



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	20MHz
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/mc68332gceh20">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68332gceh20</a>

Table 1 Ordering Information (Continued)

Package Type	TPU Type	Temperature	Frequency (MHz)	Package Order Quantity	Order Number
144-Pin QFP	Motion Control	-40 to +85 °C	16 MHz	2 pc tray	SPAKMC332GCFV16
				44 pc tray	MC68332GCFV16
			20 MHz	2 pc tray	SPAKMC332GCFV20
				44 pc tray	MC68332GCFV20
		-40 to +105 °C	16 MHz	2 pc tray	SPAKMC332GVFV16
				44 pc tray	MC68332GVFV16
			20 MHz	2 pc tray	SPAKMC332GVFV20
				44 pc tray	MC68332GVFV20
		-40 to +125 °C	16 MHz	2 pc tray	SPAKMC332GMFV16
				44 pc tray	MC68332GMFV16
			20 MHz	2 pc tray	SPAKMC332GMFV20
				44 pc tray	MC68332GMFV20
	Standard	-40 to +85 °C	16 MHz	2 pc tray	SPAKMC332CFV16
				44 pc tray	MC68332CFV16
			20 MHz	2 pc tray	SPAKMC332CFV20
				44 pc tray	MC68332CFV20
		-40 to +105 °C	16 MHz	2 pc tray	SPAKMC332V16
				44 pc tray	MC68332V16
			20 MHz	2 pc tray	SPAKMC332V20
				44 pc tray	MC68332V20
		-40 to +125 °C	16 MHz	2 pc tray	SPAKMC332MFV16
				44 pc tray	MC68332MFV16
			20 MHz	2 pc tray	SPAKMC332MFV20
				44 pc tray	MC68332MFV20
	Std w/enhanced PPWA	-40 to +85 °C	16 MHz	2 pc tray	SPAKMC332ACFV16
				44 pc tray	MC68332ACFV16
			20 MHz	2 pc tray	SPAKMC332ACFV20
				44 pc tray	MC68332ACFV20
		-40 to +105 °C	16 MHz	2 pc tray	SPAKMC332AVFV16
				44 pc tray	MC68332AVFV16
			20 MHz	2 pc tray	SPAKMC332AVFC20
				44 pc tray	MC68332AVFV20
		-40 to +125 °C	16 MHz	2 pc tray	SPAKMC332AMFV16
				44 pc tray	MC68332AMFV16
			20 MHz	2 pc tray	SPAKMC332AMFV20
				44 pc tray	MC68332AMFV20

## 1.3 Pin Assignments

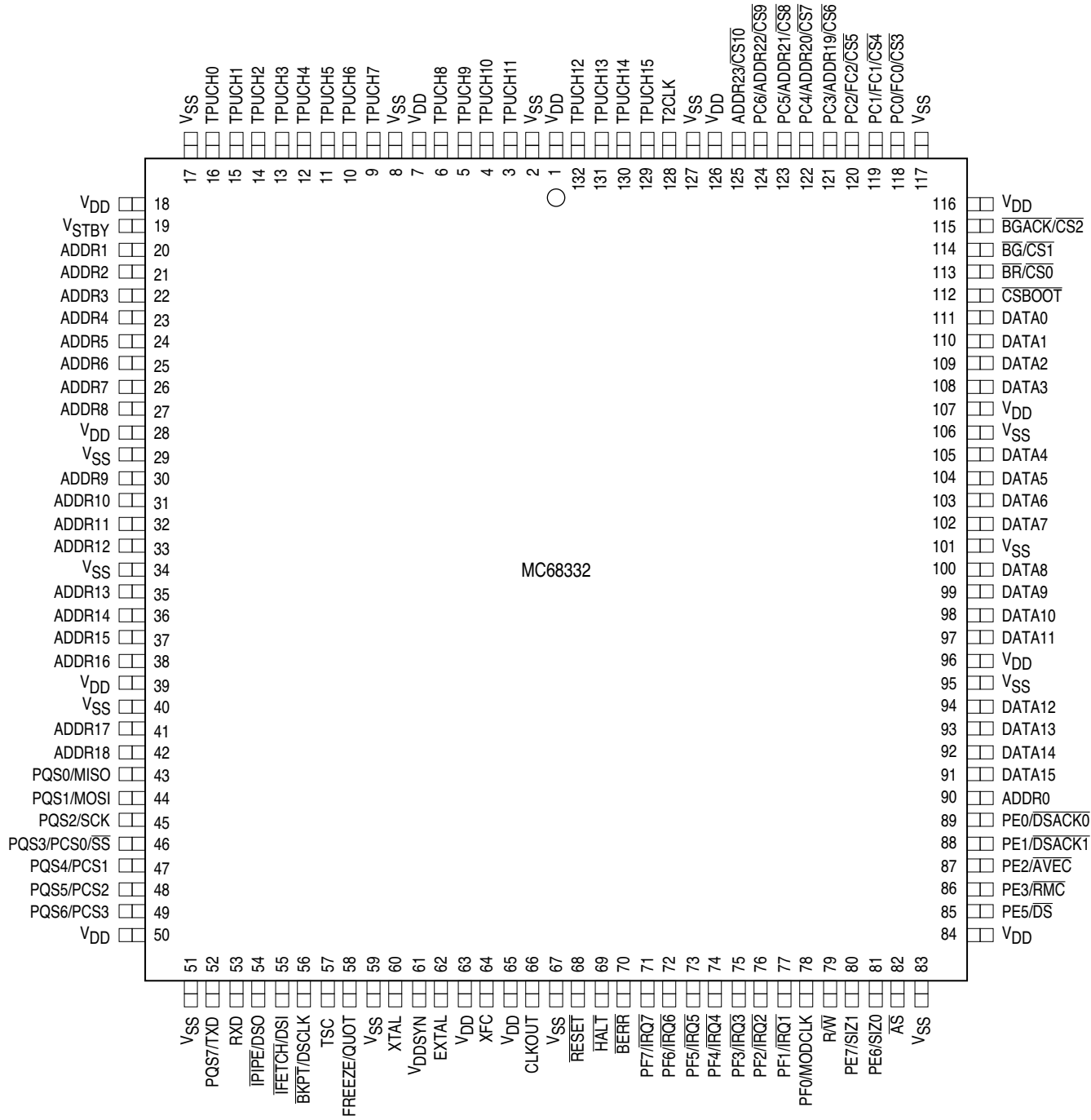
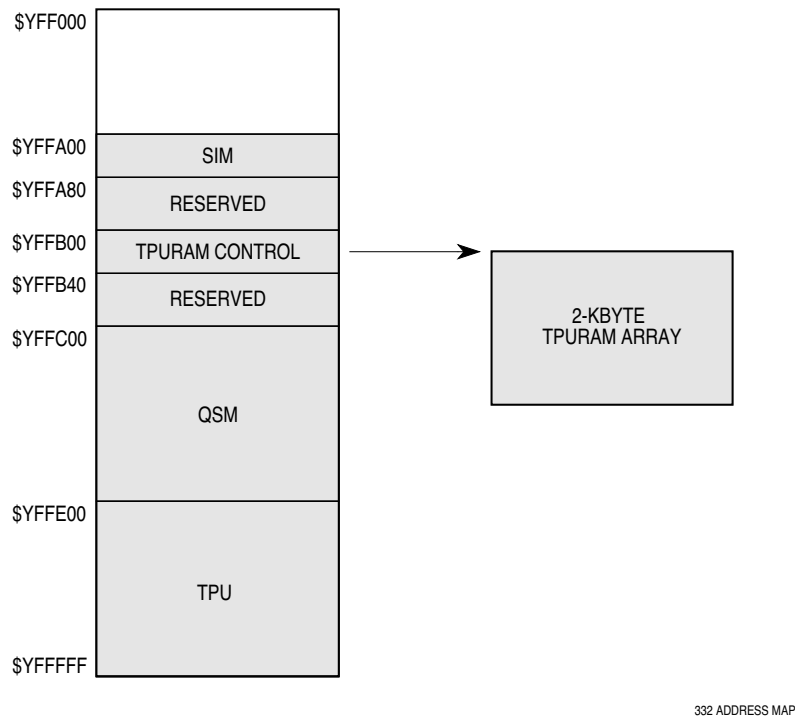


