



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	Not For New Designs
Core Processor	CPU32
Core Size	32-Bit Single-Core
Speed	16.78MHz
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)	3V ~ 3.6V
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/mc68lk332gceh16">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68lk332gceh16</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

## 1.1 Features

- Central Processing Unit (CPU32)
  - 32-Bit Architecture
  - Virtual Memory Implementation
  - Table Lookup and Interpolate Instruction
  - Improved Exception Handling for Controller Applications
  - High-Level Language Support
  - Background Debugging Mode
  - Fully Static Operation
- System Integration Module (SIM)
  - External Bus Support
  - Programmable Chip-Select Outputs
  - System Protection Logic
  - Watchdog Timer, Clock Monitor, and Bus Monitor
  - Two 8-Bit Dual Function Input/Output Ports
  - One 7-Bit Dual Function Output Port
  - Phase-Locked Loop (PLL) Clock System
- Time Processor Unit (TPU)
  - Dedicated Microengine Operating Independently of CPU32
  - 16 Independent, Programmable Channels and Pins
  - Any Channel can Perform any Time Function
  - Two Timer Count Registers with Programmable Prescalers
  - Selectable Channel Priority Levels
- Queued Serial Module (QSM)
  - Enhanced Serial Communication Interface
  - Queued Serial Peripheral Interface
  - One 8-Bit Dual Function Port
- Static RAM Module with TPU Emulation Capability (TPURAM)
  - 2-Kbytes of Static RAM
  - May be Used as Normal RAM or TPU Microcode Emulation RAM

