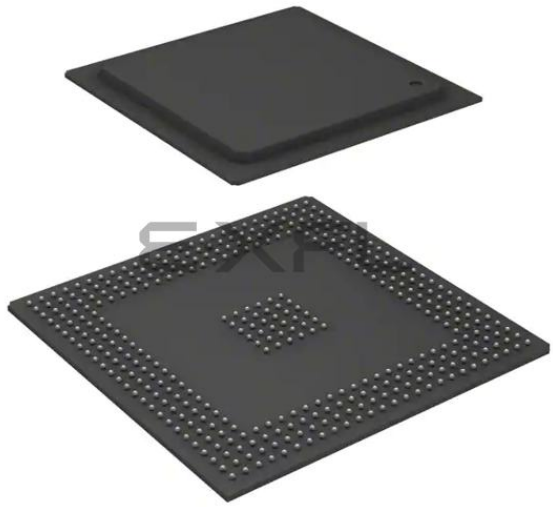


Welcome to E-XFL.COM

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	66MHz
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/mpc564czp66

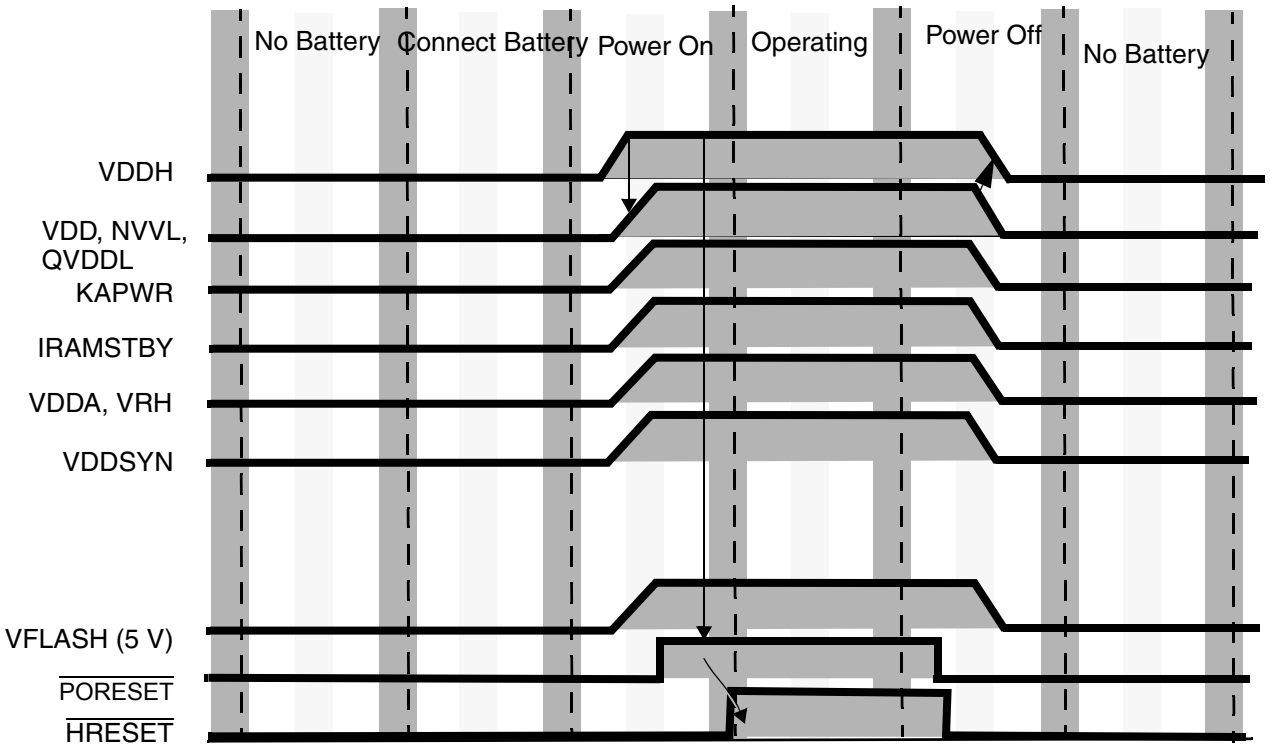
- 12 dedicated PWM sub-modules (PWMSM)
- One MIOS14 16-bit parallel port I/O sub-modules (MPIOSM)

1.3.3.3 Two Enhanced Queued Analog-to-Digital Converter Modules (QADC64E)

- Two queued analog-to-digital converter modules (QADC64E_A, QADC64E_B) providing a total of 32 analog channels
- 16 analog input channels on each QADC64E module using internal multiplexing
- Directly supports up to four external multiplexers
- Up to 41 total input channels on the two QADC64E modules with external multiplexing
- Software configurable to operate in enhanced or legacy (MPC555 compatible) mode
- Unused analog channels can be used as digital input/output signals
  - GPIO on all channels in enhanced mode
- 10-bit A/D converter with internal sample/hold
- Minimum conversion time of 7 µs (with typical QCLK frequency, 2 MHz) and +/- 2 bits accuracy
- Two conversion command queues of variable length
- Automated queue modes initiated by:
  - External edge trigger
  - Software command
  - Periodic/interval timer within the QADC64E module, that can be assigned to both queue 1 and 2
  - External gated trigger (queue 1 only)
- 64 result registers
  - Output data is right- or left-justified, signed or unsigned.
- Alternate reference input (ALTREF), with control in the conversion command word (CCW)

1.3.3.4 Three CAN 2.0B Controller (TouCAN) Modules

- Three TouCAN modules (TouCAN_A, TouCAN_B, TouCAN_C)
- Each TouCAN provides the following features:
  - 16 message buffers, programmable I/O modes
  - Maskable interrupts
  - Independent of the transmission medium (external transceiver is assumed)
  - Open network architecture, multi-master concept
  - High immunity to EMI
  - Short latency time for high-priority messages
  - Low-power sleep mode, with programmable wake-up on bus activity
  - TouCAN_C pins are shared with MIOS14 GPIO or QSMCM



1 VDDH ≥ QVDDL - 0.5 V
VDDA can lag VDDH, and VDDSYN can lag QVDDL, but both must be at a valid level before resets are negated.

2 If keep-alive functions are NOT used, then when system power is on: KAPWR = QVDDL ± 0.1 V; KAPWR ≤ 2.7 V

3 If keep-alive functions ARE used, then KAPWR = QVDDL = NVDDL = 2.6 V ± 0.1 V when system power is on KAPWR = 2.6 V ± 0.1 V when system power is off. IRAMSTBY should be powered prior to the other supplies. If IRAMSTBY is powered at the same time as the other supplies, it should be allowed to stabilize before $\overline{\text{PORESET}}$ is negated. Normal system power is defined as QVDDL = VDD = VDDF = VDDSYN = KAPWR = 2.6 ± 0.1 V and VDDA = VDDH = VFLASH = 5.0 ± 0.25 V. Flash programming requirements are the same as normal system power. VFLASH should always be 5.0 ± 0.25 V. Note: Flash is not implemented on the MPC561.

