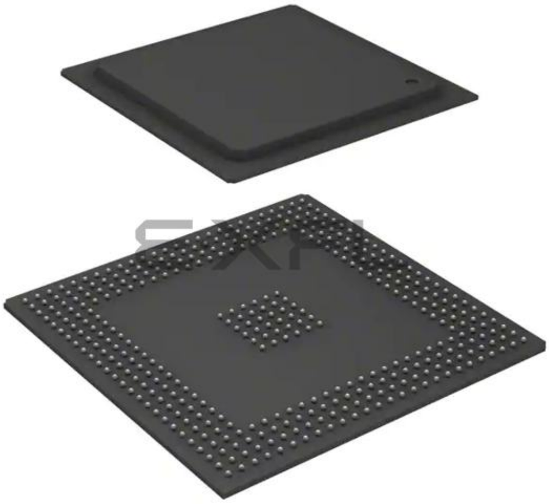
NXP USA Inc. - MPC561CVR40 Datasheet



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	40MHz
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 ~ 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/mpc561cvr40

1.3.3.5 Queued Serial Multi-Channel Module (QSMCM)

- One queued serial module with one queued SPI and two SCIs (QSMCM)
- QSMCM matches full MPC555 QSMCM functionality
- Queued SPI
  - — Provides full-duplex communication port for peripheral expansion or inter-processor communication
  - — Up to 32 preprogrammed transfers, reducing overhead
  - — Synchronous serial interface with baud rate of up to system clock / 4
  - — Four programmable peripheral-selects signals:
  - — Supports up to 16 devices with external decoding
  - — Supports up to eight devices with internal decoding
  - — Special wrap-around mode allows continuous sampling of a serial peripheral for efficient interfacing to serial analog-to-digital (A/D) converters
- SCI
  - — UART mode provides NRZ format and half- or full-duplex interface
  - — 16 register receive buffers and 16 register transmit buffers on one SCI
  - — Advanced error detection and optional parity generation and detection
  - — Word-length programmable as eight or nine bits
  - — Separate transmitter and receiver enable bits, and double buffering of data
  - — Wake-up functions allow the CPU to run uninterrupted until either a true idle line is detected, or a new address byte is received

1.3.3.6 Peripheral Pin Multiplexing (PPM)

- Synchronous serial interface between the microprocessor and an external device
- Four internal parallel data sources can be multiplexed through the PPM
  - — TPU3_A: 16 channels
  - — TPU3_B: 16 channels
  - — MIOS14: 12 PWM channels, four MDA channels
  - — Internal GPIO: 16 general-purpose inputs, 16 general-purpose outputs
- Software configurable stream size
- Software configurable clock (TCLK) based on system clock
- Software selectable clock modes (SPI mode and TDM mode)
- Software selectable operation modes
  - — Continuous mode
  - — Start-transmit-receive (STR) mode
- Software configurable internal modules interconnect (shorting)

Chapter 2
Signal Descriptions

This chapter describes the MPC561/MPC563 microcontroller's external signals. It contains a description of individual signals, shows their behavior, shows whether the signal is an input or an output, and indicates signal multiplexing.

NOTE

> A bar over a signal name indicates that the signal is active-low—for example, $\overline{\text{TA}}$ (transfer acknowledge). Active-low signals are referred to as asserted (active) when they are low and negated when they are high. Signals that are not active-low, such as ADDR[8:31] (address bus signals) and DATA[0:31] (data bus signals) are referred to as asserted when they are high and negated when they are low.

Refer to Appendix F, "Electrical Characteristics," and Appendix G, "66-MHz Electrical Characteristics," for detailed electrical information for each signal.

2.1 Signal Groupings

Figure 2-1 illustrates the external signals of the MPC561/MPC563 grouped by functional module.

to enable the pull resistors in the pads. On negation of reset, based on which functionality is selected for the signals, this signal is set to disable the pull resistors, or is continued to be held in its reset state to indicate that the pulls are disabled only when the output driver is enabled.

2.6.4.4 Pull Device Select (PULL_SEL)

The MIOS14 and the TPU signals have selectable pull-up or pull-down devices. The devices are controlled by the PULL_SEL signal. A high on the PULL_SEL signal enables pull-up devices on the MIOS14 and TPU signals. A low enables pull-down devices. Note that the pull devices can be disabled by the PULL_DIS0 (MIOS14) and PRDS (TPU) bits in the PDMCR register. See Section 2.3, “Pad Module Configuration Register (PDMCR).”

2.6.5 Signal Reset States

Table 2-14 summarizes the reset states of all signals on the MPC561/MPC563. Note that PD refers to a weak pull-down, PU2.6 refers to a weak pull-up to 2.6 V, and PU5 refers to a weak pull-up to 5 V. All control of the weak-pull devices is in the pad module configuration register, described in Table 2-5.

NOTE

2.6-V inputs are 5-V tolerant, but 2.6-V outputs are not. Do not connect 2.6-V outputs to a driver or pull-up greater than 3.1 V.

NOTE

Depending on the application, pins may require a pull-down resistor to avoid getting any command due to noise.

