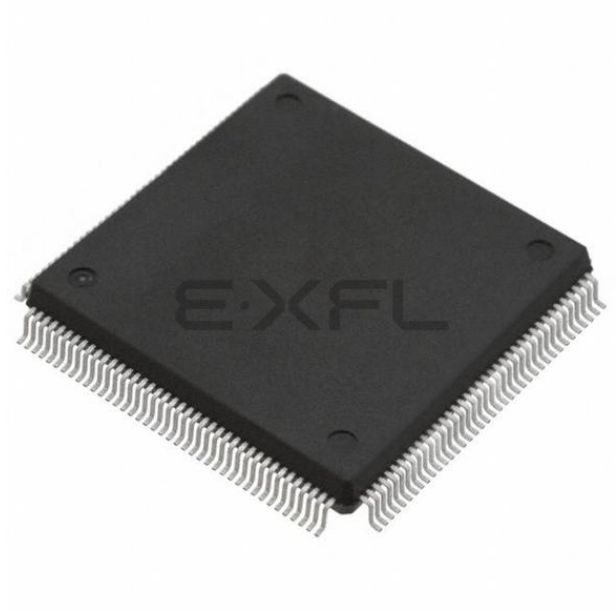


Welcome to E-XFL.COM

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"

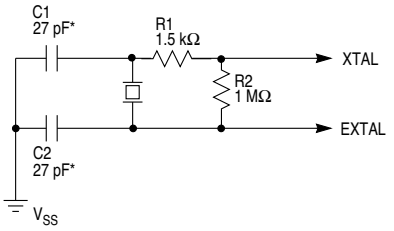
Details	
Product Status	Obsolete
Core Processor	CPU32
Core Size	32-Bit Single-Core
Speed	20MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	18
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	7.5K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data 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/mc68336gcab20

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

Figure	Title	Page

To generate a reference frequency using the crystal oscillator, a reference crystal must be connected between the EXTAL and XTAL pins. Typically, a 4.194 MHz crystal is used, but the frequency may vary between 1 and 6 MHz. **Figure 5-3** shows a typical circuit.



* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A KDS041-18 4.194 MHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

32 OSCILLATOR 4M

Figure 5-3 System Clock Oscillator Circuit

If a fast reference frequency is provided to the PLL from a source other than a crystal, or an external system clock signal is applied through the EXTAL pin, the XTAL pin must be left floating.

When an external system clock signal is applied (MODCLK = 0 during reset), the PLL is disabled. The duty cycle of this signal is critical, especially at operating frequencies close to maximum. The relationship between clock signal duty cycle and clock signal period is expressed as follows:

$$\text{Minimum External Clock Period} = \frac{\text{Minimum External Clock High/Low Time}}{50\% - \text{Percentage Variation of External Clock Input Duty Cycle}}$$

5.3.2 Clock Synthesizer Operation

V_{DDSYN} is used to power the clock circuits when the system clock is synthesized from either a crystal or an externally supplied reference frequency. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. A quiet power supply must be used as the V_{DDSYN} source. Adequate external bypass capacitors should be placed as close as possible to the V_{DDSYN} pin to assure a stable operating frequency. When an external system clock signal is applied and the PLL is disabled, V_{DDSYN} should be connected to the V_{DD} supply. Refer to the *SIM Reference Manual* (SIMRM/AD) for more information regarding system clock power supply conditioning.

