



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

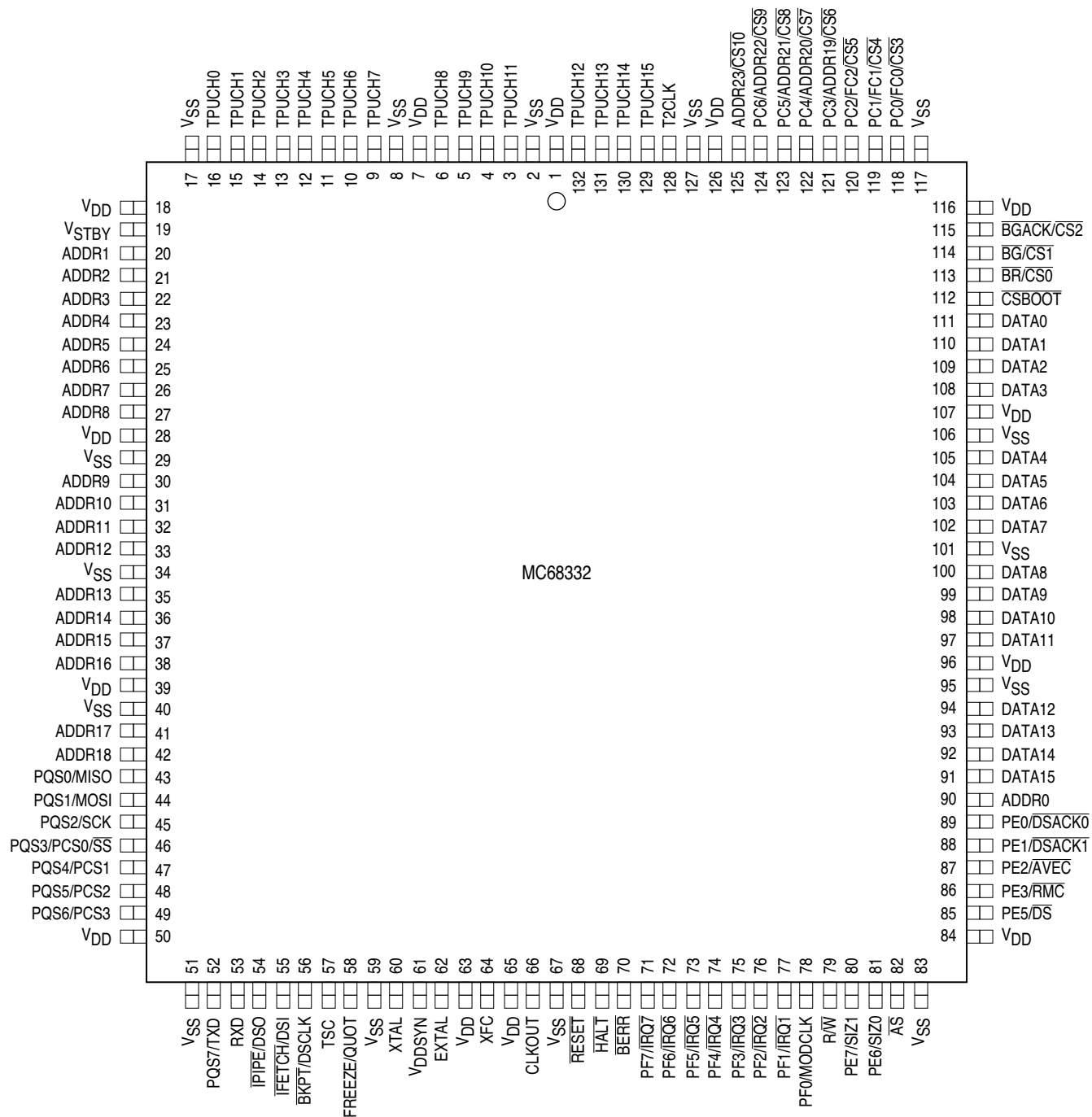
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Not For New Designs
Core Processor	CPU32
Core Size	32-Bit Single-Core
Speed	16MHz
Connectivity	EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68332aceh16">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68332aceh16</a>

### 1.3 Pin Assignments



### Figure 2 MC68332 132-Pin QFP Pin Assignments

## 2 Signal Descriptions

### 2.1 Pin Characteristics

The following table shows MCU pins and their characteristics. All inputs detect CMOS logic levels. All inputs can be put in a high-impedance state, but the method of doing this differs depending upon pin function. Refer to the table, MCU Driver Types, for a description of output drivers. An entry in the discrete I/O column of the MCU Pin Characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to the MCU Block Diagram for information about port organization.

**Table 2 MCU Pin Characteristic**

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/CS10/ECLK	A	Y	N	O	—
ADDR[22:19]/CS[9:6]	A	Y	N	O	PC[6:3]
ADDR[18:0]	A	Y	N	—	—
$\overline{AS}$	B	Y	N	I/O	PE5
$\overline{AVEC}$	B	Y	N	I/O	PE2
$\overline{BERR}$	B	Y	N	—	—
$\overline{BG/CS1}$	B	—	—	—	—
$\overline{BGACK/CS2}$	B	Y	N	—	—
$\overline{BKPT/DSCLK}$	—	Y	Y	—	—
$\overline{BR/CS0}$	B	Y	N	—	—
CLKOUT	A	—	—	—	—
$\overline{CSBOOT}$	B	—	—	—	—
DATA[15:0] <sup>1</sup>	Aw	Y	N	—	—
$\overline{DS}$	B	Y	N	I/O	PE4
$\overline{DSACK1}$	B	Y	N	I/O	PE1
$\overline{DSACK0}$	B	Y	N	I/O	PE0
DSI/IFETCH	A	Y	Y	—	—
DSO/IPIPE	A	—	—	—	—
EXTAL <sup>2</sup>	—	—	Special	—	—
FC[2:0]/CS[5:3]	A	Y	N	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
$\overline{HALT}$	Bo	Y	N	—	—
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	PF[7:1]
MISO	Bo	Y	Y	I/O	PQS0
MODCLK <sup>1</sup>	B	Y	N	I/O	PF0
MOSI	Bo	Y	Y	I/O	PQS1
PCS0/ $\overline{SS}$	Bo	Y	Y	I/O	PQS3
PCS[3:1]	Bo	Y	Y	I/O	PQS[6:4]
R/W	A	Y	N	—	—
$\overline{RESET}$	Bo	Y	Y	—	—
$\overline{RMC}$	B	Y	N	I/O	PE3
RXD	—	N	N	—	—
SCK	Bo	Y	Y	I/O	PQS2
SIZ[1:0]	B	Y	N	I/O	PE[7:6]

