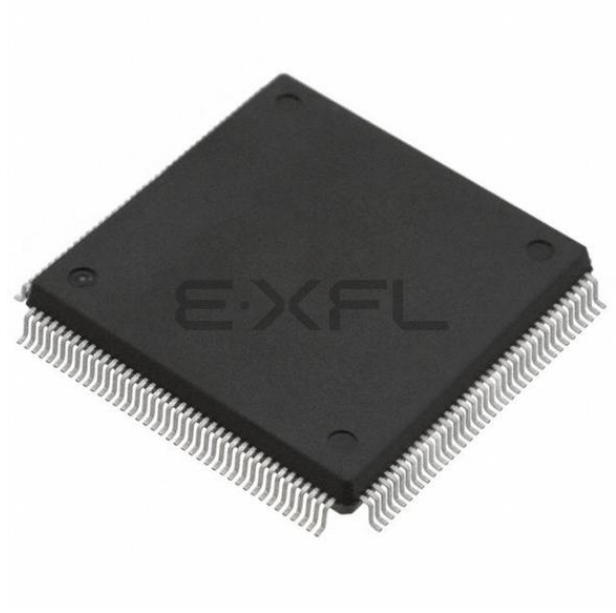


Welcome to E-XFL.COM

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	25MHz
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/mc68336gvab25

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

SECTION 11 TIME PROCESSOR UNIT

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	—

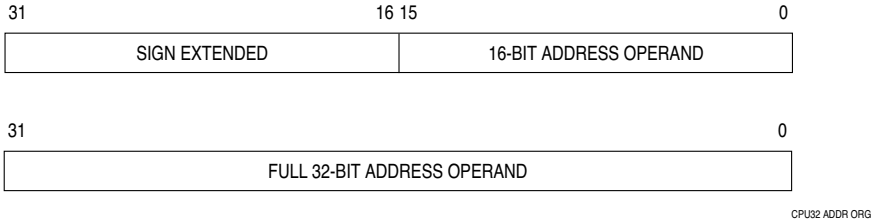


Figure 4-5 Address Organization in Address Registers

4.2.3 Program Counter

The PC contains the address of the next instruction to be executed by the CPU32. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC as appropriate.

4.2.4 Control Registers

The control registers described in this section contain control information for supervisor functions and vary in size. With the exception of the condition code register (the user portion of the status register), they are accessed only by instructions at the supervisor privilege level.

4.2.4.1 Status Register

The status register (SR) stores the processor status. It contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The condition codes are extend (X), negative (N), zero (Z), overflow (V), and carry (C). The user (low-order) byte containing the condition codes is the only portion of the SR information available at the user privilege level; it is referenced as the condition code register (CCR) in user programs.

At the supervisor privilege level, software can access the full status register. The upper byte of this register includes the interrupt priority (IP) mask (three bits), two bits for placing the processor in one of two tracing modes or disabling tracing, and the supervisor/user bit for placing the processor at the desired privilege level.

Undefined bits in the status register are reserved by Motorola for future definition. The undefined bits are read as zeros and should be written as zeros for future compatibility.

All operations to the SR and CCR are word-size operations, but for all CCR operations, the upper byte is read as all zeros and is ignored when written, regardless of privilege level.

Refer to **D.1.2 Status Register** for bit/field definitions and a diagram of the status register.

