No DMA request when free space in transmit FIFO 1 = DMA request when free space in transmit FIFO	0	R/W
14	RDR	Receive DMA Request 0 = No DMA request when data in receive FIFO 1 = DMA request when data in receive FIFO	0	R/W
13	TIR	Transmit Interrupt Request 0 = No interrupt when free space in transmit FIFO 1 = Send SIOTX interrupt when free space in transmit FIFO	0	R/W
12	RIR	Receive Interrupt Request 0 = No interrupt when error or time out occurs 1 = When error or time-out occurs send • SIORX interrupt, when RDR=0, • SIOEXC interrupt, when RDR=1	0	R/W

Bit	Name	Function	Reset Value	R/W
11	SPIR	Special Interrupt Request 0 = No interrupt when errors occur 1 = Send SIOEXC interrupt, when errors occur	0	R/W
10:9	CTSAC	CTSS Status Active Condition Sets condition of CTS. 00 = Disable CTS 01 = CTS terminal rising edge 10 = CTS terminal falling edge 11 = Both edges	0	R/W
8:6	—	Wired to zero	0	R
5	SIOE	Special Interrupt on Overrun Error 0 = No actions on overrun error 1 = When an overrun error occurs, send SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR)	0	R/W
4	SICTS	Special Interrupt on Receive of CTS 0 = No actions on CTS 1 = When receiving CTS, send SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR)	0	R/W
3	SIBRK	Special Interrupt on Break of UART Transfer 0 = No actions on break 1 = When a break occurs, send SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR) This bit must have the same value of SIUB	0	R/W
2	SITR	Special Interrupt on Free Space in Transmit FIFO (RBRKD) 0 = No actions on free space in transmit FIFO 1 = When free space is detected, send SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR)	0	R/W
1	SIAS	Special Interrupt, when all data sent 0 = No actions, when all data sent 1 = When all data sent, transmit SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR)	0	R/W
0	SIUB	Special Interrupt on Break of UART Transfer (UBRKD) 0 = No actions on break 1 = When a break occurs, send SIOEXC interrupt and set STIS bit in Interrupt Status Register (SIDISR) This bit must have the same value of SIBRK	0	R/W

The SIUB and SIBRK bits have the same behavior for the Exception Interrupt, since there is only one Break flag in the DMA/Interrupt Status Register (SIDISR). However the SIBRK masks the UART being in break status, the SIUB masks the detection of a BREAK. Obviously as soon as a Break is detected (if any of the two is unmasked) the UART will send an Interrupt Request. To know if a Break has already occurred or is still active refer to the Status Change Register (SISCIR). Setting these two bits to 1 or 0 will not affect the behavior of the Status Change Register.

10.2.4 DMA/Interrupt Status Register

Bit	31							16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	UBRK	UVALID	UFER	UPER	UOER	ERI	TOUT	TDIS

Bit	7	6	5	4	3	2	1	0
Name	RDIS	STIS	—	RFDN				

Bit	Name	Function	Reset Value	R/W
31:16	—	Wired to zero	0	R
15	UBRK	Break This bit will be set to 1, when break is detected	0	R
14	UVALID	No Data Available This bit will be set to 1, when the receiver FIFO contains no data.	1	R
13	UFER	Frame Error This bit will be set to 1, when an error occurred during transfer of the current frame.	0	R
12	UPER	Parity Error This bit will be set to 1, when a parity error has been detected.	0	R
11	UOER	Over Run Error This bit will be set to 1, when an overrun error has occurred.	0	R
10	ERI	Error Interrupt This bit will be set to 1, when a frame error, parity error or overrun error has occurred. Writing 0 to this bit clears it. Writing 1 does not change contents of this bit.	0	R/W
9	TOUT	Receive Time Out This bit will be set to 1 immediately after a receive time out occurs. Writing 0 to this bit clears it. Writing 1 does not change contents of this bit.	0	R/W
8	TDIS	Transmit DMA/Interrupt Status This bit will be set to 1, when there is free space in the transmit FIFO.	1	R/W
7	RDIS	Receive DMA/Interrupt Status This bit will be set to 1, when there are valid data in the receive FIFO.	0	R/W
6	STIS	Status Interrupt Status This bit will be set to 1, when the status, selected in STIR of Interrupt Control Register (SIDICE), has changed.	0	R/W
5	—	Wired to zero	0	R
4:0	RFDN	Receive FIFO Data Number Status Indicating the number of received data frames stored in the receiver FIFO (0 to 2 entries).	00000	R/W

UBRK, UPER and UOER show the status of the upper FIFO entry. After reading the next data entry, the UART will update the status information in UBRK, UPER and UOER. That is the reason, why the software must read status information before reading the data.

10.2.5 Status Change Register (SISCISR)

Bit	31							8
Name	—							

Bit	7	6	5	4	3	2	1	0
Name	—		OERS	CTSS	RBRKD	TRDY	TXALS	UBRKD

Bit	Name	Function	Reset Value	R/W
31:6	—	Wired to zero	0	R
5	OERS	Overrun Error This bit will be set to 1, when an overrun error occurs. Cleared by writing 0	0	R/W
4	CTSS	CTS terminal Indicates the CTS terminal status. 0 = CTS is deasserted 1 = CTS is asserted	0	R
3	RBRKD	Receive Break This bit will be set to 1, when the UART is in break status.	0	R
2	TRDY	Tx Ready Set to 1 when the transmitter FIFO has free space at least for one entry of data.	1	R
1	TXALS	Tx All Set to 1 when transmitter FIFO and transmitter shift register are empty.	1	R
0	UBRKD	UART Break Detect Set to 1 immediately when a break is detected. Cleared by writing 0.	0	R/W

UBRKD will be set to 1 as soon as a Break is detected, it will remain 1 until it is explicitly cleared by the software. RBRKD, instead, will be 1 only for the duration of the Break.

10.2.6 FIFO control register (SIFCR)

Bit	31							16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	SWRST	—						RDIL

Bit	7	6	5	4	3	2	1	0
Name	RDIL	—		TDIL		TFRST	RFRST	FRSTEW

Bit	Name	Function	Reset Value	R/W
31:16	—	Wired to zero	0	R
15	SWRST	Software Reset 0 = Normal operation 1 = Softreset of UART This software reset lasts for 4 clock cycles. The channel will not react to any requests during this time period. Warning: While using instruction cache of TX39 the following problem occurs: As it takes about 5 clock cycles to activate software reset this might affect the next write action to a register of the channel being reset. Solution: Insert other instructions between resetting command and next write command of the same channel.	0	W
14:9	—	Wired to zero	0	R
8:7	RDIL	Receive DMA/Interrupt trigger level These bits determine at which fill level of the receive FIFO the UART sends an interrupt or DMA request. 00 = If 1 byte in receive FIFO make a request 01 = If 2 bytes in receive FIFO make a request others = invalid setting	00	R/W
6:5	—	Wired to zero	0	R
4:3	TDIL	Transmit DMA/Interrupt trigger level These bits determine at which fill level of the transmit FIFO the UART sends an interrupt or DMA request. 00 = If 1 byte in transmit FIFO make a request 01 = If 2 bytes in transmit FIFO make a request others = Invalid setting	00	R/W
2	TFRST	Transmit FIFO Reset 0 = Normal operation 1 = Reset of transmit FIFO (only when FRSTEW = 1)	0	R/W
1	RFRST	Receive FIFO Reset 0 = Normal operation 1 = Reset of receive FIFO (only when FRDTEW = 1)	0	R/W
0	FRSTEW	FIFO Reset Enable 0 = Resets of receive and transmit FIFO are inhibited 1 = Resets of receive and transmit FIFO are possible	0	R/W

To properly reset the Receiver or the Transmitter FIFO, bit FRSTEW has to be set to 1 <u>together and at the same time</u> in which TFRST or RFRST are set to 1. E.g.: to reset the Receiver FIFO value 0x00000003 must be written. This operation will reset the Receiver FIFO until either RFRST or FRSTEW are set to 0 again by writing in the SIFCR register

To perform a software reset, the value of the FRSTEW bit is meaningless.

10.2.7 Flow control register (SIFLCR)

Bit	31							16
Name	—							

Bit	15	14	13	12	11	10	9	8
Name	—			RCS	TRS	—	RTSSC	RSDR

Bit	7	6	5	4	3	2	1	0
Name	TSDR	—		RTSTL				TBRK

Bit	Name	Function	Reset Value	R/W
31:13	—	Wired to zero	0	R
12	RCS	RTS Control Select Selects the method to control the RTS terminal. 0 = Software control 1 = Software or hardware control	0	R/W
11	TRS	Tx Request Select Selects the transmit request. 0 = Control by transmit serial data request (TSDR). 1 = Control by transmit request command or the CTS terminal (hardware control)	0	R/W
10	—	Wired to zero	0	R
9	RTSSC	RTS Software Control Determines the output of the RTS terminal. 0 = Sets the RTS terminal to 0 1 = Sets the RTS terminal to 1	0	R/W
8	RSDR	Receive Serial Data Request 0 = Received data will stored 1 = Received data will not be written into FIFO	1	R/W
7	TSDR	Transmit Serial Data Request 0 = Transmission runs 1 = Halts transmission. A running transmission will be finished.	1	R/W
6:2	—	Wired to zero	0	R
1	RTSTL	RTS Trigger Level Sets the RTS hardware control assert level at the number of receive data entries in the receiver FIFO. Possible settings: 01, 10	1	R/W
0	TBRK	Transmitter Break 0 = Normal operation 1 = Transmit a break	0	R/W

When the RSDR bit is set to 1, received data will be ignored: it will not be written into FIFO. Frame, Overrun and parity errors will not be generated.

10.2.8 Baud rate control register (SIBGR)

Bit	31	16
Name	—	

Bit	15	11	10	9	8	7	0
Name	—		BCLK			BRD	

Bit	Name	Function	Reset Value	R/W
31:11	—	Wired to zero	0	R
10:8	BCLK	Baud Rate Generator Clock Specifies the prescaler for the input clock of baud rate generator. 000 = 1/2 system frequency 001 = 1/8 system frequency 010 = 1/32 system frequency 011 = 1/128 system frequency 1xx = system frequency (prescaler bypass)	011	R/W
7:0	BRD	Baud Rate Divisor Set the baud rate divisor.	0xFF	R/W

10.2.9 Transmitter FIFO register (SITFIFO)

Bit	31	8	7	0
Name	—		TXD	

Bit	Name	Function	Reset Value	R/W
31:8	—	Wired to zero	0	R
7:0	TXD	Transmit data Data written to this register are carried to transmit FIFO. Note: The bits are write-only.	0	W

10.2.10 Receiver FIFO register (SIRFIFO)

Bit	31	8	7	0
Name	—		RXD	

Bit	Name	Function	Reset Value	R/W
31:8	—	Wired to zero	0	R
7:0	RXD	Receive Data Read this register to get next data item from the receiver FIFO.	0	R

The Receiver FIFO Register can only be read by a 32-bit-word access.

