Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Blackfin Embedded Processor

ADSP-BF504/ADSP-BF504F/ADSP-BF506F

FEATURES

- Up to 400 MHz high performance Blackfin processor
  - Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bit shifter
  - RISC-like register and instruction model for ease of programming and compiler-friendly support
  - Advanced debug, trace, and performance monitoring
- Accepts a range of supply voltages for internal and I/O operations. See Operating Conditions on Page 26
- Internal 32M bit flash (available on ADSP-BF504F and ADSP-BF506F processors)
- Internal ADC (available on ADSP-BF506F processor)
- Off-chip voltage regulator interface
- 88-lead (12 mm × 12 mm) LFCSP package for ADSP-BF504 and ADSP-BF504F processors
- 120-lead (14 mm × 14 mm) LQFP package for ADSP-BF506F processor

MEMORY

- 68K bytes of L1 SRAM (processor core-accessible) memory (See Table 1 on Page 3 for L1 and L3 memory size details)
- External (interface-accessible) memory controller with glueless support for internal 32M bit flash and boot ROM
- Flexible booting options from internal flash and SPI memory or from host devices including SPI, PPI, and UART
- Memory management unit providing memory protection

PERIPHERALS

- Two 32-bit up/down counters with rotary support
- Eight 32-bit timers/counters with PWM support
- Two 3-phase 16-bit center-based PWM units
- 2 dual-channel, full-duplex synchronous serial ports (SPORTs), supporting eight stereo I^2S channels
- 2 serial peripheral interface (SPI) compatible ports
- 2 UARTs with IrDA support
- Parallel peripheral interface (PPI), supporting ITU-R 656 video data formats
- Removable storage interface (RSI) controller for MMC, SD, SDIO, and CE-ATA
- Internal ADC with 12 channels, 12 bits, and up to 2 MSPS
- ADC controller module (ACM), providing a glueless interface between Blackfin processor and internal or external ADC
- Controller Area Network (CAN) controller
- 2-wire interface (TWI) controller
- 12 peripheral DMAs
- 2 memory-to-memory DMA channels
- Event handler with 52 interrupt inputs
- 35 general-purpose I/Os (GPIOs), with programmable hysteresis
- Debug/JTAG interface
- On-chip PLL capable of frequency multiplication

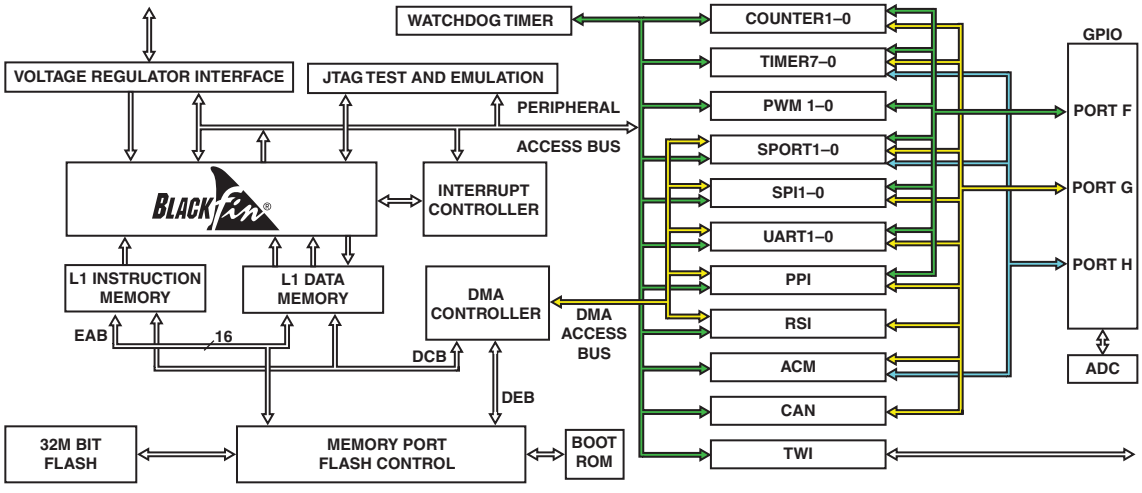


Figure 1. Processor Block Diagram

Rev. B Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106 U.S.A.
Tel: 781.329.4700 **Technical Support**

www.analog.com

TABLE OF CONTENTS

REVISION HISTORY

04/14—Rev. A to Rev. B

Updated Development Tools 17

Corrected RCKFE bit setting and description in Table 9, The SPORTx Receive Configuration 1 Register (SPORTx_RCR1) 19

Updated footnote 6 in Operating Conditions 26

Updated Table 18 with revised data for Static Current—IDD-DEEPSLEEP (mA) 30

Revised package diagram (Figure 93) to include U-Groove in Outline Dimensions 79

Package thickness changed from 0.75/0.80/0.85 to 0.75/0.85/0.90 in Figure 94 in Outline Dimensions 79

GENERAL DESCRIPTION

The ADSP-BF50x processors are members of the Blackfin® family of products, incorporating the Analog Devices/Intel Micro Signal Architecture (MSA). Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction-set architecture.

