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

TigerSHARC Embedded Processor

ADSP-TS101S

FEATURES

- 300 MHz, 3.3 ns instruction cycle rate
- 6M bits of internal—on-chip—SRAM memory
- 19 mm × 19 mm (484-ball) or 27 mm × 27 mm (625-ball) PBGA package
- Dual computation blocks—each containing an ALU, a multiplier, a shifter, and a register file
- Dual integer ALUs, providing data addressing and pointer manipulation
- Integrated I/O includes 14-channel DMA controller, external port, 4 link ports, SDRAM controller, programmable flag pins, 2 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 with up to 8 TigerSHARC processors on a bus

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 and Table 2)
- Supports low overhead DMA transfers between internal memory, external memory, memory-mapped peripherals, link ports, other DSPs (multiprocessor), and host processors
- Eases DSP programming through extremely flexible instruction set and high-level language-friendly DSP architecture
- Enables scalable multiprocessing systems with low communications overhead

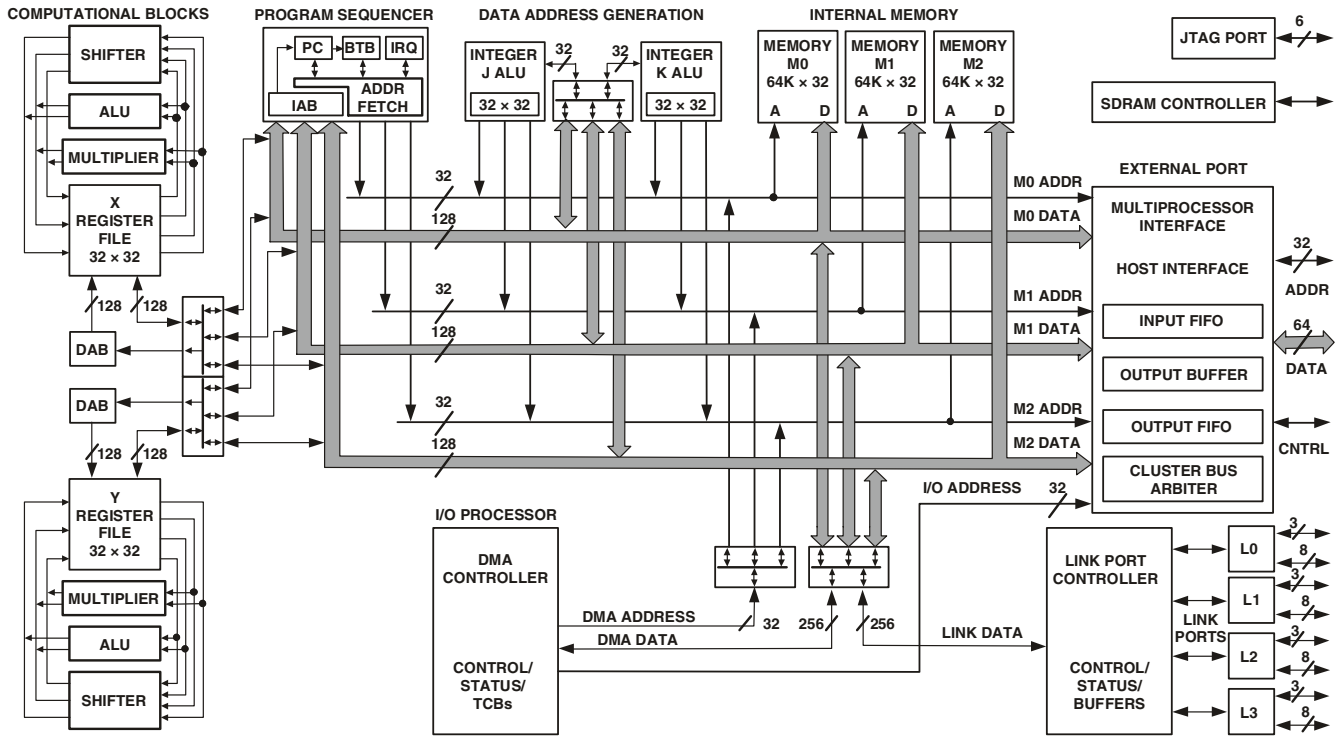

Figure 1. Functional Block Diagram

Rev. C

One Technology Way, P.O.Box 9106, Norwood, MA 02062-9106 U.S.A.
Tel: 781/329-4700 **www.analog.com**
Fax: 781/326-8703

ADSP-TS101S* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

COMPARABLE PARTS

View a parametric search of comparable parts.

EVALUATION KITS

- EZ-KIT Lite Evaluation Kit for ADSP-TS201S Processor
- USB-Based Emulator and High Performance USB-Based Emulator

DOCUMENTATION

Application Notes

- AN-911: A Detailed Guide to Powering the TigerSHARC Processors
- EE-104: Setting Up Streams with the VisualDSP Debugger
- EE-110: A Quick Primer on ELF and DWARF File Formats
- EE-120: Interfacing Assembly Language Programs to C
- EE-126: The ABCs of SDRAMemories
- EE-128: DSP in C++: Calling Assembly Class Member Functions From C++
- EE-143: Understanding DMA on the ADSP-TS101
- EE-147: Tuning C Source Code for the TigerSHARC® DSP Compiler
- EE-157: Explaining the Branch Target Buffer on the ADSP-TS101
- EE-159: Initializing DSP System & Control Registers From C and C++
- EE-167: Introduction to TigerSHARC® Multiprocessor Systems Using VisualDSP++™
- EE-169: Estimating Power For The ADSP-TS101S
- EE-174: ADSP-TS101S TigerSHARC® Processor Boot Loader Kernels Operation
- EE-175: Emulator and Evaluation Hardware Troubleshooting Guide for VisualDSP++ Users
- EE-176: Hardware Design Checklist For ADSP-TS101S TigerSHARC® Processors
- EE-178: The ADSP-TS101S TigerSHARC® On-chip SDRAM Controller
- EE-202: Using the Expert Linker for Multiprocessor LDFs
- EE-217: Updating the ADSP-TS101S TigerSHARC® EZ-KIT Lite™ Firmware
- EE-261: Understanding Jitter Requirements of PLL-Based Processors
- EE-263: Parallel Implementation of Fixed-Point FFTs on TigerSHARC® Processors
- EE-330: Windows Vista Compatibility in VisualDSP++ 5.0 Development Tools
- EE-332: Cycle Counting and Profiling
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users
- EE-68: Analog Devices JTAG Emulation Technical Reference

Data Sheet

- ADSP-TS101S: TigerSHARC Embedded Processor, 300 MHz, 6 Mbits, Data Sheet

Evaluation Kit Manuals

- ADSP-TS101S EZ-KIT Lite® Manual

Integrated Circuit Anomalies

- ADSP-TS101S TigerSHARC Anomaly List for Revision(s) 0.2, 0.4

Processor Manuals

- ADSP-TS101 TigerSHARC Processor Hardware Reference
- ADSP-TS101 TigerSHARC Processor Programming Reference
- TigerSHARC Processors: Manuals

Product Highlight

- General-Purpose TigerSHARC Processor Product Brief

Software Manuals

- VisualDSP++® 5.0 Assembler and Preprocessor Manual
- VisualDSP++® 5.0 C/C++ Compiler and Library Manual for TigerSHARC Processors
- VisualDSP++® 5.0 Kernel (VDK) Users Guide
- VisualDSP++® 5.0 Licensing Guide
- VisualDSP++® 5.0 Linker and Utilities Manual
- VisualDSP++® 5.0 Loader and Utilities Manual
- VisualDSP++® 5.0 Product Release Bulletin
- VisualDSP++® 5.0 Quick Installation Reference Card
- VisualDSP++® 5.0 Users Guide

SOFTWARE AND SYSTEMS REQUIREMENTS

- TigerSHARC Evaluation Kits

TOOLS AND SIMULATIONS

- ADSP-TS101 TigerSHARC BSDL File 19x19mm PBGA Package for Revision 0.4, (11/2006)
- ADSP-TS101 TigerSHARC BSDL File 27x27mm PBGA Package for Revision 0.4, (11/2006)
- ADSP-TS101: 19x19mm PBGA Silicon Revision 0.0 and 0.1 [BSDL Original File], 02/08/2001
- ADSP-TS101: 19x19mm PBGA Package Silicon Revision 0.2, [BSDL Original File], 09/09/2003
- ADSP-TS101: 27x27 PBGA Package Siicon Revision 0.2, [BSDL Original File], 09/09/2003
- ADSP-TS101: 27x27 PBGA Package Silicon Revision 0.0 and 0.1 [BSDL Original File], 10/12/2001
- Designing with BGA
- TigerSHARC Processors: Software and Tools
- ADSP-TS101S IBIS Datafile BGA Package

REFERENCE MATERIALS

Product Selection Guide

- ADI Complementary Parts Guide - Supervisory Devices and DSP Processors

Technical Articles

- A Software Solution for Chip Rate Processing in CDMA Wireless Infrastructure
- ADSP-TS101S MP System Simulation and Analysis
- Continuous Real-Time Signal Processing -- Comparing TigerSHARC and PowerPC Via Continuous cFFTs
- Rethinking Base Station, Baseband Processing for Wireless Communication
- SHARC Bites Back - The Memory Inside: TigerSHARC Swallows Its DRAM

White Papers

- ADSP-TS101S MP System Simulation and Analysis

DESIGN RESOURCES

- ADSP-TS101S Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

DISCUSSIONS

View all ADSP-TS101S EngineerZone Discussions.

