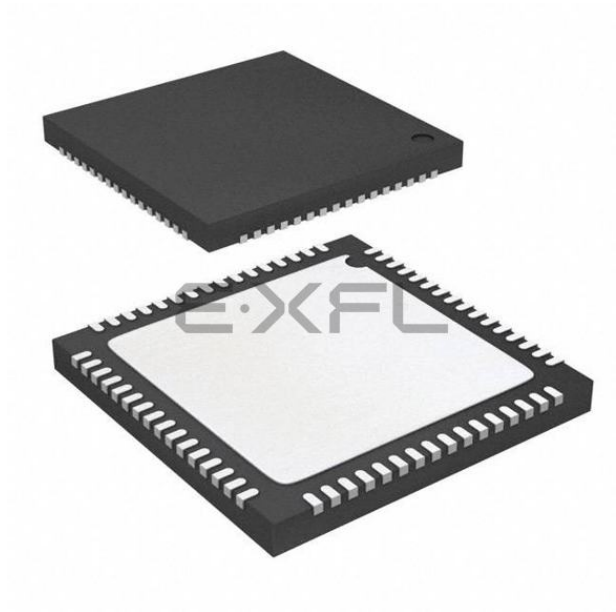

Welcome to E-XFL.COM

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	I²C, I²S, IrDA, PPI, SPI, SPORT, UART
Clock Rate	400MHz
Non-Volatile Memory	ROM (64kB)
On-Chip RAM	64kB
Voltage - I/O	1.8V, 2.5V, 3.3V
Voltage - Core	1.29V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad, CSP
Supplier Device Package	64-LFCSP-VQ (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf592kcpz

- EE-302: Interfacing ADSP-BF53x Blackfin® Processors to NAND FLASH Memory
- EE-303: Using VisualDSP++® Thread-Safe Libraries with a Third-Party RTOS
- EE-304: Using the Blackfin® Processor SPORT to Emulate a SPI Interface
- EE-306: PGO Linker - A Code Layout Tool for Blackfin Processors
- EE-312: Building Complex VDK/LwIP Applications Using Blackfin® Processors
- EE-323: Implementing Dynamically Loaded Software Modules
- EE-325: Interfacing Atmel Fingerprint Sensor AT77C104B with Blackfin® Processors
- EE-326: Blackfin® Processor and SDRAM Technology
- EE-330: Windows Vista Compatibility in VisualDSP++ 5.0 Development Tools
- EE-332: Cycle Counting and Profiling
- EE-333: Interfacing Blackfin® Processors to Winbond W25X16 SPI Flash Devices
- EE-334: Using Blackfin® Processor Hibernate State for Low Standby Power
- EE-336: Putting ADSP-BF54x Blackfin® Processor Booting into Practice
- EE-339: Using External Switching Regulators with Blackfin® Processors
- EE-340: Connecting SHARC® and Blackfin® Processors over SPI
- EE-341: Expert Pin Multiplexing Plug-in for Blackfin® Processors
- EE-347: Formatted Print to a UART Terminal with Blackfin® Processors
- EE-350: Seamlessly Interfacing MEMS Microphones with Blackfin Processors
- EE-351: Using the ADSP-BF592 Blackfin® Processor Tools Utility ROM
- EE-352: Soldering Considerations for Exposed-Pad Packages
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users

Data Sheet

- ADSP-BF592 Blackfin Embedded Processor Data Sheet

Emulator Manuals

- HPUSB, USB, and HPPCI Emulator User's Guide
- ICE-1000/ICE-2000 Emulator User's Guide
- ICE-100B Emulator User's Guide

Evaluation Kit Manuals

- ADSP-BF592 EZ-KIT Lite® Evaluation System Manual
- Blackfin®/SHARC® USB EZ-Extender® Manual

Integrated Circuit Anomalies

- ADSP-BF592 Blackfin Silicon Anomaly List for Revisions 0.1, 0.2

Processor Manuals

- ADSP-BF59x Blackfin ® Processor Hardware Reference
- ADSP-BF5xx/ADSP-BF60x Blackfin® Processor Programming Reference
- Blackfin Processors: Manuals

Software Manuals

- CrossCore® Embedded Studio 2.5.0 Assembler and Preprocessor Manual
- CrossCore® Embedded Studio 2.5.0 C/C++ Compiler and Library Manual for Blackfin Processors
- CrossCore® Embedded Studio 2.5.0 Linker and Utilities Manual
- CrossCore® Embedded Studio 2.5.0 Loader and Utilities Manual
- CrossCore® Software Licensing Guide
- lwIP for CrossCore® Embedded Studio 1.0.0 User's Guide
- VisualDSP++® 5.0 Assembler and Preprocessor Manual
- VisualDSP++® 5.0 C/C++ Compiler and Library Manual for Blackfin Processors
- VisualDSP++® 5.0 Device Drivers and System Services Manual for Blackfin Processors
- VisualDSP++® 5.0 Kernel (VDK) Users Guide
- VisualDSP++® 5.0 Licensing Guide
- VisualDSP++® 5.0 Linker and Utilities Manual
- VisualDSP++® 5.0 Loader and Utilities Manual
- VisualDSP++® 5.0 Product Release Bulletin
- VisualDSP++® 5.0 Quick Installation Reference Card
- VisualDSP++® 5.0 Users Guide

SOFTWARE AND SYSTEMS REQUIREMENTS

- Software and Tools Anomalies Search

TOOLS AND SIMULATIONS

- ADSP-BF592 Blackfin Processor BSDL File 64-Lead LFCSP (11/2011)
- Blackfin Processors Software and Tools
- ADSP-BF592 Blackfin Processor IBIS Datafile 64 Lead LFCSP Package (11/2011)

GENERAL DESCRIPTION

The ADSP-BF592 processor is a member 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.

The ADSP-BF592 processor is completely code compatible with other Blackfin processors. The ADSP-BF592 processor offers performance up to 400 MHz and reduced static power consumption. The processor features are shown in Table 1.

Table 1. Processor Features

Feature		ADSP-BF592
Timer/Counters with PWM		3
SPORTs		2
SPIs		2
UART		1
Parallel Peripheral Interface		1
TWI		1
GPIOs		32
Memory (bytes)	L1 Instruction SRAM	32K
	L1 Instruction ROM	64K
	L1 Data SRAM	32K
	L1 Scratchpad SRAM	4K
	L3 Boot ROM	4K
Maximum Instruction Rate[1]		400 MHz
Maximum System Clock Speed		100 MHz
Package Options		64-Lead LFCSP

[1] Maximum instruction rate is not available with every possible SCLK selection.

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.

PORTABLE LOW POWER ARCHITECTURE

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

SYSTEM INTEGRATION

The ADSP-BF592 processor is a highly integrated system-on-a-chip solution for the next generation of digital communication and consumer multimedia applications. By combining industry standard interfaces with a high performance signal processing core, cost-effective applications can be developed quickly, without the need for costly external components. The system peripherals include a watchdog timer; three 32-bit timers/counters with PWM support; two dual-channel, full-duplex synchronous serial ports (SPORTs); two serial peripheral interface (SPI) compatible ports; one UART® with IrDA support; a parallel peripheral interface (PPI); and a 2-wire interface (TWI) controller.