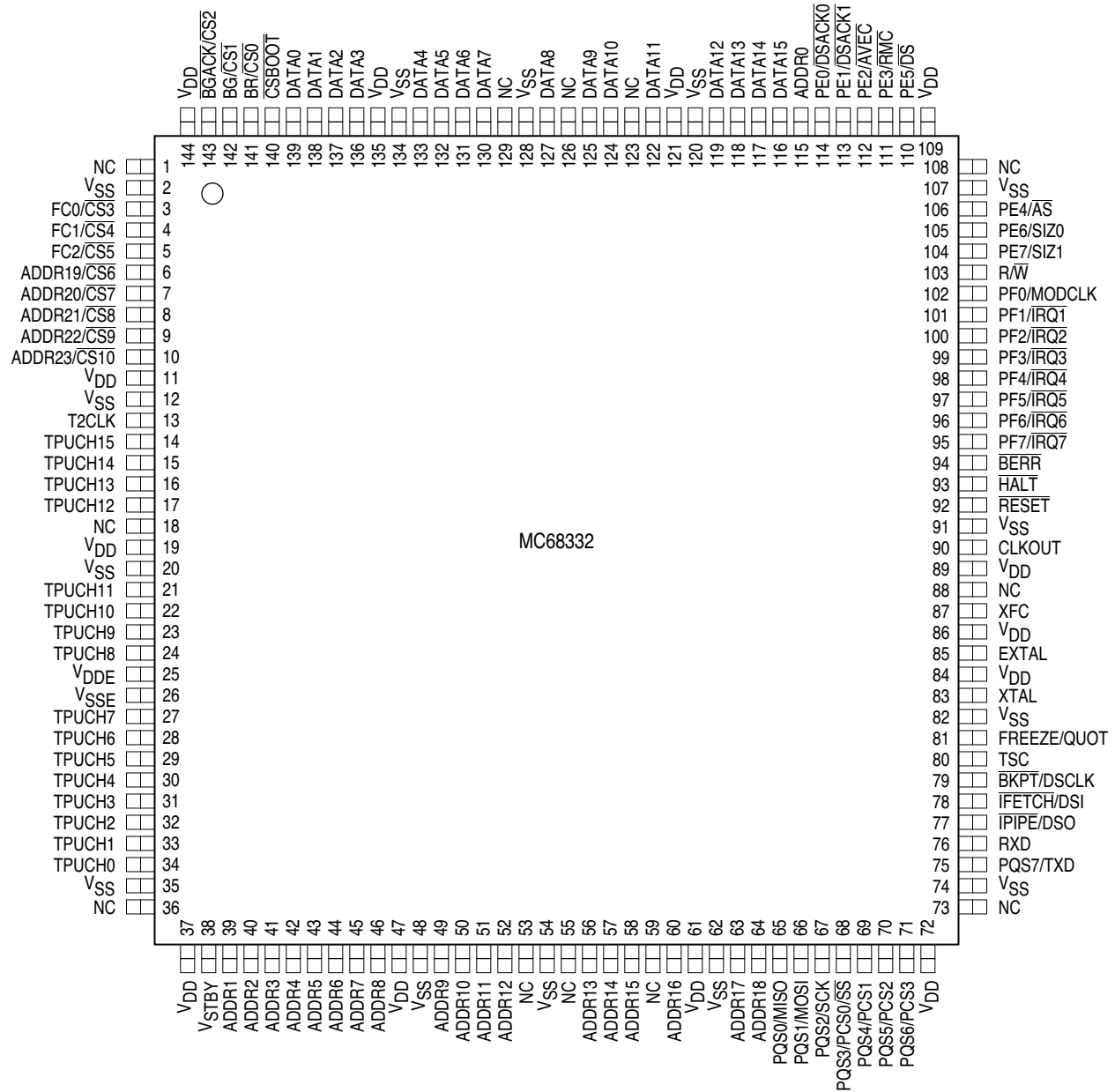


Figure 3 MC68332 144-Pin QFP Pin Assignments

332 144-PIN QFP

Table 2 MCU Pin Characteristic (Continued)

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
T2CLK	—	Y	Y	—	—
TPUCH[15:0]	A	Y	Y	—	—
TSC	—	Y	Y	—	—
TXD	Bo	Y	Y	I/O	PQS7
XFC <sup>2</sup>	—	—	—	Special	—
XTAL <sup>2</sup>	—	—	—	Special	—

NOTES:

1. DATA[15:0] are synchronized during reset only. MODCLK is synchronized only when used as an input port pin.
2. EXTAL, XFC, and XTAL are clock reference connections.

## 2.2 MCU Power Connections

Table 3 MCU Power Connections

$V_{STBY}$	Standby RAM Power/Clock Synthesizer Power
$V_{DDSYN}$	Clock Synthesizer Power
$V_{SSE}/V_{DDE}$	External Periphery Power (Source and Drain)
$V_{SSI}/V_{DDI}$	Internal Module Power (Source and Drain)

## 2.3 MCU Driver Types

Table 4 MCU Driver Types

Type	I/O	Description
A	O	Output-only signals that are always driven; no external pull-up required
Aw	O	Type A output with weak P-channel pull-up during reset
B	O	Three-state output that includes circuitry to pull up output before high impedance is established, to ensure rapid rise time. An external holding resistor is required to maintain logic level while the pin is in the high-impedance state.
Bo	O	Type B output that can be operated in an open-drain mode

Table 6 MCU Signal Function (Continued)

Signal Name	Mnemonic	Function
Quotient Out	QUOT	Provides the quotient bit of the polynomial divider
Reset	RESET	System reset
Read-Modify-Write Cycle	RMC	Indicates an indivisible read-modify-write instruction
Read/Write	R/W	Indicates the direction of data transfer on the bus
SCI Receive Data	RXD	Serial input to the SCI
QSPI Serial Clock	SCK	Clock output from QSPI in master mode; clock input to QSPI in slave mode
Size	SIZ[1:0]	Indicates the number of bytes to be transferred during a bus cycle
Slave Select	SS	Causes serial transmission when QSPI is in slave mode; causes mode fault in master mode
TCR2 Clock	T2CLK	External clock source for TCR2 counter
TPU Channel Pins	TPUCH[15:0]	Bidirectional pins associated with TPU channels
Three-State Control	TSC	Places all output drivers in a high-impedance state
SCI Transmit Data	TXD	Serial output from the SCI
External Filter Capacitor	XFC	Connection for external phase-locked loop filter capacitor

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.

## 3.2.2 System Protection Control Register

The system protection control register controls system monitor functions, software watchdog clock prescaling, and bus monitor timing. This register can be written only once following power-on or reset, but can be read at any time.

### SYPCR —System Protection Control Register

**\$YFFA21**

15	8	7	6	5	4	3	2	1	0
NOT USED								SWE	BMT
RESET:								1	0
								MODCLK	0

SWE — Software Watchdog Enable

0 = Software watchdog disabled

1 = Software watchdog enabled

SWP — Software Watchdog Prescale

This bit controls the value of the software watchdog prescaler.

