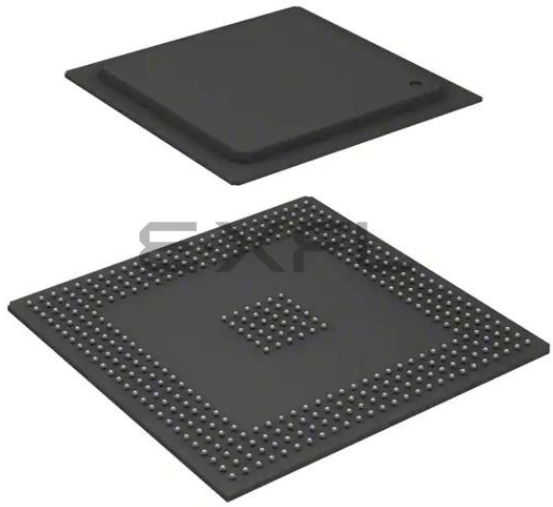

Welcome to E-XFL.COM

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	Obsolete
Core Processor	PowerPC
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	56
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
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 ~ 85°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/mpc564cvr40

Table 2-14. MPC561/MPC563 Signal Reset State (continued)

Signal List[1]	Voltage	Slew Rate Controlled Option?	Drive Load (pF)[2]	Reset State	Hysteresis Enabled?	Function After HRESET, PORESET/TRST
SGPIOC6 [3] /	5 V	Yes	50 ; 50 [5]	PD until PRDS is set	No	PTR
FRZ /	2.6 V	No	50 ; 25	PD until reset negates	No	
PTR	2.6 V	No	50 ; 25	PD until reset negates	No	
SGPIOC7 [3]/	5 V	Yes	50 ; 50 [5]	PD until PRDS is set	Yes	LWP0
IRQOUT /	2.6 V	No	50 ; 25	PD until reset negates [6]	No	
LWP0	2.6 V	No	50 ; 25	PD until reset negates [6]	No	
BG [7]/	2.6 V	No	50 ; 25	PU2.6 when driver not enabled or until SPRDS is set	No	Controlled by DBGC in reset config word. See Table 6-8.
VF0 [7]/	2.6 V	No	50 ; 25		No	
LWP1 [7]	2.6 V	No	50 ; 25		No	
BR [7]/	2.6 V	No	50 ; 25	PU2.6 when driver not enabled or until SPRDS is set	No	Controlled by DBGC in reset config word. See Table 6-8.
VF1 [7]/	2.6 V	No	50 ; 25		No	
IWP2 [7]	2.6 V	No	50 ; 25		No	
BB [7, 12]/	2.6 V	No	50 ; 25	PU2.6 when driver not enabled or until SPRDS is set	No	Controlled by DBGC in reset config word. See Table 6-8.
VF2 [7]/	2.6 V	No	50 ; 25		No	
IWP3 [7]	2.6 V	No	50 ; 25		No	
IWP[0:1] [7]/	2.6 V	No	50 ; 25	PU2.6 until reset negates	No	Controlled by DBGC in the reset config word. See Table 6-8.
VFLS[0:1] [7]	2.6 V	No	50 ; 25		No	
JTAG/BDM/READI						
TMS /	2.6 V	No	NA	PU2.6 if in JTAG mode, otherwise PD until SPRDS is set[14]	No	TMS unless the Nexus (READI) port is enabled, then EVTI. See Section 2.5.
EVTI	2.6 V	No	NA		No	
TDI /	2.6 V	No	NA	PU2.6 if in JTAG mode, otherwise PD until SPRDS is set [6]	No	DSDI unless the Nexus (READI) port (MDI0) or JTAG mode (TDI) is enabled. See Section 2.5.
DSDI /	2.6 V	No	NA		No	
MDI0	2.6 V	No	NA		No	
TCK /	2.6 V	No	NA	PD until reset negates [6]	Yes	DSCK unless the Nexus (READI) port (MCKI) or JTAG mode (TCK) is enabled. See Section 2.5.
DSCK /	2.6 V	No	NA		Yes	
MCKI	2.6 V	No	NA		Yes	

Table 3-3. Development Support SPRs[1] (continued)

SPR Number (Decimal)	Special-Purpose Register
158	I-bus Support Control Register (ICTRL) See Table 23-26 for bit descriptions.
159	Breakpoint Address Register (BAR) See Table 23-28 for bit descriptions.
630	Development Port Data Register (DPDR) See Section 23.6.13, "Development Port Data Register (DPDR)," for bit descriptions.

1 All development-support SPRs are implementation-specific.

Unless otherwise noted, reserved fields should be written with a zero when written and return zero when read. An exception to this rule is XER[16:23]; see Section 3.7.5, "Integer Exception Register (XER)." These bits are set to the value written to them and return that value when read.

3.7 User Instruction Set Architecture (UISA) Register Set

The UISA registers can be accessed by either user- or supervisor-level instructions. The general-purpose registers are accessed through instruction operands.

3.7.1 General-Purpose Registers (GPRs)

Integer data is manipulated in the integer unit's thirty-two 32-bit GPRs, shown below. These registers are accessed as source and destination registers through operands in the instruction syntax.

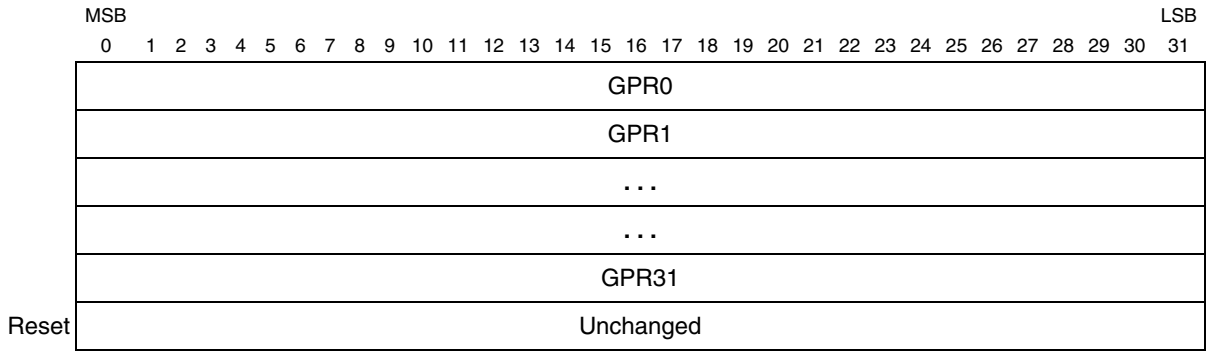

Figure 3-4. General-Purpose Registers (GPRs)

3.7.2 Floating-Point Registers (FPRs)

The PowerPC ISA architecture provides 32 64-bit FPRs. These registers are accessed as source and destination registers through operands in floating-point instructions. Each FPR supports the double-precision, floating-point format. Every instruction that interprets the contents of an FPR as a

3.8 VEA Register Set — Time Base (TB)

The virtual environment architecture (VEA) defines registers in addition to the UISA register set. The VEA register set can be accessed by all software with either user- or supervisor-level privileges. Refer to Section 6.1.7, "Time Base (TB)," for more information.

