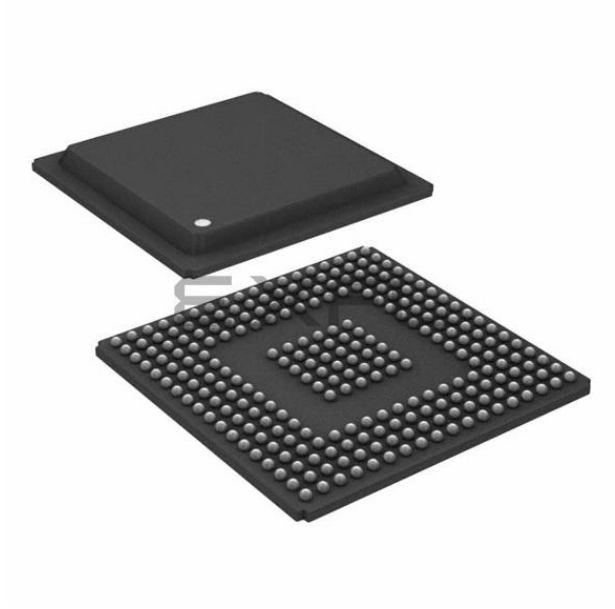

Welcome to E-XFL.COM

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	350MHz
Non-Volatile Memory	External
On-Chip RAM	308kB
Voltage - I/O	3.30V
Voltage - Core	1.60V
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-350

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

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

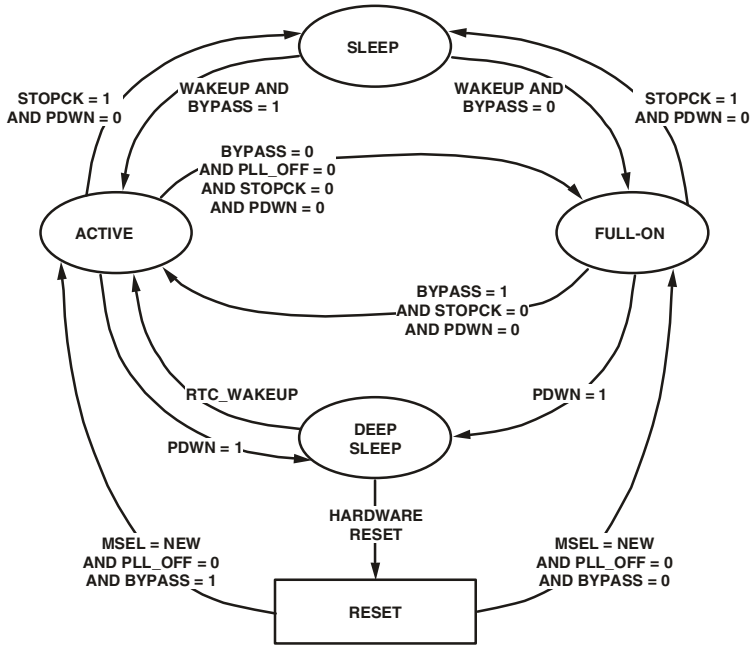

Figure 6. Mode Transitions

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.

SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

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

Specifications subject to change without notice.

[1] There is no requirement for sequencing of the voltage supplies on powerup, however, the supply regulators must be able to provide the required current $I_{DDRESET}$ at all times. See Table 26.
[2] Applies to input and bidirectional pins, except PCI and USB_CLK.
[3] Applies to USB_CLK.
[4] Applies to PCI input and bidirectional pins: PCI_AD31–0, $\overline{\text{PCI\_CBE3–0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, $\overline{\text{PCI\_DEVSEL}}$, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, $\overline{\text{PCI\_PAR}}$, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_GNT}}$, PCI_IDSEL, $\overline{\text{PCI\_LOCK}}$, PCI_CLK, $\overline{\text{PCI\_INTA}}$, $\overline{\text{PCI\_INTB}}$, $\overline{\text{PCI\_INTC}}$, $\overline{\text{PCI\_INTD}}$.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Max	Unit
V_{OH}	High Level Output Voltage[1]	@ V_{DDEXT} = min, I_{OH} = –0.5 mA	2.4		V
V_{OL}	Low Level Output Voltage[1]	@ V_{DDEXT} = max, I_{OL} = 2.0 mA		0.4	V
V_{OHPCI}	PCI High Level Output Voltage[2]	@ $V_{DDPCIEXT}$ = min, I_{OH} = –0.5 mA	0.9 × $V_{DDPCIEXT}$		V
V_{OLPCI}	PCI Low Level Output Voltage[2]	@ $V_{DDPCIEXT}$ = max, I_{OL} = 2.0 mA		0.1 × $V_{DDPCIEXT}$	V
I_{IH}	High Level Input Current[3]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max		10	µA
I_{IL}	Low Level Input Current[3]	@ V_{DDEXT} = max, V_{IN} = 0 V		10	µA
I_{OZH}	Three-State Leakage Current[4]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max		10	µA
I_{OZL}	Three-State Leakage Current[4]	@ V_{DDEXT} = max, V_{IN} = 0 V		10	µA
C_{IN}	Input Capacitance[5, 6]	f_{IN} = 1 MHz, T_A = 25°C, V_{IN} = 2.5 V		5	pF

Specifications subject to change without notice.

[1] Applies to output and bidirectional pins, except PCI.
[2] Applies to PCI output and bidirectional pins: PCI_AD31–0, $\overline{\text{PCI\_CBE3–0}}$, $\overline{\text{PCI\_FRAME}}$, $\overline{\text{PCI\_IRDY}}$, $\overline{\text{PCI\_TRDY}}$, $\overline{\text{PCI\_DEVSEL}}$, $\overline{\text{PCI\_STOP}}$, $\overline{\text{PCI\_PERR}}$, PCI_PAR, $\overline{\text{PCI\_REQ}}$, $\overline{\text{PCI\_SERR}}$, $\overline{\text{PCI\_RST}}$, $\overline{\text{PCI\_INTA}}$.
[3] Applies to input pins.
[4] Applies to three-statable pins.
[5] Applies to all signal pins.
[6] 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.

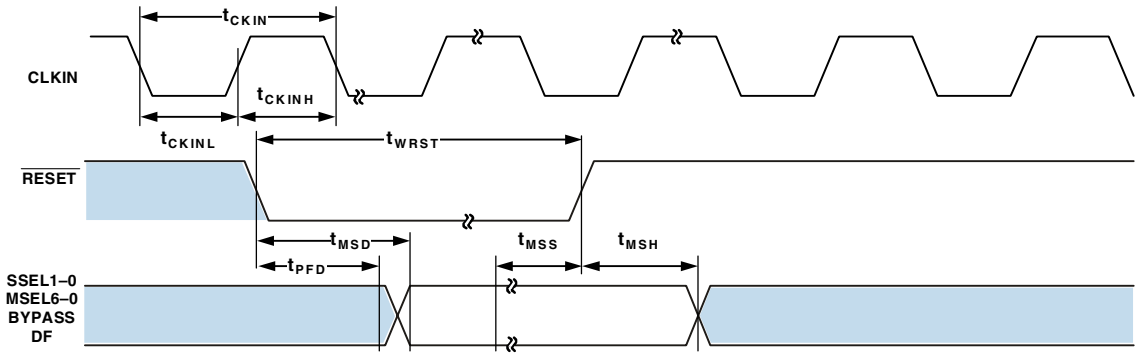

Figure 8. Clock and Reset Timing

Programmable Flags Cycle Timing

Table 12 and Figure 9 describe programmable flag operations.

Table 12. Programmable Flags Cycle Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{HFIES}	Edge Sensitive Flag Input Hold is Asynchronous	3.0		ns
t_{HFILS}	Level Sensitive Flag Input Hold	t_{SCLK}+3		ns
Switching Characteristics				
t_{DFO}	Flag Output Delay with Respect to SCLK		6.0	ns
t_{HFO}	Flag Output Hold After SCLK High		6.0	ns