BLACKFIN PROCESSOR CORE

As shown in Figure 2, 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-, 16-, 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 includes 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. The compare/select and vector search instructions are also provided.

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). If the second ALU is used, 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 over

head 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. Data memory holds data, and a dedicated scratchpad data memory stores stack and local variable information.

Multiple L1 memory blocks are provided. 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.

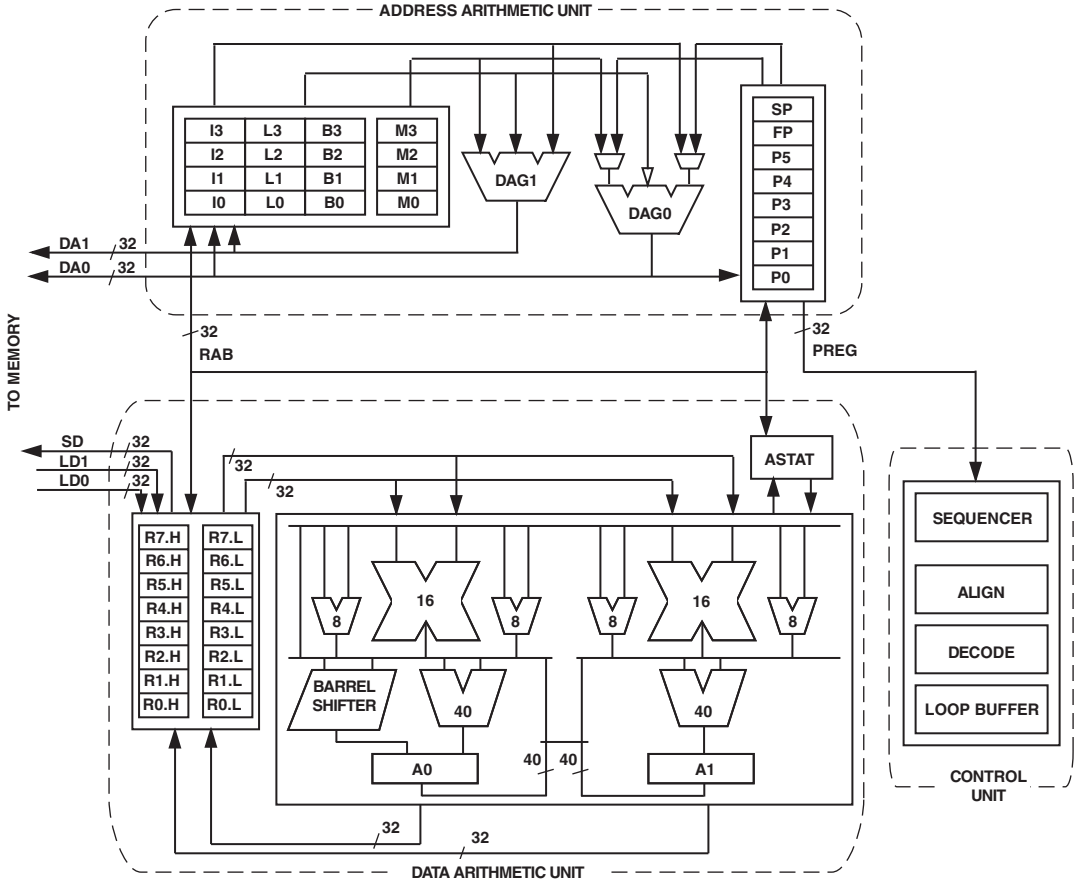

Figure 2. Blackfin Processor Core

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 processor peripherals. After a reset, software can determine whether 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 four general-purpose programmable timer units in the processor. Three timers have 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, to 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 UART 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 three general-purpose programmable timers, a fourth timer is also provided. 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.

Serial Ports

The ADSP-BF592 processor incorporates two dual-channel synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

Serial port data can be automatically transferred to and from on-chip memory/external memory via dedicated DMA channels. Each of the serial ports can work in conjunction with another serial port to provide TDM support. In this configuration, one SPORT provides two transmit signals while the other SPORT provides the two receive signals. The frame sync and clock are shared.

Serial ports operate in five modes:

- Standard DSP serial mode
- Multichannel (TDM) mode
- I^2S mode
- Packed I^2S mode
- Left-justified mode

Serial Peripheral Interface (SPI) Ports

The processor has two SPI-compatible ports that enable the processor to communicate with multiple SPI-compatible devices.

The SPI interface uses three pins for transferring data: two data pins (Master Output-Slave Input, MOSI, and Master Input-Slave Output, MISO) and a clock pin (serial clock, SCK). An SPI chip select input pin ($\overline{\text{SPIx\_SS}}$) lets other SPI devices select the processor, and many SPI chip select output pins ($\overline{\text{SPIx\_SEL7–1}}$) let the processor select other SPI devices. The SPI select pins are reconfigured general-purpose I/O pins. Using these pins, the SPI port provides a full-duplex, synchronous serial interface, which supports both master/slave modes and multimaster environments.

UART Port

The ADSP-BF592 processor provides a full-duplex universal asynchronous receiver/transmitter (UART) port, which is fully compatible with PC-standard UARTs. The UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. The UART port includes support for five to eight data bits, one or two stop bits, and none, even, or odd parity. The UART port supports two modes of operation:

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

Parallel Peripheral Interface (PPI)

The processor provides a parallel peripheral interface (PPI) that can connect directly to parallel analog-to-digital and digital-to-analog converters, video encoders and decoders, and other general-purpose peripherals. The PPI consists of a dedicated input clock pin, up to three frame synchronization pins, and up to 16 data pins. The input clock supports parallel data rates up to half the system clock rate, and the synchronization signals can be configured as either inputs or outputs.

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

General-Purpose Mode Descriptions

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

- Input mode – Frame syncs and data are inputs into the PPI. Input mode is intended for ADC applications, as well as video communication with hardware signaling.
- Frame capture mode – Frame syncs are outputs from the PPI, but data are inputs. This mode allows the video source(s) to act as a slave (for frame capture for example).
- Output mode – Frame syncs and data are outputs from the PPI. Output mode is used for transmitting video or other data with up to three output frame syncs.

ITU-R 656 Mode Descriptions

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

- Active video only mode – Active video only mode is used when only the active video portion of a field is of interest and not any of the blanking intervals.
- Vertical blanking only mode – In this mode, the PPI only transfers vertical blanking interval (VBI) data.
- Entire field mode – In this mode, the entire incoming bit stream is read in through the PPI.

TWI Controller Interface

The processor includes a 2-wire interface (TWI) module for providing a simple exchange method of control data between multiple devices. The TWI is functionally compatible with the widely used I²C® bus standard. The TWI module offers the capabilities of simultaneous master and slave operation and support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (SCL) and data (SDA) and supports the protocol at speeds up to 400K bits/sec.

The TWI module is compatible with serial camera control bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

Ports

The processor groups the many peripheral signals to two ports—Port F and Port G. Most of the associated pins are shared by multiple signals. The ports function as multiplexer controls.

General-Purpose I/O (GPIO)

The processor has 32 bidirectional, general-purpose I/O (GPIO) pins allocated across two separate GPIO modules—PORTFIO and PORTGIO, associated with Port F and Port G respectively. Each GPIO-capable pin shares functionality with other processor peripherals via a multiplexing scheme; however, the GPIO functionality is the default state of the device upon power-up. Neither GPIO output nor input drivers are active by default. Each general-purpose port pin can be individually controlled by manipulation of the port control, status, and interrupt registers.

DYNAMIC POWER MANAGEMENT

The 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. When configured for a 0 V core supply voltage, the processor enters the hibernate state. Control of clocking to each of the processor peripherals also reduces power consumption. See Table 2 for a summary of the power settings for each mode.

Table 2. Power Settings

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

Full-On Operating Mode—Maximum Performance

In the full-on mode, the PLL is enabled and is not bypassed, providing capability for maximum operational frequency. This is the 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 Dynamic 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.

For more information about PLL controls, see the "Dynamic Power Management" chapter in the *ADSP-BF59x Blackfin Processor Hardware Reference*.

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 wakes up the processor.

System DMA access to L1 memory is not supported in sleep mode.

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 may still be running but cannot 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 a GPIO pin.

Note that when a GPIO pin is used to trigger wake from deep sleep, the programmed wake level must linger for at least 10ns to guarantee detection.

Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling clocks to the processor core (CCLK) and to all of the peripherals (SCLK), as well as signaling an external voltage regulator that V_{DDINT} can be shut off. Any critical information stored internally (for example, memory contents, register contents, and other information) must be written to a nonvolatile storage device prior to removing power if the processor state is to be preserved. Writing b#0 to the $\overline{\text{HIBERNATE}}$ bit causes EXT_WAKE to transition low, which can be used to signal an external voltage regulator to shut down.

Since V_{DDEXT} can still be supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that may be connected to the processor to still have power applied without drawing unwanted current.

As long as V_{DDEXT} is applied, the VR_CTL register maintains its state during hibernation. All other internal registers and memories, however, lose their content in the hibernate state.

Power Savings

As shown in Table 3, the processor supports two different power domains, which maximizes flexibility while maintaining compliance with industry standards and conventions. By isolating the internal logic of the processor into its own power domain, separate from other I/O, the processor can take advantage of dynamic power management without affecting the other I/O devices. There are no sequencing requirements for the various power domains, but all domains must be powered according to the appropriate Specifications table for processor operating conditions, even if the feature/peripheral is not used.

Table 3. Power Domains

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

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

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

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

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

where:

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

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

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

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

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

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

VOLTAGE REGULATION

The ADSP-BF592 processor requires an external voltage regulator to power the V_{DDINT} domain. To reduce standby power consumption, the external voltage regulator can be signaled through EXT_WAKE to remove power from the processor core. This signal is high-true for power-up and may be connected directly to the low-true shut-down input of many common regulators.

While in the hibernate state, the external supply, V_{DDEXT}, can still be applied, eliminating the need for external buffers. The external voltage regulator can be activated from this power-down state by asserting the $\overline{\text{RESET}}$ pin, which then initiates a boot sequence. EXT_WAKE indicates a wakeup to the external voltage regulator.

The power good ($\overline{\text{PG}}$) input signal allows the processor to start only after the internal voltage has reached a chosen level. In this way, the startup time of the external regulator is detected after hibernation. For a complete description of the power-good functionality, refer to the *ADSP-BF59x Blackfin Processor Hardware Reference*.

CLOCK SIGNALS

The 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 processor includes an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 4. 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

BOOTING MODES

The processor has several mechanisms (listed in Table 6) for automatically loading internal and external memory after a reset. The boot mode is defined by the 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 external host devices.

Table 6. Booting Modes

BMODE2–0	Description
000	Idle/No Boot
001	Reserved
010	SPI1 master boot from Flash, using $\overline{\text{SPI1\_SSEL5}}$ on PG11
011	SPI1 slave boot from external master
100	SPI0 master boot from Flash, using $\overline{\text{SPI0\_SSEL2}}$ on PF8
101	Boot from PPI port
110	Boot from UART host device
111	Execute from Internal L1 ROM

The boot modes listed in Table 6 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. The BMODE pins of the reset configuration register, sampled during power-on resets and software-initiated resets, implement the modes shown in Table 6.

- IDLE State/No Boot (BMODE - 0x0) — In this mode, the boot kernel transitions the processor into Idle state. The processor can then be controlled through JTAG for recovery, debug, or other functions.
- SPI1 master boot from flash (BMODE = 0x2) — In this mode, SPI1 is configured to operate in master mode and to connect to 8-, 16-, 24-, or 32-bit addressable devices. The processor uses the PG11/$\overline{\text{SPI1\_SSEL5}}$ to select a single SPI EEPROM/flash device, submits a read command and successive address bytes (0×00) until a valid 8-, 16-, 24-, or 32-bit addressable device is detected, and begins clocking data into the processor. Pull-up resistors are required on the SSEL and MISO pins. By default, a value of 0×85 is written to the SPI_BAUD register.
- SPI1 slave boot from external master (BMODE = 0x3) — In this mode, SPI1 is configured to operate in slave mode and to receive the bytes of the .LDR file from a SPI host (master) agent. To hold off the host device from transmitting while the boot ROM is busy, the Blackfin processor asserts a GPIO pin, called host wait (HWAIT), to signal to the host device not to send any more bytes until the pin is deasserted. The host must interrogate the HWAIT signal, available on PG4, before transmitting every data unit to the processor. A pull-up resistor is required on the $\overline{\text{SPI1\_SS}}$ input. A pull-down on the serial clock may improve signal quality and booting robustness.
- SPI0 master boot from flash (BMODE = 0x4) — In this mode SPI0 is configured to operate in master mode and to connect to 8-, 16-, 24-, or 32-bit addressable devices. The processor uses the PF8/$\overline{\text{SPI0\_SSEL2}}$ to select a single SPI EEPROM/flash device, submits a read command and successive address bytes (0×00) until a valid 8-, 16-, 24-, or 32-bit addressable device is detected, and begins clocking data into the processor. Pull-up resistors are required on the SSEL and MISO pins. By default, a value of 0×85 is written to the SPI_BAUD register.
- Boot from PPI host device (BMODE = 0x5) — The processor operates in PPI slave mode and is configured to receive the bytes of the LDR file from a PPI host (master) agent.
- Boot from UART host device (BMODE = 0x6) — In this mode UART0 is used as the booting source. Using an autobaud handshake sequence, a boot-stream formatted program is downloaded by the host. The host selects a bit rate within the UART clocking capabilities. When performing the autobaud, the UART expects a "@" (0×40) character (eight bits data, one start bit, one stop bit, no parity bit) on the RXD pin to determine the bit rate. The UART then replies with an acknowledgment which is composed of 4 bytes (0xBF—the value of UART_DLL) and (0×00—the value of UART_DLH). The host can then download the boot stream. To hold off the host the processor signals the host with the boot host wait (HWAIT) signal. Therefore, the host must monitor the HWAIT, (on PG4), before every transmitted byte.
- Execute from internal L1 ROM (BMODE = 0x7) — In this mode the processor begins execution from the on-chip 64k byte L1 instruction ROM starting at address 0xFFA1 0000.

For each of the boot modes (except Execute from internal L1 ROM), a 16 byte header is first brought in from an external 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 start of L1 instruction SRAM.

