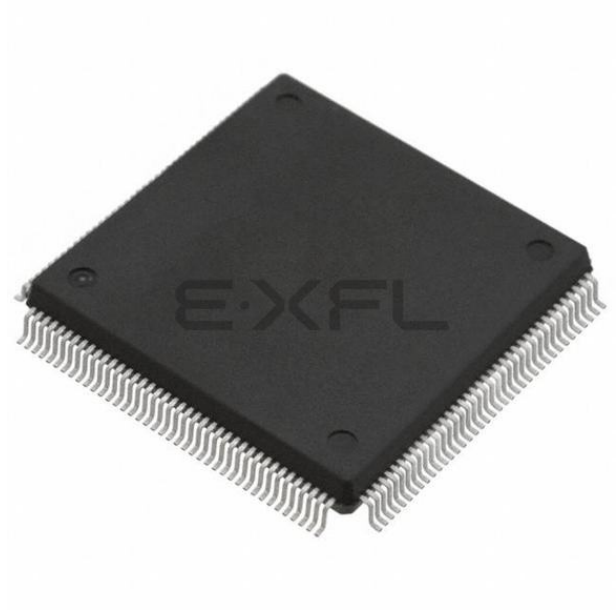

Welcome to E-XFL.COM

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 ~ 105°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/mc68336gvab20

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

3.5 Signal Descriptions

The following tables define the MC68336/376 signals. **Table 3-4** shows signal origin, type, and active state. **Table 3-5** describes signal functions. Both tables are sorted alphabetically by mnemonic. MCU pins often have multiple functions. More than one description can apply to a pin.

Table 3-4 MC68336/376 Signal Characteristics

Signal Name	MCU Module	Signal Type	Active State
ADDR[23:0]	SIM	Bus	—
AN[59:48]/[3:0]	QADC	Input	—
AN[w, x, y, z]	QADC	Input	—
$\overline{\text{AS}}$	SIM	Output	0
$\overline{\text{AVEC}}$	SIM	Input	0
$\overline{\text{BERR}}$	SIM	Input	0
$\overline{\text{BG}}$	SIM	Output	0
$\overline{\text{BGACK}}$	SIM	Input	0
$\overline{\text{BKPT}}$	CPU32	Input	0
$\overline{\text{BR}}$	SIM	Input	0
CLKOUT	SIM	Output	—
CANRX0 (MC68376 Only)	TouCAN	Input	—
CANTX0 (MC68376 Only)	TouCAN	Output	—
$\overline{\text{CS}}$[10:0]	SIM	Output	0
$\overline{\text{CSBOOT}}$	SIM	Output	0
CPWM[8:5]	CTM4	Output	—
CTD[10:9]/[4:3]	CTM4	Input/Output	—
CTM2C	CTM4	Input	—
DATA[15:0]	SIM	Bus	—
DS	SIM	Output	0
$\overline{\text{DSACK}}$[1:0]	SIM	Input	0
DSCLK	CPU32	Input	Serial Clock
DSI	CPU32	Input	Serial Data
DSO	CPU32	Output	Serial Data
ECLK	SIM	Output	—
ETRIG[2:1]	QADC	Input	—
EXTAL	SIM	Input	—
FC[2:0]	SIM	Output	—
FREEZE	SIM	Output	1
$\overline{\text{HALT}}$	SIM	Input/Output	0
$\overline{\text{IFETCH}}$	CPU32	Output	0
$\overline{\text{IPIPE}}$	CPU32	Output	0
$\overline{\text{IRQ}}$[7:1]	SIM	Input	0
MA[2:0]	QADC	Output	1
MISO	QSM	Input/Output	—
MODCLK	SIM	Input	—
MOSI	QSM	Input/Output	—
PC[6:0]	SIM	Output	—
PCS[3:0]	QSM	Input/Output	—
PE[7:0]	SIM	Input/Output	—
PF[7:0]	SIM	Input/Output	—

Table 3-5 MC68336/376 Signal Functions (Continued)

