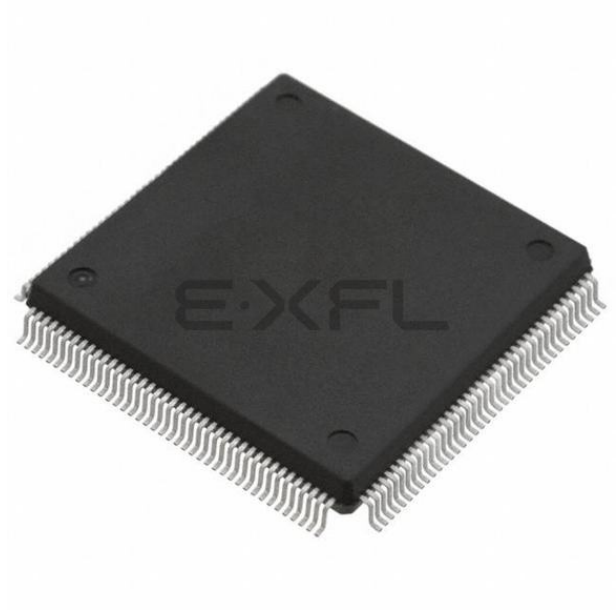


Welcome to E-XFL.COM

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/mc68376bamab20

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
(Continued)

SECTION 2 NOMENCLATURE

The following nomenclature is used throughout the manual. Nomenclature used only in certain sections, such as register bit mnemonics, is defined in those sections.

2.1 Symbols and Operators

+	—	Addition
–	—	Subtraction or negation (two's complement)
*	—	Multiplication
/	—	Division
>	—	Greater
<	—	Less
=	—	Equal
≥	—	Equal or greater
≤	—	Equal or less
≠	—	Not equal
•	—	AND
✣	—	Inclusive OR (OR)
⊕	—	Exclusive OR (EOR)
$\overline{\text{NOT}}$	—	Complementation
:	—	Concatenation
⇒	—	Transferred
⇔	—	Exchanged
±	—	Sign bit; also used to show tolerance
«	—	Sign extension
%	—	Binary value
$	—	Hexadecimal value

SECTION 3 OVERVIEW

This section contains information about the entire MC68336/376 modular microcontroller. It lists the features of each module, shows device functional divisions and pin assignments, summarizes signal and pin functions, discusses the intermodule bus, and provides system memory maps. Timing and electrical specifications for the entire microcontroller and for individual modules are provided in **APPENDIX A ELECTRICAL CHARACTERISTICS**. Comprehensive module register descriptions and memory maps are provided in **APPENDIX D REGISTER SUMMARY**.

3.1 MCU Features

The following paragraphs highlight capabilities of each of the microcontroller modules. Each module is discussed separately in a subsequent section of this user's manual.

3.1.1 Central Processing Unit (CPU32)

- 32-bit architecture
- Virtual memory implementation
- Table look-up and interpolate instruction
- Improved exception handling for controller applications
- High level language support
- Background debug mode
- Fully static operation

3.1.2 System Integration Module (SIM)

- External bus support
- Programmable chip select outputs
- System protection logic
- Watchdog timer, clock monitor and bus monitor
- Two 8-bit dual function input/output ports
- One 7-bit dual function output port
- Phase-locked loop (PLL) clock system

3.1.3 Standby RAM Module (SRAM)

- 4-Kbytes of static RAM
- No standby supply

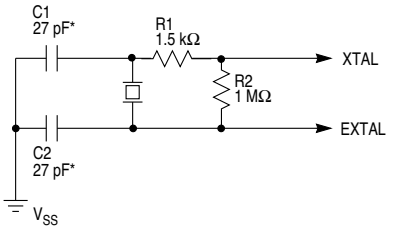
3.1.4 Masked ROM Module (MRM)

- 8-Kbyte array, accessible as bytes or words
- User selectable default base address
- User selectable bootstrap ROM function
- User selectable ROM verification code