10.3 Timings

Receive operation:

(7, 8-bit data length)

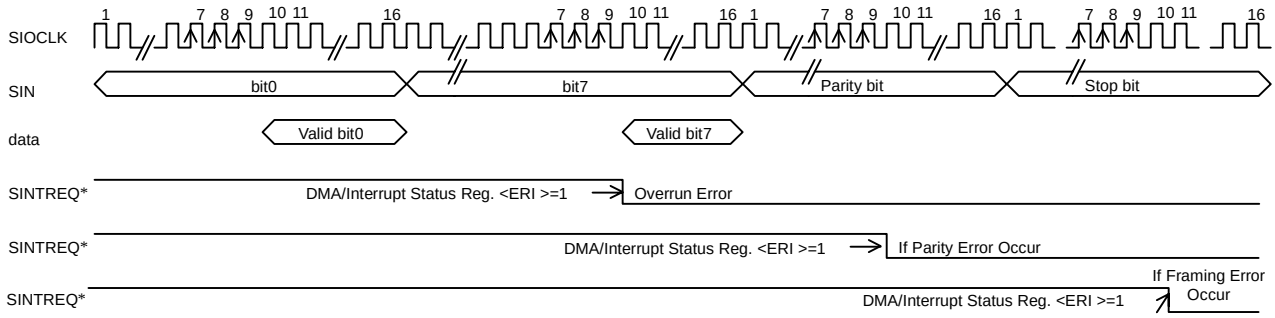


Figure 10.3.1 Receiving 7 or 8 Bit Data

Receive operation:

(7, 8-bit length multi controller system; RWUB=1 for ID receive standby)

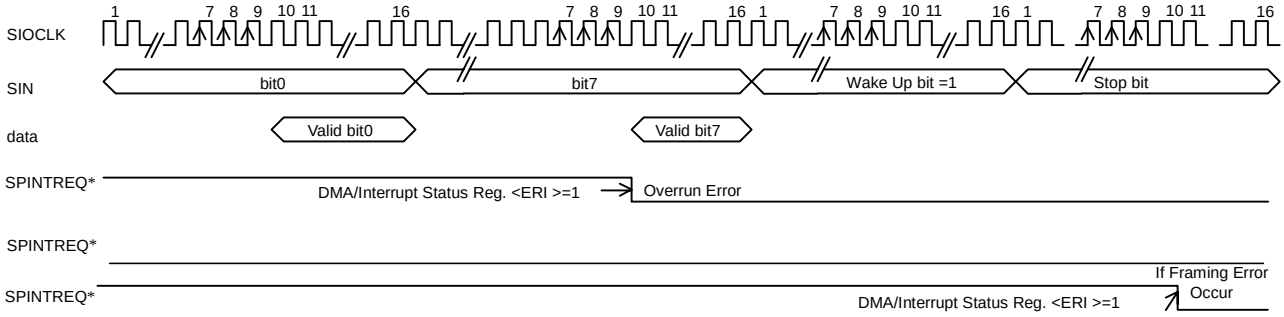


Figure 10.3.2 Receiving 7 or 8 Bit ID in Multi Controller Environment

Receive operation:

(7, 8-bit length multi controller system; RWUB=0 for data receive standby)

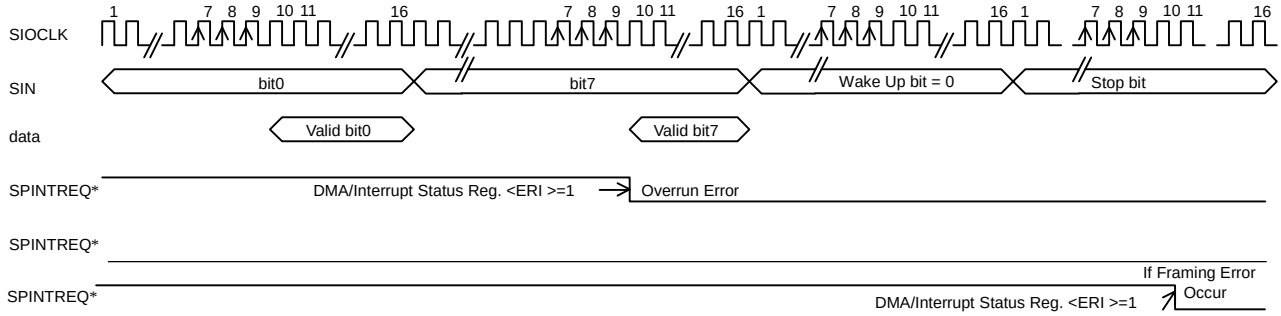


Figure 10.3.3 Receiving 7 or 8 Bit Data in Multi Controller Environment

Receive operation:

(7, 8-bit length multi controller system; RWUB=1 for data skip)

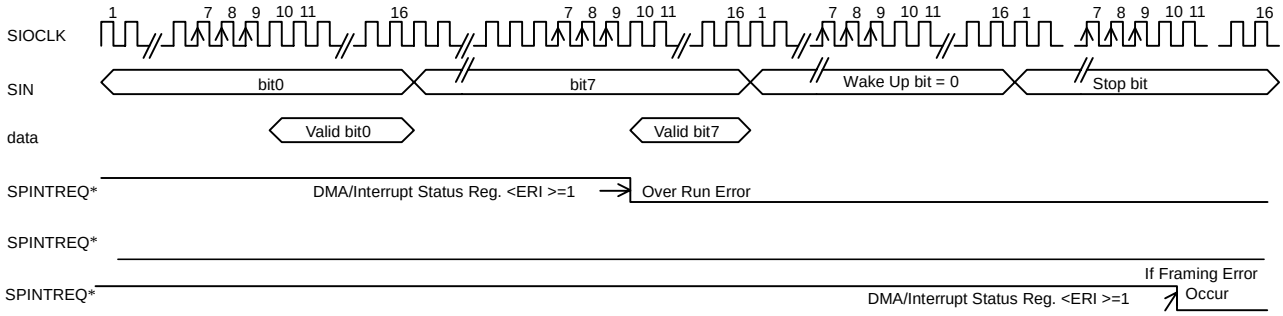


Figure 10.3.4 Receiving 7 or 8 Bit Data Skip Multi Controller Environment

Transmit operation

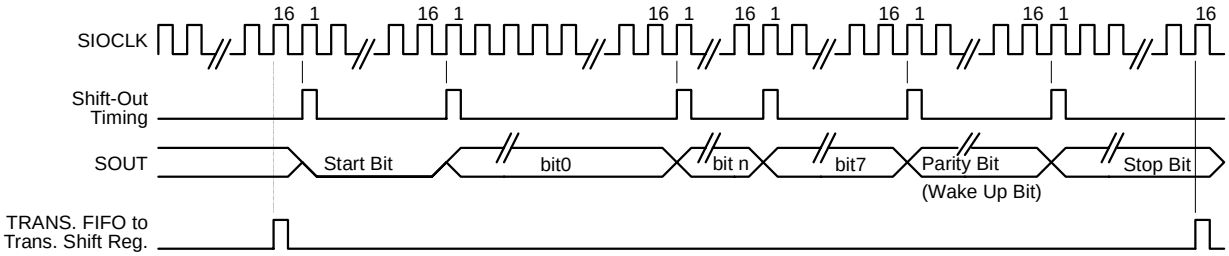


Figure 10.3.5 Transmitting 8 Bit Data

Transmit halt timing by CTS*

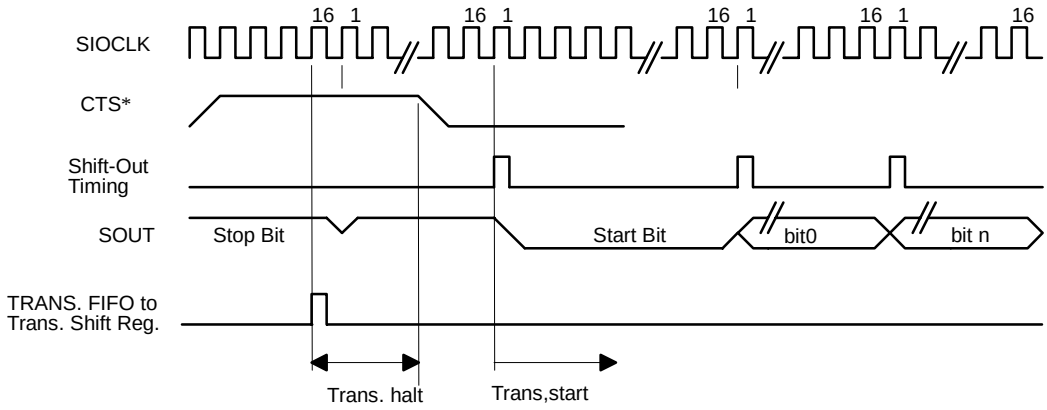


Figure 10.3.6 Transmitting HALT Command

When CTS becomes 1 during data transmission, the data transfer halts after completing the current data transmission. Despite the halt, the next data is stored in the transmitter shift register. The transmission is restarted at the first shift out pulse after CTS becomes low.

10.4 Flow Charts

In this paragraph some flow charts are presented, which should help the user to perform basic operations on the UART.

In these examples an upper layer which involves a software protocol is used: the transmitter will send as first data frame a “block length” which will be used by the receiver to determine the number of incoming data frames. So if *n* data frames have to be transmitted, the actual length of the transmission will be of *n+1* since the protocol uses the first frame as a control frame.

The user may choose to implement or not such a protocol, the reason for using it here is to give an easy example on how to operate the UART.

It is not recommended to use the Timeout error in the receiver to determine the end of a transmission: if the FIFO is empty the error will not occur. This bit is not suitable for flow control.

10.4.1 Transmitter programming

Figure 10.4.1 shows a basic example of transmission operation. The hardware RTS/CTS flow control is used, the other parameters refer to:

- 8 bit data length
- 1 stop bit
- even parity
- MSB first
- software operates in polling mode on the TX FIFO

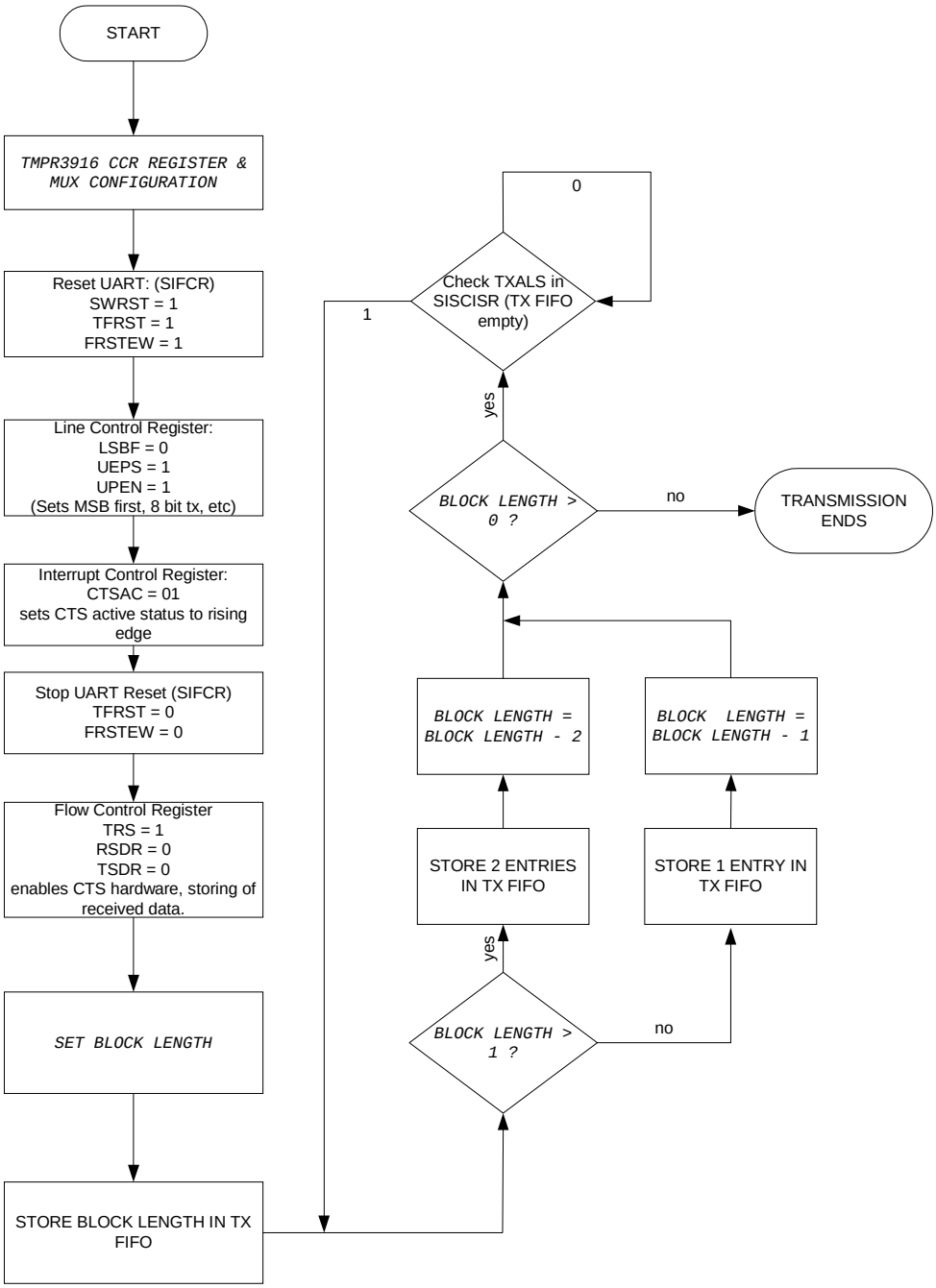


Figure 10.4.1 Example flow for a transmit operation. The operations written with the courier font are not to be performed on the UART, but may involve TMPR3916 programming or only software operations.

