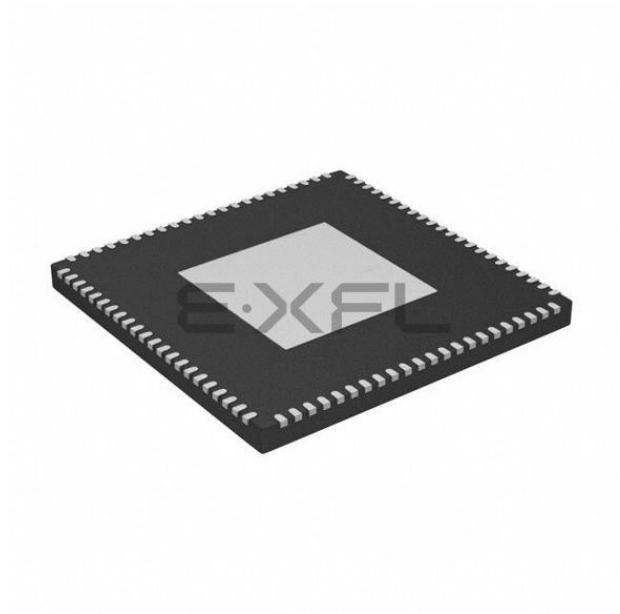


Welcome to E-XFL.COM

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	CAN, EBI/EMI, I²C, IrDA, PPI, SPI, SPORT, UART/USART
Clock Rate	400MHz
Non-Volatile Memory	External
On-Chip RAM	68kB
Voltage - I/O	1.8V, 2.5V, 3.3V
Voltage - Core	1.31V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	88-VFQFN Exposed Pad, CSP
Supplier Device Package	88-LFCSP-VQ (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf504kcpz-4

Booting

The processor contains a small on-chip boot kernel, which configures the appropriate peripheral for booting. If the processor is configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM. For more information, see Booting Modes on Page 16.

Event Handling

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

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

Each event type has an associated register to hold the return address and an associated return-from-event instruction. When an event is triggered, an interrupt service routine (ISR) must save the state of the processor to the supervisor stack.

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

Core Event Controller (CEC)

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

Table 2. Core Event Controller (CEC)

Priority (0 is Highest)	Event Class	EVT Entry
0	Emulation/Test Control	EMU
1	Reset	RST
2	Nonmaskable Interrupt	NMI
3	Exception	EVX
4	Reserved	—
5	Hardware Error	IVHW
6	Core Timer	IVTMR
7	General-Purpose Interrupt 7	IVG7
8	General-Purpose Interrupt 8	IVG8
9	General-Purpose Interrupt 9	IVG9
10	General-Purpose Interrupt 10	IVG10
11	General-Purpose Interrupt 11	IVG11
12	General-Purpose Interrupt 12	IVG12
13	General-Purpose Interrupt 13	IVG13
14	General-Purpose Interrupt 14	IVG14
15	General-Purpose Interrupt 15	IVG15

System Interrupt Controller (SIC)

The system interrupt controller provides the mapping and routing of events from the many peripheral interrupt sources to the prioritized general-purpose interrupt inputs of the CEC. Although the processor provides a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the interrupt assignment registers (SIC_IARx). Table 3 describes the inputs into the SIC and the default mappings into the CEC.

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

Peripheral Interrupt Source	General-Purpose Interrupt (at Reset)	Peripheral Interrupt ID	Default Core Interrupt ID	SIC Registers	
MDMA Stream 1	IVG13	43	6	IAR5	IMASK1, ISR1, IWR1
Software Watchdog Timer	IVG13	44	6	IAR5	IMASK1, ISR1, IWR1
Port H Interrupt A	IVG13	45	6	IAR5	IMASK1, ISR1, IWR1
Port H Interrupt B	IVG13	46	6	IAR5	IMASK1, ISR1, IWR1
ACM Status Interrupt	IVG7	47	0	IAR5	IMASK1, ISR1, IWR1
ACM Interrupt	IVG10	48	3	IAR6	IMASK1, ISR1, IWR1
Reserved	—	49	—	IAR6	IMASK1, ISR1, IWR1
Reserved	—	50	—	IAR6	IMASK1, ISR1, IWR1
PWM0 Trip Interrupt	IVG10	51	3	IAR6	IMASK1, ISR1, IWR1
PWM0 Sync Interrupt	IVG10	52	3	IAR6	IMASK1, ISR1, IWR1
PWM1 Trip Interrupt	IVG10	53	3	IAR6	IMASK1, ISR1, IWR1
PWM1 Sync Interrupt	IVG10	54	3	IAR6	IMASK1, ISR1, IWR1
RSI Mask 1 Interrupt	IVG10	55	3	IAR6	IMASK1, ISR1, IWR1
Reserved	—	56 through 63	—	—	IMASK1, ISR1, IWR1

Event Control

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

- CEC interrupt latch register (ILAT)—Indicates when events have been latched. The appropriate bit is set when the processor has latched the event and is cleared when the event has been accepted into the system. This register is updated automatically by the controller, but it may be written only when its corresponding IMASK bit is cleared.
- CEC interrupt mask register (IMASK)—Controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event is unmasked and is processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, preventing the processor from servicing the event even though the event may be latched in the ILAT register. This register may be read or written while in supervisor mode. (Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.)
- CEC interrupt pending register (IPEND)—The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing three pairs of 32-bit interrupt control and status registers. Each register contains a bit, corresponding to each of the peripheral interrupt events shown in Table 3 on Page 7.

- SIC interrupt mask registers (SIC_IMASKx)—Control the masking and unmasking of each peripheral interrupt event. When a bit is set in these registers, the corresponding peripheral event is unmasked and is forwarded to the CEC when asserted. A cleared bit in these registers masks the corresponding peripheral event, preventing the event from propagating to the CEC.
- SIC interrupt status registers (SIC_ISRx)—As multiple peripherals can be mapped to a single event, these registers allow the software to determine which peripheral event source triggered the interrupt. A set bit indicates that the peripheral is asserting the interrupt, and a cleared bit indicates that the peripheral is not asserting the event.
- SIC interrupt wakeup enable registers (SIC_IWRx)—By enabling the corresponding bit in these registers, a peripheral can be configured to wake up the processor should the core be idled or in sleep mode when the event is generated. For more information, see Dynamic Power Management on Page 13.

Because multiple interrupt sources can map to a single general-purpose interrupt, multiple pulse assertions can occur simultaneously, before or during interrupt processing for an interrupt event already detected on this interrupt input. The IPEND register contents are monitored by the SIC as the interrupt acknowledgement.

The appropriate ILAT register bit is set when an interrupt rising edge is detected (detection requires two core clock cycles). The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC recognizes and queues the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the general-purpose interrupt to the IPEND output asserted is three core clock cycles; however, the latency can be much higher, depending on the activity within and the state of the processor.

- DMA operations with single-cycle overhead—Each SPORT can automatically receive and transmit multiple buffers of memory data. The processor can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts—Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer, or buffers, through DMA.
- Multichannel capability—Each SPORT supports 128 channels out of a 1024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

SERIAL PERIPHERAL INTERFACE (SPI) PORTS

The ADSP-BF50x processors have 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 MOSI (Master Output-Slave Input) and MISO (Master Input-Slave Output) 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 three SPI chip select output pins ($\overline{\text{SPIx\_SEL3–1}}$) 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.

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

The SPI port's clock rate is calculated as:

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

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

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

UART PORTS (UARTS)

The ADSP-BF50x Blackfin processors provide two full-duplex universal asynchronous receiver/transmitter (UART) ports. Each UART port provides a simplified UART interface to other peripherals or hosts, enabling full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port 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 7 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.

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

PARALLEL PERIPHERAL INTERFACE (PPI)

The processor provides a parallel peripheral interface (PPI) that can connect directly to parallel A/D and D/A 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.
- Frame capture mode—Frame syncs are outputs from the PPI, but data are inputs.
- Output mode—Frame syncs and data are outputs from the PPI.

Input Mode

Input mode is intended for ADC applications, as well as video communication with hardware signaling. In its simplest form, PPI_FS1 is an external frame sync input that controls when to read data. The PPI_DELAY MMR allows for a delay (in PPI_CLK cycles) between reception of this frame sync and the initiation of data reads. The number of input data samples is user programmable and defined by the contents of the PPI_COUNT register. The PPI supports 8-bit and 10-bit through 16-bit data, programmable in the PPI_CONTROL register.

Frame Capture Mode

Frame capture mode allows the video source(s) to act as a slave (for frame capture for example). The ADSP-BF50x processors control when to read from the video source(s). PPI_FS1 is an HSYNC output and PPI_FS2 is a VSYNC output.

Output Mode

Output mode is used for transmitting video or other data with up to three output frame syncs. Typically, a single frame sync is appropriate for data converter applications, whereas two or three frame syncs could be used for sending video with hardware signaling.

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
- Vertical blanking only mode
- Entire field mode

Active Video 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. The PPI does not read in any data between the end of active video (EAV) and start of active video (SAV) preamble symbols, or any data present during the vertical blanking intervals. In this mode, the control byte sequences are not stored to memory; they are filtered by the PPI. After synchronizing to the start of Field 1, the PPI ignores incoming samples until it sees an SAV code. The user specifies the number of active video lines per frame (in PPI_COUNT register).

