DIGITAL
SYSTEMS

1154 DUNSMUIR PL LIVERMORE, CA. 94550
Telephone (415) 443-4078

DIGITAL SYSTEMS EQUIPMENT WARRANTY

1. All parts and materials supplied by DIGITAL SYSTEMS are warranted to be free from defects in materials and workmanship for a period of ninety (90) days from date of shipment. During this period, all defective PC boards, modules, or systems returned freight prepaid to DIGITAL SYSTEMS will be repaired or replaced at no charge and returned to the customer freight prepaid.

2. Beyond the 90 day warranty period, equipment will be repaired or replaced according to the following schedule:

FDC-1 disk controller board	$60
HB-1 bus interface board	$15
FDS-1 single or dual system	$95
(Plus any disk drive charges-see Sec. 3)	

   Above prices are exclusive of freight charges.

3. Shugart disk drives carry a one year parts only warranty. Disk drive repair charges are:

   $55 in warranty
   $125 out of warranty

4. The above warranties are contingent upon proper use in the application for which equipment was intended and does not cover equipment which was modified without DIGITAL SYSTEMS' approval or which was subjected to unusual physical or electrical stress.

5. EXCEPT FOR THE EXPRESS WARRANTIES SET FORTH ABOVE, DIGITAL SYSTEMS GRANTS NO OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, ON EQUIPMENT, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, AND THE STATED EXPRESS WARRANTY IS IN LIEU OF ALL LIABILITIES OR OBLIGATIONS OF SELLER FOR DAMAGES INCLUDING, BUT NOT LIMITED TO, CONSEQUENTIAL DAMAGES OCCURRING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF DIGITAL SYSTEMS' PRODUCT.

ASSEMBLY INSTRUCTIONS

THIS IS A DESCRIPTION OF HOW A COMPLETE DUAL DRIVE SYSTEM IS PUT TOGETHER. IF YOU HAVE PURCHASED A COMPLETE SYSTEM, ALL CONNECTIONS SHOULD BE IN PLACE WHEN THE SYSTEM ARRIVES EXCEPT FOR THE CONNECTION TO THE INTERFACE CARD. NEVERTHELESS, YOUR SYSTEM SHOULD BE CHECKED FOR LOOSE CONNECTIONS BEFORE POWERING ON. IF YOU HAVE PURCHASED A BASIC SYSTEM, THE APPROPRIATE PARTS OF THIS DESCRIPTION WILL HELP YOU ASSEMBLE YOUR SYSTEM.

THE DISK DRIVES ARE BOLTED INTO THE CABINET WITH FOUR SCREWS TO THE FLOOR. THE A DRIVE IS ON THE RIGHT SIDE OF THE CABINET, BELOW THE ON/OFF SWITCH. THE SHUGART DISK DRIVE MANUAL TELLS HOW TO MAKE A DRIVE BE EITHER A OR B. THE PC BOARD SIDE OF THE DRIVE IS ON THE BOTTOM WHEN THE DRIVE IS MOUNTED.

POWER SUPPLIES ARE BOLTED ON THE BACK WALL. THE FOLLOWING COLOR CODE IS USED FOR CONNECTING WIRES.

+5V	BLUE
+24V	BROWN
-11V	RED
GROUND	GREEN
110V AC	RED AND WHITE TWISTED

THE FDC-2 REQUIRES +5V ON PIN 5 OF THE SIX PIN AMP CONNECTOR (J5) AND GROUND ON PIN 6. THE DISK DRIVE POWER REQUIREMENTS ARE OUTLINED IN THE SHUGART MANUAL PAGE ON 18.

SIGNALS FROM THE FDC-1 TO THE DRIVES ARE CARRIED IN A 50 CONDUCTOR FLAT CABLE MARKED WITH RED DOTS. THERE IS A SOCKET CONNECTOR ON ONE END AND ONE OR TWO PC BOARD CONNECTORS ON THE OTHER. THE PC BOARD CONNECTORS GO ON THE DISK DRIVES. THE PIN 1 EDGE OF THE CONNECTOR (RED EDGE OF CABLE) SHOULD BE MATCHED UP WITH PIN 1 END OF THE PC BOARD (MARKED WITH A RED DOT). THE SOCKET CONNECTOR ON THE CABLE PLUGS INTO CONNECTOR J3 ON THE FDC-1 WITH PIN 1 TOWARD THE CENTER OF THE BOARD.(ONCE AGAIN, MATCH UP THE RED DOTS).

THE CONNECTON BETWEEN THE FDC-1 AND THE INTERFACE CARD IS VIA ANOTHER 50 CONDUCTOR FLAT CABLE; THIS ONE IS MARKED WITH GREEN DOTS. THE PRINTED CIRCUIT BOARD CONNECTOR PLUGS INTO THE INTERFACE CARD. THE SOCKET CONNECTOR PLUGS INTO THE SOCKET ON THE CABLE HARNESS FOR THE FDC-1.

THE SYSTEM USES FUSE TYPE 3AG 5 AMP.

NOTE:

WE SUGGEST THAT THE INTERFACE CARD BE PLACED NEAR THE PROCESSOR CARD AT THE FRONT OF THE BUS TO MINIMIZE THE EFFECTS OF BUS NOISE.

FDS-1 SYSTEM OPERATION

THE FOLLOWING PROCEEDURE SHOULD BE USED TO BRING UP YOUR FDS-1 DISK SYSTEM:

1. INSERT THE INTERFACE CARD (HB-1) IN YOUR ALTAIR/IMSAI COMPATIBLE BUS. ATTACH THE CABLE FROM THE FDS-1 TO THE TOP EDGE CONNECTOR ON THE PC BOARD. IT IS BETTER TO PLACE THE INTERFACE CARD NEAR THE FRONT.

2. POWER ON BOTH THE FDS-1 AND YOUR COMPUTER. THE ACTIVITY LED ON DRIVE A (RIGHT DRIVE) SHOULD GO ON. IF NOT, RECHECK ALL CABLES, CONNECTORS AND POWER SUPPLY VOLTAGES.

3. PERFORM A STOP-RESET FROM THE FRONT PANEL.

4. INSERT A DISKETTE CONTAINING THE CP/M OPERATING SYSTEM INTO DRIVE A (LABEL SIDE UP) AND CLOSE THE DOOR. THE DRIVE SHOULD SEEK TRACK 0, TRANSFER SECTOR 1, AND THE ACTIVITY LIGHT SHOULD GO OUT.

5. EXAMINE MEMORY LOCATIONS 0-7FH AND COMPARE WITH THE LISTING OF THE BOOTSTRAP LOADER PROGRAM. IF IT HAS BEEN CORRECTLY LOADED, EXECUTE STARTING AT 0. THIS BOOTSTRAP LOADER SHOULD LOAD THE ENTIRE OPERATING SYSTEM OFF TRACKS 0 AND 1.

6. IF THE SYSTEM WILL NOT LOAD THE BOOTSTRAP PROGRAM, TRY THE PROCEEDURE AGAIN, WITH A DIFFERENT DISKETTE IF POSSIBLE. YOU SHOULD ALSO VERIFY THAT YOUR MACHINE IS STILL WORKING BY REMOVING THE INTERFACE CARD (POWER DOWN FIRST!) AND RUNNING A SMALL PROGRAM.