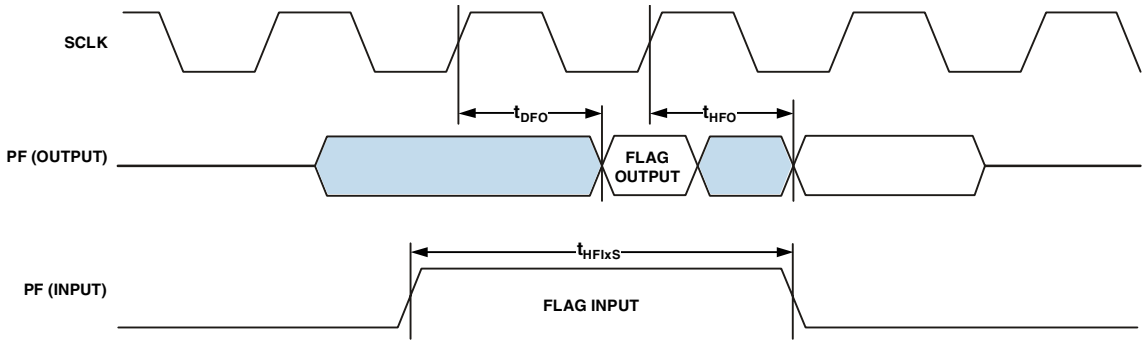

Figure 9. Programmable Flags Cycle Timing

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.

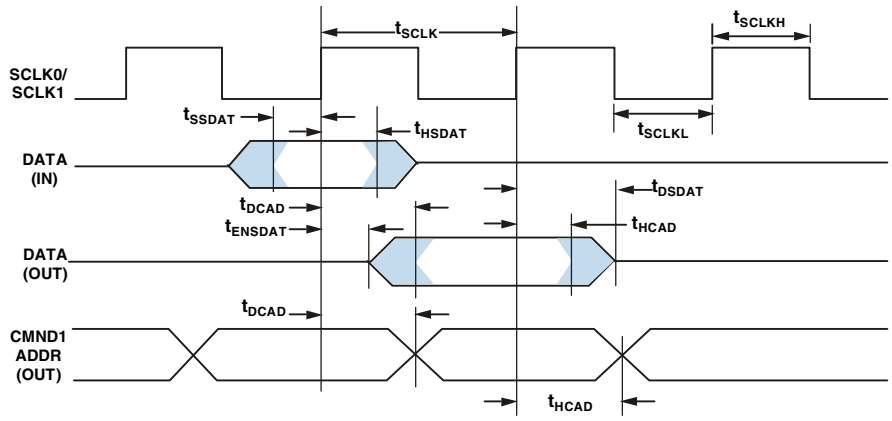

Figure 13. SDRAM Interface Timing

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.