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

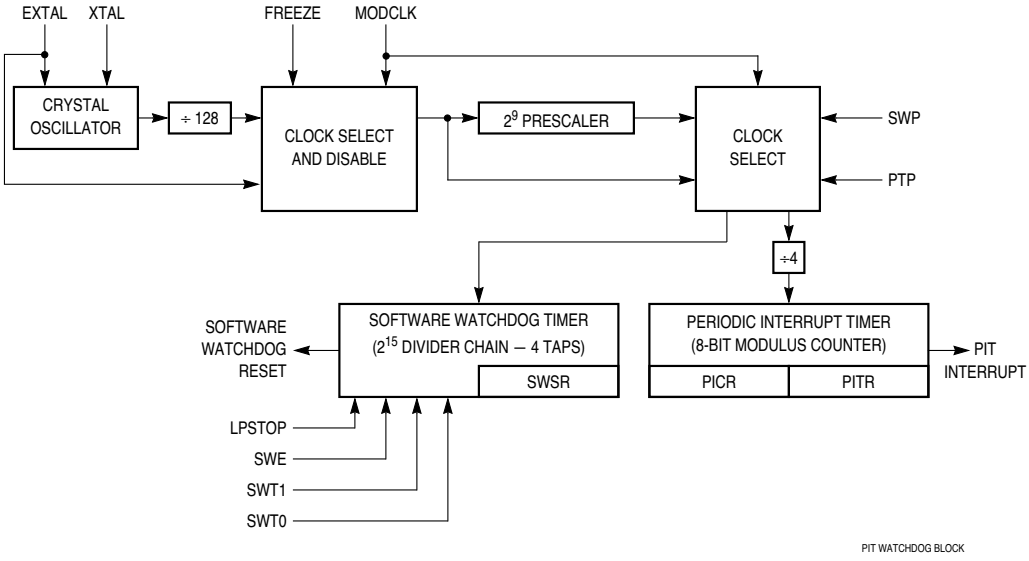


Figure 5-7 Periodic Interrupt Timer and Software Watchdog Timer

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)

If bus termination signals remain unasserted, the MCU will continue to insert wait states, and the bus cycle will never end. If no peripheral responds to an access, or if an access is invalid, external logic should assert the BERR or HALT signals to abort the bus cycle (when BERR and HALT are asserted simultaneously, the CPU32 acts as though only BERR is asserted). When enabled, the SIM bus monitor asserts BERR when DSACK response time exceeds a predetermined limit. The bus monitor timeout period is determined by the BMT[1:0] field in SYPCR. The maximum bus monitor timeout period is 64 system clock cycles.

5.6.2.1 Read Cycle

During a read cycle, the MCU transfers data from an external memory or peripheral device. If the instruction specifies a long-word or word operation, the MCU attempts to read two bytes at once. For a byte operation, the MCU reads one byte. The portion of the data bus from which each byte is read depends on operand size, peripheral address, and peripheral port size. **Figure 5-10** is a flowchart of a word read cycle. Refer to **5.5.2 Dynamic Bus Sizing**, **5.5.4 Misaligned Operands**, and the *SIM Reference Manual* (SIMRM/AD) for more information.

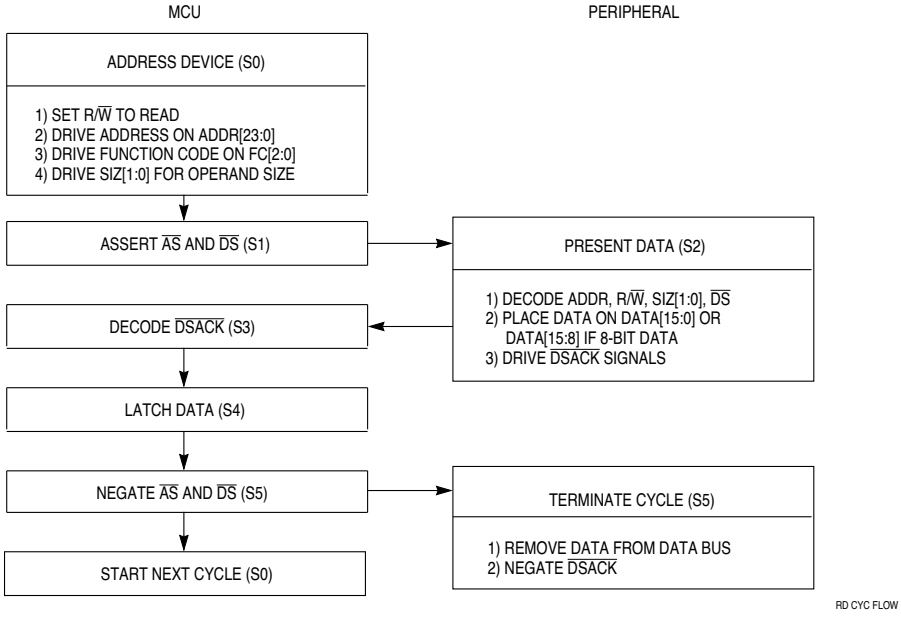


Figure 5-10 Word Read Cycle Flowchart

Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobe, or interrupt acknowledge signals. Chip-select logic can also generate $\overline{\text{DSACK}}$ and $\overline{\text{AVEC}}$ signals internally. A single $\overline{\text{DSACK}}$ generator is shared by all chip-selects. Each signal can also be synchronized with the ECLK signal available on ADDR23.

