



Technical notes on using Analog Devices DSPs, processors and development tools  
Visit our Web resources <http://www.analog.com/ee-notes> and <http://www.analog.com/processors> or  
e-mail [processor.support@analog.com](mailto:processor.support@analog.com) or [processor.tools.support@analog.com](mailto:processor.tools.support@analog.com) for technical support.

## Connecting Blackfin® Processors to the AD7656 SAR ADC

Contributed by Phil Burkert and Eugene Trosman

Rev 1 – March 27, 2008

### Introduction

This EE-Note discusses the process for developing the hardware and software required to support interfacing an ADSP-BF5xx Blackfin® processor with the AD7656 SAR ADC. The ADSP-BF537 EZ-KIT Lite® evaluation system was used for demonstration purposes; however, the same hardware and general software concepts may be applied to other Blackfin processors.

The following elements comprise the ADSP-BF537/AD7656 development environment:

- AD7656 Evaluation Module (EVM)
- ADSP-BF537 EZ-KIT Lite Evaluation System (part number ADZS-BF537-EZLITE)
- Custom Interface Board (CIB)
- ADSP-BF537/AD7656 system software, developed using VisualDSP++ 4.5

### AD7656 Evaluation Module (EVM)

The AD7656 EVM provides a standalone means for AD7656 testing and evaluation using the EVAL-CONTROL-BOARD, which interfaces to a PC printer port. The PC runs EVAL-CONTROL-BOARD software. The ADSP-BF537/AD7656 platform adds the CIB, ADZS-BF537 EZLITE, and system software to provide Blackfin embedded processor evaluation capabilities using VisualDSP++ development tools.

The AD7656 operates in either hardware (HW) or software (SW) mode. HW mode implies device operation as configured via several AD7656 inputs. The input levels are set using AD7656 EVM jumpers. AD7656 configuration options include internal/external reference selection, internal reference buffer enable, etc., as defined by the AD7656 data sheet. SW mode involves AD7656 configuration via an AD7656 internal register write operation. SW mode provides configuration flexibility and allows for control via an embedded processor. The AD7656 EVM exclusively supports HW mode evaluation, but it can be modified to support SW mode evaluation via the ADSP-BF537 processor.

To enable a software mode, jumpers on the AD7656 EVM must be properly configured – please refer to the AD7656 EVM manual. In addition, the CONVST<sub>B</sub> and CONVST<sub>C</sub> inputs should be tied low, since only CONVST<sub>A</sub> input is used to initiate a simultaneous conversion on the combination of channels selected via the AD7656 internal configuration register. Also, the following modifications are required for the AD7656 EVM:

- The bus transceiver (U2) used on the EVM is a Pericom 74FCT162245 (a 5V only device). It must be replaced with a 3.3V part (for example, a Fairchild 74LVTH16245MEA) to accommodate the maximum voltage (3.6V) of the ADSP-BF537 EBIU. This will also allow the  $V_{drive}$  rail of the EVM to be connected directly to a  $V_{cc}$  supply on the EZ-KIT Lite development system.
- Pin 24 of the bi-directional bus transceiver (U2) must be disconnected (lifted) from  $V_{drive}$  and connected to a buffered  $\overline{WR}$  signal (pin 4 of U13) to allow for proper direction control when writing into the AD7656 internal configuration register.