10.4.2 Receiver programming

In case of a reception operation, the flow of a simple driver could be like the one depicted in Figure 10.4.2. Here the parameters used to program the UART are:

- 8 bit data length
- 1 stop bit
- even parity
- MSB first
- RTS hardware flow control
- Software operates in polling mode

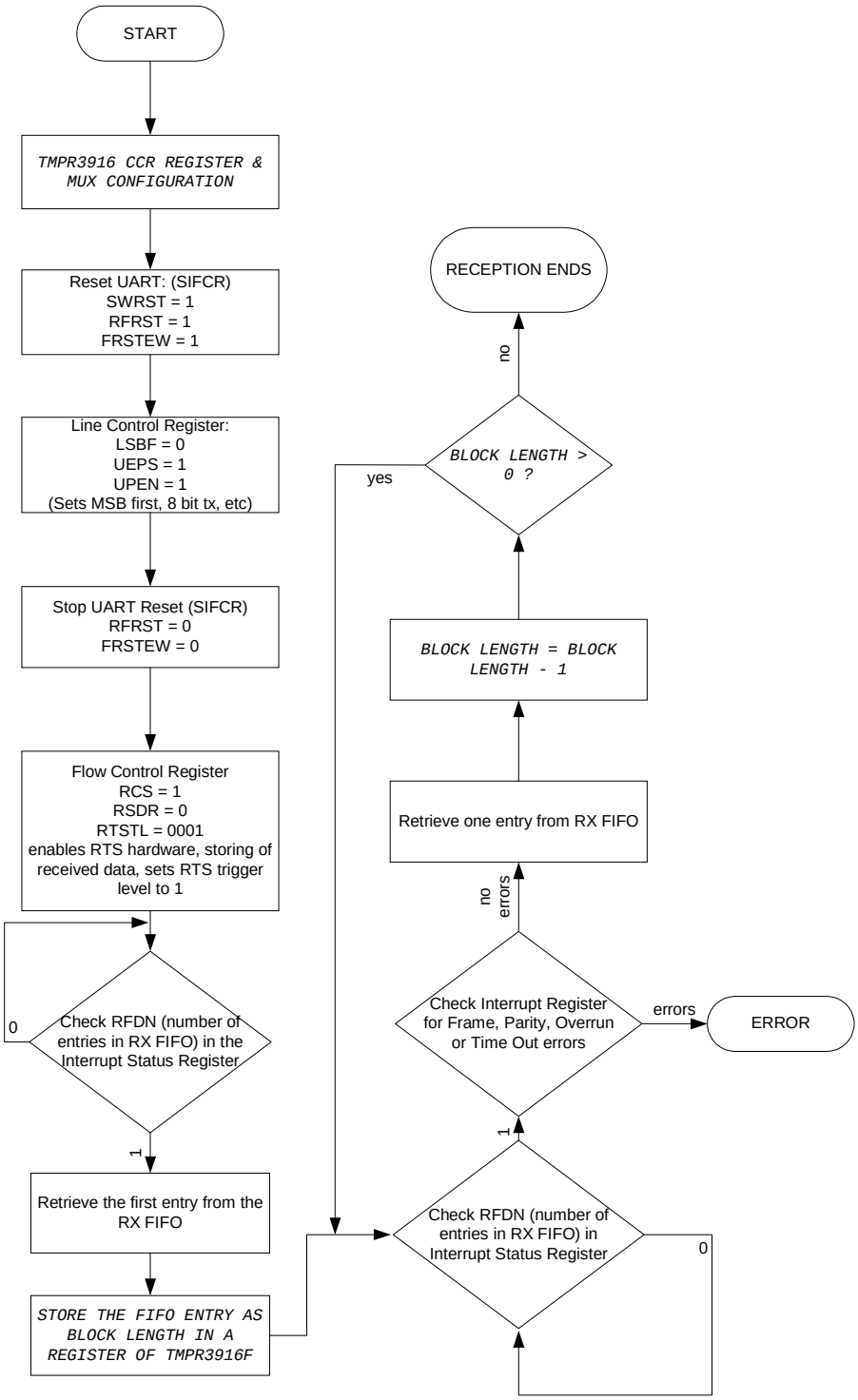


Figure 10.4.2 Example flow for a receive operation. The operations written with the courier font are not to be performed on the UART, but may involve TMPR3916 programming or only software operations.

11. Electrical Characteristics

11.1 DC Characteristics of TMPR3916

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{DD}	3.0	3.3	3.6	V
Operating Temperature	Ta	–40	—	85	°C
Storage Temperature	T_{STG}	–40	—	125	°C
Soldering Temperature (Time: 10 s)	T_{SOLDER}	—	—	260	°C
Power Dissipation (Normal Mode)	P_D	—	—	1200	mW
Operating Current Normal Mode	I_{DD}	150	—	333	mA
Standby Mode	I_{DDS}	—	—	500	µA
Low Level Input Voltage	V_{IL}	–0.3	—	0.2VDD	V
High Level Input Voltage	V_{IH}				
5 V Tolerant		0.7VDD	—	5.8	V
Others		0.7VDD	—	VDD+0.3	V
Low Level Input Current	I_{IL}		—		
Standard Input Buffer		–10	—	10	µA
Input Buffer with Pull-up (1)		–200	—	–10	µA
High Level Input Current	I_{IH}	–10	—	10	µA
Low Level Output Voltage	V_{OL}	—	—	0.4	V
High Level Output Voltage	V_{OH}	2.4	—	—	V
Output Current	I_{OL}				
4 mA Buffer (2)		4	—	—	mA
8 mA Buffer (3)		8	—	—	mA

(1) The following inputs have an integrated pull-up resistance:

VSYNC, HSYNC, DOTCLK, DREQ0, RESET, EXT0, EXT1, EXT2, NMI, dbge, dreset, BUSERR, ACK, RXCAN0, RXCAN1, PIO16, PIO18, PIO19, PIO21, PIO22, PIO23, PIO25, PIO26, PIO27, PIO29

(2) The following outputs are 4mA buffer outputs:

TXCAN1, TXCAN0, HDISP, VSYNC, HSYNC, DOTCLK, PIO29, PIO28, PIO27, PIO26, PIO25, PIO24, PIO23, PIO22, PIO21, PIO20, PIO19, PIO18, PIO17, PIO16, PIO15, PIO14, PIO13, PIO12, PIO11, PIO10, PIO9, PIO8, PIO7, PIO6, PIO5, PIO4, PIO3, PIO2, PIO1, PIO0

(3) The following outputs are 8mA buffer outputs:

A26, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, RD, WR, LAST, BSTART, BURST, CKE, WE, BE3, BE2, BE1, BE0, DREQ0, DACK0, SYSCLK, RAS, CAS, CS5, CS4, CS3, CS2, CS1, CS0, dclk, pcst2, pcst1, pcst0, sdao, D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25, D26, D27, D28, D29, D30, D31

11.2 Power Up Sequence

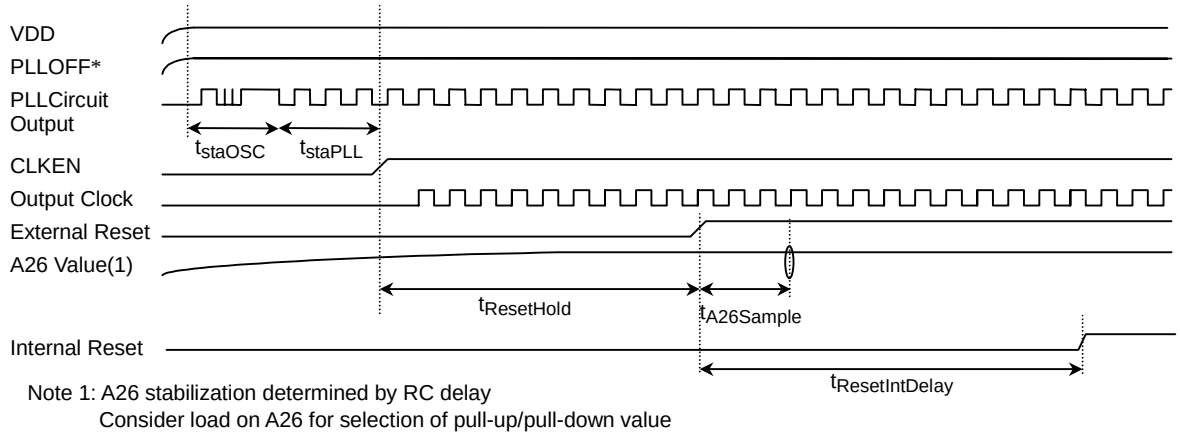


Figure 11.2.1 Waveform for TMPR3916's Power-up Sequence

Parameter	Symbol	Min	Typ	Max	Unit
Oscillator Starting Time(2)	t_{staOSC}	—	500	—	µs
PLL Starting Time	t_{staPLL}	—	—	500	µs
Reset Hold Time	$t_{ResetHold}$	25	—	—	cycles (1)
A26 Sample Time	$t_{A26Sample}$	—	—	4	cycles (1)
Initial Delay After Reset	$t_{ResetIntDelay}$	1000	—	1050	cycles (1)

(1) Cycle means one systemcycle of TMPR3916. (at 60 MHz one cycle is 16,7 ns)

(2) Determined by external oscillator.

11.3 Crystal Oscillator

An example of application circuit:

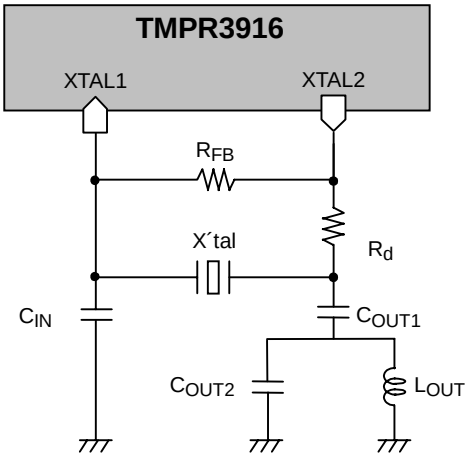


Figure 11.3.1 Connecting Crystal Oscillator

Parameter	Symbol	Min	Typ	Max	Unit
Oscillator Frequency (1)	f_{OSC}	6.5	7.5	8.5	MHz
System Clock Frequency	f_{SYS}	—	—	60.0	MHz
Oscillator Starting Time (2)	t_{staOSC}	—	500	—	µs
Phase-Locked-Loop Multiplier ($f_{SYS} = n * f_{OSC}$)	n	—	8	—	

(1) Restricted by the PLL input frequency capture range.

(2) Determined by external oscillator.

11.4 View DAC

DC Characteristics:

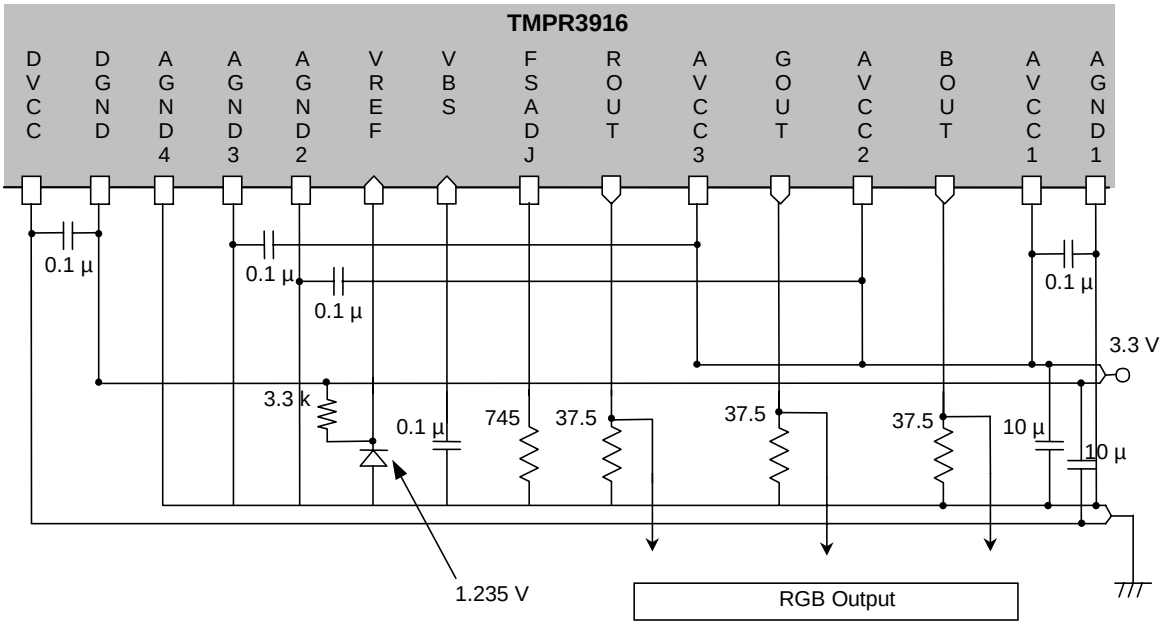
Parameter	Symbol	Min	Typ	Max	Unit
Resolution (Each DAC)		—	8 (1)	—	Bits
Accuracy (Each DAC)					
Integral Linearity Error	I_L	—	± ½	—	LSB
Differential Linearity Error	D_L	—	± ½	—	LSB
Analog Output Current					
White Level		—	19.05	—	mA
White Level from Black Level		—	17.61	—	mA
Black Level		—	1.44	—	mA
Blank Level		—	0.00	—	mA
LSB Size		—	69.06	—	µA
Voltage-reference Input Current		—	1.66	—	mA
Stand-by Current		—	0.00	—	mA