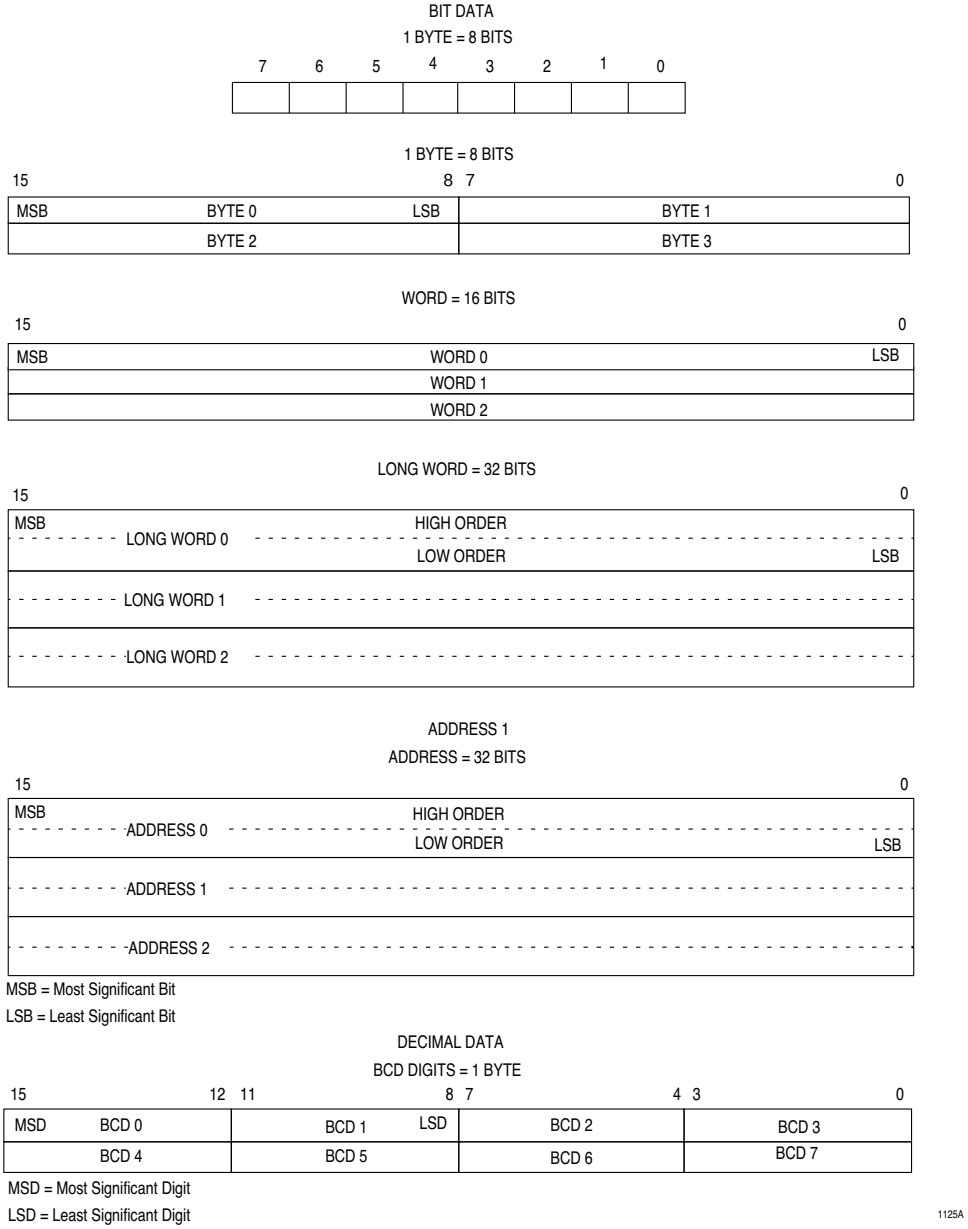
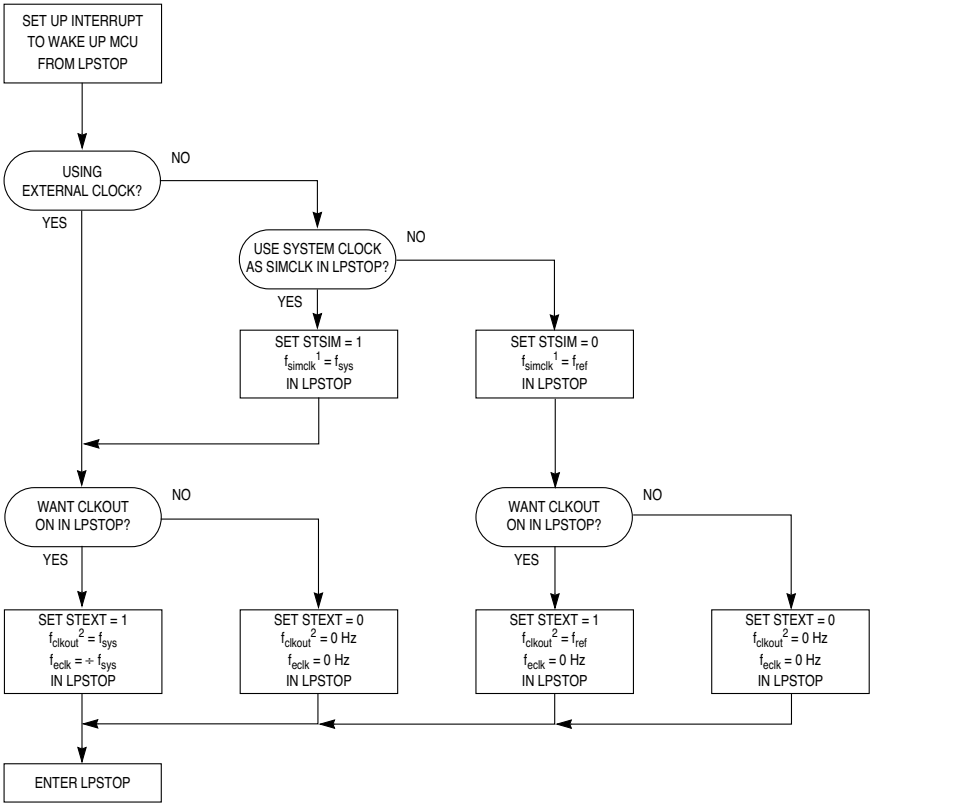


Figure 4-6 Memory Operand Addressing



NOTES:
1. THE SIMCLK IS USED BY THE PIT, $\overline{IRQ}$, AND INPUT BLOCKS OF THE SIM.
2. CLKOUT CONTROL DURING LPSTOP IS OVERRIDDEN BY THE EXOFF BIT IN SIMCR. IF EXOFF = 1, THE CLKOUT PIN IS ALWAYS IN A HIGH IMPEDANCE STATE AND STEXT HAS NO EFFECT IN LPSTOP. IF EXOFF = 0, CLKOUT IS CONTROLLED BY STEXT IN LPSTOP.

LPSTOPFLOW

Figure 5-5 LPSTOP Flowchart

Serial transfers of eight to sixteen can be specified. Programmable transfer length simplifies interfacing to devices that require different data lengths.

An inter-transfer delay of 17 to 8192 system clocks can be specified (default is 17 system clocks). Programmable delay simplifies the interface to devices that require different delays between transfers.

A dedicated 80-byte RAM is used to store received data, data to be transmitted, and a queue of commands. The CPU32 can access these locations directly. This allows serial peripherals to be treated like memory-mapped parallel devices.

The command queue allows the QSPI to perform up to 16 serial transfers without CPU32 intervention. Each queue entry contains all the information needed by the QSPI to independently complete one serial transfer.

A pointer identifies the queue location containing the data and command for the next serial transfer. Normally, the pointer address is incremented after each serial transfer, but the CPU32 can change the pointer value at any time. Support of multiple-tasks can be provided by segmenting the queue.

The QSPI has four peripheral chip-select pins. The chip-select signals simplify interfacing by reducing CPU32 intervention. If the chip-select signals are externally decoded, 16 independent select signals can be generated.

Wrap-around mode allows continuous execution of queued commands. In wrap-around mode, newly received data replaces previously received data in the receive RAM. Wrap-around mode can simplify the interface with A/D converters by continuously updating conversion values stored in the RAM.

Continuous transfer mode allows transfer of an uninterrupted bit stream. Any number of bits in a range from 8 to 256 can be transferred without CPU32 intervention. Longer transfers are possible, but minimal intervention is required to prevent loss of data. A standard delay of 17 system clocks is inserted between the transfer of each queue entry.

9.3.1 QSPI Registers

The programmer's model for the QSPI consists of the QSM global and pin control registers, four QSPI control registers (SPCR[0:3]), the status register (SPSR), and the 80-byte QSPI RAM. Registers and RAM can be read and written by the CPU32. Refer to **D.6 Queued Serial Module** for register bit and field definitions.

9.3.1.1 Control Registers

Control registers contain parameters for configuring the QSPI and enabling various modes of operation. The CPU32 has read and write access to all control registers. The QSM has read access only to all bits except the SPE bit in SPCR1. Control registers must be initialized before the QSPI is enabled to insure defined operation. SPCR1 must be written last because it contains the QSPI enable bit (SPE).

Writing a new value to any control register except SPCR2 while the QSPI is enabled disrupts operation. SPCR2 is buffered. New SPCR2 values become effective after completion of the current serial transfer. Rewriting NEWQP in SPCR2 causes execu-

9.4.1.2 Status Register

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by reading SCSR, then reading or writing SCDR. A long-word read can consecutively access both SCSR and SCDR. This action clears receiver status flag bits that were set at the time of the read, but does not clear TDRE or TC flags.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before reading or writing SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set, and SCDR must be read or written before the status bit is cleared.

Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

9.4.1.3 Data Register

SCDR contains two data registers at the same address. The receive data register (RDR) is a read-only register that contains data received by the SCI serial interface. Data enters the receive serial shifter and is transferred to RDR. The transmit data register (TDR) is a write-only register that contains data to be transmitted. Data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When it is configured for 8-bit operation, they have no meaning or effect.

9.4.2 SCI Pins

Two unidirectional pins, TXD (transmit data) and RXD (receive data), are associated with the SCI. TXD can be used by the SCI or for general-purpose I/O. Function is assigned by the port QS pin assignment register (PQSPAR). The receive data (RXD) pin is dedicated to the SCI. **Table 9-4** shows SCI pin function.

Table 9-4 SCI Pins

Pin Names	Mnemonics	Mode	Function
Receive Data	RXD	Receiver disabled Receiver enabled	Not used Serial data input to SCI
Transmit Data	TXD	Transmitter disabled Transmitter enabled	General-purpose I/O Serial data output from SCI

9.4.3 SCI Operation

SCI operation can be polled by means of status flags in SCSR, or interrupt-driven operation can be employed by the interrupt enable bits in SCCR1.

8.4.1.1 Port A Analog Input Pins

When used as analog inputs, the eight port A pins are referred to as AN[59:52]. Due to the digital output drivers associated with port A, the analog characteristics of port A are different from those of port B. All of the analog signal input pins may be used for at least one other purpose.

8.4.1.2 Port A Digital Input/Output Pins

Port A pins are referred to as PQA[7:0] when used as a bidirectional 8-bit digital input/output port. These eight pins may be used for general-purpose digital input signals or digital open drain pull-down output signals.

Port A pins are connected to a digital input synchronizer during reads and may be used as general purpose digital inputs.

Each port A pin is configured as an input or output by programming the port data direction register (DDRQA). Digital input signal states are read from the PORTQA data register when DDRQA specifies that the pins are inputs. Digital data in PORTQA is driven onto the port A pins when the corresponding bits in DDRQA specify outputs. Refer to **D.5.5 Port Data Direction Register** for more information. Since the outputs are open drain drivers (so as to minimize the effects to the analog function of the pins), external pull-up resistors must be used when port A pins are used to drive another device.

8.4.2 Port B Pin Functions

The eight port B pins can be used as analog inputs, or as an 8-bit digital input only port. Refer to the following paragraphs for more information.

8.4.2.1 Port B Analog Input Pins

When used as analog inputs, the eight port B pins are referred to as AN[51:48]/AN[3:0]. Since port B functions as analog and digital input only, the analog characteristics are different from those of port A. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for more information on analog signal characteristics. All of the analog signal input pins may be used for at least one other purpose.

8.4.2.2 Port B Digital Input Pins

Port B pins are referred to as PQB[7:0] when used as an 8-bit digital input only port. In addition to functioning as analog input pins, the port B pins are also connected to the input of a synchronizer during reads and may be used as general-purpose digital inputs.

Since port B pins are input only, there is no associated data direction register. Digital input signal states are read from the PORTQB data register. Refer to **D.5.5 Port Data Direction Register** for more information.

— This mode keeps the result registers updated more frequently than any of the other queue operating modes. Software can always read the result table to get the latest converted value for each channel. The channels scanned are kept up to date by the QADC without software involvement.

— This mode may be chosen for either queue, but is normally used only with queue 2. When the software initiated continuous-scan mode is chosen for queue 1, that queue operates continuously and queue 2, being lower in priority, never gets executed. The short interval of time between a queue 1 pause and the internally generated trigger event, or between a queue 1 completion and the subsequent trigger event is not sufficient to allow queue 2 execution to begin.

• External trigger rising or falling edge continuous-scan mode

— The QADC provides external trigger pins for both queues. When this mode is selected, a transition on the associated external trigger pin initiates queue execution. The external trigger is programmable, so that queue execution can begin on either a rising or a falling edge. Each CCW is read and the indicated conversions are performed until an end-of-queue condition is encountered. When the next external trigger edge is detected, queue execution begins again automatically. Software initialization is not needed between trigger events.

• Periodic timer continuous-scan mode

— In addition to the previous modes, queue 2 can also be programmed for the periodic timer continuous-scan mode, where a scan is initiated at a selectable time interval using the on-chip periodic/interval timer. The queue operating mode for queue 2 is selected by the MQ2 field in QACR2.

— The QADC includes a dedicated periodic/interval timer for initiating a scan sequence for queue 2 only. A programmable timer interval can be selected ranging from 2^7 to 2^{17} times the QCLK period in binary multiples.

— When this mode is selected, the timer begins counting. After the programmed interval elapses, the timer generated trigger event starts the queue. The timer is then reloaded and begins counting again. Meanwhile, the QADC automatically performs the conversions in the queue until an end-of-queue condition or a pause is encountered. When a pause is encountered, the QADC waits for the periodic interval to expire again, then continues with the queue. When an end-of-queue is encountered, the next trigger event causes queue execution to begin again with the first CCW in queue 2.

— The periodic timer generates a trigger event whenever the time interval elapses. The trigger event may cause queue execution to continue following a pause or queue completion, or may be considered a trigger overrun. As with all continuous-scan queue operating modes, software action is not needed between trigger events.

— If the queue completion interrupt is enabled when using this mode, software can read the analog results that have just been collected. Software can use this interrupt to obtain non-analog inputs as well, as part of a periodic look at all inputs.

The time base buses originate in a counter submodule and are used by the action submodules. Two time base buses are accessible to each submodule.

The bus interface unit submodule (BIUSM) allows all the CTM4 submodules to pass data to and from the IMB via the submodule bus (SMB).

The counter prescaler submodule (CPSM) generates six different clock frequencies which can be used by any counter submodule. This submodule is contained within the BIUSM.

The free-running counter submodule (FCSM) has a 16-bit up counter with an associated clock source selector, selectable time-base bus drivers, writable control registers, readable status bits, and interrupt logic.The CTM4 has one FCSM.

The modulus counter submodule (MCSM) is an enhanced FCSM. A modulus register gives the additional flexibility of recycling the counter at a count other than 64K clock cycles. The CTM4 has two MCSMs.

The double-action submodule (DASM) provides two 16-bit input capture or two 16-bit output compare functions that can occur automatically without software intervention. The CTM4 has four DASMs.

The pulse width modulation submodule (PWMSM) can generate pulse width modulated signals over a wide range of frequencies, independently of other CTM output signals. PWMSMs are not affected by time base bus activity. The CTM4 has four PWMSMs.

10.2 Address Map

The CTM4 address map occupies 256 bytes from address $YFF400 to $YFF4FF. All CTM4 registers are accessible only when the CPU32 is in supervisor mode. All reserved addresses return zero when read, and writes have no effect. Refer to **D.7 Configurable Timer Module 4** for information concerning CTM4 address map and register bit/field descriptions.

10.3 Time Base Bus System

The CTM4 time base bus system is composed of three 16-bit buses: TBB1, TBB2, and TBB4. These buses are used to transfer timing information from the counter submodules to the action submodules. Two time base buses are available to each submodule. A counter submodule can drive one of the two time base buses to which it is connected. Each action submodule can choose one of the two time base buses to which it is connected as its time base. Control bits within each CTM4 submodule select connection to the appropriate time base bus.

The time base buses are precharge/discharge type buses with wired-OR capability. Therefore, no hardware damage occurs when more than one counter drives the same bus at the same time.

SECTION 12 STANDBY RAM WITH TPU EMULATION

The standby RAM module with TPU emulation capability (TPURAM) consists of a control register block and a 3.5-Kbyte array of fast (two system clock) static RAM, which is especially useful for system stacks and variable storage. The TPURAM responds to both program and data space accesses. The TPURAM can also be used to emulate TPU microcode ROM.

12.1 General

The TPURAM can be mapped to the lower 3.5 Kbytes of any 4-Kbyte boundary in the address map, but must not overlap the module control registers as overlap makes the registers inaccessible. Data can be read or written in bytes, words or long words. The TPURAM is powered by V_{DD} in normal operation. During power-down, TPURAM contents can be maintained by power from the V_{STBY} input. Power switching between sources is automatic.

12.2 TPURAM Register Block

There are three TPURAM control registers: the TPURAM module configuration register (TRAMMCR), the TPURAM test register (TRAMTST), and the TPURAM base address and status register (TRAMBAR). To protect these registers from accidental modification, they are always mapped to supervisor data space.

The TPURAM control register block begins at address $7FFB00 or $FFFB00, depending on the value of the module mapping (MM) bit in the SIM configuration register (SIMCR). Refer to **5.2.1 Module Mapping** for more information on how the state of MM affects the system.

The TPURAM control register block occupies eight bytes of address space. Unimplemented register addresses are read as zeros, and writes have no effect. Refer to **D.9 Standby RAM Module with TPU Emulation Capability (TPURAM)** for register block address map and register bit/field definitions.

12.3 TPURAM Array Address Mapping

The base address and status register TRAMBAR specifies the TPURAM array base address in the MCU memory map. TRAMBAR[15:4] specify the 12 high-order bits of the base address. The TPU bus interface unit compares these bits to address lines ADDR[23:12]. If the two match, then the low order address lines and the SIZ[1:0] signals are used to access the RAM location in the array.

The RAM disable (RAMDS) bit, the LSB of TRAMBAR, indicates whether the TPURAM array is active (RAMDS = 0) or disabled (RAMDS = 1). The array is disabled coming out of reset and remains disabled if the base address field is programmed with an address that overlaps the address of the module control register block. Writing a valid base address to TRAMBAR[15:4] clears RAMDS and enables the array.

SECTION 13 CAN 2.0B CONTROLLER MODULE (TouCAN)

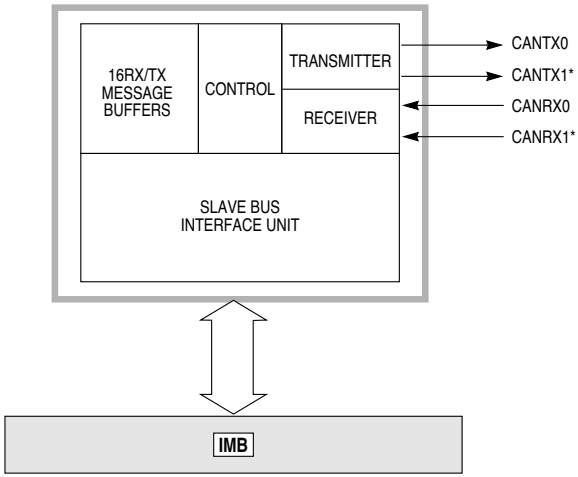
This section is an overview of the TouCAN module. Refer to **D.10 TouCAN Module** for information concerning TouCAN address map and register structure.

13.1 General

The TouCAN module is a communication controller that implements the controller area network (CAN) protocol, an asynchronous communications protocol used in automotive and industrial control systems. It is a high speed (1 Mbit/sec), short distance, priority based protocol which can communicate using a variety of mediums (for example, fiber optic cable or an unshielded twisted pair of wires). The TouCAN supports both the standard and extended identifier (ID) message formats specified in the CAN protocol specification, revision 2.0, part B.

The TouCAN module contains 16 message buffers, which are used for transmit and receive functions. It also contains message filters, which are used to qualify the received message IDs when comparing them to the receive buffer identifiers.

Figure 13-1 shows a block diagram of the TouCAN.



