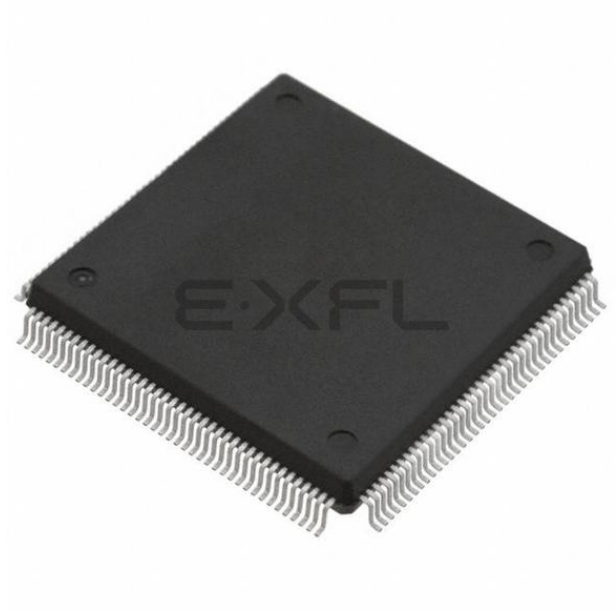


Welcome to E-XFL.COM

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 ~ 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/mc68376bgmft20

Because the SIM routes external interrupt requests to the CPU32, the SIM IARB field value is used for arbitration between internal and external interrupts of the same priority. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000, which prevents SIM interrupts from being discarded during initialization. Refer to **5.8 Interrupts** for a discussion of interrupt arbitration.

5.2.3 Show Internal Cycles

A show cycle allows internal bus transfers to be monitored externally. The SHEN field in SIMCR determines what the external bus interface does during internal transfer operations. **Table 5-1** shows whether data is driven externally, and whether external bus arbitration can occur. Refer to **5.6.6.1 Show Cycles** for more information.

Table 5-1 Show Cycle Enable Bits

SHEN[1:0]	Action
00	Show cycles disabled, external arbitration enabled
01	Show cycles enabled, external arbitration disabled
10	Show cycles enabled, external arbitration enabled
11	Show cycles enabled, external arbitration enabled; internal activity is halted by a bus grant

5.2.4 Register Access

The CPU32 can operate at one of two privilege levels. Supervisor level is more privileged than user level — all instructions and system resources are available at supervisor level, but access is restricted at user level. Effective use of privilege level can protect system resources from uncontrolled access. The state of the S bit in the CPU status register determines access level, and whether the user or supervisor stack pointer is used for stacking operations. The SUPV bit places SIM global registers in either supervisor or user data space. When SUPV = 0, registers with controlled access are accessible from either the user or supervisor privilege level; when SUPV = 1, registers with controlled access are restricted to supervisor access only.

5.2.5 Freeze Operation

The FREEZE signal halts MCU operations during debugging. FREEZE is asserted internally by the CPU32 if a breakpoint occurs while background mode is enabled. When FREEZE is asserted, only the bus monitor, software watchdog, and periodic interrupt timer are affected. The halt monitor and spurious interrupt monitor continue to operate normally. Setting the freeze bus monitor (FRZBM) bit in SIMCR disables the bus monitor when FREEZE is asserted. Setting the freeze software watchdog (FRZSW) bit disables the software watchdog and the periodic interrupt timer when FREEZE is asserted.

When the clock synthesizer is used, SYNCR determines the system clock frequency and certain operating parameters. The W and Y[5:0] bits are located in the PLL feedback path, enabling frequency multiplication by a factor of up to 256. When the W or Y values change, VCO frequency changes, and there is a VCO relock delay. The SYNCR X bit controls a divide-by circuit that is not in the synthesizer feedback loop. When X = 0 (reset state), a divide-by-four circuit is enabled, and the system clock frequency is one-fourth the VCO frequency (f_{VCO}). When X = 1, a divide-by-two circuit is enabled and system clock frequency is one-half the VCO frequency (f_{VCO}). There is no relock delay when clock speed is changed by the X bit.

Clock frequency is determined by SYNCR bit settings as follows:

$$f_{sys} = \frac{f_{ref}}{128}[4(Y + 1)(2^{(2W + X)})]$$

The reset state of SYNCR ($3F00) results in a power-on f_{sys} of 8.388 MHz when f_{ref} is 4.194 MHz.

For the device to operate correctly, the clock frequency selected by the W, X, and Y bits must be within the limits specified for the MCU.

Internal VCO frequency is determined by the following equations:

$$f_{VCO} = 4f_{sys} \text{ if } X = 0$$

or

$$f_{VCO} = 2f_{sys} \text{ if } X = 1$$

Table 5-2 shows clock control multipliers for all possible combinations of SYNCR bits. To obtain clock frequency, find counter modulus in the leftmost column, then multiply the reference frequency by the value in the appropriate prescaler cell. Shaded cells exceed the maximum system clock frequency at the time of manual publication; however, they may be usable in the future. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for maximum allowable clock rate.

