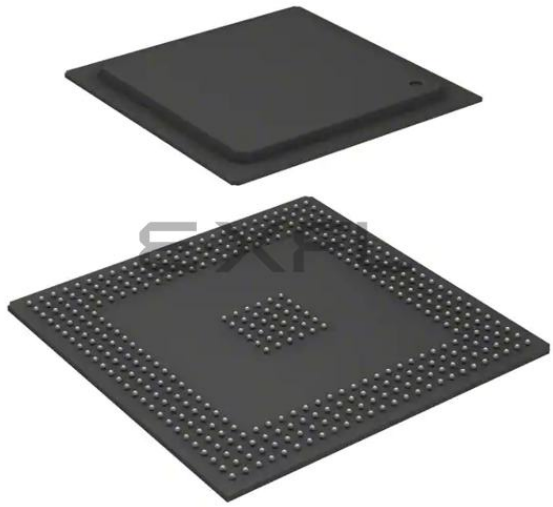


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	PowerPC
Core Size	32-Bit Single-Core
Speed	56MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	64
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 2.7V
Data Converters	A/D 32x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	388-BBGA
Supplier Device Package	388-PBGA (27x27)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mpc561mzp56r2

1.3.1.7 32-Kbyte Static RAM (CALRAM)

- Composed of one 32-Kbyte CALRAM module
  - 28-Kbyte static RAM
  - 4-Kbyte calibration (overlay) RAM feature that allows calibration of Flash-based constants
- Eight 512-byte overlay regions
- One clock fast accesses
- Two clock cycle access option for power saving
- Standby power supply (IRAMSTBY) for data retention

1.3.1.8 General Purpose I/O Support (GPIO)

- 24 address signals and 32 data signals can be used for general-purpose I/O in single-chip mode
- 16 GPIO in MIOS14
- Many peripheral signals can be used as GPIO when not used as primary functions
- 5-V outputs with slew rate control

1.3.2 Nexus Debug Port (Class 3)

- Compliant with Class 3 of the IEEE-ISTO 5001-1999
- Program trace via branch trace messaging (BTM)
- Data trace via data write messaging (DWM) and data read messaging (DRM)
- Ownership trace via ownership trace messaging (OTM)
- Run-time access to on-chip memory map and special-purpose registers (SPRs) via the READI read/write access protocol
- Watchpoint messaging via the auxiliary port
- 9 or 16 full-duplex auxiliary pin interface for medium and high visibility throughput
- All features configurable and controllable via the auxiliary port
- Supports the RCPU debug mode via the auxiliary port

1.3.3 Integrated I/O System

1.3.3.1 Two Time Processor Units (TPU3)

- True 5 V I/O
- Two time processing units (TPU3) with 16 channels each
- Each TPU3 is a micro-coded timer subsystem
- 8 Kbytes of dual port TPU RAM (DPTRAM) shared by two TPU3 modules for TPU micro-code

1.3.3.2 22-Channel Modular I/O System (MIOS14)

- Six modulus counter sub-modules (MCSM)
- 10 double-action sub-modules (DASM)

Table 1-2. Differences Between MPC555 and MPC561/MPC563 (continued)

Module	MPC555	MPC561/MPC563
TouCAN	(2) Identical (3)	
TPU3	(2) Identical (2)	
DPTRAM	(6 Kbytes) Identical (8 Kbytes)	
PPM	—	New Module

1.6 Additional MPC561/MPC563 Differences

- The MPC561/MPC563 devices are very similar to the MPC555 with the following differences:
  - Up to 66 MHz operating frequency (Refer to the applicable electrical characteristics document for more information.)
  - CDR3 technology
  - Two power supplies: 5.0-V I/O, 2.6-V external bus signals, 2.6-V internal logic
  - New modules: READI, CALRAM, PPM
  - Extra TouCAN module, additional 6 Kbytes of SRAM on L-bus (32 Kbytes total) with CALRAM overlay features, extra 2 Kbytes of DPTRAM (8 Kbytes total)
- QADC64E
  - GPO on all channel signals in addition to GPI functions
- TouCAN, TPU3, QSMCM, UIMB, Core, L2U
  - No changes
- BBC2
  - Enhanced interrupt controller support
  - Enhanced exception relocation table
  - Branch target buffer
  - 2 Kbytes of decompression RAM for code compression. This may also be used as general-purpose RAM while not used for code compression.
- CALRAM (with overlay features)
  - New module
  - Overlay features allow calibration of Flash-based constants
- UC3F (U-bus CDR3 Flash module) on MPC563/MPC564 only
  - 512 Kbytes of non-volatile memory (NVM)
  - Designed for use in embedded microcontroller (MCU) applications targeted for high speed read performance and high density byte count requirements
- READI
  - New module
- USIU
  - Enhanced interrupt controller
  - ENGCLK default frequency

Table 2-1. MPC561/MPC563 Signal Descriptions (continued)