   INCORRECT MEMORY CONFIGURATION CAN ALSO CAUSE YOU PROBLEMS. YOU NEED RAM FROM LOCATIONS 0-7FH AND FROM 28FDH TO 3FFFH TO LOAD THE OPERATING SYSTEM. REMEMBER THAT THE FDC-1 USES AND RESTORES THE THREE BYTES BELOW THE DMA AREA (EXCEPT FOR THE BOOTSTRAP). TO FULLY UTILIZE ALL OF THE CP/M SYSTEM PROGRAMS, YOU SHOULD HAVE RAM FROM 00-3FFFH.

7. IF THE SYSTEM WILL NOT LOAD THE BOOTSTRAP PROGRAM, IT IS POSSIBLE TO ATTEMPT TO USE THE OTHER DRIVE IN A DUAL DRIVE SYSTEM. REMOVE THE DRIVES AND CONSULT THE SHUGART MANUAL FOR THE LOCATION OF THE DRIVE SELECT JUMPERS.

A>

BRINGING UP THE SOFTWARE

AS DESCRIBED IN THE CP/M SYSTEM ALTERATION GUIDE, DISK AND CONSOLE INPUT AND OUTPUT DRIVERS MUST BE ADDED BY THE USER TO CUSTOMIZE CP/M TO A SPECIFIC CONFIGURATION. THESE PATCHES ARE MADE IN A PROGRAM CALLED CBIOS WHICH IS LISTED IN THE ALTERATION GUIDE. CBIOS RESIDES FROM 3E00H TO 3FFFH IN A 16K SYSTEM.

DIGITAL SYSTEMS HAS ADDED THE DISK DRIVERS TO CBIOS TO CORRECTLY INTERFACE WITH THE FDS HARDWARE. THIS VERSION OF CBIOS OS ON THE DISKETTE YOU RECEIVE FROM DIGITAL SYSTEMS. A LISTING OF THIS VERSION IS INCLUDED IN YOUR DOCUMENTATION. ALSO INCLUDED IS CODE TO ENABLE YOU TO PATCH IN YOUR I/O DRIVERS AND MAKE A COPY OF THE COMPLETED SYSTEM ON A NEW DISKETTE. THIS CODE MAKES KEYING IN A GETSYS, PUTSYS, AND LOADER, AS DESCRIBED IN THE MANUAL, UNNECESSARY.

THE FOLLOWING PROCEEDURE SHOULD BE FOLLOWED IF IT IS NECESSARY TO PATCH IN YOUR OWN I/O. FIRST WRITE AND HAND ASSEMBLE A CONSOLE INPUT, OUTPUT, AND STATUS ROUTINE. THE ROUTINES WE USE FOR OUR CONFIGURATION ARE LISTED AS GUIDES. FOR THOSE TERMINALS INTERFACED VIA SERIAL BOARDS USING THE 8251 USART, AN INITIALIZATION MUST BE PERFORMED AFTER EVERY RESET. TO GET THE SYSTEM RUNNING, DO THIS FROM THE FRONT PANEL. LATER, INCORPORATE THIS INITIALIZATION INTO YOUR SYSTEM BY REASSEMBLING YOUR BOOTSTRAP LOADER AND DOING A SECOND LEVEL SYSTEM GENERATION. (SEE SYSTEM ALTERATION GUIDE)

FOR A FIRST LEVEL SYSTEM GENERATION, DO THE FOLLOWING:

1. BOOTSTRAP THE SYSTEM AS DESCRIBED IN THE SYSTEM OPERATION SECTION.
2. AFTER ALL DISK ACTIVITY STOPS, PRESS THE STOP SWITCH ON YOUR FRONT PANEL.

3. YOU SHOULD FIND THE MACHINE IN A LOOP AT LOCATION 3EC0H. IF THIS IS NOT THE CASE, GO TO THE SYSTEM OPERATION SECTION AND REVIEW YOUR STEPS AND TEST FOR ERROR CONDITIONS.

4. CONSULT THE CBIOS LISTING AND THE LISTING OF YOUR I/O DRIVERS AND PATCH IN THE CODE. MAKE SURE THAT YOUR ROUTINES ARE NOT SO LONG THAT THEY OVERLAY CODE IN OTHER ROUTINES.

   PATCH CONSOLE-IN STARTING AT 3EADH
   PATCH CONSOLE-OUT STARTING AT 3EC0H
   PATCH CONSOLE-STATUS STARTING AT 3E9AH

5. AFTER CHECKING YOUR CODE, START THE COMPUTER EXECUTING AT LOCATION 3EC0H.
6. THE CONSOLE SHOULD RESPOND WITH A>
   IF IT DOES NOT, THERE IS PROBABLY AN ERROR IN YOUR OUTPUT ROUTINE.
7. TO TEST THE INPUT ROUTINE AND TO SAVE THE COMPLETED SYSTEM, TYPE:

   WRITECPM

   THIS IS THE NAME OF THE PROGRAM THAT WILL SAVE YOUR PATCHES. IF THE SYSTEM ECHOED YOUR TYPING BACK TO YOU, THE I/O DRIVERS ARE WORKING CORRECTLY.

8. TO EXECUTE WRITECPM TYPE A CARRIAGE RETURN. THE PROGRAM WILL ASK YOU IF THE CORRECT DISKETTE IS IN THE DRIVE. IF SO, TYPE A RETURN. THE DISKETTE WILL NOW CONTAIN A PATCHED OPERATING SYSTEM THAT WILL COME UP ON YOUR CONSOLE ON A BOOTSTRAP.

IF YOU HAVE TO INITIALIZE YOUR I/O PORT, THERE ARE MANY PLACES IN THE BOOTSTRAP OR THE CBIOS THAT WOULD BE APPROPRIATE. THE FOLLOWING IS A WAY TO PATCH THE CBIOS TO ACCOMPLISH YOUR INITIALIZATION SUCH THAT THE PATCHES WILL BE SAVED BY WRITECPM DURING FIRST LEVEL SYSTEM ALTERATION.

NOTICE THAT THE CBIOS CODE ENDS AT 3FCFH. WRITECPM WILL ONLY SAVE PATCHES MADE IN THE AREA 3E00-3FFF. THEREFORE DO THE FOLLOWING:

LOC	NOW CONTAINS	CHANGE TO
3E00	C3	C3
3E01	75	D0
2	3E	3F
3FD0	??	YOUR INITIALIZATION CODE
-	-	-
-	-	-
-		-
-		C3
-		75
-		3E

PUT THESE PATCHES IN AT THE SAME TIME AS YOU DO YOUR I/O DRIVERS AS DESCRIBED ON THE PREVIOUS PAGE. YOU SHOULD START EXECUTION AT LOCATION 3E00 INSTEAD OF 3EC0 .

YOU WILL WANT TO REASSEMBLE YOUR CBIOS SOURCE FILE TO CONTAIN THESE AND YOUR CONSOLE DRIVERS. AT THAT TIME YOU CAN GET RID OF THE ABSOLUTE JUMPS.

