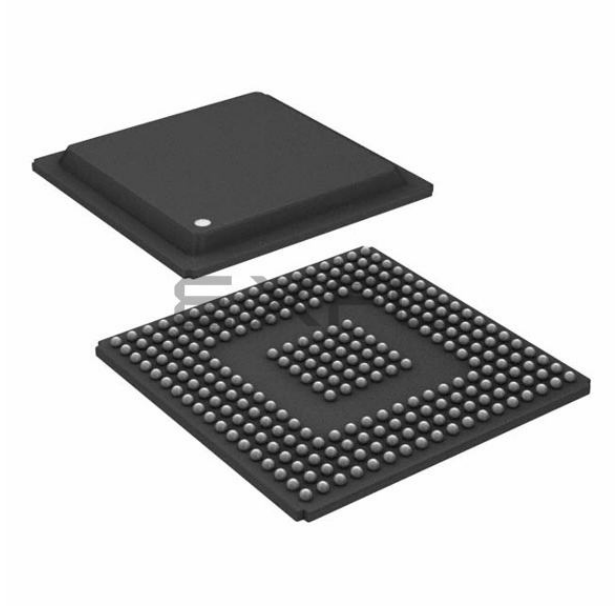

**Welcome to E-XFL.COM**

## 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 | 300MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 308kB |
| Voltage - I/O | 3.30V |
| Voltage - Core | 1.50V |
| Operating Temperature | 0°C ~ 70°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-bf535pkbz-300 |

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.

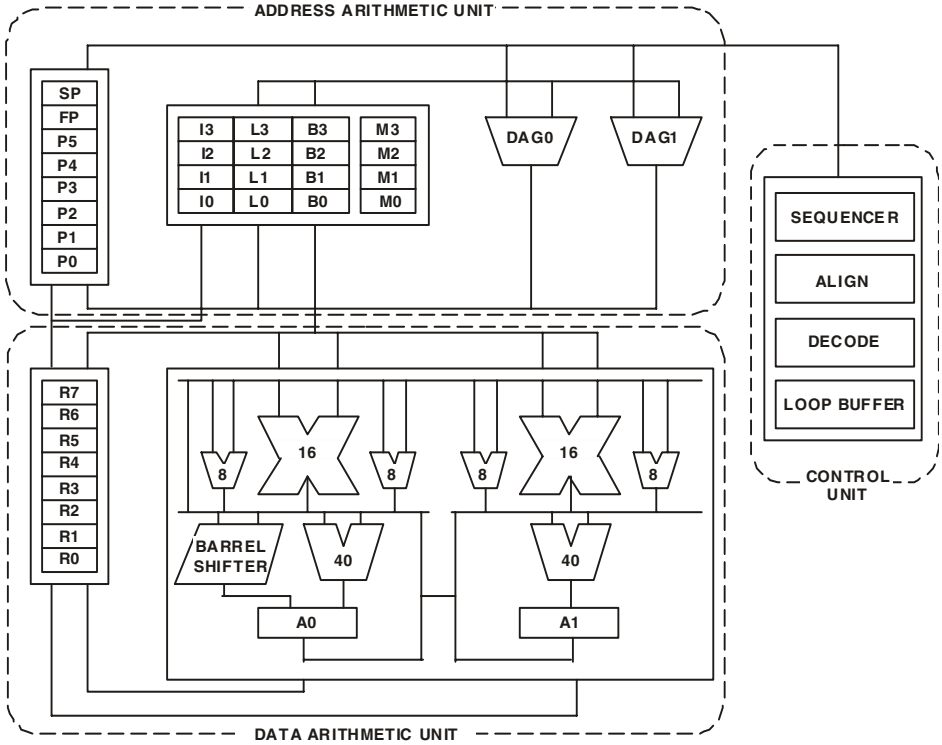


*Figure 1. Processor Core*

### *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 |

**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.

- 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 ($\overline{\text{SPISSx}}$) 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 $\overline{\text{SPISSx}}$ input pin (reconfigured programmable flag pin) from a master.

