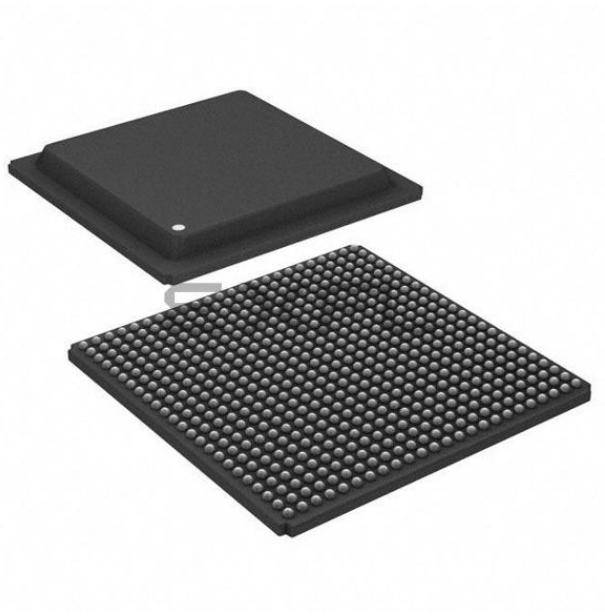


**Welcome to E-XFL.COM**

## 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 | 600MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 3MB |
| Voltage - I/O | 2.50V |
| Voltage - Core | 1.20V |
| 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-ts201sabpz060 |

