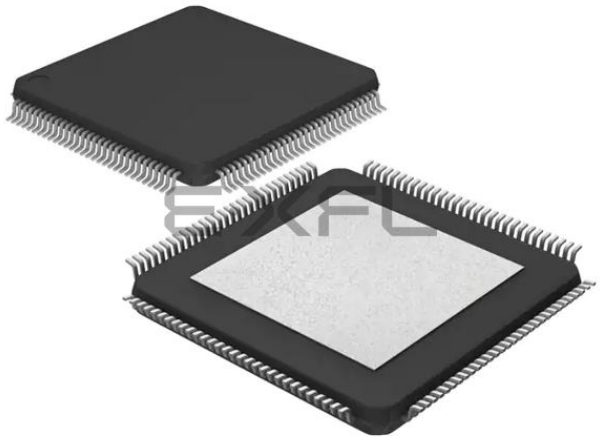


Welcome to E-XFL.COM

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	Obsolete
Type	Fixed Point
Interface	CAN, EBI/EMI, I²C, IrDA, PPI, SPI, SPORT, UART/USART
Clock Rate	300MHz
Non-Volatile Memory	FLASH (16MB)
On-Chip RAM	68kB
Voltage - I/O	3.30V
Voltage - Core	1.29V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	120-LQFP Exposed Pad
Supplier Device Package	120-LQFP-EP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-bf506kswz-3f

Blackfin processors support a modified Harvard architecture in combination with a hierarchical memory structure. Level 1 (L1) memories are those that typically operate at the full processor speed with little or no latency. At the L1 level, the instruction memory holds instructions only. The data memory holds data, and a dedicated scratchpad data memory stores stack and local variable information.

In addition, multiple L1 memory blocks are provided, offering a configurable mix of SRAM and cache. The memory management unit (MMU) provides memory protection for individual tasks that may be operating on the core and can protect system registers from unintended access.

The architecture provides three modes of operation: user mode, supervisor mode, and emulation mode. User mode has restricted access to certain system resources, thus providing a protected software environment, while supervisor mode has unrestricted access to the system and core resources.

The Blackfin processor instruction set has been optimized so that 16-bit opcodes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit opcodes, representing fully featured multifunction instructions. Blackfin processors support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

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

MEMORY ARCHITECTURE

The Blackfin processor views memory as a single unified 4G byte address space, using 32-bit addresses. All resources, including internal memory, external memory, and I/O control registers, occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low latency core-accessible memory as cache or SRAM and to provide larger, lower cost and performance interface-accessible memory systems. See Figure 3.

The core-accessible L1 memory system is the highest performance memory available to the Blackfin processor. The interface-accessible memory system, accessed through the external bus interface unit (EBIU), provides access to the internal flash memory and boot ROM.

The memory DMA controller provides high bandwidth data movement capability. It can perform block transfers of code or data between the internal memory and the external memory spaces.

Internal (Core-Accessible) Memory

The processor has three blocks of core-accessible memory, providing high-bandwidth access to the core.

The first block is the L1 instruction memory, consisting of 32K bytes SRAM, of which 16K bytes can be configured as a four-way set-associative cache. This memory is accessed at full processor speed.

The second core-accessible memory block is the L1 data memory, consisting of 32K bytes of SRAM, of which 16K bytes may be configured as cache. This memory block is accessed at full processor speed.

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

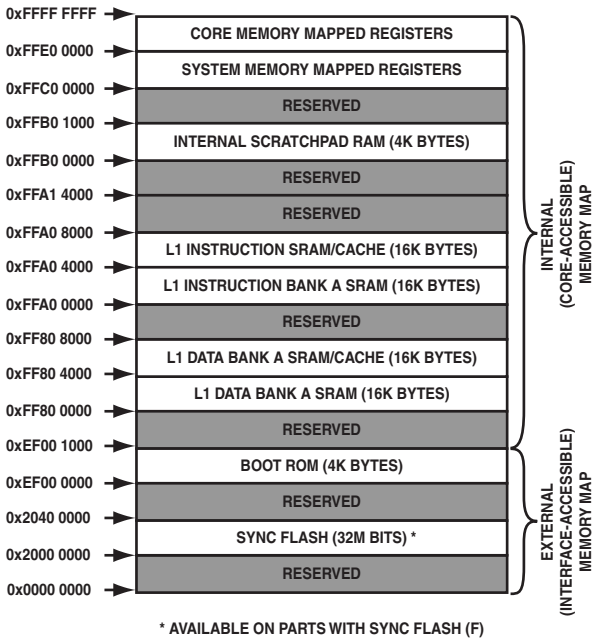


Figure 3. Internal/External Memory Map

External (Interface-Accessible) Memory

External memory is accessed via the EBIU memory port. This 16-bit interface provides a glueless connection to the internal flash memory and boot ROM. Internal flash memory ships from the factory in an erased state except for Block 0 of the parameter bank. Block 0 of the Flash memory parameter bank ships from the factory in an unknown state. An erase operation should be performed prior to programming this block.

I/O Memory Space

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

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)

Peripheral Interrupt Source	General-Purpose Interrupt (at Reset)	Peripheral Interrupt ID	Default Core Interrupt ID	SIC Registers	
PLL Wakeup Interrupt	IVG7	0	0	IAR0	IMASK0, ISR0, IWR0
DMA Error (generic)	IVG7	1	0	IAR0	IMASK0, ISR0, IWR0
PPI Status	IVG7	2	0	IAR0	IMASK0, ISR0, IWR0
SPORT0 Status	IVG7	3	0	IAR0	IMASK0, ISR0, IWR0
SPORT1 Status	IVG7	4	0	IAR0	IMASK0, ISR0, IWR0
UART0 Status	IVG7	5	0	IAR0	IMASK0, ISR0, IWR0
UART1 Status	IVG7	6	0	IAR0	IMASK0, ISR0, IWR0
SPI0 Status	IVG7	7	0	IAR0	IMASK0, ISR0, IWR0
SPI1 Status	IVG7	8	0	IAR1	IMASK0, ISR0, IWR0
CAN Status	IVG7	9	0	IAR1	IMASK0, ISR0, IWR0
RSI Mask 0 Interrupt	IVG7	10	0	IAR1	IMASK0, ISR0, IWR0
Reserved	—	11	—	IAR1	IMASK0, ISR0, IWR0
CNT0 Interrupt	IVG8	12	1	IAR1	IMASK0, ISR0, IWR0
CNT1 Interrupt	IVG8	13	1	IAR1	IMASK0, ISR0, IWR0
DMA Channel 0 (PPI Rx/Tx)	IVG9	14	2	IAR1	IMASK0, ISR0, IWR0
DMA Channel 1 (RSI Rx/Tx)	IVG9	15	2	IAR1	IMASK0, ISR0, IWR0
DMA Channel 2 (SPORT0 Rx)	IVG9	16	2	IAR2	IMASK0, ISR0, IWR0
DMA Channel 3 (SPORT0 Tx)	IVG9	17	2	IAR2	IMASK0, ISR0, IWR0
DMA Channel 4 (SPORT1 Rx)	IVG9	18	2	IAR2	IMASK0, ISR0, IWR0
DMA Channel 5 (SPORT1 Tx)	IVG9	19	2	IAR2	IMASK0, ISR0, IWR0
DMA Channel 6 (SPI0 Rx/Tx)	IVG10	20	3	IAR2	IMASK0, ISR0, IWR0
DMA Channel 7 (SPI1 Rx/Tx)	IVG10	21	3	IAR2	IMASK0, ISR0, IWR0
DMA Channel 8 (UART0 Rx)	IVG10	22	3	IAR2	IMASK0, ISR0, IWR0
DMA Channel 9 (UART0 Tx)	IVG10	23	3	IAR2	IMASK0, ISR0, IWR0
DMA Channel 10 (UART1 Rx)	IVG10	24	3	IAR3	IMASK0, ISR0, IWR0
DMA Channel 11 (UART1 Tx)	IVG10	25	3	IAR3	IMASK0, ISR0, IWR0
CAN Receive	IVG11	26	4	IAR3	IMASK0, ISR0, IWR0
CAN Transmit	IVG11	27	4	IAR3	IMASK0, ISR0, IWR0
TWI	IVG11	28	4	IAR3	IMASK0, ISR0, IWR0
Port F Interrupt A	IVG11	29	4	IAR3	IMASK0, ISR0, IWR0
Port F Interrupt B	IVG11	30	4	IAR3	IMASK0, ISR0, IWR0
Reserved	—	31	—	IAR3	IMASK0, ISR0, IWR0
Timer 0	IVG12	32	5	IAR4	IMASK1, ISR1, IWR1
Timer 1	IVG12	33	5	IAR4	IMASK1, ISR1, IWR1
Timer 2	IVG12	34	5	IAR4	IMASK1, ISR1, IWR1
Timer 3	IVG12	35	5	IAR4	IMASK1, ISR1, IWR1
Timer 4	IVG12	36	5	IAR4	IMASK1, ISR1, IWR1
Timer 5	IVG12	37	5	IAR4	IMASK1, ISR1, IWR1
Timer 6	IVG12	38	5	IAR4	IMASK1, ISR1, IWR1
Timer 7	IVG12	39	5	IAR4	IMASK1, ISR1, IWR1
Port G Interrupt A	IVG12	40	5	IAR5	IMASK1, ISR1, IWR1
Port G Interrupt B	IVG12	41	5	IAR5	IMASK1, ISR1, IWR1
MDMA Stream 0	IVG13	42	6	IAR5	IMASK1, ISR1, IWR1

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 eight general-purpose programmable timers, a ninth 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.

UP/DOWN COUNTERS AND THUMBWHEEL INTERFACES

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

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

Internal signals forwarded to each timer unit enable these timers to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmable count values are exceeded.

3-PHASE PWM UNITS

The two/dual 3-phase PWM generation units each feature:

- 16-bit center-based PWM generation unit
- Programmable PWM pulse width
- Single/double update modes
- Programmable dead time and switching frequency
- Twos-complement implementation which permits smooth transition to full ON and full OFF states
- Possibility to synchronize the PWM generation to either externally-generated or internally-generated synchronization pulses
- Special provisions for BDCM operation (crossover and output enable functions)
- Wide variety of special switched reluctance (SR) operating modes
- Output polarity and clock gating control
- Dedicated asynchronous PWM shutdown signal

Each PWM block integrates a flexible and programmable 3-phase PWM waveform generator that can be programmed to generate the required switching patterns to drive a 3-phase voltage source inverter for ac induction motor (ACIM) or permanent magnet synchronous motor (PMSM) control. In addition, the PWM block contains special functions that considerably simplify the generation of the required PWM switching patterns for control of the electronically commutated motor (ECM) or brushless dc motor (BDCM). Software can enable a special mode for switched reluctance motors (SRM).

The six PWM output signals (per PWM unit) consist of three high-side drive signals (PWMx_AH, PWMx_BH, and PWMx_CH) and three low-side drive signals (PWMx_AL, PWMx_BL, and PWMx_CL). The polarity of the generated PWM signal can be set with software, so that either active HI or active LO PWM patterns can be produced.

The switching frequency of the generated PWM pattern is programmable using the 16-bit PWM_TM register. The PWM generator can operate in single update mode or double update mode. In single update mode, the duty cycle values are programmable only once per PWM period, so that the resultant PWM patterns are symmetrical about the midpoint of the PWM period. In the double update mode, a second updating of the PWM registers is implemented at the midpoint of the PWM period. In this mode, it is possible to produce asymmetrical PWM patterns that produce lower harmonic distortion in 3-phase PWM inverters.

Pulses synchronous to the switching frequency can be generated internally and output on the PWMx_SYNC pin. The PWM unit can also accept externally generated synchronization pulses through PWMx_SYNC.

Each PWM unit features a dedicated asynchronous shutdown pin, $\overline{\text{PWMx\_TRIP}}$, which (when brought low) instantaneously places all six PWM outputs in the OFF state.

SERIAL PORTS

The processors incorporate two dual-channel synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

- I^2S capable operation.
- Bidirectional operation—Each SPORT has two sets of independent transmit and receive pins, enabling eight channels of I^2S stereo audio.
- Buffered (8-deep) transmit and receive ports—Each port has a data register for transferring data words to and from other processor components and shift registers for shifting data in and out of the data registers.
- Clocking—Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from (f_{SCLK}/131,070) Hz to (f_{SCLK}/2) Hz.
- Word length—Each SPORT supports serial data words from 3 to 32 bits in length, transferred most significant bit first or least significant bit first.
- Framing—Each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two pulse widths and early or late frame sync.
- Companding in hardware—Each SPORT can perform A-law or μ-law companding according to ITU recommendation G.711. Companding can be selected on the transmit and/or receive channel of the SPORT without additional latencies.

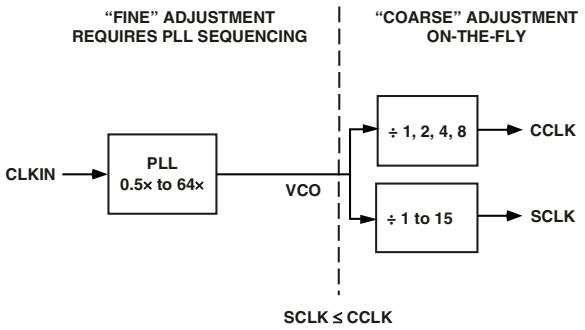


Figure 5. Frequency Modification Methods

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 6 illustrates typical system clock ratios.

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

Table 6. Example System Clock Ratios

Signal Name SSEL3–0	Divider Ratio VCO/SCLK	Example Frequency Ratios (MHz)	
		VCO	SCLK
0001	1:1	50	50
0110	6:1	300	50
1010	10:1	400	40

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

Table 7. Core Clock Ratios

Signal Name CSEL1–0	Divider Ratio VCO/CCLK	Example Frequency Ratios (MHz)	
		VCO	CCLK
00	1:1	300	300
01	2:1	300	150
10	4:1	400	100
11	8:1	200	25

The maximum CCLK frequency *both* depends on the part's speed grade *and* depends on the applied V_{DDINT} voltage. See Table 14 for details. The maximal system clock rate (SCLK) depends on the applied V_{DDINT} and V_{DDEXT} voltages (see Table 16).

BOOTING MODES

The processor has several mechanisms (listed in Table 8) 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 8. Booting Modes

BMODE2–0	Description
000	Idle/No Boot
001	Boot from internal parallel flash in async mode[1]
010	Boot from internal parallel flash in sync mode[1]
011	Boot through SPI0 master from SPI memory
100	Boot through SPI0 slave from host device
101	Boot through PPI from host
110	Reserved
111	Boot through UART0 slave from host device

[1] This boot mode applies to ADSP-BF504F and ADSP-BF506F processors only.

The boot modes listed in Table 8 provide a number of mechanisms for automatically loading the processor's internal and external memories after a reset. By default, all boot modes use the slowest meaningful configuration settings. Default settings can be altered via the initialization code feature at boot time. Some boot modes require a boot host wait (HWAIT) signal, which is a GPIO output signal that is driven and toggled by the boot kernel at boot time. If pulled high through an external pull-up resistor, the HWAIT signal behaves active high and will be driven low when the processor is ready for data. Conversely, when pulled low, HWAIT is driven high when the processor is ready for data. When the boot sequence completes, the HWAIT pin can be used for other purposes. The BMODE pins of the reset configuration register, sampled during power-on resets and software-initiated resets, implement the modes shown in Table 8.

- 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.
- Boot from stacked parallel flash in 16-bit asynchronous mode (BMODE = 0x1)—In this mode, conservative timing parameters are used to communicate with the flash device. The boot kernel communicates with the flash device asynchronously.
- Boot from stacked parallel flash in 16-bit synchronous mode (BMODE = 0x2)—In this mode, fast timing parameters are used to communicate with the flash device. The boot kernel configures the flash device for synchronous burst communication and boots from the flash synchronously.

- 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.

The ACM synchronizes the ADC conversion process; generating the ADC controls, the ADC conversion start signal, and other signals. The actual data acquisition from the ADC is done by the SPORT peripherals.

The serial interface on the ADC allows the part to be directly connected to the ADSP-BF504, ADSP-BF504F, and ADSP-BF506F processors using serial interface protocols.

Figure 6 shows how to connect an external ADC to the ACM and one of the two SPORTs on the ADSP-BF504 or ADSP-BF504F processors.

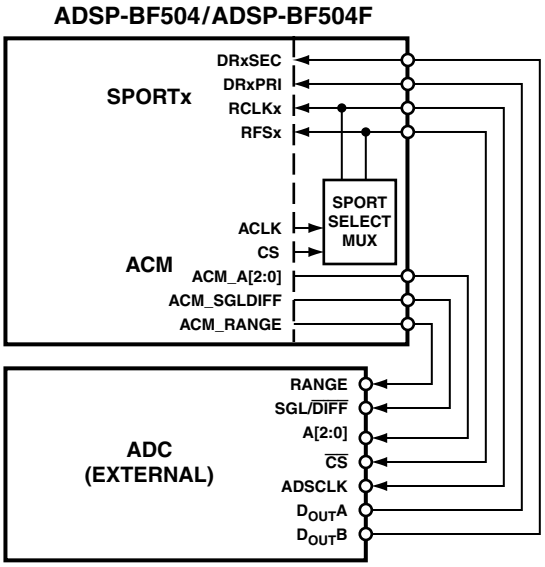


Figure 6. ADC (External), ACM, and SPORT Connections

The ADC is integrated into the ADSP-BF506F product. Figure 7 shows how to connect the internal ADC to the ACM and to one of the two SPORTs on the ADSP-BF506F processor.

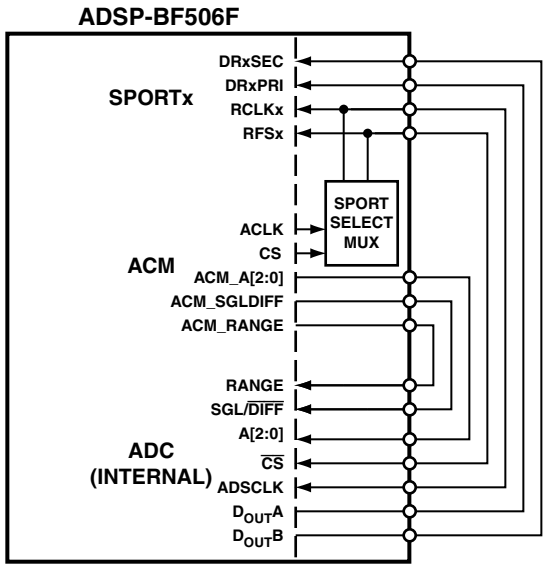


Figure 7. ADC (Internal), ACM, and SPORT Connections

The ADSP-BF504, ADSP-BF504F, and ADSP-BF506F processors interface directly to the ADC without any glue logic required. The availability of secondary receive registers on the serial ports of the Blackfin processors means only one serial port is necessary to read from both D_{OUT} pins simultaneously. Figure 7 (ADC (Internal), ACM, and SPORT Connections) shows both $D_{OUT}A$ and $D_{OUT}B$ of the ADC connected to one of the processor's serial ports. The SPORTx Receive Configuration 1 register and SPORTx Receive Configuration 2 register should be set up as outlined in Table 9 (The SPORTx Receive Configuration 1 Register (SPORTx_RCR1)) and Table 10 (The SPORTx Receive Configuration 2 Register (SPORTx_RCR2)).

Table 9. The SPORTx Receive Configuration 1 Register (SPORTx_RCR1)

Setting	Description
RCKFE = 0	Sample data with falling edge of RSCLK
LRFS = 1	Active low frame signal
RFSR = 1	Frame every word
IRFS = 0	External RFS used
RLSBIT = 0	Receive MSB first
RDTYPE = 00	Zero fill
IRCLK = 0	External receive clock
RSPEN = 1	Receive enabled
TFSR = RFSR = 1	

SIGNAL DESCRIPTIONS

Signal definitions for the ADSP-BF50x processors are listed in Table 11. All pins for the ADC (ADSP-BF506F processor only) are listed in Table 12.

In order to maintain maximum function and reduce package size and pin count, some pins have multiple, multiplexed functions. In cases where pin function is reconfigurable, the default state is shown in plain text, while the alternate functions are shown in italics.

During and immediately after reset, all processor signals (not ADC signals) are three-stated with the following exceptions: EXT_WAKE is driven high and XTAL is driven in conjunction with CLKIN to create a crystal oscillator circuit. During hibernate, all signals are three-stated with the following exceptions: EXT_WAKE is driven low and XTAL is driven to a solid logic level.

During and immediately after reset, all I/O pins have their input buffers disabled until enabled by user software with the exception of the pins that need pull-ups or pull-downs, as noted in Table 11.

Adding a parallel termination to CLKOUT may prove useful in further enhancing signal integrity. Be sure to verify overshoot/undershoot and signal integrity specifications on actual hardware.

Table 11. Processor—Signal Descriptions

Signal Name	Type	Function	Driver Type
Port F: *GPIO and Multiplexed Peripherals*			
PF0/*TSCLK0/UA0_RX/TMR6/CUD0*	I/O	GPIO/*SPORT0 TX Serial CLK/UART0 RX/Timer6/Count Up Dir 0*	C
PF1/*RSCLK0/UA0_TX/TMR5/CDG0*	I/O	GPIO/*SPORT0 RX Serial CLK/UART0 TX/Timer5/Count Down Dir 0*	C
PF2/*DT0PRI/PWM0_BH/PPI_D8/CZM0*	I/O	GPIO/*SPORT0 TX Pri Data/PWM0 Drive B Hi/PPI Data 8/Counter Zero Marker 0*	C
PF3/*TFS0/PWM0_BL/PPI_D9/CDG0*	I/O	GPIO/*SPORT0 TX Frame Sync/PWM0 Drive B Lo/PPI Data 9/Count Down Dir 0*	C
PF4/*RFS0/PWM0_CH/PPI_D10/TACLK0*	I/O	GPIO/*SPORT0 RX Frame Sync/PWM0 Drive C Hi/PPI Data 10/Alt Timer CLK 0*	C
PF5/*DR0PRI/PWM0_CL/PPI_D11/TACLK1*	I/O	GPIO/*SPORT0 Pri RX Data/PWM0 Drive C Lo/PPI Data 11/Alt Timer CLK 1*	C
PF6/*UA1_TX/$\overline{\text{PWM0\_TRIP}}$/PPI_D12*	I/O	GPIO/*UART1 TX/PWM0 TRIP/PPI Data 12*	C
PF7/*UA1_RX/PWM0_SYNC/PPI_D13/TACI3*	I/O	GPIO/*UART1 RX/PWM0 SYNC/PPI Data 13/Alt Capture In 3*	C
PF8/*$\overline{\text{UA1\_RTS}}$/DT0SEC/PPI_D7*	I/O	GPIO/*UART1 RTS/SPORT0 TX Sec Data/PPI Data 7*	C
PF9/*$\overline{\text{UA1\_CTS}}$/DR0SEC/PPI_D6/CZM0*	I/O	GPIO/*UART1 CTS/SPORT0 Sec RX Data/PPI Data 6/Counter Zero Marker 0*	C
PF10/*SPI0_SCK/TMR2/PPI_D5*	I/O	GPIO/*SPI0 SCK/Timer2/PPI Data 5*	C
PF11/*SPI0_MISO/$\overline{\text{PWM0\_TRIP}}$/PPI_D4/TACLK2*	I/O	GPIO/*SPI0 MISO/PWM0 TRIP/PPI Data 4/Alt Timer CLK 2*	C
PF12/*SPI0_MOSI/PWM0_SYNC/PPI_D3*	I/O	GPIO/*SPI0 MOSI/PWM0 SYNC/PPI Data 3*	C
PF13/*$\overline{\text{SPI0\_SEL1}}$/TMR3/PPI_D2/$\overline{\text{SPI0\_SS}}$*	I/O	GPIO/*SPI0 Slave Select 1/Timer3/PPI Data 2/SPI0 Slave Select In*	C
PF14/*$\overline{\text{SPI0\_SEL2}}$/PWM0_AH/PPI_D1*	I/O	GPIO/*SPI0 Slave Select 2/PWM0 AH/PPI Data 1*	C
PF15/*$\overline{\text{SPI0\_SEL3}}$/PWM0_AL/PPI_D0*	I/O	GPIO/*SPI0 Slave Select 3/PWM0 AL/PPI Data 0*	C
Port G: *GPIO and Multiplexed Peripherals*			
PG0/*$\overline{\text{SPI1\_SEL3}}$/TMRCLK/PPI_CLK/UA1_RX/TACI4*	I/O	GPIO/*SPI1 Slave Select 3/Timer CLK/PPI Clock/UART1 RX/Alt Capture In 4*	C
PG1/*$\overline{\text{SPI1\_SEL2}}$/PPI_FS3/CAN_RX/TACI5*	I/O	GPIO/*SPI1 Slave Select 2/PPI FS3/CAN RX/Alt Capture In 5*	C
PG2/*$\overline{\text{SPI1\_SEL1}}$/TMR4/CAN_TX/$\overline{\text{SPI1\_SS}}$*	I/O	GPIO/*SPI1 Slave Select 1/Timer4/CAN TX/SPI1 Slave Select In*	C
PG3/*HWAIT/SPI1_SCK/DT1SEC/UA1_TX*	I/O	GPIO/*HWAIT/SPI1 SCK/SPORT1 TX Sec Data/UART1 TX*	C
PG4/*SPI1_MOSI/DR1SEC/PWM1_SYNC/TACLK6*	I/O	GPIO/*SPI1 MOSI/SPORT1 Sec RX Data/PWM1 SYNC/Alt Timer CLK 6*	C
PG5/*SPI1_MISO/TMR7/$\overline{\text{PWM1\_TRIP}}$*	I/O	GPIO/*SPI1 MISO/Timer7/PWM1 TRIP*	C
PG6/*ACM_SGLDIFF/SD_D3/PWM1_AH*	I/O	GPIO/*ADC CM SGL DIFF/SD Data 3/PWM1 Drive A Hi*	C
PG7/*ACM_RANGE/SD_D2/PWM1_AL*	I/O	GPIO/*ADC CM RANGE/SD Data 2/PWM1 Drive A Lo*	C
PG8/*DR1SEC/SD_D1/PWM1_BH*	I/O	GPIO/*SPORT1 Sec RX Data/SD Data 1/PWM1 Drive B Hi*	C
PG9/*DR1PRI/SD_D0/PWM1_BL*	I/O	GPIO/*SPORT1 Pri RX Data/SD Data 0/PWM1 Drive B Lo*	C
PG10/*RFS1/SD_CMD/PWM1_CH/TACI6*	I/O	GPIO/*SPORT1 RX Frame Sync/SD CMD/PWM1 Drive C Hi/Alt Capture In 6*	C
PG11/*RSCLK1/SD_CLK/PWM1_CL/TACLK7*	I/O	GPIO/*SPORT1 RX Serial CLK/SD CLK/PWM1 Drive C Lo/Alt Timer CLK 7*	C
PG12/*UA0_RX/SD_D4/PPI_D15/TACI2*	I/O	GPIO/*UART0 RX/SD Data 4/PPI Data 15/Alt Capture In 2*	C
PG13/*UA0_TX/SD_D5/PPI_D14/CZM1*	I/O	GPIO/*UART0 TX/SD Data 5/PPI Data 14/Counter Zero Marker 1*	C

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	

Table 12. ADC—Signal Descriptions (ADSP-BF506F Processor Only)

Signal Name	Type	Function
DGND	G	Digital Ground. This is the ground reference point for all digital circuitry on the internal ADC. Both DGND pins should connect to the DGND plane of a system. The DGND and AGND voltages should ideally be at the same potential and must not be more than 0.3 V apart, even on a transient basis.
REF SELECT	I	Internal/External Reference Selection. Logic input. If this pin is tied to DGND, the on-chip 2.5 V reference is used as the reference source for both ADC A and ADC B. In addition, Pin $D_{CAP}A$ and Pin $D_{CAP}B$ must be tied to decoupling capacitors. If the REF SELECT pin is tied to a logic high, an external reference can be supplied to the internal ADC through the $D_{CAP}A$ and/or $D_{CAP}B$ pins.
AV_{DD}	P	Analog Supply Voltage, 2.7 V to 5.25 V. This is the only supply voltage for all analog circuitry on the internal ADC. The AV_{DD} and DV_{DD} voltages should ideally be at the same potential and must not be more than 0.3 V apart, even on a transient basis. This supply should be decoupled to AGND.
$D_{CAP}A$, $D_{CAP}B$ (V_{REF})	I	Decoupling Capacitor Pins. Decoupling capacitors (470 nF recommended) are connected to these pins to decouple the reference buffer for each respective ADC. Provided the output is buffered, the on-chip reference can be taken from these pins and applied externally to the rest of a system. The range of the external reference is dependent on the analog input range selected.
AGND	G	Analog Ground. Ground reference point for all analog circuitry on the internal ADC. All analog input signals and any external reference signal should be referred to this AGND voltage. All three of these AGND pins should connect to the AGND plane of a system. The AGND and DGND voltages ideally should be at the same potential and must not be more than 0.3 V apart, even on a transient basis.
V_{A1} to V_{A6}	I	Analog Inputs of ADC A. These may be programmed as six single-ended channels or three true differential analog input channel pairs. See Table 53 (Analog Input Type and Channel Selection).
V_{B1} to V_{B6}	I	Analog Inputs of ADC B. These may be programmed as six single-ended channels or three true differential analog input channel pairs. See Table 53 (Analog Input Type and Channel Selection).
RANGE	I	Analog Input Range Selection. Logic input. The polarity on this pin determines the input range of the analog input channels. If this pin is tied to a logic low, the analog input range is 0 V to V_{REF}. If this pin is tied to a logic high when $\overline{CS}$ goes low, the analog input range is $2 \times V_{REF}$. For details, see Table 53 (Analog Input Type and Channel Selection).
SGL/$\overline{DIFF}$	I	Logic Input. This pin selects whether the analog inputs are configured as differential pairs or single ended. A logic low selects differential operation while a logic high selects single-ended operation. For details, see Table 53 (Analog Input Type and Channel Selection).
A0 to A2	I	Multiplexer Select. Logic inputs. These inputs are used to select the pair of channels to be simultaneously converted, such as Channel 1 of both ADC A and ADC B, Channel 2 of both ADC A and ADC B, and so on. The pair of channels selected may be two single-ended channels or two differential pairs. The logic states of these pins need to be set up prior to the acquisition time and subsequent falling edge of $\overline{CS}$ to correctly set up the multiplexer for that conversion. For further details, see Table 53 (Analog Input Type and Channel Selection).
$\overline{CS}$	I	Chip Select. Active low logic input. This input provides the dual function of initiating conversions on the internal ADC and framing the serial data transfer. When connecting $\overline{CS}$ to a processor signal that is three-stated during reset and/or hibernate, adding a pull-up resistor may prove useful to avoid random ADC operation.
ADSCLK	I	Serial Clock. Logic input. A serial clock input provides the ADSCLK for accessing the data from the internal ADC. This clock is also used as the clock source for the conversion process.

Parameter		Test Conditions	Min	Typical	Max	Unit
$I_{DDFLASH2}$	Flash Memory Supply Current 2 — Reset/Powerdown			15	50	µA
$I_{DDFLASH3}$	Flash Memory Supply Current 3 — Standby			15	50	µA
$I_{DDFLASH4}$	Flash Memory Supply Current 4 — Automatic Standby			15	50	µA
$I_{DDFLASH5}$	Flash Memory Supply Current 5 — Program			15	40	mA
	Flash Memory Supply Current 5 — Erase			15	40	mA
$I_{DDFLASH6}$	Flash Memory Supply Current 6 — Dual Operations	Program/Erase in one bank, asynchronous read in another bank		25	60	mA
		Program/Erase in one bank, synchronous read in another bank		43	70	mA
$I_{DDFLASH7}$	Flash Memory Supply Current 7 — Program/Erase Suspended (Standby)			15	50	µA

[1] Applies to input pins.
[2] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{TRST}$).
[3] Applies to three-statable pins.
[4] Applies to bidirectional pins SCL and SDA.
[5] Applies to all signal pins, except SCL and SDA.
[6] Guaranteed, but not tested.
[7] See the *ADSP-BF50x Blackfin Processor Hardware Reference Manual* for definition of sleep, deep sleep, and hibernate operating modes.
[8] Applies to V_{DDEXT} supply only. Clock inputs are tied high or low.
[9] Guaranteed maximum specifications.
[10] Unit for V_{DDINT} is V (Volts). Unit for f_{SCLK} is MHz. Example: 1.4 V, 75 MHz would be $0.16 \times 1.4 \times 75 = 16.8$ mA adder.
[11] See the *ADSP-BF50x Blackfin Processor Hardware Reference Manual* for definition of NORCLK.

ESD SENSITIVITY

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PACKAGE INFORMATION

The information presented in Figure 9 and Table 23 provides details about the package branding for the ADSP-BF50x processors.

Figure 9. Product Information on Package

Table 23. Package Brand Information[1]

Brand Key	Field Description
ADSP-BF50x	Product Name[2]
t	Temperature Range
pp	Package Type
Z	RoHS Compliant Designation
ccc	See Ordering Guide
vvvvvv.x	Assembly Lot Code
n.n	Silicon Revision
#	RoHS Compliance Designator
yyww	Date Code

[1] Nonautomotive only. For branding information specific to Automotive products, contact Analog Devices Inc.
[2] See product names in the Ordering Guide on Page 81.

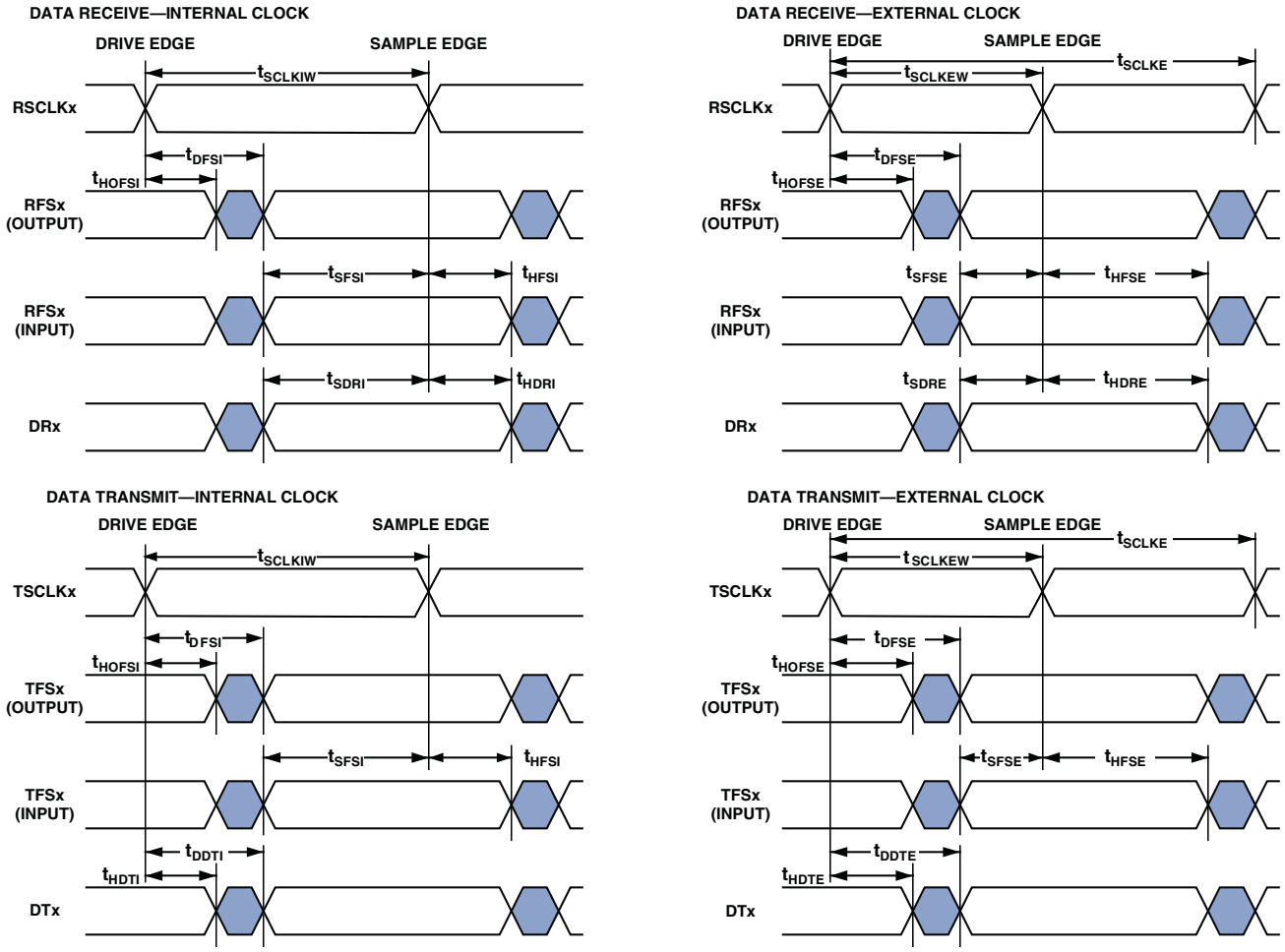


Figure 20. Serial Ports

Table 32. Serial Ports—Enable and Three-State

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_{DTENE}	Data Enable Delay from External TSCLKx[1]	0.0		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]	–2.0		–2.0		ns
t_{DDTTI}	Data Disable Delay from Internal TSCLKx[1]		$t_{SCLK}+1$		$t_{SCLK}+1$	ns

[1] Referenced to drive edge.

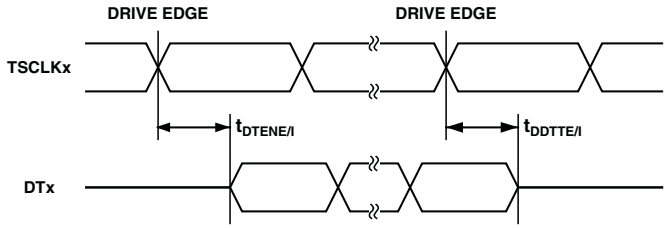


Figure 21. Serial Ports — Enable and Three-State

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.

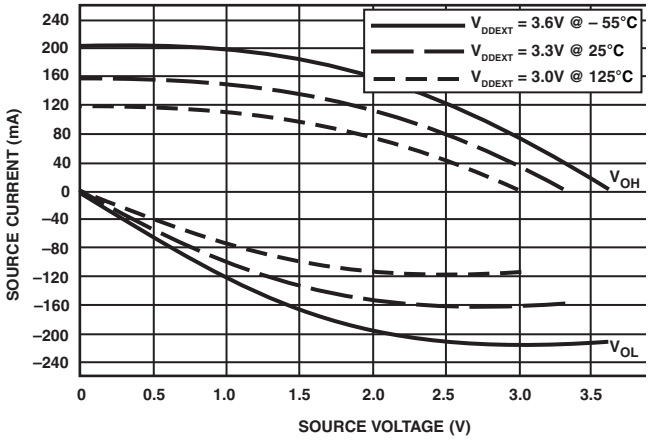


Figure 32. Driver Type B Current (3.3 V V_{DDEXT})

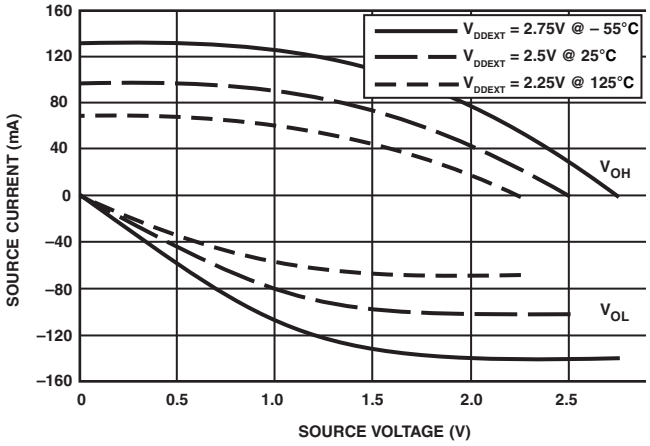


Figure 33. Driver Type B Current (2.5 V V_{DDEXT})

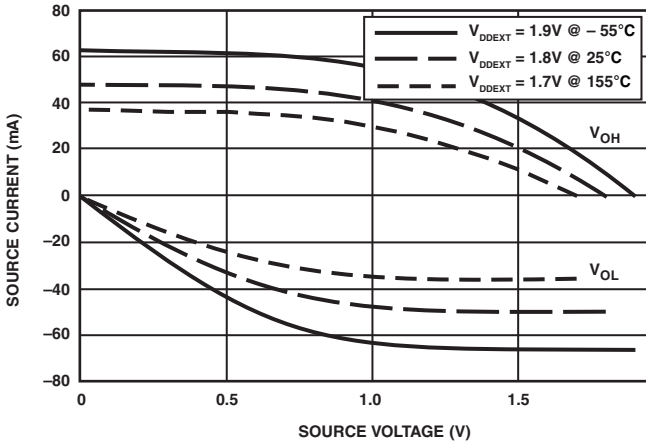


Figure 34. Driver Type B Current (1.8 V V_{DDEXT})

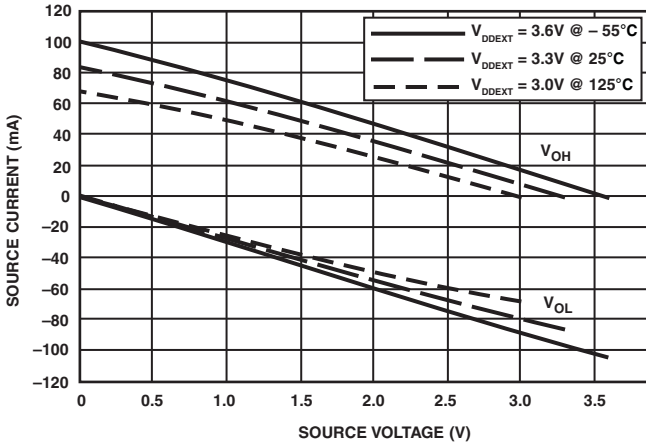


Figure 35. Driver Type C Current (3.3 V V_{DDEXT})

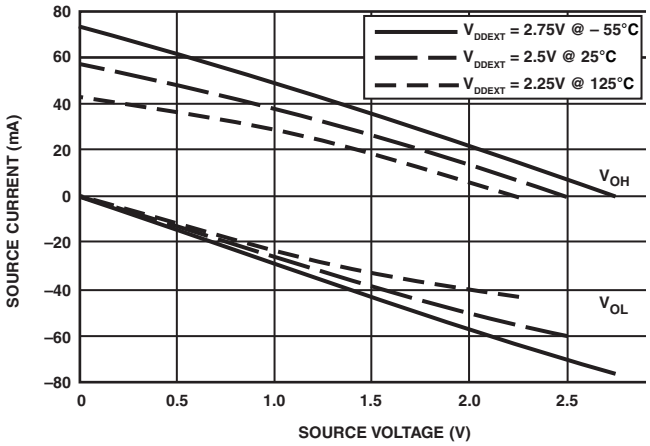


Figure 36. Drive Type C Current (2.5 V V_{DDEXT})

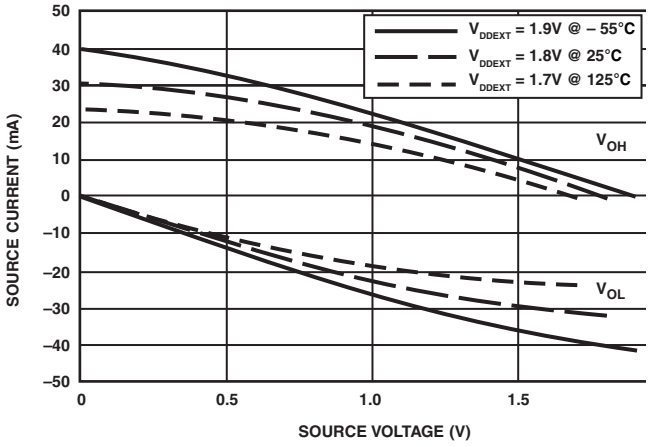


Figure 37. Driver Type C Current (1.8 V V_{DDEXT})

FLASH—SPECIFICATIONS

Specifications subject to change without notice.

FLASH—PROGRAM AND ERASE TIMES AND ENDURANCE CYCLES

The program and erase times and the number of program/ erase cycles per block are shown in Table 45. Exact erase times may change depending on the memory array condition. The best case is when all the bits in the block or bank are at '0' (pre programmed). The worst case is when all the bits in the block or bank are at '1' (not pre programmed). Usually, the system overhead is negligible with respect to the erase time.

Table 45. Program/Erase Times and Endurance Cycles

Parameter	Condition	Typical	Typical After 100k Write/Erase Cycles	Max	Unit
Erase	Parameter Block (4K word)[1]	0.3	1	2.5	s
Erase	Main Block (32K word)—preprogrammed	0.8	3	4	s
Erase	Main Block (32K word)—not preprogrammed	1		4	s
Program[2]	Word	12	12	100	µs
Program[2]	Parameter Block (4K word)	40			ms
Program[2]	Main Block (32K word)	300			ms
Suspend Latency	Program	5		10	µs
Suspend Latency	Erase	5		20	µs
Program/Erase Cycles (per Block)	Main Blocks			100,000	Cycles
Program/Erase Cycles (per Block)	Parameter Blocks			100,000	Cycles

[1] The difference between pre programmed and not pre programmed is not significant (< 30 ms).
[2] Values are liable to change with the external system-level overhead (command sequence and Status Register polling execution).

FLASH—ABSOLUTE MAXIMUM RATINGS

Table 46 shows the ADC absolute maximum ratings.

Table 46. Flash Absolute Maximum Ratings

Parameter	Rating
Junction Temperature While Biased	See Table 20 on Page 31
Storage Temperature Range	See Table 20 on Page 31
Flash Memory Supply Voltage ($V_{DDFLASH}$)	–0.2 V to +2.45 V

Table 47. Operating Conditions (Analog, Voltage Reference, and Logic I/O) (Continued)

Parameter	Specification	Unit	Test Conditions/Comments
DIGITAL LOGIC OUTPUTS			
Output High Voltage, V_{OH}	V_{DRIVE} – 0.2	V min	No DC load (I_{OH} = 0 mA)
Output Low Voltage, V_{OL}	0.4	V max	No DC load (I_{OL} = 0 mA)
Floating State Leakage Current	±1	µA max	V_{IN} = 0 V or V_{DRIVE}
Floating State Output Capacitance[4]	7	pF typ	
Output Coding[8]	Straight (natural) binary twos complement		

[1] V_{IN-} or V_{IN+} must remain within GND/V_{DD}.
[2] V_{IN-} = 0 V for specified performance. For full input range on V_{IN-} pin, see Figure 74 and Figure 75.
[3] For full common-mode range, see Figure 70 and Figure 71.
[4] Sample tested during initial release to ensure compliance.
[5] Relates to Pin D_{CAP}A or Pin D_{CAP}B (V_{REF}).
[6] See ADC—Terminology on Page 61.
[7] External voltage reference applied to Pins D_{CAP}A, Pin D_{CAP}B (V_{REF}).
[8] See Table 52 and Table 53.

Table 48. Operating Conditions (ADC Performance/Accuracy)

Parameter	Specification	Unit	Test Conditions/Comments
DYNAMIC PERFORMANCE			
Signal-to-Noise Ratio (SNR)	71	dB min	f_{IN} = 14 kHz sine wave; differential mode
	69	dB min	f_{IN} = 14 kHz sine wave; single-ended and pseudo differential modes
Signal-to-(Noise + Distortion) Ratio (SINAD)[1]	70	dB min	f_{IN} = 14 kHz sine wave; differential mode
	68	dB min	f_{IN} = 14 kHz sine wave; single-ended and pseudo differential modes
Total Harmonic Distortion (THD)[1]	–77	dB max	f_{IN} = 14 kHz sine wave; differential mode
	–73	dB max	f_{IN} = 14 kHz sine wave; single-ended and pseudo differential modes
Spurious-Free Dynamic Range (SFDR)[1]	–75	dB max	f_{IN} = 50 kHz sine wave
Intermodulation Distortion (IMD)[1,2]			f_a = 30 kHz, fb = 50 kHz
Second-Order Terms	–88	dB typ	
Third-Order Terms	–88	dB typ	
Channel-to-Channel Isolation	–88	dB typ	
SAMPLE AND HOLD			
Aperture Delay[2]	11	ns max	
Aperture Jitter[2]	50	ps typ	
Aperture Delay Matching[2]	200	ps max	
Full Power Bandwidth	33/26	MHz typ	@ 3 dB, AV_{DD}, DV_{DD} = 5 V/AV_{DD}, DV_{DD} = 3 V
	3.5/3	MHz typ	@ 0.1 dB, AV_{DD}, DV_{DD} = 5 V/AV_{DD}, DV_{DD} = 3 V

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.

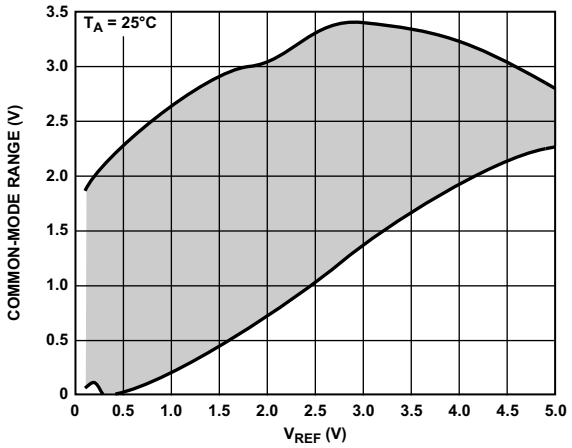


Figure 70. Input Common-Mode Range vs. V_{REF} (0 to V_{REF} Range, V_{DD} = 5 V)

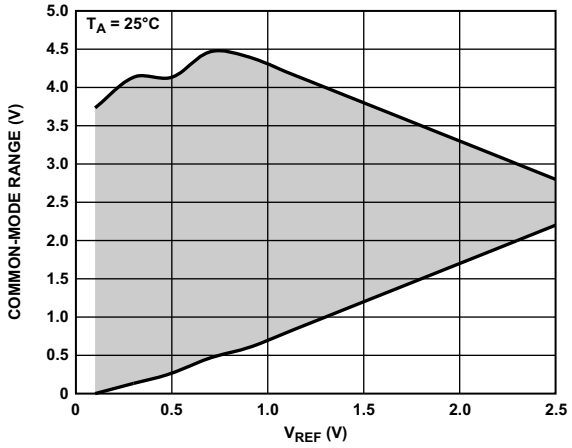


Figure 71. Input Common-Mode Range vs. V_{REF} (2 × V_{REF} Range, V_{DD} = 5 V)

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.

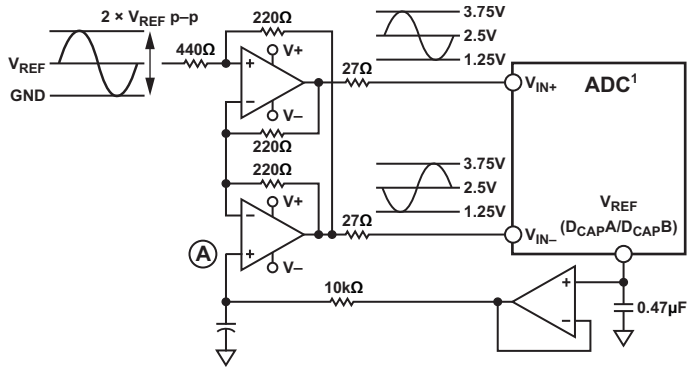
Using an Op Amp Pair

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)

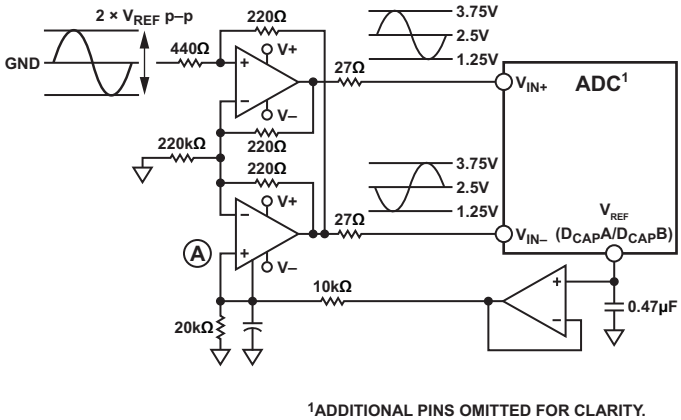


Figure 73. Dual Op Amp Circuit to Convert a Single-Ended Bipolar Signal into a Differential Unipolar Signal

to make use of the full dynamic range of the part. A dc input is applied to the V_{IN-} pin. The voltage applied to this input provides an offset from ground or a pseudo ground for the V_{IN+} input. The benefit of pseudo differential inputs is that they separate the analog input signal ground from the ADC's ground allowing dc common-mode voltages to be cancelled.

The typical voltage range for the V_{IN-} pin, while in pseudo differential mode, is shown in Figure 74 (V_{IN-} Input Voltage Range vs. V_{REF} in Pseudo Differential Mode with V_{DD} = 3 V) and Figure 75 (V_{IN-} Input Voltage Range vs. V_{REF} in Pseudo Differential Mode with V_{DD} = 5 V). Figure 76 (Pseudo Differential Mode Connection Diagram) shows a connection diagram for pseudo differential mode.

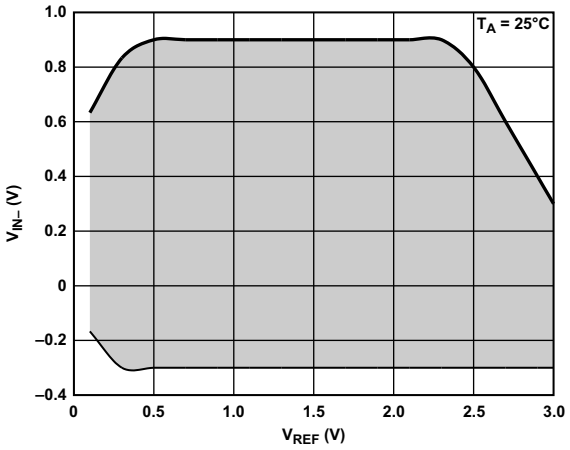


Figure 74. V_{IN-} Input Voltage Range vs. V_{REF} in Pseudo Differential Mode with V_{DD} = 3 V

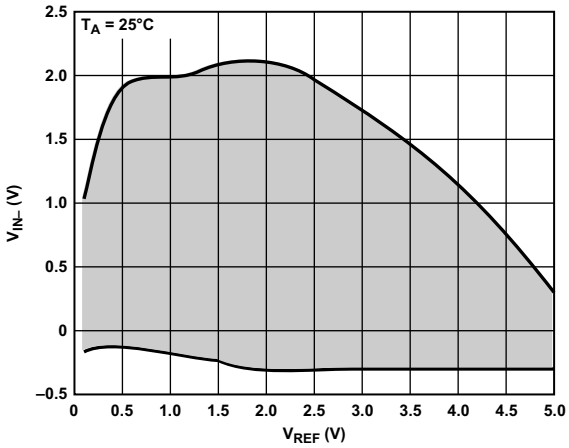


Figure 75. V_{IN-} Input Voltage Range vs. V_{REF} in Pseudo Differential Mode with V_{DD} = 5 V

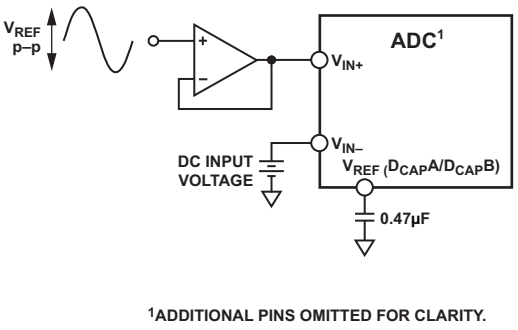


Figure 76. Pseudo Differential Mode Connection Diagram

Analog Input Selection

The analog inputs of the ADC can be configured as single-ended or true differential via the SGL/$\overline{\text{DIFF}}$ logic pin, as shown in Figure 77 (Selecting Differential or Single-Ended Configuration). If this pin is tied to a logic low, the analog input channels to each on-chip ADC are set up as three true differential pairs. If this pin is at logic high, the analog input channels to each on-chip ADC are set up as six single-ended analog inputs. The required logic level on this pin needs to be established prior to the acquisition time and remain unchanged during the conversion time until the track-and-hold has returned to track. The track-and-hold returns to track on the 13th rising edge of ADSCLK after the $\overline{\text{CS}}$ falling edge (see Figure 87 (Serial Interface Timing Diagram)). If the level on this pin is changed, it will be recognized by the ADC; therefore, it is necessary to keep the same logic level during acquisition and conversion to avoid corrupting the conversion in progress.

For example, in Figure 77 (Selecting Differential or Single-Ended Configuration) the SGL/$\overline{\text{DIFF}}$ pin is set at logic high for the duration of both the acquisition and conversion times so the analog inputs are configured as single ended for that conversion (Sampling Point A). The logic level of the SGL/$\overline{\text{DIFF}}$ changed to low after the track-and-hold returned to track and prior to the