



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, UART/USART
Peripherals	-
Number of I/O	47
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x8/10b
Oscillator Type	External
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/mc68334gcf16">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68334gcf16</a>

# TABLE OF CONTENTS

Section	Page
<b>1 Introduction</b>	<b>1</b>
1.1 Features .....	3
1.2 Block Diagram .....	3
1.3 Pin Assignments .....	5
1.4 Address Map .....	6
1.5 Intermodule Bus .....	6
<b>2 Signal Descriptions</b>	<b>7</b>
2.1 Pin Characteristics .....	7
2.2 MCU Power Connections .....	8
2.3 MCU Driver Types .....	8
2.4 Signal Characteristics .....	9
2.5 Signal Function .....	10
<b>3 System Integration Module</b>	<b>11</b>
3.1 Overview .....	11
3.2 System Configuration and Protection Block .....	13
3.3 System Clock .....	19
3.4 External Bus Interface .....	24
3.5 Chip Selects .....	28
3.6 General-Purpose Input/Output .....	35
3.7 Resets .....	37
3.8 Interrupts .....	40
3.9 Factory Test Block .....	42
<b>4 Central Processor Unit</b>	<b>43</b>
4.1 Overview .....	43
4.2 Programming Model .....	43
4.3 Status Register .....	44
4.4 Data Types .....	45
4.5 Addressing Modes .....	45
4.6 Instruction Set Summary .....	46
4.7 Background Debugging Mode .....	49
<b>5 Time Processor Unit</b>	<b>51</b>
5.1 Overview .....	51
5.2 Programmer's Model .....	51
5.3 TPU Components .....	52
5.4 TPU Operation .....	54
5.5 Emulation Support .....	55
5.6 Time Functions .....	55
5.7 TPU Registers .....	57
<b>6 Analog-to-Digital Converter Module</b>	<b>63</b>
6.1 Analog Subsystem .....	63
6.2 Digital Control Subsystem .....	63
6.3 ADC Address Map .....	64
6.4 ADC Registers .....	66
<b>7 Standby RAM with TPU Emulation</b>	<b>72</b>
7.1 Overview .....	72
7.2 TPURAM Register Block .....	72
7.3 TPURAM Registers .....	72
7.4 TPURAM Operation .....	73
<b>8 Summary of Changes</b>	<b>74</b>

## 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
- 8/10-Bit Analog-to-Digital Converter (ADC)
  - Seven Analog/Digital Input Pins (Eighth Channel Connected to V<sub>SSA</sub>)
  - Eight Result Registers
  - Eight Conversion Modes
  - Three Result Alignment Modes
  - One 7-Bit Digital Input Port
- Time Processor Unit (TPU)
  - Dedicated Microengine Operating Independently of CPU32
  - 16 Independent, Programmable Channels and Pins
  - Any Channel can Perform any Microcoded Time Function
  - Two Timer Count Registers with Programmable Prescalers
  - Selectable Channel Priority Levels
- 1-Kbyte Standby RAM with TPU Emulation (TPURAM)
  - External Standby Voltage Supply Input
  - Can be Used as Standby RAM or TPU Microcode Emulation RAM

## 1.2 Block Diagram

**Table 2 MCU Pin Characteristics**

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
RESET	Bo	Y	Y	—	—
RMC	B	Y	N	I/O	PE3
SIZ[1:0]	B	Y	N	I/O	PE[7:6]
TPUCH[15:0]	A	Y	Y	—	—
TSC	—	Y	Y	—	—
T2CLK	A	Y	Y	—	—
V <sub>RH</sub> <sup>4</sup>	—	—	—	—	—
V <sub>RL</sub> <sup>4</sup>	—	—	—	—	—
XFC <sup>3</sup>	—	—	—	—	—
XTAL <sup>3</sup>	—	—	—	—	—

**NOTES**

1. DATA[15:0] are synchronized during reset only. MODCLK and ADC pins are synchronized only when used as input port pins.
2. BERR, HALT only synchronized if late BERR or HALT.
3. EXTAL, XFC, and XTAL are clock reference connections.
4. V<sub>RH</sub> and V<sub>RL</sub> are ADC reference voltage points.

