

FEATURES

Dual symmetric 600 MHz high performance Blackfin cores
328K bytes of on-chip memory (See memory information
on Page 3)

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 program-
ming and compiler-friendly support

Advanced debug, trace, and performance monitoring

0.8 - 1.2V core V_{DD}

3.3V and 2.5V tolerant I/O

256-ball mini-BGA and 297-ball PBGA package options

PERIPHERALS

Two parallel input/output peripheral interface units support-
ing ITU-R 656 video and glueless interface to analog front
end ADCs

Two dual channel, full duplex synchronous serial ports sup-
porting eight stereo I²S channels

Dual 16-channel DMA controllers and one internal memory
DMA controller

12 general-purpose 32-bit timer/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 1x to 63x frequency
multiplication

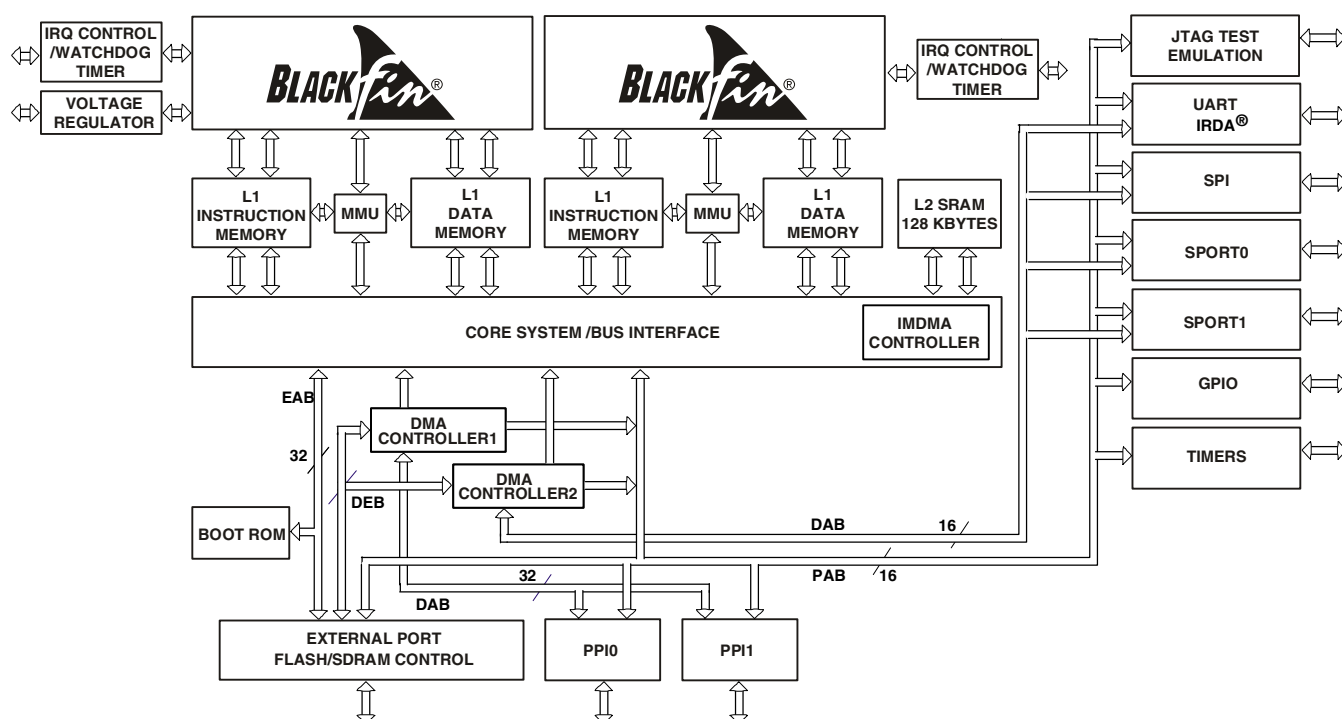


Figure 1. Functional Block Diagram

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

Rev. PrE

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

TABLE OF CONTENTS

| | | | |
|--|----|---|----|
| General Description | 3 | ESD Sensitivity | 21 |
| Portable Low Power Architecture | 3 | Timing Specifications | 22 |
| Blackfin Processor Core | 3 | Clock and Reset Timing | 23 |
| Memory Architecture | 4 | Asynchronous Memory Read Cycle Timing | 24 |
| DMA Controllers | 8 | Asynchronous Memory Write Cycle Timing | 25 |
| WatchDog Timers | 8 | SDRAM Interface Timing | 26 |
| Serial Ports (SPORTs) | 9 | External Port Bus Request and Grant Cycle Timing .. | 27 |
| Serial Peripheral Interface (SPI) Port | 9 | Parallel Peripheral Interface Timing | 28 |
| UART Port | 9 | Serial Ports | 29 |
| Programmable Flags (PFx) | 10 | Serial Peripheral Interface (SPI) Port—Master Timing | 33 |
| Timers | 10 | Serial Peripheral Interface (SPI) Port—Slave Timing . | 34 |
| Parallel Peripheral Interface | 10 | Universal Asynchronous Receiver Transmitter (UART) | |
| Dynamic Power Management | 11 | Port—Receive and Transmit Timing | 35 |
| Voltage Regulation | 12 | Timer Cycle Timing | 36 |
| Clock Signals | 12 | Programmable Flags Cycle Timing | 37 |
| Bootling Modes | 13 | JTAG Test and Emulation Port Timing | 38 |
| Instruction Set Description | 14 | Output Drive Currents | 39 |
| Development Tools | 14 | Power Dissipation | 41 |
| Designing an Emulator-Compatible | | Test Conditions | 42 |
| Processor Board (Target) | 15 | Environmental Conditions | 45 |
| Additional Information | 15 | 256-ball MBGA Pinout | 46 |
| Pin Descriptions | 16 | 297-ball PBGA Pinout | 48 |
| Specifications | 20 | Outline Dimensions | 50 |
| Recommended Operating Conditions | 20 | Outline Dimensions | 51 |
| Electrical Characteristics | 20 | Ordering Guide | 51 |
| Absolute Maximum Ratings | 21 | | |

REVISION HISTORY

12/04—Preliminary PrE

- Deletes tolerance from voltage regulator description and changes part in (Figure 4) Voltage Regulator Circuit .. 12
- Defines new nominal voltage for ADSP-BF561 in Recommended Operating Conditions..... 20
- Clarifies test voltage in Table 9, Table 10, Table 11..... 22
- Changes data in (Table 28) Internal Power Dissipation .. 41
- Changes operating voltage and adds one part number to Ordering Guide..... 51
- Changes to format throughout document.

GENERAL DESCRIPTION

The ADSP-BF561 processor is a high performance member of the Blackfin family of products targeting a variety of multimedia 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 into a single instruction set architecture. The ADSP-BF561 device integrates a general-purpose set of digital imaging peripherals.

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 for embedded signal processing applications. Blackfin processors are designed in a low power and low voltage design methodology and feature Dynamic Power Management. Dynamic Power Management is the ability to vary both the voltage and frequency of operation to significantly lower the overall power dissipation. This translates into an exponential reduction in power dissipation, providing longer battery life to portable applications.

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.

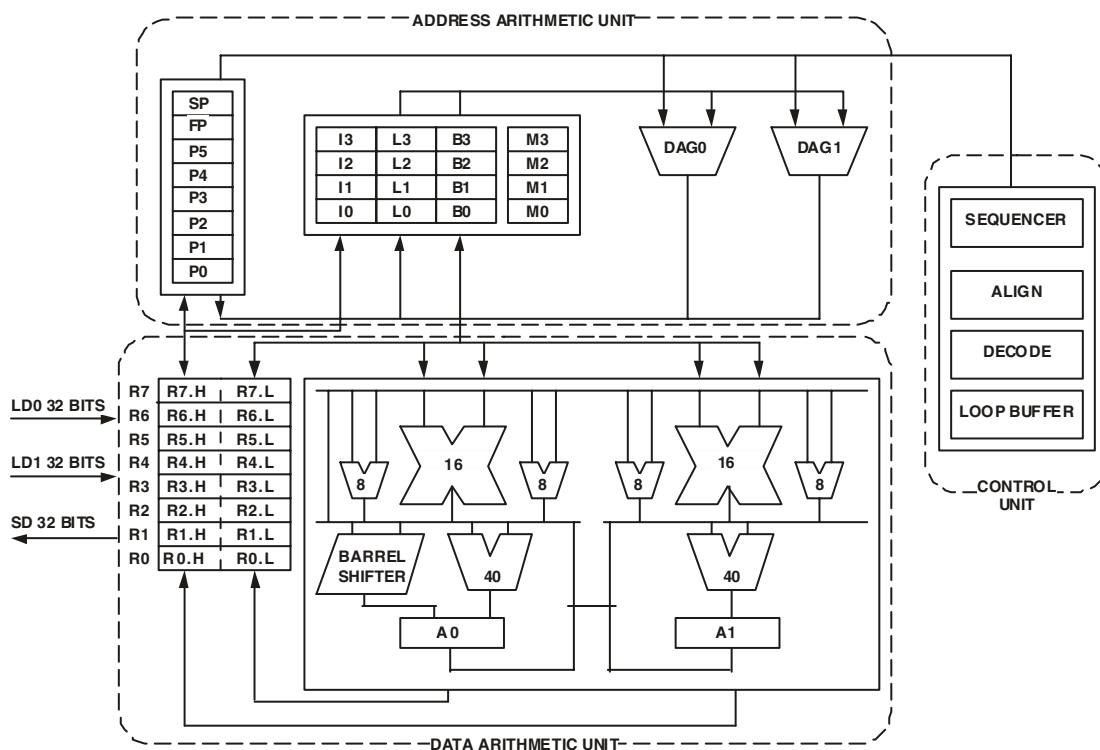


Figure 2. Blackfin Processor Core

Each MAC performs a 16-bit by 16-bit multiply in every cycle, with accumulation to a 40-bit result, providing 8 bits of extended precision. The ALUs perform a standard set of arithmetic and logical operations. With two ALUs capable of operating on 16- or 32-bit data, the flexibility of the computation units covers the signal processing requirements of a varied set of application needs.

Each of the two 32-bit input registers can be regarded as two 16-bit halves, so each ALU can accomplish very flexible single 16-bit arithmetic operations. By viewing the registers as pairs of 16-bit operands, dual 16-bit or single 32-bit operations can be accomplished in a single cycle. 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 multi-ported 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 memories 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 sup-

port 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.

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 4-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).

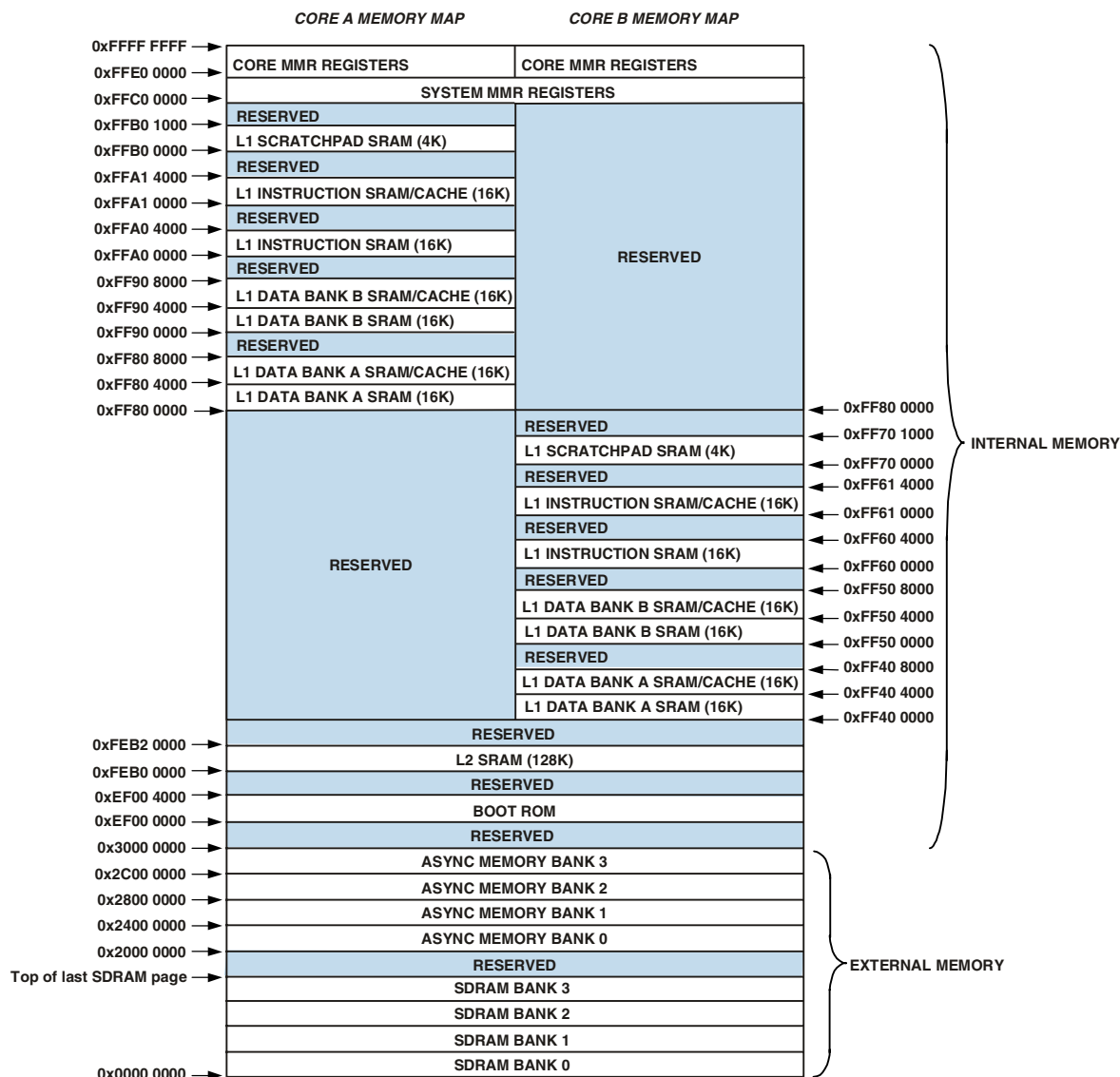


Figure 3. Memory Map

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 4 Gbyte 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.
- **Non-Maskable Interrupt (NMI)**—The NMI event can be generated by the software watchdog timer or by the NMI input signal to the processor. The NMI event is frequently used as a power down indicator to initiate an orderly shut down 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, undefined instructions, etc. 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 | EMU |
| 1 | Reset | RST |
| 2 | Non-Maskable | NMI |
| 3 | Exceptions | EVX |
| 4 | Global Enable | |
| 5 | Hardware Error | IVHW |
| 6 | Core Timer | IVTMR |
| 7 | General Interrupt 7 | IVG7 |
| 8 | General Interrupt 8 | IVG8 |
| 9 | General Interrupt 9 | IVG9 |
| 10 | General Interrupt 10 | IVG10 |
| 11 | General Interrupt 11 | IVG11 |
| 12 | General Interrupt 12 | IVG12 |
| 13 | General Interrupt 13 | IVG13 |
| 14 | General Interrupt 14 | IVG14 |
| 15 | General Interrupt 15 | IVG15 |

System Interrupt Controller (SIC)

The System Interrupt Controller provides the mapping and routing of events from the many peripheral interrupt sources, to the prioritized general-purpose interrupt inputs of the CEC.

Although the ADSP-BF561 provides a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the Interrupt Assignment Registers (SIC_IAR7–0). Table 2 describes the inputs into the SIC and the default mappings into the CEC.

