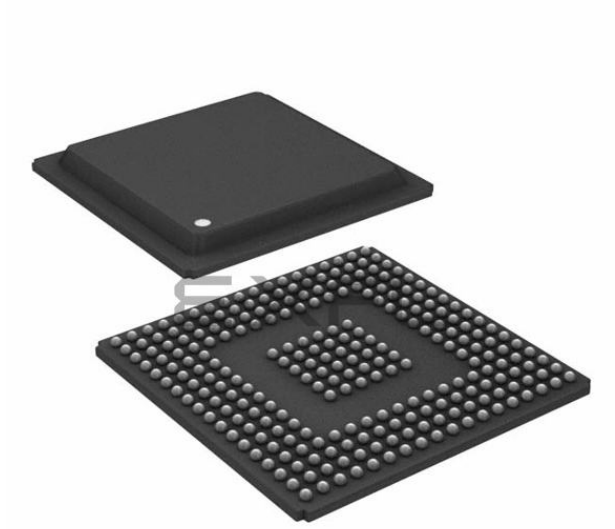

Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

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

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Obsolete
Type	Fixed Point
Interface	PCI, SPI, SSP, UART, USB
Clock Rate	200MHz
Non-Volatile Memory	External
On-Chip RAM	308kB
Voltage - I/O	3.30V
Voltage - Core	1.50V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	260-BBGA
Supplier Device Package	260-PBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf535pbb-200

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

- 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

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.