Table 5-3 shows clock frequencies available with a 4.194 MHz reference and a maximum specified clock frequency of 20.97 MHz. To obtain clock frequency, find counter modulus in the leftmost column, then refer to appropriate prescaler cell. Shaded cells exceed the maximum system clock frequency at the time of manual publication; however, they may be usable in the future. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for maximum system frequency (f_{sys}).

Table 5-3 System Frequencies from 4.194 MHz Reference

Modulus	Prescaler			
Y	[W:X] = 00	[W:X] = 01	[W:X] = 10	[W:X] = 11
000000	131 kHz	262 kHz	524 kHz	1049 kHz
000001	262	524	1049	2097
000010	393	786	1573	3146
000011	524	1049	2097	4194
000100	655	1311	2621	5243
000101	786	1573	3146	6291
000110	918	1835	3670	7340
000111	1049	2097	4194	8389
001000	1180	2359	4719	9437
001001	1311	2621	5243	10486
001010	1442	2884	5767	11534
001011	1573	3146	6291	12583
001100	1704	3408	6816	13631
001101	1835	3670	7340	14680
001110	1966	3932	7864	15729
001111	2097	4194	8389	16777
010000	2228	4456	8913	17826
010001	2359	4719	9437	18874
010010	2490	4981	9961	19923
010011	2621	5243	10486	20972
010100	2753	5505	11010	22020
010101	2884	5767	11534	23069
010110	3015	6029	12059	24117
010111	3146	6291	12583	25166
011000	3277	6554	13107	26214
011001	3408	6816	13631	27263
011010	3539	7078	14156	28312
011011	3670	7340	14680	29360
011100	3801	7602	15204	30409
011101	3932	7864	15729	31457
011110	4063	8126	16253	32506
011111	4194	8389	16777	33554

WARNING

If $\overline{\text{DSACK}}$ or $\overline{\text{BERR}}$ remain asserted into S2 of the next bus cycle, that cycle may be terminated prematurely.

5.6.5.1 Bus Errors

The CPU32 treats bus errors as a type of exception. Bus error exception processing begins when the CPU32 detects assertion of the IMB $\overline{\text{BERR}}$ signal (by the internal bus monitor or an external source) while the $\overline{\text{HALT}}$ signal remains negated.

$\overline{\text{BERR}}$ assertions do not force immediate exception processing. The signal is synchronized with normal bus cycles and is latched into the CPU32 at the end of the bus cycle in which it was asserted. Because bus cycles can overlap instruction boundaries, bus error exception processing may not occur at the end of the instruction in which the bus cycle begins. Timing of $\overline{\text{BERR}}$ detection/acknowledge is dependent upon several factors:

- Which bus cycle of an instruction is terminated by assertion of $\overline{\text{BERR}}$.
- The number of bus cycles in the instruction during which $\overline{\text{BERR}}$ is asserted.
- The number of bus cycles in the instruction following the instruction in which $\overline{\text{BERR}}$ is asserted.
- Whether $\overline{\text{BERR}}$ is asserted during a program space access or a data space access.

Because of these factors, it is impossible to predict precisely how long after occurrence of a bus error the bus error exception is processed.

CAUTION

The external bus interface does not latch data when an external bus cycle is terminated by a bus error. When this occurs during an instruction prefetch, the IMB precharge state (bus pulled high, or $FF) is latched into the CPU32 instruction register, with indeterminate results.

5.6.5.2 Double Bus Faults

Exception processing for bus error exceptions follows the standard exception processing sequence. Refer to **4.9 Exception Processing** for more information. However, a special case of bus error, called double bus fault, can abort exception processing.

$\overline{\text{BERR}}$ assertion is not detected until an instruction is complete. The $\overline{\text{BERR}}$ latch is cleared by the first instruction of the $\overline{\text{BERR}}$ exception handler. Double bus fault occurs in three ways:

1. When bus error exception processing begins and a second $\overline{\text{BERR}}$ is detected before the first instruction of the exception handler is executed.
2. When one or more bus errors occur before the first instruction after a reset exception is executed.
3. A bus error occurs while the CPU32 is loading information from a bus error stack frame during a return from exception (RTE) instruction.

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion. Writing a value to DTL[7:0] in SPCR1 specifies a delay period. The DT bit in each command RAM byte determines whether the standard delay period (DT = 0) or the specified delay period (DT = 1) is used. The following expression is used to calculate the delay:

$$\text{Delay after Transfer} = \frac{32 \times \text{DTL[7:0]}}{\text{System Clock}}$$

where DTL equals {1, 2, 3,..., 255}.

A zero value for DTL[7:0] causes a delay-after-transfer value of 8192/System Clock.

$$\text{Standard Delay after Transfer} = \frac{17}{\text{System Clock}}$$

Adequate delay between transfers must be specified for long data streams because the QSPI requires time to load a transmit RAM entry for transfer. Receiving devices need at least the standard delay between successive transfers. If the system clock is operating at a slower rate, the delay between transfers must be increased proportionately.

Operation is initiated by setting the SPE bit in SPCR1. Shortly after SPE is set, the QSPI executes the command at the command RAM address pointed to by NEWQP. Data at the pointer address in transmit RAM is loaded into the data serializer and transmitted. Data that is simultaneously received is stored at the pointer address in receive RAM.