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low.

If a chip-select function is given the same address as a microcontroller module or an internal memory array, an access to that address goes to the module or array, and the chip-select signal is not asserted. The external address and data buses do not reflect the internal access.

All chip-select circuits are configured for operation out of reset. However, all chip-select signals except $\overline{\text{CSBOOT}}$ are disabled, and cannot be asserted until the BYTE[1:0] field in the corresponding option register is programmed to a non-zero value to select a transfer size. The chip-select option register must not be written until a base address has been written to a proper base address register. Alternate functions for chip-select pins are enabled if appropriate data bus pins are held low at the release of $\overline{\text{RESET}}$. Refer to **5.7.3.1 Data Bus Mode Selection** for more information. **Figure 5-20** is a functional diagram of a single chip-select circuit.

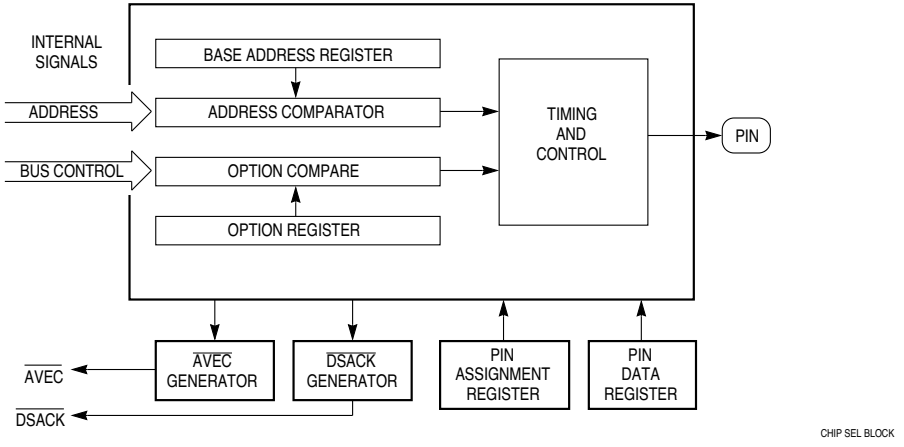


Figure 5-20 Chip-Select Circuit Block Diagram

$$\text{SCI Baud Rate} = \frac{\text{System Clock}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{\text{System Clock}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range {1, 2, 3, ..., 8191}.

The SCI receiver operates asynchronously. An internal clock is necessary to synchronize with an incoming data stream. The SCI baud rate generator produces a receive time sampling clock with a frequency 16 times that of the SCI baud rate. The SCI determines the position of bit boundaries from transitions within the received waveform, and adjusts sampling points to the proper positions within the bit period.

9.4.3.4 Parity Checking

The PT bit in SCCR1 selects either even (PT = 0) or odd (PT = 1) parity. PT affects received and transmitted data. The PE bit in SCCR1 determines whether parity checking is enabled (PE = 1) or disabled (PE = 0). When PE is set, the MSB of data in a frame is used for the parity function. For transmitted data, a parity bit is generated for received data; the parity bit is checked. When parity checking is enabled, the PF bit in the SCI status register (SCSR) is set if a parity error is detected.

Enabling parity affects the number of data bits in a frame, which can in turn affect frame size. **Table 9-6** shows possible data and parity formats.

Table 9-6 Effect of Parity Checking on Data Size

M	PE	Result
0	0	8 data bits
0	1	7 data bits, 1 parity bit
1	0	9 data bits
1	1	8 data bits, 1 parity bit

9.4.3.5 Transmitter Operation

The transmitter consists of a serial shifter and a parallel data register (TDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU32. The transmitter is double-buffered, which means that data can be loaded into the TDR while other data is shifted out. The TE bit in SCCR1 enables (TE = 1) and disables (TE = 0) the transmitter.