Table 3-1 MC68336/376 Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/$\overline{CS10}$/ECLK	A	Yes	No	O	—
ADDR[22:19]/$\overline{CS[9:6]}$	A	Yes	No	O	PC[6:3]
ADDR[18:0]	A	Yes	No	—	—
AN[51:48]	—	Yes[1]	Yes	I	PQB[7:4]
AN[3:0]/AN[w, x, y, z]	—	Yes[1]	Yes	I	PQB[3:0]
AN[59:57]	Ba	Yes	Yes	I/O	PQA[7:5]
AN[56:55]/ETRIG[2:1]	Ba	Yes	Yes	I/O	PQA[4:3]
AN[54:52]/MA[2:0]	Ba	Yes	Yes	I/O	PQA[2:0]
$\overline{AS}$	B	Yes	Yes	I/O	PE5
$\overline{AVEC}$	B	Yes	No	I/O	PE2
$\overline{BERR}$	B	Yes	No	—	—
$\overline{BG}$/$\overline{CS1}$	B	—	—	—	—
$\overline{BGACK}$/$\overline{CS2}$	B	Yes	No	—	—
$\overline{BKPT}$/DSCLK	—	Yes	Yes	—	—
$\overline{BR}$/$\overline{CS0}$	B	Yes	No	O	—
CLKOUT	A	—	—	—	—
CANRX0 (MC68376 Only)	—	Yes	Yes	—	—
CANTX0 (MC68376 Only)	Bo	—	—	—	—
$\overline{CSBOOT}$	B	—	—	—	—
CTD[10:9]/[4:3]	A	Yes	Yes	I/O	—
CPWM[8:5]	A	—	—	O	—
CTM2C	—	Yes	Yes	I	—
DATA[15:0]	Aw	Yes[1]	No	—	—
$\overline{DS}$	B	Yes	Yes	I/O	PE4
$\overline{DSACK[1:0]}$	B	Yes	No	I/O	PE[1:0]
EXTAL[2]	—	—	Special	—	—
FC[2:0]/$\overline{CS[5:3]}$	A	Yes	No	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
$\overline{IPIPE}$/DSO	A	—	—	O	—
$\overline{IFETCH}$/DSI	A	Yes	Yes	—	—
$\overline{HALT}$	Bo	Yes	No	—	—
$\overline{IRQ[7:1]}$	B	Yes	Yes	I/O	PF[7:1]
MISO	Bo	Yes[1]	Yes	I/O	PQS0
MODCLK	B	Yes[1]	Yes	I/O	PF0
MOSI	Bo	Yes[1]	Yes	I/O	PQS1
PCS0/$\overline{SS}$	Bo	Yes[1]	Yes	I/O	PQS3
PCS[3:1]	Bo	Yes[1]	Yes	I/O	PQS[6:4]
R/$\overline{W}$	A	Yes	No	—	—
$\overline{RESET}$	Bo	Yes	Yes	—	—
$\overline{RMC}$	B	Yes	Yes	I/O	PE3
RXD	—	No	Yes	—	—
SCK	Bo	Yes[1]	Yes	I/O	PQS2

To generate a reference frequency using the crystal oscillator, a reference crystal must be connected between the EXTAL and XTAL pins. Typically, a 4.194 MHz crystal is used, but the frequency may vary between 1 and 6 MHz. **Figure 5-3** shows a typical circuit.



* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A KDS041-18 4.194 MHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

32 OSCILLATOR 4M

Figure 5-3 System Clock Oscillator Circuit

If a fast reference frequency is provided to the PLL from a source other than a crystal, or an external system clock signal is applied through the EXTAL pin, the XTAL pin must be left floating.

When an external system clock signal is applied (MODCLK = 0 during reset), the PLL is disabled. The duty cycle of this signal is critical, especially at operating frequencies close to maximum. The relationship between clock signal duty cycle and clock signal period is expressed as follows:

$$\text{Minimum External Clock Period} = \frac{\text{Minimum External Clock High/Low Time}}{50\% - \text{Percentage Variation of External Clock Input Duty Cycle}}$$

5.3.2 Clock Synthesizer Operation

V_{DDSYN} is used to power the clock circuits when the system clock is synthesized from either a crystal or an externally supplied reference frequency. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. A quiet power supply must be used as the V_{DDSYN} source. Adequate external bypass capacitors should be placed as close as possible to the V_{DDSYN} pin to assure a stable operating frequency. When an external system clock signal is applied and the PLL is disabled, V_{DDSYN} should be connected to the V_{DD} supply. Refer to the *SIM Reference Manual* (SIMRM/AD) for more information regarding system clock power supply conditioning.

