IOM-2 Interfacing on TMS320C54x

Literature Number: BPRA074
Texas Instruments Europe
February 1998

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Figures

IOM-2 Interfacing on TMS320C54x

ABSTRACT

This application report describes an interface design for connecting the synchronous serial port of the TMS320C54x DSP to an IOM-2 serial bus. The optional buffering feature of this peripheral gives an easy connection to all the different data, command and communication channels of the IOM-2 bus.

1. Introduction

Various types of Telecommunication end-equipment such as phones, answering machines, line cards and modems are becoming more and more intelligent. This means that most of the different elements or ICs contributing to the final functionality contain a certain amount of processing power. There is therefore an urgent need for flexible communication with other elements of the end-equipment to be able to manage and control all the different features of the device. To provide an efficient solution, four of the major European telephone equipment manufacturers jointly invented the IOM-Bus specially adapted to the future needs of the ISDN telecommunication end-equipment.(**I**SDN **O**riented **M**odular Interface Revision **2**)

This report shows how to interface the TMS320C54x DSP family to IOM-2 bus via the standard and/or buffered serial port. It includes the schematics of the required logic together with the software running on the DSP to read/write to/from different channels of the bus. Additionally, the test setup using a SIEMENS ISDN transceiver PEB2186 in IOM-2 loopback mode is described.

2. IOM-2 Serial Bus

The IOM-2 bus provides a symmetrical, full-duplex communication link, containing user data (B1, B2), control/programming (MON0, MON1, IC0, IC1) and status channels (D, CI0, MX, MR). The various channels are time-multiplexed over a four-wire serial interface. Data are clocked by DCL (Data Clock) that operates at twice the data rate. Frames are delimited by an 8kHz Frame Synchronization Clock (FSC). Data are transferred by Data Upstream (DU) and Data Downstream lines. Each frame contains several time-multiplexed sub-channels with different bandwidths from 2 - 8 byte per channel.

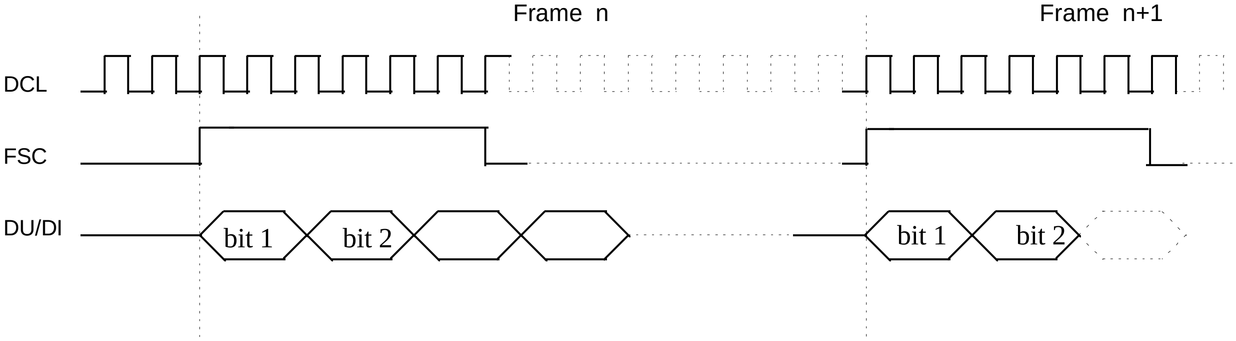


Figure 1: IOM-2 simplified timing diagram

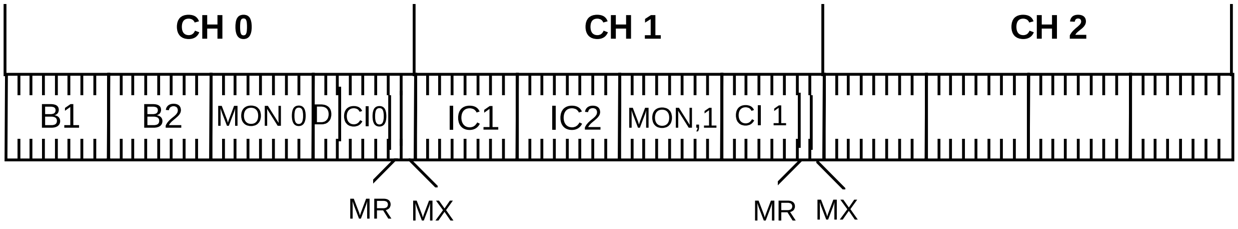


Figure 2: IOM-2 frame structure for terminal mode

The IOM-2 bus specification defines 2 different modes of operation: Terminal Mode and Line Card Mode. These modes differ in the frame structure and data rate. The frame rate remains at 8kHz for each. Figure 2 shows the channel and sub-channel structure of the Terminal Mode with 3 channels. In Line Card Mode there are 8 channels per frame with the sub-channel structure of CH0 in figure 2. The 8 channels with 32bit each in a 8kHz frame lead to a data rate of 2.048 Mbps instead of 768 kbps in Terminal Mode.

3. TMS320C54x Serial Port

The serial ports of the C54x DSP family are full duplex on-chip communication links providing a direct communication with serial devices such as CODECs or serial ADCs. There are three types of ports available: the Standard Serial Port (SSP), the TDM Serial Port and the Buffered Serial Port (BSP).

3.1 Standard Serial Port

This port is a 6-wire, full-duplex, synchronous port with 3 signals for each direction :

clock (CLKX/R)	frame synch (FSX/R)	data (DX/R)

There are 2 synchronization schemes available:

Burst Mode	Continuous Mode

In Burst Mode a frame synch is required to transfer every data word. In Continuous Mode only one frame synch at the beginning of the transfer is necessary to send and/or receive data words continuously without re-synchronisation. The word length may be set to 8bit or 16bit. Operations in both directions are double buffered, enabling a continuous communication with a data rate equal to the clock rate regardless of whether the clock signal and the frame synch signal are generated internally by the DSP or are supplied from an external source.

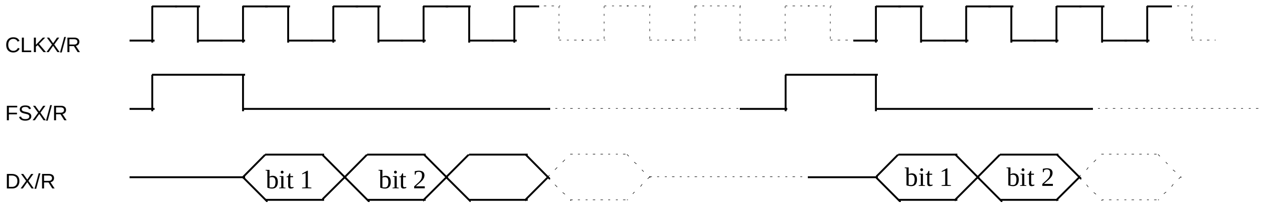


Figure 3: Simplified SSP burst mode timing

3.2 Buffered Serial Port

The BSP is an extension of the SSP which allows the connection of the IO register of the SSP to the internal memory of the DSP. By means of an Automatic Buffering Unit the transmit or receive data are stored in a maximum of two circular buffers in the memory without CPU intervention. This reduces the CPU load quite significantly.

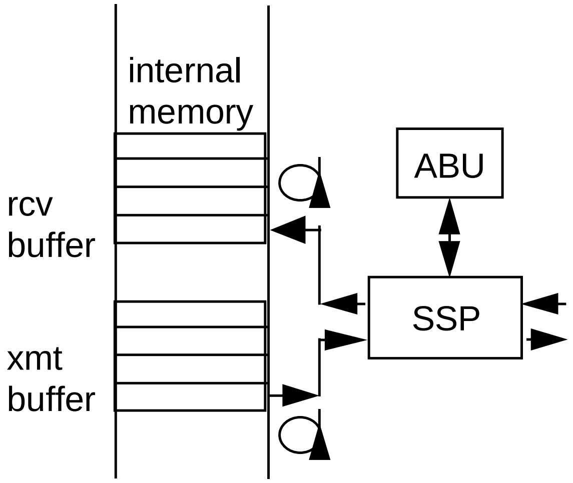


Figure 4: Simplified BSP block diagram

3.3 TDM Port

The Time Division Multiplex Port is again a special extension to the SSP, establishing a time-slot scheme on the serial bus to facilitate the communication between multiple DSPs. There are 8 time-slots available, allowing the glueless connection of up to 8 DSPs which can than communicate in a very flexible way, point-to-point or broadcast.

4. IOM-2 Interfacing

At first glance, the IOM-2 bus and the serial port timings seem to be very similar. Indeed, there are only small differences. These concern frame synchronization and data clock signal.

4.1 Serial Interface Characteristics

IOM-2: The rising edge of the frame signal (FSC) occurs together with the first data bit of every frame.

The FSC width is longer than one clock cycle

The data clock DCL is twice the frequency of the bit transfer rate

SSP: The rising edge of the frame signal (FSX/R) occurs one clock cycle before the first data bit of every transfer

The FSX/R width is one clock cycle

The clock rate CLKX/R is equal the frequency of the bit transfer rate

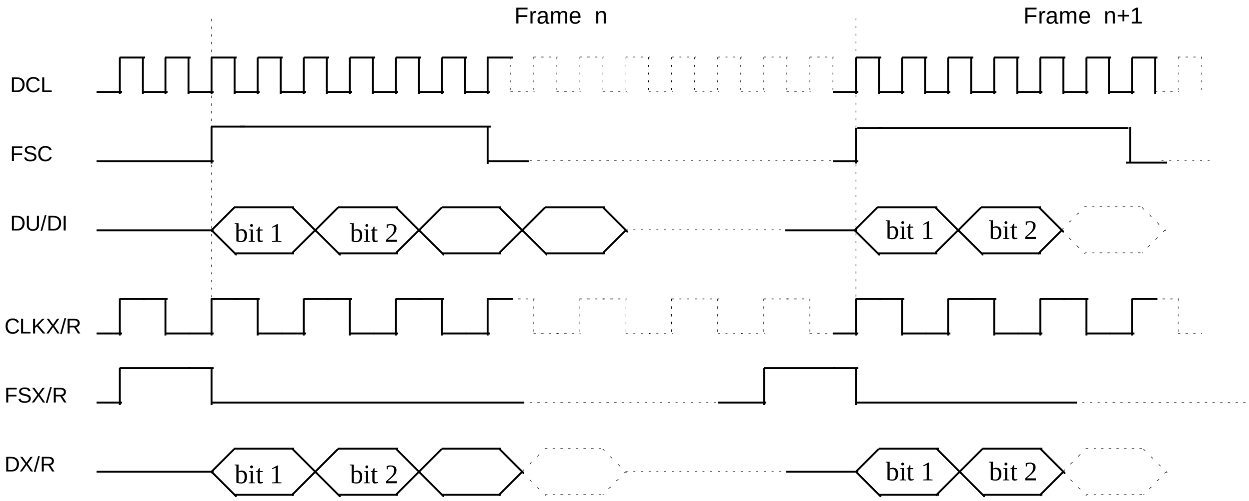


Figure 5: IOM-2 / C54x SSP timing

4.2 Interface Adaptation

To connect the SSP to the IOM-2 bus successfully there are 3 things to do:

1. Divide the DCL by 2.
2. Shift FSC
3. Shorten FSC

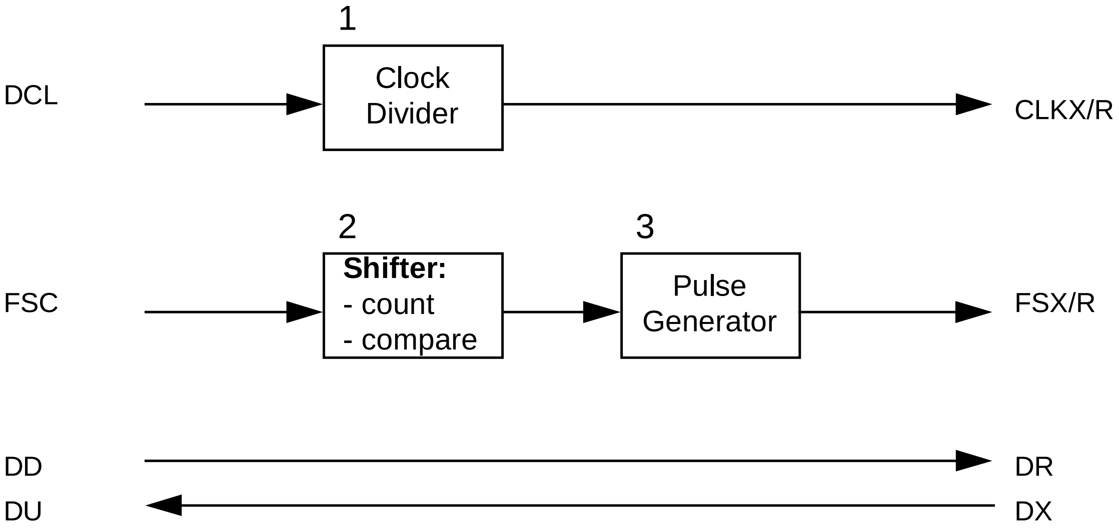


Figure 6: Interface Logic Block Diagram

The Shifter can be implemented by a counter starting with FSC counting the number of DCL cycles we want to shift. This number can be decoded by means of a PAL device which then resets the counter after getting the next FSC pulse. To generate the required FSX/R pulse with a width of 2 DCL cycles, two adjacent counter values activate the FSX/R signal. The only signal missing now is the CLKX/R, with half the cycle time of DCL. This signal can easily be derived from the first output bit of the counter which is counting DCL cycles. The 8-bit counter must provide a synchronous RESET.

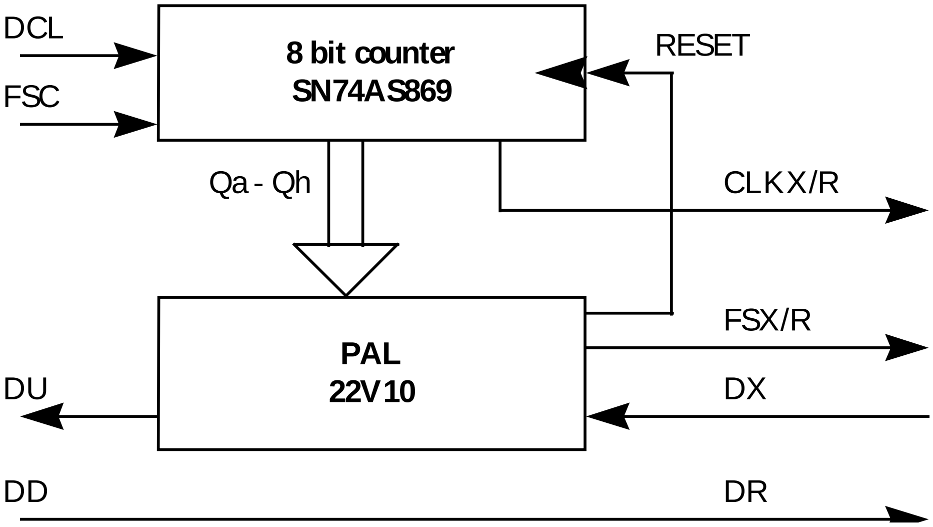


Figure 7: Interface Logic Implementation

RESET = FSC & Qh & Qg

FSX/R = (counter == (190 OR 191))

To access all 12 TDM byte-wide sub-channels of the IOM-2 bus in Terminal Mode, the FSX/R synch pulse has to be generated 12 times, activated by the corresponding timer values (190-i*16 OR191-i*16, 0<i<12).

In this case, the use of the BSP would facilitate IOM-2 access quite dramatically. Instead of providing 12 interrupts every frame transmit and receive buffers with the size of 12 words, each is served automatically by the ABU - generating only one interrupt per frame to read and update the data inside the buffers. To access specific channels out of the 12, you need only read/write from/to the corresponding memory location inside the buffers.

Running in Line Card Mode would increase the number of byte-wide sub-channels from 12 to 32 which practically forces the use of the BSP in order to avoid heavy DSP interrupt loading. The timer values to create the 32 pulses, accessing all 32 sub-channels, are derived by an equation similar to that used for the 12 sub-channels; just increasing the range of i from 12 to 32 and adjusting the number of DCL cycles per frame to 510 due to the higher data rate of 2.048 Mbps in the Line Card Mode. (510-i*16 OR511-i*16, 0<i<32)

The above interface logic implementation allows the realization of the aforementioned functions. In addition, the DX-DU signal is put through the PAL device so that it may control the output state of DU bitwise by means of a tri-state buffer. The state of DX can only be changed byte by byte, using just the SSP which does not permit writing into TDM channels smaller than one byte from more than one DSP without generating bus conflicts.

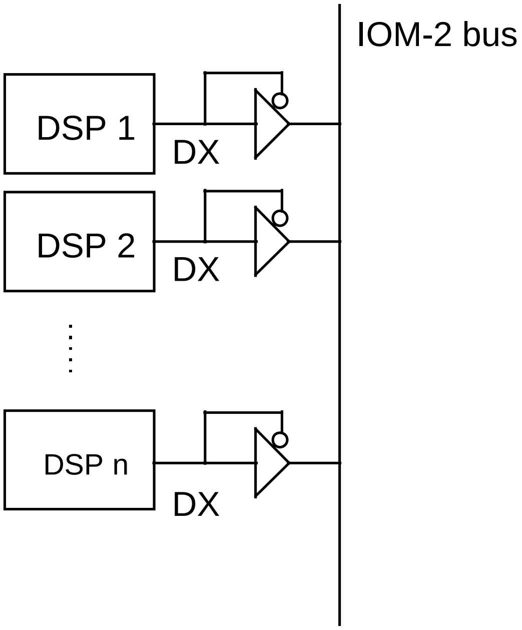


Figure 8: Bit Multiplexing

An additional latch in the FSX/R signal path avoids any glitches. This introduces one waitstate. Therefore the equation for FSX/R must be changed to

FSX/R = (counter == (189 OR 190))

and

FSX/R = (counter == (190-i*16 OR191-i*16, 0<i<12))

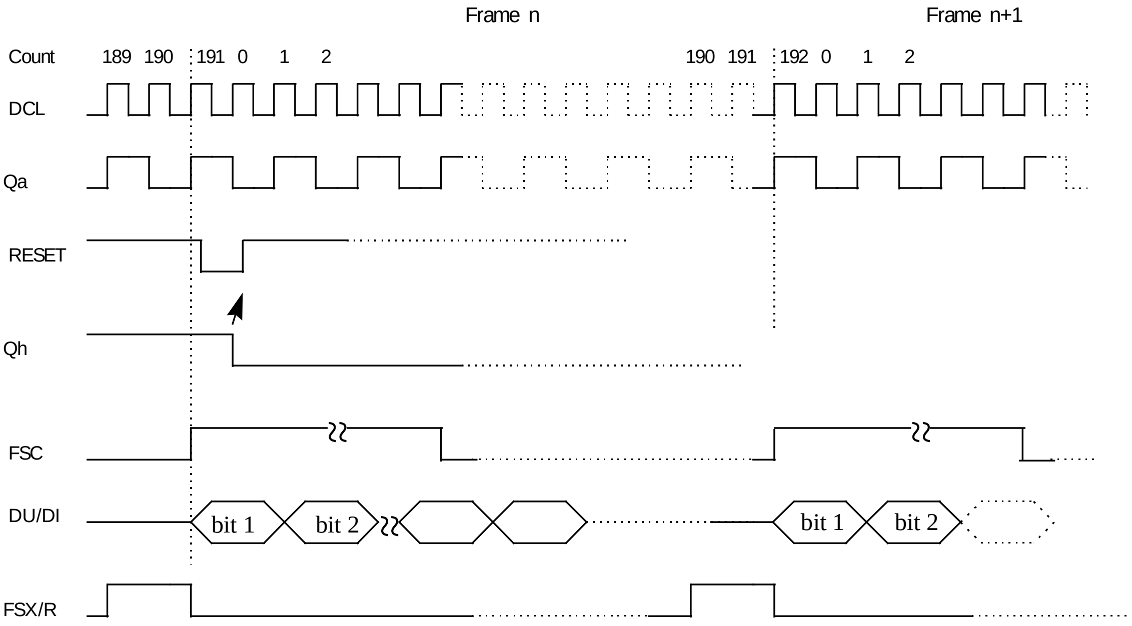


Figure 9: Interface Logic Timing

To meet all timing requirements the falling edge of DCL triggers the counter.

5. Test System

To verify the functionality of the interface implementation described in the previous chapter, a small hardware platform was built containing a TMS320C543 DSP, the described logic elements and a SIEMENS ISDN Transceiver with IOM-2 interface. As SIEMENS is the main user of this interface it was quite natural to take the PSB2186 as the IOM-2 test device. The DSP generates a test pattern and sends it on a particular IOM-2 channel running in Terminal Mode. The ISDN transceiver runs in loopback mode, sending the same data back on the receive part of the serial port. By comparing input and output data in the DSP we can verify that the interface operates correctly.

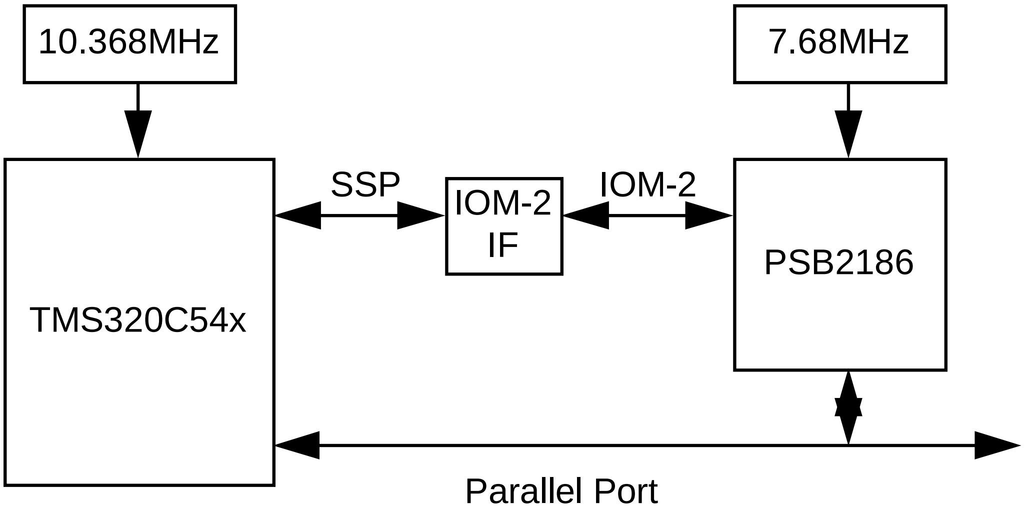


Figure 10: Test Setup Block Diagram

5.1 TMS320C543

The TMS320C543 is a RAM-based member of the ’C54x family with 10K words of on chip RAM. This device has 2 serial ports, 1 TDM port and 1 BSP. The BSP will be used for the IOM-2 connection in order to have the flexibility of selecting either SSP- or BSP-mode for test purposes.

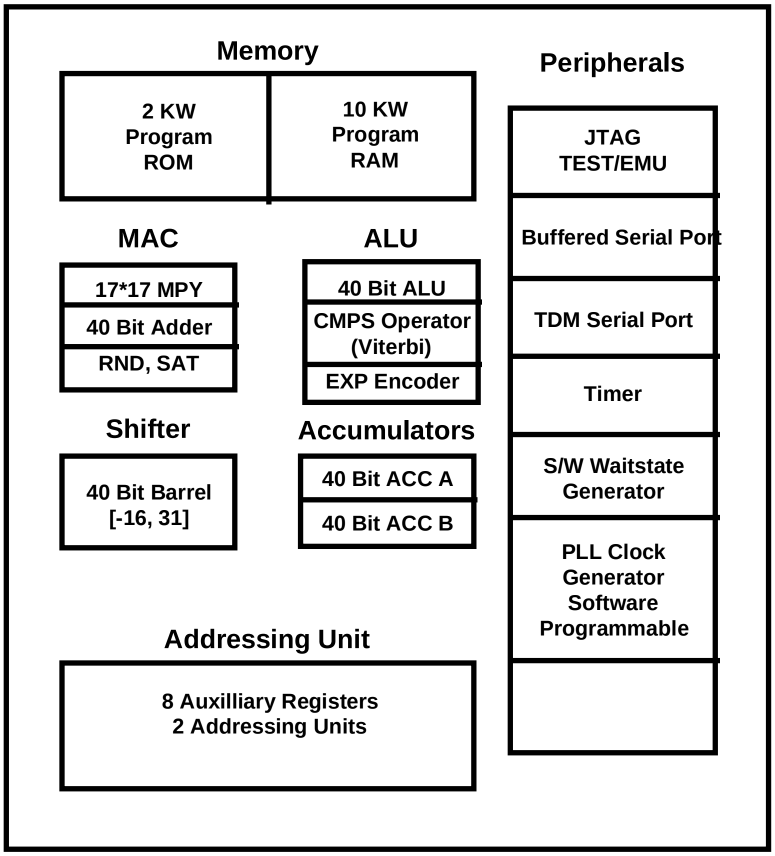


Figure 11: TMS320C543 Block Diagram

5.2 ISDN Transceiver PSB2186

The PSB2186 implements a 4-wire S/T interface used to link voice/data terminals to ISDN. This device switches B- and D- channels between the S/T side and the IOM interface where other devices - such as a DSP to handle voice compression, modem signals or answering machine functions - could be connected for further processing,. The setup and control of the PSB2186 is performed by the DSP connected via the parallel port to the microprocessor interface of the PSB2186. Simply by means of software this allows a flexible access to all the functions and features of this ISDN device. The microprocessor interface is configured for Motorola non-multiplexed bus type giving glueless access to the DSP external bus.

5.3 Test Software

After power-up reset both the DSP and the PSB2186 devices have to be initialized properly. The DSP has to set up the serial port to external clock and frame synch running in burst mode.

SPC[MCM] = 0 **SPC[FSM] = 1** **SPC[FO] = 1**

The PSB2186 is running in IOM-1 mode after reset and has to be switched to IOM-2 interface mode. In a second step, the ISAC-TE has to be set to test-looping of the IC1 channels. This allows the DSP to send data on a selected channel (IC1) and to read this data for verification in the next frame.

ADF2[IMS] = 1

SPCR[TLP] = 1	**SPCR[SPU] = 1**
SPCR[C1C0] = 1	**SPCR[C2C0] = 1**

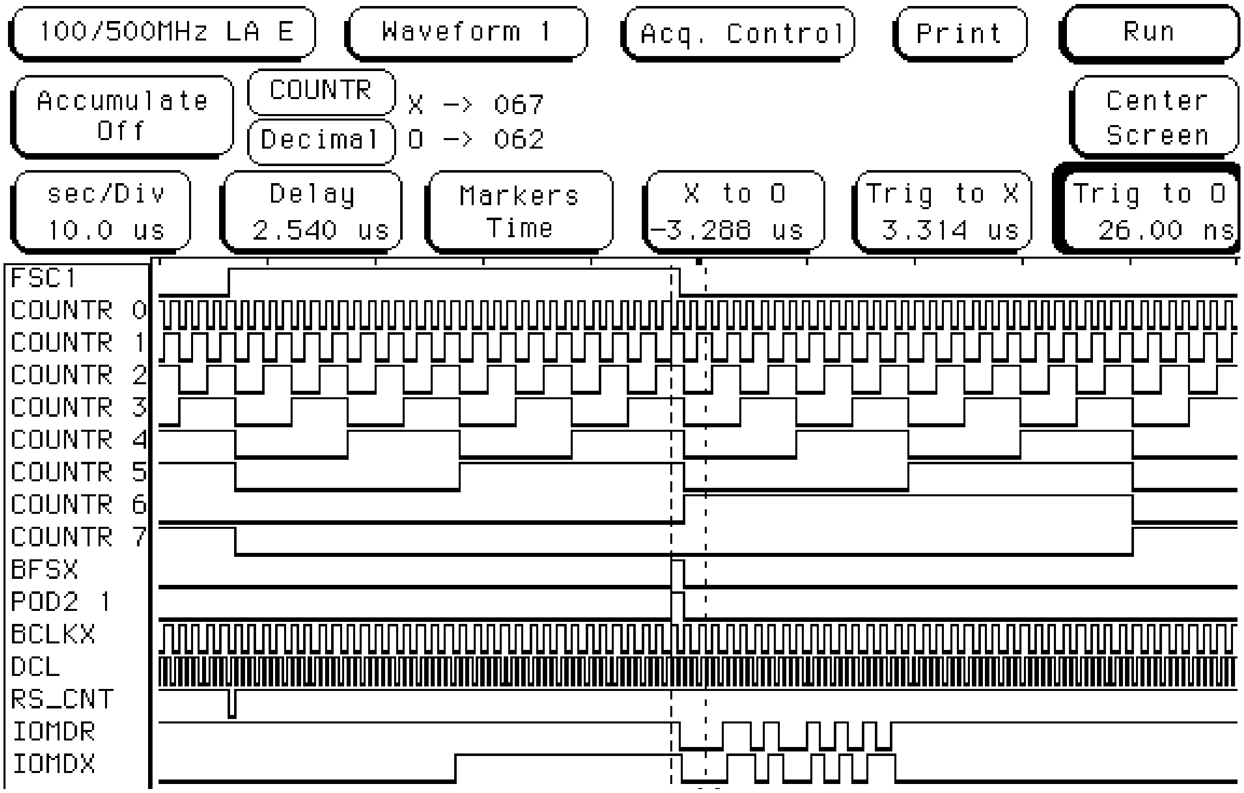
These registers can be modified by the DSP writing to the corresponding memory addresses, mapped into the IO space of the DSP.

After initialization, the DSP just waits for interrupts from the serial port to receive and send data to the ISAC-TE. The received data is then compared with the data sent one frame before to check if the loopback-transfer works. By means of the BIO pin it is possible to shut off the error check, making the startup procedure of the system easier.

5.4 Results

The test verifies the functionality of the IOM-2 IF logic described and gives a good start up example which can easily be adapted to a certain application. Simply by making small modifications of the PAL equations, all kind of different IOM2- access schemes can be achieved.

The figure below show the logic analyzer printout for the key signal of the test system.



Accumulate Off

COUNTR Decimal X -> 067

O -> 062

sec/Div 10.0 us

Delay 2.540 us

Markers Time

X to O -3.288 us

Trig to X 3.314 us

Trig to O 26.00 ns

Figure 12: Waveform Diagram

6. Summary

The interface logic described in this document demonstrates the feasibility of connecting the TMS320C54x DSP to all channels of an IOM-2 bus. Using the BSP feature of the DSP adds more flexibility concerning multiple channel access and reduces the interrupt load of the processor. The IOM-2 protocols are not covered by this report because they can be fully implemented by software alone.

The two devices added to the serial port of the TMS320C54x implement a quite flexible interface. For a more cost effective solution it is also possible to include the 8-bit counter in the 22V10 PAL.

References

1. ICs for Communications, IOM-2 Interface Reference Guide, Siemens AG 1991

2. TMS320C54x DSP Reference Set Volume 1-4, Texas Instruments,1996

3. ICs for Communications, ISDN Subscriber Access Controller for Terminals, ISAC-STE, PSB2186, User’s Manual 10.94

Appendix A Schematic Diagrams

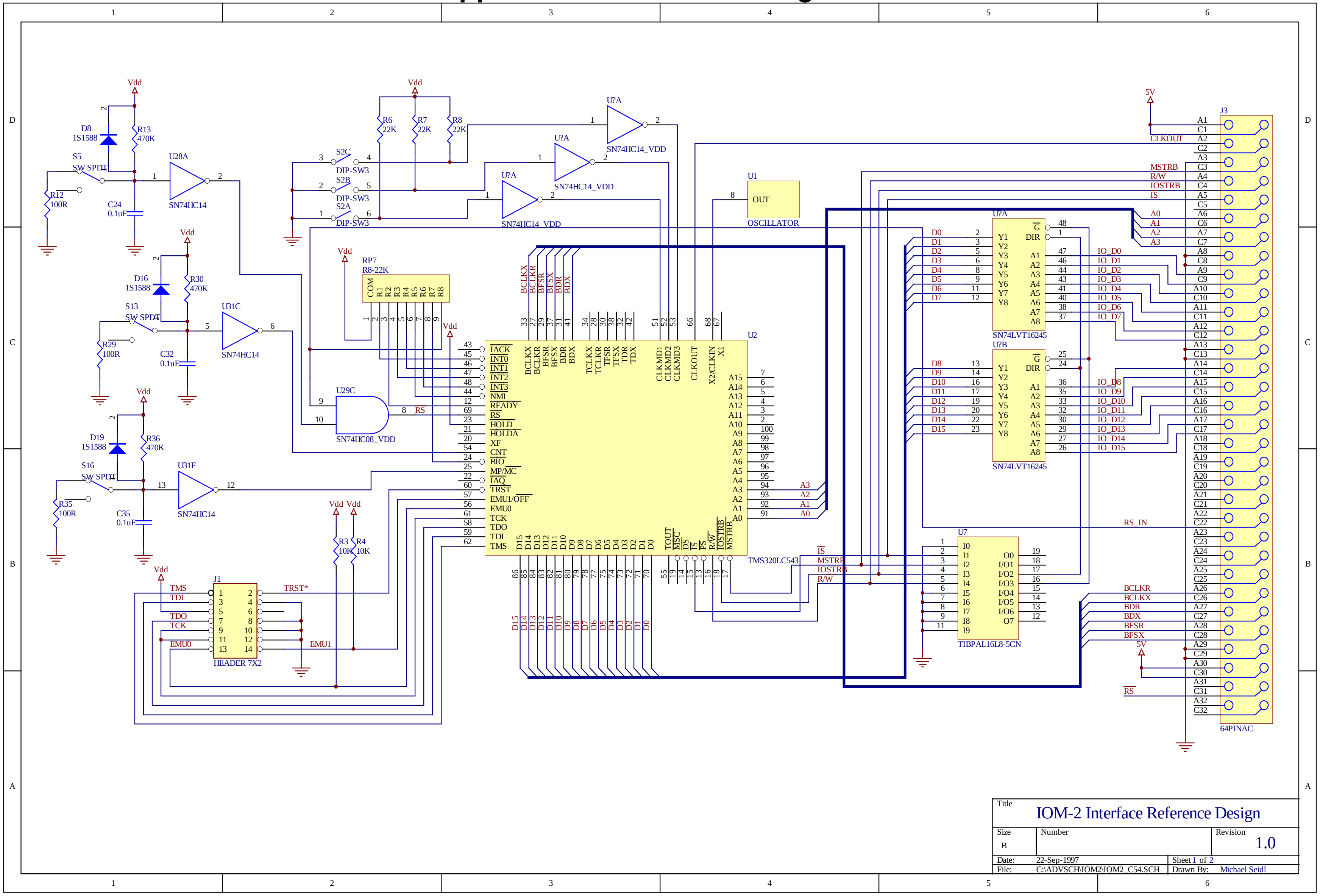


Figure 13: Schematic Test System 1 of 2

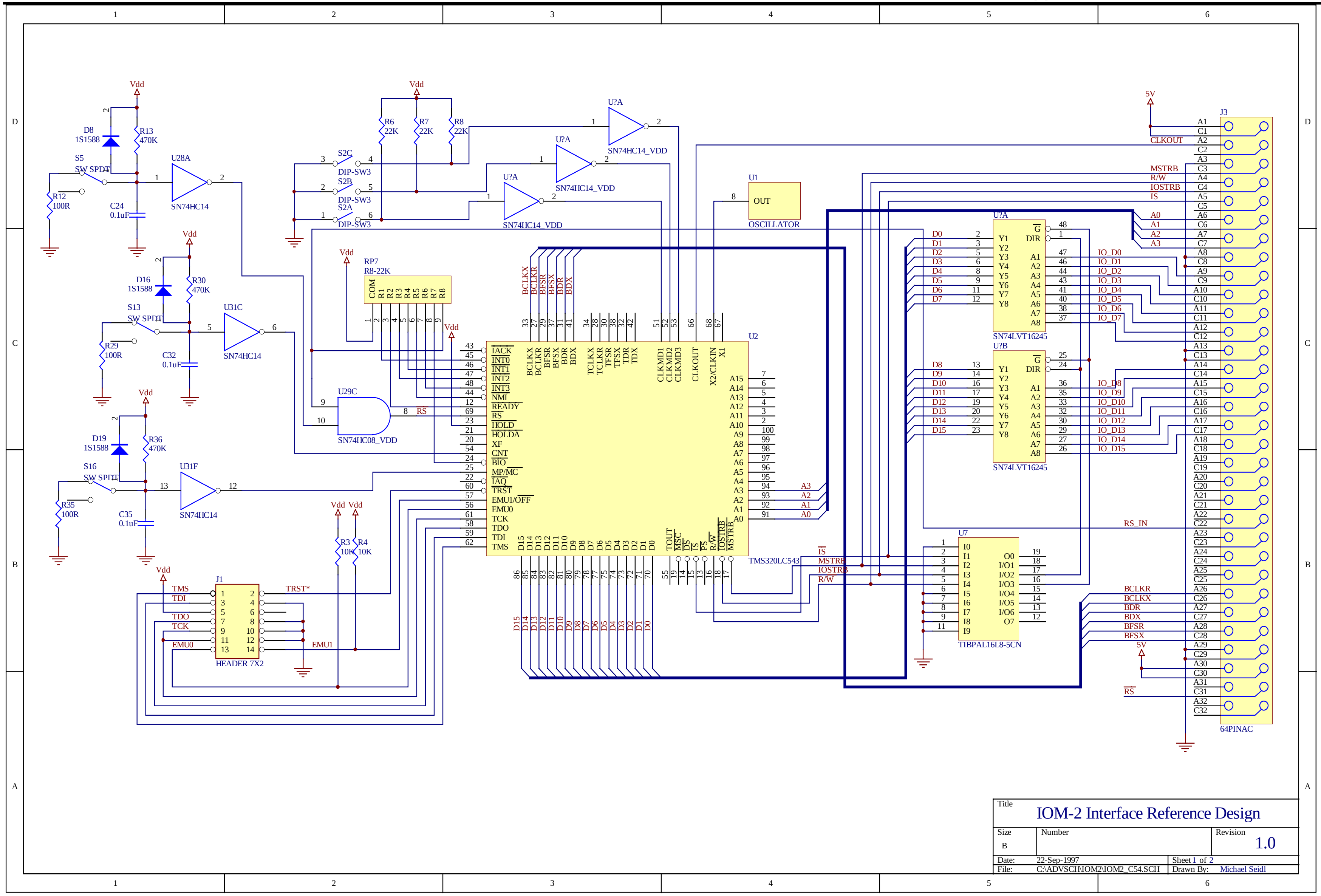


Figure 14: Schematic Test System 2 of 2

Appendix B PAL Equations

```
" logic connecting c54x’ serial port to IOM-2 bus


MODULE module_name
    iom2_ic1  DEVICE ’p22v10’;

"INPUTS
    CLK,DCL,QA,QB,QC,QD,QE      PIN 1,2,3,4,5,6,7;
    QF,QG,QH,FSC1,BDX           PIN 8,9,10,11,13;
    IOMDR                       PIN 14;


"OUTPUTS

    RESET_CNT                   PIN 23;
    BCLKX                       PIN 22;
    BDR                         PIN 21;
    IOMDX                       PIN 20;
    BFSX                        PIN 19;
    FSX_INT                     PIN 18;
    CLK_CNT                     PIN 17;
    X,C,Z=.X.,.C.,.Z.;


EQUATIONS

"Reset of counter with rising edge of FSC1

!RESET_CNT = FSC1 & QH;

"invert CLK for counter

!CLK_CNT = DCL;

"Frame sync pulse @ 61,62 on IC1
"            128    64   32   16    8    4    2     1
!FSX_INT =   (!QH & !QG & QF & QE & QD & QC & QB & !QA)
           # (!QH & !QG & QF & QE & QD & QC & !QB & QA);

BFSX.clk = CLK;

!BFSX := FSX_INT;
```

```
"bitenable for data signals

!BDR = !IOMDR;

!IOMDX = !BDX;

"serial data clock to c54x = A inverted

!BCLKX = QA;


TEST_VECTORS ([QA] -> [BCLKX])

              [ 0 ] -> [ 1 ];
              [ 1 ] -> [ 0 ];

TEST_VECTORS ([DCL] -> [CLK_CNT])

              [ 0 ] -> [ 1 ];
              [ 1 ] -> [ 0 ];

TEST_VECTORS ([BDX] -> [IOMDX])

               [ 0 ] -> [ 0 ];
               [ 1 ] -> [ 1  ];


TEST_VECTORS ([IOMDR] -> [BDR])

              [ 0 ]   -> [ 0 ];
              [ 1 ]   -> [ 1 ];

TEST_VECTORS ([FSC1,QH] -> [RESET_CNT])

              [0, 0] -> [1 ];
              [0, 1] -> [1 ];
              [1, 0] -> [1 ];
              [1, 1] -> [0 ];


TEST_VECTORS ([CLK,QA,QB,QC,QD,QE,QF,QG,QH] -> [FSX_INT,BFSX])

              [C,  0, 0, 0,  0, 0, 0, 0, 0] -> [ 1 , 0 ];
              [C,  0, 0, 1,  0, 0, 0, 0, 0] -> [ 1 , 0 ];
```

```
                        [C,  1, 0, 0,  0, 0, 0, 1, 0] -> [ 1 , 0 ];
                        [C,  1, 0, 0,  1, 0, 0, 0, 0] -> [ 1 , 0 ];
                        [C,  0, 1, 0,  1, 0, 1, 0, 1] -> [ 1 , 0 ];
                        [C,  0, 1, 1,  1, 1, 1, 1, 1] -> [ 1 , 0 ];
                        [C,  0, 1, 1,  1, 1, 1, 0, 1] -> [ 1 , 0 ];
                        [C,  1, 0, 1,  1, 1, 1, 0, 1] -> [ 1 , 0 ];
                        [C,  0, 1, 1,  1, 1, 1, 0, 0] -> [ 0 , 1 ];
                        [C,  1, 0, 1,  1, 1, 1, 0, 0] -> [ 0 , 1 ];

END module_name
```

Appendix C System Test Software

```
;/*   FILENAME:       iom2test.asm                                 */
;/* -----------------------------------------------------------*/
;/*   TITLE:         TMS320C543 IOM-2 Interface Test SW        */
;/*                                                            */
;/*   PURPOSE:       This code writes and reads data           */
;/*                  via the serial port of the DSP to/from the*/
;/*                  ISAC TE chip to test the IOM-2 IF logic   */
;/*                  inbetween those devices                   */
;/*                                                            */
;/*------------------------------------------------------------*/
;/*   DATE:          09/30/97                                  */
;/*   REV.:          1.0                                       */
;/*   AUTHOR:        Markus Tremmel                            */
;/* -----------------------------------------------------------*/

 .version 543
 .mmregs
 .global start

;-------------------------
; Interrupt Vector Table
;-------------------------
 .sect "vectors"
reset:   B   start
         NOP
         NOP
nmi:     RETE
         NOP
         NOP
         NOP
 .space 14*4*16  ; software interrupts
int0:    RETE
         NOP
         NOP
         NOP
int1:    RETE
         NOP
         NOP
         NOP
int2:    RETE
         NOP
         NOP
         NOP
```

```
tint:    RETE
         NOP
         NOP
         NOP
brint0:  B rcvxmt_int
         NOP
         NOP
bxint0:  RETE
         NOP
         NOP
         NOP
trint:   RETE
         NOP
         NOP
         NOP
txint:   RETE
         NOP
         NOP
         NOP
int3:    RETE
         NOP
         NOP
         NOP

;-----------------------------
; IOM-2 Test SW for TMS320C54x
;-----------------------------
  .text

start:

  ;----------
  ; DSP init
  ;----------

  ; disable interrupts globally

  SSBX  INTM
  SSBX  XF                  ; set XF flag connected to A4 of ISAC

  ; set OVLY =1, MP/MC = 1, IPTR = 0x0080

  STM   #0x00e0, PMST

  ; setup interrupt register  enable BRINT
  STM   #0x010, IMR         ; enable receive interrupt

  ; clear accu B

  LD   #0,A
  LD   #0,B
  STM  #0x0,DXR
  STM  #0,AR0
  STM  #0,AR1
  STM  #0,AR2
  STM  #0,AR3
  ;-------------------------
  ; Standard Serial Port Init
  ;-------------------------

  ; set burst mode FSM = 1, ext.clk MCM = 0 and frame TXM = 0,
  ; 8bit mode FO = 1
  STM   #0x8008, SPC

  ; clear pending interrrupts
  STM   #0xffff, IFR

  ; bring SSP out of reset
  ; STM   #0x80c8, SPC


  ;--------------------
  ; initialize ISAC-S TE
  ; --------------------

SPCR_VAL1     .set   0x85    ; loopback mode enabled/IC1 looping
SPCR_VAL2     .set   0x05    ; loopback mode enabled/IC1 looping
ADF2_VAL      .set   0x88    ; IOM-2 mode and tristate drivers
SPCR_MEM             .usect  "isac", 1
ADF2_MEM        .usect       "isac", 1
SPCR_REC             .usect  "isac", 1
ADF2_REC             .usect  "isac", 1

  ; set ISAC-S TE ADF2 reg. in IOM-2 mode
  ST    #ADF2_VAL,ADF2_MEM
  PORTW ADF2_MEM, 0x8039
  NOP        ; wait
  NOP
  NOP
  NOP

  ; set ISAC-S TE SPCR in testmode  with soft power up
  ST    #SPCR_VAL1,SPCR_MEM
  PORTW SPCR_MEM, 0x8030


  NOP   ; wait
  NOP
  NOP
  NOP

  ;set ISAC-S TE SPCR in testmode without soft power up
  ST    #SPCR_VAL2,SPCR_MEM
  PORTW SPCR_MEM, 0x8030

  NOP                           ; wait
  NOP
  NOP
  NOP
  PORTR 0x8039, SPCR_REC     ; read registers of ISAC-S TE
                             ;for verification
  NOP
  NOP
  NOP
  NOP

  PORTR 0x8030, ADF2_REC

  NOP
  NOP
  NOP
  NOP

  ; enable interrupts globally
  STM   #0x80c8, SPC
  RSBX  INTM

 ; idle, just wait for the rcvxmt interrupt
wait:  NOP
  NOP
  NOP
  B wait
```

```
;-----------------------------
; receive and transmit interrupt
; service routine
;-----------------------------

rcvxmt_int:

        LD   DRR,A           ; read input data
        BC   OVFL, ANEQ      ; count in AR3 each counter overflow
        LD   AR3, B          ; increment overflow counter
        ADD  #1, B
        STLM B, AR3

OVFL:   SUB  AR1, A  ; compare received and sent
        ADD  #1, A   ; new received must be equal previous sent
        AND  #0xffff,A
        BC   OK, AEQ         ; branch if equal
        BC   OK, NBIO        ; ignore error if BIO = high
        LD   AR2, B          ; or increment error counter
        ADD  #1, B
        STLM B, AR2
        NOP
OK:     LD   AR1, A          ; reload previous sent
        ADD  #1, A           ; increment value to be sent
        STLM A, DXR          ; write output data
        STLM A, AR1          ; keep sent value in AR1

        STM   #0xffff, IFR   ; clear pending interrrupts
        NOP
        NOP
        NOP
        RETE


  .end
```

Appendix C Glossary

SSP:	Standard Serial Port
BSP:	Buffered Serial Port
TDM:	Time Division Multiplex
ABU:	Automatic Buffering Unit
IO:	Input Output
CODEC:	Coder Decoder