Preliminary Technical Data ADSP-21535

SUMMARY

300 MHz High-Performance Blackfin DSP Core
- **Two 16-Bit MACs**
- **Two 32-Bit ALUs or Four 8-Bit Video ALUs**
- **Dual 40-Bit Accumulators**
- **40-Bit Shifter**
- **RISC-Like Register and Instruction Model for Ease of Programming and Compiler-Friendly Support**
- **Advanced Debug, Trace, and Performance-Monitoring Support**

Memory
- **4G-Byte Unified Address Range**
- **Two 32-Bit DAGs for General Addressing and Circular Buffer Support**

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 Management Unit Providing Memory Protection

Synchronous External Memory Controller with Glueless SDRAM Support

Asynchronous External Memory Controller with Glueless Support for SRAM, FLASH, ROM

Flexible Memory Booting Options From SPI and External Memory

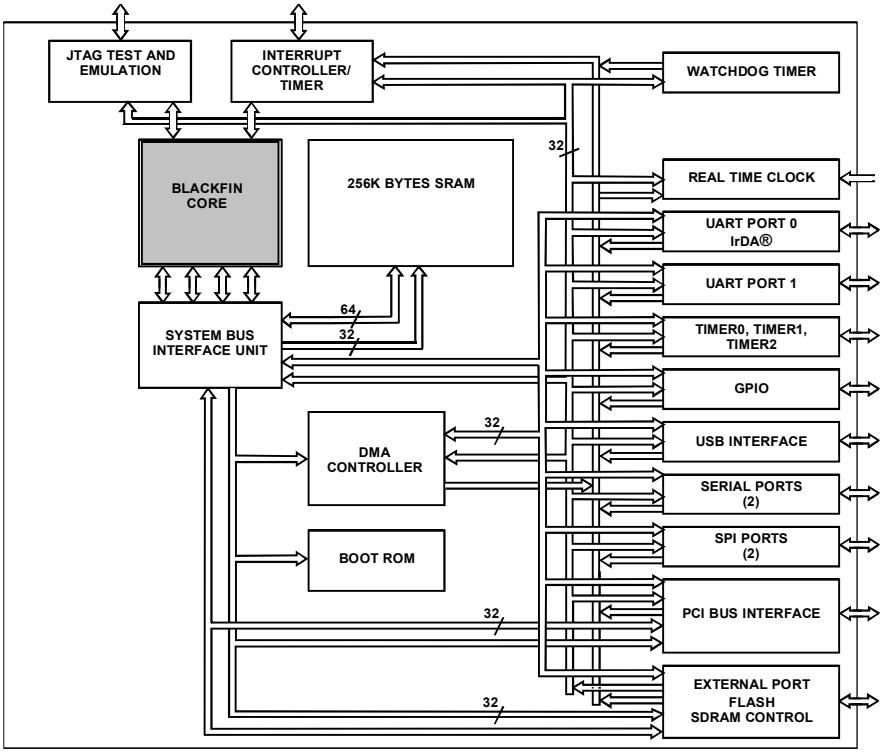

Figure 1. ADSP-21535 Block Diagram

REV. PrA

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

PERIPHERALS

- **32-Bit, 33-MHz, PCI 2.2-compliant Bus Interface with Master and Slave Support**
- **Integrated USB 1.1-compliant Device Interface**
- **Event Controller**
- **Two UARTs, One with Support for IrDA®**
- **Three Timer/counters with PWM Support**
- **Two SPI-compatible Ports**
- **Two 6-pin Full-Duplex Synchronous Serial Ports**
- **Real-Time Clock**
- **Watchdog Timer**
- **16 General Purpose I/O Pins**
- **Debug/JTAG Interface**
- **On Chip PLL Capable of 1x To 31x Frequency Multiplication**

0.9 V to 1.5 V Core V_{DD} with Dynamic Power Management
3.3 V-Tolerant I/O
0 °C To 85 °C Case Temperature Range
256-Lead PBGA Package

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 ADI's Blackfin DSP core architecture. The Blackfin DSP 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 consumption 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, 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 Figure 1 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, an interrupt controller for flexible management of interrupts from the on-chip peripherals as well as 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 are supported by a flexible DMA structure with individual DMA channels integrated into the peripherals as appropriate to their needs. 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 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 Architecture

As shown in Figure 2 on page 3, the Blackfin DSP core contains two multiplier/accumulators (MACs), two 32-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.