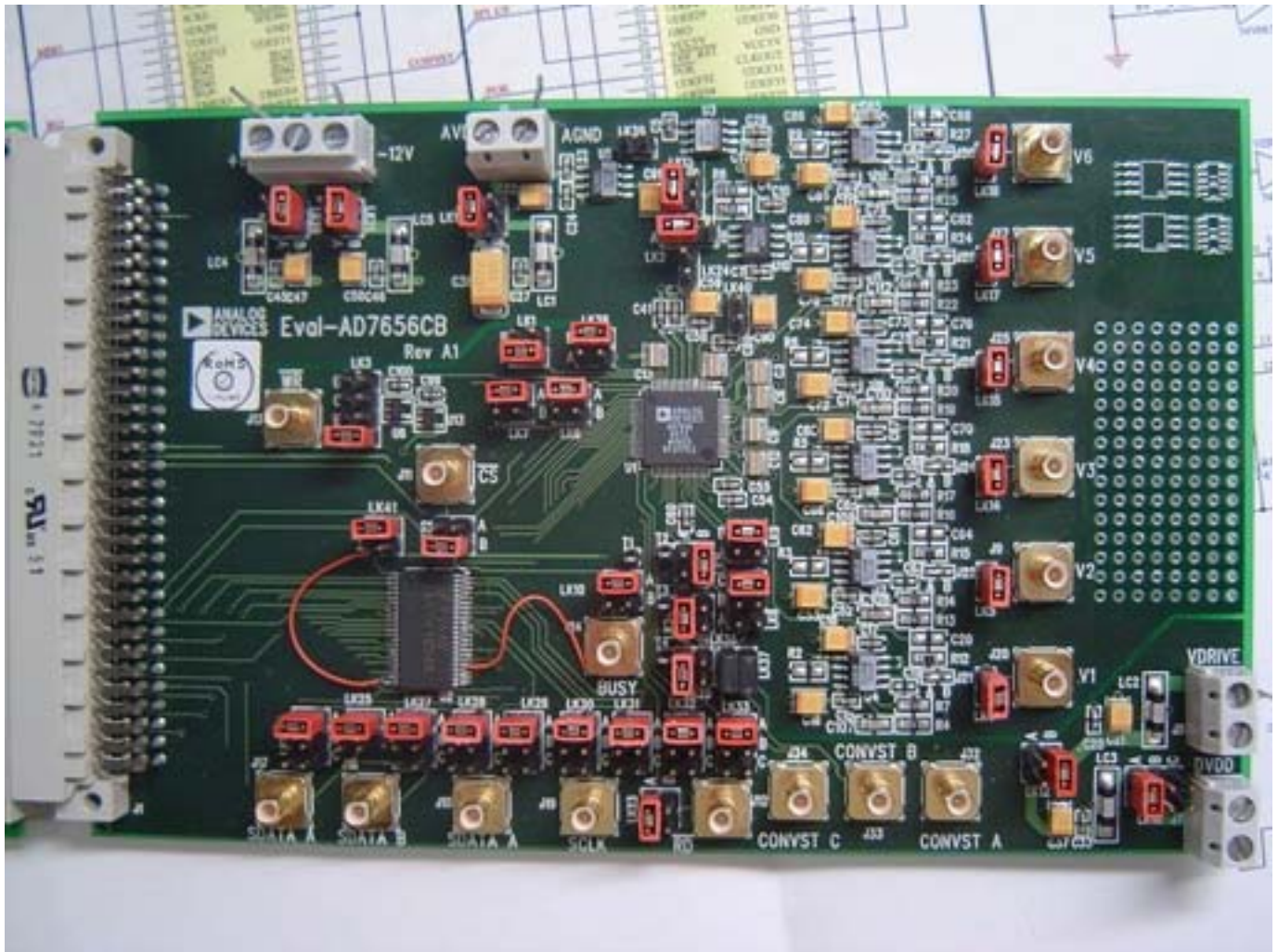


Figure 1. Jumper settings and modifications to enable SW mode on the AD7656 EVM

## ADZS-BF537 EZLITE

The ADZS-BF537 EZLITE is used to evaluate the ADSP-BF537 Blackfin processor. It incorporates a backside expansion header (U-connector) which enables access to all ADSP-BF537 processor pins. The Custom Interface Board (CIB) connects this expansion header to the AD7656 EVM. This connection provides processor access to the AD7656 internal register and data buffers via the ADSP-BF537 asynchronous external memory interface.

## Custom Interface Board (CIB)

The Custom Interface Board brings the data bus and control pins from the ADZS-BF537 EZLITE U-connector to a 96-pin connector that can be plugged directly into the AD7656 EVM. A serial SPI port option is available as well. In addition, the CIB provides buffers for the CONVST, /CS, and /RD signals (to improve noise immunity) and options for connecting /CS,  $V_{drive}$ , +5VA, AGND, and /RESET via the on-board jumpers.

