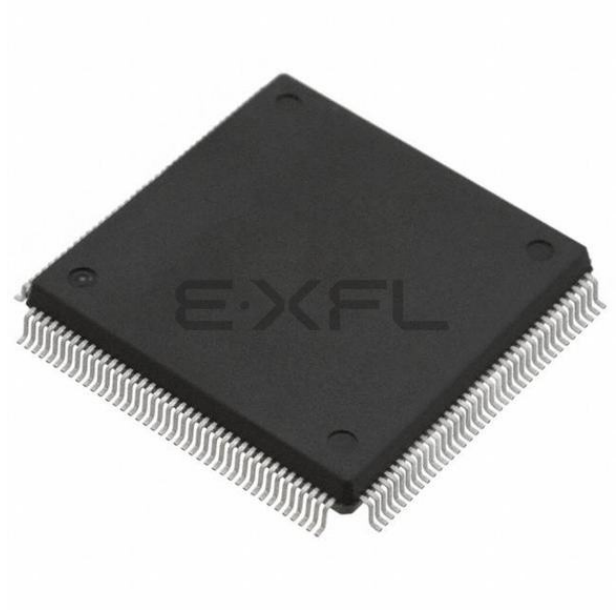


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

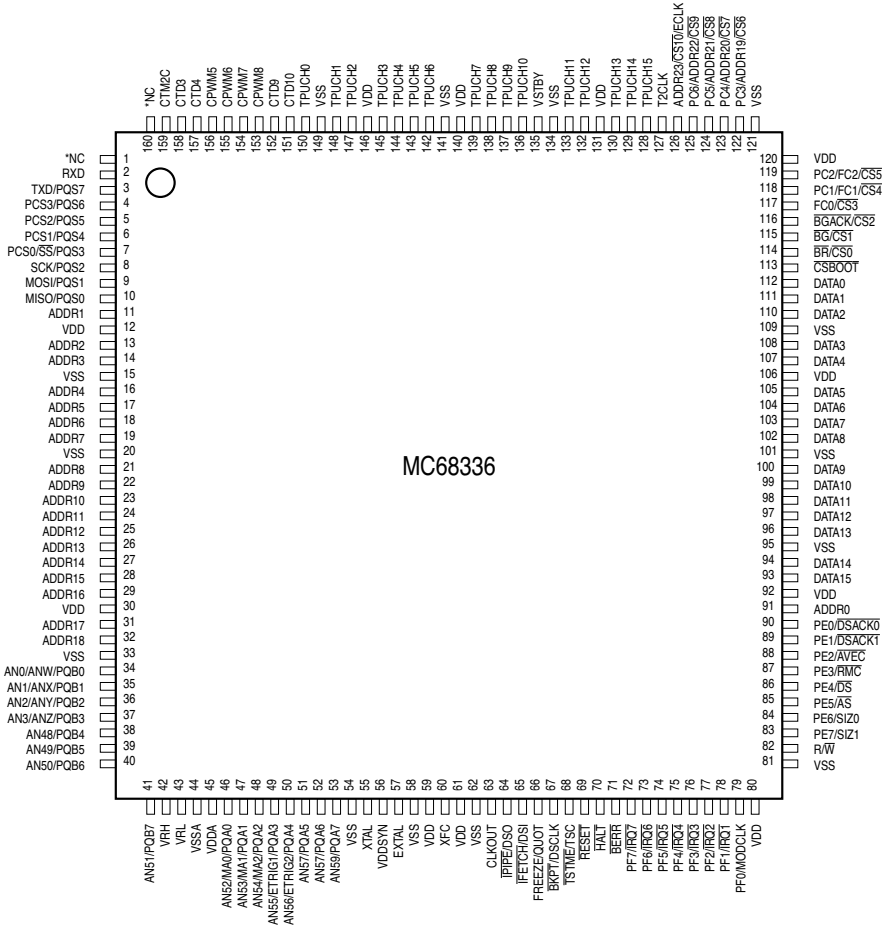
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	CPU32
Core Size	32-Bit Single-Core
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	18
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	7.5K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	160-BQFP
Supplier Device Package	160-QFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68376bacab25

LIST OF TABLES
(Continued)

