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.

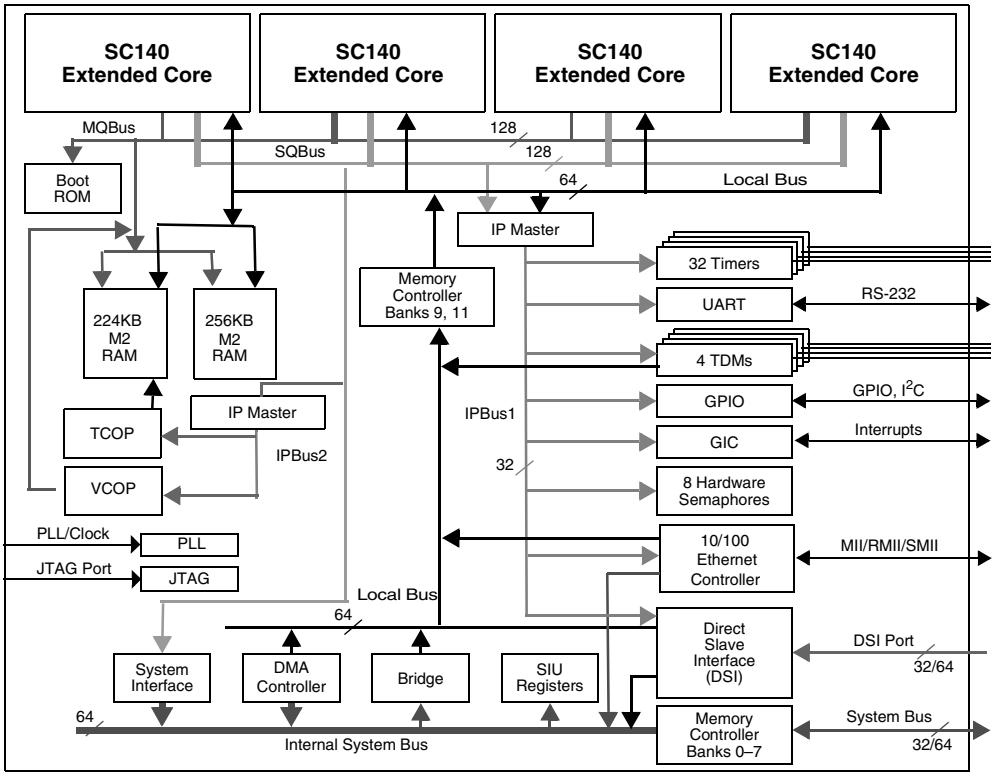
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

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

MSC8126

Quad Core 16-Bit Digital Signal Processor



Note: The arrows show the direction from which the transfer originates.

Figure 1. MSC8126 Block Diagram

The raw processing power of this highly integrated system-on-a-chip device will enable developers to create next-generation networking products that offer tremendous channel densities, while maintaining system flexibility, scalability, and upgradeability. The MSC8126 is offered in two core speed levels: 400 and 500 MHz.

What's New?

Rev. 11 includes the following:

- **Chapter 2** updates **Table 2-11** reset timing values.

The MSC8126 is a highly integrated system-on- a-chip that combines four SC140 extended cores with a turbo coprocessor (TCOP), a Viterbi coprocessor (VCOP), an RS-232 serial interface, four time-division multiplexed (TDM) serial interfaces, thirty-two general-purpose timers, a flexible system interface unit (SIU), an Ethernet interface, and a multi-channel DMA controller. The four extended cores can deliver a total 8000 DSP MMACS performance at 500 MHz. Each core has four arithmetic logic units (ALUs), internal memory, a write buffer, and two interrupt controllers (see **Figure 2**). The MSC8126 device targets high-bandwidth highly computational DSP applications and is optimized for wireless transcoding and packet telephony as well as high-bandwidth base station applications. The MSC8126 delivers enhanced performance while maintaining low power dissipation and greatly reducing system cost.

Table of Contents

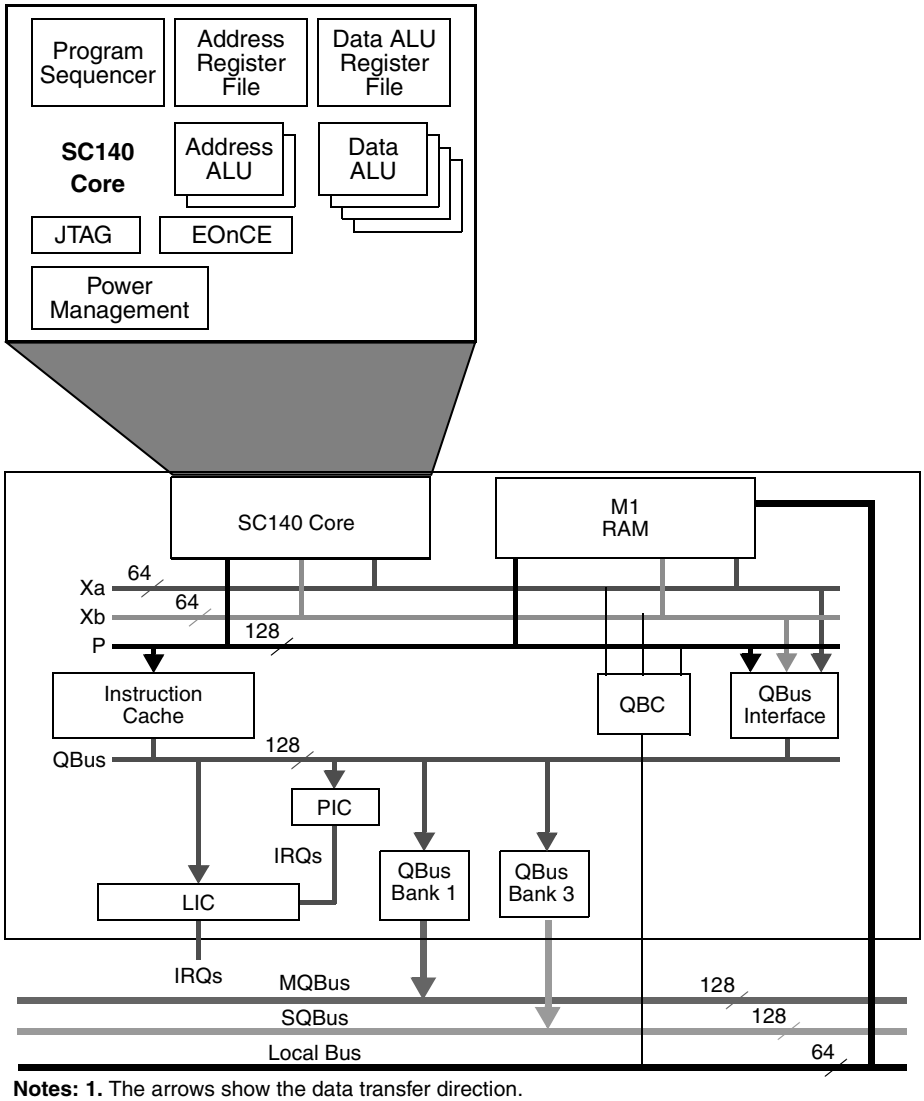
Data Sheet Conventions

$\overline{\text{OVERBAR}}$	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.