The ADSP-BF50x processors are completely code compatible with other Blackfin processors. ADSP-BF50x processors offer performance up to 400 MHz and reduced static power consumption. Differences with respect to peripheral combinations are shown in Table 1.

Table 1. Processor Comparison

Feature		ADSP-BF504	ADSP-BF504F	ADSP-BF506F
Up/Down/Rotary Counters		2	2	2
Timer/Counters with PWM		8	8	8
3-Phase PWM Units		2	2	2
SPORTs		2	2	2
SPIs		2	2	2
UARTs		2	2	2
Parallel Peripheral Interface		1	1	1
Removable Storage Interface		1	1	1
CAN		1	1	1
TWI		1	1	1
Internal 32M Bit Flash		–	1	1
ADC Control Module (ACM)		1	1	1
Internal ADC		–	–	1
GPIOs		35	35	35
Memory (bytes)	L1 Instruction SRAM	16K	16K	16K
	L1 Instruction SRAM/Cache	16K	16K	16K
	L1 Data SRAM	16K	16K	16K
	L1 Data SRAM/Cache	16K	16K	16K
	L1 Scratchpad	4K	4K	4K
	L3 Boot ROM	4K	4K	4K
Maximum Speed Grade[1]		400 MHz		
Maximum System Clock Speed		100 MHz		
Package Options		88-Lead LFCSP	88-Lead LFCSP	120-Lead LQFP

[1] For valid clock combinations, see Table 14, Table 15, Table 16, and Table 24.

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

PORTABLE LOW-POWER ARCHITECTURE

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

SYSTEM INTEGRATION

The ADSP-BF50x processors are highly integrated system-on-a-chip solutions for the next generation of embedded industrial, instrumentation, and power/motion control applications. By combining industry-standard interfaces with a high performance signal processing core, cost-effective applications can be developed quickly, without the need for costly external components. The system peripherals include a watchdog timer; two 32-bit up/down counters with rotary support; eight 32-bit timers/counters with PWM support; six pairs of 3-phase 16-bit center-based PWM units; two dual-channel, full-duplex synchronous serial ports (SPORTs); two serial peripheral interface (SPI) compatible ports; two UARTs with IrDA® support; a parallel peripheral interface (PPI); a removable storage interface (RSI) controller; an internal ADC with 12 channels, 12 bits, up to 2 MSPS, and ACM controller; a controller area network (CAN) controller; a 2-wire interface (TWI) controller; and an internal 32M bit flash.

PROCESSOR PERIPHERALS

The ADSP-BF50x processors contain a rich set of peripherals connected to the core via several high-bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see the block diagram on Page 1). These Blackfin processors contain high-speed serial and parallel ports, an interrupt controller for flexible management of interrupts from the on-chip peripherals or external sources, and power management control functions to tailor the performance and power characteristics of the processor and system to many application scenarios.

The SPORT, SPI, UART, PPI, and RSI peripherals are supported by a flexible DMA structure. There are also separate memory DMA channels dedicated to data transfers between the processor's various memory spaces, including boot ROM and internal 32M bit synchronous burst flash. Multiple on-chip buses running at up to 100 MHz provide enough bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

The ADSP-BF50x processors include an interface to an off-chip voltage regulator in support of the processor's dynamic power management capability.

