Blackfin® Embedded Symmetric Multiprocessor

ADSP-BF561

FEATURES

- Dual symmetric 600 MHz high performance Blackfin cores
- 328K bytes of on-chip memory (see memory information on Page 4)
- 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
- 0.8 V to 1.35 V core V_{DD} with on-chip voltage regulator
- 3.3 V and 2.5 V compliant I/O
- 256-ball mini-BGA and 297-ball PBGA package options

PERIPHERALS

- Two parallel input/output peripheral interface units supporting ITU-R 656 video and glueless interface to analog front end ADCs
- Two dual channel, full duplex synchronous serial ports supporting eight stereo I^2S channels
- Dual 16-channel DMA controllers and one 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
- 48 programmable flags
- On-chip phase-locked loop capable of 0.5× to 64× frequency multiplication

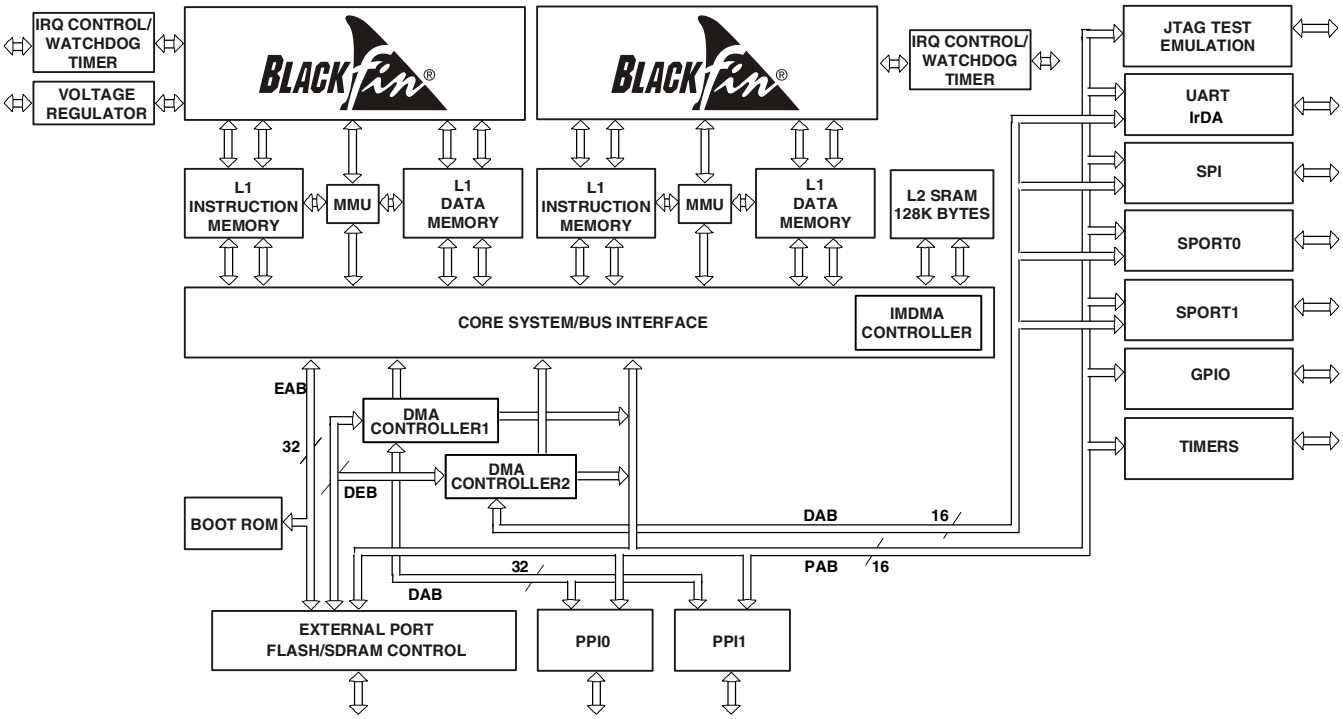

Figure 1. Functional Block Diagram

Rev. A

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106 U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.461.3113

TABLE OF CONTENTS

REVISION HISTORY

5/06—Changes from Rev. 0 to Rev. A

Minor format and wording changes throughout the docunment.

1/05—Initial version

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.

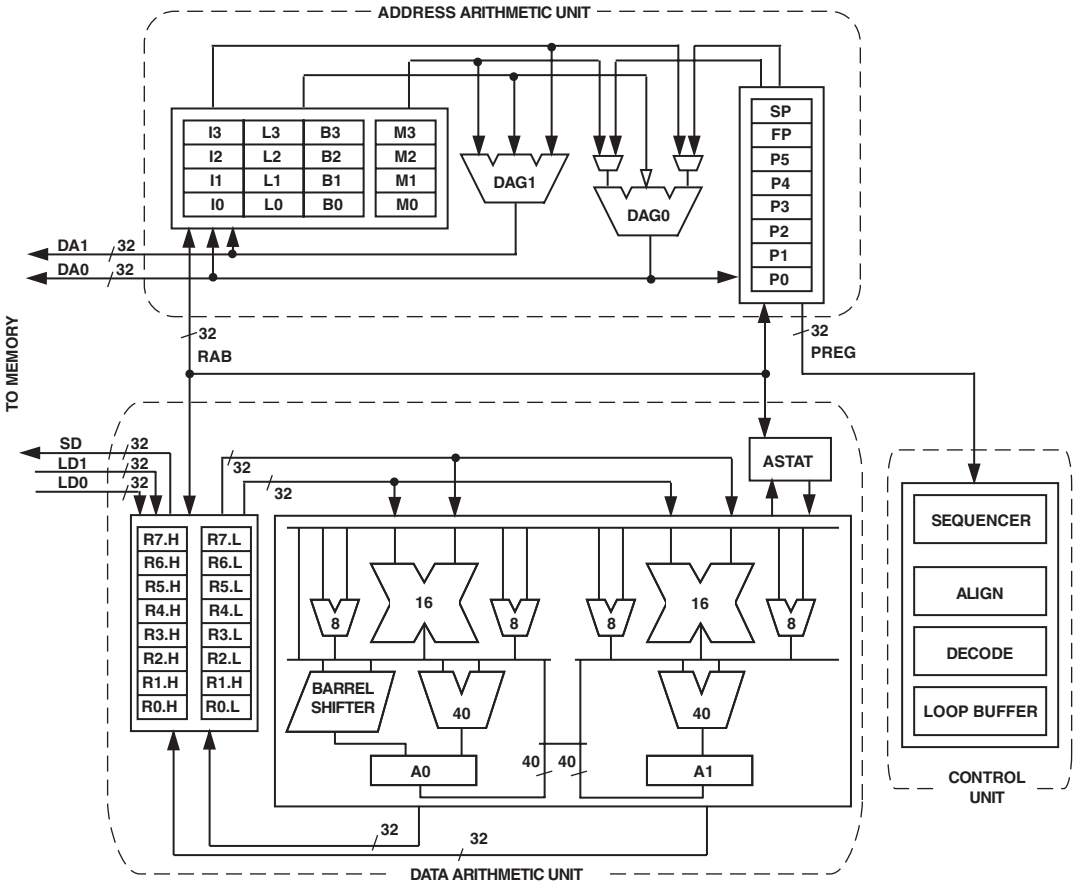

Figure 2. Blackfin Processor Core

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).

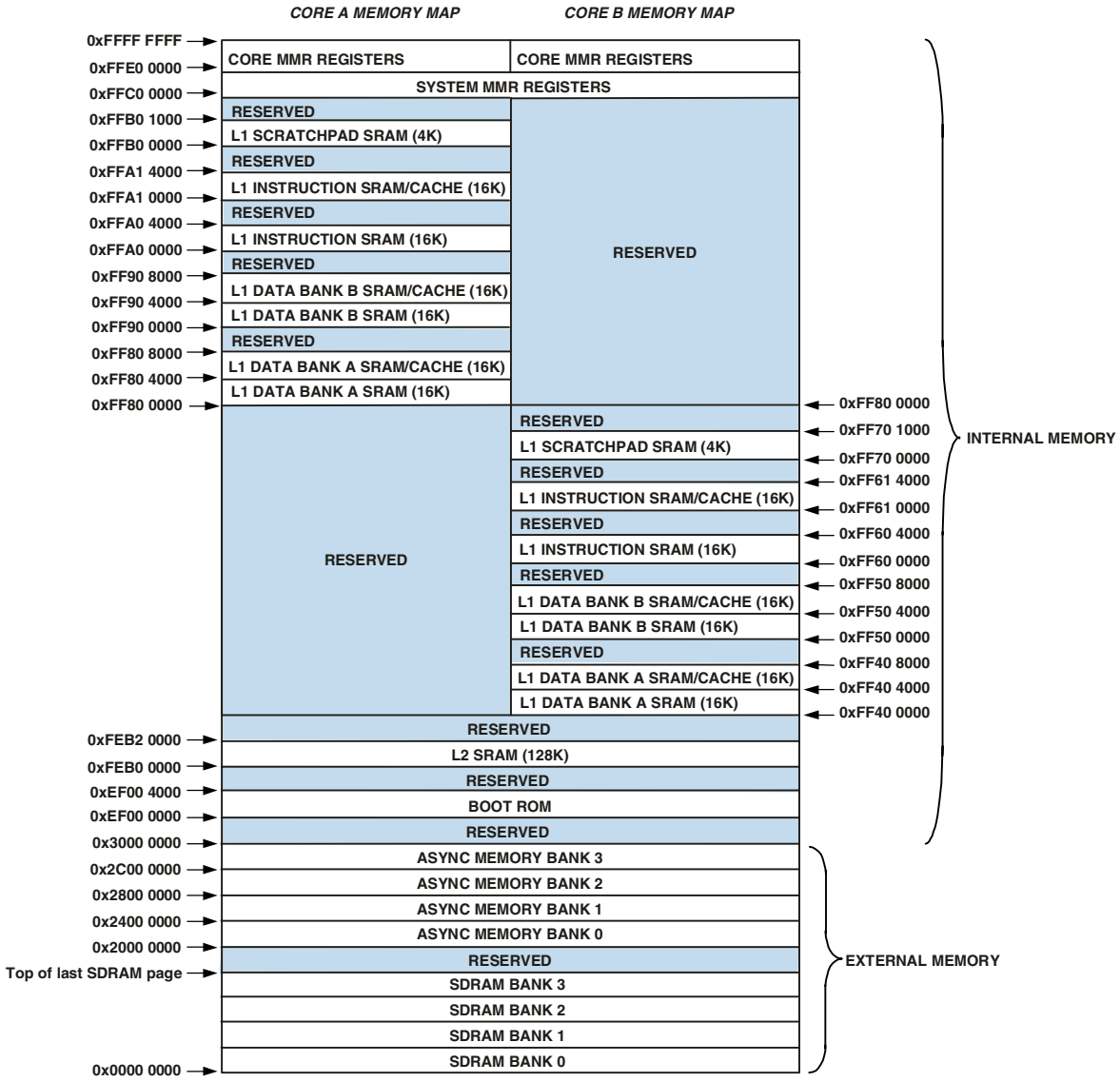

Figure 3. Memory Map

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 writ-

ing 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 be written only when its 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 super-

visor mode. (Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions.)

- 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 Register (SIC_IMASK0, SIC_IMASK1) – This register controls the masking and unmasking of each peripheral interrupt event. When a bit is set in the register, that peripheral event is unmasked and will be processed by the system when asserted. A cleared bit in the register masks the peripheral event thereby preventing the processor from servicing the event.
- SIC Interrupt Status Register (SIC_ISR0, SIC_ISR1)– As multiple peripherals can be mapped to a single event, this register allows the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt; a cleared bit indicates the peripheral is not asserting the event.
- SIC Interrupt Wakeup Enable Register (SIC_IWR0, SIC_IWR1) – By enabling the corresponding bit in this register, each peripheral can be configured to wake up the processor, should the processor be in a powered-down mode when the event is generated.

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 multiple, independent DMA controllers that support automated data transfers with minimal overhead for the DSP core. 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 PPI. 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.

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 DLH register (most significant 8 bits) and 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. The programmable flag pins have special functions for SPI port operation. 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 – The Flag Interrupt Mask 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 – The Flag Interrupt Sensitivity Registers specify whether individual PFx pins are level- or edge-sensitive and specify, if edge-sensitive, whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge sensitivity.

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, ITU-R 601/656 video encoders and decoders, and other general-purpose peripherals. Each PPI consists of a dedicated input clock pin, up to three frame synchronization pins, and up to 16 data pins. The input clock supports parallel data rates up to half the system clock rate.

In ITU-R 656 modes, the PPI receives and parses a data stream of 8-bit or 10-bit data elements. On-chip decode of embedded preamble control and synchronization information is supported.

Three distinct ITU-R 656 modes are supported:

- Active video only – 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.
- Vertical blanking only – The PPI only transfers vertical blanking interval (VBI) data, as well as horizontal blanking information and control byte sequences on VBI lines.
- Entire field – The entire incoming bitstream 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.

Though not explicitly supported, ITU-R 656 output functionality can be achieved by setting up the entire frame structure (including active video, blanking, and control information) in memory and streaming the data out the PPI in a frame sync-less mode. The processor's 2-D DMA features facilitate this transfer by allowing the static frame buffer (blanking and control codes) to be placed in memory once, and simply updating the active video information on a per-frame basis.

The general-purpose modes of the PPI are intended to suit a wide variety of data capture and transmission applications. The modes are divided into four main categories, each allowing up to 16 bits of data transfer per PPI_CLK cycle:

- Data receive with internally generated frame syncs
- Data receive with externally generated frame syncs
- Data transmit with internally generated frame syncs
- Data transmit with externally generated frame syncs

These modes support ADC/DAC connections, as well as video communication with hardware signaling. Many of the modes support more than one level of frame synchronization. If desired, a programmable delay can be inserted between assertion of a frame sync and reception/transmission of data.

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	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 interrupt ($\overline{\text{RESET}}$). 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 Operating 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:

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

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:

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

VOLTAGE REGULATION

The ADSP-BF561 processor provides an on-chip voltage regulator that can generate processor core voltage levels 0.85 V to 1.25 V from an external 2.25 V to 3.6 V supply. 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, 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.

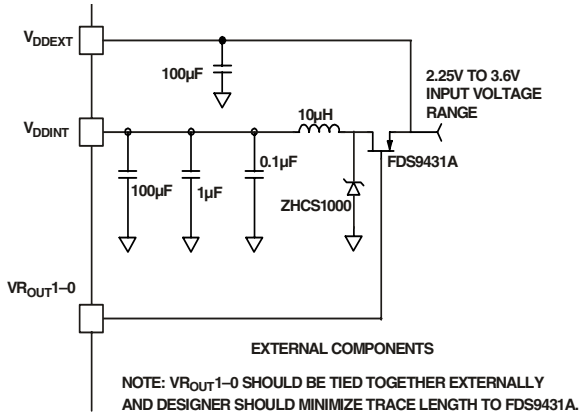

Figure 4. Voltage Regulator Circuit

CLOCK SIGNALS

The ADSP-BF561 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 should be a TTL-compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor CLKIN pin. When an external clock is used, the XTAL pin must be *left* unconnected.

Alternatively, because the ADSP-BF561 includes an on-chip oscillator circuit, an external crystal may be used. The crystal should be connected across the CLKIN and XTAL pins, with two capacitors connected as shown in Figure 5.

Capacitor values are dependent on crystal type and should be specified by the crystal manufacturer. A parallel-resonant, fundamental frequency, microprocessor-grade crystal should be used.

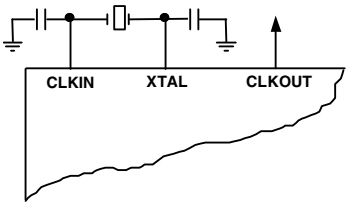

Figure 5. External Crystal Connections

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.

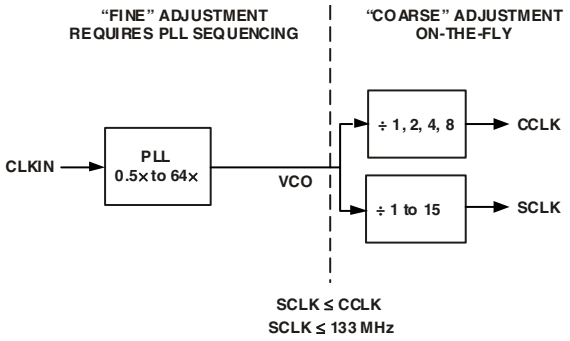

Figure 6. Frequency Modification Methods

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 or L2 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	Reserved
11	Boot from SPI serial EEPROM (16-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).
- 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 serial EEPROM (16-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-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. 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 pro-

vides 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

The ADSP-BF561 is supported with a complete set of CROSSCORE®† software and hardware development tools, including Analog Devices emulators and the VisualDSP++®‡ development environment. The same emulator hardware that supports other Analog Devices processors also fully emulates the ADSP-BF561.

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

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

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

- View mixed C/C++ and assembly code (interleaved source and object information).
- Insert breakpoints.
- Set conditional breakpoints on registers, memory, and stacks.
- Trace instruction execution.
- Perform linear or statistical profiling of program execution.
- Fill, dump, and graphically plot the contents of memory.
- Perform source level debugging.
- Create custom debugger windows.

The VisualDSP++ IDE lets programmers define and manage software development. Its dialog boxes and property pages let programmers configure and manage all development tools, including Color Syntax Highlighting in the VisualDSP++ editor. These capabilities permit programmers to:

- Control how the development tools process inputs and generate outputs.
- Maintain a one-to-one correspondence with the tool's command line switches.

The VisualDSP++ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of embedded, real-time programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning when developing new application code. The VDK features include threads, critical and unscheduled regions, semaphores, events, and device flags. The VDK also supports priority-based, pre-emptive, cooperative, and time-sliced scheduling approaches. In addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

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

† CROSSCORE is a registered trademark of Analog Devices, Inc.
‡ VisualDSP++ is a registered trademark of Analog Devices, Inc.

VCSE is Analog Devices' technology for creating, using, and reusing software components (independent modules of substantial functionality) to quickly and reliably assemble software applications. Components can be downloaded from the Web and dropped into the application. Component archives can be published from within VisualDSP++. VCSE supports component implementation in C/C++ or assembly language.

The Expert Linker can be used to visually manipulate the placement of code and data in the embedded system. Memory utilization can be viewed in a color-coded graphical form. Code and data can be easily moved to different areas of the processor or external memory with the drag of the mouse. Runtime stack and heap usage can be examined. The Expert Linker is fully compatible with existing Linker Definition File (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices emulators use the IEEE 1149.1 JTAG test access port of the ADSP-BF561 to monitor and control the target board processor during emulation. The emulator provides full-speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect the loading or timing of the target system.

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

EZ-KIT Lite Evaluation Board

For evaluation of ADSP-BF561 processors, use the ADSP-BF561 EZ-KIT Lite® board available from Analog Devices. Order part number ADDS-BF561-EZLITE. The board comes with on-chip emulation capabilities and is equipped to enable software development. Multiple daughter cards are available.

DESIGNING AN EMULATOR-COMPATIBLE PROCESSOR BOARD (TARGET)

The Analog Devices family of emulators are tools that every system developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on the ADSP-BF561. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor is set running at full speed with no impact on system timing.

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

For details on target board design issues, including mechanical layout, single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see *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.

RELATED DOCUMENTS

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*

PIN DESCRIPTIONS

ADSP-BF561 pin definitions are listed in Table 8. Unused inputs should be tied or pulled to V_{DDEXT} or GND. Output drive currents for each driver type are shown in Figure 26 through Figure 33.

Table 8. Pin Descriptions

Pin Name	Type	Function	Driver Type[1]	Pull-Up/Down Requirement
EBIU				
ADDR25–2	O	Address Bus for Async/Sync Access	A	None
DATA31–0	I/O	Data Bus for Async/Sync Access	A	None
$\overline{\text{ABE}}$3–0/SDQM3–0	O	Byte Enables/Data Masks for Async/Sync Access	A	None
$\overline{\text{BR}}$	I	Bus Request		Pull-up Required If Function Not Used
$\overline{\text{BG}}$	O	Bus Grant	A	None
$\overline{\text{BGH}}$	O	Bus Grant Hang	A	None
EBIU (ASYNC)				
$\overline{\text{AMS3–0}}$	O	Bank Select	A	None
ARDY	I	Hardware Ready Control		Pull-up Required If Function Not Used
$\overline{\text{AOE}}$	O	Output Enable	A	None
$\overline{\text{AWE}}$	O	Write Enable	A	None
$\overline{\text{ARE}}$	O	Read Enable	A	None
EBIU (SDRAM)				
$\overline{\text{SRAS}}$	O	Row Address Strobe	A	None
$\overline{\text{SCAS}}$	O	Column Address Strobe	A	None
$\overline{\text{SWE}}$	O	Write Enable	A	None
SCKE	O	Clock Enable	A	None
SCLK0/CLKOUT	O	Clock Output Pin 0	B	None
SCLK1	O	Clock Output Pin 1	B	None
SA10	O	SDRAM A10 Pin	A	None
$\overline{\text{SMS3–0}}$	O	Bank Select	A	None
PF/TIMER				
PF0/$\overline{\text{SPISS}}$/TMR0	I/O	Programmable Flag/Slave SPI Select/Timer	C	None
PF1/SPISEL1/TMR1	I/O	Programmable Flag/SPI Select/Timer	C	None
PF2/SPISEL2/TMR2	I/O	Programmable Flag/SPI Select/Timer	C	None
PF3/SPISEL3/TMR3	I/O	Programmable Flag/SPI Select/Timer	C	None
PF4/SPISEL4/TMR4	I/O	Programmable Flag/SPI Select/Timer	C	None
PF5/SPISEL5/TMR5	I/O	Programmable Flag/SPI Select/Timer	C	None
PF6/SPISEL6/TMR6	I/O	Programmable Flag/SPI Select/Timer	C	None
PF7/SPISEL7/TMR7	I/O	Programmable Flag/SPI Select/Timer	C	None
PF8	I/O	Programmable Flag	C	None
PF9	I/O	Programmable Flag	C	None
PF10	I/O	Programmable Flag	C	None
PF11	I/O	Programmable Flag	C	None
PF12	I/O	Programmable Flag	C	None
PF13	I/O	Programmable Flag	C	None
PF14	I/O	Programmable Flag	C	None
PF15/EXT CLK	I/O	Programmable Flag/External Timer Clock Input	C	None

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type[1]	Pull-Up/Down Requirement
PPI0				
PPI0D15–8/PF47–40	I/O	PPI Data/Programmable Flag Pins	C	None
PPI0D7–0	I/O	PPI Data Pins	C	None
PPI0CLK	I	PPI Clock		None
PPI0SYNC1/TMR8	I/O	PPI Sync/Timer	C	None
PPI0SYNC2/TMR9	I/O	PPI Sync/Timer	C	None
PPI0SYNC3	I/O	PPI Sync	C	None
PPI1				
PPI1D15–8/PF39–32	I/O	PPI Data/Programmable Flag Pins	C	None
PPI1D7–0	I/O	PPI Data Pins	C	None
PPI1CLK	I	PPI Clock		None
PPI1SYNC1/TMR10	I/O	PPI Sync/Timer	C	None
PPI1SYNC2/TMR11	I/O	PPI Sync/Timer	C	None
PPI1SYNC3	I/O	PPI Sync	C	None
SPORT0				
RSCLK0/PF28	I/O	Sport0/Programmable Flag	D	None
RFS0/PF19	I/O	Sport0 Receive Frame Sync/Programmable Flag	C	None
DR0PRI	I	Sport0 Receive Data Primary		None
DR0SEC/PF20	I/O	Sport0 Receive Data Secondary/Programmable Flag	C	None
TSCLK0/PF29	I/O	Sport0 Transmit Serial Clock/Programmable Flag	D	None
TFS0/PF16	I/O	Sport0 Transmit Frame Sync/Programmable Flag	C	None
DT0PRI/PF18	I/O	Sport0 Transmit Data Primary/Programmable Flag	C	None
DT0SEC/PF17	I/O	Sport0 Transmit Data Secondary/Programmable Flag	C	None
SPORT1				
RSCLK1/PF30	I/O	Sport1/Programmable Flag	D	None
RFS1/PF24	I/O	Sport1 Receive Frame Sync/Programmable Flag	C	None
DR1PRI	I	Sport1 Receive Data Primary		None
DR1SEC/PF25	I/O	Sport1 Receive Data Secondary/Programmable Flag	C	None
TSCLK1/PF31	I/O	Sport1 Transmit Serial Clock/Programmable Flag	D	None
TFS1/PF21	I/O	Sport1 Transmit Frame Sync/Programmable Flag	C	None
DT1PRI/PF23	I/O	Sport1 Transmit Data Primary/Programmable Flag	C	None
DT1SEC/PF22	I/O	Sport1 Transmit Data Secondary/Programmable Flag	C	None
SPI				
MOSI	I/O	Master Out Slave In	C	None
MISO	I/O	Master In Slave Out	C	Pull-up is Necessary if Booting via SPI
SCK	I/O	SPI Clock	D	None
UART				
RX/PF27	I/O	UART Receive/Programmable Flag	C	None
TX/PF26	I/O	UART Transmit/Programmable Flag	C	None

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type[1]	Pull-Up/Down Requirement
JTAG				
$\overline{\text{EMU}}$	O	Emulation Output	C	None
TCK	I	JTAG Clock		Internal Pull-down
TDO	O	JTAG Serial Data Out	C	None
TDI	I	JTAG Serial Data In		Internal Pull-down
TMS	I	JTAG Mode Select		Internal Pull-down
$\overline{\text{TRST}}$	I	JTAG Reset		External Pull-down Necessary If JTAG Not Used
Clock				
CLKIN	I	Clock input		Needs to be at a Level or Clocking
XTAL	O	Crystal connection		None
Mode Controls				
$\overline{\text{RESET}}$	I	Chip reset signal		Always Active if Core Power On
NMI0	I	Nonmaskable Interrupt Core A		Pull-down Required If Function Not Used
NMI1	I	Nonmaskable Interrupt Core B		Pull-down Required If Function Not Used
BMODE1–0	I	Dedicated Mode Pin, Configures the Boot Mode that Follows a Hardware or Software Reset		Pull-up or Pull-down Required
SLEEP	O	Sleep	C	None
BYPASS	I	PLL BYPASS Control		Pull-up or Pull-down Required
Voltage Regulator				
VROUT1–0	O	Regulation Output		N/A
Supplies				
VDDEXT	P	Power Supply		N/A
VDDINT	P	Power Supply		N/A
GND	G	Power Supply Return		N/A
No Connection	NC	NC		N/A

[1] Refer to Figure 27 on Page 41 to Figure 31 on Page 42.

SPECIFICATIONS

Component specifications are subject to change without notice.

RECOMMENDED OPERATING CONDITIONS

Parameter		Min	Nominal	Max	Unit
V_{DDINT}[1]	Internal Supply Voltage ADSP-BF561SKBCZ500	0.8	1.25	1.375	V
V_{DDINT}[1]	Internal Supply Voltage ADSP-BF561SKBCZ600	0.8	1.25	1.375	V
V_{DDINT}[2]	Internal Supply Voltage ADSP-BF561SBB600	0.8	1.35	1.4185	V
V_{DDINT}[3]	Internal Supply Voltage ADSP-BF561SBB500	0.8	1.25	1.375	V
V_{DDINT}	Internal Supply Voltage ADSP-BF561WBBZ-5A	0.95	1.25	1.312	V
V_{DDEXT}	External Supply Voltage	2.25	2.5 or 3.3	3.6	V
V_{DDEXT}	External Supply Voltage ADSP-BF561WBBZ-5A	2.7	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

[1] Internal voltage regulator tolerance:
ADSP-BF561SKBCZ500, ADSP-BF561SKBCZ600: V_{DDINT} = –5% to +10%
[2] Internal voltage regulator tolerance:
ADSP-BF561SBB600: V_{DDINT} = –7% to +12%
[3] Internal voltage regulator tolerance:
ADSP-BF561SBB500: V_{DDINT} = –7% to +12% except at 1.25 V: V_{DDINT} = –5% to +10%
[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.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Typical	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

[1] Applies to output and bidirectional pins.
[2] Applies to input pins except JTAG inputs.
[3] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{TRST}$).
[4] Absolute value.
[5] Applies to three-statable pins.
[6] Applies to all signal pins.
[7] Guaranteed, but not tested.

ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed in the table 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.

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	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 9.

Table 9. Maximum Duty Cycle for Input Transient Voltage[1]

V_{IN} Min (V)	V_{IN} Max (V)	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.

ESD SENSITIVITY

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-BF561 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PACKAGE INFORMATION

The information presented in Figure 7 and Table 10 provides information about how to read the package brand and relate it to specific product features. For a complete listing of product offerings, see the Ordering Guide on Page 58.

Figure 7. Product Information on Package

Table 10. Package Brand Information

Brand Key	Field Description
t	Temperature Range
pp	Package Type
Z	Lead Free Option (Optional)
ccc	See Ordering Guide
vvvvvv.x	Assembly Lot Code
n.n	Silicon Revision
yyww	Date Code

TIMING SPECIFICATIONS

Table 11 through Table 13 describe the timing requirements for the ADSP-BF561 clocks. 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 on Page 21. Table 14 describes phase-locked loop operating conditions.

Table 11. Core Clock Requirements—ADSP-BF561SKBCZ500, ADSP-BF561SKB500, ADSP-BF561SKBZ500, ADSP-BF561SBB500, ADSP-BF561SBBZ500, and ADSP-BF561WBBZ-5A

Parameter		Min	Max	Unit
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.1875 Vminimum)	2.00		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.045 Vminimum)	2.25		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.95 Vminimum)	2.86		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.855 Vminimum)[1]	3.33		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.8 V minimum)[1]	4.00		ns

[1] Not applicable to ADSP-BF561WBBZ-5A.

Table 12. Core Clock Requirements—ADSP-BF561SKBCZ600

Parameter		Min	Max	Unit
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.1875 Vminimum)	1.66		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.045 Vminimum)	2.10		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.95 Vminimum)	2.35		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.855 Vminimum)	2.66		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.8 V minimum)	4.00		ns

Table 13. Core Clock Requirements—ADSP-BF561SBB600, ADSP-BF561SBBZ600, ADSP-BF561SKB600 and ADSP-BF561SKBZ600

Parameter		Min	Max	Unit
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.2825 Vminimum)[1]	1.66		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.1875 Vminimum)	2.00		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=1.045 Vminimum)	2.25		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.95 Vminimum)	2.86		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.855 V minimum)	3.33		ns
t_{CCLK}	Core Cycle Period (V_{DDINT}=0.8 Vminimum)	4.00		ns

[1] External voltage regulator required to ensure proper operation at 600 MHz 1.35 V nominal.

Table 14. Phase-Locked Loop Operating Conditions

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

Table 15. Maximum SCLK Conditions

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

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

Clock and Reset Timing

Table 16 and Figure 8 describe clock and reset operations. Per Absolute Maximum Ratings on Page 21, combinations of CLKIN and clock multipliers must not select core/peripheral clocks in excess of 600 MHz/133 MHz.

Table 16. Clock and Reset Timing

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

[1] If DF bit in PLL_CTL register is set, then the maximum t_{CKIN} period is 50 ns.
[2] Applies to bypass mode and nonbypass mode.
[3] Applies after power-up sequence is complete. At power-up, the processor's internal phase-locked loop requires no more than 2,000 CLKIN cycles, while $\overline{\text{RESET}}$ is asserted, assuming stable power supplies and CLKIN (not including startup time of external clock oscillator).

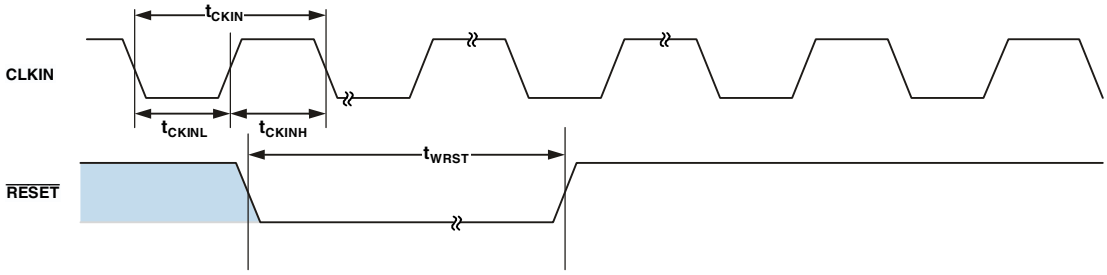

Figure 8. Clock and Reset Timing

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 $\overline{AMS3-0}$, $\overline{ABE3-0}$, ADDR25–2, $\overline{AOE}$, $\overline{ARE}$.

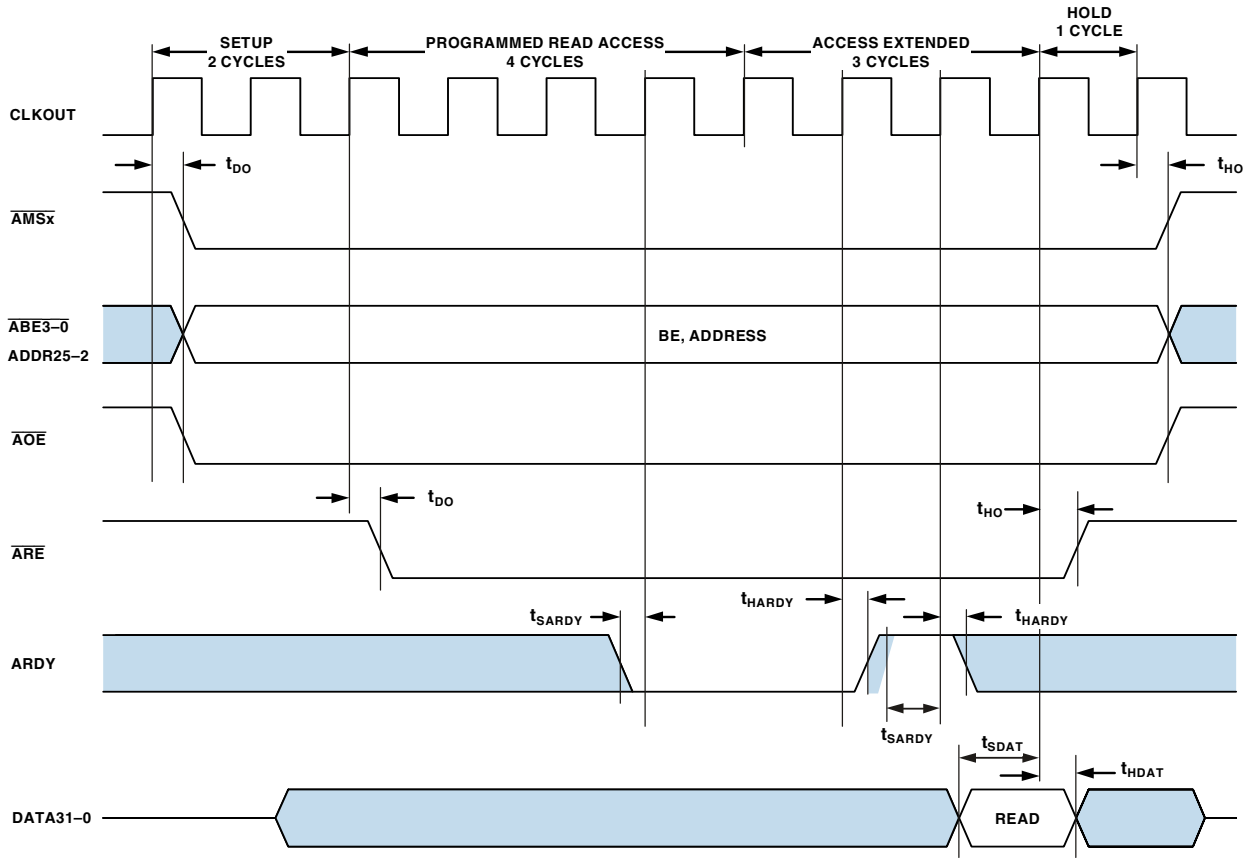

Figure 9. Asynchronous Memory Read Cycle Timing

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 $\overline{AMS3-0}$, $\overline{ABE3-0}$, ADDR25–2, DATA31–0, $\overline{AOE}$, $\overline{AWE}$.

