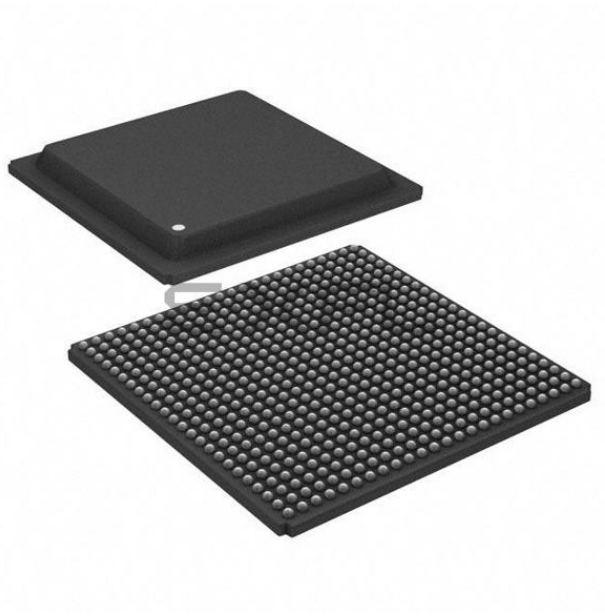

Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Obsolete
Type	Fixed/Floating Point
Interface	Host Interface, Link Port, Multi-Processor
Clock Rate	500MHz
Non-Volatile Memory	External
On-Chip RAM	3MB
Voltage - I/O	2.50V
Voltage - Core	1.05V
Operating Temperature	-40°C ~ 85°C (TC)
Mounting Type	Surface Mount
Package / Case	576-BBGA
Supplier Device Package	576-BGA-ED (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-ts201sybp-050

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

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

PROGRAM SEQUENCER

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

- A fully interruptible programming model with flexible programming in assembly and C/C++ languages; handles hardware interrupts with high throughput and no aborted instruction cycles
- A 10-cycle instruction pipeline—four-cycle fetch pipe and six-cycle execution pipe—computation results available two cycles after operands are available
- Supply of instruction fetch memory addresses; the sequencer's instruction alignment buffer (IAB) caches up to five fetched instruction lines waiting to execute; the program sequencer extracts an instruction line from the IAB and distributes it to the appropriate core component for execution
- Management of program structures and program flow determined according to JUMP, CALL, RTI, RTS instructions, loop structures, conditions, interrupts, and software exceptions
- Branch prediction and a 128-entry branch target buffer (BTB) to reduce branch delays for efficient execution of conditional and unconditional branch instructions and zero-overhead looping; correctly predicted branches occur with zero overhead cycles, overcoming the five-to-nine stage branch penalty
- Compact code without the requirement to align code in memory; the IAB handles alignment

Interrupt Controller

The DSP supports nested and nonnested interrupts. Each interrupt type has a register in the interrupt vector table. Also, each has a bit in both the interrupt latch register and the interrupt mask register. All interrupts are fixed as either level-sensitive or edge-sensitive, except the $\overline{\text{IRQ3–0}}$ 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:

- CLU instructions for communications infrastructure to govern trellis decoding (for example, Viterbi and Turbo decoders) and despreading via complex correlations
- Algebraic assembly language syntax
- Direct support for all DSP, imaging, and video arithmetic types
- Eliminates toggling DSP hardware modes because modes are supported as options (for example, rounding, saturation, and others) within instructions
- Branch prediction encoded in instruction; enables zero-overhead loops
- Parallelism encoded in instruction line
- Conditional execution optional for all instructions
- User-defined partitioning between program and data memory

DSP MEMORY

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

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.

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

The six internal memory blocks connect to the four 128-bit wide internal buses through a crossbar connection, enabling the DSP to perform four memory transfers in the same cycle. The DSP's internal bus architecture provides a total memory bandwidth of

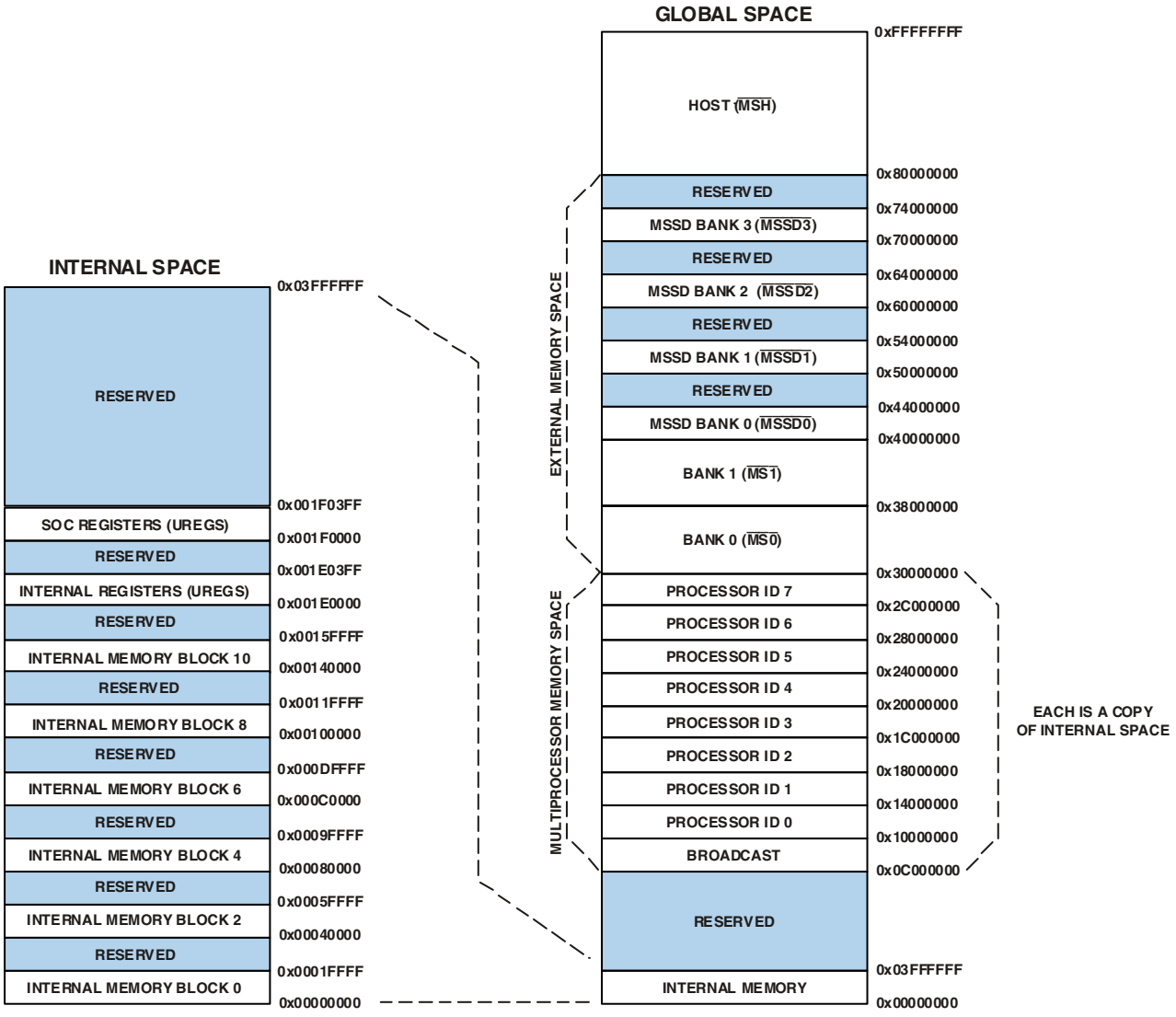

Figure 3. ADSP-TS201S Memory Map

33.6G bytes per second, enabling the core and I/O to access eight 32-bit data-words and four 32-bit instructions each cycle. The DSP's flexible memory structure enables:

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

EXTERNAL PORT (OFF-CHIP MEMORY/PERIPHERALS INTERFACE)

The ADSP-TS201S processor's external port provides the DSP's interface to off-chip memory and peripherals. The 4G word address space is included in the DSP's unified address space.

The separate on-chip buses—four 128-bit data buses and four 32-bit address buses—are multiplexed at the SOC interface and transferred to the external port over the SOC bus to create an external system bus transaction. The external system bus provides a single 64-bit data bus and a single 32-bit address bus. The external port supports data transfer rates of 1G byte per second over the external bus.

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

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

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

Host Interface

The ADSP-TS201S processor provides an easy and configurable interface between its external bus and host processors through the external port (see Figure 4). To accommodate a variety of host processors, the host interface supports pipelined or slow protocols for ADSP-TS201S processor access of the host as slave or pipelined for host access of the ADSP-TS201S processor as slave. Each protocol has programmable transmission parameters, such as idle cycles, pipe depth, and internal wait cycles.

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

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

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

Multiprocessor Interface

The ADSP-TS201S processor offers powerful features tailored to multiprocessing DSP systems through the external port and link ports (see Figure 4). This multiprocessing capability provides the highest bandwidth for interprocessor communication, including:

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

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

The external port supports a unified address space (see Figure 3) that enables direct interprocessor accesses of each ADSP-TS201S processor's internal memory and registers. The DSP's on-chip distributed bus arbitration logic provides simple, glueless connection for systems containing up to eight ADSP-TS201S processors and a host processor. Bus arbitration has a rotating priority. Bus lock supports indivisible read-modify-write sequences for semaphores. A bus fairness feature prevents one DSP from holding the external bus too long.

The DSP's four link ports provide a second path for interprocessor communications with throughput of 4G bytes per second. The cluster bus provides 1G byte per second throughput—with a total of 4.8G bytes per second interprocessor bandwidth (limited by SOC bandwidth).

