



Preliminary Technical Data ADSP-TS201S

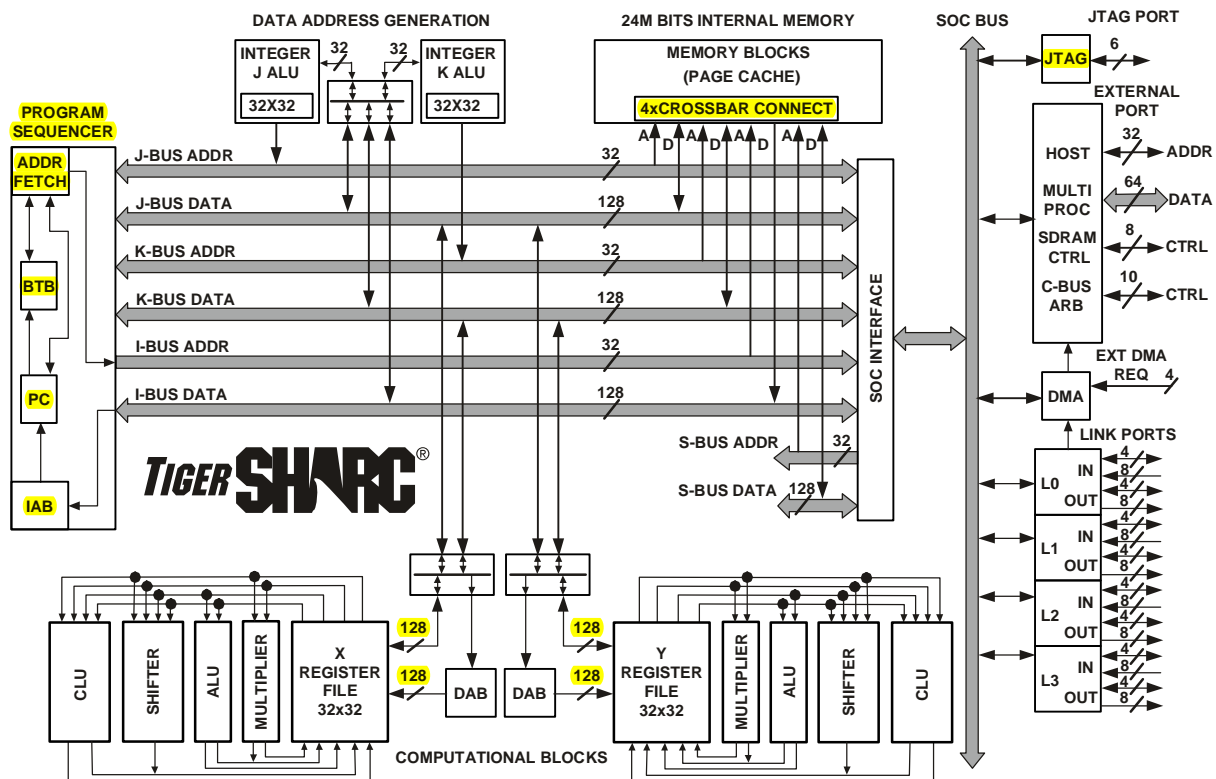
**KEY FEATURES**

- 500 MHz, 2.0 ns Instruction Cycle Rate
- 24M Bits of Internal—On-Chip—DRAM Memory
- 25×25 mm (576-Ball) Thermally Enhanced Ball Grid Array Package
- Dual Computation Blocks—Each Containing an ALU, a Multiplier, a Shifter, a Register File, and a Communications Logic Unit (CLU)
- Dual Integer ALUs, providing Data Addressing and Pointer Manipulation
- Integrated I/O Includes 14 Channel DMA Controller, External Port, Four Link Ports, SDRAM Controller, Programmable Flag Pins, Two Timers, and Timer Expired Pin for System Integration
- 1149.1 IEEE Compliant JTAG Test Access Port for On-Chip Emulation
- On-Chip Arbitration for Glueless Multiprocessing

**KEY BENEFITS**

- Provides High-Performance Static Superscalar DSP Operations, Optimized for Telecommunications Infrastructure and Other Large, Demanding Multiprocessor DSP Applications
- Performs Exceptionally Well on DSP Algorithm and I/O Benchmarks (See Benchmarks in Table 1)
- Supports Low-Overhead DMA Transfers Between Internal Memory, External Memory, Memory-Mapped Peripherals, Link Ports, Host Processors, and Other (Multiprocessor) DSPs
- Eases DSP Programming Through Extremely Flexible Instruction Set and High-Level-Language Friendly DSP Architecture
- Enables Scalable Multiprocessing Systems With Low Communications Overhead

**FUNCTIONAL BLOCK DIAGRAM**



TigerSHARC and the TigerSHARC logo are registered trademarks of Analog Devices, Inc.

REV. PrG

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

One Technology Way, P.O.Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781/329-4700 [www.analog.com](http://www.analog.com)  
Fax: 781/326-8703 © Analog Devices, Inc., 2003

TABLE OF CONTENTS

Key Features . . . . . 1  
 Key Benefits . . . . . 1  
 Table Of Contents . . . . . 2  
 General Description . . . . . 2  
     Dual Compute Blocks . . . . . 3  
     Data Alignment Buffer (DAB) . . . . . 4  
     Dual Integer ALUs (IALUs) . . . . . 4  
     Program Sequencer . . . . . 4  
         Interrupt Controller . . . . . 4  
         Flexible Instruction Set . . . . . 4  
     DSP Memory . . . . . 5  
 External Port  
     (Off-Chip Memory/Peripherals Interface) . . . . . 6  
     Host Interface . . . . . 6  
     Multiprocessor Interface . . . . . 6  
     SDRAM Controller . . . . . 6  
     EPROM Interface . . . . . 7  
 DMA Controller . . . . . 7  
 Link Ports (LVDS) . . . . . 8  
 Timer and General-Purpose I/O . . . . . 8  
 Reset and Booting . . . . . 8  
 Clock Domains . . . . . 9  
 Power Domains . . . . . 9  
 Filtering Reference Voltage and Clocks . . . . . 9  
 Development Tools . . . . . 9  
 Designing an Emulator-Compatible DSP Board  
     (Target) . . . . . 10  
     Additional Information . . . . . 11  
 Pin Function Descriptions . . . . . 11  
 Strap Pin Function Descriptions . . . . . 18  
 ADSP-TS201S—Specifications . . . . . 20  
     ABSOLUTE MAXIMUM RATINGS . . . . . 21  
     ESD SENSITIVITY . . . . . 21  
     Timing Specifications . . . . . 21  
         General AC Timing . . . . . 21  
         Link Port Low-Voltage, Differential-Signal (LVDS)  
             Electrical Characteristics and Timing . . . . . 26  
     Output Drive Currents . . . . . 31  
     Test Conditions . . . . . 32  
         Output Disable Time . . . . . 32  
         Output Enable Time . . . . . 32  
         Capacitive Loading . . . . . 33  
     Environmental Conditions . . . . . 34  
         Thermal Characteristics . . . . . 34  
 576-Ball BGA\_ED Pin Configurations . . . . . 36  
 Outline Dimensions . . . . . 40  
 Ordering Guide . . . . . 40