Mnemonic	Signal Name	Function
PQA[7:0]	QADC Port A	QADC port A digital input/output port signals
PQB[7:0]	QADC Port B	QADC port B digital input port signals
PQS[7:0]	Port QS	QSM digital input/output port signals
QUOT	Quotient Out	Provides the quotient bit of the polynomial divider (test mode only)
R/$\overline{W}$	Read/Write	Indicates the direction of data transfer on the bus
$\overline{RESET}$	Reset	System reset
$\overline{RMC}$	Read-Modify-Write Cycle	Indicates an indivisible read-modify-write instruction
RXD	SCI Receive Data	Serial input to the SCI
SCK	QSPI Serial Clock	Clock output from QSPI in master mode; clock input to QSPI in slave mode
SIZ[1:0]	Size	Indicates the number of bytes remaining to be transferred during a bus cycle
$\overline{SS}$	Slave Select	Starts serial transmission when QSPI is in slave mode; chip-select in master mode
T2CLK	TPU Clock	TPU clock input
TPUCH[15:0]	TPU I/O Channels	Bidirectional TPU channels
TSC	Three-State Control	Places all output drivers in a high impedance state
$\overline{TSTME}$	Test Mode Enable	Hardware enable for SIM test mode
TXD	SCI Transmit Data	Serial output from the SCI
XFC	External Filter Capacitor	Connection for external phase-locked loop filter capacitor

Table 4-2 Instruction Set Summary (Continued)

TBLSN/TBLUN	<ea>, Dn Dym : Dyn, Dn	8, 16, 32	Dyn – Dym ⇒ Temp (Temp ∗ Dn [7 : 0]) / 256 ⇒ Temp Dym + Temp ⇒ Dn
TRAP	#<data>	none	SSP – 2 ⇒ SSP; format/vector offset ⇒ (SSP); SSP – 4 ⇒ SSP; PC ⇒ (SSP); SR ⇒ (SSP); vector address ⇒ PC
TRAPcc	none #<data>	none 16, 32	If cc true, then TRAP exception
TRAPV	none	none	If V set, then overflow TRAP exception
TST	<ea>	8, 16, 32	Source – 0, to set condition codes
UNLK	An	32	An ⇒ SP; (SP) ⇒ An, SP + 4 ⇒ SP

NOTES:
1. Privileged instruction.

4.8.1 M68000 Family Compatibility

It is the philosophy of the M68000 family that all user-mode programs can execute unchanged on future derivatives of the M68000 family, and supervisor-mode programs and exception handlers should require only minimal alteration.

The CPU32 can be thought of as an intermediate member of the M68000 Family. Object code from an MC68000 or MC68010 may be executed on the CPU32. Many of the instruction and addressing mode extensions of the MC68020 are also supported. Refer to the *CPU32 Reference Manual* (CPU32RM/AD) for a detailed comparison of the CPU32 and MC68020 instruction set.

4.8.2 Special Control Instructions

Low-power stop (LPSTOP) and table lookup and interpolate (TBL) instructions have been added to the MC68000 instruction set for use in controller applications.

4.8.2.1 Low-Power Stop (LPSTOP)

In applications where power consumption is a consideration, the CPU32 forces the device into a low-power standby mode when immediate processing is not required. The low-power stop mode is entered by executing the LPSTOP instruction. The processor remains in this mode until a user-specified (or higher) interrupt level or reset occurs.

4.8.2.2 Table Lookup and Interpolate (TBL)

To maximize throughput for real-time applications, reference data is often precalculated and stored in memory for quick access. Storage of many data points can require an inordinate amount of memory. The table lookup instruction requires that only a sample of data points be stored, reducing memory requirements. The TBL instruction recovers intermediate values using linear interpolation. Results can be rounded with a round-to-nearest algorithm.

Bus cycles terminated by DSACK assertion normally require a minimum of three CLKOUT cycles. To support systems that use CLKOUT to generate DSACK and other inputs, asynchronous input setup time and asynchronous input hold times are specified. When these specifications are met, the MCU is guaranteed to recognize the appropriate signal on a specific edge of the CLKOUT signal.

