TMS320 DSP
DESIGNER'S NOTEBOOK

Creating a Delay Buffer on a TMS320C2x EVM

APPLICATION BRIEF: SPRA214

Tom Horner
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
February 1993

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

Figures

Examples

Creating a Delay Buffer on a TMS320C2x EVM

Abstract

This document discusses how to implement an audio delay buffer using the TMS320C2x Evaluation Module (EVM). A block diagram of the circuit and a memory map showing how the delay is implemented are included. There is a code listing of the code used for this function.

Design Problem

How can I implement an audio delay buffer with the TMS320C2x EVM?

Solution

The key to this technique is that the buffer length is equal to the sample delay time you want to use and that the input/output rates are equal. There is only one pointer required and it is used for output and input both (in that order). The delayed value is first output and then a new input value is read into memory. Finally, the pointer is incremented to the next memory location. Due to the fact that there is only one pointer overhead to check if pointer(s) is at the end of the buffer is reduced. Use a counter to determine when the pointer is at the end of the buffer. This approach can be implemented using a BANZ instruction.

Figure 1. Hardware

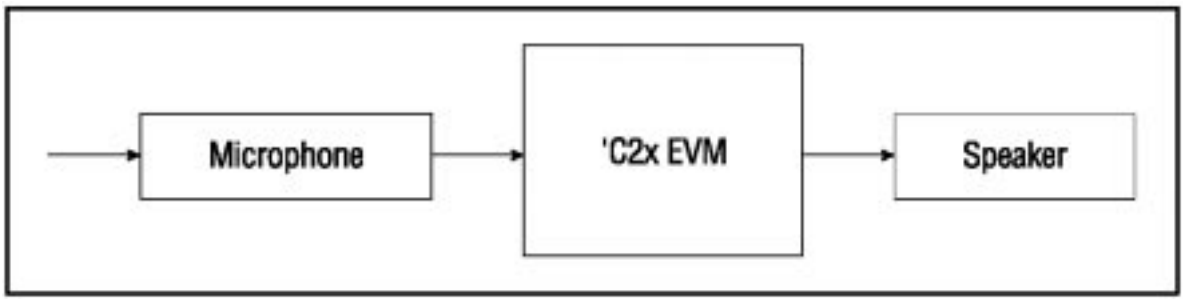
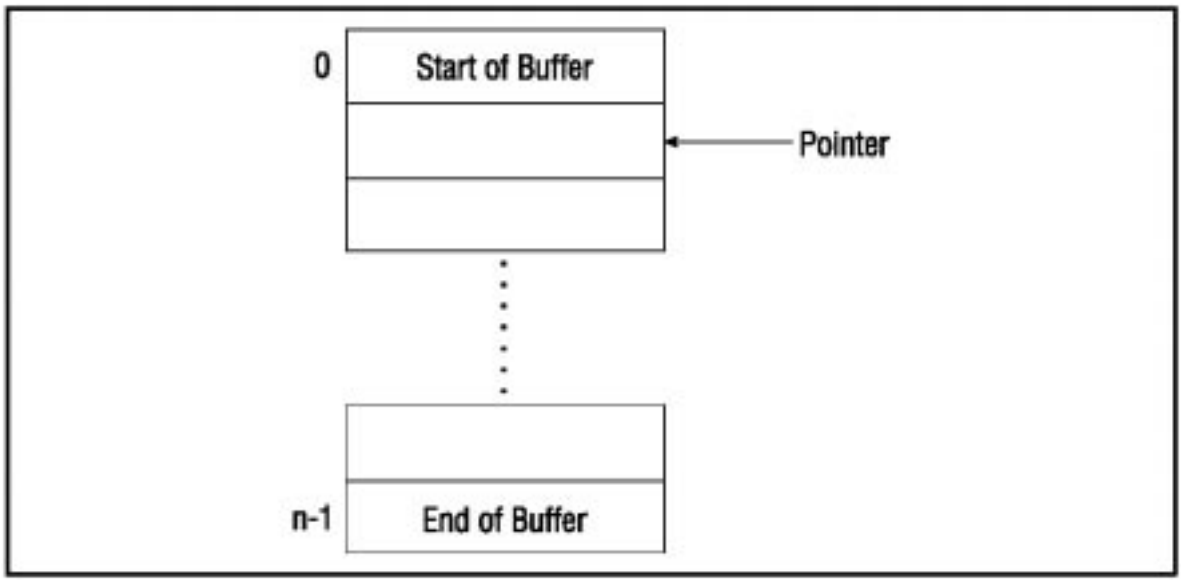


Figure 2. Memory - delay buffer



Software

This shows only the portion required for the delay buffer implementation. The entire program is on the BBS as 2XEVMBUF.EXE, which is a self-extracting zip file.

Example 1. Software Example

```
;———————— CONSTANTS

BUFFER_START     .set 08000h         ;Define delay buffer constants
BUFFER_LENGTH    .set 04000h

;———————— MEMORY DEFINITION
;Reserve ext RAM for delay buffer

DELAY            .usect “ext_mem”, 16384

;———————— ZERO DELAY BUFFER

    larp         AR1
    lrlk         AR0, BUFFER_LENGTH-1 ;AR0 = Memory block length-1
    lrlk         AR1, BUFFER_START    ;AR1 = Delay Buffer pointer
    ZAC
ZER01
    sacl         *+, AR0          ;Initialize Delay Buffer to zero
    banz         ZER01, AR1       ;Done??

;———————— INITIALIZE DELAY BUFFER ————————————————-

    lrlk         AR0, BUFFER_LENGTH-1 ;AR0 = end of buffer counter
```

```
    lrlk        AR1, BUFFER_START    ;AR1 = output/input pointer
    larp        AR1

;————————————————————————————————————————————-;
INTERRUPT SERVICE ROUTINES
;————————————————————————————————————————————-

RINT                            ;Serial Port Receive Interrupt
    LDPK        0
    lac         *               ;Read delayed input from memory
    sacl        DXR             ;Echo to AIC output
    lac         DRR             ;Read latest AIC input
    sacl        *+, AR0         ;Store to delay buffer
    banz        OUT, AR1        ;Check to see if at end of buffer
                                ;If yes reinitialize AR0 and AR1
    lrlk        AR0, BUFFER_LENGTH-1
    lrlk        AR1, BUFFER_START
OUT
    eint                        ;Re-enable GLOBAL interrupt
    ret                         ;Return to MAIN
    .end
```