Table 5-1. USIU Address Map

Address	Register
0x2F C000	USIU Module Configuration Register (SIUMCR) See Table 6-7 for bit descriptions.
0x2F C004	System Protection Control Register (SYPCR) See Table 6-15 for bit descriptions.
0x2F C008	Reserved
0x2F C00E[1]	Software Service Register (SWSR) See Table 6-16 for bit descriptions.
0x2F C010	Interrupt Pending Register (SIPEND).
0x2F C014	Interrupt Mask Register (SIMASK) See Section 6.2.2.2.4, “SIU Interrupt Mask Register (SIMASK),” for bit descriptions.
0x2F C018	Interrupt Edge Level Mask (SIEL) See Section 6.2.2.2.7, “SIU Interrupt Edge Level Register (SIEL),” for bit descriptions.
0x2F C01C	Interrupt Vector (SIVEC) See Section 6.2.2.2.8, “SIU Interrupt Vector Register (SIVEC),” for bit descriptions.
0x2F C020	Transfer Error Status Register (TESR) See Table 6-17 for bit descriptions.
0x2F C024	USIU General-Purpose I/O Data Register (SGPIODT1) See Table 6-23 for bit descriptions.
0x2F C028	USIU General-Purpose I/O Data Register 2 (SGPIODT2) See Table 6-24 for bit descriptions.
0x2F C02C	USIU General-Purpose I/O Control Register (SGPIOCR) See Table 6-25 for bit descriptions.
0x2F C030	External Master Mode Control Register (EMCR) See Table 6-13 for bit descriptions.
0x2F C038	Pads Module Configuration Register 2 (PDMCR2) See Table 2-6 for bit descriptions.
0x2F C03C	Pads Module Configuration Register (PDMCR) See Table 2-5 for bit descriptions.
0x2F C040	Interrupt Pend2 Register (SIPEND2) See Section 6.2.2.2.2, “SIU Interrupt Pending Register 2 (SIPEND2),” for bit descriptions.
0x2F C044	Interrupt Pend3 Register (SIPEND3) See Section 6.2.2.2.3, “SIU Interrupt Pending Register 3 (SIPEND3),” for bit descriptions.
0x2F C048	Interrupt Mask2 Register (SIMASK2) See Section 6.2.2.2.5, “SIU Interrupt Mask Register 2 (SIMASK2),” for details.
0x2F C04C	Interrupt Mask3 Register (SIMASK3) See Section 6.2.2.2.6, “SIU Interrupt Mask Register 3 (SIMASK3),” for details.
0x2F C050	Interrupt In-Service2 Register (SISR2) See Section 6.2.2.2.9, “Interrupt In-Service Registers (SISR2 and SISR3),” for details.
0x2F C054	Interrupt In-Service3 Register (SISR3) See Section 6.2.2.2.9, “Interrupt In-Service Registers (SISR2 and SISR3),” for details.

6.2.2.2.9 Interrupt In-Service Registers (SISR2 and SISR3)

SISR2, SISR3 are 32-bit read/write registers. Each bit in the register corresponds to an interrupt request. A bit is set if:

- There is a pending interrupt request (SIPEND2/3), that is not masked by (SIMASK2/3), and
- The BBC/IMPU acknowledges interrupt request and latches SIVEC value.

Once a bit is set, all requests with lower or equal priority become masked (i.e. they will not generate any interrupt request to the RCPU) until the bit is cleared. A bit is cleared by writing a '1' to it. Writing zero has no effect.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	IRQ0	LVL0	IMB IRQ0	IMB IRQ1	IMB IRQ2	IMB IRQ3	IRQ1	LVL1	IMB IRQ4	IMB IRQ5	IMB IRQ6	IMB IRQ7	IRQ2	LVL2	IMB IRQ8	IMB IRQ9
$\overline{\text{SRESET}}$	0000_0000_0000_0000															
Addr	0x2F C050															

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	LSB 31
Field	IMB IRQ10	IMB IRQ11	IRQ3	LVL4	IMB IRQ12	IMB IRQ13	IMB IRQ14	IMB IRQ15	IRQ4	LVL4	IMB IRQ16	IMB IRQ17	IMB IRQ18	IMB IRQ19	IRQ5	LVL5
$\overline{\text{SRESET}}$	0000_0000_0000_0000															

Figure 6-24. Interrupt In-Service Register 2 (SISR2)

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	IMB IRQ20	IMB IRQ21	IMB IRQ22	IMB IRQ23	IRQ 6	LVL 6	IMB IRQ24	IMB IRQ25	IMB IRQ26	IMB IRQ27	IRQ 7	LVL 7	IMB IRQ28	IMB IRQ29	IMB IRQ30	IMB IRQ31
$\overline{\text{SRESET}}$	0000_0000_0000_0000															
Addr	0x2F C054															

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	LSB 31
Field	—															
$\overline{\text{SRESET}}$	0000_0000_0000_0000															

Figure 6-25. Interrupt In-Service Register 3 (SISR3)

6.2.2.3 System Protection Registers

6.2.2.3.1 System Protection Control Register (SYPCR)

The system protection control register (SYPCR) controls the system monitors, the software watchdog period, and the bus monitor timing. This register can be read at any time, but can be written only once after system reset.

	MSB															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	SWTC															
HRESET	1111_1111_1111_1111															
Addr	0x2F C004															

																LSB
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Field	BMT								BME	—			SWF	SWE	SWRI	SWP
HRESET	1111_1111								0	000			0	1	1	1

Figure 6-26. System Protection Control Register (SYPCR)

Table 6-15. SYPCR Bit Descriptions

Bits	Name	Description
0:15	SWTC	Software watchdog timer count. This field contains the count value of the software watchdog timer.
16:23	BMT	Bus monitor timing. This field specifies the time-out period, in eight-system-clock resolution, of the bus monitor. BMT must be set to non zero even if the bus monitor is not enabled.
24	BME	Bus monitor enable 0 Disable bus monitor 1 Enable bus monitor
25:27	—	Reserved
28	SWF	Software watchdog freeze 0 Software watchdog continues to run while FREEZE is asserted 1 Software watchdog stops while FREEZE is asserted
29	SWE	Software watchdog enable. Software should clear this bit after a system reset to disable the software watchdog timer. 0 Watchdog is disabled 1 Watchdog is enabled
30	SWRI	Software watchdog reset/interrupt select 0 Software watchdog time-out causes a non-maskable interrupt to the RCPU 1 Software watchdog time-out causes a system reset
31	SWP	Software watchdog prescale 0 Software watchdog timer is not prescaled 1 Software watchdog timer is prescaled by 2048

6.2.2.3.2 Software Service Register (SWSR)

The SWSR is the location to which the SWT servicing sequence is written. To prevent SWT time-out, a 0x556C followed by 0xAA39 should be written to this register. The SWSR can be written at any time but returns all zeros when read.

use the MPC561/MPC563 CLKOUT signal. This source of reset can be optionally asserted if the LOLRE bit in the PLL, low-power, and reset control register (PLPRCR) is set. The enabled PLL loss of lock event generates an internal hard reset sequence. Refer to Chapter 8, "Clocks and Power Control," for more information on loss of PLL lock.

7.1.5 On-Chip Clock Switch

If the system clock is switched to the backup clock or switched from backup clock to another clock source an internal hard reset sequence is generated. Refer to Chapter 8, "Clocks and Power Control."

7.1.6 Software Watchdog Reset

When the MPC561/MPC563 software watchdog counts to zero, a software watchdog reset is asserted. The enabled software watchdog event generates an internal hard reset sequence.

7.1.7 Checkstop Reset

When the RCPU enters a checkstop state, and the checkstop reset is enabled (the CSR bit in the PLPRCR is set), a checkstop reset is asserted. The enabled checkstop event generates an internal hard reset sequence. Refer to the *RCPU Reference Manual* for more information.

