Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ANALOG DEVICES

Blackfin®
Embedded Processor

ADSP-BF535

KEY FEATURES
350 MHz High Performance Blackfin Processor Core
Two 16-Bit MACs, Two 40-Bit ALUs, One 40-Bit Shifter,
  Four 8-Bit Video ALUs, and Two 40-Bit Accumulators
RISC-Like Register and Instruction Model for Ease of
  Programming and Compiler Friendly Support
Advanced Debug, Trace, and Performance Monitoring
1.0 V–1.6 V Core V_{DD} with Dynamic Power Management
**3.3 V I/O**
260-Ball PBGA Package

MEMORY
308K Bytes of On-Chip Memory:
  16K Bytes of Instruction L1 SRAM/Cache
  32K Bytes of Data L1 SRAM/Cache
  4K Bytes of Scratch Pad L1 SRAM
  256K Bytes of Full Speed, Low Latency L2 SRAM
Memory DMA Controller

Memory Management Unit for 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)
Four Timer/Counters, Three with PWM Support
Sixteen Bidirectional Programmable Flag I/O Pins
Watchdog Timer
Real-Time Clock
On-Chip PLL with 1× to 31× Frequency Multiplier

FUNCTIONAL BLOCK DIAGRAM
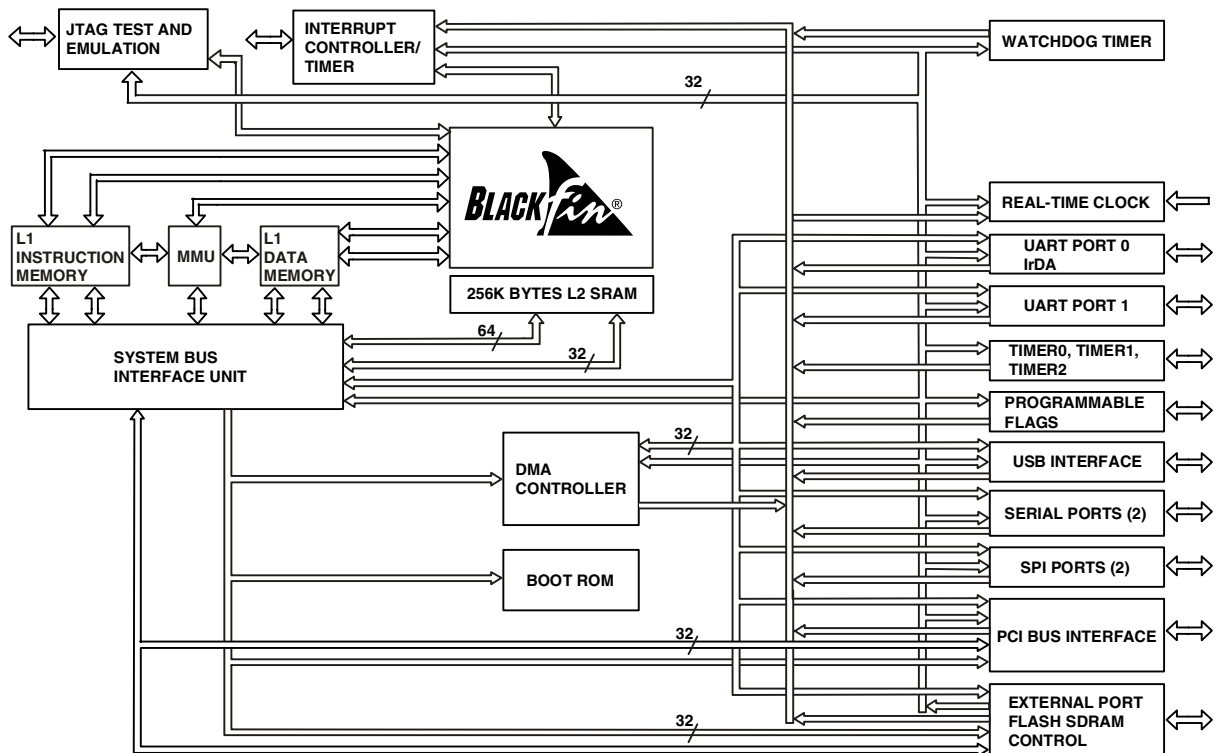
Blackfin and the Blackfin logo are registered trademarks of Analog Devices, Inc.

REV. A

ADSP-BF535* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

COMPARABLE PARTS

View a parametric search of comparable parts.

EVALUATION KITS

- USB-Based Emulator and High Performance USB-Based Emulator

DOCUMENTATION

Application Notes

- EE-104: Setting Up Streams with the VisualDSP Debugger
- EE-110: A Quick Primer on ELF and DWARF File Formats
- EE-112: Class Implementation in Analog C++
- EE-120: Interfacing Assembly Language Programs to C
- EE-126: The ABCs of SDRAMemories
- EE-128: DSP in C++: Calling Assembly Class Member Functions From C++
- EE-149: Tuning C Source Code for the Blackfin® Processor Compiler
- EE-159: Initializing DSP System & Control Registers From C and C++
- EE-162: Interfacing the ADSP-21535 to AD9860/2 High-Speed Converters over the External Memory Bus
- EE-172: Using the Dynamic Power Management Functionality of the ADSP-BF535 Blackfin® Processor
- EE-175: Emulator and Evaluation Hardware Troubleshooting Guide for VisualDSP++ Users
- EE-181: Interfacing the ADSP-BF535 Blackfin® Processor to Single-Chip CIF Digital Camera "OV6630" Over the External Memory Bus
- EE-183: Rational Sample Rate Conversion with Blackfin® Processors
- EE-184: Interfacing EPSON S1D13806 memory display controller to Blackfin® Processors
- EE-185: Fast Floating-Point Arithmetic Emulation on Blackfin® Processors
- EE-192: Using C To Create Interrupt-Driven Systems On Blackfin® Processors
- EE-193: Interfacing the ADSP-BF535 Blackfin® Processor to the AD73322L Codec
- EE-196: ADSP-BF535 Blackfin® EZ-KIT Lite™ CompactFlash® Interface
- EE-203: Interfacing the ADSP-BF535/ADSP-BF533 Blackfin® Processor to NTSC/PAL video decoder over the asynchronous port.
- EE-204: Blackfin® Processor SCCB Software Interface for Configuring I2C® Slave Devices
- EE-206: ADSP-BF535 Blackfin Processor PCI Interface Performance
- EE-207: Using the ADSP-BF535 Blackfin Processor's PCI interface in the Device Mode