In DMA mode only, reception continues until the SPI DMA word count transitions from 1 to 0. The 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

- 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.

**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.

### *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.

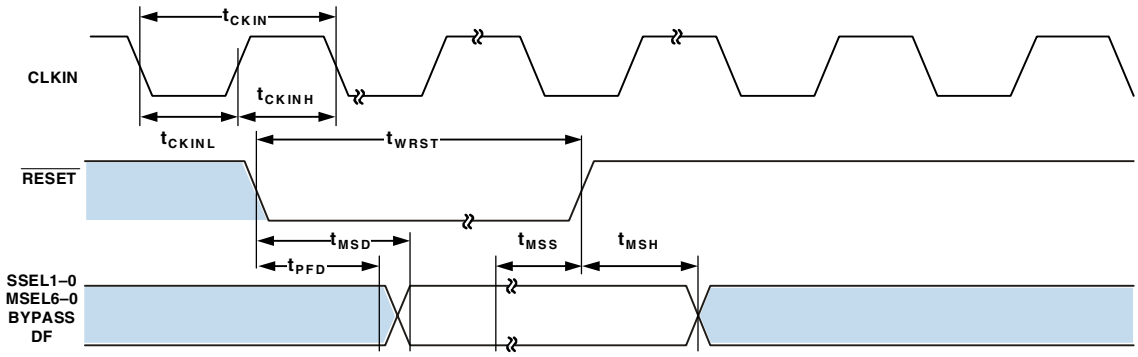


*Figure 8. Clock and Reset Timing*

***Asynchronous Memory Write Cycle Timing***

Table 14 and Figure 11 describe Asynchronous Memory Write Cycle timing.

**Table 14. Asynchronous Memory Write Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SARDY}$ | ARDY Setup Before CLKOUT | 4.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After CLKOUT | –1.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DDAT}$ | DATA31–0 Disable After CLKOUT | | 6.0 | ns |
| $t_{ENDAT}$ | DATA31–0 Enable After CLKOUT | 1.0 | | ns |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 7.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT[1] | 0.8 | | ns |

