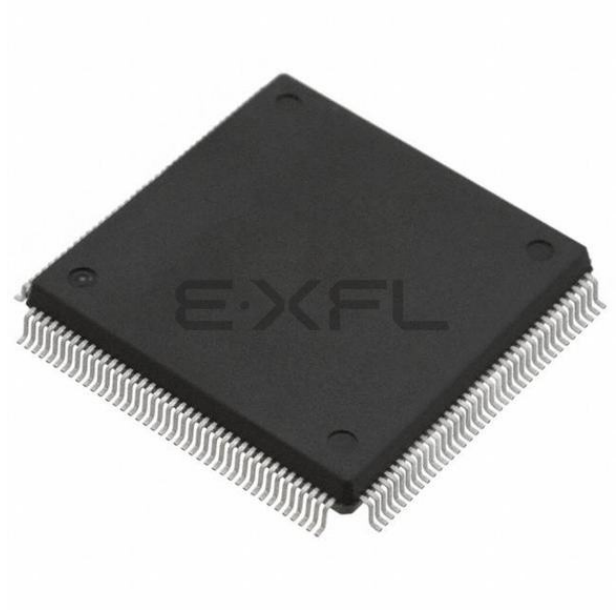


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Not For New Designs |
| Core Processor | 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 ~ 125°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/sc68376bamab20 |

**Table 5-4 Bus Monitor Period**

| BMT[1:0] | Bus Monitor Time-Out Period |
|---|---|
| 00 | 64 system clocks |
| 01 | 32 system clocks |
| 10 | 16 system clocks |
| 11 | 8 system clocks |

The monitor does not check $\overline{\text{DSACK}}$ response on the external bus unless the CPU32 initiates a bus cycle. The BME bit in SYPCR enables the internal bus monitor for internal to external bus cycles. If a system contains external bus masters, an external bus monitor must be implemented and the internal-to-external bus monitor option must be disabled.

When monitoring transfers to an 8-bit port, the bus monitor does not reset until both byte accesses of a word transfer are completed. Monitor time-out period must be at least twice the number of clocks that a single byte access requires.

### 5.4.3 Halt Monitor

The halt monitor responds to an assertion of the $\overline{\text{HALT}}$ signal on the internal bus, caused by a double bus fault. A flag in the reset status register (RSR) indicates that the last reset was caused by the halt monitor. Halt monitor reset can be inhibited by the halt monitor (HME) enable bit in SYPCR. Refer to **5.6.5.2 Double Bus Faults** for more information.

### 5.4.4 Spurious Interrupt Monitor

During interrupt exception processing, the CPU32 normally acknowledges an interrupt request, recognizes the highest priority source, and then either acquires a vector or responds to a request for autovectoring. The spurious interrupt monitor asserts the internal bus error signal ($\overline{\text{BERR}}$) if no interrupt arbitration occurs during interrupt exception processing. The assertion of $\overline{\text{BERR}}$ causes the CPU32 to load the spurious interrupt exception vector into the program counter. The spurious interrupt monitor cannot be disabled. Refer to **5.8 Interrupts** for more information. For detailed information about interrupt exception processing, refer to **4.9 Exception Processing**.

### 5.4.5 Software Watchdog

The software watchdog is controlled by the software watchdog enable (SWE) bit in SYPCR. When enabled, the watchdog requires that a service sequence be written to the software service register (SWSR) on a periodic basis. If servicing does not take place, the watchdog times out and asserts the $\overline{\text{RESET}}$ signal.

Each time the service sequence is written, the software watchdog timer restarts. The sequence to restart consists of the following steps:

1. Write $55 to SWSR.
2. Write $AA to SWSR.

This additional $\overline{BG}$ assertion allows external arbitration circuitry to select the next bus master before the current master has released the bus.

Refer to **Figure 5-15**, which shows bus arbitration for a single device. The flowchart shows $\overline{BR}$ negated at the same time $\overline{BGACK}$ is asserted.

