# Backless Engineer To Engineer Note EE-156

Technical Notes on using Analog Devices' DSP components and development tools Phone: (800) ANALOG-D, FAX: (781) 461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

# Support for the H.100 protocol on the ADSP-2191

DSP Applications Group, March 2002 - Rev 0.1

The serial ports of the ADSP-2191 provide support for the H.100 standard protocol. It is also the International Telecommunication Union's recommendation for visual telephone systems. This application note describes the configurations necessary to be compatible with the H.100 standard protocol.

The hardware used to test the support for this protocol was the ADSP-2191 EZ-KIT Lite. The software used was VisualDSP++ 2.0 with SP1. The example code is included at the end of this application note.

# Introduction

The ADSP-2191 has three independent, synchronous serial ports (SPORT0, SPORT1, and SPORT2). Each one of the serial ports supports H.100. In this example, SPORT1 is used. On each SPORT, data can be simultaneously transferred in both transmit and receive directions.

For each SPORT, data is transmitted from the IO bus to the Transmit Data register. After optional companding, data is transferred to the Transmit Shift register. Here the bits are shifted out serially on the SPORT's DT pin. The reverse happens for the receive direction. Data is accepted on the SPORT's DR pin, and serially transmitted to the Receive Shift register. After a word is received and optional decompanding, the data is transferred to the Receive Data register.

By writing to different control registers, the serial clock frequency, data format and length, multi-channel mode select, and other parameters can be programmed on the SPORTs.

Direct Memory Access (DMA) is also supported on each SPORT. The transmit channel enables DMA transfers from memory to SPORT, while the receive channel enables DMA transfers from SPORT to memory.

# Hardware Connections

On the ADSP-2191 EZ-KIT Lite, the SPORT0 and SPORT1 pins are brought out via the P6 and P7 SPORT connectors, respectively.

Using SPORT1 in this application, the DT1 and DR1 pins are connected together to create a loopback of the data transmitted and received.

Because H.100 applications always runs in multi-channel mode, the TCLK1 and RCLK1 pins must be wired together. In multi-channel mode, the TCLK pin is always an input. In this application, RCLK1 is generated internally.

Copyright 2001, Analog Devices, Inc. All rights reserved. Analog Devices assumes no responsibility for customer product design or the use or application of customers' products or for any infringements of patents or rights of others which may result from Analog Devices assistance. All trademarks and logos are property of their respective holders. Information furnished by Analog Devices Applications and Development Tools Engineers is believed to be accurate and reliable, however no responsibility is assumed by Analog Devices regarding the technical accuracy of the content provided in all Analog Devices' Engineer-to-Engineer Notes.



SPORT1



## **Software Configurations**

The assembly program consists of 4 main sections. These sections perform the tasks of initializing SPORT1, configuring SPORT1 for multi-channel mode operation, setting up autobuffer-based DMA, and enabling SPORT1.

To be compatible with the H.100 standard protocol, certain parameters must be set accordingly to meet the H.100 specification. The following is a list of these parameters:

- multi-channel mode enabled
- 8-bit words
- 1024 clock cycles per frame, 122 ns wide, 125 us period frame sync
- transmit/receive frame sync required
- active low transmit/receive frame sync
- 8.192 MHz (+/- 2% bit clock)
- no frame delay between frame sync and first data bit
- half clock cycle early frame sync

#### Initializing SPORT1

The transmit configuration register is initialized to the Hex value of 0x1CF0. This value configures SPORT1 to transmit 8-bit words, generate a transmit frame sync for every data word, and selects an active low transmit frame sync. Note that the SPORT1 Transmit Enable bit is not set here. This bit will be set last, after all other configurations have been programmed.

#### SP1\_TCR (0x03:0x000)



Figure 2: Transmit Configuration Register

The receive configuration register is initialized to the Hex value of 0x1EF2. This value configures SPORT1 to receive 8-bit words, generate a receive frame sync for every data word, and selects an active low receive frame sync. Note that the SPORT1 Receive Enable bit is not set here. This bit will be set last, after all other configurations have been programmed.

#### SP1\_RCR (0x03:0x001)



Figure 3: Receive Configuration Register

The SP1\_TFSDIV and SP1\_RFSDIV registers hold the number of transmit and receive clock cycles to count before generating a transmit or receive frame sync. Both of these registers are set to a Hex value of 0x03FF. This gives a value of 1024 clock cycles per frame.

The serial clock frequency is calculated according to the following equation:

$$SP1\_CLK = \frac{HCLK}{2 * (SP1\_SCKDIV + 1)}$$

#### EE-156

Page 2

Technical Notes on using Analog Devices' DSP components and development tools Phone: (800) ANALOG-D, FAX: (781)461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp With an input clock of 13 MHz and a multiplier of 10, the core clock frequency becomes 130 MHz. Having the peripheral clock HCLK be half of core clock and setting both the SP1\_TSCKDIV and SP1\_RSCKDIV registers to be 0x3, the SP1\_TCLK and SP1\_RCLK become 8.125 MHz. This frequency is within the +/- 2% of 8.192 MHz range.

#### Multi-channel mode operation

There are 8 multi-channel transmit selection registers and 8 multi-channel receive selection registers. Each register consists of 16 bits. Each bit corresponds to 1 channel. By setting the 8 multi-channel transmit selection registers to a Hex value of 0xFFFF, all 128 transmit channels (8 registers \* 16 channels) are enabled.

By setting the 8 multi-channel receive selection registers to a Hex value of 0xFFFF, all 128 receive channels (8 registers \* 16 channels) are enabled.

There are two multi-channel configuration registers for SPORT1. SP1\_MCMC1 is initialized to a Hex value of 0x01E1. This setting enables multi-channel mode operations and a window size corresponding to 128 channels.

#### SP1\_MCMC1 (0x03:0x019)

0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1

Figure 4: Mulit-Channel Configuration Register 1

SP1\_MCMC2 is initialized to a Hex value of 0x02AC. In this setting, a half clock cycle early frame sync is enabled.

EE-156

#### SP1\_MCMC2 (0x03:0x01A)



Figure 5: Mulit-Channel Configuration Register 2

#### Autobuffer-based DMA

In this section of code, four registers must be configured to setup SPORT1 autobuffer-based DMA on both the transmit and receive.

Both the transmit and receive DMA configuration registers are initially set to a Hex value of 0x0010. This setting enables autobuffer mode.

Then, the receive DMA start page and address registers are written with the page and address at which the receive data buffer is located. The receive DMA count register is written with the length of the receive data buffer.

The transmit DMA start page and address registers are written with the page and address at which the transmit data buffer is located. The transmit DMA count register is written with the length of the transmit data buffer.

Next, the receive DMA configuration register is written again with a Hex value of 0x0013. This setting configures a receive DMA transfer and enables DMA.

#### SP1DR\_CFG (0x03:0x101)

0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
•	0	0	•	0	•	•	•	•	•	•	-	•	•	-	-

Figure 6: SPORT1 Receive DMA Configuration Register

The transmit DMA configuration register is written again with a Hex value of 0x0015. This

setting enables DMA and an interrupt upon completion of the DMA.

#### SP1DT\_CFG (0x03:0x181)

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1																
	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

Figure 7: SPORT1 Transmit DMA Configuration Register

#### **Enable SPORT1**

After all of the above configurations are set up, SPORT1 for both transmit and receive are enabled by setting bit 0 of the SP1\_TCR and SP1\_RCR registers, respectively.

### Results

The following diagrams are some screenshots taken from the results of running this applications.



Figure 8: Timing Diagram 1



Figure 9: Timing Diagram 2

· ■ 参 物 III 回 B+ B+ 49 66 每 每 每 =															
h100.dpj 🔽	DM [Hexade	cimal]		DM [Heyadecimal]											
h100.dpj // /	DM         Hexade           [008006]         [008006]           [008006]         [008002]           [008012]         [008012]           [008012]         [008012]           [008024]         [008024]           [008036]         [008036]           [008048]         [008048]           [008048]         [008048]           [008048]         [008048]           [008055]         [008055]	cimal] spor 0000 0006 0000 0012 0012 0018 001E 0024 0024 0024 0030 0036 0036 0036 0036 0036 0036 0036 0036 0036 0054 0054 0054 0054 0054 0054 0054	C, tx_l 0001 0007 000D 0013 0019 001F 0025 002B 0031 0037 003D 0043 0049 004F 0055 0055 0055	000 4 0002 0008 0004 0014 0020 0026 0022 0032 0032 0032 0032 0032	0003 0009 000F 0015 0018 0021 0027 002D 0033 0037 003F 0045 0045 0045 0045 0057 0057 0057	00004 000A 0010 0016 0012 0022 0028 0022 0034 0034 0040 0046 0046 0046 0042 0052 0058 0055 0055 0055	00005 000B 0011 0017 001D 0023 0029 0035 003B 0047 0040 0053 0059 0059		DM [Hexaded [008080] [008086] [008080] [008098] [008098] [008089] [008080] [008080] [008080] [008080] [008080] [008080] [008002] [008004] [008004] [008004]	imal] spor 0000 0006 000C 0012 0018 0024 0024 0030 0036 0036 0036 0036 0048 0048 0048 0054 0054	t_rx_ 0001 0007 000D 0013 0019 0025 0028 0031 0037 0030 0043 0049 0044 0055 0055 0055	0002 0008 000E 0014 0014 0020 0022 0032 0038 0038 0038 0044 0044 0050 0056 0055	0003 0009 000F 0015 0018 0027 002D 0033 0039 0035 0048 0048 0051 0057 0057	00004 000A 0010 0016 0022 0028 0022 0034 0034 0034 0040 0046 0042 0052 0058 0058	00005 000B 0011 0017 0023 0029 0025 0035 0038 0041 0047 0047 00453 0059 0055
	[008066] [00806C] [008072]	0066 006C	0067 006D 0073	0068 006E 0074	0069 006F	006A 0070 0076	006B 0071		[0080E0] [0080E6] [0080EC]	0060 0066 006C	0061 0067 006D	0062 0068 006E	0063 0069 006F	0064 006A 0070	0065 006B 0071
	[008078] [00807E]	0078 007E	0079 007F	007A	007B	007C	007D		[0080F2] [0080F8] [0080FE]	0072 0078 007E	0073 0079 007F	0074 007A	0075 007B	0076 007C	0077 007D
								1							

Figure 10: Transmit and Receive Data Buffer Results

#### References

1. Analog Devices, Inc. DSP Home: http://www.analog.com/technology/dsp

Page 4

#### EE-156

Technical Notes on using Analog Devices' DSP components and development tools Phone: (800) ANALOG-D, FAX: (781)461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

```
io(IPR0) = ar;
  2. International Telecommunication Union:
                                               io(IPR2) = ar;
      www.itu.int
                                               io(IPR3) = ar;
  3. H.100 References:
                                               /*Set SPORT1 Registers*/
      http://www.scsa.org/
                                               IOPG = SPORT1 Controller Page;
      http://www.ectf.org/
                                               ar = 0x0000;
                                               io(SP1 TCR) = ar;
                                               io(SP1 RCR) = ar;
                                               io(SP1DR CFG) = ar;
                                               io(SP1DT CFG) = ar;
                                               io(SP1 MCMC1) = ar;
                                               ar = 3;
Example Code
                                               io(SP1DT IRQ) = ar;
                                               io(SP1DR IRQ) = ar;
#include <def2191.h>
                                            /*SPORT Tx and Rx Clock Divide*/
                                            ax0=0x3;
.section/dm seg dmdata;
.var sport_tx_buffer[128] = "h100.dat";
                                            io(SP1 RSCKDIV) = ax0;
                                            io(SP1 TSCKDIV) =ax0;
.var sport rx buffer[128];
                                            /*SPORT Tx and Rx Frame Sync*/
/******************************
                                            /*1024 clock cycles per frame,
/**** Reset Vector *****/
/********************************
                                            8bits*128channels*/
                                            ax0=0x3ff;
                                            io(SP1 RFSDIV) = ax0;
.section/pm seg ivt;
                                            io(SP1 TFSDIV) = ax0;
  jump start;
                                            /*SPORT Rx Configuration Register*/
.section/pm seg ivtint4;
                                            /*SLEN, TFSR/RFSR, LTFS, DITFS(only
  ay0 = IOPG;
                                            TX) */
                                            ax0=0x1Ef2;
  /*Set SPORT1 Registers*/
  IOPG = SPORT1 Controller Page;
                                            io(SP1 RCR) = ax0;
  ar = 3;
                                            /*SPORT Tx Configuration Register*/
  io(SP1DT IRQ) = ar;
                                            /*SLEN, TFSR/RFSR, LTFS, DITFS(only
  IOPG = a\overline{y}0;
                                            TX)*/
                                            ax0=0x1cf0;
  nop; nop;
                                            io(SP1 TCR) = ax0;
  rti;
                                            /** Multichannel Mode Configuration **/
.section/pm seg pmcode;
                                            start:
                                            /*MCM Tx and Rx Channel Select
                                            Register*/
/*Enable all 128 channels for Tx*/
/*****SPORT Initialization*****/
ax0=0xffff;
                                            io(SP1 MTCS0) = ax0;
                                            io(SP1 MTCS1) = ax0;
/*Disable interrupts*/
                                            io(SP1 MTCS2) = ax0;
  DIS int;
                                            io(SP1 MTCS3) = ax0;
  IRPTL = 0x0;
                                            io(SP1 MTCS4) = ax0;
  ICNTL = 0;
                                            io(SP1 MTCS5) = ax0;
  IMASK = 0;
                                            io(SP1 MTCS6) = ax0;
/*Set interrupt priorities*/
                                            io(SP1 MTCS7) = ax0;
  IOPG = Interrupt Controller Page;
                                            /*Enable all 128 channels for Rx*/
  ar = 0 \times b b 0;
                                            ax0=0xffff;
  io(IPR1) = ar;
                                            io(SP1_MRCS0) = ax0;
  ar = 0xbbbb;
```

#### EE-156

Technical Notes on using Analog Devices' DSP components and development tools Phone: (800) ANALOG-D, FAX: (781)461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

Page 5

io(SP1 MRCS1) = ax0; io(SP1 MRCS2) = ax0; io(SP1 MRCS3) = ax0; io(SP1 MRCS4) = ax0; io(SP1 MRCS5) = ax0; io(SP1 MRCS6) = ax0; io(SP1 MRCS7) = ax0; /\*SPORT MCM Configuration Reg 1\*/ ax0=0x01e1; io(SP1 MCMC1) = ax0;  $ax0=0x\overline{0}2ac;$ io(SP1 MCMC2) = ax0; /\* SPORT1 Interrupts Unmasked \*/ AX0=IMASK; AR=SETBIT 4 of AX0; IMASK=AR; /\*\*\*\*\*Autobuffer-based DMA Setup\*\*\*\*/ ax0 = 0x0010;io(SP1DR CFG) = ax0; ax1=page(sport rx buffer); io(SP1DR\_SRP) = ax1; ax1=address(sport\_rx\_buffer); io(SP1DR SRA) = ax1; ax1=length(sport rx buffer); io(SP1DR CNT) = ax1; ax0=0x0010; io(SP1DT CFG) = ax0; ax1=page(sport tx buffer); io(SP1DT SRP) = ax1;ax1=address(sport tx buffer); io(SP1DT SRA) =ax1; ax1=length(sport tx buffer); io(SP1DT CNT) = ax1; ax0 = 0x0013;io(SP1DR CFG) = ax0; ax0 = 0x0015;io(SP1DT CFG) = ax0; /\*\*\*\*\*\*\*Enable SPORT1\*\*\*\*\*\*/ ax0=io(SP1 RCR); ar=setbit 0 of ax0; io(SP1\_RCR) = ar; ax0=io(SP1 TCR); ar=setbit 0 of ax0; io(SP1 TCR) = ar; ena int;

```
wait_here: idle;
    jump wait_here;
```

EE-156

Page 6

Technical Notes on using Analog Devices' DSP components and development tools Phone: (800) ANALOG-D, FAX: (781)461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp