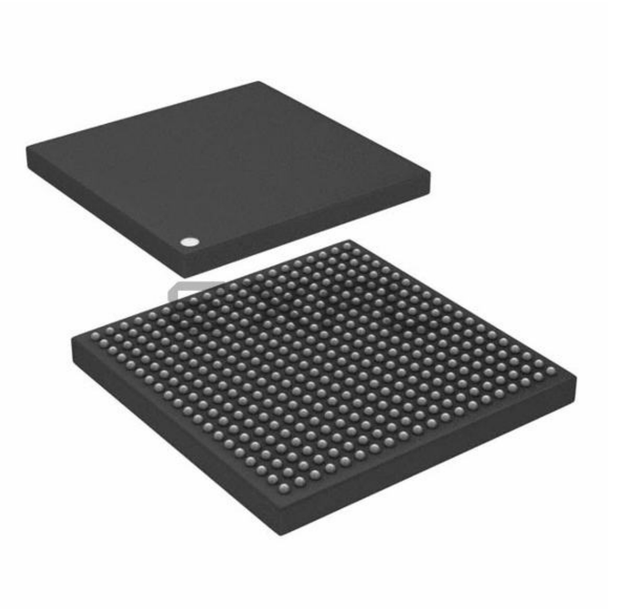
Analog Devices Inc. - ADSP-BF548BBCZ-5A Datasheet



Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Active
Type	Fixed Point
Interface	CAN, SPI, SSP, TWI, UART, USB
Clock Rate	533MHz
Non-Volatile Memory	External
On-Chip RAM	260kB
Voltage - I/O	2.50V, 3.30V
Voltage - Core	1.25V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	400-LFBGA, CSPBGA
Supplier Device Package	400-CSPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf548bbcz-5a

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.

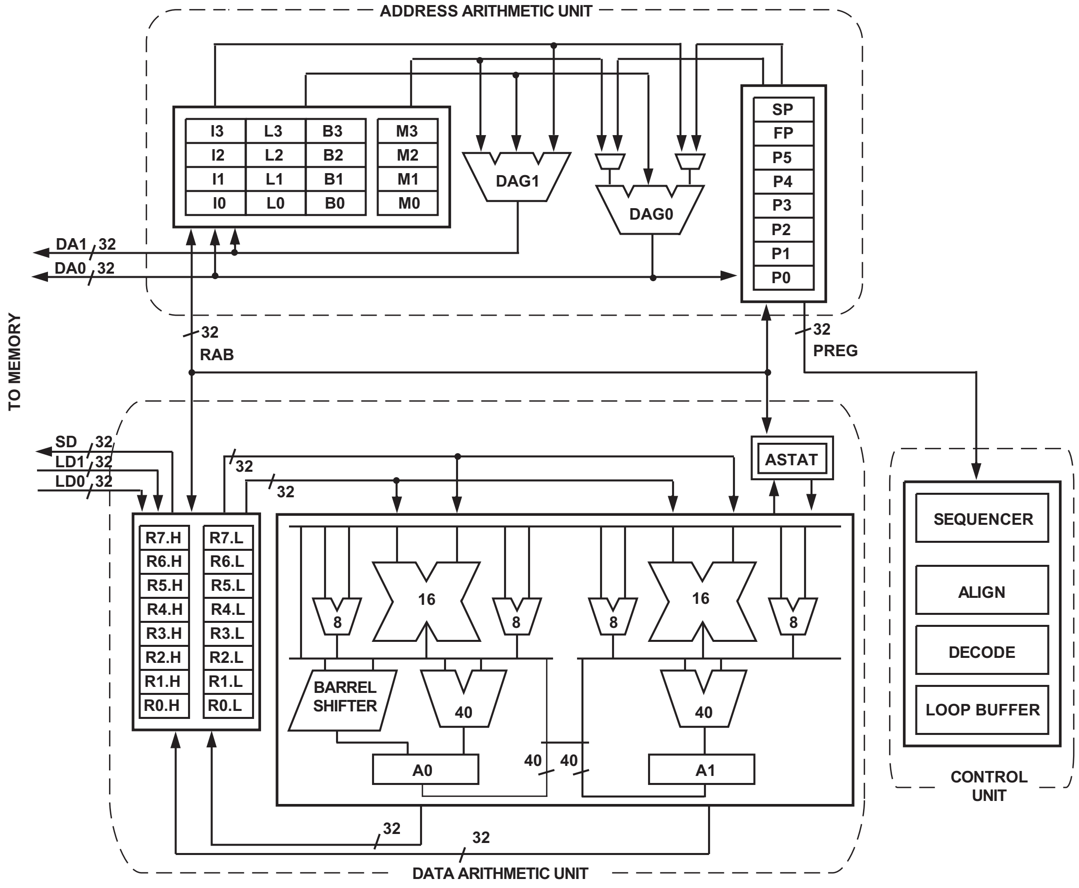


Figure 2. Blackfin Processor Core

The DDR memory controller can gluelessly manage up to two banks of double-rate synchronous dynamic memory (DDR and mobile DDR SDRAM). The 16-bit interface operates at the SCLK frequency, enabling a maximum throughput of 532M bytes/s. The DDR and mobile DDR controller is augmented with a queuing mechanism that performs efficient bursts into the DDR and mobile DDR. The controller is an industry standard DDR and mobile DDR SDRAM controller with each bank supporting from 64M bit to 512M bit device sizes and 4-, 8-, or 16-bit widths. The controller supports up to 256M bytes per external bank. With 2 external banks, the controller supports up to 512M bytes total. 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 64M byte 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. See the Ordering Guide on Page 101 for a list of specific products that provide support for DDR memory.

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

Customers should consult the Ordering Guide when selecting a specific ADSP-BF54x component for the intended application. Products that provide support for mobile DDR memory are noted in the ordering guide footnotes.

NAND Flash Controller (NFC)

The ADSP-BF54x Blackfin processors provide 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 bytes and 512 bytes. Larger page sizes can be supported in software.
- The ability to release external bus interface pins during long accesses.
- Support for internal bus requests of 16 bits or 32 bits.
- A DMA engine to transfer data between internal memory and a NAND flash device.

One-Time-Programmable Memory

The ADSP-BF54x Blackfin processors have 64K bits of one-time-programmable (OTP) non-volatile memory that can be programmed by the developer only one time. It includes the array and logic to support read access and programming. Additionally, its pages can be write protected.

OTP enables developers to store both public and private data on-chip. In addition to storing public and private key data for applications requiring security, it also allows developers to store completely user-definable data such as a customer ID, product ID, or a MAC address. By using this feature, generic parts can be shipped, which are then programmed and protected by the developer within this non-volatile memory. The OTP memory can be accessed through an API provided by the on-chip ROM.

I/O Memory Space

The ADSP-BF54x Blackfin processors do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses near the top of the 4G byte address space. These are separated into two smaller blocks, one containing the control MMRs for all core functions and the other containing 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 Blackfin processors contain a small on-chip boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF54x Blackfin processors are 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 Blackfin processors handles all asynchronous and synchronous events to the processors. The ADSP-BF54x Blackfin processors provide 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.

Host DMA Port Interface

The host DMA port (HOSTDP) facilitates a host device external to the ADSP-BF54x Blackfin processors 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 HOSTDP is enabled through the peripheral access bus. Once the port has been enabled, the transactions are controlled by the external host. The external host programs standard DMA configuration words in order to send/receive data to any valid internal or external memory location. The host DMA port controller includes the following features:

- Allows an external master to configure DMA read/write data transfers and read port status
- Uses a flexible asynchronous memory protocol for its external interface
- Allows an 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 Blackfin processors' 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 Blackfin 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.

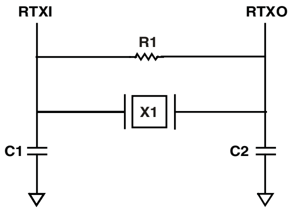
The 32.768 kHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60-second counter, a 60-minute counter, a 24-hour counter, and a 32,768-day counter.

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 processors from deep sleep mode, and it can wake up the on-chip internal voltage regulator from the hibernate state.

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 processors include a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system reliability 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, and 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 processors' 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 Blackfin processors. One unit provides eight general-purpose programmable timers, and the other unit provides three. 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 on the TMRx pins, an external clock TMRCLK input pin, or to the internal SCLK.

includes support for five to eight data bits, one or two 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 seven to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as

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

Where the 16-bit UART divisor comes from the UARTx_DLH register (most significant 8 bits) and UARTx_DLL register (least significant eight bits), and the EDBO is a bit in the UARTx_GCTL register.

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

UART1 and UART3 feature a pair of $\overline{\text{UARTxRTS}}$ (request to send) and $\overline{\text{UARTxCTS}}$ (clear to send) signals for hardware flow purposes. The transmitter hardware is automatically prevented from sending further data when the $\overline{\text{UARTxCTS}}$ input is deasserted. The receiver can automatically de-assert its $\overline{\text{UARTxRTS}}$ 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 Blackfin processors offer up to 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 Blackfin processors' 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 and 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 Blackfin processors' CAN module represents only the controller part of the interface. The controller interface supports connection to 3.3 V high speed, fault-tolerant, single-wire transceivers.

An additional crystal is not required to supply the CAN clock, as the CAN clock is derived from the processor system clock (SCLK) through a programmable divider.

TWI CONTROLLER INTERFACE

The ADSP-BF54x Blackfin processors include up to two 2-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 and support for both 7-bit addressing and multimedia data arbitration. Each TWI interface uses two pins for transferring clock (SCLx) and data (SDAx), 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 Blackfin processors' 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 Blackfin processors group the many peripheral signals to ten ports—referred to as Port A to Port J. Most ports contain 16 pins, though some have fewer. 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.

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-BF549 Blackfin 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-BF549 Blackfin processor from the hibernate state. These features allow the ADSP-BF549 processor 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 the external crystal or crystal oscillator can be 256 Fs, 384 Fs, 512 Fs, or 1024 Fs for Fs = 38 kHz, 44.1 kHz, or 48 kHz. If using a crystal to provide the MXVR clock, use a parallel-resonant, fundamental mode, microprocessor-grade crystal.

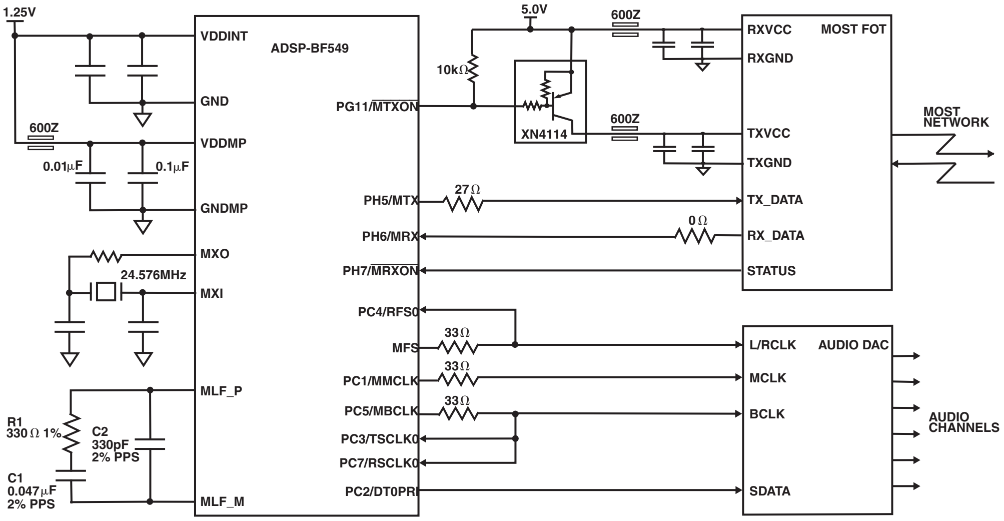


Figure 5. MXVR MOST Connection

DYNAMIC POWER MANAGEMENT

The ADSP-BF54x Blackfin processors provide 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 Blackfin processors' 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 the capability to run at the 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. DMA access is available to appropriately configured L1 memories.

In the active mode, it is possible to disable the control input to the PLL by setting the PLL_OFF bit in the PLL control register. This register can be accessed with a user-callable routine in the on-chip ROM called bfrom_SysControl(). For more information, see the "Dynamic Power Management" chapter in the *ADSP-BF54x Blackfin Processor Hardware Reference*. If disabled, the PLL must be re-enabled before transitioning to the full-on or sleep modes.

Table 4. Power Settings

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

For large page NAND flash devices, the 4-byte electronic signature is read in order to configure the kernel for booting. This allows support for multiple large page devices. The fourth byte of the electronic signature must comply with the specifications in Table 9.

Any configuration from Table 9 that also complies with the command set listed below is directly supported by the boot kernel. There are no restrictions on the page size or block size as imposed by the small-page boot kernel.

Table 9. Byte 4 Electronic Signature Specification

Page Size (excluding spare area)	D1:D0	00	1K bytes
		01	2K bytes
		10	4K bytes
		11	8K bytes
Spare Area Size	D2	0	8 bytes/512 bytes
		1	16 bytes/512 bytes
Block Size (excluding spare area)	D5:4	00	64K bytes
		01	128K bytes
		10	256K bytes
		11	512K bytes
Bus Width	D6	0	x8
		1	x16
Not Used for Configuration	D3, D7		

Large page devices must support the following command set:

Reset: 0xFF
Read Electronic Signature: 0x90
Read: 0x00, 0x30 (confirm command)

Large page devices must not support or react to NAND flash command 0x50. This is a small page NAND flash command used for device auto detection.

By default, the boot kernel will always issue five address cycles; therefore, if a large page device requires only four cycles, the device must be capable of ignoring the additional address cycle.

16-bit NAND flash memory devices must only support the issuing of command and address cycles via the lower eight bits of the data bus. Devices that use the full 16-bit bus for command and address cycles are not supported.

- Boot from OTP memory (BMODE = 0xB)—This provides a standalone booting method. The boot stream is loaded from on-chip OTP memory. By default, the boot stream is expected to start from OTP page 0x40 and can occupy all public OTP memory up to page 0xDF (2560 bytes). Since the start page is programmable, the maximum size of the boot stream can be extended to 3072 bytes.
- Boot from 16-bit host DMA (BMODE = 0xE)—In this mode, the host DMA port is configured in 16-bit acknowledge mode with little endian data format. Unlike other modes, the host is responsible for interpreting the boot stream. It writes data blocks individually into the host DMA port. Before configuring the DMA settings for each block, the host may either poll the ALLOW_CONFIG bit in HOST_STATUS or wait to be interrupted by the HWAIT signal. When using HWAIT, the host must still check ALLOW_CONFIG at least once before beginning to configure the host DMA port. After completing the configuration, the host is required to poll the READY bit in HOST_STATUS before beginning to transfer data. When the host sends an HIRQ control command, the boot kernel issues a CALL instruction to address 0xFFA0 0000. It is the host's responsibility to ensure valid code has been placed at this address. The routine at address 0xFFA0 0000 can be a simple initialization routine to configure internal resources, such as the SDRAM controller, which then returns using an RTS instruction. The routine may also be the final application, which will never return to the boot kernel.
- Boot from 8-bit host DMA (BMODE = 0xF)—In this mode, the host DMA port is configured in 8-bit interrupt mode with little endian data format. Unlike other modes, the host is responsible for interpreting the boot stream. It writes data blocks individually to the host DMA port. Before configuring the DMA settings for each block, the host may either poll the ALLOW_CONFIG bit in HOST_STATUS or wait to be interrupted by the HWAIT signal. When using HWAIT, the host must still check ALLOW_CONFIG at least once before beginning to configure the host DMA port. The host will receive an interrupt from the HOST_ACK signal every time it is allowed to send the next FIFO depth's worth (sixteen 32-bit words) of information. When the host sends an HIRQ control command, the boot kernel issues a CALL instruction to address 0xFFA0 0000. It is the host's responsibility to ensure valid code has been placed at this address. The routine at address 0xFFA0 0000 can be a simple initialization routine to configure internal resources, such as the SDRAM controller, which then returns using an RTS instruction. The routine may also be the final application, which will never return 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 disabled based on OTP programming. External hardware, especially booting hosts, may monitor the HWAIT signal to determine

Software Add-Ins for CrossCore Embedded Studio

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

Board Support Packages for Evaluation Hardware

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

Middleware Packages

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

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

Algorithmic Modules

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

Designing an Emulator-Compatible DSP Board (Target)

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

For details on target board design issues including mechanical layout, single processor connections, signal buffering, signal termination, and emulator pod logic, see the Engineer-to-Engineer Note *“Analog Devices JTAG Emulation Technical Reference”* (EE-68) on the Analog Devices website (www.analog.com)—use site search on “EE-68.” This document is updated regularly to keep pace with improvements to emulator support.

MXVR BOARD LAYOUT GUIDELINES

The MXVR Loop Filter RC network is connected between the MLF_P and MLF_M pins in the following manner:

Capacitors:

- C1: 0.047 µF (PPS type, 2% tolerance recommended)
- C2: 330 pF (PPS type, 2% tolerance recommended)

Resistor:

- R1: 330 Ω (1% tolerance)

The RC network should be located physically close to the MLF_P and MLF_M pins on the board.

The RC network should be shielded using GND_{MP} traces.

Avoid routing other switching signals near the RC network to avoid crosstalk.

MXI driven with external clock oscillator IC:

- MXI should be driven with the clock output of a clock oscillator IC running at a frequency of 49.152 MHz or 45.1584 MHz.
- MXO should be left unconnected.
- Avoid routing other switching signals near the oscillator and clock output trace to avoid crosstalk. When not possible, shield traces with ground.

MXI/MXO with external crystal:

- The crystal must be a fundamental mode crystal running at a frequency of 49.152 MHz or 45.1584 MHz.
- The crystal and load capacitors should be placed physically close to the MXI and MXO pins on the board.
- Board trace capacitance on each lead should not be more than 3 pF.
- Trace capacitance plus load capacitance should equal the load capacitance specification for the crystal.
- Avoid routing other switching signals near the crystal and components to avoid crosstalk. When not possible, shield traces and components with ground.

V_{DDMP}/GND_{MP}—MXVR PLL power domain:

- Route V_{DDMP} and GND_{MP} with wide traces or as isolated power planes.
- Drive V_{DDMP} to same level as V_{DDINT}.
- Place a ferrite bead between the V_{DDINT} power plane and the V_{DDMP} pin for noise isolation.
- Locally bypass V_{DDMP} with 0.1 µF and 0.01 µF decoupling capacitors to GND_{MP}.
- Avoid routing switching signals near to V_{DDMP} and GND_{MP} traces to avoid crosstalk.

Table 11. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)	Driver Type[2]
Port E: GPIO/*SPI0/UART0-1/PPI1/TWI0/Keypad*			
PE0/*SPI0SCK/KEY_COL7*[3]	I/O	GPIO/*SPI0 Clock/Keypad Column Output*	A
PE1/*SPI0MISO/KEY_ROW6*[3]	I/O	GPIO/*SPI0 Master In Slave Out/Keypad Row Input*	C
PE2/*SPI0MOSI/KEY_COL6*	I/O	GPIO/*SPI0 Master Out Slave In/Keypad Column Output*	C
PE3/$\overline{\text{SPI0SS}}$/*KEY_ROW5*	I/O	GPIO/*SPI0 Slave Select Input/Keypad Row Input*	A
PE4/$\overline{\text{SPI0SEL1}}$/*KEY_COL*[3]	I/O	GPIO/*SPI0 Slave Select Enable 1/Keypad Column Output*	A
PE5/$\overline{\text{SPI0SEL2}}$/*KEY_ROW4*	I/O	GPIO/*SPI0 Slave Select Enable 2/Keypad Row Input*	A
PE6/$\overline{\text{SPI0SEL3}}$/*KEY_COL4*	I/O	GPIO/*SPI0 Slave Select Enable 3/Keypad Column Output*	A
PE7/*UART0TX/KEY_ROW7*	I/O	GPIO/*UART0 Transmit/Keypad Row Input*	A
PE8/*UART0RX/TACI0*	I/O	GPIO/*UART0 Receive/Alternate Capture Input 0*	A
PE9/$\overline{\text{UART1RTS}}$	I/O	GPIO/*UART1 Request to Send*	A
PE10/$\overline{\text{UART1CTS}}$	I/O	GPIO/*UART1 Clear to Send*	A
PE11/*PPI1_CLK*	I/O	GPIO/*PPI1 Clock*	A
PE12/*PPI1_FS1*	I/O	GPIO/*PPI1 Frame Sync 1*	A
PE13/*PPI1_FS2*	I/O	GPIO/*PPI1 Frame Sync 2*	A
PE14/*SCL0*	I/O	GPIO/*TWI0 Serial Clock* (Open-drain output: requires a pull-up resistor.)	E
PE15/*SDA0*	I/O	GPIO/*TWI0 Serial Data* (Open-drain output: requires a pull-up resistor.)	E
Port F: *GPIO/PPI0/Alternate ATAPI Data*			
PF0/*PPI0_D0/ATAPI_D0A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF1/*PPI0_D1/ATAPI_D1A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF2/*PPI0_D2/ATAPI_D2A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF3/*PPI0_D3/ATAPI_D3A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF4/*PPI0_D4/ATAPI_D4A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF5/*PPI0_D5/ATAPI_D5A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF6/*PPI0_D6/ATAPI_D6A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF7/*PPI0_D7/ATAPI_D7A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF8/*PPI0_D8/ATAPI_D8A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF9/*PPI0_D9/ATAPI_D9A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF10/*PPI0_D10/ATAPI_D10A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF11/*PPI0_D11/ATAPI_D11A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF12/*PPI0_D12/ATAPI_D12A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF13/*PPI0_D13/ATAPI_D13A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF14/*PPI0_D14/ATAPI_D14A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A
PF15/*PPI0_D15/ATAPI_D15A*	I/O	GPIO/*PPI0 Data/Alternate ATAPI Data*	A

Table 11. Pin Descriptions (Continued)

Pin Name	I/O[1]	Function (First/Second/Third/Fourth)	Driver Type[2]
USB_VBUS[11]	I/O	USB VBUS Pin (Pull high or low when unused.)	
USB_VREF	A	USB Voltage Reference (Connect to GND through a 0.1 µF capacitor or leave unconnected when not used.)	
USB_RSET	A	USB Resistance Set (Connect to GND through an unpopulated resistor pad.)	
MXVR (MOST) Interface			
MFS	O	MXVR Frame Sync (Leave unconnected when unused.)	C
MLF_P	A	MXVR Loop Filter Plus (Leave unconnected when unused.)	
MLF_M	A	MXVR Loop Filter Minus (Leave unconnected when unused.)	
MXI	C	MXVR Crystal Input (Pull high or low when unused.)	
MXO	C	MXVR Crystal Output (Pull high or low when unused.)	
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	C
$\overline{\text{TRST}}$	I	JTAG Reset (Pull low when unused.)	
TMS	I	JTAG Mode Select	
TCK	I	JTAG Clock	
$\overline{\text{EMU}}$	O	Emulation Output	C
Voltage Regulator			
$VR_{OUT}0$, $VR_{OUT}1$	O	External FET/BJT Drivers (Always connect together to reduce signal impedance.)	
Real Time Clock			
RTXO	C	RTC Crystal Output (Leave unconnected when unused. Does not three-state during hibernate.)	
RTXI	C	RTC Crystal Input (Pull high or low when unused.)	
Clock (PLL) Pins			
CLKIN	C	Clock/Crystal Input	
CLKOUT	O	Clock Output	B
XTAL	C	Crystal Output (If CLKBUF is enabled, does not three-state during hibernate.)	
CLKBUF	O	Buffered Oscillator Output (If enabled, does not three-state during hibernate.)	C
EXT_WAKE	O	External Wakeup from Hibernate Output (Does not three-state during hibernate.)	A
$\overline{\text{RESET}}$	I	Reset	
$\overline{\text{NMI}}$	I	Non-maskable Interrupt (Pull high when unused.)	

Asynchronous Memory Write Cycle Timing

Table 29 and Table 30 on Page 47 and Figure 15 and Figure 16 on Page 47 describe asynchronous memory write cycle operations for synchronous and for asynchronous ARDY.

Table 29. Asynchronous Memory Write Cycle Timing with Synchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{SARDY}	ARDY Setup Before the Falling Edge of CLKOUT	5.0		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	0.0		ns
t_{DO}	Output Delay After CLKOUT[1]		6.0	ns
t_{HO}	Output Hold After CLKOUT[1]	0.3		ns

[1] Output pins include $\overline{\text{AMS3–0}}$, $\overline{\text{ABE1–0}}$, ADDR19–1, and $\overline{\text{AWE}}$.

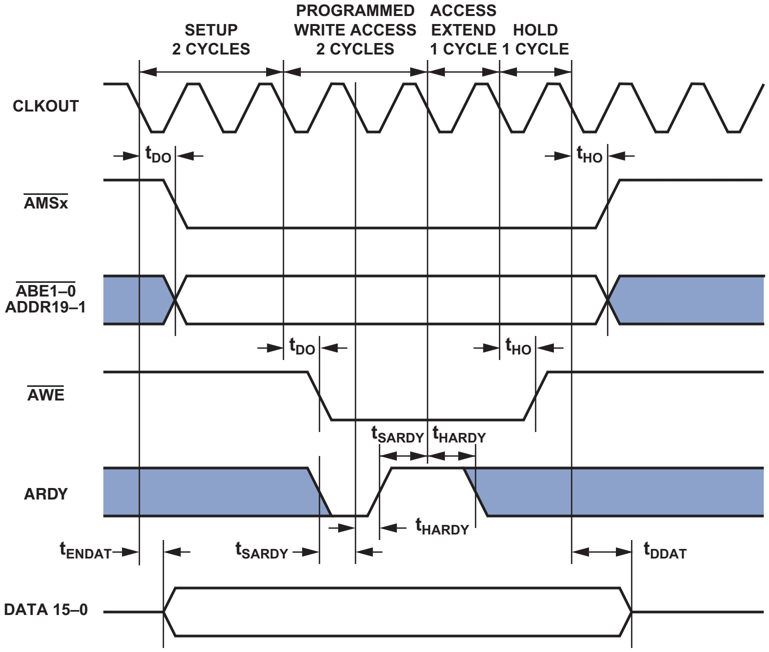


Figure 15. Asynchronous Memory Write Cycle Timing with Synchronous ARDY

Table 30. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY

Parameter		Min	Max	Unit
Timing Requirements				
t_{DANW}	ARDY Negated Delay from $\overline{\text{AMSx}}$ Asserted[1]		$(S + WA – 2) \times t_{SCLK}$	ns
t_{HAA}	ARDY Asserted Hold After $\overline{\text{AWE}}$ Negated	0.0		ns
Switching Characteristics				
t_{DDAT}	DATA15–0 Disable After CLKOUT		6.0	ns
t_{ENDAT}	DATA15–0 Enable After CLKOUT	0.0		ns
t_{DO}	Output Delay After CLKOUT[2]		6.0	ns
t_{HO}	Output Hold After CLKOUT[2]	0.3		ns

[1] S = number of programmed setup cycles, WA = number of programmed write access cycles.

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

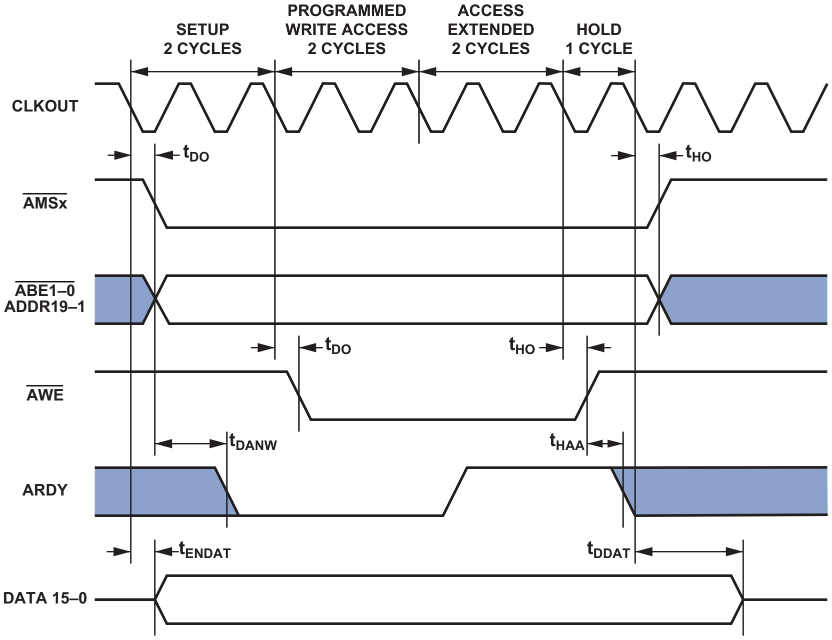


Figure 16. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY

Table 35. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{\text{BR}}$

Parameter		Min	Max	Unit
Timing Requirement				
t_{WBR}	$\overline{\text{BR}}$ Pulsewidth	2 × t_{SCLK}		ns
Switching Characteristics				
t_{SD}	CLKOUT Low to $\overline{\text{AMSx}}$, Address, and $\overline{\text{ARE}}$/$\overline{\text{AWE}}$ Disable		5.0	ns
t_{SE}	CLKOUT Low to $\overline{\text{AMSx}}$, Address, and $\overline{\text{ARE}}$/$\overline{\text{AWE}}$ Enable		5.0	ns
t_{DBG}	CLKOUT Low to $\overline{\text{BG}}$ Asserted Output Delay		4.0	ns
t_{EBG}	CLKOUT Low to $\overline{\text{BG}}$ Deasserted Output Hold		4.0	ns
t_{DBH}	CLKOUT Low to $\overline{\text{BGH}}$ Asserted Output Delay		3.6	ns
t_{EBH}	CLKOUT Low to $\overline{\text{BGH}}$ Deasserted Output Hold		3.6	ns