0 = Software watchdog clock not prescaled

1 = Software watchdog clock prescaled by 512

SWT[1:0] — Software Watchdog Timing

This field selects the divide ratio used to establish software watchdog time-out period. The following table gives the ratio for each combination of SWP and SWT bits.

SWP	SWT	Ratio
0	00	$2^9$
0	01	$2^{11}$
0	10	$2^{13}$
0	11	$2^{15}$
1	00	$2^{18}$
1	01	$2^{20}$
1	10	$2^{22}$
1	11	$2^{24}$

HME — Halt Monitor Enable

0 = Disable halt monitor function

1 = Enable halt monitor function

BME — Bus Monitor External Enable

0 = Disable bus monitor function for an internal to external bus cycle.

1 = Enable bus monitor function for an internal to external bus cycle.

BMT[1:0] — Bus Monitor Timing

This field selects a bus monitor time-out period as shown in the following table.

BMT	Bus Monitor Time-out Period
00	64 System Clocks
01	32 System Clocks
10	16 System Clocks
11	8 System Clocks



## 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.

## CSPAR1 —Chip Select Pin Assignment Register 1

\$YFFA46

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CSPA1[4]		CSPA1[3]		CSPA1[2]		CSPA1[1]		CSPA1[0]	
RESET:															
0	0	0	0	0	0	DATA7	1	DATA [7:6]	1	DATA [7:5]	1	DATA [7:4]	1	DATA [7:3]	1

CSPAR1 contains five 2-bit fields that determine the functions of corresponding chip-select pins. CSPAR1[15:10] are not used. These bits always read zero; writes have no effect.

**Table 14 CSPAR1 Pin Assignments**

CSPAR0 Field	Chip Select Signal	Alternate Signal	Discrete Output
CSPA1[4]	$\overline{CS10}$	ADDR23	ECLK
CSPA1[3]	$\overline{CS9}$	ADDR22	PC6
CSPA1[2]	$\overline{CS8}$	ADDR21	PC5
CSPA1[1]	$\overline{CS7}$	ADDR20	PC4
CSPA1[0]	$\overline{CS6}$	ADDR19	PC3

At reset, either the alternate function (01) or chip-select function (11) can be encoded. DATA pins are driven to logic level one by a weak interval pull-up during reset. Encoding is for chip-select function unless a data line is held low during reset. Note that bus loading can overcome the weak pull-up and hold pins low during reset. The following table shows the hierarchical selection method that determines the reset functions of pins controlled by CSPAR1.

**Table 15 Reset Pin Function of  $\overline{CS}[10:6]$**

Data Bus Pins at Reset					Chip-Select/Address Bus Pin Function				
DATA7	DATA6	DATA5	DATA4	DATA3	$\overline{CS10}/$ ADDR23	$\overline{CS9}/$ ADDR22	$\overline{CS8}/$ ADDR21	$\overline{CS7}/$ ADDR20	$\overline{CS6}/$ ADDR19
1	1	1	1	1	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	$\overline{CS7}$	$\overline{CS6}$
1	1	1	1	0	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	$\overline{CS7}$	ADDR19
1	1	1	0	X	$\overline{CS10}$	$\overline{CS9}$	$\overline{CS8}$	ADDR20	ADDR19
1	1	0	X	X	$\overline{CS10}$	$\overline{CS9}$	ADDR21	ADDR20	ADDR19
1	0	X	X	X	$\overline{CS10}$	ADDR22	ADDR21	ADDR20	ADDR19
0	X	X	X	X	ADDR23	ADDR22	ADDR21	ADDR20	ADDR19

A pin programmed as a discrete output drives an external signal to the value specified in the port C pin data register (PORTC), with the following exceptions:

1. No discrete output function is available on pins  $\overline{BR}$ ,  $\overline{BG}$ , or  $\overline{BGACK}$ .
2. ADDR23 provides E-clock output rather than a discrete output signal.

When a pin is programmed for discrete output or alternate function, internal chip-select logic still functions and can be used to generate  $\overline{DSACK}$  or  $\overline{AVEC}$  internally on an address match.