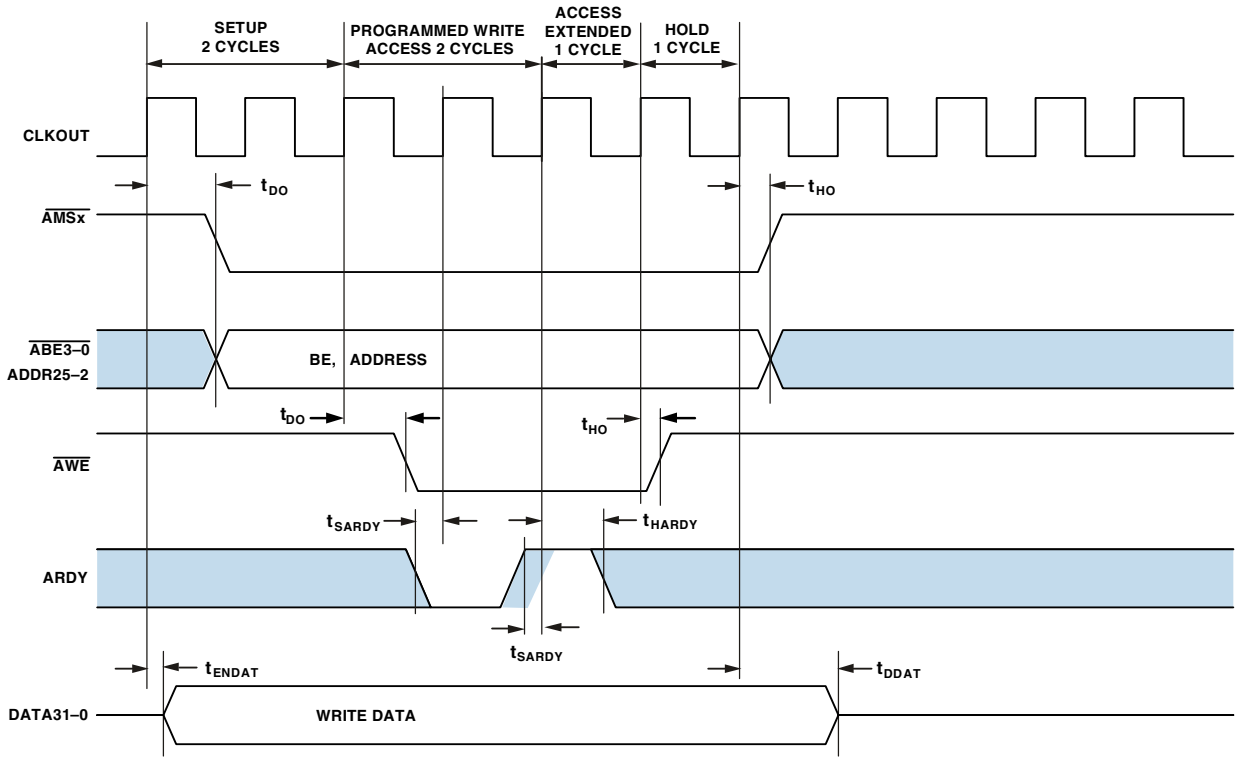

Figure 10. Asynchronous Memory Write Cycle Timing

SDRAM Interface Timing

Table 19. SDRAM Interface Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSDAT}	DATA Setup Before CLKOUT	2.1		ns
t_{HSDAT}	DATA Hold After CLKOUT	0		ns
Switching Characteristics				
t_{SCLK}	CLKOUT Period[1]	7.5		ns
t_{SCLKH}	CLKOUT Width High	2.5		ns
t_{SCLKL}	CLKOUT Width Low	2.5		ns
t_{DCAD}	Command, ADDR, Data Delay After CLKOUT[2]		4.0	ns
t_{HCAD}	Command, ADDR, Data Hold After CLKOUT[2]	0.8		ns
t_{DSDAT}	Data Disable After CLKOUT		4.0	ns
t_{ENSDAT}	Data Enable After CLKOUT	1.0		ns

[1] Refer to Table 15 on Page 22 for maximum f_{SCLK} at various V_{DDINT}.
[2] Command pins include: $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, SDQM, $\overline{SMS3–0}$, SA10, SCKE.

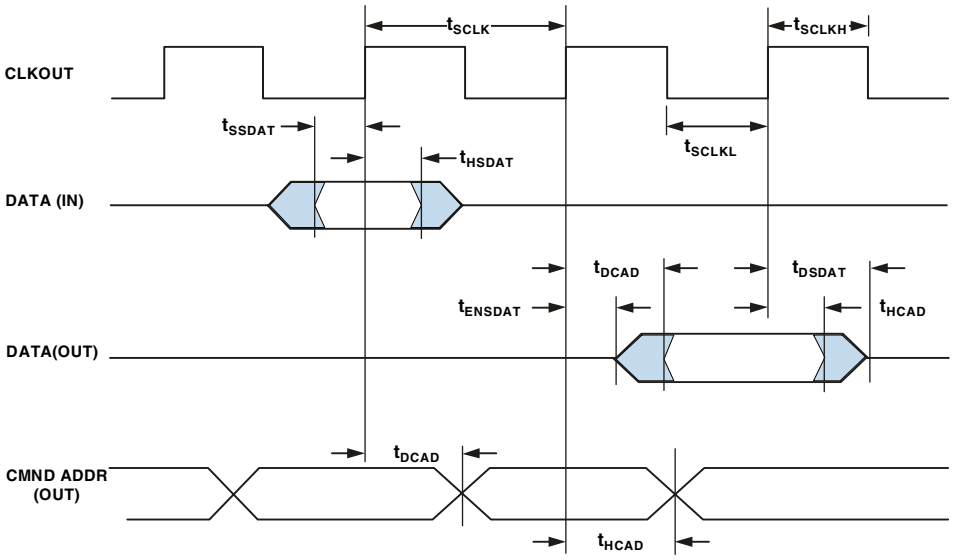

Figure 11. SDRAM Interface Timing

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}	$\overline{BR}$ Asserted to CLKOUT High Setup	4.6		ns
t_{BH}	CLKOUT High to $\overline{BR}$ Deasserted Hold Time	0.0		ns
Switching Characteristics				
t_{SD}	CLKOUT Low to $\overline{SMS}$, Address and $\overline{RD}$/$\overline{WR}$ Disable		4.5	ns
t_{SE}	CLKOUT Low to $\overline{SMS}$, Address and $\overline{RD}$/$\overline{WR}$ Enable		4.5	ns
t_{DBG}	CLKOUT High to $\overline{BG}$ Asserted Setup		3.6	ns
t_{EBG}	CLKOUT High to $\overline{BG}$ Deasserted Hold Time		3.6	ns
t_{DBH}	CLKOUT High to $\overline{BGH}$ Asserted Setup		3.6	ns
t_{EBH}	CLKOUT High to $\overline{BGH}$ 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.

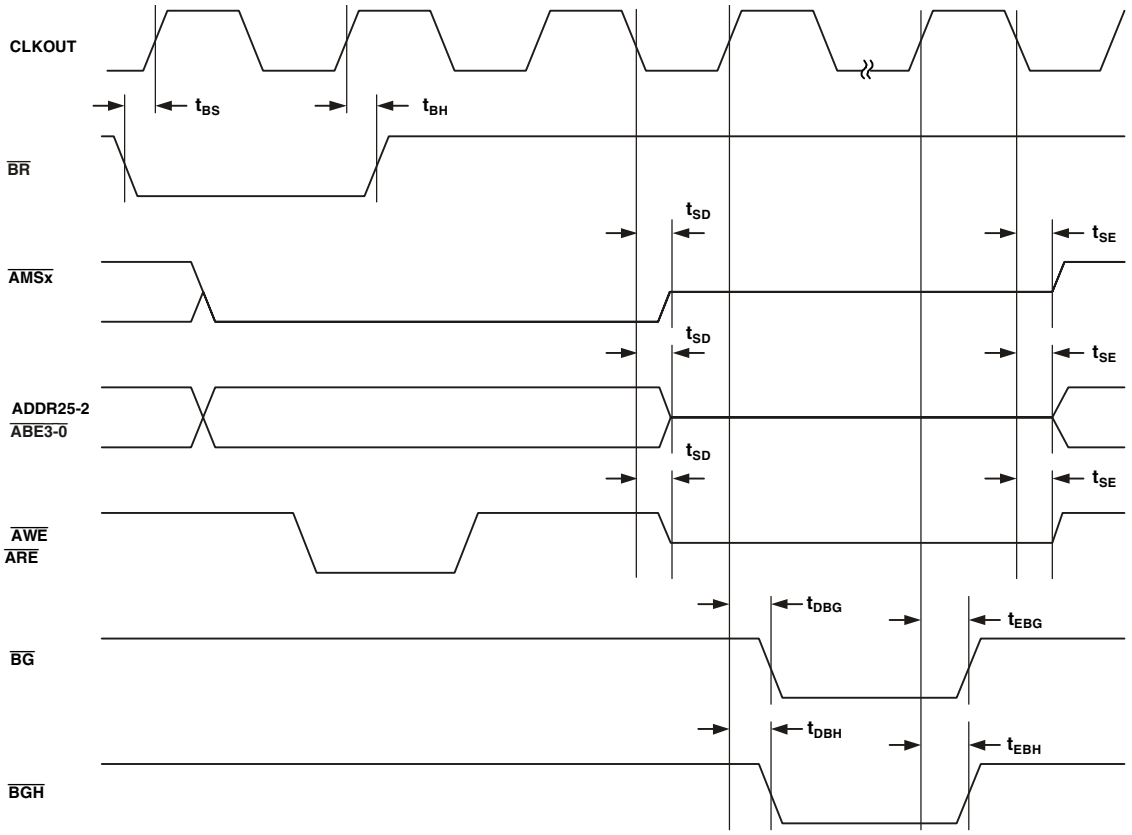

Figure 12. External Port Bus Request and Grant Cycle Timing

Parallel Peripheral Interface Timing

Table 21, and Figure 13 through Figure 16 , describe Parallel Peripheral Interface operations.

Table 21. Parallel Peripheral Interface Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{PCLKW}	PPI_CLK Width[1]	5.0		ns
t_{PCLK}	PPI_CLK Period[1]	13.3		ns
t_{SFSPE}	External Frame Sync Setup Before PPI_CLK	4.0		ns
t_{HFSPE}	External Frame Sync Hold After PPI_CLK	1.0		ns
t_{SDRPE}	Receive Data Setup Before PPI_CLK	3.5		ns
t_{HDRPE}	Receive Data Hold After PPI_CLK	2.0		ns
Switching Characteristics				
t_{DFSPE}	Internal Frame Sync Delay After PPI_CLK		8.0	ns
t_{HOFSPE}	Internal Frame Sync Hold After PPI_CLK	1.7		ns
t_{DDTPE}	Transmit Data Delay After PPI_CLK		8.0	ns
t_{HDTPE}	Transmit Data Hold After PPI_CLK	2.0		ns