FCSM12CNT	—	CTM4 FCSM12 Counter Register
FCSM12SIC	—	CTM4 FCSM12 Status/Interrupt/Control Register
HSQR[0:1]	—	TPU Host Sequence Registers [0:1]
HSRR[0:1]	—	TPU Host Service Request Registers [0:1]
LJSRR[0:27]	—	QADC Left-Justified Signed Result Registers [0:27]
LJURR[0:27]	—	QADC Left-Justified Unsigned Result Registers [0:27]
LR	—	Link Register
MCSM[2]/[11]CNT	—	CTM4 MCSM Counter Registers [2]/[11]
MCSM[2]/[11]ML	—	CTM4 MCSM Modulus Latch Registers [2]/[11]
MCSM[2]/[11]SIC	—	CTM4 MCSM Status/Interrupt/Control Registers [2]/[11]
MRMCR	—	Masked ROM Module Configuration Register
PEPAR	—	SIM Port E Pin Assignment Register
PFPAR	—	SIM Port F Pin Assignment Register
PICR	—	SIM Periodic Interrupt Control Register
PITR	—	SIM Periodic Interrupt Timer Register
PORTC	—	SIM Port C Data Register
PORTE	—	SIM Port E Data Register
PORTF	—	SIM Port F Data Register
PORTQA	—	QADC Port A Data Register
PORTQB	—	QADC Port B Data Register
PORTQS	—	QSM Port QS Data Register
PQSPAR	—	QSM Port QS Pin Assignment Register
PRESDIV	—	TouCAN Prescaler Divide Register
PWM[5:8]C	—	CTM4 PWMSM Counter Registers [5:8]
PWM[5:8]A	—	CTM4 PWMSM Period Registers [5:8]
PWM[5:8]B	—	CTM4 PWMSM Pulse Width Registers [5:8]
PWM[5:8]SIC	—	CTM4 PWMSM Status/Interrupt/Control Registers [5:8]
QACR[0:1]	—	QADC Control Registers [0:2]
QADCINT	—	QADC Interrupt Register
QADCMCR	—	QADC Module Configuration Register
QADCTEST	—	QADC Test Register
QASR	—	QADC Status Register
QILR	—	QSM Interrupt Level Register
QIVR	—	QSM Interrupt Vector Register
QSMCR	—	QSM Module Configuration Register
QTEST	—	QSM Test Register
RAMBAH	—	RAM Base Address High Register



*NOTE: MC68336 REVISION D AND LATER (F60K AND LATER MASK SETS) HAVE ASSIGNED PINS 1 AND 160 AS "NO CONNECT", TO ALLOW PIN COMPATIBILITY WITH THE MC68376. FOR REVISION C (D65J MASK SET) DEVICES, PIN 1 IS V_{SS} AND PIN 160 IS V_{DD}.

336 160-PIN QFP

Figure 3-2 MC68336 Pin Assignments for 160-Pin Package

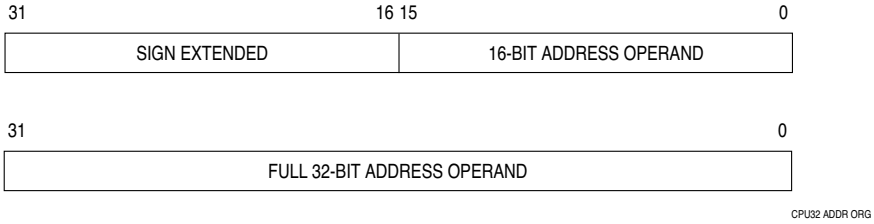


Figure 4-5 Address Organization in Address Registers

4.2.3 Program Counter

The PC contains the address of the next instruction to be executed by the CPU32. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC as appropriate.

4.2.4 Control Registers

The control registers described in this section contain control information for supervisor functions and vary in size. With the exception of the condition code register (the user portion of the status register), they are accessed only by instructions at the supervisor privilege level.

4.2.4.1 Status Register

The status register (SR) stores the processor status. It contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The condition codes are extend (X), negative (N), zero (Z), overflow (V), and carry (C). The user (low-order) byte containing the condition codes is the only portion of the SR information available at the user privilege level; it is referenced as the condition code register (CCR) in user programs.

At the supervisor privilege level, software can access the full status register. The upper byte of this register includes the interrupt priority (IP) mask (three bits), two bits for placing the processor in one of two tracing modes or disabling tracing, and the supervisor/user bit for placing the processor at the desired privilege level.

Undefined bits in the status register are reserved by Motorola for future definition. The undefined bits are read as zeros and should be written as zeros for future compatibility.

All operations to the SR and CCR are word-size operations, but for all CCR operations, the upper byte is read as all zeros and is ignored when written, regardless of privilege level.

Refer to **D.1.2 Status Register** for bit/field definitions and a diagram of the status register.

Figure 5-7 is a block diagram of the watchdog timer and the clock control for the periodic interrupt timer.