GENERAL DESCRIPTION

The ADSP-TS201S TigerSHARC processor is an ultra-high performance, static superscalar processor optimized for large signal processing tasks and communications infrastructure. The DSP combines very wide memory widths with dual computation blocks—supporting 32- and 40-bit floating-point and supporting 8-, 16-, 32-, and 64-bit fixed-point processing—to set a new standard of performance for digital signal processors. The TigerSHARC static superscalar architecture lets the DSP execute up to four instructions each cycle, performing twenty-four 16-bit fixed-point operations or six floating-point operations.

Four independent 128-bit wide internal data buses, each connecting to the six 4M bit memory banks, enable quad-word data, instruction, and I/O accesses and provide 28G bytes per second of internal memory bandwidth. Operating at 500 MHz, the ADSP-TS201S processor’s core has a 2.0 ns instruction cycle time. Using its Single-Instruction, Multiple-Data (SIMD) features, the ADSP-TS201S processor can perform four billion 40-bit MACs or one billion 80-bit MACs per second. Table 1 shows the DSP’s performance benchmarks.

Table 1. General Purpose Algorithm Benchmarks at 500 MHz

Benchmark	Speed	Clock Cycles
<b>32-bit Algorithm, 500 million MACs/s peak performance</b>		
1024 Point Complex FFT <sup>1</sup> (Radix 2)	20 μs	10061
FIR Filter (per real tap)	1 ns	0.5
[8 × 8][8 × 8] Matrix Multiply (Complex, Floating-point)	2.8 μs	1399
<b>16-bit Algorithm, 2 billion MACs/s peak performance</b>		
256 Point Complex FFT <sup>1</sup> (Radix 2)	1.9 μs	928
<b>I/O DMA Transfer Rate</b>		
External port	1G bytes/s	n/a
Link ports (each)	1G bytes/s	n/a

<sup>1</sup>Cache preloaded

The ADSP-TS201S processor is code-compatible with the other TigerSHARC processors.

The Functional Block Diagram on page 1 shows the ADSP-TS201S processor’s architectural blocks. These blocks include:

- Dual compute blocks, each consisting of an ALU, multiplier, 64-bit shifter, 128-bit CLU, and 32-word register file and associated Data Alignment Buffers (DABs)
- Dual integer ALUs (IALUs), each with its own 31-word register file for data addressing and a status register
- A program sequencer with Instruction Alignment Buffer (IAB) and Branch Target Buffer (BTB)
- An interrupt controller that supports hardware and software interrupts, supports level- or edge-triggers, and supports prioritized, nested interrupts

2 This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

June 2003

For current information contact Analog Devices at 800/262-5643

ADSP-TS201S

- Four 128-bit internal data buses, each connecting to the six 4M bit memory banks
- **On-chip DRAM (24M bit)**
- An external port that provides the interface to host processors, multiprocessing space (DSPs), off-chip memory-mapped peripherals, and external SRAM and SDRAM
- A 14 channel DMA controller
- **Four full-duplex LVDS link ports**
- Two 64-bit interval timers and timer expired pin
- A 1149.1 IEEE compliant JTAG test access port for on-chip emulation

Figure 1 on page 3 shows a typical single-processor system with external SRAM and SDRAM. Figure 3 on page 7 shows a typical multiprocessor system.

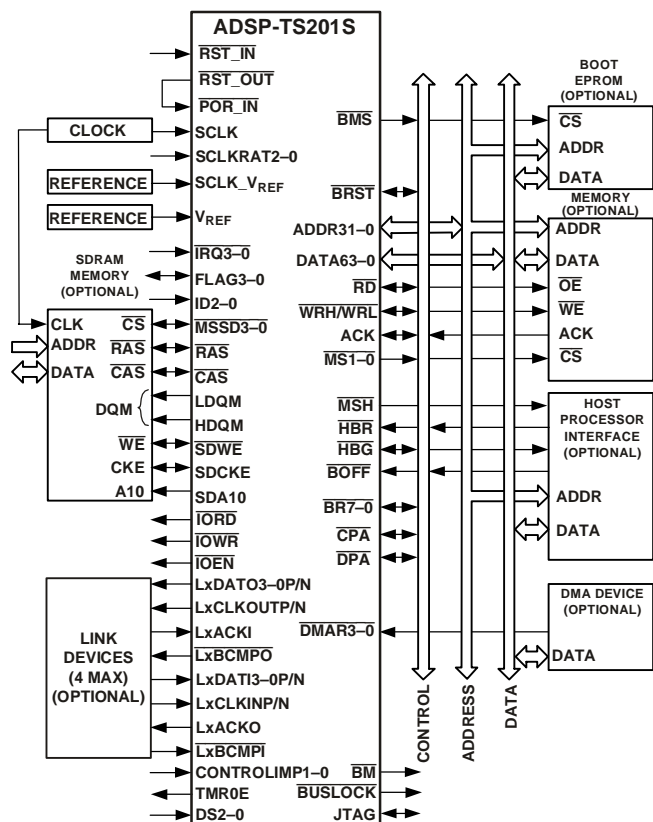


Figure 1. ADSP-TS201S Single-Processor System With External SDRAM

The TigerSHARC DSP uses a Static Superscalar<sup>1</sup> architecture. This architecture is superscalar in that the ADSP-TS201S processor's core can execute simultaneously from one to four 32-bit instructions encoded in a Very Large Instruction Word (VLIW) instruction line using the DSP's dual compute blocks. Because

the DSP does not perform instruction re-ordering at runtime—the programmer selects which operations will execute in parallel prior to runtime—the order of instructions is static.