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

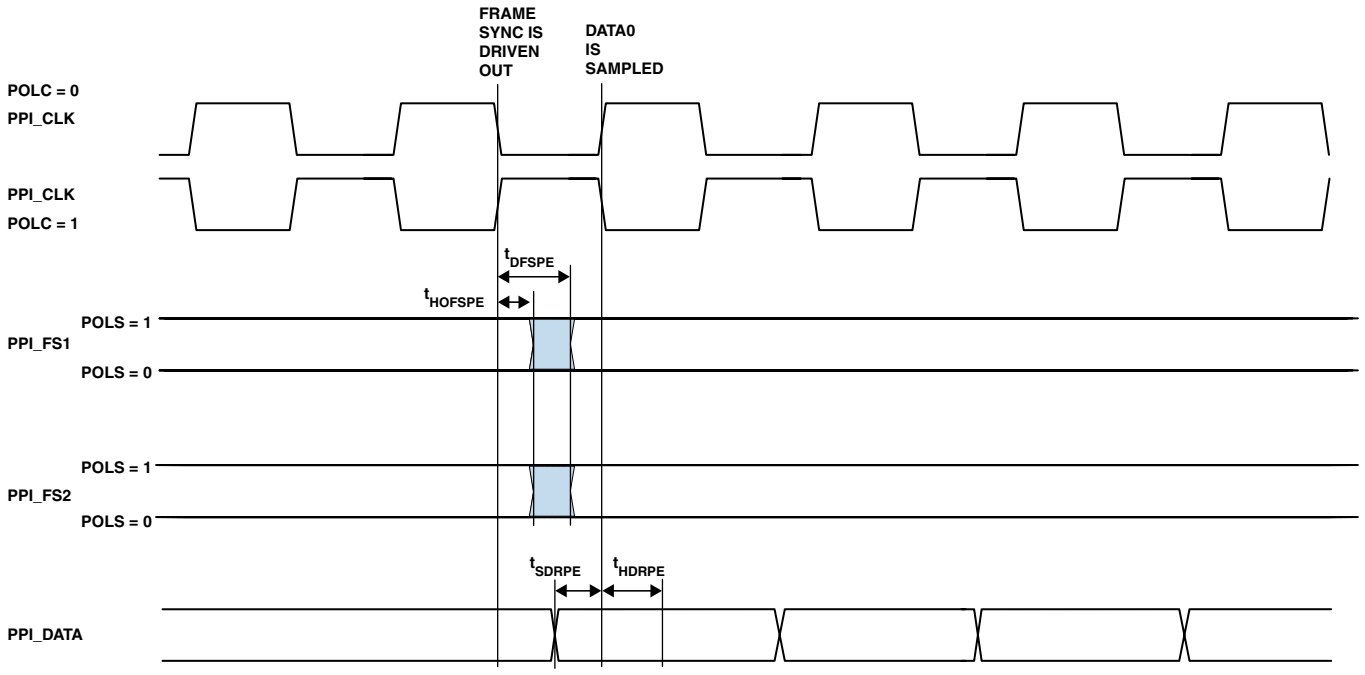

Figure 13. PPI GP Rx Mode with Internal Frame Sync Timing

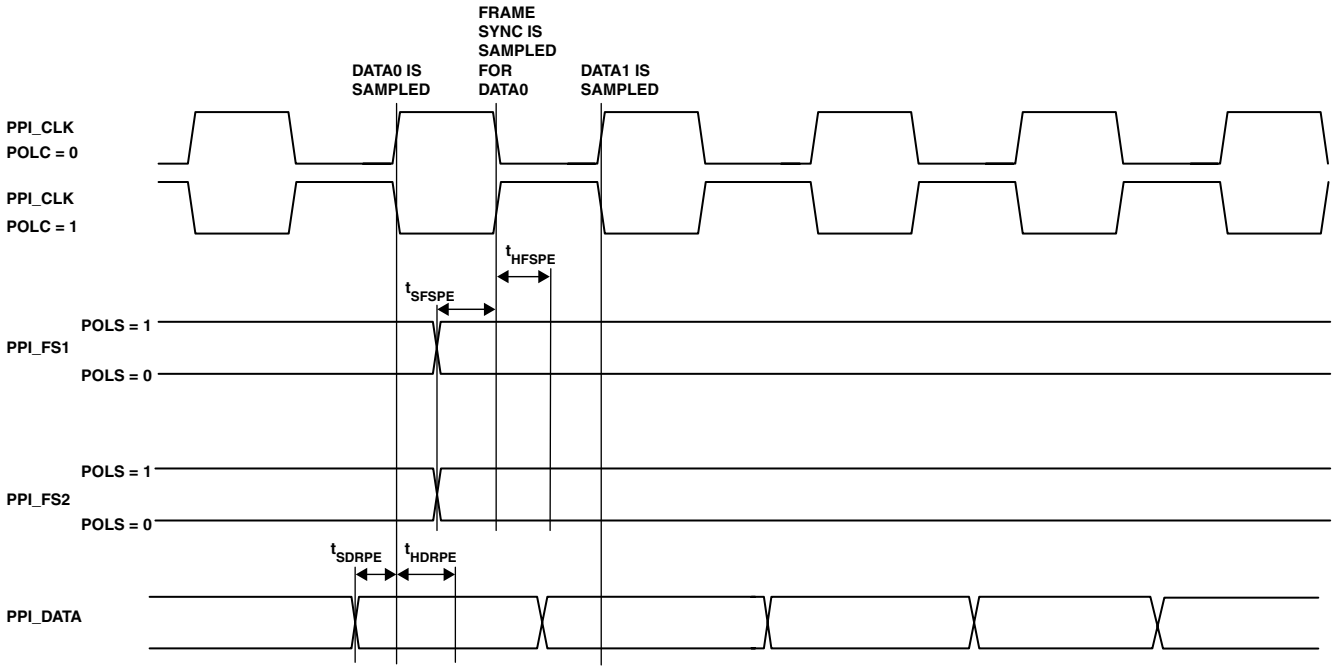

Figure 14. PPI GP Rx Mode with External Frame Sync Timing

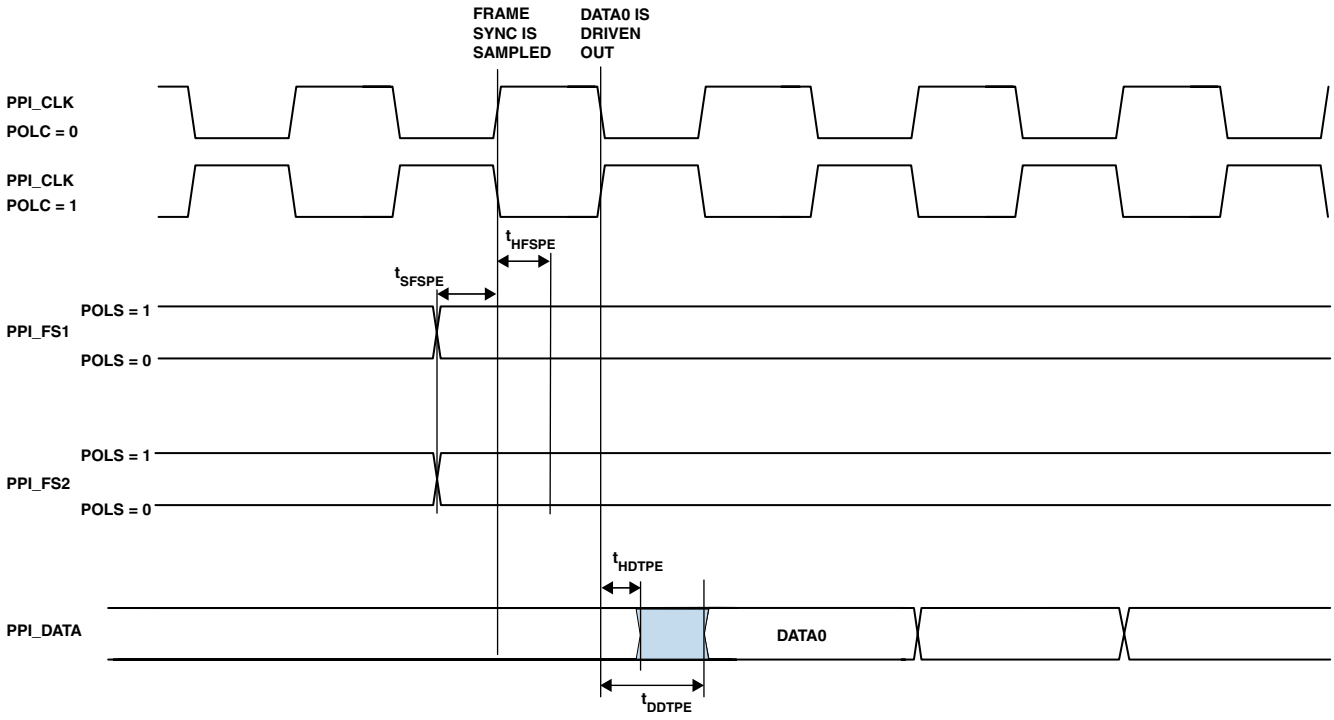

Figure 15. PPI GP Tx Mode with External Frame Sync Timing

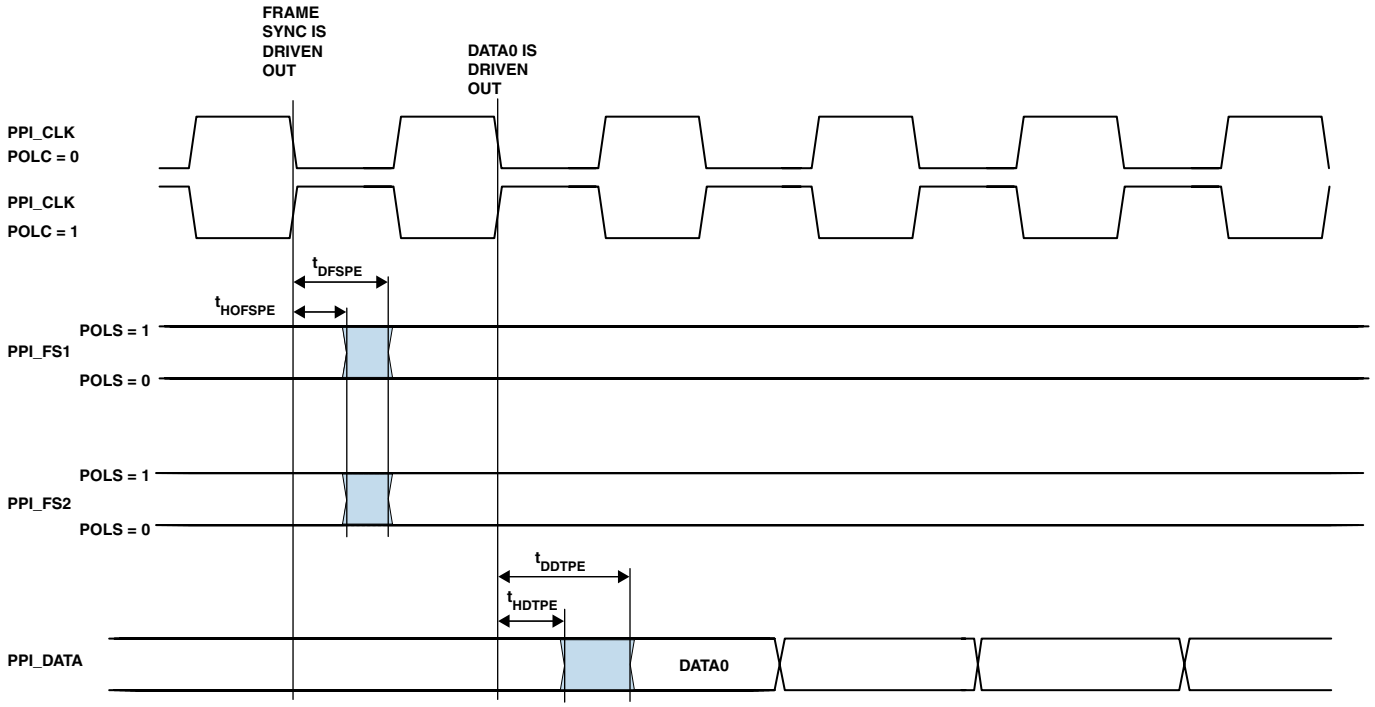

Figure 16. PPI GP Tx Mode with Internal Frame Sync Timing

Serial Ports

Table 22 on Page 31 through Table 25 on Page 33 and Figure 17 on Page 32 through Figure 19 on Page 34 describe Serial Port operations.

Table 22. Serial Ports—External Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSE}	TFS/RFS Setup Before TSCLK/RSCLK[1]	3.0		ns
t_{HFSE}	TFS/RFS Hold After TSCLK/RSCLK[1]	3.0		ns
t_{SDRE}	Receive Data Setup Before RSCLK[1]	3.0		ns
t_{HDRE}	Receive Data Hold After RSCLK[1]	3.0		ns
t_{SCLKW}	TSCLK/RSCLK Width	4.5		ns
t_{SCLK}	TSCLK/RSCLK Period	15.0		ns
Switching Characteristics				
t_{DFSE}	TFS/RFS Delay After TSCLK/RSCLK (Internally Generated TFS/RFS)[2]		10.0	ns
t_{HOFSE}	TFS/RFS Hold After TSCLK/RSCLK (Internally Generated TFS/RFS)[1]	0.0		ns
t_{DDTE}	Transmit Data Delay After TSCLK[1]		10.0	ns
t_{HDTE}	Transmit Data Hold After TSCLK[1]	0.0		ns

[1] Referenced to sample edge.
[2] Referenced to drive edge.

Table 23. Serial Ports—Internal Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSI}	TFS/RFS Setup Before TSCLK/RSCLK[1]	8.0		ns
t_{HFSI}	TFS/RFS Hold After TSCLK/RSCLK[1]	–2.0		ns
t_{SDRI}	Receive Data Setup Before RSCLK[1]	6.0		ns
t_{HDRI}	Receive Data Hold After RSCLK[1]	0.0		ns
t_{SCLKW}	TSCLK/RSCLK Width	4.5		ns
t_{SCLK}	TSCLK/RSCLK Period	15.0		ns
Switching Characteristics				
t_{DFSI}	TFS/RFS Delay After TSCLK/RSCLK (Internally Generated TFS/RFS)[2]		3.0	ns
t_{HOFSI}	TFS/RFS Hold After TSCLK/RSCLK (Internally Generated TFS/RFS)[1]	–1.0		ns
t_{DDTI}	Transmit Data Delay After TSCLK[1]		3.0	ns
t_{HDTI}	Transmit Data Hold After TSCLK[1]	–2.0		ns
t_{SCLKIW}	TSCLK/RSCLK Width	4.5		ns

[1] Referenced to sample edge.
[2] Referenced to drive edge.

Table 24. Serial Ports—Enable and Three-State

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

[1] Referenced to drive edge.

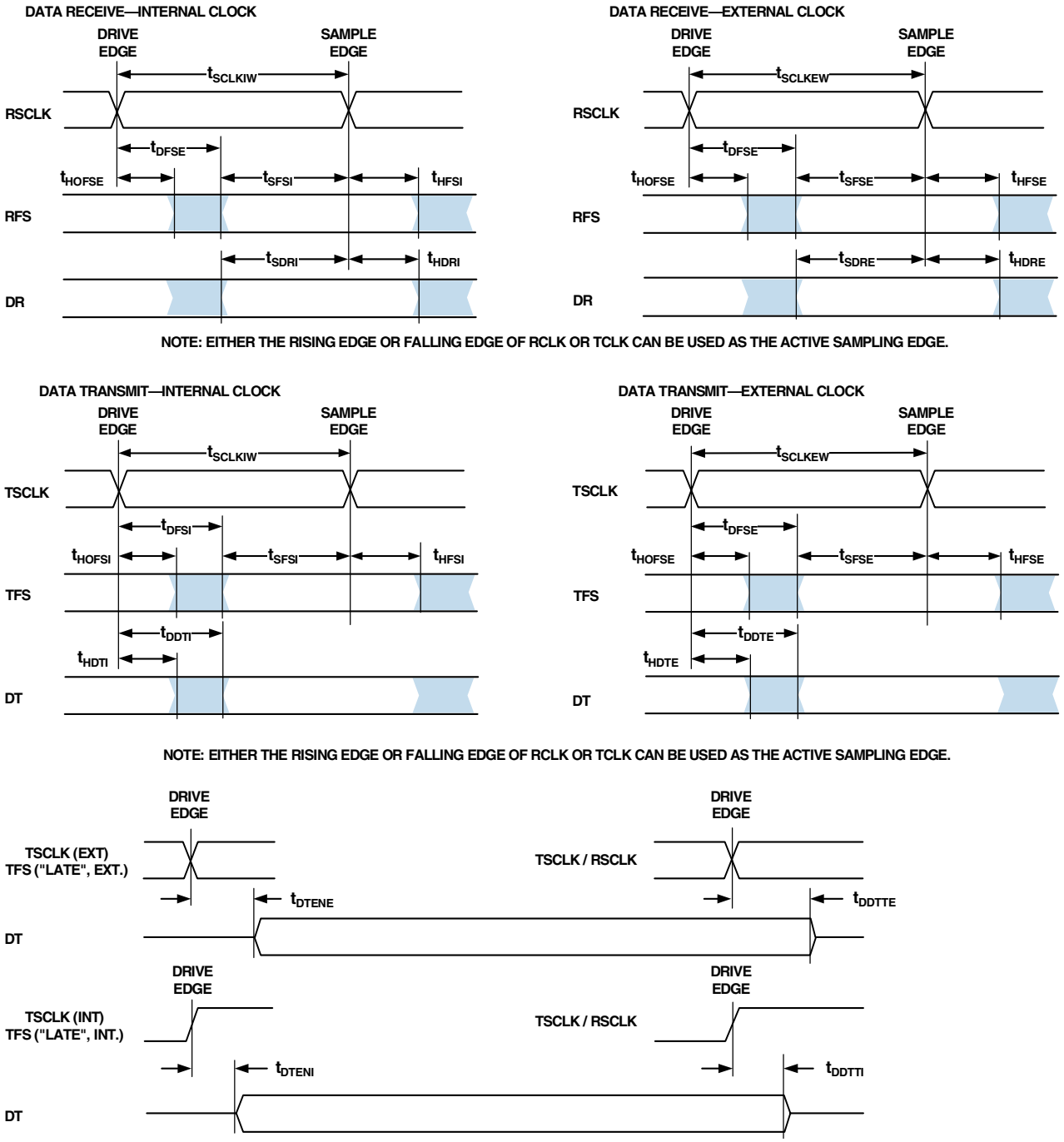

Figure 17. Serial Ports

Table 25. External Late Frame Sync

Parameter		Min	Max	Unit
Switching Characteristics				
$t_{DDTLFSE}$	Data Delay from Late External TFS or External RFS with MCE = 1, MFD = 0[1, 2]		10.0	ns
$t_{DTENLFS}$	Data Enable from Late FS or MCE = 1, MFD = 0[1, 2]	0		ns

[1] MCE = 1, TFS enable and TFS valid follow $t_{DTENLFS}$ and $t_{DDTLFSE}$.
[2] If external RFS/TFS setup to RSCLK/TSCLK > $t_{SCLKE}/2$, then $t_{DDTE/I}$ and $t_{DTENE/I}$ apply; otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

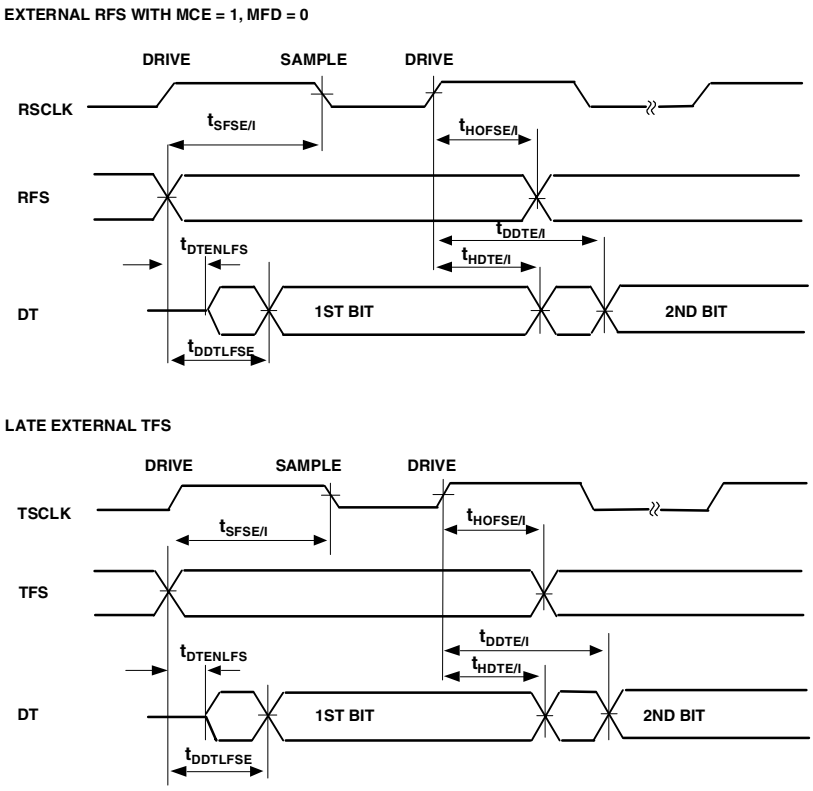

Figure 18. External Late Frame Sync (Frame Sync Setup < $t_{SCLK}/2$)

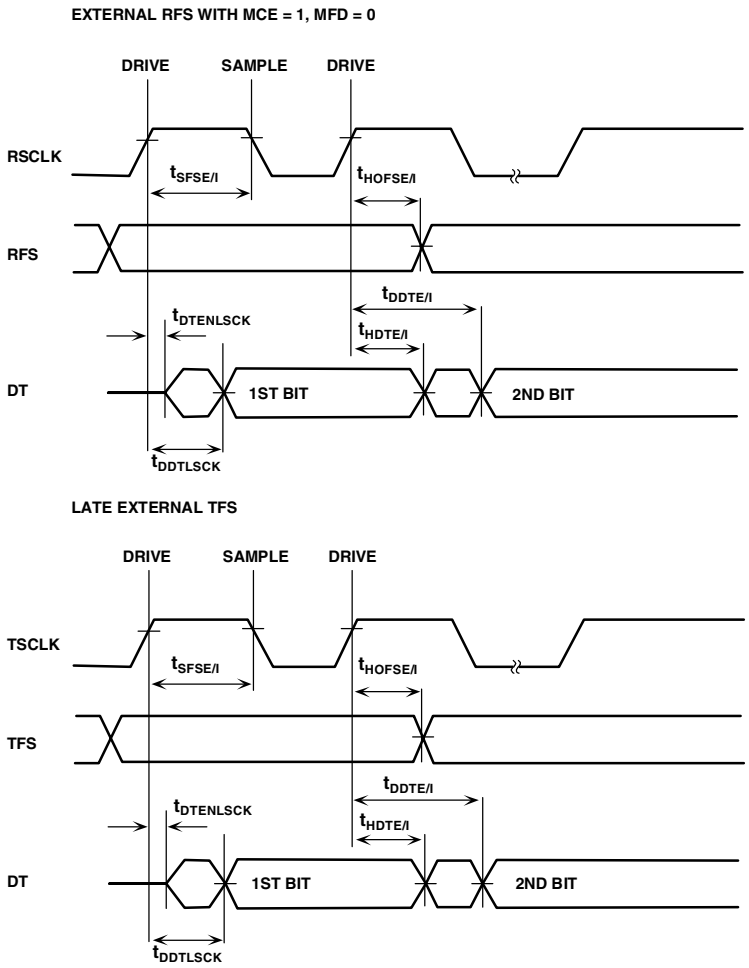

Figure 19. External Late Frame Sync (Frame Sync Setup > t_{SCLK}/2)

Serial Peripheral Interface (SPI) Port—Master Timing

Table 26 and Figure 20 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	$2t_{SCLK}$–1.5		ns
t_{SPICHM}	Serial Clock High Period	$2t_{SCLK}$–0.5		ns
t_{SPICLM}	Serial Clock Low Period	$2t_{SCLK}$–1.5		ns
t_{SPICLK}	Serial Clock Period	$4t_{SCLK}$–1.5		ns
t_{HDSM}	Last SCK Edge to $\overline{\text{SPISELx}}$ High	$2t_{SCLK}$–1.5		ns
t_{SPITDM}	Sequential Transfer Delay	$2t_{SCLK}$–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

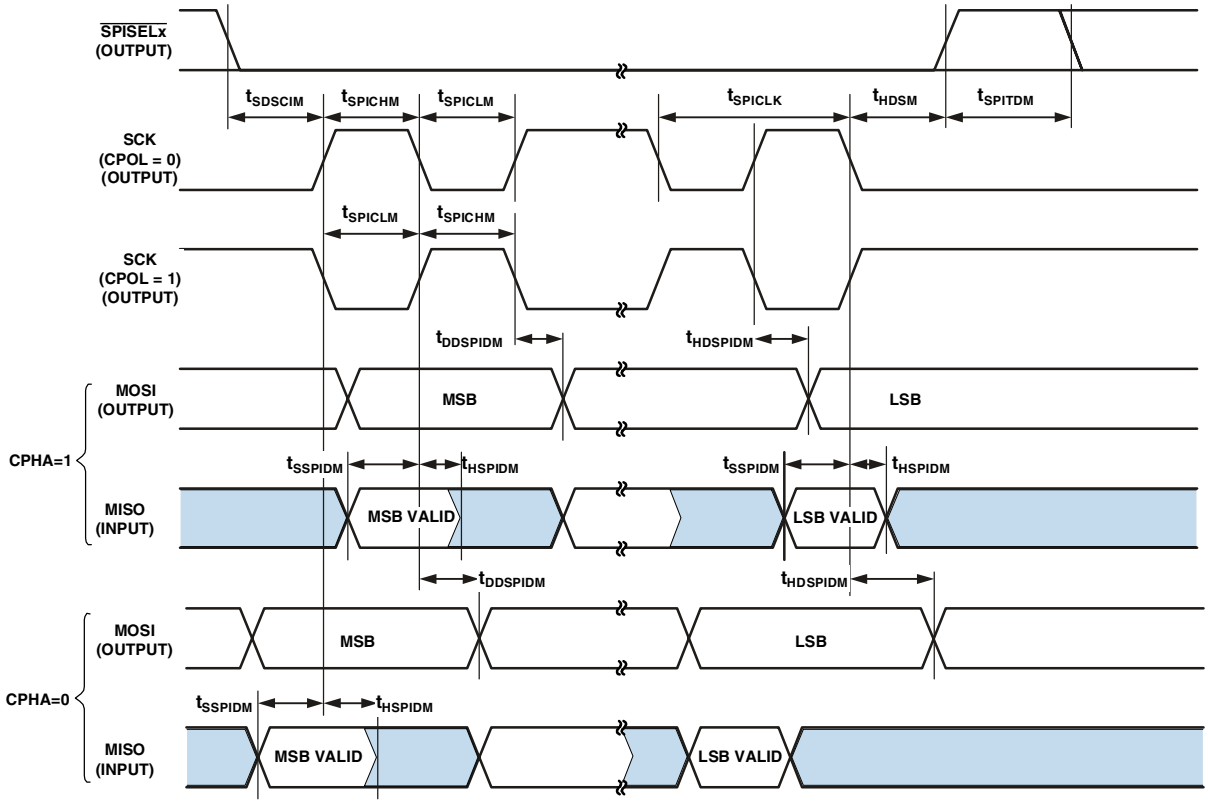

Figure 20. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port— Slave Timing