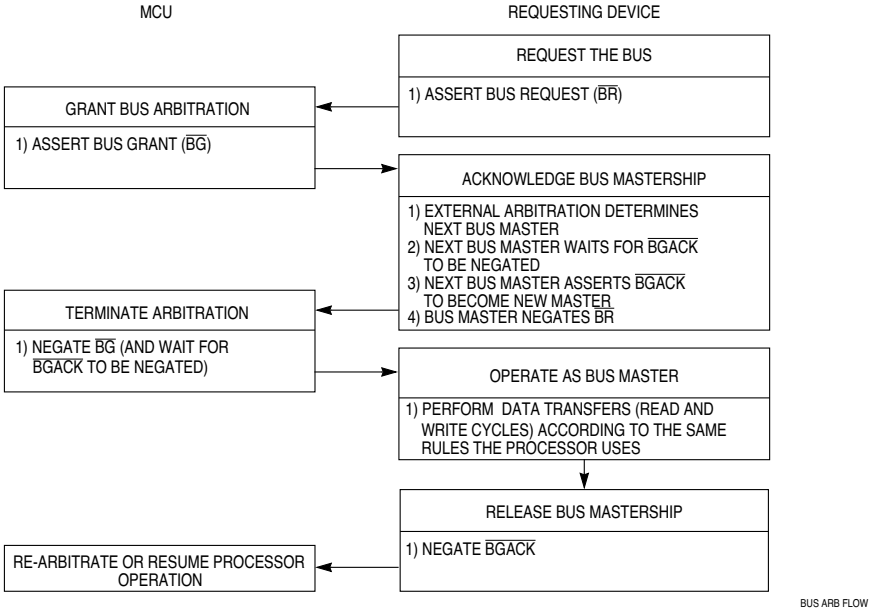



**Figure 5-15 Bus Arbitration Flowchart for Single Request**

### 5.6.6.1 Show Cycles

The MCU normally performs internal data transfers without affecting the external bus, but it is possible to show these transfers during debugging. $\overline{AS}$ is not asserted externally during show cycles.

Show cycles are controlled by SHEN[1:0] in SIMCR. This field is set to %00 by reset. When show cycles are disabled, the address bus, function codes, size, and read/write signals reflect internal bus activity, but $\overline{AS}$ and $\overline{DS}$ are not asserted externally and external data bus pins are in high-impedance state during internal accesses. Refer to **5.2.3 Show Internal Cycles** and the *SIM Reference Manual* (SIMRM/AD) for more information.

When show cycles are enabled, $\overline{DS}$ is asserted externally during internal cycles, and internal data is driven out on the external data bus. Because internal cycles normally continue to run when the external bus is granted, one SHEN encoding halts internal bus activity while there is an external master.

## 5.10 Parallel Input/Output Ports

Sixteen SIM pins can be configured for general-purpose discrete input and output. Although these pins are organized into two ports, port E and port F, function assignment is by individual pin. Pin assignment registers, data direction registers, and data registers are used to implement discrete I/O.

### 5.10.1 Pin Assignment Registers

Bits in the port E and port F pin assignment registers (PEPAR and PFPAR) control the functions of the pins on each port. Any bit set to one defines the corresponding pin as a bus control signal. Any bit cleared to zero defines the corresponding pin as an I/O pin.

### 5.10.2 Data Direction Registers

Bits in the port E and port F data direction registers (DDRE and DDRF) control the direction of the pin drivers when the pins are configured as I/O. Any bit in a register set to one configures the corresponding pin as an output. Any bit in a register cleared to zero configures the corresponding pin as an input. These registers can be read or written at any time.

### 5.10.3 Data Registers

A write to the port E and port F data registers (PORTE[0:1] and PORTF[0:1]) is stored in an internal data latch, and if any pin in the corresponding port is configured as an output, the value stored for that bit is driven out on the pin. A read of a data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the port data register. Both data registers can be accessed in two locations and can be read or written at any time.

## 5.11 Factory Test

The test submodule supports scan-based testing of the various MCU modules. It is integrated into the SIM to support production test. Test submodule registers are intended for Motorola use only. Register names and addresses are provided in **D.2.2 System Integration Test Register** and **D.2.5 System Integration Test Register (ECLK)** to show the user that these addresses are occupied. The QUOT pin is also used for factory test.

## 7.3 MRM Array Address Space Type

ASPC[1:0] in MRMCR determines ROM array address space type. The module can respond to both program and data space accesses or to program space accesses only. This allows code to be executed from ROM, and permits use of program counter relative addressing mode for operand fetches from the array. The default value of ASPC[1:0] is established during mask programming, but field value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one.

**Table 7-1** shows ASPC[1:0] field encodings.

**Table 7-1 ROM Array Space Type**

| ASPC[1:0] | State Specified |
|---|---|
| 00 | Unrestricted program and data |
| 01 | Unrestricted program |
| 10 | Supervisor program and data |
| 11 | Supervisor program |

Refer to **4.5 Addressing Modes** for more information on addressing modes. Refer to **5.5.1.7 Function Codes** for more information concerning address space types and program/data space access.

## 7.4 Normal Access

The array can be accessed by byte, word, or long word. A byte or aligned word access takes a minimum of one bus cycle (two system clocks). A long word access requires two bus cycles. Misaligned accesses are not permitted by the CPU32 and will result in an address error exception.

Access time can be optimized for a particular application by inserting wait states into each access. The number of wait states inserted is determined by the value of WAIT[1:0] in the MRMCR. Two, three, four, or five bus-cycle accesses can be specified. The default value WAIT[1:0] is established during mask programming, but field value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one.

**Table 7-2** shows WAIT[1:0] field encodings.

**Table 7-2 Wait States Field**

| WAIT[1:0] | Cycles per Transfer |
|---|---|
| 00 | 3 |
| 01 | 4 |
| 10 | 5 |
| 11 | 2 |

Refer to **5.6 Bus Operation** for more information concerning access times.

# SECTION 9 QUEUED SERIAL MODULE

This section is an overview of the queued serial module (QSM). Refer to the *QSM Reference Manual* (QSMRM/AD) for complete information about the QSM.

