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	33MHz
Connectivity	CANbus, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	48
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	10K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.25V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	217-BBGA
Supplier Device Package	217-PBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68f375mzp33r2

- The port logic provides up to 8 input-only and 8 bidirectional digital interface pins. Pins which are used as analog channels should be masked out of the digital data.

1.3.4 Analog Multiplexer – AMUX

The analog multiplexer (AMUX) submodule expands the channel capacity of the QADC64 analog-to-digital converter inputs by a maximum of 32 analog channels. 16 analog channels are bonded out on the MC68F375. The AMUX does not have an intermodule bus (IMB3) interface; control is through the QADC64. Refer to **5.13 Analog Multiplexer Submodule**.

1.3.5 Queued Serial Multi-Channel Communications Module – QSMCM

The queued serial multi-channel module (QSMCM) provides the microcontroller unit (MCU) with three serial communication interfaces divided into three submodules: the queued serial peripheral interface (QSPI) and two serial communications interfaces (SCI). These submodules communicate with the CPU via a common slave bus interface unit (SBIU). Refer to **SECTION 5 QUEUED ANALOG-TO-DIGITAL CONVERTER MODULE-64**.

The QSPI is a full-duplex, synchronous serial interface for communicating with peripherals and other MCUs. It is enhanced from the original QSM to include a total of 160 bytes of queue RAM to accommodate more receive, transmit, and control information.

The duplicate, independent, SCIs are full-duplex universal asynchronous receiver transmitter (UART) serial interface. The original QSM SCI is enhanced by the addition of an SCI, a common external baud clock source, receive and transmit buffers on one SCI.

1.3.6 TouCAN Module

The TouCAN module is a communication controller implementing the CAN protocol. It contains all the logic needed to implement the CAN2.0B protocol, supporting both standard ID format and extended ID. The protocol is a CSMA/CD type, with collision detection without loss, used mainly for vehicle systems communication and industrial applications. The module contains 16 message buffers used for transmit and receive, and masks used to qualify the received message ID before comparing it to the receive buffers. Refer to **SECTION 7 CAN 2.0B CONTROLLER MODULE**.

1.3.7 Enhanced Time Processing Unit – TPU3

The TPU3 is an intelligent, semi-autonomous co-processor designed for timing control. Operating simultaneously with the CPU, the TPU processes microinstructions, schedules and processes real-time hardware events, performs input and output, and accesses shared data without CPU intervention. Consequently, for each timer event the CPU setup and service time are minimized or eliminated.

The TPU3 can be viewed as a special-purpose microcomputer that performs a programmable series of two operations, match and capture. Each occurrence of either operation is called an event. A programmed series of events is called a function. TPU3