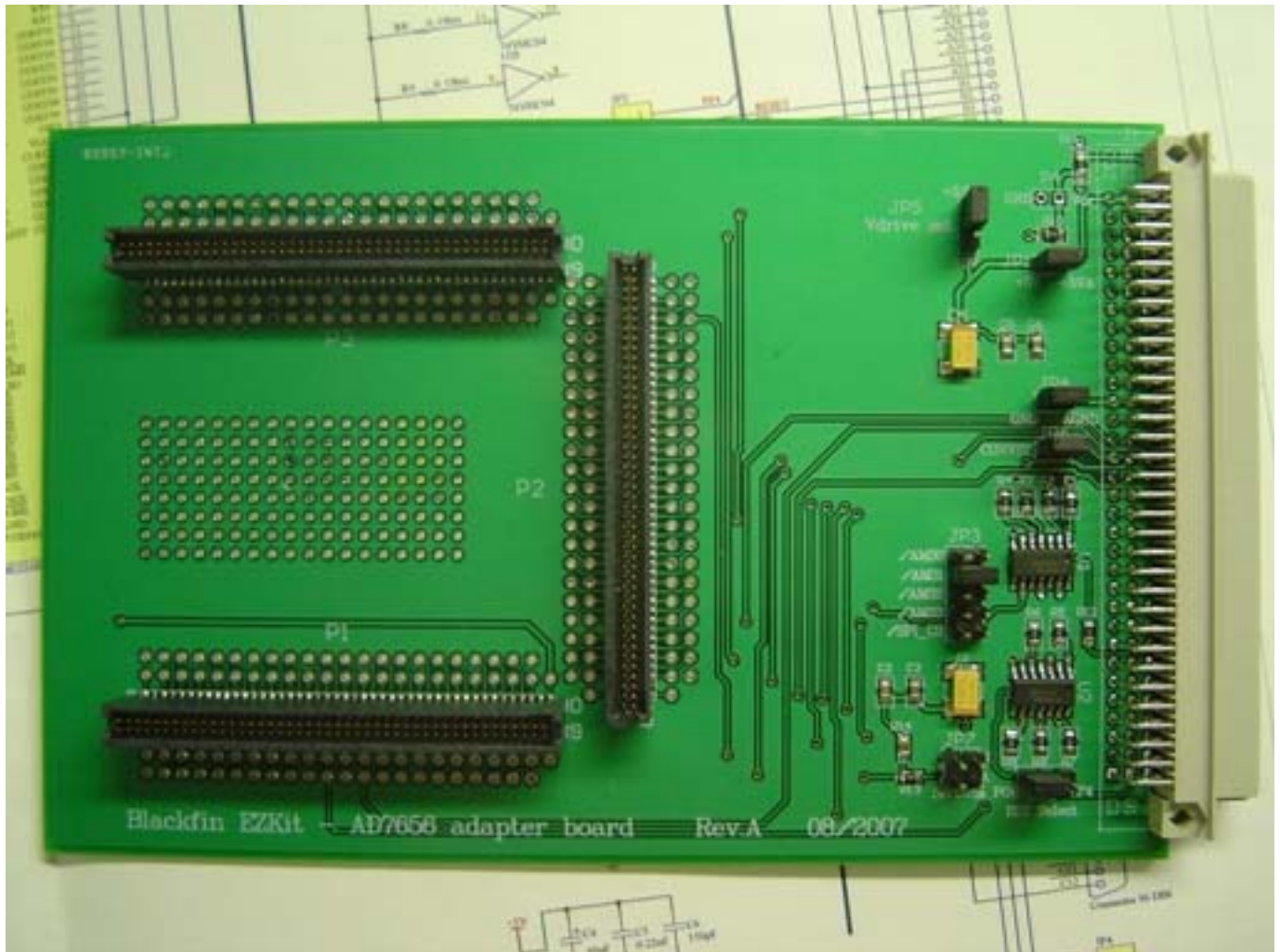


Figure 2. Custom Interface Board (CIB)

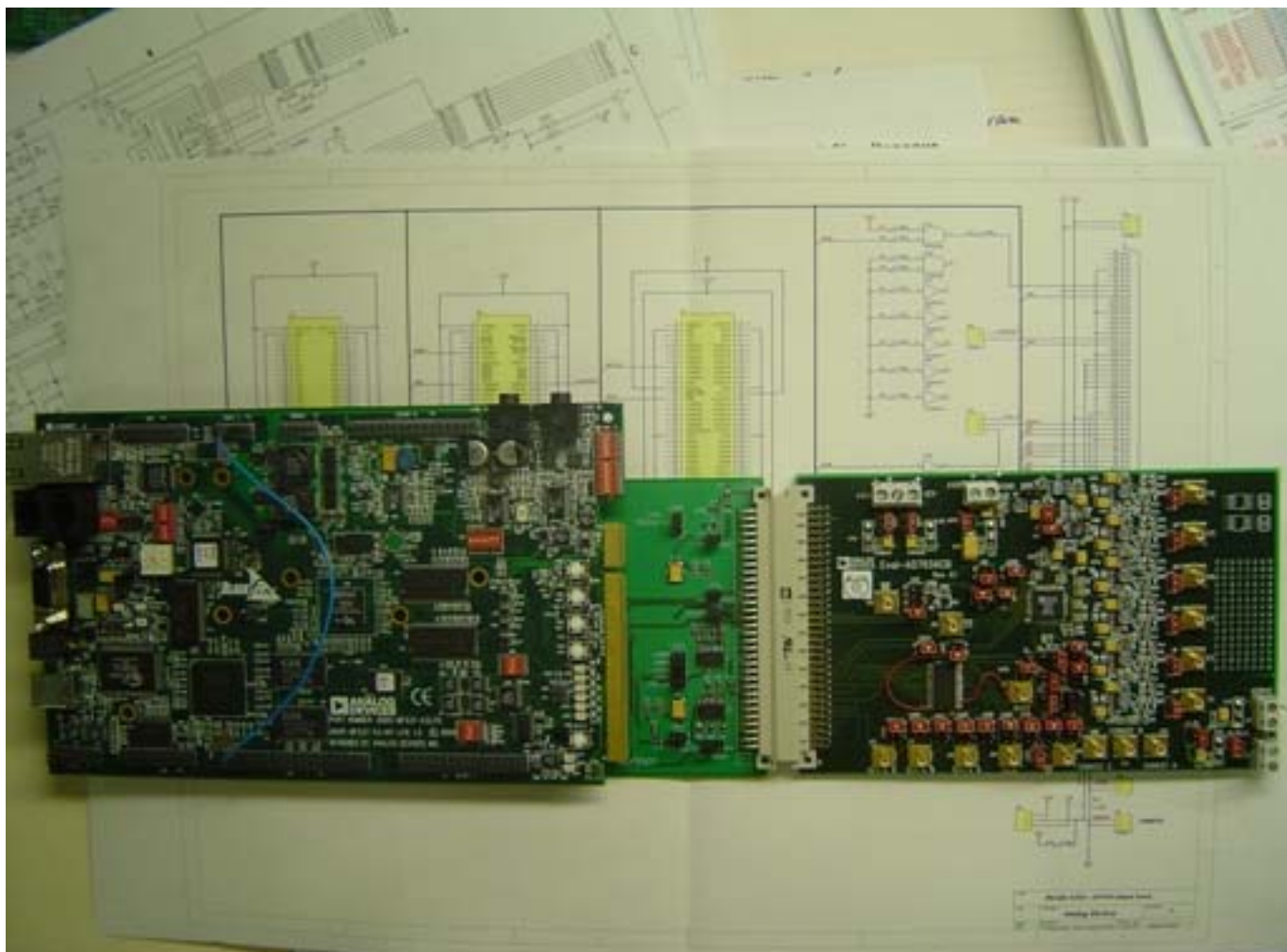


Figure 3. AD7656 ADC EVM connected to the ADZS-BF537 EZLITE via the Custom Interface Board (CIB)

## VisualDSP++-Based ADSP-BF537/AD7656 System Software

The ADSP-BF537/AD7656 system software performs the following tasks:

- Resets the AD7656
- Configures the AD7656 mode of operation
- Services push-buttons for mode switching
- Initiates AD7656 conversion
- Stores samples
- Computes FFT for AD7656 channel one data
- Provides ONE SHOT and BLOCK operation modes

## Modes of Operation

**ONE\_SHOT:** Pressing the PB1 push-button on the ADSP-BF537 EZ-KIT Lite evaluation board enables the conversion and memory storage of all six AD7656 channels. The ADSP-BF537 TMR2 output drives the AD7656 CONVSTA input, initiating AD7656 analog-to-digital conversion.