Signal Name	No. of Signals	Type	Function after Reset[1]	Description
Interrupt Controller				
$\overline{\text{IRQ0}}$ / SGPIOC0 / MDO4	1	I	MDO4 if the Nexus (READI) port is enabled, $\overline{\text{IRQ0}}$ otherwise. See Section 2.5.	Interrupt Request 0. One of the eight external signals that can request, by means of the internal interrupt controller, a service routine from the RCPU. $\overline{\text{IRQ0}}$ is a non-maskable interrupt (NMI).
		I/O		Port SGPIOC0. Allows the signal to be used as a general-purpose input/output.
		O		READI Message Data Out. Message data out (MDO4) are output signals used for uploading OTM, BTM, DTM, and read/write accesses. External latching of MDO occurs on rising edge of MCKO. Eight signals are implemented.
$\overline{\text{IRQ1}}$ / $\overline{\text{RSV}}$ / SGPIOC1	1	I	$\overline{\text{IRQ1}}$	Interrupt Request 1. One of the eight external signals that can request, by means of the internal interrupt controller, a service routine from the RCPU.
		O		Reservation. This signal is used, together with the address bus, to indicate that the internal core initiated a transfer as a result of a STWCX or a LWARX instruction.
		I/O		Port SGPIOC1. Allows the signal to be used as a general-purpose input/output.
$\overline{\text{IRQ2}}$ / $\overline{\text{CR}}$ / SGPIOC2 / $\overline{\text{MTS}}$[2]	1	I	$\overline{\text{IRQ2}}$	Interrupt Request 2. One of the eight external signals that can request, by means of the internal interrupt controller, a service routine from the RCPU.
		I		Cancel Reservation. Instructs the MPC561/MPC563 to clear its reservation because some other master has touched its reserved space. An external bus snooper asserts this signal.
		I/O		Port SGPIOC2. Allows the signal to be used as a general-purpose input/output.
		O		Memory Transfer Start. This is the transfer start signal from the MPC561's memory controller that allows external memory access by an external bus master.
$\overline{\text{IRQ3}}$ / $\overline{\text{KR}}$ / $\overline{\text{RETRY}}$ / SGPIOC3	1	I	$\overline{\text{IRQ3}}$	Interrupt Request 3. One of the eight external signals that can request, by means of the internal interrupt controller, a service routine from the RCPU.
		I/O		Kill Reservation. In case of a bus cycle initiated by a STWCX instruction issued by the RCPU core to a non-local bus on which the storage reservation has been lost, this signal is used by the non-local bus interface to back-off the cycle.
		I/O		Retry. Indicates to a master that the cycle is terminated but should be repeated. As an input, it is driven by the external slave to retry a cycle.
		I/O		Port SGPIOC3. Allows the signal to be used as a general-purpose input/output.

Table 3-2. Supervisor-Level SPRs (continued)

SPR Number (Decimal)	Special-Purpose Register
287	Processor Version Register (PVR) See Table 3-14 for bit descriptions.
528	IMPU Global Region Attribute (MI_GRA)[1] See Table 4-8 for bit descriptions.
536	L2U Region Attribute (L2U_GRA) See Table 11-10 for bit descriptions.
560	BBC Module Configuration Register (BBC_MCR)[1] See Table 4-4 for bit descriptions.
568	L2U Module Configuration Register (L2U_MCR)[1] See Table 11-7 for bit descriptions.
784	L2U Region Base Address Register 0 (L2U_RBA0)[1] See Table 4-5 for bit descriptions.
785	IMPU Region Base Address Register 1 (MI_RBA1)[1] See Table 4-5 for bits descriptions.
786	IMPU Region Base Address Register 2 (MI_RBA2)[1] See Table 4-5 for bits descriptions.
787	IMPU Region Base Address Register 3 (MI_RBA3)[1] See Table 4-5 for bits descriptions.
816	IMPU Region Attribute Register 0 (MI_RA0)[1]. See Table 4-6 for bits descriptions.
817	IMPU Region Attribute Register 1 (MI_RA1)[1]. See Table 4-6 for bits descriptions.
818	IMPU Region Attribute Register 2 (MI_RA2)[1]. See Table 4-6 for bits descriptions.
819	IMPU Region Attribute Register 3(MI_RA3)[1]. See Table 4-6 for bits descriptions.
792	L2U Region Base Address Register 0 (L2U_RBA0)[1] See Table 11-8 for bit descriptions.
793	L2U Region Base Address Register 1 (L2U_RBA1)[1] See Table 11-8 for bit descriptions.
794	L2U Region Base Address Register 2 (L2U_RBA2)[1] See Table 11-8 for bit descriptions.
795	L2U Region Base Address Register 3 (L2U_RBA3)[1] See Table 11-8 for bit descriptions.
824	L2U Region Attribute Register 0 (L2U_RA0)[1] See Table 11-9 for bit descriptions.
825	L2U Region Attribute Register 1 (L2U_RA1)[1] See Table 11-9 for bit descriptions.
826	L2U Region Attribute Register 2 (L2U_RA2)[1] See Table 11-9 for bit descriptions.

U-bus access mode of the RAM is activated by the BBCMCR[DCAE] bit setting (see Section 4.6.2.1, "BBC Module Configuration Register (BBCMCR)"). In this mode the DECRAM can be accessed from the U-bus and cannot be accessed by the ICDU logic.

In this mode:

- The DECRAM supports word, half-word and byte operations.
- The DECRAM is emulated to be 32 bits wide. For example: a load access from offset 0 in the DECRAM will deliver the concatenation of the first word in each of the DECRAM banks when RAM 1 contains the 16 LSB of the word and RAM 2 contains the 16 MSB.
- Load accesses at any width are supplied with 32 bits of valid data.
- The DECRAM communicates with the U-bus pipeline but does not support pipelined accesses to itself. If a store operation is second in the U-bus pipe, the store is carried out immediately and the U-bus acknowledgment is performed when the previous transaction in the pipe completes.
- Burst access is not supported.

NOTE

Instructions running from the DECRAM should not also perform store operations to the DECRAM.

4.4.1.1 Memory Protection Violations

The DECRAM module does not acknowledge U-bus accesses that violate the configuration defined in the BBCMCR. This causes the machine check exception for the internal RCPU or an error condition for the MPC561/MPC563 external master.

4.4.1.2 DECRAM Standby Operation Mode

The bus interface and DECRAM control logic are powered by V_{DD} supply. The memory array is supplied by a separate power pin (IRAMSTBY).

4.5 Branch Target Buffer

The burst buffer controller contains a branch target buffer (BTB) to reduce the impact of branches on processor performance. Following is a summary of the BTB features:

- Software controlled BTB enable/disable, inhibit, and invalidate
- User transparent — no user management required

The BTB consists of eight branch target entries (BTE). Refer to Figure 4-5. All entries are managed as a fully associative cache. Each entry contains a tag and several data buffers related to this tag.