With few exceptions, an instruction line, whether it contains one, two, three, or four 32-bit instructions, executes with a throughput of one cycle in a ten-deep processor pipeline.

For optimal DSP program execution, programmers must follow the DSP's set of instruction parallelism rules when encoding an instruction line. In general, the selection of instructions that the DSP can execute in parallel each cycle depends on the instruction line resources each instruction requires and on the source and destination registers used in the instructions. The programmer has direct control of three core components—the IALUs, the compute blocks, and the program sequencer.

The ADSP-TS201S processor, in most cases, has a two-cycle execution pipeline that is fully interlocked, so—whenever a computation result is unavailable for another operation dependent on it—the DSP automatically inserts one or more stall cycles as needed. Efficient programming with dependency-free instructions can eliminate most computational and memory transfer data dependencies.

In addition, the ADSP-TS201S processor supports SIMD operations two ways—SIMD compute blocks and SIMD computations. The programmer can load both compute blocks with the same data (broadcast distribution) or different data (merged distribution).

### Dual Compute Blocks

The ADSP-TS201S processor has compute blocks that can execute computations either independently or together as a Single-Instruction, Multiple-Data (SIMD) engine. The DSP can issue up to two compute instructions per compute block each cycle, instructing the ALU, multiplier, shifter, or CLU to perform independent, simultaneous operations. Each compute block can execute eight 8-bit, four 16-bit, two 32-bit, or one 64-bit SIMD computations in parallel with the operation in the other block.

The compute blocks are referred to as X and Y in assembly syntax, and each block contains four computational units—an ALU, a multiplier, a 64-bit shifter, a 128-bit CLU—and a 32-word register file.

- **Register File**—Each Compute Block has a multiported 32-word, fully orthogonal register file used for transferring data between the computation units and data buses and for storing intermediate results. Instructions can access the registers in the register file individually (word-aligned), in sets of two (dual-aligned), or in sets of four (quad-aligned).
- **ALU**—The ALU performs a standard set of arithmetic operations in both fixed- and floating-point formats. It also performs logic operations.
- **Multiplier**—The multiplier performs both fixed- and floating-point multiplication and fixed-point multiply and accumulate.

<sup>1</sup> Static Superscalar™ is a trademark of Analog Devices, Inc.

- **Shifter**—The 64-bit shifter performs logical and arithmetic shifts, bit and bitstream manipulation, and field deposit and extraction operations.
- **Communications Logic Unit (CLU)**—This is a 128-bit unit provides Trellis Decoding (for example, Viterbi and Turbo decoders) and executes complex correlations for CDMA communication applications (for example chip-rate and symbol-rate functions).

Using these features, the compute blocks can:

- Provide 8 MACs per cycle peak and 7.1 MACs per cycle sustained 16-bit performance and provide 2 MACs per cycle peak and 1.8 MACs per cycle sustained 32-bit performance (based on FIR)
- Execute six single-precision floating-point or execute twenty-four 16-bit fixed-point operations per cycle, providing 3 GFLOPS or 12.0 GOPS performance
- Perform two complex 16-bit MACs per cycle
- Execute eight Trellis butterflies in one cycle

#### Data Alignment Buffer (DAB)

The DAB is a quad-word FIFO that enables loading of quad-word data from nonaligned addresses. Normally, load instructions must be aligned to their data size so that quad words are loaded from a quad-aligned address. Using the DAB significantly improves the efficiency of some applications, such as FIR filters.

#### Dual Integer ALUs (IALUs)

The ADSP-TS201S processor has two IALUs that provide powerful address generation capabilities and perform many general-purpose integer operations. The IALUs are referred to as J and K in assembly syntax and have the following features:

- Provides memory addresses for data and update pointers
- Supports circular buffering and bit-reverse addressing
- Performs general-purpose integer operations, increasing programming flexibility
- Includes a 31-word register file for each IALU

As address generators, the IALUs perform immediate or indirect (pre- and post-modify) addressing. They perform modulus and bit-reverse operations with no constraints placed on memory addresses for the modulus data buffer placement. Each IALU can specify either a single-, dual-, or quad-word access from memory.

The IALUs have hardware support for circular buffers, bit reverse, and zero-overhead looping. Circular buffers facilitate efficient programming of delay lines and other data structures required in digital signal processing, and they are commonly used in digital filters and Fourier transforms. Each IALU provides registers for four circular buffers, so applications can set up a total of eight circular buffers. The IALUs handle address pointer wraparound automatically, reducing overhead, increasing performance, and simplifying implementation. Circular buffers can start and end at any memory location.

Because the IALU's computational pipeline is one cycle deep, in most cases integer results are available in the next cycle. Hardware (register dependency check) causes a stall if a result is unavailable in a given cycle.

#### Program Sequencer

The ADSP-TS201S processor's program sequencer supports the following:

- A fully interruptible programming model with flexible programming in assembly and C/C++ languages; handles hardware interrupts with high throughput and no aborted instruction cycles
- A ten-cycle instruction pipeline—four-cycle fetch pipe and six-cycle execution pipe—computation results available two cycles after operands are available
- Supply of instruction fetch memory addresses; the sequencer's Instruction Alignment Buffer (IAB) caches up to five fetched instruction lines waiting to execute; the program sequencer extracts an instruction line from the IAB and distributes it to the appropriate core component for execution
- Management of program structures and program flow determined according to JUMP, CALL, RTI, RTS instructions, loop structures, conditions, interrupts, and software exceptions
- Branch prediction and a 128-entry branch target buffer (BTB) to reduce branch delays for efficient execution of conditional and unconditional branch instructions and zero-overhead looping; correctly predicted branches that are taken occur with zero overhead cycles, overcoming the five-to-nine stage branch penalty
- Compact code without the requirement to align code in memory; the IAB handles alignment

#### Interrupt Controller

The DSP supports nested and nonnested interrupts. Each interrupt type has a register in the interrupt vector table. Also, each has a bit in both the interrupt latch register and the interrupt mask register. All interrupts are fixed as either level-sensitive or edge-sensitive, except the IRQ3-0 hardware interrupts, which are programmable.

The DSP distinguishes between hardware interrupts and software exceptions, handling them differently. When a software exception occurs, the DSP aborts all other instructions in the instruction pipe. When a hardware interrupt occurs, the DSP continues to execute instructions already in the instruction pipe.