SAMPLE AND BUY

Visit the product page to see pricing options.

TECHNICAL SUPPORT

Submit a technical question or find your regional support number.

DOCUMENT FEEDBACK

Submit feedback for this data sheet.

TABLE OF CONTENTS

REVISION HISTORY

5/09—Rev. B to Rev. C

Added parameter value (I_{DD_A} max) in Operating Conditions 20

Updated footnotes in 484-Ball PBGA (B-484) 43

Updated footnotes in 625-Ball PBGA (B-625) 44

Added surface-mount design info in Surface-Mount Design 44

Updated models in Ordering Guide 45

GENERAL DESCRIPTION

The ADSP-TS101S TigerSHARC® processor is an ultrahigh 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 8-, 16-, 32-, and 64-bit fixed-point processing—to set a new standard of performance for digital signal processors. The TigerSHARC processor's Static Superscalar architecture lets the processor execute up to four instructions each cycle, performing 24 fixed-point (16-bit) operations or six floating-point operations.

Three independent 128-bit-wide internal data buses, each connecting to one of the three 2M bit memory banks, enable quad word data, instruction, and I/O accesses and provide 14.4G bytes per second of internal memory bandwidth. Operating at 300 MHz, the ADSP-TS101S processor's core has a 3.3 ns instruction cycle time. Using its single-instruction, multiple-data (SIMD) features, the ADSP-TS101S can perform 2.4 billion 40-bit MACs or 600 million 80-bit MACs per second. Table 1 and Table 2 show the DSP's performance benchmarks.

Table 1. General-Purpose Algorithm Benchmarks at 300 MHz

Benchmark	Speed	Clock Cycles
32-bit algorithm, 600 million MACs/s peak performance		
1024 point complex FFT (Radix 2)	32.78 µs	9,835
50-tap FIR on 1024 input	91.67 µs	27,500
Single FIR MAC	1.83 ns	0.55
16-bit algorithm, 2.4 billion MACs/s peak performance		
256 point complex FFT (Radix 2)	3.67 µs	1,100
50-tap FIR on 1024 input	24.0 µs	7,200
Single FIR MAC	0.47 ns	0.14
Single complex FIR MAC	1.9 ns	0.57
I/O DMA transfer rate		
External port	800M bytes/s	n/a
Link ports (each)	250M bytes/s	n/a

Table 2. 3G Wireless Algorithm Benchmarks

Benchmark	Execution (MIPS)[1]
Turbo decode 384 kbps data channel	51 MIPS[2]
Viterbi decode 12.2 kbps AMR[3] voice channel	0.86 MIPS
Complex correlation 3.84 Mcps[4] with a spreading factor of 256	0.27 MIPS

[1] The execution speed is in instruction cycles per second.

[2] This value is for six iterations of the algorithm. For eight iterations of the turbo decoder, this benchmark is 67 MIPS.

[3] Adaptive multi rate (AMR)

[4] Megachips per second (Mcps)

The ADSP-TS101S is code compatible with the other TigerSHARC processors.

The Functional Block Diagram on Page 1 shows the processor's architectural blocks. These blocks include:

- Dual compute blocks, each consisting of an ALU, multiplier, 64-bit shifter, 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
- A program sequencer with instruction alignment buffer (IAB), branch target buffer (BTB), and interrupt controller
- Three 128-bit internal data buses, each connecting to one of three 2M bit memory banks
- On-chip SRAM (6M 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 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 2 shows a typical single-processor system with external SDRAM. Figure 4 on Page 8 shows a typical multiprocessor system.

The TigerSHARC processor uses a Static Superscalar architecture. This architecture is superscalar in that the ADSP-TS101S 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 reordering 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 an eight-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.

† Static Superscalar is a trademark of Analog Devices, Inc.

The ADSP-TS101S, in most cases, has a two-cycle arithmetic 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.

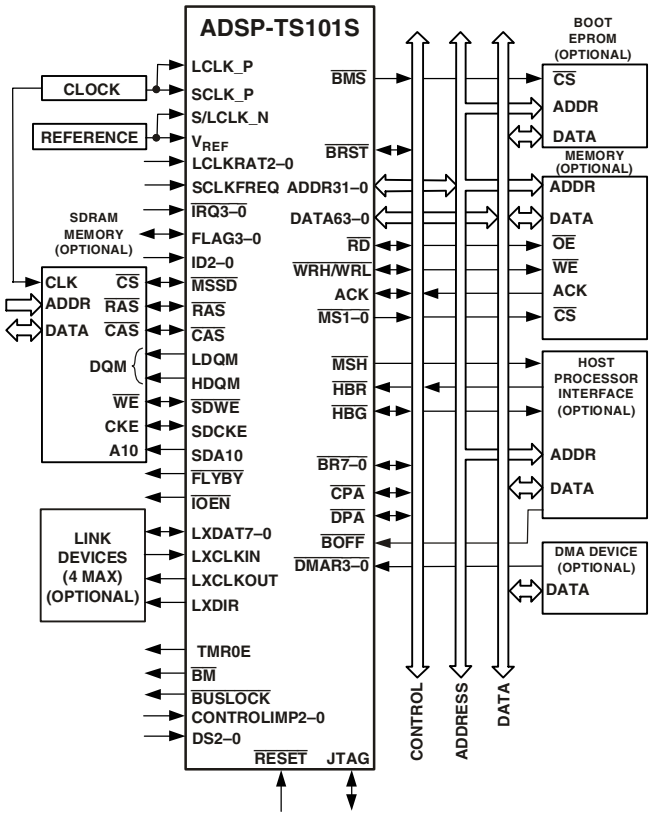

Figure 2. Single-Processor System with External SDRAM

In addition, the ADSP-TS101S supports SIMD operations two ways—SIMD compute blocks and SIMD computations. The programmer can direct both compute blocks to operate on the same data (broadcast distribution) or on different data (merged distribution). In addition, each compute block can execute four 16-bit or eight 8-bit SIMD computations in parallel.

DUAL COMPUTE BLOCKS

The ADSP-TS101S has compute blocks that can execute computations either independently or together as a SIMD engine. The DSP can issue up to two compute instructions per compute block each cycle, instructing the ALU, multiplier, or shifter to perform independent, simultaneous operations.

The compute blocks are referred to as X and Y in assembly syntax, and each block contains three computational units—an ALU, a multiplier, a 64-bit shifter, 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), or in sets of two (dual aligned) or 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.
- Shifter—the 64-bit shifter performs logical and arithmetic shifts, bit and bit stream manipulation, and field deposit and extraction operations.
- Accelerator—128-bit unit for trellis decoding (for example, Viterbi and turbo decoders) and complex correlations for communication applications.

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 24 fixed-point (16-bit) operations per cycle, providing 1,800 MFLOPS or 7.3 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-TS101S has two IALUs that provide powerful address generation capabilities and perform many general-purpose integer operations. Each of the IALUs:

- 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-TS101S processor's program sequencer supports:

- 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.
- An eight-cycle instruction pipeline—three-cycle fetch pipe and five-cycle execution pipe—with computation results available two cycles after operands are available.
- The 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.
- The management of program structures and determination of program flow 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-to-two overhead cycles, overcoming the three-to-six stage branch penalty.
- Compact code without the requirement to align code in memory; the IAB handles alignment.

Interrupt Controller

The DSP supports nested and non-nested 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 $\overline{\text{IRQ3–0}}$ 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:

- Enhanced 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, eliminating hardware modes
- Branch prediction encoded in instruction, enables zero-overhead loops
- Parallelism encoded in instruction line
- Conditional execution optional for all instructions
- User-defined, programmable partitioning between program and data memory

ON-CHIP SRAM MEMORY

The ADSP-TS101S has 6M bits of on-chip SRAM memory, divided into three blocks of 2M bits (64K words × 32 bits). Each block—M0, M1, and M2—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.

The DSP's internal and external memory (Figure 3) is organized into a unified memory map, which defines the location (address) of all elements in the system. 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.