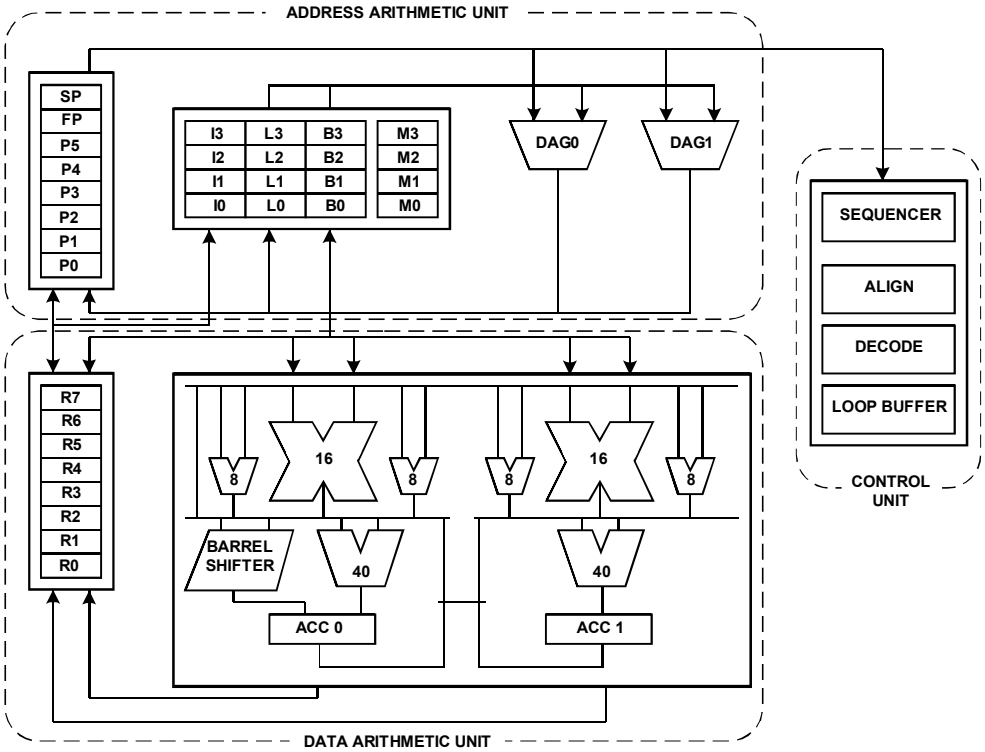

Figure 2. Blackfin DSP Core Architecture

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

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 two modes of operation, user mode and supervisor 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 3.

The L1 memory system is the primary highest-performance memory available to the Blackfin DSP core. The L2 memory provides much more capacity; however, read latency is higher. Lastly, the off-chip memory system, accessed through the External Memory Controller (EMC), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing more than 768M bytes of physical memory.

The memory DMA controller provides high-bandwidth, multi-channel, 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.

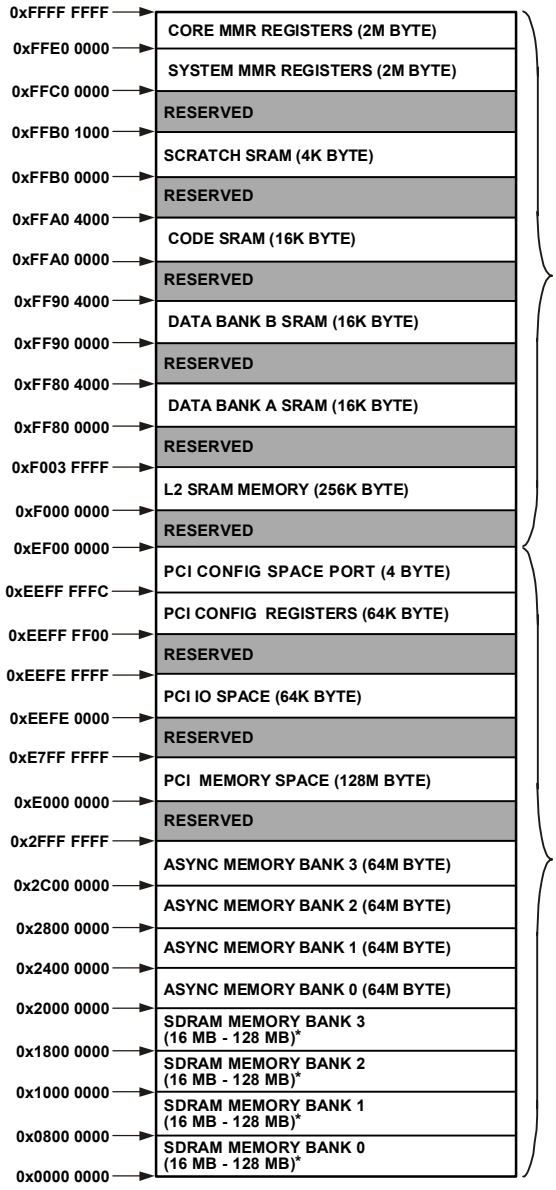

Figure 3. ADSP-21535 Internal/External Memory Map

The second on-chip memory block is the L1 data memory, consisting of two blocks of 16K bytes each. Each L1 data memory may be configured as a 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 and cannot be configured as cache memory.