#### Flexible Instruction Set

The 128-bit instruction line, which can contain up to four 32-bit instructions, accommodates a variety of parallel operations for concise programming. For example, one instruction line can direct the DSP to conditionally execute a multiply, an add, and

a subtract in both computation blocks while it also branches to another location in the program. Some key features of the instruction set include:

- CLU instructions for communications infrastructure to govern Trellis Decoding (for example, Viterbi and Turbo decoders) and Despreading via complex correlations
- Algebraic assembly language syntax
- Direct support for all DSP, imaging, and video arithmetic types
- Eliminates toggling DSP hardware modes because modes are supported as options (for example, rounding, saturation, and others) within instructions

- Branch prediction encoded in instruction; enables zero-overhead loops
- Parallelism encoded in instruction line
- Conditional execution optional for all instructions
- User defined partitioning between program and data memory

**DSP Memory**

The DSP's internal and external memory is organized into a unified memory map, which defines the location (address) of all elements in the system, as shown in Figure 2.

The memory map is divided into four memory areas—host space, external memory, multiprocessor space, and internal memory—and each memory space, except host memory, is subdivided into smaller memory spaces.

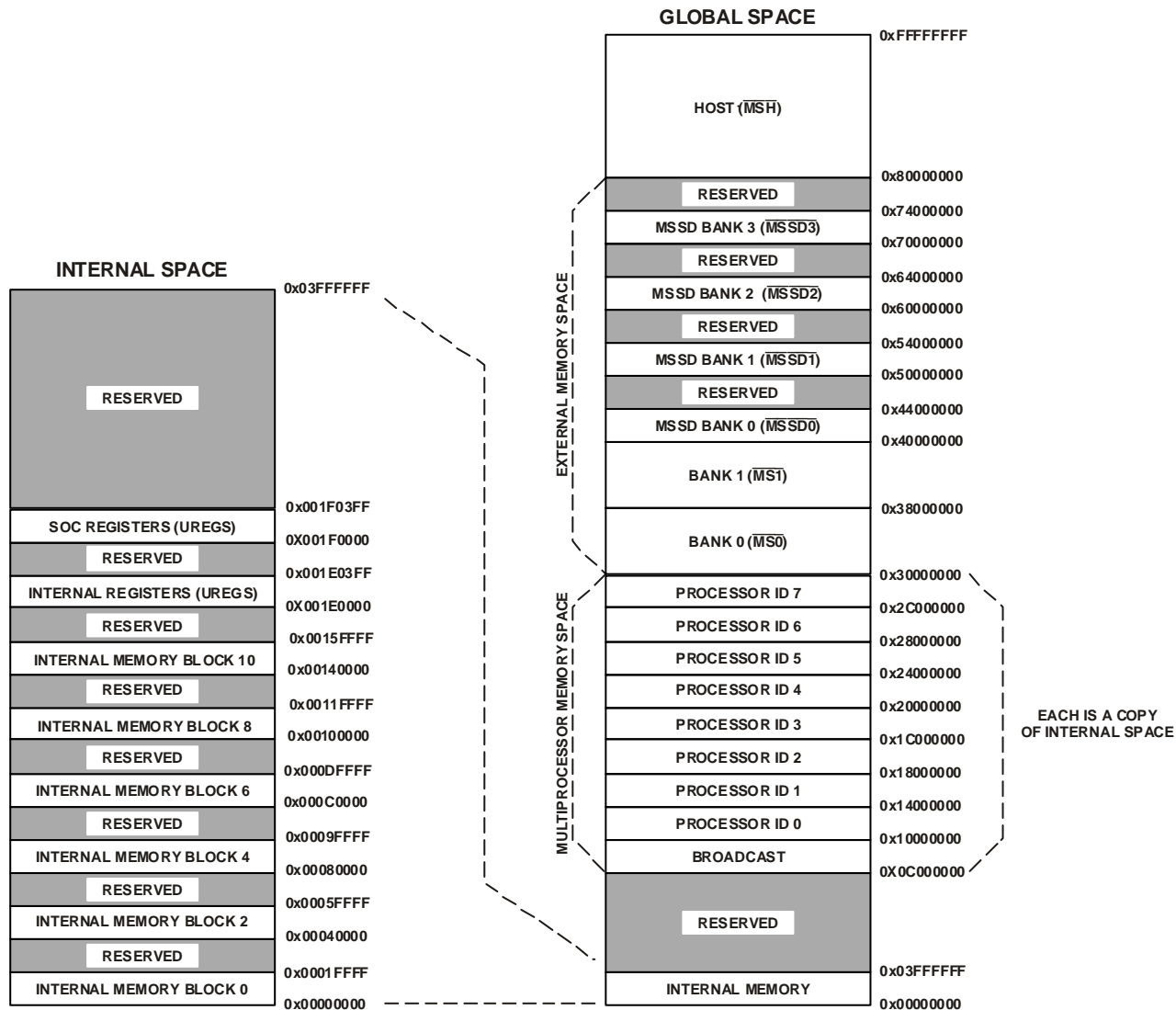


Figure 2. ADSP-TS201S Memory Map

The ADSP-TS201S processor internal memory has 24M bits of on-chip DRAM memory, divided into six blocks of 4M bits (128K words  $\times$  32 bits). Each block—M0, M2, M4, M6, M8, and M10—can store program, data, or both, so applications can configure memory to suit specific needs. **Placing program instructions and data in different memory blocks, however, enables the DSP to access data while performing an instruction fetch.** Each memory segment contains a 128K bit cache to enable single cycle accesses to internal DRAM.

The six internal memory blocks connect to the four 128-bit wide internal buses through a crossbar connection, enabling the DSP to perform four memory transfers in the same cycle. The DSP's internal bus architecture provides a total memory bandwidth of 28G bytes per second, enabling the core and I/O to access eight 32-bit data words and four 32-bit instructions each cycle. The DSP's flexible memory structure enables:

- DSP core and I/O accesses to different memory blocks in the same cycle
- DSP core access to three memory blocks in parallel—one instruction and two data accesses
- Programmable partitioning of program and data memory
- Program access of all memory as 32-, 64-, or 128-bit words—16-bit words with the DAB

#### External Port (Off-Chip Memory/Peripherals Interface)

The ADSP-TS201S processor's external port provides the DSP's interface to off-chip memory and peripherals. The 4G word address space is included in the DSP's unified address space. The separate on-chip buses—four 128-bit data buses and four 32-bit address buses—are multiplexed at the SOC interface and transferred to the external port over the SOC bus to create an external system bus transaction. The external system bus provides a single 64-bit data bus and a single 32-bit address bus. The external port supports data transfer rates of 1G bytes per second over the external bus.

