ANALOG DEVICES

Blackfin®
Embedded Processor

Preliminary Technical Data

ADSP-BF542/BF544/BF548/BF549

FEATURES

Up to 600 MHz High-Performance Blackfin Processor
- **Two 16-Bit MACs, Two 40-Bit ALUs, Four 8-Bit Video ALUs**
- **RISC-Like Register and Instruction Model**

0.8 V to TBD V Core VDD with On-chip Voltage Regulation
2.5 V and 3.3 V-Tolerant I/O with Specific 5 V-Tolerant Pins
400-ball Lead-Free mBGA Package

MEMORY

Up to 324K bytes of on-chip memory comprised of:
- **Instruction SRAM/cache; instruction SRAM; data SRAM/cache; additional dedicated data SRAM; scratchpad SRAM (see Table 1 on Page 3 for available memory configurations**

External Sync Memory Controller Supporting DDR/Mobile DDR SDRAM
External Async Memory Controller Supporting 8/16 bit Async Memories and Burst Flash Devices
NAND Flash Controller
Four Memory-to-Memory DMA pairs, two with external requests
Memory Management Unit Providing Memory Protection
Flexible Booting Options
Code Security with Lockbox™ Secure Technology
One-Time-Programmable (OTP) Memory

PERIPHERALS

High-Speed USB On-the-Go (OTG) with Integrated PHY
SD/SDIO Controller
ATA/ATAPI-6 Controller
Up to Four Synchronous Serial Ports (SPORTs)
Up to Three Serial Peripheral Interfaces (SPI-Compatible)
Up to Four UARTs, Two with Automatic Hardware Flow Control
Up to Two CAN (Controller Area Network) 2.0B Interfaces
Up to Two TWI (Two-Wire Interface) Controllers
8- or 16-Bit Asynchronous Host DMA Interface
Multiple Enhanced Parallel Peripheral Interfaces (PPI), Supporting ITU-R BT.656 Video Formats and 18/24-bit LCD Connections
Media Transceiver (MXVR) for connection to a MOST® Network
Pixel Compositor for overlays, alpha blending, and color conversion
Up to Eleven 32-Bit Timers/Counters with PWM Support
Real-Time Clock (RTC) and Watchdog Timer
Up/Down Counter With Support for Rotary Encoder
Up to 152 General Purpose I/O (GPIOs)
On-Chip PLL Capable of 1x to 63x Frequency Multiplication
Debug/JTAG Interface

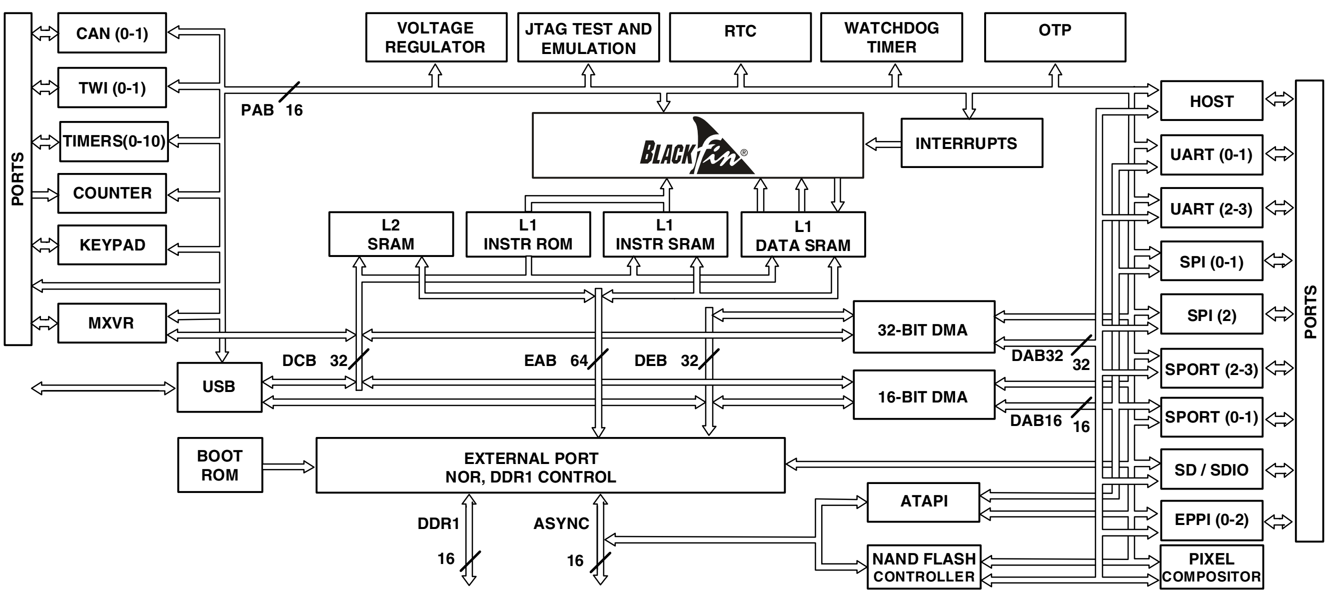


Figure 1. Functional Block Diagram

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

Rev. PrD

One Technology Way, P.O.Box 9106, Norwood, MA 02062-9106 U.S.A.
Tel:781/329-4700 www.analog.com
Fax:781/461-3113

TABLE OF CONTENTS

REVISION HISTORY

Revision PrD: Corrections and additions to PrC.

Changes are minor and appear throughout these main sections:

GENERAL DESCRIPTION

The ADSP-BF54x processors are members of the Blackfin family of products, incorporating the Analog Devices/Intel Micro Signal Architecture (MSA). Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction-set architecture.

Specific performance and memory configurations for ADSP-BF54x processors are shown in Table 1.

Table 1. ADSP-BF54x Processor Features

Processor Features		ADSP-BF549	ADSP-BF548	ADSP-BF544	ADSP-BF542
Code Security		1	1	1	1
SD/SDIO		1	1	–	1
Pixel Compositor		1	1	1	1
18- or 24-bit PPI0 with LCD		1	1	1	–
16-bit PPI1, 8-bit PPI2		1	1	1	1
Host DMA Port		1	1	1	–
DDR/NAND Flash		1	1	1	1
ATAPI		1	1	–	1
USB 2.0 HS OTG		1	1	–	1
Keyscan		1	1	–	1
MXVR		1	–	–	–
CAN ports		2	2	2	1
TWI ports		2	2	2	1
SPI ports		3	3	2	2
UART ports		4	4	3	3
SPORTs		4	4	3	3
General-purpose counter		1	1	1	1
Timers		8	8	11	8
General-purpose I/O pins		152	152	152	152
Memory Configurations (K Bytes)	L1 Instruction SRAM/Cache	16	16	16	16
	L1 Instruction SRAM	48	48	48	48
	L1 Data SRAM/Cache	32	32	32	32
	L1 Data SRAM	32	32	32	32
	L1 Scratchpad SRAM	4	4	4	4
	L1 ROM[1]	64	64	64	64
	L2	128	128	64	–
	L3 Boot ROM[1]	4	4	4	4
Maximum Core Instruction Rate (MHz)		533	600	533	600

[1] This ROM is not customer configurable.

The ADSP-BF542/BF544/BF548 processors are completely code and pin compatible. They differ only with respect to their performance, on-chip memory, and selection of I/O peripherals. Specific performance, memory, and feature configurations, are shown in Table 1. The ADSP-BF549 is completely code compatible with the other ADSP-BF54x processors, and this processor is pin compatible, *except* for the location of the HWAIT pin.

By integrating a rich set of industry-leading system peripherals and memory, Blackfin processors are the platform of choice for next-generation applications that require RISC-like programmability, multimedia support and leading-edge signal processing in one integrated package.

LOW-POWER ARCHITECTURE

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

SYSTEM INTEGRATION

The ADSP-BF54x processors are highly integrated system-on-a-chip solutions for the next generation of embedded network connected applications. By combining industry-standard interfaces with a high performance signal processing core, users can develop cost-effective solutions quickly without the need for costly external components. The system peripherals include a high speed USB OTG (On-The-Go) controller with integrated PHY, CAN 2.0B controllers, TWI controllers, UART ports, SPI ports, serial ports (SPORTs), ATAPI controller, SD/SDIO controller, a real-time clock, a watchdog timer, LCD controller, and multiple enhanced parallel peripheral interfaces.

ADSP-BF54X PROCESSOR PERIPHERALS

The ADSP-BF54x processor contains a rich set of peripherals connected to the core via several high bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see Figure 1 on Page 1). The general-purpose peripherals include functions such as UARTs, SPI, TWI, timers with pulse width modulation (PWM) and pulse measurement capability, general purpose I/O pins, a real-time clock, and a watchdog timer. This set of functions satisfies a wide variety of typical system support needs and is augmented by the system expansion capabilities of the part. The ADSP-BF54x processor contains dedicated network communication modules and high-speed serial and parallel ports, an interrupt controller for flexible management of interrupts from the on-chip peripherals or external sources, and power management control functions to tailor the performance and power characteristics of the processor and system to many application scenarios.

All of the peripherals, except for general-purpose I/O, CAN, TWI, real-time clock, and timers, are supported by a flexible DMA structure. There are also separate memory DMA channels dedicated to data transfers between the processor's various memory spaces, including external DDR and asynchronous memory. Multiple on-chip buses running at up to 133 MHz provide enough bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

The ADSP-BF54x processor includes an on-chip voltage regulator in support of the ADSP-BF54x processor dynamic power management capability. The voltage regulator provides a range of core voltage levels when supplied from a single 2.70 V to 3.6 V input. The voltage regulator can be bypassed at the user's discretion.

BLACKFIN PROCESSOR CORE

As shown in Figure 2 on Page 5, the Blackfin processor core contains two 16-bit multipliers, two 40-bit accumulators, two 40-bit ALUs, four video ALUs, and a 40-bit shifter. The computation units process 8-bit, 16-bit, or 32-bit data from the register file.

The compute register file contains eight 32-bit registers. When performing compute operations on 16-bit operand data, the register file operates as 16 independent 16-bit registers. All operands for compute operations come from the multiported register file and instruction constant fields.

Each MAC can perform a 16-bit by 16-bit multiply in each cycle, accumulating the results into the 40-bit accumulators. Signed and unsigned formats, rounding, and saturation are supported.

The ALUs perform a traditional set of arithmetic and logical operations on 16-bit or 32-bit data. In addition, many special instructions are included to accelerate various signal processing tasks. These include bit operations such as field extract and population count, modulo 2^{32} multiply, divide primitives, saturation and rounding, and sign/exponent detection. The set of video instructions include byte alignment and packing operations, 16-bit and 8-bit adds with clipping, 8-bit average operations, and 8-bit subtract/absolute value/accumulate (SAA) operations. Also provided are the compare/select and vector search instructions.

For certain instructions, two 16-bit ALU operations can be performed simultaneously on register pairs (a 16-bit high half and 16-bit low half of a compute register). By also using the second ALU, quad 16-bit operations are possible.

The 40-bit shifter can perform shifts and rotates and is used to support normalization, field extract, and field deposit instructions.

The program sequencer controls the flow of instruction execution, including instruction alignment and decoding. For program flow control, the sequencer supports PC relative and indirect conditional jumps (with static branch prediction), and subroutine calls. Hardware is provided to support zero-overhead looping. The architecture is fully interlocked, meaning that the programmer need not manage the pipeline when executing instructions with data dependencies.

The address arithmetic unit provides two addresses for simultaneous dual fetches from memory. It contains a multiported register file consisting of four sets of 32-bit index, modify, length, and base registers (for circular buffering), and eight additional 32-bit pointer registers (for C-style indexed stack manipulation).

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. 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.

In addition, multiple L1 memory blocks are provided, offering a configurable mix of SRAM and cache. The memory management unit (MMU) provides memory protection for individual tasks that may be operating on the core and can 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 processor instruction set has been optimized so that 16-bit opcodes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit opcodes, representing fully featured multifunction instructions. Blackfin processors support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

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

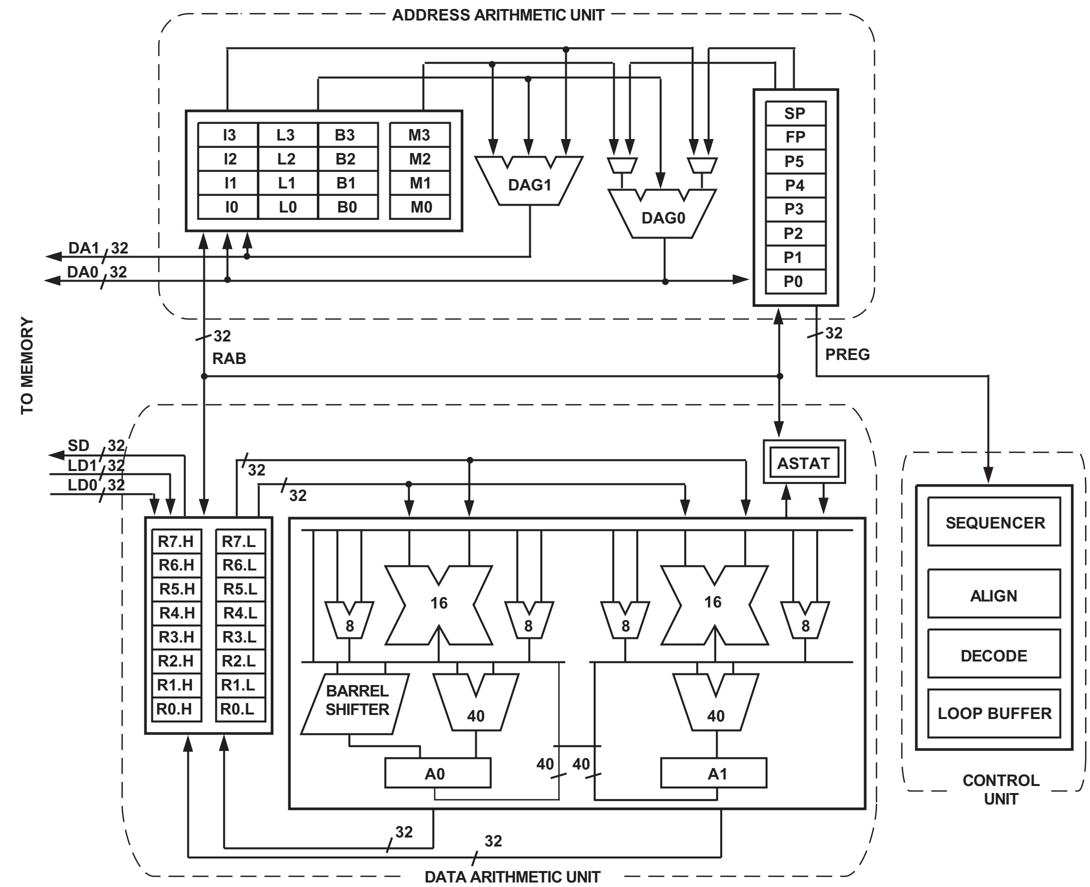


Figure 2. Blackfin Processor Core

MEMORY ARCHITECTURE

The ADSP-BF54x processor 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 on-chip memory as cache or SRAM, and larger, lower-cost and performance off-chip memory systems. See Figure 3 on Page 6.

The on-chip L1 memory system is the highest-performance memory available to the Blackfin processor. The off-chip memory system, accessed through the external bus interface unit (EBIU), provides expansion with flash memory, SRAM, and double-rate SDRAM (DDR1), optionally accessing up to 516M bytes of physical memory.

The ADSP-BF54x processor also includes an 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.

The memory DMA controllers (DMAC1 and DMAC0) provides high-bandwidth data-movement capability. They can perform block transfers of code or data between the internal memory and the external memory spaces.

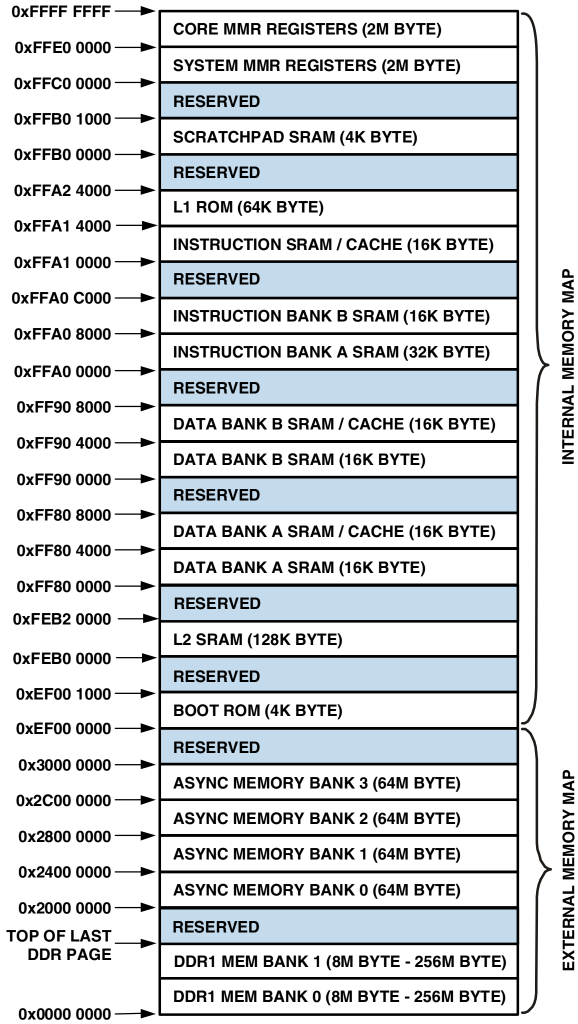


Figure 3. ADSP-BF549 Internal/External Memory Map[1]

[1] This memory map applies to all ADSP-BF54x processors, except for L2 memory population. For details, see Table 1.

Internal (On-Chip) Memory

The ADSP-BF54x processor has several blocks of on-chip memory providing high-bandwidth access to the core.

The first block is the L1 instruction memory, consisting of 48K bytes SRAM, and also 16K bytes that can be configured as a four-way set-associative cache or SRAM. This memory is accessed at full processor speed.

The second on-chip memory block is the L1 data memory, consisting of 64K bytes SRAM, of which 32K bytes can be configured as a two-way set associative cache. This memory block is accessed at full processor speed.

The third memory block is a 4K byte scratchpad SRAM which runs at the same speed as the L1 memories, but is only accessible as data SRAM and cannot be configured as cache memory.

The fourth memory block is the factory programmed L1 instruction ROM, operating at full processor speed. This ROM is not customer configurable.

The fifth memory block is the L2 SRAM, providing 128K bytes of unified Instruction and data memory, operating at one half the frequency of the core.

Finally, there is a 4K boot ROM that can be seen as L3 memory. It operates at full SCLK rate.

External (Off-Chip) Memory

Through the External Bus Interface Unit (EBIU) the ADSP-BF54x processors provide glueless connectivity to external 16-bit wide memories, such as DDR SDRAM, Mobile DDR, SRAM, NOR flash, NAND flash, and FIFO devices. To provide the best performance, the bus system of the DDR interface is completely separate from the other parallel interfaces.

The DDR memory controller can gluelessly manage up to two banks of double-rate synchronous dynamic memory (DDR1 SDRAM). The 16-bit wide interface operates at SCLK frequency enabling maximum throughput of 532 Mbyte/s. The DDR or Mobile DDR controller is augmented with a queuing mechanism that performs efficient bursts onto the DDR. The controller is an industry standard DDR SDRAM controller with each bank supporting from 64 Mbit to 512 Mbit device sizes and 4-, 8-, or 16-bit widths. The controller supports up to 512 Mbytes in one bank, but the total in two banks is limited to 512 Mbytes. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement.

Traditional 16-bit asynchronous memories, such as SRAM, EPROM, and flash devices, can be connected to one of the four 64 MByte asynchronous memory banks, represented by four memory select strobes. Alternatively, these strobes can function as bank-specific read or write strobes preventing further glue logic when connecting to asynchronous FIFO devices.

In addition, the external bus can connect to advanced flash device technologies, such as:

- Page-mode NOR flash devices
- Synchronous burst-mode NOR flash devices
- NAND flash devices

NAND Flash Controller (NFC)

The ADSP-BF54x provides a NAND Flash Controller (NFC) as part of the external bus interface. NAND flash devices provide high-density, low-cost memory. However, NAND flash devices also have long random access times, invalid blocks, and lower reliability over device lifetimes. Because of this, NAND flash is often used for read-only code storage. In this case, all DSP code can be stored in NAND flash and then transferred to a faster memory (such as DDR or SRAM) before execution. Another common use of NAND flash is for storage of multimedia files or other large data segments. In this case, a software file system may be used to manage reading and writing of the NAND flash device. The file system selects memory segments for storage with the goal of avoiding bad blocks and equally distributing memory accesses across all address locations. Hardware features of the NFC include:

- Support for page program, page read, and block erase of NAND flash devices, with accesses aligned to page boundaries.
- Error checking and correction (ECC) hardware that facilitates error detection and correction.
- A single 8-bit or 16-bit external bus interface for commands, addresses and data.
- Support for SLC (single level cell) NAND flash devices unlimited in size, with page sizes of 256 and 512 bytes. Larger page sizes can be supported in software.
- Capability of releasing external bus interface pins during long accesses.
- Support for internal bus requests of 16 or 32-bits.
- DMA engine to transfer data between internal memory and NAND flash device.

I/O Memory Space

The ADSP-BF54x processors do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses near the top of the 4G byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The MMRs are accessible only in supervisor mode and appear as reserved space to on-chip peripherals.

Booting

The ADSP-BF54x processor contains a small on-chip boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF54x processor is configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM. For more information, see Booting Modes on Page 18.

Event Handling

The event controller on the ADSP-BF54x processor handles all asynchronous and synchronous events to the processor. The ADSP-BF54x processor provides event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing of a higher-priority event takes precedence over servicing of a lower-priority event. The controller provides support for five different types of events:

- Emulation. An emulation event causes the processor to enter emulation mode, allowing command and control of the processor via the JTAG interface.
- Reset. This event resets the processor.
- Non-Maskable Interrupt (NMI). The NMI event can be generated by the software watchdog timer or by the NMI input signal to the processor. The NMI event is frequently used as a power-down indicator to initiate an orderly shutdown of the system.
- Exceptions. Events that occur synchronously to program flow (that is, the exception is taken before the instruction is allowed to complete). Conditions such as data alignment violations and undefined instructions cause exceptions.
- Interrupts. Events that occur asynchronously to program flow. They are caused by input pins, timers, and other peripherals, as well as by an explicit software instruction.

Each event type 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-BF54x processor event controller consists of two stages, the core event controller (CEC) and the system interrupt controller (SIC). The core event controller works with the system interrupt controller to prioritize and control all system events. Conceptually, interrupts from the peripherals enter into the SIC, and are then routed directly into the general-purpose interrupts of the CEC.

Core Event Controller (CEC)

The CEC supports nine general-purpose interrupts (IVG15–7), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest-priority interrupts (IVG15–14) are recommended to be reserved for software interrupt handlers, leaving seven prioritized interrupt inputs to support the peripherals of the ADSP-BF54x processor. Table 2 describes the inputs to the CEC, identifies their names in the event vector table (EVT), and lists their priorities.

Table 2. Core Event Controller (CEC)

Priority (0 is Highest)	Event Class	EVT Entry
0	Emulation/Test Control	EMU
1	Reset	RST
2	Non-Maskable Interrupt	NMI
3	Exception	EVX
4	Reserved	—
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-BF54x processor provides a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the interrupt assignment registers (IAR). Table 3 describes the inputs into the SIC and the default mappings into the CEC.

