Blackfin Embedded Symmetric Multiprocessor

ADSP-BF561

FEATURES

Dual symmetric 600 MHz high performance Blackfin cores

328K bytes of on-chip memory
(see Memory Architecture)

Each Blackfin core includes
- **Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bit shifter**
- **RISC-like register and instruction model for ease of programming and compiler-friendly support**
- **Advanced debug, trace, and performance monitoring**

Wide range of operating voltages, (see Operating Conditions)

256-ball CSP_BGA (2 sizes) and 297-ball PBGA package options

PERIPHERALS

Dual 12-channel DMA controllers (supporting 24 peripheral DMAs)

2 memory-to-memory DMAs

2 internal memory-to-memory DMAs and 1 internal memory DMA controller

12 general-purpose 32-bit timers/counters with PWM capability

SPI-compatible port

UART with support for IrDA

Dual watchdog timers

Dual 32-bit core timers

48 programmable flags (GPIO)

On-chip phase-locked loop capable of 0.5× to 64× frequency multiplication

2 parallel input/output peripheral interface units supporting ITU-R 656 video and glueless interface to analog front end ADCs

2 dual channel, full duplex synchronous serial ports supporting eight stereo I^2S channels

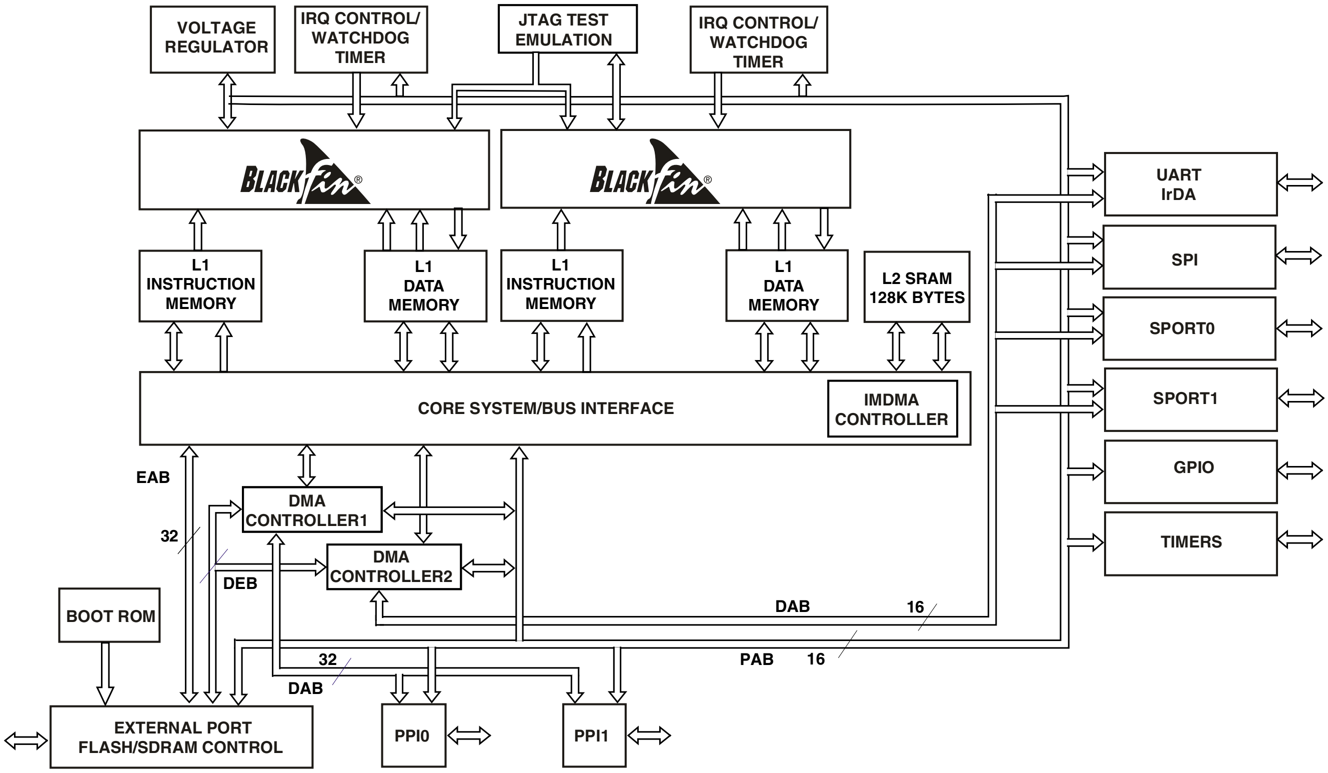


Figure 1. Functional Block Diagram

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

Rev. F **Document Feedback**

One Analog Way, Wilmington, MA 01887 U.S.A.

Technical Support **www.analog.com**

TABLE OF CONTENTS

REVISION HISTORY

7/23—Rev. E to Rev. F

Updated Development Tools 15

Added Related Signal Chains 16

Corrected footnote 2 of Table 9 in Operating Conditions 20

Removed Package Information section.

Revised Serial Ports—Enable and Three-State 33

Revised Figure 24 to match parameter names in Table 27 in Serial Peripheral Interface (SPI) Port— Slave Timing 36

Added Timer Clock Timing 39

Revised Timer Cycle Timing 39

Added figure JTAG Port Reset Timing 40

Removed obsoleted models from Ordering Guide 63

GENERAL DESCRIPTION

The ADSP-BF561 processor is a high performance member of the Blackfin® family of products targeting a variety of multimedia, industrial, and telecommunications applications. At the heart of this device are two independent Analog Devices Blackfin processors. These Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantage of a clean, orthogonal RISC-like microprocessor instruction set, and single instruction, multiple data (SIMD) multimedia capabilities in a single instruction set architecture.

The ADSP-BF561 processor has 328K bytes of on-chip memory. Each Blackfin core includes:

- 16K bytes of instruction SRAM/cache
- 16K bytes of instruction SRAM
- 32K bytes of data SRAM/cache
- 32K bytes of data SRAM
- 4K bytes of scratchpad SRAM

Additional on-chip memory peripherals include:

- 128K bytes of low latency on-chip L2 SRAM
- Four-channel internal memory DMA controller
- External memory controller with glueless support for SDRAM, mobile SDRAM, SRAM, and flash.

PORTABLE LOW POWER ARCHITECTURE

Blackfin processors provide world-class power management and performance. Blackfin processors are designed in a low power and low voltage design methodology and feature dynamic power management, the ability to vary both the voltage and frequency of operation to significantly lower overall power consumption. Varying the voltage and frequency can result in a substantial reduction in power consumption, compared with just varying the frequency of operation. This translates into longer battery life for portable appliances.

BLACKFIN PROCESSOR CORE

As shown in Figure 2, each Blackfin 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 accumulation to a 40-bit result, providing eight bits of extended precision. The ALUs perform a standard set of arithmetic and logical operations. With two ALUs capable of operating on 16-bit or 32-bit data, the flexibility of the computation units covers the signal processing requirements of a varied set of application needs.

Each of the two 32-bit input registers can be regarded as two 16-bit halves, so each ALU can accomplish very flexible single 16-bit arithmetic operations. By viewing the registers as pairs of 16-bit operands, dual 16-bit or single 32-bit operations can be accomplished in a single cycle. By further taking advantage of the second ALU, quad 16-bit operations can be accomplished simply, accelerating the per cycle throughput.

The powerful 40-bit shifter has extensive capabilities for performing shifting, rotating, normalization, extraction, and depositing of data. The data for the computational units is found in a multiported register file of sixteen 16-bit entries or eight 32-bit entries.

A powerful program sequencer controls the flow of instruction execution, including instruction alignment and decoding. The sequencer supports conditional jumps and subroutine calls, as well as zero overhead looping. A loop buffer stores instructions locally, eliminating instruction memory accesses for tight looped code.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches from memory. The DAGs share a register file containing four sets of 32-bit Index, Modify, Length, and Base registers. Eight additional 32-bit registers provide pointers for general indexing of variables and stack locations.

Blackfin processors support a modified Harvard architecture in combination with a hierarchical memory structure. Level 1 (L1) memories are those that typically operate at the full processor speed with little or no latency. Level 2 (L2) memories are other memories, on-chip or off-chip, that may take multiple processor cycles to access. At the L1 level, the instruction memory holds instructions only. The two data memories hold data, and a dedicated scratchpad data memory stores stack and local variable information. At the L2 level, there is a single unified memory space, holding both instructions and data.

In addition, half of L1 instruction memory and half of L1 data memory may be configured as either Static RAMs (SRAMs) or caches. The Memory Management Unit (MMU) provides memory protection for individual tasks that may be operating on the core and may protect system registers from unintended access.

The architecture provides three modes of operation: user mode, supervisor mode, and emulation mode. User mode has restricted access to certain system resources, thus providing a protected software environment, while supervisor mode has unrestricted access to the system and core resources.

The Blackfin instruction set has been optimized so that 16-bit op-codes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit op-codes, representing fully featured multifunction instructions. Blackfin processors support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

The Blackfin assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the VisualDSP C/C++ compiler, resulting in fast and efficient software implementations.

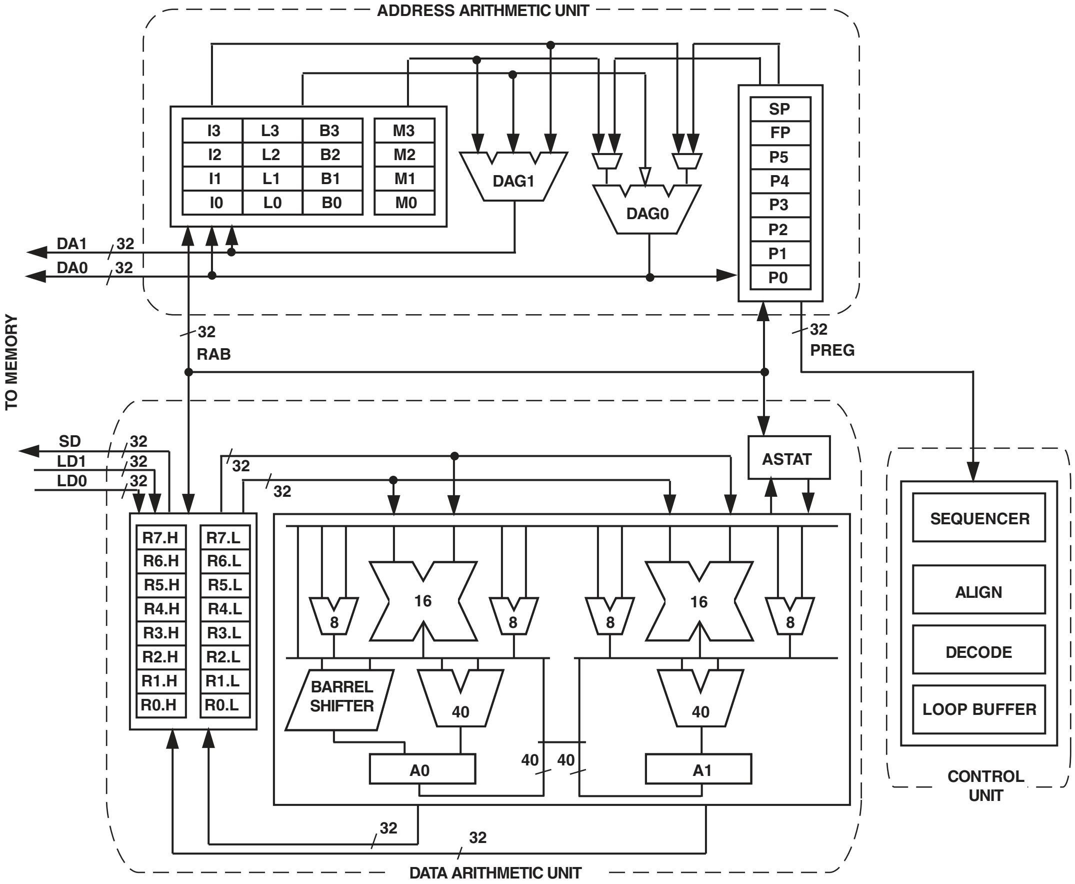


Figure 2. Blackfin Processor Core

MEMORY ARCHITECTURE

The ADSP-BF561 views memory as a single unified 4G byte address space, using 32-bit addresses. All resources including internal memory, external memory, and I/O control registers occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low latency memory as cache or SRAM very close to the processor, and larger, lower cost and performance memory systems farther away from the processor. The ADSP-BF561 memory map is shown in Figure 3.

The L1 memory system in each core is the highest performance memory available to each Blackfin core. The L2 memory provides additional capacity with lower performance. Lastly, the off-chip memory system, accessed through the External Bus Interface Unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing more than 768M bytes of physical memory. The memory DMA controllers provide high bandwidth data movement capability. They can perform block transfers of code or data between the internal L1/L2 memories and the external memory spaces.

Internal (On-Chip) Memory

The ADSP-BF561 has four blocks of on-chip memory providing high bandwidth access to the core.

The first is the L1 instruction memory of each Blackfin core consisting of 16K bytes of four-way set-associative cache memory and 16K bytes of SRAM. The cache memory may also be configured as an SRAM. This memory is accessed at full processor speed. When configured as SRAM, each of the two 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The second on-chip memory block is the L1 data memory of each Blackfin core which consists of four banks of 16K bytes each. Two of the L1 data memory banks can be configured as one way of a two-way set-associative cache or as an SRAM. The other two banks are configured as SRAM. All banks are accessed at full processor speed. When configured as SRAM, each of the four 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The third memory block associated with each core is a 4K byte scratchpad SRAM which runs at the same speed as the L1 memories, but is only accessible as data SRAM (it cannot be configured as cache memory and is not accessible via DMA).

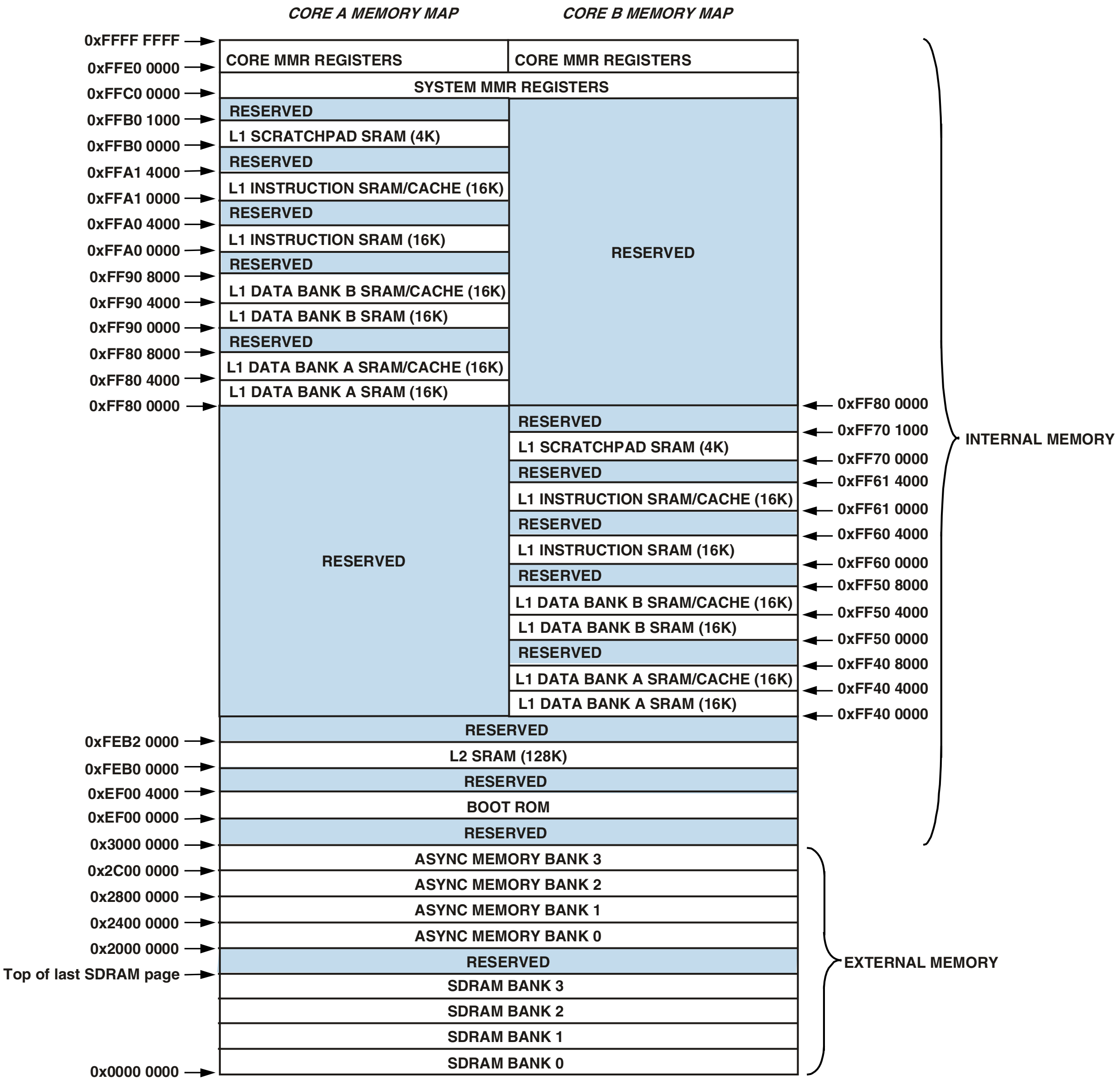


Figure 3. Memory Map

The fourth on-chip memory system is the L2 SRAM memory array which provides 128K bytes of high speed SRAM operating at one half the frequency of the core, and slightly longer latency than the L1 memory banks. The L2 memory is a unified instruction and data memory and can hold any mixture of code and data required by the system design. The Blackfin cores share a dedicated low latency 64-bit wide data path port into the L2 SRAM memory.

Each Blackfin core processor has its own set of core Memory Mapped Registers (MMRs) but share the same system MMR registers and 128K bytes L2 SRAM memory.

External (Off-Chip) Memory

The ADSP-BF561 external memory is accessed via the External Bus Interface Unit (EBIU). This interface provides a glueless connection to up to four banks of synchronous DRAM (SDRAM) as well as up to four banks of asynchronous memory devices, including flash, EPROM, ROM, SRAM, and memory mapped I/O devices.

The PC133-compliant SDRAM controller can be programmed to interface to up to four banks of SDRAM, with each bank containing between 16M bytes and 128M bytes providing access to up to 512M bytes of SDRAM. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement. This allows flexible configuration and upgradability of system memory while allowing the core to view all SDRAM as a single, contiguous, physical address space.

The asynchronous memory controller can also be programmed to control up to four banks of devices with very flexible timing parameters for a wide variety of devices. Each bank occupies a 64M byte segment regardless of the size of the devices used so that these banks will only be contiguous if fully populated with 64M bytes of memory.

I/O Memory Space

Blackfin processors do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory mapped registers (MMRs) at addresses near the top of the 4G byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The core MMRs are accessible only by the core and only in supervisor mode and appear as reserved space by on-chip peripherals. The system MMRs are accessible by the core in supervisor mode and can be mapped as either visible or reserved to other devices, depending on the system protection model desired.

Booting

The ADSP-BF561 contains a small boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF561 is configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM.

Event Handling

The event controller on the ADSP-BF561 handles all asynchronous and synchronous events to the processor. The ADSP-BF561 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.
- Nonmaskable 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, i.e., the exception will be taken before the instruction is allowed to complete. Conditions such as data alignment violations or undefined instructions cause exceptions.
- Interrupts – Events that occur asynchronously to program flow. They are caused by timers, peripherals, input pins, and an explicit software instruction.

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

The ADSP-BF561 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-BF561. 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 Control	$\overline{\text{EMU}}$
1	Reset	RST
2	Nonmaskable Interrupt	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-BF561 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 (SIC_IAR7–0). Table 2 describes the inputs into the SIC and the default mappings into the CEC.

Table 2. System Interrupt Controller (SIC)

Peripheral Interrupt Event	Default Mapping
PLL Wakeup	IVG7
DMA1 Error (Generic)	IVG7
DMA2 Error (Generic)	IVG7
IMDMA Error	IVG7
PPI0 Error	IVG7
PPI1 Error	IVG7
SPORT0 Error	IVG7
SPORT1 Error	IVG7
SPI Error	IVG7
UART Error	IVG7
Reserved	IVG7
DMA1 Channel 0 Interrupt (PPI0)	IVG8
DMA1 Channel 1 Interrupt (PPI1)	IVG8
DMA1 Channel 2 Interrupt	IVG8
DMA1 Channel 3 Interrupt	IVG8
DMA1 Channel 4 Interrupt	IVG8
DMA1 Channel 5 Interrupt	IVG8
DMA1 Channel 6 Interrupt	IVG8
DMA1 Channel 7 Interrupt	IVG8
DMA1 Channel 8 Interrupt	IVG8
DMA1 Channel 9 Interrupt	IVG8
DMA1 Channel 10 Interrupt	IVG8
DMA1 Channel 11 Interrupt	IVG8
DMA2 Channel 0 Interrupt (SPORT0 Rx)	IVG9
DMA2 Channel 1 Interrupt (SPORT0 Tx)	IVG9
DMA2 Channel 2 Interrupt (SPORT1 Rx)	IVG9
DMA2 Channel 3 Interrupt (SPORT1 Tx)	IVG9
DMA2 Channel 4 Interrupt (SPI)	IVG9
DMA2 Channel 5 Interrupt (UART Rx)	IVG9
DMA2 Channel 6 Interrupt (UART Tx)	IVG9
DMA2 Channel 7 Interrupt	IVG9
DMA2 Channel 8 Interrupt	IVG9
DMA2 Channel 9 Interrupt	IVG9
DMA2 Channel 10 Interrupt	IVG9
DMA2 Channel 11 Interrupt	IVG9
Timer0 Interrupt	IVG10
Timer1 Interrupt	IVG10
Timer2 Interrupt	IVG10
Timer3 Interrupt	IVG10
Timer4 Interrupt	IVG10
Timer5 Interrupt	IVG10
Timer6 Interrupt	IVG10
Timer7 Interrupt	IVG10
Timer8 Interrupt	IVG10
Timer9 Interrupt	IVG10
Timer10 Interrupt	IVG10
Timer11 Interrupt	IVG10
Programmable Flags 15–0 Interrupt A	IVG11
Programmable Flags 15–0 Interrupt B	IVG11
Programmable Flags 31–16 Interrupt A	IVG11
Programmable Flags 31–16 Interrupt B	IVG11
Programmable Flags 47–32 Interrupt A	IVG11
Programmable Flags 47–32 Interrupt B	IVG11
DMA1 Channel 12/13 Interrupt (Memory DMA/Stream 0)	IVG8
DMA1 Channel 14/15 Interrupt (Memory DMA/Stream 1)	IVG8
DMA2 Channel 12/13 Interrupt (Memory DMA/Stream 0)	IVG9
DMA2 Channel 14/15 Interrupt (Memory DMA/Stream 1)	IVG9
IMDMA Stream 0 Interrupt	IVG12
IMDMA Stream 1 Interrupt	IVG12
Watchdog Timer Interrupt	IVG13
Reserved	IVG7
Reserved	IVG7
Supplemental Interrupt 0	IVG7
Supplemental Interrupt 1	IVG7

Event Control

The ADSP-BF561 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, while 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 also be written to clear (cancel) latched events. This register may be read while in supervisor mode and may only be written while in supervisor mode when the corresponding IMASK bit is cleared.
- 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 six 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 Registers (SIC_IMASKx) – These registers control the masking and unmasking of each peripheral interrupt event. When a bit is set in these registers, that peripheral event is unmasked and will be processed by the system when asserted. A cleared bit in these registers masks the peripheral event, thereby preventing the processor from servicing the event.
- SIC Interrupt Status Registers (SIC_ISRx) – As multiple peripherals can be mapped to a single event, these registers allow 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 Registers (SIC_IWRx) – By enabling the corresponding bit in these registers, each peripheral can be configured to wake up the processor, should the processor be in a powered-down mode when the event is generated.

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-BF561 has two independent DMA controllers that support automated data transfers with minimal overhead for the DSP cores. DMA transfers can occur between the ADSP-BF561 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 and the asynchronous memory controller. DMA-capable peripherals include the SPORTs, SPI port, UART, and PPIs. Each individual DMA-capable peripheral has at least one dedicated DMA channel.

The ADSP-BF561 DMA controllers support both 1-dimensional (1-D) and 2-dimensional (2-D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2-D DMA capability supports arbitrary row and column sizes up to 64K elements by 64K elements, and arbitrary row and column step sizes up to ± 32K elements. Furthermore, the column step size can be less than the row step size, allowing implementation of interleaved data streams. This feature is especially useful in video applications where data can be de-interleaved on the fly.

Examples of DMA types supported by the ADSP-BF561 DMA controllers include:

- A single linear buffer that stops upon completion.
- A circular autorefreshing buffer that interrupts on each full or fractionally full buffer.
- 1-D or 2-D DMA using a linked list of descriptors.
- 2-D DMA using an array of descriptors, specifying only the base DMA address within a common page.

In addition to the dedicated peripheral DMA channels, each DMA Controller has four memory DMA channels provided for transfers between the various memories of the ADSP-BF561 system. These enable transfers of blocks of data between any of the memories—including external SDRAM, ROM, SRAM, and flash memory—with minimal processor intervention. Memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

Further, the ADSP-BF561 has a four channel Internal Memory DMA (IMDMA) Controller. The IMDMA Controller allows data transfers between any of the internal L1 and L2 memories.

WATCHDOG TIMER

Each ADSP-BF561 core includes a 32-bit timer, which can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, via generation of a hardware reset, nonmaskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the timer control register, which is set only upon a watchdog generated reset.

The timer is clocked by the system clock (SCLK) at a maximum frequency of f_{SCLK}.

TIMERS

There are 14 programmable timer units in the ADSP-BF561.

Each of the 12 general-purpose timer units can be independently programmed as a Pulse Width Modulator (PWM), internally or externally clocked timer, or pulse width counter. The general-purpose timer units can be used in conjunction with the UART to measure the width of the pulses in the data stream to provide an autobaud detect function for a serial channel. The general-purpose timers can generate interrupts to the processor core providing periodic events for synchronization, either to the processor clock or to a count of external signals.

In addition to the 12 general-purpose programmable timers, another timer is also provided for each core. These extra timers are clocked by the internal processor clock (CCLK) and are typically used as a system tick clock for generation of operating system periodic interrupts.