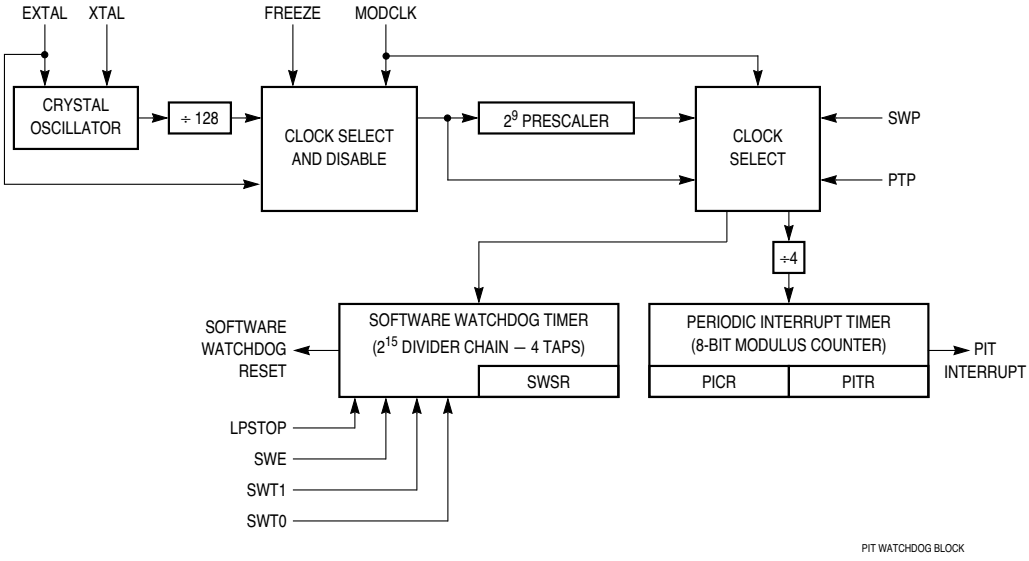


Figure 5-7 Periodic Interrupt Timer and Software Watchdog Timer

5.4.6 Periodic Interrupt Timer

The periodic interrupt timer (PIT) allows the generation of interrupts of specific priority at predetermined intervals. This capability is often used to schedule control system tasks that must be performed within time constraints. The timer consists of a prescaler, a modulus counter, and registers that determine interrupt timing, priority and vector assignment. Refer to **4.9 Exception Processing** for more information.

The periodic interrupt timer modulus counter is clocked by one of two signals. When the PLL is enabled (MODCLK = 1 during reset), $f_{ref} \div 128$ is used. When the PLL is disabled (MODCLK = 0 during reset), f_{ref} is used. The value of the periodic timer prescaler (PTP) bit in the periodic interrupt timer register (PITR) determines system clock prescaling for the periodic interrupt timer. One of two options, either no prescaling, or prescaling by a factor of 512, can be selected. The value of PTP is affected by the state of the MODCLK pin during reset, as shown in **Table 5-7**. System software can change PTP value.

Table 5-7 MODCLK Pin and PTP Bit at Reset

MODCLK	PTP
0 (PLL disabled)	1 (÷ 512)
1 (PLL enabled)	0 (÷ 1)

SIZ[1:0] signals reflect bus allocation during show cycles. Only the appropriate portion of the data bus is valid during the cycle. During a byte write to an internal address, the portion of the bus that represents the byte that is not written reflects internal bus conditions, and is indeterminate. During a byte write to an external address, the data multiplexer in the SIM causes the value of the byte that is written to be driven out on both bytes of the data bus.

5.7 Reset

Reset occurs when an active low logic level on the $\overline{\text{RESET}}$ pin is clocked into the SIM. The $\overline{\text{RESET}}$ input is synchronized to the system clock. If there is no clock when $\overline{\text{RESET}}$ is asserted, reset does not occur until the clock starts. Resets are clocked to allow completion of write cycles in progress at the time $\overline{\text{RESET}}$ is asserted.

Reset procedures handle system initialization and recovery from catastrophic failure. The MCU performs resets with a combination of hardware and software. The SIM determines whether a reset is valid, asserts control signals, performs basic system configuration and boot ROM selection based on hardware mode-select inputs, then passes control to the CPU32.

5.7.1 Reset Exception Processing

The CPU32 processes resets as a type of asynchronous exception. An exception is an event that preempts normal processing, and can be caused by internal or external events. Exception processing makes the transition from normal instruction execution to execution of a routine that deals with an exception. Each exception has an assigned vector that points to an associated handler routine. These vectors are stored in the exception vector table. The exception vector table consists of 256 four-byte vectors and occupies 1024 bytes of address space. The exception vector table can be relocated in memory by changing its base address in the vector base register (VBR). The CPU32 uses vector numbers to calculate displacement into the table. Refer to **4.9 Exception Processing** for more information.

Reset is the highest-priority CPU32 exception. Unlike all other exceptions, a reset occurs at the end of a bus cycle, and not at an instruction boundary. Handling resets in this way prevents write cycles in progress at the time the reset signal is asserted from being corrupted. However, any processing in progress is aborted by the reset exception and cannot be restarted. Only essential reset tasks are performed during exception processing. Other initialization tasks must be accomplished by the exception handler routine. Refer to **5.7.9 Reset Processing Summary** for details on exception processing.

5.7.2 Reset Control Logic

SIM reset control logic determines the cause of a reset, synchronizes reset assertion if necessary to the completion of the current bus cycle, and asserts the appropriate reset lines. Reset control logic can drive four different internal signals:

Table 5-21 Chip-Select Base and Option Register Reset Values

Fields	Reset Values
Base address	$000000
Block size	2 Kbyte
Async/sync mode	Asynchronous mode
Upper/lower byte	Disabled
Read/write	Disabled
$\overline{AS}$/$\overline{DS}$	$\overline{AS}$
$\overline{DSACK}$	No wait states
Address space	CPU space
IPL	Any level
Autovector	External interrupt vector

Following reset, the MCU fetches the initial stack pointer and program counter values from the exception vector table, beginning at $000000 in supervisor program space. The $\overline{CSBOOT}$ chip-select signal is used to select an external boot device mapped to a base address of $000000.

