



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](#)"

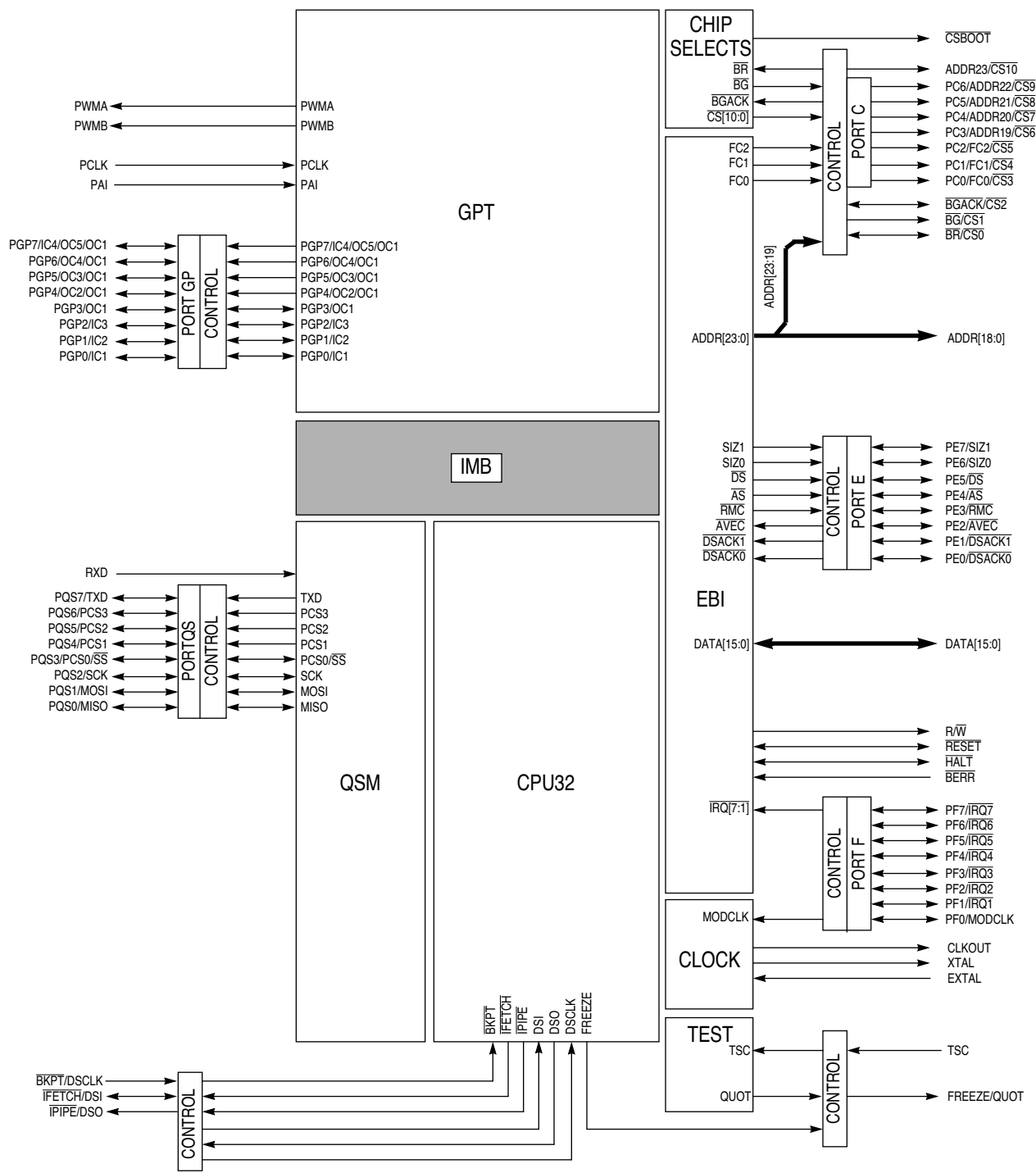
#### Details

Product Status	Obsolete
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	18
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/mc68331meh16">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68331meh16</a>

## 1.1 Features

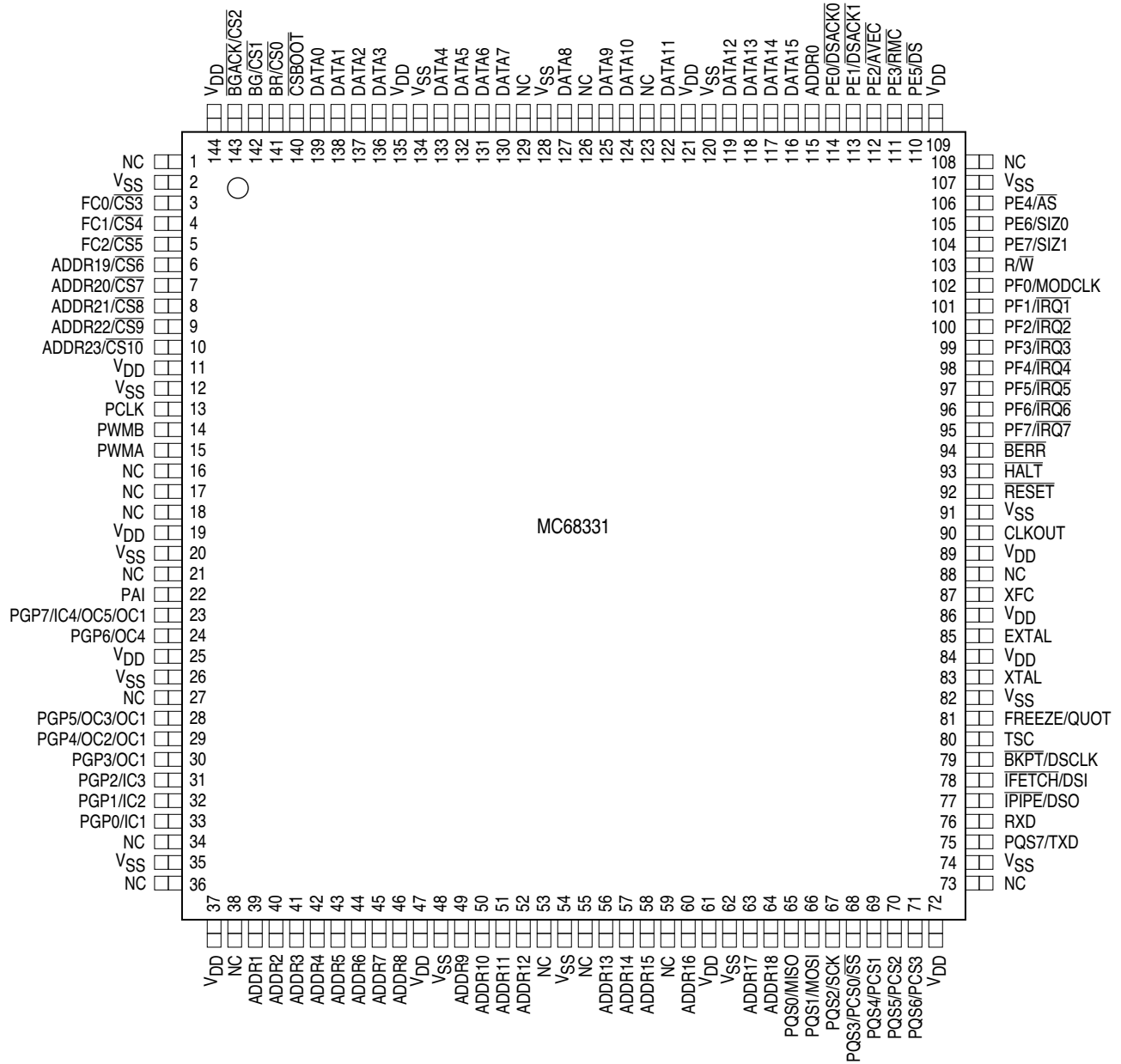
- Modular Architecture
- Central Processing Unit (CPU32)
  - Upward Object Code Compatible
  - New Instructions for Controller Applications
  - 32-Bit Architecture
  - Virtual Memory Implementation
  - Loop Mode of Instruction Execution
  - Table Lookup and Interpolate Instruction
  - Improved Exception Handling for Controller Applications
  - Trace on Change of Flow
  - Hardware Breakpoint Signal, Background 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
  - System Protection Logic
  - System Clock Based on 32.768-kHz Crystal for Low Power Operation
  - Test/Debug Submodule for Factory/User Test and Development
- Queued Serial Module (QSM)
  - Enhanced Serial Communication Interface (SCI), Universal Asynchronous Receiver Transmitter (UART): Modulus Baud Rate, Parity
  - Queued Serial Peripheral Interface (QSPI): 80-Byte RAM, Up to 16 Automatic Transfers
  - Dual Function I/O Ports
  - Continuous Cycling, 8 to 16 Bits per Transfer
- General-Purpose Timer (GPT)
  - Two 16-Bit Free-Running Counters With One Nine-Stage Prescaler
  - Three Input Capture Channels
  - Four Output Compare Channels
  - One Input Capture/Output Compare Channel
  - One Pulse Accumulator/Event Counter Input
  - Two Pulse-Width Modulation Outputs
  - Optional External Clock Input