Figure 2 MC68332 132-Pin QFP Pin Assignments

**1.4 Address Map**

The following figure is a map of the MCU internal addresses. The RAM array is positioned by the base address registers in the associated RAM control block. Unimplemented blocks are mapped externally.



**Figure 4 MCU Address Map**

**1.5 Intermodule Bus**

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

**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)			

Table 7 SIM Address Map (Continued)

Access	Address	15	8 7	0
S	\$YFFA56	CHIP-SELECT OPTION 2 (CSOR2)		
S	\$YFFA58	CHIP-SELECT BASE 3 (CSBAR3)		
S	\$YFFA5A	CHIP-SELECT OPTION 3 (CSOR3)		
S	\$YFFA5C	CHIP-SELECT BASE 4 (CSBAR4)		
S	\$YFFA5E	CHIP-SELECT OPTION 4 (CSOR4)		
S	\$YFFA60	CHIP-SELECT BASE 5 (CSBAR5)		
S	\$YFFA62	CHIP-SELECT OPTION 5 (CSOR5)		
S	\$YFFA64	CHIP-SELECT BASE 6 (CSBAR6)		
S	\$YFFA66	CHIP-SELECT OPTION 6 (CSOR6)		
S	\$YFFA68	CHIP-SELECT BASE 7 (CSBAR7)		
S	\$YFFA6A	CHIP-SELECT OPTION 7 (CSOR7)		
S	\$YFFA6C	CHIP-SELECT BASE 8 (CSBAR8)		
S	\$YFFA6E	CHIP-SELECT OPTION 8 (CSOR8)		
S	\$YFFA70	CHIP-SELECT BASE 9 (CSBAR9)		
S	\$YFFA72	CHIP-SELECT OPTION 9 (CSOR9)		
S	\$YFFA74	CHIP-SELECT BASE 10 (CSBAR10)		
S	\$YFFA76	CHIP-SELECT OPTION 10 (CSOR10)		
	\$YFFA78	NOT USED		NOT USED
	\$YFFA7A	NOT USED		NOT USED
	\$YFFA7C	NOT USED		NOT USED
	\$YFFA7E	NOT USED		NOT USED

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

## 3.2 System Configuration and Protection

This functional block provides configuration control for the entire MCU. It also performs interrupt arbitration, bus monitoring, and system test functions. MCU system protection includes a bus monitor, a HALT monitor, a spurious interrupt monitor, and a software watchdog timer. These functions have been made integral to the microcontroller to reduce the number of external components in a complete control system.

## 3.3.3 Clock Control

The clock control circuits determine system clock frequency and clock operation under special circumstances, such as following loss of synthesizer reference or during low-power operation. Clock source is determined by the logic state of the MODCLK pin during reset.