Notes: 1. The arrows show the data transfer direction.
2. The QBus interface includes a bus switch, write buffer, fetch unit, and a control unit that defines four QBus banks. In addition, the QBC handles internal memory contentions.

Figure 2. SC140 Extended Core Block Diagram

Features

The tables in this section list the features of the MSC8126 device.

Table 1. Extended SC140 Cores and Core Memories

Feature	Description
SC140 Core	Four SC140 cores: • Up to 8000 MMACS using 16 ALUs running at up to 500 MHz. • A total of 1436 KB of internal SRAM (224 KB per core + 16 KB ICache per core + the shared M2 memory). Each SC140 core provides the following: • Up to 2000 MMACS using an internal 500 MHz clock. A MAC operation includes a multiply-accumulate command with the associated data move and pointer update. • 4 ALUs per SC140 core. • 16 data registers, 40 bits each. • 27 address registers, 32 bits each. • Hardware support for fractional and integer data types. • Very rich 16-bit wide orthogonal instruction set. • Up to six instructions executed in a single clock cycle. • Variable-length execution set (VLES) that can be optimized for code density and performance. • JTAG port complies with **IEEE**® Std 1149.1™. • Enhanced on-device emulation (EOnCE) with real-time debugging capabilities.
Extended Core	Each SC140 core is embedded within an extended core that provides the following: • 224 KB M1 memory that is accessed by the SC140 core with zero wait states. • Support for atomic accesses to the M1 memory. • 16 KB instruction cache, 16 ways. • A four-entry write buffer that frees the SC140 core from waiting for a write access to finish. • External cache support by asserting the global signal (GBL) when predefined memory banks are accessed. • Programmable interrupt controller (PIC). • Local interrupt controller (LIC).
Multi-Core Shared Memories	• M2 memory (shared memory): —A 476 KB memory working at the core frequency. —Accessible from the local bus. —Accessible from all four SC140 cores using the MQBus. • 4 KB bootstrap ROM.
M2-Accessible Multi-Core Bus (MQBus)	• A QBus protocol multi-master bus connecting the four SC140 cores and the VCOP to the M2 memory. • Data bus access of up to 128-bit read and up to 64-bit write. • Operation at the SC140 core frequency. • A central efficient round-robin arbiter controlling SC140 core access on the MQBus. • Atomic operation control of access to M2 memory by the four SC140 cores and the local bus.

Table 2. Phase-Lock Loop (PLL)

Feature	Description
Internal PLL	• Generates up to 500 MHz core clock and up to 166 MHz bus clocks for the 60x-compatible local and system buses and other modules. • PLL values are determined at reset based on configuration signal values.

Table 3. Buses and Memory Controller

Feature	Description
60x-Compatible System Bus	• 64/32-bit data and 32-bit address 60x bus. • Support for multiple-master designs. • Four-beat burst transfers (eight-beat in 32-bit wide mode). • Port size of 64, 32, 16, and 8 controlled by the internal memory controller. • Bus can access external memory expansion or off-device peripherals, or it can enable an external host device to access internal resources. • Slave support, direct access by an external host to internal resources including the M1 and M2 memories. • On-device arbitration between up to four master devices.
Direct Slave Interface (DSI)	A 32/64-bit wide slave host interface that operates only as a slave device under the control of an external host processor. • 21–25 bit address, 32/64-bit data. • Direct access by an external host to internal and external resources, including the M1 and the M2 memories as well as external devices on the system bus. • Synchronous and asynchronous accesses, with burst capability in the synchronous mode. • Dual or single-strobe modes. • Write and read buffers improve host bandwidth. • Byte enable signals enables 1, 2, 4, and 8 byte write access granularity. • Sliding window mode enables access with reduced number of address pins. • Chip ID decoding enables using one $\overline{CS}$ signal for multiple DSPs. • Broadcast $\overline{CS}$ signal enables parallel write to multiple DSPs. • Big-endian, little-endian, and munged little-endian support.
3-Mode Signal Multiplexing	• 64-bit DSI, 32-bit system bus. • 32-bit DSI, 64-bit system bus. • 32-bit DSI, 32-bit system bus.
Memory Controller	Flexible eight-bank memory controller: • Three user-programmable machines (UPMs), general-purpose chip-select machine (GPCM), and a page-mode SDRAM machine. • Glueless interface to SRAM, page mode SDRAM, DRAM, EPROM, Flash memory, and other user-definable peripherals. • Byte enables for either 64-bit or 32-bit bus width mode. • Eight external memory banks (banks 0–7). Two additional memory banks (banks 9, 11) control IPBus peripherals and internal memories. Each bank has the following features: —32-bit address decoding with programmable mask. —Variable block sizes (32 KB to 4 GB). —Selectable memory controller machine. —Two types of data errors check/correction: normal odd/even parity and read-modify-write (RMW) odd/even parity for single accesses. —Write-protection capability. —Control signal generation machine selection on a per-bank basis. —Support for internal or external masters on the system bus. —Data buffer controls activated on a per-bank basis. —Atomic operation. —RMW data parity check (on system bus only). —Extensive external memory-controller/bus-slave support. —Parity byte select pin, which enables a fast, glueless connection to RMW-parity devices (on the system bus only). —Data pipeline to reduce data set-up time for synchronous devices.

Table 4. DMA Controller

Feature	Description
Multi-Channel DMA Controller	• 16 time-multiplexed unidirectional channels. • Services up to four external peripherals. • Supports $\overline{\text{DONE}}$ or $\overline{\text{DRACK}}$ protocol on two external peripherals. • Each channel group services 16 internal requests generated by eight internal FIFOs. Each FIFO generates: —A watermark request to indicate that the FIFO contains data for the DMA to empty and write to the destination. —A hungry request to indicate that the FIFO can accept more data. • Priority-based time-multiplexing between channels using 16 internal priority levels. • Round-robin time-multiplexing between channels. • A flexible channel configuration: —All channels support all features. —All channels connect to the system bus or local bus. • Flyby transfers in which a single data access is transferred directly from the source to the destination without using a DMA FIFO.

Table 5. Serial Interfaces

Feature	Description
Time-Division Multiplexing (TDM)	Up to four independent TDM modules, each with the following features: • Optional operating configurations: —Totally independent receive and transmit channels, each having one data line, one clock line, and one frame sync line. —Four data lines with one clock and one frame sync shared among the transmit and receive lines. • Connects gluelessly to most T1/E1 framers as well as to common buses such as the ST-BUS. • Hardware A-law/μ-law conversion. • Up to 62.5 Mbps per TDM (62.5 MHz bit clock if one data line is used, 31.25 MHz if two data lines are used, 15.63 MHz if four data lines are used). • Up to 256 channels. • Up to 16 MB per channel buffer (granularity 8 bytes), where A/μ law buffer size is double (granularity 16 byte). • Receive buffers share one global write offset pointer that is written to the same offset relative to their start address. • Transmit buffers share one global read offset pointer that is read from the same offset relative to their start address. • All channels share the same word size. • Two programmable receive and two programmable transmit threshold levels with interrupt generation that can be used, for example, to implement double buffering. • Each channel can be programmed to be active or inactive. • 2-, 4-, 8-, or 16-bit channels are stored in the internal memory as 2-, 4-, 8-, or 16-bit channels, respectively. • The TDM transmitter sync signal (TxTSYN) can be configured as either input or output. • Frame sync and data signals can be programmed to be sampled either on the rising edge or on the falling edge of the clock. • Frame sync can be programmed as active low or active high. • Selectable delay (0–3 bits) between the frame sync signal and the beginning of the frame. • MSB or LSB first support.