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

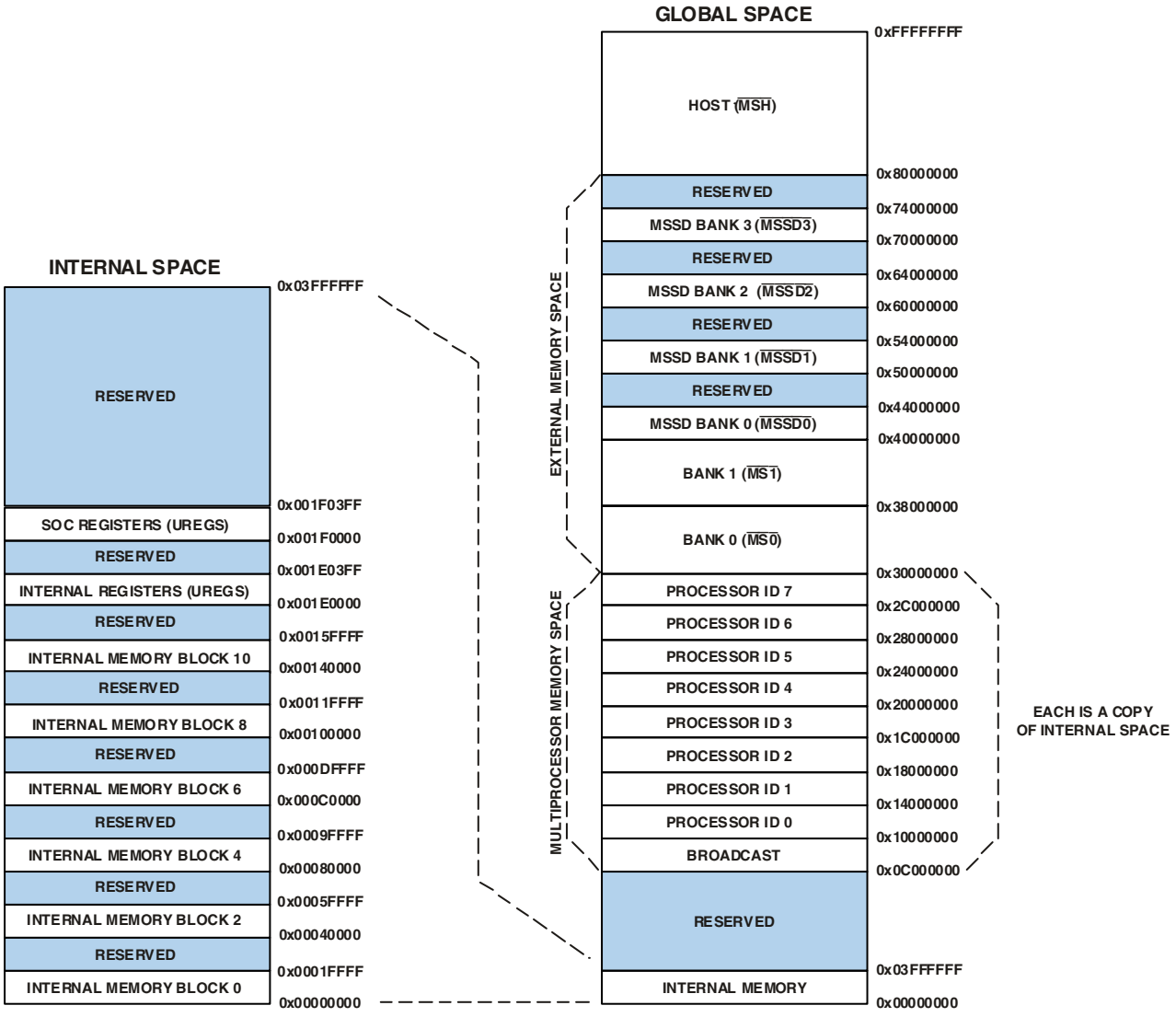



*Figure 3. ADSP-TS201S Memory Map*

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

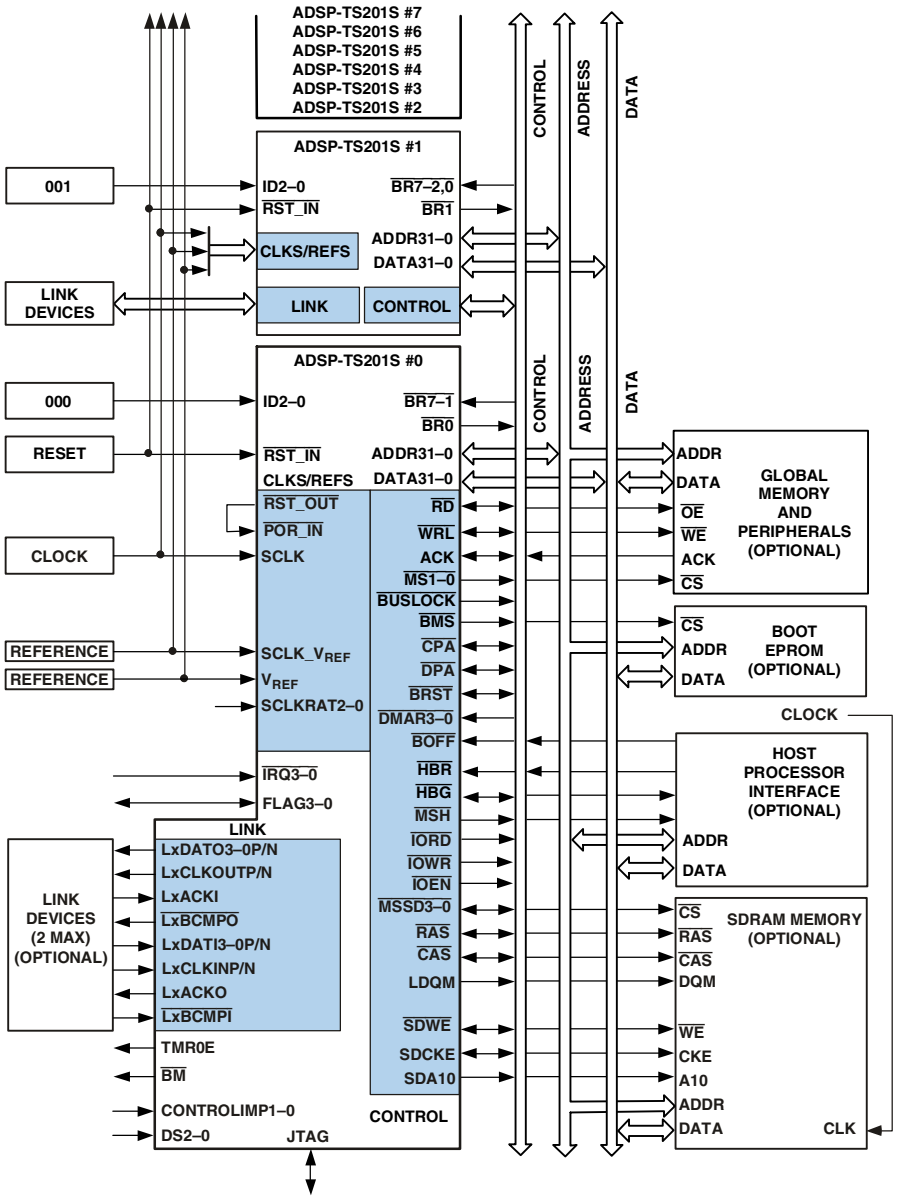



*Figure 4. ADSP-TS201S Shared Memory Multiprocessing System*

external memory. These transfers only use handshake mode protocol. DMA priority rotates between the four receive channels.

- AutoDMA transfers. Two dedicated unidirectional DMA channels transfer data received from an external bus master to internal memory or to link port I/O. These transfers only use slave mode protocol, and an external bus master must initiate the transfer.

The DMA controller provides these additional features:

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

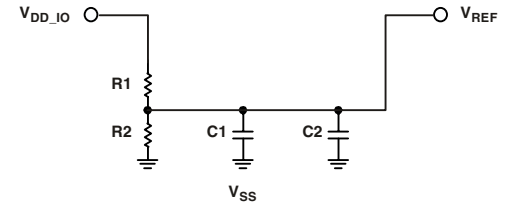
## POWER DOMAINS

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

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

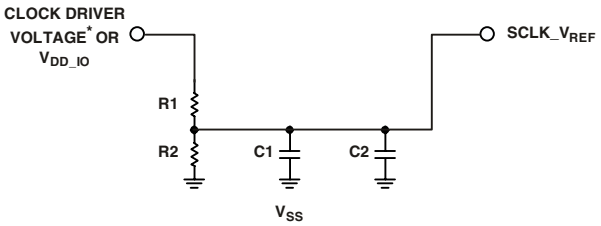
## FILTERING REFERENCE VOLTAGE AND CLOCKS

Figure 6 and Figure 7 show possible circuits for filtering $V_{REF}$, and SCLK_$V_{REF}$. These circuits provide the reference voltages for the switching voltage reference and system clock reference.




R1: 2kΩ SERIES RESISTOR (±1%)
R2: 2.55kΩ SERIES RESISTOR (±1%)
C1: 1µF CAPACITOR (SMD)
C2: 1nF CAPACITOR (HF SMD) PLACED CLOSE TO DSP'S PINS

*Figure 6. $V_{REF}$ Filtering Scheme*




R1: 2kΩ SERIES RESISTOR (±1%)
R2: 2.55kΩ SERIES RESISTOR (±1%)
C1: 1µF CAPACITOR (SMD)
C2: 1nF CAPACITOR (HF SMD) PLACED CLOSE TO DSP'S PINS
*IF CLOCK DRIVER VOLTAGE > $V_{DD\_IO}$

*Figure 7. SCLK_$V_{REF}$ Filtering Scheme*

## DEVELOPMENT TOOLS

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

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy to use assembler (which is based on an algebraic syntax), an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ run-time library that includes DSP and mathematical functions. A key point for theses tools is C/C++ code efficiency. The compiler has been developed for efficient translation of C/C++ code to DSP assembly. The DSP has architectural features that improve the efficiency of compiled C/C++ code.

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

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

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

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

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

The VisualDSP++ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of DSP programming. These capabilities enable engineers to develop code more effectively,

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

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 5. Pin Definitions—External Port Bus Controls**

| Signal | Type | Term | Description |
|---|---|---|---|
| ADDR31–0 | I/O/T (pu_ad) | nc | Address Bus. The DSP issues addresses for accessing memory and peripherals on these pins. In a multiprocessor system, the bus master drives addresses for accessing internal memory or I/O processor registers of other ADSP-TS201S processors. The DSP inputs addresses when a host or another DSP accesses its internal memory or I/O processor registers. |
| DATA63–0 | I/O/T (pu_ad) | nc | External Data Bus. The DSP drives and receives data and instructions on these pins. Pull-up or pull-down resistors on unused DATA pins are unnecessary. |
| $\overline{RD}$ | I/O/T (pu_0) | epu[1] | 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 $\overline{RD}$. $\overline{RD}$ changes concurrently with ADDR pins. |
| $\overline{WRL}$ | I/O/T (pu_0) | epu[1] | Write Low. $\overline{WRL}$ is asserted in two cases: when the ADSP-TS201S processor writes to an even address word of external memory or to another external bus agent; and when the ADSP-TS201S processor 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 $\overline{WRL}$. $\overline{WRL}$ 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. |
| $\overline{WRH}$ | I/O/T (pu_0) | epu[1] | Write High. $\overline{WRH}$ is asserted when the ADSP-TS201S processor 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 $\overline{WRH}$. $\overline{WRH}$ 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/OD (pu_od_0) | epu[1] | 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 and write accesses of its internal memory. The pull-up is 50 Ω on low-to-high transactions and is 500 Ω on all other transactions. |
| $\overline{BMS}$ | O/T (pu_0) | na | 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. In a multiprocessor system, the DSP bus master drives $\overline{BMS}$. For details, see Reset and Booting on Page 9 and the EBOOT signal description in Table 16 on Page 20. |
| $\overline{MS1–0}$ | O/T (pu_0) | nc | Memory Select. $\overline{MS0}$ or $\overline{MS1}$ is asserted whenever the DSP accesses memory banks 0 or 1, respectively. $\overline{MS1–0}$ are decoded memory address pins that change concurrently with ADDR pins. When ADDR31:27 = 0b00110, $\overline{MS0}$ is asserted. When ADDR31:27 = 0b00111, $\overline{MS1}$ is asserted. In multiprocessor systems, the master DSP drives $\overline{MS1–0}$. |
| $\overline{MSH}$ | O/T (pu_0) | nc | Memory Select Host. $\overline{MSH}$ is asserted whenever the DSP accesses the host address space (ADDR31 = 0b1). $\overline{MSH}$ is a decoded memory address pin that changes concurrently with ADDR pins. In a multiprocessor system, the bus master DSP drives $\overline{MSH}$. |
| $\overline{BRST}$ | I/O/T (pu_0) | epu[1] | 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. |

**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] This external pull-up may be omitted for the ID = 000 TigerSHARC processor.

**Table 6. Pin Definitions—External Port Arbitration**

| Signal | Type | Term | Description |
|---|---|---|---|
| $\overline{BR7–0}$ | I/O | $V_{DD\_IO}$[1] | 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 ($V_{DD\_IO}$). |
| ID2–0 | I (pd) | na | Multiprocessor ID. Indicates the DSP's ID, from which 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. |
| $\overline{BM}$ | O | na | 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 20. |
| $\overline{BOFF}$ | I | epu | Back Off. A deadlock situation can occur when the host and a DSP try to read from each other's bus at the same time. When deadlock occurs, the host can assert $\overline{BOFF}$ to force the DSP to relinquish the bus before completing its outstanding transaction. |
| $\overline{BUSLOCK}$ | O/T (pu_0) | na | Bus Lock Indication. Provides an indication that the current bus master has locked the bus. At reset, this is a strap pin. For more information, see Table 16 on Page 20. |
| $\overline{HBR}$ | I | epu | Host Bus Request. A host must assert $\overline{HBR}$ to request control of the DSP's external bus. When $\overline{HBR}$ is asserted in a multiprocessing system, the bus master relinquishes the bus and asserts $\overline{HBG}$ once the outstanding transaction is finished. |
| $\overline{HBG}$ | I/O/T (pu_0) | epu[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, $\overline{MSH}$, $\overline{MSSD3–0}$, $\overline{MS1–0}$, $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{BMS}$, $\overline{BRST}$, $\overline{IORD}$, $\overline{IOWR}$, $\overline{IOEN}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{SDWE}$, 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 it. |
| $\overline{CPA}$ | I/O/OD (pu_od_0) | epu[2] | 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. If not required in the system, leave $\overline{CPA}$ unconnected (external pull-ups will be required for DSP ID = 1 through ID = 7). |
| $\overline{DPA}$ | I/O/OD (pu_od_0) | epu[2] | 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. If not required in the system, leave $\overline{DPA}$ unconnected (external pull-ups will be required for DSP ID = 1 through ID = 7). |

**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] The $\overline{BRx}$ pin matching the ID2–0 input selection for the processor should be left nc if unused. For example, the processor with ID = 000 has $\overline{BR0}$ = nc and $\overline{BR7–1}$ = $V_{DD\_IO}$.
[2] This external pull-up resistor may be omitted for the ID = 000 TigerSHARC processor.

**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 $V_{SS}$

**Table 11. Pin Definitions—Link Ports**

| Signal | Type | Term | Description |
|---|---|---|---|
| LxDATO3–0P | O | nc | Link Ports 3–0 Data 3–0 Transmit LVDS P |
| LxDATO3–0N | O | nc | Link Ports 3–0 Data 3–0 Transmit LVDS N |
| LxCLKOUTP | O | nc | Link Ports 3–0 Transmit Clock LVDS P |
| LxCLKOUTN | O | nc | Link Ports 3–0 Transmit Clock LVDS N |
| LxACKI | I (pd) | nc | Link Ports 3–0 Receive Acknowledge. Using this signal, the receiver indicates to the transmitter that it may continue the transmission. |
| $\overline{\text{LxBCMPO}}$ | O (pu) | nc | Link Ports 3–0 Block Completion. When the transmission is executed using DMA, this signal indicates to the receiver that the transmitted block is completed. The pull-up resistor is present on $\overline{\text{L0BCMPO}}$ only. At reset, the $\overline{\text{L1BCMPO}}$, $\overline{\text{L2BCMPO}}$, and $\overline{\text{L3BCMPO}}$ pins are strap pins. For more information, see Table 16 on Page 20. |
| LxDATI3–0P | I | $V_{DD\_IO}$ | Link Ports 3–0 Data 3–0 Receive LVDS P |
| LxDATI3–0N | I | $V_{DD\_IO}$ | Link Ports 3–0 Data 3–0 Receive LVDS N |
| LxCLKINP | I/A | $V_{DD\_IO}$ | Link Ports 3–0 Receive Clock LVDS P |
| LxCLKINN | I/A | $V_{DD\_IO}$ | Link Ports 3–0 Receive Clock LVDS N |
| LxACKO | O | nc | Link Ports 3–0 Transmit Acknowledge. Using this signal, the receiver indicates to the transmitter that it may continue the transmission. |
| $\overline{\text{LxBCMPI}}$ | I (pd_l) | $V_{SS}$ | Link Ports 3–0 Block Completion. When the reception is executed using DMA, this signal indicates to the receiver that the transmitted block is completed. |

**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; **pd_l** = internal pull-down 50 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 12. Pin Definitions—Impedance Control, Drive Strength Control, and Regulator Enable**

| Signal | Type | Term | Description |
|---|---|---|---|
| CONTROLIMP0<br>CONTROLIMP1 | I (pd)<br>I (pu) | na<br>na | Impedance Control. As shown in Table 13, the CONTROLIMP1–0 pins select between normal driver mode and A/D driver mode. When using normal mode (recommended), the output drive strength is set relative to maximum drive strength according to Table 14. When using A/D mode, the resistance control operates in the analog mode, where drive strength is continuously controlled to match a specific line impedance as shown in Table 14. |
| DS2, 0<br>DS1 | I (pu)<br>I (pd) | na | Digital Drive Strength Selection. Selected as shown in Table 14. For drive strength calculation, see Output Drive Currents on Page 36. The drive strength for some pins is preset, not controlled by the DS2–0 pins. The pins that are always at drive strength 7 (100%) include: $\overline{\text{CPA}}$, $\overline{\text{DPA}}$, TDO, $\overline{\text{EMU}}$, and $\overline{\text{RST\_OUT}}$. The drive strength for the ACK pin is always x2 drive strength 7 (100%). |
| ENEDREG | I (pu) | $V_{SS}$ | Connect the ENEDREG pin to $V_{SS}$. Connect the $V_{DD\_DRAM}$ pins to a properly decoupled DRAM power supply. |

