Engineer-to-Engineer Note EE-395

ANALOG DEVICES

Technical notes on using Analog Devices products, 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 or processor.tools.support@analog.com for technical support.

Interfacing the AD7689 ADC to ADSP-BF70x Blackfin+^{TM} Processors

Contributed by Baruah, Trinayan and Tarkoff, Joe

Rev 1 – December 15, 2016

Introduction

This application note describes how to interface an Analog Devices AD7689 ADC to an ADSP-BF70x Blackfin+^{TM} processor using the SPI port. The AD7689 ADC is an 8-channel, 16-bit, charge redistribution Successive Approximation Register (SAR) ADC. The ADC has a sampling rate of 250k samples/second (kSPS) and is suitable for a variety of applications including battery power measurement, multichannel system monitoring, and medical equipment (e.g., ECG, EKG, etc.). Furnished with this application note is *example code*[1] demonstrating how to configure the Blackfin+ processor's SPI port to be enabled to configure and read conversion results from the ADC.

AD7689 ADC Overview

The AD7689 ADC is an 8-channel converter that operates at a rate of 250 kSPS and can power down between conversions, making it ideal for battery powered applications which involve data acquisition from multiple channels. It also provides other features like a temperature sensor, selectable one-pole filtering, and a channel sequencer, which enables convenient scanning of channels in a repeated fashion. Each feature can be individually configured by writing to a 14-bit ADC register using SPI communication. The ADC also features an on-board conversion clock, thus it does not require an external serial clock for conversion. For fast hosts, the ADC supports reading and writing during the conversion process, whereas the reads and writes can be held off until after conversion for slower host devices. Both modes with supporting sample code are discussed in the sections that follow.

ADSP BF70x Processor Overview

The ADSP-BF70x processor is a member of the Blackfin+ family of embedded processors. It features a dual MAC and SIMD capabilities for computation power and a clean and orthogonal RISC-like microprocessor ISA. The ADSP-BF70x processor offer performance up to 400 MHz with very low static power consumption. The low power and low voltage design methodology in these processors allow them to be used for a wide variety of markets including, but not limited to, automotive systems, instrumentation, industrial automation, video and image analysis, and motor control applications. It maintains full ISA compatibility with previous Blackfin processors.

Pin Multiplexing on ADSP-BF70x Processor

The ADSP-BF70x processor provides a pin multiplexing option, where a single pin can be used for multiple purposes. This reduces the number of pins required on the chip, thereby making the design more compact. The Pin Multiplexing tool in CCES can be used to obtain the desired pin configuration for interfacing to the AD7689 ADC.

The ADSP-BF70x processor features three SPI ports - SPI0, SPI1 and SPI2 - which are accessible on the processor's general-purpose I/O ports A, B, and C (see the pin multiplexing table in the ***processor data sheet***[2] for details). The two registers of interest when programming the port multiplexing scheme are the port multiplexing (PORTx_MUX) and port function enable (PORTx_FER) registers. While the MUX registers select which function is to be used by specific pins, each pin is individually controlled in the corresponding FER register to enable the peripheral function specified in the MUX register (see the Ports chapter in the ***processor hardware reference manual***[3] for details).

Interfacing the AD7689 ADC to the processor requires that all the critical SPI signals (SPI_CLK, SPI_MISO, and SPI_MOSI) be properly enabled.

SPI Port

SPI is a synchronous serial link that supports communicating with multiple SPI devices. In its basic form, it is a synchronous four-wire interface that consists of two data pins supporting full-duplex operation, a device-select pin, and a gated clock pin. Figure 1 depicts the SPI connections between the ADSP-BF70x processor and the AD7689 ADC.

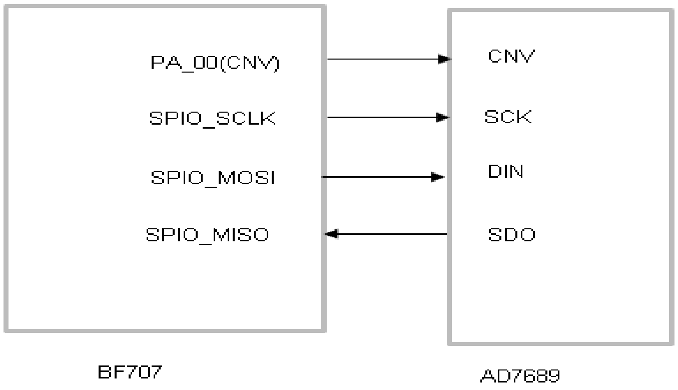

Figure 1. ADSP-BF70x/AD7689 SPI Interface

The SPI interface supports programmable baud rates and configurable clock polarity and phase. SPI also supports a flow control mechanism that can be used to ensure that only the desired amount of data is sent or received. Dedicated DMA channels allow data transfers to occur without incurring core cycles of overhead, making it suitable for using in a data acquisition system such that data can be transferred while the core is performing other tasks. An advanced feature, 2D DMA, can be used to sort data into different buffers on-the-fly.

Using the SPI in DMA mode versus non-DMA mode carries with it a unique set of requirements. Care must be taken to ensure correct register settings when using the SPI in Master mode versus Slave mode, as many of the control bits are valid only when used in a specific mode. See the SPI chapter in the ***hardware reference manual*** for more details.

SPI Configuration Settings

Before writing to or reading from the ADC, the SPI port must be correctly configured via the SPI_CTL, SPI_CLK, SPI_TXCTL and SPI_RXCTL registers. For this interface to function properly, the ADSP-BF70x processor SPI port must be configured as a master supporting 16-bit MSB-first data with an active high clock that toggles from the middle of the bit, per the ***ADC datasheet***[4].

The transmit and receive channels are enabled. The receive channel data is configured to overwrite the buffer with new content if data arrives when the SPI_RFIFO buffer is full. The SPI_CLK configuration depends on whether the ADC is in RAC or RDC mode.

Configuring the AD7689 ADC

When using the AD7689 ADC for conversion, the standard four-wire SPI mode is used. In this configuration, the ADSP-BF70x SPI port is the master, the ADC is the slave, and the data transfer is bidirectional. The ADSP-BF70x processor configures the CFG register on the AD7689 ADC, which sets its mode of operation. After configuration is completed, the ADC begins sending the 16-bit data to the processor via the SPI_MISO pin.

The processor initially provides an active-high CNV signal to the ADC to indicate that the ADC should start conversion. CNV must be kept high beyond the end of the conversion period to avoid the generation of a busy indicator. After CNV reaches the End of Conversion time (t_{conv}), it is brought low to initiate the data transfer. Table 1 lists the ADC timing parameter symbols referenced here and their meanings.

Symbol	Meaning
t_{CONV}	Conversion Time
t_{CYC}	Time Between Conversions
t_{DATA}	Data Write/Read During Conversion
t_{ACQ}	Acquisition Time
t_{SCK}	SPI Clock Time

Table 1 Symbol Table

Since the ADC is not configured at startup, the desired CFG register setting must be sent to the ADC. This is a 14-bit register on the ADC, so it is latched by the ADC on the first 14 rising edges of SPI_CLK. However, the ADSP-BF70x processor SPI port only supports 8-, 16-, or 32-bit transfers. As 16-bit mode is used, the fact that the interface's data format is MSB-first requires that the intended 14-bit configuration word be left-shifted by two bits such that it is left-aligned in the ADSP-BF70x processor's SPI buffer. For example, if the desired ADC CFG register setting is 0x3C41, it must be left-aligned in the SPI buffer as 0xF104 (0x3C41 << 2), otherwise the two zero-fill MSBs above the 14-bit configuration word (b#00) would be sent first, followed by the intended 14-bit configuration word's most significant 12 bits. The ADC will latch the two leading zeros and the 12 MSBs of the word, and then it will subsequently ignore the two LSBs of the intended configuration word (b#01) at the end of the 16-bit transfer, and the ADC will be misconfigured.

Further, the CFG register must be programmed at least twice before conversions can be expected to be occurring properly. After two conversions, the received 16-bit data is correct until the configuration is changed again. Any configuration change before or during operation requires two conversions before the data is valid again.