The fourth on-chip memory system is the L2 SRAM memory array which provides 256 KBytes of high speed SRAM at the full bandwidth of the core, and slightly longer latency than the L1 memory blocks. 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 in excess of 2.4G bytes per second.

External (Off-Chip) Memory

External memory is accessed via the External Memory Controller. 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 RAM 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 4 gigabyte 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 CPU core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the CPU 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.

Boot Memory Space

The internal boot ROM 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 15.

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 allow 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 kernel stack.

The ADSP-21535 event controller consists of two stages, the Core Interrupt Controller (CIC) and the System Interrupt Controller (SIC). The Core Interrupt 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 CIC.

Core Interrupt Controller (CIC)

The CIC supports nine general-purpose interrupts (IVG7:15), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest-priority interrupts (IVG14:15) 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 CIC, identifies their names in the Event Vector Table (EVT), and lists their priorities.

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 CIC. 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 ECIC and the default mappings into the

Table 1. Core Interrupt Controller (CIC)

Priority (0 is Highest)	Event Class	EVT Entry
0	Emulation/Test Control	EMUN
1	Reset	IRST
2	Non-Maskable Interrupt	NMI
3	Exceptions	EVSW
4	Global Interrupt 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

Table 2. System Interrupt Controller (SIC)

Peripheral Interrupt Event	Peripheral Interrupt ID	Default Mapping
Real-Time Clock Interrupt	0	IVG7
Reserved	1	-
USB Interrupt	2	IVG7
PCI Interrupt	3	IVG7
SPORT 0 RCV DMA Interrupt	4	IVG8
SPORT 0 XMT DMA Interrupt	5	IVG8
SPORT 1 RCV DMA Interrupt	6	IVG8

Table 2. System Interrupt Controller (SIC) (Continued)

Peripheral Interrupt Event	Peripheral Interrupt ID	Default Mapping
SPORT 1 XMT DMA Interrupt	7	IVG8
SPI 0 DMA Interrupt	8	IVG9
SPI 1 DMA Interrupt	9	IVG9
UART 0 RCV Interrupt	10	IVG10
UART 0 XMT Interrupt	11	IVG10
UART 1 RCV Interrupt	12	IVG10
UART 1 XMT Interrupt	13	IVG10
Timer A Interrupt	14	IVG11
Timer B Interrupt	15	IVG11
Timer C Interrupt	16	IVG11
GPIO Interrupt A	17	IVG12
GPIO Interrupt B	18	IVG12
Memory DMA Interrupt	19	IVG13
Software Watchdog Timer Interrupt	20	IVG13
Reserved	21:26	-
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 CIC, three registers are used to coordinate and control events. Each register is 16-bits wide, while each bit represents a particular event class:

- CIC 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.
- CIC 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.)
- CIC 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 on page 6.

- SIC Interrupt Mask Register – 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 – 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 – 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 “Low-Power Operation” on page 12.)

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 CIC 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 Controller

The ADSP-21535 has a DMA controller that supports automated data transfers with minimal overhead for the DSP core. Cycle stealing 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 a 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 transfer control block (TCB). When successive DMA sequences are needed, these TCBs can be linked or chained together, so the completion of one DMA sequence auto-initiates and starts the next sequence. The TCBs 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 Controller

The external memory controller 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, 32-bit PCI (Peripheral Component Interconnect), Revision 2.2-compliant interface. 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:

- 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 memory spaces of PCI space (memory, IO, and configuration space) are mapped into the flat 32-bit memory space of the ADSP-21535. Since the PCI memory space is as large as the ADSP-21535 memory address space, a segmented, or 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 Port

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, stop watch, 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 1Hz 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.

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.

If configured to generate a hardware reset, the timer can be programmed to reset only the ADSP-21535 CPU, or both the CPU and the ADSP-21535 peripherals. 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 three general-purpose programmable timer units in the ADSP-21535. Each timer has one 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 timer units can be independently programmed as a PWM, internally or externally clocked timer, or pulse width counter.

The 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 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 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 (f_{SCLK}) 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 Big Endian (MSB) or Little Endian (LSB) format.
- 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, one data word each DSP cycle. Either the DSP's core or a host processor can link or chain sequences of DMA transfers between a SPORT and memory. The chained DMA can be dynamically allocated and updated through the Transfer Control Blocks (TCBs, or DMA parameters) 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 24 or 32 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 both transmit and receive data streams. The SPI's DMA controller can only service unidirectional accesses at any given time.

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

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).
   a. 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 the 16450 standard. 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:

- 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 to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

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

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 (PF0:15) 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 14. Each programmable flag can be individually controlled 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.