(1) Only upper six bits are used in TMPR3916, lower two bits are wired to zero

AC Characteristics:

Parameter	Symbol	Min	Typ	Max	Unit
Clock Rate	f_{max}	—	—	175	MHz
Clock Cycle Time	t_{ck}	5.72	—	—	ns
Analog Output Full Scale Delay	t_{OD}	—	1.8	—	ns
Full Scale Rise/Fall Time (10% to 90%)	t_{OR}	—	0.78	—	ns
Glitch Impulse		—	32.5	—	pV - sec
Power Supply Current	I_{AA}	—	—	70	mA

Connectivity of ViewDAC:



* To reduce the noise, please place Ceramic Capacitors of 0.1uF between DVCC/DGND, AVCC3/AGND3, AVCC2/AGND2 and AVCC1/AGND1 as close as possible to the pads

* Use seperate ground lines/planes for the digital and the analog ground in order to avoid analog ground level shift as a result of the current through the DGND terminal

Figure 11.4.1 Applying External Connectivity for Digital/Analog Converter

11.5 Standby Mode Timing

To resume the PLL circuit operations, set the PLLOFF* pin to “High” and the CLKEN pin to high. At that time, a period of 500 μs is required for the PLL circuit oscillation to stabilize. The following diagram shows the corresponding timing:

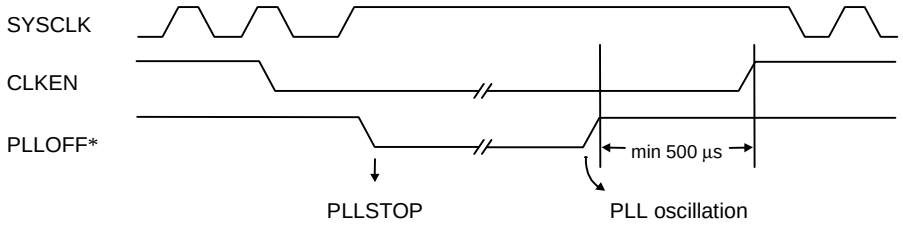


Figure 11.5.1 Standby Mode Waveform

11.6 Boot Device

By using the following application circuit, the user can choose between 16 bit and 32 bit boot device.

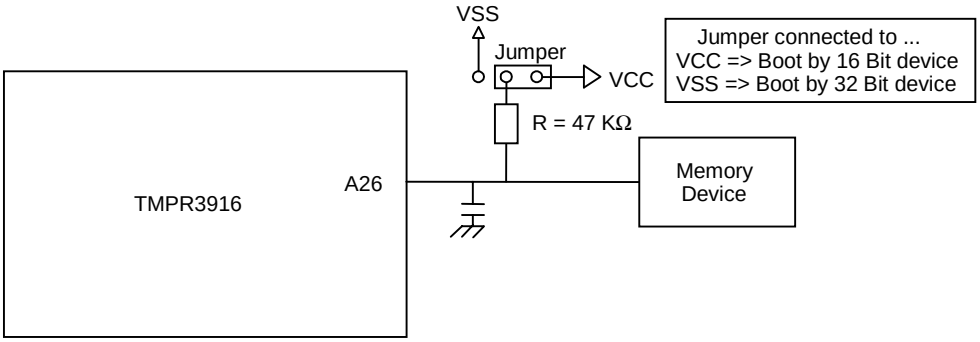


Figure 11.6.1 Applying External Boot Mode Selection Circuit

11.7 SDRAM Timing

Designing the SDRAM interface the following guidelines should be considered:

- SDRAM signals should be routed using a comparable wire-length for each interface signal
- capacitive load mismatches between different interface signals should be avoided

For the following electrical specification of the TMPR3916F external bus interface, balanced loads and equal wiring lengths of the interface signals are assumed.

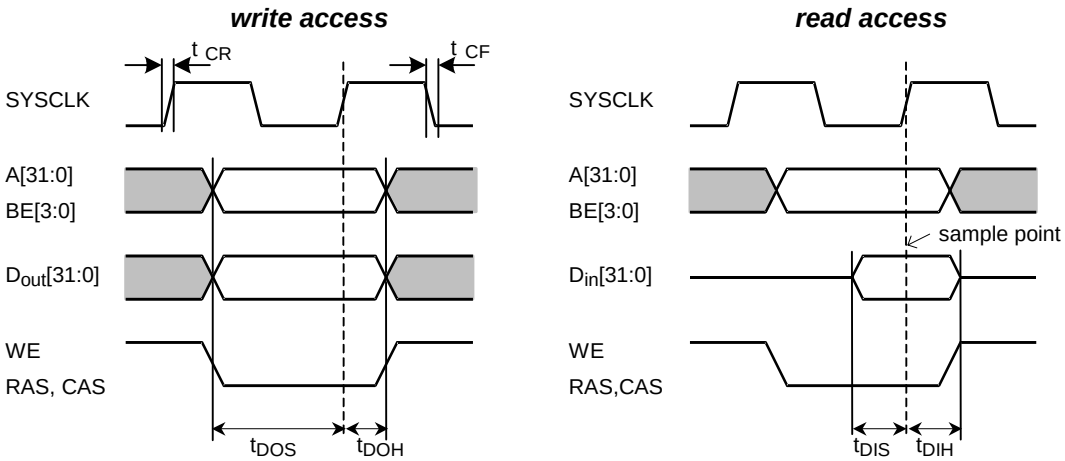


Figure 11.7.1 SDRAM Interface Timing Diagram

Parameter	Symbol	Min	Typ	Max	Unit
Output Setup Time	t_{DOS}	4.5	—	—	ns
Output Hold Time	t_{DOH}	2.5	—	—	ns
Input Setup Time	t_{DIS}	3,5	—	—	ns
Input Hold Time	t_{DIH}	1.5	—	—	ns
Clock Rise Time	t_{CR}	—	—	4.0	ns
Clock Fall Time	t_{CF}	—	—	4.0	ns
Load on Interface Signals	C_{IF}	—	—	30	pF

11.8 ROM / SRAM Timing

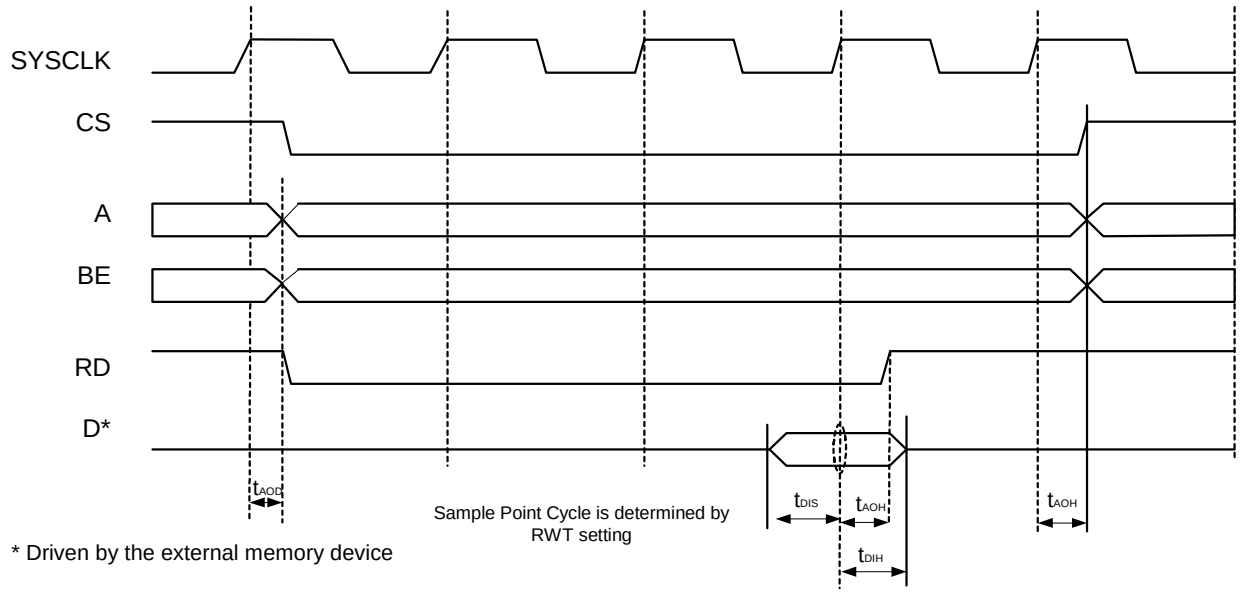


Figure 11.8.1 ROM/Flash/SRAM Interface Timing Diagram (Read)

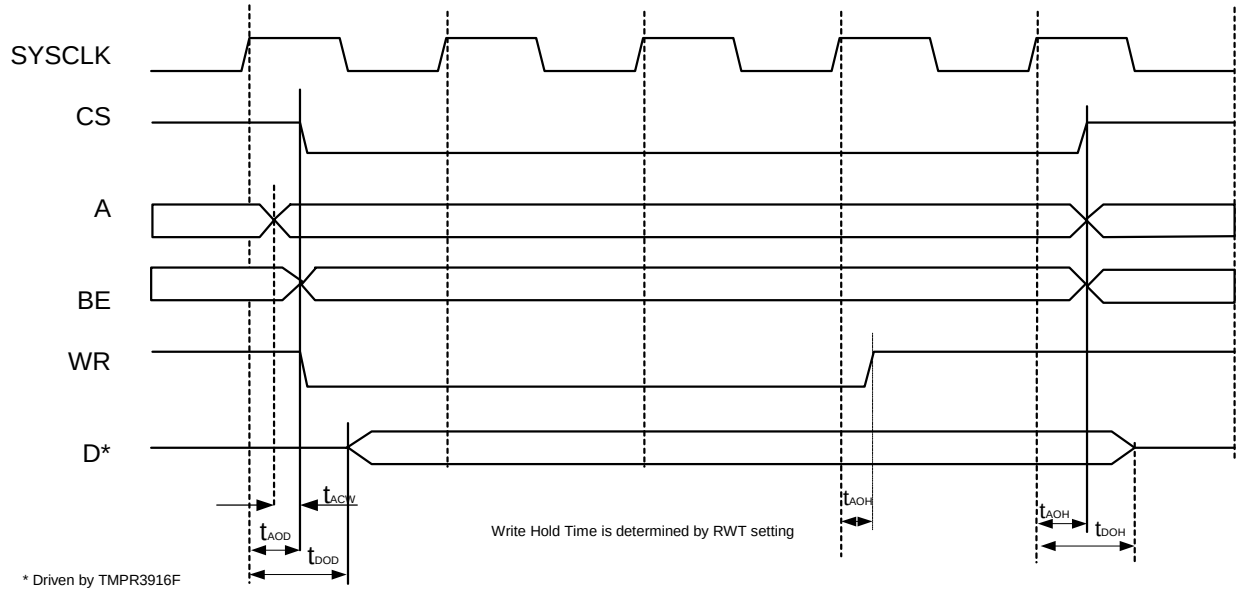


Figure 11.8.2 ROM/Flash/SRAM Interface Timing Diagram (Write)

Parameter	Symbol	Min	Typ	Max	Unit
Address, Control Output Drive Time	t_{AOD}	—	—	12	ns
Data Output Drive Time	t_{DOD}	—	—	12,5	ns
Address, Control Output Hold Time	t_{AOH}	2,0	—	—	ns
Data Output Hold Time	t_{DOH}	2,0	—	—	ns
Address to CS or WR[1]	t_{ACW}	0	—	—	ns
Data Input Setup Time	t_{DIS}	3	—	—	ns
Data Input Hold Time	t_{DIH}	2	—	—	ns
Load on Interface Signals	C_{IF}	—	—	30	pF

[1] Assumes balanced loads on WR, CS and Address pins.

