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	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	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/product-detail/nxp-semiconductors/mc68334gfc16

2 Signal Descriptions

2.1 Pin Characteristics

The following table shows MCU pins and their characteristics. All inputs detect CMOS logic levels. All inputs can be put in a high-impedance state, but the method of doing this differs depending upon pin function. Refer to the table MCU Driver Types for a description of output drivers. An entry in the discrete I/O column of the MCU Pin Characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to the MCU Block Diagram for information about port organization.

Table 2 MCU Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/$\overline{CS10}$/ECLK	A	Y	N	—	—
ADDR[22:19]/$\overline{CS[9:6]}$	A	Y	N	O	PC[6:3]
ADDR[18:0]	A	Y	N	—	—
AN[6:0]	—	Y[1]	Y	I	PADA[6:0]
$\overline{AS}$	B	Y	N	I/O	PE5
$\overline{AVEC}$	B	Y	N	I/O	PE2
$\overline{BERR}$[2]	B	Y	N	—	—
$\overline{BG}$/$\overline{CS1}$	B	—	—	—	—
$\overline{BGACK}$/$\overline{CS2}$	B	Y	N	—	—
$\overline{BKPT}$/DSCLK	—	Y	Y	—	—
$\overline{BR}$/$\overline{CS0}$	B	Y	N	—	—
CLKOUT	A	—	—	—	—
$\overline{CSBOOT}$	B	—	—	—	—
DATA[15:0][1]	Aw	Y	N	—	—
$\overline{DS}$	B	Y	N	I/O	PE4
$\overline{DSACK1}$	B	Y	N	I/O	PE1
$\overline{DSACK0}$	B	Y	N	I/O	PE0
DSI/$\overline{IFETCH}$	A	Y	Y	—	—
DSO/$\overline{IPIPE}$	A	—	—	—	—
EXTAL[3]	—	—	—	—	—
FC[2:0]/$\overline{CS[5:3]}$	A	Y	—	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
$\overline{HALT}$[2]	Bo	Y	N	—	—
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	PF[7:1]
MODCLK[1]	B	Y	N	I/O	PF0
R/$\overline{W}$	A	Y	N	—	—
$\overline{RESET}$	Bo	Y	Y	—	—
$\overline{RMC}$	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]	—	—	—	—	—
R/$\overline{W}$	A	Y	N	—	—

2.5 Signal Function

Table 6 MCU Signal Function

Mnemonic	Signal Name	Function
ADDR[23:0]	Address Bus	24-bit address bus used by CPU32
AN[6:0]	ADC Analog Input	Inputs to ADC multiplexer
$\overline{AS}$	Address Strobe	Indicates that a valid address is on the address bus
$\overline{AVEC}$	Autovector	Requests an automatic vector during interrupt acknowledge
$\overline{BERR}$	Bus Error	Indicates that a bus error has occurred
$\overline{BG}$	Bus Grant	Indicates that the MCU has relinquished the bus
$\overline{BGACK}$	Bus Grant Acknowledge	Indicates that an external device has assumed bus mastership
$\overline{BKPT}$	Breakpoint	Signals a hardware breakpoint to the CPU
$\overline{BR}$	Bus Request	Indicates that an external device requires bus mastership
CLKOUT	System Clockout	System clock output
$\overline{CS}$[10:0]	Chip Selects	Select external devices at programmed addresses
$\overline{CSBOOT}$	Boot Chip Select	Chip select for external boot start-up ROM
DATA[15:0]	Data Bus	16-bit data bus
$\overline{DS}$	Data Strobe	During a read cycle, indicates when it is possible for an external device to place data on the data bus. During a write cycle, indicates that valid data is on the data bus.
$\overline{DSACK}$[1:0]	Data and Size Acknowledge	Provide asynchronous data transfers and dynamic bus sizing
DSI, DSO, DSCLK	Development Serial In, Out, Clock	Serial I/O and clock for background debugging mode
ECLK	E-Clock	External M6800 bus clock output
EXTAL, XTAL	Crystal Oscillator	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
FC[2:0]	Function Codes	Identify processor state and current address space
FREEZE	Freeze	Indicates that the CPU has entered background mode
$\overline{HALT}$	Halt	Suspend external bus activity
$\overline{IFETCH}$	Instruction Fetch	Identifies bus cycles in which operand is loaded into pipeline
$\overline{IPIPE}$	Instruction Pipeline	Indicates instruction pipeline activity
$\overline{IRQ}$[7:1]	Interrupt Request Level	Provide prioritized interrupts to the CPU
MODCLK	Clock Mode Select	Selects the source and type of system clock
PADA[6:0]	Port ADA	ADC digital input port signals
PC[6:0]	Port C	SIM digital output port signals
PE[7:0]	Port E	SIM digital I/O port signals
PF[7:0]	Port F	SIM digital I/O port signals
QUOT	Quotient Out	Provides the quotient bit of the polynomial divider
$\overline{RESET}$	Reset	System reset
$\overline{RMC}$	Read-Modify-Write Cycle	Indicates an indivisible read-modify-write instruction
R/$\overline{W}$	Read/Write	Indicates the direction of data transfer on the bus
SIZ[1:0]	Size	Indicates the number of bytes to be transferred during a bus cycle
TPUCH[15:0]	TPU I/O Channels	Bidirectional TPU channels
TSC	Three-State Control	Places all output drivers in a high-impedance state
T2CLK	TCR2 Clock	TPU clock input
$V_{RH, VRL}$	ADC Reference Voltage	Provide precise reference for A/D conversion
XFC	External Filter Capacitor	Connection for external phase-locked loop filter capacitor

3.2.7 Periodic Interrupt Timer

The periodic interrupt timer (PIT) generates interrupts of specified priorities at specified intervals. Timing for the PIT is provided by a programmable prescaler driven by the system clock.

PICR — Periodic Interrupt Control Register **$YFFA22**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	PIRQL			PIV							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

This register contains information concerning periodic interrupt priority and vectoring. Bits [10:0] can be read or written at any time. Bits [15:11] are unimplemented and always return zero.

PIRQL[2:0] — Periodic Interrupt Request Level

The following table shows what interrupt request level is asserted when a periodic interrupt is generated. If a PIT interrupt and an external $\overline{\text{IRQ}}$ signal of the same priority occur simultaneously, the PIT interrupt is serviced first. The periodic timer continues to run when the interrupt is disabled.

PIRQL	Interrupt Request Level
000	Periodic Interrupt Disabled
001	Interrupt Request Level 1
010	Interrupt Request Level 2
011	Interrupt Request Level 3
100	Interrupt Request Level 4
101	Interrupt Request Level 5
110	Interrupt Request Level 6
111	Interrupt Request Level 7

PIV[7:0] — Periodic Interrupt Vector

The bits of this field contain the vector generated in response to an interrupt from the periodic timer. When the SIM responds, the periodic interrupt vector is placed on the bus.

PITR — Periodic Interrupt Timer Register **$YFFA24**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PTP	PITM							
RESET:															
0	0	0	0	0	0	0	$\overline{\text{MODCLK}}$	0	0	0	0	0	0	0	0

The PITR contains the count value for the periodic timer. A zero value turns off the periodic timer. This register can be read or written at any time.

PTP — Periodic Timer Prescaler Control

0 = Periodic timer clock not prescaled

1 = Periodic timer clock prescaled by a value of 512

The reset state of PTP is the complement of the state of the MODCLK signal during reset.

PITM[7:0] — Periodic Interrupt Timing Modulus Field

This is an 8-bit timing modulus. The period of the timer can be calculated as follows:

$$\text{PIT Period} = \frac{(\text{PITM})(\text{Prescale})(4)}{\text{EXTAL Frequency}}$$

where

PIT Period = Periodic interrupt timer period
PITM = Periodic interrupt timer register modulus (PITR[7:0])
EXTAL Frequency = Crystal frequency
Prescale = 512 or 1 depending on the state of the PTP bit in the PITR

3.3 System Clock

The system clock in the SIM provides timing signals for the IMB modules and for an external peripheral bus. Because the MCU is a fully static design, register and memory contents are not affected when the clock rate changes. System hardware and software support changes in clock rate during operation.

The system clock signal can be generated in one of three ways. An internal phase-locked loop can synthesize the clock from either an internal reference or an external reference, or the clock signal can be input from an external frequency source. Keep these clock sources in mind while reading the rest of this section. The figure below is a block diagram of the system clock.

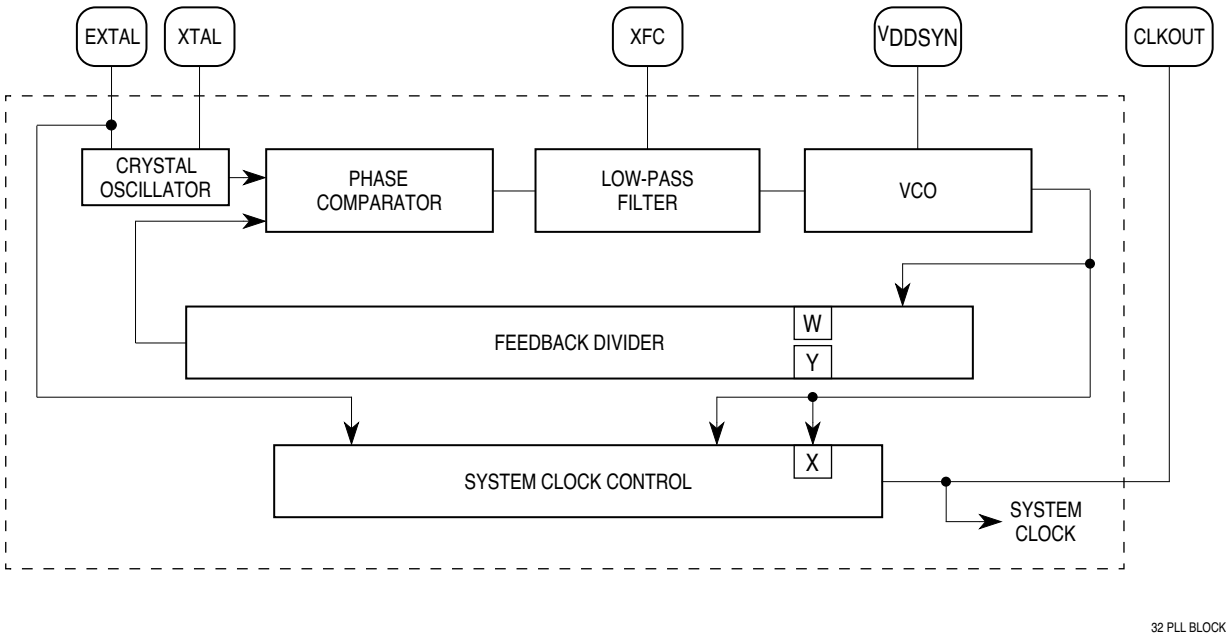
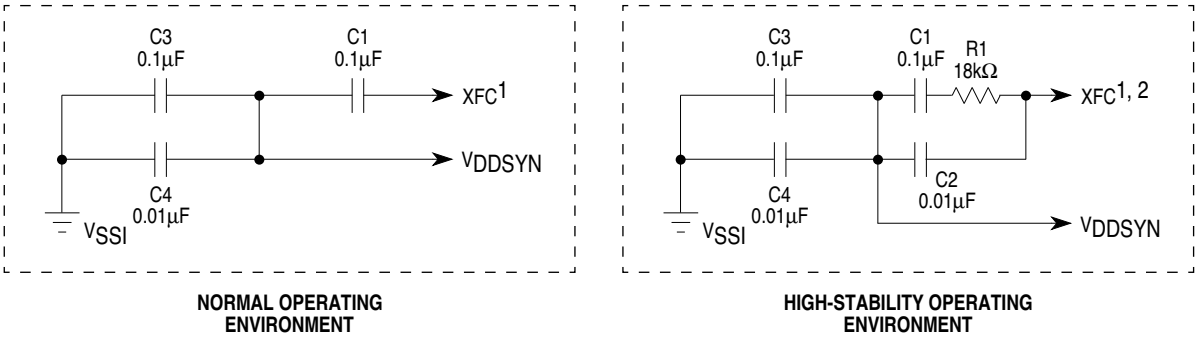


Figure 6 System Clock Block Diagram

3.3.1 Clock Sources

The state of the clock mode (MODCLK) pin during reset determines clock source. When MODCLK is held high during reset, the clock synthesizer generates a clock signal from either an internal or an external reference frequency — the clock synthesizer control register (SYNCR) determines operating frequency and mode of operation. When MODCLK is held low during reset, the clock synthesizer is disabled and an external system clock signal must be applied — SYNCR control bits have no effect.

To generate a reference frequency using the internal oscillator a reference crystal must be connected between the EXTAL and XTAL pins. The figure below shows a recommended circuit.



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.

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.

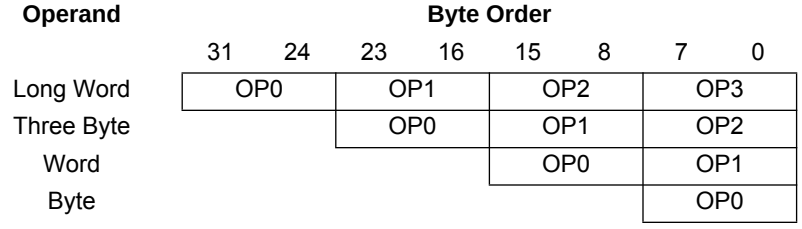


Figure 9 Operand Byte Order

3.4.10 Operand Alignment

The data multiplexer establishes the necessary connections for different combinations of address and data sizes. The multiplexer takes the two bytes of the 16-bit bus and routes them to their required positions. Positioning of bytes is determined by the size and address outputs. SIZ1 and SIZ0 indicate the remaining number of bytes to be transferred during the current bus cycle. The number of bytes transferred is equal to or less than the size indicated by SIZ1 and SIZ0, depending on port width.

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

3.4.11 Misaligned Operands

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

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

3.4.12 Operand Transfer Cases

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

Table 12 Operand Alignment

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

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

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 DSACK

SPACE — Address Space

Use this option field to select an address space for the chip-select logic. The CPU32 normally operates in supervisor or user space, but interrupt acknowledge cycles must take place in CPU space.

Space	Address Space
00	CPU Space
01	User Space
10	Supervisor Space
11	Supervisor/User Space

IPL — Interrupt Priority Level

If the space field is set for CPU space (00), chip-select logic can be used for interrupt acknowledge. During an interrupt acknowledge cycle, the priority level on address lines ADDR[3:1] is compared to the value in the IPL field. If the values are the same, a chip select is asserted, provided that other option register conditions are met. The following table shows IPL field encoding.

IPL	Description
000	Any Level
001	IPL1
010	IPL2
011	IPL3
100	IPL4
101	IPL5
110	IPL6
111	IPL7

This field only affects the response of chip selects and does not affect interrupt recognition by the CPU. Any level means that chip select is asserted regardless of the level of the interrupt acknowledge cycle.

AVEC — Autovector Enable

0 = External interrupt vector enabled

1 = Autovector enabled

This field selects one of two methods of acquiring an interrupt vector number during an external interrupt acknowledge cycle.

If the chip select is configured to trigger on an interrupt acknowledge cycle (SPACE = 00) and the $\overline{\text{AVEC}}$ field is set to one, the chip select circuit generates an internal $\overline{\text{AVEC}}$ signal in response to an external interrupt cycle, and the SIM supplies an automatic vector number. Otherwise, the vector number must be supplied by the requesting device. An internal autovector is generated only in response to interrupt requests from the SIM $\overline{\text{IRQ}}$ pins — interrupt requests from other IMB modules are ignored.

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

3.5.5 Port C Data Register

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

PORTC — Port C Data Register **$YFFA41**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								0	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:								0	1	1	1	1	1	1	1

3.6 General-Purpose Input/Output

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

PORTE0, PORTE1 — Port E Data Register **$YFFA11, $YFFA13**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:								U	U	U	U	U	U	U	U

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

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

DDRE — Port E Data Direction Register **$YFFA15**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
RESET:								0	0	0	0	0	0	0	0

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

PEPAR — Port E Pin Assignment Register **$YFFA17**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PEPA7	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0
RESET:								DATA8	DATA8	DATA8	DATA8	DATA8	DATA8	DATA8	DATA8

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

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

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

Table 19 Port E Pin Assignments

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

PORTF0, PORTF1 — Port F Data Register **$YFFA19, $YFFA1B**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:								U	U	U	U	U	U	U	U

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

The port F data register is a single register that can be accessed in two locations. When accessed at $YFFA19, the register is referred to as PORTF0; when accessed at $YFFA1B, the register is referred to as PORTF1. The register can be read or written at any time. It is unaffected by reset.

DDRF — Port F Data Direction Register **$YFFA1D**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:								0	0	0	0	0	0	0	0

The bits in this register control the direction of the pin drivers when the pins are configured for I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input.

4.6 Instruction Set Summary

Table 23 Instruction Set Summary

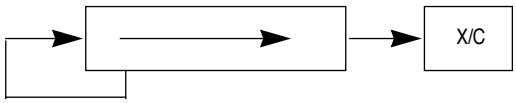
Instruction	Syntax	Operand Size	Operation
ABCD	Dn, Dn – (An), – (An)	8 8	$Source_{10}$ + $Destination_{10}$ + X ⇒ Destination
ADD	Dn, ⟨ea⟩ ⟨ea⟩, Dn	8, 16, 32 8, 16, 32	Source + Destination ⇒ Destination
ADDA	⟨ea⟩, An	16, 32	Source + Destination ⇒ Destination
ADDI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate data + Destination ⇒ Destination
ADDQ	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate data + Destination ⇒ Destination
ADDX	Dn, Dn – (An), – (An)	8, 16, 32 8, 16, 32	Source + Destination + X ⇒ Destination
AND	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source ∗ Destination ⇒ Destination
ANDI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Data ∗ Destination ⇒ Destination
ANDI to CCR	#⟨data⟩, CCR	8	Source ∗ CCR ⇒ CCR
ANDI to SR[1]	#⟨data⟩, SR	16	Source ∗ SR ⇒ SR
ASL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ASR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
Bcc	⟨label⟩	8, 16, 32	If condition true, then PC + d ⇒ PC
BCHG	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	(⟨bit number⟩ of destination) ⇒ Z ⇒ bit of destination
BCLR	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	(⟨bit number⟩ of destination) ⇒ Z; 0 ⇒ bit of destination
BGND	none	none	If background mode enabled, then enter background mode, else format/vector offset ⇒ – (SSP); PC ⇒ – (SSP); SR ⇒ – (SSP); (vector) ⇒ PC
BKPT	#⟨data⟩	none	If breakpoint cycle acknowledged, then execute returned operation word, else trap as illegal instruction.
BRA	⟨label⟩	8, 16, 32	PC + d ⇒ PC
BSET	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	(⟨bit number⟩ of destination) ⇒ Z; 1 ⇒ bit of destination
BSR	⟨label⟩	8, 16, 32	SP – 4 ⇒ SP; PC ⇒ (SP); PC + d ⇒ PC
BTST	Dn, ⟨ea⟩ #⟨data⟩, ⟨ea⟩	8, 32 8, 32	(⟨bit number⟩ of destination) ⇒ Z
CHK	⟨ea⟩, Dn	16, 32	If Dn < 0 or Dn < (ea), then CHK exception
CHK2	⟨ea⟩, Rn	8, 16, 32	If Rn < lower bound or Rn > upper bound, then CHK exception
CLR	⟨ea⟩	8, 16, 32	0 ⇒ Destination
CMP	⟨ea⟩, Dn	8, 16, 32	(Destination – Source), CCR shows results
CMPA	⟨ea⟩, An	16, 32	(Destination – Source), CCR shows results
CMPI	#⟨data⟩, ⟨ea⟩	8, 16, 32	(Destination – Data), CCR shows results
CMPM	(An) +, (An) +	8, 16, 32	(Destination – Source), CCR shows results
CMP2	⟨ea⟩, Rn	8, 16, 32	Lower bound ≤ Rn ≤ Upper bound, CCR shows result

Table 23 Instruction Set Summary

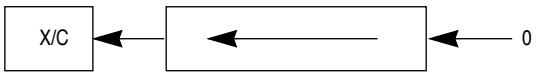
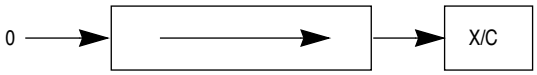
Instruction	Syntax	Operand Size	Operation
DBcc	Dn, ⟨label⟩	16	If condition false, then Dn – 1 ⇒ PC; if Dn ≠ (– 1), then PC + d ⇒ PC
DIVS/DIVU	⟨ea⟩, Dn	32/16 ⇒ 16: 16	Destination / Source ⇒ Destination (signed or unsigned)
DIVSL/DIVUL	⟨ea⟩, Dr: Dq ⟨ea⟩, Dq ⟨ea⟩, Dr : Dq	64/32 ⇒ 32 : 32 32/32 ⇒ 32 32/32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

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.

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.

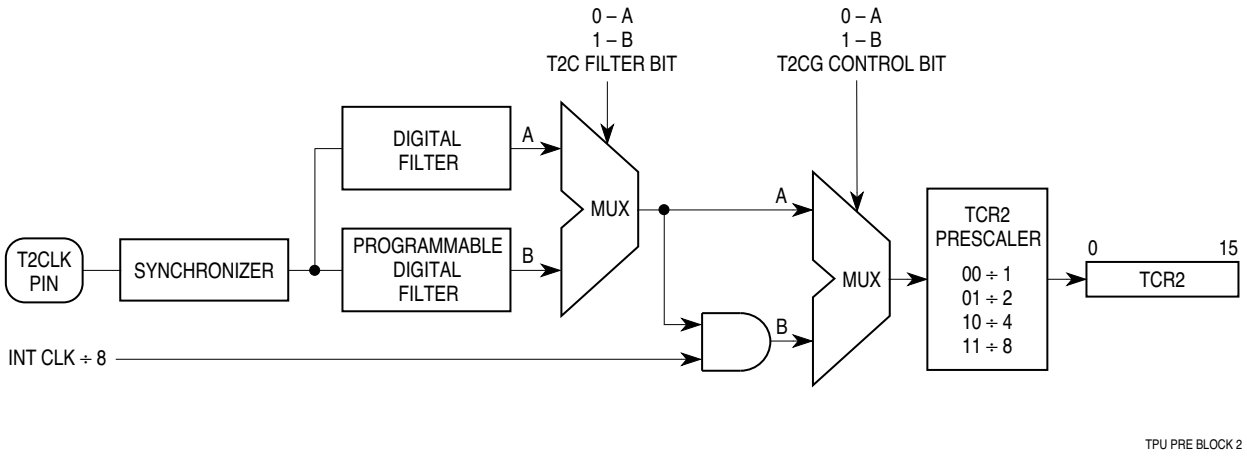


Figure 13 Prescaler Control 2

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

ADCTL0 — A/D Control Register 0 **$YFF70A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								RES10	STS		PRS				
RESET:								0	0	0	0	0	0	1	1

ADCTL0 is used to select ADC clock source and to set up prescaling. Writes to it have immediate effect.

RES10 — 10-Bit Resolution

0 = 8-bit resolution

1 = 10-bit resolution

Conversion results are appropriately aligned in result registers to reflect conversion status.

STS[1:0] — Sample Time Select

Total conversion time depends on initial sample time, transfer time, final sample time, and resolution time. Initial sample time is fixed at two clocks. Transfer time is fixed at two clocks. Resolution time is fixed at ten ADC clock cycles for an 8-bit conversion and twelve ADC clock cycles for a 10-bit conversion. Final sample time depends on the STS field, as shown below.

STS[1:0]	Sample Time
00	2 A/D Clock Periods
01	4 A/D Clock Periods
10	8 A/D Clock Periods
11	16 A/D Clock Periods

PRS[4:0] — Prescaler Rate Selection Field

ADC clock is generated from system clock using a modulo counter and a divide-by-two circuit. The binary value of this field is the counter modulus. System clock is divided by the PRS value plus one, then sent to the divide-by-two circuit, as shown in the following table. Maximum ADC clock rate is 2 MHz. Reset value of PRS is a divisor value of eight, resulting in a nominal 2-MHz ADC clock.

PRS[4:0]	Divisor Value
00000	4
00001	4
00010	6
...	...
11101	60
11110	62
11111	64

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor