Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Freescale Semiconductor
Technical Data

MSC8101
Rev. 19, 5/2008

MSC8101

Network Digital Signal Processor

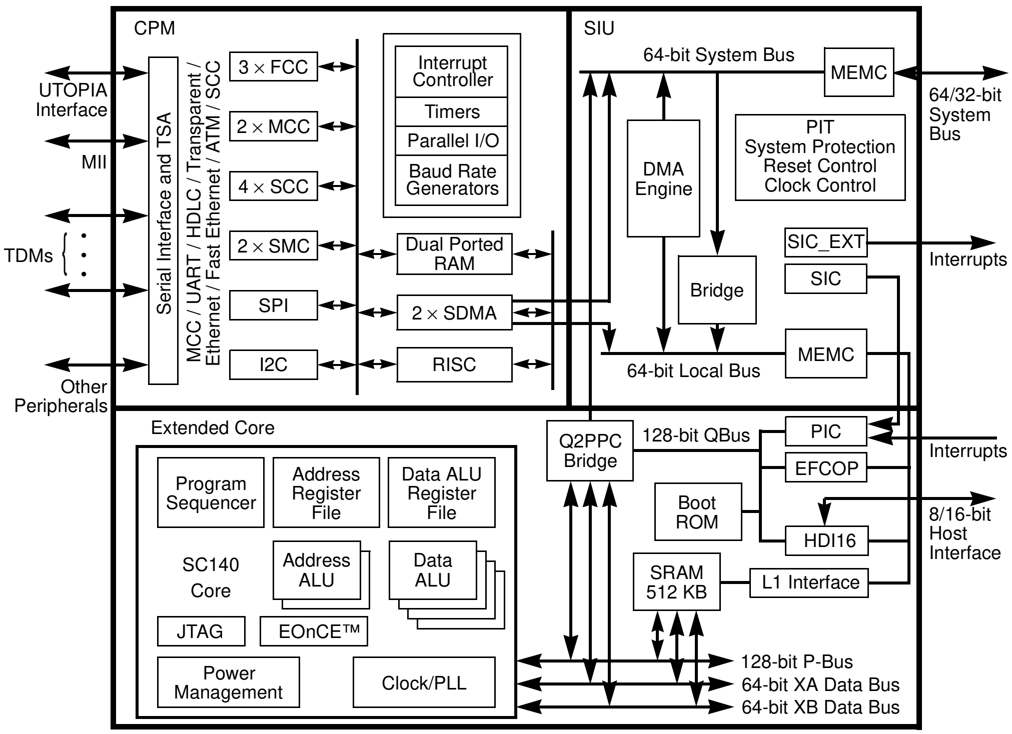


Figure 1. MSC8101 Block Diagram

The Freescale MSC8101 16-bit DSP is the first member of the family of DSPs based on the StarCore SC140 DSP core. The MSC8101 is available in three core speed levels: 250, 275, and 300 MHz.

What's New?

Rev. 19 includes the following changes:

- **Table 2-4** changes V_{IL} reference for signal low input current to 0.8 V..

The Freescale MSC8101 DSP is a very versatile device that integrates the high-performance SC140 four-ALU (arithmetic logic unit) DSP core along with 512 KB of internal memory, a communications processor module (CPM), a 64-bit bus, a very flexible System Integration Unit (SIU), and a 16-channel DMA engine on a single device. With its four-ALU core, the MSC8101 can execute up to four multiply-accumulate (MAC) operations in a single clock cycle. The MSC8101 CPM is a 32-bit RISC-based communications protocol engine that can network to time-division multiplexed (TDM) highways, Ethernet, and asynchronous transfer mode (ATM) backbones. The MSC8101 60x-compatible bus interface facilitates its connection to multi-master system architectures. The very large internal memory, 512 KB, reduces the need for external program and data memories. The MSC8101 offers 1500 DSP MMACS (1200 core and 300 EFCOP) performance using an internal 300 MHz clock with a 1.6 V core and independent 3.3 V input/output (I/O).

Table of Contents

Data Sheet Conventions

pin and pin-out	Although the device package does not have pins, the term pins and pin-out are used for convenience and indicate specific signal locations within the ball-grid array.
$\overline{\text{OVERBAR}}$	Used to indicate a signal that is active when pulled low (For example, the $\overline{\text{RESET}}$ pin is active when low.)
"asserted"	Means that a high true (active high) signal is high or that a low true (active low) signal is low
"deasserted"	Means that a high true (active high) signal is low or that a low true (active low) signal is high

Examples:

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{\text{PIN}}$	True	Asserted	V_{IL}/V_{OL}
$\overline{\text{PIN}}$	False	Deasserted	V_{IH}/V_{OH}
PIN	True	Asserted	V_{IH}/V_{OH}
PIN	False	Deasserted	V_{IL}/V_{OL}

Note: Values for V_{IL}, V_{OL}, V_{IH}, and V_{OH} are defined by individual product specifications.

MSC8101 Features

- SC140 core
  - Architecture optimized for efficient C/C++ code compilation
  - Four 16-bit ALUs and two 32-bit AGUs
  - 1200 DSP MMACS running at 300 MHz
  - Very low power dissipation
  - Variable-length execution set (VLES) execution model
  - JTAG/Enhanced OnCE debug port
- Communications processor module (CPM)
  - Programmable protocol machine using a 32-bit RISC engine
  - 155 Mbps ATM interface (including AAL 0/1/2/5)
  - 10/100 Mbit Ethernet interface
  - Up to four E1/T1 interfaces or one E3/T3 interface and one E1/T1 interface
  - HDLC support up to T3 rates, or 256 channels
- 64- or 32-bit wide bus interface
  - Support for bursts for high efficiency
  - Glueless interface to 60x-compatible bus systems
  - Multi-master support
- Enhanced filter coprocessor (EFCOP)
  - Independently and concurrently executes long filters (such as echo cancellation)
  - Runs at 250/275/300 MHz and provides 250/275/300 MMACS performance
- Programmable memory controller
  - Control for up to eight banks of external memory
  - User-programmable machines (UPM) allowing glueless interface to various memory types (SRAM, DRAM, EPROM, and Flash memory) and other user-definable peripherals
  - Dedicated pipelined SDRAM memory interface
- Large internal SRAM
  - 256K 16-bit words (512 KB)
  - Unified program and data space configurable by the application
  - Word and byte addressable
- DMA controller
  - 16 DMA channels, FIFO based, with burst capabilities
  - Sophisticated addressing capabilities
- Small foot print package
  - 17 mm × 17 mm lidded FC-PBGA lead-bearing or lead-free package
- Very low power consumption
  - Separate power supply for internal logic (1.6 V) and for I/O (3.3 V)
- Enhanced 16-bit parallel host interface (HDI16)
  - Supports a variety of microcontroller, microprocessor, and DSP bus interfaces
- Phase-lock loops (PLLs)
  - System PLL
  - CPM DPLLs (SCC and SCM)
- Process technology
  - 0.13 micron copper interconnect process technology

Target Applications

The MSC8101 targets applications requiring very high performance, very large amounts of internal memory, and such networking capabilities as:

- Third-generation wideband wireless infrastructure systems
- Packet Telephony systems
- Multi-channel modem banks
- Multi-channel xDSL

Product Documentation

The documents listed in **Table 1** are required for a complete description of the MSC8101 and are necessary to design properly with the part. Documentation is available from the following sources (see back cover for details):

- A local Freescale distributor
- A Freescale Semiconductor sales office
- A Freescale Semiconductor Literature Distribution Center
- The world wide web (WWW)

Table 1. MSC8101 Documentation

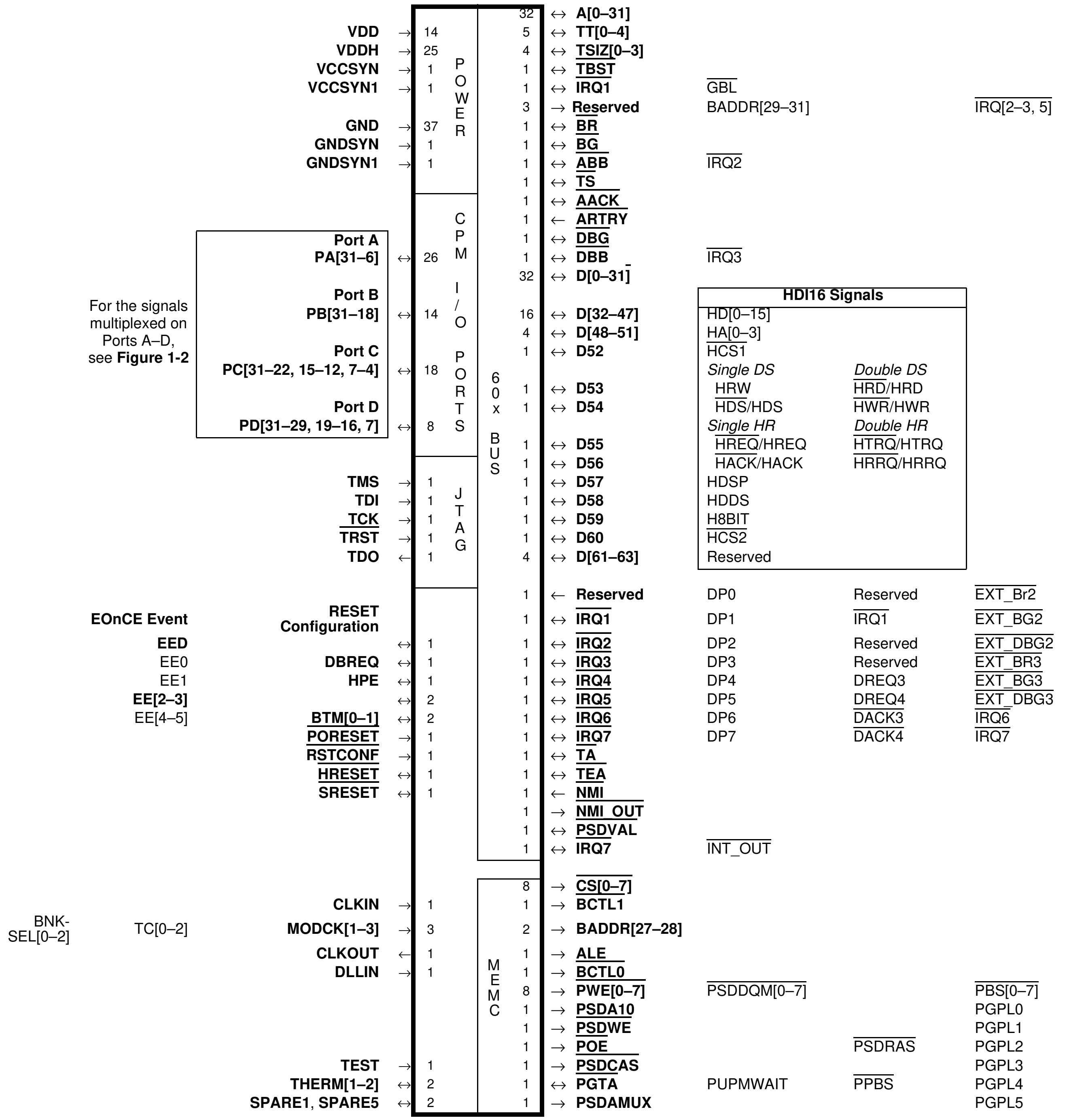
Name	Description	Order Number
MSC8101 Technical Data	MSC8101 features list and physical, electrical, timing, and package specifications	MSC8101/D
MSC8101 User's Guide	Detailed functional description of the MSC8101 memory configuration, operation, and register programming	MSC8101UG/D
MSC8101 Pocket Guide	Quick reference information for application development.	MSC8101PG/D
MSC8101 Reference Manual	Detailed description of the MSC8101 processor core and instruction set	MSC8101RM/D
SC140 DSP Core Reference Manual	Detailed description of the SC140 family processor core and instruction set	MNSC140CORE/D
Application Notes	Documents describing specific applications or optimized device operation including code examples	See the MSC8101 product website

Signals/Connections 1

The MSC8101 external signals are organized into functional groups, as shown in **Table 1-1**, **Figure 1-1**, and **Figure 1-2**. **Table 1-1** lists the functional groups, states the number of signal connections in each group, and references the table that gives details on multiplexed signals within each group. **Figure 1-1** shows MSC8101 external signals organized by function. **Figure 1-2** indicates how the parallel input/output (I/O) ports signals are multiplexed. Because the parallel I/O design supported by the MSC8101 communications processor module (CPM) is a subset of the parallel I/O signals supported by the MPC8260 device, port pins are not numbered sequentially.

Table 1-1. MSC8101 Functional Signal Groupings

Functional Group		Number of Signal Connections	Detailed Description
Power (V_{CC}, V_{DD}, and GND)		80	**Table 1-2** on page 1-4
Clock		6	**Table 1-3** on page 1-4
Reset, configuration, and EOnCE		11	**Table 1-4** on page 1-5
System bus, HDI16, and interrupts		133	**Table 1-5** on page 1-7
Memory Controller		27	**Table 1-6** on page 1-13
CPM Input/Output Parallel Ports	Port A	26	**Table 1-7** on page 1-16
	Port B	14	**Table 1-8** on page 1-21
	Port C	18	**Table 1-9** on page 1-24
	Port D	8	**Table 1-10** on page 1-33
JTAG Test Access Port		5	**Table 1-11** on page 1-36
Reserved (denotes connections that are always reserved)		5	**Table 1-12** on page 1-36



Note: Refer to the *System Interface Unit (SIU)* chapter in the *MSC8101 Reference Manual* for details on how to configure these pins.

Figure 1-1. MSC8101 External Signals

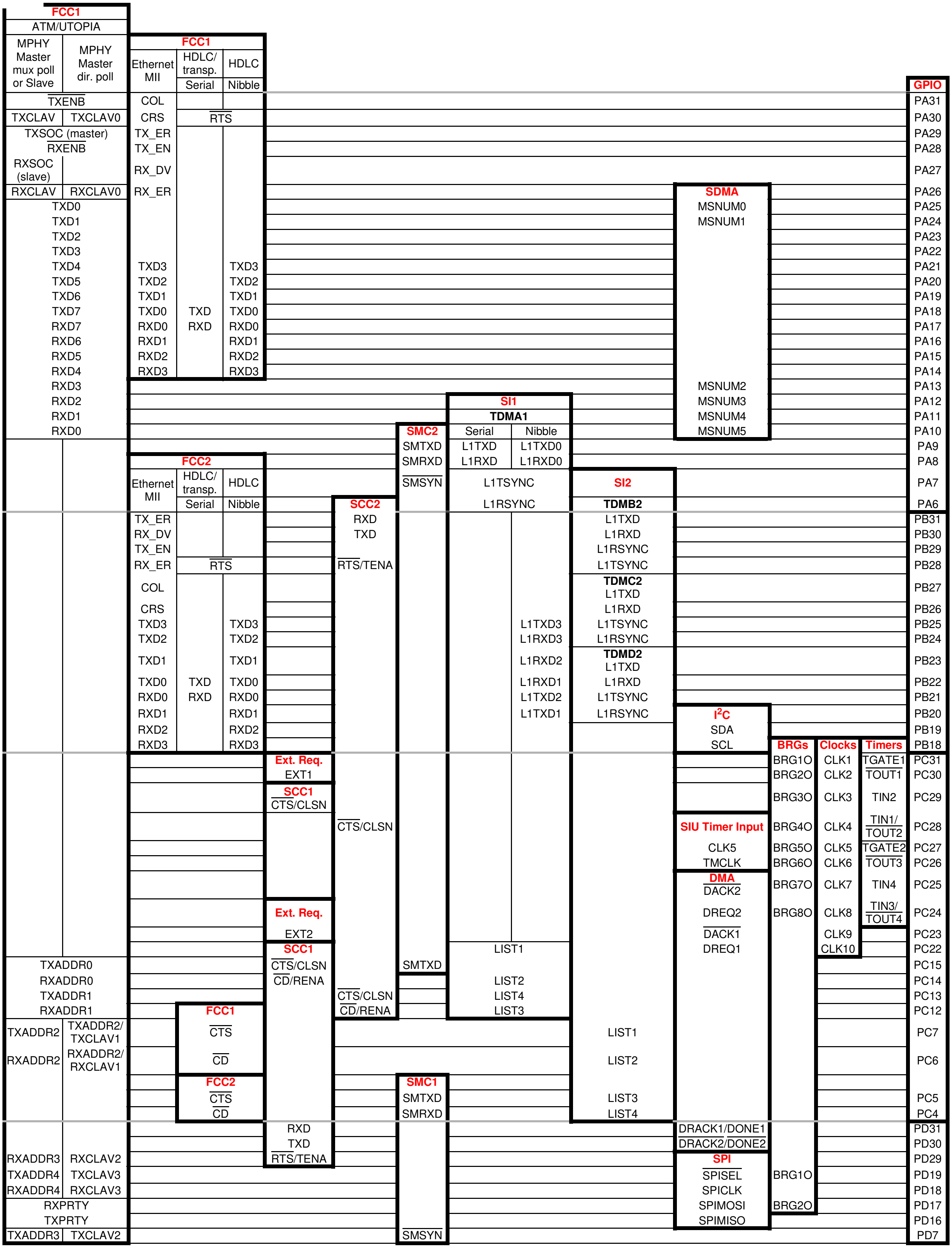


Figure 1-2. CPM Port A–D Pin Multiplexed Functionality

1.1 Power Signals

Table 1-2. Power and Ground Signal Inputs

Power Name	Description
V_{DD}	**Internal Logic Power** V_{DD} dedicated for use with the device core. The voltage should be well-regulated and the input should be provided with an extremely low impedance path to the V_{DD} power rail.
V_{DDH}	**Input/Output Power** This source supplies power for the I/O buffers. The user must provide adequate external decoupling capacitors.
V_{CCSYN}	**System PLL Power** V_{CC} dedicated for use with the system Phase Lock Loop (PLL). The voltage should be well-regulated and the input should be provided with an extremely low impedance path to the V_{CC} power rail.
V_{CCSYN1}	**SC140 PLL Power** V_{CC} dedicated for use with the SC140 core PLL. The voltage should be well-regulated and the input should be provided with an extremely low impedance path to the V_{CC} power rail.
GND	**System Ground** An isolated ground for the internal processing logic. This connection must be tied externally to all chip ground connections, except GND_{SYN} and GND_{SYN1}. The user must provide adequate external decoupling capacitors.
GND_{SYN}	**System PLL Ground** Ground dedicated for system PLL use. The connection should be provided with an extremely low-impedance path to ground.
GND_{SYN1}	**SC140 PLL Ground 1** Ground dedicated for SC140 core PLL use. The connection should be provided with an extremely low-impedance path to ground.

1.2 Clock Signals

Table 1-3. Clock Signals

Signal Name	Type	Signal Description
CLKIN	Input	**Clock In** Primary clock input to the MSC8101 PLL.
$\overline{\text{MODCK1}}$ TC0 BNKSEL0	Input Output Output	**Clock Mode Input 1** Defines the operating mode of internal clock circuits. **Transfer Code 0** Supplies information that can be useful for debugging bus transactions initiated by the MSC8101. **Bank Select 0** Selects the SDRAM bank when the MSC8101 is in 60x-compatible bus mode.
$\overline{\text{MODCK2}}$ TC1 BNKSEL1	Input Output Output	**Clock Mode Input 2** Defines the operating mode of internal clock circuits. **Transfer Code 1** Supplies information that can be useful for debugging bus transactions initiated by the MSC8101. **Bank Select 1** Selects the SDRAM bank when the MSC8101 is in 60x-compatible bus mode.
$\overline{\text{MODCK3}}$ TC2 BNKSEL2	Input Output Output	**Clock Mode Input 3** Defines the operating mode of internal clock circuits. **Transfer Code 2** Supplies information that can be useful for debugging bus transactions initiated by the MSC8101. **Bank Select 2** Selects the SDRAM bank when the MSC8101 is in 60x-compatible bus mode.

Table 1-3. Clock Signals (Continued)

Signal Name	Type	Signal Description
CLKOUT	Output	**Clock Out** The system bus clock.
DLLIN	Input	**DLLIN** Synchronizes with an external device. **Note:** When the DLL is disabled, connect this signal to GND.

1.3 Reset, Configuration, and EOnCE Event Signals

Table 1-4. Reset, Configuration, and EOnCE Event Signals

Signal Name	Type	Signal Description
DBREQ	Input	**Debug Request** Determines whether to go into SC140 Debug mode when $\overline{\text{PORESET}}$ is deasserted.
EE0[1]		**Enhanced OnCE (EOnCE) Event 0** After $\overline{\text{PORESET}}$ is deasserted, you can configure EE0 as an input (default) or an output.
	Input	Debug request, enable Address Event Detection Channel 0, or generate an EOnCE event.
	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 system data bus is 32 bits wide, and the Host *must* program the reset configuration word.
EE1[1]		**EOnCE Event 1** After $\overline{\text{PORESET}}$ is deasserted, you can configure EE1 as an input (default) or an output.
	Input	Enable Address Event Detection Channel 1 or generate an EOnCE event.
	Output	Debug Acknowledge or detection by Address Event Detection Channel 1. Used to trigger external debugging equipment.
EE2[1]		**EOnCE Event 2** After $\overline{\text{PORESET}}$ is deasserted, you can configure EE2 as an input (default) or an output.
	Input	Enable Address Event Detection Channel 2 or generate an EOnCE event or enable the Event Counter.
	Output	Detection by Address Event Detection Channel 2. Used to trigger external debugging equipment.
EE3[1]		**EOnCE Event 3** After $\overline{\text{PORESET}}$ 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 the ERCV Register.
	Input	Enable Address Event Detection Channel 3 or generate one of the EOnCE events.
	Output	The DSP has read the EOnCE Receive Register (ERCV). Triggers external debugging equipment.

Table 1-4. Reset, Configuration, and EOnCE Event Signals (Continued)

Signal Name	Type	Signal Description
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[1]		**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 the ETRSMT Register.
	Input	Enable Address Event Detection Channel 4 or generate an EOnCE event.
	Output	The DSP wrote the EOnCE Transmit Register (ETRSMT). Triggers external debugging equipment.
EE5[1]		**EOnCE Event 5** After PORESET is deasserted, you can configure EE5 as an input (default) or an output.
	Input	Enable Address Event Detection Channel 5.
	Output	Detection by Address Event Detection Channel 5. Triggers external debugging equipment.
EED[1]		**Enhanced OnCE (EOnCE) Event Detection** After PORESET is deasserted, you can configure EED as an input (default) or output:
	Input	Enable the Data Event Detection Channel.
	Output	Detection by the Data Event Detection Channel. Triggers external debugging equipment.
PORESET	Input	**Power-On Reset** When asserted, this line causes the MSC8101 to enter power-on reset state.
RSTCONF	Input	**Reset Configuration** Used during reset configuration sequence of the chip. A detailed explanation of its function is provided in the “Power-On Reset Flow” and “Hardware Reset Configuration” sections of the *MSC8101 Reference Manual*.
HRESET	Input	**Hard Reset** When asserted, this open-drain line causes the MSC8101 to enter the hard reset state.
SRESET	Input	**Soft Reset** When asserted, this open-drain line causes the MSC8101 to enter the soft reset state.
Note: See the emulation and debug chapter in the *SC140 DSP Core Reference Manual* for details on how to configure these pins.		

1.4 System Bus, HDI16, and Interrupt Signals

The system bus, HDI16, and interrupt signals are grouped together because they use a common set of signal lines. Individual assignment of a signal to a specific signal line is configured through registers in the System Interface Unit (SIU) and the Host Interface (HDI16). 1-5 describes the signals in this group.

Note: To boot from the host interface, the HDI16 must be enabled by pulling up the HPE signal line during PORESET. The configuration word must then be loaded from the host. The configuration word must set the Internal Space Port Size bit in the Bus Control Register (BCR[ISPS]) to change the system data bus width from 64 bits to 32 bits and reassign the upper 32 bits to their HDI16 functions. Never set the Host Port Enable (HEN) bit in the Host Port Control Register (HPCR) to enable the HDI16, unless the bus size is first changed from 64 bits to 32 bits. Otherwise, unpredictable operation may occur.

Although there are eight interrupt request ($\overline{\text{IRQ}}$) connections to the core processor, there are multiple external lines that can connect to these internal signal lines. After reset, the default configuration includes two $\overline{\text{IRQ1}}$ and two $\overline{\text{IRQ7}}$ input lines. The designer must select one line for each required interrupt and reconfigure the other external signal line or lines for alternate functions.

Table 1-5. System Bus, HDI16, and Interrupt Signals

Signal	Data Flow	Description
A[0–31]	Input/Output	**Address Bus** When the MSC8101 is in external master bus mode, these pins function as the address bus. The MSC8101 drives the address of its internal bus masters and responds to addresses generated by external 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	**Bus Transfer Type** The bus master drives these pins during the address tenure to specify the type of transaction.
TSIZ[0–3]	Input/Output	**Transfer Size** The bus master drives these pins with a value indicating the number of bytes transferred in the current transaction.
$\overline{\text{TBST}}$	Input/Output	**Bus Transfer Burst** The bus master asserts this pin to indicate that the current transaction is a burst transaction (transfers four quad words).
$\overline{\text{IRQ1}}$ $\overline{\text{GBL}}$	Input Input/Output	**Interrupt Request 1**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core. **Global**[1] When a master within the chip initiates a bus transaction, it drives this pin. When an external 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.
Reserved BADDR29 $\overline{\text{IRQ2}}$	Output Output Input	The primary configuration is reserved. **Burst Address 29**[1] One of five outputs of the memory controller. These pins connect directly to memory devices controlled by the MSC8101 memory controller. **Interrupt Request 2**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
Reserved BADDR30 $\overline{\text{IRQ3}}$	Output Output Input	The primary configuration is reserved. **Burst Address 30**[1] One of five outputs of the memory controller. These pins connect directly to memory devices controlled by the MSC8101 memory controller. **Interrupt Request 3**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
Reserved BADDR31 $\overline{\text{IRQ5}}$	Output Output Input	The primary configuration is reserved. **Burst Address 31**[1] One of five outputs of the memory controller. These pins connect directly to memory devices controlled by the MSC8101 memory controller. **Interrupt Request 5**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
$\overline{BR}$	Input/Output Output Input	**Bus Request**[2] An output when an external arbiter is used. The MSC8101 asserts this pin to request ownership of the bus. An input when an internal arbiter is used. An external master should assert this pin to request bus ownership from the internal arbiter.
$\overline{BG}$	Input/Output Output Input	**Bus Grant**[2] An output when an internal arbiter is used. The MSC8101 asserts this pin to grant bus ownership to an external bus master. An input when an external arbiter is used. The external arbiter should assert this pin to grant bus ownership to the MSC8101.
$\overline{ABB}$ $\overline{IRQ2}$	Input/Output Output Input Input	**Address Bus Busy**[1] The MSC8101 asserts this pin for the duration of the address bus tenure. Following an address acknowledge ($\overline{AACK}$) signal, which terminates the address bus tenure, the MSC8101 deasserts $\overline{ABB}$ for a fraction of a bus cycle and then stops driving this pin. The MSC8101 does not assume bus ownership while it this pin is asserted by an external bus master. **Interrupt Request 2**[1] One of the eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{TS}$	Input/Output	**Bus Transfer Start** Signals the beginning of a new address bus tenure. The MSC8101 asserts this signal when one of its internal bus masters (SC140 core or DMA controller) begins an address tenure. When the MSC8101 senses this pin being asserted by an external bus master, it responds to the address bus tenure as required (snoop if enabled, access internal MSC8101 resources, memory controller support).
$\overline{AACK}$	Input/Output	**Address Acknowledge** A bus slave asserts this signal to indicate that it identified the address tenure. Assertion of this signal terminates the address tenure.
$\overline{ARTRY}$	Input	**Address Retry** Assertion of this signal indicates that the bus transaction should be retried by the bus master. The MSC8101 asserts this signal to enforce data coherency with its internal cache and to prevent deadlock situations.
$\overline{DBG}$	Input/Output Output Input	**Data Bus Grant**[2] An output when an internal arbiter is used. The MSC8101 asserts this pin as an output to grant data bus ownership to an external bus master. An input when an external arbiter is used. The external arbiter should assert this pin as an input to grant data bus ownership to the MSC8101.
$\overline{DBB}$ $\overline{IRQ3}$	Input/Output Output Input Input	**Data Bus Busy**[1] The MSC8101 asserts this pin as an output for the duration of the data bus tenure. Following a $\overline{TA}$, which terminates the data bus tenure, the MSC8101 deasserts $\overline{DBB}$ for a fraction of a bus cycle and then stops driving this pin. The MSC8101 does not assume data bus ownership while $\overline{DBB}$ is asserted by an external bus master. **Interrupt Request 3**[1] 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	**Data Bus Most Significant Word** In write transactions the bus master drives the valid data on this bus. In read transactions the slave drives the valid data on this bus. In Host Port Disabled mode, these 32 bits are part of the 64-bit data bus. In Host Port Enabled mode, these bits are used as the bus in 32-bit mode.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
D[32–47]	Input/Output	**Data Bus Bits 32–47** In write transactions the bus master drives the valid data on this bus. In read transactions the slave drives the valid data on this bus.
HD[0–15]	Input/Output	**Host Data**[2] When the HDI16 interface is enabled, these signals are lines 0-15 of the bidirectional tri-state data bus.
D[48–51]	Input/Output	**Data Bus Bits 48–51** In write transactions the bus master drives the valid data on these pins. In read transactions the slave drives the valid data on these pins.
HA[0–3]	Input	**Host Address Line 0–3**[3] When the HDI16 interface bus is enabled, these lines address internal host registers.
D52	Input/Output	**Data Bus Bit 52** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin.
$\overline{HCS1}$	Input	**Host Chip Select**[3] When the HDI16 interface is enabled, this is one of the two chip-select pins. The HDI16 chip select is a logical OR of $\overline{HCS1}$ and $\overline{HCS2}$.
D53	Input/Output	**Data Bus Bit 53** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin.
HRW	Input	**Host Read Write Select**[3] When the HDI16 interface is enabled in Single Strobe mode, this is the read/write input (HRW).
$\overline{HRD}$/HRD	Input	**Host Read Strobe**[3] When the HDI16 is programmed to interface with a double data strobe host bus, this pin is the read data strobe Schmitt trigger input ($\overline{HRD}$/HRD). The polarity of the data strobe is programmable.
D54	Input/Output	**Data Bus Bit 54** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin.
$\overline{HDS}$/HDS	Input	**Host Data Strobe**[3] When the HDI16 is programmed to interface with a single data strobe host bus, this pin is the data strobe Schmitt trigger input ($\overline{HDS}$/HDS). The polarity of the data strobe is programmable.
$\overline{HWR}$/HWR	Input	**Host Write Data Strobe**[3] When the HDI16 is programmed to interface with a double data strobe host bus, this pin is the write data strobe Schmitt trigger input ($\overline{HWR}$/HWR). The polarity of the data strobe is programmable.
D55	Input/Output	**Data Bus Bit 55** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin.
$\overline{HREQ}$/HREQ	Output	**Host Request**[3] When the HDI16 is programmed to interface with a single host request host bus, this pin is the host request output ($\overline{HREQ}$/HREQ). The polarity of the host request is programmable. The host request may be programmed as a driven or open-drain output.
$\overline{HTRQ}$/HTRQ	Output	**Transmit Host Request**[3] When the HDI16 is programmed to interface with a double host request host bus, this pin is the transmit host request output ($\overline{HTRQ}$/HTRQ). The signal can be programmed as driven or open drain. The polarity of the host request is programmable.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
D56 $\overline{\text{HACK}}$/HACK $\overline{\text{HRRQ}}$/HRRQ	Input/Output Output Output	**Data Bus Bit 56** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin. **Host Acknowledge**[3] When the HDI16 is programmed to interface with a single host request host bus, this pin is the host acknowledge Schmitt trigger input (HACK). The polarity of the host acknowledge is programmable. **Receive Host Request**[3] When the HDI16 is programmed to interface with a double host request host bus, this pin is the receive host request output ($\overline{\text{HRRQ}}$/HRRQ). The signal can be programmed as driven or open drain. The polarity of the host request is programmable.
D57 HDSP	Input/Output Input	**Data Bus Bit 57** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin. **Host Data Strobe Polarity**[3] When the HDI16 interface is enabled, this pin is the host data strobe polarity (HDSP).
D58 HDDS	Input/Output Input	**Data Bus Bit 58** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin. **Host Dual Data Strobe**[3] When the HDI16 interface is enabled, this pin is the host dual data strobe (HDDS).
D59 H8BIT	Input/Output Input	**Data Bus Bit 59** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin. **H8BIT**[3] When the HDI16 interface is enabled, this bit determines if the interface is in 8-bit or 16-bit mode.
D60 $\overline{\text{HCS2}}$	Input/Output Input	**Data Bus Bit 60** In write transactions the bus master drives the valid data on this pin. In read transactions the slave drives the valid data on this pin. **Host Chip Select** [3] When the HDI16 interface is enabled, this is one of the two chip-select pins. The HDI16 chip select is a logical OR of $\overline{\text{HCS1}}$ and $\overline{\text{HCS2}}$.
D[61–63] Reserved	Input/Output	**Data Bus Bits 61–63** Used only in 60x-mode-only mode. In write transactions the bus master drives the valid data on this bus. In read transactions the slave drives the valid data on this bus. These dedicated signals are reserved when the HDI16 is enabled.[3]
Reserved DP0 $\overline{\text{EXT\_BR2}}$	Input Input/Output Input	The primary configuration is reserved. **Data Parity 0**[1] The 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]. **External Bus Request 2**[1,2] An external master asserts this pin to request bus ownership from the internal arbiter.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
$\overline{\text{IRQ1}}$	Input	**Interrupt Request 1**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP1	Input/Output	**Data Parity 1**[1] The 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**[1,2] The MSC8101 asserts this pin to grant bus ownership to an external bus master.
$\overline{\text{IRQ2}}$	Input	**Interrupt Request 2**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP2	Input/Output	**Data Parity 2**[1] The 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**[1,2] The MSC8101 asserts this pin to grant data bus ownership to an external bus master.
$\overline{\text{IRQ3}}$	Input	**Interrupt Request 3**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP3	Input/Output	**Data Parity 3**[1] The 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 Bus Request 3**[1,2] An external master asserts this pin to request bus ownership from the internal arbiter.
$\overline{\text{IRQ4}}$	Input	**Interrupt Request 4**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP4	Input/Output	**Data Parity 4**[1] The 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**[1] An external peripheral uses this pin to request DMA service.
$\overline{\text{EXT\_BG3}}$	Output	**External Bus Grant 3**[1,2] The MSC8101 asserts this pin to grant bus ownership to an external bus master.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
IRQ5	Input	**Interrupt Request 5**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP5	Input/Output	**Data Parity 5**[1] The 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**[1] An external peripheral uses this pin to request DMA service.
EXT_DBG3	Output	**External Data Bus Grant 3**[1,2] The MSC8101 asserts this pin to grant data bus ownership to an external bus master.
IRQ6	Input	**Interrupt Request 6**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP6	Input/Output	**Data Parity 6**[1] The 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].
DACK3	Output	**DMA Acknowledge 3**[1] The DMA controller drives this output to acknowledge the DMA transaction on the bus.
IRQ7	Input	**Interrupt Request 7**[1] One of eight external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP7	Input/Output	**Data Parity 7**[1] The 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].
DACK4	Output	**DMA Acknowledge**[1] The DMA controller drives this output to acknowledge the DMA transaction on the bus.
TA	Input/Output	**Transfer Acknowledge** Indicates that a data beat is valid on the data bus. For single beat transfers, assertion of TA indicates the termination of the transfer. For burst transfers, TA is asserted four times to indicate the transfer of four data beats with the last assertion indicating the termination of the burst transfer.
TEA	Input/Output	**Transfer Error Acknowledge** Indicates a bus error. masters within the MSC8101 monitor the state of this pin. The MSC8101 internal bus monitor can assert this pin if it identifies a bus transfer that is hung.
NMI	Input	**Non-Maskable Interrupt** When an external device asserts this line, the MSC8101 NMI input is asserted.
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.
PSDVAL	Input/Output	**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 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.

Table 1-5. System Bus, HDI16, and Interrupt Signals (Continued)

Signal	Data Flow	Description
IRQ7	Input	**Interrupt Request 7**[1] 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**[1] Driven from the MSC8101 internal interrupt controller. Assertion of this output indicates that an unmasked interrupt is pending in the MSC8101 internal interrupt controller.

Notes:
1. See the SIU chapter in the *MSC8101 Reference Manual* for details on how to configure these pins.
2. When used as the bus control arbiter for the system bus, the MSC8101 can support up to three external bus masters. Each master uses its own set of Bus Request, Bus Grant, and Data Bus Grant signals (BR/BG/DBG, EXT_BR2/EXT_BG2/EXT_DBG2, and EXT_BR3/EXT_BG3/EXT_DBG3). Each of these signal sets must be configured to indicate whether the external master is or is not a MSC8101 master device. See the Bus Configuration Register (BCR) description in the SIU chapter in the *MSC8101 Reference Manual* for details on how to configure these pins. The second and third set of pins is defined by EXT_xxx to indicate that they can only be used with external master devices. The first set of pins (BR/BG/DBG) have a dual function. When the MSC8101 is not the bus arbiter, these signals (BR/BG/DBG) are used by the MSC8101 to obtain master control of the bus.
3. See the host interface (HDI16) chapter in the *MSC8101 Reference Manual* for details on how to configure these pins.

1.5 Memory Controller Signals

Refer to the memory controller chapter in the *MSC8101 Reference Manual (MSC8101RM/D)* for detailed information about configuring these signals.

Table 1-6. Memory Controller Signals

Signal	Data Flow	Description
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 data bus. Usually used with BCTL0. The exact function of this pin is defined by the value of SIUMCR[BCTLC]. See the *System Interface Unit (SIU)* chapter in the *MSC8101 Reference Manual* for details.
BADDR[27–28]	Output	**Burst Address 27–28** Two of five outputs of the memory controller. These pins 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 bus configuration.
BCTL0	Output	**Buffer Control 0** Controls buffers on the data bus. The exact function of this pin is defined by the value of SIUMCR[BCTLC]. See the *System Interface Unit (SIU)* chapter in the *MSC8101 Reference Manual* for details.
PWE[0–7]	Output	**Bus Write Enable** Outputs of the bus General-Purpose Chip-select Machine (GPCM). These pins select byte lanes for write operations.
PSDDQM[0–7]	Output	**Bus SDRAM DQM** Outputs of the SDRAM control machine. These pins select specific byte lanes of SDRAM devices.
PBS[0–7]	Output	**Bus UPM Byte Select** Outputs of the User-Programmable Machine (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.

Table 1-6. Memory Controller Signals (Continued)

Signal	Data Flow	Description
PSDA10	Output	**Bus SDRAM A10** Output from the 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	**Bus UPM General-Purpose Line 0** One of six general-purpose output lines of the UPM. The values and timing of this pin are programmed in the UPM.
$\overline{\text{PSDWE}}$	Output	**Bus SDRAM Write Enable** Output from the bus SDRAM controller. This pin should connect to the SDRAM WE input signal.
PGPL1	Output	**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	**Bus Output Enable** Output of the bus GPCM. Controls the output buffer of memory devices during read operations.
$\overline{\text{PSDRAS}}$	Output	**Bus SDRAM RAS** Output from the bus SDRAM controller. This pin should connect to the SDRAM Row Address Strobe (RAS) input signal.
PGPL2	Output	**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	**Bus SDRAM CAS** Output from the bus SDRAM controller. This pin should connect to the SDRAM Column Address Strobe (CAS) input signal.
PGPL3	Output	**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	**GPCM TA** Terminates transactions during GPCM operation. Requires an external pull up resistor for proper operation.
PUPMWAIT	Input	**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.
$\overline{\text{PPBS}}$	Output	**Bus Parity Byte Select** In systems that store data parity in a separate chip, this output is the byte-select for that chip.
PGPL4	Output	**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.
PSDAMUX	Output	**Bus SDRAM Address Multiplexer** Controls the SDRAM address multiplexer when the MSC8101 is in External Master mode.
PGPL5	Output	**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.

1.6 CPM Ports

The MSC8101 CPM supports the subset of MPC8260 signals as described below.

- The MSC8101 CPM includes the following set of communication controllers:
- Two full-duplex fast serial communications controllers (FCCs) that support:
  - Asynchronous transfer mode (ATM) through a UTOPIA 8 interface (FCC1 only)—The MSC8101 can operate as one of the following:
    - UTOPIA slave device
    - UTOPIA multi-PHY master device using direct polling for up to 4 PHY devices
    - UTOPIA multi-PHY master device using multiplex polling that can address up to 31 PHY devices at addresses 0–30 (address 31 is reserved as a null port).
  - IEEE 802.3/Fast Ethernet through a Media-Independent Interface (MII)
  - High-level data link control (HDLC) Protocol:
    - Serial mode—Transfers data one bit at a time
    - Nibble mode—Transfers data four bits at a time
  - Transparent mode serial operation
- One FCC that operates with the TSA only
- Two multi-channel controllers (MCCs) that together can handle up to 256 HDLC/transparent channels at 64 Kbps each, multiplexed on up to four TDM interfaces
- Two full-duplex serial communications controllers (SCCs) that support the following protocols:
  - IEEE 802.3/fast Ethernet through a media-independent interface (MII)
  - HDLC Protocol:
    - Serial mode—Transfers data one bit at a time
    - Nibble mode—Transfers data four bits at a time
  - Synchronous data link control (SDLC)
  - LocalTalk (HDLC-based local area network protocol)
  - Universal asynchronous receiver/transmitter (UART)
  - Synchronous UART (1x clock mode)
  - Binary synchronous (BISYNC) communication
  - Transparent mode serial operation
- Two additional SCCs that operate with the TSA only
- Two full-duplex serial management controllers (SMCs) that support the following protocols:
  - General circuit interface (GCI)/integrated services digital network (ISDN) monitor and C/I channels (TSA only)
  - UART
  - Transparent mode serial operation
- Serial peripheral interface (SPI) support for master or slave operation
- Inter-integrated circuit (I^2C) bus controller
- Time-slot assigner (TSA) that supports multiplexing from any of the SCCs, FCCs, SMCs, and two MCCs onto four time-division multiplexed (TDM) interfaces. The TSA uses two serial interfaces (SI1 and SI2). SI1 uses TDMA1 which supports both serial and nibble mode. SI2 does not support nibble mode and includes TDMB2, TDMC2, and TDMD2, which operate only in serial mode.

The individual sets of externals signals associated with a specific protocol and data transfer mode are multiplexed across any or all of the ports, as shown in **Figure 1-2**. The following sections describe the signals supported by Ports A–D.

1.6.1 Port A Signals

Table 1-7. Port A Signals

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated Signal *Protocol*		
PA31	FCC1: $\overline{\text{TXENB}}$ *UTOPIA master*	Output	**FCC1: UTOPIA Master Transmit Enable** Asserted by the MSC8101 (UTOPIA master PHY) when there is valid transmit cell data (TXD[0–7]).
	FCC1: $\overline{\text{TXENB}}$ *UTOPIA slave*	Input	**FCC1: UTOPIA Slave Transmit Enable** Asserted by an external UTOPIA master PHY when there is valid transmit cell data (TXD[0–7]).
	FCC1: COL *MII*	Input	**FCC1: Media Independent Interface Collision Detect** Asserted by an external fast Ethernet PHY when collision is detected.
PA30	FCC1: TXCLAV *UTOPIA slave*	Output	**FCC1: UTOPIA Slave Transmit Cell Available** Asserted by the MSC8101 (UTOPIA slave PHY) when the MSC8101 can accept one complete ATM cell.
	FCC1: TXCLAV *UTOPIA master*, or	Input	**FCC1: UTOPIA Master Transmit Cell Available** Asserted by an external UTOPIA slave PHY to indicate that it can accept one complete ATM cell.
	FCC1: TXCLAV0 *UTOPIA master, Multi-PHY, direct polling*	Input	**FCC1: UTOPIA Master Transmit Cell Available Multi-PHY Direct Polling** Asserted by an external UTOPIA slave PHY using direct polling to indicate that it can accept one complete ATM cell.
	FCC1: $\overline{\text{RTS}}$ *HDLC, Serial and Nibble*	Output	**FCC1: Request To Send** In the standard modem interface signals supported by FCC1 ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$). $\overline{\text{RTS}}$ is asynchronous with the data. $\overline{\text{RTS}}$ is typically used in conjunction with $\overline{\text{CD}}$. The MSC8101 FCC1 transmitter requests the receiver to send data by asserting $\overline{\text{RTS}}$ low. The request is accepted when $\overline{\text{CTS}}$ is returned low.
	FCC1: CRS *MII*	Input	**FCC1: Media Independent Interface Carrier Sense** Asserted by an external fast Ethernet PHY to indicate activity on the cable.
PA29	FCC1: TXSOC *UTOPIA master*	Output	**FCC1: UTOPIA Transmit Start of Cell** Asserted by the MSC8101 (UTOPIA master PHY) when TXD[0–7] contains the first valid byte of the cell.
	FCC1: TX_ER *MII*	Output	**FCC1: Media Independent Interface Transmit Error** Asserted by the MSC8101 to force propagation of transmit errors.
PA28	FCC1: $\overline{\text{RXENB}}$ *UTOPIA master*	Output	**FCC1: UTOPIA Master Receive Enable** Asserted by the MSC8101 (UTOPIA master PHY) to indicate that RXD[0–7] and RXSOC are to be sampled at the end of the next cycle. RXD[0–7] and RXSOC are enabled only in cycles following those with $\overline{\text{RXENB}}$ asserted.
	FCC1: $\overline{\text{RXENB}}$ *UTOPIA slave*	Input	**FCC1: UTOPIA Master Receive Enable** Asserted by an external PHY to indicate that RXD[0–7] and RXSOC is to be sampled at the end of the next cycle. RXD[0–7] and RXSOC are enabled only in cycles following those with $\overline{\text{RXENB}}$ asserted.
	FCC1: TX_EN *MII*	Output	**FCC1: Media Independent Interface Transmit Enable** Asserted by the MSC8101 when transmitting data.

Table 1-7. Port A Signals (Continued)

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated Signal *Protocol***		
PA27	FCC1: RXSOC *UTOPIA slave*	Output	**FCC1: UTOPIA Receive Start of Cell** Asserted by the MSC8101 (UTOPIA slave) for an external PHY when RXD[0–7] contains the first valid byte of the cell.
	FCC1: RX_DV *MII*	Input	**FCC1: Media Independent Interface Receive Data Valid** Asserted by an external fast Ethernet PHY to indicate that valid data is being sent. The presence of carrier sense but not RX_DV indicates reception of broken packet headers, probably due to bad wiring or a bad circuit.
PA26	FCC1: RXCLAV *UTOPIA slave*	Output	**FCC1: UTOPIA Slave Receive Cell Available** Asserted by the MSC8101 (UTOPIA slave PHY) when one complete ATM cell is available for transfer.
	FCC1: RXCLAV *UTOPIA master, or*	Input	**FCC1: UTOPIA Master Receive Cell Available** Asserted by an external PHY when one complete ATM cell is available for transfer.
	RXCLAV0 *UTOPIA master, Multi-PHY, direct polling*	Input	**FCC1: UTOPIA Master Receive Cell Available 0 Direct Polling** Asserted by an external PHY when one complete ATM cell is available for transfer.
	FCC1: RX_ER *MII*	Input	**FCC1: Media Independent Interface Receive Error** Asserted by an external fast Ethernet PHY to indicate a receive error, which often indicates bad wiring.
PA25	FCC1: TXD0 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 0** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. TXD0 is the least significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	SDMA: MSNUM0	Output	**Module Serial Number Bit 0** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.
PA24	FCC1: TXD1 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 1** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 1 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	SDMA: MSNUM1	Output	**Module Serial Number Bit 1** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.
PA23	FCC1: TXD2 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 2** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 2 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.

Table 1-7. Port A Signals (Continued)

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated Signal *Protocol***		
PA22	FCC1: TXD3 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 3** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 3 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
PA21	FCC1: TXD4 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 4** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 4 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	FCC1: TXD3 *MII* and *HDLC nibble*	Output	**FCC1: MII and HDLC Nibble Transmit Data Bit 3** TXD[3–0] supports MII and HDLC nibble modes in FCC1. TXD3 is the most significant bit.
PA20	FCC1: TXD5 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 5** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 5 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	FCC1: TXD2 *MII* and *HDLC nibble*	Output	**FCC1: MII and HDLC Nibble Transmit Data Bit 2** TXD[3–0] is supported by MII and HDLC nibble modes in FCC1. This is bit 2 of the transmit data. TXD3 is the most significant bit.
PA19	FCC1: TXD6 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 6** The MSC8101MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. This is bit 6 of the transmit data. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	FCC1: TXD1 *MII* and *HDLC nibble*	Output	**FCC1: MII and HDLC Nibble Transmit Data Bit 1** TXD[3–0] is supported by MII and HDLC transparent nibble modes in FCC1. This is bit 1 of the transmit data. TXD3 is the most significant bit.
PA18	FCC1: TXD7 *UTOPIA*	Output	**FCC1: UTOPIA Transmit Data Bit 7.** The MSC8101 outputs ATM cell octets (UTOPIA interface data) on TXD[0–7]. TXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes.
	FCC1: TXD0 *MII* and *HDLC nibble*	Output	**FCC1: MII and HDLC Nibble Transmit Data Bit 0** TXD[3–0] is supported by MII and HDLC nibble modes in FCC1. TXD0 is the least significant bit.
	FCC1: TXD *HDLC serial* and *transparent*	Output	**FCC1: HDLC Serial and Transparent Transmit Data Bit** This is the single transmit data bit in supported by HDLC serial and transparent modes.

Table 1-7. Port A Signals (Continued)

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	**Peripheral Controller: Dedicated Signal *Protocol***		
PA17	FCC1: RXD7 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 7.** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. RXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when RXENB is asserted.
	FCC1: RXD0 *MII* and *HDLC nibble*	Input	**FCC1: MII and HDLC Nibble Receive Data Bit 0** RXD[3–0] is supported by MII and HDLC nibble mode in FCC1. RXD0 is the least significant bit.
	FCC1: RXD *HDLC serial* and *transparent*	Input	**FCC1: HDLC Serial and Transparent Receive Data Bit** This is the single receive data bit supported by HDLC and transparent modes.
PA16	FCC1: RXD6 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 6.** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. This is bit 6 of the receive data. RXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when RXENB is asserted.
	FCC1: RXD1 *MII* and *HDLC nibble*	Input	**FCC1: MII and HDLC Nibble Receive Data Bit 1** This is bit 1 of the receive nibble data. RXD3 is the most significant bit.
PA15	FCC1: RXD5 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 5** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. This is bit 5 of the receive data. RXD7 is the most significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when RXENB is asserted.
	RXD2 *MII* and *HDLC nibble*	Input	**FCC1: MII and HDLC Nibble Receive Data Bit 2** This is bit 2 of the receive nibble data. RXD3 is the most significant bit.
PA14	FCC1: RXD4 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 4.** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. RXD7 is the most significant bit. RXD0 is the least significant bit. When no ATM data is available, idle cells are inserted. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when RXENB is asserted.
	FCC1: RXD3 *MII* and *HDLC nibble*	Input	**FCC1: MII and HDLC Nibble Receive Data Bit 3** RXD3 is the most significant bit of the receive nibble bit.
PA13	FCC1: RXD3 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 3** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. RXD7 is the most significant bit. RXD0 is the least significant bit. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when RXENB is asserted.
	SDMA: MSNUM2	Output	**Module Serial Number Bit 2** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.

Table 1-7. Port A Signals (Continued)

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated Signal *Protocol*		
PA12	FCC1: RXD2 *UTOPIA*	Input	**FCC1: UTOPIA Receive Data Bit 2** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. This is bit 2 of the receive data. RXD7 is the most significant bit. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when $\overline{\text{RXENB}}$ is asserted.
	SDMA: MSNUM3	Output	**Module Serial Number Bit 3** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.
PA11	FCC1: RXD1 *UTOPIA*	Input	**FCC1: UTOPIA RX Receive Data Bit 1** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. This is bit 1 of the receive data. RXD7 is the most significant bit. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when $\overline{\text{RXENB}}$ is asserted.
	SDMA: MSNUM4	Output	**Module Serial Number Bit 4** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.
PA10	FCC1: RXD0 *UTOPIA*	Input	**FCC1: UTOPIA RX Receive Data Bit 0** The MSC8101 inputs ATM cell octets (UTOPIA interface data) on RXD[0–7]. RXD0 is the least significant bit of the receive data. A cell is 53 bytes. To support Multi-PHY configurations, RXD[0–7] is tri-stated, enabled only when $\overline{\text{RXENB}}$ is asserted.
	SDMA: MSNUM5	Output	**Module Serial Number Bit 5** The MSNUM has 6 bits that identify devices using the serial DMA (SDMA) modules. MSNUM[0–4] is the sub-block code of the current peripheral controller using SDMA. MSNUM5 indicates the section, transmit (0) or receive (1), that is active during the transfer. The information is recorded in the SDMA transfer error registers.
PA9	SMC2: SMTXD	Output	**SMC2: Serial Management Transmit Data** The SMC interface consists of SMTXD, SMRXD, $\overline{\text{SMSYN}}$, and a clock. Not all signals are used for all applications. SMCs are full-duplex ports that supports three protocols or modes: UART, transparent, or general-circuit interface (GCI). See also PC15.
	SI1 TDMA1: L1TXD0 *TDM nibble*	Output	**Time-Division Multiplexing A1: Layer 1 Transmit Data Bit 0** L1TXD0 is the least significant bit of the TDM nibble data.
PA8	SMC2: SMRXD	Input	**SMC2: Serial Management Receive Data** The SMC interface consists of SMTXD, SMRXD, $\overline{\text{SMSYN}}$, and a clock. Not all signals are used for all applications. SMCs are full-duplex ports that supports three protocols or modes: UART, transparent, or general-circuit interface (GCI).
	SI1 TDMA1: L1RXD0 *TDM nibble*	Input	**Time-Division Multiplexing A1: Layer 1 Nibble Receive Data Bit 0** L1RXD0 is the least significant bit received in nibble mode.
	SI1 TDMA1: L1RXD *TDM serial*	Input	**Time-Division Multiplexing A1: Layer 1 Serial Receive Data** TDMA1 receives serial data from L1RXD.

Table 1-7. Port A Signals (Continued)

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated Signal *Protocol*		
PA7	SMC2: $\overline{\text{SMSYN}}$	Input	**SMC2: Serial Management Synchronization** The SMC interface consists of SMTXD, SMRXD, SMSYN, and a clock. Not all signals are used for all applications. SMCs are full-duplex ports that supports three protocols or modes: UART, transparent, or general-circuit interface (GCI).
	SI1 TDMA1: L1TSYNC *TDM nibble* and *TDM serial*	Input	**Time-Division Multiplexing A1: Layer 1 Transmit Synchronization** The synchronizing signal for the transmit channel. See the *Serial Interface with* time-slot assigner chapter in the *MSC8101 Reference Manual*.
PA6	SI1 TDMA1: L1RSYNC *TDM nibble* and *TDM serial*	Input	**Time-Division Multiplexing A1: Layer 1 Receive Synchronization.** The synchronizing signal for the receive channel.

1.6.2 Port B Signals

Table 1-8. Port B Signals

Name		Dedicated I/O Data Direction	Description
General-Purpose I/O	Peripheral Controller: Dedicated I/O *Protocol*		
PB31	FCC2: TX_ER *MII*	Output	**FCC2: Media Independent Interface Transmit Error** Asserted by the MSC8101 to force propagation of transmit errors.
	SCC2: RXD	Input	**SCC2: Receive Data** SCC2 receives serial data from RXD.
	SI2 TDMB2: L1TXD *TDM serial*	Output	**Time-Division Multiplexing B2: Layer 1 Transmit Data** TDMB2 transmits serial data out of L1TXD.
PB30	SCC2: TXD	Output	**SCC2: Transmit Data.** SCC2 transmits serial data out of TXD.
	FCC2: RX_DV *MII*	Input	**FCC2: Media Independent Interface Receive Data Valid** Asserted by an external fast Ethernet PHY to indicate that valid data is being sent. The presence of carrier sense, but not RX_DV, indicates reception of broken packet headers, probably due to bad wiring or a bad circuit.
	SI2 TDMB2: L1RXD *TDM serial*	Input	**Time-Division Multiplexing B2: Layer 1 Receive Data** TDMB2 receives serial data from L1RXD.
PB29	FCC2: TX_EN *MII*	Output	**FCC2: Media Independent Interface Transmit Enable** Asserted by the MSC8101 when transmitting data.
	SI2 TDMB2: L1RSYNC *TDM serial*	Input	**Time-Division Multiplexing B2: Layer 1 Receive Synchronization** The synchronizing signal for the receive channel.