Port size is determined when a pin is assigned as a chip select. When a pin is assigned to an 8-bit port, the chip select is asserted at all addresses within the block range. If a pin is assigned to a 16-bit port, the upper/lower byte field of the option register selects the byte with which the chip select is associated.

## PEPAR — Port E Pin Assignment Register

\$YFFA17

15	8	7	6	5	4	3	2	1	0
NOT USED		PEPA7	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0

RESET:

DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8

The bits in this register control the function of each port E pin. Any bit set to one configures the corresponding pin as a bus control signal, with the function shown in the following table. Any bit cleared to zero defines the corresponding pin to be an I/O pin, controlled by PORTE and DDRE.

Data bus bit 8 controls the state of this register following reset. If DATA8 is set to one during reset, the register is set to \$FF, which defines all port E pins as bus control signals. If DATA8 is cleared to zero during reset, this register is set to \$00, configuring all port E pins as I/O pins.

Any bit cleared to zero defines the corresponding pin to be an I/O pin. Any bit set to one defines the corresponding pin to be a bus control signal.

**Table 16 Port E Pin Assignments**

PEPAR Bit	Port E Signal	Bus Control Signal
PEPA7	PE7	SIZ1
PEPA6	PE6	SIZ0
PEPA5	PE5	$\overline{AS}$
PEPA4	PE4	$\overline{DS}$
PEPA3	PE3	$\overline{RMC}$
PEPA2	PE2	$\overline{AVEC}$
PEPA1	PE1	$\overline{DSACK1}$
PEPA0	PE0	$\overline{DSACK0}$

## PORTF0, PORTF1 — Port F Data Register

\$YFFA19, \$YFFA1B

15	8	7	6	5	4	3	2	1	0
NOT USED		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0

RESET:

U U U U U U U U

The write to the port F data register is stored in the internal data latch, and if any port F pin is configured as an output, the value stored for that bit is driven onto the pin. A read of the port F 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 F data register is a single register that can be accessed in two locations. When accessed at \$YFFA19, the register is referred to as PORTF0; when accessed at \$YFFA1B, the register is referred to as PORTF1. The register can be read or written at any time. It is unaffected by reset.

## DDRF — Port F Data Direction Register

\$YFFA1D

15	8	7	6	5	4	3	2	1	0
NOT USED		DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0

RESET:

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 for 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.

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:

## 4 Central Processor Unit

Based on the powerful MC68020, the CPU32 processing module provides enhanced system performance and also uses the extensive software base for the Motorola M68000 family.

### 4.1 Overview

The CPU32 is fully object code compatible with the M68000 Family, which excels at processing calculation-intensive algorithms and supporting high-level languages. The CPU32 supports all of the MC68010 and most of the MC68020 enhancements, such as virtual memory support, loop mode operation, instruction pipeline, and 32-bit mathematical operations. Powerful addressing modes provide compatibility with existing software programs and increase the efficiency of high-level language compilers. Special instructions, such as table lookup and interpolate and low-power stop, support the specific requirements of controller applications. Also included is the background debugging mode, an alternate operating mode that suspends normal operation and allows the CPU to accept debugging commands from the development system.

Ease of programming is an important consideration in using a microcontroller. The CPU32 instruction set is optimized for high performance. The eight 32-bit general-purpose data registers readily support 8-bit (byte), 16-bit (word), and 32-bit (long word) operations. Ease of program checking and diagnosis is further enhanced by trace and trap capabilities at the instruction level.

Use of high-level languages is increasing as controller applications become more complex and control programs become larger. High-level languages aid rapid development of software, with less error, and are readily portable. The CPU32 instruction set supports high-level languages.

### 4.2 Programming Model

The CPU32 has sixteen 32-bit general registers, a 32-bit program counter, one 32-bit supervisor stack pointer, a 16-bit status register, two alternate function code registers, and a 32-bit vector base register.

The programming model of the CPU32 consists of a user model and supervisor model, corresponding to the user and supervisor privilege levels. Some instructions available at the supervisor level are not available at the user level, allowing the supervisor to protect system resources from uncontrolled access. Bit S in the status register determines the privilege level.

