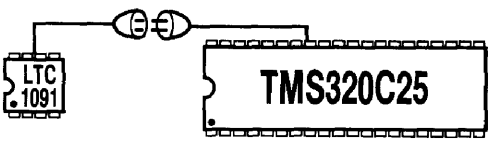
Interfacing the LTC1091/92 to the TMS320C25 DSP



Guy Hoover

Introduction

This application note describes the hardware and software required for communication between the LTC1091 10-bit data acquisition system and the TMS320C25 digital signal processor (DSP). In particular two interfaces will be demonstrated. The first interface requires only one inverter in addition to the LTC1091 and the TMS320C25. The second interface, which is optimized for speed of transfer and minimum processor supervision, can complete a conversion and shift the data in only 32μs. Configuration of the TMS320C25 and LTC1091 will be discussed as it applies to this interface as well as the minor modifications required for the interface to work with the LTC1092. Schematics, code, and timing diagrams will be presented. Finally, a summary of results including data throughput rates will be provided.

Interface Details

The LTC1091 clock line controls the A/D conversion rate and the data shift rate. Data is transferred in a synchronous, half duplex format over D_{IN} and D_{OUT}. The serial port of the TMS320C25 is not directly compatible with that of the LTC1091. The data shift clock lines (CLKR, CLKX) are inputs only. Therefore the data shift clock must be externally generated. Inverting the shift clock is also necessary because the LTC1091 and the TMS320C25 clock data on opposite edges. This prevents a race condition.

The schematic of Figure 1 has the shift clock generated by the XF pin of the TMS320C25. The signal is inverted with a 74HC04 and fed into the CLK pin of the LCT1091. The framing pulse of the TMS320C25 is fed directly to the $\overline{CS}$ of the LTC1091. DX and DR are tied directly to D_{IN} and D_{OUT} respectively. This method results in the simplest hardware configuration but has the drawbacks of requiring more processor supervision and a slower data shift time.

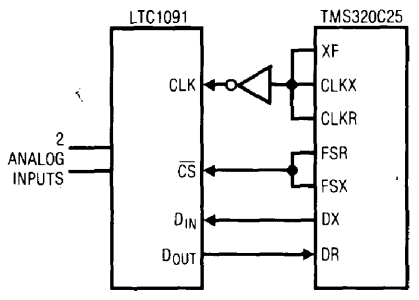


Figure 1. Circuit 1: Minimum Hardware Interface

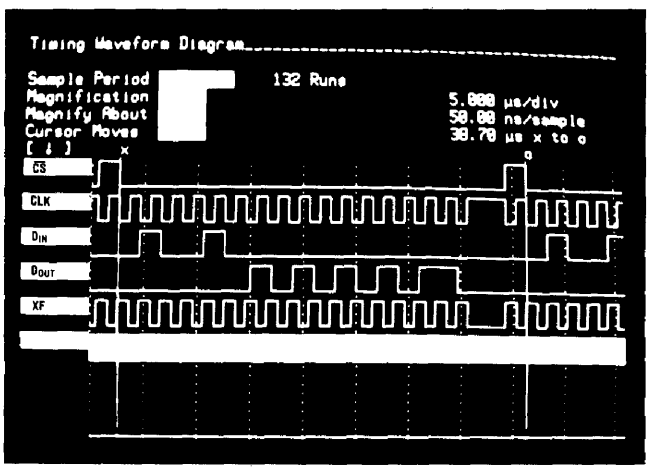


Figure 2. Timing for Circuit 1 Shows 39μs Throughput Rate

The schematic of Figure 6 has the clock generated by dividing down the CLK OUT pin by a factor of 16 with a 74HC163 counter. The signal is inverted with a 74HC04 and fed into the CLK pin of the LTC1091. The $\overline{CS}$ signal is generated directly from the FSX pin of the TMS320C25. The obvious disadvantage of this method is that it requires considerably more hardware to implement but it provides a faster data shift rate and requires less processor supervision.

Hardware Description

The DSP was emulated and the code for this interface was developed on a TMS320C25 Software Development System (SWDS). The SWDS requires a PC compatible computer to run.

The timing diagram of Figure 2 was obtained using the circuit of Figure 1. The timing diagram of Figure 7 was obtained using the circuit of Figure 6. Both pictures were taken with an HP1631 logic analyzer. The CLK was 500kHz for the timing diagram of Figure 2 and 625kHz for the timing diagram of Figure 7. The TMS320C25 clock rate was 40MHz.

The analog sections of the schematics of Figure 1 and Figure 6 are omitted for clarity. For a complete discussion of the analog considerations involved in using the LTC1091 please see the data sheet.