Table 27 and Figure 21 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	$2t_{SCLK}$–1.5		ns
t_{SPICLS}	Serial Clock Low Period	$2t_{SCLK}$–1.5		ns
t_{SPICLK}	Serial Clock Period	$4t_{SCLK}$–1.5		ns
t_{HDS}	Last SCK Edge to $\overline{SPISS}$ Not Asserted	$2t_{SCLK}$–1.5		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{SCLK}$–1.5		ns
t_{SDSCI}	$\overline{SPISS}$ Assertion to First SCK Edge	$2t_{SCLK}$–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}	$\overline{SPISS}$ Assertion to Data Out Active	0	8	ns
t_{DSDHI}	$\overline{SPISS}$ 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

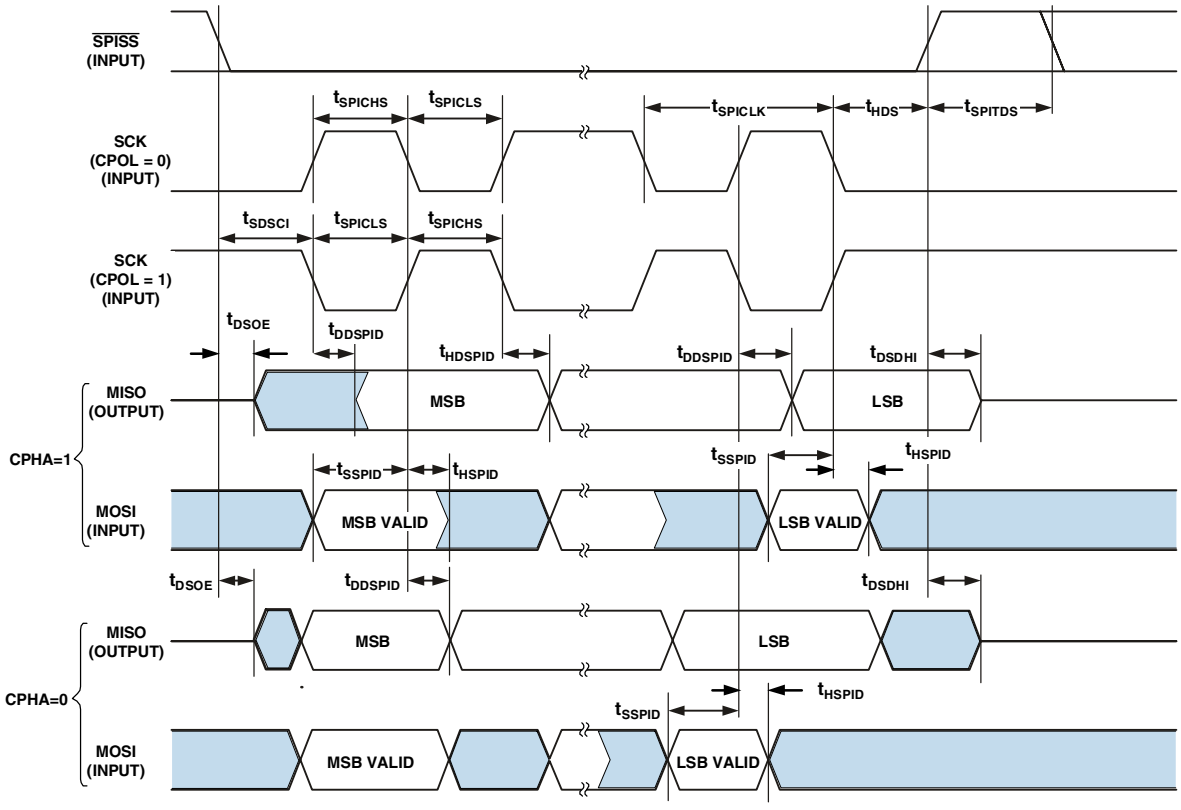

Figure 21. Serial Peripheral Interface (SPI) Port—Slave Timing

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

Figure 22 describes UART port receive and transmit operations. The maximum baud rate is SCLK/16. As shown in Figure 22, 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.

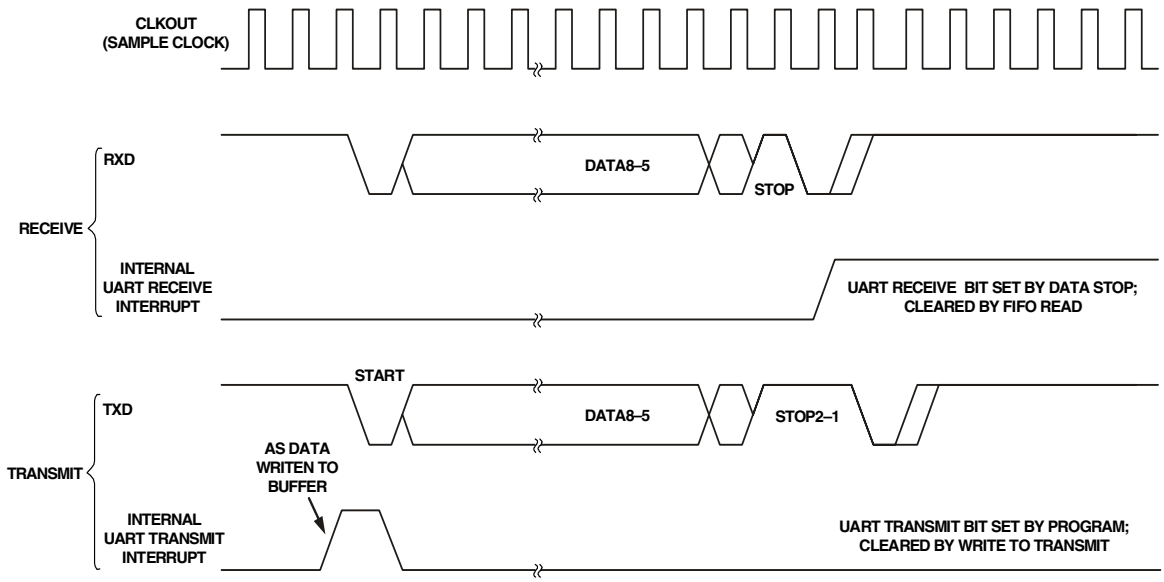

Figure 22. UART Port—Receive and Transmit Timing

Programmable Flags Cycle Timing

Table 28 and Figure 23 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

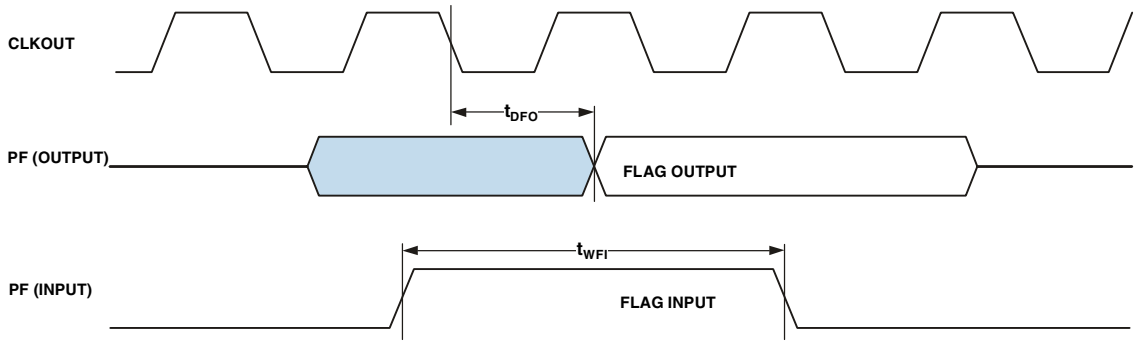

Figure 23. Programmable Flags Cycle Timing

Timer Cycle Timing

Table 29 and Figure 24 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 29. Timer Cycle Timing

Parameter		Min	Max	Unit
Timing Characteristics				
t_{WL}	Timer Pulse Width Input Low[1] (Measured in SCLK Cycles)	1		SCLK
t_{WH}	Timer Pulse Width Input High[1] (Measured in SCLK Cycles)	1		SCLK
Switching Characteristic				
t_{HTO}	Timer Pulse Width Output[2] (Measured in SCLK Cycles)	1	$(2^{32}-1)$	SCLK

[1] The minimum pulse-widths apply for TMRx input pins in width capture and external clock modes. They also apply to the PF1 or PPICLK input pins in PWM output mode.
[2] The minimum time for t_{HTO} is one cycle, and the maximum time for t_{HTO} equals $(2^{32}-1)$ cycles.

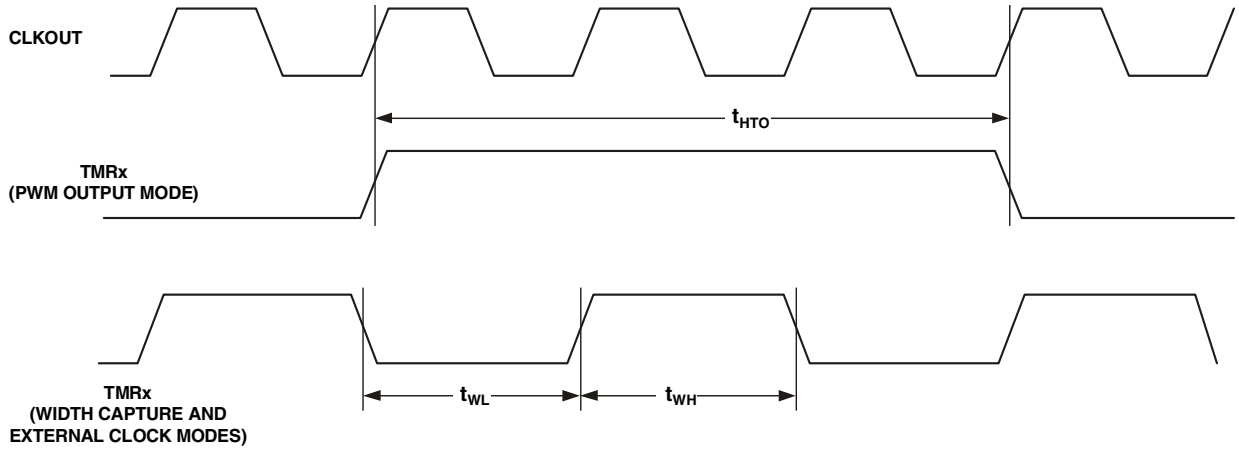

Figure 24. Timer PWM_OUT Cycle Timing

JTAG Test and Emulation Port Timing

Table 30 and Figure 25 describe JTAG port operations.

Table 30. JTAG Port Timing

Parameter		Min	Max	Unit
Timing Parameters				
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] (Measured in TCK Cycles)	4		TCK
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, TMR2–0, PF47–0, PPIx_CLK, RSCLK0–1, RFS0–1, DR0PRI, DR0SEC, TSCLK0–1, TFS0–1, DR1PRI, DR1SEC, MOSI, MISO, SCK, RX, $\overline{\text{RESET}}$, NMI0 and NMI1, BMODE1–0, $\overline{\text{BR}}$, 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, $\overline{\text{BG}}$, $\overline{\text{BGH}}$, PPIxD7–0.

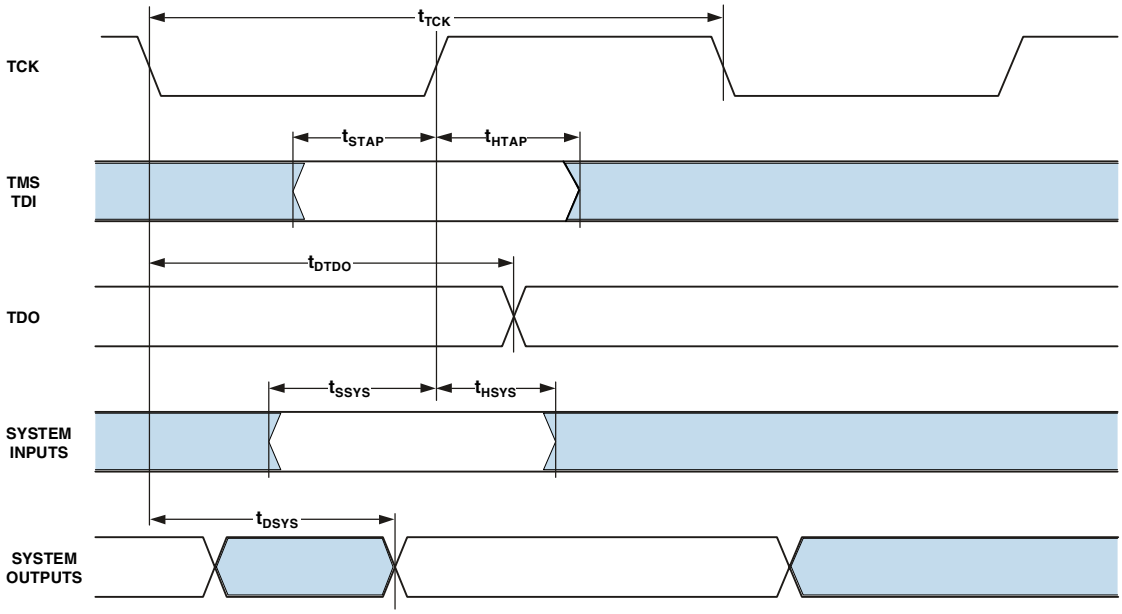

Figure 25. JTAG Port Timing

OUTPUT DRIVE CURRENTS

Figure 26 through Figure 33 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 on Page 17 to identify the driver type for a pin.

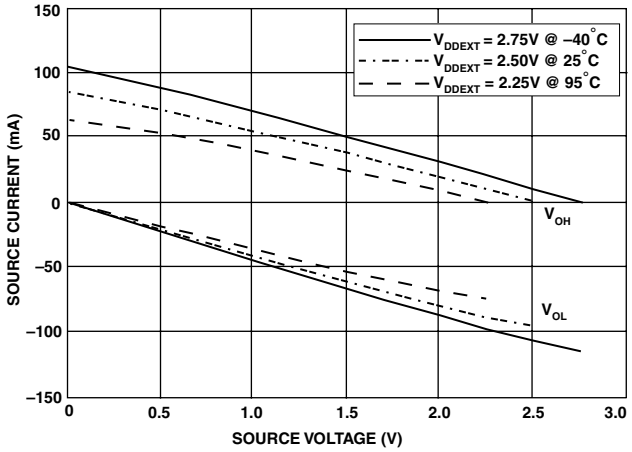

Figure 26. Drive Current A (Low V_{DDEXT})

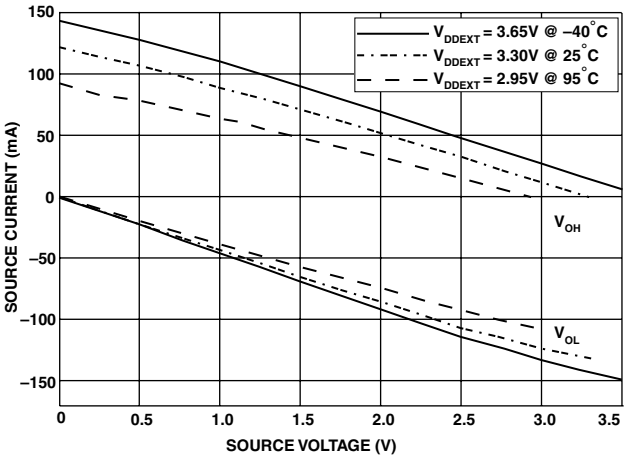

Figure 27. Drive Current A (High V_{DDEXT})

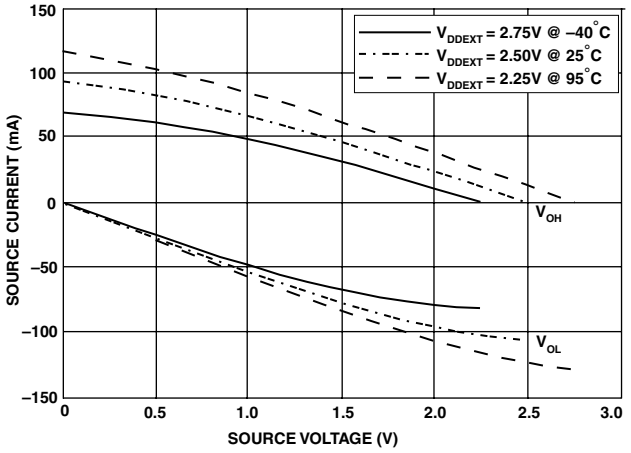

Figure 28. Drive Current B (Low V_{DDEXT})

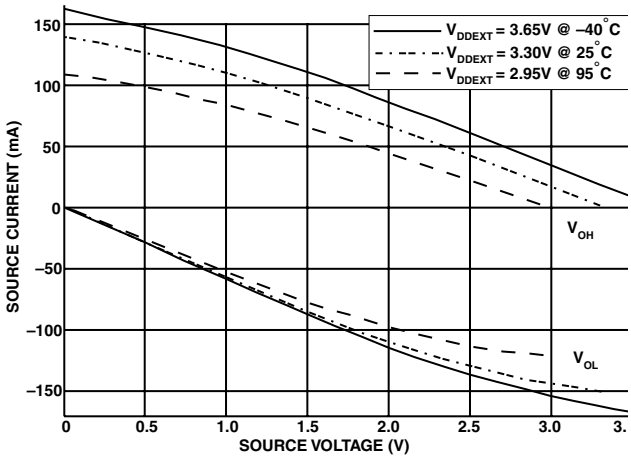

Figure 29. Drive Current B (High V_{DDEXT})

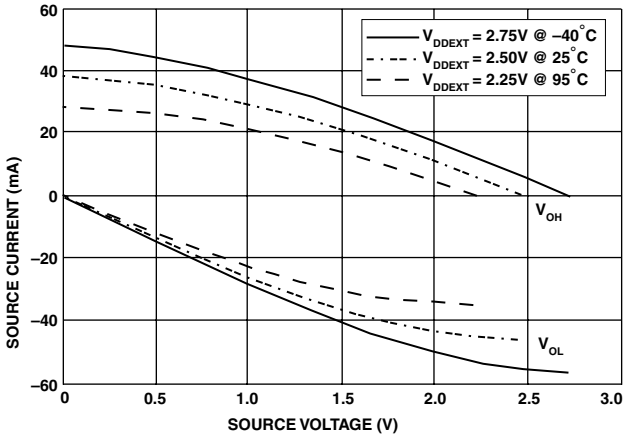

Figure 30. Drive Current C (Low V_{DDEXT})

