ANALOG DEVICES

BLACKfin DSP™

Preliminary Technical Data ADSP-21535

SUMMARY

300 MHz High-Performance Blackfin DSP Core
- **Two 16-Bit MACs, Two 40-Bit ALUs, Two 40-Bit Accumulators, Four 8-Bit Video ALUs, and a 40-Bit Shifter**

RISC-Like Register and Instruction Model for Ease of Programming and Compiler-Friendly Support

Advanced Debug, Trace, and Performance- Monitoring

0.9–1.5 V Core V_{DD} with Dynamic Power Management

3.3 V I/O

0ºC To +85ºC Case Commercial Temperature Range

-40ºC To +105ºC Case Industrial Temperature Range (200 MHz)

260-Lead PBGA Package

MEMORY

4G-Byte Unified Address Range

308K Bytes of On-Chip Memory:
- **16K Bytes of Instruction SRAM/Cache**
- **32K Bytes of Data SRAM/Cache**
- **4K Bytes of Scratchpad SRAM**
- **256K Bytes of Full Speed, Low Latency SRAM**

Memory DMA Controller

Memory Mgmt Unit Providing Memory Protection

Glueless External Memory Controllers
- **Synchronous SDRAM Support**
- **Asynchronous with SRAM, Flash, ROM Support**

PERIPHERALS

32-Bit, 33-MHz, 3.3 V, PCI 2.2-compliant Bus Interface with Master and Slave Support

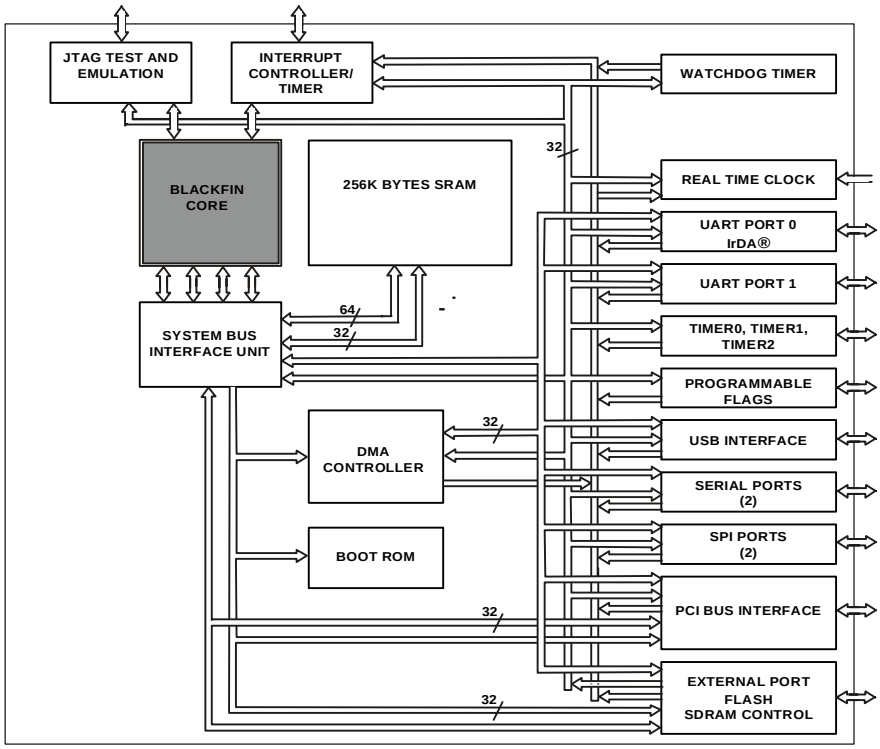
Integrated USB 1.1-compliant Device Interface

Two UARTs, One with IrDA®

Two SPI-compatible Ports

Two Full-Duplex Synchronous Serial Ports (SPORTs)

FUNCTIONAL BLOCK DIAGRAM



REV. PrC

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

Four Timer/Counters, Three with PWM Support
Sixteen Bi-Directional Programmable Flag I/O Pins
Watchdog Timer
Real-Time Clock
On-Chip PLL with 1x To 31x Frequency Multiplier

General Note
This data sheet provides preliminary information for the ADSP-21535 Blackfin DSP.

GENERAL DESCRIPTION

The ADSP-21535 is a member of the Blackfin DSP family of products, incorporating the Micro Signal Architecture (MSA), jointly developed by Analog Devices, Inc. and Intel Corporation. The architecture combines a dual-MAC state-of-the-art DSP engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction set architecture.

By integrating a rich set of industry leading system peripherals and memory, Blackfin DSPs are the platform of choice for next generation applications that require RISC like programmability, multimedia support and leading edge signal processing in one integrated DSP.

Portable Low-Power Architecture

Blackfin DSPs provide world class power dissipation and performance compared to other Digital Signal Processors. Blackfin DSPs are designed in a Low-Power and Low-Voltage Design Methodology and feature Dynamic Power Management, the ability to vary both the voltage and frequency of operation to significantly lower overall power consumption. Varying the voltage and frequency can result in a three-fold reduction in power consumption, by comparison to just varying the frequency of operation. This translates into longer battery life for portable appliances.

System Integration

The ADSP-21535 is a highly integrated system-on-a-chip solution for the next generation of digital communication and portable Internet appliances. By combining industry-standard interfaces with a high performance Digital Signal Processing core, users can develop cost effective solutions quickly without the need for costly external components. The ADSP-21535 system peripherals include UARTs, SPIs, SPORTs, General Purpose Timers, a Real-Time Clock, Programmable Flags, Watchdog Timer, and USB and PCI buses for glueless peripheral expansion.

ADSP-21535 Peripherals

The ADSP-21535 contains a rich set of peripherals connected to the core via several high bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance. See Functional Block Diagram on page 1. The base peripherals include general purpose functions such as UARTs, Timers with PWM (Pulse Width Modulator) and pulse measurement capability, general purpose flag I/O pins, a Real-Time Clock, and a Watchdog Timer. This set of functions satisfies a wide variety of typical system support needs and is augmented by the system expansion capabilities of the part. In addition to these general-purpose peripherals, the ADSP-21535 contains high speed serial ports for interfaces to a variety of audio and modem CODEC functions. It also contains an event handler for flexible management of interrupts from the on-chip peripherals and external sources and power management control functions to tailor the performance and power characteristics of the processor and system to many application scenarios.

The on-chip peripherals can be easily augmented in many system designs with little or no glue logic due to the inclusion of several interfaces providing expansion on industry-standard buses. These include a 32-bit, 33-MHz, V2.2-compliant PCI bus, SPI serial expansion ports and a device type USB port. These enable the connection of a large variety of peripheral devices to tailor the system design to specific applications with a minimum of design complexity.

All of the peripherals, except for programmable flags, Real-Time Clock, and timers, are supported by a flexible DMA structure with individual DMA channels integrated into the peripherals. There is also a separate memory DMA channel dedicated to data transfers between the DSP's various memory spaces including external SDRAM and asynchronous memory, internal Level 1 and Level 2 SRAM and PCI memory spaces. Multiple on-chip 32-bit buses running at up to 133 MHz provide adequate bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

Blackfin DSP Core

As shown in Figure 1, the Blackfin DSP core contains two multiplier/accumulators (MACs), two 40-bit ALUs, four video ALUs, and a single shifter. The computational units process 8-bit, 16-bit, or 32-bit data from the register file.

Each MAC performs a 16-bit by 16-bit multiply in every cycle, with an accumulation to a 40-bit result, providing 8 bits of extended precision.

The ALUs perform a standard set of arithmetic and logical operations. With two ALUs capable of operating on 16- or 32-bit data, the flexibility of the computation units covers the signal processing requirements of a varied set of application needs. Each of the two 32-bit input registers can be regarded as two 16-bit halves, so each ALU can accomplish very flexible single 16-bit arithmetic operations. By viewing the registers as pairs of 16-bit operands, dual 16-bit or single 32-bit operations can be accomplished in a single cycle. By further taking advantage of the second ALU, quad 16-bit operations can be accomplished simply, accelerating the per cycle throughput.

The powerful 40-bit shifter has extensive capabilities for performing shifting, rotating, normalization, extraction, and depositing of data.

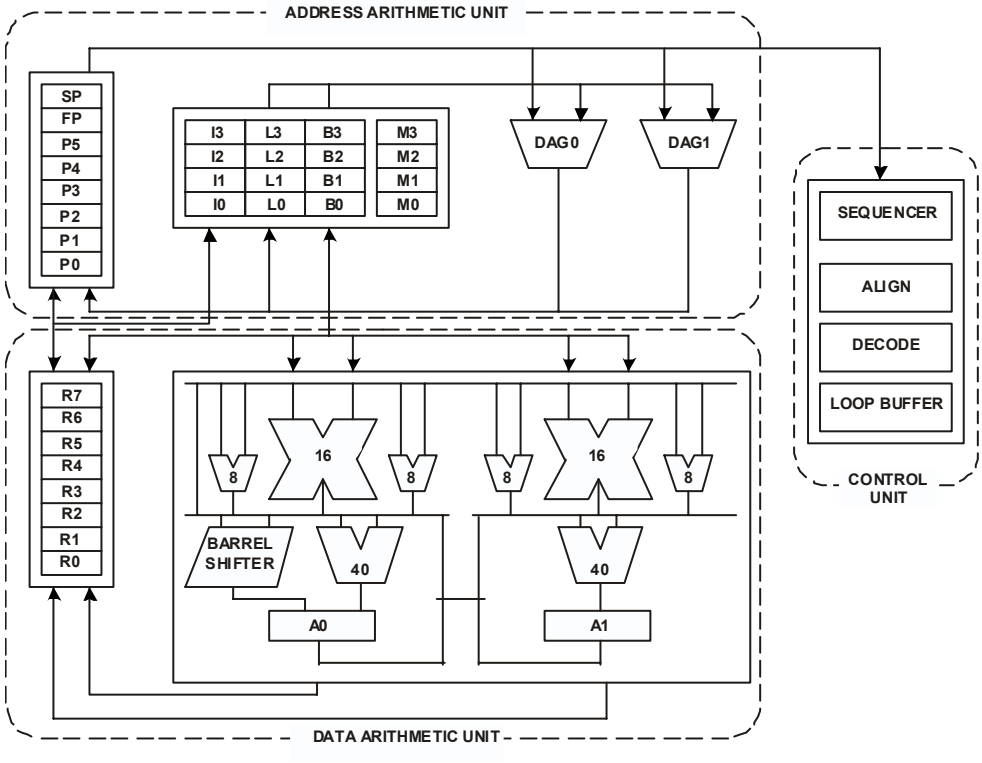


Figure 1. Blackfin DSP Core

The data for the computational units is found in a multi-ported register file of sixteen 16-bit entries or eight 32-bit entries.

A powerful program sequencer controls the flow of instruction execution, including instruction alignment and decoding. The sequencer supports conditional jumps and subroutine calls, as well as zero-overhead looping. A loop buffer stores instructions locally, eliminating instruction memory accesses for tight looped code.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches from memory. The DAGs share a register file containing four sets of 32-bit Index, Modify, Length, and Base registers. Eight additional 32-bit registers provide pointers for general indexing of variables and stack locations.

Blackfin DSPs support a modified Harvard architecture in combination with a hierarchical memory structure. Level 1 (L1) memories are those that typically operate at the full processor speed with little or no latency. Level 2 (L2) memories are other memories, on-chip or off-chip, that may take multiple processor cycles to access. At the L1 level, the instruction memory holds instructions only. The two data memories hold data, and a dedicated scratchpad data memory stores stack and local variable information. At the L2 level, there is a single unified memory space, holding both instructions and data.

In addition, the L1 instruction memory and L1 data memories may be configured as either Static RAMs (SRAMs) or caches. The Memory Management Unit (MMU) provides memory protection for individual tasks that may be operating on the core and may protect system registers from unintended access.

The architecture provides three modes of operation: user mode, supervisor mode, and emulation mode. User mode has restricted access to certain system resources, thus providing a protected software environment, while supervisor mode has unrestricted access to the system and core resources.

The Blackfin DSP instruction set has been optimized so that 16-bit op-codes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit op-codes, representing fully featured multifunction instructions. Blackfin DSPs support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

The Blackfin DSP assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the C-compiler, resulting in fast and efficient software implementations.

Memory Architecture

The ADSP-21535 views memory as a single unified 4G-byte address space, using 32-bit addresses. All resources including internal memory, external memory, PCI address spaces, and I/O control registers occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low-latency memory as cache or SRAM very close to the processor, and larger, lower-cost and performance-memory systems farther away from the processor. See Figure 2.

The L1 memory system is the primary highest-performance memory available to the Blackfin DSP core. The L2 memory provides additional capacity with slightly lower performance. Lastly, the off-chip memory system, accessed through the External Bus Interface Unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing more than 768M bytes of physical memory.

The memory DMA controller provides high-bandwidth data-movement capability. It can perform block transfers of code or data between the internal L1/L2 memories and the external memory spaces (including PCI memory space).

Internal (On-chip) Memory

The ADSP-21535 has four blocks of on-chip memory providing high-bandwidth access to the core.

The first is the L1 instruction memory consisting of 16K bytes of 4-way set-associative cache memory. In addition the memory may be configured as an SRAM. This memory is accessed at full processor speed.

The second on-chip memory block is the L1 data memory, consisting of two banks of 16K bytes each. Each L1 data memory bank can be configured as one way of a two-way set associative cache or as an SRAM, and is accessed at full speed by the core.

The third memory block is a 4K-byte scratchpad RAM which runs at the same speed as the L1 memories, but is only accessible as data SRAM (it cannot be configured as cache memory and is not accessible via DMA).

The fourth on-chip memory system is the L2 SRAM memory array which provides 256K bytes of high speed SRAM at the full bandwidth of the core, and slightly longer latency than the L1 memory banks. The L2 memory is a unified instruction and data memory and can hold any mixture of code and data required by the system design.

The Blackfin DSP core has a dedicated low-latency 64-bit wide datapath port into the L2 SRAM memory. For example, at a core frequency of 300 MHz, the peak data transfer rate across this interface is up to 2.4G bytes per second.

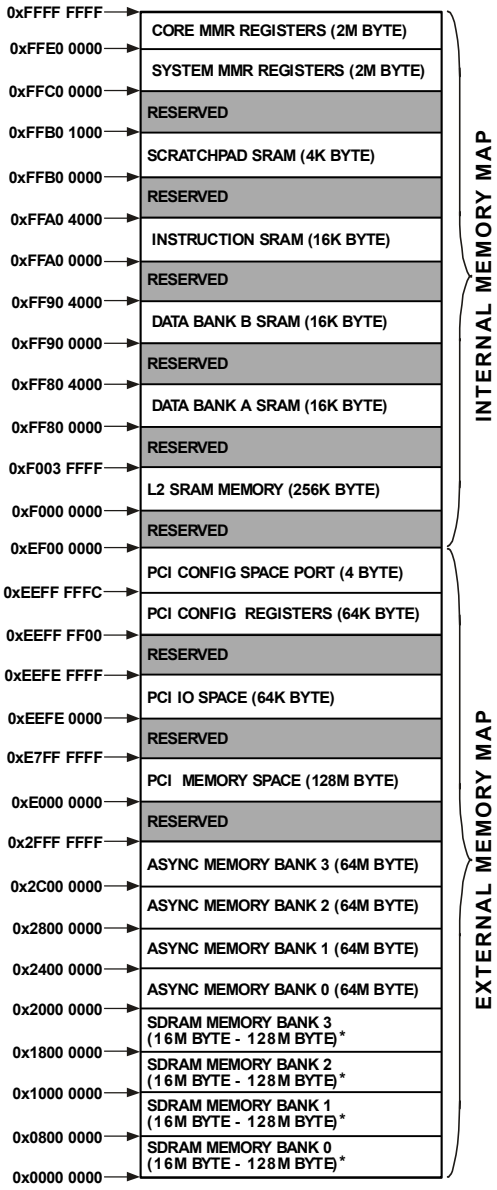


Figure 2. Internal/External Memory Map

External (Off-Chip) Memory

External memory is accessed via the External Bus Interface Unit (EBIU). This interface provides a glueless connection to up to four banks of synchronous DRAM (SDRAM) as

well as up to four banks of asynchronous memory devices including flash, EPROM, ROM, SRAM, and memory mapped I/O devices.

The PC133-compliant SDRAM controller can be programmed to interface to up to four banks of SDRAM, with each bank containing between 16M bytes and 128M bytes providing access to up to 512M bytes of SDRAM. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement. This allows flexible configuration and upgradability of system memory while allowing the core to view all SDRAM as a single, contiguous, physical address space.

The asynchronous memory controller can also be programmed to control up to four banks of devices with very flexible timing parameters for a wide variety of devices. Each bank occupies a 64M-byte segment regardless of the size of the devices used so that these banks will only be contiguous if fully populated with 64M bytes of memory.

PCI

The PCI bus defines three separate address spaces, which are accessed through windows in the ADSP-21535 memory space. These are PCI memory, PCI I/O, and PCI configuration space.

In addition, the PCI interface can either be used as a bridge from the processor core as the controlling CPU in the system, or as a host port where another CPU in the system is the host and the ADSP-21535 is functioning as an intelligent I/O device on the PCI bus.

When the ADSP-21535 acts as the system controller, it views the PCI address spaces through its mapped windows and can initialize all devices in the system and maintain a map of the topology of the environment.

The PCI memory region is a 4G-byte space that appears on the PCI bus and can be used to map memory I/O devices on the bus. The ADSP-21535 uses a 128M-byte window in memory space to see a portion of the PCI memory space. A base address register is provided to position this window anywhere in the 4G-byte PCI memory space while its position with respect to the processor addresses remains fixed.

The PCI I/O region is also a 4G-byte space. However, most systems and I/O devices only use a 64K-byte subset of this space for I/O mapped addresses. The ADSP-21535 implements a 64K-byte window into this space along with a base address register which can be used to position it anywhere in the PCI I/O address space, while the window remains at the same address in the processor's address space.

PCI configuration space is a limited address space, which is used for system enumeration and initialization and which is a very low-performance communication mode between the processor and PCI devices. The ADSP-21535 provides a one-value window to access a single data value at any address in PCI configuration space. This window is fixed and receives the address of the value, and the value if the operation is a write. Otherwise the device returns the value into the same address on a read operation.

I/O Memory Space

Blackfin DSPs do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses near the top of the 4G-byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The core MMRs are accessible only by the core and only in supervisor mode and appear as reserved space by on-chip peripherals, as as well as external devices accessing resources through the PCI bus. The system MMRs are accessible by the core in supervisor mode and can be mapped as either visible or reserved to other devices, depending on the system protection model desired.

Booting

The ADSP-21535 contains a small boot kernel, which configures the appropriate peripheral for booting. If the ADSP-21535 is configured to boot from boot ROM memory space, the DSP starts executing from the on-chip boot ROM. For more information, see Booting Modes on page 14.

Event Handling

The event controller on the ADSP-21535 handles all asynchronous and synchronous events to the processor. The ADSP-21535 provides event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing of a higher-priority event takes precedence over servicing of a lower-priority event. The controller provides support for five different types of events:

- Emulation – An emulation event causes the processor to enter emulation mode, allowing command and control of the processor via the JTAG interface.
- Reset – This event resets the processor.
- Non-Maskable Interrupt (NMI) – The NMI event can be generated by the software watchdog timer or by the NMI input signal to the processor. The NMI event is frequently used as a power-down indicator to initiate an orderly shut down of the system.

- Exceptions – Exceptions are events that occur synchronously to program flow, i.e., the exception will be taken before the instruction is allowed to complete. Conditions such as data alignment violations, undefined instructions, etc. cause exceptions.
- Interrupts – Interrupts are events that occur asynchronously to program flow. They are caused by timers, peripherals, input pins, etc.

Each event has an associated register to hold the return address and an associated return-from-event instruction. When an event is triggered, the state of the processor is saved on the supervisor stack.

The ADSP-21535 event controller consists of two stages, the Core Event Controller (CEC) and the System Interrupt Controller (SIC). The Core Event Controller works with the System Interrupt Controller to prioritize and control all system events. Conceptually, interrupts from the peripherals enter into the SIC, and are then routed directly into the general-purpose interrupts of the CEC.

Core Event Controller (CEC)

The CEC supports nine general-purpose interrupts (IVG15–7), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest-priority interrupts (IVG15–14) are recommended to be reserved for software interrupt handlers, leaving seven prioritized interrupt inputs to support the peripherals of the ADSP-21535. Table 1 describes the inputs to the CEC, identifies their names in the Event Vector Table (EVT), and lists their priorities.

Table 1. Core Event Controller (CEC)

Priority (0 is Highest)	Event Class	EVT Entry
0	Emulation/Test	EMU
1	Reset	RST
2	Non-Maskable	NMI
3	Exceptions	EVX
4	Global Enable	-
5	Hardware Error	IVHW
6	Core Timer	IVTMR
7	General Interrupt 7	IVG7
8	General Interrupt 8	IVG8
9	General Interrupt 9	IVG9
10	General Interrupt 10	IVG10
11	General Interrupt 11	IVG11
12	General Interrupt 12	IVG12
13	General Interrupt 13	IVG13
14	General Interrupt 14	IVG14
15	General Interrupt 15	IVG15

System Interrupt Controller (SIC)

The System Interrupt Controller provides the mapping and routing of events from the many peripheral interrupt sources, to the prioritized general-purpose interrupt inputs of the CEC. Although the ADSP-21535 provides a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the Interrupt Assignment Registers (IAR). Table 2 describes the inputs into the SIC and the default mappings into the CEC.

Table 2. System Interrupt Controller (SIC)

Peripheral Interrupt Event	Peripheral Interrupt ID	Default Mapping
Real-Time Clock	0	IVG7
Reserved	1	-
USB	2	IVG7
PCI Interrupt	3	IVG7
SPORT 0 Rx DMA	4	IVG8
SPORT 0 Tx DMA	5	IVG8
SPORT 1 Rx DMA	6	IVG8
SPORT 1 Tx DMA	7	IVG8
SPI 0 DMA	8	IVG9
SPI 1 DMA	9	IVG9
UART 0 Rx	10	IVG10
UART 0 Tx	11	IVG10
UART 1 Rx	12	IVG10
UART 1 Tx	13	IVG10
Timer 0	14	IVG11
Timer 1	15	IVG11
Timer 2	16	IVG11
GPIO Interrupt A	17	IVG12
GPIO Interrupt B	18	IVG12
Memory DMA	19	IVG13
Software Watchdog Timer	20	IVG13
Reserved	26–21	-
Software Interrupt 1	27	IVG14
Software Interrupt 2	28	IVG15

Event Control

The ADSP-21535 provides the user with a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each of the registers,as follows, is 16-bits wide, while each bit represents a particular event class:

- CEC Interrupt Latch Register (ILAT) – The ILAT register indicates when events have been latched. The appropriate bit is set when the processor has latched the event and cleared when the event has been accepted into the system. This register is updated automatically by the controller but may be read while in supervisor mode.
- CEC Interrupt Mask Register (IMASK) – The IMASK register controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event

is unmasked and will be processed by the system when asserted. A cleared bit in the IMASK register masks the event thereby preventing the processor from servicing the event even though the event may be latched in the ILAT register. This register may be read from or written to while in supervisor mode. (Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.)

- CEC Interrupt Pending Register (IPEND) – The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing three 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in Table 2.

- SIC Interrupt Mask Register (SIC_IMASK) – This register controls the masking and unmasking of each peripheral interrupt event. When a bit is set in the register, that peripheral event is unmasked and will be processed by the system when asserted. A cleared bit in the register masks the peripheral event thereby preventing the processor from servicing the event.
- SIC Interrupt Status Register (SIC_ISTAT) – As multiple peripherals can be mapped to a single event, this register allows the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt, a cleared bit indicates the peripheral is not asserting the event.
- SIC Interrupt Wakeup Enable Register (SIC_IWR) – By enabling the corresponding bit in this register, each peripheral can be configured to wake up the processor, should the processor be in a powered down mode when the event is generated. (For more information, see Dynamic Power Management on page 11.)

Because multiple interrupt sources can map to a single general-purpose interrupt, multiple pulse assertions can occur simultaneously, before or during interrupt processing for an interrupt event already detected on this interrupt input. The IPEND register contents are monitored by the SIC as the interrupt acknowledgement.

The appropriate ILAT register bit is set when an interrupt rising edge is detected (detection requires two processor clock cycles). The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC will recognize and queue the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the general-purpose interrupt to the IPEND output asserted is three processor clock cycles; however, the latency can be much higher, depending on the activity within and the mode of the processor.

DMA CONTROLLERS

The ADSP-21535 has multiple, independent DMA controllers that support automated data transfers with minimal overhead for the DSP core. DMA transfers can occur between the ADSP-21535's internal memories and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including the SDRAM controller, the asynchronous memory controller and the PCI bus interface. DMA-capable peripherals include the SPORTs, SPI ports, UARTs, and USB port. Each individual DMA-capable peripheral has at least one dedicated DMA channel. DMA to and from PCI is accomplished by the memory DMA channel.

To describe each DMA sequence, the DMA controller uses a set of parameters, called a descriptor block. When successive DMA sequences are needed, these descriptor blocks can be linked or chained together, so the completion of one DMA sequence auto-initiates and starts the next sequence. The descriptor blocks include full 32-bit addresses for the base pointers for source and destination enabling access to the entire ADSP-21535 address space.

In addition to the dedicated peripheral DMA channels, there is a separate memory DMA channel provided for transfers between the various memories of the ADSP-21535 system. This enables transfers of blocks of data between any of the memories including on-chip Level 2 memory, external SDRAM, ROM, SRAM and flash memory, and PCI address spaces with little processor intervention.

EXTERNAL MEMORY CONTROL

The External Bus Interface Unit (EBIU) on the ADSP-21535 provides a high performance, glueless interface to a wide variety of industry-standard memory devices. The controller is made up of two sections: the first is an SDRAM controller for connection of industry-standard synchronous DRAM devices and DIMMs, while the second is an asynchronous memory controller intended to interface to a variety of memory devices.

PC133 SDRAM Controller

The SDRAM controller provides an interface to up to four separate banks of industry-standard SDRAM devices or DIMMs, at speeds up to f_{SCLK}. Fully compliant with the PC133 SDRAM standard, each bank can be configured to contain between 16M bytes and 128M bytes of memory.

The controller maintains all of the banks as a contiguous address space so that the processor sees this as a single address space, even if different size devices are used in the different banks. This enables system designs that are delivered with an initial configuration that can be upgraded at a future time with either similar or different memories.

A set of programmable timing parameters is available to configure the SDRAM banks to support slower memory devices. The memory banks can be configured as either 32-bits wide for maximum performance and bandwidth or 16-bits wide for minimum device count and lower system cost.

All four banks share common SDRAM control signals and have their own bank select lines providing a completely glueless interface for most system configurations.

Asynchronous Controller
The asynchronous memory controller provides a configurable interface for up to four separate banks of memory or I/O devices. Each bank can be independently programmed with different timing parameters, enabling connection to a wide variety of memory devices including SRAM, ROM, and flash EPROM, as well as I/O devices that interface with standard memory control lines. Each bank occupies a 64M-byte window in the processor's address space but, if not fully populated, these are not made contiguous by the memory controller logic. The banks can also be configured as 16-bit wide or 32-bit wide buses for ease of interfacing to a range of memories and I/O devices tailored either to high performance or to low cost and power.

PCI INTERFACE
The ADSP-21535 provides a glueless logical and electrical, 33-Mhz, 3.3 V, 32-bit PCI (Peripheral Component Interconnect), Revision 2.2-compliant interface. The PCI interface is designed for a 3-volt signalling environment. The PCI interface provides a bus bridge function between the processor core and on-chip peripherals and an external PCI bus. The PCI interface of the ADSP-21535 supports two PCI functions, as follows

- A Host to PCI Bridge function, in which the ADSP-21535 resources (the processor core, internal and external memory, and the memory DMA controller) provide the necessary hardware components to emulate a host PC PCI interface, from the perspective of a PCI target device.
- A PCI Target function, in which an ADSP-21535 based intelligent peripheral can be designed to easily interface to a Revision 2.2-compliant PCI bus.

PCI Host Function
As the PCI host, the ADSP-21535 provides the necessary PCI host (platform) functions required to support and control a variety of off-the-shelf PCI I/O devices (e.g., Ethernet controllers, bus bridges, etc.) in a system in which the ADSP-21535 processor is the host.

Note that the Blackfin DSP architecture defines only memory space (no I/O or config address spaces). The three address spaces of PCI space (memory, IO, and configuration space) are mapped into the flat 32-bit memory space of the ADSP-21535. Because the PCI memory space is as large as the ADSP-21535 memory address space, a windowed approach is employed, with separate windows in the ADSP-21535 address space used for accessing the three PCI address spaces. Base address registers are provided so that these windows can be positioned to view any range in the PCI address spaces while they remain fixed in position in the ADSP-21535 processor's address range.

For devices on the PCI bus viewing the ADSP-21535's resources, several mapping registers are provided to enable resources to be viewed in the PCI address space. The ADSP-21535's external memory space, internal L2, and some I/O MMRs can be selectively enabled as memory spaces that devices on the PCI bus can use as targets for PCI memory transactions.

PCI Target Function
As a PCI target device, the PCI host processor can configure the ADSP-21535 subsystem during enumeration of the PCI bus system. Once configured, the ADSP-21535 subsystem acts as an intelligent I/O device. When configured as a target device, the PCI controller uses the memory DMA controller to perform DMA transfers as required by the PCI host.

USB DEVICE
The ADSP-21535 provides a USB 1.1- compliant device type interface to support direct connection to a host system. The USB core interface provides a flexible programmable environment with up to eight endpoints. Each endpoint can support all of the USB data types including Control, Bulk, Interrupt, and Isochronous. Each endpoint provides a memory-mapped buffer for transferring data to the application. The ADSP-21535 USB port has a dedicated DMA controller and interrupt input to minimize processor polling overhead and to enable asynchronous requests for CPU attention only when transfer management is required.

REAL-TIME CLOCK
The ADSP-21535 Real-Time Clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 KHz crystal external to the ADSP-21535. The RTC peripheral has dedicated power supply pins, so that it can remain powered up and clocked, even when the rest of the processor is in a low-power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, or day clock ticks, interrupt on programmable stopwatch countdown, or interrupt at a programmed alarm time.

The 32.768 KHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 6-bit second counter, a 6-bit minute counter, a 5-bit hours counter, and an 8-bit day counter.

When enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register. There are two alarms: The first alarm is for a time of day. The second alarm is for a day and time of that day.

The stopwatch function counts down from a programmed value, with one minute resolution. When the stopwatch is enabled and the counter underflows, an interrupt is generated.

Like the other peripherals, the RTC can wake up the ADSP-21535 processor from a low-power state upon generation of any interrupt.

Connect RTC pins XTALI and XTALO with external components, as shown in Figure 3.

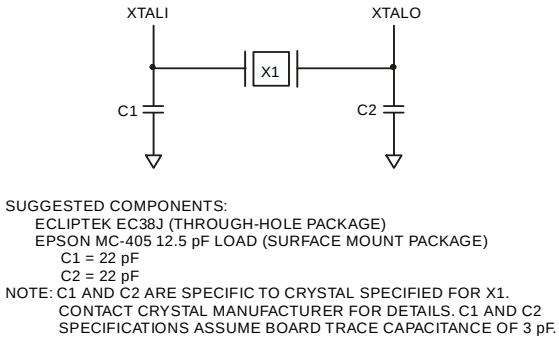


Figure 3. External Components for RTC

WATCHDOG TIMER

The ADSP-21535 includes a 32-bit timer, which can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, via generation of a hardware reset, non-maskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the timer control register, which is set only upon a watchdog generated reset.

The timer is clocked by the system clock (SCLK), at a maximum frequency of f_{SCLK}.

TIMERS

There are four programmable timer units in the ADSP-21535. Three general-purpose timers have an external pin that can be configured either as a Pulse Width Modulator (PWM) or timer output, as an input to clock the timer, or for measuring pulse widths of external events. Each of the three general-purpose timer units can be independently programmed as a PWM, internally or externally clocked timer, or pulse width counter.

The general-purpose timer units can be used in conjunction with the UARTs to measure the width of the pulses in the data stream to provide an auto-baud detect function for a serial channel.

The general-purpose timers can generate interrupts to the processor core providing periodic events for synchronization, either to the processor clock or to a count of external signals.

In addition to the three general-purpose programmable timers, a fourth timer is also provided. This extra timer is clocked by the internal processor clock (CCLK) and is typically used as a system tick clock for generation of operating system periodic interrupts.

SERIAL PORTS (SPORTS)

The ADSP-21535 incorporates two complete synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

- Bidirectional operation – Each SPORT has independent transmit and receive pins.
- Buffered (8-deep) transmit and receive ports – Each port has a data register for transferring data words to and from other DSP components and shift registers for shifting data in and out of the data registers.
- Clocking – Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from (f_{SCLK}/131070) Hz to (f_{SCLK}/2) Hz.
- Word length – Each SPORT supports serial data words from 3 to 16 bits in length transferred in a format of most significant bit first or least significant bit first.
- Framing – Each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two pulsewidths and early or late frame sync.
- Companding in hardware – Each SPORT can perform A-law or μ-law companding according to ITU recommendation G.711. Companding can be selected on the transmit and/or receive channel of the SPORT without additional latencies.
- DMA operations with single-cycle overhead – Each SPORT can automatically receive and transmit multiple buffers of memory data. The DSP can link or chain

sequences of DMA transfers between a SPORT and memory. The chained DMA can be dynamically allocated and updated through the descriptor blocks that set up the chain.

- Interrupts – Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability – Each SPORT supports 128 channels and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

SERIAL PERIPHERAL INTERFACE (SPI) PORTS

The ADSP-21535 has two SPI-compatible ports that enable the processor to communicate with multiple SPI-compatible devices.

The SPI interface uses three pins for transferring data: two data pins (Master Output-Slave Input, MOSIx, and Master Input-Slave Output, MISOx) and a clock pin (Serial Clock, SCKx). Two SPI chip select input pins ($\overline{\text{SPISSx}}$) let other SPI devices select the DSP, and fourteen SPI chip select output pins (SPIxSEL7–1) let the DSP select other SPI devices. The SPI select pins are reconfigured Programmable Flag pins. Using these pins, the SPI ports provide a full duplex, synchronous serial interface, which supports both master and slave modes and multimaster environments.