The MSB of the CSBTPA field in CSPAR0 has a reset value of one, so that chip-select function is selected by default out of reset. The BYTE field in chip-select option register CSORBT has a reset value of “both bytes” so that the select signal is enabled out of reset. The LSB of the $\overline{CSBOOT}$ field, determined by the logic level of DATA0 during reset, selects the boot ROM port size. When DATA0 is held low during reset, port size is eight bits. When DATA0 is held high during reset, port size is 16 bits. DATA0 has a weak internal pull-up driver, so that a 16-bit port is selected by default out of reset. However, the internal pull-up driver can be overcome by bus loading effects. To ensure a particular configuration out of reset, use an active device to put DATA0 in a known state during reset.

The base address field in the boot chip-select base address register CSBARBT has a reset value of all zeros, so that when the initial access to address $000000 is made, an address match occurs, and the $\overline{CSBOOT}$ signal is asserted. The block size field in CSBARBT has a reset value of one Mbyte. **Table 5-22** shows $\overline{CSBOOT}$ reset values.

Table 5-22 $\overline{CSBOOT}$ Base and Option Register Reset Values

Fields	Reset Values
Base address	$000000
Block size	1 Mbyte
Async/sync mode	Asynchronous mode
Upper/lower byte	Both bytes
Read/write	Read/write
$\overline{AS}$/$\overline{DS}$	$\overline{AS}$
$\overline{DSACK}$	13 wait states
Address space	Supervisor/user space
IPL[1]	Any level
Autovector	Interrupt vector externally

NOTES:
1. These fields are not used unless “Address space” is set to CPU space.

7.5 Low-Power Stop Mode Operation

Low-power stop mode minimizes MCU power consumption. Setting the STOP bit in MRMCR places the MRM in low-power stop mode. In low-power stop mode, the array cannot be accessed. The reset state of STOP is the complement of the logic state of DATA14 during reset. Low-power stop mode is exited by clearing STOP.

7.6 ROM Signature

Signature registers RSIGHI and RSIGLO contain a user-specified mask-programmed signature pattern. A special signature algorithm allows the user to verify ROM array content.

7.7 Reset

The state of the MRM following reset is determined by the default values programmed into the MRMCR $\overline{\text{BOOT}}$, LOCK, ASPC[1:0], and WAIT[1:0] bits. The default array base address is determined by the values programmed into ROMBAL and ROMBAH.

When the mask programmed value of the MRMCR $\overline{\text{BOOT}}$ bit is zero, the contents of MRM bootstrap words ROMBS[0:3] are used as reset vectors. When the mask programmed value of the MRMCR $\overline{\text{BOOT}}$ bit is one, reset vectors are fetched from external memory, and system integration module chip-select logic is used to assert the boot ROM select signal $\overline{\text{CSBOOT}}$. Refer to **5.9.4 Chip-Select Reset Operation** for more information concerning external boot ROM selection.

The SCI provides a standard non-return to zero (NRZ) mark/space format. It operates in either full- or half-duplex mode. There are separate transmitter and receiver enable bits and dual data buffers. A modulus-type baud rate generator provides rates from 110 baud to 655 kbaud with a 20.97 MHz system clock. Word length of either eight or nine bits is software selectable. Optional parity generation and detection provide either even or odd parity check capability. Advanced error detection circuitry catches glitches of up to 1/16 of a bit time in duration. Wake-up functions allow the CPU32 to run uninterrupted until meaningful data is available.

9.2 QSM Registers and Address Map

There are four types of QSM registers: QSM global registers, QSM pin control registers, QSPI registers, and SCI registers. Refer to **9.2.1 QSM Global Registers** and **9.2.2 QSM Pin Control Registers** for a discussion of global and pin control registers. Refer to **9.3.1 QSPI Registers** and **9.4.1 SCI Registers** for further information about QSPI and SCI registers. Writes to unimplemented register bits have no effect, and reads of unimplemented bits always return zero.

The QSM address map includes the QSM registers and the QSPI RAM. The MM bit in the system integration module configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each module.

Refer to **D.6 Queued Serial Module** for a QSM address map and register bit and field definitions. **5.2.1 Module Mapping** contains more information about how the state of MM affects the system.

9.2.1 QSM Global Registers

The QSM configuration register (QSMCR) contains parameters for interfacing to the CPU32 and the intermodule bus. The QSM test register (QTEST) is used during factory test of the QSM. The QSM interrupt level register (QILR) determines the priority of interrupts requested by the QSM and the vector used when an interrupt is acknowledged. The QSM interrupt vector register (QIVR) contains the interrupt vector for both QSM submodules. QILR and QIVR are 8-bit registers located at the same word address.

9.2.1.1 Low-Power Stop Operation

When the STOP bit in QSMCR is set, the system clock input to the QSM is disabled and the module enters a low-power operating state. QSMCR is the only register guaranteed to be readable while STOP is asserted. The QSPI RAM is not readable during LPSTOP. However, writes to RAM or any register are guaranteed valid while STOP is asserted. STOP can be set by the CPU32 and by reset.

System software must bring the QSPI and SCI to an orderly stop before asserting STOP to avoid data corruption. The IRQ mask level in the CPU32 status register should be set to a higher value than the IRQ level generated by the QSM module. The SCI receiver and transmitter should be disabled after transfers in progress are complete. The QSPI can be halted by setting the HALT bit in SPCR3 and then setting STOP after the HALTA flag is set. The IRQ mask in the CPU status register should be restored to its former level. Refer to **5.3.4 Low-Power Operation** for more information about low-power stop mode.

9.2.2 QSM Pin Control Registers

The QSM uses nine pins. Eight of the pins can be used for serial communication or for parallel I/O. Clearing a bit in the port QS pin assignment register (PQSPAR) assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the QSPI. PQSPAR does not select I/O. In master mode, PQSPAR causes a bit to be assigned to the QSPI when SPE is set. In slave mode, the MISO pin, if assigned to the QSPI, remains under the control of the QSPI, regardless of the SPE bit. PQSPAR does not affect operation of the SCI.

The port QS data direction register (DDRQS) determines whether pins are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function. DDQS7 determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output. PQSPAR and DDRQS are 8-bit registers located at the same word address. **Table 9-1** is a summary of QSM pin functions.

The port QS data register (PORTQS) latches I/O data. PORTQS writes drive pins defined as outputs. PORTQS reads return data present on the pins. To avoid driving undefined data, first write PORTQS, then configure DDRQS.

Table 9-1 Effect of DDRQS on QSM Pin Function

QSM Pin	Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial data input to QSPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from QSPI
MOSI	Master	DDQS1	0	Disables data output
			1	Serial data output from QSPI
	Slave		0	Serial data input to QSPI
			1	Disables data input
SCK[1]	Master	DDQS2	—	Clock output from QSPI
	Slave		—	Clock input to QSPI
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion causes mode fault
			1	Chip-select output
	Slave		0	QSPI slave select input
			1	Disables slave select input
PCS[1:3]	Master	DDQS[4:6]	0	Disables chip-select output
			1	Chip-select output
	Slave		0	Inactive
			1	Inactive
TXD[2]	—	DDQS7	X	Serial data output from SCI
RXD	—	None	NA	Serial data input to SCI

NOTES:

1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial data output TXD.

data space accesses. The SUPV bit in QADCMCR designates the assignable space as supervisor or unrestricted.

Attempts to read supervisor-only data space when the CPU32 is not in supervisor mode causes a value of $0000 to be returned. Attempts to read assignable data space when the CPU32 is not in supervisor mode and when the space is programmed as supervisor space, causes a value of $FFFF to be returned. Attempts to write supervisor-only or supervisor-assigned data space when the CPU32 is in user mode has no effect.

The supervisor-only data space segment contains the QADC global registers, which include QADCMCR, QADCTEST, and QADCINT. The supervisor/unrestricted space designation for the CCW table, the result word table, and the remaining QADC registers is programmable. Refer to **D.5.1 QADC Module Configuration Register** for more information.

8.6.4 Interrupt Arbitration Priority

Each module that can request interrupts, including the QADC, has an interrupt arbitration number (IARB) field in its module configuration register. Each IARB field must have a different non-zero value. During an interrupt acknowledge cycle, IARB permits arbitration among simultaneous interrupts of the same priority level.

The reset value of IARB in the QADCMCR is $0. Initialization software must set the IARB field to a non-zero value in order for QADC interrupts to be arbitrated. Refer to **D.5.1 QADC Module Configuration Register** for more information.

8.7 Test Register

The QADC test register (QADCTEST) is used only during factory testing of the MCU.

8.8 General-Purpose I/O Port Operation

QADC port pins, when used as general-purpose input, are conditioned by a synchronizer with an enable feature. The synchronizer is not enabled until the QADC decodes an IMB bus cycle which addresses the port data register to minimize the high-current effect of mid-level signals on the inputs used for analog signals. Digital input signals must meet the input low voltage (V_{IL}) or input high voltage (V_{IH}) specifications in **APPENDIX A ELECTRICAL CHARACTERISTICS**. If an analog input pin does not meet the digital input pin specifications when a digital port read operation occurs, an indeterminate state is read.

During a port data register read, the actual value of the pin is reported when its corresponding bit in the data direction register defines the pin to be an input (port A only). When the data direction bit specifies the pin to be an output, the content of the port data register is read. By reading the latch which drives the output pin, software instructions that read data, modify it, and write the result, like bit manipulation instructions, work correctly.

In the CTM4, TBB2 is global and accessible to every submodule. TBB1 and TBB4 are split to form two local time base buses. **Table 10-1** shows which time base buses are available to each CTM4 submodule.