3.9 OEA Register Set

The operating environment architecture (OEA) includes a number of SPRs and other registers that are accessible only by supervisor-level instructions. Some SPRs are RCPU-specific; some RCPU SPRs may not be implemented in other PowerPC ISA processors, or may not be implemented in the same way.

3.9.1 Machine State Register (MSR)

The machine state register is a 32-bit register that defines the state of the processor. When an exception occurs, the contents of the MSR are loaded into SRR1, and the MSR is updated to reflect the exception-processing machine state. The MSR can also be modified by the mtmsr, sc, and rfi instructions. It can be read by the mfmsr instruction.

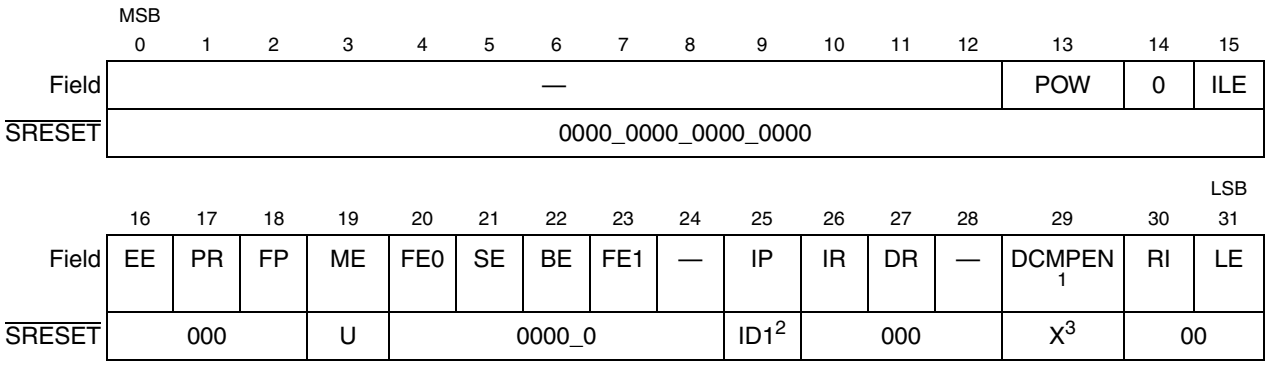

Figure 3-11. Machine State Register (MSR)

1 This bit is available only on code compression-enabled options of the MPC561/MPC563.

2 The reset value is a reset configuration word value extracted from the internal bus line. Refer to Section 7.5.2, "Hard Reset Configuration Word (RCW)."

3 The reset value is defined by the equation "BBCMCR[EN_COMP] AND BBCMCR[EXC_COMP]". At $\overline{\text{HRESET}}$ the BBCMCR[EN_COMP] and BBCMCR[EXC_COMP] bits recieve their values from RCW bits 21 and 22. The BBCMCR does not change at $\overline{\text{SRESET}}$. Thus the DCMPEN reset value may be different on $\overline{\text{SRESET}}$ and $\overline{\text{HRESET}}$, if software changes these BBCMCR bits from their reset values.

Table 3-11 shows the bit definitions for the MSR.

Table 3-11. Machine State Register Bit Descriptions

Bits	Name	Description
0:12	—	Reserved
13	POW	Power management enable. 0 Power management disabled (normal operation mode) 1 Power management enabled (reduced power mode)

3.14.5 Enforce In-Order Execution of I/O (eieio) Instruction

When executing an eieio instruction, the load/store unit will wait until all previous accesses have terminated before issuing cycles associated with load/store instructions following the eieio instruction.

3.14.6 Time Base

A description of the time base register may be found in Chapter 6, "System Configuration and Protection," and in Chapter 8, "Clocks and Power Control."

3.15 Operating Environment Architecture (OEA)

The MPC561/MPC563 has an internal memory space that includes memory-mapped control registers and internal memory used by various modules on the chip. This memory is part of the main memory as seen by the RCPU and can be accessed by an external system master.

3.15.1 Branch Processor Registers

3.15.1.1 Machine State Register (MSR)

The floating-point exception mode encoding in the RCPU is as shown in Table 3-21.

Table 3-21. Floating-Point Exception Mode Encoding

Mode	FE0	FE1
Ignore exceptions	0	0
Precise	0	1
Precise	1	0
Precise	1	1

The SF bit is reserved set to zero. The IP bit initial state after reset is set as programmed by the reset configuration as specified by the USIU characteristics.

3.15.1.2 Branch Processors Instructions

The RCPU implements all the instructions defined for the branch processor in the UISA in the hardware.

3.15.2 Fixed-Point Processor

3.15.2.1 Special Purpose Registers

- Unsupported Registers — The following registers are not supported by the MPC561/MPC563: SDR, EAR, IBAT0U, IBAT0L, IBAT1U, IBAT1L, IBAT2U, IBAT2L, IBAT3U, IBAT3L, DBAT0U, DBAT0L, DBAT1U, DBAT1L, DBAT2L, DBAT3U, DBAT3L.
- Added Registers — For a list of added special purpose registers, refer to Table 3-2, and Table 3-3.

[1] If the exception occurs during an instruction fetch in Decompression On mode, the SRR0 register will contain an indeterminate value.

Execution resumes at offset 0x1300 from the base address indicated by MSR[IP].

3.15.4.15 Implementation-Specific Data Protection Error Exception (0x1400)

The implementation-specific data protection error exception occurs in the following case:

- The data access violates the storage protection and MSR[DR]=1. See Chapter 11, “L-Bus to U-Bus Interface (L2U).”

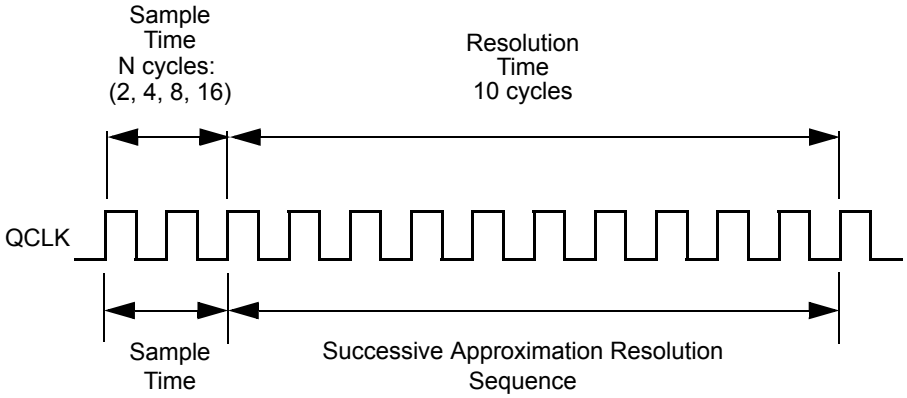

Figure 13-22. Bypass Mode Conversion Timing

13.4.2 Channel Decode and Multiplexer

The internal multiplexer selects one of the 16 analog input signals for conversion. The selected input is connected to the sample buffer amplifier. The multiplexer also includes positive and negative stress protection circuitry, which prevents deselected channels from affecting the selected channel when current is injected into the deselected channels. Refer to Appendix F, "Electrical Characteristics," for specific current levels.