Table 2. Peripheral Interrupt Source Reset State

| Peripheral Interrupt Source | Channel ¹ | IVG ² |
|--------------------------------------|----------------------|------------------|
| PLL wakeup | 0 | IVG07 |
| DMA1 Error (generic) | 1 | IVG07 |
| DMA2 Error (generic) | 2 | IVG07 |
| IMDMA Error | 3 | IVG07 |
| PPIO Error | 4 | IVG07 |
| PPI1 Error | 5 | IVG07 |
| SPORT0 Error | 6 | IVG07 |
| SPORT1 Error | 7 | IVG07 |
| SPI Error | 8 | IVG07 |
| UART Error | 9 | IVG07 |
| Reserved | 10 | IVG07 |
| DMA1 Channel 0 interrupt (PPIO) | 11 | IVG08 |
| DMA1 Channel 1 interrupt (PPI1) | 12 | IVG08 |
| DMA1 Channel 2 interrupt | 13 | IVG08 |
| DMA1 Channel 3 interrupt | 14 | IVG08 |
| DMA1 Channel 4 interrupt | 15 | IVG08 |
| DMA1 Channel 5 interrupt | 16 | IVG08 |
| DMA1 Channel 6 interrupt | 17 | IVG08 |
| DMA1 Channel 7 interrupt | 18 | IVG08 |
| DMA1 Channel 8 interrupt | 19 | IVG08 |
| DMA1 Channel 9 interrupt | 20 | IVG08 |
| DMA1 Channel 10 interrupt | 21 | IVG08 |
| DMA1 Channel 11 interrupt | 22 | IVG08 |
| DMA2 Channel 0 interrupt (SPORT0 RX) | 23 | IVG09 |
| DMA2 Channel 1 interrupt (SPORT0 TX) | 24 | IVG09 |
| DMA2 Channel 2 interrupt (SPORT1 RX) | 25 | IVG09 |
| DMA2 Channel 3 interrupt (SPORT1 TX) | 26 | IVG09 |
| DMA2 Channel 4 interrupt (SPI) | 27 | IVG09 |
| DMA2 Channel 5 interrupt (UART RX) | 28 | IVG09 |
| DMA2 Channel 6 interrupt (UART TX) | 29 | IVG09 |
| DMA2 Channel 7 interrupt | 30 | IVG09 |
| DMA2 Channel 8 interrupt | 31 | IVG09 |
| DMA2 Channel 9 interrupt | 32 | IVG09 |
| DMA2 Channel 10 interrupt | 33 | IVG09 |
| DMA2 Channel 11 interrupt | 34 | IVG09 |
| Timer0 interrupt | 35 | IVG10 |
| Timer1 interrupt | 36 | IVG10 |
| Timer2 interrupt | 37 | IVG10 |
| Timer3 interrupt | 38 | IVG10 |
| Timer4 interrupt | 39 | IVG10 |
| Timer5 interrupt | 40 | IVG10 |

Table 2. Peripheral Interrupt Source Reset State (Continued)

| Peripheral Interrupt Source | Channel ¹ | IVG ² |
|--|----------------------|------------------|
| Timer6 interrupt | 41 | IVG10 |
| Timer7 interrupt | 42 | IVG10 |
| Timer8 interrupt | 43 | IVG10 |
| Timer9 interrupt | 44 | IVG10 |
| Timer10 interrupt | 45 | IVG10 |
| Timer11 interrupt | 46 | IVG10 |
| Programmable Flags 15–0 interrupt A | 47 | IVG11 |
| Programmable Flags 15–0 interrupt B | 48 | IVG11 |
| Programmable Flags 31–16 interrupt A | 49 | IVG11 |
| Programmable Flags 31–16 interrupt B | 50 | IVG11 |
| Programmable Flags 47–32 interrupt A | 51 | IVG11 |
| Programmable Flags 47–32 interrupt B | 52 | IVG11 |
| DMA1 Channel 12/13 interrupt (Memory DMA/Stream 0) | 53 | IVG08 |
| DMA1 Channel 14/15 interrupt (Memory DMA/Stream 1) | 54 | IVG08 |
| DMA2 Channel 12/13 interrupt (Memory DMA/Stream 0) | 55 | IVG09 |
| DMA2 Channel 14/15 interrupt (Memory DMA/Stream 1) | 56 | IVG09 |
| IMDMA Stream 0 interrupt | 57 | IVG12 |
| IMDMA Stream 1 interrupt | 58 | IVG12 |
| Watchdog Timer Interrupt | 59 | IVG13 |
| Reserved | 60 | IVG07 |
| Reserved | 61 | IVG07 |
| Supplemental Interrupt 0 | 62 | IVG07 |
| Supplemental Interrupt 1 | 63 | IVG07 |

¹ Peripheral Interrupt Channel Number

² Default User IVG Interrupt

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 supervisor 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 peripheral

als. 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 (1D) and 2-dimensional (2D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2D 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 TIMERS

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, non-maskable 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} .

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:

- I²S capable operation.
- Bidirectional operation—Each SPORT has two sets of independent transmit and receive pins, enabling eight channels of I²S 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 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 pulsewidths 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 has one 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_x, and Master Input-Slave Output, MISO) and a clock pin (Serial Clock, SCK). One SPI chip select input pin (SPISS) lets other SPI devices select the DSP, and seven SPI chip select output pins (SPISEL7–1) let the DSP select other SPI devices. The SPI select pins are reconfigured Programmable Flag pins. Using these pins, the SPI ports provide a full duplex, synchronous serial interface, which supports both master and slave modes and multimaster environments.

The baud rate and clock phase/polarities for the SPI port are programmable (see SPI Clock Rate equation), and each 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.

$$SPI \text{ clock rate} = \frac{f_{SCLK}}{2 \times SPIBAUD}$$

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 provides a full duplex Universal Asynchronous Receiver/Transmitter (UART) port, 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 to 8 data bits; 1 or 2 stop bits; and none, even, or odd parity. The UART port supports two modes of operation, as follows:

- PIO (Programmed I/O)—The processor sends or receives data by writing or reading I/O-mapped UATX or UARX registers, respectively. The data is double buffered on both transmit and receive.
- DMA (Direct Memory Access)—The DMA controller transfers both transmit and receive data. This reduces the number and frequency of interrupts required to transfer data to and from memory. The UART has two dedicated DMA channels, one for transmit and one for receive. These DMA channels have lower priority than most DMA channels because of their relatively low service rates.

The baud rate (see UART Clock Rate equation), serial data format, error code generation and status, and interrupts for the UART port are programmable. In the UART Clock Rate equation, the divisor (D) can be 1 to 65536.

$$UART \text{ clock rate} = \frac{f_{SCLK}}{16 \times D}$$

The UART's programmable features include:

- Supporting bit rates ranging from ($f_{\text{SCLK}}/1048576$) to ($f_{\text{SCLK}}/16$) bits per second.
- Supporting data formats from 7 to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

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 bi-directional, 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.

TIMERS

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

Each of the twelve 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 twelve 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.

PARALLEL PERIPHERAL INTERFACE

The 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 3 frame synchronization pins, and up to 16 data pins.

General-Purpose Mode Descriptions

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.

ITU-R 656 Mode Descriptions

In ITU-R 656 mode, 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 mode
- Vertical Blanking Only mode
- Entire Field mode

Active Video Only Mode

In this mode, 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 Interval Mode

In this mode, the PPI transfers vertical blanking interval (VBI) data, as well as horizontal blanking information and control byte sequences on VBI lines.

Entire Field Mode

In this mode, 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 of the PPI in a frame syncless mode. The processor's 2D 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.

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.

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.

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 |

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 (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 non-volatile 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 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 | VDD 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:

$$\text{Power Savings Factor} = \frac{f_{CCLKRED}}{f_{CCLKNOM}} \times \left(\frac{V_{DDINTRED}}{V_{DDINTNOM}} \right)^2 \times \left(\frac{T_{RED}}{T_{NOM}} \right)$$

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:

$$\% \text{ Power Savings} = (1 - \text{Power Savings Factor}) \times 100\%$$

VOLTAGE REGULATION

The ADSP-BF561 processor provides an on-chip voltage regulator that can generate processor core voltage levels 0.85V 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 powerdown 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.

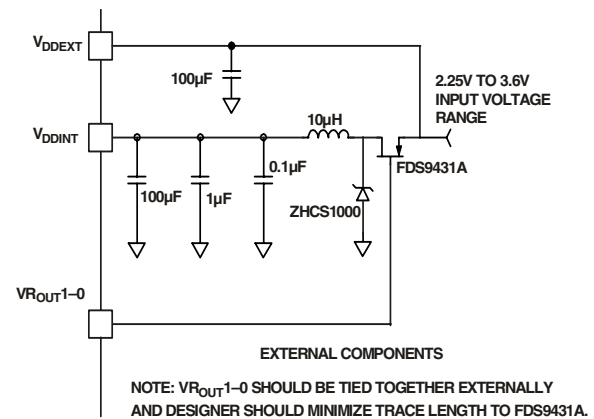


Figure 4. Voltage Regulator Circuit

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.

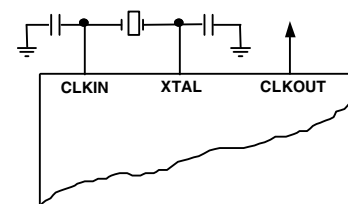


Figure 5. External Crystal Connections

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 1x to 63x multiplication factor. The default multiplier is 10x, 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.

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

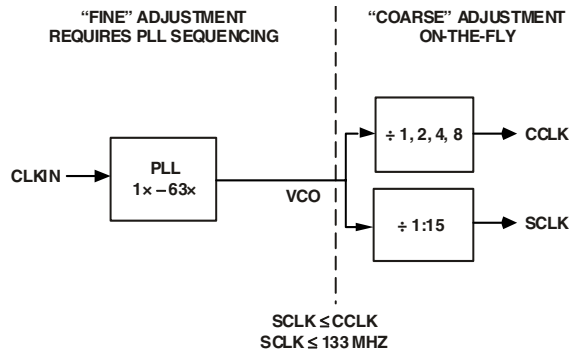


Figure 6. Frequency Modification Methods

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 SSEL[3-0] | 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 CSEL[1-0] 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 CSEL[1-0] | Divider Ratio VCO/CCLK | Example Frequency Ratios | |
|--------------------------|---------------------------|--------------------------|------|
| | | 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 lock time when a change is programmed via the PLL_CTL register is 40 μ sec. The maximum time to change the internal voltage via the internal voltage regulator is also 40 μ sec. The reset value for the PLL_LOCKCNT register is 0x200. This value should be programmed to ensure a 40 μ sec wakeup time when either the voltage is changed or a new MSEL value is programmed. The value should be programmed to

ensure an 80 μ sec wakeup time when both voltage and the MSEL value are changed. The time base for the PLL_LOCKCNT 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 8/16-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 8/16-bit external FLASH memory – The 8/16-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 provides fully featured multifunction instructions that allow the programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both a user (algorithm/application code) and a supervisor (O/S kernel, device drivers, debuggers, ISRs) mode of operation—allowing multiple levels of access to core processor resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/CPU features are optimized for both 8-bit and 16-bit operations.
- A multi-issue load/store modified Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU plus two load/store plus two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4 Gbyte memory space providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data types; and separate user and kernel stack pointers.
- Code density enhancements, which include intermixing of 16- and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded as 16-bits.

DEVELOPMENT TOOLS

The ADSP-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 trans-

lation 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 non-intrusively 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.

^{*} CROSSCORE is a registered trademark of Analog Devices, Inc.

[†] VisualDSP++ is a registered trademark of Analog Devices, Inc.

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.

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. Non-intrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect target system loading or timing.

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.

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 Ref-*

erence on the Analog Devices website (www.analog.com)—use site search on “EE-68.” This document is updated regularly to keep pace with improvements to emulator support.

ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-BF561 architecture and functionality. For detailed information on the Blackfin DSP family core architecture and instruction set, refer to the *ADSP-BF561 Hardware Reference and the Blackfin Family Instruction Set Reference*.

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 22](#) through [Figure 29](#).

Table 8. Pin Descriptions

| Block | Pin Name | Type | Signals | Function | Driver Type | Pull-up/down requirement |
|--------------|---------------------------------|------|---------|--|-------------|---|
| EBIU | ADDR[25:2] | O | 24 | Address Bus for Async/Sync Access | A | none |
| | DATA[31:0] | I/O | 32 | Data Bus for Async/Sync Access | A | none |
| | $\overline{ABE}[3:0]/SDQM[3:0]$ | O | 4 | Byte Enables/Data Masks for Async /Sync Access | A | none |
| | \overline{BG} | O | 1 | Bus Grant | A | none |
| | \overline{BR} | I | 1 | Bus Request | | pull-up required if function not used |
| | \overline{BGH} | O | 1 | Bus Grant Hang | A | none |
| EBIU (SDRAM) | \overline{SRAS} | O | 1 | Row Address Strobe | A | none |
| | \overline{SCAS} | O | 1 | Column Address Strobe | A | none |
| | \overline{SWE} | O | 1 | Write Enable | A | none |
| | SCKE | O | 1 | Clock Enable | A | none |
| | SCLK0/CLKOUT | O | 1 | Clock Output Pin 0 | B | none |
| | SCLK1 | O | 1 | Clock Output Pin 1 | B | none |
| | SA10 | O | 1 | SDRAM A10 Pin | A | none |
| | $\overline{SMS}[3:0]$ | O | 4 | Bank Select | A | none |
| EBIU (ASYNC) | $\overline{AMS}[3:0]$ | O | 4 | Bank Select | A | none |
| | ARDY | I | 1 | Hardware Ready Control | | pull-up required if function not used |
| | \overline{AOE} | O | 1 | Output Enable | A | none |
| | \overline{AWE} | O | 1 | Write Enable | A | none |
| | \overline{ARE} | O | 1 | Read Enable | A | none |
| PPI0 | PPI0D[15:8] /PF[47:40] | I/O | 8 | PPI Data / Programmable Flag Pins | C | software configurable, none |
| | PPI0D[7:0] | I/O | 8 | PPI Data Pins | C | software configurable, none |
| | PPI0CLK | I | 1 | PPI Clock | | software configurable, none |
| | PPI0SYNC1/ TMR8 | I/O | 1 | PPI Sync / Timer | C | software configurable, none |
| | PPI0SYNC2/ TMR9 | I/O | 1 | PPI Sync / Timer | C | software configurable, none |
| | PPI0SYNC3 | I/O | 1 | PPI Sync | C | software configurable, none |
| PPI1 | PPI1D[15:8] /PF[39:32] | I/O | 8 | PPI Data / Programmable Flag Pins | C | software configurable, none |
| | PPI1D[7:0] | I/O | 8 | PPI Data Pins | C | software configurable, none |
| | PPI1CLK | I | 1 | PPI Clock | | software configurable, none |
| | PPI1SYNC1/ TMR10 | I/O | 1 | PPI Sync / Timer | C | software configurable, none |
| | PPI1SYNC2/ TMR11 | I/O | 1 | PPI Sync / Timer | C | software configurable, none |
| | PPI1SYNC3 | I/O | 1 | PPI Sync | C | software configurable, none |
| JTAG | \overline{EMU} | O | 1 | Emulation Output | C | none |
| | TCK | I | 1 | JTAG Clock | | internal pull-down |
| | TDO | O | 1 | JTAG Serial Data Out | C | none |
| | TDI | I | 1 | JTAG Serial Data In | | internal pull-down |
| | TMS | I | 1 | JTAG Mode Select | | internal pull-down |
| | \overline{TRST} | I | 1 | JTAG Reset | | external pull-down necessary if JTAG not used |

Table 8. Pin Descriptions (Continued)

| Block | Pin Name | Type | Signals | Function | Driver Type | Pull-up/down requirement |
|--------|-------------|------|---------|---|-------------|---|
| UART | RX/PF27 | I/O | 1 | UART Receive / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | TX/PF26 | I/O | 1 | UART Transmit / Programmable Flag | C | software configurable, no pull-up/down necessary |
| SPI | MOSI | I/O | 1 | Master Out Slave In | C | software configurable, no pull-up/down necessary |
| | MISO | I/O | 1 | Master In Slave Out | C | pull-up is necessary if booting via SPI |
| | SCK | I/O | 1 | SPI Clock | D | software configurable, no pull-up/down necessary |
| SPORT0 | RSCLK0/PF28 | I/O | 1 | Sport0 / Programmable Flag | D | software configurable, no pull-up/down necessary |
| | RFS0/PF19 | I/O | 1 | Sport0 Receive Frame Sync / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DR0PRI | I | 1 | Sport0 Receive Data Primary | | software configurable, no pull-up/down necessary |
| | DR0SEC/PF20 | I/O | 1 | Sport0 Receive Data Secondary / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | TSCLK0/PF29 | I/O | 1 | Sport0 Transmit Serial Clock / Programmable Flag | D | software configurable, no pull-up/down necessary |
| | TFS0/PF16 | I/O | 1 | Sport0 Transmit Frame Sync / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DT0PRI/PF18 | I/O | 1 | Sport0 Transmit Data Primary / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DT0SEC/PF17 | I/O | 1 | Sport0 Transmit Data Secondary / Programmable Flag | C | software configurable, no pull-up/down necessary |
| SPORT1 | RSCLK1/PF30 | I/O | 1 | Sport1 / Programmable Flag | D | software configurable, no pull-up/down necessary |
| | RFS1/PF24 | I/O | 1 | Sport1 Receive Frame Sync / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DR1PRI | I | 1 | Sport1 Receive Data Primary | | software configurable, no pull-up/down necessary |
| | DR1SEC/PF25 | I/O | 1 | Sport1 Receive Data Secondary / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | TSCLK1/PF31 | I/O | 1 | Sport1 Transmit Serial Clock / Programmable Flag | D | software configurable, no pull-up/down necessary |
| | TFS1/PF21 | I/O | 1 | Sport1 Transmit Frame Sync / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DT1PRI/PF23 | I/O | 1 | Sport1 Transmit Data Primary / Programmable Flag | C | software configurable, no pull-up/down necessary |
| | DT1SEC/PF22 | I/O | 1 | Sport1 Transmit Data Secondary / Programmable Flag | C | software configurable, no pull-up/down necessary |

Table 8. Pin Descriptions (Continued)

| Block | Pin Name | Type | Signals | Function | Driver Type | Pull-up/down requirement |
|-----------------|---------------------------|------|---------|--|-------------|--|
| PF/TIMER | PF15/EXT CLK | I/O | 1 | Programmable Flag / external timer clock input | C | software configurable, no pull-up/down necessary |
| | PF14 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF13 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF12 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF11 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF10 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF9 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF8 | I/O | 1 | Programmable Flag | C | software configurable, no pull-up/down necessary |
| | PF7/SPISEL7/TMR7 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF6/SPISEL6/TMR6 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF5/SPISEL5/TMR5 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF4/SPISEL4/TMR4 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF3/SPISEL3/TMR3 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF2/SPISEL2/TMR2 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF1/SPISEL1/TMR1 | I/O | 1 | Programmable Flag / SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| | PF0/SPISS/TMR0 | I/O | 1 | Programmable Flag / Slave SPI Select / Timer | C | software configurable, no pull-up/down necessary |
| Clock Generator | CLKIN | I | 1 | Clock input | | needs to be at a level or clocking |
| | XTAL | O | 1 | Crystal connection | | none |
| Mode Controls | $\overline{\text{RESET}}$ | I | 1 | Chip reset signal | C | always active if core power on |
| | SLEEP | O | 1 | Sleep | | none |
| | BMODE[1:0] | I | 2 | Dedicated Mode Pin, Configures the boot mode that is employed following a hardware reset or software reset | | pull-up or pull-down required |
| | BYPASS | I | 1 | PLL BYPASS control | | pull-up or pull-down required |
| | NMI0 | I | 1 | Non-maskable interrupt Core A | | pull-down required if function not used |
| | NMI1 | I | 1 | Non-maskable interrupt Core B | | pull-down required if function not used |
| Regulator | VROUT1–0 | O | 2 | Regulation output | | N/A |

Table 8. Pin Descriptions (Continued)

| Block | Pin Name | Type | Signals | Function | Driver Type | Pull-up/down requirement |
|------------|---------------|------|---------|---------------------|-------------|--------------------------|
| Supplies | VDDEXT | P | 23 | Power Supply | | N/A |
| | VDDINT | P | 14 | Power Supply | | N/A |
| | GND | G | 41 | Power Supply Return | | N/A |
| | No Connection | NC | 2 | NC | | N/A |
| Total pins | | | 256 | | | |

SPECIFICATIONS

Note that component specifications are subject to change without notice.