REMEMBER TO DO A SOFTWARE RESET BEFORE INITIALIZING YOUR PORT.

A>

```
                        ;PATCH FOR I/O PORTS AS DEFINED BY IMSAI SIO BOARD
                        ;THE PORT INITIALIZATION WAS DONE IN THE BOOTSTRAP
                        ;
0002 =                  DATA    EQU     2
0003 =                  STAT    EQU     DATA+1
3E9A                            ORG     3E9AH
                        ;CONSOLE STATUS
                        CONST:
3E9A DB03                       IN      STAT
3E9C E602                       ANI     2
3E9E C2A33E                     JNZ     RDY
                        ;NO CHARACTER READY. RETURN A 00
3EA1 AF                         XRA     A
3EA2 C9                         RET
                        RDY:
                        ;SET CHARACTER READY
3EA3 3EFF                       MVI     A,0FFH
3EA5 C9                         RET
                        ;
3EAD                            ORG     3EADH
                        ;CONSOLE INPUT ROUTINE
                        CONIN:
3EAD DB03                       IN      STAT
3EAF E602                       ANI     02
3EB1 CAAD3E                     JZ      CONIN
3EB4 DB02                       IN      DATA
3EB6 E67F                       ANI     07FH
3EB8 C9                         RET
                        ;
3EC0                            ORG     3EC0H
                        ;CONSOLE OUTPUT
                        CONOUT:
3EC0 DB03                       IN      STAT
3EC2 E601                       ANI     1
3EC4 CAC03E                     JZ      CONOUT
3EC7 79                         MOV     A,C
3EC8 D302                       OUT     DATA
3ECA C9                         RET
3ECB                            END

A>
```

```
                ;       BOOTSTRAP LOADER
0000                    ORG     00H
                ;
                ;
                ;LOAD THE MONITOR AT 2900
2900 =          LOADP   EQU     2900H   ;LOAD POINT
3E00 =          BCP     EQU     3E00H   ;MONITOR
0040 =          HLAST   EQU     40H     ;H ADDRESS JUST PAST CBIOS
0015 =          LAST    EQU     21      ;LAST SECTOR ON TRK1 TO LOAD
0001 =          SKIP    EQU     1       ;NUMBER OF SECTORS TO SKIP
                ;
007E =          DMAH    EQU     126     ;HIGH ORDER DMA ADDR
007D =          DMAL    EQU     125     ;LOW ORDER DMA ADDR
007D =          REBOOT  EQU     125     ;REBOOT SYSTEM ON ERROR
0002 =          INIT0   EQU     (0 SHL 7) OR SKIP+1     ;TRK/SEC INITIAL
001B =          END0    EQU     (0 SHL 7) OR 27 ;END OF FIRST TRK
0081 =          INIT1   EQU     (1 SHL 7) OR 1  ;TRK/SEC INIT FOR TRK 1
0096 =          END1    EQU     (1 SHL 7) OR LAST+1     ;LAST POSITION
                ;
0004 =          T76     EQU     04H     ;TOWARD 76
0002 =          STT     EQU     02H     ;STEP TRACK
0002 =          SRF     EQU     02H     ;STEP READY FLAG
007F =          DOUT    EQU     7FH     ;DISK OUTPUT PORT
007F =          DINP    EQU     7FH     ;DISK INPUT PORT
0040 =          RDS     EQU     40H     ;READ SECTOR
00F0 =          ERR     EQU     0F0H    ;ERROR CONDITIONS
0008 =          IOF     EQU     08H     ;IO FINISH
                ;
                ;
0000 00                 NOP             ;SOME DYNAMIC MEMORIES PROHIBIT
                                        ;THIS BYTE IN THE BOOTSTRAP
                ;       INITIALIZE TRK/SEC AND DMA ADDRESS
                START:
0001 1E02               MVI     E,INIT0
0003 21FD28             LXI     H,LOADP-3
                ;
                READL:  ;READ LOOP
                ;       SEND DMA ADDRESS
0006 7C                 MOV     A,H
0007 D37E               OUT     DMAH
0009 7D                 MOV     A,L
000A D37D               OUT     DMAL
                ;
                ;       GET DATA FROM MEMORY
000C 46                 MOV     B,M
000D 23                 INX     H
000E 4E                 MOV     C,M
000F 23                 INX     H
0010 56                 MOV     D,M
                ;
                ;       SET UP TRACK AND SECTOR
0011 2B                 DCX     H
0012 7B                 MOV     A,E
0013 E61F               ANI     11111B  ;SECTOR NUMBER
0015 77                 MOV     M,A
                        ;
0016 2B                 DCX     H
0017 7B                 MOV     A,E
0018 17                 RAL
0019 17                 RAL
001A E601               ANI     1
001C 77                 MOV     M,A     ;TRACK NUMBER
```

```
                        ;
                        ;       PERFORM THE READ
001D 3E40               WRD:    MVI     A,RDS
001F D37F               WRD:    OUT     DOUT
                        ;
0021 DB7F               WRD:    IN      DINP
0023 E6F0               WRD:    ANI     ERR
0025 CA2B00             WRD:    JZ      NERROR
                        ;
                        ;       ERROR, REBOOT
0028 DB7D               WRD:    IN      125     ;REBOOT ON ERROR
002A 76                 WRD:    HLT
002B DB7F               NERROR: ;NO ERROR SO FAR
002D E608               NERROR: IN      DINP
002F CA2100             NERROR: ANI     IOF
                        NERROR: JZ      WRD     ;GO BACK AND WAIT
                        ;
                        ;       I/O FINISH, INCREMENT EVERYTHING.
                        ;       REPLACE DATA BYTES
0032 70                 NERROR: MOV     M,B
0033 23                 NERROR: INX     H
0034 71                 NERROR: MOV     M,C
0035 23                 NERROR: INX     H
0036 72                 NERROR: MOV     M,D     ;BYTES REPLACED
                        ;
                        ;       ADD 126 TO DMA ADDRESS
0037 017E00             NERROR: LXI     B,126
003A 09                 NERROR: DAD     B
                        ;       INC TRACK/SECTOR
003B 7B                 NERROR: MOV     A,E
003C 3C                 NERROR: INR     A
003D FE1B               NERROR: CPI     END0    ;END OF TRACK 0?
003F C25C00             NERROR: JNZ     CMP1
                        ;
                        ;       STEP TRACK TO TRACK 1
0042 3E04               NERROR: MVI     A,T76
0044 D37F               NERROR: OUT     DOUT
0046 F602               NERROR: ORI     STT
0048 D37F               NERROR: OUT     DOUT
004A 3E04               NERROR: MVI     A,T76
004C D37F               NERROR: OUT     DOUT
                        ;
                        WST:    ;WAIT FOR STEP AND HEAD SETTLE (18 MSEC)
004E 3E12               NERROR: MVI     A,18D
                        WST0:
0050 0E82               NERROR: MVI     C,82H   ;ONE MILLISEAOND TIMING FOR OUR
                                                ;CAN BE ADJUSTED FOR DIFFERENT
                        WST2:
0052 0D                 NERROR: DCR     C
0053 C25200             NERROR: JNZ     WST2
0056 3D                 NERROR: DCR     A
0057 C25000             NERROR: JNZ     WST0
                        ;
                        ;               TRACK STEPPED
005A 3E81               NERROR: MVI     A,INIT1
005C FE96               CMP1:   CPI     END1    ;END OF TRACK 1?
005E 5F                 NERROR: MOV     E,A     ;RESTORE TRK/SEC
005F C20600             NERROR: JNZ     READL   ;GO BACK FOR MORE
```

```
                ;
                ;       END OF LOAD
                ;SET LOC 4 TO CONTAIN ACTIVE DRIVE=0
0062 AF                 XRA     A
0063 320400             STA     04H
                ;NOW CALCULATE A CHECK SUM FROM BEGINNING OF LOAD
                ;POINT THRU ALL OF CBIOS
0066 AF                 XRA     A
0067 210029             LXI     H,LOADP
                L3:
006A 86                 ADD     M
006B 23                 INX     H
006C 47                 MOV     B,A     ;SAVE TOTAL
006D 7C                 MOV     A,H
006E FE40               CPI     HLAST
0070 78                 MOV     A,B     ;GET THE SUM BACK
0071 C26A00             JNZ     L3
                ;DISPLAY SUM ON THE IMSAI FRONT PANEL
0074 D3FF               OUT     255D
0076 C3003E             JMP     BCP     ;GO TO COMMAND PROCESSOR
                ;
007F                    ORG     7FH
007F FF                 DB      0FFH
0080                    END

A>
```

```
A>TYPE CBIOS1/4.PRN


                ;       SKELETAL CBIOS FOR FIRST LEVEL OF CP/M ALTERATIO
                ;
                ;       NOTE :  MSIZE DETERMINES WHERE THIS CBIOS IS LO
0010 =          MSIZE   EQU     16      ;CP/M VERSION MEMORY SIZE IN KI
3E00 =          PATCH   EQU     MSIZE*1024-2*256        ;START OF THE C
                ;
                ;       WE WILL USE A SCRATCH AREA STARTING AT 40H
                ;        FOR HOLDING THE VALUES OF:
                ;               TRACK   =  LAST SELECTED TRACK
                ;               SECTOR  =  LAST SELECTED SECTOR
                ;               DMAAD   =  LAST SELECTED DMA ADDRESS
                ;               DISKNO  =  LAST SELECTED DISK NUMBER
                ;       (NOTE THAT ALL ARE BYTE VALUES EXCEPT FOR DMAAD)
                ;
0040 =          SCRAT   EQU     40H             ;START OF SCRATCH AREA
0040 =          TRACK   EQU     SCRAT           ;CURRENT TRACK ON DRIVE0
0041 =          TRAK1   EQU     TRACK+1         ;CURRENT TRACK ON DRIVE1
0042 =          SECTOR  EQU     SCRAT+2         ;CURRENTLY SELECTED SEC
0043 =          DMAAD   EQU     SCRAT+3         ;CURRENT DMA ADDRESS
0045 =          DISKNO  EQU     SCRAT+5         ;CURRENT DISK NUMBER
0046 =          DUMMY   EQU     DISKNO+1        ;MUST BE 0 FOR DOUBLE AD
                ;
                ;
3E00                    ORG     PATCH   ;ORGIN OF THIS PROGRAM
0000 =          CBASE   EQU     (MSIZE-16)*1024 ;BIAS FOR SYSTEMS LARGE
2900 =          CPMB    EQU     CBASE+2900H     ;BASE OF CP/M (= BASE O
3206 =          BDOS    EQU     CBASE+3206H     ;BASE OF RESIDENT PORTI
1500 =          CPML    EQU     $-CPMB          ;LENGTH OF THE CP/M SYS
002A =          NSECTS  EQU     CPML/128        ;NUMBER OF SECTORS TO L
                ;
                ;       JUMP VECTOR FOR INDIVIDUAL SUBROUTINES
3E00 C3753E             JMP     GOCPM           ;COLD START
                WBOTE:
3E03 C32D3E             JMP     WBOOT           ;WARM START
3E06 C39A3E             JMP     CONST           ;CONSOLE STATUS
3E09 C3AD3E             JMP     CONIN           ;CONSOLE CHARACTER IN
3E0C C3C03E             JMP     CONOUT          ;CONSOLE CHARACTER OUT
3E0F C3D83E             JMP     LIST            ;LIST CHARACTER OUT
3E12 C3DA3E             JMP     PUNCH           ;PUNCH CHARACTER OUT
3E15 C3DC3E             JMP     READER          ;READER CHARACTER OUT
3E18 C3E13E             JMP     HOME            ;MOVE HEAD TO HOME POSI
3E1B C3FD3E             JMP     SELDSK          ;SELECT DISK
3E1E C30E3F             JMP     SETTRK          ;SET TRACK NUMBER
3E21 C3313F             JMP     SETSEC          ;SET SECTOR NUMBER
3E24 C3363F             JMP     SETDMA          ;SET DMA ADDRESS
3E27 C33D3F             JMP     READ            ;READ DISK
3E2A C3423F             JMP     WRITE           ;WRITE DISK
                ;
```

```
                ;       INDIVIDUAL SUBROUTINES TO PERFORM EACH FUNCTION
                ;
                WBOOT:  ;SIMPLEST CASE IS TO READ THE DISK UNTIL ALL SE
                ;
3E2D 318000             LXI     SP,80H          ;USE SPACE BELOW BUFFER
3E30 0E00               MVI     C,0             ;SELECT DISK 0
3E32 CDFD3E             CALL    SELDSK
3E35 CDE13E             CALL    HOME            ;GO TO TRACK 00
                ;
3E38 062A               MVI     B,NSECTS        ;B COUNTS THE NUMBER OF
3E3A 0E00               MVI     C,0             ;C HAS THE CURRENT TRAC
3E3C 1602               MVI     D,2             ;D HAS THE NEXT SECTOR
                ;       NOTE THAT WE BEGIN BY READING TRACK 0, SECTOR 2
                ;       CONTAINS THE COLD START LOADER, WHICH IS SKIPPE
3E3E 210029             LXI     H,CPMB          ;BASE OF CP/M (INITIAL
                LOAD1:  ;LOAD ONE MORE SECTOR
3E41 C5                 PUSH    B       ;SAVE SECTOR COUNT, CURRENT TRAC
3E42 D5                 PUSH    D       ;SAVE NEXT SECTOR TO READ
3E43 E5                 PUSH    H       ;SAVE DMA ADDRESS
3E44 4A                 MOV     C,D     ;GET SECTOR ADDRESS TO REGISTERC
3E45 CD313F             CALL    SETSEC  ;SET SECTOR ADDRESS FROM REGISTE
3E48 C1                 POP     B       ;RECALL DMA ADDRESS TO B,C
3E49 C5                 PUSH    B       ;REPLACE ON STACK FOR LATER RECA
3E4A CD363F             CALL    SETDMA  ;SET DMA ADDRESS FROM B,C
                ;
                ;       DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA ADDRE
3E4D CD3D3F             CALL    READ
3E50 B7                 ORA     A       ;ANY ERRORS?
3E51 C22D3E             JNZ     WBOOT   ;RETRY THE ENTIRE BOOT IF AN ERR
                ;
                ;       NO ERROR, MOVE TO NEXT SECTOR
3E54 E1                 POP     H       ;RECALL DMA ADDRESS
3E55 118000             LXI     D,128   ;DMA=DMA+128
3E58 19                 DAD     D       ;NEW DMA ADDRESS IS IN H,L
3E59 D1                 POP     D       ;RECALL SECTOR ADDRESS
3E5A C1                 POP     B       ;RECALL NUMBER OF SECTORS REMAI
3E5B 05                 DCR     B       ;SECTORS=SECTORS-1
3E5C CA753E             JZ      GOCPM   ;TRANSFER TO CP/M IF ALL HAVE BE
                ;
                ;       MORE SECTORS REMAIN TO LOAD, CHECK FOR TRACK CH
3E5F 14                 INR     D
3E60 7A                 MOV     A,D     ;SECTOR=27?, IF SO, CHANGE TRACK
3E61 FE1B               CPI     27
3E63 DA413E             JC      LOAD1   ;CARRY GENERATED IF SECTOR<27
                ;
                ;       END OF CURRENT TRACK, GO TO NEXT TRACK
3E66 1601               MVI     D,1     ;BEGIN WITH FIRST SECTOR OF NEX
3E68 0C                 INR     C       ;TRACK=TRACK+1
                ;
                ;       SAVE REGISTER STATE, AND CHANGE TRACKS
3E69 C5                 PUSH    B
3E6A D5                 PUSH    D
3E6B E5                 PUSH    H
3E6C CD0E3F             CALL    SETTRK  ;TRACK ADDRESS SET FROM REGISTER
3E6F E1                 POP     H
3E70 D1                 POP     D
3E71 C1                 POP     B
3E72 C3413E             JMP     LOAD1   ;FOR ANOTHER SECTOR
```

```
                        ;
                        ;       END OF LOAD OPERATION, SET PARAMETERS AND GO TO
                        GOCPM:
3E75 318000                     LXI     SP,80H
3E78 3EC3                       MVI     A,0C3H  ;C3 IS A JMP INSTRUCTION
3E7A 320000                     STA     0       ;FOR JMP TO WBOOT
3E7D 21033E                     LXI     H,WBOTE ;WBOOT ENTRY POINT
3E80 220100                     SHLD    1       ;SET ADDRESS FIELD FOR JMP AT 0
                        ;
3E83 320500                     STA     5       ;FOR JMP TO BDOS
3E86 210632                     LXI     H,BDOS  ;BDOS ENTRY POINT
3E89 220600                     SHLD    6       ;ADDRESS FIELD OF JUMP AT 5 TO B
                        ;
3E8C 018000                     LXI     B,80H   ;DEFAULT DMA ADDRESS IS 80H
3E8F CD363F                     CALL    SETDMA
                        ;
                        ;PUT ACTIVE DISK NUMBER (STORED IN LOCATION 4) IN C
3E92 3A0400                     LDA     04H
3E95 4F                         MOV     C,A
3E96 FB                         EI              ;ENABLE INTERRUPTS
                        PATLP:
                        ;WRITECPM ASSUMES THIS LOCATION
3E97 C30029                     JMP     CPMB    ;GO TO CP/M FOR FURTHER PROCESS
                        ;
                        ;
                        ;       SIMPLE I/O HANDLERS (MUST BE FILLED IN BY USER)
                        ;       IN EACH CASE, THE ENTRY POINT IS PROVIDED, WITH
                        ;       TO INSERT YOUR OWN CODE
                        ;
                        CONST:  ;CONSOLE STATUS, RETURN 0FFH IF CHARACTER READY
                        ;SPACE FOR STATUS SUBROUTINE
3E9A 00000000                   DB      0,0,0,0
3E9E 00000000                   DB      0,0,0,0
3EA2 00000000                   DB      0,0,0,0
3EA6 00000000                   DB      0,0,0,0
3EAA 3E00                       MVI     A,00H
3EAC C9                         RET
                        ;
                        CONIN:  ;CONSOLE CHARACTER INTO REGISTER A
                        ;SPACE FOR INPUT ROUTINE
3EAD 00000000                   DB      0,0,0,0
3EB1 00000000                   DB      0,0,0,0
3EB5 00000000                   DB      0,0,0,0
3EB9 00000000                   DB      0,0,0,0
3EBD E67F                       ANI     7FH     ;STRIP PARITY BIT
3EBF C9                         RET
                        ;
                        CONOUT: ;CONSOLE CHARACTER OUTPUT FROM REGISTER C
3EC0 C3C03E                     JMP     CONOUT  ;HANGS HERE SO USER CAN STOP AN
                        ;SPACE FOR OUTPUT ROUTINE
3EC3 00000000                   DB      0,0,0,0
3EC7 00000000                   DB      0,0,0,0
3ECB 00000000                   DB      0,0,0,0
3ECF 00000000                   DB      0,0,0,0
3ED3 00000000                   DB      0,0,0,0
3ED7 C9                         RET
```

```
                        ;
                        LIST:   ;LIST CHARACTER FROM REGISTER C
3ED8 79                         MOV     A,C     ;CHARACTER TO REGISTER A
3ED9 C9                         RET             ;NULL SUBROUTINE
                        ;
                        PUNCH:  ;PUNCH CHARACTER FROM REGISTER C
3EDA 79                         MOV     A,C     ;CHARACTER TO REGISTER A
3EDB C9                         RET             ;NULL SUBROUTINE
                        ;
                        ;
                        READER: ;READ CHARACTER INTO REGISTER A FROM READER DEVI
3EDC 3E1A                       MVI     A,1AH   ;ENTER END OF FILE FOR NOW (REP
3EDE E67F                       ANI     7FH     ;REMEMBER TO STRIP PARITY BIT
3EE0 C9                         RET
                        ;
                        ;
                        ;       I/O DRIVERS FOR THE DISK FOLLOW
                        ;
                        HOME:   ;MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
3EE1 3A4500                     LDA     DISKNO  ;SELECTED DISK
3EE4 4F                         MOV     C,A     ;FOLLOW PARAMETER CONVENTIONS
3EE5 CDFD3E                     CALL    SELDSK  ;ROUTINE TO SELECT THE DISK
                        ;SET UP H,L TO POINT TO WORD WITH TRACK FOR SELECTED DIS
3EE8 114000                     LXI     D,TRACK
3EEB 2A4500                     LHLD    DISKNO
3EEE 19                         DAD     D
                        HOMEL:
3EEF 3600                       MVI     M,00    ;SET CURRENT TRACK PTR BACK TO 0
3EF1 DB7F                       IN      127     ;READ FDC STATUS
3EF3 E604                       ANI     4       ;TEST TRACK 0 BIT
3EF5 C0                         RNZ             ;RETURN IF AT 0
3EF6 37                         STC             ;DIRECTION=OUT
3EF7 CDB03F                     CALL    STEP    ;STEP ONE TRACK
3EFA C3EF3E                     JMP     HOMEL   ;LOOP
                        ;
                        SELDSK: ;SELECT DISK GIVEN BY REGISTER C
                        ;MAKE SURE DUMMY IS 0 (FOR USE IN DOUBLE ADD TO H,L)
3EFD AF                         XRA     A
3EFE 324600                     STA     DUMMY
3F01 79                         MOV     A,C
3F02 324500                     STA     DISKNO
3F05 0F                         RRC             ;PUT INTO BITS 4,5
3F06 0F                         RRC
3F07 0F                         RRC
3F08 0F                         RRC
3F09 F608                       ORI     08      ;ENABLE DISK SELECT
3F0B D37F                       OUT     127     ;SELECT THE DISK
3F0D C9                         RET
                        ;
                        SETTRK: ;SET TRACK GIVEN BY REGISTER C
                        ;FIRST REFERENCE CORRECT TRACK INDICATOR ACCORDING TO
                        ;SELECTED DISK
3F0E 114000                     LXI     D,TRACK ;ADDRESS OF TRACK FOR DISK 0
3F11 2A4500                     LHLD    DISKNO  ;FIND OUT WHICH DISK IS SELECTED
3F14 19                         DAD     D
3F15 79                         MOV     A,C     ;DESIRED TRACK
3F16 BE                         CMP     M
3F17 C8                         RZ              ;WE ARE ALREADY ON THE TRACK
                        SETTKX:
3F18 CDB03F                     CALL    STEP    ;STEP TRACK-CARRY HAS DIRECTION
                                                ;STEP WILL UPDATE TRACK INDICATO
```

```
3F1B 79                 MOV     A,C
3F1C BE                 CMP     M       ;ARE WE WHERE WE WANT TO BE
3F1D C2183F             JNZ     SETTKX  ;NOT YET
                ;HAVE STEPPED ENOUGH
                SEEKRT:
                ;DELAY 18 MSEC FOR FINAL STEP TIME AND HEAD SETTLE TIME
3F20 3E12               MVI     A,18D
3F22 CD263F             CALL    DELAY
3F25 C9                 RET             ;END OF SETTRK ROUTINE
                ;
                DELAY:  ;ROUTINE TO DELAY C(A) MILLISECONDS
3F26 0E82               MVI     C,82H   ;ADJUST FOR 1 MSEC LOOP DELAY
                                        ;THIS IS THE VALUE FOR OUR IMSAI
                LDXA:
3F28 0D                 DCR     C
3F29 C2283F             JNZ     LDXA    ;LOOP 1 MSEC
3F2C 3D                 DCR     A
3F2D C2263F             JNZ     DELAY
3F30 C9                 RET             ;END OF DELAY ROUTINE
                ;
                SETSEC: ;SET SECTOR GIVEN BY REGISTER C
3F31 79                 MOV     A,C
3F32 324200             STA     SECTOR
3F35 C9                 RET
                ;
                SETDMA: ;SET DMA ADDRESS GIVEN BY REGISTERS B AND C
3F36 69                 MOV     L,C     ;LOW ORDER ADDRESS
3F37 60                 MOV     H,B     ;HIGH ORDER ADDRESS
3F38 224300             SHLD    DMAAD   ;SAVE THE ADDRESS
3F3B C9                 RET
                ;
                ;
3F3C 00         ERRORS: DB      0       ;KEEP TRACK OF NUMBER OF ERRORS
                READ:   ;PERFORM READ OPERATION.
                        ;THIS IS SIMILAR TO WRITE, SO SET UP READ COMMA
                        ;COMMON CODE IN WRITE
3F3D 1640               MVI     D,40H   ;SET READ FLAG
3F3F C3443F             JMP     WAITIO  ;TO PERFORM THE ACTUAL I/O
                ;
                WRITE:  ;PERFORM A WRITE OPERATION
3F42 1680               MVI     D,80H   ;SET WRITE COMMAND
                ;
                WAITIO:
                ;ENTER HERE FROM READ AND WRITE TO PERFORM THE ACTUAL I0
                ;OPERATION.  RETURN A 00H IN REGISTER A IF THE OPERATIO
                ;PROPERLY, AND 01H IF AN ERROR OCCURS DURING THE READ O
                ;
                ;IN THIS CASE, WE HAVE SAVED THE DISK NUMBER IN 'DISKNO
                ;                       THE TRACK NUMBER IN 'TRACK' (0-
                ;                       THE SECTOR NUMBER IN 'SECTOR' (
                ;                       THE DMA ADDRESS IN 'DMAAD' (3-3
                                        ;D STILL HAS R/W FLAG
3F44 3E0A               MVI     A,10D   ;SET ERROR COUNT
3F46 323C3F             STA     ERRORS  ;RETRY SOME FAILURES 10 TIMES
                                        ;BEFORE GIVING UP
```

```
                        TRYAGN:
                        ;FIRST WE HAVE TO FIGURE OUT WHICH DRIVE IS SELECTED
                        ;AND WHICH TRACK IS DESIRED
3F49 014000                     LXI     B,TRACK
3F4C 2A4500                     LHLD    DISKNO
3F4F 09                         DAD     B       ;H,L POINT TO CORRECT TRACK IND
3F50 7E                         MOV     A,M
3F51 F5                         PUSH    PSW     ;NEED IT LATER
3F52 2A4300                     LHLD    DMAAD   ;GET BUFFER ADDRESS
3F55 2B                         DCX     H       ;SAVE AND REPLACE 3 BYTES BELOW
                                                ;BUF WITH TRACK,SECTOR,ADDRESS M
3F56 46                         MOV     B,M
3F57 3EFB                       MVI     A,0FBH  ;ADDRESS MARK
3F59 77                         MOV     M,A
3F5A 2B                         DCX     H
3F5B 4E                         MOV     C,M
3F5C 3A4200                     LDA     SECTOR  ;NOTE THAT INVALID SECTOR NUMBER
                                                ;WILL RESULT IN HEAD UNLOADED
                                                ;ERROR, SO DONT CHECK
3F5F 77                         MOV     M,A
3F60 2B                         DCX     H
3F61 5E                         MOV     E,M
3F62 F1                         POP     PSW
3F63 77                         MOV     M,A
3F64 7C                         MOV     A,H     ;SET UP FDC DMA ADDRESS
3F65 D37E                       OUT     126     ;HIGH BYTE
3F67 7D                         MOV     A,L
3F68 D37D                       OUT     125     ;LOW BYTE
3F6A 7A                         MOV     A,D     ;GET R/W FLAG
3F6B D37F                       OUT     127     ;START DISK READ/WRITE
3F6D DB7F               RWWAIT: IN      127     ;READ FDC STATUS
3F6F E6F8                       ANI     0F8H    ;TEST FOR ANY ERROR OR IOF
3F71 CA6D3F                     JZ      RWWAIT
3F74 73                         MOV     M,E     ;RESTORE 3 BYTES BELOW BUF
3F75 23                         INX     H
3F76 71                         MOV     M,C
3F77 23                         INX     H
3F78 70                         MOV     M,B
3F79 DB7F                       IN      127     ;TEST FOR ERRORS
3F7B E6F0                       ANI     0F0H
3F7D C8                         RZ              ;A WILL BE 0 IF NO ERRORS
                        ;COME HERE ON ERROR FROM DISK
3F7E F5                         PUSH    PSW     ;SAVE ERROR CONDITION
                        ;CHECK FOR 10 ERRORS
3F7F 213C3F                     LXI     H,ERRORS
3F82 35                         DCR     M
3F83 C2943F                     JNZ     REDO    ;NOT TEN YET.  DO A RETRY
                        ;WE HAVE TOO MANY ERRORS. PRINT OUT HEX NUMBER FOR LAST
                        ;RECEIVED ERROR TYPE. CPM WILL PRINT PERM ERROR MESSAGE.
3F86 F1                         POP     PSW     ;GET CODE
3F87 0F                         RRC
3F88 0F                         RRC
3F89 0F                         RRC
3F8A 0F                         RRC
                        ;MAKE IT ASCII
3F8B F630                       ORI     030H
3F8D 4F                         MOV     C,A
3F8E CDC03E                     CALL    CONOUT
```

```
                        ;SET ERROR RETURN FOR OPERATING SYSTEM
3F91 3E01               MVI     A,1
3F93 C9                 RET
                        REDO:
                        ;D STILL HAS READ/WRITE FLAG
3F94 F1                 POP     PSW     ;GET ERROR CODE
3F95 E6E0               ANI     0E0H    ;RETRY IF NOT TRACK ERROR
3F97 C2493F             JNZ     TRYAGN
                        ;WAS A TRACK ERROR SO NEED TO RESEEK
3F9A D5                 PUSH    D       ;SAVE   READ/WRITE INDICATOR
                        ;FIGURE OUT THE DESIRED TRACK
3F9B 114000             LXI     D,TRACK
3F9E 2A4500             LHLD    DISKNO  ;SELECTED DISK
3FA1 19                 DAD     D       ;POINT TO CORRECT TRACK INDICATO
3FA2 7E                 MOV     A,M     ;DESIRED TRACK
3FA3 F5                 PUSH    PSW     ;SAVE IT
3FA4 CDE13E             CALL    HOME
3FA7 F1                 POP     PSW
3FA8 4F                 MOV     C,A
3FA9 CD0E3F             CALL    SETTRK
3FAC D1                 POP     D       ;GET READ/WRITE INDICATOR
3FAD C3493F             JMP     TRYAGN
                        ;
                        ;
                        ;
                        STEP:                           ;STEP HEAD OUT TOWARDS 0
                                                        ;IF CARRY IS SET; ELSE
                                                        ;STEP IN
                        ; H,L POINT TO CORRECT TRACK INDICATOR WORD
3FB0 F5                 PUSH    PSW     ;SAVE DIRECTION
3FB1 DB7F               STWAIT: IN      127     ;INPUT FDC STATUS
3FB3 E602               ANI     2       ;TEST STEP READY BIT
3FB5 CAB13F             JZ      STWAIT  ;WAIT FOR STEP READY(MAY 10 MSEC
3FB8 F1                 POP     PSW     ;GET DIRECTION TO STEP
3FB9 DACA3F             JC      OUTX
3FBC 34                 INR     M       ;INCREMENT CURRENT TRACK BYTE
3FBD 3E04               MVI     A,4     ;SET DIRECTION = IN
                        DOSTEP:
3FBF D37F               OUT     127     ;SET DIRECTION BIT IN FDC
3FC1 F602               ORI     2
3FC3 D37F               OUT     127     ;PULSE STEP BIT
3FC5 E6FD               ANI     0FDH
3FC7 D37F               OUT     127     ;TURN OFF PULSE
3FC9 C9                 RET
                        ;
                        OUTX:
3FCA 35                 DCR     M       ;UPDATE TRACK BYTE
3FCB AF                 XRA     A       ;SET DIRECTION = OUT
3FCC C3BF3F             JMP     DOSTEP
                        ;
                        ;
                        ;
3FCF                    END

A>
```

RELOCATING THE OPERATING SYSTEM

BEFORE YOU CAN RELOCATE THE OPERATING SYSTEM, YOU MUST BRING UP A 16K SYSTEM BY FOLLOWING THE INSTRUCTIONS UNDER "BRINGING UP THE SOFTWARE". ALSO, BE SURE TO READ THE SYSTEM ALTERATION GUIDE, PAYING SPECIAL ATTENTION TO SECOND LEVEL SYSTEM GENERATION. YOU SHOULD ALSO BE FAMILIAR WITH HOW DDT AND THE SYSGEN PROGRAM WORK.

ONCE YOU ARE UP AND RUNNING, YOU CAN USE RELOC.COM TO CREATE A CP/M SYSTEM FOR ANY MEMORY SIZE UP TO 64K BYTES. THE RELOC PROGRAM IS CALLED WITH

RELOC XX *<CARRIAGE-RETURN>

HERE, XX IS THE MEMORY SIZE IN DECIMAL KILOBYTES. THE * CAUSES THE GENERATED SYSTEM TO BE LEFT IN MEMORY FOR A SYSGEN OR A SAVE OPERATION.

WE HAVE ADDED OUR CBIOS AND BOOTSTRAP (WITHOUT THE CHECKSUM) TO THE VERSION FURNISHED BY DIGITAL RESEARCH. YOU MUST RELOCATE ANY CHANGES YOU MADE TO THE CBIOS SUCH AS YOUR I/O DRIVERS. YOU MUST ALSO ADD ANY CHANGES YOU MADE TO THE BOOTSTRAP. WRITECPM IS ONLY USED ON 16K SYSTEMS.

THERE ARE 2 METHODS YOU CAN USE.

METHOD 1.

INCORPORATE ALL OF YOUR I/O CHANGES INTO THE CBIOS SOURCE. (REMEMBER- NEVER ALTER THE ORDER OF THE JUMP TABLE AT THE BEGINNING OF THE PROGRAM!) DECIDE ON A NEW SYSTEM SIZE, AND CHANGE THE VARIABLE MSIZE TO REFLECT THIS. FOR A 32K SYSTEM, YOU SHOULD HAVE

```
MSIZE   EQU 32
```

REASSEMBLE THE PROGRAM. YOU WILL NOTICE THAT FOR THE 32K SYSTEM THE CBIOS WILL BEGIN AT 7E00H.

IF YOU HAVE CHANGED THE BOOTSTRAP A LOT, YOU MAY WANT TO REASSEMBLE IT MAKING SURE TO CHANGE LOADP AND BCP TO THE PROPER MEMORY SIZE.

NOW USE RELOC.COM TO RELOCATE THE SYSTEM (THIS EXAMPLE WILL BE A 32K SYSTEM), AND DDT TO APPEND YOUR CBIOS WHICH YOU RELOCATED WHEN YOU ASSEMBLED.

```
YOU TYPE        RELOC 32 *
SYSTEM SAYS     READY FOR "SYSGEN OR
                "SAVE 32 CPM32.COM"
YOU             SAVE 32 CPM32.COM
YOU             DDT CPM32.COM
```

YOU NOW HAVE TO DECIDE WHAT OFFSET TO USE TO READ IN YOUR CBIOS INTO THE CORRECT PART OF MEMORY FOR USE BY SYSGEN. SYSGEN STORES THE SYSTEM BEGINNING AT 900H, AND THE CBIOS AREA STARTS AT 1E80H. REMEMBER THAT YOUR 32K CBIOS WAS ASSEMBLED TO BEGIN AT 7E00H. THUS

```
YOU TYPE        H1E80 7E00
DDT RESPONDS    9C80 A080
```

SO A080 IS THE OFFSET YOU USE. NOW THAT YOU HAVE THIS, READ IN YOUR CBIOS

```
TYPE            ICBIOS.HEX
AND THEN        RA080
```

IF YOU DISPLAY 1E80 YOU SHOULD SEE YOUR CBIOS.