ADSP-BF535* PRODUCT PAGE QUICK LINKS

- EE-210: SDRAM Selection and Configuration Guidelines for ADI Processors
- EE-213: Host Communication via the Asynchronous Memory Interface for Blackfin® Processors
- EE-214: Ethernet Network Interface for ADSP-BF535 Blackfin® Processors
- EE-235: An Introduction to Scripting in VisualDSP++®
- EE-242: PWM and Class-D Amplifiers with ADSP-BF535 Blackfin® Processors
- EE-261: Understanding Jitter Requirements of PLL-Based Processors
- EE-269: A Beginner's Guide to Ethernet 802.3
- EE-273: Using the VisualDSP++ Command-Line Installer
- EE-281: Hardware Design Checklist for the Blackfin® Processors
- EE-302: Interfacing ADSP-BF53x Blackfin® Processors to NAND FLASH Memory
- EE-303: Using VisualDSP++® Thread-Safe Libraries with a Third-Party RTOS
- EE-304: Using the Blackfin® Processor SPORT to Emulate a SPI Interface
- EE-306: PGO Linker - A Code Layout Tool for Blackfin Processors
- EE-307: Blackfin® Processor Troubleshooting Tips Using VisualDSP++® Tools
- EE-312: Building Complex VDK/LwIP Applications Using Blackfin® Processors
- EE-323: Implementing Dynamically Loaded Software Modules
- EE-324: System Optimization Techniques for Blackfin® Processors
- EE-325: Interfacing Atmel Fingerprint Sensor AT77C104B with Blackfin® Processors
- EE-326: Blackfin® Processor and SDRAM Technology
- EE-330: Windows Vista Compatibility in VisualDSP++ 5.0 Development Tools
- EE-332: Cycle Counting and Profiling
- EE-333: Interfacing Blackfin® Processors to Winbond W25X16 SPI Flash Devices
- EE-334: Using Blackfin® Processor Hibernate State for Low Standby Power
- EE-336: Putting ADSP-BF54x Blackfin® Processor Booting into Practice
- EE-339: Using External Switching Regulators with Blackfin® Processors
- EE-340: Connecting SHARC® and Blackfin® Processors over SPI

- EE-347: Formatted Print to a UART Terminal with Blackfin® Processors
- EE-350: Seamlessly Interfacing MEMS Microphones with Blackfin Processors
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users
- EE-68: Analog Devices JTAG Emulation Technical Reference

Data Sheet

- ADSP-BF535: Embedded Processor Data Sheet

Emulator Manuals

- HPUSB, USB, and HPPCI Emulator User's Guide
- ICE-1000/ICE-2000 Emulator User's Guide
- ICE-100B Emulator User's Guide

Evaluation Kit Manuals

- Blackfin® USB-LAN EZ-Extender® Manual
- Blackfin®/SHARC® USB EZ-Extender® Manual

Integrated Circuit Anomalies

- ADSP-BF535 Anomaly List for Revision(s) 0.2, 1.0, 1.1, 1.2, 1.3

Processor Manuals

- ADSP-BF535 Blackfin ® Processor Hardware Reference
- ADSP-BF5xx/ADSP-BF60x Blackfin® Processor Programming Reference
- Blackfin Processors: Manuals

Software Manuals

- CrossCore® Embedded Studio 2.5.0 Assembler and Preprocessor Manual
- CrossCore® Embedded Studio 2.5.0 C/C++ Compiler and Library Manual for Blackfin Processors
- CrossCore® Embedded Studio 2.5.0 Linker and Utilities Manual
- CrossCore® Embedded Studio 2.5.0 Loader and Utilities Manual
- CrossCore® Software Licensing Guide
- lwIP for CrossCore® Embedded Studio 1.0.0 User's Guide
- VisualDSP++® 5.0 Assembler and Preprocessor Manual
- VisualDSP++® 5.0 C/C++ Compiler and Library Manual for Blackfin Processors
- VisualDSP++® 5.0 Device Drivers and System Services Manual for Blackfin Processors
- VisualDSP++® 5.0 Kernel (VDK) Users Guide
- VisualDSP++® 5.0 Licensing Guide
- VisualDSP++® 5.0 Linker and Utilities Manual

- VisualDSP++® 5.0 Loader and Utilities Manual
- VisualDSP++® 5.0 Product Release Bulletin
- VisualDSP++® 5.0 Quick Installation Reference Card
- VisualDSP++® 5.0 Users Guide

SOFTWARE AND SYSTEMS REQUIREMENTS 🗖

- Software and Tools Anomalies Search

TOOLS AND SIMULATIONS 🗖

- Blackfin Processors Software and Tools

REFERENCE MATERIALS 🗖

Product Selection Guide

- ADI Complementary Parts Guide - Supervisory Devices and DSP Processors

DESIGN RESOURCES 🗖

- ADSP-BF535 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

DISCUSSIONS 🗖

View all ADSP-BF535 EngineerZone Discussions.

SAMPLE AND BUY 🗖

Visit the product page to see pricing options.

TECHNICAL SUPPORT 🗖

Submit a technical question or find your regional support number.

DOCUMENT FEEDBACK 🗖

Submit feedback for this data sheet.

ADSP-BF535

GENERAL DESCRIPTION

The ADSP-BF535 processor is a member of the Blackfin processor 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 signal processing 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 processors are the platform of choice for next generation applications that require RISC-like programmability, multimedia support, and leading edge signal processing in one integrated package.

Portable Low Power Architecture

Blackfin processors provide world class power management and performance. Blackfin processors are designed in a low power and low voltage design methodology and feature dynamic power management, the ability to independently vary both the voltage and frequency of operation to significantly lower overall power consumption. Varying the voltage and frequency can result in a substantial 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-BF535 Blackfin processor 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 signal processing core, users can develop cost-effective solutions quickly without the need for costly external components. The ADSP-BF535 Blackfin processor 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-BF535 Peripherals

The ADSP-BF535 Blackfin processor 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 Modulation) 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-BF535 Blackfin processor 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 it contains 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 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.