Each internal memory block connects to one of the 128-bit-wide internal buses—block M0 to bus MD0, block M1 to bus MD1, and block M2 to bus MD2—enabling the DSP to perform three memory transfers in the same cycle. The DSP's internal bus architecture provides a total memory bandwidth of 14.4G bytes per second, enabling the core and I/O to access eight 32-bit data words (256 bits) and four 32-bit instructions each cycle. The DSP's flexible memory structure enables:

- DSP core and I/O access of different memory blocks in the same cycle
- DSP core access of all 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
- Complete context switch in less than 20 cycles (66 ns)

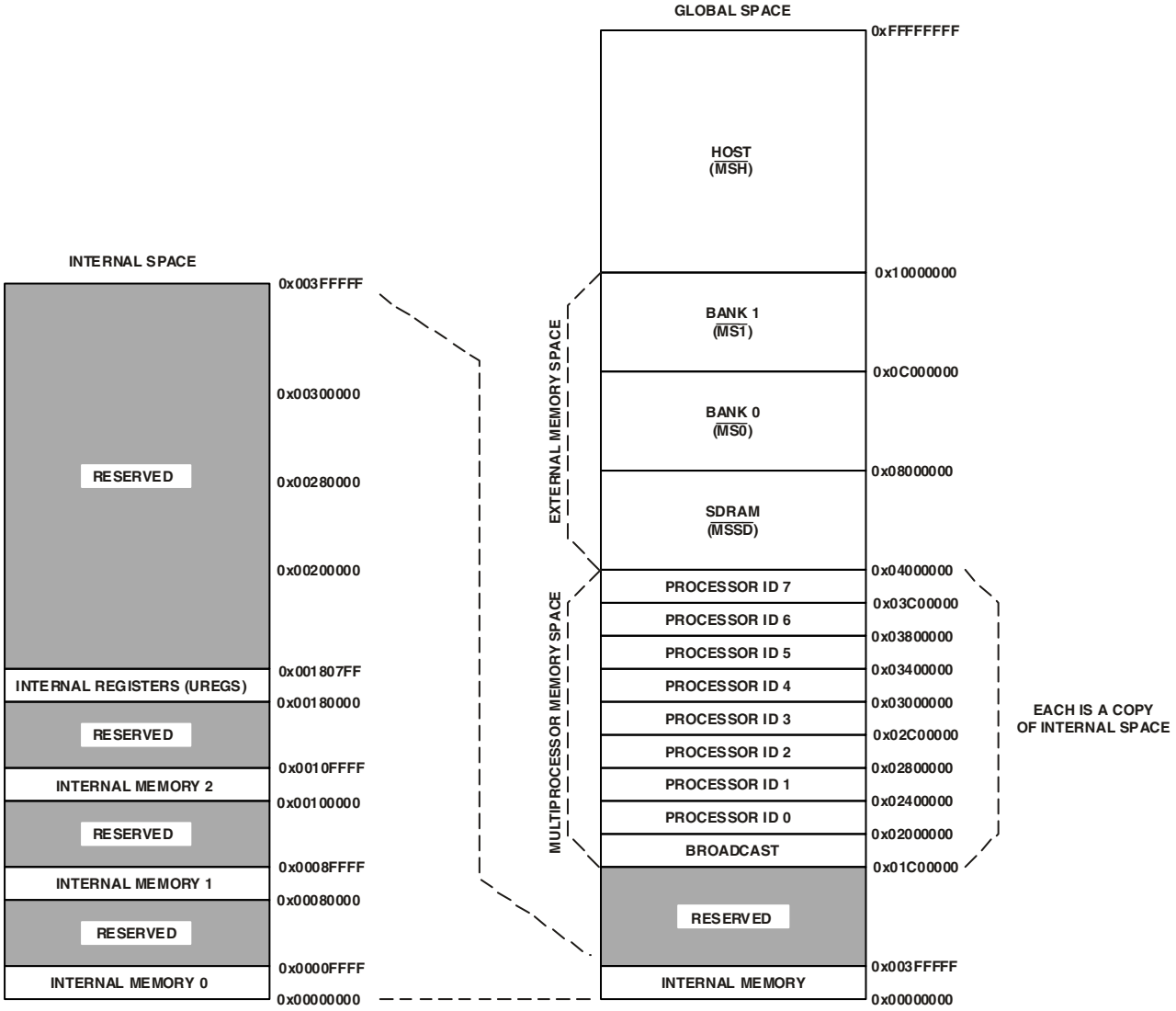

Figure 3. Memory Map

EXTERNAL PORT (OFF-CHIP MEMORY/PERIPHERALS INTERFACE)

The ADSP-TS101S processor's external port provides the processor'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—three 128-bit data buses and three 32-bit address buses—are multiplexed at the external port to create an external system bus with a single 64-bit data bus and a single 32-bit address bus. The external port supports data transfer rates of 800M bytes per second over external bus.

The external bus can be configured for 32- or 64-bit operation. When the system bus is configured for 64-bit operation, 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-TS101S 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-TS101S 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 accesses of the host 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-TS101S, 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-TS101S 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 3) that enables direct interprocessor accesses of each ADSP-TS101S 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-TS101S 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 1G bytes per second. The cluster bus provides 800M bytes per second throughput—with a total of 1.8G bytes per second interprocessor bandwidth.

SDRAM Controller

The SDRAM controller controls the ADSP-TS101S processor's transfers of data to and from synchronous DRAM (SDRAM). The throughput is 32 or 64 bits per SCLK cycle using the external port and SDRAM control pins.

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

EPROM Interface

The ADSP-TS101S can be configured to boot from 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 16 wait cycles for each read access. During booting, the $\overline{\text{BMS}}$ 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 (24 address bits). The EPROM or flash memory interface can be used after boot via a DMA.

DMA CONTROLLER

The ADSP-TS101S processor's on-chip DMA controller, with 14 DMA channels, provides zero-overhead data transfers without processor intervention. The DMA controller operates independently 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
- Internal memory of other DSPs on a common bus, a host processor, or link port I/O
- External memory and external peripherals or link port I/O
- External bus master and internal memory or link port I/O

The DMA controller provides a number of additional features.

The DMA controller supports 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 one external device to another through external memory. During a transaction, the DSP:

- Relinquishes the external data bus
- Outputs addresses, memory selects ($\overline{\text{MS1–0}}$, $\overline{\text{MSSD}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{SDWE}}$) and the $\overline{\text{FLYBY}}$, $\overline{\text{IOEN}}$, and $\overline{\text{RD}}$/$\overline{\text{WR}}$ strobes
- Responds to ACK

DMA chaining is also supported by the DMA controller. 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.

The DMA controller also supports 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.

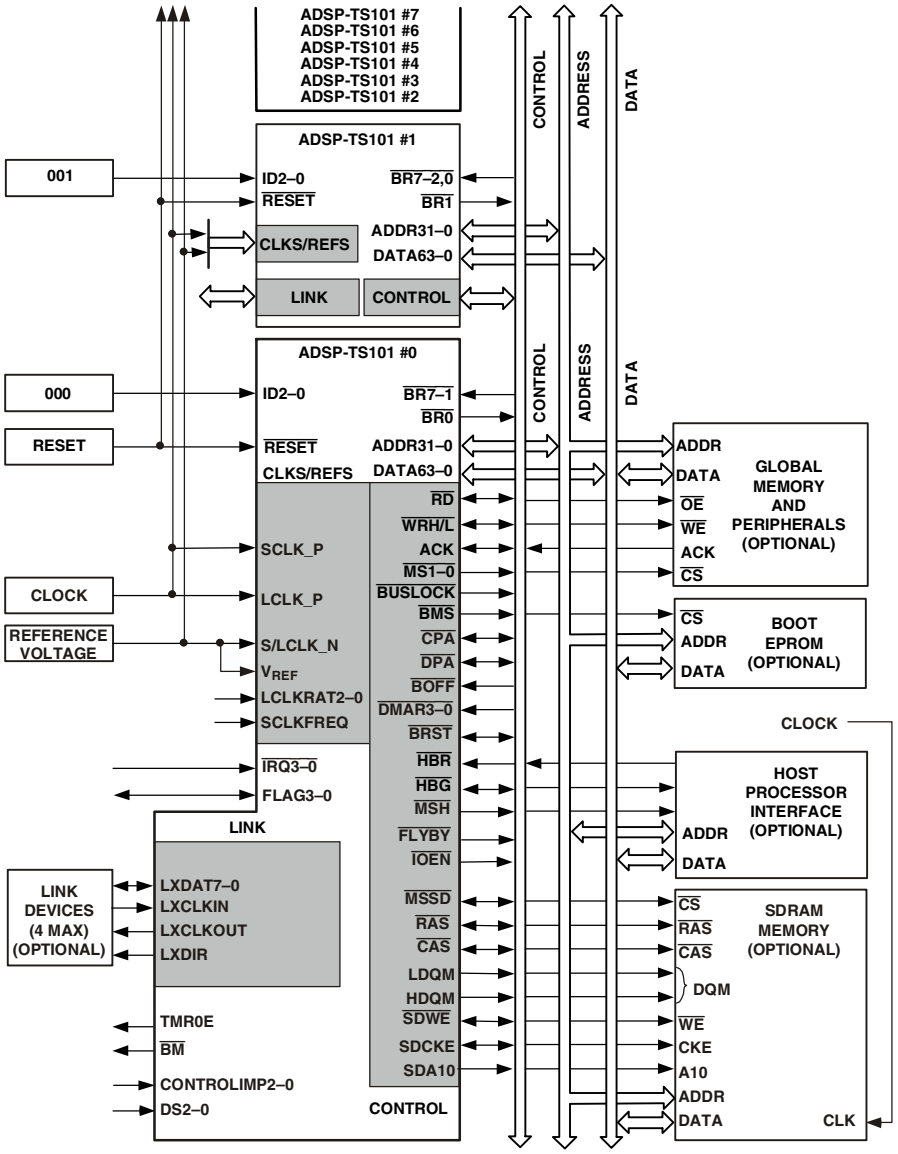

Figure 4. Shared Memory Multiprocessing System

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.