Shifter output is connected to the TXD pin while the transmitter is operating (TE = 1, or TE = 0 and transmission in progress). Wired-OR operation should be specified when more than one transmitter is used on the same SCI bus. The WOMS bit in SCCR1 determines whether TXD is an open-drain (wired-OR) output or a normal CMOS output. An external pull-up resistor on TXD is necessary for wired-OR operation. WOMS controls TXD function whether the pin is used by the SCI or as a general-purpose I/O pin.

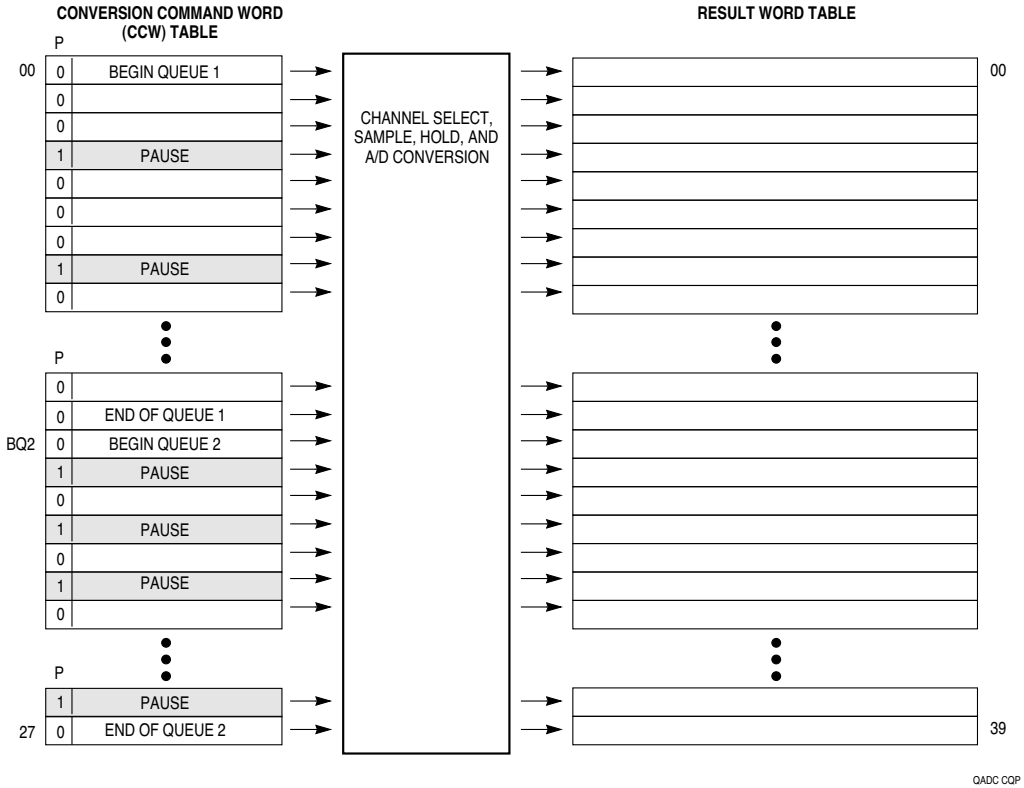


Figure 8-7 QADC Queue Operation with Pause

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

The choice of single-scan or continuous-scan applies to the full queue, and is not applied to each subqueue. Once a subqueue is initiated, each CCW is executed sequentially until the last CCW in the subqueue is executed and the pause state is entered. Execution can only continue with the next CCW, which is the beginning of the next subqueue. A subqueue cannot be executed a second time before the overall queue execution has been completed.

Trigger events which occur during the execution of a subqueue are ignored, except that the trigger overrun flag is set. When continuous-scan mode is selected, a trigger event occurring after the completion of the last subqueue (after the queue completion flag is set), causes execution to continue with the first subqueue, starting with the first CCW in the queue.

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

output flip-flop will remain unchanged.

- The state of the PWMSM output flip-flop will remain unchanged.

If the IMB FREEZE signal is asserted and FRZ = 0, the freeze condition is ignored, and all CTM4 submodules will continue to operate normally.

10.4.3 LPSTOP Effect on the BIUSM

When the CPU32 LPSTOP instruction is executed, the system clock is stopped. All dependent modules, including the CTM4, are shut down until low-power STOP mode is exited.