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

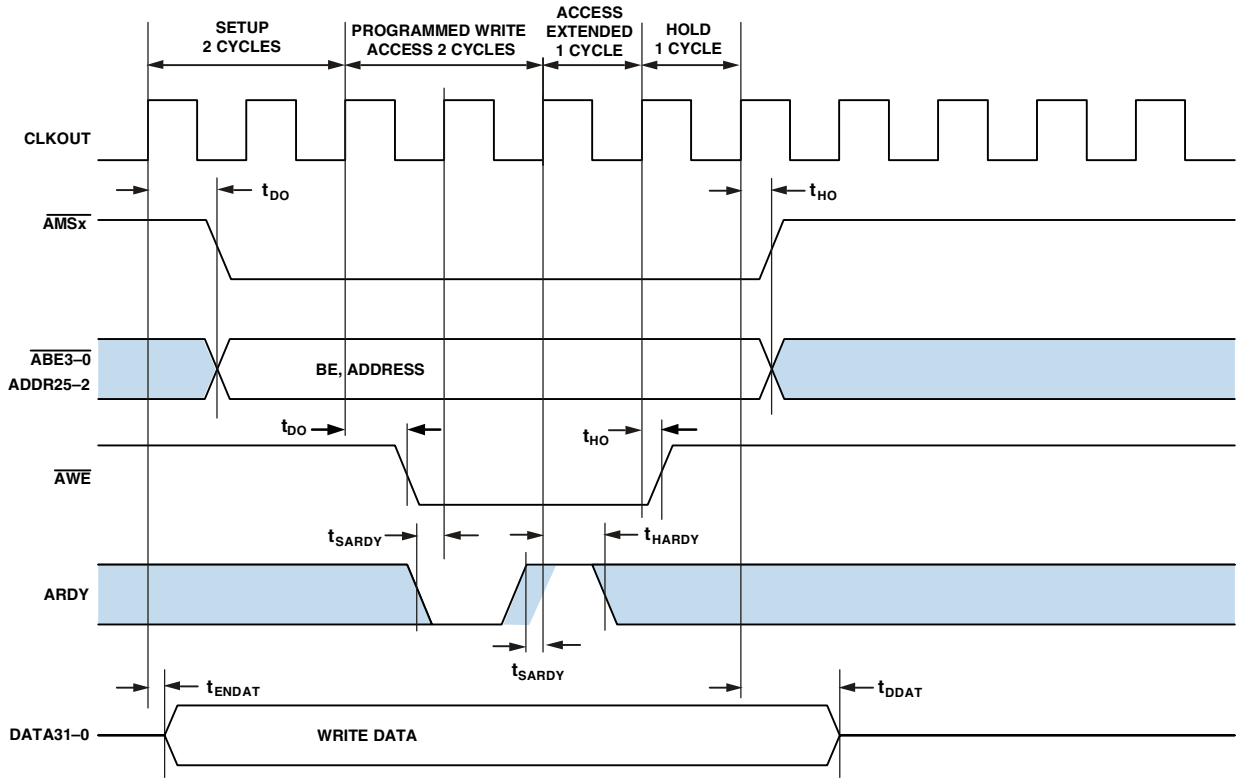


*Figure 11. Asynchronous Memory Write Cycle Timing*

***Asynchronous Memory Read Cycle Timing***

Table 15 and Figure 12 describe Asynchronous Memory Read Cycle timing.

**Table 15. Asynchronous Memory Read Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SDAT}$ | DATA31–0 Setup Before CLKOUT | 2.1 | | ns |
| $t_{HDAT}$ | DATA31–0 Hold After CLKOUT | 2.6 | | ns |
| $t_{SARDY}$ | ARDY Setup Before CLKOUT | 4.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After CLKOUT | –1.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 7.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT [1] | 0.8 | | ns |