LINK PORTS

The DSP's four link ports provide additional 8-bit bidirectional I/O capability. With the ability to operate at a double data rate—latching data on both the rising and falling edges of the clock—running at 125 MHz, each link port can support up to 250M bytes per second, for a combined maximum throughput of 1G 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 double-buffered input and 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 has three signals that control its operation. LxCLKOUT and LxCLKIN implement clock/acknowledge handshaking. LxDIR indicates the direction of transfer and is used only when buffering the LxDAT signals. An example application would be using differential low-swing buffers for long twisted-pair wires. LxDAT provides the 8-bit data bus input/output.

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.

Under certain conditions, the link port receiver can initiate a token switch to reverse the direction of transfer; the transmitter becomes the receiver and vice versa.

TIMER AND GENERAL-PURPOSE I/O

The ADSP-TS101S has a timer pin (TMR0E) that generates output when a programmed timer counter has expired. Also, the DSP has four programmable general-purpose I/O pins (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-TS101S has two levels of reset (see reset specifications Page 24):

- Power-up reset—after power-up of the system, and strap options are stable, the $\overline{\text{RESET}}$ pin must be asserted (low).
- Normal reset—for any resets following the power-up reset sequence, the $\overline{\text{RESET}}$ pin must be asserted.

The DSP can be reset internally (core reset) by setting the SWRST bit in SQCTL. The core is reset, but not the external port or I/O.

After reset, the ADSP-TS101S has four boot options for beginning operation:

- Boot from EPROM. The DSP defaults to EPROM booting when the $\overline{\text{BMS}}$ pin strap option is set low. See Strap Pin Function Descriptions on Page 19.
- Boot by an external master (host or another ADSP-TS101S). Any master on the cluster bus can boot the ADSP-TS101S through writes to its internal memory or through autoDMA.
- Boot by link port. All four receive link DMA channels are initialized after reset to transfer a 256-word block to internal memory address 0 to 255, and to issue an interrupt at the end of the block (similar to EP DMA). The corresponding DMA interrupts are set to address zero (0).
- No boot—Start running from an external memory. Using the "no boot" option, the ADSP-TS101S must start running from an external memory, caused by asserting one of the $\overline{\text{IRQ3–0}}$ interrupt signals.

The ADSP-TS101S 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.

LOW POWER OPERATION

The ADSP-TS101S can enter a low power sleep mode in which its core does not execute instructions, reducing power consumption to a minimum. The ADSP-TS101S exits sleep mode when it senses a falling edge on any of its $\overline{\text{IRQ3–0}}$ interrupt inputs. The interrupt, if enabled, causes the ADSP-TS101S to execute the corresponding interrupt service routine. This feature is useful for systems that require a low power standby mode.

CLOCK DOMAINS

As shown in Figure 5, the ADSP-TS101S has two clock inputs, SCLK (system clock) and LCLK (local clock).

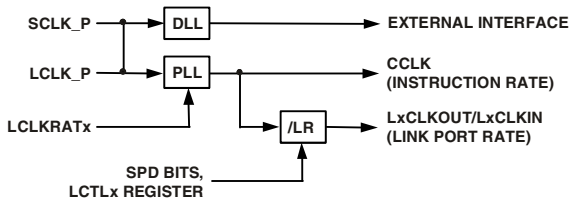

Figure 5. Clock Domains

These inputs drive its two major clock domains:

- SCLK (system clock). 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 1× the SCLK frequency. A DLL locks internal SCLK to SCLK input.
- LCLK (local clock). Provides clock input to the internal clock driver, CCLK, which is the internal clock for the core, internal buses, memory, and link ports. The instruction execution rate is equal to CCLK. A PLL from LCLK gener-

ates CCLK, which is phase-locked. The LCLKRAT pins define the clock multiplication of LCLK to CCLK (see Table 4). The link port clock is generated from CCLK via a software programmable divisor. $\overline{\text{RESET}}$ must be asserted until LCLK is stable and within specification for at least 2 ms. This applies to power-up as well as any dynamic modification of LCLK after power-up. Dynamic modification may include LCLK going out of specification as long as $\overline{\text{RESET}}$ is asserted.

Connecting SCLK and LCLK to the same clock source is a requirement for the device. Using an integer clock multiplication value provides predictable cycle-by-cycle operation, a requirement of fault-tolerant systems and some multiprocessing systems.

Noninteger values are completely functional and acceptable for applications that do not require predictable cycle-by-cycle operation.

OUTPUT PIN DRIVE STRENGTH CONTROL

Pins CONTROLIMP2–0 and DS2–0 work together to control the output drive strength of two groups of pins, the Address/Data/Control pin group and the Link pin group. CONTROLIMP2–0 independently configures the two pin groups to the maximum drive strength or to a digitally controlled drive strength that is selectable by the DS2–0 pins (see Table 13 on Page 18). If the digitally controlled drive strength is selected for a pin group, the DS2–0 pins determine one of eight strength levels for that group (see Table 14 on Page 18). The drive strength selected varies the slew rate of the driver. Drive strength 0 (DS2–0 = 000) is the weakest and slowest slew rate. Drive strength 7 (DS2–0 = 111) is the strongest and fastest slew rate.

The stronger drive strengths are useful for high frequency switching while the lower strengths may allow use of a relaxed design methodology. The strongest drive strengths have a larger di/dt and thus require more attention to signal integrity issues such a ringing, reflections and coupling. Also, a larger di/dt can increase external supply rail noise, which impacts power supply and power distribution design.

The drive strengths for the $\overline{\text{EMU}}$, $\overline{\text{CPA}}$, and $\overline{\text{DPA}}$ pins are not controllable and are fixed to the maximum level.

For drive strength calculation, see Output Drive Currents on Page 32.

POWER SUPPLIES

The ADSP-TS101S has separate power supply connections for internal logic (V_{DD}), analog circuits (V_{DD_A}), and I/O buffer (V_{DD_IO}) power supply. The internal (V_{DD}) and analog (V_{DD_A}) supplies must meet the 1.2 V requirement. The I/O buffer (V_{DD_IO}) supply must meet the 3.3 V requirement.

The analog supply (V_{DD_A}) 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.

The required power-on sequence for the DSP is to provide V_{DD} (and V_{DD_A}) before V_{DD_IO}.

FILTERING REFERENCE VOLTAGE AND CLOCKS

Figure 6 shows a possible circuit for filtering V_{REF}, SCLK_N, and LCLK_N. This circuit provides the reference voltage for the switching voltage, system clock, and local clock references.

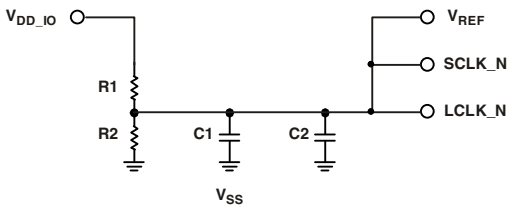

Figure 6. V_{REF}, SCLK_N, and LCLK_N Filter

DEVELOPMENT TOOLS

The ADSP-TS101S is supported with a complete set of CROSSCORE®† software and hardware development tools, including Analog Devices emulators and VisualDSP++®‡ development environment. The same emulator hardware that supports other TigerSHARC processors also fully emulates the ADSP-TS101S.

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++ run-time library that includes DSP and mathematical functions. A key point for these tools is C/C++ code 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 real-time 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.

† CROSSCORE is a registered trademark of Analog Devices, Inc.
‡ VisualDSP++ is a registered trademark of Analog Devices, Inc.

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++ integrated development and debugging environment (IDDE) lets programmers define and manage DSP software development. Its dialog boxes and property pages let programmers configure and manage all of the TigerSHARC development tools, including the color syntax highlighting in the VisualDSP++ editor. This capability permits 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.

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 a 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-TS101S 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 *EE-68: Analog Devices JTAG Emulation Technical Reference* on the Analog Devices website (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-TS101S processor's architecture and functionality. For detailed information on the ADSP-TS101S processor's core architecture and instruction set, see the *ADSP-TS101 TigerSHARC Processor Programming Reference* and the *ADSP-TS101 TigerSHARC Processor Hardware Reference*. For detailed information on the development tools for this processor, see the *VisualDSP++ User's Guide*.