RECOMMENDED OPERATING CONDITIONS

| Parameter | Parameter | Minimum | Nominal | Maximum | Unit |
|---------------|---|---------|------------|---------|------|
| V_{DDINT}^1 | Internal Supply Voltage ADSP-BF561SKBCZ600, ADSP-BF561SKBCZ500 | 0.8 | 1.25 | 1.375 | V |
| V_{DDINT}^2 | Internal Supply Voltage ADSP-BF561SBB600 | 0.8 | 1.35 | 1.43 | V |
| V_{DDINT}^3 | Internal Supply Voltage ADSP-BF561SBB500 | 0.8 | 1.25 | 1.375 | V |
| V_{DDEXT} | External Supply Voltage | 2.25 | 2.5 or 3.3 | 3.6 | V |
| V_{IH} | High Level Input Voltage ⁴ , @ V_{DDEXT} = maximum | 2.0 | | 3.6 | V |
| V_{IL} | Low Level Input Voltage, @ V_{DDEXT} = minimum | -0.3 | | 0.6 | V |
| $T_{AMBIENT}$ | Ambient Operating Temperature | | | | |
| | Industrial | -40 | | 85 | °C |
| | Commercial | 0 | | 70 | °C |

¹Internal Voltage Regulator tolerance:

ADSP-BF561SKBCZ500, ADSP-BF561SKBCZ600: V_{DDINT} = -5% to +10%

²Internal Voltage Regulator tolerance:

ADSP-BF561SBB600: V_{DDINT} = -7% to +12%

³Internal Voltage Regulator tolerance:

ADSP-BF561SBB500: V_{DDINT} = -7% to +12% except at 1.25 V: V_{DDINT} = -5% to +10%

⁴The 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 bi-directional and input only pins.

ELECTRICAL CHARACTERISTICS

| Parameter | Test Conditions | Minimum | Maximum | Unit |
|-----------|--|---------|---------|------|
| V_{OH} | High Level Output Voltage ¹ | 2.4 | | V |
| V_{OL} | Low Level Output Voltage ¹ | | 0.4 | V |
| I_{IL} | Low Level Input Current ² | -10 | | V |
| I_{IH} | High Level Input Current ³ | | 10 | μA |
| I_{IH} | High Level Input Current ⁴ | | 50 | μA |
| I_{OZH} | Three-State Leakage Current ⁵ | | 10 | μA |
| I_{OZL} | Three-State Leakage Current ⁵ | -10 | | μA |
| C_{IN} | Input Capacitance ^{6,7} | | TBD | pF |

¹Applies to output and bidirectional pins.

²Applies to all input pins.

³Applies to all input pins except TCK, TDI, TMS, and \overline{TRST} .

⁴Applies to TCK, TDI, TMS, and \overline{TRST} .

⁵Applies to three-statable pins.

⁶Applies to all signal pins.

⁷Guaranteed, but not tested.

ABSOLUTE MAXIMUM RATINGS

| Parameter | Value |
|---|-------------------------------|
| Internal (Core) Supply Voltage ¹ (V_{DDINT}) | –0.3 V to +1.45 V |
| External (I/O) Supply Voltage ¹ (V_{DDEXT}) | –0.3 V to +3.8 V |
| Input Voltage ¹ | –0.5 V to 3.6 V |
| Output Voltage Swing ¹ | –0.5 V to $V_{DDEXT} + 0.5$ V |
| Load Capacitance ^{1,2} | 200 pF |
| Core Clock (CCLK) ¹ | |
| ADSP-BF561SKBCZ600/ADSP-BF561SBB600 | 600 MHz |
| ADSP-BF561SKBCZ500 | 500 MHz |
| System Clock (SCLK) ¹ | 133 MHz |
| Storage Temperature Range ¹ | –65 °C to +150 °C |
| Junction Temperature Under Bias | 125 °C |

¹ Stresses greater than those listed above may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

² For proper SDRAM controller operation, the maximum load capacitance is 50 pF (at 3.3 V) or 30 pF (at 2.5 V) for ADDR25–2, DATA31–0, ABE3–0/SDQM3–0, CLKOUT, SCKE, SA10, SRAS, SCAS, SWE, and SMS.

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.



TIMING SPECIFICATIONS

Table 9 and Table 12 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 fre-

quencies, as described in [For more information, see “Absolute Maximum Ratings” on page 21.](#) Table 12 describes Phase Locked Loop operating conditions.

Table 9. Core and System Clock Requirements—ADSP-BF561SKBCZ500 and ADSP-BF561SBB500

| Parameter | | Minimum | Maximum | Unit |
|-------------------|---|---------|---------|------|
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.1875 V minimum) | 2 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.045 V minimum) | 2.25 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 0.95 V minimum) | 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 V minimum) | 4.00 | | ns |

Table 10. Core and System Clock Requirements—ADSP-BF561SKBCZ600

| Parameter | | Minimum | Maximum | Unit |
|-------------------|---|---------|---------|------|
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.1875 V minimum) | 1.66 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.045 V minimum) | 2.10 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 0.95 V minimum) | 2.35 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 0.855 V minimum) | 2.66 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 0.8 V minimum) | 4.00 | | ns |

Table 11. Core and System Clock Requirements—ADSP-BF561SBB600

| Parameter | | Minimum | Maximum | Unit |
|-------------------|--|---------|---------|------|
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.2825 V minimum) ¹ | 1.66 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.14 V minimum) | 2.0 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 1.045 V minimum) | 2.25 | | ns |
| t _{CCLK} | Core Cycle Period (V _{DDINT} = 0.95 V minimum) | 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 V minimum) | 4.00 | | ns |

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

Table 12. Phase Locked Loop Operating Conditions

| Parameter | Minimum | Maximum | Unit |
|---|---------|--------------|------|
| Voltage Controlled Oscillator (VCO) Frequency | 50 | Maximum CCLK | MHz |

Clock and Reset Timing

Table 13 and Figure 7 describe clock and reset operations. Per Figure 7, combinations of CLKIN and clock multipliers must not select core/peripheral clocks in excess of 600/133 MHz.

Table 13. Clock and Reset Timing

| Parameter | Min | Max | Unit |
|--|------------------|-------|------|
| <i>Timing Requirements</i> | | | |
| t_{CKIN} CLKIN Period | 25.0 | 100.0 | ns |
| t_{CKINL} CLKIN Low Pulse ¹ | 10.0 | | ns |
| t_{CKINH} CLKIN High Pulse ¹ | 10.0 | | ns |
| t_{WRST} \overline{RESET} Asserted Pulsewidth Low ² | 11 t_{CKIN} | | ns |
| <i>Switching Characteristics</i> | | | |
| t_{SCLK} CLKOUT Period ³ | 7.5 ⁴ | | ns |

¹ Applies to bypass mode and non-bypass mode.

² Applies after power-up sequence is complete. At power-up, the processor's internal phase locked loop requires no more than 2,000 CLKIN cycles, while \overline{RESET} is asserted, assuming stable power supplies and CLKIN (not including startup time of external clock oscillator).

³ The figure below shows a $\times 2$ ratio between t_{CKIN} and t_{SCLK} , but the ratio has many programmable options. For more information, see the System Design chapter of the ADSP-BF561 Hardware Reference.

⁴ t_{SCLK} must always also be larger than t_{CCLK} .

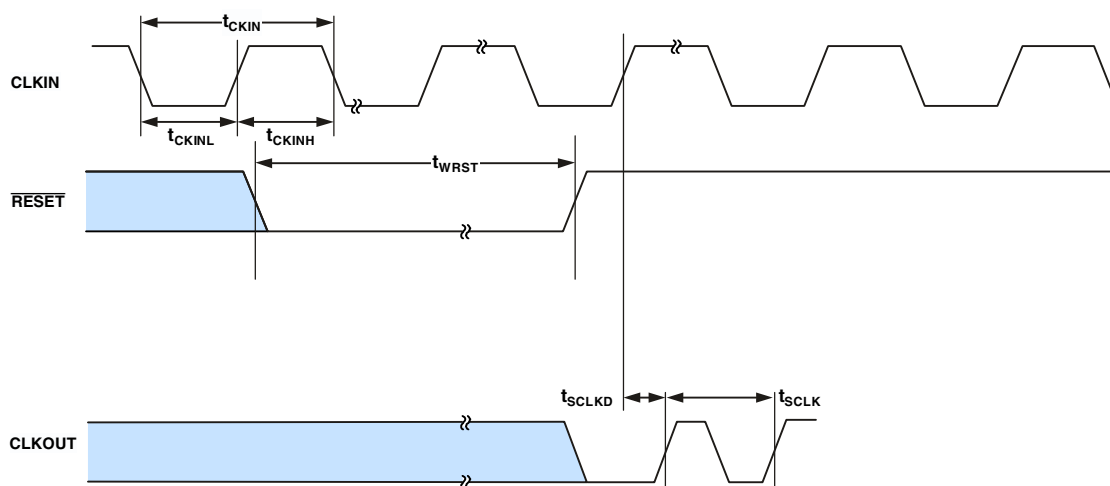


Figure 7. Clock and Reset Timing

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

| Parameter | | Min | Max | Unit |
|---------------------------------|--|-----|-----|------|
| <i>Timing Requirements</i> | | | | |
| 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 |
| <i>Switching Characteristic</i> | | | | |
| t_{DO} | Output Delay After CLKOUT ¹ | | 6.0 | ns |
| t_{HO} | Output Hold After CLKOUT ¹ | 0.8 | | ns |

¹ Output pins include $\overline{AMS}3-0$, $\overline{ABE}3-0$, $\overline{ADDR}25-2$, \overline{AOE} , \overline{ARE} .

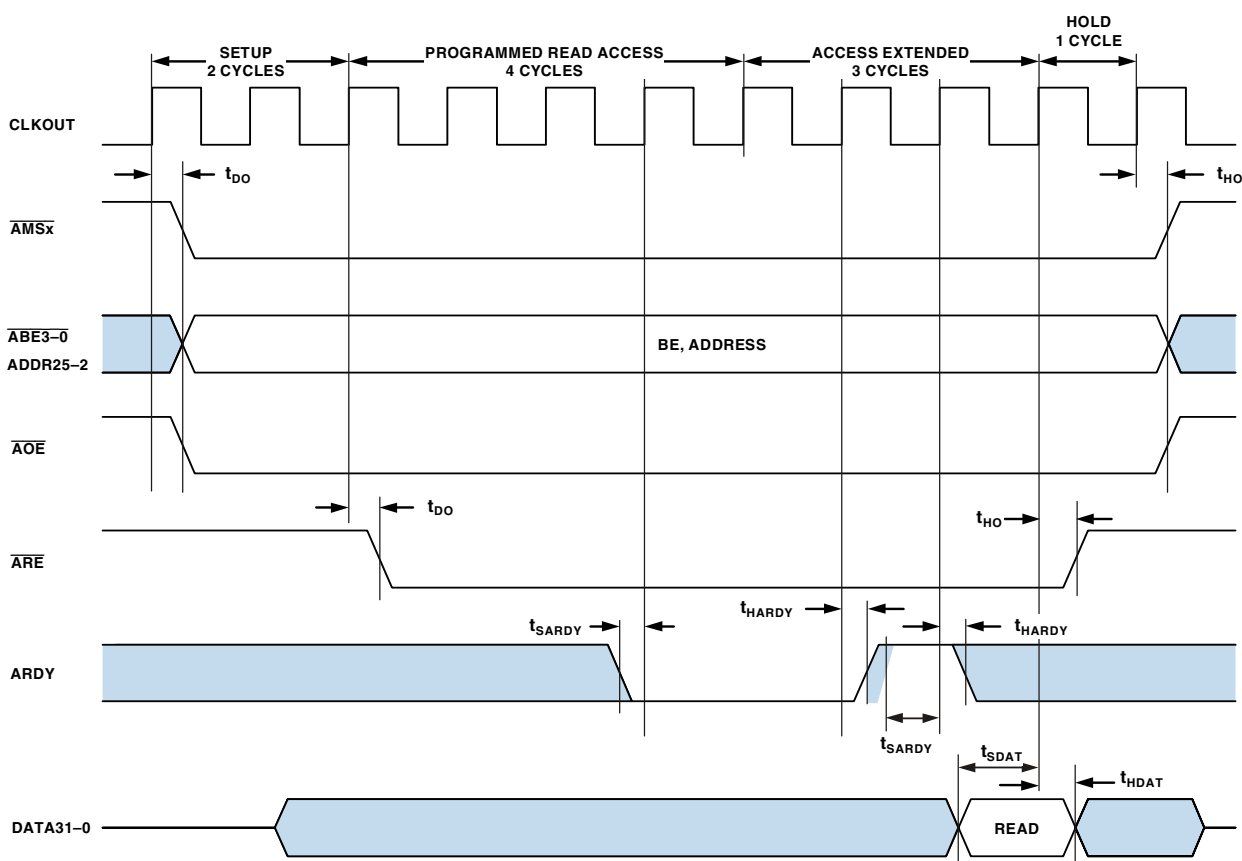
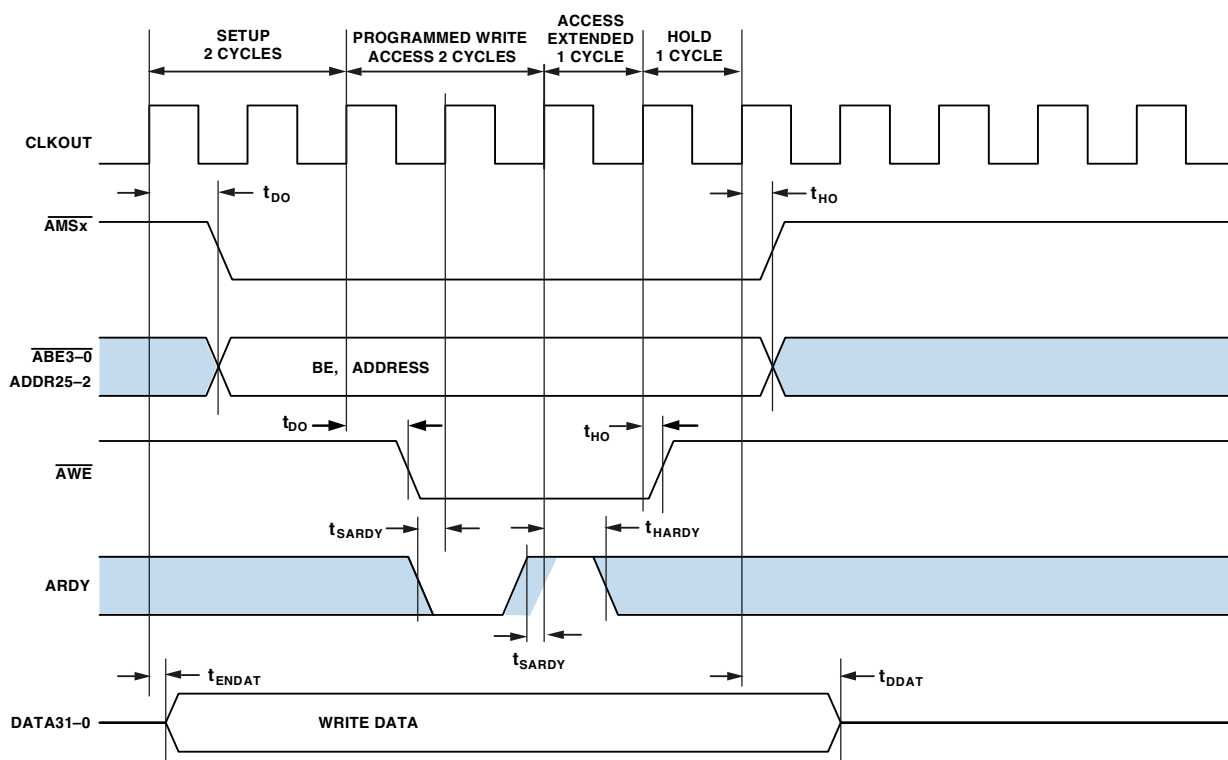


Figure 8. Asynchronous Memory Read Cycle Timing

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

| Parameter | | Min | Max | Unit |
|---------------------------------|--|-----|-----|------|
| <i>Timing Requirements</i> | | | | |
| t_{SARDY} | ARDY Setup Before CLKOUT | 4.0 | | ns |
| t_{HARDY} | ARDY Hold After CLKOUT | 0.0 | | ns |
| <i>Switching Characteristic</i> | | | | |
| 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 |

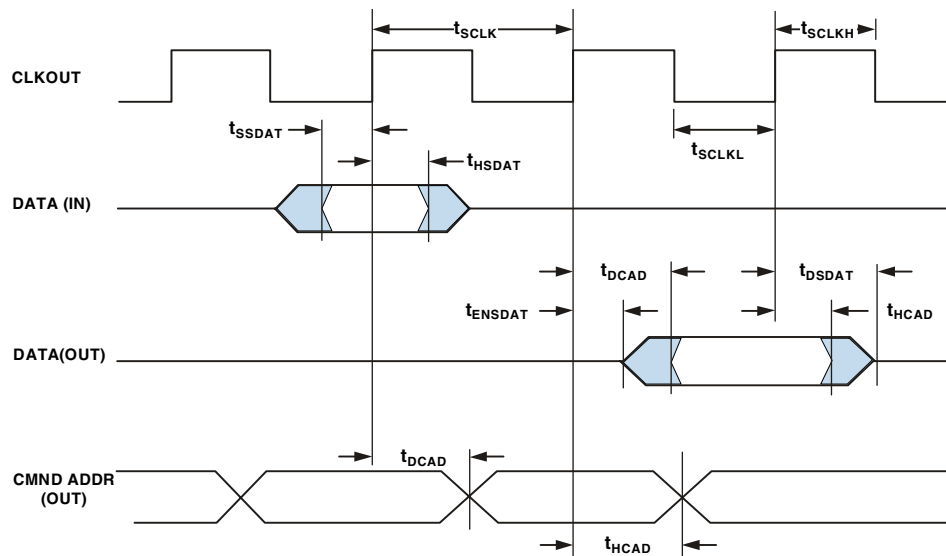
¹ Output pins include AMS3–0, ABE3–0, ADDR25–2, DATA31–0, AOE, AWE.

