

**BFD-68
SYSTEM MANUAL**

BFD-68 REFERENCE MANUAL

INSTALLATION INSTRUCTIONS

PART 1:

OPERATING DOS68

INTRODUCTION
COMMAND DESCRIPTIONS
 RESIDENT COMMANDS
 TRANSIENT COMMANDS
WRITE PROTECT
BOOTING DOS68
FORMATTING NEW DISCS
CREATING NEW DISCS
SYSTEM MEMORY MAP
LOCATING MONITOR AT \$D000

PART 2:

DOS68 SYSTEM PROGRAMMERS GUIDE

INTRODUCTION
MONITOR SYSTEM
 MONITOR ENTRY POINTS
 USER COMMAND TABLE
CREATING TRANSIENT COMMANDS
DFM68 SYSTEM
 INTRODUCTION
 FCB FORMAT
 DIRECTORY FORMAT
 INTERFACING
 INITIALIZATION
 CLOSING
 I/O REQUESTS
 USING DFM
 READING A FILE
 WRITING A FILE

PART 3:

SYSTEM HARDWARE
 ROM ADDRESSING
 DISC INTERFACE DESCRIPTION
 DISC INTERFACE SIGNALS
 PREVENTATIVE MAINTENANCE
 INSTALLING ADDITIONAL DRIVES
 DISKETTE REQUIREMENTS

PART 4:

GENERAL INFORMATION
WARRANTEE REGISTRATION
SOFTWARE LICENSE
SYSTEM ACCESSORIES

APPENDICES:

DFM68 FUNCTION CODES
DFM68 ERROR CODES
BASIC PATCHES
CORES PATCHES

INSTALLATION INSTRUCTIONS

UNPACKING

CAREFULLY REMOVE THE BFD-68 DISC SYSTEM FROM ITS SHIPPING CONTAINER. REMOVE THE TOP FROM THE DISC SYSTEM AND CHECK THAT ALL BOARDS ARE PROPERLY SEATED IN PLACE. REMOVE THE BFD-68 INTERFACE BOARD. REMOVE THE PROTECTIVE PACKING MATERIAL WRAPPED AROUND THE BOARD. INSPECT BOTH DISC SYSTEM AND CONTROLLER BOARD FOR ANY SHIPPING DAMAGE. ANY DAMAGE SHOULD BE REPORTED TO THE SHIPPING AGENT. REINSTALL THE DISC SYSTEM TOP AND ATTACH WITH THE ENCLOSED SCREWS.

INSTALLATION

CAUTION ! ! !

DURING INSTALLATION ALL POWER SHOULD BE REMOVED FROM THE COMPUTER TO AVOID DAMAGE TO COMPUTER AND INTERFACE.

FOR PROPER OPERATION OF THE BFD-68 SYSTEM IT IS REQUIRED THAT THE SYSTEM HAVE 4K OF READ/WRITE MEMORY LOCATED AT '7000'H. THE USER MAY OPTIONALLY CHANGE THE MEMORY REQUIREMENTS TO START AT 'D000'H (SEE SOFTWARE MANUAL). AFTER MEMORY HAS BEEN PROPERLY SITUATED THE BFD-68 DISC CONTROLLER IS INSTALLED. INSTALL THE BFD-68 CONTROLLER IN ANY OF THE LARGE SLOTS SO THAT THE COMPONENT SIDE FACES FORWARD. NEXT INSTALL THE DISC INTERFACE CABLE. THE INTERFACE CABLE IS INSTALLED SUCH THAT THE NUMBER ONE PIN (INDICATED BY AN ARROW NEAR THE LOWER EDGE OF THE CONNECTOR) IS TO THE RIGHT AS VIEWED FROM THE FRONT OF THE COMPUTER. SLIDE THE CABLE CONNECTOR ONTO THE CONNECTOR ON THE TOP MIDDLE EDGE OF THE BFD-68 DISC CONTROLLER BOARD.

POWER FOR THE BFD-68 DISCS IS DERIVED FROM ANY SOURCE OF 115 VOLTS 60 CYCLE POWER. WHEN THE A.C. LINE CORD IS PLUGGED IN POWER MAY BE APPLIED BY PRESSING THE POWER SWITCH LOCATED ON THE FRONT PANEL OF THE DISC SYSTEM.

POWER MAY NOW BE REAPPLIED TO THE COMPUTER AND DISC SYSTEM.

DISKETTES SHOULD BE INSERTED INTO THE DRIVES WITH THE LABEL FACING THE LEFT SIDE OF THE SYSTEM. THE EDGE OF THE DISKETTE WITH THE LONG OVAL CUTOUT SHOULD BE THE FIRST EDGE OF THE DISKETTE TO ENTER THE DRIVE.

CAUTION !!!

NEVER TURN POWER ON OR OFF TO THE BFD-68 OR THE COMPUTER TO WHICH THE BFD-68 IS ATTACHED WHILE A DISKETTE IS INSTALLED IN ANY OF THE DISC DRIVE UNITS. DURING POWER ON OR POWER OFF, A FALSE WRITE COULD OCCUR WHICH MAY DESTROY THE DATA STORED ON THE DISKETTE.

IMPORTANT

READ LIMITED WARRANTY AND SOFTWARE LICENSE INFORMATION IN THIS MANUAL PRIOR TO USING THE SYSTEM.

IN THE COMMAND DESCRIPTIONS TO FOLLOW, THE FOLLOWING CONVENTIONS ARE USED:

THE DISC DRIVE UNIT NUMBERS ARE 0, 1, AND 2.

ANGLE BRACKETS, < >, ARE USED TO ENCLOSE A STRING OF CHARACTERS TO INDICATE THAT THE STRING INDICATES ONE ITEM. FOR EXAMPLE: <UNIT NUMBER> IS USED BELOW TO REPRESENT THE DISC UNIT NUMBER FOR A FILE.

IF A FIELD IN A COMMAND IS OPTIONAL, THE OPTIONAL PORTION IS BRACKETED IN SQUARE BRACKETS. FOR EXAMPLE: [, <UNIT NUMBER>] MEANS THAT THE COMMA FOLLOWED BY A UNIT NUMBER IS OPTIONAL.

MANY FILES REQUIRE THE SPECIFICATION OF A FILE; <FILE SPEC> WILL BE USED AS AN ABBREVIATION FOR THE FOLLOWING:

[<UNIT NUMBER>:]<FILE NAME>[.<EXTENSION>]

WHERE <FILE NAME> MAY BE SIX ALPHANUMERIC CHARACTERS AND <EXTENSION> MAY BE THREE ALPHANUMERIC.

SOME LEGAL FILE SPECIFICATIONS MIGHT BE:

1:FILE.ONE	A FILE CALLED "FILE.ONE" ON UNIT NUMBER 1
ABC.1	A FILE CALLED "ABC.1" ASSUMED TO BE ON UNIT 0
FNAME	A FILE CALLED "FNAME" ASSUMED TO BE ON UNIT 0
0:FILE.ONE	NOTE THAT 0:FILE.ONE IS DIFFERENT FILE FROM
	1:FILE.ONE

COMMAND DESCRIPTIONS

DOS68 INDICATES ITS READINESS TO ACCEPT A NEW COMMAND BY TYPING AN AMPERSAND, "&", ON THE TERMINAL. AT THIS TIME A NEW COMMAND LINE MAY BE ENTERED.

COMMANDS GIVEN TO DOS68 ARE ENTERED ON A LINE INPUT BASIS. THIS MEANS THAT THE ENTIRE LINE IS TYPED IN BEFORE DOS68 BEGINS TO PROCESS IT. USING SUCH LINE INPUT GIVES THE USER A CHANCE TO EASILY CORRECT TYPING ERRORS OR TO COMPLETELY CANCEL THE LINE BEFORE DOS68 STARTS ANY PROCESSING. THE PREVIOUSLY ENTERED CHARACTER CAN BE DELETED BY USING CONTROL O OR SHIFT O (BACK ARROW). DOS68 WILL ECHO ALL DELETED CHARACTERS ENCLOSING THEM IN BACKSLASHES, "\". CONTROL U OR CONTROL X CAN USED TO DELETE THE ENTIRE LINE; DOS68 WILL ECHO ^U OR ^X AND DO A CARRIAGE RETURN LINE FEED AND BE READY FOR A NEW COMMAND (A NEW PROMPT CHARACTER IS NOT ISSUED).

NOTE: DOS68 ASSUMES THAT THE CHARACTER INPUT ROUTINE FOR THE TERMINAL DOES NOT AUTOMATICALLY ECHO THE CHARACTER. THUS, WHILE THE USER IS TYPING A COMMAND, THE INPUT ROUTINE WITHIN DOS68 SUPPRESSES THE NORMAL AUTOMATIC ECHO USED BY THE MIKBUG (TM MOTOROLA) ROM.