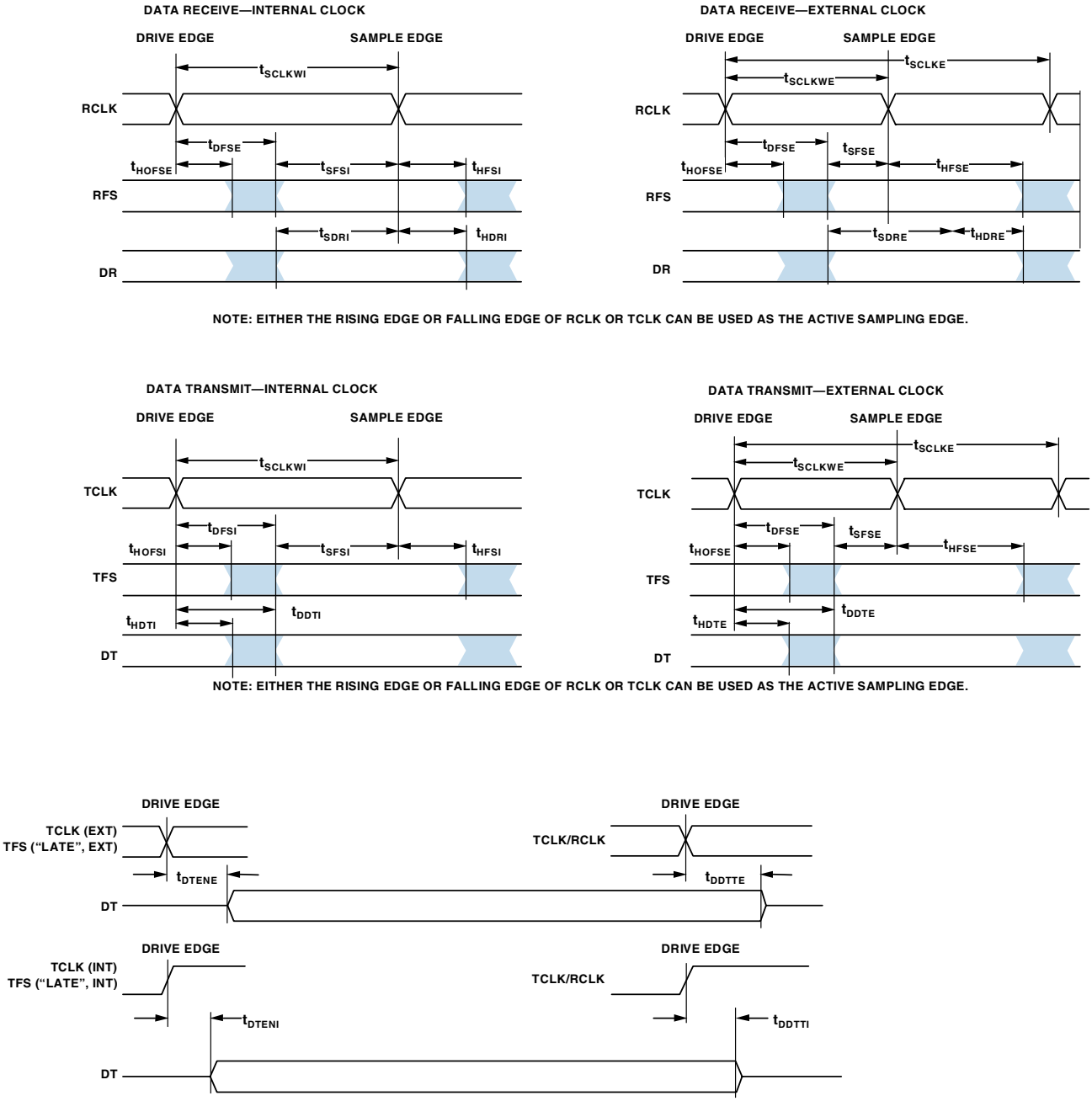

Figure 14. Serial Ports

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

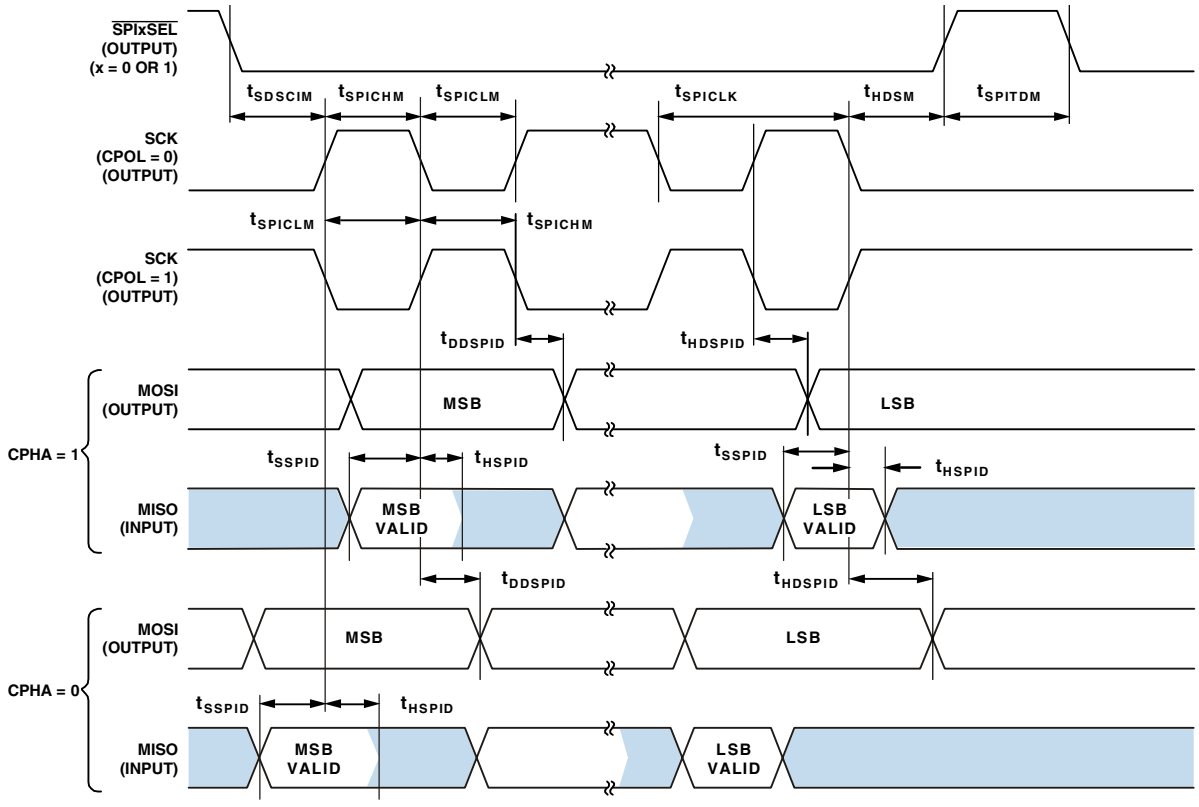

Figure 15. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port —Slave Timing

Table 24 and Figure 16 describe SPI port slave operations.