10.4.4 BIUSM Registers

The BIUSM contains a module configuration register, a time base register, and a test register. The BIUSM register block occupies the first four register locations in the CTM4 register space. All unused bits and reserved address locations return zero when read. Writes to unused bits and reserved address locations have no effect. Refer to **D.7.1 BIU Module Configuration Register**, **D.7.2 BIUSM Test Configuration Register**, and **D.7.3 BIUSM Time Base Register** for information concerning BIUSM register and bit descriptions.

10.5 Counter Prescaler Submodule (CPSM)

The counter prescaler submodule (CPSM) is a programmable divider system that provides the CTM4 counters with a choice of six clock signals (PCLK[1:6]) derived from the main MCU system clock. Five of these frequencies are derived from a fixed divider chain. The divide ratio of the last clock frequency is software selectable from a choice of four divide ratios.

The CPSM is part of the BIUSM. **Figure 10-2** shows a block diagram of the CPSM.

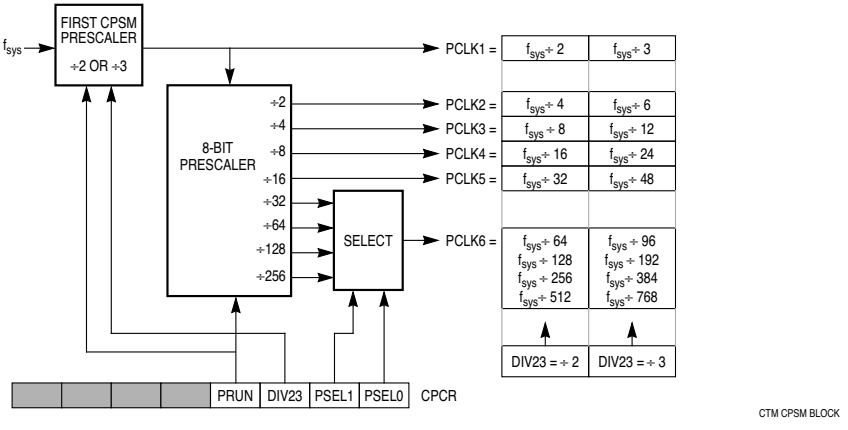


Figure 10-2 CPSM Block Diagram

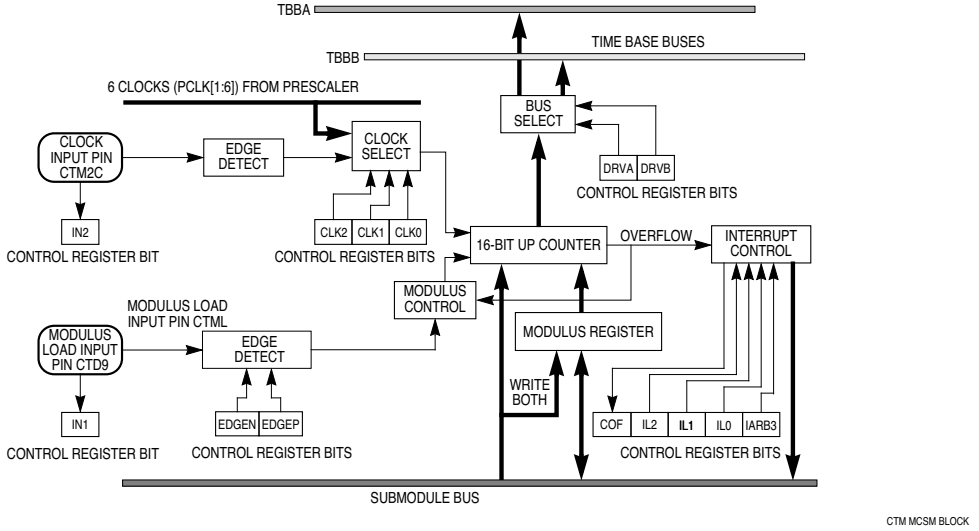


Figure 10-4 MCSM Block Diagram

10.7.1 MCSM Modulus Latch

The 16-bit modulus latch is a read/write register that is used to reload the counter automatically with a predetermined value. The contents of the modulus latch register can be read at any time. Writing to the register loads the modulus latch with the new value. This value is then transferred to the counter register when the next load condition occurs. However, writing to the corresponding counter register loads the modulus latch and the counter register immediately with the new value. The modulus latch register is cleared to $0000 by reset.