**Figure 9. Asynchronous Memory Write Cycle Timing**

SDRAM Interface Timing**Table 16. SDRAM Interface Timing**

| Parameter | Min | Max | Unit |
|--|-----|-----|------|
| <i>Timing Requirement</i> | | | |
| t_{SSDAT} DATA Setup Before CLKOUT | 2.1 | | ns |
| t_{HSDAT} DATA Hold After CLKOUT | 0.8 | | ns |
| <i>Switching Characteristic</i> | | | |
| t_{SCLK} CLKOUT Period | 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 ¹ | | 6.0 | ns |
| t_{HCAD} Command, ADDR, Data Hold After CLKOUT ¹ | 0.8 | | ns |
| t_{DSDAT} Data Disable After CLKOUT | | 6.0 | ns |
| t_{ENSDAT} Data Enable After CLKOUT | 1.0 | | ns |

¹ Command pins include: \overline{SRAS} , \overline{SCAS} , \overline{SWE} , \overline{SDQM} , \overline{SMS} 3-0, SA10, SCKE.



NOTE: COMMAND = \overline{SRAS} , \overline{SCAS} , \overline{SWE} , \overline{SDQM} , \overline{SMS} , SA10, SCKE.

Figure 10. SDRAM Interface Timing

External Port Bus Request and Grant Cycle Timing

Table 17 and Figure 11 describe external port bus request and bus grant operations.

Table 17. External Port Bus Request and Grant Cycle Timing

| Parameter ^{1, 2} | Min | Max | Unit |
|--|-----|-----|------|
| <i>Timing Requirements</i> | | | |
| t_{BS} \overline{BR} Asserted to Clkout High Setup | 4.6 | | ns |
| t_{BH} Clkout High to \overline{BR} De-asserted Hold Time | 0.0 | | ns |
| <i>Switching Characteristics</i> | | | |
| 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} De-asserted Hold Time | | 3.6 | ns |
| t_{DBH} Clkout High to \overline{BGH} Asserted Setup | | 3.6 | ns |
| t_{EBH} Clkout High to \overline{BGH} De-asserted 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.

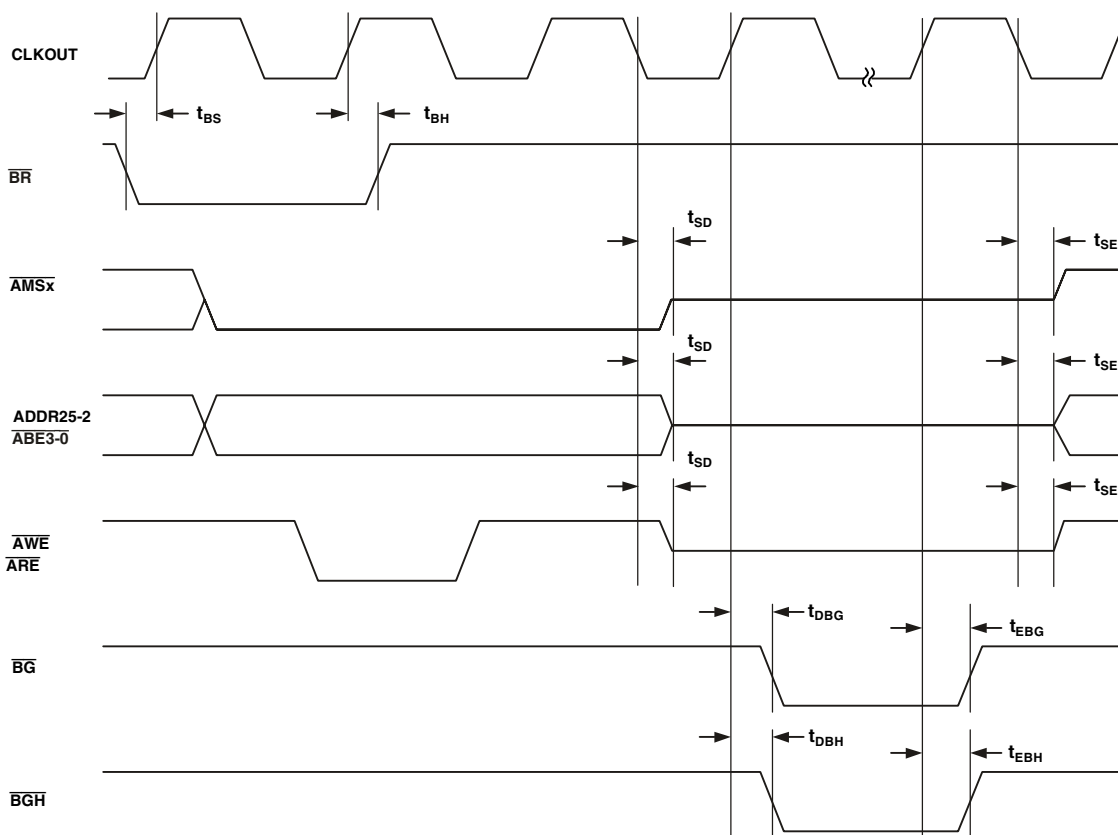


Figure 11. External Port Bus Request and Grant Cycle Timing

Parallel Peripheral Interface Timing

Table 18, Figure 12, describes Parallel Peripheral Interface operations.

Table 18. Parallel Peripheral Interface Timing

| Parameter | Min | Max | Unit |
|--|------|------|------|
| <i>Timing Requirements</i> | | | |
| t_{PCLKW} PPIx_CLK Width ¹ | 6.0 | | ns |
| t_{PCLK} PPI_CLK Period ¹ | 15.0 | | ns |
| t_{SFSPE} External Frame Sync Setup Before PPI_CLK | 3.0 | | ns |
| t_{HFSPE} External Frame Sync Hold After PPI_CLK | 3.0 | | ns |
| t_{SDRPE} Receive Data Setup Before PPI_CLK | 2.0 | | ns |
| t_{HDRPE} Receive Data Hold After PPI_CLK | 4.0 | | ns |
| <i>Switching Characteristics</i> | | | |
| t_{DFSPE} Internal Frame Sync Delay After PPI_CLK | | 10.0 | ns |
| $t_{HOFSPPE}$ Internal Frame Sync Hold After PPI_CLK | 0.0 | | ns |
| t_{DDTPE} Transmit Data Delay After PPI_CLK | | 10.0 | ns |
| t_{HDTPE} Transmit Data Hold After PPI_CLK | 0.0 | | ns |

¹ PPI_CLK frequency cannot exceed $f_{SCLK}/2$

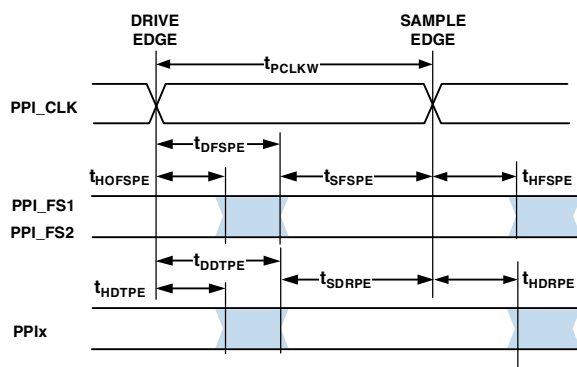


Figure 12. Timing Diagram PPI

Serial Ports

Table 19 through Table 22 on Page 30 and Figure 13 on Page 30 through Figure 15 on Page 32 describe Serial Port operations.

Table 19. Serial Ports—External Clock

| Parameter | Min | Max | Unit |
|---|------|------|------|
| <i>Timing Requirements</i> | | | |
| 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 |
| <i>Switching Characteristics</i> | | | |
| 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 |

¹Referenced to sample edge.

²Referenced to drive edge.

Table 20. Serial Ports—Internal Clock

| Parameter | Min | Max | Unit |
|---|------|-----|------|
| <i>Timing Requirements</i> | | | |
| 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 |
| <i>Switching Characteristics</i> | | | |
| 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 |

¹Referenced to sample edge.

²Referenced to drive edge.

Table 21. Serial Ports—Enable and Three-State

| Parameter | Min | Max | Unit |
|--|------|------|------|
| <i>Switching Characteristics</i> | | | |
| 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 |

¹Referenced to drive edge.

Table 22. External Late Frame Sync

| Parameter | Min | Max | Unit |
|--|-----|------|------|
| <i>Switching Characteristics</i> | | | |
| $t_{DDTLFSE}$ Data Delay from Late External TFS or External RFS with MCE = 1, MFD = 0 ^{1,2} | | 10.0 | ns |
| $t_{DTENLFSE}$ Data Enable from Late FS or MCE = 1, MFD = 0 ^{1,2} | 0 | | ns |

¹MCE = 1, TFS enable and TFS valid follow $t_{DDTENFS}$ and $t_{DDTLFSE}$.

²If external RFS/TFS setup to $RSCLK/TSCLK > t_{SCLK}/2$ then t_{DDTLCK} and $t_{DTENLCK}$ apply, otherwise $t_{DDTLFSE}$ and $t_{DTENLFSE}$ apply.

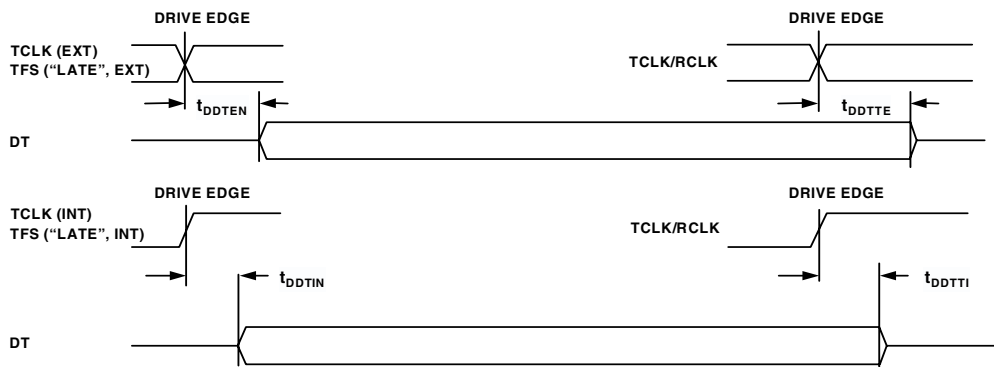
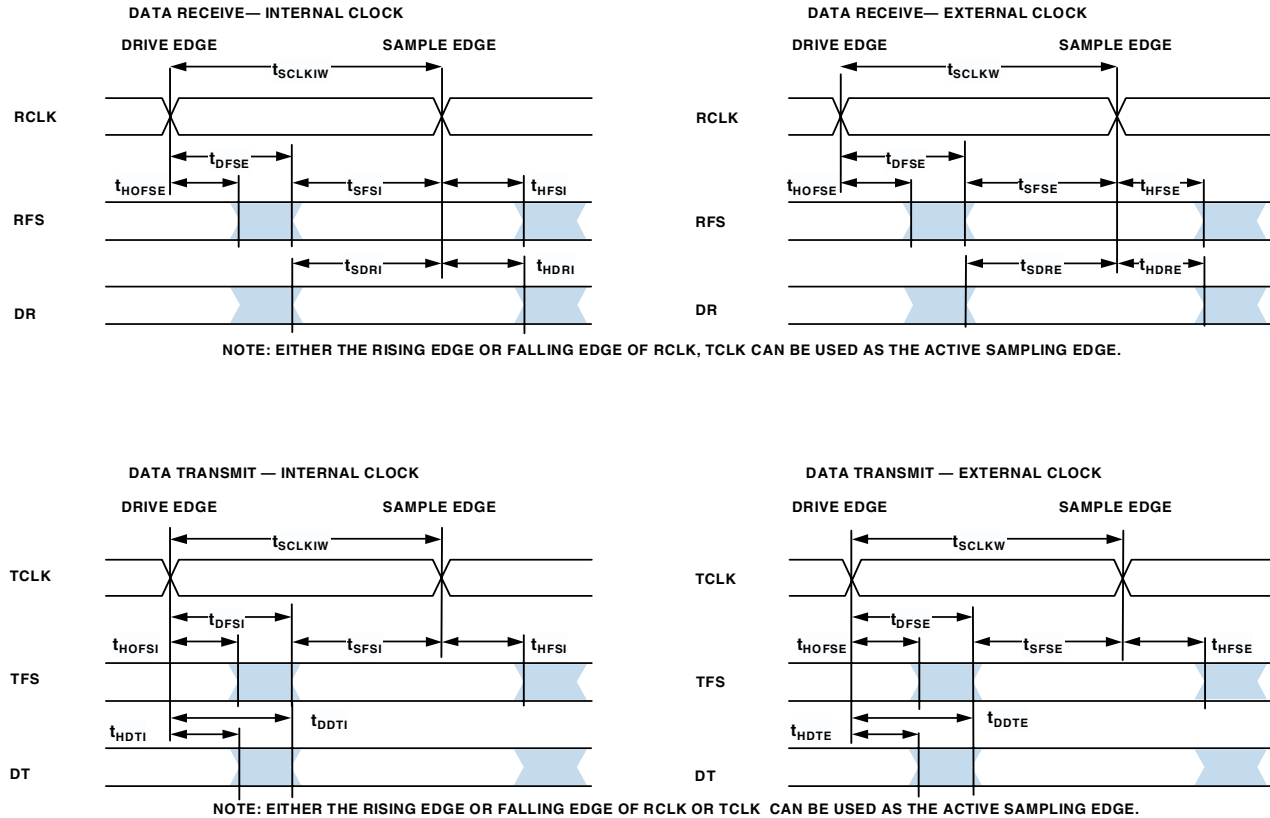
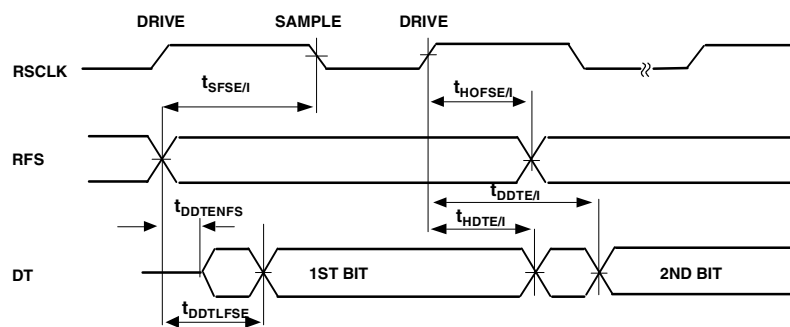