B15							B8
0	1 START	0 S/D	0 O/S	1 MSBF	0	0	0

Figure 3. D_{IN} Word in ACC of TMS320C25 for Circuits 1 and 2

MSB														LSB	
X	X	X	X	X	X	9	8	7	6	5	4	3	2	1	0	>200

D_{OUT} from LTC1091 stored in TMS320C25 RAM

Figure 4. Memory Map for Circuits 1 and 2

```
LABEL    MNEMONIC          COMMENTS
         AORG    0         ON RST CODE STARTS AT 0
         B       INIT      BRANCH TO INIT ROUTINE

         AORG    >26       ADDRESS OF RINT VECTOR
         B       RINT      BRANCH TO RINT ROUTINE

         AORG    >32       MAIN PROGRAM ADDRESS
INIT     DINT              DISABLE INTERRUPTS
         LDPK    >0        DATA PAGE POINTER IS 0
         LARP    >1        AUX. REG. POINTER IS 1
         LRLK    AR1,>200  AUX. REG. 1 = >200
         LACK    >10       CONFIG. WORD FOR IMR
         SACL    >4        PUT CONFIG. WORD IN IMR
         STXM              CONFIGURE FSX AS OUTPUT
         FORT    0         SET SERIAL PORT TO 16 BITS
         RXF               RESET XF

TXRX     LACK    >48       LOAD D_IN WORD INTO ACC
         SFSM              FSX STARTS ON XSR LOAD
         RPTK    7         REPEAT 8 TIMES
         SFL               SHIFT D_IN TO LEFT
         SACL    >1        D_IN PUT IN TX REGISTER
         EINT              ENABLE INTERRUPTS

TIMER    SXF               SET XF (CLK)
         RPTK    5         REPEAT 6 TIMES
         NOP               TIMING
         RXF               RESET XF (CLK)
         RPTK    3         REPEAT 4 TIMES
         NOP               NOP FOR TIMING
         B       TIMER     CLKS UNTIL RINT

RINT     ZALS    >0        STORE D_OUT WORD IN ACC
         SACL    *,0       STORE ACC IN >200
         B       TXRX      GO TO TXRX ROUTINE
         END
```

Figure 5. TMS320C25 Code for Circuit 1

Software Description

The software configures and controls the serial port of the TMS320C25.

The first 13 lines of code are the same for circuit 1 and circuit 2. The code first sets up the interrupt and reset vectors. On reset the TMS320C25 starts executing code at the label INIT. Upon completion of a 16-bit data transfer, an interrupt is generated and the DSP will begin executing code at the label RINT.

Next, the code initializes registers in the TMS320C25 that will be used in the transfer routine. The interrupts are temporarily disabled. The data memory page pointer register is set to zero. The auxiliary register pointer is loaded with one and auxiliary register one is loaded with the value 200 hexadecimal. This is the data memory location where the data from the LTC1091 will be stored. The interrupt mask register (IMR) is configured to recognize the RINT interrupt, which is generated after receiving the last of 16 bits on the serial port. This interrupt is still disabled at this time however. The transmit framing synchronization pin (FSX) is configured to be an output. The F0 bit of the status register ST1, is initialized to zero which sets up the serial port to operate in the 16-bit mode. For circuit 1 the XF bit is first initialized to zero.

The code for transmitting and receiving data is the same for the two circuits except for the section of code labelled TIMER. The D_{IN} word is loaded into the ACC and shifted left eight times so that it appears as in Figure 3. This D_{IN} word configures the LTC1091 for CH0 with respect to CH1 and MSB first. The D_{IN} word is then put in the transmit register and the RINT interrupt is enabled. For circuit 1 the XF bit is set which causes the FSX pin to generate a $\overline{CS}$ signal which is fed into the $\overline{CS}$ pin of the LTC1091 and the FSR pin of the TMS320C25.

The XF bit is then reset and set with a 2.0μs period until the RINT interrupt is generated. For the circuit 2 code the timer consists of only one instruction that loops upon itself until the RINT interrupt is generated. All clocking and $\overline{CS}$ functions are performed by the hardware. This time could be used to do some simple processing of the data.

Once RINT is generated the code begins execution at the label RINT. This code stores the D_{OUT} word from the LTC1091 in the ACC and then stores it in location 200 hex. The data appears in location 200 hex right justified as shown in Figure 4. The code is set up to continually loop, so at this point the code jumps to label TXRX and repeats from there.

The code for circuits 1 and 2 can be made to work with the LTC1092 as well with only minor modifications. It is not necessary to use a D_{IN} word for the LTC1092, which reduces the number of lines required by the interface. After the data has been shifted into the TMS320C25 it must be shifted twice to the left for left justified data or shifted four times to the right for right justified data.

Summary

Two interfaces between the LTC1091 10-bit data acquisition and the TMS320C25 DSP were demonstrated. The first interface required only one inverter in addition to the A/D and the DSP. The combined data conversion and transfer time of this interface was 39μs. The data was placed in the internal RAM of the TMS320C25 in a right justified format. The second circuit, which required a counter and an inverter to implement, was able to perform a conversion and shift the data to the processor in only 32μs. The data again was placed in the RAM of the TMS320C25 in a right justified format. With only minor modifications these interfaces can also be used with the LTC1092.

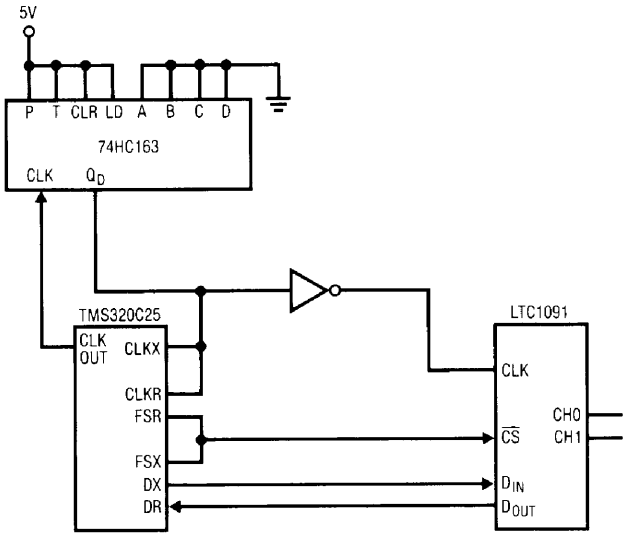


Figure 6. Circuit 2: Minimum Software Interface

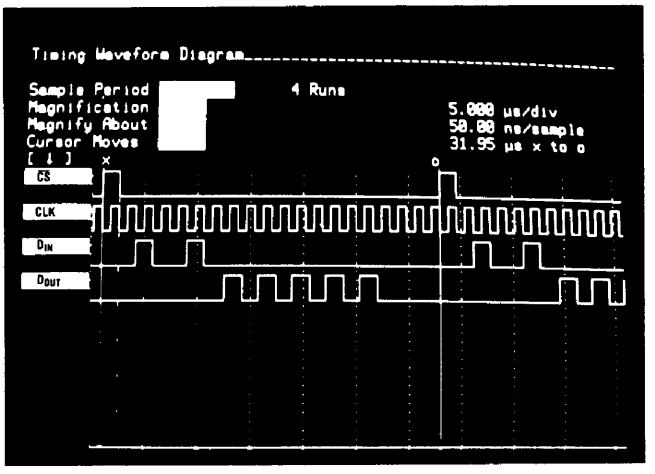


Figure 7. Timing for Circuit 2 Shows 32μs Throughput Rate

```
LABEL    MNEMONIC   COMMENTS
         AORG  0    ON RST CODE STARTS AT 0
         B     INIT BRANCH TO INIT ROUTINE

         AORG  >26  ADDRESS OF RINT VECTOR
         B     RINT BRANCH TO RINT ROUTINE

         AORG  >32  MAIN PROGRAM ADDRESS
INIT     DINT       DISABLE INTERRUPTS
         LDPK  >0   DATA PAGE POINTER IS 0
         LARP  >1   AUX. REG. POINTER IS 1
         LRLK  AR1,>200 AUX. REG. 1 = >200
         LACK  >10  CONFIG. WORD FOR IMR
         SACL  >4   PUT CONFIG. WORD IN IMR
         STXM       CONFIGURE FSX AS OUTPUT
         FORT  0    SET SERIAL PORT TO 16 BITS

TXRX     LACK  >48  LOAD DIN WORD INTO ACC
         SFSM       FSX STARTS ON XSR LOAD
         RPTK  7    REPEAT 8 TIMES
         SFL        SHIFT DIN TO LEFT
         SACL  >1   DIN PUT IN TX REGISTER
         EINT       ENABLE INTERRUPTS

TIMER    B     TIMER LOOP UNTIL FINISHED

RINT     ZALS  >0   STORE DOUT WORD IN ACC
         SACL  *,0  STORE ACC IN >200
         B     TXRX GO TO TXRX ROUTINE
         END
```

Figure 8. TMS320C25 Code for Circuit 2