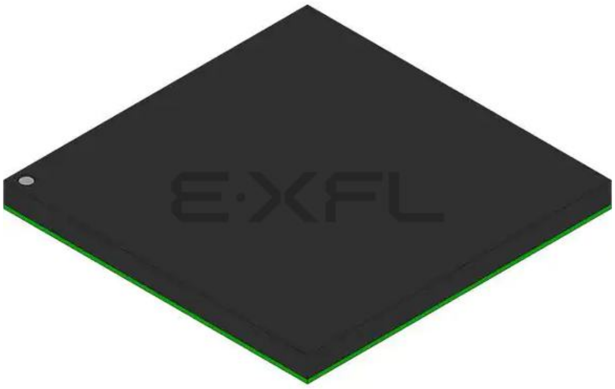

Welcome to E-XFL.COM

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	56MHz
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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	388-BBGA
Supplier Device Package	388-PBGA (27x27)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mpc564mvr56

Contents

Chapter 18
Peripheral Pin Multiplexing (PPM) Module

Chapter 19
Time Processor Unit 3

3.9.10.3 Additional Implementation-Specific Registers

Refer to the following sections for details on additional implementation-specific registers in the MPC561/MPC563:

- Section 4.6, "BBC Programming Model"
- Section 6.2.2.1.2, "Internal Memory Map Register (IMMR)"
- Section 11.8, "L2U Programming Model"
- Chapter 23, "Development Support"

3.10 Instruction Set

All PowerPC ISA instructions are encoded as single words (32 bits) and are consistent among all instruction types. The fixed instruction length and consistent format simplify instruction pipelining and permit efficient decoding to occur in parallel with operand accesses.

The PowerPC ISA instructions are divided into the following categories:

- Integer instructions, which include computational and logical instructions
  - Integer arithmetic instructions
  - Integer compare instructions
  - Integer logical instructions
  - Integer rotate and shift instructions
- Floating-point instructions, which include floating-point computational instructions, as well as instructions that affect the floating-point status and control register (FPSCR)
  - Floating-point arithmetic instructions
  - Floating-point multiply/add instructions
  - Floating-point rounding and conversion instructions
  - Floating-point compare instructions
  - Floating-point status and control instructions
- Load/store instructions., which include integer and floating-point load and store instructions
  - Integer load and store instructions
  - Integer load and store multiple instructions
  - Floating-point load and store
  - Primitives used to construct atomic memory operations (lwarx and stwcx. instructions)
- Flow control instructions, which include branching instructions, condition register logical instructions, trap instructions, and other instructions that affect the instruction flow
  - Branch and trap instructions
  - Condition register logical instructions
- Processor control instructions, which are used for synchronizing memory accesses.
  - Move to/from SPR instructions
  - Move to/from MSR

Table 3-28. Register Settings following Program Exception

Register	Bits	Setting Description
Save/Restore Register 0 (SRR0)[1]	All	Contains the effective address of the excepting instruction
Save/Restore Register 1 (SRR1)[2]	[0:10]	Cleared to 0
	11	Set for a floating-point enabled program exception; otherwise cleared.
	12	Cleared to 0.
	13	Set for a privileged instruction program exception; otherwise cleared.
	14	Set for a trap program exception; otherwise cleared.
	15	Cleared to 0 if SRR0 contains the address of the instruction causing the exception, and set if SRR0 contains the address of a subsequent instruction.
	[16:31]	Loaded from bits [16:31] of MSR. In the current implementation, bit 30 of the SRR1 is never cleared, except by loading a zero value from MSR[RI].
Machine State Register (MSR)	IP	No change
	ME	No change
	LE	Set to value of ILE bit prior to the exception
	DCMPEN	This bit is set according to (BBCMCR[EN_COMP] AND BBCMCR[EXC_COMP])
	Other	Cleared to 0

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

[2] Only one of bits 11, 13, and 14 can be set.

When a program exception is taken, instruction execution resumes at offset 0x0700 from the physical base address indicated by MSR[IP].

3.15.4.8 Floating-Point Unavailable Exception (0x0800)

A floating-point unavailable exception occurs when no higher priority exception exists, an attempt is made to execute a floating-point instruction (including floating-point load, store, and move instructions), and the floating-point available bit in the MSR is disabled, (MSR[FP] = 0).

Table 3-29. Register Settings following a Floating-Point Unavailable Exception

Register	Bits	Setting Description
Save/Restore Register 0 (SRR0)[1]	All	Set to the effective address of the instruction that caused the exception.
Save/Restore Register 1 (SRR1)	[0:15]	Cleared to 0
	[16:31]	Loaded from MSR[16:31]

The return to normal-high mode from normal-low, doze-high, low, and sleep mode is accomplished with the asynchronous interrupt. The sources of the asynchronous interrupt are:

- Asynchronous wake-up interrupt from the interrupt controller
- RTC, PIT, or time base interrupts (if enabled)
- Decrementer exception

The system responds quickly to asynchronous interrupts. The wake-up time from normal-low, doze-high, doze-low, and sleep mode caused by an asynchronous interrupt or a decrementer exception is only three to four clock cycles of maximum system frequency. In 40-MHz systems, this wake-up requires 75 to 100 ns. The asynchronous wake-up interrupt from the interrupt controller is level sensitive one. It will therefore be negated only after the reset of interrupt cause in the interrupt controller.

The timers' (RTC, PIT, time base, or decrementer) interrupts indications set status bits in the PLPRCR (TMIST). The clock module considers this interrupt to be pending asynchronous interrupt as long as the TMIST is set. The TMIST status bit should be cleared before entering any low-power mode.

Table 8-7 summarizes wake-up operation for each of the low-power modes.

Table 8-6. Power Mode Wake-Up Operation

Operation Mode	Wake-up Method	Return Time from Wake-up Event to Normal-High
Normal-low ("gear")	Software or Interrupt	Asynchronous interrupts: 3-4 maximum system cycles Synchronous interrupts: 3-4 actual system cycles
Doze-high	Interrupt	
Doze-low	Interrupt	
Sleep	Interrupt	3-4 maximum system clocks
Deep-sleep	Interrupt	< 500 Oscillator Cycles 125 µs – 4 MHz 25 µs – 20 MHz
Power-down	Interrupt	< 500 oscillator cycles + power supply wake-up
IRAMSTBY	External	Power-on sequence

8.7.3.1 Exiting from Normal-Low Mode

In normal mode (as well as doze mode), if the PLPRCR[CSRC] bit is set, the system toggles between low frequency (defined by PLPRCR[DFNL]) and high frequency (defined by PLPRCR[DFNH]. The system switches from normal-low mode to normal-high mode if either of the following conditions is met:

- An interrupt is pending from the interrupt controller; or
- The MSR[POW] bit is cleared (power management is disabled).