**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 13. Impedance Control Selection**

| CONTROLIMP1-0 | Driver Mode |
|---|---|
| 00 (recommended) | Normal |
| 01 | Reserved |
| 10 (default) | A/D Mode |
| 11 | Reserved |

**Table 14. Drive Strength/Output Impedance Selection**

| DS2–0 Pins | Drive Strength[1] | Output Impedance [2] |
|---|---|---|
| 000 | Strength 0 (11.1%) | 26 Ω |
| 001 | Strength 1 (23.8%) | 32 Ω |
| 010 | Strength 2 (36.5%) | 40 Ω |
| 011 | Strength 3 (49.2%) | 50 Ω |
| 100 | Strength 4 (61.9%) | 62 Ω |
| 101 (default) | Strength 5 (74.6%) | 70 Ω |
| 110 | Strength 6 (87.3%) | 96 Ω |
| 111 | Strength 7 (100%) | 120 Ω |

[1] CONTROLIMP1 = 0, A/D mode disabled.
[2] CONTROLIMP1 = 1, A/D mode enabled.

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

| Signal | Type | Term | Description |
|---|---|---|---|
| $V_{DD}$ | P | na | $V_{DD}$ pins for internal logic. |
| $V_{DD\_A}$ | P | na | $V_{DD}$ pins for analog circuits. Pay critical attention to bypassing this supply. |
| $V_{DD\_IO}$ | P | na | $V_{DD}$ pins for I/O buffers. |
| $V_{DD\_DRAM}$ | P | na | $V_{DD}$ pins for internal DRAM. |
| $V_{REF}$ | I | na | Reference voltage defines the trip point for all input buffers, except SCLK, $\overline{\text{RST\_IN}}$, $\overline{\text{POR\_IN}}$, $\overline{\text{IRQ3–0}}$, FLAG3–0, $\overline{\text{DMAR3–0}}$, ID2–0, CONTROLIMP1–0, LxDATO3–0P/N, LxCLKOUTP/N, LxDATI3–0P/N, LxCLKINP/N, TCK, TDI, TMS, and $\overline{\text{TRST}}$. $V_{REF}$ can be connected to a power supply or set by a voltage divider circuit as shown in Figure 6. For more information, see Filtering Reference Voltage and Clocks on Page 10. |
| $SCLK\_V_{REF}$ | I | na | System Clock Reference. Connect this pin to a reference voltage as shown in Figure 7. For more information, see Filtering Reference Voltage and Clocks on Page 10. |
| $V_{SS}$ | G | na | Ground pins. |
| NC | — | nc | No Connect. Do not connect these pins to anything (not to any supply, signal, or each other). These pins are reserved and must be left unconnected. |

**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<br>(pd_0) | $\overline{\text{BMS}}$ | EPROM Boot.<br>0 = boot from EPROM immediately after reset (default)<br>1 = idle after reset and wait for an external device to boot DSP through the external port or a link port |
| IRQEN | I<br>(pd) | $\overline{\text{BM}}$ | Interrupt Enable.<br>0 = disable and set $\overline{\text{IRQ3–0}}$ interrupts to edge-sensitive after reset (default)<br>1 = enable and set $\overline{\text{IRQ3–0}}$ interrupts to level-sensitive immediately after reset |
| LINK_DWIDTH | I<br>(pd) | TMR0E | Link Port Input Default Data Width.<br>0 = 1-bit (default)<br>1 = 4-bit |
| SYS_REG_WE | I<br>(pd_0) | $\overline{\text{BUSLOCK}}$ | SYSCON and SDRCON Write Enable.<br>0 = one-time writable after reset (default)<br>1 = always writable |
| TM1 | I<br>(pu) | $\overline{\text{L1BCMPO}}$ | Test Mode 1. Do not overdrive default value during reset. |
| TM2 | I<br>(pu) | $\overline{\text{L2BCMPO}}$ | Test Mode 2. Do not overdrive default value during reset. |
| TM3 | I<br>(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.

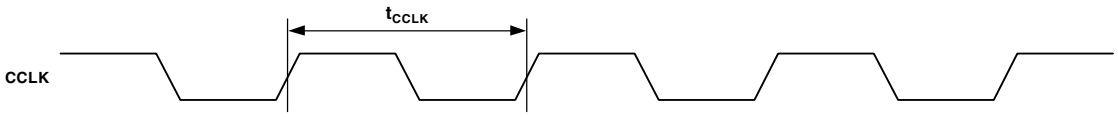



*Figure 9. Reference Clocks—Core Clock (CCLK) Cycle Time*

**Table 25. Power-Up Timing[1]**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirement* | | | | |
| $t_{VDD\_DRAM}$ | $V_{DD\_DRAM}$ Stable After $V_{DD}$, $V_{DD\_A}$, $V_{DD\_IO}$ Stable | >0 | | ms |

[1] For information about power supply sequencing and monitoring solutions, please visit www.analog.com/sequencing.