Table 5-3 System Frequencies from 4.194 MHz Reference (Continued)

Modulus	Prescaler			
Y	[W:X] = 00	[W:X] = 01	[W:X] = 10	[W:X] = 11
100000	4325 kHz	8651 kHz	17302 kHz	34603 kHz
100001	4456	8913	17826	35652
100010	4588	9175	18350	36700
100011	4719	9437	18874	37749
100100	4850	9699	19399	38797
100101	4981	9961	19923	39846
100110	5112	10224	20447	40894
100111	5243	10486	20972	41943
101000	5374	10748	21496	42992
101001	5505	11010	22020	44040
101010	5636	11272	22544	45089
101011	5767	11534	23069	46137
101100	5898	11796	23593	47186
101101	6029	12059	24117	48234
101110	6160	12321	24642	49283
101111	6291	12583	25166	50332
110000	6423	12845	25690	51380
110001	6554	13107	26214	52428
110010	6685	13369	26739	53477
110011	6816	13631	27263	54526
110100	6947	13894	27787	55575
110101	7078	14156	28312	56623
110110	7209	14418	28836	57672
110111	7340	14680	29360	58720
111000	7471	14942	2988	59769
111001	7602	15204	30409	60817
111010	7733	15466	30933	61866
111011	7864	15729	31457	62915
111100	7995	15991	31982	63963
111101	8126	16253	32506	65011
111110	8258	16515	33030	66060
111111	8389	16777	33554	67109

5.4.8 Low-Power STOP Mode Operation

When the CPU32 executes the LPSTOP instruction, the current interrupt priority mask is stored in the clock control logic, internal clocks are disabled according to the state of the STSIM bit in the SYNCR, and the MCU enters low-power stop mode. The bus monitor, halt monitor, and spurious interrupt monitor are all inactive during low-power stop mode.

During low-power stop mode, the clock input to the software watchdog timer is disabled and the timer stops. The software watchdog begins to run again on the first rising clock edge after low-power stop mode ends. The watchdog is not reset by low-power stop mode. A service sequence must be performed to reset the timer.

The periodic interrupt timer does not respond to the LPSTOP instruction, but continues to run during LPSTOP. To stop the periodic interrupt timer, PITR must be loaded with a zero value before the LPSTOP instruction is executed. A PIT interrupt, or an external interrupt request, can bring the MCU out of low-power stop mode if it has a higher priority than the interrupt mask value stored in the clock control logic when low-power stop mode is initiated. LPSTOP can be terminated by a reset.

5.5 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. **Figure 5-8** shows a basic system with external memory and peripherals.

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.

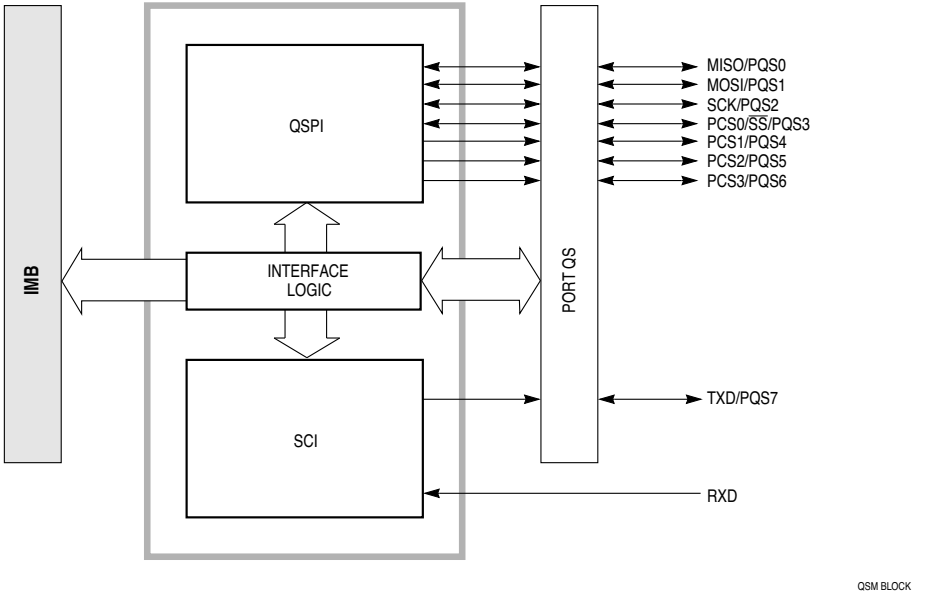


Figure 9-1 QSM Block Diagram

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.

The SCI provides a standard non-return to zero (NRZ) mark/space format. It operates in either full- or half-duplex mode. There are separate transmitter and receiver enable bits and dual data buffers. A modulus-type baud rate generator provides rates from 110 baud to 655 kbaud with a 20.97 MHz system clock. Word length of either eight or nine bits is software selectable. Optional parity generation and detection provide either even or odd parity check capability. Advanced error detection circuitry catches glitches of up to 1/16 of a bit time in duration. Wake-up functions allow the CPU32 to run uninterrupted until meaningful data is available.

9.2 QSM Registers and Address Map

There are four types of QSM registers: QSM global registers, QSM pin control registers, QSPI registers, and SCI registers. Refer to **9.2.1 QSM Global Registers** and **9.2.2 QSM Pin Control Registers** for a discussion of global and pin control registers. Refer to **9.3.1 QSPI Registers** and **9.4.1 SCI Registers** for further information about QSPI and SCI registers. Writes to unimplemented register bits have no effect, and reads of unimplemented bits always return zero.

The QSM address map includes the QSM registers and the QSPI RAM. The MM bit in the system integration module configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each module.

Refer to **D.6 Queued Serial Module** for a QSM address map and register bit and field definitions. **5.2.1 Module Mapping** contains more information about how the state of MM affects the system.

9.2.1 QSM Global Registers

The QSM configuration register (QSMCR) contains parameters for interfacing to the CPU32 and the intermodule bus. The QSM test register (QTEST) is used during factory test of the QSM. The QSM interrupt level register (QILR) determines the priority of interrupts requested by the QSM and the vector used when an interrupt is acknowledged. The QSM interrupt vector register (QIVR) contains the interrupt vector for both QSM submodules. QILR and QIVR are 8-bit registers located at the same word address.

9.2.1.1 Low-Power Stop Operation

When the STOP bit in QSMCR is set, the system clock input to the QSM is disabled and the module enters a low-power operating state. QSMCR is the only register guaranteed to be readable while STOP is asserted. The QSPI RAM is not readable during LPSTOP. However, writes to RAM or any register are guaranteed valid while STOP is asserted. STOP can be set by the CPU32 and by reset.

System software must bring the QSPI and SCI to an orderly stop before asserting STOP to avoid data corruption. The IRQ mask level in the CPU32 status register should be set to a higher value than the IRQ level generated by the QSM module. The SCI receiver and transmitter should be disabled after transfers in progress are complete. The QSPI can be halted by setting the HALT bit in SPCR3 and then setting STOP after the HALTA flag is set. The IRQ mask in the CPU status register should be restored to its former level. Refer to **5.3.4 Low-Power Operation** for more information about low-power stop mode.

9.4.3.6 Receiver Operation

The RE bit in SCCR1 enables (RE = 1) and disables (RE = 0) the receiver. The receiver contains a receive serial shifter and a parallel receive data register (RDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU32. The receiver is double-buffered, allowing data to be held in the RDR while other data is shifted in.

Receiver bit processor logic drives a state machine that determines the logic level for each bit-time. This state machine controls when the bit processor logic is to sample the RXD pin and also controls when data is to be passed to the receive serial shifter. A receive time clock is used to control sampling and synchronization. Data is shifted into the receive serial shifter according to the most recent synchronization of the receive time clock with the incoming data stream. From this point on, data movement is synchronized with the MCU system clock. Operation of the receiver state machine is detailed in the *QSM Reference Manual* (QSMRM/AD).

The number of bits shifted in by the receiver depends on the serial format. However, all frames must end with at least one stop bit. When the stop bit is received, the frame is considered to be complete, and the received data in the serial shifter is transferred to the RDR. The receiver data register flag (RDRF) is set when the data is transferred.

Noise errors, parity errors, and framing errors can be detected while a data stream is being received. Although error conditions are detected as bits are received, the noise flag (NF), the parity flag (PF), and the framing error (FE) flag in SCSR are not set until data is transferred from the serial shifter to the RDR.

RDRF must be cleared before the next transfer from the shifter can take place. If RDRF is set when the shifter is full, transfers are inhibited and the overrun error (OR) flag in SCSR is set. OR indicates that the RDR needs to be serviced faster. When OR is set, the data in the RDR is preserved, but the data in the serial shifter is lost. Because framing, noise, and parity errors are detected while data is in the serial shifter, FE, NF, and PF cannot occur at the same time as OR.