Each SPI port's baud rate and clock phase/polarities are programmable (see Figure 4), and each has an integrated DMA controller, configurable to support transmit or receive data streams. The SPI's DMA controller can only service unidirectional accesses at any given time.

$$SPI\ Clock\ Rate = \frac{f_{SCLK}}{2 \times SPIBAUD}$$

Figure 4. SPI Clock Rate Calculation

During transfers, the SPI ports simultaneously transmit and receive by serially shifting data in and out on their two serial data lines. The serial clock line synchronizes the shifting and sampling of data on the two serial data lines.

In master mode, the DSP performs the following sequence to set up and initiate SPI transfers:

1. Enables and configures the SPI port's operation (data size, and transfer format).
2. Selects the target SPI slave with an SPIxSELy output pin (reconfigured Programmable Flag pin).
3. Defines one or more TCBs in the DSP's memory space (optional in DMA mode only).
4. Enables the SPI DMA engine and specifies transfer direction (optional in DMA mode only).
5. In non-DMA mode only, reads or writes the SPI port receive or transmit data buffer.

   The SCKx line generates the programmed clock pulses for simultaneously shifting data out on MOSIx and shifting data in on MISOx. In DMA mode only, transfers continue until the SPI DMA word count transitions from 1 to 0.

In slave mode, the DSP performs the following sequence to set up the SPI port to receive data from a master transmitter:

1. Enables and configures the SPI slave port to match the operation parameters set up on the master (data size and transfer format) SPI transmitter.
2. Defines and generates a receive TCB in the DSP's memory space to interrupt at the end of the data transfer (optional in DMA mode only).
3. Enables the SPI DMA engine for a receive access (optional in DMA mode only).
4. Starts receiving the data on the appropriate SPI SCKx edges after receiving an SPI chip select on an $\overline{\text{SPISSx}}$ input pin (reconfigured Programmable Flag pin) from a master.

   In DMA mode only, reception continues until the SPI DMA word count transitions from 1 to 0. The DSP can continue, by queuing up the next command TCB.

A slave mode transmit operation is similar, except the DSP specifies the data buffer in memory from which to transmit data, generates and relinquishes control of the transmit TCB, and begins filling the SPI port's data buffer. If the SPI controller isn't ready on time to transmit, it can transmit a "zero" word.

UART PORT

The ADSP-21535 provides two full duplex Universal Asynchronous Receiver/Transmitter (UART) ports (UART0 and UART1) fully compatible with PC-standard UARTs. The UART ports provide a simplified UART interface to other peripherals or hosts, supporting full duplex, DMA supported, asynchronous transfers of serial data. Each UART port includes support for 5 to 8 data bits; 1 or 2 stop bits; and none, even, or odd parity. The UART ports support two modes of operation, as follows:

- PIO (Programmed I/O) – The processor sends or receives data by writing or reading I/O-mapped UATX or UARX registers, respectively. The data is double-buffered on both transmit and receive.
- DMA (Direct Memory Access) – The DMA controller transfers both transmit and receive data. This reduces the number and frequency of interrupts required to transfer data to and from memory. Each UART has two dedicated DMA channels, one for transmit and one for recieve. These DMA channels have lower priority than most DMA channels because of their relatively low service rates.

Each UART port's baud rate (see Figure 5), serial data format, error code generation and status, and interrupts are programmable:

- Supporting bit rates ranging from (f_{SCLK}/ 1048576) to (f_{SCLK}/16) bits per second.
- Supporting data formats from 7 to12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

$$UART\ Clock\ Rate = \frac{f_{SCLK}}{16 \times D}$$

Figure 5. UART Clock Rate Calculation[1]

[1]Where D = 1 to 65536

In conjunction with the general-purpose timer functions, autobaud detection is supported.

The capabilities of UART0 are further extended with support for the InfraRed Data Association (IrDA®) Serial InfraRed Physical Layer Link Specification (SIR) protocol.

PROGRAMMABLE FLAGS (PFX)

The ADSP-21535 has 16 bi-directional, general-purpose I/O, Programmable Flag (PF15–0) pins. The Programmable Flag pins have special functions for clock multiplier selection, SROM boot mode, and SPI port operation. For more information, see Serial Peripheral Interface (SPI) Ports on page 10 and Clock Signals on page 13. Each programmable flag can be individually controlled as follows by manipulation of the flag control, status, and interrupt registers:

- Flag Direction Control Register – Specifies the direction of each individual PFx pin as input or output.
- Flag Control and Status Registers – Rather than forcing the software to use a read-modify-write process to control the setting of individual flags, the ADSP-21535 employs a "write one to set" and "write one to clear" mechanism that allows any combination of individual flags to be set or cleared in a single instruction, without affecting the level of any other flags. Two control registers are provided, one register is written to in order to set flag values while another register is written to in order to clear flag values. Reading the flag status register allows software to interrogate the sense of the flags.
- Flag Interrupt Mask Registers – The two Flag Interrupt Mask Registers allow each individual PFx pin to function as an interrupt to the processor. Similar to the two Flag Control Registers that are used to set and clear individual flag values, one Flag Interrupt Mask Register sets bits to enable interrupt function, and the other Flag Interrupt Mask register clears bits to disable interrupt function. PFx pins defined as inputs can be configured to generate hardware interrupts, while output PFx pins can be configured to generate software interrupts.
- Flag Interrupt Sensitivity Registers – The two Flag Interrupt Sensitivity Registers specify whether individual PFx pins are level- or edge-sensitive and specify-if edge-sensitive-whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge-sensitivity.

DYNAMIC POWER MANAGEMENT

The ADSP-21535 provides four operating modes, each with a different performance/power-dissipation profile. In addition, Dynamic Power Management provides the control functions, with the appropriate external power regulation capability, to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the ADSP-21535 peripherals also reduces power dissipation. See Table 3 for a summary of the power settings for each mode.

Full On Operating Mode – Maximum Performance

In the Full On mode, the PLL is enabled, and is not bypassed, providing the maximum operational frequency. This is the normal execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

Active Operating Mode – Moderate Power Savings

In the Active mode, the PLL is enabled, but bypassed. The input clock (CLKIN) is used to generate the clocks for the processor core (CCLK) and peripherals (SCLK). When the PLL is bypassed, CCLK runs at one-half the CLKIN frequency. Significant power savings can be achieved with the processor running at one-half the CLKIN frequency. In this mode the PLL multiplication ratio can be changed by setting the appropriate values in the SSEL fields of the PLL control register (PLL_CTL).

When in the Active mode, system DMA access to appropriately configured L1 memory is supported.

Table 3. Operating Mode Power Settings

Mode	PLL	PLL Bypassed	Core Clock (CCLK)	System Clock (SCLK)
Full On	Enabled	No	Enabled	Enabled
Active	Enabled	Yes	Enabled	Enabled
Sleep	Enabled	Yes or No	Disabled	Enabled
Deep-Sleep	Disabled	–	Disabled	Disabled

Sleep Operating Mode – High Power Savings
The Sleep mode reduces power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK) however, continue to operate in this mode. Any interrupt, typically via some external event or RTC activity, will wake up the processor. When in the Sleep mode, assertion of any interrupt will cause the processor to sense the value of the bypass bit (BYPASS) in the PLL control register (PLL_CTL). If bypass is disabled, the processor will transition to the Full On mode. If bypass is enabled, the processor will transition to the Active mode.

When in the Sleep mode, system DMA access to L1 memory is not supported.

Deep-Sleep Operating Mode – Maximum Power Savings
The Deep-Sleep mode maximizes power savings by disabling the clocks to the processor core (CCLK) and to all synchronous systems (SCLK). Asynchronous systems, such as the RTC, may still be running but will not be able to access internal resources or external memory. This powered down mode can only be exited by assertion of the reset interrupt ($\overline{\text{RESET}}$) or by an asynchronous interrupt generated by the RTC. When in Deep-Sleep mode, assertion of $\overline{\text{RESET}}$ causes the processor to sense the value of the BYPASS pin. If bypass is disabled, the processor will transition to the Full On mode. If bypass is enabled, the processor will transition to the Active mode. When in Deep-Sleep mode, assertion of the RTC asynchronous interrupt causes the processor to transition to the Full On mode, regardless of the value of the BYPASS pin.

The DEEPSLEEP output is asserted in this mode.

Mode Transitions
The available mode transitions diagrammed in Figure 6 are accomplished either by the interrupt events described in the sections below or by programming the PLLCTL register with the appropriate values and then executing the PLL programming sequence.

This instruction sequence takes the processor to a known, idle state, with the interrupts disabled. Note that all DMA activity should be disabled during mode transitions.

Power Savings
As shown in Table 4, the ADSP-21535 supports five different power domains. The use of multiple power domains maximizes flexibility, while maintaining compliance with industry standards and conventions. By isolating the internal logic of the ADSP-21535 into its own power domain, separate from the PLL, RTC, PCI, and other I/O, the processor can take advantage of dynamic power management, without affecting the PLL, RTC, or other I/O devices.

Table 4. Power Domains

Power Domain	VDD Range
All internal logic, except PLL and RTC	V_{DDINT}
Analog PLL internal logic	V_{DDPLL}
RTC internal logic and crystal I/O	V_{DDRTC}
PCI I/O	$V_{DDPCIEXT}$
All other I/O	V_{DDEXT}

The power dissipated by a processor is largely a function of the clock frequency of the processor and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in power dissipation, while reducing the voltage by 25% reduces power dissipation by more than 40%. Further, these power savings are additive, in that if the clock frequency and power are both reduced the power savings are dramatic.

Dynamic Power Management allows both the processor's input voltage (V_{DDINT}) and clock frequency (f_{CLK}) to be dynamically controlled.

As explained above, the savings in power dissipation can be modeled by the following equation:

Power Dissipation Factor = $(f_{CCLKRED}/f_{CCLKNOM}) \times (V_{DDINTRED}/V_{DDINTNOM})^2$

where

- $f_{CCLKNOM}$ is the nominal core clock frequency (300 MHz)
- $f_{CCLKRED}$ is the reduced core clock frequency
- $V_{DDINTNOM}$ is the nominal internal supply voltage (1.5 V)
- $V_{DDINTRED}$ is the reduced internal supply voltage

As an example of how significant the power savings of Dynamic Power Management are, when both frequency and voltage are reduced, consider an example where the frequency is reduced from its nominal value to 50 MHz and the voltage is reduced from its nominal value to 1.2 V. At this reduced frequency and voltage, the processor dissipates about 10% of the power dissipated at nominal frequency and voltage.

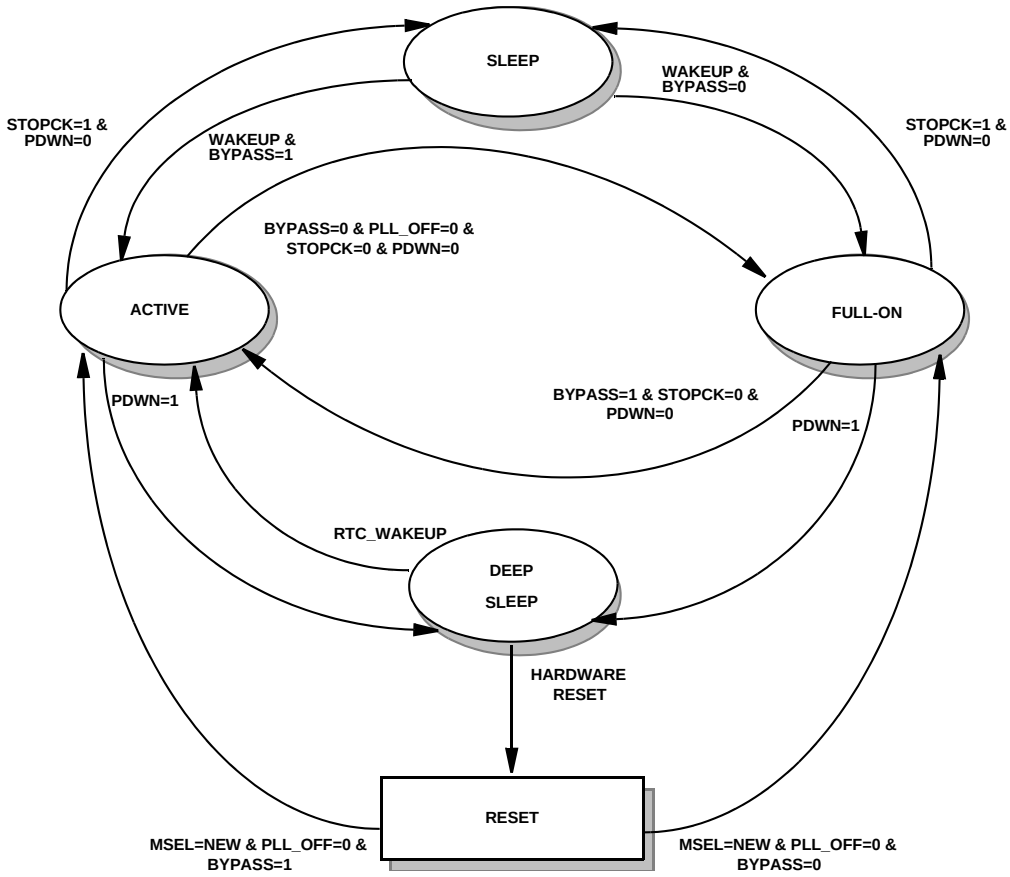


Figure 6. Mode Transitions

Peripheral Power Control

The ADSP-21535 provides additional power control capability by allowing dynamic scheduling of clock inputs to each of the peripherals. Clocking to each of the peripherals listed below can be enabled or disabled by appropriately setting the peripheral's control bit in the Peripheral Clock Enable Register (PLL_IOCK). The Peripheral Clock Enable Register allows individual control for each of the following peripherals:

- PCI
- EBIU controller
- Programmable flags
- MemDMA controller
- SPORT 0
- SPORT 1
- SPI 0
- SPI 1
- UART 0
- UART 1
- Timer 0, Timer 1, Timer 2
- USB CLK

CLOCK SIGNALS

The ADSP-21535 can be clocked by a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

If a buffered, shaped clock is used, this external clock connects to the DSP's CLKIN pin. CLKIN input cannot be halted, changed, or operated below the specified frequency during normal operation. This clock signal should be a TTL-compatible signal. The DSP provides a user-programmable 1x to 31x multiplication of the input clock, to support external to internal (DSP core) clock ratios. The MSEL6–0, BYPASS, and DF pins decide the PLL multiplication factor at reset. At runtime, the multiplication factor can be controlled in software. The combination of pullup and pull-down resistors in Figure 7 sets up a core clock ratio of 6:1, which, for example, produces a 150-MHz core clock from the 25-MHz input. For other clock multiplier settings, see the *ADSP-21535 DSP Hardware Reference*.

The peripheral clock is supplied to the CLKOUT_SCLK0 pin.

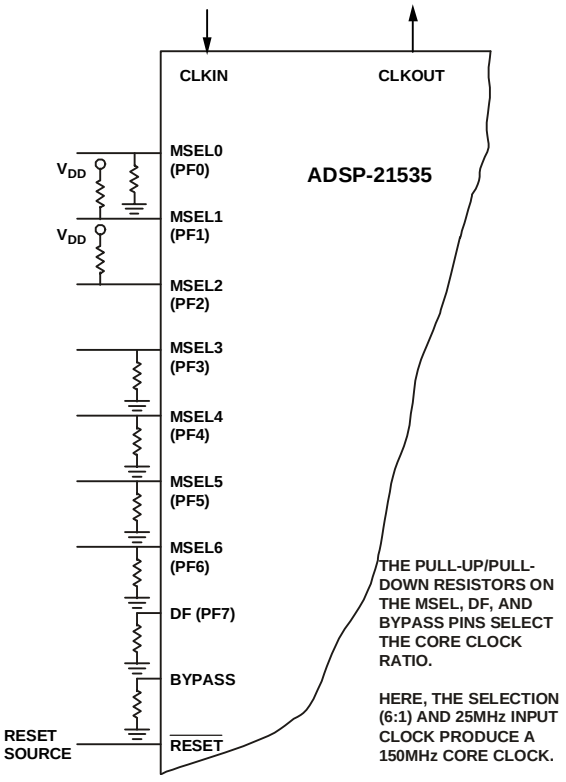


Figure 7. Clock Ratio Example

All on-chip peripherals operate at the rate set by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL pins. At run time the system clock frequency can be controlled in software by writing to the SSEL fields in the PLL control register (PLL_CTL). The values programmed into the SSEL fields define a divide ratio between the core clock (CCLK) and the system clock. Table 5 illustrates the system clock ratios.

Table 5. System Clock Ratios

Signal Name SSEL1–0	Divider Ratio CCLK/ SCLK	Example Frequency Ratios (MHz)	
		CCLK	SCLK
00	2:1	266	133
01	2.5:1	275	110
10	3:1	300	100
11	4:1	300	75

The maximum frequency of the system clock is f_{SCLK}. Note that the divisor ratio must be chosen to limit the system clock frequency to its maximum of f_{SCLK}. The reset value of the SSEL1–0 is determined by sampling the Programmable Flag input pins (PF9–8) during reset. The SSEL value can be changed dynamically by writing the appropriate values to the PLL control register (PLL_CTL), as described in the *ADSP-21535 DSP Hardware Reference*.