10.7.2 MCSM Counter

The counter is composed of a 16-bit read/write register associated with a 16-bit incrementer. Reading the counter transfers the contents of the counter register to the data bus. Writing to the counter loads the modulus latch and the counter register immediately with the new value.

10.7.2.1 Loading the MCSM Counter Register

The MCSM counter is loaded either by writing to the counter register or by loading it from the modulus latch when a counter overflow occurs. Counter overflow will set the COF bit in the MCSM status/interrupt/control register (MCSMSIC).

NOTE

When the modulus latch is loaded with $FFFF, the overflow flag is set on every counter clock pulse.

10.7.7 MCSM Registers

The MCSM contains a status/interrupt/control register, a counter, and a modulus latch. All unused bits and reserved address locations return zero when read. Writes to unused bits and reserved address locations have no effect. The CTM4 contains three MCSMs, each with its own set of registers. Refer to **D.7.8 MCSM Status/Interrupt/Control Registers**, **D.7.9 MCSM Counter Registers**, and **D.7.10 MCSM Modulus Latch Registers** for information concerning MCSM register and bit descriptions.

10.8 Double-Action Submodule (DASM)

The double-action submodule (DASM) allows two 16-bit input capture or two 16-bit output compare functions to occur automatically without software intervention. The input edge detector can be programmed to trigger the capture function on user-specified edges. The output flip flop can be set by one of the output compare functions, and reset by the other one. Interrupt requests can optionally be generated by the input capture and the output compare functions. The user can select one of two incoming time bases for the input capture and output compare functions.

Six operating modes allow the DASM input capture and output compare functions to perform pulse width measurement, period measurement, single pulse generation, and continuous pulse width modulation, as well as standard input capture and output compare. The DASM can also function as a single I/O pin.

DASM operating mode is determined by the mode select field (MODE[3:0]) in the DASM status/interrupt/control register (DASMSIC). **Table 10-2** shows the different DASM operating modes.

Table 10-2 DASM Modes of Operation

MODE[3:0]	Mode	Description of Mode
0000	DIS	Disabled — Input pin is high impedance; IN gives state of input pin
0001	IPWM	Input pulse width measurement — Capture on leading edge and the trailing edge of an input pulse
0010	IPM	Input period measurement — Capture two consecutive rising/falling edges
0011	IC	Input capture — Capture when the designated edge is detected
0100	OCB	Output compare, flag set on B compare — Generate leading and trailing edges of an output pulse and set the flag
0101	OCAB	Output compare, flag set on A and B compare — Generate leading and trailing edges of an output pulse and set the flag
0110	—	Reserved
0111	—	Reserved
1xxx	OPWM	Output pulse width modulation — Generate continuous PWM output with 7, 9, 11, 12, 13, 14, 15, or 16 bits of resolution

The DASM is composed of two timing channels (A and B), an output flip-flop, an input edge detector, some control logic and an interrupt interface. All control and status bits are contained in DASMSIC.

Channel A consists of one 16-bit data register and one 16-bit comparator. To the user, channel B also appears to consist of one 16-bit data register and one 16-bit compar-

Once debug mode is exited, the TouCAN will resynchronize with the CAN bus by waiting for 11 consecutive recessive bits before beginning to participate in CAN bus communication.

13.6.2 Low-Power Stop Mode

Before entering low-power stop mode, the TouCAN will wait for the CAN bus to be in an idle state, or for the third bit of intermission to be recessive. The TouCAN then waits for the completion of all internal activity (except in the CAN bus interface) to be complete. Afterwards, the following events occur:

- The TouCAN shuts down its clocks, stopping most internal circuits, thus achieving maximum power savings.
- The bus interface unit continues to operate, allowing the CPU32 to access the module configuration register.
- The TouCAN ignores its RX pins and drives its TX pins as recessive.
- The TouCAN loses synchronization with the CAN bus, and the STOPACK and NOTRDY bits in the module configuration register are set.