Table 5. Serial Interfaces (Continued)

Feature	Description
Ethernet Controller	• Complies with **IEEE** Std 802® including Std. 802.3™, 802.3u™, 802.3x™, and 802.3ac™. • Three Ethernet physical interfaces: —10/100 Mbps MII. —10/100 Mbps RMII. —10/100 Mbps SMII. • Full and half-duplex support. • Full-duplex flow control (automatic PAUSE frame generation or software programmed PAUSE frame generation and recognition). • Out-of-sequence transmit queue for initiating flow-control. • Programmable maximum frame length supports jumbo frames (up to 9.6 K) and virtual local area network (VLAN) tags and priority. • Retransmission from transmit FIFO following a collision. • CRC generation and verification of inbound/outbound packets. • Address recognition: —Each exact match can be programmed to be accepted or rejected. —Broadcast address (accept/reject). —Exact match 48-bit individual (unicast) address. —Hash (256-bit hash) check of individual (unicast) addresses. —Hash (256-bit hash) check of group (multicast) addresses. —Promiscuous mode. • Pattern matching: —Up to 16 unique 4-byte patterns. —Pattern match on bit-basis. —Matching range up to 256 bytes deep into the frame. —Offsets to a maximum of 252 bytes. —Programmable pattern size in 4-byte increments up to 64 bytes. —Accept or reject frames if a match is detected. —Up to eight unicast addresses for exact matches. —Pattern matching accepts/rejects IP addresses. • Filing of receive frames based on pattern match; prioritization of frames. • Insertion with expansion or replacement for transmit frames; VLAN tag insertion. • RMON statistics. • Master DMA on the local bus for fetching descriptors and accessing the buffers. • Ethernet PHY can be exposed either on GPIO pins or on the high most significant bits of the DSI/system when the DSI and the system bus are both 32 bits. • MPC8260 8-byte width buffer descriptor mode as well as 32-byte width buffer descriptor mode. • MII Bridge (MIIGSK): —Programmable selection of the 50 MHz RMII reference clock source (external or internal). —Independent 2 bit wide transmit and receive data paths. —Six operating modes. —Four general-purpose control signals. —Programmable transmitted inter-frame bits to support inter-frame gap for frames in the SMII domain. • SMII features: —Convey complete MII information between the PHY and MAC. —Allow direct MAC-to-MAC communication in SMII mode. —Can generate an interrupt request line while receiving inter-frame segments.

Table 5. Serial Interfaces (Continued)

Feature	Description
UART	• Two signals for transmit data and receive data. • No clock, asynchronous mode. • Can be serviced either by the SC140 DSP cores or an external host on the system bus or the DSI. • Full-duplex operation. • Standard mark/space non-return-to-zero (NRZ) format. • 13-bit baud rate selection. • Programmable 8-bit or 9-bit data format. • Separately enabled transmitter and receiver. • Programmable transmitter output polarity. • Two receiver wake-up methods: —Idle line wake-up. —Address mark wake-up. • Separate receiver and transmitter interrupt requests. • Nine flags, the first five can generate interrupt request: —Transmitter empty. —Transmission complete. —Receiver full. —Idle receiver input. —Receiver overrun. —Receiver active. —Noise error. —Framing error. —Parity error. • Receiver framing error detection. • Hardware parity checking. • 1/16 bit-time noise detection. • Maximum bit rate 6.25 Mbps. • Single-wire and loop operations.
General-Purpose I/O (GPIO) Port	• 32 bidirectional signal lines that either serve the peripherals or act as programmable I/O ports. • Each port can be programmed separately to serve up to two dedicated peripherals, and each port supports open-drain output mode.
I^2C Software Module	• Supports booting from a serial EEPROM • Uses GPIO timing.

Table 6. Miscellaneous Modules

Feature	Description
Timers	Two modules of 16 timers each. • Cyclic or one-shot. • Input clock polarity control. • Interrupt request when counting reaches a programmed threshold. • Pulse or level interrupts. • Dynamically updated programmed threshold. • Read counter any time. Watchdog mode for the timers that connect to the device.
Hardware Semaphores	Eight coded hardware semaphores, locked by simple write access without need for read-modify-write mechanism.
Global Interrupt Controller (GIC)	• Consolidation of chip maskable interrupt and non-maskable interrupt sources and routing to $\overline{\text{INT\_OUT}}$, $\overline{\text{NMI\_OUT}}$, and to the cores. • Generation of 32 virtual interrupts (eight to each SC140 core) by a simple write access. • Generation of virtual $\overline{\text{NMI}}$ (one to each SC140 core) by a simple write access.

Table 7. Coprocessors

Feature	Description
VCOP	• Fully programmable feed-forward channel decoding, feed-forward channel equalization and traceback sessions. • Up to 400 3GPP 12.2kbps AMR channels (channel decoding, number of channels linear to frequency). • Up to 200 blind transport format detect (BTFD) channels according to the 3GPP standard. Number of channels linear to frequency. • For channel decoding: —Constraint length between K = 5 and K = 9. —Puncture Codes. —Rate 1/2, 1/3, 1/4 and 1/6. —Four fully programmable polynomials (rate 1/6 is implemented by three polynomials only). —History buffer with up to 768 stages for 3G standards. —Input symbols are 8-bit (256 levels) signed soft symbols. —Output is hard decision (1-bit). • For GSM channel equalization: —Fully programmable 4 to 6 estimated channel autocorrelation coefficients (S-Parameters). —History buffer with up to 4090 stages for GSM. —Matched filter input is 8-bit (256 levels). —SOVA assist algorithm. —Output 8-bit coded delta values for SOVA assist algorithm, 1-bit hard decision traceback and history buffer or recursive traceback. • Fully programmable block length for all sessions. • Programmable traceback methods of Max Path, Min Path or End State. • Programmable learning period length for the traceback session. • Supports the start of feed-forward according to a presaved PM memory content. However the history buffer is not saved. Therefore the traceback is according to the current block only. • Each SC140 can program the VCOP parameters while the VCOP is in IDLE mode and then the VCOP can run independently on the whole block of data. • Dumping path metrics to the internal memory on up to 12 predefined stages; this is needed for BTFD applications. • Interrupt lines and status bits notify the cores on session completion. • Performance monitoring unit with 6 monitored behaviors.
TCOP	• Full support of 3GPP and CDMA2000 standards in Turbo decode. • Up to 20 turbo-coding 384 kbps channels. • 8 state PCCC with polynomial as supported by the 3G standards. • Iterative decoding structure based on Maximum A-Posteriori probability (MAP), with calculations performed in the LOG domain. • Encoding rate of 1/2, 1/3, 1/4, 1/5 with programmable puncturing for the parity symbols. • Full flexibility interleave function via a look-up table. • Flexible block size (1–32767 bits). • MAX log MAP and log MAP (MAX*) approximation. • Programmable MAX* using linear approximation. • Programmable number of iterations, with resolution of half iteration (one MAP). • Fully automatic execution when the GO command executes. • High data rates (for multi-channel systems or multiple channel accumulating to high data rates). • Can stop processing after every MAP when soft lambda all reach a programmable quality threshold. • Minimum and maximum number of iterations to execute in conjunction with the stop criteria. • The SC140 core or host can stop the processing after every MAP during run time. • Automatic, internal normalization for α, β overflow handling, with zero overhead. • Automatic, internal Λ clipping for Λ overflow handling, with zero overhead. • Additional least significant bit in α, β, γ arithmetic guarding against precision loss during the gamma calculation due to the division by 2.

Table 8. Power and Packaging

Feature	Description
Reduced Power Dissipation	• Low-power CMOS design. • Separate power supply for internal logic (1.2 V for 400 MHz or 500 MHz) and I/O (3.3 V). • Low-power standby modes. • Optimized power management circuitry (instruction-dependent, peripheral-dependent, and mode-dependent).
Packaging	• 0.8 mm pitch Flip-Chip Plastic Ball-Grid Array (FC-PBGA). • 431-connection (ball). • 20 mm × 20 mm.

Table 9. Software Support

Feature	Description
Real-Time Operating System (RTOS)	The real-time operating system (RTOS) fully supports device architecture (multi-core, memory hierarchy, ICache, timers, DMA controller, interrupts, peripherals), as follows: • High-performance and deterministic, delivering predictive response time. • Optimized to provide low interrupt latency with high data throughput. • Preemptive and priority-based multitasking. • Fully interrupt/event driven. • Small memory footprint. • Comprehensive set of APIs.
Multi-Core Support	• One instance of kernel code in all four SC140 cores. • Dynamic and static memory allocation from local memory (M1) and shared memory (M2).
Distributed System Support	Transparent inter-task communications between tasks running inside the SC140 cores and the other tasks running in on-board devices or remote network devices: • Messaging mechanism between tasks using mailboxes and semaphores. • Networking support; data transfer between tasks running inside and outside the device using networking protocols. • Includes integrated device drivers for such peripherals as TDM, UART, and external buses.
Software Support	• Task debugging utilities integrated with compilers and vendors. • Board support package (BSP) for the application development system (ADS). • Integrated development environment (IDE): —C/C++ compiler with in-line assembly so developers can generate highly optimized DSP code. Translates C/C++ code into parallel fetch sets and maintains high code density. —Librarian. User can create libraries for modularity. —A collection of C/C++ functions for developer use. —Highly efficient linker to produce executables from object code. —Seamlessly integrated real-time, non-intrusive multi-mode debugger for debugging highly optimized DSP algorithms. The developer can choose to debug in source code, assembly code, or mixed mode. —Device simulation models enable design and simulation before hardware availability. —Profiler using a patented binary code instrumentation (BCI) technique helps developers identify program design inefficiencies. —Version control. Metrowerks® CodeWarrior® includes plug-ins for ClearCase, Visual SourceSafe, and CVS.
Boot Options	• External memory. • External host. • UART. • TDM. • I^2C

Table 10. Application Development System (ADS) Board

Feature	Description
MSC8126ADS	• Host debug through single JTAG connector supports both processors. • MSC8103 as the MSC8126 host with both devices on the board. The MSC8103 system bus connects to the MSC8126 DSI. • Flash memory for stand-alone applications. • Communications ports: —10/100Base-T. —155 Mbit ATM over Optical. —T1/E1 TDM interface. —H.110. —Voice codec. —RS-232. —High-density (MICTOR) logic analyzer connectors to monitor MSC8126 signals —6U CompactPCI form factor. • Emulates MSC8126 DSP farm by connecting to three other ADS boards.

Product Documentation

The documents listed in **Table 11** are required for a complete description of the MSC8126 and are necessary to design properly with the part. Documentation is available from a local Freescale distributor, a Freescale Semiconductor sales office, or a Freescale Literature Distribution Center. For documentation updates, visit the Freescale DSP website. See the contact information on the back of this document.

Table 11. MSC8126 Documentation

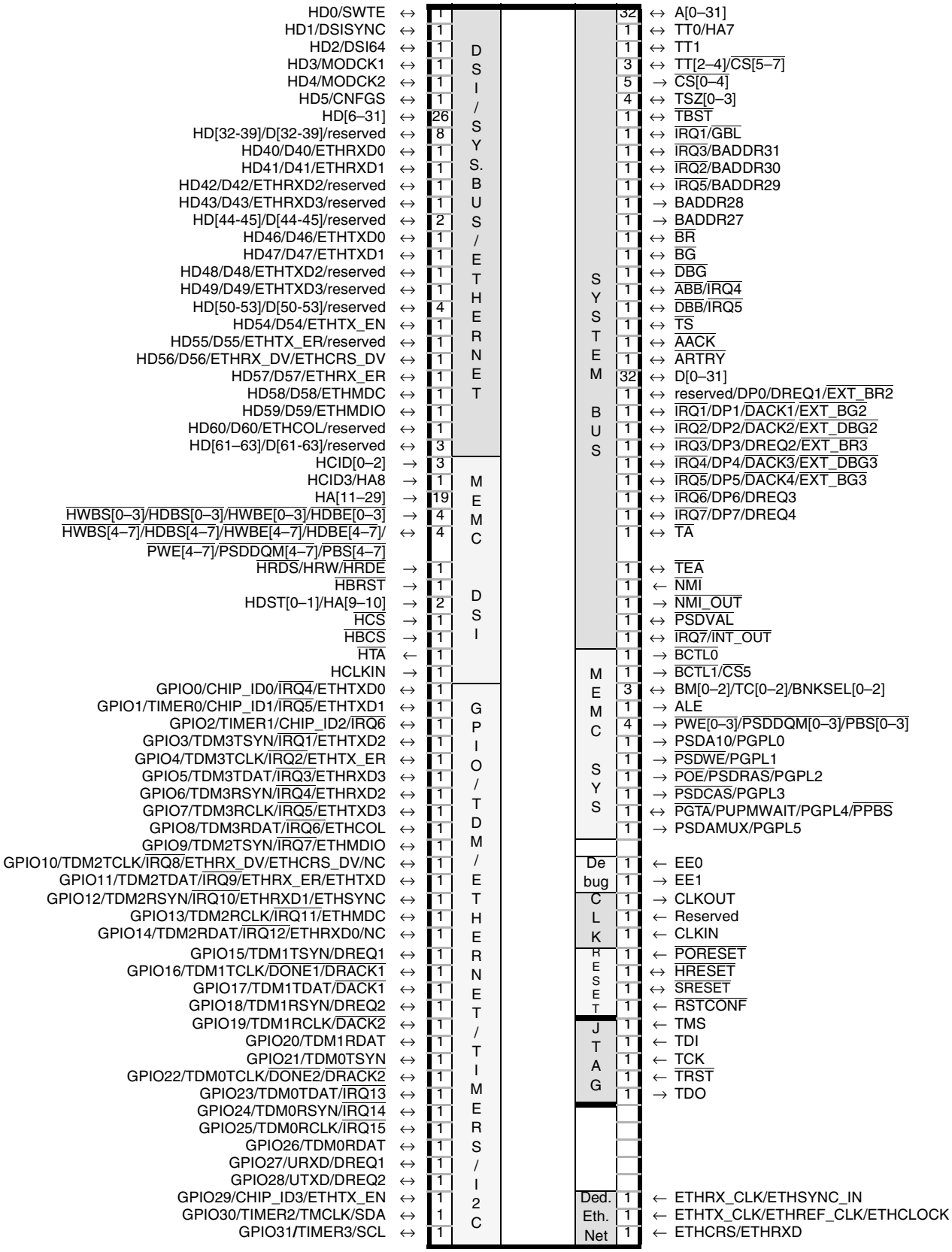
Name	Description	Order Number
MSC8126 Technical Data	MSC8126 features list and physical, electrical, timing, and package specifications.	MSC8126
MSC8126 User's Guide	User information includes system functionality, getting started, and programming topics.	Availability TBD
MSC8126 Reference Manual	Detailed functional description of the MSC8126 memory and peripheral configuration, operation, and register programming.	MSC8126RM
StarCore™ SC140 DSP Core Reference Manual	Detailed description of the SC140 family processor core and instruction set.	MNSC140CORE
Application Notes	Documents describing specific applications or optimized device operation including code examples.	See the website product page.

Signals/Connections 1

The MSC8126 external signals are organized into functional groups, as shown in **Table 1-1** and **Figure 1-1**. **Table 1-1** lists the functional groups, the number of signal connections in each group, and references the table that gives a detailed listing of multiplexed signals within each group. **Figure 1-1** shows MSC8126 external signals organized by function.

Table 1-1. MSC8126 Functional Signal Groupings

Functional Group	Number of Signal Connections	Description
Power (V_{DD}, V_{CC}, and GND)	155	**Table 1-2** on page 1-3
Clock	3	**Table 1-3** on page 1-3
Reset and configuration	4	**Table 1-4** on page 1-3
DSI, system bus, Ethernet, and interrupts	210	**Table 1-5** on page 1-4
Memory controller	16	**Table 1-6** on page 1-14
General-purpose input/output (GPIO), time-division multiplexed (TDM) interface, universal asynchronous receiver/ transmitter (UART), Ethernet, and timers	32	**Table 1-7** on page 1-16
Dedicated Ethernet signals	3	**Table 1-8** on page 1-23
EOnCE and JTAG test access port	7	**Table 1-9** on page 1-24
Reserved (denotes connections that are always reserved)	1	**Table 1-10** on page 1-24



Power signals are: V_{DD}, V_{DDH}, V_{CCSYN}, GND, GND_H, and GND_{SYN}. Reserved signals can be left unconnected. NC signals must not be connected.

Figure 1-1. MSC8126 External Signals

1.1 Power Signals

Table 1-2. Power and Ground Signal Inputs

Signal 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.
GND	**System Ground** An isolated ground for the internal processing logic and I/O buffers. This connection must be tied externally to all chip ground connections, except GND_{SYN}. 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.

1.2 Clock Signals

Table 1-3. Clock Signals

Signal Name	Type	Signal Description
CLKIN	Input	**Clock In** Primary clock input to the MSC8126 PLL.
CLKOUT	Output	**Clock Out** The bus clock.
Reserved	Input	Reserved. Pull down to ground.

1.3 Reset and Configuration Signals

Table 1-4. Reset and Configuration Signals

Signal Name	Type	Signal Description
$\overline{\text{PORESET}}$	Input	**Power-On Reset** When asserted, this line causes the MSC8126 to enter power-on reset state.
$\overline{\text{RSTCONF}}$	Input	**Reset Configuration** Used during reset configuration sequence of the chip. A detailed explanation of its function is provided in the *MSC8126 Reference Manual*. This signal is sampled upon deassertion of $\overline{\text{PORESET}}$. **Note:** When $\overline{\text{PORESET}}$ is deasserted, the MSC8126 also samples the following signals: • BM[0–2]—Selects the boot mode. • MODCK[1–2]—Selects the clock configuration. • SWTE—Enables the software watchdog timer. • DSISYNC, DSI64, CNFGS, and CHIP_ID[0–3]—Configures the DSI. Refer to **Table 1-5** for details on these signals.
$\overline{\text{HRESET}}$	Input/Output	**Hard Reset** When asserted as an input, this signal causes the MSC8126 to enter the hard reset state. After the device enters a hard reset state, it drives the signal as an open-drain output.
$\overline{\text{SRESET}}$	Input/Output	**Soft Reset** When asserted as an input, this signal causes the MSC8126 to enter the soft reset state. After the device enters a soft reset state, it drives the signal as an open-drain output.

1.4 Direct Slave Interface, System Bus, Ethernet, and Interrupt Signals

The direct slave interface (DSI) is combined with the system bus because they share some common signal lines. Individual assignment of a signal to a specific signal line is configured through internal registers. **Table 1-5** describes the signals in this group.

Note: Although there are fifteen interrupt request (IRQ) connections to the core processors, there are multiple external lines that can connect to these internal signal lines. After reset, the default configuration enables only $\overline{\text{IRQ}}$[1–7], but includes two input lines each for $\overline{\text{IRQ}}$[1–3] and $\overline{\text{IRQ}}$7. The designer must select one line for each required interrupt and reconfigure the other external signal line or lines for alternate functions. Additional alternate IRQ lines and $\overline{\text{IRQ}}$[8–15] are enabled through the GPIO signal lines.

Table 1-5. DSI, System Bus, Ethernet, and Interrupt Signals

Signal Name	Type	Description
HD0	Input/ Output	**Host Data Bus 0** Bit 0 of the DSI data bus.
SWTE	Input	**Software Watchdog Timer Disable**. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD1	Input/ Output	**Host Data Bus 1** Bit 1 of the DSI data bus.
DSISYNC	Input	**DSI Synchronous** Distinguishes between synchronous and asynchronous operation of the DSI. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD2	Input/ Output	**Host Data Bus 2** Bit 2 of the DSI data bus.
DSI64	Input	**DSI 64** Defines the width of the DSI and SYSTEM Data buses. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD3	Input/ Output	**Host Data Bus 3** Bit 3 of the DSI data bus.
MODCK1	Input	**Clock Mode 1** Defines the clock frequencies. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD4	Input/ Output	**Host Data Bus 4** Bit 4 of the DSI data bus.
MODCK2	Input	**Clock Mode 2** Defines the clock frequencies. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD5	Input/ Output	**Host Data Bus 5** Bit 5 of the DSI data bus.
CNFGS	Input	**Configuration Source** One signal out of two that indicates reset configuration mode. It is sampled on the rising edge of $\overline{\text{PORESET}}$ signal.
HD[6–31]	Input/ Output	**Host Data Bus 6–31** Bits 6–31 of the DSI data bus.

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

Signal Name	Type	Description
HD[32–39]	Input/ Output	**Host Data Bus 32–39** Bits 32–39 of the DSI data bus.
D[32–39]	Input/ Output	**System Bus Data 32–39** For write transactions, the bus master drives valid data on this bus. For read transactions, the slave drives valid data on this bus.
Reserved	Input	If the Ethernet port is enabled and multiplexed with the DSI/System bus, these signals are reserved and can be left unconnected.
HD40	Input/ Output	**Host Data Bus 40** Bit 40 of the DSI data bus.
D40	Input/ Output	**System Bus Data 40** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRXD0	Input	**Ethernet Receive Data 0** In MII and RMII modes, bit 0 of the Ethernet receive data.
HD41	Input/ Output	**Host Data Bus 41** Bit 41 of the DSI data bus.
D41	Input/ Output	**System Bus Data 41** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRXD1	Input	**Ethernet Receive Data 1** In MII and RMII modes, bit 1 of the Ethernet receive data.
HD42	Input/ Output	**Host Data Bus 42** Bit 42 of the DSI data bus.
D42	Input/ Output	**System Bus Data 42** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRXD2	Input	**Ethernet Receive Data 2** In MII mode only, bit 2 of the Ethernet receive data.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD43	Input/ Output	**Host Data Bus 43** Bit 43 of the DSI data bus.
D43	Input/ Output	**System Bus Data 43** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRXD3	Input	**Ethernet Receive Data 3** In MII mode only, bit 3 of the Ethernet receive data.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD[44–45]	Input/ Output	**Host Data Bus 44–45** Bits 44–45 of the DSI data bus.
D[44–56]	Input/ Output	**System Bus Data 44–45** For write transactions, the bus master drives valid data on this bus. For read transactions, the slave drives valid data on this bus.
Reserved	Input	If the Ethernet port is enabled and multiplexed with the DSI/System bus, these signals are reserved and can be left unconnected.

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

Signal Name	Type	Description
HD46	Input/ Output	**Host Data Bus 46** Bit 46 of the DSI data bus.
D46	Input/ Output	**System Bus Data 46** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTXD0	Output	**Ethernet Transmit Data 0** In MII and RMII modes, bit 0 of the Ethernet transmit data.
HD47	Input/ Output	**Host Data Bus 47** Bit 47 of the DSI data bus.
D47	Input/ Output	**System Bus Data 47** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTXD1	Output	**Ethernet Transmit Data 1** In MII and RMII modes, bit 1 of the Ethernet transmit data.
HD48	Input/ Output	**Host Data Bus 48** Bit 48 of the DSI data bus.
D48	Input/ Output	**System Bus Data 48** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTXD2	Output	**Ethernet Transmit Data 2** In MII mode only, bit 2 of the Ethernet transmit data.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD49	Input/ Output	**Host Data Bus 49** Bit 49 of the DSI data bus.
D49	Input/ Output	**System Bus Data 49** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTXD3	Output	**Ethernet Transmit Data 3** In MII mode only, bit 3 of the Ethernet transmit data.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD[50–53]	Input/ Output	**Host Data Bus 50–53** Bits 50–53 of the DSI data bus.
D[50–53]	Input/ Output	**System Bus Data 50–53** For write transactions, the bus master drives valid data on this bus. For read transactions, the slave drives valid data on this bus.
Reserved	Input	If the Ethernet port is enabled and multiplexed with the DSI/System bus, these signals are reserved and can be left unconnected.
HD54	Input/ Output	**Host Data Bus 54** Bit 54 of the DSI data bus.
D54	Input/ Output	**System Bus Data 54** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTX_EN	Output	**Ethernet Transmit Data Enable** In MII and RMII modes, indicates that the transmit data is valid.

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

Signal Name	Type	Description
HD55	Input/ Output	**Host Data Bus 55** Bit 55 of the DSI data bus.
D55	Input/ Output	**System Bus Data 55** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHTX_ER	Output	**Ethernet Transmit Data Error** In MII mode only, indicates a transmit data error.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD56	Input/ Output	**Host Data Bus 56** Bit 56 of the DSI data bus.
D56	Input/ Output	**System Bus Data 56** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRX_DV	Input	**Ethernet Receive Data Valid** Indicates that the receive data is valid.
ETHCRS_DV	Input	**Ethernet Carrier Sense/Receive Data Valid** In RMII mode, indicates that a carrier is detected and after the connection is established that the receive data is valid.
HD57	Input/ Output	**Host Data Bus 57** Bit 57 of the DSI data bus.
D57	Input/ Output	**System Bus Data 57** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHRX_ER	Input	**Ethernet Receive Data Error** In MII and RMII modes, indicates a receive data error.
HD58	Input/ Output	**Host Data Bus 58** Bit 58 of the DSI data bus.
D58	Input/ Output	**System Bus Data 58** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHMDC	Output	**Ethernet Management Clock** In MII and RMII modes, used for the MDIO reference clock.
HD59	Input/ Output	**Host Data Bus 59** Bit 59 of the DSI data bus.
D59	Input/ Output	**System Bus Data 59** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHMDIO	Input/ Output	**Ethernet Management Data** In MII and RMII modes, used for station management data input/output.

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

Signal Name	Type	Description
HD60	Input/ Output	**Host Data Bus 60** Bit 60 of the DSI data bus.
D60	Input/ Output	**System Bus Data 60** For write transactions, the bus master drives valid data on this line. For read transactions, the slave drives valid data on this bus.
ETHCOL	Input/ Output	**Ethernet Collision** In MII mode only, indicates that a collision was detected.
Reserved	Input	In RMII mode, this signal is reserved and can be left unconnected.
HD[61–63]	Input/ Output	**Host Data Bus 61–63** Bits 61–63 of the DSI data bus.
D[61–63]	Input/ Output	**System Bus Data 61–63** For write transactions, the bus master drives valid data on this bus. For read transactions, the slave drives valid data on this bus.
Reserved	Input	If the Ethernet port is enabled and multiplexed with the DSI/System bus, these signals are reserved and can be left unconnected.
HCID[0–2]	Input	**Host Chip ID 0–2** With HCID3, carries the chip ID of the DSI. The DSI is accessed only if $\overline{\text{HCS}}$ is asserted and HCID[0–3] matches the Chip_ID, or if $\overline{\text{HBCS}}$ is asserted.
HCID3	Input	**Host Chip ID 3** With HCI[0–2], carries the chip ID of the DSI. The DSI is accessed only if $\overline{\text{HCS}}$ is asserted and HCID[0–3] matches the Chip_ID, or if $\overline{\text{HBCS}}$ is asserted.
HA8	Input	**Host Bus Address 8** Used by an external host to access the internal address space.
HA[11–29]	Input	**Host Bus Address 11–29** Used by external host to access the internal address space.
$\overline{\text{HWBS[0–3]}}$	Input	**Host Write Byte Strobes** (In Asynchronous dual mode) One bit per byte is used as a strobe for host write accesses.
$\overline{\text{HDBS[0–3]}}$	Input	**Host Data Byte Strobe** (in Asynchronous single mode) One bit per byte is used as a strobe for host read or write accesses
$\overline{\text{HWBE[0–3]}}$	Input	**Host Write Byte Enable** (In Synchronous dual mode) One bit per byte is used to indicate a valid data byte for host read or write accesses.
$\overline{\text{HDBE[0–3]}}$	Input	**Host Data Byte Enable** (in Synchronous single mode) One bit per byte is used as a strobe enable for host write accesses

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

Signal Name	Type	Description
HWBS[4–7]	Input	**Host Write Byte Strobes** (In Asynchronous dual mode) One bit per byte is used as a strobe for host write accesses.
HDBS[4–7]	Input	**Host Data Byte Strobe** (in Asynchronous single mode) One bit per byte is used as a strobe for host read or write accesses
HWBE[4–7]	Input	**Host Write Byte Enable** (In Synchronous dual mode) One bit per byte is used to indicate a valid data byte for host write accesses.
HDBE[4–7]	Input	**Host Data Byte Enable** (in Synchronous single mode) One bit per byte is used as a strobe enable for host read or write accesses
PWE[4–7]	Output	**System Bus Write Enable** Outputs of the bus general-purpose chip-select machine (GPCM). These signals select byte lanes for write operations.
PSDDQM[4–7]	Output	**System Bus SDRAM DQM** From the SDRAM control machine. These signals select specific byte lanes of SDRAM devices.
PBS[4–7]	Output	**System Bus UPM Byte Select** From the UPM in the memory controller, these signals select specific byte lanes during memory operations. The timing of these signals 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.
HRDS	Input	**Host Read Data Strobe** (In Asynchronous dual mode) Used as a strobe for host read accesses.
HRW	Input	**Host Read/Write Select** (in Asynchronous/Synchronous single mode) Host read/write select.
HRDE	Input	**Host Read Data Enable** (In Synchronous dual mode) Indicates valid data for host read accesses.
HBRST	Input	**Host Burst** The host asserts this signal to indicate that the current transaction is a burst transaction in synchronous mode only.
HDST[0–1]	Input	**Host Data Structure 0–1** Defines the data structure of the host access in DSI little-endian mode.
HA[9–10]		**Host Bus Address 9–10** Used by an external host to access the internal address space.
HCS	Input	**Host Chip Select** DSI chip select. The DSI is accessed only if HCS is asserted and HCID[0–3] matches the Chip_ID.
HBCS	Input	**Host Broadcast Chip Select** DSI chip select for broadcast mode. Enables more than one DSI to share the same host chip-select signal for broadcast write accesses.
HTA	Output	**Host Transfer Acknowledge** Upon a read access, indicates to the host when the data on the data bus is valid. Upon a write access, indicates to the host that the data on the data bus was written to the DSI write buffer.
HCLKIN	Input	**Host Clock Input** Host clock signal for DSI synchronous mode.
A[0–31]	Input/ Output	**Address Bus** When the MSC8126 is in external master bus mode, these signals function as the system address bus. The MSC8126 drives the address of its internal bus masters and responds to addresses generated by external bus masters. When the MSC8126 is in internal master bus mode, these signals are used as address lines connected to memory devices and are controlled by the MSC8126 memory controller.
TT0	Input/ Output	**Bus Transfer Type 0** The bus master drives this signals during the address tenure to specify the type of the transaction.
HA7		**Host Bus Address 7** Used by an external host to access the internal address space.

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

Signal Name	Type	Description
TT1	Input/ Output	**Bus Transfer Type 1** The bus master drives this signals during the address tenure to specify the type of the transaction. Some applications use only the TT1 signal, for example, from MSC8126 to MSC8126 or MSC8126 to MSC8101 and *vice versa*. In these applications, TT1 functions as read/write signal.
TT[2–4]	Input/ Output	**Bus Transfer Type 2–4** The bus master drives these signals during the address tenure to specify the type of the transaction.
$\overline{\text{CS}}$[5–7]	Output	**Chip Select 5–7** Enables specific memory devices or peripherals connected to the system bus.
$\overline{\text{CS}}$[0–4]	Output	**Chip Select 0–4** Enables specific memory devices or peripherals connected to the system bus.
TSZ[0–3]	Input/ Output	**Transfer Size 0–3** The bus master drives these signals 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 signal to indicate that the current transaction is a burst transaction (transfers eight words).
$\overline{\text{IRQ1}}$	Input	**Interrupt Request 1**[1] One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{\text{GBL}}$	Output	**Global**[1] When a master within the MSC8126 initiates a bus transaction, it drives this signal. This signal indicates whether the transfer is global and should be snooped by caches in the system.
$\overline{\text{IRQ3}}$	Input	**Interrupt Request 3**[1] One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
BADDR31	Output	**Burst Address 31**[1] One of five burst address outputs of the memory controller. These signals connect directly to burstable memory devices without internal address incrementors controlled by the MSC8126 memory controller.
$\overline{\text{IRQ2}}$	Input	**Interrupt Request 2**[1] One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
BADDR30	Output	**Burst Address 30**[1] One of five address outputs of the memory controller. These signals connect directly to burstable memory devices without internal address incrementors controlled by the MSC8126 memory controller.
$\overline{\text{IRQ5}}$	Input	**Interrupt Request 5**[1] One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
BADDR29	Output	**Bus Burst Address 29**[1] One of five burst address outputs of the memory controller. These signals connect directly to burstable memory devices without internal address incrementors controlled by the MSC8126 memory controller.
BADDR28	Output	**Burst Address 28** One of five burst address outputs of the memory controller. These signals connect directly to burstable memory devices without internal address incrementors controlled by the MSC8126 memory controller.
BADDR27	Output	**Burst Address 27** One of five burst address outputs of the memory controller. These signals connect directly to burstable memory devices without internal address incrementors controlled by the MSC8126 memory controller.
$\overline{\text{BR}}$	Input/ Output	**Bus Request**[2] When an external arbiter is used, the MSC8126 asserts this signal as an output to request ownership of the bus. When the MSC8126 controller is used as an internal arbiter, an external master asserts this signal as an input to request bus ownership.

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

Signal Name	Type	Description
$\overline{\text{BG}}$	Input/ Output	**Bus Grant**[2] When the MSC8126 acts as an internal arbiter, it asserts this signal as an output to grant bus ownership to an external bus master. When an external arbiter is used, it asserts this signal as an input to grant bus ownership to the MSC8126.
$\overline{\text{DBG}}$	Input/ Output	**Data Bus Grant**[2] When the MSC8126 acts as an internal arbiter, it asserts this signal as an output to grant data bus ownership to an external bus master. When an external arbiter is used, it asserts this signal as an input to grant data bus ownership to the MSC8126.
$\overline{\text{ABB}}$	Input/ Output	**Address Bus Busy**[1] The MSC8126 asserts this signal as an output for the duration of the address bus tenure. Following an $\overline{\text{AACK}}$, which terminates the address bus tenure, the MSC8126 deasserts $\overline{\text{ABB}}$ for a fraction of a bus cycle and then stops driving this signal. The MSC8126 does not assume bus ownership as long as it senses this signal is asserted as an input by an external bus master.
$\overline{\text{IRQ4}}$	Input	**Interrupt Request 4** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{\text{DBB}}$	Input/ Output	**Data Bus Busy**[1] The MSC8126 asserts this signal as an output for the duration of the data bus tenure. Following a $\overline{\text{TA}}$, which terminates the data bus tenure, the MSC8126 deasserts $\overline{\text{DBB}}$ for a fraction of a bus cycle and then stops driving this signal. The MSC8126 does not assume data bus ownership as long as it senses that this signal is asserted as an input by an external bus master.
$\overline{\text{IRQ5}}$	Input	**Interrupt Request 5** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
$\overline{\text{TS}}$	Input/ Output	**Bus Transfer Start** This signal indicates the beginning of a new address bus tenure. The MSC8126 asserts this signal when one of its internal bus masters begins an address tenure. When the MSC8126 senses that this signal is asserted by an external bus master, it responds to the address bus tenure as required (snoop if enabled, access internal MSC8126 resources, memory controller support).
$\overline{\text{AACK}}$	Input/ Output	**Address Acknowledge** A bus slave asserts this signal to indicate that it has identified the address tenure. This signal terminates the address tenure.
$\overline{\text{ARTRY}}$	Input/ Output	**Address Retry** This signal indicates whether the bus master should retry the bus transaction. An external master asserts this signal to enforce data coherency with its caches and to prevent deadlock situations.
D[0–31]	Input/ Output	**Data Bus Bits 0–31** 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.
Reserved	Input	The primary configuration selection (default after reset) is reserved.
DP0	Input/ Output	**System Bus Data Parity 0** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 0 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 0 and D[0–7].
DREQ1	Input	**DMA Request 1** Used by an external peripheral to request DMA service.
$\overline{\text{EXT\_BR2}}$	Input	**External Bus Request 2** An external master asserts this signal to request bus ownership from the internal arbiter.

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

Signal Name	Type	Description
$\overline{\text{IRQ1}}$	Input	**Interrupt Request 1** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP1	Input/ Output	**System Bus Data Parity** 1 The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 1 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 1 and D[8–15].
$\overline{\text{DACK1}}$	Output	**DMA Acknowledge 1** The DMA controller drives this output to acknowledge the DMA transaction on the bus.
$\overline{\text{EXT\_BG2}}$	Output	**External Bus Grant 2**[2] The MSC8126 asserts this signal to grant bus ownership to an external bus master.
$\overline{\text{IRQ2}}$	Input	**Interrupt Request 2** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP2	Input/ Output	**System Bus Data Parity 2** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 2 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 2 and D[16–23].
$\overline{\text{DACK2}}$	Output	**DMA Acknowledge 2** The DMA controller drives this output to acknowledge the DMA transaction on the bus.
$\overline{\text{EXT\_DBG2}}$	Output	**External Data Bus Grant 2**[2] The MSC8126 asserts this signal to grant data bus ownership to an external bus master.
$\overline{\text{IRQ3}}$	Input	**Interrupt Request 3** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP3	Input/ Output	**System Bus Data Parity 3** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 3 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 3 and D[24–31].
DREQ2	Input	**DMA Request 2** Used by an external peripheral to request DMA service.
$\overline{\text{EXT\_BR3}}$	Input	**External Bus Request 3**[2] An external master should assert this signal to request bus ownership from the internal arbiter.
$\overline{\text{IRQ4}}$	Input	**Interrupt Request 4** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP4	Input/ Output	**System Bus Data Parity 4** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 4 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 4 and D[32–39].
$\overline{\text{DACK3}}$	Output	**DMA Acknowledge 3** The DMA controller drives this output to acknowledge the DMA transaction on the bus.
$\overline{\text{EXT\_DBG3}}$	Output	**External Data Bus Grant 3**[2] The MSC8126 asserts this signal to grant data bus ownership to an external bus master.

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

Signal Name	Type	Description
$\overline{\text{IRQ5}}$	Input	**Interrupt Request 5** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP5	Input/ Output	**System Bus Data Parity 5** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 5 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 5 and D[40–47].
$\overline{\text{DACK4}}$	Output	**DMA Acknowledge 4** The DMA controller drives this output to acknowledge the DMA transaction on the bus.
$\overline{\text{EXT\_BG3}}$	Output	**External Bus Grant 3**[2] The MSC8126 asserts this signal to grant bus ownership to an external bus.
$\overline{\text{IRQ6}}$	Input	**Interrupt Request 6** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP6	Input/ Output	**System Bus Data Parity 6** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 6 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 6 and D[48–55].
DREQ3	Input	**DMA Request 3** Used by an external peripheral to request DMA service.
$\overline{\text{IRQ7}}$	Input	**Interrupt Request 7** One of fifteen external lines that can request a service routine, via the internal interrupt controller, from the SC140 core.
DP7	Input/ Output	**System Bus Data Parity 7** The agent that drives the data bus also drives the data parity signals. The value driven on the data parity 7 signal should give odd parity (odd number of ones) on the group of signals that includes data parity 7 and D[56–63].
DREQ4	Input	**DMA Request 4** Used by an external peripheral to request DMA service.
$\overline{\text{TA}}$	Input/ Output	**Transfer Acknowledge** Indicates that a data beat is valid on the data bus. For single-beat transfers, $\overline{\text{TA}}$ assertion indicates the termination of the transfer. For burst transfers, $\overline{\text{TA}}$ is asserted eight times to indicate the transfer of eight data beats, with the last assertion indicating the termination of the burst transfer.
$\overline{\text{TEA}}$	Input/ Output	**Transfer Error Acknowledge** This signal indicates a failure of the data tenure transaction.The masters within the MSC8126 monitor the state of this signal. The MSC8126 internal bus monitor can assert this signal if it identifies a bus transfer that does not complete.
$\overline{\text{NMI}}$	Input	**Non-Maskable Interrupt** When an external device asserts this line, it generates an non-maskable interrupt in the MSC8126, which is processed internally (default) or is directed to an external host for processing (see $\overline{\text{NMI\_OUT}}$).
$\overline{\text{NMI\_OUT}}$	Output	**Non-Maskable Interrupt Output** An open-drain signal driven from the MSC8126 internal interrupt controller. This output indicates whether a non-maskable interrupt is pending in the MSC8126 internal interrupt controller, waiting to be handled by an external host.
$\overline{\text{PSDVAL}}$	Input/ Output	**Port Size Data Valid** Indicates that a data beat is valid on the data bus. The difference between the $\overline{\text{TA}}$ signal and the $\overline{\text{PSDVAL}}$ signal is that the $\overline{\text{TA}}$ signal is asserted to indicate data transfer terminations, while the $\overline{\text{PSDVAL}}$ signal is asserted with each data beat movement. When $\overline{\text{TA}}$ is asserted, $\overline{\text{PSDVAL}}$ is always asserted. However, when $\overline{\text{PSDVAL}}$ is asserted, $\overline{\text{TA}}$ is not necessarily asserted. For example, if the DMA controller initiates a double word ($2 \times$ 64 bits) transaction to a memory device with a 32-bit port size, $\overline{\text{PSDVAL}}$ is asserted three times without $\overline{\text{TA}}$ and, finally, both signals are asserted to terminate the transfer.