For a read cycle, when assertion of DSACK is recognized on a particular falling edge of the clock, valid data is latched into the MCU on the next falling clock edge, provided that the data meets the data setup time. In this case, the parameter for asynchronous operation can be ignored.

When a system asserts DSACK for the required window around the falling edge of S2 and obeys the bus protocol by maintaining DSACK and BERR or HALT until and throughout the clock edge that negates AS (with the appropriate asynchronous input hold time), no wait states are inserted. The bus cycle runs at the maximum speed of three clocks per cycle.

To ensure proper operation in a system synchronized to CLKOUT, when either BERR or BERR and HALT is asserted after DSACK, BERR (or BERR and HALT) assertion must satisfy the appropriate data-in setup and hold times before the falling edge of the clock cycle after DSACK is recognized.

5.6.2 Regular Bus Cycles

The following paragraphs contain a discussion of cycles that use external bus control logic. Refer to **5.6.3 Fast Termination Cycles** for information about fast termination cycles.

To initiate a transfer, the MCU asserts an address and the SIZ[1:0] signals. The SIZ signals and ADDR0 are externally decoded to select the active portion of the data bus. Refer to **5.5.2 Dynamic Bus Sizing**. When AS, DS, and R/W are valid, a peripheral device either places data on the bus (read cycle) or latches data from the bus (write cycle), then asserts a DSACK[1:0] combination that indicates port size.

The DSACK[1:0] signals can be asserted before the data from a peripheral device is valid on a read cycle. To ensure valid data is latched into the MCU, a maximum period between DSACK assertion and DS assertion is specified.

There is no specified maximum for the period between the assertion of AS and DSACK. Although the MCU can transfer data in a minimum of three clock cycles when the cycle is terminated with DSACK, the MCU inserts wait cycles in clock period increments until either DSACK signal goes low.

The internal pointer is initialized to the same value as NEWQP. During normal operation, the command pointed to by the internal pointer is executed, the value in the internal pointer is copied into CPTQP, the internal pointer is incremented, and then the sequence repeats. Execution continues at the internal pointer address unless the NEWQP value is changed. After each command is executed, ENDQP and CPTQP are compared. When a match occurs, the SPIF flag is set and the QSPI stops and clears SPE, unless wrap-around mode is enabled.

At reset, NEWQP is initialized to $0. When the QSPI is enabled, execution begins at queue address $0 unless another value has been written into NEWQP. ENDQP is initialized to $0 at reset, but should be changed to show the last queue entry before the QSPI is enabled. NEWQP and ENDQP can be written at any time. When NEWQP changes, the internal pointer value also changes. However, if NEWQP is written while a transfer is in progress, the transfer is completed normally. Leaving NEWQP and ENDQP set to $0 transfers only the data in transmit RAM location $0.

9.3.5 QSPI Operating Modes

The QSPI operates in either master or slave mode. Master mode is used when the MCU initiates data transfers. Slave mode is used when an external device initiates transfers. Switching between these modes is controlled by MSTR in SPCR0. Before entering either mode, appropriate QSM and QSPI registers must be initialized properly.

In master mode, the QSPI executes a queue of commands defined by control bits in each command RAM queue entry. Chip-select pins are activated, data is transmitted from the transmit RAM and received by the receive RAM.

In slave mode, operation proceeds in response to $\overline{SS}$ pin activation by an external SPI bus master. Operation is similar to master mode, but no peripheral chip selects are generated, and the number of bits transferred is controlled in a different manner. When the QSPI is selected, it automatically executes the next queue transfer to exchange data with the external device correctly.

Although the QSPI inherently supports multi-master operation, no special arbitration mechanism is provided. A mode fault flag (MODF) indicates a request for SPI master arbitration. System software must provide arbitration. Note that unlike previous SPI systems, MSTR is not cleared by a mode fault being set nor are the QSPI pin output drivers disabled. The QSPI and associated output drivers must be disabled by clearing SPE in SPCR1.