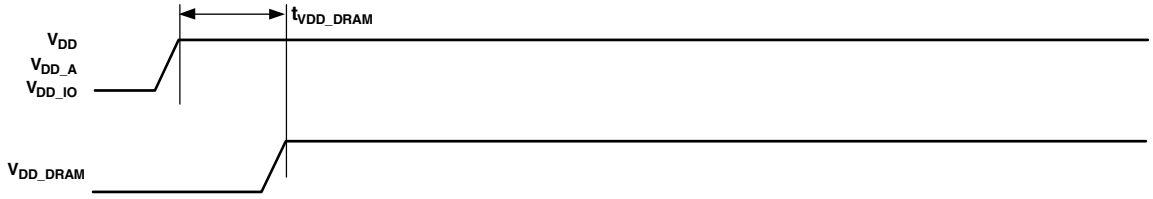



*Figure 12. Power-Up Timing*

**Table 26. Power-Up Reset Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{RST\_IN\_PWR}$ | $\overline{RST\_IN}$ Deasserted After $V_{DD}$, $V_{DD\_A}$, $V_{DD\_IO}$, $V_{DD\_DRAM}$, SCLK, and Static/ Strap Pins Stable | 2 | | ms |
| $t_{TRST\_IN\_PWR}$[1] | $\overline{TRST}$ Asserted During Power-Up Reset | 100 × $t_{SCLK}$ | | ns |
| *Switching Characteristic* | | | | |
| $t_{RST\_OUT\_PWR}$ | $\overline{RST\_OUT}$ Deasserted After $\overline{RST\_IN}$ Deasserted | 1.5 | | ms |

[1] Applies after $V_{DD}$, $V_{DD\_A}$, $V_{DD\_IO}$, $V_{DD\_DRAM}$, and SCLK are stable and before $\overline{RST\_IN}$ deasserted.

