Application Report
SLAA175 – JUNE 2003

Interfacing the ADS8320/ADS8325 to The TMS320C6711 DSP

Lijoy Philipose *Data Acquisition Applications*

ABSTRACT

This application note presents a method for interfacing the ADS8320/ADS8325 16-bit 100 KSPS SAR analog-to-digital converter to the TMS320C6711 DSP. Software developed for use in this application reads continuously from the converter. Explained within is how to use the BIOS/CSL and the configuration tool in Code Composer Studio™ to set the McBSP and hardware interrupt vectors. The software can also be used for interfacing the ADS8321 and ADS8324 analog-to-digital converters to any McBSP. In an effort to reduce development time, the source code is available via TI's website.

Contents

Figures

1 Introduction

The ADS8320/ADS8325 is a single-channel, 100-KSPS, 16-bit analog-to-digital converter. The converter gluelessly interfaces to the TMS320C6711 digital signal processor (C6711 DSP). The TMS320C6711 DSP starter kit (DSK) and DEM-MSOP8 EVM were used for system development.

This application note explains the software interface, including how to set the McBSP register settings and data recovery. The application code continously reads samples from the converter and stores them in an array. Software written for this note is available for download from TI's website.

2 Hardware and Software Tools

The TMS320C6711 DSP starter kit is a low-cost medium for interfacing the TMS320C6711 DSP to the ADS8320/ADS8325. Figure 1 shows how the digital lines should be connected together. In this report the ADS8320/ADS8325 is interfaced to a C6711 DSP via multichannel buffered serial port 1 (McBSP1).

The ADS8320/ADS8325 communicates with microprocessors via a synchronous 3-wire serial interface. The DCLOCK signal synchronizes the data transfer, with each bit being transmitted on the falling edge of DCLOCK. Therefore the McBSP is programmed to receive data on the rising edge of CLKX1. A falling $\overline{CS}$ signal initiates the conversion and data transfer. As mentioned above, FSX1 is wired to $\overline{CS}$. The first 4.5 to 5.0 clock periods of the conversion cycle are used to sample the input signal. After the fifth falling DCLOCK edge, D_{OUT} is enabled and outputs a LOW value for one clock period. For the next 16 DCLOCK periods, D_{OUT} outputs the conversion result—most significant bit first. After the least significant bit (B0) has been output, subsequent clocks repeat the output data, but in a least significant bit first format. The full conversion cycle is therefore (5+1+16) = 22 clocks, for this reason the McBSP transmit and receive word size are set to 24 bits.

The transmit clock pin, CLKX1 of McBSP1 is wired to DCLOCK pin. The transmit frame sync, FSX1 of McBSP1, is wired to $\overline{CS}$. Finally, the receive-data register (DRR1) is wired to Dout pin.

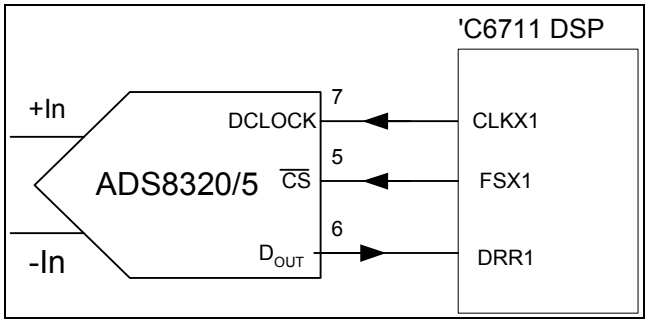

Figure 1. Hardware Interface

2.1 TMS320C6711 DSK

The TMS320C6711 DSP starter kit (DSK) not only provides an introduction to C6000™ DSP technology, but is powerful enough to use for fast development of networking, communications, imaging and other applications like data acquisition. Search TI's website (www.ti.com) for more information on the C6711™ DSP starter kit or go to the link below:

http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=TMDS320006711

2.2 5-6K Interface Card

The 5-6K interface card provides data converter customers with a great amount of flexibility for the evaluation of data acquisition products from Texas Instruments. The interface card provides direct plugin of different analog-to-digital, (or digital-to-anlaog) converter evaluation modules (EVMs) onto this interface card, which then sits on the C5000™ DSP or C6000™ DSP starter kit. This hardware stackup allows software coding to begin right away. The card consists of two signal conditioning sites, two serial EVM sites, and a parallel EVM site. It enables rapid prototyping of the complete signal chain. The link below provides more information on this interface board.

http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=5-6KINTERFACE

2.3 DEM-MSOP8

The DEM-ADS-MSOP8 A/D converter evaluation fixture provides for quickly testing the ADS8320/ADS8325 on the platform. This particular board does not plug directly into a 5-6k interface card, so for this report three wires are used to connect the digital I/O from ADS8320/ADS8325 to the McBSPa slot of the 5-6k interface card. Daughtercard EVMs are currently being developed for plugging directly into the interface card. For more information on the DEM-MSOP8 evaluation fixture go to the link below:

http://focus.ti.com/docs/tool/toolfolder.jhtml?PartNumber=DEM-ADS-MSOP8

2.4 Software Development Tool

The software is written using Code Composer Studio™. The Code Composer Studio (CCStudio) development tools provide a fully integrated development environment (IDE) supporting Texas Instruments industry-leading TMS320™ DSP platforms. Code Composer Studio IDE is the first intelligent development environment to offer TMS320C2000™ DSP, TMS320C5000™ DSP, TMS320C6000™ DSP, and OMAP™ application development for multiprocessor, multiuser and multisite projects.

CCStudio integrates all host and target tools in a unified environment to simplify DSP system configuration and application design. The unified environment includes TI's DSP/BIOS™ kernel, code-generation tools, fast simulators, debugger, and real-time data exchange (RTDX) technology. CCStudio also has an open architecture that allows TI and third party developers to extend the IDE functionality by seamlessly plugging in additional specialized tools.

For more information, readers may search TI's website.

3 Program

The code developed for this application note uses the sample rate generator of McBSP1 to trigger conversions every 10 µs. The DSP, via an interrupt service routine, reads a 24-bit word, and moves the sample into a 16-bit array called *ad_buffer*. BIOS/CSL is used to initialize all the DSP and McBSP registers; therefore, all the registers are set before branching to the main function. As a result, in the main function, only two actions need to be performed—first to enable interrupts, and second to start McBSP port 1. The McBSP receiver interrupt service routine is called to read the sample of the receiver and store the data.

3.1 McBSP Setting

The following sections highlight how to use the configuration tool to set up the McBSP and hardware interrupts.

1. As shown in Figure 2, on the Operating Mode tab, place the McBSP in SPI mode by selecting Delay (SPI) from the pulldown options in the Clock Stop Mode field.

