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 Dual Core Embedded Processor

ADSP-BF606/ADSP-BF607/ADSP-BF608/ADSP-BF609

FEATURES

- Dual-core symmetric high-performance Blackfin processor, up to 500 MHz per core
  - Each core contains two 16-bit MACs, two 40-bit ALUs, and a 40-bit barrel shifter
  - RISC-like register and instruction model for ease of programming and compiler-friendly support
  - Advanced debug, trace, and performance monitoring
- Pipelined Vision Processor provides hardware to process signal and image algorithms used for pre- and co-processing of video frames in ADAS or other video processing applications
- Accepts a range of supply voltages for I/O operation. See Operating Conditions on Page 52
- Off-chip voltage regulator interface
- 349-ball BGA package (19 mm × 19 mm), RoHS compliant

MEMORY

- Each core contains 148K bytes of L1 SRAM memory (processor core-accessible) with multi-parity bit protection
- Up to 256K bytes of L2 SRAM memory with ECC protection
- Dynamic memory controller provides 16-bit interface to a single bank of DDR2 or LPDDR DRAM devices
- Static memory controller with asynchronous memory interface that supports 8-bit and 16-bit memories
- 4 Memory-to-memory DMA streams, 2 of which feature CRC protection
- Flexible booting options from flash, SD EMMC, and SPI memories and from SPI, link port and UART hosts
- Memory management unit provides memory protection

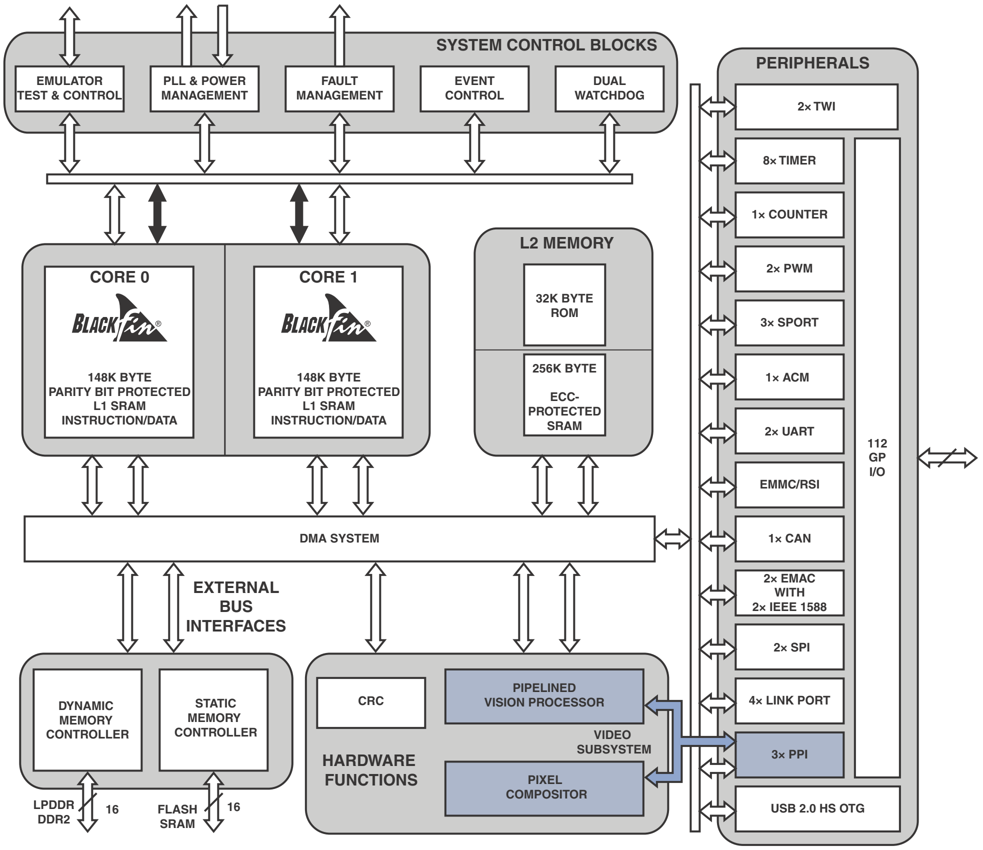


Figure 1. Processor Block Diagram

Blackfin and the Blackfin logo are registered trademarks of Analog Devices, Inc.

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

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

2/14—Rev. 0 to Rev. A

Added the system clock output specification and additional peripheral external clocks in Clock Related Operating Conditions on Page 53. These changes affect the following peripheral timing sections.

Enhanced Parallel Peripheral Interface Timing 74
Link Ports 78
Serial Ports—External Clock 80
Serial Peripheral Interface (SPI) Port—Master Timing 86
Serial Peripheral Interface (SPI) Port—Slave Timing 88
ADC Controller Module (ACM) Timing 96

Additional revisions include the following.

Corrected S0SEL and S1SEL in Figure 8 Clock Relationships and Divider Values 54

Revised the dynamic and static current tables CCLK Dynamic Current per core (mA, with ASF = 1) 57
Static Current—IDD_DEEPSLEEP (mA) 58

Corrected the t_{WARE} parameter in Asynchronous Page Mode Read 64

Corrected the timing diagram in Bus Request/Bus Grant . 69

Corrected the signal names in the following figures:
DDR2 SDRAM Clock and Control Cycle Timing 69
DDR2 SDRAM Controller Input AC Timing 70
Mobile DDR SDRAM Clock and Control Cycle Timing ... 72

Added Figure 29 and updated Table 42 in Enhanced Parallel Peripheral Interface Timing 74

Corrected the t_{HSPIDM}, t_{SDSCIM}, t_{SPICLK}, t_{HDSM}, and t_{SPITDM} specifications in Serial Peripheral Interface (SPI) Port—Master Timing 86

Corrected the t_{HDSPID} specification in Serial Peripheral Interface (SPI) Port—Slave Timing 88

Corrected $t_{SRDYSCKM1}$ in Serial Peripheral Interface (SPI) Port—SPI_RDY Timing 92

Revised all parameters in Timer Cycle Timing 94

Corrected the timing diagram in ADC Controller Module (ACM) Timing 96

Removed TWI signals in footnote 3 in JTAG Test And Emulation Port Timing 101

Added models to Automotive Products 112

GENERAL DESCRIPTION

The ADSP-BF60x 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 processors offer performance up to 500 MHz, as well as low static power consumption. Produced with a low-power and low-voltage design methodology, they provide world-class power management and performance.

By integrating a rich set of industry-leading system peripherals and memory (shown in Table 1), 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. These applications span a wide array of markets, from automotive systems to embedded industrial, instrumentation and power/motor control applications.

Table 1. Processor Comparison

Processor Feature	ADSP-BF606	ADSP-BF607	ADSP-BF608	ADSP-BF609
Up/Down/Rotary Counters	1			
Timer/Counters with PWM	8			
3-Phase PWM Units (4-pair)	2			
SPORTs	3			
SPIs	2			
USB OTG	1			
Parallel Peripheral Interface	3			
Removable Storage Interface	1			
CAN	1			
TWI	2			
UART	2			
ADC Control Module (ACM)	1			
Link Ports	4			
Ethernet MAC (IEEE 1588)	2			
Pixel Compositor (PIXC)	No		1	1
Pipelined Vision Processor (PVP) Video Resolution[1]	No		VGA	HD
Maximum PVP Line Buffer Size	N/A		640	1280
GPIOs	112			
Memory (bytes, per core)				
L1 Instruction SRAM	64K			
L1 Instruction SRAM/Cache	16K			
L1 Data SRAM	32K			
L1 Data SRAM/Cache	32K			
L1 Scratchpad	4K			
L2 Data SRAM	128K	256K		
L2 Boot ROM	32K			
Maximum Speed Grade (MHz)[2]	400	500		
Maximum SYSCLK (MHz)	250			
Package Options	349-Ball CSP_BGA			

[1] VGA is 640 × 480 pixels per frame. HD is 1280 × 960 pixels per frame.

[2] Maximum speed grade is not available with every possible SYSCLK selection.

BLACKFIN PROCESSOR CORE

As shown in Figure 1, the processor integrates two Blackfin processor cores. Each core, shown in Figure 2, 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.

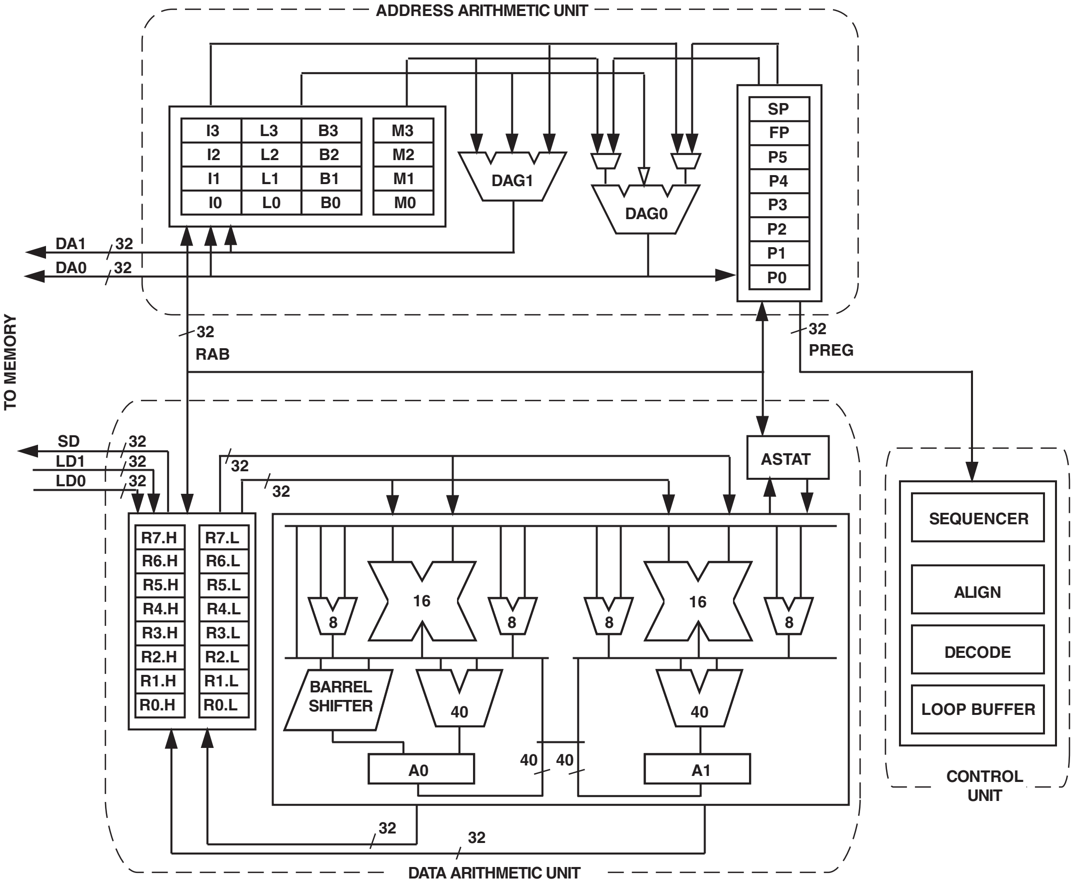


Figure 2. Blackfin Processor Core

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 supports 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).

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.

INSTRUCTION SET DESCRIPTION

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 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.
- Control of all asynchronous and synchronous events to the processor is handled by two subsystems: the Core Event Controller (CEC) and the System Event Controller (SEC).
- 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.

PROCESSOR INFRASTRUCTURE

The following sections provide information on the primary infrastructure components of the ADSP-BF609 processor.

DMA Controllers

The processor uses Direct Memory Access (DMA) to transfer data within memory spaces or between a memory space and a peripheral. The processor can specify data transfer operations and return to normal processing while the fully integrated DMA controller carries out the data transfers independent of processor activity.

DMA transfers can occur between memory and a peripheral or between one memory and another memory. Each Memory-to-memory DMA stream uses two channels, where one channel is the source channel, and the second is the destination channel.

All DMAs can transport data to and from all on-chip and off-chip memories. Programs can use two types of DMA transfers, descriptor-based or register-based. Register-based DMA allows the processor to directly program DMA control registers to initiate a DMA transfer. On completion, the control registers may be automatically updated with their original setup values for continuous transfer. Descriptor-based DMA transfers require a set of parameters stored within memory to initiate a DMA sequence. Descriptor-based DMA transfers allow multiple DMA sequences to be chained together and a DMA channel can be programmed to automatically set up and start another DMA transfer after the current sequence completes.

The DMA controller supports the following DMA operations.

- A single linear buffer that stops on completion.
- A linear buffer with negative, positive or zero stride length.
- A circular, auto-refreshing buffer that interrupts when each buffer becomes full.
- A similar buffer that interrupts on fractional buffers (for example, 1/2, 1/4).
- 1D DMA – uses a set of identical ping-pong buffers defined by a linked ring of two-word descriptor sets, each containing a link pointer and an address.
- 1D DMA – uses a linked list of 4 word descriptor sets containing a link pointer, an address, a length, and a configuration.
- 2D DMA – uses an array of one-word descriptor sets, specifying only the base DMA address.
- 2D DMA – uses a linked list of multi-word descriptor sets, specifying everything.

CRC Protection

The two CRC protection modules allow system software to periodically calculate the signature of code and/or data in memory, the content of memory-mapped registers, or communication message objects. Dedicated hardware circuitry compares the signature with pre calculated values and triggers appropriate fault events.

For example, every 100 ms the system software might initiate the signature calculation of the entire memory contents and compare these contents with expected, pre calculated values. If a mismatch occurs, a fault condition can be generated (via the processor core or the trigger routing unit).

The CRC is a hardware module based on a CRC32 engine that computes the CRC value of the 32-bit data words presented to it. Data is provided by the source channel of the memory-to-memory DMA (in memory scan mode) and is optionally forwarded to the destination channel (memory transfer mode). The main features of the CRC peripheral are:

- Memory scan mode
- Memory transfer mode
- Data verify mode
- Data fill mode
- User-programmable CRC32 polynomial
- Bit/byte mirroring option (endianness)
- Fault/error interrupt mechanisms
- 1D and 2D fill block to initialize array with constants.
- 32-bit CRC signature of a block of a memory or MMR block.

Event Handling

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 processor 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 $\overline{\text{NMI}}$ 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.

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. For more information, see the *ADSP-BF60x Processor Programmer's Reference*.

System Event Controller (SEC)

The SEC manages the enabling, prioritization, and routing of events from each system interrupt or fault source. Additionally, it provides notification and identification of the highest priority active system interrupt request to each core and routes system fault sources to its integrated fault management unit.

Trigger Routing Unit (TRU)

The TRU provides system-level sequence control without core intervention. The TRU maps trigger masters (generators of triggers) to trigger slaves (receivers of triggers). Slave endpoints can be configured to respond to triggers in various ways. Common applications enabled by the TRU include:

- Automatically triggering the start of a DMA sequence after a sequence from another DMA channel completes
- Software triggering
- Synchronization of concurrent activities

Pin Interrupts

Every port pin on the processor can request interrupts in either an edge-sensitive or a level-sensitive manner with programmable polarity. Interrupt functionality is decoupled from GPIO operation. Six system-level interrupt channels (PINT0–5) are reserved for this purpose. Each of these interrupt channels can manage up to 32 interrupt pins. The assignment from pin to interrupt is not performed on a pin-by-pin basis. Rather, groups of eight pins (half ports) can be flexibly assigned to interrupt channels.

Every pin interrupt channel features a special set of 32-bit memory-mapped registers that enable half-port assignment and interrupt management. This includes masking, identification, and clearing of requests. These registers also enable access to the respective pin states and use of the interrupt latches, regardless of whether the interrupt is masked or not. Most control registers feature multiple MMR address entries to write-one-to-set or write-one-to-clear them individually.

General-Purpose I/O (GPIO)

Each general-purpose port pin can be individually controlled by manipulation of the port control, status, and interrupt registers:

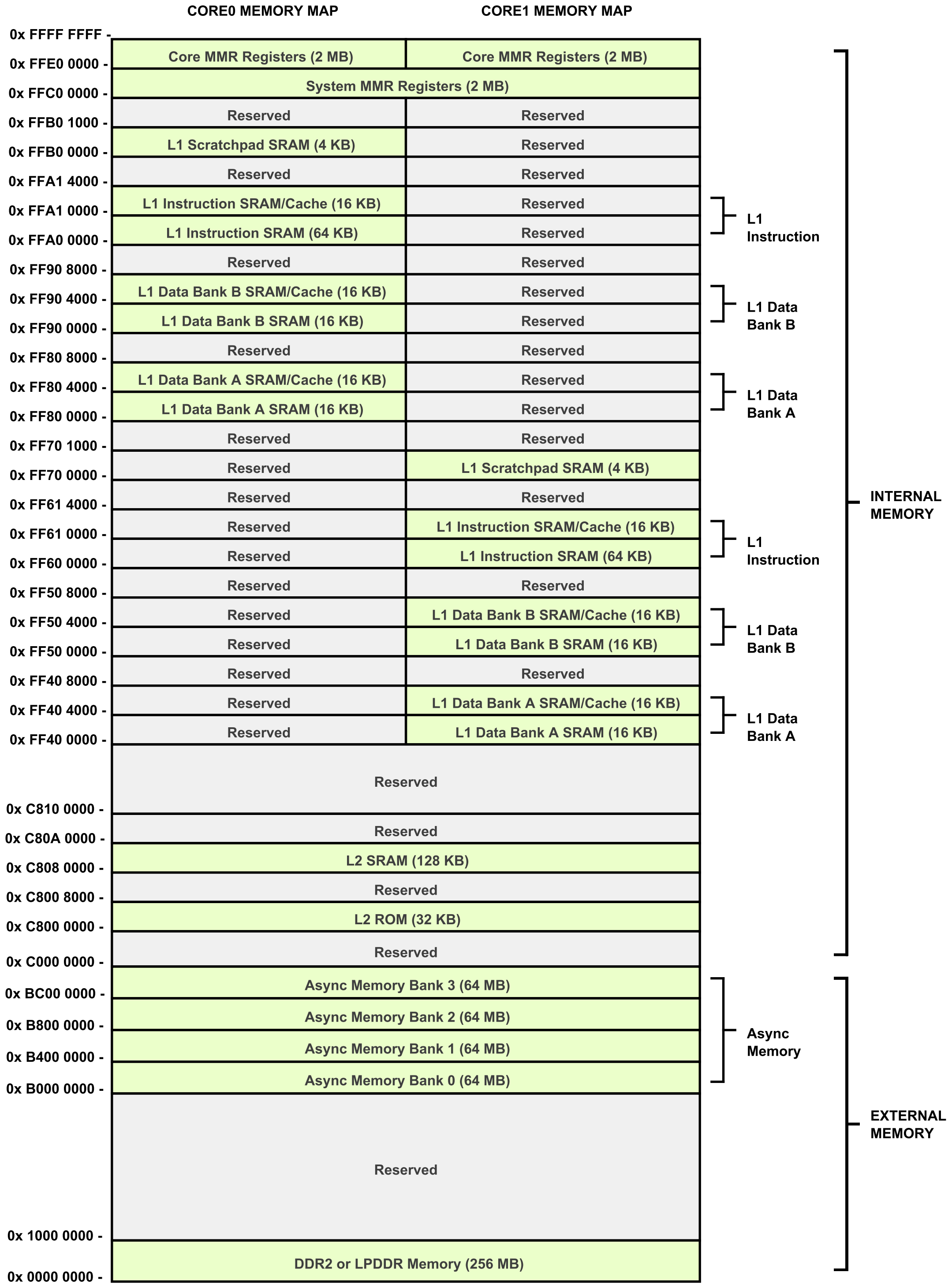
- GPIO direction control register – Specifies the direction of each individual GPIO pin as input or output.
- GPIO control and status registers – A "write one to modify" mechanism allows any combination of individual GPIO pins to be modified in a single instruction, without affecting the level of any other GPIO pins.
- GPIO interrupt mask registers – Allow each individual GPIO pin to function as an interrupt to the processor. 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 – 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.

Pin Multiplexing

The processor supports a flexible multiplexing scheme that multiplexes the GPIO pins with various peripherals. A maximum of 4 peripherals plus GPIO functionality is shared by each GPIO pin. All GPIO pins have a bypass path feature – that is, when the output enable and the input enable of a GPIO pin are both active, the data signal before the pad driver is looped back to the receive path for the same GPIO pin. For more information, see GP I/O Multiplexing for 349-Ball CSP_BGA on Page 33.