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

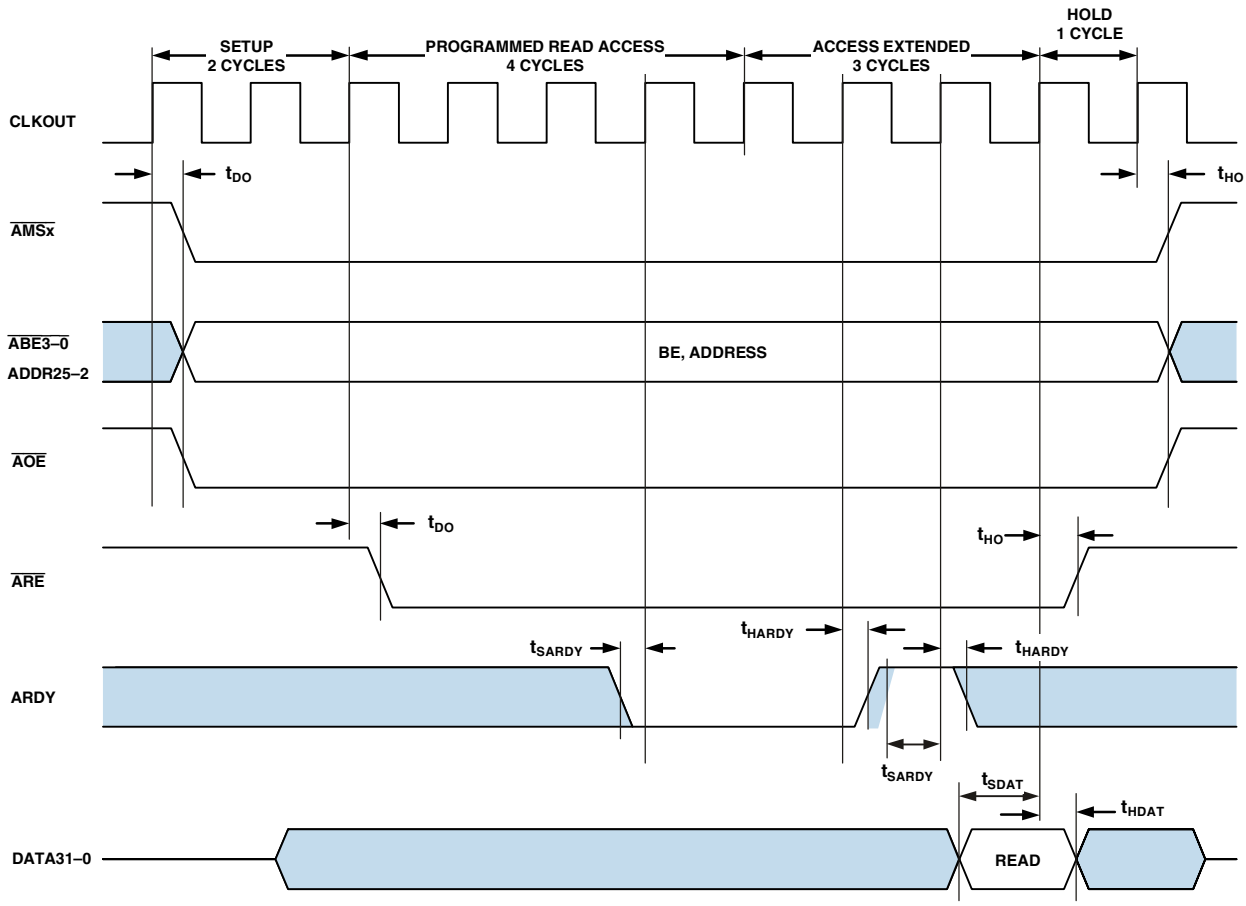


*Figure 12. Asynchronous Memory Read Cycle Timing*

***SDRAM Interface Timing***

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 $\overline{SMS3-0}$.

**Table 16. SDRAM Interface Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SSDAT}$ | DATA Setup Before SCLK0/SCLK1 | 2.1 | | ns |
| $t_{HSDAT}$ | DATA Hold After SCLK0/SCLK1 | 2.8 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SCLK}$ | SCLK0/SCLK1 Period | 7.5 | | ns |
| $t_{SCLKH}$ | SCLK0/SCLK1 Width High | 2.5 | | ns |
| $t_{SCLKL}$ | SCLK0/SCLK1 Width Low | 2.5 | | ns |
| $t_{DCAD}$ | Command, ADDR, Data Delay After SCLK0/SCLK1[1] | | 6.0 | ns |
| $t_{HCAD}$ | Command, ADDR, Data Hold After SCLK0/SCLK1[1] | 0.8 | | ns |
| $t_{DSDAT}$ | Data Disable After SCLK0/SCLK1 | | 6.0 | ns |
| $t_{ENSDAT}$ | Data Enable After SCLK0/SCLK1 | 1.0 | | ns |

[1] Command pins include: $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, SDQM3–0, $\overline{SMS}$, SA10, and SCKE.

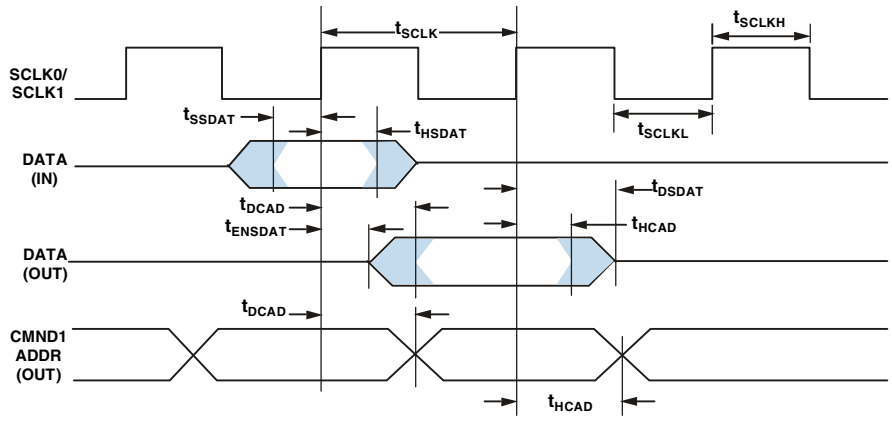


*Figure 13. SDRAM Interface Timing*

***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.

**Table 22. Serial Ports—Enable and Three-State (Multichannel Mode Only)**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DTENE}$ | Data Enable Delay from External TCLK[1] | 3.0 | | ns |
| $t_{DDTTE}$ | Data Disable Delay from External TCLK[1] | | 12.0 | ns |
| $t_{DTENI}$ | Data Enable Delay from Internal TCLK[1] | 2.0 | | ns |
| $t_{DDTTI}$ | Data Disable Delay from Internal TCLK[1] | | 12.0 | ns |

[1] Referenced to drive edge and TCLK is tied to RCLK.

