ANALOG DEVICES

TIGER SHARC®

DSP Microcomputer

ADSP-TS101S

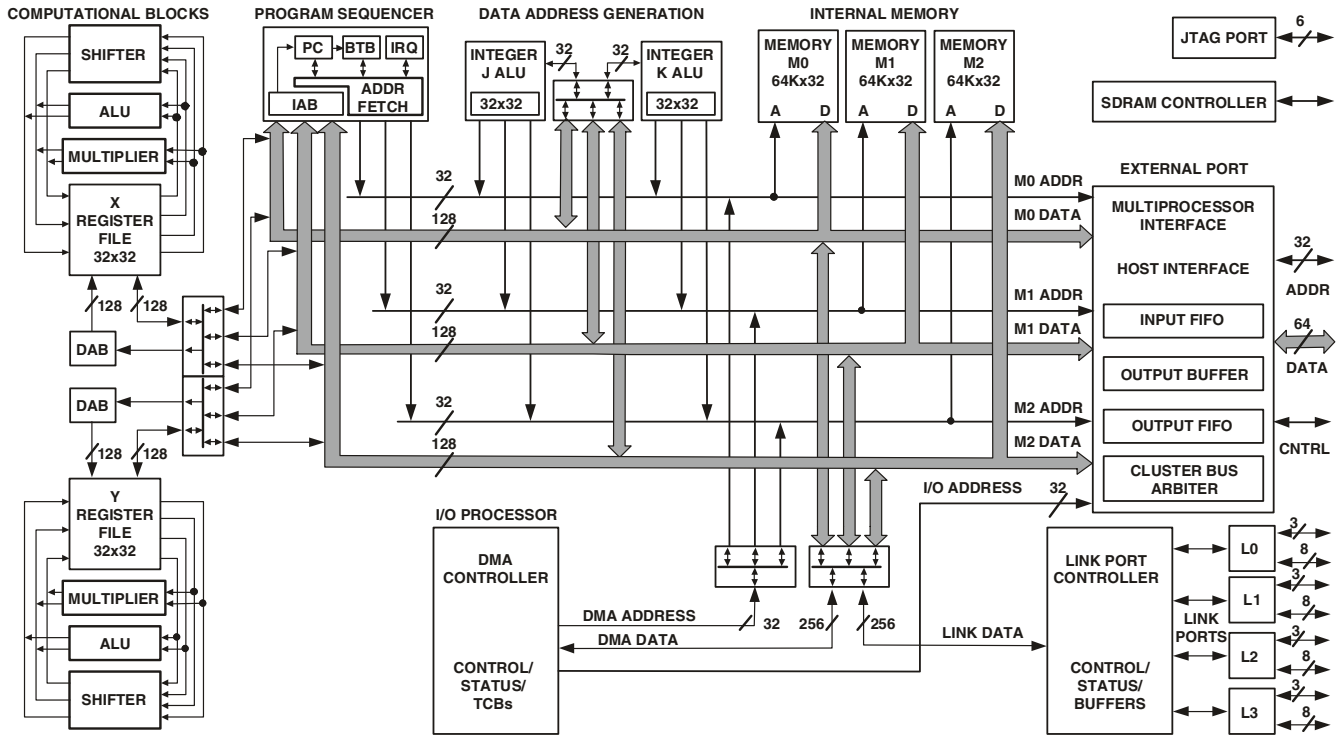
KEY FEATURES

- 250 MHz, 4.0 ns Instruction Cycle Rate
- 6M Bits of Internal—On-Chip—SRAM Memory
- 19 × 19 mm (484-Ball) or 27 × 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, 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 with up to Eight TigerSHARC® DSPs on a Common Bus

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 and Table 2)
- 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



REV. 0

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

TABLE OF CONTENTS

GENERAL DESCRIPTION

The ADSP-TS101S TigerSHARC DSP 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 8-, 16-, 32-, and 64-bit fixed-point processing—to set a new standard of performance for digital signal processors. The TigerSHARC DSP's static superscalar architecture lets the processor execute up to four instructions each cycle, performing twenty-four 16-bit fixed-point 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 12G bytes per second of internal memory bandwidth. Operating at 250 MHz, the ADSP-TS101S DSP's core has a 4.0 ns instruction cycle time. Using its Single-Instruction, Multiple-Data (SIMD) features, the ADSP-TS101S can perform 2 billion 40-bit MACs or 500 million 80-bit MACs per second. Table 1 and Table 2 show the DSP's performance benchmarks.

Table 1. General-Purpose Algorithm Benchmarks at 250 MHz

Benchmark	Speed	Clock Cycles
32-bit Algorithm, 500 million MACs/s peak performance		
1024 Point Complex FFT (Radix 2)	39.34 µs	9,835
50-tap FIR on 1024 input	110 µs	27,500
Single FIR MAC	2.2 ns	0.55
16-bit Algorithm, 2 billion MACs/s peak performance		
256 Point Complex FFT (Radix 2)	4.4 µs	1,100
50-tap FIR on 1024 input	28.8 µs	7,200
Single FIR MAC	0.56 ns	0.14
Single Complex FIR MAC	2.28 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
Viterbi Decode 12.2 kbps AMR[2] Voice Channel	0.86 MIPS
Complex Correlation 3.84 Mcps[3] with a Spreading Factor of 256	0.27 MIPS

[1]The Execution Speed is in Instruction Cycles Per Second.
[2]Adaptive Multi Rate (AMR)
[3]Megachips per second (Mcps)

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

The Functional Block Diagram on Page 1 shows the ADSP-TS101S DSP'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 32-bit interval timers and timer expired pin
- A 1149.1 IEEE compliant JTAG test access port for on-chip emulation

Figure 1 shows a typical single processor system with external SDRAM. Figure 3 on Page 7 shows a typical multiprocessor system.

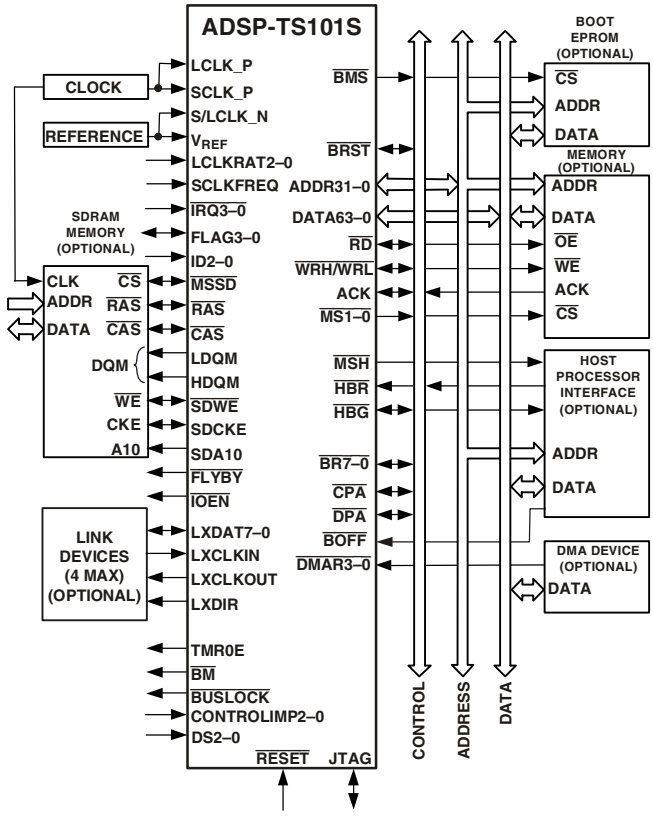


Figure 1. Single Processor System with External SDRAM

The TigerSHARC DSP uses a Static Superscalar™ architecture. This architecture is superscalar in that the ADSP-TS101S DSP'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.

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.

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.

Static Superscalar is a trademark of Analog Devices, Inc.

- 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 twenty-four 16-bit fixed-point operations per cycle, providing 1500 MFLOPS or 6.0 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 two quad word FIFO that enables loading of quad word data from non-aligned 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 DSP'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 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 2) 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 12G 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 (80 ns)

External Port (Off-Chip Memory/Peripherals Interface)

The ADSP-TS101S DSP'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 2) that enables direct interprocessor accesses of each ADSP-TS101S DSP'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 DSPs 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.

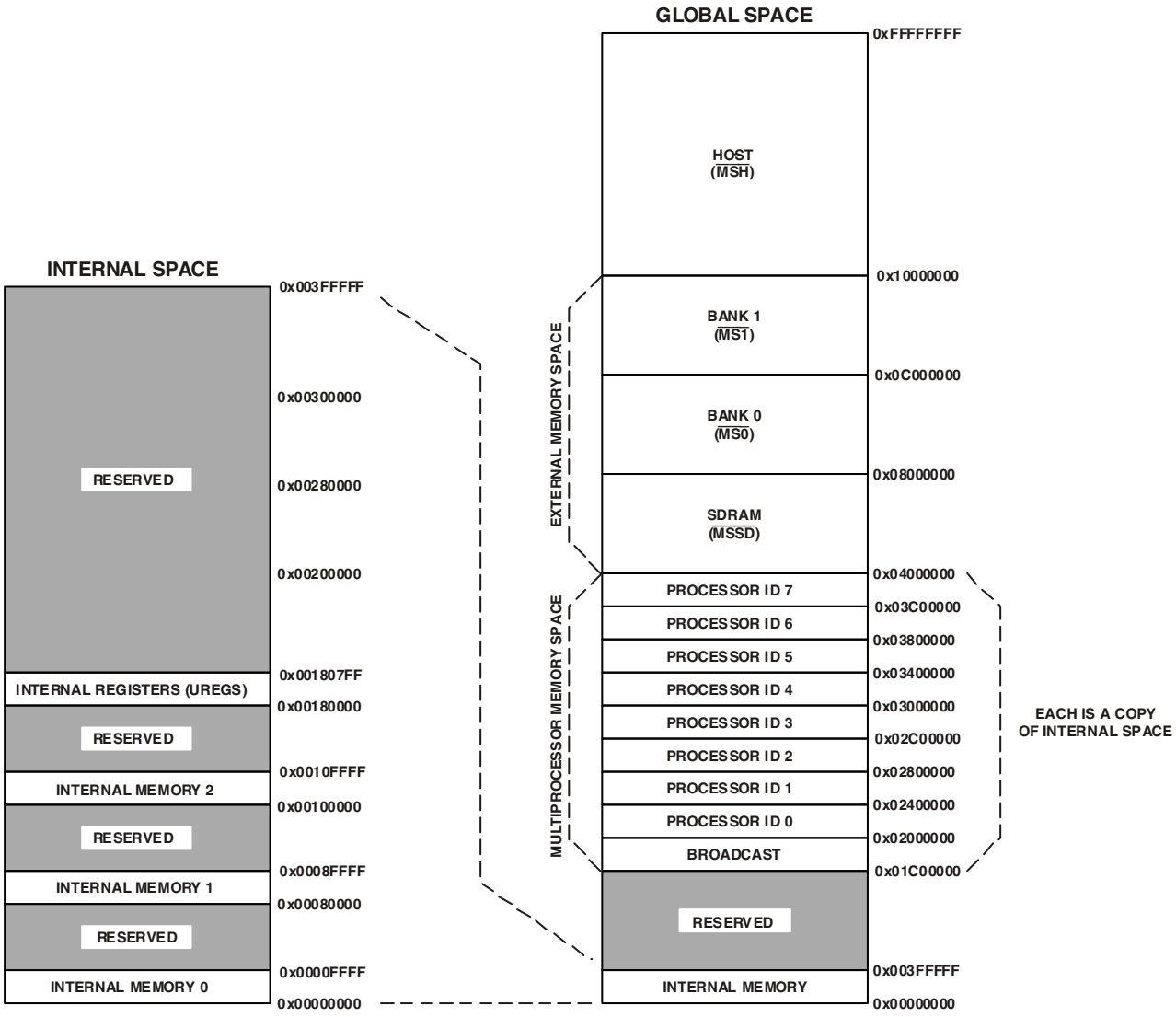


Figure 2. Memory Map

SDRAM Controller

The SDRAM controller controls the ADSP-TS101S DSP'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 bit 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 DSP'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

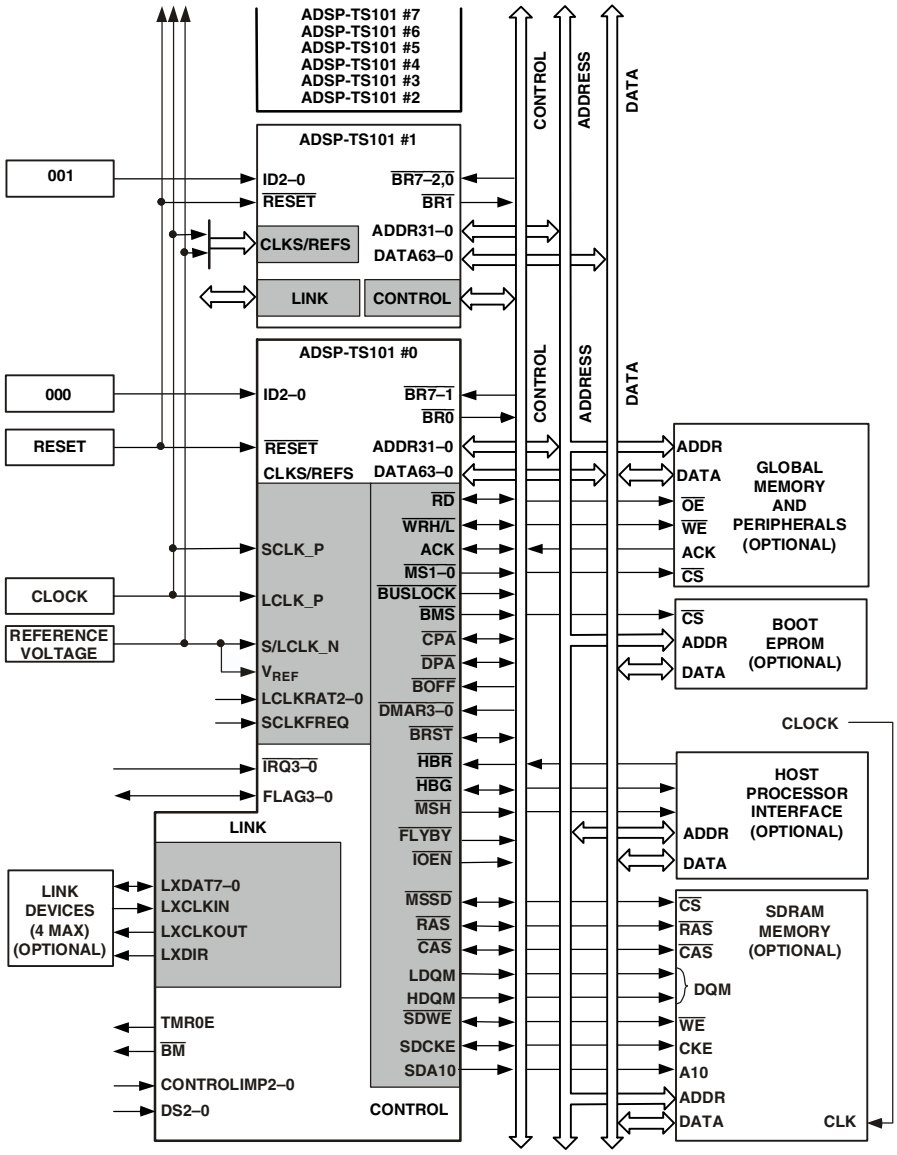


Figure 3. Shared Memory Multiprocessing System

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 ($\overline{MS1–0}$, $\overline{MSSD}$, $\overline{RAS}$, $\overline{CAS}$, and $\overline{SDWE}$) and the $\overline{FLYBY}$, $\overline{IOEN}$, and $\overline{RD}/\overline{WR}$ 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.