7.1.8 Debug Port Hard Reset

When the development port receives a hard reset request from the development tool, an internal hard reset sequence is generated. In this case the development tool must reconfigure the debug port. Refer to Chapter 23, "Development Support," for more information.

7.1.9 Debug Port Soft Reset

When the development port receives a soft reset request from the development tool, an internal soft reset sequence is generated. In this case the development tool must reconfigure the debug port. Refer to Chapter 23, "Development Support," for more information.

7.1.10 JTAG Reset

When the JTAG logic asserts the JTAG soft reset signal, an internal soft reset sequence is generated. Refer to Chapter 25, "IEEE 1149.1-Compliant Interface (JTAG)," for more information.

7.1.11 ILBC Illegal Bit Change

When locked bits in the PLPRCR register are changed, an internal hard reset sequence is generated. Refer to Chapter 8, "Clocks and Power Control."

7.2 Reset Actions Summary

Table 7-1 summarizes the action taken for each reset.

The MPC561/MPC563 also supports burst-inhibited transfers for slave devices that are unable to support bursting. For this type of bus cycle, the selected slave device supplies or samples the first word the MPC561/MPC563 points to and asserts the burst-inhibit signal with $\overline{\text{TA}}$ for the first transfer of the burst access. The MPC561/MPC563 responds by terminating the burst and accessing the remainder of the 16-byte block. These remaining accesses use up to three read/write bus cycles (each one for a word) in the case of a 32-bit port width slave, up to seven read/write bus cycles in the case of a 16-bit port width slave, or up to fifteen read/write bus cycles in the case of a 8-bit port width slave.

The general case of burst transfers assumes that the external memory has a 32-bit port size. The MPC561/MPC563 provides an effective mechanism for interfacing with 16-bit and 8-bit port size memories, allowing bursts transfers to these devices when they are controlled by the internal memory controller.

In this case, the MPC561/MPC563 attempts to initiate a burst transfer as in the normal case. If the memory controller signals to the bus interface that the external device has a small port size (8 or 16 bits), and if the burst is accepted, the bus interface completes a burst of 16 or 8 beats respectively for four words. Eight words requires 32 or 16 beats. Each beat of the burst transfers only one or two bytes effectively. Note that this burst of 8 or 16 beats is considered an atomic transaction, so the MPC561/MPC563 does not allow other unrelated master accesses or bus arbitration to intervene between the transfers.

9.5.5 Burst Mechanism

In addition to the standard bus signals, the MPC561/MPC563 burst mechanism uses the following signals:

- The $\overline{\text{BURST}}$ signal indicates that the cycle is a burst cycle.
- The burst data in progress ($\overline{\text{BDIP}}$) signal indicates the duration of the burst data.
- The burst inhibit ($\overline{\text{BI}}$) signal indicates whether the slave is burstable.

At the start of the burst transfer, the master drives the address, the address attributes, and the $\overline{\text{BURST}}$ signal to indicate that a burst transfer is being initiated, and asserts $\overline{\text{TS}}$. If the slave is burstable, it negates the burst-inhibit ($\overline{\text{BI}}$) signal. If the slave cannot burst, it asserts $\overline{\text{BI}}$. For additional details, refer to Section 10.2.5, “Burst Support.”

During the data phase of a burst-write cycle, the master drives the data. It also asserts $\overline{\text{BDIP}}$ if it intends to drive the data beat following the current data beat. When the slave has received the data, it asserts $\overline{\text{TA}}$ to indicate to the master that it is ready for the next data transfer. The master again drives the next data and asserts or negates the $\overline{\text{BDIP}}$ signal. If the master does not intend to drive another data beat following the current one, it negates $\overline{\text{BDIP}}$ to indicate to the slave that the next data beat transfer is the last data of the burst-write transfer.

$\overline{\text{BDIP}}$ has two basic timings: normal and late (see Figure 9-14 and Figure 9-15). In the late timing mode, assertion of $\overline{\text{BDIP}}$ is delayed by the number of wait states in the first data beat. This implies that for zero-wait-state cycles, $\overline{\text{BDIP}}$ assertion time is identical in normal and late modes. Cycles with late $\overline{\text{BDIP}}$ generation can occur only during cycles for which the memory controller generates $\overline{\text{TA}}$ internally. Refer to Chapter 10, “Memory Controller” for more information.

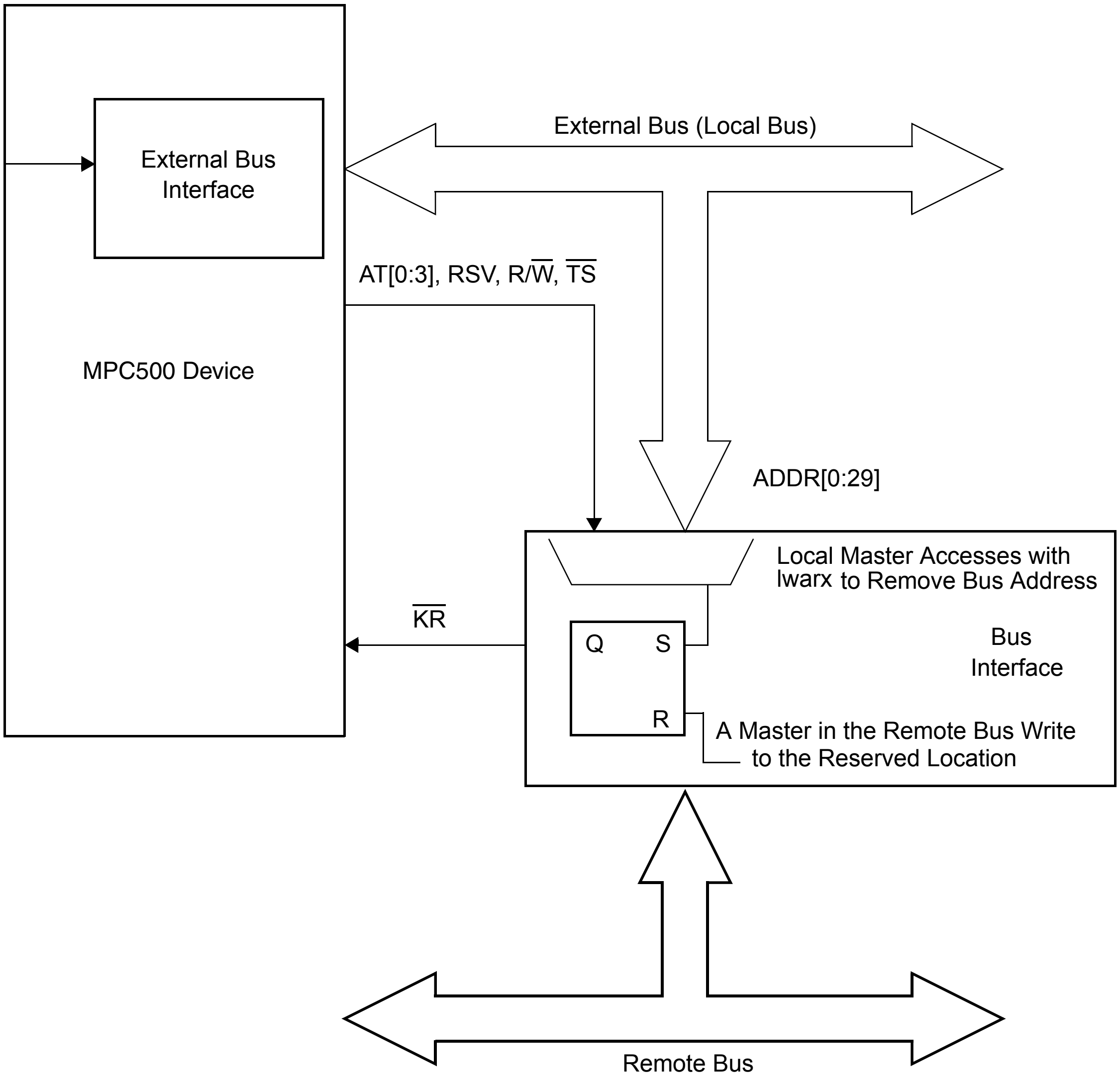


Figure 9-31. Reservation on Multi-level Bus Hierarchy

In this case, the bus interface block implements a reservation flag for the local bus master. The reservation flag is set by the bus interface when a load with reservation is issued by the local bus master and the reservation address is located on the remote bus. The flag is reset (negated) when an alternative master on the remote bus accesses the same location in a write cycle. If the MPC561/MPC563 begins a memory cycle to the previously reserved address (located in the remote bus) as a result of an stwcx instruction, the following two cases can occur:

- If the reservation flag is set, the buses interface acknowledges the cycle in a normal way
- If the reservation flag is reset, the bus interface should assert the KR. However, the bus interface should not perform the remote bus write-access or abort it if the remote bus supports aborted cycles. In this case the failure of the stwcx instruction is reported to the RCPU.

register fields can be read or written but reserved fields read zero and writes have no effect. Typically, they are written once when software initializes the QADC64E and are not changed afterwards.

	MSB															LSB
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	EMUX	—		TRG	—			PSH					PSA	PSL		
$\overline{\text{SRESET}}$	0	00		0	000			0_0001					0	011		
Addr	0x30 480A (QACR0_A); 0x30 4C0A (QACR0_B)															

Figure 13-9. Control Register 0 (QACR0)

Table 13-9. QACR0 Bit Descriptions

Bits	Name	Description
0	EMUX	Externally multiplexed mode. The EMUX bit configures the QADC64E for externally multiplexed mode, which affects the interpretation of the channel numbers and forces the MA[2:0] signals to be outputs. See Table 13-7 for more information. 0 Internally multiplexed, 16 possible channels 1 Externally multiplexed, 41 possible channels
1:2	—	Reserved
3	TRG	Trigger assignment. TRG allows the software to assign the ETRIG[2:1] signals to queue 1 and queue 2. 0 ETRIG1 triggers queue 1; ETRIG2 triggers queue 2 1 ETRIG1 triggers queue 2; ETRIG2 triggers queue 1 Refer to Section 13.7.2, "External Trigger Input Signals."
4:6	—	Reserved
7:11	PSH	Prescaler clock high time. The PSH field selects the QCLK high time in the prescaler. PSH value plus 1 represents the high time in IMB3 clocks
12	PSA	Note that this bit location is maintained for software compatibility with previous versions of the QADC64E. It serves no functional benefit in the MPC561/MPC563 and is not operational.
13:15	PSL	Prescaler clock low time. The PSL field selects the QCLK low time in the prescaler. PSL value plus 1 represents the low time in IMB3 clocks

NOTE

Details of how to calculate values for PSH, PSA, and PSL, as well as examples, are given in Section 13.5.5, "QADC64E Clock (QCLK) Generation."

13.3.6 Control Register 1 (QACR1)

Control register 1 is the mode control register for the operation of queue 1. The application software defines the queue operating mode for the queue, and may enable a completion and/or pause interrupt. All of the control register fields are read/write data. However, the SSE1 bit always reads as zero. Most of the bits are typically written once when the software initializes the QADC64E, and not changed afterwards.