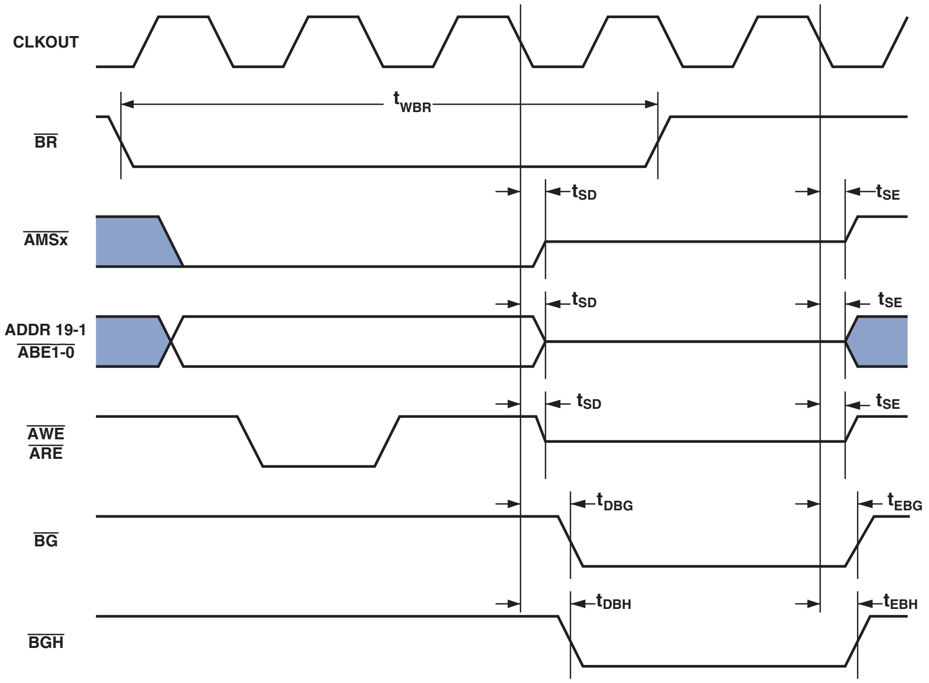


Figure 22. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{\text{BR}}$

Enhanced Parallel Peripheral Interface Timing

Table 39 and Figure 32 on Page 60, Figure 30 on Page 59, Figure 33 on Page 60, and Figure 31 on Page 59 describe enhanced parallel peripheral interface timing operations.

Table 39. Enhanced Parallel Peripheral Interface Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{PCLKW}	PPIx_CLK Width	6.0		ns
t_{PCLK}	PPIx_CLK Period	13.3		ns
Timing Requirements—GP Input and Frame Capture Modes				
t_{SFSPE}	External Frame Sync Setup Before PPIx_CLK	0.9		ns
t_{HFSPE}	External Frame Sync Hold After PPIx_CLK	1.9		ns
t_{SDRPE}	Receive Data Setup Before PPIx_CLK	1.6		ns
t_{HDRPE}	Receive Data Hold After PPIx_CLK	1.5		ns
Switching Characteristics—GP Output and Frame Capture Modes				
t_{DFSPE}	Internal Frame Sync Delay After PPIx_CLK		10.5	ns
t_{HOFSPE}	Internal Frame Sync Hold After PPIx_CLK	2.4		ns
t_{DDTPE}	Transmit Data Delay After PPIx_CLK		9.9	ns
t_{HDTPE}	Transmit Data Hold After PPIx_CLK	2.4		ns

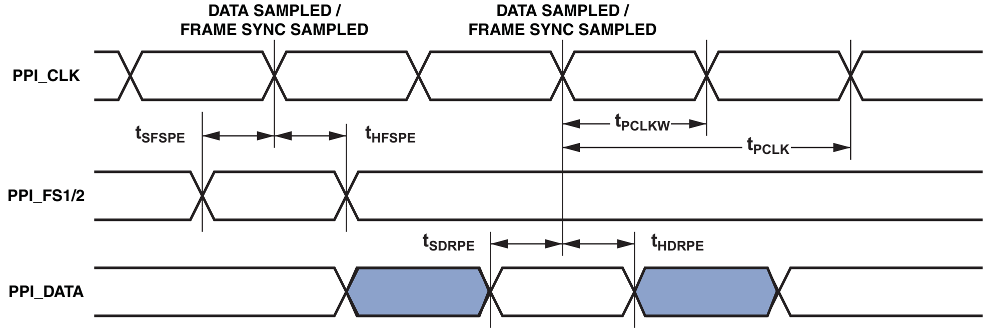


Figure 30. EPPI GP Rx Mode with External Frame Sync Timing

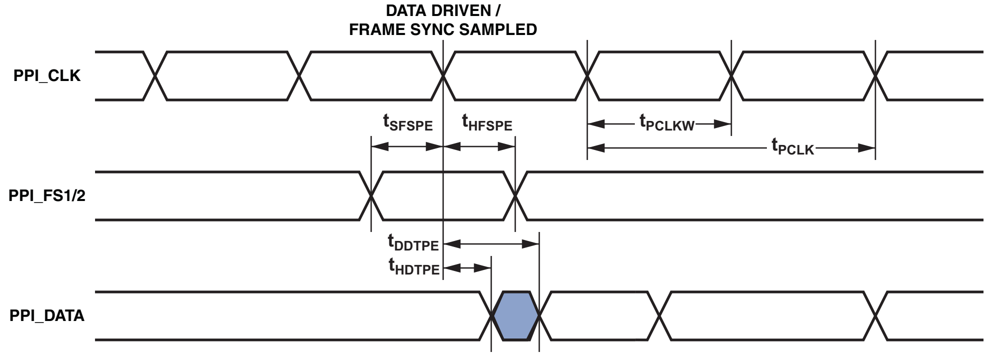


Figure 31. EPPI GP Tx Mode with External Frame Sync Timing

Serial Peripheral Interface (SPI) Port—Master Timing

Table 44 and Figure 38 describe SPI port master operations.

Table 44. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSPIDM}	Data Input Valid to SPIxSCK Edge (Data Input Setup)	9.0		ns
t_{HSPIDM}	SPIxSCK Sampling Edge to Data Input Invalid	–1.5		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIxSELy}}$ Low to First SPIxSCK Edge	$2t_{SCLK}-1.5$		ns
t_{SPICHM}	SPIxSCK High Period	$2t_{SCLK}-1.5$		ns
t_{SPICLM}	SPIxSCK Low Period	$2t_{SCLK}-1.5$		ns
t_{SPICLK}	SPIxSCK Period	$4t_{SCLK}-1.5$		ns
t_{HDSM}	Last SPIxSCK Edge to $\overline{\text{SPIxSELy}}$ High	$2t_{SCLK}-1.5$		ns
t_{SPITDM}	Sequential Transfer Delay	$2t_{SCLK}-1.5$		ns
$t_{DDSPIDM}$	SPIxSCK Edge to Data Out Valid (Data Out Delay)		6	ns
$t_{HDSPIDM}$	SPIxSCK Edge to Data Out Invalid (Data Out Hold)	–1.0		ns

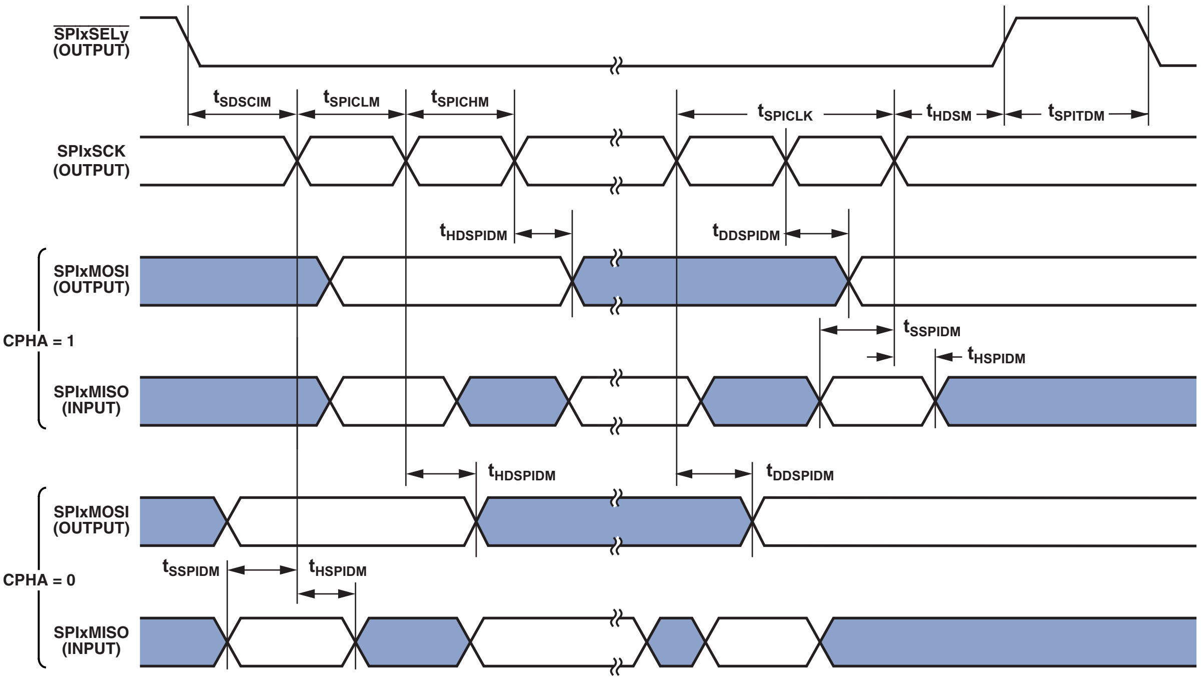


Figure 38. Serial Peripheral Interface (SPI) Port—Master Timing

Up/Down Counter/Rotary Encoder Timing

Table 49 and Figure 43 describe up/down counter/rotary encoder timing.

Table 49. Up/Down Counter/Rotary Encoder Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{WCOUNT}	CUD/CDG/CZM Input Pulse Width	t_{SCLK} + 1		ns
t_{CIS}	CUD/CDG/CZM Input Setup Time Before CLKOUT High[1]	7.2		ns
t_{CIH}	CUD/CDG/CZM Input Hold Time After CLKOUT High[1]	0.0		ns

[1] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize counter inputs.

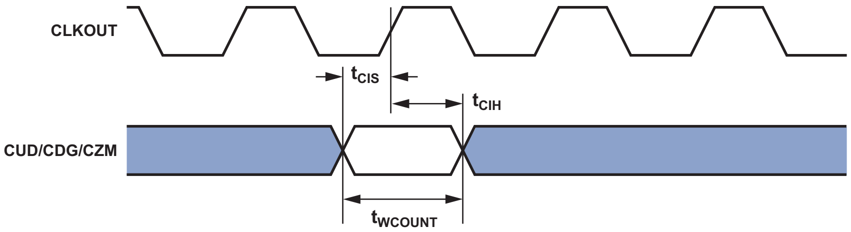


Figure 43. Up/Down Counter/Rotary Encoder Timing

ATA/ATAPI-6 Interface Timing

The following tables and figures specify ATAPI timing parameters. For detailed parameter descriptions, refer to the ATAPI specification (ANSI INCITS 361-2002). Table 58 to Table 61 include ATAPI timing parameter equations. System designers should use these equations along with the parameters provided in Table 56 and Table 57. ATAPI timing control registers should be programmed such that ANSI INCITS 361-2002 specifications are met for the desired transfer type and mode.

Table 56. ATA/ATAPI-6 Timing Parameters

Parameter		Min	Max	Unit
t_{SK1}	Difference in output delay after CLKOUT for ATAPI output pins[1]		6	ns
t_{OD}	Output delay after CLKOUT for outputs[1]		12	ns
t_{SUD}	ATAPI_D0-15 or ATAPI_D0-15A Setup Before CLKOUT	6		ns
t_{SUI}	ATAPI_IORDY Setup Before CLKOUT	6		ns
t_{SUDU}	ATAPI_D0-15 or ATAPI_D0-15A Setup Before ATAPI_IORDY (UDMA-in only)	2		ns
t_{HDU}	ATAPI_D0-15 or ATAPI_D0-15A Hold After ATAPI_IORDY (UDMA-in only)	2.6		ns

[1] ATAPI output pins include ATAPI_CS0, ATAPI_CS1, A1-3, ATAPI_DIOR, ATAPI_DIOW, ATAPI_DMACK, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.

Table 57. ATA/ATAPI-6 System Timing Parameters

Parameter		Source
t_{SK2}	Maximum difference in board propagation delay between any 2 ATAPI output pins[1]	System Design
t_{BD}	Maximum board propagation delay.	System Design
t_{SK3}	Maximum difference in board propagation delay during a read between ATAPI_IORDY and ATAPI_D0-15/ATAPI_D0-15A.	System Design
t_{SK4}	Maximum difference in ATAPI cable propagation delay between output pin group A and output pin group B[2]	ATAPI Cable Specification
t_{CDD}	ATAPI cable propagation delay for ATAPI_D0-15 and ATAPI_D0-15A signals.	ATAPI Cable Specification
t_{CDC}	ATAPI cable propagation delay for ATAPI_DIOR, ATAPI_DIOW, ATAPI_IORDY, and ATAPI_DMACK signals.	ATAPI Cable Specification

[1] ATAPI output pins include ATAPI_CS0, ATAPI_CS1, A1-3, ATAPI_DIOR, ATAPI_DIOW, ATAPI_DMACK, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.

[2] Output pin group A includes ATAPI_DIOR, ATAPI_DIOW, and ATAPI_DMACK. Output pin group B includes ATAPI_CS0, ATAPI_CS1, A1-3, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.

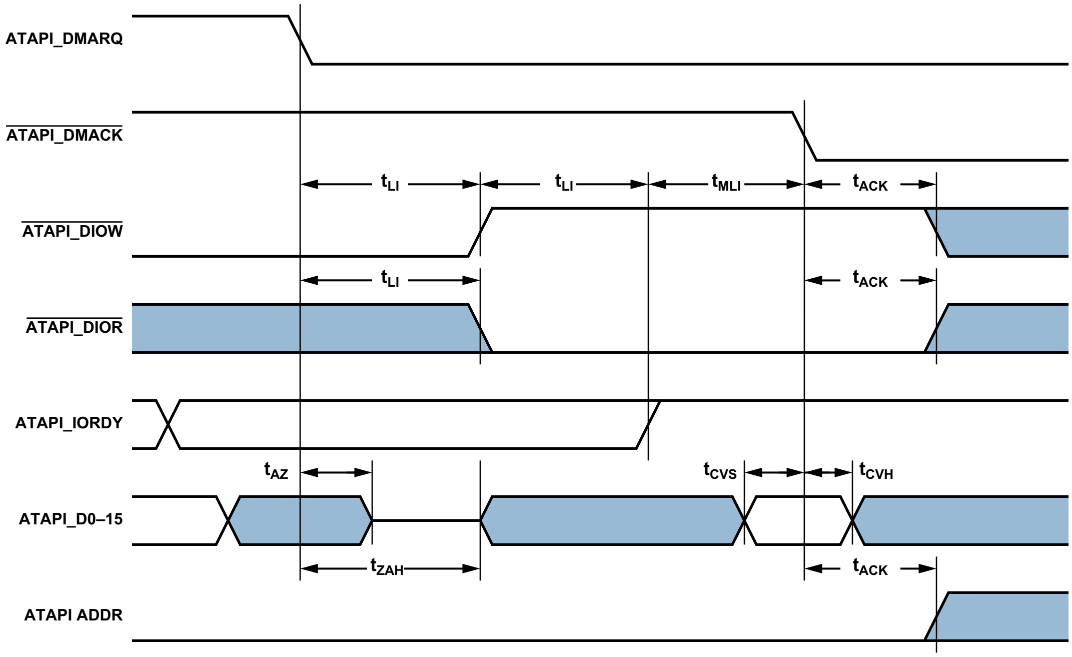


Figure 55. Device Terminating an Ultra DMA Data-In Burst

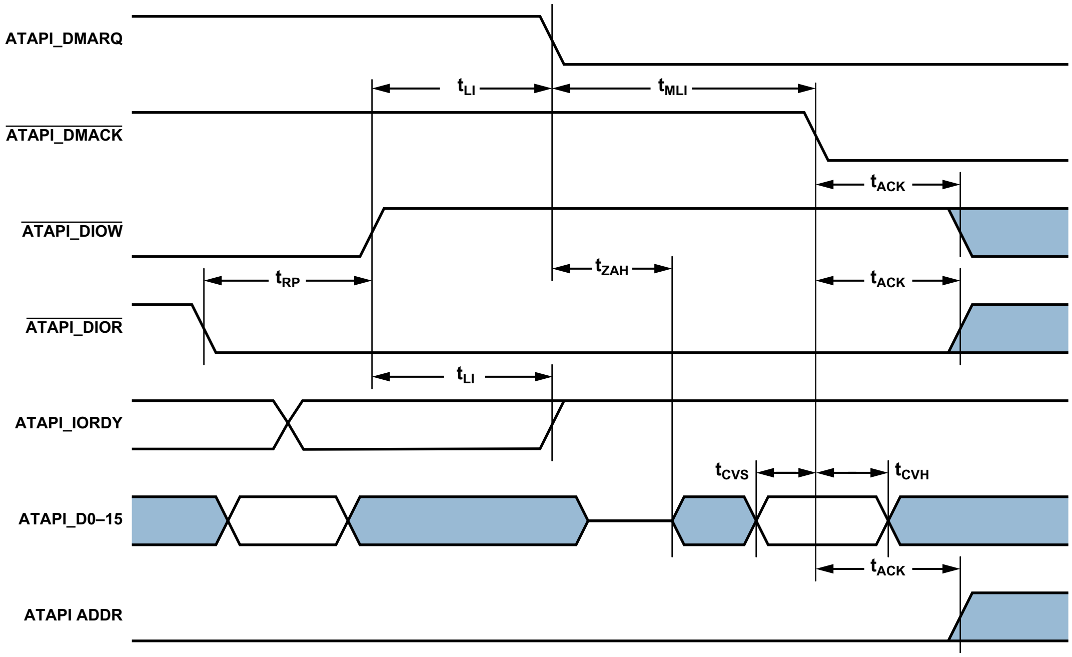


Figure 56. Host Terminating an Ultra DMA Data-In Burst

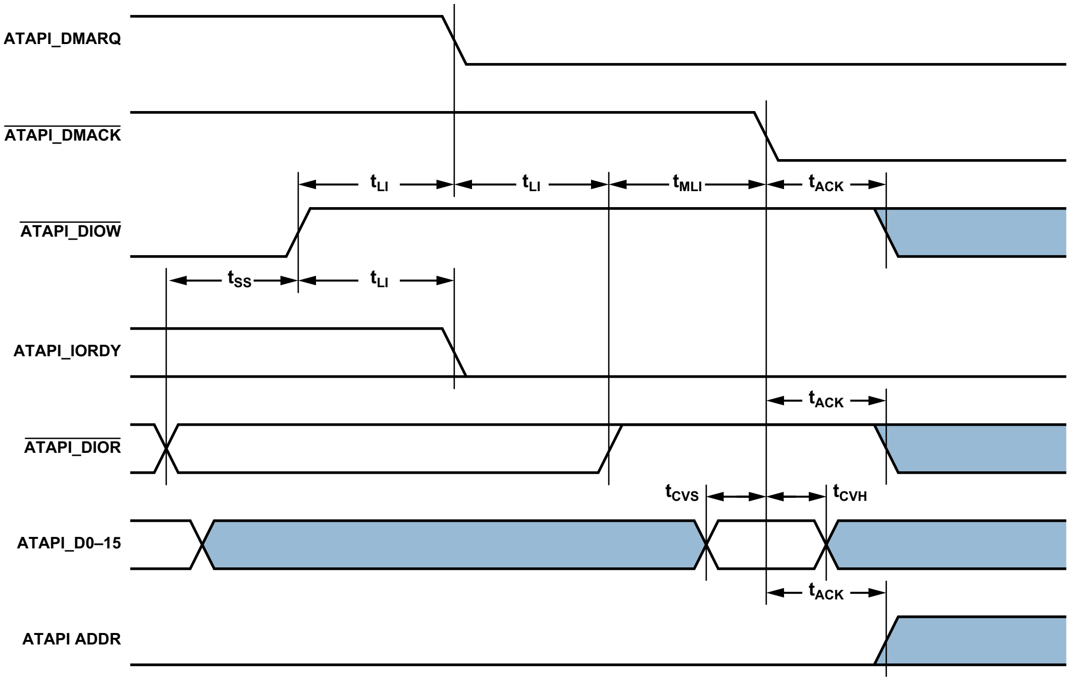


Figure 59. Host terminating an Ultra DMA Data-Out Burst

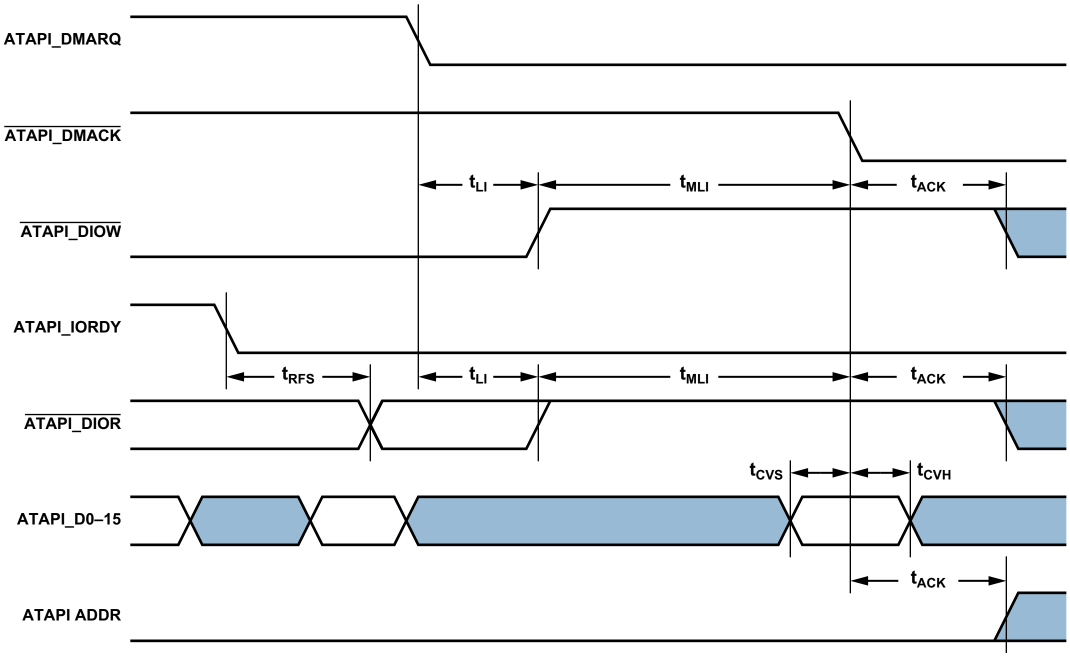


Figure 60. Device Terminating an Ultra DMA Data-Out Burst

USB On-The-Go-Dual-Role Device Controller Timing

Table 62 describes the USB On-The-Go Dual-Role Device Controller timing requirements.

Table 62. USB On-The-Go Dual-Role Device Controller Timing Requirements

Parameter		Min	Max	Unit
Timing Requirements				
f_{USB}	USB_XI frequency	9	33.3	MHz
FS_{USB}	USB_XI Clock Frequency Stability	–50	+50	ppm

JTAG Test And Emulation Port Timing

Table 63 and Figure 61 describe JTAG port operations.

Table 63. JTAG Port Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{TCK}	TCK Period	20		ns
t_{STAP}	TDI, TMS Setup Before TCK High	4		ns
t_{HTAP}	TDI, TMS Hold After TCK High	4		ns
t_{SSYS}	System Inputs Setup Before TCK High[1]	4		ns
t_{HSYS}	System Inputs Hold After TCK High[1]	11		ns
t_{TRSTW}	$\overline{\text{TRST}}$ Pulse-Width[2] (measured in TCK cycles)	4		t_{TCK}
Switching Characteristics				
t_{DTDO}	TDO Delay from TCK Low		10	ns
t_{DSYS}	System Outputs Delay After TCK Low[3]	0	16.5	ns

[1] System inputs = PA15–0, PB14–0, PC13–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ13–0, DQ15–0, DQS1–0, D15–0, $\overline{\text{ATAPI\_PDIAG}}$, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, and BMODE3–0.

[2] 50 MHz Maximum.

[3] System outputs = PA15–0, PB14–0, PC13–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ13–0, DQ15–0, DQS1–0, D15–0, DA12–0, DBA1–0, DQM1–0, DCLK0-1, $\overline{\text{DCLK0–1}}$, $\overline{\text{DCS1–0}}$, DCLKE, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, $\overline{\text{DWE}}$, $\overline{\text{AMS3–0}}$, $\overline{\text{ABE1–0}}$, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, CLKOUT, A3–1, and MFS.

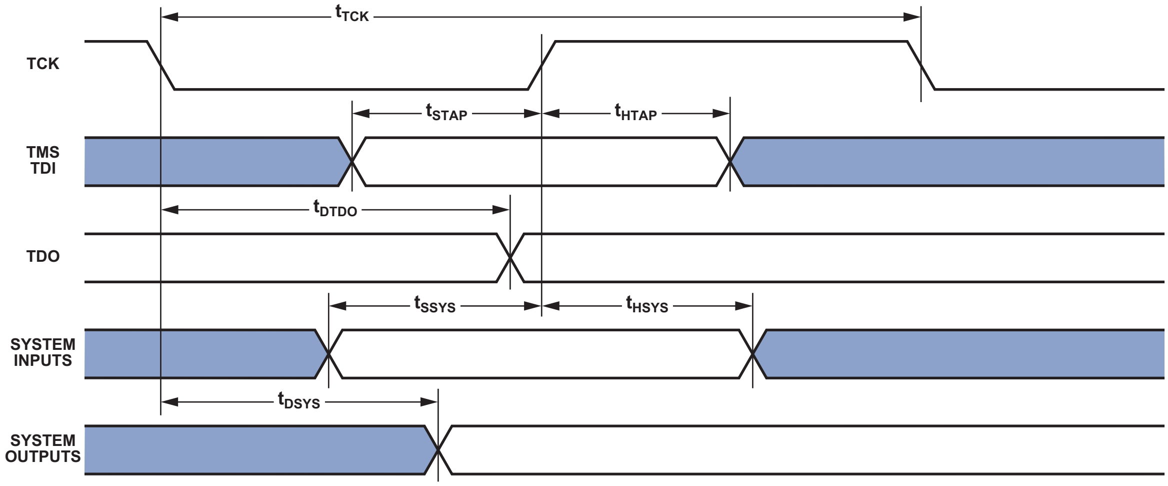


Figure 61. JTAG Port Timing

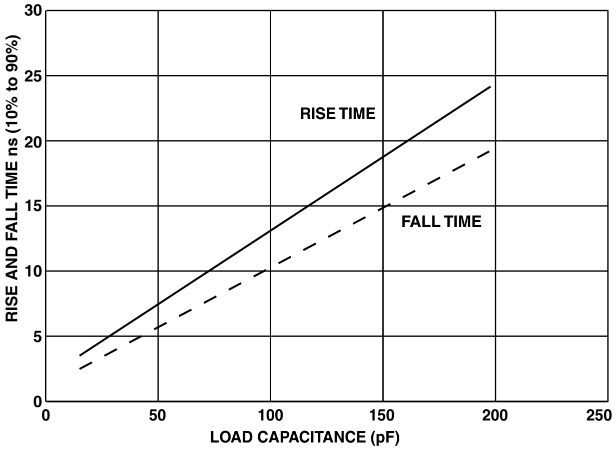


Figure 79. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at V_{DDEXT} = 2.25 V

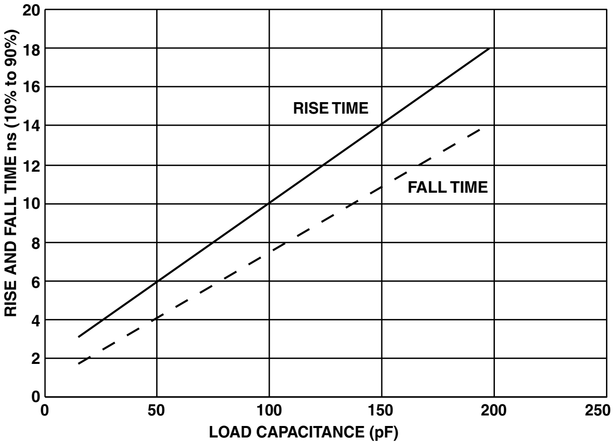


Figure 80. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at V_{DDEXT} = 3.65 V

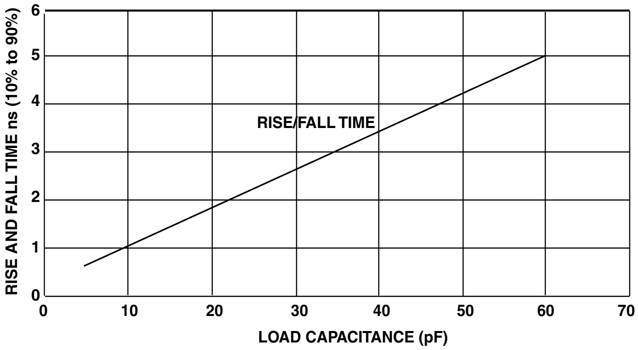


Figure 81. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D DDR SDRAM at V_{DDDDR} = 2.5 V

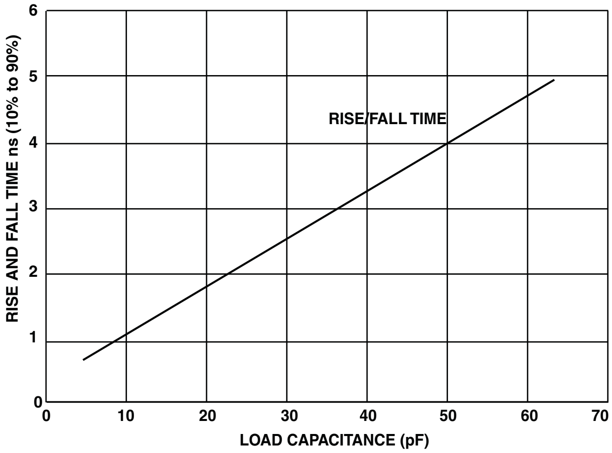


Figure 82. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D DDR SDRAM at V_{DDDDR} = 2.7 V

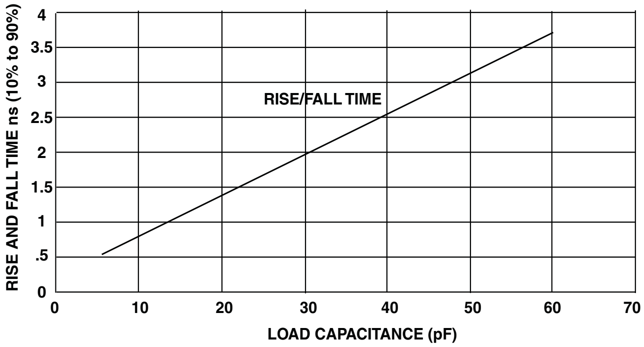


Figure 83. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D Mobile DDR SDRAM at V_{DDDDR} = 1.8 V

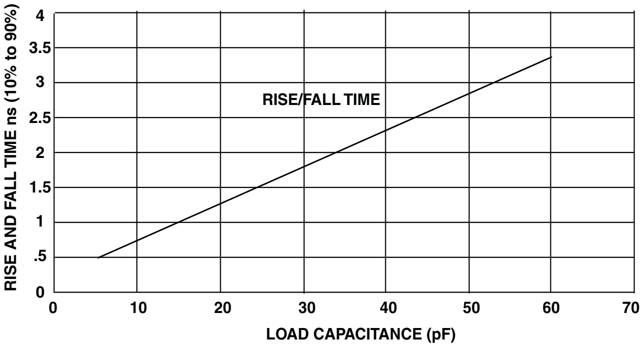


Figure 84. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D Mobile DDR SDRAM at V_{DDDDR} = 1.95 V