Processor Core

As shown in Figure 1, the Blackfin processor 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. Quad 16-bit operations can be accomplished simply, by taking advantage of the second ALU. This accelerates the per cycle throughput.
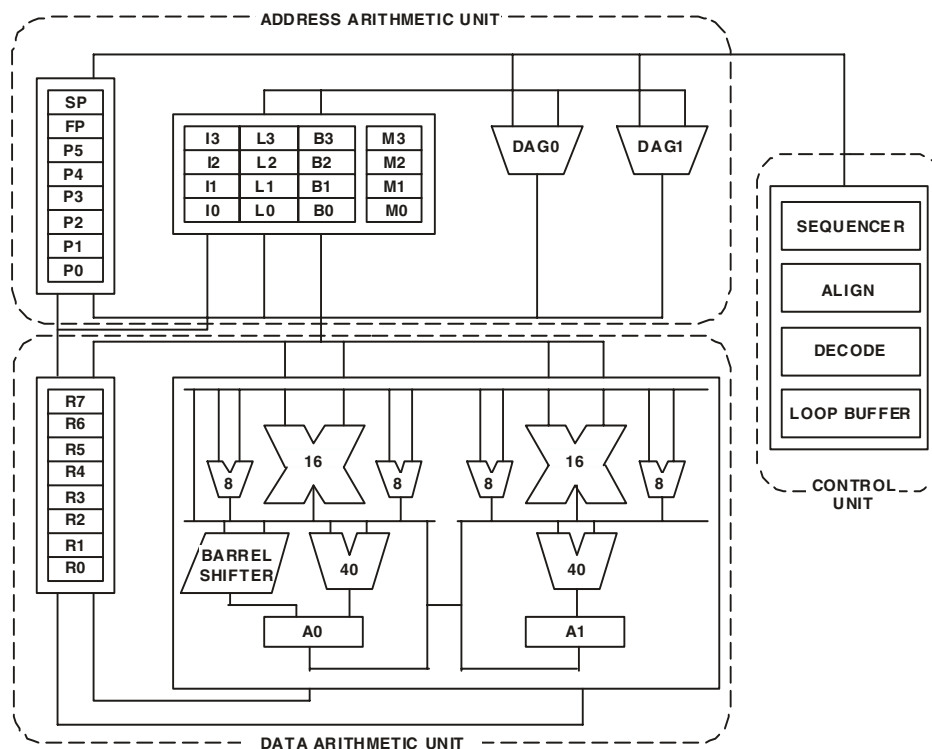

Figure 1. Processor Core

ADSP-BF535

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

The data for the computational units is found in a multiported 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 tightly 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 processors 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 scratch pad 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.
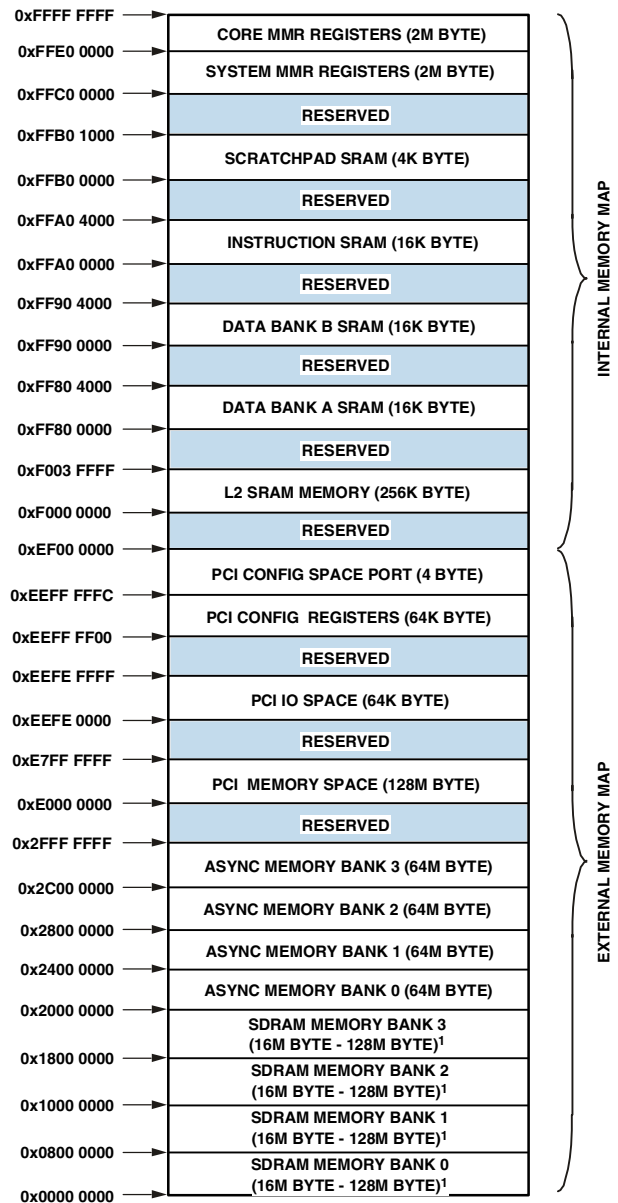
The Blackfin processor instruction set has been optimized so that 16-bit op-codes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit op-codes, representing fully featured multifunction instructions. Blackfin processors support a limited multiple 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 processor assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the C/C++ compiler, resulting in fast and efficient software implementations.

Memory Architecture

The ADSP-BF535 Blackfin processor views memory as a single unified 4 Gbyte 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 with very fast, low latency memory as cache or SRAM very close to the processor; and larger, lower cost, and lower performance memory systems farther away from the processor. See Figure 2.