PIN FUNCTION DESCRIPTIONS

While most of the ADSP-TS101S 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. The synchronous ac specification for asynchronous signals is used only when predictable cycle-by-cycle behavior is required.

All inputs are sampled by a clock reference, therefore input specifications (asynchronous minimum pulse widths or synchronous input setup and hold) must be met to guarantee recognition.

PIN STATES AT RESET

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 pull-up or pull-down state. Some output pins (control signals) have a pull-up or pull-down that maintains a known value during transitions between different drivers.

PIN DEFINITIONS

The Type column in the following pin definitions tables describes the pin type, when the pin is used in the system. The Term (for termination) column describes the pin termination type if the pin is not used by the system. Note that some pins are always used (indicated with au symbol).

Table 3. Pin Definitions—Clocks and Reset

Signal	Type	Term	Description
LCLK_N	I	au	Local Clock Reference. Connect this pin to V_{REF} as shown in Figure 6.
LCLK_P	I	au	Local Clock Input. DSP clock input. The instruction cycle rate = n × LCLK, where n is user-programmable to 2, 2.5, 3, 3.5, 4, 5, or 6. For more information, see Clock Domains on Page 9.
LCLKRAT2–0[1]	I (pd[2])	au	LCLK Ratio. The DSP's core clock (instruction cycle rate) = n × LCLK, where n is user-programmable to 2, 2.5, 3, 3.5, 4, 5, or 6 as shown in Table 4. These pins must have a constant value while the DSP is powered.
SCLK_N	I	au	System Clock Reference. Connect this pin to V_{REF} as shown in Figure 6.
SCLK_P	I	au	System Clock Input. The DSP's system input clock for cluster bus. This pin must be connected to the same clock source as LCLK_P. For more information, see Clock Domains on Page 9.
SCLKFREQ[3]	I (pu[2])	au	SCLK Frequency. SCLKFREQ = 1 is required. The SCLKFREQ pin must have a constant value while the DSP is powered.
$\overline{\text{RESET}}$	I/A	au	Reset. Sets the DSP to a known state and causes program to be in idle state. $\overline{\text{RESET}}$ must be asserted at specified time according to the type of reset operation. For details, see Reset and Booting on Page 9.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.
[3] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.

Table 4. LCLK Ratio

LCLKRAT2–0	Ratio
000 (default)	2
001	2.5
010	3
011	3.5
100	4
101	5
110	6
111	Reserved

Table 5. Pin Definitions—External Port Bus Controls

Signal	Type	Term	Description
ADDR31–0[1]	I/O/T	nc	Address Bus. The DSP issues addresses for accessing memory and peripherals on these pins. In a multiprocessor system, the bus master drives addresses for accessing internal memory or I/O processor registers of other ADSP-TS101S processors. The DSP inputs addresses when a host or another DSP accesses its internal memory or I/O processor registers.
DATA63–0[1]	I/O/T	nc	External Data Bus. Data and instructions are received, and driven by the DSP, on these pins.
$\overline{RD}$[2]	I/O/T (pu[3])	nc	Memory Read. $\overline{RD}$ is asserted whenever the DSP reads from any slave in the system, excluding SDRAM. When the DSP is a slave, $\overline{RD}$ is an input and indicates read transactions that access its internal memory or universal registers. In a multiprocessor system, the bus master drives $\overline{RD}$. The $\overline{RD}$ pin changes concurrently with ADDR pins.
$\overline{WRL}$[2]	I/O/T (pu[3])	nc	Write Low. $\overline{WRL}$ is asserted in two cases: When the ADSP-TS101S writes to an even address word of external memory or to another external bus agent; and when the ADSP-TS101S writes to a 32-bit zone (host, memory, or DSP programmed to 32-bit bus). An external master (host or DSP) asserts $\overline{WRL}$ for writing to a DSP's low word of internal memory. In a multiprocessor system, the bus master drives $\overline{WRL}$. The $\overline{WRL}$ pin changes concurrently with ADDR pins. When the DSP is a slave, $\overline{WRL}$ is an input and indicates write transactions that access its internal memory or universal registers.
$\overline{WRH}$[2]	I/O/T (pu[3])	nc	Write High. $\overline{WRH}$ is asserted when the ADSP-TS101S writes a long word (64 bits) or writes to an odd address word of external memory or to another external bus agent on a 64-bit data bus. An external master (host or another DSP) must assert $\overline{WRH}$ for writing to a DSP's high word of 64-bit data bus. In a multiprocessing system, the bus master drives $\overline{WRH}$. The $\overline{WRH}$ pin changes concurrently with ADDR pins. When the DSP is a slave, $\overline{WRH}$ is an input and indicates write transactions that access its internal memory or universal registers.
ACK	I/O/T	epu	Acknowledge. External slave devices can deassert ACK to add wait states to external memory accesses. ACK is used by I/O devices, memory controllers, and other peripherals on the data phase. The DSP can deassert ACK to add wait states to read accesses of its internal memory. The ADSP-TS101S does not drive ACK during slave writes. Therefore, an external (approximately 10 kΩ) pull-up is required.
$\overline{BMS}$[2, 4]	O/T (pu/pd[3])	au	Boot Memory Select. $\overline{BMS}$ is the chip select for boot EPROM or flash memory. During reset, the DSP uses $\overline{BMS}$ as a strap pin (EBOOT) for EPROM boot mode. When the DSP is configured to boot from EPROM, $\overline{BMS}$ is active during the boot sequence. Pull-down enabled during $\overline{RESET}$ (asserted); pull-up enabled after $\overline{RESET}$ (deasserted). In a multiprocessor system, the DSP bus master drives $\overline{BMS}$. For details see Reset and Booting on Page 9 and the EBOOT signal description in Table 16 on Page 19.
$\overline{MS1–0}$[2]	O/T (pu[3])	nc	Memory Select. $\overline{MS0}$ or $\overline{MS1}$ is asserted whenever the DSP accesses memory banks 0 or 1, respectively. $\overline{MS1–0}$ are decoded memory address pins that change concurrently with ADDR pins. When ADDR31:26 = 0b000010, $\overline{MS0}$ is asserted. When ADDR31:26 = 0b000011, $\overline{MS1}$ is asserted. In multiprocessor systems, the master DSP drives $\overline{MS1–0}$.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

Table 5. Pin Definitions—External Port Bus Controls (Continued)

Signal	Type	Term	Description
$\overline{MSH}$[2]	O/T (pu[3])	nc	Memory Select Host. $\overline{MSH}$ is asserted whenever the DSP accesses the host address space (ADDR31:28 ≠ 0b0000). $\overline{MSH}$ is a decoded memory address pin that changes concurrently with ADDR pins. In a multiprocessor system, the bus master DSP drives $\overline{MSH}$.
$\overline{BRST}$[2]	I/O/T (pu[3])	nc	Burst. The current bus master (DSP or host) asserts this pin to indicate that it is reading or writing data associated with consecutive addresses. A slave device can ignore addresses after the first one and increment an internal address counter after each transfer. For host-to-DSP burst accesses, the DSP increments the address automatically while $\overline{BRST}$ is asserted.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] The address and data buses may float for several cycles during bus mastership transitions between a TigerSHARC processor and a host. Floating in this case means that these inputs are not driven by any source and that dc-biased terminations are not present. It is not necessary to add pull-ups as there are no reliability issues and the worst-case power consumption for these floating inputs is negligible. Unconnected address pins may require pull-ups or pull-downs to avoid erroneous slave accesses, depending on the system. Unconnected data pins may be left floating.

[2] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.

[3] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.

[4] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.