11.9 External Slave

Write timing:

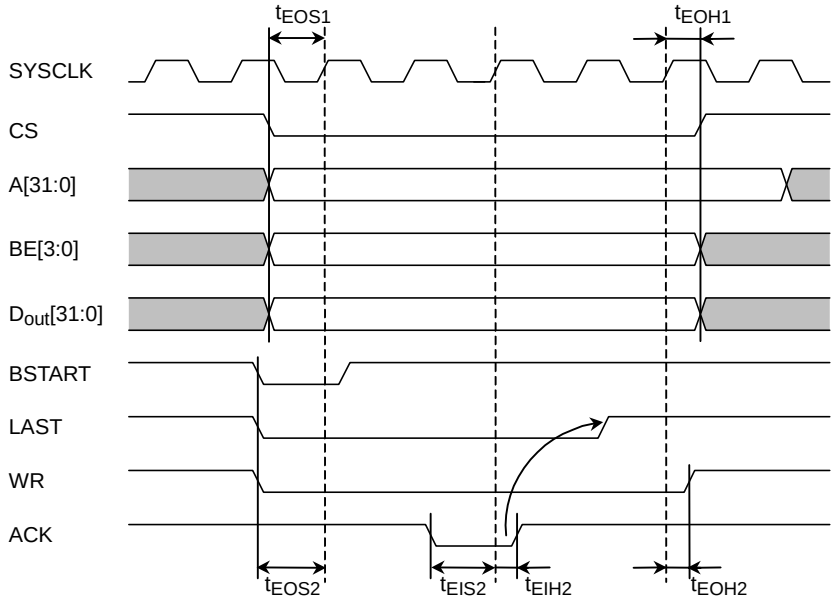


Figure 11.9.1 Timing Diagram for Write Access to External Slave Initiated by TMPR3916

Parameter	Symbol	Min	Typ	Max	Unit
Output Setup Time of CS, A, BE, D	t_{EOS1}	4.0	—	—	ns
Output Hold Time of CS, A, BE, D	t_{EOH1}	2.0	—	—	ns
Output Setup Time of BSTART, LAST, WR, DACK	t_{EOS2}	4.0	—	—	ns
Output Hold Time of BSTART, LAST, WR, DACK	t_{EOH2}	2.0	—	—	ns
Input Setup Time of ACK	t_{EIS2}	6.0	—	—	ns
Input Hold Time of ACK	t_{EIH2}	2.0	—	—	ns
Load on Interface Signals	C_{IF}	—	—	30	pF

Read timing:

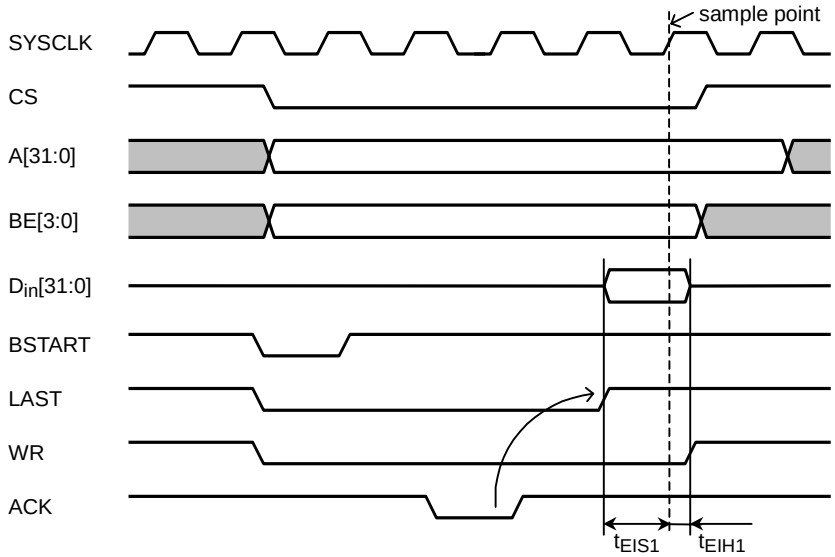


Figure 11.9.2 Timing Diagram for Read Access to External Slave Initiated by TMPR3916

Parameter	Symbol	Min	Typ	Max	Unit
Input Setup Time of D[31:0]	t_{EIS1}	4.0	—	—	ns
Input Hold Time of D[31:0]	t_{EIH1}	2.0	—	—	ns
Load on Interface Signals	C_{IF}	—	—	30	pF

DMA Timing:

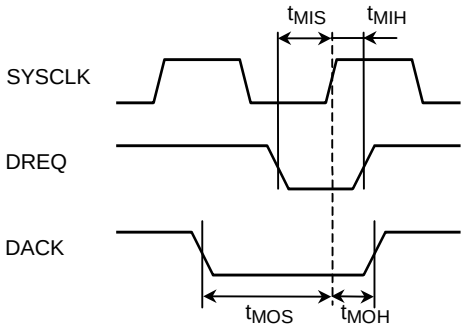


Figure 11.9.3 Timing Diagram for DMA Request

Parameter	Symbol	Min	Typ	Max	Unit
Output Setup Time of DACK	t_{MOS}	5.0	—	—	ns
Output Hold Time of DACK	t_{MOH}	2.0	—	—	ns
Input Setup Time of DREQ	t_{MIS}	6.0	—	—	ns
Input Hold Time of DREQ	t_{MIH}	2.0	—	—	ns
Load on Interface Signals	C_{IF}	—	—	30	pF

11.10 External Interrupts and NMI

The external interrupts (EXT0, EXT1, EXT2) and the NMI can be applied asynchronous to the system clock. All interrupt inputs are low active.

Parameter	Symbol	Min	Typ	Max	Unit
Length of Asynchronous Interrupt	t_{IRQL}	2	—	—	cycles (1)

(1) Cycle means one system cycle of TMPR3916 (at 60 MHz one cycle is 16.67 ns)

11.11 General Purpose I/O (PORT Module)

The inputs to the general purpose I/O´s can be applied asynchronous to the system clock.

Parameter	Symbol	Min	Typ	Max	Unit
Length of Input Signal	t_{GPL}	10	—	—	cycles (1)
Sample Frequency	f_{GPS}	—	7.5 (2)	—	MHz

(1) Cycle means one system cycle of TMPR3916 (at 60 MHz one cycle is 16.67 ns)

(2) at 60 MHz system clock

11.12 TXSEI Timing

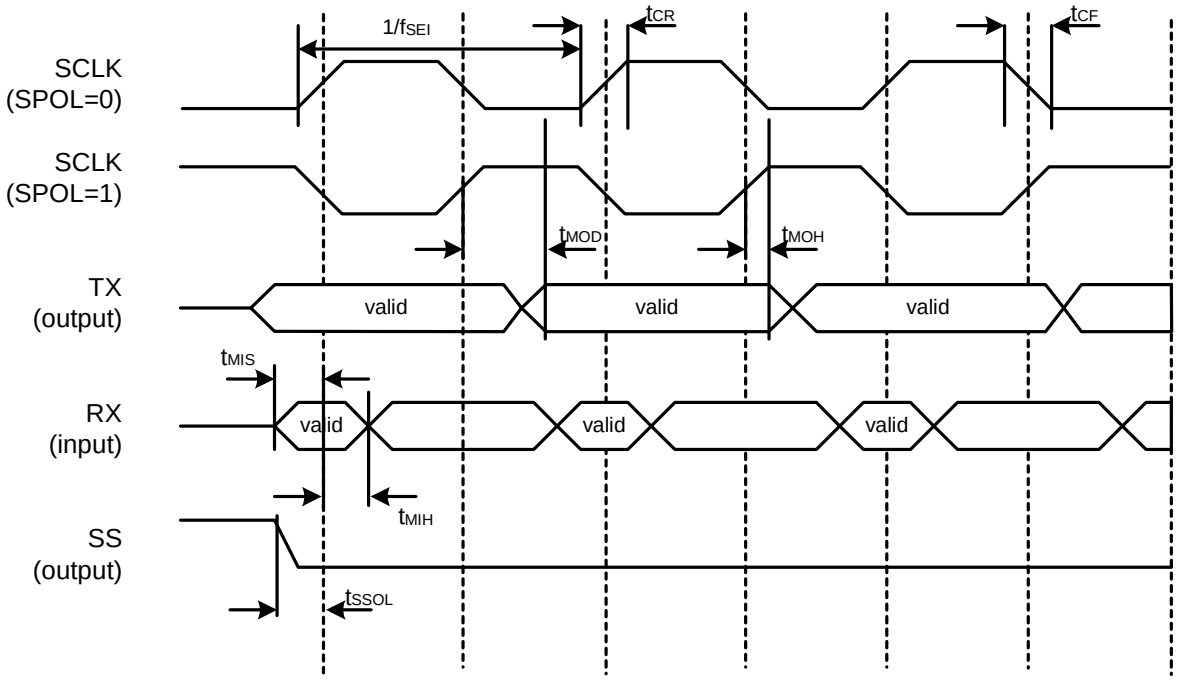


Figure 11.12.1 Timing Diagram for TXSEI Timing (Master, SPHASE = 0)

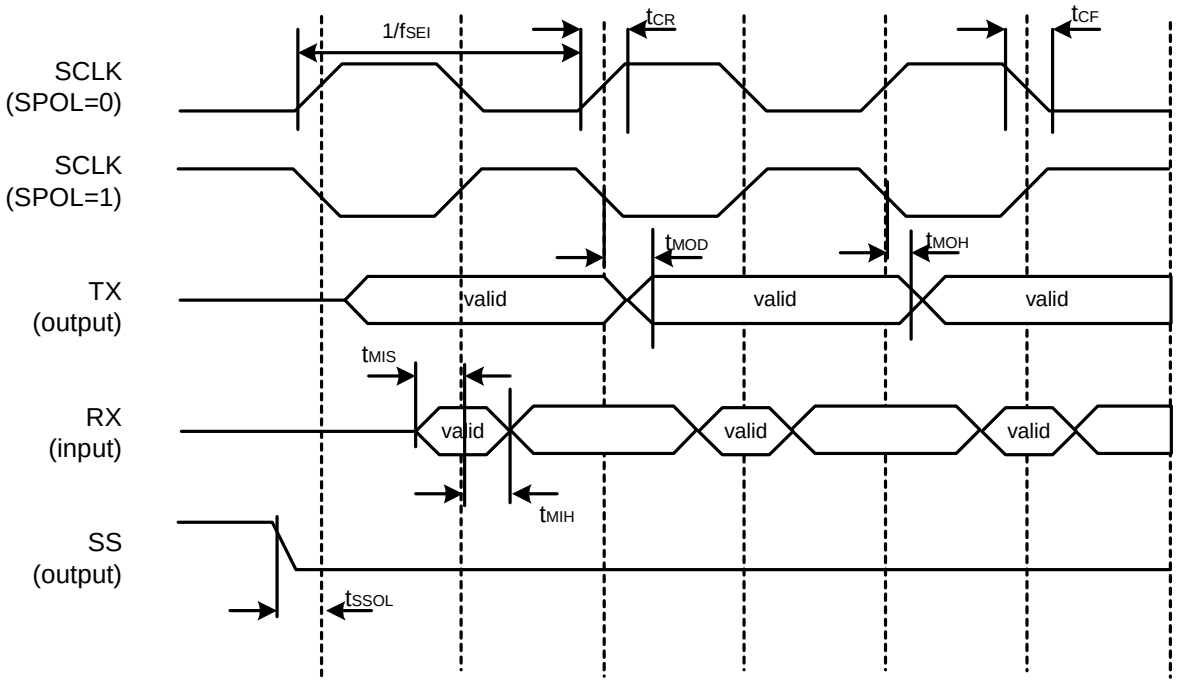


Figure 11.12.2 Timing Diagram for TXSEI Timing (Master, SPHASE = 1)

Parameter	Symbol	Min	Typ	Max	Unit
SEI Clock Frequency	f_{SEI}	—	—	15 (1)	MHz
SEI Clock Rise Time	t_{CR}	—	—	12	ns
SEI Clock Fall Time	t_{CF}	—	—	12	ns
Master Data in Setup Time	t_{MIS}	20	—	—	ns
Master Data in Hold Time	t_{MIH}	5	—	—	ns
Master Data Out Drive Time	t_{MOD}	—	—	20	ns
Master Data Out Hold Time	t_{MOH}	0	—	—	ns
Slave Select Output Lead Time	t_{SSOL}	1	—	—	cyc(2)
Load on SEI Interface Signals	C_{IF}	—	—	50	pF

(1) At 60 MHz system clock. In general: ¼ of the system clock.

(2) System cycles, means 16.67 ns at 60 MHz clock frequency.