When the proper number of bits have been transferred, the QSPI stores the working queue pointer value in CPTQP, increments the working queue pointer, and loads the next data for transfer from transmit RAM. The command pointed to by the incremented working queue pointer is executed next, unless a new value has been written to NEWQP. If a new queue pointer value is written while a transfer is in progress, that transfer is completed normally.

When the CONT bit in a command RAM byte is set, PCS pins are continuously driven in specified states during and between transfers. If the chip-select pattern changes during or between transfers, the original pattern is driven until execution of the following transfer begins. When CONT is cleared, the data in register PORTQS is driven between transfers. The data in PORTQS must match the inactive states of SCK and any peripheral chip-selects used.

When the QSPI reaches the end of the queue, it sets the SPIF flag. If the SPIFIE bit in SPCR2 is set, an interrupt request is generated when SPIF is asserted. At this point, the QSPI clears SPE and stops unless wrap-around mode is enabled.

- The pause bit is set in CCW0A and EOQ is programmed into CCW0A.
- During queue 1 operation, the pause bit is set in CCW20, which is also BQ2.

8.12.3 Scan Modes

The QADC queuing mechanism provides several methods for automatically scanning input channels. In single-scan mode, a single pass through a sequence of conversions defined by a queue is performed. In continuous-scan mode, multiple passes through a sequence of conversions defined by a queue are executed. The following paragraphs describe the disabled/reserved, single-scan, and continuous-scan operations.

8.12.3.1 Disabled Mode and Reserved Mode

When the disabled mode or a reserved mode is selected, the queue is not active.

NOTE

Do not use a reserved mode. Unspecified operations may result.

Trigger events cannot initiate queue execution. When both queue 1 and queue 2 are disabled, no wait states will be inserted by the QADC for accesses to the CCW and result word tables. When both queues are disabled, it is safe to change the QADC clock prescaler values.

8.12.3.2 Single-Scan Modes

When application software requires execution of a single pass through a sequence of conversions defined by a queue, a single-scan queue operating mode is selected.

In all single-scan queue operating modes, software must enable a queue for execution by writing the single-scan enable bit to one in the queue's control register. The single-scan enable bits, SSE1 and SSE2, are provided for queue 1 and queue 2, respectively.

Until the single-scan enable bit is set, any trigger events for that queue are ignored. The single-scan enable bit may be set to one during the write cycle that selects the single-scan queue operating mode. The single-scan enable bit can be written as a one or a zero but is always read as a zero.

After the single-scan enable bit is set, a trigger event causes the QADC to begin execution with the first CCW in the queue. The single-scan enable bit remains set until the queue scan is complete; the QADC then clears the single-scan enable bit to zero. If the single-scan enable bit is written to one or zero before the queue scan is complete, the queue is not affected. However, if software changes the queue operating mode, the new queue operating mode and the value of the single-scan enable bit are recognized immediately. The current conversion is aborted and the new queue operating mode takes effect.

By properly programming the MQ1 field in QACR1 or the MQ2 field in QACR2, the following modes can be selected for queue 1 and/or 2:

When any of the end-of-queue conditions is recognized, a queue completion flag is set, and if enabled, an interrupt request is generated. The following situations prematurely terminate queue execution:

- Since queue 1 is higher in priority than queue 2, when a trigger event occurs on queue 1 during queue 2 execution, the execution of queue 2 is suspended by aborting execution of the CCW in progress, and queue 1 execution begins. When queue 1 execution is complete, queue 2 conversions restart with the first CCW entry in queue 2 or the first CCW of the queue 2 subqueue being executed when queue 2 was suspended. Alternately, conversions can restart with the aborted queue 2 CCW entry. The resume RES bit in QACR2 allows software to select where queue 2 begins after suspension. By choosing to re-execute all of the suspended queue 2 and subqueue CCWs, all of the samples are guaranteed to have been taken during the same scan pass. However, a high trigger event rate for queue 1 can prohibit the completion of queue 2. If this occurs, execution of queue 2 may begin with the aborted CCW entry.
- When a queue is disabled, any conversion taking place for that queue is aborted. Putting a queue into disabled mode does not power down the converter.
- When the operating mode of a queue is changed to another valid mode, any conversion taking place for that queue is aborted. The queue operating restarts at the beginning of the queue, once an appropriate trigger event occurs.
- When placed in low-power stop mode, the QADC aborts any conversion in progress.
- When the FRZ bit in the QADCMCR is set and the IMB FREEZE line is asserted, the QADC freezes at the end of the current conversion. When FREEZE is negated, the QADC resumes queue execution beginning with the next CCW entry.

8.12.8 Result Word Table

The result word table is a 40-word long, 10-bit wide RAM. The QADC writes a result word after completing an analog conversion specified by the corresponding CCW. The result word table can be read or written, but in normal operation, software reads the result word table to obtain analog conversions from the QADC. Unimplemented bits are read as zeros, and write operations have no effect. Refer to **D.5.9 Result Word Table** for register descriptions.

While there is only one result word table, the data can be accessed in three different alignment formats:

1. Right justified, with zeros in the higher order unused bits.
2. Left justified, with the most significant bit inverted to form a sign bit, and zeros in the unused lower order bits.
3. Left justified, with zeros in the unused lower order bits.

