TMS320 DSP
DESIGNER'S NOTEBOOK
Number 72
TEXAS INSTRUMENTS

Interfacing Two Analog Interface Circuits to One TMS320C5x Serial Port

Contributed by Manuel Rodrigues

Design Problem How can I integrate two analog lines when I only have one serial port?

Solution In some cases, like wireless communications applications, a system requires two communication channels but only one serial port is available. Figure 1 shows how to multiplex this port and how to interface two analog lines, while an example of code is given in Figure 2. The analog interface circuits (AICs) chosen for this solution were purposely selected to address as many potential problems as possible.

The first AIC, a TLC320AC02 that we will call AIC A, has 14-bit linear resolution and audio frequency with integral anti-aliasing and reconstruction filters. It is designed for easy connection to many DSP chips and is appropriate for a wide variety of applications.

The second AIC, a TCM320AC46 that we will call AIC B, is a 13-bit linear resolution Voice Band Audio Processor (VBAP) with transmit and receive filtering. It is also designed for many general-purpose applications.

Data Transmission on the Serial Port

The DSP's serial port, which runs in burst mode, consists of three types of signals: the clock (CLK), which imposes the pace of the bit sequence and equals the bit rate of the communication; the frame synchro (FS), which indicates the beginning of a bit sequence, on a negative slope, to the other device; and finally, the data line (D), which conveys the bits.

The bit rate and the FS signal are imposed by the active AIC in both emission and reception, and the same FS simultaneously triggers one reception and one emission. Data is transmitted and loaded most significant bit first and is right justified.

The TLC320AC02 (AIC A)

The 'AC02 works in variable data rate transmission and is driven by a master clock (MCLK), which is divided by 4 to provide the shift clock (SCLK). It contains programmable counters with internal registers to adjust the sample frequency, the FS frequency, and filter parameters. The 'AC02 also has a power-down pin for sleep mode, saving the internal programmed states. In this example, it is used in stand-alone mode, but it also has a master-slave and a codec mode.

The TCM320AC46 (AIC B)

The 'AC46 operates in fixed data rate transmission and the CLK and FS is generated externally. The clock frequency should be 2.048 MHz and the FS should be 8 kHz. In this example, it performs a linear conversion, but it can also perform μ-law companded conversion. Also, a power-down pin can be used as a chip select.

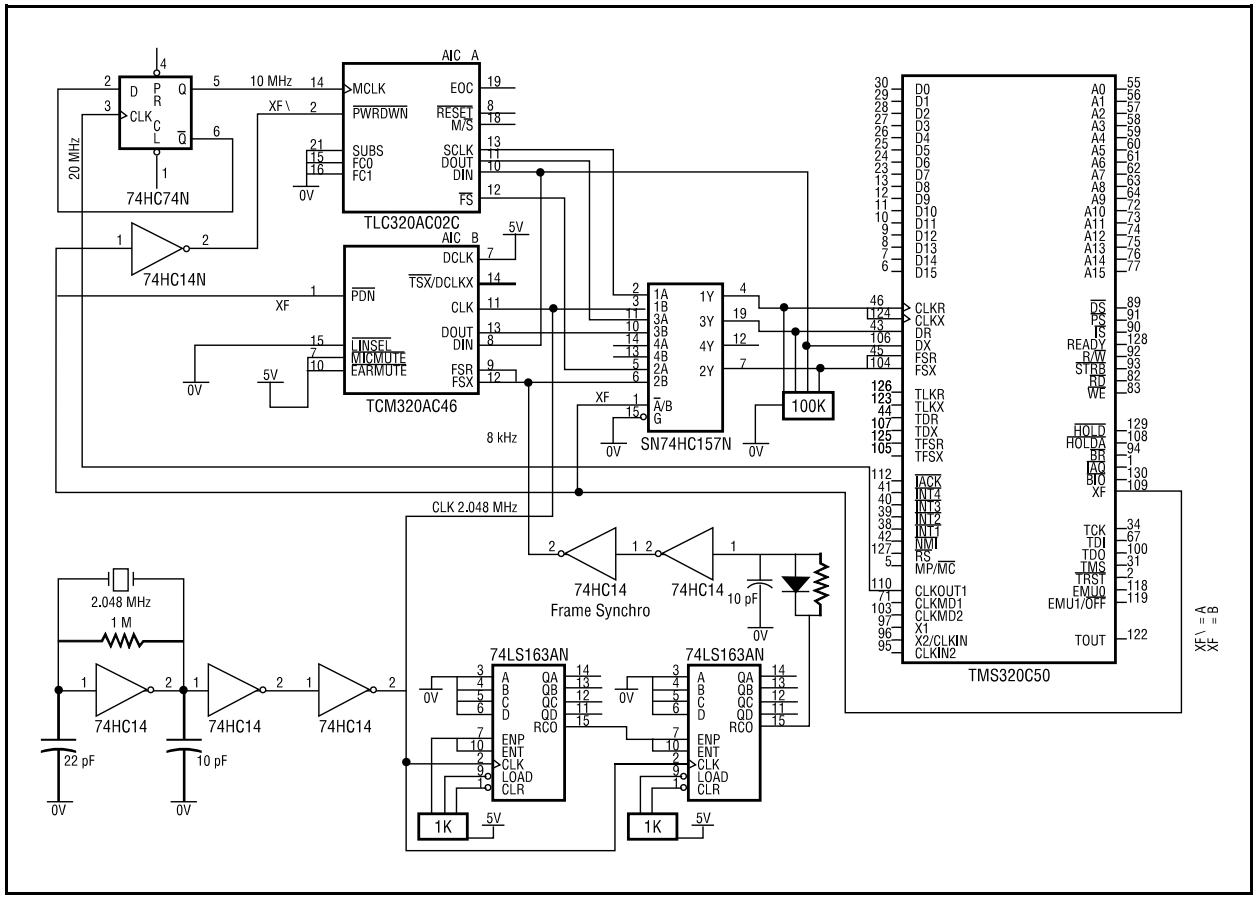

Figure 1. Connecting two AICs to one 'C5x serial port

Schematic Details

The AIC B clock is generated with a 2.048-MHz quartz. The AIC B FS is generated by dividing the clock by 256 due to two 4-bit counters in succession. The 74LS163 counters have a Ripple-Carry-Output (RCO) pin that provides a pulse every 256th clock period. Another solution is to use a single 8-bit 74LS590 counter that also has an RCO pin. Then FS is filtered through an RC circuit and two trigger inverters in order to eliminate parasitic spikes.