SDRAM Controller

The SDRAM controller controls the ADSP-TS201S processor's transfers of data to and from external synchronous DRAM (SDRAM) at a throughput of 32 bits 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, 256M bit, and 512M bit. The DSP supports directly a maximum of four banks of 64M words × 32 bits of SDRAM. The SDRAM interface is mapped in external memory in each DSP's unified memory map.

EPROM Interface

The ADSP-TS201S processor can be configured to boot from an external 8-bit EPROM at reset through the external port. An automatic process (which follows reset) loads a program from the EPROM into internal memory. This process uses 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-TS201S processor's on-chip DMA controller, with 14 DMA channels, provides zero-overhead data transfers without processor intervention. The DMA controller operates independently and invisibly to the DSP's core, enabling DMA operations to occur while the DSP's core continues to execute program instructions.

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

- External port block transfers. Four dedicated bidirectional DMA channels transfer blocks of data between the DSP's internal memory and any external memory or memory-mapped peripheral on the external bus. These transfers support master mode and handshake mode protocols.
- Link port transfers. Eight dedicated DMA channels (four transmit and four receive) transfer quad-word data only between link ports and between a link port and internal or

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. It also is used for downloading components from the Web, dropping them into the application, and publishing 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 use in a color-coded graphical form, easily move code and data to different areas of the DSP or external memory with a drag of the mouse, and examine runtime stack and heap usage. The expert linker is fully compatible with existing linker definition file (LDF), allowing the developer to move between the graphical and textual environments.

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

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

EVALUATION KIT

Analog Devices offers a range of EZ-KIT Lite®† evaluation platforms to use as a cost-effective method to learn more about developing or prototyping applications with Analog Devices processors, platforms, and software tools. Each EZ-KIT Lite includes an evaluation board along with an evaluation suite of the VisualDSP++ development and debugging environment with the C/C++ compiler, assembler, and linker. Also included are sample application programs, power supply, and a USB cable. All evaluation versions of the software tools are limited for use only with the EZ-KIT Lite product.

The USB controller on the EZ-KIT Lite board connects the board to the USB port of the user's PC, enabling the VisualDSP++ evaluation suite to emulate the on-board processor in-circuit. This permits the customer to download, execute, and debug programs for the EZ-KIT Lite system. It also allows in-circuit programming of the on-board flash device to store user-specific boot code, enabling the board to run as a standalone unit, without being connected to the PC.

With a full version of VisualDSP++ installed (sold separately), engineers can develop software for the EZ-KIT Lite or any custom-defined system. Connecting one of Analog Devices JTAG emulators to the EZ-KIT Lite board enables high speed, nonintrusive emulation.

DESIGNING AN EMULATOR-COMPATIBLE DSP BOARD (TARGET)

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

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

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

ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-TS201S processor's architecture and functionality. For detailed information on the ADSP-TS201S processor's core architecture and instruction set, see the *ADSP-TS201 TigerSHARC Processor Hardware Reference* and the *ADSP-TS201 TigerSHARC Processor Programming Reference*. For detailed information on the development tools for this processor, see the *VisualDSP++ User's Guide for TigerSHARC Processors*.

† EZ-Kit Lite is a registered trademark of Analog Devices, Inc.

Table 8. Pin Definitions—External Port SDRAM Controller

Signal	Type	Term	Description
$\overline{\text{MSSD3–0}}$	I/O/T (pu_0)	nc	Memory Select SDRAM. $\overline{\text{MSSD0}}$, $\overline{\text{MSSD1}}$, $\overline{\text{MSSD2}}$, or $\overline{\text{MSSD3}}$ is asserted whenever the DSP accesses SDRAM memory space. $\overline{\text{MSSD3–0}}$ are decoded memory address pins that are asserted whenever the DSP issues an SDRAM command cycle (access to ADDR31:30 = 0b01—except reserved spaces shown in Figure 3 on Page 6). In a multiprocessor system, the master DSP drives $\overline{\text{MSSD3–0}}$.
$\overline{\text{RAS}}$	I/O/T (pu_0)	nc	Row Address Select. When sampled low, $\overline{\text{RAS}}$ indicates that a row address is valid in a read or write of SDRAM. In other SDRAM accesses, it defines the type of operation to execute according to SDRAM specification.
$\overline{\text{CAS}}$	I/O/T (pu_0)	nc	Column Address Select. When sampled low, $\overline{\text{CAS}}$ indicates that a column address is valid in a read or write of SDRAM. In other SDRAM accesses, it defines the type of operation to execute according to the SDRAM specification.
LDQM	O/T (pu_0)	nc	Low Word SDRAM Data Mask. When sampled high, three-states the SDRAM DQ buffers. LDQM is valid on SDRAM transactions when $\overline{\text{CAS}}$ is asserted, and 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	O/T (pu_0)	nc	High Word SDRAM Data Mask. When sampled high, three-states the SDRAM DQ buffers. HDQM is valid on SDRAM transactions when $\overline{\text{CAS}}$ is asserted, and inactive on read transactions. On write transactions, HDQM is active when accessing an even address in word accesses or when memory is configured for a 32-bit bus to disable the write of the high word.
SDA10	O/T (pu_0)	nc	SDRAM Address Bit 10. Separate A10 signals enable SDRAM refresh operation while the DSP executes non-SDRAM transactions.
SDCKE	I/O/T (pu_m/ pd_m)	nc	SDRAM Clock Enable. Activates the SDRAM clock for SDRAM self-refresh or suspend modes. A slave DSP in a multiprocessor system does not have the pull-up or pull-down. A master DSP (or ID = 0 in a single processor system) has a 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 pull-down before granting the bus to the host.
$\overline{\text{SDWE}}$	I/O/T (pu_0)	nc	SDRAM Write Enable. When sampled low while $\overline{\text{CAS}}$ is active, $\overline{\text{SDWE}}$ indicates an SDRAM write access. When sampled high while $\overline{\text{CAS}}$ is active, $\overline{\text{SDWE}}$ indicates an SDRAM read access. In other SDRAM accesses, $\overline{\text{SDWE}}$ defines the type of operation to execute according to SDRAM specification.

I = input; **A** = asynchronous; **O** = output; **OD** = open-drain output; **T** = three-state; **P** = power supply; **G** = ground; **pd** = internal pull-down 5 kΩ; **pu** = internal pull-up 5 kΩ; **pd_0** = internal pull-down 5 kΩ on DSP ID = 0; **pu_0** = internal pull-up 5 kΩ on DSP ID = 0; **pu_od_0** = internal pull-up 500 Ω on DSP ID = 0; **pd_m** = internal pull-down 5 kΩ on DSP bus master; **pu_m** = internal pull-up 5 kΩ on DSP bus master; **pu_ad** = internal pull-up 40 kΩ. For more pull-down and pull-up information, see Electrical Characteristics on Page 22.

Term (termination of unused pins) column symbols: epd = external pull-down approximately 5 kΩ to V_{SS}; epu = external pull-up approximately 5 kΩ to V_{DD_IO}, nc = not connected; na = not applicable (always used); V_{DD_IO} = connect directly to V_{DD_IO}; V_{SS} = connect directly to V_{SS}

Table 9. Pin Definitions—JTAG Port

Signal	Type	Term	Description
$\overline{\text{EMU}}$	O/OD	nc[1]	Emulation. Connected to the DSP's JTAG emulator target board connector only.
TCK	I	epd or epu[1]	Test Clock (JTAG). Provides an asynchronous clock for JTAG scan.
TDI	I (pu_ad)	nc[1]	Test Data Input (JTAG). A serial data input of the scan path.
TDO	O/T	nc[1]	Test Data Output (JTAG). A serial data output of the scan path.
TMS	I (pu_ad)	nc[1]	Test Mode Select (JTAG). Used to control the test state machine.
$\overline{\text{TRST}}$	I/A (pu_ad)	na	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 9.

I = input; **A** = asynchronous; **O** = output; **OD** = open-drain output; **T** = three-state; **P** = power supply; **G** = ground; **pd** = internal pull-down 5 kΩ; **pu** = internal pull-up 5 kΩ; **pd_0** = internal pull-down 5 kΩ on DSP ID = 0; **pu_0** = internal pull-up 5 kΩ on DSP ID = 0; **pu_od_0** = internal pull-up 500 Ω on DSP ID = 0; **pd_m** = internal pull-down 5 kΩ on DSP bus master; **pu_m** = internal pull-up 5 kΩ on DSP bus master; **pu_ad** = internal pull-up 40 kΩ. For more pull-down and pull-up information, see Electrical Characteristics on Page 22.

Term (termination of unused pins) column symbols: epd = external pull-down approximately 5 kΩ to V_{SS}; epu = external pull-up approximately 5 kΩ to V_{DD_IO}, nc = not connected; na = not applicable (always used); V_{DD_IO} = connect directly to V_{DD_IO}; V_{SS} = connect directly to V_{SS}

[1] See the reference on Page 11 to the JTAG emulation technical reference EE-68.

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

Signal	Type	Term	Description
FLAG3–0	I/O/A (pu)	nc	FLAG pins. Bidirectional input/output pins can be used as program conditions. Each pin can be configured individually for input or for output. FLAG3–0 are inputs after power-up and reset.
$\overline{\text{IRQ3–0}}$	I/A (pu)	nc	Interrupt Request. When asserted, the DSP generates an interrupt. Each of the $\overline{\text{IRQ3–0}}$ pins can be independently set for edge-triggered or level-sensitive operation. After reset, these pins are disabled unless the $\overline{\text{IRQ3–0}}$ strap option and interrupt vectors are initialized for booting.
TMR0E	O	na	Timer 0 expires. This output pulses whenever timer 0 expires. At reset, this is a strap pin. For more information, see Table 16 on Page 20.

