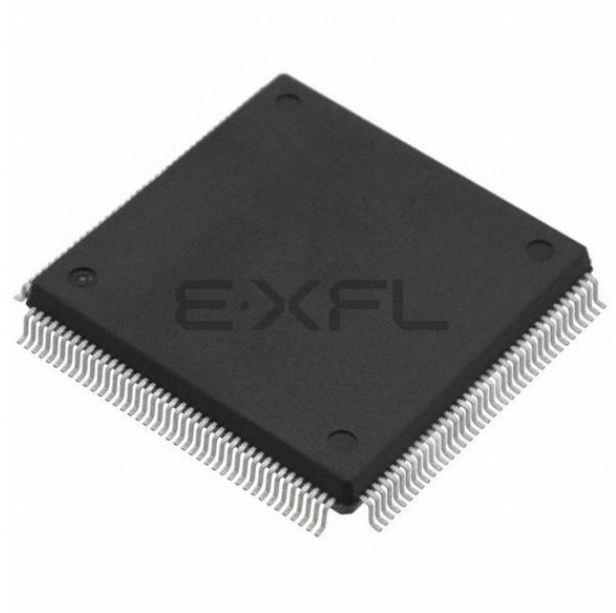

Welcome to E-XFL.COM

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/mc68376bgvab20

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

SECTION 12 STANDBY RAM WITH TPU EMULATION

SECTION 13 CAN 2.0B CONTROLLER MODULE (TouCAN)

LIST OF ILLUSTRATIONS

LIST OF TABLES

3.1.10 CAN 2.0B Controller Module (TouCAN)

- Full implementation of CAN protocol specification, version 2.0 A and B
- 16 receive/transmit message buffers of 0 to 8 bytes data length
- Global mask register for message buffers 0 to 13
- Independent mask registers for message buffers 14 and 15
- Programmable transmit-first scheme: lowest ID or lowest buffer number
- 16-bit free-running timer for message time-stamping
- Low power sleep mode with programmable wake-up on bus activity

3.2 Intermodule Bus

The intermodule bus (IMB) is a standardized bus developed to facilitate both design and operation of modular microcontrollers. It contains circuitry to support exception processing, address space partitioning, multiple interrupt levels, and vectored interrupts. The standardized modules in the MCU communicate with one another through the IMB. The IMB in the MCU uses 24 address and 16 data lines.

3.3 System Block Diagram and Pin Assignment Diagrams

Figure 3-1 is a functional diagram of the MCU. There is not a one-to-one correspondence between location and size of blocks in the diagram and location and size of integrated-circuit modules. **Figure 3-2** shows the MC68336 pin assignment package; **Figure 3-3** shows the MC68376 pin assignment package. Note that the MC68376 is a pin-compatible upgrade for the MC68336 that provides a CAN protocol controller and an 8-Kbyte masked ROM module. Both devices use a 160-pin plastic surface-mount package. Refer to **B.1 Obtaining Updated MC68336/376 Mechanical Information** for package dimensions. Refer to subsequent paragraphs in this section for pin and signal descriptions.

Table 3-4 MC68336/376 Signal Characteristics (Continued)

Signal Name	MCU Module	Signal Type	Active State
PQA[7:0]	QADC	Input/Output	—
PQB[7:0]	QADC	Input	—
PQS[7:0]	QSM	Input/Output	—
QUOT	SIM	Output	—
R/$\overline{W}$	SIM	Output	1/0
$\overline{RESET}$	SIM	Input/Output	0
$\overline{RMC}$	SIM	Output	0
RXD	QSM	Input	—
SCK	QSM	Input/Output	—
SIZ[1:0]	SIM	Output	1
$\overline{SS}$	QSM	Input	0
T2CLK	TPU	Input	—
TPUCH[15:0]	TPU	Input/Output	—
$\overline{TSTME}$/TSC	SIM	Input	0/1
TXD	QSM	Output	—
XFC	SIM	Input	—
XTAL	SIM	Output	—

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.

The MCU always attempts to transfer the maximum amount of data on all bus cycles. For any bus access, it is assumed that the port is 16 bits wide when the bus cycle begins.