SERIAL PORTS (SPORTs)

The ADSP-BF561 incorporates two dual-channel synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

- I^2S capable operation.
- Bidirectional operation – Each SPORT has two sets of independent transmit and receive pins, enabling eight channels of I^2S stereo audio.
- Buffered (8-deep) transmit and receive ports – Each port has a data register for transferring data words to and from other DSP components and shift registers for shifting data in and out of the data registers.
- Clocking – Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from (f_{SCLK}/131,070) Hz to (f_{SCLK}/2) Hz.
- Word length – Each SPORT supports serial data words from 3 bits to 32 bits in length, transferred most significant bit first or least significant bit first.
- Framing – Each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two 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 DSP can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts – Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability – Each SPORT supports 128 channels out of a 1,024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

An additional 250 mV of SPORT input hysteresis can be enabled by setting Bit 15 of the PLL_CTL register. When this bit is set, all SPORT input pins have the increased hysteresis.

SERIAL PERIPHERAL INTERFACE (SPI) PORT

The ADSP-BF561 processor has an SPI-compatible port that enables 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, MOSI, and master input-slave output, MISO) and a clock pin (serial clock, SCK). An SPI chip select input pin ($\overline{\text{SPISS}}$) lets other SPI devices select the processor, and seven SPI chip select output pins ($\overline{\text{SPISEL7–1}}$) let the processor select other SPI devices. The SPI select pins are reconfigured programmable flag pins. Using these pins, the SPI port provides a full-duplex, synchronous serial interface which supports both master/slave modes and multimaster environments.

The baud rate and clock phase/polarities for the SPI port are programmable, and it has an integrated DMA controller, configurable to support transmit or receive data streams. The SPI DMA controller can only service unidirectional accesses at any given time.

The SPI port clock rate is calculated as:

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

Where the 16-bit SPI_BAUD register contains a value of 2 to 65,535.

During transfers, the SPI port simultaneously transmits and receives by serially shifting data in and out on its two serial data lines. The serial clock line synchronizes the shifting and sampling of data on the two serial data lines.

UART PORT

The ADSP-BF561 processor provides a full-duplex universal asynchronous receiver/transmitter (UART) port, which is fully compatible with PC-standard UARTs. The UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. The UART port includes support for 5 data bits to 8 data bits, 1 stop bit or 2 stop bits, and none, even, or odd parity. The UART port supports two modes of operation:

- PIO (programmed I/O) – The processor sends or receives data by writing or reading I/O-mapped UART registers. 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. The UART has two dedicated DMA channels, one for transmit and one for receive. These DMA channels have lower default priority than most DMA channels because of their relatively low service rates.

The baud rate, serial data format, error code generation and status, and interrupts for the UART port are programmable.

The UART programmable features include:

- Supporting bit rates ranging from (f_{SCLK}/1,048,576) bits per second to (f_{SCLK}/16) bits per second.
- Supporting data formats from seven bits to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as:

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

Where the 16-bit UART_Divisor comes from the UART_DLH register (most significant 8 bits) and UART_DLL register (least significant 8 bits).

In conjunction with the general-purpose timer functions, autobaud detection is supported.

The capabilities of the UART are further extended with support for the Infrared Data Association (IrDA®) serial infrared physical layer link specification (SIR) protocol.

PROGRAMMABLE FLAGS (PFx)

The ADSP-BF561 has 48 bidirectional, general-purpose I/O, programmable flag (PF47–0) pins. Some programmable flag pins are used by peripherals (see Pin Descriptions). When not used as a peripheral pin, each programmable flag can be individually controlled by manipulation of the flag control, status, and interrupt registers as follows:

- 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-BF561 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 – These registers allow each individual PFx pin to function as an interrupt to the processor. Similar to the flag control registers that are used to set and clear individual flag values, one flag interrupt mask register sets bits to enable an interrupt function, and the other flag interrupt mask register clears bits to disable an 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 – These 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.

PARALLEL PERIPHERAL INTERFACE

The ADSP-BF561 processor provides two parallel peripheral interfaces (PPI0, PPI1) that can connect directly to parallel A/D and D/A converters, video encoders and decoders, and other general-purpose peripherals. The PPI consists of a dedicated input clock pin, up to 3 frame synchronization pins, and up to 16 data pins. The input clock supports parallel data rates at up to f_{SCLK}/2 MHz, and the synchronization signals can be configured as either inputs or outputs.

The PPI supports a variety of general-purpose and ITU-R 656 modes of operation. In general-purpose mode, the PPI provides half-duplex, bi-directional data transfer with up to 16 bits of data. Up to 3 frame synchronization signals are also provided. In ITU-R 656 mode, the PPI provides half-duplex, bi-directional transfer of 8- or 10-bit video data. Additionally, on-chip decode of embedded start-of-line (SOL) and start-of-field (SOF) preamble packets is supported.

General-Purpose Mode Descriptions

The general-purpose modes of the PPI are intended to suit a wide variety of data capture and transmission applications. Three distinct submodes are supported:

- Input mode – frame syncs and data are inputs into the PPI.
- Frame capture mode – frame syncs are outputs from the PPI, but data are inputs.
- Output mode – frame syncs and data are outputs from the PPI.

Input Mode

Input mode is intended for ADC applications, as well as video communication with hardware signaling. In its simplest form, PPI_FS1 is an external frame sync input that controls when to read data. The PPI_DELAY MMR allows for a delay (in PPI_CLK cycles) between reception of this frame sync and the initiation of data reads. The number of input data samples is user programmable and defined by the contents of the PPI_COUNT register. The PPI supports 8-bit, and 10-bit through 16-bit data, and are programmable in the PPI_CONTROL register.

Frame Capture Mode

Frame capture mode allows the video source(s) to act as a slave (e.g., for frame capture). The ADSP-BF561 processors control when to read from the video source(s). PPI_FS1 is an HSYNC output and PPI_FS2 is a VSYNC output.

Output Mode

Output mode is used for transmitting video or other data with up to three output frame syncs. Typically, a single frame sync is appropriate for data converter applications, whereas two or three frame syncs could be used for sending video with hardware signaling.

ITU-R 656 Mode Descriptions

The ITU-R 656 modes of the PPI are intended to suit a wide variety of video capture, processing, and transmission applications. Three distinct submodes are supported:

- Active video only mode
- Vertical blanking only mode
- Entire field mode

Active Video Only Mode

Active video only mode is used when only the active video portion of a field is of interest and not any of the blanking intervals. The PPI does not read in any data between the end of active video (EAV) and start of active video (SAV) preamble symbols, or any data present during the vertical blanking intervals. In this mode, the control byte sequences are not stored to memory; they are filtered by the PPI. After synchronizing to the start of Field 1, the PPI ignores incoming samples until it sees an SAV code. The user specifies the number of active video lines per frame (in the PPI_COUNT register).

Vertical Blanking Interval Mode

In this mode, the PPI only transfers vertical blanking interval (VBI) data.

Entire Field Mode

In this mode, the entire incoming bit stream is read in through the PPI. This includes active video, control preamble sequences, and ancillary data that may be embedded in horizontal and vertical blanking intervals. Data transfer starts immediately after synchronization to Field 1.

DYNAMIC POWER MANAGEMENT

The ADSP-BF561 provides four power management modes and one power management state, each with a different performance/power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the ADSP-BF561 peripherals also reduces power consumption. See Table 3 for a summary of the power settings for each mode.

Table 3. Power Settings

Mode/State	PLL	PLL Bypassed	Core Clock (CCLK)	System Clock (SCLK)	Core Power
Full-On	Enabled	No	Enabled	Enabled	On
Active	Enabled/ Disabled	Yes	Enabled	Enabled	On
Sleep	Enabled	–	Disabled	Enabled	On
Deep Sleep	Disabled	–	Disabled	Disabled	On
Hibernate	Disabled	–	Disabled	Disabled	Off

Full-On Operating Mode—Maximum Performance

In the full-on mode, the PLL is enabled and is not bypassed, providing capability for maximum operational frequency. This is the default execution state in which maximum performance can be achieved. The processor cores and all enabled peripherals run at full speed.

Active Operating Mode—Moderate Power Savings

In the active mode, the PLL is enabled but bypassed. Because the PLL is bypassed, the processor's core clock (CCLK) and system clock (SCLK) run at the input clock (CLKIN) frequency. In this mode, the CLKIN to CCLK multiplier ratio can be changed, although the changes are not realized until the full-on mode is entered. DMA access is available to appropriately configured L1 and L2 memories.

In the active mode, it is possible to disable the PLL through the PLL control register (PLL_CTL). If disabled, the PLL must be re-enabled before transitioning to the full-on or sleep modes.

Sleep Operating Mode—High Dynamic 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. Typically an external event will wake up the processor. When in the sleep mode, assertion of wakeup will cause the processor to sense the value of the BYPASS bit in the PLL control register (PLL_CTL).

When in the sleep mode, system DMA access is only available to external memory, not to L1 or on-chip L2 memory.

Deep Sleep Operating Mode—Maximum Dynamic Power Savings

The deep sleep mode maximizes power savings by disabling the clocks to the processor cores (CCLK) and to all synchronous peripherals (SCLK). Asynchronous peripherals will not be able to access internal resources or external memory. This powered-down mode can only be exited by assertion of the reset pin ($\overline{\text{RESET}}$). If BYPASS is disabled, the processor will transition to the full-on mode. If BYPASS is enabled, the processor will transition to the active mode.

Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all the synchronous peripherals (SCLK). The internal voltage regulator for the processor can be shut off by writing b#00 to the FREQ bits of the VR_CTL register. This disables both CCLK and SCLK. Furthermore, it sets the internal power supply voltage (V_{DDINT}) to 0 V to provide the lowest static power dissipation. Any critical information stored internally (memory contents, register contents, etc.) must be written to a nonvolatile storage device prior to removing power if the processor state is to be preserved. Since V_{DDEXT} is still supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that may be connected to the processor to have power still applied without drawing unwanted current. The internal supply regulator can be woken up by asserting the $\overline{\text{RESET}}$ pin.

Power Savings

As shown in Table 4, the ADSP-BF561 supports two 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-BF561 into its own power domain, separate from the I/O, the processor can take advantage of Dynamic Power Management, without affecting the I/O devices. There are no sequencing requirements for the various power domains.

Table 4. ADSP-BF561 Power Domains

Power Domain	V_{DD} Range
All internal logic	V_{DDINT}
I/O	V_{DDEXT}

The power dissipated by a processor is largely a function of the clock frequency of the processor and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in dynamic power dissipation, while reducing the voltage by 25% reduces dynamic power dissipation by more than 40%. Further, these power savings are additive, in that if the clock frequency and supply voltage are both reduced, the power savings can be dramatic.

The dynamic power management feature of the ADSP-BF561 allows both the processor's input voltage (V_{DDINT}) and clock frequency (f_{CCLK}) to be dynamically controlled.

The savings in power dissipation can be modeled using the power savings factor and % power savings calculations.

The power savings factor is calculated as:

$$\mathit{power\ savings\ factor}$$

$$= \frac{f_{CCLKRED}}{f_{CCLKNOM}} \times \left(\frac{V_{DDINTRED}}{V_{DDINTNOM}}\right)^2 \times \left(\frac{t_{RED}}{t_{NOM}}\right)$$

where the variables in the equations are:

$f_{CCLKNOM}$ is the nominal core clock frequency

$f_{CCLKRED}$ is the reduced core clock frequency

$V_{DDINTNOM}$ is the nominal internal supply voltage

$V_{DDINTRED}$ is the reduced internal supply voltage

t_{NOM} is the duration running at $f_{CCLKNOM}$

t_{RED} is the duration running at $f_{CCLKRED}$

The percent power savings is calculated as:

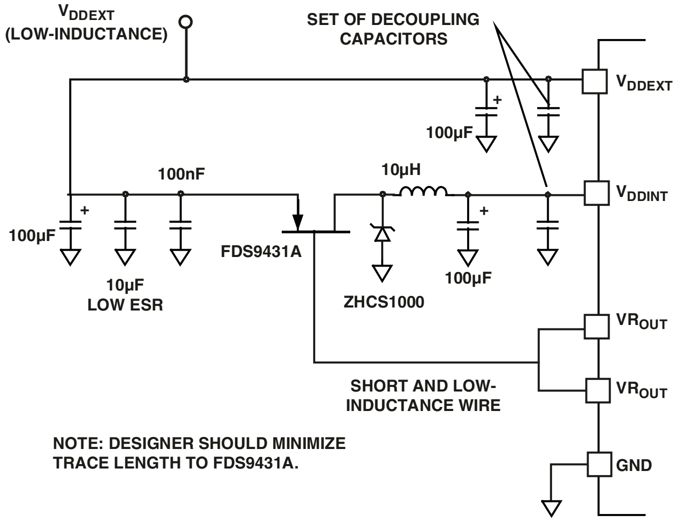
$$\%\ \mathit{power\ savings} = (1 - \mathit{power\ savings\ factor}) \times 100\%$$

VOLTAGE REGULATION

The ADSP-BF561 processor provides an on-chip voltage regulator that can generate appropriate V_{DDINT} voltage levels from the V_{DDEXT} supply. See Operating Conditions for regulator tolerances and acceptable V_{DDEXT} ranges for specific models.

Figure 4 shows the typical external components required to complete the power management system. The regulator controls the internal logic voltage levels and is programmable with the voltage regulator control register (VR_CTL) in increments of 50 mV. To reduce standby power consumption, the internal voltage regulator can be programmed to remove power to the processor core while keeping I/O power (V_{DDEXT}) supplied. While in the hibernate state, V_{DDEXT} can still be applied, thus eliminating the need for external buffers. The voltage regulator can be activated from this power-down state by asserting $\overline{\text{RESET}}$, which will then initiate a boot sequence. The regulator can also be disabled and bypassed at the user's discretion.

The internal voltage regulation feature is not available on any of the 600 MHz speed grade models. External voltage regulation is required to ensure correct operation of these parts at 600 MHz.



NOTE: DESIGNER SHOULD MINIMIZE TRACE LENGTH TO FDS9431A.

Figure 4. Voltage Regulator Circuit

Voltage Regulator Layout Guidelines

Regulator external component placement, board routing, and bypass capacitors all have a significant effect on noise injected into the other analog circuits on-chip. The VROUT1–0 traces and voltage regulator external components should be considered as noise sources when doing board layout and should not be routed or placed near sensitive circuits or components on the board. All internal and I/O power supplies should be well bypassed with bypass capacitors placed as close to the ADSP-BF561 processors as possible.

For further details on the on-chip voltage regulator and related board design guidelines, see the *Switching Regulator Design Considerations for ADSP-BF533 Blackfin Processors (EE-228)* applications note at www.analog.com—use site search on “EE-228”.

CLOCK SIGNALS

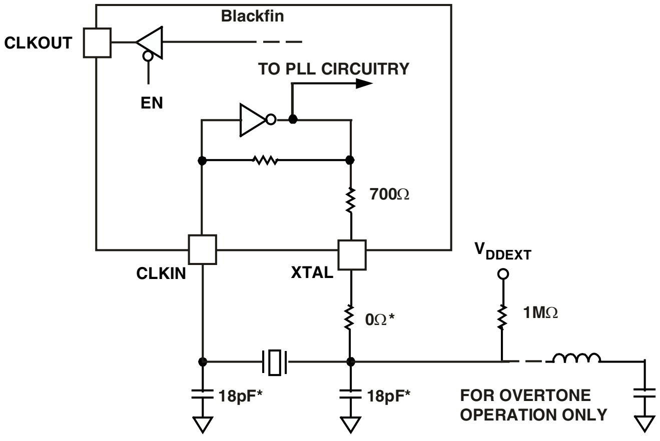
The ADSP-BF561 processor can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

If an external clock is used, it must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor’s CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the ADSP-BF561 processor includes an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 5. A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor shown in Figure 5 fine tune the phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 5 are typical values only. The capacitor values are dependent upon the crystal manufacturer’s load capacitance recommendations and the physical PCB layout. The resistor value depends on the drive level specified by the crystal manufacturer. System designs should verify the customized values based on careful investigation on multiple devices over the allowed temperature range.

A third-overtone crystal can be used at frequencies above 25 MHz. The circuit is then modified to ensure crystal operation only at the third overtone, by adding a tuned inductor circuit as shown in Figure 5.

As shown in Figure 6, the core clock (CCLK) and system peripheral clock (SCLK) are derived from the input clock (CLKIN) signal. An on-chip PLL is capable of multiplying the CLKIN signal by a user-programmable 0.5× to 64× multiplication factor. The default multiplier is 10×, but it can be modified by a software instruction sequence. On the fly frequency changes can be effected by simply writing to the PLL_DIV register.



NOTE: VALUES MARKED WITH * MUST BE CUSTOMIZED DEPENDING ON THE CRYSTAL AND LAYOUT. PLEASE ANALYZE CAREFULLY.

Figure 5. External Crystal Connections

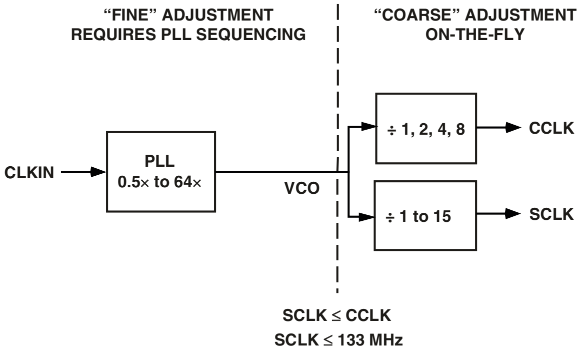


Figure 6. Frequency Modification Methods

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 5 illustrates typical system clock ratios.

Table 5. Example System Clock Ratios

Signal Name SSEL3–0	Divider Ratio VCO/SCLK	Example Frequency Ratios (MHz)	
		VCO	SCLK
0001	1:1	100	100
0110	6:1	300	50
1010	10:1	500	50

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 SSEL value can be changed dynamically without any PLL lock latencies by writing the appropriate values to the PLL divisor register (PLL_DIV).

The core clock (CCLK) frequency can also be dynamically changed by means of the CSEL1–0 bits of the PLL_DIV register. Supported CCLK divider ratios are 1, 2, 4, and 8, as shown in Table 6. This programmable core clock capability is useful for fast core frequency modifications.

Table 6. Core Clock Ratios

Signal Name CSEL1–0	Divider Ratio VCO/CCLK	Example Frequency Ratios (MHz)	
		VCO	CCLK
00	1:1	500	500
01	2:1	500	250
10	4:1	200	50
11	8:1	200	25

The maximum PLL clock time when a change is programmed via the PLL_CTL register is 40 μs. The maximum time to change the internal voltage via the internal voltage regulator is also 40 μs. The reset value for the PLL_LOCKCNT register is 0x200. This value should be programmed to ensure a 40 μs wakeup time when either the voltage is changed or a new MSEL value is programmed. The value should be programmed to ensure an 80 μs wakeup time when both voltage and the MSEL value are changed. The time base for the PLL_LOCKCNT register is the period of CLKIN.

BOOTING MODES

The ADSP-BF561 has three mechanisms (listed in Table 7) for automatically loading internal L1 instruction memory, L2, or external memory after a reset. A fourth mode is provided to execute from external memory, bypassing the boot sequence.

Table 7. Booting Modes

BMODE1–0	Description
00	Execute from 16-bit external memory (Bypass Boot ROM)
01	Boot from 8-bit/16-bit flash
10	Boot from SPI host slave mode
11	Boot from SPI serial EEPROM (16-, 24-bit address range)

The BMODE pins of the reset configuration register, sampled during power-on resets and software initiated resets, implement the following modes:

- Execute from 16-bit external memory – Execution starts from address 0x2000 0000 with 16-bit packing. The boot ROM is bypassed in this mode. All configuration settings are set for the slowest device possible (3-cycle hold time, 15-cycle R/W access times, 4-cycle setup). Note that, in bypass mode, only Core A can execute instructions from external memory.
- Boot from 8-bit/16-bit external flash memory – The 8-bit/16-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 host device – The Blackfin processor operates in SPI slave mode and is configured to receive the bytes of the .LDR file from an SPI host (master) agent. To hold off the host device from transmitting while the boot ROM is busy, the Blackfin processor asserts a GPIO pin, called host wait (HWAIT), to signal the host device not to send any more bytes until the flag is deasserted. The flag is chosen by the user and this information is transferred to the Blackfin processor via bits 10:5 of the FLAG header.
- Boot from SPI serial EEPROM (16-, 24-bit addressable) – The SPI uses the PF2 output pin to select a single SPI EPROM device, submits a read command at address 0x0000, and begins clocking data into the beginning of L1 instruction memory. A 16-, 24-bit addressable SPI-compatible EPROM must be used.

For each of the boot modes, a boot loading protocol is used to transfer program and data blocks from an external memory device to their specified memory locations. Multiple memory blocks may be loaded by any boot sequence. Once all blocks are loaded, Core A program execution commences from the start of L1 instruction SRAM (0xFFA0 0000). Core B remains in a held-off state until Bit 5 of SICA_SYSCR is cleared by Core A. After that, Core B will start execution at address 0xFF60 0000.

In addition, Bit 4 of the reset configuration register can be set by application code to bypass the normal boot sequence during a software reset. For this case, the processor jumps directly to the beginning of L1 instruction memory.

INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax that was designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the 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 operation—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 multi-issue load/store modified Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU plus two load/store plus two pointer updates per cycle.

- All registers, I/O, and memory are mapped into a unified 4G byte 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-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded as 16-bits.

DEVELOPMENT TOOLS

Analog Devices supports its processors with a complete line of software and hardware development tools, including integrated development environments (which include CrossCore® Embedded Studio and/or VisualDSP++®), evaluation products, emulators, and a wide variety of software add-ins.

Integrated Development Environments (IDEs)

For C/C++ software writing and editing, code generation, and debug support, Analog Devices offers two IDEs.

The newest IDE, CrossCore Embedded Studio (CCES), is based on the Eclipse framework. Supporting most Analog Devices processor families, it is the IDE of choice for future processors, including multicore devices. CCES Studio seamlessly integrates available software add-ins to support real time operating systems, file systems, TCP/IP stacks, USB stacks, algorithmic software modules, and evaluation hardware board support packages. For more information visit www.analog.com/cces.

The other Analog Devices IDE, VisualDSP++, supports processor families introduced prior to the release of CCES. This IDE includes the Analog Devices VDK real time operating system and an open source TCP/IP stack. For more information visit www.analog.com/visualdsp. Note that VisualDSP++ will not support future Analog Devices processors.

EZ-KIT Lite Evaluation Board

For processor evaluation, Analog Devices provides wide range of EZ-KIT Lite® evaluation boards. Including the processor and key peripherals, the evaluation board also supports on-chip emulation capabilities and other evaluation and development features. Also available are various EZ-Extenders®, which are daughter cards delivering additional specialized functionality, including audio and video processing. For more information visit www.analog.com and search on “ezkit” or “ezextender”.

EZ-KIT Lite Evaluation Kits

For a cost-effective way to learn more about developing with Analog Devices processors, Analog Devices offer a range of EZ-KIT Lite evaluation kits. Each evaluation kit includes an EZ-KIT Lite evaluation board, directions for downloading an evaluation version of the available IDE(s), a USB cable, and a power supply. The USB controller on the EZ-KIT Lite board connects to the USB port of the user’s PC, enabling the chosen IDE evaluation suite to emulate the on-board processor in-circuit. This permits the customer to download, execute, and debug programs for the EZ-KIT Lite system. It also supports in-circuit programming of the on-board Flash device to store user-specific boot code, enabling standalone operation. With the full version of CCES or VisualDSP++ installed (sold separately), engineers can develop software for supported EZ-KITs or any custom system utilizing supported Analog Devices processors.

Software Add-Ins for CCES

Analog Devices offers software add-ins which seamlessly integrate with CCES to extend its capabilities and reduce development time. Add-ins include board support packages for evaluation hardware, various middleware packages, and algorithmic modules. Documentation, help, configuration dialogs, and coding examples present in these add-ins are viewable through the CCES IDE once the add-in is installed.

Board Support Packages for Evaluation Hardware

Software support for the EZ-KIT Lite evaluation boards and EZ-Extender daughter cards is provided by software add-ins called Board Support Packages (BSPs). The BSPs contain the required drivers, pertinent release notes, and select example code for the given evaluation hardware. A download link for a specific BSP is located on the web page for the associated EZ-KIT or EZ-Extender product. The link is found in the *Product Download* area of the product web page.

Middleware Packages

Analog Devices separately offers middleware add-ins such as real time operating systems, file systems, USB stacks, and TCP/IP stacks. For more information see the following web pages:

- www.analog.com/ucos3
- www.analog.com/ucfs
- www.analog.com/ucusbd
- www.analog.com/lwip

Algorithmic Modules

To speed development, Analog Devices offers add-ins that perform popular audio and video processing algorithms. These are available for use with both CCES and VisualDSP++. For more information visit www.analog.com and search on “Blackfin software modules” or “SHARC software modules”.

Designing an Emulator-Compatible DSP Board (Target)

For embedded system test and debug, Analog Devices provides a family of emulators. On each JTAG DSP, Analog Devices supplies an IEEE 1149.1 JTAG Test Access Port (TAP). In-circuit emulation is facilitated by use of this JTAG interface. The emulator accesses the processor’s internal features via the processor’s TAP, allowing the developer to load code, set breakpoints, and view variables, memory, and registers. The processor must be halted to send data and commands, but once an operation is completed by the emulator, the DSP system is set to run at full speed with no impact on system timing. The emulators require the target board to include a header that supports connection of the DSP’s JTAG port to the emulator.

For details on target board design issues including mechanical layout, single processor connections, 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

The following publications that describe the ADSP-BF561 processors (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on our website:

- *Getting Started With Blackfin Processors*
- *ADSP-BF561 Blackfin Processor Hardware Reference*
- *ADSP-BF53x/BF56x Blackfin Processor Programming Reference*
- ADSP-BF561 *Blackfin Processor Anomaly List*

RELATED SIGNAL CHAINS

A signal chain is a series of signal-conditioning electronic components that receive input (data acquired from sampling either real-time phenomena or from stored data) in tandem, with the output of one portion of the chain supplying input to the next. Signal chains are often used in signal processing applications to gather and process data or to apply system controls based on analysis of real-time phenomena.

Analog Devices eases signal processing system development by providing signal processing components that are designed to work together well. A tool for viewing relationships between specific applications and related components is available on the www.analog.com website.

The Application Signal Chains page in the Circuits from the Lab® site (www.analog.com/circuits) provides:

- Graphical circuit block diagram presentation of signal chains for a variety of circuit types and applications
- Drill down links for components in each chain to selection guides and application information
- Reference designs applying best practice design techniques

PIN DESCRIPTIONS

ADSP-BF561 pin definitions are listed in Table 8. In order to maintain maximum function and reduce package size and pin count, some pins have multiple functions. In cases where pin function is reconfigurable, the default state is shown in plain text, while alternate functionality is shown in italics.

All pins are three-stated during and immediately after reset, except the external memory interface, asynchronous memory control, and synchronous memory control pins. These pins are all driven high, with the exception of CLKOUT, which toggles at the system clock rate. However if $\overline{\text{BR}}$ is active, the memory pins are also three-stated.

All I/O pins have their input buffers disabled, with the exception of the pins that need pull-ups or pull-downs if unused, as noted in Table 8.