13.4.3 Sample Buffer Amplifier

The sample buffer is used to raise the effective input impedance of the A/D converter, so that external components (higher bandwidth or higher impedance) are less critical to accuracy. The input voltage is buffered onto the sample capacitor to reduce crosstalk between channels.

13.4.4 Digital-to-Analog Converter (DAC) Array

The digital to analog converter (DAC) array consists of binary-weighted capacitors and a resistor-divider chain. The reference voltages, V_{RH} and V_{RL}, are used by the DAC to perform ratiometric conversions. The DAC also converts the following three internal channels:

- V_{RH} — Reference voltage high
- V_{RL} — Reference voltage low
- $(V_{RH} - V_{RL})/2$ — Reference voltage

The DAC array serves to provide a mechanism for the successive approximation A/D conversion.

Resolution begins with the most significant bit (MSB) and works down to the least significant bit (LSB). The switching sequence is controlled by the comparator and successive-approximation register (SAR) logic.

- Sample capacitor — The sample capacitor is employed to sample and hold the voltage to be converted.

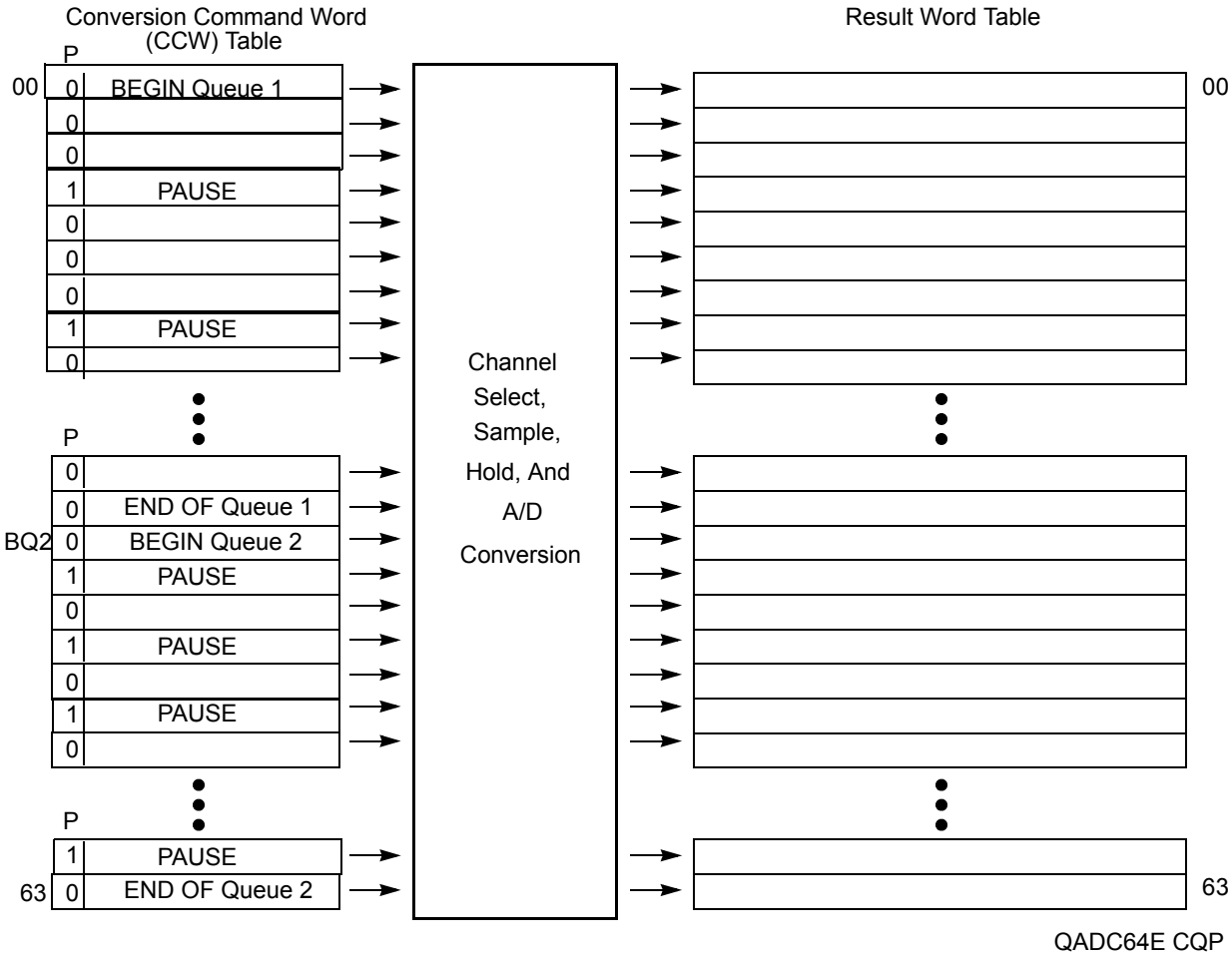

Figure 13-23. QADC64E Queue Operation with Pause

The queue operating mode selected for queue 1 determines what type of trigger event causes the execution of each of the sub-queues within queue 1. Similarly, the queue operating mode for queue 2 determines the type of trigger event required to execute each of the sub-queues within queue 2.

For example, when the external trigger rising edge continuous-scan mode is selected for queue 1, and there are six sub-queues within queue 1, a separate rising edge is required on the external trigger signal after every pause to begin the execution of each sub-queue (refer to Figure 13-23). Refer to Section 13.5.4, "Scan Modes," for information on different scan modes.

The choice of single-scan or continuous-scan applies to the full queue, and is not applied to each sub-queue. Once a sub-queue is initiated, each CCW is executed sequentially until the last CCW in the sub-queue is executed and the pause state is entered. Execution can only continue with the next CCW, which is the beginning of the next sub-queue. A sub-queue cannot be executed a second time before the overall queue execution has been completed. Refer to Section 13.3.7, "Control Register 2 (QACR2)," for more information.

Trigger events which occur during the execution of a sub-queue are ignored, except that the trigger overrun flag is set. When a continuous-scan mode is selected, a trigger event occurring after the completion of the

14.2 Key Features and Quick Reference Diagrams

This section gives an overview of the implementation of the two QADC64E modules on the MPC561/MPC563. It can also be used for a quick reference while programming the modules.

14.2.1 Features of the QADC64E Enhanced Mode Operation

- Internal sample and hold
- Directly supports up to four external multiplexers (for example, the MC14051)
- Up to 41 analog input channels using QADC64 external multiplexing
- Programmable input sample time for various source impedances
- Minimum conversion time of 7µs (with typical QCLK frequency, 2 MHz)
- Two conversion command queues with a total of 64 entries
- Sub-queues possible using pause mechanism
- Queue complete and pause software interrupts available on both queues
- Queue pointers indicate current location for each queue
- Automated queue modes initiated by
  - External edge trigger
  - Periodic/Interval timer, within QADC64E module
  - Software command
  - External gated trigger (queue 1 only)
- Single-scan or continuous-scan of queues
- 64 result registers in each QADC64E module
- Output readable in three formats
  - Right-justified unsigned
  - Left-justified signed
  - Left-justified unsigned
- Unused analog channels can be used as digital input/output signals
- Modulus prescaler can divide the system clock for the converter by two to 128
- Alternate reference input, with control in the conversion command word (CCW)

The analog section includes input signals, an analog multiplexer, and the sample and hold circuits. The analog conversion is performed by the digital-to-analog converter (DAC) resistor-capacitor array and a high-gain comparator.

The digital control section contains queue control logic to sequence the conversion process and interrupt generation logic. Also included are the periodic/interval timer, control and status registers, the conversion command word (CCW) table RAM, and the result table RAM.

The bus interface unit (BIU) allows the QADC64E to operate with the applications software through the IMB3 environment.

WARNING

A change in the prescaler value while a conversion is in progress is likely to corrupt the result from any conversion in progress. Therefore, any prescaler write operation should be done only when both queues are in the disabled modes.

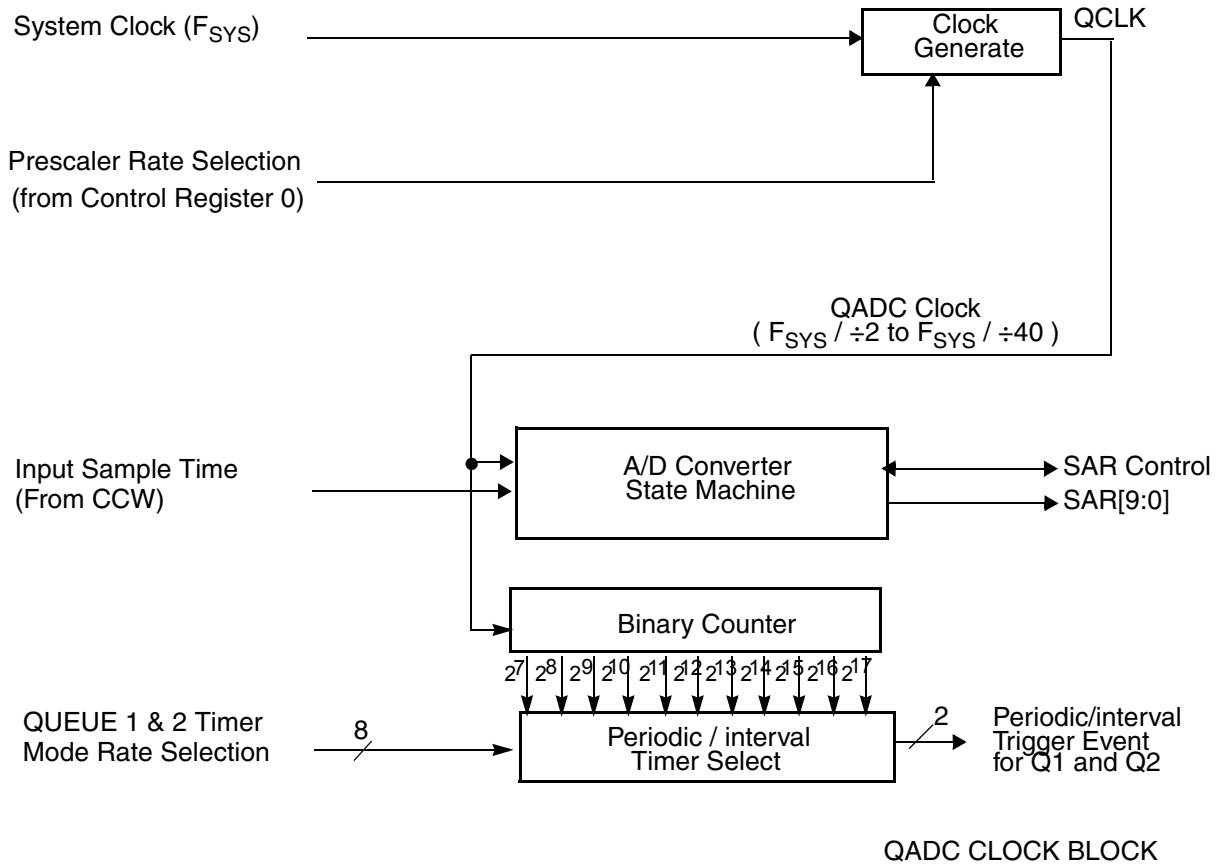

Figure 14-23. QADC64E Clock Subsystem Functions

The MCU system clock frequency (IMB3 system clock – f_{SYSCLK}) is the basis of the QADC64E timing. QCLK is generated by a software selectable prescaler that divides f_{SYSCLK} thus allowing the A/D conversion time to be maximized across f_{SYSCLK}. The software establishes the frequency of QCLK waveform by setting the PRESCALER field in the QACR0 register.

When the value of PRESCALER > 0 the resulting frequency of QCLK is calculated using the following formula:

$$f_{QCLK} = f_{SYSCLK} / (PRESCALER + 1)$$

The QADC64E requires that f_{SYSCLK} be at least twice f_{QCLK}. Therefore if the value in the PRESCALER field is set to Zero, the resulting QCLK frequency is calculated to be:

$$f_{QCLK} = f_{SYSCLK} / 2$$

15.6.1.3 QSPI Control Register 2 (SPCR2)

SPCR2 contains QSPI queue pointers, wraparound mode control bits, and an interrupt enable bit. The CPU has read/write access to SPCR2, but the QSPI has read access only. Writes to this register are buffered. New SPCR2 values become effective only after completion of the current serial transfer. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location. Reads of SPCR2 return the current value of the register, not the buffer.

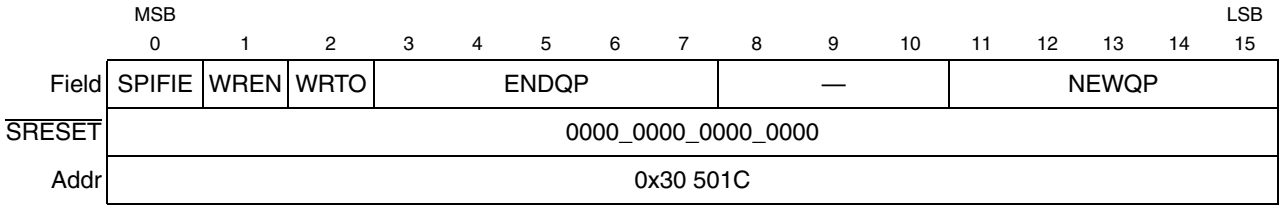

Figure 15-13. SPCR2 — QSPI Control Register 2

Table 15-16. SPCR2 Bit Descriptions

Bits	Name	Description
0	SPIFIE	SPI finished interrupt enable. Refer to Section 15.6.4.2, “QSPI Interrupts.” 0 QSPI interrupts disabled 1 QSPI interrupts enabled
1	WREN	Wrap enable. Refer to Section 15.6.5.8, “Master Wraparound Mode.” 0 Wraparound mode disabled. 1 Wraparound mode enabled.
2	WRTO	Wrap to. When wraparound mode is enabled and after the end of queue has been reached, WRTO determines which address the QSPI executes next. The end of queue is determined by an address match with ENDQP. 0 Wrap to pointer address 0x0 1 Wrap to address in NEWQP
3:7	ENDQP	Ending queue pointer. This field determines the last absolute address in the queue to be completed by the QSPI. After completing each command, the QSPI compares the queue pointer value of the just-completed command with the value of ENDQP. If the two values match, the QSPI sets SPIF to indicate it has reached the end of the programmed queue. Refer to Section 15.6.4, “QSPI Operation” for more information.
8:10	—	Reserved
11:15	NEWQP	New queue pointer value. This field contains the first QSPI queue address. Refer to Section 15.6.4, “QSPI Operation” for more information.