Upon completion of the conversion, the AD7656 BUSY signal transitions, which drives an ADSP-BF537 GPIO interrupt request (AD7656 BUSY interrupt). The BUSY interrupt service handler (BUSY ISR) initiates a DMA transfer. Six AD7656 channels are read, and the data is stored to SDRAM via a memory DMA transfer from external asynchronous memory to SDRAM. The VisualDSP++ development tools allow you to view memory in a variety of data formats to verify that the data is being stored properly.

**BLOCK\_MODE:** Pressing the PB4 push-button on the ADSP-BF537 EZ-KIT Lite evaluation board enables BLOCK\_MODE operation. The ADSP-BF537 TMR2 output drives the AD7656 CONVSTA input, initiating the AD7656 analog-to-digital conversion of the six AD7656 channels. Upon completion of the conversion, the AD7656 BUSY signal transitions, which drives an ADSP-BF537 GPIO interrupt request (BUSY interrupt). The BUSY interrupt service handler (BUSY ISR) initiates a DMA transfer. The BUSY signal transitions 256 times so that a block of 256 words is read from each of the six AD7656 channels, and the data is stored to SDRAM via 256 memory DMA transfers from the external asynchronous memory to SDRAM. Upon completion (256 \* 6 words transferred), the BUSY ISR sets the RunFFT flag. When RunFFT is set, the application extracts the AD7656 channel one data from memory and calls an optimized assembly FFT. The FFT completes and clears the RunFFT flag. The FFT results can be displayed using the VisualDSP++ memory view or plot window features. For convenience, the associated .ZIP file<sup>[1]</sup> includes a pre-configured VisualDSP++ workspace file, <Project\_Directory>\debug\AD7656\_ADSP-BF537.vdw, which can be used to automatically configure the plot window.



The system software included in the file associated with this EE-Note is specific to the ADSP-BF537 Blackfin processor; however, it can be easily modified to support all Blackfin processors. Blackfin EZ-KIT Lite evaluation boards feature the expansion header for use with the Custom Interface Board (CIB). This software is not optimized for performance, as it is intended as a starting point for a variety of potential applications. Additional system-level details provided in the software are described in [Appendix A: System Software Function Descriptions](#), and specific programming details appear in the code's comments and in the DesignNotes.doc file included in the .ZIP file associated with this EE-Note.

## Summary

This EE-Note demonstrates how to interface the AD7656 SAR ADC evaluation board to the ADSP-BF537 EZ-KIT Lite evaluation system. It describes the required hardware modifications and provides software developed using the VisualDSP++ 4.5 development tools to control the ADC, acquire samples, and perform an FFT function on a single input channel.

## Appendix A: System Software Function Descriptions

### File main.c

This module initializes control variables and sets the clock rates for the processor. The system clock rate is set to 100 MHz, based on a core clock rate of 500 MHz. The core clock rate is configured as part of the VisualDSP++ project options, so the auto-generated `AD7656_ADSP-BF537_basiccrt.s` assembly file contains the source code to set the clock. The following initialization functions are then called to initialize the system:

```
Init_Flags();  
Init_EBIU();  
Init_AD7656();  
Init_Interrupts();  
Init_Timers();
```

The `twidfttrad2_fr16` function (a VisualDSP++ math library function) is called to generate FFT twiddle factors.

The `rfft_fr16` function (a VisualDSP++ math library function) is called to compute the FFT if the `RunFFT` flag is set, and the `RunFFT` flag is subsequently cleared after the FFT is run.

### File Initialization.c

This file contains the initialization routines called by the `main()` routine - `Init_Flags()`, `Init_EBIU()`, `Init_AD7656()`, `Init_Interrupts()`, and `Init_Timers()`.

#### *Init\_Flags()*

This module enables the `TMR2`, `TMR3`, and `TMRCLK` pin functions. It also enables the GPIO input buffers for the push-button interrupts and properly configures the GPIO direction, polarity, and sensitivity to support interrupts on these GPIO pins.