The schematic contains six inverters, two of which have to be Schmidt triggers which come from the same 74HC14 package. The AIC A has a maximum MCLK frequency of 15 MHz. It is highly recommended that AIC A and the DSP are clocked from a common master crystal. However, depending on which speed grade is used, the DSP's CLKOUT frequency can be 20 or 40 MHz. So, CLKOUT has to be divided by one or two stages of the 74HC74 flip-flops. Here, CLKOUT frequency is 20 MHz, and is divided by two, giving a 10-MHz MCLK. The chip selection for AIC A and AIC B is done by the external flag (XF) signal coming from the DSP. The XF value is fixed by the XF bit in the DSP's status register, ST1. Both AICs are then set to alternate so when one is on, the other is powered down.

The 74LS157, an 8-to-4 multiplexor, drives the signal coming from the selected AIC to the DSP. CLK, FS, and DR signals then pass through the MUX. DX is an output of the DSP, and is wired directly to the AIC's data-in (DIN) pins.

Software Explanation

Both AICs are driven by the XF bit in the status register, ST1. Therefore, AIC selection is made by setting or clearing this bit. Notice that the XF bit is not pushed by the interrupt context save, so pay special attention when changing the XF value.

At the beginning of the program, some specific initializations will be made concerning the AICs. Indeed, AIC A provides internal registers useful to select sampling frequency or others parameters like input and output gains, or synchro between master and slave devices. A normal communication (16-bit frame) happens every sample, but in order to read or write in internal registers, a second communication happens between each normal frame. Two external pins, FC0 and FC1, are used in addition with the two LSBs of the primary frame to force this second communication. When FC0 and FC1 are tied low, the secondary frame is forced only by setting the two LSBs to one logical.

Finally, do not request an untimely secondary frame. To prevent this, a mask value 0FFFCh is used before sending any frame. Unlike AIC A, when using the AIC B, the sampling frequency cannot be programmed as it is externally imposed by the FS signal.

```
*  XF in ST1 drives both the AICs as follow:
*         CLRC      XF              ; Enables 320AC02C
*         SETC      XF              ; Enables 320AC46

MASK      .set      60h             ; MASK for both AICs

* Initialization of AICs
          DINT                      ; Disable interrupts during initialization
          SST       #1,STATUS       ; Status register ST1 provides the
          BIT       STATUS,11       ; value of bit XF in order to branch
          BCND      AIC02,NTC       ; at the corresponding routine.
AIC46     SPLK      #0FFF8h,MASK    ; MASK value for AIC 46 (13 bits)
          B         RESTORE

*******************************************************************************
*  DESCRIPTION: This routine initializes the TLC320AC02C for a 13.88 kHz      *
*  sample rate with a gain setting of 1, a master clock = 10 MHz and a shift  *
*  clock frequency SCF = 250 kHz.                                             *
*******************************************************************************
          .def    AIC02
AICCFG    .set      $
          .word     0003h           ; Ask for secondary communication
          .word     0114h           ; Reg A = 20
          .word     0003h           ; Ask for secondary communication
          .word     2100h           ; Read A reg = ?
          .word     0003h           ; Ask for secondary communication
          .word     2200h           ; Read B reg = ?
          .word     0003h           ; Ask for secondary communication
CFGEND    .word     0000h           ; No-Op waiting for answer
```

Figure 2. Example of code

```
AIC02       .set        $
            SPLK        #0FFFCh,MASK      ; MASK value for AIC 02 (14 bits)
            MAR         *,AR1             ; Load arp
            LAR         AR1,#0300h        ; Point to FIFO (@0300h is an example)
            LACC        #AICCFG           ; Load configuration table start address
            RPT         #7                ; Move 8 configuration words to end of
            TBLR        *+                ; FIFO so it will be send first to AIC.
            SPLK        #0308h,ARCR       ; Load end of table
            SPLK        #0300h,AR1        ; and begining of table.
                                          ; Sequence of emission and reception
TXWAIT      BIT         SPC,4             ; Wait for transmit ready
            BCND        TXWAIT,NTC        ; until xrdy goes low.
            LACC        *0+               ; Read next control word
            SACL        DXR               ; and send it to DXR.
RCWAIT      BIT         SPC,5             ; Wait for receive ready
            BCND        RCWAIT,NTC        ; until rrdy goes low.
            LACL        DRR               ; Get answer word from DRR
            SACL        *0-               ; and store it to check answers.
            MAR         *+                ; Go to next control word
            CMPR        0                 ; Check for end of table, else loops back
            BCND        TXWAIT,NTC        ; to a new Transmit/Receive sequence
                                          ; End of AIC-02 initialization
RESTORE:
            CLRC        INTM              ; Enables Interrupts
            SPLK        #0030h,IMR        ; Enables RINT and XINT
            B                   MAIN      ; Check which part of table was sent
```

Figure 2. Example of code (continued)