BLACKFIN PROCESSOR CORE

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

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

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

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

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

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

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

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

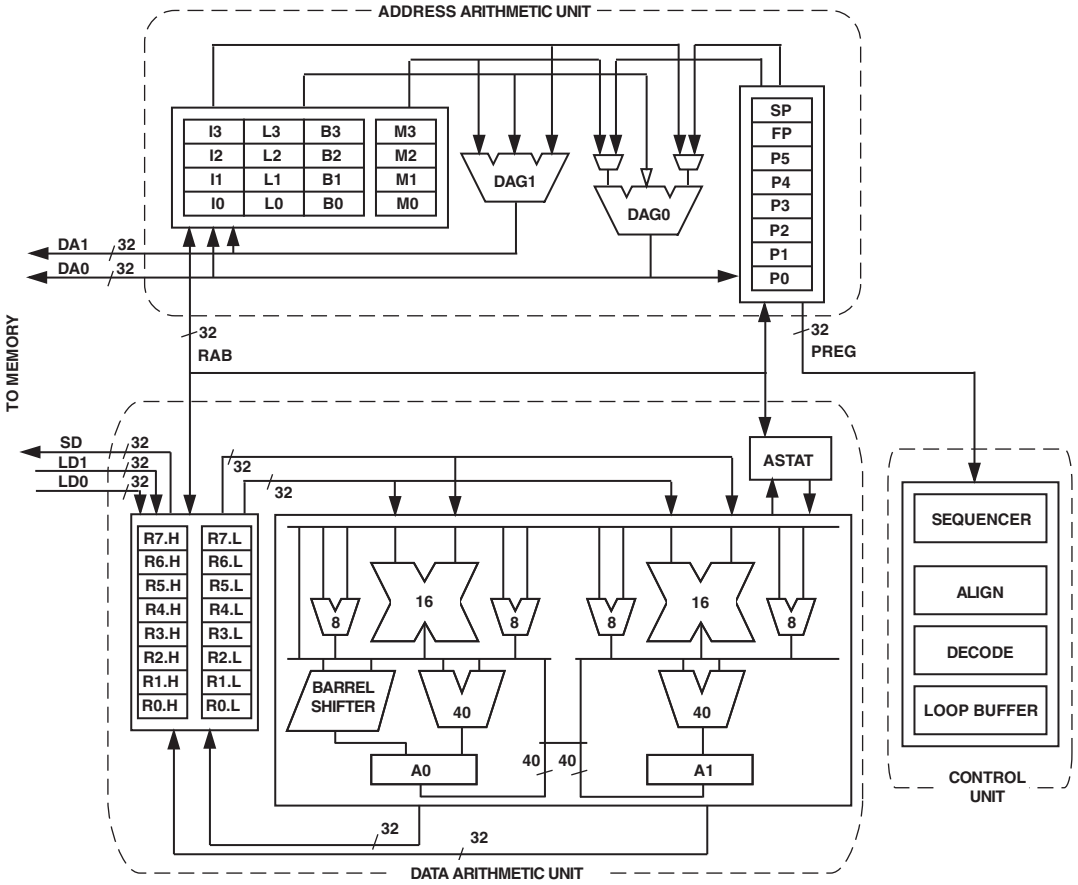


Figure 2. Blackfin Processor Core

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.

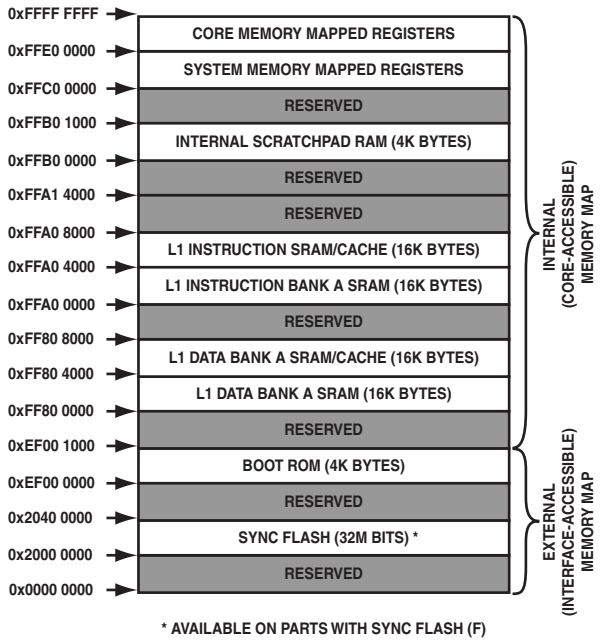


Figure 3. Internal/External Memory Map

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

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.

FLASH MEMORY

The ADSP-BF504F and ADSP-BF506F processors include an on-chip 32M bit (×16, multiple bank, burst) Flash memory. The features of this memory include:

- Synchronous/asynchronous read
  - Synchronous burst read mode: 50 MHz
  - Asynchronous/synchronous read mode
  - Random access times: 70 ns
- Synchronous burst read suspend
- Memory blocks
  - Multiple bank memory array: 4M bit banks
  - Parameter blocks (top location)
- Dual operations
  - Program erase in one bank while read in others
  - No delay between read and write operations
- Block locking
  - All blocks locked at power-up
  - Any combination of blocks can be locked or locked down
- Security
  - 128-bit user programmable OTP cells
  - 64-bit unique device number
- Common Flash interface (CFI)
- 100,000 program/erase cycles per block

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.

DMA CONTROLLERS

The processor has multiple, independent DMA channels that support automated data transfers with minimal overhead for the processor core. DMA transfers can occur between the processor's internal memories and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interface. DMA-capable peripherals include the SPORTs, SPI ports, UARTs, RSI, and PPI. Each individual DMA-capable peripheral has at least one dedicated DMA channel.

The processor DMA controller supports both one-dimensional (1-D) and two-dimensional (2-D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2-D DMA capability supports arbitrary row and column sizes up to 64K elements by 64K elements, and arbitrary row and column step sizes up to ±32K elements. Furthermore, the column step size can be less than the row step size, allowing implementation of interleaved data streams. This feature is especially useful in video applications where data can be de-interleaved on the fly.