#### *Init\_Timers()*

This module initializes timer 3 in `PWM_OUT` mode to output a 10 MHz square wave based on the system clock of 100 MHz. The `TMR3` output is jumpered to the ADSP-BF537 `TMRCLK` input and will serve as a time base for deriving the desired AD7656 sample rate of 61440 Hz. Timer 2 is initialized in `PWM_OUT` mode for 61728 Hz. This is as close to the desired 61440 Hz as possible, using the 10 MHz time base. This arrangement illustrates realizable frequency granularity versus reference frequency via Blackfin timer configuration. The `TMR2` output drives the AD7656 `CONVSTA` input. A reference of higher frequency will provide greater frequency resolution. For instance, the user can program timer 2 using the 100 MHz system clock as a reference to realize finer AD7656 `CONVSTA` granularity and forego use of the jumper wire. Because Blackfin timers are ultimately derived from the PLL, a precision oscillator may be selected to ensure AD7656 performance purity. This oscillator would drive the ADSP-BF537 `TMRCLK` input.

#### *Init\_Interrupts()*

This module assigns core interrupt ID numbers, installs interrupt handlers, and enables the timer 2 interrupt (optional), the memory DMA channel 0 interrupt, and the port F interrupts A and B.

### *Init\_EBIU()*

This module configures and enables the external bus interface.

*/AMS0* is the ADSP-BF537 EZ-KIT Lite board's flash memory select and is configured for default timing.

*/AMS1* drives the AD7656 device select input and is set up for ten SCLK cycles for read/write accesses and three SCLK cycles for hold time.

The SDRAM controller is configured to use the default EZ-KIT Lite board's SDRAM register configuration.

### *Init\_AD7656()*

This module resets the AD7656 via the ADSP-BF537 PF4 GPIO pin. It also initializes the AD7656 configuration register as follows:

- Convert all channels upon CONVSTA transition
- Set range for  $\pm 2 * V_{REF}$
- Select internal reference
- Enable reference buffer

### **File ISRs.c**

This file contains the interrupt handlers for timer 2, the GPIO port F interrupts, and the memory DMA.

#### *EX\_INTERRUPT\_HANDLER(Timer2\_ISR)*

This handler clears the interrupt status and increments the `Int2Cnt` variable, which supports a cycle count operation to verify that a 256-point FFT will complete before the next transition of the CONVSTA signal (TMR2). This is convenience code written to aid in development. This interrupt is disabled for release code.

#### *EX\_INTERRUPT\_HANDLER(Mem0StreamDMA\_ISR)*

This handler has no processing associated with it. It simply returns to the application.

#### *EX\_INTERRUPT\_HANDLER(PORTF\_IntA\_ISR)*

This handler clears the interrupt status and sets the operation mode.

#### *EX\_INTERRUPT\_HANDLER(PORTF\_IntB\_ISR)*

This handler clears the interrupt status, executes mode-dependent memory DMA operations, and manages the `RunFFT` flag.



The associated .ZIP file also contains a VisualDSP++ 4.5 project file, a schematic, a Bill of Materials (BOM), and Gerber files for the CIB, as well as two text files (`readme.txt` and `DesignNotes.doc`) which should be read before attempting to run the software, as they contain essential general system design notes.

## References

- [1] *VisualDSP++ Project Code*. Associated .ZIP file with *Connecting Blackfin Processors to the AD7656 SAR ADC (EE-321)*. March 2008. Analog Devices, Inc.
- [2] *ADSP-BF534/ADSP-BF536/ADSP-BF537 Blackfin Embedded Processor Data Sheet*. Rev E, March 2008. Analog Devices, Inc.
- [3] *250kSPS, 6-Channel, Simultaneous Sampling, Bipolar 16-/14-/12-Bit ADC Data Sheet*. Rev A, May 2006. Analog Devices, Inc.
- [4] *EVAL-AD7656: Evaluation Board for 16-bit, 6-Channel Simultaneous Sampling ADC Data Sheet* Rev 0, September 2007. Analog Devices, Inc.
- [5] *ADSP-BF537Blackfin Processor Hardware Reference*. Rev 3.0, December 2007. Analog Devices, Inc.
- [6] *ADSP-BF537Blackfin EZ-KIT Lite Evaluation System Manual* Rev 2.2, May 2007. Analog Devices, Inc.

## Document History

Revision	Description
<i>Rev 1 – March 27, 2008 by Eugene Trosman and Phil Burkert</i>	Initial Release