4 Do not hold the 2.6-V supplies at ground while VDDH/VDDA is ramping to 5 V.

5 If 5 V is applied before the 2.6-V supply, all 5-V outputs will be in indeterminate states until the 2.6-V supply reaches a level that allows reset to be distributed throughout the device If 5 V is applied before the 2.6-V supply, all 5-V outputs will be in indeterminate states until the 2.6-V supply reaches a level that allows reset to be distributed throughout the device

Figure 8-15. Standby and KAPWR, Other Power-On/Off

NOTE

For more detailed information on power sequencing see Section F.8, "Power-Up/Down Sequencing."

8.11 Clocks Unit Programming Model

8.11.1 System Clock Control Register (SCCR)

The SPLL has a 32-bit control register, SCCR, which is powered by keep-alive power.

Table 8-11. PLPRCR Bit Descriptions (continued)

Bits	Name	Description
16	SPLSS	SPLL lock status sticky bit. An out-of-lock sets the SPLSS bit. The bit remains set until software clears it by writing a one to it. A write of zero has no effect on this bit. The bit is cleared at power-on reset. This bit is not affected due to a software initiated loss-of-lock (MF change and entering deep-sleep or power-down mode). The SPLSS bit is not affected by hard reset. 0 SPLL has remained in lock 1 SPLL has gone out of lock at least once (not due to software-initiated loss of lock)
17	TEXPS	Timer expired status bit. This bit controls whether the chip negates the TEXP pin in deep-sleep mode, thus enabling external circuitry to switch off the VDD (power-down mode). When LPM = 11, CSRC = 0, and TEXPS is high, the TEXP pin remains asserted. When LPM = 11, CSRC = 0, and TEXPS is low, the TEXPS pin is negated. To enable automatic wake-up TEXPS is asserted when one of the following occurs: • The PIT is expired • The real-time clock alarm is set • The time base clock alarm is set • The decrementer exception occurs • The bit remains set until software clears it by writing a one to it. A write of zero has no effect on this bit. TEXPS is set by power-on or hard reset. 0 TEXP is negated in deep-sleep mode 1 TEXP pin remains asserted always
18	TEXP_INVP	Timer Expired Pin Inversed Polarity – The TEX_INVP bit controls whether the polarity of the TEXP pin will be active high (normal default) or active low. 0 The TEXP pin is active high 1 The TEXP pin is active low
19	TMIST	Timers interrupt status.TMIST is set when an interrupt from the RTC, PIT, TB or DEC occurs. The TMIST bit is cleared by writing a one to it. Writing a zero has no effect on this bit. The system clock frequency remains at its high frequency value (defined by DFNH) if the TMIST bit is set, even if the CSRC bit in the PLPRCR is set (DFNL enabled) and conditions to switch to normal-low mode do not exist. This bit is cleared during power-on or hard reset. 0 No timer expired event was detected 1 A timer expire event was detected
20	—	Reserved
21	CSRC	Clock source. This bit is cleared at hard reset. 0 General system clock is determined by the DFNH value 1 General system clock is determined by the DFNL value
22:23	LPM	Low-power mode select. These bits are encoded to provide one normal operating mode and four low-power modes. In normal and doze modes, the system can be in high state (frequency determined by the DFNH bits) or low state (frequency defined by the DFNL bits). The LPM field can be write-protected by setting the LPM and CSRC lock (LPML) bit in the SCCR Refer to Table 8-4 and Table 8-5.
24	CSR	Checkstop reset enable. If this bit is set, then an automatic reset is generated when the RCPU signals that it has entered checkstop mode, unless debug mode was enabled at reset. If the bit is clear and debug mode is not enabled, then the USIU will not do anything upon receiving the checkstop signal from the RCPU. If debug mode is enabled, then the part enters debug mode upon entering checkstop mode. In this case, the RCPU will not assert the checkstop signal to the reset circuitry. This bit is writable once after soft reset. 0 No reset will occur when checkstop is asserted 1 Reset will occur when checkstop is asserted

9.5.7.3 Bus Busy

$\overline{BB}$ assertion indicates that the current bus master is using the bus. New masters should not begin transfer until this signal is negated. The bus owner should not relinquish or negate this signal until the transfer is complete. To avoid contention on the $\overline{BB}$ line, the master should three-state this signal when it gets a logical one value. This requires the connection of an external pull-up resistor to ensure that a master that acquires the bus is able to recognize the $\overline{BB}$ line negated, regardless of how many cycles have passed since the previous master relinquished the bus. Refer to Figure 9-25.