The left justified, signed format corresponds to a half-scale, offset binary, two's complement data format. The data is routed onto the IMB according to the selected format. The address used to access the table determines the data alignment format. All write operations to the result word table are right justified.

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-

11.2.3 Scheduler

When a service request is received, the scheduler determines which TPU channel is serviced by the microengine. A channel can request service for one of four reasons: for host service, for a link to another channel, for a match event, or for a capture event. The host system assigns each active channel one of three priorities: high, middle, or low. When multiple service requests are received simultaneously, a priority-scheduling mechanism grants service based on channel number and assigned priority.

11.2.4 Microengine

The microengine is composed of a control store and an execution unit. Control-store ROM holds the microcode for each factory-masked time function. When assigned to a channel by the scheduler, the execution unit executes microcode for a function assigned to that channel by the CPU32. Microcode can also be executed from the TPURAM module instead of the control store. The TPURAM allows emulation and development of custom TPU microcode without the generation of a microcode ROM mask. Refer to **11.3.6 Emulation Support** for more information.

11.2.5 Host Interface

The host interface registers allow communication between the CPU32 and the TPU, both before and during execution of a time function. The registers are accessible from the IMB through the TPU bus interface unit. Refer to **11.6 Host Interface Registers** and **D.8 Time Processor Unit (TPU)** for register bit/field definitions and address mapping.

11.2.6 Parameter RAM

Parameter RAM occupies 256 bytes at the top of the system address map. Channel parameters are organized as 128 16-bit words. Although all parameter word locations in RAM can be accessed by all channels, only 100 are normally used: channels 0 to 13 use six parameter words, while channels 14 and 15 each use eight parameter words. The parameter RAM address map in **D.8.15 TPU Parameter RAM** shows how parameter words are organized in memory.

The CPU32 specifies function parameters by writing to the appropriate RAM address. The TPU reads the RAM to determine channel operation. The TPU can also store information to be read by the CPU32 in the parameter RAM. Detailed descriptions of the parameters required by each time function are beyond the scope of this manual. Refer to the *TPU Reference Manual* (TPURM/AD) and the Motorola TPU Literature Package (TPULITPAK/D) for more information.

11.3 TPU Operation

All TPU functions are related to one of the two 16-bit time bases. Functions are synthesized by combining sequences of match events and capture events. Because the primitives are implemented in hardware, the TPU can determine precisely when a match or capture event occurs, and respond rapidly. An event register for each channel provides for simultaneity of match/capture event occurrences on all channels.

effect at the same time. Parameter RAM hardware supports coherent access of two adjacent 16-bit parameters. The host CPU must use a long-word operation to guarantee coherency.

11.3.6 Emulation Support

Although factory-programmed time functions can perform a wide variety of control tasks, they may not be ideal for all applications. The TPU provides emulation capability that allows the user to develop new time functions. Emulation mode is entered by setting the EMU bit in TPUMCR. In emulation mode, an auxiliary bus connection is made between TPURAM and the TPU, and access to TPURAM via the intermodule bus is disabled. A 9-bit address bus, a 32-bit data bus, and control lines transfer information between the modules. To ensure exact emulation, RAM module access timing remains consistent with access timing of the TPU microcode ROM control store.

To support changing TPU application requirements, Motorola has established a TPU function library. The function library is a collection of TPU functions written for easy assembly in combination with each other or with custom functions. Refer to Motorola Programming Note TPUPN00/D, *Using the TPU Function Library and TPU Emulation Mode* for information about developing custom functions and accessing the TPU function library. Refer to the *TPU Reference Manual* (TPURM/AD) and the Motorola TPU Literature Package (TPULITPAK/D) for more information about specific functions.

11.3.7 TPU Interrupts

Each of the TPU channels can generate an interrupt service request. Interrupts for each channel must be enabled by writing to the appropriate control bit in the channel interrupt enable register (CIER). The channel interrupt status register (CISR) contains one interrupt status flag per channel. Time functions set the flags. Setting a flag bit causes the TPU to make an interrupt service request if the corresponding channel interrupt enable bit is set and the interrupt request level is non-zero.

The value of the channel interrupt request level (CIRL) field in the TPU interrupt configuration register (TICR) determines the priority of all TPU interrupt service requests. CIRL values correspond to MCU interrupt request signals $\overline{\text{IRQ}}[7:1]$. $\overline{\text{IRQ7}}$ is the highest-priority request signal; $\overline{\text{IRQ1}}$ has the lowest priority. Assigning a value of %111 to CIRL causes $\overline{\text{IRQ7}}$ to be asserted when a TPU interrupt request is made; lower field values cause corresponding lower-priority interrupt request signals to be asserted. Assigning CIRL a value of %000 disables all interrupts.

The CPU32 recognizes only interrupt requests of a priority greater than the value contained in the interrupt priority (IP) mask in the status register. When the CPU32 acknowledges an interrupt request, the priority of the acknowledged interrupt is written to the IP mask and is driven out onto the IMB address lines.