When neither of these conditions are met, the PLPRCR[CSRC] bit is set, and the asynchronous interrupt status bits are reset, the system returns to normal-low mode.

8.11.4 IRAMSTBY Control Register (VSRMCR)

This register contains control bits for enabling or disabling the IRAMSTBY supply detection circuit. There are also four bits that indicate the failure detection. All four bits have the same function and are required to improve the detection capability in extreme cases.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	—	LVSRS				VSRDE[1]	LVDRS	ZOREG	—							
PORESET		Unaffected				0	U	0_0000_0000								
Addr	0x2F C290															

U = Unaffected by reset

Figure 8-19. IRAMSTBY Control Register (VSRMCR)

[1] This bit is reserved on mask sets which implement bit 7 (ZOREG)

Table 8-13. VSRMCR Bit Descriptions

Bits	Name	Description
0	—	Reserved
1:4	LVSRS	Loss of IRAMSTBY sticky. These status bits indicate whether a IRAMSTBY supply failure occurred. In addition, when the power is turned on for the first time, IRAMSTBY rises and these bits are set. The LVSRS bits are cleared by writing them to ones. A write of zero has no effect on these bits. 0 No IRAMSTBY supply failure was detected 1 IRAMSTBY supply failure was detected
5	VSRDE[1]	IRAMSTBY detector disable. 0 IRAMSTBY detection circuit is enabled 1 IRAMSTBY detection circuit is disabled
6	LVDRS	Loss of IRAMSTBY for DECRAM Sticky — The status bit, dedicated especially for the BBC DECRAM, which indicates if there was IRAMSTBY supply failure. When the power is turned on for the first time, IRAMSTBY rises also and the bits will be asserted. The LVDECRAM bit can be cleared by writing ones to LVDECRAM. A write of zero has no effect on this bit. The bit may be used by application software, to decide if there is need to load decompression vocabularies during reset routine. 0 IRAMSTBY supply failure was not detected 1 IRAMSTBY supply failure was detected NOTE: The LVDRS bit is provided as a convenience for indicating that the DECRAM has lost power. It requires that the IRAMSTBY pins are connected to the same power supply. It actually only monitors the IRAMSTBY supply.
7	ZOREG[2]	This bit indicates the status of the internal IRAMSTBY supply. This bit is cleared by writing a 1 to it. 0 Internal IRAMSTBY zener regulator has not gone out of regulation 1 Internal IRAMSTBY zener regulator has gone out of regulation. **Note:** ZOREG may get set inadvertently if IRAMSTBY is not supplied with at least 150µA.
8:15	—	Reserved

[1] Removed on all parts that have the ZOREG bit.

Table 9-4. Priority Between Internal and External Masters over External Bus[1]

Type	Direction	Priority
Parked access[2]	Internal → external	0
Instruction access	Internal → external	3
Data access	Internal → external	4
External access	external → external/internal	EARP (could be programmed to 0 – 7)

[1] External master will be granted external bus ownership if EARP is greater than the internal access priority.

[2] Parked access is instruction or data access from the RCPU which is initiated on the internal bus without requesting it first in order to improve performance.

Figure 9-27 illustrates the internal finite-state machine that implements the arbiter protocol.

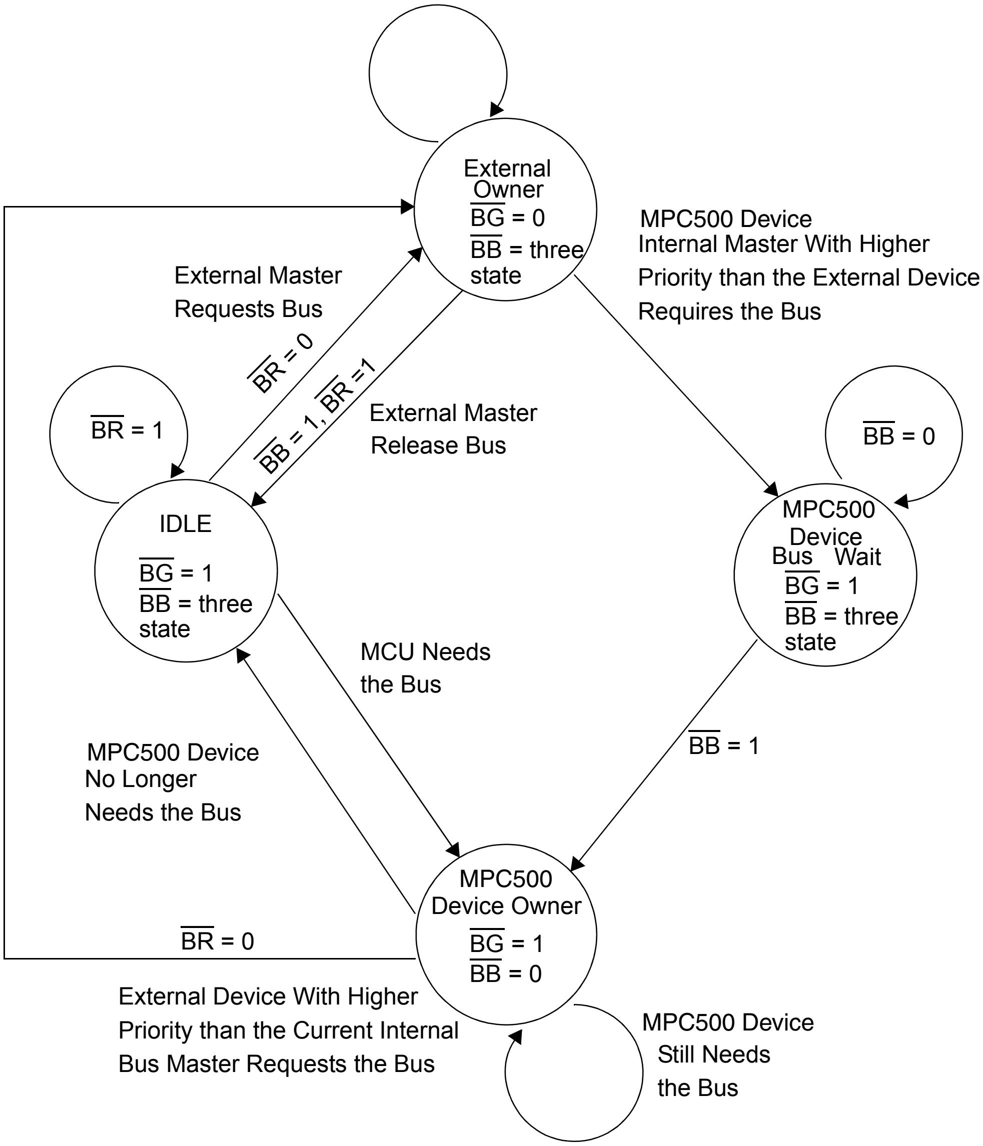
External Owner
BG = 0
BB = three state
MPC500 Device Internal Master With Higher Priority than the External Device Requires the Bus
External Master Requests Bus
BR = 0
BB = 1, BR =1
External Master Release Bus
BR = 1
BB = 0
IDLE
BG = 1
BB = three state
MPC500 Device Bus Wait
BG = 1
BB = three state
MCU Needs the Bus
BB = 1
MPC500 Device No Longer Needs the Bus
MPC500 Device Owner
BG = 1
BB = 0
BR = 0
External Device With Higher Priority than the Current Internal Bus Master Requests the Bus
MPC500 Device Still Needs the Bus