Figure 9-4 shows QSPI initialization. **Figures 9-5** through **9-9** show QSPI master and slave operation. The CPU32 must initialize the QSM global and pin registers and the QSPI control registers before enabling the QSPI for either mode of operation (refer to **9.5 QSM Initialization**). The command queue must be written before the QSPI is enabled for master mode operation. Any data to be transmitted should be written into transmit RAM before the QSPI is enabled. During wrap-around operation, data for subsequent transmissions can be written at any time.

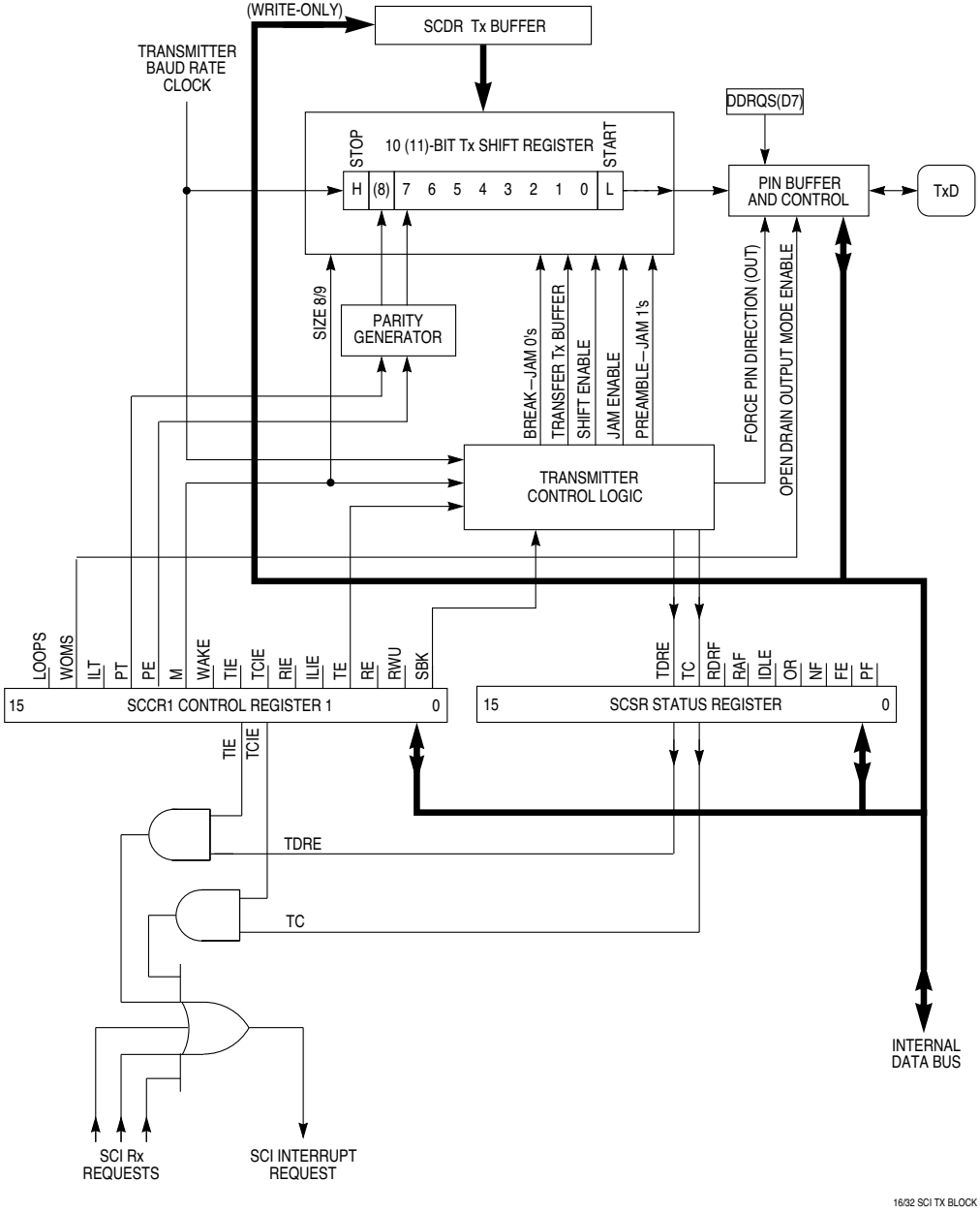

Figure 9-10 SCI Transmitter Block Diagram

- 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:

