



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



### FEATURES

Dual symmetric 600 MHz high performance Blackfin cores  
328K bytes of on-chip memory

(see [Memory Architecture on Page 4](#))

Each Blackfin core includes

Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs,  
40-bit shifter

RISC-like register and instruction model for ease of pro-  
gramming and compiler-friendly support

Advanced debug, trace, and performance monitoring

Wide range of operating voltages, (see [Operating Conditions on Page 20](#))

256-ball CSP\_BGA (2 sizes) and 297-ball PBGA  
package options

### PERIPHERALS

Dual 12-channel DMA controllers  
(supporting 24 peripheral DMAs)

2 memory-to-memory DMAs

2 internal memory-to-memory DMAs and 1 internal memory  
DMA controller

12 general-purpose 32-bit timers/counters with PWM  
capability

SPI-compatible port

UART with support for IrDA

Dual watchdog timers

Dual 32-bit core timers

48 programmable flags (GPIO)

On-chip phase-locked loop capable of 0.5× to 64× frequency  
multiplication

2 parallel input/output peripheral interface units supporting  
ITU-R 656 video and glueless interface to analog front end  
ADCs

2 dual channel, full duplex synchronous serial ports support-  
ing eight stereo I<sup>2</sup>S channels

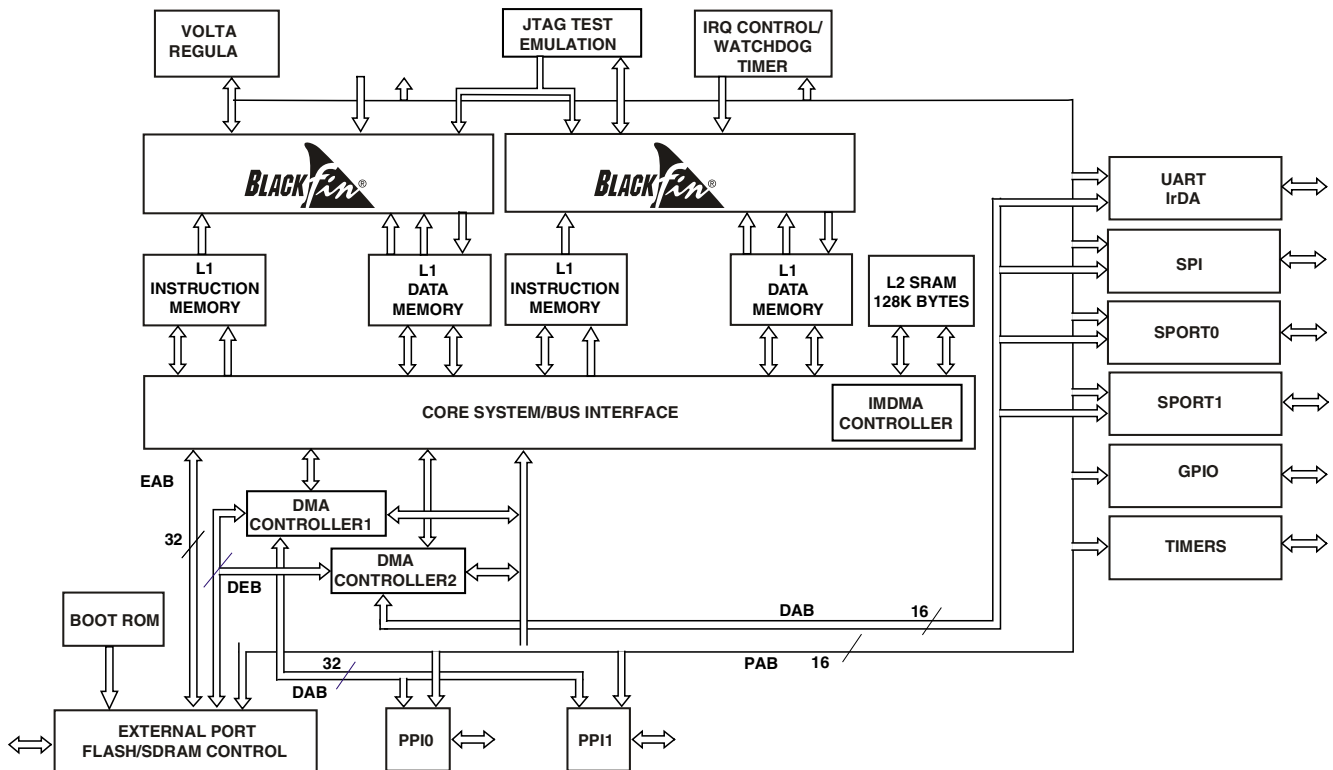


Figure 1. Functional Block Diagram

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

### Rev. E

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.

# ADSP-BF561\* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

---

## COMPARABLE PARTS

View a parametric search of comparable parts.

## EVALUATION KITS

- Low Cost ICE-1000 and High Performance ICE-2000 USB-based JTAG Emulators
- Multimedia Starter Kit
- The ADSP-BF561 EZ-Kit Lite evaluation hardware provides a low-cost hardware solution for evaluating the ADSP-BF561 Blackfin processor.
- USB-Based Emulator and High Performance USB-Based Emulator

## DOCUMENTATION

### Application Notes

- AN-813: Interfacing the ADSP-BF533/ADSP-BF561 Blackfin®; Processors to High Speed Parallel ADCs
- EE-120: Interfacing Assembly Language Programs to C
- EE-126: The ABCs of SDRAM Memories
- EE-175: Emulator and Evaluation Hardware Troubleshooting Guide for VisualDSP++ Users
- EE-183: Rational Sample Rate Conversion with Blackfin® Processors
- EE-185: Fast Floating-Point Arithmetic Emulation on Blackfin® Processors
- EE-228: Switching Regulator Design Considerations for ADSP-BF533 Blackfin® Processors
- EE-261: Understanding Jitter Requirements of PLL-Based Processors
- EE-269: A Beginner's Guide to Ethernet 802.3
- EE-281: Hardware Design Checklist for the Blackfin® Processors
- EE-289: Implementing FAT32 File Systems on ADSP-BF533 Blackfin® Processors
- EE-293: Estimating Power for ADSP-BF561 Blackfin® Processors
- EE-294: Energy-Aware Programming on Blackfin Processors
- EE-300: Interfacing Blackfin® EZ-KIT Lite® Boards to CMOS Image Sensors
- EE-314: Booting the ADSP-BF561 Blackfin® Processor
- EE-323: Implementing Dynamically Loaded Software Modules
- EE-326: Blackfin® Processor and SDRAM Technology
- EE-330: Windows Vista Compatibility in VisualDSP++ 5.0 Development Tools
- EE-332: Cycle Counting and Profiling
- EE-336: Putting ADSP-BF54x Blackfin® Processor Booting into Practice
- EE-339: Using External Switching Regulators with Blackfin® Processors
- EE-340: Connecting SHARC® and Blackfin® Processors over SPI
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users

### Data Sheet

---



- 
- ADSP-BF561: Blackfin Embedded Symmetric Multiprocessor Data Sheet

### Emulator Manuals

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

### Evaluation Kit Manuals

- ADSP-BF561 EZ-KIT Lite<sup>®</sup> Evaluation System Manual
- Blackfin<sup>®</sup> A-V EZ-Extender<sup>®</sup> Manual
- Blackfin<sup>®</sup> EZ-Extender<sup>®</sup> Manual
- Blackfin<sup>®</sup> FPGA EZ-Extender<sup>®</sup> Manual
- Blackfin<sup>®</sup>/SHARC<sup>®</sup> USB EZ-Extender<sup>®</sup> Manual

### Integrated Circuit Anomalies

- ADSP-BF561 Blackfin Anomaly List for Revisions 0.3, 0.5

### Processor Manuals

- ADSP-BF561 Blackfin<sup>®</sup> Processor Hardware Reference
- ADSP-BF5xx/ADSP-BF60x Blackfin<sup>®</sup> Processor Programming Reference
- Blackfin Processors: Manuals

### Product Highlight

- ADSP-BF561 Blackfin Dual-Core Embedded Processor
- Blackfin Processor Family Product Highlight
- EZ-KIT Lite for Analog Devices ADSP-BF561 Blackfin Processor

### Software Manuals

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

- VisualDSP++<sup>®</sup> 5.0 Linker and Utilities Manual
- VisualDSP++<sup>®</sup> 5.0 Loader and Utilities Manual
- VisualDSP++<sup>®</sup> 5.0 Product Release Bulletin
- VisualDSP++<sup>®</sup> 5.0 Quick Installation Reference Card
- VisualDSP++<sup>®</sup> 5.0 Users Guide

## SOFTWARE AND SYSTEMS REQUIREMENTS

- Software and Tools Anomalies Search

## TOOLS AND SIMULATIONS

- ADSP-BF561 Blackfin Processor BSDL File 256-Ball CSP BGA Package
  - ADSP-BF561 Blackfin Processor BSDL File 256-Ball Sparse CSP\_BGA Package
  - ADSP-BF561 Blackfin Processor BSDL File 297-Ball PBGA Package
  - ADSP-BF561 Blackfin Processor Core B BSDL File All Packages
  - Designing with BGA
  - Blackfin Processors Software and Tools
  - ADSP-BF561 Blackfin Processor IBIS Datafile for 12x12 CSP BGA Package (02/2008)
  - ADSP-BF561 Blackfin Processor IBIS Datafile for 17x17 CSP BGA Package (11/2008)
  - ADSP-BF561 Blackfin Processor IBIS Datafile for 27x27 PBGA Package (02/2008)
-

---

## REFERENCE MATERIALS

### Customer Case Studies

- Dahua Case Study
- UTAS Medical Equipment Ensures High Quality Patient Care with Help from Analog Devices

### Technical Articles

- An Efficient Asynchronous Sampling-rate Conversion Algorithm for Multi-channel Audio Applications
- Blackfin Processor Targets Digital Media Open Source Applications
- Blackfin Processor's Parallel Peripheral Interface Simplifies LCD Connection in Portable Multimedia
- Designing IPTV Set-top Boxes Without Getting Boxed In
- Enhance Processor Performance in Open-Source Applications
- High Performance DSPs for Portable Applications
- Is it Really Possible to Play DVD Quality Media While Executing Linux Applications?
- Understanding Advanced Processor Features Promotes Efficient Coding
- Video Filtering Considerations for Media Processors

### White Papers

- A BDTI Analysis of the Analog Devices ADSP-BF5xx
- Blackfin Car Telematics Platform Brings Low Cost Telematics to the Mass Market
- Device-Based Social Networking

- LabVIEW 1000m Below the Waves: Synchronized Sampling of Autonomous Units Through Sound
- Secure, Field Upgradeable OS Architecture for Blackfin
- Security Without Compromise
- Unifying Microarchitecture for Embedded Media Processing

## DESIGN RESOURCES

- ADSP-BF561 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

## DISCUSSIONS

View all ADSP-BF561 EngineerZone Discussions.