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.

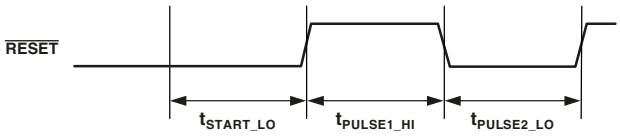
Timer and General-Purpose I/O

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 three levels of reset:

- Power-up reset—After power-up of the system, and strap options are stable, the $\overline{\text{RESET}}$ pin must be asserted (low) for a minimum of 2 ms followed by a deasserted (high) pulse of a minimum of 50 SCLK cycles and asserted (low) for a minimum of 100 SCLK cycles. $\overline{\text{TRST}}$ must also be asserted (low) during power-up to ensure proper operation of the device. See Figure 4.
- Normal reset—For any resets following the power-up reset sequence, the $\overline{\text{RESET}}$ pin must be asserted for at least 100 SCLK cycles.
- Core reset—When setting the SQRST bit in SQCTL, the core is reset, but not the external port or I/O.



NOTES:

t_{START_LO} = 2ms MINIMUM AFTER POWER SUPPLIES ARE STABLE
t_{PULSE1_HI} = 50×t_{SCLK} MINIMUM TO 100×t_{SCLK} MAXIMUM
t_{PULSE2_LO} = 100×t_{SCLK} MINIMUM

Figure 4. Power-up Reset Waveform

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 auto DMA.
- 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

The ADSP-TS101S has two clock inputs that 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. The maximum SCLK frequency is the minimum of either 100 MHz or CCLK/2, where CCLK is the internal DSP clock frequency. SCLK must be connected to the same clock source as LCLK.
- 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 generates CCLK which is phase-locked. The LCLKRAT pins define the clock multiplication of LCLK to CCLK (see Table 4 on Page 12). 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.

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 ideal power-on sequence for the DSP is to provide power-up of all supplies simultaneously. If there is going to be some delay between power-up of the supplies, provide V_{DD} (and V_{DD_A}) first, then V_{DD_IO}.

Filtering Reference Voltage and Clocks

Figure 5 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.

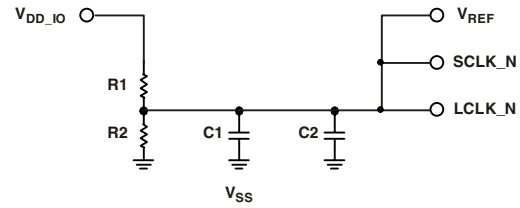


Figure 5. V_{REF}, SCLK_N, and LCLK_N Filter

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 DSPs 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++ runtime 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 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 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-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 DSP 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's design must include the interface between an Analog Devices JTAG DSP and the emulation header on a custom DSP target board.

Target Board Header

The emulator interface to an Analog Devices JTAG DSP is a 14-pin header, as shown in Figure 6. The customer must supply this header on the target board in order to communicate with the emulator. The interface consists of a standard dual row 0.025" square post header, set on 0.1" × 0.1" spacing, with a minimum post length of 0.235". Pin 3 is the key position used to prevent the pod from being inserted backwards. This pin must be clipped on the target board.

Also, the clearance (length, width, and height) around the header must be considered. Leave a clearance of at least 0.15" and 0.10" around the length and width of the header, and reserve a height clearance to attach and detach the pod connector.

As can be seen in Figure 6, there are two sets of signals on the header. There are the standard JTAG signals TMS, TCK, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{EMU}}$ used for emulation purposes (via an emulator). There are also secondary JTAG signals BTMS, BTCK, BTDI, and $\overline{\text{BTRST}}$ that are optionally used for board-level (boundary scan) testing.

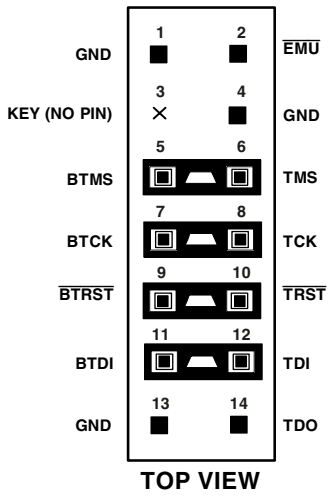


Figure 6. JTAG Target Board Connector for JTAG Equipped Analog Devices DSP (Jumpers in Place)

When the emulator is not connected to this header, place jumpers across BTMS, BTCK, BTRST, and BTDI as shown in Figure 7. These jumpers hold the JTAG signals in the correct state to allow the DSP to run free. Remove all the jumpers when connecting the emulator to the JTAG header.

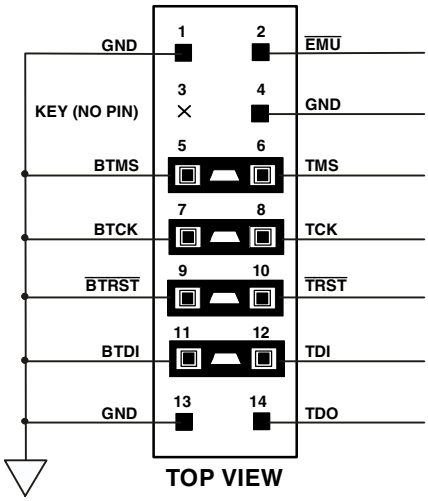


Figure 7. JTAG Target Board Connector with No Local Boundary Scan

JTAG Emulator Pod Connector

Figure 8 details the dimensions of the JTAG pod connector at the 14-pin target end. Figure 9 displays the keep-out area for a target board header. The keep-out area allows the pod connector to properly seat onto the target board header. This board area should contain no components (chips, resistors, capacitors, and so on). The dimensions are referenced to the center of the 0.25" square post pin.

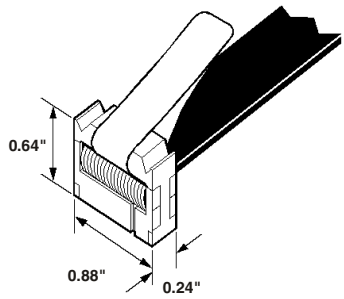


Figure 8. JTAG Pod Connector Dimensions

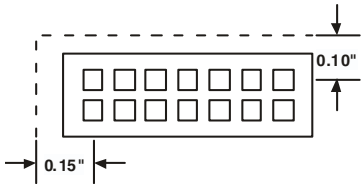


Figure 9. JTAG Pod Connector Keep-Out Area

Design for Emulation Circuit Information

For details on target board design issues including 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)—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 DSP's architecture and functionality. For detailed information on the ADSP-TS101S DSP's core architecture and instruction set, see the *TigerSHARC DSP Hardware Specification* and the *TigerSHARC DSP Instruction Set Specification*. For detailed information on the development tools for this processor, see the *VisualDSP++ User's Guide for TigerSHARC DSP.*

PIN FUNCTION DESCRIPTIONS

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

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.

Table 3. Pin Definitions—Clocks and Reset

Signal	Type	Description
LCLK_N	I	Local Clock Reference. Connect this pin to V_{REF} as shown in Figure 5.
LCLK_P	I	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. See Clock Domains on Page 9.
LCLKRAT2–0[1]	I (pd[2])	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	System Clock Reference. Connect this pin to V_{REF} as shown in Figure 5.
SCLK_P	I	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. See Clock Domains on Page 9.
SCLKFREQ[3]	I (pu[2])	SCLK Frequency. SCLKFREQ = 1 is required. The SCLKFREQ pin must have a constant value while the DSP is powered.
$\overline{\text{RESET}}$	I/A	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 8.

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

[1]The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2]See Electrical Characteristics on Page 20.
[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	Description
ADDR31–0[1]	I/O/T	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 DSPs. 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	External Data Bus. The DSP drives and receives data and instructions on these pins.
$\overline{RD}$[2]	I/O/T (pu^3)	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)	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)	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	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)	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 8 and the EBOOT signal description in Table 16 on Page 19.

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

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

Signal	Type	Description
$\overline{MS1–0}$[2]	O/T (pu[3])	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 $\overline{MS1–0}$.
$\overline{MSH}$[2]	O/T (pu[3])	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])	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.

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

[1]The address and data buses may float for several cycles during bus mastership transitions between a TigerSHARC DSP 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.
[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	Description
$\overline{BR7–0}$	I/O	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])	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])	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	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.
$\overline{BUSLOCK}$[3]	O/T (pu[2])	Bus Lock Indication. Provides an indication that the current bus master has locked the bus.
$\overline{HBR}$	I	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.

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

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

Signal	Type	Description
$\overline{HBG}$[3]	I/O/T (pu[2])	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)	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 not required in the system, leave $\overline{CPA}$ unconnected (external pull-ups will be required for ID1–ID7).
$\overline{DPA}$	I/O (o/d)	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 not required in the system, leave $\overline{DPA}$ unconnected (external pull-ups will be required for IDs 1 through 7).

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

[1]The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2]See Electrical Characteristics on Page 20.
[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	Description
$\overline{DMAR3–0}$	I/A	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])	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])	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.

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

[1]The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2]See Electrical Characteristics on Page 20.

Table 8. Pin Definitions—External Port SDRAM Controller

Signal	Type	Description
$\overline{\text{MSSD}}$[1]	I/O/T (pu[2])	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). In a multiprocessor system, the master DSP drives $\overline{\text{MSSD}}$.
$\overline{\text{RAS}}$[1]	I/O/T (pu[2])	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, $\overline{\text{RAS}}$ defines the type of operation to execute according to SDRAM specification.
$\overline{\text{CAS}}$[1]	I/O/T (pu[2])	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])	Low Word SDRAM Data Mask. When LDQM 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])	High Word SDRAM Data Mask. When HDQM 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])	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])	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])	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.

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

[1]The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2]See Electrical Characteristics on Page 20.
[3]The internal pull-down may not be sufficient. A stronger pull-down may be necessary.

Table 9. Pin Definitions—JTAG Port

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

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

Table 9. Pin Definitions—JTAG Port (continued)

Signal	Type	Description
TDO	O/T	Test Data Output (JTAG). A serial data output of the scan path.
TMS[1]	I (pu[2])	Test Mode Select (JTAG). Used to control the test state machine.
$\overline{\text{TRST}}$[1]	I/A (pu[2])	Test Reset (JTAG). Resets the test state machine. $\overline{\text{TRST}}$ must be asserted or pulsed low after power-up for proper device operation. For more information, see Reset and Booting on Page 8.

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

[1]The internal pull-up may not be sufficient. A stronger pull-up may be necessary.
[2]See Electrical Characteristics on Page 20.

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

Signal	Type	Description
FLAG3–0[1]	I/O/A (pd[2])	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])	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])	Timer 0 expires. This output pulses for four SCLK cycles whenever timer 0 expires. At reset this is a strap pin. For more information, see Table 16 on Page 19.

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

[1]The internal pull-down may not be sufficient. A stronger pull-down may be necessary.
[2]See Electrical Characteristics on Page 20.
[3]The internal pull-up may not be sufficient. A stronger pull-up may be necessary.

Table 11. Pin Definitions—Link Ports

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

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

Table 11. Pin Definitions—Link Ports (continued)

Signal	Type	Description
L1DIR	O	Link1 Direction. (0 = input, 1 = output)
L2DIR[2]	O (pd[3])	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])	Link3 Direction. (0 = input, 1 = output)

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

[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.

Table 12. Pin Definitions—Impedance and Drive Strength Control

Signal	Type	Description
CONTROLIMP2–1[1] CONTROLIMP0[2]	I (pu[3]) I (pd[3])	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])	Digital Drive Strength Selection. Selected as shown in Table 14. For drive strength calculation, see Output Drive Currents on Page 28.

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

[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.

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	Description
V_{DD}	P	V_{DD} pins for internal logic.
V_{DD_A}	P	V_{DD} pins for analog circuits. Pay critical attention to bypassing this supply.
V_{DD_IO}	P	V_{DD} pins for I/O buffers.
V_{REF}	I	Reference voltage defines the trip point for all input buffers, except $\overline{\text{RESET}}$, $\overline{\text{IRQ3–0}}$, $\overline{\text{DMAR3–0}}$, ID2–0, CONTROLIMP2–0, TCK, TDI, TMS, and $\overline{\text{TRST}}$. 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. See Filtering Reference Voltage and Clocks on Page 9.
V_{SS}	G	Ground pins.
V_{SS_A}	G	Ground pins for analog circuits.

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

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. 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	$\overline{\text{BMS}}$	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	$\overline{\text{BM}}$	Interrupt Enable. 0 = disable and set $\overline{\text{IRQ3–0}}$ interrupts to level sensitive after reset (default) 1 = enable and set $\overline{\text{IRQ3–0}}$ 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

RECOMMENDED 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_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
V_{REF}	Voltage reference		1.4		1.6	V

Specifications subject to change without notice.

[1]Applies to input and bidirectional pins.
[2]For details on Typical Activity used for these measurements, see *EE-169, Estimating Power for the ADSP-TS101S*.

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)[3]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max	17.2	44.5	µA
I_{IL}	Low Level Input Current[4]	@V_{DD_IO} = max, V_{IN} = 0 V		10	µA
I_{ILP}	Low Level Input Current (pu)[5]	@V_{DD_IO} = max, V_{IN} = 0 V	–69	–23	µA
I_{OZH}	Three-State Leakage Current High[6, 7]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max		10	µA
I_{OZHP}	Three-State Leakage Current High (pd)[8]	@V_{DD_IO} = max, V_{IN} = V_{DD_IO} max	17.2	44.5	µA
I_{OZL}	Three-State Leakage Current Low[9]	@V_{DD_IO} = max, V_{IN} = 0 V		10	µA
I_{OZLP}	Three-State Leakage Current Low (pu)[10]	@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	µA
C_{IN}	Input Capacitance[11, 12]	@f_{IN} = 1 MHz, T_{CASE} = 25C, V_{IN} = 2.5 V		5	pF

Specifications subject to change without notice.

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

ABSOLUTE MAXIMUM RATINGS

Internal (Core) Supply Voltage (V_{DD})[1] . . . –0.3 V to +1.40 V
Analog (PLL) Supply Voltage (V_{DD_A})[1] . . –0.3 V to +1.40 V
External (I/O) Supply Voltage (V_{DD_IO})[1] . . . –0.3 V to +4.6 V
Input Voltage[1] . . . . . –0.5 V to V_{DD_IO}+0.5 V
Output Voltage Swing[1] . . . . . –0.5 V to V_{DD_IO}+0.5 V
Storage Temperature Range[1] . . . . . –65°C to +150°C

[1]Stresses greater than those listed above 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.