When the IP mask value driven out on the address lines is the same as the CIRL value, the TPU contends for arbitration priority. The IARB field in TPUMCR contains the TPU arbitration number. Each module that can make an interrupt service request must be assigned a unique non-zero IARB value in order to implement an arbitration scheme.

Table 13-8 Example System Clock, CAN Bit Rate and S-Clock Frequencies

System Clock Frequency (MHz)	CAN Bit-Rate (MHz)	Possible S-Clock Frequency (MHz)	Possible Number of Time Quanta/Bit	PRESDIV Value + 1
25	1	25	25	1
20	1	10, 20	10, 20	2, 1
16	1	8, 16	8, 16	2, 1
25	0.125	1, 1.25, 2.5	8,10, 20	25, 20,10
20	0.125	1, 2, 2.5	8, 16, 20	20, 10, 8
16	0.125	1, 2	8,16	16, 8

13.4.3.1 Configuring the TouCAN Bit Timing

The following considerations must be observed when programming bit timing functions.

- If the programmed PRESDIV value results in a single system clock per one time quantum, then the PSEG2 field in CANCTRL2 register should not be programmed to zero.
- If the programmed PRESDIV value results in a single system clock per one time quantum, then the information processing time (IPT) equals three time quanta, otherwise it equals two time quanta. If PSEG2 equals two, then the TouCAN transmits one time quantum late relative to the scheduled sync segment.
- If the prescaler and bit timing control fields are programmed to values that result in fewer than ten system clock periods per CAN bit time and the CAN bus loading is 100%, anytime the rising edge of a start-of-frame (SOF) symbol transmitted by another node occurs during the third bit of the intermission between messages, the TouCAN may not be able to prepare a message buffer for transmission in time to begin its own transmission and arbitrate against the message which transmitted the early SOF.
- The TouCAN bit time must be programmed to be greater than or equal to nine system clocks, or correct operation is not guaranteed.

13.4.4 Error Counters

The TouCAN has two error counters, the transmit (TX) error counter and the receive (RX) error counter. Refer to **APPENDIX D REGISTER SUMMARY** for more information on error counters. The rules for increasing and decreasing these counters are described in the CAN protocol, and are fully implemented in the TouCAN. Each counter has the following features:

- 8-bit up/down counter
- 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

Table A-6 AC Timing (Continued)

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

Num	Characteristic	Symbol	Min	Max	Unit
76	Mode Select Hold Time	t_{MSH}	0	—	ns
77	$\overline{\text{RESET}}$ Assertion Time[13]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{\text{RESET}}$ Rise Time[14, 15]	t_{RSTR}	—	10	t_{cyc}

NOTES:

1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. The base configuration of the MC68336/376 requires a 20.97 MHz crystal reference.
3. When an external clock is used, minimum high and low times are based on a 50% duty cycle. The minimum allowable t_{Xcyc} period is reduced when the duty cycle of the external clock signal varies. The relationship between external clock input duty cycle and minimum t_{Xcyc} is expressed:
   Minimum t_{Xcyc} period = minimum t_{XCHL} / (50% –external clock input duty cycle tolerance).
4. Parameters for an external clock signal applied while the internal PLL is disabled (MODCLK pin held low during reset). Does not pertain to an external VCO reference applied while the PLL is enabled (MODCLK pin held high during reset). When the PLL is enabled, the clock synthesizer detects successive transitions of the reference signal. If transitions occur within the correct clock period, rise/fall times and duty cycle are not critical.
5. Address access time = (2.5 + WS) $t_{cyc} - t_{CHAV} - t_{DICL}$
   Chip select access time = (2 + WS) $t_{cyc} - t_{LSA} - t_{DICL}$
   Where: WS = number of wait states. When fast termination is used (2 clock bus) WS = –1.
6. Specification 9A is the worst-case skew between $\overline{\text{AS}}$ and $\overline{\text{DS}}$ or $\overline{\text{CS}}$. The amount of skew depends on the relative loading of these signals. When loads are kept within specified limits, skew will not cause $\overline{\text{AS}}$ and $\overline{\text{DS}}$ to fall outside the limits shown in specification 9.
7. If multiple chip selects are used, $\overline{\text{CS}}$ width negated (specification 15) applies to the time from the negation of a heavily loaded chip select to the assertion of a lightly loaded chip select. The $\overline{\text{CS}}$ width negated specification between multiple chip selects does not apply to chip selects being used for synchronous ECLK cycles.
8. Hold times are specified with respect to $\overline{\text{DS}}$ or $\overline{\text{CS}}$ on asynchronous reads and with respect to CLKOUT on fast cycle reads. The user is free to use either hold time.
9. Maximum value is equal to (t_{cyc} / 2) + 25 ns.
10. If the asynchronous setup time (specification 47A) requirements are satisfied, the $\overline{\text{DSACK}}$[1:0] low to data setup time (specification 31) and $\overline{\text{DSACK}}$[1:0] low to $\overline{\text{BERR}}$ low setup time (specification 48) can be ignored. The data must only satisfy the data-in to clock low setup time (specification 27) for the following clock cycle. $\overline{\text{BERR}}$ must satisfy only the late $\overline{\text{BERR}}$ low to clock low setup time (specification 27A) for the following clock cycle.
11. To ensure coherency during every operand transfer, $\overline{\text{BG}}$ will not be asserted in response to $\overline{\text{BR}}$ until after all cycles of the current operand transfer are complete and $\overline{\text{RMC}}$ is negated.
12. In the absence of $\overline{\text{DSACK}}$[1:0], $\overline{\text{BERR}}$ is an asynchronous input using the asynchronous setup time (specification 47A).
13. After external $\overline{\text{RESET}}$ negation is detected, a short transition period (approximately 2 t_{cyc}) elapses, then the SIM drives $\overline{\text{RESET}}$ low for 512 t_{cyc}.
14. External assertion of the $\overline{\text{RESET}}$ input can overlap internally-generated resets. To insure that an external reset is recognized in all cases, $\overline{\text{RESET}}$ must be asserted for at least 590 CLKOUT cycles.
15. External logic must pull $\overline{\text{RESET}}$ high during this period in order for normal MCU operation to begin.