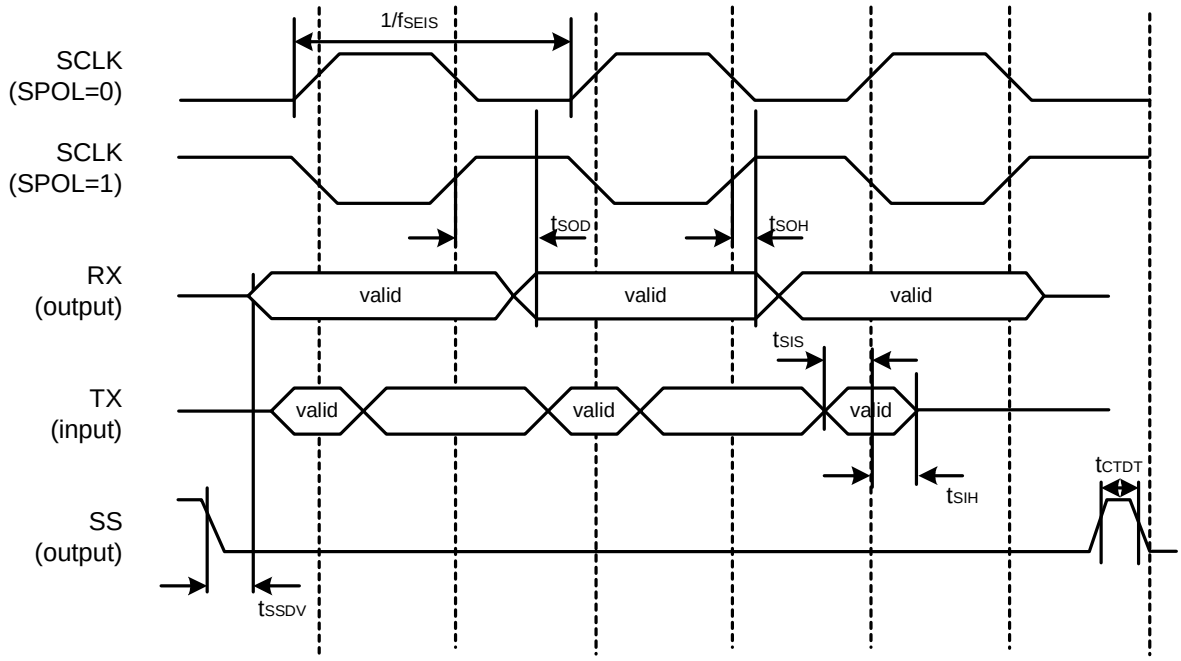


Figure 11.12.3 Timing Diagram for TXSEI Timing (Slave, SPHASE = 0)

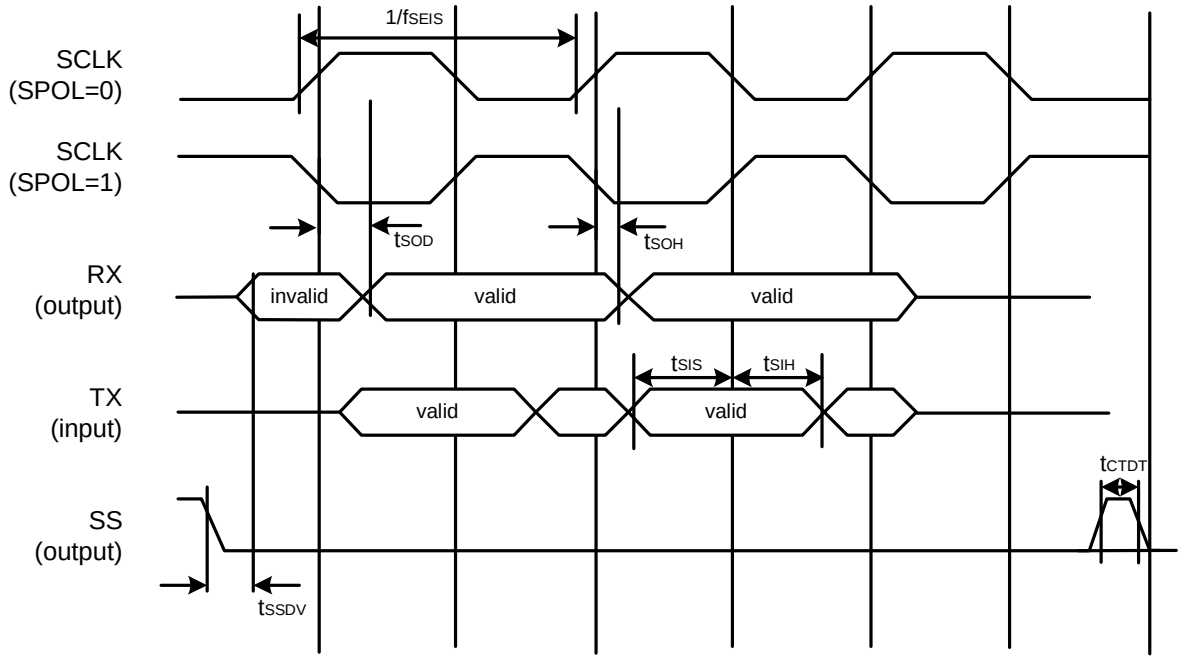


Figure 11.12.4 Timing Diagram for TXSEI Timing (Slave, SPHASE = 1)

Parameter	Symbol	Min	Typ	Max	Unit
Slave Mode SEI Clock Frequency	f_{SEIS}	—	—	7.25 (1)	MHz
Clock to Data Valid Time	t_{SOD}	—	—	20	ns
Clock to Data Invalid Time	t_{SOH}	0	—	—	ns
Slave Data in Setup Time	t_{SIS}	20	—	—	ns
Slave Data in Hold Time	t_{SIH}	5	—	—	ns
Slave Select to Data Valid Time	t_{SSDV}	—	—	50	ns
Slave Select to Clock	t_{SSTC}	$3/f_{SYS}$	—	—	ns
Consecutive Transfer Delay Time(2)	t_{CTDT}	$1/f_{SEIS}$	—	—	ns
Load on SEI Interface Signals	C_{IF}	—	—	50	pF

(1) At 60 MHz system clock. In general: 1/8 of the system clock frequency.

(2) f_{SYS} refers to the TMPR3916 system clock (usually 60 MHz).

(3) For SPHASE = 1 the slave does not require to be deselected between consecutive transfers.

12. Package

12.1 Pin Assignment

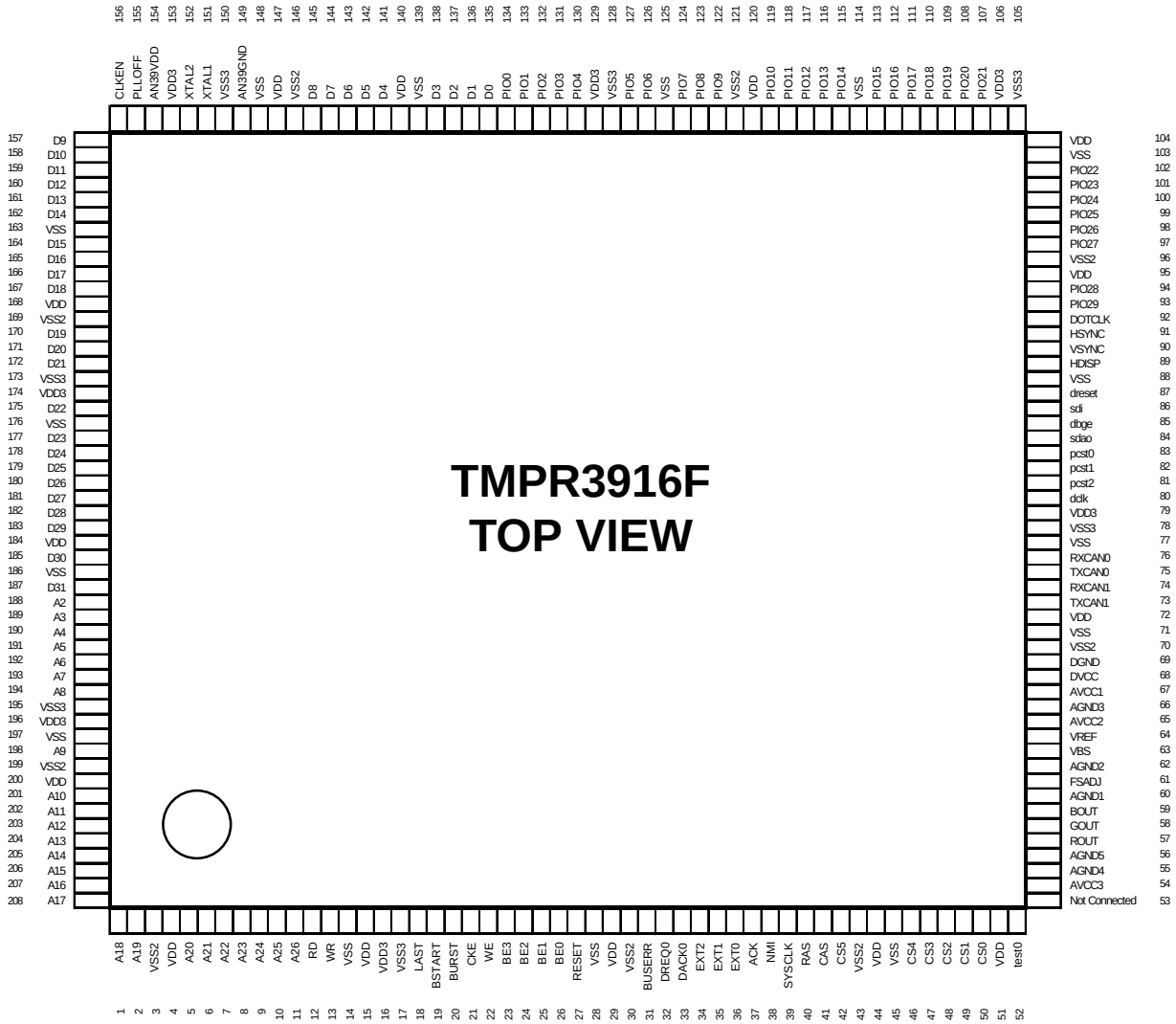


Figure 12.1.1 TMPR3916F's Pin Assignment

The following table divides the different pins into functional groups. TMPR3916F provides pins, which have a shared functionality. Therefore you can find one and the same pin up to three times in different groups (like PIO18/TX(SEI)/TX(SIO0) pin).

Classification	Pin Name	Pin No.	I/O	Level Active
CORE	A[26] / BOOT16	11	I/O	—
	A[25:20]	10 ~ 5	O	—
	A[19:18]	2 ~ 1	O	—
	A[17:10]	208 ~ 201	O	—
	A[9]	198	O	—
	A[8:2]	194 ~ 188	O	—
	BE[3:0]*	23 ~ 26	O	low
	D[31]	187	I/O	—
	D[30]	185	I/O	—
	D[29:23]	183 ~ 177	I/O	—
	D[22]	175	I/O	—
	D[21:19]	172 ~170	I/O	—
	D[18:15]	167 ~ 164	I/O	—
	D[14:9]	162 ~ 157	I/O	—
	D[8:4]	145 ~ 141	I/O	—
	D[3:0]	138 ~ 135	I/O	—
	Control Signals			
	RD*	12	O	low
	WR*	13	O	low
	LAST*	18	O	low
	BSTART*	19	O	low
	BURST*	20	O	low
	BUSERR*	31	I	low
	ACK*	37	I	low
	RESET*	27	I	low
	Clock Signals			
	XTAL1	151	I	—
	XTAL2	152	O	—
	SYSCLK	39	O	—
	PLLOFF*	155	I	low
	CLKEN	156	I	high
SDRAM	CS1*	49	O	low
	CS0*	50	O	low
	RAS*	40	O	low
	CAS*	41	O	low
	CKE	21	O	high
	WE*	22	O	low
MEMC	CS5*	42	O	low
	CS4*	46	O	low
	CS3*	47	O	low
	CS2*	48	O	low
GDC	HSYNC*	91	I/O	low
	VSYNC*/CSYNC*	90	I/O	low
	HDISP	89	O	high
	DOTCLK	92	I/O	—
	PIO0/Digital B Out[0]	134	I/O	—
	PIO1/Digital B Out[1]	133	I/O	—
	PIO2/Digital B Out[2]	132	I/O	—
	PIO3/Digital B Out[3]	131	I/O	—
	PIO4/Digital B Out[4]	130	I/O	—