## 9.1 General

The QSM contains two serial interfaces, the queued serial peripheral interface (QSPI) and the serial communication interface (SCI). **Figure 9-1** is a block diagram of the QSM.

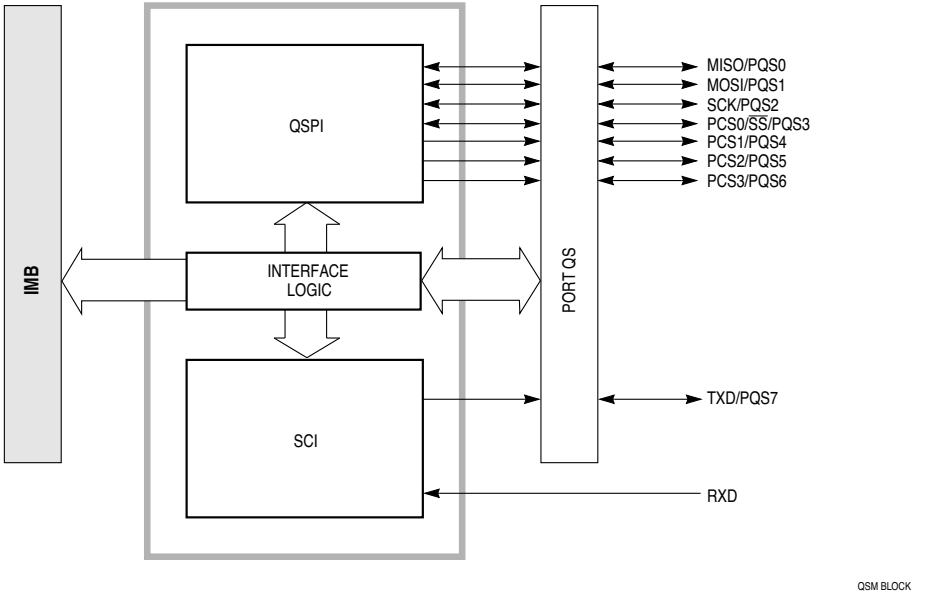



**Figure 9-1 QSM Block Diagram**

The QSPI provides peripheral expansion or interprocessor communication through a full-duplex, synchronous, three-line bus. Four programmable peripheral chip-selects can select up to sixteen peripheral devices by using an external one of sixteen line selector. A self-contained RAM queue allows up to sixteen serial transfers of eight to sixteen bits each or continuous transmission of up to a 256-bit data stream without CPU32 intervention. A special wrap-around mode supports continuous transmission/reception modes.

Baud rate is selected by writing a value from 2 to 255 into SPBR[7:0] in SPCR0. The QSPI uses a modulus counter to derive SCK baud rate from the MCU system clock.

The following expressions apply to SCK baud rate:

$$\text{SCK Baud Rate} = \frac{\text{System Clock}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{\text{System Clock}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value.

The DSCK bit in each command RAM byte inserts either a standard or user-specified delay from chip-select assertion until the leading edge of the serial clock. The DSCKL field in SPCR1 determines the length of the user-defined delay before the assertion of SCK. The following expression determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{\text{System Clock}}$$

where DSCKL[6:0] equals {1,2,3,..., 127}.

When DSCK equals zero, DSCKL[6:0] is not used. Instead, the PCS valid-to-SCK transition is one-half the SCK period.

There are two transfer length options. The user can choose a default value of eight bits, or a programmed value of eight to sixteen bits, inclusive. The programmed value must be written into BITS[3:0] in SPCR0. The BITSE bit in each command RAM byte determines whether the default value (BITSE = 0) or the BITS value (BITSE = 1) is used. **Table 9-3** shows BITS[3:0] encoding.

**Table 9-3 Bits Per Transfer**

| BITS[3:0] | Bits per Transfer |
|---|---|
| 0000 | 16 |
| 0001 | Reserved |
| 0010 | Reserved |
| 0011 | Reserved |
| 0100 | Reserved |
| 0101 | Reserved |
| 0110 | Reserved |
| 0111 | Reserved |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | 10 |
| 1011 | 11 |
| 1100 | 12 |
| 1101 | 13 |
| 1110 | 14 |
| 1111 | 15 |

In the low-power stop mode, QADCMCR, the interrupt register (QADCINT), and the test register (QADCTEST) are not reset and fully accessible. The data direction register (DDRQA) and port data registers (PORTQA and PORTQB) are not reset and are read-only accessible. Control register 0 (QACR0), control register 1 (QACR1), control register 2 (QACR2), and status register (QASR) are reset and are read-only accessible. The CCW table and result table are not reset and not accessible. In addition, the QADC clock (QCLK) and the periodic/interval timer are held in reset during low-power stop mode.

If the STOP bit is clear, low-power stop mode is disabled. Refer to **D.5.1 QADC Module Configuration Register** for more information.

### 8.6.2 Freeze Mode

The QADC enters freeze mode when background debug mode is enabled and a breakpoint is processed. This is indicated by assertion of the FREEZE line on the IMB. The FRZ bit in QADCMCR determines whether or not the QADC responds to an IMB FREEZE assertion. Freeze mode is useful when debugging an application.