MEMORY ARCHITECTURE

The 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 larger, lower-cost and performance interface-accessible memory systems. See Figure 3 and Figure 4.



	CORE0 MEMORY MAP	CORE1 MEMORY MAP
0x FFFF FFFF -		
0x FFE0 0000 -	Core MMR Registers (2 MB)	Core MMR Registers (2 MB)
0x FFC0 0000 -	System MMR Registers (2 MB)	
0x FFB0 1000 -	Reserved	Reserved
0x FFB0 0000 -	L1 Scratchpad SRAM (4 KB)	Reserved
0x FFA1 4000 -	Reserved	Reserved
0x FFA1 0000 -	L1 Instruction SRAM/Cache (16 KB)	Reserved
0x FFA0 0000 -	L1 Instruction SRAM (64 KB)	Reserved
0x FF90 8000 -	Reserved	Reserved
0x FF90 4000 -	L1 Data Bank B SRAM/Cache (16 KB)	Reserved
0x FF90 0000 -	L1 Data Bank B SRAM (16 KB)	Reserved
0x FF80 8000 -	Reserved	Reserved
0x FF80 4000 -	L1 Data Bank A SRAM/Cache (16 KB)	Reserved
0x FF80 0000 -	L1 Data Bank A SRAM (16 KB)	Reserved
0x FF70 1000 -	Reserved	Reserved
0x FF70 0000 -	Reserved	L1 Scratchpad SRAM (4 KB)
0x FF61 4000 -	Reserved	Reserved
0x FF61 0000 -	Reserved	L1 Instruction SRAM/Cache (16 KB)
0x FF60 0000 -	Reserved	L1 Instruction SRAM (64 KB)
0x FF50 8000 -	Reserved	Reserved
0x FF50 4000 -	Reserved	L1 Data Bank B SRAM/Cache (16 KB)
0x FF50 0000 -	Reserved	L1 Data Bank B SRAM (16 KB)
0x FF40 8000 -	Reserved	Reserved
0x FF40 4000 -	Reserved	L1 Data Bank A SRAM/Cache (16 KB)
0x FF40 0000 -	Reserved	L1 Data Bank A SRAM (16 KB)
	Reserved	
0x C810 0000 -	Reserved	
0x C80A 0000 -	L2 SRAM (128 KB)	
0x C808 0000 -	Reserved	
0x C800 8000 -	L2 ROM (32 KB)	
0x C800 0000 -	Reserved	
0x C000 0000 -	Async Memory Bank 3 (64 MB)	
0x BC00 0000 -	Async Memory Bank 2 (64 MB)	
0x B800 0000 -	Async Memory Bank 1 (64 MB)	
0x B400 0000 -	Async Memory Bank 0 (64 MB)	
0x B000 0000 -	Reserved	
0x 1000 0000 -	DDR2 or LPDDR Memory (256 MB)	
0x 0000 0000 -		

Figure 3. ADSP-BF606 Internal/External Memory Map

	CORE0 MEMORY MAP	CORE1 MEMORY MAP		
0x FFFF FFFF -				
0x FFE0 0000 -	Core MMR Registers (2 MB)	Core MMR Registers (2 MB)		INTERNAL MEMORY
0x FFC0 0000 -	System MMR Registers (2 MB)			
0x FFB0 1000 -	Reserved	Reserved		
0x FFB0 0000 -	L1 Scratchpad SRAM (4 KB)	Reserved		
0x FFA1 4000 -	Reserved	Reserved		
0x FFA1 0000 -	L1 Instruction SRAM/Cache (16 KB)	Reserved	L1 Instruction	
0x FFA0 0000 -	L1 Instruction SRAM (64 KB)	Reserved		
0x FF90 8000 -	Reserved	Reserved		
0x FF90 4000 -	L1 Data Bank B SRAM/Cache (16 KB)	Reserved	L1 Data Bank B	
0x FF90 0000 -	L1 Data Bank B SRAM (16 KB)	Reserved		
0x FF80 8000 -	Reserved	Reserved		
0x FF80 4000 -	L1 Data Bank A SRAM/Cache (16 KB)	Reserved	L1 Data Bank A	
0x FF80 0000 -	L1 Data Bank A SRAM (16 KB)	Reserved		
0x FF70 1000 -	Reserved	Reserved		
0x FF70 0000 -	Reserved	L1 Scratchpad SRAM (4 KB)		
0x FF61 4000 -	Reserved	Reserved		
0x FF61 0000 -	Reserved	L1 Instruction SRAM/Cache (16 KB)	L1 Instruction	
0x FF60 0000 -	Reserved	L1 Instruction SRAM (64 KB)		
0x FF50 8000 -	Reserved	Reserved		
0x FF50 4000 -	Reserved	L1 Data Bank B SRAM/Cache (16 KB)	L1 Data Bank B	
0x FF50 0000 -	Reserved	L1 Data Bank B SRAM (16 KB)		
0x FF40 8000 -	Reserved	Reserved		
0x FF40 4000 -	Reserved	L1 Data Bank A SRAM/Cache (16 KB)	L1 Data Bank A	
0x FF40 0000 -	Reserved	L1 Data Bank A SRAM (16 KB)		
0x C810 0000 -	Reserved			
0x C80C 0000 -	Reserved			
0x C808 0000 -	L2 SRAM (256 KB)			
0x C800 8000 -	Reserved			
0x C800 0000 -	L2 ROM (32 KB)			
0x C000 0000 -	Reserved			
0x BC00 0000 -	Async Memory Bank 3 (64 MB)		Async Memory	EXTERNAL MEMORY
0x B800 0000 -	Async Memory Bank 2 (64 MB)			
0x B400 0000 -	Async Memory Bank 1 (64 MB)			
0x B000 0000 -	Async Memory Bank 0 (64 MB)			
0x 1000 0000 -	Reserved			
0x 0000 0000 -	DDR2 or LPDDR Memory (256 MB)			

Figure 4. ADSP-BF607/ADSP-BF608/ADSP-BF609 Internal/External Memory Map

Internal (Core-Accessible) Memory

The L1 memory system is the highest-performance memory available to the Blackfin processor cores.

Each core has its own private L1 memory. The modified Harvard architecture supports two concurrent 32-bit data accesses along with an instruction fetch at full processor speed which provides high bandwidth processor performance. In each core a 64K-byte block of data memory partners with an 80K-byte memory block for instruction storage. Each data block is multibanked for efficient data exchange through DMA and can be configured as SRAM. Alternatively, 16K bytes of each block can be configured in L1 cache mode. The four-way set-associative instruction cache and the 2 two-way set-associative data caches greatly accelerate memory access performance, especially when accessing external memories.

The L1 memory domain also features a 4K-byte scratchpad SRAM block which is ideal for storing local variables and the software stack. All L1 memory is protected by a multi-parity bit concept, regardless of whether the memory is operating in SRAM or cache mode.

Outside of the L1 domain, L2 and L3 memories are arranged using a Von Neumann topology. The L2 memory domain is a unified instruction and data memory and can hold any mixture of code and data required by the system design. The L2 memory domain is accessible by both Blackfin cores through a dedicated 64-bit interface. It operates at SYSCLK frequency.

The processor features up to 256K bytes of L2 SRAM which is ECC-protected and organized in eight banks. Individual banks can be made private to any of the cores or the DMA subsystem. There is also a 32K-byte single-bank ROM in the L2 domain. It contains boot code and safety functions.

Static Memory Controller (SMC)

The SMC can be programmed to control up to four banks of external memories or memory-mapped devices, with very flexible timing parameters. Each bank occupies a 64M byte segment regardless of the size of the device used, so that these banks are only contiguous if each is fully populated with 64M bytes of memory.

Dynamic Memory Controller (DMC)

The DMC includes a controller that supports JESD79-2E compatible double data rate (DDR2) SDRAM and JESD209A low power DDR (LPDDR) SDRAM devices.

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 which contains the control MMRs for all core functions, and the other which contains the registers needed for setup and control of the on-chip peripherals outside of the core. The MMRs are accessible only in supervisor mode and appear as reserved space to on-chip peripherals.

Booting

The processor has several mechanisms for automatically loading internal and external memory after a reset. The boot mode is defined by the SYS_BMODE input pins dedicated for 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.

The boot modes are shown in Table 2. These modes are implemented by the SYS_BMODE bits of the reset configuration register and are sampled during power-on resets and software-initiated resets.

Table 2. Boot Modes

SYS_BMODE Setting	Boot Mode
000	No boot/Idle
001	Memory
010	RSI0 Master
011	SPI0 Master
100	SPI0 Slave
101	Reserved
110	LP0 Slave
111	UART0 Slave

VIDEO SUBSYSTEM

The following sections describe the components of the processor's video subsystem. These blocks are shown with blue shading in Figure 1 on Page 1.

Video Interconnect (VID)

The Video Interconnect provides a connectivity matrix that interconnects the Video Subsystem: three PPIs, the PIXC, and the PVP. The interconnect uses a protocol to manage data transfer among these video peripherals.

Pipelined Vision Processor (PVP)

The PVP engine provides hardware implementation of signal and image processing algorithms that are required for co-processing and pre-processing of monochrome video frames in ADAS applications, robotic systems, and other machine applications.

The PVP works in conjunction with the Blackfin cores. It is optimized for convolution and wavelet based object detection and classification, and tracking and verification algorithms. The PVP has the following processing blocks.

- Four 5 × 5 16-bit convolution blocks optionally followed by down scaling
- A 16-bit cartesian-to-polar coordinate conversion block
- A pixel edge classifier that supports 1st and 2nd derivative modes
- An arithmetic unit with 32-bit addition, multiply and divide

- A 32-bit threshold block with 16 thresholds, a histogram, and run-length encoding
- Two 32-bit integral blocks that support regular and diagonal integrals
- An up- and down-scaling unit with independent scaling ratios for horizontal and vertical components
- Input and output formatters for compatibility with many data formats, including Bayer input format

The PVP can form a pipe of all the constituent algorithmic modules and is dynamically reconfigurable to form different pipeline structures.

The PVP supports the simultaneous processing of up to four data streams. The memory pipe stream operates on data received by DMA from any L1, L2, or L3 memory. The three camera pipe streams operate on a common input received directly from any of the three PPI inputs. Optionally, the PIXC can convert color data received by the PPI and forward luma values to the PVP's monochrome engine. Each stream has a dedicated DMA output. This preprocessing concept ensures careful use of available power and bandwidth budgets and frees up the processor cores for other tasks.

The PVP provides for direct core MMR access to all control/status registers. Two hardware interrupts interface to the system event controller. For optimal performance, the PVP allows register programming through its control DMA interface, as well as outputting selected status registers through the status DMA interface. This mechanism enables the PVP to automatically process job lists completely independent of the Blackfin cores.

Pixel Compositor (PIXC)

The pixel compositor (PIXC) provides image overlays with transparent-color support, alpha blending, and color space conversion capabilities for output to TFT LCDs and NTSC/PAL video encoders. It provides all of the control to allow two data streams from two separate data buffers to be combined, blended, and converted into appropriate forms for both LCD panels and digital video outputs. The main image buffer provides the basic background image, which is presented in the data stream. The overlay image buffer allows the user to add multiple foreground text, graphics, or video objects on top of the main image or video data stream.

Parallel Peripheral Interface (PPI)

The processor provides up to three parallel peripheral interfaces (PPIs), supporting data widths up to 24 bits. The PPI supports direct connection to TFT LCD panels, parallel analog-to-digital and digital-to-analog converters, video encoders and decoders, image sensor modules and other general-purpose peripherals.

The following features are supported in the PPI module:

- Programmable data length: 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, 18 bits, and 24 bits per clock.
- Various framed, non-framed, and general-purpose operating modes. Frame syncs can be generated internally or can be supplied by an external device.
- ITU-656 status word error detection and correction for ITU-656 receive modes and ITU-656 preamble and status word decode.
- Optional packing and unpacking of data to/from 32 bits from/to 8 bits, 16 bits and 24 bits. If packing/unpacking is enabled, endianness can be configured to change the order of packing/unpacking of bytes/words.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various de-interleaving/interleaving modes for receiving/transmitting 4:2:2 YCrCb data.
- Configurable LCD data enable (DEN) output available on Frame Sync 3.

PROCESSOR SAFETY FEATURES

The ADSP-BF60x processor has been designed for functional safety applications. While the level of safety is mainly dominated by the system concept, the following primitives are provided by the devices to build a robust safety concept.

Dual Core Supervision

The processor has been implemented as dual-core devices to separate critical tasks to large independency. Software models support mutual supervision of the cores in symmetrical fashion.

Multi-Parity-Bit-Protected L1 Memories

In the processor's L1 memory space, whether SRAM or cache, each word is protected by multiple parity bits to detect the single event upsets that occur in all RAMs. This applies both to L1 instruction and data memory spaces.

ECC-Protected L2 Memories

Error correcting codes (ECC) are used to correct single event upsets. The L2 memory is protected with a Single Error Correct-Double Error Detect (SEC-DED) code. By default ECC is enabled, but it can be disabled on a per-bank basis. Single-bit errors are transparently corrected. Dual-bit errors can issue a system event or fault if enabled. ECC protection is fully transparent to the user, even if L2 memory is read or written by 8-bit or 16-bit entities.

CRC-Protected Memories

While parity bit and ECC protection mainly protect against random soft errors in L1 and L2 memory cells, the CRC engines can be used to protect against systematic errors (pointer errors) and static content (instruction code) of L1, L2 and even L3 memories (DDR2, LPDDR). The processors feature two CRC engines which are embedded in the memory-to-memory DMA controllers. CRC check sums can be calculated or compared on the fly during memory transfers, or one or multiple memory regions can be continuously scrubbed by single DMA work unit as per DMA descriptor chain instructions. The CRC engine also protects data loaded during the boot process.

Memory Protection

The Blackfin cores feature a memory protection concept, which grants data and/or instruction accesses from enabled memory regions only. A supervisor mode vs. user mode programming model supports dynamically varying access rights. Increased flexibility in memory page size options supports a simple method of static memory partitioning.

System Protection

All system resources and L2 memory banks can be controlled by either the processor cores, memory-to-memory DMA, or the system debug unit (SDU). A system protection unit (SPU) enables write accesses to specific resources that are locked to any of four masters: Core 0, Core 1, Memory DMA, and the System Debug Unit. System protection is enabled in greater granularity for some modules (L2, SEC and GPIO controllers) through a *global lock* concept.

Watchpoint Protection

The primary purpose of watchpoints and hardware breakpoints is to serve emulator needs. When enabled, they signal an emulator event whenever user-defined system resources are accessed or a core executes from user-defined addresses. Watchpoint events can be configured such that they signal the events to the other Blackfin core or to the fault management unit.

Dual Watchdog

The two on-chip watchdog timers each may supervise one Blackfin core.

Bandwidth Monitor

All DMA channels that operate in memory-to-memory mode (Memory DMA, PVP Memory Pipe DMA, PIXC DMA) are equipped with a bandwidth monitor mechanism. They can signal a system event or fault when transactions tend to starve because system buses are fully loaded with higher-priority traffic.

Signal Watchdogs

The eight general-purpose timers feature two new modes to monitor off-chip signals. The Watchdog Period mode monitors whether external signals toggle with a period within an expected range. The Watchdog Width mode monitors whether the pulse widths of external signals are in an expected range. Both modes help to detect incorrect undesired toggling (or lack thereof) of system-level signals.

Up/Down Count Mismatch Detection

The up/down counter can monitor external signal pairs, such as request/grant strobes. If the edge count mismatch exceeds the expected range, the up/down counter can flag this to the processor or to the fault management unit.

Fault Management

The fault management unit is part of the system event controller (SEC). Any system event, whether a dual-bit uncorrectable ECC error, or any peripheral status interrupt, can be defined as being a “fault”. Additionally, the system events can be defined as an interrupt to the cores. If defined as such, the SEC forwards the event to the fault management unit which may automatically reset the entire device for reboot, or simply toggle the SYS_FAULT output pins to signal off-chip hardware. Optionally, the fault management unit can delay the action taken via a keyed sequence, to provide a final chance for the Blackfin cores to resolve the crisis and to prevent the fault action from being taken.

ADDITIONAL PROCESSOR PERIPHERALS

The processor contains 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). The 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 following sections describe additional peripherals that were not described in the previous sections.

Timers

The processor includes several timers which are described in the following sections.

General-Purpose Timers

There is one GP timer unit and it provides eight general-purpose programmable timers. Each timer has an external pin that can be configured either as a pulse width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input on the TMRx pins, an external clock TMRCLK input pin, or to the internal SCLK0.

The timer units can be used in conjunction with the UARTs and the CAN controller 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 to either the system clock or to external signals. Timer events can also trigger other peripherals via the TRU (for instance, to signal a fault).

Core Timers

Each processor core also has its own dedicated timer. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generating periodic operating system interrupts.

Watchdog Timers

Each core includes a 32-bit timer, which may be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, via generation of a hardware 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.

After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the timer control register, which is set only upon a watchdog generated reset.

3-Phase PWM Units

The Pulse Width Modulator (PWM) module is a flexible and programmable waveform generator. With minimal CPU intervention the PWM peripheral is capable of generating complex waveforms for motor control, Pulse Coded Modulation (PCM), Digital to Analog Conversion (DAC), power switching and power conversion. The PWM module has 4 PWM pairs capable of 3-phase PWM generation for source inverters for AC induction and DC brush less motors.

The two 3-phase PWM generation units each feature:

- 16-bit center-based PWM generation unit
- Programmable PWM pulse width
- Single update mode with option for asymmetric duty
- Programmable dead time and switching frequency
- Twos-complement implementation which permits smooth transition to full ON and full OFF states
- Dedicated asynchronous PWM shutdown signal

Link Ports

Four DMA-enabled, 8-bit-wide link ports can connect to the link ports of other DSPs or processors. Link ports are bidirectional ports having eight data lines, an acknowledge line and a clock line.

Serial Ports (SPORTs)

Three synchronous serial ports that provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices such as Analog Devices' AD183x family of audio codecs, ADCs, and DACs. The serial ports are made up of two data lines, a clock, and frame sync. The data lines can be programmed to either transmit or receive and each data line has a dedicated DMA channel.

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

Serial ports operate in five modes:

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

ACM Interface

The ADC control module (ACM) provides an interface that synchronizes the controls between the processor and an analog-to-digital converter (ADC). 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.

Figure 5 shows how to connect an external ADC to the ACM and one of the SPORTs.

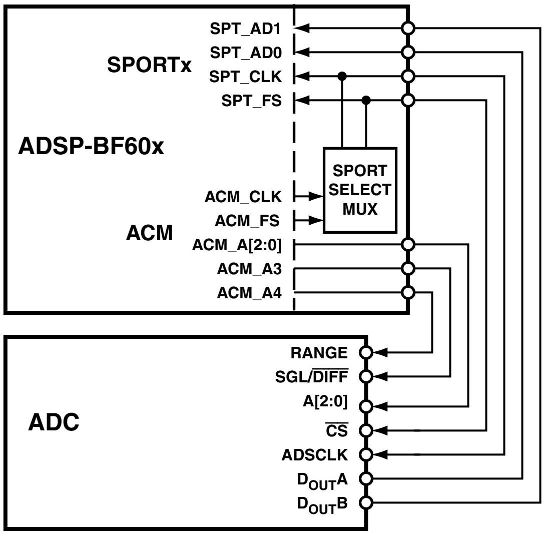


Figure 5. ADC, ACM, and SPORT Connections

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 a peripheral such as a SPORT or a SPI.

The processor interfaces directly to many ADCs without any glue logic required.

General-Purpose Counters

A 32-bit counter is provided that can operate in general-purpose up/down count modes and can sense 2-bit quadrature or binary codes as typically emitted by industrial drives or manual thumbwheels. 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 general-purpose timer 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.

Serial Peripheral Interface (SPI) Ports

The processors have two SPI-compatible ports that allow the processor to communicate with multiple SPI-compatible devices.

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

In a multi-master or multi-slave SPI system, the MOSI and MISO data output pins can be configured to behave as open drain outputs (using the ODM bit) to prevent contention and possible damage to pin drivers. An external pull-up resistor is required on both the MOSI and MISO pins when this option is selected.