Operand bytes are designated as shown in **Figure 5-9**. OP[0:3] represent the order of access. For instance, OP0 is the most significant byte of a long-word operand, and is accessed first, while OP3, the least significant byte, is accessed last. The two bytes of a word-length operand are OP0 (most significant) and OP1. The single byte of a byte-length operand is OP0.

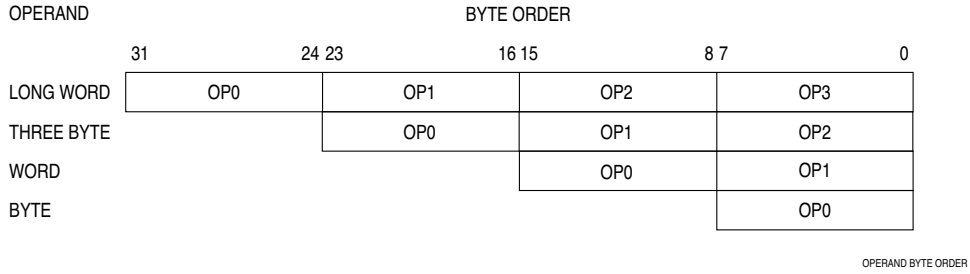

Figure 5-9 Operand Byte Order

5.5.3 Operand Alignment

The EBI data multiplexer establishes the necessary connections for different combinations of address and data sizes. The multiplexer takes the two bytes of the 16-bit bus and routes them to their required positions. Positioning of bytes is determined by the size and address outputs. SIZ1 and SIZ0 indicate the number of bytes remaining to be transferred during the current bus cycle. The number of bytes transferred is equal to or less than the size indicated by SIZ1 and SIZ0, depending on port width.

ADDR0 also affects the operation of the data multiplexer. During an operand transfer, ADDR[23:1] indicate the word base address of the portion of the operand to be accessed. ADDR0 indicates the byte offset from the base.

5.5.4 Misaligned Operands

The CPU32 uses a basic operand size of 16 bits. An operand is misaligned when it overlaps a word boundary. This is determined by the value of ADDR0. When ADDR0 = 0 (an even address), the address is on a word and byte boundary. When ADDR0 = 1 (an odd address), the address is on a byte boundary only. A byte operand is aligned at any address; a word or long-word operand is misaligned at an odd address. The CPU32 does not support misaligned transfers.

The largest amount of data that can be transferred by a single bus cycle is an aligned word. If the MCU transfers a long-word operand through a 16-bit port, the most significant operand word is transferred on the first bus cycle and the least significant operand word is transferred on a following bus cycle.

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.

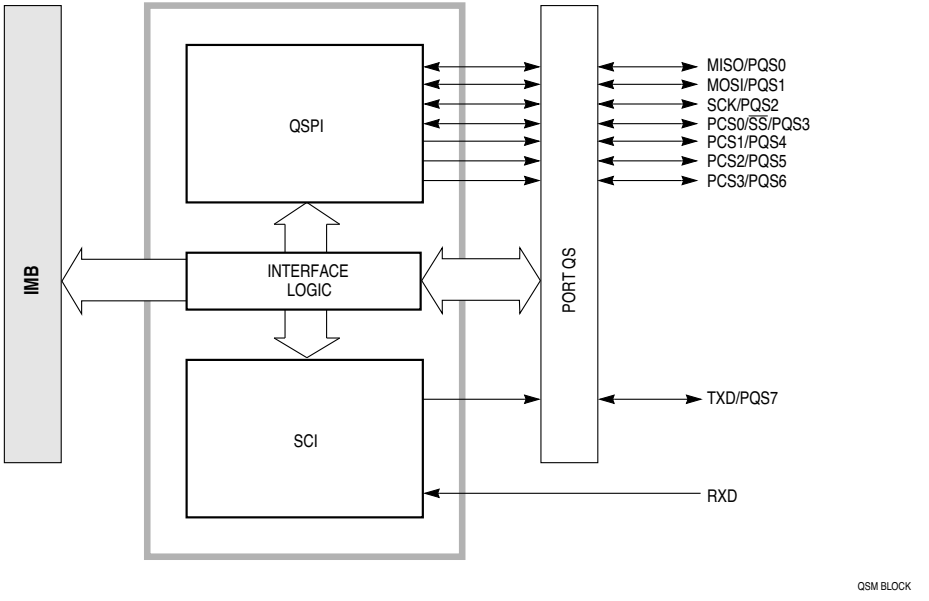

Figure 9-1 QSM Block Diagram

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.