10.8.1 DASM Interrupts

The DASM can optionally request an interrupt when the FLAG bit in DASMSIC is set. To enable interrupts, set the IL[2:0] field in DASMSIC to a non-zero value. The CTM4 compares the CPU32 IP mask value to the priority of the requested interrupt designated by IL[2:0] to determine whether it should contend for arbitration priority. During arbitration, the BIUSM provides the arbitration value specified by IARB[2:0] in BIUMCR and IARB3 in DASMSIC. If the CTM4 wins arbitration, it responds with a vector number generated by concatenating VECT[7:6] in BIUMCR and the six low-order bits specified by the number of the submodule requesting service. Thus, for DASM9 in the CTM4, the six low-order bits would be nine in decimal, or %001001 in binary.

10.8.2 DASM Registers

The DASM contains one status/interrupt/control register and two data registers (A and B). 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 four DASMs, each with its own set of registers. Refer to **D.7.11 DASM Status/Interrupt/Control Registers**, **D.7.12 DASM Data Register A**, and **D.7.13 DASM Data Register B** for information concerning DASM register and bit descriptions.

10.9 Pulse-Width Modulation Submodule (PWMSM)

The PWMSM allows pulse width modulated signals to be generated over a wide range of frequencies, independently of other CTM4 output signals. The output pulse width duty cycle can vary from 0% to 100%, with 16 bits of resolution. The minimum pulse width is twice the MCU system clock period. For example, the minimum pulse width is 95.4 ns when using a 20.97 MHz clock.

The PWMSM is composed of:

- An output flip-flop with output polarity control
- Clock prescaler and selection logic
- A 16-bit up-counter
- Two registers to hold the current and next pulse width values
- Two registers to hold the current and next pulse period values
- A pulse width comparator
- A system state sequencer
- Logic to create 0% and 100% pulses
- Interrupt logic
- A status, interrupt and control register
- A submodule bus interface

The PWMSM includes its own time base counter and does not use the CTM4 time base buses; however, it does use the prescaled clock signal PCLK1 generated by the CPSM. Refer to **10.5 Counter Prescaler Submodule (CPSM)** and **Figure 10-1** for more information. **Figure 10-6** shows a block diagram of the PWMSM.

11.6.2.4 Host Service Registers

The host service request field selects the type of host service request for the time function selected on a given channel. The meaning of the host service request bits is determined by time function microcode. Refer to the *TPU Reference Manual* (TPURM/AD) and the Motorola TPU Literature Package (TPULITPAK/D) for more information.

A host service request field of %00 signals the CPU that service is completed and that there are no further pending host service requests. The host can request service on a channel by writing the corresponding host service request field to one of three nonzero states. It is imperative for the CPU to monitor the host service request register and wait until the TPU clears the service request for a channel before changing any parameters or issuing a new service request to the channel.

11.6.2.5 Channel Priority Registers

The channel priority registers (CPR1, CPR2) assign one of three priority levels to a channel or disable the channel. **Table 11-4** indicates the number of time slots guaranteed for each channel priority encoding.

Table 11-4 Channel Priority Encodings

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

11.6.3 Development Support and Test Registers

These registers are used for custom microcode development or for factory test. Describing the use of these registers is beyond the scope of this manual. Register descriptions are provided in **D.8 Time Processor Unit (TPU)**. Refer to the *TPU Reference Manual* (TPURM/AD) for more information.

Table A-15 FCSM Timing Characteristics

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

Num	Parameter	Symbol	Min	Max	Unit
1	Input pin frequency[1]	f_{PCNTR}	0	$f_{sys}/4$	MHz
2	Input pin low time	t_{PINL}	$2.0/f_{sys}$	—	µs
3	Input pin high time	t_{PINH}	$2.0/f_{sys}$	—	µs
4	Clock pin to counter increment	t_{PINC}	$4.5/f_{sys}$	$6.5/f_{sys}$	µs
5	Clock pin to new TBB value	t_{PTBB}	$5.0/f_{sys}$	$7.0/f_{sys}$	µs
6	Clock pin to COF set ($FFFF)	t_{PCOF}	$4.5/f_{sys}$	$6.5/f_{sys}$	µs
7	Pin to IN bit delay	t_{PINB}	$1.5/f_{sys}$	$2.5/f_{sys}$	µs
8	Flag to IMB interrupt request	t_{FIRQ}	$1.0/f_{sys}$	$1.0/f_{sys}$	µs
9	Counter resolution[2]	t_{CRES}	—	$2.0/f_{sys}$	µs

NOTES:
1. Value applies when using external clock.
2. Value applies when using internal clock. Minimum counter resolution depends on prescaler divide ratio selection.

Table A-16 MCSM 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 frequency[1]	f_{PCNTR}	0	$f_{sys}/4$	MHz
2	Input pin low time	t_{PINL}	$2.0/f_{sys}$	—	µs
3	Input pin high time	t_{PINH}	$2.0/f_{sys}$	—	µs
4	Clock pin to counter increment	t_{PINC}	$4.5/f_{sys}$	$6.5/f_{sys}$	µs
5	Clock pin to new TBB value	t_{PTBB}	$5.0/f_{sys}$	$7.0/f_{sys}$	µs
6	Clock pin to COF set ($FFFF)	t_{PCOF}	$4.5/f_{sys}$	$6.5/f_{sys}$	µs
7	Load pin to new counter value	t_{PLOAD}	$2.5/f_{sys}$	$3.5/f_{sys}$	µs
8	Pin to IN bit delay	t_{PINB}	$1.5/f_{sys}$	$2.5/f_{sys}$	µs
9	Flag to IMB interrupt request	t_{FIRQ}	$1.0/f_{sys}$	$1.0/f_{sys}$	µs
10	Counter resolution[2]	t_{CRES}	—	$2.0/f_{sys}$	µs

NOTES:
1. Value applies when using external clock.
2. Value applies when using internal clock. Minimum counter resolution depends on prescaler divide ratio selection.

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.

Table D-7 Software Watchdog Timing Field

SWP	SWT[1:0]	Watchdog Time-Out Period
0	00	$2^9 \div f_{sys}$
0	01	$2^{11} \div f_{sys}$
0	10	$2^{13} \div f_{sys}$
0	11	$2^{15} \div f_{sys}$
1	00	$2^{18} \div f_{sys}$
1	01	$2^{20} \div f_{sys}$
1	10	$2^{22} \div f_{sys}$
1	11	$2^{24} \div f_{sys}$

HME — Halt Monitor Enable

0 = Halt monitor is disabled.
1 = Halt monitor is enabled.

BME — Bus Monitor External Enable

0 = Disable bus monitor for internal to external bus cycle.
1 = Enable bus monitor for internal to external bus cycle.

BMT[1:0] — Bus Monitor Timing

This field selects the bus monitor time-out period. Refer to **Table D-8**.

Table D-8 Bus Monitor Time-Out 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

D.2.13 Periodic Interrupt Control Register

PICR — Periodic Interrupt Control Register **$YFFA22**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	PIRQL[2:0]			PIV[7:0]							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

PICR sets the interrupt level and vector number for the periodic interrupt timer (PIT). Bits [10:0] can be read or written at any time. Bits [15:11] are unimplemented and always read zero.

PIRQL[2:0] — Periodic Interrupt Request Level

This field determines the priority of periodic interrupt requests. A value of %000 disables PIT interrupts.

ENDQP[3:0] — Ending Queue Pointer
This field contains the last QSPI queue address.

Bits [7:4] — Not Implemented

NEWQP[3:0] — New Queue Pointer Value
This field contains the first QSPI queue address.

