TMS320 DSP
DESIGNER'S NOTEBOOK

Serial ROM Boot

APPLICATION BRIEF: SPRA233

Alex Tessarolo
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
May 1994

TEXAS INSTRUMENTS

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 application 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.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Examples

Serial ROM Boot

Abstract

With ever increasing speeds of DSP devices such as the ’C5x, and the increasing demand for reduced form factors, as in HDD applications, there is a need for integrating the program memory into the DSP device. For volume production where 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).

This document discusses how to use a serial ROM for minimum form factor storage of boot code. A code listing is included.

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 ROM (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 that 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.

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.

Example 1. Software Kernel

```
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.
```

The above program running from a 50-ns ’C5x DSP can transfer 8K words of data from a 128K by 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 bootloader 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.