**Table 5 MCU Signal Characteristics (Continued)**

Signal Name	MCU Module	Signal Type	Active State
TSC	SIM	Input	—
TXD	QSM	Output	—
XFC	SIM	Input	—
XTAL	SIM	Output	—

## 2.5 Signal Function

**Table 6 MCU Signal Function**

Signal Name	Mnemonic	Function
Address Bus	ADDR[23:0]	24-bit address bus
Address Strobe	$\overline{AS}$	Indicates that a valid address is on the address bus
Autovector	$\overline{AVEC}$	Requests an automatic vector during interrupt acknowledge
Bus Error	$\overline{BERR}$	Indicates that a bus error has occurred
Bus Grant	$\overline{BG}$	Indicates that the MCU has relinquished the bus
Bus Grant Acknowledge	$\overline{BGACK}$	Indicates that an external device has assumed bus mastership
Breakpoint	$\overline{BKPT}$	Signals a hardware breakpoint to the CPU
Bus Request	$\overline{BR}$	Indicates that an external device requires bus mastership
System Clockout	CLKOUT	System clock output
Chip Selects	$\overline{CS}[10:0]$	Select external devices at programmed addresses
Boot Chip Select	$\overline{CSBOOT}$	Chip select for external boot start-up ROM
Data Bus	DATA[15:0]	16-bit data bus
Data Strobe	$\overline{DS}$	During a read cycle, indicates when it is possible for an external device to place data on the data bus. During a write cycle, indicates that valid data is on the data bus.
Data and Size Acknowledge	$\overline{DSACK}[1:0]$	Provide asynchronous data transfers and dynamic bus sizing
Development Serial In, Out, Clock	DSI, DSO, DSCLK	Serial I/O and clock for background debugging mode
Crystal Oscillator	EXTAL, XTAL	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
Function Codes	FC[2:0]	Identify processor state and current address space
Freeze	FREEZE	Indicates that the CPU has entered background mode
Halt	$\overline{HALT}$	Suspend external bus activity
Instruction Pipeline	$\overline{IFETCH}$ $\overline{IPIPE}$	Indicate instruction pipeline activity
Interrupt Request Level	$\overline{IRQ}[7:1]$	Provides an interrupt priority level to the CPU
Master In Slave Out	MISO	Serial input to QSPI in master mode; serial output from QSPI in slave mode
Clock Mode Select	MODCLK	Selects the source and type of system clock
Master Out Slave In	MOSI	Serial output from QSPI in master mode; serial input to QSPI in slave mode
Port C	PC[6:0]	SIM digital output port signals
Peripheral Chip Select	PCS[3:0]	QSPI peripheral chip selects
Port E	PE[7:0]	SIM digital I/O port signals
Port F	PF[7:0]	SIM digital I/O port signals
Port QS	PQS[7:0]	QSM digital I/O port signals

**Table 7 SIM Address Map**

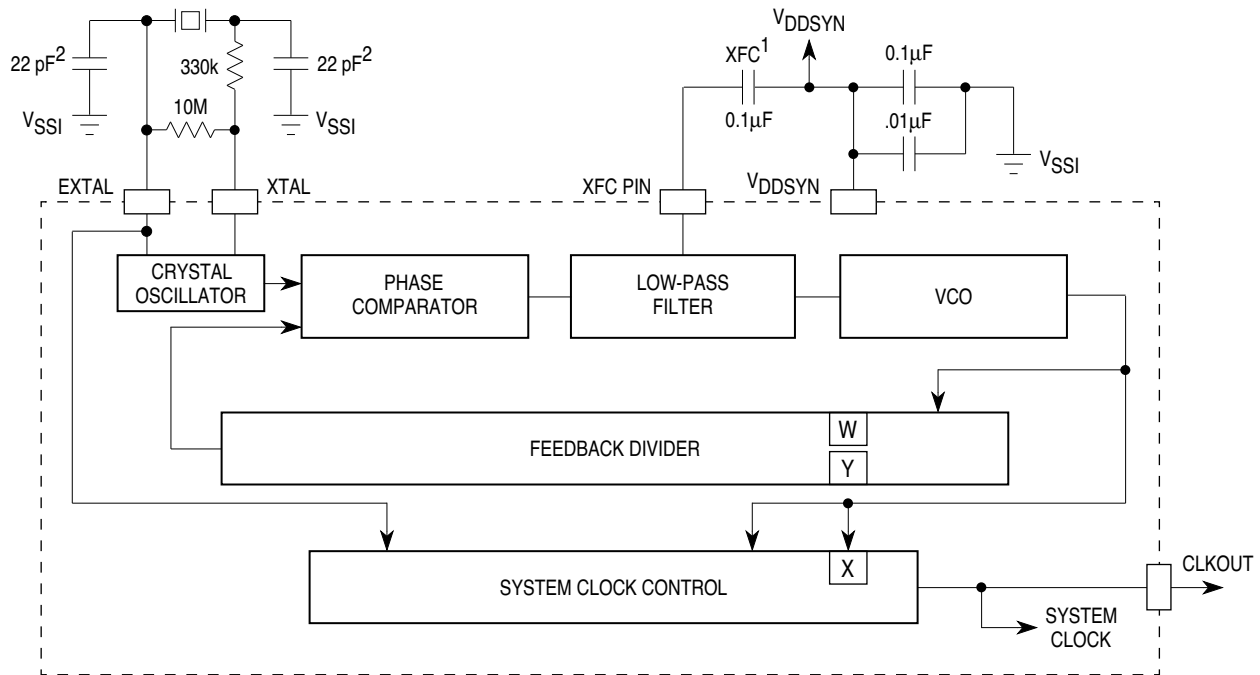
Access	Address	15	8	7	0
S	\$YFFA00	SIM CONFIGURATION (SIMCR)			
S	\$YFFA02	FACTORY TEST (SIMTR)			
S	\$YFFA04	CLOCK SYNTHESIZER CONTROL (SYNCR)			
S	\$YFFA06	NOT USED		RESET STATUS REGISTER (RSR)	
S	\$YFFA08	MODULE TEST E (SIMTRE)			
S	\$YFFA0A	NOT USED		NOT USED	
S	\$YFFA0C	NOT USED		NOT USED	
S	\$YFFA0E	NOT USED		NOT USED	
S/U	\$YFFA10	NOT USED		PORT E DATA (PORTE0)	
S/U	\$YFFA12	NOT USED		PORT E DATA (PORTE1)	
S/U	\$YFFA14	NOT USED		PORT E DATA DIRECTION (DDRE)	
S	\$YFFA16	NOT USED		PORT E PIN ASSIGNMENT (PEPAR)	
S/U	\$YFFA18	NOT USED		PORT F DATA (PORTF0)	
S/U	\$YFFA1A	NOT USED		PORT F DATA (PORTF1)	
S/U	\$YFFA1C	NOT USED		PORT F DATA DIRECTION (DDRF)	
S	\$YFFA1E	NOT USED		PORT F PIN ASSIGNMENT (PFPAR)	
S	\$YFFA20	NOT USED		SYSTEM PROTECTION CONTROL (SYPCR)	
S	\$YFFA22	PERIODIC INTERRUPT CONTROL (PICR)			
S	\$YFFA24	PERIODIC INTERRUPT TIMING (PITR)			
S	\$YFFA26	NOT USED		SOFTWARE SERVICE (SWSR)	
S	\$YFFA28	NOT USED		NOT USED	
S	\$YFFA2A	NOT USED		NOT USED	
S	\$YFFA2C	NOT USED		NOT USED	
S	\$YFFA2E	NOT USED		NOT USED	
S	\$YFFA30	TEST MODULE MASTER SHIFT A (TSTMSRA)			
S	\$YFFA32	TEST MODULE MASTER SHIFT B (TSTMSRB)			
S	\$YFFA34	TEST MODULE SHIFT COUNT (TSTSC)			
S	\$YFFA36	TEST MODULE REPETITION COUNTER (TSTRC)			
S	\$YFFA38	TEST MODULE CONTROL (CREG)			
S/U	\$YFFA3A	TEST MODULE DISTRIBUTED REGISTER (DREG)			
	\$YFFA3C	NOT USED		NOT USED	
	\$YFFA3E	NOT USED		NOT USED	
S/U	\$YFFA40	NOT USED		PORT C DATA (PORTC)	
	\$YFFA42	NOT USED		NOT USED	
S	\$YFFA44	CHIP-SELECT PIN ASSIGNMENT (CSPAR0)			
S	\$YFFA46	CHIP-SELECT PIN ASSIGNMENT (CSPAR1)			
S	\$YFFA48	CHIP-SELECT BASE BOOT (CSBARBT)			
S	\$YFFA4A	CHIP-SELECT OPTION BOOT (CSORBT)			
S	\$YFFA4C	CHIP-SELECT BASE 0 (CSBAR0)			
S	\$YFFA4E	CHIP-SELECT OPTION 0 (CSOR0)			
S	\$YFFA50	CHIP-SELECT BASE 1 (CSBAR1)			
S	\$YFFA52	CHIP-SELECT OPTION 1 (CSOR1)			
S	\$YFFA54	CHIP-SELECT BASE 2 (CSBAR2)			