## SAMPLE AND BUY

Visit the product page to see pricing options.

## TECHNICAL SUPPORT

Submit a technical question or find your regional support number.

## DOCUMENT FEEDBACK

Submit feedback for this data sheet.

## TABLE OF CONTENTS

Features .....	1	Designing an Emulator-Compatible Processor Board ...	16
Peripherals .....	1	Related Documents .....	16
Table of Contents .....	2	Pin Descriptions .....	17
Revision History .....	2	Specifications .....	20
General Description .....	3	Operating Conditions .....	20
Portable Low Power Architecture .....	3	Electrical Characteristics .....	21
Blackfin Processor Core .....	3	Absolute Maximum Ratings .....	22
Memory Architecture .....	4	Package Information .....	22
DMA Controllers .....	8	ESD Sensitivity .....	22
Watchdog Timer .....	8	Timing Specifications .....	23
Timers .....	9	Output Drive Currents .....	41
Serial Ports (SPORTs) .....	9	Power Dissipation .....	42
Serial Peripheral Interface (SPI) Port .....	9	Test Conditions .....	42
UART Port .....	9	Environmental Conditions .....	44
Programmable Flags (PFx) .....	10	256-Ball CSP_BGA (17 mm) Ball Assignment .....	46
Parallel Peripheral Interface .....	10	256-Ball CSP_BGA (12 mm) Ball Assignment .....	51
Dynamic Power Management .....	11	297-Ball PBGA Ball Assignment .....	56
Voltage Regulation .....	12	Outline Dimensions .....	61
Clock Signals .....	13	Surface-Mount Design .....	63
Bootling Modes .....	14	Automotive Products .....	63
Instruction Set Description .....	14	Ordering Guide .....	63
Development Tools .....	15		

## REVISION HISTORY

### 9/09—Rev. D to Rev. E

Correct all outstanding document errata.

Revised <a href="#">Figure 5</a> .....	13
Added 533 MHz operation <a href="#">Table 10</a> .....	20
Removed reference to 1.8 V operation <a href="#">Table 12</a> .....	21
Added <a href="#">Table 17</a> and <a href="#">Figure 9 Power-Up Reset Timing</a> ....	23
Removed references to $T_j$ from $t_{\text{CLK}}$ parameter <a href="#">Table 20</a> .....	26
Added new SPORT timing parameters and diagram <a href="#">Table 23</a> .....	32
<a href="#">Figure 21</a> .....	33

## GENERAL DESCRIPTION

The ADSP-BF561 processor is a high performance member of the Blackfin<sup>®</sup> family of products targeting a variety of multimedia, industrial, and telecommunications applications. At the heart of this device are two independent Analog Devices Blackfin processors. These Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantage of a clean, orthogonal RISC-like microprocessor instruction set, and single instruction, multiple data (SIMD) multimedia capabilities in a single instruction set architecture.

The ADSP-BF561 processor has 328K bytes of on-chip memory. Each Blackfin core includes:

- 16K bytes of instruction SRAM/cache
- 16K bytes of instruction SRAM
- 32K bytes of data SRAM/cache
- 32K bytes of data SRAM
- 4K bytes of scratchpad SRAM

Additional on-chip memory peripherals include:

- 128K bytes of low latency on-chip L2 SRAM
- Four-channel internal memory DMA controller
- External memory controller with glueless support for SDRAM, mobile SDRAM, SRAM, and flash.

### PORTABLE LOW POWER ARCHITECTURE

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

### BLACKFIN PROCESSOR CORE

As shown in [Figure 2](#), each Blackfin core contains two multiplier/accumulators (MACs), two 40-bit ALUs, four video ALUs, and a single shifter. The computational units process 8-bit, 16-bit, or 32-bit data from the register file.

Each MAC performs a 16-bit by 16-bit multiply in every cycle, with accumulation to a 40-bit result, providing eight bits of extended precision. The ALUs perform a standard set of arithmetic and logical operations. With two ALUs capable of operating on 16-bit or 32-bit data, the flexibility of the computation units covers the signal processing requirements of a varied set of application needs.

Each of the two 32-bit input registers can be regarded as two 16-bit halves, so each ALU can accomplish very flexible single 16-bit arithmetic operations. By viewing the registers as pairs of 16-bit operands, dual 16-bit or single 32-bit operations can be accomplished in a single cycle. By further taking advantage of the second ALU, quad 16-bit operations can be accomplished simply, accelerating the per cycle throughput.

The powerful 40-bit shifter has extensive capabilities for performing shifting, rotating, normalization, extraction, and depositing of data. The data for the computational units is found in a multiported register file of sixteen 16-bit entries or eight 32-bit entries.

A powerful program sequencer controls the flow of instruction execution, including instruction alignment and decoding. The sequencer supports conditional jumps and subroutine calls, as well as zero overhead looping. A loop buffer stores instructions locally, eliminating instruction memory accesses for tight looped code.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches from memory. The DAGs share a register file containing four sets of 32-bit Index, Modify, Length, and Base registers. Eight additional 32-bit registers provide pointers for general indexing of variables and stack locations.

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. Level 2 (L2) memories are other memories, on-chip or off-chip, that may take multiple processor cycles to access. At the L1 level, the instruction memory holds instructions only. The two data memories hold data, and a dedicated scratchpad data memory stores stack and local variable information. At the L2 level, there is a single unified memory space, holding both instructions and data.

In addition, half of L1 instruction memory and half of L1 data memory may be configured as either Static RAMs (SRAMs) or caches. The Memory Management Unit (MMU) provides memory protection for individual tasks that may be operating on the core and may protect system registers from unintended access.

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

The Blackfin instruction set has been optimized so that 16-bit op-codes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit op-codes, 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 assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the VisualDSP C/C++ compiler, resulting in fast and efficient software implementations.

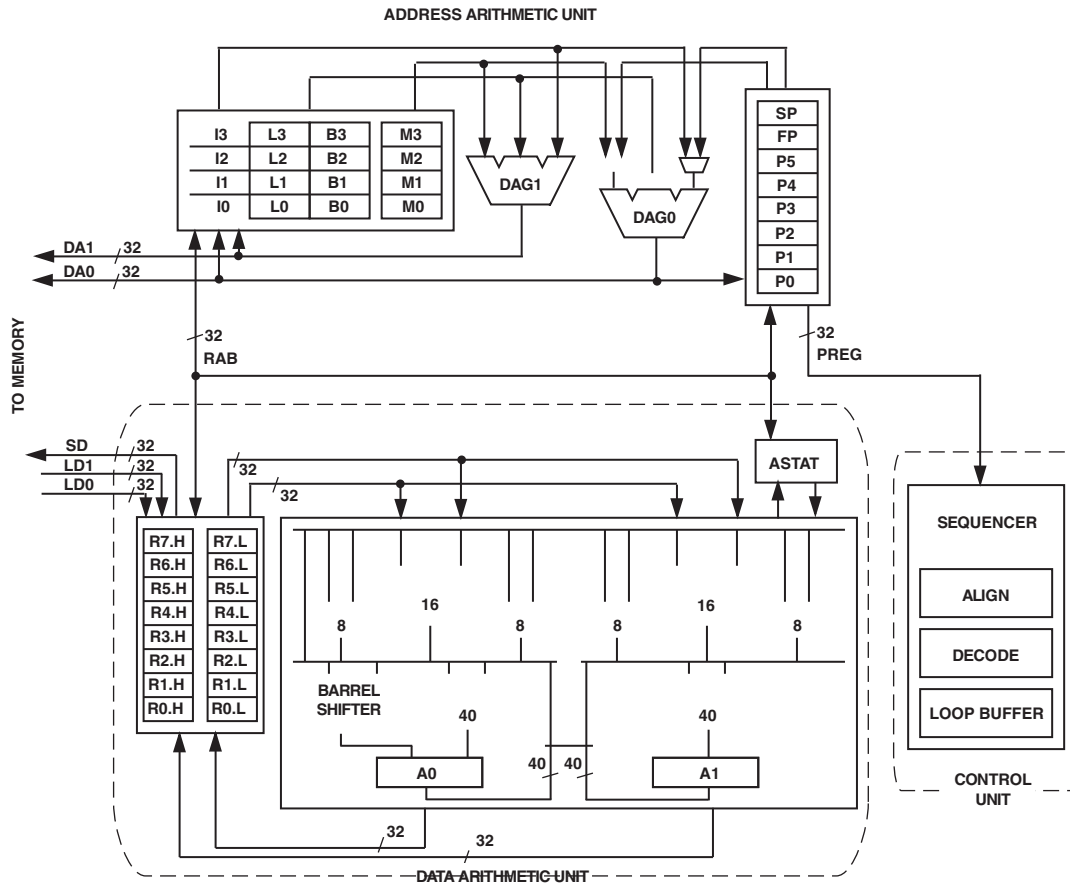


Figure 2. Blackfin Processor Core

## MEMORY ARCHITECTURE

The ADSP-BF561 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 memory as cache or SRAM very close to the processor, and larger, lower cost and performance memory systems farther away from the processor. The ADSP-BF561 memory map is shown in [Figure 3](#).

The L1 memory system in each core is the highest performance memory available to each Blackfin core. The L2 memory provides additional capacity with lower performance. Lastly, the off-chip memory system, accessed through the External Bus Interface Unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing more than 768M bytes of physical memory. The memory DMA controllers provide high bandwidth data movement capability. They can perform block transfers of code or data between the internal L1/L2 memories and the external memory spaces.

## Internal (On-Chip) Memory

The ADSP-BF561 has four blocks of on-chip memory providing high bandwidth access to the core.

