TMS320 DSP
DESIGNER'S NOTEBOOK
Number 18

Creating a Delay Buffer on a TMS320C2x EVM

Contributed by Tom Horner

Design Problem How can I implement an audio delay buffer with the 'C2x 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 a the end of the buffer. This approach can be implemented using a BANZ instruction.

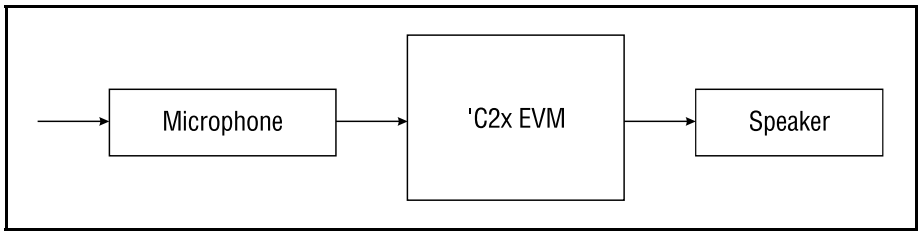

Figure 1. Hardware

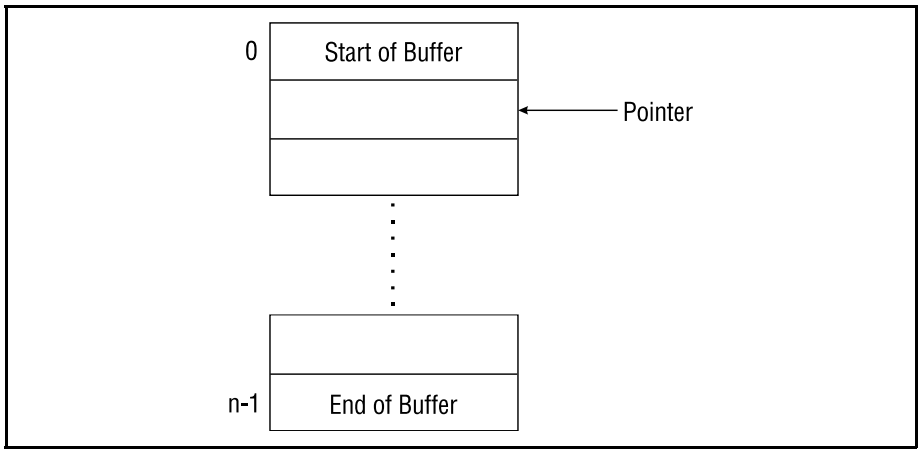

Figure 2. Memory - delay buffer

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.

```
 ;————————   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      ZERO1, 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
```

Figure 3. Software