ESD SENSITIVITY

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-TS101S features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

TIMING SPECIFICATIONS

With the exception of Link port, $\overline{\text{DMAR3–0}}$, and $\overline{\text{IRQ3–0}}$ 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 on Page 22. For information on link port transfer timing, see Link Ports Data Transfer and Token Switch Timing on Page 25.

General AC Timing

Timing is measured on signals when they cross the 1.5 V level as described in Figure 10 on Page 24. 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 $\overline{\text{IRQ3–0}}$ and $\overline{\text{DMAR3–0}}$ pins appears in Table 17.

The general ac timing data appears in Table 18 and Table 19. All ac specifications are measured with the load specified in Figure 25 on Page 30, and with the output drive strength set to strength 4. In order to calculate the output valid and hold times for different load conditions and/or output drive strengths, refer to Figure 26 on Page 30 through Figure 33 on Page 31 (Rise and Fall Time vs. Load Capacitance) and Figure 34 on Page 31 (Output Valid vs. Load Capacitance vs. Drive Strength).

Table 17. AC Asynchronous Signal Specifications (all values in this table are in nanoseconds)

Name	Description	Pulsewidth Low (min)	Pulsewidth High (min)
$\overline{\text{IRQ3–0}}$[1]	Interrupt request input	t_{CCLK} + 3 ns	–
$\overline{\text{DMAR3–0}}$[1]	DMA request output	t_{CCLK} + 4 ns	t_{CCLK} + 4 ns
TMR0E	Timer 0 expired output	–	4 × t_{SCLK} ns
FLAGS3–0[1, 2]	Flag pins input	3 × t_{CCLK} ns	3 × t_{CCLK} ns
$\overline{\text{TRST}}$	JTAG test reset input	1 ns	–

[1]These input pins do not need to be synchronized to a clock reference.
[2]For output specifications, see Table 19.

Table 18. Reference Clocks

Signal	Type	Description	Speed Grade (MHz)	Clock Cycle Min (ns)	Clock Cycle Max (ns)	Clock High Min (ns)	Clock Low Min (ns)	Skew to LCLK Max (ps)	Input Jitter[1] Tolerance (ps)
CCLK[2, 3]	–	Core Clock	250	4.0	12.5	–	–	–	–
LCLK_P[4, 5, 6]	Input	Local Clock	250	CR × 4.0	CR × 12.5	{40% to 60% Duty Cycle}		–	100
SCLK_P[5, 7, 8]	Input	System Clock, SCLKFREQ = 1	All	Greater of 10 or CCLK × 2	20	{40% to 60% Duty Cycle}		50	100
TCK[9]	Input	Test Clock (JTAG)	All	Greater of 30 or CCLK × 4	–	12.5	12.5	–	–

[1]Actual input jitter should be combined with ac specifications for accurate timing analysis.
[2]CCLK is the internal DSP clock or instruction cycle time. The period of this clock is equal to the Local Clock (LCLK_P) period divided by the Local Clock Ratio (LCLKRAT2–0). For information on available internal DSP clock rates, see the ORDERING GUIDE on Page 40.
[3]The period of CCLK is t_{CCLK}.
[4]The Core Clock Ratio (CR) is 2, 2.5, 3, 3.5, 4, 5, or 6 as set by the LCLKRAT2–0 pins. For more information, see Table 4 on Page 12.
[5]See Clock Domains on Page 9.
[6]The period of LCLK is t_{LCLK}.
[7]For more information, see Table 3 on Page 12.
[8]The period of SCLK is t_{SCLK}.
[9]The period of TCK is t_{TCK}.

Table 19. AC Signal Specifications—All values in this table are in nanoseconds.

Name	Description	Input Setup (min)	Input Hold (min)	Output Valid (max)[1]	Output Hold (min)	Output Enable (min)[2]	Output Disable (max)[2]	Reference Clock
ADDR31–0	External Address Bus	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
DATA63–0	External Data Bus	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{MSH}}$	Memory Select HOST Line	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{MSSD}}$	Memory Select SDRAM Line	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{MS1–0}}$	Memory Select for Static Blocks	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{RD}}$	Memory Read	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{WRL}}$	Write Low Word	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{WRH}}$	Write High Word	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
ACK	Acknowledge for Data	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
SDCKE	SDRAM Clock Enable	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{RAS}}$	Row Address Select	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{CAS}}$	Column Address Select	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{SDWE}}$	SDRAM Write Enable	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
LDQM	Low Word SDRAM Data Mask	—	—	4.2	1.0	0.9	2.5	SCLK
HDQM	High Word SDRAM Data Mask	—	—	4.2	1.0	0.9	2.5	SCLK
SDA10	SDRAM ADDR10	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{HBR}}$	Host Bus Request	2.2	0.5	—	—	—	—	SCLK
$\overline{\text{HBG}}$	Host Bus Grant	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{BOFF}}$	Back Off Request	2.2	0.5	—	—	—	—	SCLK
$\overline{\text{BUSLOCK}}$	Bus Lock	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{BRST}}$	Burst pin	2.2	0.5	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{BR7–0}}$	Multiprocessing Bus Request pins	2.2	0.5	4.2	1.0	—	—	SCLK
$\overline{\text{FLYBY}}$	FLYBY pin	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{IOEN}}$	FLYBY pin	—	—	4.2	1.0	0.9	2.5	SCLK
$\overline{\text{CPA}}$ [3, 4]	Core Priority Access	2.2	0.5	5.8	—	—	2.5	SCLK
$\overline{\text{DPA}}$ [3, 4]	DMA Priority Access	2.2	0.5	5.8	—	—	2.5	SCLK
$\overline{\text{BMS}}$[5]	Boot Memory Select	—	—	4.2	1.0	0.9	2.5	SCLK
FLAG3–0[6]	FLAG pins	—	—	4.2	1.0	1.0	4.0	SCLK
TMR0E[5]	Timer 0 Expired	—	—	4.2	1.0	—	—	SCLK
$\overline{\text{RESET}}$[4, 7]	Global Reset pin	—	—	—	—	—	—	SCLK
TMS[4]	Test Mode Select (JTAG)	1.5	1.0	—	—	—	—	TCK
TDI[4]	Test Data Input (JTAG)	1.5	1.0	—	—	—	—	TCK
TDO	Test Data Output (JTAG)	—	—	6.0	1.0	1.0	5.0	TCK_FE[8]
$\overline{\text{TRST}}$[4, 7, 9]	Test Reset (JTAG)	—	—	—	—	—	—	TCK
$\overline{\text{BM}}$[5]	Bus Master Debug aid only	—	—	4.2	1.0	—	—	SCLK
$\overline{\text{EMU}}$[10]	Emulation	—	—	5.5	—	—	5.0	TCK or LCLK
JTAG_SYS_IN[11]	System input	1.5	11.0	—	—	—	—	TCK
JTAG_SYS_OUT[12]	System output	—	—	16.0	—	—	—	TCK_FE[8]
ID2–0[9]	Chip ID – must be constant	—	—	—	—	—	—	—
CONTROLIMP2–0[9]	Static pins – must be constant	—	—	—	—	—	—	—

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

Name	Description	Input Setup (min)	Input Hold (min)	Output Valid (max)[1]	Output Hold (min)	Output Enable (min)[2]	Output Disable (max)[2]	Reference Clock
DS2–0[9]	Static pins – must be constant	—	—	—	—	—	—	—
LCLKRAT2–0[9]	Static pins – must be constant	—	—	—	—	—	—	—
SCLKFREQ[9]	Static pins – must be constant	—	—	—	—	—	—	—

[1]The output valid (max) value in this column applies for the standard 30 pF capacitive load used in testing. To see how output valid varies with capacitive loading, see Figure 34 on Page 31.
[2]The external port protocols employ bus IDLE cycles for bus mastership transitions as well as slave address boundary crossings to avoid any potential bus contention. The apparent driver overlap, due to output disables being larger than output enables, is not actual.
[3]$\overline{\text{CPA}}$ and $\overline{\text{DPA}}$ pins are open drains and have 0.5 kΩ internal pull-ups.
[4]These input pins have Schmitt triggers and therefore do not need to be synchronized to a clock reference. These synchronous specifications only apply for recognition in the current clock reference cycle.
[5]This pin is a strap option. During reset, an internal resistor pulls the pin low.
[6]For input specifications, see Table 17.
[7]For additional requirement details, see Reset and Booting on Page 8.
[8]TCK_FE indicates TCK falling edge.
[9]These pins may change only during reset; recommend connecting it to V_{DD_IO}/V_{SS}.
[10]Reference clock depends on function.
[11]System inputs are: $\overline{\text{IRQ3–0}}$, $\overline{\text{BMS}}$, LCLKRAT2–0, SCLKFREQ, $\overline{\text{BM}}$, TMR0E, FLAG3–0, ID2–0, $\overline{\text{BRST}}$, $\overline{\text{WRH}}$, $\overline{\text{WRL}}$, $\overline{\text{RD}}$, $\overline{\text{MSSD}}$, SDCKE, $\overline{\text{SDWE}}$, $\overline{\text{CAS}}$, $\overline{\text{RAS}}$, ADDR31–0, DATA63–0, $\overline{\text{DPA}}$, $\overline{\text{CPA}}$, $\overline{\text{HBG}}$, $\overline{\text{BOFF}}$, $\overline{\text{HBR}}$, ACK, $\overline{\text{BR7–0}}$, L0CLKIN, L0DAT7–0, L1CLKIN, L1DAT7–0, L2CLKIN, L2DAT7–0, L2DIR, L3CLKIN, L3DAT7–0, DS2–0, CONTROLIMP2–0, $\overline{\text{RESET}}$, $\overline{\text{DMAR3–0}}$.
[12]System outputs are: $\overline{\text{BMS}}$, $\overline{\text{BM}}$, $\overline{\text{BUSLOCK}}$, TMR0E, FLAG3–0, $\overline{\text{FLYBY}}$, $\overline{\text{IOEN}}$, $\overline{\text{MSH}}$, $\overline{\text{BRST}}$, $\overline{\text{WRH}}$, $\overline{\text{WRL}}$, $\overline{\text{RD}}$, $\overline{\text{MS1–0}}$, HDQM, LDQM, $\overline{\text{MSSD}}$, SDCKE, $\overline{\text{SDWE}}$, $\overline{\text{CAS}}$, $\overline{\text{RAS}}$, ADDR31–0, DATA63–0, $\overline{\text{DPA}}$, $\overline{\text{CPA}}$, $\overline{\text{HBG}}$, ACK, $\overline{\text{BR7–0}}$, L0CLKOUT, L0DAT7–0, L0DIR, L1CLKOUT, L1DAT7–0, L1DIR, L2CLKOUT, L2DAT7–0, L2DIR, L3CLKOUT, L3DAT7–0, L3DIR, $\overline{\text{EMU}}$.

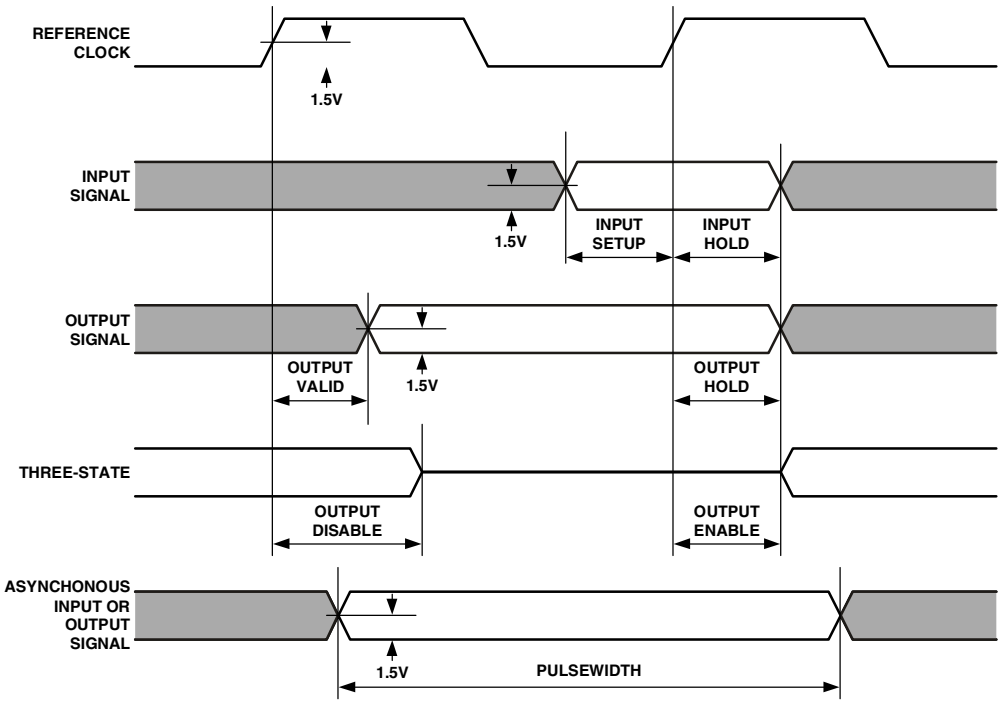


Figure 10. General AC Parameters Timing

Link Ports Data Transfer and Token Switch Timing

Table 20, Table 21, Table 22, and Table 23 with Figure 11, Figure 12, Figure 13, and Figure 14 provide the timing specifications for the link ports data transfer and token switch.

Table 20. Link Ports—Transmit

Parameter		Min	Max	Unit
Timing Requirements				
t_{CONNS}[1]	Connectivity Pulse Setup	$2 \times t_{CCLK} + 3.5$		ns
t_{CONNS}[2]	Connectivity Pulse Setup	8		ns
t_{CONNIW}[3]	Connectivity Pulse Input Width	$t_{LxCLK\_Tx} + 1$		ns
t_{ACKS}	Acknowledge Setup	$0.5 \times t_{LxCLK\_Tx}$		ns
Switching Characteristics				
t_{LxCLK_Tx}[4]	Transmit Link Clock Period	$0.9 \times LR \times t_{CCLK}$	$1.1 \times LR \times t_{CCLK}$	ns
t_{LxCLKH_Tx}[1]	Transmit Link Clock Width High	$0.33 \times t_{LxCLK\_Tx}$	$0.66 \times t_{LxCLK\_Tx}$	ns
t_{LxCLKH_Tx}[2]	Transmit Link Clock Width High	$0.4 \times t_{LxCLK\_Tx}$	$0.6 \times t_{LxCLK\_Tx}$	ns
t_{LxCLKL_Tx}[1]	Transmit Link Clock Width Low	$0.33 \times t_{LxCLK\_Tx}$	$0.66 \times t_{LxCLK\_Tx}$	ns
t_{LxCLKL_Tx}[2]	Transmit Link Clock Width Low	$0.4 \times t_{LxCLK\_Tx}$	$0.6 \times t_{LxCLK\_Tx}$	ns
t_{DIRS}	LxDIR Transmit Setup	$0.5 \times t_{LxCLK\_Tx}$	$2 \times t_{LxCLK\_Tx}$	ns
t_{DIRH}	LxDIR Transmit Hold	$0.5 \times t_{LxCLK\_Tx}$	$2 \times t_{LxCLK\_Tx}$	ns
t_{DOS}[1]	LxDAT7–0 Output Setup	$0.25 \times t_{LxCLK\_Tx} - 1$		ns
t_{DOH}[1]	LxDAT7–0 Output Hold	$0.25 \times t_{LxCLK\_Tx} - 1$		ns
t_{DOS}[2]	LxDAT7–0 Output Setup	$0.17 \times t_{LxCLK\_Tx} - 1$		ns
t_{DOH}[2]	LxDAT7–0 Output Hold	$0.17 \times t_{LxCLK\_Tx} - 1$		ns
t_{LDOE}	LxDAT7–0 Output Enable	1		ns
t_{LDOD}[5]	LxDAT7–0 Output Disable	1		ns

[1]The formula for this parameter applies when LR is 2.
[2]The formula for this parameter applies when LR is 3, 4, or 8.
[3]LxCLKIN shows the connectivity pulse with each of the three possible transitions to "Acknowledge." After a connectivity pulse low minimum, LxCLKIN may [1] return high and remain high for "Acknowledge," [2] return high and subsequently go low (meeting tacks) for "Not Acknowledge," or [3] remain low for "Not Acknowledge."
[4]The Link clock Ratio (LR) is 2, 3, 4, or 8 as set by the SPD bits in the LCTLx register.
[5]This specification applies to the last data byte or the "Dummy" byte that follows the verification byte if enabled. For more information, see the *TigerSHARC DSP Hardware Specification*.

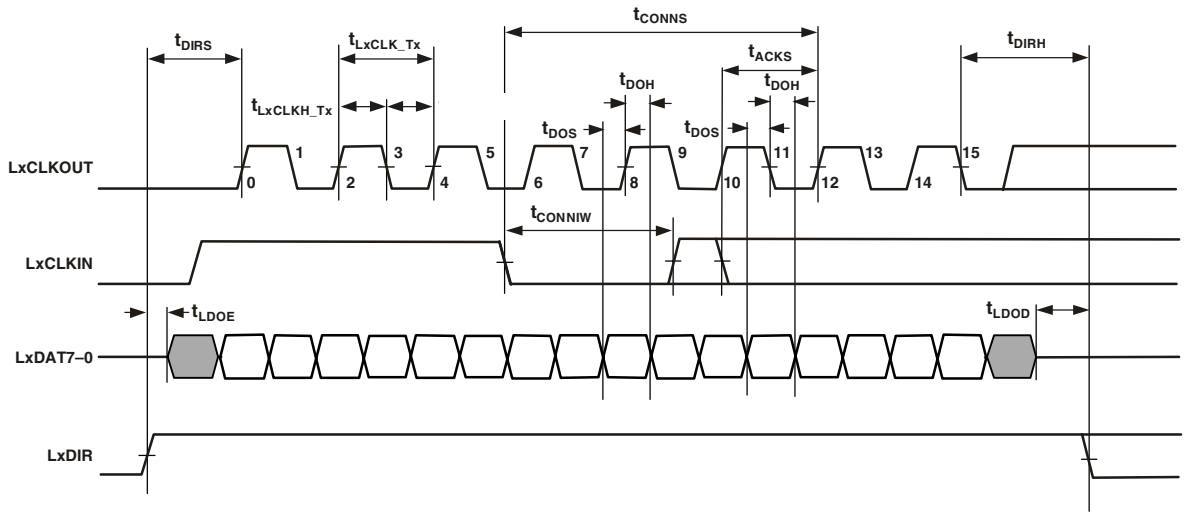


Figure 11. Link Ports—Transmit

Table 21. Link Ports—Receive

Parameter		Min	Max	Unit
Timing Requirements				
t_{LxCLK_Rx}[1]	Receive Link Clock Period	$0.9 \times LR \times t_{CCLK}$	$1.1 \times LR \times t_{CCLK}$	ns
t_{LxCLKH_Rx}[2]	Receive Link Clock Width High	$0.33 \times t_{LxCLK\_Rx}$	$0.66 \times t_{LxCLK\_Rx}$	ns
t_{LxCLKH_Rx}[3]	Receive Link Clock Width High	$0.4 \times t_{LxCLK\_Rx}$	$0.6 \times t_{LxCLK\_Rx}$	ns
t_{LxCLKL_Rx}[2]	Receive Link Clock Width Low	$0.33 \times t_{LxCLK\_Rx}$	$0.66 \times t_{LxCLK\_Rx}$	ns
t_{LxCLKL_Rx}[3]	Receive Link Clock Width Low	$0.4 \times t_{LxCLK\_Rx}$	$0.6 \times t_{LxCLK\_Rx}$	ns
t_{DIS}	LxDAT7–0 Input Setup	0.6		ns
t_{DIH}	LxDAT7–0 Input Hold	0.6		ns
Switching Characteristics				
t_{CONNV}	Connectivity Pulse Valid	0	$2.5 \times t_{LxCLK\_Rx}$	ns
t_{CONNOW}	Connectivity Pulse Output Width	$1.5 \times t_{LxCLK\_Rx}$		ns

[1]The link clock ratio (LR) is 2, 3, 4, or 8 as set by the SPD bits in the LCTLx register.
[2]The formula for this parameter applies when LR is 2.
[3]The formula for this parameter applies when LR is 3, 4, or 8.

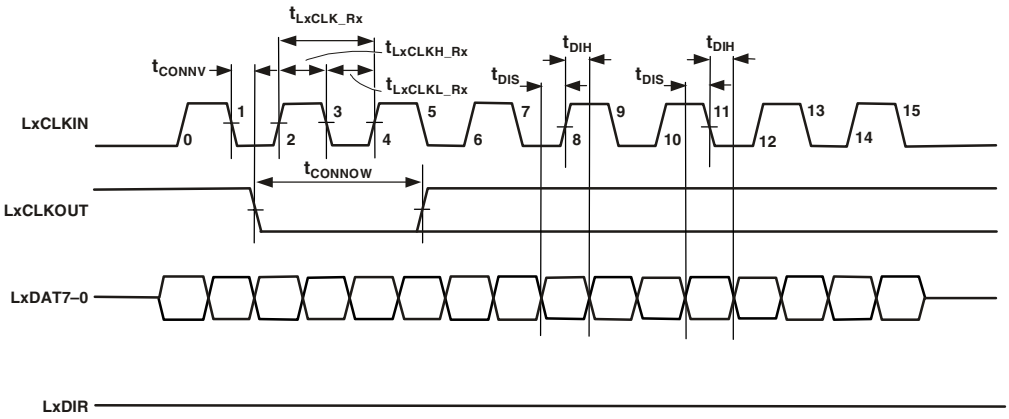


Figure 12. Link Ports—Receive

Table 22. Link Ports—Token Switch, Token Master

Parameter		Min	Max	Unit
Timing Requirements				
t_{REQI}	Token Request Input Width	$5.0 \times t_{LxCLK\_Rx}$		ns
t_{TKRQ}	Token Request from Token Enable[1]		$3.0 \times t_{LxCLK\_Tx}$	ns
Switching Characteristics				
t_{TKENO}	Token Switch Enable Output	$8.0 \times t_{LxCLK\_Tx}$		ns
t_{REQO}	Token Request Output Width[2]	$6.0 \times t_{LxCLK\_Tx}$		ns

[1]For guaranteeing token switch during token enable.
[2]LxCLKOUT shows both possible responses to the token request: [1] a "Token Grant" (LxCLKOUT remains high), and [2] a "Token Regret" (LxCLKOUT goes low).

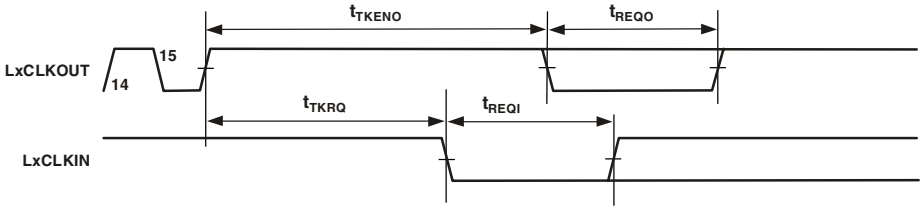


Figure 13. Link Ports—Token Switch, Token Master

Table 23. Link Ports—Token Switch, Token Requester

Parameter		Min	Max	Unit
Timing Requirements				
t_{TKENI}[1]	Token Switch Enable Input	$8.0 \times t_{LxCLK\_Rx}$		ns
Switching Characteristics				
t_{REQO}	Token Request Output Width[2]	$6.0 \times t_{LxCLK\_Rx}$		ns

[1]Required whenever there is a break in transmission.
[2]LxCLKOUT shows both possible responses to the token request: [1] a "Token Grant" (LxCLKOUT remains high), and [2] a "Token Regret" (LxCLKOUT goes low).

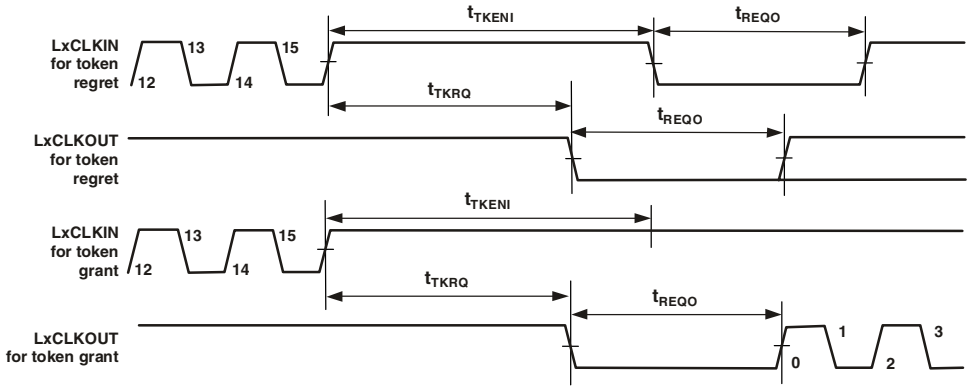


Figure 14. Link Ports—Token Switch, Token Requester

Output Drive Currents

Figure 15 through Figure 22 show typical I–V characteristics for the output drivers of the ADSP-TS101S. The curves in these diagrams represent the current drive capability of the output drivers as a function of output voltage over the range of drive strengths.

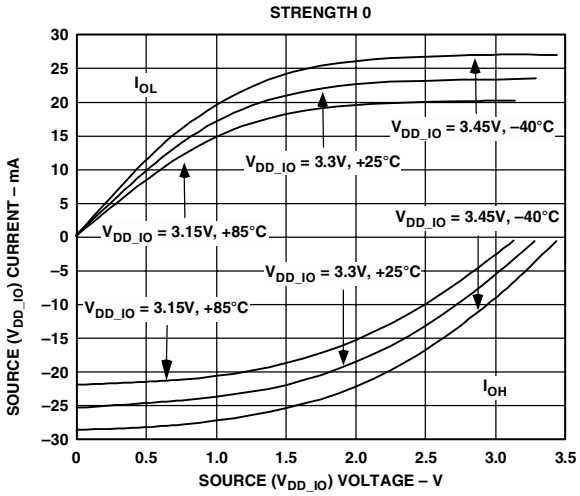


Figure 15. Typical Drive Currents at Strength 0

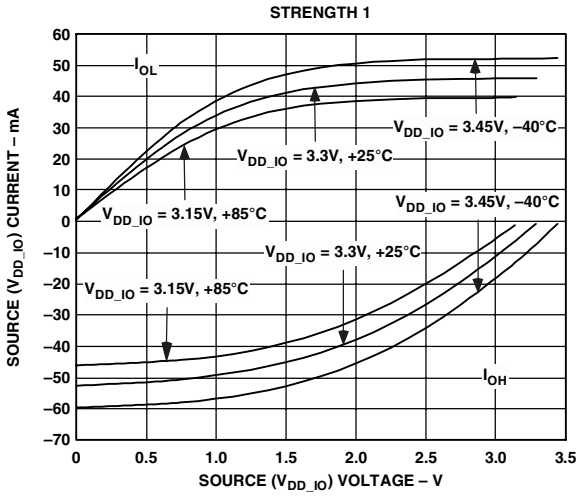


Figure 16. Typical Drive Currents at Strength 1

Power Dissipation

Total power dissipation has two components, one due to internal circuitry (I_{DD}) and one due to the switching of external output drivers (I_{DD_IO}).

For details on internal and external power calculation issues including: power vector definitions, current usage descriptions, and formulas, see the *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.