## 3.3 System Clock

The system clock in the SIM provides timing signals for the IMB modules and for an external peripheral bus. Because MCU operation is fully static, register and memory contents are not affected when the clock rate changes. System hardware and software support changes in the clock rate during operation.

The system clock signal can be generated in three ways. An internal phase-locked loop can synthesize the clock from an internal or external frequency source, or the clock signal can be input from an external source.

Following is a block diagram of the clock submodule.



1. MUST BE LOW-LEAKAGE CAPACITOR (INSULATION RESISTANCE 30,000 MΩ OR GREATER).
2. RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A DAISHINKU DMX-38 32.768-kHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

SYN CLOCK  
BLOCK 32KHZ

**Figure 7 System Clock Block Diagram**

### 3.3.1 Clock Sources

The state of the clock mode (MODCLK) pin during reset determines the clock source. When MODCLK is held high during reset, the clock synthesizer generates a clock signal from either a crystal oscillator or an external reference input. Clock synthesizer control register SYNCR determines operating frequency and various modes of operation. When MODCLK is held low during reset, the clock synthesizer is disabled, and an external system clock signal must be applied. When the synthesizer is disabled, SYNCR control bits have no effect.

A reference crystal must be connected between the EXTAL and XTAL pins to use the internal oscillator. Use of a 32.768-kHz crystal is recommended. These crystals are inexpensive and readily available. If an external reference signal or an external system clock signal is applied through the EXTAL pin, the XTAL pin must be left floating. External reference signal frequency must be less than or equal to maximum specified reference frequency. External system clock signal frequency must be less than or equal to maximum specified system clock frequency.

## 3.4 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. The external bus has 24 address lines and 16 data lines.

The EBI provides dynamic sizing between 8-bit and 16-bit data accesses. It supports byte, word, and long-word transfers. Ports are accessed through the use of asynchronous cycles controlled by the data transfer (SIZ1 and SIZ0) and data size acknowledge pins (DSACK1 and DSACK0). Multiple bus cycles may be required for a transfer to or from an 8-bit port.

Port width is the maximum number of bits accepted or provided during a bus transfer. External devices must follow the handshake protocol described below. Control signals indicate the beginning of the cycle, the address space, the size of the transfer, and the type of cycle. The selected device controls the length of the cycle. Strobe signals, one for the address bus and another for the data bus, indicate the validity of an address and provide timing information for data. The EBI operates in an asynchronous mode for any port width.

To add flexibility and minimize the necessity for external logic, MCU chip-select logic can be synchronized with EBI transfers. Chip-select logic can also provide internally-generated bus control signals for these accesses. Refer to **3.5 Chip Selects** for more information.

### 3.4.1 Bus Control Signals

The CPU initiates a bus cycle by driving the address, size, function code, and read/write outputs. At the beginning of the cycle, size signals SIZ0 and SIZ1 are driven along with the function code signals. The size signals indicate the number of bytes remaining to be transferred during an operand cycle. They are valid while the address strobe ( $\overline{AS}$ ) is asserted. The following table shows SIZ0 and SIZ1 encoding. The read/write ( $R/\overline{W}$ ) signal determines the direction of the transfer during a bus cycle. This signal changes state, when required, at the beginning of a bus cycle, and is valid while  $\overline{AS}$  is asserted.  $R/\overline{W}$  only changes state when a write cycle is preceded by a read cycle or vice versa. The signal can remain low for two consecutive write cycles.

**Table 8 Size Signal Encoding**

SIZ1	SIZ0	Transfer Size
0	1	Byte
1	0	Word
1	1	Three Byte
0	0	Long Word

### 3.4.2 Function Codes

The CPU32 automatically generates function code signals FC[2:0]. The function codes can be considered address extensions that automatically select one of eight address spaces to which an address applies. These spaces are designated as either user or supervisor, and program or data spaces. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid while  $\overline{AS}$  is asserted.

**Table 9 CPU32 Address Space Encoding**

FC2	FC1	FC0	Address Space
0	0	0	Reserved
0	0	1	User Data Space
0	1	0	User Program Space
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Supervisor Data Space
1	1	0	Supervisor Program Space
1	1	1	CPU Space

### 3.4.3 Address Bus

Address bus signals ADDR[23:0] define the address of the most significant byte to be transferred during a bus cycle. The MCU places the address on the bus at the beginning of a bus cycle. The address is valid while  $\overline{AS}$  is asserted.

### 3.4.4 Address Strobe

$\overline{AS}$  is a timing signal that indicates the validity of an address on the address bus and the validity of many control signals. It is asserted one-half clock after the beginning of a bus cycle.

### 3.4.5 Data Bus

Data bus signals DATA[15:0] make up a bidirectional, non-multiplexed parallel bus that transfers data to or from the MCU. A read or write operation can transfer 8 or 16 bits of data in one bus cycle. During a read cycle, the data is latched by the MCU on the last falling edge of the clock for that bus cycle. For a write cycle, all 16 bits of the data bus are driven, regardless of the port width or operand size. The MCU places the data on the data bus one-half clock cycle after  $\overline{AS}$  is asserted in a write cycle.

### 3.4.6 Data Strobe

Data strobe ( $\overline{DS}$ ) is a timing signal. For a read cycle, the MCU asserts  $\overline{DS}$  to signal an external device to place data on the bus.  $\overline{DS}$  is asserted at the same time as  $\overline{AS}$  during a read cycle. For a write cycle,  $\overline{DS}$  signals an external device that data on the bus is valid. The MCU asserts  $\overline{DS}$  one full clock cycle after the assertion of  $\overline{AS}$  during a write cycle.

### 3.4.7 Bus Cycle Termination Signals

During bus cycles, external devices assert the data transfer and size acknowledge signals ( $\overline{DSACK1}$  and  $\overline{DSACK0}$ ). During a read cycle, the signals tell the MCU to terminate the bus cycle and to latch data. During a write cycle, the signals indicate that an external device has successfully stored data and that the cycle can end. These signals also indicate to the MCU the size of the port for the bus cycle just completed. (Refer to 3.4.9 Dynamic Bus Sizing.)

The bus error ( $\overline{BERR}$ ) signal is also a bus cycle termination indicator and can be used in the absence of  $\overline{DSACK1}$  and  $\overline{DSACK0}$  to indicate a bus error condition. It can also be asserted in conjunction with these signals, provided it meets the appropriate timing requirements. The internal bus monitor can be used to generate the  $\overline{BERR}$  signal for internal and internal-to-external transfers. When  $\overline{BERR}$  and  $\overline{HALT}$  are asserted simultaneously, the CPU takes a bus error exception.

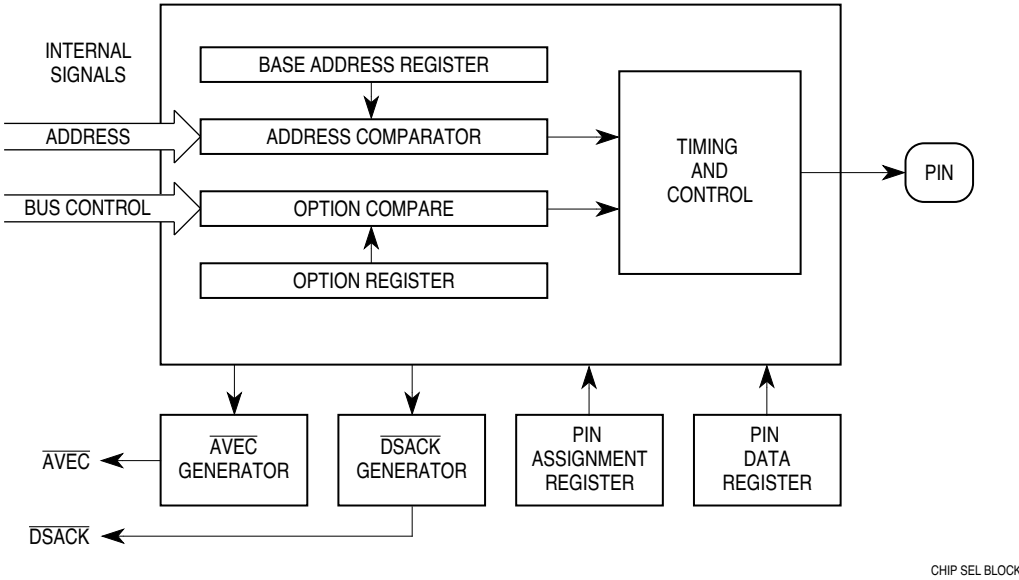
Autovector signal ( $\overline{AVEC}$ ) can terminate external  $\overline{IRQ}$  pin interrupt acknowledge cycles.  $\overline{AVEC}$  indicates that the MCU will internally generate a vector number to locate an interrupt handler routine. If it is continuously asserted, autovectors will be generated for all external interrupt requests.  $\overline{AVEC}$  is ignored during all other bus cycles.



Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobes, or interrupt acknowledge signals. Logic can also generate  $\overline{\text{DSACK}}$  signals internally. A single  $\overline{\text{DSACK}}$  generator is shared by all circuits. Multiple chip selects assigned to the same address and control must have the same number of wait states.

Chip selects can also be synchronized with the ECLK signal available on ADDR23.

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low. Refer to the following block diagram of a single chip-select circuit.



**Figure 9 Chip-Select Circuit Block Diagram**

The following table lists allocation of chip-selects and discrete outputs on the pins of the MCU.

Pin	Chip Select	Discrete Outputs
$\overline{\text{CSBOOT}}$	$\overline{\text{CSBOOT}}$	—
BR	CS0	—
BG	CS1	—
BGACK	CS2	—
FC0	CS3	PC0
FC1	CS4	PC1
FC2	CS5	PC2
ADDR19	CS6	PC3
ADDR20	CS7	PC4
ADDR21	CS8	PC5
ADDR22	CS9	PC6
ADDR23	CS10	ECLK

**Table 18 Reset Mode Selection**

DATA0	CSBOOT 16-Bit	CSBOOT 8-Bit
DATA1	CS0 CS1 CS2	BR BG BGACK
DATA2	CS3 CS4 CS5	FC0 FC1 FC2
DATA3 DATA4 DATA5 DATA6 DATA7	CS6 CS[7:6] CS[8:6] CS[9:6] CS[10:6]	ADDR19 ADDR[20:19] ADDR[21:19] ADDR[22:19] ADDR[23:19]
DATA8	DSACK0, DSACK1, AVEC, DS, AS, SIZ[1:0]	PORTE
DATA9	IRQ[7:1] MODCLK	PORTF
DATA11	Test Mode Disabled	Test Mode Enabled
MODCLK	VCO = System Clock	EXTAL = System Clock
BKPT	Background Mode Disabled	Background Mode Enabled

### 3.7.2 Functions of Pins for Other Modules During Reset

Generally, pins associated with modules other than the SIM default to port functions, and input/output ports are set to input state. This is accomplished by disabling pin functions in the appropriate control registers, and by clearing the appropriate port data direction registers. Refer to individual module sections in this manual for more information. The following table is a summary of module pin function out of reset.