Address	Region
0xFFFF FFFF	CORE MMR REGISTERS (2M BYTE)
0xFFE0 0000	SYSTEM MMR REGISTERS (2M BYTE)
0xFFC0 0000	RESERVED
0xFFB0 1000	SCRATCHPAD SRAM (4K BYTE)
0xFFB0 0000	RESERVED
0xFFA0 4000	INSTRUCTION SRAM (16K BYTE)
0xFFA0 0000	RESERVED
0xFF90 4000	DATA BANK B SRAM (16K BYTE)
0xFF90 0000	RESERVED
0xFF80 4000	DATA BANK A SRAM (16K BYTE)
0xFF80 0000	RESERVED
0xF003 FFFF	L2 SRAM MEMORY (256K BYTE)
0xF000 0000	RESERVED
0xEF00 0000	PCI CONFIG SPACE PORT (4 BYTE)
0xEEFF FFFC	PCI CONFIG REGISTERS (64K BYTE)
0xEEFF FF00	RESERVED
0xEEFE FFFF	PCI IO SPACE (64K BYTE)
0xEEFE 0000	RESERVED
0xE7FF FFFF	PCI MEMORY SPACE (128M BYTE)
0xE000 0000	RESERVED
0x2FFF FFFF	ASYNC MEMORY BANK 3 (64M BYTE)
0x2C00 0000	ASYNC MEMORY BANK 2 (64M BYTE)
0x2800 0000	ASYNC MEMORY BANK 1 (64M BYTE)
0x2400 0000	ASYNC MEMORY BANK 0 (64M BYTE)
0x2000 0000	SDRAM MEMORY BANK 3 (16M BYTE - 128M BYTE)[1]
0x1800 0000	SDRAM MEMORY BANK 2 (16M BYTE - 128M BYTE)[1]
0x1000 0000	SDRAM MEMORY BANK 1 (16M BYTE - 128M BYTE)[1]
0x0800 0000	SDRAM MEMORY BANK 0 (16M BYTE - 128M BYTE)[1]
0x0000 0000	

[1] THE ADDRESSES SHOWN FOR THE SDRAM BANKS REFLECT A FULLY POPULATED SDRAM ARRAY WITH 512M BYTES OF MEMORY. IF ANY BANK CONTAINS LESS THAN 128M BYTES OF MEMORY, THAT BANK WOULD EXTEND ONLY TO THE LENGTH OF THE REAL MEMORY SYSTEMS, AND THE END ADDRESS WOULD BECOME THE START ADDRESS OF THE NEXT BANK. THIS WOULD CONTINUE FOR ALL FOUR BANKS, WITH ANY REMAINING SPACE BETWEEN THE END OF MEMORY BANK 3 AND THE BEGINNING OF ASYNC MEMORY BANK 0, AT ADDRESS 0x2000 0000, TREATED AS RESERVED ADDRESS SPACE.

Figure 2. Internal/External Memory Map

REV. A

The L1 memory system is the primary highest performance memory available to the Blackfin processor 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 external 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-BF535 Blackfin processor 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 2-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 scratch pad 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 processor core has a dedicated low latency 64-bit wide datapath port into the L2 SRAM memory.

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 64 Mbyte 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-BF535 Blackfin processor memory space. These spaces are PCI memory, PCI I/O, and PCI configuration.

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-BF535 is functioning as an intelligent I/O device on the PCI bus.

When the ADSP-BF535 Blackfin processor 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 4 Gbyte space that appears on the PCI bus and can be used to map memory I/O devices on the bus. The ADSP-BF535 Blackfin processor uses a 128 Mbyte 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 Gbyte PCI memory space while its position with respect to the processor addresses remains fixed.

The PCI I/O region is also a 4 Gbyte space. However, most systems and I/O devices only use a 64 Kbyte subset of this space for I/O mapped addresses. The ADSP-BF535 Blackfin processor 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. This address space is a very low performance communication mode between the processor and PCI devices. The ADSP-BF535 Blackfin processor 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 processors 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 4 Gbyte 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 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.

ADSP-BF535

Booting

The ADSP-BF535 Blackfin processor contains a small boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF535 Blackfin processor is configured to boot from boot ROM memory space, the processor 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-BF535 Blackfin processor handles all asynchronous and synchronous events to the processor. The ADSP-BF535 Blackfin processor 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 shutdown of the system.

- Exceptions—Events that occur synchronously to program flow, for example, the exception will be taken before the instruction is allowed to complete. Conditions such as data alignment violations, undefined instructions, and so on, cause exceptions.

- Interrupts—Events that occur asynchronously to program flow. They are caused by timers, peripherals, input pins, explicit software instructions, and so on.

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

The ADSP-BF535 Blackfin processor 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-BF535 Blackfin processor. 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-BF535 Blackfin processor 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

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

Peripheral Interrupt Event	Peripheral Interrupt ID	Default Mapping
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-BF535 Blackfin processor 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 is 16 bits wide, and 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 CEC 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. (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 core 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 core clock cycles; however, the latency can be much higher, depending on the activity within and the mode of the processor.

DMA Controllers

The ADSP-BF535 Blackfin processor has multiple, independent DMA controllers that support automated data transfers with minimal overhead for the Blackfin processor core. DMA transfers can occur between the ADSP-BF535 Blackfin processor'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-BF535 Blackfin processor 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-BF535 Blackfin processor 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.

ADSP-BF535

External Memory Control

The External Bus Interface Unit (EBIU) on the ADSP-BF535 Blackfin processor 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 (Dual Inline Memory Module), 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 a system design where the configuration can be upgraded after delivery 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.

The SDRAM controller address, data, clock, and command pins can drive loads up to 50 pF. For larger memory systems, the SDRAM controller external buffer timing should be selected and external buffering should be provided so that the load on the SDRAM controller pins does not exceed 50 pF.

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 64 Mbyte window in the processor's address space but, if not fully populated, these windows 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-BF535 Blackfin processor 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 V 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-BF535 Blackfin processor supports two PCI functions:

- A host to PCI bridge function, in which the ADSP-BF535 Blackfin processor resources (the processor core, internal and external memory, and the memory DMA controller) provide the necessary hardware components to emulate a host computer PCI interface, from the perspective of a PCI target device.

- A PCI target function, in which an ADSP-BF535 Blackfin processor 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-BF535 Blackfin processor provides the necessary PCI host (platform) functions required to support and control a variety of off-the-shelf PCI I/O devices (for example, Ethernet controllers, bus bridges, and so on) in a system in which the ADSP-BF535 Blackfin processor is the host.

Note that the Blackfin processor architecture defines only memory space (no I/O or configuration address spaces). The three address spaces of PCI space (memory, I/O, and configuration space) are mapped into the flat 32-bit memory space of the ADSP-BF535 Blackfin processor. Because the PCI memory space is as large as the ADSP-BF535 Blackfin processor memory address space, a windowed approach is employed, with separate windows in the ADSP-BF535 Blackfin processor 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 the windows remain fixed in position in the ADSP-BF535 Blackfin processor's address range.