When ODM is set and the SPI is configured as a master, the MOSI pin is three-stated when the data driven out on MOSI is a logic-high. The MOSI pin is not three-stated when the driven data is a logic-low. Similarly, when ODM is set and the SPI is configured as a slave, the MISO pin is three-stated if the data driven out on MISO is a logic-high.

The SPI port's baud rate and clock phase/polarities are programmable, and it has integrated DMA channels for both transmit and receive data streams.

UART Ports

The processors provide two full-duplex universal asynchronous receiver/transmitter (UART) ports, which are fully compatible with PC-standard UARTs. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for five to eight data bits, and none, even, or odd parity. Optionally, an additional address bit can be transferred to interrupt only addressed nodes in multi-drop bus (MDB) systems. A frame is terminates by one, one and a half, two or two and a half stop bits.

The UART ports support automatic hardware flow control through the Clear To Send (CTS) input and Request To Send (RTS) output with programmable assertion FIFO levels.

To help support the Local Interconnect Network (LIN) protocols, a special command causes the transmitter to queue a break command of programmable bit length into the transmit buffer. Similarly, the number of stop bits can be extended by a programmable inter-frame space.

The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA®) serial infrared physical layer link specification (SIR) protocol.

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 module is compatible with the widely used I^{2}C bus standard. The TWI module offers the capabilities of simultaneous master and slave operation and support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (TWI_SCL) and data (TWI_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.

Removable Storage Interface (RSI)

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). The following list describes the main features of the RSI controller.

- Support for a single MMC, SD memory, SDIO card
- Support for 1-bit and 4-bit SD modes
- Support for 1-bit, 4-bit, and 8-bit MMC modes
- Support for eMMC 4.3 embedded NAND flash devices
- A ten-signal external interface with clock, command, and up to eight data lines
- Card interface clock generation from SCLK0
- SDIO interrupt and read wait features

Controller Area Network (CAN)

A CAN controller implements the CAN 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well suited for control applications due to its capability to communicate reliably over a network. This is because the protocol incorporates CRC checking, message error tracking, and fault node confinement.

The CAN controller offers the following features:

- 32 mailboxes (8 receive only, 8 transmit only, 16 configurable for receive or transmit).
- Dedicated acceptance masks for each mailbox.
- Additional data filtering on first two bytes.
- Support for both the standard (11-bit) and extended (29-bit) identifier (ID) message formats.

- Support for remote frames.
- Active or passive network support.
- CAN wakeup from hibernation mode (lowest static power consumption mode).
- Interrupts, including: TX complete, RX complete, error and global.

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

10/100 Ethernet MAC

The processor can directly connect to a network by way of an embedded fast Ethernet media access controller (MAC) that supports both 10-BaseT (10M bits/sec) and 100-BaseT (100M bits/sec) operation. The 10/100 Ethernet MAC peripheral on the processor is fully compliant to the IEEE 802.3-2002 standard and it provides programmable features designed to minimize supervision, bus use, or message processing by the rest of the processor system.

Some standard features are:

- Support and RMII protocols for external PHYs
- Full duplex and half duplex modes
- Media access management (in half-duplex operation)
- Flow control
- Station management: generation of MDC/MDIO frames for read-write access to PHY registers

Some advanced features are:

- Automatic checksum computation of IP header and IP payload fields of RX frames
- Independent 32-bit descriptor-driven receive and transmit DMA channels
- Frame status delivery to memory through DMA, including frame completion semaphores for efficient buffer queue management in software
- TX DMA support for separate descriptors for MAC header and payload to eliminate buffer copy operations
- Convenient frame alignment modes
- 47 MAC management statistics counters with selectable clear-on-read behavior and programmable interrupts on half maximum value
- Advanced power management
- Magic packet detection and wakeup frame filtering
- Support for 802.3Q tagged VLAN frames
- Programmable MDC clock rate and preamble suppression

IEEE 1588 Support

The IEEE 1588 standard is a precision clock synchronization protocol for networked measurement and control systems. The processor includes hardware support for IEEE 1588 with an integrated precision time protocol synchronization engine (PTP_TSYNC). This engine provides hardware assisted time stamping to improve the accuracy of clock synchronization between PTP nodes. The main features of the engine are:

- Support for both IEEE 1588-2002 and IEEE 1588-2008 protocol standards
- Hardware assisted time stamping capable of up to 12.5 ns resolution
- Lock adjustment
- Automatic detection of IPv4 and IPv6 packets, as well as PTP messages
- Multiple input clock sources (SCLK0, RMII clock, external clock)
- Programmable pulse per second (PPS) output
- Auxiliary snapshot to time stamp external events

USB 2.0 On-the-Go Dual-Role Device Controller

The USB 2.0 OTG dual-role device controller provides a low-cost connectivity solution for the growing adoption of this bus standard in industrial applications, as well as consumer mobile devices such as cell phones, digital still cameras, and MP3 players. The USB 2.0 controller allows these devices to transfer data using a point-to-point USB connection without the need for a PC host. The module can operate in a traditional USB peripheral-only mode as well as the host mode presented in the On-the-Go (OTG) supplement to the USB 2.0 specification.

The USB clock (USB_CLKIN) is provided through a dedicated external crystal or crystal oscillator.

The USB On-the-Go dual-role device controller includes a Phase Locked Loop with programmable multipliers to generate the necessary internal clocking frequency for USB.

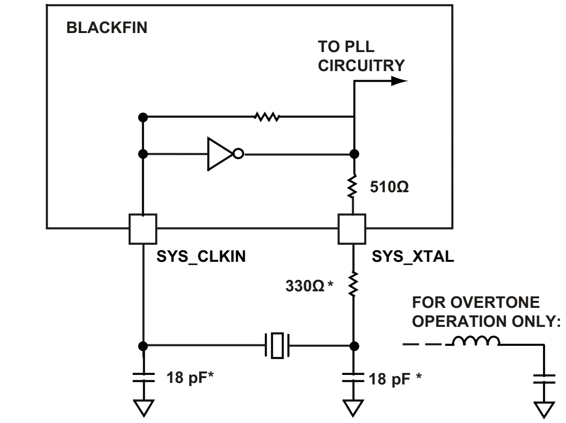
POWER AND CLOCK MANAGEMENT

The processor provides four operating modes, each with a different performance/power profile. When configured for a 0 V internal supply voltage (V_{DD_INT}), the processor enters the hibernate state. Control of clocking to each of the processor peripherals also reduces power consumption. See Table 5 for a summary of the power settings for each mode.

Crystal Oscillator (SYS_XTAL)

The processor can be clocked by an external crystal (Figure 6), 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 SYS_CLKIN pin. When an external clock is used, the SYS_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 6. A parallel-resonant, fundamental frequency, microprocessor grade crystal is connected across the SYS_CLKIN and XTAL pins. The on-chip resistance between SYS_CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended.



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 18pF SHOULD BE TREATED AS A MAXIMUM, AND THE SUGGESTED RESISTOR VALUE SHOULD BE REDUCED TO 0 Ω.

Figure 6. External Crystal Connection

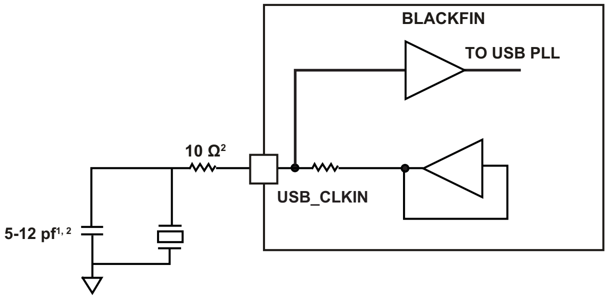
The two capacitors and the series resistor shown in Figure 6 fine tune phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 6 are typical values only. The capacitor values are dependent upon the crystal manufacturers' load capacitance recommendations and the PCB 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 6. A design procedure for third-overtone operation is discussed in detail in application note (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."

USB Crystal Oscillator

The USB 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 USB_CLKIN pin. 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 7. A parallel-resonant, fundamental frequency, microprocessor grade crystal is connected between the USB_CLKIN pin and ground. A load capacitor is placed in parallel with the crystal. The combined capacitive value of the board trace parasitic, the case capacitance of the crystal (from crystal manufacturer) and the parallel capacitor in the diagram should be in the range of 8 pF to 15 pF.



NOTES:
1. CAPACITANCE VALUE SHOWN INCLUDES BOARD PARASITICS
2. VALUES ARE A PRELIMINARY ESTIMATE.

Figure 7. External USB Crystal Connection

The crystal should be chosen so that its rated load capacitance matches the nominal total capacitance on this node. A series resistor may be added between the USB_CLKIN pin and the parallel crystal and capacitor combination, in order to further reduce the drive level of the crystal.

The parallel capacitor and the series resistor shown in Figure 7 fine tune phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 7 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.

Clock Generation

The clock generation unit (CGU) generates all on-chip clocks and synchronization signals. Multiplication factors are programmed to the PLL to define the PLLCLK frequency. Programmable values divide the PLLCLK frequency to generate the core clock (CCLK), the system clocks (SYSCLK, SCLK0 and SCLK1), the LPDDR or DDR2 clock (DCLK) and the output clock (OCLK). This is illustrated in Figure 8 on Page 54.

Writing to the CGU control registers does not affect the behavior of the PLL immediately. Registers are first programmed with a new value, and the PLL logic executes the changes so that it transitions smoothly from the current conditions to the new ones.

SYS_CLKIN oscillations start when power is applied to the V_{DD_EXT} pins. The rising edge of $\overline{\text{SYS\_HWRST}}$ can be applied after all voltage supplies are within specifications (see Operating Conditions on Page 52), and SYS_CLKIN oscillations are stable.

Clock Out/External Clock

The SYS_CLKOUT output pin has programmable options to output divided-down versions of the on-chip clocks. By default, the SYS_CLKOUT pin drives a buffered version of the SYS_CLKIN input. Clock generation faults (for example PLL unlock) may trigger a reset by hardware. The clocks shown in Table 3 can be outputs from SYS_CLKOUT.

