



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

- Up to 533 MHz high performance Blackfin processor
- Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bit shifter
- RISC-like register and instruction model for ease of programming and compiler friendly support
- Advanced debug, trace, and performance monitoring
- Wide range of operating voltages; see [Operating Conditions on Page 26](#)
- Qualified for automotive applications
- Programmable on-chip voltage regulator
- 316-ball Pb-free CSP\_BGA package

### MEMORY

- 148K bytes of on-chip memory
- 16K bytes of instruction SRAM/cache
- 64K bytes of instruction SRAM
- 32K bytes of data SRAM
- 32K bytes of data SRAM/cache
- 4K bytes of scratchpad SRAM
- Optional 8M bit parallel flash with boot option
- Memory management unit providing memory protection

- External memory controller with glueless support for SDRAM, SRAM, flash, and ROM
- Flexible memory booting options from SPI and external memory

### PERIPHERALS

- Parallel peripheral interface (PPI), supporting ITU-R 656 video data formats
- 4 dual-channel, full-duplex synchronous serial ports, supporting 16 stereo I<sup>2</sup>S channels
- 2 DMA controllers supporting 26 peripheral DMAs
- 4 memory-to-memory DMAs
- Controller area network (CAN) 2.0B controller
- Media transceiver (MXVR) for connection to a MOST network
- 3 SPI-compatible ports
- Three 32-bit timer/counters with PWM support
- 3 UARTs with support for IrDA
- 2 TWI controllers compatible with I<sup>2</sup>C industry standard
- Up to 38 general-purpose I/O pins (GPIO)
- Up to 16 general-purpose flag pins (GPF)
- Real-time clock, watchdog timer, and 32-bit core timer
- On-chip PLL capable of frequency multiplication
- Debug/JTAG interface

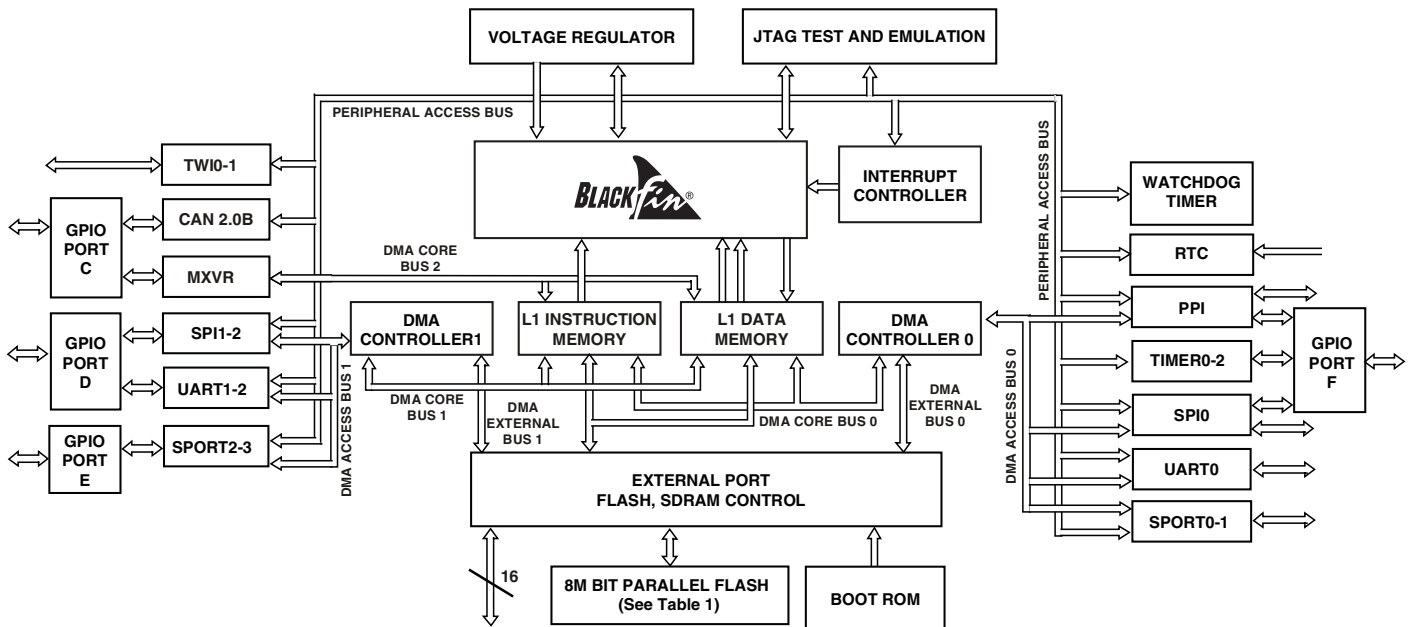


Figure 1. Functional Block Diagram

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

### Rev. F

### Document Feedback

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-BF539/ADSP-BF539F

## TABLE OF CONTENTS

Features .....	1	Booting Modes .....	16
Memory .....	1	Instruction Set Description .....	17
Peripherals .....	1	Development Tools .....	17
General Description .....	3	Example Connections and Layout Considerations .....	18
Low Power Architecture .....	3	MXVR Board Layout Guidelines .....	18
System Integration .....	3	Voltage Regulator Layout Guidelines .....	19
ADSP-BF539/ADSP-BF539F Processor Peripherals .....	3	Additional Information .....	20
Blackfin Processor Core .....	4	Related Signal Chains .....	20
Memory Architecture .....	5	Pin Descriptions .....	21
DMA Controllers .....	8	Specifications .....	26
Real-Time Clock .....	9	Operating Conditions .....	26
Watchdog Timer .....	9	Electrical Characteristics .....	27
Timers .....	9	Absolute Maximum Ratings .....	30
Serial Ports (SPORTs) .....	10	ESD Sensitivity .....	30
Serial Peripheral Interface (SPI) Ports .....	10	Package Information .....	30
2-Wire Interface .....	10	Timing Specifications .....	31
UART Ports .....	10	Output Drive Currents .....	50
Programmable I/O Pins .....	11	Test Conditions .....	52
Parallel Peripheral Interface .....	12	Thermal Characteristics .....	55
Controller Area Network (CAN) Interface .....	12	316-Ball CSP_BGA Ball Assignment .....	56
Media Transceiver MAC layer (MXVR) .....	13	Outline Dimensions .....	59
Dynamic Power Management .....	13	Surface-Mount Design .....	59
Voltage Regulation .....	15	Ordering Guide .....	60
Clock Signals .....	15		

## REVISION HISTORY

### 10/13—Rev. E to Rev. F

Updated <a href="#">Development Tools</a> .....	17
Added notes to Table 32 in <a href="#">Serial Ports—Enable and Three-State</a> .....	43
Added <a href="#">Timer Clock Timing</a> .....	48
Revised <a href="#">Timer Cycle Timing</a> .....	48

To view product/process change notifications (PCNs) related to this data sheet revision, please visit the processor's product page on the [www.analog.com](http://www.analog.com) website and use the View PCN link.

## GENERAL DESCRIPTION

The ADSP-BF539/ADSP-BF539F processors are members of the Blackfin® family of products, incorporating the Analog Devices, Inc./Intel Micro Signal Architecture (MSA). Blackfin processors combine a dual-MAC, state-of-the-art signal processing engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction set architecture.

The ADSP-BF539/ADSP-BF539F processors are completely code compatible with other Blackfin processors, differing only with respect to performance, peripherals, and on-chip memory. These features are shown in [Table 1](#).

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

**Table 1. Processor Features**

Feature	ADSP-BF539	ADSP-BF539F8
SPORTs	4	4
UARTs	3	3
SPI	3	3
TWI	2	2
CAN	1	1
MXVR	1	1
PPI	1	1
Internal 8M bit Parallel Flash	—	1
Instruction SRAM/Cache	16K bytes	16K bytes
Instruction SRAM	64K bytes	64K bytes
Data SRAM/Cache	32K bytes	32K bytes
Data SRAM	32K bytes	32K bytes
Scratchpad	4K bytes	4K bytes
Maximum Frequency	533 MHz 1066 MMACS	533 MHz 1066 MMACS
Package Option	BC-316	BC-316

## 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 simply varying the frequency of operation. This translates into longer battery life and lower heat dissipation.

## SYSTEM INTEGRATION

The ADSP-BF539/ADSP-BF539F processors are highly integrated system-on-a-chip solutions for the next generation of industrial and automotive applications including audio and video signal processing. By combining advanced memory configurations, such as on-chip flash memory, with industry-standard interfaces with a high performance signal processing core, users can develop cost-effective solutions quickly without the need for costly external components. The system peripherals include a MOST Network Media Transceiver (MXVR), three UART ports, three SPI ports, four serial ports (SPORT), one CAN interface, two 2-wire interfaces (TWI), four general-purpose timers (three with PWM capability), a real-time clock, a watchdog timer, a parallel peripheral interface, general-purpose I/O, and general-purpose flag pins.

## ADSP-BF539/ADSP-BF539F PROCESSOR PERIPHERALS

The ADSP-BF539/ADSP-BF539F processors contain a rich set of peripherals connected to the core via several high bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see [Figure 1 on Page 1](#)). The general-purpose peripherals include functions such as UART, timers with PWM (pulse-width modulation) and pulse measurement capability, general-purpose flag I/O pins, a real-time clock, and a watchdog timer. This set of functions satisfies a wide variety of typical system support needs and is augmented by the system expansion capabilities of the device. In addition to these general-purpose peripherals, the processors contain high speed serial and parallel ports for interfacing to a variety of audio, video, and modem codec functions. An MXVR transceiver transmits and receives audio and video data and control information on a MOST automotive multimedia network. A CAN 2.0B controller is provided for automotive control networks. An interrupt controller manages interrupts from the on-chip peripherals or external sources. And power management control functions tailor the performance and power characteristics of the processor and system to many application scenarios.

All of the peripherals, GPIO, CAN, TWI, real-time clock, and timers, are supported by a flexible DMA structure. There are also four separate memory DMA channels dedicated to data transfers between the processor's various memory spaces, including external SDRAM and asynchronous memory. Multiple on-chip buses running at up to 133 MHz provide enough bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

The ADSP-BF539/ADSP-BF539F processors include an on-chip voltage regulator in support of the processor's dynamic power management capability. The voltage regulator provides a range of core voltage levels from  $V_{DDEXT}$ . The voltage regulator can be bypassed at the user's discretion.

# ADSP-BF539/ADSP-BF539F

## BLACKFIN PROCESSOR CORE

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

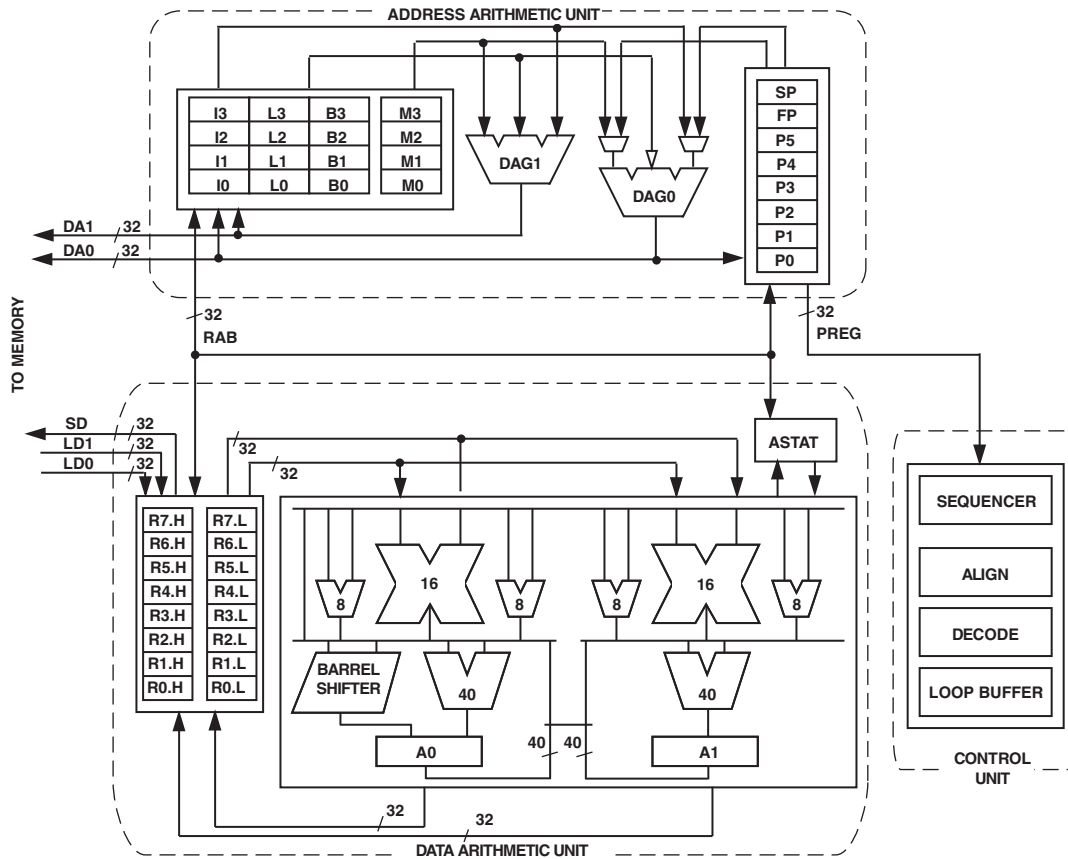


Figure 2. Blackfin Processor Core

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). By also using the second ALU, quad 16-bit operations are possible.

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

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

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

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 two data memories hold 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 can be operating on the core and can protect system registers from unintended access.

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

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

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

## MEMORY ARCHITECTURE

The ADSP-BF539/ADSP-BF539F processors view 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 on-chip memory as cache or SRAM, and larger, lower cost and performance off-chip memory systems. See [Figure 3](#).

The L1 memory system is the primary highest performance memory available to the Blackfin processor. The off-chip memory system, accessed through the external bus interface unit (EBIU), provides expansion with SDRAM, flash memory, and SRAM, optionally accessing up to 132M bytes of physical memory.

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

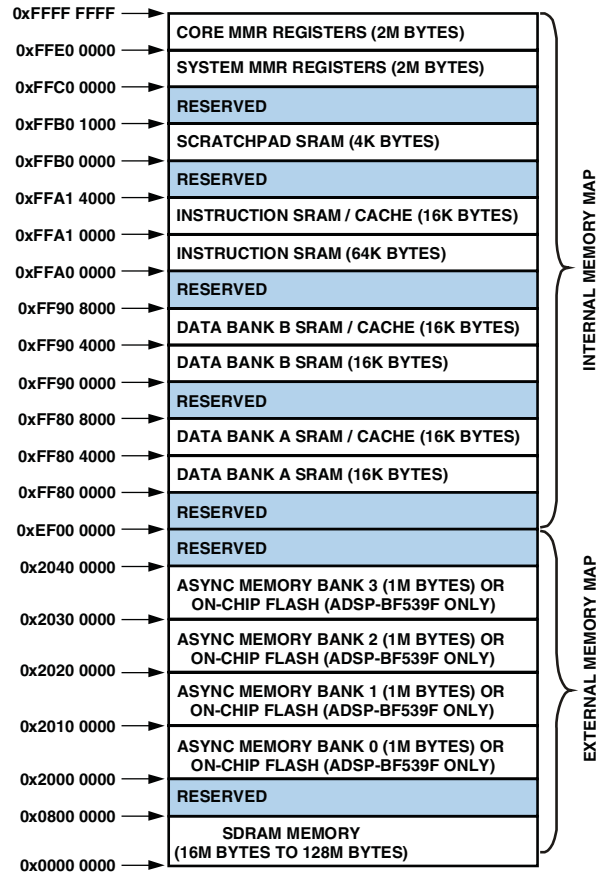


Figure 3. ADSP-BF539/ADSP-BF539F Internal/External Memory Map

### Internal (On-Chip) Memory

The ADSP-BF539/ADSP-BF539F processor has three blocks of on-chip memory, providing high bandwidth access to the core.

The first is the L1 instruction memory, consisting of 80K bytes SRAM, of which 16K bytes can be configured as a four-way set-associative cache. This memory is accessed at full processor speed.

The second on-chip memory block is the L1 data memory, consisting of two banks of up to 32K bytes each. Each memory bank is configurable, offering both cache and SRAM functionality. This memory block is accessed at full processor speed.

The third memory block is a 4K byte scratch pad SRAM, which runs at the same speed as the L1 memories, but is only accessible as data SRAM and cannot be configured as cache memory.

### External (Off-Chip) Memory

External memory is accessed via the EBIU. This 16-bit interface provides a glueless connection to a bank 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.

# ADSP-BF539/ADSP-BF539F

The PC133-compliant SDRAM controller can be programmed to interface to up to 128M bytes of SDRAM. The SDRAM controller allows one row to be open for each internal SDRAM bank, for up to four internal SDRAM banks, improving overall system performance.

The asynchronous memory controller can 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 1M byte segment regardless of the size of the devices used, so that these banks will only be contiguous if each is fully populated with 1M byte of memory.

## Flash Memory (ADSP-BF539F Only)

The ADSP-BF539F8 processor contains a separate flash die, connected to the EBIU bus, within the package of the processor. Figure 4 shows how the flash memory die and Blackfin processor die are connected.

The ADSP-BF539F8 contains an 8M bit (512K × 16-bit) bottom boot sector Spansion S29AL008J known good die flash memory. Additional information for this product can be found in the Spansion data sheet at [www.spansion.com](http://www.spansion.com). Features include the following:

- Access times as fast as 70 ns (EBIU registers must be set appropriately)
- Sector protection
- One million write cycles per sector
- 20 year data retention

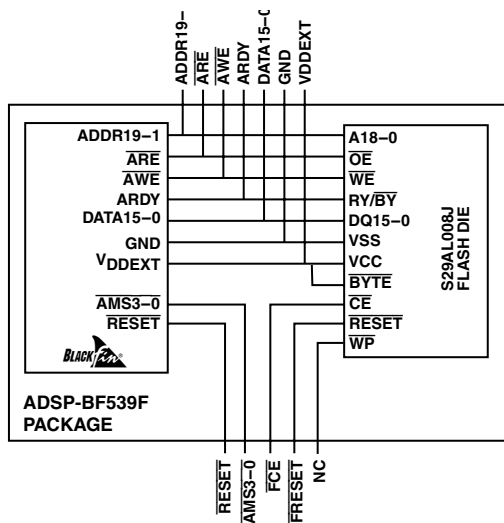


Figure 4. Internal Connection of Flash Memory (ADSP-BF539F8)

The Blackfin processor connects to the flash memory die with address, data, chip enable, write enable, and output enable controls as if it were an external memory device. Note that the write-protect input pin to the flash is not connected and inaccessible, disabling this feature.

The flash chip enable pin  $\overline{FCE}$  must be connected to  $\overline{AMS0}$  or  $\overline{AMS3-1}$  through a printed circuit board trace. When connected to  $\overline{AMS0}$ , the Blackfin processor can boot from the flash die. When connected to  $\overline{AMS3-1}$ , the flash memory appears as non-volatile memory in the processor memory map, shown in Figure 3.

## Flash Memory Programming

The ADSP-BF539F8 flash memory can be programmed before or after mounting on the printed circuit board.

To program the flash prior to mounting on the printed circuit board, use a hardware programming tool that can provide the data, address, and control stimuli to the flash die through the external pins on the package. During this programming,  $V_{DDEXT}$  and GND must be provided to the package and the Blackfin must be held in reset with bus request ( $\overline{BR}$ ) asserted and a CLKIN provided.

The VisualDSP++ tools can be used to program the flash memory after the device is mounted on a printed circuit board.

## Flash Memory Sector Protection

To use the sector protection feature, a high voltage (+8.5 V to +12.5 V) must be applied to the flash FRESET pin. Refer to the flash data sheet for details.

## 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 of which contains the control MMRs for all core functions, and the other of 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 ADSP-BF539/ADSP-BF539F processors contain a small boot kernel, which configures the appropriate peripheral for booting. If the processors are configured to boot from boot ROM memory space, they start executing from the on-chip boot ROM. For more information, see [Booting Modes on Page 16](#).

## Event Handling

The event controller handles all asynchronous and synchronous events to the processor. The processors provide 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 and undefined instructions cause exceptions.
- Interrupts – Events that occur asynchronously to program flow. They are caused by input pins, timers, and other peripherals, as well as by an explicit software instruction.

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

The ADSP-BF539/ADSP-BF539F processor's 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 processor's peripherals. [Table 2](#) describes the inputs to the CEC, identifies their names in the event vector table (EVT), and lists their priorities.

### System Interrupt Controller (SIC)

The system interrupt controller (SIC) 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-BF539/ADSP-BF539F processors provide a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the interrupt assignment registers (SIC\_IARx). [Table 3](#) describes the inputs into the SIC and the default mappings into the CEC.

### Event Control

The ADSP-BF539/ADSP-BF539F processors provide 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 register is 32 bits wide:

- 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 is cleared when the event has been accepted into the system. This register is updated automatically by the controller, but it can 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, preventing the processor from servicing the event even though the event can be latched in the ILAT register. This register can be read or written while in supervisor mode. 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 whether the event is currently active or nested at some level. This register is updated automatically by the controller but can be read while in supervisor mode.

The SIC allows further control of event processing by providing three 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in [Table 3 on Page 8](#).

- 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, 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 that the peripheral is asserting the interrupt, and a cleared bit indicates that the peripheral is not asserting the event.
- SIC interrupt wake-up enable registers (SIC\_IWRx) – By enabling the corresponding bit in these registers, a peripheral can be configured to wake up the processor, should the core be idled or in sleep mode when the event is generated. (For more information, see [Dynamic Power Management on Page 13](#).)

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

The appropriate ILAT register bit is set when an interrupt rising edge is detected (detection requires two core clock cycles). The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC will recognize and queue the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the

# ADSP-BF539/ADSP-BF539F

general-purpose interrupt to the IPEND output asserted is three core clock cycles; however, the latency can be much higher, depending on the activity within and the state of the processor.

**Table 2. Core Event Controller (CEC)**

Priority (0 is Highest)	Event Class	EVT Entry
0	Emulation/Test Control	EMU
1	Reset	RST
2	Nonmaskable Interrupt	NMI
3	Exception	E VX
4	Reserved	—
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

**Table 3. System and Core Event Mapping**

Event Source	Core Event Name
PLL Wake-Up Interrupt	IVG7
DMA Controller 0 Error	IVG7
DMA Controller 1 Error	IVG7
PPI Error Interrupt	IVG7
SPORT0 Error Interrupt	IVG7
SPORT1 Error Interrupt	IVG7
SPORT2 Error Interrupt	IVG7
SPORT3 Error Interrupt	IVG7
MXVR Synchronous Data Interrupt	IVG7
SPI0 Error Interrupt	IVG7
SPI1 Error Interrupt	IVG7
SPI2 Error Interrupt	IVG7
UART0 Error Interrupt	IVG7
UART1 Error Interrupt	IVG7
UART2 Error Interrupt	IVG7
CAN Error Interrupt	IVG7
Real-Time Clock Interrupt	IVG8
DMA0 Interrupt (PPI)	IVG8
DMA1 Interrupt (SPORT0 Rx)	IVG9
DMA2 Interrupt (SPORT0 Tx)	IVG9

**Table 3. System and Core Event Mapping (Continued)**

Event Source	Core Event Name
DMA3 Interrupt (SPORT1 Rx)	IVG9
DMA4 Interrupt (SPORT1 Tx)	IVG9
DMA8 Interrupt (SPORT2 Rx)	IVG9
DMA9 Interrupt (SPORT2 Tx)	IVG9
DMA10 Interrupt (SPORT3 Rx)	IVG9
DMA11 Interrupt (SPORT3 Tx)	IVG9
DMA5 Interrupt (SPI0)	IVG10
DMA14 Interrupt (SPI1)	IVG10
DMA15 Interrupt (SPI2)	IVG10
DMA6 Interrupt (UART0 Rx)	IVG10
DMA7 Interrupt (UART0 Tx)	IVG10
DMA16 Interrupt (UART1 Rx)	IVG10
DMA17 Interrupt (UART1 Tx)	IVG10
DMA18 Interrupt (UART2 Rx)	IVG10
DMA19 Interrupt (UART2 Tx)	IVG10
Timer0, Timer1, Timer2 Interrupts	IVG11
TWI0 Interrupt	IVG11
TWI1 Interrupt	IVG11
CAN Receive Interrupt	IVG11
CAN Transmit Interrupt	IVG11
MXVR Status Interrupt	IVG11
MXVR Control Message Interrupt	IVG11
MXVR Asynchronous Packet Interrupt	IVG11
Programmable Flags Interrupts	IVG12
MDMA0 Stream 0 Interrupt	IVG13
MDMA0 Stream 1 Interrupt	IVG13
MDMA1 Stream 0 Interrupt	IVG13
MDMA1 Stream 1 Interrupt	IVG13
Software Watchdog Timer	IVG13

## DMA CONTROLLERS

The processors have multiple, independent DMA controllers that support automated data transfers with minimal overhead for the processor core. DMA transfers can occur between the ADSP-BF539/ADSP-BF539F processor 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 ports, UARTs, and PPI. Each individual DMA capable peripheral has at least one dedicated DMA channel. In addition, the MXVR peripheral has its own dedicated DMA controller, which supports its own unique set of operating modes.

The 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 deinterleaved on the fly.

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

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

In addition to the dedicated peripheral DMA channels, there are four memory DMA channels provided for transfers between the various memories of the ADSP-BF539/ADSP-BF539F processor system. This enables 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.

## REAL-TIME CLOCK

The ADSP-BF539/ADSP-BF539F processor real-time clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 kHz crystal external to the Blackfin processors. The RTC peripheral has dedicated power supply pins so that it can remain powered up and clocked even when the rest of the processor is in a low power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, hour, or day clock ticks, interrupt on programmable stopwatch count-down, or interrupt at a programmed alarm time.

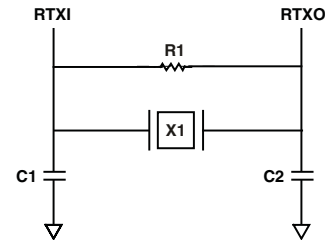
The 32.768 kHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60-second counter, a 60-minute counter, a 24-hour counter, and an 32,768-day counter.

When enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register. There are two alarms: the first alarm is for a time of day. The second alarm is for a day and time of that day.

The stopwatch function counts down from a programmed value, with one second resolution. When the stopwatch is enabled and the counter underflows, an interrupt is generated.

Like the other peripherals, the RTC can wake up the processor from sleep mode upon generation of any RTC wake-up event. Additionally, an RTC wake-up event can wake up the processor from deep sleep mode, and wake up the on-chip internal voltage regulator from a powered down state.

Connect RTC pins RTXI and RTXO with external components as shown in Figure 5.



**SUGGESTED COMPONENTS:**  
 ECLIPTEK EC38J (THROUGH-HOLE PACKAGE)  
 EPSON MC405 12 pF LOAD (SURFACE-MOUNT PACKAGE)  
 C1 = 22pF  
 C2 = 22pF  
 R1 = 10M $\Omega$

**NOTE:** C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1. CONTACT CRYSTAL MANUFACTURER FOR DETAILS. C1 AND C2 SPECIFICATIONS ASSUME BOARD TRACE CAPACITANCE OF 3pF.

Figure 5. External Components for RTC

## WATCHDOG TIMER

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

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

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

## TIMERS

There are four general-purpose programmable timer units in the ADSP-BF539/ADSP-BF539F processors. Three timers have an external pin that can be configured either as a pulse-width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to

# ADSP-BF539/ADSP-BF539F

an external clock input to the PF1 pin (TACLK), an external clock input to the PPI\_CLK pin (TMRCLK), or to the internal SCLK.

The timer units can be used in conjunction with UART0 to measure the width of the pulses in the data stream to provide an auto-baud detect function for a serial channel.

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

In addition to the three general-purpose programmable timers, a fourth timer is also provided. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generation of operating system periodic interrupts.

## SERIAL PORTS (SPORTS)

The ADSP-BF539/ADSP-BF539F processors incorporate four dual-channel synchronous serial ports 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 16 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 processor components and shift registers for shifting data in and out of the data registers.
- Clocking – Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from ( $f_{SCLK}/131,070$ ) Hz to ( $f_{SCLK}/2$ ) Hz.
- Word length – Each SPORT supports serial data words from 3 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 processor can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts – Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability – Each SPORT supports 128 channels out of a 1024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

## SERIAL PERIPHERAL INTERFACE (SPI) PORTS

The processors incorporate three SPI-compatible ports that enable the processor to communicate with multiple SPI compatible devices.

The SPI interface uses three pins for transferring data: two data pins (master output-slave input, MOSIx, and master input-slave output, MISOx) and a clock pin (serial clock, SCKx). An SPI chip select input pin ( $\overline{SPIxSS}$ ) lets other SPI devices select the processor. For SPI0, seven SPI chip select output pins ( $\overline{SPI0SEL7-1}$ ) let the processor select other SPI devices. SPI1 and SPI2 each have a single SPI chip select output pin ( $\overline{SPI1SEL1}$  and  $\overline{SPI2SEL1}$ ) for SPI point-to-point communication. Each of the SPI select pins is a reconfigured GPIO pin. Using these pins, the SPI ports provide a full-duplex, synchronous serial interface, which supports both master/slave modes and multimaster environments.

The SPI ports' baud rate and clock phase/polarities are programmable, and they each have an integrated DMA controller, configurable to support transmit or receive data streams. Each SPI DMA controller can only service unidirectional accesses at any given time.

The SPI port clock rate is calculated as:

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

where the 16-bit  $SPIx\_BAUD$  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.

## 2-WIRE INTERFACE

The processors incorporate two 2-wire interface (TWI) modules that are compatible with the Philips Inter-IC bus standard. The TWI modules offer the capabilities of simultaneous master and slave operation, support for 7-bit addressing, and multimedia data arbitration. The TWI also includes master clock synchronization and support for clock low extension.

The TWI interface uses two pins for transferring clock (SCLx) and data (SDAx) and supports the protocol at speeds up to 400 kbps.

The TWI interface pins are compatible with 5 V logic levels.

## UART PORTS

The processors incorporate three full-duplex universal asynchronous receiver/transmitter (UART) ports, which are fully compatible with PC standard UARTs. The UART ports provide a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA supported, asynchronous transfers of serial data. The UART ports include support for 5 data bits to 8 data bits, 1 stop bit or 2 stop bits, and none, even, or odd parity. The UART ports support two modes of operation:

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

Each UART port’s baud rate, serial data format, error code generation and status, and interrupts are programmable:

- Supporting bit rates ranging from ( $f_{SCLK}/1,048,576$ ) to ( $f_{SCLK}/16$ ) bits per second.
- Supporting data formats from 7 bits to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

Each 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 *UARTx\_DLH* register (most significant 8 bits) and *UARTx\_DLL* register (least significant 8 bits).

In conjunction with the general-purpose timer functions, auto-baud detection is supported on *UART0*.

The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA<sup>®</sup>) Serial Infrared Physical Layer Link Specification (SIR) protocol.

## PROGRAMMABLE I/O PINS

The ADSP-BF539/ADSP-BF539F processor has numerous peripherals that may not all be required for every application. Therefore, many of the pins have a secondary function as programmable I/O pins. There are two types of programmable I/O pins with slightly different functionality: programmable flags and general-purpose I/O.

### Programmable Flags (GPIO Port F)

There are 16 bidirectional, general-purpose programmable flag (PF15–0) pins on GPIO Port F. Each programmable flag can be individually controlled by manipulation of the flag control, status, and interrupt registers:

- Flag direction control register – Specifies the direction of each individual PFX pin as input or output.
- Flag control and status registers – The processors employ a “write one to modify” mechanism that allows any combination of individual flags to be modified in a single instruction, without affecting the level of any other flags. Four control registers are provided. One register is written in order to set flag values, one register is written in order to clear flag values, one register is written in order to toggle flag values, and one register is written in order to specify a flag value. Reading the flag status register allows software to interrogate the sense of the flags.

- Flag interrupt mask registers – The two flag interrupt mask registers allow each individual PFX pin to function as an interrupt to the processor. Similar to the two flag control registers that are used to set and clear individual flag values, one flag interrupt mask register sets bits to enable interrupt function, and the other flag interrupt mask register clears bits to disable interrupt function. PFX pins defined as inputs can be configured to generate hardware interrupts, while output PFX pins can be triggered by software interrupts.
- Flag interrupt sensitivity registers – The two flag interrupt sensitivity 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.

The PFX pins can also be used by the SPI0 and PPI ports as shown in [Table 4](#), depending on how the peripherals are configured. Care must be taken so that these pins are not used for multiple purposes simultaneously.

### General-Purpose I/O Ports C, D, and E

There are 38 general-purpose I/O pins that are multiplexed with other peripherals. They are arranged into Ports C, D, and E as shown in [Table 4](#). The GPIO differ from the programmable flags on Port F in that the GPIO pins cannot generate interrupts to the processor.

**Table 4. Programmable Flag/GPIO Ports**

Peripheral	Alternate Programmable Flag/ GPIO Port Function
PPI	PF15–3
SPORT2	PE7–0
SPORT3	PE15–8
SPI0	PF7–0
SPI1	PD4–0
SPI2	PD9–5
UART1	PD11–10
UART2	PD13–12
CAN	PC1–0 <sup>1</sup>
MXVR	PC9–4 <sup>1</sup>

<sup>1</sup> PC1 and PC4 are open-drain when configured as GPIO outputs.

The general-purpose I/O pins can be individually controlled by manipulation of the control and status registers. These pins will not cause interrupts to be generated to the processor but can be polled to determine their status.

- GPIO direction control register – Specifies the direction of each individual GPIOx pin as input or output.
- GPIO control and status registers – The processors employ a “write one to modify” mechanism that allows any combination of individual GPIO pins to be modified in a single

# ADSP-BF539/ADSP-BF539F

instruction, without affecting the level of any other GPIO pin. Four control registers and a data register are provided for each GPIO port. One register is written in order to set GPIO pin values, one register is written in order to clear GPIO pin values, one register is written in order to toggle GPIO pin values, and one register is written in order to specify a GPIO input or output. Reading the GPIO data register allows software to determine the state of the input GPIO pins.

Note that the GP pin is used to specify the status of the GPIO pins PC9–PC4 at power up. If GP is tied high, then pins PC9–PC4 are configured as GPIO after reset. The pins cannot be reconfigured through software, and special care must be taken with the MLF pin. If the GP pin is tied low, then the pins are configured as MXVR pins after reset but can be reconfigured as GPIO pins through software.