When the IMB FREEZE line is asserted and the FRZ bit is set, the QADC finishes any conversion in progress and then freezes. Depending on when the FREEZE is asserted, there are three possible queue freeze scenarios:

- When a queue is not executing, the QADC freezes immediately.
- When a queue is executing, the QADC completes the current conversion and then freezes.
- If during the execution of the current conversion, the queue operating mode for the active queue is changed, or a queue 2 abort occurs, the QADC freezes immediately.

When the QADC enters the freeze mode while a queue is active, the current CCW location of the queue pointer is saved.

In freeze mode, the analog logic is held in reset and is not clocked. Although QCLK is unaffected, the periodic/interval timer is held in reset. External trigger events that occur during freeze mode are not recorded. The CPU32 may continue to access all QADC registers, the CCW table, and the result table. Although the QADC saves a pointer to the next CCW in the current queue, software can force the QADC to execute a different CCW by writing new queue operating modes before normal operation resumes. The QADC looks at the queue operating modes, the current queue pointer, and any pending trigger events to decide which CCW to execute.

If the FRZ bit is clear, assertion of the IMB FREEZE line is ignored. Refer to **D.5.1 QADC Module Configuration Register** for more information.

### 8.6.3 Supervisor/Unrestricted Address Space

The QADC memory map is divided into two segments: supervisor-only data space and assignable data space. Access to supervisor-only data space is permitted only when the CPU32 is operating in supervisor mode. Assignable data space can have either restricted to supervisor-only data space access or unrestricted supervisor and user

**Figure 8-5** illustrates the timing for conversions. This diagram assumes a final sampling period of two QCLKs.

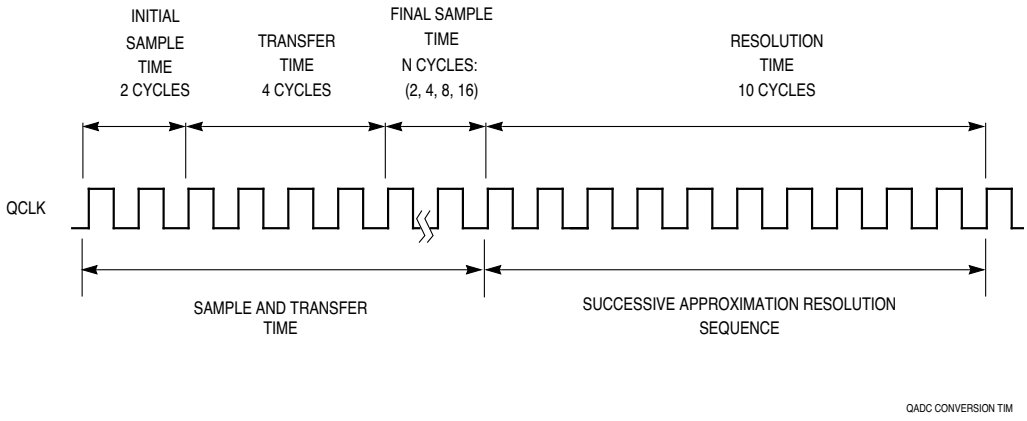



**Figure 8-5 Conversion Timing**

### 8.11.1.1 Amplifier Bypass Mode Conversion Timing

If the amplifier bypass mode is enabled for a conversion by setting the amplifier bypass (BYP) bit in the CCW, the timing changes to that shown in **Figure 8-6**. The initial sample time and the transfer time are eliminated, reducing the potential conversion time by six QCLKs. However, due to internal RC effects, a minimum final sample time of four QCLKs must be allowed. This results in a savings of four QCLKs. When using the bypass mode, the external circuit should be of low source impedance, typically less than 10 kΩ. Also, the loading effects of the external circuitry by the QADC need to be considered, since the benefits of the sample amplifier are not present.

# SECTION 11 TIME PROCESSOR UNIT

The time processor unit (TPU) is an intelligent, semi-autonomous microcontroller designed for timing control. Operating simultaneously with the CPU32, the TPU schedules tasks, processes microcode ROM instructions, accesses shared data with the CPU32, and performs input and output functions. **Figure 11-1** is a simplified block diagram of the TPU.