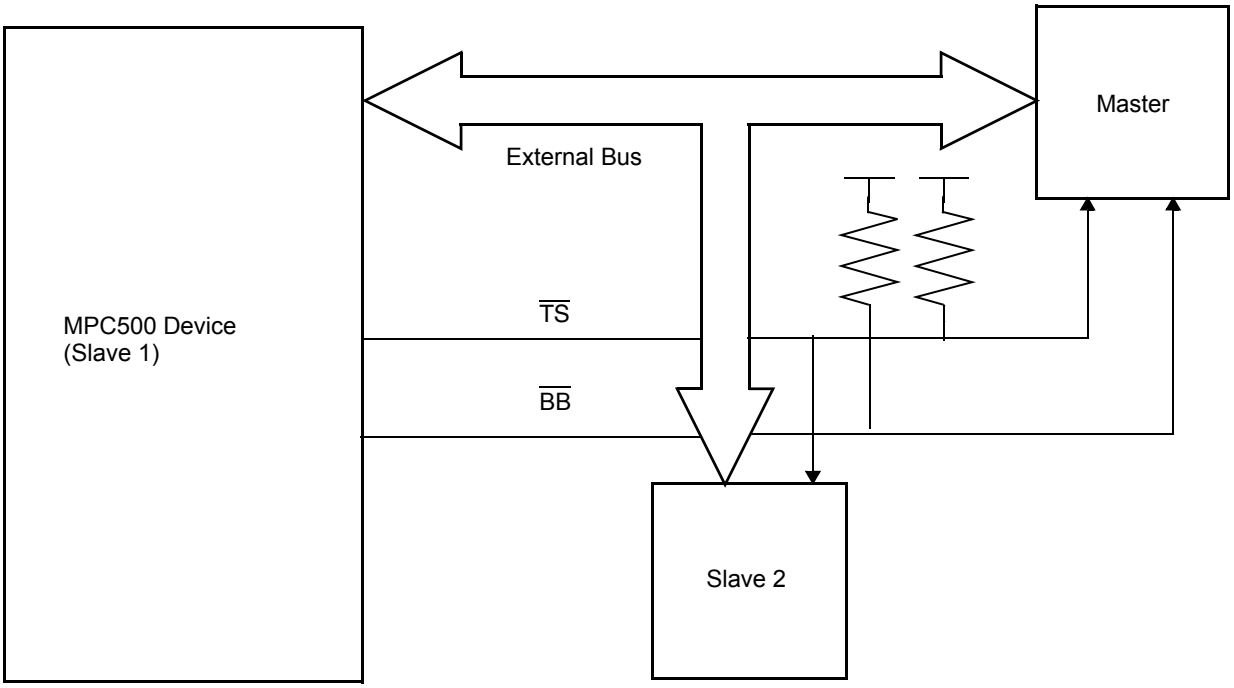


Figure 9-25. Master Signals Basic Connection

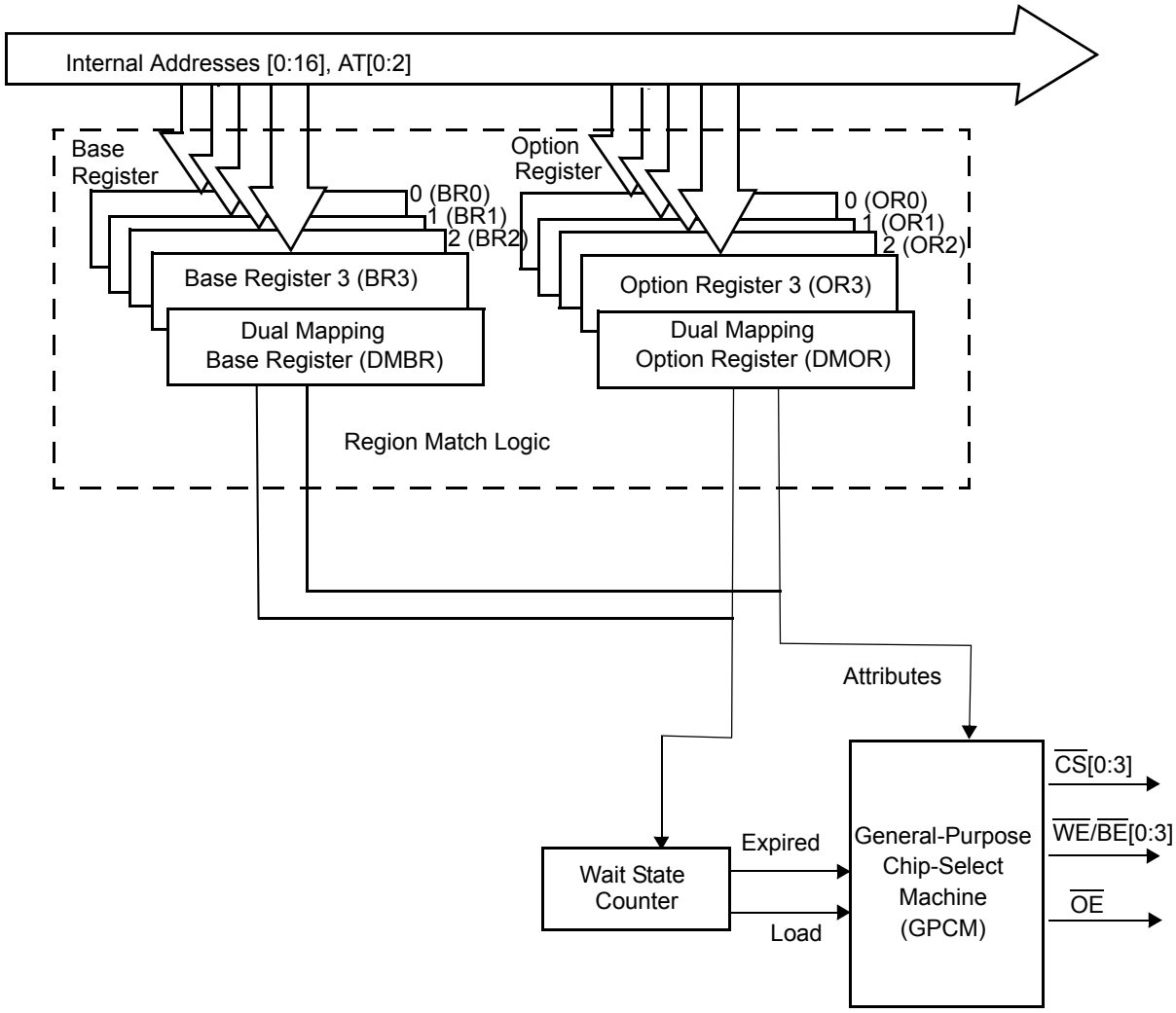


Figure 10-2. Memory Controller Block Diagram

Most memory controller features are common to all four banks. (For features unique to the $\overline{CS0}$ bank, refer to Section 10.7, "Global (Boot) Chip-Select Operation.") A full 32-bit address decode for each memory bank is possible with 17 bits having address masking. The full 32-bit decode is available, even if all 32 address bits are not MPC561/MPC563 signals connected to the external device.

Each memory bank includes a variable block size of 32 Kbytes, 64 Kbytes and up to four Gbytes. Each memory bank can be selected for read-only or read/write operation. The access to a memory bank can be restricted to certain address type codes for system protection. The address type comparison occurs with a mask option as well.

From 0 to 30 wait states can be programmed with $\overline{TA}$ generation. Four write-enable and byte-enable signals ($\overline{WE}/\overline{BE}[0:3]$) are available for each byte that is written to memory. An output enable ($\overline{OE}$) signal is provided to eliminate external glue logic. A memory transfer start ($\overline{MTS}$) strobe permits one master on a bus to access external memory through the chip selects on another.

The memory controller functionality allows MPC561/MPC563-based systems to be built with little or no glue logic. A minimal system using no glue logic is shown in Figure 10-3. In this example $\overline{CS0}$ is used for