For devices on the PCI bus viewing the ADSP-BF535 Blackfin processor's resources, several mapping registers are provided to enable resources to be viewed in the PCI address space. The ADSP-BF535 Blackfin processor'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-BF535 Blackfin processor subsystem during enumeration of the PCI bus system. Once configured, the ADSP-BF535 Blackfin processor 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-BF535 Blackfin processor 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-BF535 Blackfin processor 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.

The USB device requires an external 48 MHz oscillator. The value of SCLK must always exceed 48 MHz for proper USB operation.

Real-Time Clock

The ADSP-BF535 Blackfin processor 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-BF535 Blackfin processor. 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: one is for a time of day, the second 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-BF535 Blackfin 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.
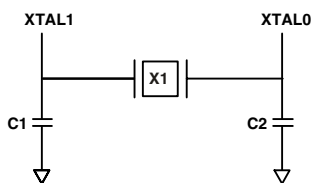
SUGGESTED COMPONENTS:
ECLIPTEK EC38J (THROUGH-HOLE PACKAGE)
EPSON MC405 12pF LOAD (SURFACE-MOUNT PACKAGE)
C1 = 22pF
C2 = 22pF

NOTE: C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1. CONTACT CRYSTAL MANUFACTURER FOR DETAILS. C1 AND C2 SPECIFICATIONS ASSUME BOARD TRACE CAPACITANCE OF 3pF.

Figure 3. External Components for RTC

Watchdog Timer

The ADSP-BF535 Blackfin processor 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 because of external noise conditions or a 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-BF535 Blackfin processor. 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 autobaud 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 the generation of operating system periodic interrupts.

Serial Ports (Sports)

The ADSP-BF535 Blackfin processor incorporates two complete synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support these 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 processor 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.

ADSP-BF535

- 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, with either of two pulse widths 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 Blackfin 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 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 the 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-BF535 Blackfin processor 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 ($\overline{SPISSx}$) let other SPI devices select the processor, and fourteen SPI chip select output pins (SPIxSEL7–1) let the processor 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 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 processor 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 processor's memory space (optional in DMA mode only).

4. Enables the SPI DMA engine and specifies transfer direction (optional in DMA mode only).

5. Reads or writes the SPI port receive or transmit data buffer (in non-DMA mode only).

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

In slave mode, the processor 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 processor'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 data on the appropriate SPI SCKx edges after receiving an SPI chip select on an $\overline{SPISSx}$ 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 processor can continue, by queuing up the next command TCB.

A slave mode transmit operation is similar, except the processor 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 is not ready to transmit, it can transmit a "zero" word.

UART Port
The ADSP-BF535 Blackfin processor 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.

- 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 receive. The 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:

- Bit rates ranging from (f_{SCLK}/1048576) to (f_{SCLK}/16) bits per second

- Data formats from 7 to 12 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

Autobaud detection is supported, in conjunction with the general-purpose timer functions.

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-BF535 Blackfin processor has 16 bidirectional, 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 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-BF535 Blackfin processor 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-BF535 Blackfin processor 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-BF535 Blackfin processor 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.

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 sleep mode, assertion of any interrupt will cause the processor to sense the value of the bypass bit

ADSP-BF535

(BYPASS) in the PLL Control register (PLL_CTL). If bypass is disabled, the processor transitions to the full on mode. If bypass is enabled, the processor transitions to the Active mode.

When in 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 peripherals (SCLK). Asynchronous peripherals, 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 full on mode. If bypass is enabled, the processor will transition to 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 following sections 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.

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 +	Disabled		Disabled	Disabled
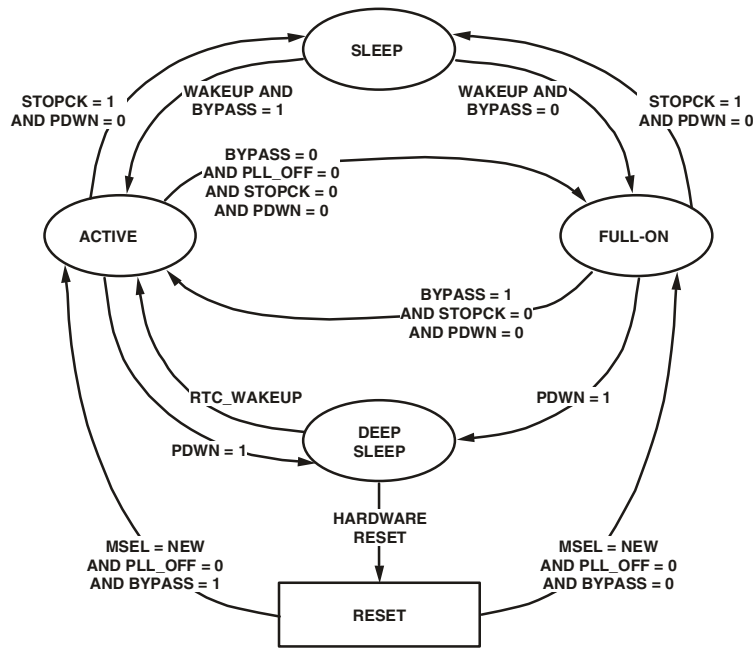
Figure 6. Mode Transitions

Power Savings

As shown in Table 4, the ADSP-BF535 Blackfin processor 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-BF535 Blackfin processor 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	V_{DD} 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_{CCLK}) to be dynamically and independently controlled.

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

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

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.