## PARALLEL PERIPHERAL INTERFACE

The ADSP-BF539/ADSP-BF539F processors provide a parallel peripheral interface (PPI) that can connect directly to parallel ADC and DAC 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 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, bidirectional 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, bidirectional transfer of 8- or 10-bit video data. Additionally, on-chip decode of embedded start-of-line (SOL) and start-of-field (SOF) preamble packets are 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

This 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

This mode allows the video source(s) to act as a slave (e.g., for frame capture). The 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

This 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

This mode is used when only the active video portion of a field is of interest and not any of the blanking intervals. The PPI will 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 can be embedded in horizontal and vertical blanking intervals. Data transfer starts immediately after synchronization to Field 1.

## CONTROLLER AREA NETWORK (CAN) INTERFACE

The ADSP-BF539/ADSP-BF539F processors provide a CAN controller that is a communication controller implementing the controller area network (CAN) V2.0B protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. CAN is well suited for control applications due to its ability to communicate reliably over a network since the protocol incorporates CRC checking, message error tracking, and fault node confinement.

The CAN controller is based on a 32-entry mailbox RAM and supports both the standard and extended identifier (ID) message formats specified in the CAN protocol specification, Revision 2.0, Part B.

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

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

The electrical characteristics of each network connection are very stringent; therefore, the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The ADSP-BF539/ADSP-BF539F CAN module represents the controller part of the interface. This module's network I/O is a single transmit output and a single receive input, which connect to a line transceiver.

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

## MEDIA TRANSCEIVER MAC LAYER (MXVR)

The ADSP-BF539/ADSP-BF539F processors provide a media transceiver (MXVR) MAC layer, allowing the processor to be connected directly to a MOST network through just an FOT or electrical PHY.

The MXVR is fully compatible with industry standard standalone MOST controller devices, supporting 22.579 Mbps or 24.576 Mbps data transfer. It offers faster lock times, greater jitter immunity, and a sophisticated DMA scheme for data transfers. The high speed internal interface to the core and L1 memory allows the full bandwidth of the network to be utilized. The MXVR can operate as either the network master or as a network slave.

Synchronous data is transferred to or from the synchronous data channels through eight programmable DMA engines. The synchronous data DMA engines can operate in various modes, including modes that trigger DMA operation when data patterns are detected in the receive data stream. Furthermore, two DMA engines support asynchronous traffic and control message traffic.

Interrupts are generated when a user-defined amount of synchronous data has been sent or received by the processor or when asynchronous packets or control messages have been sent or received.

The MXVR peripheral can wake up the processor from sleep mode when a wake-up preamble is received over the network or based on any other MXVR interrupt event. Additionally, detection of network activity by the MXVR can be used to wake up the processor from sleep mode and wake up the on-chip internal voltage regulator from the powered-down hibernate state. These features allow the processor to operate in a low-power state when there is no network activity or when data is not currently being received or transmitted by the MXVR.

The MXVR clock is provided through a dedicated external crystal or crystal oscillator. For 44.1 kHz frame syncs, use a 45.1584 MHz crystal or oscillator; for 48 kHz frame syncs, use a 49.152 MHz crystal or oscillator. If using a crystal to provide the MXVR clock, use a parallel-resonant, fundamental mode, microprocessor-grade crystal.

## DYNAMIC POWER MANAGEMENT

The ADSP-BF539/ADSP-BF539F processors provide four operating modes, each with a different performance/power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the ADSP-BF539/ADSP-BF539F processor peripherals also reduces power consumption. See Table 5 for a summary of the power settings for each mode.

### Full-On Operating Mode—Maximum Performance

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

### Active Operating Mode—Moderate Dynamic Power Savings

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

In the active mode, it is possible to disable the 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.

**Table 5. 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

# ADSP-BF539/ADSP-BF539F

## Sleep Operating Mode—High Dynamic Power Savings

The sleep mode reduces dynamic power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK), however, continue to operate in this mode. Typically, an external event or RTC activity wakes up the processor. When in the sleep mode, assertion of a wake-up event enabled in the SIC\_IWRx register causes the processor to sense the value of the BYPASS bit in the PLL control register (PLL\_CTL). If BYPASS is disabled, the processor transitions to the full on mode. If BYPASS is enabled, the processor will transition to the active mode. When in the sleep mode, system DMA access to L1 memory is not supported.

## Deep Sleep Operating Mode—Maximum Dynamic Power Savings

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

## Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all the synchronous peripherals (SCLK). The internal voltage regulator for the processor can be shut off by writing b#00 to the FREQ bits of the VR\_CTL register. This sets the internal power supply voltage ( $V_{\text{DDINT}}$ ) 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  $V_{\text{DDEXT}}$  can still be supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that may be connected to the processor to still have power applied without drawing unwanted current. The internal supply regulator can be woken up either by a real-time clock wake-up, by CAN bus traffic, by asserting the  $\overline{\text{RESET}}$  pin, or by an external source via the GPW pin.

## Power Savings

As shown in Table 6, the ADSP-BF539/ADSP-BF539F processors support five different power domains. The use of multiple power domains maximizes flexibility, while maintaining compliance with industry standards and conventions:

- The 3.3 V  $V_{\text{DDRTC}}$  power domain supplies the RTC I/O and logic so that the RTC can remain functional when the rest of the chip is powered off.
- The 3.3 V MXEVDD power domain supplies the MXVR crystal and is separate to provide noise isolation.
- The 1.25 V MPIVDD power domain supplies the MXVR PLL and is separate to provide noise isolation.

- The 1.25 V  $V_{\text{DDINT}}$  power domain supplies all internal logic except for the RTC logic and the MXVR PLL.
- The 3.3 V  $V_{\text{DDEXT}}$  power domain supplies all I/O except for the RTC and MXVR crystals.

There are no sequencing requirements for the various power domains.

**Table 6. Power Domains**

Power Domain	$V_{\text{DD}}$ Range
RTC Crystal I/O and Logic	$V_{\text{DDRTC}}$
MXVR Crystal I/O	MXEVDD
MXVR PLL Analog and Logic	MPIVDD
All Internal Logic Except RTC and MXVR PLL	$V_{\text{DDINT}}$
All I/O Except RTC and MXVR Crystals	$V_{\text{DDEXT}}$

The  $V_{\text{DDRTC}}$  should either be connected to an isolated supply such as a battery (if the RTC is to operate while the rest of the chip is powered down) or should be connected to the  $V_{\text{DDEXT}}$  plane on the board. The  $V_{\text{DDRTC}}$  should remain powered when the processor is in hibernate state and should also remain powered even if the RTC functionality is not being used in an application. The MXEVDD should be connected to the  $V_{\text{DDEXT}}$  plane on the board at a single location with local bypass capacitors. The MXEVDD should remain powered when the processor is in hibernate state and should also remain powered even when the MXVR functionality is not being used in an application. The MPIVDD should be connected to the  $V_{\text{DDINT}}$  plane on the board at a single location through a ferrite bead with local bypass capacitors.

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-BF539/ADSP-BF539F processors allow both the processor input voltage ( $V_{\text{DDINT}}$ ) and clock frequency ( $f_{\text{CCLK}}$ ) 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

*Power Savings Factor*

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

where:

$f_{\text{CCLKNOM}}$  is the nominal core clock frequency.

$f_{\text{CCLKRED}}$  is the reduced core clock frequency.

$V_{\text{DDINTNOM}}$  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 Power Savings Factor is calculated as

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

## VOLTAGE REGULATION

The Blackfin processors provide an on-chip voltage regulator that can generate appropriate  $V_{DDINT}$  voltage levels from the  $V_{DDEXT}$  supply. See [Operating Conditions on Page 26](#) for regulator tolerances and acceptable  $V_{DDEXT}$  ranges for specific models.<sup>†</sup>

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 I/O power ( $V_{DDRTC}$ ,  $MXEVD$ ,  $V_{DDEXT}$ ) is still supplied. While in the hibernate state, I/O power is still being applied, eliminating the need for external buffers. The voltage regulator can be activated from this power-down state through an RTC wake-up, a CAN wake-up, an MXVR wake-up, a general-purpose wake-up, or by asserting  $\overline{\text{RESET}}$ , all of which will then initiate a boot sequence. The regulator can also be disabled and bypassed at the user's discretion.