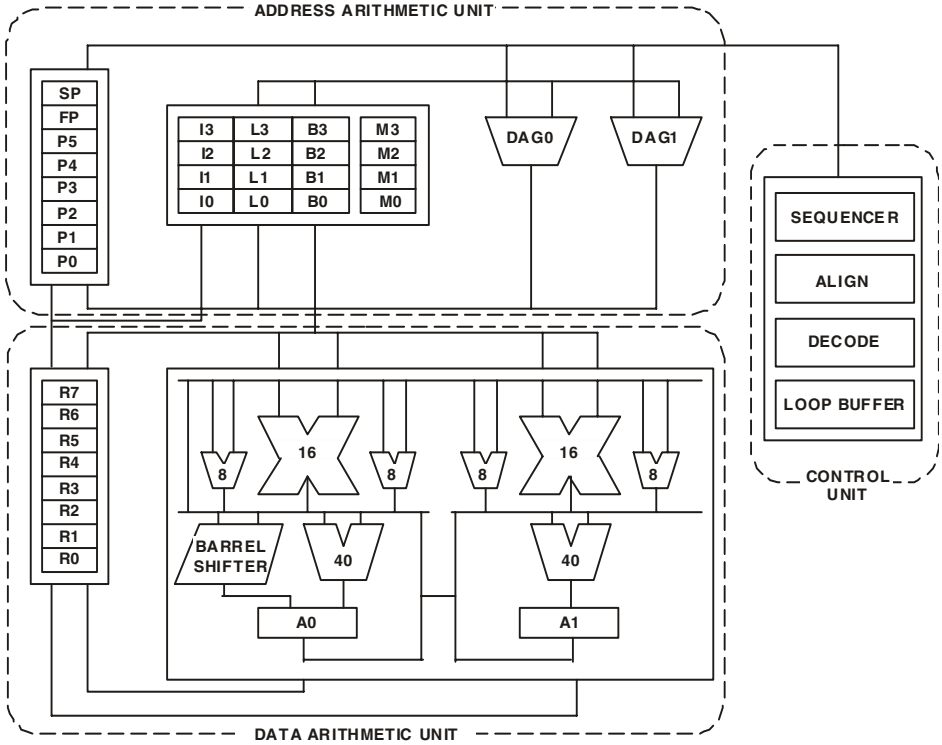

Figure 1. Processor Core

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

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

(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

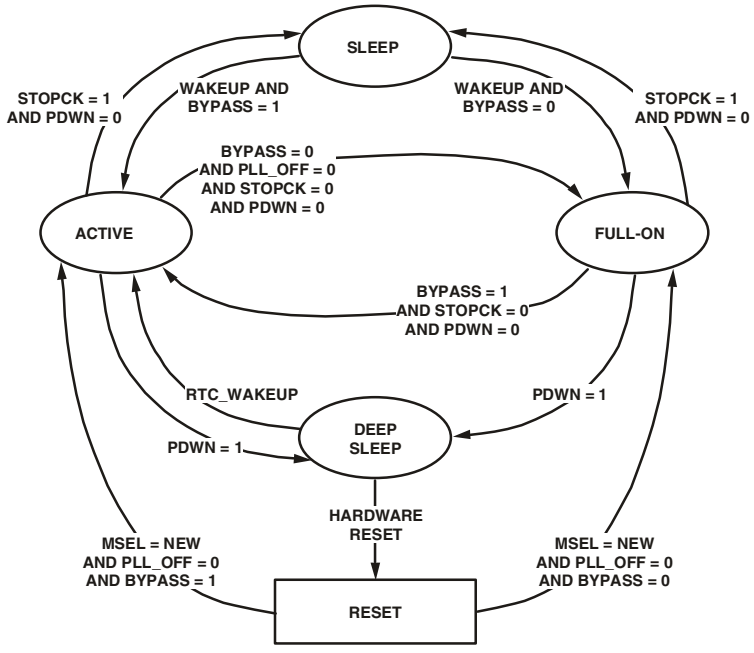

Figure 6. Mode Transitions

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}

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.

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.

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

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}
$\overline{\text{PCI\_CB3–0}}$	V_{DDEXT}
PCI_CLK	GND
$\overline{\text{PCI\_DEVSEL}}$	V_{DDEXT}
$\overline{\text{PCI\_FRAME}}$	V_{DDEXT}
$\overline{\text{PCI\_GNT}}$	V_{DDEXT}
PCI_IDSEL	GND
$\overline{\text{PCI\_INTA}}$	V_{DDEXT}
$\overline{\text{PCI\_INTB}}$	V_{DDEXT}
$\overline{\text{PCI\_INTC}}$	V_{DDEXT}
$\overline{\text{PCI\_INTD}}$	V_{DDEXT}
$\overline{\text{PCI\_IRDY}}$	V_{DDEXT}
$\overline{\text{PCI\_LOCK}}$	V_{DDEXT}
PCI_PAR	V_{DDEXT}
$\overline{\text{PCI\_PERR}}$	V_{DDEXT}
$\overline{\text{PCI\_RST}}$	V_{DDEXT}
$\overline{\text{PCI\_STOP}}$	V_{DDEXT}
$\overline{\text{PCI\_SERR}}$	V_{DDEXT}
$\overline{\text{PCI\_TRDY}}$	V_{DDEXT}
PF0/$\overline{\text{SPISS0}}$/MSEL0	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF1/$\overline{\text{SPISS1}}$/MSEL1	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF2/$\overline{\text{SPI0SEL1}}$/MSEL2	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF3/$\overline{\text{SPI1SEL1}}$/MSEL3	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF4/$\overline{\text{SPI0SEL2}}$/MSEL4	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF5/$\overline{\text{SPI1SEL2}}$/MSEL5	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF6/$\overline{\text{SPI0SEL3}}$/MSEL6	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF7/$\overline{\text{SPI1SEL3}}$/DF	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF8/$\overline{\text{SPI0SEL4}}$/SSEL0	V_{DDEXT} or GND (10 kΩ pull-up/pull-down required)
PF9/$\overline{\text{SPI1SEL4}}$/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}
$\overline{\text{TRST}}$	GND
USB_CLK	GND
$V_{DDPCIEXT}$	V_{DDEXT}
V_{DDRTC}	V_{DDEXT}
XTAL1	V_{DDEXT} or GND
XVER_DATA	GND

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 $\overline{SMS3\text{-}0}$.

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.

TIMING SPECIFICATIONS

Table 9 and Table 10 describe the timing requirements for the ADSP-BF535 Blackfin processor clocks. Take care in selecting MSEL and SSEL ratios so as not to exceed the maximum core clock, system clock and Voltage Controlled Oscillator (VCO) operating frequencies, as described in Absolute Maximum Ratings on Page 22. Table 10 describes phase-locked loop operating conditions.