## 1.2 Block Diagram



331 BLOCK

Figure 1 MCU Block Diagram



331 144-PIN QFP

Figure 3 MC68331 144-Pin QFP Pin Assignments

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

Signal Name	MCU Module	Signal Type	Active State
DSACK[1:0]	SIM	Input	0
DSCLK	CPU32	Input	Serial Clock
DSI	CPU32	Input	(Serial Data)
DSO	CPU32	Output	(Serial Data)
EXTAL	SIM	Input	—
FC[2:0]	SIM	Output	—
FREEZE	SIM	Output	1
HALT	SIM	Input/Output	0
IC[4:1]	GPT	Input	—
IFETCH	CPU32	Output	—
IPIPE	CPU32	Output	—
IRQ[7:1]	SIM	Input	0
MISO	QSM	Input/Output	—
MODCLK	SIM	Input	—
MOSI	QSM	Input/Output	—
OC[5:1]	GPT	Output	—
PAI	GPT	Input	—
PC[6:0]	SIM	Output	(Port)
PCS[3:0]	QSM	Input/Output	—
PE[7:0]	SIM	Input/Output	(Port)
PF[7:0]	SIM	Input/Output	(Port)
PQS[7:0]	QSM	Input/Output	(Port)
PCLK	GPT	Input	—
PWMA, PWMB	GPT	Output	—
QUOT	SIM	Output	—
RESET	SIM	Input/Output	0
RMC	SIM	Output	0
R/W	SIM	Output	1/0
RXD	QSM	Input	—
SCK	QSM	Input/Output	—
SIZ[1:0]	SIM	Output	—
SS	QSM	Input	0
TSC	SIM	Input	—
TXD	QSM	Output	—
XFC	SIM	Input	—
XTAL	SIM	Output	—

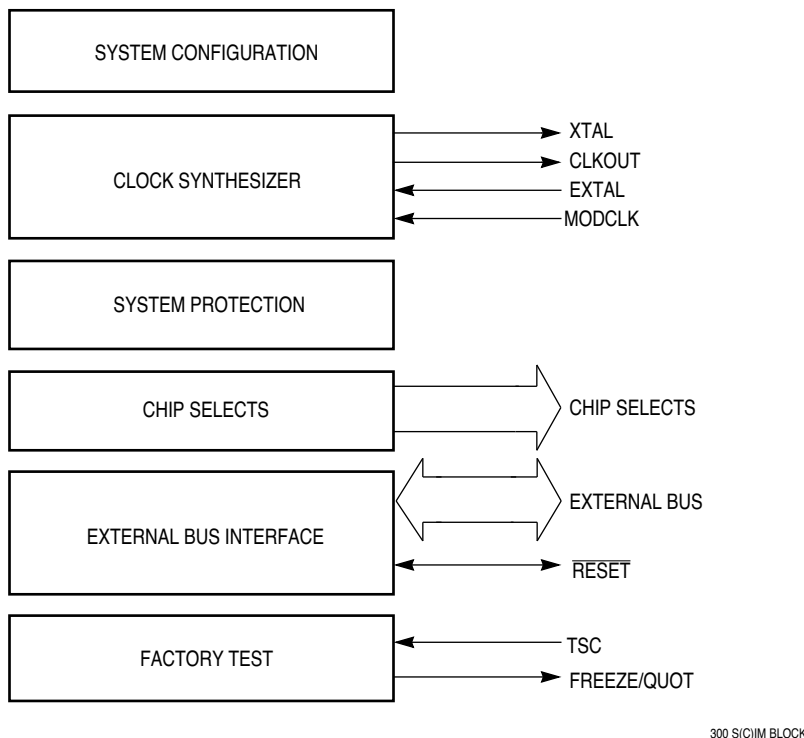
## 2.5 Signal Function

**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

### 3 System Integration Module

The system integration module (SIM) consists of five functional blocks that control system start-up, initialization, configuration, and external bus.



**Figure 5 SIM Block Diagram**

#### 3.1 Overview

The system configuration and protection block controls MCU configuration and operating mode. The block also provides bus and software watchdog monitors.

The system clock generates clock signals used by the SIM, other IMB modules, and external devices. In addition, a periodic interrupt generator supports execution of time-critical control routines.

The external bus interface handles the transfer of information between IMB modules and external address space.

The chip-select block provides eleven general-purpose chip-select signals and a boot ROM chip-select signal. Both general-purpose and boot ROM chip-select signals have associated base address registers and option registers.

The system test block incorporates hardware necessary for testing the MCU. It is used to perform factory tests, and its use in normal applications is not supported.

The SIM control register address map occupies 128 bytes. Unused registers within the 128-byte address space return zeros when read. The “Access” column in the SIM 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 SIMCR.

## 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

### 3.2.3 Bus Monitor

The internal bus monitor checks for excessively long  $\overline{DSACK}$  response times during normal bus cycles and for excessively long  $\overline{DSACK}$  or  $\overline{AVEC}$  response times during interrupt acknowledge cycles. The monitor asserts  $\overline{BERR}$  if response time is excessive.

$\overline{DSACK}$  and  $\overline{AVEC}$  response times are measured in clock cycles. The maximum allowable response time can be selected by setting the BMT field.

The monitor does not check  $\overline{DSACK}$  response on the external bus unless the CPU initiates the bus cycle. The BME bit in the SYPCR enables the internal bus monitor for internal to external bus cycles. If a system contains external bus masters, an external bus monitor must be implemented and the internal to external bus monitor option must be disabled.

### 3.2.4 Halt Monitor

The halt monitor responds to an assertion of  $\overline{HALT}$  on the internal bus. A flag in the reset status register (RSR) indicates that the last reset was caused by the halt monitor. The halt monitor reset can be inhibited by the HME bit in the SYPCR.

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/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/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.



Byte	Description
00	Disable
01	Lower Byte
10	Upper Byte
11	Both Bytes

## R/W —Read/Write

This field causes a chip select to be asserted only for a read, only for a write, or for both read and write. Refer to the following table for options available.

R/W	Description
00	Reserved
01	Read Only
10	Write Only
11	Read/Write

## STRB —Address Strobe/Data Strobe

- 0 = Address strobe
- 1 = Data strobe

This bit controls the timing for assertion of a chip select in asynchronous mode. Selecting address strobe causes chip select to be asserted synchronized with address strobe. Selecting data strobe causes chip select to be asserted synchronized with data strobe.

## DSACK —Data and Size Acknowledge

This field specifies the source of  $\overline{\text{DSACK}}$  in asynchronous mode. It also allows the user to adjust bus timing with internal  $\overline{\text{DSACK}}$  generation by controlling the number of wait states that are inserted to optimize bus speed in a particular application. The following table shows the  $\overline{\text{DSACK}}$  field encoding. The fast termination encoding (1110) is used for two-cycle access to external memory.

DSACK	Description
0000	No Wait States
0001	1 Wait State
0010	2 Wait States
0011	3 Wait States
0100	4 Wait States
0101	5 Wait States
0110	6 Wait States
0111	7 Wait States
1000	8 Wait States
1001	9 Wait States
1010	10 Wait States
1011	11 Wait States
1100	12 Wait States
1101	13 Wait States
1110	Fast Termination
1111	External $\overline{\text{DSACK}}$

## PORTF0, PORTF1 — Port F Data Register

\$YFFA19, \$YFFA1B

15	8	7	6	5	4	3	2	1	0
NOT USED								PF7	PF6

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

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.

## PFPA0 — Port F Pin Assignment Register

\$YFFA1F

15	8	7	6	5	4	3	2	1	0
NOT USED								PFPA7	PFPA6

RESET:

DATA9 DATA9 DATA9 DATA9 DATA9 DATA9 DATA9 DATA9

The bits in this register control the function of each port F pin. 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 an interrupt request signal or MODCLK. The MODCLK signal has no function after reset.

Table 17 Port F Pin Assignments

PFPA Field	Port F Signal	Alternate Signal
PFPA7	PF7	IRQ7
PFPA6	PF6	IRQ6
PFPA5	PF5	IRQ5
PFPA4	PF4	IRQ4
PFPA3	PF3	IRQ3
PFPA2	PF2	IRQ2
PFPA1	PF1	IRQ1
PFPA0	PF0	MODCLK

Data bus pin 9 controls the state of this register following reset. If DATA9 is set to one during reset, the register is set to \$FF, which defines all port F pins as interrupt request inputs. If DATA9 is cleared to zero during reset, this register is set to \$00, defining all port F pins as I/O pins.

**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 10 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 10 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.

Interrupt requests are sampled on consecutive falling edges of the system clock. Interrupt request input circuitry has hysteresis. To be valid, a request signal must be asserted for at least two consecutive clock periods. Valid requests do not cause immediate exception processing, but are left pending. Pending requests are processed at instruction boundaries or when exception processing of higher-priority exceptions is complete.

The CPU32 does not latch the priority of a pending interrupt request. If an interrupt source of higher priority makes a service request while a lower priority request is pending, the higher priority request is serviced. If an interrupt request of equal or lower priority than the current IP mask value is made, the CPU does not recognize the occurrence of the request in any way.

### 3.8.1 Interrupt Acknowledge and Arbitration

Interrupt acknowledge bus cycles are generated during exception processing. When the CPU detects one or more interrupt requests of a priority higher than the interrupt priority mask value, it performs a CPU space read from address \$FFFF : [IP] : 1.

The CPU space read cycle performs two functions: it places a mask value corresponding to the highest priority interrupt request on the address bus, and it acquires an exception vector number from the interrupt source. The mask value also serves two purposes: it is latched into the CCR IP field in order to 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.



## 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 nonprivileged 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.

## 4.7 Background Debugging Mode

The background debugger on the CPU32 is implemented in CPU microcode. The background debugging commands are summarized below.

**Table 21 Background Debugging Mode**

Command	Mnemonic	Description
Read D/A Register	RDREG/RAREG	Read the selected address or data register and return the results through the serial interface.
Write D/A Register	WDREG/WAREG	The data operand is written to the specified address or data register.
Read System Register	RSREG	The specified system control register is read. All registers that can be read in supervisor mode can be read in background mode.
Write System Register	WSREG	The operand data is written into the specified system control register.
Read Memory Location	READ	Read the sized data at the memory location specified by the long-word address. The source function code register (SFC) determines the address space accessed.
Write Memory Location	WRITE	Write the operand data to the memory location specified by the long-word address. The destination function code (DFC) register determines the address space accessed.
Dump Memory Block	DUMP	Used in conjunction with the READ command to dump large blocks of memory. An initial READ is executed to set up the starting address of the block and retrieve the first result. Subsequent operands are retrieved with the DUMP command.
Fill Memory Block	FILL	Used in conjunction with the WRITE command to fill large blocks of memory. Initially, a WRITE is executed to set up the starting address of the block and supply the first operand. The FILL command writes subsequent operands.
Resume Execution	GO	The pipe is flushed and refilled before resuming instruction execution at the current PC.
Patch User Code	CALL	Current program counter is stacked at the location of the current stack pointer. Instruction execution begins at user patch code.
Reset Peripherals	RST	Asserts RESET for 512 clock cycles. The CPU is not reset by this command. Synonymous with the CPU RESET instruction.
No Operation	NOP	NOP performs no operation and can be used as a null command.

## PORTQS — Port QS Data Register

**\$YFFC14**

15	8	7	6	5	4	3	2	1	0
NOT USED								PQS7	PQS0
RESET:									
0 0 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.

**Table 23 QSPAR Pin Assignments**

PQSPAR Field	PQSPAR Bit	Pin Function
PQSPA0	0 1	PQS0 MISO
PQSPA1	0 1	PQS1 MOSI
PQSPA2	0 1	PQS2 <sup>1</sup> SCK
PQSPA3	0 1	PQS3 PCS0/ $\overline{SS}$
PQSPA4	0 1	PQS4 PCS1
PQSPA5	0 1	PQS5 PCS2
PQSPA6	0 1	PQS6 PCS3
PQSPA7	0 1	PQS7 <sup>2</sup> TXD

### NOTES:

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



DDRQS determines whether pins are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function.

**Table 24 Effect of DDRQS on QSM Pin Function**

QSM Pin	Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQ0	0	Serial Data Input to QSPI
			1	Disables Data Input
	Slave		0	Disables Data Output
			1	Serial Data Output from QSPI
MOSI	Master	DDQ1	0	Disables Data Output
			1	Serial Data Output from QSPI
	Slave		0	Serial Data Input to QSPI
			1	Disables Data Input
SCK <sup>1</sup>	Master	DDQ2	0	Disables Clock Output
			1	Clock Output from QSPI
	Slave		0	Clock Input to QSPI
			1	Disables Clock Input
PCS0/ $\overline{\text{SS}}$	Master	DDQ3	0	Assertion Causes Mode Fault
			1	Chip-Select Output
	Slave		0	QSPI Slave Select Input
			1	Disables Select Input
PCS[3:1]	Master	DDQ[4:6] ]	0	Disables Chip-Select Output
			1	Chip-Select Output
	Slave		0	Inactive
			1	Inactive
TXD <sup>2</sup>	Transmit	DDQ7	X	Serial Data Output from SCI
RXD	Receive	None	NA	Serial Data Input to SCI

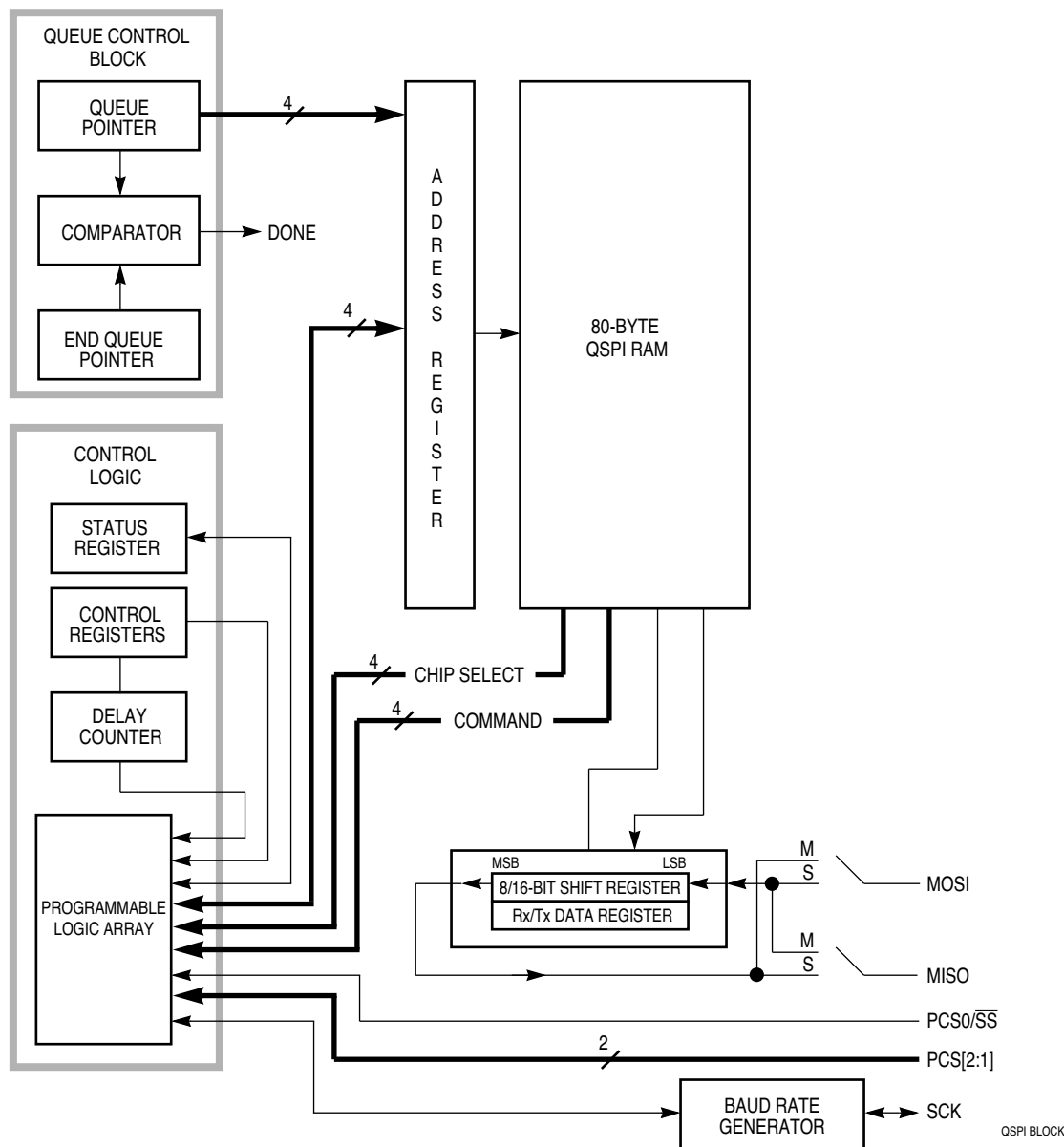
**NOTES:**

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

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

## 5.4 QSPI Submodule

The QSPI submodule communicates with external devices through a synchronous serial bus. The QSPI is fully compatible with the serial peripheral interface (SPI) systems found on other Motorola products. A block diagram of the QSPI is shown below.



**Figure 13 QSPI Block Diagram**

### 5.4.1 QSPI Pins

Seven pins are associated with the QSPI. When not needed for a QSPI application, they can be configured as general-purpose I/O pins. The PCS0/SS pin can function as a peripheral chip select output, slave select input, or general-purpose I/O. Refer to the following table for QSPI input and output pins and their functions.

**Table 25 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

\*A rate of 64 baud is not available with a 20.97-MHz system clock. To achieve this rate, the SYNCR can be programmed to generate a lower system clock rate.

## SCCR1 — SCI Control Register 1

**\$YFFC0A**

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

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))

## CPR[2:0] — Timer Prescaler/PCLK Select Field

This field selects one of seven prescaler taps or PCLK to be TCNT input.

CPR[2:0]	System Clock Divide-By Factor
000	4
001	8
010	16
011	32
100	64
101	128
110	256
111	PCLK

## TFLG1/TFLG2 — Timer Interrupt Flag Registers 1–2

**\$YFF922**

15	14	11	10	8	7	6	5	4	3	2	1	0
I4/O5F	OCF	ICF	TOF	0	PAOVF	PAIF	0	0	0	0	0	0

RESET:

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

These registers show condition flags that correspond to various GPT events. If the corresponding interrupt enable bit in TMSK1/TMSK2 is set, an interrupt occurs.

### I4/O5F — Input Capture 4/Output Compare 5 Flag

When I4/O5 in PACTL is zero, this flag is set each time TCNT matches the value in TOC5. When I4/O5 in PACTL is one, the flag is set each time a selected edge is detected at the I4/O5 pin.

### OCF[4:1] — Output Compare Flags

An output compare flag is set each time TCNT matches the corresponding TOC register. OCF[4:1] correspond to OC[4:1].

### ICF[3:1] — Input Capture Flags

A flag is set each time a selected edge is detected at the corresponding input capture pin. ICF[3:1] correspond to IC[3:1].

### TOF — Timer Overflow Flag

This flag is set each time TCNT advances from a value of \$FFFF to \$0000.

### PAOVF — Pulse Accumulator Overflow Flag

This flag is set each time the pulse accumulator counter advances from a value of \$FF to \$00.

### PAIF — Pulse Accumulator Flag

In event counting mode, this flag is set when an active edge is detected on the PAI pin. In gated time accumulation mode, PAIF is set at the end of the timed period.

## CFORC/PWMC — Compare Force Register/PWM Control Register C

**\$YFF924**

15	11	10	9	8	7	6	4	3	2	1	0
FOC	0	FPWMA	FPWMB	PPROUT	PPR	SFA	SFB	F1A	F1B		

RESET:

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

Setting a bit in CFORC causes a specific output on OC or PWM pins. PWMC sets PWM operating conditions.

## 7 Summary of Changes

This is a complete revision, with complete reprint. All known errors in the publication have been corrected. The following summary lists significant changes. Typographical errors that do not affect content are not annotated.

Page 2	Revised ordering information.
Page 5	New block diagram drawn.
Page 6	New 132-pin assignment diagram drawn.
Page 7	New 144-pin assignment diagram drawn.
Page 8	New address map drawn.
Page 9-13	Added Signal Description section.
Page 14-46	Expanded and revised SIM section. Made all register diagrams and bit mnemonics consistent. Incorporated new information concerning the system clock, resets, interrupts, and chip-select circuits.
Page 47-55	Expanded and revised CPU section. Made all register diagrams and bit mnemonics consistent. Revised instruction set summary information.
Page 56-76	Expanded and revised QSM section. Made all register diagrams and bit mnemonics consistent. Added information concerning SPI and SCI operation.
Page 77-89	Expanded and revised GPT section. Made all register diagrams and bit mnemonics consistent. Added information concerning input capture, output compare, and PWM operation.