BOOTING MODES

The ADSP-21535 has three mechanisms (listed in Table 6) for automatically loading internal L2 memory after a reset. A fourth mode is provided to execute from external memory, bypassing the boot sequence.

Table 6. Booting Modes

BMODE2–0	Description
000	Execute from 16-bit external memory (Bypass Boot ROM)
001	Boot from 8-bit flash
010	Boot from SPI0 serial ROM (8-bit address range)
011	Boot from SPI0 serial ROM (16-bit address range)
100–111	Reserved

The BMODE pins of the Reset Configuration Register, sampled during power on resets and software initiated resets, implement the following modes:

- Execute from 16-bit external memory – Execution starts from address 0x2000000 with 16-bit packing. The boot ROM is bypassed in this mode.
- Boot from 8-bit external flash memory – The 8-bit flash boot routine located in boot ROM memory space is set up using asynchronous Memory Bank 0. All configuration settings are set for the slowest device possible (3-cycle hold time; 15-cycle R/W access times; 4-cycle setup).
- Boot from SPI serial EEPROM (8-bit addressable) – The SPI0 uses PF10 output pin to select a single SPI EPROM device, submits a read command at address 0x00, and begins clocking data into the beginning of L2 memory. An 8-bit addressable SPI-compatible EPROM must be used.
- Boot from SPI serial EEPROM (16-bit addressable) – The SPI0 uses PF10 output pin to select a single SPI EPROM device, submits a read command at address 0x0000, and begins clocking data into the beginning of L2 memory. A 16-bit addressable SPI-compatible EPROM must be used.

For each of the boot modes described above, a four-byte value is first read from the memory device. This value is used to specify a subsequent number of bytes to be read into the beginning of L2 memory space. Once each of the loads is complete, the processor jumps to the beginning of L2 space and begins execution.

In addition, the Reset Configuration Register can be set by application code to bypass the normal boot sequence during a software reset. For this case, the processor jumps directly to the beginning of L2 memory space.

To augment the boot modes described above, a secondary software loader is provided that adds additional booting mechanisms. This secondary loader provides the capability to boot from 16-bit flash memory, fast flash, variable baud rate, etc.

INSTRUCTION SET DESCRIPTION

The Blackfin DSP family assembly language instruction set employs an algebraic syntax that was designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the DSP core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both a user (algorithm/application code) and a supervisor (O/S kernel, device drivers, debuggers, ISRs) mode of operations, allowing multiple levels of access to core DSP resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/CPU features are optimized for both 8-bit and 16-bit operations.
- A super-pipelined multi-issue load/store modified-Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU + two load/store + two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4G-byte memory space providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data-types; and separate user and kernel stack pointers.
- Code density enhancements, which include intermixing of 16- and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded as 16-bits.

DEVELOPMENT TOOLS

The ADSP-21535 is supported with a complete set of software and hardware development tools, including Analog Devices' emulators and the VisualDSP++™ development environment. The same emulator hardware that supports other Analog Devices JTAG DSPs, also fully emulates the ADSP-21535.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy-to-use assembler that is based on an algebraic syntax, an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ run-time library that includes DSP and mathematical functions. A key point for these tools is C/C++ code efficiency. The compiler has been developed for efficient translation of C/C++ code to Blackfin DSP assembly. The Blackfin 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 break-points
- 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 development tools, including Color Syntax Highlighting in the VisualDSP++ editor. These capabilities permit programmers to:

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

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

Analog Devices' DSP emulators use the IEEE 1149.1 JTAG test access port of the ADSP-21535 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 Blackfin DSP family. Hardware tools include the ADSP-21535 EZ-KIT Lite™ standalone evaluation/development cards. Third Party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

Designing an Emulator-Compatible DSP Board (Target)

The Analog Devices family of emulators are tools that every DSP developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on the ADSP-21535. The emulator uses the TAP to access the internal features of the DSP, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The DSP must be halted to send data and commands, but once an operation has been completed by the emulator, the DSP system is set running at full speed with no impact on system timing.

To use these emulators, the target's design must include the interface between an Analog Devices' JTAG DSP and the emulation header on a custom DSP target board.

Target Board Header

The emulator interface to an Analog Devices' JTAG DSP is a 14-pin header, as shown in Figure 8. The customer must supply this header on the target board in order to communicate with the emulator. The interface consists of a standard dual row 0.025" square post header, set on 0.1" × 0.1" spacing, with a minimum post length of 0.235". Pin 3 is the key position used to prevent the pod from being inserted backwards. This pin must be clipped on the target board.

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

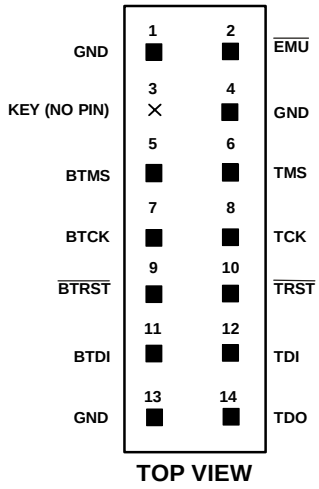


Figure 8. JTAG Target Board Connector for JTAG Equipped Analog Devices DSP (Jumpers in Place)

As can be seen in Figure 8, there are two sets of signals on the header. There are the standard JTAG signals TMS, TCK, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{EMU}}$ used for emulation purposes (via an emulator). There are also secondary JTAG signals BTMS, BTCK, BTDI, and $\overline{\text{BTRST}}$ that are optionally used for board-level (boundary scan) testing.

When the emulator is not connected to this header, place jumpers across BTMS, BTCK, $\overline{\text{BTRST}}$, and BTDI as shown in Figure 9. This holds the JTAG signals in the correct state to allow the DSP to run free. Remove all the jumpers when connecting the emulator to the JTAG header.

JTAG Emulator Pod Connector

Figure 10 details the dimensions of the JTAG pod connector at the 14-pin target end. Figure 11 displays the keep-out area for a target board header. The keep-out area allows the pod connector to properly seat onto the target board header.

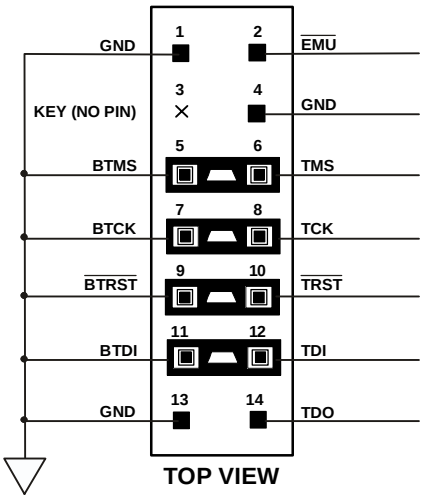


Figure 9. JTAG Target Board Connector with No Local Boundary Scan

This board area should contain no components (chips, resistors, capacitors, etc.). The dimensions are referenced to the center of the 0.25" square post pin.

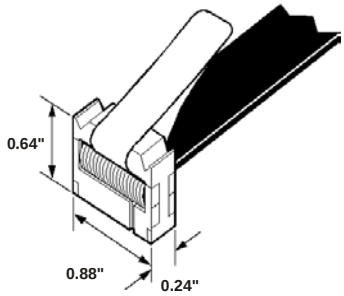


Figure 10. JTAG Pod Connector Dimensions

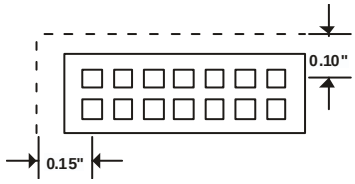


Figure 11. JTAG Pod Connector Keep-Out Area

Design-for-Emulation Circuit Information

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

ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-21535 architecture and functionality. For detailed information on the Blackfin DSP Family core architecture and instruction set, refer to the *ADSP-21535 Hardware Reference* and the *Blackfin DSP Family Instruction Set Reference*.

PIN DESCRIPTIONS

ADSP-21535 pin definitions are listed in Table 7. The following pins are asynchronous: ARDY, PF15–0, USB_CLK, NMI, TRST, RESET, PCI_CLK, XTALI, XTALO.

The following symbols appear in the Type column of Table 7: I = Input, O = Output, T = Three-State, P = Power, and G = Ground.

Table 7. Pin Descriptions

Pin	Type	Function
ADDR25–2	O/T	External address bus.
DATA31–0[1]	I/O/T	External data bus.
$\overline{\text{ABE}}$ /SDQM3–0	O/T	Asynchronous memory byte enables, SDRAM data masks.
$\overline{\text{AMS3–0}}$	O/T	Chip selects for asynchronous memories.
ARDY[1, 2]	I	Acknowledge signal for asynchronous memories.
$\overline{\text{AOE}}$	O/T	Memory output enable for asynchronous memories.
$\overline{\text{ARE}}$	O	Read enable for asynchronous memories.
$\overline{\text{AWE}}$	O	Write enable for asynchronous memories.
CLKOUT /SCLK1	O	SDRAM clock output pin. Same frequency and timing as SCLK0. Provided to reduce capacitance loading on SCLK0. Connect to SDRAM's CK pin.
SCLK0	O	SDRAM clock output pin 0. Switches at system clock frequency. Connect to the SDRAM's CK pin.
SCKE	O/T	SDRAM clock enable pin. Connect to SDRAM's CKE pin.
SA10	O/T	SDRAM A10 pin. SDRAM interface uses this pin to retain control of the SDRAM device during host bus requests. Connect to SDRAM's A10 pin.
$\overline{\text{SRAS}}$	O/T	SDRAM row address strobe pin. Connect to SDRAM's RAS pin.
$\overline{\text{SCAS}}$	O/T	SDRAM column address select pin. Connect to SDRAM's CAS pin.
$\overline{\text{SWE}}$	O/T	SDRAM write enable pin. Connect to SDRAM's WE or W buffer pin.
$\overline{\text{SMS3–0}}$	O/T	Memory select pin of external memory bank configured for SDRAM. Connect to SDRAM's chip select pin.
TMR0[2]	I/O/T	Timer 0 pin. Functions as an output pin in PWMOUT mode and as an input pin in WIDTH_CNT and EXT_CLK modes.
TMR1[2]	I/O/T	Timer 1 pin. Functions as an output pin in PWMOUT mode and as an input pin in WIDTH_CNT and EXT_CLK modes.
TMR2[2]	I/O/T	Timer 2 pin. Functions as an output pin in PWMOUT mode and as an input pin in WIDTH_CNT and EXT_CLK modes.
PF15 /$\overline{\text{SPI1SEL7}}$[2]	I/O/T	Programmable flag pin. SPI output select pin.
PF14 /$\overline{\text{SPI0SEL7}}$[2]	I/O/T	Programmable flag pin. SPI output select pin.
PF13 /$\overline{\text{SPI1SEL6}}$[2]	I/O/T	Programmable flag pin. SPI output select pin.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
PF12 /SPI0SEL6[2]	I/O/T	Programmable flag pin. SPI output select pin.
PF11 /SPI1SEL5[2]	I/O/T	Programmable flag pin. SPI output select pin.
PF10 /SPI0SEL5[2]	I/O/T	Programmable flag pin. SPI output select pin (used during SPI boot).
PF9 /SPI1SEL4 /SSEL1[3]	I/O	Programmable flag pin. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF8 /SPI0SEL4 /SSEL0[3]	I/O	Programmable flag pin. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF7 /SPI1SEL3 /DF[3]	I/O	Programmable flag pin. SPI output select pin.Sensed for configuration state during hardware reset, used to configure the PLL. DF=1 is for high frequency clock and divides the input clock by 2. DF=0 passes input clock directly to PLL phase detector.
PF6 /SPI0SEL3 /MSEL6[3]	I/O	Programmable flag pin. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF5 /SPI1SEL2 /MSEL5[3]	I/O	Programmable flag pin. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF4 /SPI0SEL2 /MSEL4[3]	I/O	Programmable flag pin. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF3 /SPI1SEL1 /MSEL3[3]	I/O	Programmable flag pin. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF2 /SPI0SEL1 /MSEL2[3]	I/O	Programmable flag pin. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF1 /SPISS1 /MSEL1[3]	I/O	Programmable flag pin. SPI slave select input pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF0 /SPISS0 /MSEL0[3]	I/O	Programmable flag pin. SPI slave select input pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
RSCLK0[2]	I/O/T	Receive serial clock for SPORT0.
RFS0[2]	I/O/T	Receive frame synchronization for SPORT0.
DR0[3]	I	Serial data receive for SPORT0.
TSCLK0[2]	I/O/T	Transmit serial clock for SPORT0.
TFS0[2]	I/O/T	Transmit frame synchronization for SPORT0.
DT0	O	Serial data transmit for SPORT0.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
RSCLK1[2]	I/O/T	Receive serial clock for SPORT1.
RFS1[2]	I/O/T	Receive frame synchronization for SPORT1.
DR1[3]	I	Serial data receive for SPORT1.
TSCLK1[2]	I/O/T	Transmit serial clock for SPORT1.
TFS1[2]	I/O/T	Transmit frame synchronization for SPORT1.
DT1	O	Serial data transmit for SPORT1.
MOSI0[2]	I/O	Master out slave in pin for SPI0. Supplies the output data from the master device and receives the input data to a slave device.
MISO0[2]	I/O	Master in slave out pin for SPI0. Supplies the output data from the slave device and receives the input data to the master device.
SCK0[4]	I/O	Clock line for SPI0. Master device output clock signal. Slave device input clock signal.
MOSI1[2]	I/O	Master out slave in pin for SPI1. Supplies the output data from the master device and receives the input data to a slave device.
MISO1[2]	I/O	Master in slave out pin for SPI1. Supplies the output data from the slave device and receives the input data to the master device.
SCK1[4]	I/O	Clock line for SPI1. Master device output clock signal. Slave device input clock signal.
RX0[3]	I	UART0 receive pin.
TX0	O	UART0 transmit pin.
RX1[3]	I	UART1 receive pin.
TX1	O	UART1 transmit pin.
USB_CLK[4]	I	USB clock.
XVER_DATA[4]	I	Single ended receive data output from USB transceiver to the USBD module.
DPLS[4]	I	Differential D+ receive data output from the USB transceiver to the UBD module.
DMNS[4]	I	Differential D- receive data output from the USB transceiver to the USBD module.
TXDPLS	O	Transmitted D+ from the USBD module to the USB transceiver.
TXDMNS	O	Transmitted D- from the USBD module to the USB transceiver.
$\overline{\text{TXEN}}$	O	Transmit enable from the USBD module to the USB transceiver.
SUSPEND	O	Suspend mode enable output from the USBD module to the USB transceiver. This signal can also be routed internally by the SoC to support low power operations.
NMI[4]	I	Non-maskable interrupt.
TCK[2]	I	JTAG clock.
TDO	O/T	JTAG serial data out.
TDI[2]	I	JTAG serial data in.
TMS[2]	I	Test mode select.
$\overline{\text{TRST}}$[4]	I	JTAG reset.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
$\overline{\text{RESET}}$	I	When this pin is asserted to logic zero level for at least 10 CLKIN cycles, a hardware reset is initiated. The minimum pulse width for power-on reset is 40 µsec.
CLKIN1	I	Clock in.
BYPASS	I	Dedicated mode pin. May be permanently strapped to VDD or VSS. Bypasses the on-chip PLL.
DEEPSLEEP	O	Denotes that the Blackfin DSP Core is in Deep-Sleep mode.
BMODE2–0	I	Dedicated mode pin. May be permanently strapped to VDD or VSS. Configures the boot mode that is employed following hardware reset or software reset.
PCI_AD31–0[2]	I/O/T	PCI address and data bus.
$\overline{\text{PCI\_CBE3–0}}$[2]	I/O/T	PCI byte enables.
$\overline{\text{PCI\_FRAME}}$[2]	I/O/T	PCI frame signal. Used by PCI initiators for signalling the beginning and end of a PCI transaction.
$\overline{\text{PCI\_IRDY}}$[2]	I/O/T	PCI initiator ready signal.
$\overline{\text{PCI\_TRDY}}$[2]	I/O/T	PCI target ready signal.
PCI_DEVSEL[2]	I/O/T	PCI device select signal. Asserted by targets of PCI transactions to claim the transaction.
$\overline{\text{PCI\_STOP}}$[2]	I/O/T	PCI stop signal.
$\overline{\text{PCI\_PERR}}$[2]	I/O/T	PCI parity error signal.
PCI_PAR[2]	I/O/T	PCI parity signal.
$\overline{\text{PCI\_REQ}}$	O	PCI request signal. Used for requesting the use of the PCI bus.
$\overline{\text{PCI\_SERR}}$[2]	I/O/T	PCI system error signal. Requires a pullup on the system board.
$\overline{\text{PCI\_RST}}$[2]	I/O/T	PCI reset signal.
$\overline{\text{PCI\_GNT}}$[2]	I	PCI grant signal. Used for granting access to the PCI bus.
$\overline{\text{PCI\_IDSEL}}$[4]	I	PCI initialization device select signal. Individual device selects for targets of PCI configuration transactions.
$\overline{\text{PCI\_LOCK}}$[2]	I	PCI lock signal. Used to lock a target or the entire PCI bus for use by the master that asserts the lock.
PCI_CLK[4]	I	PCI clock.
$\overline{\text{PCI\_INTA}}$[2]	I/O/T	PCI interrupt A line on PCI bus. Asserted by the ADSP-21535 as a device to signal an interrupt to the system processor. Monitored by the ADSP-21535 when acting as the system processor.
$\overline{\text{PCI\_INTB}}$[2]	I	PCI interrupt B line. Monitored by ADSP-21535 when acting as the system processor.
$\overline{\text{PCI\_INTC}}$[2]	I	PCI interrupt C line. Monitored by the ADSP-21535 when acting as the system processor.
$\overline{\text{PCI\_INTD}}$[2]	I	PCI interrupt D line. Monitored by the ADSP-21535 when acting as the system processor.
XTALI	I	Real-Time Clock oscillator input.
XTALO	O	Real-Time Clock oscillator output.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
$\overline{EMU}$	O	Emulator acknowledge, open drain. Must be connected to the ADSP-21535 emulator target board connector only.
V_{DDPLL}	P	PLL power supply (1.5 V nominal).
V_{DDRTC}	P	Real-Time Clock power supply (3.3 V nominal).
V_{DDEXT}	P	I/O (except PCI) power supply (3.3 V nominal).
$V_{DDPCIEXT}$	P	PCI I/O power supply (3.3 V nominal).
V_{DDINT}	P	Internal power supply (1.5 V nominal).
GND	G	Power supply return.

[1] Pin has a logic-level hold circuit that prevents the input from floating internally.
[2] Pull pin high, if not used.
[3] Pull pin high or low, if not used.
[4] Pull pin low, if not used.

ADSP-21535—SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter [1]	K Grade Parameter	Min	Nominal	Max	Unit
V_{DDINT}	Internal (Core) Supply Voltage	0.86	1.5	1.575	V
V_{DDEXT}	External (I/O) Supply Voltage	2.5	3.3	3.45	V
V_{DDPLL}	PLL Power Supply Voltage	1.425	1.5	1.575	V
V_{DDRTC}	Real Time Clock Power Supply Voltage	2.60	3.3	3.45	V
$V_{DDPCIEXT}$	PCI I/O Power Supply Voltage	3.15	3.3	3.45	V
V_{IH}	High Level Input Voltage[2], @ V_{DDEXT}=max	2.0		V_{DDEXT}+0.5	V
V_{IL}	Low Level Input Voltage[2], @ V_{DDEXT}=min	–0.3		0.6	V
V_{IHPCI}	High Level Input Voltage[3], @ V_{DDEXT}=max	0.5 $V_{DDPCIEXT}$		$V_{DDPCIEXT}$+0.5	V
V_{ILPCI}	Low Level Input Voltage[3], @ V_{DDINT}=min	–0.5		0.3 $V_{DDPCIEXT}$	V
T_{CASE}	Case Operating Temperature	0		85	°C

[1] Specifications subject to change without notice.
[2] Applies to input and bidirectional pins, except PCI.
[3] Applies to PCI input and bidirectional pins: PCI_AD31–0, $\overline{\text{PCI\_CBE3–0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, PCI_DEVSEL, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, PCI_PAR, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_GNT}}$, $\overline{\text{PCI\_IDSEL}}$, $\overline{\text{PCI\_LOCK}}$, PCI_CLK, $\overline{\text{PCI\_INTA}}$, $\overline{\text{PCI\_INTB}}$, $\overline{\text{PCI\_INTC}}$, $\overline{\text{PCI\_INTD}}$.

ELECTRICAL CHARACTERISTICS

Parameter[1]		Test Conditions	Min	Max	Unit
V_{OH}	High Level Output Voltage[2]	@ V_{DDEXT}=min, I_{OH} = –0.5 mA	2.4		V
V_{OL}	Low Level Output Voltage[2]	@ V_{DDEXT}=min, I_{OL} = 2.0 mA		0.4	V
V_{OHPCI}	PCI High Level Output Voltage[3]	@ V_{DDEXT}=min, I_{OH} = –0.5 mA	0.9 $V_{DDPCIEXT}$		V
V_{OLPCI}	PCI Low Level Output Voltage[3]	@ V_{DDEXT}=min, I_{OL} = 1.5mA		0.1 $V_{DDPCIEXT}$	V
I_{IH}	High Level Input Current[4]	@ V_{DDEXT}=max, V_{IN} = V_{DD} max		TBD	µA
I_{IL}	Low Level Input Current[4]	@ V_{DDEXT}=max, V_{IN} = 0 V		TBD	µA
I_{OZH}	Three-State Leakage Current[5]	@ V_{DDEXT}= max, V_{IN} = V_{DD} max		TBD	µA
I_{OZL}	Three-State Leakage Current[5]	@ V_{DDEXT}= max, V_{IN} = 0 V		TBD	µA
C_{IN}	Input Capacitance[6, 7]	f_{IN} = 1 MHz, T_{CASE}= 25°C, V_{IN} = 2.5 V		TBD	pF

[1] Specifications subject to change without notice.
[2] Applies to output and bidirectional pins, except PCI.
[3] Applies to PCI output and bidirectional pins: PCI_AD31–0, $\overline{\text{PCI\_CBE3–0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, PCI_DEVSEL, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, PCI_PAR, $\overline{\text{PCI\_REQ}}$, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_INTA}}$.
[4] Applies to input pins.
[5] Applies to three-statable pins.
[6] Applies to all signal pins.
[7] Guaranteed, but not tested.

ABSOLUTE MAXIMUM RATINGS

Internal (Core) Supply Voltage (V_{DDINT})[1] . . −0.3 V to +1.8 V
External (I/O) Supply Voltage (V_{DDEXT}) . . . −0.3 V to +4.0 V
Input Voltage. . . . . −0.5 V to V_{DDEXT} +0.5 V
Output Voltage Swing . . . . . −0.5 V to V_{DDEXT} +0.5 V
Load Capacitance . . . . . 200 pF
Core Clock . . . . . 300 MHz
Peripheral Clock (SCLK) . . . . . 133 MHz
Storage Temperature Range. . . . . −65°C to +150°C
Lead Temperature (5 seconds). . . . . 185°C

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

ESD SENSITIVITY

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-21535 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high-energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

TIMING SPECIFICATIONS

Table 8 and Table 9 describe the timing requirements for the ADSP-21535 clocks. Take care in selecting MSEL and SSEL ratios so as not to exceed the maximum core clock and system clock operating frequencies, as described in *ABSOLUTE MAXIMUM RATINGS*. Table 9 describes Phase-Locked Loop operating conditions.

Table 8. Core and System Clock Requirements

Parameter	Description	Min	Max	Unit
$t_{CCLK1.5}$	Core Cycle Period (V_{DDINT} = 1.5 V–5%)	3.3	TBD	ns
$t_{CCLK1.4}$	Core Cycle Period (V_{DDINT} = 1.4 V–5%)	TBD	TBD	ns
$t_{CCLK1.3}$	Core Cycle Period (V_{DDINT} = 1.3 V–5%)	TBD	TBD	ns
$t_{CCLK1.2}$	Core Cycle Period (V_{DDINT} = 1.2 V–5%)	TBD	TBD	ns
$t_{CCLK1.1}$	Core Cycle Period (V_{DDINT} = 1.1 V–5%)	TBD	TBD	ns
$t_{CCLK1.0}$	Core Cycle Period (V_{DDINT} = 1.0 V–5%)	TBD	TBD	ns
$t_{CCLK0.9}$	Core Cycle Period (V_{DDINT} = 0.9 V–5%)	TBD	TBD	ns
f_{CCLKNN}	Core Clock Frequency at t_{CCLKNN}		$1/t_{CCLKNN}$	Hz
t_{SCLK}	System Clock Period	Max. of (7.5 or $t_{CCLKNN} \times 2$)		ns
f_{SCLK}	System Clock Frequency		$1/t_{SCLK}$	Hz

Table 9. Phase-Locked Loop Operating Conditions

Parameter	Min	Nominal	Max	Unit
Operating Voltage	1.425	1.5	1.575	V
Jitter, Rising Edge To Rising Edge, Per Output			120	ps
Jitter, Rising Edge To Falling Edge, Per Output			60	ps

Table 9. Phase-Locked Loop Operating Conditions (Continued)

Parameter	Min	Nominal	Max	Unit
Skew, Rising Edge To Rising Edge, Any Two outputs			120	ps
Voltage Controlled Oscillator (VCO) Frequency	40		400	MHz
V_{DDPLL} induced jitter			1	ps/mV

Clock and Reset Timing

Table 10 and Figure 12 describe clock and reset operations. Per ABSOLUTE MAXIMUM RATINGS on page 24, combinations of CLKIN and clock multipliers must not select core/peripheral clocks in excess of 300/133 MHz.

Table 10. Clock and Reset Timing

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{CKIN}	CLKIN Period	30.0	100.0	ns
t_{CKINL}	CLKIN Low Pulse[1]	10.0		ns
t_{CKINH}	CLKIN High Pulse[1]	10.0		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulsewidth Low[2]	$11 \times t_{CKIN}$		ns
t_{PFD}	Delay from $\overline{\text{RESET}}$ Asserted to PFx I/O Terminated[3]		TBD	ns
t_{MSD}	Delay from $\overline{\text{RESET}}$ Asserted to MSELx and DF Valid[4]		TBD	ns
t_{MSS}	MSELx/DF/BYPASS Stable Setup Before $\overline{\text{RESET}}$ Deasserted[5]	TBD		ns
t_{MSH}	MSELx/DF/BYPASS Stable Hold After $\overline{\text{RESET}}$ Deasserted	TBD		ns
Switching Characteristics				
t_{SCLKD}	CLKOUT Delay from CLKIN	TBD	TBD	ns
t_{SCLK}	CLKOUT Period[6]	7.5		ns

[1] Applies to bypass mode and non-bypass mode.
[2] Applies after power-up sequence is complete. At power-up, the processor's internal phase-locked loop requires no more than 2000 CLKIN cycles, while $\overline{\text{RESET}}$ is asserted, assuming stable power supplies and CLKIN (not including start-up time of external clock oscillator).
[3] From this point, the MSELx and DF values begin stabilizing to a valid state.
[4] MSELx and DF values can change from this point, but the values must be valid.
[5] MSELx and DF values must be held from this time, until the hold time expires.
[6] The figure below shows a ×2 ratio between t_{CKIN} and t_{SCLK}, but the ratio has many programmable options. For more information, see the System Design chapter of the *ADSP-21535 DSP Hardware Reference*.

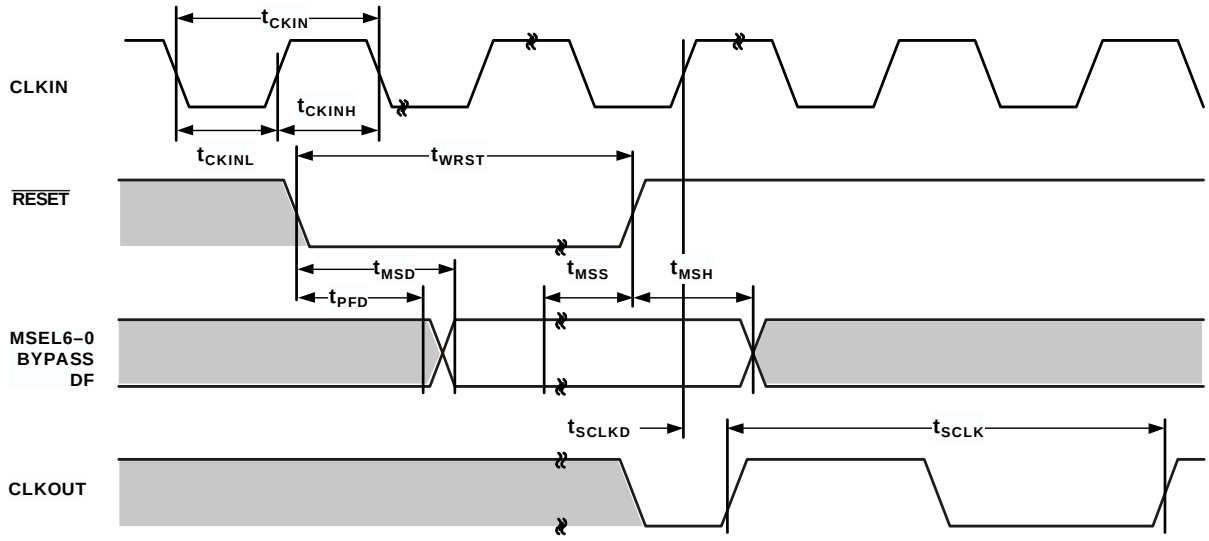


Figure 12. Clock and Reset Timing

Programmable Flags Cycle Timing

Table 11 and Figure 13 describe programmable flag operations.

Table 11. Programmable Flags Cycle Timing

Parameter	Description	Min	Max	Unit
Switching Characteristic				
t_{DFO}	Flag output delay with respect to SCLK		6	ns
t_{HFO}	Flag output hold after SCLK high	TBD	TBD	ns
Timing Requirement				
t_{HFI}	Flag input hold is asynchronous	3		ns

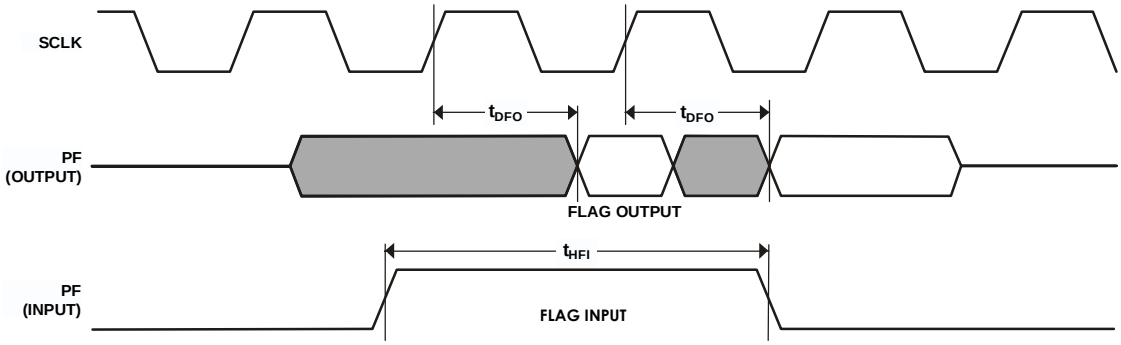


Figure 13. Programmable Flags Cycle Timing

Timer PWM_OUT Cycle Timing

Table 12 and Figure 14 describe timer expired operations. The input signal is asynchronous in "width capture mode" and has an absolute maximum input frequency of TBD MHz.

Table 12. Timer PWM_OUT Cycle Timing

Parameter	Description	Min	Max	Unit
Switching Characteristic				
t_{HTO}	Timer Pulsewidth Output[1]	7.5	$(2^{32}–1)$ cycles	ns

[1]The minimum time for t_{HTO} is one cycle, and the maximum time for t_{HTO} equals $(2^{32}–1)$ cycles.

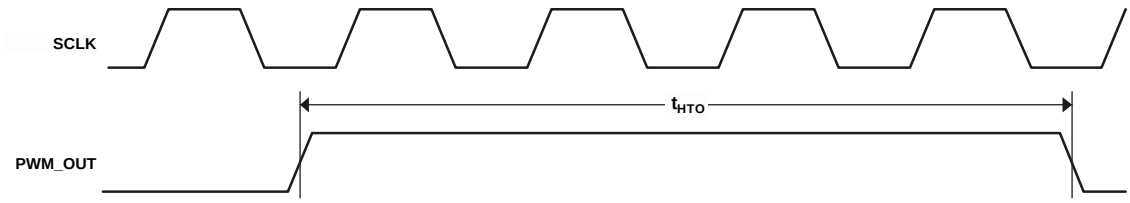


Figure 14. Timer PWM_OUT Cycle Timing

Asynchronous Memory Write Cycle Timing

Table 13. Asynchronous Memory Write Cycle Timing

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SARDY}	ARDY Setup Before CLKOUT	5.5		ns
t_{HARDY}	ARDY Hold After CLKOUT	0.0		ns
t_{DDAT}	DATA31–0 Disable After CLKOUT		6.0	ns
t_{ENDAT}	DATA31–0 Enable After CLKOUT	1.0		ns
Switching Characteristic				
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT [1]	0.8		ns

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE3-0}$, ADDR25–2, DATA31–0, $\overline{AOE}$, $\overline{AWE}$.

