APPENDIX E
BOOTSTRAP MODE — OPERATING MODE 1

The bootstrap feature of the DSP56001 consists of four special on-chip modules: the 512 words of PRAM, a 32-word bootstrap ROM, the bootstrap control logic, and the bootstrap firmware program.

BOOTSTRAP ROM

This 32-word on-chip ROM has been factory programmed to perform the actual bootstrap operation from the memory expansion port (Port A) or from the Host Interface. You have no access to the bootstrap ROM other than through the bootstrap process. Control logic will disable the bootstrap ROM during normal operations.

BOOTSTRAP CONTROL LOGIC

The bootstrap mode control logic is activated when the DSP56001 is placed in Operating Mode 1. The control logic maps the bootstrap ROM into program memory space as long as the DSP56001 remains in Operating Mode 1. The bootstrap firmware changes operating modes when the bootstrap load is completed. When the DSP56001 exits the reset state in Mode 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 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, or through the Host Interface.
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.

You may also select the bootstrap mode by writing Operating Mode 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 (e.g., see Bootstrap code for DSP56001) to begin the bootstrap process as described above in steps 1-4. This technique allows the DSP56001 user to reboot the system (with a different program if desired).

BOOTSTRAP FIRMWARE PROGRAM

Bootstrap ROM contains the bootstrap firmware program that performs initial loading of the DSP56001 PRAM. The program is written in DSP56000/DSP56001 assembly language. It contains two separate methods of initializing the PRAM: loading from a byte-wide memory starting at location P:$C000 or loading through the Host Interface. The particular method used is selected by the level of program memory location $C000, bit 23. If location P:$C000, bit 23 is read as a one, the external bus version of the bootstrap program will be selected. Typically, a byte wide EPROM will be connected to the DSP56001 Address and Data Bus as shown in 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:$0000 mid byte
P:$C002	P:$0000 high byte
•	•
•	•
•	•
P:$C5FD	P:$01FF low byte
P:$C5FE	P:$01FF mid byte
P:$C5FF	P:$01FF high byte

If location P:$C000, bit 23 is read as a zero, the Host Interface version of the bootstrap program will be selected. Typically a host microprocessor will be connected to the DSP56001 Host Interface. The host microprocessor must write the Host Interface registers THX, TXM, and then TSL with the desired contents of PRAM from location P:$0000 up to P:$01FF. If less than 512 words are to be loaded, the host programmer can exit the bootstrap program and force the DSP56001 to begin executing at location P:$0000 by setting HF0=1 in the Host Interface during the bootstrap load. In most systems, the DSP56001 responds so fast that handshaking between the DSP56001 and the host is not necessary.

The bootstrap program is shown in flowchart form in Figure E-1 and in assembler listing format in Figure E-2.