### SYNCR — Clock Synthesizer Control Register

\$YFFA04

15	14	13		8	7	6	5	4	3	2	1	0
W	X	Y			EDIV	0	0	SLIMP	SLOCK	RSTEN	STSIM	STEXT

RESET:

0 0 1 1 1 1 1 1 0 0 0 U U 0 0 0

When the on-chip clock synthesizer is used, system clock frequency is controlled by the bits in the upper byte of SYNCR. Bits in the lower byte show status of or control operation of internal and external clocks. The SYNCR can be read or written only when the CPU is operating at the supervisor privilege level.

#### W — Frequency Control (VCO)

This bit controls a prescaler tap in the synthesizer feedback loop. Setting the bit increases the VCO speed by a factor of four. VCO relock delay is required.

#### X — Frequency Control Bit (Prescale)

This bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting the bit doubles clock speed without changing the VCO speed. There is no VCO relock delay.

#### Y[5:0] — Frequency Control (Counter)

The Y field controls the modulus down counter in the synthesizer feedback loop, causing it to divide by a value of Y + 1. Values range from 0 to 63. VCO relock delay is required.

#### EDIV — E Clock Divide Rate

0 = ECLK frequency is system clock divided by 8.

1 = ECLK frequency is system clock divided by 16.

ECLK is an external M6800 bus clock available on pin ADDR23. Refer to **3.5 Chip Selects** for more information.

#### SLIMP — Limp Mode Flag

0 = External crystal is VCO reference.

1 = Loss of crystal reference.

When the on-chip synthesizer is used, loss of reference frequency causes SLIMP to be set. The VCO continues to run using the base control voltage. Maximum limp frequency is maximum specified system clock frequency. X-bit state affects limp frequency.

#### SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.

1 = VCO has locked on the desired frequency (or system clock is external).

The MCU maintains reset state until the synthesizer locks, but SLOCK does not indicate synthesizer lock status until after the user writes to SYNCR.

#### RSTEN — Reset Enable

0 = Loss of crystal causes the MCU to operate in limp mode.

1 = Loss of crystal causes system reset.

#### STSIM — Stop Mode SIM Clock

0 = When LPSTOP is executed, the SIM clock is driven from the crystal oscillator and the VCO is turned off to conserve power.

1 = When LPSTOP is executed, the SIM clock is driven from the VCO.

#### STEXT — Stop Mode External Clock

0 = When LPSTOP is executed, the CLKOUT signal is held negated to conserve power.

1 = When LPSTOP is executed, the CLKOUT signal is driven from the SIM clock, as determined by the state of the STSIM bit.

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

## 3.4.11 Misaligned Operands

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

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

## 3.4.12 Operand Transfer Cases

The following table summarizes how operands are aligned for various types of transfers. OPn entries are portions of a requested operand that are read or written during a bus cycle and are defined by SIZ1, SIZ0, and ADDR0 for that bus cycle.

**Table 11 Operand Alignment**

Transfer Case	SIZ1	SIZ0	ADDR0	DSACK1	DSACK0	DATA [15:8]	DATA [7:0]
Byte to 8-Bit Port (Even/Odd)	0	1	X	1	0	OP0	(OP0)
Byte to 16-Bit Port (Even)	0	1	0	0	X	OP0	(OP0)
Byte to 16-Bit Port (Odd)	0	1	1	0	X	(OP0)	OP0
Word to 8-Bit Port (Aligned)	1	0	0	1	0	OP0	(OP1)
Word to 8-Bit Port (Misaligned) <sup>3</sup>	1	0	1	1	0	OP0	(OP0)
Word to 16-Bit Port (Aligned)	1	0	0	0	X	OP0	OP1
Word to 16-Bit Port (Misaligned) <sup>3</sup>	1	0	1	0	X	(OP0)	OP0
3 Byte to 8-Bit Port (Aligned) <sup>2</sup>	1	1	0	1	0	OP0	(OP1)
3 Byte to 8-Bit Port (Misaligned) <sup>2, 3</sup>	1	1	1	1	0	OP0	(OP0)
3 Byte to 16-Bit Port (Aligned) <sup>2</sup>	1	1	0	0	X	OP0	OP1
3 Byte to 16-Bit Port (Misaligned) <sup>2, 3</sup>	1	1	1	0	X	(OP0)	OP0
Long Word to 8-Bit Port (Aligned)	0	0	0	1	0	OP0	(OP1)
Long Word to 8-Bit Port (Misaligned) <sup>3</sup>	1	0	1	1	0	OP0	(OP0)
Long Word to 16-Bit Port (Aligned)	0	0	0	0	X	OP0	OP1
Long Word to 16-Bit Port (Misaligned) <sup>3</sup>	1	0	1	0	X	(OP0)	OP0

### NOTES:

1. Operands in parentheses are ignored by the CPU32 during read cycles.
2. Three-byte transfer cases occur only as a result of a long word to byte transfer.
3. The CPU32 does not support misaligned word or long-word transfers.

## 3.5 Chip Selects

Typical microcontrollers require additional hardware to provide external chip-select signals. Twelve independently programmable chip selects provide fast two-cycle access to external memory or peripherals. Address block sizes of 2 Kbytes to 1 Mbyte can be selected.



$\overline{\text{AVEC}}$  — Autovector Enable

0 = External interrupt vector enabled

1 = Autovector enabled

This field selects one of two methods of acquiring the interrupt vector during the interrupt acknowledge cycle. It is not usually used in conjunction with a chip-select pin.