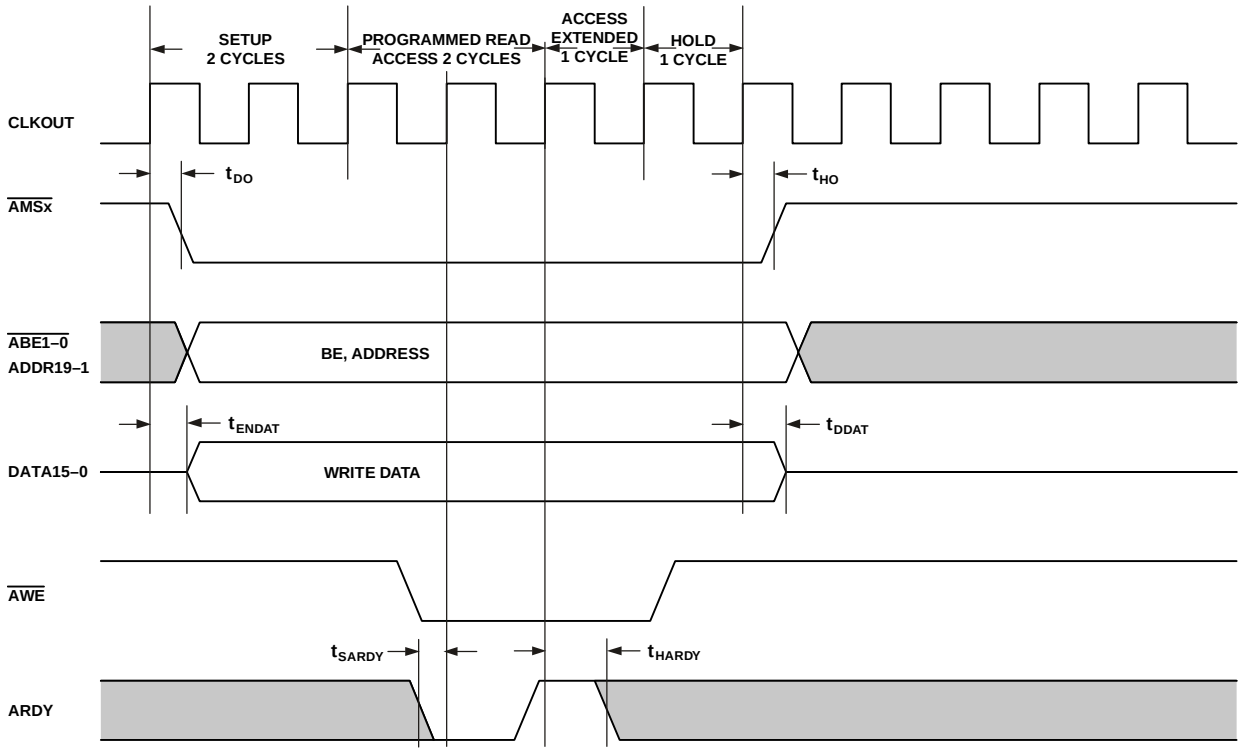


Figure 15. Asynchronous Memory Write Cycle Timing

Asynchronous Memory Read Cycle Timing

Table 14. Asynchronous Memory Read Cycle Timing

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SDAT}	DATA31–0 Setup Before CLKOUT	2.1		ns
t_{HDAT}	DATA31–0 Hold After CLKOUT	0.8		ns
t_{SARDY}	ARDY Setup Before CLKOUT	5.5		ns
t_{HARDY}	ARDY Hold After CLKOUT	0.0		ns
Switching Characteristic				
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT [1]	0.8		ns

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE3-0}$, ADDR25–2, $\overline{AOE}$, $\overline{ARE}$.

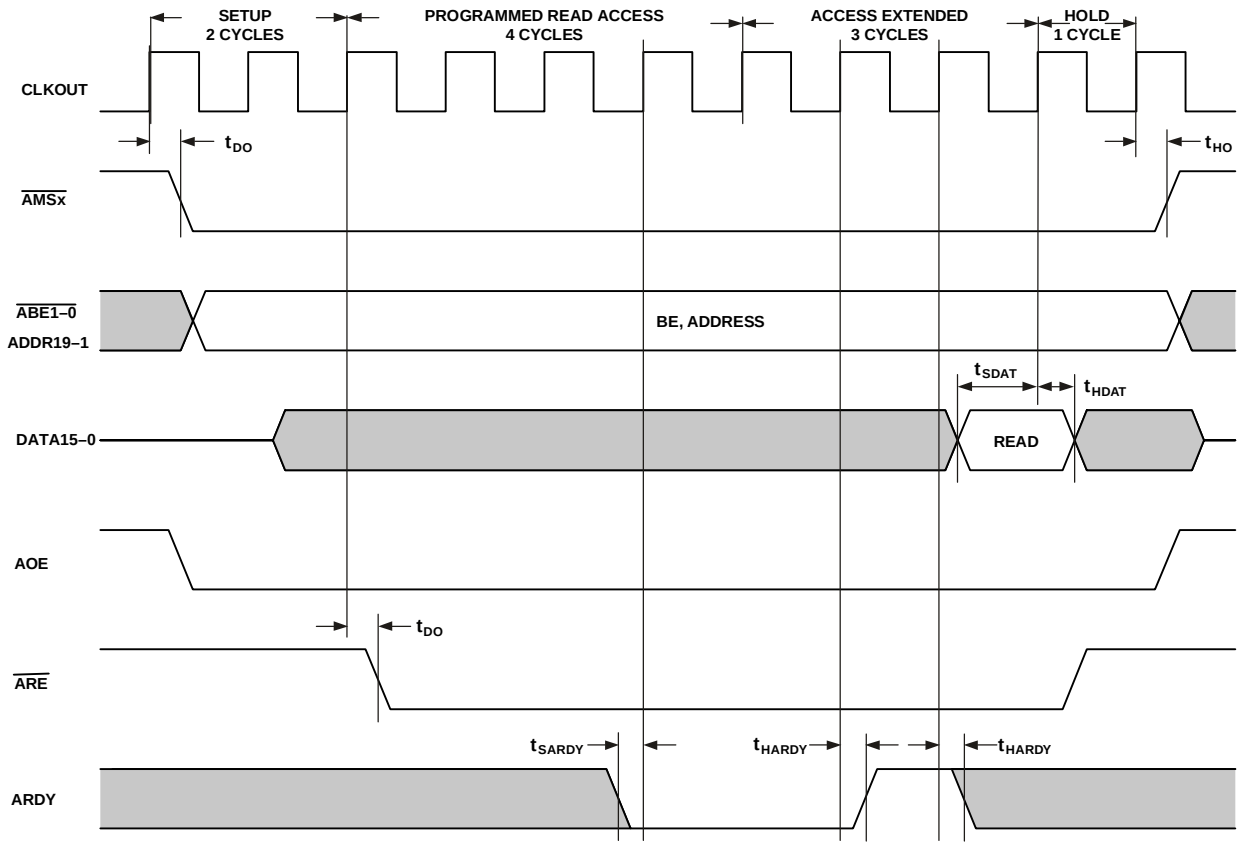


Figure 16. Asynchronous Memory Read Cycle Timing

SDRAM Interface Timing

Table 15. SDRAM Interface Timing

Parameter	Description	Min	Max	Unit
Timing Requirement				
t_{SSDAT}	DATA Setup Before CLKOUT	2.1		ns
t_{HSDAT}	DATA Hold After CLKOUT	0.8		ns
Switching Characteristic				
t_{SCLK}	CLKOUT Period	7.5		ns
t_{SCLKH}	CLKOUT Width High	TBD		ns
t_{SCLKL}	CLKOUT Width Low	TBD		ns
t_{DCAD}	Command, ADDR, Data Delay After CLKOUT[1]		6.0	ns
t_{HCAD}	Command, ADDR, Data Hold After CLKOUT[1]	0.8		ns
t_{DSDAT}	Data Disable After CLKOUT		6.0	ns
t_{ENSDAT}	Data Enable After CLKOUT	1.0		ns

[1] Command pins include: $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, SDQM, $\overline{SMS}$, SA10, SCKE.

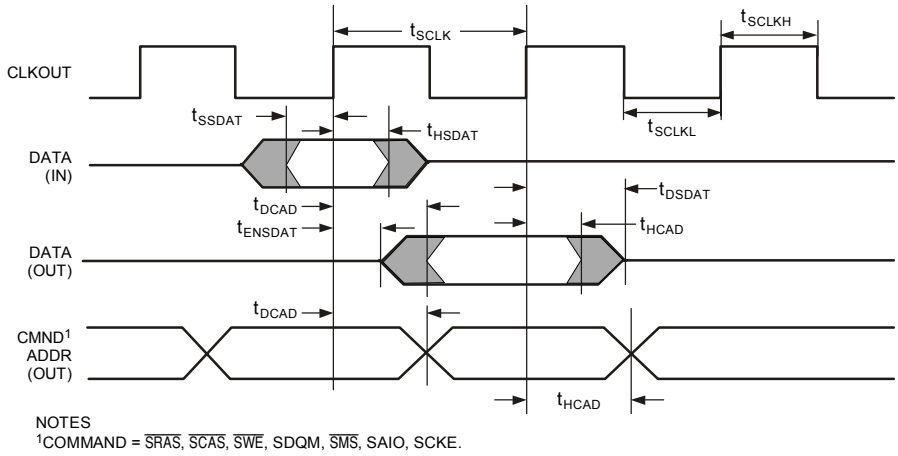


Figure 17. SDRAM Interface Timing

Serial Ports

Table 16. Serial Ports—External Clock

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SFSE}	TFS/RFS Setup Before TCLK/RCLK[1]	3.0		ns
t_{HFSE}	TFS/RFS Hold After TCLK/RCLK[1]	3.0		ns
t_{SDRE}	Receive Data Setup Before RCLK[1]	3.0		ns
t_{HDRE}	Receive Data Hold Before RCLK[1]	3.0		ns
t_{SCLKW}	TCLK/RCLK Width	4.5		ns
t_{SCLK}	TCLK/RCLK Period	15.0		ns

[1]Referenced to sample edge.

Table 17. Serial Ports—Internal Clock

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SFSI}	TFS/RFS Setup Before TCLK/RCLK[1]	7.0		ns
t_{HFSI}	TFS/RFS Hold After TCLK/RCLK[1]	2.0		ns
t_{SDRI}	Receive Data Setup Before RCLK[1]	7.0		ns
t_{HDRI}	Receive Data Hold Before RCLK[1]	4.0		ns

[1]Referenced to sample edge.

Table 18. Serial Ports—External or Internal Clock

Parameter	Description	Min	Max	Unit
Switching Characteristics				
t_{DFSE}	RFS Delay After RCLK (Internally Generated RFS)[1]		10.0	ns
t_{HOFSE}	RFS Hold After RCLK (Internally Generated RFS)[1]	6.0		ns

[1]Referenced to drive edge.

Table 19. Serial Ports—External Clock

Parameter	Description	Min	Max	Unit
Switching Characteristics				
t_{DFSE}	TFS Delay After TCLK (Internally Generated TFS)[1]		10.0	ns
t_{HOFSE}	TFS Hold After TCLK (Internally Generated TFS)[1]	6.0		ns
t_{DDTE}	Transmit Data Delay After TCLK[1]		10.0	ns
t_{HDTE}	Transmit Data Hold After TCLK[1]	6.0		ns

[1]Referenced to drive edge.

Table 20. Serial Ports—Internal Clock

Parameter	Description	Min	Max	Unit
Switching Characteristics				
t_{DFSI}	TFS Delay After TCLK (Internally Generated TFS)[1]		4.0	ns
t_{HOFSI}	TFS Hold After TCLK (Internally Generated TFS)[1]	0.0		ns
t_{DDTI}	Transmit Data Delay After TCLK[1]		4.0	ns
t_{HDTI}	Transmit Data Hold After TCLK[1]	0.0		ns
t_{SCLKIW}	TCLK/RCLK Width	4.5		ns

[1]Referenced to drive edge.

Table 21. Serial Ports—Enable and Three-State

Parameter	Description	Min	Max	Unit
Switching Characteristics				
t_{DTENE}	Data Enable Delay from External TCLK[1]	5.0		ns
t_{DDTTE}	Data Disable Delay from External TCLK[1]		12.0	ns
t_{DTENI}	Data Enable Delay from Internal TCLK	2.0		ns
t_{DDTTI}	Data Disable Delay from Internal TCLK[1]		5.0	ns

[1] Referenced to drive edge.

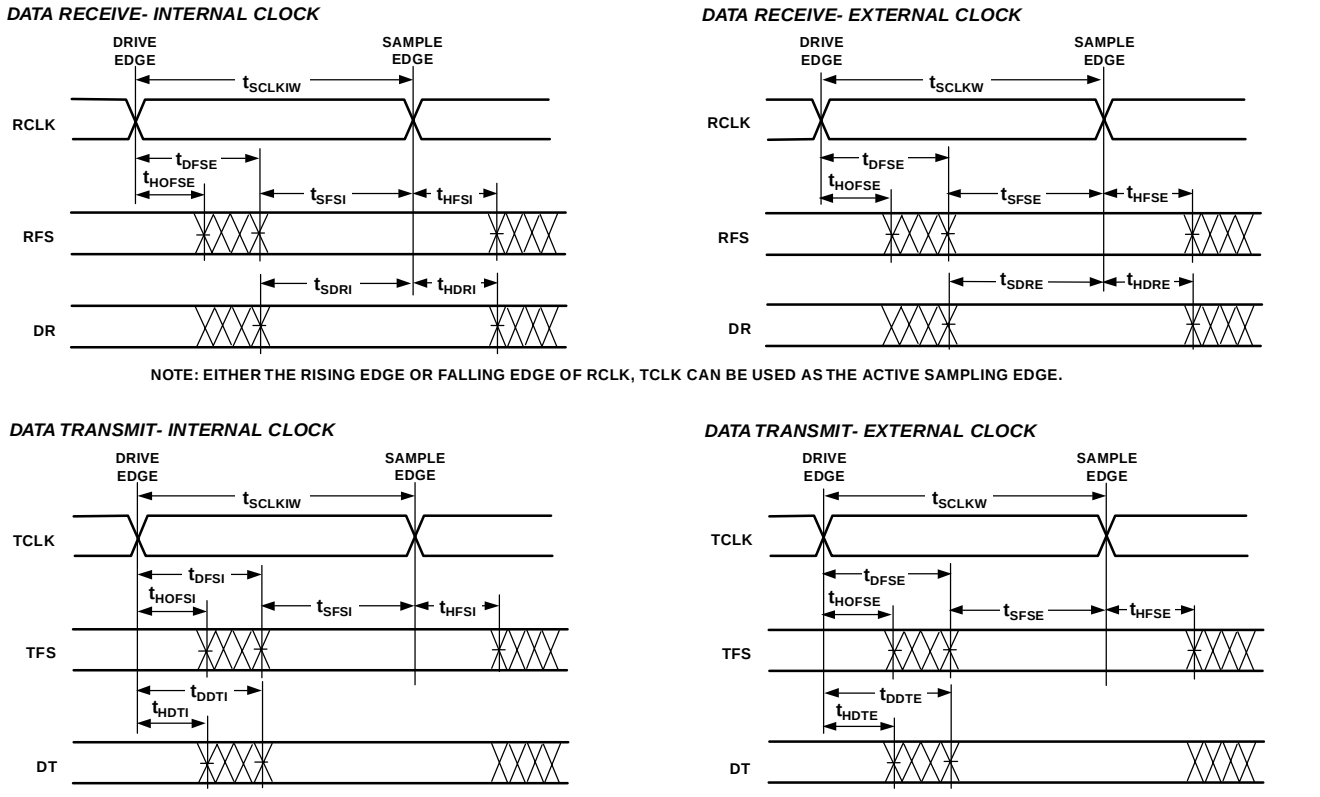


Figure 18. Serial Ports

Serial Peripheral Interface (SPI) Port—Master Timing

Table 22 and Figure 19 describe SPI port master operations.

Table 22. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SSPID}	Data input valid to SCLK edge (data input setup)	1.6		ns
t_{HSPID}	SCLK sampling edge to data input invalid	1.6		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIxSEL}}$ low to first SCLK edge (x=0 or 1)	$2t_{SCLK}$		ns
t_{SPICHM}	Serial clock high period	$2t_{SCLK}$		ns
t_{SPICLM}	Serial clock low period	$2t_{SCLK}$		ns
t_{SPICLK}	Serial clock period	$4t_{SCLK}$		ns
t_{HDSM}	Last SCLK edge to $\overline{\text{SPIxSEL}}$ high (x=0 or 1)	$2t_{SCLK}$		ns
t_{SPITDM}	Sequential transfer delay	$2t_{SCLK}$		ns
t_{DDSPID}	SCLK edge to data out valid (data out delay)	0	6	ns
t_{HDSPID}	SCLK edge to data out invalid (data out hold)	0	5	ns

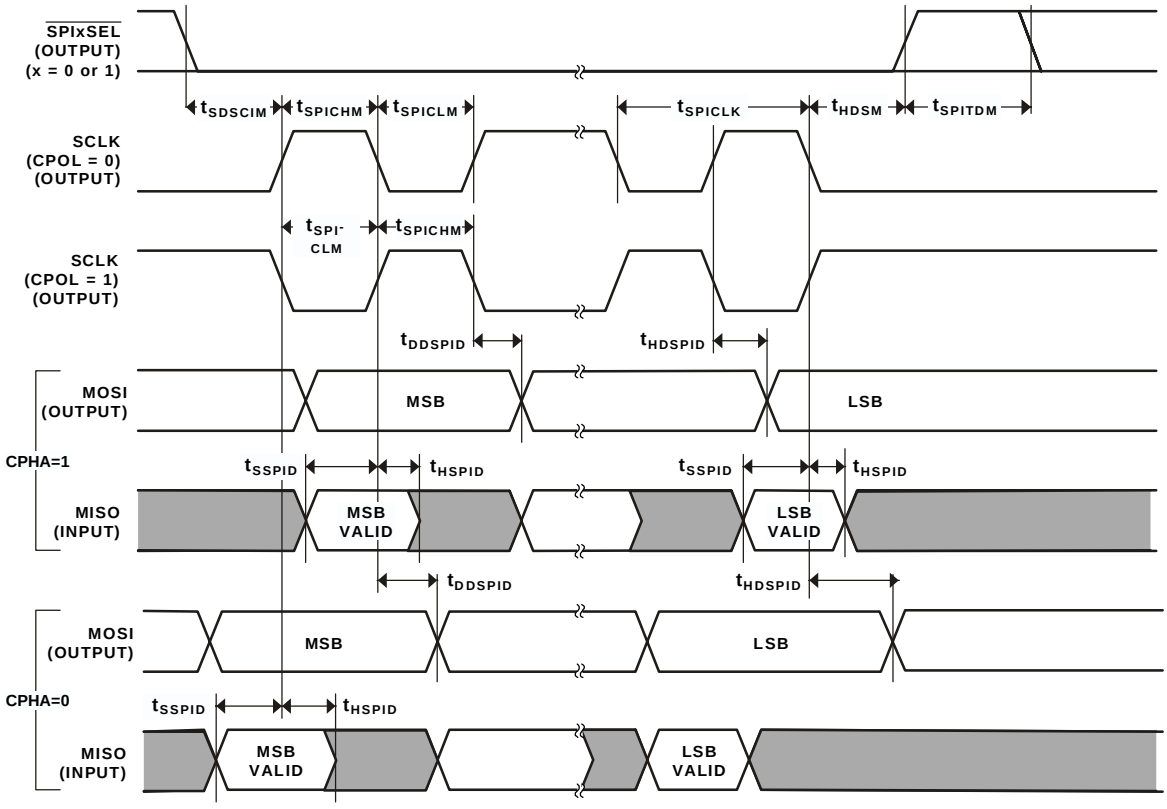


Figure 19. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 23 and Figure 20 describe SPI port slave operations.