I = input; **A** = asynchronous; **O** = output; **OD** = open-drain output; **T** = three-state; **P** = power supply; **G** = ground; **pd** = internal pull-down 5 kΩ; **pu** = internal pull-up 5 kΩ; **pd_0** = internal pull-down 5 kΩ on DSP ID = 0; **pu_0** = internal pull-up 5 kΩ on DSP ID = 0; **pu_od_0** = internal pull-up 500 Ω on DSP ID = 0; **pd_m** = internal pull-down 5 kΩ on DSP bus master; **pu_m** = internal pull-up 5 kΩ on DSP bus master; **pu_ad** = internal pull-up 40 kΩ. For more pull-down and pull-up information, see Electrical Characteristics on Page 22.

Term (termination of unused pins) column symbols: epd = external pull-down approximately 5 kΩ to V_{SS}; epu = external pull-up approximately 5 kΩ to V_{DD_IO}, nc = not connected; na = not applicable (always used); V_{DD_IO} = connect directly to V_{DD_IO}; V_{SS} = connect directly to V_{SS}

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 internal pull-up or pull-down for the default value. If a strap pin is not connected to an overdriving external pull-up, pull-down, or logic load, the DSP samples the default value during reset. If strap pins are connected to logic inputs, a stronger external pull-up or 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 or pull-down. Table 16 lists and describes each of the DSP's strap pins.

Table 16. Pin Definitions—I/O Strap Pins

Signal	Type (at Reset)	On Pin …	Description
EBOOT	I (pd_0)	$\overline{\text{BMS}}$	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	I (pd)	$\overline{\text{BM}}$	Interrupt Enable. 0 = disable and set $\overline{\text{IRQ3–0}}$ interrupts to edge-sensitive after reset (default) 1 = enable and set $\overline{\text{IRQ3–0}}$ interrupts to level-sensitive immediately after reset
LINK_DWIDTH	I (pd)	TMR0E	Link Port Input Default Data Width. 0 = 1-bit (default) 1 = 4-bit
SYS_REG_WE	I (pd_0)	$\overline{\text{BUSLOCK}}$	SYSCON and SDRCON Write Enable. 0 = one-time writable after reset (default) 1 = always writable
TM1	I (pu)	$\overline{\text{L1BCMPO}}$	Test Mode 1. Do not overdrive default value during reset.
TM2	I (pu)	$\overline{\text{L2BCMPO}}$	Test Mode 2. Do not overdrive default value during reset.
TM3	I (pu)	$\overline{\text{L3BCMPO}}$	Test Mode 3. Do not overdrive default value during reset.

I = input; **A** = asynchronous; **O** = output; **OD** = open-drain output; **T** = three-state; **P** = power supply; **G** = ground; **pd** = internal pull-down 5 kΩ; **pu** = internal pull-up 5 kΩ; **pd_0** = internal pull-down 5 kΩ on DSP ID = 0; **pu_0** = internal pull-up 5 kΩ on DSP ID = 0; **pu_od_0** = internal pull-up 500 Ω on DSP ID = 0; **pd_m** = internal pull-down 5 kΩ on DSP bus master; **pu_m** = internal pull-up 5 kΩ on DSP bus master; **pu_ad** = internal pull-up 40 kΩ. For more pull-down and pull-up information, see Electrical Characteristics on Page 22.

When default configuration is used, no external resistor is needed on the strap pins. To apply other configurations, a 500 Ω resistor connected to V_{DD_IO} is required. If providing external pull-downs, do not strap these pins directly to V_{SS}; the strap pins require 500 Ω resistor straps.

All strap pins are sampled on the rising edge of $\overline{\text{RST\_IN}}$ (deassertion edge). Each pin latches the strapped pin state (state of the strap pin at the rising edge of $\overline{\text{RST\_IN}}$). Shortly after deassertion of $\overline{\text{RST\_IN}}$, these pins are reconfigured to their normal functionality.

These strap pins have an internal pull-down resistor, pull-up resistor, or no-resistor (three-state) on each pin. The resistor type, which is connected to the I/O pad, depends on whether $\overline{\text{RST\_IN}}$ is active (low) or if $\overline{\text{RST\_IN}}$ is deasserted (high). Table 17 shows the resistors that are enabled during active reset and during normal operation.

Table 17. Strap Pin Internal Resistors—Active Reset ($\overline{\text{RST\_IN}}$ = 0) vs. Normal Operation ($\overline{\text{RST\_IN}}$ = 1)

Pin	$\overline{\text{RST\_IN}}$ = 0	$\overline{\text{RST\_IN}}$ = 1
$\overline{\text{BMS}}$	(pd_0)	(pu_0)
$\overline{\text{BM}}$	(pd)	Driven
TMR0E	(pd)	Driven
$\overline{\text{BUSLOCK}}$	(pd_0)	(pu_0)
$\overline{\text{L1BCMPO}}$	(pu)	Driven
$\overline{\text{L2BCMPO}}$	(pu)	Driven
$\overline{\text{L3BCMPO}}$	(pu)	Driven

pd = internal pull-down 5 kΩ; **pu** = internal pull-up 5 kΩ;
pd_0 = internal pull-down 5 kΩ on DSP ID = 0;
pu_0 = internal pull-up 5 kΩ on DSP ID = 0

TIMING SPECIFICATIONS

With the exception of $\overline{\text{DMAR3–0}}$, $\overline{\text{IRQ3–0}}$, TMR0E, and FLAG3–0 (input only) pins, all ac timing for the ADSP-TS201S processor 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-TS201S processor has few calculated (formula-based) values. For information on ac timing, see General AC Timing. For information on link port transfer timing, see Link Port Low Voltage, Differential-Signal (LVDS) Electrical Characteristics, and Timing on Page 30.

General AC Timing

Timing is measured on signals when they cross the 1.25 V level as described in Figure 15 on Page 29. All delays (in nanoseconds) are measured between the point that the first signal reaches 1.25 V and the point that the second signal reaches 1.25 V.

The general ac timing data appears in Table 22 and Table 29. All ac specifications are measured with the load specified in Figure 36 on Page 38, 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 37 on Page 38 through Figure 44 on Page 39 (Rise and Fall Time vs. Load Capacitance) and Figure 45 on Page 39 (Output Valid vs. Load Capacitance and Drive Strength).

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

Table 21. AC Asynchronous Signal Specifications

Name	Description	Pulse Width Low (Min)	Pulse Width High (Min)
$\overline{\text{IRQ3–0}}$[1]	Interrupt Request	$2 \times t_{SCLK}$ ns	$2 \times t_{SCLK}$ ns
$\overline{\text{DMAR3–0}}$[1]	DMA Request	$2 \times t_{SCLK}$ ns	$2 \times t_{SCLK}$ ns
FLAG3–0[2]	FLAG3–0 Input	$2 \times t_{SCLK}$ ns	$2 \times t_{SCLK}$ ns
TMR0E[3]	Timer 0 Expired	$4 \times t_{SCLK}$ ns	—

[1] These input pins have Schmitt triggers and therefore do not need to be synchronized to a clock reference.
[2] For output specifications on FLAG3–0 pins, see Table 29.
[3] This pin is a strap option. During reset, an internal resistor pulls the pin low.

Table 22. Reference Clocks—Core Clock (CCLK) Cycle Time

		Grade = 060 (600 MHz)		Grade = 050 (500 MHz)		
Parameter	Description	Min	Max	Min	Max	Unit
t_{CCLK}[1]	Core Clock Cycle Time	1.67	12.5	2.0	12.5	ns

[1] CCLK is the internal processor clock or instruction cycle time. The period of this clock is equal to the system clock period (t_{SCLK}) divided by the system clock ratio (SCLKRAT2–0). For information on available part numbers for different internal processor clock rates, see the Ordering Guide on Page 46.

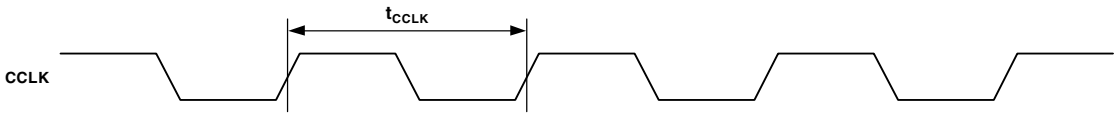

Figure 9. Reference Clocks—Core Clock (CCLK) Cycle Time

Table 29. AC Signal Specifications

(All values in this table are in nanoseconds.)