Table 3. Clock Dividers

Clock Source	Divider
CCLK (core clock)	By 4
SYSCLK (System clock)	By 2
SCLK0 (system clock for PVP, all peripherals not covered by SCLK1)	None
SCLK1 (system clock for SPORTS, SPI, ACM)	None
DCLK (LPDDR/DDR2 clock)	By 2
OCLK (output clock)	Programmable
CLKBUF	None, direct from SYS_CLKIN

Power Management

As shown in Table 4, the processor supports five different power domains, which maximizes flexibility while maintaining compliance with industry standards and conventions. 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 4. Power Domains

Power Domain	V_{DD} Range
All internal logic	V_{DD_INT}
DDR2/LPDDR	V_{DD_DMC}
USB	V_{DD_USB}
Thermal diode	V_{DD_TD}
All other I/O (includes SYS, JTAG, and Ports pins)	V_{DD_EXT}

The dynamic power management feature of the processor allows the processor's core 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.

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 cores 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 clocks and system clocks run at the input clock (SYS_CLKIN) frequency. DMA access is available to appropriately configured L1 memories.

For more information about PLL controls, see the "Dynamic Power Management" chapter in the *ADSP-BF60x Blackfin Processor Hardware Reference.*

See Table 5 for a summary of the power settings for each mode.

Table 5. Power Settings

Mode/State	PLL	PLL Bypassed	f_{CCLK}	f_{SYSCLK}, f_{DCLK}, f_{SCLK0}, f_{SCLK1}	Core Power
Full On	Enabled	No	Enabled	Enabled	On
Active	Enabled/ Disabled	Yes	Enabled	Enabled	On
Deep Sleep	Disabled	—	Disabled	Disabled	On
Hibernate	Disabled	—	Disabled	Disabled	Off

Deep Sleep Operating Mode—Maximum Dynamic Power Savings

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core and to all synchronous peripherals. Asynchronous peripherals may still be running but cannot access internal resources or external memory.

Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor cores and to all of the peripherals. This setting signals the external voltage regulator supplying the V_{DD_INT} pins to shut off using the SYS_ EXTWAKE signal, which provides 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.

Since the V_{DD_EXT} pins 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.

Reset Control Unit

Reset is the initial state of the whole processor or one of the cores and is the result of a hardware or software triggered event. In this state, all control registers are set to their default values and functional units are idle. Exiting a full system reset starts with Core-0 only being ready to boot. Exiting a Core-n only reset starts with this Core-n being ready to boot.

The Reset Control Unit (RCU) controls how all the functional units enter and exit reset. Differences in functional requirements and clocking constraints define how reset signals are generated. Programs must guarantee that none of the reset functions puts the system into an undefined state or causes resources to stall. This is particularly important when only one of the cores is reset (programs must ensure that there is no pending system activity involving the core that is being reset).

From a system perspective reset is defined by both the reset target and the reset source as described below.

Target defined:

- Hardware Reset – All functional units are set to their default states without exception. History is lost.
- System Reset – All functional units except the RCU are set to their default states.
- Core-n only Reset – Affects Core-n only. The system software should guarantee that the core in reset state is not accessed by any bus master.

Source defined:

- Hardware Reset – The $\overline{\text{SYS\_HWRST}}$ input signal is asserted active (pulled down).
- System Reset – May be triggered by software (writing to the RCU_CTL register) or by another functional unit such as the dynamic power management (DPM) unit (Hibernate) or any of the system event controller (SEC), trigger routing unit (TRU), or emulator inputs.
- Core-n-only reset – Triggered by software.
- Trigger request (peripheral).

Voltage Regulation

The processor requires an external voltage regulator to power the V_{DD_INT} pins. To reduce standby power consumption, the external voltage regulator can be signaled through SYS_EXTWAKE 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 supply pins (V_{DD_EXT}, V_{DD_USB}, V_{DD_DMC}) can still be powered, eliminating the need for external buffers. The external voltage regulator can be activated from this power down state by asserting the $\overline{\text{SYS\_HWRST}}$ pin, which then initiates a boot sequence. SYS_EXTWAKE indicates a wakeup to the external voltage regulator.

SYSTEM DEBUG

The processor includes various features that allow for easy system debug. These are described in the following sections.

System Watchpoint Unit

The System Watchpoint Unit (SWU) is a single module which connects to a single system bus and provides for transaction monitoring. One SWU is attached to the bus going to each system slave. The SWU provides ports for all system bus address channel signals. Each SWU contains four match groups of registers with associated hardware. These four SWU match groups operate independently, but share common event (interrupt, trigger and others) outputs.

System Debug Unit

The System Debug Unit (SDU) provides IEEE-1149.1 support through its JTAG interface. In addition to traditional JTAG features, present in legacy Blackfin products, the SDU adds more features for debugging the chip without halting the core processors.

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”.

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.

ADDITIONAL INFORMATION

The following publications that describe the ADSP-BF606/ADSP-BF607/ADSP-BF608/ADSP-BF609 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-BF60x Blackfin Processor Hardware Reference*
- *Blackfin Processor Programming Reference*
- *ADSP-BF60x Blackfin Processor Anomaly List*

RELATED SIGNAL CHAINS

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

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

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

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

ADSP-BF60x DETAILED SIGNAL DESCRIPTIONS

Table 6 provides a detailed description of each signal.

Table 6. Detailed Signal Descriptions

Signal Name	Direction	Description
ACM_An	Output	**ADC Control Signals** Function varies by mode.
ACM_CLK	Output	**Clock** SCLK derived clock for connecting to an ADC.
ACM_FS	Output	**Frame Sync** Typically used as an ADC chip select.
ACM_Tn	Input	**External Trigger n** Input for external trigger events.
CAN_RX	Input	**Receive** Typically an external CAN transceiver's RX output.
CAN_TX	Output	**Transmit** Typically an external CAN transceiver's TX input.
CNT_DG	Input	**Count Down and Gate** Depending on the mode of operation this input acts either as a count down signal or a gate signal. Count Down: This input causes the GP counter to decrement. Gate: Stops the GP counter from incrementing or decrementing.
CNT_UD	Input	**Count Up and Direction** Depending on the mode of operation this input acts either as a count up signal or a direction signal. Count Up: This input causes the GP counter to increment. Direction: Selects whether the GP counter is incrementing or decrementing.
CNT_ZM	Input	**Count Zero Marker** Input that connects to the zero marker output of a rotary device or detects the pressing of a push button.
DMC_Ann	Output	**Address n** Address bus.
DMC_BAn	Output	**Bank Address Input n** Defines which internal bank an ACTIVATE, READ, WRITE, or PRECHARGE command is being applied to on the dynamic memory. Also defines which mode registers (MR, EMR, EMR2, and/or EMR3) are loaded during the LOAD MODE REGISTER command.
$\overline{\text{DMC\_CAS}}$	Output	**Column Address Strobe** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the CAS input of dynamic memory.
$\overline{\text{DMC\_CK}}$	Output	**Clock (complement)** Complement of DMC_CK.
DMC_CK	Output	**Clock** Outputs DCLK to external dynamic memory.
DMC_CKE	Output	**Clock enable** Active high clock enables. Connects to the dynamic memory's CKE input.
$\overline{\text{DMC\_CSn}}$	Output	**Chip Select n** Commands are recognized by the memory only when this signal is asserted.
DMC_DQnn	I/O	**Data n** Bidirectional data bus.
DMC_LDM	Output	**Data Mask for Lower Byte** Mask for DMC_DQ07:DMC_DQ00 write data when driven high. Sampled on both edges of the data strobe by the dynamic memory.
$\overline{\text{DMC\_LDQS}}$	I/O	**Data Strobe for Lower Byte (complement)** Complement of LDQS. Not used in single-ended mode.
DMC_LDQS	I/O	**Data Strobe for Lower Byte** DMC_DQ07:DMC_DQ00 data strobe. Output with Write Data. Input with Read Data. May be single-ended or differential depending on register settings.
DMC_ODT	Output	**On-die Termination** Enables dynamic memory termination resistances when driven high (assuming the memory is properly configured). ODT is enabled/disabled regardless of read or write commands.
$\overline{\text{DMC\_RAS}}$	Output	**Row Address Strobe** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the RAS input of dynamic memory.
DMC_UDM	Output	**Data Mask for Upper Byte** Mask for DMC_DQ15:DMC_DQ08 write data when driven high. Sampled on both edges of the data strobe by the dynamic memory.
$\overline{\text{DMC\_UDQS}}$	I/O	**Data Strobe for Upper Byte (complement)** Complement of UDQS. Not used in single-ended mode.
DMC_UDQS	I/O	**Data Strobe for Upper Byte** DMC_DQ15:DMC_DQ08 data strobe. Output with Write Data. Input with Read Data. May be single-ended or differential depending on register settings.
$\overline{\text{DMC\_WE}}$	Output	**Write Enable** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the $\overline{\text{WE}}$ input of dynamic memory.

Table 6. Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
ETH_CRS	Input	**Carrier Sense/RMII Receive Data Valid** Multiplexed on alternate clock cycles. CRS: Asserted by the PHY when either the transmit or receive medium is not idle. De-asserted when both are idle. RXDV: Asserted by the PHY when the data on RXDn is valid.
ETH_MDC	Output	**Management Channel Clock** Clocks the MDC input of the PHY.
ETH_MDIO	I/O	**Management Channel Serial Data** Bidirectional data bus for PHY control.
ETH_PTPAUXIN	Input	**PTP Auxiliary Trigger Input** Assert this signal to take an auxiliary snapshot of the time and store it in the auxiliary time stamp FIFO.
ETH_PTPCLKIN	Input	**PTP Clock Input** Optional external PTP clock input.
ETH_PTPPPS	Output	**PTP Pulse-Per-Second Output** When the Advanced Time Stamp feature is enabled, this signal is asserted based on the PPS mode selected. Otherwise, PTPPPS is asserted every time the seconds counter is incremented.
ETH_REFCLK	Input	**Reference Clock** Externally supplied Ethernet clock.
ETH_RXDn	Input	**Receive Data n** Receive data bus.
ETH_TXDn	Output	**Transmit Data n** Transmit data bus.
ETH_TXEN	I/O	**Transmit Enable** When asserted indicates that the data on TXDn is valid.
JTG_EMU	Output	**Emulation Output** JTAG emulation flag.
JTG_TCK	Input	**Clock** JTAG test access port clock.
JTG_TDI	Input	**Serial Data In** JTAG test access port data input.
JTG_TDO	Output	**Serial Data Out** JTAG test access port data output.
JTG_TMS	Input	**Mode Select** JTAG test access port mode select.
JTG_TRST	Input	**Reset** JTAG test access port reset.
LP_ACK	I/O	**Acknowledge** Provides handshaking. When the link port is configured as a receiver, ACK is an output. When the link port is configured as a transmitter, ACK is an input.
LP_CLK	I/O	**Clock** When the link port is configured as a receiver, CLK is an input. When the link port is configured as a transmitter, CLK is an output.
LP_Dn	I/O	**Data n** Data bus. Input when receiving, output when transmitting.
PPI_CLK	I/O	**Clock** Input in external clock mode, output in internal clock mode.
PPI_Dnn	I/O	**Data n** Bidirectional data bus.
PPI_FS1	I/O	**Frame Sync 1 (HSYNC)** Behavior depends on PPI mode. See the PPI chapter in the processor hardware reference for more details.
PPI_FS2	I/O	**Frame Sync 2 (VSYNC)** Behavior depends on PPI mode. See the PPI chapter in the processor hardware reference for more details.
PPI_FS3	I/O	**Frame Sync 3 (FIELD)** Behavior depends on PPI mode. See the PPI chapter in the processor hardware reference for more details.
PWM_AH	Output	**Channel A High Side** High side drive signal.
PWM_AL	Output	**Channel A Low Side** Low side drive signal.
PWM_BH	Output	**Channel B High Side** High side drive signal.
PWM_BL	Output	**Channel B Low Side** Low side drive signal.
PWM_CH	Output	**Channel C High Side** High side drive signal.
PWM_CL	Output	**Channel C Low Side** Low side drive signal.
PWM_DH	Output	**Channel D High Side** High side drive signal.
PWM_DL	Output	**Channel D Low Side** Low side drive signal.
PWM_SYNC	Input	**PWM External Sync** This input is for an externally generated sync signal. If the sync signal is internally generated no connection is necessary.
PWM_TRIPn	Input	**Shutdown Input n** When asserted the selected PWM channel outputs are shut down immediately.
Px_nn	I/O	**Position n** General purpose input/output. See the GP Ports chapter in the processor hardware reference for programming information.

Table 6. Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
RSI_CLK	Output	**Clock** The clock signal applied to the connected device from the RSI.
RSI_CMD	I/O	**Command** Used to send commands to and receive responses from the connected device.
RSI_Dn	I/O	**Data n** Bidirectional data bus.
$\overline{\text{SMC\_ABEn}}$	Output	**Byte Enable n** Indicate whether the lower or upper byte of a memory is being accessed. When an asynchronous write is made to the upper byte of a 16-bit memory, $\overline{\text{SMC\_ABE1}}$ =0 and $\overline{\text{SMC\_ABE0}}$ =1. When an asynchronous write is made to the lower byte of a 16-bit memory, $\overline{\text{SMC\_ABE1}}$ =1 and $\overline{\text{SMC\_ABE0}}$ =0.
$\overline{\text{SMC\_AMSn}}$	Output	**Memory Select n** Typically connects to the chip select of a memory device.
SMC_Ann	Output	**Address n** Address bus.
$\overline{\text{SMC\_AOE}}$	Output	**Output Enable** Asserts at the beginning of the setup period of a read access.
SMC_ARDY	Input	**Asynchronous Ready** Flow control signal used by memory devices to indicate to the SMC when further transactions may proceed.
$\overline{\text{SMC\_ARE}}$	Output	**Read Enable** Asserts at the beginning of a read access.
$\overline{\text{SMC\_AWE}}$	Output	**Write Enable** Asserts for the duration of a write access period.
$\overline{\text{SMC\_BG}}$	Output	**Bus Grant** Output used to indicate to an external device that it has been granted control of the SMC buses.
$\overline{\text{SMC\_BGH}}$	Output	**Bus Grant Hang** Output used to indicate that the SMC has a pending transaction which requires control of the bus to be restored before it can be completed.
$\overline{\text{SMC\_BR}}$	Input	**Bus Request** Input used by an external device to indicate that it is requesting control of the SMC buses.
SMC_Dnn	I/O	**Data n** Bidirectional data bus.
SMC_NORCLK	Output	**NOR Clock** Clock for synchronous burst mode.
SMC_NORDV	Output	**NOR Data Valid** Asserts for the duration of a synchronous burst mode read setup period.
SMC_NORWT	Input	**NOR Wait** Flow control signal used by memory devices in synchronous burst mode to indicate to the SMC when further transactions may proceed.
SPI_CLK	I/O	**Clock** Input in slave mode, output in master mode.
SPI_D2	I/O	**Data 2** Used to transfer serial data in quad mode. Open drain in ODM mode.
SPI_D3	I/O	**Data 3** Used to transfer serial data in quad mode. Open drain in ODM mode.
SPI_MISO	I/O	**Master In, Slave Out** Used to transfer serial data. Operates in the same direction as SPI_MOSI in dual and quad modes. Open drain in ODM mode.
SPI_MOSI	I/O	**Master Out, Slave In** Used to transfer serial data. Operates in the same direction as SPI_MISO in dual and quad modes. Open drain in ODM mode.
SPI_RDY	I/O	**Ready** Optional flow signal. Output in slave mode, input in master mode.
$\overline{\text{SPI\_SELn}}$	Output	**Slave Select Output n** Used in master mode to enable the desired slave.
$\overline{\text{SPI\_SS}}$	Input	**Slave Select Input** Slave mode: acts as the slave select input. Master mode: optionally serves as an error detection input for the SPI when there are multiple masters.
SPT_ACLK	I/O	**Channel A Clock** Data and frame sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_AD0	I/O	**Channel A Data 0** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.
SPT_AD1	I/O	**Channel A Data 1** Secondary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.
SPT_AFS	I/O	**Channel A Frame Sync** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally.
SPT_ATDV	Output	**Channel A Transmit Data Valid** This signal is optional and only active when SPORT is configured in multi-channel transmit mode. It is asserted during enabled slots.
SPT_BCLK	I/O	**Channel B Clock** Data and frame sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_BD0	I/O	**Channel B Data 0** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.

Table 6. Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
SPT_BD1	I/O	**Channel B Data 1** Secondary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data.
SPT_BFS	I/O	**Channel B Frame Sync** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally.
SPT_BTDV	Output	**Channel B Transmit Data Valid** This signal is optional and only active when SPORT is configured in multi-channel transmit mode. It is asserted during enabled slots.
SYS_BMODEn	Input	**Boot Mode Control n** Selects the boot mode of the processor.
SYS_CLKIN	Input	**Clock/Crystal Input** Connect to an external clock source or crystal.
SYS_CLKOUT	Output	**Processor Clock Output** Outputs internal clocks. Clocks may be divided down. See the CGU chapter in the processor hardware reference for more details.
SYS_EXTWAKE	Output	**External Wake Control** Drives low during hibernate and high all other times. Typically connected to the enable input of the voltage regulator controlling the V_{DD_INT} supply.
$\overline{\text{SYS\_FAULT}}$	I/O	**Complementary Fault** Complement of SYS_FAULT.
SYS_FAULT	I/O	**Fault** Indicates internal faults or senses external faults depending on the operating mode.
$\overline{\text{SYS\_HWRST}}$	Input	**Processor Hardware Reset Control** Resets the device when asserted.
$\overline{\text{SYS\_IDLEn}}$	Output	**Core n Idle Indicator** When low indicates that core n is in idle mode or being held in reset.
$\overline{\text{SYS\_NMI}}$	Input	**Non-maskable Interrupt** Priority depends on the core that receives the interrupt. See the processor hardware and programming references for more details.
SYS_PWRGD	Input	**Power Good Indicator** When high it indicates to the processor that the V_{DD_INT} level is within specifications such that it is safe to begin booting upon return from hibernate.
$\overline{\text{SYS\_RESOUT}}$	Output	**Reset Output** Indicates that the device is in the reset state.
$\overline{\text{SYS\_SLEEP}}$	Output	**Processor Sleep Indicator** When low indicates that the processor is in the deep sleep power saving mode.
SYS_TDA	Input	**Thermal Diode Anode** May be used by an external temperature sensor to measure the die temperature.
SYS_TDK	Input	**Thermal Diode Cathode** May be used by an external temperature sensor to measure the die temperature.
SYS_XTAL	Output	**Crystal Output** Drives an external crystal. Must be left unconnected if an external clock is driving CLKIN.
TMR_ACIn	Input	**Alternate Capture Input n** Provides an additional input for WIDCAP, WATCHDOG, and PININT modes.
TMR_ACLKn	Input	**Alternate Clock n** Provides an additional time base for use by an individual timer.
TMR_CLK	Input	**Clock** Provides an additional global time base for use by all the GP timers.
TMR_TMRn	I/O	**Timer n** The main input/output signal for each timer.
TWI_SCL	I/O	**Serial Clock** Clock output when master, clock input when slave.
TWI_SDA	I/O	**Serial Data** Receives or transmits data.
$\overline{\text{UART\_CTS}}$	Input	**Clear to Send** Flow control signal.
$\overline{\text{UART\_RTS}}$	Output	**Request to Send** Flow control signal.
$\overline{\text{UART\_RX}}$	Input	**Receive** Receive input. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with.
$\overline{\text{UART\_TX}}$	Output	**Transmit** Transmit output. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with.
USB_CLKIN	Input	**Clock/Crystal Input** This clock input is multiplied by a PLL to form the USB clock. See Universal Serial Bus (USB) On-The-Go—Receive and Transmit Timing for frequency/tolerance information.
USB_DM	I/O	**Data –** Bidirectional differential data line.
USB_DP	I/O	**Data +** Bidirectional differential data line.
USB_ID	Input	**OTG ID** Senses whether the controller is a host or device. This signal is pulled low when an A-type plug is sensed (signifying that the USB controller is the A device), but the input is high when a B-type plug is sensed (signifying that the USB controller is the B device).
USB_VBC	Output	**VBUS Control** Controls an external voltage source to supply VBUS when in host mode. May be configured as open drain. Polarity is configurable as well.
USB_VBUS	I/O	**Bus Voltage** Connects to bus voltage in host and device modes.

349-BALL CSP_BGA SIGNAL DESCRIPTIONS

The processors' pin definitions are shown in the table. The columns in this table provide the following information:

- Signal Name: The Signal Name column in the table includes the Signal Name for every pin.
- Description: The Description column in the table provides a verbose (descriptive) name for the signal.
- Port: The General-Purpose I/O Port column in the table shows whether or not the signal is multiplexed with other signals on a general-purpose I/O port pin.
- Pin Name: The Pin Name column in the table identifies the name of the package pin (at power-on reset) on which the signal is located (if a single function pin) or is multiplexed (if a general-purpose I/O pin).

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions

Signal Name	Description	Port	Pin Name
ACM0_A0	ACM0 Address 0	F	PF_14
ACM0_A1	ACM0 Address 1	F	PF_15
ACM0_A2	ACM0 Address 2	F	PF_12
ACM0_A3	ACM0 Address 3	F	PF_13
ACM0_A4	ACM0 Address 4	F	PF_10
ACM0_CLK	ACM0 Clock	E	PE_04
ACM0_FS	ACM0 Frame Sync	E	PE_03
ACM0_T0	ACM0 External Trigger 0	E	PE_08
ACM0_T1	ACM0 External Trigger 1	G	PG_05
CAN0_RX	CAN0 Receive	G	PG_04
CAN0_TX	CAN0 Transmit	G	PG_01
CNT0_DG	CNT0 Count Down and Gate	G	PG_12
CNT0_UD	CNT0 Count Up and Direction	G	PG_11
CNT0_ZM	CNT0 Count Zero Marker	G	PG_07
DMC0_A00	DMC Address 0	Not Muxed	DMC0_A00
DMC0_A01	DMC Address 1	Not Muxed	DMC0_A01
DMC0_A02	DMC Address 2	Not Muxed	DMC0_A02
DMC0_A03	DMC Address 3	Not Muxed	DMC0_A03
DMC0_A04	DMC Address 4	Not Muxed	DMC0_A04
DMC0_A05	DMC Address 5	Not Muxed	DMC0_A05
DMC0_A06	DMC Address 6	Not Muxed	DMC0_A06
DMC0_A07	DMC Address 7	Not Muxed	DMC0_A07
DMC0_A08	DMC Address 8	Not Muxed	DMC0_A08
DMC0_A09	DMC Address 9	Not Muxed	DMC0_A09
DMC0_A10	DMC Address 10	Not Muxed	DMC0_A10
DMC0_A11	DMC Address 11	Not Muxed	DMC0_A11
DMC0_A12	DMC Address 12	Not Muxed	DMC0_A12
DMC0_A13	DMC Address 13	Not Muxed	DMC0_A13
DMC0_BA0	DMC Bank Address Input 0	Not Muxed	DMC0_BA0
DMC0_BA1	DMC Bank Address Input 1	Not Muxed	DMC0_BA1
DMC0_BA2	DMC Bank Address Input 2	Not Muxed	DMC0_BA2
$\overline{\text{DMC0\_CAS}}$	DMC Column Address Strobe	Not Muxed	$\overline{\text{DMC0\_CAS}}$
DMC0_CK	DMC Clock	Not Muxed	DMC0_CK
DMC0_CKE	DMC Clock Enable	Not Muxed	DMC0_CKE
$\overline{\text{DMC0\_CK}}$	DMC Clock (complement)	Not Muxed	$\overline{\text{DMC0\_CK}}$
$\overline{\text{DMC0\_CS0}}$	DMC Chip Select 0	Not Muxed	$\overline{\text{DMC0\_CS0}}$

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
DMC0_DQ00	DMC Data 0	Not Muxed	DMC0_DQ00
DMC0_DQ01	DMC Data 1	Not Muxed	DMC0_DQ01
DMC0_DQ02	DMC Data 2	Not Muxed	DMC0_DQ02
DMC0_DQ03	DMC Data 3	Not Muxed	DMC0_DQ03
DMC0_DQ04	DMC Data 4	Not Muxed	DMC0_DQ04
DMC0_DQ05	DMC Data 5	Not Muxed	DMC0_DQ05
DMC0_DQ06	DMC Data 6	Not Muxed	DMC0_DQ06
DMC0_DQ07	DMC Data 7	Not Muxed	DMC0_DQ07
DMC0_DQ08	DMC Data 8	Not Muxed	DMC0_DQ08
DMC0_DQ09	DMC Data 9	Not Muxed	DMC0_DQ09
DMC0_DQ10	DMC Data 10	Not Muxed	DMC0_DQ10
DMC0_DQ11	DMC Data 11	Not Muxed	DMC0_DQ11
DMC0_DQ12	DMC Data 12	Not Muxed	DMC0_DQ12
DMC0_DQ13	DMC Data 13	Not Muxed	DMC0_DQ13
DMC0_DQ14	DMC Data 14	Not Muxed	DMC0_DQ14
DMC0_DQ15	DMC Data 15	Not Muxed	DMC0_DQ15
DMC0_LDM	DMC Data Mask for Lower Byte	Not Muxed	DMC0_LDM
DMC0_LDQS	DMC Data Strobe for Lower Byte	Not Muxed	DMC0_LDQS
$\overline{\text{DMC0\_LDQS}}$	DMC Data Strobe for Lower Byte (complement)	Not Muxed	$\overline{\text{DMC0\_LDQS}}$
DMC0_ODT	DMC On-die Termination	Not Muxed	DMC0_ODT
$\overline{\text{DMC0\_RAS}}$	DMC Row Address Strobe	Not Muxed	$\overline{\text{DMC0\_RAS}}$
DMC0_UDM	DMC Data Mask for Upper Byte	Not Muxed	DMC0_UDM
DMC0_UDQS	DMC Data Strobe for Upper Byte	Not Muxed	DMC0_UDQS
$\overline{\text{DMC0\_UDQS}}$	DMC Data Strobe for Upper Byte (complement)	Not Muxed	$\overline{\text{DMC0\_UDQS}}$
$\overline{\text{DMC0\_WE}}$	DMC Write Enable	Not Muxed	$\overline{\text{DMC0\_WE}}$
ETH0_CRS	EMAC0 Carrier Sense/RMII Receive Data Valid	C	PC_05
ETH0_MDC	EMAC0 Management Channel Clock	C	PC_06
ETH0_MDIO	EMAC0 Management Channel Serial Data	C	PC_07
ETH0_PTPPPS	EMAC0 PTP Pulse-Per-Second Output	B	PB_15
ETH0_REFCLK	EMAC0 Reference Clock	B	PB_14
ETH0_RXD0	EMAC0 Receive Data 0	C	PC_00
ETH0_RXD1	EMAC0 Receive Data 1	C	PC_01
ETH0_TXD0	EMAC0 Transmit Data 0	C	PC_02
ETH0_TXD1	EMAC0 Transmit Data 1	C	PC_03
ETH0_TXEN	EMAC0 Transmit Enable	B	PB_13
ETH1_CRS	EMAC1 Carrier Sense/RMII Receive Data Valid	E	PE_13
ETH1_MDC	EMAC1 Management Channel Clock	E	PE_10
ETH1_MDIO	EMAC1 Management Channel Serial Data	E	PE_11
ETH1_PTPPPS	EMAC1 PTP Pulse-Per-Second Output	C	PC_09
ETH1_REFCLK	EMAC1 Reference Clock	G	PG_06
ETH1_RXD0	EMAC1 Receive Data 0	G	PG_00
ETH1_RXD1	EMAC1 Receive Data 1	E	PE_15
ETH1_TXD0	EMAC1 Transmit Data 0	G	PG_03
ETH1_TXD1	EMAC1 Transmit Data 1	G	PG_02
ETH1_TXEN	EMAC1 Transmit Enable	G	PG_05
ETH_PTPAUXIN	EMAC0/EMAC1 PTP Auxiliary Trigger Input	C	PC_11

Table 7. ADSP-BF60x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
ETH_PTPCLKIN	EMAC0/EMAC1 PTP Clock Input	C	PC_13
GND	Ground	Not Muxed	GND
$\overline{\text{JTG\_EMU}}$	Emulation Output	Not Muxed	$\overline{\text{JTG\_EMU}}$
JTG_TCK	JTAG Clock	Not Muxed	JTG_TCK
JTG_TDI	JTAG Serial Data Input	Not Muxed	JTG_TDI
JTG_TDO	JTAG Serial Data Output	Not Muxed	JTG_TDO
JTG_TMS	JTAG Mode Select	Not Muxed	JTG_TMS
$\overline{\text{JTG\_TRST}}$	JTAG Reset	Not Muxed	$\overline{\text{JTG\_TRST}}$
LP0_ACK	LP0 Acknowledge	B	PB_01
LP0_CLK	LP0 Clock	B	PB_00
LP0_D0	LP0 Data 0	A	PA_00
LP0_D1	LP0 Data 1	A	PA_01
LP0_D2	LP0 Data 2	A	PA_02
LP0_D3	LP0 Data 3	A	PA_03
LP0_D4	LP0 Data 4	A	PA_04
LP0_D5	LP0 Data 5	A	PA_05
LP0_D6	LP0 Data 6	A	PA_06
LP0_D7	LP0 Data 7	A	PA_07
LP1_ACK	LP1 Acknowledge	B	PB_02
LP1_CLK	LP1 Clock	B	PB_03
LP1_D0	LP1 Data 0	A	PA_08
LP1_D1	LP1 Data 1	A	PA_09
LP1_D2	LP1 Data 2	A	PA_10
LP1_D3	LP1 Data 3	A	PA_11
LP1_D4	LP1 Data 4	A	PA_12
LP1_D5	LP1 Data 5	A	PA_13
LP1_D6	LP1 Data 6	A	PA_14
LP1_D7	LP1 Data 7	A	PA_15
LP2_ACK	LP2 Acknowledge	E	PE_08
LP2_CLK	LP2 Clock	E	PE_09
LP2_D0	LP2 Data 0	F	PF_00
LP2_D1	LP2 Data 1	F	PF_01
LP2_D2	LP2 Data 2	F	PF_02
LP2_D3	LP2 Data 3	F	PF_03
LP2_D4	LP2 Data 4	F	PF_04
LP2_D5	LP2 Data 5	F	PF_05
LP2_D6	LP2 Data 6	F	PF_06
LP2_D7	LP2 Data 7	F	PF_07
LP3_ACK	LP3 Acknowledge	E	PE_07
LP3_CLK	LP3 Clock	E	PE_06
LP3_D0	LP3 Data 0	F	PF_08
LP3_D1	LP3 Data 1	F	PF_09
LP3_D2	LP3 Data 2	F	PF_10
LP3_D3	LP3 Data 3	F	PF_11
LP3_D4	LP3 Data 4	F	PF_12
LP3_D5	LP3 Data 5	F	PF_13