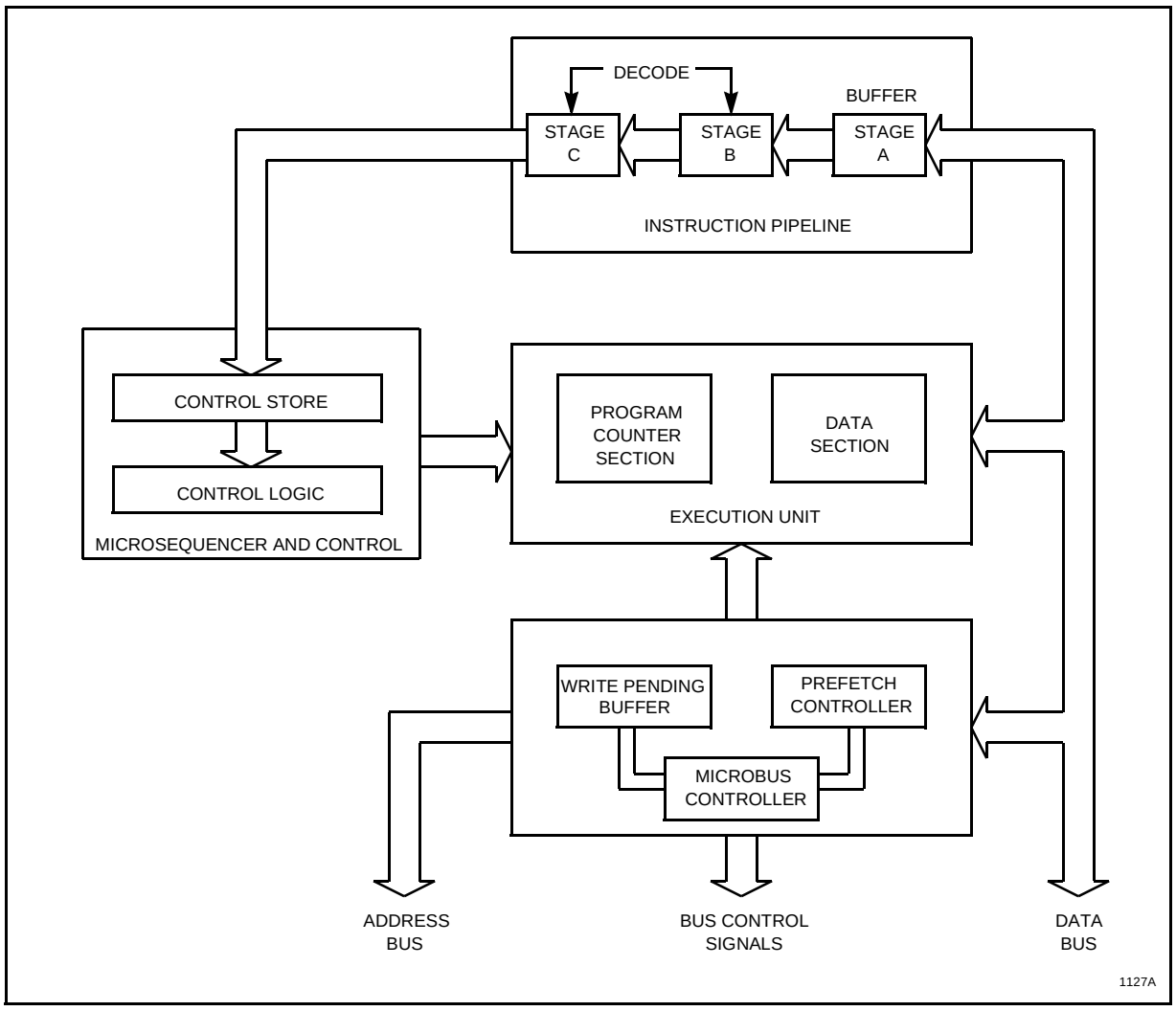

Figure 3-1 CPU32 Block Diagram

3.2 CPU32 Registers

The CPU32 programming model consists of two groups of registers that correspond to the user and supervisor privilege levels. User programs can use only the registers of the user model. The supervisor programming model, which supplements the user programming model, is used by CPU32 system programmers who wish to protect sensitive operating system functions. The supervisor model is identical to that of the MC68010 and later processors.

The CPU32 has eight 32-bit data registers, seven 32-bit address registers, a 32-bit program counter, separate 32-bit supervisor and user stack pointers, a 16-bit status register, two alternate function code registers, and a 32-bit vector base register. Refer to **Figure 3-2** and **Figure 3-3**.

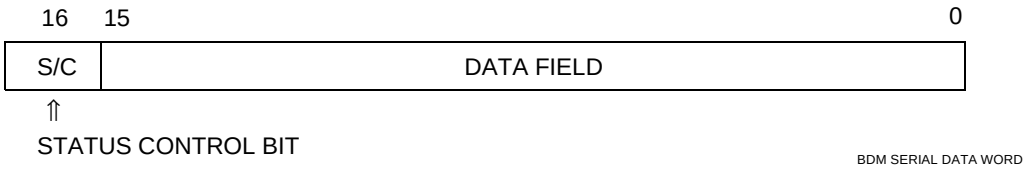

Figure 3-11 BDM Serial Data Word

Table 3-7 CPU Generated Message Encoding

Bit 16	Data	Message Type
0	XXXX	Valid data transfer
0	FFFF	Command complete; Status OK
1	0000	Not ready with response; Come again
1	0001	BERR terminated bus cycle; Data invalid
1	FFFF	Illegal command

Command and data transfers initiated by the development system should clear bit 16. The current implementation ignores this bit; however, Motorola reserves the right to use this bit for future enhancements.

3.10.10 Recommended BDM Connection

In order to provide for use of development tools when an MCU is installed in a system, Motorola recommends that appropriate signal lines be routed to a male Berg connector or double-row header installed on the circuit board with the MCU. Refer to **Figure 3-12**.

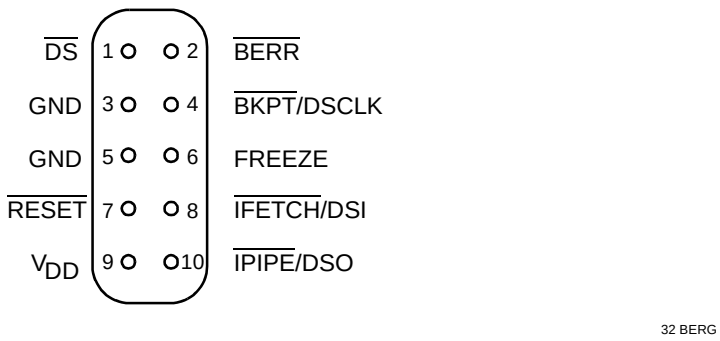

Figure 3-12 BDM Connector Pinout

3.10.11 Deterministic Opcode Tracking

CPU32 function code outputs are augmented by two supplementary signals to monitor the instruction pipeline. The instruction pipe (IPIPE) output indicates the start of each new instruction and each mid-instruction pipeline advance. The instruction fetch (IFETCH) output identifies the bus cycles in which the operand is loaded into the

In order to reset the ports to their post reset state listed in **Table 4-8**, an internal clock and the reset signal must be present. The clocks are generated with the SCIM2E voltage controlled oscillator (VCO). The VCO is biased to operate at approximately eight KHz whenever the crystal oscillator is not detected. This feature causes the VCO to start before the crystal oscillator. (See **APPENDIX E ELECTRICAL CHARACTERISTICS** for the exact frequencies.)

Table 4-8 Port Reset Condition

Port	State of Pins after Reset
A[1]	Input
B	Input
G	Input
H	Input
E	Input
F	Input
C	Output (PC[6:2, 0]) are driven high
PQS0, PQS1, PQS2	Input
PQA, PQB	Input
TPU3	Input

NOTES:
1. Each port requires approximately 4 clocks to assume their post reset state. The VCO startup time is no more than 15 msec after VDD reaches minimum value.

Only single byte or aligned word writes on the IMB to the RAM module will be guaranteed to complete without data corruption for synchronous resets. A long-word write, a misaligned operand write, a write to a peripheral module other than the RAM or a read cycle are not guaranteed. External writes are also guaranteed to complete, provided the external configuration logic on the data bus is conditioned by R/W. Asynchronous reset sources usually indicate a catastrophic failure and require the reset control logic to assert reset to the system immediately.

4.3.8.6 Low Power Operation

Low power operation is initiated by the CPU32. To reduce power consumption selectively, the CPU32 can set the STOP bits in each module configuration register. To minimize overall microcontroller power consumption, the CPU32 can execute the LPSTOP instruction which causes the SCIM2E to turn off the system clock.

A loss of clock will be recognized while the part is in low power stop, unless the RC oscillator is disabled. If it is disabled, external RESET will re-enable it so that RESET will be recognized. If a loss of clock occurs in LPSTOP mode and RSTEN=0, the part will continue to operate normally on the alternate clock. LPSTOP can then be exited normally, either by an interrupt request or by external RESET. If RSTEN=1 in LPSTOP mode, the loss of clock event will cause reset. For more information, see **4.4.9 Low Power Stop Mode**.

variations in the output of the RC oscillator due to processing and/or operating conditions.

The loss of clock detector can be disabled by writing a one to the loss of clock oscillator bit (LOSCD). This disables the free-running RC oscillator and prevents f_{ref} loss from being detected. The reset state of LOSCD is zero which enables the RC oscillator and loss of clock detector.

The reset enable bit (RSTEN) determines how the MCU will process a loss of clock detection. The default state out of reset for RSTEN is zero. This forces the clock synthesizer into the limp mode operating state. In limp mode, the RC oscillator used by the loss of clock detector provides the system clock. Limp mode frequency varies from device-to-device but does not exceed one-half the maximum system clock frequency.

When set to one, RSTEN allows the clock synthesizer to reset the MCU when the loss of clock detector triggers. After powering-up from a loss of clock reset, the MCU will set the LOC bit in the reset status register (RSR) and begin operation in limp mode.

The limp status bit (SLIMP) in SYNCR indicates that f_{ref} has failed and that the MCU has entered limp mode. SLIMP will remain set until normal f_{ref} operation is restored.

4.4 System Protection

The system protection block reports reset status information, monitors internal bus activity, and provides periodic interrupt generation. **Figure 4-7** is a block diagram of the submodule.

RESET is released. Mode select inputs are driven to the appropriate states at this time. Use an active circuit, such as that shown in **Figure 4-19**, for this purpose. **Table 4-24** shows the state of SCIM2E pins during reset.

Table 4-24 SCIM2E Pin States During Reset

Pin(s)	Pin State During RESET
ADDR[2:0]	High-Z
ADDR[10:3]/PB[7:0]	High-Z
ADDR[18:11]/PA]7:0]	High-Z
ADDR[22:19]/CS[9:6]/PC[6:3]	V_{DD}
ADDR23/CS10/ECLK	V_{DD}
AS/PE5	High-Z
AVEC/PE2	High-Z
BERR	Mode Select Input
BG/CSM	V_{DD}
BGACK/CSE	V_{DD}
BR/CS0	V_{DD}
CLKOUT	Output
CSBOOT	V_{DD}
DATA[7:0]/PH[7:0]	Mode Select Inputs
DATA[15:8]/PG[7:0]	Mode Select Inputs
DS/PE4	High-Z
DSACK0/PE0	High-Z
DSACK1/PE1	High-Z
FASTREF/PF0	Mode Select Input
FC0/CS3/PC0	V_{DD}
FC1/PC1	V_{DD}
FC2/CS3/PC2	V_{DD}
HALT	High-Z
IRQ[7:1]/PF[7:1]	High-Z
PE3	High-Z
R/W	High-Z
RESET	Asserted
SIZ[1:0]/PE[7:6]	High-Z
TSC	Three State Enable Input

4.7.7.2 Reset States of Pins Assigned to Other MCU Modules

As a rule, module pins that can be configured for general purpose I/O go into a high-impedance state during reset. However, during power-on reset, module port pins may be in an indeterminate state for a short period of time. Refer to **4.7.6 Power-On Reset** for more information.

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 DSACK and AVEC signals internally. 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. Chip-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 signals except CSBOOT are disabled after the release of RESET, and cannot be asserted until the R/W[1:0] and BYTE[1:0] fields in the corresponding option register are programmed to non-zero values. CSBOOT is automatically enabled out of reset in 8-bit and 16-bit expanded modes. Alternate functions for chip-select pins are enabled if appropriate data bus pins are held low at the release of RESET. Refer to **4.7.8.2 Data Bus Mode Selection** for more information. **Figure 4-22** is a functional diagram of a single chip-select circuit.

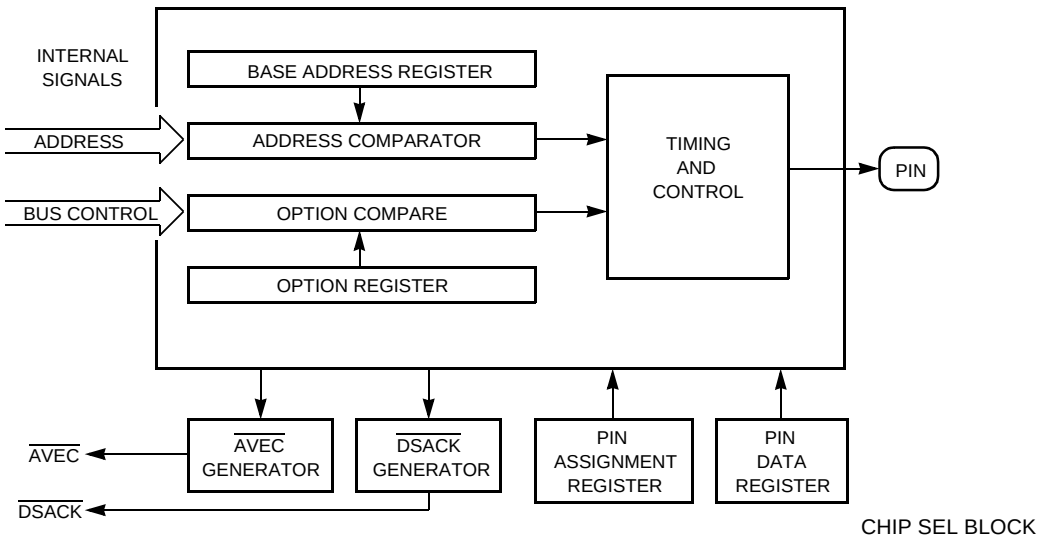

Figure 4-22 Chip-Select Circuit Block Diagram

4.9.1 Chip-Select Pin Assignment Register

The pin assignment registers contain twelve 2-bit fields that determine the functions of the chip-select pins. Each pin has two or three possible functions, as shown in **Table 4-31** and **Table 4-32**.

Table 4-40 Chip-Select Base and Option Register Reset Values

Fields	Reset Values
Base address	0x000000
Block size	2 Kbytes
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 initialization values from the reset vector, beginning at 0x000000 in supervisor program space. The $\overline{CSBOOT}$ chip-select signal is enabled and can select an external boot device mapped to a base address of 0x000000.

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, a port size of eight bits is selected. When DATA0 is held high during reset, a port size of 16 bits is selected. DATA0 has a weak internal pull-up device, so that a 16-bit port is selected by default out of reset. As mentioned above, the internal pull-up device can be overcome by bus loading effects. To ensure a particular configuration out of reset, use a pull-up resistor or an active device to place 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 0x000000 is made, an address match occurs, and the $\overline{CSBOOT}$ signal is asserted. The block size field in CSBARBT has a reset value of 0b111 (one Mbyte on CPU32-based MCUs and 512 Kbytes on CPU16-based MCUs). **Table 4-41** shows $\overline{CSBOOT}$ reset values.

Table 6-12 QSPI Register Map

Access[1]	Address	MSB[2]	LSB
S/U	0xYF FC18	QSPI Control Register 0 (SPCR0) See **Table 6-13** for bit descriptions.	
S/U	0xYF FC1A	QSPI Control Register 1 (SPCR1) See **Table 6-15** for bit descriptions.	
S/U	0xYF FC1C	QSPI Control Register 2 (SPCR2) See **Table 6-16** for bit descriptions.	
S/U	0xYF FC1E/ 0xYF FC1F	QSPI Control Register 3 (SPCR3) See **Table 6-17** for bit descriptions.	QSPI Status Register (SPSR) See **Table 6-18** for bit descriptions.
S/U	0xYF FD40 – 0xYF FD7F	Receive Data RAM (32 half-words)	
S/U	0xYF FD80 – 0xYF FDBF	Transmit Data RAM (32 half-words)	
S/U	0xYF FDC0 – 0xYF FDDF	Command RAM (32 bytes)	

NOTES:
1. S = Supervisor access only
   S/U = Supervisor access only or unrestricted user access (assignable data space).
2. 8-bit registers, such as SPCR3 and SPSR, are on 8-bit boundaries. 16-bit registers such as SPCR0 are on 16-bit boundaries.

To ensure proper operation, set the QSPI enable bit (SPE) in SPCR1 only after initializing the other control registers. Setting this bit starts the QSPI.

Rewriting the same value to a control register does not affect QSPI operation with the exception of writing NEWQP in SPCR2. Rewriting the same value to these bits causes the RAM queue pointer to restart execution at the designated location.

Before changing control bits, the user should halt the QSPI. Writing a different value into a control register other than SPCR2 while the QSPI is enabled may disrupt operation. SPCR2 is buffered, preventing any disruption of the current serial transfer. After the current serial transfer is completed, the new SPCR2 value becomes effective.

6.7.1.1 QSPI Control Register 0

SPCR0 contains parameters for configuring the QSPI before it is enabled. The CPU has read/write access to SPCR0, but the QSPI has read access only. SPCR0 must be initialized before QSPI operation begins. Writing a new value to SPCR0 while the QSPI is enabled disrupts operation.

SPCR0 — QSPI Control Register 0 **0xYF FC18**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
MSTR	WOMQ	BITS				CPOL	CPHA	SPBR							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

operation. Any data to be transmitted should be written into transmit RAM before the QSPI is enabled. During wraparound operation, data for subsequent transmissions can be written at any time.

