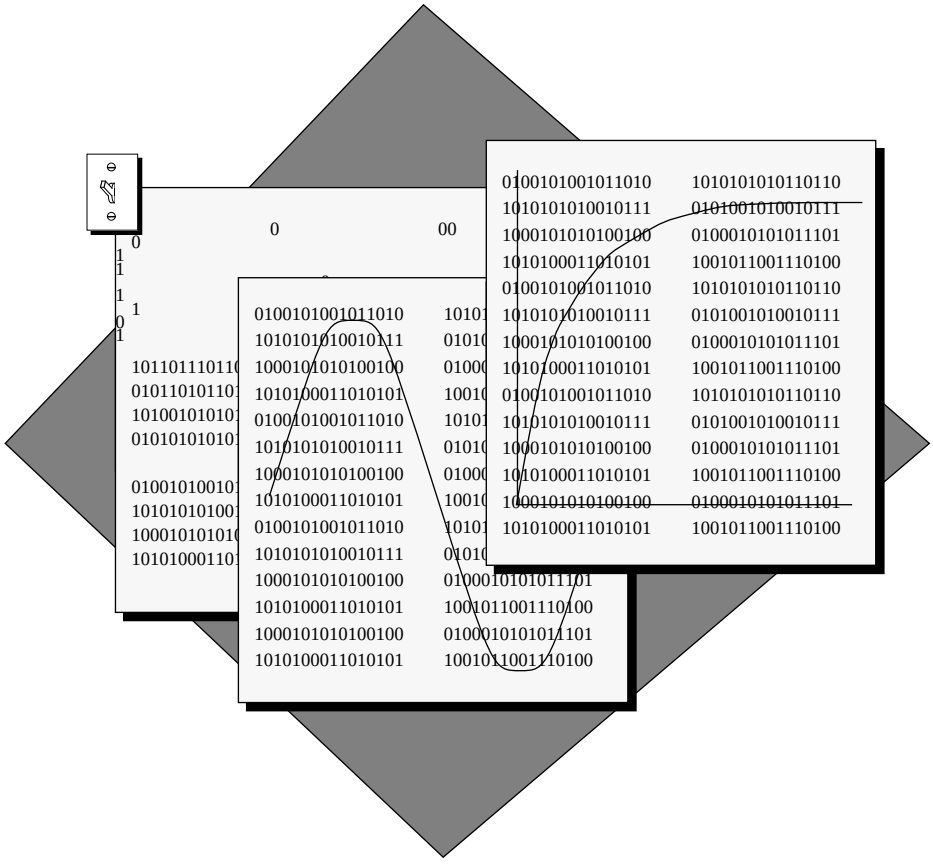
SECTION A

DSP56166 RAM BOOTSTRAP MODES



SECTION CONTENTS

A.1 INTRODUCTION

The bootstrap feature of the DSP56166 consists of four special on-chip modules: the 2048 words of PRAM, a 64-word bootstrap ROM, the bootstrap control logic, and the bootstrap firmware program.

Note: The bootstrap feature is only available on the DSP56166 **RAM** based part. The ROM Based DSP56166 does not have the bootstrap feature available. **As a result, this appendix only applies to DSP56166 RAM based part.**

A.2 BOOTSTRAP ROM

This 64-word on-chip ROM has been factory programmed to perform the actual bootstrap operation from the memory expansion port (Port A), the Host Interface, or the RSSI0. There is no access to the bootstrap ROM other than through the bootstrap process. Control logic will disable the bootstrap ROM during normal operations.

A.2.1 Bootstrap Control Logic

The bootstrap mode control logic is activated when the DSP56166 is placed in Operating Mode 0 or 1. The control logic maps the bootstrap ROM into program memory space as long as the DSP56166 remains in Operating Mode 0 or Mode 1. If the DSP is in Operating Mode 0 it will load 4096 bytes from a byte-wide memory (usually an EPROM) beginning at location P:$C000. If the DSP is in Operating Mode 1 it will load from either the Host Interface or RSSI0 depending on whether P:$C000 bit 15 is zero or one, respectively. The bootstrap firmware changes operating modes when the bootstrap load is completed. When the DSP56166 exits the reset state in Mode 0 or 1, the following actions occur:

1. The control logic maps the bootstrap ROM into the internal DSP program memory space starting at location $0000. This P: space is read-only.
2. The control logic forces the entire P: space, including the internal program RAM, to be write-only memory during the bootstrap loading process. Attempts to read from this space will result in fetches from the read-only bootstrap ROM.
3. Program execution begins at location $0000 in the bootstrap ROM. The bootstrap ROM program is able to perform the PRAM load through either the memory expansion port from a byte-wide external memory, through the Host Interface, or through RSSI0.
4. The bootstrap ROM program executes the following sequence to end the bootstrap operation and begin your program execution.
   A. Enter Operating Mode 2 by writing to the OMR. This action will be timed to remove the bootstrap ROM from the program memory map and re-enable read/write access to the PRAM.
   B. The change to Mode 2 is timed exactly to allow the boot program to execute a single cycle instruction then a JMP #00 and begin execution of the program at location $0000.