Table 3. System Interrupt Controller (SIC)

Peripheral IRQ (IRQ) Source	IRQ ID	GP IRQ (at Reset)	Core IRQ ID
PLL Wakeup IRQ	0	IVG7	0
DMAC0 Error (generic)	1	IVG7	0
PPI0 Error IRQ	2	IVG7	0
SPORT0 Error IRQ	3	IVG7	0
SPORT1 Error IRQ	4	IVG7	0
SPI0 Status IRQ	5	IVG7	0
UART0 Status IRQ	6	IVG7	0
Real-Time Clock IRQ	7	IVG8	1
DMA12 IRQ (PPI0)	8	IVG8	1
DMA0 IRQ (SPORT0 RX)	9	IVG9	2
DMA1 IRQ (SPORT0 TX)	10	IVG9	2
DMA2 IRQ (SPORT1 RX)	11	IVG9	2
DMA3 IRQ (SPORT1 TX)	12	IVG9	2
DMA4 IRQ (SPI0)	13	IVG10	3
DMA6 IRQ (UART0 RX)	14	IVG10	3
DMA7 IRQ (UART0 TX)	15	IVG10	3
Timer 8 IRQ	16	IVG11	4
Timer 9 IRQ	17	IVG11	4
Timer 10 IRQ	18	IVG11	4
Pin IRQ 0 (PINT0)	19	IVG12	5
Pin IRQ 1 (PINT1)	20	IVG12	5
MDMA Stream 0 IRQ	21	IVG13	6
MDMA Stream 1 IRQ	22	IVG13	6
Software Watchdog Timer IRQ	23	IVG13	6
DMAC1 Error (generic)	24	IVG7	0
SPORT2 Error IRQ	25	IVG7	0
SPORT3 Error IRQ	26	IVG7	0
MXVR Synchronous Data IRQ	27	IVG7	0
SPI1 Status IRQ	28	IVG7	0
SPI2 Status IRQ	29	IVG7	0
UART1 Status IRQ	30	IVG7	0
UART2 Status IRQ	31	IVG7	0

Table 3. System Interrupt Controller (SIC) (Continued)

Peripheral IRQ (IRQ) Source	IRQ ID	GP IRQ (at Reset)	Core IRQ ID
CAN0 Status IRQ	32	IVG7	0
DMA18 IRQ (SPORT2 RX)	33	IVG9	2
DMA19 IRQ (SPORT2 TX)	34	IVG9	2
DMA20 IRQ (SPORT3 RX)	35	IVG9	2
DMA21 IRQ (SPORT3 TX)	36	IVG9	2
DMA13 IRQ (PPI1)	37	IVG9	2
DMA14 IRQ (PPI2, HOST)	38	IVG9	2
DMA5 IRQ (SPI1)	39	IVG10	3
DMA23 IRQ (SPI2)	40	IVG10	3
DMA8 IRQ (UART1 RX)	41	IVG10	3
DMA9 IRQ (UART1 TX)	42	IVG10	3
DMA10 IRQ (ATAPI RX)	43	IVG10	3
DMA11 IRQ (ATAPI TX)	44	IVG10	3
TWI0 IRQ	45	IVG11	4
TWI1 IRQ	46	IVG11	4
CAN0 Receive IRQ	47	IVG11	4
CAN0 Transmit IRQ	48	IVG11	4
MDMA Stream 2 IRQ	49	IVG13	6
MDMA Stream 3 IRQ	50	IVG13	6
MXVR Status IRQ	51	IVG11	4
MXVR Control Message IRQ	52	IVG11	4
MXVR Asynchronous Packet IRQ	53	IVG11	4
PPI1 Error IRQ	54	IVG7	0
PPI2 Error IRQ	55	IVG7	0
UART3 Status IRQ	56	IVG7	0
HOST Error IRQ	57	IVG7	0
Reserved	58	IVG7	0
Pixel Compositor (PIXC) Error IRQ	59	IVG7	0
NFC Error IRQ	60	IVG7	0
ATAPI Error IRQ	61	IVG7	0
CAN1 Status IRQ	62	IVG7	0
DMAR0 Block IRQ	63	IVG7	0
DMAR1 Block IRQ	63	IVG7	0
DMAR0 Overflow Error IRQ	63	IVG7	0
DMAR1 Overflow Error IRQ	63	IVG7	0
DMA15 IRQ (PIXC0)	64	IVG8	1
DMA16 IRQ (PIXC1)	65	IVG8	1
DMA17 IRQ (PIXC2)	66	IVG8	1
DMA22 IRQ (SDH/NFC)	67	IVG8	1

Table 3. System Interrupt Controller (SIC) (Continued)

Peripheral IRQ (IRQ) Source	IRQ ID	GP IRQ (at Reset)	Core IRQ ID
Counter (CNT) IRQ	68	IVG8	1
Keypad (KEY) IRQ	69	IVG8	1
CAN1 RX IRQ	70	IVG11	4
CAN1 TX IRQ	71	IVG11	4
SDH Mask 0 IRQ	72	IVG11	4
SDH Mask 1 IRQ	73	IVG11	4
Reserved	74	IVG11	4
USB_INT0 IRQ	75	IVG11	4
USB_INT1 IRQ	76	IVG11	4
USB_INT2 IRQ	77	IVG11	4
USB_DMAINT IRQ	78	IVG11	4
OTPSEC IRQ	79	IVG11	4
Reserved	80	IVG11	4
Reserved	81	IVG11	4
Reserved	82	IVG11	4
Reserved	83	IVG11	4
Reserved	84	IVG11	4
Reserved	85	IVG11	4
Timer 0 IRQ	86	IVG11	4
Timer 1 IRQ	87	IVG11	4
Timer 2 IRQ	88	IVG11	4
Timer 3 IRQ	89	IVG11	4
Timer 4 IRQ	90	IVG11	4
Timer 5 IRQ	91	IVG11	4
Timer 6 IRQ	92	IVG11	4
Timer 7 IRQ	93	IVG11	4
Pin IRQ 2 (PINT2)	94	IVG12	5
Pin IRQ 3 (PINT3)	95	IVG12	5

Event Control

The ADSP-BF54x processor provides the user with a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each register is 16 bits wide:

- 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 it 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 is processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, preventing the processor from servicing the event even though the event may be latched in the ILAT register. This register may be read or written while in supervisor mode. (Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.)
- CEC interrupt pending register (IPEND). The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing three 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in Table 3 on Page 8.

- SIC interrupt mask register (SIC_IMASK). This register controls the masking and unmasking of each peripheral interrupt event. When a bit is set in the register, that peripheral event is unmasked and is processed by the system when asserted. A cleared bit in the register masks the peripheral event, preventing the processor from servicing the event.
- SIC interrupt status register (SIC_ISR). 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, and a cleared bit indicates the peripheral is not asserting the event.
- SIC interrupt wakeup enable register (SIC_IWR). By enabling the corresponding bit in this register, a peripheral can be configured to wake up the processor, should the core be idled when the event is generated. (For more information, see Dynamic Power Management on Page 15.)

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 recognizes and queues 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 state of the processor.

DMA CONTROLLERS

ADSP-BF54x processors have multiple, independent DMA channels that support automated data transfers with minimal overhead for the processor core. DMA transfers can occur between the ADSP-BF54x processor's internal memories and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including DDR and asynchronous memory controllers.

While the USB controller and MXVR have their own dedicated DMA controllers, the other on-chip peripherals are managed by two centralized DMA controllers, called DMAC1 (32-bit) and DMAC0 (16-bit). Both operate in the SCLK domain. Each DMA controller manages twelve independent DMA channels. The DMAC1 controller masters high-bandwidth peripherals over a dedicated 32-bit DMA access bus (DAB32). Similarly, the DMAC0 controller masters most of serial interfaces over the 16-bit DAB16 bus. Individual DMA channels have fixed access priority on the DAB buses. DMA priority of peripherals is managed by flexible peripheral-to-DMA channel assignment.

All four DMA controllers use the same 32-bit DCB bus to exchange data with L1 memory. This includes L1 ROM, but excludes scratchpad memory. Fine granulation of L1 memory and special DMA buffers minimize potential memory conflicts, if the L1 memory is accessed by the core contemporaneously. Similarly, there are dedicated DMA buses between the DMAC1, DMAC0, and USB DMA controllers and the external bus interface unit (EBIU) that arbitrates DMA accesses to external memories and boot ROM.

The ADSP-BF54x processor 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-BF54x processor DMA controller include:

- A single, linear buffer that stops upon completion
- A circular, auto-refreshing 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, both the DMAC1 and the DMAC0 controllers feature two memory DMA channel pairs for transfers between the various memories of the ADSP-BF54x processor system. This enables transfers of blocks of data between any of the memories—including external DDR, ROM, SRAM, and flash memory—with minimal processor intervention. Like peripheral DMAs, memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

The memory DMA channels of the DMAC1 controller (MDMA2 and MDMA3) can be optionally controlled by the external DMA request input pins. When used in conjunction with the External Bus Interface Unit (EBIU) this so-called Handshaked Memory DMA (HMDMA) scheme can be used to efficiently exchange data with block-buffered or FIFO-style devices connected externally. Users can select whether the DMA request pins control the source or the destination side of the memory DMA. It allows control of the number of data transfers for memory DMA. The number of transfers per edge is programmable. This feature can be programmed to allow memory DMA to have an increased priority on the external bus relative to the core.

Host DMA Interface

The host DMA port (HOST) facilitates a host device external to the ADSP-BF54x to be a DMA master and transfer data back and forth. The host device always masters the transactions and the processor is always a DMA slave device.

The HOST port is enabled through the peripheral access bus. Once enabled, the DMA is controlled by the external host. The external host can then program the DMA to send/receive data to any valid internal and external memory location. The HOST Port controller includes the following features:

- Allows an external master to configure DMA read/write data transfers and read port status
- Uses an asynchronous memory protocol for its external interface
- Allows 8 or 16-bit external data interface to the host device
- Supports half-duplex operation
- Supports Little/Big Endian data transfers
- Acknowledge mode allows flow control on host transactions
- Interrupt mode guarantees a burst of FIFO depth host transactions

REAL-TIME CLOCK

The ADSP-BF54x processor Real-Time Clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 KHz crystal external to the ADSP-BF54x processors. The RTC peripheral has dedicated power supply pins so that it can remain powered up and clocked even when the rest of the processor is in a low-power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, hour, or day clock ticks, interrupt on programmable stopwatch countdown, or interrupt at a programmed alarm time.

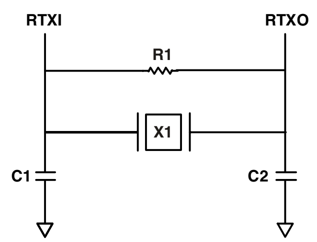
The 32.768 KHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60-second counter, a 60-minute counter, a 24-hour counter, and an 32,768-day counter.

When enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register. There are two alarms: The first alarm is for a time of day. The second alarm is for a day and time of that day.

The stopwatch function counts down from a programmed value, with one-second resolution. When the stopwatch is enabled and the counter underflows, an interrupt is generated.

Like the other peripherals, the RTC can wake up the ADSP-BF54x processor from sleep mode upon generation of any RTC wakeup event. Additionally, an RTC wakeup event can wake up the ADSP-BF54x processor from deep sleep mode, and wake up the on-chip internal voltage regulator from the hibernate operating mode.

Connect RTC pins RTXI and RTXO with external components as shown in Figure 4.



SUGGESTED COMPONENTS:
ECLIPTEK EC38J (THROUGH-HOLE PACKAGE)
EPSON MC405 12 PF LOAD (SURFACE MOUNT PACKAGE)
C1 = 22 PF
C2 = 22 PF
R1 = 10 MΩ
NOTE: C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1. CONTACT CRYSTAL MANUFACTURER FOR DETAILS. C1 AND C2 SPECIFICATIONS ASSUME BOARD TRACE CAPACITANCE OF 3 PF.

Figure 4. External Components for RTC

WATCHDOG TIMER

The ADSP-BF54x processor includes a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state through 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.

If configured to generate a hardware reset, the watchdog timer resets both the core and the ADSP-BF54x processor peripherals. After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the watchdog timer control register.

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

TIMERS

There are up to two timer units in the ADSP-BF54x processors. While one unit provides eight general-purpose programmable timers, the other unit provide three of them. processors. Each timer has an external pin that can be configured either as a Pulse Width Modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input to the several other associated PF pins, an external clock input to the PPI_CLK input pin, or to the internal SCLK.

The timer units can be used in conjunction with the two UARTs and the CAN controller to measure the width of the pulses in the data stream to provide a software auto-baud detect function for the respective serial channels.

The timers can generate interrupts to the processor core providing periodic events for synchronization, either to the system clock or to a count of external signals.

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

UP/DOWN COUNTER AND THUMBWHEEL INTERFACE

A 32-bit up/down counter is provided that can sense 2-bit quadrature or binary codes as typically emitted by industrial drives or manual thumb wheels. The counter can also operate in general-purpose up/down count modes. Then, count direction is either controlled by a level-sensitive input pin or by two edge detectors.

A third input can provide flexible zero marker support and can alternatively be used to input the push-button signal of thumb wheels. All three pins have a programmable debouncing circuit.

An internal signal forwarded to the timer unit enables one timer to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmable count values are exceeded.

SERIAL PORTS (SPORTS)

The ADSP-BF54x processor incorporates four dual-channel synchronous serial ports (SPORT0, SPORT1, SPORT2, SPORT3) for serial and multiprocessor communications. The SPORTs support the following features:

- I^2S capable operation.
- Bidirectional operation. Each SPORT has two sets of independent transmit and receive pins, enabling eight channels of I^2S stereo audio.
- Buffered (8-deep) transmit and receive ports. Each port has a data register for transferring data words to and from other processor 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 processor 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 1024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

SERIAL PERIPHERAL INTERFACE (SPI) PORTS

The ADSP-BF54x processor has three SPI-compatible ports that allow the processor to communicate with multiple SPI-compatible devices.

Each SPI port uses three pins for transferring data: two data pins (master output-slave input, MOSI, and master input-slave output, MISO) and a clock pin (serial clock, SCK). An SPI chip select input pin ($\overline{\text{SPISS}}$) lets other SPI devices select the processor, and seven SPI chip select output pins (SPISEL7–1) let the processor select other SPI devices. The SPI select pins are reconfigured programmable flag pins. Using these pins, the SPI ports provide a full-duplex, synchronous serial interface, which supports both master/slave modes and multimaster environments.

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

The SPI port's clock rate is calculated as:

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

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

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

UART PORTS (UARTS)

The ADSP-BF54x processor provides four full-duplex Universal Asynchronous Receiver/Transmitter (UART) ports. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for 5 to 8 data bits, 1 or 2 stop bits, and none, even, or odd parity. Each UART port supports two modes of operation:

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

Each UART port's baud rate, serial data format, error code generation and status, and interrupts are programmable:

- Supporting bit rates ranging from (f_{SCLK}/ 1,048,576) to (f_{SCLK}) bits per second.
- Supporting data formats from 7 to12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as:

$$UART\ Clock\ Rate = \frac{f_{SCLK}}{16^{(1 - EBIO)} \times UART\_Divisor}$$

Where the 16-bit UART Divisor comes from the UARTx_DLH register (most significant 8 bits) and UARTx_DLL register (least significant 8 bits).

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

UART1 and UART3 feature a pair of RTS (request to send) and CTS (clear to send) signals for hardware flow purposes. The transmitter hardware is automatically prevented from sending further data when the CTS input is de-asserted. The receiver can automatically de-assert its RTS output when the enhanced receive FIFO exceeds a certain high-water level. The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA®) Serial Infrared Physical Layer Link Specification (SIR) protocol.

CONTROLLER AREA NETWORK (CAN)

The ADSP-BF54x processor offers two CAN controllers that are communication controllers that implement the Controller Area Network (CAN) 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well suited for control applications due to its capability to communicate reliably over a network since the protocol incorporates CRC checking message error tracking, and fault node confinement.

The ADSP-BF54x CAN controllers offer the following features:

- 32 mailboxes (8 receive only, 8 transmit only, 16 configurable for receive or transmit).
- Dedicated acceptance masks for each mailbox.
- Additional data filtering on first two bytes.
- Support for both the standard (11-bit) and extended (29-bit) identifier (ID) message formats.
- Support for remote frames.
- Active or passive network support.
- CAN wakeup from hibernation mode (lowest static power consumption mode).
- Interrupts, including: TX complete, RX complete, error, global.

The electrical characteristics of each network connection are very demanding so the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The ADSP-BF54x CAN module represents only the controller part of the interface. The controller interface supports connection to 3.3V high-speed, fault-tolerant, single-wire transceivers.

TWI CONTROLLER INTERFACE

The ADSP-BF54x processor includes two Two Wire Interface (TWI) modules for providing a simple exchange method of control data between multiple devices. The modules are compatible with the widely used I^{2}C bus standard. The TWI modules offer the capabilities of simultaneous Master and Slave operation, support for both 7-bit addressing and multimedia data arbitration. Each TWI interface uses two pins for transferring clock (SCL) and data (SDA) and supports the protocol at speeds up to 400k bits/sec. The TWI interface pins are compatible with 5 V logic levels.

Additionally, the ADSP-BF54x processor's TWI modules are fully compatible with Serial Camera Control Bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

PORTS

Because of their rich set of peripherals, the ADSP-BF54x processors group the many peripheral signals to ten ports—referred to as Port A to Port J. Most ports contain 16 pins, a few have less. Many of the associated pins are shared by multiple signals. The ports function as multiplexer controls. Every port has its own set of memory-mapped registers to control port muxing and GPIO functionality.

General-Purpose I/O (GPIO)

Every pin in Port A to Port J can function as a GPIO pin resulting in a GPIO pin count of 154. While it is unlikely that all GPIOs will be used in an application as all pins have multiple functions, the richness of GPIO functionality guarantees unrestrictive pin usage. Every pin that is not used by any function can be configured in GPIO mode on an individual basis.

After reset, all pins are in GPIO mode by default. Neither GPIO output nor input drivers are active by default. Unused pins can be left unconnected, therefore. GPIO data and direction control registers provide flexible write-one-to-set and write-one-to-clear mechanisms so that independent software threads do not need to protect against each other because of expensive read-modify-write operations when accessing the same port.

Pin Interrupts

Due to the huge number of port pins, the ADSP-BF54x processors introduce a new scheme to manage pin interrupts. Every port pin can request interrupts in either an edge-sensitive or a level-sensitive manner with programmable polarity. Interrupt functionality is decoupled from GPIO operation. Four system-level interrupt channels (INT0, INT1, INT2 and INT3) are reserved for this purpose. Each of these interrupt channels can manage up to 32 interrupt pins. The assignment from pin to interrupt is not performed at a pin by pin level. Rather, groups of eight pins (half ports) can be flexibly assigned to interrupt channels.

Every pin interrupt channel features a special set of 32-bit memory-mapped registers, that enable half port assignment and interrupt management. This not only includes masking, identification, and clearing of requests, it also enables access to the respective pin states and use of the interrupt latches regardless of whether the interrupt is masked or not. Most control registers feature multiple MMR address entries to write-one-to-set or write-one-to-clear them individually.

PIXEL COMPOSITOR (PIXC)

The pixel compositor (PIXC) provides image overlay with transparent-color support, alpha blending, and color space conversion capability for output to TFT-LCDs as well as NTSC/PAL video encoders. It provides all of the control to allow two data streams from two separate data buffers to be combined, blended, and converted into appropriate forms for both LCD panels and digital video outputs. The main image buffer provides the basic background image, which is presented in the data stream. The overlay image buffer allows the user to add multiple foreground text, graphics, or video on top of the main image or video data stream.

ENHANCED PARALLEL PERIPHERAL INTERFACE (PPI)

The ADSP-BF54x processor provides multiple Enhanced Parallel Peripheral Interfaces (PPIs), one 16 bits wide and one 18 bits wide. The PPI supports the direct connection to active TFT LCD, parallel A/D and D/A converters, video encoders and decoders, image sensor module and other general purpose peripherals.

The following features are supported in the PPI module.

- Programmable data length: 8, 10, 12, 14, 16, 18, 24 and 32 bits per clock.
- Bi-directional and half-duplex port.
- PPI_CLK can be provided externally or can be generated internally.
- Various framed and non-framed operating modes. Frame syncs can be generated internally or can be supplied by an external device.
- Various general purpose modes with one frame syncs, two frame syncs, three frame syncs and zero frame sync modes for both receive and transmit.
- ITU-656 status word error detection and correction for ITU-656 Receive modes.
- ITU-656 preamble and status word decode.
- Three different modes for ITU-656 receive modes: active video only, vertical blanking only, and entire field mode.
- Horizontal and vertical windowing for GP 2 and 3 FS Modes.
- Optional packing and unpacking of data to/from 32 bits from/to 8, 16 and 24 bits. If packing/unpacking is enabled, endianness can be changed to change the order of packing/unpacking of bytes/words.
- Optional sign extension or zero fill for receive modes.
- During receive modes, alternate even or odd data sample can be filtered out.
- Programmable clipping of data values for 8-bit transmit modes.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various de-interleaving/interleaving modes for receiving/transmitting 4:2:2 YCrCb data.
- FIFO watermarks and urgent DMA features.
- Clock gating by an external device asserting the clock gating control signal.

USB ON-THE-GO DUAL-ROLE DEVICE CONTROLLER

The USB OTG controller provides a low-cost connectivity solution for consumer mobile devices such as cell phones, digital still cameras and MP3 players, allowing these devices to transfer data using a point-to-point USB connection without the need for a PC host. The USBDRC module can operate in a traditional USB peripheral-only mode as well as the host mode presented in the On-The-Go (OTG) supplement [1] to the USB 2.0 Specification [2]. In host mode, the USB module supports transfers at high-speed (480Mbps), full-speed (12Mbps), and low-speed (1.5Mbps) rates. Peripheral-only mode supports the high- and full-speed transfer rates.

ATA/ATAPI–6 INTERFACE

The ATAPI interface connects to CD/DVD and HDD drives, and is ATAPI-6 compliant. The controller implements the peripheral I/O mode, the multi-DMA mode, and the Ultra DMA mode. The DMA modes enable faster data transfer and reduced host management. The ATAPI Controller supports PIO, Multi-DMA, and Ultra DMA ATAPI accesses. Key features include:

- Supports PIO modes 0,1,2,3,4
- Supports Multiword DMA modes 0,1,2
- Supports Ultra DMA modes 0,1,2,3,4,5 (up to UDMA 100)
- Programmable timing for ATA interface unit
- Supports CompactFlash Card using True IDE mode

KEYPAD INTERFACE

The keypad interface is a 16 pin interface module that is used to detect the key pressed in a 8x8 (maximum) keypad matrix. The size of the input keypad matrix is programmable. The interface is capable of filtering the bounce on the input pins, which is common in keypad applications. The width of the filtered bounce is programmable. The Interface module is capable of generating an interrupt request to the core once it identifies that any key has been pressed.

The interface supports a press-release-press mode and infrastructure for a press-hold mode. The former mode identifies a press, release and press of a key as two consecutive presses of the same key where as the later mode checks the input key's state in periodic intervals to determine the number of times the same key is meant to be pressed. Simultaneous multiple keys pressed detection possible and limited key resolution capability. Key features include:

- Supports a maximum of 8x8 keypad matrix
- Programmable input keypad matrix size
- Debounce filter on input signals
- Programmable debounce filter width
- Press-Release-Press mode supported
- Infrastructure for Press-Hold mode present
- Interrupt on any key pressed capability
- Multiple key pressed detection and limited multiple key resolution capability

SECURE DIGITAL (SD)/SDIO CONTROLLER

The SD/SDIO controller is a serial interface that stores data at a data rate of up to 10M bytes per second using a 4-bit data line. The interface runs at 25 MHz.

The SD/SDIO controller supports the SD memory mode only. The interface supports all the power modes and performs error checking by CRC.

CODE SECURITY

An OTP/security system consisting of a blend of hardware and software provides customers with a flexible and rich set of code security features with Lockbox™[1] secure technology. Key features include:

- OTP memory
- Unique chip ID
- Code authentication
- Secure mode of operation

MEDIA TRANSCEIVER MAC LAYER (MXVR)

The ADSP-BF54x processor provides a Media Transceiver (MXVR) MAC layer, allowing the processor to be connected directly to a MOST®[2] network through just an FOT or Electrical PHY.

The MXVR is fully compatible with the industry standard standalone MOST controller devices, supporting 22.579 Mbps or 24.576 Mbps data transfer. It offers faster lock times, greater jitter immunity, a sophisticated DMA scheme for data transfers, and the high-speed internal interface to the core and L1 memory allows the full bandwidth of the network to be utilized. The MXVR can operate as either the network master or as a network slave.

The MXVR supports synchronous data, asynchronous packets, and control messages using dedicated DMA channels which operate autonomously from the processor core moving data to and from L1 memory. Synchronous data is transferred to or from the synchronous data physical channels on the MOST bus through eight programmable DMA channels. The synchronous data DMA channels can operate in various modes including modes which trigger DMA operation when data patterns are detected in the receive data stream. Furthermore two DMA channels support asynchronous traffic and a further two support control message traffic.

Interrupts are generated when a user defined amount of synchronous data has been sent or received by the processor or when asynchronous packets or control messages have been sent or received.

The MXVR peripheral can wake up the ADSP-BF54x processor from sleep mode when a wakeup preamble is received over the network or based on any other MXVR interrupt event. Additionally, detection of network activity by the MXVR can be used to wake up the ADSP-BF54x processor from sleep mode or the hibernate state, and wake up the on chip internal voltage regulator from a powered-down state. These features allow the ADSP-BF54x to operate in a low-power state when there is no network activity or when data is not currently being received or transmitted by the MXVR.

The MXVR clock is provided through a dedicated external crystal or crystal oscillator. The frequency of external crystal or crystal oscillator can be 256Fs, 384Fs, 512Fs, or 1024Fs for Fs = 38kHz, 44.1kHz, or 48kHz. If using a crystal to provide the MXVR clock, use a parallel-resonant, fundamental mode, microprocessor-grade crystal.

DYNAMIC POWER MANAGEMENT

The ADSP-BF54x processor provides five operating modes, 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-BF54x processor peripherals also reduces power consumption. See Table 4 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 power-up default execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

Active Operating Mode – Moderate Power Savings

In the active mode, the PLL is enabled but bypassed. 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 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 4. 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

[1] Lockbox is a trademark of Analog Devices, Inc.
[2] MOST is a registered trademark of Standard Microsystems, Corp.

Sleep Operating Mode – High Dynamic Power Savings

The sleep mode reduces dynamic 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 or RTC activity 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). 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.

When in the sleep mode, system DMA access to L1 memory is not supported.

Deep Sleep Operating Mode – Maximum Dynamic Power Savings

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core (CCLK) and to all synchronous peripherals (SCLK). Asynchronous peripherals, such as the RTC, may still be running but will not be able to access internal resources or external memory. This powered-down mode can only be exited by assertion of the reset interrupt ($\overline{\text{RESET}}$) or by an asynchronous interrupt generated by the RTC. When in deep sleep mode, an RTC asynchronous interrupt causes the processor to transition to the active mode. Assertion of $\overline{\text{RESET}}$ while in deep sleep mode causes the processor to transition to the full on mode.

Hibernate Operating Mode – Maximum Static Power Savings

The hibernate mode 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 0V to provide the greatest power savings mode. 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 tri-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 CAN, by the MXVR, by the keypad, by the up/down counter, and by some GPIO pins. It can also be woken up by a real-time clock wakeup event or by asserting the $\overline{\text{RESET}}$ pin, both of which initiate the hardware reset sequence.

With the exception of the VR_CTL and the RTC registers, all internal registers and memories lose their content in hibernate state. State variables may be held in external SRAM or SDRAM.

Power Savings

As shown in Table 5, the ADSP-BF54x processor supports 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-BF54x processor into its own power domain, separate from the RTC and other I/O, the processor can take advantage of dynamic power management, without affecting the RTC or other I/O devices. There are no sequencing requirements for the various power domains.

Table 5. Power Domains

Power Domain	VDD Range
All internal logic, except RTC, DDR, and USB	V_{DDINT}
RTC internal logic and crystal I/O	V_{DDRTC}
DDR external memory supply	V_{DDDDR}
USB internal logic and crystal I/O	V_{DDUSB}
MXVR crystal I/O	V_{DDMC}
MXVR I/O	V_{DDMX}
MXVR PLL and logic	V_{DDMP}
All other I/O	V_{DDEXT}

VOLTAGE REGULATION

The ADSP-BF54x processor provides an on-chip voltage regulator that can generate processor core voltage levels from an external supply. (Note specifications as indicated in Operating Conditions on Page 30.) Figure 5 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 supplied. While in hibernate mode, V_{DDEXT} can still be applied, eliminating the need for external buffers. The voltage regulator can be activated from this power down state by assertion of the RESET pin, which will then initiate a boot sequence. The regulator can also be disabled and bypassed at the user's discretion.

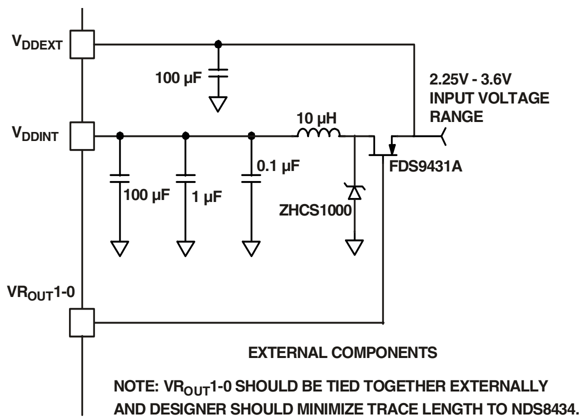


Figure 5. Voltage Regulator Circuit

CLOCK SIGNALS

The ADSP-BF54x processor can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

If an external clock is used, it 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's CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the ADSP-BF54x processor includes an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 6. A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor shown in Figure 6 fine tune phase and amplitude of the sine frequency.

The capacitor and resistor values shown in Figure 6 are typical values only. The capacitor values are dependent upon the crystal manufacturers' load capacitance recommendations and the PCB physical layout. The resistor value depends on the drive level specified by the crystal manufacturer. System designs should verify the customized values based on careful investigations on multiple devices over temperature range.

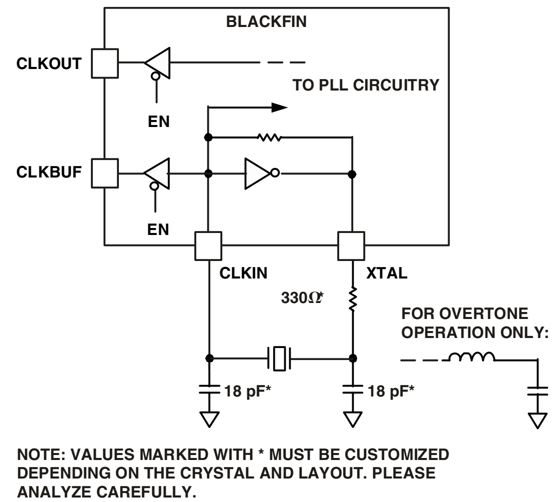


Figure 6. External Crystal Connections

A third-overtone crystal can be used at frequencies above 25 MHz. The circuit is then modified to ensure crystal operation only at the third overtone, by adding a tuned inductor circuit as shown in Figure 6. A design procedure for third-overtone operation is discussed in detail in application note EE-168.

The Blackfin core runs at a different clock rate than the on-chip peripherals. As shown in Figure 7 on Page 18, 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 programmable 1× to 63× multiplication factor (bounded by specified minimum and maximum VCO frequencies). The default multiplier is 10×, but it can be modified by a software instruction sequence. On-the-fly frequency changes can be effected by simply writing to the PLL_DIV register.

On-the-fly CCLK and SCLK frequency changes can be effected by simply writing to the PLL_DIV register. Whereas the maximum allowed CCLK and SCLK rates depend on the applied voltages V_{DDINT} and V_{DDEXT}, the VCO is always permitted to run up to the frequency specified by the part's speed grade. The CLKOUT pin reflects the SCLK frequency to the off-chip world.

It functions as reference for many timing specifications. While inactive by default, it can be enabled using the EBIU_SDGCTL and EBIU_AMGCTL registers.

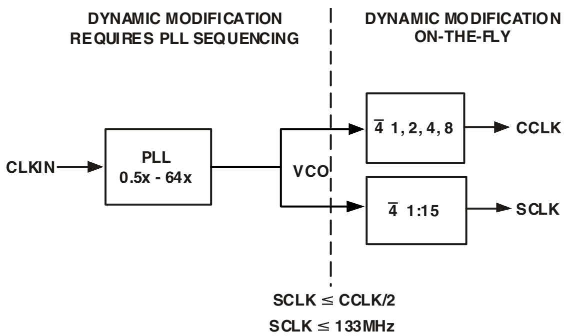


Figure 7. Frequency Modification Methods

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are two through 15. Table 6 illustrates typical system clock ratios. The default ratio is 5.

Table 6. Example System Clock Ratios

Signal Name SSEL3–0	Divider Ratio VCO/SCLK	Example Frequency Ratios (MHz)	
		VCO	SCLK
0010	2:1	200	100
0110	6:1	300	50
1010	10:1	500	50

Note that the divisor ratio must be chosen to limit the system clock frequency to its maximum of f_{SCLK}. The SSEL value can be changed dynamically without any PLL lock latencies by writing the appropriate values to the PLL divisor register (PLL_DIV).

The core clock (CCLK) frequency can also be dynamically changed by means of the CSEL1–0 bits of the PLL_DIV register. Supported CCLK divider ratios are 1, 2, 4, and 8, as shown in Table 7. The default ratio is 1. This programmable core clock capability is useful for fast core frequency modifications.

The maximum CCLK frequency not only depends on the part's speed grade, it also depends on the applied V_{DDINT} voltage. See Table 12 through Table 14 for details.

Table 7. Core Clock Ratios

Signal Name CSEL1–0	Divider Ratio VCO/CCLK	Example Frequency Ratios (MHz)	
		VCO	CCLK
00	1:1	300	300
01	2:1	300	150
10	4:1	500	125
11	8:1	200	25

BOOTING MODES

The ADSP-BF54x processor has many mechanisms (listed in Table 8) for automatically loading internal and external memory after a reset. The boot mode is defined by four BMODE input pins dedicated to this purpose. There are two categories of boot modes: In master boot modes the processor actively loads data from parallel or serial memories. In slave boot modes the processor receives data from an external host devices.

Table 8. Booting Modes

BMODE3–0	Description
0000	Idle–no boot
0001	Boot from 8- or 16-bit external flash memory
0010	Boot from 16-bit asynchronous FIFO
0011	Boot from serial SPI memory (EEPROM or flash)
0100	Boot from SPI host device
0101	Boot from serial TWI memory (EEPROM/flash)
0110	Boot from TWI host
0111	Boot from UART host
1000	Reserved
1001	Reserved
1010	Boot from (DDR) SDRAM
1011	Reserved
1100	Reserved
1101	Reserved
1110	Boot from 16-Bit Host DMA
1111	Boot from 8-Bit Host DMA

The boot modes listed in Table 8 provide a number of mechanisms for automatically loading the processor's internal and external memories after a reset. By default all boot modes use the slowest meaningful configuration settings. Default settings can be altered via the initialization code feature at boot time or by proper OTP programming at pre-boot time.The BMODE pins of the reset configuration register, sampled during power-on resets and software-initiated resets, implement the following modes:

- Idle–no boot mode (BMODE=0x0) — In this mode, the processor goes into idle. The idle boot mode helps to recover from illegal operating modes, in the case the user misconfigured the OTP memory.
- Boot from 8- or 16-bit external flash memory (BMODE=0x1) — In this mode, the boot kernel loads the first block header from address 0x2000 0000 and—depend-

ing on instructions containing in the header—the boot kernel performs 8-bit or 16-bit boot or starts program execution at the address provided by the header. By default, 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 16-bit asynchronous FIFO (BMODE=0x2) — In this mode, the boot kernel starts booting from address 0x2030 0000. Every 16-bit word that boot kernel has to read from the FIFO must be requested by an low pulse on the DMAR1 pin.
- Boot from serial SPI memory, EEPROM or flash (BMODE=0x3) — Eight-, 16-, 24- or 32-bit addressable devices are supported. (internal note: no special support for DataFlashes, as they understand now also standard SPI protocol). The processor uses the PE4 GPIO pin to select a single SPI EEPROM/flash device, submits a read command and successive address bytes (0x00) until a valid 8-, 16-, 24-, or 32-bit addressable device is detected. Pull-up resistors are required on the SSEL and MISO pins. By default, a value of 0x85 is written to the SPI_BAUD register.
- Boot from SPI host device (BMODE=0x4) — The processor operates in SPI slave mode (using SPI0) and is configured to receive the bytes of the .LDR file from an SPI host (master) agent. In the host, the HWAIT signal must be interrogated by the host before every transmitted byte. A pull-up resistor is required on the SPISS input. A pull-down on the serial clock may improve signal quality and booting robustness.
- Boot from serial TWI memory, EEPROM/flash (BMODE=0x5) — The processor operates in master mode (using TWI0) and selects the TWI slave with the unique id 0xA0. The processor submits successive read commands to the memory device starting at two byte internal address 0x0000 and begins clocking data into the processor. The TWI memory device should comply with Philips I2C Bus Specification version 2.1 and have the capability to auto-increment its internal address counter such that the contents of the memory device can be read sequentially. By default, a prescale value of 0xA and CLKDIV value of 0x0811 is used. Unless, altered by OTP settings an I2C memory that takes two address bytes is assumed. Development tools ensure that data that is booted to memories that cannot be accessed by the Blackfin core is written to intermediate storage place and then copied to final destination via Memory DMA.
- Boot from TWI host (BMODE=0x6) — The TWI host agent selects the slave with the unique id 0x5F. The processor (using TWI0) replies with an acknowledgement and the host can then download the boot stream. The TWI host agent should comply with Philips I2C Bus Specification version 2.1. An I2C multiplexer can be used to select one processor at a time when booting multiple processors from a single TWI.
- Boot from UART host (BMODE=0x7) — In this mode, the processor uses UART1 as booting source. Using an autobaud handshake sequence, a boot-stream-formatted program is downloaded by the host. The host agent selects a bit rate within the UART's clocking capabilities.

  When performing the autobaud, the UART expects a "@" (0x40) character (eight bits data, one start bit, one stop bit, no parity bit) on the RXD pin to determine the bit rate. It then replies with an acknowledgement which is composed of 4 bytes: 0xBF, the value of UART_DLL, the value of UART_DLH, 0x00. The host can then download the boot stream. The processor deasserts the RTS output to hold off the host; CTS functionality is not enabled at boot time.
- Boot from (DDR) SDRAM (BMODE=0xA) — In this mode, the boot kernel starts booting from address 0x0000 0010. This is a warm boot scenery only. The SDRAM is expected to contain a valid boot stream and the SDRAM controller must have been configured by the OTP settings.
- Boot from 16-Bit Host DMA (BMODE=0xE) — In this mode, the host DMA port is configured in 16-bit ACK mode, little endian. Unlike in other modes, here the host is responsible for interpreting the boot stream. It writes data block per data block into the Host DMA port. The host can interrogate the HRDY bit in the HOST_STATUS register or the HWAIT signal to know, whether the Blackfin is ready or busy with internal processing. When the host issues a HIRQ command the boot kernel issues a CALL instruction to 0xFFA0 0000 address. It is the host's responsibility that there is valid code at this address at this time. The routine at 0xFFA0 0000 may be a simply initialization routine that configures internal resources, such as the SDRAM controller, that returns by an RTS instruction. The routine may also by the final application that never returns to the boot kernel.
- Boot from 8-Bit Host DMA (BMODE=0xF) — In this mode, the Host DMA port is configured in 8-bit INT mode, little endian. Unlike in other modes, here the host is responsible for interpreting the boot stream. It writes data block per data block into the Host DMA port. The host can interrogate the HRDY bit in the HOST_STATUS register or the HWAIT signal to know, whether the Blackfin is ready or busy with internal processing. When the host issues a HIRQ command the boot kernel issues a CALL instruction to 0xFFA0 0000 address. It is the host's responsibility that there is valid code at this address at this time. The routine at 0xFFA0 0000 may be a simply initialization routine that configures internal resources, such as the SDRAM controller, that returns by an RTS instruction. The routine may also by the final application that never returns to the boot kernel.

For each of the boot modes, a 16-byte header is first read from an external memory device. The header specifies the number of bytes to be transferred and the memory destination address. Multiple memory blocks may be loaded by any boot sequence. Once all blocks are loaded, program execution commences from the address stored in the EVT1 register.

Prior to booting, the pre-boot routine interrogates the OTP memory. Individual boot modes can be customized or even disabled based on OTP programming. External hardware, especially booting hosts may watch the HWAIT signal to determine when the pre-boot has finished and the boot kernel starts the boot process.

The boot kernel differentiates between a regular hardware reset and a wakeup-from-hibernate event to speed up booting in the later case. Bits 6-5 in the system reset configuration (SYSCR) register can be used to bypass pre-boot routine and/or boot kernel in case of a software reset. They can also be used to simulate a wakeup-from-hibernate boot in the software reset case.

The boot process can be further customized by "initialization code." This is a piece of code that is loaded and executed prior to the regular application boot. Typically, this is used to configure the DDR controller or to speed up booting by managing PLL, clock frequencies, wait states, or serial bit rates.

The boot ROM also features C-callable function entries that can be called by the user application at run time. This enables second-stage boot or boot management schemes to be implemented with ease.

INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both user (algorithm/application code) and supervisor (O/S kernel, device drivers, debuggers, ISRs) modes 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/MCU 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 + two load/store + two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4G byte memory space, providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data-types; and separate user and supervisor 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 in 16 bits.

DEVELOPMENT TOOLS

The ADSP-BF54x processor is supported with a complete set of CROSSCORE® software and hardware development tools, including Analog Devices emulators and VisualDSP++® development environment. The same emulator hardware that supports other Blackfin processors also fully emulates the ADSP-BF54x processor.

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 each JTAG processor. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor system is set running at full speed with no impact on system timing.

To use these emulators, the target 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 *Analog Devices JTAG Emulation Technical Reference* (EE-68) on the Analog Devices web site under www.analog.com/ee-notes. This document is updated regularly to keep pace with improvements to emulator support.

RELATED DOCUMENTS

The following publications that describe the ADSP-BF54x processors (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on our Website:

- *Getting Started With Blackfin Processors*
- *ADSP-BF54x Blackfin Processor Hardware Reference*
- *ADSP-BF53x/BF56x Blackfin Processor Programming Reference*
- *ADSP-BF54x Blackfin Processor Anomaly List*

PIN DESCRIPTIONS

ADSP-BF54x processor pin multiplexing scheme is listed in Table 9 and the pin definitions are listed in Table 10.

Table 9. Pin Multiplexing

Primary Pin Function (Number of Pins)	First Peripheral Function	Second Peripheral Function	Third Peripheral Function	Fourth Peripheral Function	Interrupt Capability
Port A					
GPIO (16 pins)	SPORT2 (8 pins)	TMR4 (1 pin)	TACI7 (1 shared pin)		Interrupts (16 pins)
		TMR5 (1 pin)	TACLK7-0 (8 pins)		
	SPORT3 (8 pins)	TMR6 (1 pin)			
		TMR7 (1 pin)			
Port B					
GPIO (15 pins)	TWI1 (2 pins) UART2 or 3 CTL (2 pins) UART2 (2 pins) UART3 (2 pins)		TACI2-3 (2 pins)		Interrupts (15 pins)
	SPI2 SEL (4 pins)	TMR0–2 (3 pins)			
	SPI2 (3 pins)	HWAIT (1 pin)			
Port C					
GPIO (16 pins)	SPORT0 (8 pins)	MXVR MMCLK, MBCLK (2 pins)			Interrupts (8 pins)[1]
	SDH (6 pins)				Interrupts (8 pins)
Port D					
GPIO (16 pins)	PPI1 D0–15 (16 pins)	HOST D0–15 (16 pins)	SPORT1 (8 pins)	PPI0 D18– 23 (6 pins)	Interrupts (8 pins)
			PPI2 D 0–7 (8 pins)	Keypad Row 0–3 Col 0–3 (8 pins)	Interrupts (8 pins)
Port E					
GPIO (16 pins)	SPI0 (7 pins)	KEYPAD Row 4–6 Col 4–7 (7 pins)	TACI0 (1 pin)		Interrupts (8 pins)
	UART0 TX (1 pin)	KEYPAD R7 (1 pin)			
	UART0 RX (1 pin) UART0 or 1 CTL (2 pins)				Interrupts (8 pins)
	PPI1 CLKFS (3 pins)				
5V IN	TWI0 (2 pins)				
Port F					
GPIO (16 pins)	PPI0 D0–15 (16 pins)				Interrupts (8 pins)
					Interrupts (8 pins)
Port G					
GPIO (16 pins)	PPI0 CLKFS (3 pins) DATA 16–17 (2 pins)	TMRCLK (1 pin)			Interrupts (8 pins)
	SPI1 SEL1–3 (3 pins)	HOST CTL (3 pins)	PPI2 clkfs (3 pins)	CZM (1 pin)	
	SPI1 (4 pins)	MXVR $\overline{MTXON}$ (1 pin)	TACI4-5 (2 pins)		Interrupts (8 pins)
	CAN0 (2 pins)				
	CAN1 (2 pins)				

Table 9. Pin Multiplexing

Primary Pin Function (Number of Pins)	First Peripheral Function	Second Peripheral Function	Third Peripheral Function	Fourth Peripheral Function	Interrupt Capability
Port H					
GPIO (14 pins)	UART1 (2 pins)	PPI1–2_FS3 (2 pins)	TACI1 (1 pin)		Interrupts (8 pins)
	ATAPI_RST (1 pin)	TMR8 (1 pin)	PPI0_FS3 (1 pin)		
	HOST_ADDR (1 pin)	TMR9 (1 pin)	Counter UD Dir (1 pin)		
	HOST_ACK (1 pin)	TMR10 (1 pin)	Counter UD Gate (1 pin)		
	MXVR MRX, MTX, $\overline{\text{MRXON}}$ (3 pins)		DMAR 0–1 (2 pins)	TACI8-10 (3 shared pins) TACLK8-10 (3 shared pins)	
		AMC ADDR 4-9 (6 pins)			Interrupts (6 pins)
Port I					
GPIO (16 pins)	ASYNC ADDR10–25 (16 pins)				Interrupts (8 pins)
					Interrupts (8 pins)
Port J					
GPIO (14 pins)	ASYNC CTL and MISC				Interrupts (8 pins)
					Interrupts (6 pins)

[1] A total of 32 interrupts at once are available from Ports C through J, configurable in byte-wide blocks.

ADSP-BF54x processor pin definitions are listed in Table 10. To see the pin multiplexing scheme, see Table 9.