Modes of Operation

The AD7689 can operate with both slow hosts and fast hosts. These modes are defined to be:

- Read/Write During Conversion
- Read/Write After Conversion
- Read/Write Spanning Conversion

The first option is only applicable to fast hosts, whereas the second and third can be used with any host. The sample code demonstrates the first and second options, as they cover configuration of and using fast and slow hosts, respectively.

Read/Write During Conversion

This mode is useful if the host wants to read and write at high speeds, as shown in Figure 2.

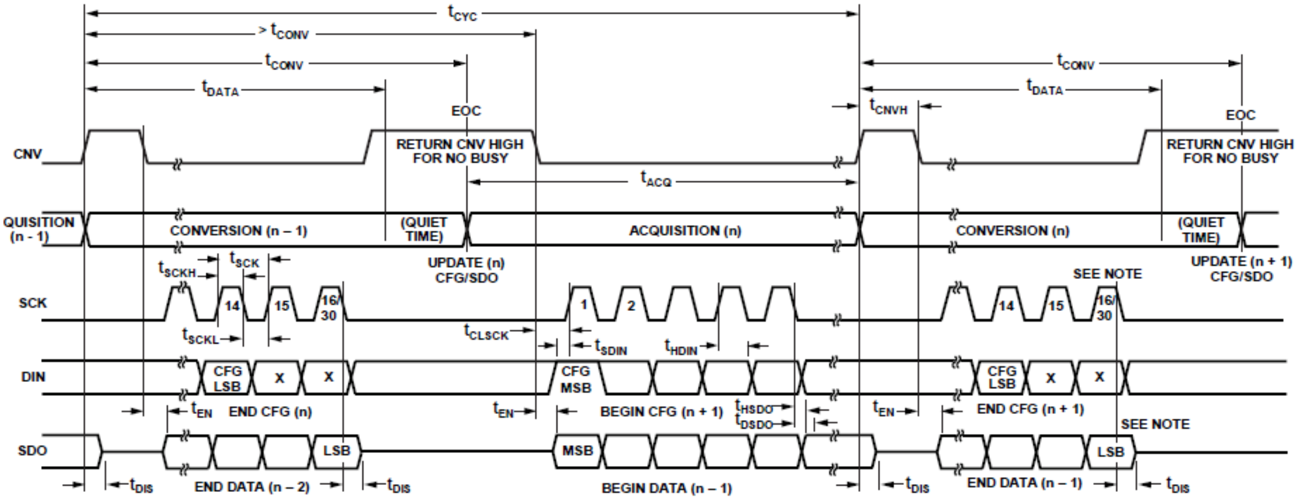

Figure 2 Read/Write During Conversion Timing Diagram

During an acquisition (n), the conversion results are for the preceding conversion (n-1) and writing is for the next acquisition (n+1). CNV is initially pulled high to signal the ADC to start conversion and is then pulled low. After this, the SPI read/write takes place. It is important that the SPI_CLK is fast enough to ensure that this happens before t_{data} has elapsed, otherwise the results being read/written can be corrupted. The programmer must ensure that there is no digital activity from t_{data} to t_{EOC}. CNV must be returned to high before t_{data} and held high beyond t_{EOC} to avoid the generation of the busy signal indicator.

For fast hosts, the AD7689 ***ADC datasheet*** indicates that the SPI clock frequency should be:

$$fsck \geq Number\_SCK\_Edges \div t\_data$$

Since it is 16-bit data being transferred, the *Number_SCK_Edges* is 16. The value of t_{data} is 1.2 µs; thus, *fsck* would have to be greater than or equal to 13.33 MHz to ensure that the correct results are read back.

Read/Write After Conversion

This mode is useful for slow hosts, with timing as depicted in Figure 3.