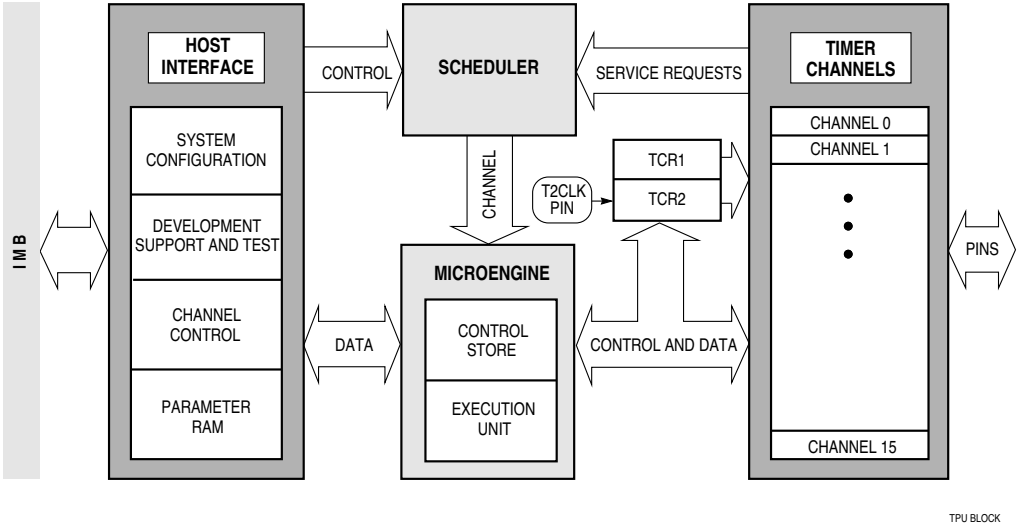



**Figure 11-1 TPU Block Diagram**

## 11.1 General

The TPU 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. TPU functions replace software functions that would require CPU32 interrupt service. Two sets of microcode ROM functions are currently available for most MCU derivatives with the TPU.

The A mask set (or original mask set) includes the following functions:

- Discrete input/output
- Input capture/input transition counter
- Output compare
- Pulse width modulation

bit is set and the CIRL field has a non-zero value. To clear a status flag, read CISR, then write a zero to the appropriate bit. CISR is the only TPU register that can be accessed on a byte basis.

### 11.6.2.2 Channel Function Select Registers

Encoded 4-bit fields within the channel function select registers specify one of 16 time functions to be executed on the corresponding channel. Encodings for predefined functions in the TPU ROM are found in **Table 11-3**.

**Table 11-3 TPU Function Encodings**

| A Mask Set | | G Mask Set | |
|---|---|---|---|
| **Function Name** | **Function Code** | **Function Name** | **Function Code** |
| PPWA<br>Period/pulse width accumulator | $F | PTA<br>Programmable time accumulator | $F |
| OC<br>Output compare | $E | QOM<br>Queued output match | $E |
| SM<br>Stepper motor | $D | TSM<br>Table stepper motor | $D |
| PSP<br>Position-synchronized pulse generator | $C | FQM<br>Frequency measurement | $C |
| PMA/PMM<br>Period measurement with additional/missing transition detect | $B | UART<br>Universal asynchronous receiver/transmitter | $B |
| ITC<br>Input capture/input transition counter | $A | NITC<br>New input transition counter | $A |
| PWM<br>Pulse width modulation | $9 | COMM<br>Multiphase motor commutation | $9 |
| DIO<br>Discrete input/output | $8 | HALLD<br>Hall effect decode | $8 |
| SPWM<br>Synchronized pulse width modulation | $7 | — | — |
| QDEC<br>Quadrature decode | $6 | — | — |

### 11.6.2.3 Host Sequence Registers

The host sequence field selects the mode of operation for the time function selected on a given channel. The meaning of the host sequence bits depends on the time function specified. Refer to **Table 11-4**, which is a summary of the host sequence and host service request bits for each time function. Refer to the *TPU Reference Manual* (TPURM/AD) and the Motorola TPU Literature Package (TPULITPAK/D) for more information.

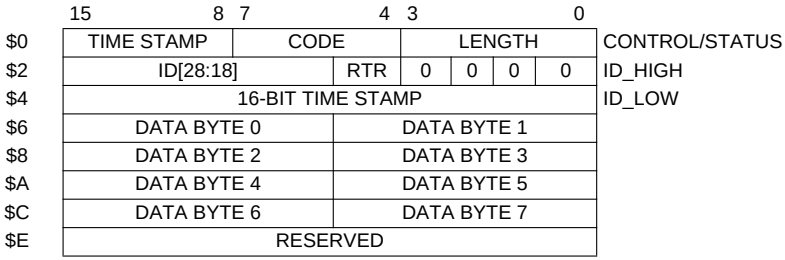



**Figure 13-4 Standard ID Message Buffer Structure**

### 13.4.1.1 Common Fields for Extended and Standard Format Frames

**Table 13-1** describes the message buffer fields that are common to both extended and standard identifier format frames.

**Table 13-1 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 **Tables 13-2** and **13-3** |
| RX Length | Length (in bytes) of the RX data stored in offset $6 through $D 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 $D of the buffer. This field is written by the CPU32, 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 CPU32 provides the data to be transmitted within the frame. |
| Reserved | This word entry field (16 bits) should not be accessed by the CPU32. |

**Table 13-2 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 CPU32 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 13-3 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.