Table 8. Pin Descriptions

Pin Name	Type	Function	Driver Type[1]
EBIU			
ADDR25–2	O	Address Bus for Async/Sync Access	A
DATA31–0	I/O	Data Bus for Async/Sync Access	A
$\overline{\text{ABE3–0}}$/SDQM3–0	O	Byte Enables/Data Masks for Async/Sync Access	A
$\overline{\text{BR}}$	I	Bus Request (This pin should be pulled HIGH if not used.)	
$\overline{\text{BG}}$	O	Bus Grant	A
$\overline{\text{BGH}}$	O	Bus Grant Hang	A
EBIU (ASYNC)			
$\overline{\text{AMS3–0}}$	O	Bank Select	A
ARDY	I	Hardware Ready Control (This pin should be pulled HIGH if not used.)	
$\overline{\text{AOE}}$	O	Output Enable	A
$\overline{\text{AWE}}$	O	Write Enable	A
$\overline{\text{ARE}}$	O	Read Enable	A
EBIU (SDRAM)			
$\overline{\text{SRAS}}$	O	Row Address Strobe	A
$\overline{\text{SCAS}}$	O	Column Address Strobe	A
$\overline{\text{SWE}}$	O	Write Enable	A
SCKE	O	Clock Enable	A
SCLK0/CLKOUT	O	Clock Output Pin 0	B
SCLK1	O	Clock Output Pin 1	B
SA10	O	SDRAM A10 Pin	A
$\overline{\text{SMS3–0}}$	O	Bank Select	A

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type[1]
PF/SPI/TIMER			
PF0/*$\overline{\text{SPISS}}$/TMR0*	I/O	Programmable Flag/*Slave SPI Select/Timer*	C
PF1/*$\overline{\text{SPISEL1}}$/TMR1*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF2/*$\overline{\text{SPISEL2}}$/TMR2*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF3/*$\overline{\text{SPISEL3}}$/TMR3*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF4/*$\overline{\text{SPISEL4}}$/TMR4*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF5/*$\overline{\text{SPISEL5}}$/TMR5*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF6/*$\overline{\text{SPISEL6}}$/TMR6*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF7/*$\overline{\text{SPISEL7}}$/TMR7*	I/O	Programmable Flag/*SPI Select/Timer*	C
PF8	I/O	Programmable Flag	C
PF9	I/O	Programmable Flag	C
PF10	I/O	Programmable Flag	C
PF11	I/O	Programmable Flag	C
PF12	I/O	Programmable Flag	C
PF13	I/O	Programmable Flag	C
PF14	I/O	Programmable Flag	C
PF15/*EXT CLK*	I/O	Programmable Flag/*External Timer Clock Input*	C
PPI0			
PPI0D15–8/*PF47–40*	I/O	PPI Data/*Programmable Flag Pins*	C
PPI0D7–0	I/O	PPI Data Pins	C
PPI0CLK	I	PPI Clock	
PPI0SYNC1/*TMR8*	I/O	PPI Sync/*Timer*	C
PPI0SYNC2/*TMR9*	I/O	PPI Sync/*Timer*	C
PPI0SYNC3	I/O	PPI Sync	C
PPI1			
PPI1D15–8/*PF39–32*	I/O	PPI Data/*Programmable Flag Pins*	C
PPI1D7–0	I/O	PPI Data Pins	C
PPI1CLK	I	PPI Clock	
PPI1SYNC1/*TMR10*	I/O	PPI Sync/*Timer*	C
PPI1SYNC2/*TMR11*	I/O	PPI Sync/*Timer*	C
PPI1SYNC3	I/O	PPI Sync	C
SPORT0			
RSCLK0/*PF28*	I/O	Sport0 Receive Serial Clock/*Programmable Flag*	D
RFS0/*PF19*	I/O	Sport0 Receive Frame Sync/*Programmable Flag*	C
DR0PRI	I	Sport0 Receive Data Primary	
DR0SEC/*PF20*	I/O	Sport0 Receive Data Secondary/*Programmable Flag*	C
TSCLK0/*PF29*	I/O	Sport0 Transmit Serial Clock/*Programmable Flag*	D
TFS0/*PF16*	I/O	Sport0 Transmit Frame Sync/*Programmable Flag*	C
DT0PRI/*PF18*	I/O	Sport0 Transmit Data Primary/*Programmable Flag*	C
DT0SEC/*PF17*	I/O	Sport0 Transmit Data Secondary/*Programmable Flag*	C

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type[1]
SPORT1			
RSCLK1/*PF30*	I/O	Sport1 Receive Serial Clock/*Programmable Flag*	D
RFS1/*PF24*	I/O	Sport1 Receive Frame Sync/*Programmable Flag*	C
DR1PRI	I	Sport1 Receive Data Primary	
DR1SEC/*PF25*	I/O	Sport1 Receive Data Secondary/*Programmable Flag*	C
TSCLK1/*PF31*	I/O	Sport1 Transmit Serial Clock/*Programmable Flag*	D
TFS1/*PF21*	I/O	Sport1 Transmit Frame Sync/*Programmable Flag*	C
DT1PRI/*PF23*	I/O	Sport1 Transmit Data Primary/*Programmable Flag*	C
DT1SEC/*PF22*	I/O	Sport1 Transmit Data Secondary/*Programmable Flag*	C
SPI			
MOSI	I/O	Master Out Slave In	C
MISO	I/O	Master In Slave Out (This pin should be pulled HIGH through a 4.7 kΩ resistor if booting via the SPI port.)	C
SCK	I/O	SPI Clock	D
UART			
RX/*PF27*	I/O	UART Receive/*Programmable Flag*	C
TX/*PF26*	I/O	UART Transmit/*Programmable Flag*	C
JTAG			
$\overline{\text{EMU}}$	O	Emulation Output	C
TCK	I	JTAG Clock	
TDO	O	JTAG Serial Data Out	C
TDI	I	JTAG Serial Data In	
TMS	I	JTAG Mode Select	
$\overline{\text{TRST}}$	I	JTAG Reset (This pin should be pulled LOW if JTAG is not used.)	
Clock			
CLKIN	I	Clock/Crystal Input (This pin needs to be at a level or clocking.)	
XTAL	O	Crystal Connection	
Mode Controls			
$\overline{\text{RESET}}$	I	Reset (This pin is always active during core power-on.)	
NMI0	I	Nonmaskable Interrupt Core A (This pin should be pulled LOW when not used.)	
NMI1	I	Nonmaskable Interrupt Core B (This pin should be pulled LOW when not used.)	
BMODE1–0	I	Boot Mode Strap (These pins must be pulled to the state required for the desired boot mode.)	
SLEEP	O	Sleep	C
BYPASS	I	PLL BYPASS Control (Pull-up or pull-down Required.)	
Voltage Regulator			
V_{ROUT}1–0	O	External FET Drive	
Supplies			
V_{DDEXT}	P	Power Supply	
V_{DDINT}	P	Power Supply	
GND	G	Power Supply Return	
No Connection	NC	NC	

[1] Refer to Figure 32 to Figure 36.

SPECIFICATIONS

Component specifications are subject to change without notice.

OPERATING CONDITIONS

Parameter		Conditions	Min	Nominal	Max	Unit
V_{DDINT}	Internal Supply Voltage[1]	500 MHz and 533 MHz speed grade models[2]	0.8	1.25	1.375	V
V_{DDINT}	Internal Supply Voltage[3]	600 MHz speed grade models[2]	0.8	1.35	1.4185	V
V_{DDEXT}	External Supply Voltage		2.25	2.5, or 3.3	3.6	V
V_{IH}	High Level Input Voltage[4, 5]		2.0		3.6	V
V_{IL}	Low Level Input Voltage[5]		–0.3		+0.6	V
T_J	Junction Temperature	256-Ball CSP_BGA (12 mm × 12 mm) @ $T_{AMBIENT}$ = 0°C to 70°C	0		105	°C
T_J	Junction Temperature	256-Ball CSP_BGA (17 mm × 17 mm) @ $T_{AMBIENT}$ = 0°C to 70°C	0		95	°C
T_J	Junction Temperature	256-Ball CSP_BGA (17 mm × 17 mm) @ $T_{AMBIENT}$ = –40°C to +85°C	–40		+115	°C
T_J	Junction Temperature	297-Ball PBGA @ $T_{AMBIENT}$ = 0°C to 70°C	0		95	°C
T_J	Junction Temperature	297-Ball PBGA @ $T_{AMBIENT}$ = –40°C to +85°C	–40		+115	°C

[1] Internal voltage (V_{DDINT}) regulator tolerance is –5% to +10% for all models.

[2] See Ordering Guide.

[3] The internal voltage regulation feature is not available. External voltage regulation is required to ensure correct operation.

[4] The ADSP-BF561 is 3.3 V tolerant (always accepts up to 3.6 V maximum V_{IH}), but voltage compliance (on outputs, V_{OH}) depends on the input V_{DDEXT}, because V_{OH} (maximum) approximately equals V_{DDEXT} (maximum). This 3.3 V tolerance applies to bidirectional and input only pins.

[5] Applies to all signal pins.

Table 9 and Table 10 describe the timing requirements for the ADSP-BF561 clocks ($t_{CCLK} = 1/f_{CCLK}$). Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock, system clock, and Voltage Controlled Oscillator (VCO) operating frequencies, as described in Absolute Maximum Ratings. Table 11 describes phase-locked loop operating conditions.

Table 9. Core Clock (CCLK) Requirements—500 MHz and 533 MHz Speed Grade Models[1]

Parameter		Max	Unit
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.235 Vminimum)[2]	533	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.1875 Vminimum)	500	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.045 Vminimum)	444	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.95 Vminimum)	350	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.855 Vminimum	300	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.8 V minimum)	250	MHz

[1] See Ordering Guide.

[2] Does not apply to 500 MHz speed grade models.

Table 10. Core Clock (CCLK) Requirements—600 MHz Speed Grade Models[1]

Parameter		Max	Unit
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.2825 V minimum)[2]	600	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.235 V minimum)	533	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.1875 V minimum)	500	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 1.045 V minimum)	444	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.95 V minimum)	350	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.855 V minimum)	300	MHz
f_{CCLK}	CCLK Frequency (V_{DDINT} = 0.8 V minimum)	250	MHz

[1] See Ordering Guide.

[2] External voltage regulator required to ensure proper operation at 600 MHz.

Table 11. Phase-Locked Loop Operating Conditions

Parameter	Min	Max	Unit
Voltage Controlled Oscillator (VCO) Frequency	50	Maximum f_{CCLK}	MHz

Table 12. System Clock (SCLK) Requirements

Parameter[1]		Max V_{DDEXT} = 2.5V/3.3V	Unit
f_{SCLK}	CLKOUT/SCLK Frequency ($V_{DDINT} \geq 1.14$ V)	133[2]	MHz
f_{SCLK}	CLKOUT/SCLK Frequency ($V_{DDINT} < 1.14$ V)	100	MHz

[1] t_{SCLK} (= 1/f_{SCLK}) must be greater than or equal to t_{CCLK}.

[2] Rounded number. Guaranteed to t_{SCLK} = 7.5 ns. See Table 19.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{OH}	High Level Output Voltage[1]	V_{DDEXT} = 3.0 V, I_{OH} = –0.5 mA	2.4			V
V_{OL}	Low Level Output Voltage[1]	V_{DDEXT} = 3.0 V, I_{OL} = 2.0 mA			0.4	V
I_{IH}	High Level Input Current[2]	V_{DDEXT} = Maximum, V_{IN} = V_{DD} Maximum			10.0	μA
I_{IHP}	High Level Input Current JTAG[3]	V_{DDEXT} = Maximum, V_{IN} = V_{DD} Maximum			50.0	μA
I_{IL}[4]	Low Level Input Current[2]	V_{DDEXT} = Maximum, V_{IN} = 0 V			10.0	μA
I_{OZH}	Three-State Leakage Current[5]	V_{DDEXT} = Maximum, V_{IN} = V_{DD} Maximum			10.0	μA
I_{OZL}[4]	Three-State Leakage Current[5]	V_{DDEXT} = Maximum, V_{IN} = 0 V			10.0	μA
C_{IN}	Input Capacitance[6]	f_{IN} = 1 MHz, $T_{AMBIENT}$ = 25°C, V_{IN} = 2.5 V		4	8[7]	pF
$I_{DDHIBERNATE}$[8]	V_{DDEXT} Current in Hibernate Mode	CLKIN = 0 MHz, V_{DDEXT} = 3.65 V with Voltage Regulator Off (V_{DDINT} = 0 V)		50		μA
$I_{DDDEEPSLEEP}$[9]	V_{DDINT} Current in Deep Sleep Mode	V_{DDINT} = 0.8 V, $T_{JUNCTION}$ = 25°C		70		mA
I_{DD_TYP}[9, 10]	V_{DDINT} Current	V_{DDINT} = 0.8 V, f_{CCLK} = 50 MHz, $T_{JUNCTION}$ = 25°C		127		mA
I_{DD_TYP}[9, 10]	V_{DDINT} Current	V_{DDINT} = 1.25 V, f_{CCLK} = 500 MHz, $T_{JUNCTION}$ = 25°C		660		mA
I_{DD_TYP}[9, 10]	V_{DDINT} Current	V_{DDINT} = 1.35 V, f_{CCLK} = 600 MHz, $T_{JUNCTION}$ = 25°C		818		mA

[1] Applies to output and bidirectional pins.

[2] Applies to input pins except JTAG inputs.

[3] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{\text{TRST}}$).

[4] Absolute value.

[5] Applies to three-statable pins.

[6] Applies to all signal pins.

[7] Guaranteed, but not tested.

[8] CLKIN must be tied to V_{DDEXT} or GND during hibernate.

[9] Maximum current drawn. See *Estimating Power for ADSP-BF561 Blackfin Processors (EE-293)* on the Analog Devices website (www.analog.com)—use site search on “EE-293”.

[10] Both cores executing 75% dual MAC, 25% ADD instructions with moderate data bus activity.

System designers should refer to *Estimating Power for the ADSP-BF561 (EE-293)*, which provides detailed information for optimizing designs for lowest power. All topics discussed in this section are described in detail in EE-293. Total power dissipation has two components:

1. Static, including leakage current
2. Dynamic, due to transistor switching characteristics

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. Electrical Characteristics shows the current dissipation for internal circuitry (V_{DDINT}).

ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed in Table 13 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.

Table 13. Absolute Maximum Ratings

Parameter	Value
Internal (Core) Supply Voltage (V_{DDINT})	−0.3 V to +1.42 V
External (I/O) Supply Voltage (V_{DDEXT})	−0.5 V to +3.8 V
Input Voltage[1]	−0.5 V to +3.8 V
Output Voltage Swing	−0.5 V to V_{DDEXT} + 0.5 V
Load Capacitance[2]	200 pF
Storage Temperature Range	−65°C to +150°C
Junction Temperature Under Bias	125°C

[1] Applies to 100% transient duty cycle. For other duty cycles see Table 14.
[2] For proper SDRAM controller operation, the maximum load capacitance is 50 pF (at 3.3 V) or 30 pF (at 2.5 V) for ADDR19–1, DATA15–0, ABE1–0/SDQM1–0, CLKOUT, SCKE, SA10, SRAS, SCAS, SWE, and SMS.

Table 14. Maximum Duty Cycle for Input Transient Voltage[1]

V_{IN} Min (V)	V_{IN} Max (V)[2]	Maximum Duty Cycle
−0.50	3.80	100%
−0.70	4.00	40%
−0.80	4.10	25%
−0.90	4.20	15%
−1.00	4.30	10%

[1] Applies to all signal pins with the exception of CLKIN, XTAL, VROUT1–0.
[2] Only one of the listed options can apply to a particular design.

ESD SENSITIVITY

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

TIMING SPECIFICATIONS

Clock and Reset Timing

Table 15 and Figure 7 describe clock and reset operations. Per Absolute Maximum Ratings, combinations of CLKIN and clock multipliers must not result in core/system clocks exceeding the maximum limits allowed for the processor, including system clock restrictions related to supply voltage.

Table 15. Clock and Normal Reset Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{CKIN}	CLKIN (to PLL) Period[1, 2, 3]	25.0	100.0	ns
t_{CKINL}	CLKIN Low Pulse	10.0		ns
t_{CKINH}	CLKIN High Pulse	10.0		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulse Width Low[4]	11 × t_{CKIN}		ns

[1] If DF bit in PLL_CTL register is set t_{CLKIN} is divided by two before going to PLL, then the t_{CLKIN} maximum period is 50 ns and the t_{CLKIN} minimum period is 12.5 ns.

[2] Applies to PLL bypass mode and PLL nonbypass mode.

[3] Combinations of the CLKIN frequency and the PLL clock multiplier must not exceed the allowed f_{VCO}, f_{CCLK}, and f_{SCLK} settings discussed in Table 9 through Table 12.

[4] Applies after power-up sequence is complete. See Table 16 and Figure 8 for power-up reset timing.

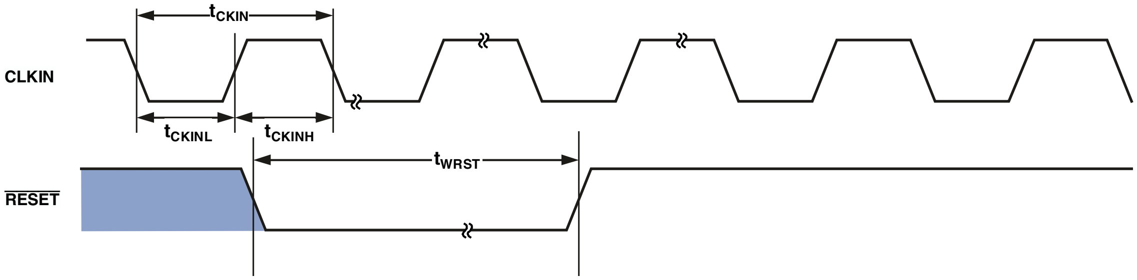


Figure 7. Clock and Normal Reset Timing

Table 16. Power-Up Reset Timing

Parameter		Min	Max	Unit
Timing Requirement				
$t_{RST\_IN\_PWR}$	$\overline{\text{RESET}}$ Deasserted after the V_{DDINT}, V_{DDEXT}, and CLKIN Pins are Stable and Within Specification	3500 × t_{CKIN}		µs

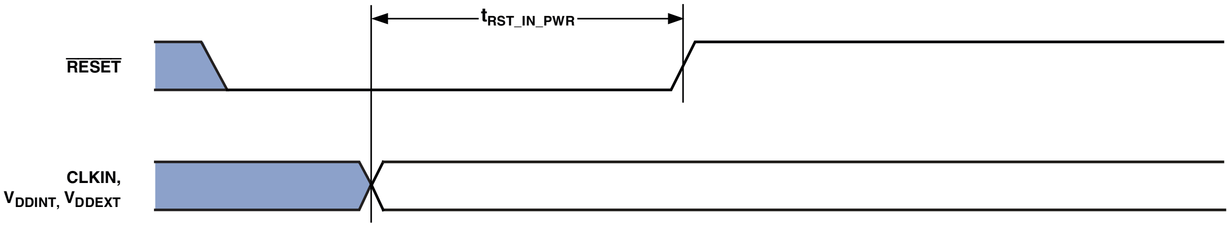


Figure 8. Power-Up Reset Timing

Asynchronous Memory Read Cycle Timing

Table 17. 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	0.8		ns
t_{SARDY}	ARDY Setup Before CLKOUT	4.0		ns
t_{HARDY}	ARDY Hold After CLKOUT	0.0		ns
Switching Characteristics				
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT [1]	0.8		ns

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

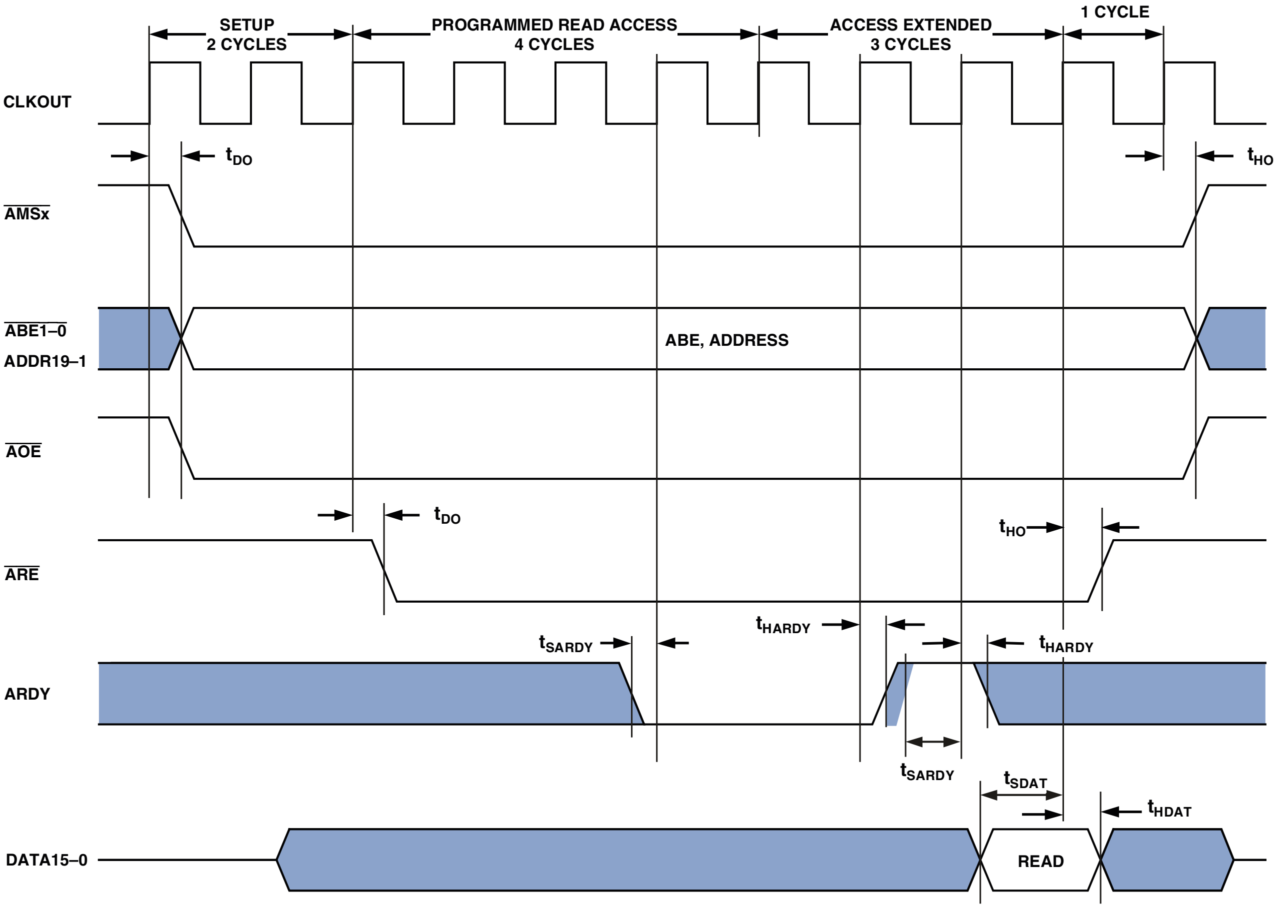


Figure 9. Asynchronous Memory Read Cycle Timing

Asynchronous Memory Write Cycle Timing

Table 18. 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	0.0		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]		6.0	ns
t_{HO}	Output Hold After CLKOUT [1]	0.8		ns

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

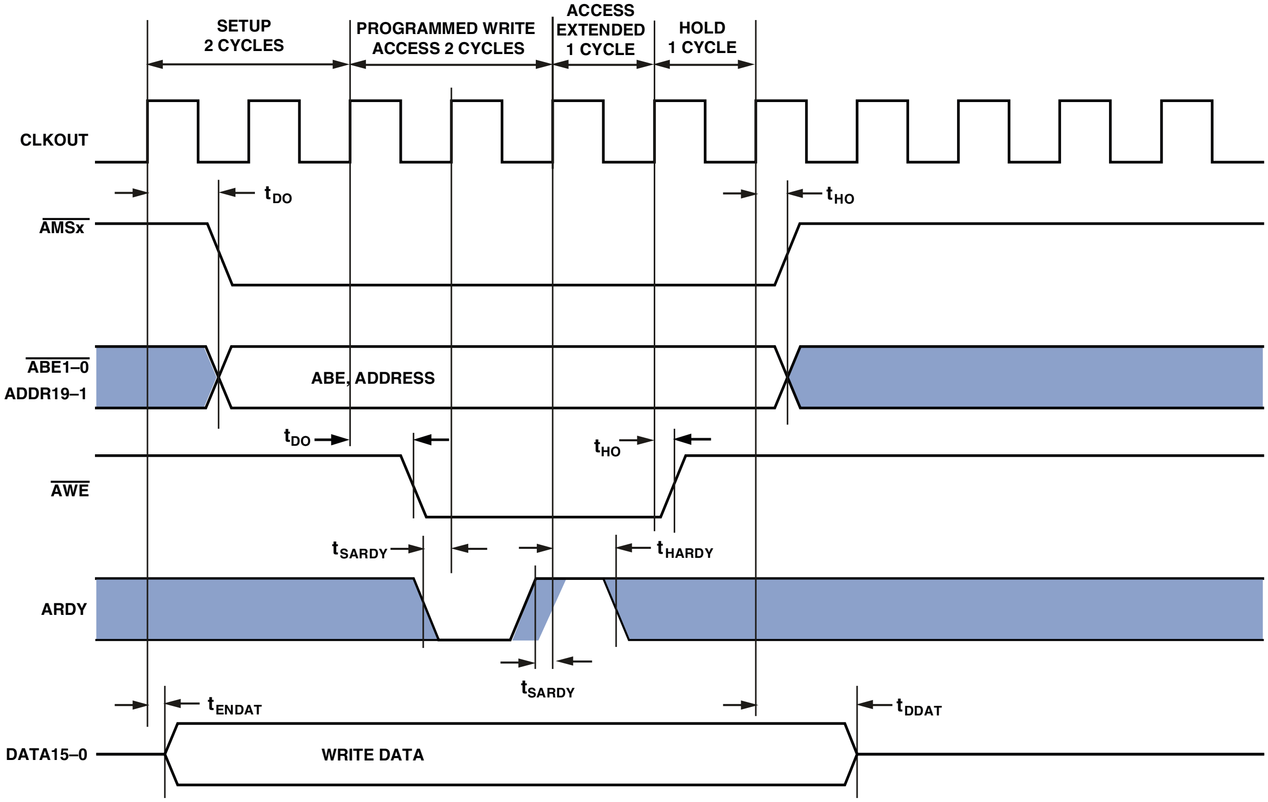


Figure 10. Asynchronous Memory Write Cycle Timing

SDRAM Interface Timing

Table 19. SDRAM Interface Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSDAT}	DATA Setup Before CLKOUT	1.5		ns
t_{HSDAT}	DATA Hold After CLKOUT	0.8		ns
Switching Characteristics				
t_{DCAD}	Command, ADDR, Data Delay After CLKOUT[1]		4.0	ns
t_{HCAD}	Command, ADDR, Data Hold After CLKOUT[1]	0.8		ns
t_{DSDAT}	Data Disable After CLKOUT		4.0	ns
t_{ENSDAT}	Data Enable After CLKOUT	1.0		ns
t_{SCLK}	CLKOUT Period	7.5		ns
t_{SCLKH}	CLKOUT Width High	2.5		ns
t_{SCLKL}	CLKOUT Width Low	2.5		ns

[1] Command pins include: $\overline{\text{SRAS}}$, $\overline{\text{SCAS}}$, $\overline{\text{SWE}}$, SDQM, $\overline{\text{SMS3–0}}$, SA10, SCKE.