## 2.2 MCU Power Connections

**Table 3 MCU Power Connections**

Pin	Description
V <sub>STBY</sub>	Standby RAM Power/Clock Synthesizer Power
V <sub>DDSYN</sub>	Clock Synthesizer Power
V <sub>DDA</sub> /V <sub>SSA</sub>	A/D Converter Power
V <sub>RH</sub> /V <sub>RL</sub>	A/D Reference Voltage
V <sub>SSE</sub> /V <sub>DDE</sub>	External Periphery Power (Source and Drain)
V <sub>SSI</sub> /V <sub>DDI</sub>	Internal Module Power (Source and Drain)

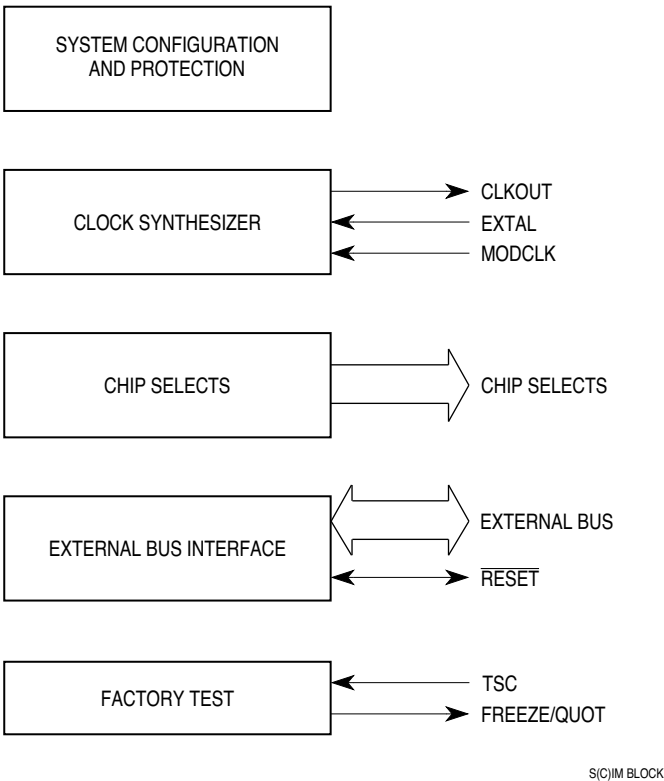
## 2.3 MCU Driver Types

**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 asserted output before high impedance is established, to ensure rapid rise time. An external holding resistor is required to maintain logic level while in the high-impedance state. Pins with this type of driver may only go into high-impedance state under certain conditions. The TSC signal can put all pins with this type of driver in high-impedance state.
Bo	O	Type B output that can be operated in an open-drain mode.

### 3 System Integration Module

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



**Figure 4 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.

**Table 7 SIM Address Map**

Access	Address	15	8	7	0
S	\$YFFA00	SIM MODULE CONFIGURATION REGISTER (SIMCR)			
S	\$YFFA02	FACTORY TEST REGISTER (SIMTR)			
S	\$YFFA04	CLOCK SYNTHESIZER CONTROL REGISTER (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 REGISTER (PICR)			
S	\$YFFA24	PERIODIC INTERRUPT TIMING REGISTER (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)			
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)			

#### EXOFF — External Clock Off

- 0 = The CLKOUT pin is driven from an internal clock source.
- 1 = The CLKOUT pin is placed in a high-impedance state.

#### FRZSW — Freeze Software Enable

- 0 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters continue to run.
- 1 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters are disabled, preventing interrupts during software debug.

#### FRZBM — Freeze Bus Monitor Enable

- 0 = When FREEZE is asserted, the bus monitor continues to operate.
- 1 = When FREEZE is asserted, the bus monitor is disabled.

#### SLVEN — Factory Test Mode Enabled

This bit is a read-only status bit that reflects the state of DATA11 during reset.

- 0 = IMB is not available to an external master.
- 1 = An external bus master has direct access to the IMB.

#### SHEN[1:0] — Show Cycle Enable

This field determines what the EBI does with the external bus during internal transfer operations. A show cycle allows internal transfers to be externally monitored. The table below shows whether show cycle data is driven externally, and whether external bus arbitration can occur. To prevent bus conflict, external peripherals must not be enabled during show cycles.

SHEN	Action
00	Show cycles disabled, external arbitration enabled
01	Show cycles enabled, external arbitration disabled
10	Show cycles enabled, external arbitration enabled
11	Show cycles enabled, external arbitration enabled, internal activity is halted by a bus grant

#### SUPV — Supervisor/Unrestricted Data Space

The SUPV bit places the SIM global registers in either supervisor or user data space.

- 0 = Registers with access controlled by the SUPV bit are accessible from either the user or supervisor privilege level.
- 1 = Registers with access controlled by the SUPV bit are restricted to supervisor access only.

#### MM — Module Mapping

- 0 = Internal modules are addressed from \$7FF000 – \$7FFFFF.
- 1 = Internal modules are addressed from \$FFF000 – \$FFFFFF.

#### IARB[3:0] — Interrupt Arbitration Field

Each module that can generate interrupt requests has an interrupt arbitration (IARB) field. Arbitration between interrupt requests of the same priority is performed by serial contention between IARB field bit values. Contention must take place whenever an interrupt request is acknowledged, even when there is only a single pending request. An IARB field must have a nonzero value for contention to take place. If an interrupt request from a module with an IARB field value of %0000 is recognized, the CPU processes a spurious interrupt exception. Because the SIM routes external interrupt requests to the CPU, the SIM IARB field value is used for arbitration between internal and external interrupts of the same priority. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000, which prevents SIM interrupts from being discarded during initialization.

### 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.

#### SYPCCR — System Protection Control Register

**\$YFFA21**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								SWE	SWP	SWT		HME	BME	BMT	
RESET:								1	MODCLK	0	0	0	0	0	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.

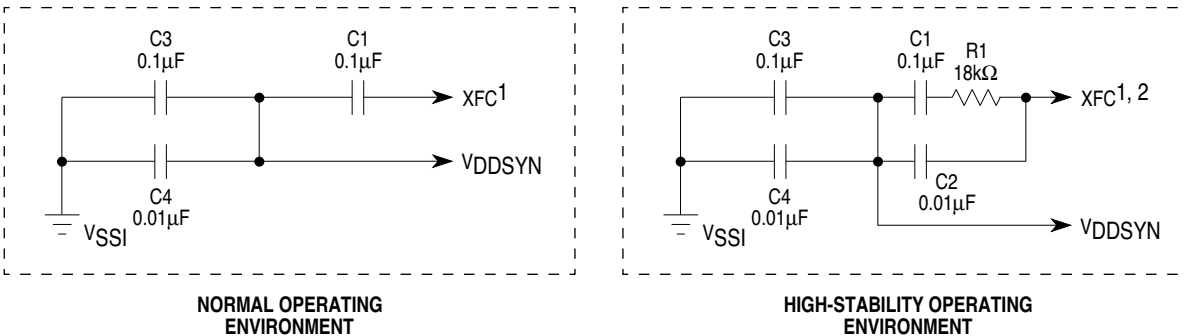
1 = Enable bus monitor function for an internal to external bus cycle.

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





1. MAINTAIN LOW LEAKAGE ON THE XFC NODE.
2. RECOMMENDED LOOP FILTER FOR REDUCED SENSITIVITY TO LOW-FREQUENCY NOISE.

16/32 XFC CONN

**Figure 8 System Clock Filter Networks**

The synthesizer locks when VCO frequency is equal to EXTAL frequency. Lock time is affected by the filter time constant and by the amount of difference between the two comparator inputs. Whenever comparator input changes, the synthesizer must relock. Lock status is shown by the SLOCK bit in SYNCR. During power-up, the MCU does not come out of reset state until the synthesizer locks. Crystal type, characteristic frequency, and layout of external oscillator circuitry affect lock time.

When the clock synthesizer is used, control register SYNCR determines operating frequency and various modes of operation. The SYNCR W bit controls a three-bit prescaler in the feedback divider. Setting W increases VCO speed by a factor of four. The SYNCR Y field determines the count modulus for a modulo 64 down counter, causing it to divide by a value of Y + 1. When W or Y values change, VCO frequency changes, and there is a VCO relock delay. The SYNCR X bit controls a divide-by-two circuit that is not in the synthesizer feedback loop. When X = 0 (reset state), the divider is enabled, and system clock frequency is one-fourth VCO frequency; setting X disables the divider, doubling clock speed without changing VCO speed. There is no relock delay when clock speed is changed by the X bit.

Clock frequency is determined by SYNCR bit settings as follows:

$$F_{\text{SYSTEM}} = F_{\text{REFERENCE}} [4(Y + 1)(2^{2W + X})]$$

The reset state of SYNCR (\$3F00) produces a modulus-64 count.

For the device to perform correctly, system clock and VCO frequencies selected by the W, X, and Y bits must be within the limits specified for the MCU. Do not use a combination of bit values that selects either an operating frequency or a VCO frequency greater than the maximum specified values.

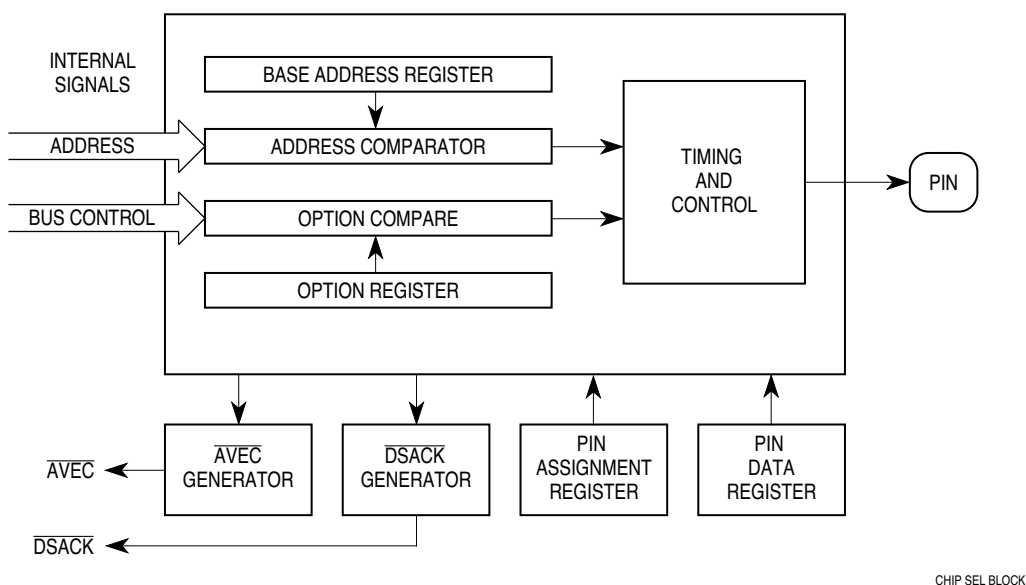
### 3.3.3 External Bus Clock

The state of the external clock division bit (EDIV) in SYNCR determines clock rate for the external bus clock signal (ECLK) available on pin ADDR23. ECLK is a bus clock for MC6800 devices and peripherals. ECLK frequency can be set to system clock frequency divided by eight or system clock frequency divided by sixteen. The clock is enabled by the CS10 field in chip select pin assignment register 1 (CSPAR1). ECLK operation during low-power stop is described in the following paragraph. Refer to **3.5 Chip Selects** for more information about the external bus clock.

### 3.5 Chip Selects

Typical microcontrollers require additional hardware to provide external chip-select and address de-code signals. The MCU includes 12 programmable chip-select circuits that can provide 2- to 20-clock-cycle access to external memory and peripherals. Address block sizes of two Kbytes to one Mbyte can be selected. The figure below is a functional diagram of a chip-select circuit.

Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobe, or interrupt acknowledge signals. Chip select logic can also generate  $\overline{DSACK}$  and  $\overline{AVEC}$  signals internally. Each signal can also be synchronized with the ECLK signal available on ADDR23.



**Figure 10 Chip-Select Circuit Block Diagram**

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low. If a chip-select function is given the same address as a microcontroller module or an internal memory array, an access to that address goes to the module or array, and the chip-select signal is not asserted. The external address and data buses do not reflect the internal access.

All chip-select circuits are configured for operation out of reset. However, all chip-select signals except  $\overline{CSBOOT}$  are disabled, and cannot be asserted until the BYTE field in the corresponding option register is programmed to a nonzero value, selecting a transfer size. The chip-select option register must not be written until a base address has been written to a proper base address register.  $\overline{CSBOOT}$  is automatically asserted out of reset. Alternate functions for chip-select pins are enabled if appropriate data bus pins are held low at the release of the reset signal.

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

Port size determines the way in which bus transfers to an external address are allocated. Port size of eight bits or sixteen bits can be selected when a pin is assigned as a chip select. Port size and transfer size affect how the chip-select signal is asserted. Refer to **3.5.4 Option Registers** for more information.

Out of reset, chip-select pin function is determined by the logic level on a corresponding data bus pin. These pins have weak internal pull-up drivers, but can be held low by external devices. Either 16-bit chip-select function (%11) or alternate function (%01) can be selected during reset. All pins except the boot ROM select pin ( $\overline{\text{CSBOOT}}$ ) are disabled out of reset. There are twelve chip-select functions and only eight associated data bus pins. There is not a one-to-one correspondence. The table below shows reset states.

**Table 18 Reset Pin Function of  $\overline{\text{CS}}[10:6]$**

Data Bus Pins at Reset					Chip-Select/Address Bus Pin Function				
DATA7	DATA6	DATA5	DATA4	DATA3	$\overline{\text{CS}}10/ADDR23$	$\overline{\text{CS}}9/ADDR22$	$\overline{\text{CS}}8/ADDR21$	$\overline{\text{CS}}7/ADDR20$	$\overline{\text{CS}}6/ADDR19$
1	1	1	1	1	$\overline{\text{CS}}10$	$\overline{\text{CS}}9$	CS8	$\overline{\text{CS}}7$	CS6
1	1	1	1	0	$\overline{\text{CS}}10$	$\overline{\text{CS}}9$	CS8	$\overline{\text{CS}}7$	ADDR19
1	1	1	0	X	$\overline{\text{CS}}10$	$\overline{\text{CS}}9$	$\overline{\text{CS}}8$	ADDR20	ADDR19
1	1	0	X	X	$\overline{\text{CS}}10$	$\overline{\text{CS}}9$	ADDR21	ADDR20	ADDR19
1	0	X	X	X	$\overline{\text{CS}}10$	ADDR22	ADDR21	ADDR20	ADDR19
0	X	X	X	X	ADDR23	ADDR22	ADDR21	ADDR20	ADDR19

The  $\overline{\text{CSBOOT}}$  signal is normally enabled out of reset. The state of the DATA0 line during reset determines what port width  $\overline{\text{CSBOOT}}$  uses. If DATA0 is held high (either by the weak internal pull-up driver or by an external pull-up device), 16-bit width is selected. If DATA0 is held low, 8-bit port size is selected.

A pin programmed as a discrete output drives an external signal to the value specified in the pin data register. No discrete output function is available on pins  $\overline{\text{CSBOOT}}$ ,  $\overline{\text{BR}}$ ,  $\overline{\text{BG}}$ , or  $\overline{\text{BGACK}}$ . ADDR23 provides ECLK 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  $\overline{\text{DSACK}}$  or  $\overline{\text{AVEC}}$  internally on an address and control signal match.

### 3.5.3 Base Address Registers

Each chip select has an associated base address register. A base address is the lowest address in the block of addresses enabled by a chip select. Block size is the extent of the address block above the base address. Block size is determined by the value contained in a BLKSZ field. Block addresses for different chip selects can overlap.

The BLKSZ field determines which bits in the base address field are compared to corresponding bits on the address bus during an access. Provided other constraints determined by option register fields are also satisfied, when a match occurs, the associated chip-select signal is asserted.

The chip-select address compare logic uses only the most significant bits to match an address within a block. The value of the base address must be a multiple of block size. Base address register diagrams show how base register bits correspond to address lines.

After reset, the MCU fetches the initialization routine from the address contained in the reset vector, located beginning at address \$000000 of program space. To support bootstrap operation from reset, the base address field in chip-select base address register boot (CSBARBT) has a reset value of all zeros. A memory device containing the reset vector and initialization routine can be automatically enabled by  $\overline{\text{CSBOOT}}$  after a reset. The block size field in CSBARBT has a reset value of 512 Kbytes.

**Table 21 Reset Mode Selection**

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

Data lines have weak internal pull-up drivers. External bus loading can overcome the weak internal pull-up drivers on data bus lines, and hold pins low during reset. Use an active device to hold data bus lines low. Data bus configuration logic must release the bus before the first bus cycle after reset to prevent conflict with external memory devices. The first bus cycle occurs ten CLKOUT cycles after RESET is released. If external mode selection logic causes a conflict of this type, an isolation resistor on the driven lines may be required.

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

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

**Table 22 Module Pin Functions**

Module	Pin Mnemonic	Function
CPU32	DSI/IFETCH	DSI/IFETCH
	DSO/IPIPE	DSO/IPIPE
	BKPT/DSCLK	BKPT/DSCLK
TPU	TPUCH[15:0]	TPU Input
	T2CLK	TCR2 Clock
ADC	AN[6:0]/PADA[6:0]	Analog Input

### 3.7.3 Reset Timing

The RESET input must be asserted for a specified minimum period in order for reset to occur. External 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 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 ten cycles, then it is tested again. The process repeats until  $\overline{\text{RESET}}$  is released.

### 3.7.4 Power-On Reset

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

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

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

### 3.7.5 Use of Three State Control Pin

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

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

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

When TSC assertion takes effect, internal signals are forced to values that can cause inadvertent mode selection. Once the output drivers change state, the MCU must be powered down and restarted before normal operation can resume.

### 3.8 Interrupts

Interrupt recognition and servicing involve complex interaction between the central processing unit, the system integration module, and a device or module requesting interrupt service.

The CPU32 provides eight levels of interrupt priority. All interrupts with priorities less than seven can be masked by the interrupt priority (IP) field in status register.

There are seven interrupt request signals ( $\overline{\text{IRQ}}[7:1]$ ). These signals are used internally on the IMB, and there are corresponding pins for external interrupt service requests. The CPU treats all interrupt requests as though they come from internal modules — external interrupt requests are treated as interrupt service requests from the SIM. Each of the interrupt request signals corresponds to an interrupt priority level.  $\overline{\text{IRQ}}1$  has the lowest priority and  $\overline{\text{IRQ}}7$  the highest.

Interrupt recognition is determined by interrupt priority level and interrupt priority mask value. The interrupt priority mask consists of three bits in the CPU32 status register. Binary values %000 to %111 provide eight priority masks. Masks prevent an interrupt request of a priority less than or equal to the mask value from being recognized and processed.  $\overline{\text{IRQ}}7$ , however, is always recognized, even if the mask value is %111.

$\overline{\text{IRQ}}[7:1]$  are active-low level-sensitive inputs. The low on the pin must remain asserted until an interrupt acknowledge cycle corresponding to that level is detected.

$\overline{\text{IRQ}}7$  is transition-sensitive as well as level-sensitive: a level-7 interrupt is not detected unless a falling edge transition is detected on the  $\overline{\text{IRQ}}7$  line. This prevents redundant servicing and stack overflow. A nonmaskable interrupt is generated each time  $\overline{\text{IRQ}}7$  is asserted as well as each time the priority mask changes from %111 to a lower number while  $\overline{\text{IRQ}}7$  is asserted.

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 with a priority equal to or lower than the current IP mask value is made, the CPU32 does not recognize the occurrence of the request. If simultaneous interrupt requests of different priorities are made, and both have a priority greater than the mask value, the CPU32 recognizes the higher-level request.

#### 3.8.1 Interrupt Acknowledge and Arbitration

When the CPU32 detects one or more interrupt requests of a priority higher than the interrupt priority mask value, it places the interrupt request level on the address bus and initiates a CPU space read cycle. The request level serves two purposes: it is decoded by modules or external devices that have requested interrupt service, to determine whether the current interrupt acknowledge cycle pertains to them, and it is latched into the interrupt priority mask field in the CPU32 status register, to preclude further interrupts of lower priority during interrupt service.

Modules or external devices that have requested interrupt service must decode the interrupt priority mask value placed on the address bus during the interrupt acknowledge cycle and respond if the priority of the service request corresponds to the mask value. However, before modules or external devices respond, interrupt arbitration takes place.

Arbitration is performed by means of serial contention between values stored in individual module interrupt arbitration (IARB) fields. Each module that can make an interrupt service request, including the SIM, has an IARB field in its configuration register. IARB fields can be assigned values from %0000 to

3. The request level is latched from the address bus into the interrupt priority mask field in the status or condition code register.
- D. Modules that have requested interrupt service decode the priority value in ADDR[3:1]. If request priority is the same as acknowledged priority, arbitration by IARB contention takes place.
- E. After arbitration, the interrupt acknowledge cycle is completed in one of the following ways:
  1. When there is no contention (IARB = %0000), the spurious interrupt monitor asserts  $\overline{\text{BERR}}$ , and the CPU generates the spurious interrupt vector number.
  2. The dominant interrupt source supplies a vector number and  $\overline{\text{DSACK}}$  signals appropriate to the access. The CPU acquires the vector number.
  3. The  $\overline{\text{AVEC}}$  signal is asserted (the signal can be asserted by the dominant interrupt source or the pin can be tied low), and the CPU generates an autovector number corresponding to interrupt priority.
  4. The bus monitor asserts  $\overline{\text{BERR}}$  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.

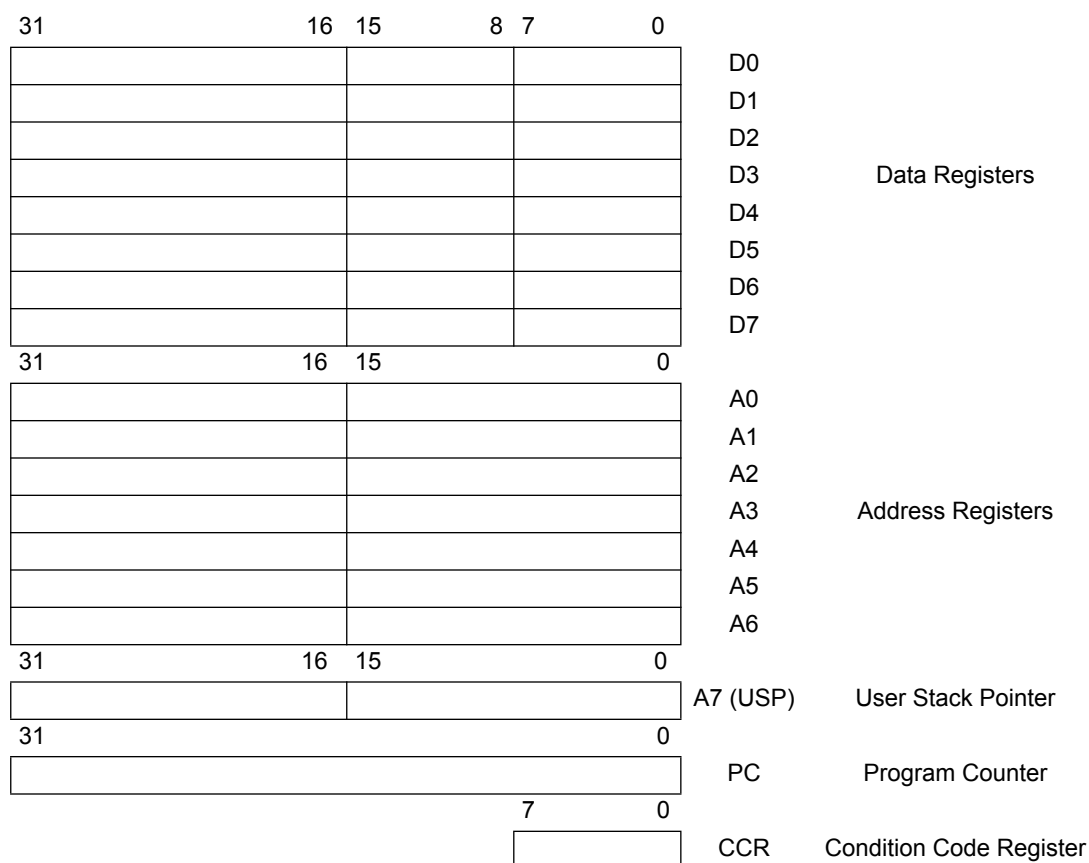
### 3.9 Factory Test Block

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

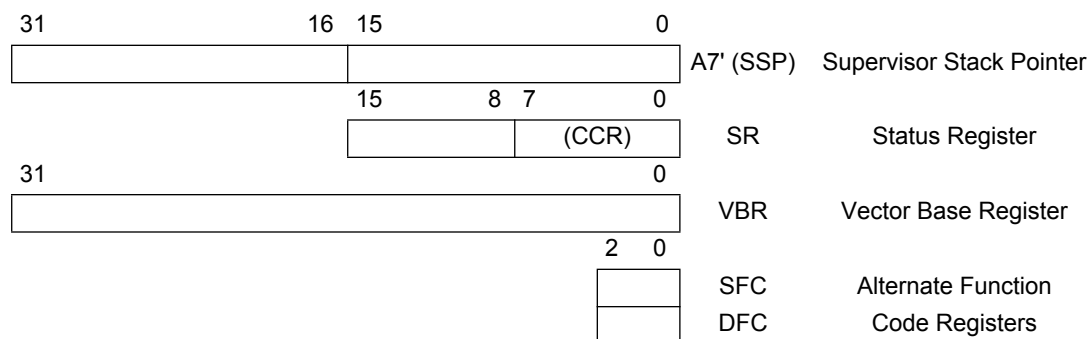
Test submodule registers are intended for Freescale use. Register names and addresses are provided to indicate that these addresses are occupied.

<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>

## User Programming Model



## Supervisor Programming Model Supplement

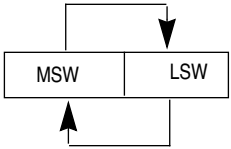


### 4.3 Status Register

The status register contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The lower byte containing the condition codes is the only portion of the register available at the user privilege level; it is referenced as the condition code register (CCR) in user programs. At the supervisor privilege level, software can access the full status register, including the interrupt priority mask and additional control bits.



**Table 23 Instruction Set Summary**

Instruction	Syntax	Operand Size	Operation
SUBA	$\langle ea \rangle, An$	16, 32	Destination – Source $\Rightarrow$ Destination
SUBI	$\# \langle data \rangle, \langle ea \rangle$	8, 16, 32	Destination – Data $\Rightarrow$ Destination
SUBQ	$\# \langle data \rangle, \langle ea \rangle$	8, 16, 32	Destination – Data $\Rightarrow$ Destination
SUBX	$Dn, Dn$ $- (An), - (An)$	8, 16, 32 8, 16, 32	Destination – Source – X $\Rightarrow$ Destination
SWAP	$Dn$	16	
TBLS/TBLU	$\langle ea \rangle, Dn$ Dym : Dyn, Dn	8, 16, 32	Dyn – Dym $\Rightarrow$ Temp (Temp * Dn [7:0]) $\Rightarrow$ Temp (Dym * 256) + Temp $\Rightarrow$ Dn
TBLSN/TBLUN	$\langle ea \rangle, Dn$ Dym : Dyn, Dn	8, 16, 32	Dyn – Dym $\Rightarrow$ Temp (Temp * Dn [7:0]) / 256 $\Rightarrow$ Temp Dym + Temp $\Rightarrow$ Dn
TRAP	$\# \langle data \rangle$	none	SSP – 2 $\Rightarrow$ SSP; format/vector offset $\Rightarrow$ (SSP); SSP – 4 $\Rightarrow$ SSP; PC $\Rightarrow$ (SSP); SR $\Rightarrow$ (SSP); vector address $\Rightarrow$ PC
TRAPcc	none $\# \langle data \rangle$	none 16, 32	If cc true, then TRAP exception
TRAPV	none	none	If V set, then overflow TRAP exception
TST	$\langle ea \rangle$	8, 16, 32	Source – 0, to set condition codes
UNLK	$An$	32	$An \Rightarrow SP$ ; $(SP) \Rightarrow An$ , $SP + 4 \Rightarrow SP$

NOTE:

1. Privileged instruction

## 4.7 Background Debugging Mode

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

**Table 24 Background Debugging Command Summary**

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.

### 5.3.3 Scheduler

When a service request is received, the scheduler determines which TPU channel is serviced by the microengine. A channel can request service for one of four reasons: for host service, for a link to another channel, for a match event, or for a capture event. The host system assigns each active channel one of three priorities: high, middle, or low. When multiple service requests are received simultaneously, a priority-scheduling mechanism grants service based on channel number and assigned priority.

### 5.3.4 Microengine

The microengine is composed of a control store and an execution unit. Control-store ROM holds the microcode for each factory-masked time function. When assigned to a channel by the scheduler, the execution unit executes microcode for a function assigned to that channel by the host CPU. Microcode can also be executed from the TPURAM module instead of the control store. The TPURAM module allows emulation and development of custom TPU microcode without the generation of a microcode ROM mask. Refer to **5.5 Emulation Support** for more information.

### 5.3.5 Host Interface

Host interface registers allow communication between the host CPU and the TPU, both before and during execution of a time function. The registers are accessible from the IMB through the TPU bus interface unit.

### 5.3.6 Parameter RAM

Parameter RAM occupies 256 bytes at the top of the system address map. Channel parameters are organized as 128 16-bit words. Although all parameter word locations in RAM can be accessed by all channels, only 100 are normally used: channels 0 to 13 use six parameter words, while channels 14 and 15 each use eight parameter words. The parameter RAM address map shows how parameter words are organized in memory.

The host CPU specifies function parameters by writing the appropriate RAM address. The TPU reads the RAM to determine channel operation. The TPU can also store information to be read by the CPU in RAM. Detailed descriptions of the parameters required by each time function are beyond the scope of this technical summary. Refer to the *TPU Reference Manual* (TPURM/AD) for more information.

For pre-programmed functions, one of the parameter words associated with each channel contains three channel control fields. These fields perform the following functions:

PSC — Forces the output level of the pin.

PAC — For input capture, PAC specifies the edge transition to be detected. For output comparison, PAC specifies the logic level to be output when a match occurs.

TBS — Specifies channel direction (input or output) and assigns a time base to the input capture and output compare functions of the channel.

## ADSTAT — ADC Status Register

**\$YFF70E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCF	NOT USED				CCTR			CCF							
RESET:															
0					0	0	0	0	0	0	0	0	0	0	0

ADSTAT contains information related to the status of a conversion sequence.

### SCF — Sequence Complete Flag

0 = Sequence not complete

1 = Sequence complete

SCF is set at the end of the conversion sequence when SCAN is cleared, and at the end of the first conversion sequence when SCAN is set. SCF is cleared when ADCTL1 is written and a new conversion sequence begins.

### CCTR[2:0] — Conversion Counter Field

This field reflects the contents of the conversion counter pointer in either four or eight count conversion sequence. The value corresponds to the number of the next result register to be written, and thus indicates which channel is being converted.

### CCF[7:0] — Conversion Complete Field

Each bit in this field corresponds to an A/D result register (CCF7 to RSLT7, etc.). A bit is set when conversion for the corresponding channel is complete, and remains set until the result register is read.

## RSLT[7:0] — A/D Result Registers

**\$YFF710–\$YFF73E**

The result registers store data after conversion is complete. Each register can be read from three different addresses in the register block. Data format depends on the address from which the result register is read.

### RJURR — Unsigned Right-Justified Format

**\$YFF710–\$YFF71F**

Conversion result is unsigned right-justified data. Bits [9:0] are used for 10-bit resolution, bits [7:0] are used for 8-bit resolution (bits [9:8] are zero). Bits [15:10] always return zero when read.

### LJSRR — Signed Left-Justified Format

**\$YFF720–\$YFF72F**

Conversion result is signed left-justified data. Bits [15:6] are used for 10-bit resolution, bits [15:8] are used for 8-bit resolution (bits [7:6] are zero). Although the ADC is unipolar, it is assumed that the zero point is halfway between low and high reference when this format is used. For positive input, bit 15 = 0, for negative input, bit 15 = 1. Bits [5:0] always return zero when read.

### LJURR — Unsigned Left-Justified Format

**\$YFF730–\$YFF73F**

Conversion result is unsigned left-justified data. Bits [15:6] are used for 10-bit resolution, bits [15:8] are used for 8-bit resolution (bits [7:6] are zero). Bits [5:0] always return zero when read.