Table 9. Core Clock Requirements

Parameter		Min	Max	Unit
$t_{CCLK1.6}$	Core Cycle Period (V_{DDINT} = 1.6 V–50 mV)	2.86	200	ns
$t_{CCLK1.5}$	Core Cycle Period (V_{DDINT} = 1.5 V–5%)	3.33	200	ns
$t_{CCLK1.4}$	Core Cycle Period (V_{DDINT} = 1.4 V–5%)	3.70	200	ns
$t_{CCLK1.3}$	Core Cycle Period (V_{DDINT} = 1.3 V–5%)	4.17	200	ns
$t_{CCLK1.2}$	Core Cycle Period (V_{DDINT} = 1.2 V–5%)	4.76	200	ns
$t_{CCLK1.1}$	Core Cycle Period (V_{DDINT} = 1.1 V–5%)	5.56	200	ns
$t_{CCLK1.0}$	Core Cycle Period (V_{DDINT} = 1.0 V–5%)	6.67	200	ns

Table 10. Phase-Locked Loop Operating Conditions

Parameter	Min	Nominal	Max	Unit
Operating Voltage	1.425	1.5	1.575	V
Jitter, Rising Edge to Rising Edge, Per Output[1]			120	ps
Jitter, Rising Edge to Falling Edge, Per Output[1]			60	ps
Skew, Rising Edge to Rising Edge, Any Two Outputs[1]			120	ps
Voltage Controlled Oscillator (VCO) Frequency[1]	40		400	MHz
V_{DDPLL} Induced Jitter[1]			1	ps/mV

[1] Guaranteed but not tested.

Clock and Reset Timing

Table 11 and Figure 8 describe clock and reset operations. Per ABSOLUTE MAXIMUM RATINGS on Page 22, combinations of CLKIN and clock multipliers must not select core and system clocks in excess of 350/300/200 MHz and 133 MHz, respectively.

Table 11. Clock and Reset Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{CKIN}	CLKIN Period	25.0	100.0	ns
t_{CKINL}	CLKIN Low Pulse[1]	10.0		ns
t_{CKINH}	CLKIN High Pulse[1]	10.0		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulse Width Low[2]	$11 \times t_{CKIN}$		ns
t_{MSD}	Delay from $\overline{\text{RESET}}$ Asserted to MSELx, SSELx, BYPASS, and DF Valid[3]		15.0	ns
t_{MSS}	MSELx/SSELx/DF/BYPASS Stable Setup Before $\overline{\text{RESET}}$ Deasserted[4]	$2 \times t_{CKIN}$		ns
t_{MSH}	MSELx/SSELx/DF/BYPASS Stable Hold After $\overline{\text{RESET}}$ Deasserted	$2 \times t_{CKIN}$		ns
Switching Characteristics				
t_{PFD}	Flag Output Disable Time After $\overline{\text{RESET}}$ Asserted		15.0	ns

[1] Applies to Bypass mode and Non-bypass mode.
[2] Applies after power-up sequence is complete. At power-up, the processor's internal phase-locked loop requires no more than 2000 CLKIN cycles, while $\overline{\text{RESET}}$ is asserted, assuming stable power supplies and CLKIN (not including start-up time of external clock oscillator).
[3] SSELx, MSELx and DF values can change from this point, but the values must be valid.
[4] SSELx, MSELx and DF values must be held from this time, until the hold time expires.

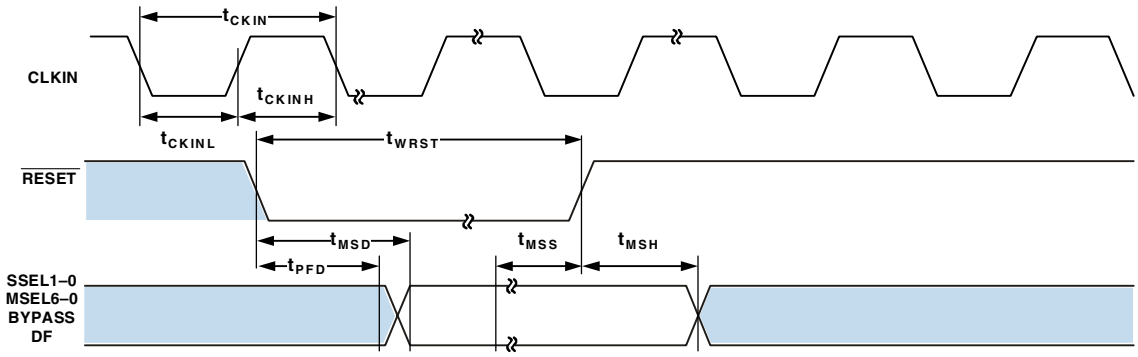

Figure 8. Clock and Reset Timing

Serial Ports

Table 17 through Table 22 and Figure 14 describe Serial Port timing.