Name	Description	Input Setup (Min)	Input Hold (Min)	Output Valid (Max)	Output Hold (Min)	Output Enable (Min)[1]	Output Disable (Max)[1]	Reference Clock
ADDR31–0	External Address Bus	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
DATA63–0	External Data Bus	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{MSH}}$	Memory Select HOST Line	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{MSSD3–0}}$	Memory Select SDRAM Lines	1.5	0.5	4.0	1.0	1.0	2.0	SCLK
$\overline{\text{MS1–0}}$	Memory Select for Static Blocks	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{RD}}$	Memory Read	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{WRL}}$	Write Low Word	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{WRH}}$	Write High Word	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
ACK	Acknowledge for Data High to Low	1.5	0.5	3.6	1.0	1.15	2.0	SCLK
	Acknowledge for Data Low to High	1.5	0.5	4.2	0.9	1.15	2.0	SCLK
SDCKE	SDRAM Clock Enable	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{RAS}}$	Row Address Select	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{CAS}}$	Column Address Select	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{SDWE}}$	SDRAM Write Enable	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
LDQM	Low Word SDRAM Data Mask	—	—	4.0	1.0	1.15	2.0	SCLK
HDQM	High Word SDRAM Data Mask	—	—	4.0	1.0	1.15	2.0	SCLK
SDA10	SDRAM ADDR10	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{HBR}}$	Host Bus Request	1.5	0.5	—	—	—	—	SCLK
$\overline{\text{HBG}}$	Host Bus Grant	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{BOFF}}$	Back Off Request	1.5	0.5	—	—	—	—	SCLK
$\overline{\text{BUSLOCK}}$	Bus Lock	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{BRST}}$	Burst Pin	1.5	0.5	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{BR7–0}}$	Multiprocessing Bus Request Pins	1.5	0.5	4.0	1.0	—	—	SCLK
$\overline{\text{BM}}$	Bus Master Debug Aid Only	—	—	4.0	1.0	—	—	SCLK
$\overline{\text{IORD}}$	I/O Read Pin	—	—	4.0	1.0	1.0	2.0	SCLK
$\overline{\text{IOWR}}$	I/O Write Pin	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{IOEN}}$	I/O Enable Pin	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{CPA}}$	Core Priority Access High to Low	1.5	0.5	4.0	1.0	0.75	2.0	SCLK
	Core Priority Access Low to High	1.5	0.5	29.5	2.0	0.75	2.0	SCLK
$\overline{\text{DPA}}$	DMA Priority Access High to Low	1.5	0.5	4.0	1.0	0.75	2.0	SCLK
	DMA Priority Access Low to High	1.5	0.5	29.5	2.0	0.75	2.0	SCLK
$\overline{\text{BMS}}$	Boot Memory Select	—	—	4.0	1.0	1.15	2.0	SCLK
FLAG3–0[2]	FLAG Pins	—	—	4.0	1.0	1.15	2.0	SCLK
$\overline{\text{RST\_IN}}$[3, 4]	Global Reset Pin	1.5	2.5	—	—	—	—	SCLK[5]
TMS	Test Mode Select (JTAG)	1.5	0.5	—	—	—	—	TCK
TDI	Test Data Input (JTAG)	1.5	0.5	—	—	—	—	TCK
TDO	Test Data Output (JTAG)	—	—	4.0	1.0	0.75	2.0	TCK[6]
$\overline{\text{TRST}}$[3, 4]	Test Reset (JTAG)	1.5	0.5	—	—	—	—	TCK
$\overline{\text{EMU}}$[7]	Emulation High to Low	—	—	5.5	2.0	1.15	4.0	TCK or SCLK
ID2–0[8]	Static Pins—Must Be Constant	—	—	—	—	—	—	—
CONTROLIMP1–0[8]	Static Pins—Must Be Constant	—	—	—	—	—	—	—

Table 29. AC Signal Specifications (Continued)

(All values in this table are in nanoseconds.)

Name	Description	Input Setup (Min)	Input Hold (Min)	Output Valid (Max)	Output Hold (Min)	Output Enable (Min)[1]	Output Disable (Max)[1]	Reference Clock
DS2–0[8]	Static Pins—Must Be Constant	—	—	—	—	—	—	—
SCLKRAT2–0[8]	Static Pins—Must Be Constant	—	—	—	—	—	—	—
ENEDREG	Static Pins—Must Be Connected to V_{SS}	—	—	—	—	—	—	—
STRAP SYS[9, 10]	Strap Pins	1.5	0.5	—	—	—	—	SCLK
JTAG SYS[11, 12]	JTAG System Pins	+2.5	+10.0	+12.0	–1.0	—	—	TCK

[1] The external port protocols employ bus IDLE cycles for bus mastership transitions as well as slave access boundary crossings to avoid any potential bus contention. The apparent driver overlap, due to output disables being larger than output enables, is not actual.
[2] For input specifications on FLAG3–0 pins, see Table 21.
[3] These input pins are asynchronous and therefore do not need to be synchronized to a clock reference.
[4] For additional requirement details, see Reset and Booting on Page 9.
[5] $\overline{\text{RST\_IN}}$ clock reference is the falling edge of SCLK.
[6] TDO output clock reference is the falling edge of TCK.
[7] Reference clock depends on function.
[8] These pins may change only during reset; recommend connecting it to V_{DD_IO}/V_{SS}.
[9] STRAP pins include: $\overline{\text{BMS}}$, $\overline{\text{BM}}$, $\overline{\text{BUSLOCK}}$, TMR0E, $\overline{\text{L1BCMPO}}$, $\overline{\text{L2BCMPO}}$, and $\overline{\text{L3BCMPO}}$.
[10] Specifications applicable during reset only.
[11] JTAG system pins include: $\overline{\text{RST\_IN}}$, $\overline{\text{RST\_OUT}}$, $\overline{\text{POR\_IN}}$, $\overline{\text{IRQ3–0}}$, $\overline{\text{DMAR3–0}}$, $\overline{\text{HBR}}$, $\overline{\text{BOFF}}$, $\overline{\text{MS1–0}}$, $\overline{\text{MSH}}$, SDCKE, LDQM, HDQM, $\overline{\text{BMS}}$, $\overline{\text{IOWR}}$, $\overline{\text{IORD}}$, $\overline{\text{BM}}$, $\overline{\text{EMU}}$, SDA10, $\overline{\text{IOEN}}$, $\overline{\text{BUSLOCK}}$, TMR0E, DATA63–0, ADDR31–0, $\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{BRST}}$, $\overline{\text{MSSD3–0}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{SDWE}}$, $\overline{\text{HBG}}$, $\overline{\text{BR7–0}}$, FLAG3–0, L0DATOP3–0, L0DATON3–0, L1DATOP3–0, L1DATON3–0, L2DATOP3–0, L2DATON3–0, L3DATOP3–0, L3DATON3–0, L0CLKOUTP, L0CLKOUTN, L1CLKOUTP, L1CLKOUTN, L2CLKOUTP, L2CLKOUTN, L3CLKOUTP, L3CLKOUTN, L0ACKI, L1ACKI, L2ACKI, L3ACKI, L0DATIP3–0, L0DATIN3–0, L1DATIP3–0, L1DATIN3–0, L2DATIP3–0, L2DATIN3–0, L3DATIP3–0, L3DATIN3–0, L0CLKINP, L0CLKINN, L1CLKINP, L1CLKINN, L2CLKINP, L2CLKINN, L3CLKINP, L3CLKINN, L0ACKO, L1ACKO, L2ACKO, L3ACKO, ACK, $\overline{\text{CPA}}$, $\overline{\text{DPA}}$, $\overline{\text{L0BCMPO}}$, $\overline{\text{L1BCMPO}}$, $\overline{\text{L2BCMPO}}$, $\overline{\text{L3BCMPO}}$, $\overline{\text{L0BCMPI}}$, $\overline{\text{L1BCMPI}}$, $\overline{\text{L2BCMPI}}$, $\overline{\text{L3BCMPI}}$, ID2–0, CTRL_IMPD1–0, SCLKRAT2–0, DS2–0, ENEDREG.
[12] JTAG system output timing clock reference is the falling edge of TCK.

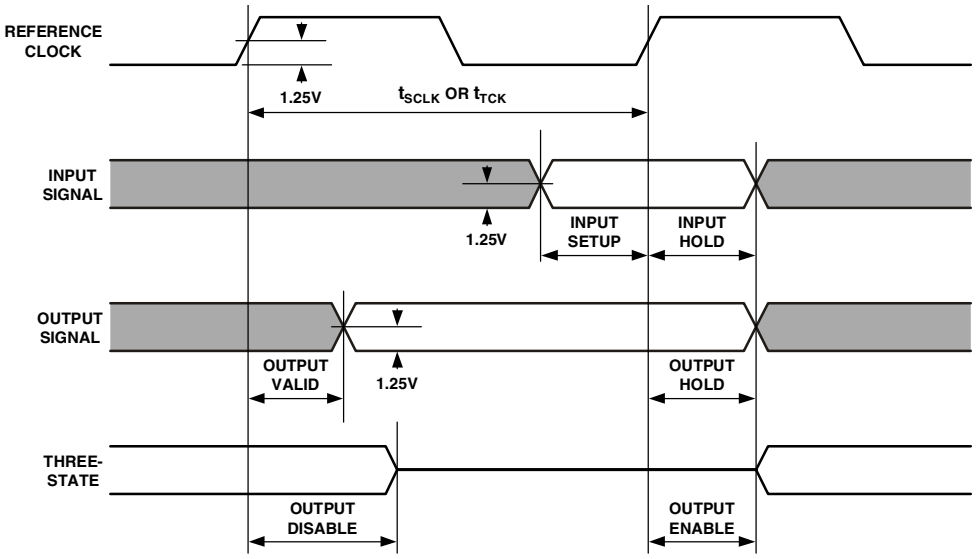

Figure 15. General AC Parameters Timing

Link Port Low Voltage, Differential-Signal (LVDS) Electrical Characteristics, and Timing

Table 30 and Table 31 with Figure 16 provide the electrical characteristics for the LVDS link ports. The LVDS link port signal definitions represent all differential signals with a V_{OD} = 0 V level and use signal naming without N (negative) and P (positive) suffixes (see Figure 17).