The first is the L1 instruction memory of each Blackfin core consisting of 16K bytes of four-way set-associative cache memory and 16K bytes of SRAM. The cache memory may also be configured as an SRAM. This memory is accessed at full processor speed. When configured as SRAM, each of the two 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The second on-chip memory block is the L1 data memory of each Blackfin core which consists of four banks of 16K bytes each. Two of the L1 data memory banks can be configured as one way of a two-way set-associative cache or as an SRAM. The other two banks are configured as SRAM. All banks are accessed at full processor speed. When configured as SRAM, each of the four 16K banks of memory is broken into 4K sub-banks which can be independently accessed by the processor and DMA.

The third memory block associated with each core is a 4K byte scratchpad SRAM which runs at the same speed as the L1 memories, but is only accessible as data SRAM (it cannot be configured as cache memory and is not accessible via DMA).



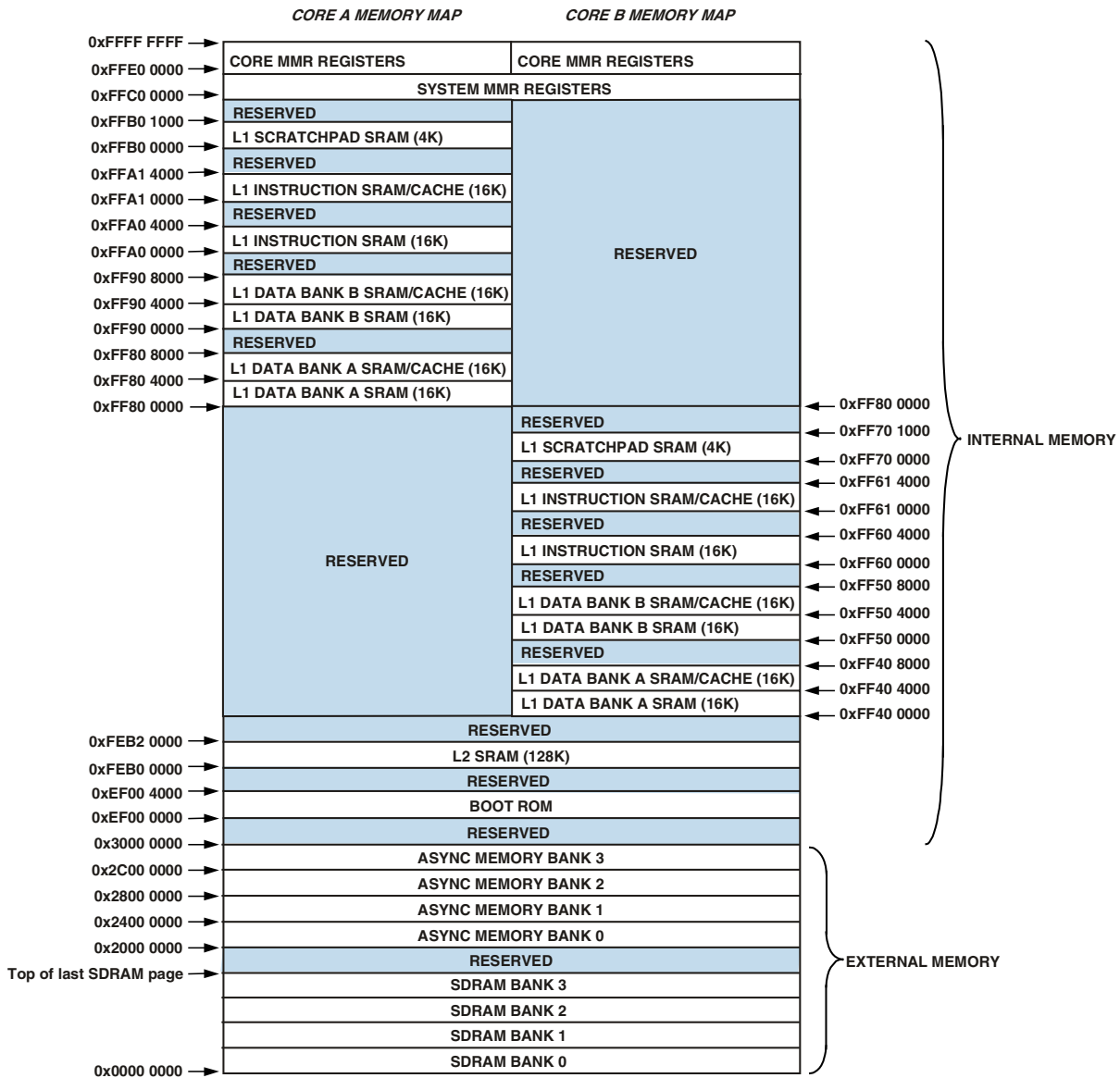


Figure 3. Memory Map

The fourth on-chip memory system is the L2 SRAM memory array which provides 128K bytes of high speed SRAM operating at one half the frequency of the core, and slightly longer latency than the L1 memory banks. The L2 memory is a unified instruction and data memory and can hold any mixture of code and data required by the system design. The Blackfin cores share a dedicated low latency 64-bit wide data path port into the L2 SRAM memory.

Each Blackfin core processor has its own set of core Memory Mapped Registers (MMRs) but share the same system MMR registers and 128K bytes L2 SRAM memory.

### External (Off-Chip) Memory

The ADSP-BF561 external memory is accessed via the External Bus Interface Unit (EBIU). This interface provides a glueless connection to up to four banks of synchronous DRAM (SDRAM) as well as up to four banks of asynchronous memory devices, including flash, EPROM, ROM, SRAM, and memory mapped I/O devices.

The PC133-compliant SDRAM controller can be programmed to interface to up to four banks of SDRAM, with each bank containing between 16M bytes and 128M bytes providing access to up to 512M bytes of SDRAM. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement. This allows

# ADSP-BF561

flexible configuration and upgradability of system memory while allowing the core to view all SDRAM as a single, contiguous, physical address space.

The asynchronous memory controller can also be programmed to control up to four banks of devices with very flexible timing parameters for a wide variety of devices. Each bank occupies a 64M byte segment regardless of the size of the devices used so that these banks will only be contiguous if fully populated with 64M bytes of memory.

## I/O Memory Space

Blackfin processors do 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 core MMRs are accessible only by the core and only in supervisor mode and appear as reserved space by on-chip peripherals. The system MMRs are accessible by the core in supervisor mode and can be mapped as either visible or reserved to other devices, depending on the system protection model desired.

## Booting

The ADSP-BF561 contains a small boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF561 is configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM.

## Event Handling

The event controller on the ADSP-BF561 handles all asynchronous and synchronous events to the processor. The ADSP-BF561 provides event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing of a higher priority event takes precedence over servicing of a lower priority event. The controller provides support for five different types of events:

- Emulation – An emulation event causes the processor to enter emulation mode, allowing command and control of the processor via the JTAG interface.
- Reset – This event resets the processor.
- Nonmaskable Interrupt (NMI) – The NMI event can be generated by the software watchdog timer or by the NMI input signal to the processor. The NMI event is frequently used as a power-down indicator to initiate an orderly shut-down of the system.
- Exceptions – Events that occur synchronously to program flow, i.e., the exception will be taken before the instruction is allowed to complete. Conditions such as data alignment violations or undefined instructions cause exceptions.

- Interrupts – Events that occur asynchronously to program flow. They are caused by timers, peripherals, input pins, and an explicit software instruction.

Each event has an associated register to hold the return address and an associated “return from event” instruction. When an event is triggered, the state of the processor is saved on the supervisor stack.

The ADSP-BF561 event controller consists of two stages: the Core Event Controller (CEC) and the System Interrupt Controller (SIC). The Core Event Controller works with the System Interrupt Controller to prioritize and control all system events. Conceptually, interrupts from the peripherals enter into the SIC, and are then routed directly into the general-purpose interrupts of the CEC.

## Core Event Controller (CEC)

The CEC supports nine general-purpose interrupts (IVG15–7), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest priority interrupts (IVG15–14) are recommended to be reserved for software interrupt handlers, leaving seven prioritized interrupt inputs to support the peripherals of the ADSP-BF561. [Table 1](#) describes the inputs to the CEC, identifies their names in the Event Vector Table (EVT), and lists their priorities.

**Table 1. Core Event Controller (CEC)**

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

## System Interrupt Controller (SIC)

The System Interrupt Controller provides the mapping and routing of events from the many peripheral interrupt sources to the prioritized general-purpose interrupt inputs of the CEC. Although the ADSP-BF561 provides a default mapping, the user can alter the mappings and priorities of interrupt events by

writing the appropriate values into the Interrupt Assignment Registers (SIC\_IAR7–0). Table 2 describes the inputs into the SIC and the default mappings into the CEC.

Table 2. System Interrupt Controller (SIC)

Peripheral Interrupt Event	Default Mapping
PLL Wakeup	IVG7
DMA1 Error (Generic)	IVG7
DMA2 Error (Generic)	IVG7
IMDMA Error	IVG7
PPIO Error	IVG7
PPI1 Error	IVG7
SPORT0 Error	IVG7
SPORT1 Error	IVG7
SPI Error	IVG7
UART Error	IVG7
Reserved	IVG7
DMA1 Channel 0 Interrupt (PPIO)	IVG8
DMA1 Channel 1 Interrupt (PPI1)	IVG8
DMA1 Channel 2 Interrupt	IVG8
DMA1 Channel 3 Interrupt	IVG8
DMA1 Channel 4 Interrupt	IVG8
DMA1 Channel 5 Interrupt	IVG8
DMA1 Channel 6 Interrupt	IVG8
DMA1 Channel 7 Interrupt	IVG8
DMA1 Channel 8 Interrupt	IVG8
DMA1 Channel 9 Interrupt	IVG8
DMA1 Channel 10 Interrupt	IVG8
DMA1 Channel 11 Interrupt	IVG8
DMA2 Channel 0 Interrupt (SPORT0 Rx)	IVG9
DMA2 Channel 1 Interrupt (SPORT0 Tx)	IVG9
DMA2 Channel 2 Interrupt (SPORT1 Rx)	IVG9
DMA2 Channel 3 Interrupt (SPORT1 Tx)	IVG9
DMA2 Channel 4 Interrupt (SPI)	IVG9
DMA2 Channel 5 Interrupt (UART Rx)	IVG9
DMA2 Channel 6 Interrupt (UART Tx)	IVG9
DMA2 Channel 7 Interrupt	IVG9
DMA2 Channel 8 Interrupt	IVG9
DMA2 Channel 9 Interrupt	IVG9
DMA2 Channel 10 Interrupt	IVG9
DMA2 Channel 11 Interrupt	IVG9
Timer0 Interrupt	IVG10
Timer1 Interrupt	IVG10
Timer2 Interrupt	IVG10
Timer3 Interrupt	IVG10
Timer4 Interrupt	IVG10
Timer5 Interrupt	IVG10
Timer6 Interrupt	IVG10