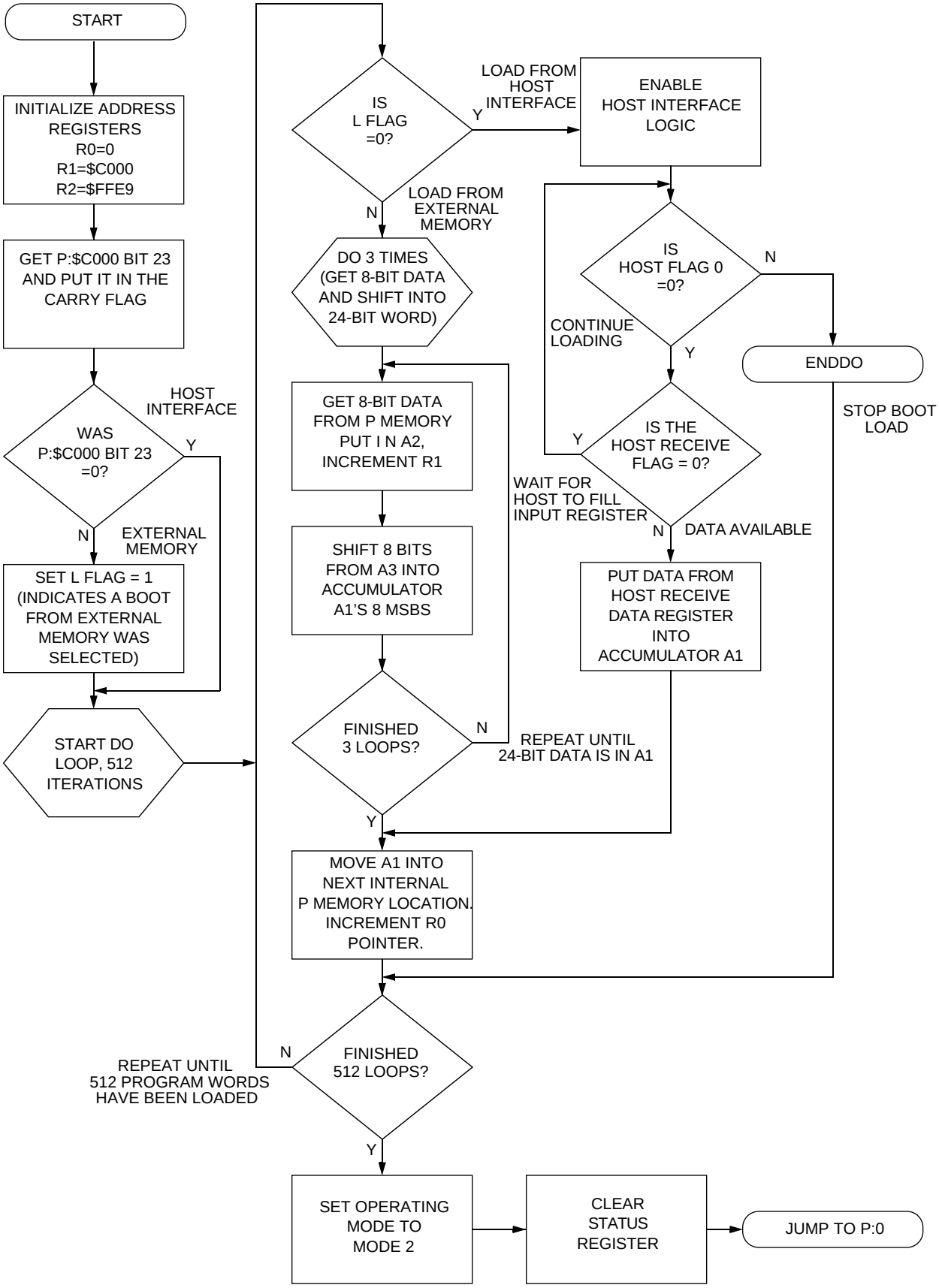

Figure E-1. Bootstrap Program Flowchart

```
1                                       PAGE 132,50,0,10
2                               ; BOOTSTRAP SOURCE CODE FOR DSP56001 - (C) Copyright 1986 Motorola Inc.
4                               ; Host algorithm / AND / external bus method
6                               ; This is the Bootstrap source code contained in the DSP56001 32 word boot ROM.
7                               ; This program can load the internal program memory from one of two external sources.
8                               ; The program reads P:$C000 bit 23 to decide which external source to access. If
9                               ; P:$C000 bit 23 = 0 then it loads internal PRAM from H0-H7, using the Host Interface
10                              ; logic. If P:$C000 bit 23 = 1 then it loads from 1,536 consecutive byte-wide P:
11                              ; memory locations (starting at P:$C000).
13  0000C000                    BOOT        EQU     $C000               ; The location in P: memory
14                                                                      ; where the external byte-wide
15                                                                      ; EPROM is expected to be mapped.
16
17  p:0000                              ORG     PL:$0                   ; Bootstrap code starts at P:$0
18
19  P:0000      62F400          START   MOVE                #$FFE9,R2   ; R2 = address of the Host
                00FFE9
20                                                                      ; Interface status register.
21  P:0002      61F400                  MOVE                #BOOT,R1    ; R1 = starting P: address of
                00C000
22                                                                      ; external bootstrap byte-wide ROM.
23  P:0004      300000                  MOVE                #0,R0       ; R0 = starting P: address of
24                                                                      ; internal memory where program
25                                                                      ; will begin loading.
26
27  P:0005      07E18C                  MOVE                P:(R1),Al   ; Get the data at P:$C000
28  P:0006      200037                  ROL     A                       ; Shift bit 23 into the Carry flag
29  P:0007      0E0009                  JCC     <INLOOP                 ; Perform load from Host Interface
30                                                                      ; if carry is zero.
31
32                              ; IMPORTANT NOTE: This routine assumes that the L bit has been cleared before entering
33                              ; this program and that M0 and M1 have been preloaded with $FFFF (linear addressing).
34                              ; This would be the case after a reset. If this program is entered by changing the OMR
```