D.6.14 QSPI Control Register 3

SPCR3 — QSPI Control Register **$YFFC1E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LOOPQ	HMIE	HALT	SPSR							
RESET:															
0	0	0	0	0	0	0	0								

SPCR3 contains the loop mode enable bit, halt and mode fault interrupt enable, and the halt control bit. The CPU32 has read/write access to SPCR3, but the QSM has read access only. SPCR3 must be initialized before QSPI operation begins. Writing a new value to SPCR3 while the QSPI is enabled disrupts operation.

Bits [15:11] — Not Implemented

LOOPQ — QSPI Loop Mode
0 = Feedback path disabled.
1 = Feedback path enabled.
LOOPQ controls feedback on the data serializer for testing.

HMIE — HALTA and MODF Interrupt Enable
0 = HALTA and MODF interrupts disabled.
1 = HALTA and MODF interrupts enabled.
HMIE enables interrupt requests generated by the HALTA status flag or the MODF status flag in SPSR.

HALT — Halt QSPI
0 = QSPI operates normally.
1 = QSPI is halted for subsequent restart.
When HALT is set, the QSPI stops on a queue boundary. It remains in a defined state from which it can later be restarted.

Table D-48 DASMB Operations

Mode	DASMB Operation
DIS	DASMB can be accessed to prepare a value for a subsequent mode selection. In this mode, register B1 is accessed in order to prepare a value for the OPWM mode. Unused register B2 is hidden and cannot be read, but is written with the same value as register B1 is written.
IPWM	DASMB contains the captured value corresponding to the trailing edge of the measured pulse. In this mode, register B2 is accessed. Buffer register B1 is hidden and cannot be accessed.
IPM	DASMB contains the captured value corresponding to the most recently detected user-specified rising or falling edge. In this mode, register B2 is accessed. Buffer register B1 is hidden and cannot be accessed.
IC	DASMB contains the captured value corresponding to the most recently detected user-specified rising or falling edge. In this mode, register B2 is accessed. Buffer register B1 is hidden and cannot be accessed.
OCB	DASMB is loaded with the value corresponding to the trailing edge of the pulse to be generated. Writing to DASMB in the OCB and OCAB modes also enables the corresponding channel B comparator until the next successful comparison. In this mode, register B2 is accessed. Buffer register B1 is hidden and cannot be accessed.
OCAB	DASMB is loaded with the value corresponding to the trailing edge of the pulse to be generated. Writing to DASMB in the OCB and OCAB modes also enables the corresponding channel B comparator until the next successful comparison. In this mode, register B2 is accessed. Buffer register B1 is hidden and cannot be accessed.
OPWM	DASMB is loaded with the value corresponding to the trailing edge of the PWM pulse to be generated. In this mode, register B1 is accessed. Buffer register B2 is hidden and cannot be accessed.

D.7.14 PWM Status/Interrupt/Control Register

PWM5SIC — PWM5 Status/Interrupt/Control Register **$YFF428**
PWM6SIC — PWM6 Status/Interrupt/Control Register **$YFF430**
PWM7SIC — PWM7 Status/Interrupt/Control Register **$YFF438**
PWM8SIC — PWM8 Status/Interrupt/Control Register **$YFF440**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAG	IL[2:0]			IARB3	NOT USED			PIN	NOT USED	LOAD	POL	EN	CLK[2:0]		
RESET:															
0	0	0	0	0				0		0	0	0	0	0	0

FLAG — Period Completion Status

This status bit indicates when the PWM output period has been completed.

0 = PWM period is not complete.
1 = PWM period is complete.

The FLAG bit is set each time a PWM period is completed. Whenever the PWM is enabled, the FLAG bit is set immediately to indicate that the contents of the buffer registers PWMA2 and PWMB2 have been updated, and that the period using these new values has started. It also indicates that the user accessible period and pulse width registers PWMA1 and PWMB1 can be loaded with values for the next PWM period. Once set, the FLAG bit remains set and is not affected by any subsequent period completions, until it is cleared.