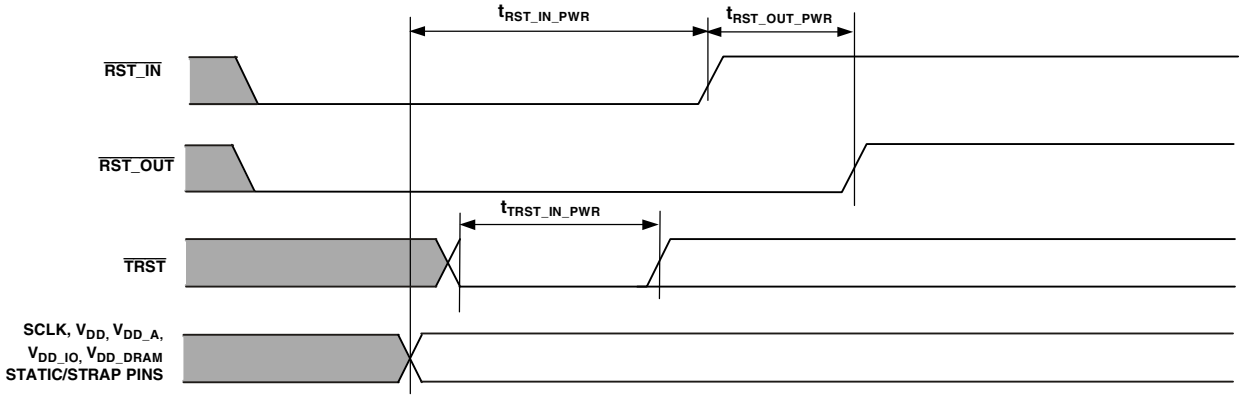



*Figure 13. Power-Up Reset Timing*

**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 | — | — | — | — | — | — | — |

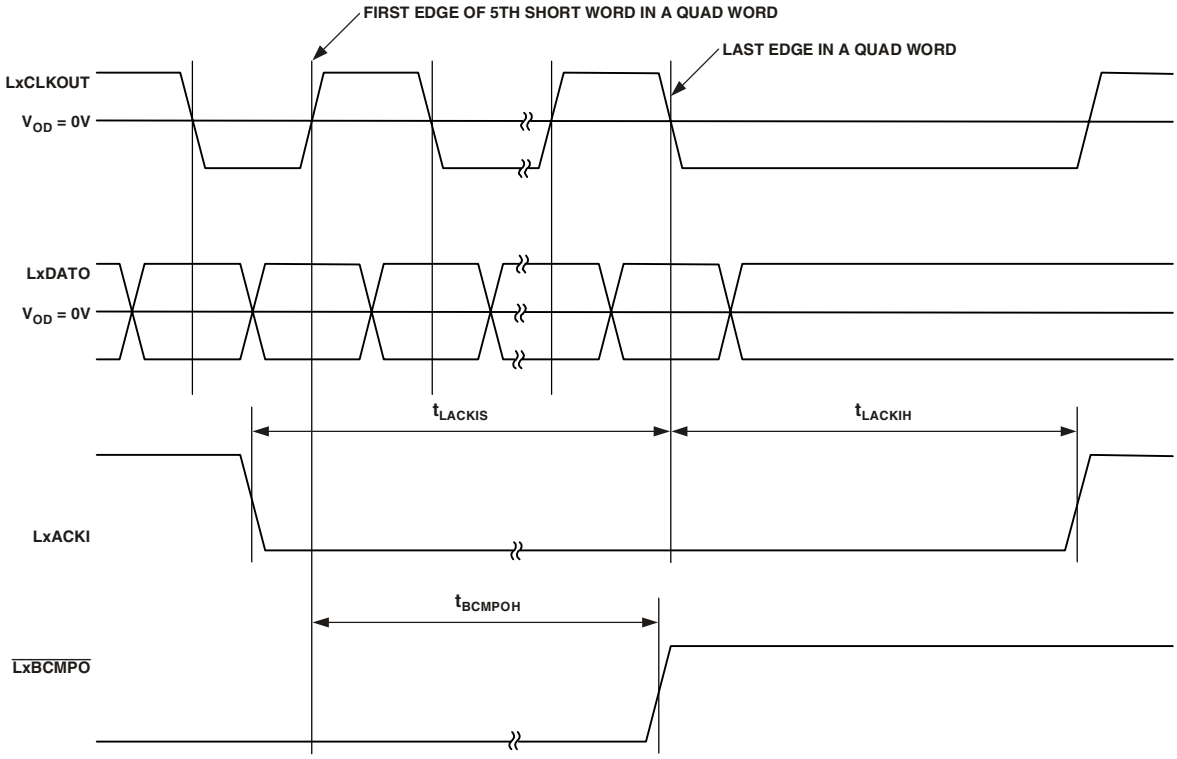



*Figure 22. Link Ports—Transmission End and Stops*

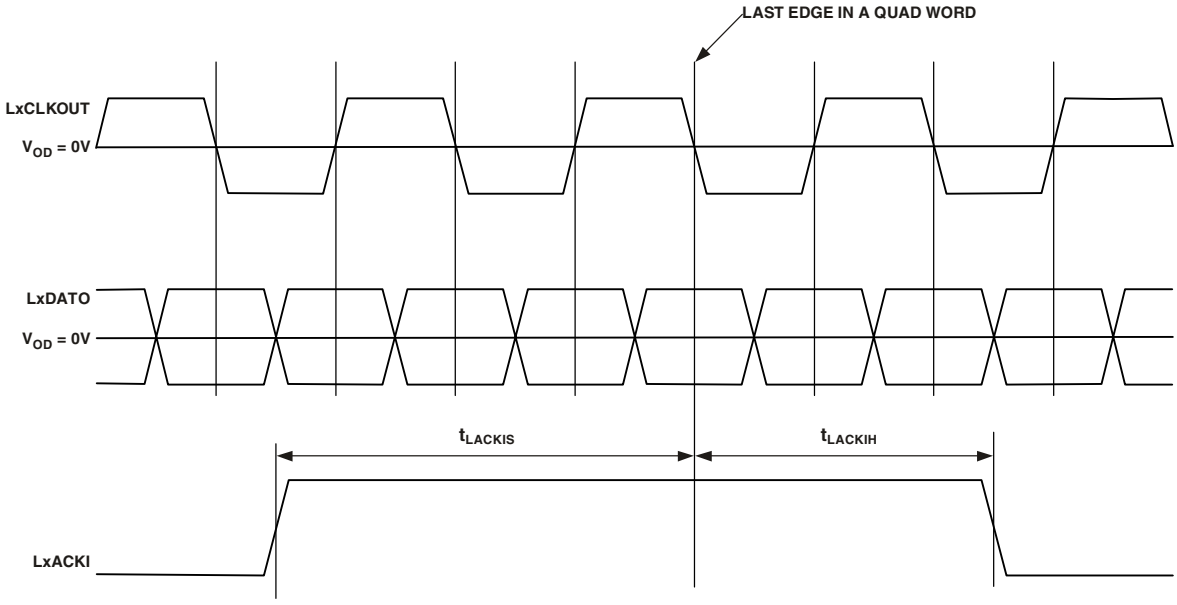



*Figure 23. Link Ports—Back to Back Transmission*

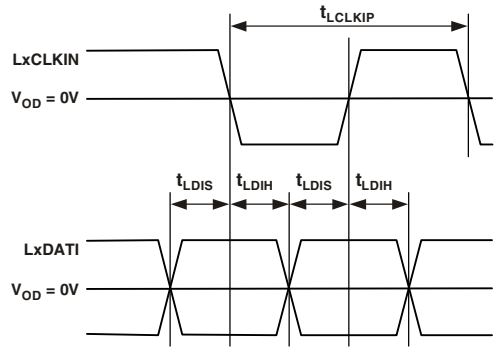



*Figure 25. Link Ports—Data Input Setup and Hold*[1]

[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).

