

DESIGNER'S NOTEBOOK



Interfacing 20-MSPS TLC5510 Flash A/D Converter to TMS320C2xx and TMS320C5x Fixed-Point DSPs

Contributed by Martin Staebler

Design Problem

How do I interface the TLC5510 Flash A/D converter to a TMS320C2xx or 'C5x DSP at a data rate of 20 MSPS?

Solution

The 8-bit TLC5510 flash A/D converter can be directly interfaced to the 'C2xx and 'C5x DSPs as shown in Figures 1a and 1b. With the repeated block move instruction BLDD, it is possible to read a sample from the TLC5510 and store it into internal memory in one instruction cycle at zero wait states, thus achieving 20-MSPS data transfer rate with the DSP clocked at 40 MHz.

Schematics

The 16-bit fixed-point 'C2xx and 'C5x DSPs utilizing an advanced Harvard architecture with separate strobes for program (\overline{PS}), data (\overline{DS}), and I/O space (\overline{IS}), each with a 64k address range. An additional strobe enables a global memory space, which overlays the upper data memory space. The size is 0–32k, determined by the memory-mapped register GREG. With GREG = 0FFh, for example, the global data memory ranges from FF00h–FFFFh. If it is accessed, \overline{BR} goes low in addition to the \overline{DS} strobe. Figures 1a and 1b show the interface of the TLC5510 to the 'C2xx and the 'C5x, respectively. The TLC5510 is connected to the lower 8 bits of the 'C2xx/'C5x data bus. The CLK signal is connected with the read signal \overline{RD} of the DSP and is active when external memory is accessed. The corresponding strobe signal stays low during consecutive reads.

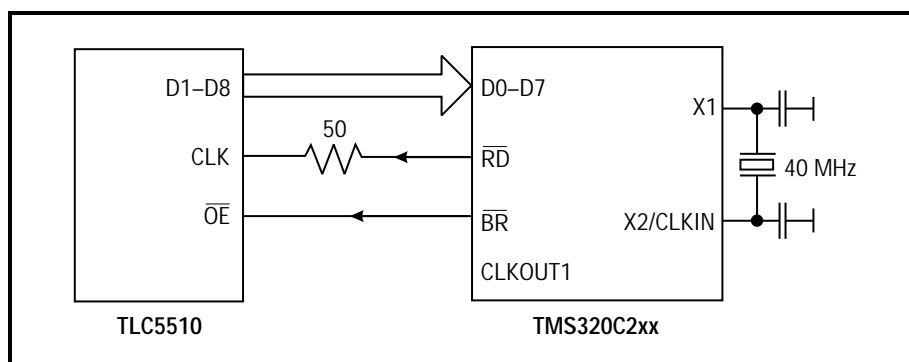


Figure 1a. TMS320C2xx-to-TLC5510 interface

The TMS320C2xx doesn't incorporate a memory-mapped I/O range, so the TLC5510 is accessed through the global data memory space, where \overline{BR} strobe is active. If global memory overlaps the data memory, the two active strobe signals \overline{DS} and \overline{BR} have to be decoded, respectively.

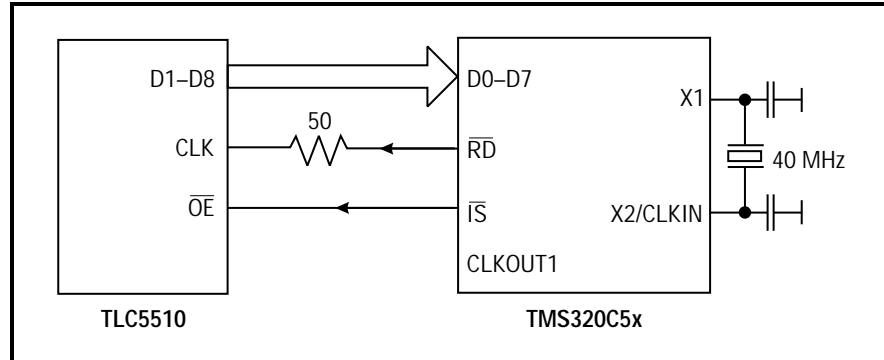


Figure 1b. TMS320C5x-to-TLC5510 interface

For the TMS320C5x, the TLC5510 is accessed through the memory-mapped I/O space (50h–60h), where the I/O strobe \overline{IS} is active. If it is the only device within the 64k I/O range, an address decoder is not required.

The interface length should be as short as possible, and a line matching resistor, especially for the CLK signal, is recommended to reduce overshoot and ringing since the analog input is sampled on the falling edge of CLK. If the 2.5 CLK cycle latency becomes an issue after accessing the TLC5510, it can also be clocked with the inverted CLKOUT1 signal, instead of using \overline{RD} , or with an external clock. However, if an external clock is used, it must be phase locked to the CLKOUT1 frequency with a delay of no more than 5 ns to meet the 'C2xx/'C5x timing requirements.

Because the high data byte of the 'C2xx/'C5x data bus (D8–D15) is not defined, when the TLC5510 is accessed, it can be pulled low by hardware or the higher data byte can be masked by software. The TLC5510 provides 5-V CMOS compatible inputs and outputs, thus AC(T) logic families are recommended for glue logic, such as clock inverters, clock dividers, or line drivers.

Interface Timing

For the TLC5510, analog input data is sampled on the falling edge of the CLK signal and converted with a latency of 2.5 clock cycles, due to the semi-flash architecture. The digital output D1–D8 is valid after the rising edge of $\text{CLK} = \overline{RD}$ with a delay of $t_{DD} < 30$ ns.

One DSP instruction cycle – $t_{c(CO)}$ – is determined by the falling edge of CLKOUT1. Data is read in a single cycle and has to be valid $t_{su(R-RD)} \geq 13$ ns ('C2xx-40 MHz) and $t_{su(R-RD)} \geq 10$ ns ('C5x-40 MHz), and before the rising edge of \overline{RD} with a minimum hold time of zero ns. Since the TLC5510 data is valid no later than $t_{c(CO)} - t_{dd(max)} = 20$ ns before the next rising edge of $\text{CLK} = \overline{RD}$, the TLC5510 meets the data setup and hold timing requirements.

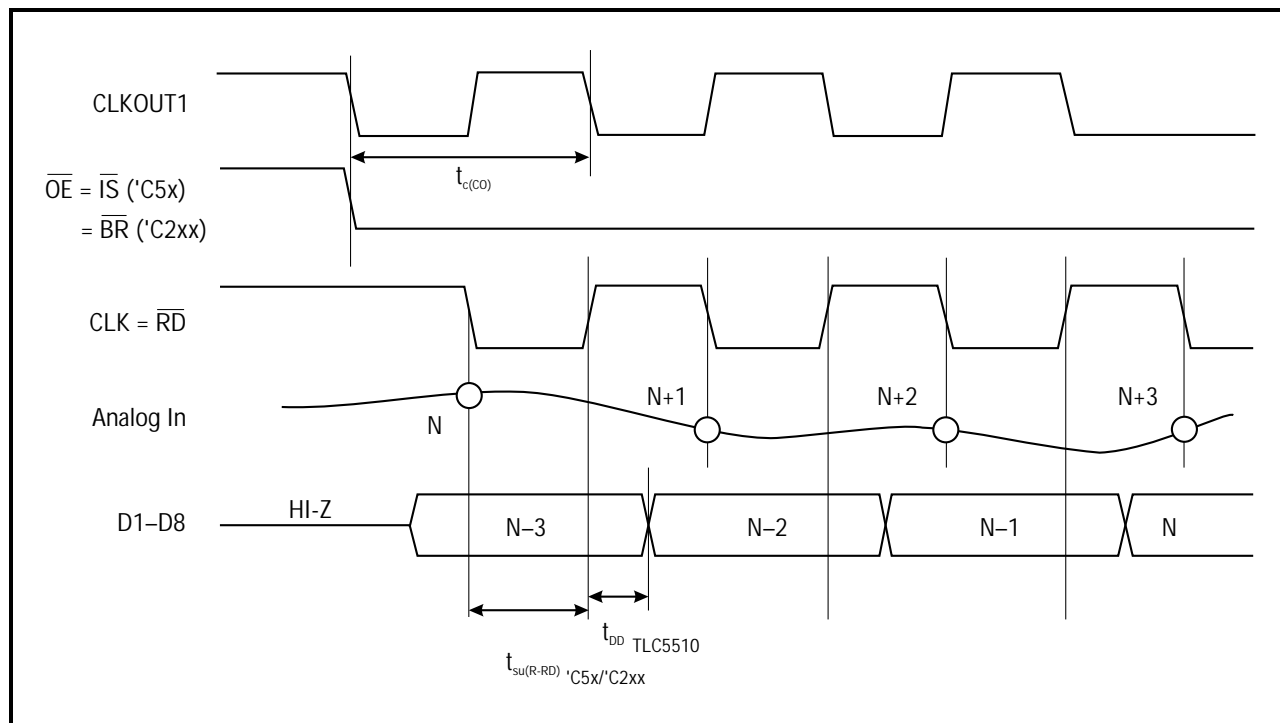


Figure 2. TLC5510-to-TMS320C5x Interface Timing

Software Example for the TMS320C2xx and TMS320C5x

The software is written with the Texas Instruments Fixed-Point Assembler V 6.63. By enabling either TMS320C2xx or TMS320C5x within the program header, the program can be compiled to run on both devices. The data is transferred using a repeated block move instruction BLDD where the number of words to be moved is one greater than the number in the repeat counter (RPTC), and is loaded from the address B_LEN.

```

MAR      *,AR0           ;AR0 points to the external address (TLC5510)
LDP      #B_LEN          ;load RPTC (repeat counter) with 16-bit value
RPT      B_LEN           ;using dma address B_LEN
BLDD     *,#SAMPLES      ;SINGLE cycle data transfer ext.-->internal memory

```

If immediate addressing is used, only an 8-bit value could be stored into RPTC and the maximum number of repeated instructions would then be 255. Since the source is external and the destination is internal, the repeated BLDD instruction is executed in a single cycle with one data transfer. The TLC5510 is addressed indirect, via AR0 and the start address within the internal memory is addressed with the long immediate constant SAMPLES. There's no need to increment the long immediate destination address, because it is automatically incremented by one, when BLDD is in repeat mode. In the software example, 512 bytes (D0-D7) are read from the TLC5510 (one byte each 50-ns instruction cycle for a 40-MHz quartz) and stored within the internal single access RAM, SARAM from address SAMPLE to SAMPLE + 511 in consecutive order. Remember that an instruction within a repeated loop is **not interruptible**.

```

*****
* (C) TEXAS INSTRUMENTS DEUTSCHLAND GMBH, 1996                                     *
* File: TLC5510.ASM                                                                *
*****
C5x      .set      0          ; !!! --- set to 1, when a TMS320C5x is used
C2xx     .set      1          ; !!! --- set to 1, when a TMS320C203/209 is used
-----
WSGR     .set      0FFFFh      ;C209 I/O space mapped waite state register

        .mmregs
        .global    START      ;define memory mapped registers

        .bss       SAMPLES, 512 ;has to be mapped to on-chip memory!

        .data
B_LEN:   .word      511        ;mapped to data space
                                   ;no of data transfers -1

        .sect      "vectors"
        .word      START

        .sect      "rd_adc"
READ_TLC5510:
        .if        'C5x
        LAR        AR0,#50h    ;memory-mapped I/O space,  $\overline{IS}$  = active strobe
        .endif
        .if        'C2xx
        LAR        AR0,#0FFFFh ;global data memory access,  $\overline{BR}$  (&  $\overline{DS}$ ) = active strobe
        .endif
        MAR        *,AR0
        LDP        #B_LEN
        RPT        B_LEN
        BLDD       *, #SAMPLES ;SINGLE cycle data transfer ext.--internal memory
        RET

START:   .text
        SETC       INIM        ;disable all interrupts
        SETC       CNF         ;map B0 into program space
        LDP        #0          ;set data page pointer

        .if        'C5x
        OPL        #0834h,PMST ;0000 1000 0011 0000 --> UG, page 3-38
                                   ;      |      ||  +-+ reset NDX
                                   ;      |      +-+ map on chip SARAM into
                                   ;      |      program and data space
                                   ;      +-+ interrupt vect. start at 0800h

        LACC       #0
        LDP        #0
        SACL       CWSR        ;--> zero wait state for
        SACL       PDWSR        ; program and data 0000h-0FFFFh
        SACL       IOWSR        ; 0000h-FFFFh i/o space
        .endif

```

Figure 3. Program listing to interface the TLC5510 to a 'C2xx or 'C5x DSP

```

RAM1    .if      'C2xx
        .set     60h          ; RAM1 is mapped to scratch pad RAM
        LDP      #0
        SPLK     #0, RAM1     ; --> zero wait state for program, data and
        OUT      RAM1,WSGR    ;      i/o space 0000h-FFFFh

        LDP      #0          ;  $\overline{BR}$  active from FF00h-FFFFh (global data memory)
        SPLK     #0FFh, GREG  ; when accessing data memory
        .endif

        CLRC     OVM          ; prevent arithmetic saturation
        SPM      0            ; product shift mode = 0 --> no shift
        SETC     SXM          ; sign extension

        CALL     READ_TLC5510

;===== end of main program =====;

```

Figure 3. Program listing to interface the TLC5510 to a 'C2xx or 'C5x DSP (continued)