Table 23. Serial Peripheral Interface (SPI) Port—Slave Timing

Parameter	Description	Min	Max	Unit
Timing Requirements				
t_{SPICHS}	Serial clock high period	$2t_{SCLK}$		ns
t_{SPICLS}	Serial clock low period	$2t_{SCLK}$		ns
t_{SPICLK}	Serial clock period	$4t_{SCLK}$		ns
t_{HDS}	Last SPICLK edge to $\overline{SPISS}$ not asserted	$2t_{SCLK}$		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{SCLK}$		ns
t_{SDSCI}	$\overline{SPISS}$ assertion to first SCLK edge	$2t_{SCLK}$		ns
t_{SSPID}	Data input valid to SCLK edge (data input setup)	1.6		ns
t_{HSPID}	SCLK sampling edge to data input invalid	1.6		ns
Switching Characteristics				
t_{DSOE}	$\overline{SPISS}$ assertion to data out active	0	6	ns
t_{DSDHI}	$\overline{SPISS}$ deassertion to data high impedance	0	6	ns
t_{DDSPID}	SCLK edge to data out valid (data out delay)	0	5	ns
t_{HDSPID}	SCLK edge to data out invalid (data out hold)	0	5	ns

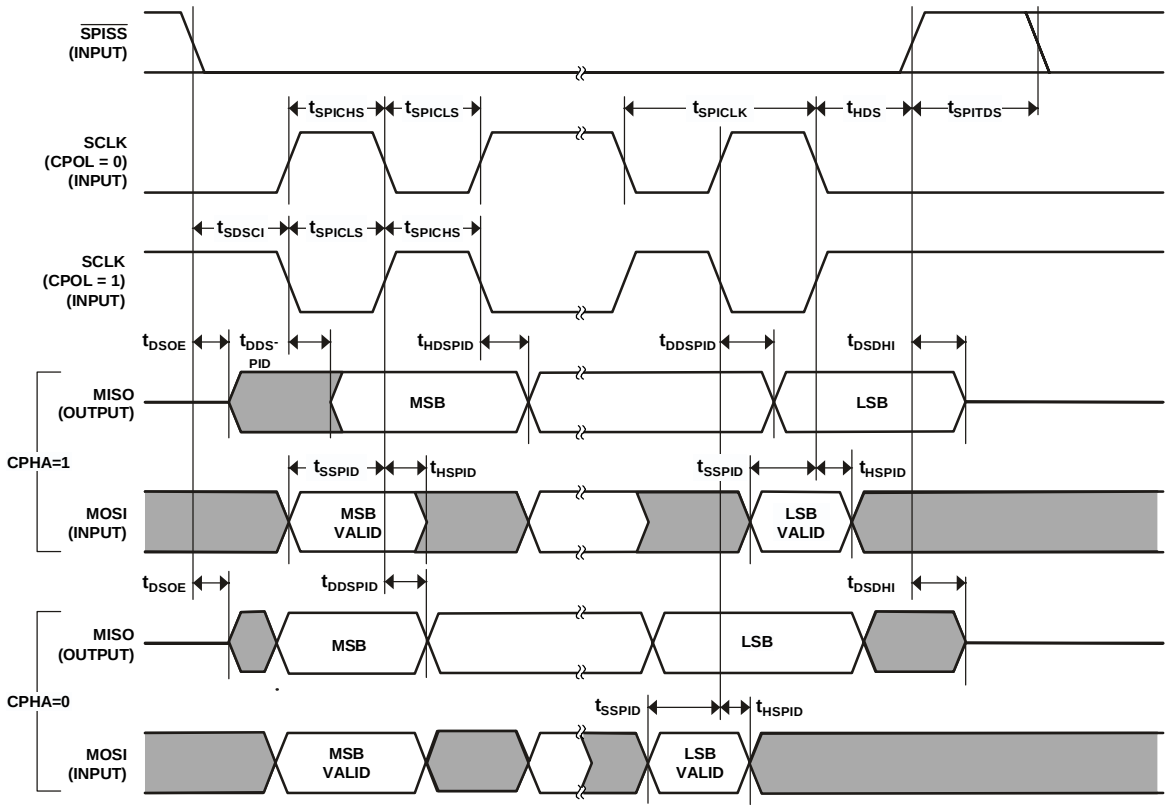


Figure 20. Serial Peripheral Interface (SPI) Port—Slave Timing

Universal Asynchronous Receiver-Transmitter (UART) Port—Receive and Transmit Timing

Figure 21 describes UART port receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 21 there is some latency between the generation internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

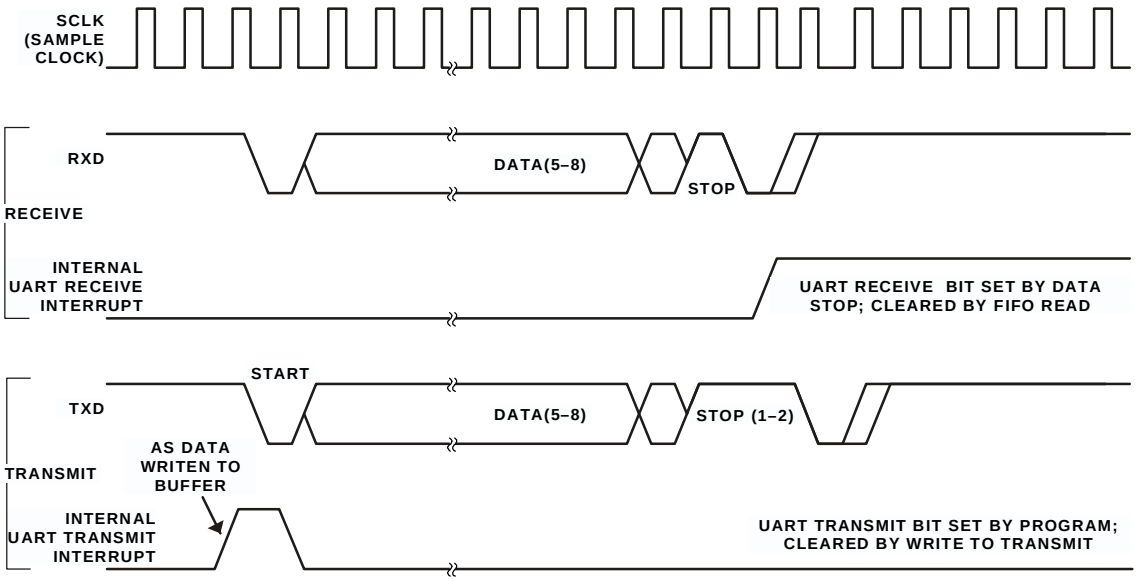


Figure 21. UART Port—Receive and Transmit Timing

JTAG Test And Emulation Port Timing

Table 24 and Figure 22 describe JTAG port operations.

Table 24. JTAG Port Timing

Parameter	Description	Min	Max	Unit
Timing Parameters				
t_{TCK}	TCK Period	20		ns
t_{STAP}	TDI, TMS Setup Before TCK High		4	ns
t_{HTAP}	TDI, TMS Hold After TCK High		4	ns
t_{SSYS}	System Inputs Setup Before TCK Low[1]		4	ns
t_{HSYS}	System Inputs Hold After TCK Low[1]		5	ns
t_{TRSTW}	$\overline{\text{TRST}}$ Pulsewidth[2]	4		ns
Switching Characteristics				
t_{DTDO}	TDO Delay from TCK Low		4	ns
t_{DSYS}	System Outputs Delay After TCK Low[3]	0	5	ns

[1] System Inputs=DATA31-0, ADDR25-2, ARDY, TMR2-0, PF15-0, RSCLK0, RFS0, DR0, TSCLK0, TFS0, RSCLK1, RFS1, DR1, TSCLK1, TFS1, MOSI0, MISO0, SCK0, MOSI1, MISO1, SCK1, RX0, RX1, TSB_CLK, XVER_DATA, DPLS, DMNS, NMI, $\overline{\text{RESET}}$, BYPASS, BMODE2-0, PCI_AD31-0, $\overline{\text{PCI\_CBE3-0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, PCI_DEVSEL, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, PCI_PAR, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_GNT}}$, PCI_IDSEL, $\overline{\text{PCI\_LOCK}}$, PCI_CLK, $\overline{\text{PCI\_INTA}}$, $\overline{\text{PCI\_INTB}}$, $\overline{\text{PCI\_INTC}}$, $\overline{\text{PCI\_INTD}}$.

[2] 50 MHz max.

[3] System Outputs=DATA31-0, ADDR25-2, $\overline{\text{ABE}}$/SDQM3-0, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, CLKOUT/SCLK1, SCLK0, SCKE, SA10, $\overline{\text{SRAS}}$, $\overline{\text{SCAS}}$, $\overline{\text{SWE}}$, $\overline{\text{SMS3-0}}$, TMR2-0, PF15-0, RSCLK0, RFS0, TSCLK0, TFS0, DT0, RSCLK1, RFS1, TSCLK1, TFS1, DT1, MOSI0, MISO0, SCK0, MOSI1, MISO1, SCK1, TX0, TX1, TXDPLS, TXDMNS, $\overline{\text{TXEN}}$, SUSPEND, DEEPSLEEP, PCI_AD31-0, $\overline{\text{PCI\_CBE3-0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, PCI_DEVSEL, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, PCI_PAR, $\overline{\text{PCI\_REQ}}$, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_INTA}}$, $\overline{\text{EMU}}$.

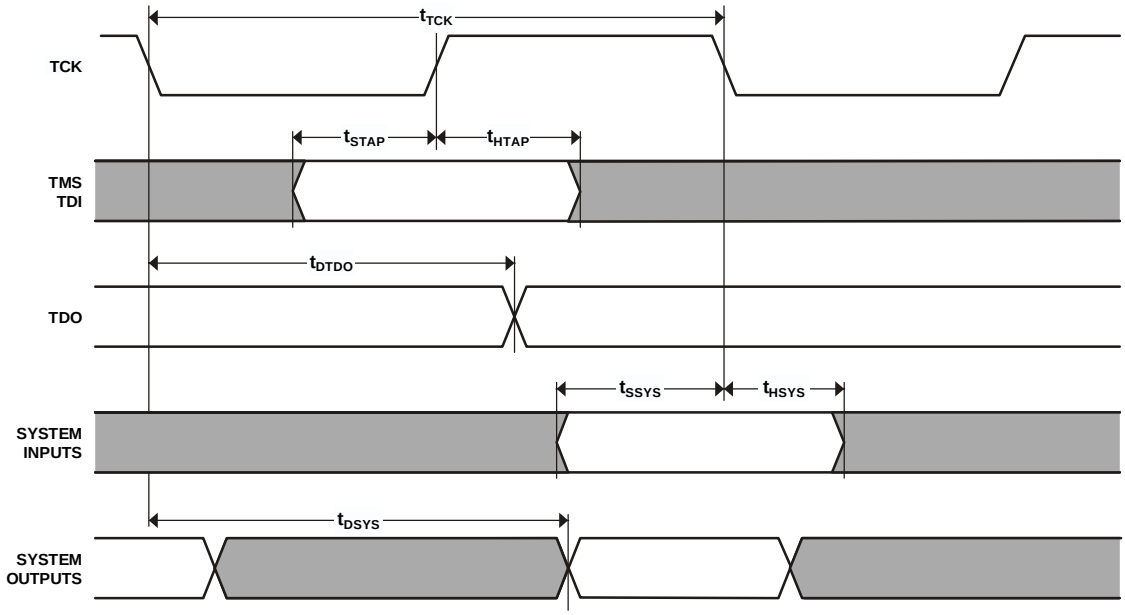


Figure 22. JTAG Port Timing

Power Dissipation

Total power dissipation has two components, one due to internal circuitry (P_{INT}) and one due to the switching of external output drivers (P_{EXT}). Table 25 shows the power dissipation for internal circuitry. Internal power dissipation is dependent on the instruction execution sequence and the data operands involved. Table 26 lists the conditions under which the values in Table 25 are obtained.

Table 25. Internal Power Dissipation

Parameter	Test Conditions	Typical (V_{DDINT}=1.5 V)[1]	Typical (V_{DDINT}=1.0 V)[1]	Units
I_{DDHIGH}	$t_{CCLKMIN}$, 25°C	TBD	TBD	mA
I_{DDTYP}	$t_{CCLKMIN}$, 25°C	TBD	TBD	mA
I_{DDLOW}	$t_{CCLKMIN}$, 25°C	TBD	TBD	mA
I_{DDSYS}	$t_{CCLKMIN}$, 25°C	TBD	TBD	mA
I_{DDEFR}	$t_{CCLKMIN}$, 25°C	TBD	TBD	mA
$I_{DDACTIVE}$	25°C	TBD	TBD	mA
$I_{DDSLEEP}$	25°C	TBD	TBD	mA
$I_{DDDEEPSLEEP}$	25°C	TBD	TBD	mA

[1]Typical IDD data is specified for nominal V_{DDINT} and typical process parameters.Maximum I_{DD} is within TBD% of typical values.

Table 26. Internal Power Dissipation Conditions

Parameter	Mode	PLL	CCLK	SCLK	Activity
I_{DDHIGH}[1]	Full-On	Enabled	Enabled	Enabled	TBD
I_{DDTYP}[1]	Full-On	Enabled	Enabled	Enabled	TBD
I_{DDLOW}[1]	Full-On	Enabled	Enabled	Enabled	TBD
I_{DDSYS}[2]	Full-On	Enabled	Enabled	Enabled	TBD
I_{DDEFR}[3]	Full-On	Enabled	Enabled	Enabled	Algorithm-dependent
$I_{DDACTIVE}$	Active	Enabled/Bypassed	Enabled	Enabled	TBD
$I_{DDSLEEP}$	Sleep	Enabled	Disabled	Enabled	TBD
$I_{DDDEEPSLEEP}$	Deep-Sleep	Disabled	Disabled	Disabled	TBD

[1]TBD instruction mix.
[2]TBD instruction mix and system DMA every cycle.
[3]Implementation of Enhanced Full Rate (EFR) GSM algorithm, instruction and data fetch from L1/L2 memories and cache.

The external component of total power dissipation is caused by the switching of output pins. Its magnitude depends on

- The number of output pins that switch during each cycle (O)
- The maximum frequency at which they can switch (f)
- Their load capacitance (C)
- Their voltage swing (V_{DDEXT})

The external component is calculated using

$$P_{EXT} = O \times C \times V_{DD} \times f$$

The frequency f includes driving the load high and then back low. For example: DATA31–0 pins can drive high and low at a maximum rate of $1/(2 \times t_{SCLK})$ while in SDRAM burst mode.

A typical power consumption can now be calculated for these conditions by adding a typical internal power dissipation.

$$P_{TOTAL} = P_{EXT} + (I_{DD} \times V_{DDINT})$$

Note that the conditions causing a worst-case P_{EXT} differ from those causing a worst-case P_{INT}. Maximum P_{INT} cannot occur while 100% of the output pins are switching from all ones (1s) to all zeros (0s). Note also that it is not common for an application to have 100%,or even 50%, of the outputs switching simultaneously.

Environmental Conditions

The ADSP-21535 is offered in a 260-lead PBGA package.

The ADSP-21535 is specified for a case temperature (T_{CASE}). To ensure that T_{CASE} is not exceeded, an airflow source may be used. T_{CASE} is calculated using

$$T_{CASE} = T_{AMB} + (PD \times \theta_{CA})$$

T_{CASE} = Case temperature (measured on top surface of package
PD = Power dissipation in W (this value depends upon the specific application)

See Table 27 for three instances of linear feet per minute of airflow.