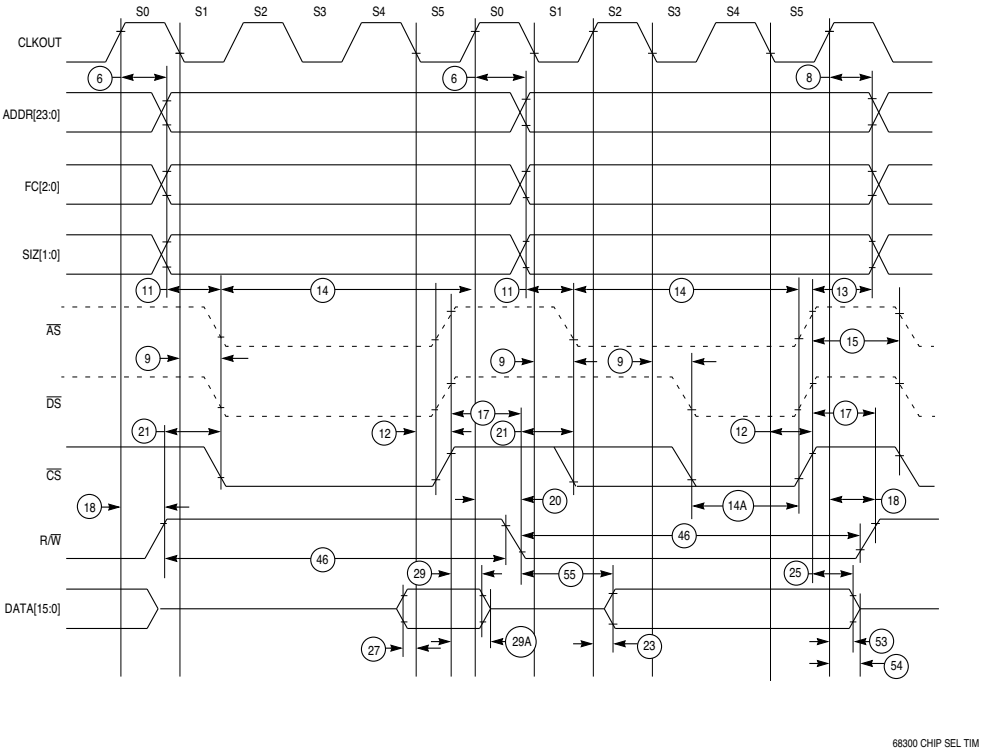


Figure A-11 Chip-Select Timing Diagram

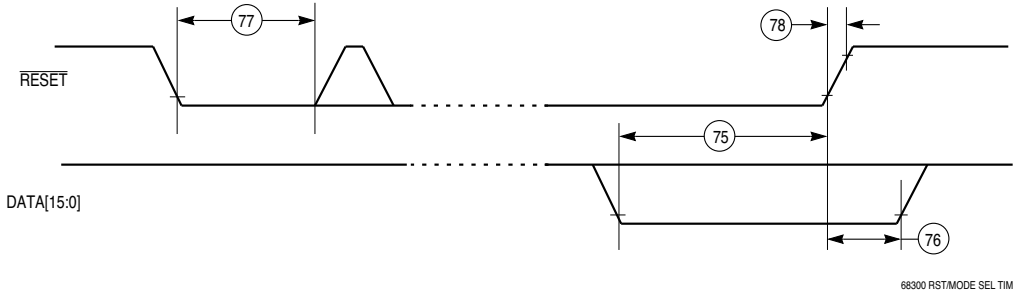


Figure A-12 Reset and Mode Select Timing Diagram

B.1 Obtaining Updated MC68336/376 Mechanical Information

Although all devices manufactured by Motorola conform to current JEDEC standards, complete mechanical information regarding MC68336/376 microcontrollers is available through Motorola's website at motorola.com

To download updated package specifications, go to website

B.2 Ordering Information

Refer to **Table B-1** for MC68336 ordering information and **Table B-2** for MC68376 ordering information. Contact a Motorola sales representative for information on ordering a custom ROM device.