Classification	Pin Name	Pin No.	I/O	Level Active
GDC	PIO5/Digital B Out[5]	127	I/O	—
	PIO6/Digital G Out[1]	126	I/O	—
	PIO7/Digital G Out[2]	124	I/O	—
	PIO8/Digital G Out[3]	123	I/O	—
	PIO9/Digital G Out[4]	122	I/O	—
	PIO10/Digital G Out[5]	119	I/O	—
	PIO11/Digital R Out[1]	118	I/O	—
	PIO12/Digital R Out[2]	117	I/O	—
	PIO13/Digital R Out[3]	116	I/O	—
	PIO14/Digital R Out[4]	115	I/O	—
	PIO15/Digital R Out[5]	113	I/O	—
	ROUT	57	A. O	—
	GOUT	58	A. O	—
	BOUT	59	A. O	—
	VBS	63		—
	VREF	64		—
	FSADJ	61		—
DMAC	DREQ0*	32	I	low
	DACK0*	33	O	low
TXCAN	TX(CAN1)	73	O	—
	RX(CAN1)	74	I	—
	TX(CAN0)	75	O	—
	RX(CAN0)	76	I	—
TXSEI	PIO18/TX(SEI)/TX(SIO0)	110	I/O	—
	PIO17/RX(SEI)/RX(SIO0)	111	I/O	—
	PIO16/CLK(SEI)/CLK(SIO0)	112	I/O	—
	PIO19/SSI*(SEI)/CLK(SIO1)	109	I/O	low
	PIO22/SSO*(SEI)/RTS*(SIO1)	102	I/O	low
UART	PIO29/TX(SIO3)	93	I/O	—
	PIO28/RX(SIO3)	94	I/O	—
	PIO27/CTS*(SIO2)	97	I/O	low
	PIO26/RTS*(SIO2)	98	I/O	low
	PIO25/TX(SIO2)	99	I/O	—
	PIO24/RX(SIO2)	100	I/O	—
	PIO23/CTS*(SIO1)	101	I/O	low
	PIO22/SSO*(SEI)/RTS*(SIO1)	102	I/O	low
	PIO21/TX(SIO1)	107	I/O	—
	PIO20/RX(SIO1)	108	I/O	—
	PIO19/SSI*(SEI)/CLK(SIO1)	109	I/O	—
	PIO18/TX(SEI)/TX(SIO0)	110	I/O	—
	PIO17/RX(SEI)/RX(SIO0)	111	I/O	—
	PIO16/CLK(SEI)/CLK(SIO0)	112	I/O	—
PORT	PIO29/TX(SIO3)	93	I/O	—
	PIO28/RX(SIO3)	94	I/O	—
	PIO27/CTS*(SIO2)	97	I/O	—
	PIO26/RTS*(SIO2)	98	I/O	—
	PIO25/TX(SIO2)	99	I/O	—
	PIO24/RX(SIO2)	100	I/O	—
	PIO23/CTS*(SIO1)	101	I/O	—
	PIO22/SSO*(SEI)/RTS*(SIO1)	102	I/O	—
	PIO21/TX(SIO1)	107	I/O	—
	PIO20/RX(SIO1)	108	I/O	—
	PIO19/SSI*(SEI)/CLK(SIO1)	109	I/O	—
	PIO18/TX(SEI)/TX(SIO0)	110	I/O	—

Classification	Pin Name	Pin No.	I/O	Level Active
PORT	PIO17/RX(SEI)/RX(SIO0)	111	I/O	—
	PIO16/CLK(SEI)/CLK(SIO0)	112	I/O	—
	PIO15/Digital R Out [5]	113	I/O	—
	PIO14/Digital R Out [4]	115	I/O	—
	PIO13/Digital R Out [3]	116	I/O	—
	PIO12/Digital R Out [2]	117	I/O	—
	PIO11/Digital R Out [1]	118	I/O	—
	PIO10/Digital G Out [5]	119	I/O	—
	PIO9/Digital G Out [4]	122	I/O	—
	PIO8/Digital G Out [3]	123	I/O	—
	PIO7/Digital G Out [2]	124	I/O	—
	PIO6/Digital G Out [1]	126	I/O	—
	PIO5/Digital B Out [5]	127	I/O	—
	PIO4/Digital B Out [4]	130	I/O	—
	PIO3/Digital B Out [3]	131	I/O	—
	PIO2/Digital B Out [2]	132	I/O	—
	PIO1/Digital B Out [1]	133	I/O	—
	PIO0/ Digital B Out [0]	134	I/O	—
INTC	EXT2*	34	I	low
	EXT1*	35	I	low
	EXT0*	36	I	low
	NMI*	38	I	low
POWER	VDD	4, 15, 29, 44, 51, 72, 95, 104, 120, 140, 147, 168, 184, 200		—
	VDD3	16, 79, 106, 129, 153, 174, 196		—
	VSS	14, 28, 45, 71, 77, 88, 103, 114, 125, 139, 148, 163, 176, 186, 197		—
	VSS2	3, 30, 43, 70, 96, 121, 146, 169, 199		—
	VSS3	17, 78, 105,128, 150, 173, 195		—
	AVCC1	67		
	AVCC2	65		
	AVCC3	54		
	DVCC	68		
	DGND	69		
	AN39VDD	154		—
	AN39GND	149		—
	AGND1	60		—
	AGND2	62		—
	AGND3	66		—
	AGND4	55		—
	AGND5	56		—
DSU	dclk	80	O	—
	pcst2	81	O	—
	pcst1	82	O	—
	pcst0	83	O	—
	sdao/tpc	84	O	—
	dbge*	85	I	low
	sdi/dint*	86	I	low
	dreset*	87	I	low
TEST	test0	52	I	high
Not Connected	N/C(1)	53		

(1) Recommendation: Connect unconnected pins to ground.

12.2 Pin Functions

Classification	Pin Name	Pin Function
CORE	A[26:2]	Address signal output pins. The A26 pin has a special functionality. The level supplied to this pin is latched with the rising edge of the RESET* signal. The level determines whether to boot from a device with 16-bit or 32-bit width. Supplying “high” lets the TMPR3916F boot from a 16-bit device.
	BE[3:0]*	Byte Enable output pins The byte enable signals are used to select the bytes within the word, which are accessed by the current write or read access. The following list shows the relationship between byte enable signals and the valid bytes on the data bus. BE*[0] low => D[7:0] valid BE*[1] low => D[15:8] valid BE*[2] low => D[23:16] valid BE*[3] low => D[31:24] valid
	D[31:0]	Data input/output pins
	RD*	The RD* signal is asserted during a read access to the external bus interface.
	WR*	The WR* signal is asserted during a write access to the external bus interface.
	LAST*	The LAST* signal is asserted if the final data of the current bus operation is read or written.
	BSTART*	The BSTART* signal is asserted for one cycle at the beginning of an external bus interface access.
	BURST*	The BURST* signal indicates that the current access is a burst access.
	BUSERR*	Bus Error input pin. If an error occurs during the current transaction the external device has the opportunity to signal this event to the TMPR3916F by asserting the BUSERR* signal. Thereupon the TMPR3916F will finish the access and will create a bus-error exception.
	ACK*	Acknowledge signal input pin. During a read access the external device acknowledges data-transfers of a transaction by asserting the ACK* signal. The data on the external bus interface will be sampled on the rising edge of system clock. During a write access the external device signals the TMPR3916F that the data was captured by asserting the ACK* signal. The TMPR3916F will complete the transaction.
	RESET*	If a low level is applied to the RESET* signal the chip will go into the reset state.
Clock Generator	XTAL1	Input pin for the crystal.
	XTAL2	Feedback output pin for the crystal.
	SYSCLK	Output of system clock, which is the reference clock for bus operation.
	PLLOFF*	Master clock switching pin. Inputting a “High” signal to this pin uses the built-in PLL circuit as the master clock. Master clock frequency is eight times the external clock. Inputting a low signal to this pin halts the built-in PLL circuit oscillation and uses the external clock as master clock.
	CLKEN	The clock enable pin enables supply of crystal input to internal PLL. This signal is high active.
SDRAM	CS1*	Chip select 1* signal for the external SDRAM device.
	CS0*	Chip select 0* signal for the external SDRAM device.
	RAS* CAS* WE*	The three signals row access strobe (RAS*), column access strobe (CAS*) and write enable (WE*) are used to supply the SDRAM with commands.
	CKE	The clock enable pin is an output to SDRAM used for power saving purposes.

Classification	Pin Name	Pin Function
MC	CS5*	The chip select signal 5 will be asserted if an access to the address range specified in the RCCR5 register will take place.
	CS4*	The chip select signal 4 will be asserted if an access to the address range specified in the RCCR4 register will take place.
	CS3*	The chip select signal 3 will be asserted if an access to the address range specified in the RCCR3 register will take place.
	CS2*	The chip select signal 2 will be asserted if an access to the address range specified in the RCCR2 register will take place.
GDC	HSYNC*	Horizontal sync signal input/output pin.
	VSYNC*/CSYNC*	Vertical sync signal input/output pin or the composite sync signal output pin. The composite sync signal is the logic EX-NOR (exclusive nor) operation on signals HSYNC* and VSYNC*
	HDISP	While the viewable area of the current line is output by the GDC the HDISP is set to logic one. Data can be read with one cycle latency to this signal.
	DOTCLK	Dot clock input/output pin. The dot clock is the reference clock for the display. This clock is either input to or output by the TMPR3916F.
	PIO15/Digital R Out[5] PIO14/Digital R Out[4] PIO13/Digital R Out[3] PIO12/Digital R Out[2] PIO11/Digital R Out[1]	The general purpose input/output signals PIO11 to PIO15 (5bit) can be switched in that way that the red intensity of the current pixel is output. PIO15/Digital R Out[5] is the MSB.
	PIO10/Digital G Out[5] PIO9/Digital G Out[4] PIO8/Digital G Out[3] PIO7/Digital G Out[2] PIO6/Digital G Out[1]	The general purpose input/output signals PIO6 to PIO10 (5 bit) can be switched in that way that the green intensity of the current pixel is output. PIO10/Digital G Out[5] is the MSB.
	PIO5/Digital B Out[5] PIO4/Digital B Out[4] PIO3/Digital B Out[3] PIO2/Digital B Out[2] PIO1/Digital B Out[1] PIO0/Digital B Out[0]	The general purpose input/output signals PIO0 to PIO5 (6 bit) can be switched in that way that the blue intensity of the current pixel is output. PIO5/Digital B Out[5] is the MSB.
	ROUT GOUT BOUT	Output pins for the three primary color video (analog) signals used as color source for display. ROUT is the red video signal output pin, GOUT the green and BOUT the blue. All these signal are analog signals output by the VIEWDAC.
	VBS	This terminal is used for noise rejection of DAC's current adjustment bias. It is recommended to connect a capacitance of 0.1*F to the ground.
	VREF	External voltage reference-bias input for the digital-to-analog converter.
	FSADJ	Current-mirror output. This pin is used to set the current level in the DAC outputs via an internal current-mirror. This pin is usually connected to ground via a 745 Ohm resistor.
DMAC	DREQ0	External DMA request signal.
	DACK0	DMA acknowledge signal output pin.
TXCAN	TX(CAN1)	Transmit pin of CAN channel 1
	RX(CAN1)	Receive pin of CAN channel 1
	TX(CAN0)	Transmit pin of CAN channel 0
	RX(CAN0)	Receive pin of CAN channel 0
TXSEI	PIO18/TX(SEI)/TX(SIO0)	In master mode this pin is the data output of the SEI interface. In slave mode this is the data input pin of the SEI interface.
	PIO17/RX(SEI)/RX(SIO0)	In master mode this pin is the data input pin of the TXSEI device. In slave mode this pin is the data output.
	PIO16/CLK(SEI)/CLK(SIO)	In master mode the clock is output during transmission from the TXSEI module. In slave mode the clock is received from the device the TMPR3916F is communicating with.

Classification	Pin Name	Pin Function
TXSEI	PIO19/SSI*(SEI)/CLK(SIO1)	When the TXSEI module is configured as a slave the SSI* (slave-select-input) signal shows that the TXSEI module is accessed in the current transfer. In master mode this pin can be used to check the bus for a second master on the bus. By definition more than one master is not allowed because such a configuration might damage the circuits!
	PIO22/SSO*(SEI)/RTS*(SIO1)	During master mode the SSO* (slave-select output) is used to enable the outputs of an SEI device connected to the TMPR3916F.
UART	PIO29/TX(SIO3) PIO25/TX(SIO2) PIO21/TX(SIO1) PIO18/TX(SEI)/TX(SIO0)	Serial data transmit (output) pin.
	PIO28/RX(SIO3) PIO24/RX(SIO2) PIO20/RX(SIO1) PIO17/RX(SEI)/RX(SIO0)	Serial data receive (input) pin.
	PIO26/RTS*(SIO2) PIO22/SSO*(SEI)/RTS*(SIO1)	Request to send signal output pin.
	PIO27/CTS*(SIO2) PIO23/CTS*(SIO1)	Clear to send signal output pin.
	PIO19/SSI*(SEI)/CLK(SIO1) PIO16/CLK(SEI)/CLK(SIO0)	UART clock output for synchronous transfer mode
PORT	PIO[29:0]	30-bit parallel I/O port pins.
INTC	EXT[2:0]*	Interrupt request signal input pins.
	NMI*	Non-maskable interrupt signal input pin. If this signal is asserted the TX39 core jumps to the non-maskable-interrupt service routine.
DSU[1]	dclk	Debug clock This pin outputs a clock for a real time debug system.
	pcst[2:0]	PC trace status Outputs PC trace status information and the mode of the serial monitor bus.
	sdao/tpc	Serial data and address Output / target PC
	dbge*	Debugger enable The external real time debug system signals to the DSU by asserting this pin, that it is connected.
	sdi/dint*	Serial data input / debug interrupt When DSU mode is not used, this pin must be tied to High.
	dreset*	Debug reset A reset input for a real-time debug system. When dreset* is asserted, the debug support unit (DSU) is initialized.
TEST	test0	This pin is used for manufacturing test purposes. For regular operation this pin must be tied to zero. Otherwise the TMPR3916F and connected devices may be damaged.