Vertical Blanking Interval 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. This includes active video, control preamble sequences, and ancillary data that may be embedded in horizontal and vertical blanking intervals. Data transfer starts immediately after synchronization to Field 1. Data is transferred to or from the synchronous channels through eight DMA engines that work autonomously from the processor core.

RSI INTERFACE

The removable storage interface (RSI) controller acts as the host interface for multimedia cards (MMC), secure digital memory cards (SD), secure digital input/output cards (SDIO), and CE-ATA hard disk drives. The following list describes the main features of the RSI controller.

- Support for a single MMC, SD memory, SDIO card or CE-ATA hard disk drive
- Support for 1-bit and 4-bit SD modes
- Support for 1-bit, 4-bit, and 8-bit MMC modes
- Support for 4-bit and 8-bit CE-ATA hard disk drives
- A ten-signal external interface with clock, command, and up to eight data lines
- Card detection using one of the data signals
- Card interface clock generation from SCLK
- SDIO interrupt and read wait features
- CE-ATA command completion signal recognition and disable

CONTROLLER AREA NETWORK (CAN) INTERFACE

The ADSP-BF50x processors provide a CAN controller that is a communication controller implementing the Controller Area Network (CAN) V2.0B protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. CAN 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 CAN controller is based on a 32-entry mailbox RAM and supports both the standard and extended identifier (ID) message formats specified in the CAN protocol specification, revision 2.0, part B.

- Boot from serial SPI memory, EEPROM or flash (BMODE = 0x3)—8-, 16-, 24-, or 32-bit addressable devices are supported. The processor uses the PF13 GPIO pin to select a single SPI EEPROM/flash device (connected to the SPI0 interface) and submits a read command and successive address bytes (0x00) until a valid 8-, 16-, 24-, or 32-bit addressable device is detected. Pull-up resistors are required on the SPI0_SEL1 and MISO pins. By default, a value of 0x85 is written to the SPI_BAUD register.
- Boot from SPI host device (BMODE = 0x4)—The processor operates in SPI slave mode and is configured to receive the bytes of the LDR file from an SPI host (master) agent. The HWAIT signal must be interrogated by the host before every transmitted byte. A pull-up resistor is required on the SPI0_SS input. A pull-down on the serial clock (SCK) may improve signal quality and booting robustness.
- 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 UART0 host on Port G (BMODE = 0x7)—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 detection, the UART expects an "@" (0x40) character (eight bits data, one start bit, one stop bit, no parity bit) on the UA0_RX pin to determine the bit rate. The UART then replies with an acknowledgement composed of 4 bytes (0xBF, the value of UART0_DLL, the value of UART0_DLH, then 0x00). The host can then download the boot stream. The processor deasserts the UA0_RTS output to hold off the host; UA0_CTS functionality is not enabled at boot time.

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.

The boot kernel differentiates between a regular hardware reset and a wakeup-from-hibernate event to speed up booting in the later case. Bits 6-4 in the system reset configuration (SYSCR) register can be used to bypass the pre-boot routine and/or boot kernel in case of a software reset. They can also be used to simulate a wakeup-from-hibernate boot in the software reset case.

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

Table 11. Processor—Signal Descriptions (Continued)

Signal Name	Type	Function	Driver Type
PG14/*UA0_RTS*/SD_D6/TMR0/PPI_FS1/CUD1	I/O	GPIO/*UART0 RTS*/SD Data 6/Timer0/PPI FS1/Count Up Dir 1	C
PG15/*UA0_CTS*/SD_D7/TMR1/PPI_FS2/CDG1	I/O	GPIO/*UART0 CTS*/SD Data 7/Timer1/PPI FS2/Count Down Dir 1	C
Port H: *GPIO and Multiplexed Peripherals*			
PH0/*ACM_A2*/DT1PRI/*SPI0_SEL3*/WAKEUP	I/O	GPIO/*ADC CM A2*/SPORT1 TX Pri Data/SPI0 Slave Select 3/Wake-up Input	C
PH1/*ACM_A1*/TFS1/*SPI1_SEL3*/TACLK3	I/O	GPIO/*ADC CM A1*/SPORT1 TX Frame Sync/SPI1 Slave Select 3/Alt Timer CLK 3	C
PH2/*ACM_A0*/TSCLK1/*SPI1_SEL2*/TACI7	I/O	GPIO/*ADC CM A0*/SPORT1 TX Serial CLK/SPI1 Slave Select 2/Alt Capture In 7	C
TWI (2-Wire Interface) Port			
SCL	I/O 5 V	TWI Serial Clock (This signal is an open-drain output and requires a pull-up resistor. Consult version 2.1 of the I^2C specification for the proper resistor value.)	D
SDA	I/O 5 V	TWI Serial Data (This signal is an open-drain output and requires a pull-up resistor. Consult version 2.1 of the I^2C specification for the proper resistor value.)	D
JTAG Port			
TCK	I	JTAG CLK	
TDO	O	JTAG Serial Data Out	C
TDI	I	JTAG Serial Data In	
TMS	I	JTAG Mode Select	
$\overline{\text{TRST}}$	I	JTAG Reset (This signal should be pulled low if the JTAG port is not used.)	
$\overline{\text{EMU}}$	O	Emulation Output	C
Clock			
CLKIN	I	CLK/Crystal In	
XTAL	O	Crystal Output	
EXTCLK	O	Clock Output	B
Mode Controls			
$\overline{\text{RESET}}$	I	Reset	
$\overline{\text{NMI}}$	I	Nonmaskable Interrupt (This signal should be pulled high when not used.)	
BMODE2–0	I	Boot Mode Strap 2-0	
ADSP-BF50x Voltage Regulation I/F			
EXT_WAKE	O	Wake up Indication	C
$\overline{\text{PG}}$	I	Power Good	
Power Supplies		**ALL SUPPLIES MUST BE POWERED** See Operating Conditions on Page 26.	
V_{DDEXT}	P	I/O Power Supply	
V_{DDINT}	P	Internal Power Supply	
$V_{DDFLASH}$	P	Flash Memory Power Supply	
GND	G	Ground for All Supplies	

SPECIFICATIONS

Specifications are subject to change without notice.

OPERATING CONDITIONS

Parameter		Conditions	Min	Nominal	Max	Unit
V_{DDINT}	Internal Supply Voltage	Industrial Models	1.14		1.47	V
	Internal Supply Voltage	Commercial Models	1.10		1.47	V
	Internal Supply Voltage	Automotive Models	1.33		1.47	V
V_{DDEXT}[1, 2]	External Supply Voltage	1.8 V I/O, ADSP-BF504, Nonautomotive and Non Flash Models	1.7	1.8	1.9	V
	External Supply Voltage	2.5 V I/O, ADSP-BF504, Nonautomotive and Non Flash Models	2.25	2.5	2.75	V
	External Supply Voltage	3.3 V I/O, ADSP-BF50x, All Models	2.7	3.3	3.6	V
$V_{DDFLASH}$[1, 3]	Flash Memory Supply Voltage		1.7	1.8	2.0	V
V_{IH}	High Level Input Voltage[4, 5]	V_{DDEXT} = 1.90 V	1.2			V
	High Level Input Voltage[4, 6]	V_{DDEXT} = 2.75 V	1.7			V
	High Level Input Voltage[4, 6]	V_{DDEXT} = 3.6 V	2.0			V
V_{IHTWI}	High Level Input Voltage[5]	V_{DDEXT} = 1.90 V/2.75 V/3.6 V	0.7 × V_{BUSTWI}[7, 8]		V_{BUSTWI}[7, 8]	V
V_{IL}	Low Level Input Voltage[4, 5]	V_{DDEXT} = 1.7 V			0.6	V
	Low Level Input Voltage[4, 6]	V_{DDEXT} = 2.25 V			0.7	V
	Low Level Input Voltage[4, 6]	V_{DDEXT} = 3.0 V			0.8	V
V_{ILTWI}	Low Level Input Voltage[5]	V_{DDEXT} = minimum			0.3 × V_{BUSTWI}[8]	V
T_J	Junction Temperature	88-Lead LFCSP @ $T_{AMBIENT}$ = –40°C to +85°C	–40		+105	°C
	Junction Temperature	88-Lead LFCSP @ $T_{AMBIENT}$ = 0°C to +70°C	0		+90	°C
	Junction Temperature	120-Lead LQFP @ $T_{AMBIENT}$ = –40°C to +85°C	–40		+105	°C
	Junction Temperature	120-Lead LQFP @ $T_{AMBIENT}$ = 0°C to +70°C	0		+90	°C
	Junction Temperature	88-Lead LFCSP @ $T_{AMBIENT}$ = –40°C to +105°C	–40		+125	°C

[1] Must remain powered (even if the associated function is not used).
[2] 1.8 V and 2.5 V I/O are supported only on ADSP-BF504 nonautomotive models. All ADSP-BF50x flash and automotive models support 3.3 V I/O only.
[3] For ADSP-BF504, $V_{DDFLASH}$ pins should be connected to GND.
[4] Parameter value applies to all input and bidirectional pins, except SDA and SCL.
[5] Bidirectional pins (PF15–0, PG15–0, PH15–0) and input pins (TCK, TDI, TMS, $\overline{TRST}$, CLKIN, $\overline{RESET}$, $\overline{NMI}$, and BMODE2–0) of the ADSP-BF50x processors are 2.5 V tolerant (always accept up to 2.7 V maximum V_{IH}). Voltage compliance (on outputs, V_{OH}) is limited by the V_{DDEXT} supply voltage.
[6] Bidirectional pins (PF15–0, PG15–0, PH2–0) and input pins (TCK, TDI, TMS, $\overline{TRST}$, CLKIN, $\overline{RESET}$, $\overline{NMI}$, and BMODE2–0) of the ADSP-BF50x processors 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.
[7] The V_{IHTWI} min and max value vary with the selection in the TWI_DT field of the NONGPIO_DRIVE register. See V_{BUSTWI} min and max values in Table 13.
[8] SDA and SCL are pulled up to V_{BUSTWI}. See Table 13.