If the chip select is configured to trigger on an interrupt acknowledge cycle (SPACE = 00) and the  $\overline{\text{AVEC}}$  field is set to one, the chip select automatically generates an  $\overline{\text{AVEC}}$  in response to the interrupt cycle. Otherwise, the vector must be supplied by the requesting device.

The  $\overline{\text{AVEC}}$  bit must not be used in synchronous mode, as autovector response timing can vary because of ECLK synchronization.

### 3.5.5 Port C Data Register

Bit values in port C determine the state of chip-select pins used for discrete output. When a pin is assigned as a discrete output, the value in this register appears at the output. This is a read/write register. Bit 7 is not used. Writing to this bit has no effect, and it always returns zero when read.

**PORTC** — Port C Data Register

**\$YFFA41**

15	8	7	6	5	4	3	2	1	0
NOT USED	0	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
RESET:	0	1	1	1	1	1	1	1	1

### 3.6 General-Purpose Input/Output

SIM pins can be configured as two general-purpose I/O ports, E and F. The following paragraphs describe registers that control the ports.

**PORTE0, PORTE1** —Port E Data Register

**\$YFFA11, \$YFFA13**

15	8	7	6	5	4	3	2	1	0
NOT USED	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
RESET:	U	U	U	U	U	U	U	U	U

A write to the port E data register is stored in the internal data latch and, if any port E pin is configured as an output, the value stored for that bit is driven on the pin. A read of the port E data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register.

The port E data register is a single register that can be accessed in two locations. When accessed at \$YFFA11, the register is referred to as PORTE0; when accessed at \$YFFA13, the register is referred to as PORTE1. The register can be read or written at any time. It is unaffected by reset.

**DDRE** — Port E Data Direction Register

**\$YFFA15**

15	8	7	6	5	4	3	2	1	0
NOT USED	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	
RESET:	0	0	0	0	0	0	0	0	0

The bits in this register control the direction of the pin drivers when the pins are configured as I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input. This register can be read or written at any time.

mask lower-priority interrupts during exception processing, and it is decoded by modules that have requested interrupt service to determine whether the current interrupt acknowledge cycle pertains to them.

Modules that have requested interrupt service decode the IP value placed on the address bus at the beginning of the interrupt acknowledge cycle, and if their requests are at the specified IP level, respond to the cycle. Arbitration between simultaneous requests of the same priority is performed by means of serial contention between module interrupt arbitration (IARB) field bit values.

Each module that can make an interrupt service request, including the SIM, has an IARB field in its configuration register. An IARB field can be assigned a value from %0001 (lowest priority) to %1111 (highest priority). A value of %0000 in an IARB field causes the CPU to process a spurious interrupt exception when an interrupt from that module is recognized.

Because the EBI manages external interrupt requests, the SIM IARB value is used for arbitration between internal and external interrupt requests. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000. Initialization software must assign different IARB values in order to implement an arbitration scheme.

Each module must have a unique IARB value. When two or more IARB fields have the same nonzero value, the CPU interprets multiple vector numbers simultaneously, with unpredictable consequences.

Arbitration must always take place, even when a single source requests service. This point is important for two reasons: the CPU interrupt acknowledge cycle is not driven on the external bus unless the SIM wins contention, and failure to contend causes an interrupt acknowledge bus cycle to be terminated by a bus error, which causes a spurious interrupt exception to be taken.

When arbitration is complete, the dominant module must place an interrupt vector number on the data bus and terminate the bus cycle. In the case of an external interrupt request, because the interrupt acknowledge cycle is transferred to the external bus, an external device must decode the mask value and respond with a vector number, then generate bus cycle termination signals. If the device does not respond in time, a spurious interrupt exception is taken.

The periodic interrupt timer (PIT) in the SIM can generate internal interrupt requests of specific priority at predetermined intervals. By hardware convention, PIT interrupts are serviced before external interrupt service requests of the same priority. Refer to 3.2.7 Periodic Interrupt Timer for more information.

## 3.8.2 Interrupt Processing Summary

A summary of the interrupt processing sequence follows. When the sequence begins, a valid interrupt service request has been detected and is pending.

- A. The CPU finishes higher priority exception processing or reaches an instruction boundary.
- B. Processor state is stacked. The contents of the status register and program counter are saved.
- C. The interrupt acknowledge cycle begins:
  1. FC[2:0] are driven to %111 (CPU space) encoding.
  2. The address bus is driven as follows. ADDR[23:20] = %1111; ADDR[19:16] = %1111, which indicates that the cycle is an interrupt acknowledge CPU space cycle; ADDR[15:4] = %111111111111; ADDR[3:1] = the level of the interrupt request being acknowledged; and ADDR0 = %1.
  3. Request priority level is latched into the IP field in the status register from the address bus.
- D. Modules or external peripherals that have requested interrupt service decode the request level in ADDR[3:1]. If the request level of at least one interrupting module or device is the same as the value in ADDR[3:1], interrupt arbitration contention takes place. When there is no contention, the spurious interrupt monitor asserts BERR, and a spurious interrupt exception is processed.
- E. After arbitration, the interrupt acknowledge cycle can be completed in one of three ways:

1. The dominant interrupt source supplies a vector number and  $\overline{DSACK}$  signals appropriate to the access. The CPU32 acquires the vector number.
  2. The  $\overline{AVEC}$  signal is asserted (the signal can be asserted by the dominant interrupt source or the pin can be tied low), and the CPU32 generates an autovector number corresponding to interrupt priority.
  3. The bus monitor asserts  $\overline{BERR}$  and the CPU32 generates the spurious interrupt vector number.
- F. The vector number is converted to a vector address.
- G. The content of the vector address is loaded into the PC, and the processor transfers control to the exception handler routine.