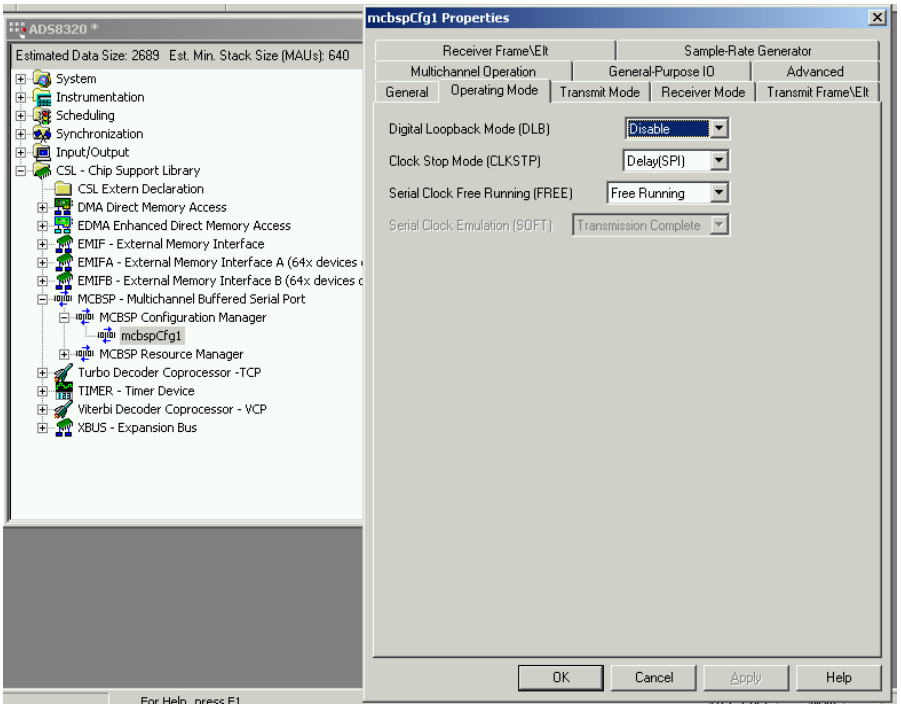

Figure 2. Operating Mode Tab

2. As we see in the hardware interface diagram (Figure 1), the transmit portion of the McBSP is used for DCLOCK and CS to the ADS8320/ADS8325. On the Transmit Mode tab (Figure 3), CLKX1 is set as output, with rising-edge polarity. The first edge after $\overline{CS}$ goes low is a rising edge. In this case the sample-rate generator is set to trigger new conversion cycles. NOTE: The Sample-Rate Generator tab is where the conversion frequency is set, described in greater detail in step 6. Also, use the Transmit Mode tab to set the polarity of the $\overline{CS}$. It must be set to Active Low to ensure FSX1 goes low when a new cycle begins.

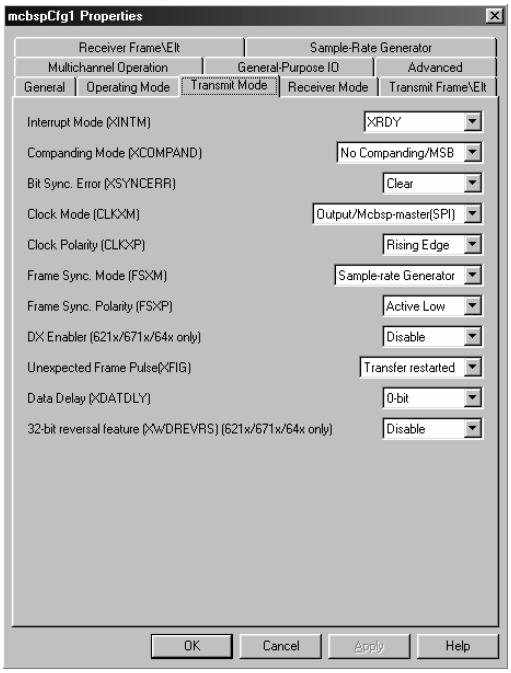

Figure 3. Transmit Mode Tab

3. As shown in Figure 4, the phase, number of words per frame, and word length are set using the Transmit Frame tab. Set the options for 24-bit words per single-phase.

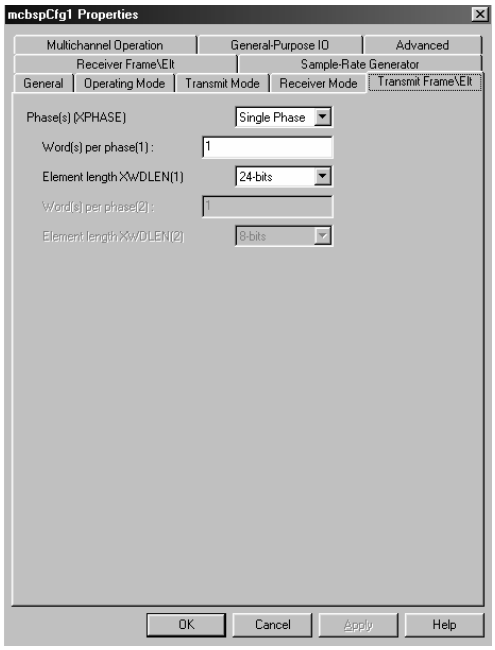

Figure 4. Transmit Frame Tab

4. As shown in Figure 5 (Receive Mode tab), the receive register setting is ignored in SPI mode. CLKR and FSR are internally tied to CLKX and FSX, respectively. The polarity and phase of clock are set by CLKSTP and CLKXP bit fields in SPCR.

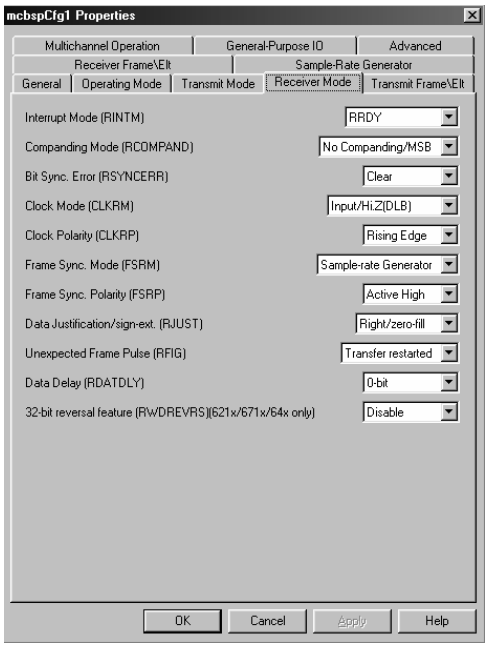

Figure 5. Receive Mode Tab

5. Set the Receive Frame tab as shown in Figure 6. Again, each conversion is a single-phase, 24-bit word.

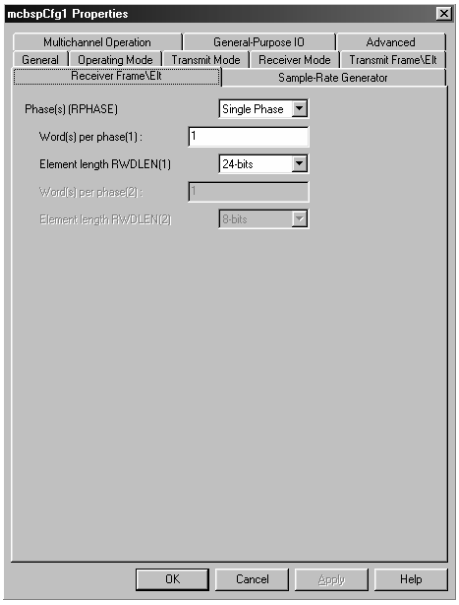

Figure 6. Receive Frame Tab

6. In the Sample-Rate Generator tab, select the options shown in Figure 7. The analog-to-digital converter used for developing this application note was sampled at 100 kHz. Set the Frame period to 25. This generates 24 clocks in each cycle. Set Frame width to 1, ($\overline{CS}$ is high for only one cycle). There are two ways to trigger a conversion cycle. The first is by writing data into the data transmit register, and the second is by using the sample-rate generator. The sample-rate generator is used in this report because it generates a periodic conversion rate—select Frame Sync Signal from the dropdown menu. Using the Sample-Rate Generator Clock Mode dropdown menu, set the sample-rate generator to be derived off the CPU clock. The input clock to the C6711 DSP sample-rate generator is half the CPU clock; in the case of the C6711 DSP starter kit it for this application note it was 80 MHz. The clock frequency divider field is set to 30 to generate 2.5-MHz DCLOCK.

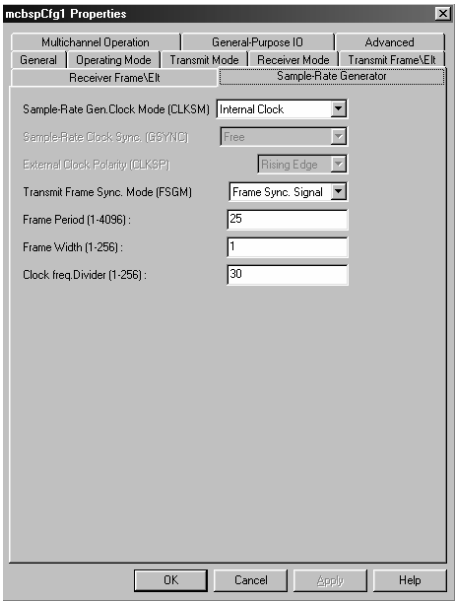

Figure 7. Sample-Rate Generator Tab

7. After the McBSP configuration object is set, the object then must be mapped into the McBSP_Port1. To do this, right-click on the Mcbsp_Port1 and select the options shown in Figure 8.

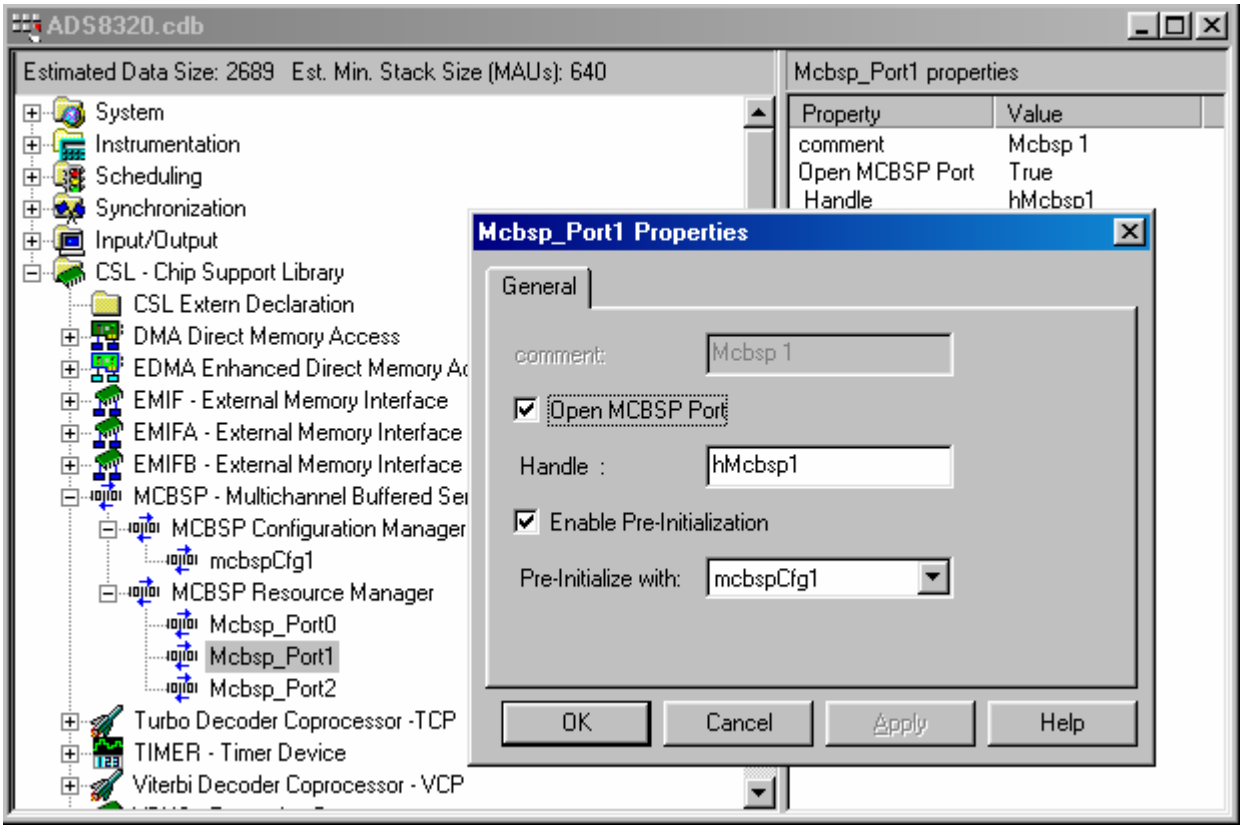

Figure 8. McBSP1 Configuration Object Mapping

3.2 HWI Setup

The only remaining item to set up via the configuration tool is the McBSP receive Hardware Interrupt.

1. Chose to map the McBSP1 Receive interrupt to HWI_INT13. Type the name of the interrupt service function with an underscore, as shown in Figure 9.

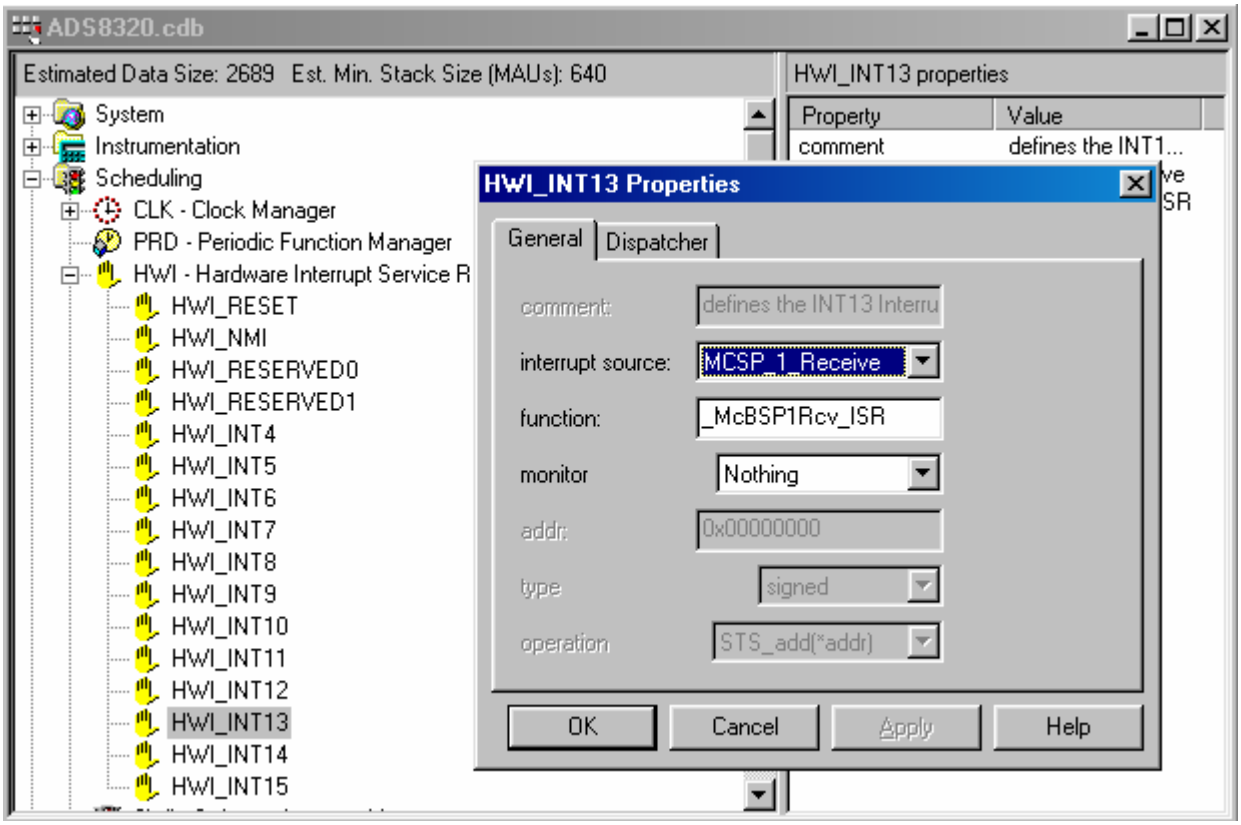

Figure 9. McBSP1Rcv_ISR General Tab

2. Select Dispatcher tab, and select options as shown in Figure 10. BIOS enables the interrupt and handles branching to and from the McBSP1Rcv_ISR function.

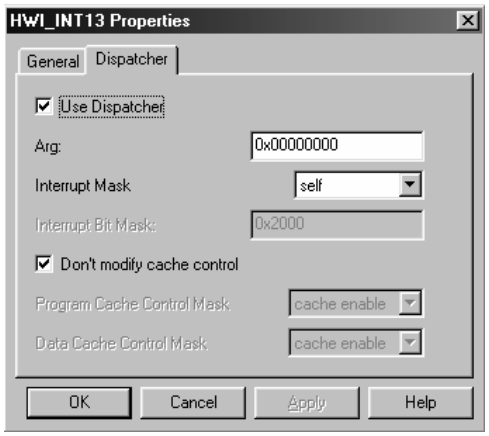

Figure 10. McBSP1Rcv_ISR Dispatcher Tab

3.3 Software Flow

BIOS/CSL initializes all the DSP and registers of the peripherals discussed in the previous sections before branching to the Main function. In the Main function, the user needs only to enable interrupts and start the McBSP. The McBSP sample-rate generator triggers conversions at 100 kHz. Each conversion cycle generates an interrupt from the McBSP receiver. This interrupt is serviced by McBSP1Rcv_ISR, which reads the 24-bit word, captures the 16-bit sample, and stores it in *ad_buffer*. When the *ad_buffer* array is full, the index to the buffer is reset and the sample index vector is reset to 0. A flow chart of the functions is shown as Figure 11. The user program is shown in Appendix A.

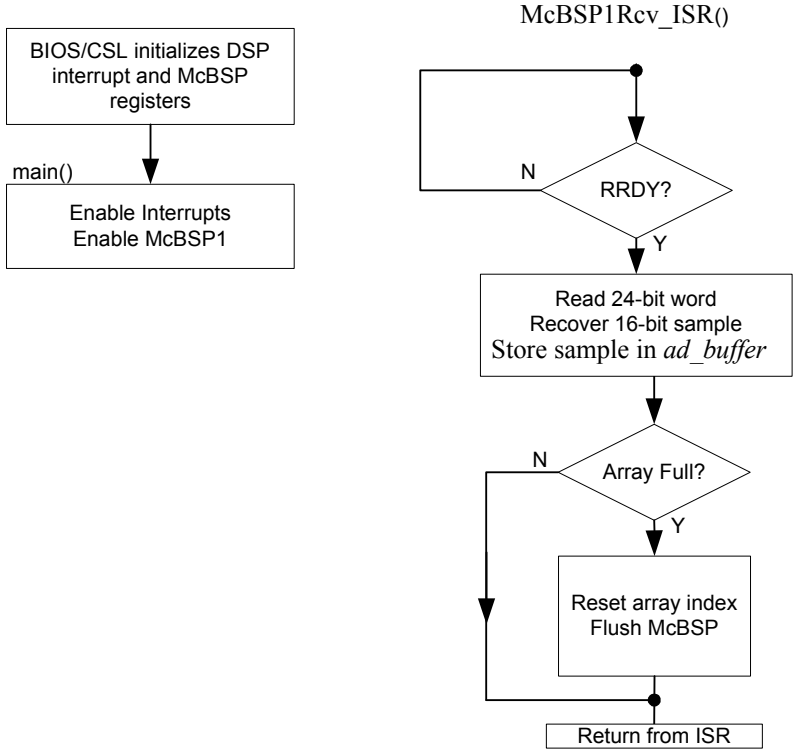

Figure 11. Flow Chart

4 Conclusion

This application note explained the software interface, including how to set the McBSP register settings and data recovery. The application code provided continously reads samples from the converter and stores them in an array. Software written for this note is available for download from TI's website.