Table 30. Link Port LVDS Transmit Electrical Characteristics

Parameter	Description	Test Conditions	Min	Max	Unit
V_{OH}	Output Voltage High, V_{O_P} or V_{O_N}	R_L = 100 Ω		1.85	V
V_{OL}	Output Voltage Low, V_{O_P} or V_{O_N}	R_L = 100 Ω	0.92		V
$\|V_{OD}\|$	Output Differential Voltage	R_L = 100 Ω	300	650	mV
I_{OS}	Short-Circuit Output Current	V_{O_P} or V_{O_N} = 0 V		+5/– 55	mA
		V_{OD} = 0 V		±10	mA
V_{OCM}	Common-Mode Output Voltage		1.20	1.50	V

Table 31. Link Port LVDS Receive Electrical Characteristics

Parameter	Description	Test Conditions	Min	Max	Unit
$\|V_{ID}\|$	Differential Input Voltage	$t_{LDIS}/t_{LDIH} \geq 0.20$ ns	250	850	mV
		$t_{LDIS}/t_{LDIH} \geq 0.25$ ns	217	850	mV
		$t_{LDIS}/t_{LDIH} \geq 0.30$ ns	206	850	mV
		$t_{LDIS}/t_{LDIH} \geq 0.35$ ns	195	850	mV
V_{ICM}	Common-Mode Input Voltage		0.6	1.57	V

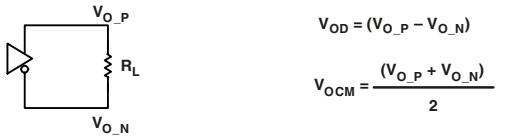

Figure 16. Link Ports—Transmit Electrical Characteristics

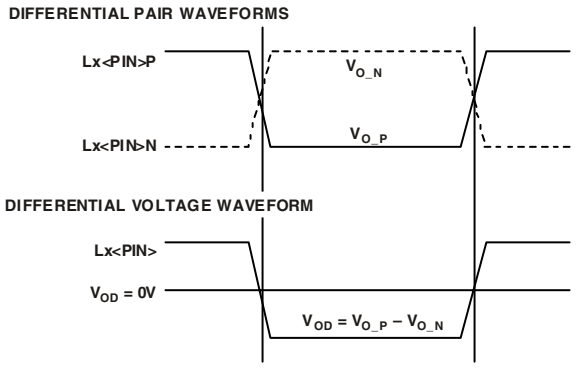

Figure 17. Link Ports—Signals Definition

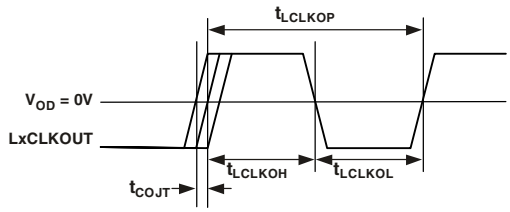

Figure 18. Link Ports—Output Clock

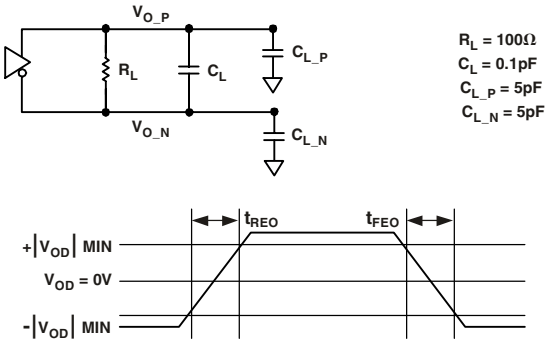

Figure 19. Link Ports—Differential Output Signals Transition Time

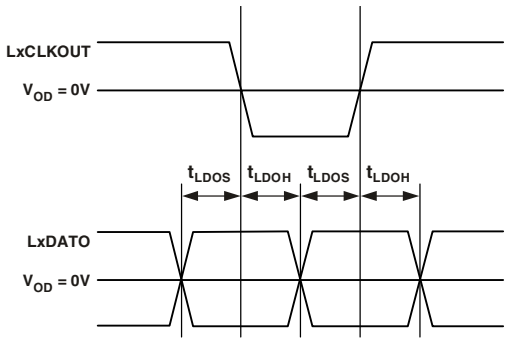

Figure 20. Link Ports—Data Output Setup and Hold[1]

[1] These parameters are valid for both clock edges.