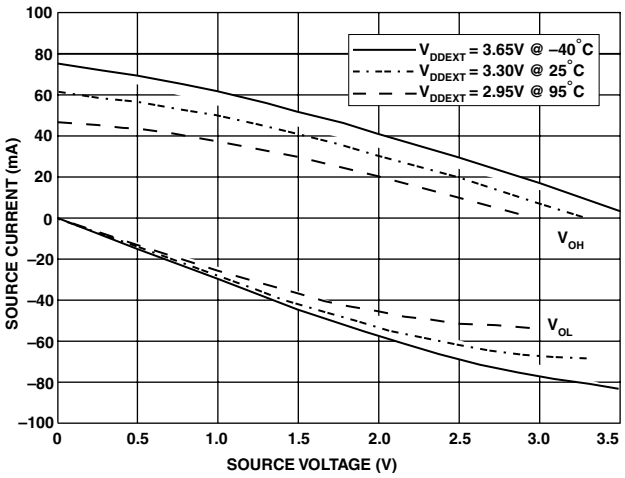

Figure 31. Drive Current C (High V_{DDEXT})

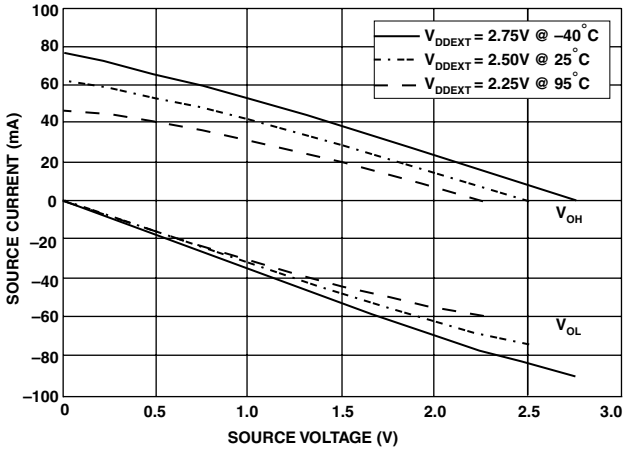

Figure 32. Drive Current D (Low V_{DDEXT})

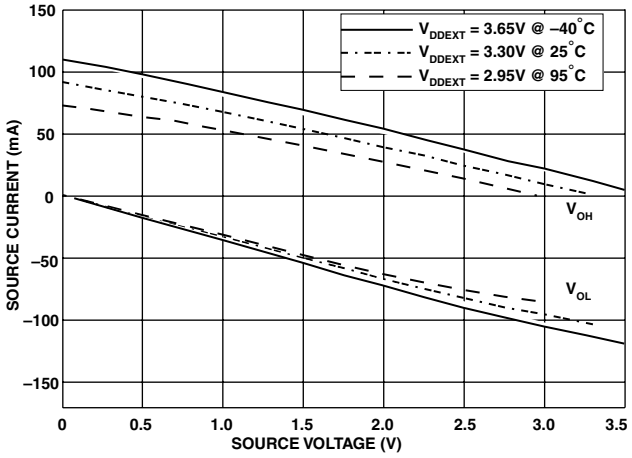

Figure 33. Drive Current D (High V_{DDEXT})

POWER DISSIPATION

Total power dissipation has two components: one due to internal circuitry (P_{INT}) and one due to the switching of external output drivers (P_{EXT}). Table 31 through Table 33 show the power dissipation for internal circuitry (V_{DDINT}).

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.

Many operating conditions can affect power dissipation. System designers should refer to *EE-293: Estimating Power for ADSP-BF561 Blackfin Processors* 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.

Table 31. Internal Power Dissipation (Hibernate mode)

	I_{DD} (nominal[1])	Unit
$I_{DDHIBERNATE}$[2]	50	µA

[1] Nominal assumes an operating temperature of 25°C.
[2] Measured at V_{DDEXT} = 3.65 V with voltage regulator off (V_{DDINT} = 0 V).

Table 32. Internal Power Dissipation (Deep Sleep mode)

V_{DDINT}[1]	I_{DD} (nominal[2])	Unit
0.8	32	mA
0.9	40	mA
1.0	50	mA
1.1	62	mA
1.25	84	mA
1.35	95	mA

[1] Assumes V_{DDINT} is regulated externally.
[2] Nominal assumes an operating temperature of 25°C.

Table 33. Internal Power Dissipation (Full On[1] mode)

V_{DDINT}[2] @ f_{CCLK}	I_{DD} (nominal[3])	Unit
0.8 @ 50 MHz	66	mA
0.8 @ 250 MHz	144	mA
0.9 @ 300 MHz	194	mA
1.0 @ 350 MHz	249	mA
1.1 @ 444 MHz	346	mA
1.25 @ 500 MHz	469	mA
1.35 @ 600 MHz	588	mA

[1] Processor executing 75% dual MAC, 25% ADD with moderate data bus activity.
[2] Assumes V_{DDINT} is regulated externally.
[3] Nominal assumes an operating temperature of 25°C.

TEST CONDITIONS

All timing parameters appearing in this data sheet were measured under the conditions described in this section. Figure 34 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.

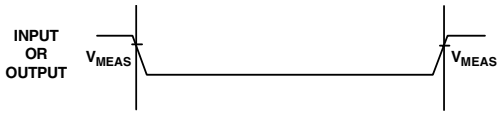

Figure 34. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

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 35 on Page 43.

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 35.

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

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 on Page 22 (for example t_{DSDAT} for an SDRAM write cycle as shown in SDRAM Interface Timing on Page 26).

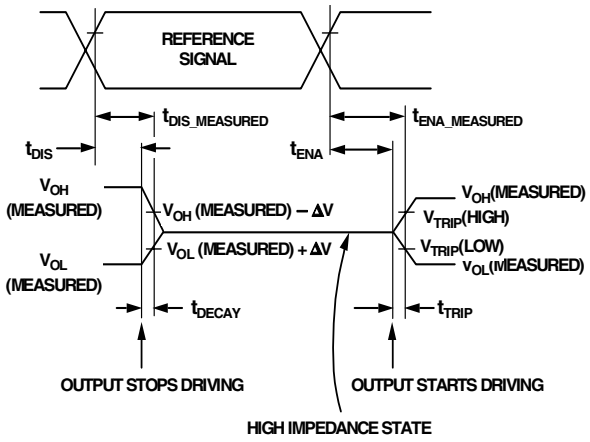

Figure 35. Output Enable/Disable

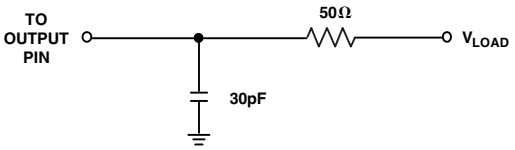

Figure 36. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

Capacitive Loading

Output delays and holds are based on standard capacitive loads: 30 pF on all pins (see Figure 36). V_{LOAD} is 1.5 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V. Figure 37 on Page 44 through Figure 44 on Page 45 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.

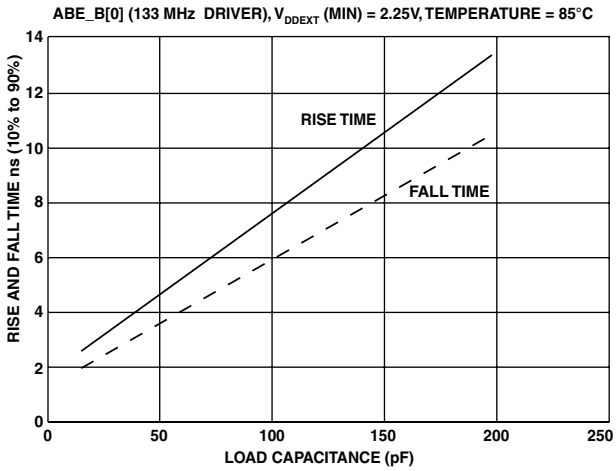

Figure 37. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver A at V_{DDEXT} (min)

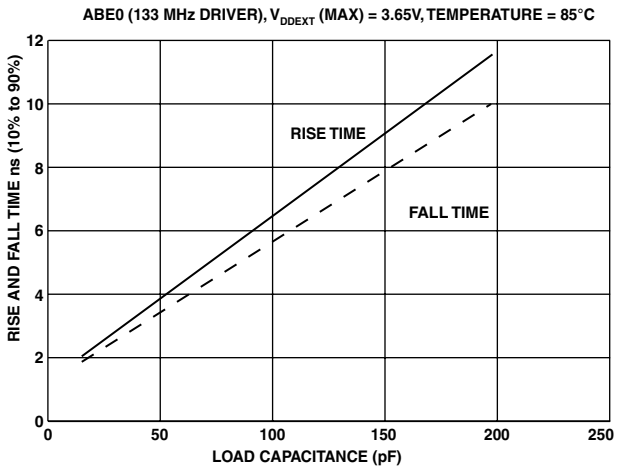

Figure 38. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver A at V_{DDEXT} (max)

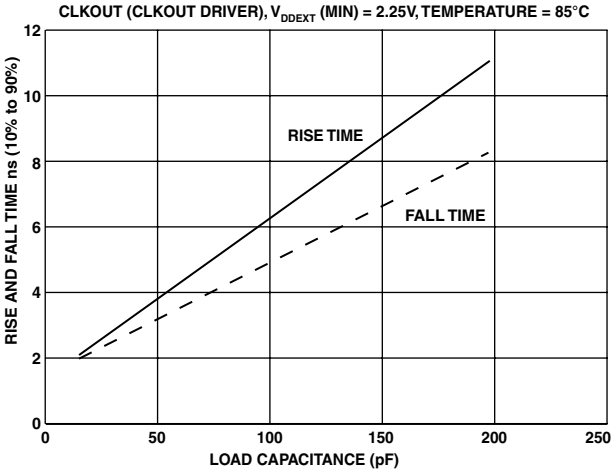

Figure 39. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver B at V_{DDEXT} (min)

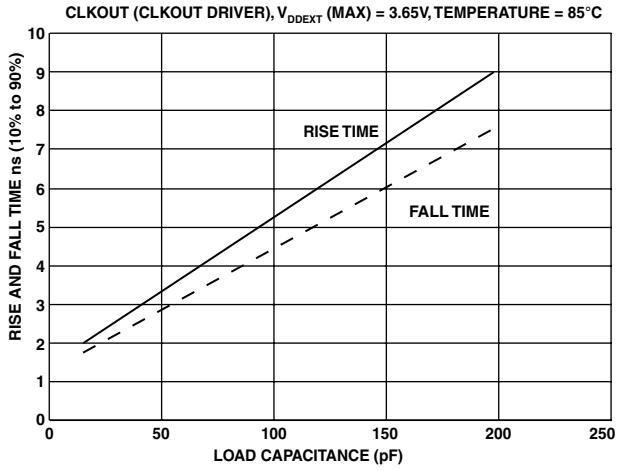

Figure 40. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver B at V_{DDEXT} (max)

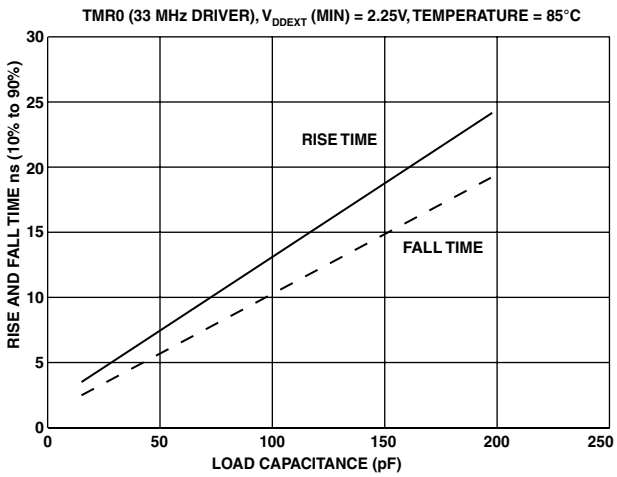

Figure 41. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver C at V_{DDEXT} (min)

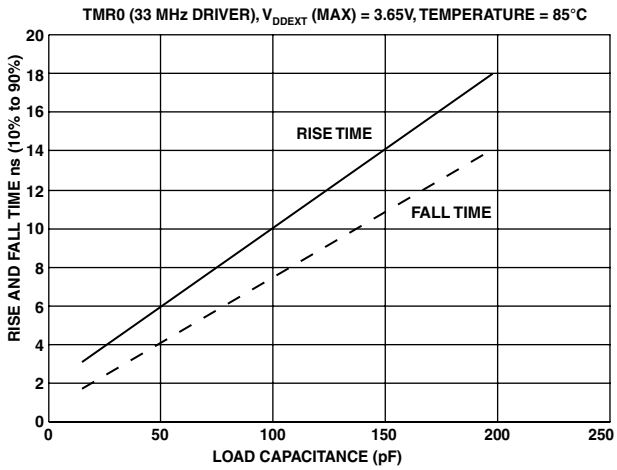

Figure 42. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver C at V_{DDEXT} (max)

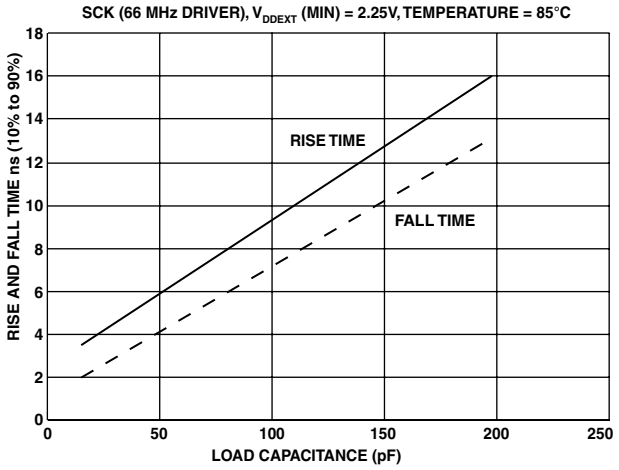

Figure 43. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver D at V_{DDEXT} (min)

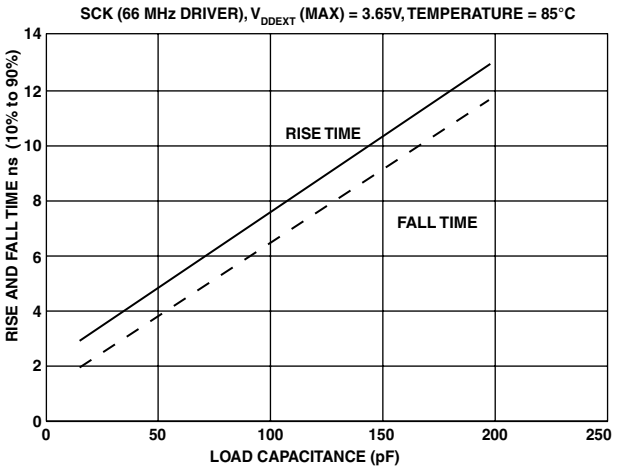

Figure 44. Typical Rise and Fall Times (10% to 90%) versus Load Capacitance for Driver D at V_{DDEXT} (max)

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 34 and Table 35.

P_D = power dissipation (see Power Dissipation on Page 42 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 34 and Table 35, 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 34 and Table 35 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 34. Thermal Characteristics for BC-256 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

Table 35. 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

256-BALL MBGA PINOUT

Table 36 lists the 256-Ball MBGA pinout by ball number.
Table 37 on Page 48 lists the 256-Ball MBGA pinout alphabetically by signal.