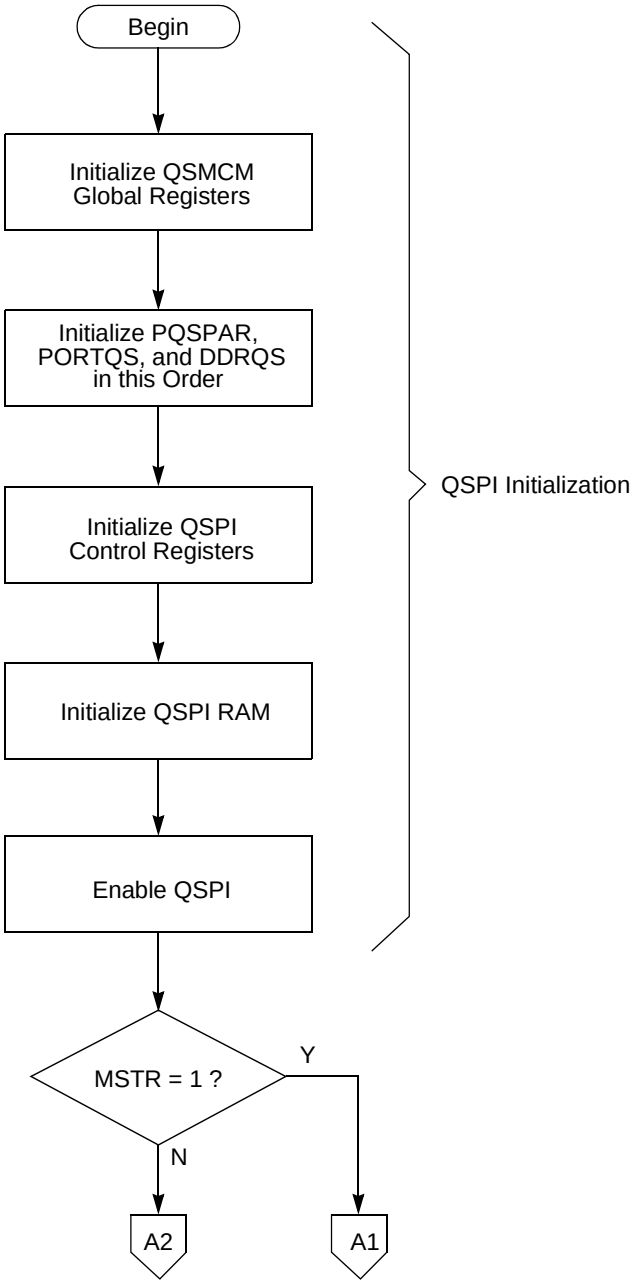

Figure 6-4 Flowchart of QSPI Initialization Operation

Some serial communication systems require a mark on the TXD pin even when the transmitter is disabled. Configure the TXD pin as an output, then write a one to either QDTX1 or QDTX2 of the PORTQS register. See **6.6.1**. When the transmitter releases control of the TXD pin, it reverts to driving a logic one output.

To insert a delimiter between two messages, to place non-listening receivers in wake-up mode between transmissions, or to signal a re-transmission by forcing an idle-line, clear and then set TE before data in the serial shifter has shifted out. The transmitter finishes the transmission, then sends a preamble. After the preamble is transmitted, if TDRE is set, the transmitter marks idle. Otherwise, normal transmission of the next sequence begins.

Both TDRE and TC have associated interrupts. The interrupts are enabled by the transmit interrupt enable (TIE) and transmission complete interrupt enable (TCIE) bits in SCCxR1. Service routines can load the last data frame in a sequence into SCxDR, then terminate the transmission when a TDRE interrupt occurs.

Two SCI messages can be separated with minimum idle time by using a preamble of 10 bit-times (11 if a 9-bit data format is specified) of marks (logic ones). Follow these steps:

1. Write the last data frame of the first message to the TDRx
2. Wait for TDRE to go high, indicating that the last data frame is transferred to the transmit serial shifter
3. Clear TE and then set TE back to one. This queues the preamble to follow the stop bit of the current transmission immediately.
4. Write the first data frame of the second message to register TDRx

In this sequence, if the first data frame of the second message is not transferred to TDRx prior to the finish of the preamble transmission, then the transmit data line (TXDx pin) marks idle (logic one) until TDRx is written. In addition, if the last data frame of the first message finishes shifting out (including the stop bit) and TE is clear, TC goes high and transmission is considered complete. The TXDx pin reverts to being a general-purpose output pin.

6.8.7.6 Receiver Operation

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

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

Table 7-1 Common Extended/Standard Format Frames

Field	Description
Time Stamp	Contains a copy of the high byte of the free running timer, which is captured at the beginning of the identifier field of the frame on the CAN bus.
Code	Refer to Table 7-2 and Table 7-3.
RX Length	Length (in bytes) of the RX data stored in offset 0x6 through 0xD of the buffer.This field is written by the TouCAN module, copied from the DLC (data length code) field of the received frame.
TX Length	Length (in bytes) of the data to be transmitted, located in offset $6 through 0xD of the buffer. This field is written by the CPU and is used as the DLC field value. If RTR (remote transmission request) = 1, the frame is a remote frame and will be transmitted without data field, regardless of the value in TX length.
Data	This field can store up to eight data bytes for a frame. For RX frames, the data is stored as it is received from the bus. For TX frames, the CPU provides the data to be transmitted within the frame.
Reserved	The CPU controls access to this word entry field (16 bits).

Table 7-2 Message Buffer Codes for Receive Buffers

RX Code Before RX New Frame	Description	RX Code After RX New Frame	Comment
0000	NOT ACTIVE — message buffer is not active.	—	—
0100	EMPTY — message buffer is active and empty.	0010	—
0010	FULL — message buffer is full.	0110	If a CPU read occurs before the new frame, new receive code is 0010.
0110	OVERRUN — second frame was received into a full buffer before the CPU read the first one.		
0XY1[1]	BUSY — message buffer is now being filled with a new receive frame. This condition will be cleared within 20 cycles.	0010	An empty buffer was filled (XY was 10).
		0110	A full/overrun buffer was filled (Y was 1).

NOTES:
1. For TX message buffers, upon read, the BUSY bit should be ignored.

Table 7-3 Message Buffer Codes for Transmit Buffers

RTR	Initial TX Code	Description	Code After Successful Transmission
X	1000	Message buffer not ready for transmit.	—
0	1100	Data frame to be transmitted once, unconditionally.	1000
1	1100	Remote frame to be transmitted once, and message buffer becomes an RX message buffer for data frames.	0100
0	1010[1]	Data frame to be transmitted only as a response to a remote frame, always.	1010
0	1110	Data frame to be transmitted only once, unconditionally, and then only as a response to remote frame, always.	1010

NOTES:
1. When a matching remote request frame is detected, the code for such a message buffer is changed to be 1110.

- Increment by 8 (RX error counter also increments by one)
- Decrement by one
- Avoid decrement when equal to zero
- RX error counter reset to a value between 119 and 127 inclusive, when the TouCAN transitions from error passive to error active
- Following reset, both counters reset to zero
- Detect values for error passive, bus off and error active transitions
- Cascade usage of TX error counter with an additional internal counter to detect the 128 occurrences of 11 consecutive recessive bits necessary to transition from bus off into error active.

Both counters are read only (except in test/freeze/halt modes).

The TouCAN responds to any bus state as described in the CAN protocol, transmitting an error active or error passive flag, delaying its transmission start time (error passive) and avoiding any influence on the bus when in the bus off state. The following are the basic rules for TouCAN bus state transitions:

- If the value of the TX error counter or RX error counter increments to a value greater than or equal to 128, the fault confinement state (FCS[1:0]) field in the error status register is updated to reflect an error passive state.
- If the TouCAN is in an error passive state, and either the TX error counter or RX error counter decrements to a value less than or equal to 127 while the other error counter already satisfies this condition, the FCS[1:0] field in the error status register is updated to reflect an error active state.
- If the value of the TX error counter increases to a value greater than 255, the FCS[1:0] field in the error status register is updated to reflect a bus off state, and an interrupt may be issued. The value of the TX error counter is reset to zero.
- If the TouCAN is in the bus off state, the TX error counter and an additional internal counter are cascaded to count 128 occurrences of 11 consecutive recessive bits on the bus. To do this, the TX error counter is first reset to zero, and then the internal counter begins counting consecutive recessive bits. Each time the internal counter counts 11 consecutive recessive bits, the TX error counter is incremented by one and the internal counter is reset to zero. When the TX error counter reaches the value of 128, the FCS[1:0] field in the error status register is updated to be error active, and both error counters are reset to zero. Any time a dominant bit is detected following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero but does not affect the TX error counter value.
- If only one node is operating in a system, the TX error counter is incremented with each message it attempts to transmit, due to the resulting acknowledgment errors. However, acknowledgment errors never cause the TouCAN to change from the error passive state to the bus off state.
- If the RX error counter increments to a value greater than 127, it stops incrementing, even if more errors are detected while being a receiver. After the next successful message reception, the counter is reset to a value between 119 and 127, to enable a return to the error active state.

NOTE

A word (32-bit) write will be completed coherently only if the reset occurs during the second (16-bit) write bus cycle. If reset occurs during the first write bus cycle, only the first half word will be written to the RAM array and the second write will not be allowed to occur. In this case, the word data contained in the DPTRAM will not be coherent. The first half word will contain the most significant half of the new word information and the second half word will contain the least significant half of the old word information.

If a reset is generated by an asynchronous reset such as the loss of clocks or software watchdog time-out, the contents of the RAM array are not guaranteed. (Refer to **4.7 Reset** for a description of the MC68F375 reset sources, operation, control, and status.)

Reset will also reconfigure some of the fields and bits in the DPTRAM control registers to their default reset state. See the description of the control registers to determine the effect of reset on these registers.

9.5.4 Stop Operation

Setting the STOP control bit in the DPTMCR causes the module to enter its lowest power-consuming state. The DPTMCR can still be written to allow the STOP control bit to be cleared.

In stop mode, the DPTRAM array cannot be read or written. All data in the array is retained. The BIU continues to operate to allow the CPU to access the STOP bit in the DPTMCR. The system clock remains stopped until the STOP bit is cleared or the DPTRAM module is reset.