**Table 19 Module Pin Functions**

Module	Pin Mnemonic	Function
CPU32	DSI/IFETCH	DSI/IFETCH
	DSO/IPIPE	DSO/IPIPE
	BKPT/DSCLK	BKPT/DSCLK
GPT	PGP7/IC4/OC5	Discrete Input
	PGP[6:3]/OC[4:1]	Discrete Input
	PGP[2:0]/IC[3:1]	Discrete Input
	PAI	Discrete Input
	PCLK	Discrete Input
	PWMA, PWMB	Discrete Output
QSM	PQS7/TXD	Discrete Input
	PQS[6:4]/PCS[3:1]	Discrete Input
	PQS3/PCS0/SS	Discrete Input
	PQS2/SCK	Discrete Input
	PQS1/MOSI	Discrete Input
	PQS0/MISO	Discrete Input
	RXD	RXD

## 3.7.3 Reset Timing

The  $\overline{\text{RESET}}$  input must be asserted for a specified minimum period in order for reset to occur. External  $\overline{\text{RESET}}$  assertion can be delayed internally for a period equal to the longest bus cycle time (or the bus monitor time-out period) in order to protect write cycles from being aborted by reset. While  $\overline{\text{RESET}}$  is asserted, SIM pins are either in a disabled high-impedance state or are driven to their inactive states.

When an external device asserts  $\overline{\text{RESET}}$  for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the  $\overline{\text{RESET}}$  pin low for an additional 512 CLKOUT cycles after it detects that the  $\overline{\text{RESET}}$  signal is no longer being externally driven, to guarantee this length of reset to the entire system.

If an internal source asserts a reset signal, the reset control logic asserts  $\overline{\text{RESET}}$  for a minimum of 512 cycles. If the reset signal is still asserted at the end of 512 cycles, the control logic continues to assert  $\overline{\text{RESET}}$  until the internal reset signal is negated.

After 512 cycles have elapsed, the reset input pin goes to an inactive, high-impedance state for ten cycles. At the end of this 10-cycle period, the reset input is tested. When the input is at logic level one, reset exception processing begins. If, however, the reset input is at logic level zero, the reset control logic drives the pin low for another 512 cycles. At the end of this period, the pin again goes to high-impedance state for ten cycles, then it is tested again. The process repeats until  $\overline{\text{RESET}}$  is released.

## 3.7.4 Power-On Reset

When the SIM clock synthesizer is used to generate the system clock, power-on reset involves special circumstances related to application of system and clock synthesizer power. Regardless of clock source, voltage must be applied to clock synthesizer power input pin  $V_{\text{DDSYN}}$  in order for the MCU to operate. The following discussion assumes that  $V_{\text{DDSYN}}$  is applied before and during reset. This minimizes crystal start-up time. When  $V_{\text{DDSYN}}$  is applied at power-on, start-up time is affected by specific crystal parameters and by oscillator circuit design.  $V_{\text{DD}}$  ramp-up time also affects pin state during reset.

During power-on reset, an internal circuit in the SIM drives the internal (IMB) and external reset lines. The circuit releases the internal reset line as  $V_{\text{DD}}$  ramps up to the minimum specified value, and SIM pins are initialized. When  $V_{\text{DD}}$  reaches the specified minimum value, the clock synthesizer VCO begins operation. Clock frequency ramps up to the specified limp mode frequency. The external  $\overline{\text{RESET}}$  line remains asserted until the clock synthesizer PLL locks and 512 CLKOUT cycles elapse.

The SIM clock synthesizer provides clock signals to the other MCU modules. After the clock is running and the internal reset signal is asserted for four clock cycles, these modules reset.  $V_{\text{DD}}$  ramp time and VCO frequency ramp time determine how long these four cycles take. Worst case is approximately 15 milliseconds. During this period, module port pins may be in an indeterminate state. While input-only pins can be put in a known state by means of external pull-up resistors, external logic on input/output or output-only pins must condition the lines during this time. Active drivers require high-impedance buffers or isolation resistors to prevent conflict.

## 3.7.5 Use of Three State Control Pin

Asserting the three-state control (TSC) input causes the MCU to put all output drivers in an inactive, high-impedance state. The signal must remain asserted for ten clock cycles in order for drivers to change state. There are certain constraints on use of TSC during power-on reset:

When the internal clock synthesizer is used (MODCLK held high during reset), synthesizer ramp-up time affects how long the ten cycles take. Worst case is approximately 20 milliseconds from TSC assertion.

When an external clock signal is applied (MODCLK held low during reset), pins go to high-impedance state as soon after TSC assertion as ten clock pulses have been applied to the EXTAL pin.

## 4.3 Status Register

The status register contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The lower byte containing the condition codes is the only portion of the register 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, including the interrupt priority mask and additional control bits.

### SR —Status Register

15	14	13	12	11	10	8	7	6	5	4	3	2	1	0
T1	T0	S	0	0	IP	0	0	0	X	N	Z	V	C	
RESET:														
0	0	1	0	0	1	1	1	0	0	0	U	U	U	U

### System Byte

- T[1:0] —Trace Enable
- S —Supervisor/User State
- Bits [12:11] —Unimplemented
- IP[2:0] —Interrupt Priority Mask

### User Byte (Condition Code Register)

- Bits [7:5] — Unimplemented
- X — Extend
- N — Negative
- Z — Zero
- V — Overflow
- C — Carry

## 4.4 Data Types

Six basic data types are supported:

- Bits
- Packed Binary Coded Decimal Digits
- Byte Integers (8 bits)
- Word Integers (16 bits)
- Long-Word Integers (32 bits)
- Quad-Word Integers (64 bits)

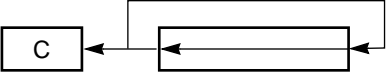

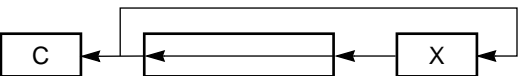
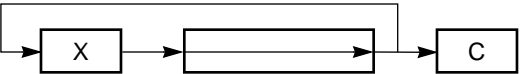
## 4.5 Addressing Modes