Table 17. Serial Ports—External Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSE}	TFS/RFS Setup Before TCLK/RCLK[1]	3.0		ns
t_{HFSE}	TFS/RFS Hold After TCLK/RCLK[1]	3.0		ns
t_{SDRE}	Receive Data Setup Before RCLK[1]	3.0		ns
t_{HDRE}	Receive Data Hold Before RCLK[1]	3.0		ns
t_{SCLKWE}	TCLK/RCLK Width	$(0.5 \times t_{SCLKE}) - 1$		ns
t_{SCLKE}	TCLK/RCLK Period	$2 \times t_{SCLK}$		ns

[1]Referenced to sample edge.

Table 18. Serial Ports—Internal Clock

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

[1]Referenced to sample edge.

Table 19. Serial Ports—External or Internal Clock

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFSE}	RFS Delay After RCLK (Internally Generated RFS)[1]		10.0	ns
t_{HOFSE}	RFS Hold After RCLK (Internally Generated RFS)[1]	3.0		ns

[1]Referenced to drive edge.

Table 20. Serial Ports—External Clock

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFSE}	TFS Delay After TCLK (Internally Generated TFS)[1]		10.0	ns
t_{HOFSE}	TFS Hold After TCLK (Internally Generated TFS)[1]	3.0		ns
t_{DDTE}	Transmit Data Delay After TCLK[1]		10.0	ns
t_{HDTE}	Transmit Data Hold After TCLK[1]	3.0		ns

[1]Referenced to drive edge.

Table 21. Serial Ports—Internal Clock

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFSI}	TFS Delay After TCLK (Internally Generated TFS)[1]		6.0	ns
t_{HOFSI}	TFS Hold After TCLK (Internally Generated TFS)[1]	0.0		ns
t_{DDTI}	Transmit Data Delay After TCLK[1]		8.0	ns
t_{HDTI}	Transmit Data Hold After TCLK[1]	0.0		ns
t_{SCLKWI}	TCLK/RCLK Width	$0.5 \times t_{SCLK}$		ns

[1]Referenced to drive edge.

Serial Peripheral Interface (SPI) Port —Master Timing

Table 23 and Figure 15 describe SPI port master operations.

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

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSPID}	Data Input Valid to SCK Edge (Data Input Setup)	6.5		ns
t_{HSPID}	SCK Sampling Edge to Data Input Invalid	1.6		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIxSEL}}$ Low to First SCK Edge (x=0 or 1)	$(2 \times t_{SCLK}) - 3$		ns
t_{SPICHM}	Serial Clock High Period	$(2 \times t_{SCLK}) - 3$		ns
t_{SPICLM}	Serial Clock Low Period	$(2 \times t_{SCLK}) - 3$		ns
t_{SPICLK}	Serial Clock Period	$4 \times t_{SCLK}$		ns
t_{HDSM}	Last SCK Edge to $\overline{\text{SPIxSEL}}$ High (x=0 or 1)	$(2 \times t_{SCLK}) - 3$		ns
t_{SPITDM}	Sequential Transfer Delay	$2 \times t_{SCLK}$		ns
t_{DDSPID}	SCK Edge to Data Out Valid (Data Out Delay)	0.0	6.0	ns
t_{HDSPID}	SCK Edge to Data Out Invalid (Data Out Hold)	0.0	5.0	ns

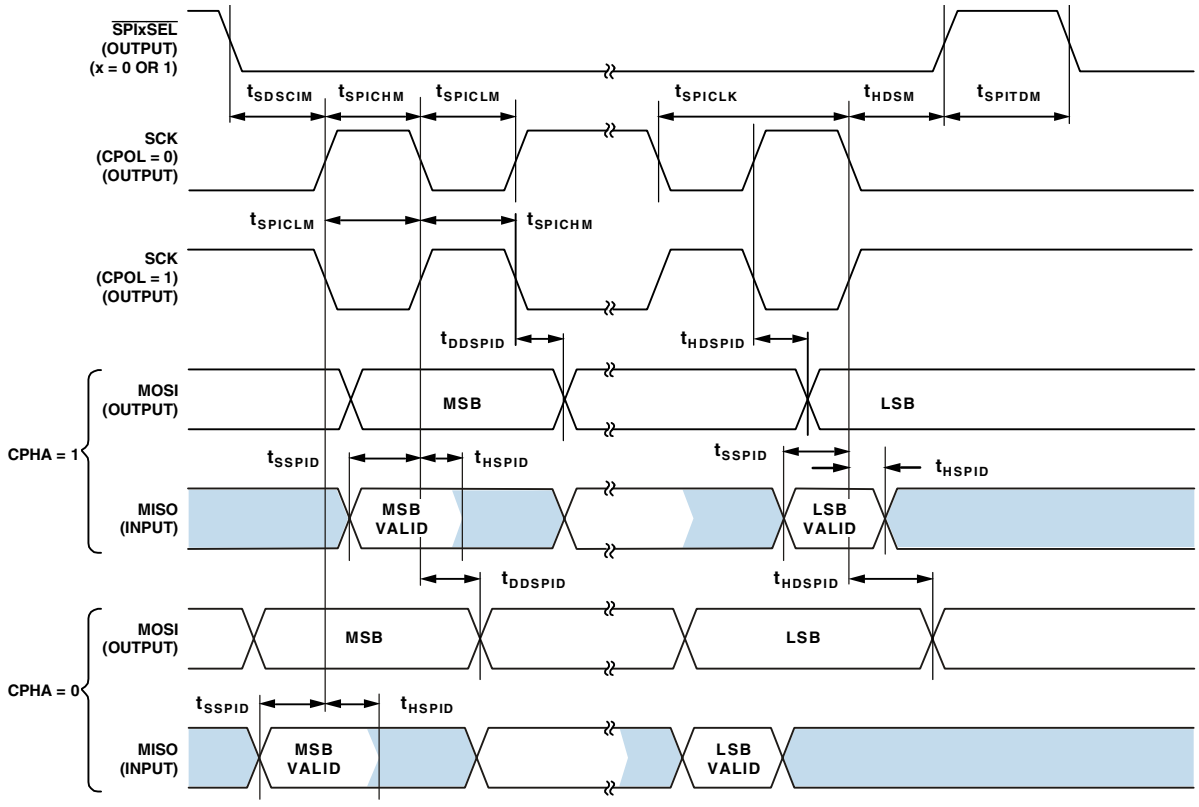

Figure 15. Serial Peripheral Interface (SPI) Port—Master Timing

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

Figure 17 describes UART port receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 17, there is some latency between the generation of internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