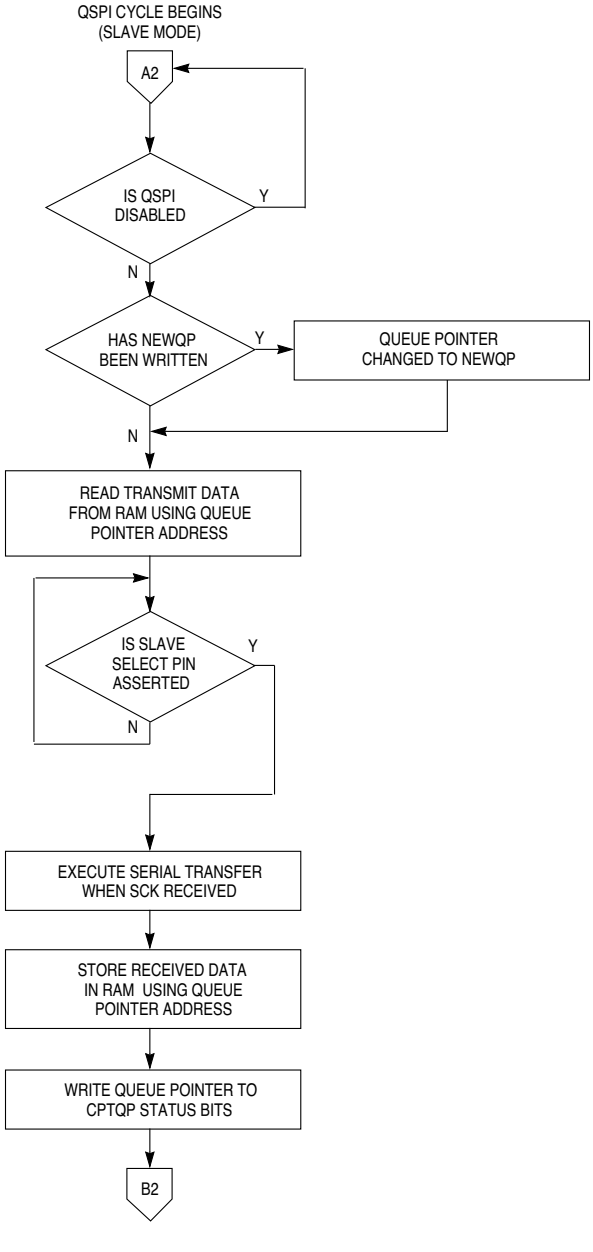

Figure 9-8 Flowchart of QSPI Slave Operation (Part 1)