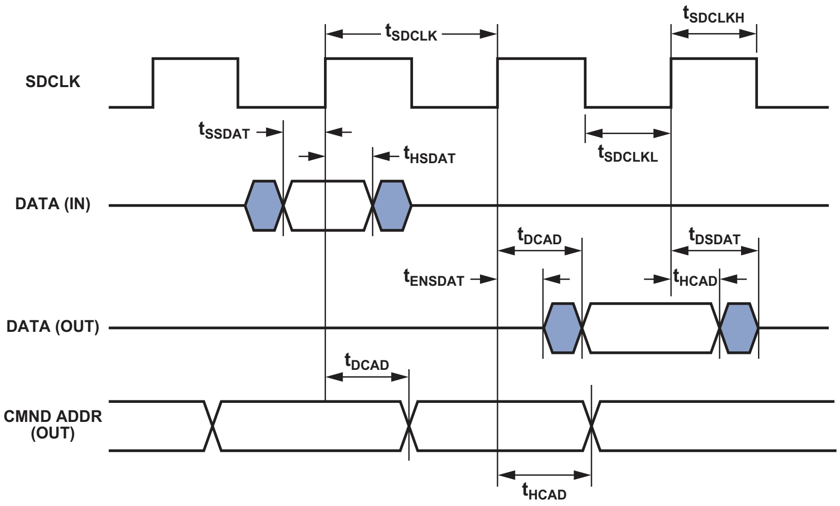


Figure 11. SDRAM Interface Timing

External Port Bus Request and Grant Cycle Timing

Table 20 and Figure 12 describe external port bus request and bus grant operations.

Table 20. External Port Bus Request and Grant Cycle Timing

Parameter[1, 2]		Min	Max	Unit
Timing Requirements				
t_{BS}	$\overline{\text{BR}}$ Asserted to CLKOUT High Setup	4.6		ns
t_{BH}	CLKOUT High to $\overline{\text{BR}}$ Deasserted Hold Time	0.0		ns
Switching Characteristics				
t_{SD}	CLKOUT Low to $\overline{\text{AMSx}}$, Address and $\overline{\text{ARE}}$/$\overline{\text{AWE}}$ Disable		4.5	ns
t_{SE}	CLKOUT Low to $\overline{\text{AMSx}}$, Address and $\overline{\text{ARE}}$/$\overline{\text{AWE}}$ Enable		4.5	ns
t_{DBG}	CLKOUT High to $\overline{\text{BG}}$ Asserted Setup		3.6	ns
t_{EBG}	CLKOUT High to $\overline{\text{BG}}$ Deasserted Hold Time		3.6	ns
t_{DBH}	CLKOUT High to $\overline{\text{BGH}}$ Asserted Setup		3.6	ns
t_{EBH}	CLKOUT High to $\overline{\text{BGH}}$ Deasserted Hold Time		3.6	ns

[1] These are preliminary timing parameters that are based on worst-case operating conditions.

[2] The pad loads for these timing parameters are 20 pF.

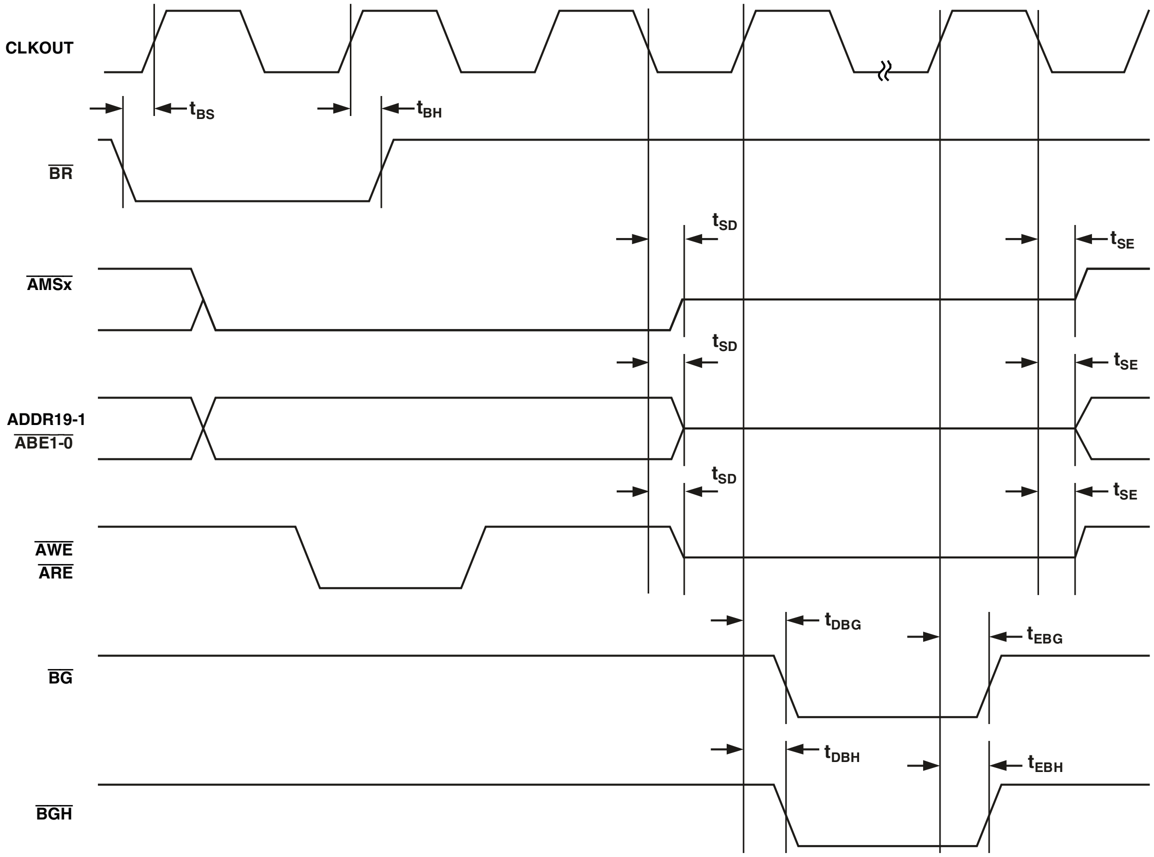


Figure 12. External Port Bus Request and Grant Cycle Timing

Parallel Peripheral Interface Timing

Table 21, and Figure 13 through Figure 18, describe default parallel peripheral interface operations.

Table 21. Parallel Peripheral Interface Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{PCLKW}	PPIxCLK Width[1]	5.0		ns
t_{PCLK}	PPIxCLK Period[1]	13.3		ns
t_{SFSPE}	External Frame Sync Setup Before PPIxCLK (Nonsampling Edge for Rx, Sampling Edge for Tx)[2]	4.0		ns
t_{HFSPE}	External Frame Sync Hold After PPIxCLK[2]	1.0		ns
t_{SDRPE}	Receive Data Setup Before PPIxCLK	3.5		ns
t_{HDRPE}	Receive Data Hold After PPIxCLK	2.0		ns
Switching Characteristics—GP Output and Frame Capture Modes				
t_{DFSPE}	Internal Frame Sync Delay After PPIxCLK		8.0	ns
t_{HOFSPE}	Internal Frame Sync Hold After PPIxCLK	1.7		ns
t_{DDTPE}	Transmit Data Delay After PPIxCLK		8.0	ns
t_{HDTPE}	Transmit Data Hold After PPIxCLK	2.0		ns

[1] For PPI modes that use an internally generated frame sync, the PPIxCLK frequency cannot exceed $f_{SCLK}/2$. For modes with no frame syncs or external frame syncs, PPIxCLK cannot exceed 75 MHz and f_{SCLK} should be equal to or greater than PPIxCLK.

[2] Applies when PLL_CTL bit 4 is cleared. See Figure 15 and Figure 18.

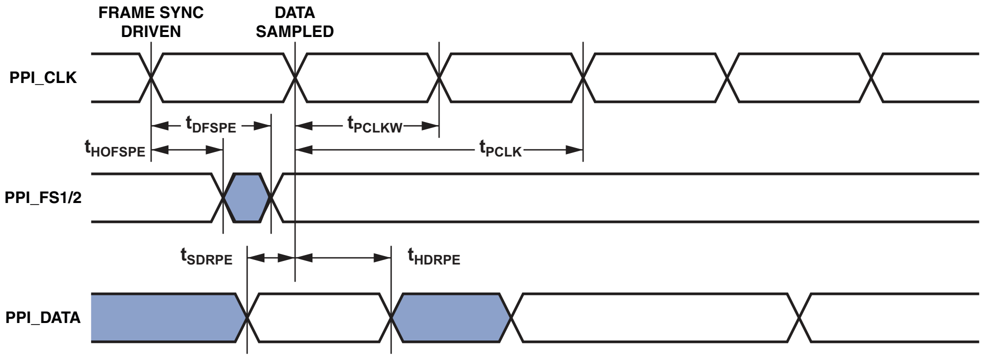


Figure 13. PPI GP Rx Mode with Internal Frame Sync Timing

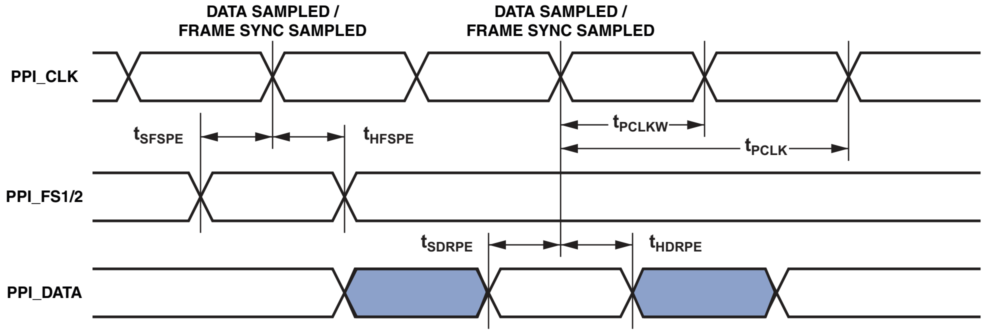


Figure 14. PPI GP Rx Mode with External Frame Sync Timing (PLL_CTL Bit 4 = 1)

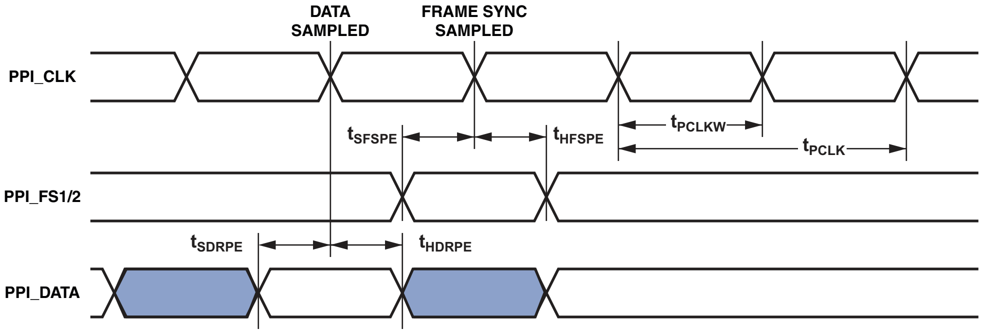


Figure 15. PPI GP Rx Mode with External Frame Sync Timing (PLL_CTL Bit 4 = 0)

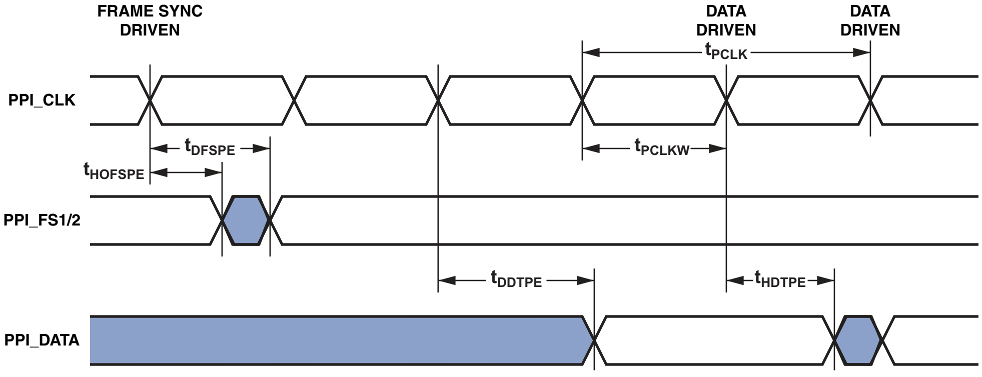


Figure 16. PPI GP Tx Mode with Internal Frame Sync Timing

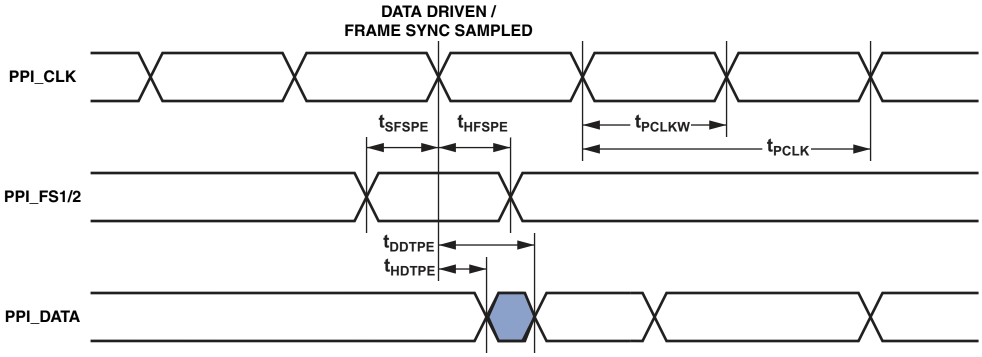


Figure 17. PPI GP Tx Mode with External Frame Sync Timing (PLL_CTL Bit 4 = 1)

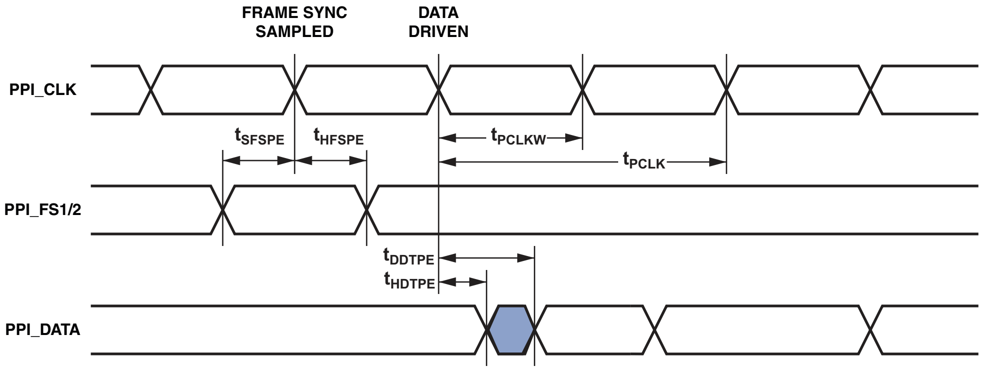


Figure 18. PPI GP Tx Mode with External Frame Sync Timing (PLL_CTL Bit 4 = 0)

Serial Ports

Table 22 through Table 25 and Figure 19 through Figure 22 describe Serial Port operations.

Table 22. Serial Ports—External Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSE}	TFSx/RFSx Setup Before TSCLKx/RSCLKx[1]	3.0		ns
t_{HFSE}	TFSx/RFSx Hold After TSCLKx/RSCLKx[1]	3.0		ns
t_{SDRE}	Receive Data Setup Before RSCLKx[1]	3.0		ns
t_{HDRE}	Receive Data Hold After RSCLKx[1]	3.0		ns
t_{SCLKEW}	TSCLKx/RSCLKx Width	4.5		ns
t_{SCLKE}	TSCLKx/RSCLKx Period	15.0		ns
t_{SUDTE}	Start-Up Delay From SPORT Enable To First External TFSx[2]	$4 \times t_{SCLKE}$		ns
t_{SUDRE}	Start-Up Delay From SPORT Enable To First External RFSx[2]	$4 \times t_{SCLKE}$		ns
Switching Characteristics				
t_{DFSE}	TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[3]		10.0	ns
t_{HOFSE}	TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[3]	0.0		ns
t_{DDTE}	Transmit Data Delay After TSCLKx[3]		10.0	ns
t_{HDTE}	Transmit Data Hold After TSCLKx[3]	0.0		ns

[1] Referenced to sample edge.

[2] Verified in design but untested. After being enabled, the serial port requires external clock pulses—before the first external frame sync edge—to initialize the serial port.

[3] Referenced to drive edge.

Table 23. Serial Ports—Internal Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSI}	TFSx/RFSx Setup Before TSCLKx/RSCLKx[1]	8.0		ns
t_{HFSI}	TFSx/RFSx Hold After TSCLKx/RSCLKx[1]	–2.0		ns
t_{SDRI}	Receive Data Setup Before RSCLKx[1]	6.0		ns
t_{HDRI}	Receive Data Hold After RSCLKx[1]	0.0		ns
Switching Characteristics				
t_{DFSI}	TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[2]		3.0	ns
t_{HOFSI}	TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[2]	–1.0		ns
t_{DDTI}	Transmit Data Delay After TSCLKx[2]		3.0	ns
t_{HDTI}	Transmit Data Hold After TSCLKx[2]	–2.0		ns
t_{SCLKIW}	TSCLKx/RSCLKx Width	4.5		ns

[1] Referenced to sample edge.

[2] Referenced to drive edge.

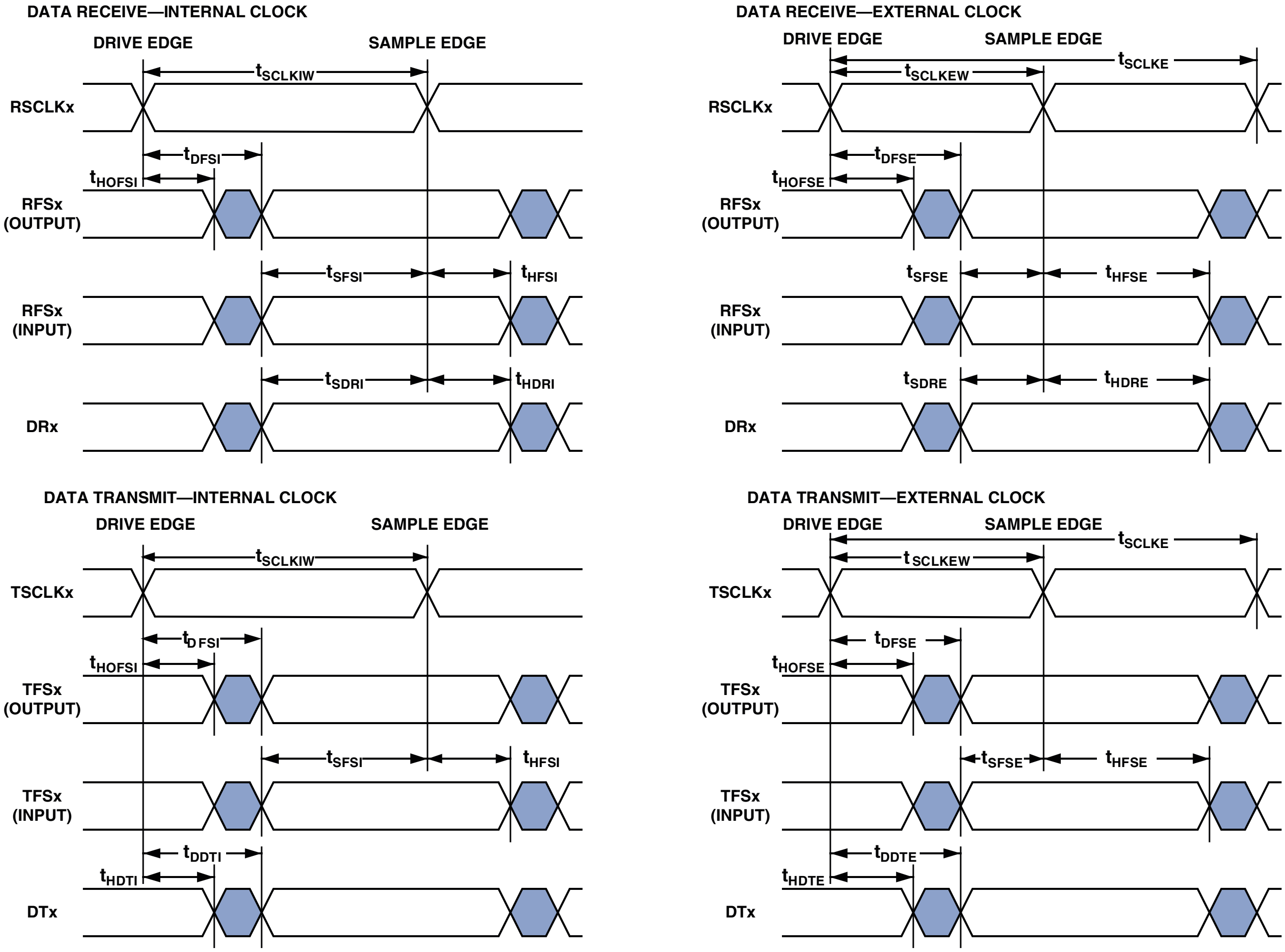


Figure 19. Serial Ports

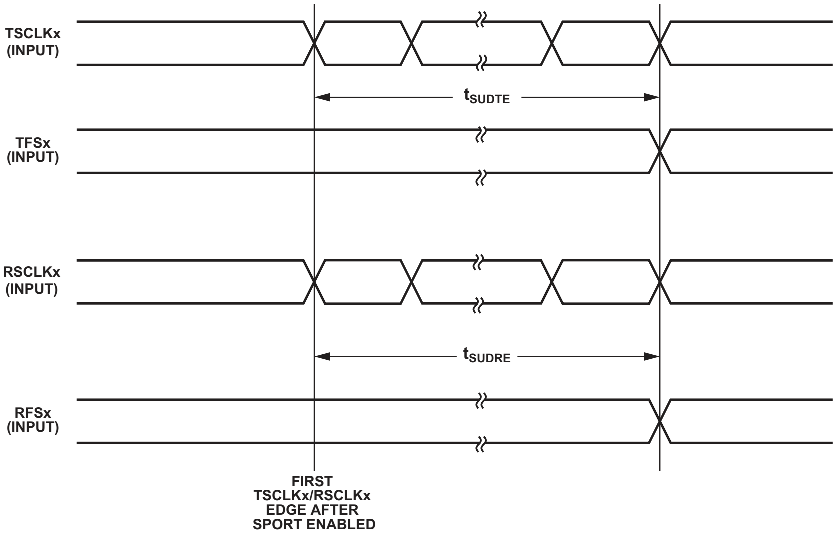


Figure 20. Serial Port Start Up with External Clock and Frame Sync

Table 24. Serial Ports—Enable and Three-State

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DTENE}	Data Enable Delay from External TSCLKx[1]	0		ns
t_{DDTTE}	Data Disable Delay from External TSCLKx[1, 2, 3]		10.0	ns
t_{DTENI}	Data Enable Delay from Internal TSCLKx[1]	−2.0		ns
t_{DDTTI}	Data Disable Delay from Internal TSCLKx[1, 2, 3]		3.0	ns

[1] Referenced to drive edge.

[2] Applicable to multichannel mode only.

[3] TSCLKx is tied to RSCLKx.

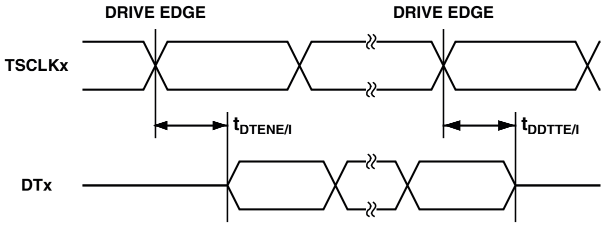


Figure 21. Enable and Three-State

Table 25. External Late Frame Sync

Parameter		Min	Max	Unit
Switching Characteristics				
$t_{DDTLFSE}$	Data Delay from Late External TFSx or External RFSx in multichannel mode with MFD = $0^{1, 2}$		10.0	ns
$t_{DTENLFS}$	Data Enable from Late FS or in multichannel mode with MFD = $0^{1, 2}$	0		ns

[1] In multichannel mode, TFSx enable and TFSx valid follow $t_{DTENLFS}$ and $t_{DDTLFSE}$.

[2] If external RFSx/TFSx setup to RSCLKx/TSCLKx > $t_{SCLKE}/2$, then $t_{DDTTE/I}$ and $t_{DTENE/I}$ apply; otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

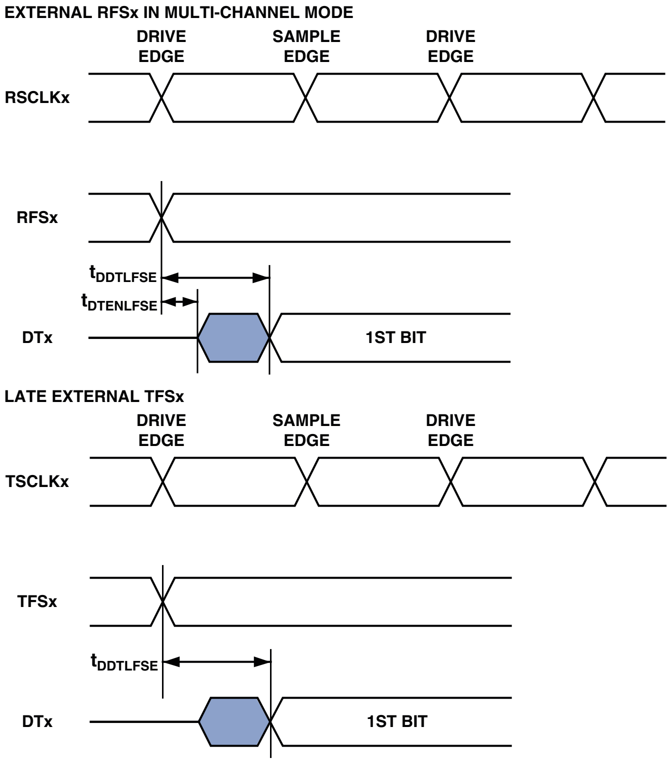


Figure 22. External Late Frame Sync

Serial Peripheral Interface (SPI) Port—Master Timing

Table 26 and Figure 23 describe SPI port master operations.