4.5.1 BTB Operation

When the RCPU generates a change of flow (COF) address for instruction fetch, the BTB control logic compares it to the tag values currently stored in the tag register file where the following events can happen:

Table 8-9. SCCR Bit Descriptions (continued)

Bits	Name	Description
3	DCSLR	Disable clock switching at loss of lock during reset. When DCSLR is clear and limp mode is enabled, the chip will switch automatically to the backup clock if the PLL losses lock during HRESET. When DCSLR is asserted, a PLL loss-of-lock event does not cause clock switching. If HRESET is asserted and DCSLR is set, the chip will not negate HRESET until the PLL acquires lock. 0 Enable clock switching if the PLL loses lock during reset 1 Disable clock switching if the PLL loses lock during reset
4	MFPDL	MF and pre-divider lock. Setting this control bit disables writes to the MF and DIVF bits. This helps prevent runaway software from changing the VCO frequency and causing the SPLL to lose lock. In addition, to protect against hardware interference, a hardware reset will be asserted if these fields are changed while LPML is asserted. This bit is writable once after power-on reset. 0 MF and DIVF fields are writable 1 MF and DIVF fields are locked
5	LPML	LPM lock. Setting this control bit disables writes to the LPM and CSRC control bits. In addition, for added protection, a hardware reset is asserted if any mode is entered other than normal-high mode. This protects against runaway software causing the MCU to enter low-power modes. (The MSR[POW] bit provides additional protection). LPML is writable once after power-on reset.) 0 LPM and CSRC bits are writable 1 LPM and CSRC bits are locked and hard reset will occur if the MCU is not in normal-high mode
6	TBS	Time base source. 0 Source is OSCCLK divided by either 4 or 16 1 Source is system clock divided by 16
7	RTDIV	RTC (and PIT) clock divider. At power-on reset this bit is cleared if MODCK[1:3] are all low; otherwise the bit is set. 0 RTC and PIT clock divided by 4 1 RTC and PIT clock divided by 256
8	STBUC	Switch to backup clock control. When software sets this bit, the system clock is switched to the on-chip backup clock ring oscillator, and the chip undergoes a hard reset. The STBUC bit is ignored if LME is cleared. 0 Do not switch to the backup clock ring oscillator 1 Switch to backup clock ring oscillator
9	CQDS	Clock quarter drive strength — The COM and CQDS bits control the output buffer strength of the CLKOUT, see Table 8-10.
10	PRQEN	Power management request enable 0 Remains in the lower frequency (defined by DFNL) even if the power management bit in the MSR is reset (normal operational mode) or if there is a pending interrupt from the interrupt controller 1 Switches to high frequency (defined by DFNH) when the power management bit in the MSR is reset (normal operational mode) or there is a pending interrupt from the interrupt controller

Table 9-1. MPC561/MPC563 BIU Signals (continued)

Signal Name	Pins	Active	I/O	Description
$\overline{\text{RETRY}}$ Retry	1	Low	I	In the case of regular transaction, this signal is driven by the slave device to indicate that the MPC561/MPC563 must relinquish the ownership of the bus and retry the cycle.
			O	When an external master owns the bus and the internal MPC561/MPC563 bus initiates access to the external bus at the same time, this signal is used to cause the external master to relinquish the bus for one clock to solve the contention.

9.5 Bus Operations

This section provides a functional description of the system bus, the signals that control it, and the bus cycles provided for data transfer operations. It also describes the error conditions, bus arbitration, and reset operation.

The MPC561/MPC563 generates a system clock output (CLKOUT). This output sets the frequency of operation for the bus interface directly. Internally, the MPC561/MPC563 uses a phase-lock loop (PLL) circuit to generate a master clock for all of the MPC561/MPC563 circuitry (including the bus interface) which is phase-locked to the CLKOUT output signal.

All signals for the MPC561/MPC563 bus interface are specified with respect to the rising edge of the external CLKOUT and are guaranteed to be sampled as inputs or changed as outputs with respect to that edge. Since the same clock edge is referenced for driving or sampling the bus signals, the possibility of clock skew could exist between various modules in a system due to routing or the use of multiple clock lines. It is the responsibility of the system to handle any such clock skew problems that could occur.

9.5.1 Basic Transfer Protocol

The basic transfer protocol defines the sequence of actions that must occur on the MPC561/MPC563 bus to perform a complete bus transaction. A simplified scheme of the basic transfer protocol is illustrated in Figure 9-3.

Figure 9-3. Basic Transfer Protocol

The basic transfer protocol provides for an arbitration phase and an address and data transfer phase. The address phase specifies the address for the transaction and the transfer attributes that describe the transaction. The data phase performs the transfer of data (if any is to be transferred). The data phase may transfer a single beat of data (four bytes or less) for nonburst operations, a 4-beat burst of data (4 x 4 bytes), an 8-beat burst of data (8 x 2 bytes) or a 16-beat burst of data (16 x 1 bytes).

For addition details, refer to Section 9.5.4, “Burst Transfer."

10.2.6 Reduced Data Setup Time

In order to meet timing requirements when interfacing to external memories, the data setup time can be reduced. This mode can be selected by programming the BR*x* registers. Thus there is flexibility in how each region can be configured to operate. The operation mode will be determined dynamically according to a particular access type. This means that for a memory region with the reduced setup time mode enabled, the mode will automatically switch to disabled when there is no requirement for the reduced setup time, (e.g., a back-to-back load/store access). For a new access with burst length more than 1, the operation mode will be automatically switched back to the reduced setup time mode.

Reduced setup time can be selected via the SST bit in BR[0:3]. See Section 10.9.3, “Memory Controller Base Registers (BR0–BR3)” for more details. If SCCR[EBDF] is greater than 0, however, an external burst access with reduced data setup time will corrupt a load/store to any USIU register.

The reduced setup time mode may or may not have a performance impact, depending on the properties of the memory. Namely, there is always an additional empty cycle between two burst sequences. On the other hand, this additional cycle, under certain conditions, may be compensated for by reducing the number of cycles in initial data access and sequential burst beats.

Table 10-1. Timing Requirements for Reduced Setup Time

CPU Specification	Memory Device Requirements
Cycle time at 56 MHz = 17.9 ns	Initial access time = 49 ns
Short setup time = 3 ns	Burst access time = 13 ns
Normal setup time = 6 ns	
Additional delay arising from on-board wires and clock skew between internal clock and CLKOUT	

10.2.6.1 Case 1: Normal Setup Time

Initial access:

$$\text{Initial access time of memory + Data setup time of CPU + Delays} = 49 + 6 + 1 = 56\text{ns}$$

To derive the number of clocks required, divide by the system clock cycle time:

$\frac{56}{17.9} = 3.13$ therefore 4 cycles are required

Burst access:

$$\text{Burst access time of memory + Data setup time of CPU + Delays} = 13 + 6 + 1 = 20\text{ns}$$

The number of clocks required $= \frac{20}{17.9} = 1.11$ therefore 2 clocks are required.