The boot kernel differentiates between a regular hardware reset and a wakeup-from-hibernate event to speed up booting in the latter case. Bits 7–4 in the system reset configuration (SYSCR) register can be used to bypass the boot kernel or simulate a wakeup-from-hibernate boot in case of a software reset.

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 speed up booting by managing the PLL, clock frequencies, or serial bit rates.

The boot ROM also features C-callable functions 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-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

DEVELOPMENT TOOLS

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

Integrated Development Environments (IDEs)

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

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

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

EZ-KIT Lite Evaluation Board

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

EZ-KIT Lite Evaluation Kits

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

Software Add-Ins for 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.

ADDITIONAL INFORMATION

The following publications that describe the ADSP-BF592 processor (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on our website:

- *Getting Started With Blackfin Processors*
- *ADSP-BF59x Blackfin Processor Hardware Reference*
- *Blackfin Processor Programming Reference*
- *ADSP-BF592 Blackfin Processor Anomaly List*

RELATED SIGNAL CHAINS

A *signal chain* is a series of signal conditioning electronic components that receive input (data acquired from sampling either real-time phenomena or from stored data) in tandem, with the output of one portion of the chain supplying input to the next. Signal chains are often used in signal processing applications to gather and process data or to apply system controls based on analysis of real-time phenomena. For more information about this term and related topics, see the "signal chain" entry in the Glossary of EE Terms on the Analog Devices website.

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

The Circuits from the Lab™ site (www.analog.com\circuits) provides:

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

SPECIFICATIONS

Specifications are subject to change without notice.

OPERATING CONDITIONS

Parameter		Conditions	Min	Nominal	Max	Unit
V_{DDINT}	Internal Supply Voltage	Non-Automotive Models	1.1		1.47	V
	Internal Supply Voltage	Automotive Models	1.33		1.47	V
V_{DDEXT}	External Supply Voltage	Non-Automotive Models	1.7	1.8/2.5/3.3	3.6	V
	External Supply Voltage	Automotive Models	2.7		3.6	V
V_{IH}	High Level Input Voltage[1, 2]	V_{DDEXT} = 1.9 V	1.1			V
$V_{IHCLKIN}$	High Level Input Voltage[1, 2]	V_{DDEXT} = 1.9 V	1.2			V
V_{IH}	High Level Input Voltage[1, 2]	V_{DDEXT} = 2.75 V	1.7			V
V_{IH}	High Level Input Voltage[1, 2]	V_{DDEXT} = 3.6 V	2.0			V
$V_{IHCLKIN}$	High Level Input Voltage[1, 2]	V_{DDEXT} = 3.6 V	2.2			V
V_{IHTWI}	High Level Input Voltage[3]	V_{DDEXT} = 1.90 V/2.75 V/3.6 V	0.7 × V_{DDEXT}		3.6	V
V_{IL}	Low Level Input Voltage[1, 2]	V_{DDEXT} = 1.7 V			0.6	V
V_{IL}	Low Level Input Voltage[1, 2]	V_{DDEXT} = 2.25 V			0.7	V
V_{IL}	Low Level Input Voltage[1, 2]	V_{DDEXT} = 3.0 V			0.8	V
V_{ILTWI}	Low Level Input Voltage[3]	V_{DDEXT} = Minimum			0.3 × V_{DDEXT}	V
T_J	Junction Temperature	64-Lead LFCSP @ $T_{AMBIENT}$ = 0°C to +70°C	0		80	°C
T_J	Junction Temperature	64-Lead LFCSP @ $T_{AMBIENT}$ = –40°C to +85°C	–40		+95	°C
T_J	Junction Temperature	64-Lead LFCSP @ $T_{AMBIENT}$ = –40°C to +105°C	–40		+115	°C

[1] Bidirectional leads (PF15–0, PG15–0) and input leads (TCK, TDI, TMS, $\overline{TRST}$, CLKIN, $\overline{RESET}$, $\overline{NMI}$, and BMODE2–0) of the ADSP-BF592 processor are 3.3 V tolerant (always accept up to 3.6 V maximum V_{IH}). Voltage compliance (on outputs, V_{OH}) is limited by the V_{DDEXT} supply voltage.

[2] Parameter value applies to all input and bidirectional leads, except SDA and SCL.

[3] Parameter applies to SDA and SCL.

Total Power Dissipation

Total power dissipation has two components:

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

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. Electrical Characteristics on Page 18 shows the current dissipation for internal circuitry (V_{DDINT}). $I_{DDDEEPSLEEP}$ specifies static power dissipation as a function of voltage (V_{DDINT}) and temperature (see Table 12), and I_{DDINT} specifies the total power specification for the listed test conditions, including the dynamic component as a function of voltage (V_{DDINT}) and frequency (Table 13).

There are two parts to the dynamic component. The first part is due to transistor switching in the core clock (CCLK) domain. This part is subject to an Activity Scaling Factor (ASF), which represents application code running on the processor core and L1 memories (Table 11).

The ASF is combined with the CCLK frequency and V_{DDINT} dependent data in Table 13 to calculate this part. The second part is due to transistor switching in the system clock (SCLK) domain, which is included in the I_{DDINT} specification equation.

Table 11. Activity Scaling Factors (ASF)[1]

I_{DDINT} Power Vector	Activity Scaling Factor (ASF)
$I_{DD-PEAK}$	1.29
$I_{DD-HIGH}$	1.26
I_{DD-TYP}	1.00
I_{DD-APP}	0.83
I_{DD-NOP}	0.66
$I_{DD-IDLE}$	0.33

[1] See *Estimating Power for ASDP-BF534/BF536/BF537 Blackfin Processors (EE-297)*. The power vector information also applies to the ADSP-BF592 processor.

Table 12. Static Current - $I_{DD-DEEPSLEEP}$ (mA)

	Voltage (V_{DDINT})[1]							
T_J (°C)[1]	1.15 V	1.20 V	1.25 V	1.30 V	1.35 V	1.40 V	1.45 V	1.50 V
25	0.85	0.98	1.13	1.29	1.46	1.62	1.85	2.07
40	1.57	1.8	2.01	2.16	2.51	2.74	3.05	3.36
55	2.57	2.88	3.2	3.5	3.84	4.22	4.63	5.05
70	4.04	4.45	4.86	5.3	5.81	6.31	6.87	7.45
85	6.52	7.12	7.73	8.36	9.09	9.86	10.67	11.54
100	9.67	10.51	11.37	12.24	13.21	14.26	15.37	16.55
115	14.18	15.29	16.45	17.71	19.05	20.45	21.96	23.56

[1] Valid temperature and voltage ranges are model-specific. See Operating Conditions on Page 16.

Table 13. Dynamic Current in CCLK Domain (mA, with ASF = 1.0)[1]

f_{CCLK} (MHz)[2]	Voltage (V_{DDINT})[2]							
	1.15 V	1.20 V	1.25 V	1.30 V	1.35 V	1.40 V	1.45 V	1.50 V
400	N/A	N/A	N/A	N/A	85.31	88.96	92.81	96.63
350	N/A	N/A	N/A	72.08	75.41	78.70	82.07	85.46
300	N/A	57.52	60.38	63.22	66.14	69.02	71.93	75.05
250	46.10	48.43	50.76	53.19	55.68	58.17	60.69	63.23
200	37.86	39.80	41.76	43.79	45.81	47.85	49.97	52.09
100	21.45	22.56	23.78	24.98	25.97	26.64	27.92	29.98

[1] The values are not guaranteed as stand-alone maximum specifications. They must be combined with static current per the equations of Electrical Characteristics on Page 18.
[2] Valid frequency and voltage ranges are model-specific. See Operating Conditions on Page 16 and Table 8 on Page 17.

TIMING SPECIFICATIONS

Specifications are subject to change without notice.

Clock and Reset Timing

Table 18 and Figure 7 describe clock and reset operations. Per the CCLK and SCLK timing specifications in Table 8 to Table 10, combinations of CLKIN and clock multipliers must not select core/peripheral clocks in excess of the processor's instruction rate.

Table 18. Clock and Reset Timing

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Timing Requirements						
f_{CKIN}	CLKIN Period[1, 2, 3, 4]	12	50	12	50	MHz
t_{CKINL}	CLKIN Low Pulse[1]	10		10		ns
t_{CKINH}	CLKIN High Pulse[1]	10		10		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulse Width Low[5]	11 × t_{CKIN}		11 × t_{CKIN}		ns
Switching Characteristic						
$t_{BUFDLAY}$	CLKIN to CLKBUF[6] Delay		11		10	ns

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

[2] Combinations of the CLKIN frequency and the PLL clock multiplier must not exceed the allowed f_{VCO}, f_{CCLK}, and f_{SCLK} settings discussed in Table 8 on Page 17 through Table 10 on Page 17.

[3] The t_{CKIN} period (see Figure 7) equals $1/f_{CKIN}$.

[4] If the DF bit in the PLL_CTL register is set, the minimum f_{CKIN} specification is 24 MHz.

[5] Applies after power-up sequence is complete. See Table 19 and Figure 8 for power-up reset timing.

[6] The ADSP-BF592 processor does not have a dedicated CLKBUF pin. Rather, the EXTCLK pin may be programmed to serve as CLKBUF or CLKOUT. This parameter applies when EXTCLK is programmed to output CLKBUF.

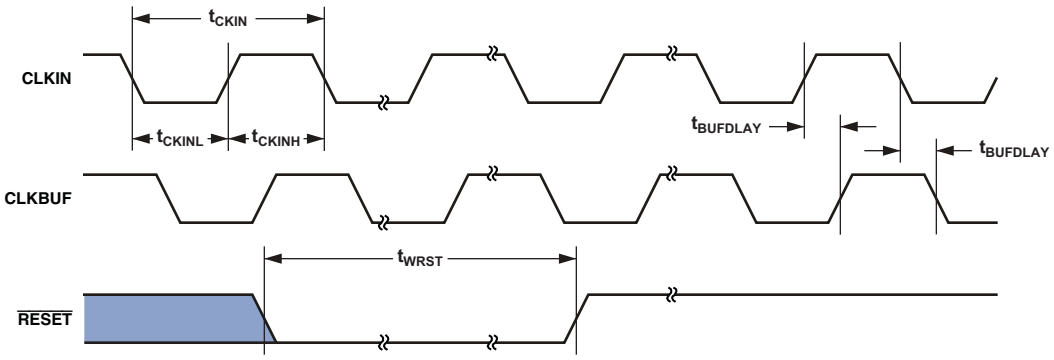

Figure 7. Clock and Reset Timing

Serial Ports

Table 21 through Table 25 and Figure 14 through Figure 18 describe serial port operations.

Table 21. Serial Ports—External Clock

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Timing Requirements						
t_{SFSE}	TFSx/RFSx Setup Before TSCLKx/RSCLKx[1]	3		3		ns
t_{HFSE}	TFSx/RFSx Hold After TSCLKx/RSCLKx[1]	3		3		ns
t_{SDRE}	Receive Data Setup Before RSCLKx[1]	3		3		ns
t_{HDRE}	Receive Data Hold After RSCLKx[1]	3.5		3		ns
t_{SCLKEW}	TSCLKx/RSCLKx Width	4.5		4.5		ns
t_{SCLKE}	TSCLKx/RSCLKx Period	$2 \times t_{SCLK}$		$2 \times t_{SCLK}$		ns
t_{SUDTE}	Start-Up Delay From SPORT Enable To First External TFSx[2]	$4 \times t_{TSCLKE}$		$4 \times t_{TSCLKE}$		ns
t_{SUDRE}	Start-Up Delay From SPORT Enable To First External RFSx[2]	$4 \times t_{RSCLKE}$		$4 \times t_{RSCLKE}$		ns
Switching Characteristics						
t_{DFSE}	TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[3]		10		10	ns
t_{HOFSE}	TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[1]	0		0		ns
t_{DDTE}	Transmit Data Delay After TSCLKx[1]		11		10	ns
t_{HDTE}	Transmit Data Hold After TSCLKx[1]	0		0		ns

[1] Referenced to sample edge.
[2] Verified in design but untested.
[3] Referenced to drive edge.

Table 22. Serial Ports—Internal Clock

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Timing Requirements						
t_{SFSI}	TFSx/RFSx Setup Before TSCLKx/RSCLKx[1]	11.5		9.6		ns
t_{HFSI}	TFSx/RFSx Hold After TSCLKx/RSCLKx[1]	–1.5		–1.5		ns
t_{SDRI}	Receive Data Setup Before RSCLKx[1]	11.5		11.3		ns
t_{HDRI}	Receive Data Hold After RSCLKx[1]	–1.5		–1.5		ns
Switching Characteristics						
t_{SCLKIW}	TSCLKx/RSCLKx Width	7		8		ns
t_{DFSI}	TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[2]		4		3	ns
t_{HOFSI}	TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[1]	–2		–2		ns
t_{DDTI}	Transmit Data Delay After TSCLKx[1]		4		3	ns
t_{HDTI}	Transmit Data Hold After TSCLKx[1]	–1.8		–1.5		ns

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

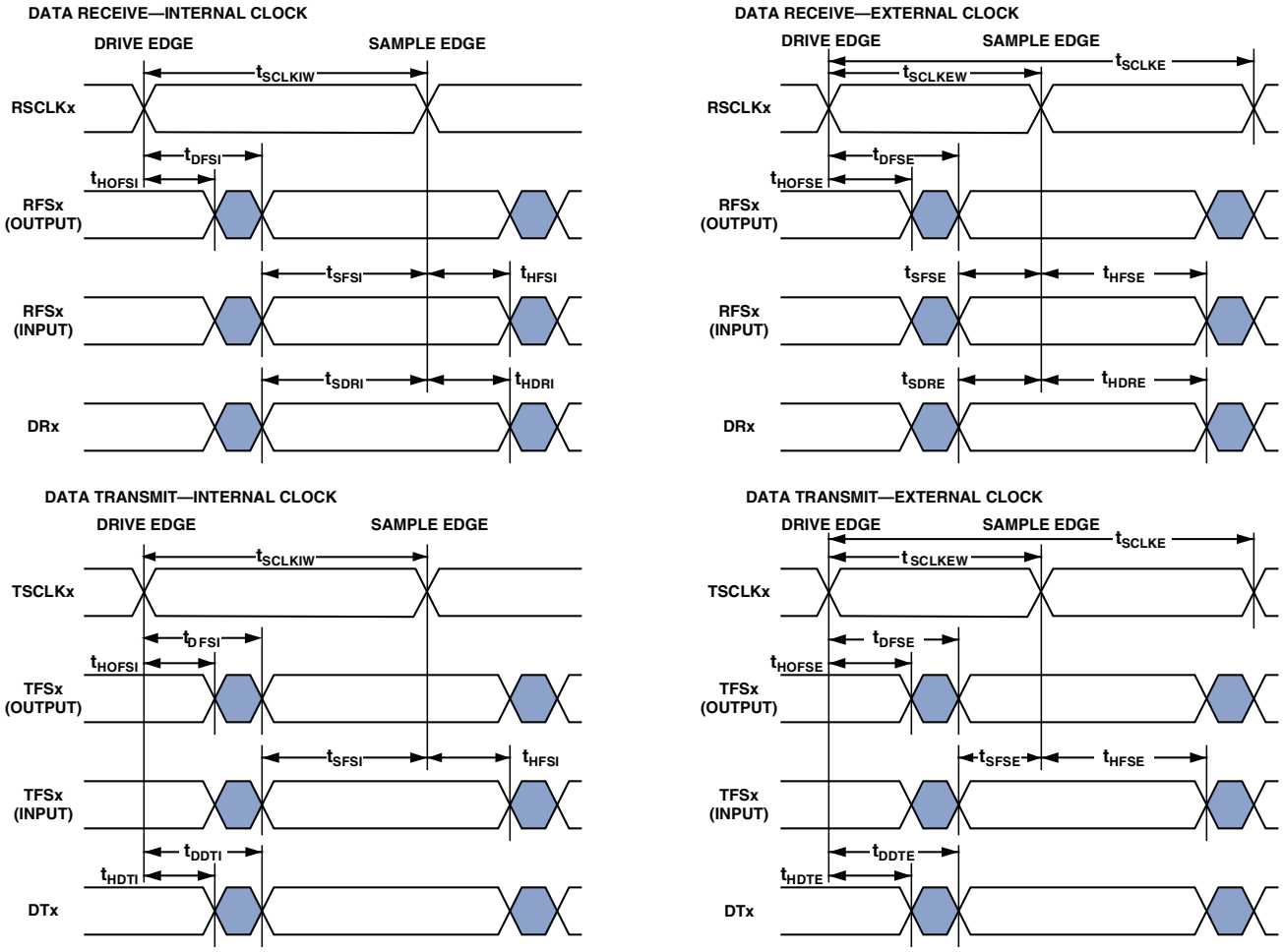

Figure 14. Serial Ports

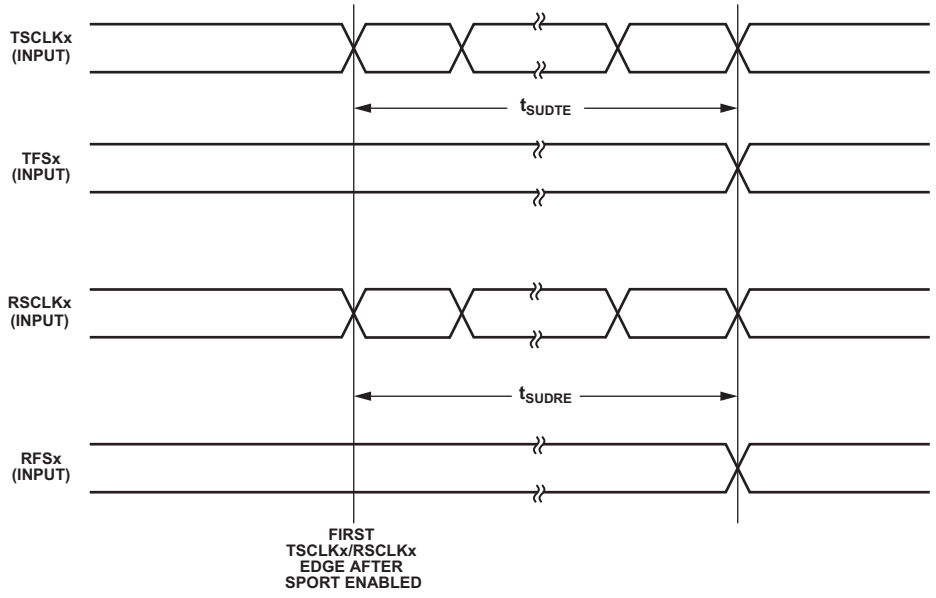

Figure 15. Serial Port Start Up with External Clock and Frame Sync

Table 23. Serial Ports—Enable and Three-State

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Switching Characteristics						
t_{DTENE}	Data Enable Delay from External TSCLKx[1]	0		0		ns
t_{DDTTE}	Data Disable Delay from External TSCLKx[1]		t_{SCLK} + 1		t_{SCLK} + 1	ns
t_{DTENI}	Data Enable Delay from Internal TSCLKx[1]	–2		–2		ns
t_{DDTTI}	Data Disable Delay from Internal TSCLKx[1]		t_{SCLK} + 1		t_{SCLK} + 1	ns

[1] Referenced to drive edge.

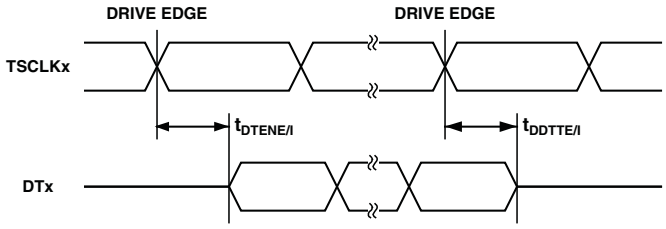

Figure 16. Serial Ports — Enable and Three-State

Table 24. Serial Ports—External Late Frame Sync

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Switching Characteristics						
$t_{DDTLFSE}$	Data Delay from Late External TFSx or External RFSx in multi-channel mode with MFD = 0[1, 2]		12		10	ns
$t_{DTENLFSE}$	Data Enable from External RFSx in multi-channel mode with MFD = 0[1, 2]	0		0		ns

[1] When in multi-channel mode, TFSx enable and TFSx valid follow $t_{DTENLFSE}$ and $t_{DDTLFSE}$.
[2] If external RFSx/TFSx setup to RSCLKx/TSCLKx > $t_{SCLKE}/2$ then $t_{DDTTE/I}$ and $t_{DTENE/I}$ apply, otherwise $t_{DDTLFSE}$ and $t_{DTENLFSE}$ apply.

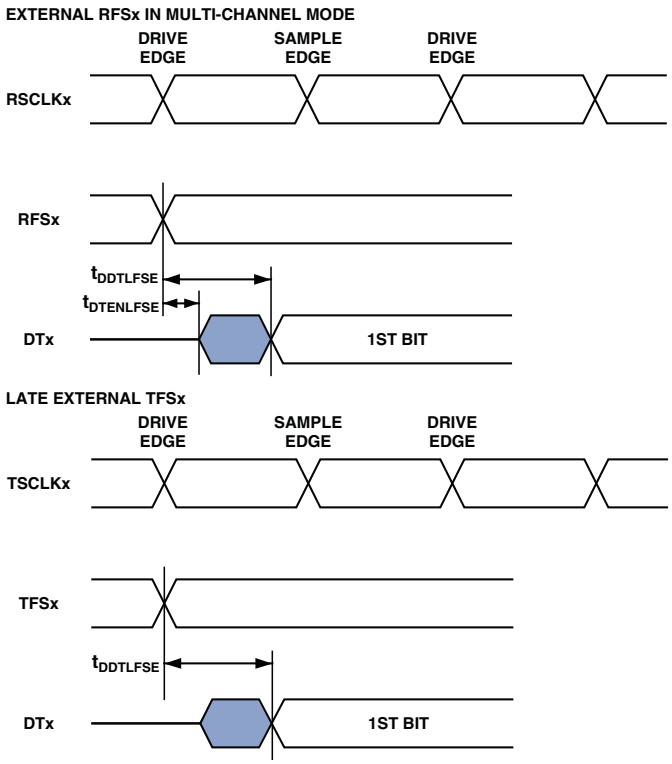

Figure 17. Serial Ports — External Late Frame Sync

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 27 and Figure 20 describe SPI port slave operations.

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

Parameter		V_{DDEXT} 1.8 V Nominal Min	V_{DDEXT} 1.8 V Nominal Max	V_{DDEXT} 2.5 V/3.3 V Nominal Min	V_{DDEXT} 2.5 V/3.3 V Nominal Max	Unit
Timing Requirements						
t_{SPICHS}	Serial Clock High Period	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLS}	Serial Clock Low Period	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLK}	Serial Clock Period	$4 \times t_{SCLK}$		$4 \times t_{SCLK}$		ns
t_{HDS}	Last SCK Edge to $\overline{SPI\_SS}$ Not Asserted	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPITDS}	Sequential Transfer Delay	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SDSCI}	$\overline{SPI\_SS}$ Assertion to First SCK Edge	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SSPID}	Data Input Valid to SCK Edge (Data Input Setup)	1.6		1.6		ns
t_{HSPID}	SCK Sampling Edge to Data Input Invalid	2		1.6		ns
Switching Characteristics						
t_{DSOE}	$\overline{SPI\_SS}$ Assertion to Data Out Active	0	12	0	10.3	ns
t_{DSDHI}	$\overline{SPI\_SS}$ Deassertion to Data High Impedance	0	11	0	9	ns
t_{DDSPID}	SCK Edge to Data Out Valid (Data Out Delay)		10		10	ns
t_{HDSPID}	SCK Edge to Data Out Invalid (Data Out Hold)	0		0		ns

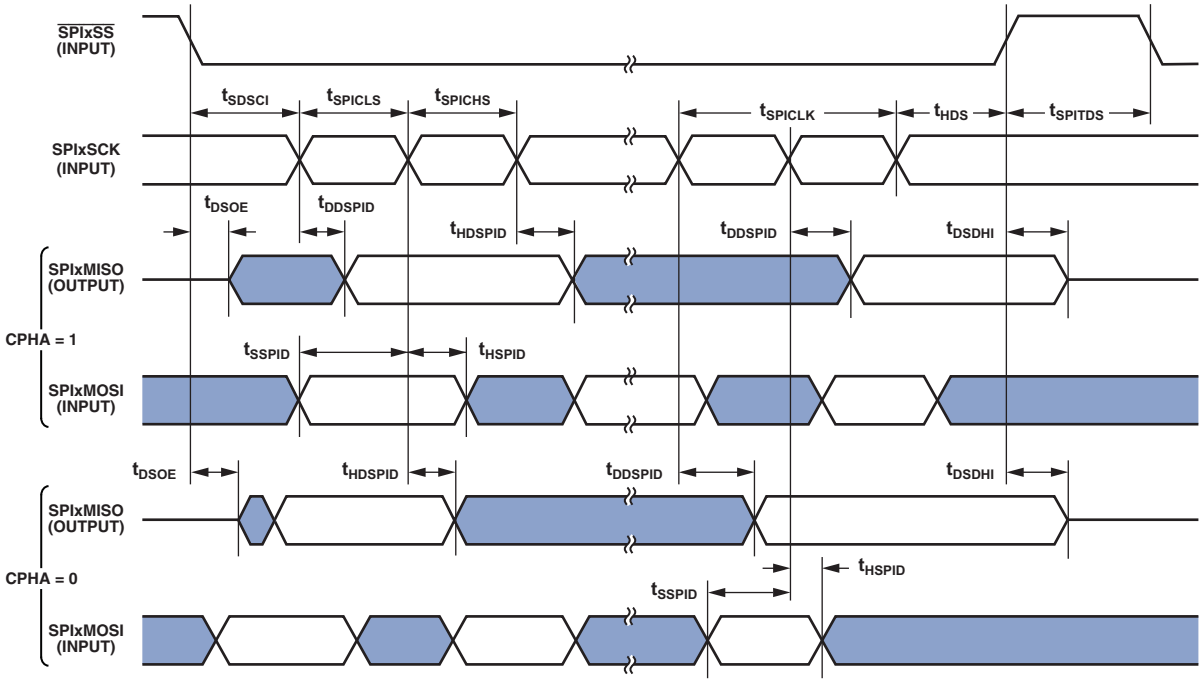

Figure 20. Serial Peripheral Interface (SPI) Port—Slave Timing

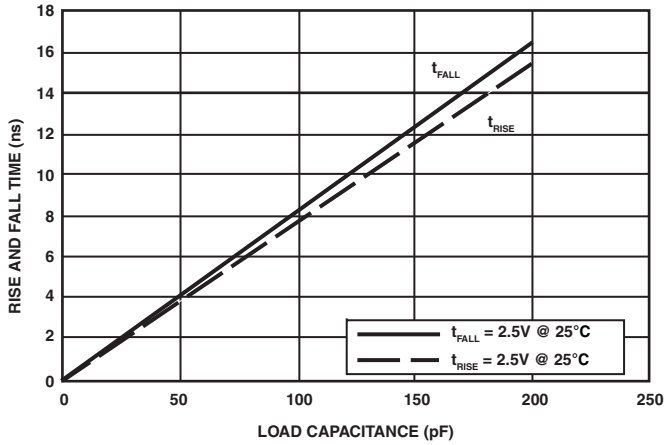

Figure 38. Driver Type A Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (2.5V V_{DDEXT})

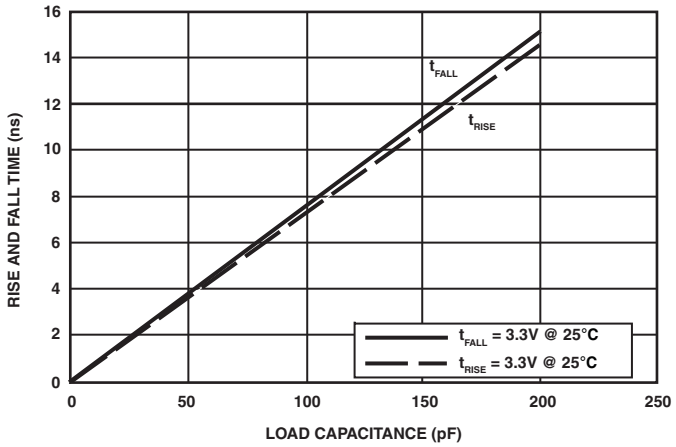

Figure 39. Driver Type A Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (3.3V V_{DDEXT})

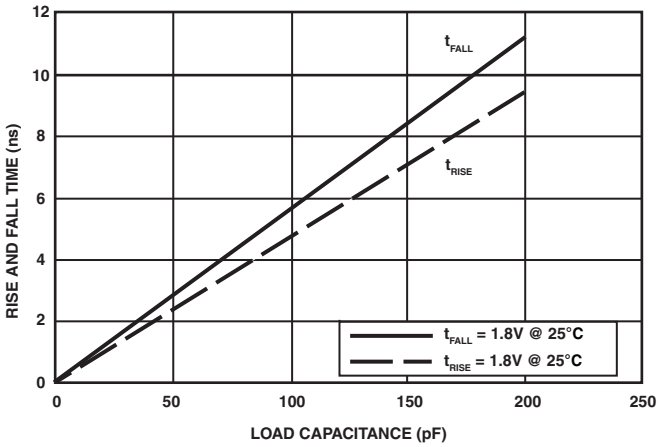

Figure 40. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (1.8V V_{DDEXT})

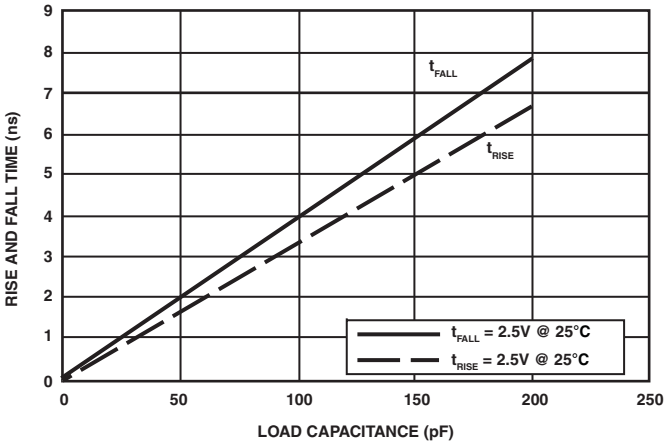

Figure 41. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (2.5V V_{DDEXT})

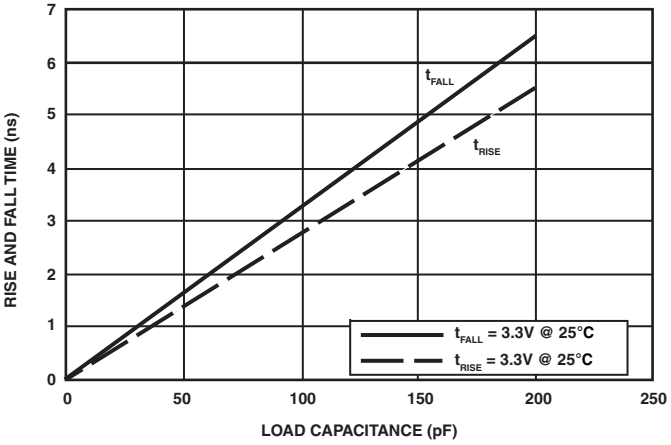

Figure 42. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (3.3V V_{DDEXT})

ENVIRONMENTAL CONDITIONS

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

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

where:

T_J = junction temperature (°C)

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

Ψ_{JT} = from Table 32

P_D = power dissipation (see Total Power Dissipation on Page 19 for the method to calculate P_D)

Table 32. Thermal Characteristics

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	23.5	°C/W
θ_{JMA}	1 linear m/s air flow	20.9	°C/W
θ_{JMA}	2 linear m/s air flow	20.2	°C/W
θ_{JB}		11.2	°C/W
θ_{JC}		9.5	°C/W
Ψ_{JT}	0 linear m/s air flow	0.21	°C/W
Ψ_{JT}	1 linear m/s air flow	0.36	°C/W
Ψ_{JT}	2 linear m/s air flow	0.43	°C/W

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 heat sink is required.

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

In Table 32, 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.

64-LEAD LFCSP LEAD ASSIGNMENT

Table 33 lists the LFCSP leads by signal mnemonic. Table 34 lists the LFCSP by lead number.

Table 33. 64-Lead LFCSP Lead Assignment (Alphabetical by Signal)

Signal	Lead No.	Signal	Lead No.	Signal	Lead No.	Signal	Lead No.
BMODE0	29	PF7	7	PG6	38	TDO	23
BMODE1	28	PF8	10	PG7	39	TMS	21
BMODE2	27	PF9	11	PG8	42	$\overline{\text{TRST}}$	20
EXTCLK/SCLK	57	PF10	12	PG9	43	V_{DDEXT}	3
CLKIN	61	PF11	13	PG10	44	V_{DDEXT}	14
$\overline{\text{EMU}}$	19	PF12	15	PG11	45	V_{DDEXT}	25
EXT_WAKE	51	PF13	16	PG12	47	V_{DDEXT}	35
GND	30	PF14	17	PG13	48	V_{DDEXT}	46
$\overline{\text{NMI}}$	54	PF15	18	PG14	49	V_{DDEXT}	58
PF0	63	$\overline{\text{PG}}$	52	PG15	50	V_{DDINT}	8
PF1	64	PG0	31	PPI_CLK	56	V_{DDINT}	9
PF2	1	PG1	32	$\overline{\text{RESET}}$	53	V_{DDINT}	26
PF3	2	PG2	33	SCL	60	V_{DDINT}	40
PF4	4	PG3	34	SDA	59	V_{DDINT}	41
PF5	5	PG4	36	TCK	24	V_{DDINT}	55
PF6	6	PG5	37	TDI	22	XTAL	62
						GND*	65

* Lead no. 65 is the GND supply (see Figure 43 and Figure 44) for the processor (6.2 mm × 6.2 mm); this pad **must** connect to GND.

Table 34. 64-Lead LFCSP Lead Assignment (Numerical by Lead Number)

Lead No.	Signal	Lead No.	Signal	Lead No.	Signal	Lead No.	Signal
1	PF2	17	PF14	33	PG2	49	PG14
2	PF3	18	PF15	34	PG3	50	PG15
3	V_{DDEXT}	19	$\overline{\text{EMU}}$	35	V_{DDEXT}	51	EXT_WAKE
4	PF4	20	$\overline{\text{TRST}}$	36	PG4	52	$\overline{\text{PG}}$
5	PF5	21	TMS	37	PG5	53	$\overline{\text{RESET}}$
6	PF6	22	TDI	38	PG6	54	$\overline{\text{NMI}}$
7	PF7	23	TDO	39	PG7	55	V_{DDINT}
8	V_{DDINT}	24	TCK	40	V_{DDINT}	56	PPI_CLK
9	V_{DDINT}	25	V_{DDEXT}	41	V_{DDINT}	57	EXTCLK/SCLK
10	PF8	26	V_{DDINT}	42	PG8	58	V_{DDEXT}
11	PF9	27	BMODE2	43	PG9	59	SDA
12	PF10	28	BMODE1	44	PG10	60	SCL
13	PF11	29	BMODE0	45	PG11	61	CLKIN
14	V_{DDEXT}	30	GND	46	V_{DDEXT}	62	XTAL
15	PF12	31	PG0	47	PG12	63	PF0
16	PF13	32	PG1	48	PG13	64	PF1
						65	GND*

* Pin no. 65 is the GND supply (see Figure 43 and Figure 44) for the processor (6.2 mm × 6.2 mm); this pad **must** connect to GND.