* THESE PINS ARE NOT BONDED ON THE MC68376

TOUCAN BLOCK

Figure 13-1 TouCAN Block Diagram

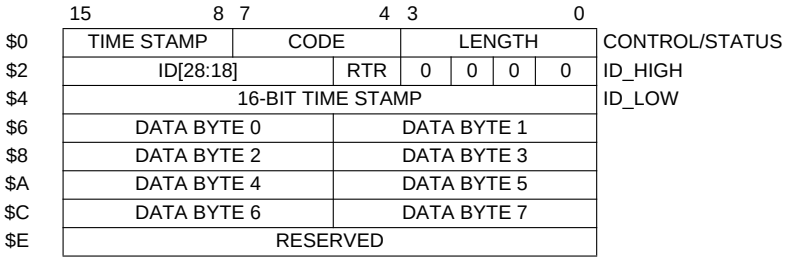


Figure 13-4 Standard ID Message Buffer Structure

13.4.1.1 Common Fields for Extended and Standard Format Frames

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

Table 13-1 Common Extended/Standard Format Frames

Field	Description
Time Stamp	Contains a copy of the high byte of the free running timer, which is captured at the beginning of the identifier field of the frame on the CAN bus.
Code	Refer to **Tables 13-2** and **13-3**
RX Length	Length (in bytes) of the RX data stored in offset $6 through $D of the buffer. This field is written by the TouCAN module, copied from the DLC (data length code) field of the received frame.
TX Length	Length (in bytes) of the data to be transmitted, located in offset $6 through $D of the buffer. This field is written by the CPU32, and is used as the DLC field value. If RTR (remote transmission request) = 1, the frame is a remote frame and will be transmitted without data field, regardless of the value in TX length.
Data	This field can store up to eight data bytes for a frame. For RX frames, the data is stored as it is received from the bus. For TX frames, the CPU32 provides the data to be transmitted within the frame.
Reserved	This word entry field (16 bits) should not be accessed by the CPU32.

Table 13-2 Message Buffer Codes for Receive Buffers

RX Code Before RX New Frame	Description	RX Code After RX New Frame	Comment
0000	NOT ACTIVE — message buffer is not active.	—	—
0100	EMPTY — message buffer is active and empty.	0010	—
0010	FULL — message buffer is full.	0110	If a CPU32 read occurs before the new frame, new receive code is 0010.
0110	OVERRUN — second frame was received into a full buffer before the CPU read the first one.		
0XY1[1]	BUSY — message buffer is now being filled with a new receive frame. This condition will be cleared within 20 cycles.	0010	An empty buffer was filled (XY was 10).
		0110	A full/overrun buffer was filled (Y was 1).

NOTES:
1. For TX message buffers, upon read, the BUSY bit should be ignored.

Table 13-3 Message Buffer Codes for Transmit Buffers

RTR	Initial TX Code	Description	Code After Successful Transmission
X	1000	Message buffer not ready for transmit.	—
0	1100	Data frame to be transmitted once, unconditionally.	1000
1	1100	Remote frame to be transmitted once, and message buffer becomes an RX message buffer for data frames.	0100
0	1010[1]	Data frame to be transmitted only as a response to a remote frame, always.	1010
0	1110	Data frame to be transmitted only once, unconditionally, and then only as a response to remote frame, always.	1010

NOTES:
1. When a matching remote request frame is detected, the code for such a message buffer is changed to be 1110.

13.4.1.2 Fields for Extended Format Frames

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

Table 13-4 Extended Format Frames

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

13.4.1.3 Fields for Standard Format Frames

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

bit in the matching transmit message buffer is set, the TouCAN will transmit a remote frame as a response.

A received remote frame is not stored in a receive message buffer. It is only used to trigger the automatic transmission of a frame in response. The mask registers are not used in remote frame ID matching. All ID bits (except RTR) of the incoming received frame must match for the remote frame to trigger a response transmission.

13.5.6 Overload Frames

Overload frame transmissions are not initiated by the TouCAN unless certain conditions are detected on the CAN bus. These conditions include:

- Detection of a dominant bit in the first or second bit of intermission.
- Detection of a dominant bit in the seventh (last) bit of the end-of-frame (EOF) field in receive frames.
- Detection of a dominant bit in the eighth (last) bit of the error frame delimiter or overload frame delimiter.

13.6 Special Operating Modes

The TouCAN module has three special operating modes:

- Debug mode
- Low-power stop mode
- Auto power save mode

13.6.1 Debug Mode

Debug mode is entered by setting the HALT bit in the CANMCR, or by assertion of the IMB FREEZE line. In both cases, the FRZ1 bit in CANMCR must also be set to allow HALT or FREEZE to place the TouCAN in debug mode.