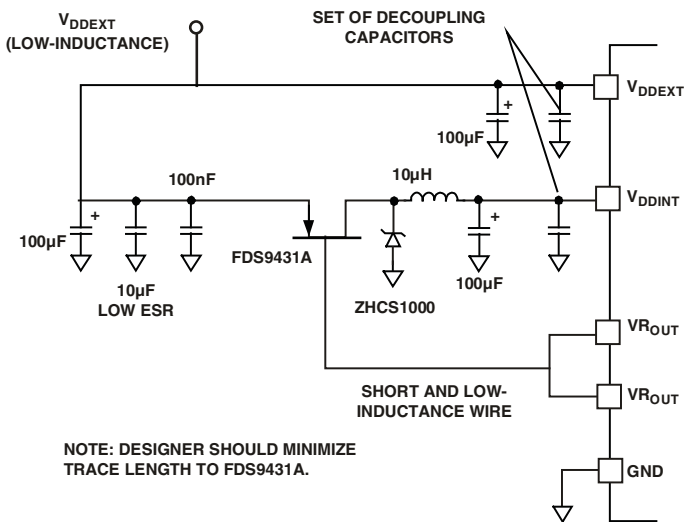


Figure 6. Voltage Regulator Circuit

## CLOCK SIGNALS

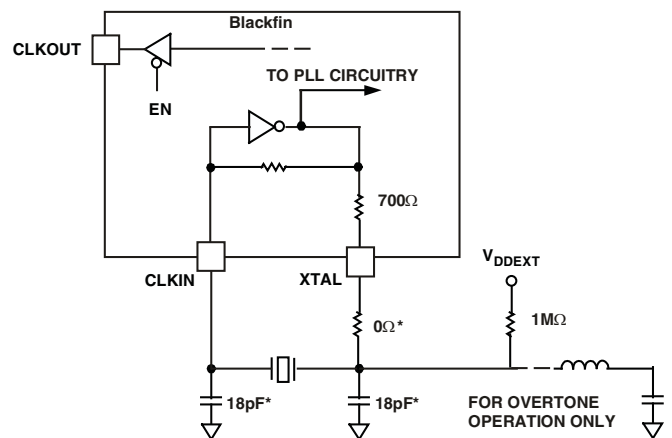
The ADSP-BF539/ADSP-BF539F processors can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

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 processors include an on-chip oscillator circuit, an external crystal can be used. For fundamental frequency operation, use the circuit shown in [Figure 7](#). A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 k $\Omega$  range. Further parallel resistors are typically not recommended. The two capacitors and the series resistor, shown in [Figure 7](#), fine tune the 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 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 7](#).

As shown in [Figure 8 on Page 16](#), 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 $\times$  to 64 $\times$  multiplication factor (bounded by specified minimum and maximum VCO frequencies). The default multiplier is 10 $\times$ , 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.



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

Figure 7. External Crystal Connections

<sup>†</sup> See *Switching Regulator Design Considerations for ADSP-BF533 Blackfin Processors (EE-228)*.

# ADSP-BF539/ADSP-BF539F

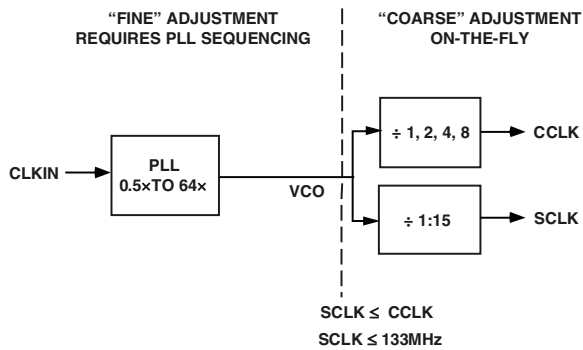


Figure 8. Frequency Modification Methods

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL\_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 7 illustrates typical system clock ratios.

Table 7. 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).

Note that when the SSEL value is changed, it will affect all the peripherals that derive their clock signals from the SCLK signal.

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 8. This programmable core clock capability is useful for fast core frequency modifications.

Table 8. Core Clock Ratios

Signal Name CSEL1–0	Divider Ratio VCO/CCLK	Example Frequency Ratios	
		VCO	CCLK
00	1:1	300	300
01	2:1	300	150
10	4:1	500	125
11	8:1	200	25

## BOOTING MODES

The ADSP-BF539/ADSP-BF539F processors have three mechanisms (listed in Table 9) for automatically loading internal L1 instruction memory after a reset. A fourth mode is provided to execute from external memory, bypassing the boot sequence.

Table 9. Booting Modes

BMODE1–0	Description
00	Execute from 16-bit external memory (bypass boot ROM)
01	Boot from 8-bit or 16-bit flash or boot from on-chip flash (ADSP-BF539F only)
10	Boot from SPI serial master connected to SPI0
11	Boot from SPI serial slave EEPROM/flash (8-, 16-, or 24-bit address range, or Atmel AT45DB041, AT45DB081, or AT45DB161 serial flash) connected to SPI0

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).
- Boot from 8-bit or 16-bit external flash memory – The 8-bit flash boot routine located in boot ROM memory space is set up using asynchronous memory bank 0. For ADSP-BF539F processors, if  $\overline{FCE}$  is connected to  $\overline{AMS0}$ , then the on-chip flash is booted. 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 serial EEPROM/flash (8-, 16-, or 24-bit addressable, or Atmel AT45DB041, AT45DB081, or AT45DB161) connected to SPI0 – The SPI0 port uses the PF2 output pin to select a single SPI EEPROM/flash device, submits a read command and successive address bytes (0x00) until a valid 8-, 16-, or 24-bit, or Atmel addressable device is detected, and begins clocking data into the beginning of the L1 instruction memory.
- Boot from SPI host device connected to SPI0 – 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 in the .LDR image.

For each of the boot modes, a 10-byte header is first read from an external memory device. The header specifies the number of bytes to be transferred and the memory destination address.

Multiple memory blocks can be loaded by any boot sequence. Once all blocks are loaded, program execution commences from the start of L1 instruction SRAM.

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.

To augment the boot modes, a secondary software loader is provided that adds additional booting mechanisms. This secondary loader provides the ability to boot from 16-bit flash memory, fast flash, variable baud rate, and other sources. In all boot modes except bypass, program execution starts from on-chip L1 memory address 0xFFA0 0000.

## INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both user (algorithm/application code) and supervisor (O/S kernel, device drivers, debuggers, ISRs) modes of operation, allowing multiple levels of access to core processor resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/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 supervisor stack pointers.
- Code density enhancements, which include intermixing of 16-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

## DEVELOPMENT TOOLS

Analog Devices supports its processors with a complete line of software and hardware development tools, including integrated development environments (which include CrossCore<sup>®</sup> Embedded Studio and/or VisualDSP++<sup>®</sup>), 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](http://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](http://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<sup>®</sup> 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<sup>®</sup>, which are daughter cards delivering additional specialized functionality, including audio and video processing. For more information visit [www.analog.com](http://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.

# ADSP-BF539/ADSP-BF539F

## 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](http://www.analog.com/ucos3)
- [www.analog.com/ucfs](http://www.analog.com/ucfs)
- [www.analog.com/ucusbdb](http://www.analog.com/ucusbdb)
- [www.analog.com/lwip](http://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](http://www.analog.com) and search on “Blackfin software modules” or “SHARC software modules”.

## Designing an Emulator-Compatible DSP Board (Target)

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

For details on target board design issues including mechanical layout, single processor connections, signal buffering, signal termination, and emulator pod logic, see the Engineer-to-Engineer Note “*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.

## EXAMPLE CONNECTIONS AND LAYOUT CONSIDERATIONS

Figure 9 shows an example circuit connection of the ADSP-BF539/ADSP-BF539F to a MOST network. This diagram is intended as an example, and exact connections and recommended circuit values should be obtained from Analog Devices.

## MXVR BOARD LAYOUT GUIDELINES

### MLF pin

- Capacitors:
  - C1: 0.1  $\mu$ F (PPS type, 2% tolerance recommended)
  - C2: 0.01  $\mu$ F (PPS type, 2% tolerance recommended)
- Resistor:
  - R1: 220  $\Omega$  (1% tolerance)
- The RC network connected to the MLF pin should be located physically close to the MLF pin on the board.
- The RC network should be wired up and connected to the MLF pin using wide traces.
- The capacitors in the RC network should be grounded to MXEGND.
- The RC network should be shielded using MXEGND traces.
- Avoid routing other switching signals near the RC network to avoid crosstalk.

### MXI driven with external clock oscillator IC (recommended)

- MXI should be driven with the clock output of a 49.152 MHz or 45.1584 MHz clock oscillator IC.
- MXO should be left unconnected.
- Avoid routing other switching signals near the oscillator and clock output trace to avoid crosstalk. When not possible, shield traces with ground.

### MXI/MXO with external crystal

- The crystal must be a 49.152 MHz or 45.1584 MHz fundamental mode crystal.
- The crystal and load capacitors should be placed physically close to the MXI and MXO pins on the board.
- The load capacitors should be grounded to MXEGND.
- The crystal and load capacitors should be wired up using wide traces.
- Board trace capacitance on each lead should not be more than 3 pF.
- Trace capacitance plus load capacitance should equal the load capacitance specification for the crystal.
- Avoid routing other switching signals near the crystal and components to avoid crosstalk. When not possible, shield traces and components with ground.

### MXEGND–MXVR crystal oscillator and MXVR PLL ground

- Should be routed with wide traces or as ground plane.
- Should be tied together to other board grounds at only one point on the board.
- Avoid routing other switching signals near to MXEGND to avoid crosstalk.