Table 26. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSPIDM}	Data Input Valid to SCK Edge (Data Input Setup)	7.5		ns
t_{HSPIDM}	SCK Sampling Edge to Data Input Invalid	–1.5		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPISELx}}$ Low to First SCK Edge	$2 \times t_{SCLK} - 1.5$		ns
t_{SPICHM}	Serial Clock High Period	$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLM}	Serial Clock Low Period	$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLK}	Serial Clock Period	$4 \times t_{SCLK} - 1.5$		ns
t_{HDSM}	Last SCK Edge to $\overline{\text{SPISELx}}$ High	$2 \times t_{SCLK} - 1.5$		ns
t_{SPITDM}	Sequential Transfer Delay	$2 \times t_{SCLK} - 1.5$		ns
$t_{DDSPIDM}$	SCK Edge to Data Out Valid (Data Out Delay)	0	6	ns
$t_{HDSPIDM}$	SCK Edge to Data Out Invalid (Data Out Hold)	–1.0	+4.0	ns

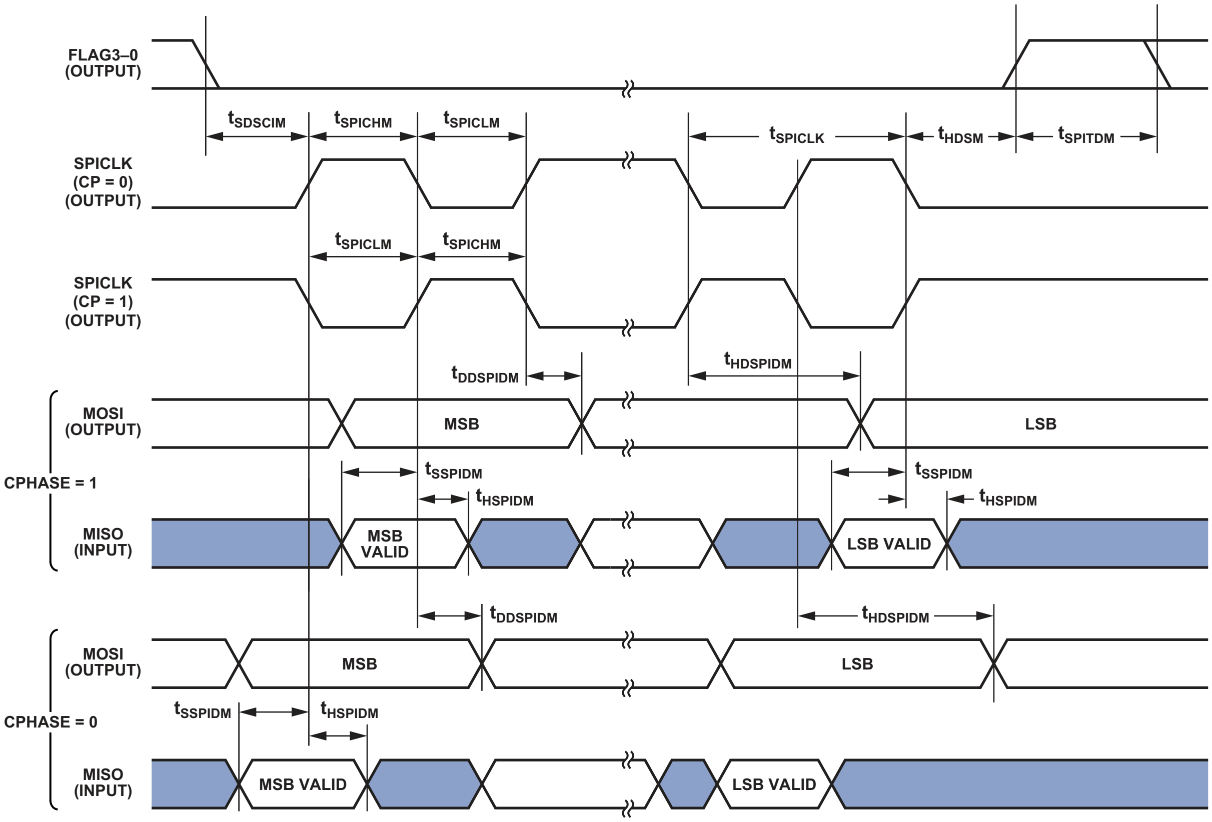


Figure 23. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 27 and Figure 24 describe SPI port slave operations.

Table 27. Serial Peripheral Interface (SPI) Port—Slave Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SPICHS}	Serial Clock High Period	$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLS}	Serial Clock Low Period	$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLK}	Serial Clock Period	$4 \times t_{SCLK}$		ns
t_{HDS}	Last SCK Edge to $\overline{\text{SPISS}}$ Not Asserted	$2 \times t_{SCLK} - 1.5$		ns
t_{SPITDS}	Sequential Transfer Delay	$2 \times t_{SCLK} - 1.5$		ns
t_{SDSCI}	$\overline{\text{SPISS}}$ Assertion to First SCK Edge	$2 \times t_{SCLK} - 1.5$		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}	$\overline{\text{SPISS}}$ Assertion to Data Out Active	0	8	ns
t_{DSDHI}	$\overline{\text{SPISS}}$ Deassertion to Data High Impedance	0	8	ns
t_{DDSPID}	SCK Edge to Data Out Valid (Data Out Delay)	0	10	ns
t_{HDSPID}	SCK Edge to Data Out Invalid (Data Out Hold)	0	10	ns

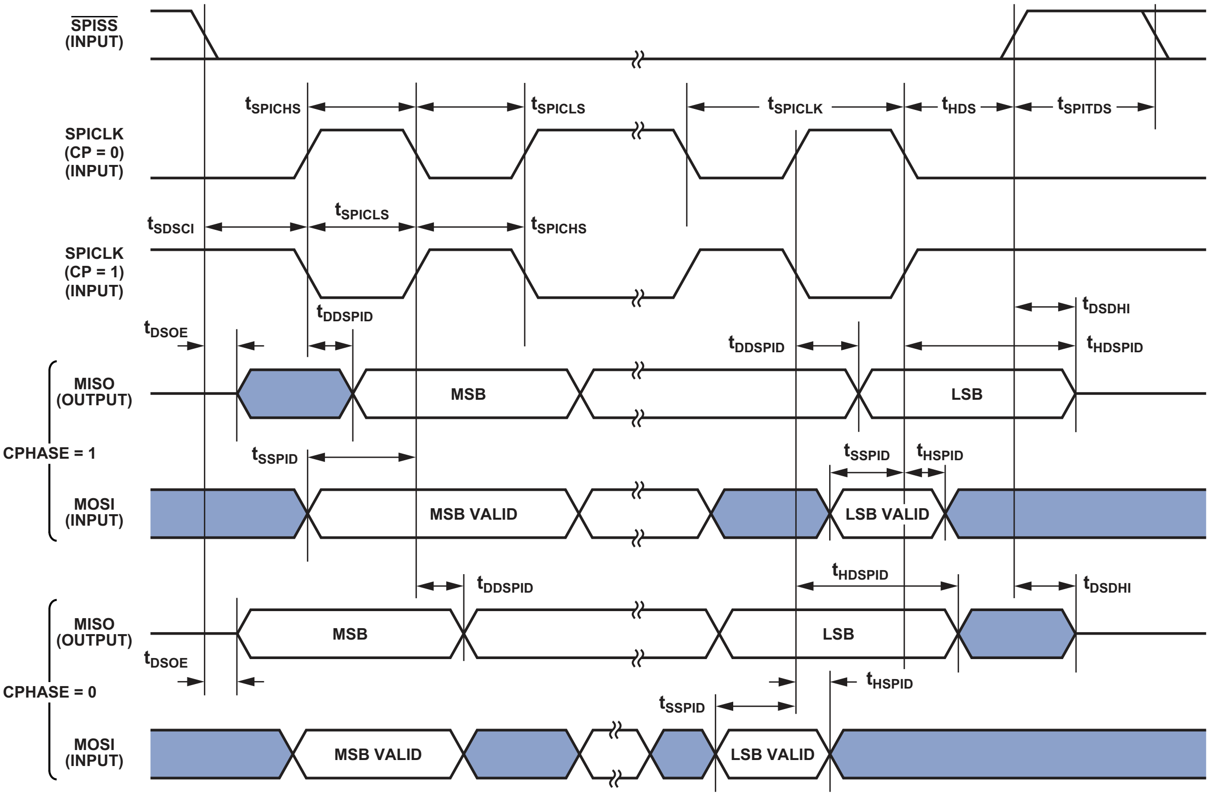


Figure 24. Serial Peripheral Interface (SPI) Port—Slave Timing

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

Figure 25 describes UART port receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 25, there is some latency between the generation internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

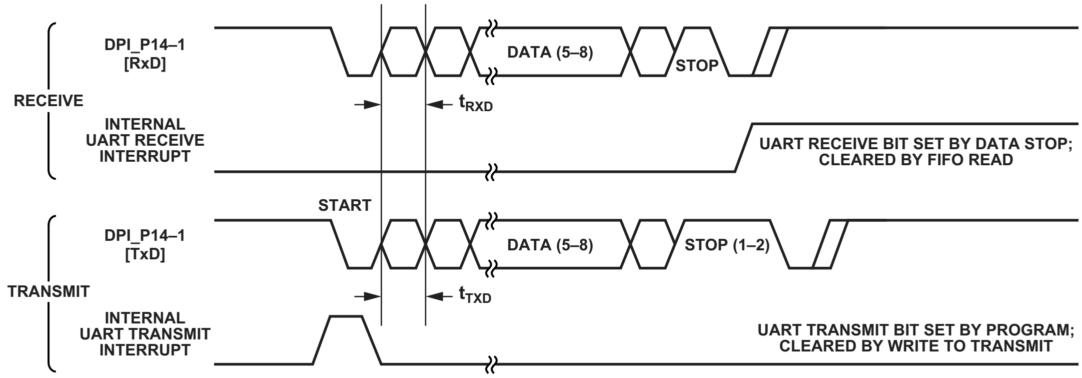


Figure 25. UART Port—Receive and Transmit Timing

Programmable Flags Cycle Timing

Table 28 and Figure 26 describe programmable flag operations.

Table 28. Programmable Flags Cycle Timing

Parameter		Min	Max	Unit
Timing Requirement				
t_{WFI}	Flag Input Pulse Width	$t_{SCLK} + 1$		ns
Switching Characteristic				
t_{DFO}	Flag Output Delay from CLKOUT Low		6	ns

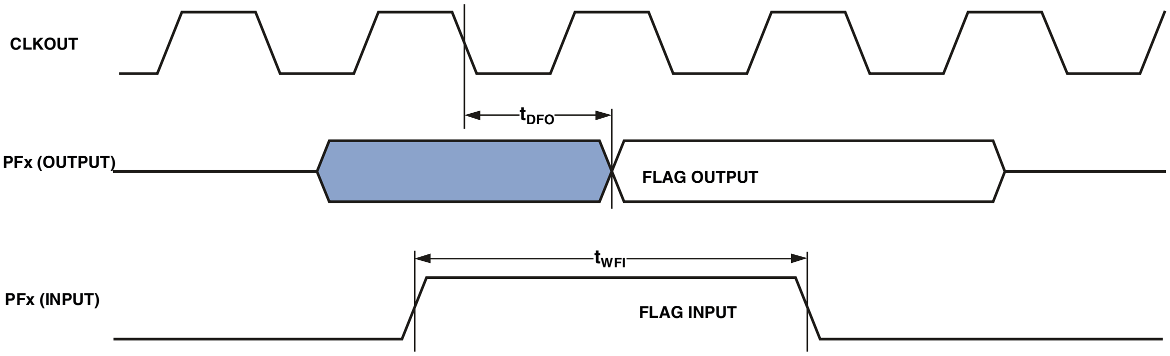


Figure 26. Programmable Flags Cycle Timing

Timer Clock Timing

Table 29 and Figure 27 describe timer clock timing.

Table 29. Timer Clock Timing

Parameter		Min	Max	Unit
Switching Characteristic				
t_{TODP}	Timer Output Update Delay After PPI_CLK High		12	ns

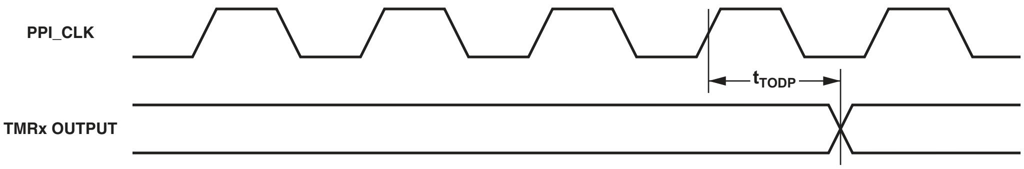


Figure 27. Timer Clock Timing

Timer Cycle Timing

Table 30 and Figure 28 describe timer expired operations. The input signal is asynchronous in "width capture mode" and "external clock mode" and has an absolute maximum input frequency of $f_{SCLK}/2$ MHz.

Table 30. Timer Cycle Timing

Parameter		Min	Max	Unit
Timing Characteristics				
t_{WL}	Timer Pulse Width Low[1]	$1 \times t_{SCLK}$		ns
t_{WH}	Timer Pulse Width High[1]	$1 \times t_{SCLK}$		ns
t_{TIS}	Timer Input Setup Time Before CLKOUT Low[2]	6.5		ns
t_{TIH}	Timer Input Hold Time After CLKOUT Low[2]	1.5		ns
Switching Characteristics				
t_{HTO}	Timer Pulse Width Output	$1 \times t_{SCLK}$	$(2^{32}-1) \times t_{SCLK}$	ns
t_{TOD}	Timer Output Update Delay After CLKOUT High		6.0	ns

[1] The minimum pulse widths apply for TMRx input pins in width capture and external clock modes. They also apply to the PF1 or PPI_CLK input pins in PWM output mode.
[2] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize programmable flag inputs.

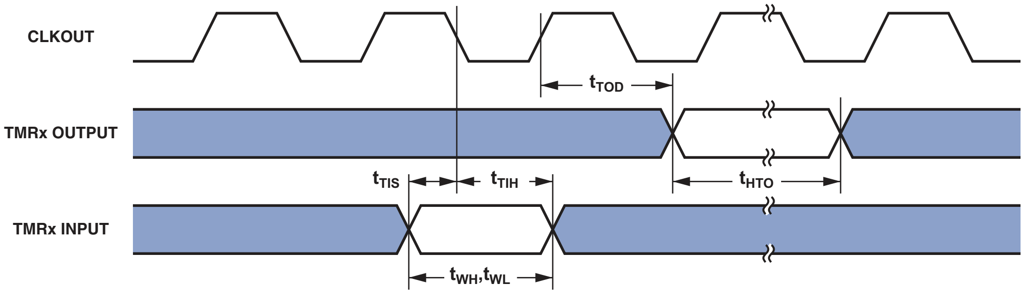


Figure 28. Timer PWM_OUT Cycle Timing

JTAG Test and Emulation Port Timing

Table 31, Figure 29, and Figure 30 describe JTAG port operations.

Table 31. JTAG Port Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{TCK}	TCK Period	20		ns
t_{STAP}	TDI, TMS Setup Before TCK High	4		ns
t_{HTAP}	TDI, TMS Hold After TCK High	4		ns
t_{SSYS}	System Inputs Setup Before TCK High[1]	4		ns
t_{HSYS}	System Inputs Hold After TCK High[1]	5		ns
t_{TRSTW}	$\overline{\text{TRST}}$ Pulse Width[2]	$4 \times t_{TCK}$		ns
Switching Characteristics				
t_{DTDO}	TDO Delay from TCK Low		10	ns
t_{DSYS}	System Outputs Delay After TCK Low[3]	0	12	ns

[1] System Inputs= DATA31–0, ARDY, PF47–0, PPI0CLK, PPI1CLK, RSCLK0–1, RFS0–1, DR0PRI, DR0SEC, TSCLK0–1, TFS0–1, DR1PRI, DR1SEC, MOSI, MISO, SCK, RX, $\overline{\text{RESET}}$, NMI0, NMI1, BMODE1–0, $\overline{\text{BR}}$, and PPIxD7–0.

[2] 50 MHz maximum

[3] System Outputs = DATA31–0, ADDR25–2, $\overline{\text{ABE3–0}}$, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{AMS3–0}}$, $\overline{\text{SRAS}}$, $\overline{\text{SCAS}}$, $\overline{\text{SWE}}$, SCKE, CLKOUT, SA10, $\overline{\text{SMS3–0}}$, PF47–0, RSCLK0–1, RFS0–1, TSCLK0–1, TFS0–1, DT0PRI, DT0SEC, DT1PRI, DT1SEC, MOSI, MISO, SCK, TX, $\overline{\text{BG}}$, $\overline{\text{BGH}}$, and PPIxD7–0.

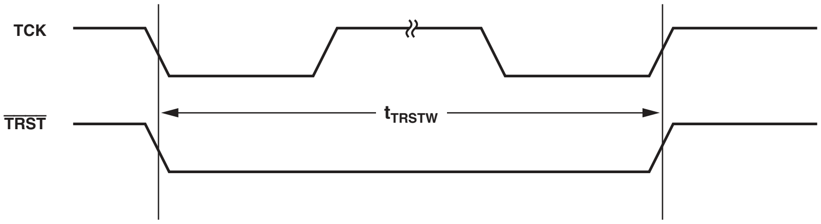


Figure 29. JTAG Port Reset Timing

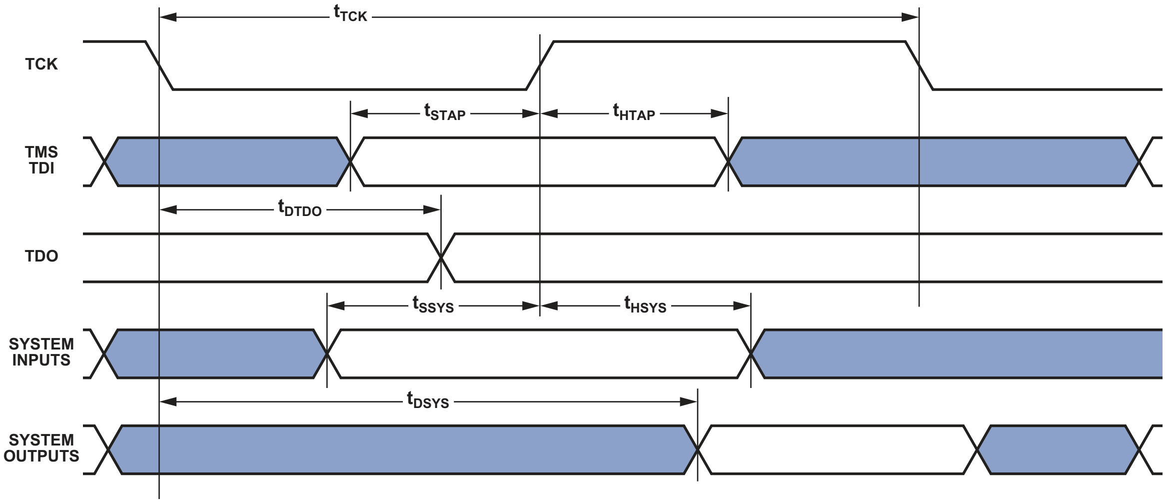


Figure 30. JTAG Port Timing

OUTPUT DRIVE CURRENTS

Figure 31 through Figure 38 show typical current voltage characteristics for the output drivers of the ADSP-BF561 processor. The curves represent the current drive capability of the output drivers as a function of output voltage. Refer to Table 8 to identify the driver type for a pin.

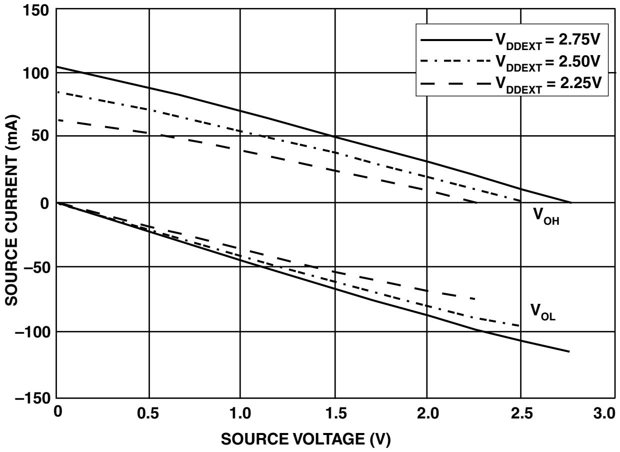


Figure 31. Drive Current A (Low V_{DDEXT})

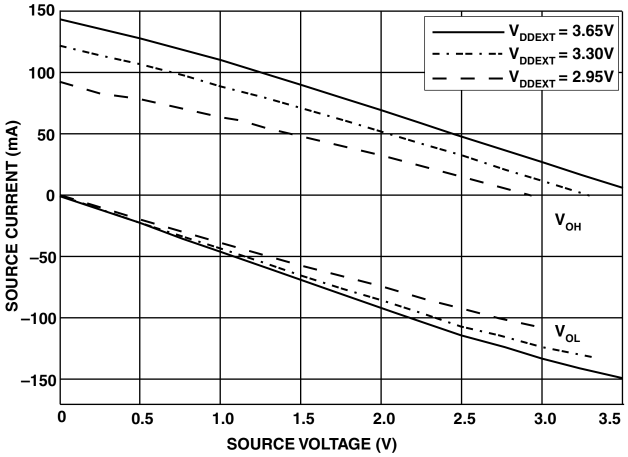


Figure 32. Drive Current A (High V_{DDEXT})

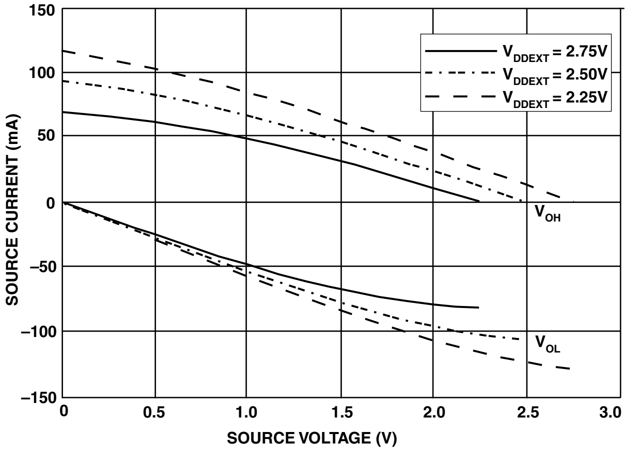


Figure 33. Drive Current B (Low V_{DDEXT})

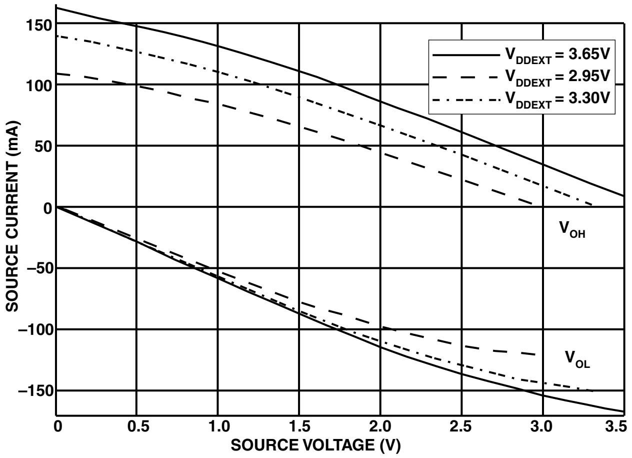


Figure 34. Drive Current B (High V_{DDEXT})

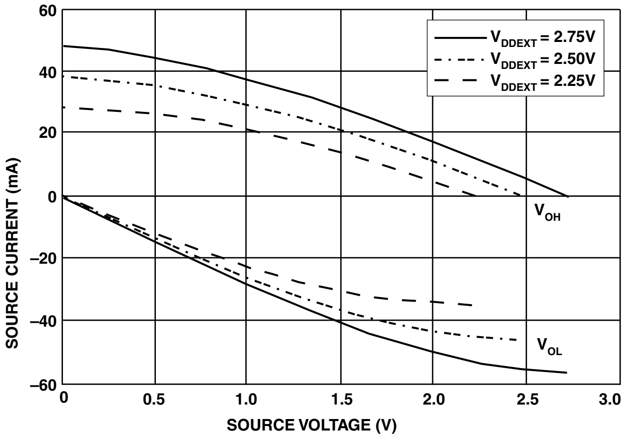


Figure 35. Drive Current C (Low V_{DDEXT})

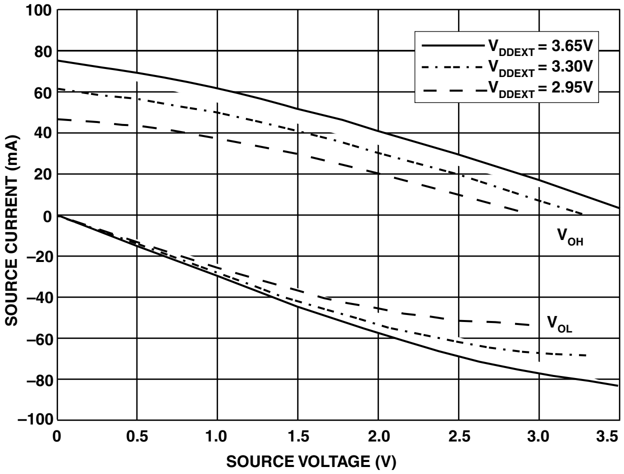


Figure 36. Drive Current C (High V_{DDEXT})

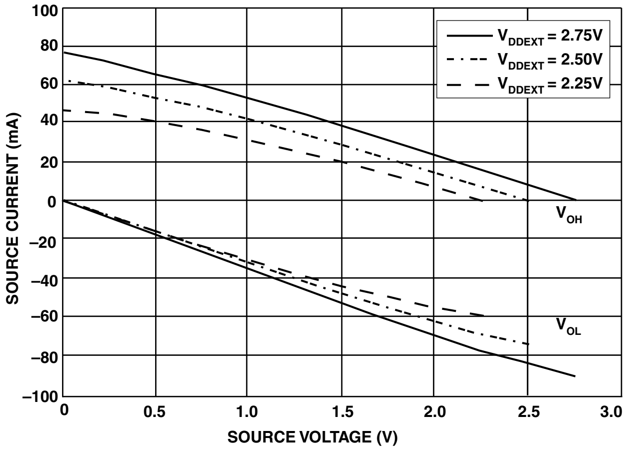


Figure 37. Drive Current D (Low V_{DDEXT})

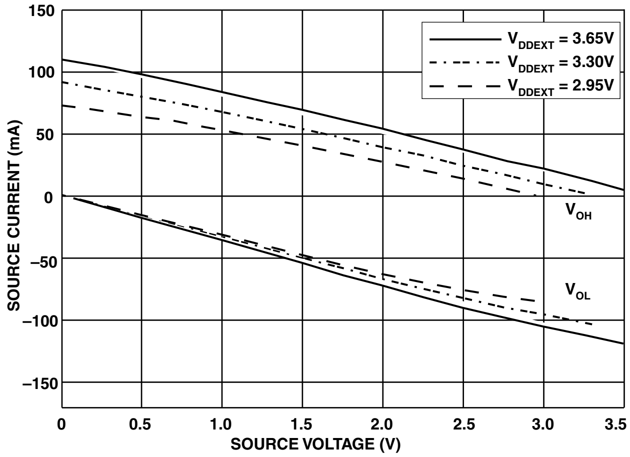


Figure 38. Drive Current D (High V_{DDEXT})

POWER DISSIPATION

Many operating conditions can affect power dissipation. System designers should refer to *Estimating Power for ADSP-BF561 Blackfin Processors (EE-293)* on the Analog Devices website (www.analog.com)—use site search on “EE-293.” This document provides detailed information for optimizing your design for lowest power.