Peripheral Power Control

The ADSP-BF535 Blackfin processor 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 these 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-BF535 Blackfin processor 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 processor CLKIN pin. The CLKIN input cannot be halted, changed, or operated below the specified frequency during normal operation. This clock signal should be a 3.3 V LVTTL compatible signal. The processor provides a user-programmable 1× to 31× multiplication of the input clock to support external-to-internal clock ratios. The MSEL6–0, BYPASS, and DF pins decide the PLL multiplication factor at reset. At run time, the multiplication factor can be controlled in software. The combination of pull-up 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-BF535 Blackfin Processor Hardware Reference*.
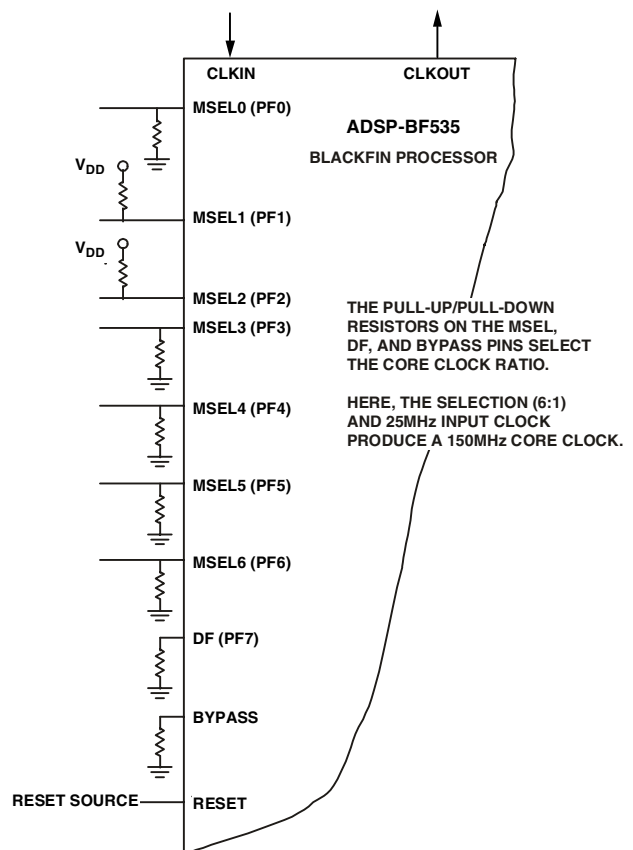
Figure 7. Clock Ratio Example

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

ADSP-BF535

SSEL fields define a divide ratio between the core clock (CCLK) and the system clock. Table 5 illustrates the system clock ratios. The system clock is supplied to the CLKOUT_SCLK0 pin.

Table 5. System Clock Ratios

Signal Name	Divider Ratio	Example Frequency Ratios (MHz)	
SSEL1–0	CCLK/SCLK	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 SSEL1 and SSEL0 pins 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-BF535 Blackfin Processor Hardware Reference*.

Booting Modes

The ADSP-BF535 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 these 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, a secondary software loader is provided that adds additional booting mechanisms. This secondary loader provides the capability to boot from PCI, 16-bit flash memory, fast flash, variable baud rate, and so on.

Instruction Set Description

The Blackfin processor family assembly language instruction set employs an algebraic syntax 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 processor 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 processor 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-BF535 Blackfin processor 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 processors, also fully emulates the ADSP-BF535 Blackfin processor.

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

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

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

- View mixed C/C++ and assembly code (interleaved source and object information)

- Insert breakpoints

- Set conditional breakpoints on registers, memory, and stacks

- Trace instruction execution

- View the internal pipeline to further optimize peripherals

- 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++ IDDE lets programmers define and manage 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 embedded, real-time programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning, when developing new application code. The VDK features include threads, critical and unscheduled regions, semaphores, events, and device flags. The VDK also supports priority-based, preemptive, cooperative, and time-sliced scheduling approaches. In addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

Because the VDK is a library, a developer can decide whether to use it or not. The VDK is integrated into the VisualDSP++ development environment, but can also be used via standard command line tools. When the VDK is used, the development environment assists the developer with many error-prone tasks and assists in managing system resources, automating the generation of various VDK based objects, and visualizing the system state, when debugging an application that uses the VDK.

VCSE is Analog Devices technology for creating, using, and reusing software components (independent modules of substantial functionality) to quickly and reliably assemble software applications. Download components from the Web and drop them into the application. Publish component archives from within VisualDSP++. VCSE supports component implementation in C/C++ or assembly language.

Use the Expert Linker to visually manipulate the placement of code and data on the embedded system. View memory utilization in a color-coded graphical form, easily move code and data to different areas of the processor or external memory with the drag of the mouse, examine run-time stack and heap usage. The Expert Linker is fully compatible with existing Linker Definition File (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices emulators use the IEEE 1149.1 JTAG test access port of the ADSP-BF535 Blackfin processor to monitor and control the target board processor during emulation. The emulator provides full speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusively in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect target system loading or timing.

VisualDSP++ is a trademark of Analog Devices, Inc.

ADSP-BF535

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

EZ-KIT Lite™ for ADSP-BF535 Blackfin Processor

The EZ-KIT Lite provides developers with a cost-effective method for initial evaluation of the ADSP-BF535 Blackfin processor. The EZ-KIT Lite includes a desktop evaluation board and fundamental debugging software to facilitate architecture evaluations via a PC hosted toolset. With the EZ-KIT Lite, users can learn more about Analog Devices hardware and software development tools and prototype applications. The EZ-KIT Lite includes an evaluation suite of the VisualDSP++ development environment with C/C++ compiler, assembler, and linker. The VisualDSP++ software included with the kit is limited in program memory size and limited to use with the EZ-KIT Lite product.

Designing an Emulator Compatible Processor Board (Target)

The Analog Devices family of emulators are tools that every system 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-BF535 Blackfin processor. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor system is set running at full speed with no impact on system timing.

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

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-BF535 Blackfin processor architecture and functionality. For detailed information on the Blackfin processor family core architecture and instruction set, refer to the *ADSP-BF535 Blackfin Processor Hardware Reference* and the *Blackfin Processor Instruction Set Reference*.

EZ-KIT Lite is a trademark of Analog Devices, Inc.

PIN DESCRIPTIONS

ADSP-BF535 Blackfin processor pin definitions are listed in
Table 7. The following pins are asynchronous: ARDY, PF15–0,
USB_CLK, NMI, $\overline{TRST}$, $\overline{RESET}$, PCI_CLK, XTALI,
XTALO.