This case is illustrated in Figure 10-5.

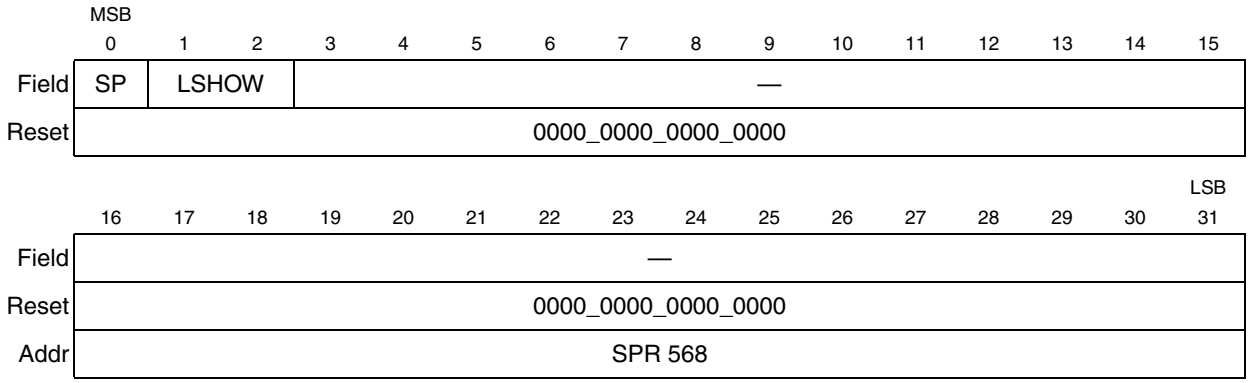


Figure 11-4. L2U Module Configuration Register (L2U_MCR)

Table 11-7. L2U_MCR Bit Descriptions

Bits	Name	Description
0	SP	CALRAM Protection (SP) bit is used to protect the CALRAM on the L-bus from U-bus accesses. Any attempt to set or clear the SP bit from the U-bus side has no affect. Once this bit is set, the L2U blocks all CALRAM accesses initiated by the U-bus masters and the access is terminated with a data error on the U-bus. If L-bus show cycles are enabled, setting this bit will disable L-bus CALRAM show cycles.
1:2	LSHOW	LSHOW bits are used to configure the show cycle mode for cycles accessing the L-bus slave e.g. CALRAM 00 Disable show cycles 01 Show address and data of all L-bus space write cycles 10 Reserved 11 Show address and data of all L-bus space read and write cycles
3:31	—	Reserved

11.8.4 Region Base Address Registers (L2U_RBAx)

The L2U region base address register (L2U_RBAx) defines the base address of a specific region protected by the data memory protection unit. There are four registers (x = 0...3), one for each supported region.

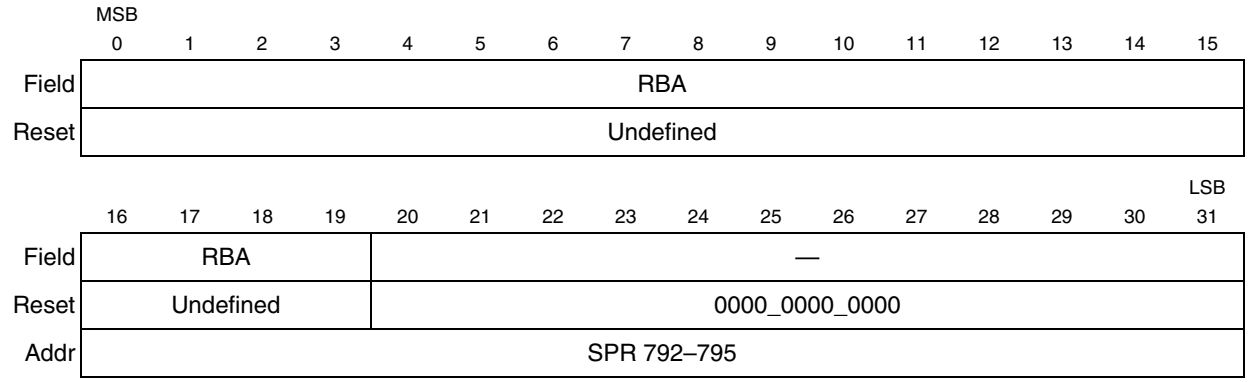


Figure 11-5. L2U Region x Base Address Register (L2U_RBAx)