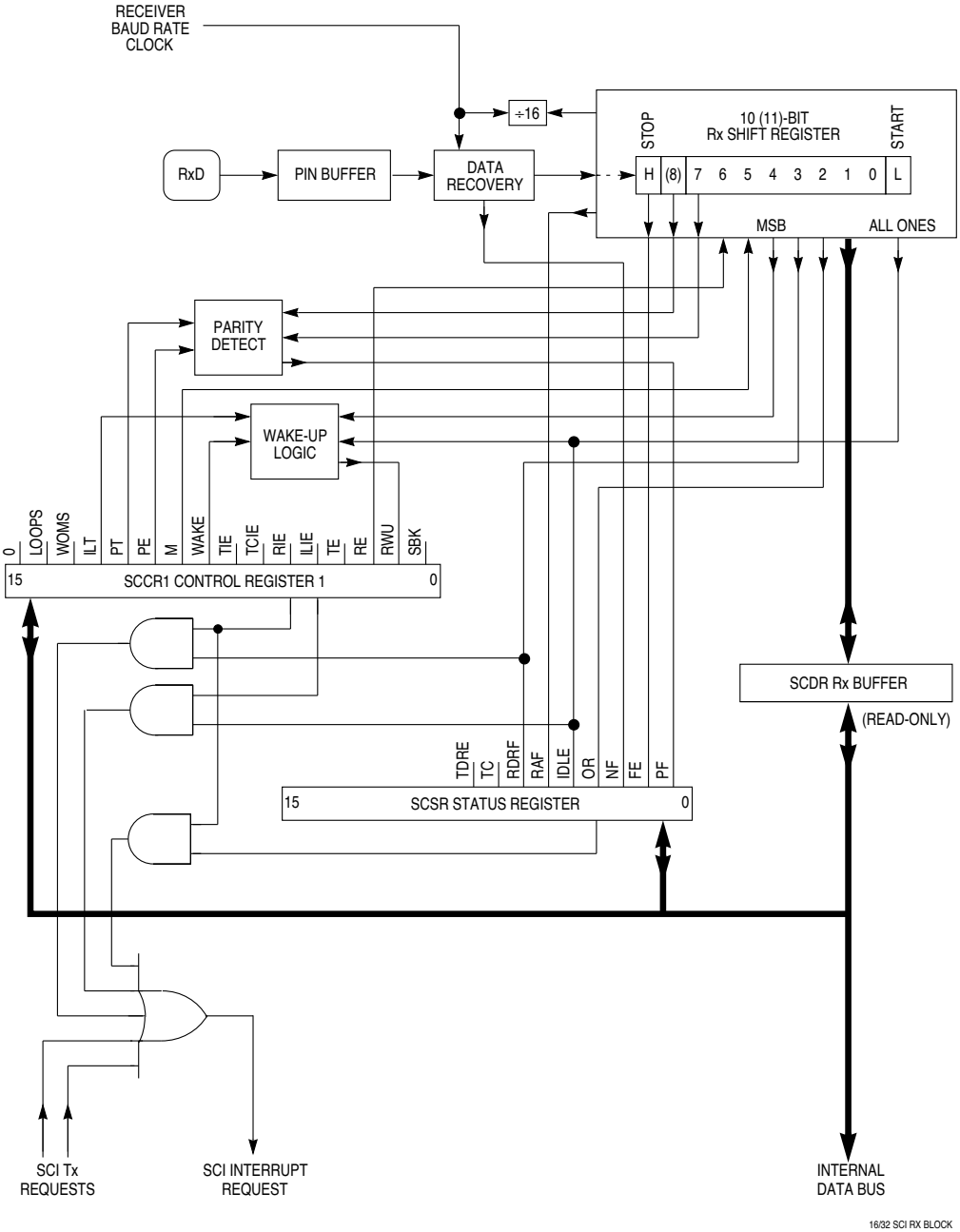

Figure 9-11 SCI Receiver Block Diagram

SECTION 8 QUEUED ANALOG-TO-DIGITAL CONVERTER MODULE

This section is an overview of the queued analog-to-digital converter (QADC) module. Refer to the *QADC Reference Manual* (QADCRM/AD) for a comprehensive discussion of QADC capabilities.

8.1 General

The QADC consists of an analog front-end and a digital control subsystem, which includes an intermodule bus (IMB) interface block. Refer to **Figure 8-1**.

The analog section includes input pins, an analog multiplexer, and two sample and hold analog circuits. The analog conversion is performed by the digital-to-analog converter (DAC) resistor-capacitor array and a high-gain comparator.

The digital control section contains the conversion sequencing logic, channel selection logic, and a successive approximation register (SAR). Also included are the periodic/ interval timer, control and status registers, the conversion command word (CCW) table RAM, and the result word table RAM.