Figure 13. Serial Ports

EXTERNAL RFS WITH MCE = 1, MFD = 0



LATE EXTERNAL TFS

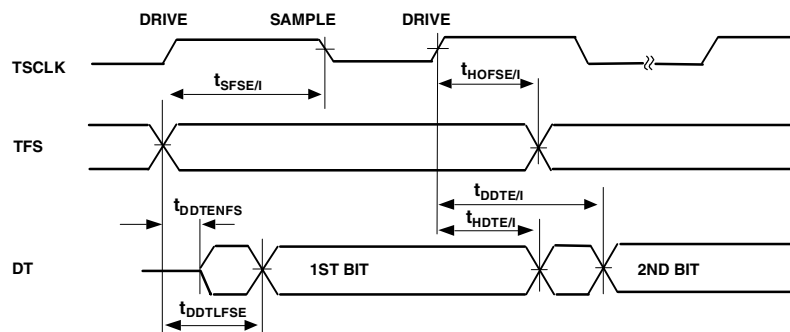


Figure 14. External Late Frame Sync (Frame Sync Setup < $t_{SCLK/2}$)

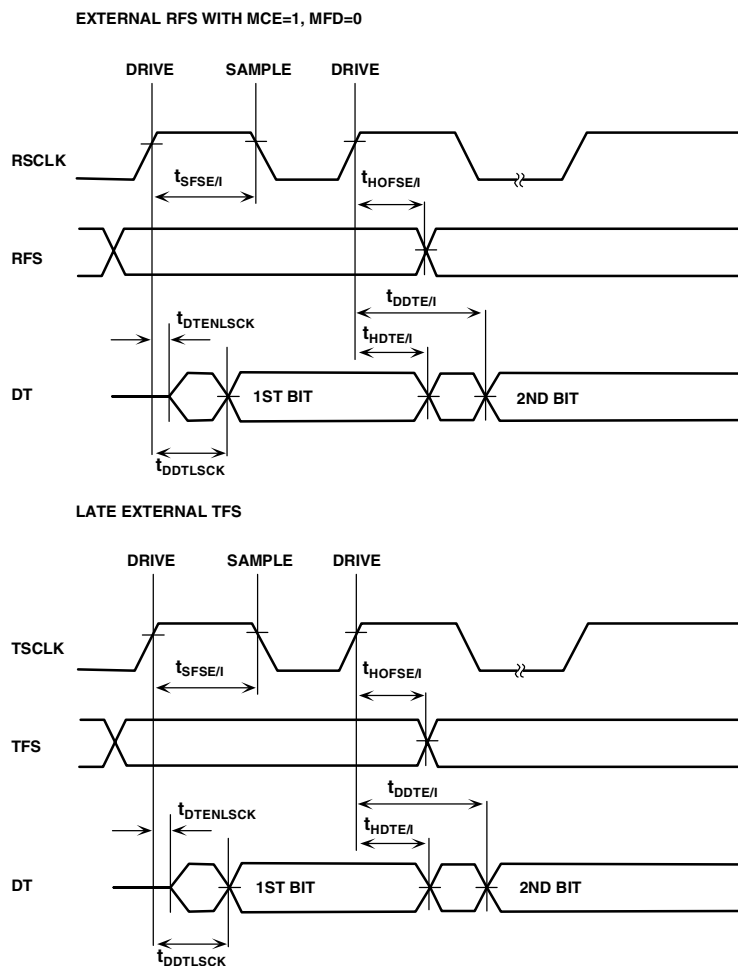


Figure 15. External Late Frame Sync (Frame Sync Setup > $t_{SCLK/2}$)

Serial Peripheral Interface (SPI) Port—Master Timing

Table 23 and Figure 16 describe SPI port master operations.

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

| Parameter | Min | Max | Unit |
|--|-----------------|-----|------|
| <i>Timing Requirements</i> | | | |
| 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 |
| <i>Switching Characteristics</i> | | | |
| t_{SDSCIM} $\overline{SPISELx}$ Low to First Sck Edge | $2t_{SCLK}-1.5$ | | ns |
| t_{SPICHM} Serial Clock High Period | $2t_{SCLK}-.5$ | | 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 $\overline{SPISELx}$ 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 |

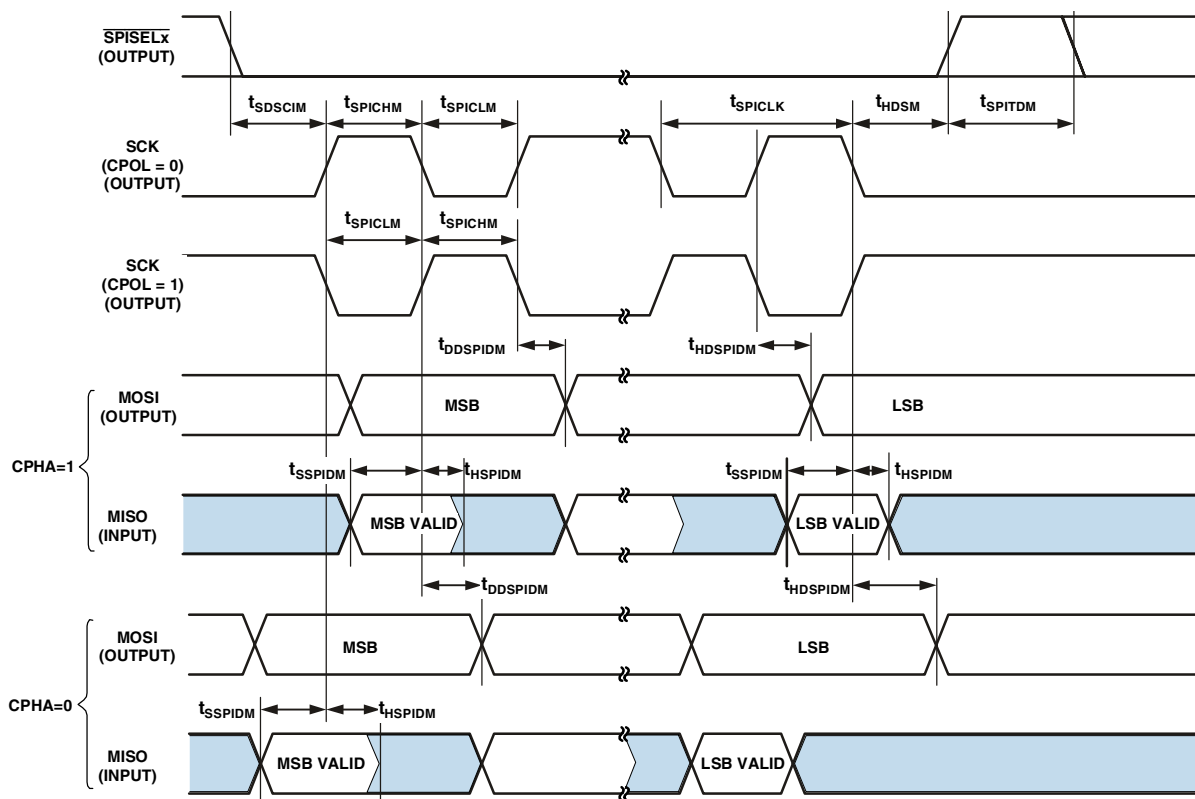


Figure 16. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 24 and Figure 17 describe SPI port slave operations.

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

| Parameter | Min | Max | Unit |
|---|-----------------|-----|------|
| <i>Timing Requirements</i> | | | |
| 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 |
| <i>Switching Characteristics</i> | | | |
| 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 |

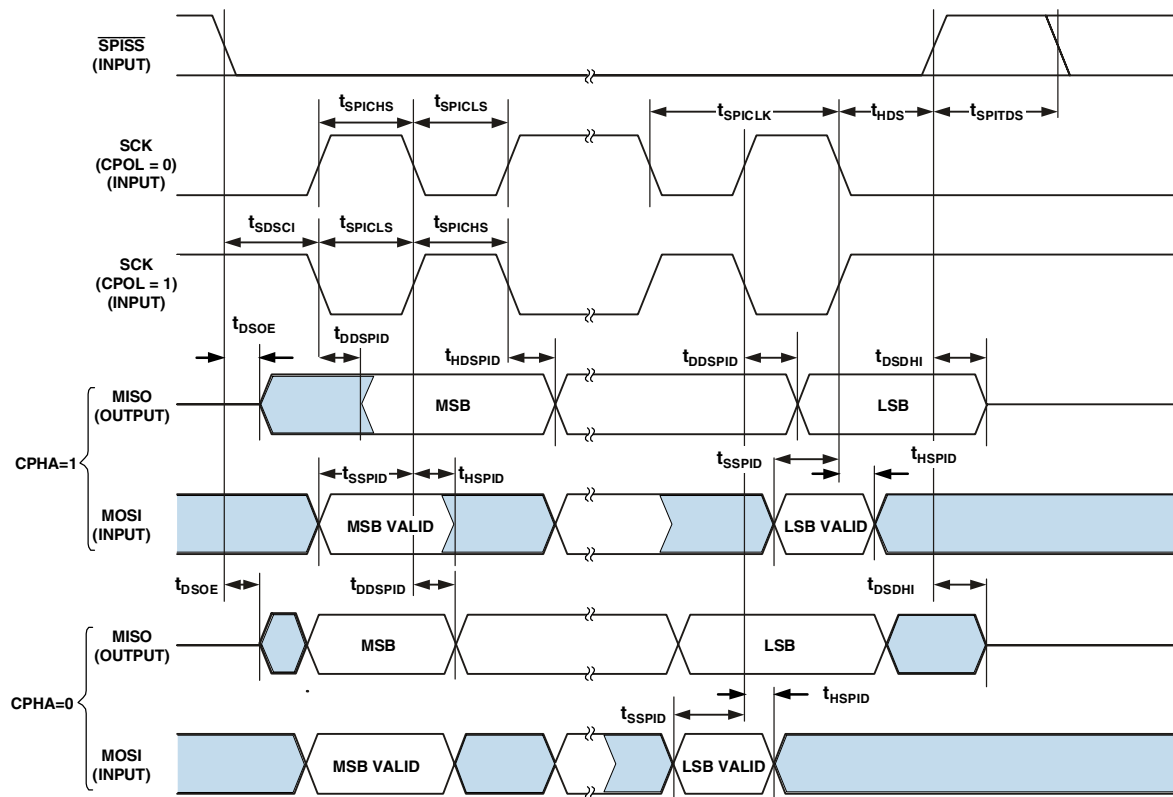


Figure 17. Serial Peripheral Interface (SPI) Port—Slave Timing

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

Figure 18 describes UART port receive and transmit operations. The maximum baud rate is $SCLK/16$. As shown in Figure 18 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.

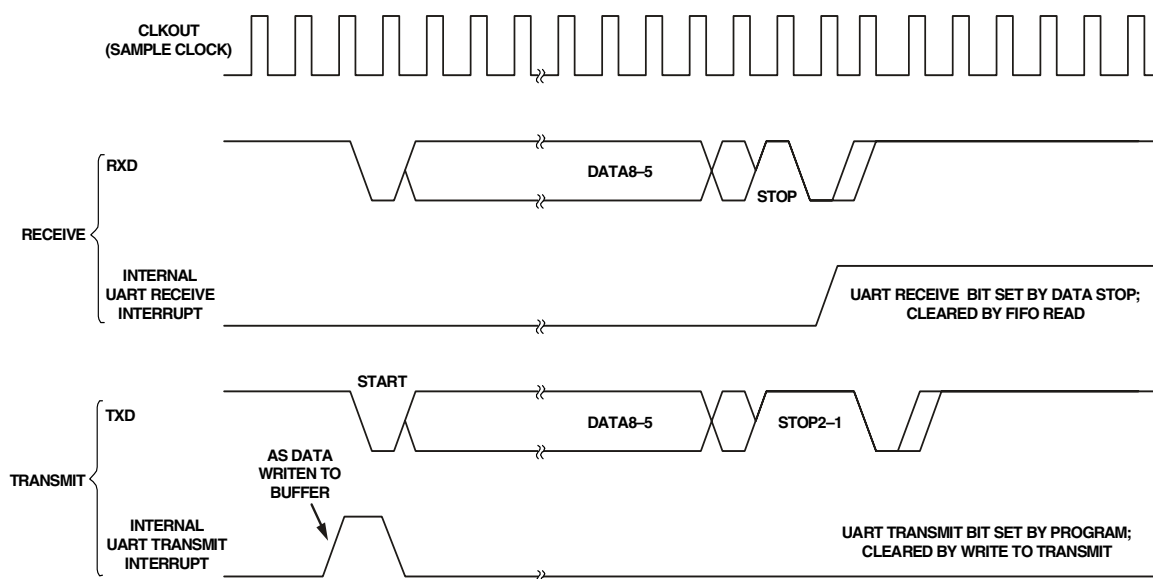


Figure 18. UART Port—Receive and Transmit Timing

Timer Cycle Timing

Table 25 and Figure 19 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 $f_{\text{SCLK}}/2$ MHz.

Table 25. Timer Cycle Timing

| Parameter | | Min | Max | Unit |
|--------------------------|--|-----|--------------|-------------|
| Timing Characteristics | | | | |
| t_{WL} | Timer Pulsewidth Input Low ¹ | 1 | | SCLK cycles |
| t_{WH} | Timer Pulsewidth Input High ¹ | 1 | | SCLK cycles |
| Switching Characteristic | | | | |
| t_{HTO} | Timer Pulsewidth Output ² | 1 | $(2^{32}-1)$ | SCLK cycles |

¹The minimum pulsewidths 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.
²The minimum time for t_{HTO} is one cycle, and the maximum time for t_{HTO} equals $(2^{32}-1)$ cycles.

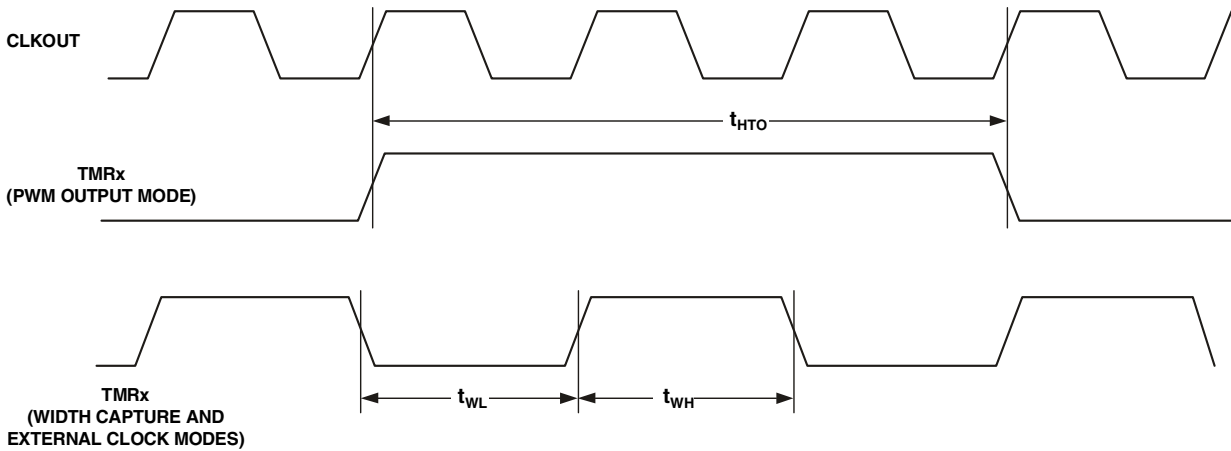


Figure 19. Timer PWM_OUT Cycle Timing

Programmable Flags Cycle Timing

Table 26 and Figure 20 describe programmable flag operations.

Table 26. Programmable Flags Cycle Timing

| Parameter | Min | Max | Unit |
|---|----------------|-----|------|
| <i>Timing Requirement</i> | | | |
| t_{WFI} Flag Input Pulsewidth | $t_{SCLK} + 1$ | | ns |
| <i>Switching Characteristic</i> | | | |
| t_{DFO} Flag Output Delay from CLKOUT Low | | 6 | ns |

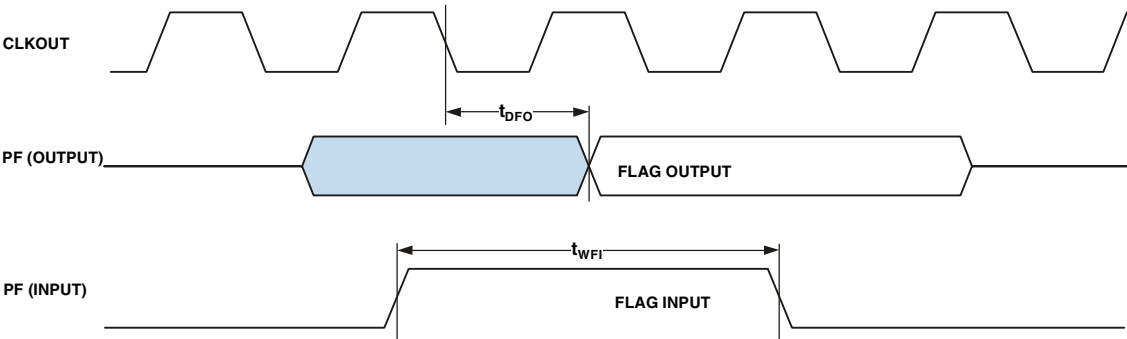


Figure 20. Programmable Flags Cycle Timing

JTAG Test and Emulation Port Timing

Table 27 and Figure 21 describe JTAG port operations.

Table 27. JTAG Port Timing

| Parameter | Min | Max | Unit |
|---|-----|-----|------------|
| <i>Timing Parameters</i> | | | |
| 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{TRST} Pulsewidth ² | 4 | | TCK cycles |
| <i>Switching Characteristics</i> | | | |
| t_{DIDO} 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, RESET, NMI0 and NMI1, BMODE1–0, BR, PPIxD7–0.

² 50 MHz max.

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

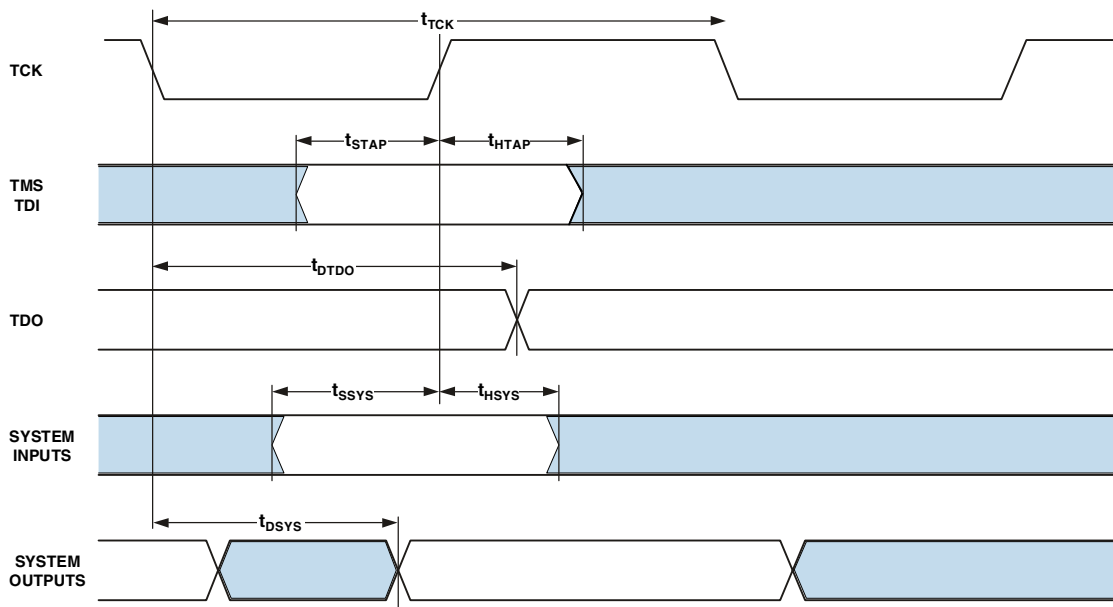


Figure 21. JTAG Port Timing

OUTPUT DRIVE CURRENTS

Figure 22 through Figure 29 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 16 to identify the driver type for a pin.

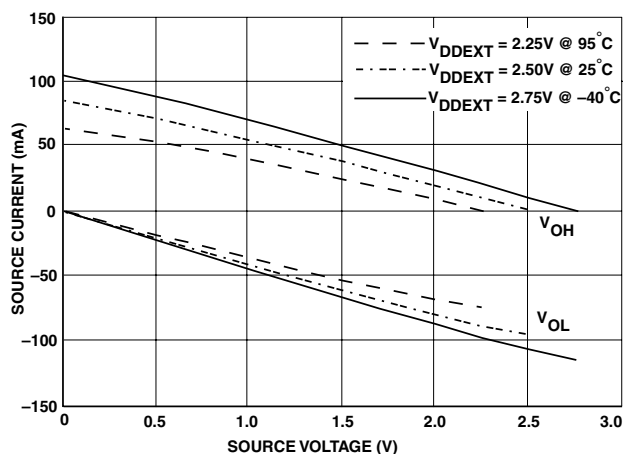


Figure 22. Drive Current A (Low V_{DDEXT})

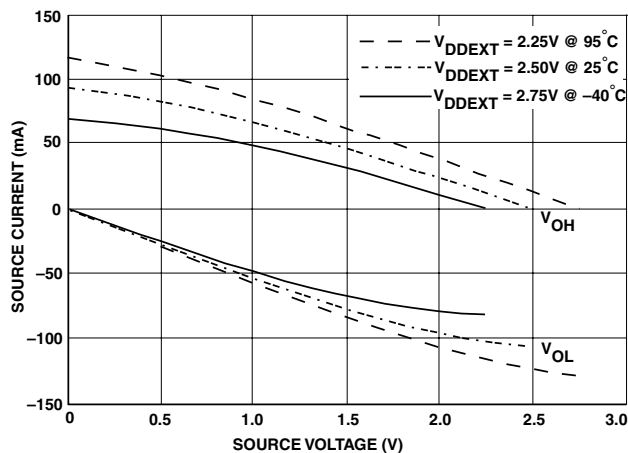


Figure 24. Drive Current B (Low V_{DDEXT})

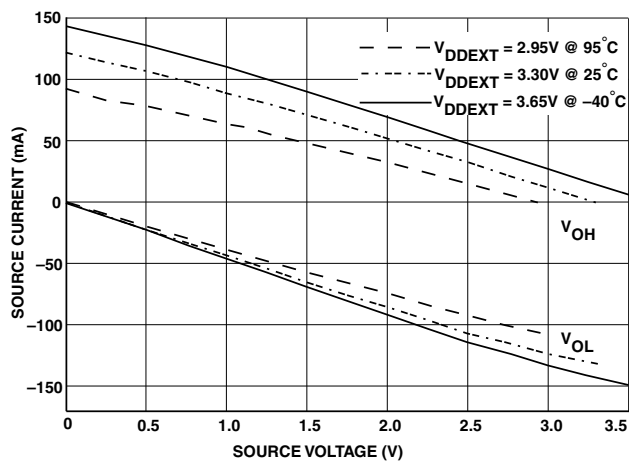


Figure 23. Drive Current A (High V_{DDEXT})

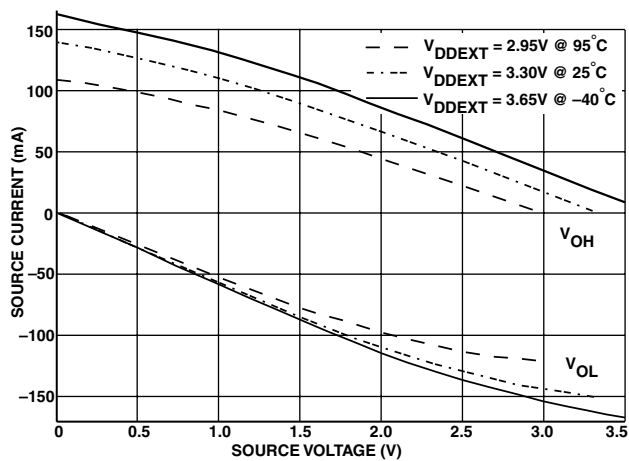


Figure 25. Drive Current B (High V_{DDEXT})

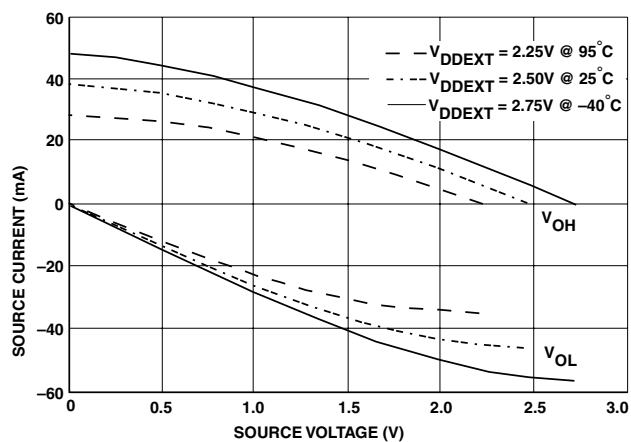


Figure 26. Drive Current C (Low V_{DDEXT})

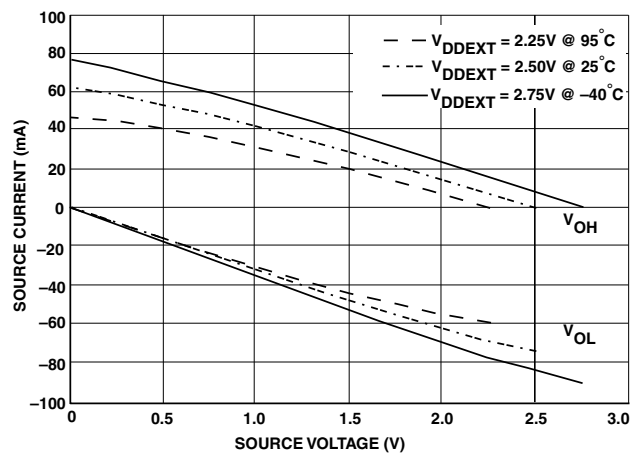


Figure 28. Drive Current D (Low V_{DDEXT})

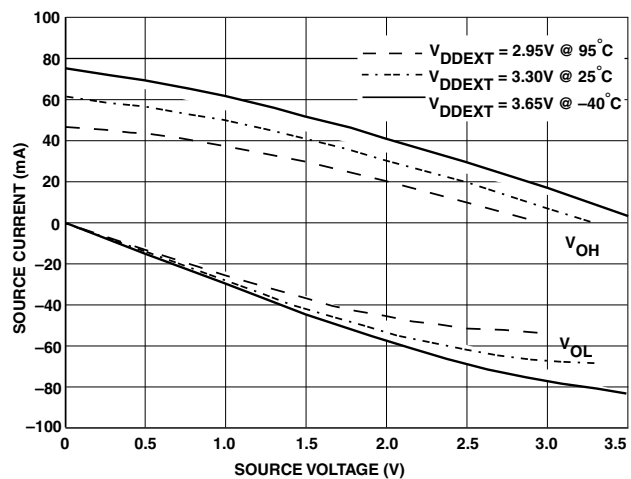


Figure 27. Drive Current C (High V_{DDEXT})

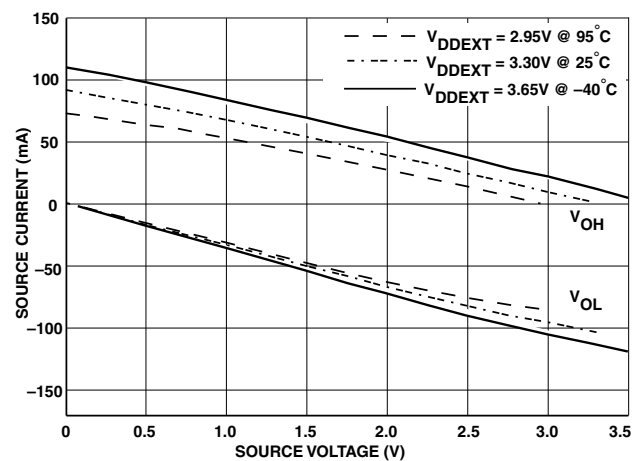


Figure 29. Drive Current D (High V_{DDEXT})

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 28 shows the power dissipation for internal circuitry (V_{DDINT}). Internal power dissipation is dependent on the instruction execution sequence and the data operands involved.

Table 28. Internal Power Dissipation

| Parameter | Test Conditions ¹ | | | Unit |
|--------------------------------|--|--|--|---------------|
| | $f_{CCLK} = 50 \text{ MHz}$ $V_{DDINT} = 0.8 \text{ V}$ | $f_{CCLK} = 500 \text{ MHz}$ $V_{DDINT} = 1.25 \text{ V}$ | $f_{CCLK} = 600 \text{ MHz}$ $V_{DDINT} = 1.25 \text{ V}$ | |
| I_{DDTYP} ² | 66 | 450 | 520 | mA |
| $I_{DDSLEEP}$ ³ | 30 | 91 | 91 | mA |
| $I_{DDDEEPSLEEP}$ ³ | 27 | 84 | 84 | mA |
| $I_{DDHIBERNATE}$ ⁴ | 50 | | | μA |

¹ I_{DD} data is specified for typical process parameters. All data at 25 °C.

² Processor executing 75% dual Mac, 25% ADD with moderate data bus activity.

³ See the ADSP-BF561 Blackfin Processor Hardware Reference Manual for definitions of Sleep and Deep Sleep operating modes.

⁴ Measured at $V_{DDEXT} = 3.65\text{V}$ with voltage regulator off ($V_{DDINT} = 0\text{V}$).

The external component of total power dissipation is caused by the switching of output pins. Its magnitude depends on

- The number of output pins that switch during each cycle (O)
- The maximum frequency at which they can switch (f)
- Their load capacitance (C)
- Their voltage swing (V_{DDEXT})

The external component is calculated using:

$$P_{EXT} = O \times C \times V_{DD}^2 \times f$$

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

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

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

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

TEST CONDITIONS

The ac signal specifications (timing parameters) appear in [Timing Specifications on Page 22](#). These include output disable time, output enable time, and capacitive loading. The timing specifications for the processor apply for the voltage reference levels in [Figure 32](#).

Output Enable Time

Output pins are considered to be enabled when they have made a transition from a high impedance state to the point when they start driving. The output enable time t_{ENA} is the interval from the point when a reference signal reaches a high or low voltage level to the point when the output starts driving as shown in the Output Enable/Disable diagram ([Figure 30](#)). The time $t_{ENA_MEASURED}$ is the interval from when the reference signal switches to when the output voltage reaches 2.0V (output high) or 1.0V (output low). Time t_{TRIP} is the interval from when the output starts driving to when the output reaches the 1.0V or 2.0V trip voltage. Time t_{ENA} is calculated as:

$$t_{ENA} = t_{ENA_MEASURED} - t_{TRIP}$$

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

Output Disable Time

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

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

The output disable time t_{DIS} is the difference between $t_{DIS_MEASURED}$ and t_{DECAY} as shown in [Figure 30](#). 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. t_{DECAY} is calculated with test loads C_L and I_L , and with ΔV equal to 0.5 V.

Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate t_{DECAY} using the equation given above. Choose ΔV to be the difference between the ADSP-BF561 output voltage and the input threshold for the device requiring the hold time. A typical ΔV will be 0.4 V. C_L is the total bus capacitance (per data line), and I_L is the total leakage or three-state current (per data line). The hold time will be t_{DECAY} plus the minimum disable time (i.e., t_{DSDAT} for an SDRAM write cycle).

Capacitive Loading

Output delays and holds are based on standard capacitive loads — 30 pF on all pins (see [Figure 31 on Page 42](#)). [Figure 33 on Page 43](#) to [Figure 40 on Page 44](#) show graphically how output

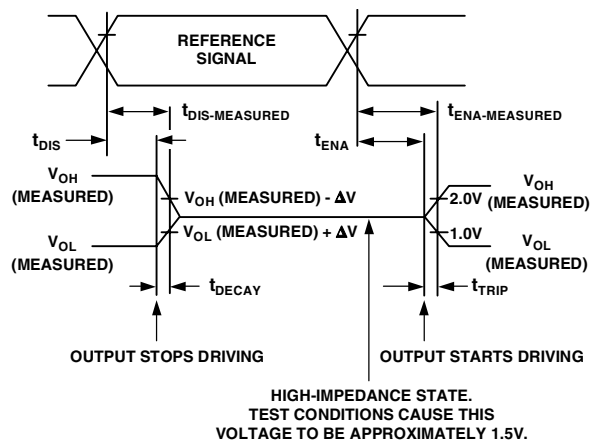


Figure 30. Output Enable/Disable

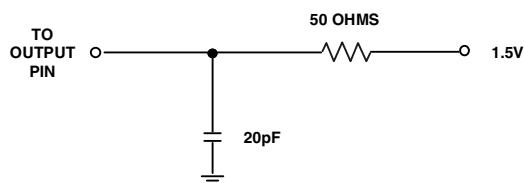


Figure 31. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

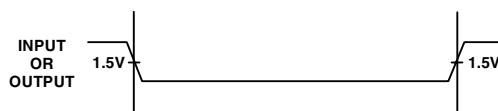


Figure 32. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

delays and holds vary with load capacitance (Note that these graphs or deratings do not apply to output disable delays; see [Output Disable Time on Page 42](#)). The graphs may not be linear outside the ranges shown.

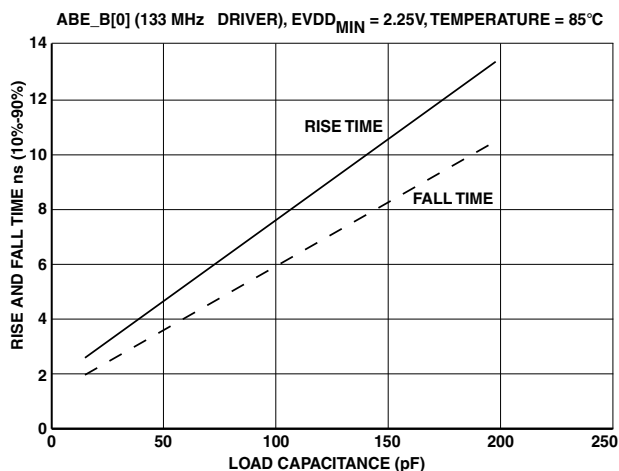


Figure 33. Typical Output Delay or Hold for Driver A at EVDD_{MIN}

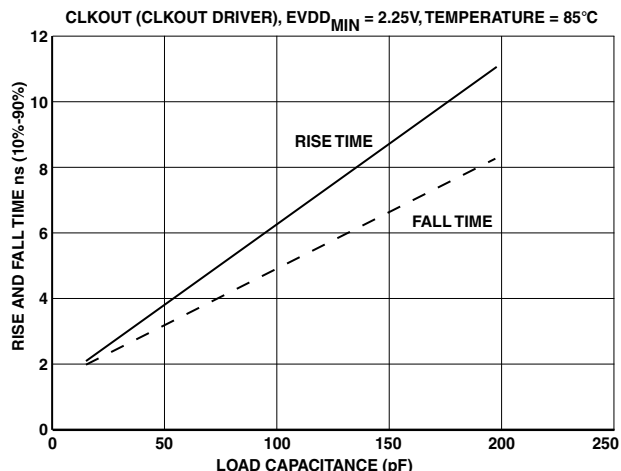


Figure 35. Typical Output Delay or Hold for Driver B at EVDD_{MIN}

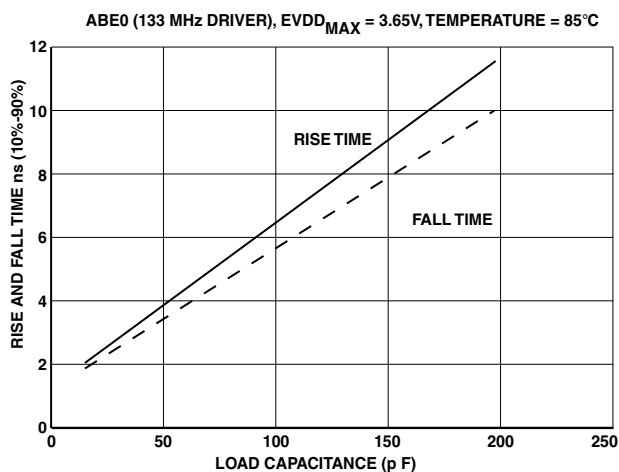


Figure 34. Typical Output Delay or Hold for Driver A at EVDD_{MAX}

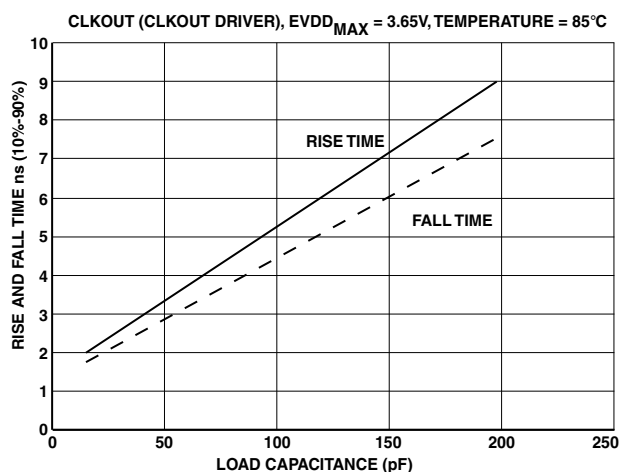
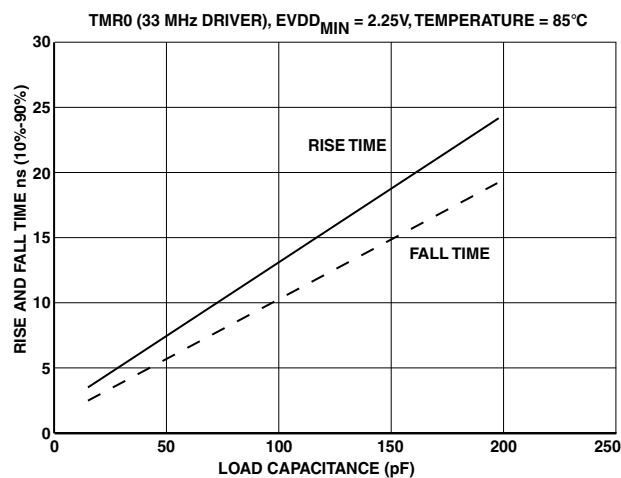
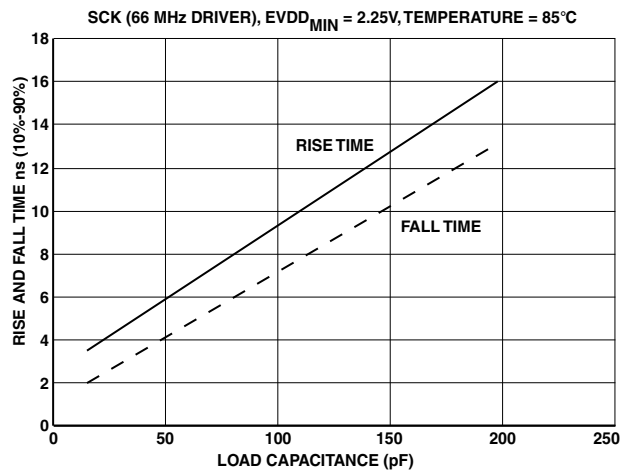
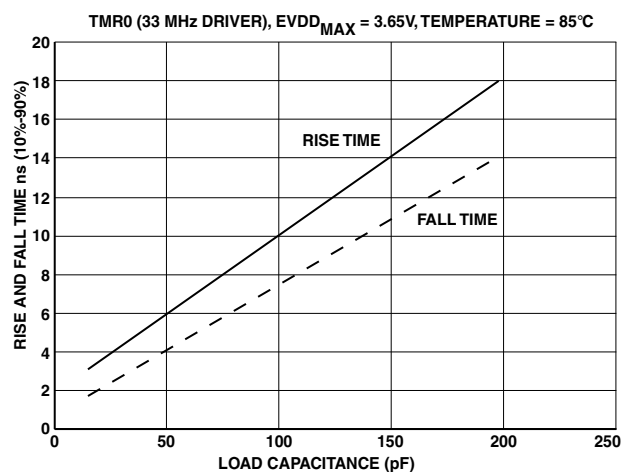
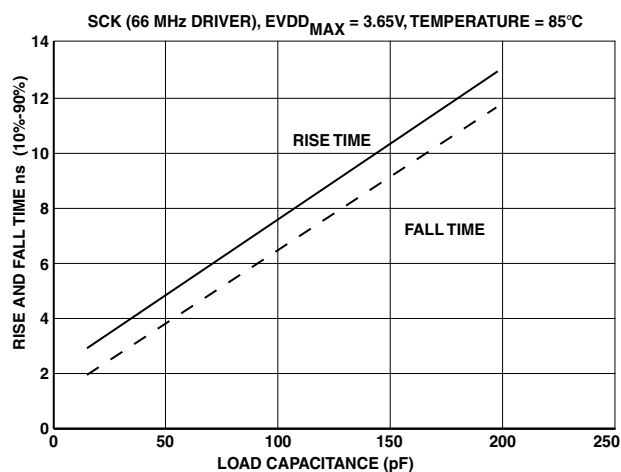


Figure 36. Typical Output Delay or Hold for Driver B at EVDD_{MAX}

Figure 37. Typical Output Delay or Hold for Driver C at $EVDD_{MIN}$ Figure 39. Typical Output Delay or Hold for Driver D at $EVDD_{MIN}$ Figure 38. Typical Output Delay or Hold for Driver C at $EVDD_{MAX}$ Figure 40. Typical Output Delay or Hold for Driver D at $EVDD_{MAX}$

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 ($^{\circ}\text{C}$)

T_{CASE} = Case temperature ($^{\circ}\text{C}$) measured by customer at top center of package.

Ψ_{JT} = From [Table 29](#) and [Table 30](#)

P_D = Power dissipation (see [Power Dissipation on Page 41](#) 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 ($^{\circ}\text{C}$)

In [Table 29](#) and [Table 30](#), 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 29](#) and [Table 30](#) 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 29. Thermal Characteristics for BC-256 Package

| Parameter | Condition | Typical | Unit |
|----------------|----------------------|---------|-----------------------------|
| θ_{JA} | 0 Linear m/s Airflow | 25.6 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JMA} | 1 Linear m/s Airflow | 22.4 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JMA} | 2 Linear m/s Airflow | 21.6 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JB} | Not applicable | 18.9 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JC} | Not applicable | 4.85 | $^{\circ}\text{C}/\text{W}$ |
| Ψ_{JT} | 0 Linear m/s Airflow | 0.15 | $^{\circ}\text{C}/\text{W}$ |

Table 30. Thermal Characteristics for B-297 Package

| Parameter | Condition | Typical | Unit |
|----------------|----------------------|---------|-----------------------------|
| θ_{JA} | 0 Linear m/s Airflow | 20.6 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JMA} | 1 Linear m/s Airflow | 17.8 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JMA} | 2 Linear m/s Airflow | 17.4 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JB} | Not applicable | 16.3 | $^{\circ}\text{C}/\text{W}$ |
| θ_{JC} | Not applicable | 7.15 | $^{\circ}\text{C}/\text{W}$ |
| Ψ_{JT} | 0 Linear m/s Airflow | 0.37 | $^{\circ}\text{C}/\text{W}$ |

256-BALL MBGA PINOUT

Table 31. 256-Ball MBGA Pin Assignment (Numerically by Ball Number)

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|----------|---------------------------------------|----------|---------------------------------------|----------|--------------------------|----------|--------------------------|
| A01 | VDDEXT | B01 | PPI1CLK | C01 | PPI0SYNC2/TMR9 | D01 | PPI0D13/PF45 |
| A02 | ADDR24 | B02 | ADDR22 | C02 | PPI0CLK | D02 | PPI0D15/PF47 |
| A03 | ADDR20 | B03 | ADDR18 | C03 | ADDR25 | D03 | PPI0SYNC3 |
| A04 | VDDEXT | B04 | ADDR16 | C04 | ADDR19 | D04 | ADDR23 |
| A05 | ADDR14 | B05 | ADDR12 | C05 | GND | D05 | GND |
| A06 | ADDR10 | B06 | VDDEXT | C06 | ADDR11 | D06 | GND |
| A07 | $\overline{\text{AMS3}}$ | B07 | $\overline{\text{AMST}}$ | C07 | $\overline{\text{AOE}}$ | D07 | ADDR09 |
| A08 | $\overline{\text{AWE}}$ | B08 | $\overline{\text{ARE}}$ | C08 | $\overline{\text{AMS0}}$ | D08 | GND |
| A09 | VDDEXT | B09 | $\overline{\text{SMS1}}$ | C09 | $\overline{\text{SMS2}}$ | D09 | ARDY |
| A10 | $\overline{\text{SMS3}}$ | B10 | SCKE | C10 | $\overline{\text{SRAS}}$ | D10 | $\overline{\text{SCAS}}$ |
| A11 | SCLK0/CLKOUT | B11 | VDDEXT | C11 | GND | D11 | SA10 |
| A12 | SCLK1 | B12 | $\overline{\text{BR}}$ | C12 | $\overline{\text{BGH}}$ | D12 | VDDEXT |
| A13 | $\overline{\text{BG}}$ | B13 | $\overline{\text{ABE1}}/\text{SDQM1}$ | C13 | GND | D13 | ADDR02 |
| A14 | $\overline{\text{ABE2}}/\text{SDQM2}$ | B14 | ADDR06 | C14 | ADDR07 | D14 | GND |
| A15 | $\overline{\text{ABE3}}/\text{SDQM3}$ | B15 | ADDR04 | C15 | DATA1 | D15 | DATA5 |
| A16 | VDDEXT | B16 | DATA0 | C16 | DATA3 | D16 | DATA6 |
| E01 | GND | F01 | CLKIN | G01 | XTAL | H01 | GND |
| E02 | PPI0D11/PF43 | F02 | VDDEXT | G02 | GND | H02 | GND |
| E03 | PPI0D12/PF44 | F03 | $\overline{\text{RESET}}$ | G03 | VDDEXT | H03 | PPI0D9/PF41 |
| E04 | PPI0SYNC1/TMR8 | F04 | PPI0D10/PF42 | G04 | BYPASS | H04 | PPI0D7 |
| E05 | ADDR15 | F05 | ADDR21 | G05 | PPI0D14/PF46 | H05 | PPI0D5 |
| E06 | ADDR13 | F06 | ADDR17 | G06 | GND | H06 | VDDINT |
| E07 | $\overline{\text{AMS2}}$ | F07 | VDDINT | G07 | GND | H07 | VDDINT |
| E08 | VDDINT | F08 | GND | G08 | GND | H08 | GND |
| E09 | $\overline{\text{SMS0}}$ | F09 | VDDINT | G09 | VDDINT | H09 | GND |
| E10 | $\overline{\text{SWE}}$ | F10 | GND | G10 | ADDR05 | H10 | GND |
| E11 | $\overline{\text{ABE0}}/\text{SDQM0}$ | F11 | ADDR08 | G11 | ADDR03 | H11 | VDDINT |
| E12 | DATA2 | F12 | DATA10 | G12 | DATA15 | H12 | DATA16 |
| E13 | GND | F13 | DATA8 | G13 | DATA14 | H13 | DATA18 |
| E14 | DATA4 | F14 | DATA12 | G14 | GND | H14 | DATA20 |
| E15 | DATA7 | F15 | DATA9 | G15 | DATA13 | H15 | DATA17 |
| E16 | VDDEXT | F16 | DATA11 | G16 | VDDEXT | H16 | DATA19 |

Table 31. 256-Ball MBGA Pin Assignment (Numerically by Ball Number) (Continued)

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|----------|------------------|----------|--------------------------|----------|-------------------------|----------|------------------|
| J01 | VROUT0 | K01 | PPI0D6 | L01 | PPI0D0 | M01 | PPI1D15/PF39 |
| J02 | VROUT1 | K02 | PPI0D4 | L02 | PPI1SYNC2/TMR11 | M02 | PPI1D13/PF37 |
| J03 | PPI0D2 | K03 | PPI0D8/PF40 | L03 | GND | M03 | PPI1D9/PF33 |
| J04 | PPI0D3 | K04 | PPI1SYNC1/TMR10 | L04 | PPI1SYNC3 | M04 | GND |
| J05 | PPI0D1 | K05 | PPI1D14/PF38 | L05 | VDDEXT | M05 | NC |
| J06 | VDDEXT | K06 | VDDEXT | L06 | PPI1D11/PF35 | M06 | PF3/SPISEL3/TMR3 |
| J07 | GND | K07 | GND | L07 | GND | M07 | PF7/SPISEL7/TMR7 |
| J08 | VDDINT | K08 | VDDINT | L08 | VDDINT | M08 | VDDINT |
| J09 | VDDINT | K09 | GND | L09 | GND | M09 | GND |
| J10 | VDDINT | K10 | GND | L10 | VDDEXT | M10 | BMODE0 |
| J11 | GND | K11 | VDDINT | L11 | GND | M11 | SCK |
| J12 | DATA30 | K12 | DATA28 | L12 | DR0PRI | M12 | DR1PRI |
| J13 | DATA22 | K13 | DATA26 | L13 | TFS0/PF16 | M13 | NC |
| J14 | GND | K14 | DATA24 | L14 | GND | M14 | VDDEXT |
| J15 | DATA21 | K15 | DATA25 | L15 | DATA27 | M15 | DATA31 |
| J16 | DATA23 | K16 | VDDEXT | L16 | DATA29 | M16 | DT0PRI/PF18 |
| N01 | PPI1D12/PF36 | P01 | PPI1D8/PF32 | R01 | PPI1D7 | T01 | VDDEXT |
| N02 | PPI1D10/PF34 | P02 | GND | R02 | PPI1D6 | T02 | PPI1D4 |
| N03 | PPI1D3 | P03 | PPI1D5 | R03 | PPI1D2 | T03 | VDDEXT |
| N04 | PPI1D1 | P04 | PF0/SPISS/TMR0 | R04 | PPI1D0 | T04 | PF2/SPISEL2/TMR2 |
| N05 | PF1/SPISEL1/TMR1 | P05 | GND | R05 | PF4/SPISEL4/TMR4 | T05 | PF6/SPISEL6/TMR6 |
| N06 | PF9 | P06 | PF5/SPISEL5/TMR5 | R06 | PF8 | T06 | VDDEXT |
| N07 | GND | P07 | PF11 | R07 | PF10 | T07 | PF12 |
| N08 | PF13 | P08 | PF15/EXTCLK | R08 | PF14 | T08 | VDDEXT |
| N09 | TDO | P09 | GND | R09 | NMI1 | T09 | TCK |
| N10 | BMODE1 | P10 | $\overline{\text{TRST}}$ | R10 | TDI | T10 | TMS |
| N11 | MOSI | P11 | NMI0 | R11 | $\overline{\text{EMU}}$ | T11 | SLEEP |
| N12 | GND | P12 | GND | R12 | MISO | T12 | VDDEXT |
| N13 | RFS1/PF24 | P13 | RSCLK1/PF30 | R13 | TX/PF26 | T13 | RX/PF27 |
| N14 | GND | P14 | TFS1/PF21 | R14 | TSCLK1/PF31 | T14 | DR1SEC/PF25 |
| N15 | DT0SEC/PF17 | P15 | RSCLK0/PF28 | R15 | DT1PRI/PF23 | T15 | DT1SEC/PF22 |
| N16 | TSCLK0/PF29 | P16 | DR0SEC/PF20 | R16 | RFS0/PF19 | T16 | VDDEXT |

297-BALL PBGA PINOUT

Table 32. 297-Ball PBGA Pin Assignment (Numerically by Ball Number)