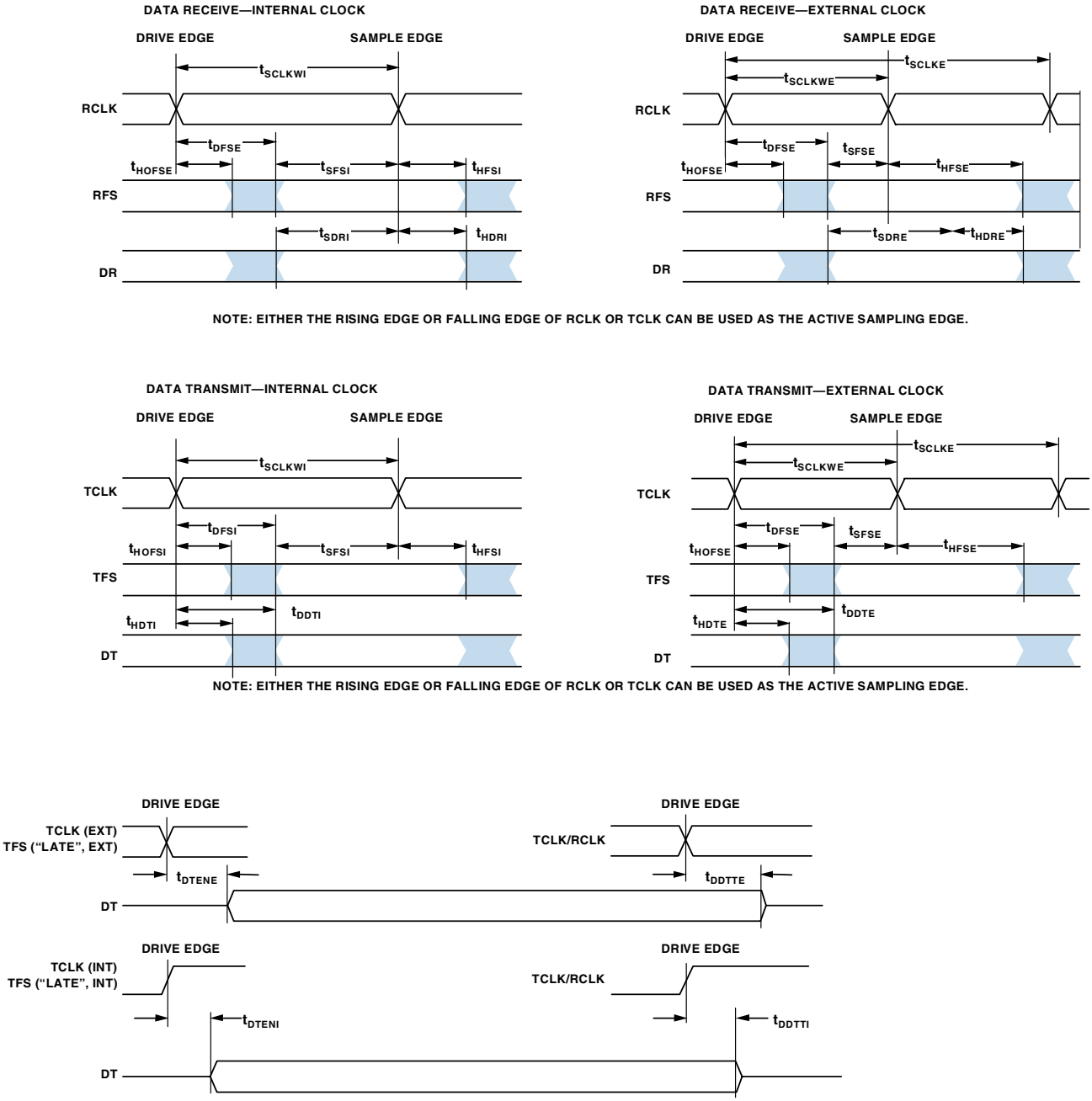


*Figure 14. Serial Ports*

***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.

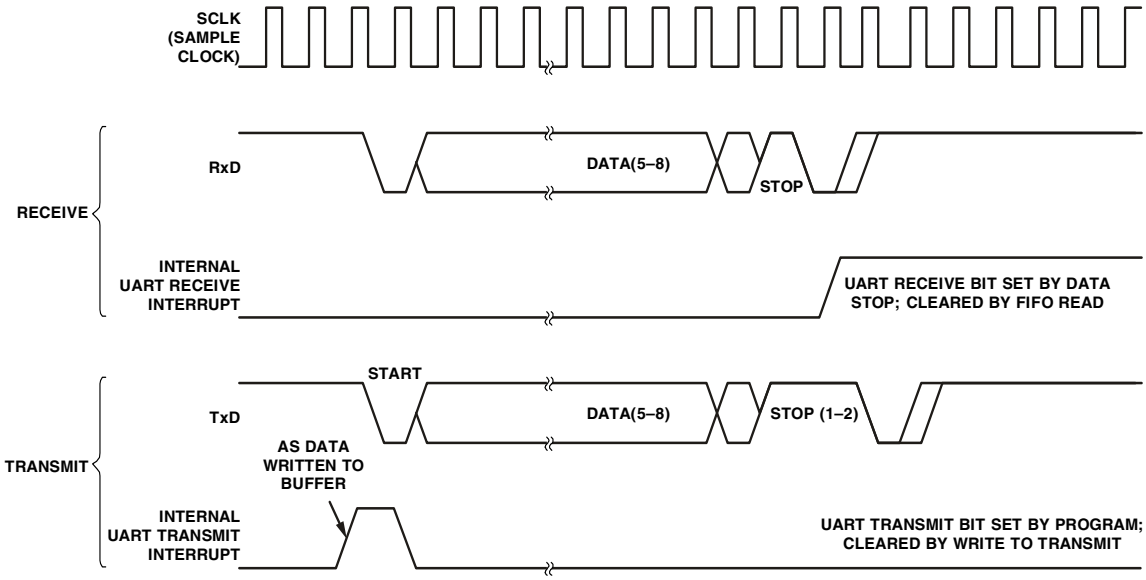


*Figure 17. UART Port—Receive and Transmit Timing*

**Table 27. PLL Power Dissipation**

| Parameter | Test Conditions | Typical | Unit |
|---|---|---|---|
| $I_{DDPLL}$ | $V_{DDPLL}$ = 1.5 V, 25°C | 4.0 | mA |

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

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

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

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

### *Test Conditions*

All timing parameters appearing in this data sheet were measured under the conditions described in this section.

### *Output Enable Time*

Output pins are considered to be enabled when they have made a transition from a high impedance state to the point when they start driving. The output enable time $t_{ENA}$ is the interval from the point when a reference signal reaches a high or low voltage level to the point when the output starts driving as shown in the Output Enable/Disable diagram (Figure 22). The time $t_{ENA\_MEASURED}$ is the interval from when the reference signal switches to when the output voltage reaches 2.0 V (output high) or 1.0 V (output low). Time $t_{TRIP}$ is the interval from when the output starts driving to when the output reaches the 1.0 V or 2.0 V trip voltage. Time $t_{ENA}$ is calculated as shown in the equation:

$$t_{ENA} = t_{ENA\_MEASURED} - t_{TRIP}$$

If multiple pins (such as the data bus) are enabled, the measurement value is that of the first pin to start driving.

### *Output Disable Time*

Output pins are considered to be disabled when they stop driving, go into a high impedance state, and start to decay from their output high or low voltage. The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load, $C_L$ and the load current, $I_L$. This decay time can be approximated by the equation:

$$t_{DECAY} = (C_L \Delta V) / I_L$$

The output disable time $t_{DIS}$ is the difference between $t_{DIS\_MEASURED}$ and $t_{DECAY}$ as shown in Figure 22. The time $t_{DIS\_MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output high or output low voltage. The time $t_{DECAY}$ is calculated with test loads $C_L$ and $I_L$, and with ΔV equal to 0.5 V.

### *Example System Hold Time Calculation*

To determine the data output hold time in a particular system, first calculate $t_{DECAY}$ using the equation given above. Choose ΔV to be the difference between the ADSP-BF535 Blackfin processor's output voltage and the input threshold for the device requiring the hold time. A typical ΔV will be 0.4 V. $C_L$ is the total bus capacitance (per data line), and $I_L$ is the total leakage or three-state current (per data line). The hold time will be $t_{DECAY}$ plus the minimum disable time (for example, $t_{DSDAT}$ for an SDRAM write cycle).

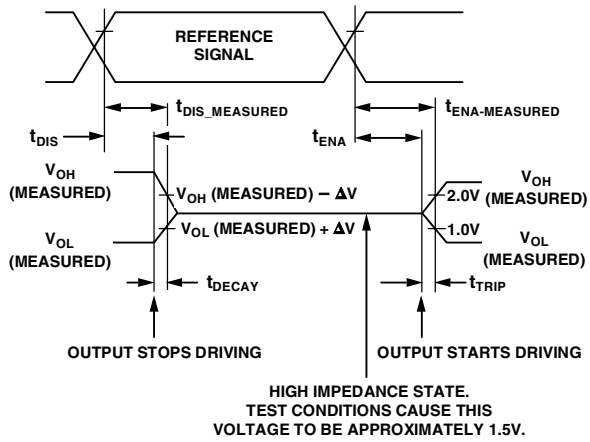


*Figure 22. Output Enable/Disable*

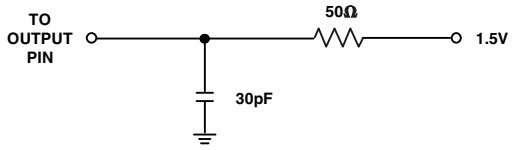


*Figure 23. Equivalent Device Loading for AC Measurements (Includes All Fixtures)*