When the CPU32 reads SCSR and SCDR in sequence, it acquires status and data, and also clears the status flags. Reading SCSR acquires status and arms the clearing mechanism. Reading SCDR acquires data and clears SCSR.

When RIE in SCCR1 is set, an interrupt request is generated whenever RDRF is set. Because receiver status flags are set at the same time as RDRF, they do not have separate interrupt enables.

9.4.3.7 Idle-Line Detection

During a typical serial transmission, frames are transmitted isochronally and no idle time occurs between frames. Even when all the data bits in a frame are logic ones, the start bit provides one logic zero bit-time during the frame. An idle line is a sequence of contiguous ones equal to the current frame size. Frame size is determined by the state of the M bit in SCCR1.

The SCI receiver has both short and long idle-line detection capability. Idle-line detection is always enabled. The idle line type (ILT) bit in SCCR1 determines which type of detection is used. When an idle line condition is detected, the IDLE flag in SCSR is set.

For short idle-line detection, the receiver bit processor counts contiguous logic one bit-times whenever they occur. Short detection provides the earliest possible recognition of an idle line condition, because the stop bit and contiguous logic ones before and after it are counted. For long idle-line detection, the receiver counts logic ones after the stop bit is received. Only a complete idle frame causes the IDLE flag to be set.

In some applications, software overhead can cause a bit-time of logic level one to occur between frames. This bit-time does not affect content, but if it occurs after a frame of ones when short detection is enabled, the receiver flags an idle line.

When the ILIE bit in SCCR1 is set, an interrupt request is generated when the IDLE flag is set. The flag is cleared by reading SCSR and SCDR in sequence. IDLE is not set again until after at least one frame has been received (RDRF = 1). This prevents an extended idle interval from causing more than one interrupt.

9.4.3.8 Receiver Wake-Up

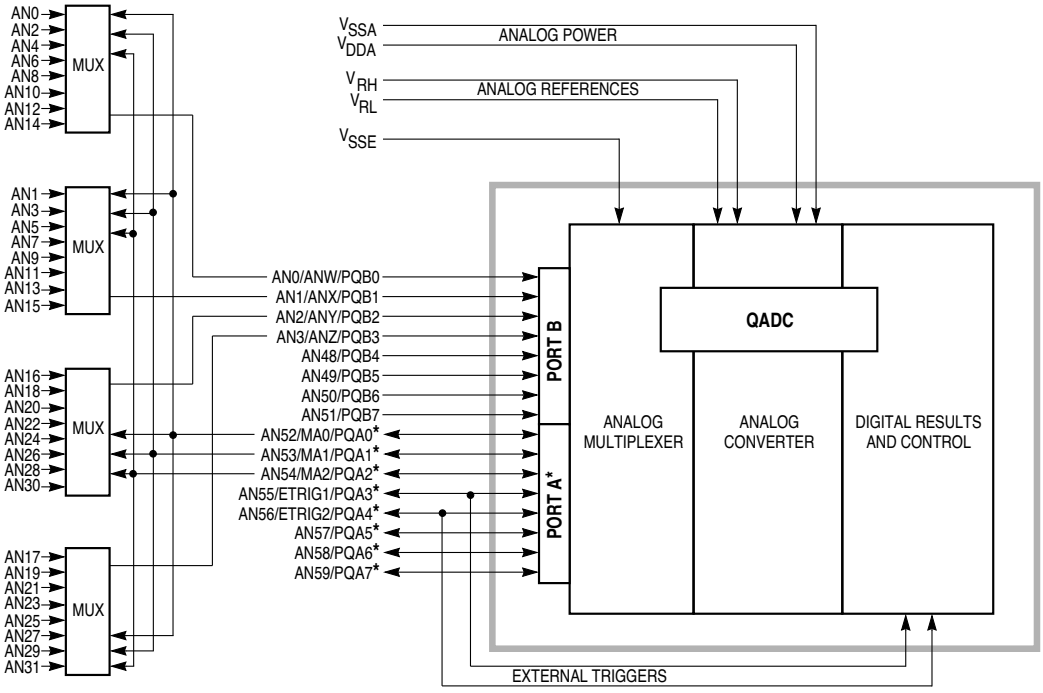
The receiver wake-up function allows a transmitting device to direct a transmission to a single receiver or to a group of receivers by sending an address frame at the start of a message. Hardware activates each receiver in a system under certain conditions. Resident software must process address information and enable or disable receiver operation.

A receiver is placed in wake-up mode by setting the RWU bit in SCCR1. While RWU is set, receiver status flags and interrupts are disabled. Although the CPU32 can clear RWU, it is normally cleared by hardware during wake-up.

The WAKE bit in SCCR1 determines which type of wake-up is used. When WAKE = 0, idle-line wake-up is selected. When WAKE = 1, address-mark wake-up is selected. Both types require a software-based device addressing and recognition scheme.

Idle-line wake-up allows a receiver to sleep until an idle line is detected. When an idle-line is detected, the receiver clears RWU and wakes up. The receiver waits for the first frame of the next transmission. The byte is received normally, transferred to the RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. For idle-line wake-up to work, there must be a minimum of one frame of idle line between transmissions. There must be no idle time between frames within a transmission.

Address-mark wake-up uses a special frame format to wake up the receiver. When the MSB of an address-mark frame is set, that frame contains address information. The first frame of each transmission must be an address frame. When the MSB of a frame is set, the receiver clears RWU and wakes up. The byte is received normally, transferred to the RDR, and the RDRF flag is set. If software does not recognize the address, it can set RWU and put the receiver back to sleep. Address-mark wake-up allows idle time between frames and eliminates idle time between transmissions. However, there is a loss of efficiency because of an additional bit-time per frame.



* PORT A PINS INCORPORATE OPEN DRAIN PULL DOWN DRIVERS.

QADC EXT MUX CONN

Figure 8-3 Example of External Multiplexing

When the external multiplexed mode is selected, the QADC automatically creates the MA[2:0] open drain output signals from the channel number in each CCW. The QADC also converts the proper input channel (ANw, ANx, ANy, and ANz) by interpreting the CCW channel number. As a result, up to 32 externally multiplexed channels appear to the conversion queues as directly connected signals. Software simply puts the channel number of an externally multiplexed channel into a CCW.

Figure 8-3 shows that MA[2:0] may also be analog or digital input pins. When external multiplexing is selected, none of the MA[2:0] pins can be used for analog or digital inputs. They become multiplexed address outputs.

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

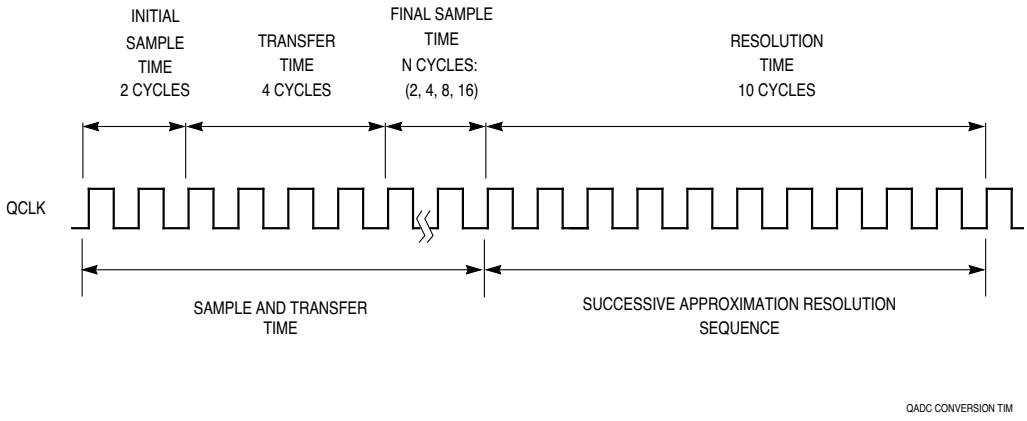


Figure 8-5 Conversion Timing

8.11.1.1 Amplifier Bypass Mode Conversion Timing

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

Multiple PWMs generated by MCPWM have two types of high time alignment: edge aligned and center aligned. Edge aligned mode uses n + 1 TPU channels for n PWMs; center aligned mode uses 2n + 1 channels. Center aligned mode allows a user-defined “dead time” to be specified so that two PWMs can be used to drive an H-bridge without destructive current spikes. This feature is important for motor control applications.

Refer to TPU programming note *Multichannel Pulse-Width Modulation (MCPWM) TPU Function* (TPUPN05/D) for more information.

11.5.6 Fast Quadrature Decode (FQD)

FQD is a position feedback function for motor control. It decodes the two signals from a slotted encoder to provide the CPU32 with a 16-bit free running position counter. FQD incorporates a “speed switch” which disables one of the channels at high speed, allowing faster signals to be decoded. A time stamp is provided on every counter update to allow position interpolation and better velocity determination at low speed or when low resolution encoders are used. The third index channel provided by some encoders is handled by the NITC function.

Refer to TPU programming note *Fast Quadrature Decode (FQD) TPU Function* (TPUPN02/D) for more information.

11.5.7 Universal Asynchronous Receiver/Transmitter (UART)

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

Refer to TPU programming note *Universal Asynchronous Receiver/Transmitter (UART) TPU Function* (TPUPN07/D) for more information.

11.5.8 Brushless Motor Commutation (COMM)

This function generates the phase commutation signals for a variety of brushless motors, including three-phase brushless DC motors. It derives the commutation state directly from the position decoded in FQD, thus eliminating the need for hall effect sensors.

The state sequence is implemented as a user-configurable state machine, thus providing a flexible approach with other general applications. An offset parameter is provided to allow all the switching angles to be advanced or retarded on the fly by the CPU32. This feature is useful for torque maintenance at high speeds.

Refer to TPU programming note *Brushless Motor Commutation (COMM) TPU Function* (TPUPN09/D) for more information.

13.2 External Pins

The TouCAN module interface to the CAN bus is composed of four pins: CANTX0 and CANTX1, which transmit serial data, and CANRX0 and CANRX1, which receive serial data. **Figure 13-2** shows a typical CAN system.

NOTE

Pins CANTX1 and CANRX1 are not used on the MC68376.

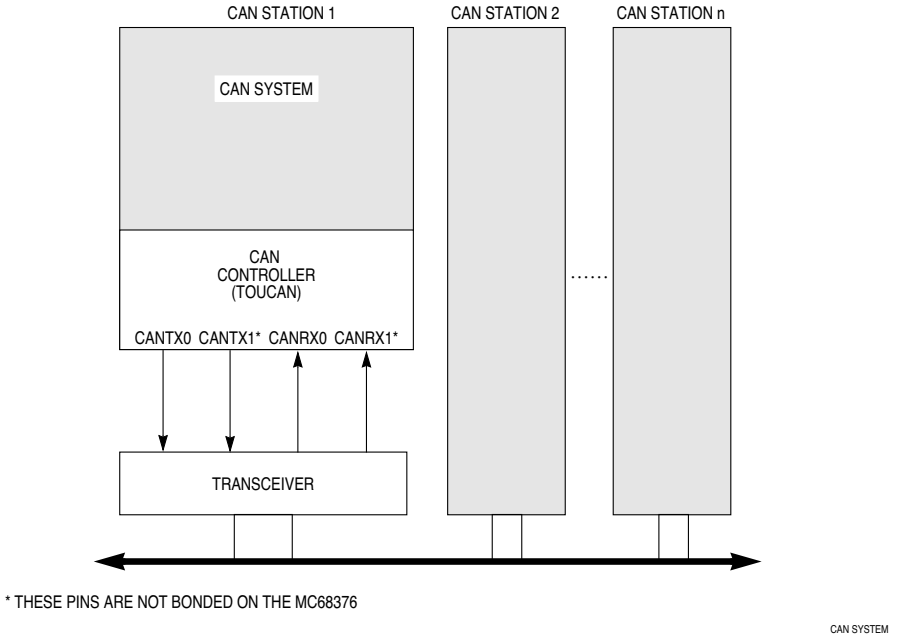


Figure 13-2 Typical CAN Network

Each CAN station is connected physically to the CAN bus through a transceiver. The transceiver provides the transmit drive, waveshaping, and receive/compare functions required for communicating on the CAN bus. It can also provide protection against damage to the TouCAN caused by a defective CAN bus or a defective CAN station.

13.3 Programmer's Model

The TouCAN module address space is split into 128 bytes starting at the base address, and then an extra 256 bytes starting at the base address +128. The upper 256 are fully used for the message buffer structures. Out of the lower 128 bytes, only part is occupied by various registers. Refer to **D.10 TouCAN Module** for detailed information on the TouCAN address map and register structure.

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.

Table D-26 Queue 2 Operating Modes

MQ2[4:0]	Queue 2 Operating Mode
00000	Disabled mode, conversions do not occur
00001	Software triggered single-scan mode (started with SSE2)
00010	External trigger rising edge single-scan mode (on ETRIG2 pin)
00011	External trigger falling edge single-scan mode (on ETRIG2 pin)
00100	Interval timer single-scan mode: interval = QCLK period x 2^7
00101	Interval timer single-scan mode: interval = QCLK period x 2^8
00110	Interval timer single-scan mode: interval = QCLK period x 2^9
00111	Interval timer single-scan mode: interval = QCLK period x 2^{10}
01000	Interval timer single-scan mode: interval = QCLK period x 2^{11}
01001	Interval timer single-scan mode: interval = QCLK period x 2^{12}
01010	Interval timer single-scan mode: interval = QCLK period x 2^{13}
01011	Interval timer single-scan mode: interval = QCLK period x 2^{14}
01100	Interval timer single-scan mode: interval = QCLK period x 2^{15}
01101	Interval timer single-scan mode: interval = QCLK period x 2^{16}
01110	Interval timer single-scan mode: interval = QCLK period x 2^{17}
01111	Reserved mode
10000	Reserved mode
10001	Software triggered continuous-scan mode (started with SSE2)
10010	External trigger rising edge continuous-scan mode (on ETRIG2 pin)
10011	External trigger falling edge continuous-scan mode (on ETRIG2 pin)
10100	Periodic timer continuous-scan mode: period = QCLK period x 2^7
10101	Periodic timer continuous-scan mode: period = QCLK period x 2^8
10110	Periodic timer continuous-scan mode: period = QCLK period x 2^9
10111	Periodic timer continuous-scan mode: period = QCLK period x 2^{10}
11000	Periodic timer continuous-scan mode: period = QCLK period x 2^{11}
11001	Periodic timer continuous-scan mode: period = QCLK period x 2^{12}
11010	Periodic timer continuous-scan mode: period = QCLK period x 2^{13}
11011	Periodic timer continuous-scan mode: period = QCLK period x 2^{14}
11100	Periodic timer continuous-scan mode: period = QCLK period x 2^{15}
11101	Periodic timer continuous-scan mode: period = QCLK period x 2^{16}
11110	Periodic timer continuous-scan mode: period = QCLK period x 2^{17}
11111	Reserved mode

RES — Queue 2 Resume

RES selects the resumption point after queue 2 is suspended by queue 1. If RES is changed during execution of queue 2, the change is not recognized until an end-of-queue condition is reached, or the queue operating mode of queue 2 is changed.

0 = After suspension, begin execution with the first CCW in queue 2 or the current subqueue.

1 = After suspension, begin execution with the aborted CCW in queue 2.

D.7.7 FCSM Counter Register

FCSMCNT — FCSM Counter Register **$YFF462**

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

The FCSM counter register is a read/write register. A read returns the current value of the counter. A write loads the counter with the specified value. The counter then begins incrementing from this new value.

D.7.8 MCSM Status/Interrupt/Control Registers

MCSM2SIC — MCSM2 Status/Interrupt/Control Register **$YFF410**
MCSM11SIC — MCSM11 Status/Interrupt/Control Register **$YFF458**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COF	IL[2:0]			IARB3	NOT USED	DRVA	DRVB	IN2	IN1	EDGEN	EDGEP	NOT USED	CLK[2:0]		
RESET:															
U	0	0	0	0	0	0	0	U	U	0	0	0	0	0	0

COF — Counter Overflow Flag

This bit indicates whether or not a counter overflow has occurred. An overflow of the MCSM counter is defined as the transition of the counter from $FFFF to $xxxx, where $xxxx is the value contained in the modulus latch. If the IL[2:0] field is non-zero, an interrupt request is generated when the COF bit is set.

0 = Counter overflow has not occurred
1 = Counter overflow has occurred

This flag bit is set only by hardware and cleared only by software or by system reset. To clear the flag, first read the bit as a one, then write a zero to the bit.

IL[2:0] — Interrupt Level Field

When the MCSM 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 MCSM to time base buses A and B. Refer to **Table D-41**.

To clear an interrupt flag, first read the flag as a one, and then write it as a zero. Should a new flag setting event occur between the time that the CPU32 reads the flag as a one and writes the flag as a zero, the flag will not be cleared. This register can be written to zeros only.

D.10.15 Error Counters

RXECTR — Receive Error Counter **$YFF0A6**
TXECTR — Transmit Error Counter **$YFF0A7**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXECTR								TXECTR							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Both counters are read only, except when the TouCAN is in test or debug mode.