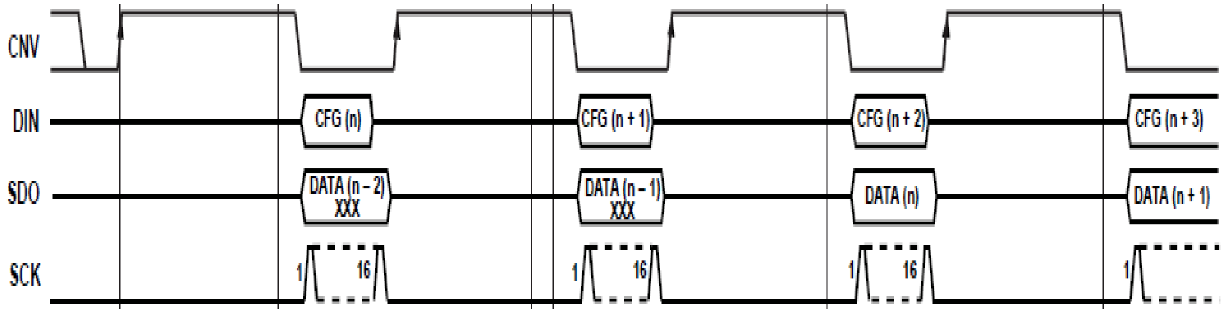

Figure 3 Read/Write After Conversion Timing Diagram

When using this mode during a particular acquisition (n), the conversion results are for the preceding conversion (n-1) and writing is for the next acquisition (n+1). The ADC is provided with an active-high `CNV` signal by the ADSP-BF70x processor to signal the ADC to start conversion. `CNV` must be kept high beyond the end of conversion period (see the ***ADC datasheet***) to avoid the generation of the busy signal indicator. After the EOC period is over, `CNV` is bought low again. This is followed by reading from and writing to the ADC over the SPI. The only timing restriction important in this mode is to ensure that `CNV` remains high beyond the EOC point.

Sample Output in CCES

To test the driver furnished with this EE-note, the AD7689 ADC was supplied with two analog waveforms using a signal generator:

- A 20 kHz/500 mV sine wave
- A 2 kHz/500 mV square wave

The analog signal was converted into a digital signal by the ADC, which was then received by the ADSP-BF70x processor via the SPI port and stored to on-chip data memory. This data buffer was then plotted from the processor's memory using the CCES plot feature, and the results are shown in Figure 4 (for the 20 kHz/500 mV sine wave) and Figure 5 (for the 2 kHz/500 mV square wave).