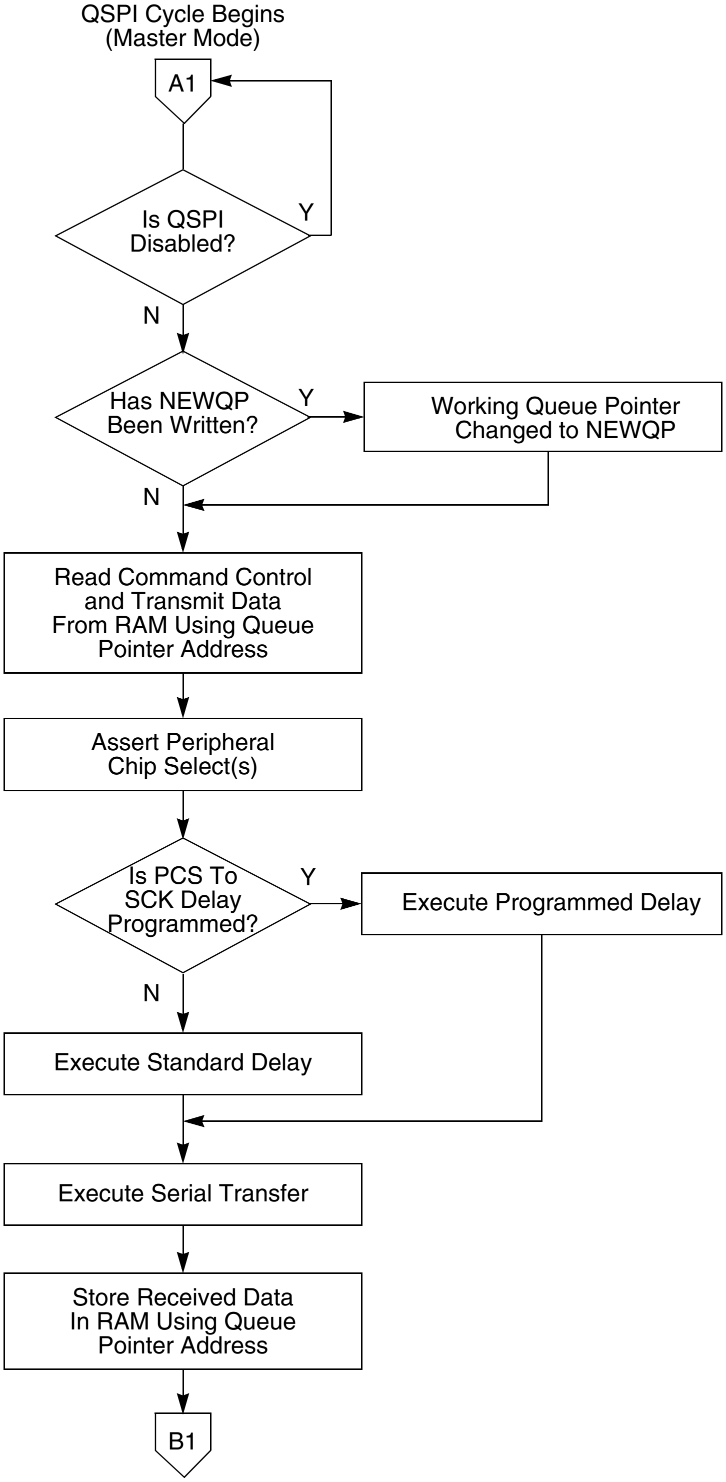


Figure 15-19. Flowchart of QSPI Master Operation (Part 1)

The operation of the receiver bit processor is shown in Figure 15-30. This example demonstrates the search for a valid start bit and the synchronization procedure as outlined above. The possibilities of noise durations greater than one bit-time are not considered in this examples.

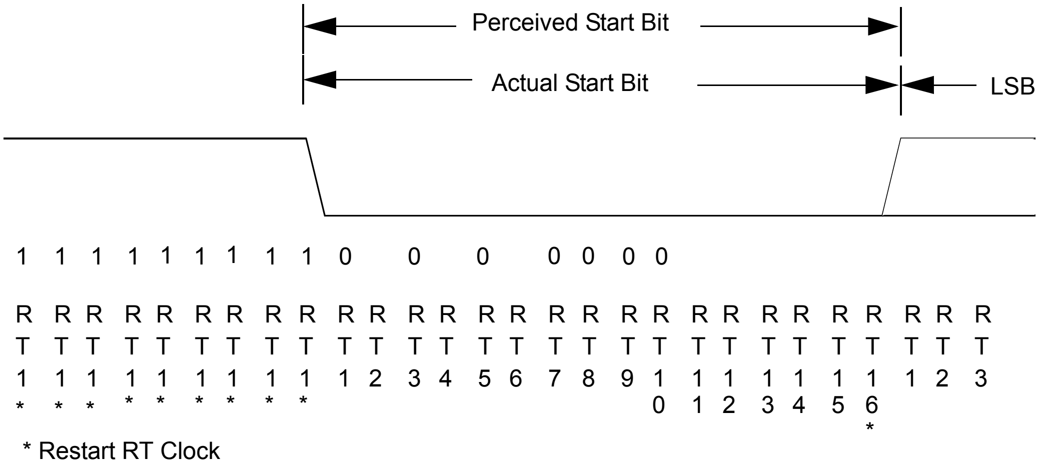

Perceived Start Bit
Actual Start Bit
LSB
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
RT1* RT1* RT1* RT1* RT1* RT1* RT1* RT1* RT1* RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16* RT1 RT2 RT3
* Restart RT Clock

Figure 15-30. Start Search Example

15.7.7.8 Receiver Functional Operation

The RE bit in SCCxR1 enables (RE = 1) and disables (RE = 0) the receiver. The receiver contains a receive serial shifter and a parallel receive data register (RDRx) located in the SCI data register (SCxDR). The serial shifter cannot be directly accessed by the CPU. The receiver is double-buffered, allowing data to be held in the RDRx while other data is shifted in.

Receiver bit processor logic drives a state machine that determines the logic level for each bit-time. This state machine controls when the bit processor logic is to sample the RXD pin and also controls when data is to be passed to the receive serial shifter. A receive time clock is used to control sampling and synchronization. Data is shifted into the receive serial shifter according to the most recent synchronization of the receive time clock with the incoming data stream. From this point on, data movement is synchronized with the MCU IMB3 clock. Operation of the receiver state machine is detailed in the *Queued Serial Module Reference Manual*.

The number of bits shifted in by the receiver depends on the serial format. However, all frames must end with at least one stop bit. When the stop bit is received, the frame is considered to be complete, and the received data in the serial shifter is transferred to the RDRx. The receiver data register flag (RDRF) is set when the data is transferred.

The stop bit is always a logic one. If a logic zero is sensed during this bit-time, the FE flag in SCxSR is set. A framing error is usually caused by mismatched baud rates between the receiver and transmitter or by a significant burst of noise. Note that a framing error is not always detected; the data in the expected stop bit-time may happen to be a logic one.

Noise errors, parity errors, and framing errors can be detected while a data stream is being received. Although error conditions are detected as bits are received, the noise flag (NF), the parity flag (PF), and the framing error (FE) flag in SCxSR are not set until data is transferred from the serial shifter to the RDRx.

Table 15-33. QSCI1SR Bit Descriptions (continued)

Bits	Name	Description
5	QBHF	Receiver queue bottom-half full. QBHF is set when the receive queue locations SCRQ[8:15] are completely filled with new data received via the serial shifter. QBHF is cleared when register QSCI1SR is read with QBHF set, followed by a write of QBHF to zero. 0 The queue locations SCRQ[8:15] are partially filled with newly received data or is empty 1 The queue locations SCRQ[8:15] are completely full of newly received data
6	QTHE	Transmitter queue top-half empty. QTHE is set when all the data frames in the transmit queue locations SCTQ[0:7] have been transferred to the transmit serial shifter. QTHE is cleared when register QSCI1SR is read with QTHE set, followed by a write of QTHE to zero. 0 The queue locations SCTQ[0:7] still contain data to be sent to the transmit serial shifter 1 New data may now be written to the queue locations SCTQ[0:7]
7	QBHE	Transmitter queue bottom-half empty. QBHE is set when all the data frames in the transmit queue locations SCTQ[8:15] has been transferred to the transmit serial shifter. QBHE is cleared when register QSCI1SR is read with QBHE set, followed by a write of QBHE to zero. 0 The queue locations SCTQ[8:15] still contain data to be sent to the transmit serial shifter 1 New data may now be written to the queue locations SCTQ[8:15]
8:11	QRPNT	Queue receive pointer. QRPNT is a 4-bit counter used to indicate the position where the next valid data frame will be stored within the receive queue. This field is writable in test mode only; otherwise it is read-only.
12:15	QPEND	Queue pending. QPEND is a 4-bit decrementer used to indicate the number of data frames in the queue that are awaiting transfer to the SC1DR. This field is writable in test mode only; otherwise it is read-only. From 1 (QPEND = 0b0000) to 16 (or done, QPEND = 1111) data frames can be specified.

15.8.3 QSCI1 Transmitter Block Diagram

The block diagram of the enhancements to the SCI transmitter is shown in Figure 15-33.

19.4.1 TPU Module Configuration Register (TPUMCR)

	MSB 0	1 2	3 4	5	6	7	8	9	10	11	12 13 14 LSB 15
Field	STOP	TCR1P	TCR2P	EMU	T2CG	STF	SUPV	PSCK	TPU3	T2CSL	—
$\overline{\text{SRESET}}$	0	00	00	0	0	0	1	0	1	0	0000
Addr	0x30 4000(TPU_A), 0x30 4400 (TPU_B)										

Figure 19-5. TPUMCR — TPU Module Configuration Register

Table 19-7. TPUMCR Bit Description

Bits	Name	Description
0	STOP	Low-power stop mode enable. If the STOP bit in TPUMCR is set, the TPU3 shuts down its internal clocks, shutting down the internal microengine. TCR1 and TCR2 cease to increment and retain the last value before the stop condition was entered. The TPU3 asserts the stop flag (STF) in TPUMCR to indicate that it has stopped. 0 Enable TPU3 clocks 1 Disable TPU3 clocks
1:2	TCR1P	Timer Count Register 1 prescaler control. TCR1 is clocked from the output of a prescaler. The prescaler divides its input by 1, 2, 4, or 8. This is a write-once field unless the PWOD bit in TPUMCR3 is set. 00 Divide by 1 01 Divide by 2 10 Divide by 4 11 Divide by 8 Refer to Section 19.3.8, “Prescaler Control for TCR1” for more information.
3:4	TCR2P	Timer Count Register 2 prescaler control. TCR2 is clocked from the output of a prescaler. The prescaler divides this input by 1, 2, 4, or 8. This is a write-once field unless the PWOD bit in TPUMCR3 is set. 00 Divide by 1 01 Divide by 2 10 Divide by 4 11 Divide by 8 Refer to Section 19.3.9, “Prescaler Control for TCR2” for more information.
5	EMU	Emulation control. In emulation mode, the TPU3 executes microinstructions from DPTRAM exclusively. Access to the DPTRAM via the IMB3 is blocked, and the DPTRAM is dedicated for use by the TPU3. After reset, this bit can be written only once. 0 TPU3 and DPTRAM operate normally 1 TPU3 and DPTRAM operate in emulation mode[1]
6	T2CG	TCR2 clock/gate control 0 TCR2 pin used as clock source for TCR2 1 TCR2 pin used as gate of DIV8 clock for TCR2 Refer to Section 19.3.9, “Prescaler Control for TCR2” for more information.
7	STF	Stop flag. 0 TPU3 is operating normally 1 TPU3 is stopped (STOP bit has been set)
8	SUPV	Supervisor data space 0 Assignable registers are accessible from user or supervisor privilege level 1 Assignable registers are accessible from supervisor privilege level only