The external bus can be configured for 32- or 64-bit, little-endian operations. When the system bus is configured for 64-bit operations, the lower 32 bits of the external data bus connect to even addresses, and the upper 32 bits connect to odd addresses.

The external port supports pipelined, slow, and SDRAM protocols. Addressing of external memory devices and memory-mapped peripherals is facilitated by on-chip decoding of high-order address lines to generate memory bank select signals.

The ADSP-TS201S processor provides programmable memory, pipeline depth, and idle cycle for synchronous accesses, and external acknowledge controls to support interfacing to pipelined or slow devices, host processors, and other memory-mapped peripherals with variable access, hold, and disable time requirements.

#### Host Interface

The ADSP-TS201S processor provides an easy and configurable interface between its external bus and host processors through the external port. To accommodate a variety of host processors,

the host interface supports pipelined or slow protocols for ADSP-TS201S processor accesses of the host as slave or pipelined for host accesses of the ADSP-TS201S processor as slave. Each protocol has programmable transmission parameters, such as idle cycles, pipe depth, and internal wait cycles.

The host interface supports burst transactions initiated by a host processor. After the host issues the starting address of the burst and asserts the  $\overline{\text{BRST}}$  signal, the DSP increments the address internally while the host continues to assert  $\overline{\text{BRST}}$ .

The host interface provides a deadlock recovery mechanism that enables a host to recover from deadlock situations involving the DSP. The  $\overline{\text{BOFF}}$  signal provides the deadlock recovery mechanism. When the host asserts  $\overline{\text{BOFF}}$ , the DSP backs off the current transaction and asserts  $\overline{\text{HBG}}$  and relinquishes the external bus.

The host can directly read or write the internal memory of the ADSP-TS201S processor, and it can access most of the DSP registers, including DMA control (TCB) registers. Vector interrupts support efficient execution of host commands.

#### Multiprocessor Interface

The ADSP-TS201S processor offers powerful features tailored to multiprocessing DSP systems through the external port and link ports. This multiprocessing capability provides highest bandwidth for interprocessor communication, including:

- **Up to eight DSPs on a common bus**
- **On-chip arbitration for glueless multiprocessing**
- Link ports for point to point communication

The external port and link ports provide integrated, glueless multiprocessing support.

The external port supports a unified address space (see [Figure 2](#)) that enables direct interprocessor accesses of each ADSP-TS201S processor's internal memory and registers. The DSP's on-chip distributed bus arbitration logic provides simple, glueless connection for systems containing up to eight ADSP-TS201S processors and a host processor. Bus arbitration has a rotating priority. Bus lock supports indivisible read-modify-write sequences for semaphores. A bus fairness feature prevents one DSP from holding the external bus too long.

The DSP's four link ports provide a second path for interprocessor communications with throughput of 4G bytes per second. The cluster bus provides 1G bytes per second throughput—with a total of 4G bytes per second interprocessor bandwidth (limited by SOC bandwidth).

#### SDRAM Controller

The SDRAM controller controls the ADSP-TS201S processor's transfers of data to and from external synchronous DRAM (SDRAM) at a throughput of 32 or 64 bits per SCLK cycle using the external port and SDRAM control pins.

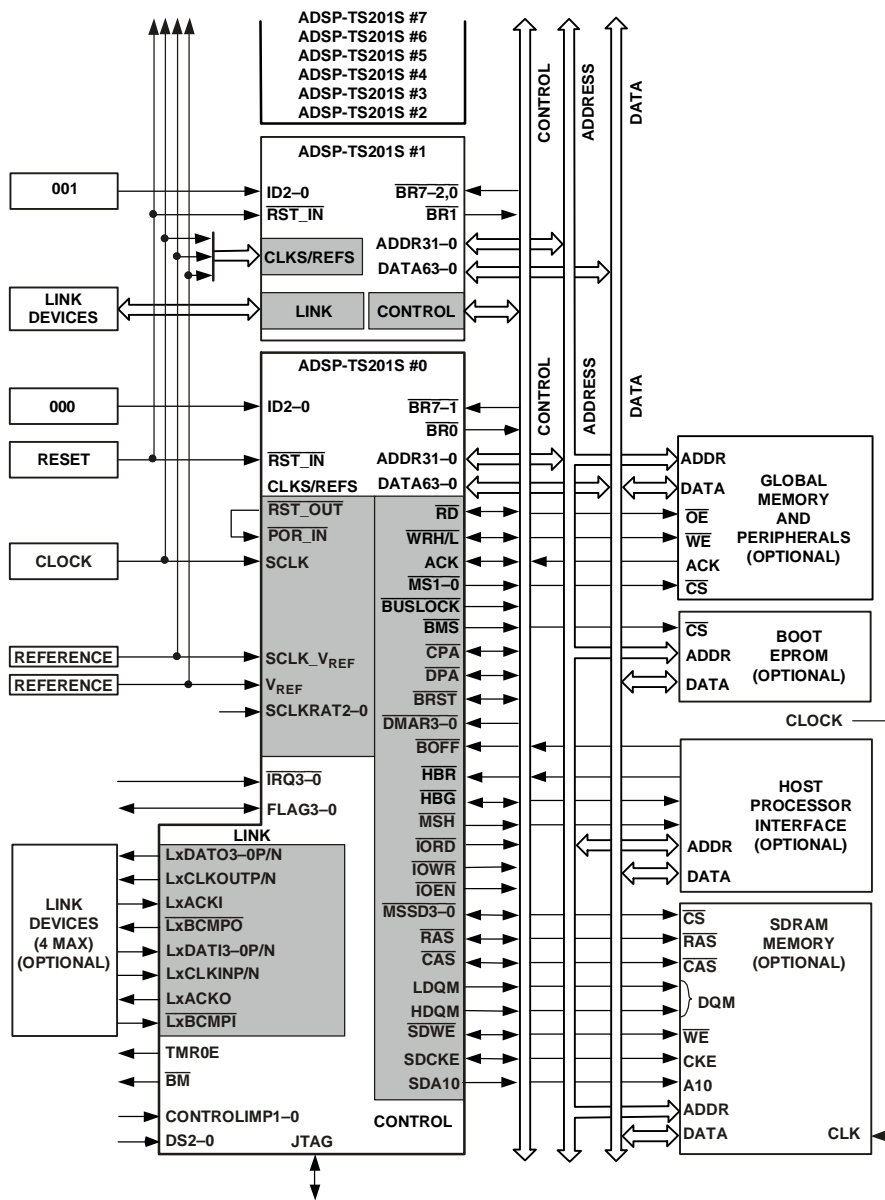


Figure 3. ADSP-TS201S Shared Memory Multiprocessing System

The SDRAM interface provides a glueless interface with standard SDRAMs—16M bit, 64M bit, 128M bit, and 256M bit. The DSP supports directly a maximum of four banks of 64M words × 32 bit of SDRAM. The SDRAM interface is mapped in external memory in each DSP’s unified memory map.