The bootstrap mode may also be selected by writing Operating Mode 0 or 1 into the OMR. This initiates a timed operation to map the bootstrap ROM into the program address space after a delay to allow execution of a single cycle instruction and then a JMP #<00 to begin the bootstrap process as described above in steps 1-4. This technique allows the DSP56166 user to reboot the system (with a different program if desired).

A.2.2 Bootstrap Firmware Program

Bootstrap ROM contains the bootstrap firmware program that performs initial loading of the DSP56166 PRAM. The program is written in DSP5616 core assembly language. It contains three separate methods of initializing the PRAM: loading from a byte-wide memory starting at location P:$C000, loading through the Host Interface, or loading serially through RSSI0. The particular method used is selected by whether (1) Operating Mode 0 or 1 is chosen and (2) the level of program memory location $C000, bit 15.

If the DSP is in Operating Mode 0 it will load 4096 bytes from a byte-wide memory (usually an EPROM) located in the lower byte beginning at location P:$C000 (see Figure B-1 of the applications examples given in **APPENDIX B APPLICATIONS EXAMPLES**). The data contents of the EPROM must be organized as shown below.

Address of External Byte Wide P Memory	Contents Loaded to Internal PRAM at:
P:$C000	P:$0000 low byte
P:$C001	P:$0000 high byte
•	•
•	•
•	•
P:$CFFE	P:$07FF low byte
P:$CFFF	P:$07FF high byte

If the DSP is in Operating Mode 1 and bit 15 at location P:$C000 is low then the DSP will load from the Host Interface. Typically a host microprocessor will be connected to the DSP56166 Host Interface (see Figure B-3 of the applications examples given in **APPENDIX B — APPLICATIONS EXAMPLES**). The host microprocessor must write the Host Interface registers TXH and then TXL with the desired contents of PRAM from location P:$0000 up to P:$0FFF. If less than 2048 words are to be loaded, the host programmer can exit the bootstrap program and force the DSP56166 to begin executing at location P:$0000 by setting HF0=1 in the Host Interface during the bootstrap load. In most systems, the DSP56166 responds so fast that handshaking between the DSP56166 and the host is not necessary.

If the DSP is in Operating Mode 1 and bit 15 at location P:$C000 is high, then the DSP will load from RSSI0 starting with the least significant byte first.

The bootstrap program listing is shown in Figure A-1.

```
; Bootstrap source code for the Motorola 16-bit DSP
; (C) Copyright 1989 Motorola Inc.
;
; Host Bootstrap, RSSI0 Bootstrap and External Bus Bootstrap
;
; This is the Bootstrap program contained in the DSP56166 RAM Based. This program
; can load the internal program memory from one of 3 external sources.
; The program reads the OMR bits MA and MB to decide which external  source to access.
; If MB:MA = 00 - load from 4,096 consecutive byte-wide P: memory locations (starting at P:$C000).
; If MB:MA = 01 - load internal PRAM through the Host Interface if bit 15 of P:$C000 is zero
; and load internal PRAM through RSSI0 if bit 15 of P:$C000 is set.
;
PRAMSIZE        EQU     2048            ; On-chip program RAM size
BOOT            EQU     $C000           ; The location in P: memory
                                        ; where the external byte-wide
                                        ; EPROM is to be mapped
M_PBC           EQU     $FFC0           ; Port B Control Register
M_PCC           EQU     $FFC1           ; Port C Control Register
M_HSR           EQU     $FFE4           ; Host Status Register
M_HRX           EQU     $FFE5           ; Host Receive Data Register
M_CRA0          EQU     $FFD0           ; RSSI0 Control register A
M_CRB0          EQU     $FFD1           ; RSSI0 Control register B
M_SR0           EQU     $FFF0           ; RSSI0 Status register
M_RX0           EQU     $FFF1           ; RSSI0 Serial receive register
;
        ORG     PL:$0                   ; Bootstrap code starts at P:$0

        MOVE    #M_PBC,R2               ; R2= Port B Control Register
        MOVE    #BOOT,R1                ; R1= External P: address of
                                        ; bootstrap byte-wide ROM
        LEA     (R2)+,R3                ; R3= Port C control Register

; If this program is entered by changing the OMR to bootstrap mode, make certain that
; registers M0 and M1 have been set to $FFFF (linear addressing).
; Make sure the BCR register is set to $xxxF since EPROMs are slow.
;
; The first routine will load 4,096 bytes from the external P memory space beginning at
; P:$C000 (bits 7-0). These will be condensed into 2,048 16-bit words and stored in
; contiguous internal PRAM memory locations starting at p:$0.
; Note that the first routine loads data starting with the least significant byte of P:$0 first.

; The second routine loads the internal PRAM using the HOST interface logic
; or the RSSI0 interface logic
; It will load 4,096 bytes from the parallel host processor interface if bit 15 of P:$C000 is cleared
; and from the Serial Synchronous Interface RSSI0 if bit 15 of P:$C000 is set.
; These will be condensed into 2,048 16-bit words and stored in contiguous internal PRAM memory
; locations starting at P:$0. Note that when using the RSSI0, the routine loads data starting with the
; least significant byte of P:$0 first.
; If the host processor only wants to load a portion of the p memory, and then start execution of
; the loaded program, the host interface bootstrap load program routine may be killed by setting
; HF0 = 1.
```

Figure A-1. DSP56166 Bootstrap Program Listing

```
          MOVE      P:(R1),A            ; Get P:$C000 (clears A0)
          MOVE      A0,R0               ; R0=(0) Starting P: address of
                                        ; internal memory where program
                                        ; will begin loading
          ROL       A                   ; Shift bit 15 into the carry flag
          BCC       <INLOOP             ; Perform load from memory or
                                        ; host interface if carry is zero.
          ORI       #$40,CCR            ; Set L bit if not (0)
          MOVE      R0,X:M_CRA0         ; Set CRA0 of RSSI0 to 8 bit mode
          MOVE      #$2400,A
          MOVE      A,X:M_CRB0          ; Set CRB0 to external clock, sync.mode,
                                        ; Late FS mode with reception enabled
          BFSET     #$E,X:(R3)          ; Set PC1,PC2,PC3 to SRD0,SCK0,RFS0
;
INLOOP    MOVE      #PRAMSIZE,B1        ; Load PRAM size into B1
          DO        B1,_LOOP1           ; Load PRAMSIZE instruction words.

          BFTSTH    #1,OMR              ; Perform load from Host
          BCC       <_MEMLD             ; Load from memory if MA=0.
;
; This is the second routine. It loads from the Host Interface pins or from the RSSI0 pins.
;
          BLC       <_HOSTLD            ; Load from Host Interface
                                        ; if the limit flag is clear
; Bootstrap byte per byte from RSSI0
;
_RSSILD   DO #2,_LOOP2
_RSSIWT   BFTSTL    #$80,X:M_SR0        ; Test RDF flag
          BCS       _RSSIWT             ; Wait for RDF to go high
          MOVEP     X:M_RX0,B           ; Put receive RSSI0 data in B
          ASR4      B
          ASR4      B
          BRA       <_PACK              ; where the received byte
;
; This is the first routine. Its loads from external P: memory
;
_MEMLD    DO        #2,_LOOP2           ; Each instruction has 2 bytes.
          MOVE      P:(R1)+,B           ; Get 8-bit from external P:
_PACK     MOVE      B1,A2               ; Move the 8-bit into A2
          ASR4      A                   ; Shift 4 bit data into A1
          ASR4      A                   ; Shift 4 bit data into A1
_LOOP2
          BRA       <_STORE             ; Then put the word in P: memory
```

Figure A-1. Listing of the DSP56166 Bootstrap Program (Continued)

```
;
; Bootstrap from the parallel host interface
;
_HOSTLD   BFSET    #1,X:(R2)          ; Configure Port B as Host Interface.
_LBLA     BFTSTH   #8,X:<<M_HSR       ; Test HF0.
          BRKCS                      ; Stop loading if HF0=1.
;
          BFTSTL   #1,X:M_HSR        ; Test HRDF flag
          BCS      _LBLA             ; Wait for HRDF to go high
                                     ; (meaning the data is present)
          MOVEP    X:M_HRX,A         ; Put 16-bit host data in X0
_STORE    MOVE     A,P:(R0)+         ; Store 16-bit result in PRAM
;
_LOOP1
;
          ANDI     #$FE,OMR          ; Clear OMR bit 0
          ORI      #$2,OMR           ; Set the operating mode to 2
                                     ; (and trigger an exit from
                                     ; bootstrap mode).
          AND      #$0,CCR           ; Clear SR as if HW reset and
                                     ; introduce delay needed for
                                     ; operating mode change.

          BRA      <$0               ; Start fetching from PRAM.
          END
```

Figure A-1. Listing of the DSP56166 Bootstrap Program (Continued)