Only software can clear the FLAG bit. To clear FLAG, first read the bit as one then write a zero to the bit. Writing a one to FLAG has no effect. When the PWM is disabled, FLAG remains cleared.

TCR1P[1:0] — Timer Count Register 1 Prescaler Control

TCR1 is clocked from the output of a prescaler. The prescaler's input is the internal TPU system clock divided by either 4 or 32, depending on the value of the PSCK bit. The prescaler divides this input by 1, 2, 4, or 8. Channels using TCR1 have the capability to resolve down to the TPU system clock divided by four. **Table D-52** is a summary of prescaler output.

Table D-52 TCR1 Prescaler Control Bits

TCR1P[1:0]	Prescaler Divide By	TCR1 Clock Input	
		PSCK = 0	PSCK = 1
00	1	$f_{sys} \div 32$	$f_{sys} \div 4$
01	2	$f_{sys} \div 64$	$f_{sys} \div 8$
10	4	$f_{sys} \div 128$	$f_{sys} \div 16$
11	8	$f_{sys} \div 256$	$f_{sys} \div 32$

TCR2P[1:0] — Timer Count Register 2 Prescaler Control

TCR2 is clocked from the output of a prescaler. If T2CG = 0, the input to the TCR2 prescaler is the external TCR2 clock source. If T2CG = 1, the input is the TPU system clock divided by eight. The TCR2P field specifies the value of the prescaler: 1, 2, 4, or 8. Channels using TCR2 have the capability to resolve down to the TPU system clock divided by eight. **Table D-53** is a summary of prescaler output.

Table D-53 TCR2 Prescaler Control Bits

TCR2P[1:0]	Prescaler Divide By	Internal Clock Divided By	External Clock Divided By
00	1	8	1
01	2	16	2
10	4	32	4
11	8	64	8

EMU — Emulation Control

In emulation mode, the TPU executes microinstructions from TPURAM exclusively. Access to the TPURAM module via the IMB is blocked, and the TPURAM module is dedicated for use by the TPU. After reset, this bit can be written only once.

0 = TPU and TPURAM operate normally.

1 = TPU and TPURAM operate in emulation mode.

T2CG — TCR2 Clock/Gate Control

When T2CG is set, the external TCR2 pin functions as a gate of the DIV8 clock (the TPU system clock divided by eight). In this case, when the external TCR2 pin is low, the DIV8 clock is blocked, preventing it from incrementing TCR2. When the external TCR2 pin is high, TCR2 is incremented at the frequency of the DIV8 clock. When T2CG is cleared, an external clock input from the TCR2 pin, which has been synchronized and fed through a digital filter, increments TCR2.

0 = TCR2 pin used as clock source for TCR2.

1 = TCR2 pin used as gate of DIV8 clock for TCR2.

Table D-57 Parameter RAM Address Map

Channel Number	Base Address	Parameter							
		0	1	2	3	4	5	6	7
0	$YFFF##[1, 2]	00	02	04	06	08	0A	—	—
1	$YFFF##	10	12	14	16	18	1A	—	—
2	$YFFF##	20	22	24	26	28	2A	—	—
3	$YFFF##	30	32	34	36	38	3A	—	—
4	$YFFF##	40	42	44	46	48	4A	—	—
5	$YFFF##	50	52	54	56	58	5A	—	—
6	$YFFF##	60	62	64	66	68	6A	—	—
7	$YFFF##	70	72	74	76	78	7A	—	—
8	$YFFF##	80	82	84	86	88	8A	—	—
9	$YFFF##	90	92	94	96	98	9A	—	—
10	$YFFF##	A0	A2	A4	A6	A8	AA	—	—
11	$YFFF##	B0	B2	B4	B6	B8	BA	—	—
12	$YFFF##	C0	C2	C4	C6	C8	CA	—	—
13	$YFFF##	D0	D2	D4	D6	D8	DA	—	—
14	$YFFF##	E0	E2	E4	E6	E8	EA	EC	EE
15	$YFFF##	F0	F2	F4	F6	F8	FA	FC	FE

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.
2. ## = Not implemented.