Table 36. 256-Ball MBGA Pin Assignment (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A01	VDDEXT	C09	$\overline{SMS2}$	F01	CLKIN	H09	GND	L01	PPI0D0
A02	ADDR24	C10	$\overline{SRAS}$	F02	VDDEXT	H10	GND	L02	PPI1SYNC2/TMR11
A03	ADDR20	C11	GND	F03	$\overline{RESET}$	H11	VDDINT	L03	GND
A04	VDDEXT	C12	$\overline{BGH}$	F04	PPI0D10/PF42	H12	DATA16	L04	PPI1SYNC3
A05	ADDR14	C13	GND	F05	ADDR21	H13	DATA18	L05	VDDEXT
A06	ADDR10	C14	ADDR07	F06	ADDR17	H14	DATA20	L06	PPI1D11/PF35
A07	$\overline{AMS3}$	C15	DATA1	F07	VDDINT	H15	DATA17	L07	GND
A08	$\overline{AWE}$	C16	DATA3	F08	GND	H16	DATA19	L08	VDDINT
A09	VDDEXT	D01	PPI0D13/PF45	F09	VDDINT	J01	VROUT0	L09	GND
A10	$\overline{SMS3}$	D02	PPI0D15/PF47	F10	GND	J02	VROUT1	L10	VDDEXT
A11	SCLK0/CLKOUT	D03	PPI0SYNC3	F11	ADDR08	J03	PPI0D2	L11	GND
A12	SCLK1	D04	ADDR23	F12	DATA10	J04	PPI0D3	L12	DR0PRI
A13	$\overline{BG}$	D05	GND	F13	DATA8	J05	PPI0D1	L13	TFS0/PF16
A14	$\overline{ABE2}$/SDQM2	D06	GND	F14	DATA12	J06	VDDEXT	L14	GND
A15	$\overline{ABE3}$/SDQM3	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/PF39
B02	ADDR22	D10	$\overline{SCAS}$	G02	GND	J10	VDDINT	M02	PPI1D13/PF37
B03	ADDR18	D11	SA10	G03	VDDEXT	J11	GND	M03	PPI1D9/PF33
B04	ADDR16	D12	VDDEXT	G04	BYPASS	J12	DATA30	M04	GND
B05	ADDR12	D13	ADDR02	G05	PPI0D14/PF46	J13	DATA22	M05	NC
B06	VDDEXT	D14	GND	G06	GND	J14	GND	M06	PF3/SPISEL3/TMR3
B07	$\overline{AMS1}$	D15	DATA5	G07	GND	J15	DATA21	M07	PF7/SPISEL7/TMR7
B08	$\overline{ARE}$	D16	DATA6	G08	GND	J16	DATA23	M08	VDDINT
B09	$\overline{SMS1}$	E01	GND	G09	VDDINT	K01	PPI0D6	M09	GND
B10	SCKE	E02	PPI0D11/PF43	G10	ADDR05	K02	PPI0D4	M10	BMODE0
B11	VDDEXT	E03	PPI0D12/PF44	G11	ADDR03	K03	PPI0D8/PF40	M11	SCK
B12	$\overline{BR}$	E04	PPI0SYNC1/TMR8	G12	DATA15	K04	PPI1SYNC1/TMR10	M12	DR1PRI
B13	$\overline{ABE1}$/SDQM1	E05	ADDR15	G13	DATA14	K05	PPI1D14/PF38	M13	NC
B14	ADDR06	E06	ADDR13	G14	GND	K06	VDDEXT	M14	VDDEXT
B15	ADDR04	E07	$\overline{AMS2}$	G15	DATA13	K07	GND	M15	DATA31
B16	DATA0	E08	VDDINT	G16	VDDEXT	K08	VDDINT	M16	DT0PRI/PF18
C01	PPI0SYNC2/TMR9	E09	$\overline{SMS0}$	H01	GND	K09	GND	N01	PPI1D12/PF36
C02	PPI0CLK	E10	$\overline{SWE}$	H02	GND	K10	GND	N02	PPI1D10/PF34
C03	ADDR25	E11	$\overline{ABE0}$/SDQM0	H03	PPI0D9/PF41	K11	VDDINT	N03	PPI1D3
C04	ADDR19	E12	DATA2	H04	PPI0D7	K12	DATA28	N04	PPI1D1
C05	GND	E13	GND	H05	PPI0D5	K13	DATA26	N05	PF1/SPISEL1/TMR1
C06	ADDR11	E14	DATA4	H06	VDDINT	K14	DATA24	N06	PF9
C07	$\overline{AOE}$	E15	DATA7	H07	VDDINT	K15	DATA25	N07	GND
C08	$\overline{AMS0}$	E16	VDDEXT	H08	GND	K16	VDDEXT	N08	PF13

Table 36. 256-Ball MBGA Pin 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/SPISEL5/TMR5	R02	PPI1D6	R14	TSCLK1/PF31	T10	TMS
N11	MOSI	P07	PF11	R03	PPI1D2	R15	DT1PRI/PF23	T11	SLEEP
N12	GND	P08	PF15/EXTCLK	R04	PPI1D0	R16	RFS0/PF19	T12	VDDEXT
N13	RFS1/PF24	P09	GND	R05	PF4/SPISEL4/TMR4	T01	VDDEXT	T13	RX/PF27
N14	GND	P10	$\overline{\text{TRST}}$	R06	PF8	T02	PPI1D4	T14	DR1SEC/PF25
N15	DT0SEC/PF17	P11	NMI0	R07	PF10	T03	VDDEXT	T15	DT1SEC/PF22
N16	TSCLK0/PF29	P12	GND	R08	PF14	T04	PF2/SPISEL2/TMR2	T16	VDDEXT
P01	PPI1D8/PF32	P13	RSCLK1/PF30	R09	NMI1	T05	PF6/SPISEL6/TMR6		
P02	GND	P14	TFS1/PF21	R10	TDI	T06	VDDEXT		
P03	PPI1D5	P15	RSCLK0/PF28	R11	$\overline{\text{EMU}}$	T07	PF12		
P04	PF0/SPISS/TMR0	P16	DR0SEC/PF20	R12	MISO	T08	VDDEXT		

Table 37. 256-Ball MBGA Pin Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
ABE0/SDQM0	E11	BR	B12	DT0SEC/PF17	N15	GND	N14
ABE1/SDQM1	B13	BYPASS	G04	DT1PRI/PF23	R15	GND	P02
ABE2/SDQM2	A14	CLKIN	F01	DT1SEC/PF22	T15	GND	P05
ABE3/SDQM3	A15	DATA0	B16	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/SPISS/TMR0	P04
ADDR10	A06	DATA9	F15	GND	E13	PF1/SPISEL1/TMR1	N05
ADDR11	C06	DATA10	F12	GND	F08	PF2/SPISEL2/TMR2	T04
ADDR12	B05	DATA11	F16	GND	F10	PF3/SPISEL3/TMR3	M06
ADDR13	E06	DATA12	F14	GND	G02	PF4/SPISEL4/TMR4	R05
ADDR14	A05	DATA13	G15	GND	G06	PF5/SPISEL5/TMR5	P06
ADDR15	E05	DATA14	G13	GND	G07	PF6/SPISEL6/TMR6	T05
ADDR16	B04	DATA15	G12	GND	G08	PF7/SPISEL7/TMR7	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/EXTCLK	P08
ADDR25	C03	DATA24	K14	GND	J14	PPI0CLK	C02
AMS0	C08	DATA25	K15	GND	K07	PPI0D0	L01
AMS1	B07	DATA26	K13	GND	K09	PPI0D1	J05
AMS2	E07	DATA27	L15	GND	K10	PPI0D2	J03
AMS3	A07	DATA28	K12	GND	L03	PPI0D3	J04
AOE	C07	DATA29	L16	GND	L07	PPI0D4	K02
ARDY	D09	DATA30	J12	GND	L09	PPI0D5	H05
ARE	B08	DATA31	M15	GND	L11	PPI0D6	K01
AWE	A08	DR0PRI	L12	GND	L14	PPI0D7	H04
BG	A13	DR0SEC/PF20	P16	GND	M04	PPI0D8/PF40	K03
BGH	C12	DR1PRI	M12	GND	M09	PPI0D9/PF41	H03
BMODE0	M10	DR1SEC/PF25	T14	GND	N07	PPI0D10/PF42	F04
BMODE1	N10	DT0PRI/PF18	M16	GND	N12	PPI0D11/PF43	E02

Table 37. 256-Ball MBGA Pin Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
PPI0D12/PF44	E03	PPI1SYNC1/TMR10	K04	TDO	N09	VDDEXT	M14
PPI0D13/PF45	D01	PPI1SYNC2/TMR11	L02	TFS0/PF16	L13	VDDEXT	T01
PPI0D14/PF46	G05	PPI1SYNC3	L04	TFS1/PF21	P14	VDDEXT	T03
PPI0D15/PF47	D02	$\overline{\text{RESET}}$	F03	TMS	T10	VDDEXT	T06
PPI0SYNC1/TMR8	E04	RFS0/PF19	R16	$\overline{\text{TRST}}$	P10	VDDEXT	T08
PPI0SYNC2/TMR9	C01	RFS1/PF24	N13	TSCLK0/PF29	N16	VDDEXT	T12
PPI0SYNC3	D03	RSCLK0/PF28	P15	TSCLK1/PF31	R14	VDDEXT	T16
PPI1CLK	B01	RSCLK1/PF30	P13	TX/PF26	R13	VDDINT	E08
PPI1D0	R04	RX/PF27	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/CLKOUT	A11	VDDEXT	B11	VDDINT	H11
PPI1D6	R02	SCLK1	A12	VDDEXT	D12	VDDINT	J08
PPI1D7	R01	SLEEP	T11	VDDEXT	E16	VDDINT	J09
PPI1D8/PF32	P01	$\overline{\text{SMS0}}$	E09	VDDEXT	F02	VDDINT	J10
PPI1D9/PF33	M03	$\overline{\text{SMS1}}$	B09	VDDEXT	G03	VDDINT	K08
PPI1D10/PF34	N02	$\overline{\text{SMS2}}$	C09	VDDEXT	G16	VDDINT	K11
PPI1D11/PF35	L06	$\overline{\text{SMS3}}$	A10	VDDEXT	J06	VDDINT	L08
PPI1D12/PF36	N01	$\overline{\text{SRAS}}$	C10	VDDEXT	K06	VDDINT	M08
PPI1D13/PF37	M02	$\overline{\text{SWE}}$	E10	VDDEXT	K16	VROUT0	J01
PPI1D14/PF38	K05	TCK	T09	VDDEXT	L05	VROUT1	J02
PPI1D15/PF39	M01	TDI	R10	VDDEXT	L10	XTAL	G01

Figure 45 lists the top view of the 256-Ball MBGA ball configuration. Figure 46 lists the bottom view of the 256-Ball MBGA ball configuration.

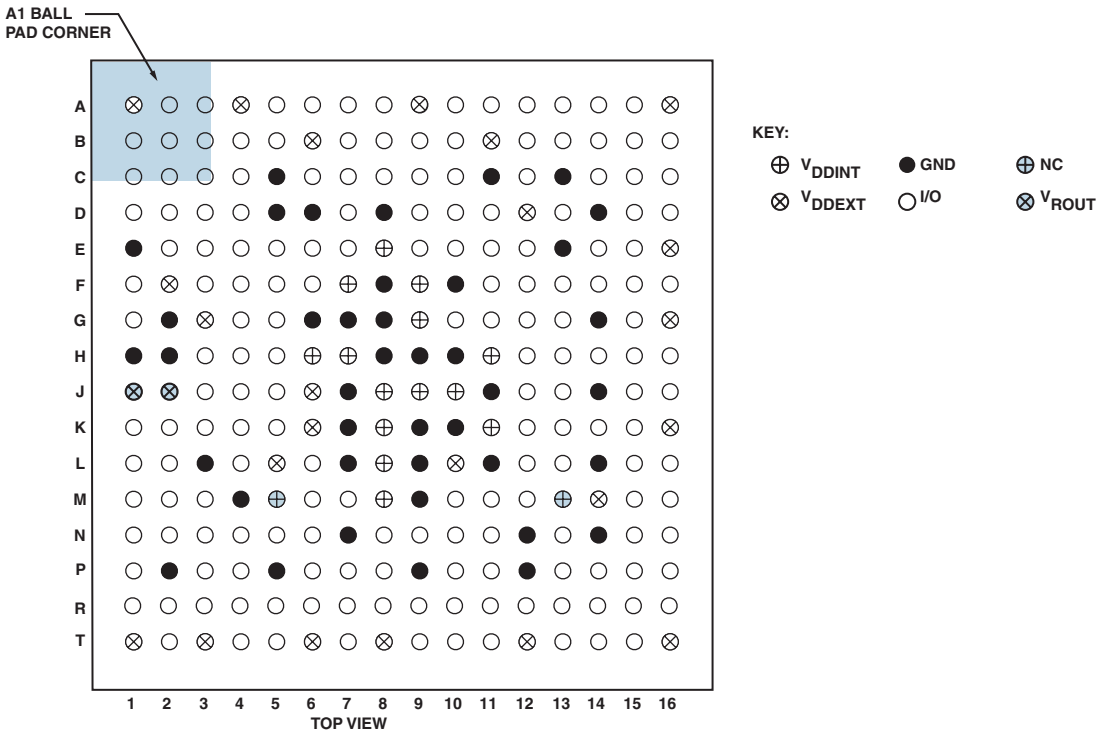

Figure 45. 256-Ball MBGA Ball Configuration (Top View)

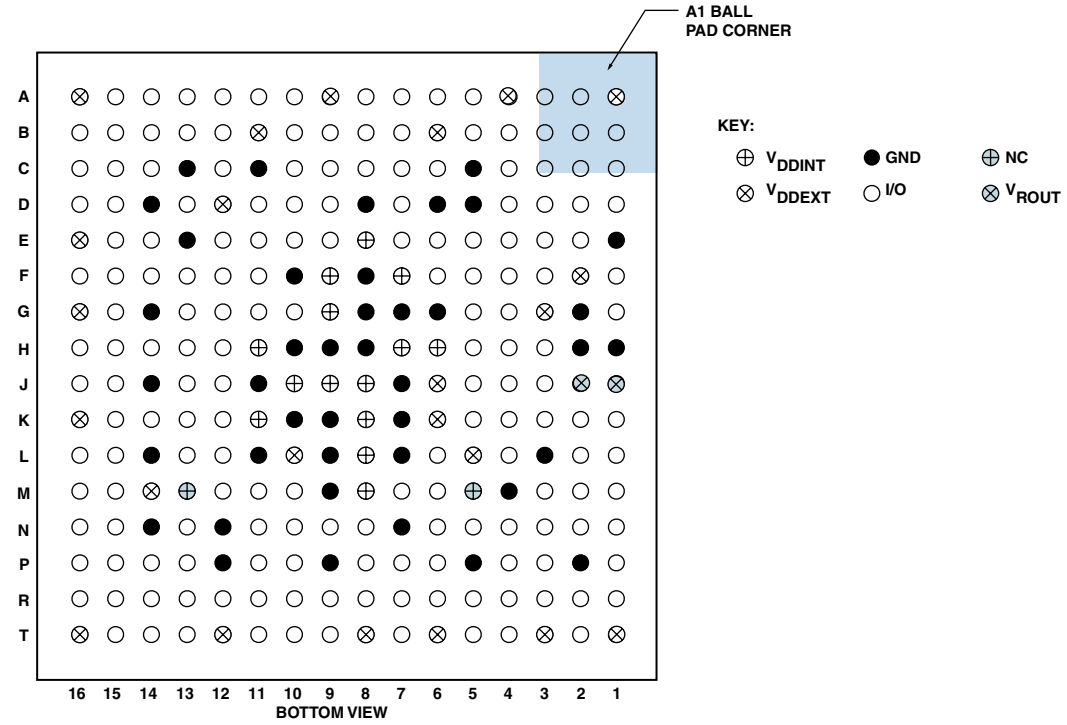

Figure 46. 256-Ball MBGA Ball Configuration (Bottom View)

297-BALL PBGA PINOUT

Table 38 lists the 297-Ball PBGA pinout numerically by ball number. Table 39 on Page 53 lists the 297-Ball PBGA pinout alphabetically by signal.

Table 38. 297-Ball PBGA Pin 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/PF43	L14	GND
A02	ADDR25	B16	$\overline{\text{SMS3}}$	G02	PPI0D10/PF42	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	$\overline{\text{ABE1}}$/SDQM1	H26	DATA9	M01	VROUT0
A09	ADDR11	B23	$\overline{\text{ABE3}}$/SDQM3	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/CLKOUT	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	$\overline{\text{ABE0}}$/SDQM0	C26	ADDR03	K01	XTAL	N02	PPI0D9/PF41
A23	$\overline{\text{ABE2}}$/SDQM2	D01	PPI0SYNC1/TMR8	K02	NC	N10	VDDEXT
A24	ADDR08	D02	PPI0SYNC2/TMR9	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/PF47	K17	VDDINT	N18	VDDINT
B06	ADDR18	E02	PPI0D14/PF46	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/PF40
B10	ADDR10	E26	DATA3	L02	NC	P10	VDDEXT
B11	$\overline{\text{AMS2}}$	F01	PPI0D13/PF45	L10	VDDEXT	P11	GND
B12	$\overline{\text{AMS0}}$	F02	PPI0D12/PF44	L11	GND	P12	GND
B13	$\overline{\text{AOE}}$	F25	DATA2	L12	GND	P13	GND
B14	ARDY	F26	DATA5	L13	GND	P14	GND

Table 38. 297-Ball PBGA Pin 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/PF27
P16	GND	U12	VDDEXT	AC23	GND	AE22	RFS1/PF24
P17	GND	U13	VDDEXT	AC24	GND	AE23	DR1SEC/PF25
P18	VDDINT	U14	GND	AC25	DR0SEC/PF20	AE24	TFS1/PF21
P25	DATA18	U15	VDDINT	AC26	RFS0/PF19	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/SPISEL1/TMR1
R13	GND	V02	PPI0D0	AD23	GND	AF06	PF3/SPISEL3/TMR3
R14	GND	V25	DATA26	AD24	GND	AF07	PF5/SPISEL5/TMR5
R15	GND	V26	DATA29	AD25	NC	AF08	PF7/SPISEL7/TMR7
R16	GND	W01	PPI1SYNC1/TMR10	AD26	RSCLK0/PF28	AF09	PF9
R17	GND	W02	PPI1SYNC2/TMR11	AE01	PPI1D5	AF10	PF11
R18	VDDINT	W25	DATA28	AE02	GND	AF11	PF13
R25	DATA20	W26	DATA31	AE03	PPI1D3	AF12	PF15/EXT CLK
R26	DATA23	Y01	PPI1D15/PF39	AE04	PPI1D1	AF13	NMI1
T01	PPI0D3	Y02	PPI1D14/PF38	AE05	PF0/SPISS/TMR0	AF14	TCK
T02	PPI0D4	Y25	DATA30	AE06	PF2/SPISEL2/TMR2	AF15	TDI
T10	VDDEXT	Y26	DT0PRI/PF18	AE07	PF4/SPISEL4/TMR4	AF16	TMS
T11	GND	AA01	PPI1D13/PF37	AE08	PF6/SPISEL6/TMR6	AF17	SLEEP
T12	GND	AA02	PPI1D12/PF36	AE09	PF8	AF18	NMI0
T13	GND	AA25	DT0SEC/PF17	AE10	PF10	AF19	SCK
T14	GND	AA26	TSCLK0/PF29	AE11	PF12	AF20	TX/PF26
T15	GND	AB01	PPI1D11/PF35	AE12	PF14	AF21	RSCLK1/PF30
T16	GND	AB02	PPI1D10/PF34	AE13	NC	AF22	DR1PRI
T17	GND	AB03	GND	AE14	TDO	AF23	TSCLK1/PF31
T18	VDDINT	AB24	GND	AE15	$\overline{\text{TRST}}$	AF24	DT1SEC/PF22
T25	DATA22	AB25	TFS0/PF16	AE16	$\overline{\text{EMU}}$	AF25	DT1PRI/PF23
T26	DATA25	AB26	DR0PRI	AE17	BMODE1	AF26	GND
U01	PPI0D1	AC01	PPI1D9/PF33	AE18	BMODE0		
U02	PPI0D2	AC02	PPI1D8/PF32	AE19	MISO		
U10	VDDEXT	AC03	GND	AE20	MOSI		

Table 39. 297-Ball PBGA Pin Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
ABE0/SDQM0	A22	BR	B20	DT0SEC/PF17	AA25	GND	N15
ABE1/SDQM1	B22	BYPASS	H01	DT1PRI/PF23	AF25	GND	N16
ABE2/SDQM2	A23	CLKIN	J01	DT1SEC/PF22	AF24	GND	N17
ABE3/SDQM3	B23	DATA0	E25	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
AMS0	B12	DATA25	T26	GND	M02	GND	AC04
AMS1	A12	DATA26	V25	GND	M11	GND	AC23
AMS2	B11	DATA27	U26	GND	M12	GND	AC24
AMS3	A11	DATA28	W25	GND	M13	GND	AD03
AOE	B13	DATA29	V26	GND	M14	GND	AD04
ARDY	B14	DATA30	Y25	GND	M15	GND	AD05
ARE	A14	DATA31	W26	GND	M16	GND	AD22
AWE	A13	DR0PRI	AB26	GND	M17	GND	AD23
BG	B21	DR0SEC/PF20	AC25	GND	N11	GND	AD24
BGH	A21	DR1PRI	AF22	GND	N12	GND	AE02
BMODE0	AE18	DR1SEC/PF25	AE23	GND	N13	GND	AE25
BMODE1	AE17	DT0PRI/PF18	Y26	GND	N14	GND	AF01