To exit low-power stop mode:

- Reset the TouCAN either by asserting one of the IMB reset lines or by asserting the SOFTRST bit CANMCR.
- Clear the STOP bit in CANMCR.
- The TouCAN module can optionally exit low-power stop mode via the self-wake mechanism. If the SELFWAKE bit in CANMCR was set at the time the TouCAN entered stop mode, then upon detection of a recessive to dominant transition on the CAN bus, the TouCAN clears the STOP bit in CANMCR and its clocks begin running.

When in low-power stop mode, a recessive to dominant transition on the CAN bus causes the WAKEINT bit in the error and status register (ESTAT) to be set. This event can generate an interrupt if the WAKEMSK bit in CANMCR is set.

Consider the following notes regarding low-power stop mode:

- When the self-wake mechanism activates, the TouCAN tries to receive the frame that woke it up. (It assumes that the dominant bit detected is a start-of-frame bit). It will not arbitrate for the CAN bus at this time.
- If the STOP bit is set while the TouCAN is in the bus off state, then the TouCAN will enter low-power stop mode and stop counting recessive bit times. The count will continue when STOP is cleared.
- To place the TouCAN in low-power stop mode with the self-wake mechanism engaged, write to CANMCR with both STOP and SELFWAKE set, then wait for the TouCAN to set the STOPACK bit.
- To take the TouCAN out of low-power stop mode when the self-wake mechanism is enabled, write to CANMCR with both STOP and SELFWAKE clear, then wait for the TouCAN to clear the STOPACK bit.
- The SELFWAKE bit should not be set after the TouCAN has already entered low-power stop mode.

Table A-9 QSPI Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H 200 pF load on all QSPI pins)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{QSPI}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	30 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	0 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	t_{ri} t_{ro}	— —	2 30	µs ns
14	Fall Time Input Output	t_{fi} t_{fo}	— —	2 30	µs ns

NOTES:
1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. For high time, n = External SCK rise time; for low time, n = External SCK fall time.

Table A-10 Time Processor Unit Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, f_{sys} = 20.97 MHz)[1, 2]

Num	Rating	Symbol	Min	Max	Unit
1	CLKOUT High to TPU Output Channel Valid[3, 4]	t_{CHTOV}	2	18	ns
2	CLKOUT High to TPU Output Channel Hold	t_{CHTOH}	0	15	ns
3	TPU Input Channel Pulse Width	t_{TIPW}	4	—	t_{cyc}

NOTES:
1. AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels.
2. Timing not valid for external T2CLK input.
3. Maximum load capacitance for CLKOUT pin is 90 pF.
4. Maximum load capacitance for TPU output pins is 100 pF.

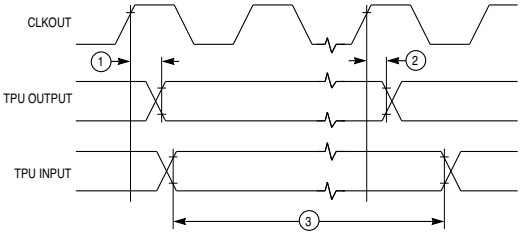


Figure A-20 TPU Timing Diagram

This field only affects the response of chip-selects and does not affect interrupt recognition by the CPU32.

$\overline{\text{AVEC}}$ — Autovector Enable

This field selects one of two methods of acquiring an interrupt vector during an interrupt acknowledge cycle. It is not usually used with a chip-select pin.

0 = External interrupt vector enabled
1 = Autovector enabled

If the chip select is configured to trigger on an interrupt acknowledge cycle (SPACE[1:0] = %00) and the $\overline{\text{AVEC}}$ field is set to one, the chip-select automatically generates $\overline{\text{AVEC}}$ in response to the interrupt acknowledge cycle. Otherwise, the vector must be supplied by the requesting device.

D.2.22 Master Shift Registers

TSTMSRA — Master Shift Register A **$YFFA30**

Used for factory test only.

TSTMSRB — Master Shift Register B **$YFFA32**

Used for factory test only.

D.2.23 Test Module Shift Count Register

TSTSC — Test Module Shift Count **$YFFA34**

Used for factory test only.

D.2.24 Test Module Repetition Count Register

TSTRC — Test Module Repetition Count **$YFFA36**

Used for factory test only.