See the *ADSP-BF561 Blackfin Processor Hardware Reference Manual* for definitions of the various operating modes and for instructions on how to minimize system power.

TEST CONDITIONS

All timing parameters appearing in this data sheet were measured under the conditions described in this section. Figure 39 shows the measurement point for ac measurements (except output enable/disable). The measurement point V_{MEAS} is 1.5 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V.

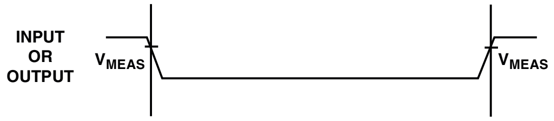


Figure 39. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

Output Enable Time Measurement

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 on the right side of Figure 40.

The time $t_{ENA\_MEASURED}$ is the interval, from when the reference signal switches, to when the output voltage reaches V_{TRIP}(high) or V_{TRIP}(low). V_{TRIP}(high) is 2.0 V and V_{TRIP}(low) is 1.0 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V. Time t_{TRIP} is the interval from when the output starts driving to when the output reaches the V_{TRIP}(high) or V_{TRIP}(low) 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 Measurement

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 output disable time t_{DIS} is the difference between $t_{DIS\_MEASURED}$ and t_{DECAY} as shown on the left side of Figure 40.

$$t_{DIS} = t_{DIS\_MEASURED} - t_{DECAY}$$

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 time t_{DECAY} is calculated with test loads C_L and I_L, and with ΔV equal to 0.5 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V.

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.

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-BF561 processor's output voltage and the input threshold for the device requiring the hold time. 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 various output disable times as specified in the Timing Specifications (for example t_{DSDAT} for an SDRAM write cycle as shown in SDRAM Interface Timing).

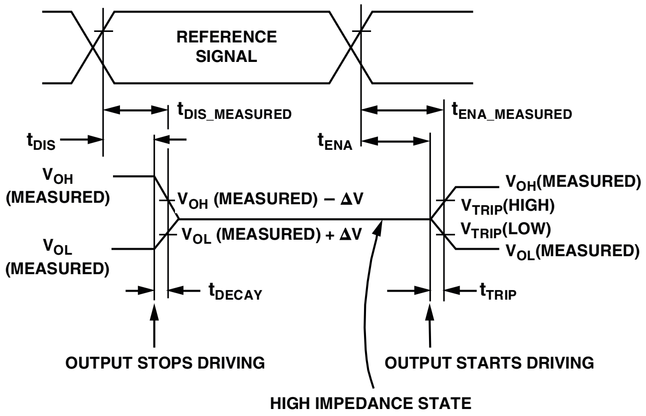


Figure 40. Output Enable/Disable

Capacitive Loading

Output delays and holds are based on standard capacitive loads: 30 pF on all pins (see Figure 41). V_{LOAD} is 1.5 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V. Figure 42 through Figure 49 show how output rise time varies with capacitance. The delay and hold specifications given should be derated by a factor derived from these figures. The graphs in these figures may not be linear outside the ranges shown.

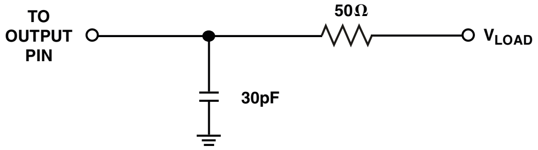


Figure 41. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

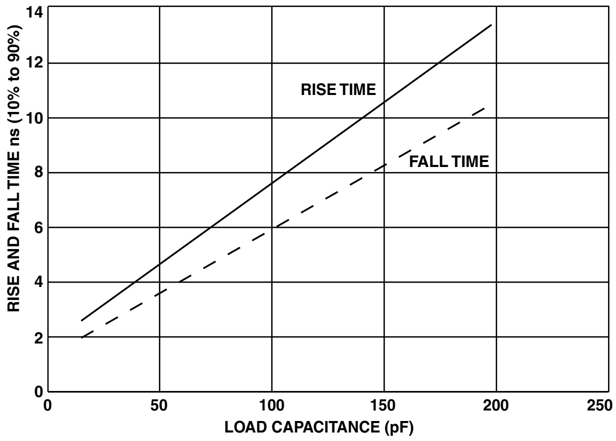


Figure 42. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver A at V_{DDEXT} (min)

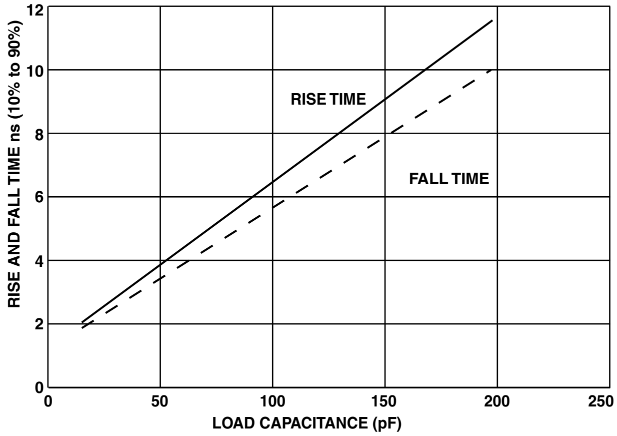


Figure 43. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver A at V_{DDEXT} (max)

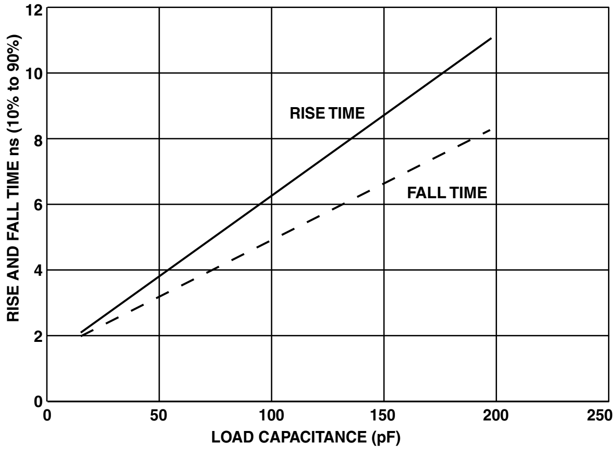


Figure 44. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver B at V_{DDEXT} (min)

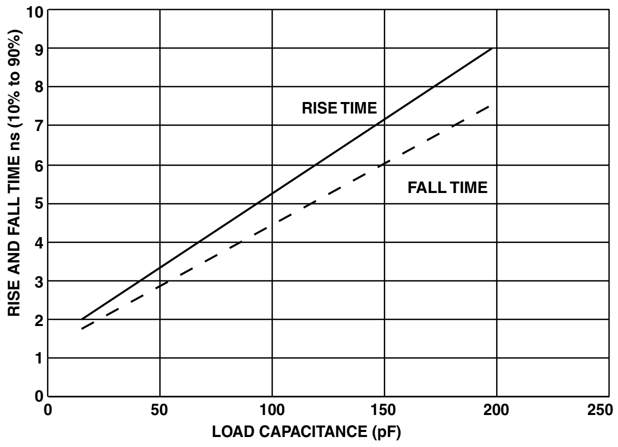


Figure 45. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver B at V_{DDEXT} (max)

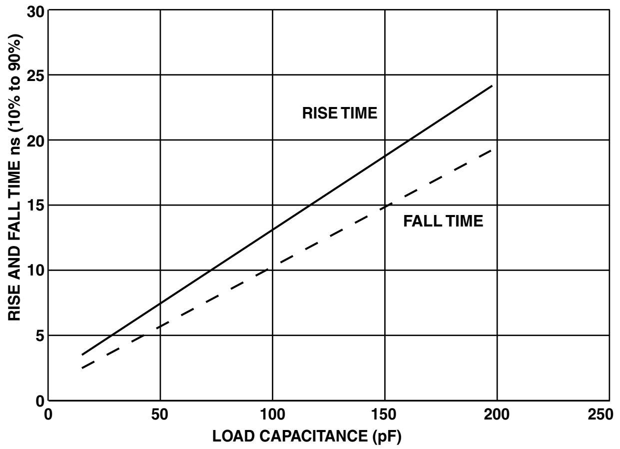


Figure 46. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at V_{DDEXT} (min)

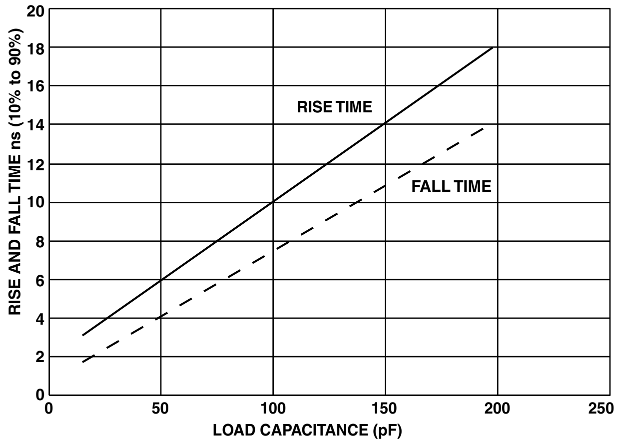


Figure 47. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at V_{DDEXT} (max)

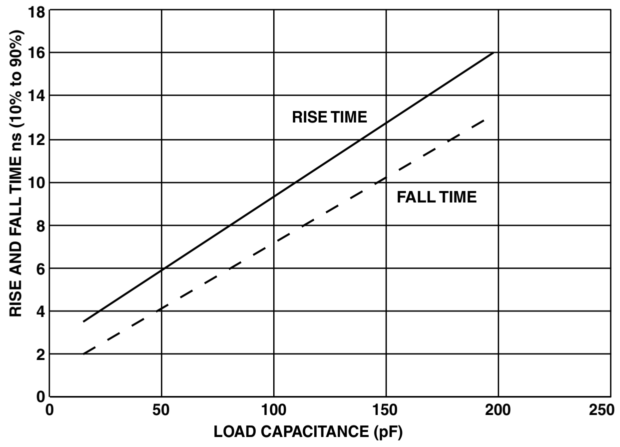


Figure 48. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D at V_{DDEXT} (min)

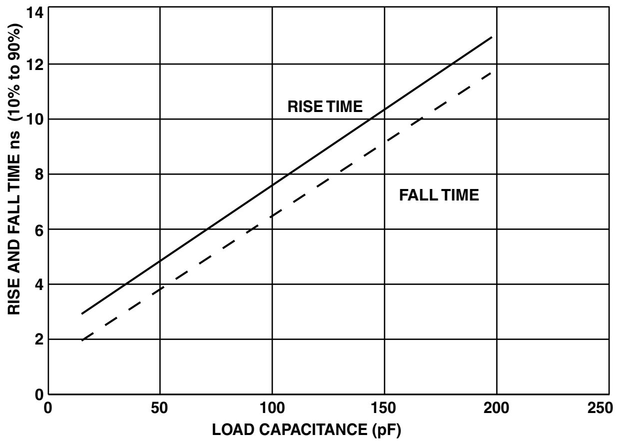


Figure 49. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D at V_{DDEXT} (max)

ENVIRONMENTAL CONDITIONS

To determine the junction temperature on the application printed circuit board use:

$$T_J = T_{CASE} + (\Psi_{JT} \times P_D)$$

where:

T_J = junction temperature (°C).

T_{CASE} = case temperature (°C) measured by customer at top center of package.

Ψ_{JT} = from Table 32 through Table 34.

P_D = power dissipation (see Power Dissipation for the method to calculate P_D).

Values of θ_{JA} are provided for package comparison and printed circuit board design considerations. θ_{JA} can be used for a first order approximation of T_J by the equation:

$$T_J = T_A + (\theta_{JA} \times P_D)$$

where:

T_A = ambient temperature (°C).

In Table 32 through Table 34, airflow measurements comply with JEDEC standards JESD51–2 and JESD51–6, and the junction-to-board measurement complies with JESD51–8. The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC test board.

Thermal resistance θ_{JA} in Table 32 through Table 34 is the figure of merit relating to performance of the package and board in a convective environment. θ_{JMA} represents the thermal resistance under two conditions of airflow. θ_{JB} represents the heat extracted from the periphery of the board. Ψ_{JT} represents the correlation between T_J and T_{CASE}. Values of θ_{JB} are provided for package comparison and printed circuit board design considerations.

Table 32. Thermal Characteristics for BC-256-4 (17 mm × 17 mm) Package

Parameter	Condition	Typical	Unit
θ_{JA}	0 Linear m/s Airflow	18.1	°C/W
θ_{JMA}	1 Linear m/s Airflow	15.9	°C/W
θ_{JMA}	2 Linear m/s Airflow	15.1	°C/W
θ_{JC}	Not Applicable	3.72	°C/W
Ψ_{JT}	0 Linear m/s Airflow	0.11	°C/W
Ψ_{JT}	1 Linear m/s Airflow	0.18	°C/W
Ψ_{JT}	2 Linear m/s Airflow	0.18	°C/W

Table 33. Thermal Characteristics for BC-256-1 (12 mm × 12 mm) Package

Parameter	Condition	Typical	Unit
θ_{JA}	0 Linear m/s Airflow	25.6	°C/W
θ_{JMA}	1 Linear m/s Airflow	22.4	°C/W
θ_{JMA}	2 Linear m/s Airflow	21.6	°C/W
θ_{JB}	Not Applicable	18.9	°C/W
θ_{JC}	Not Applicable	4.85	°C/W
Ψ_{JT}	0 Linear m/s Airflow	0.15	°C/W
Ψ_{JT}	1 Linear m/s Airflow	n/a	°C/W
Ψ_{JT}	2 Linear m/s Airflow	n/a	°C/W

Table 34. Thermal Characteristics for B-297 Package

Parameter	Condition	Typical	Unit
θ_{JA}	0 Linear m/s Airflow	20.6	°C/W
θ_{JMA}	1 Linear m/s Airflow	17.8	°C/W
θ_{JMA}	2 Linear m/s Airflow	17.4	°C/W
θ_{JB}	Not Applicable	16.3	°C/W
θ_{JC}	Not Applicable	7.15	°C/W
Ψ_{JT}	0 Linear m/s Airflow	0.37	°C/W
Ψ_{JT}	1 Linear m/s Airflow	n/a	°C/W
Ψ_{JT}	2 Linear m/s Airflow	n/a	°C/W

256-BALL CSP_BGA (17 mm) BALL ASSIGNMENT

Table 35 lists the 256-Ball CSP_BGA (17 mm × 17 mm) ball assignment by ball number. Table 36 lists the ball assignment alphabetically by signal.

Table 35. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A1	VDDEXT	C9	$\overline{\text{SMS3}}$	F1	CLKIN	H9	GND	L1	PPI0D3
A2	ADDR22	C10	$\overline{\text{SWE}}$	F2	PPI0D10	H10	GND	L2	PPI0D2
A3	ADDR18	C11	SA10	F3	$\overline{\text{RESET}}$	H11	GND	L3	PPI0D1
A4	ADDR14	C12	$\overline{\text{ABE0}}$	F4	BYPASS	H12	GND	L4	PPI0D0
A5	ADDR11	C13	ADDR07	F5	VDDEXT	H13	GND	L5	VDDEXT
A6	$\overline{\text{AMS3}}$	C14	ADDR04	F6	VDDEXT	H14	DATA21	L6	VDDEXT
A7	$\overline{\text{AMS0}}$	C15	DATA0	F7	VDDEXT	H15	DATA19	L7	VDDEXT
A8	ARDY	C16	DATA05	F8	GND	H16	DATA23	L8	VDDEXT
A9	$\overline{\text{SMS2}}$	D1	PPI0D15	F9	GND	J1	VROUT1	L9	GND
A10	SCLK0	D2	PPI0SYNC3	F10	VDDEXT	J2	PPI0D8	L10	VDDEXT
A11	SCLK1	D3	PPI0SYNC2	F11	VDDEXT	J3	PPI0D7	L11	VDDEXT
A12	$\overline{\text{ABE2}}$	D4	ADDR21	F12	VDDEXT	J4	PPI0D9	L12	VDDEXT
A13	$\overline{\text{ABE3}}$	D5	ADDR15	F13	DATA11	J5	GND	L13	NC
A14	ADDR06	D6	ADDR09	F14	DATA08	J6	GND	L14	DT0PRI
A15	ADDR03	D7	$\overline{\text{AWE}}$	F15	DATA10	J7	GND	L15	DATA31
A16	VDDEXT	D8	$\overline{\text{SMS0}}$	F16	DATA16	J8	GND	L16	DATA28
B1	ADDR24	D9	$\overline{\text{SRAS}}$	G1	XTAL	J9	GND	M1	PPI1SYNC2
B2	ADDR23	D10	$\overline{\text{SCAS}}$	G2	VDDEXT	J10	GND	M2	PPI1D15
B3	ADDR19	D11	$\overline{\text{BGH}}$	G3	VDDEXT	J11	GND	M3	PPI1D14
B4	ADDR17	D12	$\overline{\text{ABE1}}$	G4	GND	J12	VDDINT	M4	PPI1D9
B5	ADDR12	D13	DATA02	G5	GND	J13	VDDINT	M5	VDDINT
B6	ADDR10	D14	DATA01	G6	VDDEXT	J14	DATA20	M6	VDDINT
B7	$\overline{\text{AMS1}}$	D15	DATA03	G7	GND	J15	DATA22	M7	GND
B8	$\overline{\text{AOE}}$	D16	DATA07	G8	GND	J16	DATA24	M8	VDDINT
B9	$\overline{\text{SMS1}}$	E1	PPI0D11	G9	GND	K1	PPI0D6	M9	GND
B10	SCKE	E2	PPI0D13	G10	GND	K2	PPI0D5	M10	VDDINT
B11	$\overline{\text{BR}}$	E3	PPI0D12	G11	VDDEXT	K3	PPI0D4	M11	GND
B12	$\overline{\text{BG}}$	E4	PPI0D14	G12	VDDEXT	K4	PPI1SYNC3	M12	VDDINT
B13	ADDR08	E5	PPI1CLK	G13	DATA17	K5	VDDEXT	M13	RSCLK0
B14	ADDR05	E6	VDDINT	G14	DATA14	K6	VDDEXT	M14	DR0PRI
B15	ADDR02	E7	GND	G15	DATA15	K7	GND	M15	TSCLK0
B16	DATA04	E8	VDDINT	G16	DATA18	K8	GND	M16	DATA29
C1	PPI0SYNC1	E9	GND	H1	VROUT0	K9	GND	N1	PPI1SYNC1
C2	ADDR25	E10	VDDINT	H2	GND	K10	GND	N2	PPI1D10
C3	PPI0CLK	E11	GND	H3	GND	K11	VDDEXT	N3	PPI1D7
C4	ADDR20	E12	VDDINT	H4	VDDINT	K12	GND	N4	PPI1D5
C5	ADDR16	E13	DATA06	H5	VDDINT	K13	GND	N5	PF0
C6	ADDR13	E14	DATA13	H6	GND	K14	DATA26	N6	PF04
C7	$\overline{\text{AMS2}}$	E15	DATA09	H7	GND	K15	DATA25	N7	PF09
C8	$\overline{\text{ARE}}$	E16	DATA12	H8	GND	K16	DATA27	N8	PF12

Table 35. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Numerically by Ball Number) (Continued)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
N9	GND	P5	PF01	R1	PPI1D12	R13	RSCLK1	T9	TDO
N10	BMODE1	P6	PF06	R2	PPI1D11	R14	TSCLK1	T10	TDI
N11	BMODE0	P7	PF08	R3	PPI1D4	R15	NC	T11	$\overline{\text{EMU}}$
N12	RX	P8	PF15	R4	PPI1D1	R16	TFS0	T12	MISO
N13	DR1SEC	P9	NMI1	R5	PF02	T1	VDDEXT	T13	TX
N14	DT1SEC	P10	TMS	R6	PF07	T2	NC	T14	DR1PRI
N15	RFS0	P11	NMI0	R7	PF11	T3	PPI1D3	T15	DT1PRI
N16	DATA30	P12	SCK	R8	PF14	T4	PPI1D2	T16	VDDEXT
P1	PPI1D13	P13	RFS1	R9	TCK	T5	PF03		
P2	PPI1D8	P14	TFS1	R10	$\overline{\text{TRST}}$	T6	PF05		
P3	PPI1D6	P15	DR0SEC	R11	SLEEP	T7	PF10		
P4	PPI1D0	P16	DT0SEC	R12	MOSI	T8	PF13		

Table 36. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
$\overline{\text{ABE0}}$	C12	$\overline{\text{BR}}$	B11	DT0SEC	P16	GND	M9	PPI0D13	E2
$\overline{\text{ABE1}}$	D12	BYPASS	F4	DT1PRI	T15	GND	M11	PPI0D14	E4
$\overline{\text{ABE2}}$	A12	CLKIN	F1	DT1SEC	N14	GND	N9	PPI0D15	D1
$\overline{\text{ABE3}}$	A13	DATA0	C15	$\overline{\text{EMU}}$	T11	MISO	T12	PPI0SYNC1	C1
ADDR02	B15	DATA01	D14	GND	E7	MOSI	R12	PPI0SYNC2	D3
ADDR03	A15	DATA02	D13	GND	E9	NC	L13	PPI0SYNC3	D2
ADDR04	C14	DATA03	D15	GND	E11	NC	R15	PPI1CLK	E5
ADDR05	B14	DATA04	B16	GND	F8	NC	T2	PPI1D0	P4
ADDR06	A14	DATA05	C16	GND	F9	NMI0	P11	PPI1D1	R4
ADDR07	C13	DATA06	E13	GND	G4	NMI1	P9	PPI1D2	T4
ADDR08	B13	DATA07	D16	GND	G5	PF0	N5	PPI1D3	T3
ADDR09	D6	DATA08	F14	GND	G7	PF01	P5	PPI1D4	R3
ADDR10	B6	DATA09	E15	GND	G8	PF02	R5	PPI1D5	N4
ADDR11	A5	DATA10	F15	GND	G9	PF03	T5	PPI1D6	P3
ADDR12	B5	DATA11	F13	GND	G10	PF04	N6	PPI1D7	N3
ADDR13	C6	DATA12	E16	GND	H2	PF05	T6	PPI1D8	P2
ADDR14	A4	DATA13	E14	GND	H3	PF06	P6	PPI1D9	M4
ADDR15	D5	DATA14	G14	GND	H6	PF07	R6	PPI1D10	N2
ADDR16	C5	DATA15	G15	GND	H7	PF08	P7	PPI1D11	R2
ADDR17	B4	DATA16	F16	GND	H8	PF09	N7	PPI1D12	R1
ADDR18	A3	DATA17	G13	GND	H9	PF10	T7	PPI1D13	P1
ADDR19	B3	DATA18	G16	GND	H10	PF11	R7	PPI1D14	M3
ADDR20	C4	DATA19	H15	GND	H11	PF12	N8	PPI1D15	M2
ADDR21	D4	DATA20	J14	GND	H12	PF13	T8	PPI1SYNC1	N1
ADDR22	A2	DATA21	H14	GND	H13	PF14	R8	PPI1SYNC2	M1
ADDR23	B2	DATA22	J15	GND	J5	PF15	P8	PPI1SYNC3	K4
ADDR24	B1	DATA23	H16	GND	J6	PPI0CLK	C3	$\overline{\text{RESET}}$	F3
ADDR25	C2	DATA24	J16	GND	J7	PPI0D0	L4	RFS0	N15
$\overline{\text{AMS0}}$	A7	DATA25	K15	GND	J8	PPI0D1	L3	RFS1	P13
$\overline{\text{AMS1}}$	B7	DATA26	K14	GND	J9	PPI0D2	L2	RSCLK0	M13
$\overline{\text{AMS2}}$	C7	DATA27	K16	GND	J10	PPI0D3	L1	RSCLK1	R13
$\overline{\text{AMS3}}$	A6	DATA28	L16	GND	J11	PPI0D4	K3	RX	N12
$\overline{\text{AOE}}$	B8	DATA29	M16	GND	K7	PPI0D5	K2	SA10	C11
ARDY	A8	DATA30	N16	GND	K8	PPI0D6	K1	$\overline{\text{SCAS}}$	D10
$\overline{\text{ARE}}$	C8	DATA31	L15	GND	K9	PPI0D7	J3	SCK	P12
$\overline{\text{AWE}}$	D7	DR0PRI	M14	GND	K10	PPI0D8	J2	SCKE	B10
$\overline{\text{BG}}$	B12	DR0SEC	P15	GND	K12	PPI0D9	J4	SCLK0	A10
$\overline{\text{BGH}}$	D11	DR1PRI	T14	GND	K13	PPI0D10	F2	SCLK1	A11
BMODE0	N11	DR1SEC	N13	GND	L9	PPI0D11	E1	SLEEP	R11
BMODE1	N10	DT0PRI	L14	GND	M7	PPI0D12	E3	$\overline{\text{SMS0}}$	D8

Table 36. 256-Ball CSP_BGA (17 mm × 17 mm) Ball Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
$\overline{\text{SMS1}}$	B9	TSCLK0	M15	VDDEXT	G3	VDDEXT	L11	VDDINT	M5
$\overline{\text{SMS2}}$	A9	TSCLK1	R14	VDDEXT	G6	VDDEXT	L12	VDDINT	M6
$\overline{\text{SMS3}}$	C9	TX	T13	VDDEXT	G11	VDDEXT	T1	VDDINT	M8
$\overline{\text{SRAS}}$	D9	VDDEXT	A1	VDDEXT	G12	VDDEXT	T16	VDDINT	M10
$\overline{\text{SWE}}$	C10	VDDEXT	A16	VDDEXT	K5	VDDINT	E6	VDDINT	M12
TCK	R9	VDDEXT	F5	VDDEXT	K6	VDDINT	E8	VROUT0	H1
TDI	T10	VDDEXT	F6	VDDEXT	K11	VDDINT	E10	VROUT1	J1
TDO	T9	VDDEXT	F7	VDDEXT	L5	VDDINT	E12	XTAL	G1
TFS0	R16	VDDEXT	F10	VDDEXT	L6	VDDINT	H4		
TFS1	P14	VDDEXT	F11	VDDEXT	L7	VDDINT	H5		
TMS	P10	VDDEXT	F12	VDDEXT	L8	VDDINT	J12		
$\overline{\text{TRST}}$	R10	VDDEXT	G2	VDDEXT	L10	VDDINT	J13		

Figure 50 lists the top view of the 256-Ball CSP_BGA (17 mm × 17 mm) ball configuration. Figure 51 lists the bottom view.

