



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, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68332ameh16

1.1 Features

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

1.2 Block Diagram

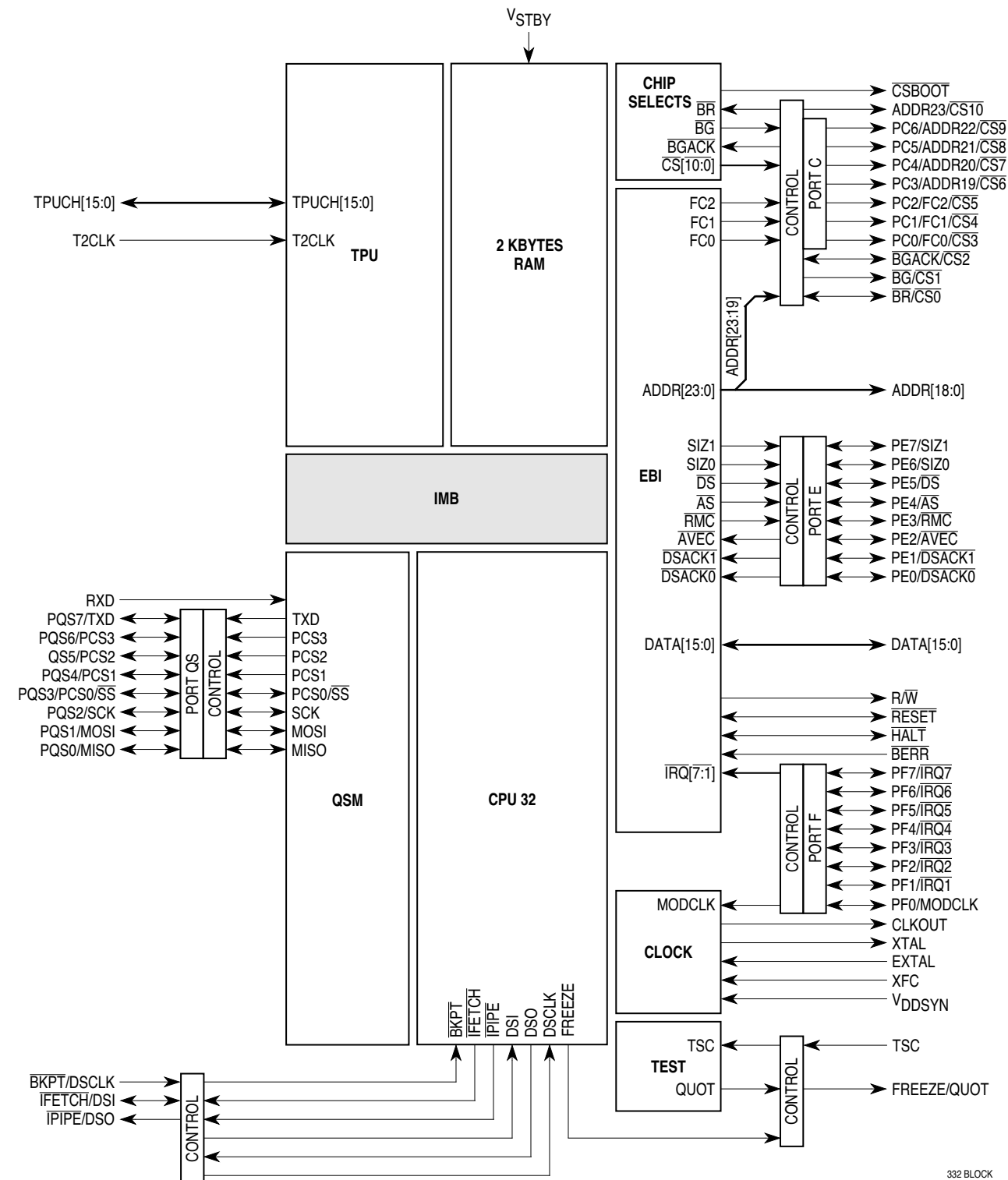


Figure 1 MCU Block Diagram

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 Characteristic

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/CS10/ECLK	A	Y	N	O	—
ADDR[22:19]/CS[9:6]	A	Y	N	O	PC[6:3]
ADDR[18:0]	A	Y	N	—	—
\overline{AS}	B	Y	N	I/O	PE5
\overline{AVEC}	B	Y	N	I/O	PE2
\overline{BERR}	B	Y	N	—	—
$\overline{BG/CS1}$	B	—	—	—	—
$\overline{BGACK/CS2}$	B	Y	N	—	—
$\overline{BKPT/DSCLK}$	—	Y	Y	—	—
$\overline{BR/CS0}$	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/IFETCH	A	Y	Y	—	—
DSO/IPIPE	A	—	—	—	—
EXTAL ²	—	—	Special	—	—
FC[2:0]/CS[5:3]	A	Y	N	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
\overline{HALT}	Bo	Y	N	—	—
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	PF[7:1]
MISO	Bo	Y	Y	I/O	PQS0
MODCLK ¹	B	Y	N	I/O	PF0
MOSI	Bo	Y	Y	I/O	PQS1
PCS0/ \overline{SS}	Bo	Y	Y	I/O	PQS3
PCS[3:1]	Bo	Y	Y	I/O	PQS[6:4]
R/W	A	Y	N	—	—
\overline{RESET}	Bo	Y	Y	—	—
\overline{RMC}	B	Y	N	I/O	PE3
RXD	—	N	N	—	—
SCK	Bo	Y	Y	I/O	PQS2
SIZ[1:0]	B	Y	N	I/O	PE[7:6]

3.2.2 System Protection Control Register

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

SYPCR — System Protection Control Register

\$YFFA21

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

SWE — Software Watchdog Enable

0 = Software watchdog disabled

1 = Software watchdog enabled

SWP — Software Watchdog Prescale

This bit controls the value of the software watchdog prescaler.

0 = Software watchdog clock not prescaled

1 = Software watchdog clock prescaled by 512

SWT[1:0] — Software Watchdog Timing

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

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

HME — Halt Monitor Enable

0 = Disable halt monitor function

1 = Enable halt monitor function

BME — Bus Monitor External Enable

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

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

BMT[1:0] — Bus Monitor Timing

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

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

3.2.3 Bus Monitor

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

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

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

3.2.4 Halt Monitor

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

3.2.5 Spurious Interrupt Monitor

The spurious interrupt monitor issues \overline{BERR} if no interrupt arbitration occurs during an interrupt-acknowledge cycle.

3.2.6 Software Watchdog

The software watchdog is controlled by SWE in the SYPCR. Once enabled, the watchdog requires that a service sequence be written to SWSR on a periodic basis. If servicing does not take place, the watchdog times out and issues a reset. This register can be written at any time, but returns zeros when read.

SWSR —Software Service Register

\$YFFA27

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

Register shown with read value

Perform a software watchdog service sequence as follows:

- Write \$55 to SWSR.
- Write \$AA to SWSR.

Both writes must occur before time-out in the order listed, but any number of instructions can be executed between the two writes.

The watchdog clock rate is affected by SWP and SWT in SYPCR. When SWT[1:0] are modified, a watchdog service sequence must be performed before the new time-out period takes effect.

The reset value of SWP is affected by the state of the MODCLK pin on the rising edge of reset, as shown in the following table.

MODCLK	SWP
0	1
1	0

When an external system clock signal is applied (i.e., the PLL is not used), duty cycle of the input is critical, especially at near maximum operating frequencies. The relationship between clock signal duty cycle and clock signal period is expressed:

$$\text{Minimum external clock period} = \frac{\text{minimum external clock high/low time}}{50\% - \text{percentage variation of external clock input duty cycle}}$$

3.3.2 Clock Synthesizer Operation

A voltage controlled oscillator (VCO) generates the system clock signal. A portion of the clock signal is fed back to a divider/counter. The divider controls the frequency of one input to a phase comparator. The other phase comparator input is a reference signal, either from the internal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between its two inputs. The signal is low-pass filtered and used to correct VCO output frequency.

The synthesizer locks when VCO frequency is identical to reference 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 re-lock. Lock status is shown by the SLOCK bit in SYNCR.

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.

The low-pass filter requires an external low-leakage capacitor, typically 0.1 μF , connected between the XFC and V_{DDSYN} pins.

V_{DDSYN} is used to power the clock circuits. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. Use a quiet power supply as the V_{DDSYN} source, since PLL stability depends on the VCO, which uses this supply. Place adequate external bypass capacitors as close as possible to the V_{DDSYN} pin to ensure stable operating frequency.

When the clock synthesizer is used, control register SYNCR determines operating frequency and various modes of operation. SYNCR can be read only when the processor is operating at the supervisor privilege level.

The SYNCR X bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting X doubles clock speed without changing VCO speed. There is no VCO relock delay. The SYNCR W bit controls a 3-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 either W or Y value changes, there is a VCO relock delay.

Clock frequency is determined by SYNCR bit settings as follows:

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

In order for the device to perform correctly, the clock frequency selected by the W, X, and Y bits must be within the limits specified for the MCU.

The VCO frequency is twice the system clock frequency if $X = 1$ or four times the system clock frequency if $X = 0$.

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

3.3.3 Clock Control

The clock control circuits determine system clock frequency and clock operation under special circumstances, such as following loss of synthesizer reference or during low-power operation. Clock source is determined by the logic state of the MODCLK pin during reset.

SYNCR — Clock Synthesizer Control Register

\$YFFA04

15	14	13		8	7	6	5	4	3	2	1	0
W	X	Y			EDIV	0	0	SLIMP	SLOCK	RSTEN	STSIM	STEXT

RESET:

0 0 1 1 1 1 1 1 0 0 0 U U 0 0 0

When the on-chip clock synthesizer is used, system clock frequency is controlled by the bits in the upper byte of SYNCR. Bits in the lower byte show status of or control operation of internal and external clocks. The SYNCR can be read or written only when the CPU is operating at the supervisor privilege level.

W — Frequency Control (VCO)

This bit controls a prescaler tap in the synthesizer feedback loop. Setting the bit increases the VCO speed by a factor of four. VCO relock delay is required.

X — Frequency Control Bit (Prescale)

This bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting the bit doubles clock speed without changing the VCO speed. There is no VCO relock delay.

Y[5:0] — Frequency Control (Counter)

The Y field controls the modulus down counter in the synthesizer feedback loop, causing it to divide by a value of Y + 1. Values range from 0 to 63. VCO relock delay is required.

EDIV — E Clock Divide Rate

0 = ECLK frequency is system clock divided by 8.

1 = ECLK frequency is system clock divided by 16.

ECLK is an external M6800 bus clock available on pin ADDR23. Refer to **3.5 Chip Selects** for more information.

SLIMP — Limp Mode Flag

0 = External crystal is VCO reference.

1 = Loss of crystal reference.

When the on-chip synthesizer is used, loss of reference frequency causes SLIMP to be set. The VCO continues to run using the base control voltage. Maximum limp frequency is maximum specified system clock frequency. X-bit state affects limp frequency.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.

1 = VCO has locked on the desired frequency (or system clock is external).

The MCU maintains reset state until the synthesizer locks, but SLOCK does not indicate synthesizer lock status until after the user writes to SYNCR.

RSTEN — Reset Enable

0 = Loss of crystal causes the MCU to operate in limp mode.

1 = Loss of crystal causes system reset.

STSIM — Stop Mode SIM Clock

0 = When LPSTOP is executed, the SIM clock is driven from the crystal oscillator and the VCO is turned off to conserve power.

1 = When LPSTOP is executed, the SIM clock is driven from the VCO.

STEXT — Stop Mode External Clock

0 = When LPSTOP is executed, the CLKOUT signal is held negated to conserve power.

1 = When LPSTOP is executed, the CLKOUT signal is driven from the SIM clock, as determined by the state of the STSIM bit.

3.5.3 Base Address Registers

A base address is the starting address for the block enabled by a given chip select. Block size determines the extent of the block above the base address. Each chip select has an associated base register so that an efficient address map can be constructed for each application. If a chip-select base address register is programmed with the same address as a microcontroller module or memory array, an access to that address goes to the module or array and the chip-select signal is not asserted.

CSBARBT — Chip-Select Base Address Register Boot ROM

\$YFFA48

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

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.

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 K	ADDR[23:11]
001	8 K	ADDR[23:13]
010	16 K	ADDR[23:14]
011	64 K	ADDR[23:16]
100	128 K	ADDR[23:17]
101	256 K	ADDR[23:18]
110	512 K	ADDR[23:19]
111	1 M	ADDR[23:20]

3.5.4 Option Registers

The option registers contain eight fields that determine timing of and conditions for assertion of chip-select signals. For a chip-select signal to be asserted, all bits in the base address register must match the corresponding internal upper address lines, and all conditions specified in the option register must be satisfied. These conditions also apply to providing DSACK or autovector support.

CSORBT —Chip-Select Option Register Boot ROM

\$YFFA4A

15	14	13	12	11	10	9	6	5	4	3	1	0
MODE	BYTE	R/W	STRB	DSACK	SPACE	IPL	AVEC					

RESET:

0 1 1 1 1 0 1 1 0 1 1 0 0 0 0

CSOR[10:0] —Chip-Select Option Registers

\$YFFA4E–\$YFFA76

15	14	13	12	11	10	9	6	5	4	3	1	0
MODE	BYTE	R/W	STRB	DSACK	SPACE	IPL	AVEC					

RESET:

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

CSORBT, the option register for \overline{CSBOOT} , contains special reset values that support bootstrap operations from peripheral memory devices.

The following bit descriptions apply to both CSORBT and CSOR[10:0] option registers.

MODE — Asynchronous/Synchronous Mode

0 = Asynchronous mode selected (chip-select assertion determined by internal or external bus control signals)

1 = Synchronous mode selected (chip-select assertion synchronized with ECLK signal)

In asynchronous mode, the chip select is asserted synchronized with \overline{AS} or \overline{DS} .

The \overline{DSACK} field is not used in synchronous mode because a bus cycle is only performed as a synchronous operation. When a match condition occurs on a chip select programmed for synchronous operation, the chip select signals the EBI that an ECLK cycle is pending.

BYTE — Upper/Lower Byte Option

This field is used only when the chip-select 16-bit port option is selected in the pin assignment register. The following table lists upper/lower byte options.

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

R/W — Read/Write

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

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

STRB — Address Strobe/Data Strobe

0 = Address strobe

1 = Data strobe

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

3.7.3 Reset Timing

The $\overline{\text{RESET}}$ input must be asserted for a specified minimum period in order for reset to occur. External $\overline{\text{RESET}}$ assertion can be delayed internally for a period equal to the longest bus cycle time (or the bus monitor time-out period) in order to protect write cycles from being aborted by reset. While $\overline{\text{RESET}}$ is asserted, SIM pins are either in a disabled high-impedance state or are driven to their inactive states.

When an external device asserts $\overline{\text{RESET}}$ for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the $\overline{\text{RESET}}$ pin low for an additional 512 CLKOUT cycles after it detects that the $\overline{\text{RESET}}$ signal is no longer being externally driven, to guarantee this length of reset to the entire system.

If an internal source asserts a reset signal, the reset control logic asserts $\overline{\text{RESET}}$ for a minimum of 512 cycles. If the reset signal is still asserted at the end of 512 cycles, the control logic continues to assert $\overline{\text{RESET}}$ until the internal reset signal is negated.

After 512 cycles have elapsed, the reset input pin goes to an inactive, high-impedance state for ten 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_{DDSYN} in order for the MCU to operate. The following discussion assumes that V_{DDSYN} is applied before and during reset. This minimizes crystal start-up time. When V_{DDSYN} is applied at power-on, start-up time is affected by specific crystal parameters and by oscillator circuit design. V_{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_{DD} ramps up to the minimum specified value, and SIM pins are initialized. When V_{DD} reaches the specified minimum value, the clock synthesizer VCO begins operation. Clock frequency ramps up to the specified limp mode frequency. The external $\overline{\text{RESET}}$ line remains asserted until the clock synthesizer PLL locks and 512 CLKOUT cycles elapse.

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_{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 milliseconds 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 re-started 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 for eight levels of interrupt priority (0–7), seven automatic interrupt vectors, and 200 assignable interrupt vectors. All interrupts with priorities less than seven can be masked by the interrupt priority (IP) field in the status register. The CPU32 handles interrupts as a type of asynchronous exception.

Interrupt recognition is based on the states of interrupt request signals $\overline{\text{IRQ}}[7:1]$ and the IP mask value. Each of the signals corresponds to an interrupt priority. $\overline{\text{IRQ}}1$ has the lowest priority, and $\overline{\text{IRQ}}7$ has the highest priority.

The IP field consists of three bits. 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 (except for $\overline{\text{IRQ}}7$) from being recognized and processed. When IP contains %000, no interrupt is masked. During exception processing, the IP field is set to the priority of the interrupt being serviced.

Interrupt request signals can be asserted by external devices or by microcontroller modules. Request lines are connected internally by means of a wired NOR — simultaneous requests of differing priority can be made. Internal assertion of an interrupt request signal does not affect the logic state of the corresponding MCU pin.

External interrupt requests are routed to the CPU via the external bus interface and SIM interrupt control logic. The CPU treats external interrupt requests as though they come from the SIM.

External $\overline{\text{IRQ}}[6:1]$ are active-low level-sensitive inputs. External $\overline{\text{IRQ}}7$ is an active-low transition-sensitive input. $\overline{\text{IRQ}}7$ requires both an edge and a voltage level for validity.

$\overline{\text{IRQ}}[6:1]$ are maskable. $\overline{\text{IRQ}}7$ is nonmaskable. The $\overline{\text{IRQ}}7$ input is transition-sensitive in order to prevent redundant servicing and stack overflow. A nonmaskable interrupt is generated each time $\overline{\text{IRQ}}7$ is asserted, and 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 of equal or lower priority than the current IP mask value is made, the CPU does not recognize the occurrence of the request in any way.

3.8.1 Interrupt Acknowledge and Arbitration

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

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

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.

SR —Status Register

15	14	13	12	11	10	8	7	6	5	4	3	2	1	0
T1	T0	S	0	0	IP	0	0	0	X	N	Z	V	C	
RESET:														
0	0	1	0	0	1	1	1	0	0	0	U	U	U	U

System Byte

- T[1:0] —Trace Enable
- S —Supervisor/User State
- Bits [12:11] —Unimplemented
- IP[2:0] —Interrupt Priority Mask

User Byte (Condition Code Register)

- Bits [7:5] — Unimplemented
- X — Extend
- N — Negative
- Z — Zero
- V — Overflow
- C — Carry

4.4 Data Types

Six basic data types are supported:

- Bits
- Packed Binary Coded Decimal Digits
- Byte Integers (8 bits)
- Word Integers (16 bits)
- Long-Word Integers (32 bits)
- Quad-Word Integers (64 bits)

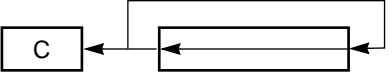

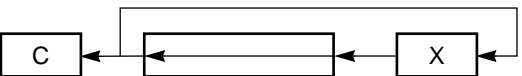
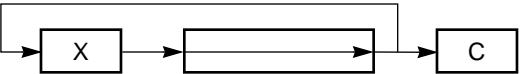
4.5 Addressing Modes

Addressing in the CPU32 is register-oriented. Most instructions allow the results of the specified operation to be placed either in a register or directly in memory. This flexibility eliminates the need for extra instructions to store register contents in memory. The CPU32 supports seven basic addressing modes:

- Register direct
- Register indirect
- Register indirect with index
- Program counter indirect with displacement
- Program counter indirect with index
- Absolute
- Immediate

Included in the register indirect addressing modes are the capabilities to post-increment, predecrement, and offset. The program counter relative mode also has index and offset capabilities. In addition to these addressing modes, many instructions implicitly specify the use of the status register, stack pointer, or program counter.

Table 20 Instruction Set Summary(Continued)

Instruction	Syntax	Operand Size	Operation
MOVES ¹	Rn, <ea> <ea>, Rn	8, 16, 32	Rn \Rightarrow Destination using DFC Source using SFC \Rightarrow Rn
MULS/MULU	<ea>, Dn <ea>, DI <ea>, Dh : DI	16 * 16 \Rightarrow 32 32 * 32 \Rightarrow 32 32 * 32 \Rightarrow 64	Source * Destination \Rightarrow Destination (signed or unsigned)
NBCD	\dot{I}	8 8	0 – Destination ₁₀ – X \Rightarrow Destination
NEG	\dot{I}	8, 16, 32	0 – Destination \Rightarrow Destination
NEGX	\dot{I}	8, 16, 32	0 – Destination – X \Rightarrow Destination
NOP	none	none	PC + 2 \Rightarrow PC
NOT	\dot{I}	8, 16, 32	Destination \Rightarrow Destination
OR	<ea>, Dn Dn, <ea>	8, 16, 32 8, 16, 32	Source + Destination \Rightarrow Destination
ORI	#<data>, <ea>	8, 16, 32	Data + Destination \Rightarrow Destination
ORI to CCR	#<data>, CCR	16	Source + CCR \Rightarrow SR
ORI to SR ¹	#<data>, SR	16	Source ; SR \Rightarrow SR
PEA	\dot{I}	32	SP – 4 \Rightarrow SP; <ea> \Rightarrow SP
RESET ¹	none	none	Assert RESET line
ROL	Dn, Dn #<data>, Dn \dot{I}	8, 16, 32 8, 16, 32 16	
ROR	Dn, Dn #<data>, Dn \dot{I}	8, 16, 32 8, 16, 32 16	
ROXL	Dn, Dn #<data>, Dn \dot{I}	8, 16, 32 8, 16, 32 16	
ROXR	Dn, Dn #<data>, Dn \dot{I}	8, 16, 32 8, 16, 32 16	
RTD	#d	16	(SP) \Rightarrow PC; SP + 4 + d \Rightarrow SP
RTE ¹	none	none	(SP) \Rightarrow SR; SP + 2 \Rightarrow SP; (SP) \Rightarrow PC; SP + 4 \Rightarrow SP; Restore stack according to format
RTR	none	none	(SP) \Rightarrow CCR; SP + 2 \Rightarrow SP; (SP) \Rightarrow PC; SP + 4 \Rightarrow SP
RTS	none	none	(SP) \Rightarrow PC; SP + 4 \Rightarrow SP
SBCD	Dn, Dn – (An), – (An)	8 8	Destination ₁₀ – Source ₁₀ – X \Rightarrow Destination
Scc	\dot{I}	8	If condition true, then destination bits are set to 1; else, destination bits are cleared to 0
STOP ¹	#<data>	16	Data \Rightarrow SR; STOP
SUB	<ea>, Dn Dn, <ea>	8, 16, 32	Destination – Source \Rightarrow Destination
SUBA	<ea>, An	16, 32	Destination – Source \Rightarrow Destination
SUBI	#<data>, <ea>	8, 16, 32	Destination – Data \Rightarrow Destination
SUBQ	#<data>, <ea>	8, 16, 32	Destination – Data \Rightarrow Destination
SUBX	Dn, Dn – (An), – (An)	8, 16, 32 8, 16, 32	Destination – Source – X \Rightarrow Destination

6.2 Address Map

The “Access” column in the QSM 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 QSMCR.

Table 24 QSM Address Map

Access	Address	15	8	7	0
S	\$YFFC00	QSM MODULE CONFIGURATION (QSMCR)			
S	\$YFFC02	QSM TEST (QTEST)			
S	\$YFFC04	QSM INTERRUPT LEVEL (QILR)		QSM INTERRUPT VECTOR (QIVR)	
S/U	\$YFFC06	NOT USED			
S/U	\$YFFC08	SCI CONTROL 0 (SCCR0)			
S/U	\$YFFC0A	SCI CONTROL 1 (SCCR1)			
S/U	\$YFFC0C	SCI STATUS (SCSR)			
S/U	\$YFFC0E	SCI DATA (SCDR)			
S/U	\$YFFC10	NOT USED			
S/U	\$YFFC12	NOT USED			
S/U	\$YFFC14	NOT USED		PQS DATA (PORTQS)	
S/U	\$YFFC16	PQS PIN ASSIGNMENT (PQSPAR)		PQS DATA DIRECTION (DDRQS)	
S/U	\$YFFC18	SPI CONTROL 0 (SPCR0)			
S/U	\$YFFC1A	SPI CONTROL 1 (SPCR1)			
S/U	\$YFFC1C	SPI CONTROL 2 (SPCR2)			
S/U	\$YFFC1E	SPI CONTROL 3 (SPCR3)		SPI STATUS (SPSR)	
S/U	\$YFFC20– \$YFFCFF	NOT USED			
S/U	\$YFFD00– \$YFFD1F	RECEIVE RAM (RR[0:F])			
S/U	\$YFFD20– \$YFFD3F	TRANSMIT RAM (TR[0:F])			
S/U	\$YFFD40– \$YFFD4F	COMMAND RAM (CR[0:F])			

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

The system software must stop each submodule before asserting STOP to avoid complications at re-start and to avoid data corruption. The SCI submodule receiver and transmitter should be disabled, and the operation should be verified for completion before asserting STOP. The QSPI submodule should be stopped by asserting the HALT bit in SPCR3 and by asserting STOP after the HALTA flag is set.

FRZ1 — Freeze 1

- 0 = Ignore the FREEZE signal on the IMB
- 1 = Halt the QSPI (on a transfer boundary)

FRZ1 determines what action is taken by the QSPI when the FREEZE signal of the IMB is asserted. FREEZE is asserted whenever the CPU enters the background mode.

FRZ0 — Freeze 0

Reserved

Bits [12:8] — Not Implemented

SUPV — Supervisor/Unrestricted

- 0 = User access
- 1 = Supervisor access

SUPV defines the assignable QSM registers as either supervisor-only data space or unrestricted data space.

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, non-zero IARB field value. Refer to **3.8 Interrupts** for more information.

QTEST — QSM Test Register

\$YFFC02

QTEST is used during factory testing of the QSM. Accesses to QTEST must be made while the MCU is in test mode.

QILR — QSM Interrupt Levels Register

\$YFFC04

15	14	13	11	10	8	7	0
0	0	ILQSPI			ILSCI		QIVR

RESET:

0 0 0 0 0 0 0 0

QILR determines the priority level of interrupts requested by the QSM and the vector used when an interrupt is acknowledged.

ILQSPI — Interrupt Level for QSPI

ILQSPI determines the priority of QSPI interrupts. This field must be given a value between \$0 (interrupts disabled) to \$7 (highest priority).

ILSCI — Interrupt Level of SCI

ILSCI determines the priority of SCI interrupts. This field must be given a value between \$0 (interrupts disabled) to \$7 (highest priority).

If ILQSPI and ILSCI are the same nonzero value, and both submodules simultaneously request interrupt service, QSPI has priority.

HALT — Halt

0 = Halt not enabled

1 = Halt enabled

When HALT is asserted, the QSPI stops on a queue boundary. It is in a defined state from which it can later be restarted.

SPSR — QSPI Status Register

\$YFFC1F

15	8	7	6	5	4	3	0
SPCR3							
SPIF							
MODF							
HALTA							
0							
CPTQP							
RESET:							
0 0 0 0 0 0 0 0							

SPSR contains QSPI status information. Only the QSPI can assert the bits in this register. The CPU reads this register to obtain status information and writes it to clear status flags.

SPIF — QSPI Finished Flag

0 = QSPI not finished

1 = QSPI finished

SPIF is set after execution of the command at the address in ENDQP.

MODF — Mode Fault Flag

0 = Normal operation

1 = Another SPI node requested to become the network SPI master while the QSPI was enabled in master mode (\overline{SS} input taken low).

The QSPI asserts MODF when the QSPI is the serial master (MSTR = 1) and the \overline{SS} input pin is negated by an external driver.

HALTA — Halt Acknowledge Flag

0 = QSPI not halted

1 = QSPI halted

HALTA is asserted when the QSPI halts in response to CPU assertion of HALT.

Bit 4 — Not Implemented

CPTQP — Completed Queue Pointer

CPTQP points to the last command executed. It is updated when the current command is complete. When the first command in a queue is executing, CPTQP contains either the reset value (\$0) or a pointer to the last command completed in the previous queue.

6.5.3 QSPI RAM

The QSPI contains an 80-byte block of dual-access static RAM that is used by both the QSPI and the CPU. The RAM is divided into three segments: receive data, transmit data, and command control data. Receive data is information received from a serial device external to the MCU. Transmit data is information stored by the CPU for transmission to an external peripheral. Command control data is used to perform the transfer.

Refer to the following illustration of the organization of the RAM.

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

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

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

CONT — Continue

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

BITSE — Bits per Transfer Enable

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

DT — Delay after Transfer

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

DSCK — PCS to SCK Delay

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

PCS[3:0] — Peripheral Chip Select

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

\overline{SS} — Slave Mode Select

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

6.5.4 Operating Modes

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

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

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

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

IDLE — Idle-Line Detected Flag

- 0 = SCI receiver did not detect an idle-line condition.
- 1 = SCI receiver detected an idle-line condition.

IDLE is disabled when RWU in SCCR1 is set. IDLE is set when the SCI receiver detects the idle-line condition specified by ILT in SCCR1. If cleared, IDLE will not set again until after RDRF is set. RDRF is set when a break is received, so that a subsequent idle line can be detected.

OR — Overrun Error Flag

- 0 = RDRF is cleared before new data arrives.
- 1 = RDRF is not cleared before new data arrives.

OR is set when a new byte is ready to be transferred from the receive serial shifter to the RDR, and RDRF is still set. Data transfer is inhibited until OR is cleared. Previous data in RDR remains valid, but data received during overrun condition (including the byte that set OR) is lost.

NF — Noise Error Flag

- 0 = No noise detected on the received data
- 1 = Noise occurred on the received data

NF is set when the SCI receiver detects noise on a valid start bit, on any data bit, or on a stop bit. It is not set by noise on the idle line or on invalid start bits. Each bit is sampled three times. If none of the three samples are the same logic level, the majority value is used for the received data value, and NF is set. NF is not set until an entire frame is received and RDRF is set.

FE — Framing Error Flag

- 0 = No framing error on the received data.
- 1 = Framing error or break occurred on the received data.

FE is set when the SCI receiver detects a zero where a stop bit was to have occurred. FE is not set until the entire frame is received and RDRF is set. A break can also cause FE to be set. It is possible to miss a framing error if RXD happens to be at logic level one at the time the stop bit is expected.

PF — Parity Error Flag

- 0 = No parity error on the received data
- 1 = Parity error occurred on the received data

PF is set when the SCI receiver detects a parity error. PF is not set until the entire frame is received and RDRF is set.

SCDR — SCI Data Register

\$YFFC0E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0

RESET:

0 0 0 0 0 0 0 U U U U U U U U U

SCDR contains two data registers at the same address. Receive data register (RDR) is a read-only register that contains data received by the SCI. The data comes into the receive serial shifter and is transferred to RDR. Transmit data register (TDR) is a write-only register that contains data to be transmitted. The data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When it is configured for 8-bit operation, they have no meaning or effect.

7 Standby RAM with TPU Emulation RAM

The TPURAM module contains a 2-Kbyte array of fast (two bus cycle) static RAM, which is especially useful for system stacks and variable storage. Alternately, it can be used by the TPU as emulation RAM for new timer algorithms.

7.1 Overview

The TPURAM can be mapped to any 4-Kbyte boundary in the address map, but must not overlap the module control registers. (Overlap makes the registers inaccessible.) Data can be read or written in bytes, word, or long words. TPURAM responds to both program and data space accesses. Data can be read or written in bytes, words, or long words. The TPURAM is powered by V_{DD} in normal operation. During power-down, the TPURAM contents are maintained by power on standby voltage pin V_{STBY} . Power switching between sources is automatic.

Access to the TPURAM array is controlled by the RASP field in TRAMMCR. This field can be encoded so that TPURAM responds to both program and data space accesses. This allows code to be executed from TPURAM, and permits the use of program counter relative addressing mode for operand fetches from the array.

An address map of the TPURAM control registers follows. All TPURAM control registers are located in supervisor data space.

Table 28 TPURAM Control Register Address Map

Access	Address	15	8	7	0
S	\$YFFB00	TPURAM MODULE CONFIGURATION REGISTER (TRAMMCR)			
S	\$YFFB02	TPURAM TEST REGISTER (TRAMTST)			
S	\$YFFB04	TPURAM BASE ADDRESS REGISTER (TRAMBAR)			
	\$YFFB06– \$YFFB3F	NOT USED			

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

7.2 TPURAM Register Block

There are three TPURAM control registers: the RAM module configuration register (TRAMMCR), the RAM test register (TRAMTST), and the RAM array base address registers (TRAMBAR).

There is an 8-byte minimum register block size for the module. Unimplemented register addresses are read as zeros, and writes have no effect.

7.3 TPURAM Registers

TRAMMCR —TPURAM Module Configuration Register

\$YFFB00

15	14	13	12	11	10	9	8	7	0
STOP	0	0	0	0	0	0	RASP	NOT USED	

RESET:

0 0 0 0 0 0 0 1

TSTOP —Stop Control

0 = RAM array operates normally.

1 = RAM array enters low-power stop mode.

This bit controls whether the RAM 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.