Table 10. Pin Descriptions

Pin Name	I/O[1]	Function (First / Second / Third / Fourth)
Port A: GPIO / *SPORT2–3 / TMR4–7*		
PA0 / *TFS2*	I/O	GPIO / *SPORT2 Transmit Frame Sync*
PA1 / *DT2SEC / TMR4*	I/O	GPIO / *SPORT2 Transmit Data Secondary / Timer 4*
PA2 / *DT2PRI*	I/O	GPIO / *SPORT2 Transmit Data Primary*
PA3 / *TSCLK2*	I/O	GPIO / *SPORT2 Transmit Serial Clock*
PA4 / *RFS2*	I/O	GPIO / *SPORT2 Receive Frame Sync*
PA5 / *DR2SEC/TMR5*	I/O	GPIO / *SPORT2 Receive Data Secondary / Timer 5*
PA6 / *DR2PRI*	I/O	GPIO / *SPORT2 Receive Data Primary*
PA7 / *RSCLK2 / TACLK0*	I/O	GPIO / *SPORT2 Receive Serial Clock / Alternate Input Clock 0*
PA8 / *TFS3 / TACLK1*	I/O	GPIO / *SPORT3 Transmit Frame Sync / Alternate Input Clock 1*
PA9 / *DT3SEC / TMR6*	I/O	GPIO / *SPORT3 Transmit Data Secondary / Timer 6*
PA10 / *DT3PRI / TACLK2*	I/O	GPIO / *SPORT3 Transmit Data Primary / Alternate Input Clock 2*
PA11 / *TSCLK3 / TACLK3*	I/O	GPIO / *SPORT3 Transmit Serial Clock / Alternate Input Clock 3*
PA12 / *RFS3 / TACLK4*	I/O	GPIO / *SPORT3 Receive Frame Sync / Alternate Input Clock 4*
PA13 / *DR3SEC/TMR7 / TACLK5*	I/O	GPIO / *SPORT3 Receive Data Secondary / Timer 7 / Alternate Input Clock 5*
PA14 / *DR3PRI / TACLK6*	I/O	GPIO / *SPORT3 Receive Data Primary / Alternate Input Clock 6*
PA15 / *RSCLK3 / TACLK7 and TACI7*	I/O	GPIO / *SPORT3 Receive Serial Clock / Alt Input Clock 7 and Alt Capture Input 7*

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Port B: GPIO/*TWI/UART2–3/SPI2/TMR0–3*		
PB0/*SCL1*	I/O	GPIO/*TWI1 Serial Data*
PB1/*SDA1*	I/O	GPIO/*TWI1 Serial Data*
PB2/*UART3RTS*	I/O	GPIO/*UART3 Request To Send*
PB3/*UART3CTS*	I/O	GPIO/*UART3 Clear To Send*
PB4/*UART2TX*	I/O	GPIO/*UART2 Transmit*
PB5/*UART2RX/TACI2*	I/O	GPIO/*UART2 Receive/Alternate Capture Input 2*
PB6/*UART3TX*	I/O	GPIO/*UART3 Transmit*
PB7/*UART3RX/TACI3*	I/O	GPIO/*UART3 Receive/Alternate Capture Input 3*
PB8/*$\overline{SPI2SS}$/TMR0*	I/O	GPIO/*SPI2 Slave Select Input/Timer 0*
PB9/*SPI2SEL1/TMR1*	I/O	GPIO/*SPI2 Slave Select Enable 1/Timer 1*
PB10/*SPI2SEL2/TMR2*	I/O	GPIO/*SPI2 Slave Select Enable 2/Timer 2*
PB11/*SPI2SEL3/HWAIT*	I/O	GPIO/*SPI2 Slave Select Enable 3/Boot Wait*
PB12/*SPI2SCK*	I/O	GPIO/*SPI2 Clock*
PB13/*SPI2MOSI*	I/O	GPIO/*SPI2 Master Out Slave In*
PB14/*SPIMISO*	I/O	GPIO/*SPI2 Master In Slave Out*
Port C: GPIO/*SPORT0/SD/SDIO Controller/MXVR (MOST)*		
PC0/*TFS0*	I/O	GPIO/*SPORT0 Transmit Frame Sync*
PC1/*DT0SEC/MMCLK*	I/O	GPIO/*SPORT0 Transmit Data Secondary/MXVR Master Clock*
PC2/*DT0PRI*	I/O	GPIO/*SPORT0 Transmit Data Primary*
PC3/*TSCLK0*	I/O	GPIO/*SPORT0 Transmit Serial Clock*
PC4/*RFS0*	I/O	GPIO/*SPORT0 Receive Frame Sync*
PC5/*DR0SEC/MBCLK*	I/O	GPIO/*SPORT0 Receive Data Secondary/MXVR Bit Clock*
PC6/*DR0PRI*	I/O	GPIO/*SPORT0 Receive Data Primary*
PC7/*RSCLK0*	I/O	GPIO/*SPORT0 Receive Serial Clock*
PC8/*SD_D0*	I/O	GPIO/*SD Data Bus*
PC9/*SD_D1*	I/O	GPIO/*SD Data Bus*
PC10/*SD_D2*	I/O	GPIO/*SD Data Bus*
PC11/*SD_D3*	I/O	GPIO/*SD Data Bus*
PC12/*SD_CLK*	I/O	GPIO/*SD Clock Output*
PC13/*SD_CMD*	I/O	GPIO/*SD Command*

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Port D: GPIO/*PPI0–2/SPORT 1/Keypad/HOST*		
PD0/*PPI1_D0/HOST_D8/ TFS1/PPI0_D18*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Transmit Frame Sync/PPI0 Data*
PD1/*PPI1_D1/HOST_D9/ DT1SEC/PPI0_D19*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Transmit Data Secondary/PPI0 Data*
PD2/*PPI1_D2/HOST_D10/ DT1PRI/PPI0_D20*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Transmit Data Primary/PPI0 Data*
PD3/*PPI1_D3/HOST_D11/ TSCLK1/PPI0_D21*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Transmit Serial Clock/PPI0 Data*
PD4/*PPI1_D4/HOST_D12/RFS1/PPI0_D22*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Receive Frame Sync/PPI0 Data*
PD5/*PPI1_D5/HOST_D13/DR1SEC/PPI0_D23*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Receive Data Secondary/PPI0 Data*
PD6/*PPI1_D6/HOST_D14/DR1PRI*	I/O	GPIO/*PPI1 Data/HOST Data/SPORT 1 Receive Data Primary*
PD7/*PPI1_D7/HOST_D15/RSCLK1*	I/O	GPIO/*PPI1 Data /HOST Data/SPORT 1 Receive Serial Clock*
PD8/*PPI1_D8/HOST_D0/ PPI2_D0/KEY_ROW0*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Row Input*
PD9/*PPI1_D9/HOST_D1/PPI2_D1/KEY_ROW1*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Row Input*
PD10/*PPI1_D10/HOST_D2/PPI2_D2/KEY_ROW2*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Row Input*
PD11/*PPI1_D11/HOST_D3/PPI2_D3/KEY_ROW3*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Row Input*
PD12/*PPI1_D12/HOST_D4/PPI2_D4/KEY_COL0*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Column Output*
PD13/*PPI1_D13/HOST_D5/PPI2_D5/KEY_COL1*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Column Output*
PD14/*PPI1_D14/HOST_D6/PPI2_D6/KEY_COL2*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Column Output*
PD15/*PPI1_D15/HOST_D7/PPI2_D7/KEY_COL3*	I/O	GPIO/*PPI1 Data/HOST Data/PPI2 Data/Keypad Column Output*
Port E: GPIO/*SPI0/UART0/PPI1/TWI0/Keypad*		
PE0/*SPI0SCK/KEY_COL7*	I/O	GPIO/*SPI0 Clock/Keypad Column Output*
PE1/*SPI0MISO/KEY_ROW6*	I/O	GPIO/*SPI0 Master In Slave Out/Keypad Row Input*
PE2/*SPI0MOSI/KEY_COL6*	I/O	GPIO/*SPI0 Master Out Slave In/Keypad Column Output*
PE3/*$\overline{\text{SPI0SS}}$/KEY_ROW5*	I/O	GPIO/*SPI0 Slave Select Input/Keypad Row Input*
PE4/*SPI0SEL1/KEY_COL5*	I/O	GPIO/*SPI0 Slave Select Enable 1/Keypad Column Output*
PE5/*SPI0SEL2/KEY_ROW4*	I/O	GPIO/*SPI0 Slave Select Enable 2/Keypad Row Input*
PE6/*SPI0SEL3/KEY_COL4*	I/O	GPIO/*SPI0 Slave Select Enable 3/Keypad Column Output*
PE7/*UART0TX/KEY_ROW7*	I/O	GPIO/*UART0 Transmit/Keypad Row Input*
PE8/*UART0RX/TACI0*	I/O	GPIO/*UART0 Receive/Alternate Capture Input 0*
PE9/*UART1RTS*	I/O	GPIO/*UART1 Request To Send*
PE10/*UART1CTS*	I/O	GPIO/*UART1 Clear To Send*
PE11/*PPI1_CLK*	I/O	GPIO/*PPI Clock*
PE12/*PPI1_FS1*	I/O	GPIO/*PPI Frame Sync 1*
PE13/*PPI1_FS2*	I/O	GPIO/*PPI Frame Sync 2*
PE14/*SCL0*	I/O	GPIO/*TWI0 Serial Data*
PE15/*SDA0*	I/O	GPIO/*TWI0 Serial Data*

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Port F: GPIO / *PPI0*		
PF0/*PPI0_D0*	I/O	GPIO/*PPI Data*
PF1/*PPI0_D1*	I/O	GPIO/*PPI Data*
PF2/*PPI0_D2*	I/O	GPIO/*PPI Data*
PF3/*PPI0_D3*	I/O	GPIO/*PPI Data*
PF4/*PPI0_D4*	I/O	GPIO/*PPI Data*
PF5/*PPI0_D5*	I/O	GPIO/*PPI Data*
PF6/*PPI0_D6*	I/O	GPIO/*PPI Data*
PF7/*PPI0_D7*	I/O	GPIO/*PPI Data*
PF8/*PPI0_D8*	I/O	GPIO/*PPI Data*
PF9/*PPI0_D9*	I/O	GPIO/*PPI Data*
PF10/*PPI0_D10*	I/O	GPIO/*PPI Data*
PF11/*PPI0_D11*	I/O	GPIO/*PPI Data*
PF12/*PPI0_D12*	I/O	GPIO/*PPI Data*
PF13/*PPI0_D13*	I/O	GPIO/*PPI Data*
PF14/*PPI0_D14*	I/O	GPIO/*PPI Data*
PF15/*PPI0_D15*	I/O	GPIO/*PPI Data*
Port G: GPIO / *PPI0 / SPI1 / PPI2 / Up-Down Counter / CAN0–1 / HOST / MXVR (MOST)*		
PG0/*PPI0_CLK/TMRCLK*	I/O	GPIO/*PPI Clock/External Timer Reference*
PG1/*PPI0_FS1*	I/O	GPIO/*PPI Frame Sync 1*
PG2/*PPI0_FS2*	I/O	GPIO/*PPI Frame Sync 2*
PG3/*PPI0_D16*	I/O	GPIO/*PPI Data*
PG4/*PPI0_D17*	I/O	GPIO/*PPI Data*
PG5/*SPI1SEL1/$\overline{\text{HOST\_CE}}$/PPI2_FS2/ CZM*	I/O	GPIO/*SPI1 Slave Select/HOST $\overline{\text{CE}}$/PPI2 Frame Sync/Counter Zero Marker*
PG6/*SPI1SEL2/$\overline{\text{HOST\_RD}}$/ PPI2_FS1*	I/O	GPIO/*SPI1 Slave Select/HOST $\overline{\text{RD}}$/PPI2 Frame Sync*
PG7/*SPI1SEL3/$\overline{\text{HOST\_WR}}$/ PPI2_CLK*	I/O	GPIO/*SPI1 Slave Select/HOST $\overline{\text{WR}}$/PPI2 Clock*
PG8/*SPI1SCK*	I/O	GPIO/*SPI1 Clock*
PG9/*SPI1MISO*	I/O	GPIO/*SPI1 Master In Slave Out*
PG10/*SPI1MOSI*	I/O	GPIO/*SPI2 Master Out Slave In*
PG11/*$\overline{\text{SPI1SS}}$/$\overline{\text{MTXON}}$*	I/O	GPIO/*SPI1 Slave Select Input/MXVR Transmit Phy On*
PG12/*CAN0TX*	I/O	GPIO/*CAN0 Transmit*
PG13/*CAN0RX/TACI4*	I/O	GPIO/*CAN0 Receive/Alternate Capture Input 4*
PG14/*CAN1TX*	I/O	GPIO/*CAN1 Transmit*
PG15/*CAN1RX/TACI5*	I/O	GPIO/*CAN1 Receive/Alternate Capture Input 5*

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Port H: GPIO / *AMC / EXTDMA / UART1 / PPI0–2 / ATAPI Interface / Up-Down Counter / Timerx 0–2 / HOST / MXVR (MOST)*		
PH0/*UART1_TX/PPI1_FS3*	I/O	GPIO/*UART1 Transmit/PPI1 Frame Sync*
PH1/*UART1_RX/PPI2_FS3/TACI1*	I/O	GPIO/*UART 1 Receive/PPI2 Frame Sync/Alternate Capture Input 1*
PH2/*ATAPI_RESET/TMR8/PPI0_FS3*	I/O	GPIO/*ATAPI Interface Hard Reset Signal/Timer 8/PPI0 Frame Sync*
PH3/*HOST_ADDR/TMR9/CUD*	I/O	GPIO/*HOST Address/Timer 9/Counter Up and Direction*
PH4/*HOST_ACK/TMR10/CDG*	I/O	GPIO/*HOST Acknowledge/Timer 10/Counter Down and Gate*
PH5/*MTX/DMAR0/TACI8 and TACLK8*	I/O	GPIO/*MXVR Transmit Data/Ext. DMA Request/Alt Capt. In. 8 and Alt In. Clk 8*
PH6/*MRX/DMAR1/TACI9 and TACLK9*	I/O	GPIO/*MXVR Receive Data/Ext. DMA Request/Alt Capt. In. 9 and Alt In. Clk 9*
PH7/*$\overline{\text{MRXON}}$/HWAIT/TACI10 and TACLK10*	I/O	GPIO/*MXVR Receive Phy On/Boot Host Wait/Alt Capt. In. 10 and Alt In. Clk 10*
PH8/*A4*	I/O	GPIO/*Address Bus for Async Access*
PH9/*A5*	I/O	GPIO/*Address Bus for Async Access*
PH10/*A6*	I/O	GPIO/*Address Bus for Async Access*
PH11/*A7*	I/O	GPIO/*Address Bus for Async Access*
PH12/*A8*	I/O	GPIO/*Address Bus for Async Access*
PH13/*A9*	I/O	GPIO/*Address Bus for Async Access*
Port I: GPIO / *AMC*		
PI0/*A10*	I/O	GPIO/*Address Bus for Async Access*
PI1/*A11*	I/O	GPIO/*Address Bus for Async Access*
PI2/*A12*	I/O	GPIO/*Address Bus for Async Access*
PI3/*A13*	I/O	GPIO/*Address Bus for Async Access*
PI4/*A14*	I/O	GPIO/*Address Bus for Async Access*
PI5/*A15*	I/O	GPIO/*Address Bus for Async Access*
PI6/*A16*	I/O	GPIO/*Address Bus for Async Access*
PI7/*A17*	I/O	GPIO/*Address Bus for Async Access*
PI8/*A18*	I/O	GPIO/*Address Bus for Async Access*
PI9/*A19*	I/O	GPIO/*Address Bus for Async Access*
PI10/*A20*	I/O	GPIO/*Address Bus for Async Access*
PI11/*A21*	I/O	GPIO/*Address Bus for Async Access*
PI12/*A22*	I/O	GPIO/*Address Bus for Async Access*
PI13/*A23*	I/O	GPIO/*Address Bus for Async Access*
PI14/*A24*	I/O	GPIO/*Address Bus for Async Access*
PI15/*A25*	I/O	GPIO/*Address Bus for Async Access*

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Port J: GPIO / *AMC* / *ATAPI Controller*		
PJ0/*ARDY*/*WAIT*	I/O	GPIO/*Hardware Ready Control*
PJ1/$\overline{ND\_CE}$	I/O	GPIO/*NAND Chip Enable*
PJ2/$\overline{ND\_RB}$	I/O	GPIO/*Ready Busyb Signal*
PJ3/*ATAPI_DIORB*	I/O	GPIO/*ATAPI RDb*
PJ4/*ATAPI_DIOWB*	I/O	GPIO/*ATAPI WBb*
PJ5/*ATAPI_CS0B*	I/O	GPIO/*ATAPI Chip Select Signal Command Block*
PJ6/*ATAPI_CS1B*	I/O	GPIO/*ATAPI Chip Select Signal*
PJ7/*ATAPI_DMACKB*	I/O	GPIO/*ATAPI DMA Acknowledge Signal*
PJ8/*ATAPI_DMARQ*	I/O	GPIO/*ATAPI DMA Request Signal*
PJ9/*ATAPI_INTRQ*	I/O	GPIO/*Interrupt Request from the Device*
PJ10/*ATAPI_IORDY*	I/O	GPIO/*ATAPI Ready Handshake Signal*
PJ11/$\overline{BR}$	I/O	GPIO/*Bus Request*
PJ12/$\overline{BG}$	I/O	GPIO/*Bus Grant*
PJ13/$\overline{BGH}$	I/O	GPIO/*Bus Grant Hang*
Memory Interface		
DA0–12	O	DDR Address Bus
DBA0–1	O	DDR Bank Active Strobe
DQ0–15	I/O	DDR Data Bus
DQS0–1	I/O	DDR Data Strobe
DQM0–1	O	DDR Data Mask for Reads and Writes
DCLK1–2	O	DDR Output Clock
$\overline{DCLK1–2}$	O	DDR Complementary Output Clock
$\overline{DCS0–1}$	O	DDR Chip Selects
DCKE	O	DDR Clock Enable
$\overline{DRAS}$	O	DDR Row Address Strobe
$\overline{DCAS}$	O	DDR Column Address Strobe
$\overline{DWE}$	O	DDR Write Enable
DDR_V_{REF}	I	DDR Voltage Reference
DDR_V_{SSR}	I	DDR Voltage Reference Shield (connect to GND)
Asynchronous Memory Interface		
ADDR1–3	O	Address Bus for SRAM, NAND, NOR, and ATAPI Accesses
D0–15	I/O	Data Bus for SRAM, NAND, NOR, and ATAPI Accesses
$\overline{AMS0–3}$	O	Bank Selects
$\overline{ABE0}$/*NR_CLK*/*ND_CLE*	O	Byte Enables:Data Masks for Asynchronous Access/*NOR Clock*/*NAND Clock Enable*
$\overline{ABE1}$/*ND_ALE*	O	Byte Enables:Data Masks for Asynchronous Access/*NAND Address Latch Enable*
$\overline{AOE}$/*NR_ADV*	O	Output Enable/*NOR Address Data Valid*
$\overline{ARE}$	O	Read Enable/*NOR Output Enable*
$\overline{AWE}$	O	Write Enable
ATAPI Controller Pins		
ATAPI_PDIAG	I	

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
USB 2.0 HS OTG Pins		
USB_DP	I/O	USB D+ pin
USB_DM	I/O	USB D- pin
USB_XI	C	Clock XTAL input 1
USB_XO	C	Clock XTAL input 2
USB_ID	I	USB ID pin
USB_VBUS	I/O	USB VBUS pin
USB_V_{REF}	O	USB voltage reference source (Test purposes only)
USB_RSET	O	USB resistance set (Test purposes only)
MXVR (MOST) Interface		
MFS	O	MXVR Frame Sync
MLF_P	A	MXVR Loop Filter Plus
MLF_M	A	MXVR Loop Filter Minus
MXI	C	MXVR Crystal Input
MXO	C	MXVR Crystal Output
Mode Control Pins		
BMODE0–3	I	Boot Mode Strap 0–3
JTAG Port Pins		
TDI	I	JTAG Serial Data In
TDO	O	JTAG Serial Data Out
$\overline{\text{TRST}}$[2]	I	JTAG Reset
TMS	I	JTAG Mode Select
TCK	I	JTAG Clock
$\overline{\text{EMU}}$	O	Emulation Output
Voltage Regulator		
$V_{ROUT0-1}$	O	External FET/BJT Drives
Real Time Clock		
RTXO	C	RTC Crystal Output
RTXI	C	RTC Crystal Input
Clock (PLL) Pins		
CLKIN	C	Clock/Crystal Input
CLKOUT	O	Clock Output
XTAL	C	Crystal Output
CLKBUF	O	Buffered Oscillator Output
EXT_WAKE	O	External Wakeup Output
$\overline{\text{RESET}}$	I	Reset
$\overline{\text{NMI}}$[3]	I	Non-maskable Interrupt

Table 10. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)
Supplies		
V_{DDINT}	P	Internal Power Supply
V_{DDEXT}	P	External Power Supply
V_{DDDDR}	P	External DDR Power Supply
V_{DDUSB}	P	External USB Power Supply
V_{DDRTC}	P	RTC Clock Supply
GND	G	Ground
V_{DDMC}	P	MXVR Crystal Power Supply
GND_{MC}	G	MXVR Crystal Ground
V_{DDMX}	P	MXVR I/O Power Supply
GND_{MX}	G	MXVR I/O Ground
V_{DDMP}	P	MXVR PLL Power Supply
GND_{MP}	G	MXVR PLL Ground

[1] I = Input, O = Output, P =Power, G = Ground, C = Crystal, A = Analog.
[2] This pin should be pulled LOW if the JTAG port will not be used.
[3] This pin should always be pulled HIGH when not used.

SPECIFICATIONS

Note that component specifications are subject to change without notice.

OPERATING CONDITIONS

Parameter[1]		Minimum	Nominal	Maximum	Unit
V_{DDINT}	Internal Supply Voltage	0.8	TBD[2]	TBD[2]	V
	Internal Supply Voltage for Automotive Grade	1.0	TBD[2]	TBD[2]	V
V_{DDEXT}	External Supply Voltage[3]	2.25	2.5 or 3.3	3.6	V
	External Supply Voltage for Automotive Grade	2.7	3.3	3.6	V
V_{DDUSB}	USB External Supply Voltage	3.0	3.3	3.6	V
V_{DDMC}	MXVR Crystal Supply Voltage	3.0	3.3	3.6	V
V_{DDMX}	MXVR I/O Supply Voltage	3.0	3.3	3.6	V
V_{DDMP}	MXVR PLL Supply Voltage	1.0	TBD[2]	TBD[2]	V
V_{DDRTC}	Real Time Clock Power Supply Voltage	2.25	2.5 or 3.3	3.6	V
	Real Time Clock Power Supply Voltage for Automotive Grade	2.7	3.3	3.6	V
V_{DDDDR}	DDR Memory Supply Voltage	2.25	2.5	2.75	V
V_{IH}	High Level Input Voltage[3,4], @ V_{DDEXT} = maximum	2.0		3.6	V
$V_{IHCLKIN}$	High Level Input Voltage[5], @ V_{DDEXT} = maximum	2.2		3.6	V
V_{IHDDR}	High Level Input Voltage[6], @ V_{DDDDR} = maximum	1.7		3.0	V
V_{IH5V}	High Level Input Voltage[7], @ V_{DDEXT} = maximum	2.0		5.5	V
V_{IL}	Low Level Input Voltage[3, 8], @ V_{DDEXT} = minimum	–0.3		0.6	V
V_{IL5V}	Low Level Input Voltage[9], @ V_{DDEXT} = minimum	–0.3		0.8	V
V_{ILDDR}	Low Level Input Voltage[6], @ V_{DDEXT} = minimum	–0.3		0.8	V
T_J	Junction Temperature, 400-Ball Chip Scale Ball Grid Array (mini-BGA) @$T_{AMBIENT}$ = –40ºC to +85ºC	–40		+105	ºC
T_J	Junction Temperature, 400-Ball Chip Scale Ball Grid Array (mini-BGA) @$T_{AMBIENT}$ = 0ºC to +70ºC	0		+90	ºC

[1] Specifications subject to change without notice.

[2] It is recommended that preliminary designs be designed with an adjustable voltage regulator which supports 0.8 V to 1.6 V.

[3] The ADSP-BF54x processor 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 pins (D15–0, PA15–0, PB14–0, PC15–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ14–0) and input only pins (ATAPI_PDIAG, USB_ID, TCK, TDI, TMS, $\overline{\text{TRST}}$, CLKIN, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, and BMODE3–0).

[4] Parameter value applies to all input and bi-directional pins, except CLKIN, PB0, PB1, PE14, PE15, PG15–11, PH6, PH7, and the pins listed in table note 6 of the Operating Conditions table.

[5] Parameter value applies to CLKIN pin only.

[6] Parameter value applies to DA0–12, DBA0–1, DQ0–15, DQS0–1, DQM0–1, DCLK1–2, $\overline{\text{DCLK1–2}}$, $\overline{\text{DCS0–1}}$, DCKE, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, and $\overline{\text{DWE}}$ pins only.