**EPROM Interface**

The ADSP-TS201S processor can be configured to boot from an external 8-bit EPROM at reset through the external port. An automatic process (which follows reset) loads a program from the EPROM into internal memory. This process uses sixteen wait cycles for each read access. During booting, the BMS pin functions as the EPROM chip select signal. The EPROM boot

procedure uses DMA channel 0, which packs the bytes into 32-bit instructions. Applications can also access the EPROM (write flash memories) during normal operation through DMA.

The EPROM or Flash Memory interface is not mapped in the DSP’s unified memory map. It is a byte address space limited to a maximum of 16M bytes (twenty-four address bits). The EPROM or Flash Memory interface can be used after boot via a DMA.

**DMA Controller**

The ADSP-TS201S processor’s on-chip DMA controller, with 14 DMA channels, provides zero-overhead data transfers without processor intervention. The DMA controller operates indepen-

dently and invisibly to the DSP's core, enabling DMA operations to occur while the DSP's core continues to execute program instructions.

The DMA controller performs DMA transfers between internal memory and external memory and memory-mapped peripherals, the internal memory of other DSPs on a common bus, a host processor, or link port I/O; between external memory and external peripherals or link port I/O; and between an external bus master and internal memory or link port I/O. The DMA controller performs the following DMA operations:

- External port block transfers. Four dedicated bidirectional DMA channels transfer blocks of data between the DSP's internal memory and any external memory or memory-mapped peripheral on the external bus. These transfers support master mode and handshake mode protocols.
- Link port transfers. Eight dedicated DMA channels (four transmit and four receive) transfer quad-word data only between link ports and between a link port and internal or external memory. These transfers only use handshake mode protocol. DMA priority rotates between the four receive channels.
- AutoDMA transfers. Two dedicated unidirectional DMA channels transfer data received from an external bus master to internal memory or to link port I/O. These transfers only use slave mode protocol, and an external bus master must initiate the transfer.

The DMA controller provides these additional features:

- Flyby transfers. Flyby operations only occur through the external port (DMA channel 0) and do not involve the DSP's core. The DMA controller acts as a conduit to transfer data from an I/O device to external SDRAM memory. During a transaction, the DSP relinquishes the external data bus; outputs addresses, memory selects ( $\overline{\text{MSSD3-0}}$ ) and the  $\overline{\text{IORD}}$ ,  $\overline{\text{IOWR}}$ ,  $\overline{\text{IOEN}}$ , and  $\overline{\text{RD}}/\overline{\text{WR}}$  strobes; and responds to ACK.
- DMA chaining. DMA chaining operations enable applications to automatically link one DMA transfer sequence to another for continuous transmission. The sequences can occur over different DMA channels and have different transmission attributes.
- Two-dimensional transfers. The DMA controller can access and transfer two-dimensional memory arrays on any DMA transmit or receive channel. These transfers are implemented with index, count, and modify registers for both the X and Y dimensions.

#### Link Ports (LVDS)

The DSP's four full-duplex link ports each provide additional four-bit receive and four-bit transmit I/O capability, using Low-Voltage, Differential-Signal (LVDS) technology. With the ability to operate at a double data rate—latching data on both the rising

and falling edges of the clock—running at 500 MHz, each link port can support up to 500M bytes per second per direction, for a combined maximum throughput of 4G bytes per second.

The link ports provide an optional communications channel that is useful in multiprocessor systems for implementing point-to-point interprocessor communications. Applications can also use the link ports for booting.

Each link port has its own triple-buffered quad-word input and double-buffered quad-word output registers. The DSP's core can write directly to a link port's transmit register and read from a receive register, or the DMA controller can perform DMA transfers through eight (four transmit and four receive) dedicated link port DMA channels.

Each link port direction has three signals that control its operation. For the transmitter,  $\text{LxCLKOUT}$  is the output transmit clock,  $\text{LxACKI}$  is the handshake input to control the data flow, and the  $\overline{\text{LxBCMPO}}$  output indicates that the block transfer is complete. For the receiver,  $\text{LxCLKIN}$  is the input receive clock,  $\text{LxACKO}$  is the handshake output to control the data flow, and the  $\overline{\text{LxBCMPI}}$  input indicates that the block transfer is complete. The  $\text{LxDATO3-0}$  pins are the data output bus for the transmitter and the  $\text{LxDATI3-0}$  pins are the input data bus for the receiver.

Applications can program separate error detection mechanisms for transmit and receive operations (applications can use the checksum mechanism to implement consecutive link port transfers), the size of data packets, and the speed at which bytes are transmitted.

#### Timer and General-Purpose I/O

The ADSP-TS201S processor has a timer pin ( $\text{TMR0E}$ ) that generates output when a programmed timer counter has expired and four programmable general-purpose I/O pins ( $\text{FLAG3-0}$ ) that can function as either single-bit input or output. As outputs, these pins can signal peripheral devices; as inputs, they can provide the test for conditional branching.

#### Reset and Booting

The ADSP-TS201S processor has three levels of reset:

- Power-up reset—After power-up of the system ( $\text{SCLK}$ , all static inputs, and strap pins are stable), the  $\overline{\text{RST\_IN}}$  pin must be asserted (low).
- Normal reset—For any chip reset following the power-up reset, the  $\overline{\text{RST\_IN}}$  pin must be asserted (low).
- DSP-core reset—When setting the  $\text{SWRST}$  bit in  $\text{EMUCTL}$ , the DSP core is reset, but not the external port or I/O.

For normal operations, tie the  $\overline{\text{RST\_OUT}}$  pin to the  $\overline{\text{POR\_IN}}$  pin.

After reset, the ADSP-TS201S processor has four boot options for beginning operation:

- Boot from EPROM.
- Boot by an external master (host or another ADSP-TS201S processor).



- Boot by link port.
- No boot—Start running from memory address selected with one of the  $\overline{\text{IRQ}}3\text{--}0$  interrupt signals. See [Table 2](#).

Using the ‘no boot’ option, the ADSP-TS201S processor must start running from memory when one of the interrupts is asserted.

**Table 2. No Boot, Run From Memory Addresses**

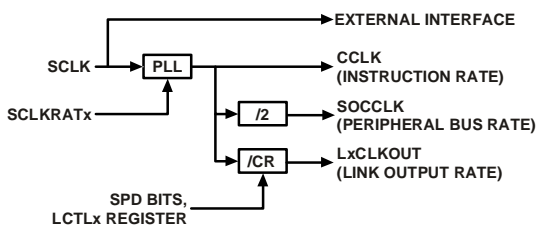
Interrupt	Address
$\overline{\text{IRQ}}0$	0x3000 0000 (External Memory)
$\overline{\text{IRQ}}1$	0x3800 0000 (External Memory)
$\overline{\text{IRQ}}2$	0x8000 0000 (External Memory)
$\overline{\text{IRQ}}3$	0x0000 0000 (Internal Memory)

The ADSP-TS201S processor core always exits from reset in the idle state and waits for an interrupt. Some of the interrupts in the interrupt vector table are initialized and enabled after reset.

For more information on boot options, see the *EE-174: ADSP-TS101S Booting Methods* on the Analog Devices website ([www.analog.com](http://www.analog.com))

### Clock Domains

The DSP uses calculated ratios of the SCLK clock to operate as shown in [Figure 4](#). The instruction execution rate is equal to CCLK. A PLL from SCLK generates CCLK which is phase-locked. The SCLKRATx pins define the clock multiplication of SCLK to CCLK (see [Table 4 on page 11](#)). The link port clock is generated from CCLK via a software programmable divisor, and the SOC bus operates at 1/2 CCLK. Memory transfers to external and link port buffers operate at the SOCCLK rate. SCLK also provides clock input for the external bus interface and defines the AC specification reference for the external bus signals. The external bus interface runs at the SCLK frequency. The maximum SCLK frequency is one quarter the internal DSP clock (CCLK) frequency.



**Figure 4. Clock Domains**

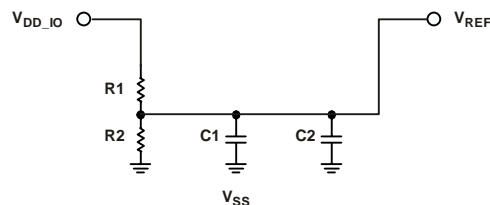
### Power Domains

The ADSP-TS201S processor has separate power supply connections for internal logic ( $V_{DD}$ ), analog circuits ( $V_{DD,A}$ ), I/O buffer ( $V_{DD,IO}$ ), and internal DRAM ( $V_{DD,DRAM}$ ) power supply.

Note that the analog ( $V_{DD,A}$ ) supply powers the clock generator PLLs. To produce a stable clock, systems must provide a clean power supply to power input  $V_{DD,A}$ . Designs must pay critical attention to bypassing the  $V_{DD,A}$  supply.

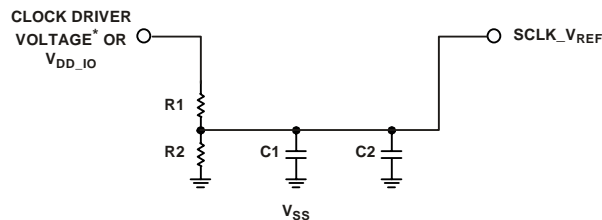
### Filtering Reference Voltage and Clocks

[Figure 5](#) and [Figure 6](#) show possible circuits for filtering  $V_{REF}$  and  $\text{SCLK}_{VREF}$ . These circuits provide the reference voltages for the switching voltage reference and system clock reference.



- R1: 2 k $\Omega$  SERIES RESISTOR ( $\pm 1\%$ )
- R2: 2 k $\Omega$  SERIES RESISTOR ( $\pm 1\%$ )
- C1: 1  $\mu\text{F}$  CAPACITOR (SMD)
- C2: 1 nF CAPACITOR (HF SMD) PLACED CLOSE TO DSP'S PINS

**Figure 5.  $V_{REF}$  filtering scheme**



- R1: 2 k $\Omega$  SERIES RESISTOR ( $\pm 1\%$ )
- R2: 2 k $\Omega$  SERIES RESISTOR ( $\pm 1\%$ )
- C1: 1  $\mu\text{F}$  CAPACITOR (SMD)
- C2: 1 nF CAPACITOR (HF SMD) PLACED CLOSE TO DSP'S PINS

\* IF CLOCK DRIVER VOLTAGE >  $V_{DD,IO}$

**Figure 6.  $\text{SCLK}_{VREF}$  filtering scheme**

### Development Tools

The ADSP-TS201S processor is supported with a complete set of CROSSCORE<sup>2</sup> software and hardware development tools, including Analog Devices emulators and VisualDSP++<sup>3</sup> development environment. The same emulator hardware that supports other TigerSHARC processors also fully emulates the ADSP-TS201S processor.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy to use assembler (which is based on an algebraic syntax), an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ runtime library that includes DSP and mathematical functions. A key point for these tools is C/C++ code

<sup>2</sup>CROSSCORE is a trademark of Analog Devices, Inc.

<sup>3</sup>VisualDSP++ is a trademark of Analog Devices, Inc.

efficiency. The compiler has been developed for efficient translation of C/C++ code to DSP assembly. The DSP has architectural features that improve the efficiency of compiled C/C++ code.

The VisualDSP++ debugger has a number of important features. Data visualization is enhanced by a plotting package that offers a significant level of flexibility. This graphical representation of user data enables the programmer to quickly determine the performance of an algorithm. As algorithms grow in complexity, this capability can have increasing significance on the designer's development schedule, increasing productivity. Statistical profiling enables the programmer to nonintrusively poll the processor as it is running the program. This feature, unique to VisualDSP++, enables the software developer to passively gather important code execution metrics without interrupting the realtime characteristics of the program. Essentially, the developer can identify bottlenecks in software quickly and efficiently. By using the profiler, the programmer can focus on those areas in the program that impact performance and take corrective action.