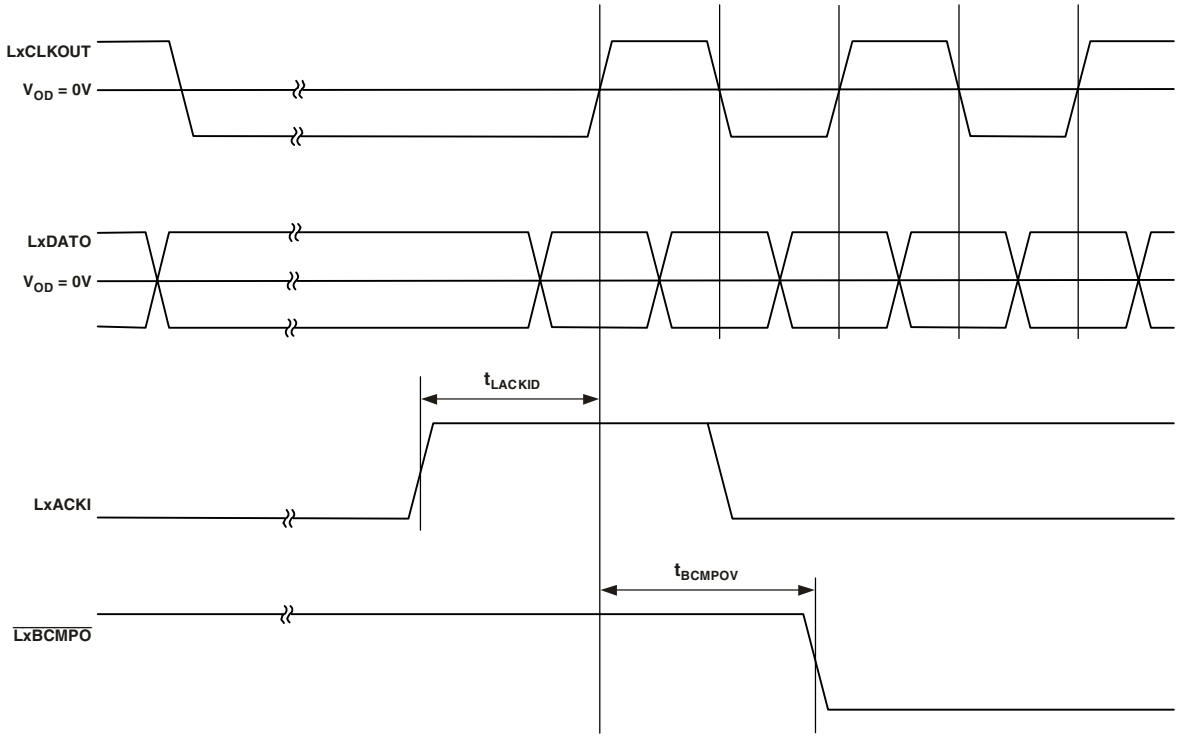

Figure 21. Link Ports—Transmission Start

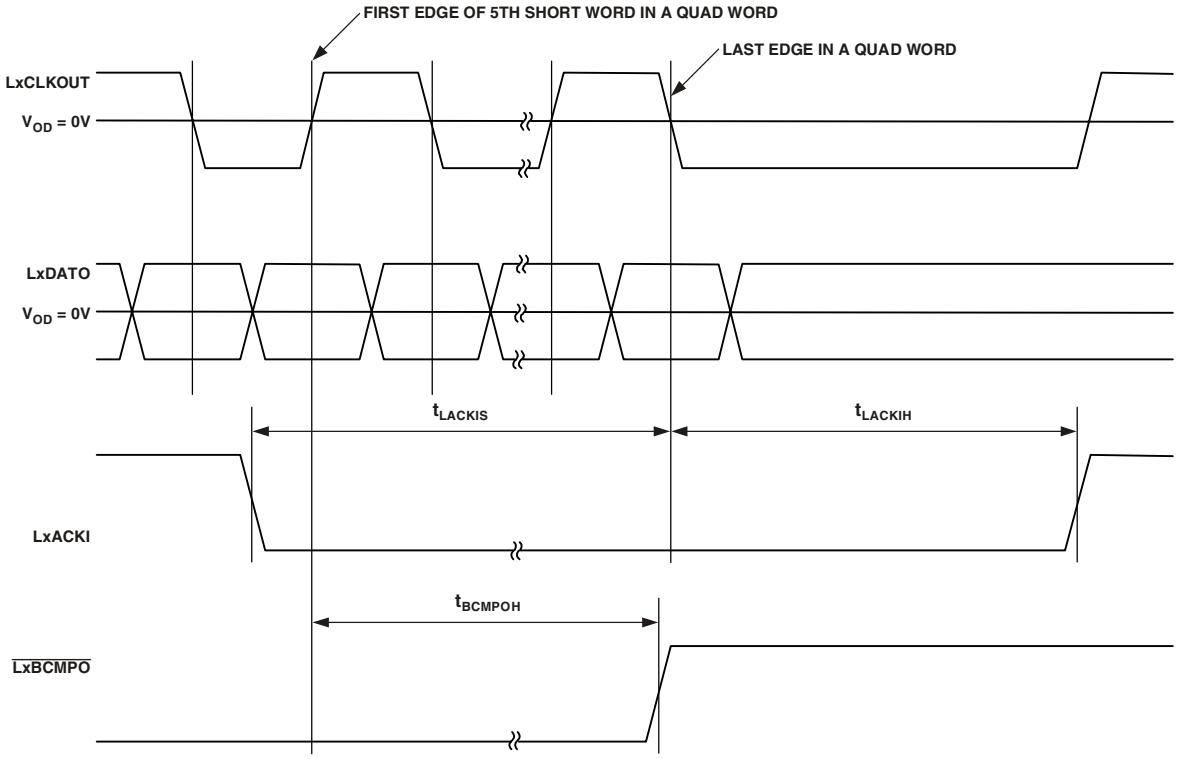

Figure 22. Link Ports—Transmission End and Stops

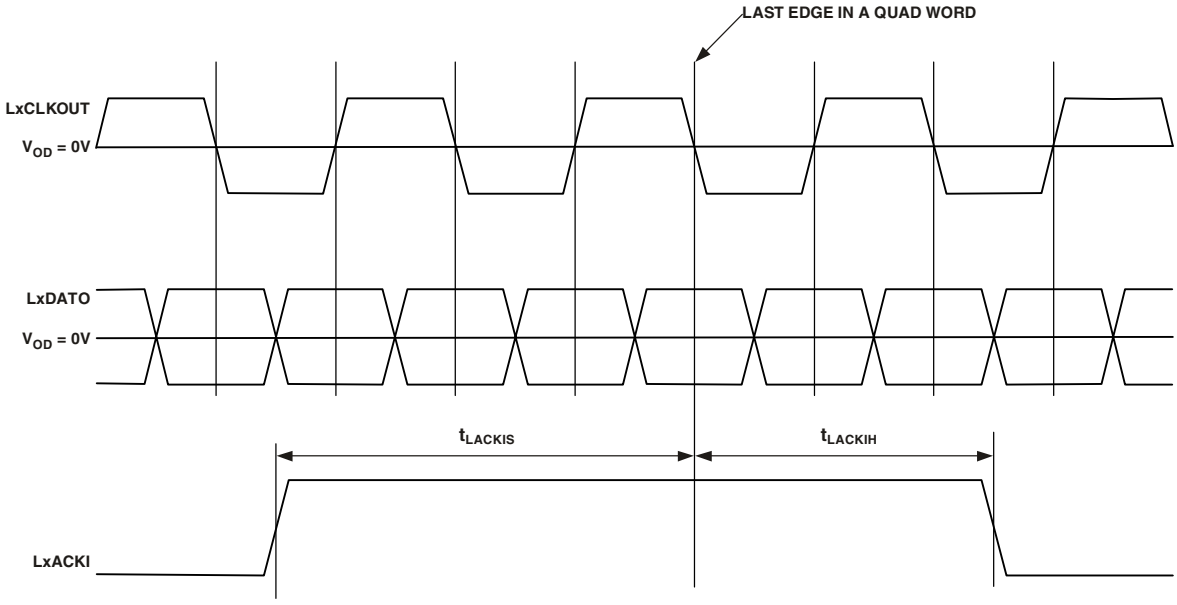

Figure 23. Link Ports—Back to Back Transmission

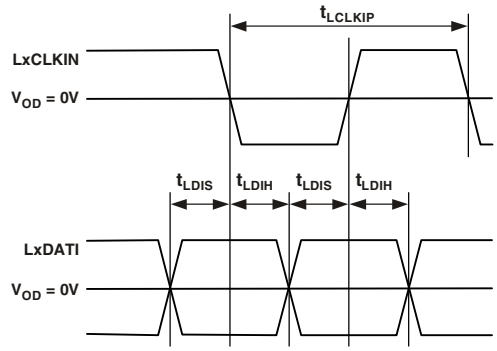

Figure 25. Link Ports—Data Input Setup and Hold[1]

[1] These parameters are valid for both clock edges.

OUTPUT DRIVE CURRENTS

Figure 26 through Figure 33 show typical I–V characteristics for the output drivers of the ADSP-TS201S processor. 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. Typical drive currents for intermediate temperatures (such as 85°C) should be obtained from the curves using linear interpolation. For complete output driver characteristics, refer to the DSP's IBIS models, available on the Analog Devices website (www.analog.com).

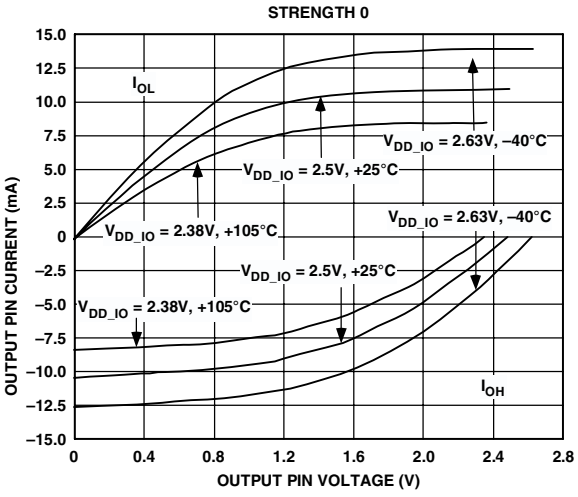

Figure 26. Typical Drive Currents at Strength 0

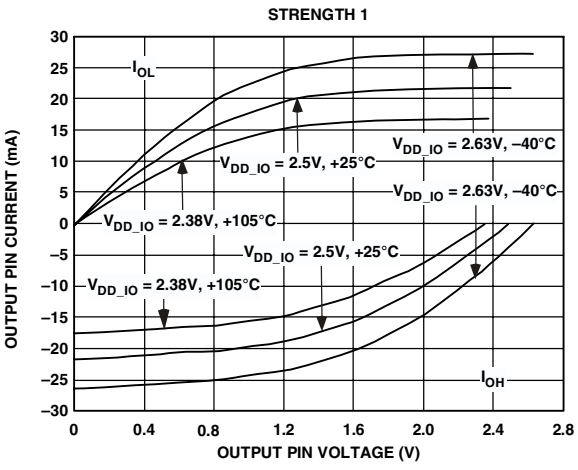

Figure 27. Typical Drive Currents at Strength 1

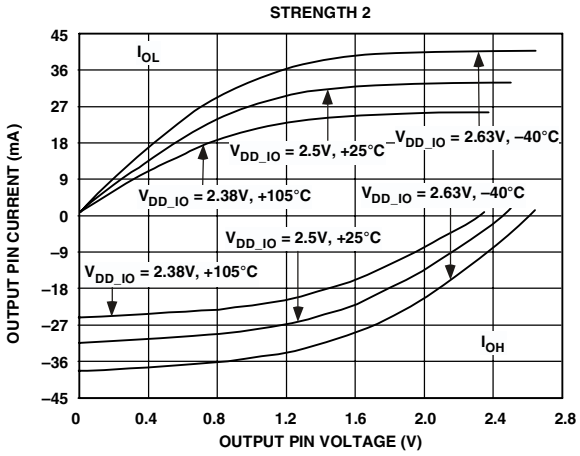

Figure 28. Typical Drive Currents at Strength 2

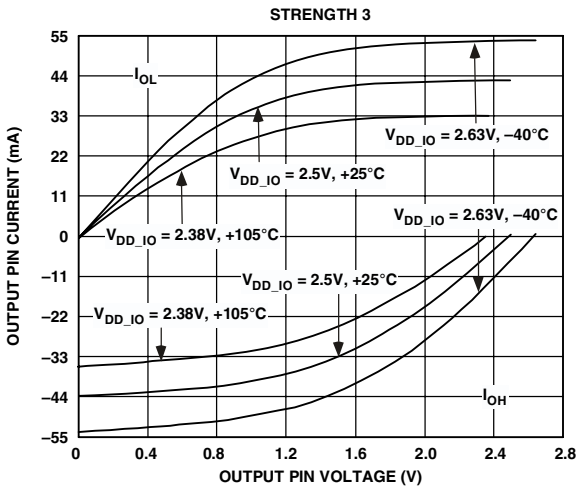

Figure 29. Typical Drive Currents at Strength 3

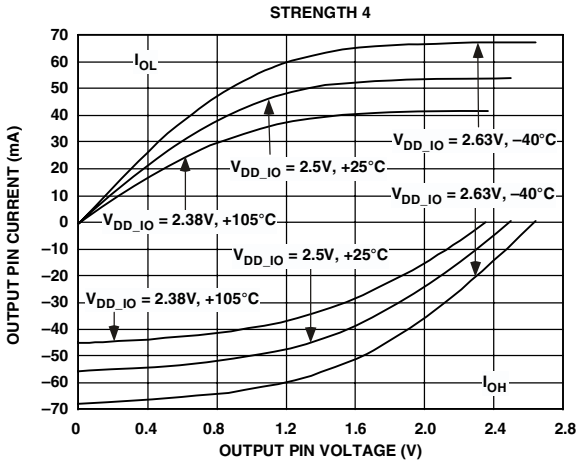

Figure 30. Typical Drive Currents at Strength 4

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:

$$t_{RAMP} = (C_L \Delta V)/I_D$$

The output enable time t_{ENA} is the difference between $t_{MEASURED\_ENA}$ and t_{RAMP} as shown in Figure 35. 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. t_{RAMP} is calculated with test load C_L, drive current I_D, and with ΔV equal to 0.4 V.

Capacitive Loading

Output valid and hold are based on standard capacitive loads: 30 pF on all pins (see Figure 36). 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 37 through Figure 44 show how output rise time varies with capacitance. Figure 45 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 37.) The graphs of Figure 37 through Figure 45 may not be linear outside the ranges shown.