Low-Power Operation

The ADSP-21535 has four low-power operating modes described below that significantly reduce power consumption when the processor operates in reduced performance conditions. In addition, the extended core power management controller provides the control functions, with the appropriate external power regulation capability, to dynamically alter the processor core supply voltage, further reducing power consumption. Control of clocking to each of the ADSP-21535 peripherals also reduces power consumption. See Table 3 for a summary of the power settings for each mode.

Full On Operating Mode – No power savings

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 – Low power savings

In the Active mode, the PLL is enabled, but bypassed. The input clock (CLKIN) is used to directly generate the clocks for the processor core (CCLK) and peripherals (SCLK). 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 (PLLCTL). The PLL lock counter (PLL LOCKCNT) determines when the new multiplier ratio takes effect.

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

Table 3. Power Settings

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

Relaxed Operating Mode – Medium power savings

The Relaxed mode reduces power consumption by not only bypassing the PLL but also disabling it. The input clock (CLKIN) is directly used to generate the clocks for the processor core (CCLK) and peripherals (SCLK). As in the Active mode, significant dynamic power savings can be achieved with the processor running at one-half the CLKIN frequency and, unlike the Active mode, further power savings are accomplished with the PLL being disabled. In this mode the PLL multiplication ratio can be changed by setting the appropriate values in the SSEL fields of the PLL control register (PLLCTL). The PLL lock counter (PLL LOCKCNT) determines when the new multiplier ratio takes effect.

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

Sleep Operating Mode – High power savings

The Sleep mode reduces power consumption by disabling the clock to the processor core (CCLK). The system clock (SCLK) however, continues 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 (PLLCTL). 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 (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 SLEEP output is asserted in this mode, thereby enabling an external power regulator to determine when it is safe to vary the processor core's V_{DD}.

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 following instruction sequence:

```
CLI;      // disable interrupts
IDLE;     // source NOPs into the pipeline and
          assert IDLE output on SSYNC
SSYNC;    // drain the pipeline, IDLE asserts after
          system acknowledge
STI;      // re-enable interrupts after wakeup
```

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.

Dynamic Power Management

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. ADSP-21535 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, including CLKIN input buffer	V_{DDEXT}

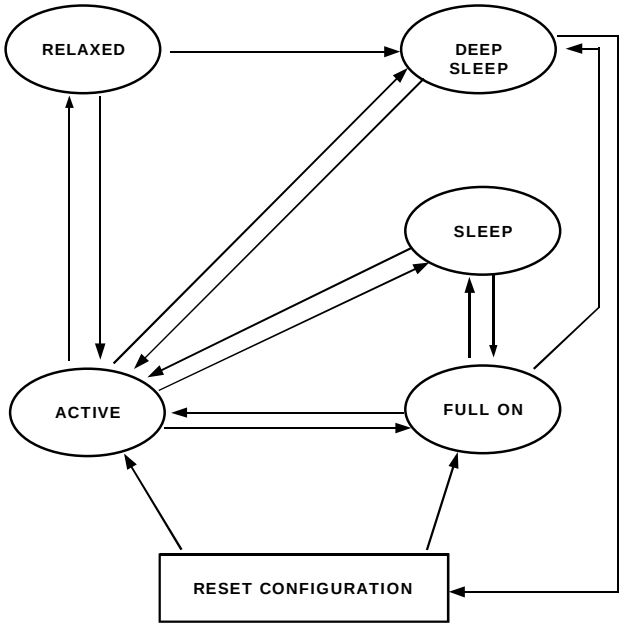

Figure 6. Mode Transitions

The power consumed 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 consumption, while reducing the voltage by 25% reduces power consumption 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.

The Dynamic Power Management feature of the ADSP-21535 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 consumption can be modeled by the following equation:

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

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 consumes about 10% of the power consumed at nominal frequency and voltage.

Peripheral Power Control

The ADSP-21535 provides additional power control capability by allowing dynamic scheduling of clock inputs to each of the peripherals. This allows finer control of power by enabling or disabling clocking 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 (IOCKREG). The Peripheral Clock Enable Register allows individual control for each of the following peripherals:

- PCI
- EBIU controller
- GPIO
- 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 1 x to 31 x 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 *Hardware Reference*.

The peripheral clock is supplied to the CLKOUT_SCLK0 pin.