Table 39. 297-Ball PBGA Pin Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
GND	AF26	PPI0D7	P01	RSCLK0/PF28	AD26	VDDEXT	K13
MISO	AE19	PPI0D8/PF40	P02	RSCLK1/PF30	AF21	VDDEXT	K14
MOSI	AE20	PPI0D9/PF41	N02	RX/PF27	AE21	VDDEXT	K15
NC	K02	PPI0D10/PF42	G02	SA10	B19	VDDEXT	L10
NC	L01	PPI0D11/PF43	G01	$\overline{\text{SCAS}}$	A18	VDDEXT	M10
NC	L02	PPI0D12/PF44	F02	SCK	AF19	VDDEXT	N10
NC	AD25	PPI0D13/PF45	F01	SCKE	B17	VDDEXT	P10
NC	AE13	PPI0D14/PF46	E02	SCLK0/CLKOUT	A19	VDDEXT	R10
NC	AE26	PPI0D15/PF47	E01	SCLK1	A20	VDDEXT	T10
NMI0	AF18	PPI0SYNC1/TMR8	D01	SLEEP	AF17	VDDEXT	U10
NMI1	AF13	PPI0SYNC2/TMR9	D02	$\overline{\text{SMS0}}$	A15	VDDEXT	U11
PF0/SPISS/TMR0	AE05	PPI0SYNC3	C01	$\overline{\text{SMS1}}$	B15	VDDEXT	U12
PF1/SPISEL1/TMR1	AF05	PPI1CLK	B01	$\overline{\text{SMS2}}$	A16	VDDEXT	U13
PF2/SPISEL2/TMR2	AE06	PPI1D0	AF04	$\overline{\text{SMS3}}$	B16	VDDINT	J16
PF3/SPISEL3/TMR3	AF06	PPI1D1	AE04	$\overline{\text{SRAS}}$	A17	VDDINT	J17
PF4/SPISEL4/TMR4	AE07	PPI1D2	AF03	$\overline{\text{SWE}}$	B18	VDDINT	J18
PF5/SPISEL5/TMR5	AF07	PPI1D3	AE03	TCK	AF14	VDDINT	K16
PF6/SPISEL6/TMR6	AE08	PPI1D4	AF02	TDI	AF15	VDDINT	K17
PF7/SPISEL7/TMR7	AF08	PPI1D5	AE01	TDO	AE14	VDDINT	K18
PF8	AE09	PPI1D6	AD02	TFS0/PF16	AB25	VDDINT	L18
PF9	AF09	PPI1D7	AD01	TFS1/PF21	AE24	VDDINT	M18
PF10	AE10	PPI1D8/PF32	AC02	TMS	AF16	VDDINT	N18
PF11	AF10	PPI1D9/PF33	AC01	$\overline{\text{TRST}}$	AE15	VDDINT	P18
PF12	AE11	PPI1D10/PF34	AB02	TSCLK0/PF29	AA26	VDDINT	R18
PF13	AF11	PPI1D11/PF35	AB01	TSCLK1/PF31	AF23	VDDINT	T18
PF14	AE12	PPI1D12/PF36	AA02	TX/PF26	AF20	VDDINT	U15
PF15/EXT CLK	AF12	PPI1D13/PF37	AA01	VDDEXT	J10	VDDINT	U16
PPI0CLK	C02	PPI1D14/PF38	Y02	VDDEXT	J11	VDDINT	U17
PPI0D0	V02	PPI1D15/PF39	Y01	VDDEXT	J12	VDDINT	U18
PPI0D1	U01	PPI1SYNC1/TMR10	W01	VDDEXT	J13	VROUT0	M01
PPI0D2	U02	PPI1SYNC2/TMR11	W02	VDDEXT	J14	VROUT1	N01
PPI0D3	T01	PPI1SYNC3	V01	VDDEXT	J15	XTAL	K01
PPI0D4	T02	$\overline{\text{RESET}}$	H02	VDDEXT	K10		
PPI0D5	R01	RFS0/PF19	AC26	VDDEXT	K11		
PPI0D6	R02	RFS1/PF24	AE22	VDDEXT	K12		

Figure 47 lists the top view of the 297-Ball PBGA ball configuration. Figure 48 lists the bottom view of the 297-Ball PBGA ball configuration.

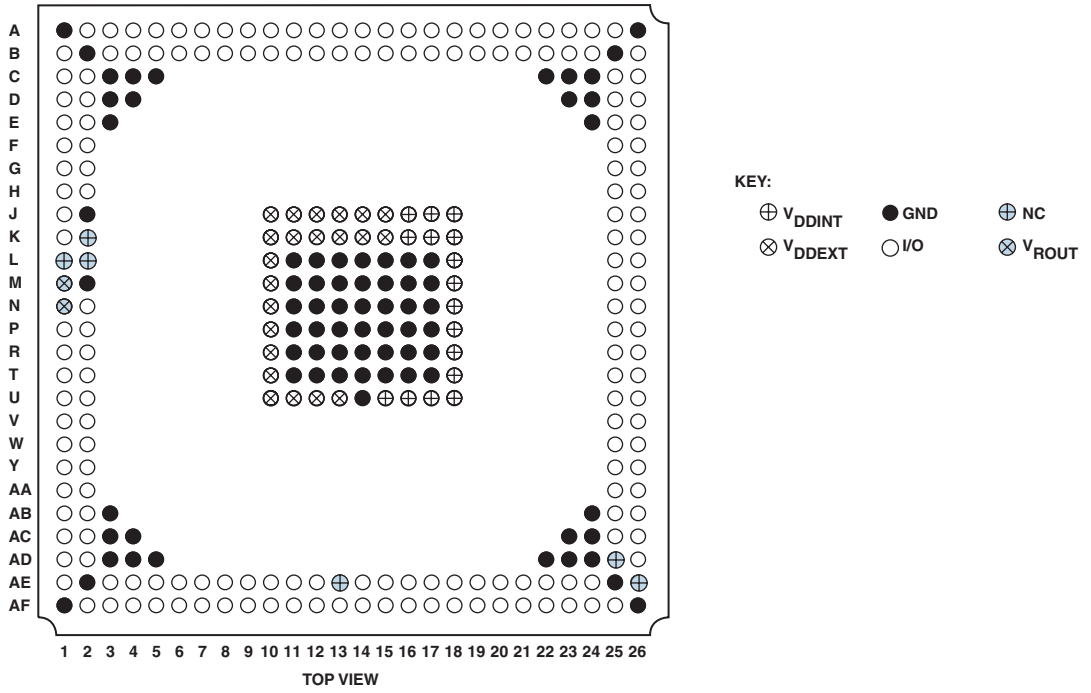

Figure 47. 297-Ball PBGA Ball Configuration (Top View)

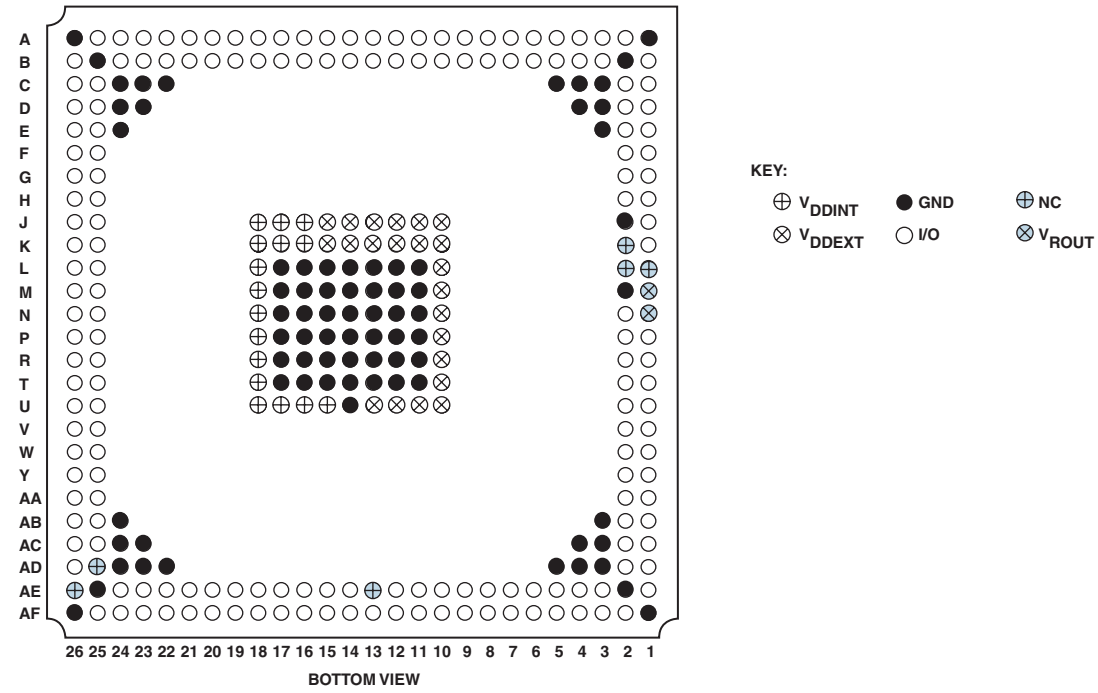

Figure 48. 297-Ball PBGA Ball Configuration (Bottom View)

OUTLINE DIMENSIONS

Dimensions in the outline dimension figures are shown in millimeters.

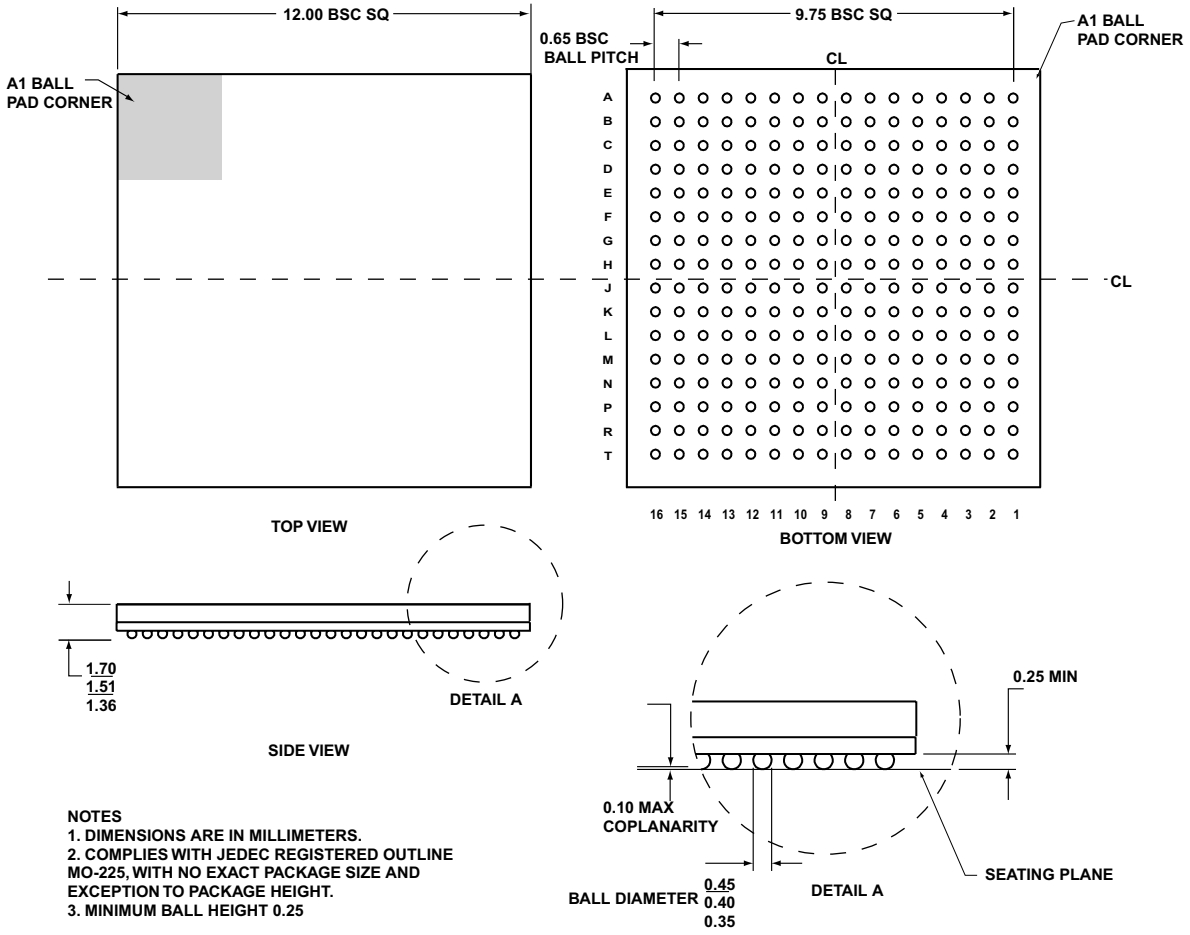

Figure 49. 256-Ball Mini-Ball Grid Array (BC-256)

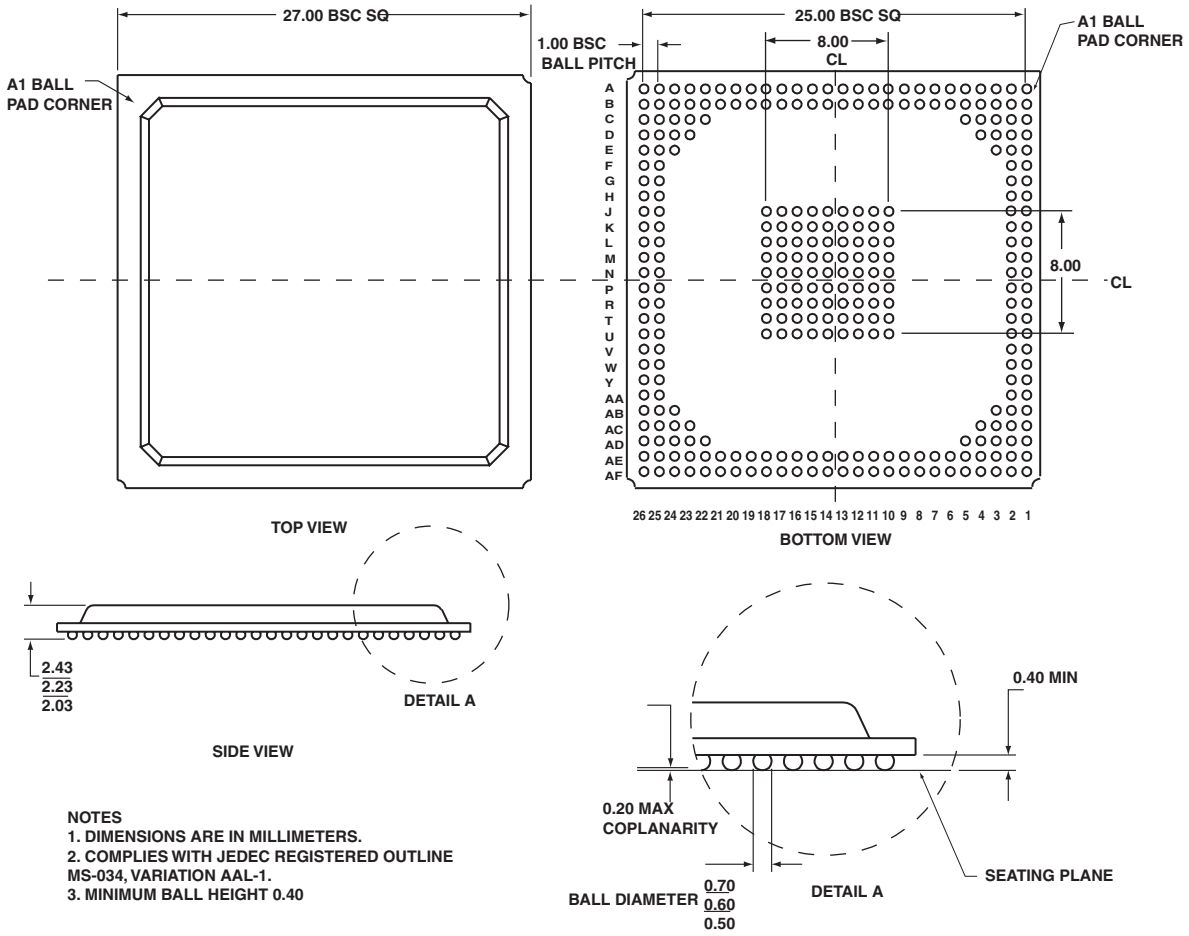

Figure 50. 297-Ball PBGA Grid Array (B-297)

SURFACE MOUNT DESIGN

Table 40 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 40. BGA Data for Use with Surface Mount Design

Package	Ball Attach Type	Solder Mask Opening	Ball Pad Size
256-Ball Mini-Ball Grid Array (BC-256)	Solder Mask Defined	0.30 mm diameter	0.43 mm diameter
297-Ball PBGA Grid Array (B-297)	Solder Mask Defined	0.43 mm diameter	0.58 mm diameter

ORDERING GUIDE

Model	Temperature Range[1]	Package Description	Package Option	Instruction Rate (Max)	Operating Voltage (Nom)
ADSP-BF561SKBCZ600[2]	0°C to +70°C	256-Ball Chip Scale Package Ball Grid Array (Mini-BGA)	BC-256	600 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SKBCZ500[2]	0°C to +70°C	256-Ball Chip Scale Package Ball Grid Array (Mini-BGA)	BC-256	500 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SKB500	0°C to +70°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	500 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SKB600	0°C to +70°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	600 MHz	1.35 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SKBZ500[2]	0°C to +70°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	500 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SKBZ600[2]	0°C to +70°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	600 MHz	1.35 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SBB600	–40°C to +85°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	600 MHz	1.35 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SBB500	–40°C to +85°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	500 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SBBZ600[2]	–40°C to +85°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	600 MHz	1.35 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561SBBZ500[2]	–40°C to +85°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	500 MHz	1.25 V Internal, 2.5 V or 3.3 V I/O
ADSP-BF561WBBZ-5A[2, 3]	–40°C to +85°C	297-Ball Plastic Ball Grid Array (PBGA)	B-297	500 MHz	1.2 V Internal, 2.5 V or 3.3 V I/O

[1] Referenced temperature is ambient temperature.
[2] Z = Pb-free part.
[3] Automotive grade part.

D04696-0-5/06(A)