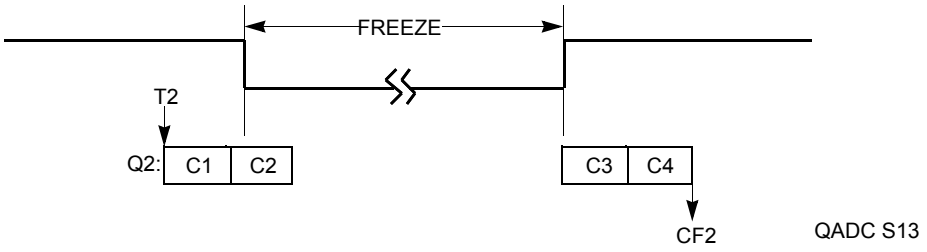


Figure 14-37. CCW Freeze Situation 13

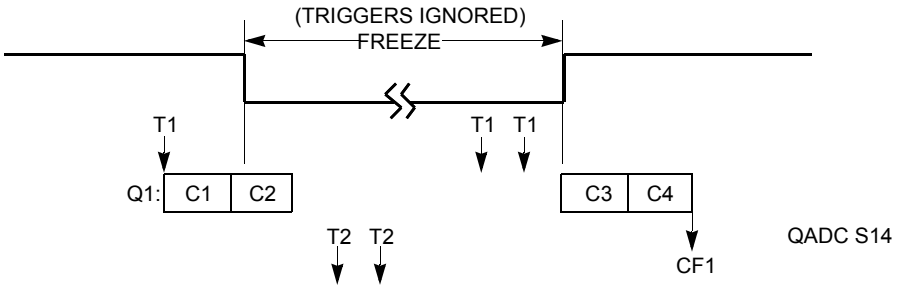


Figure 14-38. CCW Freeze Situation 14

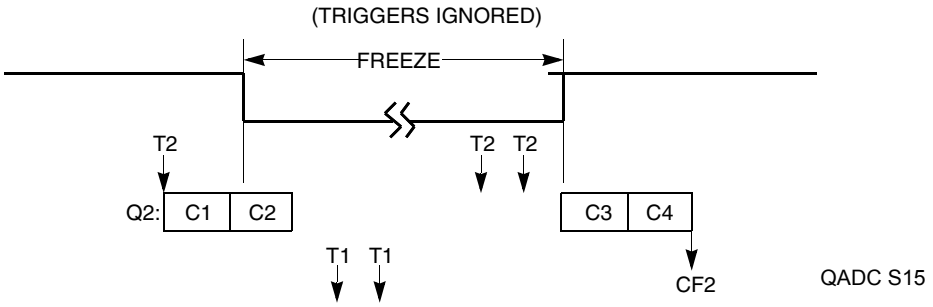


Figure 14-39. CCW Freeze Situation 15

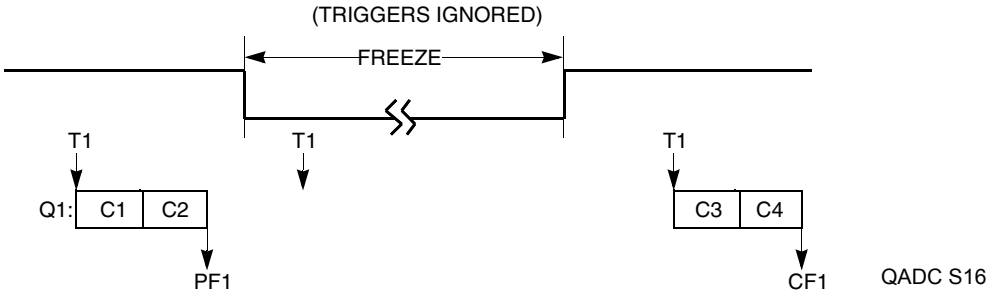


Figure 14-40. CCW Freeze Situation 16

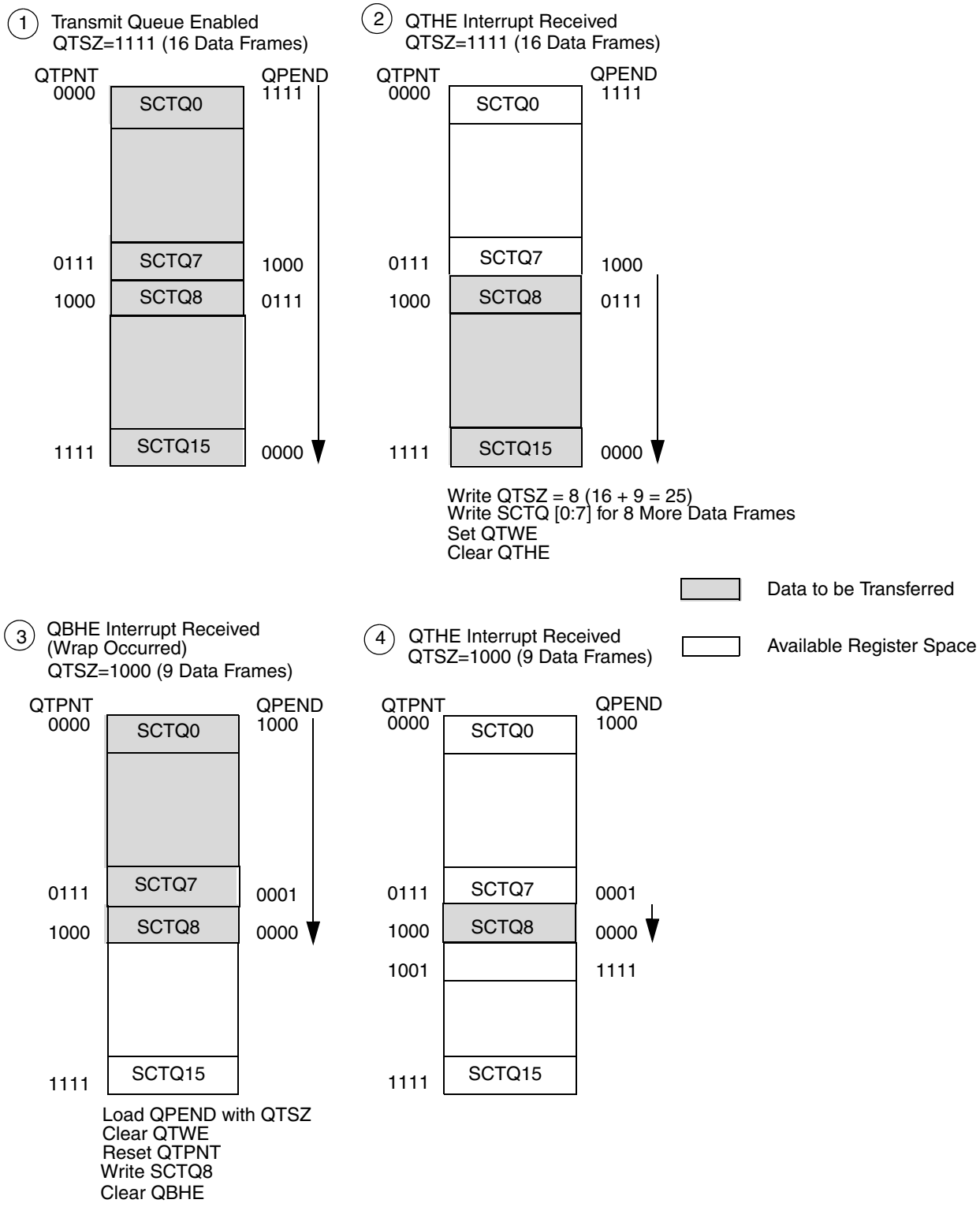


Figure 15-37. Queue Transmit Example for 25 Data Frames

For example, with 0x00FF in the counter and 0x0002 in MPWMPULR2, the period is 255 PWM clock count and the pulse width is 2 PWM clock counts.