Figure 9-27. Internal Bus Arbitration State Machine

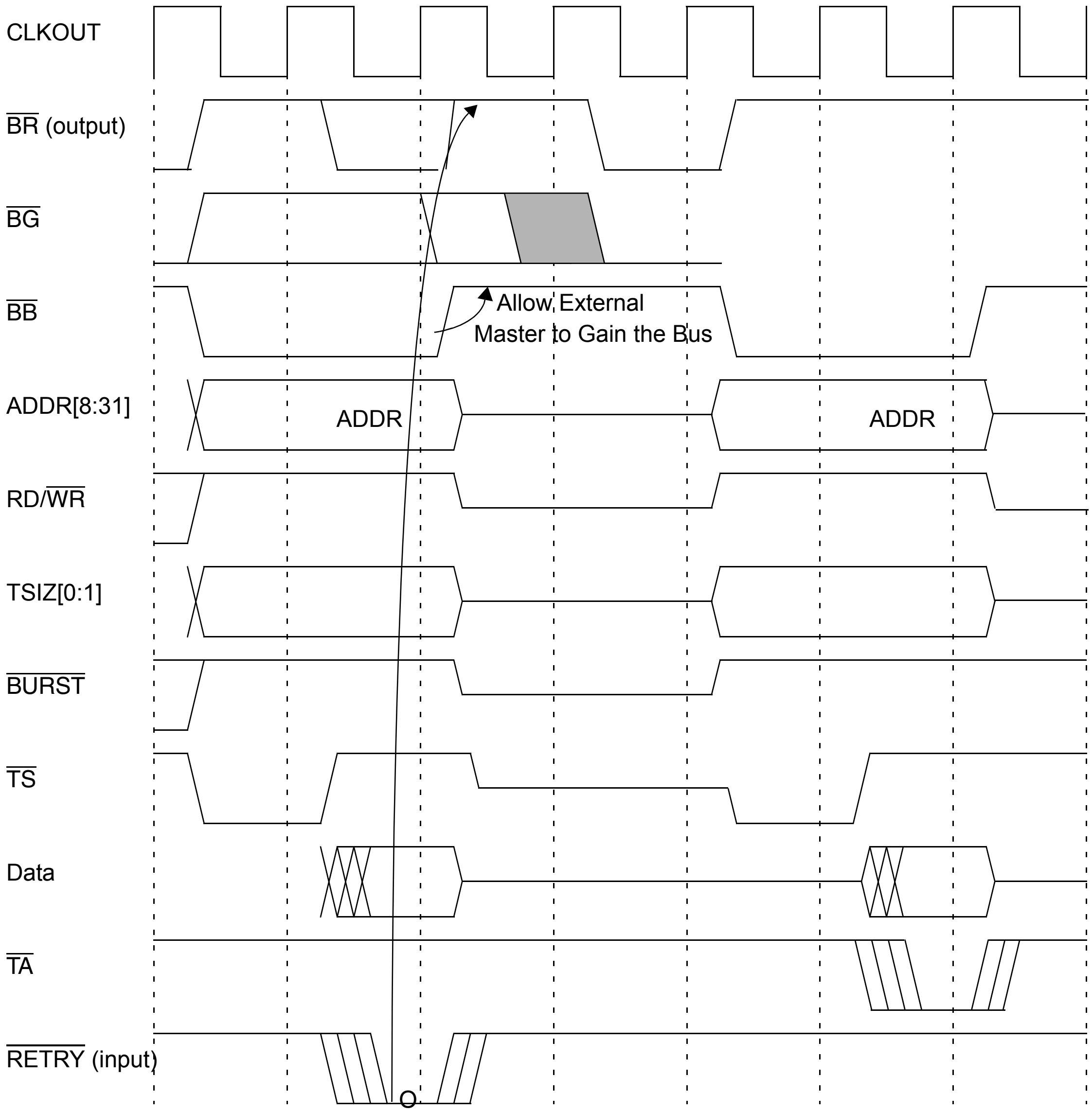

Figure 9-33. Retry Transfer Timing – External Arbiter

When the MPC561/MPC563 initiates a burst access, the bus interface recognizes the RETRY assertion as a retry termination only if it detects it before the first data beat was acknowledged by the slave device. When the RETRY signal is asserted as a termination signal on any data beat of the access after the first (being the first data beat acknowledged by a normal TA assertion), the MPC561/MPC563 recognizes RETRY as a transfer error acknowledge.

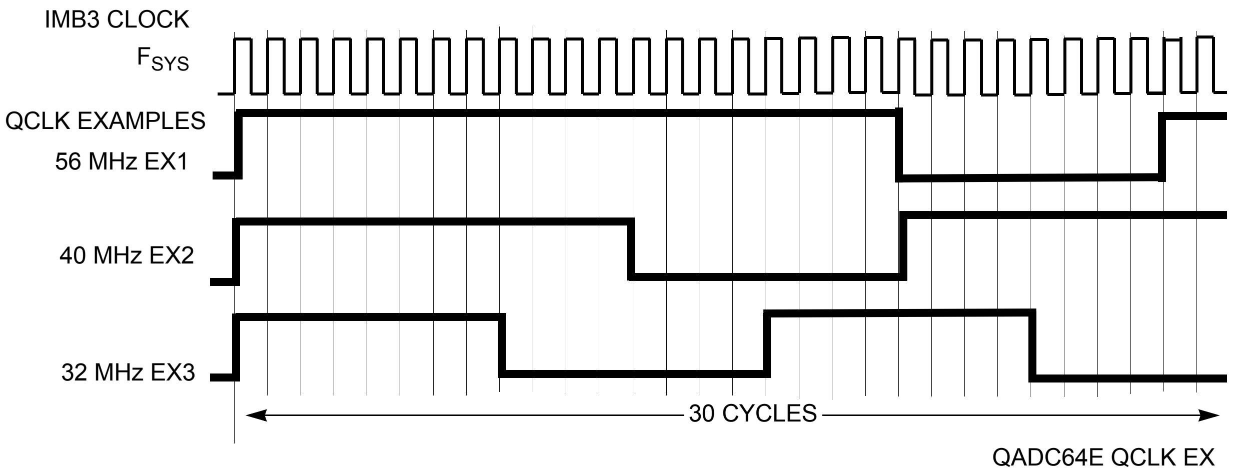

Figure 13-25. QADC64E Clock Programmability Examples

Table 13-21. QADC64E Clock Programmability

Control Register 0 Information					Input Sample Time (IST) =0b00	
Example Number	**Frequency**	**PSH**	**PSA**	**PSL**	**QCLK (MHz)**	**Conversion Time (μs)**
1	56 MHz	19	0	7	2.0	7.0
2	40 MHz	11	0	7	2.0	7.0
3	32 MHz	7	0	7	2.0	7.0

NOTE

PSA is maintained for software compatibility but has no functional benefit to this version of the module.

The MCU IMB3 clock frequency is the basis of the QADC64E timing. The QADC64E requires that the IMB3 clock frequency be at least twice the QCLK frequency. The QCLK frequency is established by the combination of the PSH and PSL parameters in QACR0. The 5-bit PSH field selects the number of IMB3 clock cycles in the high phase of the QCLK wave. The 3-bit PSL field selects the number of IMB3 clock cycles in the low phase of the QCLK wave.

Example 1 in Table 13-21 shows that when the PSH = 19, the QCLK remains high for 20 cycles if the IMB3 clock and with PSL = 7 the QCLK remains low for 8 IMB3 clock cycles. Example 2 shows that when PSH = 11, QCLK is high for 12 IMB3 clock cycles and with PSL = 7, QCLK is low for 8 IMB3 clock cycles. Finally, example 3 shows that with PSH = 7 and PSL = 7, QCLK alternates between high and low every 8 IMB3 cycles.

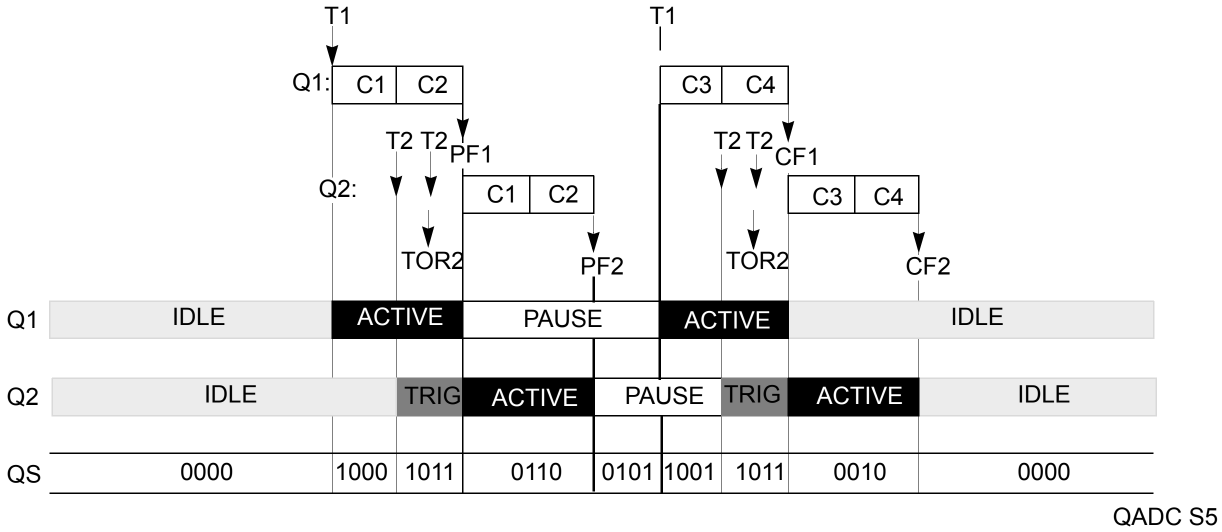

Figure 13-31. CCW Priority Situation 5

The remaining situations, S6 through S11, show the impact of a queue 1 trigger event occurring during queue 2 execution. Queue 1 is higher in priority the conversion taking place in queue 2 is aborted, so that there is not a variable latency time in responding to queue 1 trigger events.

In situation S6 (Figure 13-32), the conversion initiated by the second CCW in queue 2 is aborted just before the conversion is complete, so that queue 1 execution can begin. Queue 2 is considered suspended. After queue 1 is finished, queue 2 starts over with the first CCW, when the RES (resume) control bit is set to 0. Situation S7 (Figure 13-33) shows that when pause operation is not in use with queue 2, queue 2 suspension works the same way.