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 f ields in the PLL control register (PLLCTL). The

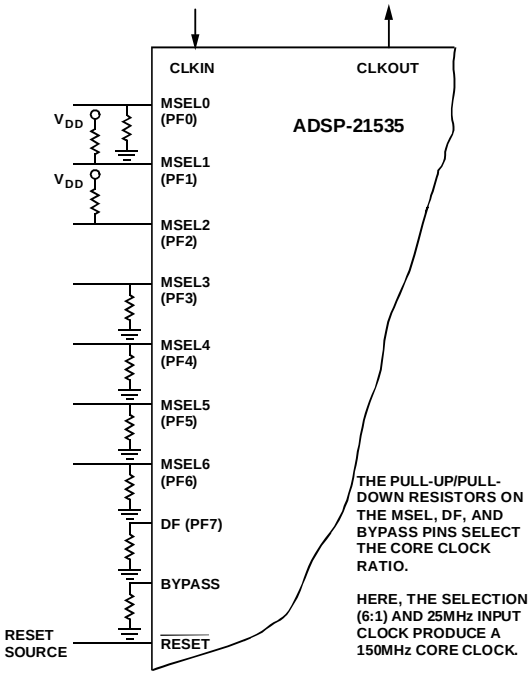

Figure 7. Clock Ratio Example

values programmed into the SSEL fields define a divide ratio between the core clock (CLKIN) and the system clock. Table 5 illustrates the system clock ratios:

Table 5. System Clock Ratios

Signal Name SSEL [1:0]	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 SSEL [1:0] is determined by sampling the Programmable Flag input pins (PF[9:8]) during reset. The SSEL value can be changed dynamically by writing the appropriate values to the PLL control register (PLLCTL), 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

BMODE[2:0]	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 4. 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, bit 4 of 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 4-Gbyte 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 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.

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 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 on page 17. The customer must supply this header on the target board in order to communicate with the emulator. The interface consists of a standard dual row 0.025" square post header, set on 0.1" × 0.1" spacing, with a minimum post length of 0.235". Pin 3 is the key position used to prevent the pod from being inserted backwards. This pin must be clipped on the target board.

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

As can be seen in Figure 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

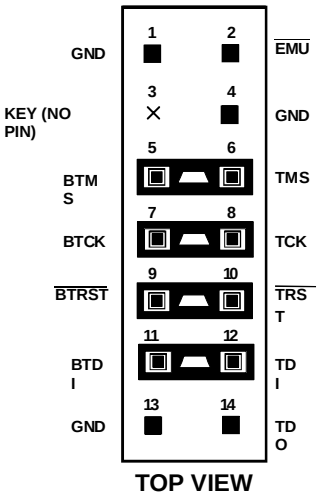

Figure 8. JTAG Target Board Connector for JTAG Equipped Analog Devices DSP (Jumpers in Place)

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.

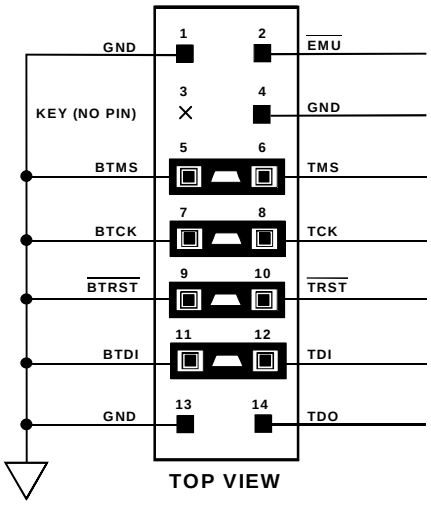

Figure 9. JTAG Target Board Connector with No Local Boundary Scan

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. 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.

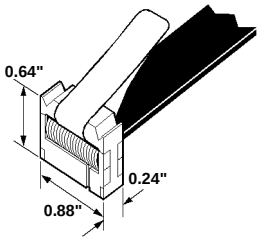

Figure 10. JTAG Pod Connector Dimensions

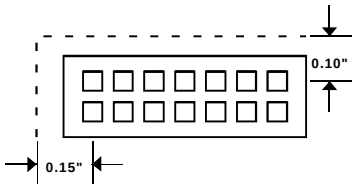

Figure 11. JTAG Pod Connector Keep-Out Area

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, PF[15:0], USB_CLK, NMI, $\overline{\text{TRST}}$, $\overline{\text{RESET}}$, $\overline{\text{PCI\_CLK}}$, XTALI, XTALO.

Unused inputs should be tied or pulled to V_{DDEXT} or GND.