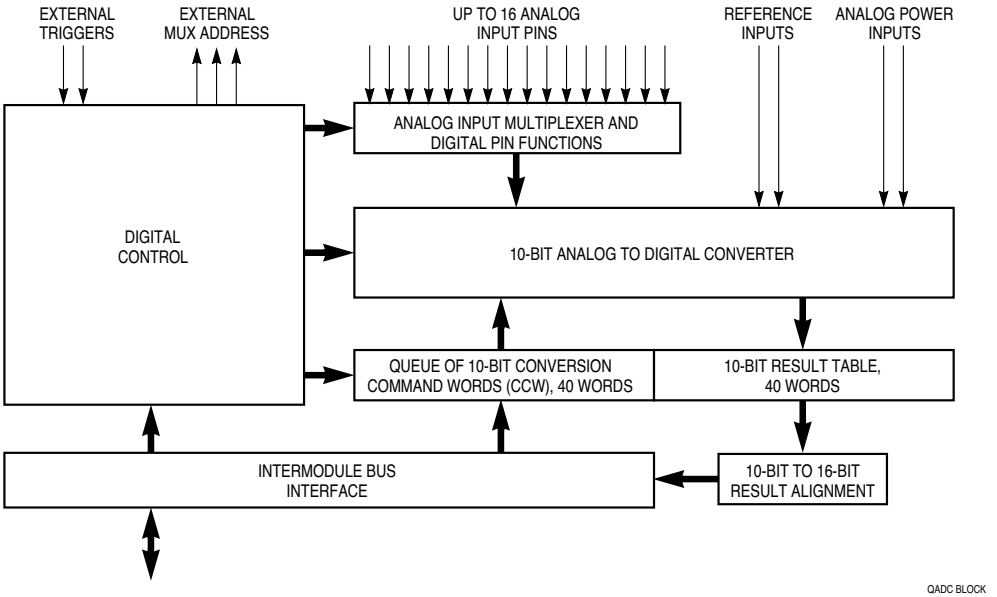

Figure 8-1 QADC Block Diagram

8.12.3.3 Continuous-Scan Modes

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

When a queue is programmed for a continuous-scan mode, the single-scan enable bit in the queue control register does not have any meaning or effect. As soon as the queue operating mode is programmed, the selected trigger event can initiate queue execution.

In the case of the software initiated continuous-scan mode, the trigger event is generated internally and queue execution begins immediately. In the other continuous-scan queue operating modes, the selected trigger event must occur before the queue can start. A trigger overrun is recorded if a trigger event occurs during queue execution in the external trigger continuous-scan mode and the periodic timer continuous-scan mode. When a pause is encountered during a scan, another trigger event is required for queue execution to continue. Software involvement is not required for queue execution to continue from the paused state.

After queue execution is complete, the queue status is shown as idle. Since the continuous-scan queue operating modes allow an entire queue to be scanned multiple times, software involvement is not required for queue execution to continue from the idle state. The next trigger event causes queue execution to begin again, starting with the first CCW in the queue.

NOTE

It may not be possible to guarantee coherent samples when using the continuous-scan queue operating modes since the relationship between any two conversions may be variable due to programmable trigger events and queue priorities.

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

- Software initiated continuous-scan mode
  - When this mode is programmed, the trigger event is generated automatically by the QADC, and queue execution begins immediately. If a pause is encountered, queue execution ceases for two QCLKs, while another trigger event is generated internally; execution then continues. When the end-of-queue is reached, another internal trigger event is generated, and queue execution begins again from the beginning of the queue.
  - While the time to internally generate and act on a trigger event is very short, software can momentarily read the status conditions, indicating that the queue is paused or idle. The trigger overrun flag is never set while in the software initiated continuous-scan mode.

To prepare the QADC for a scan sequence, the desired channel conversions are written to the CCW table. Software establishes the criteria for initiating the queue execution by programming queue operating mode. The queue operating mode determines what type of trigger event initiates queue execution.

A scan sequence may be initiated by the following trigger events:

- A software command
- Expiration of the periodic/interval timer
- An external trigger signal

Software also specifies whether the QADC is to perform a single pass through the queue or is to scan continuously. When a single-scan mode is selected, queue execution begins when software sets the single-scan enable bit. When a continuous-scan mode is selected, the queue remains active in the selected queue operating mode after the QADC completes each queue scan sequence.

During queue execution, the QADC reads each CCW from the active queue and executes conversions in four stages:

1. Initial sample
2. Transfer
3. Final sample
4. Resolution

During initial sample, the selected input channel is connected to the sample capacitor at the input of the sample buffer amplifier.

During the transfer period, the sample capacitor is disconnected from the multiplexer, and the stored voltage is buffered and transferred to the RC DAC array.