Addressing in the CPU32 is register-oriented. Most instructions allow the results of the specified operation to be placed either in a register or directly in memory. This flexibility eliminates the need for extra instructions to store register contents in memory. The CPU32 supports seven basic addressing modes:

- Register direct
- Register indirect
- Register indirect with index
- Program counter indirect with displacement
- Program counter indirect with index
- Absolute
- Immediate

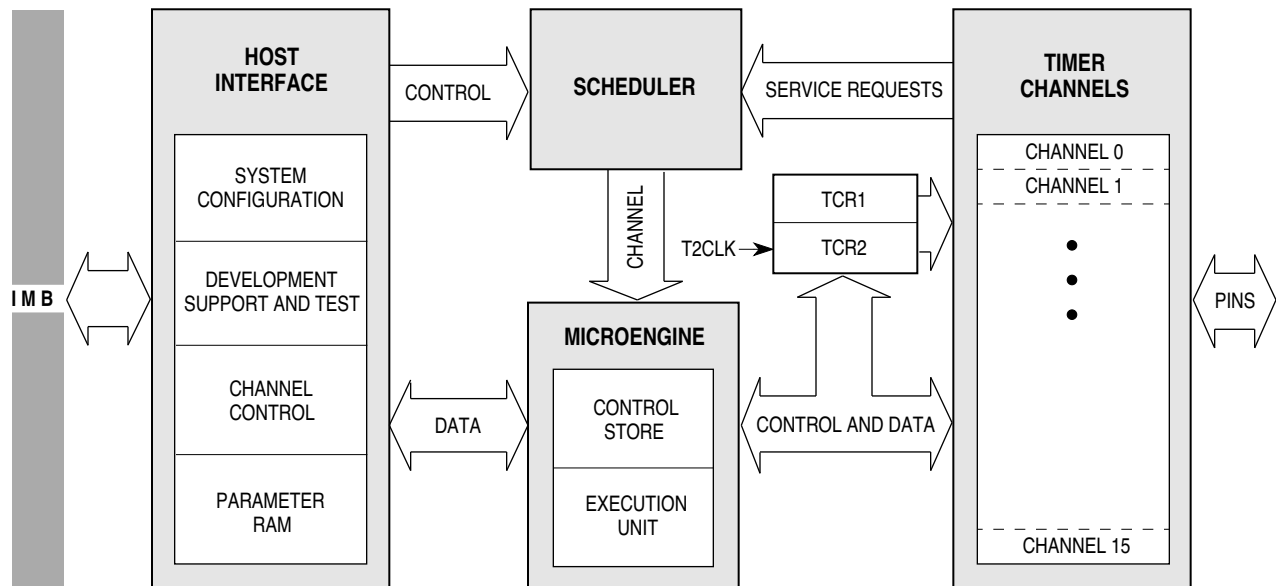
Included in the register indirect addressing modes are the capabilities to post-increment, predecrement, and offset. The program counter relative mode also has index and offset capabilities. In addition to these addressing modes, many instructions implicitly specify the use of the status register, stack pointer, or program counter.

Table 20 Instruction Set Summary(Continued)

Instruction	Syntax	Operand Size	Operation
MOVES <sup>1</sup>	Rn, <ea> <ea>, Rn	8, 16, 32	Rn $\Rightarrow$ Destination using DFC Source using SFC $\Rightarrow$ Rn
MULS/MULU	<ea>, Dn <ea>, DI <ea>, Dh : DI	16 * 16 $\Rightarrow$ 32 32 * 32 $\Rightarrow$ 32 32 * 32 $\Rightarrow$ 64	Source * Destination $\Rightarrow$ Destination (signed or unsigned)
NBCD	$\dot{I}$	8 8	0 – Destination <sub>10</sub> – X $\Rightarrow$ Destination
NEG	$\dot{I}$	8, 16, 32	0 – Destination $\Rightarrow$ Destination
NEGX	$\dot{I}$	8, 16, 32	0 – Destination – X $\Rightarrow$ Destination
NOP	none	none	PC + 2 $\Rightarrow$ PC
NOT	$\dot{I}$	8, 16, 32	Destination $\Rightarrow$ Destination
OR	<ea>, Dn Dn, <ea>	8, 16, 32 8, 16, 32	Source + Destination $\Rightarrow$ Destination
ORI	#<data>, <ea>	8, 16, 32	Data + Destination $\Rightarrow$ Destination
ORI to CCR	#<data>, CCR	16	Source + CCR $\Rightarrow$ SR
ORI to SR <sup>1</sup>	#<data>, SR	16	Source ; SR $\Rightarrow$ SR
PEA	$\dot{I}$	32	SP – 4 $\Rightarrow$ SP; <ea> $\Rightarrow$ SP
RESET <sup>1</sup>	none	none	Assert RESET line
ROL	Dn, Dn #<data>, Dn $\dot{I}$	8, 16, 32 8, 16, 32 16	
ROR	Dn, Dn #<data>, Dn $\dot{I}$	8, 16, 32 8, 16, 32 16	
ROXL	Dn, Dn #<data>, Dn $\dot{I}$	8, 16, 32 8, 16, 32 16	
ROXR	Dn, Dn #<data>, Dn $\dot{I}$	8, 16, 32 8, 16, 32 16	
RTD	#d	16	(SP) $\Rightarrow$ PC; SP + 4 + d $\Rightarrow$ SP
RTE <sup>1</sup>	none	none	(SP) $\Rightarrow$ SR; SP + 2 $\Rightarrow$ SP; (SP) $\Rightarrow$ PC; SP + 4 $\Rightarrow$ SP; Restore stack according to format
RTR	none	none	(SP) $\Rightarrow$ CCR; SP + 2 $\Rightarrow$ SP; (SP) $\Rightarrow$ PC; SP + 4 $\Rightarrow$ SP
RTS	none	none	(SP) $\Rightarrow$ PC; SP + 4 $\Rightarrow$ SP
SBCD	Dn, Dn – (An), – (An)	8 8	Destination <sub>10</sub> – Source <sub>10</sub> – X $\Rightarrow$ Destination
Scc	$\dot{I}$	8	If condition true, then destination bits are set to 1; else, destination bits are cleared to 0
STOP <sup>1</sup>	#<data>	16	Data $\Rightarrow$ SR; STOP
SUB	<ea>, Dn Dn, <ea>	8, 16, 32	Destination – Source $\Rightarrow$ Destination
SUBA	<ea>, An	16, 32	Destination – Source $\Rightarrow$ Destination
SUBI	#<data>, <ea>	8, 16, 32	Destination – Data $\Rightarrow$ Destination
SUBQ	#<data>, <ea>	8, 16, 32	Destination – Data $\Rightarrow$ Destination
SUBX	Dn, Dn – (An), – (An)	8, 16, 32 8, 16, 32	Destination – Source – X $\Rightarrow$ Destination

## 5 Time Processor Unit

The time processor unit (TPU) provides optimum performance in controlling time-related activity. The TPU contains a dedicated execution unit, a tri-level prioritized scheduler, data storage RAM, dual-time bases, and microcode ROM. The TPU controls 16 independent, orthogonal channels, each with an associated I/O pin, and is capable of performing any microcoded time function. Each channel contains dedicated hardware that allows input or output events to occur simultaneously on all channels.