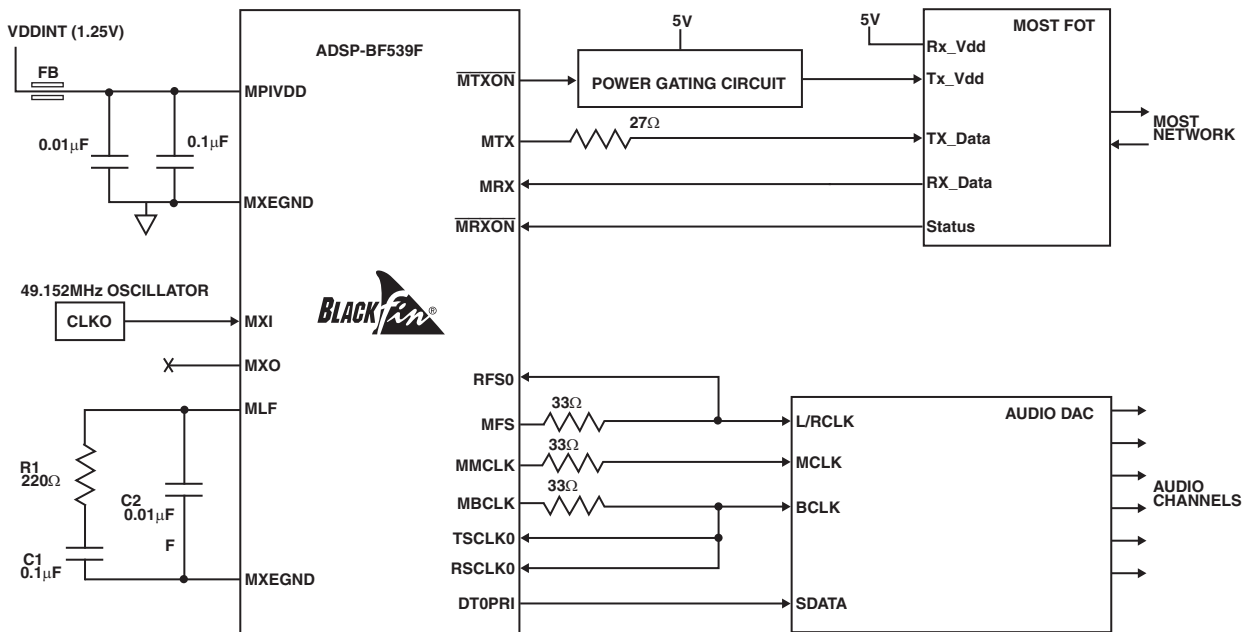


Figure 9. Example Connections of ADSP-BF539/ADSP-BF539F to MOST Network

#### MXEVDD–MXVR crystal oscillator 3.3 V power

- Should be routed with wide traces or as power plane.
- Locally bypass MXEVDD with 0.1 µF and 0.01 µF decoupling capacitors to MXEGND.
- Avoid routing other switching signals near to MXEVDD to avoid crosstalk.

#### MPIVDD–MXVR PLL 1.25 V power

- Should be routed with wide traces or as power plane.
- A ferrite bead should be placed between the 1.25 V  $V_{DDINT}$  power plane and the MPIVDD pin for noise isolation.
- Locally bypass MPIVDD with 0.1 µF and 0.01 µF decoupling capacitors to MXEGND.
- Avoid routing other switching signals near to MPIVDD to avoid crosstalk.

#### Fiber optic transceiver (FOT) connections

- The traces between the ADSP-BF539/ADSP-BF539F processor and the FOT should be kept as short as possible.
- The receive data trace connecting the FOT receive data output pin to the ADSP-BF539/ADSP-BF539F MRX input pin should not have a series termination resistor. The edge rate of the FOT receive data signal driven by the FOT is typically very slow, and further degradation of the edge rate is not desirable.

- The transmit data trace connecting the processor’s MTX output pin to the FOT Transmit Data input pin should have a 27 W series termination resistor placed close to the ADSP-BF539/ADSP-BF539F MTX pin.
- The receive data trace and the transmit data trace between the processor and the FOT should not be routed close to each other in parallel over long distances to avoid crosstalk.

### 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-BF539/ADSP-BF539F 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 website ([www.analog.com](http://www.analog.com))—use site search on “EE-228”.

# ADSP-BF539/ADSP-BF539F

## ADDITIONAL INFORMATION

The following publications that describe the ADSP-BF539/ADSP-BF539F 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-BF539 Blackfin Processor Hardware Reference*
- *ADSP-BF53x/ADSP-BF56x Blackfin Processor Programming Reference*
- *ADSP-BF539 Blackfin Processor Anomaly List*

## RELATED SIGNAL CHAINS

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

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

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

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

## PIN DESCRIPTIONS

ADSP-BF539/ADSP-BF539F processor pin definitions are listed in [Table 10](#).

All pins are three-stated during and immediately after reset, except the 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. If  $\overline{BR}$  is active (whether or not  $\overline{RESET}$  is asserted), the memory pins are also three-stated. All unused I/O pins have their input buffers disabled with the exception of the

pins that need pull-ups or pull-downs, as noted in the table. During hibernate, all outputs are three-stated unless otherwise noted in [Table 10](#).

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

**Table 10. Pin Descriptions**

Pin Name	Type	Description	Driver Type <sup>1</sup>
<i>Memory Interface</i>			
ADDR19–1	O	Address Bus for Async/Sync Access	A
DATA15–0	I/O	Data Bus for Async/Sync Access	A
$\overline{ABE1-0}/\overline{SDQM1-0}$	O	Byte Enables/Data Masks for Async/Sync Access	A
$\overline{BR}$	I	Bus Request (This pin should be pulled high when not used.)	
$\overline{BG}$	O	Bus Grant	A
BGH	O	Bus Grant Hang	A
<i>Asynchronous Memory Control</i>			
$\overline{AMS3-0}$	O	Bank Select	A
ARDY	I	Hardware Ready Control (This pin should always be pulled low when not used.)	
$\overline{AOE}$	O	Output Enable	A
$\overline{ARE}$	O	Read Enable	A
$\overline{AWE}$	O	Write Enable	A
<i>Flash Control</i>			
$\overline{FCE}$	I	Flash Enable (This pin is internally connected to GND on the ADSP-BF539.)	
$\overline{FRESET}$	I	Flash Reset (This pin is internally connected to GND on the ADSP-BF539.)	
<i>Synchronous Memory Control</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 (This pin must be pulled low through a 10 k $\Omega$ resistor if hibernate state is used and SDRAM contents need to be preserved during hibernate.)	A
CLKOUT	O	Clock Output	B
SA10	O	A10 Pin	A
$\overline{SMS}$	O	Bank Select	A
<i>Timers</i>			
TMR0	I/O	Timer 0	C
TMR1/PPI_FS1	I/O	Timer 1/PPI Frame Sync1	C
TMR2/PPI_FS2	I/O	Timer 2/PPI Frame Sync2	C

# ADSP-BF539/ADSP-BF539F

Table 10. Pin Descriptions (Continued)