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
ADDR	O/T	External address bus.
DATA	I/O/T	External data bus.
$\overline{\text{ABE}}$ /SDQM	O/T	Asynchronous memory byte enables, SDRAM data masks.
$\overline{\text{AMS}}$	O/T	Chip selects for asynchronous memories.
ARDY	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 CLK 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.
$\overline{\text{SMS}}$	O/T	Memory select pin of external memory bank configured for SDRAM. Connect to SDRAM's chip select pin.
TMR0	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	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	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.
PF[15] /SPI1SEL[7]	I/O/T	General purpose I/O pins. SPI output select pin.
PF[14] /SPI0SEL[7]	I/O/T	General purpose I/O pins. SPI output select pin.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
PF[13] /SPI1SEL[6]	I/O/T	General purpose I/O pins. SPI output select pin.
PF[12] /SPI0SEL[6]	I/O/T	General purpose I/O pins. SPI output select pin.
PF[11] /SPI1SEL[5]	I/O/T	General purpose I/O pins. SPI output select pin.
PF[10] /SPI0SEL[5]	I/O/T	General purpose I/O pins. SPI output select pin.
PF[9] /SPI1SEL[4] /SSEL[1]	I/O	General purpose I/O pins. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF[8] /SPI0SEL[4] /SSEL[0]	I/O	General purpose I/O pins. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF[7] /SPI1SEL[3] /DF	I/O	General purpose I/O pins. 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.
PF[6] /SPI0SEL[3] /MSEL[6]	I/O	General purpose I/O pins. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[5] /SPI1SEL[2] /MSEL[5]	I/O	General purpose I/O pins. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[4] /SPI0SEL[2] /MSEL[4]	I/O	General purpose I/O pins. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[3] /SPI1SEL[1] /MSEL[3]	I/O	General purpose I/O pins. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[2] /SPI0SEL[1] /MSEL[2]	I/O	General purpose I/O pins. SPI output select pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[1] /SPISS1 /MSEL[1]	I/O	General purpose I/O pins. SPI slave select input pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
PF[0] /SPISS0 /MSEL[0]	I/O	General purpose I/O pins. SPI slave select input pin. Sensed for configuration state during hardware reset, used to configure the PLL. Selects CK to CLKIN ratio.
RSCLK0	I/O/T	Receive serial clock for SPORT0.
RFS0	I/O/T	Receive frame synchronization for SPORT0.
DR0	I	Serial data receive for SPORT0.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
TSCLK0	I/O/T	Transmit serial clock for SPORT0.
TFS0	I/O/T	Transmit frame synchronization for SPORT0.
DT0	O	Serial data transmit for SPORT0.
RSCLK1	I/O/T	Receive serial clock for SPORT1.
RFS1	I/O/T	Receive frame synchronization for SPORT1.
DR1	I	Serial data receive for SPORT1.
TSCLK1	I/O/T	Transmit serial clock for SPORT1.
TFS1	I/O/T	Transmit frame synchronization for SPORT1.
DT1	O	Serial data transmit for SPORT1.
MOSI0	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	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	I/O	Clock line for SPI0. Master device output clock signal. Slave device input clock signal.
MOSI1	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	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	I/O	Clock line for SPI1. Master device output clock signal. Slave device input clock signal.
RX0	I	UART0 receive pin.
TX0	O	UART0 transmit pin.
RX1	I	UART1 receive pin.
TX1	O	UART1 transmit pin.
USB_CLK	I	USB clock.
XVER_DATA	I	Single ended receive data output from USB transceiver to the USBD module.
DPLS	I	Differential D+ receive data output from the USB transceiver to the UBD module.
DMNS	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	I	Non-maskable interrupt.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
TCK	I	JTAG clock.
TDO	O	JTAG serial data out.
TDI	I	JTAG serial data in.
TMS	I	JTAG master slave.
$\overline{\text{TRST}}$	I	JTAG reset. Tie to ground if not used.
$\overline{\text{RESET}}$	I	When this pin is asserted to logic zero level for at least 10 CLKIN cycles, a hardware reset is initiated. TShe 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.
SLEEP	O	Denotes that the Blackfin DSP Core is in Deep Sleep mode.
BMODE[2:0]	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_AD	I/O/T	PCI address and data bus.
$\overline{\text{PCI\_CBE}}$	I/O/T	PCI byte enables.
$\overline{\text{PCI\_FRAME}}$	I/O/T	PCI frame signal. Used by PCI initiators for signalling the beginning and end of a PCI transaction.
$\overline{\text{PCI\_IRDY}}$	I/O/T	PCI initiator ready signal.
$\overline{\text{PCI\_TRDY}}$	I/O/T	PCI target ready signal.
PCI_DEVSEL	I/O/T	PCI device select signal. Asserted by targets of PCI transactions to claim the transaction.
$\overline{\text{PCI\_STOP}}$	I/O/T	PCI stop signal.
$\overline{\text{PCI\_PERR}}$	I/O/T	PCI parity error signal.
PCI_PAR	I/O/T	PCI parity signal.
$\overline{\text{PCI\_REQ}}$	O	PCI request signal. Used for requesting the use of the PCI bus.
$\overline{\text{PCI\_SERR}}$	I/O/T	PCI system error signal. Requires a pullup on the system board.
$\overline{\text{PCI\_RST}}$	I/O/T	PCI reset signal.
$\overline{\text{PCI\_GNT}}$	I	PCI grant signal. Used for granting access to the PCI bus.
$\overline{\text{PCI\_IDSEL}}$	I	PCI initialization device select signal. Individual device selects for targets of PCI configuration transactions.
$\overline{\text{PCI\_LOCK}}$	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	I	PCI clock.