Table 6. Pin Definitions—External Port Arbitration

Signal	Type	Term	Description
$\overline{BR7–0}$	I/O	epu	Multiprocessing Bus Request Pins. Used by the DSPs in a multiprocessor system to arbitrate for bus mastership. Each DSP drives its own $\overline{BRx}$ line (corresponding to the value of its ID2–0 inputs) and monitors all others. In systems with fewer than eight DSPs, set the unused $\overline{BRx}$ pins high.
ID2–0[1]	I (pd[2])	au	Multiprocessor ID. Indicates the DSP's ID. From the ID, the DSP determines its order in a multiprocessor system. These pins also indicate to the DSP which bus request ($\overline{BR0}$–$\overline{BR7}$) to assert when requesting the bus: 000 = $\overline{BR0}$, 001 = $\overline{BR1}$, 010 = $\overline{BR2}$, 011 = $\overline{BR3}$, 100 = $\overline{BR4}$, 101 = $\overline{BR5}$, 110 = $\overline{BR6}$, or 111 = $\overline{BR7}$. ID2–0 must have a constant value during system operation and can change during reset only.
$\overline{BM}$[1]	O (pd[2])	au	Bus Master. The current bus master DSP asserts $\overline{BM}$. For debugging only. At reset this is a strap pin. For more information, see Table 16 on Page 19.
$\overline{BOFF}$	I	epu	Back Off. A deadlock situation can occur when the host and a DSP try to read from each other's bus at the same time. When deadlock occurs, the host can assert $\overline{BOFF}$ to force the DSP to relinquish the bus before completing its outstanding transaction, but only if the outstanding transaction is to host memory space ($\overline{MSH}$).
$\overline{BUSLOCK}$[3]	O/T (pu[2])	nc	Bus Lock Indication. Provides an indication that the current bus master has locked the bus.
$\overline{HBR}$	I	epu	Host Bus Request. A host must assert $\overline{HBR}$ to request control of the DSP's external bus. When $\overline{HBR}$ is asserted in a multiprocessing system, the bus master relinquishes the bus and asserts $\overline{HBG}$ once the outstanding transaction is finished.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

Table 6. Pin Definitions—External Port Arbitration (Continued)

Signal	Type	Term	Description
$\overline{HBG}$[3]	I/O/T (pu[2])	nc	Host Bus Grant. Acknowledges $\overline{HBR}$ and indicates that the host can take control of the external bus. When relinquishing the bus, the master DSP three-states the ADDR31–0, DATA63–0, $\overline{MSH}$, $\overline{MSSD}$, $\overline{MS1–0}$, $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{BMS}$, $\overline{BRST}$, $\overline{FLYBY}$, $\overline{IOEN}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{SDWE}$, SDA10, SDCKE, LDQM and HDQM pins, and the DSP puts the SDRAM in self-refresh mode. The DSP asserts $\overline{HBG}$ until the host deasserts $\overline{HBR}$. In multiprocessor systems, the current bus master DSP drives $\overline{HBG}$, and all slave DSPs monitor $\overline{HBG}$.
$\overline{CPA}$	I/O (o/d)	See next column	Core Priority Access. Asserted while the DSP's core accesses external memory. This pin enables a slave DSP to interrupt a master DSP's background DMA transfers and gain control of the external bus for core-initiated transactions. $\overline{CPA}$ is an open drain output, connected to all DSPs in the system. The $\overline{CPA}$ pin has an internal 500 Ω pull-up resistor, which is only enabled on the DSP with ID2–0 = 0. If ID0 is not used, terminate this pin as either epu or nc. If ID7–1 is not used, terminate this pin as epu.
$\overline{DPA}$	I/O (o/d)	See next column	DMA Priority Access. Asserted while a high-priority DSP DMA channel accesses external memory. This pin enables a high-priority DMA channel on a slave DSP to interrupt transfers of a normal-priority DMA channel on a master DSP and gain control of the external bus for DMA-initiated transactions. $\overline{DPA}$ is an open drain output, connected to all DSPs in the system. The $\overline{DPA}$ pin has an internal 500 Ω pull-up resistor, which is only enabled on the DSP with ID2–0 = 0. If ID0 is not used, terminate this pin as either epu or nc. If ID7–1 is not used, terminate this pin as epu.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.
[3] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.

Table 7. Pin Definitions—External Port DMA/Flyby

Signal	Type	Term	Description
$\overline{DMAR3–0}$	I/A	epu	DMA Request Pins. Enable external I/O devices to request DMA services from the DSP. In response to $\overline{DMARx}$, the DSP performs DMA transfers according to the DMA channel's initialization. The DSP ignores DMA requests from uninitialized channels.
$\overline{FLYBY}$[1]	O/T (pu[2])	nc	Flyby Mode. When a DSP DMA channel is initiated in $\overline{FLYBY}$ mode, it generates flyby transactions on the external bus. During flyby transactions, the DSP asserts $\overline{FLYBY}$, which signals the source or destination I/O device to latch the next data or strobe the current data, respectively, and to prepare for the next data on the next cycle.
$\overline{IOEN}$[1]	O/T (pu[2])	nc	I/O Device Output Enable. Enables the output buffers of an external I/O device for flyby transactions between the device and external memory. Active on flyby transactions.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.

Table 8. Pin Definitions—External Port SDRAM Controller

Signal	Type	Term	Description
$\overline{\text{MSSD}}$[1]	I/O/T (pu[2])	nc	Memory Select SDRAM. $\overline{\text{MSSD}}$ is asserted whenever the DSP accesses SDRAM memory space. $\overline{\text{MSSD}}$ is a decoded memory address pin that is asserted whenever the DSP issues an SDRAM command cycle (access to ADDR31:26 = 0b000001). MSSD in a multiprocessor system is driven by the master DSP.
$\overline{\text{RAS}}$[1]	I/O/T (pu[2])	nc	Row Address Select. When sampled low, $\overline{\text{RAS}}$ indicates that a row address is valid in a read or write of SDRAM. In other SDRAM accesses, RAS defines the type of operation to execute according to SDRAM specification.
$\overline{\text{CAS}}$[1]	I/O/T (pu[2])	nc	Column Address Select. When sampled low, $\overline{\text{CAS}}$ indicates that a column address is valid in a read or write of SDRAM. In other SDRAM accesses, $\overline{\text{CAS}}$ defines the type of operation to execute according to the SDRAM specification.
LDQM[1]	O/T (pu[2])	nc	Low Word SDRAM Data Mask. When LDQM is sampled high, the DSP three-states the SDRAM DQ buffers. LDQM is valid on SDRAM transactions when $\overline{\text{CAS}}$ is asserted and is inactive on read transactions. On write transactions, LDQM is active when accessing an odd address word on a 64-bit memory bus to disable the write of the low word.
HDQM[1]	O/T (pu[2])	nc	High Word SDRAM Data Mask. When HDQM is sampled high, the DSP three-states the SDRAM DQ buffers. HDQM is valid on SDRAM transactions when $\overline{\text{CAS}}$ is asserted and is inactive on read transactions. On write transactions, HDQM is active when accessing an even address in word accesses or is active when memory is configured for a 32-bit bus to disable the write of the high word.
SDA10[1]	O/T (pu[2])	nc	SDRAM Address bit 10 pin. Separate A10 signals enable SDRAM refresh operation while the DSP executes non-SDRAM transactions.
SDCKE[1, 3]	I/O/T (pu/pd[2])	nc	SDRAM Clock Enable. Activates the SDRAM clock for SDRAM self-refresh or suspend modes. A slave DSP in a multiprocessor system does not have the pull-up or pull-down. A master DSP (or ID = 0 in a single processor system) has a 100 kΩ pull-up before granting the bus to the host, except when the SDRAM is put in self-refresh mode. In self-refresh mode, the master has a 100 kΩ pull-down before granting the bus to the host.
$\overline{\text{SDWE}}$[1]	I/O/T (pu[2])	nc	SDRAM Write Enable. When sampled low while $\overline{\text{CAS}}$ is active, $\overline{\text{SDWE}}$ indicates an SDRAM write access. When sampled high while $\overline{\text{CAS}}$ is active, $\overline{\text{SDWE}}$ indicates an SDRAM read access. In other SDRAM accesses, $\overline{\text{SDWE}}$ defines the type of operation to execute according to SDRAM specification.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to $V_{DD\text{-}IO}$, nc = not connected; au = always used.

[1] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.
[3] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.

Table 9. Pin Definitions—JTAG Port

Signal	Type	Term	Description
$\overline{\text{EMU}}$	O (o/d)	nc[1]	Emulation. Connected only to the DSP's JTAG emulator target board connector.
TCK	I	epd or epu[1]	Test Clock (JTAG). Provides an asynchronous clock for JTAG scan.
TDI[2]	I (pu[3])	nc[1]	Test Data Input (JTAG). A serial data input of the scan path.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to $V_{DD\text{-}IO}$, nc = not connected; au = always used.

Table 9. Pin Definitions—JTAG Port (Continued)

Signal	Type	Term	Description
TDO	O/T	nc[1]	Test Data Output (JTAG). A serial data output of the scan path.
TMS[2]	I (pu[3])	nc[1]	Test Mode Select (JTAG). Used to control the test state machine.
$\overline{\text{TRST}}$[2]	I/A (pu[3])	au	Test Reset (JTAG). Resets the test state machine. $\overline{\text{TRST}}$ must be asserted or pulsed low after power-up for proper device operation.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] See the reference Page 11 to the JTAG emulation technical reference EE-68.
[2] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[3] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.

Table 10. Pin Definitions—Flags, Interrupts, and Timer

Signal	Type	Term	Description
FLAG3–0[1]	I/O/A (pd[2])	nc	FLAG pins. Bidirectional input/output pins can be used as program conditions. Each pin can be configured individually for input or for output. FLAG3–0 are inputs after power-up and reset.
$\overline{\text{IRQ3–0}}$[3]	I/A (pu[2])	nc	Interrupt Request. When asserted, the DSP generates an interrupt. Each of the $\overline{\text{IRQ3–0}}$ pins can be independently set for edge triggered or level sensitive operation. After reset, these pins are disabled unless the $\overline{\text{IRQ3–0}}$ strap option is initialized for booting.
TMR0E[1]	O (pd[2])	au	Timer 0 expires. This output pulses for four SCLK cycles whenever timer 0 expires. At reset this is a strap pin. For additional information, see Table 16 on Page 19.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

[1] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.
[3] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.