### 13.4.1.2 Fields for Extended Format Frames

**Table 13-4** describes the message buffer fields used only for extended identifier format frames.

**Table 13-4 Extended Format Frames**

| Field | Description |
|---|---|
| ID[28:18]/[17:15] | Contains the 14 most significant bits of the extended identifier, located in the ID HIGH word of the message buffer. |
| Substitute Remote Request (SRR) | Contains a fixed recessive bit, used only in extended format. Should be set to one by the user for TX buffers. It will be stored as received on the CAN bus for RX buffers. |
| ID Extended (IDE) | If extended format frame is used, this field should be set to one. If zero, standard format frame should be used. |
| ID[14:0] | Bits [14:0] of the extended identifier, located in the ID LOW word of the message buffer. |
| Remote Transmission Request (RTR) | This bit is located in the least significant bit of the ID LOW word of the message buffer;<br>0 = Data Frame, 1 = Remote Frame. |

### 13.4.1.3 Fields for Standard Format Frames

**Table 13-5** describes the message buffer fields used only for standard identifier format frames.

## Table A-12 QADC DC Electrical Characteristics (Operating)

($V_{SSI}$ and $V_{SSA}$ = 0Vdc, $f_{QCLK}$ = 2.1 MHz, $T_A = T_L$ to $T_H$)

| Num | Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| 1 | Analog Supply[1] | $V_{DDA}$ | 4.5 | 5.5 | V |
| 2 | Internal Digital Supply[1] | $V_{DDI}$ | 4.5 | 5.5 | V |
| 3 | $V_{SS}$ Differential Voltage | $V_{SSI} - V_{SSA}$ | – 1.0 | 1.0 | mV |
| 4 | $V_{DD}$ Differential Voltage | $V_{DDI} - V_{DDA}$ | – 1.0 | 1.0 | V |
| 5 | Reference Voltage Low[2] | $V_{RL}$ | $V_{SSA}$ | — | V |
| 6 | Reference Voltage High[2] | $V_{RH}$ | — | $V_{DDA}$ | V |
| 7 | $V_{REF}$ Differential Voltage[3] | $V_{RH} - V_{RL}$ | 4.5 | 5.5 | V |
| 8 | Mid-Analog Supply Voltage | $V_{DDA}/2$ | 2.25 | 2.75 | V |
| 9 | Input Voltage | $V_{INDC}$ | $V_{SSA}$ | $V_{DDA}$ | V |
| 10 | Input High Voltage, PQA and PQB | $V_{IH}$ | 0.7 ($V_{DDA}$) | $V_{DDA}$ + 0.3 | V |
| 11 | Input Low Voltage, PQA and PQB | $V_{IL}$ | $V_{SSA}$ – 0.3 | 0.2 ($V_{DDA}$) | V |
| 12 | Input Hysteresis[4] | $V_{HYS}$ | 0.5 | — | V |
| 13 | Output Low Voltage, PQA[5]<br>$I_{OL}$ = 5.3 mA<br>$I_{OL}$ = 10.0 µA | $V_{OL}$ | —<br>— | 0.4<br>0.2 | V |
| 14 | Analog Supply Current<br>Normal Operation[6]<br>Low-Power Stop | $I_{DDA}$ | —<br>— | 1.0<br>10.0 | mA<br>µA |
| 15 | Reference Supply Current | $I_{REF}$ | — | 150 | µA |
| 16 | Load Capacitance, PQA | $C_L$ | — | 90 | pF |
| 17 | Input Current, Channel Off[7]<br>PQA<br>PQB | $I_{OFF}$ | —<br>— | 250<br>150 | nA |
| 18 | Total Input Capacitance[8]<br>PQA Not Sampling<br>PQA Sampling<br>PQB Not Sampling<br>PQB Sampling | $C_{IN}$ | —<br>—<br>—<br>— | 15<br>20<br>10<br>15 | pF |

NOTES:

1. Refers to operation over full temperature and frequency range.
2. To obtain full-scale, full-range results, $V_{SSA} \le V_{RL} \le V_{INDC} \le V_{RH} \le V_{DDA}$.
3. Accuracy tested and guaranteed at $V_{RH} - V_{RL}$ = 5.0V ± 10%.
4. Parameter applies to the following pins:
   Port A: PQA[7:0]/AN[59:58]/ETRIG[2:1]
   Port B: PQB[7:0]/AN[3:0]/AN[51:48]/AN[Z:W]
5. Open drain only.
6. Current measured at maximum system clock frequency with QADC active.
7. Maximum leakage occurs at maximum operating temperature. Current decreases by approximately one-half for each 10° C decrease from maximum temperature.
8. This parameter is periodically sampled rather than 100% tested.

## Table A-17 SASM Timing Characteristics

($V_{DD}$ = 5.0 Vdc ± 5%, $V_{SS}$ = 0Vdc, $T_A = T_L$ to $T_H$)