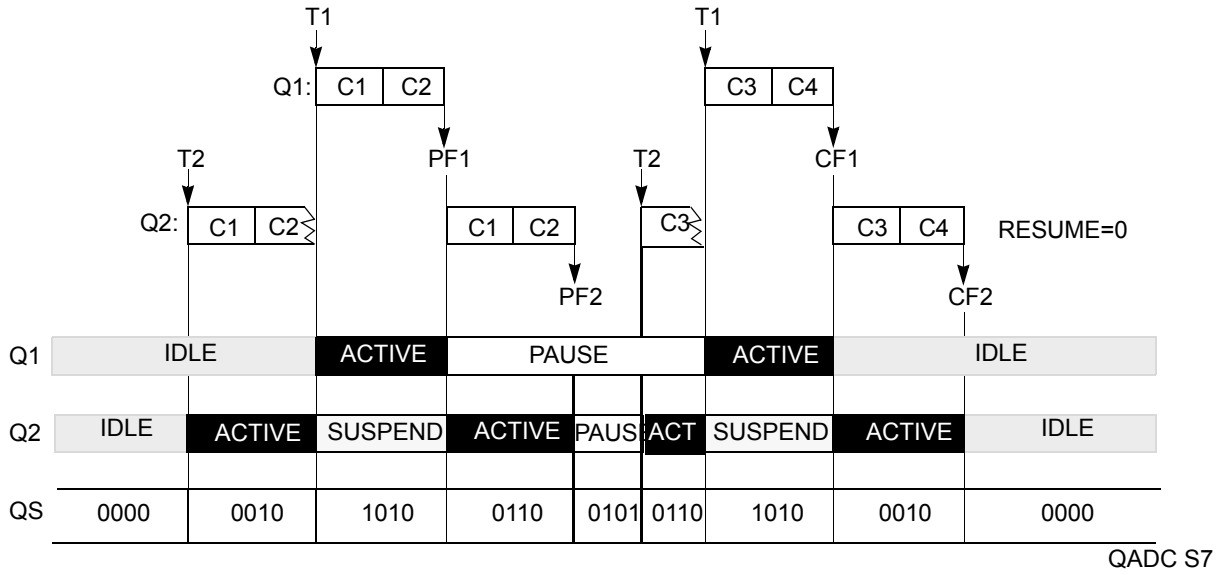


Figure 13-33. CCW Priority Situation 7

Situations S8 and S9 (Figure 13-34 and Figure 13-35) repeat the same two situations with the resume bit set to a one. When the RES bit is set, following suspension, queue 2 resumes execution with the aborted CCW, not the first CCW in the queue.

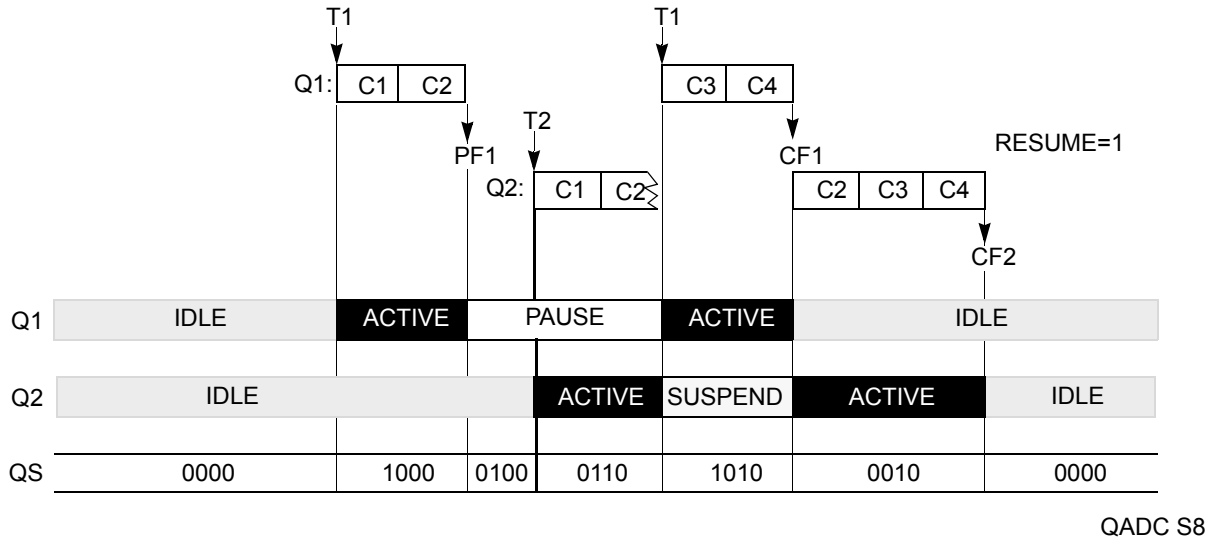


Figure 13-34. CCW Priority Situation 8

Table 14-13. QACR2 Bit Descriptions

Bits	Name	Description
0	CIE2	Queue 2 Completion Software Interrupt Enable. CIE2 enables an interrupt upon completion of queue 2. The interrupt request is initiated when the conversion is complete for the CCW in queue 2. 0 Disable the queue completion interrupt associated with queue 2 1 Enable an interrupt after the conversion of the sample requested by the last CCW in queue 2
1	PIE2	Queue 2 Pause Software Interrupt Enable. PIE2 enables an interrupt when queue 2 enters the pause state. The interrupt request is initiated when conversion is complete for a CCW that has the pause bit set. 0 Disable the pause interrupt associated with queue 2 1 Enable an interrupt after the conversion of the sample requested by a CCW in queue 2 which has the pause bit set
2	SSE2	Queue 2 Single-Scan Enable Bit. SSE2 enables a single-scan of queue 2 to start after a trigger event occurs. The SSE2 bit may be set to a one during the same write cycle when the MQ2 bits are set for one of the single-scan queue operating modes. The single-scan enable bit can be written as a one or a zero, but is always read as a zero. The SSE2 bit enables a trigger event to initiate queue execution for any single-scan operation on queue 2. The QADC64E clears the SSE2 bit when the single-scan is complete. Refer to Table 14-14 for more information. 0 Trigger events are not accepted for single-scan modes 1 Accept a trigger event to start queue 2 in a single-scan mode
3:7	MQ2	Queue 2 Operating Mode. The MQ2 field selects the queue operating mode for queue 2. Refer to Table 14-14 for more information.
8	RESUME	0 After suspension, begin executing with the first CCW in queue 2 or the current sub-queue 1 After suspension, begin executing with the aborted CCW in queue 2

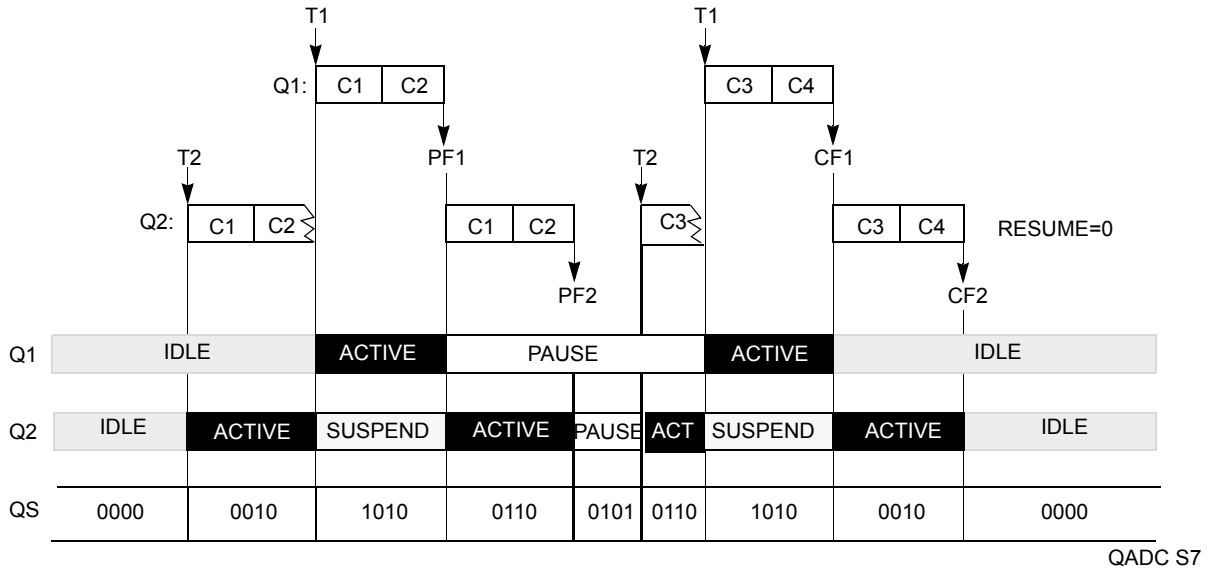


Figure 14-31. CCW Priority Situation 7

Situations S8 and S9 (Figure 14-32 and Figure 14-33) repeat the same two situations with the resume bit set to a one. When the RES bit is set, following suspension, queue 2 resumes execution with the aborted CCW, not the first CCW in the queue.

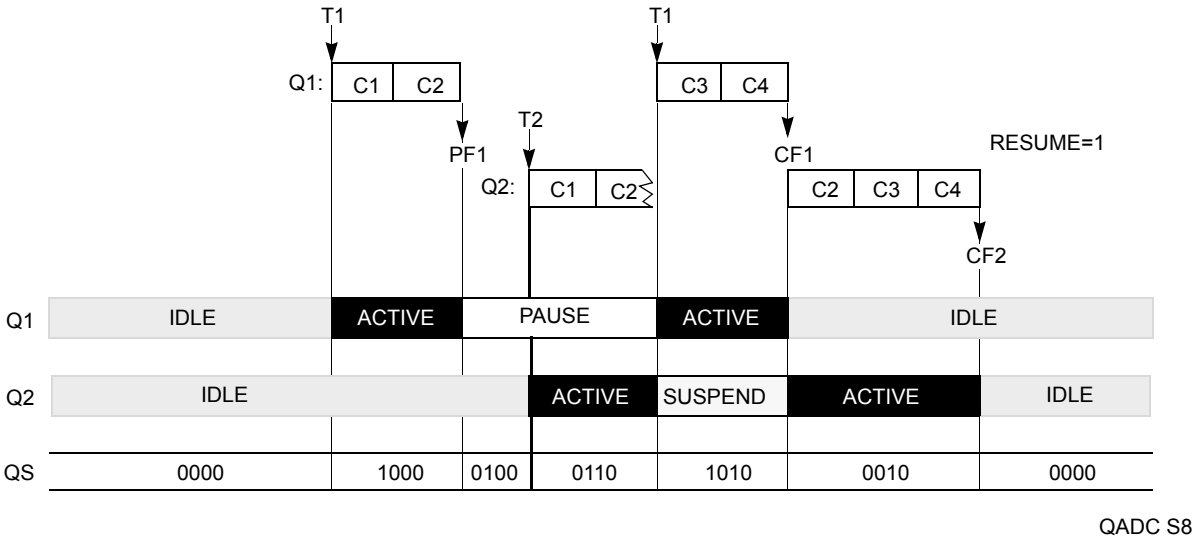


Figure 14-32. CCW Priority Situation 8

16.4.4.1 Receive Message Buffer Deactivation

Any write access to the control/status word of a receive message buffer during the process of selecting a message buffer for reception immediately deactivates that message buffer, removing it from the reception process.

If a receive message buffer is deactivated while a message is being transferred into it, the transfer is halted and no interrupt is requested. If this occurs, that receive message buffer may contain mixed data from two different frames.

The CPU should not write data into a receive message buffer. If this occurs while a message is being transferred from a serial message buffer, the control/status word will reflect a full or overrun condition, but no interrupt is requested.

16.4.4.2 Locking and Releasing Message Buffers

The lock/release/busy mechanism is designed to guarantee data coherency during the receive process. The following examples demonstrate how the lock/release/busy mechanism affects TouCAN operation:

1. Reading a control/status word of a message buffer triggers a lock for that message buffer. A new received message frame which matches the message buffer cannot be written into this message buffer while it is locked.
2. To release a locked message buffer, the CPU either locks another message buffer by reading its control/status word or globally releases any locked message buffer by reading the free-running timer.
3. If a receive frame with a matching ID is received during the time the message buffer is locked, the receive frame is not immediately transferred into that message buffer, but remains in the serial message buffer. There is no indication when this occurs.
4. When a locked message buffer is released, if a frame with a matching identifier exists within the serial message buffer, then this frame is transferred to the matching message buffer.
5. If two or more receive frames with matching IDs are received while a message buffer with a matching ID is locked, the last received frame with that ID is kept within the serial message buffer, while all preceding ones are lost. There is no indication when this occurs.
6. If the control/status word of a receive message buffer is read while a frame is being transferred from a serial message buffer, the BUSY code is indicated. The user should wait until this code is cleared before continuing to read from the message buffer to ensure data coherency. In this situation, the read of the control/status word does not lock the message buffer.

Polling the control/status word of a receive message buffer can lock it, preventing a message from being transferred into that buffer. If the control/status word of a receive message buffer is read, it should be followed by a read of the control/status word of another buffer, or by a read of the free-running timer, to ensure that the locked buffer is unlocked.

17.8.5.3 MMCSM Modulus Latch Register (MMCSMML)

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	ML															
SRESET	Undefined															
Addr	0x30 6032, 0x30 603A, 0x30 6042, 0x30 60B2, 0x30 60BA, 0x30 60C2															

Figure 17-13. MMCSM Modulus Latch Register (MMCSMML)

Table 17-11. MMCSMML Bit Descriptions

Bits	Name	Description
0:15	ML	Modulus latches — These bits are read/write data bits containing the 16-bit modulus value to be loaded into the up-counter. The value loaded in this register must be the one's complement of the desired modulus count. The up-counter increments from this one's complement value up to 0xFFFF to get the correct number of steps before an overflow is generated to reload the modulus value into the up-counter.

17.8.5.4 MMCSM Status/Control Register (MMCSMSCRD) (Duplicated)

The MMCSMSCRD and the MMCSMSCR are the same registers accessed at two different addresses. Reading or writing to one of these two addresses has exactly the same effect.

The duplication of the SCR register allows coherent 32-bit accesses when using a RCPU.

WARNING

The user should not write directly to the address of the MMCSMSCRD. This register's address may be reserved for future use and should not be accessed by the software to ensure future software compatibility.

17.8.5.5 MMCSM Status/Control Register (MMCSMSCR)

The status/control register (SCR) is a collection of read-only signal status bits, read/write control bits and an 8-bit read/write data register, as detailed below.

	MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	LSB 15
Field	PINC	PINL	FREN	EDGN	EDGP	CLS		—	CP							
SRESET	Undefined															
Addr	0x30 6036, 0x30 603E, 0x30 6046, 0x30 60B6, 0x30 60BE, 0x30 60C6															

Figure 17-14. MMCSM Status/Control Register (MMCSMSCR)

mode selection. Data registers A and B are accessible at consecutive addresses. Writing to data register B stores the same value in registers B1 and B2.

WARNING

When changing modes, it is imperative to go through the DIS mode. Failure to do this could lead to invalid and unexpected output compare or input capture results, and to flags being set incorrectly.

17.9.3.2 Input Pulse Width Measurement (IPWM) Mode

IPWM mode is selected by setting MODE[0:3] to 0b0001.

This mode allows the width of a positive or negative pulse to be determined by capturing the leading edge of the pulse on channel B and the trailing edge of the pulse on channel A; successive captures are done on consecutive edges of opposite polarity. The edge sensitivity is selected by the EDPOL bit in the MDASMSCR register.

This mode also allows the software to determine the logic level on the input signal at any time by reading the PIN bit in the MDASMSCR register.

The channel A input capture function remains disabled until the first leading edge triggers the first input capture on channel B (refer to Figure 17-16). When this leading edge is detected, the count value of the 16-bit counter bus selected by the BSL[1:0] bits is latched in the 16-bit data register B1; the FLAG line is not activated. When the next trailing edge is detected, the count value of the 16-bit counter bus is latched into the 16-bit data register A and, at the same time, the FLAG line is activated and the contents of register B1 are transferred to register B2.

Reading data register B returns the value in register B2. If subsequent input capture events occur while the FLAG bit is set in the corresponding MIRSM, data registers A and B will be updated with the latest captured values and the FLAG line will remain active.

If a 32-bit coherent operation is in progress when the trailing edge is detected, the transfer from B1 to B2 is deferred until the coherent operation is completed. Operation of the MDASM then continues on channels B and A as previously described.

The input pulse width is calculated by subtracting the value in data register B from the value in data register A.

Figure 17-16 provides an example of how the MDASM can be used for input pulse width measurement.

- An output flip-flop with output buffer and polarity control
- An input/output signal with data direction control
- An 8-bit prescaler and clock selection logic
- A 16-bit down-counter (MPWMCNTR)
- A register to hold the next period values (MPWMPERR)
- Two registers to hold the current and next pulse width values (MPWMPULR)
- A less-than or equal comparator
- A status and control register (MPWMSCR)

17.10.3.1 Clock Selection

The MPWMSM contains an 8-bit prescaler clocked by the output signal from the MIOS14 counter prescaler submodule ($f_{SYS}/2$ to $f_{SYS}/16$). The MPWMSM clock selector allows the choice, by software, of one of 256 divide ratios which give to the MPWMSM a large choice of frequencies available for the down-counter. The MPWMSM down-counter is thus capable of counting with a clock frequency ranging from $f_{SYS}/2$ to $f_{SYS}/4096$.

Switching the MPWMSM from disable to enable will reload the value of MPWMSCR[CP] into the 8-bit prescaler counter.

17.10.3.2 Counter

A 16-bit down-counter in the MPWMSM provides the time reference for the output signal. The counter is software writable. When writing to the counter (i.e., at the MPWMCNTR address), it also writes to the MPWMPERR register. When in transparent mode (TRSP = 1), writing to the MPWMPERR will also write to the counter. The down-counter is readable at anytime. The value loaded in the down-counter corresponds to the period of the output signal.

When the MPWMSM is enabled, the counter begins counting. As long as it is enabled, the counter counts down freely. The counter counts at the rate established by the prescaler. When the count down reaches 0x0001, the load operation is executed and the value in the MPWMPERR register is loaded in the MPWMCNTR register, (i.e., the counter). Then the counter restarts to count down from that value.

17.10.3.3 Period Register

The period section is composed of a 16-bit data register (MPWMPERR). The software establishes the period of the output signal in register MPWMPERR.

When the MPWMSM is running in transparent mode, the period value in register MPWMPERR is immediately transferred to the counter on a write to the MPWMPERR.

When the MPWMSM is running in double-buffered mode, the period value in register MPWMPERR can be changed at any time without affecting the current period of the output signal. The new value of MPWMPERR will be transferred to the counter only when the counter reaches the value of 0x0001 and generates a load signal.

Period values of 0x0000, 0x0001, and 0x0002 are MPWMSM special cases:

- The value 0x0000 in the period register, causes the counter to act like a free running counter. This condition creates a period of 65536 PWM clock periods.
- The value 0x0001 in the period register will always cause a period match to occur and the counter will never decrement below 0x0001. This condition is defined as a period of "1" PWM clock count. The output flip-flop is always set unless MPWMPULR = 0x0000, when the output flip-flop is always reset. Refer to Section 17.10.3.5, "Duty Cycles (0% and 100%)" for details about 0% and 100% duty cycles.
- Writing value 0x0002 in the period register causes a period match to occur every two clock periods. The counter decrements from 0x0002 to 0x0001, and then it is initialized back to 0x0002. This condition is defined as a period of 2 clock counts. Note that the value 0x0002 loaded in the period register and a value of 0x0001 in the pulse width register is the condition to obtain the maximum possible output frequency for a given clock period.

The relationship between the output frequency obtained (F_{PWMO}) and the MIOS14 CLOCK frequency (f_{SYS}), the MCPSM clock divide ratio (N_{MCPSM}), the counter divide ratio (N_{MPWMSM}) and the value loaded in the counter ($V_{COUNTER}$) is given by the following equation:

$$f_{PWMO} = \frac{f_{SYS}}{N_{MCPSM} \cdot N_{MPWMSM} \cdot V_{COUNTER}}$$

17.10.3.4 Pulse Width Registers

The pulse width section is composed of two 16-bit data registers (MPWMPULR1 and MPWMPULR2). Only MPWMPULR1 is accessible by software. The software establishes the pulse width of the MPWMSM output signal in MPWMPULR1. MPWMPULR2 is used as a double buffer of MPWMPULR1.

When the MPWMSM is running in transparent mode, the pulse width value in MPWMPULR1 is immediately transferred in MPWMPULR2 so that the new value takes effect immediately.

NOTE

When the MPWMSM is in disable mode, writing to MPWMPULR1 will write automatically to MPWMPULR2.

When the MPWMSM is not running in double-buffered mode, the pulse width value in MPWMPULR1 can be changed at any time without affecting the current pulse width of the output signal. The new value in MPWMPULR1 will be transferred to MPWMPULR2 only when the down-counter reaches the value of 0x0001.

When the counter first reaches the value in MPWMPULR2, the output flip-flop is set. The output is reset when the counter reaches 0x0001. The pulse width match starts the width of the output signal, it does not affect the counter. MPWMPULR1 is software readable and writable at any time. The MPWMSM does not modify the content of MPWMPULR1.

The PWM output pulse width can be as wide as one period minus one MPWMSM clock count: (i.e., MPWMPULR2 = MPWMPERR — [one MPWMSM clock count]). At the other end of the pulse width range, MPWMPULR2 can contain 0x0001 to create a pulse width of one PWM clock count.

Chapter 22
CALRAM Operation

The calibration static random access memory (CALRAM) module provides the MPC561/MPC563 with a general purpose memory that may be read from or written to as either bytes, half-words, or words. In addition to this, a portion of the CALRAM, called the overlay region, can be used for calibration. Calibration in this context is defined as overlaying portions of the U-bus Flash with a portion of the CALRAM array. During normal Flash access, the RISC central processing unit (RPCU) reads data from U-bus Flash (through L-bus and L2U) as shown in Figure 22-1. During calibration access, instead of Flash providing the data, the overlay regions of CALRAM provide the data to the RPCU.

22.1 Features

Standard CALRAM features are listed below:

- One-clock accesses
  - Two-cycle access for power savings
- Byte, half-word (16-bits), or word (32-bit) read/write accesses
- Each 8-Kbyte block has individual protection control bits.
- Low power standby operation for data retention

Special overlay features are:

- Eight overlay regions; each can be programmed to be 4-, 16-, 32-, 64-, 128-, 256-, or 512-bytes long
- Each overlay region size can be forced to 4 bytes long
- Data driven from the CALRAM module for overlay access has the same timing as the data that would have come from the U-bus Flash
- Overlay is for data read from the U-bus Flash space and does not affect instruction fetches from the Flash
- Overlay block is naturally aligned
  - For example, 128-byte block is 128-byte aligned
- Normal access to overlaid portion of CALRAM array can be made to generate an error (machine check) if so configured

Table 23-14. Development Support Programming Model (continued)

SPR Number (Decimal)	Name
154	Comparator G Value Register (CMPG) See Table 23-23 for bit descriptions.
155	Comparator H Value Register (CMPH) See Table 23-23 for bit descriptions.
156	L-Bus Support Control Register 1 (LCTRL1) See Table 23-24 for bit descriptions.
157	L-Bus Support Control Register 2 (LCTRL2) See Table 23-25 for bit descriptions.
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.

23.6.1 Register Protection

Table 23-15 and Table 23-16 summarize protection features of development support registers during read and write accesses, respectively.

Table 23-15. Development Support Registers Read Access Protection

MSR[PR]	Debug Mode Enable	In Debug Mode	Result
0	0	X	Read is performed. ECR is cleared when read. Reading DPDR yields indeterminate data.
0	1	0	Read is performed. ECR is not cleared when read. Reading DPDR yields indeterminate data.
0	1	1	Read is performed. ECR is cleared when read.
1	X	X	Program exception is generated. Read is not performed. ECR is not cleared when read.

4. Once the read access is completed, the upload/download information public message (TCODE = 19) is transmitted to the tool along with the data read from the UDI register. This message also indicates that the device is ready for next access.
5. The SC field in the RWA register is cleared.

24.10.3.2 Block Read Operation

For a block read access to memory-mapped locations and SPR registers, the following sequence of operations need to be performed via the auxiliary port:

1. The tool confirms that the device is ready before transmitting download request public message (TCODE = 18).
2. The download request public message contains:
   a) TCODE(18)
   b) Access opcode 0xF which signals that subsequent data needs to be stored in the RWA register.
   c) Configure the RWA fields as follows:
      – Start/complete (1 to indicate start access) -> SC
      – Read/write address (starting read address of block) -> RWAD
      – Read/write (0 to indicate a read access) -> RW
      – Word size (32 bits, 16 bits, 8 bits) -> SZ
      – Write data (0xXXXXXXXX-> WD [don’t care])
      – Privilege (user data/instruction, supervisor data/instruction) > PRV
      – Map select (select memory map 0b0) -> MAP
      – Access count (non-zero number to indicate block access) -> CNT
3. Data read from the specified address is stored in the UDI register.
4. After the completion of this read operation, the upload/download information public message (TCODE=19) is transmitted to the tool along with the data read from the UDI register. This message also indicates that the device is ready to perform the next read operation.
5. The specified address (in RWAD field) is incremented to the next word size and the number in the CNT field is decremented. The SC field is not cleared.
6. The data read from the new address is stored in the UDI register.
7. Steps 4 through 7 are repeated until the count value in the CNT field of RWA register equals zero. The SC bit is cleared to indicate end of the block read access.

24.10.4 Read/Write Access to Internal READI Registers

24.10.4.1 Write Operation

For a write access to internal READI registers, the following sequence of operations need to be performed via the auxiliary port:

1. The tool confirms that the device is ready before transmitting download request public message (TCODE = 18).

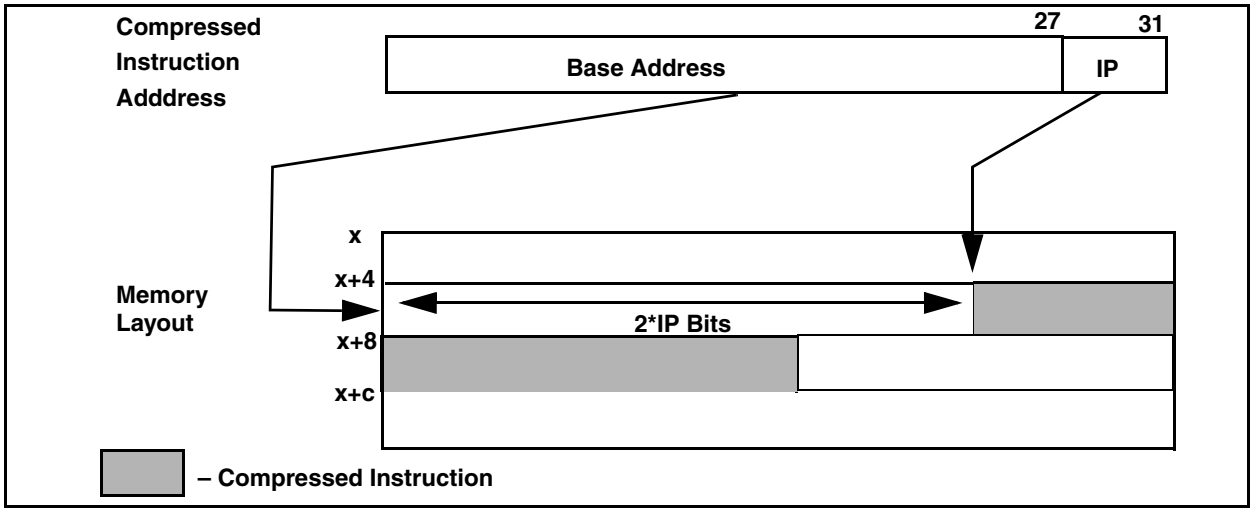


Figure A-2. Addressing Instructions with Compressed Address

A.2.4 Compressed Address Generation with Direct Branches

During the compression process, compressed instructions change their location in the memory and are not word aligned. Displacement fields in the direct branch instructions have to be updated by the compression tool to make compressed instruction addressing possible. Four LSB bits of the displacement immediate field (LI or BD) in the compressed direct branch instructions are used for bit addressing in the 32-bit memory word. The remaining bits of the fields are used in the branch target calculation of the base address (word address). The RCPU branch unit copies the bit pointer into the IP field of issued compressed branch target address. The branch compressed target base address is calculated according the direct branch addressing mode.

If a branch has absolute addressing mode, the branch target base address is calculated as a sign extension of the base address portion of the LI (or BD) field.

If a branch has relative addressing mode, the branch target base address is calculated as a sum of the base address of the branch and sign extended base address portion of the branch LI (or BD) field.

Figure A-3 illustrates direct branch target address generation in “Decompression On” mode. The base address for the unconditional branch has 20 bits This yields an unconditional branch displacement limit of 4 Mbytes. The word pointer for the conditional branch has 10 bits. This yields a conditional branch displacement limit of 4 Kbytes.

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

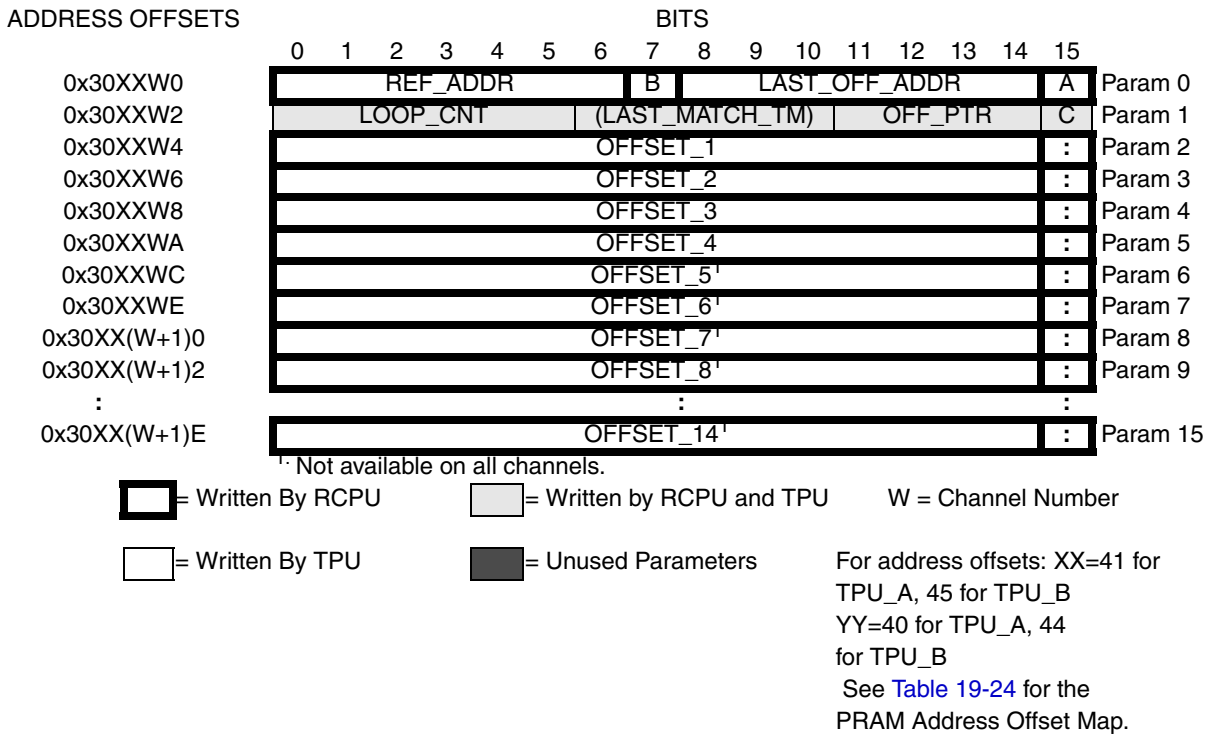


Figure D-3. QOM Parameters

D.20.1.6 SIOP_DATA

This parameter is the data register for all SIOP transfers. Data is shifted out of one end of SIOP_DATA and shifted in at the other end, the shift direction being determined by the value of BIT_D. In output-only mode, zero will be shifted into SIOP_DATA and in input-only mode, the data shifted out is ignored. In clock-only mode, SIOP_DATA is still shifted.

NOTE

The TPU3 does not "justify" the data position in SIOP_DATA (for example, if an 8-bit bidirectional transfer is made, shifting LSB first, then the bottom byte of SIOP_DATA will be shifted out and the input data will be shifted into the upper byte of SIOP_DATA).

NOTE

SIOP_DATA is not buffered. The RCPU should only access it between completion of one transfer and the start of the next.

D.20.2 Host RCPU Initialization of the SIOP Function

The RCPU initializes the SIOP function by:

1. Disabling the channel by clearing the two channel-priority bits
2. Selecting the SIOP function on the channel by writing the assigned SIOP function number to the function-select bits
3. Writing CHAN_CONTROL in the clock channel parameter RAM
4. Writing HALF_PERIOD, BIT_D, and XFER_SIZE in the clock-channel parameter RAM to determine the speed, shift direction, and size of the transfer
5. Writing SIOP_DATA if the data output is to be used
6. Selecting the required operating mode via the two host-sequence bits
7. Issuing a host service request type 0b11
8. Enabling service by assigning H, M, or L priority to the clock channel via the two channel-priority bits

The TPU3 then starts the data transfer, and issues an interrupt request when the transfer is complete.

Once the function has been initialized, the RCPU only needs to write SIOP_DATA with the new data and issue a HSR 0b11 to initiate a new transfer. In input-only or clock-only modes, just the HSR 0b11 is required.

D.20.3 SIOP Function Performance

Like all TPU3 functions, the performance limit of the SIOP function depends, because of the operational nature of the scheduler, on the service time (latency) associated with other active TPU3 channels. Where two channels are used for a uni-directional system and no other TPU3 channels are active, the maximum baud rate is approximately 230 at a bus speed of 16.77 MHz. A three-channel bidirectional system under the same conditions has a maximum baud rate of approximately 200. When more TPU3 channels are