Table 10-1 CTM4 Time Base Bus Allocation

Submodule	Global/Local Time Base Bus Allocation		Submodule	Global/Local Time Base Bus Allocation	
	Global Bus A	Global Bus B		Global Bus A	Global Bus B
DASM9	TBB1	TBB2	MCSM 2	TBB4	TBB2
DASM10	TBB1	TBB2	DASM 3	TBB4	TBB2
MCSM 11	TBB1	TBB2	DASM 4	TBB4	TBB2
FCSM 12	TBB1	TBB2			

Each PWMSM has an independent 16-bit counter and 8-bit prescaler clocked by the PCLK1 signal, which is generated by the CPSM. The PWMSMs are not connected to any of the time base buses. Refer to **10.9 Pulse-Width Modulation Submodule (PWMSM)** for more information.

10.4 Bus Interface Unit Submodule (BIUSM)

The BIUSM connects the SMB to the IMB and allows the CTM4 submodules to communicate with the CPU32. The BIUSM also communicates CTM4 submodule interrupt requests to the IMB, and transfers the interrupt level, arbitration number and vector number to the CPU32 during the interrupt acknowledge cycle.

10.4.1 STOP Effect On the BIUSM

When the CPU32 STOP instruction is executed, only the CPU32 is stopped; the CTM4 continues to operate as normal.

10.4.2 Freeze Effect On the BIUSM

CTM4 response to assertion of the IMB FREEZE signal is controlled by the FRZ bit in the BIUSM configuration register (BIUMCR). Since the BIUSM propagates FREEZE to the CTM4 submodules via the SMB, the setting of FRZ affects all CTM4 submodules.

If the IMB FREEZE signal is asserted and FRZ = 1, all CTM4 submodules freeze. The following conditions apply when the CTM4 is frozen:

- All submodule registers can still be accessed.
- The CPSM, FCSM, MCSM, and PWMSM counters stop counting.
- The IN status bit still reflects the state of the FCSM external clock input pin.
- The IN2 status bit still reflects the state of the MCSM external clock input pin, and the IN1 status bit still reflects the state of the MCSM modulus load input pin.
- DASM capture and compare functions are disabled.
- The DASM IN status bit still reflects the state of its associated pin in the DIS, IPWM, IPM, and IC modes. In the OCB, OCAB, and OPWM modes, IN reflects the state of the DASM output flip flop.
- When configured for OCB, OCAB, or OPWM modes, the state of the DASM

When a match or input capture event requiring service occurs, the affected channel generates a service request to the scheduler. The scheduler determines the priority of the request and assigns the channel to the microengine at the first available time. The microengine performs the function defined by the content of the control store or emulation RAM, using parameters from the parameter RAM.

11.3.1 Event Timing

Match and capture events are handled by independent channel hardware. This provides an event accuracy of one time-base clock period, regardless of the number of channels that are active. An event normally causes a channel to request service. The time needed to respond to and service an event is determined by which channels and the number of channels requesting service, the relative priorities of the channels requesting service, and the microcode execution time of the active functions. Worst-case event service time (latency) determines TPU performance in a given application. Latency can be closely estimated. For more information, refer to the *TPU Reference Manual* (TPURM/AD)

11.3.2 Channel Orthogonality

Most timer systems are limited by the fixed number of functions assigned to each pin. All TPU channels contain identical hardware and are functionally equivalent in operation, so that any channel can be configured to perform any time function. Any function can operate on the calling channel, and, under program control, on another channel determined by the program or by a parameter. The user controls the combination of time functions.

11.3.3 Interchannel Communication

The autonomy of the TPU is enhanced by the ability of a channel to affect the operation of one or more other channels without CPU32 intervention. Interchannel communication can be accomplished by issuing a link service request to another channel, by controlling another channel directly, or by accessing the parameter RAM of another channel.

11.3.4 Programmable Channel Service Priority

The TPU provides a programmable service priority level to each channel. Three priority levels are available. When more than one channel of a given priority requests service at the same time, arbitration is accomplished according to channel number. To prevent a single high-priority channel from permanently blocking other functions, other service requests of the same priority are performed in channel order after the lowest-numbered, highest-priority channel is serviced.

11.3.5 Coherency

For data to be coherent, all available portions of the data must be identical in age, or must be logically related. As an example, consider a 32-bit counter value that is read and written as two 16-bit words. The 32-bit value is read-coherent only if both 16-bit portions are updated at the same time, and write-coherent only if both portions take

11.4.6 Period Measurement with Additional Transition Detect (PMA)

This function and the following function are used primarily in toothed-wheel speed-sensing applications, such as monitoring rotational speed of an engine. The period measurement with additional transition detect function allows for a special-purpose 23-bit period measurement. It can detect the occurrence of an additional transition (caused by an extra tooth on the sensed wheel) indicated by a period measurement that is less than a programmable ratio of the previous period measurement.

Once detected, this condition can be counted and compared to a programmable number of additional transitions detected before TCR2 is reset to $FFFF. Alternatively, a byte at an address specified by a channel parameter can be read and used as a flag. A non-zero value of the flag indicates that TCR2 is to be reset to $FFFF once the next additional transition is detected.

Refer to TPU programming note *Period Measurement, Additional Transition Detect (PMA) TPU Function* (TPUPN15A/D) for more information.