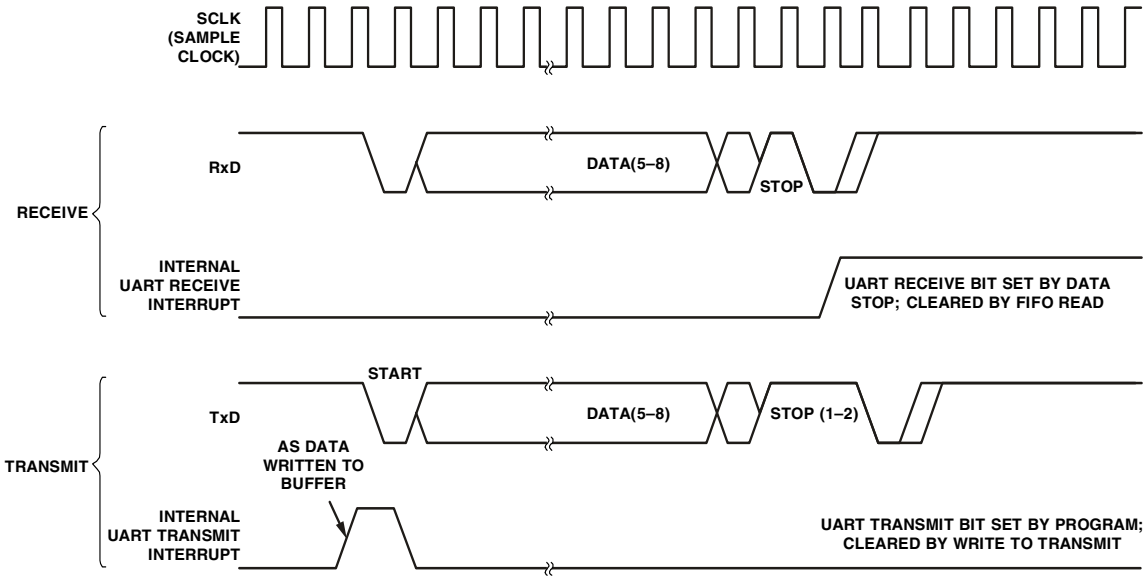

Figure 17. UART Port—Receive and Transmit Timing

Table 30. 260-Ball PBGA Pin Assignment (Numerically by Pin Number)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A01	ADDR12	D12	DEEPSLEEP	K01	SCLK0	R08	PF1/$\overline{\text{SPISS1}}$/MSEL1
A02	ADDR17	D13	$\overline{\text{PCI\_INTD}}$	K02	GND	R09	PF5/$\overline{\text{SPI1SEL2}}$/MSEL5
A03	ADDR15	D14	PCI_CLK	K03	$\overline{\text{SMS3}}$	R10	XTAL1
A04	ADDR8	D15	$\overline{\text{PCI\_PERR}}$	K04	V_{DDEXT}	R11	PF7/$\overline{\text{SPI1SEL3}}$/DF
A05	ADDR6	D16	$\overline{\text{PCI\_REQ}}$	K07	GND	R12	PF12/$\overline{\text{SPI0SEL6}}$
A06	ADDR2	D17	$\overline{\text{PCI\_CBE3}}$	K08	GND	R13	RSCLK0
A07	RX0	D18	$\overline{\text{PCI\_RST}}$	K09	GND	R14	DT0
A08	TX0	E01	$\overline{\text{AOE}}$	K10	GND	R15	TFS1
A09	XVER_DATA	E02	$\overline{\text{ABE0}}$/SDQM0	K11	GND	R16	SCK1
A10	V_{SSPLL}	E03	ADDR22	K12	GND	R17	PCI_AD30
A11	SUSPEND	E04	V_{DDEXT}	K15	$V_{DDPCIEXT}$	R18	PCI_AD22
A12	TMS	E15	$\overline{\text{PCI\_IRDY}}$	K16	PCI_AD17	T01	DATA2
A13	$\overline{\text{EMU}}$	E16	$\overline{\text{PCI\_CBE2}}$	K17	PCI_AD15	T02	DATA9
A14	BMODE1	E17	PCI_AD0	K18	PCI_AD10	T03	DATA14
A15	$\overline{\text{PCI\_INTC}}$	E18	PCI_AD1	L01	SCKE	T04	DATA15
A16	$\overline{\text{PCI\_LOCK}}$	F01	$\overline{\text{ARE}}$	L02	$\overline{\text{SRAS}}$	T05	DATA20
A17	$\overline{\text{PCI\_STOP}}$	F02	$\overline{\text{AMS0}}$	L03	$\overline{\text{SCAS}}$	T06	DATA25
A18	N/C	F03	ADDR24	L04	V_{DDEXT}	T07	DATA30
B01	$\overline{\text{ABE1}}$/SDQM1	F04	V_{DDINT}	L07	GND	T08	PF2/$\overline{\text{SPI0SEL1}}$/MSEL2
B02	ADDR20	F15	$\overline{\text{PCI\_CBE1}}$	L08	GND	T09	PF6/$\overline{\text{SPI0SEL3}}$/MSEL6
B03	ADDR16	F16	$\overline{\text{PCI\_CBE0}}$	L09	GND	T10	XTAL0
B04	ADDR11	F17	PCI_AD3	L10	GND	T11	PF8/$\overline{\text{SPI0SEL4}}$/SSEL0
B05	ADDR7	F18	PCI_AD4	L11	GND	T12	PF11/$\overline{\text{SPI1SEL5}}$
B06	ADDR3	G01	$\overline{\text{AWE}}$	L12	V_{DDINT}	T13	PF15/$\overline{\text{SPI1SEL7}}$
B07	TMR0	G02	$\overline{\text{AMS3}}$	L15	$V_{DDPCIEXT}$	T14	TFS0
B08	RX1	G03	$\overline{\text{ABE2}}$/SDQM2	L16	PCI_AD21	T15	TSCLK1
B09	$\overline{\text{RESET}}$	G04	V_{DDEXT}	L17	PCI_AD20	T16	MISO0
B10	TXDPLS	G07	USB_CLK	L18	PCI_AD12	T17	MOSI1
B11	NMI	G08	V_{DDEXT}	M01	SA10	T18	PCI_AD23
B12	$\overline{\text{TRST}}$	G09	V_{DDPLL}	M02	$\overline{\text{SMS0}}$	U01	DATA7
B13	BMODE2	G10	TXDMNS	M03	DATA1	U02	DATA8
B14	BMODE0	G11	V_{DDINT}	M04	V_{DDEXT}	U03	DATA13
B15	$\overline{\text{PCI\_INTB}}$	G12	V_{DDINT}	M07	GND	U04	DATA16
B16	$\overline{\text{PCI\_SERR}}$	G15	V_{DDINT}	M08	V_{DDINT}	U05	DATA21
B17	$\overline{\text{PCI\_TRDY}}$	G16	PCI_AD2	M09	GND	U06	DATA24
B18	PCI_IDSEL	G17	PCI_AD6	M10	GND	U07	DATA28
C01	ADDR23	G18	PCI_AD5	M11	V_{DDINT}	U08	PF0/$\overline{\text{SPISS0}}$/MSEL0
C02	ADDR21	H01	CLKOUT/SCLK1	M12	V_{DDINT}	U09	PF4/$\overline{\text{SPI0SEL2}}$/MSEL4
C03	ADDR18	H02	GND	M15	$V_{DDPCIEXT}$	U10	V_{DDRTC}
C04	ADDR13	H03	$\overline{\text{AMS2}}$	M16	PCI_AD25	U11	PF9/$\overline{\text{SPI1SEL4}}$/SSEL1
C05	ADDR9	H04	V_{DDINT}	M17	PCI_AD24	U12	PF13/$\overline{\text{SPI1SEL6}}$
C06	ADDR5	H07	$\overline{\text{ABE3}}$/SDQM3	M18	PCI_AD14	U13	RFS0
C07	TMR1	H08	GND	N01	$\overline{\text{SMS2}}$	U14	RSCLK1
C08	TX1	H09	V_{DDINT}	N02	DATA0	U15	DR1
C09	DPLS	H10	GND	N03	DATA4	U16	MOSI0
C10	$\overline{\text{TXEN}}$	H11	GND	N04	V_{DDINT}	U17	SCK0
C11	TDI	H12	V_{DDINT}	N15	V_{DDINT}	U18	MISO1
C12	BYPASS	H15	$V_{DDPCIEXT}$	N16	PCI_AD29	V01	N/C
C13	GND	H16	PCI_AD11	N17	PCI_AD26	V02	DATA10

