Engineer To Engineer Note EE-36

Notes on using Analog Devices' DSP, audio, & video components from the Computer Products Division
Phone: (800) ANALOG-D or (781) 461-3881, FAX: (781) 461-3010, EMAIL: dsp.support@analog.com

ADSP-21xx interface to the IOM-2 bus

Last Modified: 7/23/97

Introduction

The IOM Revision 2 (IOM-2) standard defines an industry standard serial bus for interconnecting telecommunications ICs. The serial interface satisfies the requirements of ISDN and analog applications and has been defined by four major European telephone equipment manufacturers.

The IOM-2 Bus

The IOM-2 bus is a synchronous full duplex communication link, containing user data, control/programming, and status channels and has been designed for terminal mode and line card mode. Both the line card mode and terminal portions of the IOM-2 standard utilize the same basic frame and clocking structure, but differ in the number and usage of the individual channels that are time-multiplexed over a four-wire serial interface. Data is clocked by a Data Clock (DCL) that operates at twice the data rate. A frame is delimited by an 8-kHz Frame Synchronization Clock (FSC). Data is transmitted over Data Upstream (DU) and Data Downstream (DD).

Line Card Mode

The line card version of the IOM-2 provides a connection path between line transceivers (ISDN) and codecs (analog), and the line card controller; the line card controller provides the connection to the switch backbone. In this mode up to 8 ISDN transceiver or up to 16 codecs/filters could be connected to the bus. Data, control and status information is multiplexed into frame, which are transmitted in an 8kHz rate. The frames are divided into 8 sub-frames, with one sub-frame being dedicated to each transceiver or pair of codecs.

Terminal Mode

The Terminal Mode is designed for ISDN and NT1 applications.

The IOM-2 terminal version consists of three subframes, each containing 32 bits. This 12 byte frame is repeated at 8kHz, giving an aggregate data rate of 768kbit/s.

ADSP-21xx Serial Port

The ADSP-21xx has one/two synchronous serial full-duplex 5 wire ports (SPORT) and one of these interfaces supports time-multiplexed data streams. Data is clocked by the Serial Clock (SCLK) that operates at the data rate. Optional there is a Frame Sync Signal for the receive & transmit side that indicates the start of a new word or frame in multichannel mode. Data is sent over Data Transmit (DT) and Data Receive (DR) lines. In normal operation an interrupt is generated when an word has been received or when a transmitted word has been shifted out of the SPORT. There are two independent interrupt vector locations to handle the receiving and transmitting section separately.

Autobuffering

The generation of an interrupt after each transmitted/ received word would cause a huge overhead and for this reason you could use an autobuffering mode that automatically transfers SPORT data to/from the internal memory of the DSP. For this application you have to define a circular buffer in the internal memory that should hold the data to be sent/received. The length of the buffer is user defined and when the buffer is full or has been read completely an interrupt will be generated and the buffer will be filled/ read again by the SPORT. This means we have to process the data in the buffer before the SPORT accesses this memory again.

Multichannel Mode

As already mentioned above one of the SPORTs of the DSP has the possibility to handle time-multiplexed data streams. It is possible to choose between 24 or 32 time slots. The user can define in which time slots he wants to send data and which time slots should be ignored during reception. This can be done independently for the receiving & and transmitting side. The autobuffering mode could also be combined with the MC mode. The best way to combine autobuffering and multichannel mode is to set the length of the circular autobuffer to the number of time slots that should be sent/ received. In the following

sections it is assumed that an external device is generating the SCLK and Frame Sync. Signals.
As the IOM-2 standard is using only one Frame Sync. Signal, RFS and TFS of the DSP have to be shorted.

IOM-2 Line-Card Mode Interfacing

Remember that the IOM-2 standard is using a clock that operates at twice the data rate. The ADSP-21xx family does not have a double serial clock feature and therefore we have to divide the DCL using a Flip Flop. In the Line-Card Mode we have 8 channels each consisting of 32 bits (256 bits total length). The number of DSP channels is set to 32 and therefore each DSP time slot is 1 Byte long (256 bits total frame length/32 time slots). The word length (SLEN) of the DSP is set to 7 and the DSP now interprets all incoming data as words of the length of one byte.