Debugging both C/C++ and assembly programs with the VisualDSP++ debugger, programmers can:

- View mixed C/C++ and assembly code (interleaved source and object information)
- Insert breakpoints
- Set conditional breakpoints on registers, memory, and stacks
- Trace instruction execution
- Perform linear or statistical profiling of program execution
- Fill, dump, and graphically plot the contents of memory
- Perform source level debugging
- Create custom debugger windows

The VisualDSP++ IDE lets programmers define and manage DSP software development. Its dialog boxes and property pages let programmers configure and manage all of the TigerSHARC processor development tools, including the color syntax highlighting in the VisualDSP++ editor. This capability permit programmers to:

- Control how the development tools process inputs and generate outputs
- Maintain a one-to-one correspondence with the tool's command line switches

The VisualDSP++™ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of DSP programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning, when developing new application code. The VDK features include Threads, Critical and Unscheduled regions, Semaphores, Events, and Device flags. The VDK also supports Priority-based, Preemptive, Cooperative and Time -Sliced scheduling approaches. In

addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

Because the VDK is a library, a developer can decide whether to use it or not. The VDK is integrated into the VisualDSP++™ development environment, but can also be used via standard command line tools. When the VDK is used, the development environment assists the developer with many error-prone tasks and assists in managing system resources, automating the generation of various VDK based objects, and visualizing the system state, when debugging an application that uses the VDK.

VCSE is Analog Devices' technology for creating, using, and reusing software components (independent modules of substantial functionality) to quickly and reliably assemble software applications. Download components from the Web and drop them into the application. Publish component archives from within VisualDSP++™. VCSE supports component implementation in C/C++ or assembly language.

Use the Expert Linker to visually manipulate the placement of code and data on the embedded system. View memory utilization in a color-coded graphical form, easily move code and data to different areas of the DSP or external memory with the drag of the mouse, examine run-time stack and heap usage. The Expert Linker is fully compatible with existing Linker Definition File (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices DSP emulators use the IEEE 1149.1 JTAG Test Access Port of the ADSP-TS201S processor to monitor and control the target board processor during emulation. The emulator provides full speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect target system loading or timing.

In addition to the software and hardware development tools available from Analog Devices, third parties provide a wide range of tools supporting the TigerSHARC processor family. Hardware tools include TigerSHARC processor PC plug-in cards. Third party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

#### **Designing an Emulator-Compatible DSP Board (Target)**

The Analog Devices family of emulators are tools that every DSP developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on each JTAG DSP. The emulator uses the TAP to access the internal features of the DSP, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The DSP must be halted to send data and commands, but once an operation has been completed by the emulator, the DSP system is set running at full speed with no impact on system timing.

To use these emulators, the target board must include a header that connects the DSP's JTAG port to the emulator.

For details on target board design issues including mechanical layout, single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see the *EE-68: Analog Devices JTAG Emulation Technical Reference* on the Analog Devices website ([www.analog.com](http://www.analog.com))—use site search on “EE-68”. This document is updated regularly to keep pace with improvements to emulator support.

### Additional Information

This data sheet provides a general overview of the ADSP-TS201S processor's architecture and functionality. For detailed information on the ADSP-TS201S processor's core architecture and instruction set, see the *ADSP-TS201 TigerSHARC Processor Hardware Reference* and the *ADSP-TS201 TigerSHARC Processor*

*Programming Reference*. For detailed information on the development tools for this processor, see the *VisualDSP++ User's Guide for TigerSHARC Processors*.

### PIN FUNCTION DESCRIPTIONS

While most of the ADSP-TS201S processor's input pins are normally synchronous—tied to a specific clock—a few are asynchronous. For these asynchronous signals, an on-chip synchronization circuit prevents metastability problems. Use the AC specification for asynchronous signals when the system design requires predictable, cycle-by-cycle behavior for these signals.

The output pins can be three-stated during normal operation. The DSP three-states all outputs during reset, allowing these pins to get to their internal pullup or pulldown state. Some pins have an internal pullup or pulldown resistor ( $\pm 30\%$  tolerance) that maintains a known value during transitions between different drivers.

**Table 3. Pin Definitions—Clocks and Reset**

Signal	Type	Description
SCLKRAT2–0	I (pd)	Core Clock Ratio. The DSP's core clock (CCLK) rate = $n \times \text{SCLK}$ , where $n$ is user-programmable using the SCLKRAT $x$ pins to the values shown in <a href="#">Table 4</a> . These pins must have a constant value while the DSP is powered. The core clock rate (CCLK) is the instruction cycle rate.
SCLK	I <sup>1</sup>	System Clock Input. The DSP's system input clock for cluster bus. The core clock rate is user-programmable using the SCLKRAT $x$ pins. <a href="#">For more information, see Clock Domains on page 9.</a>
$\overline{\text{RST\_IN}}$	I/A	Reset. Sets the DSP to a known state and causes program to be in idle state. $\overline{\text{RST\_IN}}$ must be asserted a specified time according to the type of reset operation. For details, see <a href="#">Reset and Booting on page 8</a> , <a href="#">Table 19 on page 22</a> , and <a href="#">Figure 9 on page 23</a> .
$\overline{\text{RST\_OUT}}$	O	Reset Output. Indicates that the DSP reset is complete. Connect to $\overline{\text{POR\_IN}}$ .
$\overline{\text{POR\_IN}}$	I/A	Power On Reset for internal DRAM. Connect to $\overline{\text{RST\_OUT}}$ .

I = input; A = asynchronous; O = output; OD = open drain output; T = Three-State; P = power supply; G = ground;  
 pd = internal pulldown 5 k $\Omega$ ; pu = internal pullup 5 k $\Omega$ ; pd<sub>0</sub> = internal pulldown 5 k $\Omega$  on DSP ID=0; pu<sub>0</sub> = internal pullup 5 k $\Omega$  on DSP ID=0;  
 pu<sub>od\_0</sub> = internal pullup 500 $\Omega$  on DSP ID=0; pd<sub>m</sub> = internal pulldown 5 k $\Omega$  on DSP bus master; pu<sub>m</sub> = internal pullup 5 k $\Omega$  on DSP bus master;  
 pu<sub>ad</sub> = internal pullup 40 k $\Omega$ ; For more pulldown and pullup information, see [Electrical characteristics on page 20](#).

<sup>1</sup>For more information on SCLK and SCLK<sub>VREF</sub> on revision 0.0 silicon, see the *EE-179: ADSP-TS201S System Design Guidelines* on the Analog Devices website ([www.analog.com](http://www.analog.com)).

**Table 4. SCLK Ratio**

SCLKRAT2–0	Ratio
000 (default)	4
001	5
010	6
011	7
100	8
101	10
110	12
111	Reserved