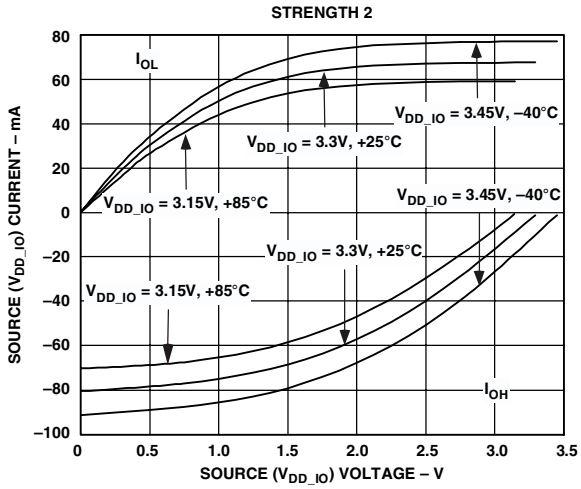


Figure 17. Typical Drive Currents at Strength 2

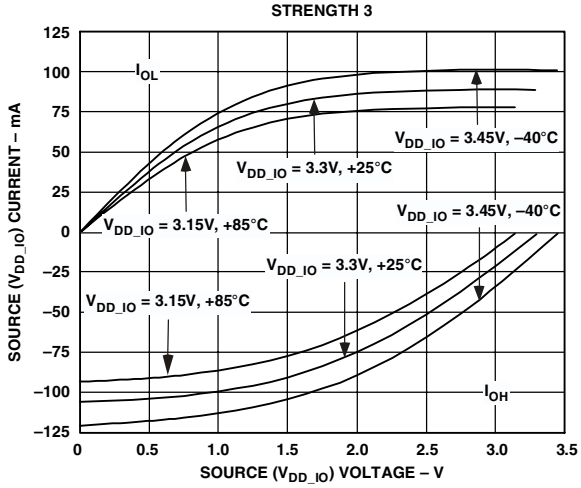


Figure 18. Typical Drive Currents at Strength 3

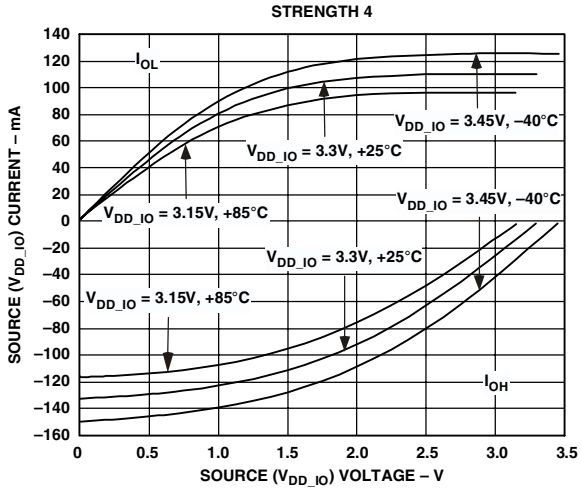


Figure 19. Typical Drive Currents at Strength 4

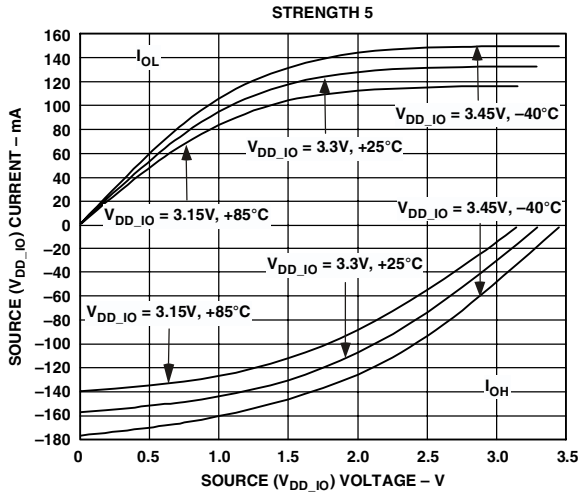


Figure 20. Typical Drive Currents at Strength 5

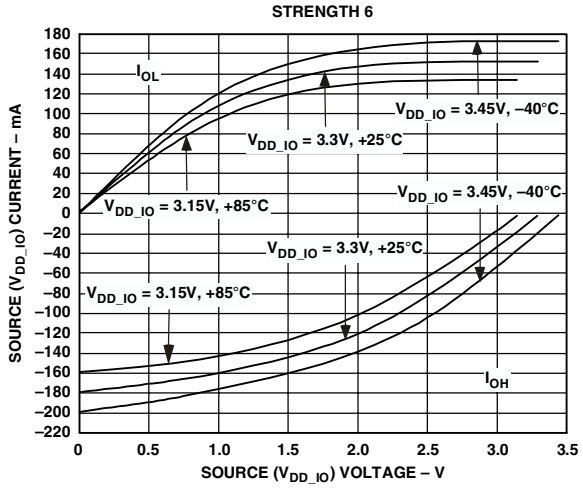


Figure 21. Typical Drive Currents at Strength 6

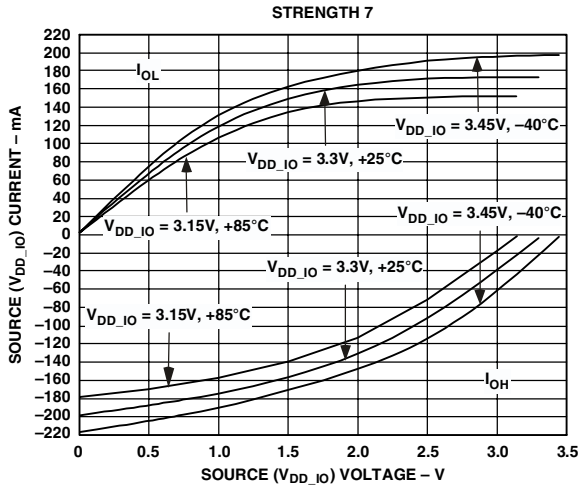


Figure 22. Typical Drive Currents at Strength 7

Test Conditions

The test conditions for timing parameters appearing in Table 19 on Page 23 include output disable time, output enable time, and capacitive loading. The timing specifications for the DSP apply for the voltage reference levels in Figure 23.

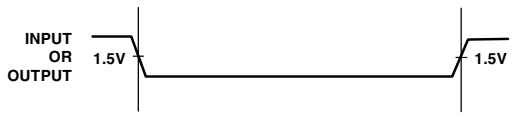


Figure 23. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

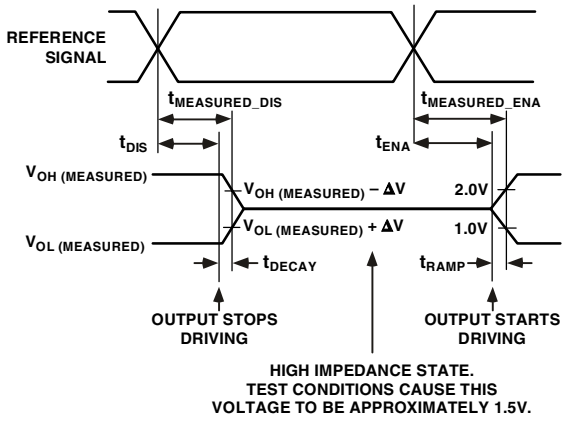


Figure 24. Output Enable/Disable

Output Disable Time

Output pins are considered to be disabled when they stop driving, go into a high impedance state, and start to decay from their output high or low voltage. The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load, C_L and the load current, I_L. This decay time can be approximated by the following equation:

$$t_{DECAY} = \frac{C_L \Delta V}{I_L}$$

The output disable time t_{DIS} is the difference between $t_{MEASURED\_DIS}$ and t_{DECAY} as shown in Figure 24. The time $t_{MEASURED\_DIS}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output high or output low voltage. The t_{DECAY} value is calculated with test loads C_L and I_L, and with ΔV equal to 0.5 V.

Output Enable Time

Output pins are considered to be enabled when they have made a transition from a high impedance state to when they start driving. The time for the voltage on the bus to ramp by ΔV is dependent on the capacitive load, C_L, and the drive current, I_D. This ramp time can be approximated by the following equation:

$$t_{RAMP} = \frac{C_L \Delta V}{I_D}$$

The output enable time t_{ENA} is the difference between $t_{MEASURED\_ENA}$ and t_{RAMP} as shown in Figure 24. The time $t_{MEASURED\_ENA}$ is the interval from when the reference signal switches to when the output voltage ramps ΔV from the measured three-stated output level. The t_{RAMP} value is calculated with test load C_L, drive current I_D, and with ΔV equal to 0.5 V.

Capacitive Loading

Output valid and hold are based on standard capacitive loads: 30 pF on all pins (see Figure 25). 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. Figure 26 through Figure 33 show how output rise time varies with capacitance. Figure 34 graphically shows how output valid varies with load capacitance. (Note that this graph or derating does not apply to output disable delays; see Output Disable Time on Page 29.) The graphs of Figure 26 through Figure 34 may not be linear outside the ranges shown.

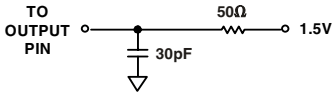


Figure 25. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

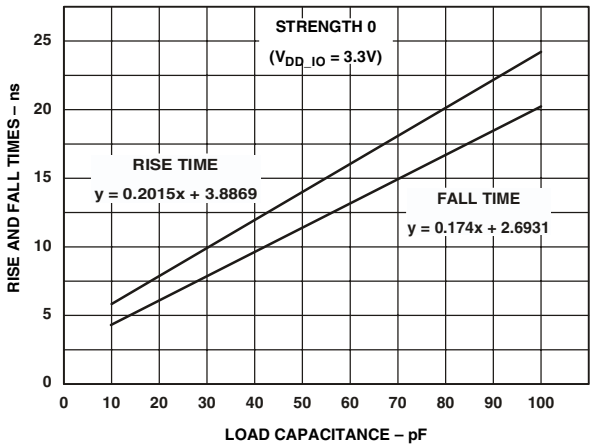


Figure 26. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 0

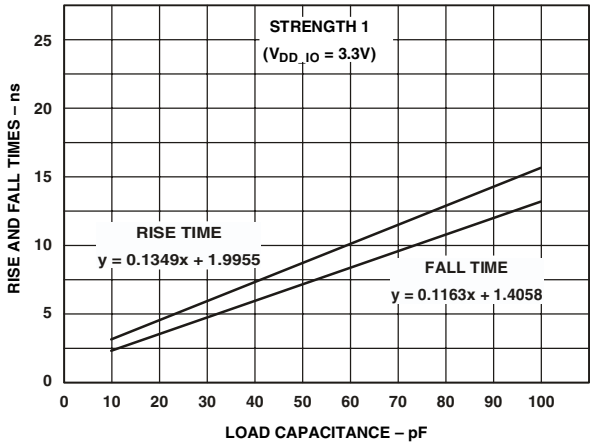


Figure 27. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 1

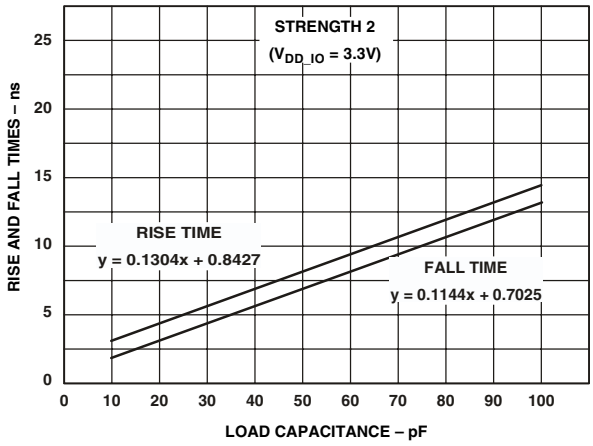


Figure 28. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 2

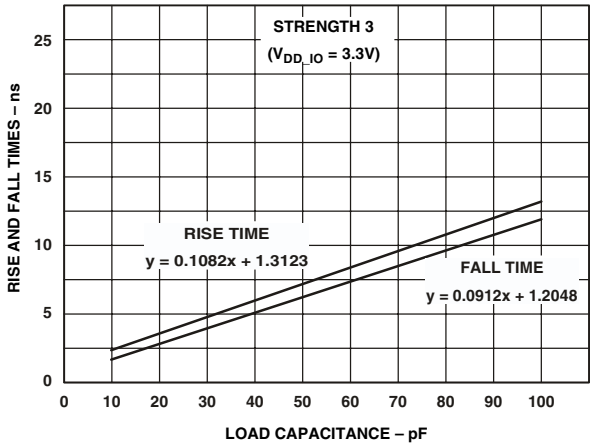


Figure 29. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 3

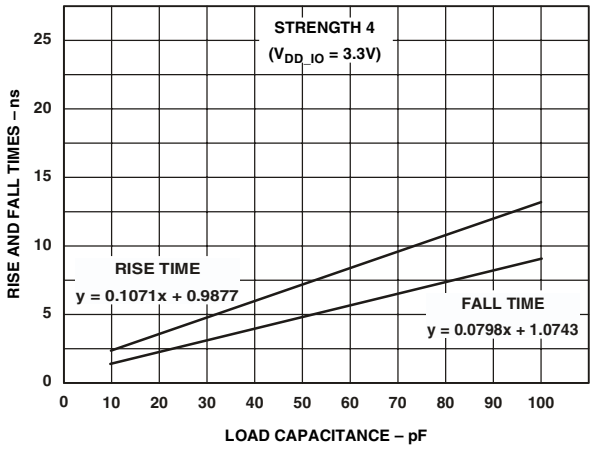


Figure 30. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 4

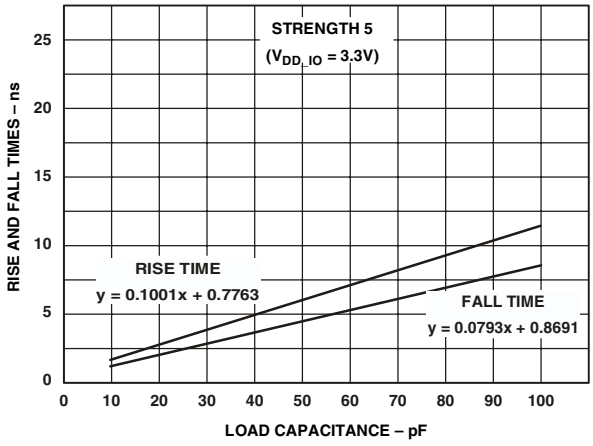


Figure 31. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 5

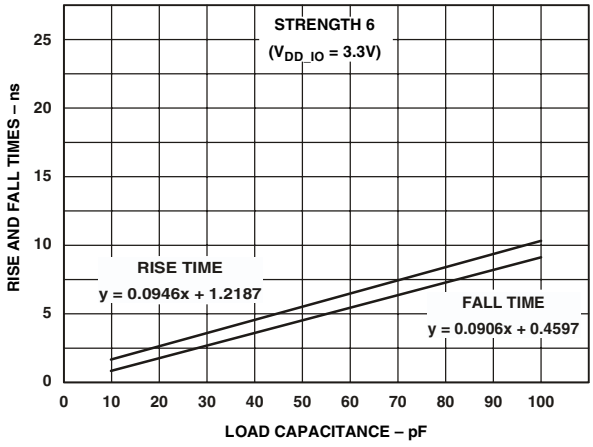


Figure 32. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 6

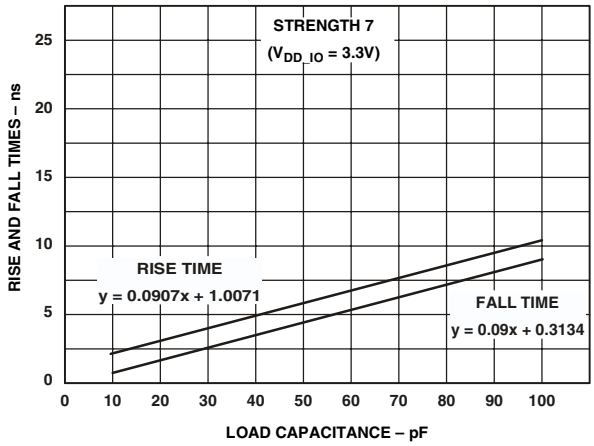


Figure 33. Typical Output Rise and Fall Time (10%–90%, V_{DD_IO} = 3.3 V) vs. Load Capacitance at Strength 7

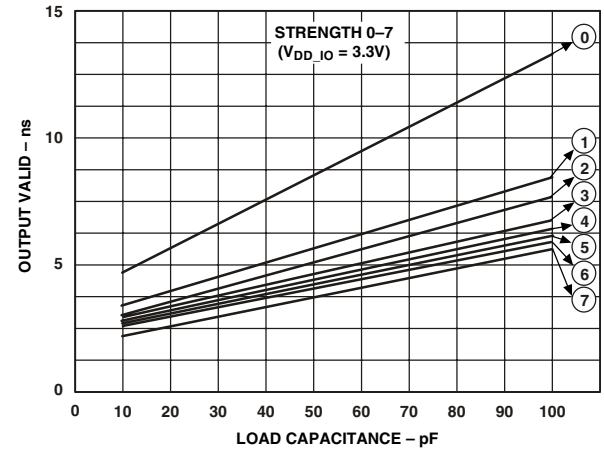


Figure 34. Typical Output Valid (V_{DD_IO} = 3.3 V) vs. Load Capacitance at Max Case Temperature and Strength 0–7[1]

[1]The line equations for the output valid versus load capacitance are:
Strength 0: y = 0.0956x + 3.5662
Strength 1: y = 0.0523x + 3.2144
Strength 2: y = 0.0433x + 3.1319
Strength 3: y = 0.0391x + 2.9675
Strength 4: y = 0.0393x + 2.7653
Strength 5: y = 0.0373x + 2.6515
Strength 6: y = 0.0379x + 2.1206
Strength 7: y = 0.0399x + 1.9080

Environmental Conditions

The ADSP-TS101S is rated for performance over the extended commercial temperature range, T_{CASE} = –40°C to +85°C.

