MSC8101 Programmer's Quick Reference

16-Bit Digital Signal Processor

MSC8101PG/D
Revision 0, December 2000

MOTOROLA

How to reach us:

USA/EUROPE
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-303-675-2140
1-800-441-2447

Technical Information Center
1-800-521-6274

JAPAN
Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC
Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

Home Page
http://www.mot.com/SPS/DSP

DSP Helpline
http://www.motorola-dsp.com/contact
email: dsphelp@dsp.sps.mot.com

Contents

Figures

Tables

1 Introduction

This quick reference is designed to give programmers fast, easy access to summary information on key aspects of the MSC8101 device. It synthesizes and condenses information from multiple sources, including documentation on both the SC140 core and the MSC8101 device. For details on the topics covered here, refer to the documents listed in Table 1-1.

Table 1-1. MSC8101 and Related Documentation

Name	Description	Order Number
MSC8101 Data Sheet	Details the signals, AC/DC characteristics, PLL/DLL performance issues, package and pinout, and electrical design considerations of the MSC8101.	MSC8101/D
MSC8101 User's Guide	Details how to program the MSC8101. Outlines the system-level components and describes how the MSC8101 functions at the system level.	MSC8101UG/D
MSC8101 Reference Manual	Describes the MSC8101 architecture and functionality in detail, with a chapter on each of the MSC8101 blocks.	MSC8101RM/D
SC140 DSP Core Reference Manual	Covers the SC140 core architecture, instruction set, PLL and clock generator, and enhanced OnCE™ (EOnCE). Available in PDF at http://www.mot.com/SPS/DSP	MNSC140CORE/D
Application Notes	Cover various programming topics related to StarCore and the MSC8101; available at http://www.mot.com/SPS/DSP	N/A
• *SC100 Application Binary Interface Reference Manual* • *SC100 Assemply Language Tools User's Manual* • *SC100 C/C++ Compiler User's Manual*	Tools-related documentation available at http://www.mot.com/SPS/DSP	N/A

2 MSC8101 Block Diagram

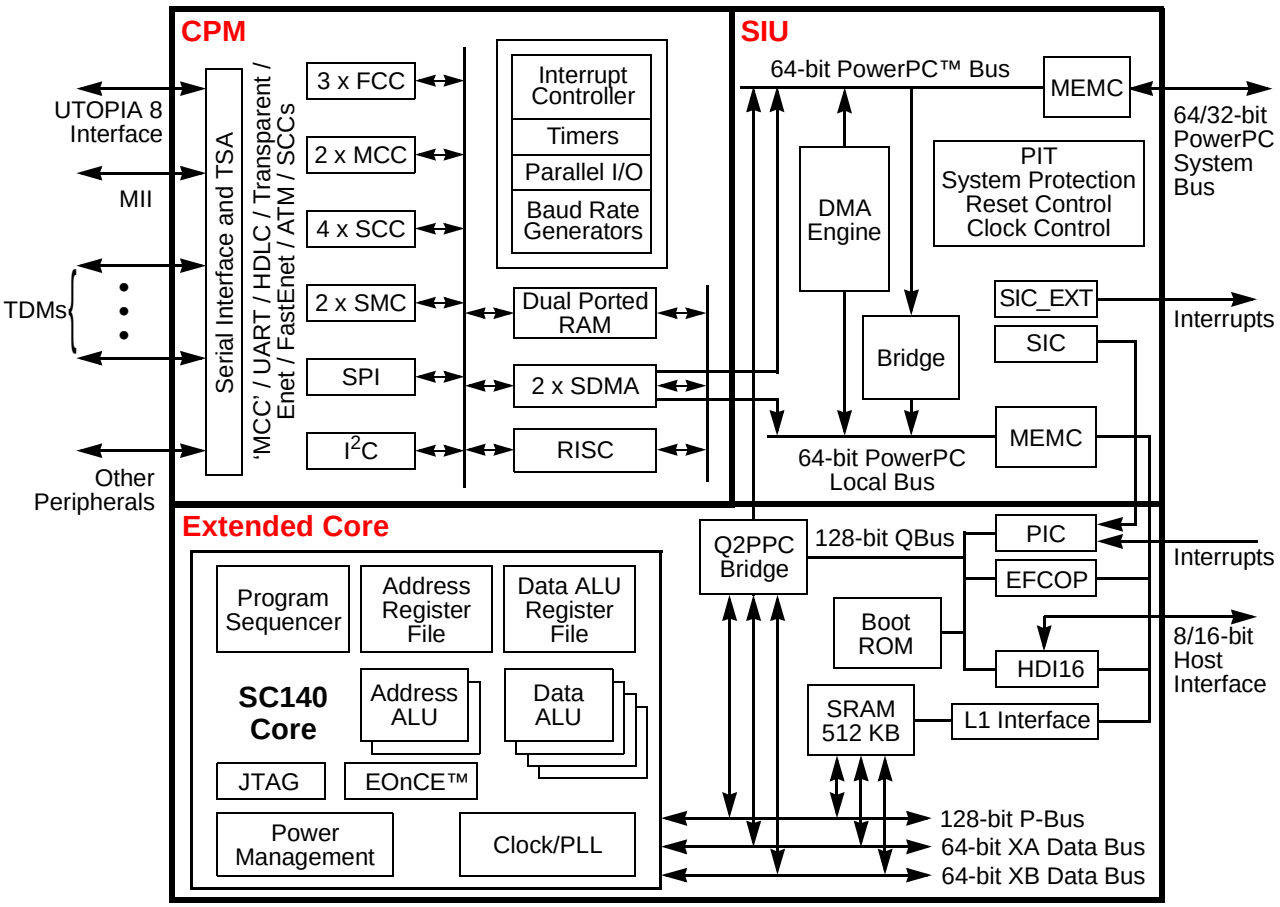
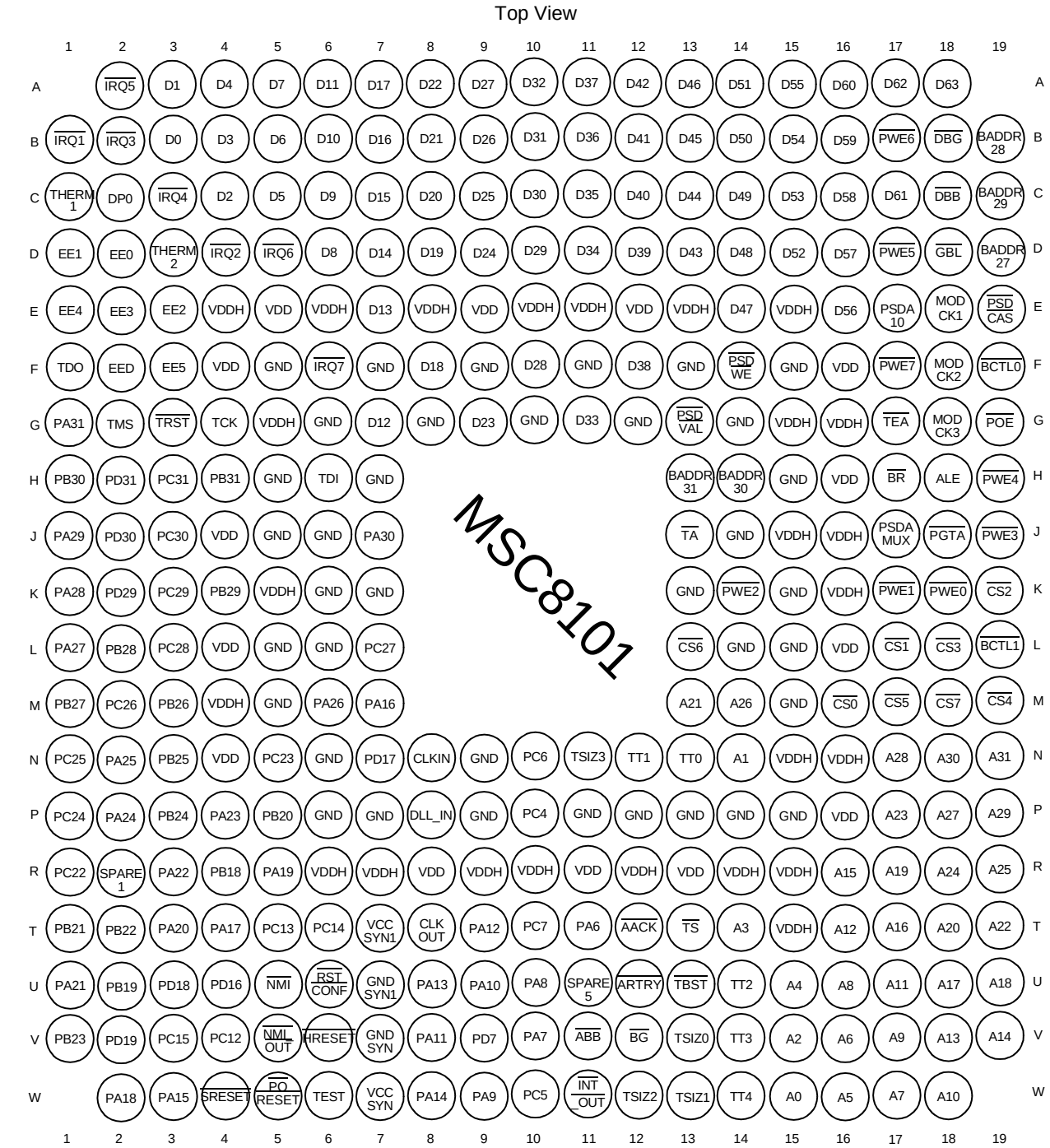


Figure 2-1. MSC8101 Block Diagram

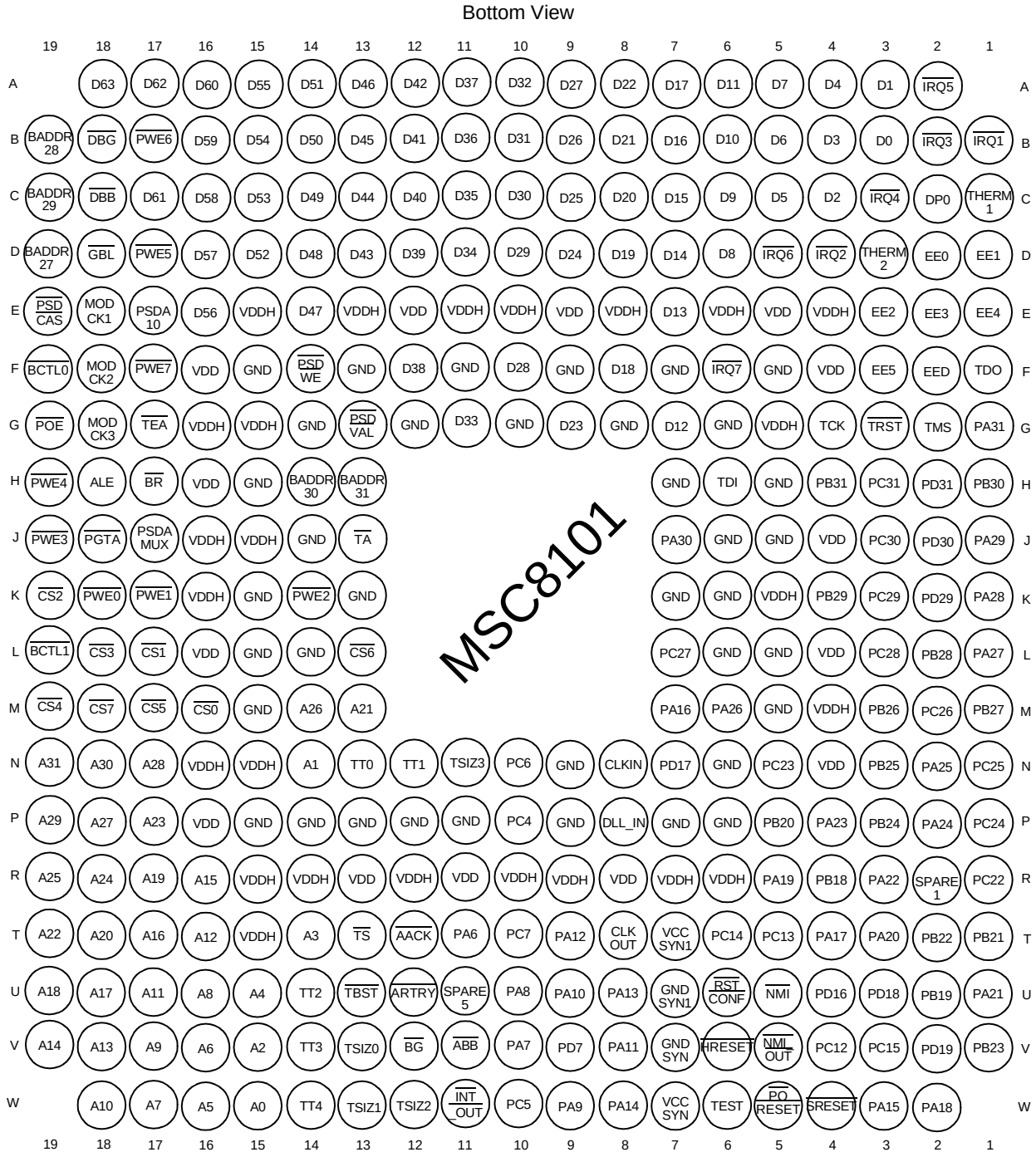
3 Pins and External Signals

3.1 Package Pinout



Note: Signal names in this figure are the default signals at reset, except for signals D1, D2, D18, E1, F3, and W11 which show the secondary signal at reset.

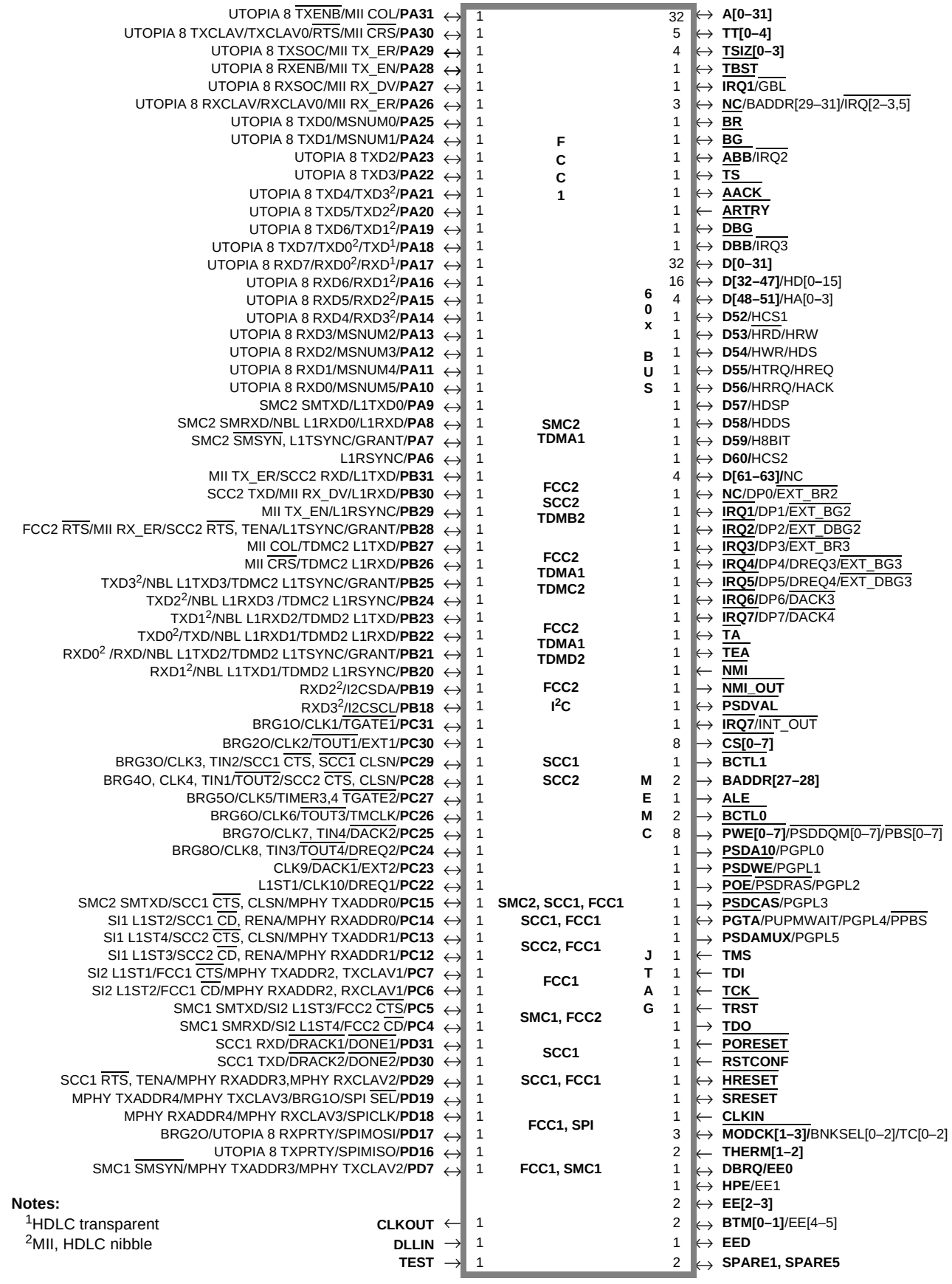
Figure 3-1. MSC8101 Flip Chip Plastic Ball Grid Array (FC-PBGA), Top View



Note: Signal names in this figure are the default signals at reset, except for signals D1, D2, D18, E1, F3, and W11 which show the secondary signal at reset.

Figure 3-2. MSC8101 Flip Chip Plastic Ball Grid Array (FC-PBGA), Bottom View

3.2 External Signals



Notes:

[1]HDLC transparent

[2]MII, HDLC nibble

Table 3-1. External Signals–SIU and Extended Core

Name	Data Direction	Description
A[0–31]	Input/Output	**60x Address Bus** When the MSC8101 is in external master bus mode, these pins function as the 60x address bus. The MSC8101 drives the address of its internal 60x bus masters and responds to addresses generated by external 60x bus masters. When the MSC8101 is in Internal Master Bus mode, these pins are used as address lines connected to memory devices and are controlled by the MSC8101 memory controller.
TT[0–4]	Input/Output	**60x Bus Transfer Type** The 60x bus master drives these pins during the address tenure to specify the type of transaction.
TSIZ[0–3]	Input/Output	**60x Transfer Size** The 60x bus master drives these pins with a value indicating the number of bytes transferred in the current transaction.
$\overline{\text{TBST}}$	Input/Output	**60x Bus Transfer Burst** The 60x bus master asserts this pin to indicate that the current transaction is a burst transaction (transfers four quad words).
$\overline{\text{IRQ1}}$	Input	**Interrupt Request 1** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{\text{GBL}}$	Input/Output	**Global** When a 60x master within the chip initiates a bus transaction, it drives this pin. When an external 60x master initiates a bus transaction, it should drive this pin. Assertion of this pin indicates that the transfer is global and it should be snooped by caches in the system.
NC		The primary (general-purpose) signal is a no connect (**NC**).
BADDR[29–31]	Output	**Burst Addresses 29–31** Outputs of the 60x memory controller. These pins are used in external master configuration. They connect directly to memory devices controlled by the MSC8101 memory controller.
$\overline{\text{IRQ[2–3,5]}}$	Input	**Interrupt Requests 2–3, 5** External lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{\text{BR}}$	Input/Output Output Input	**60x Bus Request** An output when an external arbiter is used. The MSC8101 asserts this pin to request ownership of the 60x bus. An input when an internal arbiter is used. An external master should assert this pin to request 60x bus ownership from the internal arbiter.
$\overline{\text{BG}}$	Input/Output Output Input	**60x Bus Grant** An output when an internal arbiter is used. The MSC8101 asserts this pin to grant 60x bus ownership to an external PowerPC bus master. An input when an external arbiter is used. The external arbiter should assert this pin to grant 60x bus ownership to the MSC8101.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
ABB	Input/Output Output Input	**60x Address Bus Busy** The MSC8101 asserts this pin for the duration of the address bus tenure. Following an address acknowledge (AACK) signal, which terminates the address bus tenure, the MSC8101 negates ABB for a fraction of a bus cycle and then stops driving this pin. The MSC8101 does not assume PowerPC 60x bus ownership as long as it senses that this pin is asserted by an external 60x bus master.
IRQ2	Input	**Interrupt Request 2** One of the eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
TS	Input/Output	**60x Bus Transfer Start** Signals the beginning of a new address bus tenure. The MSC8101 asserts this signal when one of its internal 60x bus masters (SC140 core or DMA) begins an address tenure. When the MSC8101 senses this pin being asserted by an external 60x bus master, it responds to the address bus tenure as required (snoop if enabled, access internal MSC8101 resources, memory controller support).
AACK	Input/Output	**60x Address Acknowledge** A 60x bus slave asserts this signal to indicate that it identified the address tenure. Assertion of this signal terminates the address tenure.
ARTRY	Input	**60x Address Retry** Assertion of this signal indicates that the bus transaction should be retried by the 60x bus master. The MSC8101 asserts this signal to enforce data coherency with its internal cache and to prevent deadlock situations.
DBG	Input/Output Output Input	**60x Data Bus Grant** An output when an internal arbiter is used. The MSC8101 asserts this pin as an output to grant 60x data bus ownership to an external PowerPC bus master. An input when an external arbiter is used. The external arbiter should assert this pin as an input to grant 60x data bus ownership to the MSC8101.
DBB	Input/Output Output Input	**60X Data Bus Busy** The MSC8101 asserts this pin as an output for the duration of the data bus tenure. Following a TA, which terminates the data bus tenure, the MSC8101 negates DBB for a fraction of a bus cycle and then stops driving this pin. The MSC8101 does not assume PowerPC 60x data bus ownership as long as it senses DBB is asserted by an external 60x bus master.
IRQ3	Input	**Interrupt Request 3** One of the eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
D[0–31]	Input/Output	**60x Data Bus Most Significant Word** In write transactions the 60x bus master drives the valid data on this bus. In read transactions the 60x slave drives the valid data on this bus. In Host Port Disabled mode, these 32 bits are part of the 64-bit PowerPC data bus. In Host Port Enabled mode, these bits are used as the PowerPC bus in 32-bit mode.
D[32–47]	Input/Output	**60x Data Bus Bits 32–47** In write transactions the 60x bus master drives the valid data on this bus. In read transactions the 60x slave drives the valid data on this bus.
HD[0–15]	Input/Output	**Host Data** When the HDI16 interface is enabled, these signals are lines 0-15 of the bidirectional tri-state data bus.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
D[48–51]	Input/Output	**60x Data Bus Bits 48–51** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HA[0–3]	Input	**Host Address Lines 0–3** When the HDI16 interface bus is enabled, this address line addresses internal host registers.
D52	Input/Output	**60x Data Bus Bit 52** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HCS1	Input	**Host Chip Select** When the HDI16 interface is enabled, this is the chip-select pin. The polarity of this pin is programmable.
D53	Input/Output	**60x Data Bus Bit 53** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
$\overline{\text{HRD}}$	Input	**Host Read Strobe** When the HDI16 is programmed to interface with a double data strobe host bus, this pin is the read data strobe input (HRD). The polarity of the data strobe is programmable.
HRW	Input	**Host Read Write Select** When the HDI16 interface is enabled in Single Strobe mode, this is the read/write input (HRW).
D54	Input/Output	**60x Data Bus Bit 54** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HWR	Input	**Host Write Data Strobe** When the HDI16 is programmed to interface with a double data strobe host bus, this pin is the write data strobe input (HWR). The polarity of the data strobe is programmable.
HDS	Input	**Host Data Strobe** When the HDI16 is programmed to interface with a single data strobe host bus, this pin is the data strobe input (HDS). The polarity of the data strobe is programmable.
D55	Input/Output	**60x Data Bus Bit 55** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HTRQ	Output	**Transmit Host Request** When the HDI16 is programmed to interface with a double host request host bus, this pin is the transmit host request output (HTRQ). The polarity of the host request is programmable. The host request is an open-drain output.
HREQ	Output	**Host Request** When the HDI16 is programmed to interface with a single host request host bus, this pin is the host request output (HREQ). The polarity of the host request is programmable. The host request may be programmed as a driven or open-drain output.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
D56	Input/Output	**60x Data Bus Bit 56** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HRRQ	Output	**Receive Host Request** When the HDI16 is programmed to interface with a double host request host bus, this pin is the receive host request output (HRRQ). The polarity of the host request is programmable. The host request is an open-drain output.
HACK	Input	**Host Acknowledge** When the HDI16 is programmed to interface with a single host request host bus, this pin is the host acknowledge input (HACK). The polarity of the host acknowledge is programmable.
D57	Input/Output	**60x Data Bus Bit 57** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HDSP	Input	**Host Data Strobe Polarity** When the HDI16 interface is enabled, this pin is the host data strobe polarity (HDSP).
D58	Input/Output	**60x Data Bus Bit 58** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HDDS	Input	**Host Dual Data Strobe** When the HDI16 interface is enabled, this pin is the host dual data strobe (HDDS).
D59	Input/Output	**60x Data Bus Bit 59** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
H8BIT	Input	**H8BIT** When the HDI16 interface is enabled, this bit determines if the interface is in 8-bit or 16-bit mode.
D60	Input/Output	**60x Data Bus Bit 60** In write transactions the 60x bus master drives the valid data on this pin. In read transactions the 60x slave drives the valid data on this pin.
HCS2	Input	**Host Chip Select 2** When the HDI16 interface is enabled, this is the chip-select pin. The polarity of this pin is programmable.
D[61–63]	Input/Output	**60x Data Bus Bits 61–63** Used only in PowerPC-only mode. In write transactions the 60x bus master drives the valid data on this bus. In read transactions the 60x slave drives the valid data on this bus.
NC		The dedicated signal is a no connect (**NC**).
NC		The primary (general-purpose) signal is a no connect (**NC**).
DP0	Input/Output	**60x Data Parity 0** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity zero pin should give odd parity (odd number of ones) on the group of signals that includes data parity 0 and D[0–7].
$\overline{\text{EXT\_BR2}}$	Input	**External PowerPC Bus Request 2** An external master asserts this pin to request PowerPC 60x bus ownership from the internal arbiter.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
$\overline{\text{IRQ1}}$	Input	**Interrupt Request 1** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP1	Input/Output	**60x Data Parity 1** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity one pin should give odd parity (odd number of ones) on the group of signals that includes data parity 1 and D[8–15].
$\overline{\text{EXT\_BG2}}$	Output	**External Bus Grant 2** The MSC8101 asserts this pin to grant PowerPC 60x bus ownership to an external PowerPC bus master.
$\overline{\text{IRQ2}}$	Input	**Interrupt Request 2** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP2	Input/Output	**60x Data Parity 2** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity two pin should give odd parity (odd number of ones) on the group of signals that includes data parity 2 and D[16–23].
$\overline{\text{EXT\_DBG2}}$	Output	**External Data Bus Grant 2** The MSC8101 asserts this pin to grant PowerPC 60x data bus ownership to an external PowerPC bus master.
$\overline{\text{IRQ3}}$	Input	**Interrupt Request 3** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP3	Input/Output	**60x Data Parity 3** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity three pin should give odd parity (odd number of ones) on the group of signals that includes data parity 3 and D[24–31].
$\overline{\text{EXT\_BR3}}$	Input	**External PowerPC Bus Request 3** An external master asserts this pin to request PowerPC 60x bus ownership from the internal arbiter.
$\overline{\text{IRQ4}}$	Input	**Interrupt Request 4** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP4	Input/Output	**60x Data Parity 4** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity four pin should give odd parity (odd number of ones) on the group of signals that includes data parity 4 and D[32–39].
DREQ3	Input	**DMA Request 3** An external peripheral uses this pin to request DMA service.
$\overline{\text{EXT\_BG3}}$	Output	**External PowerPC Bus Grant 3** The MSC8101 asserts this pin to grant PowerPC 60x bus ownership to an external PowerPC bus master.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
$\overline{\text{IRQ5}}$	Input	**Interrupt Request 5** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP5	Input/Output	**60x Data Parity 5** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity five pin should give odd parity (odd number of ones) on the group of signals that includes data parity 5 and D[40–47].
DREQ4	Input	**DMA Request 4** An external peripheral uses this pin to request DMA service.
$\overline{\text{EXT\_DBG3}}$	Output	**External Data Bus Grant 3** The MSC8101 asserts this pin to grant PowerPC 60x data bus ownership to an external PowerPC bus master.
$\overline{\text{IRQ6}}$	Input	**Interrupt Request 6** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP6	Input/Output	**60x Data Parity 6** The 60x agent that drives the data bus also drives the data parity signals. The value driven on the data parity six pin should give odd parity (odd number of ones) on the group of signals that includes data parity 6 and D[48–55].
$\overline{\text{DACK3}}$	Output	**DMA Acknowledge 3** The DMA drives this output to acknowledge the DMA transaction on the PowerPC 60x bus.
$\overline{\text{IRQ7}}$	Input	**Interrupt Request 7** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP7	Input/Output	**60x Data Parity 7** The 60x master or slave that drives the data bus also drives the data parity signals. The value driven on the data parity seven pin should give odd parity (odd number of ones) on the group of signals that includes data parity 7 and D[56–63].
$\overline{\text{DACK4}}$	Output	**DMA Acknowledge 4** The DMA drives this output to acknowledge the DMA transaction on the PowerPC 60x bus.
$\overline{\text{TA}}$	Input/Output	**Transfer Acknowledge** Indicates that a 60x data beat is valid on the data bus. For 60x single beat transfers, assertion of $\overline{\text{TA}}$ indicates the termination of the transfer. For 60x burst transfers, $\overline{\text{TA}}$ is asserted four times to indicate the transfer of four data beats with the last assertion indicating the termination of the burst transfer.
$\overline{\text{TEA}}$	Input/Output	**Transfer Error Acknowledge** Indicates a bus error. 60x masters within the MSC8101 monitor the state of this pin. The MSC8101 internal PowerPC bus monitor can assert this pin if it identifies a PowerPC 60x bus transfer that is hung.
$\overline{\text{NMI}}$	Input	**Non-Maskable Interrupt** When an external device asserts this line, the MSC8101 NMI input is asserted.
$\overline{\text{NMI\_OUT}}$	Output	**Non-Maskable Interrupt** Driven from the MSC8101 internal interrupt controller. Assertion of this output indicates that a non-maskable interrupt, pending in the MSC8101 internal interrupt controller, is waiting to be handled by an external host.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
PSDVAL	Input/Output	**60x Data Valid** Indicates that a data beat is valid on the data bus. The difference between the TA pin and PSDVAL is that the TA pin is asserted to indicate 60x data transfer terminations while the PSDVAL signal is asserted with each data beat movement. Thus, when TA is asserted, PSDVAL is asserted, but when PSDVAL is asserted, TA is not necessarily asserted. For example when the SDMA initiates a double word (2x64 bits) transfer to a memory device that has a 32-bit port size, PSDVAL is asserted three times without TA, and finally both pins are asserted to terminate the transfer.
IRQ7	Input	**Interrupt Request 7** One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
INT_OUT	Output	**Interrupt Output** Driven from the MSC8101 internal interrupt controller. Assertion of this output indicates that an unmasked interrupt is pending in the MSC8101 internal interrupt controller.
CS[0–7]	Output	**Chip Select** Enable specific memory devices or peripherals connected to MSC8101 buses.
BCTL1	Output	**Buffer Control 1** Controls buffers on the PowerPC 60x data bus. Usually used with BCTL0. The exact function of this pin is defined by the value of SIUMCR[BCTLC]. See **Table 6-8**, *SIU Registers*, on page -68, for SIUMCR[BCTLC] values.
BADDR[27–28]	Output	**Burst Address 27–28** Two of five outputs of the 60x memory controller. These pins are used in external master configuration. They connect directly to memory devices controlled by the MSC8101 memory controller.
ALE	Output	**Address Latch Enable** Controls the external address latch used in external master 60x bus configuration.
BCTL0	Output	**Buffer Control 0** Controls buffers on the PowerPC 60x data bus. The exact function of this pin is defined by the value of SIUMCR[BCTLC]. See **Table 6-8**, *SIU Registers*, on page -68, for SIUMCR[BCTLC] values.
PWE[0–7]	Output	**60x Bus Write Enable** Outputs of the PowerPC 60x bus general-purpose chip-select machine (GPCM). These pins select byte lanes for write operations.
PSDDQM[0–7]	Output	**60x Bus SDRAM DQM** Outputs of the SDRAM control machine. These pins select specific byte lanes of SDRAM devices.
PBS[0–7]	Output	**60x Bus UPM Byte Select** Outputs of the user-programmable memory (UPM) in the memory controller. These pins select specific byte lanes during memory operations. The timing of these pins is programmed in the UPM. The actual driven value depends on the address and size of the transaction and the port size of the accessed device.
PSDA10	Output	**60x Bus SDRAM A10** Output from the 60x bus SDRAM controller. This pin is part of the address when a row address is driven. It is part of the command when a column address is driven.
PGPL0	Output	**60x Bus UPM General-Purpose Line 0** One of six general-purpose output lines of the UPM. The values and timing of this pin is programmed in the UPM.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
$\overline{\text{PSDWE}}$	Output	**60x Bus SDRAM Write Enable** Output from the 60x bus SDRAM controller. This pin should connect to the SDRAM WE input signal.
PGPL1	Output	**60x Bus UPM General-Purpose Line 1** One of six general-purpose output lines from the UPM. The values and timing of this pin are programmed in the UPM.
$\overline{\text{POE}}$	Output	**60x Bus Output Enable** Output of the 60x bus GPCM. Controls the output buffer of memory devices during read operations.
$\overline{\text{PSDRAS}}$	Output	**60x Bus SDRAM Ras** Output from the 60x bus SDRAM controller. This pin should connect to the SDRAM RAS input signal.
PGPL2	Output	**60x Bus UPM General-Purpose Line 2** One of six general-purpose output lines from the UPM. The values and timing of this pin are programmed in the UPM.
$\overline{\text{PSDCAS}}$	Output	**60x Bus SDRAM CAS** Output from the 60x bus SDRAM controller. This pin should connect to the SDRAM CAS input signal.
PGPL3	Output	**60x Bus UPM General-Purpose Line 3** One of six general-purpose output lines from the UPM. The values and timing of this pin are programmed in the UPM.
$\overline{\text{PGTA}}$	Input	**60x GPCM TA** Terminates transactions during GPCM operation. Requires an external pull up resistor for proper operation.
PUPMWAIT	Input	**60x Bus UPM Wait** Input to the UPM. An external device can hold this pin high to force the UPM to wait until the device is ready for the operation to continue.
PGPL4	Output	**60x Bus UPM General-Purpose Line 4** One of six general-purpose output lines from the UPM. The values and timing of this pin are programmed in the UPM.
$\overline{\text{PPBS}}$	Output	**60x Bus Parity Byte Select** In systems in which data parity is stored in a separate chip, this output is the byte-select for that chip.
PSDAMUX	Output	**60x Bus SDRAM Address Multiplexer** Controls the 60x SDRAM address multiplexer when the MSC8101 is in External Master mode.
PGPL5	Output	**60x Bus UPM General-Purpose Line 5** One of six general-purpose output lines from the UPM. The values and timing of this pin are programmed in the UPM.
TMS	Input	**Test Mode Select (JTAG)** Controls the state of the MSC8101 JTAG/COP controller.
TDI	Input	**Test Data In (JTAG)** Data input to the MSC8101 JTAG/COP controller.
TCK	Input	**Test Clock (JTAG)** Provides the clock input for the MSC8101 JTAG/COP controller.
$\overline{\text{TRST}}$	Input	**Test Reset (JTAG)** The reset input to the MSC8101 JTAG/COP controller.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
TDO	Output	**Test Data Out (JTAG)** Data output from the MSC8101 JTAG/COP controller.
$\overline{\text{PORESET}}$	Input	**Power-On Reset** When asserted, this input line causes the MSC8101 to enter power-on reset state.
$\overline{\text{RSTCONF}}$	Input	**Reset Configuration** Used during the reset configuration sequence of the chip. For a detailed explanation of its function, see **Section 5.1.1**, *Power-On Reset Flow*, and **Section 5.2**, *Hardware Reset Configuration*, of the *MSC8101 Reference Manual*.
$\overline{\text{HRESET}}$	Input/Output	**Hard Reset** When asserted, this open-drain line causes the MSC8101 to enter hard reset state.
$\overline{\text{SRESET}}$	Input/Output	**Soft Reset** When asserted, this open-drain line causes the MSC8101 to enter soft reset state.
CLKIN	Input	**Clock In** Primary clock input to the MSC8101 PLL.
MODCK[1–3]	Input	**Clock Modes 1–3** Define the operating mode of internal clock circuits.
BNKSEL[0–2]	Output	**Bank Select 0–2** Select the SDRAM bank when the MSC8101 is in 60x-compatible bus mode. BNKSEL*x* is the msb of the three BNKSEL signals.
TC[0–2]	Output	**Transfer Code 0–2** Supply information for debug purposes for each of the bus transactions initiated by the MSC8101.
THERM[1–2]	—	Leave disconnected.
DBREQ	Input	**Debug Request** Determines whether to go immediately into SC140 Debug mode when $\overline{\text{PORESET}}$ is deasserted.
EE0		**Enhanced OnCE (EOnCE) Event 0** After $\overline{\text{PORESET}}$ is deasserted, you can configure EE0 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin.
	Input	Debug request, enable Address Event Detection Channel 0, or generate one of the EOnCE events.
	Output	Detection by Address Event Detection Channel 0. Used to trigger external debugging equipment.
HPE	Input	**Host Port Enable** When this pin is asserted during $\overline{\text{PORESET}}$, the Host port is enabled, the PowerPC 60x data bus is 32 bits wide, and the Host *must* program the reset configuration word.
EE1		**EOnCE Event 1** After $\overline{\text{PORESET}}$ is deasserted, you can configure EE1 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin.
	Input	Enable Address Event Detection Channel 1 or generate one of the EOnCE events.
	Output	Debug Acknowledge or detection by Address Event Detection Channel 1. Used to trigger external debugging equipment.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
EE2		**EOnCE Event 2** After PORESET is deasserted, you can configure EE2 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin.
	Input	Enable Address Event Detection Channel 2 or generate one of the EOnCE events or enable the Event Counter.
	Output	Detection by Address Event Detection Channel 2. Used to trigger external debugging equipment.
EE3		**EOnCE Event 3** After PORESET is deasserted, you can configure EE3 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin, as well as information on the ERCV Register.
	Input	Enable Address Event Detection Channel 3 or generate one of the EOnCE events.
	Output	EOnCE Receive Register (ERCV) was read by the DSP. Used to trigger external debugging equipment.
BTM[0–1]	Input	**Boot Mode 0–1** Determines the MSC8101 boot mode when PORESET is deasserted. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to set these pins.
EE4		**EOnCE Event 4** After PORESET is deasserted, you can configure EE4 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin, as well as information on the ETRSMT Register.
	Input	Enable Address Event Detection Channel 4 or generate one of the EOnCE events
	Output	EOnCE Transmit Register (ETRSMT) was written by the DSP. Used to trigger external debugging equipment.
EE5		**EOnCE Event 5** After PORESET is deasserted, you can configure EE5 as an input (default) or an output. See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin.
	Input	Enable Address Event Detection Channel 5.
	Output	Detection by Address Event Detection Channel 5. Used to trigger external debugging equipment.
EED		**Enhanced OnCE (EOnCE) Event Detection** After PORESET is deasserted, you can configure EED as an input (default) or output: See the *Emulation and Debug* chapter in the *SC140 DSP Core Reference Manual* for details on how to configure this pin.
	Input	Enable the Data Event Detection Channel.
	Output	Detection by the Data Event Detection Channel. Used to trigger external debugging equipment.
SPARE1, 5	—	**Spare Pins** Leave disconnected for backward compatibility with future revisions of this device.
CLKOUT	Output	**CLKOUT** The 60x bus clock.
DLL_IN	Input	**DLL_IN** For synchronization with an external device.
TEST	Input	**TEST** For test purposes. You *must* connect it to GND.

Table 3-1. External Signals–SIU and Extended Core (Continued)

Name	Data Direction	Description
Power Supply		**VDD** The power supply of the internal logic. **VDDH** The power supply of the I/O Buffers. **VCCSYN** The power supply of the PLL circuitry. **GNDSYN** A special ground of the PLL circuitry. **GNDSYN1** A special ground of the SC140 core PLL circuitry. **VCCSYN1** The power supply of the SC140 core PLL circuitry.

Table 3-2. External Signals–CPM

Name		Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated I/O**		
PA31	FCC1:UTOPIA 8 master $\overline{\text{TxEnb}}$	Output	FCC1: UTOPIA 8 Master Transmit Enable
	FCC1:UTOPIA 8 slave $\overline{\text{TxEnb}}$	Input	FCC1: UTOPIA 8 Slave Transmit Enable
	FCC1:MII COL	Input	FCC1: Media Independent Interface Collision Detect
PA30	FCC1:UTOPIA 8 slave TxClav	Output	FCC1: UTOPIA 8 Slave Transmit Cell Available
	FCC1:UTOPIA 8 master TxClav/UTOPIA 8 master TxClav0 MPHY, direct polling	Input	FCC1: UTOPIA 8 Master Transmit Cell Available FCC1: UTOPIA 8 Master Transmit Cell Available 0 Multiple PHY Direct Polling
	FCC1:$\overline{\text{RTS}}$	Output	FCC1: Request To Send
	FCC1:MII $\overline{\text{CRS}}$	Input	FCC1: Media Independent Interface Carrier Sense
PA29	FCC1:UTOPIA 8 TxSOC	Output	FCC1: UTOPIA 8 Transmit Start of Cell
	FCC1:MII TX_ER	Output	FCC1: Media Independent Interface Transmit Error
PA28	FCC1:UTOPIA 8 master $\overline{\text{RxEnb}}$	Output	FCC1: UTOPIA 8 Master Receive Enable
	FCC1:UTOPIA 8 slave $\overline{\text{RxEnb}}$	Input	FCC1: UTOPIA 8 Slave Receive Enable
	FCC1:MII TX_EN	Output	FCC1: Media Independent Interface Transmit Enable
PA27	FCC1:UTOPIA 8 RxSOC	Input	FCC1: UTOPIA 8 Receive Start of Cell
	FCC1:MII RX_DV	Input	FCC1: Media Independent Interface Receive Data Valid

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated I/O		
PA26	FCC1:UTOPIA 8 slave RxClav	Output	FCC1: UTOPIA 8 Slave Receive Cell Available
	FCC1:UTOPIA 8 master RxClav	Input	FCC1: UTOPIA 8 Master Receive Cell Available
	FCC1:UTOPIA 8 master RxClav0 MPHY, direct polling	Input	FCC1: UTOPIA 8 Master Receive Cell Available 0 Direct Polling
	FCC1:MII RX_ER	Input	FCC1: Media Independent Interface Receive Error
PA25	FCC1:UTOPIA 8 TxD0	Output	FCC1: UTOPIA 8 Transmit Data Bit 0
	MSNUM0	Output	Module Serial Number Bit 0
PA24	FCC1:UTOPIA 8 TxD1	Output	FCC1: UTOPIA 8 Transmit Data Bit 1
	MSNUM1	Output	Module Serial Number Bit 1
PA23	FCC1:UTOPIA 8 TxD2	Output	FCC1: UTOPIA 8 Transmit Data Bit 2
PA22	FCC1:UTOPIA 8 TxD3	Output	FCC1: UTOPIA 8 Transmit Data Bit 3
PA21	FCC1:UTOPIA 8 TxD4	Output	FCC1: UTOPIA 8 Transmit Data Bit 4
	FCC1:MII, HDLC nibble TxD3	Output	FCC1: Media Independent Interface, HDLC Nibble Transmit Data Bit 3
PA20	FCC1:UTOPIA 8 TxD5	Output	FCC1: UTOPIA 8 Transmit Data Bit 5
	FCC1:MII, HDLC nibble TxD2	Output	FCC1: Media Independent Interface, HDLC Nibble Transmit Data Bit 2
PA19	FCC1:UTOPIA 8 TxD6	Output	FCC1: UTOPIA 8 Transmit Data Bit 6
	FCC1:MII, HDLC nibble TxD1	Output	FCC1: Media Independent Interface, HDLC Nibble Transmit Data Bit 1
PA18	FCC1:UTOPIA 8 TxD7	Output	FCC1: UTOPIA 8 Transmit Data Bit 7
	FCC1:MII, HDLC nibble TxD0	Output	FCC1: Media Independent Interface, HDLC Nibble Transmit Data Bit 0
	FCC1:HDLC transparent TxD	Output	FCC1: HDLC, Transparent Serial Transmit Data Bit
PA17	FCC1:UTOPIA 8 RxD7	Input	FCC1: UTOPIA 8 Receive Data Bit 7
	FCC1:MII, HDLC nibble RxD0	Input	FCC1: Media Independent Interface, HDLC Nibble Receive Data Bit 0
	FCC1:HDLC transparent RxD	Input	FCC1: HDLC Transparent Serial Receive Data Bit
PA16	FCC1:UTOPIA 8 RxD6	Input	FCC1: UTOPIA 8 Receive Data Bit 6
	FCC1:MII, HDLC nibble RxD1	Input	FCC1: Media Independent Interface, HDLC Nibble Receive Data Bit 1

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated I/O**		
PA15	FCC1:UTOPIA 8 RxD5	Input	FCC1: UTOPIA 8 Receive Data Bit 5
	FCC1:MII, HDLC nibble RxD2	Input	FCC1: Media Independent Interface, HDLC Nibble Receive Data Bit 2
PA14	FCC1:UTOPIA 8 RxD4	Input	FCC1: UTOPIA 8 Receive Data Bit 4
	FCC1:MII, HDLC nibble RxD3	Input	FCC1: Media Independent Interface, HDLC Nibble Receive Data Bit 3
PA13	FCC1:UTOPIA 8 RxD3	Input	FCC1: UTOPIA 8 Receive Data Bit 3
	MSNUM2	Output	Module Serial Number Bit 2
PA12	FCC1:UTOPIA 8 RxD2	Input	FCC1: UTOPIA 8 Receive Data Bit 2
	MSNUM3	Output	Module Serial Number Bit 3
PA11	FCC1:UTOPIA 8 RxD1	Input	FCC1: UTOPIA 8 Receive Data Bit 1
	MSNUM4	Output	Module Serial Number Bit 4
PA10	FCC1:UTOPIA 8 RxD0	Input	FCC1: UTOPIA 8 Receive Data Bit 0
	MSNUM5	Output	Module Serial Number Bit 5
PA9	SMC2:SMTXD	Output	SMC2: Serial Management Transmit Data
	TDM_A1:L1TXD0	Inout	Time-Division Multiplexing A1: Layer 1 Transmit Data Bit 0
PA8	SMC2:SMRXD	Input	SMC2: Serial Management Receive Data
	TDM_A1:nibble L1RXD0	Input	Time-Division Multiplexing A1: Layer 1 Nibble Receive Data Bit 0
	TDM_A1:serial L1RXD	Inout	Time-Division Multiplexing A1: Layer 1 Serial Receive Data
PA7	SMC2:$\overline{\text{SMSYN}}$	Input	SMC2: Serial Management Synchronization
	TDM_A1:L1TSYNC/GRANT	Input	Time-Division Multiplexing A1: Layer 1 Transmit Synchronization/Grant
PA6	TDM_A1:L1RSYNC	Input	Time-Division Multiplexing A1: Layer 1 Receive Synchronization
PB31	FCC2:MII TX_ER	Output	FCC2: Media Independent Interface Transmit Error
	SCC2:RxD	Input	SCC2: Receive Data
	TDM_B2:L1TXD	Inout	Time-Division Multiplexing B2: Layer 1 Transmit Data
PB30	SCC2:TxD	Output	SCC2: Transmit Data
	FCC2:MII RX_DV	Input	FCC2: Media Independent Interface Receive Data Valid
	TDM_B2:L1RXD	Inout	Time-Division Multiplexing B2: Layer 1 Receive Data

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated I/O		
PB29	FCC2:MII TX_EN	Output	FCC2: Media Independent Interface Transmit Enable
	TDM_B2:L1RSYNC	Input	Time-Division Multiplexing B2: Layer 1 Receive Synchronization
PB28	FCC2:$\overline{\text{RTS}}$	Output	FCC2: Request to Send
	FCC2:MII RX_ER	Input	FCC2: Media Independent Interface Receive Error
	SCC2:$\overline{\text{RTS}}$, TENA	Output	SCC2: Request to Send, Transmit Enable
	TDM_B2:L1TSYNC/GRANT	Input	Time-Division Multiplexing B2: Layer 1 Transmit Synchronization/Grant
PB27	FCC2:MII COL	Input	FCC2: Media Independent Interface Collision Detect
	TDM_C2:L1TXD	Inout	Time-Division Multiplexing C2: Layer 1 Transmit Data
PB26	FCC2:MII $\overline{\text{CRS}}$	Input	FCC2: Media Independent Interface Carrier Sense Input
	TDM_C2:L1RXD	Inout	Time-Division Multiplexing C2: Layer 1 Receive Data
PB25	FCC2:MII, HDLC nibble TxD3	Output	FCC2: Media Independent Interface, HDLC Nibble Transmit Data Bit 3
	TDM_A1:nibble L1TXD3	Output	Time-Division Multiplexing A1:Nibble Layer 1 Transmit Data Bit 3
	TDM_C2:L1TSYNC/GRANT	Input	Time-Division Multiplexing C2: Layer 1 Transmit Synchronization/Grant
PB24	FCC2:MII, HDLC nibble TxD2	Output	FCC2: Media Independent Interface, HDLC Nibble Transmit Data Bit 2
	TDM_A1:nibble L1RXD3	Input	Time-Division Multiplexing A1:Nibble Layer 1 Receive Data Bit 3
	TDM_C2:L1RSYNC	Input	Time-Division Multiplexing C2: Layer 1 Receive Synchronization
PB23	FCC2:MII, HDLC nibble TxD1	Output	FCC2: Media Independent Interface, HDLC Nibble Transmit Data Bit 1
	TDM_A1:nibble L1RXD2	Input	Time-Division Multiplexing A1:Nibble Layer 1 Receive Data Bit 2
	TDM_D2:L1TXD	Inout	Time-Division Multiplexing D2: Layer 1 Transmit Data
PB22	FCC2:MII, HDLC nibble TxD0	Output	FCC2: Media Independent Interface, HDLC Nibble Transmit Data Bit 0
	HDLC transparent TxD	Output	FCC2: HDLC Transparent Serial Transmit Data
	TDM_A1:nibble L1RXD1	Input	Time-Division Multiplexing A1:Nibble Layer 1 Receive Data Bit 1
	TDM_D2:L1RXD	Inout	Time-Division Multiplexing D2: Layer 1 Receive Data

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated I/O**		
PB21	FCC2:MII, HDLC nibble RxD0	Input	FCC2: Media Independent Interface, HDLC Nibble Receive Data Bit 0
	FCC2:HDLC transparent RxD	Input	FCC2: HDLC Transparent Serial Receive Data Bit
	TDM_A1:nibble L1TXD2	Output	Time-Division Multiplexing A1:Nibble Layer 1 Transmit Data Bit 2
	TDM_D2:L1TSYNC/GRANT	Input	Time-Division Multiplexing D2: Layer 1 Transmit Synchronization/Grant
PB20	FCC2:MII, HDLC nibble RxD1	Input	FCC2: Media Independent Interface, HDLC Nibble Receive Data Bit 1
	TDM_A1:nibble L1TXD1	Output	Time-Division Multiplexing A1:Nibble Layer 1 Transmit Data Bit 1
	TDM_D2:L1RSYNC	Input	Time-Division Multiplexing D2: Layer 1 Receive Synchronization
PB19	FCC2:MII, HDLC nibble RxD2	Input	FCC2: Media Independent Interface, HDLC Nibble Receive Data Bit 2
	I2C:SDA	Inout	I2C: Inter-Integrated Circuit Serial Data
PB18	FCC2:MII, HDLC nibble RxD3	Input	FCC2: Media Independent Interface, HDLC Nibble Receive Data Bit 3
	I2C:SCL	Inout	I2C: Inter-Integrated Circuit Serial Clock
PC31	BRG1O	Output	Baud-Rate Generator 1 Output
	CLK1	Input	Clock 1
	Timer1/2: $\overline{\text{TGATE1}}$	Input	Timer 1/2: Timer Gate 1
PC30	BRG2O	Output	Baud-Rate Generator 2 Output
	CLK2	Input	Clock 2
	Timer1:$\overline{\text{TOUT1}}$	Output	Timer 1: Timer Output 1
	EXT1	Input	External Request Line 1
PC29	BRG3O	Output	Baud-Rate Generator 3 Output
	CLK3/TIN2	Input	Clock 3
		Input	Timer Input 2
	SCC1:$\overline{\text{CTS}}$, CLSN	Input	SCC1: Clear to Send, Collision

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated I/O		
PC28	BRG4O	Output	Baud-Rate Generator 4 Output
	CLK4/TIN1	Input	Clock 4
		Input	Timer Input 1
	Timer2:$\overline{\text{TOUT2}}$	Output	Timer 2: Timer Output 2
	SCC2:$\overline{\text{CTS}}$, CLSN	Input	SCC2: Clear to Send, Collision
PC27	BRG5O	Output	Baud-Rate Generator 5 Output
	CLK5	Input	Clock 5
	Timer3/4:$\overline{\text{TGATE2}}$	Input	Timer 3/4: Timer Gate 2
PC26	BRG6O	Output	Baud-Rate Generator 6 Output
	CLK6	Input	Clock 6
	Timer3:$\overline{\text{TOUT3}}$	Output	Timer 3: Timer Output 3
	TMCLK	Input	Timer Clock
PC25	BRG7O	Output	Baud-Rate Generator 7 Output
	CLK7/TIN4	Input	Clock 7
		Input	Timer Input 4
	DMA:$\overline{\text{DACK2}}$	Output	DMA: Data Acknowledge 2
PC24	BRG8O	Output	Baud-Rate Generator 8 Output
	CLK8/TIN3	Input	Clock 8
		Input	Timer Input 3
	Timer4:$\overline{\text{TOUT4}}$	Output	Timer 4: Timer Output 4
	DMA:DREQ2	Input	DMA: Data Request 2
PC23	CLK9	Input	Clock 9
	DMA:$\overline{\text{DACK1}}$	Output	DMA: Data Acknowledge 1
	EXT2	Input	External Request Line 2
PC22	SI1:L1ST1	Output	Serial Interface 1: Layer 1 Strobe 1
	CLK10	Input	Clock 10
	DMA:DREQ1	Inout	DMA: Request 1

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated I/O**		
PC15	SMC2:SMTXD	Output	SMC2: Serial Management Transmit Data
	SCC1:$\overline{\text{CTS}}$, CLSN	Input	SCC1: Clear To Send, Collision
	FCC1:MPHY master TxAddr0	Output	FCC1: Multiple PHY Master Transmit Address Bit 0
	FCC1:MPHY slave TxAddr0	Input	FCC1: Multiple PHY Slave Transmit Address Bit 0
PC14	SI1:L1ST2	Output	Serial Interface 1: Layer 1 Strobe 2
	SCC1:$\overline{\text{CD}}$, RENA	Input	SCC1: Carrier Detect, Receive Enable
	FCC1:MPHY master RxAddr0	Output	FCC1: Multiple PHY Master Receive Address Bit 0
	FCC1:MPHY slave RxAddr0	Input	FCC1: Multiple PHY Slave Receive Address Bit 0
PC13	SI1:L1ST4	Output	Serial Interface 1: Layer 1 Strobe 4
	SCC2:$\overline{\text{CTS}}$, CLSN	Input	SCC2: Clear to Send, Collision
	FCC1:MPHY master TxAddr1	Output	FCC1: Multiple PHY Master Transmit Address Bit 1
	FCC1:MPHY slave TxAddr1	Input	FCC1: Multiple PHY Slave Transmit Address Bit 1
PC12	SI1:L1ST3	Output	Serial Interface 1: Layer 1 Strobe 3
	SCC2:$\overline{\text{CD}}$, RENA	Input	SCC2: Carrier Detect, Request Enable
	FCC1:MPHY master RxAddr1	Output	FCC1: Multiple PHY Master Receive Address Bit 1
	FCC1:MPHY slave RxAddr1	Input	FCC1: Multiple PHY Slave Receive Address Bit 1
PC7	SI2:L1ST1	Output	Serial Interface 2: Layer 1 Strobe 1
	FCC1:$\overline{\text{CTS}}$	Input	FCC1: Clear To Send
	FCC1:MPHY master TxAddr2 multiplexed polling	Output	FCC1: Multiple PHY Master Transmit Address Bit 2 Multiplexed Polling
	FCC1:MPHY slave TxAddr2 multiplexed polling	Input	FCC1: Multiple PHY Slave Transmit Address Bit 2 Multiplexed Polling
	FCC1:MPHY master TxClav1 direct polling	Input	FCC1: Multiple PHY Master Transmit Cell Available 1 Direct Polling

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated I/O		
PC6	SI2:L1ST2	Output	Serial Interface 2: Layer 1 Strobe 2
	FCC1:$\overline{\text{CD}}$	Input	FCC1: Carrier Detect
	FCC1:MPHY master RxAddr2 multiplexed polling	Output	FCC1: Multiple PHY Master Receive Address Bit 2 Multiplexed Polling
	FCC1:MPHY slave RxAddr2 multiplexed polling	Input	FCC1: Multiple PHY Slave Receive Address Bit 2 Multiplexed Polling
	FCC1:MPHY master RxClav1 direct polling	Input	FCC1: Multiple PHY Master Receive Cell Available 1 Direct Polling
PC5	SMC1:SMTXD	Output	SMC1: Serial Management Transmit Data
	SI2:L1ST3	Output	Serial Interface 2: Layer 1 Strobe 3
	FCC2:$\overline{\text{CTS}}$	Input	FCC2: Clear To Send
PC4	SMC1:SMRXD	Input	SMC1: Serial Management Receive Data
	SI2:L1ST4	Output	Serial Interface 2: Layer 1 Strobe 4
	FCC2:$\overline{\text{CD}}$	Input	FCC2: Carrier Detect
PD31	SCC1:RxD	Input	SCC1: Receive Data
	DMA:$\overline{\text{DRACK1}}$	Output	DMA: Data Request Acknowledge 1
	DMA:$\overline{\text{DONE1}}$	Inout	DMA: Done 1
PD30	SCC1:TxD	Output	SCC1: Transmit Data
	DMA:$\overline{\text{DRACK2}}$	Output	DMA: Data Request Acknowledge 2
	DMA:$\overline{\text{DONE2}}$	Inout	DMA: Done 2
PD29	SCC1:$\overline{\text{RTS}}$, TENA	Output	SCC1: Request to Send, Transmit Enable
	FCC1:MPHY master RxAddr3 multiplexed polling	Output	FCC1: Multiple PHY Master Receive Address Bit 3 Multiplexed Polling
	FCC1:MPHY slave RxAddr3 multiplexed polling	Input	FCC1: Multiple PHY Slave Receive Address Bit 3 Multiplexed Polling
	FCC1:MPHY master RxClav2 direct polling	Input	FCC1: Multiple PHY Master Receive Cell Available 2 Direct Polling

Table 3-2. External Signals–CPM (Continued)

Name		Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated I/O**		
PD19	FCC1:MPHY master TxAddr4 multiplexed polling	Output	FCC1: Multiple PHY Master Transmit Address Bit 4 Multiplexed Polling
	FCC1:MPHY slave TxAddr4 multiplexed polling	Input	FCC1: Multiple PHY Slave Transmit Address Bit 4 Multiplexed Polling
	FCC1:MPHY master TxClav3 direct polling	Input	FCC1: Multiple PHY Master Transmit Cell Available 3 Direct Polling
	BRG1O	Output	Baud Rate Generator 1 Output
	SPI:$\overline{\text{SEL}}$	Input	SPI: Select
PD18	FCC1:MPHY master RxAddr4 multiplexed polling	Output	FCC1: Multiple PHY Master Receive Address Bit 4 Multiplexed Polling
	FCC1:MPHY slave RxAddr4 multiplexed polling	Input	FCC1: Multiple PHY Slave Receive Address Bit 4 Multiplexed Polling
	FCC1:MPHY master RxClav3 direct polling	Input	FCC1: Multiple PHY Master Receive Cell Available 3 Direct Polling
	SPI:CLK	Inout	SPI: Clock
PD17	BRG2O	Output	Baud Rate Generator 2 Output
	FCC1:UTOPIA 8 RxPrty	Input	FCC1: UTOPIA 8 Receive Parity
	SPI:MOSI	Inout	SPI: Master Output Slave Input
PD16	FCC1:UTOPIA 8 TxPrty	Output	FCC1: UTOPIA 8 Transmit Parity
	SPI:MISO	Inout	SPI: Master Input Slave Output
PD7	SMC1:$\overline{\text{SMSYN}}$	Input	SMC1: Serial Management Synchronization
	FCC1:MPHY master TxAddr3 multiplexed polling	Output	FCC1: Multiple PHY Master Transmit Address Bit 3 Multiplexed Polling
	FCC1:MPHY slave TxAddr3 multiplexed polling	Input	FCC1: Multiple PHY Slave Transmit Address Bit 3 Multiplexed Polling
	FCC1:MPHY master TxClav2 direct polling	Input	FCC1: Multiple PHY Master Transmit Cell Available 2 Direct Polling

3.3 Dedicated Pin Assignments by Port

Table 3-3. Port A—Dedicated Pin Assignment (PPARA = 1)

Pin	Pin Function					
	PSORA[*x*] = 0			PSORA[*x*] = 1		
	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input)	Default Input	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PA31	FCC1: $\overline{\text{TxEnb}}$ UTOPIA 8 master	FCC1: $\overline{\text{TxEnb}}$ UTOPIA 8 slave	GND		FCC1: COL MII	GND
PA30	FCC1: TxClav UTOPIA 8 slave	FCC1: TxClav UTOPIA 8 master FCC1: TxClav0 MPHY, master, direct polling	GND	FCC1: $\overline{\text{RTS}}$	FCC1: $\overline{\text{CRS}}$ MII	GND
PA29	FCC1: TxSOC UTOPIA 8			FCC1: TX_ER MII		
PA28	FCC1: $\overline{\text{RxEnb}}$ UTOPIA 8 master	FCC1: $\overline{\text{RxEnb}}$ UTOPIA 8 slave	GND	FCC1: TX_EN MII		
PA27		FCC1: RxSOC UTOPIA 8	GND		FCC1: RX_DV MII	GND
PA26	FCC1: RxClav UTOPIA 8 slave	FCC1: RxClav UTOPIA 8 master FCC1: RxClav0 MPHY, master, direct polling	GND		FCC1: RX_ER MII	GND
PA25	FCC1: TxD0 UTOPIA 8			MSNUM0[1]		
PA24	FCC1: TxD1 UTOPIA 8			MSNUM1[1]		
PA23	FCC1: TxD2 UTOPIA 8					
PA22	FCC1: TxD3 UTOPIA 8					
PA21	FCC1: TxD4 UTOPIA 8 FCC1: TxD3 MII/HDLC nibble					
PA20	FCC1: TxD5 UTOPIA 8 FCC1: TxD2 MII/HDLC nibble					

Table 3-3. Port A—Dedicated Pin Assignment (PPARA = 1) (Continued)

Pin	Pin Function					
	PSORA[*x*] = 0			PSORA[*x*] = 1		
	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input)	Default Input	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PA19	FCC1: TxD6 UTOPIA 8 FCC1: TxD1 MII/HDLC nibble					
PA18	FCC1: TxD7 UTOPIA 8 FCC1: TxD0 MII/HDLC nibble FCC1: TxD HDLC transparent					
PA17		FCC1: RxD7 UTOPIA 8 FCC1: RxD0 MII/HDLC nibble FCC1: RxD HDLC transparent	GND			
PA16		FCC1: RxD6 UTOPIA 8 FCC1: RxD1 MII/HDLC nibble	GND			
PA15		FCC1: RxD5 UTOPIA 8 FCC1: RxD2 MII/HDLC nibble	GND			
PA14		FCC1: RxD4 UTOPIA 8 FCC1: RxD3 MII/HDLC nibble	GND			
PA13		FCC1: RxD3 UTOPIA 8	GND	MSNUM2[1]		
PA12		FCC1: RxD2 UTOPIA 8	GND	MSNUM3[1]		
PA11		FCC1: RxD1 UTOPIA 8	GND	MSNUM4[1]		
PA10		FCC1: RxD0 UTOPIA 8	GND	MSNUM5[1]		
PA9	SMC2: SMTXD				TDM_A1: L1TXD0 Inout, nibble	GND

Table 3-3. Port A—Dedicated Pin Assignment (PPARA = 1) (Continued)

Pin	Pin Function					
	PSORA[*x*] = 0			PSORA[*x*] = 1		
	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input)	Default Input	PDIRA[*x*] = 1 (Output)	PDIRA[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PA8		SMC2: SMRXD	GND		TDM_A1: L1RXD0 Inout, nibble TDM_A1: L1RXD Inout, serial	GND
PA7		SMC2: $\overline{\text{SMSYN}}$	GND		TDM_A1: L1TSYNC/GRANT	GND
PA6					TDM_A1: L1RSYNC	GND

Notes:
1. MSNUM[0–4] is the sub-block code of the peripheral controller using SDMA; MSNUM[5] indicates which section, transmit or receive, is active during the transfer. See the SDMA Programming Model information in the *MSC8101 Reference Manual*.
2. *x* = The port signal number.

Table 3-4. Port B Dedicated Pin Assignment (PPARB = 1)

Pin	Pin Function					
	PSORB[*x*] = 0			PSORB[*x*] = 1		
	PDIRB[*x*] = 1 (Output)	PDIRB[*x*] = 0 (Input)	Default Input	PDIRB[*x*] = 1 (Output)	PDIRB[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PB31	FCC2: TX_ER MII	SCC2: RxD	GND		TDM_B2: L1TXD Inout	GND
PB30	SCC2: TxD	FCC2: RX_DV MII	GND		TDM_B2: L1RXD Inout	GND
PB29			GND	FCC2: TX_EN MII	TDM_B2: L1RSYNC	GND
PB28	FCC2: $\overline{\text{RTS}}$	FCC2: RX_ER MII	GND	SCC2: $\overline{\text{RTS}}$ SCC2: TENA Ethernet	TDM_B2: L1TSYNC/GRANT	GND
PB27		FCC2: COL MII	GND		TDM_C2: L1TXD Inout	GND
PB26		FCC2: $\overline{\text{CRS}}$ MII	GND		TDM_C2: L1RXD Inout	GND
PB25	FCC2: TxD3 MII/HDLC nibble			TDM_A1: L1TXD3 nibble	TDM_C2: L1TSYNC/GRANT	GND
PB24	FCC2: TxD2 MII/HDLC nibble	TDM_A1: L1RXD3 nibble	GND		TDM_C2: L1RSYNC	GND

Table 3-4. Port B Dedicated Pin Assignment (PPARB = 1) (Continued)

Pin	Pin Function					
	PSORB[*x*] = 0			PSORB[*x*] = 1		
	PDIRB[*x*] = 1 (Output)	PDIRB[*x*] = 0 (Input)	Default Input	PDIRB[*x*] = 1 (Output)	PDIRB[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PB23	FCC2: TxD1 MII/HDLC nibble	TDM_A1: L1RXD2 nibble	GND		TDM_D2: L1TXD Inout	GND
PB22	FCC2: TxD0 MII/HDLC nibble FCC2: TxD HDLC transparent	TDM_A1: L1RXD1 nibble	GND		TDM_D2: L1RXD Inout	GND
PB21		FCC2: RxD0 MII/HDLC nibble FCC2: RxD HDLC transparent	GND	TDM_A1: L1TXD2 nibble	TDM_D2: L1TSYNC/GRANT	GND
PB20		FCC2: RxD1 MII/HDLC nibble	GND	TDM_A1: L1TXD1 nibble	TDM_D2: L1RSYNC	GND
PB19		FCC2: RxD2 MII/HDLC nibble	GND		I2C: SDA Inout	VCC
PB18		FCC2: RxD3 MII/HDLC nibble	GND		I2C: SCL Inout	GND

Notes: 1. *x* = The port signal number.

Table 3-5. Port C Dedicated Pin Assignment (PPARC = 1)

PIN	Pin Function					
	PSORC[*x*] = 0			PSORC[*x*] = 1		
	PDIRC[*x*] = 1 (Output)	PDIRC[*x*] = 0 (Input)	Default Input	PDIRC[*x*] = 1 (Output)	PDIRC[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PC31	BRG1: BRGO	CLK1	CLK5		Timer1/2: $\overline{\text{TGATE1}}$	GND
PC30	BRG2: BRGO	CLK2	CLK6	Timer1: $\overline{\text{TOUT}}$	EXT1	
PC29	BRG3: BRGO	CLK3/TIN2	CLK7		SCC1: $\overline{\text{CTS}}$[1] SCC1: CLSN[1] Ethernet	GND
PC28	BRG4: BRGO	CLK4/TIN1	CLK8	Timer2: $\overline{\text{TOUT}}$	SCC2: $\overline{\text{CTS}}$[1] SCC2: CLSN[1] Ethernet	GND
PC27	BRG5: BRGO	CLK5	GND		Timer3/4: $\overline{\text{TGATE2}}$	GND

Table 3-5. Port C Dedicated Pin Assignment (PPARC = 1) (Continued)

PIN	Pin Function					
	PSORC[*x*] = 0			PSORC[*x*] = 1		
	PDIRC[*x*] = 1 (Output)	PDIRC[*x*] = 0 (Input)	Default Input	PDIRC[*x*] = 1 (Output)	PDIRC[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PC26	BRG6: BRGO	CLK6	GND	Timer3: $\overline{\text{TOUT}}$	TMCLK Real-time counter	BRGO1
PC25	BRG7: BRGO	CLK7/TIN4	GND	DMA2: $\overline{\text{DACK2}}$		
PC24	BRG8: BRGO	CLK8/TIN3	GND	Timer4: $\overline{\text{TOUT}}$	DMA: DREQ2	GND
PC23		CLK9	CLK3	DMA: $\overline{\text{DACK1}}$	EXT2	
PC22	SI1: L1ST1 Strobe	CLK10	CLK4		DMA: DREQ1	GND
PC15	SMC2: SMTXD	SCC1: $\overline{\text{CTS}}$ SCC1: CLSN Ethernet	by PC29	FCC1: TxAddr0 MPHY, master	FCC1: TxAddr0[1] MPHY, slave	GND
PC14	SI1: L1ST2 Strobe	SCC1: $\overline{\text{CD}}$ SCC1: RENA Ethernet	GND	FCC1: RxAddr0 MPHY, master	FCC1: RxAddr0[2] MPHY, slave	GND
PC13	SI1: L1ST4 Strobe	SCC2: $\overline{\text{CTS}}$ SCC2: CLSN Ethernet	by PC28	FCC1: TxAddr1 MPHY, master	FCC1: TxAddr1[2] MPHY, slave	GND
PC12	SI1: L1ST3 Strobe	SCC2: $\overline{\text{CD}}$ SCC2: RENA Ethernet	GND	FCC1: RxAddr1 MPHY, master	FCC1: RxAddr1[2] MPHY, slave	GND
PC7	SI2: L1ST1 Strobe	FCC1: $\overline{\text{CTS}}$	GND	FCC1: TxAddr2 MPHY, master, multiplexed polling	FCC1: TxAddr2[2] MPHY, slave, multiplexed polling FCC1: TxClav1[2] MPHY, master, direct polling	GND
PC6	SI2: L1ST2 Strobe	FCC1: $\overline{\text{CD}}$	GND	FCC1: RxAddr2 MPHY, master, multiplexed polling	FCC1: RxAddr2[2] MPHY, slave, multiplexed polling FCC1: RxClav1[2] MPHY, master, direct polling	GND
PC5	SMC1: SMTXD			SI2: L1ST3 Strobe	FCC2: $\overline{\text{CTS}}$	GND
PC4		SMC1: RXD	GND	SI2: L1ST4 Strobe	FCC2: $\overline{\text{CD}}$	GND

Notes:
1. Available only when the primary option for this function is not used.
2. MPHY Address pins 3,4 (master mode) can come from FCC2, depending on CMXUAR programming. See the CPM Multiplexing Programming model information in the *MSC8101 Reference Manual*.
3. *x* = The port signal number.

Table 3-6. Port D Dedicated Pin Assignment (PPARD = 1)

Pin	Pin Function					
	PSORD[*x*] = 0			PSORD = 1		
	PDIRD[*x*] = 1 (Output)	PDIRD[*x*] = 0 (Input)	Default Input	PDIRD[*x*] = 1 (Output)	PDIRD[*x*] = 0 (Input Unless Inout Is Specified)	Default Input
PD31		SCC1: RXD	GND	DMA: $\overline{\text{DRACK1}}$	DMA: $\overline{\text{DONE1}}$ Inout	
PD30	SCC1: TXD		GND	DMA: $\overline{\text{DRACK2}}$	DMA: $\overline{\text{DONE2}}$ Inout	
PD29	SCC1: $\overline{\text{RTS}}$ SCC1: TENA Ethernet			FCC1: RxAddr3[1] MPHY, master, multiplexed polling	FCC1: RxAddr3[2] MPHY, slave, multiplexed polling FCC1: RxClav2 [2] MPHY, master, direct polling	GND
PD19	FCC1: TxAddr4[1] MPHY, master, multiplexed polling	FCC1: TxAddr4[2] MPHY, slave, multiplexed polling FCC1: TxClav3[2] MPHY, master, direct polling	GND	BRG1: BRGO	SPI: $\overline{\text{SEL}}$	V_{DD}
PD18	FCC1: RxAddr4[1] MPHY, master, multiplexed polling	FCC1: RxAddr4[2] MPHY, slave, multiplexed polling FCC1: RxClav3[2] MPHY, master, direct polling	GND		SPI: CLK Inout	GND
PD17	BRG2: BRGO	FCC1: RxPrty UTOPIA 8	GND		SPI: MOSI Inout	V_{DD}
PD16	FCC1: TxPrty UTOPIA 8		GND		SPI: MISO Inout	MOSI
PD7		SMC1: $\overline{\text{SMSYN}}$	GND	FCC1: TxAddr3[1] MPHY, master, multiplexed polling	FCC1: TxAddr3[2] MPHY, slave, multiplexed polling FCC1: TxClav2[2] MPHY, master, direct polling	GND

Notes:
1. MPHY address pins 3 and 4 (master mode) can come from FCC2, depending on CMXUAR programming. See the CPM Multiplexing Programming model information in the *MSC8101 Reference Manual*.
2. MPHY address pins 0–4 (slave mode) can come from FCC2, depending on CMXUAR programming. See the CPM Multiplexing Programming model information in the *MSC8101 Reference Manual*.
3. *x* = The port signal number.

4 Reset

4.1 Reset Causes

Table 4-1. Reset Causes

Name	Direction	Description
Power-on reset (PORESET)	Input	PORESET initiates the power-on reset flow that resets all the MSC8101s and configures various attributes of the MSC8101, including its clock mode.
Hard reset (HRESET)	I/O	The MSC8101 can detect an external assertion of HRESET only if it occurs while the MSC8101 is not asserting reset. During HRESET, SRESET is asserted. HRESET is an open-drain pin.
Soft reset (SRESET)	I/O	The MSC8101 can detect an external assertion of SRESET only if it occurs while the MSC8101 is not asserting reset. SRESET is an open-drain pin.
Software watchdog reset		When the MSC8101 watchdog count reaches zero, a software watchdog reset is signalled. The enabled software watchdog event then generates an internal hard reset sequence.
Bus monitor reset		When the MSC8101 bus monitor count reaches zero, a bus monitor hard reset is asserted. The enabled bus monitor event then generates an internal hard reset sequence.
JTAG reset		When JTAG logic asserts the JTAG soft reset signal, an internal soft reset sequence is generated.

4.2 Reset Actions for Each Reset Source

Table 4-2. Reset Actions for Each Reset Source

Reset Source	Reset Logic PLL and DLL States	System Configuration Sampled	HRESET Driven	Other Internal Logic Reset	SRESET Driven	Core Reset
Power-on reset	Yes	Yes	Yes	Yes	Yes	Yes
External hard reset Software watchdog Bus monitor	No	Yes	Yes	Yes	Yes	Yes
JTAG reset External soft reset						

4.3 External Configuration Signals

Table 4-3. External Configuration Signals

Pin	Description	Settings
RSTCONF	**Reset Configuration** Input line sampled by the MSC8101 at the rising edge of PORESET.	0 Reset Configuration Master. 1 Reset Configuration Slave.
EE0	**EONCE Event Bit 0** Input line sampled after core PLL locks. Holding EE[0] at logic 1 at the exit from reset puts the SC140 core into Debug mode.	0 SC140 core starts the normal processing mode after reset. 1 SC140 core enters Debug mode immediately after reset.
HPE/EE1	**Host Port Enable** Input line sampled at the rising edge of PORESET. If asserted, the Host port is enabled, the PowerPC 60x data bus is 32-bit wide, and the Host *must* program the reset configuration word.	0 Host port disabled (hardware reset configuration enabled). 1 Host port enabled.
BTM[0-1]/ EE[4-5]	**Boot Mode** Input lines sampled at the rising edge of PORESET, which determine the MSC8101 Boot mode.	00 MSC8101 boots from external memory. 01 MSC8101 boots from HDI16. 10 Reserved. 11 Reserved.

4.4 Hard Reset Configuration Word

Table 4-4. Hard Reset Configuration Word

Reset: 0 Type: R/W

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	EARB	EXMC	IRQ7 INT	EBM	BPS		SCDIS	ISPS	IRPC		DPPC		NMI OUT	ISB		
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	BBD	MMR		—		TCPC		BC1PC		—	DLLDIS	MODCK_H			—

Table 4-4. Hard Reset Configuration Word (Continued)

Hard Reset Configuration Word Bit Descriptions

Bits	Name	Description	Settings
0	EARB	External Arbitration	0 = Internal arbitration 1 = External arbitration
1	EXMC	External MEMC	0 = None 1 = External memory controller
2	IRQ7 INT	$\overline{\text{IRQ7}}$ or $\overline{\text{INT\_OUT}}$ Selection	0 = Pin is $\overline{\text{IRQ7}}$ 1 = Pin is $\overline{\text{INT\_OUT}}$
3	EBM	External PowerPC Bus Mode	See the SIU programming model information in the *MSC8101 Reference Manual*.
4–5	BPS	Boot Port Size	00 = 64-bit port 01 = 8-bit port 10 = 16-bit port 11 = 32-bit port
6	SCDIS	SC140 Disabled	0 = Enabled 1 = Disabled
7	ISPS	Internal Space Port Size	See the Reset programming model information in the *MSC8101 Reference Manual*.
8–9	IRPC	Interrupt Pin Configuration	See the Reset programming model information in the *MSC8101 Reference Manual*.
10–11	DPPC	Data Parity Pin Configuration	See the Reset programming model information in the *MSC8101 Reference Manual*.
12	NMI OUT	Non-Maskable Interrupt Handling	0 = Serviced by SC140 core 1 = Routed to external pin and serviced by external host
13–15	ISB	Initial Internal Space Base Select	000 = 0xF000_0000 100 = Reserved. 001 = 0xF0F0_0000 101 = 0x00F0_0000 010 = 0xFF00_0000 110 = 0x0F00_0000 011 = 0xFFF0_0000 111 = 0x0FF0_0000
17	BBD	Bus Busy Disable	See the Reset programming model information in the *MSC8101 Reference Manual*.
18–19	MMR	Mask Master's Request	See the Reset programming model information in the *MSC8101 Reference Manual*.
22–23	TCPC	Transfer Code Pin Configuration	See the Reset programming model information in the *MSC8101 Reference Manual*.
24–25	BC1PC	BC1PC Value	See the Reset programming model information in the *MSC8101 Reference Manual*.
27	DLLDIS	DLL Disable	0 = No DLL bypass 1 = DLL bypass
28–30	MODCK_H	MODCK High-order Bits	See the Reset programming model information in the *MSC8101 Reference Manual*.

4.5 Host-Port Registers After Reset

Table 4-5. Core-Side Registers After Reset

Register Name	Register Data	Reset Type	
		HW Reset	IR Reset
HCR	All bits	0	—
HPCR	All bits	0	—
HSR	HF[0–3]	0	—
	HTFE	1	1
	HTFNF	1	1
	HRFNE	0	0
	HRFF	0	0
HCVR	HCP	0	0
	HV	0	0
HORX	All bits	empty	empty
HOTX	All bits	empty	empty

Notes: 1. A long dash (—) denotes bit value is indeterminate after reset.
2. "Empty" means that the data at this location is invalid (trash).

Table 4-6. Host-Side Registers After Reset

Register Name	Register Data	Reset Type	
		HW Reset	IR Reset
ICR	All bits	0	—
CVR	NMI	0	0
	HC	0	0
	HV[0–6]	0	—
ISR	HREQ	0	1 if TREQ is set; 0 otherwise
	HF[4–7]	0	—
	TRDY	1	1
	TXDE	1	1
	RXDF	0	0
RX	RX[0–3]	empty	empty
TX	TX[0–3]	empty	empty

Notes: 1. A long dash (—) denotes bit value is indeterminate after reset.
2. “Empty” means that the data at this location is invalid (trash).

5 Memory Maps

Table 5-1. SC140 Core Internal Memory Map

SC140 Core Internal Address	Mnemonic	Name	Size
SC140 Core Internal RAM			
00000000–0007FFFF	DSPRAM	Internal DSP RAM (Ports 1, 2, 3)	512 KB
00080000–00EFFDFF	Reserved	— Leave unchanged for future compatibility	14.5 MB
EOnCE registers			
00EFFE00–00EFFEFF	EOnCE	EOnCE registers	256 bytes

.

Table 5-2. QBus Memory Map—Bank0

SC140 Core Internal Address	Mnemonic	Name	Size
HDI16 - Port 1			
0000	HCR	Host Control Register	16 bits
0020	HPCR	Host Port Control Register	16 bits
0040	HSR	Host Status Register	16 bits
0060	HCVR	Host Command Vector Register	16 bits
0080	HOTX	Host Transmit Register	64 bits
00A0	HORX	Host Receive Register	64 bits
EFCOP—Port 1			
0C00	FDIR	EFCOP Data Input Register	32/64 bits
0C20	FDOR	EFCOP Data Output Register	32/64 bits
0C40	FKIR	EFCOP K-Constant Register	32 bits
0C60	FCNT	EFCOP Filter Count Register	16 bits
0C80	FCTL	EFCOP Control Register	16 bits
0CA0	FACR	EFCOP ALU Control Register	16 bits
0CC0	FDBA	EFCOP Data Base Address	16 bits
0CE0	FCBA	EFCOP Coefficient Base Address	16 bits
0D00	FDCH	EFCOP Decimation/Channel Count Register	16 bits
0D20	FSTR	EFCOP Status Register	16 bits
0D40–1BFF	Reserved	— Leave unchanged for future compatibility	3776 bytes

Table 5-2. QBus Memory Map—Bank0 (Continued)

SC140 Core Internal Address	Mnemonic	Name	Size
PIC			
1C00	ELIRA	Edge/Level-Triggered Interrupt Register A	16 bits
1C08	ELIRB	Edge/Level-Triggered Interrupt Register B	16 bits
1C10	ELIRC	Edge/Level-Triggered Interrupt Register C	16 bits
1C18	ELIRD	Edge/Level-Triggered Interrupt Register D	16 bits
1C20	ELIRE	Edge/Level-Triggered Interrupt Register E	16 bits
1C28	ELIRF	Edge/Level-Triggered Interrupt Register F	16 bits
1C30	IPRA	Interrupt Pending Register A	16 bits
1C38	IPRB	Interrupt Pending Register B	16 bits
1C48–FEFF	Reserved	— Leave unchanged for future compatibility	58.04 KB
QBus Banks			
FF00	QBUSMR0	QBus Mask Register 0	16 bits
FF02	QBUSBR0	QBus Base Address Register 0 Select DSP Peripheral	16 bits
FF04	QBUSMR1	QBus Mask Register 1	16 bits
FF06	QBUSBR1	QBus Base Address Register 1 Select DSP Peripheral	16 bits
FF08	QBUSMR2	QBus Mask Register 2	16 bits
FF0a	QBUSBR2	QBus Base Address Register 2 Select PowerPC 60x Bus	16 bits
FF0C–FFFF	Reserved	— Leave unchanged for future compatibility	244 bytes

Table 5-3. QBus Memory Map—Bank1

SC140 Core Internal Address	Mnemonic	Name	Size
Boot ROM			
0000–07FF	BOOTROM	MSC8101 Boot ROM	2 KB
0800–FFFF	Reserved	— Leave unchanged for future compatibility	63 KB

Table 5-4. PowerPC Local Bus Memory Map

PowerPC Local Bus Address	Mnemonic	Name	Size
SC140 Core Internal RAM - Port 4, Bank10			
00000–7FFFF	DSPRAM	Internal DSP RAM, Port 4	512 KB
HDI16 - Port 2, Bank11			
0080	HOTX	Host Transmit Register	64 bits
00A0	HORX	Host Receive Register	64 bits
EFCOP - Port 2, Bank11			
0C00	FDIR	EFCOP Data Input Register	32/64 bits
0C20	FDOR	EFCOP Data Output Register	32/64 bits
0C20–FFFF	Reserved	— Leave unchanged for future compatibility	3776 bytes

Table 5-5. PowerPC 60x Bus Memory Map

Internal Address	Mnemonic	Name	Size
CPM Dual-Port RAM			
00000–03FFF	DPRAM1	Dual-Port RAM	16 KB
04000–07FFF	Reserved	— Leave unchanged for future compatibility	16 KB
08000–08FFF	DPRAM2	Dual-Port RAM	4 KB
09000–0AFFF	Reserved	— Leave unchanged for future compatibility	8 KB
0B000–0BFFF	DPRAM3	Dual-Port RAM	4 KB
0C000–0FFFF	Reserved	— Leave unchanged for future compatibility	16 KB
General SIU			
10000	SIUMCR	SIU Module Configuration Register	32 bits
10004	SYPCR	System Protection Control Register	32 bits
10008	Reserved	— Leave unchanged for future compatibility	6 bytes
1000E	SWSR	Software Service Register	16 bits
10010–10023	Reserved	— Leave unchanged for future compatibility	20 bytes
10024	BCR	Bus Configuration Register	32 bits
10028	PPC_ACR	PowerPC 60x Bus Arbiter Configuration Register	8 bits
10029	Reserved	— Leave unchanged for future compatibility	24 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
1002C	PPC_ALRH	PowerPC 60x Bus Arbitration Level Register (bus masters 0–7)	32 bits
10030	PPC_ALRL	PowerPC 60x Bus Arbitration Level Register (bus masters 8–15)	32 bits
10034	LCL_ACR	Local Arbiter Configuration Register	8 bits
10035	Reserved	— Leave unchanged for future compatibility	24 bits
10038	LCL_ALRH	Local Arbitration Level Register (bus masters 0–7)	32 bits
1003C	LCL_ALRL	Local Arbitration Level Register (bus masters 8–15)	32 bits
10040	TESCR1	PowerPC 60x Bus Transfer Error Status Control Register 1	32 bits
10044	TESCR2	PowerPC 60x Bus Transfer Error Status Control Register 2	32 bits
10048	L_TESCR1	PowerPC Local Bus Transfer Error Status Control Register 1	32 bits
1004C	L_TESCR2	PowerPC Local Bus Transfer Error Status Control Register 2	32 bits
10050	PDTEA	PowerPC 60x bus SDMA Transfer Error Address	32 bits
10054	PDTEM	PowerPC 60x Bus SDMA Transfer Error MSNUM	8 bits
10055	Reserved	— Leave unchanged for future compatibility	24 bits
10058	LDTEA	PowerPC Local Bus SDMA Transfer Error Address	32 bits
1005C	LDTEM	PowerPC Local Bus SDMA Transfer Error MSNUM	8 bits
1005D–1005F	Reserved	— Leave unchanged for future compatibility	24 bits
10060	PDMTEA	PowerPC 60x Bus DMA Transfer Error Address	32 bits
10064	PDMTER	PowerPC 60x Bus DMA Transfer Error RQNUM	8 bits
10065	Reserved	— Leave unchanged for future compatibility	24 bits
10068	LDMTEA	PowerPC Local Bus DMA Transfer Error Address	32 bits
1006C	LDMTER	PowerPC Local Bus DMA Transfer Error RQNUM	8 bits
1006D–100FF	Reserved	— Leave unchanged for future compatibility	147 bytes
Memory Controller			
10100	BR0	Base Register Bank0	32 bits
10104	OR0	Option Register Bank0	32 bits
10108	BR1	Base Register Bank1	32 bits
1010C	OR1	Option Register Bank1	32 bits
10110	BR2	Base Register Bank2	32 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10114	OR2	Option Register Bank2	32 bits
10118	BR3	Base Register Bank3	32 bits
1011C	OR3	Option Register Bank3	32 bits
10120	BR4	Base Register Bank4	32 bits
10124	OR4	Option Register Bank4	32 bits
10128	BR5	Base Register Bank5	32 bits
1012C	OR5	Option Register Bank5	32 bits
10130	BR6	Base Register Bank6	32 bits
10134	OR6	Option Register Bank6	32 bits
10138	BR7	Base Register Bank7	32 bits
1013C	OR7	Option Register Bank7	32 bits
10140	Reserved	— Leave unchanged for future compatibility	32 bits
10144	Reserved	— Leave unchanged for future compatibility	32 bits
10148	Reserved	— Leave unchanged for future compatibility	32 bits
1014C	Reserved	— Leave unchanged for future compatibility	32 bits
10150	BR10	Base Register Bank10	32 bits
10154	OR10	Option Register Bank10	32 bits
10158	BR11	Base Register Bank11	32 bits
1015C	OR11	Option Register Bank11	32 bits
10160	Reserved	— Leave unchanged for future compatibility	8 bytes
10168	MAR	Memory Address Register	32 bits
1016C	Reserved	— Leave unchanged for future compatibility	32 bits
10170	MAMR	Machine A Mode Register	32 bits
10174	MBMR	Machine B Mode Register	32 bits
10178	MCMR	Machine C Mode Register	32 bits
1017C	Reserved	— Leave unchanged for future compatibility	8 bytes
10184	MPTPR	Memory Refresh Timer Prescaler	16 bits
10186	Reserved	— Leave unchanged for future compatibility	16 bits
10188	MDR	Memory Data Register	32 bits
1018C	Reserved	— Leave unchanged for future compatibility	32 bits
10190	PSDMR	PowerPC 60x Bus SDRAM Mode Register	32 bits
10194	Reserved	— Leave unchanged for future compatibility	32 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10198	PURT	PowerPC 60x Bus-Assigned UPM Refresh Timer	8 bits
10199	Reserved	— Leave unchanged for future compatibility	24 bits
1019C	PSRT	PowerPC 60x Bus-Assigned SDRAM Refresh Timer	8 bits
1019D	Reserved	— Leave unchanged for future compatibility	24 bits
101A0	Reserved	— Leave unchanged for future compatibility	32 bits
101A4	Reserved	— Leave unchanged for future compatibility	32 bits
101A8	IMMR	Internal Memory Map Register	32 bits
101AC–101FF	Reserved	— Leave unchanged for future compatibility	84 bytes
System Integration Timers			
10200–1021F	Reserved	— Leave unchanged for future compatibility	32 bytes
10220	TMCNTSC	Time Counter Status and Control Register	16 bits
10222	Reserved	— Leave unchanged for future compatibility	16 bits
10224	TMCNT	Time Counter Register	32 bits
10228	Reserved	— Leave unchanged for future compatibility	32 bits
1022C	TMCNTAL	Time Counter Alarm Register	32 bits
10230–1023F	Reserved	— Leave unchanged for future compatibility	16 bytes
10240	PISCR	Periodic Interrupt Status and Control Register	16 bits
10242	Reserved	— Leave unchanged for future compatibility	16 bits
10244	PITC	Periodic Interrupt Count Register	32 bits
10248	PITR	Periodic Interrupt Timer Register	32 bits
1024C–1029F	Reserved	— Leave unchanged for future compatibility	84 bytes
102AO–106FF	Reserved	— Leave unchanged for future compatibility	1120 bytes
DMA			
10700	DCHCR0	DMA Channel 0 Configuration Register	32 bits
10704	DCHCR1	DMA Channel 1 Configuration Register	32 bits
10708	DCHCR2	DMA Channel 2 Configuration Register	32 bits
1070C	DCHCR3	DMA Channel 3 Configuration Register	32 bits
10710	DCHCR4	DMA Channel 4 Configuration Register	32 bits
10714	DCHCR5	DMA Channel 5 Configuration Register	32 bits
10718	DCHCR6	DMA Channel 6 Configuration Register	32 bits
1071C	DCHCR7	DMA Channel 7 Configuration Register	32 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10720	DCHCR8	DMA Channel 8 Configuration Register	32 bits
10724	DCHCR9	DMA Channel 9 Configuration Register	32 bits
10728	DCHCR10	DMA Channel 10 Configuration Register	32 bits
1072C	DCHCR11	DMA Channel 11 Configuration Register	32 bits
10730	DCHCR12	DMA Channel 12 Configuration Register	32 bits
10734	DCHCR13	DMA Channel 13 Configuration Register	32 bits
10738	DCHCR14	DMA Channel 14 Configuration Register	32 bits
1073C	DCHCR15	DMA Channel 15 Configuration Register	32 bits
10740–1077F	Reserved	— Leave unchanged for future compatibility	64 bytes
10780	DIMR	DMA Internal Mask Register	32 bits
10784	DSTR	DMA Status Register	32 bits
10788	DTEAR	DMA TEA Status Register	8 bits
10789–1078B	Reserved	— Leave unchanged for future compatibility	24 bits
1078C	DPCR	DMA Pin Configuration Register	8 bits
1078D–1078F	Reserved	— Leave unchanged for future compatibility	24 bits
10790	DEMR	DMA External Mask Register	32 bits
10794–107FF	Reserved	— Leave unchanged for future compatibility	108 bytes
10800–10BFF	DCPRAM	DMA Channel Parameter RAM	1024 bytes
Interrupt Controller			
10C00	SICR	SIU Interrupt Configuration Register	16 bits
10C02	Reserved	— Leave unchanged for future compatibility	16 bits
10C04	SIVEC	SIU Interrupt Vector Register	32 bits
10C08	SIPNR_H	SIU Interrupt Pending Register (high)	32 bits
10C0C	SIPNR_L	SIU Interrupt Pending Register (low)	32 bits
10C10	SIPRR	SIU Interrupt Priority Register	32 bits
10C14	SCPRR_H	CPM Interrupt Priority Register (high)	32 bits
10C18	SCPRR_L	CPM Interrupt Priority Register (low)	32 bits
10C1C	SIMR_H	SIU Interrupt Mask Register (high)	32 bits
10C20	SIMR_L	SIU Interrupt Mask Register (low)	32 bits
10C24	SIEXR	SIU External Interrupt Control Register	32 bits
10C28–10C3F	Reserved	— Leave unchanged for future compatibility	24 bytes

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10C40	SICR_EXT	SIU Interrupt Configuration Register	16 bits
10C42	Reserved	— Leave unchanged for future compatibility	16 bits
10C44	SIVEC_EXT	SIU Interrupt Vector Register	32 bits
10C48	SIPNR_H_EXT	SIU Interrupt Pending Register (high)	32 bits
10C4C	SIPNR_L_EXT	SIU Interrupt Pending Register (low)	32 bits
10C50	SIPRR_EXT	SIU Interrupt Priority Register	32 bits
10C54	SCPRR_H_EXT	CPM Interrupt Priority Register (high)	32 bits
10C58	SCPRR_L_EXT	CPM Interrupt Priority Register (low)	32 bits
10C5C	SIMR_H_EXT	SIU Interrupt Mask Register (high)	32 bits
10C60	SIMR_L_EXT	SIU Interrupt Mask Register (low)	32 bits
10C64	SIEXR_EXT	SIU External Interrupt Control Register	32 bits
10C68–10C7F	Reserved	— Leave unchanged for future compatibility	24 bytes
Clocks and Reset			
10C80	SCCR	System Clock Control Register	32 bits
10C84	Reserved	— Leave unchanged for future compatibility	32 bits
10C88	SCMR	System Clock Mode Register	32 bits
10C8C	Reserved	— Leave unchanged for future compatibility	32 bits
10C90	RSR	Reset Status Register	32 bits
10C94	Reserved	— Leave unchanged for future compatibility	32 bits
10C98–10CFF	Reserved	— Leave unchanged for future compatibility	104 bytes
Input/Output Port			
10D00	PDIRA	Port A Data Direction Register	32 bits
10D04	PPARA	Port A Pin Assignment Register	32 bits
10D08	PSORA	Port A Special Options Register	32 bits
10D0C	PODRA	Port A Open-Drain Register	32 bits
10D10	PDATA	Port A Data Register	32 bits
10D14–10D1F	Reserved	— Leave unchanged for future compatibility	12 bytes
10D20	PDIRB	Port B Data Direction Register	32 bits
10D24	PPARB	Port B Pin Assignment Register	32 bits
10D28	PSORB	Port B Special Options Register	32 bits
10D2C	PODRB	Port B Open-Drain Register	32 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10D30	PDATB	Port B Data Register	32 bits
10D34–10D3F	Reserved	— Leave unchanged for future compatibility	12 bytes
10D40	PDIRC	Port C Data Direction Register	32 bits
10D44	PPARC	Port C Pin Assignment Register	32 bits
10D48	PSORC	Port C Special Options Register	32 bits
10D4C	PODRC	Port C Open-Drain Register	32 bits
10D50	PDATC	Port C Data Register	32 bits
10D54–10D5F	Reserved	— Leave unchanged for future compatibility	12 bytes
10D60	PDIRD	Port D Data Direction Register	32 bits
10D64	PPARD	Port D Pin Assignment Register	32 bits
10D68	PSORD	Port D special options register	32 bits
10D6C	PODRD	Port D Open Drain Register	32 bits
10D70	PDATD	Port D Data Register	32 bits
10D74–10D7F	Reserved	— Leave unchanged for future compatibility	12 bytes
CPM Timers			
10D80	TGCR1	Timer 1 and Timer 2 Global Configuration Register	8 bits
10D81	Reserved	— Leave unchanged for future compatibility	24 bits
10D84	TGCR2	Timer 3 and Timer 4 Global Configuration Register	8 bits
10D85–10D8F	Reserved	— Leave unchanged for future compatibility	11 bytes
10D90	TMR1	Timer 1 Mode Register	16 bits
10D92	TMR2	Timer 2 Mode Register	16 bits
10D94	TRR1	Timer 1 Reference Register	16 bits
10D96	TRR2	Timer 2 Reference Register	16 bits
10D98	TCR1	Timer 1 Capture Register	16 bits
10D9A	TCR2	Timer 2 Capture Register	16 bits
10D9C	TCN1	Timer 1 Counter	16 bits
10D9E	TCN2	Timer 2 Counter	16 bits
10DA0	TMR3	Timer 3 Mode Register	16 bits
10DA2	TMR4	Timer 4 Mode Register	16 bits
10DA4	TRR3	Timer 3 Reference Register	16 bits
10DA6	TRR4	Timer 4 Reference Register	16 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
10DA8	TCR3	Timer 3 Capture Register	16 bits
10DAA	TCR4	Timer 4 Capture Register	16 bits
10DAC	TCN3	Timer 3 Counter	16 bits
10DAE	TCN4	Timer 4 Counter	16 bits
10DB0	TER1	Timer 1 Event Register	16 bits
10DB2	TER2	Timer 2 Event Register	16 bits
10DB4	TER3	Timer 3 Event Register	16 bits
10DB6	TER4	Timer 4 Event Register	16 bits
10DB8	Reserved	— Leave unchanged for future compatibility	608 bytes
SDMA–General			
11018	SDSR	SDMA Status Register	8 bits
11019	Reserved	— Leave unchanged for future compatibility	24 bits
1101C	SDMR	SDMA Mask Register	8 bits
1101D	Reserved	— Leave unchanged for future compatibility	24 bits
11020–112FF	Reserved	Reserved	736 bytes
FCC1			
11300	GFMR1	FCC1 General Mode Register	32 bits
11304	FPSMR1	FCC1 Protocol-Specific Mode Register	32 bits
11308	FTODR1	FCC1 Transmit on Demand Register	16 bits
1130A	Reserved	— Leave unchanged for future compatibility	16 bits
1130C	FDSR1	FCC1 Data Synchronization Register	16 bits
1130E	Reserved	— Leave unchanged for future compatibility	16 bits
11310	FCCE1	FCC1 Event Register	32 bits
11314	FCCM1	FCC1 Mask Register	32 bits
11318	FCCS1	FCC1 Status Register	8 bits
11319–1131B	Reserved	— Leave unchanged for future compatibility	24 bits
1131C	FTIRR1_PHY0	FCC1 Transmit Internal Rate Registers for PHY[0–3]	8 bits
1131D	FTIRR1_PHY1		8 bits
1131E	FTIRR1_PHY2		8 bits
1131F	FTIRR1_PHY3		8 bits
FCC2			

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
11320	GFMR2	FCC2 General Mode Register	32 bits
11324	FPSMR2	FCC2 Protocol-Specific Mode Register	32 bits
11328	FTODR2	FCC2 Transmit On-Demand Register	16 bits
1132A	Reserved	— Leave unchanged for future compatibility	16 bits
1132C	FDSR2	FCC2 Data Synchronization Register	16 bits
1132E	Reserved	— Leave unchanged for future compatibility	16 bits
11330	FCCE2	FCC2 Event Register	32 bits
11334	FCCM2	FCC2 Mask Register	32 bits
11338	FCCS2	FCC2 Status Register	8 bits
11339	Reserved	— Leave unchanged for future compatibility	24 bits
1133C	FTIRR2_PHY0	FCC2 Transmit Internal Rate Registers for PHY[0–3]	8 bits
1133D	FTIRR2_PHY1		8 bits
1133E	FTIRR2_PHY2		8 bits
1133F	FTIRR2_PHY3		8 bits
FCC3			
11340	GFMR3	FCC3 General Mode Register	32 bits
11344	FPSMR3	FCC3 Protocol-Specific Mode Register	32 bits
11348	FTODR3	FCC3 Transmit On-Demand Register	16 bits
1134A	Reserved	— Leave unchanged for future compatibility	16 bits
1134C	FDSR3	FCC3 Data Synchronization Register	16 bits
1134E	Reserved	— Leave unchanged for future compatibility	16 bits
11350	FCCE3	FCC3 Event Register	32 bits
11354	FCCM3	FCC3 Mask Register	32 bits
11358	FCCS3	FCC3 Status Register	8 bits
11359–115EF	Reserved	— Leave unchanged for future compatibility	663 bytes
BRGs[5–8]			
115F0	BRGC5	BRG5 Configuration Register	32 bits
115F4	BRGC6	BRG6 Configuration Register	32 bits
115F8	BRGC7	BRG7 Configuration Register	32 bits
115FC	BRGC8	BRG8 Configuration Register	32 bits
11600–1185F	Reserved	— Leave unchanged for future compatibility	608 bytes

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
I^2C			
11860	I2MOD	I^2C Mode Register	8 bits
11862	Reserved	— Leave unchanged for future compatibility	24 bits
11864	I2ADD	I^2C Address Register	8 bits
11866	Reserved	— Leave unchanged for future compatibility	24 bits
11868	I2BRG	I^2C BRG Register	8 bits
1186A	Reserved	— Leave unchanged for future compatibility	24 bits
1186C	I2COM	I^2C Command Register	8 bits
1186E	Reserved	— Leave unchanged for future compatibility	24 bits
11870	I2CER	I^2C Event Register	8 bits
11872	Reserved	— Leave unchanged for future compatibility	24 bits
11874	I2CMR	I^2C Mask Register	8 bits
11875–119BF	Reserved	— Leave unchanged for future compatibility	331 bytes
Communications Processor			
119C0	CPCR	Communications Processor Command Register	32 bits
119C4	RCCR	CP Configuration Register	32 bits
119C8	Reserved	— Leave unchanged for future compatibility	14 bytes
119D6	RTER	CP Timers Event Register	16 bits
119D8	Reserved	— Leave unchanged for future compatibility	16 bits
119DA	RTMR	CP Timers Mask Register	16 bits
119DC	RTSCR	CP Time-Stamp Timer Control Register	16 bits
119DE	Reserved	— Leave unchanged for future compatibility	16 bits
119E0	RTSR	CP Time-Stamp Register	32 bits
119E4	Reserved	— Leave unchanged for future compatibility	12 bytes
BRGs[1–4]			
119F0	BRGC1	BRG1 Configuration Register	32 bits
119F4	BRGC2	BRG2 Configuration Register	32 bits
119F8	BRGC3	BRG3 Configuration Register	32 bits
119FC	BRGC4	BRG4 Configuration Register	32 bits
SCC1			
11A00	GSMR_L1	SCC1 General Mode Register (low)	32 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
11A04	GSMR_H1	SCC1 General Mode Register (high)	32 bits
11A08	PSMR1	SCC1 Protocol-Specific Mode Register	16 bits
11A0A	Reserved	— Leave unchanged for future compatibility	16 bits
11A0C	TODR1	SCC1 Transmit-on-Demand Register	16 bits
11A0E	DSR1	SCC1 Data Synchronization Register	16 bits
11A10	SCCE1	SCC1 Event Register	16 bits
11A12	Reserved	— Leave unchanged for future compatibility	16 bits
11A14	SCCM1	SCC1 Mask Register	16 bits
11A16	Reserved	— Leave unchanged for future compatibility	8 bits
11A17	SCCS1	SCC1 Status Register	8 bits
11A18–11A1F	Reserved	— Leave unchanged for future compatibility	8 bytes
SCC2			
11A20	GSMR_L2	SCC2 General Mode Register (low)	32 bits
11A24	GSMR_H2	SCC2 General Mode Register (high)	32 bits
11A28	PSMR2	SCC2 Protocol-Specific Mode Register	16 bits
11A2A	Reserved	— Leave unchanged for future compatibility	16 bits
11A2C	TODR2	SCC2 Transmit-on-Demand Register	16 bits
11A2E	DSR2	SCC2 Data Synchronization Register	16 bits
11A30	SCCE2	SCC2 Event Register	16 bits
11A32	Reserved	— Leave unchanged for future compatibility	16 bits
11A34	SCCM2	SCC2 Mask Register	16 bits
11A36	Reserved	— Leave unchanged for future compatibility	8 bits
11A37	SCCS2	SCC2 Status Register	8 bits
11A38–11A3F	Reserved	— Leave unchanged for future compatibility	8 bytes
SCC3			
11A40	GSMR_L3	SCC3 General Mode Register (low)	32 bits
11A44	GSMR_H3	SCC3 General Mode Register (high)	32 bits
11A48	PSMR3	SCC3 Protocol-Specific Mode Register	16 bits
11A4A	Reserved	— Leave unchanged for future compatibility	16 bits
11A4C	TODR3	SCC3 Transmit-on-Demand Register	16 bits
11A4E	DSR3	SCC3 Data Synchronization Register	16 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
11A50	SCCE3	SCC3 Event Register	16 bits
11A52	Reserved	— Leave unchanged for future compatibility	16 bits
11A54	SCCM3	SCC3 Mask Register	16 bits
11A56	Reserved	— Leave unchanged for future compatibility	8 bits
11A57	SCCS3	SCC3 Status Register	8 bits
11A58–11A5F	Reserved	— Leave unchanged for future compatibility	8 bytes
SCC4			
11A60	GSMR_L4	SCC4 General Mode Register (low)	32 bits
11A64	GSMR_H4	SCC4 General Mode Register (high)	32 bits
11A68	PSMR4	SCC4 Protocol-Specific Mode Register	16 bits
11A6A	Reserved	— Leave unchanged for future compatibility	16 bits
11A6C	TODR4	SCC4 Transmit-on-Demand Register	16 bits
11A6E	DSR4	SCC4 Data Synchronization Register	16 bits
11A70	SCCE4	SCC4 Event Register	16 bits
11A72	Reserved	— Leave unchanged for future compatibility	16 bits
11A74	SCCM4	SCC4 Mask Register	16 bits
11A76	Reserved	— Leave unchanged for future compatibility	8 bits
11A77	SCCS4	SCC4 Status Register	8 bits
11A40–11A7F	Reserved	— Leave unchanged for future compatibility	8 bytes
SMC1			
11A80	Reserved	— Leave unchanged for future compatibility	16 bits
11A82	SMCMR1	SMC1 Mode Register	16 bits
11A84	Reserved	— Leave unchanged for future compatibility	16 bits
11A86	SMCE1	SMC1 Event Register	8 bits
11A87	Reserved	— Leave unchanged for future compatibility	24 bits
11A8A	SMCM1	SMC1 Mask Register	8 bits
11A8B–11A8F	Reserved	— Leave unchanged for future compatibility	5 bytes
SMC2			
11A90	Reserved	— Leave unchanged for future compatibility	16 bits
11A92	SMCMR2	SMC2 Mode Register	16 bits
11A94	Reserved	— Leave unchanged for future compatibility	16 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
11A96	SMCE2	SMC2 Event Register	8 bits
11A97	Reserved	— Leave unchanged for future compatibility	24 bits
11A9A	SMCM2	SMC2 Mask Register	8 bits
11A9B–11A9F	Reserved	— Leave unchanged for future compatibility	5 bytes
SPI			
11AA0	SPMODE	SPI Mode Register	16 bits
11AA2	Reserved	— Leave unchanged for future compatibility	32 bits
11AA6	SPIE	SPI Event Register	8 bits
11AA7	Reserved	— Leave unchanged for future compatibility	24 bits
11AAA	SPIM	SPI Mask Register	8 bits
11AAB	Reserved	— Leave unchanged for future compatibility	16 bits
11AAD	SPCOM	SPI Command Register	8 bits
11AAE–11AFF	Reserved	— Leave unchanged for future compatibility	82 bytes
CPM Mux			
11B00	CMXSI1CR	CPM MUX SI1 Clock Route Register	8 bits
11B01	Reserved	— Leave unchanged for future compatibility	8 bits
11B02	CMXSI2CR	CPM MUX SI2 Clock Route Register	8 bits
11B03	Reserved	— Leave unchanged for future compatibility	8 bits
11B04	CMXFCR	CPM MUX FCC Clock Route Register	32 bits
11B08	CMXSCR	CPM MUX SCC Clock Route Register	32 bits
11B0C	CMXSMR	CPM MUX SMC Clock Route Register	8 bits
11B0D	Reserved	— Leave unchanged for future compatibility	8 bits
11B0E	CMXUAR	CPM MUX UTOPIA Address Register	16 bits
11B10–11B1F	Reserved	— Leave unchanged for future compatibility	16 bytes
SI1 Registers			
11B20	SI1AMR	SI1 TDMA1 Mode Register	16 bits
11B22	Reserved	— Leave unchanged for future compatibility	16 bits
11B24	Reserved	— Leave unchanged for future compatibility	16 bits
11B26	Reserved	— Leave unchanged for future compatibility	16 bits
11B28	SI1GMR	SI1 Global Mode Register	8 bits
11B29	Reserved	— Leave unchanged for future compatibility	8 bits

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
11B2A	SI1CMDR	SI1 Command Register	8 bits
11B2B	Reserved	— Leave unchanged for future compatibility	8 bits
11B2C	SI1STR	SI1 Status Register	8 bits
11B2D	Reserved	— Leave unchanged for future compatibility	8 bits
11B2E	SI1RSR	SI1 RAM Shadow Address Register	16 bits
MCC1 Registers			
11B30	MCCE1	MCC1 Event Register	16 bits
11B32	Reserved	— Leave unchanged for future compatibility	16 bits
11B34	MCCM1	MCC1 Mask Register	16 bits
11B36	Reserved	— Leave unchanged for future compatibility	16 bits
11B38	MCCF1	MCC1 Configuration Register	8 bits
11B39–11B3F	Reserved	— Leave unchanged for future compatibility	7 bytes
SI2 Registers			
11B40	Reserved	— Leave unchanged for future compatibility	16 bits
11B42	SI2BMR	SI2 TDMB2 Mode Register	16 bits
11B44	SI2CMR	SI2 TDMC2 Mode Register	16 bits
11B46	SI2DMR	SI2 TDMD2 Mode Register	16 bits
11B48	SI2GMR	SI2 Global Mode Register	8 bits
11B49	Reserved	— Leave unchanged for future compatibility	8 bits
11B4A	SI2CMDR	SI2 Command Register	8 bits
11B4B	Reserved	— Leave unchanged for future compatibility	8 bits
11B4C	SI2STR	SI2 Status Register	8 bits
11B4D	Reserved	— Leave unchanged for future compatibility	8 bits
11B4E	SI2RSR	SI2 RAM Shadow Address Register	16 bits
MCC2 Registers			
11B50	MCCE2	MCC2 Event Register	16 bits
11B52	Reserved	— Leave unchanged for future compatibility	16 bits
11B54	MCCM2	MCC2 Mask Register	16 bits
11B56	Reserved	— Leave unchanged for future compatibility	16 bits
11B58	MCCF2	MCC2 Configuration Register	8 bits
11B59–11FFF	Reserved	— Leave unchanged for future compatibility	1191 bytes

Table 5-5. PowerPC 60x Bus Memory Map (Continued)

Internal Address	Mnemonic	Name	Size
SI1 RAM			
12000–121FF	SI1TxRAM	SI1 Transmit Touting RAM	512 bytes
12200–123FF	Reserved	— Leave unchanged for future compatibility	512 bytes
12400–125FF	SI1RxRAM	SI1 Receive Touting RAM	512 bytes
12600–127FF	Reserved	— Leave unchanged for future compatibility	512 bytes
SI2 RAM			
12800–129FF	SI2TxRAM	SI2 Transmit Routing RAM	512 bytes
12A00–12BFF	Reserved	— Leave unchanged for future compatibility	512 bytes
12C00–12DFF	SI2RxRAM	SI2 Receive Routing RAM	512 bytes
12E00–12FFF	Reserved	— Leave unchanged for future compatibility	512 bytes
13000–137FF	Reserved	— Leave unchanged for future compatibility	2048 bytes
13800–13FFF	Reserved	— Leave unchanged for future compatibility	2048 bytes

6 Registers

Table 6-1. Register Description Conventions

Convention	Meaning	Convention	Meaning
R	Read-only bit. Writing this bit has no effect.	—	Reserved bit. Write to zero for future compatibility
W	Write-only bit.	0	Bit resets to a logic 0
R/W	Standard read/write bit.	1	Bit resets to a logic 1

6.1 Core Registers

Table 6-2. Core Registers Summary

Mnemonic	Register Name	Description
R[0–15]	Address	32-bit, R/W. Contain addresses or general-purpose data
NSP, ESP	Stack pointer	32-bit. Used implicitly in all PUSH and POP instructions: NSP in Normal mode, ESP in Exception mode
	Shadow stack pointer	Contain decremented values of the stack pointers
B[0–7]	Base address	32-bit, R/W. Used in modulo calculations and associated with R registers (B0 with R0, and so on)
N[0–3]	Offset	32-bit, R/W. Contain offset values to increment or decrement address registers. Also used for 32-bit general-purpose storage
M[0–3]	Modifier	32-bit, R/W. Contain the value of the modulus modifier. Also used for general-purpose storage
MCTL	Modifier control	32-bit, R/W. Programs the address mode for each address, R[0–7]
D[0–15]	Data	40-bit. Used to perform arithmetic and logical operations on data operands
PC	Program counter	
SR	Status	32-bit
EMR	Exception and mode	32-bit. Reflects and controls exception situations in the core
SA[0–3]	Start address	
LC[0–3]	Loop counter	

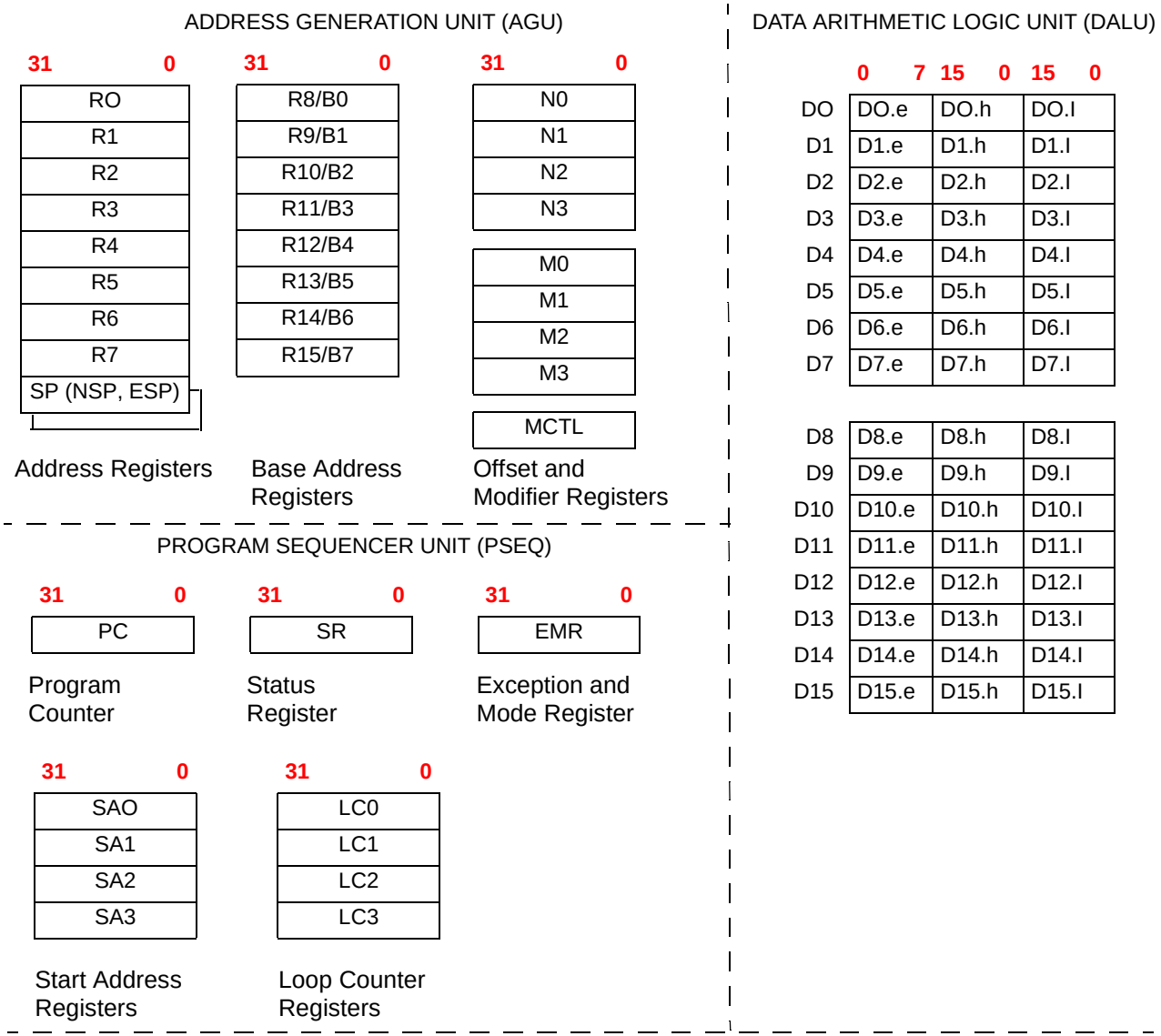


Figure 6-1. SC140 Programming Model

Table 6-3. Core Registers

MCTL **Modifier Control Register** Reset: 0 Type: R/W 0x10C92

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R7 AM[3–0]				R6 AM[3–0]				R5 AM[3–0]				R4 AM[3–0]			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R3 AM[3–0]				R2 AM[3–0]				R1 AM[3–0]				R0 AM[3–0]			

Address Modifier (AM[3–0]) Bit Descriptions

AM Mode Descriptions[1]	AM3	AM2	AM1	AM0
Linear addressing	0	0	0	0
Reverse-carry addressing	0	0	0	1
M0 used—Modulo addressing	1	0	0	0
M1 used—Modulo addressing	1	0	0	1
M2 used—Modulo addressing	1	0	1	0
M3 used—Modulo addressing	1	0	1	1
M0 used—Multiple wrap-around modulo addressing	1	1	0	0
M1 used—Multiple wrap-around modulo addressing	1	1	0	1
M2 used—Multiple wrap-around modulo addressing	1	1	1	0
M3 used—Multiple wrap-around modulo addressing	1	1	1	1

Notes: 1. All other AM field combinations are reserved and should not be used.

SR **Status Register** Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SLF	LF3	LF2	LF1	LF0	—	—	—	I2	I1	I0	OVE	DI	EXP	—	—
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VF3	VF2	VF1	VF0	—	S	S1	S0	RM	SM	T	C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-3. Core Registers (Continued)

SR Bit Descriptions

Bits	Name	Description	Settings
31	SLF	Short Loop Flag	0 = Active loop length is three or more execution sets 1 = Active loop length is one or two execution sets
30	LF3	Loop Flag 3	0 = Hardware loop #4 not enabled 1 = Hardware loop #4 enabled
29	LF2	Loop Flag 2	0 = Hardware loop #3 not enabled 1 = Hardware loop #3 enabled
28	LF1	Loop Flag 1	0 = Hardware loop #2 not enabled 1 = Hardware loop #2 enabled
27	LF0	Loop Flag 0	0 = Hardware loop #1 not enabled 1 = Hardware loop #1 enabled
23–21	I[2–0]	Interrupt Mask	See table below
20	OVE	Overflow Exception Enable	0 = Overflow exception generation is disabled 1 = Overflow exception generation is enabled, unless EMR[DOVF] bit = 1
19	DI	Disable Interrupts	0 = Interrupts enabled 1 = Interrupts disabled
18	EXP	Exception Mode	0 = Normal processing mode, active stack pointer is NSP 1 = Exception processing mode, active stack pointer is ESP
11–8	VF[3–0]	Viterbi Flags	0 = Appropriate 16-bit portion transferred 1 = Appropriate 16-bit portion not transferred

I2	I1	I0	Exceptions Permitted	Exceptions Masked
0	0	0	IPL[1–7]	IPL0
0	0	1	IPL[2–7]	IPL[0–1]
0	1	0	IPL[3–7]	IPL[0–2]
0	1	1	IPL[4–7]	IPL[0–3]
1	0	0	IPL[5–7]	IPL[0–4]
1	0	1	IPL[6–7]	IPL[0–5]
1	1	0	IPL7	IPL[0–6]
1	1	1	NMI	IPL[0–7]

Notes: 1. An IPL0 exception is always masked.

Table 6-3. Core Registers (Continued)

<table>
<tr><td>6</td><td>S</td><td>Scaling</td><td>
<table>
<tr><th>S1</th><th>S0</th><th>Scaling Mode</th><th>S Equation</th></tr>
<tr><td>0</td><td>0</td><td>No scaling</td><td>S = (D30 XOR D29) OR S (previous)</td></tr>
<tr><td>0</td><td>1</td><td>Scale down</td><td>S = (D31 XOR D30) OR S (previous)</td></tr>
<tr><td>1</td><td>0</td><td>Scale up</td><td>S = (D29 XOR D28) OR S (previous)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td><td>S = Undefined</td></tr>
</table>
</td></tr>
<tr><td>5–4</td><td>S[1–0]</td><td>Scaling Mode</td><td>
<table>
<tr><th>S1</th><th>S0</th><th>Scaling Mode</th><th>Rounding Bit</th></tr>
<tr><td>0</td><td>0</td><td>No scaling</td><td>15</td></tr>
<tr><td>0</td><td>1</td><td>Scale down
(1-bit Arithmetic right shift)</td><td>16</td></tr>
<tr><td>1</td><td>0</td><td>Scale up
(1-bit Arithmetic left shift)</td><td>14</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>RM</td><td>Rounding Mode</td><td>0 = Convergent rounding selected
1 = Two’s complement rounding selected</td></tr>
<tr><td>2</td><td>SM</td><td>Arithmetic Saturation Mode</td><td>0 = Mode not selected 1 = Mode selected</td></tr>
<tr><td>1</td><td>T</td><td>True</td><td>0 = Condition tested by compare or test instruction is false
1 = Condition tested by compare or test instruction is true</td></tr>
<tr><td>0</td><td>C</td><td>Carry</td><td>0 = No carry or borrow generated
1 = Carry generated from last addition, or borrow generated from last subtraction</td></tr>
</table>

Table 6-3. Core Registers (Continued)

EMR **Exception and Mode Register** Type: R/W 0x10C92

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—								GP6	GP5	GP4	GP3	GP2	GP1	GP0	BEM
Type	R															
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—												NMID	DOVF	ILST	ILIN
Type	R												R/W			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EMR Bit Descriptions

Bits	Name	Description	Settings
23–17	GP[6-0]	General Purpose Flags	
16	BEM	Big Endian Memory	0 = Little endian configuration 1 = Big endian configuration
3	NMID	Non-Maskable Interrupt (NMI) Disable	0 = No NMI service executing 1 = NMI Service executing
2	DOVF	DALU Overflow	0 = No overflow or arithmetic saturation occurred 1 = Overflow or arithmetic saturation occurred
1	ILST	Illegal Execution Set	0 = No execution set rule violated 1 = Execution set rule violated
0	ILIN	Illegal Instruction	0 = No instruction set violation 1 = One or more opcodes received are not part of the SC140 instruction set

6.2 Extended Core Registers

Table 6-4. QBus Registers

Bank Registers					Examples of Bank Address and Mask Register Values			
Name	**Description**	**DSP Internal Address**	**Bank Slaves**	**Reset Value**	**Base Reg.**	**Mask Reg.**	**Bank Size**	**Address Range for a Match**
QBUSMR0	DSP Mask0 (option) Register	0x{Base0,FF00}	Bank registers DSP Peripherals: PIC Host Interface EFCOP	0xFFFF	0x001F	0xFFFF	64 KB	0x001F0000– 0x001FFFFF
QBUSBR0	DSP Base0 Address Register	0x{Base0,FF02}		0x00F0	0x001C	0xFFFC	256 KB	0x001C0000– 0x001FFFFF
QBUSMR1	DSP Mask1 (option) Register	0x{Base0,FF04}	Boot ROM	0xFFFF	0x001F	0xFFFF		Null (no possible match)
QBUSBR1	DSP Base1 Address Register	0x{Base0,FF06}		0x00F8				
QBUSMR2	DSP Mask2 (option) Register	0x{Base0,FF08}	Cacheable memory on the PowerPC 60x bus	0xFFFF				
QBUSBR2	DSP Base2 Address Register	0x{Base0,FF0A}		0x0000				

Table 6-5. HDI16 Registers

HCR **Host Control Register (HICR = 0)** Reset: Depends on reset configuration sequence Type: R/W 0x0000

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	HF4	HF5	HF6	HF7	HICR	HDM0	HDM1	HDM2	—	DBTE	DBRE	HCIE	HTFIE	HTEIE	HRFIE	HREIE

HCR **Host Control Register (HICR = 1)** Reset: Depends on reset configuration sequence 0x0000

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	HF4	HF5	HF6	HF7	HICR	RREQ	HM		—	DBTE	DBRE	HCIE	HTFIE	HTEIE	HRFIE	HREIE
Type	R/W					R	R		R/W							

Table 6-5. HDI16 Registers (Continued)

HCR Bit Descriptions

Bits	Name	Description	Settings
0–3	HF[4–7]	Host Flags	Values reflected in the ISR
4	HICR	ICR/HCR priority for DMA/Last Address Mode	0 = DMA/last address mode defined in HCR; 1 = Defined in ICR
5–7	HDM[0–2]	Host DMA/Last Address Mode Control	See information on host DMA mode control values in the *MSC8101 Reference Manual.*
5	RREQ (HICR = 1)	RREQ Status	
6–7	HM (HICR = 1)	ICR[HM] Status	
9	DBTE	DMA Transmit Burst Enable	0 = Disabled 1 = Enabled
10	DBRE	DMA Receive Burst Enable	0 = Disabled 1 = Enabled
11	HCIE	Host Command Interrupt Enable	0 = Disabled 1 = Enabled
12	HTFIE	Host Transmit Not Full Interrupt Enable	0 = Disabled 1 = Enabled
13	HTEIE	Host Transmit Empty Interrupt Enable	0 = Disabled 1 = Enabled
14	HRFIE	Host Receive Full Interrupt Enable	0 = Disabled 1 = Enabled
15	HREIE	Host Receive Not Empty Interrupt Enable	0 = Disabled 1 = Enabled

HSR **Host Status Register** Reset: Depends on reset configuration sequence Type: R 0x0040

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	HF0	HF1	HF2	HF3	—								HTFNF	HTFE	HRFF	HRFNE

HSR Bit Descriptions

Bits	Name	Description	Settings
0–3	HF[0–3]	Host Flags	Values reflect ICR(HF[0–3])
12	HTFNF	Host Transmit FIFO Not Full	0 = Full 1 = Not full
13	HTFE	Host Transmit FIFO Empty	0 = Not empty 1 = Empty
14	HRFF	Host Receive FIFO Full	0 = Not full 1 = Full
15	HRFNE	Host Receive FIFO Not Empty	0 = Empty 1 = Not empty

Table 6-5. HDI16 Registers (Continued)

HCVR **Host Command Vector Register** Reset: 0 Type: R 0x0060

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—								HCP	HV						

HCVR Bit Descriptions

Bits	Name	Description	Settings
8	HCP	Host Command Pending	Value reflects CVR[HC]
9–12	HV	Host Vector	Value reflects CVR[HV]

HPCR **Host Port Control Register** Reset: Depends on reset configuration sequence Type: R/W 0x0020

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	HAP	HRP	HCSP	HDDS	—		HDSP	—	HEN	H8BIT	—				DMA	OAD

HPCR Bit Descriptions

Bits	Name	Description	Settings
0	HAP	Host Acknowledge Polarity	0 = HACK as active low input 1 = HACK as active high input
1	HRP	Host Request Polarity	Single host request mode: 0 = Request signal is active low 1 = Request signal is active high Double request mode: 0 = HTRQ, HRRQ are active low 1 = HDRQ, HRRQ are active high
2	HCSP	Host Chip-Select Polarity	0 = HCS1 or HCS2 is active low 1 = HCS1 or HCS2 is active high
3	HDDS	Host Dual Data Strobe	0 = HDI16 operates in single-strobe bus mode 1 = HDI16 operates in dual-strobe bus mode
6	HDSP	Host Data Strobe Polarity	0 = Data strobe is active low 1 = Data strobe is active high
8	HEN	Host Enable	0 = HDI16 internal clock frozen 1 = HDI16 operates as the host interface
9	H8BIT	H8-Bit Mode	0 = 16-bit mode (default) enabled 1 = 8-bit mode enabled
14	DMA	Host DMA Mode Enable	0 = Disabled 1 = Enabled
15	OAD	One-Address Host DMA Mode Enable	0 = HACK input pin used as a DMA transfer acknowledge input 1 = Host address 0x4 used as a host DMA transfer acknowledge input

Table 6-5. HDI16 Registers (Continued)

HOTX **Host Transmit Data Register** Type: W 0x0080

A four 64-bit word FIFO. If HOTX is empty, writing it clears HSR[HTFE]. If HOTX contains three 64-bit words, writing it clears HTFNF.

HORX **Host Receive Data Register** Type: R 0x00A0

A four 64-bit word FIFO. If HORX is full, reading it clears HSR[HRFF]. If HORX contains one 64-bit word, reading it clears HSR[HRFNE].

ICR **Host Interface Control Register (DMA = 0, HICR = 1)** Reset:: Depends on reset configuration sequence Type: R/W 0x0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—					HF0	HF1	—	INIT	HM		HF2	HF3	HDRQ	TREQ	RREQ

ICR **Host Interface Control Register (DMA = 1, HICR = 1)** Reset:: Depends on reset configuration sequence Type: R/W 0x0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—					HF0	HF1	—	INIT	HM		HF2	HF3	—	TREQ	RREQ

ICR **Host Interface Control Register (DMA = 0, HICR = 0)** Reset:: Depends on reset configuration sequence 0x0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—					HF0	HF1	—	INIT	HDM		HF2	HF3	HDRQ	TREQ	RREQ
Type	R/W									R		R/W				

ICR **Host Interface Control Register (DMA = 1, HICR = 0)** Reset:: Depends on reset configuration sequence 0x0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—					HF0	HF1	—	INIT	HDM		HF2	HF3	—	Note 1	Note 2
Type	R/W									R		R/W				

Notes: 1. $\overline{\text{HDM0}}$ when read, TREQ when written. 2. HDM0 when read, RREQ when written.

ICR Bit Descriptions

Bits	Name	Description	Settings
5–6	HF[0–1]	Host Flags 0–1	Host can set or clear; these bits are reflected in the HSR[HF] bits.
8	INIT	Force Initialization	Host can set, then HDI16 runs the INIT command and clears the INIT bit.
9–10	HM/HDM[0–2]	Host Mode/Host DMA Mode	See information on host DMA mode control values in the *MSC8101 Reference Manual*.
11–12	HF[2–3]	Host Flags 2–3	Host can set or clear; these bits are reflected in the HSR[HF] bits.
13	HDRQ	HREQ/HTRQ and HACK/HRRQ Pin Control (Available only in non-DMA [interrupt] mode)	0 = HREQ and HACK 1 = HTRQ and HRRQ

Table 6-5. HDI16 Registers (Continued)

14	TREQ/$\overline{\text{HDM0}}$	HREQ/HTREQ Pin Control	TREQ when written; $\overline{\text{HDM0}}$ when read
15	RREQ/$\overline{\text{HDM0}}$	HREQ and HRREQ Pin Control	RREQ when written; $\overline{\text{HDM0}}$ when read

CVR **Command Vector Register** Reset: Depends on reset configuration sequence Type: R/W 0x1

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—							NMI	HC	HV						

CVR Bit Descriptions

Bits	Name	Description	Settings
7	NMI	Non-Maskable Interrupt	
8	HC	Host Command	
9–15	HV	Host Vector bits	

ISR **Interface Status Register** Reset: Depends on reset configuration sequence Type: R 0x2

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—								HREQ	HF4	HF5	HF6	HF7	TRDY	TXDE	RXDF

ISR Bit Descriptions

Bits	Name	Description	Settings
8	HREQ	HREQ Status	HDRQ cleared: 0 = No host processor interrupts requested 1 = Interrupt requested HDRQ set: 0 = No host processor interrupts requested (HTRQ or HRRQ cleared) 1 = Interrupt requested (HTRQ or HRRQ set)
9–12	HF[4–7]	Host flags 4–7	Reflect state of HCR[HF] bits
13	TRDY	TRDY Status	0 = TX[0–3] and the HORX FIFO are empty 1 = TTX[0–3] and the HORX FIFO are not empty
14	TXDE	Transmit Data Empty	0 = TX registers are not empty 1 = TX registers are empty and can be written by the host processor
15	RXDF	Receive Data Full	0 = RX registers are not full 1 = RX registers are full and can be read by the host processor

Table 6-5. HDI16 Registers (Continued)

RXx	**Receive Word Registers [0–3]**	Type: R	(RX0) 0x7 (RX1) 0x6	(RX2) 0x5 (RX3) 0x4
TXx	**Transmit Word Registers [0–3]**	Type: W	(TX0) 0x7 (TX1) 0x6	(TX2) 0x5 (TX3) 0x4
RSCF	**Reset Configuration Registers [0–3]**	Type: W	(RSCFG0) 0x8 (RSCFG1) 0x9	(RSCFG2) 0xA (RSCFG3) 0xB

Table 6-6. EFCOP Registers

FCNT **Filter Count Register** Reset: 0 Type: R/W 0x0C60

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	FCNT (Number of coefficients minus 1)															

FCTL **EFCOP Control Register** Reset: 0 Type: R/W 0x0C80

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	FDOM	FDIM	FONEIE	FOFIE	FINFIE	FIEIE	FUDIE	FCIM	FPRC	FMLC	FOM		FUPD	FAPD	FLT	FEN

FCTL Bit Descriptions

Bits	Name	Description	Settings
0	FDOM	Data Output Mode	0 = DMA triggered on output buffer not empty 1 = DMA triggered on output buffer full
1	FDIM	Data Input Mode	0 = DMA triggered on input buffer not full 1 = DMA triggered on input buffer empty
2	FONEIE	Data Output Not Empty Interrupt Enable	0 = Disabled 1 = Enabled
3	FOFIE	Data Output Full Interrupt Enable	0 = Disabled 1 = Enabled
4	FINFIE	Data Input Not Full Interrupt Enable	0 = Disabled 1 = Enabled
5	FIEIE	Data Input Empty Interrupt Enable	0 = Disabled 1 = Enabled
6	FUDIE	Coefficients Update Done Interrupt Enable	0 = Disabled 1 = Enabled
7	FCIM	Coefficients Initialization Mode	0 = Disabled 1 = Enabled
8	FPRC	Filter Processing State Initialization Mode	0 = EFCOP processing starts after state initialization 1 = EFCOP processing starts with no state initialization
9	FMLC	Filter Multichannel Mode	0 = Disabled 1 = Enabled

Table 6-6. EFCOP Registers (Continued)

10	FOM	Filter Operation Mode	00 = Mode 0: Real FIR filter 01 = Mode 1: Full complex FIR filter 10 = Mode 2: Complex FIR filter with alternate real and imaginary outputs 11 = Mode 3: Magnitude
12	FUPD	Filter Update	0 = Update mode disabled 1 = Update mode enabled
13	FADP	Filter Adaptive Mode	0 = Disabled 1 = Enabled
14	FLT	Filter Type	0 = FIR filter 1 = IIR filter
15	FEN	Filter Enable	0 = EFCOP disabled (individual reset) 1 = EFCOP enabled

FACR **EFCOP ALU Control Register** Reset: 0 Type: R/W 0x0CA0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—								FSCO	FISL	—		FRM		FSCL	

FACR Bit Descriptions

Bits	Name	Description	Settings
8	FSCO	Filter Shared Coefficients Mode	0 = Coefficients stored sequentially for each channel 1 = Same coefficients used for each channel
9	FISL	Filter Input Scale	0 = Scale both IIR feedback terms and IIR input 1 = Scale IIR feedback terms only
12–13	FRM	Filter Rounding Mode	00 = Convergent rounding 01 = Two's complement rounding 10 = Truncation (no rounding) 11 = Reserved
14–15	FSCL	Filter Scaling	00 = Scaling factor = 1 (no shift) 01 = Scaling factor = 8 (3-bit arithmetic left shift) 10 = Scaling factor = 16 (43-bit arithmetic left shift) 11 = Reserved

FDBA **EFCOP Data Base Address** Reset: 0 Type: R/W 0x0CC0

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

FCBA **EFCOP Coefficient Base Address** Reset: 0 Type: R/W 0x0CE0

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

Table 6-6. EFCOP Registers (Continued)

FDCH **EFCOP Decimation/Channel Count Register** Reset: 0 Type: R/W 0x0D00

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—				FDCM				—		FCHL					

FDCH Bit Descriptions

Bits	Name	Description	Settings
4–7	FDCM	Filter Decimation	0–15 = Decimation factor minus one
10–15	FCHL	Filter Channels	0–63 = Number of channels to process minus one

FSTR **EFCOP Status Register** Reset: 0 Type: R 0x0D20

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—									FOBNE	FDOBF	FIBNF	FDIBE	FUDN	FOVF	FSAT

FSTR Bit Descriptions

Bits	Name	Description	Settings	
9	FONBE	Filter Data Output Buffer Not Empty	0 = FDOR empty	1 = FDOR not empty
10	FDOBF	Filter Data Output Buffer Full	0 = FDOR not full	1 = FDOR full
11	FIBNF	Filter Data Input Buffer Not Full	0 = FDIR full	1 = FDIR not full
12	FDIBE	Filter Data Input Buffer Empty	0 = FDIR not empty	1 = FDIR empty
13	FUDN	Coefficients Update Done	0 = Session has not ended	1 = Session has ended
14	FOVF	Filter Overflow	0 = FMAC addition over- or underflow	1 = No over- or underflow
15	FSAT	Filter Saturation	0 = No over- or underflow saturation	1 = Saturation

Table 6-7. PIC Registers

ELIRA **Edge/Level-Triggered Interrupt Priority Register A** Reset: 0 Type: R/W 0x1C00

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED3	PIL30	PIL31	PIL32	PED2	PIL20	PIL21	PIL22	PED1	PIL10	PIL11	PIL12	PED0	PIL00	PIL01	PIL02

ELIRB **Edge/Level-Triggered Interrupt Priority Register B** Reset: 0 Type: R/W 0x1C08

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED7	PIL70	PIL71	PIL72	PED6	PIL60	PIL61	PIL62	PED5	PIL50	PIL51	PIL52	PED4	PIL40	PIL41	PIL42

Table 6-7. PIC Registers (Continued)

ELIRC Edge/Level-Triggered Interrupt Priority Register C — Reset: 0 — Type: R/W — 0x1C10

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED11	PIL110	PIL111	PIL112	PED10	PIL100	PIL101	PIL102	PED9	PIL 90	PIL91	PIL 92	PED8	PIL80	PIL81	PIL82

ELIRD Edge/Level-Triggered Interrupt Priority Register D — Reset: 0 — Type: R/W — 0x1C18

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED15	PIL150	PIL151	PIL152	PED14	PIL140	PIL141	PIL142	PED13	PIL130	PIL131	PIL132	PED12	PIL120	PIL121	PIL122

ELIRE Edge/Level-Triggered Interrupt Priority Register E — Type: R/W — 0x1C20

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED19	PIL190	PIL191	PIL192	PED18	PIL180	PIL181	PIL182	PED17	PIL170	PIL171	PIL172	PED16	PIL160	PIL161	PIL162
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ELIRF Edge/Level-Triggered Interrupt Priority Register F — Type: R/W — 0x1C28

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PED23	PIL230	PIL231	PIL232	PED22	PIL220	PIL221	PIL222	PED21	PIL210	PIL211	PIL212	PED20	PIL200	PIL201	PIL202
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ELIR*x* Bit Descriptions

Bits	Name	Description	Settings	
0, 4, 8, 12	PEDxx	Trigger Mode for IR Input xx	0 = Level-triggered mode	1 = Edge-triggered mode
1–3, 5–7, 9–11, 13-15	PILxxx	Priority Level for IR Input xx	000 = Interrupts disabled 001 = IPL 0 (lowest priority) 010 = IPL 1 011 = IPL 2/IPL3	100 = IPL 4 101 = IPL 5 110 = IPL 6 111 = IPL 7 (highest priority)

IPRA Interrupt Pending Register A — Reset: 0 — Type: R/W — 0x1C30

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	IP15	IP14	IP13	IP12	IP11	IP10	IP9	IP8	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0

Table 6-7. PIC Registers (Continued)

IPRA Bit Descriptions

Bits	Name	Description	Settings
0–15	IP[15–0]	Status of IR Inputs 15–0	Level-triggered mode: 0 = No IR pending 1 = IR pending Edge-triggered mode: 0 = No IR pending 1 = IR acknowledged by SC140 core

IPRB **Interrupt Pending Register B** Reset: 0 Type: R/W 0x1C38

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	IP31	IP30	IP29	IP28	IP27	IP26	IP25	IP24	IP23	IP22	IP21	IP20	IP19	IP18	IP17	IP16

IPRB Bit Descriptions

Bits	Name	Description	Settings
0–7	IP[31–24]	Status of NMI Inputs 31–24	0 = No NMI pending 1 = NMI acknowledged by SC140 core
8–15	IP[23–16]	Status of IR Input 23–16	Level-triggered mode: 0 = No IR pending 1 = IR pending Edge-triggered mode: 0 = No IR pending 1 = IR acknowledged by the SC140 core

6.3 SIU Registers

Table 6-8. SIU Registers

BCR **Bus Configuration Register** Reset: Depends on reset configuration sequence Type: R/W 0x10024

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	EBM	APD			—				PLDP	—		EAV	ETM	LETM	EPAR	—
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	NPQM			—		EXDD	—					ISPS	—			

BCR Bit Descriptions

Bits	Name	Description	Settings
0	EBM	External Bus Mode	0 = Single MSC8101 bus mode 1 = 60x-compatible bus mode

Table 6-8. SIU Registers (Continued)

1–3	APD	Address Phase Delay	Specifies the number of address tenure wait states for address operations initiated by 60x bus master.
8	PLDP	Pipeline Maximum Depth	0 = One 1 = Zero
11	EAV	Enable Address Visibility	0 = Bank select signals driven on 60x bus address lines (no full address visibility). 1 = Bank select signals not driven on address bus (during READ and WRITE commands to SDRAM devices, full address is driven on 60x bus address lines).
12	ETM	Compatibility Mode Enable	0 = Strict 60x bus mode (extended transfer mode disabled) 1 = Extended transfer mode enabled
13	LETM	Local Bus Compatibility Mode Enable	0 = Extended transfer mode disabled 1 = Extended transfer mode enabled
14	EPAR	Even Parity	0 = Odd parity 1 = Even parity
16–18	NPQM	Non-MSC8101 Master Connected to Arbitration Lines	0 = Bus master is an MSC8101 1 = Bus master is not an MSC8101
21	EXDD	External Master Delay Disable	0 = Memory controller inserts one wait state between the assertion of $\overline{TS}$ and of CS when the external master accesses an address space controlled by the memory controller. 1 = Memory controller asserts CS on the cycle following the assertion of $\overline{TS}$ by an external master accessing an address space controlled by the memory controller.
27	ISPS	Internal Space Port Size	0 = MSC8101 acts as 64-bit slave. 1 = MSC8101 acts as 32-bit slave.

PPC_ACR PowerPC 60x Bus Arbiter Configuration Register Reset: Depends on reset configuration sequence Type: R/W 0x10028

Bit	0	1	2	3	4	5	6	7
	—		DBGD	EARB	PRKM			

PPC_ACR Bit Descriptions

Bits	Name	Description	Settings
2	DBGD	Data Bus Grant Delay	0 = $\overline{DBG}$ asserted with $\overline{TS}$ if the data bus is free. 1 = $\overline{DBG}$ asserted one cycle after $\overline{TS}$ if the data bus is not busy.
3	EARB	External Arbitration	0 = Internal arbitration performed. 1 = External arbitration assumed.

Table 6-8. SIU Registers (Continued)

4–7	PRKM	Parking Master	0000 = CPM high request level 0001 = CPM middle request level 0010 = CPM low request level 0011 = Reserved 0100 = Reserved 0101 = SC140 core interface 0110 = Reserved	0111 = External master 1 1000 = External master 2 1001 = External master 3 1010 = DMA priority 0 1011 = DMA priority 1 1100 = DMA priority 2 1101–1111 = Reserved

PPC_ALRH **PowerPC 60x Bus Arbitration-Level Register** Type: R/W 0x1002C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Priority Field 0				Priority Field 1				Priority Field 2				Priority Field 3			
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Priority Field 4				Priority Field 5				Priority Field 6				Priority Field 7			
Reset	0	1	1	1	1	0	0	0	1	0	0	1	0	0	1	1

PPC_ALRL **PowerPC 60x Bus Arbitration-Level Register** Type: R/W 0x10030

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Priority Field 8				Priority Field 9				Priority Field 10				Priority Field 11			
Reset	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Priority Field 12				Priority Field 13				Priority Field 14				Priority Field 15			
Reset	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1

LCL_ACR **PowerPC Local Bus Arbiter Configuration Register** Type: R/W 0x10034

Bit	0	1	2	3	4	5	6	7
	—		DBGD	—	PRKM			
Reset	0	0	0	0	0	0	1	0

LCL_ACR Bit Descriptions

Bits	Name	Description	Settings
2	DBGD	Data Bus Grant Delay	0 = $\overline{\text{DBG}}$ asserted with $\overline{\text{TS}}$ if the data bus is free. 1 = $\overline{\text{DBG}}$ asserted one cycle after $\overline{\text{TS}}$ if the data bus is not busy.

Table 6-8. SIU Registers (Continued)

4–7	PRKM	Parking Master	0000 = CPM high request level 0001 = CPM middle request level 0010 = CPM low request level 0011 = Host bridge 0100–1001 = Reserved 1010 = DMA priority 0 1011 = DMA priority 1 1100 = DMA priority 2 1101–1111 = Reserved

LCL_ALRH **PowerPC Local Bus Arbitration-Level Register** Type: R/W 0x10038

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Priority Field 0				Priority Field 1				Priority Field 2				Priority Field 3			
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Priority Field 4				Priority Field 5				Priority Field 6				Priority Field 7			
Reset	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	1

LCL_ALRL **PowerPC Local Bus Arbitration-Level Register** Type: R/W 0x1003C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Priority Field 8				Priority Field 9				Priority Field 10				Priority Field 11			
Reset	1	0	0	0	1	0	0	1	1	0	1	0	1	0	1	1
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Priority Field 12				Priority Field 13				Priority Field 14				Priority Field 15			
Reset	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1

SIUMCR **SIU Model Configuration Register** Reset: 0 Type: Depends on reset configuration sequence 0x10000

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BBD	ESE	PBSE	IRQ7INT	DPPC		IRPC		—		TCPC		BC1PC		BCTLC	
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	MMR		—													

SIUMCR Bit Descriptions

Bits	Name	Description	Settings
0	BBD	Bus Busy Disable	0 = $\overline{\text{ABB}}$/$\overline{\text{IRQ2}}$ pin is $\overline{\text{ABB}}$, $\overline{\text{DBB}}$/$\overline{\text{IRQ3}}$ pin is $\overline{\text{DBB}}$ 1 = $\overline{\text{ABB}}$/$\overline{\text{IRQ2}}$ pin is IRQ2, $\overline{\text{DBB}}$/$\overline{\text{IRQ3}}$ pin is IRQ3
1	ESE	External Snoop Enable	0 = Disabled 1 = Enabled

Table 6-8. SIU Registers (Continued)

2	PBSE	Parity Byte Select Enable	0 = Disabled 1 = Enabled
3	IRQ7INT	IRQ7 or INT_OUT selection	0 = $\overline{\text{IRQ7}}$ 1 = $\overline{\text{INT\_OUT}}$
4–5	DPPC	Data Parity Pins Configuration	00 = NC, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$, $\overline{\text{IRQ6}}$ 01 = DP0, DP1, DP2, DP3, DP4, DP5, DP6 10 = NC, $\overline{\text{IRQ1}}$, NC, NC, DREQ3, DREQ4, DACK3 11 = $\overline{\text{EXT\_BR2}}$, $\overline{\text{EXT\_BG2}}$, $\overline{\text{EXT\_DBG2}}$, $\overline{\text{EXT\_BR3}}$, $\overline{\text{EXT\_BG3}}$, $\overline{\text{EXT\_DBG3}}$, $\overline{\text{IRQ6}}$
6–7	IRPC	Interrupt Pin Configuration (multiplexing)	00 = NC, NC, NC 01 = $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, $\overline{\text{IRQ5}}$ 10 = BADDR29, BADDR30, BADDR31 11 = Reserved
10–11	TCPC	Transfer Codes Pin Configuration	00 = TC0, TC1, TC2 01 = reserved 10 = BNKSEL0, BNKSEL1, BNKSEL2 11 = reserved
12–13	BC1PC	Buffer Control 1-Pin Configuration	00 = $\overline{\text{BCTL1}}$ 01 = $\overline{\text{BCTL1}}$ 10 = reserved 11 = reserved
14-15	BCTLC	Buffer Control Configuration	00 = $\overline{\text{BCTL0}}$ controls W/$\overline{\text{R}}$. $\overline{\text{BCTL1}}$ controls $\overline{\text{OE}}$. 01 = $\overline{\text{BCTL0}}$ controls W/$\overline{\text{R}}$. $\overline{\text{BCTL1}}$ controls $\overline{\text{OE}}$. 10 = $\overline{\text{BCTL0}}$ controls $\overline{\text{WE}}$. $\overline{\text{BCTL1}}$ controls $\overline{\text{RE}}$. 11 = reserved
16–17	MMR	Mask Master's Requests	00 = No masking on bus request lines 01 = Reserved 10 = Reserved 11 = All external bus requests masked (boot master is the internal core)

IMMR **Internal Memory Map Register** Reset: Depends on reset configuration sequence 0x101A8

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	ISB															—
Type	R/W															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	PARTNUM								MASKNUM							
Type	R															

Table 6-8. SIU Registers (Continued)

IMMR Bit Descriptions

Bits	Name	Description	Settings
0–14	ISB	Internal Space Base	Configured at reset to one of seven addresses. See the *MSC8101 Reference Manual* chapters on memory maps and reset for details.
16–23	PARTNUM	Part Number	Mask-programmed to 0x50: code corresponds to part number.
24–31	MASKNUM	Mask Number	Mask-programmed to 0x00: code corresponds to mask number

SYPCR **System Protection Control Register** Type: R/W 0x10004

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	SWTC															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	BMT								PBME	LBME	—			SWE	SWRI	SWP
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1

SYPCR Bit Descriptions

Bits	Name	Description	Settings
0–15	SWTC	Software Watchdog Timer Count	Contains the count value for the software watchdog timer (SWT)
16–23	BMT	Bus Monitor Timing	Defines the time-out period for the bus monitor. Granularity: 8 bus clocks
24	PBME	PowerPC 60x Bus Monitor Enable	0 = Disabled 1 = Enabled
25	LBME	PowerPC Local Bus Monitor Enable	0 = Disabled 1 = Enabled
29	SWE	Software Watchdog Enable	0 = Disabled 1 = Enabled
30	SWRI	Software Watchdog Reset/Interrupt Select	0 = SWT and bus monitor time-out cause machine check interrupt to core 1 = Same conditions cause a hard reset
31	SWP	Software Watchdog Prescale	0 = SWT clock not prescaled 1 = SWT clock prescaled

SWSR **Software Service Register** Type: W 0x1000E

Note: 1. To prevent SWT time-out , write 0x556C followed by 0xAA39.

Table 6-8. SIU Registers (Continued)

TESCR1 **PowerPC 60x Bus Transfer Error Status and Control Register 1** Reset: 0 Type: R/W 0x10040

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BM	ISBE	PAR	ECC2	ECC1	WP	EXT	TC			—	TT				
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	DMD	—						ECNT							

TESCR1 Bit Descriptions

Bits	Name	Description	Bits	Name	Description
0	BM	PowerPC 60x Bus Monitor Time-Out	6	EXT	External Error
1	ISBE	Internal Space Bus Error	7–9	TC	Transfer Code
2	PAR	PowerPC 60x Bus Parity Error	11–15	TT	Transfer Type
3	ECC2	Double ECC Error	17	DMD	Data Error Disable
4	ECC1	Single ECC Error	24–31	ECNT	Single ECC Error Counter
5	WP	Write Protect Error			

TESCR2 **PowerPC 60x Bus Transfer Error Status and Control Register 2** Reset: 0 Type: R/W 0x10044

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—	REGS	DPR	—				LCL	PB							
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	BNK								—							

TESCR2 Bit Descriptions

Bits	Name	Description	Bits	Name	Description
1	REGS	SEB1 Internal Registers Error	8–15	PB	Parity error on byte
2	DPR	Dual Port RAM Error	16–23	BNK	Memory controller bank
7	LCL	PowerPC Local Bus Bridge Error			

Table 6-8. SIU Registers (Continued)

L_TESCR1 **Local Bus Transfer Error Status and Control Register 1** Reset: 0 Type: R/W 0x10048

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BM	—				WP	—	TC			—	TT				
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	DMD	—													

L_TESCR1 Bit Descriptions: See TESCR1

TMCNTSC **Time Counter Status and Control Register** Reset: 0 Type: R/W 0x10220

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—								SEC	ALR	—		SIE	ALE	TCF	TCE

TMCNTSC Bit Descriptions

Bits	Name	Description	Settings
8	SEC	Once Per Second Interrupt	Status bit: set every second; should be cleared by software.
9	ALR	Alarm Interrupt	Status bit: set when TMCNT value equals TMCNTAL value.
12	SIE	Second Interrupt Enable	0 = Time counter does not generate an interrupt when SEC is set. 1 = Time counter generates an interrupt when SEC is set.
13	ALE	Alarm Interrupt Enable	1 = Time counter generates an interrupt when ALR is set.
14	TCF	Time Counter Frequency	0 = 4 MHz 1 = 32 KHz
15	TCE	Time Counter Enable	0 = Disabled 1 = Enabled

TMCNT **Time Counter Register** Reset: — Type: R/W 0x10224

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	TMCNT (Time Counter)															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	TMCNT (Time Counter)															

Table 6-8. SIU Registers (Continued)

TMCNTAL **Time Counter Alarm Register** Reset: — Type: R/W 0x1022C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	ALARM															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	ALARM															

TMCNTAL Bit Descriptions

Bits	Name	Description	Settings
0–31	ALARM	Alarm	Match corresponding TMCNT bits = generate the alarm interrupt.

PISCR **Periodic Interrupt Status and Control Register** Reset: 0 Type: R/W 0x10240

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—								PS	—				PIE	PTF	PTE

PISCR Bit Descriptions

Bits	Name	Description	Settings
8	PS	Periodic Interrupt Status	Asserted if the PIT issues an interrupt
13	PIE	Periodic Interrupt Enable	1 = PIT generates an interrupt when PS = 1
14	PTF	Periodic Interrupt Frequency	0 = 4 MHz 1 = 32 KHz.
15	PTE	Periodic Timer Enable	0 = Disable counter 1 = Enable counter

PITC **Periodic Interrupt Timer Count Register** Reset: 0 Type: R/W 0x10244

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PITC															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

PITC Bit Descriptions

Bits	Name	Description	Settings
0–15	PITC	Periodic Interrupt Timing Count	0xFFFF = Maximum count period

Table 6-8. SIU Registers (Continued)

PITR **Periodic Interrupt Timer Register** Reset: 0 Type: R 0x10248

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PIT															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

PITR Bit Descriptions

Bits	Name	Description	Settings
0–15	PIT	Periodic Interrupt Timer	Current count remaining for the periodic timer

Table 6-9. Reset Registers

RSR **Reset Status Register** Type: R/W 0x10C92

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—										JTRS	—	SWRS	BMRS	ESRS	EHRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

RSR Bit Descriptions

Bits	Name	Description	Settings
26	JTRS	JTAG Reset Status	0 = No reset event occurred 1 = Reset event occurred
28	SWRS	Software Watchdog Reset Status	0 = No reset event occurred 1 = Reset event occurred
29	BMRS	Bus Monitor Reset Status	0 = No reset event occurred 1 = Reset event occurred
30	ESRS	External Soft Reset Status	0 = No reset event occurred 1 = Reset event occurred
31	EHRS	External Hard Reset Status	0 = No reset event occurred 1 = Reset event occurred

Table 6-10. Interrupt Registers

SICR/SICR_EXT **SIU Interrupt Configuration Register** Reset: 0 Type: R/W (SICR) 0x10C00 (SICR_EXT) 0x10C40

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—		HP						—						GSIU	SPS

SICR/SICR_EXT Bit Descriptions

Bits	Name	Description	Settings
2–7	HP	Highest Priority	To retain original priority, program HP to the XSIU1 interrupt number.
14	GSIU	Group SIU (relative XSIU priority scheme)	0 = Grouped 1 = Spread
15	SPS	Spread Priority Scheme (relative YCC priority scheme)	0 = Grouped 1 = Spread

SIPRR/SIPRR_EXT **SIU Interrupt Priority Register** Type: R/W (SIPRR) 0x10C10 (SIPRR_EXT) 0x10C50

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	XS1P			XS2P			XS3P			XS4P			—			
Reset	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	XS5P			XS6P			XS7P			XS8P			—			
Reset	1	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0

SIPRR/SIPRR_EXT Bit Descriptions

Bits	Name	Description	Settings
0–2	XS1P–XSIU1	Priority order	000 = TMCNT asserts its request in the XSIU1 position 001 = PIT asserts its request in the XSIU1 position 010 = Reserved 011 = IRQ1 asserts its request in the XSIU1 position 100 = IRQ2 asserts its request in the XSIU1 position 101 = IRQ3 asserts its request in the XSIU1 position 110 = IRQ4 asserts its request in the XSIU1 position 111 = IRQ5 asserts its request in the XSIU1 position
3–11, 16–27	XS2P–XS8P	Same as XS1P, but for XSIU2–XSIU8.	

Table 6-10. Interrupt Registers (Continued)

SIPNR_H/SIPNR_H_EXT SIU High Interrupt Pending Register Reset: 0[1] Type: R/W (SIPNR_H) 0x10C08 (SIPNR_H_EXT) 0x10C48

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—				PC4	PC5	PC6	PC7	—				PC12	PC13	PC14	PC15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	—					TMCNT	PIT	—

Notes: 1. These bits are zero after reset because their corresponding mask register bits are cleared (disabled).

SIPNR_L/SIPNR_L_EXT SIU Low Interrupt Pending Register Reset: 0[1] Type: R/W (SIPNR_L) 0x10C0C (SIPNR_L_EXT) 0x10C4C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	FCC1	FCC2	FCC3	—	MCC1	MCC2	—		SCC1	SCC2	SCC3	SCC4	—	DMA	—	—
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	I2C	SPI	RTT	SMC1	SMC2	—				SDMA	—	TIME1	TIMER2	TIMER3	TIMER4	—

Notes: 1. These bits are zero after reset because their corresponding mask register bits are cleared (disabled).

SIMR_H/SIMR_H_EXT SIU High Interrupt Mask Register Reset: 0 Type: R/W (SIMR_H) 0x10C1C (SIMR_H_EXT) 0x10C5C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—				PC4	PC5	PC6	PC7	—				PC12	PC13	PC14	PC15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	—					TMCNT	PIT	—

Notes: 1. TMCNT = time counter; PIT = periodic interrupt timer

SIMR_L/SIMR_L_EXT SIU Low Interrupt Mask Register Reset: 0 Type: R/W (SIMR_L) 0x10C20 (SIMR_L_EXT) 0x10C60

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	FCC1	FCC2	FCC3	—	MCC1	MCC2	—		SCC1	SCC2	SCC3	SCC4	—	DMA[1]	—	—
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	I2C	SPI	RTT	SMC1	SMC2	—				SDMA	—	TIME1	TIMER2	TIMER3	TIMER4	—

Notes: 1. This bit is valid for SIMR_L_EXT. It is *Reserved* in SIMR_L.

Table 6-10. Interrupt Registers (Continued)

SIVEC/SIVEC_EXT **SIU Interrupt Vector Register** Reset: 0 Type: R (SIVEC) 0x10C04 (SIVEC_EXT) 0x10C44

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Interrupt Code						—									
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

SIEXR/SIEXR_EXT **SIU External Interrupt Control Register** Reset: 0 Type: R/W (SIEXR) 0x10C24 (SIEXR_EXT) 0x10C64

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—				EDPC4	EDPC5	EDPC6	EDPC7	—				EDPC12	EDPC13	EDPC14	EDPC15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—	EDI1	EDI2	EDI3	EDI4	EDI5	EDI6	EDI7	—							

SIEXR/SIEXR_EXT Bit Descriptions

Bits	Name	Description	Settings
0–15	EDPCx	Edge Detect Mode for Port Cx	0 = Any change on PCx generates an interrupt request 1 = High-to-low change on PCx generates an interrupt request
16–23	EDIx	Edge Detect Mode for $\overline{\text{IRQx}}$	0 = Low assertion on $\overline{\text{IRQx}}$ generates an interrupt request 1 = High-to-low change on $\overline{\text{IRQx}}$ generates an interrupt request

Table 6-11. Clocks Registers

SCCR **System Clock Control Register** Reset: — Type: R/W 0x10C80

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0														DFBRG	

Table 6-11. Clocks Registers (Continued)

SCCR Bit Descriptions

Bits	Name	Description	Settings
30–31	DFBRG	Division Factor for the BRG Clock	00 = Divide by 4 01 = Divide by 16 (default) 10 = Divide by 64 11 = Divide by 256

SCMR **System Clock Mode Register** Reset: — Type: R 0x10C88

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	—		COREPDF		COREMF				BUSDF				CPMDF			
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	SPLLPDF				SPLLMF				—	DLLDIS	—					

SCMR Bit Descriptions

Bits	Name	Description	Settings
2–3	COREPDF	Core PLL Pre-Division Factor	00 = CPLL PDF = 1 01 = CPLL PDF = 2 10 = CPLL PDF = 3 11 = CPLL PDF = 4
4–7	COREMF	Core Multiplication Factor	0101 = MF = 10 0110 = MF = 12 (No other combinations are used.)
8–11	BUSDF	PowerPC 60x Bus Division Factor	0010 = Bus DF = 3 0011 = Bus DF = 4 0100 = Bus DF = 5 (No other combinations are used.)
12–15	CPMDF	CPM Division Factor	0001 - CPM DF = 2 (No other combinations are used.)
16–19	SPLLDF	SPLL Pre-Division Factor	0000 = SPLL PDF = 1 0001 = SPLL PDF = 2 0010 = SPLL PDF = 3 0011 = SPLL PDF = 4 (No other combinations are used.)
20–23	SPLLMF	SPLL Multiplication Factor	0110 = SPLL MF = 12 0111 = SPLL MF = 14 1000 = SPLL MF = 16 1001 = SPLL MF = 18 1010 = SPLL MF = 20 1011 = SPLL MF = 22 1100 = SPLL MF = 24 1101 = SPLL MF = 26 1110 = SPLL MF = 28 1111 = SPLL MF = 30 (No other combinations are used.)
25	DLLDIS	DLL disable	0 = DLL operation enabled 1 = DLL operation disabled

Table 6-12. Memory Controller Registers

BR*x* **Base Registers [0–7, 10–11]** Reset: Depends on reset configuration sequence Type: R/W

Addr (BR0) 0x10100 (BR2) 0x10110 (BR4) 0x10120 (BR6) 0x10130 (BR10) 0x10150
(BR1) 0x10108 (BR3) 0x10118 (BR5) 0x10128 (BR7) 0x10138 (BR11) 0x10158

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BA															
	Reset: 0 (bits 16–31)															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	BA	—		PS		DECC		WP	MS			EMEMC	ATOM		DR	V

BR*x* Bit Descriptions

Bits	Name	Description	Settings
0–16	BA	Base Address	
19–20	PS	Port Size	01 = 8-bit 10 = 16-bit 11 = 32-bit 00 = 64-bit
21–22	DECC	Data Error Correction and Checking	00 = Disabled 01 = Normal parity checking 10 = R/modify/W parity checking 11 = ECC correction and checking
23	WP	Write Protect	0 = R/W access 1 = Read-only
24–26	MS	Machine Select	000 = GPCM—60x bus (reset value) 001 = GPCM—PowerPC Local bus 010 = SDRAM—PowerPC 60x Bus 011 = SDRAM—reserved 100 = UPMA 101 = UPMB 110 = UPMC 111 = Reserved
27	EMEMC	External MEMC Enable	0 = MEMC handles accesses according to MS. 1 = External memory controller handles accesses on the PowerPC 60x bus
28–29	ATOM	Atomic Operation	00 = Address space controlled by MEMC bank not used for atomic operations 01 = Read-after write atomic (RAWA) 10 = Write-after-read-atomic (WARA) 11 = Reserved
30	DR	Data Pipelining	0 = Disabled 1 = Data beats delayed by one cycle
31	V	Valid Bit	0 = Bank invalid 1 = Bank valid Note: After system reset, the V bit is set in BR0 and reset in BR[1–11].

Table 6-12. Memory Controller Registers (Continued)

OR*x* **Option Register—SDRAM Mode** Reset: 0 Type: R/W

Addr (OR0) 0x10104 (OR2) 0x10114 (OR4) 0x10124 (OR6) 0x10134 (OR10) 0x10154
(OR1) 0x1010C (OR3) 0x1011C (OR5) 0x1012C (OR7) 0x1013C (OR11) 0x1015C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	SDAM												LSDAM			
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	LSDAM	BPD		ROWST				NUMR			PMSEL	IBID	—			

OR*x*—SDRAM Mode Bit Descriptions

Bits	Name	Description	Settings
0–11	SDAM	SDRAM Address Mask	0000_0000_0000 = 4 GB 1000_0000_0000 = 2 GB 1100_0000_0000 = 1 GB 1110_0000_0000 = 512 MB 1111_0000_0000 = 256 MB 1111_1000_0000 = 128 MB 1111_1100_0000 = 64 MB 1111_1110_0000 = 32 MB 1111_1111_0000 = 16 MB 1111_1111_1000 = 8 MB 1111_1111_1100 = 4 MB 1111_1111_1110 = 2 MB 1111_1111_1111 = 1 MB
12–16	LSDAM	Lower SDRAM Address Mask	Note: Reset to 0x0 to implement a 1 MB minimum when using SDRAM.
SDRAM Page Information			
17–18	BPD	Internal Banks Per Device (128-MB devices, 10 =invalid)	00 = 2 01 = 4 10 = 8 11 = Reserved
19–22	ROWST	Row Start Address Bit	For PSDMR[PBI] = 0: 0010 = A7 0100 = A8 0110 = A9 1000 = A10 1010 = A11 1100 = A12 1110 = A13 Other values reserved For PSDMR[PBI] = 1: 0000 = A0 0001 = A1 ... 1100 = A12 1101–1111 Reserved
23–25	NUMR	Number of Row Address Lines	000 = 9 010 = 11 100 = 13 110 = 15 001 = 10 011 = 12 101 = 14 111 = 16
26	PMSEL	Page Mode Select	0 = Back-to-back page mode (normal operation) 1 = Page kept open until a page miss or refresh occurs

Table 6-12. Memory Controller Registers (Continued)

27	IBID	Internal Bank Interleaving Within Same Device Disable	0 = Enabled 1 = Disabled

OR*x* **Option Register—GPCM Mode** Type: R/W

Addr (OR0) 0x10104 (OR2) 0x10114 (OR4) 0x10124 (OR6) 0x10134 (OR10) 0x10154
(OR1) 0x1010C (OR3) 0x1011C (OR5) 0x1012C (OR7) 0x1013C (OR11) 0x1015C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	AM															
Reset	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	AM	—		BCTLD	CSNT	ACS		—	SCY				SETA	TRLX	EHTR	
Reset	0	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0

OR*x*—GPCM Mode Bit Descriptions

Bits	Name	Description	Settings
0–16	AM	Address Mask	0 = Corresponding bits masked 1 = Corresponding address bits used
19	BCTLD	Data Buffer Control Disable	0 = $\overline{\text{BCTL}x}$ asserted. 1 = $\overline{\text{BCTL}x}$ not asserted
20	CSNT	Chip-Select Negation Time	0 = $\overline{\text{CS}}$/$\overline{\text{WE}}$ negated normally 1 = $\overline{\text{CS}}$/$\overline{\text{WE}}$ negated a quarter of a clock earlier
21–22	ACS	Address to Chip-Select Setup ($\overline{\text{CS}}$ output /address change)	00 = Same time 10 = $\overline{\text{CS}}$ output 1/4 clock after 01 = Reserved 11 = $\overline{\text{CS}}$ output 1/2 clock after
24–27	SCY	Cycle Length in Clocks	0000 = 0 wait states … 1111 = 15 wait states
28	SETA	External Access Termination	0 = $\overline{\text{PSDVAL}}$ generated internally by the memory controller unless $\overline{\text{GTA}}$ is asserted earlier externally. 1 = PSDVAL generated after external logic asserts $\overline{\text{GTA}}$
29	TRLX	Timing Relaxed	0 = Normal 1 = Relaxed
30–31	EHTR	Extended Hold Time on Read Accesses (clock cycles inserted)	00 = 0 01 = 1 10 = 4 11 = 8

Table 6-12. Memory Controller Registers (Continued)

OR*x* **Option Register—UPM Mode** Reset: 0 Type: R/W

Addr (OR0) 0x10104 (OR2) 0x10114 (OR4) 0x10124 (OR6) 0x10134 (OR10) 0x10154
(OR1) 0x1010C (OR3) 0x1011C (OR5) 0x1012C (OR7) 0x1013C (OR11) 0x1015C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	AM															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	AM	—		BCTLD	—			BI	—					EHTR		—

OR*x*—UPM Mode Bit Descriptions

Bits	Name	Description	Settings
0–16	AM	Address Mask	0 = Corresponding bits masked 1 = Corresponding address bits used
19	BCTLD	Data Buffer Control Disable	0 = $\overline{\text{BCTL}x}$ asserted. 1 = $\overline{\text{BCTL}x}$ not asserted
23	BI	Burst Inhibit (memory bank burst access support)	0 = Supports 1 = Does not support
29–30	EHTR	Extended Hold Time on Read Accesses (clock cycles inserted)	00 = 0 01 = 1 10 = 4 11 = 8

PSDMR **60x SDRAM Protocol-Specific Mode Register** Reset: 0 Type: R/W 0x1090

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PBI	RFEN	OP			SDAM			BSMA			SDA10			RFRC	
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	RFRC	PRETOACT			ACTTORW			BL	LDOTOPRE		WRC		EAMUX	BUFCMD	CL	

PSDMR Bit Descriptions

Bits	Name	Description	Settings
0	PBI	Page-Based Interleaving	0 = Bank-based interleaving 1 = Page-based interleaving
1	RFEN	Refresh Enable	0 = Refresh services not required. 1 = Refresh services required
2–4	OP	SDRAM Operation	000 = Normal 001 = CBR refresh 010 = Self refresh 011 = Mode register write 100 = Precharge bank 101 = Precharge all banks 110 = Activate bank 111 = R/W

Table 6-12. Memory Controller Registers (Continued)

5–7	SDAM	Address Multiplex Size	SDAM	External. PowerPC Bus Addr Pin		Signal on External Pin
			000	A13–A31		A5–A23
			001	A14–A31		A5–A22
			010	A15–A31		A5–A21
			011	A16–A31		A5–A20
			100	A17–A31		A5–A19
			101	A18–A31		A5–A18
8–10	BSMA	Bank Select Multiplexed Address Line	000 = A12–A14 001 = A13–A15 010 = A14–A16 011 = A15–A17		100 = A16–A18 101 = A17–A19 110 = A18–A20 111 = A19–A21	
11–13	SDA10	"A10" Control	For PBI = 0: 000 = A12 001 = A11 010 = A10 011 = A9	 100 = A8 101 = A7 110 = A6 111 = A5	For PBI = 1: 000 = A10 001 = A9 010 = A8 011 = A7	 100 = A6 101 = A5 110 = A4 111 = A3
SDRAM Device-Specific Parameters:						
14–16	RFRC	Refresh Recovery (recovery interval in clock cycles)	000 = Reserved 001 = 3	010 = 4 011 = 5	100 = 6 101 = 7	110 = 8 111 = 16
17–19	PRETOACT	Precharge to Activate Interval (clock-cycle wait states)	001 = 1	010 = 2 …	111 = 7	000 = 8
20–22	ACTTORW	Activate to Read/Write Interval (clock-cycle wait states)	001 = 1	010 = 2 …	111 = 7	000 = 8
23	BL	Burst Length	0 = 4 (for device port size 64 or 16)		1 = 8 (for device port size 32 or 8)	
24–25	LDOTOPRE	Last Data Out to Precharge (in clock cycles)	00 = 0	01 = -1	10 = -2	11 = Reserved
26–27	WRC	Write Recovery Time (in clock cycles)	01 = 1	10 = 2	11 = 3	00 = 4
28	EAMUX	External Address Multiplexing Enable/Disable	0 = Disabled (fastest timing) 1 = Enabled. Memory controller asserts SDAMUX for an extra cycle before issuing an ACTIVATE command to the SDRAM			
29	BUFCMD	Command Buffer	0 = Normal timing for control lines 1 = All control lines except $\overline{CS}$ are asserted for two cycles			
30–31	CL	CAS Latency	00 = Reserved	01 = 1	10 = 2	11 = 3

Table 6-12. Memory Controller Registers (Continued)

M*x*MR **Machine A/B/C Mode Registers** Type: R/W

Addr (MAMR) 0x10170 (MBMR) 0x10174 (MCMR) 0x10178

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BSEL	RFEN	OP		—	AM*x*			DS*x*		G0CL*x*			GPL_x4DIS	RLF*x*	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	RLF*x*		WLF*x*				TLF*x*				MAD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M*x*MR Bit Descriptions

Bits	Name	Description	Settings
0	BSEL	Bus Select (for banks that select UPM*x*)	0 = Assigned to PowerPC 60x bus 1 = Assigned to PowerPC Local bus
1	RFEN	Refresh Enable	0 = Refresh services not required 1 = Refresh services required
2–3	OP	Command Opcode	00 = Normal operation 01 = Write to array 10 = Read from array 11 = Run pattern
5–7	AM*x*	Address Multiplex Size	**AM*x*** / **External. PowerPC Bus Addr Pin** / **Signal on External Pin** 000 / A[16–31] / A[8–23] 001 / A[16–31] / A[7–22] 010 / A[16–31] / A[6–21] 011 / A[16–31] / A[5–20] 100 / A[17–31] / A[5–19] 101 / A[18–31] / A[5–18]
8–9	DS*x*	Disable Timer Period	00 = 1 cycle 01 = 2 cycles 10 = 3 cycles 11 = 4 cycles
10–12	G0CL*x*	General Line 0 Control	000 = A12 001 = A11 010 = A10 011 = A9 100 = A8 101 = A7 110 = A6 111 = A5

Table 6-12. Memory Controller Registers (Continued)

13	GPL_*x*4DIS	GPL_A4 Output Line Disable	0 = UPWAIT/$\overline{\text{GTA}}$/GPL_x4 behaves as GPL_4, UPMx[G4T4/DLT3] is interpreted as G4T4, UPMx[G4T3/WAEN] is interpreted as G4T3 1 = UPWAIT/$\overline{\text{GTA}}$/GPL_x4 behaves as UPWAIT, UPMx[G4T4/DLT3] is interpreted as DLT3, UPMx[G4T3/WAEN] is interpreted as WAEN
14–15	RLF*x*	Read Loop Field	0001 = 1 time 0010 = 2 times ... 1111 = 15 times 0000 = 16 times
18–21	WLF*x*	Write Loop Field	0001 = 1 time 0010 = 2 times … 1111 = 15 times 0000 = 16 times
22–25	TLF*x*	Refresh Loop Field	0001 = 1 time 0010 = 2 times … 1111 = 15 times 0000 = 16 times
26–31	MAD	Machine Address	Incremented by one each time the UPM is accessed and the OP field is set to WRITE or READ

MDR **Memory Data Register** Reset: 0 Type: R/W 0x10188

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	MD (memory Data)															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	MD															

MAR **Memory Address Register** Reset: 0 Type: R/W 0x10168

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	A (Memory Address)															
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	A															

PURT **PowerPC 60x Bus-Assigned UPM Refresh Timer** Reset: 0 Type: R/W 0x10198

Bit	0	1	2	3	4	5	6	7
	PURT (Refresh Timer Period)							

Equation: $TimerPeriod = \left(\frac{PURT}{F_{MPTC}}\right)$

PSRT **PowerPC 60x Bus-Assigned SDRAM Refresh Timer** Reset: 0 Type: R/W 0x1019C

Bit	0	1	2	3	4	5	6	7
	PSRT (Refresh Timer Period)							

Equation: $TimerPeriod = \left(\frac{PSRT + 1}{F_{MPTC}}\right)$

Table 6-12. Memory Controller Registers (Continued)

MPTPR **Memory Refresh Timer Prescaler Register** Type: R/W 0x10184

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	PTP (Memory Timers Prescaler)								—							
Reset	0	0	0	0	0	0	1	*x*	0	0	0	0	0	0	0	0

Notes: 1. Where *x* can be a zero or a one—that is, 0000_0010 or 0000_0011

Table 6-13. DMA Registers

DCHCR*x* **DMA Channel Configuration Registers** Reset: 0 Type: R/W

Addr	(DCHCR0) 0x10700	(DCHCR4) 0x10710	(DCHCR8) 0x10720	(DCHCR12) 0x10730
	(DCHCR1) 0x10704	(DCHCR5) 0x10714	(DCHCR9) 0x10724	(DCHCR13) 0x10734
	(DCHCR2) 0x10708	(DCHCR6) 0x10718	(DCHCR10) 0x10728	(DCHCR14) 0x10738
	(DCHCR3) 0x1070C	(DCHCR7) 0x1071C	(DCHCR11) 0x1072C	(DCHCR15) 0x1073C

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	ACTV	PPC	—			EXP			DRS	DPL	BDPTR					
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	DRACK	FLY	—	RQNUM					FRZ	INT	—		PRIO			

DCHCR*x* Bit Descriptions

Bits	Name	Description	Settings
0	ACTV	Active DMA Channel *x*	0 = Disabled 1 = Enabled
1	PPC	PowerPC Bus	0 = Channel assigned to PowerPC Local bus 1 = Assigned to PowerPC 60x bus
5–7	EXP	Expiration Timer	Channel ignores EXP+1 bus cycles
8	DRS	DREQ Sensitivity Mode	0 = DREQ edge-triggered 1 = DREQ level-triggered
9	DPL	DREQ Polarity	0 = Active high or rising edge-triggered 1 = Active low or falling edge-triggered
10–15	BDPTR	Buffer Pointer	Pointer to the line in the DCPRAM assigned to this channel
16	DRACK	DRACK Protocol	0 = Channel does not use DRACK 1 = Channel uses DRACK

Table 6-13. DMA Registers (Continued)

17	FLY	Flyby Transaction	0 = Dual-access transaction 1 = Single-access transaction (flyby mode)	
19–23	RQNUM	Requestor Number	00000 = HDI16 read request 00001 = HDI16 write request 00010 = EFCOP read request 00011 = EFCOP write request 001xx = Reserved 01000 = External request 1,DREQ1	01001 = External request 2, DREQ2 01010 = External request 3, DREQ3 01011 = External request 4, DREQ4 011xx = Reserved 1xxxx = Reserved
24	FRZ	Freezes Channel	0 = Channel operation normal	1 = Channel frozen
25	INT	Internal Requestor	0 = External request	1 = Internal request
28–31	PRIO	Channel Priority	0000 = Highest priority	111 = Lowest priority

DPCR **DMA Pin Configuration Register** Reset: 0 Type: R/W 0x1078C

Bit	0	1	2	3	4	5	6	7
	—				SDN0	SDN1	—	

DPCR Bit Descriptions

Bit	Name	Description	Settings	
4	SDN0	Select DONE[0]	0 = PD31 functionality is DONE	1 = PD31 functionality is DRACK
5	SDN1	Select DONE[1]	0 = PD30 functionality is DONE	1 = PD30 functionality is DRACK

DSTR **DMA Status Register** Reset: 0 Type: R/W 0x10784

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

DIMR **DMA Internal Mask Register** Reset: 0 Type: R/W 0x10780

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

Table 6-13. DMA Registers (Continued)

DEMR **DMA External Mask Register** Reset: 0 Type: R/W 0x10790

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—															

DTEAR **DMA Transfer Error Address Status Register** Reset: 0 Type: R/W 0x10788

Bit	0	1	2	3	4	5	6	7
	DBER_P	DBER_L	—					

DTEAR Bit Descriptions

Bits	Name	Description	Settings
0	DBER_P	DMA Channel PowerPC 60x Bus Error	DMA transfer error address: PDMTEA. Channel number: PDMTER
1	DBER_L	DMA Channel PowerPC Local Bus Error	SDMA transfer error address: LDMTEA. Channel number: LDMTER

xDMTER **DMA Transfer Error Request Number Registers** Reset:: Undefined Type: R (PDMTER) 0x10064 (LDMTER) 0x1006C

Bit	0	1	2	3	4	5	6	7
	RQNUM					—		

PDMTER and LDMTER Bit Description

Bits	Name	Description	Settings
0–4	RQNUM	Requestor Number	Code number of requestor accessing the bus when the bus error occurred

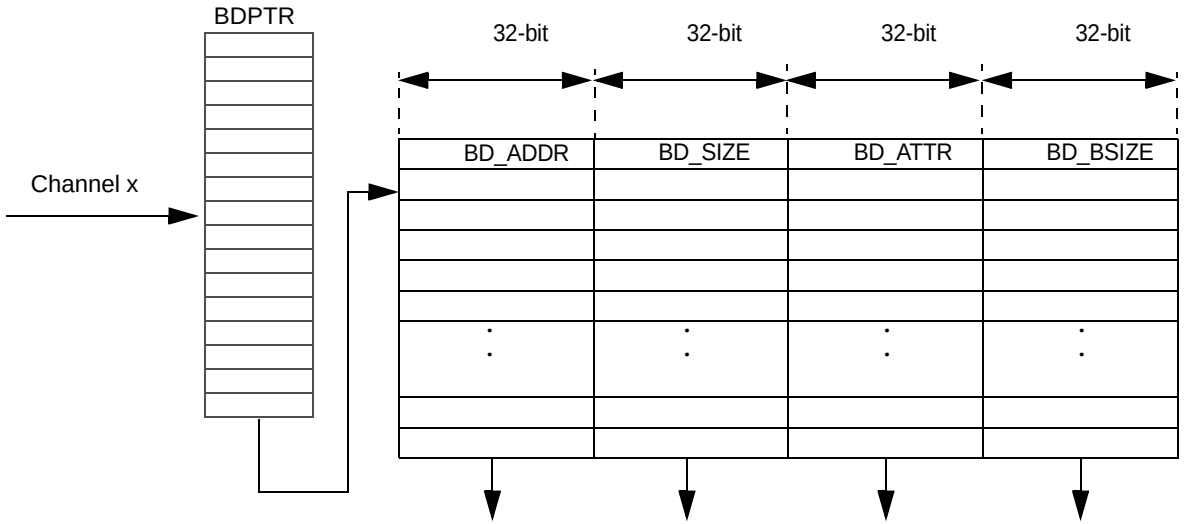
xDMTEA **DMA Transfer Error Address Registers** Reset: Undefined Type: R

	PDMTEA	Holds the system address accessed during a DMA transfer error on the 32-bit PowerPC 60x bus.	0x10060
	LDMTEA	Holds the system address accessed during a DMA transfer error on the 32-bit PowerPC Local bus.	0x10068

Table 6-13. DMA Registers (Continued)

DCPRAM **DMA Channel Parameters RAM** Type: R/W 0x10800–10BFF

DCPRAM Structure



DCPRAM Bit Descriptions

Bits	Name	Description	Bits	Name	Description
0–31	BD_ADDR	Buffer's current address	64–95	BD_ATTR	Buffer's attributes and temporary data
32–63	BD_SIZE	Size of transfer left for the current buffer	96–127	BD_BSIZE	Buffer's base size

BD_ATTR **Buffer Attributes Parameter** Reset: Undefined Type: R/W See DCPRAM

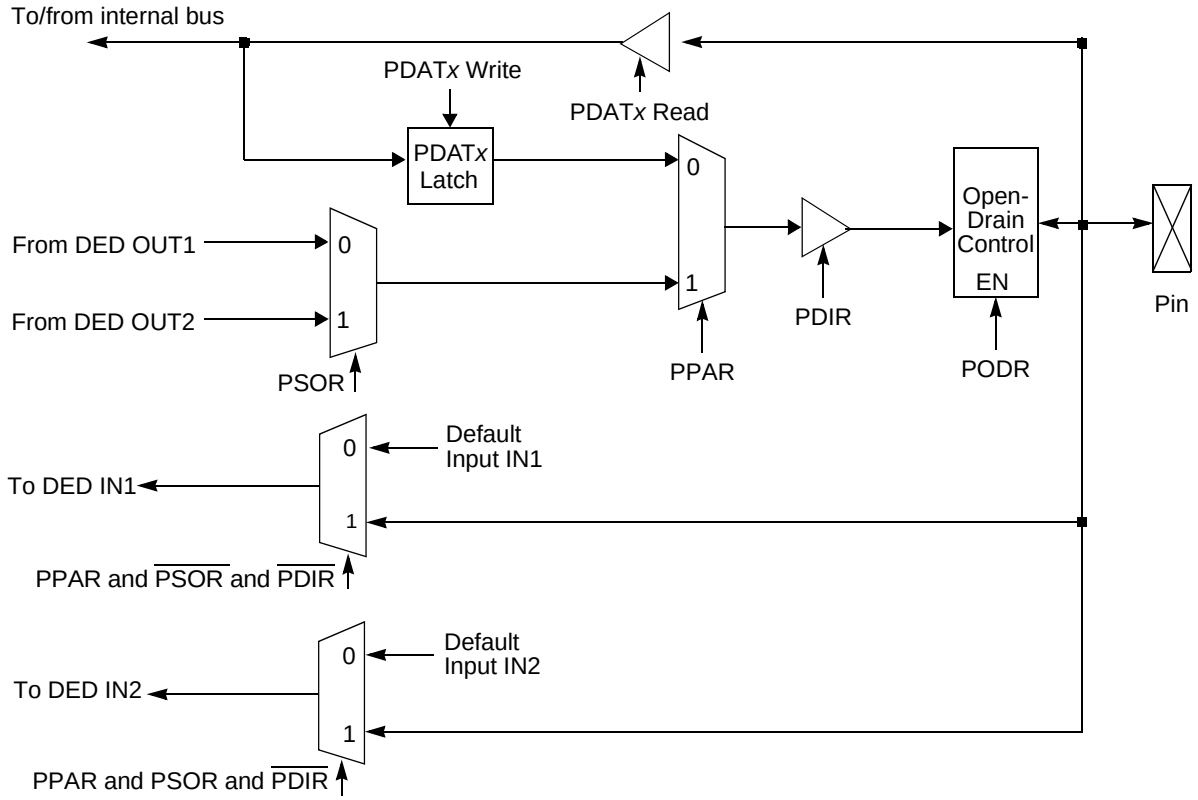
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	INTRPT	CYC	CONT	—	NO_INC	BP		—		NBUS	NBD					
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—						TSZ			—	FLS	RD	—	TC	—	GBL

Table 6-13. DMA Registers (Continued)

BD_ATTR Bit Descriptions

Bits	Name	Description	Settings
0	INTRPT	Interrupt	0 = Do not issue interrupt 1 = Issue interrupt when size reaches zero
1	CYC	Cyclic Address	0 = Sequential address; 1 = Cyclic address
2	CONT	Continuous Buffer Mode	0 = Buffer closed when BD_SIZE = 0 1 = Buffer continues to operate
4	NO_INC	Increments Address	0 = Increment address after request is serviced 1 = Do not increment
5–6	BP	Bus Priority (arbitrate for bus mastership)	00 = Request 1010 01 = Request 1011 10 = Request 1100 11 = Reserved
9	NBUS	Next Bus	0 = PowerPC Local bus 1 = PowerPC 60x bus
10–15	NBD	Next Buffer	Points the next buffer to call when size reaches zero and CONT is set
22–24	TSZ	Transfer Size (maximum transaction size)	001 = 8 bits 010 = 16 bits 011 = 32 bits 000 = 64 bits 100 = One burst 101 = Reserved 11x = Reserved
26	FLS	Flush FIFO	0 = Do not flush. 1 = Flush.
27	RD	Read Channel	0 = Write transaction. 1 = Read transaction.
29	TC	Transfer Code	0 = TC[0–2] value is 100. 1 = TC[0–2] value is 101.
31	GBL	Global Transaction	0 = Non-global transaction. 1 = Global transaction.

6.4 CPM, Parallel I/O Ports



Port Register Name	Description	0	1
PDATx	Port Data Registers [A–D]	0	1
PDIRx	Port Data Direction[1] Registers [A–D]	Input	Output
PPARx	Port Pin Assignment Registers [A–D]	General-purpose	Dedicated
PSORx	Port Special Options Registers [A–D]	Dedicated 1	Dedicated 2
PODRx	Port Open-Drain Registers [A–D]	Regular	Open drain
Notes: 1. Bidirectional signals must be programmed as inputs (PDIR = 0).			

Figure 6-2. Port Functional Block Diagram

Table 6-14. Parallel I/O Port Registers

PDAT*x* **Port Data Registers [A–D]** Reset: — Type: R/W

Addr (PDATA) 0x10D10 (PDATB) 0x10D30 (PDATC) 0x10D50 (PDATD) 0x10D70

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	D0[1]	D1[1]	D2[1]	D3[1]	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31

Notes: 1. These bits are valid for PDATA and PDATC only.

PDIR*x* **Port Data Direction Registers [A–D]** Reset: 0 Type: R/W

Addr (PDIRA) 0x10D00 (PDIRB) 0x10D20 (PDIRC) 0x10D40 (PDIRD) 0x10D60

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	DR0[1]	DR1[1]	DR2[1]	DR3[1]	DR4	DR5	DR6	DR7	DR8	DR9	DR10	DR11	DR12	DR13	DR14	DR15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	DR16	DR17	DR18	DR19	DR20	DR21	DR22	DR23	DR24	DR25	DR26	DR27	DR28	DR29	DR30	DR31

Notes: 1. These bits are valid for PDIRA and PDIRC only.

PDIR*x* Bit Descriptions

Bits	Name	Description	Settings
0–31	DR*x*	Direction	0 = Input 1 = Output

PPAR*x* **Port Pin Assignment Registers [A–D]** Reset: 0 Type: R/W

Addr (PPARA) 0x10D04 (PPARB) 0x10D24 (PPARC) 0x10D44 (PPARD) 0x10D64

Bit	0[1]	1[1]	2[1]	3[1]	4	5	6	7	8	9	10	11	12	13	14	15
	DD0	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	DD11	DD12	DD13	DD14	DD15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	DD16	DD17	DD18	DD19	DD20	DD21	DD22	DD23	DD24	DD25	DD26	DD27	DD28	DD29	DD30	DD31

Notes: 1. These bits are valid for PPARA and PPARC only.

Table 6-14. Parallel I/O Port Registers (Continued)

PPAR*x* Bit Descriptions

Bits	Name	Description	Settings
0–31	DD*x*	Dedicated Enable	0 = GPIO 1 = Dedicated peripheral function.

PSOR*x* **Port Special Options Registers [A–D]** Reset: 0 Type: R/W

Addr (PSORA) 0x10D08 (PSORB) 0x10D28 (PSORC) 0x10D48 (PSORD) 0x10D68

Bit	0[1]	1[1]	2[1]	3[1]	4	5	6	7	8	9	10	11	12	13	14	15
	SO0	SO1	SO2	SO3	SO4	SO5	SO6	SO7	SO8	SO9	SO10	SO11	SO12	SO13	SO14	SO15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	SO16	SO17	SO18	SO19	SO20	SO21	SO22	SO23	SO24	SO25	SO26	SO27	SO28	SO29	SO30	SO31

Notes: 1. These bits are valid for PSORA and PSORC only.

PSOR*x* Bit Descriptions

Bits	Name	Description	Settings
0–31	SO*x*	Special Option (dedicate peripheral function)	0 = Option 1 1 = Option 2

PODR*x* **Port Open-Drain Registers [A–D]** Reset: 0 Type: R/W

Addr (PODRA) 0x10D0C (PODRB) 0x10D2C (PODRC) 0x10D4C (PODRD) 0x10D6C

Bit	0[1]	1[1]	2[1]	3[1]	4	5	6	7	8	9	10	11	12	13	14	15
	OD0	OD1	OD2	OD3	OD4	OD5	OD6	OD7	OD8	OD9	OD10	OD11	OD12	OD13	OD14	OD15
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	OD16	OD17	OD18	OD19	OD20	OD21	OD22	OD23	OD24	OD25	OD26	OD27	OD28	OD29	OD30	OD31

Notes: 1. These bits are valid for PODRA and PODRC only.

PODR*x* Bit Descriptions

Bits	Name	Description	Settings
0–31	OD*x*	Open-Drain Configuration	0 = Output 1 = Open-drain driver

7 Interrupts

7.1 Interrupt Structure

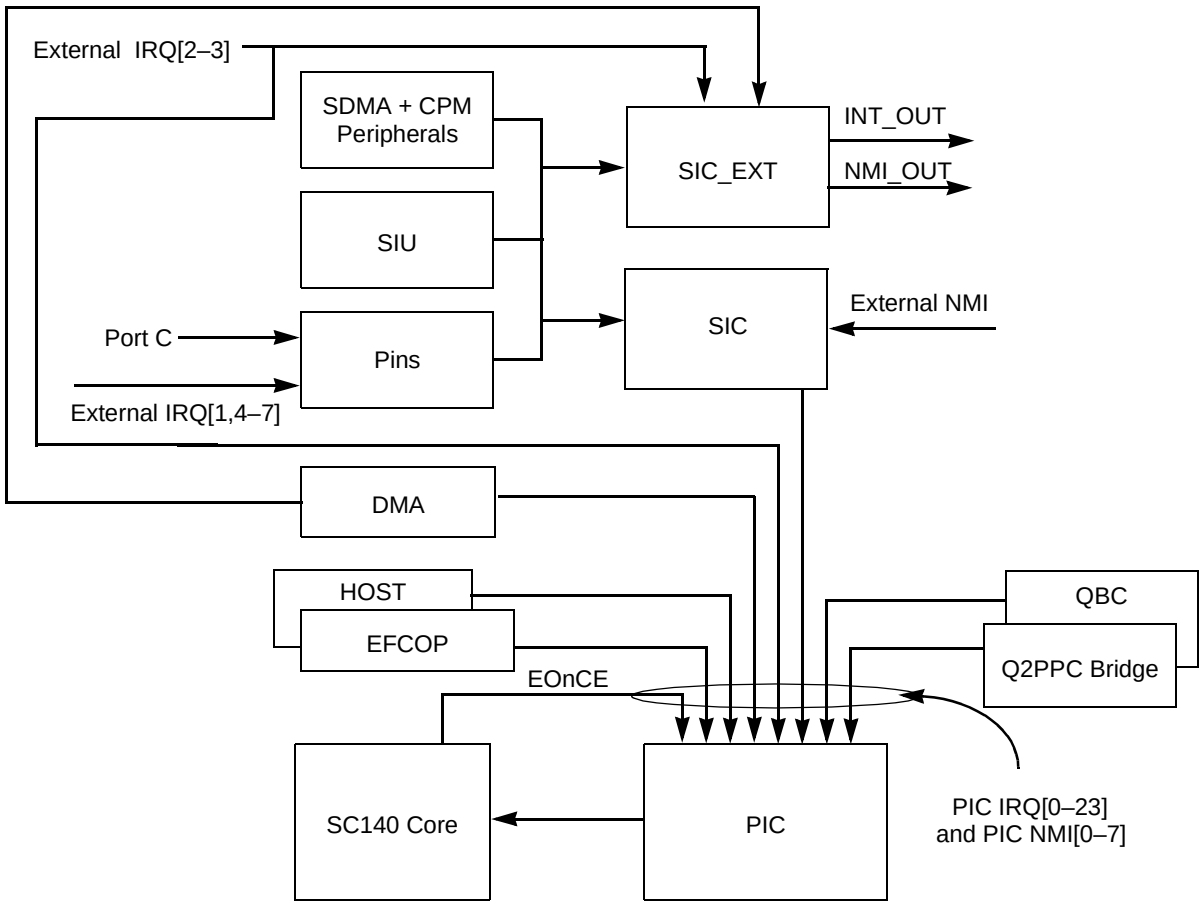


Figure 7-1. MSC8101 Interrupt Flow Diagram

7.2 Interrupt Priorities and Vector Tables

Table 7-1. SIC and SIC_EXT Interrupt Source Priority

Priority Level (Highest to Lowest)	**Description**	**Multiple Events**
1	Highest	—
2	XSIU1	No (TMCNT,PIT = Yes)
3	XSIU2 (GSIU = 0)	No (TMCNT,PIT = Yes)
4	XSIU3 (GSIU = 0)	No (TMCNT,PIT = Yes)
5	XSIU4 (GSIU = 0)	No (TMCNT,PIT = Yes)
6	XCC1	Yes
7	XCC2	Yes
8	XCC3	Yes
9	XCC4	Yes
10	XSIU2 (GSIU = 1)	No (TMCNT,PIT = Yes)
11	XCC5	Yes
12	XCC6	Yes
13	XCC7	Yes
14	XCC8	Yes
15	XSIU5 (GSIU = 0)	No (TMCNT,PIT = Yes)
16	XSIU6 (GSU = 0)	No (TMCNT,PIT = Yes)
17	XSIU7 (GSU = 0)	No (TMCNT,PIT = Yes)
18	XSIU8 (GSU = 0)	No (TMCNT,PIT = Yes)
19	XSIU3 (GSIU = 1)	No (TMCNT,PIT = Yes)
20	YCC1 (Grouped)	Yes
21	YCC2 (Grouped)	Yes
22	YCC3 (Grouped)	Yes
23	YCC4 (Grouped)	Yes
24	YCC5 (Grouped)	Yes
25	YCC6 (Grouped)	Yes
26	YCC7 (Grouped)	Yes
27	YCC8 (Grouped)	Yes
28	XSIU4 (GSIU = 1)	No (TMCNT,PIT = Yes)
29	Parallel I/O–PC15	Yes

Table 7-1. SIC and SIC_EXT Interrupt Source Priority (Continued)

Priority Level (Highest to Lowest)	**Description**	**Multiple Events**
30	Timer 1	Yes
31	Parallel I/O–PC14	Yes
32	YCC1 (Spread)	Yes
33	Parallel I/O–PC13	Yes
34	SDMA Bus Error	Yes
35	Reserved	No
36	YCC2 (Spread)	Yes
37	Parallel I/O–PC12	No
38	Reserved	No
39	Reserved	No
40	Timer 2	Yes
41	Reserved	No
42	XSIU5 (GSIU = 1)	No (TMCNT,PIT = Yes)
43	YCC3 (Spread)	Yes
44	RISC Timer Table	Yes
45	I^2C	Yes
46	YCC4 (Spread)	Yes
47	Reserved	No
48	Reserved	No
49	IRQ6	No
50	Reserved	No
51	IRQ7	No
52	Timer 3	Yes
53	XSIU6 (GSIU = 1)	No (TMCNT,PIT = Yes)
54	YCC5 (Spread)	Yes
55	Parallel I/O–PC7	No
56	Parallel I/O–PC6	No
57	Parallel I/O–PC5	No
58	Timer 4	Yes
59	YCC6 (Spread)	Yes
60	Parallel I/O–PC4	No

Table 7-1. SIC and SIC_EXT Interrupt Source Priority (Continued)

Priority Level (Highest to Lowest)	**Description**	**Multiple Events**
61	XSIU7 (GSIU = 1)	No (TMCNT,PIT = Yes)
62	Reserved	No
63	SPI	Yes
64	Reserved	No
65	Reserved	No
66	SMC1	Yes
67	YCC7 (Spread)	Yes
68	SMC2	Yes
69	Reserved	No
70	Reserved	No
71	XSIU8 (GSIU = 1)	No (TMCNT,PIT = Yes)
72	YCC8 (Spread)	Yes
73	Reserved	—

Table 7-2. SIC and SIC_EXT Interrupt Vectors

Interrupt Number	**Description**	**Interrupt Vector**
0	Error (No interrupt)	0b00_0000
1	I^2C	0b00_0001
2	SPI	0b00_0010
3	RISC Timers	0b00_0011
4	SMC1	0b00_0100
5	SMC2	0b00_0101
6–9	Reserved	0b00_0110–0b00_1001
10	Reserved	0b00_1001
11	SDMA	0b00_1010
12	Reserved	0b00_1011
13	Timer1	0b00_1100
14	Timer2	0b00_1101
15	Timer3	0b00_1110
16	Timer4	0b00_1111

Table 7-2. SIC and SIC_EXT Interrupt Vectors (Continued)

Interrupt Number	Description	Interrupt Vector
17	TMCNT	0b01_0000
18	PIT	0b01_0001
19	Reserved	0b01_0010
20	IRQ1	0b01_0011
21	IRQ2	0b01_0100
22	IRQ3	0b01_0101
23	IRQ4	0b01_0110
24	IRQ5	0b01_0111
25	IRQ6	0b01_1000
26	IRQ7	0b01_1001
27–31	Reserved	0b01_1010–0b01_1111
32	FCC1	0b10_0000
33	FCC2	0b10_0001
34	FCC3	0b10_0010
35	Reserved	0b10_0011
36	MCC1	0b10_0100
37	MCC2	0b10_0101
38	Reserved	0b10_0110
39	Reserved	0b10_0111
40	SCC1	0b10_1000
41	SCC2	0b10_1001
42	SCC3	0b10_1010
43	SCC4	0b10_1011
44	Reserved	0b10_1100
45	DMA	0b10_1101
46–47	Reserved	0b10_1110–0b10_1111
48	PC15	0b11_0000
49	PC14	0b11_0001
50	PC13	0b11_0010
51	PC12	0b11_0011
52–55	Reserved	0b11_0100–0b11_0111
56	PC7	0b11_1000

Table 7-2. SIC and SIC_EXT Interrupt Vectors (Continued)

Interrupt Number	Description	Interrupt Vector
57	PC6	0b11_1001
58	PC5	0b11_1010
59	PC4	0b11_1011
60–63	Reserved	0b11_1100–0b11_1111

Table 7-3. PIC Interrupt Vectors

VAB[0–5]	Signal	Description	Service Routine Address (Offset from VBA)
0x0	TRAP	Internal exception (generated by trap instruction)	0x0
0x1	—	Reserved	0x40
0x2	ILLEGAL	Illegal instruction or set	0x80
0x3	DEBUG	Debug exception (EOnCE)	0xC0
0x4	—	Reserved	0x100
0x5	OVERFLOW	Overflow exception (DALU)	0x140
0x6	DEFAULT NMI	In VAB disabled mode only* *Not supported in the MSC8101	0x180
0x7	DEFAULT $\overline{\text{IRQ}}$	In VAB disabled mode only* *Not supported in the MSC8101	0x1C0
0x8-0x1F	—	Reserved	0x200–0x7FF
0x20	IRQ0	EFCOP (0): Input FIFO not full	0x800
0x21	IRQ1	EFCOP (1): Input FIFO empty	0x840
0x22	IRQ2	EFCOP (2): Output FIFO full	0x880
0x23	IRQ3	EFCOP (3): Output FIFO not empty	0x8C0
0x24	IRQ4	EFCOP (4): Update done	0x900
0x25	IRQ5	HDI16 (0): Receive FIFO full	0x940
0x26	IRQ6	HDI16 (1): Receive FIFO not empty	0x980
0x27	IRQ7	HDI16 (2): Transmit FIFO empty	0x9C0
0x28	IRQ8	HDI16 (3): Transmit FIFO not full	0xA00
0x29	IRQ9	HDI16 (4): External HOST command	0xA40
0x2A	IRQ10	Bus controller (x-y contention)	0xA80
0x2B	IRQ11	Bus controller (level1 contention)	0xAC0
0x2C	IRQ12	Bus controller (p-x contention)	0xB00
0x2D	IRQ13	Bus controller (non-aligned data error)	0xB40
0x2E	IRQ14	Reserved	0xB80
0x2F	IRQ15	External IRQ2 (edge/level configurable)	0xBC0

Table 7-3. PIC Interrupt Vectors (Continued)

VAB[0–5]	Signal	Description	Service Routine Address (Offset from VBA)
0x30	IRQ16	SIC interrupt	0xC00
0x31	IRQ17	External IRQ3 (edge/level configurable)	0xC40
0x32	IRQ18	DMA interrupt (channel/buffer terminated)	0xC80
0x33	IRQ19	Reserved	0xCC0
0x34	IRQ20	EOnCE interrupt (edge-triggered)	0xD00
0x35	IRQ21	Reserved	0xD40
0x36	IRQ22	Reserved	0xD80
0x37	IRQ23	Reserved	0xDC0
0x38	NMI0	HDI16: External Host NMI	0xE00
0x39	NMI1	Reserved	0xE40
0x3A	NMI2	Bus controller (memory write error)	0xE80
0x3B	NMI3	Bus controller (non-aligned error)	0xEC0
0x3C	NMI4	Bus controller (bus error)	0xF00
0x3D	NMI5	Q2PPC (TEA)	0xF40
0x3E	NMI6	Reserved	0xF80
0x3F	NMI7	SIC NMI, for example, S/W watchdog, external NMI, parity error	0xFC0

8 CPM

8.1 Protocols Supported

Table 8-1. MSC8101 Protocols Supported versus Channels

Protocol	FCC1	FCC2	FCC3	SCC[1–2]	SCC[3–4]	MCC[1–2]	SMC[1–2]	SPI	I^2C
ATM (Utopia 8)	+								
ATM (serial)	+								
Ethernet (100BaseT)	+	+							
Ethernet (10BaseT)	+	+		+					
HDLC	+	+	+	+	+				
HDLC_BUS		+	+	+	+				
TRANSPARENT	+			+			+		
UART				+			+		
Multi channel HDLC						+			
Multi channel transparent						+			
Bisync				+	+				
SPI								+	
I^2C									+

8.2 Interfaces Supported

Table 8-2. Interfaces Supported

Communication Controllers	Interfaces Supported
FCC1	UTOPIA 8 MII (Ethernet:100BaseT/10BaseT) TDM NMSI
FCC2	MII (Ethernet:100BaseT/10BaseT) TDM NMSI
FCC3	TDM
MCC[1–2]	TDM
SCC[1–2]	Ethernet: 10BaseT RS232 with modem controls TDM NMSI
SCC[3–4]	TDM
SMC[1–2]	RS232 TDM NMSI
SPI	SPI NMSI
I^2C	I^2C NMSI

8.3 Serial Performance

Table 8-3. MSC8101 Serial Performance

FCC1	FCC2	MCC	CPM Clock	60x Bus Clock
155 Mbps ATM	100 BaseT		150 MHz	100 MHz
100BaseT	100 BaseT		150 MHz	100 MHz
155 Mbps ATM		128 * 64-Kbps channels	150 MHz	100 MHz
100BaseT	100 BaseT	128 * 64-Kbps channels	150 MHz	100 MHz
155 Mbps ATM		256 * 64-Kbps channels	150 MHz	100 MHz
100BaseT		256 * 64-Kbps channels	150 MHz	100 MHz
45 Mbps HDLC		256 * 64-Kbps channels	150 MHz	100 MHz
45 Mbps HDLC	100 BaseT	256 * 64-Kbps channels	150 MHz	100 MHz
100BaseT		16 * 576-Kbps channels	150 MHz	100 MHz

8.4 Parameter RAM Values

Table 8-4. Parameter RAM

Page	Address [1]	Peripheral	Size (Bytes)
1	0x8000	SCC1	256
2	0x8100	SCC2	256
3	0x8200	SCC3	256
4	0x8300	SCC4	256
5	0x8400	FCC1	256
6	0x8500	FCC2	256
7	0x8600	FCC3	256
8	0x8700	MCC1	128
	0x8780	Reserved	124
	0x87FC	SMC1_BASE	2
	0x87FE	Reserved	2
9	0x8800	MCC2	128
	0x8880	Reserved	124
	0x88FC	SMC2_BASE	2
	0x88FE	Reserved	2
10	0x8900	Reserved	252
	0x89FC	SPI_BASE	2
	0x89FE	Reserved	2
11	0x8A00	Reserved	224
	0x8AE0	RISC Timers	16
	0x8AF0	REV_NUM	2
	0x8AF2	Reserved	2
	0x8AF4	Reserved	4
	0x8AF8	RAND	4
	0x8AFC	I^2C_BASE	2
	0x8AFE	Reserved	2
12–16	0x8B00	Reserved	1280

Notes: 1. Offset from RAM_BASE

8.5 CP Command Operation Codes (Opcodes)

Table 8-5. CP Command Register

CPCR	CP Command Register	.	Reset: 0	Type: R/W	0x119C0

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	RST	PAGE					SBC				—					FLG
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	—		MCN								—		OPCODE			

CPCR Bit Descriptions

Bits	Name	Description	Settings
0	RST	Software Reset Command	
1–5	PAGE	Page Number	

Table 8-5. CP Command Register (Continued)

<table>
<tr><td>6–10</td><td>SBC</td><td>Sub-Block Code
<table>
<tr><th>Sub-Block</th><th>Code</th><th>Page</th><th>Sub-Block</th><th>Code</th><th>Page</th></tr>
<tr><td>FCC1</td><td>10000 (for ATM: 01110)</td><td>00100</td><td>SPI</td><td>01010</td><td>01001</td></tr>
<tr><td>FCC2</td><td>10001 (for ATM: 01110)</td><td>00101</td><td>I^2C</td><td>01011</td><td>01010</td></tr>
<tr><td>FCC3</td><td>10010</td><td>00110</td><td>Timer</td><td>01111</td><td>01010</td></tr>
<tr><td>SCC1</td><td>00100</td><td>00000</td><td>MCC1</td><td>11100</td><td>00111</td></tr>
<tr><td>SCC2</td><td>00101</td><td>00001</td><td>MCC2</td><td>11101</td><td>01000</td></tr>
<tr><td>SCC3</td><td>00110</td><td>00010</td><td></td><td></td><td></td></tr>
<tr><td>SCC4</td><td>00111</td><td>00011</td><td></td><td></td><td></td></tr>
<tr><td>SMC1</td><td>01000</td><td>00111</td><td></td><td></td><td></td></tr>
<tr><td>SMC2</td><td>01001</td><td>01000</td><td></td><td></td><td></td></tr>
<tr><td>RAND</td><td>01110</td><td>01010</td><td></td><td></td><td></td></tr>
</table></td><td></td></tr>
<tr><td>15</td><td>FLG</td><td>Command Semaphore Flag</td><td></td></tr>
<tr><td>18–25</td><td>MCN</td><td>MCC Channel Number</td><td>0x00 HDLC
0x0A ATM
0x0C Ethernet
0x0F Transparent</td></tr>
<tr><td>28–31</td><td>OPCODE</td><td>Operation Code</td><td>See Table 8-6. CP Command Operation Codes (Opcodes) below.</td></tr>
</table>

Table 8-6. CP Command Operation Codes (Opcodes)

Opcode	Channel								
	FCC	**SCC**	**SMC (UART/Transparent)**	**SMC (GCI)**	**SPI**	**I^2C**	**MCC**	**Timer**	**Special**
0000	init rx and tx params	init rx and tx params	init rx and tx params	init rx and tx params	init rx and tx params	init rx and tx params	init rx and tx params	—	—
0001	init rx params	init rx params	init rx params	—	init rx params	init rx params	init rx params	—	—
0010	init tx params	init tx params	init tx params	—	init tx params	init tx params	init tx params	—	—
0011	enter hunt mode	enter hunt mode	enter hunt mode	—	—	—	—	—	—
0100	stop tx	stop tx	stop tx	—	—	—	stop tx	—	—
0101	graceful stop tx	graceful stop tx	—	—	—	—	—	—	—
0110	restart tx	restart tx	restart tx	—	—	—	—	—	—
0111	close rx bd	close rx bd	close rx bd	—	close rx bd	close rx bd	—	—	—
1000	set group address	set group address	—	—	—	—	—	set timer	—
1001	—	—	—	gci timeout	—	—	stop rx	—	—
1010	atm transmit command	reset bcs	—	gci abort request	—	—	—	—	—
1011	—	—	—	—	—	—	—	—	—
1100	—	—	—	—	—	—	—	—	random number
11xx	Undefined. Reserved for use by Motorola-supplied RAM microcodes.								

9 Instructions

Section 9.1, *SC140 Instruction Type Grouping Rules*, and **Section 9.2**, *Conventions, Syntax, Abbreviations*, provide general information about the MSC8101 instructions and how this guide represents them. **Section 9.3**, *Instructions Grouped Alphabetically*, contains details about the instructions.

9.1 SC140 Instruction Type Grouping Rules

Each SC140 instruction belongs to one of the following four types:

- Type 1—Basic Data ALU and move. Single-word; used very frequently. Terminates execution set.
- Type 2—Additional Data ALU, move and AGU arithmetic. Single-word; less frequent than Type 1. Terminates execution set.
- Type 3—Two-word and three-word Data ALU, move and AGU arithmetic. Terminates execution set.
- Type 4—All others. One- or two-word. Cannot be grouped with any other instructions.

Serial grouping options:

- One to six Type 1
- One Type 2 with up to five Type 1
- One Type 3 with up to five Type 1
- One Type 4

9.2 Conventions, Syntax, Abbreviations

Table 9-1. Instruction Conventions

Convention	Definition
( )	Indirect address
aa	Absolute address
Cn	Control registers
Da	Single source/destination data register
Da:Db	Source/destination data register pair
De.E; Do.E	Data register extension (bits 39–32 + Ln bit)
De.E:Do.E	Data register extension pair (for example, D4.E–D5.E)

Table 9-1. Instruction Conventions (Continued)

Convention	Definition
Db	Single source data register
De	Even numbered data/core register
Dn	Destination data register
Do	Odd numbered data/core register
DR	Data or address register
Ea	Effective address
HP	High portion (bits [31–16]) of a register
Ln	Limit tag bit
LP	Low portion (bits [15–0]) of a register
rc	Rounding constant
Rn	Address register
rx	AGU source register
Rx	AGU source/destination register
{ }	Not part of the assembler syntax of an instruction, used for clarity
[a–b]	Bit range a to b in a register or memory
[a]	Bit number a in a register or memory

Table 9-2. Operations Syntax

Operator	Description
+	Add
–	Subtract
*	Multiply
/	Divide
| |	Absolute value

Table 9-2. Operations Syntax (Continued)

Operator	Description
&	Logical AND
|	Logical OR
⊕	Exclusive OR
~	Bitwise complement
= =	Test for equality, 1 if equal, 0 if not equal
→	Transfer left to right
↔	Either right or left transfers, but not both at once
>>	Arithmetic right shift (sign bits shifted right)
<<	Arithmetic or logical left shift (functionally the same)
>>>	Logical right shift
>	Compare for greater than
Rnd()	Rounding function
x:y	Concatenation of x and y

Table 9-3. Assembler Syntax

Operator	Description
#	Prefix for an immediate value (for example, #u5 means an immediate 5-bit unsigned number and #s16 is an immediate 16-bit signed number).
$	Prefix to a hexadecimal value (for example, #$1A4F as an immediate value or $2FC as an address offset).
>	Prefix for long addressing, forces the assembler to use an extra word to encode the displacement/offset.
<	Prefix for short addressing, forces the assembler to use the smallest instruction when encoding the displacement/offset.

Table 9-3. Assembler Syntax (Continued)

Operator	Description
label	Replace the word “label” in an instruction with the label name of an execution set in code. The instruction determines if the assembler substitutes an absolute address or a relative displacement in the opcode.
*	Assembler variable containing the address of the current execution set.
{ }	Not part of assembler syntax; used to define the range of addressing.

Table 9-4. Register Abbreviations

Abbreviation	Register Name
D[0–15]	General-Purpose Data Register
R[0–15]	General-Purpose Address Register
EMR	Exception and Mode Register
VBA	Vector Base Address Register
PCTL0	Phase-Locked Loop Control Register
PCTL1	Clock Control Register
SP	Stack Pointer Registers: normal (NSP) and exception (ESP)
PC	Program Counter
SR	StatusRegister
MCTL	Modifier Control Register
SA[0–3]	Start Address Registers
LC[0–3]	Loop Counter Registers
B[0–7]	Modulo Base Registers
N[0–3]	Offset Registers
M[0–3]	Modulo Registers

9.3 Instructions Grouped Alphabetically

Table 9-5. Instructions Grouped Alphabetically

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
ABS	Absolute value					DALU Arithmetic
	$\lvert Dn \rvert \rightarrow Dn$	ABS Dn	1	1	1	
ADC	Add long with carry					DALU Arithmetic
	Da + Db + C → Db	ADC Da,Db	1	1	1	
ADD	Add					DALU Arithmetic
	#u5 + Dn → Dn Da + Db → Dn	ADD #u5,Dn {0 ≤ u5 < 32} ADD Da,Db,Dn	1 1	1 1	1 1	
ADD2	Add two 16-bit values					DALU Arithmetic
	Da.H + Dn.H → Dn.H Da.L + Dn.L → Dn.L	ADD2 Da,Dn	1	1	2	
ADDA	Add (affected by the modifier mode)					AGU Arithmetic
	#u5 + Rx → Rx #s16 + rx → Rn rx + Rx → Rx	ADDA #u5,Rx {0 ≤ u5 < 32} ADDA #s16,rx,Rn {-2^{15} ≤ s16 < 2^{15}} ADDA rx,Rx	1 2 1	1 1 1	2 4 2	
ADDL1A	Add with 1-bit left shift of source operand (affected by the modifier mode)					AGU Arithmetic
	(rx<<1) + Rx → Rx	ADDL1A rx,Rx	1	1	2	
ADDL2A	Add with 2-bit left shift of source operand (affected by the modifier mode)					AGU Arithmetic
	(rx<<2) + Rx → Rx	ADDL2A rx,Rx	1	1	2	
ADDNC.W	Add without changing the carry bit in the status register (SR)					DALU Arithmetic
	#s16 + Da → Dn	ADDNC.W #s16,Da,Dn {-2^{15} ≤ s16 <2^{15}}	2	1	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
ADR	Add and round					DALU Arithmetic
	Rnd(Da + Dn) → Dn	ADR Da,Dn	1	1	1	
AND	Bitwise AND					DALU Logical
	#$000000u16 • Da → Dn #u16$0000 • Da → Dn Da • Dn → Dn	AND #0{u16},Da,Dn {0 ≤ u16 < 2^{16}} AND #{u16}$0000,Da,Dn {0 ≤ u16 < 2^{16}} AND Da,Dn	2 2 1	1 1 1	4 4 2	
AND	Bitwise AND with 16-bit operand					Bit-Mask
	#u16 • DR.L → DR.L #u16 • DR.L → DR.L	AND #u16,DR.L AND #u16,DR.H	2 2	2 2	3 3	
AND.W	Bitwise AND with 16-bit Immediate					Bit-Mask
	#u16 • (R) → (R) #u16 • (SP – u5) → (SP – u5) #u16 • (a16) → (a16) #u16 • (SP + s16) → (SP + s16)	AND.W #u16,(Rn) {0 ≤ u16 < 2^{16}} AND.W #u16,(SP – u5) {0 ≤ u16 < 2^{16}} {0 ≤ u5 < 64, W} AND.W #u16,(a16) {0 ≤ u16 < 2^{16}} {0 ≤ a16 < 2^{16}, W} AND.W #u16,(SP + s16) {0 ≤ u16 < 2^{16}} {–2^{15} ≤ s16 < 2^{15}, W}	2 2 3 3	2 3 2 3	3 3 3 3	
ASL	Arithmetic shift left by one bit					DALU Arithmetic
	Da << 1→ Dn	ASL Da,Dn	1	1	1	
ASL2A	Arithmetic shift left by 2 bits					AGU Arithmetic
	Rx<<2 → Rx	ASL2A Rx	1	1	2	
ASLA	Arithmetic shift left					AGU Arithmetic
	Rx<<1 → Rx	ASLA Rx	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
ASLL	Multi-bit arithmetic shift left					DALU Logical
	Dn << #u5 → Dn If Da[6:0] > 0, then Dn << Da[6:0] → Dn else Dn >> |Da[6:0]| → Dn	ASLL #u5,Dn {0 ≤ u5 < 32} ASLL Da,Dn {–40 ≤ Da[6:0] ≤ 40}	1 1	1 1	1 2	
ASLW	Word arithmetic shift left (16 bit shift)					DALU Logical
	Da<<16 → Dn	ASLW Da,Dn	1	1	2	
ASR	Arithmetic shift right by one bit					DALU Arithmetic
	Da>>1 → Dn	ASR Da,Dn	1	1	1	
ASRA	Arithmetic shift right					AGU Arithmetic
	(Rx>>1) → Rx	ASRA Rx	1	1	2	
ASRR	Multi-bit arithmetic shift right					DALU Logical
	Dn >> #u5 → Dn If Da[6:0] > 0, then Dn >> Da[6:0]→ Dn else Dn << |Da[6:0]| → Dn	ASRR #u5,Dn {0 ≤ u5 < 32} ASRR Da,Dn	1 1	1 1	1 2	
ASRW	Word arithmetic shift right (16 bit shift)					DALU Logical
	Da>>16 → Dn	ASRW Da,Dn	1	1	2	
BF	Branch if false (Branch not taken uses 1 cycle; taken uses 4.)					Change-of-Flow
	If T==0, then PC + displacement → PC	BF <label BF >label	1 2	1/4 1/4	4 4	
BFD	Branch if false (delayed) (Not taken: 1 cycle; taken: 4 minus time for delay slot.)					Change-of-Flow
	If T==0, then PC + displacement → PC	BFD <label BFD >label	1 2	1/4 1/4	4 4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
BMCHG	Bit-mask change a 16-bit operand					Bit-Mask
	$\sim C1.H_i \rightarrow C1.H_i$ (i denotes bits=1 in #u16)	BMCHG #u16,C1.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	$\sim C1.L_i \rightarrow C1.L_i$	BMCHG #u16,C1.L {$0 \leq u16 < 2^{16}$}	2	2	3	
	$\sim DR.H_i \rightarrow DR.H_i$	BMCHG #u16,DR.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	$\sim DR.L_i \rightarrow DR.L_i$	BMCHG #u16,DR.L {$0 \leq u16 < 2^{16}$}	2	2	3	
BMCHG.W	Bit-mask change a 16-bit operand in memory					Bit-Mask
	$\sim(SP - u5)_i \rightarrow (SP - u5)_i$ (i denotes bits=1 in #u16)	BMCHG.W #u16,(SP–u5) {$0 \leq u16 < 2^{16}$} {$0 \leq u5 < 64$,W}	2	3	3	
	$\sim(SP + s16)_i \rightarrow (SP+s16)_i$	BMCHG.W #u16,(SP+s16) {$0 \leq u16 < 2^{16}$} {$-2^{15} \leq s16 < 2^{15}$, W}	3	3	3	
	$\sim(Rn)_i \rightarrow (Rn)_i$	BMCHG.W #u16,(Rn) {$0 \leq u16 < 2^{16}$}	2	2	3	
	$\sim(a16)_i \rightarrow (a16)_i$	BMCHG.W #u16,(a16) {$0 \leq u16 < 2^{16}$} {$0 \leq a16 < 2^{16}$, W}	3	2	3	
BMCLR	Bit-mask clear a 16-bit operand					Bit-Mask
	$0 \rightarrow C1.H_i$ (i denotes bits=1 in #u16)	BMCLR #u16,C1.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	$0 \rightarrow C1.L_i$	BMCLR #u16,C1.L {$0 \leq u16 < 2^{16}$}	2	2	3	
	$0 \rightarrow DR.H_i$	BMCLR #u16,DR.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	$0 \rightarrow DR.L_i$	BMCLR #u16,DR.L {$0 \leq u16 < 2^{16}$}	2	2	3	
BMCLR.W	Bit-mask clear a 16-bit operand in memory					Bit-Mask
	$0 \rightarrow (SP - u5)_i$ (i denotes bits=1 in #u16)	BMCLR.W #u16,(SP–u5) {$0 \leq u16 < 2^{16}$} {$0 \leq u5 < 64$,W}	2	3	3	
	$0 \rightarrow (SP + s16)_i$	BMCLR.W #u16,(SP+s16) {$0 \leq u16 < 2^{16}$} {$-2^{15} \leq s16 < 2^{15}$, W}	3	3	3	
	$0 \rightarrow (Rn)_i$	BMCLR.W #u16,(Rn) {$0 \leq u16 < 2^{16}$}	2	2	3	
	$0 \rightarrow (a16)_i$	BMCLR.W #u16,(a16) {$0 \leq u16 < 2^{16}$} {$0 \leq a16 < 2^{16}$, W}	3	2	3	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
BMSET	Bit-mask set a 16-bit operand					Bit-Mask
	$1 \rightarrow C1.H_i$ (i denotes bits=1 in #u16)	BMSET #u16,C1.H {$0 < u16 < 2^{16}$}	2	2	3	
	$1 \rightarrow C1.L_i$	BMSET #u16,C1.L {$0 < u16 < 2^{16}$}	2	2	3	
	$1 \rightarrow DR.H_i$	BMSET #u16,DR.H {$0 < u16 < 2^{16}$}	2	2	3	
	$1 \rightarrow DR.L_i$	BMSET #u16,DR.L {$0 < u16 < 2^{16}$}	2	2	3	
BMTSET	Bit-mask test and set a 16-bit operand					Bit-Mask
	$1 \rightarrow DR.H_i$ (i denotes bits = 1 in #u16) if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET #u16,DR.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	$1 \rightarrow DR.L_i$ (selected bits) if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET #u16,DR.L {$0 \leq u16 < 2^{16}$}	2	2	3	
BMTSET.W	Bit-mask test and set a 16-bit operand in memory					Bit-Mask
	$1 \rightarrow (SP - u5)_i$ (i denotes bits = 1 in #u16) if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET.W #u16, (SP–u5) {$0 \leq u16 < 2^{16}$} {$0 \leq u5 < 64$,W}	2	3	3	
	$1 \rightarrow (SP + s16)_i$ if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET.W #u16, (SP+s16) {$0 \leq u16 < 2^{16}$ {$-2^{15} \leq s16 < 2^{15}$,W}	3	3	3	
	$1 \rightarrow (Rn)_i$ if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET.W #u16, (Rn) {$0 \leq u16 < 2^{16}$}	2	2	3	
	$1 \rightarrow (a16)_i$ if (all selected bits are set), then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSET.W #u16, (a16) {$0 \leq u16 < 2^{16}$} {$0 \leq a16 < 2^{16}$,W}	3	2	3	
BMTSTC	Bit-mask test if clear					Bit-Mask
	if (#u16 & C1.H) == $0000 then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSTC #u16,C1.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	if (#u16 & C1.L) == $0000, then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSTC #u16,C1.L {$0 \leq u16 < 2^{16}$}	2	2	3	
	if (#u16 & DR.H) == $0000, then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSTC #u16,DR.H {$0 \leq u16 < 2^{16}$}	2	2	3	
	if (#u16 & DR.L) == $0000, then $1 \rightarrow T$, else $0 \rightarrow T$	BMTSTC #u16,DR.L {$0 \leq u16 < 2^{16}$}	2	2	3	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
BMTSTC.W	Bit-mask test a 16-bit operand in memory if clear					Bit-Mask
	if (#u16 & (SP – u5)) == $0000, then 1→ T, else 0 → T	BMTSTC.W #u16,(SP–u5) {0 ≤ u16 < 2^{16}} {0 ≤ u5 < 64,W}	2	3	3	
	if (#u16 & (SP + s16)) == $0000, then 1→ T, else 0 → T	BMTSTC.W #u16,(SP+s16) {0 ≤ u16 < 2^{16}} {–2^{15} ≤ s16 < 2^{15}, W}	3	3	3	
	if (#u16 & (Rn)) == $0000, then 1→ T, else 0 → T	BMTSTC.W #u16,(Rn) {0 ≤ u16 < 2^{16}}	2	2	3	
	if (#u16 & (a16)) == $0000, then 1→ T, else 0 → T	BMTSTC.W #u16,(a16) {0 ≤ u16 < 2^{16}} {0 ≤ a16 < 2^{16}, W}	3	2	3	
BMTSTS	Bit-mask test if set					Bit-Mask
	if (#u16 & ~C1.H = $0000), then 1 → T, else 0 → T	BMTSTS #u16,C1.H {0 ≤ u16 < 2^{16}}	2	2	3	
	if (#u16 & ~C1.L = $0000), then 1 → T, else 0 → T	BMTSTS #u16,C1.L {0 ≤ u16 < 2^{16}}	2	2	3	
	if (#u16 & ~DR.H = $0000), then 1 → T, else 0 → T	BMTSTS #u16,DR.H {0 ≤ u16 < 2^{16}}	2	2	3	
	if (#u16 & ~DR.L = $0000), then 1 → T, else 0 → T	BMTSTS #u16,DR.L {0 ≤ u16 < 2^{16}}	2	2	3	
BMTSTS.W	Bit-mask test a 16-bit operand in memory					Bit-Mask
	if (#u16 & ~(SP – u5) = $0000), then 1→ T, else 0 → T	BMTSTS.W #u16,(SP–u5) {0 ≤ u16 < 2^{16}} {0 ≤ u5 < 64,W}	2	3	3	
	if (#u16 & ~(SP + s16) = $0000), then 1→ T, else 0 → T	BMTSTS.W #u16,(SP+s16) {0 ≤ u16 < 2^{16}} {–2^{15} ≤ s16 < 2^{15}, W}	3	3	3	
	if (#u16 & ~(Rn) = $0000), then 1→ T, else 0 → T	BMTSTS.W #u16,(Rn) {0 ≤ u16 < 2^{16}}	2	2	3	
	if (#u16 & ~(a16) = $0000), then 1→ T, else 0 → T	BMTSTS.W #u16,(a16) {0 ≤ u16 < 2^{16}} {0 ≤ a16 < 2^{16}, W}	3	2	3	
BRA	Branch					Change-of-Flow
	PC + displacement → PC	BRA <label	1	4	4	
		BRA >label	2	4	4	
BRAD	Branch (delayed) (4 cycles minus time for delay slot, but not less than 1 cycle.)					Change-of-Flow
	PC + displacement → PC	BRAD <label	1	4	4	
		BRAD >label	2	4	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
BREAK	Terminate the loop and branch to an address					Loop
	PC + displacement → PC 0 → LFn	BREAK label	2	4	4	
BSR	Branch to subroutine					Change-of-Flow
	(Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; PC + displacement → PC	BSR <label BSR >label	1 2	4 4	4 4	
BSRD	Branch to subroutine (delayed) (4 cycles minus time for delay slot, but not less than 2 cycles.)					Change-of-Flow
	(Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; PC + displacement → PC	BSRD <label BSRD >label	1 2	4 4	4 4	
BT	Branch if true (Not taken: 1 cycle; taken: 4.)					Change-of-Flow
	If T==1, then PC + displacement → PC	BT <label BT >label	1 2	1/4 1/4	4 4	
BTD	Branch if true (delayed) (4 cycles minus time for delay slot, but not less than 1 cycle.)					Change-of-Flow
	If T==1, then PC + displacement → PC	BTD <label BTD >label	1 2	1/4 1/4	4 4	
CLB	Count leading bits (ones or zeros)					DALU Logical
	If Da[39] == 0, then 9 – (number of consecutive leading zeros in Da[39:0]) → Dn else 9 – (number of consecutive leading ones in Da[39:0]) → Dn	CLB Da,Dn	1	1	2	
CLR	Clear					DALU Arithmetic
	0 → Dn	CLR Dn	1	1	1	
CMPEQ	Compare for equal					DALU Arithmetic
	If Da = = Dn, then 1 → T, else 0 → T	CMPEQ Da,Dn	1	1	1	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
CMPEQ.W	Compare for equal					DALU Arithmetic
	If #u5 = = Dn, then 1 → T, else 0 → T If #s16 = = Dn, then 1 → T, else 0 → T	CMPEQ.W #u5,Dn {0 ≤ u5 < 32} CMPEQ.W #s16,Dn {$-2^{15} \leq$ s16 < 2^{15}}	1 2	1 1	2 4	
CMPEQA	Compare for equal					AGU Arithmetic
	If rx == Rx, then 1 → T, else 0 → T	CMPEQA rx,Rx	1	1	2	
CMPGT	Compare for greater than					DALU Arithmetic
	Dn > Da → T	CMPTG Da,Dn	1	1	1	
CMPGT.W	Compare for greater than					DALU Arithmetic
	Dn > #u5 → T Dn > #s16 → T	CMPTG.W #u5,Dn {0 ≤ u5 < 32} CMPTG.W #s16,Dn {$-2^{15} \leq$ s16 < 2^{15}}	1 2	1 1	2 4	
CMPGTA	Compare for greater than					AGU Arithmetic
	Rx > rx → T	CMPGTA rx,Rx	1	1	2	
CMPHI	Compare for higher (unsigned)					DALU Arithmetic
	Dn > Da → T	CMPHI Da,Dn	1	1	1	
CMPHIA	Compare for higher (unsigned)					AGU Arithmetic
	Rx > rx → T	CMPHIA rx,Rx	1	1	2	
CONT	Continue to the next loop iteration	(LC >1: 3 cycles. LC ≤ 1: 4 cycles.)				Loop
	If LCn > 1, then SAn → PC else PC + displacement → PC 0 →LFn; LCn – 1 → LCn (always)	CONT label	2	3/4	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
CONTD	Jump to the start of the loop to start the next iteration (delayed) (LC >1: 3 cycles; LC = 1: 4 cycles.)					Loop
	If LCn > 1, then SAn → PC else PC + displacement → PC 0 →LFn; LCn – 1 → LCn (always)	CONTD label	2	3/4	4	
DEBUG	Enter debug mode					Program Control
		DEBUG	1	2	4	
DEBUGEV	Signal debug event					Program Control
		DEBUGEV	1	2	4	
DECA	Decrement register					AGU Arithmetic
	Rx – 1 → Rx	DECA Rx	1	1	2	
DECEQ	Decrement a data register and set T if zero					DALU Arithmetic
	Dn – 1 → Dn; if Dn == 0, then 1 → T, else 0 → T	DECEQ Dn	1	1	1	
DECEQA	Decrement and set T if zero					AGU Arithmetic
	Rx – 1 → Rx; if Rx ==0 → T, then 1 → T, else 0 → T	DECEQA Rx	1	1	2	
DECGE	Decrement a data register and set T if greater than or equal to zero					DALU Arithmetic
	Dn – 1 → Dn; Dn ≥ 0 → T	DECGE Dn	1	1	1	
DECGEA	Decrement and set T if greater than or equal to zero					AGU Arithmetic
	Rx – 1 → Rx; Rx ≥ 0 → T	DECGEA Rx	1	1	2	
DI	Disable interrupts (sets the DI bit in the status register)					Program Control
	1 → DI	DI	1	1	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
DIV	Divide iteration					DALU Arithmetic
	If Dn[39] ⊕ Da[39] = 1, then 2 * Dn + C + (Da & $FF FFFF 0000) → Dn else 2 * Dn + C – (Da & $FF FFFF 0000) → Dn where ⊕ denotes the bitwise exclusive OR operator	DIV Da,Dn	1	1	1	
DMACSS	Multiply signed by signed and accumulate with data register right shifted by word size					DALU Arithmetic
	[Dn>>16] + Da.H * Db.H → Dn (Da signed, Db signed)	DMACSS Da,Db,Dn	1	1	1	
DMACSU	Multiply signed by unsigned and accumulate with data register right shifted by word size					DALU Arithmetic
	[Dn>>16] + Da.H * Db.L → Dn (Da signed, Db unsigned)	DMACSU Da,Db,Dn	1	1	1	
DOENn	Do enable—set the loop counter n and enable loop n as a long loop					Loop
	#u6 → LCn; 1 → LFn #u16 → LCn; 1 → LFn DR → LCn; 1 → LFn	DOENn #u6 {0 ≤ u6 < 64} DOENn #u16 {0 ≤ u16 < 2^{16}} DOENn DR	1 2 1	1 1 1	4 4 4	
DOENSHn	Do enable short—set loop counter n and enable loop n as a short loop					Loop
	#u6 → LCn; 1 → LFn; 1 → SLF #u16 → LCn; 1 → LFn; 1 → SLF DR → LCn; 1 → LFn; 1 → SLF	DOENSHn #u6 {0 ≤ u6 < 64} DOENSHn #u16 {0 ≤ u16 < 2^{16}} DOENSHn DR	1 2 1	1 1 1	4 4 4	
DOSETUPn	Setup loop start address n					Loop
	PC + displacement → SAn	DOSETUPn label	3	1	4	
EI	Enable interrupts (clears the DI bit in the status register)					Program Control
	0 → DI	EI	1	1	4	
EOR	Bitwise exclusive OR					DALU Logical
	Da ⊕ Dn → Dn	DOR Da,Dn	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
EOR	Bitwise Exclusive OR on a 16-bit operand					Bit-Mask
	#u16 ⊕ DR.L → DR.L	EOR #u16,DR.L {0 ≤ u16 < 2^{16}}	2	2	3	
	#u16 ⊕ DR.H → DR.H	EOR #u16,DR.H {0 ≤ u16 < 2^{16}}	2	2	3	
EOR.W	Bitwise Exclusive OR on a 16-bit operand in memory					Bit-Mask
	#u16 ⊕ (R) → (R)	EOR.W #u16,(Rn) {0 ≤ u16 < 2^{16}}	2	2	3	
	#u16 ⊕ (SP – u5) → (SP – u5)	EOR.W #u16,(SP–u5) {0 ≤ u16 < 2^{16}} {0 ≤ u5 < 64, W}	2	3	3	
	#u16 ⊕ (SP + s16) → (SP + s16)	EOR.W #u16,(SP+s16) {0 ≤ u16 < 2^{16}} {–2^{15} ≤ s16 < 2^{15}, W}	3	3	3	
	#u16 ⊕ (a16) → (a16)	EOR.W #u16,(a16) {0 ≤ u16 < 2^{16}} {0 ≤ a16 < 2^{16}, W}	3	2	3	
EXTRACT	Extract signed bit field					DALU Logical
	Db[(offset + width – 1):offset] → Dn[(width – 1):0] Db[offset + width – 1] → Dn[39:width] (sign-extension) width = #U6; offset = #u6	EXTRACT #U6,#u6,Db,Dn {0 ≤ U6 ≤ 40} {0 ≤ u6 ≤ 40} {#U6+#u6 ≤ 40}	2	1	4	
	width = Da[13:8]; offset = Da[5:0]	EXTRACT Da,Db,Dn {0 ≤ Da [13:8] ≤ 40} {0 ≤ Da [5:0] ≤ 40} {Da [13:8] +Da[5:0] ≤ 40}	2	1	4	
EXTRACTU	Extract unsigned bit field					DALU Logical
	Db[(offset + width – 1):offset] → Dn[(width – 1):0] 0 → Dn[39:width] width = #U6; offset = #u6	EXTRACTU #U6,#u6,Db,Dn {0 ≤ U6 ≤ 40} {0 ≤ u6 ≤ 40} {#U6+#u6 ≤ 40}	2	1	4	
	width = Da[13:8]; offset = Da[5:0]	EXTRACTU Da,Db,Dn {0 ≤ Da [13:8] ≤ 40} {0 ≤ Da [5:0] ≤ 40} {Da [13:8] +Da[5:0] ≤ 40}	2	1	4	
IADD	Add integers (not affected by saturation)					DALU Arithmetic
	Da + Dn → Dn	IADD Da,Dn	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
IFc	Execute current execution set or subgroup unconditionally					Program Control
	IF T == 0, then execute set/subgroup else treat as NOP	IFF	1	1	4	
	IF T == 1, then execute set/subgroup else treat as NOP	IFT	2	1	4	
	execute set/subgroup	IFA				
ILLEGAL	Generate an illegal exception request (Cycle count depends upon machine state.)					Program Control
	upon service: PC → (ESP); SR → (ESP + 4); SP + 8 → SP; {VBA[31:12]:illegal_vector} → PC; 1 → EXP 111→ I[2:0] 1 → ILN 0 → C 0 → T 00 → S[1:0] 0 → SLF 0000 → LF[3:0]	ILLEGAL {illegal vector = $080}	1	4	5	
IMAC	Multiply-accumulate integers					DALU Arithmetic
	Dn ± (Da.L * Db.L) → Dn	IMAC ±Da,Db,Dn	1	1	1	
IMACLHUU	Multiply-accumulate unsigned integers; first source from low portion, second from high portion					DALU Arithmetic
	Dn + (Da.L * Db.H) → Dn	IMACLHUU Da,Db,Dn	2	1	4	
IMACUS	Multiply-accumulate unsigned integer and signed integer					DALU Arithmetic
	Dn + (Da.L * Db.H) → Dn	IMACUS Da,Db,Dn	2	1	4	
IMPY	Multiply integer					DALU Arithmetic
	Da.L * Db.L → Dn	IMPY Da,Db,Dn	1	1	1	
IMPY.W	Multiply integer signed immediate					DALU Arithmetic
	#s16 * Dn.L → Dn	IMPY.W #s16,Dn {$-2^{15} \leq s16 < 2^{15}$}	2	1	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
IMPYHLUU	Multiply unsigned integer and unsigned integer; first source from high portion, second from low portion					DALU Arithmetic
	Da.H * Db.L → Dn	IMPHYLUU Da,Db,Dn	2	1	4	
IMPYSU	Multiply signed integer and unsigned integer					DALU Arithmetic
	Da.H * Db.L → Dn	IMPYSU Da,Db,Dn	2	1	4	
IMPYUU	Multiply unsigned integer and unsigned integer					DALU Arithmetic
	Da.L * Db.L → Dn	IMPYUU Da,Db,Dn	2	1	4	
INC	Increment a data register (as integer data)					DALU Arithmetic
	Dn + 1 → Dn	INC Dn	1	1	1	
INC.F	Increment a data register (as fractional data)					DALU Arithmetic
	Dn + $00:00010000 → Dn	INC.F Dn	1	1	1	
INCA	Increment register					AGU Arithmetic
	Rx + 1 → Rx	INCA Rx	1	1	2	
INSERT	Insert bit field					DALU Logical
	Db[(width – 1):0] → Dn[(offset + width – 1):offset] width = #U6; offset = #u6 width = Da[13:8]; offset = Da[5:0]	 INSERT #U6,#u6,Db,Dn {0 ≤ U6 ≤ 40} {0 ≤ u6 ≤ 40} [#U6 + #u6 ≤ 40] INSERT Da,Db,Dn {0 ≤ Da[5:0] ≤ 40} {0 ≤ Da[13:8] ≤ 16} {Da[13:8] + Da[5:0] ≤ 40}	 2 2	 1 1	 4 4	
JF	Jump if false (Not taken: 1 cycle; taken: 4 cycles)					Change-of-Flow
	If T==0, then label → PC If T==0, then Rn → PC	JF label {0 ≤ label < 2^{32},W} JF Rn	3 1	1/4 1/4	3 4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
JFD	Jump if false (delayed) (4 cycles minus time for delay slot, but not less than 1 cycle)					Change-of-Flow
	If T==0, then label → PC If T==0, then Rn → PC	JFD label [0 ≤ label < 2^{32},W] JFD Rn	3 1	1/4 1/4	3 4	
JMP	Jump					Change-of-Flow
	label → PC Rn → PC	JMP label {0 ≤ label < 2^{32},W} JMP Rn	3 1	3 3	3 4	
JMPD	Jump (delayed) (3 cycles minus time for delay slot, but not less than 1 cycle)					Change-of-Flow
	label → PC Rn → PC	JMPD label {0 ≤ label < 2^{32},W} JMP Rn	3 1	3 3	3 4	
JSR	Jump to subroutine					Change-of-Flow
	(Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; label → PC (Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; Rn → PC	JSR label {0 ≤ label < 2^{32},W} JSR Rn	3 1	3/4 3/4	3 4	
JSRD	Jump to subroutine (delayed) (Normal: 2 cycles. 3 if cycles of set in delay slots + cycles of longest instruction with JSRD ≥ 3 cycles)					Change-of-Flow
	(Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; label → PC (Next PC) → (SP); SR → (SP + 4); SP + 8 → SP; Rn → PC	JSRD label {0 ≤ label < 2^{32},W} JSRD Rn	3 1	2/3 2/3	3 4	
JT	Jump if true (Not taken: 1 cycle; taken: 4cycles)					Change-of-Flow
	If T=1, then label → PC If T=1, then Rn → PC	JT label {0 ≤ label < 2^{32},W} JT Rn	3 1	1/4 1/4	3 4	
JTD	Jump if true (delayed) (Not taken: 1 cycle; taken: 4 minus time for delay slot.)					Change-of-Flow
	If T=1, then label → PC If T=1, then Rn → PC	JTD label {0 ≤ label < 2^{32},W} JTD Rn	3 1	1/4 1/4	3 4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
LSLL	Multi-bit bitwise shift left					DALU Logical
	If Da[6:0] > 0, then Dn << Da[6:0] → Dn else Dn >>> |Da[6:0]| → Dn	LSLL Da,Dn {-40 ≤ Da[6:0] < 40}	1	1	1	
LSR	Bitwise shift right by one bit					DALU Logical
	(Dn>>>1) → Dn; 0 → Dn[39]	LSR Dn	1	1	1	
LSRA	Bitwise shift right by one bit					AGU Arithmetic
	(Rx>>>1) → Rx; 0 → Rx[31]	LSRA Rx	1	1	2	
LSRR	Multi-bit bitwise shift right					DALU Logical
	If Da[6:0] > 0, then Dn>>>Da→ Dn else Dn << |Da| → Dn	LSRR Da,Dn {–40 ≤ Da[6:0] ≤40|	1	1	2	
	Dn >> #u5 → Dn	LSRR #u5,Dn {0 ≤ u5 < 32}	1	1	2	
LSRW	Word bitwise shift right (16-bit shift)					DALU Logical
	Da>>>16 → Dn	LSRW Da,Dn	1	1	2	
MAC	Multiply-accumulate signed fractions					DALU Arithmetic
	Dn + (#s16 * Da.H) → Dn Dn ± (Da.H * Db.H) → Dn	MAC #s16,Da,Dn {$-2^{15} \le s16 < 2^{15}$} MAC ±Da,Db,Dn	2 1	1 1	4 1	
MACR	Multiply-accumulate signed fractions and round					DALU Arithmetic
	Rnd(Dn ± (Da.H * Db.H)) → Dn	MACR ±Da,Db,Dn	1	1	1	
MACSU	Multiply-accumulate signed fraction and unsigned fraction					DALU Arithmetic
	Dn + (Da.H * Db.L) → Dn	MACSU Da,Db,Dn	1	1	1	
MACUS	Multiply-accumulate unsigned fraction and signed fraction					DALU Arithmetic
	Dn + (Da.L * Db.H) → Dn	MACUS Da,Db,Dn	1	1	1	

Table 9-5. Instructions Grouped Alphabetically (Continued)

<table>
<tr><th rowspan="2">Mnemomic</th><th colspan="2">Description</th><th rowspan="2">Words</th><th rowspan="2">Cycles</th><th rowspan="2">Type</th><th rowspan="2">Functional Group</th></tr>
<tr><th>Operation</th><th>Assembler Syntax</th></tr>
<tr><td rowspan="2">MACUU</td><td colspan="2">Multiply-accumulate unsigned fraction and unsigned fraction</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>Dn + (Da.L * Db.L) → Dn</td><td>MACUU Da,Db,Dn</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td rowspan="2">MARK</td><td colspan="2">Push the PC into the trace buffer</td><td></td><td></td><td></td><td rowspan="2">Program Control</td></tr>
<tr><td>PC → trace buffer</td><td>MARK</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td rowspan="2">MAX</td><td colspan="2">Transfer maximum signed value</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>If Da > Db, then Da → Db</td><td>MAX Da,Db</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td rowspan="2">MAX2</td><td colspan="2">Transfer two 16-bit maximum signed value</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>If Da.H > Db.H, then Da.H → Db.H
If Da.L > Db.L, then Da.L → Db.L</td><td>MAX2 Da,Db</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td rowspan="2">MAX2VIT</td><td colspan="2">Special MAX2 version for Viterbi kernel</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>If Da.L > Db.L, then 0 → VFn, Da.L → Db.L
else 1 → VFn
<table>
<tr><th>Da</th><th>Db</th><th>VFn</th><th>Da</th><th>Db</th><th>VFn</th></tr>
<tr><td>D4.L</td><td>D2.L</td><td>VF0</td><td>D12.L</td><td>D10.L</td><td>VF0</td></tr>
<tr><td>D4.H</td><td>D2.H</td><td>VF1</td><td>D12.H</td><td>D10.H</td><td>VF1</td></tr>
<tr><td>D0.L</td><td>D6.L</td><td>VF2</td><td>D8.L</td><td>D14.L</td><td>VF2</td></tr>
<tr><td>D0.H</td><td>D6.H</td><td>VF3</td><td>D8.H</td><td>D14.H</td><td>VF3</td></tr>
</table></td><td>MAX2VIT Da,Db</td><td>1</td><td>1</td><td>2</td></tr>
<tr><td rowspan="2">MAXM</td><td colspan="2">Transfer maximum magnitude value</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>If |Da| > |Db|, then Da → Db
If Da == –Db, then |Da| → Db</td><td>MAXM Da,Db</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td rowspan="2">MIN</td><td colspan="2">Transfer minimum signed value</td><td></td><td></td><td></td><td rowspan="2">DALU Arithmetic</td></tr>
<tr><td>If Da < Db, then Da → Db</td><td>MIN Da,Db</td><td>1</td><td>1</td><td>1</td></tr>
</table>

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MOVE.2F	Move two fractional words from memory to a register pair 39 32 16 0 Da: SIGN EXTENSION | **(EA) OPERAND** | ZERO FILL Db: SIGN EXTENSION | **(EA+2) OPERAND** | ZERO FILL					Move
	(EA) → Da:Db	MOVE.2F (EA),Da:Db {0 ≤ EA < 2^{32},L}	1	1^1	1	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVE.2L	Move two integer longs to/from a register pair 39 32 0 Da: SIGN EXTENSION | **(EA)** Db: SIGN EXTENSION | **(EA+4)**					Move
	Da,Db ↔ (EA	MOVE.2L Da:Db, (EA) {0 ≤ EA < 2^{32},Q} MOVE.2L (EA),Da:Db {0 ≤ EA < 2^{32},Q}	1	1^1	2	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVE.2W	Move two integer words to/from a register pair 39 16 0 Da: SIGN EXTENSION | **(EA)** Db: SIGN EXTENSION | **(EA+2)**					Move
	(EA) ↔ Da:Db	MOVE.2W (EA),Da:Db {0 ≤ EA < 2^{32},L} MOVE.2W Da:Db, (EA) {0 ≤ EA < 2^{32},L}	1	1^1	1	
Notes: 1. Add one cycle when EA = (Rn + N0).						

Table 9-5. Instructions Grouped Alphabetically (Continued)

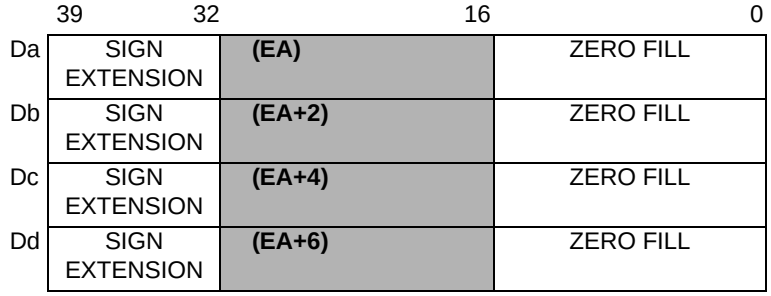
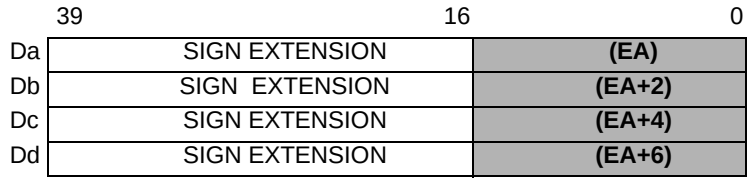
<table>
<tr><th rowspan="2">Mnemomic</th><th colspan="2">Description</th><th rowspan="2">Words</th><th rowspan="2">Cycles</th><th rowspan="2">Type</th><th rowspan="2">Functional Group</th></tr>
<tr><th>Operation</th><th>Assembler Syntax</th></tr>
<tr><td>MOVE.4F</td><td colspan="2">Move four fractional words from memory to a register quad
 
</td><td></td><td></td><td></td><td>Move</td></tr>
<tr><td></td><td>(EA) → Da:Db:Dc:Dd</td><td>MOVE.4F (EA),Da:Db:Dc:Dd {0 ≤ EA 2^{32},Q)</td><td>1</td><td>1^{1}</td><td>1</td><td></td></tr>
<tr><td colspan="7">Notes: 1. Add one cycle when EA = (Rn + N0).</td></tr>
<tr><td>MOVE.4W</td><td colspan="2">Move four integer words to/from a register quadrant
 
</td><td></td><td></td><td></td><td>Move</td></tr>
<tr><td></td><td>(EA) ↔ Da:Db:Dc:Dd</td><td>MOVE.4W (EA) ,Da:Db:Dc:Dd
{0 ≤ EA < 2^{32},Q}
MOVE.4W Da:Db:Dc:Dd,
(EA) {0 ≤ EA < 2^{32},Q}</td><td>1</td><td>2^{1}</td><td>1</td><td></td></tr>
<tr><td colspan="7">Notes: 1. Add one cycle when EA = (Rn + N0).</td></tr>
</table>

Table 9-5. Instructions Grouped Alphabetically (Continued)

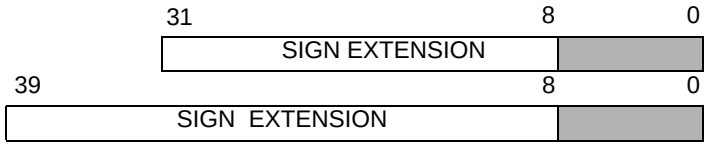
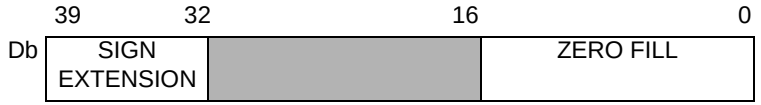
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
MOVE.B	Move byte to/from memory  					Move
	(aa) ↔ DR	MOVE.B (a16),DR {$0 \le a16 < 2^{16}$} MOVE.B DR, (a16) {$0 \le a16 < 2^{16}$}	2	1	3	
	DR → (aa)	MOVE.B Dr, (a32) {$0 \le a32 < 2^{32}$}	3	1	3	
	(Rn + s15) → DR	MOVE.B DR, (Rn+s15) {$-2^{14} \le s15 < 2^{14}$}	2	2	3	
	(ea) ↔ DR	MOVE.B (ea) ,DR MOVE.B DR, (ea)	1	1[1]	4	
	(SP + s15) ↔ DR	MOVE.B (SP+s15) ,DR {$-2^{14} \le s15 < 2^{14}$} MOVE.B DR, (SP+s15) {$-2^{14} \le s15 < 2^{14}$}	2	2	3	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVE.F	Move fractional word to/from memory  					Move
	#s16 → Db	MOVE.F #s16,Db {$-2^{15} \le s16 < 2^{15}$}	2	1	4	
	(aa) → Db	MOVE.F (a16),Db {$0 \le a16 < 2^{16}$,W}	2	1	3	
	(aa) → Db	MOVE.F (a32),Db {$0 \le a32 < 2^{32}$,W}	3	1	3	
	(EA) → Db	MOVE.F (EA),Db {$0 \le EA < 2^{32}$,W}	1	1[1]	1	
	(Rn + s15) → Db	MOVE.F (Rn+s15),Db {$-2^{14} \le s15 < 2^{14}$,W}	2	2	3	
	(SP + s15) → Db	MOVE.F (Sp+s15),Db {$-2^{14} \le s15 < 2^{14}$,W}	2	2	3	
	Db → (ea)	MOVE.F Db, (ea) {$0 \le ea < 2^{32}$,W}	1	1[1]	4	
Notes: 1. Add one cycle when EA = (Rn + N0).						

Table 9-5. Instructions Grouped Alphabetically (Continued)

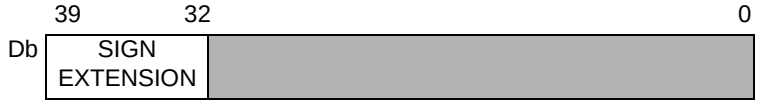
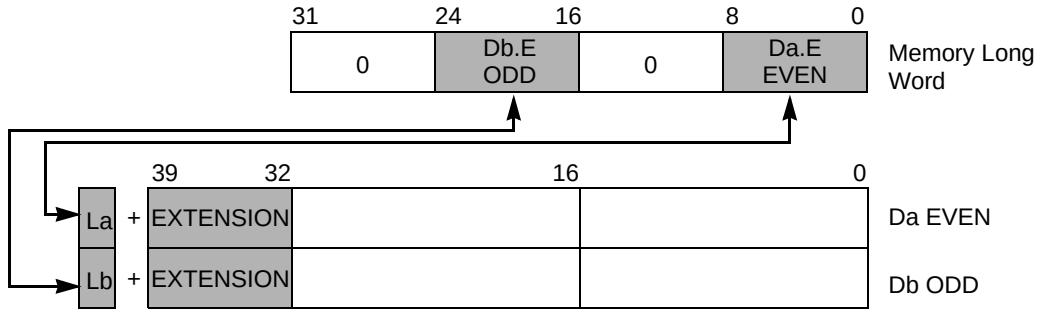
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MOVE.L	Move long word 					Move
	#s32 → C4	MOVE.L #s32,C4 {$-2^{31} \le s32 < 2^{31}$}	3	1	3	
	#u32 → C1	MOVE.L #u32,C1 {$0 \le u32 < 2^{32}$}	3	1	3	
	C4 ↔ Db	MOVE.L C4,Db MOVE.L Db,C4	1	1	2	
	C2 ↔ Db	MOVE.L C2,Db MOVE.L Db,C2	1	1	2	
MOVE.L	Move long register extensions  Note: Moves of extensions into data registers restore the corresponding limit tag bit (Ln bit) in the destination register.					Move
	((SP + s15)[8:0]) → De.E	MOVE.L (SP+s15),De.E {$-2^{14} \le s15 < 2^{14}$,L}	2	2	3	
	Da.E:Db.E → (SP + s15)	MOVE.L Da.E:Db.E,(SP+s15) {$-2^{14} \le s15 < 2^{14}$,L}	2	2	3	
	((SP + s15)[24:16]) → Do.E	MOVE.L (SP+s15),Do.E {$-2^{14} \le s15 < 2^{14}$,L}	2	2	3	
	(aa[8:0]) ↔ De.E	MOVE.L (a32),De.E {$0 \le a32 < 2^{32}$,L}	3	1	3	
	Da.E:Db.E → (aa)	MOVE.L Da.E:Db.E,(a32) {$0 \le a32 < 2^{32}$,L}	3	1	3	
	(aa[24:16]) → Do.E	MOVE.L (a32),Do.E {$0 \le a32 < 2^{32}$,L}	3	1	3	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MOVE.L	Move long 					Move
	(aa) ↔ DR	MOVE.L (a32),DR {0 ≤ a32 < 2^{32},L} MOVE.L DR, (a32)	3	1	3	
	(aa) ↔ C4	MOVE.L (a16),C4 {0 ≤ a16 < 2^{16},L} MOVE.L C4,(a16)	2	1	3	
	(Rn + u3) ↔ DR	MOVE.L (Rn+u3),DR {0 ≤ u3 < 32,L} MOVE.L DR, (Rn+u3)	1	2	4	
	(Rn + s15) ↔ DR	MOVE.L (Rn+s15),DR {–2^{14} ≤ s15 < 2^{14},L} MOVE.L DR, (Rn+s15)	2	2	3	
	(Rn + Rr) ↔ DR	MOVE.L (Rn+Rr),DR MOVE.L DR, (Rn+Rr)	1	2	4	
	(EA) ↔ DR	MOVE.L (EA) ,DR MOVE.L DR, (EA)	1	1^{1}	1	
	(Rn) ↔ C3	MOVE.L (Rn) ,C3 MOVE.L C3, (Rn)	1	1	4	
	(SP–u6) ↔ DR	MOVE.L (SP–u6),DR {0 ≤ u6 < 256,L} MOVE.L DR, (SP–u6)	1	2	2	
	(SP+s15) ↔ C4	MOVE.L (SP+s15),C4 {–2^{14} ≤ s15 < 2^{14},L} MOVE.L C4, (SP+s15)	2	2	3	

Notes: 1. Add one cycle when EA = (Rn + N0).

Table 9-5. Instructions Grouped Alphabetically (Continued)

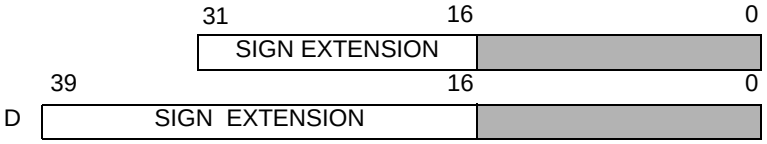
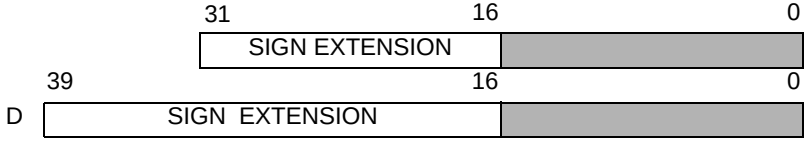
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
MOVE.W	Move integer word to/from memory, or immediate to register or memory 					Move
	#s7 → DR	MOVE.W #s7,DR {–63 ≤ s7 < 63}	1	1	2	
	#s16 → C4	MOVE.W #s16,C4 {$-2^{15} \le s16 < 2^{15}$}	2	1	4	
	#s16 → (aa)	MOVE.W #s16,(a16) {$-2^{15} \le s16 < 2^{15}$} {0 _ a16 < 2^{16},W}	3	1	3	
	#s16 → (SP–u5)	MOVE.W #s16,(SP–u5) {$-2^{15} \le s16 < 2^{15}$} {0 ≤ u5 < 64,W]	2	2	3	
	#s16 → (Rn)	MOVE.W #s16,(Rn) {$-2^{15} \le s16 < 2^{15}$}	2	1	3	
	#s16 → (SP+sa16)	MOVE.W #s16,(SP+sa16) {$-2^{15} \le s16 < 2^{15}$} {$-2^{15} \le sa16 < 2^{15}$,W}	3	1	3	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
MOVE.W	Move immediate integer word (sign extended for memory reads)  					Move
	(aa) ↔ DR	MOVE.W (a32),DR {0 ≤ a32 < 2^{32},W} MOVE.W DR,(a32)	3	1	3	
	(aa) ↔ C4	MOVE.W (a16),C4 {0 ≤ a16 < 2^{16},W} MOVE.W C4,(a16)	2	1	3	
	(Rn + u3) ↔ DR	MOVE.W (Rn+u3),DR {0 ≤ u3 < 16,W} MOVE.W DR,(Rn+u3)	1	2	4	
	(Rn + s15) ↔ DR	MOVE.W (Rn+s15),DR {–2^{14} ≤ s15 <2^{14},W} MOVE.W DR,(Rn+s15)	2	2	3	
	(Rn + Rr) ↔ DR	MOVE.W (Rn+Rr),DR MOVE.W DR,(Rn+Rr)	1	2	4	
	(EA) ↔ DR	MOVE.W (EA),DR MOVE.W DR, (EA)	1	1	1	
	(Rn) ↔ C3	MOVE.W (Rn),C3 MOVE.W C3,(Rn)	1	1	4	
	(SP – u6) ↔ DR	MOVE.W (SP–u6),DR {0 ≤ u6 < 128,W} MOVE.W DR,{SP–u6}	1	2	2	
	(SP + s15) ↔ DR	MOVE.W (SP+s15),C4 {–2^{14} ≤ s15 <2^{14},W} MOVE.W C4,{SP+s15)	2	2	3	
MOVEc	Conditional address register move, depending on T bit of SR					Move
	If T=1, then Rr → Rn If T=0, then Rr → Rn	MOVET Rr,Rn MOVEF Rr,Rn	1 1	1 1	4 4	
MOVES.2F	Move two fractional words to memory with scaling and saturation					Move
	Da:Db → (EA)	MOVES.2F Da:Db, (EA)	1	1[1]	1	

Notes: 1. Add one cycle when EA = (Rn + N0).

Table 9-5. Instructions Grouped Alphabetically (Continued)

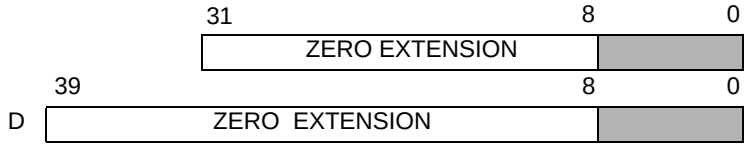
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MOVES.4F	Move four fractional words to memory with scaling and saturation					Move
	Da:Db:Dc:Dd → (EA)	MOVES.4F Da:Db:Dc:Dd, (EA)	1	1[1]	1	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVES.F	Move fractional word to memory with saturation enabled					Move
	Db → (aa) Db → (aa) Db → (Rn + s15) Db → (EA) Db → (SP + s15)	MOVES.F Db, (a16) {0 ≤ a16 < 2^{16},W} MOVES.F Db, (a32) {0 ≤ a32 < 2^{32},W} MOVES.F Db, (Rn+s15) {–2^{14} ≤ s15 < 2^{14},W} MOVES.F Db, (EA) MOVES.F Db, (SP+s15) {–2^{14} ≤ s15 < 2^{14},W}	2 3 2 1 2	1 1 2 1[1] 2	3 3 3 1 3	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVES.L	Move long to memory with scaling and saturation					Move
	Db → (EA)	MOVES.L Db, (EA)	1	1[1]	1	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MOVEU.B	Move unsigned byte from memory 					Move
	(aa) → DR (aa) → DR (Rn + s15) → DR (ea) → DR (SP + s15) → DR	MOVEU.B (a16) ,DR {0 ≤ a16 < 2^{16}} MOBEU.B (a32) ,DR {0 ≤ a32 < 2^{32}} MOVEU.B (Rn+s15) ,DR {–2^{14} ≤ s15 < 2^{14}} MOVEU.B (ea) ,DR MOBEU.B (SP+s15) ,DR {–2^{14} ≤ s15 < 2^{14}}	2 3 2 1 2	1 1 2 1[1] 2	3 3 3 4 3	
Notes: 1. Add one cycle when EA = (Rn + N0).						

Table 9-5. Instructions Grouped Alphabetically (Continued)

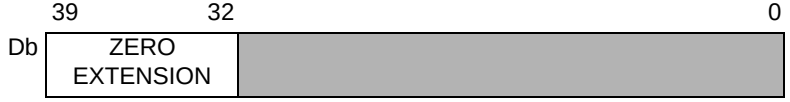
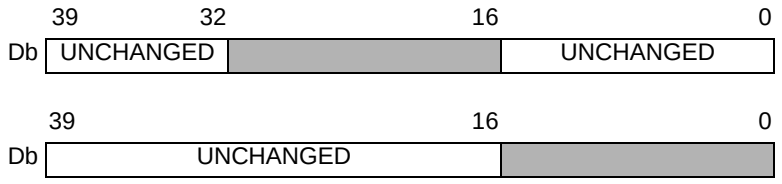
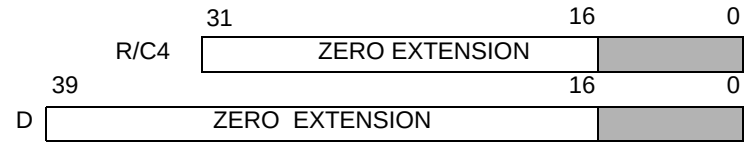
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MOVEU.L	Move unsigned long from memory 					Move
	#u32 → Db	MOVEU.L #u32,Db {0 ≤ u32 < 2^{32}}	3	1	3	
MOVEU.W	Move unsigned integer word from memory 					Move
	#u16 → Db[31:16] #u16 → Db[15:0]	MOVEU.W #u16,Db.H {0 ≤ u16 < 2^{16}} MOVEU.W #u16,Db.L {0 ≤ u16 < 2^{16}}	2 2	1 1	3 3	
MOVEU.W	Move unsigned word from memory to a register 					Move
	(aa) → C4 (aa) → DR (Rn + s15) → DR (EA) → DR (SP + s15) → C4	MOVEU.W (a16),C4 {0 ≤ a16 < 2^{16}} MOBEU.W (a32),DR {0 ≤ a32 < 2^{32}} MOVEU.W (Rn+s15),DR {–2^{14} ≤ s15 < 2^{14}} MOVEU.W (EA),DR MOVEU.W (SP+s15),C4 {–2^{14} ≤ s15 < 2^{14}}	2 3 2 1 2	1 1 2 1[1] 2	3 3 3 1 3	
Notes: 1. Add one cycle when EA = (Rn + N0).						
MPY	Multiply signed fractions					DALU Arithmetic
	Da.H * Db.H → Dn	MPY Da,Db,Dn	1	1	1	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
MPYR	Multiply signed fractions and round					DALU Arithmetic
	Rnd((Da.H * Db.H)) → Dn	MPYR Da,Db,Dn	1	1	1	
MPYSU	Multiply signed fraction and unsigned fraction					DALU Arithmetic
	Da.H * Db.L → Dn	MPYSU Da,Db,Dn	1	1	1	
MPYUS	Multiply unsigned fraction and signed fraction					DALU Arithmetic
	Da.L * Db.H → Dn	MPYUS Da,Db,Dn	1	1	1	
MPYUU	Multiply unsigned fraction and unsigned fraction					DALU Arithmetic
	Da.L * Db.L → Dn	MPYUU Da,Db,Dn	1	1	1	
NEG	Negate					DALU Arithmetic
	0 – Dn → Dn	NEG Dn	1	1	1	
NOP	No operation					Program Control
		NOP	1	1	4	
NOT	Bitwise Complement					DALU Logical
	~Da → Dn	NOT Da,Dn	1	1	2	
NOT	Binary inversion of a 16-bit operand					Bit-Mask
	~DR.L → DR.L ~DR.H → DR.H	NOT DR.L NOT DR.H	2 2	2 2	3 3	
NOT.W	Binary inversion of a 16-bit operand in memory					Bit-Mask
	~R → (R) ~(SP – u5) → (SP – u5) ~(SP + s16) → (SP + s16) ~(a16) → (a16)	NOT.W (Rn) NOT.W (SP–u5) {$0 \leq u5 < 64$,W} NOT.W (SP+s16) {$-2^{15} \leq s16 < 2^{15}$,W} NOT.W (a16) {$0 \leq a16 < 2^{16}$, W}	2 2 3 3	2 3 3 2	3 3 3 3	
OR	Bitwise inclusive OR					DALU Logical
	Da | Dn → Dn	OR Da,Dn	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

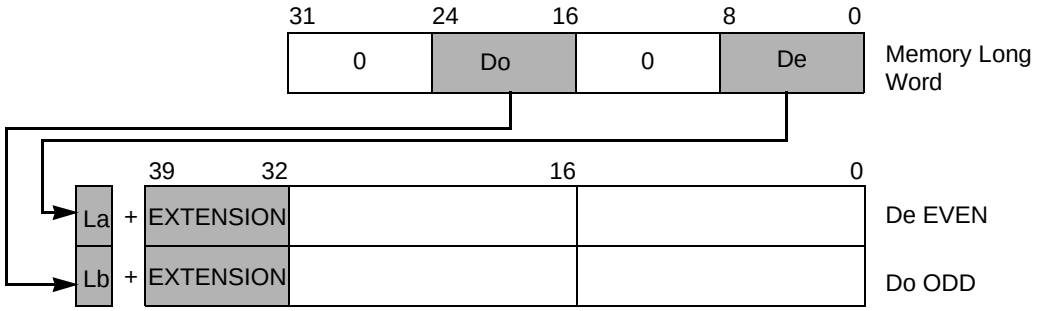
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
OR	Bitwise OR on a 16-bit operand					Bit-Mask
	#u16 | DR.L → DR.L #u16 | DR.H → DR.H	OR #u16,DR.L {0 ≤ u16 < 2^{16}} OR #u16,DR.H {0 ≤ u16 < 2^{16}}	2 2	2 2	3 3	
OR.W	Bitwise OR on a 16-bit operand in memory					Bit-Mask
	#u16 | R → (R) #u16 | (SP – u5) → (SP – u5) #u16 | (SP + s16) → (SP + s16) #u16 | (a16) → (a16)	OR.W #u16,(Rn) {0 ≤ u16 < 2^{16}} OR.W #u16,(SP–u5) {0 ≤ u16 < 2^{16}} {0 ≤ u5 < 64,W} OR.W #u16,(SP+s16) {0 ≤ u16 < 2^{16}} {–2^{15} ≤ s16 < 2^{15},W} OR.W #u16, (a16) {0 ≤ u16 < 2^{16}} {0 ≤ a16 < 2^{16},W}	2 2 3 3	2 3 3 2	3 3 3 3	
POP	Pop a register from the software stack 					Stack Support
	(SP – 8) → De; SP – 8 → SP (SP – 4) → Do; SP – 8 → SP	POP De POP Do	1 1	1 1	4 4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

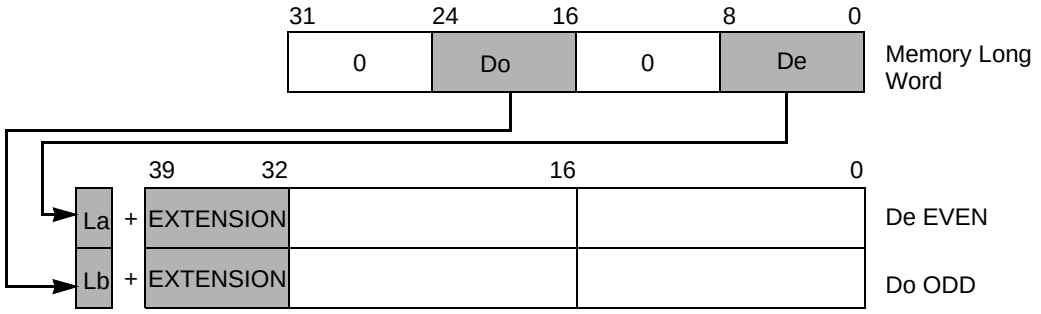
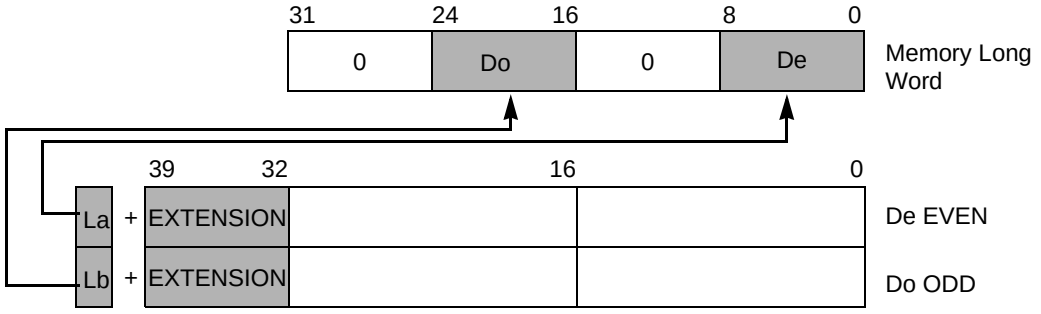
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
POPN	Pop a register from the software stack using the normal stack pointer 					Stack Support
	(NSP – 8) → De; NSP – 8 → NSP (NSP – 4) → Do; NSP – 8 → NSP	POPN De POPN Do	1 1	1 1	4 4	
PUSH	Push a register onto the software stack 					Stack Support
	De → (SP); SP + 8 → SP Do → (SP + 4); SP + 8 → SP	PUSH De PUSH Do	1 1	1 1	4 4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

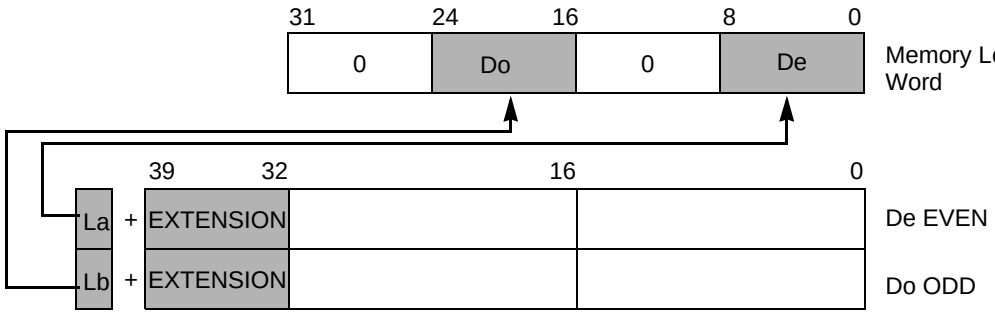
Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
PUSHN	Push a register onto the software stack using the normal stack pointer  					Stack Support
	De → (NSP); NSP + 8 → NSP Do → (NSP + 4); NSP + 8 → NSP	PUSHN De PUSHN Do	1 1	1 1	4 4	
RND	Round					DALU Arithmetic
	Rnd(Da) → Dn	RND Da,Dn	1	1	1	
ROL	Rotate one bit left through the carry bit					DALU Logical
	(Dn[38:0] <<1) → Dn[39:1] Dn[39] → C C → Dn[0]	ROL Dn	1	1	1	
ROR	Rotate one bit right through the carry bit					DALU Logical
	(Dn[39:1]>>>1) → Dn[38:0] C → Dn[39] Dn[0] → C	ROR Dn	1	1	1	
RTE	Return from exception (Shadow SP valid: 5 cycles. Not valid: 6 cycles.)					Change-of-Flow
	(SP – 8) → PC (SP – 4) → SR SP – 8 → SP 0 → NMID	RTE	1	5/6	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
RTED	Return from exception (delayed) (Shadow SP valid: 5 cycles. Not valid: 6 cycles.)					Change-of-Flow
	(SP – 8) → PC (SP – 4) → SR SP – 8 → SP 0 → NMID	RTED	1	5/6	4	
RTS	Return from subroutine (RAS valid: 3 cycles; RAS not valid, shadow SP valid: 5 cycles ;RAS and shadow SP not valid: 6 cycles.)					Change-of-Flow
	If (RAS valid), then RAS → PC; else (SP – 8) → PC; always SP – 8 → SP	RTS	1	3/5/6	4	
RTSD	Return from subroutine (delayed) (RAS valid: 3 cycles; RAS not valid, shadow SP valid: 5 cycles; RAS and shadow SP not valid: 6 cycles.)					Change-of-Flow
	If (RAS valid), then RAS → PC; else (SP – 8) → PC; always SP – 8 → SP	RTSD	1	3/5/6	4	
RTSTK	Force restore PC from the stack, updating SP (Shadow SP valid: 5 cycles; not valid: 6 cycles.)					Change-of-Flow
	(SP – 8) → PC SP – 8 → SP 0 → NMID	RTSTK	1	5/6	4	
RTSTKD	Force restore PC from the stack, updating SP (delayed) (Shadow SP valid: 5 cycles; not valid: 6 cycles.)					Change-of-Flow
	(SP – 8) → PC SP – 8 → SP 0 → NMID	RTSKD	1	5/6	4	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	Assembler Syntax				
SAT.F	Saturate fractional value in data register to fit in high portion					DALU Arithmetic
	If Da > $007FFFFFFF then $007FFF0000 → Dn If Da < $FF80000000 then $FF80000000 → Dn Else Da & $FFFFFF0000 → Dn	SAT.F Da,Dn	1	1	1	
SAT.L	Saturate value in data register to fit in 32 bits					DALU Arithmetic
	If Da > $007FFFFFFF then $007FFF0000 → Dn If Da < $FF80000000 then $FF80000000 → Dn Else Dn → Dn	SAT.L Dn	1	1	1	
SBC	Subtract long with carry					DALU Arithmetic
	Db – Da – C → Db	SBC Da,Db	1	1	1	
SBR	Subtract and round					DALU Arithmetic
	Rnd(Dn – Da) → Dn	SBR Da,Dn	1	1	1	
SKIPLS	Skip loop if LC is less than or equal to zero (Branch not taken: 1 cycle. Taken: 4 cycles.)					Loop Instructions
	If LCn ≤ 0, then PC + displacement → PC 0 → LFn	SKIPLS label	2	1/4	1	
STOP	Stop processing (lowest power stand-by)					Program Control
	Enter the stop processing state	STOP	1	8	4	
SUB	Subtract					DALU Arithmetic
	Dn – #u5 → Dn Db – Da → Dn	SUB #u5,Dn {0 ≤ u5 < 32} SUB Da,Db,Dn	1	1	1	
SUB2	Subtract two words					DALU Arithmetic
	Dn.H – Da.H → Dn.H Dn.L – Da.L → Dn.L	SUB2 Da,Dn	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
SUBA	Subtract (affected by the modifier mode)					AGU Arithmetic
	Rx – #u5 → Rx Rx – rx → Rx	SUBA #u5,Rx {0 ≤ u5 < 64} SUBA rx,Rx	1 1	1 1	2 2	
SUBL	Shift left and subtract					DALU Arithmetic
	(2 * Dn) – Da → Dn	SUBL Da,Dn	1	1	1	
SUBNC.W	Subtract without changing the carry bit in the status register					DALU Arithmetic
	Dn – #s16 → Dn	SUBNC.W #s16,Dn {$-2^{15} \leq s16 < 2^{15}$}	2	1	4	
SXT.x	Sign extend					DALU Logical
	Da[7:0] → Dn[7:0]; Da[7] → Dn[39:8] Da[15:0] → Dn[15:0]; Da[15] → Dn[39:16] Dn[31] → Dn[39:32]	SXT.B Da,DN SXT.W Da,Dn SXT.L Dn	1 1 1	1 1 1	1 1 1	
SXTA.x	Sign extend					AGU Arithmetic
	rx[7:0] → Rx[7:0]; rx[7] → Rx[31:8] Rx[15] → Rx[31:16]	SXTA.B rx,Rx SXTA.W Rx	1 1	1 1	2 2	
TFR	Transfer data register to a data register					DALU Arithmetic
	Da → Dn	TFR Da,Dn	1	1	1	
TFRA	Register transfer					AGU Arithmetic
	rx → Rx	TFRA rx,Rx	1	1	2	
TFRA	Move the “other” stack pointer to/from a register, inversely defined by the exception mode					Stack Support
	If (SR[EXP] = 1), then NSP → Rn, else ESP → Rn If (SR[EXP] = 1), then Rn → NSP, else Rn → ESP	TFRA OSP,Rn TFRA Rn,OSP	1 1	1 1	4 4	
TFRc	Conditional data register transfer					DALU Arithmetic
	If T=1, then Da → Dn If T=0, then Da → Dn	TFRT Da,Dn TFRF Da,Dn	1 1	1 1	2 2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
TRAP	Execute a software exception (Cycle count depends upon machine state.)					Program Control
	PC → (ESP), SR → (ESP + 4), ESP + 8 → ESP VBR[31:12]:trap_vector → PC	TRAP {trap_vector = 000}	1	4/5	4	
TSTEQ	Test for equal to zero					DALU Arithmetic
	If Dn == 0, then 1 → T, else 0 → T	TSTEQ Dn	1	1	1	
TSTEQA.x	Test for equal to zero					AGU Arithmetic
	If Rx[15:0] == 0, then 1 → T, else 0 → T If Rx[31:0] == 0, then 1 → T, else 0 → T	TSTEQA.W Rx TSTEQA.L Rx	1 1	1 1	2 2	
TSTGE	Test for greater than or equal to zero					DALU Arithmetic
	If Dn > = 0, then 1 → T, else 0 → T	TSTGE Dn	1	1	1	
TSTGEA .L	Test for greater than or equal to zero					AGU Arithmetic
	If Rx ≥ 0, then 1 → T, else 0 → T	TSTGEA.L Rx	1	1	2	
TSTGT	Test for greater than zero					DALU Arithmetic
	If Dn > 0, then 1 → T, else 0 → T	TSTGT Dn	1	1	1	
TSTGTA	Test for greater than zero					AGU Arithmetic
	If Rx > 0, then 1 → T, else 0 → T	TSTGTA Rx	1	1	2	

Table 9-5. Instructions Grouped Alphabetically (Continued)

Mnemomic	Description		Words	Cycles	Type	Functional Group
	Operation	**Assembler Syntax**				
VSL	Viterbi shift left: specialized move to support Viterbi kernel					Move
	If VF2 == 1, then (D3.L << 1 + 1) → (word 3) else (D1.L << 1 + 1) → (word 3) If VF0 == 1, then (D3.L << 1) → (word 2) else (D1.L << 1) → (word 2) D2.L → (word 0) D6.L → (word 1)	VSL.4W D2:D6:D1:D3, (Rn)+N0	1	1	2	
	If VF3 == 1, then (D3.H << 1 + 1) → (word 3) else (D1.H << 1 + 1) → (word 3) If VF1 == 1, then (D3.H << 1) → (word 2) else (D1.H << 1) → (word 2) D2.H → (word 0) D6.H → (word 1)	VSL.4F D2:D6:D1:D3, (Rn)+N0	1	1	2	
	If VF2 == 1, then (D3.L << 1 + 1) → (word 1) else (D1.L << 1 + 1) → (word 1) If VF0 == 1, then (D3.L << 1) → (word 0) else (D1.L << 1) → (word 0)	VSL.2W D1:D3, (Rn)+N0	1	1	2	
	If VF3 == 1, then (D3.H << 1 + 1) → (word 1) else (D1.H << 1 + 1) → (word 1) If VF1 == 1, then (D3.H << 1) → (word 0) else (D1.H << 1) → (word 0)	VSL.2F D1:D3, (Rn)+N0	1	1	2	
WAIT	Wait for interrupt (low power stand-by)					Program Control
	Disable clocks to the processor core and enter the WAIT processing state	WAIT	1	8	4	
ZXT.x	Zero extend					DALU Logical
	Da[7:0] → Dn[7:0]; 0 → Dn[39:8] Da[15:0] → Dn[15:0]; 0 → Dn[39:16] 0 → Dn[39:32]	ZXT.B Da,Dn ZXT.W Da,Dn ZXT.L Dn	1 1 1	1 1 1	1 1 1	
ZXTA.x	Zero extend					AGU Arithmetic

Index

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

V

X