TMS320 DSP
DESIGNER'S NOTEBOOK
Number 37

TEXAS INSTRUMENTS

Serial ROM Boot

Contributed by Alex Tessarolo

Design Problem How do I use a serial ROM for minimum form factor storage of boot code?

Solution With ever increasing speeds of DSP devices such as the 'C5x, and the increasing demand for reduced form factors such as in HDD applications, there is a need for integrating the program memory into the DSP device. For volume production whereby the program code has been fully debugged and changes are not foreseen, then masked ROM is appropriate. However, for prototyping and preproduction evaluation, a programmable device is desirable, i.e., a DSP with EPROM memory. However, EPROM technology has not kept pace with the speed requirements of a DSP and hence other solutions need to be found in the interim. One solution is to populate the DSP with internal RAM that can be programmed at boot time from an external source. This external source can be either a coprocessor or a smaller form factor ROM. To meet the need of minimum board area, a serial ROM is an ideal device for such applications.

Serial ROM Description

A serial ROM is a memory device that is addressed sequentially one bit at a time. Data within the ROM cannot be accessed randomly. An internal address generator points to a single data bit and is automatically incremented by an external clock signal. The address generator is reset by a separate external signal to begin a new transfer. Typically data can be clocked out at a rate of 5 MBits/second on such devices.

A serial ROM can be either a one-time programmable (OTP) or electrically-erasable and programmable (EEPROM) device. The serial ROMs described in this application note are manufactured by Xilinx and are OTP devices with capacities ranging from 36,288 bits up to 131,072 bits. Larger devices are in the pipeline. With a capacity of 131K bits, up to 8K words (1 word = 16 bits) of DSP program or data can be stored in such a device. For example, a serial ROM interfaced to a DSP device such as the TMS320C53 with 4K words of internal program/data RAM could be programmed to run a fairly sizable program internally. The program would be transferred by a boot loader program which resides in masked memory within the DSP and is initiated at power up or reset.

Serial ROMs typically come in 8-pin DIP or SOIC packages (20-pin PLCC also available) and therefore have a small form factor. Pinouts for such a device are shown below:

Serial ROM/DSP Connection

Only three connections are needed between the serial ROM and the DSP. The FSX signal is used as the reset input to the ROM. The BIO input is used as the data sampling input signal and the clock line is driven by the FX output. All of the above signals will be under software control.

A typical software kernel written for either the 'C2x or 'C5x DSP that will transfer the contents of the serial ROM to the DSP memory is described below:

```
LRLK        AR2,#dest_addr         ; AR2 = destination address pointer.
LRLK        AR3,#no_of_words       ; AR3 = number of words to transfer.
STXM                               ; FSX configured as an output. Serial ROM enabled.
LARP        AR1
Outer_Loop: ZAC
            LARK AR1,#16           ; AR1 = bit counter. Initialized to 16 bits.
Inner_Loop: SXF                    ; Drive FX = CLK high.
            SFL                    ; Shift ACC left 1 bit.
            RXF                    ; Drive FX= CLK low.
            BIOZ Loop_Inner,*-     ; Branch if bit = 0 and decrement bit counter (AR1).
            ADDK #1                ; Bit must be = 1.
            BANZ Loop_Inner,*      ; Branch if bit counter (AR1) does not = 0.
            LARP AR2               ; One word read from serial ROM.
            SACL *+,0,AR3          ; Store word in destination address (AR2).
            BANZ Outer_Loop,*-,AR2 ; Loop until all words transferred.
```

Figure 1.

The above program running from a 50-ns 'C5x DSP can transfer 8K words of data from a 128K × 1-bit serial ROM in about 50 ms. This is an effective data transfer rate of approximately 2.5 MBits/sec.

Serial ROM Data Format

A typical data storage format inside the serial ROM is shown below. The first word is a serial ROM detect sequence for the boot loader program. The following words contain the data to be transferred. Multiple blocks can be transferred to various destination addresses. The block size and destination address are found in the first two words of each block. The data words then follow and at the end of the block there is a 32-bit check-sum. The check-sum is the addition of all the data words within the block (excluding block size and destination address). The next block to transfer, immediately follows the check-sum. If the first word read is a zero, then there are no more blocks to follow.