During the final sample period, the sample capacitor and amplifier are bypassed, and the multiplexer input charges the RC DAC array directly. Each CCW specifies a final input sample time of 2, 4, 8, or 16 QCLK cycles. When an analog-to-digital conversion is complete, the result is written to the corresponding location in the result word table. The QADC continues to sequentially execute each CCW in the queue until the end of the queue is detected or a pause bit is found in a CCW.

When the pause bit is set in the current CCW, the QADC stops execution of the queue until a new trigger event occurs. The pause status flag bit is set, which may generate an interrupt request to notify software that the queue has reached the pause state. When the next trigger event occurs, the paused state ends, and the QADC continues to execute each CCW in the queue until another pause is encountered or the end of the queue is detected.

An end-of-queue condition is indicated as follows:

- The CCW channel field is programmed with 63 ($3F) to specify the end of the queue.
- The end of queue 1 is implied by the beginning of queue 2, which is specified in the BQ2 field in QACR2.
- The physical end of the queue RAM space defines the end of either queue.

Table A-19 PWMSM Timing Characteristics

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

Num	Parameter	Symbol	Min	Max	Unit
1	PWMSM output resolution[1]	t_{PWMR}	—	—	µs
2	PWMSM output pulse[2]	t_{PWMO}	$2.0/f_{sys}$	—	µs
3	PWMSM output pulse[3]	t_{PWMO}	$2.0/f_{sys}$	$2.0/f_{sys}$	µs
4	CPSM enable to output set PWMSM enabled before CPSM , DIV23 = 0 PWMSM enabled before CPSM , DIV23 = 1	t_{PWMP}	 $3.5/f_{sys}$ $6.5/f_{sys}$	—	µs
5	PWM enable to output set PWMSM enabled before CPSM , DIV23 = 0 PWMSM enabled before CPSM , DIV23 = 1	t_{PWME}	 $3.5/f_{sys}$ $5.5/f_{sys}$	 $4.5/f_{sys}$ $6.5/f_{sys}$	µs
6	FLAG to IMB interrupt request	t_{FIRQ}	$1.5/f_{sys}$	$2.5/f_{sys}$	µs

NOTES:

1. Minimum output resolution depends on counter and prescaler divide ratio selection.
2. Excluding the case where the output is always zero.
3. Excluding the case where the output is always zero.

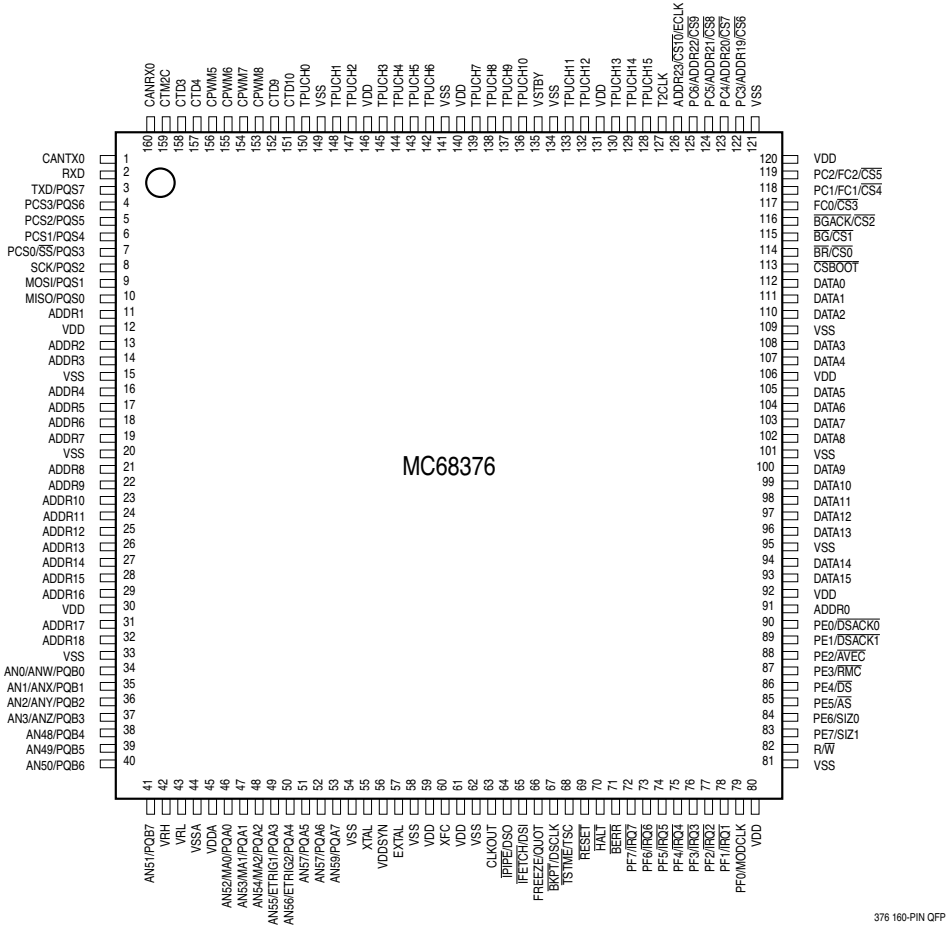

Figure B-2 MC68376 Pin Assignments for 160-Pin Package

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.

IL[2:0] — Interrupt Level

When the FCSM 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].

DRV[A:B] — Drive Time Base Bus

This field controls the connection of the FCSM to time base buses A and B. Refer to **Table D-39**.

Table D-39 Drive Time Base Bus Field

DRVA	DRVB	Bus Selected
0	0	Neither time base bus A nor bus B is driven
0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and bus B are driven

WARNING

Two time base buses should not be driven at the same time.

IN — Clock Input Pin Status

This read-only bit reflects the logic state of the clock input pin CTM2C. Writing to this bit has no effect nor does reset.

CLK[2:0] — Counter Clock Select Field

These read/write control bits select one of the six CPSM clock signals (PCLK[1:6]) or one of two external conditions on CTM2C to clock the free-running counter. The maximum frequency of an external clock signal is $f_{sys}/4$. Refer to **Table D-40**.

Table D-40 Counter Clock Select Field

CLK2	CLK1	CLK0	Free Running Counter Clock Source
0	0	0	Prescaler output 1 (/2 or /3)
0	0	1	Prescaler output 2 (/4 or /6)
0	1	0	Prescaler output 3 (/8 or /12)
0	1	1	Prescaler output 4 (/16 or /24)
1	0	0	Prescaler output 5 (/32 or /48)
1	0	1	Prescaler output 6 (/64 or /512 or /96 to /768)
1	1	0	CTM2C input pin, negative edge
1	1	1	CTM2C input pin, positive edge

CHBK — Channel Register Breakpoint Flag

CHBK is asserted if a breakpoint occurs because of a CHAN register match with the CHAN register breakpoint register. CHBK is negated when the BKPT flag is cleared.

SRBK — Service Request Breakpoint Flag

SRBK is asserted if a breakpoint occurs because of any of the service request latches being asserted along with their corresponding enable flag in the development support control register. SRBK is negated when the BKPT flag is cleared.

TPUF — TPU FREEZE Flag

TPUF is set whenever the TPU is in a halted state as a result of FREEZE being asserted. This flag is automatically negated when the TPU exits the halted state because of FREEZE being negated.

D.8.5 TPU Interrupt Configuration Register

TICR — TPU Interrupt Configuration Register **$YFFE08**

15	10	9	8	7	6	5	4	3	0
NOT USED	CIRL[2:0]			CIBV[3:0]				NOT USED	
RESET:									
	0	0	0	0	0	0	0		

CIRL[2:0] — Channel Interrupt Request Level

This three-bit field specifies the interrupt request level for all channels. Level seven for this field indicates a non-maskable interrupt; level zero indicates that all channel interrupts are disabled.

CIBV[3:0] — Channel Interrupt Base Vector

The TPU is assigned 16 unique interrupt vector numbers, one vector number for each channel. The CIBV field specifies the most significant nibble of all 16 TPU channel interrupt vector numbers. The lower nibble of the TPU interrupt vector number is determined by the channel number on which the interrupt occurs.

D.8.6 Channel Interrupt Enable Register

CIER — Channel Interrupt Enable Register **$YFFE0A**

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 Enable/Disable

0 = Channel interrupts disabled

1 = Channel interrupts enabled

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: