



Interfacing AD7676 ADCs to ADSP-21365 SHARC® Processors

Contributed by Aseem Vasudev Prabhugaonkar and Jagadeesh Rayala

Rev 1 – October 7, 2004

Introduction

This application note explains how to interface AD7676 ADCs in master serial mode (internal discontinuous clock) to ADSP-21365 SHARC® processors. This application note also provides example code to demonstrate how the ADSP-21365 processor's Serial Port (SPORT) can be programmed to receive data from the AD7676 converter when the AD7676 is configured in master serial mode, supplying a discontinuous serial data clock to the processor's SPORT interface.

About AD7676 ADCs

AD7676 ADCs are 16-bit, 500 kSPS, charge-redistribution SAR, fully differential analog-to-digital converters (ADCs) that operates from a single 5-V power supply. In addition to the high-speed 16-bit sampling ADC, these parts also contain an internal conversion clock, error correction circuits, and serial and parallel system interface ports.

AD7676 ADCs are factory-calibrated and are comprehensively tested, ensuring that they meet or exceed their AC parameters such as signal-to-noise ratio (SNR) and total harmonic distortion (THD), in addition to the more traditional DC parameters of gain, offset, and linearity.

AD7676 applications include:

- CT Scanners
- Data Acquisition

- Instrumentation
- Spectrum Analysis
- Medical Instruments
- Battery-Powered Systems
- Process Control

AD7676 ADCs can operate in serial mode as well as parallel mode. In serial data mode, they can be configured to supply a serial data clock (master serial interface – internal clock) or they can take serial data clock externally (slave serial interface). In master serial mode, AD7676 converters provide a discontinuous bit clock. This guarantees that the conversion performance is not degraded because there are no voltage transients on the digital interface during the conversion process.

AD7676 Product Highlights

The AD7676 A/D converter provides:

- Excellent INL
 The AD7676 has a maximum integral non-linearity (INL) of 1.0 LSB with no missing 16-bit code.
- Superior AC performances
 The AD7676 has a minimum dynamic of 92 dB (94 dB typical).
- Fast throughput

The AD7676 is a 500 kSPS, charge-redistribution, 16-bit SAR ADC with internal error correction circuitry.

- Single-supply operation

The AD7676 operates from a single 5 V supply and typically dissipates only 67 mW. It consumes 7 μ W maximum in power-down.

- Serial or Parallel interface

Versatile parallel (8 or 16 bits) or 2-wire serial interface arrangement compatible with 3 V or 5 V logic.

About ADSP-21365 Processors

The third generation of SHARC processors, which includes the ADSP-21262, ADSP-21266, ADSP-21267, ADSP-21364, and ADSP-21365 derivatives, offers increased performance, audio and application-focused peripherals, and new memory configurations capable of supporting the latest surround-sound decoder algorithms. All devices are pin compatible and are completely code compatible with all prior SHARC processors. The newest members of the SHARC family are based on a single-instruction, multiple-data (SIMD) core, which supports both 32-bit fixed-point and 32-/40-bit floating-point arithmetic formats, making them particularly suitable for high-performance audio applications.

The ADSP-21365 derivative offers the highest performance – 300 MHz / 1800 MFLOPs – within the third generation SHARC processor family. This level of performance makes the ADSP-21365 particularly well suited to address the increasing requirements of the professional and automotive audio market segments. In addition to its high-performance core, the ADSP-21365 includes additional value-added peripherals such as an S/PDIF transmitter/receiver, 8-channel asynchronous sample rate converter, and hardware digital transmission content protection (DTCP) encryption/decryption block.

Third-generation SHARC processors also integrate application-specific peripherals that simplify hardware design, minimize design risks, and ultimately reduce time to market. Grouped together and broadly named the digital audio interface (DAI), these functional blocks may be connected to each other or to external pins via the software programmable signal routing unit (SRU). The SRU is an innovative architectural feature that enables complete and flexible routing among DAI blocks. Peripherals connected through the SRU include (but are not limited to) SPORTs, SPI ports, S/PDIF Tx/Rx, DTCP accelerator, and an 8-channel asynchronous sample rate converter block.

ADSP-21365 Serial Ports

The ADSP-21365 processor has six independent, synchronous serial ports (SPORTs) that provide an I/O interface to a wide variety of peripheral devices. They are called SPORT0, SPORT1, SPORT2, SPORT3, SPORT4, and SPORT5. Each serial port has its own set of control registers and data buffers. With a range of clock and frame synchronization options, the SPORTs allow a variety of serial communication protocols and provide a glueless hardware interface to many industry-standard data converters and codecs.

The ADSP-21365 SPORTs are four-wire interfaces with two data pins (SPORT_x_DA and SPORT_x_DB) each, one serial clock (SPORT_x_CLK), and a frame sync (SPORT_x_FS) pin. The two bi-directional channels (A and B) per serial port are configurable as either transmitters or receivers. Each SPORT can also be configured as two receivers or two transmitters, permitting two unidirectional streams into or out of the same serial port.

This interface uses the SPORT in core driven mode.

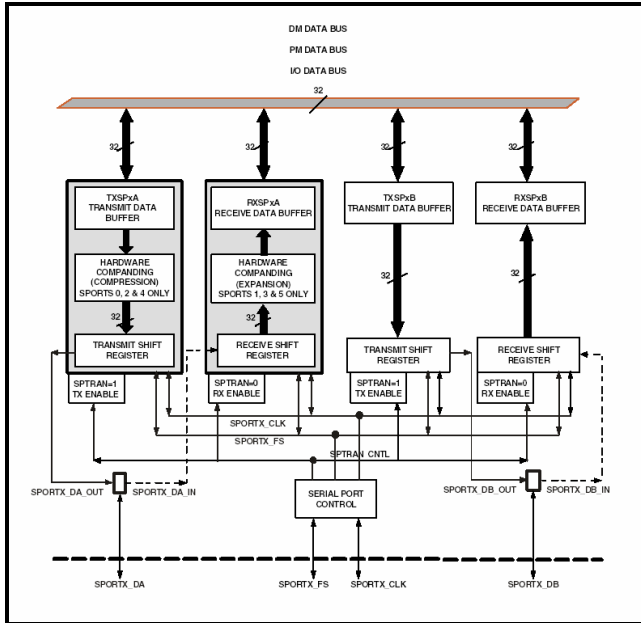


Figure 1. ADSP-21365 SPORT Block Diagram

AD7676-to-ADSP-21365 Interface

AD7676 ADCs can be configured in serial mode or in parallel mode to transfer 16-bit digitized data to the processor or micro-controller. In serial mode, the ADC can be configured to provide a serial bit clock for transferring data to a processor (this mode of operation is called master serial mode). In this mode, the serial clock is discontinuous (OR gated) and is present only while transferring the data, providing better noise immunity. This application note discusses an interface in master serial mode only since the serial clock supplied by the ADC is gated and this requires a specific sequence to receive data over serial port.

The serial interface of the AD7676 that are used with processor's serial port interface comprise the following signals:

- **/CNVST** This convert start signal starts conversion. A falling edge on this signal puts the internal sample-and-hold into hold state and initiates a conversion.
- **SDOUT** The AD7676 drives out conversion results on this pin. The data bits are clocked out on the rising edge or falling edge of the

serial clock, based on the state of the INVSCCLK pin.

- **SCLK** The converter clocks the data bits out on the serial clock edges. This can be an input (slave serial mode) or an output (master serial mode).
- **SYNC** This signal is used as digital output frame synchronization with the internal data clock. This can be configured as an active high or an active low signal from the converter.

Figure 2 shows the timing for the serial interface protocol.

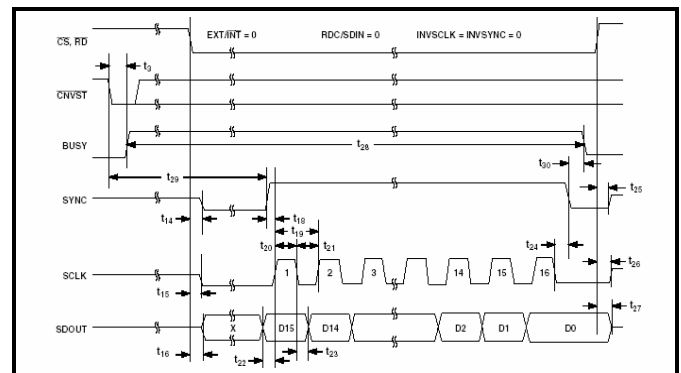


Figure 2. AD7676 Serial Interface Timing Diagram in Master Serial Mode – Read After Conversion

Refer to the AD7676 data sheet for detailed information about the timing specifications.

ADSP-21365 SPORT to AD7676

SPORT0 and SPORT1 are used for this interface. The SPORT0 frame sync (FS) is used to generate /CNVST for the ADC. The SPORT0 frame sync is configured to be active low and an early frame sync. Also, the frame sync is configured as data independent frame sync so that the frame sync is generated periodically.

SPORT1 is used to read samples from AD7676. The SPORT1 clock and frame sync are configured as external. The SPORT1 frame sync is configured to be active low and a late frame sync. The A/D converter is configured to strobe data bits on the clock's falling edges; thus, the

serial port is configured to sample data on the serial clock's rising edges.

Refer to [Figure 3](#) for details on the signal connections between processor and ADC.

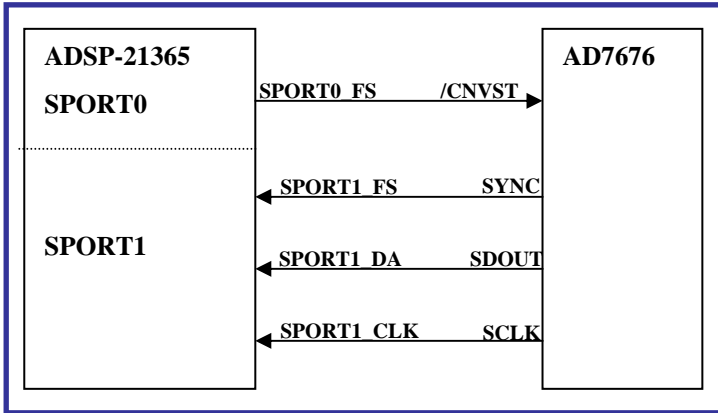


Figure 3. ADSP-21365 SPORT Interface with AD7676 - Block Diagram

SPORT1 is used to read digitized samples from the ADC. It must be configured initially for a serial word length of 15 bits. Inside the first receive interrupt service routine (ISR), the serial word length should be changed on-the-fly to 16 bits. Note that the first two reads (15-bit read

and the next immediate read) should be dummy reads for the interface to function properly.

[Figure 4](#) shows the serial interface signals.

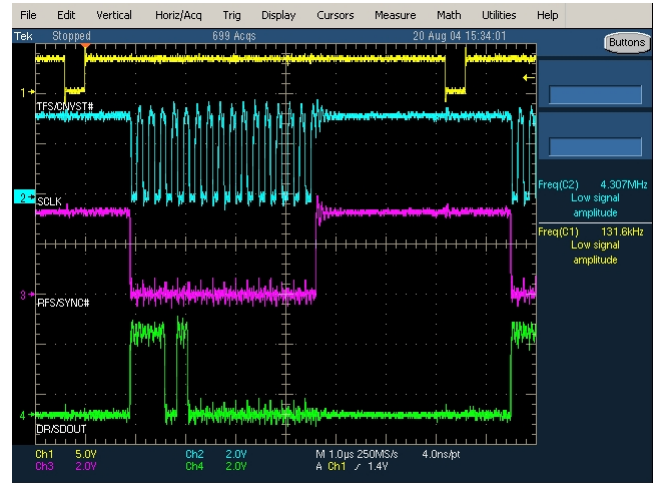


Figure 4. ADSP-21365 SPORT Interface with AD7676 – Oscilloscope Capture of Interface Signals

As can be seen in [Figure 4](#), the ADC drives the serial data on the serial clock's falling edges. The processor's serial port samples the data on the serial clock's rising edges.

Appendix

The project files are included in a ZIP file attached to this application note.

ADSP-21365_SerialPort_with_AD7676.asm

```
/*
//
// Name:      Interfacing ADSP-21365 with AD7676
//
//
//
(C) Copyright 2004 - Analog Devices, Inc. All rights reserved.

File Name:      ADSP-21365_SerialPort_with_AD7676.asm
Date Modified:  08/20/04                      Rev 1.0
Software:       VisualDSP++3.5 (July update)
Purpose:        To receive data from AD7676

*****/

#include <def21365.h>
#include <SRU.h>

#define FRAMED_MODE
//#define UNFRAMED_MODE

.section/dm seg_dmda;
    .var adc_data;
    .var counter = 0;

.section/pm seg_rth;
    nop;nop;nop;nop;
    nop;jump start;

// Sport 1 receive interrupt
.section/pm seg_sp;
    nop;jump isr; rti;rti;

.section/pm seg_pmco;
start:
    // Route the serial port signals to DAI pins
    call SRU_Init;

    // Configure TFS (CONVST) to be generated continuously
    r0 = 0x0;
    dm(SPCTL0) = r0;

    r0 = 0x120012;
    dm(DIV0) = r0;

    r0 = DITFS | SPEN_A | SLEN32 | LFS | ICLK | IFS | FSR | SPTRAN;
    dm(SPCTL0) = r0;
```

```

r0 = 0x12345678;
dm(TXSP0A) = r0;

// Enable interrupts
bit set mode1 IRPTEN;
bit set imask P3I;           // By default SPORT 1 interrupt is mapped to
                             // Peripheral interrupt 3

r0 = 0x0;
dm(SPCTL1) = r0;

r1 = 0x0;
r7 = 0x2;

#ifdef UNFRAMED_MODE
    r0 = SPEN_A | SLEN15 | LFS | LAFS | CKRE;
#endif
#ifdef FRAMED_MODE
    r0 = SPEN_A | SLEN15 | LFS | LAFS | FSR | CKRE;
#endif
dm(SPCTL1) = r0;

nop;
nop;
jump(pc, 0);

.section/pm seg_pmco;
isr:
    r3 = dm(RXSP1A);

    // If it is the first word received, reconfigure the sport for
    // 16 bits
    r0 = dm(counter);
    comp(r0,r1);
    if eq jump reconfigure;

    // Incrementing the counter
    r6 = dm(counter);
    r6 = r6 + 1;
    dm(counter) = r6;

    comp(r6,r7);
    if eq rti;

    dm(adc_data) = r3;
    rti;

.section/pm seg_pmco;
reconfigure:
    // Reconfiguring the serial port for 16 bits.
#ifdef UNFRAMED_MODE
    r0 = SPEN_A | SLEN16 | LFS | LAFS | CKRE;
#endif
#ifdef FRAMED_MODE
    r0 = SPEN_A | SLEN16 | LFS | LAFS | FSR | CKRE;

```

```

#endif
    dm(SPCTL1) = r0;

    r2 = 1;
    dm(counter) = r2;

    rti;

.section/pm seg_pmco;
SRU_Init:

    // Route SPORT0_FS signal to DAI pin 5
    SRU(SPORT0_FS_O,DAI_PB05_I);
    SRU(HIGH,PBEN05_I);

    // Route SPORT1_CLK signal to DAI pin 7
    SRU(DAI_PB07_O,SPORT1_CLK_I);
    SRU(LOW,PBEN07_I);

    // Route SPORT1_FS signal to DAI pin 9
    SRU(DAI_PB09_O,SPORT1_FS_I);
    SRU(LOW,PBEN09_I);

    // Route SPORT1_DA signal to DAI pin 11
    SRU(DAI_PB11_O,SPORT1_DA_I);
    SRU(LOW,PBEN11_I);

    rts;

```

References

- [1] *ADSP-21365 DSP Hardware Reference Manual*. Preliminary – Not yet released. Analog Devices, Inc.
- [2] *AD7676 Preliminary Technical Data Sheet*. Rev B. Analog Devices, Inc.
- [3] *ADSP-21365 DSP Evaluation System Board Schematics*. Analog Devices, Inc.

Document History

Revision	Description
<i>Rev 1 – October 07, 2004 by Aseem Vasudev Prabhugaonkar</i>	Initial Release