Table 24. Serial Peripheral Interface (SPI) Port—Slave Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SPICHS}	Serial Clock High Period	$2t_{SCLK}$		ns
t_{SPICLS}	Serial Clock Low Period	$2t_{SCLK}$		ns
t_{SPICLK}	Serial Clock Period	$4t_{SCLK}$		ns
t_{HDS}	Last SPICLK Edge to $\overline{SPISS}$ Not Asserted	$2t_{SCLK}$		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{SCLK}$		ns
t_{SDSCI}	$\overline{SPISS}$ Assertion to First SCK Edge	$2t_{SCLK}$		ns
t_{SSPID}	Data Input Valid to SCK Edge (Data Input Setup)	1.6		ns
t_{HSPID}	SCK Sampling Edge to Data Input Invalid	1.6		ns
Switching Characteristics				
t_{DSOE}	$\overline{SPISS}$ Assertion to Data Out Active	0.0	6.0	ns
t_{DSDHI}	$\overline{SPISS}$ Deassertion to Data High Impedance	0.0	6.5	ns
t_{DDSPID}	SCK Edge to Data Out Valid (Data Out Delay)	0.0	7.0	ns
t_{HDSPID}	SCK Edge to Data Out Invalid (Data Out Hold)	0.0	6.5	ns

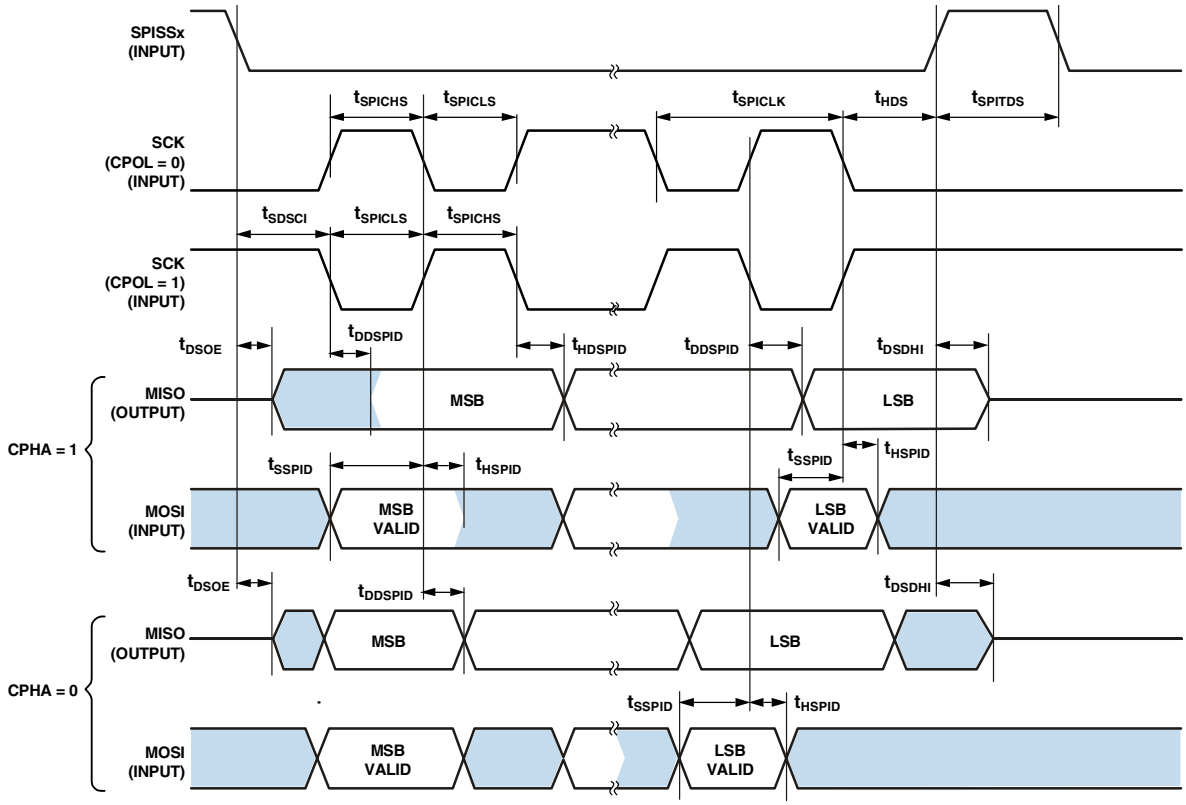

Figure 16. Serial Peripheral Interface (SPI) Port—Slave Timing

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.

