```
        page    132
;****************************************************************************
;   TXRX_ISR.ASM
;     This file contains the basic Interrupt Service Routines (ISR)
;       used to service the SSI Transmit and SSI Receive Interrupts:
;               ssi_rx_isr
;               ssi_tx_isr
;
;       Before using these routines, the appropriate interrupt
;       vectors must be initialized to jump to the associated
;       handler. (e.g., P:$000C, the SSI RX interrupt vector should
;       have an instruction which forces the DSP to jump to the
;       address of ssi_rx_isr).  In addition, the DSP's Interrupt
;       Priority Register (IPR) must choose an interrupt level for
;       the SSI interrupts and the DSP56002's Mode Register must
;       enable that interrupt level.
;
;    The ISRs which follow require that r6 has been established as
;      a pointer to a stack which grows towards increasing addresses.
;****************************************************************************
;
;   Copyright (c) MOTOROLA 1995
;             Semiconductor Products Sector
;             Digital Signal Processing Division
;
;****************************************************************************



;---------------------------------------------------------------------
;   SSI Receive ISR
;   This Interrupt Service Routine is the destination of the SSI RX
;      vector located at p:$000C.   In many cases the SSI RX w/Exception
;      vector will also jump here.
;   R6 should point to the first free location on the stack (in X:memory).
;   The data is placed in a 1 frame (4 word) buffer and sync.
;      is verified/restored every frame.
;---------------------------------------------------------------------
ssi_rx_isr
    move    r0,x:(r6)+              ; Save r0 to the stack.
    move    m0,x:(r6)+              ; Save m0 to the stack.
    move    #3,m0                   ; Modulo 4 buffer.
    move    x:RX_PTR,r0             ; Load the pointer to the rx buffer.
    jclr    #3,x:SSISR,next_rx      ; If not fr. syc, jump to receive data.
    move    #RX_BUFF_BASE,r0        ; If frame sync, reset base pointer.
    nop
next_rx
    movep   x:SSIDR,x:(r0)+         ; Read out received data to buffer.
    move    r0,x:RX_PTR             ; Update rx buffer pointer.
    move    x:-(r6),m0              ; Restore m0.
    move    x:-(r6),r0              ; Restore r0.
    rti


;---------------------------------------------------------------------
;   SSI Transmit ISR
;   This Interrupt Service Routine is the destination of the SSI TX
;      vector located at p:$0010.   In many cases the SSI TX w/Exception
;      vector will also jump here.
;   R6 should point to the first free location on the stack (in X:memory).
```

```
;   The data is taken from a 1 frame (4 word) buffer and sync.
;      is verified/restored every frame.
;---------------------------------------------------------------------
ssi_tx_isr
    move    r0,x:(r6)+              ; Save r0 to the stack.
    move    m0,x:(r6)+              ; Save m0 to the stack.
    move    #3,m0                   ; Modulus 4 buffer.
    move    x:TX_PTR,r0             ; Load the pointer to the tx buffer.
    jclr    #2,x:SSISR,next_tx      ; If not frame sync, jump to transmit data.
    move    #TX_BUFF_BASE+1,r0      ; If frame sync, reset pointer.
    nop
next_tx
    movep   x:(r0)+,x:SSIDR         ; SSI transfer data register.
    move    r0,x:TX_PTR             ; Update tx buffer pointer.
    move    x:-(r6),m0              ; Restore m0.
    move    x:-(r6),r0              ; Restore r0.
    rti

^Z
```