Thermal Characteristics

The ADSP-TS101S is packaged in a 19 mm × 19 mm and 27 mm × 27 mm Plastic Ball Grid Array (PBGA). The ADSP-TS101S is specified for a case temperature (T_{CASE}). To ensure that the T_{CASE} data sheet specification is not exceeded, a heat sink and/or an air flow source may be used. See Table 24 and Table 25 for thermal data.

Table 24. Thermal Characteristics for 19 mm × 19 mm Package

Parameter	Condition	Typical	Unit
θ_{JA}[1]	Airflow[2] = 0 m/s	16.6	°C/W
	Airflow[3] = 1 m/s	14.0	°C/W
	Airflow[3] = 2 m/s	12.9	°C/W
θ_{JC}	–	6.7	°C/W
θ_{JB}	–	5.8	°C/W

[1]The determination of θ_{JA} is system dependent and is based on a number of factors including device power dissipation, package thermal resistance, board thermal characteristics, ambient temperature, and air flow.
[2]Per JEDEC JESD51-2 procedure using a four layer board (compliant with JEDEC JESD51-9).
[3]Per SEMI Test Method G38-87 using a four layer board (compliant with JEDEC JESD51-9).

Table 25. Thermal Characteristics for 27 mm × 27 mm Package

Parameter	Condition	Typical	Unit
θ_{JA}[1]	Airflow[2] = 0 m/s	13.8	°C/W
	Airflow[3] = 1 m/s	11.7	°C/W
	Airflow[3] = 2 m/s	10.8	°C/W
θ_{JC}	–	3.1	°C/W
θ_{JB}	–	5.9	°C/W

[1]The determination of θ_{JA} is system dependent and is based on a number of factors including device power dissipation, package thermal resistance, board thermal characteristics, ambient temperature, and air flow.
[2]Per JEDEC JESD51-2 procedure using a four layer board (compliant with JEDEC JESD51-9).
[3]Per SEMI Test Method G38-87 using a four layer board (compliant with JEDEC JESD51-9).

484-BALL PBGA PIN CONFIGURATIONS

Table 26. 484-Ball (19 mm × 19 mm) PBGA Pin Assignments

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
A1	V_{SS}	B1	DATA21	C1	DATA23	D1	DATA24	E1	DATA25
A2	DATA14	B2	DATA18	C2	DATA17	D2	DATA19	E2	DATA22
A3	DATA11	B3	DATA12	C3	DATA15	D3	DATA16	E3	DATA20
A4	DATA8	B4	DATA13	C4	DATA9	D4	V_{DD_IO}	E4	V_{DD_IO}
A5	DATA4	B5	DATA7	C5	DATA10	D5	V_{DD}	E5	V_{DD}
A6	DATA1	B6	DATA5	C6	DATA6	D6	V_{DD}	E6	V_{DD}
A7	L0DIR	B7	DATA2	C7	DATA3	D7	V_{DD_IO}	E7	V_{DD_IO}
A8	L0CLKIN	B8	NC	C8	DATA0	D8	V_{DD_IO}	E8	V_{DD}
A9	L0DAT6	B9	L0DAT7	C9	L0CLKOUT	D9	V_{DD_IO}	E9	V_{DD}
A10	L0DAT3	B10	L0DAT4	C10	L0DAT5	D10	V_{DD_IO}	E10	V_{DD}
A11	L0DAT1	B11	L0DAT0	C11	L0DAT2	D11	V_{DD_IO}	E11	V_{DD_IO}
A12	V_{SS}	B12	V_{SS}	C12	LCLK_P	D12	V_{DD_IO}	E12	V_{DD}
A13	LCLK_N	B13	V_{DD_A}	C13	V_{SS}	D13	V_{DD_IO}	E13	V_{DD_IO}
A14	V_{SS_A}	B14	V_{SS_A}	C14	V_{DD_A}	D14	V_{DD_IO}	E14	V_{DD}
A15	SCLK_N	B15	V_{SS}	C15	DS0	D15	V_{DD_IO}	E15	V_{DD_IO}
A16	SCLK_P	B16	DS1	C16	DS2	D16	V_{DD}	E16	V_{DD}
A17	CONTROLIMP2	B17	CONTROLIMP0	C17	V_{REF}	D17	V_{DD_IO}	E17	V_{DD_IO}
A18	CONTROLIMP1	B18	$\overline{\text{DMAR2}}$	C18	TRST	D18	V_{DD}	E18	V_{DD_IO}
A19	$\overline{\text{RESET}}$	B19	$\overline{\text{DMAR0}}$	C19	$\overline{\text{DMAR3}}$	D19	V_{DD_IO}	E19	V_{DD_IO}
A20	$\overline{\text{DMAR1}}$	B20	TMS	C20	TCK	D20	TDO	E20	$\overline{\text{BM}}$
A21	$\overline{\text{EMU}}$	B21	TDI	C21	$\overline{\text{IRQ3}}$	D21	$\overline{\text{IRQ2}}$	E21	$\overline{\text{BMS}}$
A22	V_{SS}	B22	$\overline{\text{IRQ1}}$	C22	$\overline{\text{IRQ0}}$	D22	LCLKRAT1	E22	LCLKRAT2
F1	DATA29	G1	L3DAT1	H1	L3DAT2	J1	L3DAT5	K1	L3CLKOUT
F2	DATA30	G2	DATA28	H2	L3DAT0	J2	L3DAT3	K2	L3DAT7
F3	DATA26	G3	DATA27	H3	DATA31	J3	L3DAT4	K3	L3DAT6
F4	V_{DD_IO}	G4	V_{DD}	H4	V_{DD}	J4	V_{DD_IO}	K4	V_{DD_IO}
F5	V_{DD_IO}	G5	V_{DD}	H5	V_{DD}	J5	V_{DD_IO}	K5	V_{DD_IO}
F6	V_{SS}	G6	V_{SS}	H6	V_{SS}	J6	V_{SS}	K6	V_{SS}
F7	V_{SS}	G7	V_{SS}	H7	V_{SS}	J7	V_{SS}	K7	V_{SS}
F8	V_{SS}	G8	V_{SS}	H8	V_{SS}	J8	V_{SS}	K8	V_{SS}
F9	V_{SS}	G9	V_{SS}	H9	V_{SS}	J9	V_{SS}	K9	V_{SS}
F10	V_{SS}	G10	V_{SS}	H10	V_{SS}	J10	V_{SS}	K10	V_{SS}
F11	V_{SS}	G11	V_{SS}	H11	V_{SS}	J11	V_{SS}	K11	V_{SS}
F12	V_{SS}	G12	V_{SS}	H12	V_{SS}	J12	V_{SS}	K12	V_{SS}
F13	V_{SS}	G13	V_{SS}	H13	V_{SS}	J13	V_{SS}	K13	V_{SS}
F14	V_{SS}	G14	V_{SS}	H14	V_{SS}	J14	V_{SS}	K14	V_{SS}
F15	V_{SS}	G15	V_{SS}	H15	V_{SS}	J15	V_{SS}	K15	V_{SS}
F16	V_{SS}	G16	V_{SS}	H16	V_{SS}	J16	V_{SS}	K16	V_{SS}
F17	V_{DD}	G17	V_{SS}	H17	V_{SS}	J17	V_{SS}	K17	V_{SS}
F18	V_{DD_IO}	G18	V_{DD}	H18	V_{DD_IO}	J18	V_{DD}	K18	V_{DD}
F19	V_{DD_IO}	G19	V_{DD_IO}	H19	V_{DD_IO}	J19	V_{DD_IO}	K19	V_{DD_IO}
F20	LCLKRAT0	G20	FLAG3	H20	FLAG1	J20	ID0	K20	$\overline{\text{IOEN}}$
F21	SCLKFREQ	G21	$\overline{\text{BUSLOCK}}$	H21	FLAG2	J21	ID2	K21	$\overline{\text{FLYBY}}$
F22	TMR0E	G22	FLAG0	H22	ID1	J22	$\overline{\text{MSH}}$	K22	$\overline{\text{WRL}}$

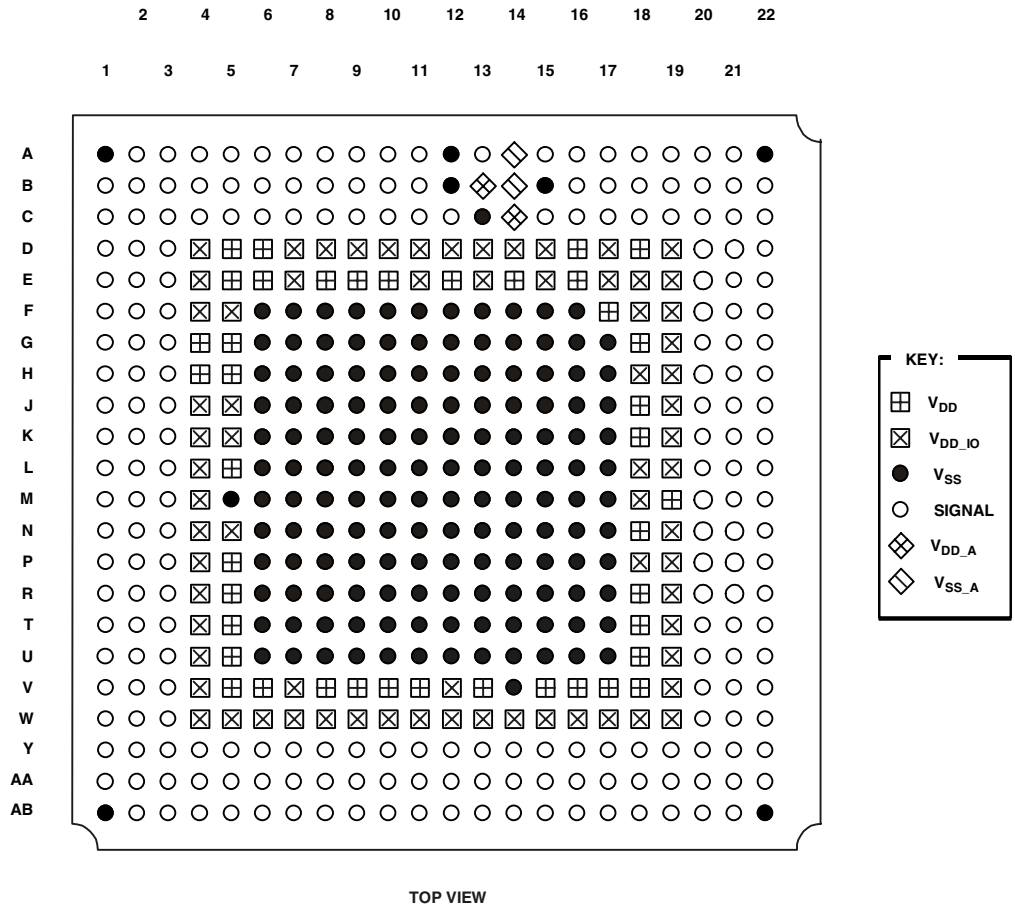
Table 26. 484-Ball (19 mm × 19 mm) PBGA Pin Assignments (continued)

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
L1	L3CLKIN	M1	L1DAT0	N1	L1DAT3	P1	L1DAT4	R1	L1DAT6
L2	NC	M2	L1DAT2	N2	L1DAT5	P2	L1CLKOUT	R2	DATA32
L3	L3DIR	M3	L1DAT1	N3	L1DAT7	P3	L1CLKIN	R3	DATA33
L4	V_{DD_IO}	M4	V_{DD_IO}	N4	V_{DD_IO}	P4	V_{DD_IO}	R4	V_{DD_IO}
L5	V_{DD}	M5	V_{SS}	N5	V_{DD_IO}	P5	V_{DD}	R5	V_{DD}
L6	V_{SS}	M6	V_{SS}	N6	V_{SS}	P6	V_{SS}	R6	V_{SS}
L7	V_{SS}	M7	V_{SS}	N7	V_{SS}	P7	V_{SS}	R7	V_{SS}
L8	V_{SS}	M8	V_{SS}	N8	V_{SS}	P8	V_{SS}	R8	V_{SS}
L9	V_{SS}	M9	V_{SS}	N9	V_{SS}	P9	V_{SS}	R9	V_{SS}
L10	V_{SS}	M10	V_{SS}	N10	V_{SS}	P10	V_{SS}	R10	V_{SS}
L11	V_{SS}	M11	V_{SS}	N11	V_{SS}	P11	V_{SS}	R11	V_{SS}
L12	V_{SS}	M12	V_{SS}	N12	V_{SS}	P12	V_{SS}	R12	V_{SS}
L13	V_{SS}	M13	V_{SS}	N13	V_{SS}	P13	V_{SS}	R13	V_{SS}
L14	V_{SS}	M14	V_{SS}	N14	V_{SS}	P14	V_{SS}	R14	V_{SS}
L15	V_{SS}	M15	V_{SS}	N15	V_{SS}	P15	V_{SS}	R15	V_{SS}
L16	V_{SS}	M16	V_{SS}	N16	V_{SS}	P16	V_{SS}	R16	V_{SS}
L17	V_{SS}	M17	V_{SS}	N17	V_{SS}	P17	V_{SS}	R17	V_{SS}
L18	V_{DD_IO}	M18	V_{DD_IO}	N18	V_{DD}	P18	V_{DD_IO}	R18	V_{DD}
L19	V_{DD_IO}	M19	V_{DD}	N19	V_{DD_IO}	P19	V_{DD_IO}	R19	V_{DD_IO}
L20	$\overline{BRST}$	M20	HDQM	N20	$\overline{SDWE}$	P20	ADDR31	R20	ADDR28
L21	$\overline{WRH}$	M21	$\overline{MS0}$	N21	$\overline{MSSD}$	P21	$\overline{RAS}$	R21	ADDR29
L22	$\overline{RD}$	M22	$\overline{MS1}$	N22	LDQM	P22	SDCKE	R22	$\overline{CAS}$
T1	L1DIR	U1	NC	V1	DATA34	W1	DATA40	Y1	DATA42
T2	DATA36	U2	DATA38	V2	DATA41	W2	DATA43	Y2	DATA45
T3	DATA37	U3	DATA39	V3	DATA35	W3	DATA46	Y3	L2DAT5
T4	V_{DD_IO}	U4	V_{DD_IO}	V4	V_{DD_IO}	W4	V_{DD_IO}	Y4	DATA48
T5	V_{DD}	U5	V_{DD}	V5	V_{DD}	W5	V_{DD_IO}	Y5	DATA52
T6	V_{SS}	U6	V_{SS}	V6	V_{DD}	W6	V_{DD_IO}	Y6	DATA58
T7	V_{SS}	U7	V_{SS}	V7	V_{DD_IO}	W7	V_{DD_IO}	Y7	DATA60
T8	V_{SS}	U8	V_{SS}	V8	V_{DD}	W8	V_{DD_IO}	Y8	DATA63
T9	V_{SS}	U9	V_{SS}	V9	V_{DD}	W9	V_{DD_IO}	Y9	L2DAT4
T10	V_{SS}	U10	V_{SS}	V10	V_{DD}	W10	V_{DD_IO}	Y10	L2CLKOUT
T11	V_{SS}	U11	V_{SS}	V11	V_{DD}	W11	V_{DD_IO}	Y11	NC
T12	V_{SS}	U12	V_{SS}	V12	V_{DD_IO}	W12	V_{DD_IO}	Y12	$\overline{BR4}$
T13	V_{SS}	U13	V_{SS}	V13	V_{DD}	W13	V_{DD_IO}	Y13	ACK
T14	V_{SS}	U14	V_{SS}	V14	V_{SS}	W14	V_{DD_IO}	Y14	$\overline{CPA}$
T15	V_{SS}	U15	V_{SS}	V15	V_{DD}	W15	V_{DD_IO}	Y15	ADDR0
T16	V_{SS}	U16	V_{SS}	V16	V_{DD}	W16	V_{DD_IO}	Y16	$\overline{BR7}$
T17	V_{SS}	U17	V_{SS}	V17	V_{DD}	W17	V_{DD_IO}	Y17	$\overline{HBG}$
T18	V_{DD}	U18	V_{DD}	V18	V_{DD}	W18	V_{DD_IO}	Y18	ADDR1
T19	V_{DD_IO}	U19	V_{DD_IO}	V19	V_{DD_IO}	W19	V_{DD_IO}	Y19	ADDR11
T20	ADDR23	U20	ADDR30	V20	ADDR14	W20	ADDR12	Y20	ADDR21
T21	ADDR25	U21	ADDR22	V21	ADDR19	W21	ADDR17	Y21	ADDR18
T22	ADDR27	U22	ADDR26	V22	ADDR24	W22	ADDR20	Y22	ADDR16