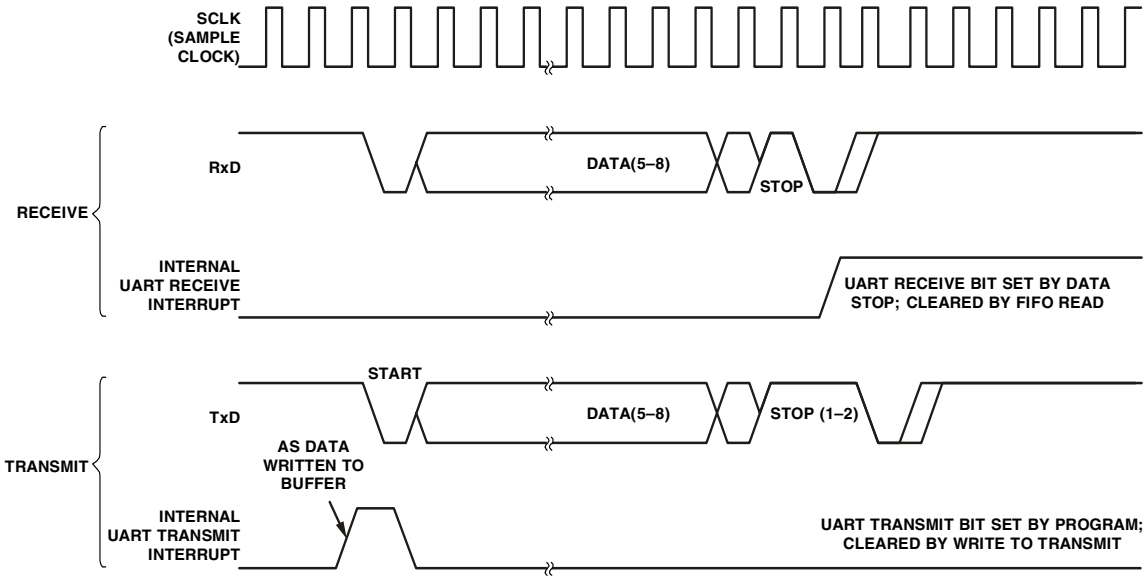

Figure 17. UART Port—Receive and Transmit Timing

260-Ball PBGA Pinout

Table 29 lists the PBGA pinout by signal name. Table 30 on Page 41 lists the pinout by pin number.