The STOP bit is initialized to logical zero during reset. Only the STOP bit in the DPTMCR can be accessed while the STOP bit is asserted. Accesses to other DPTRAM registers may result in unpredictable behavior. Note also that the STOP bit should be set and cleared independently of the other control bits in this register to guarantee proper operation. Changing the state of other bits while changing the state of the STOP bit may result in unpredictable behavior.

9.5.5 Freeze Operation

The FREEZE line on the IMB3 has no effect on the DPTRAM module. When the freeze line is set, the DPTRAM module will operate in its current mode of operation. If the DPTRAM module is not disabled, (RAMDS = 0), it may be accessed via the IMB3. If the DPTRAM array is being used by the TPU in emulation mode, the DPTRAM will still be able to be accessed by the TPU microengine.

9.5.6 TPU3 Emulation Mode Operation

To emulate TPU3 time functions, the user stores the microinstructions required for all time functions to be used, in the RAM array. This must be done with the DPTRAM in its normal operating mode and accessible from the IMB3. After the time functions are

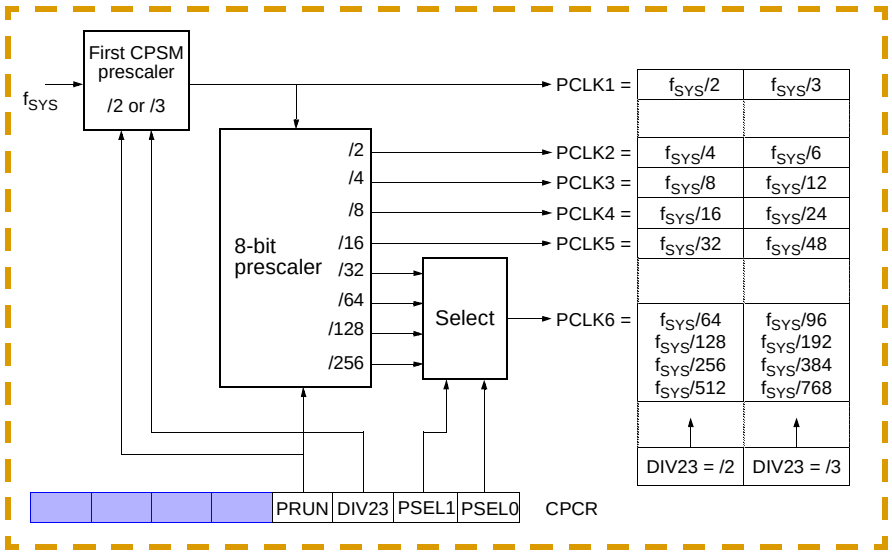

Figure 13-14 CPSM Block Diagram

13.9.1 Freeze Action on the CPSM

When the IMB FREEZE signal is recognized, the CPSM counters stop counting and remain set at their current values. When the FREEZE signal is negated, the counters start incrementing from their current values, as if nothing had happened. All registers are accessible during freeze.

13.9.2 CPSM Registers

The CPSM register map comprises four 16-bit register locations. As shown in **Table 13-20**, the register block contains two CPSM registers and two reserved registers. The CPSM register block always immediately follows the BIUSM register block in the CPSM register map. All unused bits and reserved address locations return zero when read by the software. Writing to unused bits and reserved address locations has no effect.

Table 13-20 CPSM Register Map

Address	15 8	7 0
0xYF F208	CPSM control register (CPCR)	
0xYF F20A	CPSM test register (CPTR)	

Table A-6 Overlay SRAM Modules (Static Random Access Memory)

Address	Access	Symbol	Register	Size	Reset
SRAM1					
0xYF F840	S	RAMMCR1	SRAM1 Module Configuration Register. See **Table 11-1** for bit descriptions.	16	X
0xYF F842	T	RAMTST1	SRAM1 Test Register.	16	X
0xYF F844	S	RAMBAH1	SRAM1 Base Address High Register. See **Table 11-3** for bit descriptions.	16	X
0xYF F846	S	RAMBAL1	SRAM1 Base Address Low Register. See **Table 11-3** for bit descriptions.	16	X
SRAM2					
0xYF F848	S	RAMMCR2	SRAM2 Module Configuration Register. See **Table 11-1** for bit descriptions.	16	X
0xYF F84A	T	RAMTST2	SRAM2 Test Register.	16	X
0xYF F84C	S	RAMBAH2	SRAM2 Base Address High Register. See **Table 11-3** for bit descriptions.	16	X
0xYF F84E	S	RAMBAL2	SRAM2 Base Address Low Register. See **Table 11-3** for bit descriptions.	16	X
SRAM3					
0xYF F848	S	RAMMCR3	SRAM3 Module Configuration Register. See **Table 11-1** for bit descriptions.	16	X
0xYF F84A	T	RAMTST3	SRAM3 Test Register.	16	X
0xYF F84C	S	RAMBAH3	SRAM3 Base Address High Register. See **Table 11-3** for bit descriptions.	16	X
0xYF F84E	S	RAMBAL3	SRAM3 Base Address Low Register. See **Table 11-3** for bit descriptions.	16	X
SRAM4					
0xYF F848	S	RAMMCR4	SRAM4 Module Configuration Register. See **Table 11-1** for bit descriptions.	16	X
0xYF F84A	T	RAMTST4	SRAM4 Test Register.	16	X
0xYF F84C	S	RAMBAH4	SRAM4 Base Address High Register. See **Table 11-3** for bit descriptions.	16	X
0xYF F84E	S	RAMBAL4	SRAM4 Base Address Low Register. See **Table 11-3** for bit descriptions.	16	X