Table 2. System Interrupt Controller (SIC) (Continued)

Peripheral Interrupt Event	Default Mapping
Timer7 Interrupt	IVG10
Timer8 Interrupt	IVG10
Timer9 Interrupt	IVG10
Timer10 Interrupt	IVG10
Timer11 Interrupt	IVG10
Programmable Flags 15–0 Interrupt A	IVG11
Programmable Flags 15–0 Interrupt B	IVG11
Programmable Flags 31–16 Interrupt A	IVG11
Programmable Flags 31–16 Interrupt B	IVG11
Programmable Flags 47–32 Interrupt A	IVG11
Programmable Flags 47–32 Interrupt B	IVG11
DMA1 Channel 12/13 Interrupt (Memory DMA/Stream 0)	IVG8
DMA1 Channel 14/15 Interrupt (Memory DMA/Stream 1)	IVG8
DMA2 Channel 12/13 Interrupt (Memory DMA/Stream 0)	IVG9
DMA2 Channel 14/15 Interrupt (Memory DMA/Stream 1)	IVG9
IMDMA Stream 0 Interrupt	IVG12
IMDMA Stream 1 Interrupt	IVG12
Watchdog Timer Interrupt	IVG13
Reserved	IVG7
Reserved	IVG7
Supplemental Interrupt 0	IVG7
Supplemental Interrupt 1	IVG7

**Event Control**

The ADSP-BF561 provides the user with a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each of the registers is 16 bits wide, while each bit represents a particular event class.

- CEC Interrupt Latch Register (ILAT) – The ILAT register indicates when events have been latched. The appropriate bit is set when the processor has latched the event and cleared when the event has been accepted into the system. This register is updated automatically by the controller, but may also be written to clear (cancel) latched events. This register may be read while in supervisor mode and may only be written while in supervisor mode when the corresponding IMASK bit is cleared.
- CEC Interrupt Mask Register (IMASK) – The IMASK register controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event is unmasked and will be processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, thereby preventing the processor from servicing the event

even though the event may be latched in the ILAT register. This register may be read from or written to while in supervisor mode.

Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.

- CEC Interrupt Pending Register (IPEND) – The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing six 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in [Table 2](#).

- SIC Interrupt Mask Registers (SIC\_IMASKx) – These registers control the masking and unmasking of each peripheral interrupt event. When a bit is set in these registers, that peripheral event is unmasked and will be processed by the system when asserted. A cleared bit in these registers masks the peripheral event, thereby preventing the processor from servicing the event.
- SIC Interrupt Status Registers (SIC\_ISRx) – As multiple peripherals can be mapped to a single event, these registers allow the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt; a cleared bit indicates the peripheral is not asserting the event.
- SIC Interrupt Wakeup Enable Registers (SIC\_IWRx) – By enabling the corresponding bit in these registers, each peripheral can be configured to wake up the processor, should the processor be in a powered-down mode when the event is generated.

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

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

## DMA CONTROLLERS

The ADSP-BF561 has two independent DMA controllers that support automated data transfers with minimal overhead for the DSP cores. DMA transfers can occur between the ADSP-BF561 internal memories and any of its DMA-capable

peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including the SDRAM controller and the asynchronous memory controller. DMA-capable peripherals include the SPORTs, SPI port, UART, and PPIs. Each individual DMA-capable peripheral has at least one dedicated DMA channel.

The ADSP-BF561 DMA controllers support both 1-dimensional (1-D) and 2-dimensional (2-D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

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

Examples of DMA types supported by the ADSP-BF561 DMA controllers include:

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

In addition to the dedicated peripheral DMA channels, each DMA Controller has four memory DMA channels provided for transfers between the various memories of the ADSP-BF561 system. These enable transfers of blocks of data between any of the memories—including external SDRAM, ROM, SRAM, and flash memory—with minimal processor intervention. Memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

Further, the ADSP-BF561 has a four channel Internal Memory DMA (IMDMA) Controller. The IMDMA Controller allows data transfers between any of the internal L1 and L2 memories.

## WATCHDOG TIMER

Each ADSP-BF561 core includes a 32-bit timer, which can 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.

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

## TIMERS

There are 14 programmable timer units in the ADSP-BF561.

Each of the 12 general-purpose timer units can be independently programmed as a Pulse Width Modulator (PWM), internally or externally clocked timer, or pulse width counter. The general-purpose timer units can be used in conjunction with the UART to measure the width of the pulses in the data stream to provide an autobaud detect function for a serial channel. The general-purpose timers can generate interrupts to the processor core providing periodic events for synchronization, either to the processor clock or to a count of external signals.

In addition to the 12 general-purpose programmable timers, another timer is also provided for each core. These extra timers are clocked by the internal processor clock (CCLK) and are typically used as a system tick clock for generation of operating system periodic interrupts.

## SERIAL PORTS (SPORTs)

The ADSP-BF561 incorporates two dual-channel synchronous serial ports (SPORT0 and SPORT1) for serial and multiprocessor communications. The SPORTs support the following features:

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

- DMA operations with single-cycle overhead – Each SPORT can automatically receive and transmit multiple buffers of memory data. The DSP can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts – Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability – Each SPORT supports 128 channels out of a 1,024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

An additional 250 mV of SPORT input hysteresis can be enabled by setting Bit 15 of the PLL\_CTL register. When this bit is set, all SPORT input pins have the increased hysteresis.

## SERIAL PERIPHERAL INTERFACE (SPI) PORT

The ADSP-BF561 processor has an SPI-compatible port that enables the processor to communicate with multiple SPI-compatible devices.

The SPI interface uses three pins for transferring data: two data pins (master output-slave input, MOSI, and master input-slave output, MISO) and a clock pin (serial clock, SCK). An SPI chip select input pin ( $\overline{SPISS}$ ) lets other SPI devices select the processor, and seven SPI chip select output pins ( $\overline{SPISEL7-1}$ ) let the processor select other SPI devices. The SPI select pins are reconfigured programmable flag pins. Using these pins, the SPI port provides a full-duplex, synchronous serial interface which supports both master/slave modes and multimaster environments.

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

The SPI port clock rate is calculated as:

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

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

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

## UART PORT

The ADSP-BF561 processor provides a full-duplex universal asynchronous receiver/transmitter (UART) port, which is fully compatible with PC-standard UARTs. The UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. The UART port includes support for 5 data bits to 8 data bits, 1 stop bit or 2 stop bits, and none, even, or odd parity. The UART port supports two modes of operation:

# ADSP-BF561

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

The baud rate, serial data format, error code generation and status, and interrupts for the UART port are programmable.

The UART programmable features include:

- Supporting bit rates ranging from ( $f_{SCLK}/1,048,576$ ) bits per second to ( $f_{SCLK}/16$ ) bits per second.
- Supporting data formats from seven bits to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as:

$$UART\ Clock\ Rate = \frac{f_{SCLK}}{16 \times UART\_Divisor}$$

Where the 16-bit UART\_Divisor comes from the UART\_DLH register (most significant 8 bits) and UART\_DLL register (least significant 8 bits).

In conjunction with the general-purpose timer functions, autobaud detection is supported.

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

## PROGRAMMABLE FLAGS (PFx)

The ADSP-BF561 has 48 bidirectional, general-purpose I/O, programmable flag (PF47–0) pins. Some programmable flag pins are used by peripherals (see [Pin Descriptions on Page 17](#)). When not used as a peripheral pin, each programmable flag can be individually controlled by manipulation of the flag control, status, and interrupt registers as follows:

- Flag direction control register – Specifies the direction of each individual PFx pin as input or output.
- Flag control and status registers – Rather than forcing the software to use a read-modify-write process to control the setting of individual flags, the ADSP-BF561 employs a “write one to set” and “write one to clear” mechanism that allows any combination of individual flags to be set or cleared in a single instruction, without affecting the level of any other flags. Two control registers are provided, one register is written-to in order to set flag values, while another register is written-to in order to clear flag values. Reading the flag status register allows software to interrogate the sense of the flags.

- Flag interrupt mask registers – These registers allow each individual PFx pin to function as an interrupt to the processor. Similar to the flag control registers that are used to set and clear individual flag values, one flag interrupt mask register sets bits to enable an interrupt function, and the other flag interrupt mask register clears bits to disable an interrupt function. PFx pins defined as inputs can be configured to generate hardware interrupts, while output PFx pins can be configured to generate software interrupts.
- Flag interrupt sensitivity registers – These registers specify whether individual PFx pins are level- or edge-sensitive and specify, if edge-sensitive, whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge sensitivity.

## PARALLEL PERIPHERAL INTERFACE

The ADSP-BF561 processor provides two parallel peripheral interfaces (PPI0, PPI1) that can connect directly to parallel A/D and D/A converters, video encoders and decoders, and other general-purpose peripherals. The PPI consists of a dedicated input clock pin, up to 3 frame synchronization pins, and up to 16 data pins. The input clock supports parallel data rates at up to  $f_{SCLK}/2$  MHz, and the synchronization signals can be configured as either inputs or outputs.

The PPI supports a variety of general-purpose and ITU-R 656 modes of operation. In general-purpose mode, the PPI provides half-duplex, bi-directional data transfer with up to 16 bits of data. Up to 3 frame synchronization signals are also provided. In ITU-R 656 mode, the PPI provides half-duplex, bi-directional transfer of 8- or 10-bit video data. Additionally, on-chip decode of embedded start-of-line (SOL) and start-of-field (SOF) preamble packets is supported.

### General-Purpose Mode Descriptions

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

- Input mode – frame syncs and data are inputs into the PPI.
- Frame capture mode – frame syncs are outputs from the PPI, but data are inputs.
- Output mode – frame syncs and data are outputs from the PPI.

### Input Mode

Input mode is intended for ADC applications, as well as video communication with hardware signaling. In its simplest form, PPI\_FS1 is an external frame sync input that controls when to read data. The PPI\_DELAY MMR allows for a delay (in PPI\_CLK cycles) between reception of this frame sync and the initiation of data reads. The number of input data samples is user programmable and defined by the contents of the PPI\_COUNT register. The PPI supports 8-bit, and 10-bit through 16-bit data, and are programmable in the PPI\_CONTROL register.

## Frame Capture Mode

Frame capture mode allows the video source(s) to act as a slave (e.g., for frame capture). The ADSP-BF561 processors control when to read from the video source(s). PPI\_FS1 is an HSYNC output and PPI\_FS2 is a VSYNC output.

## Output Mode

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

## ITU-R 656 Mode Descriptions

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

- Active video only mode
- Vertical blanking only mode
- Entire field mode

## Active Video Only Mode

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

## Vertical Blanking Interval Mode

In this mode, the PPI only transfers vertical blanking interval (VBI) data.

## Entire Field Mode

In this mode, the entire incoming bit stream is read in through the PPI. This includes active video, control preamble sequences, and ancillary data that may be embedded in horizontal and vertical blanking intervals. Data transfer starts immediately after synchronization to Field 1.

## DYNAMIC POWER MANAGEMENT

The ADSP-BF561 provides four power management modes and one power management state, each with a different performance/power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the ADSP-BF561 peripherals also reduces power consumption. See [Table 3](#) for a summary of the power settings for each mode.

**Table 3. Power Settings**

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

## Full-On Operating Mode—Maximum Performance

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

In the active mode, the PLL is enabled but bypassed. Because the PLL is bypassed, the processor's core clock (CCLK) and system clock (SCLK) run at the input clock (CLKIN) frequency. In this mode, the CLKIN to CCLK multiplier ratio can be changed, although the changes are not realized until the full-on mode is entered. DMA access is available to appropriately configured L1 and L2 memories.

In the active mode, it is possible to disable the PLL through the PLL control register (PLL\_CTL). If disabled, the PLL must be re-enabled before transitioning to the full-on or sleep modes.

## Sleep Operating Mode—High Dynamic Power Savings

The sleep mode reduces power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK), however, continue to operate in this mode. Typically an external event will wake up the processor. When in the sleep mode, assertion of wakeup will cause the processor to sense the value of the BYPASS bit in the PLL control register (PLL\_CTL).

When in the sleep mode, system DMA access is only available to external memory, not to L1 or on-chip L2 memory.

## Deep Sleep Operating Mode—Maximum Dynamic Power Savings

The deep sleep mode maximizes power savings by disabling the clocks to the processor cores (CCLK) and to all synchronous peripherals (SCLK). Asynchronous peripherals will not be able to access internal resources or external memory. This powered-down mode can only be exited by assertion of the reset pin ( $\overline{\text{RESET}}$ ). If BYPASS is disabled, the processor will transition to the full-on mode. If BYPASS is enabled, the processor will transition to the active mode.

## Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all the synchronous peripherals (SCLK). The internal voltage

# ADSP-BF561

regulator for the processor can be shut off by writing b#00 to the FREQ bits of the VR\_CTL register. This disables both CCLK and SCLK. Furthermore, it sets the internal power supply voltage ( $V_{DDINT}$ ) to 0 V to provide the lowest static power dissipation. Any critical information stored internally (memory contents, register contents, etc.) must be written to a nonvolatile storage device prior to removing power if the processor state is to be preserved. Since  $V_{DDEXT}$  is still 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 have power still applied without drawing unwanted current. The internal supply regulator can be woken up by asserting the RESET pin.

## Power Savings

As shown in Table 4, the ADSP-BF561 supports two different power domains. The use of multiple power domains maximizes flexibility, while maintaining compliance with industry standards and conventions. By isolating the internal logic of the ADSP-BF561 into its own power domain, separate from the I/O, the processor can take advantage of Dynamic Power Management, without affecting the I/O devices. There are no sequencing requirements for the various power domains.

**Table 4. ADSP-BF561 Power Domains**

Power Domain	V <sub>DD</sub> Range
All internal logic	$V_{DDINT}$
I/O	$V_{DDEXT}$

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

The dynamic power management feature of the ADSP-BF561 allows both the processor's input voltage ( $V_{DDINT}$ ) and clock frequency ( $f_{CLK}$ ) to be dynamically controlled.

The savings in power dissipation can be modeled using the power savings factor and % power savings calculations.

The power savings factor is calculated as:

$$\begin{aligned} \text{power savings factor} &= \frac{f_{CCLKRED}}{f_{CCLKNOM}} \times \left( \frac{V_{DDINTRED}}{V_{DDINTNOM}} \right)^2 \times \left( \frac{t_{RED}}{t_{NOM}} \right) \end{aligned}$$

where the variables in the equations are:

- $f_{CCLKNOM}$  is the nominal core clock frequency
- $f_{CCLKRED}$  is the reduced core clock frequency
- $V_{DDINTNOM}$  is the nominal internal supply voltage
- $V_{DDINTRED}$  is the reduced internal supply voltage

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

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

The percent power savings is calculated as:

$$\% \text{ power savings} = (1 - \text{power savings factor}) \times 100\%$$

## VOLTAGE REGULATION

The ADSP-BF561 processor provides an on-chip voltage regulator that can generate appropriate  $V_{DDINT}$  voltage levels from the  $V_{DDEXT}$  supply. See [Operating Conditions on Page 20](#) for regulator tolerances and acceptable  $V_{DDEXT}$  ranges for specific models.

Figure 4 shows the typical external components required to complete the power management system. The regulator controls the internal logic voltage levels and is programmable with the voltage regulator control register (VR\_CTL) in increments of 50 mV. To reduce standby power consumption, the internal voltage regulator can be programmed to remove power to the processor core while keeping I/O power ( $V_{DDEXT}$ ) supplied. While in the hibernate state,  $V_{DDEXT}$  can still be applied, thus eliminating the need for external buffers. The voltage regulator can be activated from this power-down state by asserting RESET, which will then initiate a boot sequence. The regulator can also be disabled and bypassed at the user's discretion.

The internal voltage regulation feature is not available on any of the 600 MHz speed grade models or automotive grade models. External voltage regulation is required to ensure correct operation of these parts at 600 MHz.

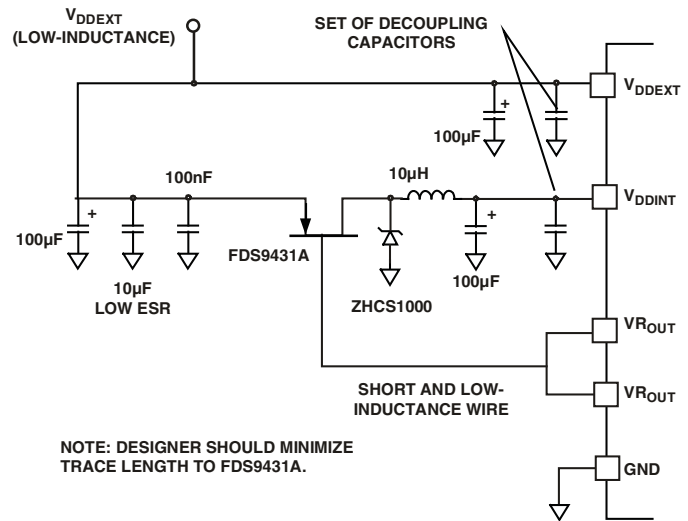


Figure 4. Voltage Regulator Circuit

## Voltage Regulator Layout Guidelines

Regulator external component placement, board routing, and bypass capacitors all have a significant effect on noise injected into the other analog circuits on-chip. The VROUT1-0 traces and voltage regulator external components should be considered as noise sources when doing board layout and should not be routed or placed near sensitive circuits or components on the



board. All internal and I/O power supplies should be well bypassed with bypass capacitors placed as close to the ADSP-BF561 processors as possible.

For further details on the on-chip voltage regulator and related board design guidelines, see the *Switching Regulator Design Considerations for ADSP-BF533 Blackfin Processors (EE-228)* applications note on the Analog Devices web site ([www.analog.com](http://www.analog.com))—use site search on “EE-228”.

## CLOCK SIGNALS

The ADSP-BF561 processor can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

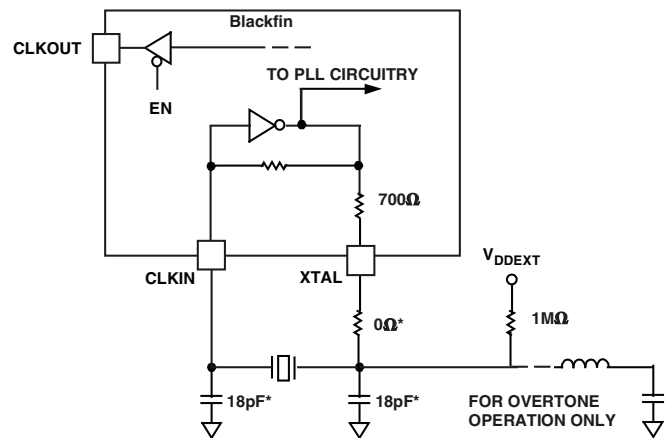
If an external clock is used, it should be a TTL compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor’s CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the ADSP-BF561 processor includes an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 5. A parallel-resonant, fundamental frequency, micro-processor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor shown in Figure 5 fine tune the phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 5 are typical values only. The capacitor values are dependent upon the crystal manufacturer’s load capacitance recommendations and the physical PCB layout. The resistor value depends on the drive level specified by the crystal manufacturer. System designs should verify the customized values based on careful investigation on multiple devices over the allowed temperature range.

A third-overtone crystal can be used at 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 5.

As shown in Figure 6, the core clock (CCLK) and system peripheral clock (SCLK) are derived from the input clock (CLKIN) signal. An on-chip PLL is capable of multiplying the CLKIN signal by a user-programmable 0.5× to 64× multiplication factor. The default multiplier is 10×, but it can be modified by a software instruction sequence. On the fly frequency changes can be effected by simply writing to the PLL\_DIV register.

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL\_DIV register. The values programmed



NOTE: VALUES MARKED WITH \* MUST BE CUSTOMIZED DEPENDING ON THE CRYSTAL AND LAYOUT. PLEASE ANALYZE CAREFULLY.

Figure 5. External Crystal Connections

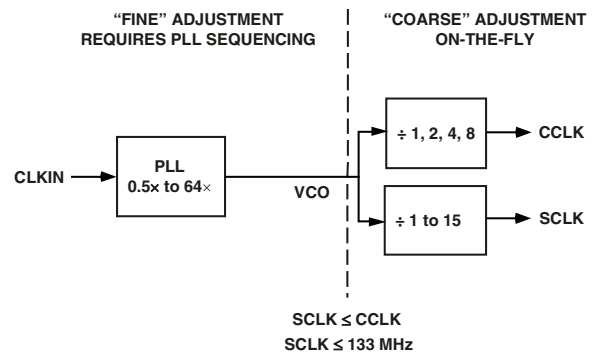


Figure 6. Frequency Modification Methods

into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 5 illustrates typical system clock ratios.

Table 5. Example System Clock Ratios

Signal Name SSEL3–0	Divider Ratio VCO/SCLK	Example Frequency Ratios (MHz)	
		VCO	SCLK
0001	1:1	100	100
0110	6:1	300	50
1010	10:1	500	50

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

# ADSP-BF561

The core clock (CCLK) frequency can also be dynamically changed by means of the CSEL1–0 bits of the PLL\_DIV register. Supported CCLK divider ratios are 1, 2, 4, and 8, as shown in Table 6. This programmable core clock capability is useful for fast core frequency modifications.

**Table 6. Core Clock Ratios**

Signal Name CSEL1–0	Divider Ratio VCO/CCLK	Example Frequency Ratios (MHz)	
		VCO	CCLK
00	1:1	500	500
01	2:1	500	250
10	4:1	200	50
11	8:1	200	25

The maximum PLL clock time when a change is programmed via the PLL\_CTL register is 40  $\mu$ s. The maximum time to change the internal voltage via the internal voltage regulator is also 40  $\mu$ s. The reset value for the PLL\_LOCKCNT register is 0x200. This value should be programmed to ensure a 40  $\mu$ s wakeup time when either the voltage is changed or a new MSEL value is programmed. The value should be programmed to ensure an 80  $\mu$ s wakeup time when both voltage and the MSEL value are changed. The time base for the PLL\_LOCKCNT register is the period of CLKIN.

## BOOTING MODES

The ADSP-BF561 has three mechanisms (listed in Table 7) for automatically loading internal L1 instruction memory, L2, or external memory after a reset. A fourth mode is provided to execute from external memory, bypassing the boot sequence.

**Table 7. Booting Modes**

BMODE1–0	Description
00	Execute from 16-bit external memory (Bypass Boot ROM)
01	Boot from 8-bit/16-bit flash
10	Boot from SPI host slave mode
11	Boot from SPI serial EEPROM (16-, 24-bit address range)

The BMODE pins of the reset configuration register, sampled during power-on resets and software initiated resets, implement the following modes:

- Execute from 16-bit external memory – Execution starts from address 0x2000 0000 with 16-bit packing. The boot ROM is bypassed in this mode. All configuration settings are set for the slowest device possible (3-cycle hold time, 15-cycle R/W access times, 4-cycle setup). Note that, in bypass mode, only Core A can execute instructions from external memory.
- Boot from 8-bit/16-bit external flash memory – The 8-bit/16-bit flash boot routine located in boot ROM memory space is set up using Asynchronous Memory Bank 0.

All configuration settings are set for the slowest device possible (3-cycle hold time; 15-cycle R/W access times; 4-cycle setup).

- Boot from SPI host device – The Blackfin processor operates in SPI slave mode and is configured to receive the bytes of the .LDR file from an SPI host (master) agent. To hold off the host device from transmitting while the boot ROM is busy, the Blackfin processor asserts a GPIO pin, called host wait (HWAIT), to signal the host device not to send any more bytes until the flag is deasserted. The flag is chosen by the user and this information is transferred to the Blackfin processor via bits 10:5 of the FLAG header.
- Boot from SPI serial EEPROM (16-, 24-bit addressable) – The SPI uses the PF2 output pin to select a single SPI EPROM device, submits a read command at address 0x0000, and begins clocking data into the beginning of L1 instruction memory. A 16-, 24-bit addressable SPI-compatible EPROM must be used.

For each of the boot modes, a boot loading protocol is used to transfer program and data blocks from an external memory device to their specified memory locations. Multiple memory blocks may be loaded by any boot sequence. Once all blocks are loaded, Core A program execution commences from the start of L1 instruction SRAM (0xFFA0 0000). Core B remains in a held-off state until Bit 5 of SICA\_SYSCR is cleared by Core A. After that, Core B will start execution at address 0xFF60 0000.

In addition, Bit 4 of the reset configuration register can be set by application code to bypass the normal boot sequence during a software reset. For this case, the processor jumps directly to the beginning of L1 instruction memory.

## INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax that was 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 a user (algorithm/application code) and a supervisor (O/S kernel, device drivers, debuggers, ISRs) mode 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/CPU 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 plus two load/store plus two pointer updates per cycle.

- All registers, I/O, and memory are mapped into a unified 4G byte memory space providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data types; and separate user and kernel 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 as 16-bits.

## DEVELOPMENT TOOLS

The ADSP-BF561 is supported with a complete set of CROSSCORE<sup>†</sup> software and hardware development tools, including Analog Devices emulators and the VisualDSP++<sup>‡</sup> development environment. The same emulator hardware that supports other Analog Devices processors also fully emulates the ADSP-BF561.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy to use assembler that is based on an algebraic syntax, an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ runtime library that includes DSP and mathematical functions. A key point for these tools is C/C++ code efficiency. The compiler has been developed for efficient translation of C/C++ code to Blackfin assembly. The Blackfin processor has architectural features that improve the efficiency of compiled C/C++ code.

The VisualDSP++ debugger has a number of important features. Data visualization is enhanced by a plotting package that offers a significant level of flexibility. This graphical representation of user data enables the programmer to quickly determine the performance of an algorithm. As algorithms grow in complexity, this capability can have increasing significance on the designer's development schedule, increasing productivity. Statistical profiling enables the programmer to nonintrusively poll the processor as it is running the program. This feature, unique to VisualDSP++, enables the software developer to passively gather important code execution metrics without interrupting the real-time characteristics of the program. Essentially, the developer can identify bottlenecks in software quickly and efficiently. By using the profiler, the programmer can focus on those areas in the program that impact performance and take corrective action.

Debugging both C/C++ and assembly programs with the VisualDSP++ debugger, programmers can:

- View mixed C/C++ and assembly code (interleaved source and object information).
- Insert breakpoints.

- Set conditional breakpoints on registers, memory, and stacks.
- Trace instruction execution.
- Perform linear or statistical profiling of program execution.
- Fill, dump, and graphically plot the contents of memory.
- Perform source level debugging.
- Create custom debugger windows.

The VisualDSP++ IDE lets programmers define and manage software development. Its dialog boxes and property pages let programmers configure and manage all development tools, including color syntax highlighting in the VisualDSP++ editor. These capabilities permit programmers to:

- Control how the development tools process inputs and generate outputs.
- Maintain a one-to-one correspondence with the tool's command line switches.

The VisualDSP++ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of embedded, real-time programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning when developing new application code. The VDK features include threads, critical and unscheduled regions, semaphores, events, and device flags. The VDK also supports priority-based, pre-emptive, cooperative, and time-sliced scheduling approaches. In addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

Because the VDK is a library, a developer can decide whether to use it or not. The VDK is integrated into the VisualDSP++ development environment, but can also be used with standard command line tools. When the VDK is used, the development environment assists the developer with many error prone tasks and assists in managing system resources, automating the generation of various VDK-based objects, and visualizing the system state when debugging an application that uses the VDK.

The Expert Linker can be used to visually manipulate the placement of code and data in the embedded system. Memory utilization can be viewed in a color-coded graphical form. Code and data can be easily moved to different areas of the processor or external memory with the drag of the mouse. Runtime stack and heap usage can be examined. The Expert Linker is fully compatible with existing Linker Definition File (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices emulators use the IEEE 1149.1 JTAG test access port of the ADSP-BF561 to monitor and control the target board processor during emulation. The emulator provides full-speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect the loading or timing of the target system.

<sup>†</sup> CROSSCORE is a registered trademark of Analog Devices, Inc.

<sup>‡</sup> VisualDSP++ is a registered trademark of Analog Devices, Inc.

# ADSP-BF561

In addition to the software and hardware development tools available from Analog Devices, third parties provide a wide range of tools supporting the Blackfin processor family. Third party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

## **EZ-KIT Lite Evaluation Board**

For evaluation of ADSP-BF561 processors, use the ADSP-BF561 EZ-KIT Lite<sup>®</sup> board available from Analog Devices. Order part number ADDS-BF561-EZLITE. The board comes with on-chip emulation capabilities and is equipped to enable software development. Multiple daughter cards are available.

## **DESIGNING AN EMULATOR-COMPATIBLE PROCESSOR BOARD**

The Analog Devices family of emulators are tools that every system developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on the ADSP-BF561. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor is set running at full speed with no impact on system timing.

To use these emulators, the target board must include a header that connects the processor's JTAG port to the emulator.

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

## **RELATED DOCUMENTS**

The following publications that describe the ADSP-BF561 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-BF561 Blackfin Processor Hardware Reference*
- *ADSP-BF53x/BF56x Blackfin Processor Programming Reference*
- *ADSP-BF561 Blackfin Processor Anomaly List*



## PIN DESCRIPTIONS

ADSP-BF561 pin definitions are listed in [Table 8](#). In order to maintain maximum function and reduce package size and pin count, some pins have multiple functions. In cases where pin function is reconfigurable, the default state is shown in plain text, while alternate functionality is shown in italics.

All pins are three-stated during and immediately after reset, except the external memory interface, asynchronous memory control, and synchronous memory control pins. These pins are

all driven high, with the exception of CLKOUT, which toggles at the system clock rate. However if  $\overline{BR}$  is active, the memory pins are also three-stated.

All I/O pins have their input buffers disabled, with the exception of the pins that need pull-ups or pull-downs if unused, as noted in [Table 8](#).

**Table 8. Pin Descriptions**

Pin Name	Type	Function	Driver Type <sup>1</sup>
<i>EBIU</i>			
ADDR25–2	O	Address Bus for Async/Sync Access	A
DATA31–0	I/O	Data Bus for Async/Sync Access	A
$\overline{ABE3-0}/\overline{SDQM3-0}$	O	Byte Enables/Data Masks for Async/Sync Access	A
$\overline{BR}$	I	Bus Request (This pin should be pulled HIGH if not used.)	
$\overline{BG}$	O	Bus Grant	A
$\overline{BGH}$	O	Bus Grant Hang	A
<i>EBIU (ASYNC)</i>			
$\overline{AMS3-0}$	O	Bank Select	A
ARDY	I	Hardware Ready Control (This pin should be pulled HIGH if not used.)	
$\overline{AOE}$	O	Output Enable	A
$\overline{AWE}$	O	Write Enable	A
$\overline{ARE}$	O	Read Enable	A
<i>EBIU (SDRAM)</i>			
$\overline{SRAS}$	O	Row Address Strobe	A
$\overline{SCAS}$	O	Column Address Strobe	A
$\overline{SWE}$	O	Write Enable	A
SCKE	O	Clock Enable	A
SCLK0/CLKOUT	O	Clock Output Pin 0	B
SCLK1	O	Clock Output Pin 1	B
SA10	O	SDRAM A10 Pin	A
$\overline{SMS3-0}$	O	Bank Select	A

# ADSP-BF561

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type <sup>1</sup>
<i>PF/SPI/TIMER</i>			
PF0/ $\overline{\text{SPISS}}$ /TMR0	I/O	Programmable Flag/Slave SPI Select/Timer	C
PF1/ $\overline{\text{SPISEL1}}$ /TMR1	I/O	Programmable Flag/SPI Select/Timer	C
PF2/ $\overline{\text{SPISEL2}}$ /TMR2	I/O	Programmable Flag/SPI Select/Timer	C
PF3/ $\overline{\text{SPISEL3}}$ /TMR3	I/O	Programmable Flag/SPI Select/Timer	C
PF4/ $\overline{\text{SPISEL4}}$ /TMR4	I/O	Programmable Flag/SPI Select/Timer	C
PF5/ $\overline{\text{SPISEL5}}$ /TMR5	I/O	Programmable Flag/SPI Select/Timer	C
PF6/ $\overline{\text{SPISEL6}}$ /TMR6	I/O	Programmable Flag/SPI Select/Timer	C
PF7/ $\overline{\text{SPISEL7}}$ /TMR7	I/O	Programmable Flag/SPI Select/Timer	C
PF8	I/O	Programmable Flag	C
PF9	I/O	Programmable Flag	C
PF10	I/O	Programmable Flag	C
PF11	I/O	Programmable Flag	C
PF12	I/O	Programmable Flag	C
PF13	I/O	Programmable Flag	C
PF14	I/O	Programmable Flag	C
PF15/EXT CLK	I/O	Programmable Flag/External Timer Clock Input	C
<i>PPI0</i>			
PPI0D15–8/PF47–40	I/O	PPI Data/Programmable Flag Pins	C
PPI0D7–0	I/O	PPI Data Pins	C
PPI0CLK	I	PPI Clock	
PPI0SYNC1/TMR8	I/O	PPI Sync/Timer	C
PPI0SYNC2/TMR9	I/O	PPI Sync/Timer	C
PPI0SYNC3	I/O	PPI Sync	C
<i>PPI1</i>			
PPI1D15–8/PF39–32	I/O	PPI Data/Programmable Flag Pins	C
PPI1D7–0	I/O	PPI Data Pins	C
PPI1CLK	I	PPI Clock	
PPI1SYNC1/TMR10	I/O	PPI Sync/Timer	C
PPI1SYNC2/TMR11	I/O	PPI Sync/Timer	C
PPI1SYNC3	I/O	PPI Sync	C
<i>SPORT0</i>			
RSCLK0/PF28	I/O	Sport0 Receive Serial Clock/Programmable Flag	D
RFS0/PF19	I/O	Sport0 Receive Frame Sync/Programmable Flag	C
DR0PRI	I	Sport0 Receive Data Primary	
DR0SEC/PF20	I/O	Sport0 Receive Data Secondary/Programmable Flag	C
TSCLK0/PF29	I/O	Sport0 Transmit Serial Clock/Programmable Flag	D
TF50/PF16	I/O	Sport0 Transmit Frame Sync/Programmable Flag	C
DT0PRI/PF18	I/O	Sport0 Transmit Data Primary/Programmable Flag	C
DT0SEC/PF17	I/O	Sport0 Transmit Data Secondary/Programmable Flag	C

Table 8. Pin Descriptions (Continued)

Pin Name	Type	Function	Driver Type <sup>1</sup>
<i>SPORT1</i>			
RSCLK1/PF30	I/O	Sport1 Receive Serial Clock/ <i>Programmable Flag</i>	D
RFS1/PF24	I/O	Sport1 Receive Frame Sync/ <i>Programmable Flag</i>	C
DR1PRI	I	Sport1 Receive Data Primary	
DR1SEC/PF25	I/O	Sport1 Receive Data Secondary/ <i>Programmable Flag</i>	C
TSCLK1/PF31	I/O	Sport1 Transmit Serial Clock/ <i>Programmable Flag</i>	D
TFS1/PF21	I/O	Sport1 Transmit Frame Sync/ <i>Programmable Flag</i>	C
DT1PRI/PF23	I/O	Sport1 Transmit Data Primary/ <i>Programmable Flag</i>	C
DT1SEC/PF22	I/O	Sport1 Transmit Data Secondary/ <i>Programmable Flag</i>	C
<i>SPI</i>			
MOSI	I/O	Master Out Slave In	C
MISO	I/O	Master In Slave Out (This pin should be pulled HIGH through a 4.7 kΩ resistor if booting via the SPI port.)	C
SCK	I/O	SPI Clock	D
<i>UART</i>			
RX/PF27	I/O	UART Receive/ <i>Programmable Flag</i>	C
TX/PF26	I/O	UART Transmit/ <i>Programmable Flag</i>	C
<i>JTAG</i>			
$\overline{\text{EMU}}$	O	Emulation Output	C
TCK	I	JTAG Clock	
TDO	O	JTAG Serial Data Out	C
TDI	I	JTAG Serial Data In	
TMS	I	JTAG Mode Select	
$\overline{\text{TRST}}$	I	JTAG Reset (This pin should be pulled LOW if JTAG is not used.)	
<i>Clock</i>			
CLKIN	I	Clock/Crystal Input (This pin needs to be at a level or clocking.)	
XTAL	O	Crystal Connection	
<i>Mode Controls</i>			
$\overline{\text{RESET}}$	I	Reset (This pin is always active during core power-on.)	
NMIO	I	Nonmaskable Interrupt Core A (This pin should be pulled LOW when not used.)	
NMI1	I	Nonmaskable Interrupt Core B (This pin should be pulled LOW when not used.)	
BMODE1-0	I	Boot Mode Strap (These pins must be pulled to the state required for the desired boot mode.)	
SLEEP	O	Sleep	C
BYPASS	I	PLL BYPASS Control (Pull-up or pull-down Required.)	
<i>Voltage Regulator</i>			
$V_{\text{ROUT}1-0}$	O	External FET Drive	
<i>Supplies</i>			
$V_{\text{DDEXT}}$	P	Power Supply	
$V_{\text{DDINT}}$	P	Power Supply	
GND	G	Power Supply Return	
No Connection	NC	NC	

<sup>1</sup> Refer to Figure 30 on Page 41 to Figure 34 on Page 42.

# ADSP-BF561

## SPECIFICATIONS

Component specifications are subject to change without notice.

### OPERATING CONDITIONS

Parameter	Conditions	Min	Nominal	Max	Unit		
V <sub>DDINT</sub>	Internal Supply Voltage <sup>1</sup>	Non automotive 500 MHz and 533 MHz speed grade models <sup>2</sup>		0.8	1.25	1.375	V
V <sub>DDINT</sub>	Internal Supply Voltage <sup>3</sup>	600 MHz speed grade models <sup>2</sup>		0.8	1.35	1.4185	V
V <sub>DDINT</sub>	Internal Supply Voltage <sup>3</sup>	Automotive grade models <sup>2</sup>		0.95	1.25	1.375	V
V <sub>DDEXT</sub>	External Supply Voltage	Non automotive grade models <sup>2</sup>		2.25	2.5, or 3.3	3.6	V
V <sub>DDEXT</sub>	External Supply Voltage	Automotive grade models <sup>2</sup>		2.7	3.3	3.6	V
V <sub>IH</sub>	High Level Input Voltage <sup>4,5</sup>			2.0		3.6	V
V <sub>IL</sub>	Low Level Input Voltage <sup>5</sup>			-0.3		+0.6	V
T <sub>J</sub>	Junction Temperature	256-Ball CSP_BGA (12 mm × 12 mm) @ T <sub>AMBIENT</sub> = 0°C to +70°C		0		+105	°C
T <sub>J</sub>	Junction Temperature	256-Ball CSP_BGA (17 mm × 17 mm) @ T <sub>AMBIENT</sub> = 0°C to +70°C		0		+95	°C
T <sub>J</sub>	Junction Temperature	256-Ball CSP_BGA (17 mm × 17 mm) @ T <sub>AMBIENT</sub> = -40°C to +85°C		-40		+115	°C
T <sub>J</sub>	Junction Temperature	297-Ball PBGA @ T <sub>AMBIENT</sub> = 0°C to +70°C		0		+95	°C
T <sub>J</sub>	Junction Temperature	297-Ball PBGA @ T <sub>AMBIENT</sub> = -40°C to +85°C		-40		+115	°C

<sup>1</sup> Internal voltage (V<sub>DDINT</sub>) regulator tolerance is -5% to +10% for all models.

<sup>2</sup> See [Ordering Guide on Page 63](#).

<sup>3</sup> The internal voltage regulation feature is not available. External voltage regulation is required to ensure correct operation.

<sup>4</sup> The ADSP-BF561 is 3.3 V tolerant (always accepts up to 3.6 V maximum V<sub>IH</sub>), but voltage compliance (on outputs, V<sub>OH</sub>) depends on the input V<sub>DDEXT</sub>, because V<sub>OH</sub> (maximum) approximately equals V<sub>DDEXT</sub> (maximum). This 3.3 V tolerance applies to bidirectional and input only pins.

<sup>5</sup> Applies to all signal pins.

Table 9 and Table 10 describe the timing requirements for the ADSP-BF561 clocks (t<sub>CCLK</sub> = 1/f<sub>CCLK</sub>). Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock, system clock, and Voltage Controlled Oscillator

(VCO) operating frequencies, as described in [Absolute Maximum Ratings on Page 22](#). Table 11 describes phase-locked loop operating conditions.

**Table 9. Core Clock (CCLK) Requirements—500 MHz and 533 MHz Speed Grade Models<sup>1</sup>**

Parameter	Max	Unit
f <sub>CCLK</sub>	533	MHz
f <sub>CCLK</sub>	500	MHz
f <sub>CCLK</sub>	444	MHz
f <sub>CCLK</sub>	350	MHz
f <sub>CCLK</sub>	300	MHz
f <sub>CCLK</sub>	250	MHz

<sup>1</sup> See [Ordering Guide on Page 63](#).

<sup>2</sup> External Voltage regulation is required on automotive grade models (see [Ordering Guide on Page 63](#)) to ensure correct operation.

<sup>3</sup> Not applicable to automotive grade models. See [Ordering Guide on Page 63](#).

**Table 10. Core Clock (CCLK) Requirements—600 MHz Speed Grade Models<sup>1</sup>**

Parameter	Max	Unit
f <sub>CCLK</sub>	600	MHz
f <sub>CCLK</sub>	533	MHz
f <sub>CCLK</sub>	500	MHz
f <sub>CCLK</sub>	444	MHz
f <sub>CCLK</sub>	350	MHz
f <sub>CCLK</sub>	300	MHz
f <sub>CCLK</sub>	250	MHz

<sup>1</sup> See [Ordering Guide on Page 63](#).

<sup>2</sup> External voltage regulator required to ensure proper operation at 600 MHz.



**Table 11. Phase-Locked Loop Operating Conditions**

Parameter	Min	Max	Unit
Voltage Controlled Oscillator (VCO) Frequency	50	Maximum $f_{\text{CCLK}}$	MHz

**Table 12. System Clock (SCLK) Requirements**

Parameter <sup>1</sup>	Max $V_{\text{DDEXT}} = 2.5\text{V}/3.3\text{V}$	Unit
$f_{\text{SCLK}}$ CLKOUT/SCLK Frequency ( $V_{\text{DDINT}} \geq 1.14\text{ V}$ )	133 <sup>2</sup>	MHz
$f_{\text{SCLK}}$ CLKOUT/SCLK Frequency ( $V_{\text{DDINT}} < 1.14\text{ V}$ )	100	MHz

<sup>1</sup>  $t_{\text{SCLK}} (= 1/f_{\text{SCLK}})$  must be greater than or equal to  $t_{\text{CCLK}}$ .

<sup>2</sup> Rounded number. Guaranteed to  $t_{\text{SCLK}} = 7.5\text{ ns}$ . See Table 20 on Page 26.

## ELECTRICAL CHARACTERISTICS

Parameter	Test Conditions	Min	Typical	Max	Unit
$V_{\text{OH}}$ High Level Output Voltage <sup>1</sup>	$V_{\text{DDEXT}} = 3.0\text{ V}$ , $I_{\text{OH}} = -0.5\text{ mA}$	2.4			V
$V_{\text{OL}}$ Low Level Output Voltage <sup>1</sup>	$V_{\text{DDEXT}} = 3.0\text{ V}$ , $I_{\text{OL}} = 2.0\text{ mA}$			0.4	V
$I_{\text{IH}}$ High Level Input Current <sup>2</sup>	$V_{\text{DDEXT}} = \text{Maximum}$ , $V_{\text{IN}} = V_{\text{DD}}$ Maximum			10.0	$\mu\text{A}$
$I_{\text{IHP}}$ High Level Input Current JTAG <sup>3</sup>	$V_{\text{DDEXT}} = \text{Maximum}$ , $V_{\text{IN}} = V_{\text{DD}}$ Maximum			50.0	$\mu\text{A}$
$I_{\text{IL}}$ <sup>4</sup> Low Level Input Current <sup>2</sup>	$V_{\text{DDEXT}} = \text{Maximum}$ , $V_{\text{IN}} = 0\text{ V}$			10.0	$\mu\text{A}$
$I_{\text{OZH}}$ Three-State Leakage Current <sup>5</sup>	$V_{\text{DDEXT}} = \text{Maximum}$ , $V_{\text{IN}} = V_{\text{DD}}$ Maximum			10.0	$\mu\text{A}$
$I_{\text{OZL}}$ <sup>4</sup> Three-State Leakage Current <sup>5</sup>	$V_{\text{DDEXT}} = \text{Maximum}$ , $V_{\text{IN}} = 0\text{ V}$			10.0	$\mu\text{A}$
$C_{\text{IN}}$ Input Capacitance <sup>6</sup>	$f_{\text{IN}} = 1\text{ MHz}$ , $T_{\text{AMBIENT}} = 25^\circ\text{C}$ , $V_{\text{IN}} = 2.5\text{ V}$		4	8 <sup>7</sup>	pF
$I_{\text{DDHIBERNATE}}$ <sup>8</sup> $V_{\text{DDEXT}}$ Current in Hibernate Mode	CLKIN=0 MHz, $V_{\text{DDEXT}} = 3.65\text{ V}$ with Voltage Regulator Off ( $V_{\text{DDINT}} = 0\text{ V}$ )		50		$\mu\text{A}$
$I_{\text{DDDEEPSLEEP}}$ <sup>9</sup> $V_{\text{DDINT}}$ Current in Deep Sleep Mode	$V_{\text{DDINT}} = 0.8\text{ V}$ , $T_{\text{JUNCTION}} = 25^\circ\text{C}$		70		mA
$I_{\text{DD\_TYP}}$ <sup>9,10</sup> $V_{\text{DDINT}}$ Current	$V_{\text{DDINT}} = 0.8\text{ V}$ , $f_{\text{CCLK}} = 50\text{ MHz}$ , $T_{\text{JUNCTION}} = 25^\circ\text{C}$		127		mA
$I_{\text{DD\_TYP}}$ <sup>9,10</sup> $V_{\text{DDINT}}$ Current	$V_{\text{DDINT}} = 1.25\text{ V}$ , $f_{\text{CCLK}} = 500\text{ MHz}$ , $T_{\text{JUNCTION}} = 25^\circ\text{C}$		660		mA
$I_{\text{DD\_TYP}}$ <sup>9,10</sup> $V_{\text{DDINT}}$ Current	$V_{\text{DDINT}} = 1.35\text{ V}$ , $f_{\text{CCLK}} = 600\text{ MHz}$ , $T_{\text{JUNCTION}} = 25^\circ\text{C}$		818		mA

<sup>1</sup> Applies to output and bidirectional pins.

<sup>2</sup> Applies to input pins except JTAG inputs.

<sup>3</sup> Applies to JTAG input pins (TCK, TDI, TMS, TRST).

<sup>4</sup> Absolute value.

<sup>5</sup> Applies to three-statable pins.

<sup>6</sup> Applies to all signal pins.

<sup>7</sup> Guaranteed, but not tested.

<sup>8</sup> CLKIN must be tied to  $V_{\text{DDEXT}}$  or GND during hibernate.

<sup>9</sup> Maximum current drawn. See *Estimating Power for ADSP-BF561 Blackfin Processors (EE-293)* on the Analog Devices website ([www.analog.com](http://www.analog.com))—use site search on “EE-293”.

<sup>10</sup> Both cores executing 75% dual MAC, 25% ADD instructions with moderate data bus activity.

System designers should refer to *Estimating Power for the ADSP-BF561 (EE-293)*, which provides detailed information for optimizing designs for lowest power. All topics discussed in this section are described in detail in EE-293. Total power dissipation has two components:

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

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. [Electrical Characteristics on Page 21](#) shows the current dissipation for internal circuitry ( $V_{\text{DDINT}}$ ).

# ADSP-BF561

## ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed in [Table 13](#) may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 13. Absolute Maximum Ratings**

Parameter	Value
Internal (Core) Supply Voltage ( $V_{DDINT}$ )	-0.3 V to +1.42 V
External (I/O) Supply Voltage ( $V_{DDEXT}$ )	-0.5 V to +3.8 V
Input Voltage <sup>1</sup>	-0.5 V to +3.8 V
Output Voltage Swing	-0.5 V to $V_{DDEXT} + 0.5$ V
Load Capacitance <sup>2</sup>	200 pF
Storage Temperature Range	-65°C to +150°C
Junction Temperature Under Bias	125°C

<sup>1</sup> Applies to 100% transient duty cycle. For other duty cycles see [Table 14](#).

<sup>2</sup> For proper SDRAM controller operation, the maximum load capacitance is 50 pF (at 3.3 V) or 30 pF (at 2.5 V) for ADDR19-1, DATA15-0, ABE1-0/SDQM1-0, CLKOUT, SCKE, SA10, SRAS, SCAS, SWE, and SMS.

**Table 14. Maximum Duty Cycle for Input Transient Voltage<sup>1</sup>**

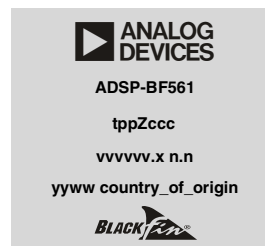
$V_{IN}$ Min (V)	$V_{IN}$ Max (V) <sup>2</sup>	Maximum Duty Cycle
-0.50	3.80	100%
-0.70	4.00	40%
-0.80	4.10	25%
-0.90	4.20	15%
-1.00	4.30	10%

<sup>1</sup> Applies to all signal pins with the exception of CLKIN, XTAL, VROUT1-0.

<sup>2</sup> Only one of the listed options can apply to a particular design.

## PACKAGE INFORMATION

The information presented in [Figure 7](#) and [Table 15](#) provides details about the package branding for the Blackfin processors. For a complete listing of product availability, see the [Ordering Guide on Page 63](#).



*Figure 7. Product Information on Package*

**Table 15. Package Brand Information**

Brand Key	Field Description
t	Temperature Range
pp	Package Type
Z	RoHS Compliant Part
ccc	See Ordering Guide
vvvvvv.x	Assembly Lot Code
n.n	Silicon Revision
yyww	Date Code

## ESD SENSITIVITY



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