The user programming model remains unchanged from previous M68000 Family microprocessors. Application software written to run at the non-privileged user level migrates without modification to the CPU32 from any M68000 platform. The move from SR instruction, however, is privileged in the CPU32. It is not privileged in the M68000.

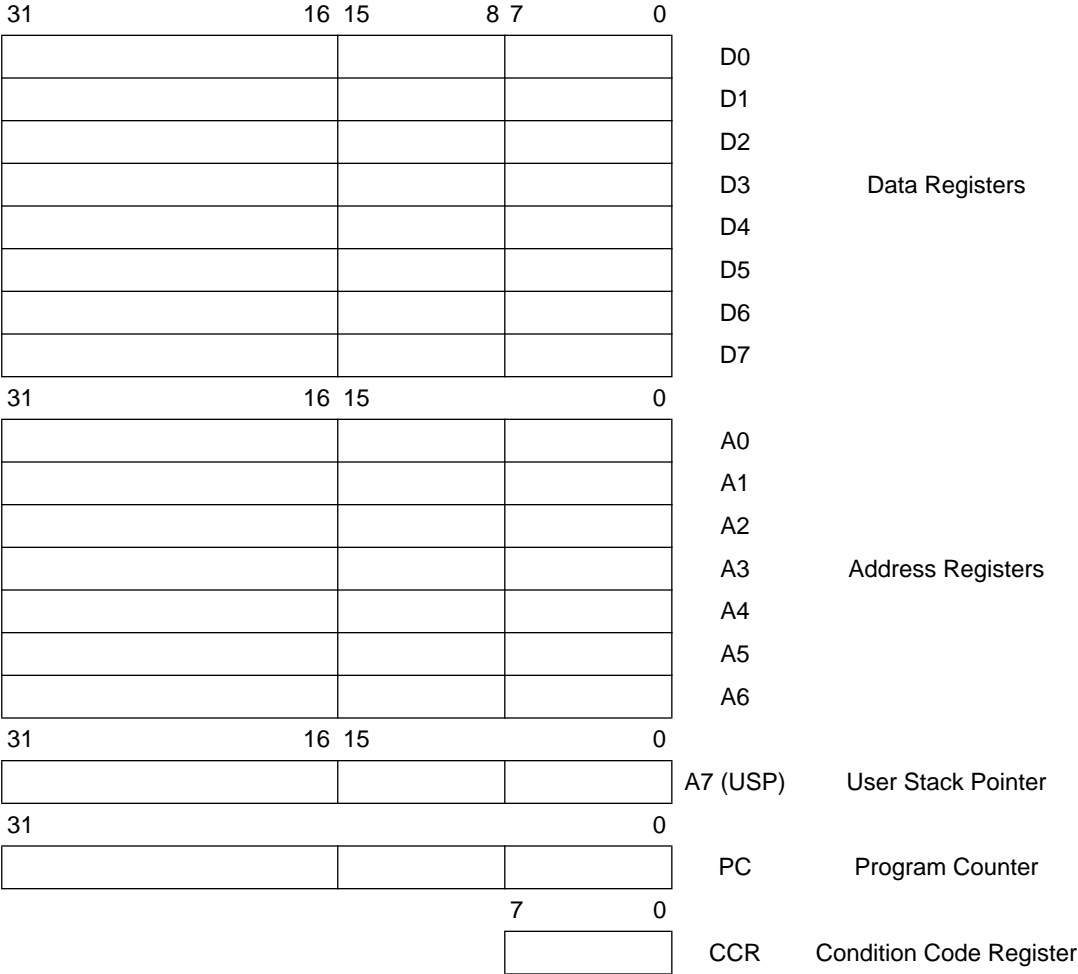


Figure 10 User Programming Model

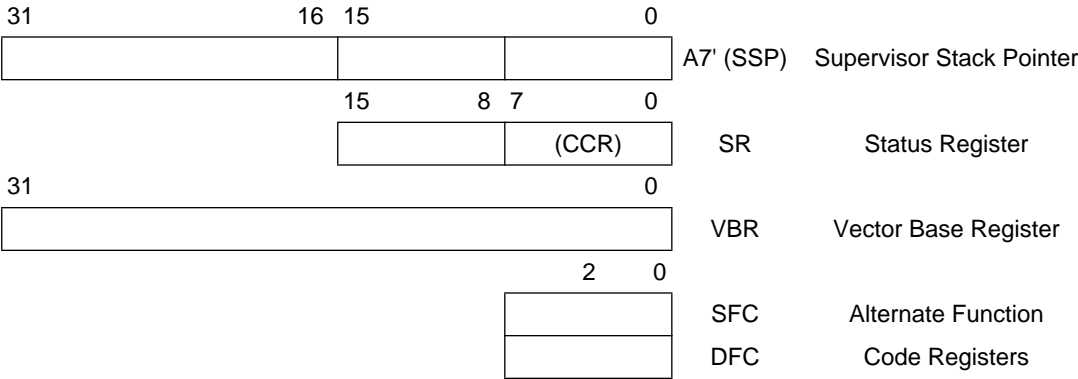
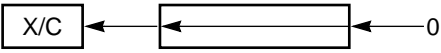
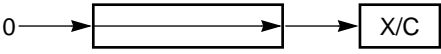


Figure 11 Supervisor Programming Model Supplement

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

## 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



## SPCR2 — QSPI Control Register 2

\$YFFC1C

15	14	13	12	11	8	7	6	5	4	3	0
SPIFIE	WREN	WRT0	0	ENDQP		0	0	0	0	NEWQP	

RESET:

0 0 0 0 0 0 0 0 0 0 0 0

SPCR2 contains QSPI configuration parameters. The CPU can read and write this register; the QSM has read access only. Writes to SPCR2 are buffered. A write to SPCR2 that changes a bit value while the QSPI is operating is ineffective on the current serial transfer, but becomes effective on the next serial transfer. Reads of SPCR2 return the current value of the register, not of the buffer.

### SPIFIE — SPI Finished Interrupt Enable

0 = QSPI interrupts disabled

1 = QSPI interrupts enabled

SPIFIE enables the QSPI to generate a CPU interrupt upon assertion of the status flag SPIF.

### WREN — Wrap Enable

0 = Wraparound mode disabled

1 = Wraparound mode enabled

WREN enables or disables wraparound mode.

### WRT0 — Wrap To

When wraparound mode is enabled, after the end of queue has been reached, WRT0 determines which address the QSPI executes.

### Bit 12 — Not Implemented

### ENDQP — Ending Queue Pointer

This field contains the last QSPI queue address.

### Bits [7:4] — Not Implemented

### NEWQP — New Queue Pointer Value

This field contains the first QSPI queue address.

## SPCR3 — QSPI Control Register 3

\$YFFC1E

15	14	13	12	11	10	9	8	7	0
0	0	0	0	0	LOOPQ	HMIE	HALT	SPSR	

RESET:

0 0 0 0 0 0 0 0

SPCR3 contains QSPI configuration parameters. The CPU can read and write SPCR3, but the QSM has read-only access.

### Bits [15:11] — Not Implemented

### LOOPQ — QSPI Loop Mode

0 = Feedback path disabled

1 = Feedback path enabled

LOOPQ controls feedback on the data serializer for testing.

### HMIE — HALTA and MODF Interrupt Enable

0 = HALTA and MODF interrupts disabled

1 = HALTA and MODF interrupts enabled

HMIE controls CPU interrupts caused by the HALTA status flag or the MODF status flag in SPSR.

Command RAM is used by the QSPI when in master mode. The CPU writes one byte of control information to this segment for each QSPI command to be executed. The QSPI cannot modify information in command RAM.

Command RAM consists of 16 bytes. Each byte is divided into two fields. The peripheral chip-select field enables peripherals for transfer. The command control field provides transfer options.

A maximum of 16 commands can be in the queue. Queue execution by the QSPI proceeds from the address in NEWQP through the address in ENDQP. (Both of these fields are in SPCR2.)

CONT — Continue

- 0 = Control of chip selects returned to PORTQS after transfer is complete.
- 1 = Peripheral chip selects remain asserted after transfer is complete.

BITSE — Bits per Transfer Enable

- 0 = 8 bits
- 1 = Number of bits set in BITS field of SPCR0

DT — Delay after Transfer

The QSPI provides a variable delay at the end of serial transfer to facilitate the interface with peripherals that have a latency requirement. The delay between transfers is determined by the SPCR1 DTL field.