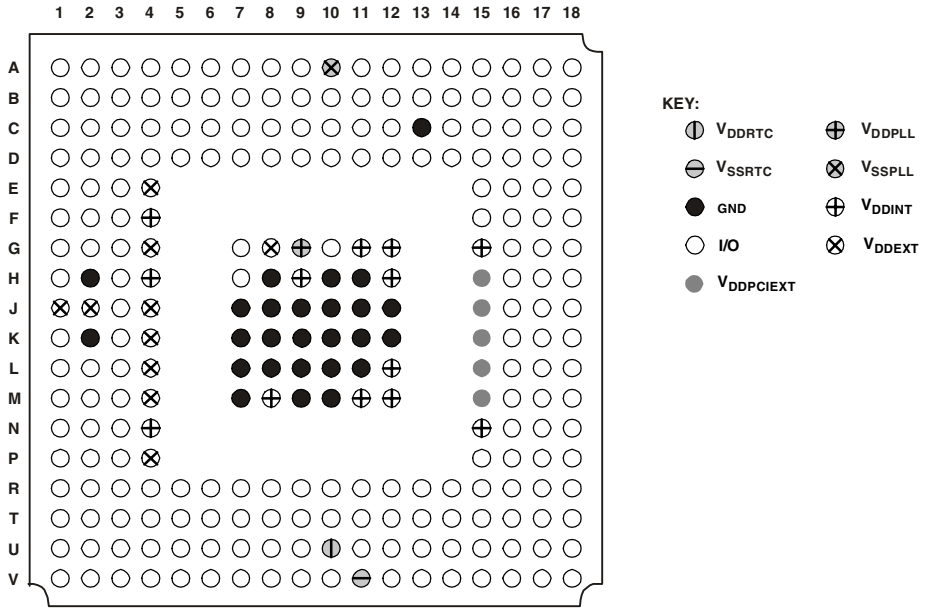

Figure 25. 260-Ball Metric PBGA Pin Configuration (Top View)