Table 13 shows settings for TWI_DT in the NONGPIO_DRIVE register. Set this register prior to using the TWI port.

Table 13. TWI_DT Field Selections and V_{DDEXT}/V_{BUSTWI}

TWI_DT	V_{DDEXT} Nominal	V_{BUSTWI} Minimum	V_{BUSTWI} Nominal	V_{BUSTWI} Maximum	Unit
000 (default)	3.3	2.97	3.3	3.63	V
001	1.8	1.7	1.8	1.98	V
010	2.5	2.97	3.3	3.63	V
011	1.8	2.97	3.3	3.63	V
100	3.3	4.5	5	5.5	V
101	1.8	2.25	2.5	2.75	V
110	2.5	2.25	2.5	2.75	V
111 (reserved)	—	—	—	—	—

Table 25. Clock Out Timing

Parameter		V_{DDEXT} = 1.8 V Min	V_{DDEXT} = 1.8 V Max	V_{DDEXT} = 2.5 V/3.3 V Min	V_{DDEXT} = 2.5 V/3.3 V Max	Unit
Switching Characteristics						
t_{SCLK}	CLKOUT[1] Period[2,3]	10		10		ns
t_{SCLKH}	CLKOUT[1] Width High	4		4		ns
t_{SCLKL}	CLKOUT[1] Width Low	4		4		ns

[1] The ADSP-BF504/ADSP-BF504F/ADSP-BF506F processor does not have a dedicated CLKOUT pin. Rather, the EXTCLK pin may be programmed to serve as CLKBUF or CLKOUT. This parameter applies when EXTCLK is programmed to output CLKOUT.

[2] The t_{SCLK} value is the inverse of the f_{SCLK} specification. Reduced supply voltages affect the best-case value of 10 ns listed here.

[3] The t_{SCLK} value does not account for the effects of jitter.

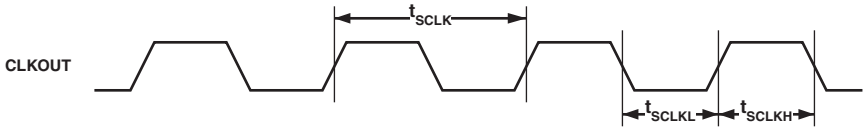
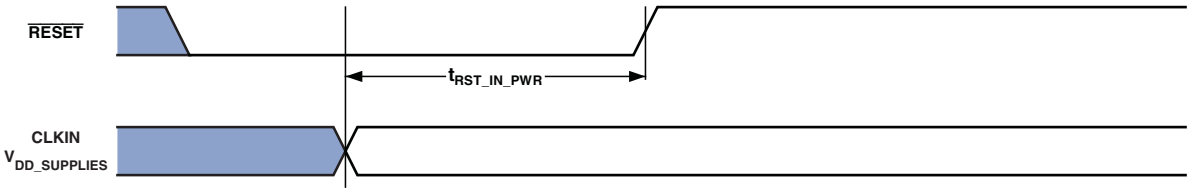


Figure 11. Clock Out Timing

Table 26. Power-Up Reset Timing

Parameter		Min	Max	Unit
Timing Requirement				
$t_{RST\_IN\_PWR}$	$\overline{RESET}$ Deasserted after the V_{DDINT}, V_{DDEXT}, $V_{DDFLASH}$, and CLKIN Pins are Stable and Within Specification	3500 × t_{CKIN}		ns



In Figure 12, $V_{DD\_SUPPLIES}$ is V_{DDINT}, V_{DDEXT}, and $V_{DDFLASH}$.

Figure 12. Power-Up Reset Timing

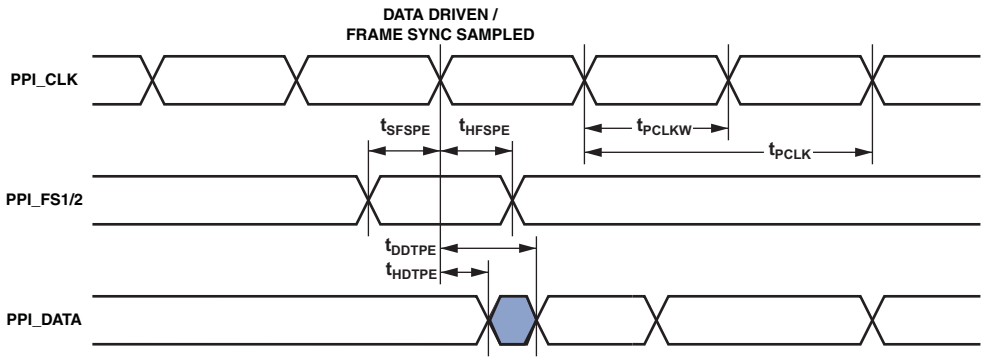


Figure 15. PPI GP Tx Mode with External Frame Sync Timing

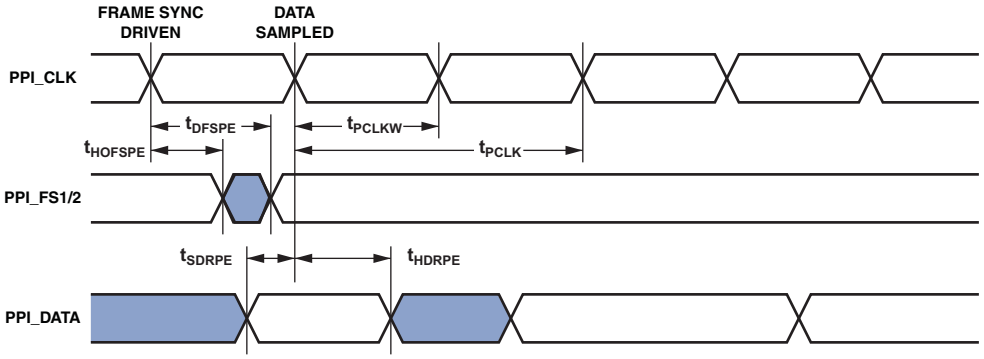


Figure 16. PPI GP Rx Mode with Internal Frame Sync Timing

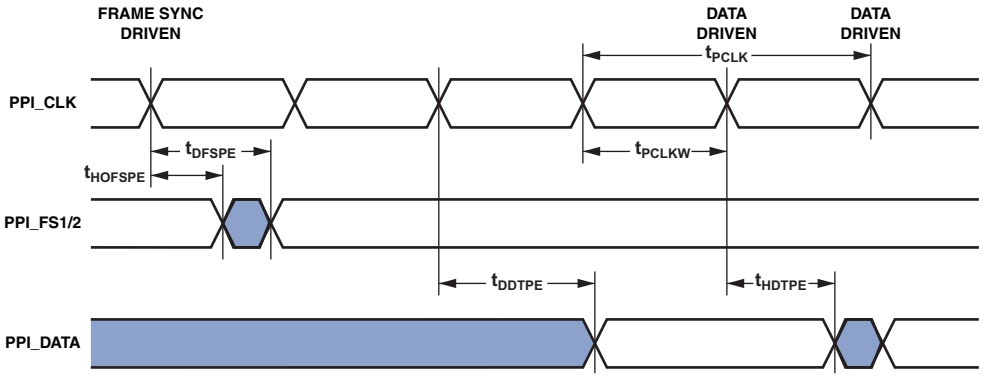


Figure 17. PPI GP Tx Mode with Internal Frame Sync Timing

RSI Controller Timing

Table 28 and Figure 18 describe RSI Controller Timing.
Table 29 and Figure 19 describe RSI controller (high speed) timing.

Table 28. RSI Controller Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{ISU}	Input Setup Time	5.75		ns
t_{IH}	Input Hold Time	2		ns
Switching Characteristics				
f_{PP}[1]	Clock Frequency Data Transfer Mode	0	25	MHz
f_{OD}	Clock Frequency Identification Mode	100[2]	400	kHz
t_{WL}	Clock Low Time	10		ns
t_{WH}	Clock High Time	10		ns
t_{TLH}	Clock Rise Time		10	ns
t_{THL}	Clock Fall Time		10	ns
t_{ODLY}	Output Delay Time During Data Transfer Mode		14	ns
t_{ODLY}	Output Delay Time During Identification Mode		50	ns

[1] $t_{PP} = 1/f_{PP}$.
[2] Specification can be 0 kHz, which means to stop the clock. The given minimum frequency range is for cases where a continuous clock is required.

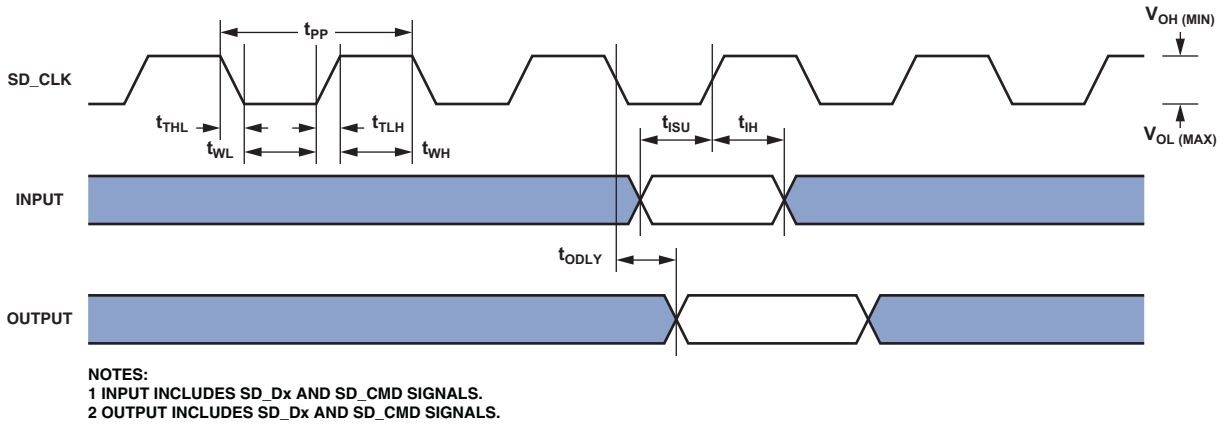


Figure 18. RSI Controller Timing

Table 33. Serial Ports — External Late Frame Sync

Parameter		V_{DDEXT} = 1.8 V Min	V_{DDEXT} = 1.8 V Max	V_{DDEXT} = 2.5 V/3.3 V Min	V_{DDEXT} = 2.5 V/3.3 V Max	Unit
Switching Characteristics						
$t_{DDTLFSE}$	Data Delay from Late External TFSx or External RFSx in Multi-channel Mode With MFD = 0[1, 2]		12.0		10.0	ns
$t_{DTENLFSE}$	Data Enable from External RFSx in Multi-channel Mode With MFD = 0[1, 2]	0.0		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.

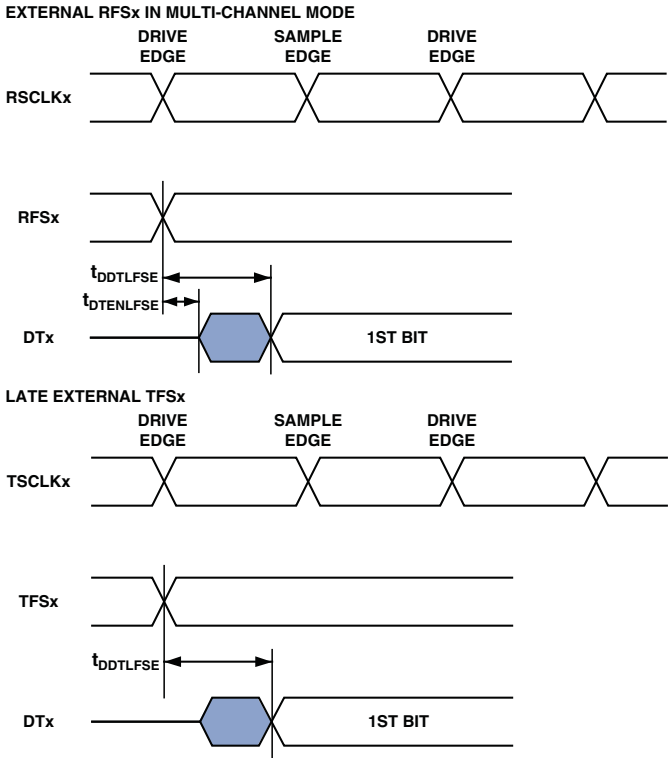


Figure 22. Serial Ports — External Late Frame Sync

Serial Peripheral Interface (SPI) Port—Master Timing

Table 34 and Figure 23 describe SPI port master operations.

Table 34. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter		V_{DDEXT} = 1.8 V Min	V_{DDEXT} = 1.8 V Max	V_{DDEXT} = 2.5 V/3.3 V Min	V_{DDEXT} = 2.5 V/3.3 V Max	Unit
Timing Requirements						
t_{SSPIDM}	Data Input Valid to SCK Edge (Data Input Setup)	11.6		9.6		ns
t_{HSPIDM}	SCK Sampling Edge to Data Input Invalid	–1.5		–1.5		ns
Switching Characteristics						
t_{SDSCIM}	$\overline{SPISELx}$ low to First SCK Edge	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPICHM}	Serial Clock High Period	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLM}	Serial Clock Low Period	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPICLK}	Serial Clock Period	$4 \times t_{SCLK} - 1.5$		$4 \times t_{SCLK} - 1.5$		ns
t_{HDSM}	Last SCK Edge to $\overline{SPISELx}$ High	$2 \times t_{SCLK} - 2.0$		$2 \times t_{SCLK} - 1.5$		ns
t_{SPITDM}	Sequential Transfer Delay	$2 \times t_{SCLK} - 1.5$		$2 \times t_{SCLK} - 1.5$		ns
$t_{DDSPIDM}$	SCK Edge to Data Out Valid (Data Out Delay)	0	6	0	6	ns
$t_{HDSPIDM}$	SCK Edge to Data Out Invalid (Data Out Hold)	–1.0		–1.0		ns

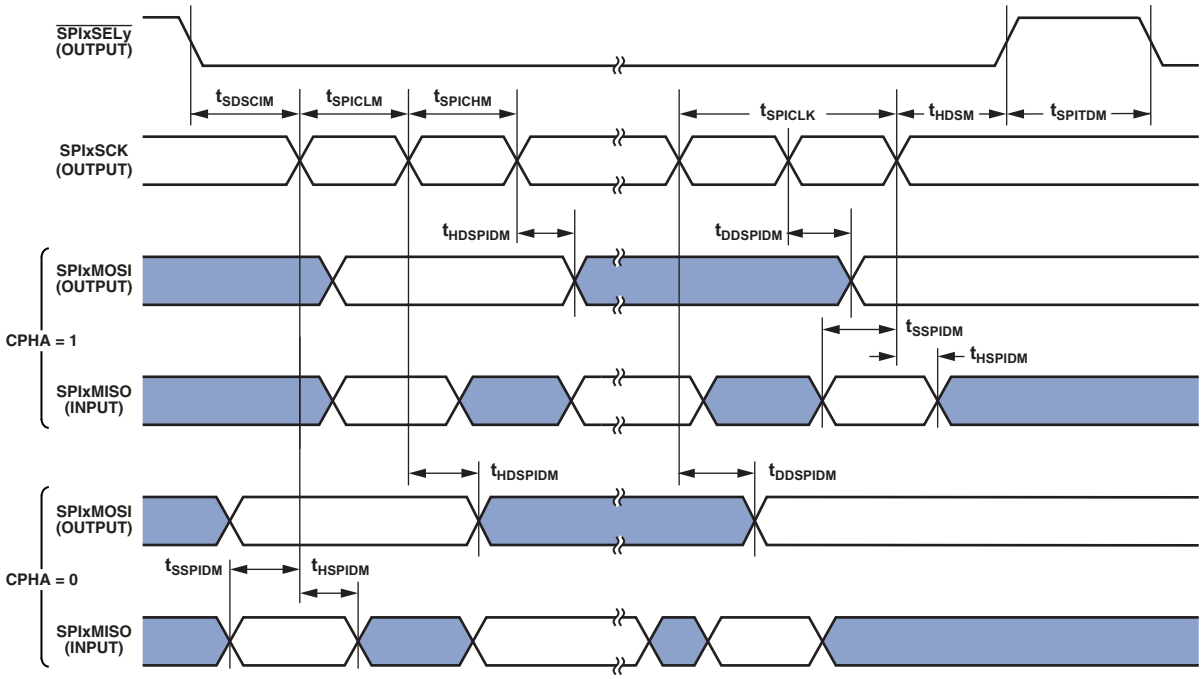


Figure 23. Serial Peripheral Interface (SPI) Port—Master Timing

PROCESSOR—OUTPUT DRIVE CURRENTS

Figure 32 through Figure 40 show typical current-voltage characteristics for the output drivers of the ADSP-BF50xF processors.

The curves represent the current drive capability of the output drivers. See Table 11 on Page 22 for information about which driver type corresponds to a particular pin.

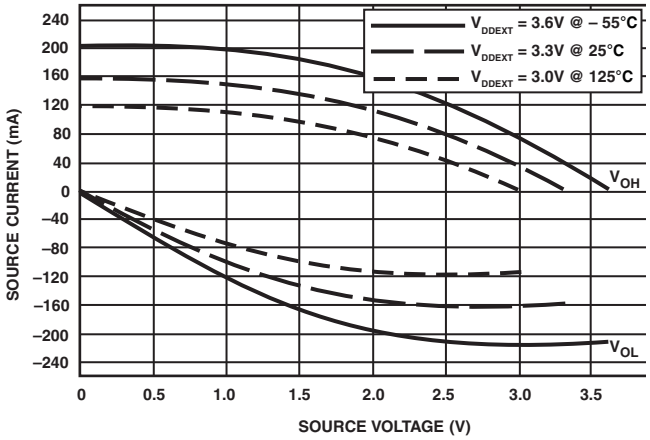


Figure 32. Driver Type B Current (3.3 V V_{DDEXT})

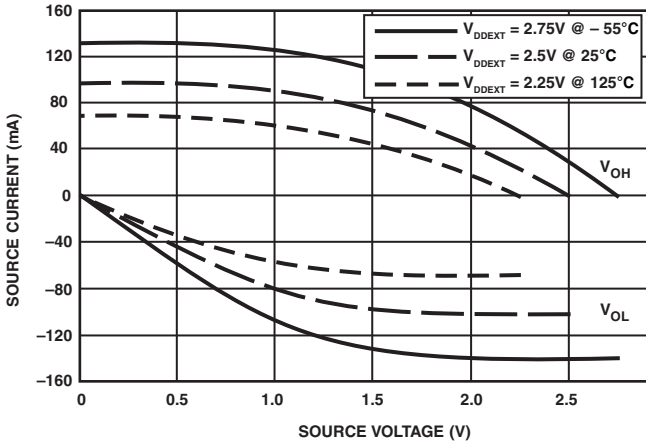


Figure 33. Driver Type B Current (2.5 V V_{DDEXT})

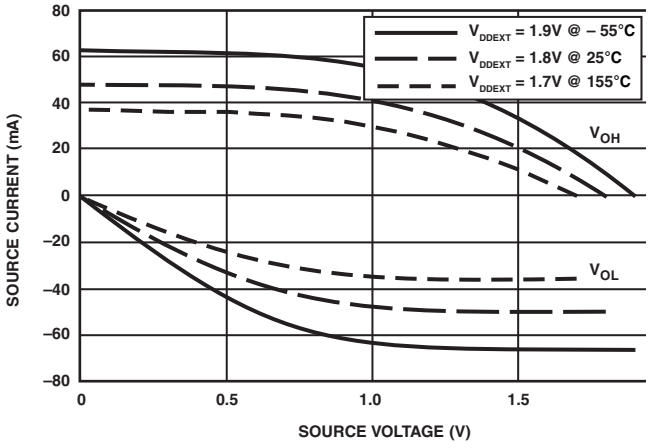


Figure 34. Driver Type B Current (1.8 V V_{DDEXT})

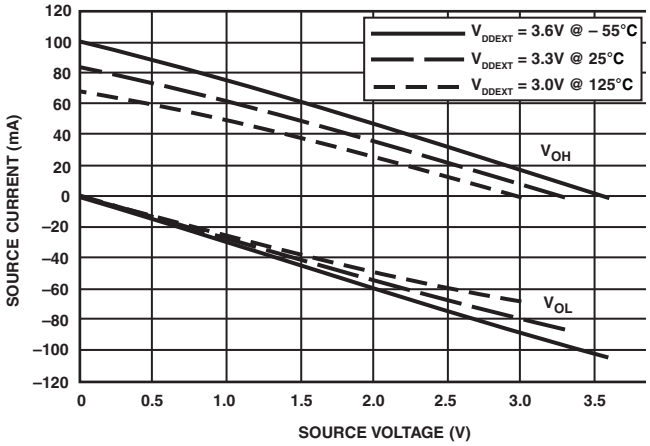


Figure 35. Driver Type C Current (3.3 V V_{DDEXT})

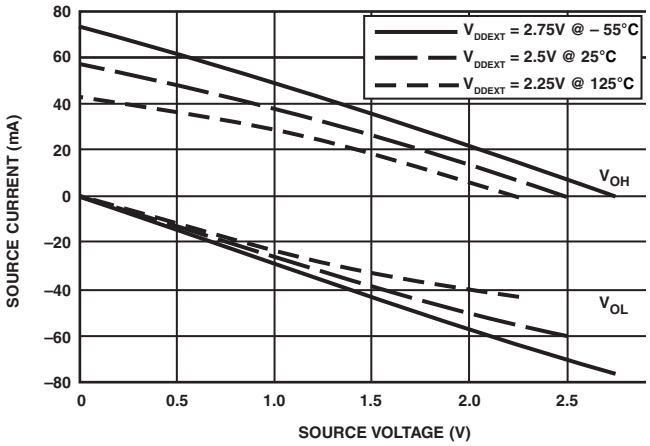


Figure 36. Drive Type C Current (2.5 V V_{DDEXT})

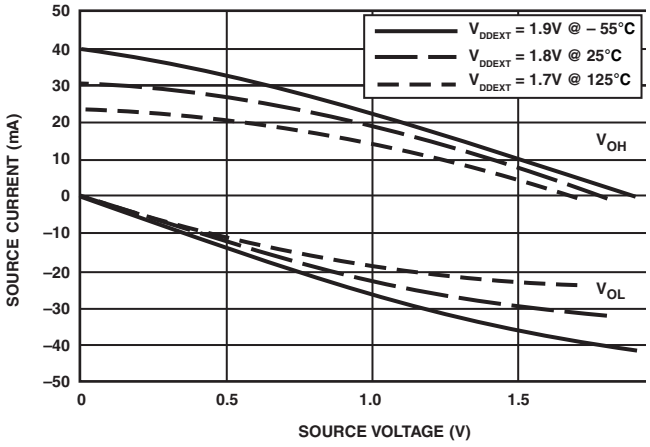


Figure 37. Driver Type C Current (1.8 V V_{DDEXT})

Output Disable Time Measurement

Output pins are considered to be disabled when they stop driving, go into a high impedance state, and start to decay from their output high or low voltage. The output disable time t_{DIS} is the difference between $t_{DIS\_MEASURED}$ and t_{DECAY} as shown on the left side of Figure 42.

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

The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load C_L and the load current I_L. This decay time can be approximated by the equation:

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

The time t_{DECAY} is calculated with test loads C_L and I_L, and with ΔV equal to 0.25 V for V_{DDEXT} (nominal) = 2.5 V/3.3 V and 0.15 V for V_{DDEXT} (nominal) = 1.8 V.

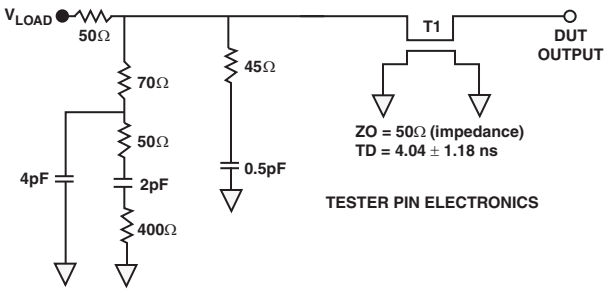
The time $t_{DIS\_MEASURED}$ is the interval from when the reference signal switches, to when the output voltage decays ΔV from the measured output high or output low voltage.

Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate t_{DECAY} using the equation given above. Choose ΔV to be the difference between the processor's output voltage and the input threshold for the device requiring the hold time. C_L is the total bus capacitance (per data line), and I_L is the total leakage or three-state current (per data line). The hold time will be t_{DECAY} plus the various output disable times as specified in the Processor—Timing Specifications on Page 33.

Capacitive Loading

Output delays and holds are based on standard capacitive loads of an average of 6 pF on all pins (see Figure 43). V_{LOAD} is equal to (V_{DDEXT}) /2. The graphs of Figure 44 through Figure 49 show how output rise time varies with capacitance. The delay and hold specifications given should be derated by a factor derived from these figures. The graphs in these figures may not be linear outside the ranges shown.



NOTES:
THE WORST CASE TRANSMISSION LINE DELAY IS SHOWN AND CAN BE USED FOR THE OUTPUT TIMING ANALYSIS TO REFELECT THE TRANSMISSION LINE EFFECT AND MUST BE CONSIDERED. THE TRANSMISSION LINE (TD), IS FOR LOAD ONLY AND DOES NOT AFFECT THE DATA SHEET TIMING SPECIFICATIONS.

ANALOG DEVICES RECOMMENDS USING THE IBIS MODEL TIMING FOR A GIVEN SYSTEM REQUIREMENT. IF NECESSARY, A SYSTEM MAY INCORPORATE EXTERNAL DRIVERS TO COMPENSATE FOR ANY TIMING DIFFERENCES.

Figure 43. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

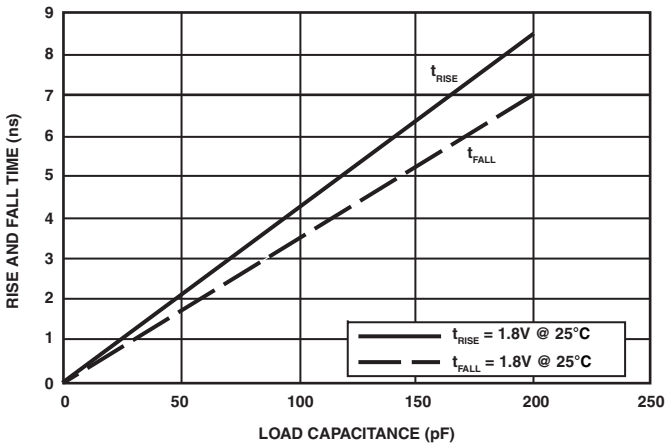


Figure 44. Driver Type B Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (1.8 V V_{DDEXT})

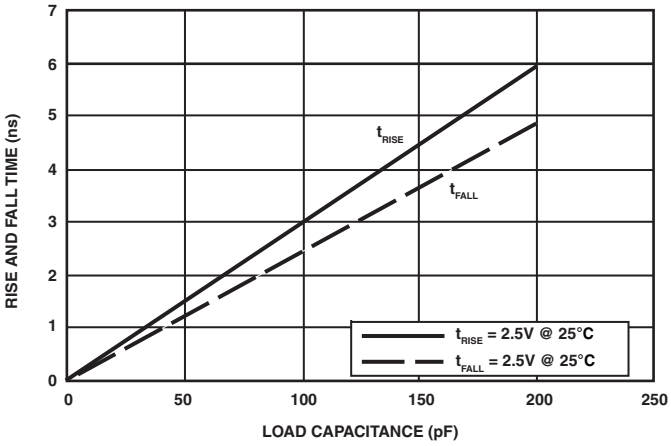


Figure 45. Driver Type B Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (2.5 V V_{DDEXT})

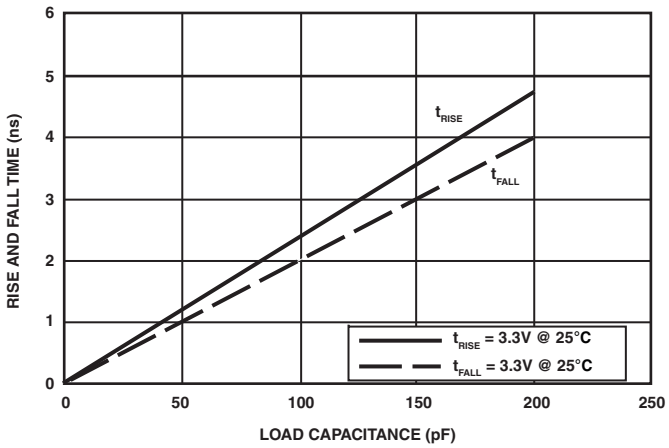


Figure 46. Driver Type B Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (3.3 V V_{DDEXT})

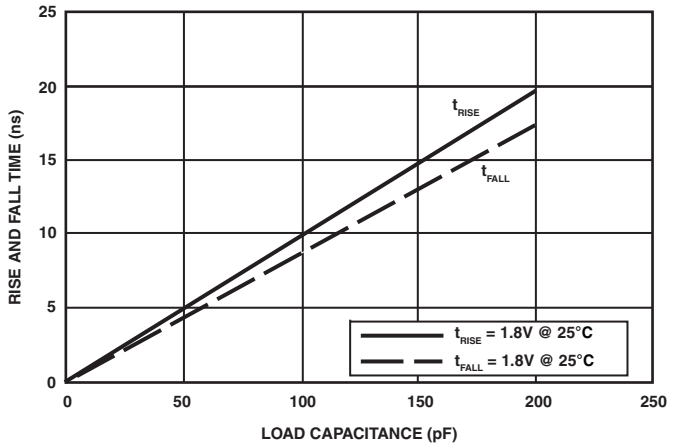


Figure 47. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (1.8 V V_{DDEXT})

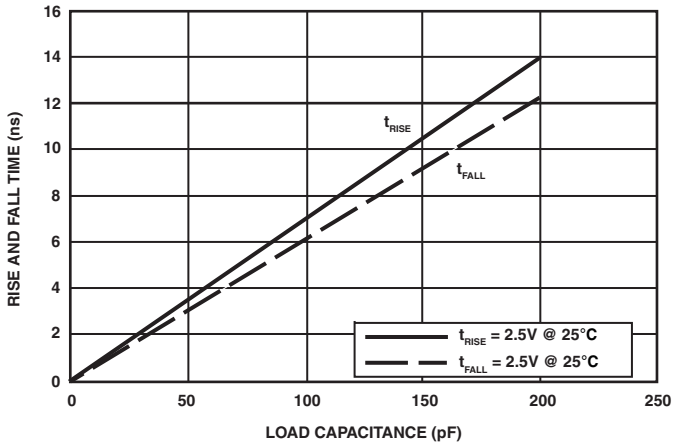


Figure 48. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (2.5 V V_{DDEXT})

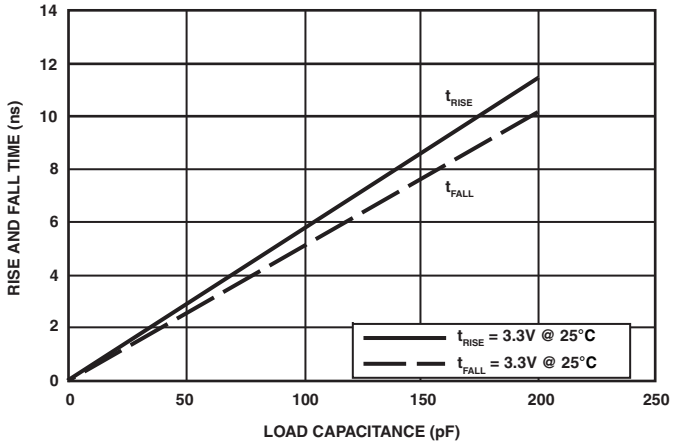


Figure 49. Driver Type C Typical Rise and Fall Times (10%–90%) vs. Load Capacitance (3.3 V V_{DDEXT})

PROCESSOR—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 43 and Table 44.

P_D = power dissipation (see Total Power Dissipation on Page 30 for the method to calculate P_D).

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

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

where T_A = ambient temperature (°C).

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

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

In Table 43 and Table 44, airflow measurements comply with JEDEC standards JESD51-2 and JESD51-6, and the junction-to-board measurement complies with JESD51-8. The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC test board.

Table 43. Thermal Characteristics (88-Lead LFCSP)

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	26.2	°C/W
θ_{JMA}	1 linear m/s air flow	23.7	°C/W
θ_{JMA}	2 linear m/s air flow	22.9	°C/W
θ_{JB}		16.0	°C/W
θ_{JC}		9.8	°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

Table 44. Thermal Characteristics (120-Lead LQFP)

Parameter	Condition	Typical	Unit
θ_{JA}	0 linear m/s air flow	26.9	°C/W
θ_{JMA}	1 linear m/s air flow	24.2	°C/W
θ_{JMA}	2 linear m/s air flow	23.3	°C/W
θ_{JB}		16.4	°C/W
θ_{JC}		12.7	°C/W
Ψ_{JT}	0 linear m/s air flow	0.50	°C/W
Ψ_{JT}	1 linear m/s air flow	0.77	°C/W
Ψ_{JT}	2 linear m/s air flow	1.02	°C/W

ADC—TIMING SPECIFICATIONS

Table 50. Serial Data Interface[1]

Parameter	Specification	Unit	Test Conditions / Comments
f_{ADSCLK}[2]	1/32	MHz min/max	
$t_{CONVERT}$	$14 \times t_{ADSCLK}$	ns max	$t_{ADSCLK} = 1/f_{ADSCLK}$
	437.5	ns max	f_{ADSCLK} = 32 MHz, f_{SAMPLE} = 2 MSPS; AV_{DD}, DV_{DD} = 5 V
	560.0	ns max	f_{ADSCLK} = 25 MHz, f_{SAMPLE} = 1.56 MSPS; AV_{DD}, DV_{DD} = 3 V
	583.3	ns max	f_{ADSCLK} = 24 MHz, f_{SAMPLE} = 1.5 MSPS; AV_{DD}, DV_{DD} = 2.7 V
t_{QUIET}	30	ns min	Minimum time between end of serial read and next falling edge of $\overline{CS}$
t_2	18/23	ns min	$\overline{CS}$ to ADSCLK setup time; V_{DD} = 5 V/3 V
t_3	15	ns max	Delay from $\overline{CS}$ until $D_{OUT}A$ and $D_{OUT}B$ are three-state disabled
t_4[3]	27/36	ns max	Data access time after ADSCLK falling edge, V_{DD} = 5 V/3 V
t_5	0.45 t_{ADSCLK}	ns min	ADSCLK low pulse width
t_6	0.45 t_{ADSCLK}	ns min	ADSCLK high pulse width
t_7	5/10	ns min	ADSCLK to data valid hold time, V_{DD} = 5 V/3 V
t_8	15	ns max	$\overline{CS}$ rising edge to $D_{OUT}A$, $D_{OUT}B$, high impedance
t_9	30	ns min	$\overline{CS}$ rising edge to falling edge pulse width
t_{10}	5/35	ns min/max	ADSCLK falling edge to $D_{OUT}A$, $D_{OUT}B$, high impedance

[1] See Figure 87 on Page 72 and Figure 88 on Page 72.
[2] Minimum ADSCLK for specified performance; with slower ADSCLK frequencies, performance specifications apply typically.
[3] The time required for the output to cross 0.4 V or 2.4 V.

ADC—ABSOLUTE MAXIMUM RATINGS

Stresses above those listed in Table 51 may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 51. Absolute Maximum Ratings

Parameter	Rating
AV_{DD}, DV_{DD} to AGND	–0.3 V to +7 V
DV_{DD} to DGND	–0.3 V to +7 V
V_{DRIVE} to DGND	–0.3 V to DV_{DD}
V_{DRIVE} to AGND	–0.3 V to AV_{DD}
AV_{DD} to DV_{DD}	–0.3 V to +0.3 V
AGND to DGND	–0.3 V to +0.3 V
Analog Input Voltage to AGND	–0.3 V to AV_{DD} + 0.3 V
Digital Input Voltage to DGND	–0.3 V to +7 V
Digital Output Voltage to GND	–0.3 V to V_{DRIVE} + 0.3 V
V_{REF} to AGND	–0.3 V to AV_{DD} + 0.3 V
Input Current to Any ADC Pin Except Supplies[1]	±10 mA
Storage Temperature Range	See Table 20 on Page 31
Junction Temperature Under Bias	See Table 20 on Page 31

[1] Transient currents of up to 100 mA will not cause latch up.

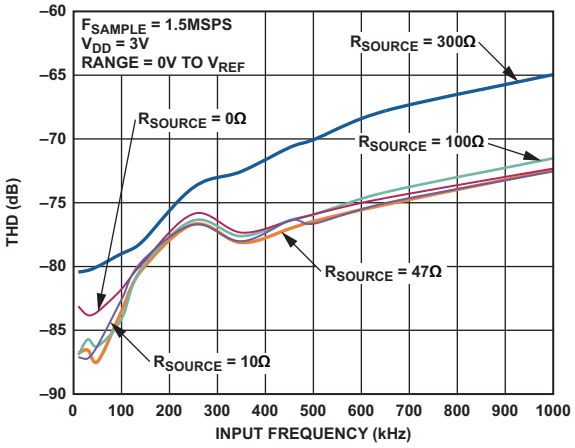


Figure 66. THD vs. Analog Input Frequency for Various Source Impedances, Differential Mode

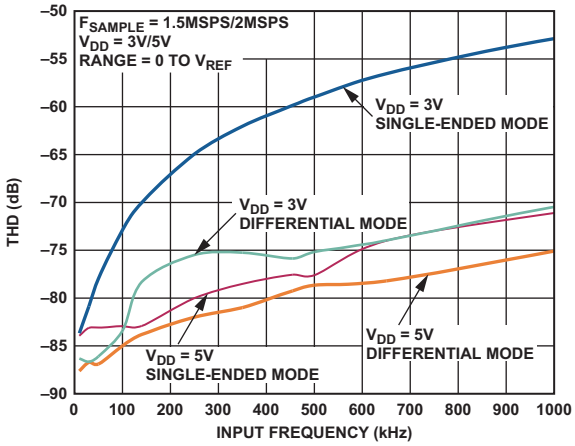


Figure 67. THD vs. Analog Input Frequency for Various Supply Voltages

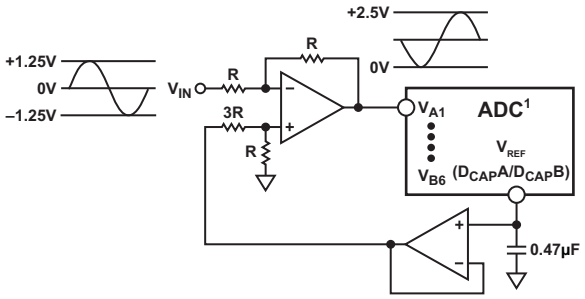
Analog Inputs

The ADC has a total of 12 analog inputs. Each on-board ADC has six analog inputs that can be configured as six single-ended channels, three pseudo differential channels, or three fully differential channels. These may be selected as described in the Analog Input Selection section.

Single-Ended Mode

The ADC can have a total of 12 single-ended analog input channels. In applications where the signal source has high impedance, it is recommended to buffer the analog input before applying it to the ADC. The analog input range can be programmed to be either 0 to V_{REF} or 0 to 2 × V_{REF}.

If the analog input signal to be sampled is bipolar, the internal reference of the ADC can be used to externally bias up this signal to make it correctly formatted for the ADC. Figure 68 shows a typical connection diagram when operating the ADC in single-ended mode.



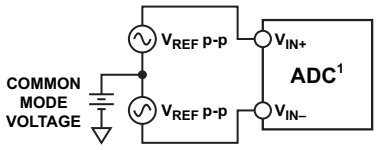
[1]ADDITIONAL PINS OMITTED FOR CLARITY.

Figure 68. Single-Ended Mode Connection Diagram

Differential Mode

The ADC can have a total of six differential analog input pairs.

Differential signals have some benefits over single-ended signals, including noise immunity based on the device's common-mode rejection and improvements in distortion performance. Figure 69 (Differential Input Definition) defines the fully differential analog input of the ADC.



[1]ADDITIONAL PINS OMITTED FOR CLARITY.

Figure 69. Differential Input Definition

The amplitude of the differential signal is the difference between the signals applied to the V_{IN+} and V_{IN-} pins in each differential pair (V_{IN+} V_{IN-}). V_{IN+} and V_{IN-} should be simultaneously driven by two signals each of amplitude V_{REF} (or 2 × V_{REF}, depending on the range chosen) that are 180° out of phase. The amplitude of the differential signal is, therefore (assuming the 0 to V_{REF} range is selected) –V_{REF} to +V_{REF} peak-to-peak (2 × V_{REF}), regardless of the common mode (CM).

The common mode is the average of the two signals

$$(V_{IN+} + V_{IN-})/2$$

and is, therefore, the voltage on which the two inputs are centered.

This results in the span of each input being CM ± V_{REF}/2. This voltage has to be set up externally and its range varies with the reference value, V_{REF}. As the value of V_{REF} increases, the common-mode range decreases. When driving the inputs with an amplifier, the actual common-mode range is determined by the amplifier's output voltage swing.

Figure 70 (Input Common-Mode Range vs. VREF (0 to VREF Range, VDD = 5 V)) and Figure 71 (Input Common-Mode Range vs. VREF (2 × VREF Range, VDD = 5 V)) show how the common-mode range typically varies with V_{REF} for a 5 V power

supply using the 0 to V_{REF} range or 2 × V_{REF} range, respectively. The common mode must be in this range to guarantee the functionality of the ADC.

When a conversion takes place, the common mode is rejected, resulting in a virtually noise free signal of amplitude –V_{REF} to +V_{REF} corresponding to the digital codes of 0 to 4096. If the 2 × V_{REF} range is used, then the input signal amplitude extends from – 2 V_{REF} to +2 V_{REF} after conversion.

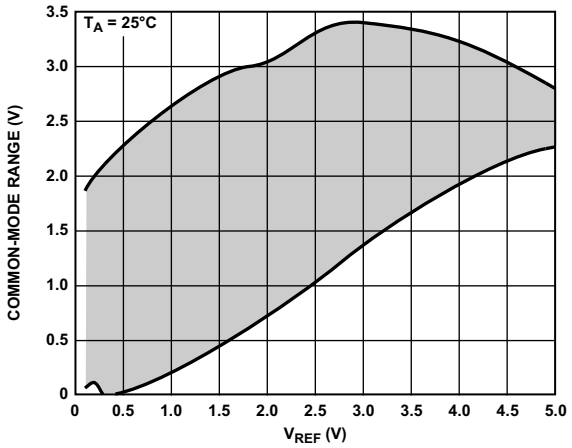


Figure 70. Input Common-Mode Range vs. V_{REF} (0 to V_{REF} Range, V_{DD} = 5 V)

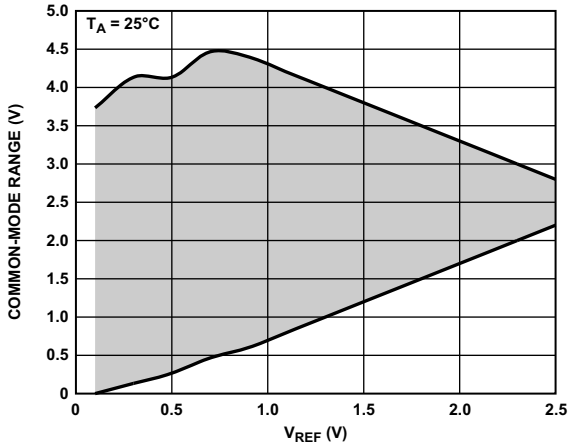


Figure 71. Input Common-Mode Range vs. V_{REF} (2 × V_{REF} Range, V_{DD} = 5 V)

Driving Differential Inputs

Differential operation requires that V_{IN+} and $V_{IN–}$ be simultaneously driven with two equal signals that are 180° out of phase. The common mode must be set up externally. The common-mode range is determined by V_{REF}, the power supply, and the particular amplifier used to drive the analog inputs. Differential modes of operation with either an ac or dc input provide the best THD performance over a wide frequency range. Because not all applications have a signal preconditioned for differential operation, there is often a need to perform single-ended-to-differential conversion.

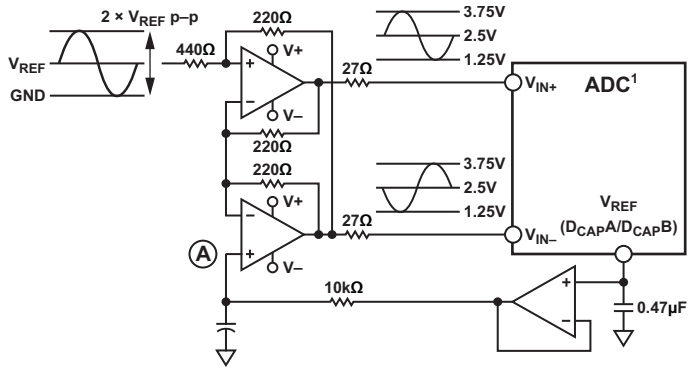
Using an Op Amp Pair

An op amp pair can be used to directly couple a differential signal to one of the analog input pairs of the ADC. The circuit configurations illustrated in Figure 72 (Dual Op Amp Circuit to Convert a Single-Ended Unipolar Signal Into a Differential Signal) and Figure 73 (Dual Op Amp Circuit to Convert a Single-Ended Bipolar Signal into a Differential Unipolar Signal) show how a dual op amp can be used to convert a single-ended signal into a differential signal for both a bipolar and unipolar input signal, respectively.

The voltage applied to Point A sets up the common-mode voltage. In both diagrams, it is connected in some way to the reference, but any value in the common-mode range can be input here to set up the common mode. The AD8022 is a suitable dual op amp that can be used in this configuration to provide differential drive to the ADC.

Take care when choosing the op amp; the selection depends on the required power supply and system performance objectives. The driver circuits in Figure 72 (Dual Op Amp Circuit to Convert a Single-Ended Unipolar Signal Into a Differential Signal) and Figure 73 (Dual Op Amp Circuit to Convert a Single-Ended Bipolar Signal into a Differential Unipolar Signal) are optimized for dc coupling applications requiring best distortion performance.

The circuit configuration shown in Figure 72 (Dual Op Amp Circuit to Convert a Single-Ended Unipolar Signal Into a Differential Signal) converts a unipolar, single-ended signal into a differential signal.



[1]ADDITIONAL PINS OMITTED FOR CLARITY.

Figure 72. Dual Op Amp Circuit to Convert a Single-Ended Unipolar Signal Into a Differential Signal

The differential op amp driver circuit shown in Figure 73 (Dual Op Amp Circuit to Convert a Single-Ended Bipolar Signal into a Differential Unipolar Signal) is configured to convert and level shift a single-ended, ground-referenced (bipolar) signal to a differential signal centered at the V_{REF} level of the ADC.

Pseudo Differential Mode

The ADC can have a total of six pseudo differential pairs. In this mode, V_{IN+} is connected to the signal source that must have an amplitude of V_{REF} (or 2 × V_{REF}, depending on the range chosen)

120-LEAD LQFP LEAD ASSIGNMENT

Table 54 lists the LQFP leads by signal mnemonic. Table 55 on Page 74 lists the LQFP leads by lead number.

Table 54. 120-Lead LQFP Lead Assignment (Alphabetical by Signal)

Signal	Lead No.	Signal	Lead No.	Signal	Lead No.	Signal	Lead No.
A0	100	NC	72	PG11	46	V_{B5}	88
A1	98	$\overline{NMI}$	11	PG12	47	V_{B6}	87
A2	97	PF0	118	PG13	48	V_{DDEXT}	1
AGND	73	PF1	119	PG14	49	V_{DDEXT}	6
AGND	78	PF2	2	PG15	50	V_{DDEXT}	15
AGND	79	PF3	4	PH0	113	V_{DDEXT}	20
AGND	82	PF4	3	PH1	115	V_{DDEXT}	23
AGND	93	PF5	5	PH2	114	V_{DDEXT}	26
AGND	99	PF6	7	RANGE	95	V_{DDEXT}	30
AV_{DD}	76	PF7	8	REF_SELECT	75	V_{DDEXT}	41
BMODE0	58	PF8	9	$\overline{RESET}$	12	V_{DDEXT}	51
BMODE1	57	PF9	10	SCL	55	V_{DDEXT}	59
BMODE2	56	PF10	14	ADSCLK	102	V_{DDEXT}	62
CLKIN	110	PF11	16	SDA	54	V_{DDEXT}	64
$\overline{CS}$	101	PF12	18	SGL/$\overline{DIFF}$	96	V_{DDEXT}	66
$D_{CAP}A$	77	PF13	19	TCK	34	V_{DDEXT}	67
$D_{CAP}B$	94	PF14	21	TDI	33	V_{DDEXT}	112
DGND	74	PF15	22	TDO	36	V_{DDEXT}	116
DGND	104	$\overline{PG}$	71	TMS	35	$V_{DDFLASH}$	25
$D_{OUT}A$	105	PG0	27	$\overline{TRST}$	37	$V_{DDFLASH}$	63
$D_{OUT}B$	103	PG1	28	V_{A1}	80	$V_{DDFLASH}$	69
DV_{DD}	107	PG2	29	V_{A2}	81	V_{DDINT}	24
$\overline{EMU}$	68	PG3	31	V_{A3}	83	V_{DDINT}	42
EXT_WAKE	70	PG4	32	V_{A4}	84	V_{DDINT}	52
EXTCLK	120	PG5	38	V_{A5}	85	V_{DDINT}	53
GND	13	PG6	39	V_{A6}	86	V_{DDINT}	61
GND	17	PG7	40	V_{B1}	92	V_{DDINT}	65
GND	108	PG8	43	V_{B2}	91	V_{DDINT}	117
GND	109	PG9	44	V_{B3}	90	V_{DRIVE}	106
NC	60	PG10	45	V_{B4}	89	XTAL	111
						GND	121*
						AGND	122**

* Pin no. 121 is the GND supply (see Figure 89 and Figure 90) for the processor (4.6mm × 6.17mm); this pad **must** connect to GND.
** Pin no. 122 is the AGND supply (see Figure 89 and Figure 90) for the ADC (2.81mm × 2.81mm); this pad **must** connect to AGND.

OUTLINE DIMENSIONS

Dimensions in Figure 93 (for the 120-lead LQFP) and in Figure 94 (for the 88-lead LFCSP) are shown in millimeters.

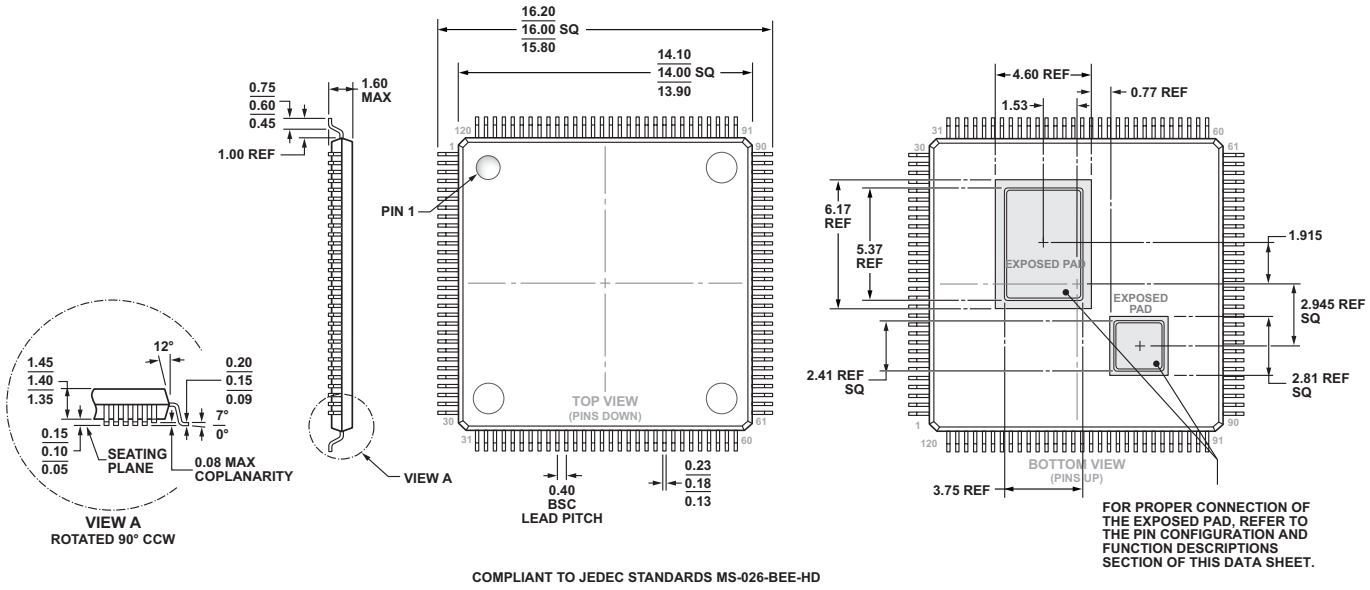


Figure 93. 120-Lead Low Profile Quad Flat Package, Exposed Pad [LQFP_EP][1]
(SW-120-2)
Dimensions shown in millimeters

[1] For information relating to the SW-120-2 package's exposed pad, see the table endnote on Page 74.