Table 7. Pin Descriptions (Continued)

Pin	Type	Function
$\overline{\text{PCI\_INTA}}$	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.
$\overline{\text{PCI\_INTB}}$	I	PCI interrupt B line. Monitored by ADSP-21535 when acting as the system processor.
$\overline{\text{PCI\_INTC}}$	I	PCI interrupt C line. Monitored by the ADSP-21535 when acting as the system processor.
$\overline{\text{PCI\_INTD}}$	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.
$\overline{\text{EMU}}$	I	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.

ADSP-21535—SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter	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	3.15	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[1], @ V_{DDINT} = max	2.0		V_{DDEXT} + 0.5	V
V_{IL}	Low Level Input Voltage[1], @ V_{DDINT} = min	–0.3		0.6	V
T_{CASE}	Case Operating Temperature	0		85	°C

[1] Applies to input and bidirectional pins.

Specifications subject to change without notice

ELECTRICAL CHARACTERISTICS

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

[1] Applies to output and bidirectional pins.

[2] Applies to input pins.

[3] Applies to three-statable pins.

[4] Applies to all signal pins.

[5] Guaranteed, but not tested.

ABSOLUTE MAXIMUM RATINGS

Parameter[1]	Absolute Maximum Rating
Internal (Core) Supply Voltage (V_{DDINT})	–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, and 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.

Specifications subject to change without notice

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 on page 25 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 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

Table 8. Core and System Clock Requirements (Continued)

Parameter	Description	Min	Max	Unit
$t_{CCLK0.9}$	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. Clock In Timing Requirements

Parameter	Description	Min	Max	Unit
t_{CKIN}	Clock In Period	30	100	ns

Power Dissipation

Total power dissipation has two components, one due to internal circuitry and one due to the switching of external output drivers. Table 10 shows the power dissipation for internal circuitry. Internal power dissipation is dependent on the instruction execution sequence and the data operands involved. Table 11 lists the conditions under which the values in Table 10 are obtained.

Table 10. 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
$I_{DDRELAXED}$	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 11. 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	
$I_{DDRELAXED}$	Relaxed	Disabled /Bypassed	Enabled	Enabled	
$I_{DDSLEEP}$	Sleep	Disabled	Disabled	Enabled	
$I_{DDDEEPSLEEP}$	Deep Sleep	Disabled	Disabled	Disabled	

[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.

ADSP-21535 256-Lead PBGA Pinout

Table 12 lists the PBGA pinout by signal name.

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal)