15.6.1.4 QSPI Control Register 3 (SPCR3)

SPCR3 contains the loop mode enable bit, halt and mode fault interrupt enable, and the halt control bit. The CPU has read/write access to SPCR3, but the QSPI has read access only. SPCR3 must be initialized before QSPI operation begins. Writing a new value to SPCR3 while the QSPI is enabled disrupts operation.

Table 15-25. SCCxR1 Bit Descriptions (continued)

Bits	Name	Description
10	RIE	Receiver interrupt enable 0 SCI RDRF and OR interrupts disabled. 1 SCI RDRF and OR interrupts enabled.
11	ILIE	Idle-line interrupt enable 0 SCI IDLE interrupts disabled. 1 SCI IDLE interrupts enabled.
12	TE	Transmitter enable 0 SCI transmitter disabled (TXD pin can be used as general-purpose output) 1 SCI transmitter enabled (TXD pin dedicated to SCI transmitter).
13	RE	Receiver Enable 0 SCI receiver disabled (RXD pin can be used as general-purpose input). 1 SCI receiver enabled (RXD pin is dedicated to SCI receiver).
14	RWU	Receiver wakeup. Refer to Section 15.7.7.10, “Receiver Wake-Up.” 0 Normal receiver operation (received data recognized). 1 Wakeup mode enabled (received data ignored until receiver is awakened).
15	SBK	Send break 0 Normal operation. 1 Break frame(s) transmitted after completion of current frame.

15.7.4 SCI Status Register (SCxSR)

SCxSR contains flags that show SCI operating conditions. These flags are cleared either by SCIx hardware or by a read/write sequence. The sequence consists of reading the SCxSR (either the upper byte, lower byte, or the entire half-word) with a flag bit set, then reading (or writing, in the case of flags TDRE and TC) the SCxDR (either the lower byte or the half-word).

The contents of the two 16-bit registers SCxSR and SCxDR appear as upper and lower half-words, respectively, when the SCxSR is read into a 32-bit register. An upper byte access of SCxSR is meaningful only for reads. Note that a word read can simultaneously access both registers SCxSR and SCxDR. This action clears the receive status flag bits that were set at the time of the read, but does not clear the TDRE or TC flags. To clear TC, the SCxSR read must be followed by a write to register SCxDR (either the lower byte or the half-word). The TDRE flag in the status register is read-only.

If an internal SCI signal for setting a status bit comes after the CPU has read the asserted status bits but before the CPU has read or written the SCxDR, the newly set status bit is not cleared. Instead, SCxSR must be read again with the bit set and SCxDR must be read or written before the status bit is cleared.

NOTE

None of the status bits are cleared by reading a status bit while it is set and then writing zero to that same bit. Instead, the procedure outlined above must be followed. Note further that reading either byte of SCxSR causes all 16 bits to be accessed, and any status bits already set in either byte are armed to clear on a subsequent read or write of SCxDR.

Table 19-8. DSCR Bit Descriptions (continued)

Bits	Name	Description
7:8	FRZ	FREEZE assertion response. The FRZ bits specify the TPU microengine response to the IMB3 FREEZE signal 00 Ignore freeze 01 Reserved 10 Freeze at end of current microcycle 11 Freeze at next time-slot boundary
9	CCL	Channel Conditions Latch. CCL controls the latching of channel conditions match recognition latch (MRL) and transition detect latch (TDL) when the CHAN register is written. Refer to the TPU Reference Manual (TPURM/AD) for further information. 0 Only the pin state condition of the new channel is latched as a result of the write CHAN register microinstruction 1 Pin state, MRL, and TDL conditions of the new channel are latched as a result of a write CHAN register microinstruction
10	BP	Breakpoint enable for microprogram counter (μPC) 0 Breakpoint not enabled 1 Break if μPC equals μPC breakpoint register
11	BC	Channel breakpoint enable 0 Breakpoint not enabled 1 Break if CHAN register equals channel breakpoint register at beginning of state or when CHAN is changed through microcode
12	BH	Host service breakpoint enable 0 Breakpoint not enabled 1 Break if host service latch is asserted at beginning of state
13	BL	Link service breakpoint enable 0 Breakpoint not enabled 1 Break if link service latch is asserted at beginning of state
14	BM	MRL breakpoint enable 0 Breakpoint not enabled 1 Break if MRL is asserted at beginning of state
15	BT	TDL breakpoint enable 0 Breakpoint not enabled 1 Break if TDL is asserted at beginning of state

[1] T4 is one of the four basic timers (T1, T2, T3 & T4) used for microengine timing.

19.4.3 Development Support Status Register (DSSR)

This register is accessible only when the TPU is in test mode; see Section 19.4.14, “Factory Test Registers.”

23.4.5.1 SGPIO6/FRZ/$\overline{\text{PTR}}$ Signal

The SGPIOC6/FRZ/$\overline{\text{PTR}}$ signal powers up as the $\overline{\text{PTR}}$ function and its function is controlled by the GPC bits in the SIUMCR.

23.4.5.2 IWP[0:1]/VFLS[0:1] Signals

The power-up state of IWP[0:1]/VFLS[0:1] is controlled by setting the SIUMCR[DBGC]; see Table 6-8. They can also be set via the reset configuration word (See Section 7.5.2, "Hard Reset Configuration Word (RCW)"). The FRZ state is indicated by the value 0b11 on the VFLS[0:1] signals.

23.4.5.3 VFLS[0:1]/MPIO32B[3:4] Signals

The VFLS[0:1]/MPIO32B[3:4] signals power up as the MPIO32B[3:4] function and their function can be changed via the VFLS bit in the MIOS14TPCR register. The FRZ state is indicated by the value 0b11 on the VFLS[0:1] signals.

23.4.6 Development Port Registers

The development port consists logically of the three registers: development port instruction register (DPIR), development port data register (DPDR), and trap enable control register (TECR). These registers are physically implemented as two registers, development port shift register and trap enable control register. The development port shift register acts as both the DPIR and DPDR depending on the operation being performed. It is also used as a temporary holding register for data to be stored into the TECR. These registers are discussed below in more detail.

23.4.6.1 Development Port Shift Register

The development port shift register is a 35-bit shift register. Instructions and data are shifted into it serially from DSDI using DSCK (or CLKOUT depending on the debug port clock mode, refer to Section 23.4.6.4, "Development Port Serial Communications — Clock Mode Selection") as the shift clock. These instructions or data are then transferred in parallel to the CPU, the trap enable control register (TECR). When the processor enters debug mode it fetches instructions from the DPIR which causes an access to the development port shift register. These instructions are serially loaded into the shift register from DSDI using DSCK (or CLKOUT) as the shift clock. In a similar way, data is transferred to the CPU by moving it into the shift register which the processor reads as the result of executing a "move from special purpose register DPDR" instruction. Data is also parallel-loaded into the development port shift register from the CPU by executing a "move to special purpose register DPDR" instruction. It is then shifted out serially to DSDO using DSCK (or CLKOUT) as the shift clock.