### 3.9 Factory Test Block

The test submodule supports scan-based testing of the various MCU modules. It is integrated into the SIM to support production testing.

Test submodule registers are intended for Motorola use. Register names and addresses are provided to indicate that these addresses are occupied.

<b>SIMTR</b> —System Integration Test Register	<b>\$YFFA02</b>
<b>SIMTRE</b> —System Integration Test Register (E Clock)	<b>\$YFFA08</b>
<b>TSTMSRA</b> —Master Shift Register A	<b>\$YFFA30</b>
<b>TSTMSRB</b> —Master Shift Register B	<b>\$YFFA32</b>
<b>TSTSC</b> —Test Module Shift Count	<b>\$YFFA34</b>
<b>TSTRC</b> —Test Module Repetition Count	<b>\$YFFA36</b>
<b>CREG</b> —Test Module Control Register	<b>\$YFFA38</b>
<b>DREG</b> —Test Module Distributed Register	<b>\$YFFA3A</b>

Table 20 Instruction Set Summary(Continued)

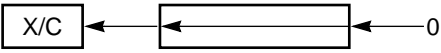
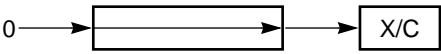
Instruction	Syntax	Operand Size	Operation
DBcc	Dn, label	16	If condition false, then Dn - 1 ⇒ PC; if Dn ≠ (-1), then PC + d ⇒ PC
DIVS/DIVU	<ea>, Dn	32/16 ⇒ 16 : 16	Destination / Source ⇒ Destination (signed or unsigned)
DIVSL/DIVUL	<ea>, Dr : Dq <ea>, Dq <ea>, Dr : Dq	64/32 ⇒ 32 : 32 32/32 ⇒ 32 32/32 ⇒ 32 : 32	Destination / Source ⇒ Destination (signed or unsigned)
EOR	Dn, <ea>	8, 16, 32	Source ⊕ Destination ⇒ Destination
EORI	# <data>, <ea>	8, 16, 32	Data ⊕ Destination ⇒ Destination
EORI to CCR	# <data>, CCR	8	Source ⊕ CCR ⇒ CCR
EORI to SR <sup>1</sup>	# <data>, SR	16	Source ⊕ SR ⇒ SR
EXG	Rn, Rn	32	Rn ⇒ Rn
EXT	Dn Dn	8 ⇒ 16 16 ⇒ 32	Sign extended Destination ⇒ Destination
EXTB	Dn	8 ⇒ 32	Sign extended Destination ⇒ Destination
ILLEGAL	none	none	SSP - 2 ⇒ SSP; vector offset ⇒ (SSP); SSP - 4 ⇒ SSP; PC ⇒ (SSP); SSP - 2 ⇒ SSP; SR ⇒ (SSP); Illegal instruction vector address ⇒ PC
JMP	í	none	Destination ⇒ PC
JSR	í	none	SP - 4 ⇒ SP; PC ⇒ (SP); destination ⇒ PC
LEA	<ea>, An	32	<ea> ⇒ An
LINK	An, # d	16, 32	SP - 4 ⇒ SP, An ⇒ (SP); SP ⇒ An, SP + d ⇒ SP
LPSTOP <sup>1</sup>	# <data>	16	Data ⇒ SR; interrupt mask ⇒ EBI; STOP
LSL	Dn, Dn # <data>, Dn í	8, 16, 32 8, 16, 32 16	
LSR	Dn, Dn # <data>, Dn í	8, 16, 32 8, 16, 32 16	
MOVE	<ea>, <ea>	8, 16, 32	Source ⇒ Destination
MOVEA	<ea>, An	16, 32 ⇒ 32	Source ⇒ Destination
MOVEA <sup>1</sup>	USP, An An, USP	32 32	USP ⇒ An An ⇒ USP
MOVE from CCR	CCR, <ea>	16	CCR ⇒ Destination
MOVE to CCR	<ea>, CCR	16	Source ⇒ CCR
MOVE from SR <sup>1</sup>	SR, <ea>	16	SR ⇒ Destination
MOVE to SR <sup>1</sup>	<ea>, SR	16	Source ⇒ SR
MOVE USP <sup>1</sup>	USP, An An, USP	32 32	USP ⇒ An An ⇒ USP
MOVEC <sup>1</sup>	Rc, Rn Rn, Rc	32 32	Rc ⇒ Rn Rn ⇒ Rc
MOVEM	list, <ea> <ea>, list	16, 32 16, 32 ⇒ 32	Listed registers ⇒ Destination Source ⇒ Listed registers
MOVEP	Dn, (d16, An)  (d16, An), Dn	16, 32	Dn [31 : 24] ⇒ (An + d); Dn [23 : 16] ⇒ (An + d + 2); Dn [15 : 8] ⇒ (An + d + 4); Dn [7 : 0] ⇒ (An + d + 6)  (An + d) ⇒ Dn [31 : 24]; (An + d + 2) ⇒ Dn [23 : 16]; (An + d + 4) ⇒ Dn [15 : 8]; (An + d + 6) ⇒ Dn [7 : 0]
MOVEQ	#<data>, Dn	8 ⇒ 32	Immediate data ⇒ Destination