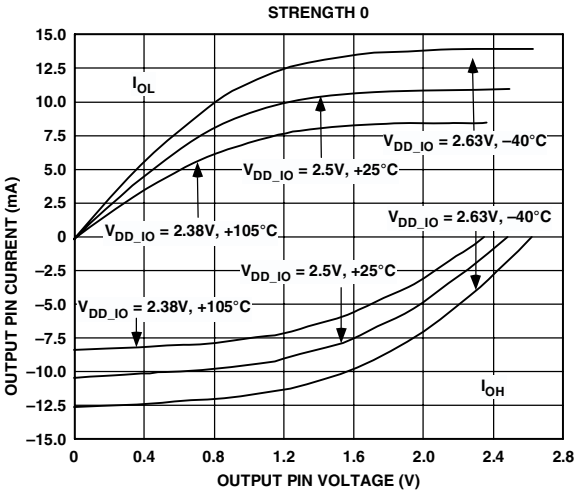



*Figure 26. Typical Drive Currents at Strength 0*

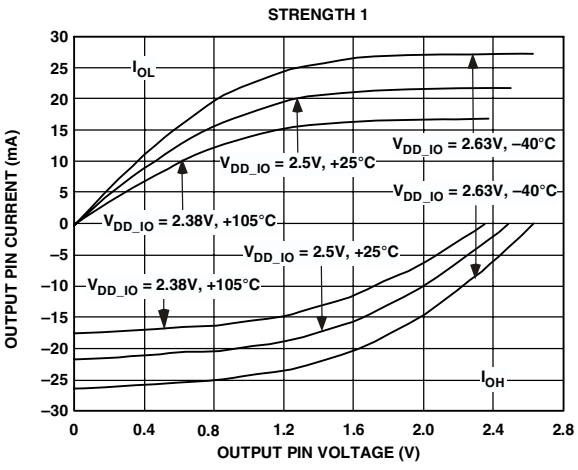



*Figure 27. Typical Drive Currents at Strength 1*

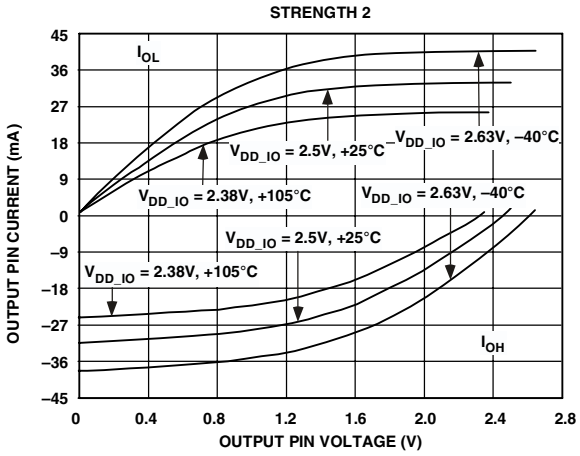



*Figure 28. Typical Drive Currents at Strength 2*

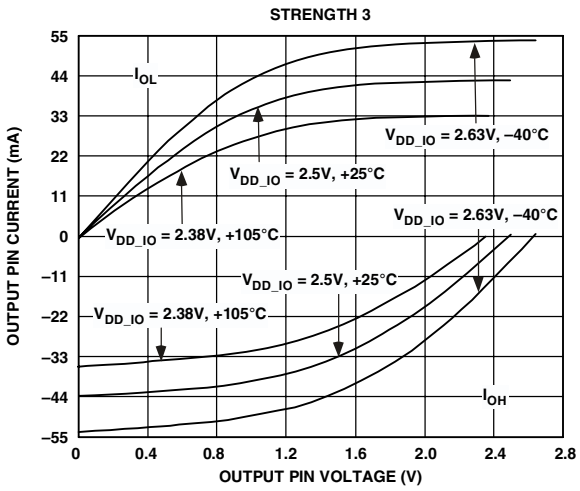



*Figure 29. Typical Drive Currents at Strength 3*

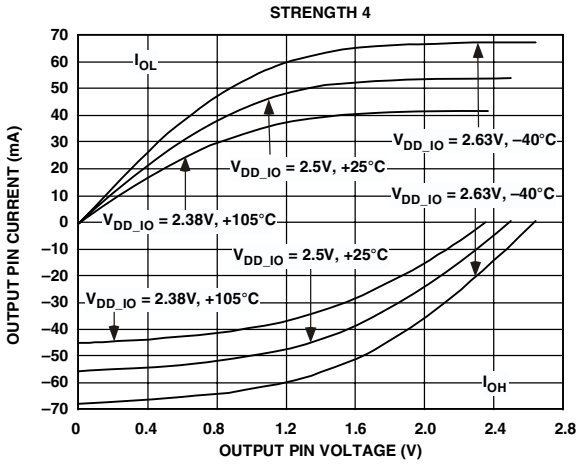



*Figure 30. Typical Drive Currents at Strength 4*

**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 |

D04324-0-11/06(C)