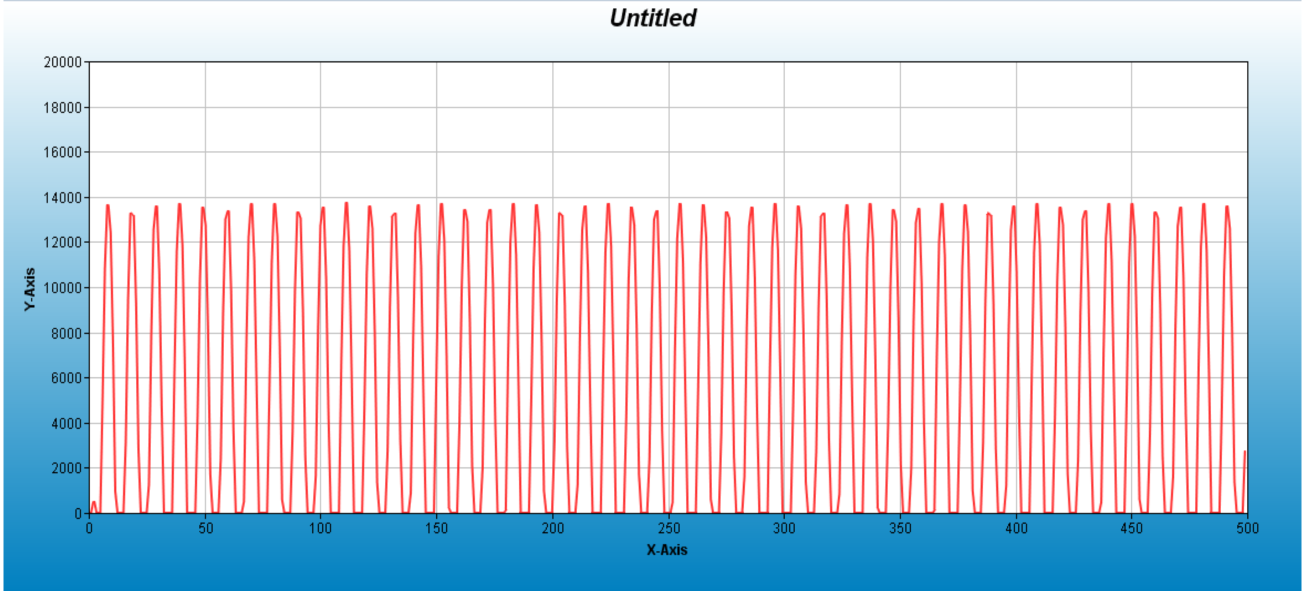

Figure 4 Sine Wave (20 kHz, 500mV) Output from ADC

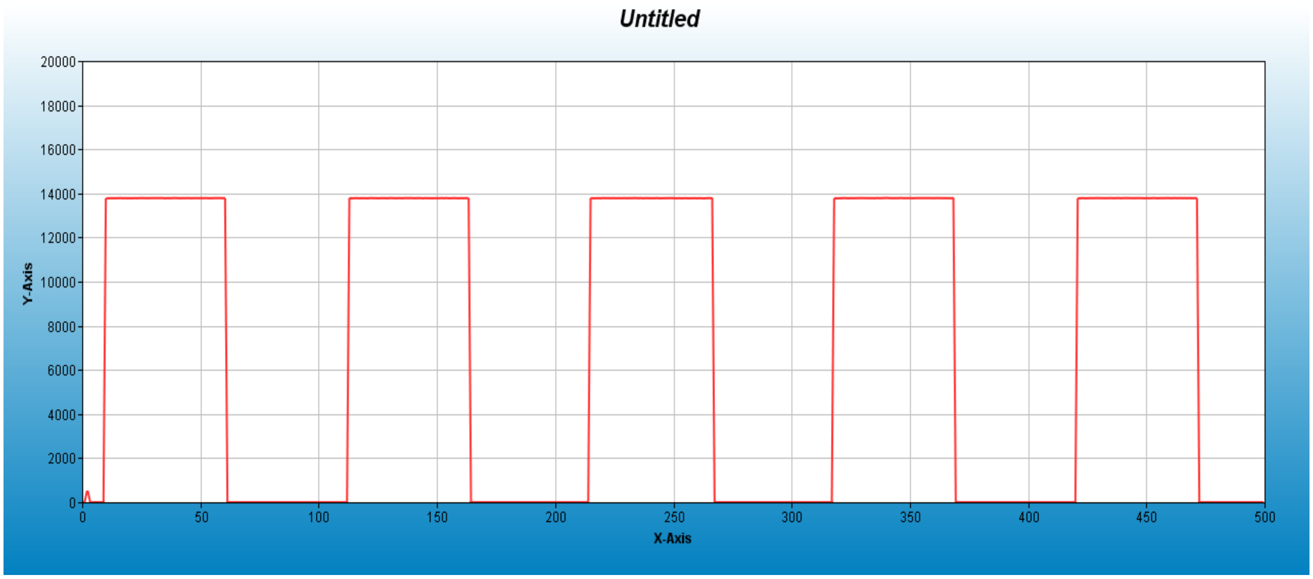

Figure 5 Square Wave (2 kHz, 500 mV) Output from ADC

References

[1] *Associated Code for Interfacing the AD7689 ADC to ADSP-BF70x Blackfin+ Processors (EE-395)*. Rev 1, Analog Devices, Inc., December 2016.

[2] *ADSP-BF700/701/702/703/704/705/706/707: Blackfin+ Core Embedded Processor Data Sheet*. Rev. A, Analog Devices, Inc., September 2015.

[3] *ADSP-BF70x Blackfin+™ Processor Hardware Reference*. Rev. 1.0, Analog Devices, Inc., October 2016.

[4] *AD7682/AD7689: 16-Bit, 4-Channel/8-Channel, 250 Ksps PulSAR ADC Data Sheet*. Rev. F, Analog Devices, Inc., 2008.

Readings

[1] *EVAL-AD76MUXEDZ: Evaluation Board For AD7682/89/99/7949 PulSAR ADCs*. Rev. PrD, Analog Devices, Inc., 2009.

[2] *ADSP-BF70x Blackfin+™ Processor Programming Reference*. Rev. 1.0, Analog Devices, Inc., October 2016.

Document History

Revision	Description
Rev 1 – December 15th, 2016 by Trinayan Baruah and Joe Tarkoff	Initial Release