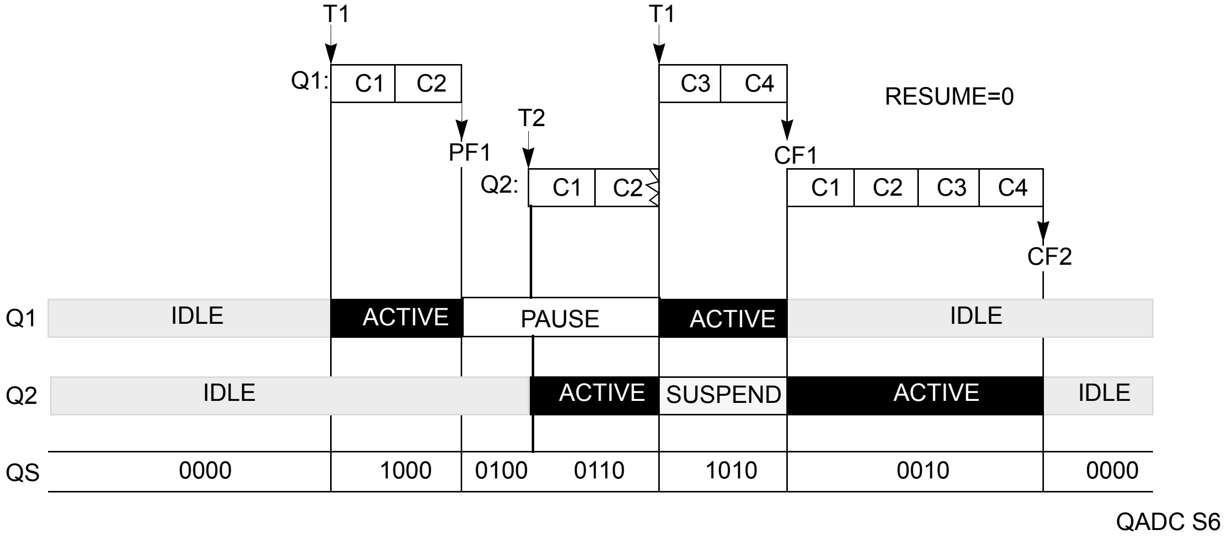

Figure 13-32. CCW Priority Situation 6

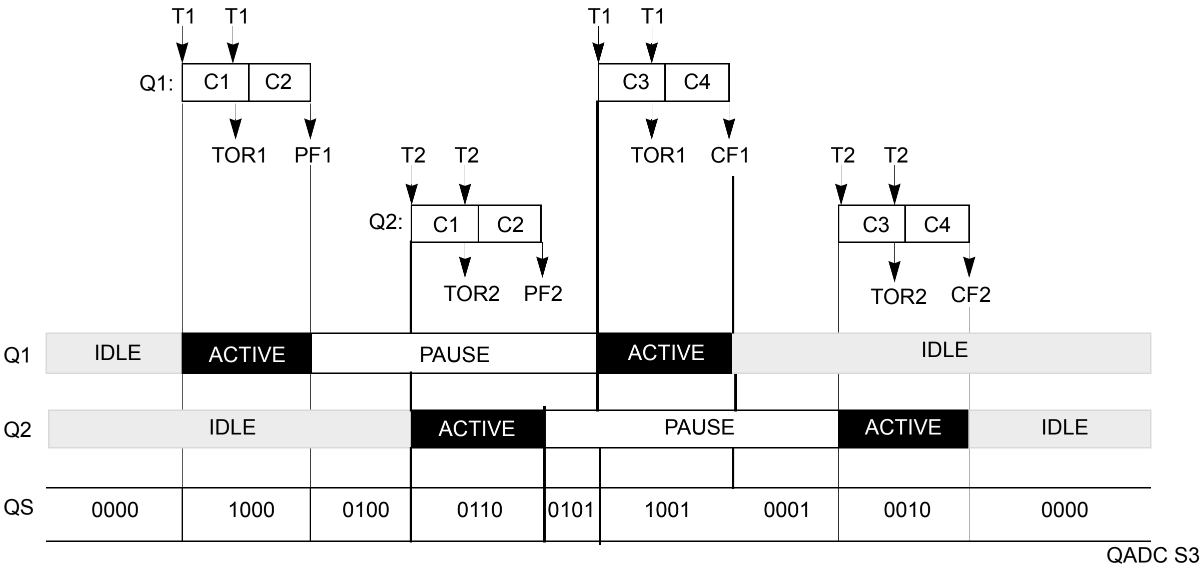

Figure 14-27. CCW Priority Situation 3

The next two situations consider trigger events that occur for the lower priority queue 2, while queue 1 is actively being serviced.

Situation S4 (Figure 14-28) shows that a queue 2 trigger event that is recognized while queue 1 is active is saved, and as soon as queue 1 is finished, queue 2 servicing begins.

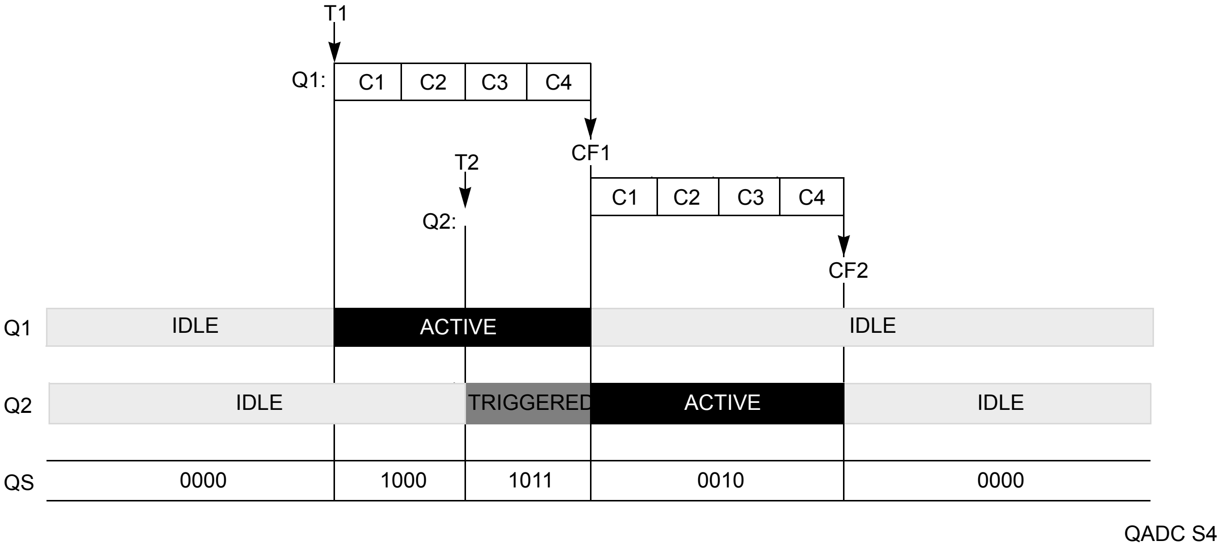

Figure 14-28. CCW Priority Situation 4

Situation S5 (Figure 14-29) shows that when multiple queue 2 trigger events are detected while queue 1 is busy, the trigger overrun error bit is set, but queue 1 execution is not disturbed. Situation S5 also shows that the effect of queue 2 trigger events during queue 1 execution is the same when the pause feature is in use in either queue.

Table 16-13. CANCTRL0 Bit Descriptions

Bits	Name	Description
0	BOFFMSK	Bus off interrupt mask. The BOFF MASK bit provides a mask for the bus off interrupt. 0 Bus off interrupt disabled 1 Bus off interrupt enabled
1	ERRMSK	Error interrupt mask. The ERRMSK bit provides a mask for the error interrupt. 0 Error interrupt disabled 1 Error interrupt enabled
2:3	—	Reserved
4:5	RXMODE	Receive signal configuration control. These bits control the configuration of the CNRX0 signals. Refer to Table 16-14.
6:7	TXMODE	Transmit signal configuration control. This bit field controls the configuration of the CNTX0 signals. Refer to Table 16-15.
8:15	CANCTRL1	See Table 16-16 and Section 16.7.5, “Control Register 1 (CANCTRL1).”

Table 16-14. Rx MODE[1:0] Configuration

Signal	RX1	RX0	Receive Signal Configuration
CNRX0	X	0	0 CNRX0 signal is interpreted as a dominant bit 1 CNRX0 signal is interpreted as a recessive bit
	X	1	0 CNRX0 signal is interpreted as a recessive bit 1 CNRX0 signal is interpreted as a dominant bit

Table 16-15. Transmit Signal Configuration

TXMODE[1:0]	TransmitSignal Configuration
00	Full CMOS[1]; positive polarity (CNTX0 = 0 is a dominant level)
01	Full CMOS[1]; negative polarity (CNTX0 = 1 is a dominant level)
1X	Open drain[2]; positive polarity

[1] Full CMOS drive indicates that both dominant and recessive levels are driven by the chip.

[2] Open drain drive indicates that only a dominant level is driven by the chip. During a recessive level, the CNTX0 signal is disabled (three stated), and the electrical level is achieved by external pull-up/pull-down devices. The assertion of both Tx mode bits causes the polarity inversion to be cancelled (open drain mode forces the polarity to be positive).

16.7.5 Control Register 1 (CANCTRL1)

MSB … LSB

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	CANCTRL0								SAMP	—	TSYNC	LBUF	—	PROPSEG		
$\overline{\text{SRESET}}$	0000_0000_0000_0000															
Addr	0x30 7086 (CANCTRL1_A); 0x30 7486 (CANCTRL1_B); 0x30 7886 (CANCTRL1_C)															

Figure 16-12. Control Register 1 (CANCTRL1)

Table 18-2. PPMMCR Bit Descriptions

Bits	Name	Description
0	STOP	Stop Mode Enable. When the STOP bit is set and the PPM enters Stop Mode, the PPM module clocks will be stopped. The PPM will only respond to accesses to the PPMMCR register. The STOP bit can only be set when the PPM is disabled, (i.e., PPMPCR[ENTX] = 0 *and* PPMPCR[ENRX] = 0). Writing to the STOP bit while either TX or RX is enabled will result in a TEA (bus error access). 0 PPM clocks enabled 1 PPM clock disabled – PPM in Stop Mode.
1:7	—	Reserved
8	SUPV	Supervisor/User Data Space. The SUPV bit places the PPM registers in either Supervisor or User Data Space. 0 Access to PPMMCR, TX_CONFIG1/2, RX_CONFIG1/2 is restricted to supervisor-only. Access to all other PPM registers is unrestricted. 1 All PPM registers are accessible in supervisor-only data space.
9:15	—	Reserved

18.4.1.1 Entering Stop Mode

The PPM module cannot, and should not, be put into stop mode while either the transmit or receive operation is enabled in PPMPCR[ENTX] and/or PPMPCR[ENRX]. Furthermore, it should not be put into stop mode if it is operating in continuous mode. In this case it should be switched to single transfer mode first.

The following steps should be taken to ensure that stop mode is entered safely and without loss of data:

1. If the PPM is operating in continous mode
   - switch to single transfer mode by clearing PPMPCR[CM]=0.
2. If the PPM is enabled for transmit or receive
   - set PPMPCR[STR] (TDM or SPI mode)
   - Disable both PPMPCR[ENTX] and PPMPCR[ENRX]
   - Wait until PPMPCR[STR] is cleared by the PPM module. This will be done when the next data frame has been sent or received.
3. If the PPM is not enabled for transmit or receive
   - Clear PPMPCR[STR] if necessary
4. Set PPMMCR[STOP]

   When PPMMCR[STOP] is set, the PPM module enters stop mode and the PPM module clocks will be stopped. While in stop mode, none of the PPM registers will be accessible, except for the PPMPCR register.

   If the STOP bit is clear, stop mode is disabled.

Table 21-5. UC3FCTL Bit Descriptions (continued)

Bits	Name	Description
30	SES	Start-end program or erase sequence. The SES bit is write protected by the STOP, HVS, and EHV bits. The SES bit is used to signal the start and end of a program or erase sequence. At the start of a program or erase sequence, SES is set. This will lock STOP, PROTECT, SBPROTECT, BLOCK, SBBLOCK, SBEN, CSC, and PE. If PE = 0 and SES = 1, SIE will be write locked. At this point, the UC3F EEPROM is ready to receive either the programming writes or the erase interlock write. NOTE: The erase interlock write is a write to any UC3F EEPROM array location after SES is set and PE = 1. If PE = 0 and SES = 1, writes to the UC3F array are programming writes. The first programming write sets the address of the location to be programmed, and the data written is captured into the program data latch for programming into the UC3F array. All programming writes after the first programming write update the program data latch but do not change the address to be programmed. At the end of the program or erase operation, the SES bit must be cleared to return to normal operation and release the STOP, PROTECT, SBPROTECT, BLOCK, SBBLOCK, CSC, SBEN, and PE bits. 0 UC3F EEPROM not configured for program or erase operation 1 Configure UC3F EEPROM for program or erase operation
31	EHV	Enable high voltage. EHV can be asserted only after the SES bit has been asserted and a valid programming write(s) or erase hardware interlock write has occurred. If an attempt is made to assert EHV when SES is negated, or if a valid programming write or erase hardware interlock write has not occurred since SES was asserted, EHV will remain negated. The external program or erase enable pin (EPEE) and EHV are used to control the application of the program or erase voltage to the UC3F EEPROM module. High voltage operations to the UC3F EEPROM array, special shadow locations or FLASH NVM registers can occur only if EHV = 1 and EPEE = 1. Only after the correct hardware and software interlocks have been applied to the UC3F EEPROM can EHV be set. Once EHV is set, SES cannot be changed and attempts to read the array will not be acknowledged. Clearing EHV during a program or erase operation will safely terminate the high voltage operation. If EHV is cleared while using the embedded hardware program/erase algorithm, the program/erase routine will abort the operation and exit normally. 0 Program or erase pulse disabled 1 Program or erase pulse enabled

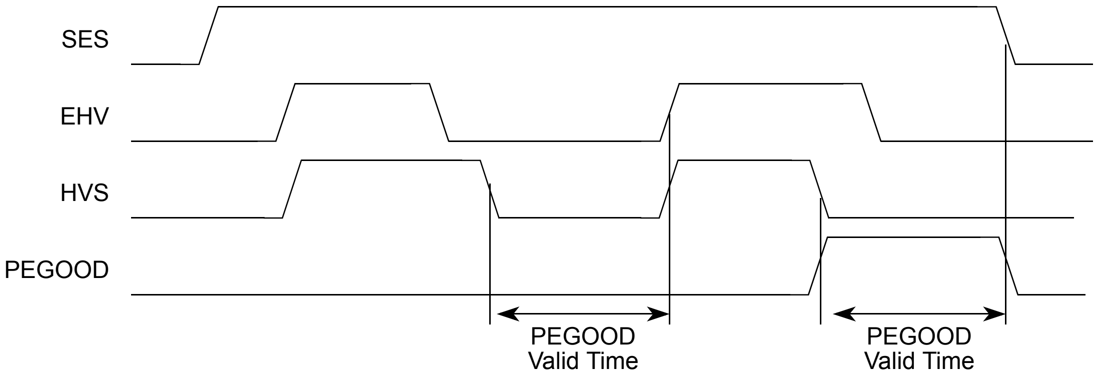

Figure 21-5. PEGOOD Valid Time

11. Negate VSYNC
12. Return to the regular code run (issue an rfi). The first report on the VF pins is a VSYNC (VF = 011)
13. The external hardware stops sampling the program trace information upon the report on the VF pins of VSYNC

23.1.4.2 Detecting the Trace Window Start Address

When using back trace, latching the value of the status pins (VF and VFLS), and the address of the cycles marked as program trace cycle, should start immediately after the negation of reset. The start address is the first address in the program trace cycle buffer.