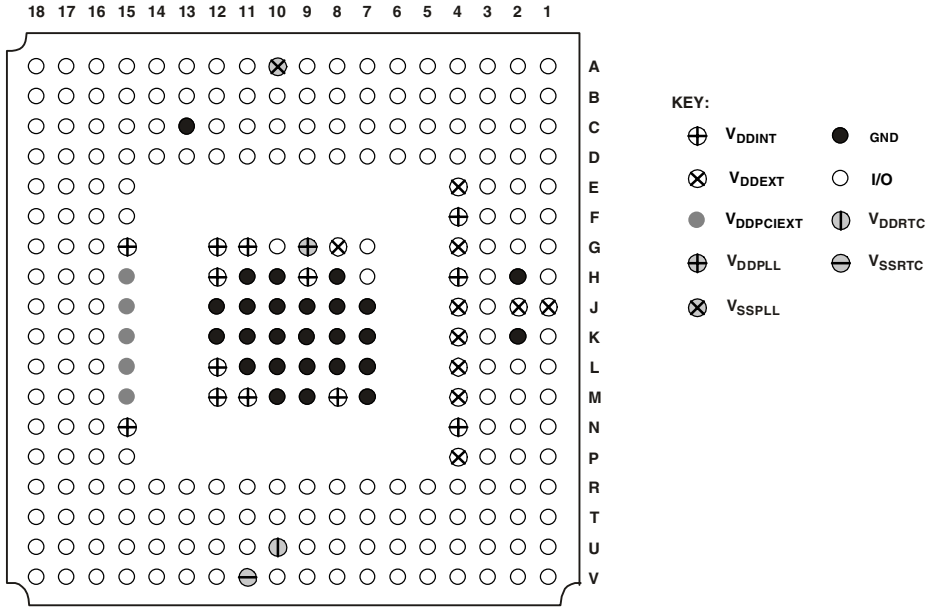

Figure 26. 260-Ball Metric PBGA Pin Configuration (Bottom View)

OUTLINE DIMENSIONS

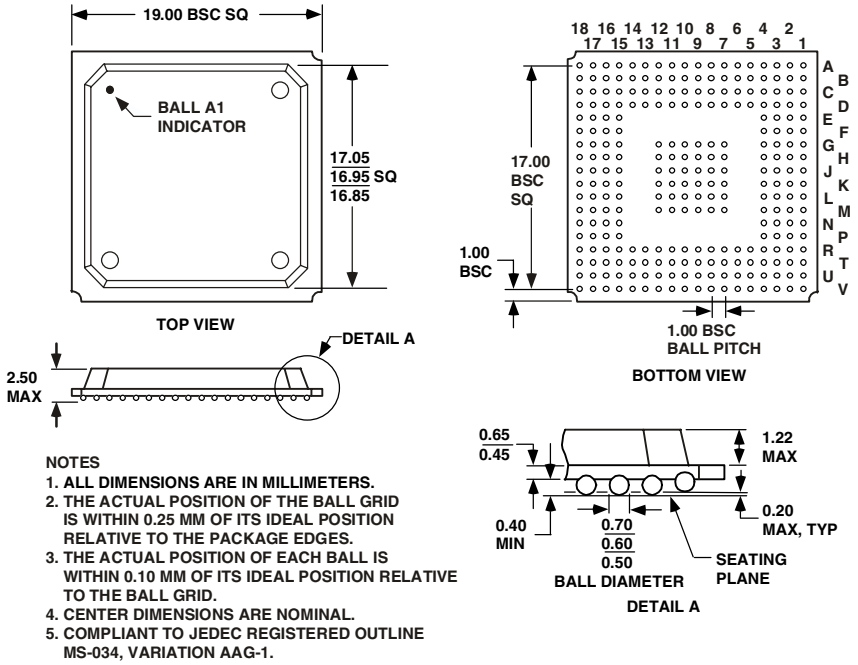

Figure 27. 260-Ball Metric Plastic Ball Grid Array (PBGA) (B-260)

ORDERING GUIDE

Part Number	Temperature Range (Ambient)	Instruction Rate	Operating Voltage (V)
ADSP-BF535PKB-350	0°C to +70°C	350 MHz	1.0 V to 1.6 V internal, 3.3 V I/O
ADSP-BF535PKB-300	0°C to +70°C	300 MHz	1.0 V to 1.5 V internal, 3.3 V I/O
ADSP-BF535PBB-300	–40°C to +85°C	300 MHz	1.0 V to 1.5 V internal, 3.3 V I/O
ADSP-BF535PBB-200	–40°C to +85°C	200 MHz	1.0 V to 1.5 V internal, 3.3 V I/O

Revision History

Location	Page
9/04—Data Sheet Changed from REV. 0 to REV. A	
Changes to Clock Signals Section	13
Changes to Recommended Operating Conditions Footnote References	21
Changes to Electrical Characteristics	21
Change to Table 11	24
Change to Figure 11	27
Change to Figure 12	28
Change to Output Drive Currents Section	36
Replaced Figures 19, 20, and 21	36
Changes to Power Dissipation Section	36
Change to Table 26	36