Table 26. 484-Ball (19 mm × 19 mm) PBGA Pin Assignments (continued)

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
AA1	DATA44	AA12	$\overline{BR2}$	AB1	V_{SS}	AB12	$\overline{BR0}$
AA2	DATA50	AA13	$\overline{BR6}$	AB2	DATA53	AB13	$\overline{BR1}$
AA3	DATA47	AA14	$\overline{HBR}$	AB3	DATA55	AB14	$\overline{BR3}$
AA4	DATA49	AA15	$\overline{DPA}$	AB4	DATA56	AB15	$\overline{BR5}$
AA5	DATA51	AA16	ADDR2	AB5	DATA59	AB16	BOFF
AA6	DATA54	AA17	ADDR5	AB6	DATA62	AB17	ADDR3
AA7	DATA57	AA18	ADDR8	AB7	L2DAT1	AB18	ADDR4
AA8	DATA61	AA19	SDA10	AB8	L2DAT2	AB19	ADDR6
AA9	L2DAT0	AA20	ADDR10	AB9	L2DAT6	AB20	ADDR7
AA10	L2DAT3	AA21	ADDR13	AB10	L2CLKIN	AB21	ADDR9
AA11	L2DAT7	AA22	ADDR15	AB11	L2DIR	AB22	V_{SS}

484-Ball PBGA Pin Configurations (Top View, Summary)



TOP VIEW

625-BALL PBGA PIN CONFIGURATIONS

Table 27. 625-Ball (27 mm × 27 mm) PBGA Pin Assignments

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
A1	V_{SS}	B1	V_{SS}	C1	V_{SS}	D1	V_{SS}	E1	DATA23
A2	DATA17	B2	V_{SS}	C2	DATA20	D2	V_{SS}	E2	DATA22
A3	DATA14	B3	DATA16	C3	DATA21	D3	DATA19	E3	V_{SS}
A4	DATA11	B4	DATA13	C4	DATA18	D4	V_{DD_IO}	E4	V_{DD_IO}
A5	DATA9	B5	DATA12	C5	DATA15	D5	V_{DD_IO}	E5	V_{DD_IO}
A6	DATA7	B6	DATA10	C6	DATA8	D6	V_{DD_IO}	E6	V_{DD}
A7	DATA4	B7	DATA5	C7	DATA6	D7	V_{DD_IO}	E7	V_{DD}
A8	DATA1	B8	DATA2	C8	DATA3	D8	V_{DD_IO}	E8	V_{DD_IO}
A9	L0DIR	B9	NC	C9	DATA0	D9	V_{DD_IO}	E9	V_{DD_IO}
A10	L0DAT7	B10	L0CLKOUT	C10	L0CLKIN	D10	V_{DD_IO}	E10	V_{DD}
A11	L0DAT4	B11	L0DAT5	C11	L0DAT6	D11	V_{DD_IO}	E11	V_{DD}
A12	L0DAT1	B12	L0DAT2	C12	L0DAT3	D12	V_{DD_IO}	E12	V_{DD_IO}
A13	LCLK_N	B13	V_{SS}	C13	L0DAT0	D13	V_{DD_IO}	E13	V_{DD_IO}
A14	LCLK_P	B14	V_{SS}	C14	V_{SS_A}	D14	V_{DD_IO}	E14	V_{DD}
A15	V_{DD_A}	B15	V_{SS_A}	C15	V_{DD_A}	D15	V_{DD_IO}	E15	V_{DD}
A16	SCLK_N	B16	SCLK_P	C16	V_{SS}	D16	V_{DD_IO}	E16	V_{DD_IO}
A17	V_{REF}	B17	V_{SS}	C17	DS0	D17	V_{DD_IO}	E17	V_{DD_IO}
A18	DS1	B18	DS2	C18	CONTROLIMP0	D18	V_{DD_IO}	E18	V_{DD}
A19	CONTROLIMP2	B19	CONTROLIMP1	C19	$\overline{\text{DMAR1}}$	D19	V_{DD_IO}	E19	V_{DD}
A20	$\overline{\text{RESET}}$	B20	$\overline{\text{DMAR3}}$	C20	TDI	D20	V_{DD_IO}	E20	V_{DD_IO}
A21	$\overline{\text{DMAR2}}$	B21	$\overline{\text{DMAR0}}$	C21	$\overline{\text{IRQ2}}$	D21	V_{DD_IO}	E21	V_{DD_IO}
A22	$\overline{\text{EMU}}$	B22	$\overline{\text{IRQ3}}$	C22	LCLKRAT0	D22	V_{DD_IO}	E22	V_{DD_IO}
A23	$\overline{\text{TRST}}$	B23	TCK	C23	LCLKRAT1	D23	$\overline{\text{BMS}}$	E23	V_{SS}
A24	TMS	B24	$\overline{\text{IRQ1}}$	C24	$\overline{\text{IRQ0}}$	D24	V_{SS}	E24	SCLKFREQ
A25	V_{SS}	B25	TDO	C25	V_{SS}	D25	V_{SS}	E25	LCLKRAT2
F1	DATA26	G1	DATA29	H1	L3DAT0	J1	L3DAT3	K1	L3DAT6
F2	DATA25	G2	DATA28	H2	DATA31	J2	L3DAT2	K2	L3DAT5
F3	DATA24	G3	DATA27	H3	DATA30	J3	L3DAT1	K3	L3DAT4
F4	V_{DD_IO}	G4	V_{DD_IO}	H4	V_{DD_IO}	J4	V_{DD_IO}	K4	V_{DD_IO}
F5	V_{DD_IO}	G5	V_{DD}	H5	V_{DD}	J5	V_{DD_IO}	K5	V_{DD_IO}
F6	V_{DD}	G6	V_{DD}	H6	V_{DD}	J6	V_{DD}	K6	V_{DD}
F7	V_{DD}	G7	V_{SS}	H7	V_{SS}	J7	V_{SS}	K7	V_{SS}
F8	V_{DD}	G8	V_{SS}	H8	V_{SS}	J8	V_{SS}	K8	V_{SS}
F9	V_{DD}	G9	V_{SS}	H9	V_{SS}	J9	V_{SS}	K9	V_{SS}
F10	V_{DD}	G10	V_{SS}	H10	V_{SS}	J10	V_{SS}	K10	V_{SS}
F11	V_{DD}	G11	V_{SS}	H11	V_{SS}	J11	V_{SS}	K11	V_{SS}
F12	V_{DD}	G12	V_{SS}	H12	V_{SS}	J12	V_{SS}	K12	V_{SS}
F13	V_{DD}	G13	V_{SS}	H13	V_{SS}	J13	V_{SS}	K13	V_{SS}
F14	V_{DD}	G14	V_{SS}	H14	V_{SS}	J14	V_{SS}	K14	V_{SS}
F15	V_{DD}	G15	V_{SS}	H15	V_{SS}	J15	V_{SS}	K15	V_{SS}
F16	V_{DD}	G16	V_{SS}	H16	V_{SS}	J16	V_{SS}	K16	V_{SS}
F17	V_{DD}	G17	V_{SS}	H17	V_{SS}	J17	V_{SS}	K17	V_{SS}
F18	V_{DD}	G18	V_{SS}	H18	V_{SS}	J18	V_{SS}	K18	V_{SS}
F19	V_{DD}	G19	V_{SS}	H19	V_{SS}	J19	V_{SS}	K19	V_{SS}
F20	V_{DD}	G20	V_{DD}	H20	V_{DD}	J20	V_{DD}	K20	V_{DD}
F21	V_{DD}	G21	V_{DD}	H21	V_{DD_IO}	J21	V_{DD_IO}	K21	V_{DD}
F22	V_{DD_IO}	G22	V_{DD_IO}	H22	V_{DD_IO}	J22	V_{DD_IO}	K22	V_{DD_IO}
F23	$\overline{\text{BM}}$	G23	FLAG3	H23	FLAG0	J23	ID0	K23	NC
F24	$\overline{\text{BUSLOCK}}$	G24	FLAG2	H24	ID2	J24	NC	K24	NC
F25	TMR0E	G25	FLAG1	H25	ID1	J25	NC	K25	NC

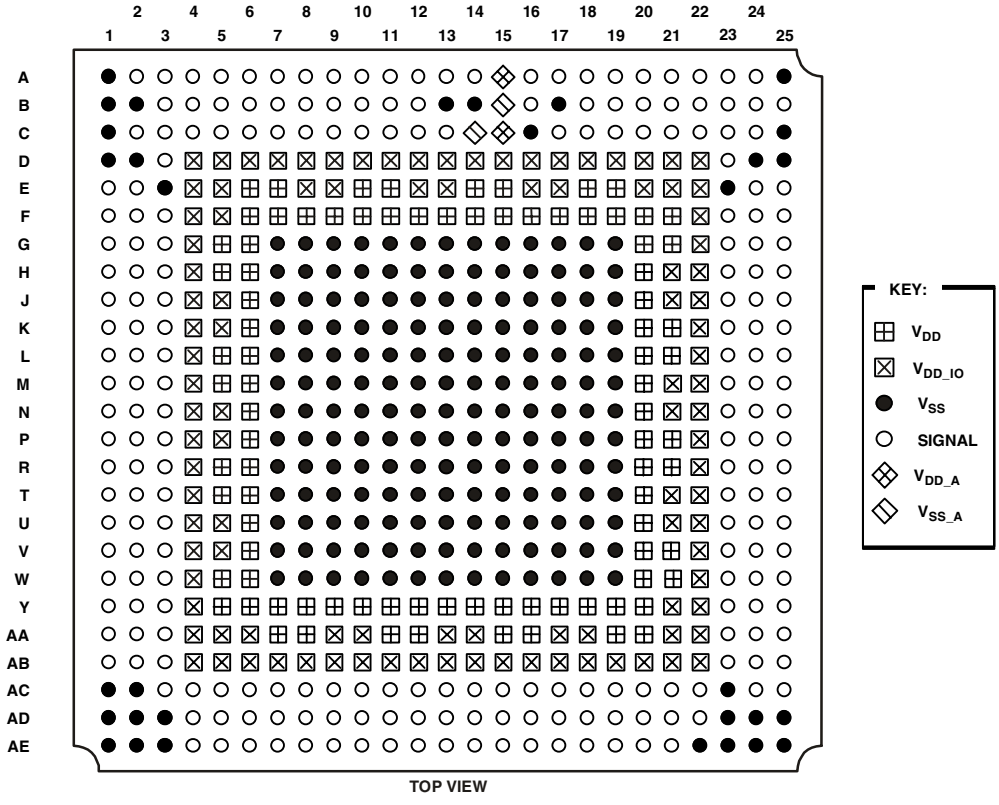
Table 27. 625-Ball (27 mm × 27 mm) PBGA Pin Assignments (continued)