5 References

1. TMS320C6000 DSP/BIOS User's Guide (spru303)
2. Code Composer Studio Getting Started Guide (spru509)
3. Code Composer Studio User's Guide (spru328b)
4. TMS320C6000 Code Composer Studio Tutorial (spru301c)
5. TMS320C6000 Chip Support Library API Reference Guide (spru401)
6. TMS320C6000 Peripherals Reference Guide (spru190)
7. ADS8320 Product Datasheet (SBAS108)
8. ADS8325 Product Datasheet (SBAS226)

6 Trademarks

Code Composer Studio is a trademark of Texas Instruments

TMS320C2000, TMS320C5000, TMS320C6000 DSP platforms, and C6711 DSP generation are trademarks of Texas Instruments

OMAP is a trademark of Texas Instruments

Appendix A – MAIN.C

```
/*********************************************************/
/* FILENAME: main.c                                      */
/* DESCRIPTION: This program uses McBSP1 to read 1024    */
/*  samples continuously from the ADS8320/ADS8325 16-bit 100KSPS    */
/*  Analog-to-Digital Converter.  The samples are stored in */
/*  the buffer called ad_buffer. Code developed using    */
/*  'C6711 DSK and DEM-MSOP8 EVM.                        */
/* AUTHOR   : DAP Application Group, L. Philipose, Dallas   */
/*    CREATED 2002(C) BY TEXAS INSTRUMENTS INCORPORATED.    */
/* VERSION: 1.0                                          */
/*********************************************************/


/* Header file */
#include "csl.h"
#include "csl_irq.h"
#include "csl_mcbsp.h"

/*Declarations*/
#define BLOCK_SZ 1024 /* size of data buffer */

/*DSP/BIOS variables*/
extern far MCBSP_Handle hMcbsp1;

/*Global variables*/
unsigned short ad_buffer[BLOCK_SZ], i=0;

/*************************************************************************\
* Function: main()
* Description: Enables McBSP1 receive interrupt and McBSP1
\*************************************************************************/

void main() {

/*Enable McBSP1 Recieve Interrupt*/
    IRQ_enable(IRQ_EVT_RINT1);
/*McBSP is pre-initialize as SPI by BIOS/CSL before entering into main()*/
/*Sample-rate generator will trigger conversions at 100KSPS, assuming  */
/*code is running on 150MHz C6711*/
/*Start McBSP1 */
    MCBSP_start(hMcbsp1,MCBSP_RCV_START | MCBSP_XMIT_START | MCBSP_SRGR_START|
MCBSP_SRGR_FRAMESYNC, 0);

/*McBSP1 Receive Interrupt Service routine will extract 16-bit and store in data array
ad_buffer*/
}


/*************************************************************************\
* Function: McBSP1Rcv_ISR()
* Description: McBSP1 Receive Interrupt Service Routine.  Reads sample out of
* receive register. extracts 16-bit data from 24-bit read and stores it
* in array call ad_buffer.  When the buffer table is full, it resets the
* pointer to the beginning and flushs receive buffer.
\*************************************************************************/
```

```
void McBSP1Rcv_ISR(void)
{   unsigned int data;

    /* Wait until a value is received then read it */
    while (!MCBSP_rrdy(hMcbsp1));
    data = (MCBSP_read(hMcbsp1) & 0x0003FFF6) >> 2;

/*Store sampled data into 16-bit array ad_buffer*/
    ad_buffer[i++] = data;

    if (i >= BLOCK_SZ )             /*Reset index?      */
    {
        i=0;
//      MCBSP_read(hMcbsp1);        /*Flush McBSP Receiver*/
//      MCBSP_read(hMcbsp1);        /*Flush McBSP Receiver*/
//      MCBSP_read(hMcbsp1);        /*Flush McBSP Receiver*/
//      MCBSP_reset(hMcbsp1);       /*Uncomment to reset McBSP1*/
    }

}
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265