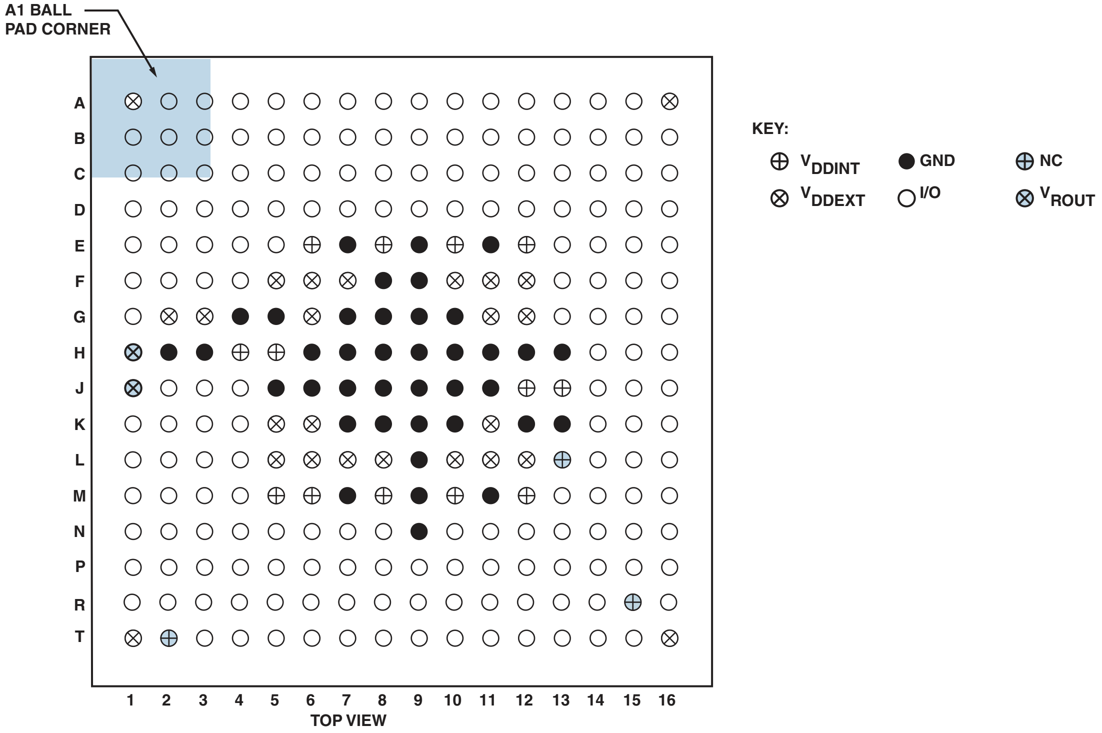


Figure 50. 256-Ball CSP_BGA Ball Configuration (Top View)

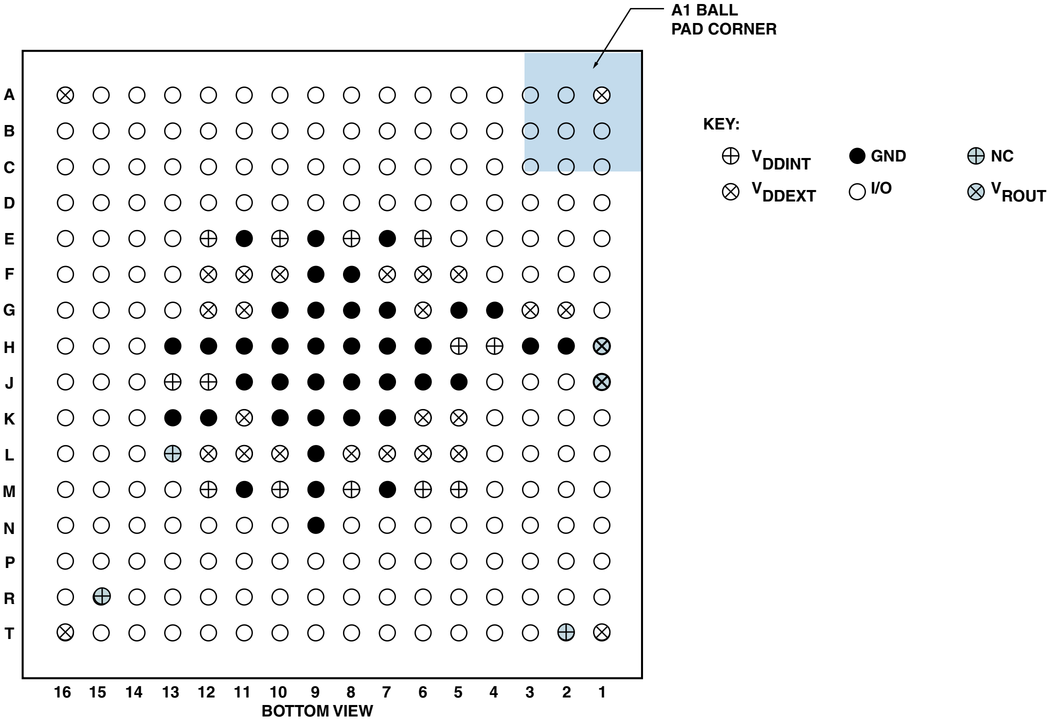


Figure 51. 256-Ball CSP_BGA Ball Configuration (Bottom View)

256-BALL CSP_BGA (12 mm) BALL ASSIGNMENT

Table 37 lists the 256-Ball CSP_BGA (12 mm × 12 mm) ball assignment by ball number. Table 38 lists the ball assignment alphabetically by signal.

Table 37. 256-Ball CSP_BGA (12 mm × 12 mm) Ball Assignment (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A01	VDDEXT	C09	$\overline{\text{SMS2}}$	F01	CLKIN	H09	GND	L01	PPI0D0
A02	ADDR24	C10	$\overline{\text{SRAS}}$	F02	VDDEXT	H10	GND	L02	PPI1SYNC2
A03	ADDR20	C11	GND	F03	$\overline{\text{RESET}}$	H11	VDDINT	L03	GND
A04	VDDEXT	C12	$\overline{\text{BGH}}$	F04	PPI0D10	H12	DATA16	L04	PPI1SYNC3
A05	ADDR14	C13	GND	F05	ADDR21	H13	DATA18	L05	VDDEXT
A06	ADDR10	C14	ADDR07	F06	ADDR17	H14	DATA20	L06	PPI1D11
A07	$\overline{\text{AMS3}}$	C15	DATA1	F07	VDDINT	H15	DATA17	L07	GND
A08	$\overline{\text{AWE}}$	C16	DATA3	F08	GND	H16	DATA19	L08	VDDINT
A09	VDDEXT	D01	PPI0D13	F09	VDDINT	J01	VROUT0	L09	GND
A10	$\overline{\text{SMS3}}$	D02	PPI0D15	F10	GND	J02	VROUT1	L10	VDDEXT
A11	SCLK0	D03	PPI0SYNC3	F11	ADDR08	J03	PPI0D2	L11	GND
A12	SCLK1	D04	ADDR23	F12	DATA10	J04	PPI0D3	L12	DR0PRI
A13	$\overline{\text{BG}}$	D05	GND	F13	DATA8	J05	PPI0D1	L13	TFS0
A14	$\overline{\text{ABE2}}$	D06	GND	F14	DATA12	J06	VDDEXT	L14	GND
A15	$\overline{\text{ABE3}}$	D07	ADDR09	F15	DATA9	J07	GND	L15	DATA27
A16	VDDEXT	D08	GND	F16	DATA11	J08	VDDINT	L16	DATA29
B01	PPI1CLK	D09	ARDY	G01	XTAL	J09	VDDINT	M01	PPI1D15
B02	ADDR22	D10	$\overline{\text{SCAS}}$	G02	GND	J10	VDDINT	M02	PPI1D13
B03	ADDR18	D11	SA10	G03	VDDEXT	J11	GND	M03	PPI1D9
B04	ADDR16	D12	VDDEXT	G04	BYPASS	J12	DATA30	M04	GND
B05	ADDR12	D13	ADDR02	G05	PPI0D14	J13	DATA22	M05	NC
B06	VDDEXT	D14	GND	G06	GND	J14	GND	M06	PF3
B07	$\overline{\text{AMS1}}$	D15	DATA5	G07	GND	J15	DATA21	M07	PF7
B08	$\overline{\text{ARE}}$	D16	DATA6	G08	GND	J16	DATA23	M08	VDDINT
B09	$\overline{\text{SMS1}}$	E01	GND	G09	VDDINT	K01	PPI0D6	M09	GND
B10	SCKE	E02	PPI0D11	G10	ADDR05	K02	PPI0D4	M10	BMODE0
B11	VDDEXT	E03	PPI0D12	G11	ADDR03	K03	PPI0D8	M11	SCK
B12	$\overline{\text{BR}}$	E04	PPI0SYNC1	G12	DATA15	K04	PPI1SYNC1	M12	DR1PRI
B13	$\overline{\text{ABE1}}$	E05	ADDR15	G13	DATA14	K05	PPI1D14	M13	NC
B14	ADDR06	E06	ADDR13	G14	GND	K06	VDDEXT	M14	VDDEXT
B15	ADDR04	E07	$\overline{\text{AMS2}}$	G15	DATA13	K07	GND	M15	DATA31
B16	DATA0	E08	VDDINT	G16	VDDEXT	K08	VDDINT	M16	DT0PRI
C01	PPI0SYNC2	E09	$\overline{\text{SMS0}}$	H01	GND	K09	GND	N01	PPI1D12
C02	PPI0CLK	E10	$\overline{\text{SWE}}$	H02	GND	K10	GND	N02	PPI1D10
C03	ADDR25	E11	$\overline{\text{ABE0}}$	H03	PPI0D9	K11	VDDINT	N03	PPI1D3
C04	ADDR19	E12	DATA2	H04	PPI0D7	K12	DATA28	N04	PPI1D1
C05	GND	E13	GND	H05	PPI0D5	K13	DATA26	N05	PF1
C06	ADDR11	E14	DATA4	H06	VDDINT	K14	DATA24	N06	PF9
C07	$\overline{\text{AOE}}$	E15	DATA7	H07	VDDINT	K15	DATA25	N07	GND
C08	$\overline{\text{AMS0}}$	E16	VDDEXT	H08	GND	K16	VDDEXT	N08	PF13

Table 37. 256-Ball CSP_BGA (12 mm × 12 mm) Ball Assignment (Numerically by Ball Number) (Continued)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
N09	TDO	P05	GND	R01	PPI1D7	R13	TX/PF26	T09	TCK
N10	BMODE1	P06	PF5	R02	PPI1D6	R14	TSCLK1	T10	TMS
N11	MOSI	P07	PF11	R03	PPI1D2	R15	DT1PRI	T11	SLEEP
N12	GND	P08	PF15	R04	PPI1D0	R16	RFS0	T12	VDDEXT
N13	RFS1	P09	GND	R05	PF4	T01	VDDEXT	T13	RX/PF27
N14	GND	P10	$\overline{\text{TRST}}$	R06	PF8	T02	PPI1D4	T14	DR1SEC
N15	DT0SEC	P11	NMI0	R07	PF10	T03	VDDEXT	T15	DT1SEC
N16	TSCLK0	P12	GND	R08	PF14	T04	PF2	T16	VDDEXT
P01	PPI1D8	P13	RSCLK1	R09	NMI1	T05	PF6		
P02	GND	P14	TFS1	R10	TDI	T06	VDDEXT		
P03	PPI1D5	P15	RSCLK0	R11	$\overline{\text{EMU}}$	T07	PF12		
P04	PF0	P16	DR0SEC	R12	MISO	T08	VDDEXT		

Table 38. 256-Ball CSP_BGA (12 mm × 12 mm) Ball Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
$\overline{\text{ABE0}}$	E11	$\overline{\text{BR}}$	B12	DT0SEC	N15	GND	N14
$\overline{\text{ABE1}}$	B13	BYPASS	G04	DT1PRI	R15	GND	P02
$\overline{\text{ABE2}}$	A14	CLKIN	F01	DT1SEC	T15	GND	P05
$\overline{\text{ABE3}}$	A15	DATA0	B16	$\overline{\text{EMU}}$	R11	GND	P09
ADDR02	D13	DATA1	C15	GND	C05	GND	P12
ADDR03	G11	DATA2	E12	GND	C11	MISO	R12
ADDR04	B15	DATA3	C16	GND	C13	MOSI	N11
ADDR05	G10	DATA4	E14	GND	D05	NC	M05
ADDR06	B14	DATA5	D15	GND	D06	NC	M13
ADDR07	C14	DATA6	D16	GND	D08	NMI0	P11
ADDR08	F11	DATA7	E15	GND	D14	NMI1	R09
ADDR09	D07	DATA8	F13	GND	E01	PF0	P04
ADDR10	A06	DATA9	F15	GND	E13	PF1	N05
ADDR11	C06	DATA10	F12	GND	F08	PF2	T04
ADDR12	B05	DATA11	F16	GND	F10	PF3	M06
ADDR13	E06	DATA12	F14	GND	G02	PF4	R05
ADDR14	A05	DATA13	G15	GND	G06	PF5	P06
ADDR15	E05	DATA14	G13	GND	G07	PF6	T05
ADDR16	B04	DATA15	G12	GND	G08	PF7	M07
ADDR17	F06	DATA16	H12	GND	G14	PF8	R06
ADDR18	B03	DATA17	H15	GND	H01	PF9	N06
ADDR19	C04	DATA18	H13	GND	H02	PF10	R07
ADDR20	A03	DATA19	H16	GND	H08	PF11	P07
ADDR21	F05	DATA20	H14	GND	H09	PF12	T07
ADDR22	B02	DATA21	J15	GND	H10	PF13	N08
ADDR23	D04	DATA22	J13	GND	J07	PF14	R08
ADDR24	A02	DATA23	J16	GND	J11	PF15	P08
ADDR25	C03	DATA24	K14	GND	J14	PPI0CLK	C02
$\overline{\text{AMS0}}$	C08	DATA25	K15	GND	K07	PPI0D0	L01
$\overline{\text{AMS1}}$	B07	DATA26	K13	GND	K09	PPI0D1	J05
$\overline{\text{AMS2}}$	E07	DATA27	L15	GND	K10	PPI0D2	J03
$\overline{\text{AMS3}}$	A07	DATA28	K12	GND	L03	PPI0D3	J04
$\overline{\text{AOE}}$	C07	DATA29	L16	GND	L07	PPI0D4	K02
ARDY	D09	DATA30	J12	GND	L09	PPI0D5	H05
$\overline{\text{ARE}}$	B08	DATA31	M15	GND	L11	PPI0D6	K01
$\overline{\text{AWE}}$	A08	DR0PRI	L12	GND	L14	PPI0D7	H04
$\overline{\text{BG}}$	A13	DR0SEC	P16	GND	M04	PPI0D8	K03
$\overline{\text{BGH}}$	C12	DR1PRI	M12	GND	M09	PPI0D9	H03
BMODE0	M10	DR1SEC	T14	GND	N07	PPI0D10	F04
BMODE1	N10	DT0PRI	M16	GND	N12	PPI0D11	E02

Table 38. 256-Ball CSP_BGA (12 mm × 12 mm) Ball Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
PPI0D12	E03	PPI1SYNC1	K04	TDO	N09	VDDEXT	M14
PPI0D13	D01	PPI1SYNC2	L02	TFS0	L13	VDDEXT	T01
PPI0D14	G05	PPI1SYNC3	L04	TFS1	P14	VDDEXT	T03
PPI0D15	D02	$\overline{\text{RESET}}$	F03	TMS	T10	VDDEXT	T06
PPI0SYNC1	E04	RFS0	R16	$\overline{\text{TRST}}$	P10	VDDEXT	T08
PPI0SYNC2	C01	RFS1	N13	TSCLK0	N16	VDDEXT	T12
PPI0SYNC3	D03	RSCLK0	P15	TSCLK1	R14	VDDEXT	T16
PPI1CLK	B01	RSCLK1	P13	TX/PF26	R13	VDDINT	E08
PPI1D0	R04	RX	T13	VDDEXT	A01	VDDINT	F07
PPI1D1	N04	SA10	D11	VDDEXT	A04	VDDINT	F09
PPI1D2	R03	$\overline{\text{SCAS}}$	D10	VDDEXT	A09	VDDINT	G09
PPI1D3	N03	SCK	M11	VDDEXT	A16	VDDINT	H06
PPI1D4	T02	SCKE	B10	VDDEXT	B06	VDDINT	H07
PPI1D5	P03	SCLK0	A11	VDDEXT	B11	VDDINT	H11
PPI1D6	R02	SCLK1	A12	VDDEXT	D12	VDDINT	J08
PPI1D7	R01	SLEEP	T11	VDDEXT	E16	VDDINT	J09
PPI1D8	P01	$\overline{\text{SMS0}}$	E09	VDDEXT	F02	VDDINT	J10
PPI1D9	M03	$\overline{\text{SMS1}}$	B09	VDDEXT	G03	VDDINT	K08
PPI1D10	N02	$\overline{\text{SMS2}}$	C09	VDDEXT	G16	VDDINT	K11
PPI1D11	L06	$\overline{\text{SMS3}}$	A10	VDDEXT	J06	VDDINT	L08
PPI1D12	N01	$\overline{\text{SRAS}}$	C10	VDDEXT	K06	VDDINT	M08
PPI1D13	M02	$\overline{\text{SWE}}$	E10	VDDEXT	K16	VROUT0	J01
PPI1D14	K05	TCK	T09	VDDEXT	L05	VROUT1	J02
PPI1D15	M01	TDI	R10	VDDEXT	L10	XTAL	G01

Figure 52 lists the top view of the 256-Ball CSP_BGA (12 mm × 12 mm) ball configuration. Figure 53 lists the bottom view.

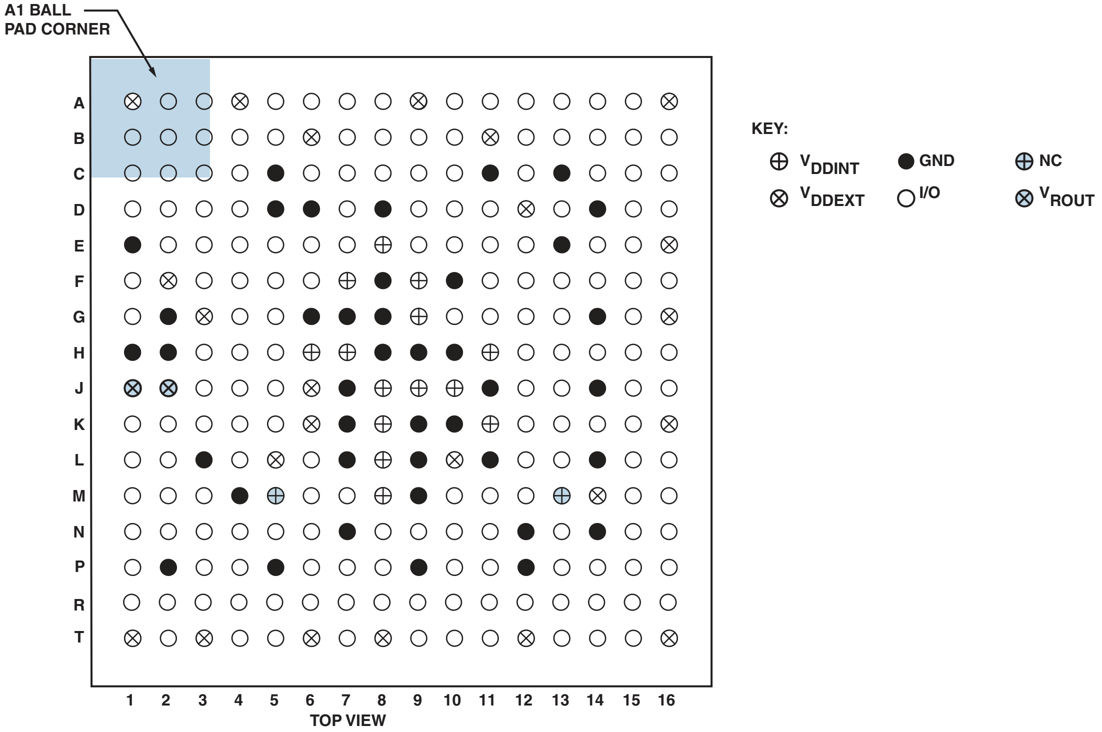


Figure 52. 256-Ball CSP_BGA Ball Configuration (Top View)

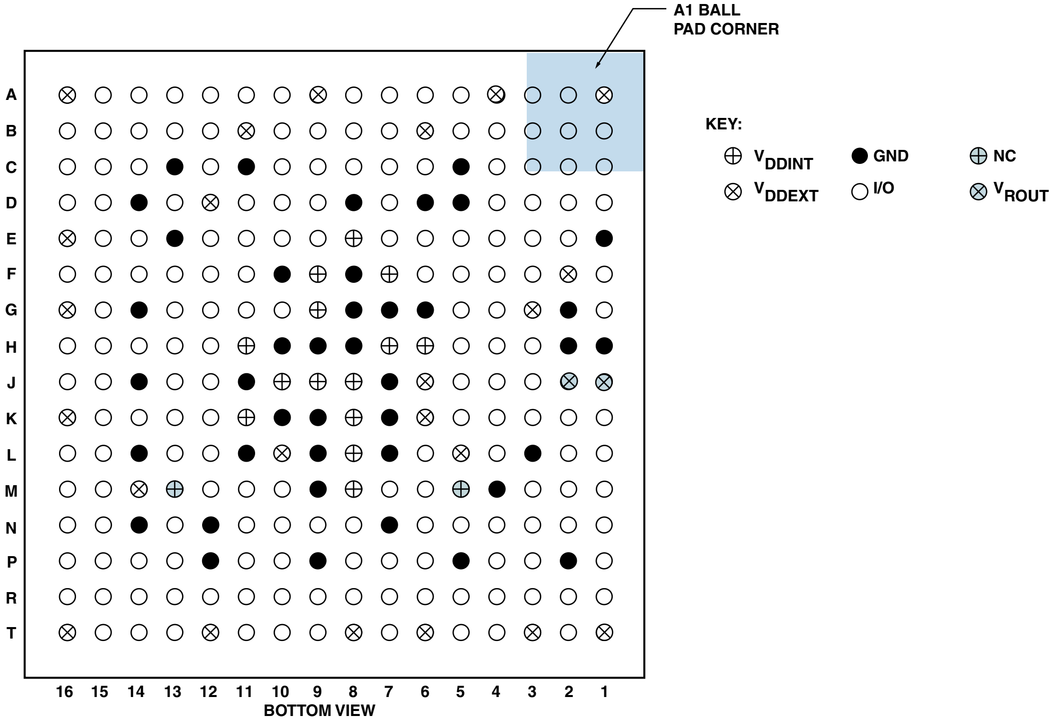


Figure 53. 256-Ball CSP_BGA Ball Configuration (Bottom View)

297-BALL PBGA BALL ASSIGNMENT

Table 39 lists the 297-Ball PBGA ball assignment numerically by ball number. Table 40 lists the ball assignment alphabetically by signal.