| Num | Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| 1 | Input pin low time | $t_{PINL}$ | $2.0/f_{sys}$ | — | µs |
| 2 | Input pin high time | $t_{PINH}$ | $2.0/f_{sys}$ | — | µs |
| 3 | Input capture resolution[1] | $t_{RESCA}$ | — | $2.0/f_{sys}$ | µs |
| 4 | Pin to input capture delay | $t_{PCAPT}$ | $2.5/f_{sys}$ | $4.5/f_{sys}$ | µs |
| 5 | Pin to FLAG set | $t_{PFLAG}$ | $2.5/f_{sys}$ | $4.5/f_{sys}$ | µs |
| 6 | Pin to IN bit delay | $t_{PINB}$ | $1.5/f_{sys}$ | $2.5/f_{sys}$ | µs |
| 7 | OCT output pulse | $t_{OCT}$ | $2.0/f_{sys}$ | — | µs |
| 8 | Compare resolution | $t_{RESCM}$ | | $2.0/f_{sys}$ | µs |
| 9 | TBB change to FLAG set | $t_{CFLAG}$ | $1.5/f_{sys}$ | $1.5/f_{sys}$ | µs |
| 10 | TBB change to pin change[2] | $t_{CPIN}$ | $1.5/f_{sys}$ | $1.5/f_{sys}$ | µs |
| 11 | FLAG to IMB interrupt request | $t_{FIRQ}$ | $1.0/f_{sys}$ | $1.0/f_{sys}$ | µs |

NOTES:
1. Minimum resolution depends on counter and prescaler divide ratio selection.
2. Time given from when new value is stable on time base bus.

# APPENDIX D REGISTER SUMMARY

This appendix contains address maps, register diagrams, and bit/field definitions for MC68336 and MC68376 microcontrollers. More detailed information about register function is provided in the appropriate sections of the manual.

Except for central processing unit resources, information is presented in the intermodule bus address order shown in **Table D-1**.

**Table D-1 Module Address Map**

| Module | Size (Bytes) | Base Address |
|---|---|---|
| SIM | 128 | $YFFA00 |
| SRAM | 8 | $YFFB40 |
| MRM (MC68376 Only) | 32 | $YFF820 |
| QADC | 512 | $YFF200 |
| QSM | 512 | $YFFC00 |
| CTM4 | 256 | $YFF400 |
| TPU | 512 | $YFFE00 |
| TPURAM | 64 | $YFFB00 |
| TouCAN (MC68376 Only) | 384 | $YFF080 |

Control registers for all the modules in the microcontroller are mapped into a 4-Kbyte block. The state of the module mapping (MM) bit in the SIM configuration register (SIMCR) determines where the control register block is located in the system memory map. When MM = 0, register addresses range from $7FF000 to $7FFFFF; when MM = 1, register addresses range from $FFF000 to $FFFFFF.

In the module memory maps in this appendix, the “Access” column specifies which registers are accessible when the CPU32 is in supervisor mode only and which registers can be assigned to either supervisor or user mode.

## D.1 Central Processor Unit

CPU32 registers are not part of the module address map. **Figures D-1** and **D-2** show a functional representation of CPU32 resources.

## D.4 Masked ROM Module

The MRM is used only in the MC68376. **Table D-21** shows the MRM address map. MRM control registers are accessible in supervisor mode only.

The reset states shown for the MRM registers are for the generic (blank ROM) versions of the device. Several MRM register bit fields can be user-specified on a custom-masked ROM device. Contact a Motorola sales representative for information on ordering a custom ROM device.

**Table D-21 MRM Address Map**

| Address | 15 0 |
|---|---|
| $YFF820 | Masked ROM Module Configuration Register (MRMCR) |
| $YFF822 | Not Implemented |
| $YFF824 | ROM Array Base Address High Register (ROMBAH) |
| $YFF826 | ROM Array Base Address Low Register (ROMBAL) |
| $YFF828 | Signature High Register (SIGHI) |
| $YFF82A | Signature Low Register (SIGLO) |
| $YFF82C | Not Implemented |
| $YFF82E | Not Implemented |
| $YFF830 | ROM Bootstrap Word 0 (ROMBS0) |
| $YFF832 | ROM Bootstrap Word 1 (ROMBS1) |
| $YFF834 | ROM Bootstrap Word 2 (ROMBS2) |
| $YFF836 | ROM Bootstrap Word 3 (ROMBS3) |
| $YFF838 | Not Implemented |
| $YFF83A | Not Implemented |
| $YFF83C | Not Implemented |
| $YFF83E | Not Implemented |

### D.4.1 Masked ROM Module Configuration Register

**MRMCR —** Masked ROM Module Configuration Register **$YFF820**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STOP | 0 | 0 | $\overline{\text{BOOT}}$ | LOCK | EMUL | ASPC[1:0] | | WAIT[1:0] | | 0 | 0 | 0 | 0 | 0 | 0 |
| RESET: | | | | | | | | | | | | | | | |
| $\overline{\text{DATA14}}$ | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

STOP — Low-Power Stop Mode Enable

The reset state of the STOP bit is the complement of DATA14 state during reset. The ROM array base address cannot be changed unless the STOP bit is set.

0 = ROM array operates normally.

1 = ROM array operates in low-power stop mode.

**NOTE**

Unless DATA14 is pulled down during reset, the MRM will be enabled. On generic MC68376 devices (blank ROM), the MRM is enabled at address $FF0000 (which is outside of the 1 Mbyte address range of $\overline{\text{CSBOOT}}$. On these devices, the MRM should be disabled (since it is blank) by setting the STOP bit during system initialization.