Figure E-2. Assembler Listing for Bootstrap Program (Sheet 1 of 3)

```
                        ; to bootstrap operating mode, make certain that the L bit is cleared and registers M0
                        ; and M1have been set to $FFFF. Also, make sure the BCR is set to $xxFx since
                        ; EPROMS are slow and BCR is set to $FFFF after a reset. If the L bit was set before
                        ; changing modes, the program will load from external program memory.

P:0008    0040F9                ORI     #$40,CCR          ; Set the L bit to indicate
                                                          ; that the bootstrap program
                                                          ; is being loaded from the
                                                          ; external P: space.

                        ; The first routine will load 1,536 bytes from the external P: memory space beginning
                        ; at P:$C000 (bits 7-0). These will be packed into 512 24-bit words and stored in
                        ; contiguous internal PRAM memory locations starting at P:$0.

                        ; The shifter moves the 8-bit input data from register A2 into register A1 eight bits
                        ; at a time. After assembling one 24-bit word (this takes three loops) it stores the
                        ; result in internal PRAM and continues until internal PRAM is filled. 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.
                        ; If the host only wants to load a portion of the PRAM, the Host Interface bootstrap
                        ; load program can be aborted and execution of the loaded program started, by setting
                        ; the Host Flag (HF0) = 1 at any time during the load from the Host Processor.

P:0009    060082    INLOOP  D0      #512,_LOOP1       ; Load 512 instruction words.
          00001B

                        ; This is the context switch

P:000B    0E6012                JLC     < _HOSTLD         ; Load from the Host Interface
                                                          ; if the Limit flag is clear.

                        ; This is the first routine. It loads from external P: memory.

P:000C    060380                DO      #3, _LOOP2        ; Each instruction has 3 bytes.
          000010
```

Figure E-2. Assembler Listing for Bootstrap Program (Sheet 2 of 3)

```
69   P:000E    07D98A              MOVE              P:(R1)+,A2    ; Get the 8 LSB from external
70                                                                 ; P: memory.
71   P:000F    0608A0              REP     #8                     ; Shift 8 bit data into A1
72   P:0010    200022              ASR     A
73                       _LOOP2                                    ; Get another byte
74   P:0011    0C001B              JMP     < _STORE               ; then put the word in PRAM.
75
76                       ; This is the second routine. It loads from the Host Interface pins.
77
78   P:0012    0AA020    _HOSTLD   BSET    #0,X:$FFE0             ; Configure Port B as Host Interface
79   P:0013    0AA983    _LBLA     JCLR    #3,X:$FFE9, _LBLB      ; If HF0=1, stop loading data.
               000017
80   P:0015    00008C              ENDDO                          ; Must terminate the DO loop
81   P:0016    0C001C              JMP     <_BOOTEND
82
83   P:0017    0A6280    _LBLB     JCLR    #0,X:(R2), _LBLA       ; Wait for HRDF to go high
               000013
84                                                                 ; (meaning 24-bit data is present)
85   P:0019    54F000              MOVE              X:$FFEB,A1   ; Put 24-bit host data in A1
               00FFEB
86
87   P:001B    07588C    _STORE    MOVE              A1,P:(R0)+   ; Store 24-bit result in PRAM.
88
89                       _LOOP1                                    ; and return for another 24-bit word
90
91                       ; This is the exit handler that returns execution to normal expanded mode
92                       ; and jumps to the RESET location.
93
94   P:001C    0502BA    _BOOTEND  MOVEC             #2,0MR       ; Set the operating mode to 2
95                                                                 ; (and trigger an exit from
96                                                                 ; bootstrap mode).

97   P:001D    0000B9              ANDI    #$0,CCR                ; Clear SR as if RESET and
98                                                                 ; introduce delay needed for
99                                                                 ; Op. Mode change.
100  P:001E    0C0000              JMP     <$0                    ; Start fetching from PRAM P:$0000
101
```


0 Errors
0 Warnings

Figure E-2. Assembler Listing for Bootstrap Program (Sheet 3 of 3)