Table 27. Airflow

Linear Ft/Min	0	200	400
θ_{CA} (°C/W)	TBD	TBD	TBD

ADSP-21535 260-Lead PBGA Pinout

Table 28 lists the PBGA pinout by signal name. Table 29 on page 43 lists the pinout by pin number.

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal)

SIGNAL	PIN #
$\overline{\text{ABE0}}$/SDQM0	E02
$\overline{\text{ABE1}}$/SDQM1	B01
$\overline{\text{ABE2}}$/SDQM2	G03
$\overline{\text{ABE3}}$/SDQM3	H07
ADDR2	A06
ADDR3	B06
ADDR4	D06
ADDR5	C06
ADDR6	A05
ADDR7	B05
ADDR8	A04
ADDR9	C05
ADDR10	D05
ADDR11	B04
ADDR12	A01
ADDR13	C04
ADDR14	D04
ADDR15	A03
ADDR16	B03
ADDR17	A02
ADDR18	C03
ADDR19	D03
ADDR20	B02
ADDR21	C02
ADDR22	E03
ADDR23	C01
ADDR24	F03
ADDR25	D02
$\overline{\text{AMS0}}$	F02
$\overline{\text{AMS1}}$	D01
$\overline{\text{AMS2}}$	H03
$\overline{\text{AMS3}}$	G02
$\overline{\text{AOE}}$	E01
ARDY	R01
$\overline{\text{ARE}}$	F01
$\overline{\text{AWE}}$	G01
BMODE0	B14
BMODE1	A14

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal) (Continued)

SIGNAL	PIN #
BMODE2	B13
BYPASS	C12
CLKIN1	D09
CLKOUT/SCLK1	H01
DATA0	N02
DATA1	M03
DATA2	T01
DATA3	P02
DATA4	N03
DATA5	R02
DATA6	P03
DATA7	U01
DATA8	U02
DATA9	T02
DATA10	V02
DATA11	V03
DATA12	R04
DATA13	U03
DATA14	T03
DATA15	T04
DATA16	U04
DATA17	V04
DATA18	V05
DATA19	R05
DATA20	T05
DATA21	U05
DATA22	V06
DATA23	R06
DATA24	U06
DATA25	T06
DATA26	V07
DATA27	V08
DATA28	U07
DATA29	R07
DATA30	T07
DATA31	V09
DMNS	D08
DPLS	C09
DR0	V14
DR1	U15
DT0	R14
DT1	V17
$\overline{\text{EMU}}$	A13
GND	C13
GND	H02
GND	H08
GND	H10
GND	H11
GND	J07
GND	J08
GND	J09

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal) (Continued)

SIGNAL	PIN #
GND	J10
GND	J11
GND	J12
GND	K02
GND	K07
GND	K08
GND	K09
GND	K10
GND	K11
GND	K12
GND	L07
GND	L08
GND	L09
GND	L10
GND	L11
GND	M07
GND	M09
GND	M10
MISO0	T16
MISO1	U18
MOSI0	U16
MOSI1	T17
N/C	A18
N/C	R03
N/C	V01
N/C	V18
NMI	B11
PCI_AD0	E17
PCI_AD1	E18
PCI_AD2	G16
PCI_AD3	F17
PCI_AD4	F18
PCI_AD5	G18
PCI_AD6	G17
PCI_AD7	H18
PCI_AD8	J18
PCI_AD9	H17
PCI_AD10	K18
PCI_AD11	H16
PCI_AD12	L18
PCI_AD13	J17
PCI_AD14	M18
PCI_AD15	K17
PCI_AD16	J16
PCI_AD17	K16
PCI_AD18	N18
PCI_AD19	P18
PCI_AD20	L17
PCI_AD21	L16
PCI_AD22	R18
PCI_AD23	T18

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal) (Continued)

SIGNAL	PIN #
PCI_AD24	M17
PCI_AD25	M16
PCI_AD26	N17
PCI_AD27	P17
PCI_AD28	P15
PCI_AD29	N16
PCI_AD30	R17
PCI_AD31	P16
PCI_CBE0	F16
PCI_CBE1	F15
PCI_CBE2	E16
PCI_CBE3	D17
PCI_CLK	D14
PCI_DEVSEL	C16
PCI_FRAME	C17
PCI_GNT	C18
PCI_IDSEL	B18
PCI_INTA	C14
PCI_INTB	B15
PCI_INTC	A15
PCI_INTD	D13
PCI_IRDY	E15
PCI_LOCK	A16
PCI_PAR	C15
PCI_PERR	D15
PCI_REQ	D16
PCI_RST	D18
PCI_SERR	B16
PCI_STOP	A17
PCI_TRDY	B17
PF0 /SPISS0 /MSEL0	U08
PF1 /SPISS1 /MSEL1	R08
PF2 /SPI0SEL1 /MSEL2	T08
PF3 /SPI1SEL1 /MSEL3	V10
PF4 /SPI0SEL2 /MSEL4	U09
PF5 /SPI1SEL2 /MSEL5	R09
PF6 /SPI0SEL3 /MSEL6	T09

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal) (Continued)

SIGNAL	PIN #
PF7	R11
/SPI1SEL3	
/DF	
PF8	T11
/SPI0SEL4	
/SSEL0	
PF9	U11
/SPI1SEL4	
/SSEL1	
PF10	V12
/SPI0SEL5	
PF11	T12
/SPI1SEL5	
PF12	R12
/SPI0SEL6	
PF13	U12
/SPI1SEL6	
PF14	V13
/SPI0SEL7	
PF15	T13
/SPI1SEL7	
RESET	B09
RFS0	U13
RFS1	V16
RSCLK0	R13
RSCLK1	U14
RX0	A07
RX1	B08
SA10	M01
SCAS	L03
SCK0	U17
SCK1	R16
SCKE	L01
SCLK0	K01
SLEEP	D12
SMS0	M02
SMS1	P01
SMS2	N01
SMS3	K03
SRAS	L02
SUSPEND	A11
SWE	J03
TCK	D10
TDI	C11
TDO	D11
TFS0	T14
TFS1	R15
TMR0	B07
TMR1	C07
TMR2	D07
TMS	A12

Table 28. 260-Lead PBGA Pin Assignment (Alphabetically by Signal) (Continued)

SIGNAL	PIN #
TRST	B12
TSCLK0	V15
TSCLK1	T15
TX0	A08
TX1	C08
TXDMNS	G10
TXDPLS	B10
TXEN	C10
USB_CLK	G07
V_{DDEXT}	E04
V_{DDEXT}	G04
V_{DDEXT}	G08
V_{DDEXT}	J01
V_{DDEXT}	J02
V_{DDEXT}	J04
V_{DDEXT}	K04
V_{DDEXT}	L04
V_{DDEXT}	M04
V_{DDEXT}	P04
V_{DDINT}	F04
V_{DDINT}	G11
V_{DDINT}	G12
V_{DDINT}	G15
V_{DDINT}	H04
V_{DDINT}	H09
V_{DDINT}	H12
V_{DDINT}	L12
V_{DDINT}	M08
V_{DDINT}	M11
V_{DDINT}	M12
V_{DDINT}	N04
V_{DDINT}	N15
$V_{DDPCIEXT}$	H15
$V_{DDPCIEXT}$	J15
$V_{DDPCIEXT}$	K15
$V_{DDPCIEXT}$	L15
$V_{DDPCIEXT}$	M15
V_{DDPLL}	G09
V_{DDRTC}	U10
V_{SSPLL}	A10
V_{SSRTC}	V11
XTALI	R10
XTALO	T10
XVER_DATA	A09

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number)

PIN #	SIGNAL
A01	ADDR12
A02	ADDR17
A03	ADDR15
A04	ADDR8
A05	ADDR6
A06	ADDR2
A07	RX0
A08	TX0
A09	XVER_DATA
A10	V_{SSPLL}
A11	SUSPEND
A12	TMS
A13	$\overline{EMU}$
A14	BMODE1
A15	$\overline{PCI\_INTC}$
A16	$\overline{PCI\_LOCK}$
A17	$\overline{PCI\_STOP}$
A18	N/C
B01	$\overline{ABE1}$/SDQM1
B02	ADDR20
B03	ADDR16
B04	ADDR11
B05	ADDR7
B06	ADDR3
B07	TMR0
B08	RX1
B09	$\overline{RESET}$
B10	TXDPLS
B11	NMI
B12	$\overline{TRST}$
B13	BMODE2
B14	BMODE0
B15	$\overline{PCI\_INTB}$
B16	$\overline{PCI\_SERR}$
B17	$\overline{PCI\_TRDY}$
B18	$\overline{PCI\_IDSEL}$
C01	ADDR23
C02	ADDR21
C03	ADDR18
C04	ADDR13
C05	ADDR9
C06	ADDR5
C07	TMR1
C08	TX1
C09	DPLS
C10	$\overline{TXEN}$
C11	TDI
C12	BYPASS
C13	GND
C14	$\overline{PCI\_INTA}$
C15	PCI_PAR

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number) (Continued)

PIN #	SIGNAL
C16	PCI_DEVSEL
C17	$\overline{PCI\_FRAME}$
C18	$\overline{PCI\_GNT}$
D01	$\overline{AMS1}$
D02	ADDR25
D03	ADDR19
D04	ADDR14
D05	ADDR10
D06	ADDR4
D07	TMR2
D08	DMNS
D09	CLKIN1
D10	TCK
D11	TDO
D12	SLEEP
D13	$\overline{PCI\_INTD}$
D14	PCI_CLK
D15	$\overline{PCI\_PERR}$
D16	$\overline{PCI\_REQ}$
D17	$\overline{PCI\_CBE3}$
D18	$\overline{PCI\_RST}$
E01	$\overline{AOE}$
E02	$\overline{ABE0}$/SDQM0
E03	ADDR22
E04	V_{DDEXT}
E15	$\overline{PCI\_IRDY}$
E16	$\overline{PCI\_CBE2}$
E17	PCI_AD0
E18	PCI_AD1
F01	$\overline{ARE}$
F02	$\overline{AMS0}$
F03	ADDR24
F04	V_{DDINT}
F15	$\overline{PCI\_CBE1}$
F16	$\overline{PCI\_CBE0}$
F17	PCI_AD3
F18	PCI_AD4
G01	$\overline{AWE}$
G02	$\overline{AMS3}$
G03	$\overline{ABE2}$/SDQM2
G04	V_{DDEXT}
G07	USB_CLK
G08	V_{DDEXT}
G09	V_{DDPLL}
G10	TXDMNS
G11	V_{DDINT}
G12	V_{DDINT}
G15	V_{DDINT}
G16	PCI_AD2
G17	PCI_AD6
G18	PCI_AD5

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number) (Continued)

PIN #	SIGNAL
H01	CLKOUT/SCLK1
H02	GND
H03	AMS2
H04	V_{DDINT}
H07	ABE3/SDQM3
H08	GND
H09	V_{DDINT}
H10	GND
H11	GND
H12	V_{DDINT}
H15	$V_{DDPCIEXT}$
H16	PCI_AD11
H17	PCI_AD9
H18	PCI_AD7
J01	V_{DDEXT}
J02	V_{DDEXT}
J03	SWE
J04	V_{DDEXT}
J07	GND
J08	GND
J09	GND
J10	GND
J11	GND
J12	GND
J15	$V_{DDPCIEXT}$
J16	PCI_AD16
J17	PCI_AD13
J18	PCI_AD8
K01	SCLK0
K02	GND
K03	SMS3
K04	V_{DDEXT}
K07	GND
K08	GND
K09	GND
K10	GND
K11	GND
K12	GND
K15	$V_{DDPCIEXT}$
K16	PCI_AD17
K17	PCI_AD15
K18	PCI_AD10
L01	SCKE
L02	SRAS
L03	SCAS
L04	V_{DDEXT}
L07	GND
L08	GND
L09	GND
L10	GND
L11	GND

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number) (Continued)

PIN #	SIGNAL
L12	V_{DDINT}
L15	$V_{DDPCIEXT}$
L16	PCI_AD21
L17	PCI_AD20
L18	PCI_AD12
M01	SA10
M02	SMS0
M03	DATA1
M04	V_{DDEXT}
M07	GND
M08	V_{DDINT}
M09	GND
M10	GND
M11	V_{DDINT}
M12	V_{DDINT}
M15	$V_{DDPCIEXT}$
M16	PCI_AD25
M17	PCI_AD24
M18	PCI_AD14
N01	SMS2
N02	DATA0
N03	DATA4
N04	V_{DDINT}
N15	V_{DDINT}
N16	PCI_AD29
N17	PCI_AD26
N18	PCI_AD18
P01	SMS1
P02	DATA3
P03	DATA6
P04	V_{DDEXT}
P15	PCI_AD28
P16	PCI_AD31
P17	PCI_AD27
P18	PCI_AD19
R01	ARDY
R02	DATA5
R03	N/C
R04	DATA12
R05	DATA19
R06	DATA23
R07	DATA29
R08	PF1 /SPISS1 /MSEL1
R09	PF5 /SPI1SEL2 /MSEL5
R10	XTALI

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number) (Continued)

PIN #	SIGNAL
R11	PF7
	/SPI1SEL3
	/DF
R12	PF12
	/SPI0SEL6
R13	RSCLK0
R14	DT0
R15	TFS1
R16	SCK1
R17	PCI_AD30
R18	PCI_AD22
T01	DATA2
T02	DATA9
T03	DATA14
T04	DATA15
T05	DATA20
T06	DATA25
T07	DATA30
T08	PF2
	/SPI0SEL1
	/MSEL2
T09	PF6
	/SPI0SEL3
	/MSEL6
T10	XTALO
T11	PF8
	/SPI0SEL4
	/SSEL0
T12	PF11
	/SPI1SEL5
T13	PF15
	/SPI1SEL7
T14	TFS0
T15	TSCLK1
T16	MISO0
T17	MOSI1
T18	PCI_AD23
U01	DATA7
U02	DATA8
U03	DATA13
U04	DATA16
U05	DATA21
U06	DATA24
U07	DATA28
U08	PF0
	/SPISS0
	/MSEL0
U09	PF4
	/SPI0SEL2
	/MSEL4
U10	V_{DDRTC}

Table 29. 260-Lead PBGA Pin Assignment (Numerically by Pin Number) (Continued)

PIN #	SIGNAL
U11	PF9
	/SPI1SEL4
	/SSEL1
U12	PF13
	/SPI1SEL6
U13	RFS0
U14	RSCLK1
U15	DR1
U16	MOSI0
U17	SCK0
U18	MISO1
V01	N/C
V02	DATA10
V03	DATA11
V04	DATA17
V05	DATA18
V06	DATA22
V07	DATA26
V08	DATA27
V09	DATA31
V10	PF3
	/SPI1SEL1
	/MSEL3
V11	V_{SSRTC}
V12	PF10
	/SPI0SEL5
V13	PF14
	/SPI0SEL7
V14	DR0
V15	TSCLK0
V16	RFS1
V17	DT1
V18	N/C

OUTLINE DIMENSIONS

Dimensions in Figure 23 are shown in millimeters.

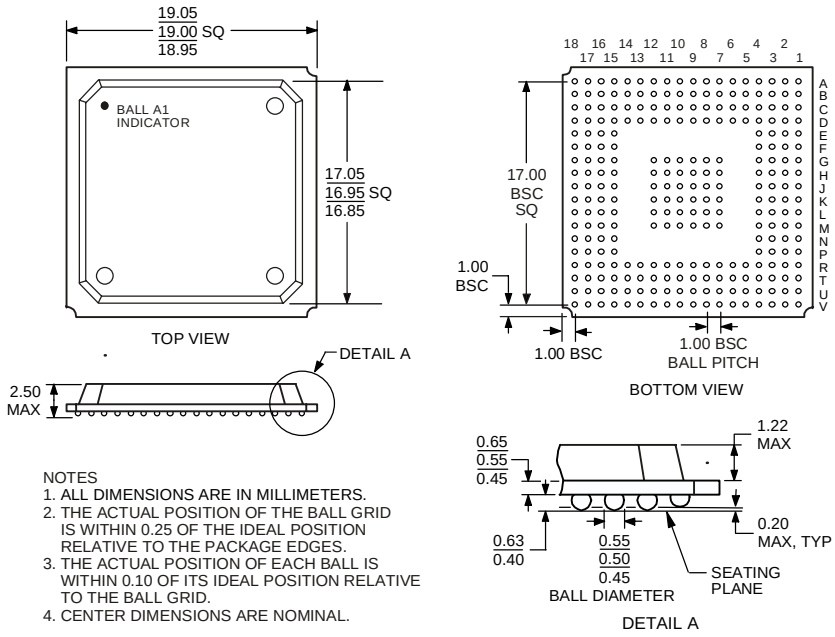


Figure 23. 260-Lead Metric Plastic Ball Grid Array (PBGA) (B-260)

ORDERING GUIDE

Part Number	Case Temperature Range	Instruction Rate	Operating Voltage
ADSP-21535PKB-300	0°C to 85°C	300 MHz	0.9 V to 1.5 V internal, 3.3 V I/O
ADSP-21535PBB-200	-40°C to +105°C	200 MHz	0.9 V to 1.5 V internal, 3.3 V I/O