### D.4.2 ROM Array Base Address Register High

**ROMBAH —** ROM Array Base Address Register High **$YFF824**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADDR 23 | ADDR 22 | ADDR 21 | ADDR 20 | ADDR 19 | ADDR 18 | ADDR 17 | ADDR 16 |
| RESET: | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### D.4.3 ROM Array Base Address Register Low

**ROMBAL —** ROM Array Base Address Register Low **$YFF826**

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

ROMBAH and ROMBAL specify ROM array base address. The reset state of these registers is specified at mask time. They can only be written when STOP = 1 and LOCK = 0. This prevents accidental remapping of the array. Because the 8-Kbyte ROM array in the MC68376 must be mapped to an 8-Kbyte boundary, ROMBAL bits [12:0] always contains $0000. ROMBAH ADDR[15:8] read zero.

### D.4.4 ROM Signature High Register

**RSIGHI —** ROM Signature High Register **$YFF828**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NOT USED | | | | | | | | | | | | | RSP18 | RSP17 | RSP16 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### D.4.5 ROM Signature Low Register

**RSIGLO —** ROM Signature Low Register **$YFF82A**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSP15 | RSP14 | RSP13 | RSP12 | RSP11 | RSP10 | RSP9 | RSP8 | RSP7 | RSP6 | RSP5 | RSP4 | RSP3 | RSP2 | RSP1 | RSP0 |
| RESET: | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

RSIGHI and RSIGLO specify a ROM signature pattern. A user-written signature identification algorithm allows identification of the ROM array content. The signature is specified at mask time and cannot be changed.

MODE[3:0] — DASM Mode Select

This bit field selects the mode of operation of the DASM. Refer to **Table D-46**.

**NOTE**

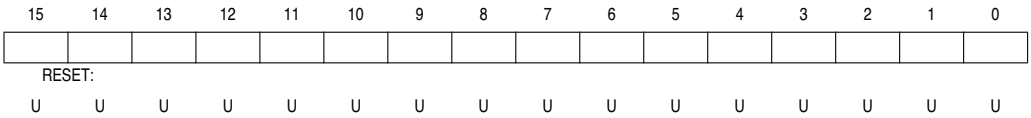
To avoid spurious interrupts, DASM interrupts should be disabled before changing the operating mode.

**Table D-46 DASM Mode Select Field**

| MODE[3:0] | Bits of Resolution | Time Base Bits Ignored | DASM Operating Mode |
|---|---|---|---|
| 0000 | — | — | DIS – Disabled |
| 0001 | 16 | — | IPWM – Input pulse width measurement |
| 0010 | 16 | — | IPM – Input measurement period |
| 0011 | 16 | — | IC – Input capture |
| 0100 | 16 | — | OCB – Output compare, flag on B compare |
| 0101 | 16 | — | OCAB – Output compare, flag on A and B compare |
| 011X | — | — | Not used |
| 1000 | 16 | — | OPWM – Output pulse width modulation |
| 1001 | 15 | 15 | OPWM – Output pulse width modulation |
| 1010 | 14 | [15:14] | OPWM – Output pulse width modulation |
| 1011 | 13 | [15:13] | OPWM – Output pulse width modulation |
| 1100 | 12 | [15:12] | OPWM – Output pulse width modulation |
| 1101 | 11 | [15:11] | OPWM – Output pulse width modulation |
| 1110 | 9 | [15:9] | OPWM – Output pulse width modulation |
| 1111 | 7 | [15:7] | OPWM – Output pulse width modulation |

### D.7.12 DASM Data Register A

**DASM3A —** DASM3 Data Register A **$YFF41A**
**DASM4A —** DASM4 Data Register A **$YFF422**
**DASM9A —** DASM9 Data Register A **$YFF44A**
**DASM10A —** DASM10 Data Register A **$YFF452**




DASMA is the data register associated with channel A. **Table D-47** shows how DASMA is used with the different modes of operation.

CH[15:0] — Encoded Channel Priority Levels

**Table D-56** shows channel priority levels.

**Table D-56 Channel Priorities**

| CHx[1:0] | Service | Guaranteed Time Slots |
|---|---|---|
| 00 | Disabled | — |
| 01 | Low | 1 out of 7 |
| 10 | Middle | 2 out of 7 |
| 11 | High | 4 out of 7 |

### D.8.11 Channel Interrupt Status Register

**CISR —** Channel Interrupt Status Register **$YFFE20**

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

CH[15:0] — Channel Interrupt Status

0 = Channel interrupt not asserted.

1 = Channel interrupt asserted.

### D.8.12 Link Register

**LR —** Link Register **$YFFE22**

Used for factory test only.

### D.8.13 Service Grant Latch Register

**SGLR** — Service Grant Latch Register **$YFFE24**

Used for factory test only.

### D.8.14 Decoded Channel Number Register

**DCNR** — Decoded Channel Number Register **$YFFE26**

Used for factory test only.

### D.8.15 TPU Parameter RAM

The channel parameter registers are organized as one hundred 16-bit words of RAM. Channels 0 to 13 have six parameters. Channels 14 and 15 each have eight parameters. The parameter registers constitute a shared work space for communication between the CPU32 and the TPU. Refer to **Table D-57**.