Table B-1 MC68336 Ordering Information

Part Number	Package Type	Frequency (MHz)	TPU	Temperature	Package Order Quantity	Order Number
MC68336	160-pin QFP	20.97 MHz	A	–40 to +85 °C	2	SPMC68336ACFT20
					24	MC68336ACFT20
					120	MC68336ACFT20B1
				–40 to +105 °C	2	SPMC68336AVFT20
					24	MC68336AVFT20
					120	MC68336AVFT20B1
				–40 to +125 °C	2	SPMC68336AMFT20
					24	MC68336AMFT20
					120	MC68336AMFT20B1
			G	–40 to +85 °C	2	SPMC68336GCFT20
					24	MC68336GCFT20
					120	MC68336GCFT20B1
				–40 to +105 °C	2	SPMC68336GVFT20
					24	MC68336GVFT20
					120	MC68336GVFT20B1
				–40 to +125 °C	2	SPMC68336GMFT20
					24	MC68336GMFT20
					120	MC68336GMFT20B1

S — Supervisor/User State
0 = CPU operates at user privilege level
1 = CPU operates at supervisor privilege level

IP[2:0] — Interrupt Priority Mask
The priority value in this field (0 to 7) is used to mask interrupts.

X — Extend Flag
Used in multiple-precision arithmetic operations. In many instructions, it is set to the same value as the C bit.

N — Negative Flag
Set when the MSB of a result register is set.

Z — Zero Flag
Set when all bits of a result register are zero.

V — Overflow Flag
Set when two's complement overflow occurs as the result of an operation.

C — Carry Flag
Set when a carry or borrow occurs during an arithmetic operation. Also used during shift and rotate instructions to facilitate multiple word operations.

D.5 QADC Module

Table D-24 shows the QADC address map. The column labeled “Access” indicates the privilege level at which the CPU32 must be operating to access the register. A designation of “S” indicates that supervisor mode is required. A designation of “S/U” indicates that the register can be programmed for either supervisor mode access or unrestricted access.

Table D-24 QADC Address Map

Access	Address[1]	15 8	7 0
S	$YFF200	Module Configuration Register (QADCMCR)	
S	$YFF202	Test Register (QADCTEST)	
S	$YFF204	Interrupt Register (QADCINT)	
S/U	$YFF206	Port A Data (PORTQA)	Port B Data (PORTQB)
S/U	$YFF208	Port Data Direction Register (DDRQA)	
S/U	$YFF20A	Control Register 0 (QACR0)	
S/U	$YFF20C	Control Register 1 (QACR1)	
S/U	$YFF20E	Control Register 2 (QACR2)	
S/U	$YFF210	Status Register (QASR)	
—	$YFF212 – $YFF22E	Reserved	
S/U	$YFF230 – $YFF27E	Conversion Command Word (CCW) Table	
—	$YFF280 – $YFF2AE	Reserved	
S/U	$YFF2B0 – $YFF2FE	Result Word Table Right Justified, Unsigned Result Register (RJURR)	
—	$YFF300 – $YFF32E	Reserved	
S/U	$YFF330 – $YFF37E	Result Word Table Left Justified, Signed Result Register (LJSRR)	
—	$YFF380 – $YFF3AE	Reserved	
S/U	$YFF3B0 – $YFF3FE	Result Word Table Left Justified, Unsigned Result Register (LJURR)	

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

D.5.1 QADC Module Configuration Register

QADCMCR — Module Configuration Register **$YFF200**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ	NOT USED						SUPV	NOT USED			IARB[3:0]			
RESET:															
0	0							1				0	0	0	0

STOP — Low-Power Stop Mode Enable

When the STOP bit is set, the clock signal to the QADC is disabled, effectively turning off the analog circuitry.

0 = Enable QADC clock.

1 = Disable QADC clock.

Table D-44 DASM Mode Flag Status Bit States

Mode	Flag Status Bit State
DIS	FLAG bit is reset
IPWM	FLAG bit is set each time there is a capture on channel A
IPM	FLAG bit is set each time there is a capture on channel A, except for the first time
IC	FLAG bit is set each time there is a capture on channel A
OCB	FLAG bit is set each time there is a successful comparison on channel B
OCAB	FLAG bit is set each time there is a successful comparison on either channel A or B
OPWM	FLAG bit is set each time there is a successful comparison on channel A

The FLAG bit is set by hardware and cleared by software, or by system reset. Clear the FLAG bit either by writing a zero to it, having first read the bit as a one, or by selecting the DIS mode.

IL[2:0] — Interrupt Level

When the DASM generates an interrupt request, IL[2:0] determines which of the interrupt request signals is asserted. When a request is acknowledged, the CTM4 compares IL[2:0] to a mask value supplied by the CPU32 to determine whether to respond. IL[2:0] must have a value in the range of $0 (interrupts disabled) to $7 (highest priority).

IARB3 — Interrupt Arbitration Bit 3

This bit and the IARB[2:0] field in BIUMCR are concatenated to determine the interrupt arbitration number for the submodule requesting interrupt service. Refer to **D.7.1 BIU Module Configuration Register** for more information on IARB[2:0].

WOR — Wired-OR Mode

In the DIS, IPWM, IPM and IC modes, the WOR bit is not used. Reading this bit returns the value that was previously written.