Pin Name	Type	Description	Driver Type <sup>1</sup>
<i>Parallel Peripheral Interface Port/GPIO</i>			
PF0/ $\overline{\text{SPI0SS}}$	I/O	Programmable Flag 0/SPI0 Slave Select Input	C
PF1/ $\overline{\text{SPI0SEL1}}$ /TACLK	I/O	Programmable Flag 1/SPI0 Slave Select Enable 1/Timer Alternate Clock	C
PF2/ $\overline{\text{SPI0SEL2}}$	I/O	Programmable Flag 2/SPI0 Slave Select Enable 2	C
PF3/ $\overline{\text{SPI0SEL3}}$ /PPI_FS3	I/O	Programmable Flag 3/SPI0 Slave Select Enable 3/PPI Frame Sync 3	C
PF4/ $\overline{\text{SPI0SEL4}}$ /PPI15	I/O	Programmable Flag 4/SPI0 Slave Select Enable 4/PPI 15	C
PF5/ $\overline{\text{SPI0SEL5}}$ /PPI14	I/O	Programmable Flag 5/SPI0 Slave Select Enable 5/PPI 14	C
PF6/ $\overline{\text{SPI0SEL6}}$ /PPI13	I/O	Programmable Flag 6/SPI0 Slave Select Enable 6/PPI 13	C
PF7/ $\overline{\text{SPI0SEL7}}$ /PPI12	I/O	Programmable Flag 7/SPI0 Slave Select Enable 7/PPI 12	C
PF8/PPI11	I/O	Programmable Flag 8/PPI 11	C
PF9/PPI10	I/O	Programmable Flag 9/PPI 10	C
PF10/PPI9	I/O	Programmable Flag 10/PPI 9	C
PF11/PPI8	I/O	Programmable Flag 11/PPI 8	C
PF12/PPI7	I/O	Programmable Flag 12/PPI 7	C
PF13/PPI6	I/O	Programmable Flag 13/PPI 6	C
PF14/PPI5	I/O	Programmable Flag 14/PPI 5	C
PF15/PPI4	I/O	Programmable Flag 15/PPI 4	C
PPI3–0	I/O	PPI3–0	C
PPI_CLK/TMRCLK	I	PPI Clock/External Timer Reference	
<i>Controller Area Network</i>			
CANTX/PC0	I/O 5 V	CAN Transmit/GPIO	C
CANRX/PC1	I/OD 5 V	CAN Receive/GPIO	C <sup>2</sup>
<i>Media Transceiver (MXVR)/General-Purpose I/O</i>			
MTX/PC5	I/O	MXVR Transmit Data/GPIO	C
$\overline{\text{MTXON}}$ /PC9	I/O	MXVR Transmit FOT On/GPIO	C
MRX/PC4	I/OD 5 V	MXVR Receive Data/GPIO (This pin should be pulled low when not used.)	C <sup>2</sup>
$\overline{\text{MRXON}}$	I 5 V	MXVR FOT Receive On (This pin should be pulled high when not used.)	C
MXI	I	MXVR Crystal Input (This pin should be pulled low when not used.)	
MXO	O	MXVR Crystal Output (This pin should be left unconnected when not used.)	
MLF	A I/O	MXVR Loop Filter (This pin should be pulled low when not used.)	
MMCLK/PC6	I/O	MXVR Master Clock/GPIO	C
MBCLK/PC7	I/O	MXVR Bit Clock/GPIO	C
MFS/PC8	I/O	MXVR Frame Sync/GPIO	C
GP	I	GPIO PC4–9 Enable (This pin should be pulled low when MXVR is used.)	
<i>2-Wire Interface Ports</i>			
These pins are open-drain and require a pull-up resistor. See version 2.1 of the I <sup>2</sup> C specification for proper resistor values.			
SDA0	I/O 5 V	TWI0 Serial Data	E
SCL0	I/O 5 V	TWI0 Serial Clock	E
SDA1	I/O 5 V	TWI1 Serial Data	E
SCL1	I/O 5 V	TWI1 Serial Clock	E

**Table 10. Pin Descriptions (Continued)**

Pin Name	Type	Description	Driver Type <sup>1</sup>
<i>Serial Port0</i>			
RSCLK0	I/O	SPORT0 Receive Serial Clock	D
RFS0	I/O	SPORT0 Receive Frame Sync	C
DR0PRI	I	SPORT0 Receive Data Primary	
DR0SEC	I	SPORT0 Receive Data Secondary	
TSCLK0	I/O	SPORT0 Transmit Serial Clock	D
TFS0	I/O	SPORT0 Transmit Frame Sync	C
DT0PRI	O	SPORT0 Transmit Data Primary	C
DT0SEC	O	SPORT0 Transmit Data Secondary	C
<i>Serial Port1</i>			
RSCLK1	I/O	SPORT1 Receive Serial Clock	D
RFS1	I/O	SPORT1 Receive Frame Sync	C
DR1PRI	I	SPORT1 Receive Data Primary	
DR1SEC	I	SPORT1 Receive Data Secondary	
TSCLK1	I/O	SPORT1 Transmit Serial Clock	D
TFS1	I/O	SPORT1 Transmit Frame Sync	C
DT1PRI	O	SPORT1 Transmit Data Primary	C
DT1SEC	O	SPORT1 Transmit Data Secondary	C
<i>Serial Port2</i>			
RSCLK2/PE0	I/O	SPORT2 Receive Serial Clock/GPIO	D
RFS2/PE1	I/O	SPORT2 Receive Frame Sync/GPIO	C
DR2PRI/PE2	I/O	SPORT2 Receive Data Primary/GPIO	C
DR2SEC/PE3	I/O	SPORT2 Receive Data Secondary/GPIO	C
TSCLK2/PE4	I/O	SPORT2 Transmit Serial Clock/GPIO	D
TFS2/PE5	I/O	SPORT2 Transmit Frame Sync/GPIO	C
DT2PRI /PE6	I/O	SPORT2 Transmit Data Primary/GPIO	C
DT2SEC/PE7	I/O	SPORT2 Transmit Data Secondary/GPIO	C
<i>Serial Port3</i>			
RSCLK3/PE8	I/O	SPORT3 Receive Serial Clock/GPIO	D
RFS3/PE9	I/O	SPORT3 Receive Frame Sync/GPIO	C
DR3PRI/PE10	I/O	SPORT3 Receive Data Primary/GPIO	C
DR3SEC/PE11	I/O	SPORT3 Receive Data Secondary/GPIO	C
TSCLK3/PE12	I/O	SPORT3 Transmit Serial Clock/GPIO	D
TFS3/PE13	I/O	SPORT3 Transmit Frame Sync/GPIO	C
DT3PRI /PE14	I/O	SPORT3 Transmit Data Primary/GPIO	C
DT3SEC/PE15	I/O	SPORT3 Transmit Data Secondary/GPIO	C

# ADSP-BF539/ADSP-BF539F

Table 10. Pin Descriptions (Continued)

Pin Name	Type	Description	Driver Type <sup>1</sup>
<i>SPI0 Port</i>			
MOSI0	I/O	SPI0 Master Out Slave In	C
MISO0	I/O	SPI0 Master In Slave Out (This pin should always be pulled high through a 4.7 kΩ resistor if booting via the SPI port.)	C
SCK0	I/O	SPI0 Clock	D
<i>SPI1 Port</i>			
MOSI1/PD0	I/O	SPI1 Master Out Slave In/GPIO	C
MISO1/PD1	I/O	SPI1 Master In Slave Out/GPIO	C
SCK1/PD2	I/O	SPI1 Clock/GPIO	D
$\overline{\text{SPI1SS}}$ /PD3	I/O	SPI1 Slave Select Input/GPIO	D
$\overline{\text{SPI1SEL1}}$ /PD4	I/O	SPI1 Slave Select Enable/GPIO	D
<i>SPI2 Port</i>			
MOSI2 /PD5	I/O	SPI2 Master Out Slave In/GPIO	C
MISO2/PD6	I/O	SPI2 Master In Slave Out/GPIO	C
SCK2/PD7	I/O	SPI2 Clock/GPIO	D
$\overline{\text{SPI2SS}}$ /PD8	I/O	SPI2 Slave Select Input/GPIO	D
$\overline{\text{SPI2SEL1}}$ /PD9	I/O	SPI2 Slave Select Enable/GPIO	D
<i>UART0 Port</i>			
RX0	I	UART Receive	
TX0	O	UART Transmit	C
<i>UART1 Port</i>			
RX1/PD10	I/O	UART1 Receive/GPIO	D
TX1/PD11	I/O	UART1 Transmit/GPIO	D
<i>UART2 Port</i>			
RX2 /PD12	I/O	UART2 Receive/GPIO	D
TX2/PD13	I/O	UART2 Transmit/GPIO	D
<i>Real-Time Clock</i>			
RTXI	I	RTC Crystal Input (This pin should be pulled low when not used.)	
RTXO	O	RTC Crystal Output (Does not three-state in hibernate.)	
<i>JTAG Port</i>			
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 the JTAG port will not be used.)	
$\overline{\text{EMU}}$	O	Emulation Output	C
<i>Clock</i>			
CLKIN	I	Clock/Crystal Input	
XTAL	O	Crystal Output	

**Table 10. Pin Descriptions (Continued)**

Pin Name	Type	Description	Driver Type <sup>1</sup>
<i>Mode Controls</i>			
$\overline{\text{RESET}}$	I	Reset	
$\overline{\text{NMI}}$	I	Nonmaskable Interrupt (This pin should be pulled high when not used.)	
BMODE1–0	I	Boot Mode Strap (These pins must be pulled to the state required for the desired boot mode.)	
<i>Voltage Regulator</i>			
VR0UT1–0	O	External FET Drive 0 (These pins should be left unconnected when not used.)	
<i>Supplies</i>			
V <sub>DDEXT</sub>	P	I/O Power Supply	
V <sub>DDINT</sub>	P	Internal Power Supply	
V <sub>DDRTC</sub>	P	Real-Time Clock Power Supply (This pin should be connected to V <sub>DDEXT</sub> when not used and should remain powered at all times.)	
MPIVDD	P	MXVR Internal Power Supply	
MXEVDD	P	MXVR External Power Supply	
MXEGND	G	MXVR Ground	
GND	G	Ground	

<sup>1</sup> Refer to [Figure 34 on Page 50](#) to [Figure 43 on Page 51](#).

<sup>2</sup> This pin is 5 V-tolerant when configured as an input and an open-drain when configured as an output; therefore, only the VOL curves in [Figure 38 on Page 50](#) and [Figure 39 on Page 51](#) and the fall time curves in [Figure 51 on Page 53](#) and [Figure 52 on Page 53](#) apply when configured as an output.