23.4.6.2 Trap Enable Control Register

The trap enable control register is a 9-bit register that is loaded from the development port shift register. The contents of the control register are used to drive the six trap enable signals, the two breakpoint signals, and the VSYNC signal to the CPU. The "transfer data to trap enable control register" commands will cause the appropriate bits to be transferred to the control register.

BDM status message (with BDM status field equal to 0b0) is sent out when the device exits BDM mode and RCPU is in normal operating mode.

The BDM status message has the following format:

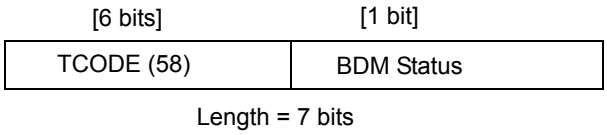

Figure 24-81. BDM Status Message Format

24.14.1.4 Error Message (Invalid Message)

An error message is sent out when an invalid message is received by READI. The error code within the error message indicates that an invalid TCODE was detected in the auxiliary input messages by the signal input formatter. Refer to Table 24-20.

The error message has the following format:

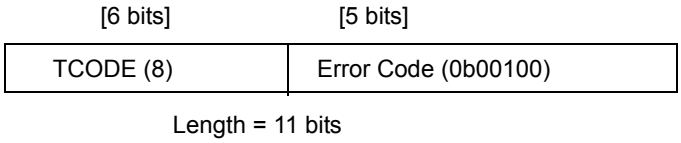

Figure 24-82. Error Message (Invalid Message) Format

24.14.2 RCPU Development Access Operation

The RCPU development access can be achieved either via the READI signals or the BDM signals.

To enable RCPU development access via the READI signals, the tool has to configure the DC register during the READI reset ($\overline{\text{RSTI}}$). Once the READI module takes the control of RCPU development access, the protocol for transmission of development serial data in (DSDI) and out (DSDO) is performed through the IEEE-ISTO 5001-1999 compliant vendor-defined messages.

After enabling RCPU development access via the READI signals, the READI module can enable debug mode and enter debug mode. When debug mode is enabled and entered, READI sends a BDM status message (BDM status field equal to 0b1) to the development tool indicating that the RCPU has entered debug mode and is now expecting instructions from the READI signals.

The development tool then uses the DSDI Data Message to send in the serial transmission data to READI. Data is transmitted to the tool using the DSDO data message.

This process continues until the RCPU exits debug mode and READI sends the BDM status message (BDM status field equal to 0b0) indicating debug mode exit.

NOTE

Only after the DSDO data message is sent out should another DSDI data message be sent in.

Synchronous self-clocked mode is selected by READI for RCPU development access. In this mode, the internal transmission between READI and the USIU is performed at system frequency.

Table B-14. TouCAN A, B and C (CAN 2.0B Controller) (continued)

Address	Access	Symbol	Register	Size	Reset
0x30 74A2	S/U	IMASK_B	TouCAN_B Interrupt Masks	16	S
0x30 74A4	S/U	IFLAG_B	TouCAN_B Interrupt Flags	16	S
0x30 74A6	S/U	RXECTR_B/ TXECTR_B	TouCAN_B Receive Error Counter/ TouCAN_B Transmit Error Counter	16	S
0x30 7500 — 0x30 750F	S/U	MBUFF0_B[1]	TouCAN_B Message Buffer 0.	—	U
0x30 7510 — 0x30 751F	S/U	MBUFF1_B[1]	TouCAN_B Message Buffer 1.	—	U
0x30 7520 — 0x30 752F	S/U	MBUFF2_B[1]	TouCAN_B Message Buffer 2.	—	U
0x30 7530 — 0x30 753F	S/U	MBUFF3_B[1]	TouCAN_B Message Buffer 3.	—	U
0x30 7540 — 0x30 754F	S/U	MBUFF4_B[1]	TouCAN_B Message Buffer 4.	—	U
0x30 7550 — 0x30 755F	S/U	MBUFF5_B[1]	TouCAN_B Message Buffer 5.	—	U
0x30 7560 — 0x30 756F	S/U	MBUFF6_B[1]	TouCAN_B Message Buffer 6.	—	U
0x30 7570 — 0x30 757F	S/U	MBUFF7_B[1]	TouCAN_B Message Buffer 7.	—	U
0x30 7580 — 0x30 758F	S/U	MBUFF8_B[1]	TouCAN_B Message Buffer 8.	—	U
0x30 7590 — 0x30 759F	S/U	MBUFF9_B[1]	TouCAN_B Message Buffer 9.	—	U
0x30 75A0 — 0x30 75AF	S/U	MBUFF10_B[1]	TouCAN_B Message Buffer 10.	—	U
0x30 75B0 — 0x30 75BF	S/U	MBUFF11_B[1]	TouCAN_B Message Buffer 11.	—	U
0x30 75C0 — 0x30 75CF	S/U	MBUFF12_B[1]	TouCAN_B Message Buffer 12.	—	U
0x30 75D0 — 0x30 75DF	S/U	MBUFF13_B[1]	TouCAN_B Message Buffer 13.	—	U
0x30 75E0 — 0x30 75EF	S/U	MBUFF14_B[1]	TouCAN_B Message Buffer 14.	—	U
0x30 75F0 — 0x30 75FF	S/U	MBUFF15_B[1]	TouCAN_B Message Buffer 15.	—	U
TouCAN_C					
0x30 7880	S	CANMCR_C	TouCAN_C Module Configuration Register	16	S
0x30 7882	T	CANTCR_C	TouCAN_C Test Register	16	S

CONTROL BITS

NAME	OPTIONS	ADDRESSES
0 1 2 3 Channel Function Select	xxxx – QOM Function Number. Assigned during microcode assembly. See Table D-1	0x30YY0C – 0x30YY12
0 1 Host Sequence	00 – Single-Shot Mode 01 – Loop Mode 10 – Continuous Mode 11 – Continuous Mode	0x30YY14 – 0x30YY16
0 1 Host Service Request	00 – No Host Service (Reset Condition) 01 – Initialize, No Pin Change 10 – Initialize, Pin Low 11 – Initialize, Pin High	0x30YY18 – 0x30YY1A
0 1 Channel Priority	00 – Disabled 01 – Low Priority 10 – Medium Priority 11 – High Priority	0x30YY1C – 0x30YY1E
0 Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled	0x30YY0A
0 Channel Interrupt Status	0 – Channel Interrupt Not Asserted 1 – Channel Interrupt Asserted	0x30YY20