Table 7. Pin Descriptions

Pin	Type	Function
ADDR25–2	O/T	External address bus.
DATA31–0	I/O/T	External data bus. (Pin has a logic-level hold circuit that prevents the input from floating internally.)
$\overline{ABE3-0}$/SDQM3–0	O/T	Asynchronous memory byte enables SDRAM data masks.
$\overline{AMS3-0}$	O/T	Chip selects for asynchronous memories.
ARDY[1]	I	Acknowledge signal for asynchronous memories.
$\overline{AOE}$	O/T	Memory output enable for asynchronous memories.
$\overline{ARE}$	O	Read enable for asynchronous memories.
$\overline{AWE}$	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.
$\overline{SRAS}$	O/T	SDRAM row address strobe pin. Connect to SDRAM's RAS pin.
$\overline{SCAS}$	O/T	SDRAM column address select pin. Connect to SDRAM's CAS pin.
$\overline{SWE}$	O/T	SDRAM write enable pin. Connect to SDRAM's WE or W buffer pin.
$\overline{SMS3-0}$	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.
PF15/$\overline{SPI1SEL7}$	I/O/T	Programmable flag pin. SPI output select pin.
PF14/$\overline{SPI0SEL7}$	I/O/T	Programmable flag pin. SPI output select pin.
PF13/$\overline{SPI1SEL6}$	I/O/T	Programmable flag pin. SPI output select pin.
PF12/$\overline{SPI0SEL6}$	I/O/T	Programmable flag pin. SPI output select pin.
PF11/$\overline{SPI1SEL5}$	I/O/T	Programmable flag pin. SPI output select pin.
PF10/$\overline{SPI0SEL5}$	I/O/T	Programmable flag pin. SPI output select pin (used during SPI boot).
PF9/$\overline{SPI1SEL4}$/SSEL1	I/O	Programmable flag pin. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF8/$\overline{SPI0SEL4}$/SSEL0	I/O	Programmable flag pin. SPI output select pin. Sampled during reset to determine core clock to system clock ratio.
PF7/$\overline{SPI1SEL3}$/DF	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/$\overline{SPI0SEL3}$/MSEL6	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/$\overline{SPI1SEL2}$/MSEL5	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.

Type column symbols: G = Ground, I = Input, O = Output, P = Power supply, T = Three-state

Table 7. Pin Descriptions (continued)

Pin	Type	Function
PF4/$\overline{\text{SPI0SEL2}}$/MSEL4	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/$\overline{\text{SPI1SEL1}}$/MSEL3	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/$\overline{\text{SPI0SEL1}}$/MSEL2	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/$\overline{\text{SPISS1}}$/MSEL1	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/$\overline{\text{SPISS0}}$/MSEL0	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	I/O/T	Receive serial clock for SPORT0.
RFS0	I/O/T	Receive frame synchronization for SPORT0.
DR0	I	Serial data receive for SPORT0.
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.
NMI	I	Non-maskable interrupt.
TCK	I	JTAG clock.
TDO	O/T	JTAG serial data out.
TDI	I	JTAG serial data in.
TMS	I	Test mode select.

Type column symbols: G = Ground, I = Input, O = Output, P = Power supply, T = Three-state

Table 7. Pin Descriptions (continued)

Pin	Type	Function
$\overline{\text{TRST}}$	I	JTAG reset.
$\overline{\text{RESET}}$	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 µs.
CLKIN1	I	Clock in.
BYPASS	I	Dedicated mode pin. May be permanently strapped to V_{DD} or V_{SS}. Bypasses the on-chip PLL.
DEEPSLEEP	O	Denotes that the Blackfin processor core is in Deep Sleep mode.
BMODE2–0	I	Dedicated mode pin. May be permanently strapped to V_{DD} or V_{SS}. Configures the boot mode that is employed following hardware reset or software reset.
PCI_AD31–0	I/O/T	PCI address and data bus.
$\overline{\text{PCI\_CBE3–0}}$	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.
$\overline{\text{PCI\_DEVSEL}}$	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 pull-up 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.
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.
$\overline{\text{PCI\_INTA}}$	I/O/T	PCI interrupt A line on PCI bus. Asserted by the ADSP-BF535 Blackfin processor as a device-to-signal an interrupt to the system processor. Monitored by the ADSP-BF535 when acting as the system processor.
$\overline{\text{PCI\_INTB}}$	I	PCI interrupt B line. Monitored by ADSP-BF535 Blackfin processor when acting as the system processor.
$\overline{\text{PCI\_INTC}}$	I	PCI interrupt C line. Monitored by the ADSP-BF535 Blackfin processor when acting as the system processor.
$\overline{\text{PCI\_INTD}}$	I	PCI interrupt D line. Monitored by the ADSP-BF535 Blackfin processor when acting as the system processor.
XTAL1	I	Real-Time Clock oscillator input.
XTAL0	O	Real-Time Clock oscillator output.
$\overline{\text{EMU}}$	O	Emulator acknowledge, open drain. Must be connected to the ADSP-BF535 Blackfin processor 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.

Type column symbols: G = Ground, I = Input, O = Output, P = Power supply, T = Three-state

ADSP-BF535

Unused Pins

Table 8 shows recommendations for tying off unused pins. All pins that are not listed in the table should be left floating.

Table 8. Recommendations for Tying Off Unused Pins

Pin	Tie Off
ARDY	V_{DDEXT}
BMODE2–0	V_{DDEXT} or GND
BYPASS	V_{DDEXT} or GND
DMNS	GND
DPLS	GND
DR0	V_{DDEXT} or GND
DR1	V_{DDEXT} or GND
NMI	GND
PCI_AD31–0	V_{DDEXT}
$\overline{PCI\_CB3-0}$	V_{DDEXT}
PCI_CLK	GND
$\overline{PCI\_DEVSEL}$	V_{DDEXT}
$\overline{PCI\_FRAME}$	V_{DDEXT}
$\overline{PCI\_GNT}$	V_{DDEXT}
PCI_IDSEL	GND
$\overline{PCI\_INTA}$	V_{DDEXT}
$\overline{PCI\_INTB}$	V_{DDEXT}
$\overline{PCI\_INTC}$	V_{DDEXT}
$\overline{PCI\_INTD}$	V_{DDEXT}
$\overline{PCI\_IRDY}$	V_{DDEXT}
$\overline{PCI\_LOCK}$	V_{DDEXT}
PCI_PAR	V_{DDEXT}
$\overline{PCI\_PERR}$	V_{DDEXT}
$\overline{PCI\_RST}$	V_{DDEXT}
$\overline{PCI\_STOP}$	V_{DDEXT}
$\overline{PCI\_SERR}$	V_{DDEXT}
$\overline{PCI\_TRDY}$	V_{DDEXT}
PF0/$\overline{SPISS0}$/MSEL0	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF1/$\overline{SPISS1}$/MSEL1	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF2/$\overline{SPI0SEL1}$/MSEL2	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF3/$\overline{SPI1SEL1}$/MSEL3	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF4/$\overline{SPI0SEL2}$/MSEL4	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF5/$\overline{SPI1SEL2}$/MSEL5	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF6/$\overline{SPI0SEL3}$/MSEL6	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF7/$\overline{SPI1SEL3}$/DF	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF8/$\overline{SPI0SEL4}$/SSEL0	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF9/$\overline{SPI1SEL4}$/SSEL1	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
RX0	V_{DDEXT} or GND
RX1	V_{DDEXT} or GND
TCK	V_{DDEXT}
TDI	V_{DDEXT}
TMS	V_{DDEXT}
$\overline{TRST}$	GND
USB_CLK	GND
$V_{DDPCIEXT}$	V_{DDEXT}
V_{DDRTC}	V_{DDEXT}
XTAL1	V_{DDEXT} or GND
XVER_DATA	GND

SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter		Min	Nominal	Max	Unit
V_{DDINT}	Internal (Core) Supply Voltage[1]				
	ADSP-BF535PKB-350	0.95	1.6	1.65	V
	ADSP-BF535PKB-300	0.95	1.5	1.575	V
	ADSP-BF535PBB-300	0.95	1.5	1.575	V
	ADSP-BF535PBB-200	0.95	1.5	1.575	V
V_{DDEXT}	External (I/O) Supply Voltage[1]	3.15	3.3	3.45	V
V_{DDPLL}	PLL Power Supply Voltage[1]	1.425	1.5	1.575	V
V_{DDRTC}	Real-Time Clock Power Supply Voltage[1]	2.60	3.3	3.45	V
$V_{DDPCIEXT}$	PCI I/O Power Supply Voltage[1]	3.15	3.3	3.45	V
V_{IH}	High Level Input Voltage[2], @ V_{DDEXT} = max	2.2		$V_{DDEXT} + 0.5$	V
V_{IL}	Low Level Input Voltage[2], @ V_{DDEXT} = min	−0.3		+0.6	V
$V_{IHUSBCLK}$	High Level Input Voltage[3], @ V_{DDEXT} = max	2.4		$V_{DDEXT} + 0.5$	V
V_{IHPCI}	High Level Input Voltage[4], @ $V_{DDPCIEXT}$ = max	$0.5 \times V_{DDPCIEXT}$		$V_{DDPCIEXT} + 0.5$	V
V_{ILPCI}	Low Level Input Voltage[4], @ $V_{DDPCIINT}$ = min	−0.5		$+0.3 \times V_{DDPCIEXT}$	V
T_A	Ambient Operating Temperature				°C
	Commercial	0		70	°C
	Industrial	−40		+85	°C

Specifications subject to change without notice.

[1] There is no requirement for sequencing of the voltage supplies on powerup, however, the supply regulators must be able to provide the required current $I_{DDRESET}$ at all times. See Table 26.
[2] Applies to input and bidirectional pins, except PCI and USB_CLK.
[3] Applies to USB_CLK.
[4] Applies to PCI input and bidirectional pins: PCI_AD31–0, PCI_CBE3–0, PCI_FRAME, PCI_IRDY, PCI_TRDY, PCI_DEVSEL, PCI_STOP, PCI_PERR, PCI_PAR, PCI_SERR, PCI_RST, PCI_GNT, PCI_IDSEL, PCI_LOCK, PCI_CLK, PCI_INTA, PCI_INTB, PCI_INTC, PCI_INTD.

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} = max, I_{OL} = 2.0 mA		0.4	V
V_{OHPCI}	PCI High Level Output Voltage[2]	@ $V_{DDPCIEXT}$ = min, I_{OH} = −0.5 mA	$0.9 \times V_{DDPCIEXT}$		V
V_{OLPCI}	PCI Low Level Output Voltage[2]	@ $V_{DDPCIEXT}$ = max, I_{OL} = 2.0 mA		$0.1 \times V_{DDPCIEXT}$	V
I_{IH}	High Level Input Current[3]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max		10	μA
I_{IL}	Low Level Input Current[3]	@ V_{DDEXT} = max, V_{IN} = 0 V		10	μA
I_{OZH}	Three-State Leakage Current[4]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max		10	μA
I_{OZL}	Three-State Leakage Current[4]	@ V_{DDEXT} = max, V_{IN} = 0 V		10	μA
C_{IN}	Input Capacitance[5, 6]	f_{IN} = 1 MHz, T_A = 25°C, V_{IN} = 2.5 V		5	pF

Specifications subject to change without notice.

[1] Applies to output and bidirectional pins, except PCI.
[2] Applies to PCI output and bidirectional pins: PCI_AD31–0, PCI_CBE3–0, PCI_FRAME, PCI_IRDY, PCI_TRDY, PCI_DEVSEL, PCI_STOP, PCI_PERR, PCI_PAR, PCI_REQ, PCI_SERR, PCI_RST, PCI_INTA.
[3] Applies to input pins.
[4] Applies to three-statable pins.
[5] Applies to all signal pins.
[6] Guaranteed but not tested.

ADSP-BF535

ABSOLUTE MAXIMUM RATINGS

Internal (Core) Supply Voltage (V_{DDINT})[1] . −0.3 V to +1.65 V
External (I/O) Supply Voltage (V_{DDEXT})[1] . . . −0.3 V to +4.0 V
Input Voltage[1] . . . . −0.5 V to V_{DDEXT} +0.5 V
Output Voltage Swing[1] . . . . −0.5 V to V_{DDEXT} +0.5 V
Load Capacitance[1, 2] . . . . 200 pF
Core Clock: [1]
    ADSP-BF535PKB-350 . . . . 350 MHz
    ADSP-BF535PKB-300 . . . . 300 MHz
    ADSP-BF535PBB-300 . . . . 300 MHz
    ADSP-BF535PBB-200 . . . . 200 MHz
System Clock (SCLK)[1] . . . . 133 MHz
Storage Temperature Range[1] . . . . −65°C to +150°C

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

[2] For proper SDRAM controller operation, the maximum load capacitance is 50 pF for ADDR, DATA, $\overline{ABE3–0}$/SDQM3–0, CLKOUT/SCLK1, SCLK0, SCKE, SA10, $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, and $\overline{SMS3-0}$.

ESD SENSITIVITY

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-BF535 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.

WARNING!
ESD SENSITIVE DEVICE

REV. A