For a given system clock frequency, with a given counter divide ratio and clock selection divide ratio, the output pulse width is given by the following equation:

$$\text{Pulse\_Width} = \frac{N_{MCPSM} \times N_{MPWMSM} \times V_{MPWMB2}}{f_{SYS}}$$

where V_{MPWMB2} is the value in the register B2

In such conditions, the minimum output pulse width that can be obtained is given by:

$$\text{Minimum\_Pulse\_Width} = \frac{N_{MCPSM} \times N_{MPWMSM}}{f_{SYS}}$$

and the maximum pulse width by:

$$\text{Maximum\_Pulse\_Width} = \frac{N_{MCPSM} \times N_{MPWMSM} \times (2^{\text{Bit\_of\_Resolution}} - 1)}{f_{SYS}}$$

17.10.3.5 Duty Cycles (0% and 100%)

The 0% and 100% duty cycles are special cases to give flexibility to the software to create a full range of outputs. The “always set” and “always clear” conditions of the output flip-flop are established by the value in register MPWMPULR2. These boundary conditions are generated by software, just like another pulse. When the PWM output is being used to generate an analog level, the 0% and 100% represent the full scale values.

The 0% output is created with a 0x0000 in register MPWMPULR2, which prevents the output flip-flop from ever being set.

The 100% output is created when the content of register MPWMPULR2 is equal to or greater than the content of register MPWMPERR. Thus, the width register match occurs on counter reload. The state sequencer provides the timing to ensure that the first appearance of a 100% value in register MPWMPULR2 causes a glitchless always-set condition of the output flip-flop when TRSP = ‘0’.

NOTE

Even if the output is forced to 100%, the 16-bit up counter continues its counting and that output changes to or from the 100% value are done synchronously to the selected period.

NOTE

When a PWM output period is selected to be 65536 PWM clocks by loading 0x0000 in the period register, it is not possible to have an 100% duty cycle output signal. In this case, the maximum duty cycle available is of 65535/65536.

- Gives an ability to control the execution of the processor and maintain control on it under all circumstances. The development port is able to force the CPU to enter to the debug mode even when external interrupts are disabled.
- It is possible to enter debug mode immediately out of reset thus allowing debugging of a ROM-less system.
- It is possible to selectively define, using an enable register, the events that will cause the machine to enter into the debug mode.
- When in debug mode detect the reason upon which the machine entered debug mode by reading a cause register.
- Entering into the debug mode in all regular cases is restartable in the sense that it is possible to continue to run the regular program from the location where it entered the debug mode.
- When in debug mode all instructions are fetched from the development port but load/store accesses are performed on the real system memory.
- Data Register of the development port is accessed using mtspr and mfspr instructions via special load/store cycles. (This feature together with the last one enables easy memory dump & load).
- Upon entering debug mode, the processor gets into the privileged state (MSR[PR] = 0). This allows execution of any instruction, and access to any storage location.
- An OR signal of all exception cause register (ECR) bits (ECR_OR) enables the development port to detect pending events while already in debug mode. An example is the ability of the development port to detect a debug mode access to a non existing memory space.

Figure 23-6 illustrates the debug mode logic implemented in the CPU.

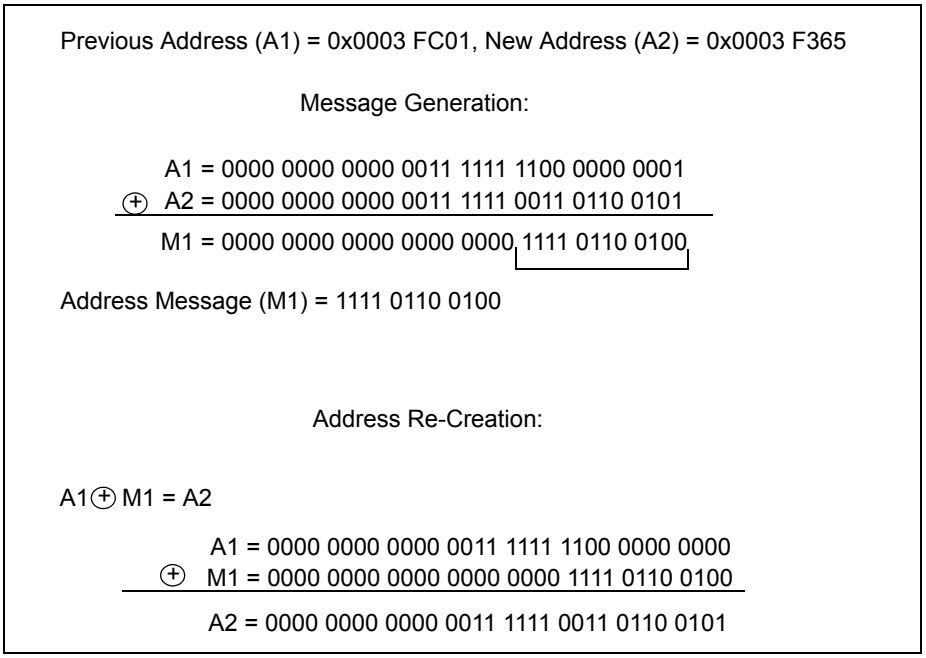


Figure 24-33. Relative Address Generation and Re-Creation

24.8.3 Queue Overflow Program Trace Error Message

A trace overrun error occurs when a trace message cannot be queued due to the queue being full, provided program trace is enabled.

The overrun error causes the message queue to be flushed, and an error message to be queued. The error code within the error message indicates that either a program/data/ownership trace overrun error has occurred or that only a program trace overrun has occurred. The next BTM will be a synchronization message. Refer to Table 24-20.