[7] Certain ADSP-BF54x processor pins are 5.0 V tolerant (accept up to 5.5 V maximum V_{IH} when power is applied to V_{DDEXT} pins). Voltage compliance on outputs (V_{OH}) depends on the input V_{DDEXT}, because V_{OH} (maximum) approximately equals V_{DDEXT} (maximum). The 5.0 V tolerance feature applies to PB0, PB1, PE14, PE15, PG15–11, PH6, and PH7 pins only. The 5.0 V tolerance exists only when power is applied to the V_{DDEXT} pins. The PB0, PB1, PE14, and PE15 pins are open drain (regardless of pin functionality) and therefore require a pullup resistor. Consult the I^2C specification version 2.1 for the proper resistor value.

[8] Parameter value applies to all input and bi-directional pins, except PB0, PB1, PE14, PE15, PG15–11, PH6, and PH7.

[9] Parameter value applies to the following pins only: PB0, PB1, PE14, PE15, PG15–11, PH6, and PH7.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Typical	Max	Unit
V_{OH}	High Level Output Voltage[1]	@ V_{DDEXT} = Minimum, I_{OH} = –0.5 mA	2.4			V
V_{OHDDR}	High Level Output Voltage[2]	@ V_{DDDDR} = Minimum	TBD			V
V_{OL}	Low Level Output Voltage[1]	@ V_{DDEXT} = Minimum, I_{OL} = 2.0 mA			0.4	V
V_{OLDDR}	Low Level Output Voltage[2]	@ V_{DDDDR} = Minimum			TBD	V
I_{IH}	High Level Input Current[3]	@ V_{DDEXT} = Maximum, V_{IN} = V_{IH} Maximum			10.0	µA
I_{IHP}	High Level Input Current JTAG[4]	@ V_{DDEXT} = Maximum, V_{IN} = V_{IH} Maximum			50.0	µA
I_{IL}[5]	Low Level Input Current[3]	@ V_{DDEXT} = Maximum, V_{IN} = 0 V			10.0	µA
I_{ILP}[5]	Low Level Input Current JTAG[4]	@ V_{DDEXT} = Maximum, V_{IN} = 0 V			TBD	µA
I_{OZH}	Three-State Leakage Current[6]	@ V_{DDEXT} = Maximum, V_{IN} = V_{IH} Maximum			10.0	µA
I_{OZL}[5]	Three-State Leakage Current[6]	@ V_{DDEXT} = Maximum, V_{IN} = 0 V			10.0	µA
C_{IN}	Input Capacitance[7]	f_{IN} = TBD MHz, $T_{AMBIENT}$ = TBD°C, V_{IN} = TBD V	TBD		TBD	pF
$I_{DDHIBERNATE}$	TBD	TBD		TBD		µA
$I_{DDDEEPSLEEP}$	TBD	TBD		TBD		mA
$I_{DDSLEEP}$	TBD	TBD		TBD		mA
I_{DDTYP}	TBD	TBD		TBD		mA
I_{DDRTC}	TBD	TBD		TBD		µA

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

ESD SENSITIVITY

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-BF54x processor 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.

ABSOLUTE MAXIMUM RATINGS

Internal (Core) Supply Voltage[1] (V_{DDINT})	–0.3 V to +1.4 V
External (I/O) Supply Voltage[1] (V_{DDEXT})	–0.3 V to +3.8 V
Input Voltage[1,2]	–0.5 V to +3.6 V
Output Voltage Swing[1]	–0.5 V to V_{DDEXT}+0.5 V
Load Capacitance[1]	200 pF
Storage Temperature Range[1]	–65°C to +150°C
Junction Temperature Underbias[1]	+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] Applies to all bidirectional and input only pins except PB0, PB1, PE14, PE15, PG15–11, PH6, and PH7. Absolute maximum input voltage range on pins PB0, PB1, PE14, PE15, PG15–11, PH6, and PH7 is –0.5 V to +5.5 V.

PACKAGE INFORMATION

The information presented in Figure 8 and Table 11 provides information about how to read the package brand and relate it to specific product features. For a complete listing of product offerings, see the Ordering Guide on Page 64.

Figure 8. Product Information on Package

Table 11. Package Information

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

TIMING SPECIFICATIONS

Table 12, Table 13, and Table 14 describe the timing requirements for the ADSP-BF54x processor clocks. Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock and system clock. Table 15 describes phase-locked loop operating conditions.

Table 12. Core Clock Requirements—600 MHz Speed Grade[1]

Parameter		Minimum	Maximum	Unit
f_{CCLK}	Core Clock Frequency (V_{DDINT}=TBD V minimum)		600	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=1.045 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.95 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.85 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.8 V)		TBD	MHz

[1] The speed grade of a given part may be seen on the Ordering Guide on Page 64. It stands for the maximum allowed CCLK frequency at V_{DDINT} = minimum and the maximum allowed VCO frequency at any supply voltage.

Table 13. Core Clock Requirements—533 MHz Speed Grade[1]

Parameter		Minimum	Maximum	Unit
f_{CCLK}	Core Clock Frequency (V_{DDINT}=TBD V minimum)		533	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=1.045 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.95 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.85 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.8 V)		TBD	MHz

[1] The speed grade of a given part may be seen on the Ordering Guide on Page 64. It stands for the maximum allowed CCLK frequency at V_{DDINT} = minimum and the maximum allowed VCO frequency at any supply voltage.

Table 14. Core Clock Requirements—400 MHz Speed Grade[1]

Parameter		Minimum	Maximum	Unit
f_{CCLK}	Core Clock Frequency (V_{DDINT}=TBD V minimum)		400	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=1.045 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}= 0.95 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.85 V minimum)		TBD	MHz
f_{CCLK}	Core Clock Frequency (V_{DDINT}=0.8 V)		TBD	MHz

[1] The speed grade of a given part may be seen on the Ordering Guide on Page 64. It stands for the maximum allowed CCLK frequency at V_{DDINT} = minimum and the maximum allowed VCO frequency at any supply voltage.

Table 15. Phase-Locked Loop Operating Conditions

Parameter		Minimum	Maximum	Unit
f_{VCO}	Voltage Controlled Oscillator (VCO) Frequency	50	Speed Grade[1]	MHz

[1] The speed grade of a given part may be seen on the "Ordering Guide" on page 64. It stands for the Maximum allowed CCLK frequency at V_{DDINT} = minimum and the maximum allowed VCO frequency at any supply voltage.

Table 16. Clock Input and Reset Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
t_{CKIN}	CLKIN Period	20.0	100.0	ns
t_{CKINL}	CLKIN Low Pulse[1]	8.0		ns
t_{CKINH}	CLKIN High Pulse[1]	8.0		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulsewidth Low[2]	11 t_{CKIN}		ns

[1] Applies to bypass mode and non-bypass mode.

[2] Applies after power-up sequence is complete. At power-up, the processor's internal phase locked loop requires no more than 2000 CLKIN cycles, while $\overline{\text{RESET}}$ is asserted, assuming stable power supplies and CLKIN (not including startup time of external clock oscillator).

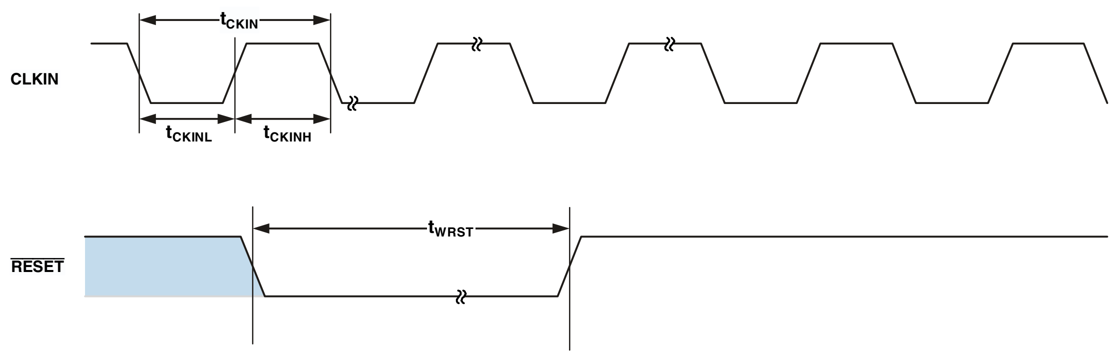


Figure 9. Clock and Reset Timing

Asynchronous Memory Read Cycle Timing

Table 17 and Table 18 on Page 36 and Figure 10 and Figure 11 on Page 36 describe asynchronous memory read cycle operations for synchronous and for asynchronous ARDY.

Table 17. Asynchronous Memory Read Cycle Timing with Synchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{SDAT}	DATA15–0 Setup Before CLKOUT	2.1		ns
t_{HDAT}	DATA15–0 Hold After CLKOUT	0.8		ns
t_{SARDY}	ARDY Setup Before the Falling Edge of CLKOUT	4.0		ns
t_{HARDY}	ARDY Hold After the Falling Edge of CLKOUT	0.0		ns
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT[1]	0.8		ns

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, $\overline{AOE}$, $\overline{ARE}$.

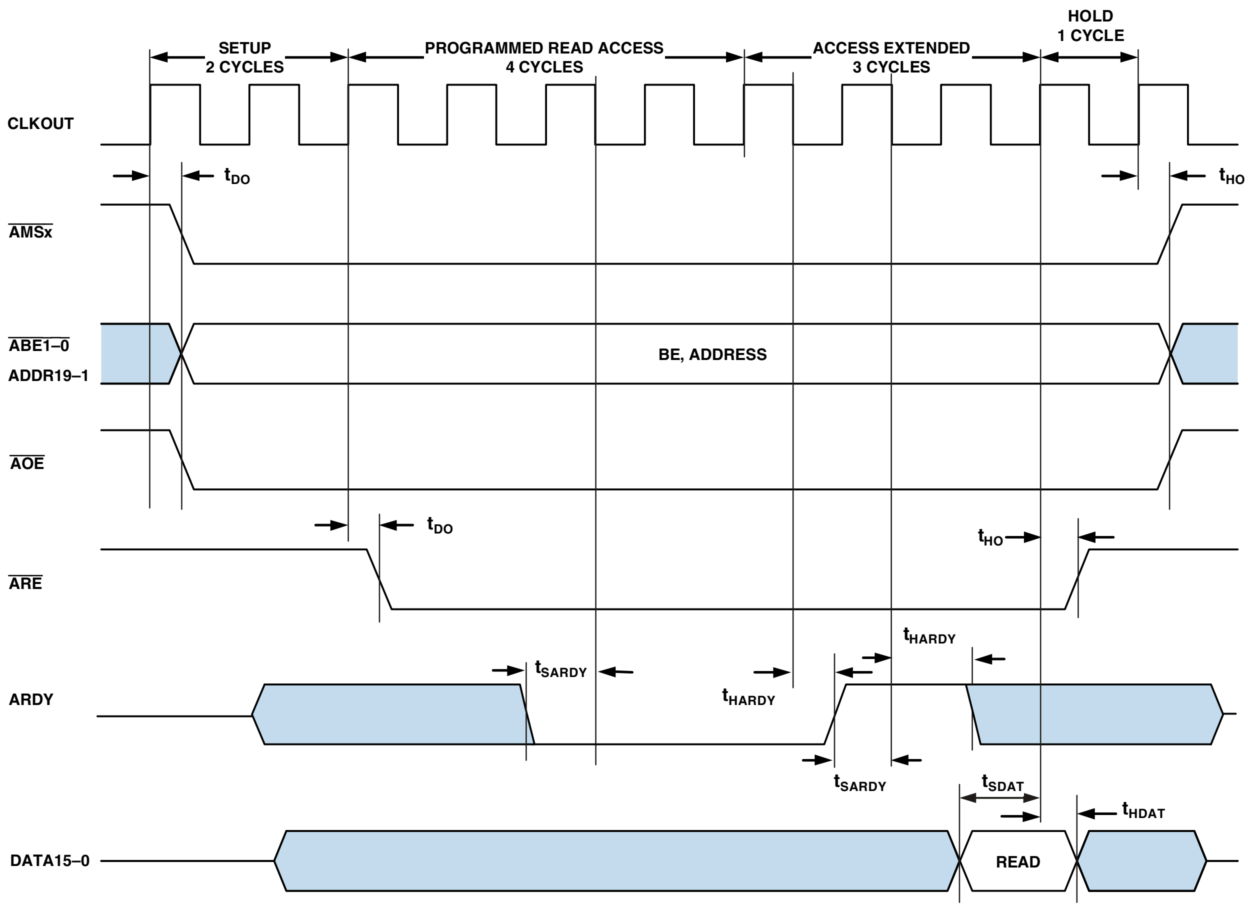


Figure 10. Asynchronous Memory Read Cycle Timing with Synchronous ARDY

Table 18. Asynchronous Memory Read Cycle Timing with Asynchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{SDAT}	DATA15–0 Setup Before CLKOUT	2.1		ns
t_{HDAT}	DATA15–0 Hold After CLKOUT	0.8		ns
t_{DANR}	ARDY Negated Delay from $\overline{AMSx}$ Asserted[1]		(S+RA–2)*t_{SCLK}	ns
t_{HAA}	ARDY Asserted Hold After $\overline{ARE}$ Negated	0.0		ns
t_{DO}	Output Delay After CLKOUT[2]		6.0	ns
t_{HO}	Output Hold After CLKOUT[2]	0.8		ns

[1] S = number of programmed setup cycles, RA = number of programmed read access cycles.
[2] Output pins include $\overline{AMS3–0}$, $\overline{ABE1–0}$, ADDR19–1, $\overline{AOE}$, $\overline{ARE}$.

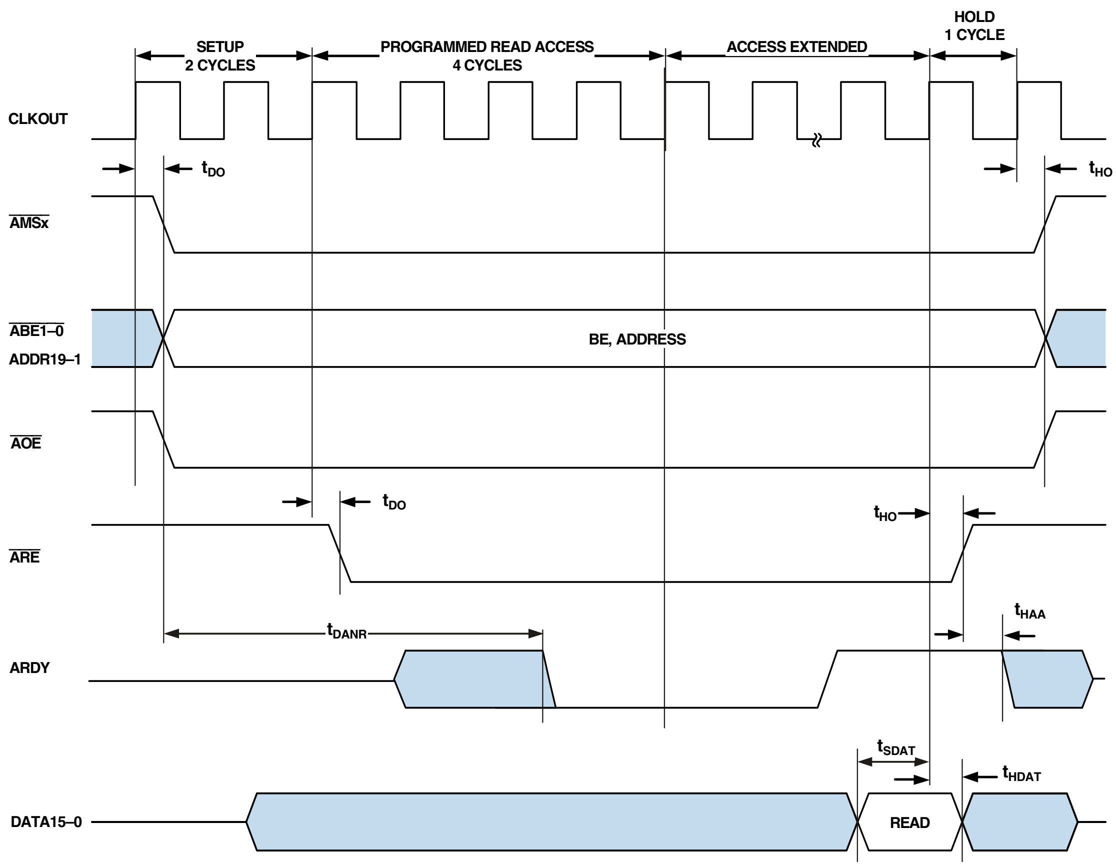


Figure 11. Asynchronous Memory Read Cycle Timing with Asynchronous ARDY

Asynchronous Memory Write Cycle Timing

Table 19 and Table 20 on Page 38 and Figure 12 and Figure 13 on Page 38 describe asynchronous memory write cycle operations for synchronous and for asynchronous ARDY.

Table 19. Asynchronous Memory Write Cycle Timing with Synchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{SARDY}	ARDY Setup Before the Falling Edge of CLKOUT	4.0		ns
t_{HARDY}	ARDY Hold After the Falling Edge of CLKOUT	0.0		ns
Switching Characteristics				
t_{DDAT}	DATA15–0 Disable After CLKOUT		6.0	ns
t_{ENDAT}	DATA15–0 Enable After CLKOUT	1.0		ns
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT[1]	0.8		ns

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, DATA15–0, $\overline{AOE}$, $\overline{AWE}$.

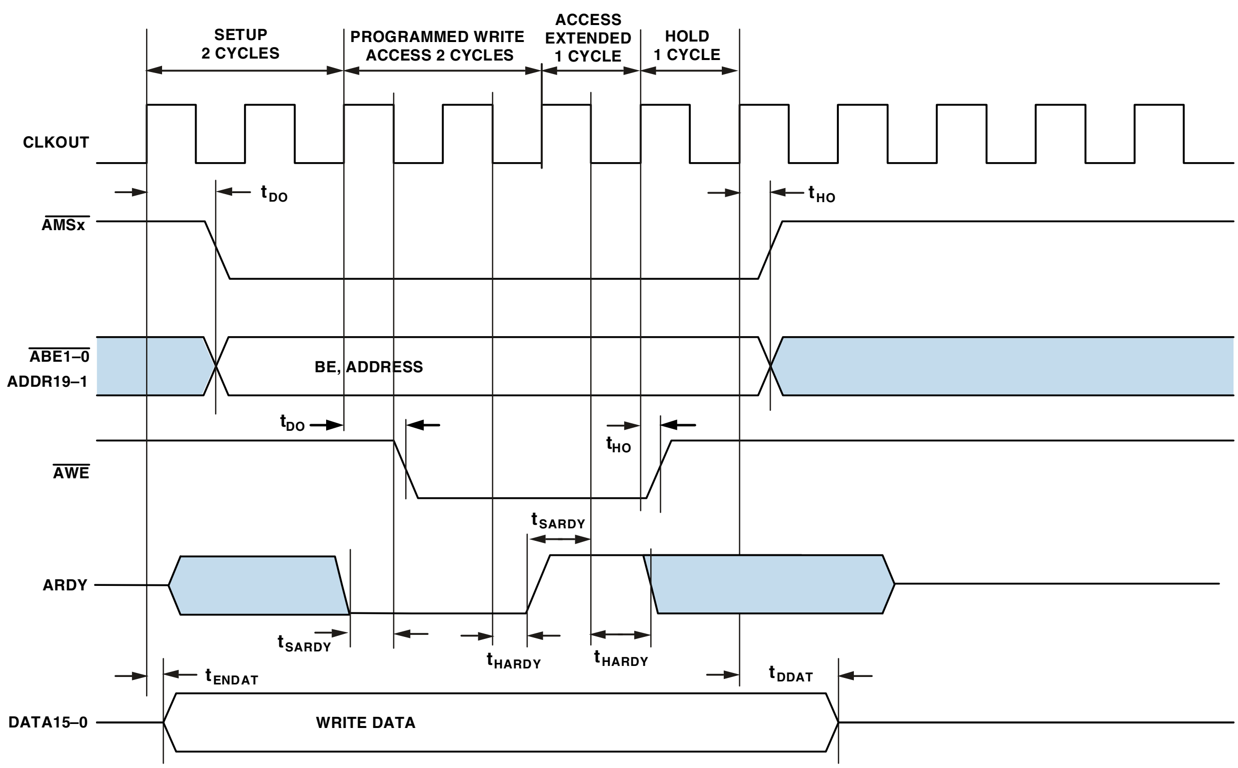


Figure 12. Asynchronous Memory Write Cycle Timing with Synchronous ARDY

Table 20. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{DANR}	ARDY Negated Delay from $\overline{AMSx}$ Asserted[1]		(S+WA−2)*t_{SCLK}	ns
t_{HAA}	ARDY Asserted Hold After $\overline{ARE}$ Negated	0.0		ns
Switching Characteristics				
t_{DDAT}	DATA15–0 Disable After CLKOUT		6.0	ns
t_{ENDAT}	DATA15–0 Enable After CLKOUT	1.0		ns
t_{DO}	Output Delay After CLKOUT[2]		6.0	ns
t_{HO}	Output Hold After CLKOUT[2]	0.8		ns

[1] S = number of programmed setup cycles, WA = number of programmed write access cycles.
[2] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, DATA15–0, $\overline{AOE}$, $\overline{AWE}$.

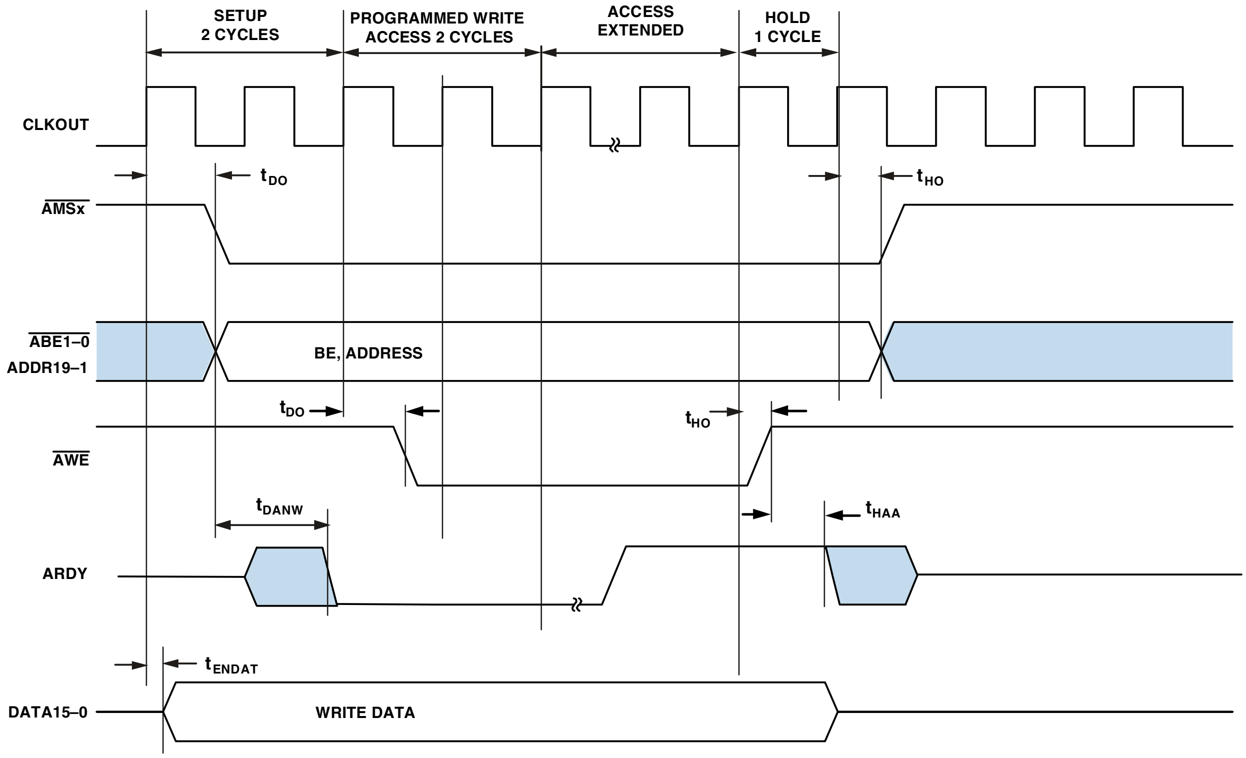


Figure 13. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY

Synchronous Memory DDR Read Cycle Timing

Table 21. Synchronous Memory DDR Read Cycle Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

Synchronous Memory DDR Read Cycle Timing is TBD

Figure 14. Synchronous Memory DDR Read Cycle Timing

Synchronous Memory DDR Write Cycle Timing

Table 22. Synchronous Memory DDR Write Cycle Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

Synchronous Memory DDR Write Cycle Timing is TBD

Figure 15. Synchronous Memory DDR Write Cycle Timing

Synchronous Memory Mobile DDR Read Cycle Timing

Table 23. Synchronous Memory DDR Read Cycle Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

Synchronous Memory Mobile DDR Read Cycle Timing is TBD

Figure 16. Synchronous Memory DDR Read Cycle Timing

Synchronous Memory Mobile DDR Write Cycle Timing

Table 24. Synchronous Memory DDR Write Cycle Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

Synchronous Memory Mobile DDR Write Cycle Timing is TBD

Figure 17. Synchronous Memory DDR Write Cycle Timing

External Port Bus Request and Grant Cycle Timing

Table 25 and Table 26 on Page 42 and Figure 18 and Figure 19 on Page 42 describe external port bus request and grant cycle operations for synchronous and for asynchronous $\overline{BR}$.

Table 25. External Port Bus Request and Grant Cycle Timing with Synchronous $\overline{BR}$

Parameter		Min	Max	Unit
Timing Requirements				
t_{BS}	$\overline{BR}$ Setup to Falling Edge of CLKOUT	4.0		ns
t_{BH}	Falling Edge of CLKOUT to $\overline{BR}$ Deasserted Hold Time	0.0		ns
Switching Characteristics				
t_{SD}	CLKOUT Low to $\overline{xMS}$, Address, and $\overline{RD}$/$\overline{WR}$ disable		4.5	ns
t_{SE}	CLKOUT Low to $\overline{xMS}$, Address, and $\overline{RD}$/$\overline{WR}$ enable		4.5	ns
t_{DBG}	CLKOUT High to $\overline{BG}$ High Setup		3.6	ns
t_{EBG}	CLKOUT High to $\overline{BG}$ Deasserted Hold Time		3.6	ns
t_{DBH}	CLKOUT High to $\overline{BGH}$ High Setup		3.6	ns
t_{EBH}	CLKOUT High to $\overline{BGH}$ Deasserted Hold Time		3.6	ns

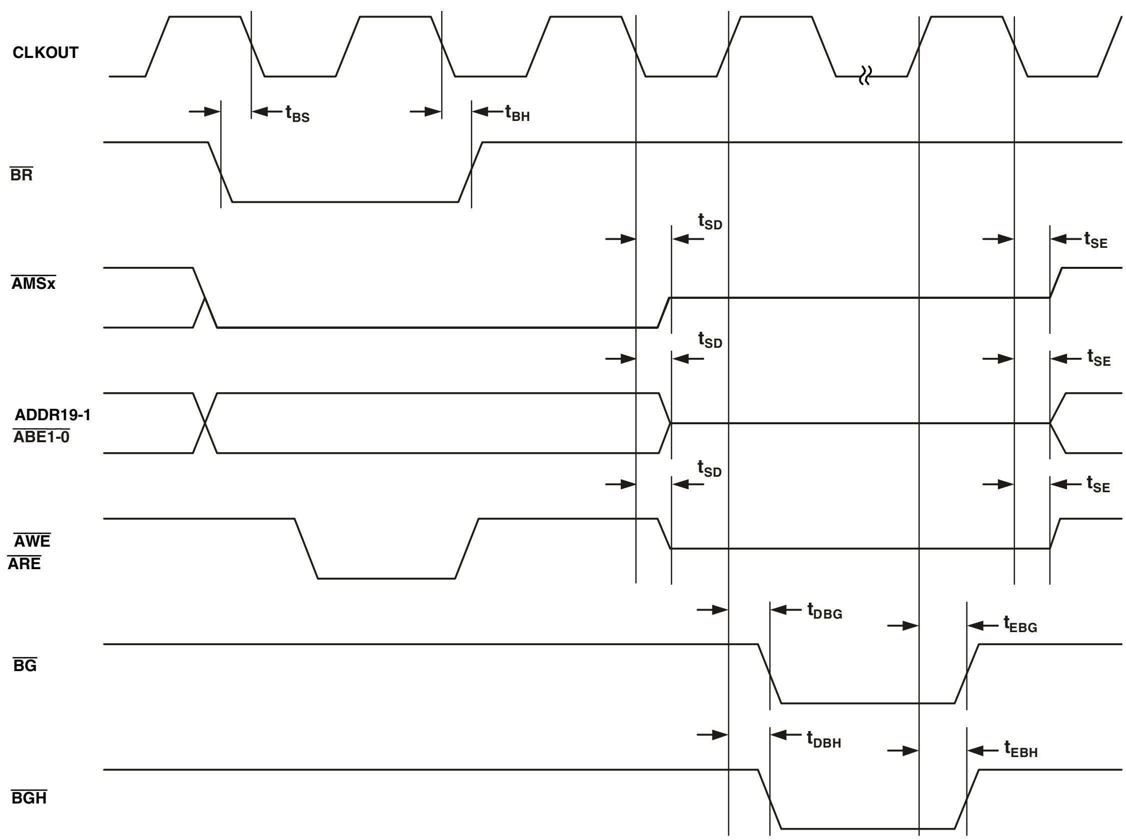


Figure 18. External Port Bus Request and Grant Cycle Timing with Synchronous $\overline{BR}$

Table 26. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{\text{BR}}$

Parameter		Min	Max	Unit
Timing Requirements				
t_{WBR}	$\overline{\text{BR}}$ Pulsewidth	2 x t_{SCLK}		ns
Switching Characteristics				
t_{SD}	CLKOUT Low to $\overline{\text{xMS}}$, Address, and $\overline{\text{RD}}/\overline{\text{WR}}$ disable		4.5	ns
t_{SE}	CLKOUT Low to $\overline{\text{xMS}}$, Address, and $\overline{\text{RD}}/\overline{\text{WR}}$ enable		4.5	ns
t_{DBG}	CLKOUT High to $\overline{\text{BG}}$ High Setup		3.6	ns
t_{EBG}	CLKOUT High to $\overline{\text{BG}}$ Deasserted Hold Time		3.6	ns
t_{DBH}	CLKOUT High to $\overline{\text{BGH}}$ High Setup		3.6	ns
t_{EBH}	CLKOUT High to $\overline{\text{BGH}}$ Deasserted Hold Time		3.6	ns

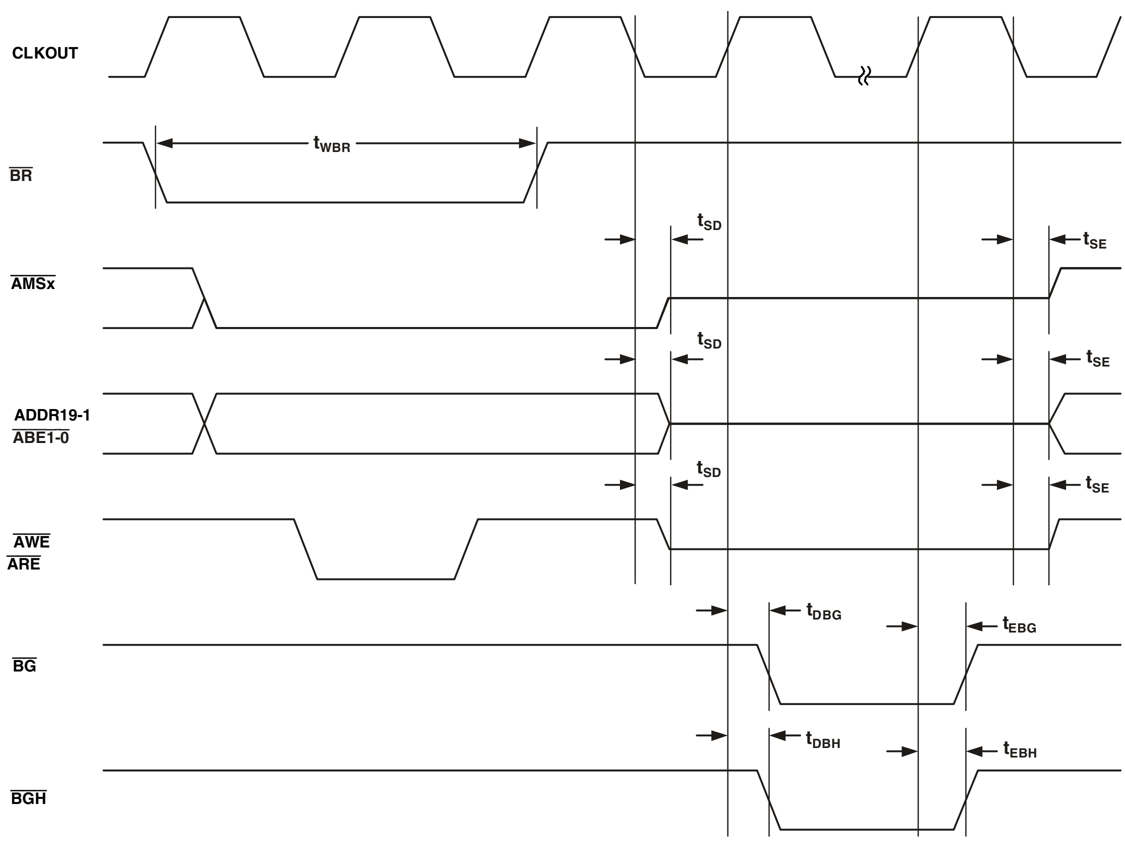


Figure 19. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{BR}$

Enhanced Parallel Peripheral Interface Timing

Table 27 and Figure 20 on Page 43 describes Enhanced Parallel Peripheral Interface operations.

Table 27. Enhanced Parallel Peripheral Interface Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
t_{PCLKW}	PPI_CLK Width	TBD		ns
t_{PCLK}	PPI_CLK Period	TBD		ns
Timing Requirements - GP Input and Frame Capture Modes				
t_{SFSPE}	External Frame Sync Setup Before PPI_CLK	TBD		ns
t_{HFSPE}	External Frame Sync Hold After PPI_CLK	TBD		ns
t_{SDRPE}	Receive Data Setup Before PPI_CLK	TBD		ns
t_{HDRPE}	Receive Data Hold After PPI_CLK	TBD		ns
Switching Characteristics - GP Output and Frame Capture Modes				
t_{DFSPE}	Internal Frame Sync Delay After PPI_CLK		TBD	ns
t_{HOFSPE}	Internal Frame Sync Hold After PPI_CLK	TBD		ns
t_{DDTPE}	Transmit Data Delay After PPI_CLK		TBD	ns
t_{HDTPE}	Transmit Data Hold After PPI_CLK	TBD		ns

Enhanced Parallel Peripheral Interface Timing is TBD

Figure 20. Enhanced Parallel Peripheral Interface Timing

Serial Ports Timing

Table 28 through Table 31 on Page 45 and Figure 21 on Page 45 through Figure 23 on Page 47 describe Serial Port operations.

Table 28. Serial Ports—External Clock

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

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

Table 29. Serial Ports—Internal Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSI}	TFS/RFS Setup Before TSCLK/RSCLK (externally generated TFS/RFS)[1]	8.0		ns
t_{HFSI}	TFS/RFS Hold After TSCLK/RSCLK (externally generated TFS/RFS)[1]	–1.5		ns
t_{SDRI}	Receive Data Setup Before RSCLK[1]	8.0		ns
t_{HDRI}	Receive Data Hold After RSCLK[1]	–1.5		ns
t_{SCLKEW}	TSCLK/RSCLK Width	4.5		ns
t_{SCLKE}	TSCLK/RSCLK Period	15.0		ns
Switching Characteristics				
t_{DFSI}	TFS/RFS Delay After TSCLK/RSCLK (Internally Generated TFS/RFS)[2]		3.0	ns
t_{HOFSI}	TFS/RFS Hold After TSCLK/RSCLK (Internally Generated TFS/RFS)[2]	–1.0		ns
t_{DDTI}	Transmit Data Delay After TSCLK[2]		3.0	ns
t_{HDTI}	Transmit Data Hold After TSCLK[2]	–2.0		ns
t_{SCLKIW}	TSCLK/RSCLK Width	4.5		ns

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

Table 30. Serial Ports—Enable and Three-State

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

[1] Referenced to drive edge.

Table 31. External Late Frame Sync

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

[1] MCE = 1, TFS enable and TFS valid follow $t_{DTENLFS}$ and $t_{DDTLFSE}$.

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

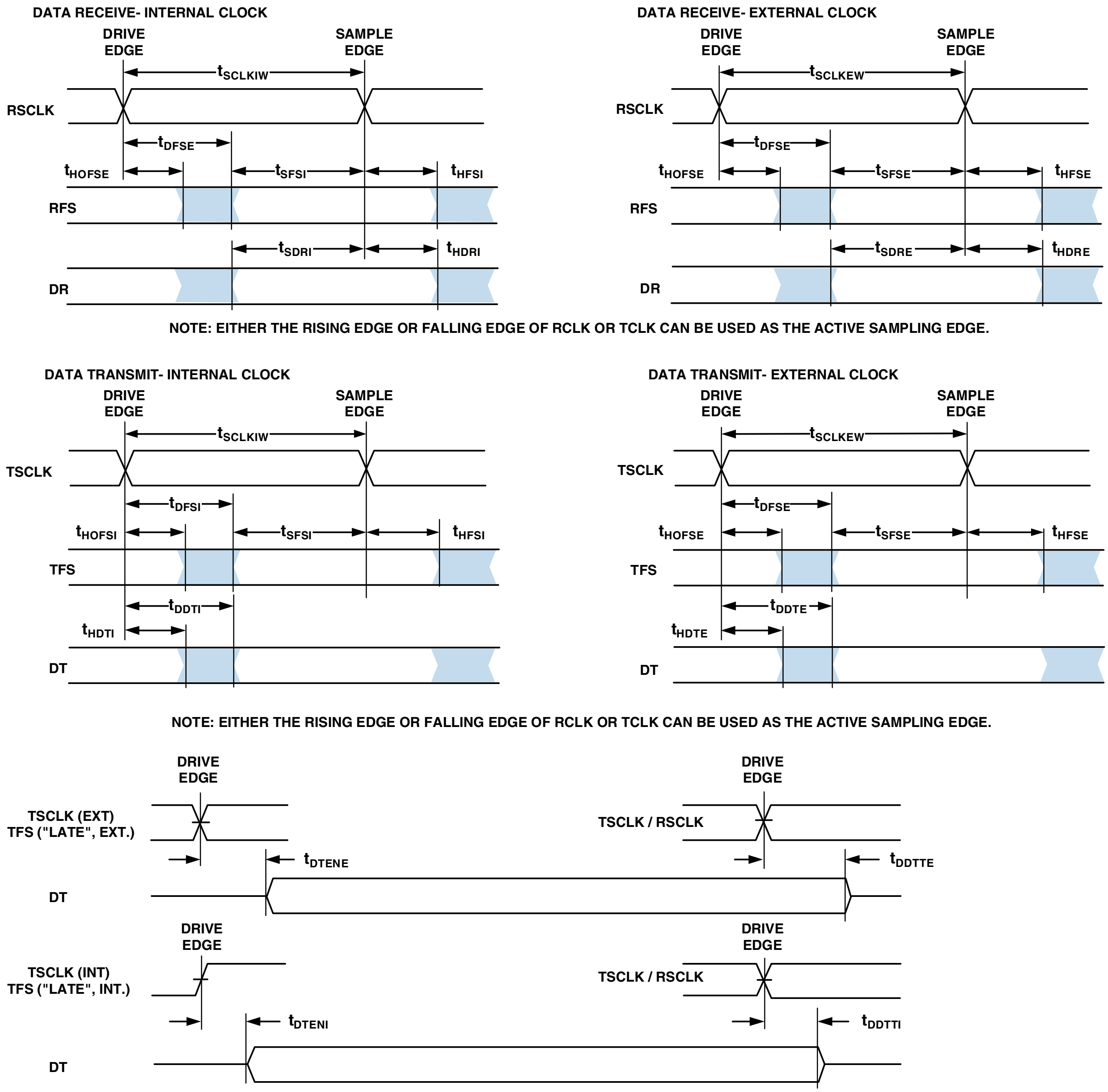


Figure 21. Serial Ports

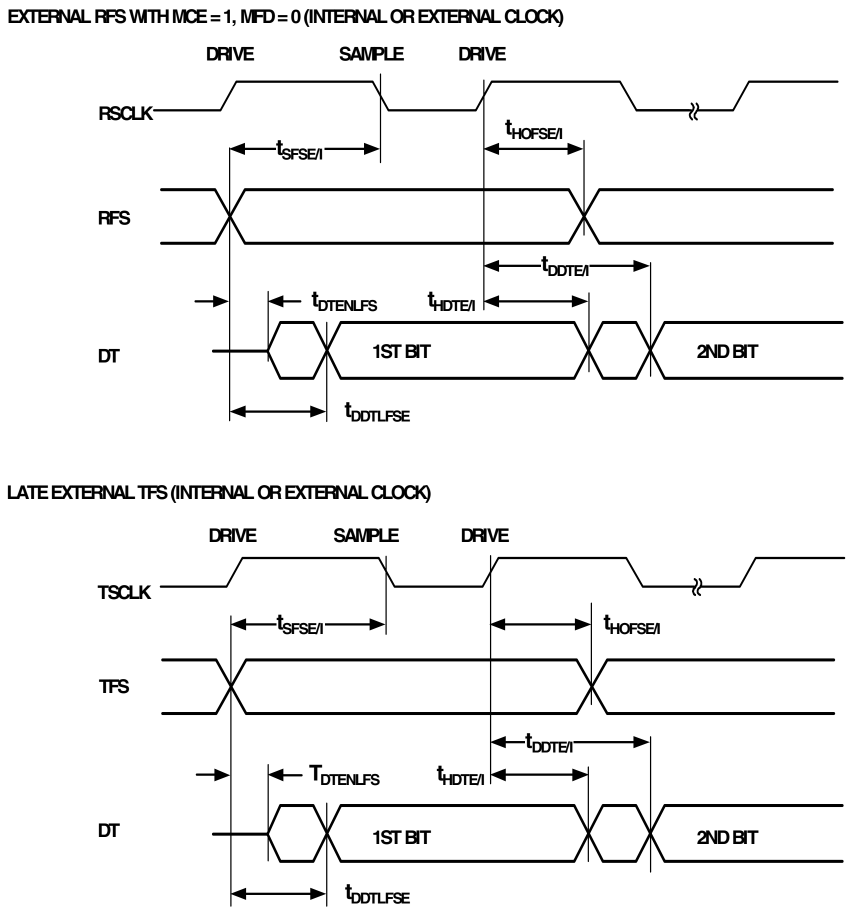


Figure 22. External Late Frame Sync (Frame Sync Setup < t_{SCLKE}/2)

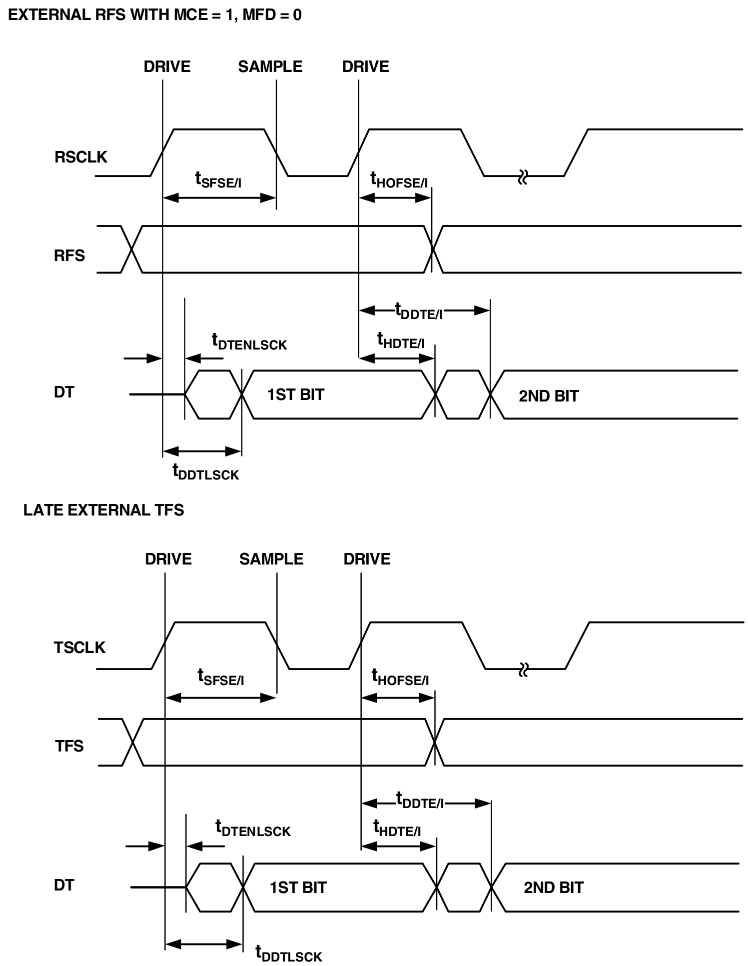


Figure 23. External Late Frame Sync (Frame Sync Setup > $t_{SCLKE}/2$)

Serial Peripheral Interface (SPI) Port—Master Timing

Table 32 and Figure 24 describe SPI port master operations.

Table 32. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
t_{SSPIDM}	Data input valid to SCK edge (data input setup)	7.5		ns
t_{HSPIDM}	SCK sampling edge to data input invalid	–1.5		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{SPISELx}$ low to first SCK edge (x=0 or 1)	$2t_{SCLK}-1.5$		ns
t_{SPICHM}	Serial clock high period	$2t_{SCLK}-1.5$		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 (x=0 or 1)	$2t_{SCLK}-1.5$		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

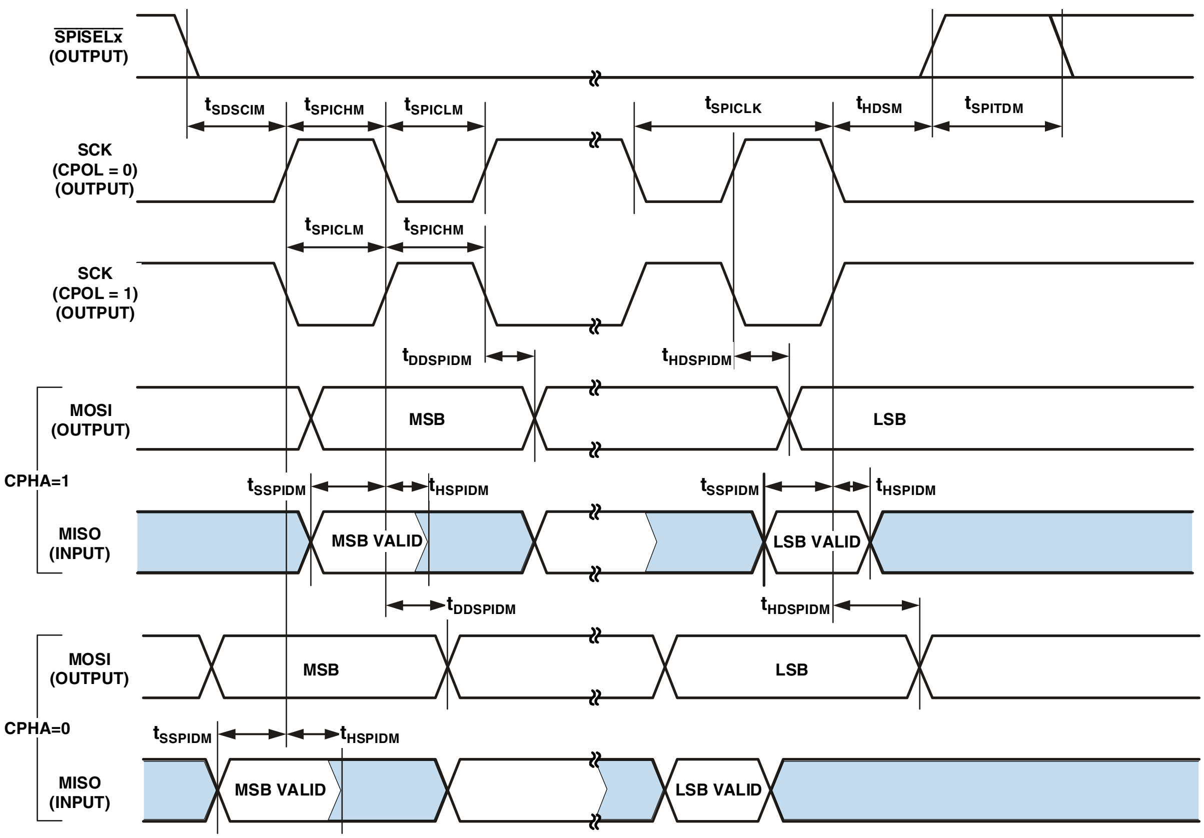


Figure 24. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 33 and Figure 25 describe SPI port slave operations.

Table 33. Serial Peripheral Interface (SPI) Port—Slave Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
t_{SPICHS}	Serial clock high period	$2t_{SCLK} - 1.5$		ns
t_{SPICLS}	Serial clock low period	$2t_{SCLK} - 1.5$		ns
t_{SPICLK}	Serial clock period	$4t_{SCLK} - 1.5$		ns
t_{HDS}	Last SCK edge to $\overline{SPISS}$ not asserted	$2t_{SCLK} - 1.5$		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{SCLK} - 1.5$		ns
t_{SDSCI}	$\overline{SPISS}$ assertion to first SCK edge	$2t_{SCLK} - 1.5$		ns
t_{SSPID}	Data input valid to SCK edge (data input setup)	1.6		ns
t_{HSPID}	SCK sampling edge to data input invalid	1.6		ns
Switching Characteristics				
t_{DSOE}	$\overline{SPISS}$ assertion to data out active	0	8	ns
t_{DSDHI}	$\overline{SPISS}$ deassertion to data high impedance	0	8	ns
t_{DDSPID}	SCK edge to data out valid (data out delay)	0	10	ns
t_{HDSPID}	SCK edge to data out invalid (data out hold)	0	10	ns

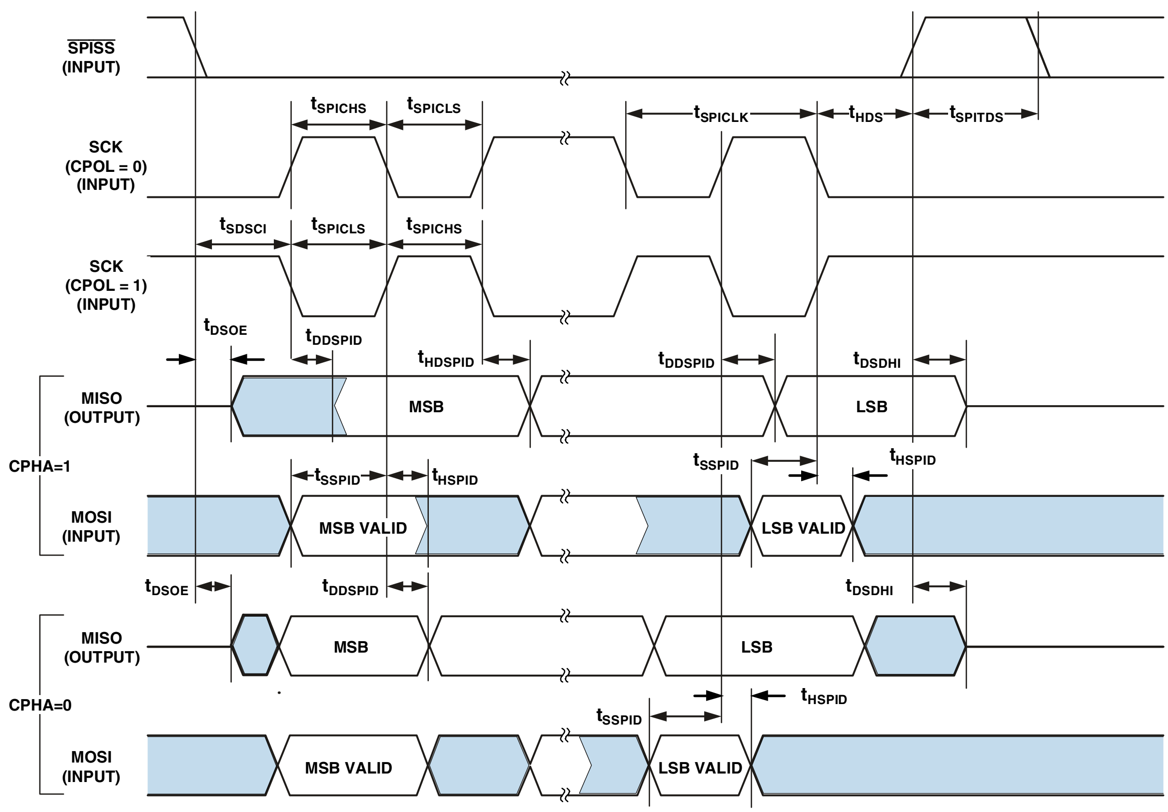


Figure 25. Serial Peripheral Interface (SPI) Port—Slave Timing

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

Figure 26 describes the UART ports receive and transmit operations. The maximum baud rate is SCLK/16. There is some latency between the generation of internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

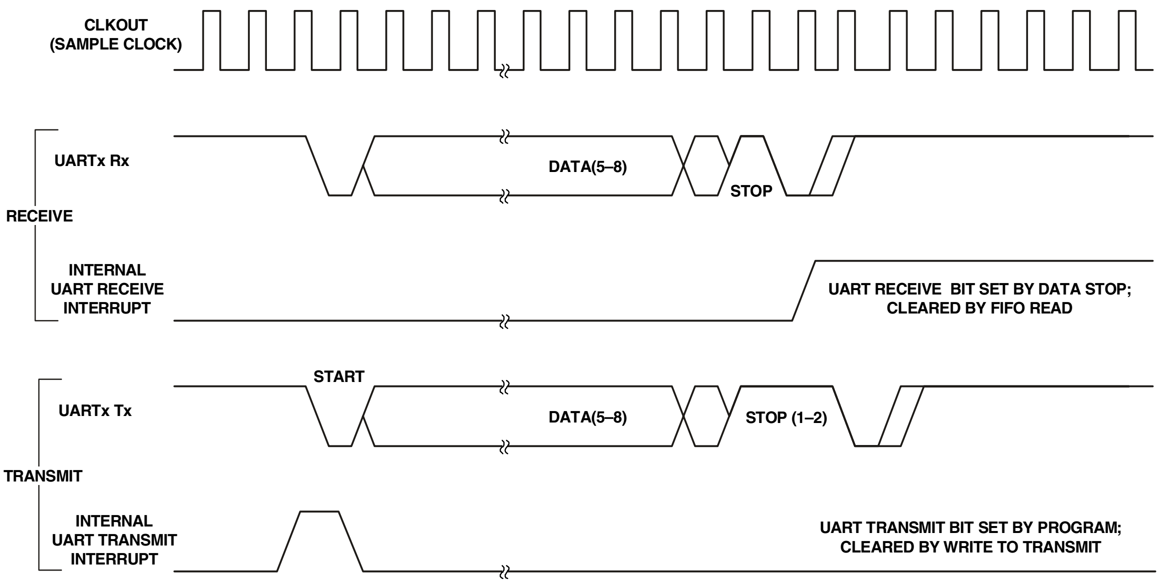


Figure 26. UART Ports—Receive and Transmit Timing

General-Purpose Port Timing

Table 34 and Figure 27 describe general-purpose port operations.

Table 34. General-Purpose Port Timing

Parameter		Minimum	Maximum	Unit
Timing Requirement				
t_{WFI}	General-Purpose Port Pin Input Pulse Width	t_{SCLK} + 1		ns
Switching Characteristic				
t_{GPOD}	General-Purpose Port Pin Output Delay from CLKOUT Low	0	6	ns

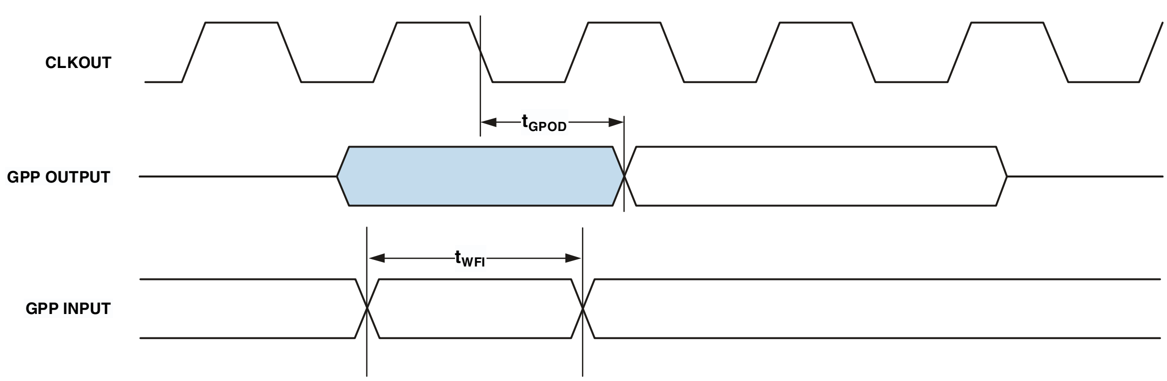


Figure 27. General-Purpose Port Timing

Timer Cycle Timing

Table 35 and Figure 28 describe timer expired operations. The input signal is asynchronous in "width capture mode" and "external clock mode" and has an absolute maximum input frequency of (f_{SCLK}/2) MHz.

Table 35. Timer Cycle Timing

Parameter		Minimum	Maximum	Unit
Timing Characteristics				
t_{WL}	Timer Pulse Width Input Low (Measured In SCLK Cycles)[1]	1t_{SCLK}		ns
t_{WH}	Timer Pulse Width Input High (Measured In SCLK Cycles)[1]	1t_{SCLK}		ns
t_{TIS}	Timer Input Setup Time Before CLKOUT Low[2]	5		ns
t_{TIH}	Timer Input Hold Time After CLKOUT Low[2]	–2		ns
Switching Characteristic				
t_{HTO}	Timer Pulse Width Output (Measured In SCLK Cycles)	1t_{SCLK}	$(2^{32}–1)t_{SCLK}$	ns
t_{TOD}	Timer Output Update Delay After CLKOUT High		6	ns

[1] The minimum pulse widths apply for TMRx signals in width capture and external clock modes. They also apply to the PF15 or PPI_CLK signals in PWM output mode.
[2] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize programmable flag inputs.

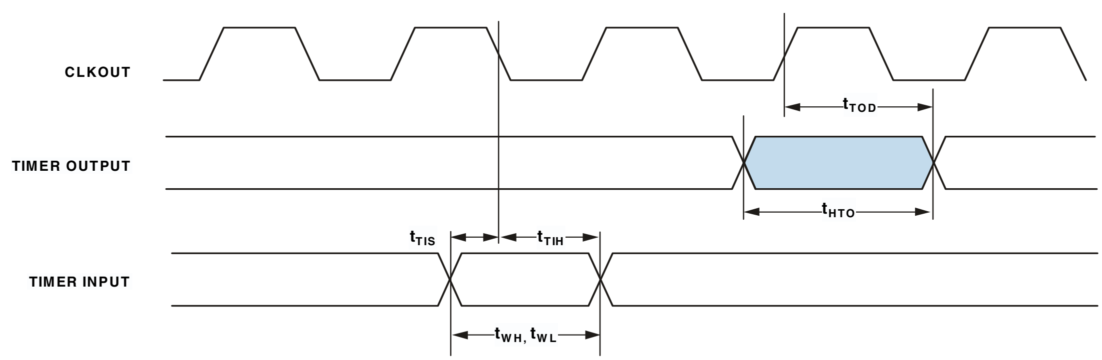


Figure 28. Timer Cycle Timing

ATA/ATAPI Controller Timing

Table 36. ATA/ATAPI Controller Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

ATA/ATAPI Controller Timing is TBD

Figure 29. ATA/ATAPI Controller Timing

Up/Down Counter/Rotary Encoder Timing

Table 37. Up/Down Counter/Rotary Encoder Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
t_{WCOUNT}	Up/Down Counter/Rotary Encoder Input Pulse Width	t_{SCLK} + 1		ns
Switching Characteristic				
TBD	TBD		TBD	ns

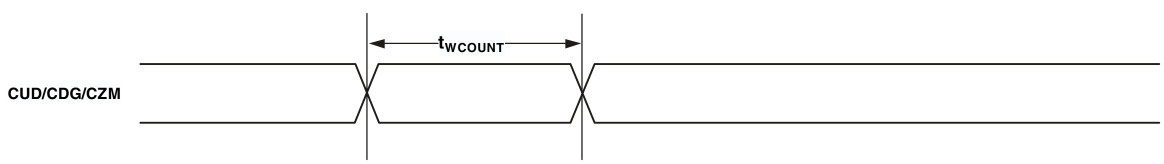


Figure 30. Up/Down Counter/Rotary Encoder Timing

SD/SDIO Controller Timing

Table 38. SD/SDIO Controller Timing

Parameter		Minimum	Maximum	Unit
Timing Requirements				
TBD	TBD	TBD		ns
Switching Characteristic				
TBD	TBD		TBD	ns

SD/SDIO Controller Timing is TBD

Figure 31. SD/SDIO Controller Timing

MXVR Timing

Table 39 and Table 40 describe the MXVR timing requirements.

Table 39. MXVR Timing—MXI Center Frequency Requirements

Parameter		Fs = 38 KHz	Fs = 44.1 KHz	Fs = 48 KHz	Unit
f_{MXI_256}	MXI Center Frequency (256Fs)	9.728	11.2896	12.288	MHz
f_{MXI_384}	MXI Center Frequency (384Fs)	14.592	16.9344	18.432	MHz
f_{MXI_512}	MXI Center Frequency (512Fs)	19.456	22.5792	24.576	MHz
f_{MXI_1024}	MXI Center Frequency (1024Fs)	38.912	45.1584	49.152	MHz

Table 40. MXVR Timing— MXI Clock Requirements

Parameter		Min	Max	Unit
Timing Requirements				
FS_{MXI}	MXI Clock Frequency Stability	–50	+50	ppm
FT_{MXI}	MXI Frequency Tolerance Over Temperature	–300	+300	ppm
DC_{MXI}	MXI Clock Duty Cycle	40	60	%

JTAG Test And Emulation Port Timing

Table 41 and Figure 32 describe JTAG port operations.

Table 41. JTAG Port Timing

Parameter		Minimum	Maximum	Unit
Timing Parameters				
t_{TCK}	TCK Period	20		ns
t_{STAP}	TDI, TMS Setup Before TCK High	4		ns
t_{HTAP}	TDI, TMS Hold After TCK High	4		ns
t_{SSYS}	System Inputs Setup Before TCK High[1]	4		ns
t_{HSYS}	System Inputs Hold After TCK High[1]	5		ns
t_{TRSTW}	$\overline{\text{TRST}}$ Pulsewidth[2] (measured in TCK cycles)	4		TCK
Switching Characteristics				
t_{DTDO}	TDO Delay from TCK Low		10	ns
t_{DSYS}	System Outputs Delay After TCK Low[3]	0	12	ns

[1] System Inputs=PA15–0, PB14–0, PC15–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ14–0, DQ15–0, DQS1–0, D15–0, ATAPI_PDIAG, CLKIN, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, BMODE3–0, MFS, MLF_P, and MLF_M.

[2] 50 MHz Maximum

[3] System Outputs=PA15–0, PB14–0, PC15–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ14–0, DQ15–0, DQS1–0, D15–0, DA12–0, DBA1–0, DQM1–0, DCLK2–1, $\overline{\text{DCLK2–1}}$, DCS1–0, $\overline{\text{DCKE}}$, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, $\overline{\text{DWE}}$, $\overline{\text{AMS3–0}}$, $\overline{\text{ABE1–0}}$, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{EMU}}$, CLKOUT, CLKBUF, EXT_WAKE.

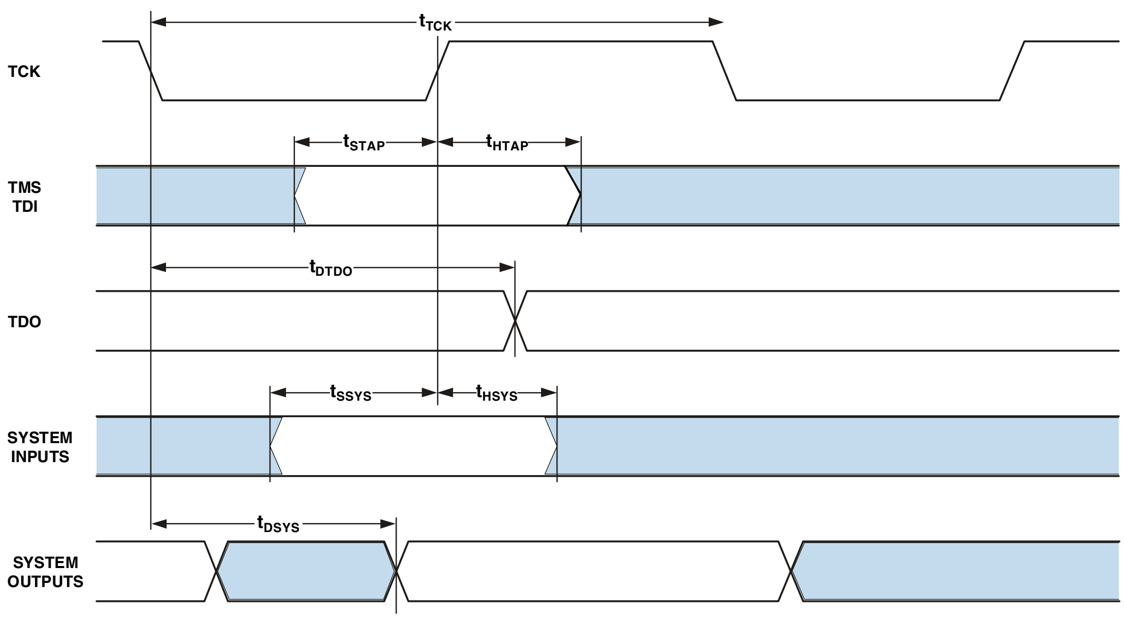


Figure 32. JTAG Port Timing

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 42 through Table 44 show the power dissipation for internal circuitry (V_{DDINT}).

See the *ADSP-BF549 Blackfin Processor Hardware Reference* for definitions of the various operating modes and for instructions on how to minimize system power.

Many operating conditions can affect power dissipation. System designers should refer to *EE-TBD: Estimating Power for ADSP-BF542/BF544/BF548/BF549 Blackfin Processors* on the Analog Devices website (www.analog.com)—use site search on "EE-TBD." This document provides detailed information for optimizing your design for lowest power.

Table 42. Internal Power Dissipation (Hibernate mode)

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

[1] Measured at V_{DDEXT} = 3.65 V with voltage regulator off (V_{DDINT} = 0 V).
[2] Measured at V_{DDRTC} = 3.3 V at 25°C.

Table 43. Internal Power Dissipation (Deep Sleep mode)

V_{DDINT}[1]	I_{DD} (nominal[2])	Unit
0.8	TBD	mA
0.9	TBD	mA
1.0	TBD	mA
1.1	TBD	mA
1.26	TBD	mA

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

Table 44. Internal Power Dissipation (Full On[1] mode)

V_{DDINT}[2] @ f_{CCLK}	I_{DD} (nominal[3])	Unit
0.8 @ TBD MHz	TBD	mA
0.8 @ TBD MHz	TBD	mA
0.9 @ TBD MHz	TBD	mA
1.0 @ TBD MHz	TBD	mA
1.1 @ TBD MHz	TBD	mA
1.26 @ TBD MHz	TBD	mA

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

TEST CONDITIONS

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

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 33). The time $t_{ENA\_MEASURED}$ is the interval from when the reference signal switches to when the output voltage reaches 2.0 V (output high) or 1.0 V (output low). Time t_{TRIP} is the interval from when the output starts driving to when the output reaches the 1.0 V or 2.0 V trip voltage. Time t_{ENA} is calculated as shown in the equation:

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

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

Output Disable Time

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

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

The output disable time t_{DIS} is the difference between $t_{DIS\_MEASURED}$ and t_{DECAY} as shown in Figure 33. The time $t_{DIS\_MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output high or output low voltage. The time t_{DECAY} is calculated with test loads C_L and I_L, and with ΔV equal to 0.5 V.

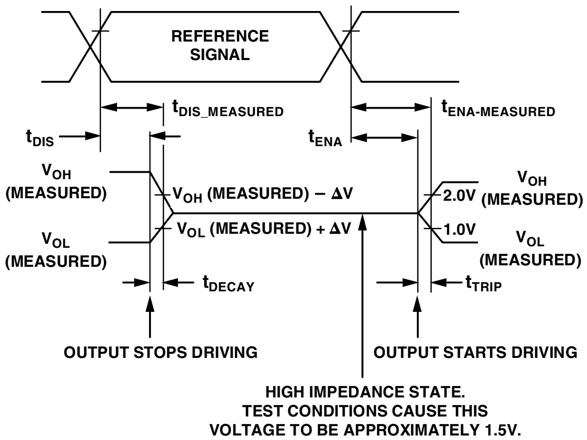


Figure 33. Output Enable/Disable

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-BF54x processor's output voltage and the input threshold for the device requiring the hold time. A typical ΔV will be 0.4 V. C_L is the total bus capacitance (per data line), and I_L is the total leakage or three-state current (per data line). The hold time will be t_{DECAY} plus the minimum disable time (for example, t_{DDAT} for an asynchronous memory write cycle).

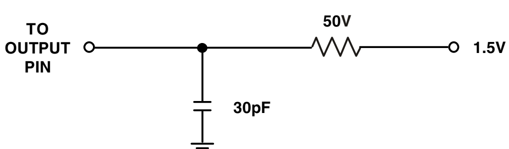


Figure 34. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

Figure 35. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

ENVIRONMENTAL CONDITIONS

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

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

where:

T_J = Junction temperature (°C)

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

Ψ_{JT} = From Table 45

P_D = Power dissipation (see Power Dissipation on Page 55 for the method to calculate P_D)

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

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

where:

T_A = Ambient temperature (°C)

Values of θ_{JC} are provided for package comparison and printed circuit board design considerations when an external heatsink is required.

Values of θ_{JB} are provided for package comparison and printed circuit board design considerations.

In Table 45, 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.

Table 45. Thermal Characteristics

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	18.4	°C/W
θ_{JA}	1 linear m/s air flow	15.8	°C/W
θ_{JA}	2 linear m/s air flow	15.0	°C/W
θ_{JB}		9.75	°C/W
θ_{JC}		6.37	°C/W
Ψ_{JT}	0 linear m/s air flow	0.27	°C/W
Ψ_{JT}	1 linear m/s air flow	0.60	°C/W
Ψ_{JT}	2 linear m/s air flow	0.66	°C/W

400-BALL BGA PINOUT

Table 46 lists the BGA pinout by signal. Table 47 on Page 60 lists the BGA pinout by ball number.

Table 46. 400-Ball BGA Ball Assignment (Alphabetically by Signal)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
$\overline{\text{ABE0}}$	C17	DA4	G16	DQS1	H18	GND	P13
$\overline{\text{ABE1}}$	C16	DA5	F19	$\overline{\text{DRAS}}$	E17	GND	V6
ADDR1	B2	DA6	D20	$\overline{\text{DWE}}$	E18	GND	G9
ADDR2	A2	DA7	C20	$\overline{\text{EMU}}$	R5	GND	G10
ADDR3	B3	DA8	F18	EXT_WAKE	M18	GND	G11
$\overline{\text{AMS0}}$	A10	DA9	E19	GND	A1	GND	L7
$\overline{\text{AMS1}}$	D9	DA10	B20	GND	A13	GND	L8
$\overline{\text{AMS2}}$	B10	DA11	F17	GND	A20	GND	L9
$\overline{\text{AMS3}}$	D10	DA12	D19	GND	E3	GND	L10
$\overline{\text{AOE}}$	C10	DBA0	H17	GND	J7	GND	L11
$\overline{\text{ARE}}$	B12	DBA1	H16	GND	J8	GND	L12
ATAPI_PDIAG	P19	$\overline{\text{DCAS}}$	F16	GND	J9	GND	L13
$\overline{\text{AWE}}$	D12	$\overline{\text{DCK1}}$	E16	GND	J10	GND	L14
BMODE0	W1	DCK1	D16	GND	J11	GND	R9
BMODE1	W2	DCK2	C18	GND	J12	GND	R13
BMODE2	W3	$\overline{\text{DCK2}}$	D18	GND	N6	GND	R14
BMODE3	W4	DCKE	B18	GND	N7	GND	R16
CLKBUF	D11	$\overline{\text{DCS0}}$	C19	GND	N8	GND	D1
CLKIN	A11	$\overline{\text{DCS1}}$	B19	GND	N9	GND	H7
CLKOUT	L16	DDR_V_{REF}	M20	GND	N10	GND	H8
D0	D13	DDR_V_{SSR}	N20	GND	N11	GND	H9
D1	C13	DQ0	L18	GND	N12	GND	H10
D2	B13	DQ1	M19	GND	N13	GND	H11
D3	B15	DQ2	L19	GND	N14	GND	H12
D4	A15	DQ3	L20	GND	U8	GND	M6
D5	B16	DQ4	L17	GND	B11	GND	M7
D6	A16	DQ5	K16	GND	F3	GND	M8
D7	B17	DQ6	K20	GND	F14	GND	M9
D8	C14	DQ7	K17	GND	K7	GND	M10
D9	C15	DQ8	K19	GND	K8	GND	M11
D10	A17	DQ9	J20	GND	K9	GND	M12
D11	D14	DQ10	K18	GND	K10	GND	M13
D12	D15	DQ11	H20	GND	K11	GND	M14
D13	E15	DQ12	J19	GND	K12	GND	Y1
D14	E14	DQ13	J18	GND	K13	GND	Y20
D15	D17	DQ14	J17	GND	P8	GND_{MC}	F6
DA0	G19	DQ15	J16	GND	P9	GND_{MP}	E7
DA1	G17	DQM0	G20	GND	P10	GND_{MX}	D4
DA2	E20	DQM1	H19	GND	P11	MFS	E6
DA3	G18	DQS0	F20	GND	P12	MLF_M	F4

Table 46. 400-Ball BGA Ball Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
MLF_P	E4	PC_5	G1	PE_15	W17	PH_7	H4
MXI	C2	PC_6	J5	PF_0	K3	PH_8	D5
MXO	C1	PC_7	H3	PF_1	J1	PH_9	C4
$\overline{\text{NMI}}$	C11	PC_8	Y14	PF_2	K2	PH_10	C7
PA_0	U12	PC_9	V13	PF_3	K1	PH_11	C5
PA_1	V12	PC_10	U13	PF_4	L2	PH_12	D7
PA_2	W12	PC_11	W14	PF_5	L1	PH_13	C6
PA_3	Y12	PC_12	Y15	PF_6	L4	PI_0	A3
PA_4	W11	PC_13	W15	PF_7	K4	PI_1	B4
PA_5	V11	PD_0	P3	PF_8	L3	PI_2	A4
PA_6	Y11	PD_1	P4	PF_9	M1	PI_3	B5
PA_7	U11	PD_2	R1	PF_10	M2	PI_4	A5
PA_8	U10	PD_3	R2	PF_11	M3	PI_5	B6
PA_9	Y10	PD_4	T1	PF_12	M4	PI_6	A6
PA_10	Y9	PD_5	R3	PF_13	N4	PI_7	B7
PA_11	V10	PD_6	T2	PF_14	N1	PI_8	A7
PA_12	Y8	PD_7	R4	PF_15	N2	PI_9	C8
PA_13	W10	PD_8	U1	PG_0	J4	PI_10	B8
PA_14	Y7	PD_9	U2	PG_1	K5	PI_11	A8
PA_15	W9	PD_10	T3	PG_2	L5	PI_12	A9
PB_0	W5	PD_11	V1	PG_3	N3	PI_13	C9
PB_1	Y2	PD_12	T4	PG_4	P1	PI_14	D8
PB_2	T6	PD_13	V2	PG_5	V15	PI_15	B9
PB_3	U6	PD_14	U4	PG_6	Y17	PJ_0	R20
PB_4	Y4	PD_15	U3	PG_7	W16	PJ_1	N18
PB_5	Y3	PE_0	V19	PG_8	V16	PJ_2	M16
PB_6	W6	PE_1	T17	PG_9	Y19	PJ_3	T20
PB_7	V7	PE_2	U18	PG_10	Y18	PJ_4	N17
PB_8	W8	PE_3	V14	PG_11	U15	PJ_5	U20
PB_9	V8	PE_4	Y16	PG_12	P16	PJ_6	P18
PB_10	U7	PE_5	W20	PG_13	R18	PJ_7	N16
PB_11	W7	PE_6	W19	PG_14	Y13	PJ_8	R19
PB_12	Y6	PE_7	R17	PG_15	W13	PJ_9	P17
PB_13	V9	PE_8	V20	PH_0	W18	PJ_10	T19
PB_14	Y5	PE_9	U19	PH_1	U14	PJ_11	M17
PC_0	H2	PE_10	T18	PH_2	V17	PJ_12	P20
PC_1	J3	PE_11	P2	PH_3	V18	PJ_13	N19
PC_2	J2	PE_12	M5	PH_4	U17	$\overline{\text{RESET}}$	C12
PC_3	H1	PE_13	P5	PH_5	C3	RTXI	A14
PC_4	G2	PE_14	U16	PH_6	D6	RTXO	B14

Table 46. 400-Ball BGA Ball Assignment (Alphabetically by Signal) (Continued)

Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
TCK	V3	V_{DDDDR}	G15	V_{DDEXT}	R7	V_{DDINT}	P6
TDI	V5	V_{DDDDR}	H13	V_{DDEXT}	R8	V_{DDINT}	P7
TDO	V4	V_{DDDDR}	H14	V_{DDEXT}	R15	V_{DDINT}	P14
TMS	U5	V_{DDDDR}	H15	V_{DDEXT}	H5	V_{DDINT}	G8
$\overline{TRST}$	T5	V_{DDEXT}	E9	V_{DDEXT}	H6	V_{DDINT}	G12
USB_DM	E2	V_{DDEXT}	E10	V_{DDEXT}	M15	V_{DDINT}	G13
USB_DP	E1	V_{DDEXT}	E11	V_{DDEXT}	T7	V_{DDINT}	L6
USB_ID	G3	V_{DDEXT}	E12	V_{DDEXT}	T8	V_{DDINT}	L15
USB_RSET	D3	V_{DDEXT}	N5	V_{DDEXT}	T9	V_{DDINT}	R10
USB_VBUS	D2	V_{DDEXT}	N15	V_{DDEXT}	T10	V_{DDINT}	R11
USB_V_{REF}	B1	V_{DDEXT}	F8	V_{DDEXT}	T11	V_{DDINT}	R12
USB_XI	F1	V_{DDEXT}	F13	V_{DDEXT}	T12	V_{DDMC}	F7
USB_XO	F2	V_{DDEXT}	F15	V_{DDEXT}	T13	V_{DDMP}	E8
V_{DDDDR}	J14	V_{DDEXT}	K6	V_{DDEXT}	T14	V_{DDMX}	E5
V_{DDDDR}	J15	V_{DDEXT}	P15	V_{DDEXT}	T15	V_{DDRTC}	E13
V_{DDDDR}	F10	V_{DDEXT}	G5	V_{DDEXT}	T16	V_{DDUSB}	F5
V_{DDDDR}	F11	V_{DDEXT}	G6	V_{DDINT}	J6	V_{DDUSB}	G4
V_{DDDDR}	F12	V_{DDEXT}	G7	V_{DDINT}	J13	V_{ROUT0}	A18
V_{DDDDR}	K14	V_{DDEXT}	G14	V_{DDINT}	U9	V_{ROUT1}	A19
V_{DDDDR}	K15	V_{DDEXT}	R6	V_{DDINT}	F9	XTAL	A12

Table 47 lists the BGA pinout by ball number. Table 46 on Page 57 lists the BGA pinout by signal.

Table 47. 400-Ball BGA Ball Assignment (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A1	GND	B1	USB_V$_{REF}$	C1	MXO	D1	GND
A2	ADDR2	B2	ADDR1	C2	MXI	D2	USB_VBUS
A3	PI_0	B3	ADDR3	C3	PH_5	D3	USB_RSET
A4	PI_2	B4	PI_1	C4	PH_9	D4	GND$_{MX}$
A5	PI_4	B5	PI_3	C5	PH_11	D5	PH_8
A6	PI_6	B6	PI_5	C6	PH_13	D6	PH_6
A7	PI_8	B7	PI_7	C7	PH_10	D7	PH_12
A8	PI_11	B8	PI_10	C8	PI_9	D8	PI_14
A9	PI_12	B9	PI_15	C9	PI_13	D9	$\overline{AMS1}$
A10	$\overline{AMS0}$	B10	$\overline{AMS2}$	C10	$\overline{AOE}$	D10	$\overline{AMS3}$
A11	CLKIN	B11	GND	C11	$\overline{NMI}$	D11	CLKBUF
A12	XTAL	B12	$\overline{ARE}$	C12	$\overline{RESET}$	D12	$\overline{AWE}$
A13	GND	B13	D2	C13	D1	D13	D0
A14	RTXI	B14	RTXO	C14	D8	D14	D11
A15	D4	B15	D3	C15	D9	D15	D12
A16	D6	B16	D5	C16	$\overline{ABE1}$	D16	DCK1
A17	D10	B17	D7	C17	$\overline{ABE0}$	D17	D15
A18	V_{ROUT0}	B18	DCKE	C18	DCK2	D18	$\overline{DCK2}$
A19	V_{ROUT1}	B19	$\overline{DCS1}$	C19	$\overline{DCS0}$	D19	DA12
A20	GND	B20	DA10	C20	DA7	D20	DA6
E1	USB_DP	F1	USB_XI	G1	PC_5	H1	PC_3
E2	USB_DM	F2	USB_XO	G2	PC_4	H2	PC_0
E3	GND	F3	GND	G3	USB_ID	H3	PC_7
E4	MLF_P	F4	MLF_M	G4	V_{DDUSB}	H4	PH_7
E5	V_{DDMX}	F5	V_{DDUSB}	G5	V_{DDEXT}	H5	V_{DDEXT}
E6	MFS	F6	GND$_{MC}$	G6	V_{DDEXT}	H6	V_{DDEXT}
E7	GND$_{MP}$	F7	V_{DDMC}	G7	V_{DDEXT}	H7	GND
E8	V_{DDMP}	F8	V_{DDEXT}	G8	V_{DDINT}	H8	GND
E9	V_{DDEXT}	F9	V_{DDINT}	G9	GND	H9	GND
E10	V_{DDEXT}	F10	V_{DDDDR}	G10	GND	H10	GND
E11	V_{DDEXT}	F11	V_{DDDDR}	G11	GND	H11	GND
E12	V_{DDEXT}	F12	V_{DDDDR}	G12	V_{DDINT}	H12	GND
E13	V_{DDRTC}	F13	V_{DDEXT}	G13	V_{DDINT}	H13	V_{DDDDR}
E14	D14	F14	GND	G14	V_{DDEXT}	H14	V_{DDDDR}
E15	D13	F15	V_{DDEXT}	G15	V_{DDDDR}	H15	V_{DDDDR}
E16	$\overline{DCK1}$	F16	$\overline{DCAS}$	G16	DA4	H16	DBA1
E17	$\overline{DRAS}$	F17	DA11	G17	DA1	H17	DBA0
E18	$\overline{DWE}$	F18	DA8	G18	DA3	H18	DQS1
E19	DA9	F19	DA5	G19	DA0	H19	DQM1
E20	DA2	F20	DQS0	G20	DQM0	H20	DQ11

Table 47. 400-Ball BGA Ball Assignment (Numerically by Ball Number) (Continued)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
J1	PF_1	K1	PF_3	L1	PF_5	M1	PF_9
J2	PC_2	K2	PF_2	L2	PF_4	M2	PF_10
J3	PC_1	K3	PF_0	L3	PF_8	M3	PF_11
J4	PG_0	K4	PF_7	L4	PF_6	M4	PF_12
J5	PC_6	K5	PG_1	L5	PG_2	M5	PE_12
J6	V_{DDINT}	K6	V_{DDEXT}	L6	V_{DDINT}	M6	GND
J7	GND	K7	GND	L7	GND	M7	GND
J8	GND	K8	GND	L8	GND	M8	GND
J9	GND	K9	GND	L9	GND	M9	GND
J10	GND	K10	GND	L10	GND	M10	GND
J11	GND	K11	GND	L11	GND	M11	GND
J12	GND	K12	GND	L12	GND	M12	GND
J13	V_{DDINT}	K13	GND	L13	GND	M13	GND
J14	V_{DDDDR}	K14	V_{DDDDR}	L14	GND	M14	GND
J15	V_{DDDDR}	K15	V_{DDDDR}	L15	V_{DDINT}	M15	V_{DDEXT}
J16	DQ15	K16	DQ5	L16	CLKOUT	M16	PJ_2
J17	DQ14	K17	DQ7	L17	DQ4	M17	PJ_11
J18	DQ13	K18	DQ10	L18	DQ0	M18	EXT_WAKE
J19	DQ12	K19	DQ8	L19	DQ2	M19	DQ1
J20	DQ9	K20	DQ6	L20	DQ3	M20	DDR_V_{REF}
N1	PF_14	P1	PG_4	R1	PD_2	T1	PD_4
N2	PF_15	P2	PE_11	R2	PD_3	T2	PD_6
N3	PG_3	P3	PD_0	R3	PD_5	T3	PD_10
N4	PF_13	P4	PD_1	R4	PD_7	T4	PD_12
N5	V_{DDEXT}	P5	PE_13	R5	$\overline{\text{EMU}}$	T5	$\overline{\text{TRST}}$
N6	GND	P6	V_{DDINT}	R6	V_{DDEXT}	T6	PB_2
N7	GND	P7	V_{DDINT}	R7	V_{DDEXT}	T7	V_{DDEXT}
N8	GND	P8	GND	R8	V_{DDEXT}	T8	V_{DDEXT}
N9	GND	P9	GND	R9	GND	T9	V_{DDEXT}
N10	GND	P10	GND	R10	V_{DDINT}	T10	V_{DDEXT}
N11	GND	P11	GND	R11	V_{DDINT}	T11	V_{DDEXT}
N12	GND	P12	GND	R12	V_{DDINT}	T12	V_{DDEXT}
N13	GND	P13	GND	R13	GND	T13	V_{DDEXT}
N14	GND	P14	V_{DDINT}	R14	GND	T14	V_{DDEXT}
N15	V_{DDEXT}	P15	V_{DDEXT}	R15	V_{DDEXT}	T15	V_{DDEXT}
N16	PJ_7	P16	PG_12	R16	GND	T16	V_{DDEXT}
N17	PJ_4	P17	PJ_9	R17	PE_7	T17	PE_1
N18	PJ_1	P18	PJ_6	R18	PG_13	T18	PE_10
N19	PJ_13	P19	ATAPI_PDIAG	R19	PJ_8	T19	PJ_10
N20	DDR_V_{SSR}	P20	PJ_12	R20	PJ_0	T20	PJ_3

Table 47. 400-Ball BGA Ball Assignment (Numerically by Ball Number) (Continued)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
U1	PD_8	V1	PD_11	W1	BMODE0	Y1	GND
U2	PD_9	V2	PD_13	W2	BMODE1	Y2	PB_1
U3	PD_15	V3	TCK	W3	BMODE2	Y3	PB_5
U4	PD_14	V4	TDO	W4	BMODE3	Y4	PB_4
U5	TMS	V5	TDI	W5	PB_0	Y5	PB_14
U6	PB_3	V6	GND	W6	PB_6	Y6	PB_12
U7	PB_10	V7	PB_7	W7	PB_11	Y7	PA_14
U8	GND	V8	PB_9	W8	PB_8	Y8	PA_12
U9	V_{DDINT}	V9	PB_13	W9	PA_15	Y9	PA_10
U10	PA_8	V10	PA_11	W10	PA_13	Y10	PA_9
U11	PA_7	V11	PA_5	W11	PA_4	Y11	PA_6
U12	PA_0	V12	PA_1	W12	PA_2	Y12	PA_3
U13	PC_10	V13	PC_9	W13	PG_15	Y13	PG_14
U14	PH_1	V14	PE_3	W14	PC_11	Y14	PC_8
U15	PG_11	V15	PG_5	W15	PC_13	Y15	PC_12
U16	PE_14	V16	PG_8	W16	PG_7	Y16	PE_4
U17	PH_4	V17	PH_2	W17	PE_15	Y17	PG_6
U18	PE_2	V18	PH_3	W18	PH_0	Y18	PG_10
U19	PE_9	V19	PE_0	W19	PE_6	Y19	PG_9
U20	PJ_5	V20	PE_8	W20	PE_5	Y20	GND

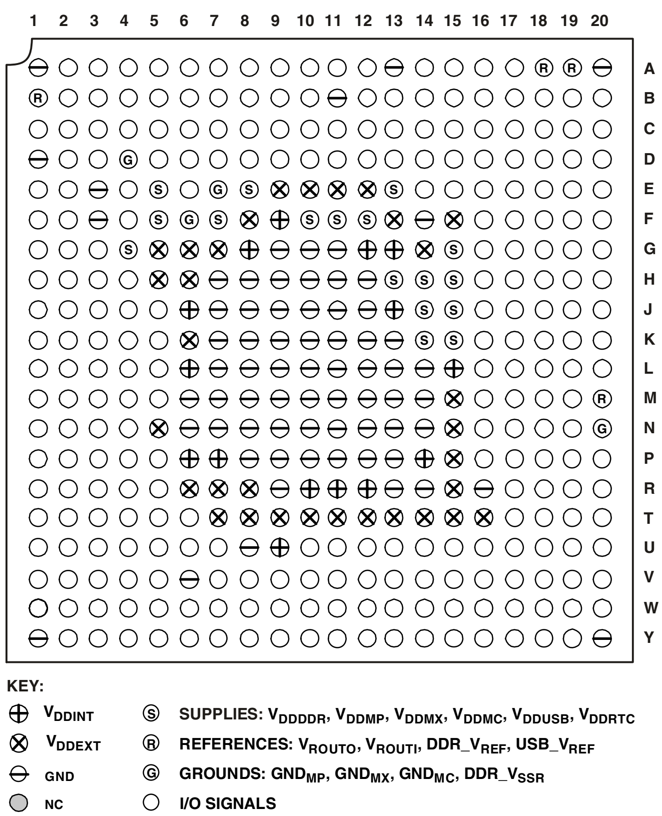


Figure 36. 400-Ball Mini-BGA Ground Configuration (Top View)

OUTLINE DIMENSIONS

Dimensions in Figure 37 are shown in millimeters.

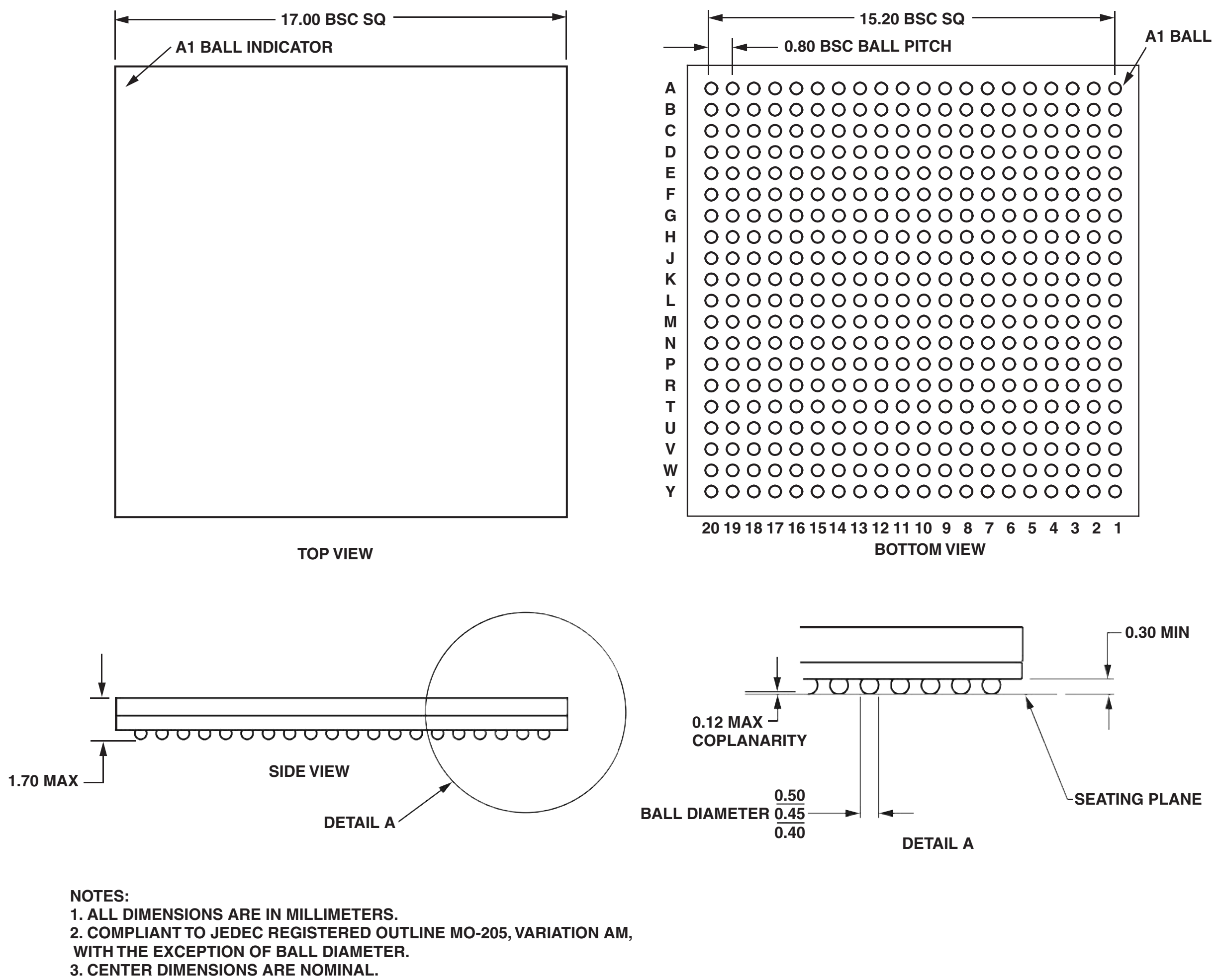


Figure 37. Chip Scale Package Ball Grid Array (Mini-BGA) BC-400

SURFACE MOUNT DESIGN

Table 48 is provided as an aid to PCB design. For industry-standard design recommendations, refer to IPC-7351, *Generic Requirements for Surface Mount Design and Land Pattern Standard.*

Table 48. BGA Data for Use with Surface Mount Design

Package	Ball Attach Type	Solder Mask Opening	Ball Pad Size
Chip Scale Package Ball Grid Array (Mini-BGA) BC-400	Solder Mask Defined	0.40 mm diameter	0.50 mm diameter

ORDERING GUIDE

Part numbers that include “Z” are lead free.

Part Number	Temperature Range (Ambient)	Speed Grade (Max)	Operating Voltage (Nominal)
ADSP-BF549-ENG	TBD[1]	TBD[1]	TBD[1]

[1] For more information, see component engineering-grade agreement.

PR06512-0-11/06(PrD)