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	Active
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	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68334geh16

1.3 Pin Assignments

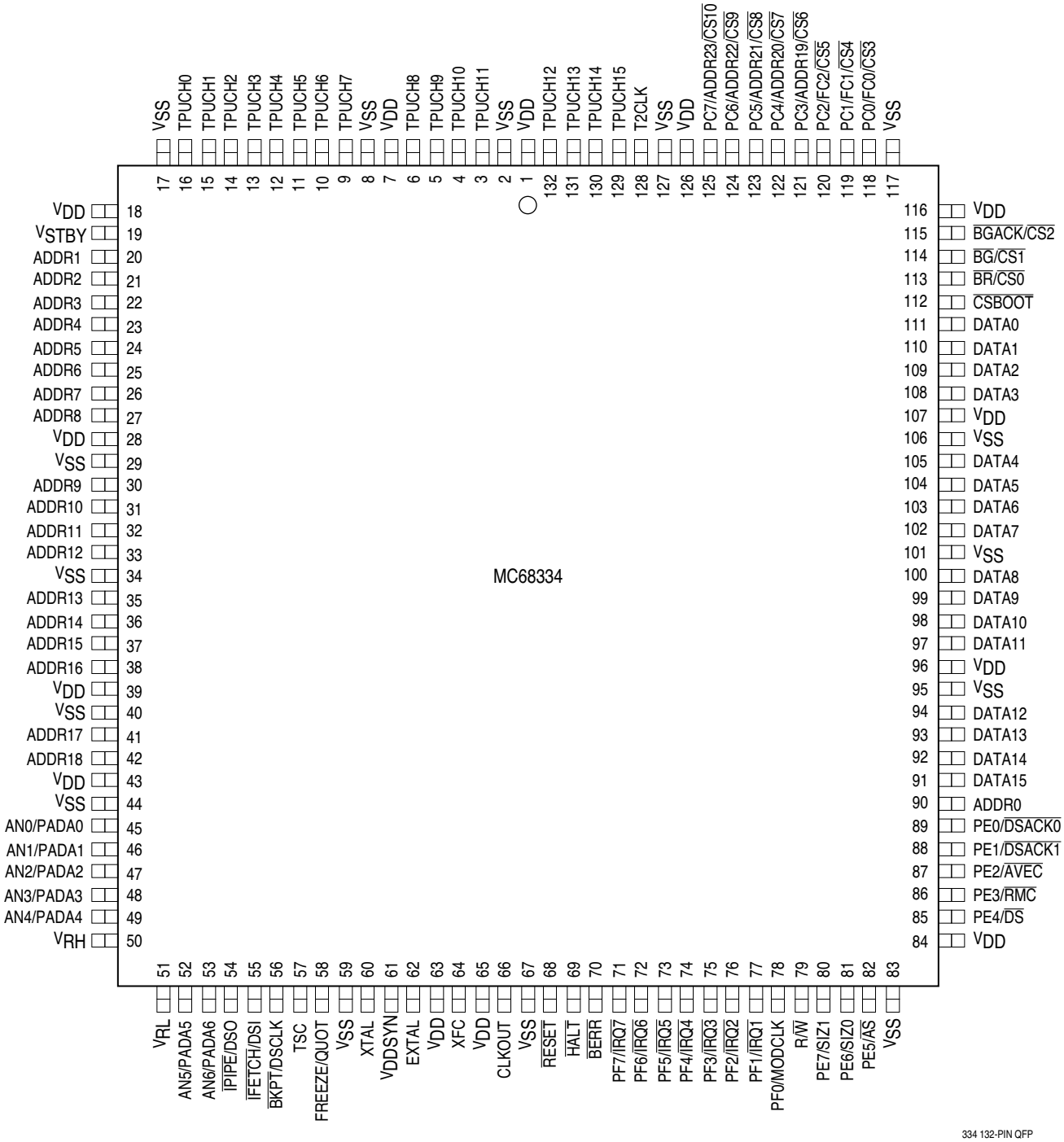


Figure 2 MC68334 132-Pin QFP Pin Assignments

1.4 Address Map

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

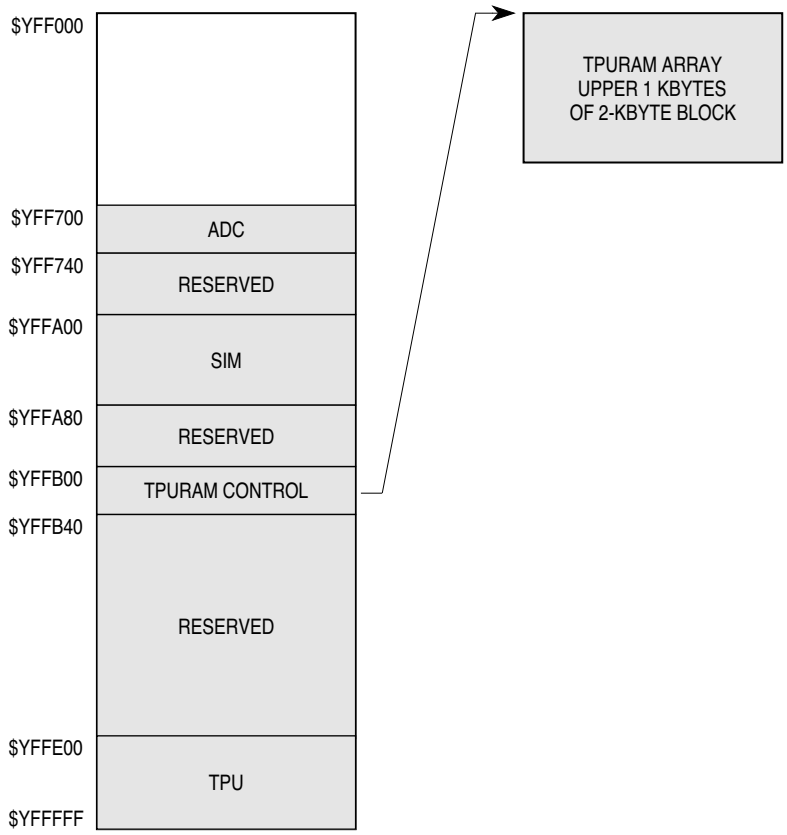


Figure 3 MC68334 Address Map

1.5 Intermodule Bus

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

Table 2 MCU Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
$\overline{\text{RESET}}$	Bo	Y	Y	—	—
$\overline{\text{RMC}}$	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_{RH}[4]	—	—	—	—	—
V_{RL}[4]	—	—	—	—	—
XFC[3]	—	—	—	—	—
XTAL[3]	—	—	—	—	—

NOTES

1. DATA[15:0] are synchronized during reset only. MODCLK and ADC pins are synchronized only when used as input port pins.
2. $\overline{\text{BERR}}$, HALT only synchronized if late $\overline{\text{BERR}}$ or HALT.
3. EXTAL, XFC, and XTAL are clock reference connections.
4. V_{RH} and V_{RL} are ADC reference voltage points.

2.2 MCU Power Connections

Table 3 MCU Power Connections

Pin	Description
V_{STBY}	Standby RAM Power/Clock Synthesizer Power
V_{DDSYN}	Clock Synthesizer Power
V_{DDA}/V_{SSA}	A/D Converter Power
V_{RH}/V_{RL}	A/D Reference Voltage
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 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.

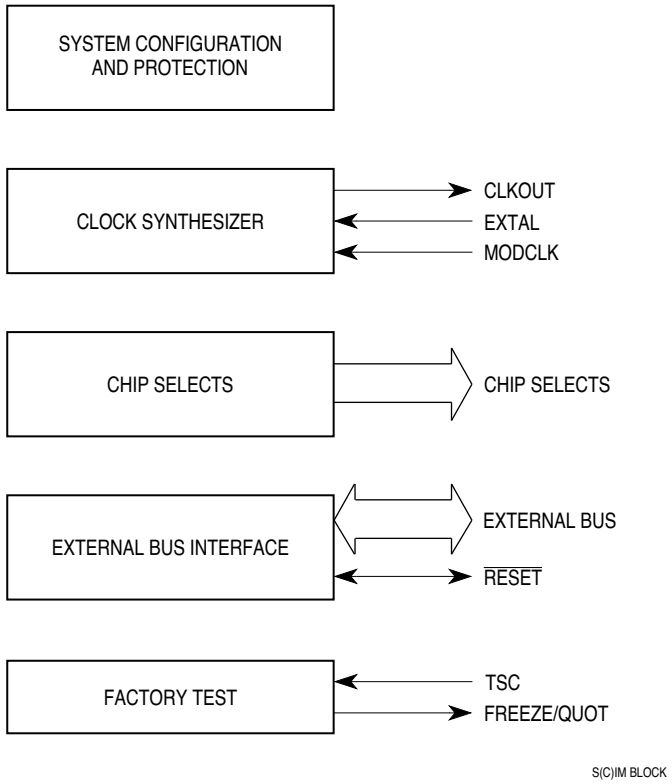


Figure 4 SIM Block Diagram

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.

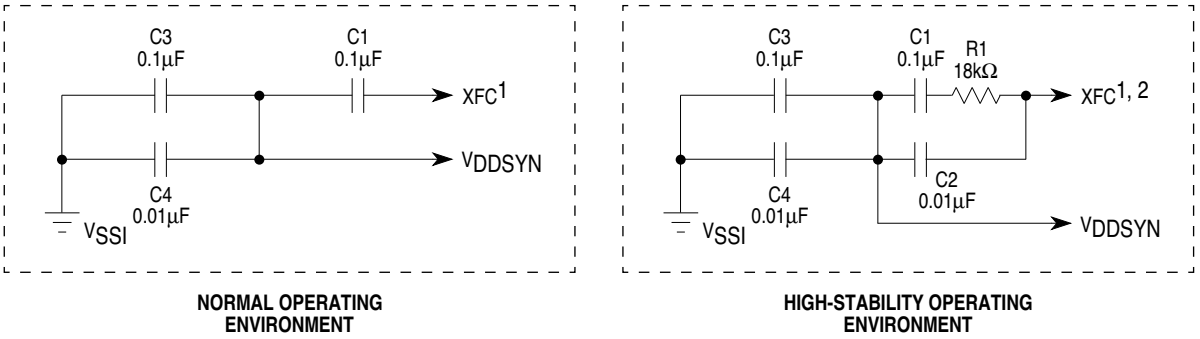
Table 7 SIM Address Map

Access	Address	15 8	7 0
S	$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)	

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

3.2 System Configuration and Protection Block

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.



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:

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

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 $\overline{CS10}$ 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.

3.3.4 Low-Power Operation

Low-power operation is initiated by the CPU32. To reduce power consumption selectively, the CPU can set the STOP bits in each module configuration register. To minimize overall microcontroller power consumption, the CPU can execute the LPSTOP instruction, which causes the SIM to turn off the system clock.

When individual module STOP bits are set, clock signals inside each module are turned off, but module registers are still accessible.

When the CPU executes LPSTOP, a special CPU space bus cycle writes a copy of the current interrupt mask into the clock control logic. The SIM brings the MCU out of low-power operation when either an interrupt of higher priority than the stored mask or a reset occurs.

During a low-power stop, unless the system clock signal is supplied by an external source and that source is removed, the SIM clock control logic and the SIM clock signal (SIMCLK) continue to operate. The periodic interrupt timer and input logic for the $\overline{\text{RESET}}$ and $\overline{\text{IRQ}}$ pins are clocked by SIMCLK. The SIM can also continue to generate the CLKOUT signal while in low-power mode.

The stop mode system integration module clock (STSIM) and stop mode external clock (STEXT) bits in SYNCR determine clock operation during low-power stop. The table below summarizes the effects of STSIM and STEXT. MODCLK value is the logic level on the MODCLK pin during the last reset before LPSTOP execution. Any clock in the off state is held low. If the synthesizer VCO is turned off during LPSTOP, there is a PLL relock delay after the VCO is turned back on.

Table 8 Clock Control

Mode	Pins		SYNCR Bits		Clock Status		
LPSTOP	**MODCLK**	**EXTAL**	**STSIM**	**STEXT**	**SIMCLK**	**CLKOUT**	**ECLK**
No	0	External Clock	X	X	External Clock	External Clock	External Clock
Yes	0	External Clock	0	0	External Clock	Off	Off
Yes	0	External Clock	0	1	External Clock	External Clock	External Clock
Yes	0	External Clock	1	0	External Clock	Off	Off
Yes	0	External Clock	1	1	External Clock	External Clock	External Clock
No	1	Crystal or Reference	X	X	VCO	VCO	VCO
Yes	1	Crystal or Reference	0	0	Crystal or Reference	Off	Off
Yes	1	Crystal or Reference	0	1	Crystal or Reference	Crystal/ Reference	Off
Yes	1	Crystal or Reference	1	0	VCO	Off	Off
Yes	1	Crystal or Reference	1	1	VCO	VCO	VCO

3.3.5 Loss of Reference Signal

The state of the reset enable (RSTEN) bit in SYNCR determines what happens when clock logic detects a reference failure.

When RSTEN is cleared (default state out of reset), the clock synthesizer is forced into an operating condition referred to as limp mode. Limp mode frequency varies from device to device, but maxi-

mum limp frequency does not exceed one half maximum system clock when X = 0, or maximum system clock frequency when X = 1.

When RSTEN is set, the SIM resets the MCU.

The limp status bit (SLIMP) in SYNCR indicates whether the synthesizer has a reference signal. It is set when a reference failure is detected.

3.3.6 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	12	11	10	9	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.

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

3.4.8 Data Transfer Mechanism

The MCU architecture supports byte, word, and long-word operands, allowing access to 8- and 16-bit data ports through the use of asynchronous cycles controlled by the data transfer and size acknowledge inputs (DSACK1 and DSACK0).

3.4.9 Dynamic Bus Sizing

The MCU dynamically interprets the port size of the addressed device during each bus cycle, allowing operand transfers to or from 8- and 16-bit ports. During an operand transfer cycle, the slave device signals its port size and indicates completion of the bus cycle to the MCU through the use of the DSACK0 and DSACK1 inputs, as shown in the following table.

Table 11 Effect of DSACK Signals

DSACK1	DSACK0	Result
1	1	Insert Wait States in Current Bus Cycle
1	0	Complete Cycle — Data Bus Port Size is 8 Bits
0	1	Complete Cycle — Data Bus Port Size is 16 Bits
0	0	Reserved

For example, if the MCU is executing an instruction that reads a long-word operand from a 16-bit port, the MCU latches the 16 bits of valid data and then runs another bus cycle to obtain the other 16 bits. The operation for an 8-bit port is similar, but requires four read cycles. The addressed device uses the DSACK0 and DSACK1 signals to indicate the port width. For instance, a 16-bit device always returns DSACK0 = 1 and DSACK1 = 0 for a 16-bit port, regardless of whether the bus cycle is a byte or word operation.

Dynamic bus sizing requires that the portion of the data bus used for a transfer to or from a particular port size be fixed. A 16-bit port must reside on data bus bits [15:0] and an 8-bit port must reside on data bus bits [15:8]. This minimizes the number of bus cycles needed to transfer data and ensures that the MCU transfers valid data.

The MCU always attempts to transfer the maximum amount of data on all bus cycles. For a word operation, it is assumed that the port is 16 bits wide when the bus cycle begins. Operand bytes are designated as shown in the following figure. OP0 is the most significant byte of a long-word operand, and OP3 is the least significant byte. The two bytes of a word-length operand are OP0 (most significant) and OP1. The single byte of a byte-length operand is OP0.

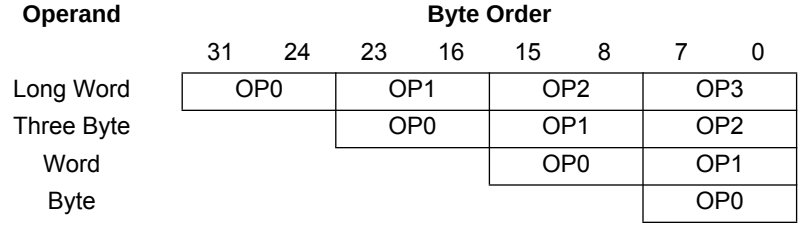


Figure 9 Operand Byte Order

CSBARBT — Chip-Select Base Address Register Boot ROM **$YFFA48**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

CSBAR[10:0] — Chip-Select Base Address Registers **$YFFA4C–$YFFA74**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BLKSZ — Block Size Field

This field determines the size of the block that must be enabled by the chip select. The following table shows bit encoding for the base address registers block size field.

Block Size Field	Block Size	Address Lines Compared
000	2 Kbytes	ADDR[23:11]
001	8 Kbytes	ADDR[23:13]
010	16 Kbytes	ADDR[23:14]
011	64 Kbytes	ADDR[23:16]
100	128 Kbytes	ADDR[23:17]
101	256 Kbytes	ADDR[23:18]
110	512 Kbytes	ADDR[23:19]
111	1 Mbyte	ADDR[23:20]

ADDR[23:11] — Base Address Field

This field sets the starting address of a particular address space. The 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.

3.5.4 Option Registers

The option registers contain eight fields that determine timing of and conditions for assertion of chip-select signals. To assert a chip-select signal, and to provide $\overline{\text{DSACK}}$ or autovector support, other constraints set by fields in the option register and in the base address register must also be satisfied.

CSORBT — Chip-Select Option Register Boot ROM **$YFFA4A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE		R/$\overline{\text{W}}$		STRB	$\overline{\text{DSACK}}$				SPACE		IPL			$\overline{\text{AVEC}}$
RESET:															
0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0

CSOR[10:0] — Chip-Select Option Registers **$YFFA4E–$YFFA76**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE		R/$\overline{\text{W}}$		STRB	$\overline{\text{DSACK}}$				SPACE		IPL			$\overline{\text{AVEC}}$
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23 Instruction Set Summary

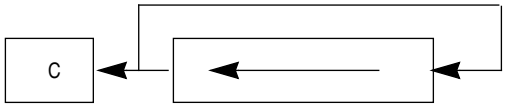
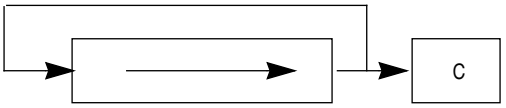
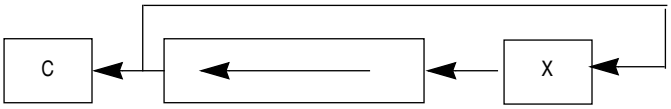
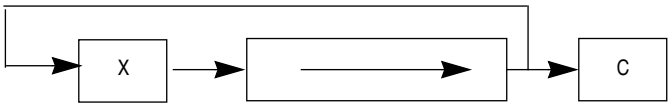
Instruction	Syntax	Operand Size	Operation
MOVEQ	#⟨data⟩, Dn	8 ⇒ 32	Immediate data ⇒ Destination
MOVES[1]	Rn, ⟨ea⟩ ⟨ea⟩, Rn	8, 16, 32	Rn ⇒ Destination using DFC Source using SFC ⇒ Rn
MULS/MULU	⟨ea⟩, Dn ⟨ea⟩, Dl ⟨ea⟩, Dh:Dl	16 ∗ 16 ⇒ 32 32 ∗ 32 ⇒ 32 32 ∗ 32 ⇒ 64	Source ∗ Destination ⇒ Destination (signed or unsigned)
NBCD	⟨ea⟩	8 8	0 – $Destination_{10}$ – X ⇒ Destination
NEG	⟨ea⟩	8, 16, 32	0 – Destination ⇒ Destination
NEGX	⟨ea⟩	8, 16, 32	0 – Destination – X ⇒ Destination
NOP	none	none	PC + 2 ⇒ PC
NOT	⟨ea⟩	8, 16, 32	$\overline{\text{Destination}}$ ⇒ Destination
OR	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source; Destination ⇒ Destination
ORI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Data; Destination ⇒ Destination
ORI to CCR	#⟨data⟩, CCR	16	Source; CCR ⇒ SR
ORI to SR[1]	#⟨data⟩, SR	16	Source; SR ⇒ SR
PEA	⟨ea⟩	32	SP – 4 ⇒ SP; ⟨ea⟩ ⇒ SP
RESET[1]	none	none	Assert $\overline{\text{RESET}}$ line
ROL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROXL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROXR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
RTD	#⟨d⟩	16	(SP) ⇒ PC; SP + 4 + d ⇒ SP
RTE[1]	none	none	(SP) ⇒ SR; SP + 2 ⇒ SP; (SP) ⇒ PC; SP + 4 ⇒ SP; restore stack according to format
RTR	none	none	(SP) ⇒ CCR; SP + 2 ⇒ SP; (SP) ⇒ PC; SP + 4 ⇒ SP
RTS	none	none	(SP) ⇒ PC; SP + 4 ⇒ SP
SBCD	Dn, Dn – (An), – (An)	8 8	$Destination_{10}$ – $Source_{10}$ – X ⇒ Destination
Scc	⟨ea⟩	8	If condition true, then destination bits are set to one; else, destination bits are cleared to zero
STOP[1]	#⟨data⟩	16	Data ⇒ SR; STOP
SUB	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32	Destination – Source ⇒ Destination

Table 25 TPU Address Map

Access	Address	15 8	7 0
S	$YFFE00	TPU MODULE CONFIGURATION REGISTER (TPUMCR)	
S	$YFFE02	TEST CONFIGURATION REGISTER (TCR)	
S	$YFFE04	DEVELOPMENT SUPPORT CONTROL REGISTER (DSCR)	
S	$YFFE06	DEVELOPMENT SUPPORT STATUS REGISTER (DSSR)	
S	$YFFE08	TPU INTERRUPT CONFIGURATION REGISTER (TICR)	
S	$YFFE0A	CHANNEL INTERRUPT ENABLE REGISTER (CIER)	
S	$YFFE0C	CHANNEL FUNCTION SELECTION REGISTER 0 (CFSR0)	
S	$YFFE0E	CHANNEL FUNCTION SELECTION REGISTER 1 (CFSR1)	
S	$YFFE10	CHANNEL FUNCTION SELECTION REGISTER 2 (CFSR2)	
S	$YFFE12	CHANNEL FUNCTION SELECTION REGISTER 3 (CFSR3)	
S/U	$YFFE14	HOST SEQUENCE REGISTER 0 (HSQR0)	
S/U	$YFFE16	HOST SEQUENCE REGISTER 1 (HSQR1)	
S/U	$YFFE18	HOST SERVICE REQUEST REGISTER 0 (HSRR0)	
S/U	$YFFE1A	HOST SERVICE REQUEST REGISTER 1 (HSRR1)	
S	$YFFE1C	CHANNEL PRIORITY REGISTER 0 (CPR0)	
S	$YFFE1E	CHANNEL PRIORITY REGISTER 1 (CPR1)	
S	$YFFE20	CHANNEL INTERRUPT STATUS REGISTER (CISR)	
S	$YFFE22	LINK REGISTER (LR)	
S	$YFFE24	SERVICE GRANT LATCH REGISTER (SGLR)	
S	$YFFE26	DECODED CHANNEL NUMBER REGISTER (DCNR)	

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

5.3 TPU Components

The TPU module consists of two 16-bit time bases, sixteen independent timer channels, a task scheduler, a microengine, and a host interface. In addition, a dual-port parameter RAM is used to pass parameters between the module and the host CPU.

5.3.1 Time Bases

Two 16-bit counters provide reference time bases for all output compare and input capture events. Prescalers for both time bases are controlled by the host CPU via bit fields in the TPU module configuration register (TPUMCR). Timer count registers TCR1 and TCR2 provide access to current counter values. TCR1 and TCR2 can be read/write accessed in microcode, but are not directly available to the host CPU. The TCR1 clock is derived from the system clock. The TCR2 clock can be derived from the system clock or from an external clock input via the T2CLK pin.

5.3.2 Timer Channels

The TPU has 16 independent channels, each connected to an MCU pin. The channels have identical hardware. Each channel consists of an event register and pin control logic. The event register contains a 16-bit capture register, a 16-bit compare/match register, and a 16-bit greater-than-or-equal-to comparator. The direction of each pin, either output or input, is determined by the TPU microengine. Each channel can either use the same time base for match and capture, or can use one time base for match and the other for capture.

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.

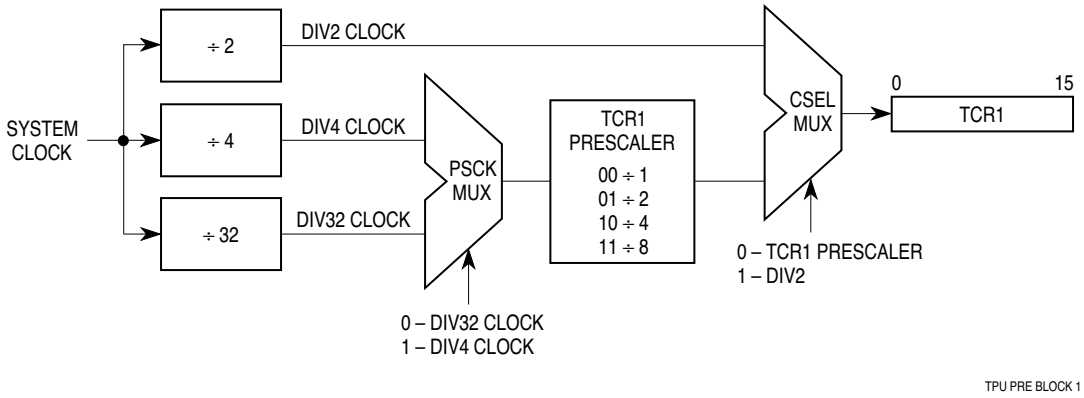


Figure 12 Prescaler Control 1

TCR1 Prescaler	Divide By	PSCK = 0		PSCK = 1	
		Number of Clocks	Rate at 16 MHz	Number of Clocks	Rate at 16 MHz
00	1	32	2 ms	4	250 ns
01	2	64	4 ms	8	500 ns
10	4	128	8 ms	16	1 ms
11	8	256	16 ms	32	2 ms

TCR2P — Timer Count Register 2 Prescaler Control

TCR2 is clocked from the output of a prescaler. If T2CG = 0, the input to the TCR2 prescaler is the external TCR2 clock source. If T2CG = 1, the input is the TPU system clock divided by eight. The TCR2P field specifies the value of the prescaler: 1, 2, 4, or 8. Channels using TCR2 have the capability to resolve down to the TPU system clock divided by 8. The following table is a summary of prescaler output.

TCR2 Prescaler	Divide By	Internal Clock Divided By	External Clock Divided By
00	1	8	1
01	2	16	2
10	4	32	4
11	8	64	8

EMU — Emulation Control

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

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

T2CG — TCR2 Clock/Gate Control

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

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

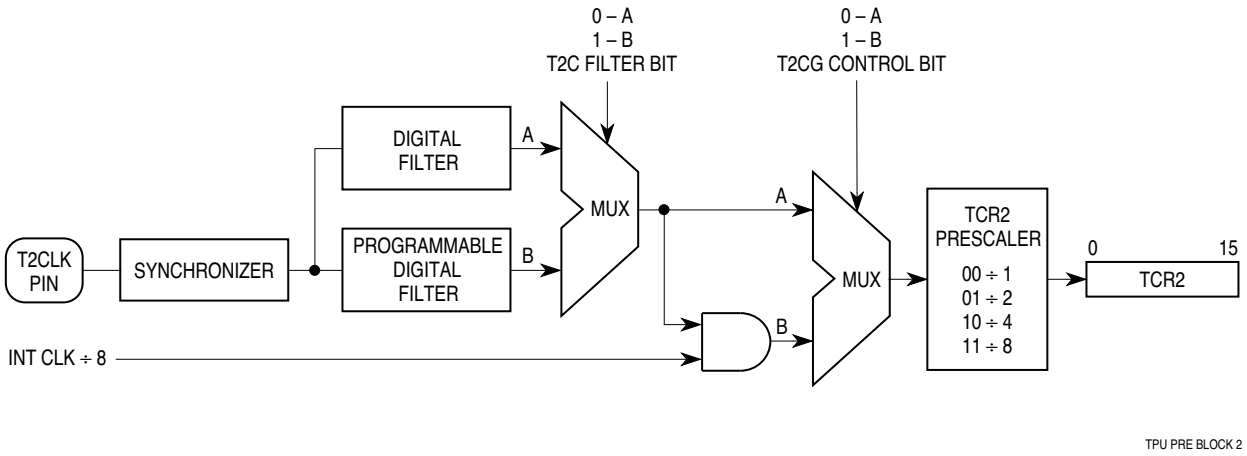


Figure 13 Prescaler Control 2

STF — Stop Flag
0 = TPU operating
1 = TPU stopped (STOP bit has been asserted)

SUPV — Supervisor Data Space
0 = Assignable registers are unrestricted (FC2 is ignored)
1 = Assignable registers are restricted (FC2 is decoded)

PSCK — Prescaler Clock
0 = System clock/32 is input to TCR1 prescaler
1 = System clock/4 is input to TCR1 prescaler

IARB — Interrupt Arbitration Identification Number
The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, nonzero IARB field value. Refer to **3.8 Interrupts** for more information.

TICR — TPU Interrupt Configuration Register **$YFFE08**

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

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

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

Table 27 ADC Module Address Map

Access	Address	15 8	7 0
S	$YFF700	ADC MODULE CONFIGURATION REGISTER (ADCMCR)	
S	$YFF702	ADC FACTORY TEST REGISTER (ADTEST)	
S	$YFF704	(RESERVED)	
S/U	$YFF706	PORT ADA DATA (PORTADA)	
S/U	$YFF708	(RESERVED)	
S/U	$YFF70A	ADC CONTROL 0 (ADCTL0)	
S/U	$YFF70C	ADC CONTROL 1 (ADCTL1)	
S/U	$YFF70E	ADC STATUS (ADSTAT)	
S/U	$YFF710	RIGHT-JUSTIFIED UNSIGNED RESULT 0 (RJURR0)	
S/U	$YFF712	RIGHT-JUSTIFIED UNSIGNED RESULT 1 (RJURR1)	
S/U	$YFF714	RIGHT-JUSTIFIED UNSIGNED RESULT 2 (RJURR2)	
S/U	$YFF716	RIGHT-JUSTIFIED UNSIGNED RESULT 3 (RJURR3)	
S/U	$YFF718	RIGHT-JUSTIFIED UNSIGNED RESULT 4 (RJURR4)	
S/U	$YFF71A	RIGHT-JUSTIFIED UNSIGNED RESULT 5 (RJURR5)	
S/U	$YFF71C	RIGHT-JUSTIFIED UNSIGNED RESULT 6 (RJURR6)	
S/U	$YFF71E	RIGHT-JUSTIFIED UNSIGNED RESULT 7 (RJURR7)	
S/U	$YFF720	LEFT-JUSTIFIED SIGNED RESULT 0 (LJSRR0)	
S/U	$YFF722	LEFT-JUSTIFIED SIGNED RESULT 1 (LJSRR1)	
S/U	$YFF724	LEFT-JUSTIFIED SIGNED RESULT 2 (LJSRR2)	
S/U	$YFF726	LEFT-JUSTIFIED SIGNED RESULT 3 (LJSRR3)	
S/U	$YFF728	LEFT-JUSTIFIED SIGNED RESULT 4 (LJSRR4)	
S/U	$YFF72A	LEFT-JUSTIFIED SIGNED RESULT 5 (LJSRR5)	
S/U	$YFF72C	LEFT-JUSTIFIED SIGNED RESULT 6 (LJSRR6)	
S/U	$YFF72E	LEFT-JUSTIFIED SIGNED RESULT 7 (LJSRR7)	
S/U	$YFF730	LEFT-JUSTIFIED UNSIGNED RESULT 0 (LJURR0)	
S/U	$YFF732	LEFT-JUSTIFIED UNSIGNED RESULT 1 (LJURR1)	
S/U	$YFF734	LEFT-JUSTIFIED UNSIGNED RESULT 2 (LJURR2)	
S/U	$YFF736	LEFT-JUSTIFIED UNSIGNED RESULT 3 (LJURR3)	
S/U	$YFF738	LEFT-JUSTIFIED UNSIGNED RESULT 4 (LJURR4)	
S/U	$YFF73A	LEFT-JUSTIFIED UNSIGNED RESULT 5 (LJURR5)	
S/U	$YFF73C	LEFT-JUSTIFIED UNSIGNED RESULT 6 (LJURR6)	
S/U	$YFF73E	LEFT-JUSTIFIED UNSIGNED RESULT 7 (LJURR7)	

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

6.4 ADC Registers

ADCMCR — ADC Module Configuration Register **$YFF700**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ		NOT USED					SUPV	NOT USED						
RESET:															
1	0	0						0							

The ADCMCR is used to initialize the ADC.

STOP — STOP Mode

0 = Normal operation

1 = Low-power operation

STOP places the ADC in low-power state by disabling the ADC clock and powering down the analog circuitry. Setting STOP aborts any conversion in progress. STOP is set to logic level one at reset, and may be cleared to logic level zero by the CPU.

Clearing STOP enables normal ADC operation. However, because analog circuitry bias current has been turned off, there is a period of recovery before output stabilization.

FRZ[1:0] — Freeze 1

The FRZ field is used to determine ADC response to assertion of the FREEZE signal. The following table shows possible responses.

FRZ	Response
00	Ignore FREEZE
01	Reserved
10	Finish conversion, then freeze
11	Freeze immediately

SUPV — Supervisor/Unrestricted

0 = Unrestricted access

1 = Supervisor access

SUPV defines access to assignable ADC registers.

ADTEST — ADC Test Register **$YFF702**

ADTEST is used during factory test of the ADC.

PORTADA — Port ADA Data Register **$YFF706**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PORT ADA DATA							
RESET:															
								Reflects state of the input pins							

Port ADA is an input port that shares pins with the A/D converter inputs. The port ADA data register resides in the lower half of the PDR. Bit 7 always returns zero when read.

PDR[6:0] — Port ADA Data

A read of PDR[6:0] will return the logic level of the port A pins. If the input is outside the defined voltage range, the result of the read will be indeterminate. Use of a port A pin for digital input does not preclude use as an analog input.

The following table summarizes the operation of S8CM and CD:CA when MULT is set (multichannel mode). Number of conversions per channel is determined by SCAN. Channel numbers are given in order of conversion.

S8CM	CD	CC	CB	CA	Input	Result Register
0	0	0	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
0	0	1	X	X	AN4	RSLT0
					AN5	RSLT1
					AN6	RSLT2
					AN7*	RSLT3
0	1	0	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
0	1	1	X	X	V_{RH}	RSLT0
					V_{RL}	RSLT1
					$(V_{RH} - V_{RL}) / 2$	RSLT2
					Test/Reserved	RSLT3
1	0	X	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
					AN4	RSLT4
					AN5	RSLT5
					AN6	RSLT6
					AN7*	RSLT7
1	1	X	X	X	Reserved	RSLT0
					Reserved	RSLT1
					Reserved	RSLT2
					Reserved	RSLT3
					V_{RH}	RSLT4
					V_{RL}	RSLT5
					$(V_{RH} - V_{RL}) / 2$	RSLT6
					Test/Reserved	RSLT7

* Since the ADC in the MCU has only seven external analog inputs, AN7 is connected to V_{SSA}.

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.

STOP — Stop Control

0 = TPURAM array operates normally.

1 = TPURAM array enters low-power stop mode.

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

RASP — RAM Array Space Field

0 = TPURAM array is placed in unrestricted space

1 = TPURAM array is placed in supervisor space

TRAMTST — TPURAM Test Register **$YFFB02**

TRAMTST is used for factory testing of the TPURAM module.

TRAMBAR — TPURAM Base Address and Status Register **$YFFB04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	NOT USED		RAMDS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0			0

ADDR[23:11] — RAM Array Base Address

This field specifies ADDR[23:11] of the TPURAM array base address when the array is enabled. ADDR11 determines the 2-Kbyte boundary that the block of addresses containing the array is mapped to. The array occupies only the upper 1024 bytes of the 2-Kbyte block of addresses. When ADDR11 bit is set, valid array addresses range from $0C00 to%0FFFF; when the ADDR11 bit is cleared, valid array addresses range from $0400 to $07FF.

RAMDS — RAM Array Disable

0 = TPURAM array is enabled

1 = TPURAM array is disabled

The TPURAM array is disabled by internal logic after a master reset. Writing a valid base address to the TPURAM array base address field (bits [15:3]) automatically clears RAMDS, enabling the TPURAM array.

7.4 TPURAM Operation

There are six TPURAM operating modes, as follows:

- The TPURAM module is in normal mode when powered by V_{DD}. The array can be accessed by byte, word, or long word. A byte or aligned word (high-order byte is at an even address) access only takes one bus cycle or two system clocks. A long word or misaligned word access requires two bus cycles.
- Standby mode is intended to preserve TPURAM contents when V_{DD} is removed. TPURAM contents are maintained by V_{STBY}. Circuitry within the TPURAM module switches to the higher of V_{DD} or V_{STBY} with no loss of data. When TPURAM is powered by V_{STBY}, access to the array is not guaranteed.
- Reset mode allows the CPU to complete the current bus cycle before resetting. When a synchronous reset occurs while a byte or word TPURAM access is in progress, the access will be completed. If reset occurs during the first word access of a long-word operation, only the first word access will be completed. If reset occurs during the second word access of a long word operation, the entire access will be completed. Data being read from or written to the RAM may be corrupted by asynchronous reset.