The error message has the following format:

[6 bits]	[5 bits]
TCODE (8)	Error Code (0b0 0000, 0b0 0001, 0b0 0010, 0b0 0111)

Length = 11 bits

Figure 24-34. Error Message (Queue Overflow) Format

24.8.4 Branch Trace Message Operation

24.8.4.1 BTM Capture and Encoding Algorithm

BTM is accomplished by capturing instruction fetch information from the U-bus and instruction execution information from the RCPU (VF and VFLS signals), and combining them to generate program trace messages.

The DSDI data field has a 3-bit status header followed by 7 or 32 bits of data/instruction, depending on the RCPU development port mode.

The DSDI message has the following format:

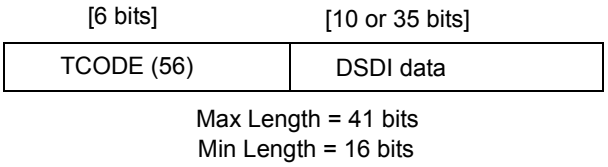


Figure 24-79. DSDI Message Format

NOTE

When sending in a DSDI data message, the DSDI data should contain the control and status bits (start, mode, control), followed by the 7 or 32-bit CPU instruction/data or trap enable, MSB first. See Figure 24-85 for DSDI data message transmission sequence.

24.14.1.2 DSDO Message

The DSDO message is used by the device to upload information from the RCPU.

The DSDO data field has a 3-bit status header followed by 7 or 32 bits of data/instruction, depending on the RCPU development port mode.

The three status bits in the DSDO data indicates if the device is ready to receive the next message from the tool.

The DSDO message has the following format:

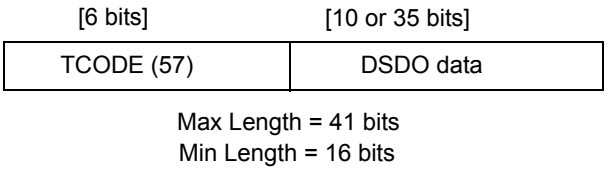


Figure 24-80. DSDO Message Format

NOTE

The DSDO data received will contain the control and status bits and data from the CPU, MSB first. See Figure 24-85 for DSDO data message transmission sequence.

24.14.1.3 BDM Status Message

BDM status message is generated by the device to let the tool know about the status of debug mode.

BDM status message (with BDM status field equal to 0b1) is sent when the RCPU is in debug mode and the device is ready to receive debug mode messages.

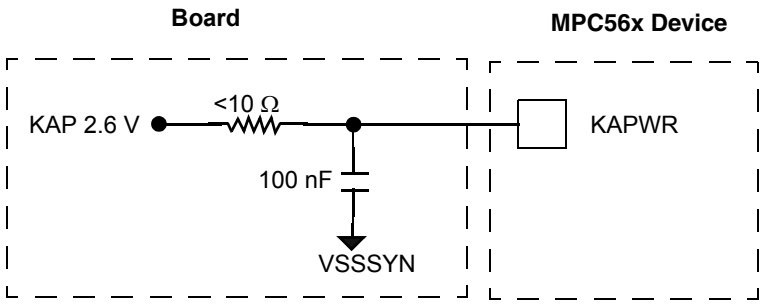
- EXTAL pad capacitance is C_{PAD} = ~7 pF

C_{SOCKET} is socket and board trace capacitance

- Socket capacitance C <=1pF
- Board trace capacitance C <=1pF. This should be low since the crystal must be located very close to the chip.

C.2.1 KAPWR Filtering

The KAPWR signal is the MPC561/MPC563 keep-alive power. KAPWR is used for the crystal oscillator circuit, and should be isolated from the noisy supplies. It is recommended that an RC filter be used on KAPWR, or bypass capacitors that are located as close as possible to the part.



Note: A filter cut off frequency of 500Hz is recommended, however this will result in a capacitor size of 33uF using a 10 Ohm resistor. This may be too expensive or large for the system. In this case the filter shown with cut- off frequency of 160kHz will suffice.

Figure C-4. RC Filter Example

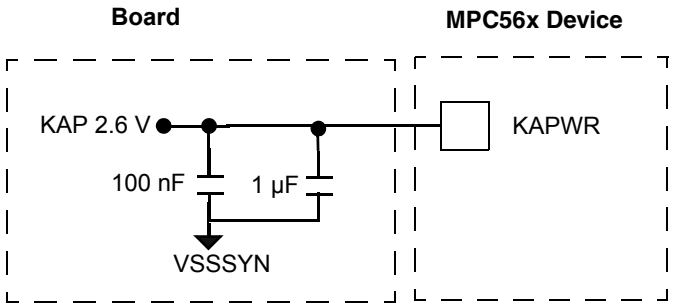


Figure C-5. Bypass Capacitors Example (Alternative)

C.2.2 PLL External Components

VDDSYN and VSSSYN are the PLL dedicated power supplies. These supplies must be used only for the PLL and isolated from all other noisy signals in the board. VDDSYN could be isolated with RC filter (see Figure C-6), or LC filter. The maximum noise allowed on VDDSYN, and VSSSYN is 50 mV with typical cut-off frequency of 500 Hz.

Table F-28. MPC561/MPC563 Signal Names and Pin Names (continued)

Signal Name	Pin Name	Ball Assignment
$\overline{BR}$/VF1/IWP2	br_b_vf1_iwp2	R4
$\overline{BB}$/VF2/IWP3	bb_b_vf2_iwp3	R2
IWP[0:1]/VFLS[0:1]	iwp0_vfls0	N2
	iwp1_vfls1	N3
TMS/$\overline{EVTI}$	tms_evti_b	M2
TDI/DSDI/MDI0	tdi_dsdi_mdi0	M1
TCK/DSCK/MCKI	tck_dsck_mcki	L2
TDO/DSDO/MDO0	tdo_dsdo_mdo0	M4
JCOMP/$\overline{RSTI}$	jcomp_rsti_b	L1
XTAL	xtal	AD26
EXTAL	extal	AC26
XFC	xfc	AA26
CLKOUT	clkout	U23
EXTCLK	extclk	V24
ENGCLK/BUCLK	engclk_buclk	V26
QSMCM		
PCS0/$\overline{SS}$/QGPIO0	pcs0_ss_b_qgpio0	N25
PCS[1:3]/QGPIO[1:3]	pcs1_qgpio1	N24
	pcs2_qgpio2	N23
	pcs3_qgpio3	P26
$\overline{MISO}$/QGPIO4	miso_b_qgpio4	P25
$\overline{MOSI}$/QGPIO5	mosi_b_qgpio5	P24
SCK/QGPIO6	sck_qgpio6	P23
TXD1/QGPO1	txd1_qgpo1	R25
TXD2/QGPO2/C_CNTX0	txd2_qgpo2_c_cntx0	R24
RXD1/QGPI1	rxd1_qgpi1	R23
RXD2/QGPI2/C_CNRX0	rxd2_qgpi2_c_cnrx0	T26
MIOS14		