When using window trace, latching the value of the status pins (VF and VFLS), and the address of the cycles marked as program trace cycle, should start immediately after the first VSYNC is reported on the VF pins. The start address of the trace window should be calculated according to first two VF pins reports.

Assuming that VF1 and VF2 are the two first VF pins reports and T1 and T2 are the two addresses of the first two cycles marked with the program trace cycle attribute that were latched in the trace buffer, use the following table to calculate the trace window start address.

Table 23-4. Detecting the Trace Buffer Start Point

VF1	VF2	Starting point	Description
011 VSYNC	001 sequential	T1	VSYNC asserted followed by a sequential instruction. The start address is T1
011 VSYNC	110 branch direct taken	T1 - 4 + offset (T1 - 4)	VSYNC asserted followed by a taken direct branch. The start address is the target of the direct branch
011 VSYNC	101 branch indirect taken	T2	VSYNC asserted followed by a taken indirect branch. The start address is the target of the indirect branch

23.1.4.3 Detecting the Assertion/Negation of VSYNC

Since the VF pins are used for reporting both instruction type information and queue flush information, the external hardware must take special care when trying to detect the assertion/negation of VSYNC. When VF = 011 it is a VSYNC assertion/negation report only if the previous VF pins value was one of the following values: 000, 001, or 010.

23.1.4.4 Detecting the Trace Window End Address

The information on the status pins that describes the last fetched instruction and the last queue/history buffer flushes, changes every clock. Cycles marked as program trace cycle are generated on the external bus only when possible (when the SIU wins the arbitration over the external bus). Therefore, there is some delay between the information reported on the status pins that a cycle marked as program trace cycle will be performed on the external bus and the actual time that this cycle can be detected on the external bus.

When VSYNC is negated (through the serial interface of the development port), the CPU delays the report of the of the assertion/negation of VSYNC on the VF pins (VF = 011) until all addresses marked with the program trace cycle attribute were visible externally. Therefore, the external hardware should stop

Table 23-24. LCTRL1 Bit Descriptions (continued)

Bits	Mnemonic	Description	Function
16:17	CSG	Compare size, comparator G	00 reserved 01 word 10 half word 11 byte (Must be programmed to word for floating point compares)
18:19	CSH	Compare size, comparator H	
20	SUSG	Signed/unsigned operating mode for comparator G	0 unsigned 1 signed (Must be programmed to signed for floating point compares)
21	SUSH	Signed/unsigned operating mode for comparator H	
22:25	CGBMSK	Byte mask for 1st L-data comparator	0000 all bytes are not masked 0001 the last byte of the word is masked . . . 1111 all bytes are masked
26:29	CHBMSK	Byte mask for 2nd L-data comparator	
30:31	—	Reserved	—

Note: LCTRL1 is cleared following reset.

23.6.10 L-Bus Support Control Register 2

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	LW0EN	LW0IA		LW0 IADC	LW0LA		LW0 LADC	LW0LD		LW0 LDDC	LW1 EN	LW1IA		LW1 IADC	LW1LA	
SRESET	0000_0000_0000_0000															

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	LSB 31
Field	LW1 LADC	LW1LD		LW1 LDDC	BRK NOMSK	—							DLW0 EN	DLW1 EN	SLW0 EN	SLW1 EN
SRESET	0000_0000_0000_0000															
Addr	SPR 157															

Figure 23-23. L-Bus Support Control Register 2 (LCTRL2)

Table 24-19. Public Messages Supported (continued)

Message Name	Minimum Packet Size (bits)	Maximum Packet Size (bits)	Packet Type	Packet Description	Direction
Program Trace — Direct Branch Message	6	6	Fixed	TCODE number = 3	From Device
	1	8	Variable	number of sequential instructions executed since last taken branch	
Program Trace — Indirect Branch Message	6	6	Fixed	TCODE number = 4	From Device
	1	8	Variable	number of sequential instructions executed since last taken branch	
	1	23	Variable	unique portion of the target address for taken branches and exceptions	
Data Trace — Data Write Message	6	6	Fixed	TCODE number = 5	From Device
	1	25	Variable	unique portion of the data write address	
	8	32	Variable	data write value (8, 16, 32 bits)	
Data Trace — Data Read Message	6	6	Fixed	TCODE number = 6	From Device
	1	25	Variable	unique portion of the data read address	
	8	32	Variable	data read value (8, 16, 32 bits)	
Error Message[1]	6	6	Fixed	TCODE number = 8	From Device
	5	5	Fixed	error code	
Program Trace Correction Message	6	6	Fixed	TCODE number = 10 (0xA)	From Device
	1	8	Variable	correcting the number of instructions in the trace	
Program Trace — Direct Branch Synchronization Message (PTSM = 0)	6	6	Fixed	TCODE number = 11 (0xB)	From Device
	1	1	Variable	number of program trace messages cancelled	
	1	23	Variable	full target address	
Program Trace — Direct Branch Synchronization[2] Message (PTSM = 1)	6	6	Fixed	TCODE number = 11 (0xB)	From Device
	1	8	Variable	number of sequential instructions executed since last taken branch	
	1	23	Variable	full target address	
Program Trace — Indirect Branch Synchronization Message (PTSM = 0)	6	6	Fixed	TCODE number = 12 (0xC)	From Device
	1	1	Variable	number of program trace messages cancelled	
	1	23	Variable	full target address	