11.4.7 Period Measurement with Missing Transition Detect (PMM)

Period measurement with missing transition detect allows a special-purpose 23-bit period measurement. It detects the occurrence of a missing transition (caused by a missing tooth on the sensed wheel), indicated by a period measurement that is greater than a programmable ratio of the previous period measurement. Once detected, this condition can be counted and compared to a programmable number of additional transitions detected before TCR2 is reset to $FFFF. In addition, one byte at an address specified by a channel parameter can be read and used as a flag. A non-zero value of the flag indicates that TCR2 is to be reset to $FFFF once the next missing transition is detected.

Refer to TPU programming note *Period Measurement, Missing Transition Detect (PMM) TPU Function* (TPUPN15B/D) for more information.

11.4.8 Position-Synchronized Pulse Generator (PSP)

Any channel of the TPU can generate an output transition or pulse, which is a projection in time based on a reference period previously calculated on another channel. Both TCRs are used in this algorithm: TCR1 is internally clocked, and TCR2 is clocked by a position indicator in the user's device. An example of a TCR2 clock source is a sensor that detects special teeth on the flywheel of an automobile using PMA or PMM. The teeth are placed at known degrees of engine rotation; hence, TCR2 is a coarse representation of engine degrees. For example, each count represents some number of degrees.

Up to 15 position-synchronized pulse generator function channels can operate with a single input reference channel executing a PMA or PMM input function. The input channel measures and stores the time period between the flywheel teeth and resets TCR2 when the engine reaches a reference position. The output channel uses the period calculated by the input channel to project output transitions at specific engine degrees. Because the flywheel teeth might be 30 or more degrees apart, a fractional

TRAMBAR can be written only once after a reset. This prevents runaway software from accidentally re-mapping the array. Because the locking mechanism is activated by the first write after a reset, the base address field should be written in a single word operation. Writing only one-half of the register prevents the other half from being written.

12.4 TPURAM Privilege Level

The RASP field in TRAMMCR specifies whether access to the TPURAM can be made from supervisor mode only, or from either user or supervisor mode. If supervisor-only access is specified, an access from user mode is ignored by the TPURAM control logic and can be decoded externally. Refer to **4.7 Privilege Levels** and **5.5.1.7 Function Codes** for more information concerning privilege levels.

12.5 Normal Operation

During normal operation, the TPURAM control registers and array can be accessed by the CPU32, by byte, word, or long word. A byte or aligned word access takes one bus cycle (two system clock cycles). A long word access requires two bus cycles. Misaligned accesses are not permitted by the CPU32 and will result in an address error exception. Refer to **5.6 Bus Operation** for more information concerning access times. The TPU cannot access the array and has no effect on the operation of the TPURAM during normal operation.

12.6 Standby Operation

Standby mode maintains the RAM array when the MCU main power supply is turned off.

Relative voltage levels of the V_{DD} and V_{STBY} pins determine whether the TPURAM is in standby mode. TPURAM circuitry switches to the standby power source when specified limits are exceeded. The TPURAM is essentially powered by the power supply pin with the greatest voltage (for example, V_{DD} or V_{STBY}). If specified standby supply voltage levels are maintained during the transition, there is no loss of memory when switching occurs. The RAM array cannot be accessed while the TPURAM is powered from V_{STBY}. If standby operation is not desired, connect the V_{STBY} pin to the V_{SS} pin.

I_{SB} (SRAM standby current) may exceed specified maximum standby current during the time V_{DD} makes the transition from normal operating level to the level specified for standby operation. This occurs within the voltage range $V_{SB} - 0.5\ V \geq V_{DD} \geq V_{SS} + 0.5\ V$. Typically, I_{SB} peaks when $V_{DD} \approx V_{SB} - 1.5\ V$, and averages 1.0 mA over the transition period.

Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for standby switching and power consumption specifications.