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|----------|--------------------------------|----------|------------------|----------|--------------------------|----------|--------------------------------|
| A01 | GND | AB03 | GND | AE11 | PF12 | AF17 | SLEEP |
| A02 | ADDR25 | AB24 | GND | AE12 | PF14 | AF18 | NMI0 |
| A03 | ADDR23 | AB25 | TFS0/PF16 | AE13 | NC | AF19 | SCK |
| A04 | ADDR21 | AB26 | DR0PRI | AE14 | TDO | AF20 | TX/PF26 |
| A05 | ADDR19 | AC01 | PPI1D9/PF33 | AE15 | $\overline{\text{TRST}}$ | AF21 | RSCLK1/PF30 |
| A06 | ADDR17 | AC02 | PPI1D8/PF32 | AE16 | $\overline{\text{EMU}}$ | AF22 | DR1PRI |
| A07 | ADDR15 | AC03 | GND | AE17 | BMODE1 | AF23 | TSCLK1/PF31 |
| A08 | ADDR13 | AC04 | GND | AE18 | BMODE0 | AF24 | DT1SEC/PF22 |
| A09 | ADDR11 | AC23 | GND | AE19 | MISO | AF25 | DT1PRI/PF23 |
| A10 | ADDR09 | AC24 | GND | AE20 | MOSI | AF26 | GND |
| A11 | $\overline{\text{AMS3}}$ | AC25 | DR0SEC/PF20 | AE21 | RX/PF27 | B01 | PPI1CLK |
| A12 | $\overline{\text{AMST}}$ | AC26 | RFS0/PF19 | AE22 | RFS1/PF24 | B02 | GND |
| A13 | $\overline{\text{AWE}}$ | AD01 | PPI1D7 | AE23 | DR1SEC/PF25 | B03 | ADDR24 |
| A14 | $\overline{\text{ARE}}$ | AD02 | PPI1D6 | AE24 | TFS1/PF21 | B04 | ADDR22 |
| A15 | $\overline{\text{SMS0}}$ | AD03 | GND | AE25 | GND | B05 | ADDR20 |
| A16 | $\overline{\text{SMS2}}$ | AD04 | GND | AE26 | NC | B06 | ADDR18 |
| A17 | $\overline{\text{SRA5}}$ | AD05 | GND | AF01 | GND | B07 | ADDR16 |
| A18 | $\overline{\text{SCA5}}$ | AD22 | GND | AF02 | PPI1D4 | B08 | ADDR14 |
| A19 | SCLK0/CLKOUT | AD23 | GND | AF03 | PPI1D2 | B09 | ADDR12 |
| A20 | SCLK1 | AD24 | GND | AF04 | PPI1D0 | B10 | ADDR10 |
| A21 | $\overline{\text{BGH}}$ | AD25 | NC | AF05 | PF1/SPISEL1/TMR1 | B11 | AMS2 |
| A22 | $\overline{\text{ABE0/SDQM0}}$ | AD26 | RSCLK0/PF28 | AF06 | PF3/SPISEL3/TMR3 | B12 | AMS0 |
| A23 | $\overline{\text{ABE2/SDQM2}}$ | AE01 | PPI1D5 | AF07 | PF5/SPISEL5/TMR5 | B13 | AOE |
| A24 | ADDR08 | AE02 | GND | AF08 | PF7/SPISEL7/TMR7 | B14 | ARDY |
| A25 | ADDR06 | AE03 | PPI1D3 | AF09 | PF9 | B15 | SMS1 |
| A26 | GND | AE04 | PPI1D1 | AF10 | PF11 | B16 | SMS3 |
| AA01 | PPI1D13/PF37 | AE05 | PF0/SPISS/TMR0 | AF11 | PF13 | B17 | SCKE |
| AA02 | PPI1D12/PF36 | AE06 | PF2/SPISEL2/TMR2 | AF12 | PF15/EXT CLK | B18 | SWE |
| AA25 | DT0SEC/PF17 | AE07 | PF4/SPISEL4/TMR4 | AF13 | NMI1 | B19 | SA10 |
| AA26 | TSCLK0/PF29 | AE08 | PF6/SPISEL6/TMR6 | AF14 | TCK | B20 | BR |
| AB01 | PPI1D11/PF35 | AE09 | PF8 | AF15 | TDI | B21 | BG |
| AB02 | PPI1D10/PF34 | AE10 | PF10 | AF16 | TMS | B22 | $\overline{\text{ABE1/SDQM1}}$ |

Table 32. 297-Ball PBGA Pin Assignment (Numerically by Ball Number)

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|----------|----------------|----------|--------------|----------|-------------|----------|-----------------|
| B23 | ABE3/SDQM3 | G01 | PPIOD11/PF43 | K25 | DATA10 | N12 | GND |
| B24 | ADDR07 | G02 | PPIOD10/PF42 | K26 | DATA13 | N13 | GND |
| B25 | GND | G25 | DATA4 | L01 | NC | N14 | GND |
| B26 | ADDR05 | G26 | DATA7 | L02 | NC | N15 | GND |
| C01 | PPI0SYNC3 | H01 | BYPASS | L10 | VDDEXT | N16 | GND |
| C02 | PPI0CLK | H02 | RESET | L11 | GND | N17 | GND |
| C03 | GND | H25 | DATA6 | L12 | GND | N18 | VDDINT |
| C04 | GND | H26 | DATA9 | L13 | GND | N25 | DATA16 |
| C05 | GND | J01 | CLKIN | L14 | GND | N26 | DATA19 |
| C22 | GND | J02 | GND | L15 | GND | P01 | PPIOD7 |
| C23 | GND | J10 | VDDEXT | L16 | GND | P02 | PPIOD8/PF40 |
| C24 | GND | J11 | VDDEXT | L17 | GND | P10 | VDDEXT |
| C25 | ADDR04 | J12 | VDDEXT | L18 | VDDINT | P11 | GND |
| C26 | ADDR03 | J13 | VDDEXT | L25 | DATA12 | P12 | GND |
| D01 | PPI0SYNC1/TMR8 | J14 | VDDEXT | L26 | DATA15 | P13 | GND |
| D02 | PPI0SYNC2/TMR9 | J15 | VDDEXT | M01 | VROUT0 | P14 | GND |
| D03 | GND | J16 | VDDINT | M02 | GND | P15 | GND |
| D04 | GND | J17 | VDDINT | M10 | VDDEXT | P16 | GND |
| D23 | GND | J18 | VDDINT | M11 | GND | P17 | GND |
| D24 | GND | J25 | DATA8 | M12 | GND | P18 | VDDINT |
| D25 | ADDR02 | J26 | DATA11 | M13 | GND | P25 | DATA18 |
| D26 | DATA1 | K01 | XTAL | M14 | GND | P26 | DATA21 |
| E01 | PPIOD15/PF47 | K02 | NC | M15 | GND | R01 | PPIOD5 |
| E02 | PPIOD14/PF46 | K10 | VDDEXT | M16 | GND | R02 | PPIOD6 |
| E03 | GND | K11 | VDDEXT | M17 | GND | R10 | VDDEXT |
| E24 | GND | K12 | VDDEXT | M18 | VDDINT | R11 | GND |
| E25 | DATA0 | K13 | VDDEXT | M25 | DATA14 | R12 | GND |
| E26 | DATA3 | K14 | VDDEXT | M26 | DATA17 | R13 | GND |
| F01 | PPIOD13/PF45 | K15 | VDDEXT | N01 | VROUT1 | R14 | GND |
| F02 | PPIOD12/PF44 | K16 | VDDINT | N02 | PPIOD9/PF41 | R15 | GND |
| F25 | DATA2 | K17 | VDDINT | N10 | VDDEXT | R16 | GND |
| F26 | DATA5 | K18 | VDDINT | N11 | GND | R17 | GND |
| R18 | VDDINT | T16 | GND | U14 | GND | W01 | PPI1SYNC1/TMR10 |
| R25 | DATA20 | T17 | GND | U15 | VDDINT | W02 | PPI1SYNC2/TMR11 |
| R26 | DATA23 | T18 | VDDINT | U16 | VDDINT | W25 | DATA28 |
| T01 | PPIOD3 | T25 | DATA22 | U17 | VDDINT | W26 | DATA31 |
| T02 | PPIOD4 | T26 | DATA25 | U18 | VDDINT | Y01 | PPI1D15/PF39 |
| T10 | VDDEXT | U01 | PPIOD1 | U25 | DATA24 | Y02 | PPI1D14/PF38 |
| T11 | GND | U02 | PPIOD2 | U26 | DATA27 | Y25 | DATA30 |
| T12 | GND | U10 | VDDEXT | V01 | PPI1SYNC3 | Y26 | DTOPRI/PF18 |
| T13 | GND | U11 | VDDEXT | V02 | PPIOD0 | | |
| T14 | GND | U12 | VDDEXT | V25 | DATA26 | | |
| T15 | GND | U13 | VDDEXT | V26 | DATA29 | | |

OUTLINE DIMENSIONS

Dimensions in the outline dimension figure are shown in millimeters.



256-BALL MINI BGA
(BC-256)

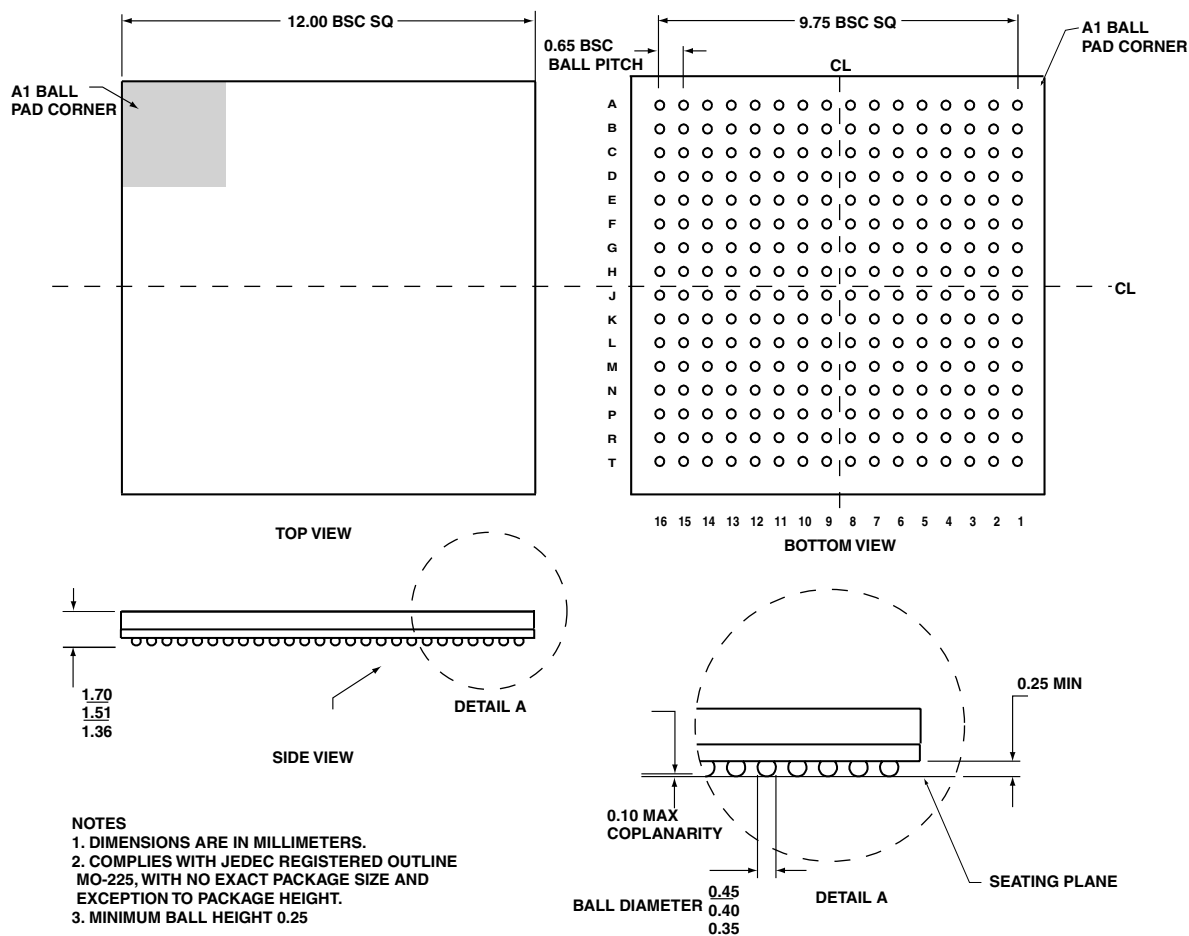


Figure 41. 256-Ball Mini-Ball Grid Array

OUTLINE DIMENSIONS

Dimensions in the outline dimension figure are shown in millimeters.

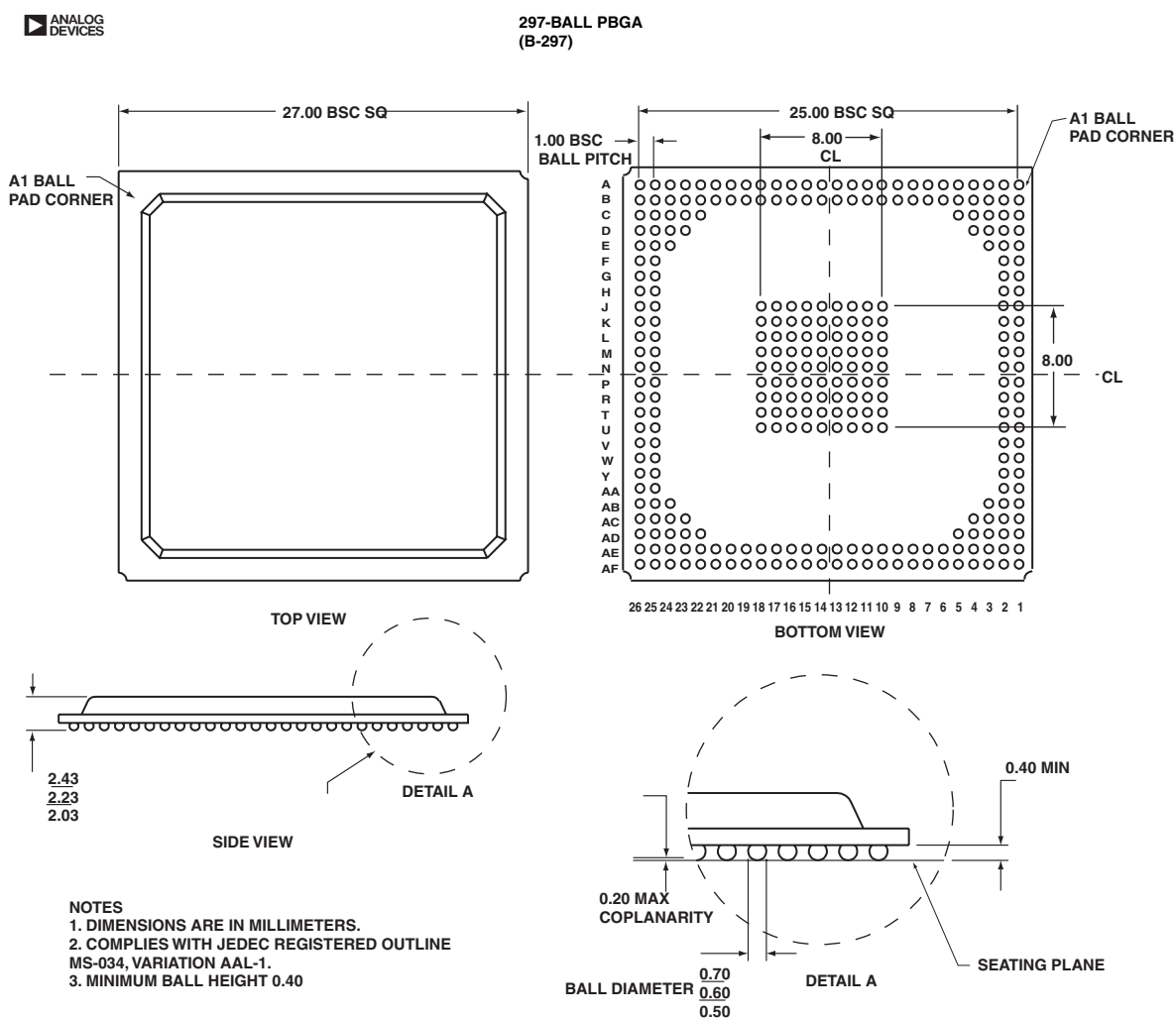


Figure 42. 297-Ball PBGA Grid Array

ORDERING GUIDE

| Part Number | Temperature Range (Ambient) | Package Description | Instruction Rate (Max) | Operating Voltage |
|---------------------------------|-----------------------------|--------------------------------------|------------------------|-------------------------------------|
| ADSP-BF561SKBCZ600 ¹ | 0°C to +70°C | Ball Grid Array (Mini-BGA) BC-256 | 600 MHz | 1.25 V internal, 2.5 V or 3.3 V I/O |
| ADSP-BF561SKBCZ500 ¹ | 0°C to +70°C | Ball Grid Array (Mini-BGA) BC-256 | 500 MHz | 1.25 V internal, 2.5 V or 3.3 V I/O |
| ADSP-BF561SBB600 | -40°C to +85°C | 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 | Plastic Ball Grid Array (PBGA) B-297 | 500 MHz | 1.25 V internal, 2.5 V or 3.3 V I/O |

¹Z = Pb-free part.