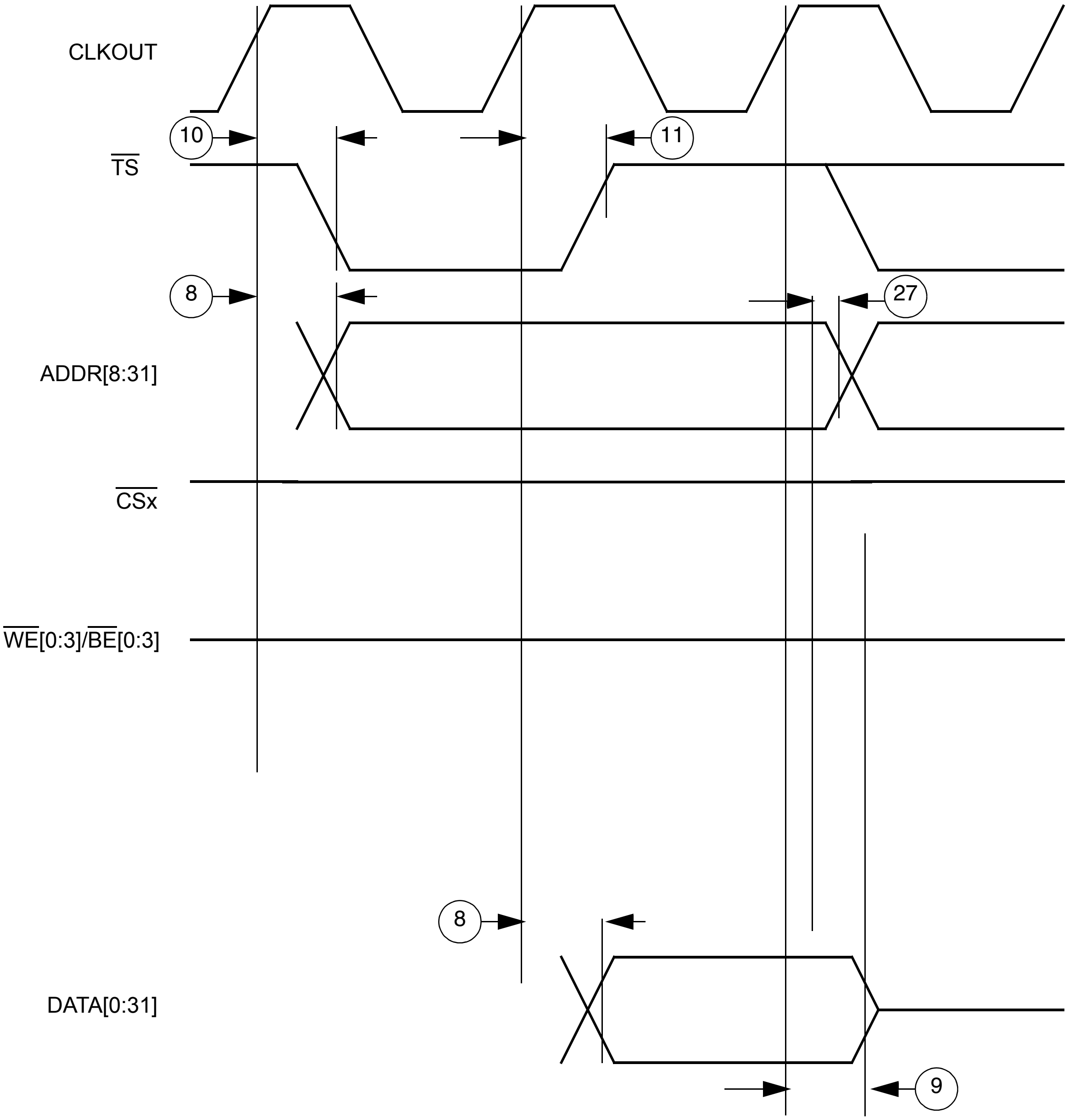

Figure F-21. Address and Data Show Cycle Bus Timing

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

Signal Name	Pin Name	Ball Assignment
VSS	vss	A19, A2, A23, A24, A26, A3, A4, AA1, AA2, AA23, AA24, AA25, AA3, AB1, AB2, AB24, AB25, AB4, AC1, AC21, AC23, AC25, AC3, AC5, AD2, AD22, AD24, AD4, AD5, AE1, AE22, AE23, AE25, AE3, AE4, AE5, AF2, AF22, AF23, AF24, AF26, AF3, AF4, AF6, B1, B19, B23, B25, B3, B4, C1, C2, C24, C26, C4, D1, D2, D23, D25, D26, D3, D5, E2, E24, E25, E26, E3, E4, L11, L12, L13, L14, L15, L16, M11, M12, M13, M14, M15, M16, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, R11, R12, R13, R14, R15, R16, T11, T12, T13, T14, T15, T16, V25
KAPWR	kapwr	W26
IRAMSTBY	IRAMSTBY	M3
QVDDL	qvddl	AA4
		AB23
		AB3
		AC2
		AC24
		AD1
		AD25
		AD6
		AE26
		AE6
		B26
		C25
		D24
		E23
		F4
USIU Power Supplies		
VDDSYN	vddsyn	Y26
VSSSYN	vsssyn	AB26
QADC64E Power Supplies		
VRH	vrh	C10
VRL	vrl	A10

G.12 READI Electrical Characteristics

The AC electrical characteristics (56 MHz) are described in the following tables and figures

Table G-13. READI AC Electrical Characteristics

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)

Number	Characteristic	Min	Max	Unit
1	MCKO Cycle Time (Tco)	17.9	—	ns
2	MCKO Duty Cycle	40	60	%
3	Output Rise and Fall Times	0	3	ns
4	MCKO low to MDO Data Valid	-1.79	3.58	ns
5	MCKI Cycle Time (Tci)	35.6	—	ns
6	MCKI Duty Cycle	40	60	%
7	Input Rise and Fall Times	0	3	ns
8	MDI, $\overline{\text{EVTI}}$, $\overline{\text{MSEI}}$ Setup Time	7.12	—	ns
9	MDI Hold TIme	3.56	—	ns
10	$\overline{\text{RSTI}}$ Pulse Width	71.6	—	ns
11	MCKO low to $\overline{\text{MSEO}}$ Valid	-1.79	3.58	ns
12	$\overline{\text{EVTI}}$ Pulse Width	71.6	—	ns
13	$\overline{\text{EVTI}}$ to $\overline{\text{RSTI}}$ Setup (at reset only)	(4.0) x TC	—	ns
14	$\overline{\text{EVTI}}$ to $\overline{\text{RSTI}}$ Hold (at reset only)	(4.0) x TC	—	ns

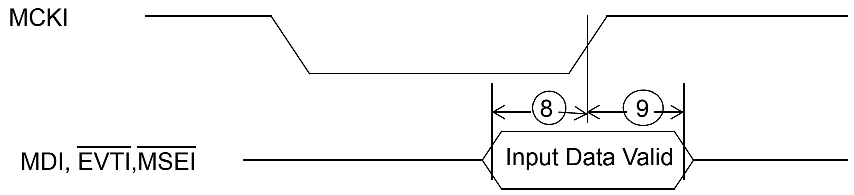

Figure G-29. Auxiliary Port Data Input Timing Diagram

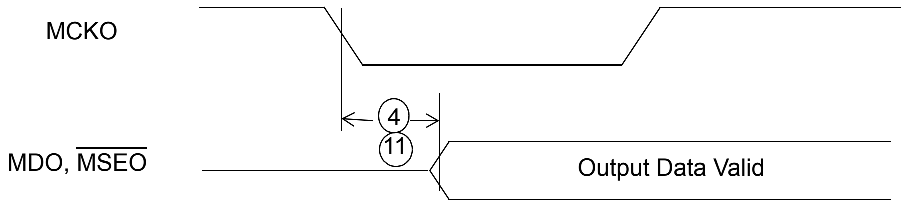

Figure G-30. Auxiliary Port Data Output Timing Diagram

MDO and $\overline{\text{MSEO}}$ data is held valid until the next MCKO low transition.

When $\overline{\text{RSTI}}$ is asserted, $\overline{\text{EVTI}}$ is used to enable or disable the auxiliary port. Because MCKO probably is not active at this point, the timing must be based on the system clock. Since the system clock is not realized on the connector, its value must be known by the tool.