DSCK — PCS to SCK Delay

- 0 = PCS valid to SCK transition is one-half SCK.
- 1 = SPCR1 DSCKL field specifies delay from PCS valid to SCK.

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select an external device for serial data transfer. More than one peripheral chip select can be activated at a time, and more than one peripheral chip can be connected to each PCS pin, provided that proper fanout is observed.

$\overline{SS}$  — Slave Mode Select

Initiates slave mode serial transfer. If  $\overline{SS}$  is taken low when the QSPI is in master mode, a mode fault will be generated.

## 6.5.4 Operating Modes

The QSPI operates in either master or slave mode. Master mode is used when the MCU originates data transfers. Slave mode is used when an external device initiates serial transfers to the MCU through the QSPI. Switching between the modes is controlled by MSTR in SPCR0. Before entering either mode, appropriate QSM and QSPI registers must be properly initialized.

In master mode, the QSPI executes a queue of commands defined by control bits in each command RAM queue entry. Chip-select pins are activated, data is transmitted from transmit RAM and received into receive RAM.

In slave mode, operation proceeds in response to  $\overline{SS}$  pin activation by an external bus master. Operation is similar to master mode, but no peripheral chip selects are generated, and the number of bits transferred is controlled in a different manner. When the QSPI is selected, it automatically executes the next queue transfer to exchange data with the external device correctly.

Although the QSPI inherently supports multimaster operation, no special arbitration mechanism is provided. A mode fault flag (MODF) indicates a request for SPI master arbitration. System software must provide arbitration. Note that unlike previous SPI systems, MSTR is not cleared by a mode fault being set, nor are the QSPI pin output drivers disabled. The QSPI and associated output drivers must be disabled by clearing SPE in SPCR1.

Writing a value of zero to SCBR disables the baud rate generator.

The following table lists the SCBR settings for standard and maximum baud rates using 16.78-MHz and 20.97-MHz system clocks.

**Table 27 SCI Baud Rates**

Nominal Baud Rate	Actual Rate with 16.78-MHz Clock	SCBR Value	Actual Rate with 20.97-MHz Clock	SCBR Value
64*	64.0	\$1FFF	—	—
110	110.0	\$129E	110.0	\$1745
300	299.9	\$06D4	300.1	\$0888
600	599.9	\$036A	600.1	\$0444
1200	1199.7	\$0165	1200.3	\$0222
2400	2405.0	\$00DA	2400.6	\$0111
4800	4810.0	\$006D	4783.6	\$0089
9600	9532.5	\$0037	9637.6	\$0044
19200	19418.1	\$0016	19275.3	\$0022
38400	37449.1	\$000E	38550.6	\$0011
76800	74898.3	\$0007	72817.8	\$0009
Maximum Rate	524288.0	\$0001	655360.0	\$0001

## SCCR1 — SCI Control Register 1

**\$YFFC0A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	LOOPS	WOMS	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SCCR1 contains SCI configuration parameters. The CPU can read and write this register at any time. The SCI can modify RWU in some circumstances. In general, interrupts enabled by these control bits are cleared by reading SCSR, then reading (receiver status bits) or writing (transmitter status bits) SCDR.

Bit 15 — Not Implemented

### LOOPS — Loop Mode

- 0 = Normal SCI operation, no looping, feedback path disabled
- 1 = Test SCI operation, looping, feedback path enabled

LOOPS controls a feedback path on the data serial shifter. When loop mode is enabled, SCI transmitter output is fed back into the receive serial shifter. TXD is asserted (idle line). Both transmitter and receiver must be enabled before entering loop mode.

### WOMS — Wired-OR Mode for SCI Pins

- 0 = If configured as an output, TXD is a normal CMOS output.
- 1 = If configured as an output, TXD is an open-drain output.

WOMS determines whether the TXD pin is an open-drain output or a normal CMOS output. This bit is used only when TXD is an output. If TXD is used as a general-purpose input pin, WOMS has no effect.

### ILT — Idle-Line Detect Type

- 0 = Short idle-line detect (start count on first one)
- 1 = Long idle-line detect (start count on first one after stop bit(s))

### PT — Parity Type

- 0 = Even parity
- 1 = Odd parity

When parity is enabled, PT determines whether parity is even or odd for both the receiver and the transmitter.

## 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.