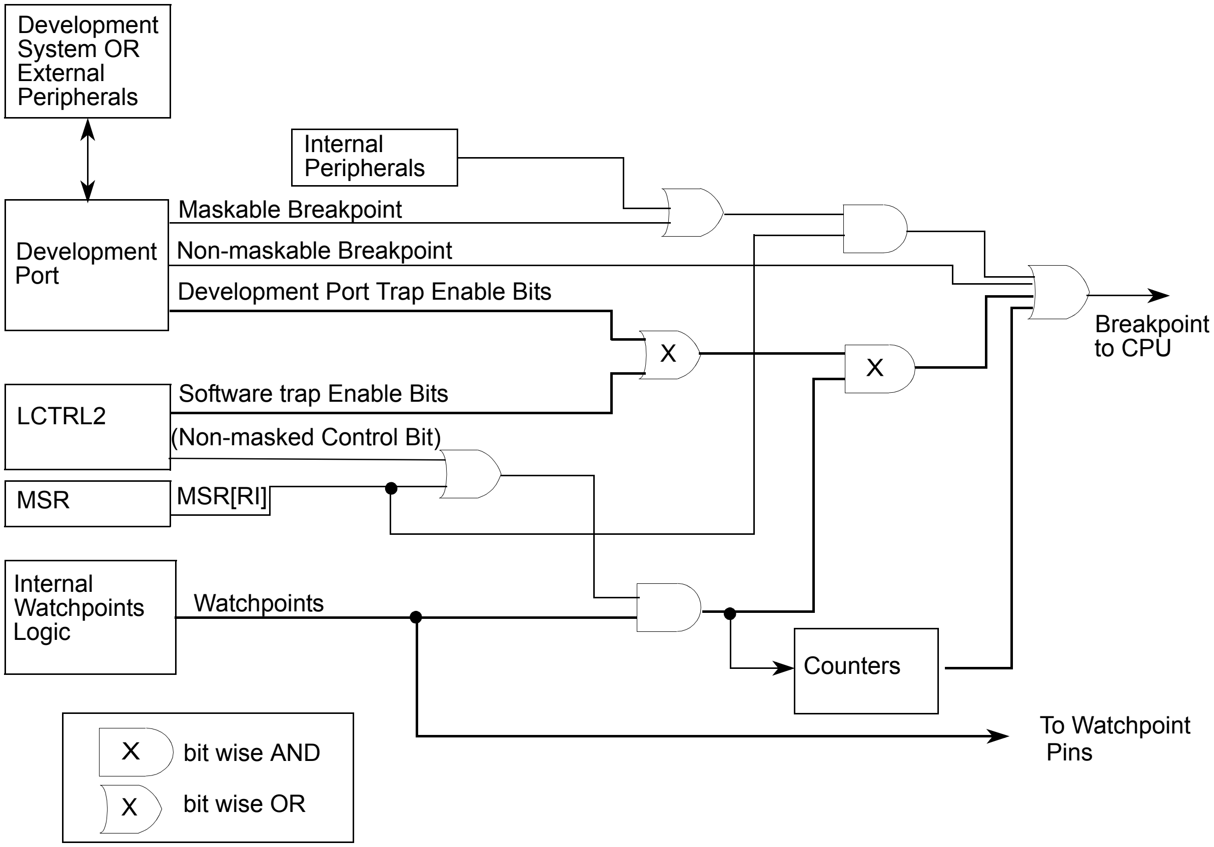


Figure 23-1. Watchpoint and Breakpoint Support in the CPU

23.2.1 Internal Watchpoints and Breakpoints

This section describes the internal breakpoints and watchpoints support of the RCPU. For information on external breakpoints support refer to Section 23.3, “Development System Interface.”

Internal breakpoint and watchpoint support is based on eight comparators comparing information on instruction and load/store cycles, two counters, and two AND-OR logic structures. The comparators perform compare on the Instruction address (I-address), on the load/store address (L-address) and on the load/store data (L-data).

The comparators are able to detect the following conditions: equal, not equal, greater than, less than (greater than or equal and less than or equal are easily obtained from these four conditions; for more information refer to Section 23.2.1.6, “Generating Six Compare Types”). Using the AND-OR logic structures “in range” and “out of range” detections (on address and on data) are supported. Using the counters, it is possible to program a breakpoint to be recognized after an event was detected a predefined number of times.

The L-data comparators can operate on fix point data of load or store. When operating on fix point data the L-data comparators are able to perform compare on bytes, half-words and words and can treat numbers either as signed or as unsigned values.

A.2.7 Class Code Compression Algorithm Rules

- Compressed instruction length may vary between 6 and 36 bits and is even.
- A compressed instruction can begin at any even location in a memory word.
- An instruction source may be compressed as a single 32-bit segment or as two independent 16-bit segments.
- Possible partitions of an instruction for compression are:
  - One 32-bit bypass segment
  - One 32-bit compressed segment
  - One 16-bit compressed segment and one 16-bit bypass segment
  - Two 16-bit compressed segments
- A bypass field is always the second field of the two possible. Length of a bypass field can be zero, 10, 15, 16 or 32 bits.
- The class prefix in a compressed instruction is 4 bits long and covers up to 16 classes.
- The vocabulary table pointer of each field may be 2 to 9 bits long.
- Vocabulary table pointers are reversed in the code. This means the pointer's LSB will be the first bit.
- In a class with a single segment of full compression, data is fetched from both memories.
- Every vocabulary table in the DECRAM is 16 bytes (8 entries) aligned (3 LSBs zeroed).

A.2.8 Bypass Field Compression Rules

The bypass field can be either a full bypass, (i.e., the whole segment from the un-compressed instruction appears as is in the compressed instruction), or it can be represented in one of several compression encoding formats. These formats are hard-wired in the decompression module.

A.2.8.1 Branch Right Segment Compression #1

For the MPC562/MPC564, a 15-bit bypass is used to indicate that the AA bit of a branch instruction should be inserted with a value of zero. The decompression process is performed as shown in Figure A-4.

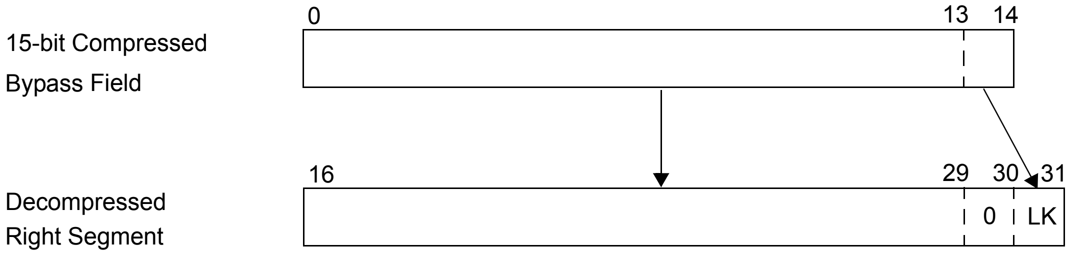


Figure A-4. Branch Right Segment Compression #1

This bypass is coded by a value of "13" (0xD) in the TP2LEN field of the DCCR register.

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

Address	Access	Symbol	Register	Size	Reset
0x307170 — 0x30717F	S/U	MBUFF7_A[1]	TouCAN_A Message Buffer 7[2]	—	U
0x30 7180 — 0x30 718F	S/U	MBUFF8_A[1]	TouCAN_A Message Buffer 8[2]	—	U
0x30 7190 — 0x30 719F	S/U	MBUFF9_A[1]	TouCAN_A Message Buffer 9[2]	—	U
0x30 71A0 — 0x30 71AF	S/U	MBUFF10_A[1]	TouCAN_A Message Buffer 10[2]	—	U
0x30 71B0 — 0x30 71BF	S/U	MBUFF11_A[1]	TouCAN_A Message Buffer 11[2]	—	U
0x30 71C0 — 0x30 71CF	S/U	MBUFF12_A[1]	TouCAN_A Message Buffer 12[2]	—	U
0x30 71D0 — 0x30 71DF	S/U	MBUFF13_A[1]	TouCAN_A Message Buffer 13[2]	—	U
0x30 71E0 — 0x30 71EF	S/U	MBUFF14_A[1]	TouCAN_A Message Buffer 14[2]	—	U
0x30 71F0 — 0x30 71FF	S/U	MBUFF15_A[1]	TouCAN_A Message Buffer 15[2]	—	U
TouCAN_B					
0x30 7480	S	CANMCR_B	TouCAN_B Module Configuration Register	16	S
0x30 7482	T	CANTCR_B	TouCAN_B Test Register	16	S
0x30 7484	S	CANICR_B	TouCAN_B Interrupt Configuration Register	16	S
0x30 7486	S/U	CANCTRL0_B/ CANCTRL1_B	TouCAN_B Control Register 0/ TouCAN_B Control Register 1	16	S
0x30 7488	S/U	PRESDIV_B/ CTRL2_B	TouCAN_B Control and Prescaler Divider Register/TouCAN_B Control Register 2	16	S
0x30 748A	S/U	TIMER_B	TouCAN_B Free-Running Timer Register		S
0x30 748C — 0x30 748E	—	—	Reserved	—	—
0x30 7490	S/U	RXGMSKHI_B	TouCAN_B Receive Global Mask High	32	S
0x30 7492	S/U	RXGMSKLO_B	TouCAN_B Receive Global Mask Low	32	S
0x30 7494	S/U	RX14MSKHI_B	TouCAN_B Receive Buffer 14 Mask High	32	S
0x30 7496	S/U	RX14MSKLO_B	TouCAN_B Receive Buffer 14 Mask Low	3	S
0x30 7498	S/U	RX15MSKHI_B	TouCAN_B Receive Buffer 15 Mask High	32	S
0x30 749A	S/U	RX15MSKLO_B	TouCAN_B Receive Buffer 15 Mask Low	32	S
0x30 749C — 0x30 749E	—	—	Reserved	—	—
0x30 74A0	S/U	ESTAT_B	TouCAN_B Error and Status Register	16	S