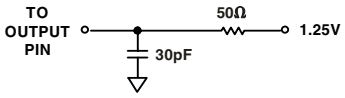

Figure 36. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

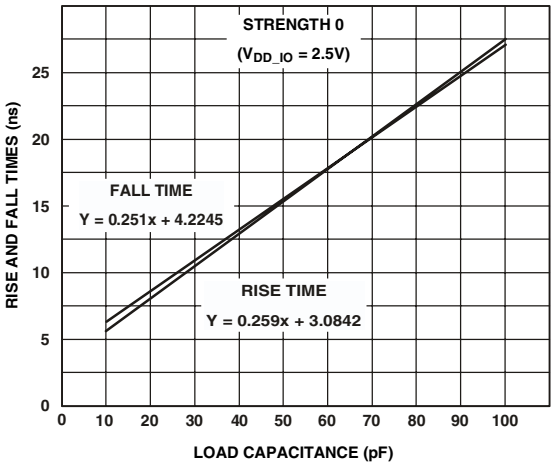

Figure 37. Typical Output Rise and Fall Time (10% to 90%, V_{DD_IO} = 2.5 V) vs. Load Capacitance at Strength 0

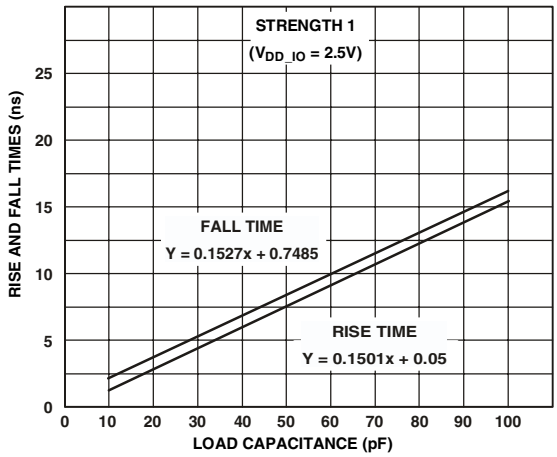

Figure 38. Typical Output Rise and Fall Time (10% to 90%, V_{DD_IO} = 2.5 V) vs. Load Capacitance at Strength 1

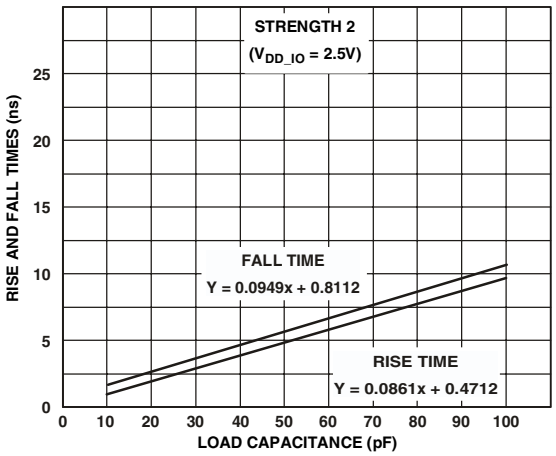

Figure 39. Typical Output Rise and Fall Time (10% to 90%, V_{DD_IO} = 2.5 V) vs. Load Capacitance at Strength 2

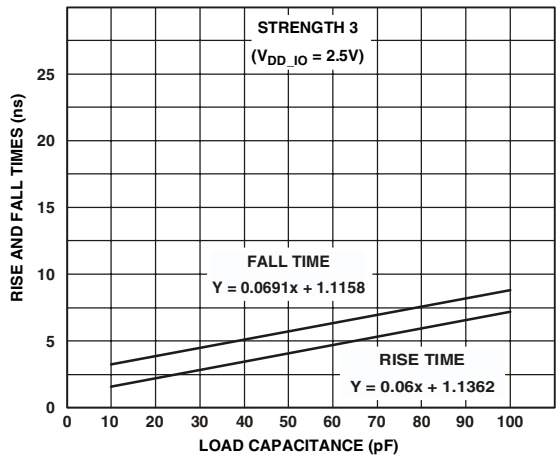

Figure 40. Typical Output Rise and Fall Time (10% to 90%, V_{DD_IO} = 2.5 V) vs. Load Capacitance at Strength 3

576-BALL BGA_ED PIN CONFIGURATIONS

Figure 46 shows a summary of pin configurations for the 576-ball BGA_ED package and Table 35 lists the signal-to-ball assignments.

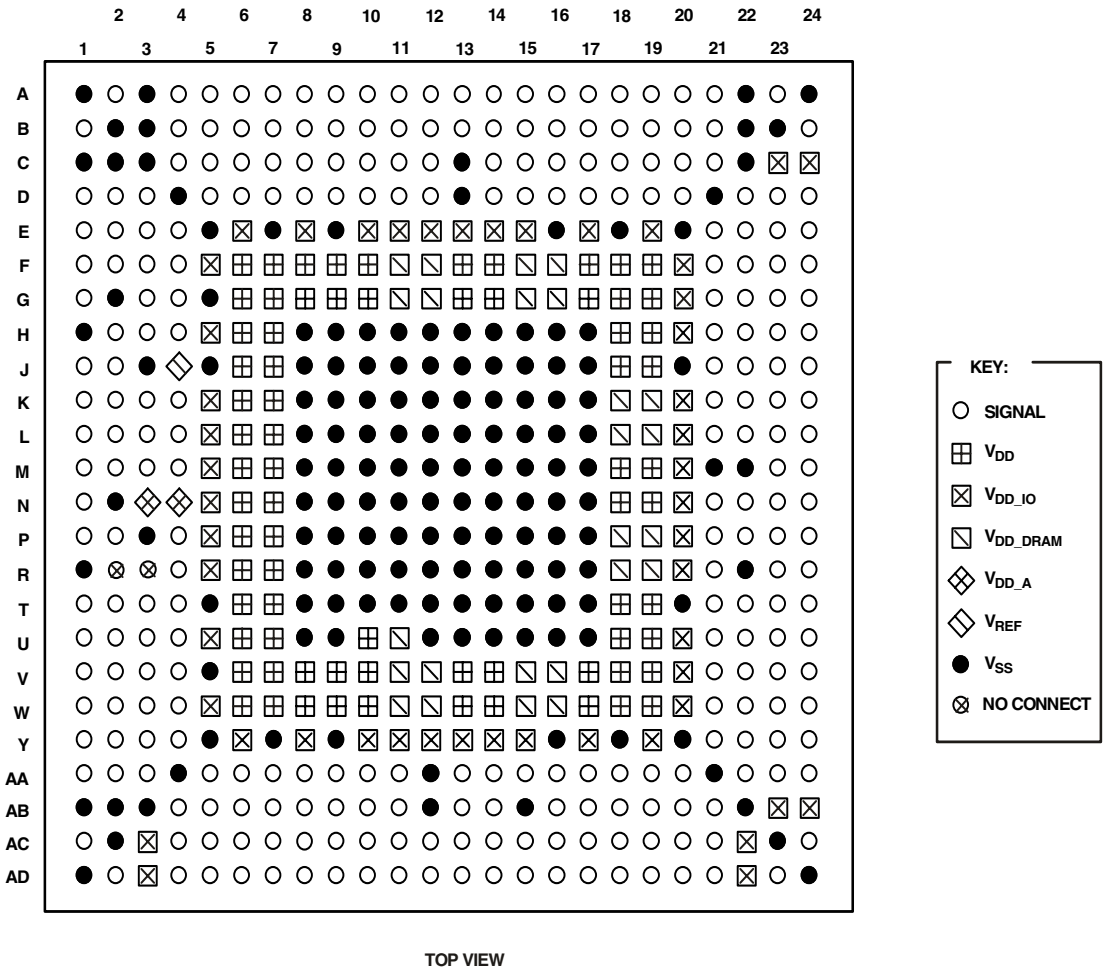

Figure 46. 576-Ball BGA_ED Pin Configurations[1] (Top View, Summary)

[1] For a more detailed pin summary diagram, see the *EE-179: ADSP-TS201S System Design Guidelines* on the Analog Devices website (www.analog.com).