PARAMETER RAM

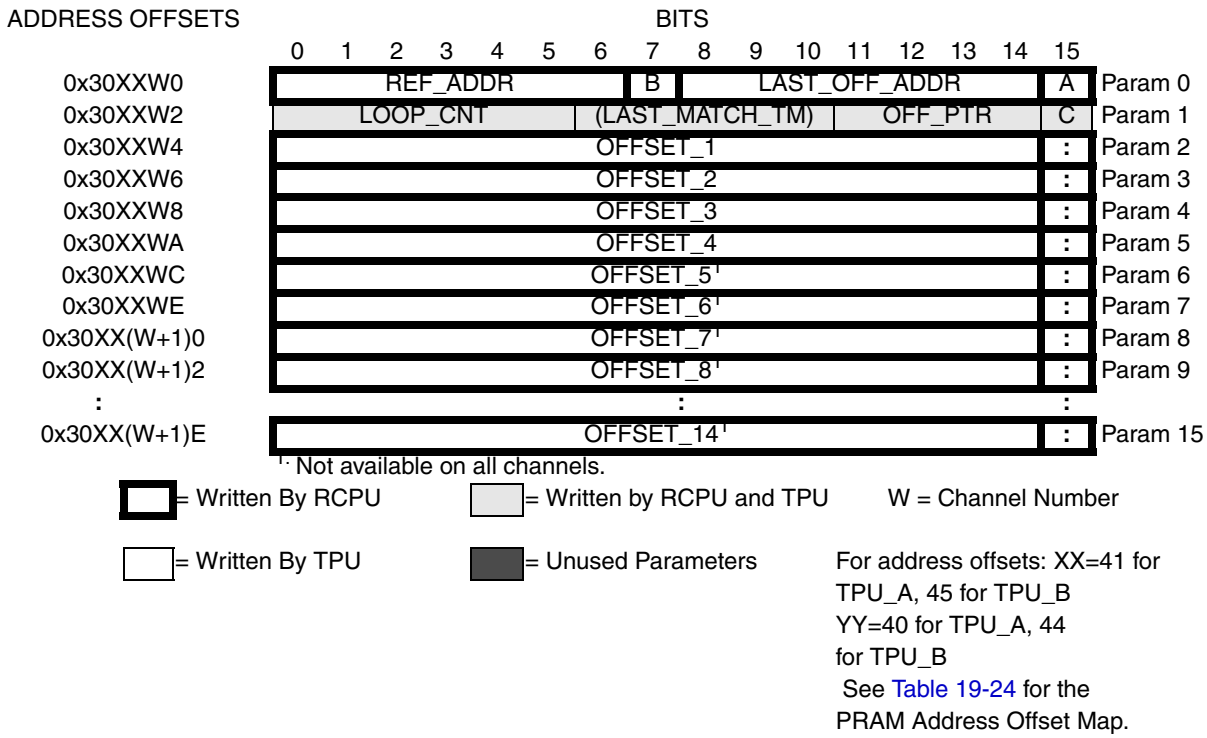

Figure D-3. QOM Parameters

Table F-10. Bus Operation Timing (continued)

Note: (V_{DD} = 2.6 V ± 0.1 V, V_{DDH} = 5.0 V ± 0.25 V, $T_A = T_L$ to T_H, 50 pF load unless noted otherwise)

	Characteristic	40 MHz		56 MHz[1]		Unit
		Min	Max	Min	Max	
8	CLKOUT to Signal Valid ADDR[8:31] RD/$\overline{WR}$ BURST D[0:31][4]	6.25	14	4.5	11	ns
8a	CLKOUT to Signal Valid TSIZ[0:1] $\overline{RSV}$ AT[0:3] $\overline{BDIP}$ PTR $\overline{RETRY}$	6.25	13	4.5	9.5	ns
8b	CLKOUT to Signal Valid[2] $\overline{BR}$ $\overline{BG}$ VFLS[0:1] VF[0:2] IWP[0:2] FRZ LWP[0:1] $\overline{STS}$ valid.	6.25	14	4.5	10.5	ns
8c	Slave Mode CLKOUT to Signal Valid D[0:31]		14		11	ns
8d	CLKOUT to Data Pre-discharge time		16		16	ns
8e	CLKOUT to Data Pre-discharge start	3		3		ns
9	CLKOUT to High Z ADDR[8:31] RD/$\overline{WR}$ BURST D[0:31] TSIZ[0:1] $\overline{RSV}$ AT[0:3] PTR $\overline{RETRY}$	6.25	13	4.5	9.5	ns
10	CLKOUT to $\overline{TS}$, $\overline{BB}$ assertion	7.25	14	5.5	10.5	ns
10a	CLKOUT to $\overline{TA}$, $\overline{BI}$ assertion (when driven by the Memory Controller)		8.5		8.5	ns

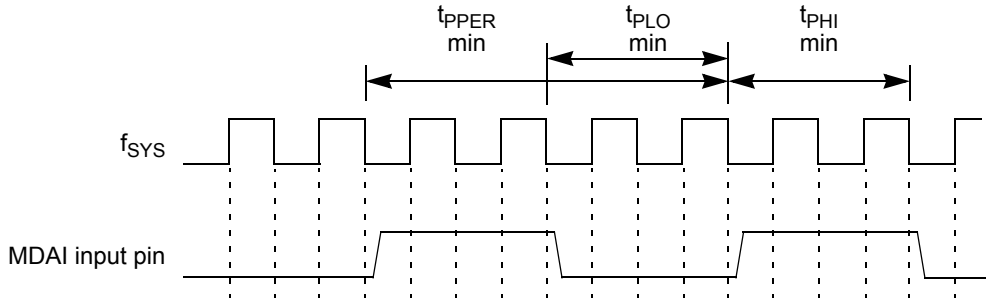

Figure F-57. MDASM Minimum Input Pin Timing Diagram

NOTE

f_{SYS} is the internal system clock for the IMB3 bus.