Table 20 Instruction Set Summary(Continued)

Instruction	Syntax	Operand Size	Operation
SWAP	Dn	16	
TAS	í	8	Destination Tested Condition Codes bit 7 of Destination
TBLS/TBLU	<ea>, Dn Dym : Dyn, Dn	8, 16, 32	$\text{Dyn} - \text{Dym} \Rightarrow \text{Temp}$ $(\text{Temp} * \text{Dn} [7 : 0]) \Rightarrow \text{Temp}$ $(\text{Dym} * 256) + \text{Temp} \Rightarrow \text{Dn}$
TBLSN/TBLUN	<ea>, Dn Dym : Dyn, Dn	8, 16, 32	$\text{Dyn} - \text{Dym} \Rightarrow \text{Temp}$ $(\text{Temp} * \text{Dn} [7 : 0]) / 256 \Rightarrow \text{Temp}$ $\text{Dym} + \text{Temp} \Rightarrow \text{Dn}$
TRAP	#<data>	none	$\text{SSP} - 2 \Rightarrow \text{SSP}$ ; format/vector offset $\Rightarrow (\text{SSP})$ ; $\text{SSP} - 4 \Rightarrow \text{SSP}$ ; PC $\Rightarrow (\text{SSP})$ ; SR $\Rightarrow (\text{SSP})$ ; vector address $\Rightarrow \text{PC}$
TRAPcc	none #<data>	none 16, 32	If cc true, then TRAP exception
TRAPV	none	none	If V set, then overflow TRAP exception
TST	í	8, 16, 32	Source – 0, to set condition codes
UNLK	An	32	$\text{An} \Rightarrow \text{SP}$ ; $(\text{SP}) \Rightarrow \text{An}$ , $\text{SP} + 4 \Rightarrow \text{SP}$

1. Privileged instruction.

## 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.

## EMU — Emulation Control

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

- 0 = TPU and TPURAM not in emulation mode
- 1 = TPU and TPURAM in emulation mode

## T2CG — TCR2 Clock/Gate Control

When the T2CG bit is set, the external TCR2 pin functions as a gate of the DIV8 clock (the TPU system clock divided by 8). In this case, when the external TCR2 pin is low, the DIV8 clock is blocked, preventing it from incrementing TCR2. When the external TCR2 pin is high, TCR2 is incremented at the frequency of the DIV8 clock. When T2CG is cleared, an external clock from the TCR2 pin, which has been synchronized and fed through a digital filter, increments TCR2.

- 0 = TCR2 pin used as clock source for TCR2
- 1 = TCR2 pin used as gate of DIV8 clock for TCR2

## STF — Stop Flag

- 0 = TPU operating
- 1 = TPU stopped (STOP bit has been asserted)

## SUPV — Supervisor Data Space

- 0 = Assignable registers are unrestricted (FC2 is ignored)
- 1 = Assignable registers are restricted (FC2 is decoded)

## PSCK — Prescaler Clock

- 0 = System clock/32 is input to TCR1 prescaler
- 1 = System clock/4 is input to TCR1 prescaler

## IARB — Interrupt Arbitration Identification Number

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. Refer to the **3.8 Interrupts** for more information.

## TICR — TPU Interrupt Configuration Register

**\$YFFE08**

15	11	10	8	7	4	3	0
NOT USED		CIRL		CIBV		NOT USED	
RESET:							
		0	0	0	0	0	0

## CIRL — Channel Interrupt Request Level

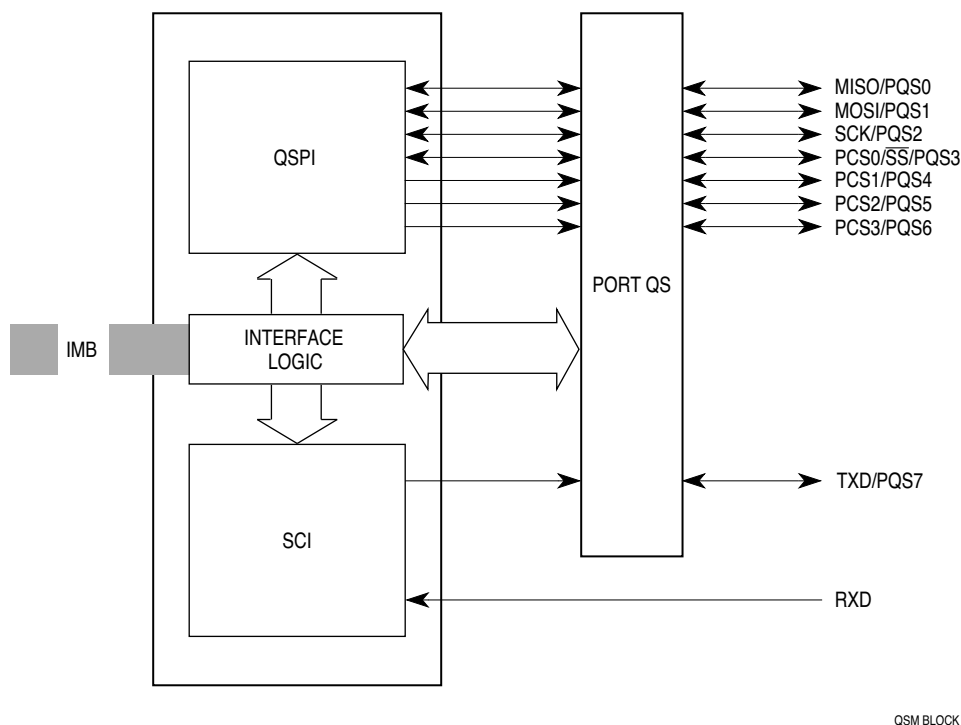
The interrupt request level for all channels is specified by this 3-bit encoded field. Level seven for this field indicates a nonmaskable interrupt; level zero indicates that all channel interrupts are disabled.

## CIBV — Channel Interrupt Base Vector

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

## 6 Queued Serial Module

The QSM contains two serial interfaces, the queued serial peripheral interface (QSPI) and the serial communication interface (SCI).



**Figure 13 QSM Block Diagram**

### 6.1 Overview

The QSPI provides easy peripheral expansion or interprocessor communication through a full-duplex, synchronous, three-line bus: data in, data out, and a serial clock. Four programmable peripheral chip-select pins provide addressability for up to 16 peripheral devices. A self-contained RAM queue allows up to 16 serial transfers of 8 to 16 bits each, or transmission of a 256-bit data stream without CPU intervention. A special wraparound mode supports continuous sampling of a serial peripheral, with automatic QSPI RAM updating, which makes the interface to A/D converters more efficient.

The SCI provides a standard nonreturn 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 64 to 524 kbaud with a 16.78-MHz system clock, or 110 to 655 kbaud with a 20.97-MHz system clock. Word length of either 8 or 9 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. Wakeup functions allow the CPU to run uninterrupted until meaningful data is available.

An address map of the QSM is shown below.



## 6.2 Address Map

The “Access” column in the QSM address map below indicates which registers are accessible only at the supervisor privilege level and which can be assigned to either the supervisor or user privilege level, according to the value of the SUPV bit in the QSMCR.

**Table 24 QSM Address Map**

Access	Address	15	8	7	0
S	\$YFFC00	QSM MODULE CONFIGURATION (QSMCR)			
S	\$YFFC02	QSM TEST (QTEST)			
S	\$YFFC04	QSM INTERRUPT LEVEL (QILR)		QSM INTERRUPT VECTOR (QIVR)	
S/U	\$YFFC06	NOT USED			
S/U	\$YFFC08	SCI CONTROL 0 (SCCR0)			
S/U	\$YFFC0A	SCI CONTROL 1 (SCCR1)			
S/U	\$YFFC0C	SCI STATUS (SCSR)			
S/U	\$YFFC0E	SCI DATA (SCDR)			
S/U	\$YFFC10	NOT USED			
S/U	\$YFFC12	NOT USED			
S/U	\$YFFC14	NOT USED		PQS DATA (PORTQS)	
S/U	\$YFFC16	PQS PIN ASSIGNMENT (PQSPAR)		PQS DATA DIRECTION (DDRQS)	
S/U	\$YFFC18	SPI CONTROL 0 (SPCR0)			
S/U	\$YFFC1A	SPI CONTROL 1 (SPCR1)			
S/U	\$YFFC1C	SPI CONTROL 2 (SPCR2)			
S/U	\$YFFC1E	SPI CONTROL 3 (SPCR3)		SPI STATUS (SPSR)	
S/U	\$YFFC20– \$YFFCFF	NOT USED			
S/U	\$YFFD00– \$YFFD1F	RECEIVE RAM (RR[0:F])			
S/U	\$YFFD20– \$YFFD3F	TRANSMIT RAM (TR[0:F])			
S/U	\$YFFD40– \$YFFD4F	COMMAND RAM (CR[0:F])			

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

## QIVR — QSM Interrupt Vector Register

\$YFFC05

15	8	7									0				
QILR								INTV							
RESET:															
								0	0	0	0	1	1	1	1

At reset, QIVR is initialized to \$0F, which corresponds to the uninitialized interrupt vector in the exception table. This vector is selected until QIVR is written. A user-defined vector (\$40–\$FF) should be written to QIVR during QSM initialization.

After initialization, QIVR determines which two vectors in the exception vector table are to be used for QSM interrupts. The QSPI and SCI submodules have separate interrupt vectors adjacent to each other. Both submodules use the same interrupt vector with the least significant bit (LSB) determined by the submodule causing the interrupt.

The value of INTV0 used during an interrupt-acknowledge cycle is supplied by the QSM. During an interrupt-acknowledge cycle, INTV[7:1] are driven on DATA[7:1] IMB lines. DATA0 is negated for an SCI interrupt and asserted for a QSPI interrupt. Writes to INTV0 have no meaning or effect. Reads of INTV0 return a value of one.

### 6.4.2 Pin Control Registers

The QSM uses nine pins, eight of which form a parallel port (PORTQS) on the MCU. Although these pins are used by the serial subsystems, any pin can alternately be assigned as general-purpose I/O on a pin-by-pin basis.

Pins used for general-purpose I/O must not be assigned to the QSPI by register PQSPAR. To avoid driving incorrect data, the first byte to be output must be written before DDRQS is configured. DDRQS must then be written to determine the direction of data flow and to output the value contained in register PORTQS. Subsequent data for output is written to PORTQS.

## PORTQS — Port QS Data Register

\$YFFC14

15	8	7	6	5	4	3	2	1	0
NOT USED		PQS7	PQS6	PQS5	PQS4	PQS3	PQS2	PQS1	PQS0
RESET:									
		0	0	0	0	0	0	0	0

PORTQS latches I/O data. Writes drive pins defined as outputs. Reads return data present on the pins. To avoid driving undefined data, first write a byte to PORTQS, then configure DDRQS.