Examples of DMA types supported by the processor DMA controller include:

- A single, linear buffer that stops upon completion
- A circular, auto-refreshing buffer that interrupts on each full or fractionally full buffer
- 1-D or 2-D DMA using a linked list of descriptors
- 2-D DMA using an array of descriptors, specifying only the base DMA address within a common page

In addition to the dedicated peripheral DMA channels, there are two memory DMA channels, which are provided for transfers between the various memories of the processor system with minimal processor intervention. Memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

WATCHDOG TIMER

The processor includes a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state through generation of a core and system reset, nonmaskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

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

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

TIMERS

There are nine general-purpose programmable timer units in the processors. Eight timers have an external pin that can be configured either as a pulse width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input to the several other associated PF pins, to an external clock input to the PPI_CLK input pin, or to the internal SCLK.

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

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

In addition to the 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.

- 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.

Each mailbox consists of eight 16-bit data words. The data is divided into fields, which includes a message identifier, a time stamp, a byte count, up to 8 bytes of data, and several control bits. Each node monitors the messages being passed on the network. If the identifier in the transmitted message matches an identifier in one of its mailboxes, the module knows that the message was meant for it, passes the data into its appropriate mailbox, and signals the processor of message arrival with an interrupt.

The CAN controller can wake up the processor from sleep mode upon generation of a wake-up event, such that the processor can be maintained in a low-power mode during idle conditions. Additionally, a CAN wake-up event can wake up the on-chip internal voltage regulator from the powered-down hibernate state.

The electrical characteristics of each network connection are very stringent. Therefore, the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The ADSP-BF50x CAN module represents the controller part of the interface. This module's network I/O is a single transmit output and a single receive input, which connect to a line transceiver.

The CAN clock is derived from the processor system clock (SCLK) through a programmable divider and therefore does not require an additional crystal.

TWI CONTROLLER INTERFACE

The processors include a 2-wire interface (TWI) module for providing a simple exchange method of control data between multiple devices. The TWI is compatible with the widely used $I^2C^®$ bus standard. The TWI module offers the capabilities of simultaneous master and slave operation, support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (SCL) and data (SDA) and supports the protocol at speeds up to 400K bits/sec. The TWI interface pins are compatible with 5 V logic levels.

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

PORTS

Because of the rich set of peripherals, the processor groups the many peripheral signals to three ports—Port F, Port G, and Port H. Most of the associated pins are shared by multiple signals. The ports function as multiplexer controls.

General-Purpose I/O (GPIO)

The processor has 35 bidirectional, general-purpose I/O (GPIO) pins allocated across three separate GPIO modules—PORTFIO, PORTGIO, and PORTHIO, associated with Port F, Port G, and Port H, respectively. Each GPIO-capable pin shares functionality with other processor peripherals via a multiplexing scheme; however, the GPIO functionality is the default state of the device upon power-up. Neither GPIO output nor input drivers are active by default. Each general-purpose port pin can be individually controlled by manipulation of the port control, status, and interrupt registers:

- GPIO direction control register – Specifies the direction of each individual GPIO pin as input or output.
- GPIO control and status registers – The processor employs a "write one to modify" mechanism that allows any combination of individual GPIO pins to be modified in a single instruction, without affecting the level of any other GPIO pins. Four control registers are provided. One register is written in order to set pin values, one register is written in order to clear pin values, one register is written in order to toggle pin values, and one register is written in order to specify a pin value. Reading the GPIO status register allows software to interrogate the sense of the pins.
- GPIO interrupt mask registers – The two GPIO interrupt mask registers allow each individual GPIO pin to function as an interrupt to the processor. Similar to the two GPIO control registers that are used to set and clear individual pin values, one GPIO interrupt mask register sets bits to enable interrupt function, and the other GPIO interrupt mask register clears bits to disable interrupt function. GPIO pins defined as inputs can be configured to generate hardware interrupts, while output pins can be triggered by software interrupts.
- GPIO interrupt sensitivity registers – The two GPIO interrupt sensitivity registers specify whether individual pins are level- or edge-sensitive and specify—if edge-sensitive—whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge-sensitivity.

DYNAMIC POWER MANAGEMENT

The processor provides five operating modes, each with a different performance/power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. When configured for a 0 volt core supply voltage, the processor enters the hibernate state. Control of clocking to each of the processor peripherals also reduces power consumption. See Table 4 for a summary of the power settings for each mode.

Full-On Operating Mode—Maximum Performance

In the full-on mode, the PLL is enabled and is not bypassed, providing capability for maximum operational frequency. This is the power-up default execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

Active Operating Mode—Moderate Dynamic Power Savings

In the active mode, the PLL is enabled but bypassed. Because the PLL is bypassed, the processor's core clock (CCLK) and system clock (SCLK) run at the input clock (CLKIN) frequency. DMA access is available to appropriately configured L1 memories.

In the active mode, it is possible to disable the control input to the PLL by setting the PLL_OFF bit in the PLL control register. This register can be accessed with a user-callable routine in the on-chip ROM called bfrom_SysControl(). If disabled, the PLL control input must be re-enabled before transitioning to the full-on or sleep modes.

Table 4. Power Settings

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

For more information about PLL controls, see the "Dynamic Power Management" chapter in the *ADSP-BF50x Blackfin Processor Hardware Reference*.

Sleep Operating Mode—High Dynamic Power Savings

The sleep mode reduces dynamic power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK), however, continue to operate in this mode. Typically, an external event wakes up the processor. When in the sleep mode, asserting a wakeup enabled in the SIC_IWRx registers causes the processor to sense the value of the BYPASS bit in the PLL control register (PLL_CTL). If BYPASS is disabled, the processor transitions to the full on mode. If BYPASS is enabled, the processor transitions to the active mode.

DMA accesses to L1 memory are not supported in sleep mode.

Deep Sleep Operating Mode—Maximum Dynamic Power Savings

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

Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all of the peripherals (SCLK). This setting sets the internal power supply voltage (V_{DDINT}) to 0 V to provide the lowest static power dissipation. Any critical information stored internally (for example, memory contents, register contents, and other information) must be written to a non-volatile storage device prior to removing power if the processor state is to be preserved.

Writing 0 to the HIBERNATE bit causes EXT_WAKE to transition low, which can be used to signal an external voltage regulator to shut down.

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

The processor can be woken up by asserting the $\overline{\text{RESET}}$ pin. All hibernate wakeup events initiate the hardware reset sequence. Individual sources are enabled by the VR_CTL register. The EXT_WAKE signal indicates the occurrence of a wakeup event.

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

Power Savings

As shown in Table 5, the processor supports three different power domains, which maximizes flexibility while maintaining compliance with industry standards and conventions. By isolating the internal logic of the processor into its own power domain, separate from other I/O, the processor can take advantage of dynamic power management without affecting the other I/O devices. There are no sequencing requirements for the various power domains, but all domains must be powered according to the appropriate Specifications table for processor operating conditions; even if the feature/peripheral is not used.

Table 5. Power Domains

Power Domain	Power Supply
All internal logic, except Memory	V_{DDINT}
Flash Memory	$V_{DDFLASH}$
All other I/O	V_{DDEXT}
ADC digital supply[1] (Logic, I/O)	DV_{DD}, V_{DRIVE}
ADC analog supply[1]	AV_{DD}

[1] On ADSP-BF506F processor only.

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

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

Power Savings Factor

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

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

where the variables in the equations are:

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

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

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

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

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

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

ADSP-BF50x VOLTAGE REGULATION

The ADSP-BF50x processors require an external voltage regulator to power the V_{DDINT} domain. To reduce standby power consumption, the external voltage regulator can be signaled through EXT_WAKE to remove power from the processor core. This signal is high-true for power-up and may be connected directly to the low-true shut-down input of many common regulators.

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

The power good ($\overline{\text{PG}}$) input signal allows the processor to start only after the internal voltage has reached a chosen level. In this way, the startup time of the external regulator is detected after hibernation. For a complete description of the power good functionality, refer to the *ADSP-BF50x Blackfin Processor Hardware Reference*.

CLOCK SIGNALS

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

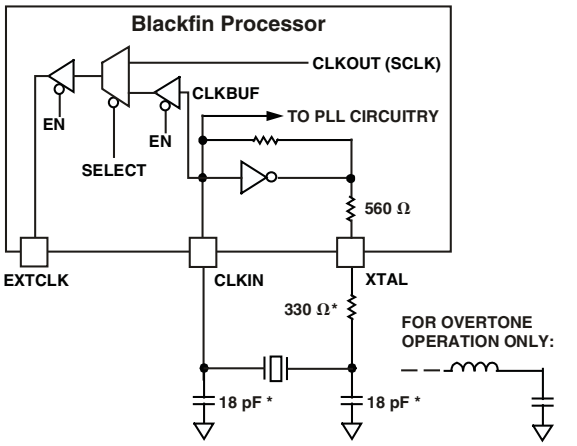
If an external clock is used, it should be a TTL-compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor's CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the processor includes an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 4. A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor shown in Figure 4 fine tune phase and amplitude of the sine frequency.

The capacitor and resistor values shown in Figure 4 are typical values only. The capacitor values are dependent upon the crystal manufacturers' load capacitance recommendations and the PCB physical layout. The resistor value depends on the drive level specified by the crystal manufacturer. The user should verify the customized values based on careful investigations on multiple devices over temperature range.

A third-overtone crystal can be used for frequencies above 25 MHz. The circuit is then modified to ensure crystal operation only at the third overtone by adding a tuned inductor circuit as shown in Figure 4. A design procedure for third-overtone operation is discussed in detail in *(EE-168) Using Third Overtone Crystals with the ADSP-218x DSP* on the Analog Devices website (www.analog.com)—use site search on "EE-168."

The Blackfin core runs at a different clock rate than the on-chip peripherals. As shown in Figure 5, the core clock (CCLK) and system peripheral clock (SCLK) are derived from the input clock (CLKIN) signal. An on-chip PLL is capable of multiplying the CLKIN signal by a programmable multiplication factor (bounded by specified minimum and maximum VCO frequencies). The default multiplier is 6×, but it can be modified by a software instruction sequence.



NOTE: VALUES MARKED WITH * MUST BE CUSTOMIZED, DEPENDING ON THE CRYSTAL AND LAYOUT. PLEASE ANALYZE CAREFULLY. FOR FREQUENCIES ABOVE 33 MHz, THE SUGGESTED CAPACITOR VALUE OF 18 pF SHOULD BE TREATED AS A MAXIMUM, AND THE SUGGESTED RESISTOR VALUE SHOULD BE REDUCED TO 0 Ω.

Figure 4. External Crystal Connections

On-the-fly frequency changes can be effected by simply writing to the PLL_DIV register. The maximum allowed CCLK and SCLK rates depend on the applied voltages V_{DDINT} and V_{DDEXT}; the VCO is always permitted to run up to the CCLK frequency specified by the part's speed grade. The EXTCLK pin can be configured to output either the SCLK frequency or the input buffered CLKIN frequency, called CLKBUF. When configured to output SCLK (CLKOUT), the EXTCLK pin acts as a reference signal in many timing specifications. While active by default, it can be disabled using the EBIU_AMGCTL register.

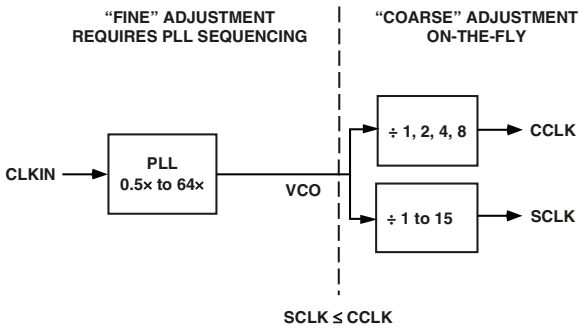


Figure 5. Frequency Modification Methods

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 $\overline{\text{SPI0\_SEL1}}$ 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 $\overline{\text{SPI0\_SS}}$ 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 $\overline{\text{UA0\_RTS}}$ output to hold off the host; $\overline{\text{UA0\_CTS}}$ 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 other Analog Devices IDE, VisualDSP++, supports processor families introduced prior to the release of CrossCore Embedded Studio. This IDE includes the Analog Devices VDK real time operating system and an open source TCP/IP stack. For more information visit www.analog.com/visualdsp. Note that VisualDSP++ will not support future Analog Devices processors.

EZ-KIT Lite Evaluation Board

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

EZ-KIT Lite Evaluation Kits

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

Software Add-Ins for CrossCore Embedded Studio

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

Board Support Packages for Evaluation Hardware

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

Middleware Packages

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

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

Algorithmic Modules

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

Designing an Emulator-Compatible DSP Board (Target)

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

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

ADC AND ACM INTERFACE

This section describes the ADC and ACM interface. System designers should also consult the *ADSP-BF50x Blackfin Processor Hardware Reference* for additional information.

The ADC control module (ACM) provides an interface that synchronizes the controls between the processor and the internal analog-to-digital converter (ADC) module. The ACM is available on the ADSP-BF504, ADSP-BF504F, and ADSP-BF506F processors, and the ADC is available on the ADSP-BF506F processor only. The analog-to-digital conversions are initiated by the processor, based on external or internal events.

The ACM allows for flexible scheduling of sampling instants and provides precise sampling signals to the ADC.

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.

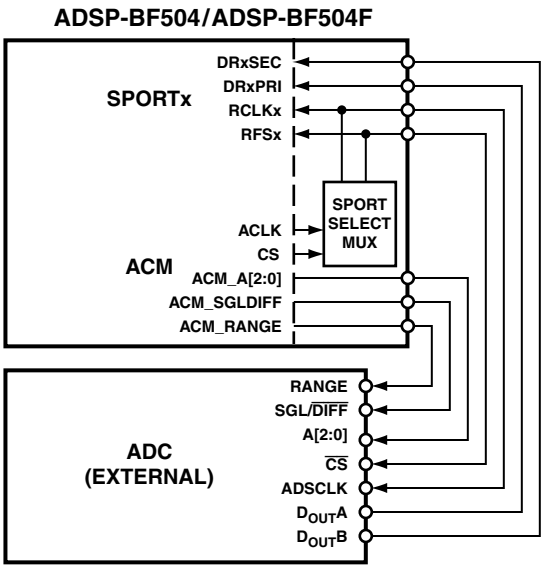


Figure 6. ADC (External), ACM, and SPORT Connections

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.

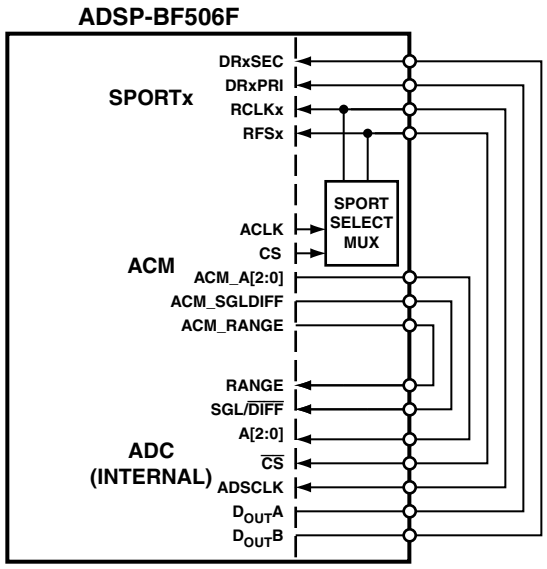


Figure 7. ADC (Internal), ACM, and SPORT Connections

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	

NOTE: The SPORT must be enabled with the following settings: external clock, external frame sync, and active low frame sync.

Table 10. The SPORTx Receive Configuration 2 Register (SPORTx_RCR2)

Setting	Description
RXSE = 1	Secondary side enabled
SLEN = 1111	16-bit data-word (or may be set to 1101 for 14-bit data-word)

To implement the power-down modes, SLEN should be set to 1001 to issue an 8-bit SCLK burst. A Blackfin driver for the ADC is available to download at www.analog.com.

INTERNAL ADC

An ADC is integrated into the ADSP-BF506F product. All ADC signals are connected out to package pins to enable maximum interconnect flexibility in mixed signal applications.

The internal ADC is a dual, 12-bit, high speed, low power, successive approximation ADC that operates from a single 2.7 V to 5.25 V power supply and features throughput rates up to 2 MSPS. The device contains two ADCs, each preceded by a 3-channel multiplexer, and a low noise, wide bandwidth track-and-hold amplifier that can handle input frequencies in excess of 30 MHz.

Figure 8 shows the functional block diagram of the internal ADC. The ADC features include:

- Dual 12-bit, 3-channel ADC
- Throughput rate: up to 2 MSPS
- Specified for DV_{DD} and AV_{DD} of 2.7 V to 5.25 V
- Pin-configurable analog inputs
  - 12-channel single-ended inputs

    or
  - 6-channel fully differential inputs

    or
  - 6-channel pseudo differential inputs
- Accurate on-chip voltage reference: 2.5 V
- Dual conversion with read 437.5 ns, 32 MHz ADSCLK
- High speed serial interface
  - SPI-/QSPI™-/MICROWIRE™-/DSP-compatible
- Low power shutdown mode

The conversion process and data acquisition use standard control inputs allowing easy interfacing to microprocessors or DSPs. The input signal is sampled on the falling edge of $\overline{CS}$; conversion is also initiated at this point. The conversion time is determined by the ADSCLK frequency. There are no pipelined delays associated with the part.

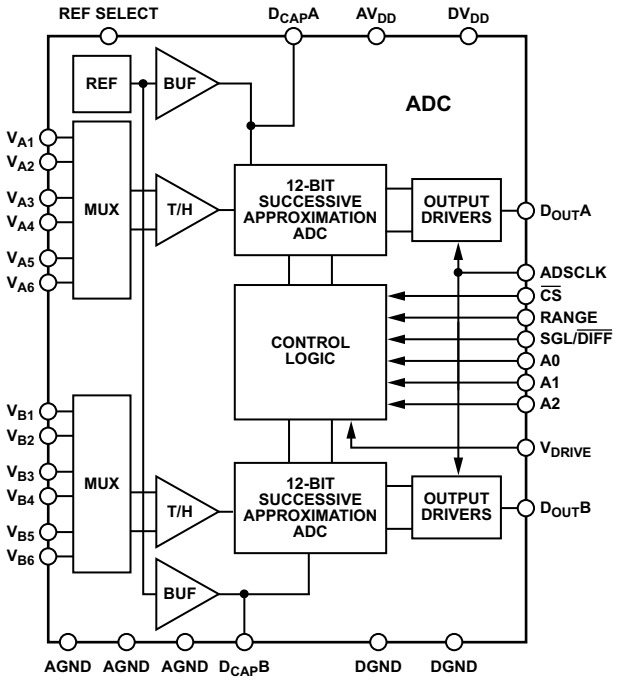


Figure 8. ADC (Internal) Functional Block Diagram

The internal ADC uses advanced design techniques to achieve very low power dissipation at high throughput rates. The part also offers flexible power/throughput rate management when operating in normal mode as the quiescent current consumption is so low.

The analog input range for the part can be selected to be a 0 V to V_{REF} (or 2 × V_{REF}) range, with either straight binary or twos complement output coding. The internal ADC has an on-chip 2.5 V reference that can be overdriven when an external reference is preferred.

Additional highlights of the internal ADC include:

- Two complete ADC functions allow simultaneous sampling and conversion of two channels—Each ADC has three fully/pseudo differential pairs, or six single-ended channels, as programmed. The conversion result of both channels is simultaneously available on separate data lines, or in succession on one data line if only one serial connection is available.
- High throughput with low power consumption
- The internal ADC offers both a standard 0 V to V_{REF} input range and a 2 × V_{REF} input range.
- No pipeline delay—The part features two standard successive approximation ADCs with accurate control of the sampling instant via a $\overline{CS}$ input and once off conversion control.

ADC APPLICATION HINTS

The following sections provide application hints for using the ADC.

Grounding and Layout Considerations

The analog and digital supplies to the ADC are independent and separately pinned out to minimize coupling between the analog and digital sections of the device. The printed circuit board (PCB) that houses the ADC should be designed so that the analog and digital sections are separated and confined to certain areas of the board. This design facilitates the use of ground planes that can be easily separated.

To provide optimum shielding for ground planes, a minimum etch technique is generally best. All AGND pins should be sunk in the AGND plane. Digital and analog ground planes should be joined in only one place. If the ADC is in a system where multiple devices require an AGND to DGND connection, the connection should still be made at one point only, a star ground point that should be established as close as possible to the ground pins on the ADC.

Avoid running digital lines under the device as this couples noise onto the die. Avoid running digital lines in the area of the AGND pad as this couples noise onto the ADC die and into the AGND plane. The power supply lines to the ADC should use as large a trace as possible to provide low impedance paths and reduce the effects of glitches on the power supply line.

To avoid radiating noise to other sections of the board, fast switching signals, such as clocks, should be shielded with digital ground, and clock signals should never run near the analog inputs. Avoid crossover of digital and analog signals. To reduce the effects of feed through within the board, traces on opposite sides of the board should run at right angles to each other.

Good decoupling is also important. All analog supplies should be decoupled with 10 µF tantalum capacitors in parallel with 0.1 µF capacitors to GND. To achieve the best results from these decoupling components, they must be placed as close as possible to the device, ideally right up against the device. The 0.1 µF capacitors should have low effective series resistance (ESR) and effective series inductance (ESI), such as the common ceramic types or surface-mount types. These low ESR and ESI capacitors provide a low impedance path to ground at high frequencies to handle transient currents due to internal logic switching.

ADDITIONAL INFORMATION

The following publications that describe the ADSP-BF50x processors (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on our website:

- *Getting Started With Blackfin Processors*
- *ADSP-BF50x Blackfin Processor Hardware Reference* (volumes 1 and 2)
- *Blackfin Processor Programming Reference*
- *ADSP-BF50x Blackfin Processor Anomaly List*

RELATED SIGNAL CHAINS

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

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

The Application Signal Chains page in the Circuits from the Lab™ site (http:\\www.analog.com\signalchains) provides:

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

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.

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

Signal Name	Type	Function
$D_{OUT}A$, $D_{OUT}B$	O	Serial Data Outputs. The data output is supplied to each pin as a serial data stream. The bits are clocked out on the falling edge of the ADSCLK input and 14 ADSCLKs are required to access the data. The data simultaneously appears on both pins from the simultaneous conversions of both ADCs. The data stream consists of two leading zeros followed by the 12 bits of conversion data. The data is provided MSB first. If $\overline{CS}$ is held low for 16 ADSCLK cycles rather than 14, then two trailing zeros will appear after the 12 bits of data. If $\overline{CS}$ is held low for a further 16 ADSCLK cycles on either $D_{OUT}A$ or $D_{OUT}B$, the data from the other ADC follows on the D_{OUT} pin. This allows data from a simultaneous conversion on both ADCs to be gathered in serial format on either $D_{OUT}A$ or $D_{OUT}B$ using only one serial port. For more information, see the ADC—Serial Interface section.
V_{DRIVE}	P	Logic Power Supply Input. The voltage supplied at this pin determines at what voltage the digital I/O interface operates. This pin should be decoupled to DGND. The voltage at this pin may be different than that at AV_{DD} and DV_{DD} but should never exceed either by more than 0.3 V.
DV_{DD}	P	Digital Supply Voltage, 2.7 V to 5.25 V. This is the supply voltage for all digital circuitry on the internal ADC. The DV_{DD} and AV_{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 DGND.