[1] If the DSU interface is not utilized, tie sdi/dint* to VCC using a 47 KΩ resistor; leave dreset* and dgbe* pins open.

Appendix A. Register Overview of TMPR3916

Classification	Address	*Register Name*	Function
UART	1C00 0000H	*SILCR0*	SIO Control Register (CH0)
	1C00 0004H	*SIDICR0*	SIO Interrupt Control Register (CH0)
	1C00 0008H	*SIDISR0*	SIO Interrupt Status Register (CH0)
	1C00 000CH	*SISCISR0*	SIO Status Change Register (CH0)
	1C00 0010H	*SIFCR0*	SIO FIFO Control Register (CH0)
	1C00 0014H	*SIFLCR0*	SIO Flow Control Register (CH0)
	1C00 0018H	*SIBGR0*	SIO Baud Rate Control Register (CH0)
	1C00 001CH	*SITFIFO0*	SIO Transmit FIFO Register (CH0)
	1C00 0020H	*SIRFIFO0*	SIO Receive FIFO Register (CH0)
	1C00 0040H	*SILCR1*	SIO Control Register (CH1)
	1C00 0044H	*SIDICR1*	SIO Interrupt Control Register (CH1)
	1C00 0048H	*SIDISR1*	SIO Interrupt Status Register (CH1)
	1C00 004CH	*SISCISR1*	SIO Status Change Register (CH1)
	1C00 0050H	*SIFCR1*	SIO FIFO Control Register (CH1)
	1C00 0054H	*SIFLCR1*	SIO Flow Control Register (CH1)
	1C00 0058H	*SIBGR1*	SIO Baud Rate Control Register (CH1)
	1C00 005CH	*SITFIFO1*	SIO Transmit FIFO Register (CH1)
	1C00 0060H	*SIRFIFO1*	SIO Receive FIFO Register (CH1)
	1C00 0080H	*SILCR2*	SIO Control Register (CH2)
	1C00 0084H	*SIDICR2*	SIO Interrupt Control Register (CH2)
	1C00 0088H	*SIDISR2*	SIO Interrupt Status Register (CH2)
	1C00 008CH	*SISCISR2*	SIO Status Change Register (CH2)
	1C00 0090H	*SIFCR2*	SIO FIFO Control Register (CH2)
	1C00 0094H	*SIFLCR2*	SIO Flow Control Register (CH2)
	1C00 0098H	*SIBGR2*	SIO Baud Rate Control Register (CH2)
	1C00 009CH	*SITFIFO2*	SIO Transmit FIFO Register (CH2)
	1C00 00A0H	*SIRFIFO2*	SIO Receive FIFO Register (CH2)
	1C00 00C0H	*SILCR3*	SIO Control Register (CH3)
	1C00 00C4H	*SIDICR3*	SIO Interrupt Control Register (CH3)
	1C00 00C8H	*SIDISR3*	SIO Interrupt Status Register (CH3)
	1C00 00CCH	*SISCISR3*	SIO Status Change Register (CH3)
	1C00 00D0H	*SIFCR3*	SIO FIFO Control Register (CH3)
	1C00 00D4H	*SIFLCR3*	SIO Flow Control Register (CH3)
	1C00 00D8H	*SIBGR3*	SIO Baud rate Control Register (CH3)
	1C00 00DCH	*SITFIFO3*	SIO Transmit FIFO Register (CH3)
	1C00 00E0H	*SIRFIFO3*	SIO Receive FIFO Register (CH3)
TXSEI	1C00 8000H	*SEMCR*	SEI Master Control Register
	1C00 8004H	*SECR0*	SEI Control Register 0
	1C00 8008H	*SECR1*	SEI Control Register 1
	1C00 800CH	*SEFS*	SEI Inter Frame Space Register
	1C00 8010H	*SESS*	SEI Slave Select Space Register
	1C00 8014H	*SESR*	SEI Status Register
	1C00 8018H	*SEDR*	SEI Data Register
	1C00 801CH	*SERS*	SEI Read Start Register
TIMER	1C01 0000H	*TIMER*	Free running counter of periodic timers
	1C01 0004H	*TITR*	Timer Interval Time Register
	1C01 0008H	*PWMVAL*	compare value for PWM counter
MEMC	1C02 0008H	*RCCR2*	ROM Channel Control Register 2
	1C02 000CH	*RCCR3*	ROM Channel Control Register 3
	1C02 0010H	*RCCR4*	ROM Channel Control Register 4
	1C02 0014H	*RCCR5*	ROM Channel Control Register 5

Classification	Address	*Register Name*	Function
SDRAMC	1C02 8000H	*DCCR*	Configuration Register
	1C02 8004H	*DCBA*	Base Address Register
	1C02 8008H	*DCAM*	Address Mask Register
	1C02 800CH	*DCTR*	Timing Register
PORT	1C03 0000H	*PA*	PORT Data Register
	1C03 0004H	*PACR*	PORT Control Register
	1C03 0008H	*PAL*	PORT Interrupt Flag
	1C03 000CH	*PALMX*	PORT Edge Select for Interrupt
	1C03 0010H	*PAMSK*	PORT Interrupt Enable
	1C03 0014H	*PAMUX*	Output Select for PORT/ TXSEI/ UART
INTC	1C04 0000H	*IRQR*	Interrupt Request Register
	1C04 0004H	*IMASKR*	Interrupt Mask Register
	1C04 0008H	*ILEXT*	External Interrupt Detection Register
GDC	1C05 0000H	*DCR*	Display Control Register
	1C05 0010H	*SARA*	Start Address Register Layer A
	1C05 0014H	*SARB*	Start Address Register Layer B
	1C05 0018H	*SARC*	Start Address Register Layer C
	1C05 001CH	*SARD*	Start Address Register Layer D
	1C05 0020H	*MWRA*	Memory Width Register Layer A
	1C05 0024H	*MWRB*	Memory Width Register Layer B
	1C05 0028H	*MWRC*	Memory Width Register Layer C
	1C05 002CH	*MWRD*	Memory Width Register Layer D
	1C05 0030H	*HTN*	Horizontal Transfer Number
	1C05 0034H	*HTND*	Horizontal Transfer Number Layer D
	1C05 0038H	*HDSER*	Horizontal Display Start Register
	1C05 003CH	*HDSERD*	Horizontal Display Start Register Layer D
	1C05 0040H	*HCR*	Horizontal Cycle Register
	1C05 0044H	*HSWR*	Horizontal Synchronous Pulse Width
	1C05 0048H	*VCR*	Vertical Cycle Register
	1C05 004CH	*VSWR*	Vertical Synchronous pulse Width
	1C05 0050H	*VDSR*	Vertical Display Start Register
	1C05 0054H	*VDSRD*	Vertical Display Start Register layer D
	1C05 0058H	*VDER*	Vertical Display End Register
	1C05 005CH	*VDERD*	Vertical Display End Register layer D
	1C05 0800H	*CPLTA0*	Color Palette Register layer A number 0
	…	…	…
	1C05 0BFCH	*CPLTA255*	Color Palette Register layer A number 255
	1C05 0C00H	*CPLTB0*	Color Palette Register layer B number 0
	…	…	…
	1C05 0FFCH	*CPLTB255*	Color Palette Register layer B number 255
	1C05 0180H	*CPLTC0*	Color Palette Register layer C number 0
	…	…	…
	1C05 01BCH	*CPLTC15*	Color Palette Register layer C number 15
	1C05 01C0H	*CPLTD0*	Color Palette Register layer D number 0
	…	…	…
	1C05 01FCH	*CPLTD15*	Color Palette Register layer D number 15
DMAC	1C06 0000H	*ODR0*	Operation Definition Register 0
	1C06 0001H	*CCR0*	Channel Control Register 0
	1C06 0002H	*CER0*	Channel Error Register 0
	1C06 0003H	*CSR0*	Channel Status Register 0
	1C06 0004H	*SAR0*	Source Address Register 0
	1C06 0008H	*DAR0*	Destination Address Register 0
	1C06 000CH	*BCR0*	Byte Control Register 0
	1C06 0010H	*ODR1*	Operation Definition Register 1

Classification	Address	*Register Name*	Function
DMAC	1C06 0011H	*CCR1*	Channel Control Register 1
	1C06 0012H	*CER1*	Channel Error Register 1
	1C06 0013H	*CSR1*	Channel Status Register 1
	1C06 0014H	*SAR1*	Source Address Register 1
	1C06 0018H	*DAR1*	Destination Address Register 1
	1C06 001CH	*BCR1*	Byte Control Register 1
TXCAN	1C07 0000H … 1C07 00F0H	*DPRAM0*	Mailbox RAM (CH0)
	1C07 0100H	*MC0*	Mailbox Configuration Register (CH0)
	1C07 0104H	*MD0*	Mailbox Direction Register (CH0)
	1C07 0108H	*TRS0*	Transmit Request Set Register (CH0)
	1C07 010CH	*TRR0*	Transmit Request Reset Register (CH0)
	1C07 0110H	*TA0*	Transmission Acknowledge Register (CH0)
	1C07 0114H	*AA0*	Abort Acknowledge Register (CH0)
	1C07 0118H	*RMP0*	Receive Message Pending Register (CH0)
	1C07 011CH	*RML0*	Receive Message Lost Register (CH0)
	1C07 0120H	*LAM0*	Local Acceptance Mask Register (CH0)
	1C07 0124H	*GAM0*	Global Acceptance Mask Register (CH0)
	1C07 0128H	*MCR0*	Master Control Register (CH0)
	1C07 012CH	*GSR0*	Global Status Register (CH0)
	1C07 0130H	*BCR10*	Bit Configuration Register 1 (CH0)
	1C07 0134H	*BCR20*	Bit Configuration Register 2 (CH0)
	1C07 0138H	*GIF0*	Global Interrupt Flag Register (CH0)
	1C07 013CH	*GIM0*	Global Interrupt Mask Register (CH0)
	1C07 0140H	*MBTIF0*	Mailbox Transmit Interrupt Flag Register (CH0)
	1C07 0144H	*MBRIF0*	Mailbox Receive Interrupt Flag Register (CH0)
	1C07 0148H	*MBIM0*	Mailbox Interrupt Mask Register (CH0)
	1C07 014CH	*CDR0*	Change Data Request (CH0)
	1C07 0150H	*RFP0*	Remote Frame Pending Register (CH0)
	1C07 0154H	*CEC0*	CAN Error Counter Register (CH0)
	1C07 0158H	*TSP0*	Time Stamp Counter Prescaler (CH0)
	1C07 015CH	*TSC0*	Time Stamp Counter (CH0)
	1C07 8000H … 1C07 80F0H	*DPRAM1*	Mailbox RAM (CH1)
	1C07 8100H	*MC1*	Mailbox Configuration Register (CH1)
	1C07 8104H	*MD1*	Mailbox Direction Register (CH1)
	1C07 8108H	*TRS1*	Transmit Request Set Register (CH1)
	1C07 810CH	*TRR1*	Transmit Request Reset Register (CH1)
	1C07 8110H	*TA1*	Transmission Acknowledge Register (CH1)
	1C07 8114H	*AA1*	Abort Acknowledge Register (CH1)
	1C07 8118H	*RMP1*	Receive Message Pending Register (CH1)
	1C07 811CH	*RML1*	Receive Message Lost Register (CH1)
	1C07 8120H	*LAM1*	Local Acceptance Mask Register (CH1)
	1C07 8124H	*GAM1*	Global Acceptance Mask Register (CH1)
	1C07 8128H	*MCR1*	Master Control Register (CH1)
	1C07 812CH	*GSR1*	Global Status Register (CH1)
	1C07 8130H	*BCR11*	Bit Configuration Register 1 (CH1)
	1C07 8134H	*BCR21*	Bit Configuration Register 2 (CH1)
	1C07 8138H	*GIF1*	Global Interrupt Flag Register (CH1)
	1C07 813CH	*GIM1*	Global Interrupt Mask Register (CH1)
	1C07 8140H	*MBTIF1*	Mailbox Transmit Interrupt Flag Register (CH1)
	1C07 8144H	*MBRIF1*	Mailbox Receive Interrupt Flag Register (CH1)

Classification	Address	*Register Name*	Function
TXCAN	1C07 8148H	*MBIM1*	Mailbox Interrupt Mask Register (CH1)
	1C07 814CH	*CDR1*	Change Data Request (CH1)
	1C07 8150H	*RFP1*	Remote Frame Pending Register (CH1)
	1C07 8154H	*CEC1*	CAN Error Counter Register (CH1)
	1C07 8158H	*TSP1*	Time Stamp Counter Prescaler (CH1)
	1C07 815CH	*TSC1*	Time Stamp Counter (CH1)
CCR	1C08 0000H	*CCR*	Chip Configuration Register