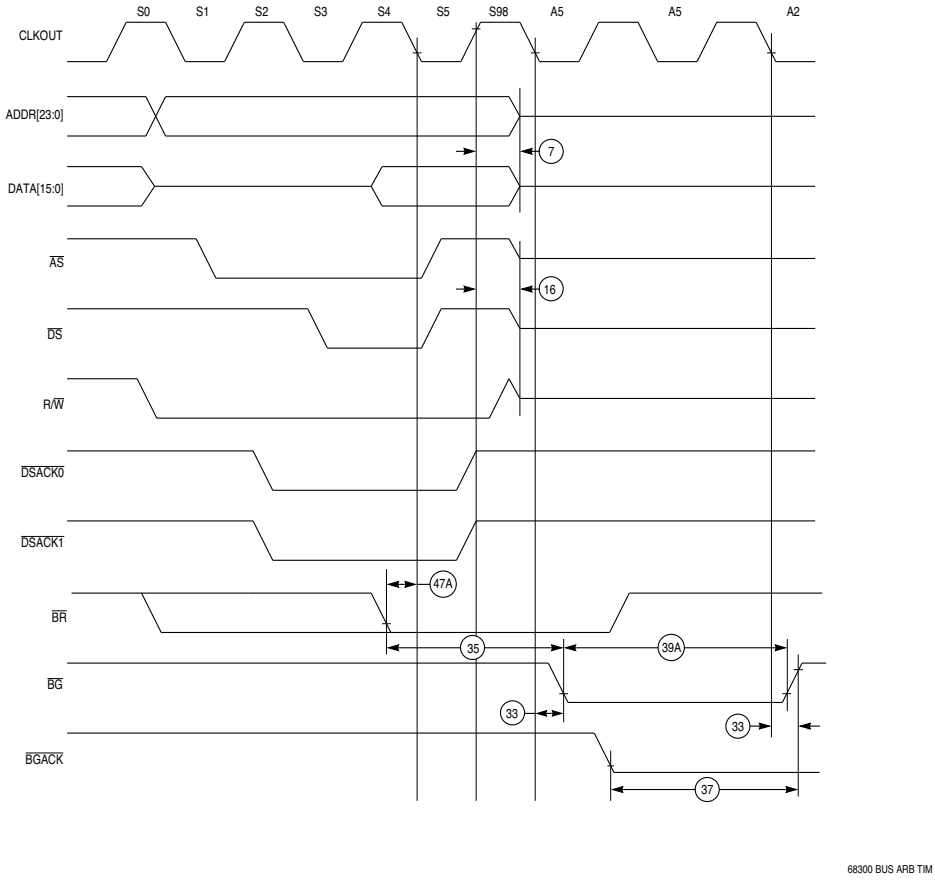


Figure A-8 Bus Arbitration Timing Diagram — Active Bus Case

D.3.2 RAM Test Register

RAMTST — RAM Test Register **$YFFB42**

Used for factory test only.

D.3.3 Array Base Address Register High

RAMBAH — Array Base Address Register High **$YFFB44**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16
RESET:								0	0	0	0	0	0	0	0

D.3.4 Array Base Address Register Low

RAMBAL — Array Base Address Register Low **$YFFB46**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 15	ADDR 14	ADDR 13	ADDR 12	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RAMBAH and RAMBAL specify the SRAM array base address in the system memory map. They can only be written while the SRAM is in low-power stop mode (STOP = 1, the default out of reset) and the base address lock is disabled (RLCK = 0, the default out of reset). This prevents accidental remapping of the array.

DSCK — PCS to SCK Delay

0 = PCS valid to SCK delay is one-half SCK.

1 = SPCR1 DSCKL[6:0] specifies delay from PCS valid to SCK.

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select an external device for serial data transfer. More than one peripheral chip select may be activated at a time, and more than one peripheral chip can be connected to each PCS pin, provided proper fanout is observed. PCS0 shares a pin with the slave select ($\overline{SS}$) signal, which initiates slave mode serial transfer. If $\overline{SS}$ is taken low when the QSPI is in master mode, a mode fault occurs.

D.10 TouCAN Module

The TouCAN is used only in the MC68376. **Table D-59** shows the TouCAN address map. The column labeled “Access” indicates the privilege level at which the CPU32 must be operating to access the register. A designation of “S” indicates that supervisor mode is required. A designation of “S/U” indicates that the register can be programmed for either supervisor mode access or unrestricted access.

TouCAN module address space is split, with 128 bytes starting at the base address, and an extra 256 bytes starting at the base address +128. The upper 256 are fully used for the message buffer structures. Of the lower 128 bytes, only part is occupied by various registers. Registers with bits marked as “reserved” should always be written as logic 0.

Table D-59 TouCAN Address Map

Access	Address[1]	15 8	7 0
S	$YFF080	TouCAN Module Configuration Register (CANMCR)	
S	$YFF082	TouCAN Test Configuration Register (CANTCR)	
S	$YFF084	TouCAN Interrupt Register (CANICR)	
S/U	$YFF086	Control Register 0 (CANCTRL0)	Control Register 1 (CANCTRL1)
S/U	$YFF088	Prescaler Divider Register (PRESDIV)	Control Register 2 (CANCTRL2)
S/U	$YFF08A	Free-Running Timer Register (TIMER)	
—	—	Reserved	
S/U	$YFF090	Receive Global Mask High (RXGMSKHI)	
S/U	$YFF092	Receive Global Mask Low (RXGMSKLO)	
S/U	$YFF094	Receive Buffer 14 Mask High (RX14MSKHI)	
S/U	$YFF096	Receive Buffer 14 Mask Low (RX14MSKLO)	
S/U	$YFF098	Receive Buffer 15 Mask High (RX15MSKHI)	
S/U	$YFF09A	Receive Buffer 15 Mask Low (RX15MSKLO)	
—	—	Reserved	
S/U	$YFF0A0	Error and Status Register (ESTAT)	
S/U	$YFF0A2	Interrupt Masks (IMASK)	
S/U	$YFF0A4	Interrupt Flags (IFLAG)	
S/U	$YFF0A6	Receive Error Counter (RXECTR)	Transmit Error Counter (TXECTR)

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in SIMCR.

–U–

–V–

–W–

–X–