D.2.25 Test Submodule Control Register

CREG — Test Submodule Control Register **$YFFA38**

Used for factory test only.

D.2.26 Distributed Register

DREG — Distributed Register **$YFFA3A**

Used for factory test only.

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.

BQ2[5:0] — Beginning of Queue 2

The BQ2 field indicates the location in the CCW table where queue 2 begins. The BQ2 field also indicates the end of queue 1 and thus creates an end-of-queue condition for queue 1.

D.5.7 QADC Status Register

QASR — Status Register **$YFFF210**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF1	PF1	CF2	PF2	TOR1	TOR2	QS[3:0]				CWP[5:0]					
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CF1 — Queue 1 Completion Flag

CF1 indicates that a queue 1 scan has been completed. CF1 is set by the QADC when the conversion is complete for the last CCW in queue 1, and the result is stored in the result table.

0 = Queue 1 scan is not complete.
1 = Queue 1 scan is complete.

PF1 — Queue 1 Pause Flag

PF1 indicates that a queue 1 scan has reached a pause. PF1 is set by the QADC when the current queue 1 CCW has the pause bit set, the selected input channel has been converted, and the result has been stored in the result table.

0 = Queue 1 has not reached a pause.
1 = Queue 1 has reached a pause.

CF2 — Queue 2 Completion Flag

CF2 indicates that a queue 2 scan has been completed. CF2 is set by the QADC when the conversion is complete for the last CCW in queue 2, and the result is stored in the result table.

0 = Queue 2 scan is not complete.
1 = Queue 2 scan is complete.

PF2 — Queue 2 Pause Flag

PF2 indicates that a queue 2 scan has reached a pause. PF2 is set by the QADC when the current queue 2 CCW has the pause bit set, the selected input channel has been converted, and the result has been stored in the result table.

0 = Queue 2 has not reached a pause.
1 = Queue 2 has reached a pause.

TOR1 — Queue 1 Trigger Overrun

TOR1 indicates that an unexpected queue 1 trigger event has occurred. TOR1 can be set only while queue 1 is active.

A trigger event generated by a transition on ETRIG1 may be recorded as a trigger overrun. TOR1 can only be set when using an external trigger mode. TOR1 cannot occur when the software initiated single-scan mode or the software initiated continuous-scan mode is selected.

D.10.9 Receive Global Mask Registers

RXGMSKHI — Receive Global Mask Register High **$YFF090**
RXGMSKLO — Receive Global Mask Register Low **$YFF092**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MID28	MID27	MID26	MID25	MID24	MID23	MID22	MID21	MID20	MID19	MID18	0	1	MID17	MID16	MID15
RESET:															
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MID14	MID13	MID12	MID11	MID10	MID9	MID8	MID7	MID6	MID5	MID4	MID3	MID2	MID1	MID0	0
RESET:															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

The receive global mask registers use four bytes. The mask bits are applied to all receive-identifiers, excluding receive-buffers 14-15, which have their own specific mask registers.

Base ID mask bits MID[28:18] are used to mask standard or extended format frames. Extended ID bits MID[17:0] are used to mask only extended format frames.

The RTR/SRR bit of a received frame is never compared to the corresponding bit in the message buffer ID field. However, remote request frames (RTR = 1) once received, are never stored into the message buffers. RTR mask bit locations in the mask registers (bits 20 and 0) are always zero, regardless of any write to these bits.

The IDE bit of a received frame is always compared to determine if the message contains a standard or extended identifier. Its location in the mask registers (bit 19) is always one, regardless of any write to this bit.

D.10.10 Receive Buffer 14 Mask Registers

RX14MSKHI — Receive Buffer 14 Mask Register High **$YFF094**
RX14MSKLO — Receive Buffer 14 Mask Register Low **$YFF096**

The receive buffer 14 mask registers have the same structure as the receive global mask registers and are used to mask buffer 14.

D.10.11 Receive Buffer 15 Mask Registers

RX15MSKHI — Receive Buffer 15 Mask Register High **$YFF098**
RX15MSKLO — Receive Buffer 15 Mask Register Low **$YFF09A**

The receive buffer 15 mask registers have the same structure as the receive global mask registers and are used to mask buffer 15.

–G–

–H–

–I–