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

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
ABE0/SDQM0	E02	DATA5	R02	GND	K08	PCI_AD25	M16
ABE1/SDQM1	B01	DATA6	P03	GND	K09	PCI_AD26	N17
ABE2/SDQM2	G03	DATA7	U01	GND	K10	PCI_AD27	P17
ABE3/SDQM3	H07	DATA8	U02	GND	K11	PCI_AD28	P15
ADDR2	A06	DATA9	T02	GND	K12	PCI_AD29	N16
ADDR3	B06	DATA10	V02	GND	L07	PCI_AD30	R17
ADDR4	D06	DATA11	V03	GND	L08	PCI_AD31	P16
ADDR5	C06	DATA12	R04	GND	L09	PCI_CBE0	F16
ADDR6	A05	DATA13	U03	GND	L10	PCI_CBE1	F15
ADDR7	B05	DATA14	T03	GND	L11	PCI_CBE2	E16
ADDR8	A04	DATA15	T04	GND	M07	PCI_CBE3	D17
ADDR9	C05	DATA16	U04	GND	M09	PCI_CLK	D14
ADDR10	D05	DATA17	V04	GND	M10	PCI_DEVSEL	C16
ADDR11	B04	DATA18	V05	MISO0	T16	PCI_FRAME	C17
ADDR12	A01	DATA19	R05	MISO1	U18	PCI_GNT	C18
ADDR13	C04	DATA20	T05	MOSI0	U16	PCI_IDSEL	B18
ADDR14	D04	DATA21	U05	MOSI1	T17	PCI_INTA	C14
ADDR15	A03	DATA22	V06	N/C	A18	PCI_INTB	B15
ADDR16	B03	DATA23	R06	N/C	R03	PCI_INTC	A15
ADDR17	A02	DATA24	U06	N/C	V01	PCI_INTD	D13
ADDR18	C03	DATA25	T06	N/C	V18	PCI_IRDY	E15
ADDR19	D03	DATA26	V07	NMI	B11	PCI_LOCK	A16
ADDR20	B02	DATA27	V08	PCI_AD0	E17	PCI_PAR	C15
ADDR21	C02	DATA28	U07	PCI_AD1	E18	PCI_PERR	D15
ADDR22	E03	DATA29	R07	PCI_AD2	G16	PCI_REQ	D16
ADDR23	C01	DATA30	T07	PCI_AD3	F17	PCI_RST	D18
ADDR24	F03	DATA31	V09	PCI_AD4	F18	PCI_SERR	B16
ADDR25	D02	DMNS	D08	PCI_AD5	G18	PCI_STOP	A17
AMS0	F02	DPLS	C09	PCI_AD6	G17	PCI_TRDY	B17
AMS1	D01	DR0	V14	PCI_AD7	H18	PF0/SPISS0/MSEL0	U08
AMS2	H03	DR1	U15	PCI_AD8	J18	PF1/SPISS1/MSEL1	R08
AMS3	G02	DT0	R14	PCI_AD9	H17	PF2/SPI0SEL1/MSEL2	T08
AOE	E01	DT1	V17	PCI_AD10	K18	PF3/SPI1SEL1/MSEL3	V10
ARDY	R01	EMU	A13	PCI_AD11	H16	PF4/SPI0SEL2/MSEL4	U09
ARE	F01	GND	C13	PCI_AD12	L18	PF5/SPI1SEL2/MSEL5	R09
AWE	G01	GND	H02	PCI_AD13	J17	PF6/SPI0SEL3/MSEL6	T09
BMODE0	B14	GND	H08	PCI_AD14	M18	PF7/SPI1SEL3/DF	R11
BMODE1	A14	GND	H10	PCI_AD15	K17	PF8/SPI0SEL4/SSEL0	T11
BMODE2	B13	GND	H11	PCI_AD16	J16	PF9/SPI1SEL4/SSEL1	U11
BYPASS	C12	GND	J07	PCI_AD17	K16	PF10/SPI0SEL5	V12
CLKIN1	D09	GND	J08	PCI_AD18	N18	PF11/SPI1SEL5	T12
CLKOUT/SCLK1	H01	GND	J09	PCI_AD19	P18	PF12/SPI0SEL6	R12
DATA0	N02	GND	J10	PCI_AD20	L17	PF13/SPI1SEL6	U12
DATA1	M03	GND	J11	PCI_AD21	L16	PF14/SPI0SEL7	V13
DATA2	T01	GND	J12	PCI_AD22	R18	PF15/SPI1SEL7	T13
DATA3	P02	GND	K02	PCI_AD23	T18	RESET	B09
DATA4	N03	GND	K07	PCI_AD24	M17	RFS0	U13

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

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

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
C14	PCI_INTA	H17	PCI_AD9	N18	PCI_AD18	V03	DATA11
C15	PCI_PAR	H18	PCI_AD7	P01	SMS1	V04	DATA17
C16	PCI_DEVSEL	J01	V_{DDEXT}	P02	DATA3	V05	DATA18
C17	PCI_FRAME	J02	V_{DDEXT}	P03	DATA6	V06	DATA22
C18	PCI_GNT	J03	SWE	P04	V_{DDEXT}	V07	DATA26
D01	AMS1	J04	V_{DDEXT}	P15	PCI_AD28	V08	DATA27
D02	ADDR25	J07	GND	P16	PCI_AD31	V09	DATA31
D03	ADDR19	J08	GND	P17	PCI_AD27	V10	PF3/SPI1SEL1/MSEL3
D04	ADDR14	J09	GND	P18	PCI_AD19	V11	V_{SSRTC}
D05	ADDR10	J10	GND	R01	ARDY	V12	PF10/SPI0SEL5
D06	ADDR4	J11	GND	R02	DATA5	V13	PF14/SPI0SEL7
D07	TMR2	J12	GND	R03	N/C	V14	DR0
D08	DMNS	J15	$V_{DDPCIEXT}$	R04	DATA12	V15	TSCLK0
D09	CLKIN1	J16	PCI_AD16	R05	DATA19	V16	RFS1
D10	TCK	J17	PCI_AD13	R06	DATA23	V17	DT1
D11	TDO	J18	PCI_AD8	R07	DATA29	V18	N/C

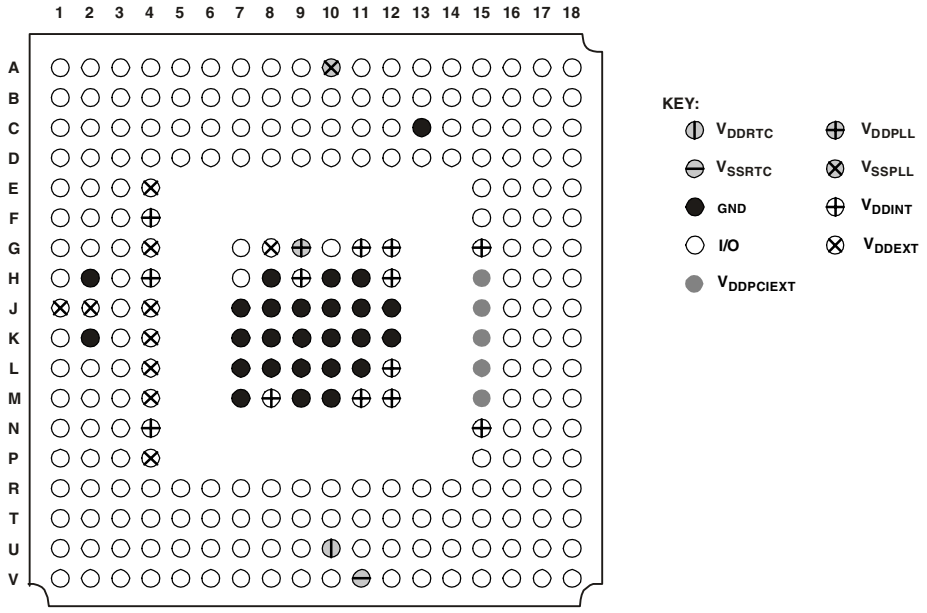

Figure 25. 260-Ball Metric PBGA Pin Configuration (Top View)

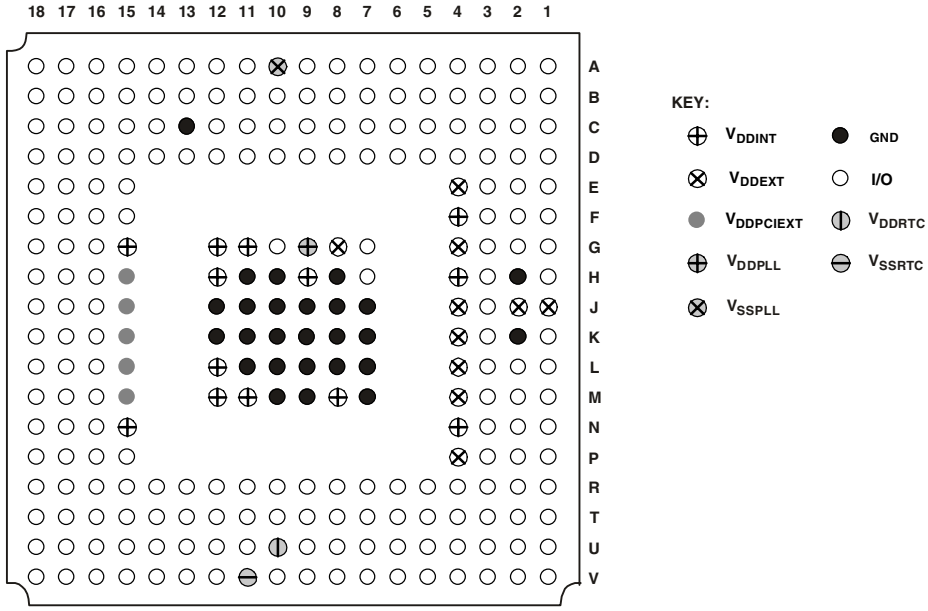

Figure 26. 260-Ball Metric PBGA Pin Configuration (Bottom View)