## PQSPAR — PORT QS Pin Assignment Register

\$YFFC16

## DDRQS — PORT QS Data Direction Register

\$YFFC17

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PQSPA6	PQSPA5	PQSPA4	PQSPA3	0	PQSPA1	PQSPA0	DDQS7	DDQS6	DDQS5	DDQS4	DDQS3	DDQS2	DDQS1	DDQS0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Clearing a bit in the PQSPAR assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the QSPI. The PQSPAR does not affect operation of the SCI.

Pin Names	Mnemonics	Mode	Function
Master In Slave Out	MISO	Master Slave	Serial Data Input to QSPI Serial Data Output from QSPI
Master Out Slave In	MOSI	Master Slave	Serial Data Output from QSPI Serial Data Input to QSPI
Serial Clock	SCK	Master Slave	Clock Output from QSPI Clock Input to QSPI
Peripheral Chip Selects	PCS[3:1]	Master	Select Peripherals
Peripheral Chip Select Slave Select	PCS0 SS	Master Master Slave	Selects Peripheral Causes Mode Fault Initiates Serial Transfer

## 6.5.2 QSPI Registers

The programmer's model for the QSPI submodule consists of the QSM global and pin control registers, four QSPI control registers, one status register, and the 80-byte QSPI RAM.

The CPU can read and write to registers and RAM. The four control registers must be initialized before the QSPI is enabled to ensure defined operation. SPCR1 should be written last because it contains QSPI enable bit SPE. Asserting this bit starts the QSPI. The QSPI control registers are reset to a defined state and can then be changed by the CPU. Reset values are shown below each register.

Refer to the following memory map of the QSPI.

Address	Name	Usage
\$YFFC18	SPCR0	QSPI Control Register 0
\$YFFC1A	SPCR1	QSPI Control Register 1
\$YFFC1C	SPCR2	QSPI Control Register 2
\$YFFC1E	SPCR3	QSPI Control Register 3
\$YFFC1F	SPSR	QSPI Status Register
\$YFFD00	RAM	QSPI Receive Data (16 Words)
\$YFFD20	RAM	QSPI Transmit Data (16 Words)
\$YFFD40	RAM	QSPI Command Control (8 Words)

Writing a different value into any control register except SPCR2 while the QSPI is enabled disrupts operation. SPCR2 is buffered to prevent disruption of the current serial transfer. After completion of the current serial transfer, the new SPCR2 values become effective.

Writing the same value into any control register except SPCR2 while the QSPI is enabled has no effect on QSPI operation. Rewriting NEWQP in SPCR2 causes execution to restart at the designated location.

### SPCR0 — QSPI Control Register 0

**\$YFFC18**

15	14	13		10	9	8	7								0
MSTR	WOMQ			BITS		CPOL	CPHA								SPBR

RESET:

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

SPCR0 contains parameters for configuring the QSPI before it is enabled. The CPU can read and write this register. The QSM has read-only access.

## PE — Parity Enable

- 0 = SCI parity disabled
- 1 = SCI parity enabled

PE determines whether parity is enabled or disabled for both the receiver and the transmitter. If the received parity bit is not correct, the SCI sets the PF error flag in SCSR.

When PE is set, the most significant bit (MSB) of the data field is used for the parity function, which results in either seven or eight bits of user data, depending on the condition of M bit. The following table lists the available choices.

M	PE	Result
0	0	8 Data Bits
0	1	7 Data Bits, 1 Parity Bit
1	0	9 Data Bits
1	1	8 Data Bits, 1 Parity Bit

## M — Mode Select

- 0 = SCI frame: 1 start bit, 8 data bits, 1 stop bit (10 bits total)
- 1 = SCI frame: 1 start bit, 9 data bits, 1 stop bit (11 bits total)

## WAKE — Wakeup by Address Mark

- 0 = SCI receiver awakened by idle-line detection
- 1 = SCI receiver awakened by address mark (last bit set)

## TIE — Transmit Interrupt Enable

- 0 = SCI TDRE interrupts inhibited
- 1 = SCI TDRE interrupts enabled

## TCIE — Transmit Complete Interrupt Enable

- 0 = SCI TC interrupts inhibited
- 1 = SCI TC interrupts enabled

## RIE — Receiver Interrupt Enable

- 0 = SCI RDRF interrupt inhibited
- 1 = SCI RDRF interrupt enabled

## ILIE — Idle-Line Interrupt Enable

- 0 = SCI IDLE interrupts inhibited
- 1 = SCI IDLE interrupts enabled

## TE — Transmitter Enable

- 0 = SCI transmitter disabled (TXD pin may be used as I/O)
- 1 = SCI transmitter enabled (TXD pin dedicated to SCI transmitter)

The transmitter retains control of the TXD pin until completion of any character transfer that was in progress when TE is cleared.

## RE — Receiver Enable

- 0 = SCI receiver disabled (status bits inhibited)
- 1 = SCI receiver enabled

## RWU — Receiver Wakeup

- 0 = Normal receiver operation (received data recognized)
- 1 = Wakeup mode enabled (received data ignored until awakened)

Setting RWU enables the wakeup function, which allows the SCI to ignore received data until awakened by either an idle line or address mark (as determined by WAKE). When in wakeup mode, the receiver status flags are not set, and interrupts are inhibited. This bit is cleared automatically (returned to normal mode) when the receiver is awakened.

## 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.