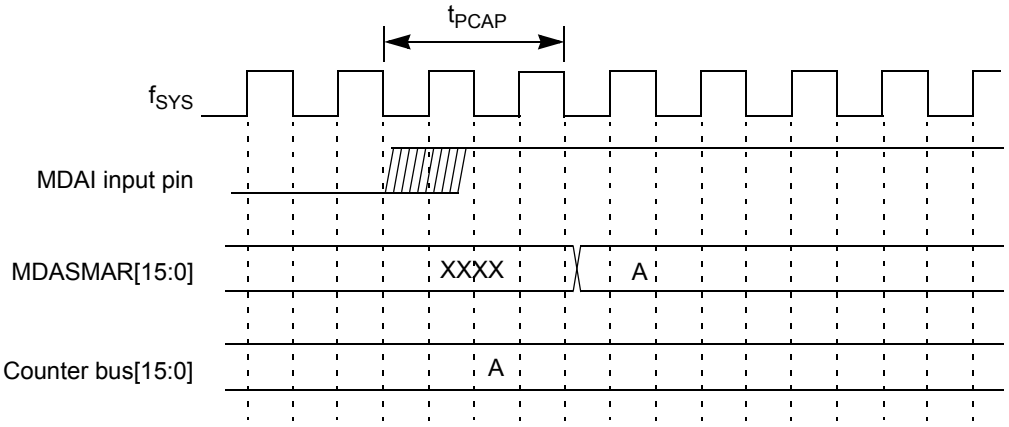

Figure F-58. MDASM Input Pin To Counter Bus Capture Timing Diagram

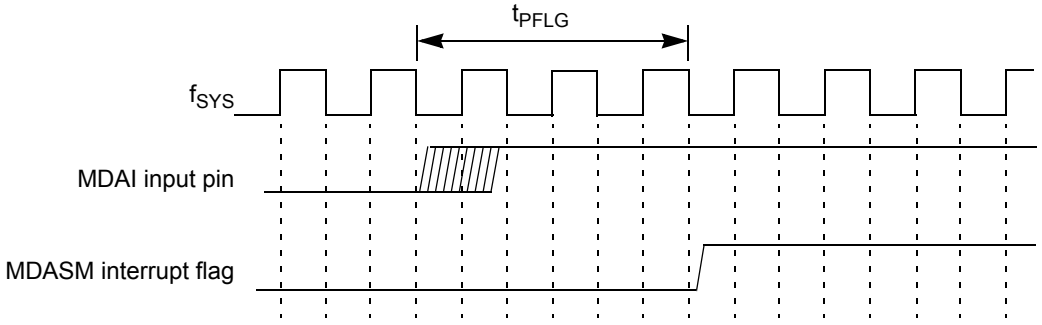

Figure F-59. MDASM Input Pin to MDASM Interrupt Flag Timing Diagram

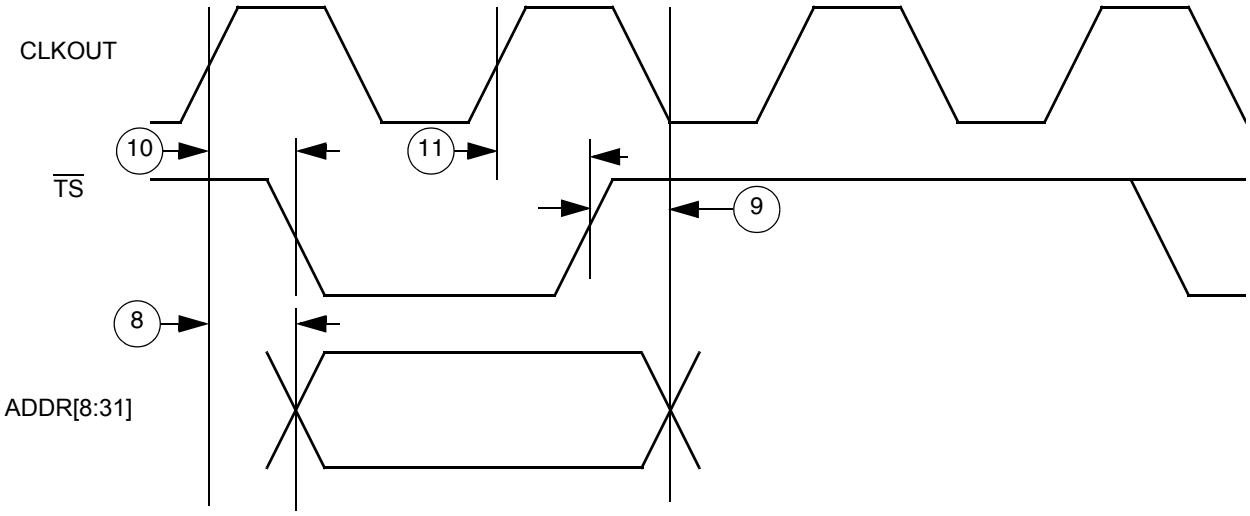

Figure G-19. Address Show Cycle Bus Timing

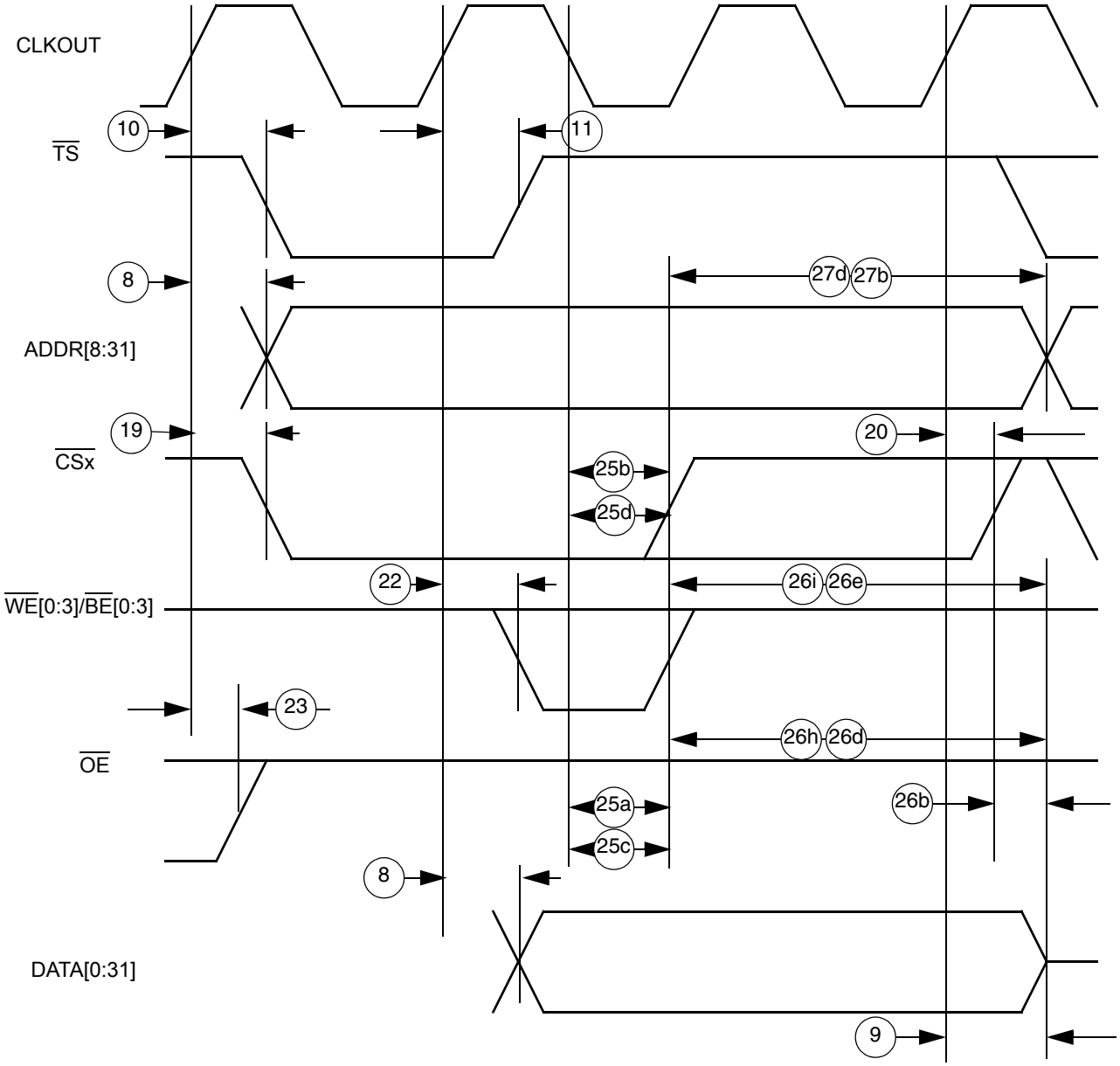

Figure G-23. External Bus Write Timing (GPCM Controlled – TRLX = '1', CSNT = '1')