Once entry into debug mode is requested, the TouCAN waits until an intermission or idle condition exists on the CAN bus, or until the TouCAN enters the error passive or bus off state. Once one of these conditions exists, the TouCAN waits for the completion of all internal activity. When this happens, the following events occur:

- The TouCAN stops transmitting/receiving frames.
- The prescaler is disabled, thus halting all CAN bus communication.
- The TouCAN ignores its RX pins and drives its TX pins as recessive.
- The TouCAN loses synchronization with the CAN bus and the NOTRDY and FRZACK bits in CANMCR are set.
- The CPU32 is allowed to read and write the error counter registers.

After engaging one of the mechanisms to place the TouCAN in debug mode, the user must wait for the FRZACK bit to be set before accessing any other registers in the TouCAN, otherwise unpredictable operation may occur.

To exit debug mode, the IMB FREEZE line must be negated or the HALT bit in CANMCR must be cleared.

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.

Table D-33 Effect of DDRQS on QSM Pin Function

QSM Pin	Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial data input to QSPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from QSPI
MOSI	Master	DDQS1	0	Disables data output
			1	Serial data output from QSPI
	Slave		0	Serial data input to QSPI
			1	Disables data input
SCK[1]	Master	DDQS2	—	Clock output from QSPI
	Slave		—	Clock input to QSPI
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion causes mode fault
			1	Chip-select output
	Slave		0	QSPI slave select input
			1	Disables slave select Input
PCS[1:3]	Master	DDQS[4:6]	0	Disables chip-select output
			1	Chip-select output
	Slave		0	Inactive
			1	Inactive
TXD[2]	—	DDQS7	X	Serial data output from SCI
RXD	—	None	NA	Serial data input to SCI

NOTES:
1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial data output TXD.

DDRQS determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output.

D.6.11 QSPI Control Register 0

SPCR0 — QSPI Control Register 0 **$YFFC18**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSTR	WOMQ	BITS[3:0]				CPOL	CPHA	SPBR[7:0]							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

SPCR0 contains parameters for configuring the QSPI and enabling various modes of operation. The CPU32 has read/write access to SPCR0, but the QSM has read access only. SPCR0 must be initialized before QSPI operation begins. Writing a new value to SPCR0 while the QSPI is enabled disrupts operation.

MSTR — Master/Slave Mode Select
0 = QSPI is a slave device.
1 = QSPI is the system master.

WOMQ — Wired-OR Mode for QSPI Pins
0 = Pins designated for output by DDRQS operate in normal mode.
1 = Pins designated for output by DDRQS operate in open-drain mode.

STF — Stop Flag

0 = TPU is operating.
1 = TPU is stopped (STOP bit has been set).

SUPV — Supervisor/Unrestricted

0 = Assignable registers are accessible in user or supervisor mode.
1 = Assignable registers are accessible in supervisor mode only.

PSCK — Prescaler Clock

0 = $f_{sys} \div 32$ is input to TCR1 prescaler.
1 = $f_{sys} \div 4$ is input to TCR1 prescaler.

IARB[3:0] — Interrupt Arbitration ID

The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value.

D.8.2 Test Configuration Register

TCR — Test Configuration Register **$YFFE02**

Used for factory test only.

D.8.3 Development Support Control Register

DSCR — Development Support Control Register **$YFFE04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HOT4	NOT USED				BLC	CLKS	FRZ[1:0]		CCL	BP	BC	BH	BL	BM	BT
RESET:															
0					0	0	0	0	0	0	0	0	0	0	0

HOT4 — Hang on T4

0 = Exit wait on T4 state caused by assertion of HOT4.
1 = Enter wait on T4 state.

BLC — Branch Latch Control

0 = Latch conditions into branch condition register before exiting halted state.
1 = Do not latch conditions into branch condition register before exiting the halted state or during the time-slot transition period.

CLKS — Stop Clocks (to TCRs)

0 = Do not stop TCRs.
1 = Stop TCRs during the halted state.

FRZ[1:0] — FREEZE Assertion Response

The FRZ bits specify the TPU microengine response to the IMB FREEZE signal. Refer to **Table D-54**.

–D–

–E–