TPU BLOCK

**Figure 12 TPU Block Diagram**

### 5.1 MC68332 and MC68332A Time Functions

The following paragraphs describe factory-programmed time functions implemented in standard and enhanced standard TPU microcode ROM. A complete description of the functions is beyond the scope of this summary. Refer to *Using the TPU Function Library and TPU Emulation Mode* (TPUPN00/D) as well as other TPU programming notes for more information about specific functions.

#### 5.1.1 Discrete Input/Output (DIO)

When a pin is used as a discrete input, a parameter indicates the current input level and the previous 15 levels of a pin. Bit 15, the most significant bit of the parameter, indicates the most recent state. Bit 14 indicates the next most recent state, and so on. The programmer can choose one of the three following conditions to update the parameter: 1) when a transition occurs, 2) when the CPU makes a request, or 3) when a rate specified in another parameter is matched. When a pin is used as a discrete output, it is set high or low only upon request by the CPU.

## 5.2.3 Queued Output Match (QOM)

QOM can generate single or multiple output match events from a table of offsets in parameter RAM. Loop modes allow complex pulse trains to be generated once, a specified number of times, or continuously. The function can be triggered by a link from another TPU channel. In addition, the reference time for the sequence of matches can be obtained from another channel. QOM can generate pulse-width modulated waveforms, including waveforms with high times of 0% or 100%. QOM also allows a TPU channel to be used as a discrete output pin.

## 5.2.4 Programmable Time Accumulator (PTA)

PTA accumulates a 32-bit sum of the total high time, low time, or period of an input signal over a programmable number of periods or pulses. The accumulation can start on a rising or falling edge. After the specified number of periods or pulses, the PTA generates an interrupt request and optionally generates links to other channels.

From 1 to 255 period measurements can be made and summed with the previous measurement(s) before the TPU interrupts the CPU, providing instantaneous or average frequency measurement capability, and the latest complete accumulation (over the programmed number of periods).

## 5.2.5 Multichannel Pulse Width Modulation (MCPWM)

MCPWM generates pulse-width modulated outputs with full 0% to 100% duty cycle range independent of other TPU activity. This capability requires two TPU channels plus an external gate for one PWM channel. (A simple one-channel PWM capability is supported by the QOM function.)

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.

## 5.2.6 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 CPU 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 ICTC function.

## 5.2.7 Universal Asynchronous Receiver/Transmitter (UART)

The UART function uses one or two TPU channels to provide asynchronous communications. Data word length is programmable from 1 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 could be implemented on the TPU.

## 5.2.8 Brushless Motor Commutation (COMM)

This function generates the phase commutation signals for a variety of brushless motors, including three-phase brushless direct current. 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. A CPU offset parameter is provided to allow all the switching angles to be advanced or retarded on the fly by the CPU. This feature is useful for torque maintenance at high speeds.

## 5.4 Parameter RAM

Parameter RAM occupies 256 bytes at the top of the TPU module address map. Channel parameters are organized as 128 16-bit words. However, only 100 words are actually implemented. The parameter RAM address map shows how parameter words are organized in memory.

**Table 23 TPU Parameter RAM Address Map**

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

—= Not Implemented

Y = M111, where M represents the logic state of the MM bit in the SIMCR.

## 5.5 TPU Registers

The TPU memory map contains three groups of registers:

- System Configuration Registers
- Channel Control and Status Registers
- Development Support and Test Verification Registers

### 5.5.1 System Configuration Registers

**TPUMCR** — TPU Module Configuration Register

**\$YFFE00**

15	14	13	12	11	10	9	8	7	6	5	4	3	0
STOP	TCR1P	TCR2P	EMU	T2CG	STF	SUPV	PSCK	0	0	0	0	IARB	0

RESET:

0 0 0 0 0 0 0 0 1 0 0 0 0 0

**STOP** — Stop Bit

- 0 = TPU operating normally
- 1 = Internal clocks shut down



## HSQR0 — Host Sequence Register 0

\$YFFE14

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

## HSQR1 — Host Sequence Register 1

\$YFFE16

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

## CH[15:0] — Encoded Host Sequence

The host sequence field selects the mode of operation for the time function selected on a given channel. The meaning of the host sequence bits depends on the time function specified.

## HSRR0 — Host Service Request Register 0

\$YFFE18

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

## HSRR1 — Host Service Request Register 1

\$YFFE1A

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

## CH[15:0] — Encoded Type of Host Service

The host service request field selects the type of host service request for the time function selected on a given channel. The meaning of the host service request bits depends on the time function specified. A host service request field cleared to %00 signals the host that service is completed by the microengine on that channel. The host can request service on a channel by writing the corresponding host service request field to one of three nonzero states. The CPU should monitor the host service request register until the TPU clears the service request to %00 before the CPU changes any parameters or issues a new service request to the channel.

## CPR0 — Channel Priority Register 0

\$YFFE1C

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

## CPR1 — Channel Priority Register 1

\$YFFE1E

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

## CH[15:0] — Encoded One of Three Channel Priority Levels

## MSTR — Master/Slave Mode Select

0 = QSPI is a slave device and only responds to externally generated serial data.

1 = QSPI is system master and can initiate transmission to external SPI devices.

MSTR configures the QSPI for either master or slave mode operation. This bit is cleared on reset and may only be written by the CPU.

## WOMQ — Wired-OR Mode for QSPI Pins

0 = Outputs have normal MOS drivers.

1 = Pins designated for output by DDRQS have open-drain drivers.

WOMQ allows the wired-OR function to be used on QSPI pins, regardless of whether they are used as general-purpose outputs or as QSPI outputs. WOMQ affects the QSPI pins regardless of whether the QSPI is enabled or disabled.

## BITS — Bits Per Transfer

In master mode, when BITSE in a command is set, the BITS field determines the number of data bits transferred. When BITSE is cleared, eight bits are transferred. Reserved values default to eight bits. BITSE is not used in slave mode.

The following table shows the number of bits per transfer.

BITS	Bits per Transfer
0000	16
0001	Reserved
0010	Reserved
0011	Reserved
0100	Reserved
0101	Reserved
0110	Reserved
0111	Reserved
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

## CPOL — Clock Polarity

0 = The inactive state value of SCK is logic level zero.

1 = The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with CPHA to produce a desired clock/data relationship between master and slave devices.

## CPHA — Clock Phase

0 = Data is captured on the leading edge of SCK and changed on the following edge of SCK.

1 = Data is changed on the leading edge of SCK and captured on the following edge of SCK.

CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices. CPHA is set at reset.

## SPBR — Serial Clock Baud Rate

The QSPI uses a modulus counter to derive SCK baud rate from the MCU system clock. Baud rate is selected by writing a value from 2 to 255 into the SPBR field. The following equation determines the

## IDLE — Idle-Line Detected Flag

- 0 = SCI receiver did not detect an idle-line condition.
- 1 = SCI receiver detected an idle-line condition.

IDLE is disabled when RWU in SCCR1 is set. IDLE is set when the SCI receiver detects the idle-line condition specified by ILT in SCCR1. If cleared, IDLE will not set again until after RDRF is set. RDRF is set when a break is received, so that a subsequent idle line can be detected.

## OR — Overrun Error Flag

- 0 = RDRF is cleared before new data arrives.
- 1 = RDRF is not cleared before new data arrives.

OR is set when a new byte is ready to be transferred from the receive serial shifter to the RDR, and RDRF is still set. Data transfer is inhibited until OR is cleared. Previous data in RDR remains valid, but data received during overrun condition (including the byte that set OR) is lost.

## NF — Noise Error Flag

- 0 = No noise detected on the received data
- 1 = Noise occurred on the received data

NF is set when the SCI receiver detects noise on a valid start bit, on any data bit, or on a stop bit. It is not set by noise on the idle line or on invalid start bits. Each bit is sampled three times. If none of the three samples are the same logic level, the majority value is used for the received data value, and NF is set. NF is not set until an entire frame is received and RDRF is set.

## FE — Framing Error Flag

- 0 = No framing error on the received data.
- 1 = Framing error or break occurred on the received data.

FE is set when the SCI receiver detects a zero where a stop bit was to have occurred. FE is not set until the entire frame is received and RDRF is set. A break can also cause FE to be set. It is possible to miss a framing error if RXD happens to be at logic level one at the time the stop bit is expected.

## PF — Parity Error Flag

- 0 = No parity error on the received data
- 1 = Parity error occurred on the received data

PF is set when the SCI receiver detects a parity error. PF is not set until the entire frame is received and RDRF is set.

## SCDR — SCI Data Register

**\$YFFC0E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0

RESET:

0 0 0 0 0 0 0 U U U U U U U U U

SCDR contains two data registers at the same address. Receive data register (RDR) is a read-only register that contains data received by the SCI. The data comes into the receive serial shifter and is transferred to RDR. Transmit data register (TDR) is a write-only register that contains data to be transmitted. The 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.

## 7 Standby RAM with TPU Emulation RAM

The TPURAM module contains a 2-Kbyte array of fast (two bus cycle) static RAM, which is especially useful for system stacks and variable storage. Alternately, it can be used by the TPU as emulation RAM for new timer algorithms.

### 7.1 Overview

The TPURAM can be mapped to any 4-Kbyte boundary in the address map, but must not overlap the module control registers. (Overlap makes the registers inaccessible.) Data can be read or written in bytes, word, or long words. TPURAM responds to both program and data space accesses. 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, the TPURAM contents are maintained by power on standby voltage pin  $V_{STBY}$ . Power switching between sources is automatic.

Access to the TPURAM array is controlled by the RASP field in TRAMMCR. This field can be encoded so that TPURAM responds to both program and data space accesses. This allows code to be executed from TPURAM, and permits the use of program counter relative addressing mode for operand fetches from the array.

An address map of the TPURAM control registers follows. All TPURAM control registers are located in supervisor data space.

**Table 28 TPURAM Control Register Address Map**

Access	Address	15	8	7	0
S	\$YFFB00	TPURAM MODULE CONFIGURATION REGISTER (TRAMMCR)			
S	\$YFFB02	TPURAM TEST REGISTER (TRAMTST)			
S	\$YFFB04	TPURAM BASE ADDRESS REGISTER (TRAMBAR)			
	\$YFFB06– \$YFFB3F	NOT USED			

Y = M111, where M is the logic state of the MM bit in the SIMCR.

### 7.2 TPURAM Register Block

There are three TPURAM control registers: the RAM module configuration register (TRAMMCR), the RAM test register (TRAMTST), and the RAM array base address registers (TRAMBAR).

There is an 8-byte minimum register block size for the module. Unimplemented register addresses are read as zeros, and writes have no effect.

### 7.3 TPURAM Registers

**TRAMMCR** —TPURAM Module Configuration Register

**\$YFFB00**

15	14	13	12	11	10	9	8	7	0
STOP	0	0	0	0	0	0	RASP	NOT USED	

RESET:

0 0 0 0 0 0 0 1

**TSTOP** —Stop Control

0 = RAM array operates normally.

1 = RAM array enters low-power stop mode.

This bit controls whether the RAM array is in stop mode or normal operation. Reset state is zero, for normal operation. In stop mode, the array retains its contents, but cannot be read or written by the CPU.

## 8 Summary of Changes

This is a partial revision. Most of the publication remains the same, but the following changes were made to improve it. Typographical errors that do not affect content are not annotated. This document has also been reformatted for use on the web.

Pages 2-3	New Ordering Information included.
Page 6	New block diagram drawn.
Page 7	New 132-pin assignment diagram drawn.
Page 8	New 144-pin assignment diagram drawn.
Page 9	New address map drawn.
Pages 10-14	Added Signal Description section.
Pages 15-47	Expanded and revised SIM section. Made all register diagrams and bit mnemonics consistent. Incorporated new information concerning the system clock, resets, interrupts, and chip-selects circuits.
Page 48-56	Expanded and revised CPU section. Made all register diagrams and bit mnemonics consistent. Revised instruction set summary information.
Page 57-70	Expanded and revised TPU section. Made all register diagrams and bit mnemonics consistent. Revised time functions information to include both MC68332A and MC68332G microcode ROM applications.
Page 71-92	Expanded and revised QSM section. Made all register diagrams and bit mnemonics consistent. Added information concerning SPI and SCI operation.
Page 93-95	Revised Standby RAM with TPU Emulation RAM section. Made all register diagrams and bit mnemonics consistent.