Table 35. 576-Ball (25 mm × 25 mm) BGA_ED Ball Assignments

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
A1	V_{SS}	B1	DATA53	C1	V_{SS}	D1	DATA55
A2	DATA51	B2	V_{SS}	C2	V_{SS}	D2	DATA56
A3	V_{SS}	B3	V_{SS}	C3	V_{SS}	D3	DATA54
A4	DATA49	B4	DATA50	C4	DATA52	D4	V_{SS}
A5	DATA43	B5	DATA44	C5	DATA47	D5	DATA48
A6	DATA41	B6	DATA42	C6	DATA45	D6	DATA46
A7	DATA37	B7	DATA38	C7	DATA39	D7	DATA40
A8	DATA33	B8	DATA34	C8	DATA35	D8	DATA36
A9	DATA29	B9	DATA30	C9	DATA31	D9	DATA32
A10	DATA25	B10	DATA26	C10	DATA27	D10	DATA28
A11	DATA23	B11	DATA24	C11	DATA21	D11	DATA22
A12	DATA19	B12	DATA20	C12	DATA17	D12	DATA18
A13	DATA15	B13	DATA16	C13	V_{SS}	D13	V_{SS}
A14	DATA11	B14	DATA12	C14	DATA13	D14	DATA14
A15	DATA9	B15	DATA10	C15	DATA7	D15	DATA8
A16	DATA5	B16	DATA6	C16	DATA3	D16	DATA4
A17	DATA1	B17	DATA2	C17	ACK	D17	DATA0
A18	$\overline{WRL}$	B18	$\overline{WRH}$	C18	$\overline{RD}$	D18	$\overline{BRST}$
A19	ADDR30	B19	ADDR31	C19	ADDR26	D19	ADDR27
A20	ADDR28	B20	ADDR29	C20	ADDR24	D20	ADDR25
A21	ADDR22	B21	ADDR23	C21	ADDR20	D21	V_{SS}
A22	V_{SS}	B22	V_{SS}	C22	V_{SS}	D22	ADDR19
A23	ADDR21	B23	V_{SS}	C23	V_{DD_IO}	D23	ADDR17
A24	V_{SS}	B24	ADDR18	C24	V_{DD_IO}	D24	ADDR16
E1	DATA61	F1	DATA63	G1	$\overline{MSSD1}$	H1	V_{SS}
E2	DATA62	F2	$\overline{MS1}$	G2	V_{SS}	H2	$\overline{MSH}$
E3	DATA57	F3	DATA59	G3	$\overline{MS0}$	H3	$\overline{MSSD3}$
E4	DATA58	F4	DATA60	G4	$\overline{BMS}$	H4	SCLKRAT0
E5	V_{SS}	F5	V_{DD_IO}	G5	V_{SS}	H5	V_{DD_IO}
E6	V_{DD_IO}	F6	V_{DD}	G6	V_{DD}	H6	V_{DD}
E7	V_{SS}	F7	V_{DD}	G7	V_{DD}	H7	V_{DD}
E8	V_{DD_IO}	F8	V_{DD}	G8	V_{DD}	H8	V_{SS}
E9	V_{SS}	F9	V_{DD}	G9	V_{DD}	H9	V_{SS}
E10	V_{DD_IO}	F10	V_{DD}	G10	V_{DD}	H10	V_{SS}
E11	V_{DD_IO}	F11	V_{DD_DRAM}	G11	V_{DD_DRAM}	H11	V_{SS}
E12	V_{DD_IO}	F12	V_{DD_DRAM}	G12	V_{DD_DRAM}	H12	V_{SS}
E13	V_{DD_IO}	F13	V_{DD}	G13	V_{DD}	H13	V_{SS}
E14	V_{DD_IO}	F14	V_{DD}	G14	V_{DD}	H14	V_{SS}
E15	V_{DD_IO}	F15	V_{DD_DRAM}	G15	V_{DD_DRAM}	H15	V_{SS}
E16	V_{SS}	F16	V_{DD_DRAM}	G16	V_{DD_DRAM}	H16	V_{SS}
E17	V_{DD_IO}	F17	V_{DD}	G17	V_{DD}	H17	V_{SS}
E18	V_{SS}	F18	V_{DD}	G18	V_{DD}	H18	V_{DD}
E19	V_{DD_IO}	F19	V_{DD}	G19	V_{DD}	H19	V_{DD}
E20	V_{SS}	F20	V_{DD_IO}	G20	V_{DD_IO}	H20	V_{DD_IO}
E21	ADDR15	F21	ADDR13	G21	ADDR7	H21	ADDR3
E22	ADDR14	F22	ADDR12	G22	ADDR6	H22	ADDR2
E23	ADDR11	F23	ADDR9	G23	ADDR5	H23	ADDR1
E24	ADDR10	F24	ADDR8	G24	ADDR4	H24	ADDR0

Table 35. 576-Ball (25 mm × 25 mm) BGA_ED Ball Assignments (Continued)

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
U1	$\overline{\text{MSSD0}}$	V1	$\overline{\text{MSSD2}}$	W1	CONTROLIMP0	Y1	$\overline{\text{EMU}}$
U2	$\overline{\text{RST\_OUT}}$	V2	DS2	W2	ENEDREG	Y2	TCK
U3	ID2	V3	$\overline{\text{POR\_IN}}$	W3	TDI	Y3	TMR0E
U4	DS1	V4	CONTROLIMP1	W4	TDO	Y4	FLAG3
U5	V_{DD_IO}	V5	V_{SS}	W5	V_{DD_IO}	Y5	V_{SS}
U6	V_{DD}	V6	V_{DD}	W6	V_{DD}	Y6	V_{DD_IO}
U7	V_{DD}	V7	V_{DD}	W7	V_{DD}	Y7	V_{SS}
U8	V_{SS}	V8	V_{DD}	W8	V_{DD}	Y8	V_{DD_IO}
U9	V_{SS}	V9	V_{DD}	W9	V_{DD}	Y9	V_{SS}
U10	V_{DD}	V10	V_{DD}	W10	V_{DD}	Y10	V_{DD_IO}
U11	V_{DD_DRAM}	V11	V_{DD_DRAM}	W11	V_{DD_DRAM}	Y11	V_{DD_IO}
U12	V_{SS}	V12	V_{DD_DRAM}	W12	V_{DD_DRAM}	Y12	V_{DD_IO}
U13	V_{SS}	V13	V_{DD}	W13	V_{DD}	Y13	V_{DD_IO}
U14	V_{SS}	V14	V_{DD}	W14	V_{DD}	Y14	V_{DD_IO}
U15	V_{SS}	V15	V_{DD_DRAM}	W15	V_{DD_DRAM}	Y15	V_{DD_IO}
U16	V_{SS}	V16	V_{DD_DRAM}	W16	V_{DD_DRAM}	Y16	V_{SS}
U17	V_{SS}	V17	V_{DD}	W17	V_{DD}	Y17	V_{DD_IO}
U18	V_{DD}	V18	V_{DD}	W18	V_{DD}	Y18	V_{SS}
U19	V_{DD}	V19	V_{DD}	W19	V_{DD}	Y19	V_{DD_IO}
U20	V_{DD_IO}	V20	V_{DD_IO}	W20	V_{DD_IO}	Y20	V_{SS}
U21	L1CLKINN	V21	L1DATI3_N	W21	L1CLKON	Y21	L1DATO1_N
U22	L1CLKINP	V22	L1DATI3_P	W22	L1CLKOP	Y22	L1DATO1_P
U23	L1DATI1_N	V23	L1DATI2_N	W23	L1DATO3_N	Y23	L1DATO2_N
U24	L1DATI1_P	V24	L1DATI2_P	W24	L1DATO3_P	Y24	L1DATO2_P
AA1	FLAG2	AB1	V_{SS}	AC1	FLAG0	AD1	V_{SS}
AA2	FLAG1	AB2	V_{SS}	AC2	V_{SS}	AD2	ID1
AA3	$\overline{\text{IRQ3}}$	AB3	V_{SS}	AC3	V_{DD_IO}	AD3	V_{DD_IO}
AA4	V_{SS}	AB4	NC	AC4	TMS	AD4	$\overline{\text{TRST}}$
AA5	$\overline{\text{IRQ0}}$	AB5	$\overline{\text{IRQ2}}$	AC5	$\overline{\text{IOWR}}$	AD5	$\overline{\text{IORD}}$
AA6	$\overline{\text{IOEN}}$	AB6	$\overline{\text{IRQ1}}$	AC6	$\overline{\text{DMAR2}}$	AD6	$\overline{\text{DMAR3}}$
AA7	$\overline{\text{DMAR0}}$	AB7	$\overline{\text{DMAR1}}$	AC7	$\overline{\text{CPA}}$	AD7	$\overline{\text{DPA}}$
AA8	$\overline{\text{HBR}}$	AB8	$\overline{\text{HBG}}$	AC8	$\overline{\text{BOFF}}$	AD8	$\overline{\text{BUSLOCK}}$
AA9	$\overline{\text{L3BCMPO}}$	AB9	L3ACKI	AC9	L3DATO0_N	AD9	L3DATO0_P
AA10	L3DATO1_N	AB10	L3DATO1_P	AC10	L3CLKON	AD10	L3CLKOP
AA11	L3DATO3_N	AB11	L3DATO3_P	AC11	L3DATO2_N	AD11	L3DATO2_P
AA12	V_{SS}	AB12	V_{SS}	AC12	L3DATI3_N	AD12	L3DATI3_P
AA13	L3DATI2_N	AB13	L3DATI2_P	AC13	L3CLKINN	AD13	L3CLKINP
AA14	L3DATI1_N	AB14	L3DATI1_P	AC14	L3DATI0_N	AD14	L3DATI0_P
AA15	NC	AB15	V_{SS}	AC15	L3ACKO	AD15	$\overline{\text{L3BCMPI}}$
AA16	L2DATO0_N	AB16	L2DATO0_P	AC16	$\overline{\text{L2BCMPO}}$	AD16	L2ACKI
AA17	L2CLKON	AB17	L2CLKOP	AC17	L2DATO1_N	AD17	L2DATO1_P
AA18	L2DATO3_N	AB18	L2DATO3_P	AC18	L2DATO2_N	AD18	L2DATO2_P
AA19	L2CLKINN	AB19	L2CLKINP	AC19	L2DATI3_N	AD19	L2DATI3_P
AA20	L2DATI1_N	AB20	L2DATI1_P	AC20	L2DATI2_N	AD20	L2DATI2_P
AA21	V_{SS}	AB21	L2ACKO	AC21	L2DATI0_N	AD21	L2DATI0_P
AA22	$\overline{\text{L1BCMPO}}$	AB22	V_{SS}	AC22	V_{DD_IO}	AD22	V_{DD_IO}
AA23	L1DATO0_N	AB23	V_{DD_IO}	AC23	V_{SS}	AD23	$\overline{\text{L2BCMPI}}$
AA24	L1DATO0_P	AB24	V_{DD_IO}	AC24	L1ACKI	AD24	V_{SS}

[1] On revision 1.x silicon, the R2 and R3 balls are NC. On revision 0.x silicon, the R2 ball is SCLK, and the R3 ball is SCLK_V_{REF}. For more information on SCLK and SCLK_V_{REF} on revision 0.x silicon, see the *EE-179: ADSP-TS20x TigerSHARC System Design Guidelines* on the Analog Devices website (*www.analog.com*).

D04324-0-11/06(C)