Table 39. 297-Ball PBGA Ball Assignment (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A01	GND	B15	$\overline{\text{SMS1}}$	G01	PPI0D11	L14	GND
A02	ADDR25	B16	$\overline{\text{SMS3}}$	G02	PPI0D10	L15	GND
A03	ADDR23	B17	SCKE	G25	DATA4	L16	GND
A04	ADDR21	B18	$\overline{\text{SWE}}$	G26	DATA7	L17	GND
A05	ADDR19	B19	SA10	H01	BYPASS	L18	VDDINT
A06	ADDR17	B20	$\overline{\text{BR}}$	H02	$\overline{\text{RESET}}$	L25	DATA12
A07	ADDR15	B21	$\overline{\text{BG}}$	H25	DATA6	L26	DATA15
A08	ADDR13	B22	$\overline{\text{ABE1}}$	H26	DATA9	M01	VROUT0
A09	ADDR11	B23	$\overline{\text{ABE3}}$	J01	CLKIN	M02	GND
A10	ADDR09	B24	ADDR07	J02	GND	M10	VDDEXT
A11	$\overline{\text{AMS3}}$	B25	GND	J10	VDDEXT	M11	GND
A12	$\overline{\text{AMS1}}$	B26	ADDR05	J11	VDDEXT	M12	GND
A13	$\overline{\text{AWE}}$	C01	PPI0SYNC3	J12	VDDEXT	M13	GND
A14	$\overline{\text{ARE}}$	C02	PPI0CLK	J13	VDDEXT	M14	GND
A15	$\overline{\text{SMS0}}$	C03	GND	J14	VDDEXT	M15	GND
A16	$\overline{\text{SMS2}}$	C04	GND	J15	VDDEXT	M16	GND
A17	$\overline{\text{SRAS}}$	C05	GND	J16	VDDINT	M17	GND
A18	$\overline{\text{SCAS}}$	C22	GND	J17	VDDINT	M18	VDDINT
A19	SCLK0	C23	GND	J18	VDDINT	M25	DATA14
A20	SCLK1	C24	GND	J25	DATA8	M26	DATA17
A21	$\overline{\text{BGH}}$	C25	ADDR04	J26	DATA11	N01	VROUT1
A22	$\overline{\text{ABE0}}$	C26	ADDR03	K01	XTAL	N02	PPI0D9
A23	$\overline{\text{ABE2}}$	D01	PPI0SYNC1	K02	NC	N10	VDDEXT
A24	ADDR08	D02	PPI0SYNC2	K10	VDDEXT	N11	GND
A25	ADDR06	D03	GND	K11	VDDEXT	N12	GND
A26	GND	D04	GND	K12	VDDEXT	N13	GND
B01	PPI1CLK	D23	GND	K13	VDDEXT	N14	GND
B02	GND	D24	GND	K14	VDDEXT	N15	GND
B03	ADDR24	D25	ADDR02	K15	VDDEXT	N16	GND
B04	ADDR22	D26	DATA1	K16	VDDINT	N17	GND
B05	ADDR20	E01	PPI0D15	K17	VDDINT	N18	VDDINT
B06	ADDR18	E02	PPI0D14	K18	VDDINT	N25	DATA16
B07	ADDR16	E03	GND	K25	DATA10	N26	DATA19
B08	ADDR14	E24	GND	K26	DATA13	P01	PPI0D7
B09	ADDR12	E25	DATA0	L01	NC	P02	PPI0D8
B10	ADDR10	E26	DATA3	L02	NC	P10	VDDEXT
B11	$\overline{\text{AMS2}}$	F01	PPI0D13	L10	VDDEXT	P11	GND
B12	$\overline{\text{AMS0}}$	F02	PPI0D12	L11	GND	P12	GND
B13	$\overline{\text{AOE}}$	F25	DATA2	L12	GND	P13	GND
B14	ARDY	F26	DATA5	L13	GND	P14	GND

Table 39. 297-Ball PBGA Ball Assignment (Numerically by Ball Number) (Continued)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
P15	GND	U11	VDDEXT	AC04	GND	AE21	RX
P16	GND	U12	VDDEXT	AC23	GND	AE22	RFS1
P17	GND	U13	VDDEXT	AC24	GND	AE23	DR1SEC
P18	VDDINT	U14	GND	AC25	DR0SEC	AE24	TFS1
P25	DATA18	U15	VDDINT	AC26	RFS0	AE25	GND
P26	DATA21	U16	VDDINT	AD01	PPI1D7	AE26	NC
R01	PPI0D5	U17	VDDINT	AD02	PPI1D6	AF01	GND
R02	PPI0D6	U18	VDDINT	AD03	GND	AF02	PPI1D4
R10	VDDEXT	U25	DATA24	AD04	GND	AF03	PPI1D2
R11	GND	U26	DATA27	AD05	GND	AF04	PPI1D0
R12	GND	V01	PPI1SYNC3	AD22	GND	AF05	PF1
R13	GND	V02	PPI0D0	AD23	GND	AF06	PF3
R14	GND	V25	DATA26	AD24	GND	AF07	PF5
R15	GND	V26	DATA29	AD25	NC	AF08	PF7
R16	GND	W01	PPI1SYNC1	AD26	RSCLK0	AF09	PF9
R17	GND	W02	PPI1SYNC2	AE01	PPI1D5	AF10	PF11
R18	VDDINT	W25	DATA28	AE02	GND	AF11	PF13
R25	DATA20	W26	DATA31	AE03	PPI1D3	AF12	PF15
R26	DATA23	Y01	PPI1D15	AE04	PPI1D1	AF13	NMI1
T01	PPI0D3	Y02	PPI1D14	AE05	PF0	AF14	TCK
T02	PPI0D4	Y25	DATA30	AE06	PF2	AF15	TDI
T10	VDDEXT	Y26	DT0PRI	AE07	PF4	AF16	TMS
T11	GND	AA01	PPI1D13	AE08	PF6	AF17	SLEEP
T12	GND	AA02	PPI1D12	AE09	PF8	AF18	NMI0
T13	GND	AA25	DT0SEC	AE10	PF10	AF19	SCK
T14	GND	AA26	TSCLK0	AE11	PF12	AF20	TX
T15	GND	AB01	PPI1D11	AE12	PF14	AF21	RSCLK1
T16	GND	AB02	PPI1D10	AE13	NC	AF22	DR1PRI
T17	GND	AB03	GND	AE14	TDO	AF23	TSCLK1
T18	VDDINT	AB24	GND	AE15	$\overline{\text{TRST}}$	AF24	DT1SEC
T25	DATA22	AB25	TFS0	AE16	$\overline{\text{EMU}}$	AF25	DT1PRI
T26	DATA25	AB26	DR0PRI	AE17	BMODE1	AF26	GND
U01	PPI0D1	AC01	PPI1D9	AE18	BMODE0		
U02	PPI0D2	AC02	PPI1D8	AE19	MISO		
U10	VDDEXT	AC03	GND	AE20	MOSI		

Table 40. 297-Ball PBGA Ball Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
$\overline{\text{ABE0}}$	A22	$\overline{\text{BR}}$	B20	DT0SEC	AA25	GND	N15
$\overline{\text{ABE1}}$	B22	BYPASS	H01	DT1PRI	AF25	GND	N16
$\overline{\text{ABE2}}$	A23	CLKIN	J01	DT1SEC	AF24	GND	N17
$\overline{\text{ABE3}}$	B23	DATA0	E25	$\overline{\text{EMU}}$	AE16	GND	P11
ADDR02	D25	DATA1	D26	GND	A01	GND	P12
ADDR03	C26	DATA2	F25	GND	A26	GND	P13
ADDR04	C25	DATA3	E26	GND	B02	GND	P14
ADDR05	B26	DATA4	G25	GND	B25	GND	P15
ADDR06	A25	DATA5	F26	GND	C03	GND	P16
ADDR07	B24	DATA6	H25	GND	C04	GND	P17
ADDR08	A24	DATA7	G26	GND	C05	GND	R11
ADDR09	A10	DATA8	J25	GND	C22	GND	R12
ADDR10	B10	DATA9	H26	GND	C23	GND	R13
ADDR11	A09	DATA10	K25	GND	C24	GND	R14
ADDR12	B09	DATA11	J26	GND	D03	GND	R15
ADDR13	A08	DATA12	L25	GND	D04	GND	R16
ADDR14	B08	DATA13	K26	GND	D23	GND	R17
ADDR15	A07	DATA14	M25	GND	D24	GND	T11
ADDR16	B07	DATA15	L26	GND	E03	GND	T12
ADDR17	A06	DATA16	N25	GND	E24	GND	T13
ADDR18	B06	DATA17	M26	GND	J02	GND	T14
ADDR19	A05	DATA18	P25	GND	L11	GND	T15
ADDR20	B05	DATA19	N26	GND	L12	GND	T16
ADDR21	A04	DATA20	R25	GND	L13	GND	T17
ADDR22	B04	DATA21	P26	GND	L14	GND	U14
ADDR23	A03	DATA22	T25	GND	L15	GND	AB03
ADDR24	B03	DATA23	R26	GND	L16	GND	AB24
ADDR25	A02	DATA24	U25	GND	L17	GND	AC03
$\overline{\text{AMS0}}$	B12	DATA25	T26	GND	M02	GND	AC04
$\overline{\text{AMS1}}$	A12	DATA26	V25	GND	M11	GND	AC23
$\overline{\text{AMS2}}$	B11	DATA27	U26	GND	M12	GND	AC24
$\overline{\text{AMS3}}$	A11	DATA28	W25	GND	M13	GND	AD03
$\overline{\text{AOE}}$	B13	DATA29	V26	GND	M14	GND	AD04
ARDY	B14	DATA30	Y25	GND	M15	GND	AD05
$\overline{\text{ARE}}$	A14	DATA31	W26	GND	M16	GND	AD22
$\overline{\text{AWE}}$	A13	DR0PRI	AB26	GND	M17	GND	AD23
$\overline{\text{BG}}$	B21	DR0SEC	AC25	GND	N11	GND	AD24
$\overline{\text{BGH}}$	A21	DR1PRI	AF22	GND	N12	GND	AE02
BMODE0	AE18	DR1SEC	AE23	GND	N13	GND	AE25
BMODE1	AE17	DT0PRI	Y26	GND	N14	GND	AF01

Table 40. 297-Ball PBGA Ball Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
GND	AF26	PPI0D7	P01	RSCLK0	AD26	VDDEXT	K13
MISO	AE19	PPI0D8	P02	RSCLK1	AF21	VDDEXT	K14
MOSI	AE20	PPI0D9	N02	RX	AE21	VDDEXT	K15
NC	K02	PPI0D10	G02	SA10	B19	VDDEXT	L10
NC	L01	PPI0D11	G01	$\overline{\text{SCAS}}$	A18	VDDEXT	M10
NC	L02	PPI0D12	F02	SCK	AF19	VDDEXT	N10
NC	AD25	PPI0D13	F01	SCKE	B17	VDDEXT	P10
NC	AE13	PPI0D14	E02	SCLK0	A19	VDDEXT	R10
NC	AE26	PPI0D15	E01	SCLK1	A20	VDDEXT	T10
NMI0	AF18	PPI0SYNC1	D01	SLEEP	AF17	VDDEXT	U10
NMI1	AF13	PPI0SYNC2	D02	$\overline{\text{SMS0}}$	A15	VDDEXT	U11
PF0	AE05	PPI0SYNC3	C01	$\overline{\text{SMS1}}$	B15	VDDEXT	U12
PF1	AF05	PPI1CLK	B01	$\overline{\text{SMS2}}$	A16	VDDEXT	U13
PF2	AE06	PPI1D0	AF04	$\overline{\text{SMS3}}$	B16	VDDINT	J16
PF3	AF06	PPI1D1	AE04	$\overline{\text{SRAS}}$	A17	VDDINT	J17
PF4	AE07	PPI1D2	AF03	$\overline{\text{SWE}}$	B18	VDDINT	J18
PF5	AF07	PPI1D3	AE03	TCK	AF14	VDDINT	K16
PF6	AE08	PPI1D4	AF02	TDI	AF15	VDDINT	K17
PF7	AF08	PPI1D5	AE01	TDO	AE14	VDDINT	K18
PF8	AE09	PPI1D6	AD02	TFS0	AB25	VDDINT	L18
PF9	AF09	PPI1D7	AD01	TFS1	AE24	VDDINT	M18
PF10	AE10	PPI1D8	AC02	TMS	AF16	VDDINT	N18
PF11	AF10	PPI1D9	AC01	$\overline{\text{TRST}}$	AE15	VDDINT	P18
PF12	AE11	PPI1D10	AB02	TSCLK0	AA26	VDDINT	R18
PF13	AF11	PPI1D11	AB01	TSCLK1	AF23	VDDINT	T18
PF14	AE12	PPI1D12	AA02	TX/PF26	AF20	VDDINT	U15
PF15	AF12	PPI1D13	AA01	VDDEXT	J10	VDDINT	U16
PPI0CLK	C02	PPI1D14	Y02	VDDEXT	J11	VDDINT	U17
PPI0D0	V02	PPI1D15	Y01	VDDEXT	J12	VDDINT	U18
PPI0D1	U01	PPI1SYNC1	W01	VDDEXT	J13	VROUT0	M01
PPI0D2	U02	PPI1SYNC2	W02	VDDEXT	J14	VROUT1	N01
PPI0D3	T01	PPI1SYNC3	V01	VDDEXT	J15	XTAL	K01
PPI0D4	T02	$\overline{\text{RESET}}$	H02	VDDEXT	K10		
PPI0D5	R01	RFS0	AC26	VDDEXT	K11		
PPI0D6	R02	RFS1	AE22	VDDEXT	K12		

Figure 54 lists the top view of the 297-Ball PBGA ball configuration. Figure 55 lists the bottom view.

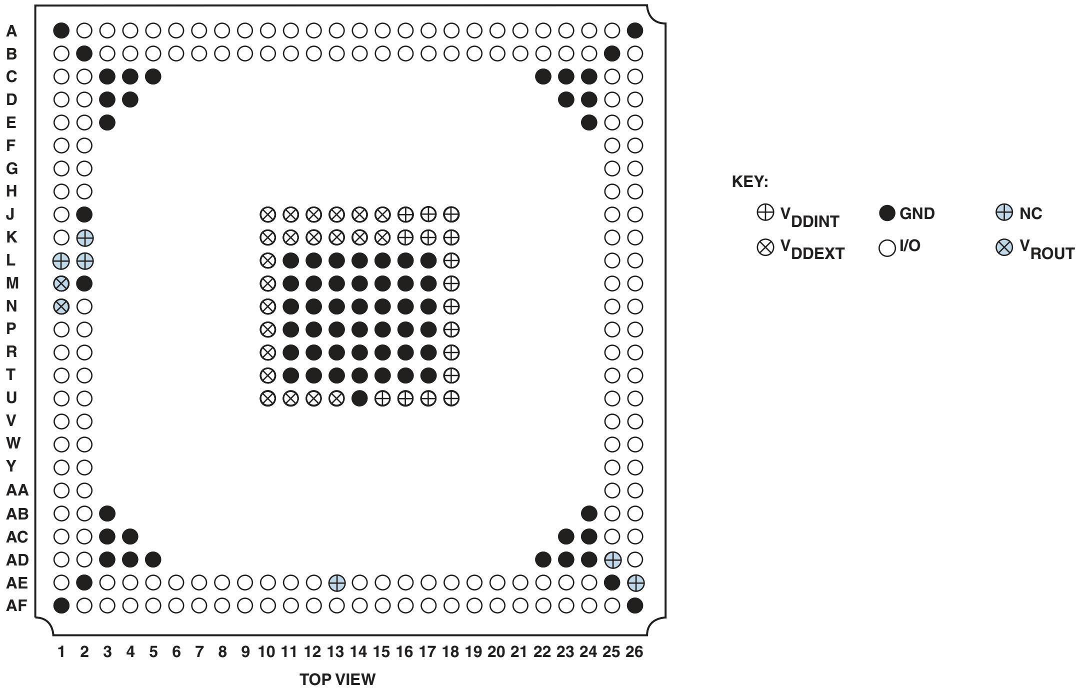


Figure 54. 297-Ball PBGA Ball Configuration (Top View)

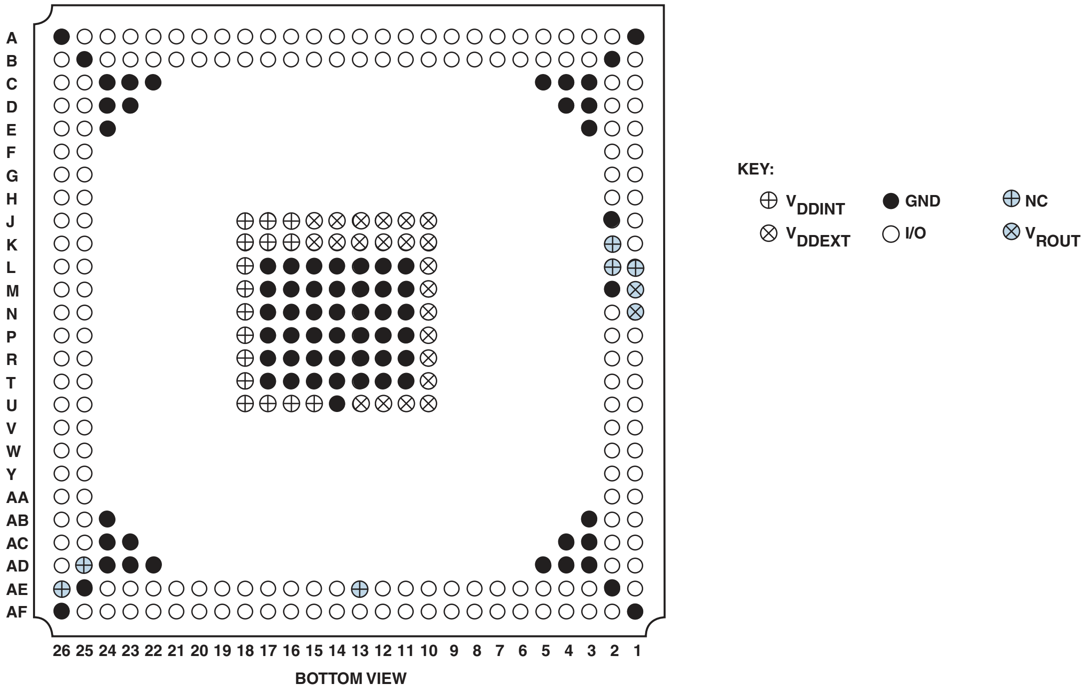
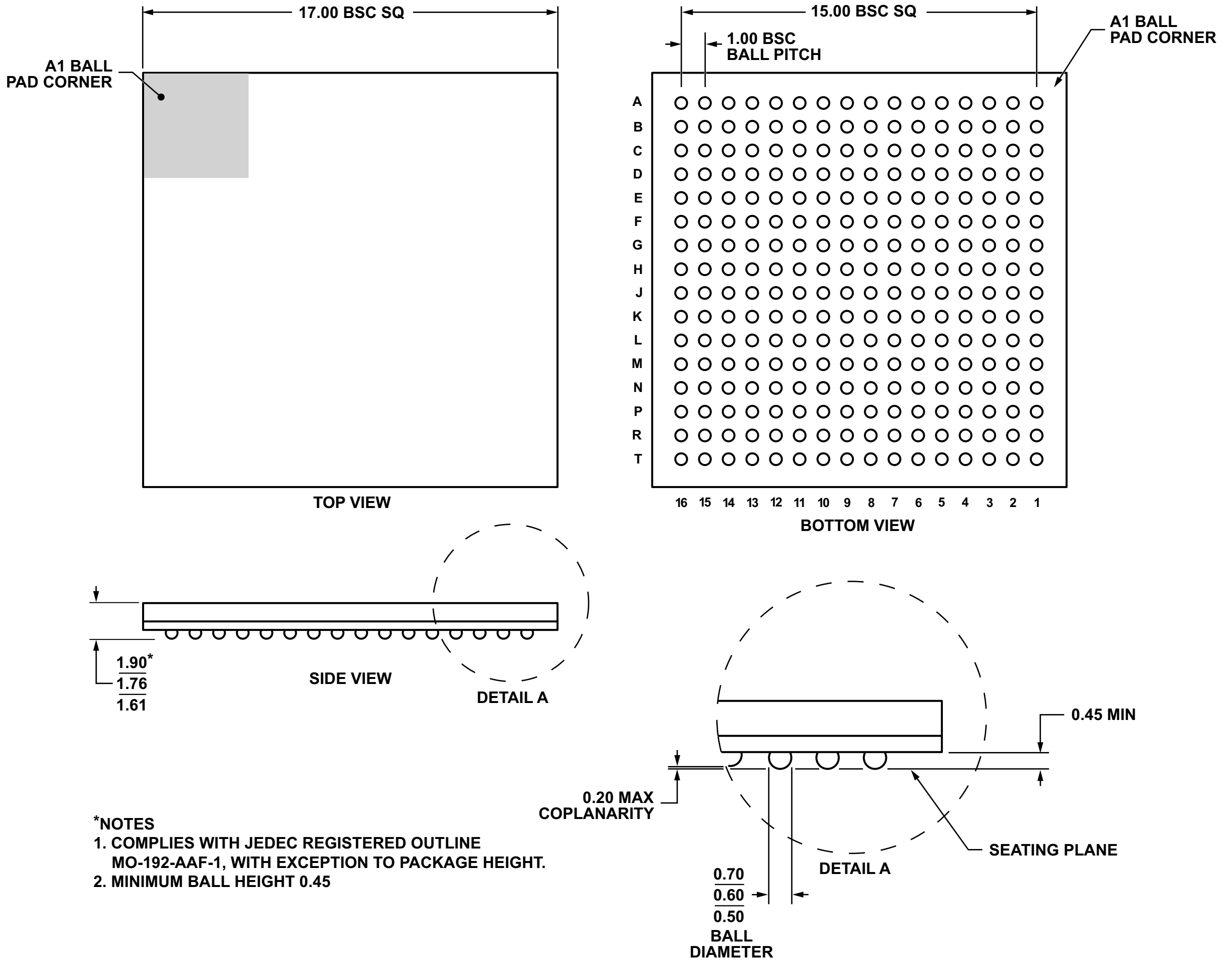


Figure 55. 297-Ball PBGA Ball Configuration (Bottom View)

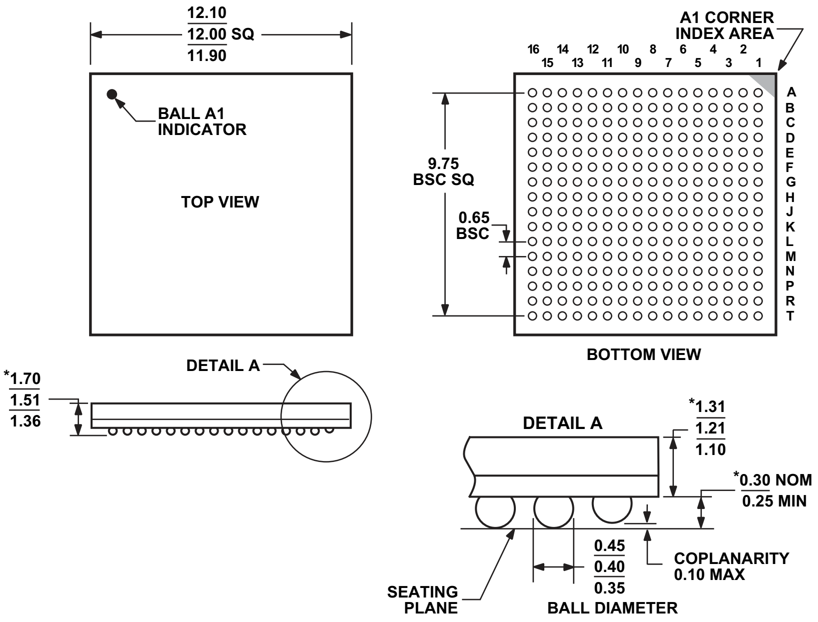
OUTLINE DIMENSIONS

Dimensions in the outline dimension figures are shown in millimeters.



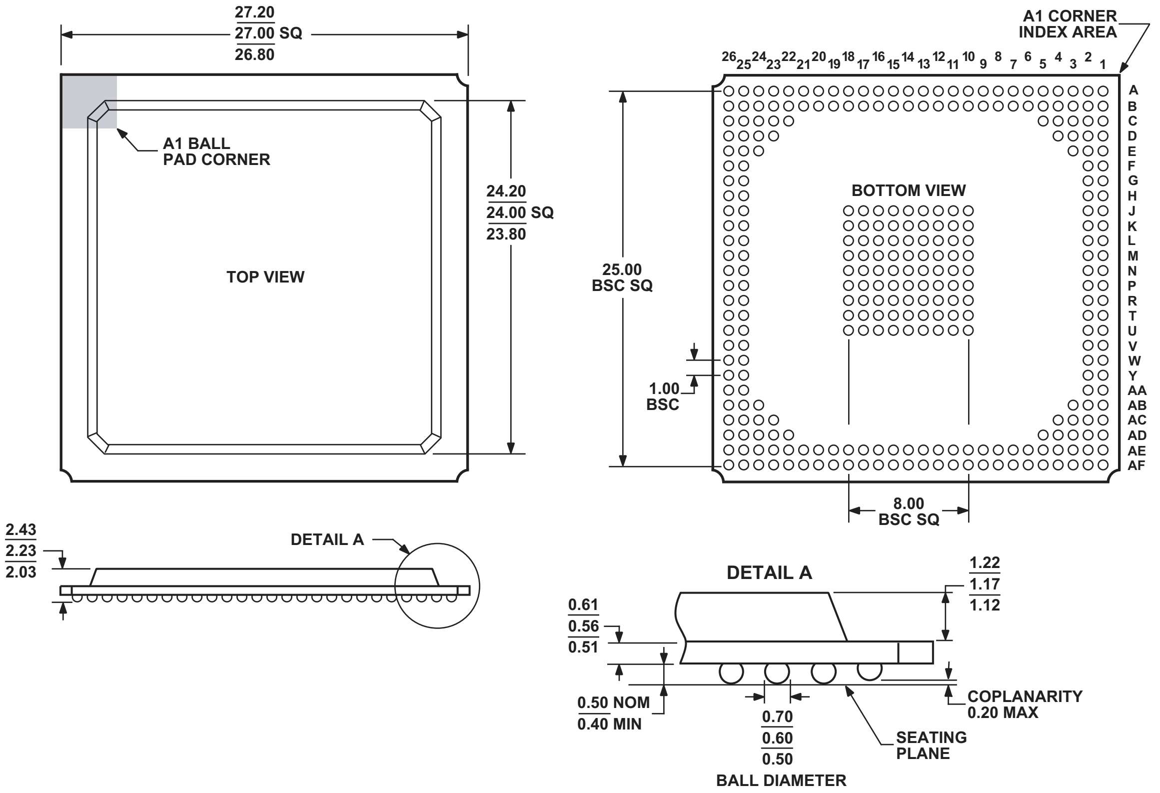
*NOTES
1. COMPLIES WITH JEDEC REGISTERED OUTLINE MO-192-AAF-1, WITH EXCEPTION TO PACKAGE HEIGHT.
2. MINIMUM BALL HEIGHT 0.45

Figure 56. 256-Ball Chip Scale Package Ball Grid Array (CSP_BGA) (BC-256-4)



*COMPLIANT TO JEDEC STANDARDS MO-225 WITH EXCEPTION TO DIMENSIONS INDICATED BY AN ASTERISK.

Figure 57. 256-Ball Chip Scale Package Ball Grid Array (CSP_BGA) (BC-256-1)



COMPLIANT TO JEDEC STANDARDS MS-034-AAL-1

Figure 58. 297-Ball Plastic Ball Grid Array (PBGA) (B-297)

SURFACE-MOUNT DESIGN

Table 41 is provided as an aid to PCB design. For industry-standard design recommendations, refer to IPC-7351, *Generic Requirements for Surface Mount Design and Land Pattern Standard*.

Table 41. BGA Data for Use with Surface-Mount Design

Package	Ball Attach Type	Solder Mask Opening	Ball Pad Size
256-Ball CSP_BGA (BC-256-1)	Solder Mask Defined	0.30 mm diameter	0.43 mm diameter
256-Ball CSP_BGA (BC-256-4)	Solder Mask Defined	0.43 mm diameter	0.55 mm diameter
297-Ball PBGA (B-297)	Solder Mask Defined	0.43 mm diameter	0.58 mm diameter

ORDERING GUIDE

Model[1]	Temperature Range[2]	Speed Grade (Max)	Package Description	Package Option
ADSP-BF561SKBCZ-6V	0°C to 70°C	600 MHz	256-Ball CSP_BGA	BC-256-1
ADSP-BF561SKBCZ-5V	0°C to 70°C	533 MHz	256-Ball CSP_BGA	BC-256-1
ADSP-BF561SKBCZ500	0°C to 70°C	500 MHz	256-Ball CSP_BGA	BC-256-1
ADSP-BF561SKBZ600	0°C to 70°C	600 MHz	297-Ball PBGA	B-297
ADSP-BF561SBBZ500	–40°C to +85°C	500 MHz	297-Ball PBGA	B-297
ADSP-BF561SKBCZ-6A	0°C to 70°C	600 MHz	256-Ball CSP_BGA	BC-256-4
ADSP-BF561SKBCZ-5A	0°C to 70°C	500 MHz	256-Ball CSP_BGA	BC-256-4
ADSP-BF561SBBCZ-5A	–40°C to +85°C	500 MHz	256-Ball CSP_BGA	BC-256-4

[1] Z = RoHS compliant part.

[2] Referenced temperature is ambient temperature.

ANALOG DEVICES

www.analog.com