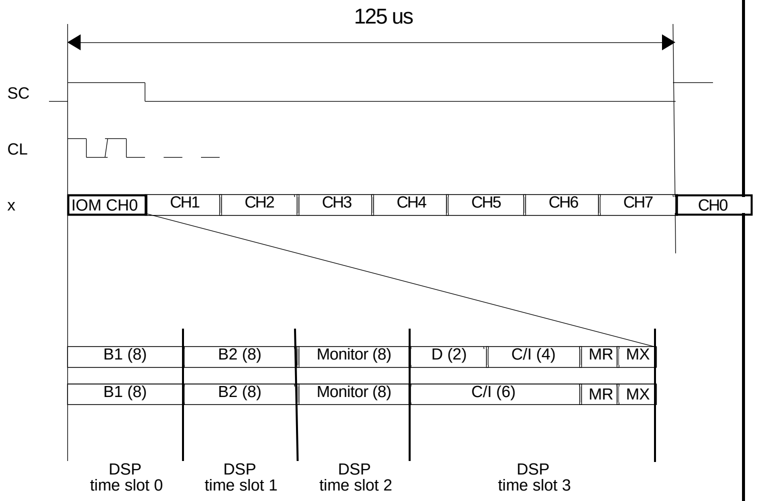

Figure 1: IOM-2 Line Card Frame Structure
DCL=4.096MHz

At last we must declare an autobuffer in the data memory (DM) of the DSP that is 32 locations deep as we are going to handle 32 time slots.

- MC enabled / 32 slots
- Autobuffering enabled
- SLEN = 7
- DTYPE = 00 (right justified MSBs zero filled)
- MCWS = FF FF (all time slots enabled)

The table below illustrates the content of the declared autobuffer.

IOM2	Addr	Autobuffer in DM Memory/16 bits wide D15 D8	D7 D0			
CH0	x00	0x00	B1 (8 bits)			
	x01	0x00	B2 (8 bits)			
	x02	0x00	Monitor (8bits)			
	x03	0x00	D2(2)	C/I(4)	MR(1)	MX(1)
CH1	x04	0x00	B1 (8 bits)			
	x05	0x00	B2 (8 bits)			
	x06	0x00	Monitor (8bits)			
	x07	0x00	D2	C/I	MR	MX
:		:	:			
CH7	x1c	0x00	B1 (8 bits)			
	x1d	0x00	B2 (8 bits)			
	x1e	0x00	Monitor (8bits)			
	x1f	0x00	D2	C/I	MR	MX

The Multichannel Word Select register could be used to ignore certain time slots. In this case you could implement a shorter autobuffer.

In the following example only the B1, B2 channels should be serviced and the monitor and control bits should be ignored.
The following settings could be used:

- MC enabled / 32 slots
- Autobuffering enabled
- SLEN = 7
- DTYPE = 00 (right justified MSBs zero filled)
- MCWS = CC CC (1st two time slots of each IOM channel)

In this case it is convenient to use a 16 (two B words * 8 channels) location deep autobuffer. This would generate an interrupt after the all B1 & B2 data of all 8 IOM channels has been sent/received and we have to process the content of the autobuffer before the SPORT is accessing again the autobuffer.
Remember: The buffer occupies only 16 address locations as we do not want to receive/transmit the monitor & control data.

IOM2	Addr	Autobuffer in DM Memory/16 bits wide D15 D0	
CH0	x00	0x00	B1 (8 bits)

	x01	0x00	B2 (8 bits)
		ignored	
		ignored	
CH1	x02	0x00	B1 (8 bits)
	x03	0x00	B2 (8 bits)
		ignored	
		ignored	
:		:	:
CH7	x0e	0x00	B1 (8 bits)
	x0f	0x00	B2 (8 bits)
		ignored	
		ignored	

IOM-2 Terminal Mode Interfacing

The IOM-2 terminal version consists of three sub-frames, each containing 32 bits (total length of 96 bits). Again we have the choice between 24 or 32 time slots. Using 32 time slots would mean that each word must be 3 bits long and using 24 time slots assumes 4 bit long words. This is not very comfortable as in this case the B1, B2 user data would be divided up into two nibbles. The best way to interface the terminal mode is to divide the Frame Synch,. Signal FSC of the IOM-2 interface by two and connect it to RFS/TFS of the DSP. In this case the DSP stores 6 IOM-2 channels in the autobuffer array. One DSP frame equals 2 IOM-2 frames = 2*96 bits. The DSP word length (SLEN-1) is 2*96/24 channels = 8 bit. This is again a format we could handle very easily. Remember: The DCL signal must still be divided as the DSP should not latch serial data twice. The following settings are used to access data:

- MC enabled / 24 slots
- Autobuffering enabled
- SLEN = 7
- DTYPE = 00
-MCWSx = 0xFF FF (enable all 24 time slots)

Remember: The IOM-2 frame is divided up into 3 subframes but we are going to receive 6 subframes consecutively. The content of the autobuffer could look like the following table:

IOM2	Addr	Autobuffer in DM Memory/16 bits wide D15 D8	D7 D0
CH0	x00	0x00	B1 (8 bits)
	x01	0x00	B2 (8 bits)
	x02	0x00	Mon0 (8bits)
	x03	0x00	D C/I0 MR MX
CH1	x04	0x00	IC1 (8 bits)
	x05	0x00	IC2 (8 bits)
	x06	0x00	Mon1 (8bits)
	x07	0x00	C/I1 MR MX
CH2	x08	0x00	
	x09	0x00	
	x0a	0x00	
	x0b	0x00	C/I2
CH0	x0c	0x00	B1 (8 bits)
	x0d	0x00	B2 (8 bits)
	x0e	0x00	Mon0 (8bits)
	x0f	0x00	D C/I0 MR MX
CH1	x10	0x00	IC1 (8 bits)
	x11	0x00	IC2 (8 bits)
	x12	0x00	Mon1 (8bits)
	x13	0x00	C/I1 MR MX
CH2	x14	0x00	
	x15	0x00	
	x16	0x00	
	x17	0x00	C/I2

Again we have the possibility to select certain time slots on the receiving and transmitting side independently. In The following example ignores control and monitor data and receives/sent B1 & B2 data.

In this case the time slots 0, 1, 12 and 13 which contain the B1 & B2 data should be enabled. The table below illustrates the content of the autobuffer in data memory that has a length of four.

IOM2	Addr	Autobuffer in DM Memory/16 bits wide D15 D8	D7 D0
CH0	x00	0x00	B1 (8 bits)
	x01	0x00	B2 (8 bits)
		0x00	ignored
		0x00	ignored
CH1		0x00	ignored
		0x00	ignored
		0x00	ignored
		0x00	ignored
CH2		0x00	ignored
		0x00	ignored
		0x00	ignored
		0x00	ignored
CH0	x02	0x00	B1 (8 bits)
	x03	0x00	B2 (8 bits)
		0x00	ignored

		0x00	ignored
CH1		0x00	ignored
		0x00	ignored
		0x00	ignored
		0x00	ignored
CH2		0x00	ignored
		0x00	ignored
		0x00	ignored
		0x00	ignored

IOM-2 Terminal Mode Interfacing II

The interface to the terminal mode could also be implemented without dividing the frame sync signal of the IOM-2 standard. In this case the incoming/outgoing data is divided up into nibbles and it is not that comfortable to handle this data. The 32 bit barrel shifter of the DSP could handle these nibbles and it takes two overhead cycles to receive/ transmit data via the IOM-2 interface. A subroutine to put together the nibbles to a 8 bit word could look like the following one:

In this case the subroutine IOM2_RX is used to build the 8 bit B1 & B2 data word of the nibbles that are stored in the autobuffer.
Assistance:

```
/*************************************************/
/* IOM2 interface to the ADSP-2181 SPORT0                  */
/* SIEMENS VIDEOPHONE                                      */
/* TDMA, 24 Slots, SLEN = 4                                */
/* autobuffering 4 words deep = 4 nibbles                  */
/* = 2 B channels used                                     */
/* CH0..CH3 enabled                                        */
/* SCLK = DCL / 2                                          */
/*************************************************/

.MODULE IOM2;
.VAR/DM/RAM         B1_RX;
.VAR/DM/RAM         B2_RX;

.GLOBAL             B1_RX;
.GLOBAL             B2_RX;

.ENTRY              IOM2_RX;
/*******************************************************/
/* subroutine IOM2_RX that should be called after the autobuffer   */
/* interrupt has occured                                           */
/*******************************************************/
IOM2_RX:

ar=dm(i1,m0);                 /* get high nibble of B1       */
sr0=dm(i1,m0);                /* get low nibble of B1        */
sr=sr or lshift ar by 4 (LO);  /* get high nibble of B2       */
dm(B1_RX)=sr0;

ar=dm(i1,m0);                 /* get high nibble of B2       */
sr0=dm(i1,m0);                /* get low nibble of B2        */
sr=sr or lshift ar by 4 (LO);
dm(B2_RX)=sr0;

RTS;
.ENDMOD;
```

IOM-2 Terminal Interfacing III

It is also possible to use no additional glue logic at all and this would mean that every data bit of the serial port is latched twice as we are now using the double clock of the IOM-2 standard. A routine is available that uses a lookup table and consumes additional 40 cycles per received 8 bit word. Please contact ANALOG DEVICES for further details or application assistance.

http://www.analog.com
ftp.analog.com

Literature:

[1] ADSP-2100 Family User’s Manual
[2] ICs for Communications
IOM-2 Interface Reference Guide