Table A-7 DPTRAM (Dual-Port TPU RAM)

Address	Access	Symbol	Register	Size	Reset
0xYF F880	S	DPTMCR	DPT Module Configuration Register. See **Table 9-2** for bit descriptions.	16	X
0xYF F882	T	DPTTCR	DPT Test Register.	16	X
0xYF F884	S	DPTBAR	DPT Array Address Register. See **Table 9-3** for bit descriptions.	16	X
0xYF F886	S	MISRH	DPT Multiple Input Signature Register High. See **9.4.4 MISR High (MISRH) and MISR Low (MISRL)** for bit descriptions.	16	X

–R–

–S–

D.6 Universal Asynchronous Receiver/Transmitter (UART)

The UART function uses one or two TPU channels to provide asynchronous communications. Data word length is programmable from one to 14 bits. The function supports detection or generation of even, odd, and no parity. Baud rate is freely programmable and can be higher than 100 Kbaud. Eight bi-directional UART channels running in excess of 9600 baud could be implemented on the TPU.

Figure D-7 and **Figure D-4** show all of the host interface areas for the UART function in transmitting and receiving modes, respectively.

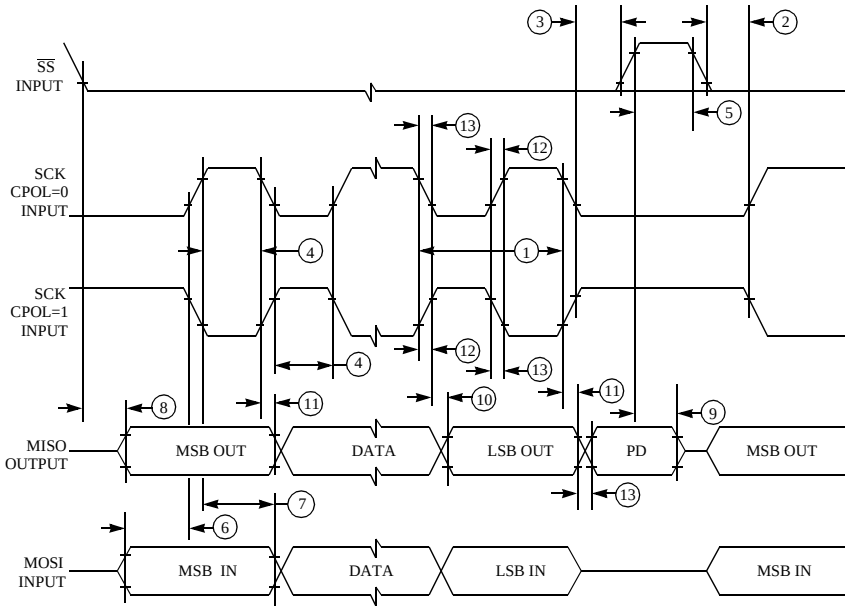

Figure E-18 QSPI Timing — Slave, CPHA = 0

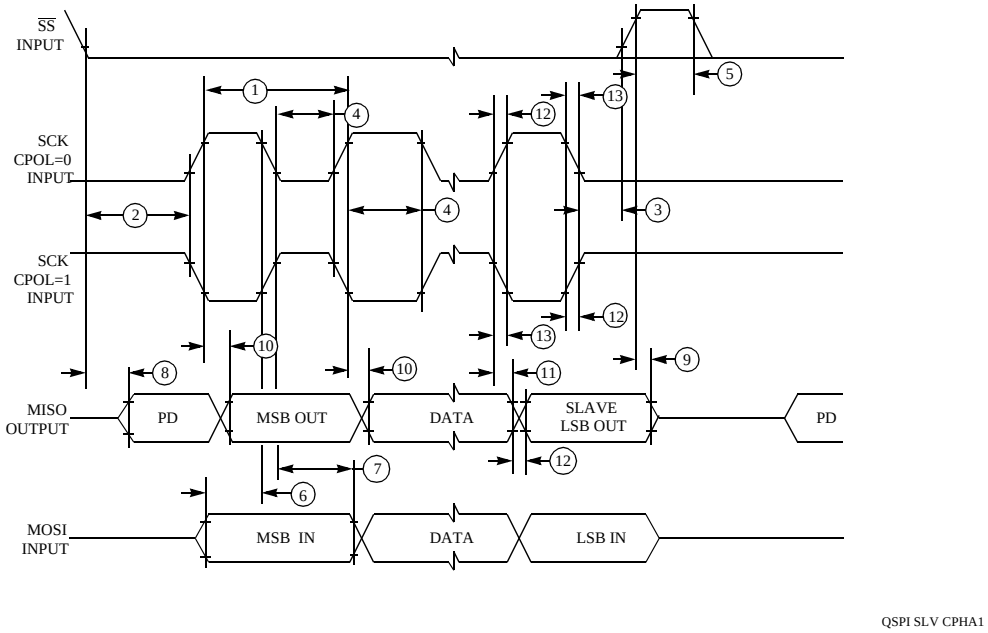

Figure E-19 QSPI Timing — Slave, CPHA = 1