SIGNAL	PIN #
$\overline{\text{ABE0}}$	TBD
$\overline{\text{ABE1}}$	TBD
$\overline{\text{ABE2}}$	TBD
$\overline{\text{ABE3}}$	TBD
ADDR2	TBD
ADDR3	TBD
ADDR4	TBD
ADDR5	TBD
ADDR6	TBD
ADDR7	TBD
ADDR8	TBD
ADDR9	TBD
ADDR10	TBD
ADDR11	TBD
ADDR12	TBD
ADDR13	TBD
ADDR14	TBD
ADDR15	TBD
ADDR16	TBD
ADDR17	TBD
ADDR18	TBD
ADDR19	TBD
ADDR20	TBD
ADDR21	TBD
ADDR22	TBD
ADDR23	TBD
ADDR24	TBD
ADDR25	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
$\overline{\text{AMS0}}$	TBD
$\overline{\text{AMS1}}$	TBD
$\overline{\text{AMS2}}$	TBD
$\overline{\text{AMS3}}$	TBD
$\overline{\text{AOE}}$	TBD
ARDY	TBD
$\overline{\text{ARE}}$	TBD
$\overline{\text{AWE}}$	TBD
BMODE0	TBD
BMODE1	TBD
BMODE2	TBD
BYPASS	TBD
CLKIN1	TBD
CLKOUT	TBD
DATA0	TBD
DATA1	TBD
DATA2	TBD
DATA3	TBD
DATA4	TBD
DATA5	TBD
DATA6	TBD
DATA7	TBD
DATA8	TBD
DATA9	TBD
DATA10	TBD
DATA11	TBD
DATA12	TBD
DATA13	TBD
DATA14	TBD
DATA15	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
DATA16	TBD
DATA17	TBD
DATA18	TBD
DATA19	TBD
DATA20	TBD
DATA21	TBD
DATA22	TBD
DATA23	TBD
DATA24	TBD
DATA25	TBD
DATA26	TBD
DATA27	TBD
DATA28	TBD
DATA29	TBD
DATA30	TBD
DATA31	TBD
DMNS	TBD
DPLS	TBD
DR0	TBD
DR1	TBD
DT0	TBD
DT1	TBD
$\overline{\text{EMU}}$	TBD
MISO0	TBD
MISO1	TBD
MOSI0	TBD
MOSI1	TBD
NMI	TBD
PCI_AD0	TBD
PCI_AD1	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
PCI_AD2	TBD
PCI_AD3	TBD
PCI_AD4	TBD
PCI_AD5	TBD
PCI_AD6	TBD
PCI_AD7	TBD
PCI_AD8	TBD
PCI_AD9	TBD
PCI_AD10	TBD
PCI_AD11	TBD
PCI_AD12	TBD
PCI_AD13	TBD
PCI_AD14	TBD
PCI_AD15	TBD
PCI_AD16	TBD
PCI_AD17	TBD
PCI_AD18	TBD
PCI_AD19	TBD
PCI_AD20	TBD
PCI_AD21	TBD
PCI_AD22	TBD
PCI_AD23	TBD
PCI_AD24	TBD
PCI_AD25	TBD
PCI_AD26	TBD
PCI_AD27	TBD
PCI_AD28	TBD
PCI_AD29	TBD
PCI_AD30	TBD
PCI_AD31	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
PCI_CBE0	TBD
PCI_CBE1	TBD
PCI_CBE2	TBD
PCI_CBE3	TBD
PCI_CLK	TBD
PCI_DEVSEL	TBD
PCI_FRAME	TBD
PCI_GNT	TBD
PCI_IDSEL	TBD
PCI_INTA	TBD
PCI_INTB	TBD
PCI_INTC	TBD
PCI_INTD	TBD
PCI_IRDY	TBD
PCI_LOCK	TBD
PCI_PAR	TBD
PCI_PERR	TBD
PCI_REQ	TBD
PCI_RST	TBD
PCI_SERR	TBD
PCI_STOP	TBD
PCI_TRDY	TBD
PF0 /SPISS0 /MSEL0	TBD
PF1 /SPISS1 /MSEL1	TBD
PF2 /SPI0SEL1 /MSEL2	TBD
PF3 /SPI1SEL1 /MSEL3	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
PF4 /SPI0SEL2 /MSEL4	TBD
PF5 /SPI1SEL2 /MSEL5	TBD
PF6 /SPI0SEL3 /MSEL6	TBD
PF7 /SPI1SEL3 /DF	TBD
PF8 /SPI0SEL4 /SSEL0	TBD
PF9 /SPI1SEL4 /SSEL1	TBD
PF10 /SPI0SEL5	TBD
PF11 /SPI1SEL5	TBD
PF12 /SPI0SEL6	TBD
PF13 /SPI1SEL6	TBD
PF14 /SPI0SEL7	TBD
PF15 /SPI1SEL7	TBD
RESET	TBD
RFS0	TBD
RFS1	TBD
RSCLK0	TBD
RSCLK1	TBD
RX0	TBD
RX1	TBD
SA10	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
$\overline{\text{SCAS}}$	TBD
SCK0	TBD
SCK1	TBD
SCKE	TBD
SCLK0	TBD
SCLK1	TBD
SLEEP	TBD
$\overline{\text{SMS0}}$	TBD
$\overline{\text{SMS1}}$	TBD
$\overline{\text{SMS2}}$	TBD
$\overline{\text{SMS3}}$	TBD
SUSPEND	TBD
$\overline{\text{SWE}}$	TBD
TCK	TBD
TDI	TBD
TDO	TBD
TFS0	TBD
TFS1	TBD
TMR0	TBD
TMR1	TBD
TMR2	TBD
TMS	TBD
$\overline{\text{TRST}}$	TBD
TSCLK0	TBD
TSCLK1	TBD
TX0	TBD
TX1	TBD
TXDPLS	TBD
TXDMNS	TBD
$\overline{\text{TXEN}}$	TBD

Table 12. 256-Lead PBGA Pin Assignment (Alphabetically By Signal) (Continued)

SIGNAL	PIN #
USB_CLK	TBD
XTALI	TBD
XTALO	TBD
XVER_DATA	TBD

OUTLINE DIMENSIONS

Dimensions in Figure 12 are shown in millimeters.

DRAWING VIEWS AND DIMENSIONS TBD

Figure 12. 256-Lead Metric Plastic Ball Grid Array (PBGA) (B-256)

ORDERING GUIDE

Part Number[1]	Case Temperature Range	Instruction Rate	Operating Voltage
ADSP-21535PKCA-300	0°C to 85°C	300 MHz	0.95 V to 1.575 V internal, 3.3 V-tolerant I/O

[1] B = Plastic Ball Grid Array (PBGA).