IF YOU HAVE REASSEMBLED YOUR BOOT, READ IT IN NOW WITH

```
IBOOT.HEX
R900
```

SINCE THE BOOTSTRAP IS IN THE MEMORY IMAGE FROM 900-97F.
IF YOU HAVE ONLY MADE SMALL BOOTSTRAP CHANGES, YOU CAN PATCH THEM DIRECTLY INTO THE AREA FROM 900-97F.

NOW GET OUT OF DDT BY G0. THE COMPLETED COPY OF YOUR NEW SYSTEM IS IN MEMORY. YOU CAN EITHER DO A SAVE AND THEN A SYSGEN OR A SYSGEN ALONE.

METHOD 2:

YOU CAN SIMPLY PATCH IN YOUR I/O DRIVERS INSTEAD OF REASSEMBLING THE CBIOS SINCE RELOC.COM RELOCATES CBIOS AS WE PROVIDE IT. YOU CAN EITHER ASSEMBLE A PATCH AT THE PROPER LOCATION AND READ IT IN WITH AN OFFSET AS DESCRIBED ABOVE, OR PATCH DIRECTLY INTO THE SYSGEN BUFFER AREA USING DDT.

A>

IBM FORMAT INITIALIZATION

IF YOU HAVE PURCHASED THE 2 INITIALIZATION MICROCODE ROMS, YOU HAVE THE CABABILITY TO INITIALIZE DISKETTES TO THE IBM FORMAT. THIS FORMAT IS DESCRIBED IN THE PROGRAM DISCRIPTION THAT FOLLOWS. ROM IH GOES IN CHIP LOCATION A3 ON THE FDC-1, AND ROM IL GOES INTO LOCATION A4. THE FOLLOWING PROCEEDURE SHOULD BE FOLLOWED TO FORMAT YOUR DISKETTES.

1. RUN THE PROGRAM INITDISK THAT WAS INCLUDED WITH YOUR SYSTEM DISKETTE.

2. THE PROGRAM WILL RESPOND WITH THE PROMPT

DISK INITIALIZATION ROMS SELECTED-YES OR NO

TO SELECT THE FORMAT ROMS YOU SHOULD SUPPLY A GROUND TO SIGNAL AUXROM WHICH IS ON PIN 46 OF THE FLAT CABLE CONNECTOR ON THE HB-1 INTERFACE CARD. (NO CONNECTION NOW EXISTS TO THIS SIGNAL.) THIS SIGNAL CAN ALSO BE FOUND ON PIN J1-60 OF THE FDC-1.

3. AFTER GROUNDING THE SIGNAL, TYPE Y FOR YES.

4. THE PROGRAM WILL RESPOND WITH

DISKETTE TO BE INITIALIZED IN DRIVE A?

AT THIS POINT YOU SHOULD MAKE SURE THAT THE CORRECT DISKETTE IS IN THE DRIVE. FORMATTING THE DISKETTE DOES DESTROY ANY PROGRAM OR DATA THAT YOU MAY HAVE ON THE DISKETTE. THE DISK DRIVE MUST BE WRITE ENABLED.

5. TYPE A RETURN. THE PROGRAM WILL NOW FORMAT THE DISKETTE AND RESPOND WITH THE FIRST MESSAGE. IF YOU WANT TO DO ANOTHER, TYPE YES AGAIN.

6. WHEN YOU ARE FINISHED, REMOVE THE GROUND AND TYPE NO IN RESPONSE TO THE FIRST PROMPT. THE PROGRAM WILL TELL YOU TO PUT A SYSTEM DISK BACK INTO THE DRIVE. TYPE A RETURN, AND THE SYSTEM WILL REBOOT.

THE FDC-1 MICROCODE EXPECTS THE HOST MACHINE TO SUPPLY THE DATA TO BE WRITTEN ON THE DISKETTE WITH THE FOLLOWING EXCEPTIONS:

1. THE FDC-1 WILL CALCULATE AND STORE ANY NECESSARY CRC BYTES.

2. THE FDC-1 WILL PLACE APPPOPRIATE INTER-FIELD GAPS ON THE DISKETTE.

AN ENTIRE TRACK'S WORTH OF DATA MUST BE SET UP IN THE DMA AREA FOR USE BY THE FDC-1 SINCE THE ENTIRE TRACK IS WRITTEN AT ONCE. THERE ARE 26 SECTORS PER TRACK; THUS THE MEMORY AREA SHOULD CONTAIN THE FOLLOWING:

NUMBER OF BYTES	CONTENTS (HEXADECIMAL)	DESCRIPTION
1	FC	INDEX MARK
1	FE	ID ADDRESS MARK
1	00	CURRENT TRACK NUMBER
1	00	BYTE OF 00
1	01	CURRENT SECTOR NUMBER
1	00	BYTE OF 00
1	FB	DATA ADDRESS MARK
128	E5	DATA
1	00	END OF SECTOR CODE
1	FE	ID ADDRESS MARK
1	00	CURRENT TRACK NUMBER
1	00	BYTE OF 00
1	02	CURRENT SECTOR
1	00	BYTE OF 00
1	FB	DATA ADDRESS MARK
128	E5	DATA
1	00	END OF SECTOR CODE
	-	
	-	
	-	
	-	
1	FE	ID ADDRESS MARK
1	00	TRACK
1	00	
1	1A	SECTOR 26
1	00	
1	FB	DATA ADDRESS MARK
128	E5	DATA
1	FF	END OF TRACK CODE

THE PROGRAM FIRST SEEKS TRACK 0. IT THEN SETS THE DMA ADDRESS TO THE BEGINNING OF THE TRACK IMAGE OUTLINED ABOVE. SETTING BIT 7 OF THE COMMAND BYTE STARTS THE INITIALIZATION.(THIS IS THE SAME BIT AS FOR A NORMAL WRITE OPERATION.) IT MONITORS IO FINISH STATUS BIT FOR OPERATION COMPLETE. THE PROGRAM THEN STEPS TO THE NEXT TRACK, MODIFIES THE 26 TRACK BYTES IN THE TRACK IMAGE IN MEMORY, AND REPEATS THE PROCESS BY SENDING OUT THE DMA ADDRESS AND THE WRITE COMMAND. THIS PROCEEDURE IS FOLLOWED UNTIL ALL 77 TRACKS ARE WRITTEN.

TO ASSURE HEAD STABILITY, THERE IS A 50 MILLISECOND DELAY AFTER STEPPING BEFORE A WRITE COMMAND IS ISSUED.