In the OCB, OCAB and OPWM modes, the WOR bit selects whether the output buffer is configured for open-drain or normal operation.

0 = Output buffer operates in normal mode.
1 = Output buffer operates in open-drain mode.

BSL — Bus Select

This bit selects the time base bus connected to the DASM.

0 = DASM is connected to time base bus A.
1 = DASM is connected to time base bus B.

IN — Input Pin Status

In the DIS, IPWM, IPM and IC modes, this read-only status bit reflects the logic level on the input pin.

In the OCB, OCAB and OPWM modes, reading this bit returns the value latched on the output flip-flop, after EDPOL polarity selection.

Writing to this bit has no effect.

MODE[3:0] — DASM Mode Select

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

NOTE

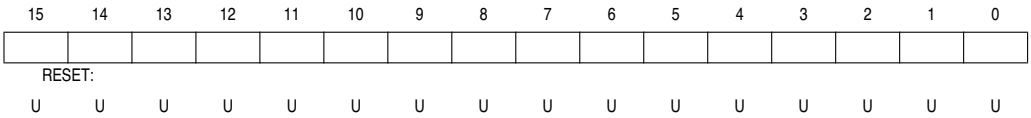
To avoid spurious interrupts, DASM interrupts should be disabled before changing the operating mode.

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.

D.10.5 Control Register 1

CANCTRL1 — Control Register 1 **$YFF087**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CANCTRL0								SAMP	LOOP	TSYNC	LBUF	RSVD	PROPSEG[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SAMP — Sampling Mode

The SAMP bit determines whether the TouCAN module will sample each received bit one time or three times to determine its value.

0 = One sample, taken at the end of phase buffer segment 1, is used to determine the value of the received bit.

1 = Three samples are used to determine the value of the received bit. The samples are taken at the normal sample point, and at the two preceding periods of the S-clock.

LOOP — TouCAN Loop Back

The LOOP bit configures the TouCAN to perform internal loop back. The bit stream output of the transmitter is fed back to the receiver. The receiver ignores the CANRX0 and CANRX1 pins. The CANTX0 and CANTX1 pins output a recessive state. In this state, the TouCAN ignores the ACK bit to ensure proper reception of its own messages.

0 = Internal loop back disabled.

1 = Internal loop back enabled.

TSYNC — Timer Synchronize Mode

The TSYNC bit enables the mechanism that resets the free-running timer each time a message is received in message buffer 0. This feature provides the means to synchronize multiple TouCAN stations with a special “SYNC” message (global network time).

0 = Timer synchronization disabled.

1 = Timer synchronization enabled.

NOTE

There can be a bit clock skew of four to five counts between different TouCAN modules that are using this feature on the same network.

LBUF — Lowest Buffer Transmitted First

The LBUF bit defines the transmit-first scheme.

0 = Message buffer with lowest ID is transmitted first.

1 = Lowest numbered buffer is transmitted first.

PROPSEG[2:0] — Propagation Segment Time

PROPSEG defines the length of the propagation segment in the bit time. The valid programmed values are 0 to 7. The propagation segment time is calculated as follows:

STUFFERR — Bit Stuff Error

The STUFFERR bit indicates whether or not the bit stuffing which occurred in the last transmitted or received message was correct.

0 = No bit stuffing error was detected since the last read of this register.

1 = A bit stuffing error was detected since the last read of this register.

TXWARN — Transmit Error Status Flag

The TXWARN status flag reflects the status of the TouCAN transmit error counter.

0 = Transmit error counter < 96.

1 = Transmit error counter ≥ 96.

RXWARN — Receiver Error Status Flag

The RXWARN status flag reflects the status of the TouCAN receive error counter.

0 = Receive error counter < 96.

1 = Receive error counter ≥ 96.

IDLE — Idle Status

The IDLE bit indicates when there is activity on the CAN bus.

0 = The CAN bus is not idle.

1 = The CAN bus is idle.

TX/$\overline{RX}$ — Transmit/Receive Status

The TX/$\overline{RX}$ bit indicates when the TouCAN module is transmitting or receiving a message. TX/$\overline{RX}$ has no meaning when IDLE = 1.

0 = The TouCAN is receiving a message if IDLE = 0.

1 = The TouCAN is transmitting a message if IDLE = 0.

FCS[1:0] — Fault Confinement State

The FCS[1:0] field describes the state of the TouCAN. Refer to **Table D-63**.

Table D-63 Fault Confinement State Encoding

FCS[1:0]	Bus State
00	Error active
01	Error passive
1X	Bus off

If the SOFTRST bit in CANMCR is asserted while the TouCAN is in the bus off state, the error and status register is reset, including FCS[1:0]. However, as soon as the TouCAN exits reset, FCS[1:0] bits will again reflect the bus off state. Refer to **13.4.4 Error Counters** for more information on entry into and exit from the various fault confinement states.

BOFFINT — Bus Off Interrupt

The BOFFINT bit is used to request an interrupt when the TouCAN enters the bus off state.

0 = No bus off interrupt requested.

1 = When the TouCAN state changes to bus off, this bit is set, and if the BOFFMSK bit in CANCTRL0 is set, an interrupt request is generated. This interrupt is not requested after reset.