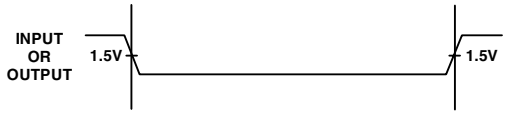


*Figure 24. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)*

**Table 29. 260-Ball PBGA Pin Assignment (Alphabetically by Signal) (continued)**

| Signal | Pin | Signal | Pin | Signal | Pin | Signal | Pin |
|---|---|---|---|---|---|---|---|
| RFS1 | V16 | $\overline{SWE}$ | J03 | USB_CLK | G07 | $V_{DDINT}$ | L12 |
| RSCLK0 | R13 | TCK | D10 | $V_{DDEXT}$ | E04 | $V_{DDINT}$ | M08 |
| RSCLK1 | U14 | TDI | C11 | $V_{DDEXT}$ | G04 | $V_{DDINT}$ | M11 |
| RX0 | A07 | TDO | D11 | $V_{DDEXT}$ | G08 | $V_{DDINT}$ | M12 |
| RX1 | B08 | TFS0 | T14 | $V_{DDEXT}$ | J01 | $V_{DDINT}$ | N04 |
| SA10 | M01 | TFS1 | R15 | $V_{DDEXT}$ | J02 | $V_{DDINT}$ | N15 |
| $\overline{SCAS}$ | L03 | TMR0 | B07 | $V_{DDEXT}$ | J04 | $V_{DDPCIEXT}$ | H15 |
| SCK0 | U17 | TMR1 | C07 | $V_{DDEXT}$ | K04 | $V_{DDPCIEXT}$ | J15 |
| SCK1 | R16 | TMR2 | D07 | $V_{DDEXT}$ | L04 | $V_{DDPCIEXT}$ | K15 |
| SCKE | L01 | TMS | A12 | $V_{DDEXT}$ | M04 | $V_{DDPCIEXT}$ | L15 |
| SCLK0 | K01 | $\overline{TRST}$ | B12 | $V_{DDEXT}$ | P04 | $V_{DDPCIEXT}$ | M15 |
| DEEPSLEEP | D12 | TSCLK0 | V15 | $V_{DDINT}$ | F04 | $V_{DDPLL}$ | G09 |
| $\overline{SMS0}$ | M02 | TSCLK1 | T15 | $V_{DDINT}$ | G11 | $V_{DDRTC}$ | U10 |
| $\overline{SMS1}$ | P01 | TX0 | A08 | $V_{DDINT}$ | G12 | $V_{SSPLL}$ | A10 |
| $\overline{SMS2}$ | N01 | TX1 | C08 | $V_{DDINT}$ | G15 | $V_{SSRTC}$ | V11 |
| $\overline{SMS3}$ | K03 | TXDMNS | G10 | $V_{DDINT}$ | H04 | XTAL1 | R10 |
| $\overline{SRAS}$ | L02 | TXDPLS | B10 | $V_{DDINT}$ | H09 | XTAL0 | T10 |
| SUSPEND | A11 | $\overline{TXEN}$ | C10 | $V_{DDINT}$ | H12 | XVER_DATA | A09 |

**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 |

**OUTLINE DIMENSIONS**

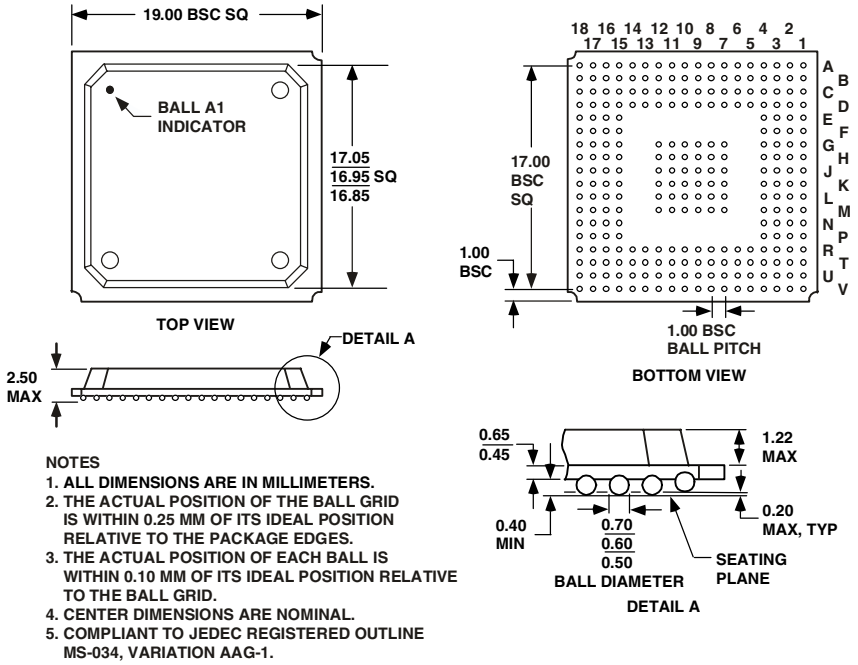


*Figure 27. 260-Ball Metric Plastic Ball Grid Array (PBGA) (B-260)*

**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 |

C04413–0–9/04(A)