D.4 Table Stepper Motor (TSM)

The TSM function provides acceleration and deceleration control of a stepper motor with up to 58 programmable step rates. TSM uses a table in parameter RAM, rather than an algorithm, to define the stepper motor acceleration profile, allowing full defininition of the profile. In addition, a slew rate parameter allows fine control of the motor's terminal running speed independent of the acceleration table. The RCPU writes a desired position, and the TPU3 accelerates, slews, and decelerates the motor to the required position. Full- and half-step support is provided for two-phase motors. See Freescale TPU3 Progamming Note *Table Stepper Motor TPU Function (TSM), (TPUPN04/D).*

Figure D-4 and Figure D-5 show all of the host interface areas for the TSM function when operating in master or slave mode.

CONTROL BITS

See Table 19-24 for the
PRAM Address Offset Map.

Figure D-28. DIO Parameters

D.18 Synchronized Pulse-Width Modulation (SPWM)

The SPWM function (Bank 0) generates a pulse-width modulated waveform (PWM). The RCPU can change the period or high time of the waveform at any time. Three different operating modes allow the function to maintain complex timing relationships between channels without RCPU intervention.

The SPWM output waveform duty cycle excludes 0% and 100%. If it is not necessary for a PWM to maintain a time relationship to another PWM, the PWM function should be used instead. See Freescale TPU Progamming Note *Synchronized Pulse-Width Modulation TPU Function (SPWM), (TPUPN19/D).*

Figure D-29 shows all of the host interface areas for the SPWM function.

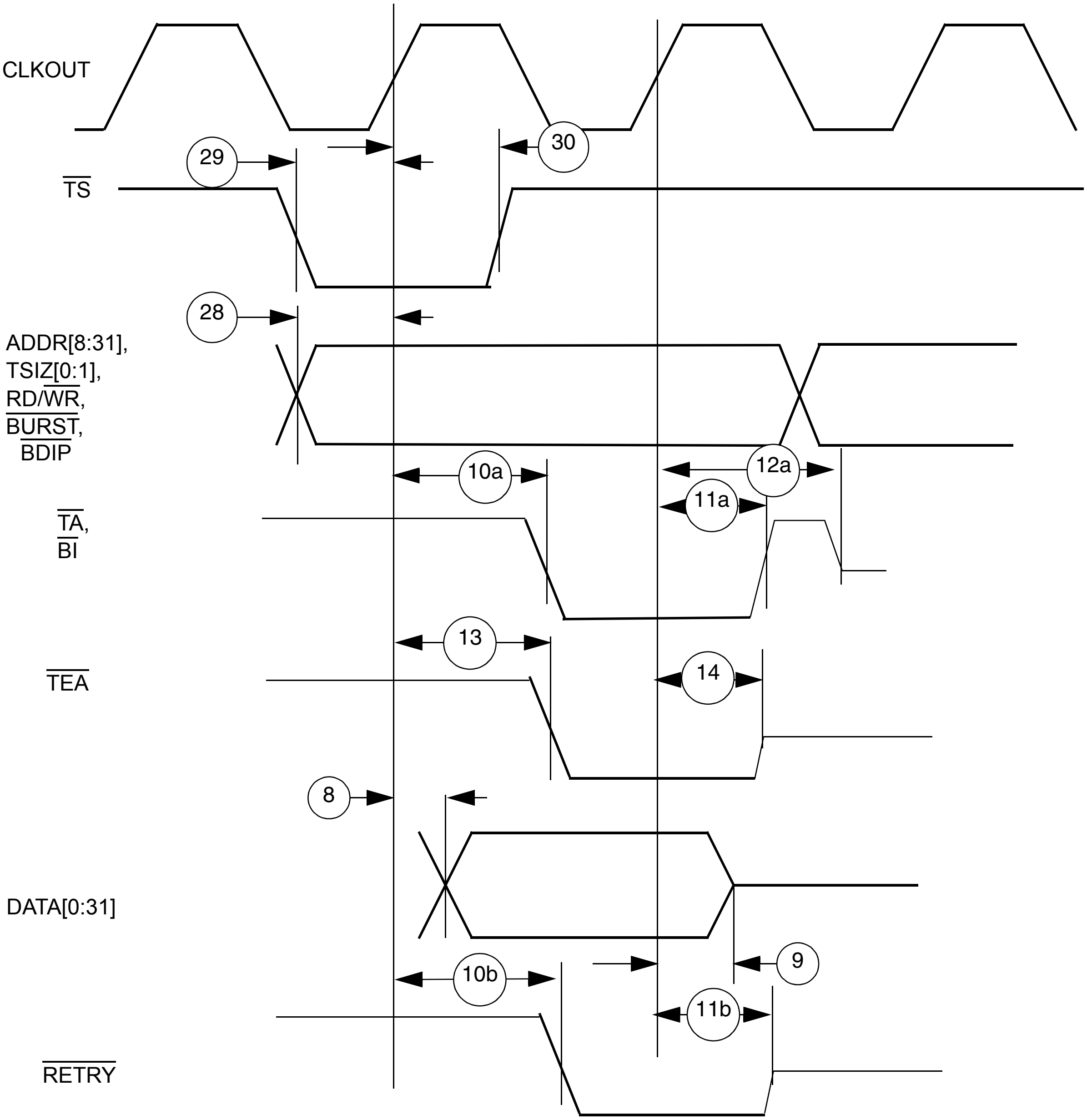


Figure F-25. External Master Read From Internal Registers Timing