Table G-2. Thermal Characteristics (continued)

Characteristic	Symbol	Value	Unit
BGA Package Thermal Resistance, Junction to Board	$R_{\theta JB}$	21.2 [3,6]	°C/W
BGA Package Thermal Resistance, Junction to Case (top)	$R_{\theta JT}$	7.0[3,7]	°C/W
BGA Package Thermal Resistance, Junction to Package Top, Natural Convection	Ψ_{JT}	1.6[8]	°C/W

1 Junction temperature is a function of on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and the board thermal resistance.

2 Per SEMI G38-87 and JESD51-2 with the board horizontal.

3 These values are the mean + 3 standard deviations of characterized data.

4 Junction temperature is a function of on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and the board thermal resistance.

5 Per JESD51-6 with the board horizontal.

6 Thermal resistance between the die and the printed circuit board (Four layer (2s2p) board, natural convection).

7 Indicates the thermal resistance between the die and the case top surface as measured by the cold plate method (MIL SPEC-883 Method 1012.1) with the cold plate temperature used for the case temperature.

8 Thermal characterization parameter indicating the temperature difference between package top and the junction temperature per EIA/JESD51-2.

An estimation of the chip junction temperature, T_J, in °C can be obtained from the equation:

$$T_J = T_A + (R_{\theta JA} \times P_D)$$

where:

T_A = ambient temperature (°C)

$R_{\theta JA}$ = package junction to ambient resistance (°C/W)

P_D = power dissipation in package

The junction to ambient thermal resistance is an industry standard value which provides a quick and easy estimation of thermal performance. Unfortunately, the answer is only an estimate; test cases have demonstrated that errors of a factor of two are possible. As a result, more detailed thermal characterization is supplied.

Historically, the thermal resistance has frequently been expressed as the sum of a junction to case thermal resistance and a case to ambient thermal resistance:

$$R_{\theta JA} = R_{\theta JC} + R_{\theta CA}$$

where:

$R_{\theta JA}$ = junction to ambient thermal resistance (°C/W)

$R_{\theta JC}$ = junction to case thermal resistance (°C/W)

$R_{\theta JA}$ = case to ambient thermal resistance (°C/W)

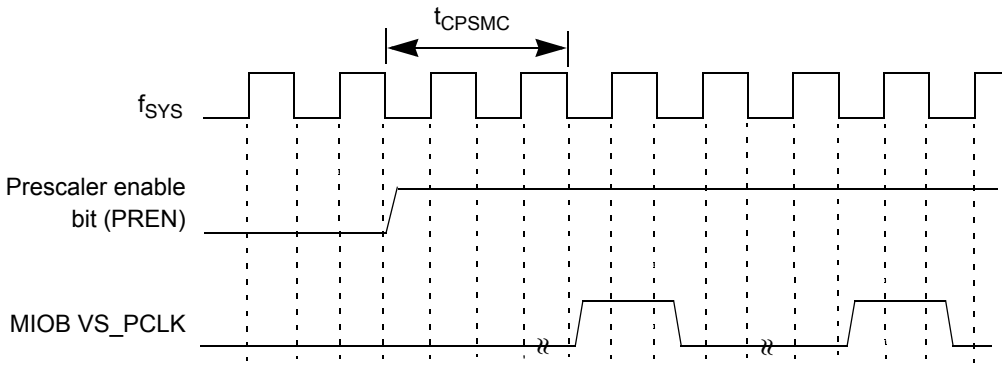
Table G-23. MCPSM Timing Characteristics

Note: After reset MCPSMSCR_PSL[3:0] is set to 0b0000.

Note: VS_PCLK is the MIOS prescaler clock which is distributed to all the counter (e.g., MPWMSM and MMCSM) submodules.

Characteristic	Symbol	Delay	Unit
MCPSM enable to VS_PCLK pulse[1]	t_{CPSMC}	(MCPSMSCR_PSL[3:0]) -1	System Clock Cycles

1 The MCPSM clock prescaler value (MCPSMSCR_PSL[3:0]) should be written to the MCPSMSCR (MCPSM Status/Control Register) before rewriting the MCPSMSCR to set the enable bit (MCPSMSCR_PREN). If this is not done the prescaler will start with the old value in the MCPSMSCR_PSL[3:0] before reloading the new value into the counter.



Note 1: f_{SYS} is the internal system clock for the IMB3 bus.
Note 2: The numbers associated with the f_{SYS} ticks refer to the IMB3 internal state.
Note 3: vs_pclk is the MIOS prescaler clock which is distributed around the MIOS to counter modules such as the MMCSM and MPWMSM.

Figure G-46. MCPSM Enable to VS_PCLK Pulse Timing Diagram

G.21.1 MPWMSM Timing Characteristics

Table G-24. MPWMSM Timing Characteristics

Note: All delays are in system clock periods.

Characteristic	Symbol	Min	Max
PWMSM output resolution	t_{PWMR}	— [1]	2.0 [2]
PWM output pulse[3]	t_{PWMO}	2.0	—
MPWMI input pin to MPWMSCR_PIN status set	t_{PIN}	1	2
CPSM enable to output set[4]	t_{PWMP}	(MPWMPERR - MPWMPULR + 1) * (256 - MPWMSCR_CP) * MCPSMSCR_PSL + 1	
MPWMSM Enable to output set (MIN)[5]	t_{PWME}	(MPWMPERR - MPWMPULR) * (256 - MPWMSCR_CP) * MCPSMSCR_PSL + 3 + (255 - MPWMSCR_CP) * MCPSMSCR_PSL[6]	