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
L1	L3CLKIN	M1	L1DAT0	N1	L1DAT2	P1	L1DAT5	R1	L1CLKOUT
L2	L3CLKOUT	M2	NC	N2	NC	P2	L1DAT4	R2	L1DAT7
L3	L3DAT7	M3	L3DIR	N3	L1DAT1	P3	L1DAT3	R3	L1DAT6
L4	V_{DD_IO}	M4	V_{DD_IO}	N4	V_{DD_IO}	P4	V_{DD_IO}	R4	V_{DD_IO}
L5	V_{DD}	M5	V_{DD}	N5	V_{DD_IO}	P5	V_{DD_IO}	R5	V_{DD}
L6	V_{DD}	M6	V_{DD}	N6	V_{DD}	P6	V_{DD}	R6	V_{DD}
L7	V_{SS}	M7	V_{SS}	N7	V_{SS}	P7	V_{SS}	R7	V_{SS}
L8	V_{SS}	M8	V_{SS}	N8	V_{SS}	P8	V_{SS}	R8	V_{SS}
L9	V_{SS}	M9	V_{SS}	N9	V_{SS}	P9	V_{SS}	R9	V_{SS}
L10	V_{SS}	M10	V_{SS}	N10	V_{SS}	P10	V_{SS}	R10	V_{SS}
L11	V_{SS}	M11	V_{SS}	N11	V_{SS}	P11	V_{SS}	R11	V_{SS}
L12	V_{SS}	M12	V_{SS}	N12	V_{SS}	P12	V_{SS}	R12	V_{SS}
L13	V_{SS}	M13	V_{SS}	N13	V_{SS}	P13	V_{SS}	R13	V_{SS}
L14	V_{SS}	M14	V_{SS}	N14	V_{SS}	P14	V_{SS}	R14	V_{SS}
L15	V_{SS}	M15	V_{SS}	N15	V_{SS}	P15	V_{SS}	R15	V_{SS}
L16	V_{SS}	M16	V_{SS}	N16	V_{SS}	P16	V_{SS}	R16	V_{SS}
L17	V_{SS}	M17	V_{SS}	N17	V_{SS}	P17	V_{SS}	R17	V_{SS}
L18	V_{SS}	M18	V_{SS}	N18	V_{SS}	P18	V_{SS}	R18	V_{SS}
L19	V_{SS}	M19	V_{SS}	N19	V_{SS}	P19	V_{SS}	R19	V_{SS}
L20	V_{DD}	M20	V_{DD}	N20	V_{DD}	P20	V_{DD}	R20	V_{DD}
L21	V_{DD}	M21	V_{DD_IO}	N21	V_{DD_IO}	P21	V_{DD}	R21	V_{DD}
L22	V_{DD_IO}	M22	V_{DD_IO}	N22	V_{DD_IO}	P22	V_{DD_IO}	R22	V_{DD_IO}
L23	NC	M23	$\overline{IOEN}$	N23	$\overline{WRH}$	P23	$\overline{MS1}$	R23	LDQM
L24	NC	M24	$\overline{MSH}$	N24	$\overline{WRL}$	P24	$\overline{MS0}$	R24	NC
L25	$\overline{FLYBY}$	M25	$\overline{BRST}$	N25	$\overline{RD}$	P25	HDQM	R25	$\overline{MSSD}$
T1	NC	U1	DATA34	V1	DATA37	W1	DATA40	Y1	DATA43
T2	L1DIR	U2	DATA33	V2	DATA36	W2	DATA39	Y2	DATA42
T3	L1CLKIN	U3	DATA32	V3	DATA35	W3	DATA38	Y3	DATA41
T4	V_{DD_IO}	U4	V_{DD_IO}	V4	V_{DD_IO}	W4	V_{DD_IO}	Y4	V_{DD_IO}
T5	V_{DD}	U5	V_{DD_IO}	V5	V_{DD_IO}	W5	V_{DD}	Y5	V_{DD}
T6	V_{DD}	U6	V_{DD}	V6	V_{DD}	W6	V_{DD}	Y6	V_{DD}
T7	V_{SS}	U7	V_{SS}	V7	V_{SS}	W7	V_{SS}	Y7	V_{DD}
T8	V_{SS}	U8	V_{SS}	V8	V_{SS}	W8	V_{SS}	Y8	V_{DD}
T9	V_{SS}	U9	V_{SS}	V9	V_{SS}	W9	V_{SS}	Y9	V_{DD}
T10	V_{SS}	U10	V_{SS}	V10	V_{SS}	W10	V_{SS}	Y10	V_{DD}
T11	V_{SS}	U11	V_{SS}	V11	V_{SS}	W11	V_{SS}	Y11	V_{DD}
T12	V_{SS}	U12	V_{SS}	V12	V_{SS}	W12	V_{SS}	Y12	V_{DD}
T13	V_{SS}	U13	V_{SS}	V13	V_{SS}	W13	V_{SS}	Y13	V_{DD}
T14	V_{SS}	U14	V_{SS}	V14	V_{SS}	W14	V_{SS}	Y14	V_{DD}
T15	V_{SS}	U15	V_{SS}	V15	V_{SS}	W15	V_{SS}	Y15	V_{DD}
T16	V_{SS}	U16	V_{SS}	V16	V_{SS}	W16	V_{SS}	Y16	V_{DD}
T17	V_{SS}	U17	V_{SS}	V17	V_{SS}	W17	V_{SS}	Y17	V_{DD}
T18	V_{SS}	U18	V_{SS}	V18	V_{SS}	W18	V_{SS}	Y18	V_{DD}
T19	V_{SS}	U19	V_{SS}	V19	V_{SS}	W19	V_{SS}	Y19	V_{DD}
T20	V_{DD}	U20	V_{DD}	V20	V_{DD}	W20	V_{DD}	Y20	V_{DD}
T21	V_{DD_IO}	U21	V_{DD_IO}	V21	V_{DD}	W21	V_{DD}	Y21	V_{DD_IO}
T22	V_{DD_IO}	U22	V_{DD_IO}	V22	V_{DD_IO}	W22	V_{DD_IO}	Y22	V_{DD_IO}
T23	SDCKE	U23	$\overline{CAS}$	V23	ADDR31	W23	ADDR28	Y23	ADDR26
T24	NC	U24	NC	V24	ADDR30	W24	NC	Y24	ADDR25
T25	$\overline{SDWE}$	U25	$\overline{RAS}$	V25	ADDR29	W25	ADDR27	Y25	ADDR24

Table 27. 625-Ball (27 mm × 27 mm) PBGA Pin Assignments (continued)

Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic	Pin No.	Mnemonic
AA1	DATA46	AB1	DATA49	AC1	V_{SS}	AD1	V_{SS}	AE1	V_{SS}
AA2	DATA45	AB2	DATA48	AC2	V_{SS}	AD2	V_{SS}	AE2	V_{SS}
AA3	DATA44	AB3	DATA47	AC3	DATA50	AD3	V_{SS}	AE3	V_{SS}
AA4	V_{DD_IO}	AB4	V_{DD_IO}	AC4	DATA51	AD4	DATA52	AE4	DATA53
AA5	V_{DD_IO}	AB5	V_{DD_IO}	AC5	DATA54	AD5	DATA55	AE5	DATA56
AA6	V_{DD_IO}	AB6	V_{DD_IO}	AC6	DATA57	AD6	DATA58	AE6	DATA59
AA7	V_{DD}	AB7	V_{DD_IO}	AC7	DATA60	AD7	DATA61	AE7	DATA62
AA8	V_{DD}	AB8	V_{DD_IO}	AC8	DATA63	AD8	L2DAT0	AE8	L2DAT1
AA9	V_{DD_IO}	AB9	V_{DD_IO}	AC9	L2DAT2	AD9	L2DAT3	AE9	L2DAT4
AA10	V_{DD_IO}	AB10	V_{DD_IO}	AC10	L2DAT5	AD10	L2DAT6	AE10	L2DAT7
AA11	V_{DD}	AB11	V_{DD_IO}	AC11	L2CLKOUT	AD11	L2CLKIN	AE11	L2DIR
AA12	V_{DD}	AB12	V_{DD_IO}	AC12	NC	AD12	$\overline{BR0}$	AE12	$\overline{BR1}$
AA13	V_{DD_IO}	AB13	V_{DD_IO}	AC13	$\overline{BR2}$	AD13	$\overline{BR3}$	AE13	$\overline{BR4}$
AA14	V_{DD_IO}	AB14	V_{DD_IO}	AC14	$\overline{BR5}$	AD14	$\overline{BR6}$	AE14	$\overline{BR7}$
AA15	V_{DD}	AB15	V_{DD_IO}	AC15	ACK	AD15	$\overline{HBR}$	AE15	BOFF
AA16	V_{DD}	AB16	V_{DD_IO}	AC16	$\overline{HBG}$	AD16	$\overline{CPA}$	AE16	$\overline{DPA}$
AA17	V_{DD_IO}	AB17	V_{DD_IO}	AC17	ADDR0	AD17	ADDR1	AE17	ADDR2
AA18	V_{DD_IO}	AB18	V_{DD_IO}	AC18	ADDR3	AD18	ADDR4	AE18	ADDR5
AA19	V_{DD}	AB19	V_{DD_IO}	AC19	ADDR6	AD19	ADDR7	AE19	ADDR8
AA20	V_{DD}	AB20	V_{DD_IO}	AC20	ADDR9	AD20	SDA10	AE20	ADDR10
AA21	V_{DD_IO}	AB21	V_{DD_IO}	AC21	ADDR11	AD21	ADDR12	AE21	ADDR13
AA22	V_{DD_IO}	AB22	V_{DD_IO}	AC22	ADDR14	AD22	ADDR15	AE22	V_{SS}
AA23	ADDR23	AB23	ADDR20	AC23	V_{SS}	AD23	V_{SS}	AE23	V_{SS}
AA24	ADDR22	AB24	ADDR19	AC24	ADDR17	AD24	V_{SS}	AE24	V_{SS}
AA25	ADDR21	AB25	ADDR18	AC25	ADDR16	AD25	V_{SS}	AE25	V_{SS}

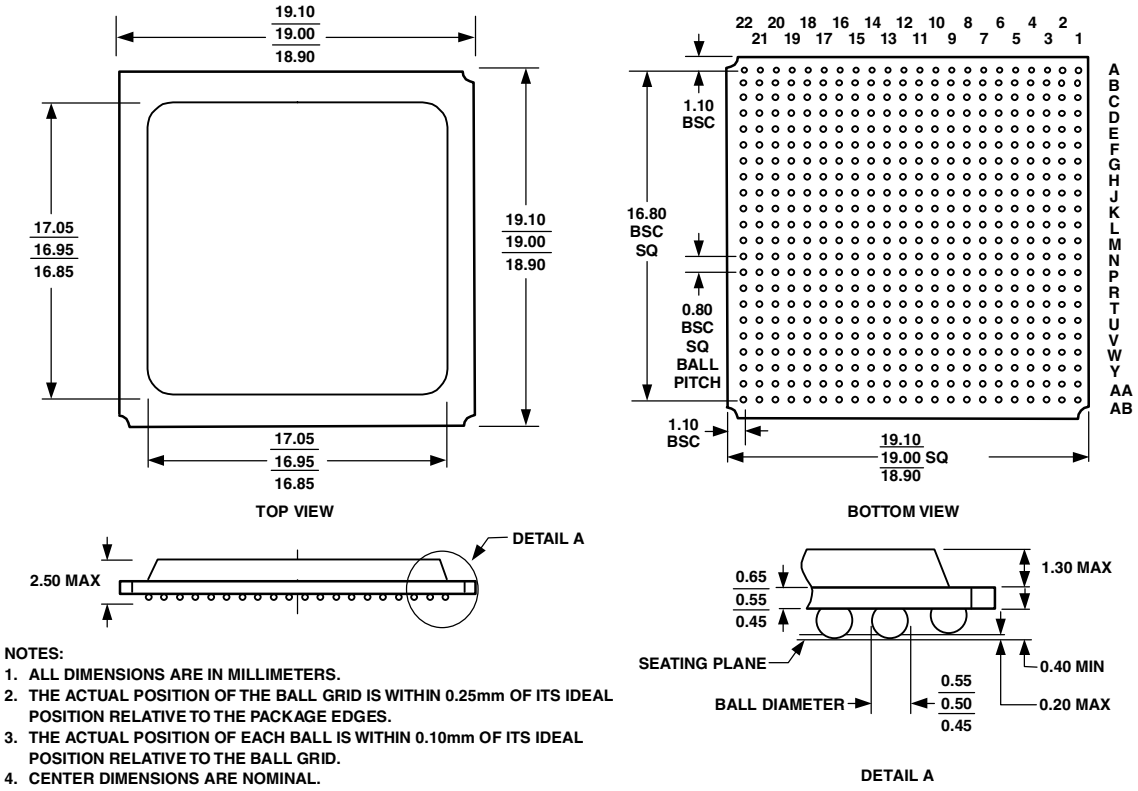
625-Ball PBGA Pin Configurations (Top View, Summary)



OUTLINE DIMENSIONS

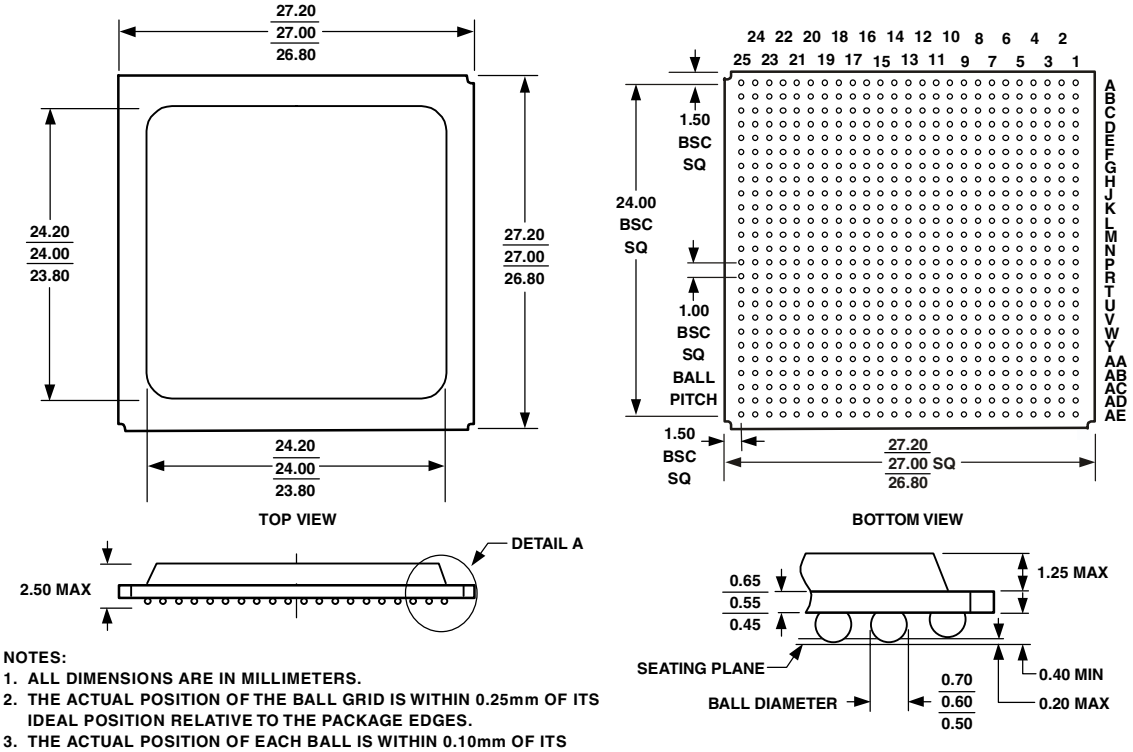
The ADSP-TS101S is available in a 19 mm × 19 mm, 484-ball PBGA package with 22 rows of balls (B-484); the DSP also is available in a 27 mm × 27 mm, 625-ball PBGA package with 25 rows of balls (B-625).

484-Ball PBGA (B-484)



NOTES:
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. THE ACTUAL POSITION OF THE BALL GRID IS WITHIN 0.25mm OF ITS IDEAL POSITION RELATIVE TO THE PACKAGE EDGES.
3. THE ACTUAL POSITION OF EACH BALL IS WITHIN 0.10mm OF ITS IDEAL POSITION RELATIVE TO THE BALL GRID.
4. CENTER DIMENSIONS ARE NOMINAL.

625-Ball PBGA (B-625)



NOTES:
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. THE ACTUAL POSITION OF THE BALL GRID IS WITHIN 0.25mm OF ITS IDEAL POSITION RELATIVE TO THE PACKAGE EDGES.
3. THE ACTUAL POSITION OF EACH BALL IS WITHIN 0.10mm OF ITS IDEAL POSITION RELATIVE TO THE BALL GRID.
4. CENTER DIMENSIONS ARE NOMINAL.
5. THIS PACKAGE COMPLIES WITH THE JEDEC MS-034 SPECIFICATION, BUT USES TIGHTER TOLERANCES THAN THE MAXIMUMS ALLOWED IN THAT SPECIFICATION.

ORDERING GUIDE

Part Number[1, 2, 3, 4]	Temperature Range (Case)	Core Clock (CCLK) Rate[5]	On-chip SRAM	Operating Voltage	Package
ADSP-TS101SAB1-000	–40°C to +85°C	250 MHz	6M Bit	1.2 V_{DD} 3.3 V_{DD_IO}	(B-625)[6]
ADSP-TS101SAB2-000	–40°C to +85°C	250 MHz	6M Bit	1.2 V_{DD} 3.3 V_{DD_IO}	(B-484)[7]

[1]S indicates 1.2 and 3.3 V supplies.
[2]A indicates –40°C to +85°C temperature.
[3]B = Plastic Ball Grid Array (PBGA) package.
[4]1-000 indicates B-625 package and 250 MHz speed grade, and 2-000 indicates B-484 package and 250 MHz speed grade.
[5]The instruction rate runs at the internal DSP clock (CCLK) rate.
[6]The B-625 package measures 27 mm × 27 mm.
[7]The B-484 package measures 19 mm × 19 mm.

C03164–0–9/02(0)

PRINTED IN U.S.A.