Table 11. Pin Definitions—Link Ports

Signal	Type	Term	Description
L0DAT7–0[1]	I/O	nc	Link0 Data 7–0
L1DAT7–0[1]	I/O	nc	Link1 Data 7–0
L2DAT7–0[1]	I/O	nc	Link2 Data 7–0
L3DAT7–0[1]	I/O	nc	Link3 Data 7–0
L0CLKOUT	O	nc	Link0 Clock/Acknowledge Output
L1CLKOUT	O	nc	Link1 Clock/Acknowledge Output
L2CLKOUT	O	nc	Link2 Clock/Acknowledge Output
L3CLKOUT	O	nc	Link3 Clock/Acknowledge Output
L0CLKIN	I/A	epu	Link0 Clock/Acknowledge Input
L1CLKIN	I/A	epu	Link1 Clock/Acknowledge Input
L2CLKIN	I/A	epu	Link2 Clock/Acknowledge Input
L3CLKIN	I/A	epu	Link3 Clock/Acknowledge Input
L0DIR	O	nc	Link0 Direction. (0 = input, 1 = output)

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

Table 11. Pin Definitions—Link Ports (Continued)

Signal	Type	Term	Description
L1DIR	O	nc	Link1 Direction. (0 = input, 1 = output)
L2DIR[2]	O (pd[3])	au	Link2 Direction. (0 = input, 1 = output) At reset this is a strap pin. For more information, see Table 16 on Page 19.
L3DIR	O (pd[3])	nc	Link3 Direction. (0 = input, 1 = output)
Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state			
Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.			

[1] The link port data pins, if connected or floated for extended periods (for example, token slave with no token master), do not require pull-ups or pull-downs as there are no reliability issues and the worst-case power consumption for these floating inputs is negligible. Floating in this case means that these inputs are not driven by any source and that dc-biased terminations are not present.
[2] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[3] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.

Table 12. Pin Definitions—Impedance and Drive Strength Control

Signal	Type	Term	Description
CONTROLIMP2–1[1] CONTROLIMP0[2]	I (pu[3]) I (pd[3])	au au	Impedance Control. For ADC (Address/Data/Controls) and LINK (all link port outputs) signals, the CONTROLIMP2–0 pins control impedance as shown in Table 13. These pins enable or disable dig_ctrl mode. When dig_ctrl: 0 = Disabled (maximum drive strength) 1 = Enabled (use DS2–0 drive strength selection)
DS2–0[1]	I (pu[3])	au	Digital Drive Strength Selection. Selected as shown in Table 14. For drive strength calculation, see Output Drive Currents on Page 32. The drive strength for some pins is preset, not controlled by the DS2–0 pins. The pins that are always at drive strength 7 (100%) are: $\overline{CPA}$, $\overline{DPA}$, and $\overline{EMU}$.
Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state			
Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.			

[1] The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2] The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[3] See Electrical Characteristics on Page 20 for maximum and minimum current consumption for pull-up and pull-down resistances.

Table 13. Control Impedance Selection

CONTROLIMP2–0	ADC dig_ctrl	LINK dig_ctrl
000	0	0
001	0	0
010	0	1
011	reserved	reserved
100	1	0
101	reserved	reserved
110 (default)	1	1
111	reserved	reserved

Table 14. Drive Strength Selection

DS2–0	Drive Strength
000	Strength 0
001	Strength 1
010	Strength 2
011	Strength 3
100	Strength 4
101	Strength 5
110	Strength 6
111 (default)	Strength 7

Table 15. Pin Definitions—Power, Ground, and Reference

Signal	Type	Term	Description
V_{DD}	P	au	V_{DD} pins for internal logic.
V_{DD_A}	P	au	V_{DD} pins for analog circuits. Pay critical attention to bypassing this supply.
V_{DD_IO}	P	au	V_{DD} pins for I/O buffers.
V_{REF}	I	au	Reference voltage defines the trip point for all input buffers, except $\overline{RESET}$, $\overline{IRQ3–0}$, $\overline{DMAR3–0}$, ID2–0, CONTROLIMP2–0, TCK, TDI, TMS, and $\overline{TRST}$. The value is 1.5 V ± 100 mV (which is the TTL trip point). V_{REF} can be connected to a power supply or set by a voltage divider circuit. The voltage divider should have an HF decoupling capacitor (1 nF HF SMD) connected to V_{SS}. Tie the decoupling capacitor between V_{REF} input and V_{SS}, as close to the DSP's pins as possible. For more information, see Filtering Reference Voltage and Clocks on Page 10.
V_{SS}	G	au	Ground pins.
V_{SS_A}	G	au	Ground pins for analog circuits.
NC			No connect. Do not connect these pins to anything (not to any supply, signal, or each other), because they are reserved and must be left unconnected.

Type column symbols: A = asynchronous; G = ground; I = input; O = output; o/d = open drain output; P = power supply; pd = internal pull-down approximately 100 kΩ; pu = internal pull-up approximately 100 kΩ; T = three-state

Term (for termination) column symbols: epd = external pull-down approximately 10 kΩ to V_{SS}; epu = external pull-up approximately 10 kΩ to V_{DD-IO}, nc = not connected; au = always used.

STRAP PIN FUNCTION DESCRIPTIONS

Some pins have alternate functions at reset. Strap options set DSP operating modes. During reset, the DSP samples the strap option pins. Strap pins have an approximately 100 kΩ pull-down for the default value. If a strap pin is not connected to an external pull-up or logic load, the DSP samples the default value during reset. If strap pins are connected to logic inputs, a stronger external pull-down may be required to ensure default value depending on leakage and/or low level input current of the logic load. To set a mode other than the default mode, connect the strap pin to a sufficiently stronger external pull-up. In a multiprocessor system, up to eight DSPs may be connected on the cluster bus, resulting in parallel combination of strap pin pull-down resistors. Table 16 lists and describes each of the DSP's strap pins.

Table 16. Pin Definitions—I/O Strap Pins

Signal	On Pin …	Description
EBOOT	$\overline{BMS}$	EPROM boot. 0 = boot from EPROM immediately after reset (default) 1 = idle after reset and wait for an external device to boot DSP through the external port or a link port
IRQEN	$\overline{BM}$	Interrupt Enable. 0 = disable and set $\overline{IRQ3–0}$ interrupts to level sensitive after reset (default) 1 = enable and set $\overline{IRQ3–0}$ interrupts to edge sensitive immediately after reset
TM1	L2DIR	Test Mode 1. 0 = required setting during reset. 1 = reserved.
TM2	TMR0E	Test Mode 2. 0 = required setting during reset. 1 = reserved.

SPECIFICATIONS

Note that component specifications are subject to change without notice.

OPERATING CONDITIONS

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{DD}	Internal Supply Voltage		1.14		1.26	V
V_{DD_A}	Analog Supply Voltage		1.14		1.26	V
V_{DD_IO}	I/O Supply Voltage		3.15		3.45	V
T_{CASE}	Case Operating Temperature		–40		+85	°C
V_{IH}	High Level Input Voltage[1]	@ V_{DD}, V_{DD_IO} = max	2		V_{DD_IO} + 0.5	V
V_{IL}	Low Level Input Voltage[1]	@ V_{DD}, V_{DD_IO} = min	–0.5		+0.8	V
I_{DD}	V_{DD} Supply Current for Typical Activity[2]	@ CCLK = 250 MHz, V_{DD} = 1.25 V, T_{CASE} = 25°C		1.2		A
I_{DD}	V_{DD} Supply Current for Typical Activity[2]	@ CCLK = 300 MHz, V_{DD} = 1.25 V, T_{CASE} = 25°C		1.5		A
$I_{DDIDLELP}$	V_{DD} Supply Current for IDLELP Instruction Execution	@ CCLK = 300 MHz, V_{DD} = 1.20 V, T_{CASE} = 25°C		173		mA
I_{DD_IO}	V_{DD_IO} Supply Current for Typical Activity[2]	@ SCLK = 100 MHz, V_{DD_IO} = 3.3 V, T_{CASE} = 25°C		137		mA
I_{DD_A}	V_{DD_A} Supply Current	@ V_{DD} = 1.25 V, T_{CASE} = 25°C		25	31.25	mA
V_{REF}	Voltage Reference		1.4		1.6	V

[1] Applies to input and bidirectional pins.
[2] For details on internal and external power estimation, including: power vector definitions, current usage descriptions, and formulas, see *EE-169, Estimating Power for the ADSP-TS101S* on the Analog Devices website—use site search on "EE-169" (www.analog.com). This document is updated regularly to keep pace with silicon revisions.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Max	Unit
V_{OH}	High Level Output Voltage[1]	@V_{DD_IO} = min, I_{OH} = –2 mA	2.4		V
V_{OL}	Low Level Output Voltage[1]	@V_{DD_IO} = min, I_{OL} = 4 mA		0.4	V
I_{IH}	High Level Input Current[2]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max		10	µA
I_{IHP}	High Level Input Current (pd)[2]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max	17.2	44.5	µA
I_{IL}	Low Level Input Current[3]	@V_{DD_IO} = max, V_{IN} = 0 V		10	µA
I_{ILP}	Low Level Input Current (pu)[4]	@V_{DD_IO} = max, V_{IN} = 0 V	–69	–23	µA
I_{OZH}	Three-State Leakage Current High[5, 6]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max		10	µA
I_{OZHP}	Three-State Leakage Current High (pd)[7]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max	17.2	44.5	µA
I_{OZL}	Three-State Leakage Current Low[8]	@V_{DD_IO} = max, V_{IN} = 0 V		10	µA
I_{OZLP}	Three-State Leakage Current Low (pu)[9]	@V_{DD_IO} = max, V_{IN} = 0 V	–69	–23	µA
I_{OZLO}	Three-State Leakage Current Low (od)[7]	@V_{DD_IO} = max, V_{IN} = 0 V	–9.8	–4.6	mA
C_{IN}	Input Capacitance[10, 11]	@f_{IN} = 1 MHz, T_{CASE} = 25°C, V_{IN} = 2.5 V		5	pF

[1] Applies to output and bidirectional pins.
[2] Applies to input pins with internal pull-downs (pd).
[3] Applies to input pins without internal pull-ups (pu).
[4] Applies to input pins with internal pull-ups (pu).
[5] Applies to three-stateable pins without internal pull-downs (pd).
[6] Applies to open drain (od) pins with 500 Ω pull-ups (pu).
[7] Applies to three-stateable pins with internal pull-downs (pd).
[8] Applies to three-stateable pins without internal pull-ups (pu).
[9] Applies to three-stateable pins with internal pull-ups (pu).
[10] Applies to all signals.
[11] Guaranteed but not tested.

ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed in Table 19 may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 17. Absolute Maximum Ratings

Parameter	Rating
Internal (Core) Supply Voltage (V_{DDINT})	–0.3 V to +1.40 V
Analog (PLL) Supply Voltage (V_{DD_A})	–0.3 V to +1.40 V
External (I/O) Supply Voltage (V_{DDEXT})	–0.3 V to +4.6 V
Input Voltage	–0.5 V to V_{DD_IO} + 0.5 V
Output Voltage Swing	–0.5 V to V_{DD_IO} + 0.5 V
Storage Temperature Range	–65°C to +150°C

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PACKAGE INFORMATION

The information presented in Figure 7 provide details about the package branding for the ADSP-TS101S processors. For a complete listing of product availability, see Ordering Guide on Page 45.

Figure 7. Typical Package Brand

Table 18. Package Brand Information

Brand Key	Field Description
t	Temperature Range
pp	Package Type
Z	Lead Free Option (optional)
ccc	See Ordering Guide
LLLLLLLLL-L	Silicon Lot Number
R.R	Silicon Revision
yyww	Date Code
vvvvvv	Assembly Lot Code

TIMING SPECIFICATIONS

With the exception of link port, $\overline{\text{IRQ3–0}}$, $\overline{\text{DMAR3–0}}$, TMR0E, FLAG3–0 (input), and $\overline{\text{TRST}}$ pins, all ac timing for the ADSP-TS101S is relative to a reference clock edge. Because input setup/hold, output valid/hold, and output enable/disable times are relative to a clock edge, the timing data for the ADSP-TS101S has few calculated (formula-based) values. For information on ac timing, see General AC Timing. For information on link port transfer timing, see Link Ports Data Transfer and Token Switch Timing on Page 29.

General AC Timing

Timing is measured on signals when they cross the 1.5 V level as described in Figure 16 on Page 28. All delays (in nanoseconds) are measured between the point that the first signal reaches 1.5 V and the point that the second signal reaches 1.5 V.

The ac asynchronous timing data for the $\overline{\text{IRQ3–0}}$, $\overline{\text{DMAR3–0}}$, TMR0E, FLAG3–0 (input), and $\overline{\text{TRST}}$ pins appears in Table 21.

The general ac timing data appears in Table 21, Table 29, and Table 30. All ac specifications are measured with the load specified in Figure 8, and with the output drive strength set to strength 4. Output valid and hold are based on standard capacitive loads: 30 pF on all pins. The delay and hold specifications given should be derated by a drive strength related factor for loads other than the nominal value of 30 pF.

In order to calculate the output valid and hold times for different load conditions and/or output drive strengths, refer to Figure 32 on Page 34 through Figure 39 on Page 36 (Rise and Fall Time vs. Load Capacitance) and Figure 40 on Page 36 (Output Valid vs. Load Capacitance and Drive Strength).

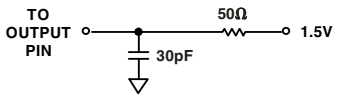

Figure 8. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

For power-up sequencing, power-up reset, and normal reset (hot reset) timing requirements, refer to Table 26 and Figure 13, Table 27 and Figure 14, and Table 28, and Figure 15 respectively.

Table 19. AC Asynchronous Signal Specifications (All values in this table are in nanoseconds)

Name	Description	Pulse Width Low (min)	Pulse Width High (min)
$\overline{IRQ3–0}$[1]	Interrupt request input	t_{CCLK} + 3 ns	
$\overline{DMAR3–0}$[1]	DMA request input	t_{CCLK} + 4 ns	t_{CCLK} + 4 ns
TMR0E[2]	Timer 0 expired output		4 × t_{SCLK} ns
FLAG3–0[1, 3]	Flag pins input	3 × t_{CCLK} ns	3 × t_{CCLK} ns
$\overline{TRST}$	JTAG test reset input	1 ns	

[1] These input pins do not need to be synchronized to a clock reference.
[2] This pin is a strap option. During reset, an internal resistor pulls the pin low.
[3] For output specifications, see Table 29 and Table 30.

Table 20. Reference Clocks—Core Clock (CCLK) Cycle Time

Parameter	Description	Grade = 100 (300 MHz) Min	Grade = 100 (300 MHz) Max	Grade = 000 (250 MHz) Min	Grade = 000 (250 MHz) Max	Unit
t_{CCLK}[1]	Core Clock Cycle Time	3.3	12.5	4.0	12.5	ns

[1] CCLK is the internal processor clock or instruction cycle time. The period of this clock is equal to the system clock period (t_{SCLK}) divided by the system clock ratio (SCLKRAT2–0). For information on available part numbers for different internal processor clock rates, see the Ordering Guide on Page 45.

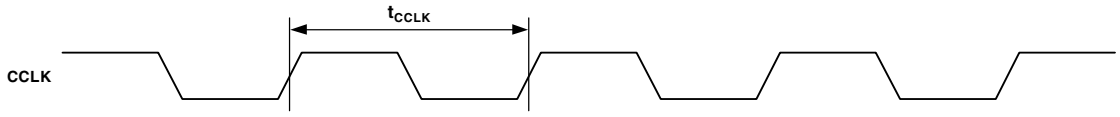

Figure 9. Reference Clocks—Core Clock (CCLK) Cycle Time

Table 21. Reference Clocks—Local Clock (LCLK) Cycle Time

Parameter	Description	Min	Max	Unit
t_{LCLK}[1, 2, 3, 4]	Local Clock Cycle Time	10	25	ns
t_{LCLKH}	Local Clock Cycle High Time	0.4 × t_{LCLK}	0.6 × t_{LCLK}	ns
t_{LCLKL}	Local Clock Cycle Low Time	0.4 × t_{LCLK}	0.6 × t_{LCLK}	ns
t_{LCLKJ}[5, 6]	Local Clock Jitter Tolerance		500	ps

[1] For more information, see Table 3 on Page 12.
[2] For more information, see Clock Domains on Page 9.
[3] LCLK_P and SCLK_P must be connected to the same source.
[4] The value of (t_{LCLK} / LCLKRAT2-0) must not violate the specification for t_{CCLK}.
[5] Actual input jitter should be combined with ac specifications for accurate timing analysis.
[6] Jitter specification is maximum peak-to-peak time interval error (TIE) jitter.

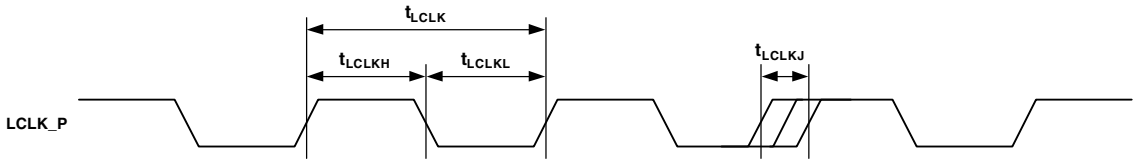

Figure 10. Reference Clocks—Local Clock (LCLK) Cycle Time