OPERATING DOS68

INTRODUCTION

DOS68 IS A DISC ORIENTED MONITOR FOR USE WITH SMOKE SIGNAL BROADCASTING'S BFD-68 MINIFLOPPY (TM SHUGART) DISC INTERFACE.

MOST DOS68 COMMANDS ARE IMPLEMENTED AS TRANSIENT PROGRAMS. THIS MEANS THAT THE ROUTINE TO PROCESS THE COMMAND IS NOT RESIDENT IN MEMORY, BUT RATHER IS LOADED INTO MEMORY WHEN IT IS NEEDED TO BE EXECUTED. THIS FACILITY ALLOWS TWO THINGS: (1) THE RESIDENT MONITOR IS SMALLER BECAUSE ALL THE ROUTINES ARE NOT PRESENT IN MEMORY. (2) THE USER IS EASILY ABL TO IMPLEMENT NEW MONITOR COMMANDS WITHOUT MODIFICATION TO THE MONITOR.

IN ADDITION, IN THE CALLING OF TRANSIENT MONITOR COMMANDS, THE USER OF MULTI-DISC SYSTEMS HAS THE ABILITY TO SPECIFY THAT THE TRANSIENT COMMAND IS TO BE FOUND ON A SPECIFIC DISC DRIVE. THIS FEATURE IS PARTICULARLY USEFUL WHEN DEBUGGING NEW MONITORS WITH TRANSIENT COMMANDS WITH NAMES WHICH CONFLICT WITH THE OLD SYSTEM'S NAMES. THE DISC NUMBER IS SPECIFIED BY PRECEEDING THE COMMAND NAME WITH THE DISC NUMBER FOLLOWED BY A COLON, ':'. FOR EXAMPLE, 2:LIST,1 WILL RETRIEVE THE LIST COMMAND PROCESSOR FROM DISC DRIVE 2 TO LIST THE DIRECTORY FOR DRIVE 1.

DOS68 IS SUPPLIED WITH THE FOLLOWING COMMANDS:

COMMAND NAME	FUNCTION

LIST	LIST THE DISC FILE DIRECTORY
SAVE	SAVE MEMORY INTO A FILE
* GET	LOAD A FILE INTO MEMORY WITHOUT BEGINNING EXECUTION
GETH	LOAD A HEX FORMATTED OBJECT FILE
* RUN	LOAD A FILE INTO MEMORY AND BEGIN EXECUTION
DELETE	REMOVE A FILE FROM A DISC
RENAME	CHANGE THE NAME OF A FILE
APPEND	MERGES TWO FILES TOGETHER TO FORM ONE FILE
PRINT	PRINT THE CONTENTS OF A FILE
COPY	ALLOWS FILES TO BE COPIED FROM°DISC TO DISC
SDC	SINGLE DISC DRIVE COPY
LINK	SET UP INFORMATION TO BOOT THE MONITOR
* CLOSE	ERROR RECOVERY TO BE ABLE TO CLOSE ALL OPEN FILES
* EXIT	EXIT TO OTHER RESIDENT MONITOR

(* INDICATES A MEMORY RESIDENT COMMAND)

RENAME COMMAND

&RENAME, <THIS>, <THAT>

THE RENAME COMMAND CHANGES THE NAME OF "THIS" FILE TO "THAT".

FOR EXAMPLE:

&RENAME, FILE.ONE, FILE.1

ON DISC 0 THE FILE "FILE.ONE" IS
RENAMED TO "FILE.1"

&RENAME, 2:THIS, THAT

ON DISC 2, THE FILE "THIS" IS RENAMED
TO "THAT"

APPEND COMMAND

&APPEND, <THIS FILE>, <THAT FILE>

THE APPEND COMMAND ALLOWS TWO FILES TO BE MERGED INTO ONE FILE. THE FILE SPECIFIED BY <THIS FILE> IS APPENDED TO <THAT FILE> WHERE BOTH FILES ARE ASSUMED TO RESIDE ON THE DISC SPECIFIED BY THE FILE SPECIFICATION OF <THAT FILE>. ONCE APPENDED, THE FILE NAME OF <THIS FILE> IS REMOVED FROM THE DISC DIRECTORY.

FOR EXAMPLE:

&APPEND, FILE.ONE, 2:FILE.TWO

THIS CAUSES THE FILE "FILE.ONE" ON DISC TWO TO BE APPENDED TO FILE "FILE.TWO" ON DISC TWO.

PRINT COMMAND

&PRINT, <FILE SPEC>[, <NUMBER OF LINES>]

THE PRINT COMMAND PRINTS THE CONTENTS OF THE SPECIFIED FILE ON THE TERMINAL. IF THE OPTIONAL NUMBER OF LINES IS NOT SPECIFIED, THE ENTIRE FILE IS LISTED WITHOUT PAUSING. IF A NUMBER OF LINES IS SUPPLIED, THE FILE IS LISTED UNTIL THAT NUMBER OF LINE FEED CHARACTERS OCCURS AT WHICH TIME THE LISTING PAUSES UNTIL A KEY ON THE TERMINAL IS STRUCK. IF THE KEY IS A CARRIAGE RETURN CHARACTER THE LISTING IS TERMINATED, OTHERWISE, THE LISTING CONTINUES UNTIL ANOTHER SET OF LINES HAS BEEN PRINTED. THE PRINT COMMAND WILL LIST ANY NON-COMMAND FILE, INCLUDING BINARY FILES WHICH WILL APPEAR AS A STRING OF ODD CHARACTERS AND CONTROL CHARACTERS.

COPY COMMAND

©, <FROM>, <TO>

THE COPY COMMAND PROVIDES A MEANS FOR COPYING FILES. THE COPY COMMAND HAS THREE FORMS:

(1) ©, <FROM FILE SPEC>, <TO UNIT SPEC>

THIS FORM COPIES A FILE FROM ONE DISC TO ANOTHER DISC PUTTING THE COPIED FILE INTO A FILE BY THE SAME NAME AS THE ORIGINAL.

EXAMPLE:

©, 0:MON, 1:

COPY THE FILE "MON" FROM DISC 0 TO DISC 1

(2) ©,<FROM FILE SPEC>,<TO FILE SPEC>

THIS FORM COPIES THE FROM FILE TO THE NEW FILE NAME SPECIFIED. NOTE: THE 'TO' AND 'FROM' UNIT NUMBERS MAY BE THE SAME; HOWEVER, WHEN THEY ARE, THE FILE NAMES MUST DIFFER.

EXAMPLES:

 ©,0:MON,2:OLDMON
 ©,1:FILE.NEW,1:FILE.BAK

(3) ©,<FROM UNIT SPEC>,<TO UNIT SPEC>

THIS FORM OF THE COPY COMMAND COPIES ALL FILES, INCLUDING TRANSIENT COMMAND FILES, TO THE UNIT SPECIFIED.

EXAMPLE:

 ©,0:,1: COPY ALL FILES FROM UNIT 0 TO UNIT 1

THE COPY COMMAND ADDS FILES TO THE DISC BEING COPIED TO; IT DOES NOT DELETE FILES FROM EITHER DISC. IF THERE IS NOT ENOUGH ROOM ON THE DISC TO WHICH THE COPY IS BEING MADE AN ERROR MESSAGE WILL BE PRINTED. IF THE COPY OPERATION IS SUCCESSFULLY COMPLETED, THE MESSAGE "COPY COMPLETE" WILL BE PRINTED.

SDC COMMAND

 &SDC,<FILE SPEC>[,,\$]

THE SDC (SINGLE DRIVE COPY) COMMAND ALLOWS USERS OF SINGLE DISC DRIVE SYSTEMS TO COPY FILES FROM ONE DISC TO ANOTHER DISC BY READING THE ENTIRE FILE INTO MEMORY, PROMPTING THE OPERATOR TO CHANGE THE DISC, AND THEN WRITING THE MEMORY IMAGE TO THE NEW DISC.

THE SDC COMMAND ASSUMES THAT THE USER HAS 16K OF RAM ORIGINATED AT 0, AND THAT THE DISC DRIVE IN THE SYSTEM IS UNIT ZERO.

LINK COMMAND

 &LINK,<FILE SPEC>

THE LINK COMMAND TELLS THE BOOTSTRAPPING ROUTINE WHICH PROGRAM TO LOAD AND RUN WHEN THE ROM BOOTSTRAP ROUTINE IS EXECUTED.

FOR EXAMPLE: &LINK,MON CAUSES THE LINKAGES ON THE DISC IN DRIVE ZERO TO BE SET UP SUCH THAT THE FILE "MON" WILL BE RUN WHEN THIS DISC IS BOOTED IN. &LINK,2:NEWMON SIMILIARLY LINKS THE BOOT ON DISC TWO TO "NEWMON". IN THIS CASE, IN ORDER TO THEN BOOT "NEWMON", YOU MUST MOVE THE DISC WITH "NEWMON" ON IT DOWN TO DRIVE ZERO BEFORE THE ROM BOOT IS EXECUTED.

BOOTING DOS68

THE INITIAL LOADING OF SOFTWARE INTO A COMPUTER WITH LITTLE OR NO PERMANENTLY RESIDENT SOFTWARE INVOLVES A PROCESS CALLED BOOTSTRAPPING. BOOTSTRAPPING IS THE USE OF A TYPICALLY SMALL, DUMB PROGRAM TO LOAD A LARGER, SMARTER LOADER WHICH IN TURN CAN LOAD THE DESIRED PROGRAM (USUALLY THE MONITOR).

THE ROM SUPPLIED ON THE BFD-68 INTERFACE BOARD CONTAINS A BOOTING ROUTINE CAPABLE OF READING IN SECTOR ZERO OF TRACK ZERO ON DISC DRIVE ZERO AND TRANSFERRING TO THE ROUTINE READ IN. THE CONTENT OF THIS SECTOR IS INITIALIZED BY THE DISC FORMATTING PROGRAM TO CONTAIN A PROGRAM CAPABLE OF LOADING THE MONITOR.

THUS TO LOAD DOS68, THE USER MUST USE MIKBUG (TM MOTOROLA) OR SOME OTHER METHOD OF TRANSFERRING TO LOCATION \$8020. THIS WILL CAUSE DOS68 TO BE BROUGHT IN OFF THE DISC AND EXECUTED. BOOTING IN THIS MANNER IS REFERRED TO AS "COLD STARTING" BECAUSE ALL MONITOR AND DISC FILE MANAGER TEMPORARIES ARE INITIALIZED. IT IS POSSIBLE TO "WARM START" THE MONITOR, THAT IS, TO RELOAD THE MONITOR WITHOUT INITIALIZING EVERYTHING BY INSTEAD TRANSFERRING TO THE ROM AT LOCATION \$8023.

NOTE 1:

FOR NEW DISCS TO KNOW WHAT FILE TO BOOT IN WHEN USING THE ROM RESIDENT BOOTING ROUTINE, THE MONITOR MUST HAVE BEEN "LINKED" (SEE THE "LINK" COMMAND). "LINKING" THE DISC IS A PROCESS WHEREBY THE BOOTING ROUTINE KNOWS WHAT FILE TO LOAD AND RUN WHEN THE DISC IS BOOTED.

NOTE 2:

THE ROM SUPPLIED WITH THE BFD-68 CONTROLLER BOARD UTILIZES MEMORY LOCATIONS \$7000 THROUGH \$707F AND \$7380 THROUGH \$73FF WHEN BOOT LOADING.

WRITE PROTECT

THE BFD-68 IS PROVIDED WITH A WRITE PROTECT FEATURE. EACH DISKETTE HAS A SMALL RECTANGULAR CUTOUT ON ONE EDGE. COVERING THE CUTOUT WITH A PIECE OF TAPE WILL PROTECT THE DISKETTE FROM AN ACCIDENTAL WRITE. AN ATTEMPT TO EXECUTE ANY COMMAND THAT WOULD WRITE TO THE DISKETTE WILL RETURN AN ERROR MESSAGE.

DISC FORMATTING

THE FILE "FORMAT.BIN" CONTAINS A STANDALONE PROGRAM FOR THE FORMATTING AND VERIFICATION OF UNINITIALIZED DISCS. THE FORMATTER ALSO SERVES TO INITIALIZE THE DISC STRUCTURE REQUIRED BY THE DISC FILE MANAGEMENT (DFM68) SOFTWARE.

TO LOAD THE FORMATTER THE MONITOR 'GET' COMMAND IS USED:
&GET,FORMAT.BIN

BEFORE THE FORMATTER EXECUTION IS BEGUN, THE DISC TO BE INITIALIZED IS PLACED IN DRIVE ZERO. WITH THE DISC IN PLACE, BEGIN EXECUTION OF THE FORMATTER AT LOCATION \$0100. THE FORMATTER REQUIRES APPROXIMATELY \$1300 BYTES OF MEMORY AND THREE AND ONE HALF MINUTES TO EXECUTE TO FORMAT AND VERIFY AN ENTIRE DISC. UPON COMPLETION, THE FORMATTER TYPES THE MESSAGE:

FORMATTING COMPLETE. XXXX YYYY ZZZZ

WHERE:

XXXX	IS THE FIRST SECTOR AVAILABLE FOR USE,
YYYY	IS THE LAST SECTOR AVAILABLE FOR USE, AND
ZZZZ	IS THE COUNT, IN HEX, OF AVAILABLE SECTORS

NORMAL EXECUTION WITHOUT ERRORS SHOULD RESULT IN:

FORMATTING COMPLETE. 8042 A251 0274

ONLY A FEW SECONDS ARE REQUIRED TO ACTUALLY FORMAT THE DISC. THE REMAINDER OF THE TIME IS SPENT CHECKING EACH SECTOR OF THE DISC TO VERIFY THAT EACH HAS NO DROPOUTS.

SECTORS WHICH FAIL THE VERIFICATION PROCEDURE ARE ELIMINATED FROM THE DISC FILE STRUCTURE AT THIS TIME TO PREVENT THEIR SUBSEQUENT USAGE.

THIS VERIFICATION PROCEDURE PROVIDES TWO VERY IMPORTANT ADVANTAGES:

- 1) IT ALLOWS THE USE OF DISCS WITH A SMALL NUMBER OF BAD SECTORS.
- 2) IT GIVES AN INDICATION OF DISC WEAR. THE NUMBER OF BAD SECTORS WILL START TO INCREASE AS THE DISC NEARS THE END OF ITS USEFUL LIFE.

ERROR MESSAGES GENERATED BY THE FORMATTER ARE OF TWO FORMS:

(1) ERROR: XX YY

OR (2) BS=TTSS

FORM (1): XX REPRESENTS A TWO DIGIT ERROR CODE, AND YY IS THE TWO DIGIT REPRESENTATION OF THE ERROR STATUS RETURNED FROM THE WD 1771-1 CHIP.

THE POSSIBLE VALUES OF XX ARE:

XX	MEANING
10	NOT ABLE TO STEP DISC IN
11	NOT ABLE TO RESTORE DISC
12	ERROR IN WRITING THE FORMAT TO THE TRACK
13	UNABLE TO RESTORE AFTER FORMATTING THE DISC
14	ERROR IN STEPPING AFTER WRITING A TRACK
F0	FATAL ERROR IN CONSTRUCTING DISC STRUCTURE
F1	FATAL ERROR IN INITIALIZING THE MASTER FILE DIRECTORY
F2	FATAL ERROR IN FINISHING DISC STRUCTURE CONSTRUCTION
F3	FATAL ERROR IN INITIALIZING BOOTING ROUTINE
FF	NO USABLE SPACE ON DISC FOR STORING FILES

ERROR CODES ABOVE WHICH START WITH THE DIGIT 'F' ARE FATAL ERRORS WHICH CAUSE THE FORMATTER TO ABORT ITS FUNCTION. ALL OTHER ERRORS ARE RETRIED UNTIL THEY SUCCEED.

FORM (2): TT REPRESENTS THE TRACK NUMBER (\$80 THRU \$A2), AND SS REPRESENTS THE SECTOR NUMBER (\$40 THRU \$51) OF A SECTOR WHICH HAS FAILED THE VERIFICATION PROCEDURE AND IS TO BE ELIMINATED FROM THE DISC STRUCTURE TO PREVENT ITS USAGE.

CREATING NEW DISCS

THIS SECTION DESCRIBES HOW TO BUILD BACKUP DISCS.

IT SHOULD BE NOTED THAT IT IS NOT NECESSARY TO HAVE A COPY OF THE DOS68 OPERATING SYSTEM ON EVERY DISC USED BY DOS68. IT IS ONLY NECESSARY TO HAVE THE SYSTEM PRESENT ON THE DISC FROM WHICH THE SYSTEM IS TO BE BOOT LOADED.

STEP 1: FORMAT A DISC

THE FIRST STEP IN BUILDING A NEW DISC IS TO FORMAT A NEW DISC. DOS68 IS SUPPLIED WITH A PROGRAM CALLED "FORMAT.BIN". FORMAT.BIN CONTAINS A PROGRAM WHICH INITIALIZES BLANK SOFT SECTORED DISCETTES (SEE THE SECTION ON FORMATTING NEW DISCS).

STEP 2: COPY ALL DESIRED FILES TO THE NEW DISC.

SINGLE DRIVE SYSTEMS

TO COPY FILES BETWEEN DISKETTES ON A SINGLE DRIVE DISC SYSTEM, A TRANSIENT MONITOR COMMAND CALLED "SDC" (SINGLE DISC COPY) HAS BEEN PROVIDED. SDC ALLOWS FOR THE READING OF A FILE INTO MEMORY, GIVING THE OPERATOR TIME TO CHANGE DISKETTES, AND WRITING THE FILE BACK OUT TO THE NEW DISC (SEE THE TRANSIENT COMMAND DESCRIPTION OF "SDC").

REPEATEDLY USE SDC TO COPY ALL DESIRED FILES FROM THE OLD DISC TO THE NEW DISC.

DUAL OR TRIPLE DRIVE SYSTEMS

ON DUAL OR TRIPLE DRIVE SYSTEMS THE TRANSIENT MONITOR COMMAND "COPY" IS USED TO COPY FILES FROM DISC TO DISC.

ASSUMING THE NEWLY FORMATTED DISC IS IN DRIVE ZERO AND THE OLD SYSTEM DISC IS IN DRIVE ONE, THE FOLLOWING COMMAND WILL COPY ALL THE FILES FROM DISC ONE TO DISC ZERO:

```
&COPY 1: 0:
```

NOTE: PREPARING SYSTEM DISCS

IF THE DISC BEING PREPARED IS TO BE USED TO BOOT FROM A COLD OR WARM START, THE DISC MUST CONTAIN A COPY OF DOS68 AND THE DISC MUST BE "LINKED" BY EXECUTING THE FOLLOWING MONITOR COMMAND:

```
&LINK,0:DOS68.V25 (ASSUMING THE NEW DISC IS STILL IN DRIVE ZERO)
```

(THE LINK COMMAND SERVES TO TELL THE BOOTING PROGRAM WHICH FILE IS TO BE LOADED AND EXECUTED WHEN THE ROM BOOTING ROUTINE IS USED.)

DOS68 V2.7 MEMORY MAP

\$7000 - \$707F	USED BY ROM TO LOAD SYSTEM BOOT FROM DISC LATER REUSED BY MONITOR FOR STACK AREA
\$7080 - \$727F	TRANSIENT PROGRAM AREA (TPA)
\$7280 - \$777F	MONITOR PROGRAM AREA
\$7780 - \$7FFF	DFM PROGRAM AREA
\$7F80 - \$7FFF	I/O BUFFER FOR ROM BOOT ROUTINE

\$0000 - \$3FFF	IS ASSUMED TO BE USER AREA
\$0100 - \$02FF	IS USED BY "COPY" COMMAND FOR I/O BUFFERS
\$0100 - \$02FF	IS USED BY "SAVE.BLD" PROGRAM

ROM ENTRY POINTS

\$8020	- MONITOR COLD START BOOT ENTRY
\$8023	- MONITOR WARM START BOOT ENTRY
\$A077 - \$A07F	ROM TEMPORARY TABLE

MONITOR ENTRY POINTS

\$7280	- COLD START ENTRY
\$7283	- WARM START ENTRY
\$7286	- JUMP TO KEYBOARD OUTPUT (OUTEEE)
\$7289	- JUMP TO KEYBOARD INPUT (INEEE)
\$728C	- JUMP TO USER'S MONITOR
\$728F	- RESIDENT COMMAND TABLE EXTENSION

LOCATING THE MONITOR AT \$D000

A DISC WITH DOS68 LOCATED AT \$D000 THROUGH \$DFFF IS AVAILABLE FROM SMOKE SIGNAL BROADCASTING. THIS WILL ALLOW THE USER TO HAVE THE ENTIRE LOWER 32K AVAILABLE FOR USER PROGRAMS, EXCEPT DURING BOOT OPERATIONS. THE BOOT ROUTINE WILL USE \$7000 THROUGH \$707F AND \$7F80 THROUGH \$7FFF WHILE IT LOADS THE MONITOR INTO THE \$D000 THROUGH \$DFFF REGION. AFTER LOADING THE MONITOR, THE ENTIRE \$7000 BLOCK IS AVAILABLE TO THE USER.

NO CHANGE TO THE CPU CARD IS NEEDED TO LOCATE MEMORY AT \$D000 THROUGH \$DFFF, HOWEVER, THE SWTPC 4K MEMORY CARDS WILL NOT LOCATE IN THAT AREA WITHOUT MODIFICATION. THE USER, DESIRING TO USE DOS68 AT \$D000, IS RESPONSIBLE FOR MODIFYING HIS MEMORY CARD TO PROPERLY OPERATE AT THAT LOCATION.

THE SMOKE SIGNAL BROADCASTING M-16A 16K MEMORY BOARD MAY BE SWITCHED SELECTED TO OCCUPY \$A000 THROUGH \$DFFF AND, THUS, CAN PROVIDE THE MEMORY REQUIRED TO OPERATE WITH DOS68 LOCATED AT \$D000 THROUGH \$DFFF. WHEN THE M-16A IS USED AT THAT LOCATION, A SIMPLE MODIFICATION TO THE SWTPC CPU CARD IS REQUIRED. MODIFICATION INSTRUCTIONS ARE GIVEN IN THE M-16A MANUAL.

WHEN USING DOS68 AT \$D000 THROUGH \$DFFF, ALL PROGRAMS WHICH ACCESS THE MONITOR OR DFM IN THE \$7000 AREA SHOULD BE CHANGED TO ACCESS CORRESPONDING LOCATIONS IN THE \$D000 AREA.

DOS68 SYSTEM PROGRAMMER'S GUIDE

INTRODUCTION

THE BFD-68 DISC OPERATING SYSTEM, DOS68, CONSISTS OF TWO DISTINCT PROGRAMS:

- (1) THE MONITOR PORTION, AND
- (2) THE DISC FILE MANAGEMENT PORTION (DFM68)

DFM68 IS STRICTLY A DISC FILE HANDLING SYSTEM AND CAN BE USED INDEPENDENTLY OF THE MONITOR PORTION OF DOS68 WHEN THE USER WISHES TO MANIPULATE DISC FILES.

THE MONITOR PORTION OF DOS68 HANDLES ALL OTHER FUNCTIONS OF DOS68 NOT HANDLED BY DFM68.

THE FOLLOWING SECTIONS PROVIDE THE NECESSARY INTERFACING INFORMATION FOR THE USER TO BE ABLE TO MAKE USE OF THE FUNCTIONS AVAILABLE THROUGH DOS68.

MONITOR SYSTEM

THIS SECTION DESCRIBES THE USER INTERFACE TO THE DOS68 MONITOR.

MONITOR JUMP TABLE

THE FIRST PORTION OF THE MONITOR CONTAINS A JUMP TABLE FOR ACCESSING SEVERAL COMMONLY USED ROUTINES WHICH ARE PRESENT WITHIN THE MONITOR. THE LAYOUT FOR THE TABLE IS AS FOLLOWS:

ENTRY ADDR	ENTRY NAME	FUNCTION
\$7280	ZCOLDS	MONITOR COLD START
\$7283	ZWARMS	MONITOR WARM START
\$7286	OUTEEE	CHARACTER OUTPUT ROUTINE
\$7289	INEEE	CHARACTER INPUT ROUTINE
\$728C	ZMON	JMP TO ROM MONITOR
\$7291	ZFLSPC	GET A FILE SPECIFICATION
\$7294	ZGCHAR	GET CURRENT CHARACTER FROM THE LINE BUFFER
\$7297	ZGNCHR	GET THE NEXT CHARACTER FROM THE LINE BUFFER
\$729A	ZANCHK	CHECK FOR ALPHANUMERIC
\$729D	ZDIE	PRINT COMMAND STRING, ERROR MESSAGE, AND EXIT
\$72A0	ZGETHN	GET A HEX VALUE FROM THE LINE BUFFER
\$72A3	ZADDX	ADD THE B REGISTER TO THE INDEX
\$72A6	ZOUTST	PRINT A STRING
\$72A9	ZTYPDE	TYPE THE DISC ERROR MESSAGE
\$72AC	ZOUTHX	PRINT A BYTE IN HEX
\$72AF	ZOUTHX	PRINT AN ADDRESS IN HEX
\$72B2	(UNUSED)	
\$72B5	ZLINEI	INPUT EDITED LINE FROM THE TERMINAL

ZCOLDS - MONITOR COLD START

THIS ENTRY TO DOS68 RESETS THE PROCESSOR'S STACK AND ALL INTERNAL STATUS OF BOTH THE MONITOR AND DFM. THE BANNER:

DOS68 VX.YR

IS PRINTED ON THE TERMINAL WHERE VX.YR REPRESENTS THE VERSION NUMBER. DOS68 THEN PROCEEDS TO DO A WARM START.

ZWARMS - MONITOR WARM START

THIS ENTRY TO DOS68 RESETS THE PROCESSOR'S STACK, SETS \$A048 TO THE ADDRESS OF 'ZWARMS' FOR SUBSEQUENT RESTARTING, CLOSES ANY FILES THAT MAY HAVE BEEN LEFT OPEN, AND THEN PROMPTS THE OPERATOR FOR A NEW COMMAND BY TYPING AN AMPERSAND, '&'.

OUTEEE - CHARACTER OUTPUT TO THE CONTROL TERMINAL

THIS JMP IS USED FOR ALL OUTPUT TO THE USER'S TERMINAL. THIS JMP IS SET TO USE "OUTEEE" AT \$E1D1 WITHIN THE MIKBUG (TM MOTOROLA) ROM. IT IS ASSUMED THAT THE OUTPUT ROUTINE PRESERVES THE B REGISTER AND THE X REGISTER BUT NOT NECESSARILY THE A REGISTER (THE DATA TO BE OUTPUT).

INEEE - CHARACTER INPUT FROM THE CONTROL TERMINAL

THIS JMP IS USED FOR ALL INPUT FROM THE USER'S CONTROL TERMINAL. THIS JMP IS SET TO USE "INEEE", \$E1AC, WITHIN THE MIKBUG (TM MOTOROLA) ROM. IT IS ASSUMED THAT THE B AND X REGISTERS ARE PRESERVED AND THAT THE CHARACTER INPUT IS RETURNED IN THE A REGISTER.

DOS68 ASSUMES THAT INEEE DOES NOT AUTOMATICALLY ECHO INPUT BACK TO THE TERMINAL. SEE DESCRIPTION OF 'ZLINEI'.

ZMON - JUMP TO ROM MONITOR

THE MONITOR "EXIT" COMMAND USES THIS JUMP TO GIVE CONTROL TO SOME OTHER RESIDENT MONITOR. THIS JUMP IS SET TO \$E0D6 TO CAUSE ENTRY INTO MIKBUG (TM MOTOROLA).

ZFLSPC - GET A FILE SPECIFICATION

"FILE SPEC" IS A ROUTINE WHICH IS USED TO PICK UP A UNIT NUMBER AND FILE NAME FROM THE INPUT BUFFER IN THE FORM:

[<UNIT NUMBER>:]<FILE NAME>[.<EXTENSION>]

THE LINE BUFFER POINTER IS ASSUMED TO BE POINTING TO THE DELIMITER OF THE PREVIOUS FIELD. TO USE ZFLSPC, THE X REGISTER MUST CONTAIN THE ADDRESS OF A FILE CONTROL BLOCK (FCB) IN WHICH THE UNIT AND FILE NAME IS TO BE PUT (SEE FCB FORMAT DESCRIPTION).

ZFLSPC RETURNS WITH THE CARRY SET IF AN ERROR OCCURS. IF NO ERROR OCCURS, THE FCB WILL HAVE THE PROPERLY SET UP UNIT AND FILE NAME AND THE A REGISTER WILL RETURN WITH THE DELIMITING CHARACTER OF THE FILE NAME.

NO REGISTERS ARE PRESERVED. THE LINE BUFFER POINTER IS LEFT POINTING TO THE DELIMITING CHARACTER.

ZGCHAR - GET CURRENT CHARACTER

THIS ROUTINE RETURNS THE CHARACTER CURRENTLY BEING POINTED TO BY THE LINE INPUT BUFFER POINTER.

WHEN CONTROL IS GIVEN TO A TRANSIENT PROGRAM, THE BUFFER POINTER IS POINTING TO THE DELIMITER OF THE COMMAND NAME.

NOTE: WHEN USING THE LINE INPUT ROUTINE, THIS ROUTINE MUST NOT BE CALLED UNTIL EITHER ZGNCHR, ZFLPSC, OR ZGETHN HAS BEEN USED BECAUSE THE LINE BUFFER POINTER IS INITIALIZED TO POINT TO THE CHARACTER PRECEDING THE LINE BUFFER.

ZGNCHR - GET THE NEXT CHARACTER

THIS ROUTINE ADVANCES THE LINE BUFFER POINTER BY ONE AND RETURNS TO THE CHARACTER BEING POINTED TO. ONCE A CARRIAGE RETURN CHARACTER IS RETURNED, THE POINTER WILL NO LONGER BE ADVANCED AND CARRIAGE RETURNS WILL BE RETURNED WITH EACH CALL.

ZANCHK - ALPHANUMERIC CHECK

THE CHARACTER IN THE A REGISTER IS CHECKED FOR BEING 0-9 OR A-Z. IF THE CHARACTER IS NOT ONE OF THESE CHARACTERS, THE CARRY BIT WILL BE SET ON RETURN.

ZDIE - ABORT COMMAND AND GIVE ERROR

THIS ROUTINE PRINTS THE CONTENTS OF THE LINE BUFFER TO THE LEFT OF THE LINE BUFFER POINTER, FOLLOWED BY '? ', FOLLOWED BY THE TEXT OF AN ERROR MESSAGE POINTED BY THE X REGISTER (PRINTING STOPS WHEN A NULL IS FOUND IN THE ERROR MESSAGE). AFTER PRINTING THE ERROR MESSAGE, CONTROL WILL RETURN TO THE MONITOR THROUGH THE WARM START ENTRY.

ZGETHN - GET A HEX NUMBER

THIS ROUTINE RETURNS THE VALUE OF A HEX NUMBER FOUND IN THE LINE BUFFER. ZGETHN STARTS BY DOING A ZGNCHR AND CONTINUES TO COLLECT HEX DIGITS UNTIL A NON-ALPHANUMERIC IS FOUND. UPON RETURN, THE LINE BUFFER POINTER IS LEFT POINTING TO THE DELIMITING CHARACTER, THE VALUE IS RETURNED AS A 16 BIT VALUE IN THE X REGISTER, AND IF THE CARRY IS NOT SET, INDICATING NO ERROR OCCURED, THE A REGISTER WILL CONTAIN THE TERMINATING CHARACTER, AND THE B REGISTERS WILL BE NON-ZERO IF ANY HEX DIGITS WERE FOUND.

ADDNDX - ADD THE B REGISTER TO THE INDEX REGISTER

THE VALUE IN THE B REGISTER IS ADDED INTO THE VALUE IN THE X REGISTER AND THE RESULT IS RETURNED IN THE X REGISTER. THE A REGISTER IS NOT AFFECTED, AND THE B REGISTER WILL CONTAIN THE LOW ORDER SUM.

ZOUTST - OUTPUT A STRING

THE CHARACTER STRING POINTED TO BY THE X REGISTER IS OUTPUT TO THE TERMINAL. THE OUTPUT STOPS WHEN A NULL, \$00, IS ENCOUNTERED. THE X REGISTER IS LEFT POINTING TO THE NULL UPON RETURN.

ZTYPDE - TYPE DISC ERROR

THIS ROUTINE IS USED TO PRINT THE MESSAGE "DISC ERROR: XX". THE X REGISTER IS ASSUMED TO BE POINTING TO AN FCB WHOSE ERROR STATUS WILL BE PRINTED AS THE "XX".

ZOUTHX - OUTPUT A BYTE IN HEX

THIS ROUTINE PRINTS TWO HEX DIGITS CORRESPONDING TO THE BYTE OF MEMORY POINTED TO BY THE X REGISTER. THE A REGISTER IS DESTROYED; THE B AND X REGISTERS ARE UNCHANGED.

ZOUTH - TYPE TWO BYTES IN HEX

THIS ROUTINE PRINTS FOUR HEX DIGITS CORRESPONDING TO THE TWO BYTES OF MEMORY POINTED TO BY THE X REGISTER. THE A REGISTER IS DESTROYED, THE B REGISTER IS PRESERVED, AND THE X REGISTER IS RETURNED ADVANCED BY ONE TO POINT TO THE LOW ORDER VALUE PRINTED.

ZLINEI - LINE INPUT ROUTINE

THIS ROUTINE ACCEPTS A LINE OF DATA FROM THE TERMINAL. THE FOLLOWING CHARACTERS HAVE SPECIAL MEANING TO THE INPUT ROUTINE:

CARRIAGE RETURN: TERMINATES THE INPUT
CONTROL U AND CONTROL X: RESTART LINE INPUT
CONTROL O AND SHIFT O: DELETE THE PREVIOUS INPUT CHARACTER

THE EDITED INFORMATION IS PUT INTO THE LINE INPUT BUFFER AND THE BUFFER POINTER IS RESET TO POINT TO THE CHARACTER POSITION PRECEDING THE LINE BUFFER. A CARRIAGE RETURN LINE FEED PAIR IS ECHOED UPON RECEIPT THE CARRIAGE RETURN ENDING THE INPUT.

NOTE: THE AUTOMATIC ECHO FEATURE OF THE STANDARD TERMINAL I/O PIA IS INHIBITED DURING THE LINE INPUT ROUTINE AND RE-ENABLED UPON EXIT. THIS IS DONE UPON ENTRY TO LINEIN BY

LDA A #\$3C
STA A \$8007

AUTO-ECHO IS RE-ENABLED AT THE LINEIN EXIT
APPROXIMATELY 35 BYTES LATER

LDA A #\$34
STA A \$8007

USER COMMAND TABLE

WHEN A COMMAND LINE IS PROCESSED BY DOS68, THE MONITOR RESIDENT COMMAND TABLE IS CHECKED FIRST. IF THE COMMAND IS NOT FOUND, THEN THE USER COMMAND TABLE IS CHECKED. IF THE COMMAND IS STILL NOT FOUND, THE DISC DIRECTORY IS CHECKED FOR THE COMMAND. IF STILL NOT FOUND, "RUN DENIED" IS OUTPUT TO THE TERMINAL.

AT MEMORY LOCATIONS \$728F AND \$7290 ARE TWO LOCATIONS WHICH ARE NORMALLY ZERO INDICATING THAT THERE ARE NO USER RESIDENT COMMANDS PRESENT IN MEMORY. IF THESE LOCATIONS ARE NOT ZERO THEY ARE ASSUMED TO BE THE ADDRESS OF THE USER COMMAND TABLE.

THE FORMAT OF THE USER COMMAND TABLE IS AS FOLLOWS:

START	FCB	5	LENGTH OF COMMAND (MUST BE 1 - 6)
	FCB	2	MINIMUM NUMBER OF CHARACTERS WHICH MUST BE ENTERED BY THE OPERATOR FOR A MATCH
	FDB	CMDADR	ADDRESS OF USER COMMAND
	FCC	/MYCMD/	TEXT OF COMMAND NAME
	FCB	0	0 MEANS END OF TABLE (TABLE MAY CONTAIN ANY NUMBER OF ENTRIES)
	ORG	\$728F	TELL DOS68 ABOUT THIS TABLE
	FDB	START	

IN THE ABOVE EXAMPLE, DOS68 WILL TRANSFER TO LOCATION 'CMDADR' IF THE OPERATOR TYPES IN ANY ONE OF THE FOLLOWING:

MY
MYC
MYCM
MYCMD

WHEN CONTROL IS PASSED TO THE USER COMMAND, THE INPUT LINE BUFFER POINTER IS LEFT POINTING TO THE CHARACTER DELIMITING THE COMMAND NAME SO THAT THE USER MAY REQUEST THE MONITOR TO PICK UP PARAMETERS FROM THE COMMAND LINE (SEE THE MONITOR JUMP TABLE DESCRIPTIONS).

THE USER SHOULD EXIT HIS COMMAND PROCESSOR BY DOING A JUMP TO THE MONITOR WARM START ENTRY POINT.

CREATING TRANSIENT MONITOR COMMANDS

THE FILE "SAVE.BLD" CONTAINS A PROGRAM TO FACILITATE THE CREATING OF NEW TRANSIENT MONITOR COMMANDS.

SINCE THE MONITOR 'SAVE' COMMAND IS A TRANSIENT PROGRAM, IT CANNOT BE USED TO SAVE NEW TRANSIENT ROUTINES (IF THE NEW ROUTINES RESIDES IN THE TRANSIENT PROGRAM AREA) SINCE THE SAVE COMMAND ITSELF WILL BE CALLED INTO THE TRANSIENT PROGRAM AREA (TPA) DESTROYING THE PROGRAM TO BE SAVED. HENCE, 'SAVE.BLD' IS A RESIDENT VERSION OF THE TRANSIENT SAVE COMMAND. SAVE.BLD LOADS AT MEMORY LOCATION \$0400 BY THE GET COMMAND AND IS APPROXIMATELY 512 BYTES IN LENGTH. WHEN LOADED, THE MONITOR WILL RECOGNIZE THE COMAND "SAVE" AS A RESIDENT COMMAND WHICH FUNCTIONS IDENTICALLY AS THE TRANSIENT SAVE COMMAND. THE NEW TRANSIENT PROGRAM CAN BE LOADED INTO THE TPA AND SAVED USING THE RESIDENT SAVE COMMAND. TO RELEASE THE RESIDENT SAVE COMMAND, IT IS NECESSARY TO ZERO THE TWO BYTE LONG RESIDENT COMMAND TABLE EXTENSION POINTER LOCATED AT LOCATIONS \$728F AND \$7290 (OR TO RETURN IT TO THE ADRESS OF THE BEGINNING OF THE USER COMMAND TABLE DESCRIBED IN THE PREVIOUS SECTION).

DFM68 SYSTEM

THIS SECTION IS DIRECTED TOWARD HOW TO USE DFM68 AND HOW TO INTERPRET THE DISC STRUCTURE USED.

INTRODUCTION

DFM68 IS A DISC FILE MANAGEMENT PROGRAM WRITTEN FOR MOTOROLA 6800 BASED MICROCOMPUTERS USING SMOKE SIGNAL BROADCASTING'S BFD-68 MINIFLOPPY (TM SHUGART) DISC CONTROLLER.

DFM68 PROVIDES THE INTERFACE BETWEEN USER PROGRAMS AND THE DISC HARDWARE BY MAINTAINING THE INFORMATION NECESSARY TO ALLOW THE USER TO TRANSMIT DATA TO AND FROM DISC FILES ON A CHARACTER-BY-CHARACTER BASIS.

BY PROVIDING THIS INTERFACE, THE USER PROGRAM NEED NOT BE CONCERNED WITH:

- 1) THE ACTUAL MECHANICS OF READING AND WRITING THE DISC,
- 2) WHAT FILES ARE ON THE DISC AND WHERE THEY ARE LOCATED,
- 3) ALLOCATING AND DEALLOCATING OF DISC SPACE.

THE USER NEED ONLY BE CONCERNED WITH:

- 1) THE NAME OF THE FILE TO BE OPERATED UPON,
- 2) THE OPERATION TO PERFORM (IE, READ OR WRITE)
- 3) THE PHYSICAL DRIVE UPON WHICH THE FILE RESIDES.

RESTRICTIONS IMPOSED BY DFM68:

- 1) A SINGLE FILE MUST RESIDE ON A SINGLE PHYSICAL DISC,
- 2) RANDOM ACCESS FILES ARE NOT SUPPORTED

ALL REQUESTS FOR SERVICES ARE COMMUNICATED TO DFM BY MEANS OF A FILE CONTROL BLOCK (FCB). THE FCB IS A TABLE IN RAM MEMORY WHICH CONTAINS INFORMATION SUCH AS THE FILE NAME, OPERATION TO PERFORM, UNIT NUMBER OF DISC FOR THE FILE, AND THE DISC I/O BUFFER SPACE.

IN ORDER TO OPERATE ON A FILE THE FILE MUST MUST BE "OPENED". OPENING THE FILE ESTABLISHES THE LINKAGES TO BE ABLE TO TRANSFER DATA TO OR FROM THE FILE. SUBSEQUENT DATA TRANSFERS ARE THEN MADE BY PASSING A BYTE OF DATA THROUGH THE A REGISTER. AFTER ALL DATA TRANSFERS ARE COMPLETE, THE FILE MUST BE "CLOSED". CLOSING THE FILE UPDATES ALL INFORMATION ON THE DISC REGARDING THE FILE.

FILE CONTROL BLOCK FORMAT

THIS SECTION DESCRIBES THE ENTRIES WITHIN THE FILE CONTROL BLOCK (FCB) USED TO ACCESS DISC FILES AND COMMUNICATE WITH DFM.

THE FCB IS A TABLE 166 BYTES IN LENGTH. THIS TABLE CONTAINS ALL THE POINTERS USED TO MANIPULATE A FILE BY DFM.

THE USER MUST ALLOCATE ONE FCB FOR EACH FILE BEING OPERATED ON AT ANY ONE MOMENT. THERE IS NO RESTRICTION UPON HOW MANY FCBS MAY BE IN USE AT ANY TIME THUS ALLOWING THE USER TO OPERATE ON AS MANY FILES AS DESIRED FROM WITHIN ANY SINGLE PROGRAM.

NAME	LOCATION	USAGE
XFC	FCB+0	FUNCTION CODE
XES	FCB+1	ERROR STATUS RETURNED TO CALLER
XUN	FCB+2	UNIT NUMBER FOR OPERATION
XFN	FCB+3	1ST CHARACTER OF FILE NAME
	FCB+4	2ND CHARACTER
	FCB+5	3RD CHARACTER
	FCB+6	4TH CHARACTER
	FCB+7	5TH CHARACTER
	FCB+8	6TH CHARACTER OF FILE NAME
	FCB+9	1ST CHARACTER OF THE FILE EXTENSION
	FCB+10	2ND CHARACTER OF THE EXTENSION
	FCB+11	3RD CHARACTER OF THE EXTENSION
	XFT	FCB+12
XFS	FCB+13	FILE STATUS
XFSU	FCB+14	TRACK # OF FIRST SECTOR USED BY THE FILE
	FCB+15	SECTOR # OF THE FIRST SECTOR USED BY THE FILE
XLSU	FCB+16	TRACK # OF THE LAST SECTOR USED BY THE FILE
	FCB+17	SECTOR # OF THE LAST SECTOR USED BY THE FILE
XSUC	FCB+18	HIGH ORDER COUNT OF SECTORS USED BY THE FILE
	FCB+19	LOW ORDER COUNT OF THE SECTORS USED BY THE FILE
	FCB+20	RESERVED
	FCB+21	"
	FCB+22	"
	FCB+23	"
	FCB+24	"
	FCB+25	"
	FCB+26	"
	XNFP	FCB+27
FCB+28		LOW ORDER ADDRESS OF NEXT FCB IN ACTIVE FCB CHAIN
XBI	FCB+29	INDEX INTO DATA BUFFER
XCT	FCB+30	TRACK # OF CURRENT SECTOR ON DISC
XCS	FCB+30	SECTOR NUMBER OF THE CURRENT SECTOR ON THE DISC
	FCB+32	RESERVED
	FCB+33	"
	FCB+34	"
	FCB+35	"
	FCB+36	"
	FCB+37	"
		(DISC I/O BUFFER BEGINS WITH THE NEXT BYTE)
XNT	FCB+38	TRACK # OF NEXT SECTOR IN THE FILE
XNS	FCB+39	SECTOR # OF THE NEXT SECTOR IN THE FILE
XPT	FCB+40	TRACK # OF PREVIOUS SECTOR IN THE FILE
XPS	FCB+41	SECTOR # OF THE PREVIOUS SECTOR IN THE FILE
XSOD	FCB+42 - FCB+165	DATA PORTION OF DISC SECTOR

THE FCB ENTRIES FROM FCB+3 THROUGH FCB+26 ARE THE EXACT ENTRIES TO BE FOUND IN THE DISC FILE DIRECTORY.

DISC FILE DIRECTORY

THE DIRECTORY OF FILES PRESENT ON THE DISC BEGINS IN SECTOR 1 OF TRACK ZERO. THE FORMAT OF THE DIRECTORY IS AS FOLLOWS:

BYTE	USAGE
0	TRACK # OF NEXT DIRECTORY BLOCK (0 IF END OF DIRECTORY)
1	SECTOR # OF NEXT DIRECTORY BLOCK
2	TRACK # OF PREVIOUS DIRECTORY BLOCK
3	SECTOR # OF PREVIOUS DIRECTORY BLOCK

FIVE REPETITIONS OF THE INFORMATION CONTAINED IN THE FCB DESCRIPTION (BYTES FCB+3 THRU FCB+26)

EXCEPTION: THE FIRST DIRECTORY BLOCK (TRACK 0 SECTOR 1) CAN ONLY DESCRIBE FOUR FILES. THE FIRST FILE INFORMATION BLOCK (FIB) IS USED TO POINT TO THE START AND END OF THE LIST OF AVAILABLE SECTOR ON THE DISC. THE FORMAT OF THIS FIRST FIB IS AS FOLLOWS:

BTYE	USAGE
4	MUST BE \$FF
5 - 14	DON'T CARES (\$FF)
15	NEXT AVAILABLE BLOCK TRACK NUMBER
16	NEXT AVAILABLE BLOCK SECTOR NUMBER
17	LAST AVAILABLE BLOCK TRACK NUMBER
18	LAST AVAILABLE BLOCK SECTOR NUMBER
19	HIGH ORDER COUNT OF AVAILABLE SECTORS
20	LOW ORDER COUNT OF AVAILABLE SECTORS

INTERFACING TO DFM

THERE ARE THREE ENTRY POINTS INTO DFM68. THEY ARE:

- 1) THE DFM68 INITIALIZATION ENTRY POINT
- 2) THE DFM68 CLOSING ENTRY POINT
- 3) THE DFM68 I/O SERVICE REQUEST ENTRY POINT

THESE THREE ENTRY POINTS CORRESPOND TO THREE "JMP" INSTRUCTIONS LOCATED IN THE FIRST NINE BYTES OF DFM68 (SEE THE MEMORY MAP FOR SPECIFIC ADDRESSES).

INITIALIZATION ENTRY POINT

DFM68 MUST BE INITIALIZED BEFORE IT CAN BE USED. INITIALIZING DFM BASICALLY TELLS DFM THAT THERE ARE NO FILES CURRENTLY IN USE.

THE ENTRY POINT TO INITIALIZE DFM IS THE FIRST OF THE THREE JUMPS LOCATED IN THE BEGINNING OF DFM. THERE ARE NO ERRORS ASSOCIATED WITH INITIALIZING DFM SINCE NO DISC OPERATIONS ARE PERFORMED AND SINCE THE FUNCTION OF THIS CALL IS TO RESET ALL INTERNAL STATUS FLAGS WITHIN DFM68.

DFM CLOSING ENTRY POINT

WHEN NO FURTHER USE IS TO BE MADE OF DFM68, DFM SHOULD BE "CLOSED". CLOSING DFM SERVES TO CLOSE ANY OPEN FILES WHICH MAY NOT HAVE CLOSED.

DFM IS CLOSED BY CALLING THE SECOND OF THE THREE JUMPS LOCATED IN THE BEGINNING OF DFM. ERRORS ARE REPORTED AS FOLLOWS: A "BNE" WILL BRANCH IF AN ERROR OCCURED. IF AN ERROR OCCURED, THE ERROR TYPE IS RETURNED IN THE A REGISTER (SEE THE APPENDIX FOR THE ERROR TYPES), THE ERROR NUMBER WILL BE RETURNED IN THE B REGISTER, AND ON TYPE 1 AND 3 ERRORS, THE X REGISTER WILL BE POINTING TO THE FCB WHICH IS IN ERROR.

I/O SERVICE REQUEST ENTRY POINT

ALL SERVICE REQUESTS MADE TO DFM ARE HANDLED BY CALLING THE THIRD OF THE THREE JUMPS LOACTED IN THE BEGINNING OF DFM.

INFORMATION REGARDING THE FUNCTIONS TO BE PERFORMED BY DFM ARE PASSED TO DFM IN THE FCB. THE ADDRESS OF THE FCB IS LOADED IN THE X REGISTER WHEN DFM IS CALLED. ALL REGISTER ARE PRESERVED BY THE CALL UNLESS DATA IS RETURNED TO THE CALLER. THE CONDITION CODES ARE NOT PRESERVED; UPON RETURN FROM DFM, A "BNE" WILL BRANCH IF AN ERROR OCCURRED. IF AN ERROR OCCURS, THE ERROR STATUS BYTE IN THE FCB WILL CONTAIN THE ERROR CODE, OTHERWISE, THE ERROR STATUS BYTE WILL BE ZERO.

REPORT FREE SPACE

DFM68 WILL RETURN THE COUNT OF AVAILABLE SECTORS FOR THE DISC DRIVE REQUESTED IN THE FCB. THE HIGH ORDER BINARY COUNT WILL BE RETURNED IN THE A REGISTERS AND THE LOW ORDER COUNT IN THE B REGISTER.

OPEN A FILE FOR WRITE

DFM68 CREATES THE FILE TO BE WRITTEN BY THE NAME SPECIFIED IN THE FCB. AN ERROR WILL OCCUR IF THE FILE ALREADY EXISTS (A FILE CANNOT BE OVERWRITTEN).

WRITE TO A FILE

THE BYTE OF DATA PASSED IN THE A REGISTER IS WRITTEN TO THE FILE SPECIFIED IN THE FCB USED TO OPEN THE FILE.

CLOSE A WRITE FILE

THE FILE SPECIFIED IN THE FCB IS CLOSED. THAT IS, THE LAST BUFFER OF DATA IS WRITTEN TO THE FILE AND THE DIRECTORY ENTRY FOR THE FILE IS UPDATED. IF THE LAST BUFFER IS NOT FULL, IT WILL BE PADDED WITH NULLS. WHEN THE FILE IS READ BACK, AN END-OF-FILE CONDITION WILL NOT OCCUR UNTIL THE LAST CHARACTER OF THE LAST BUFFER IS READ.

OPEN A FILE FOR READ

THE FILE SPECIFIED IN THE FCB IS OPENED FOR READ.

READ FROM A FILE

THE NEXT BYTE IS READ FROM THE FILE SPECIFIED IN THE FCB AND RETURNED TO THE CALLER IN THE A REGISTER.

CLOSE A READ FILE

DFM RELEASES THE LINKAGES TO THE FILE BEING READ.

DELETE A FILE

THE FILE SPECIFIED IN THE FCB IS DELETED FROM THE DISC DIRECTORY AND THE SECTORS USED BY THE FILE RETURN TO THE LIST OF AVAILABLE SECTORS.

RENAME A FILE

THE NAME OF THE FILE SPECIFIED IN THE FCB IS CHANGED TO THE NAME IN BYTES FCB+45 THROUGH FCB+53 PROVIDED THAT THE NEW NAME IS NOT ALREADY IN USE.

APPENDING TWO FILES

THE FILE SPECIFIED IN BYTES FCB+45 THROUGH FCB+53 IS APPENDED TO THE PRIMARY FILE SPECIFIED BY THE FCB AND IS THEN DELETED FROM THE DISC DIRECTORY. BOTH FILES ARE ASSUMED TO RESIDE ON THE DISC DRIVE SPECIFIED IN FCB+2. THE NULLS FILLING THE UNUSED PORTION OF THE LAST BLOCK OF THE FIRST FILE REMAIN (IE, THE FILES ARE NOT COMPRESSED TOGETHER).

DIRECTORY READ SETUP

THIS COMMAND CAUSES DFM TO OPEN THE DISC DIRECTORY SPECIFIED IN THE FCB. SUBSEQUENT CALLS USING THE DIRECTORY TRANSFER CODE ARE USED TO PICK UP ONE FILE INFORMATION BLOCK (FIB) AT A TIME UNTIL AN END-OF-FILE CONDITION OCCURS. NO OTHER FUNCTIONS OF DFM MAY BE USED UNTIL THE USER IS DONE READING THE DIRECTORY.

DIRECTORY TRANSFER

THE DIRECTORY TRANSFER COMMAND IS USED TO RETRIEVE A FILE NAME AT A TIME FROM THE DIRECTORY AND MAY ONLY USED FOLLOWING A DIRECTORY SETUP COMMAND OR FOLLOWING ANOTHER DIRECTORY TRANSFER COMMAND. THE TRANSFER COMMAND CAUSES THE NEXT ACTIVE FIB ENTRY TO BE COPIED FROM THE DIRECTORY INTO BYTES FCB+3 THROUGH FCB+26 (SEE DIRECTORY FORMAT DESCRIPTION).

SINGLE SECTOR READ

THE SINGLE SECTOR READ COMMAND CAUSES A SPECIFIED SECTOR TO BE READ FROM THE DISC AND PLACED IN THE FCB DATA BUFFER.

THE TRACK NUMBER AND SECTOR NUMBER TO BE READ IS PLACED IN BYTES FCB+30 (XCT) AND FCB+31 (XCS) RESPECTIVELY.

NOTE: THE ERROR CODE RETURNED FROM THIS COMMAND IS THE ACTUAL VALUE READ BACK FROM THE WD 1771 CHIP WITH THE LEAST SIGNIFICANT BIT BEING USED TO INDICATE A SEEK ERROR.

SINGLE SECTOR WRITE

THE SINGLE SECTOR WRITE IS THE COUNTER PART OF THE SINGLE SECTOR READ COMMAND (SEE ABOVE DESCRIPTON).

WARNING: SINGLE SECTOR I/O ALLOWS ANY SECTOR TO BE READ OR WRITTEN. IT IS UP TO THE USER TO KNOW WHAT HE IS MODIFYING.

USING THE DISC FILE MANAGEMENT SYSTEM (DFM)

DFM IS THAT PORTION OF DOS68 WHICH RELATES TO THE USAGE OF DISC FILES. THIS SECTION IS PROVIDED AS AN INTRODUCTION AS HOW TO USE DFM TO READ AND WRITE DISC FILES.

IT IS RECOMMENDED THAT THE READER FIRST READ THROUGH THE DESCRIPTIONS OF THE FUNCTIONS PROVIDED BY DFM. THEN, AFTER HAVING READ THIS SECTION, READ THROUGH SEVERAL OF THE MONITOR TRANSIENT COMMAND LISTINGS TO SEE HOW THE INFORMATION PRESENTED BELOW IS IMPLEMENTED IN PRACTICE.

HOW TO READ FROM A FILE

READING FROM A FILE IS DONE IN THREE STEPS:

- 1) OPENING THE FILE FOR READ
- 2) READING FROM THE FILE
- 3) CLOSING THE FILE

1) OPENING THE FILE SETS UP THE SOFTWARE LINKAGES WITHIN DFM TO ENABLE THE USER TO THEN REQUEST DATA BE TRANSFERED FROM THE FILE.

IN ORDER TO OPEN A FILE THE USER MUST RESERVE A 166 BYTE TABLE IN HIS PROGRAM. THIS TABLE IS REFERRED TO AS A FILE CONTROL BLOCK (FCB). THE FCB IS A TABLE IN WHICH THE POINTERS TO THE FILE BEING ACCESSED ARE KEPT.

TO READ A FILE THE USER MUST FILL IN THREE ENTRIES IN THE FCB. THESE THREE ENTRIES ARE THE UNIT NUMBER ON WHICH TO FIND THE FILE, THE NAME OF THE FILE, AND THE OPERATION TO PERFORM ON THE FILE.

THE FUNCTION CODE TO OPEN THE FILE FOR READ IS PUT IN THE FIRST BYTE OF THE FCB. THE UNIT NUMBER ON WHICH THE FILE IS TO BE FOUND IS PUT IN THE THIRD BYTE OF THE TABLE (THE UNIT NUMBER MAY BE 0,1, OR 2). THE NINE BYTES FOLLOWING THE UNIT NUMBER ARE USED TO DESIGNATE THE FILE NAME. THE FIRST SIX BYTES ARE TREATED AS THE FILE NAME WHILE THE LAST THREE ARE TREATED AS THE FILE EXTENSION.

THE FIRST PORTION OF THE FCB THEN LOOKS LIKE:

FCB+0	OPERATION CODE
FCB+1	ERROR CODE RETURNED TO THE USER
FCB+2	UNIT NUMBER
FCB+3	FILE NAME (1ST CHARACTER)
FCB+4	
.	
.	
.	
FCB+8	FILE EXTENSION (1ST CHARACTER)
FCB+9	
FCB+10	FILE EXTENSION (LAST CHARACTER)

TO ACTUALLY OPEN THE FILE, LOAD THE X REGISTER WITH THE ADDRESS OF THE FIRST BYTE OF FCB AND CALL DFM. UPON RETURN, A "BNE" WILL BRANCH IF AN ERROR OCCURED. IF AN ERROR OCCURS AT ANY TIME THE NON-ZERO CODE WILL BE RETURNED IN THE 2ND BYTE OF THE FCB. THE SIMPLEST MANNER IN WHICH TO HANDLE ERRORS IS TO REQUEST DFM TO CLOSE ALL OPEN FILES. THIS IS

DONE BY CALLING "CDFM", THE DFM CLOSING ENTRY POINT. SEE THE DESCRIPTION OF THIS ENTRY POINT TO DFM FOR FURTHER DETAILS. THUS THE FOLLOWING SECTION OF CODE REPRESENTS HOW ONE WOULD OPEN A FILE FOR READ:

```

ODFM    EQU    $7780    OPEN DFM ENTRY POINT
CDFM    EQU    $7783    CLOSE DFM ENTRY
DFM     EQU    $7786    DFM SERVICE REQUEST ENTRY

        LDX    #FCB      LOAD THE ADDRESS OF THE FCB
        LDA A  #QSO4R    LOAD THE VALUE OF THE "OPEN FOR READ"
                        FUNCTION CODE
                        (SEE THE APPENDICES FOR THE FUNCTION
                        CODE VALUES)
        STA A  XFC,X     STORE THE FUNCTION CODE IN THE FCB
        JSR   DFM        ASK DFM TO OPEN THE FILE
        BEQ   FILOPN     BRANCH IF NO ERROR OCCURED
        JSR   ZTYPDE     ASK THE MONITOR TO
                        TYPE "DISC ERROR XX"
        JSR   CDFM       ASK DFM TO CLOSE ALL FILES
        JMP   ZWARMS     RESTART THE MONITOR

FILOPN  LDA A  #QSREAD   CHANGE TO FUCTION TO READ FROM
                        THE FILE
        STA A  XFC,X

```

*WE ARE NOW READY TO READ DATA FROM THE FILE

2) TO READ A BYTE FROM THE FILE, LOAD THE X REGISTER WITH THE ADDRESS OF THE FCB AND CALL DFM. DFM WILL RETURN THE NEXT BYTE FROM THE FILE IN THE A REGISTER. IF AN ERROR OCCURS A "BNE" WILL BRANCH IF AN ERROR OCCURED (END-OF-FILE IS TREATED AS AN ERROR CONDITION ALSO). THE FOLLOWING SECTION OF CODE MAY BE USED TO READ FROM A FILE:

```

        LDX    #FCB      POINT TO THE FCB
        JSR   DFM        ASK DFM TO READ A BYTE
        BEQ   READOK     BRA IF NO ERROR OCCURED
        JSR   ZTYPDE     ASK THE MONITOR TO TYPE
                        "DISC ERROR: XX"
        JSR   CDFM       ASK DFM TO CLOSE ANY OPEN FILES
        JMP   ZWARMS     RESTART THE MONITOR

READOK  EQU    *        AT THIS POINT A BYTE HAS BEEN
                        READ FROM THE FILE AND IS
                        IN THE A REGISTER.

```

3) AFTER ALL THE DATA HAS BEEN READ FROM THE FILE, THE FILE MUST BE CLOSED. CLOSING THE FILE RELEASES THE FCB FROM ITS ROLE AS TABLE OF POINTERS TO THE FILE. CLOSING THE FILE IS ACCOMPLISHED AS FOLLOWS:

```

        LDX    #FCB      POINT TO THE FCB
        LDA A  #QSRC     SET THE FUNCTION CODE TO "READ CLOSE"
        STA A  XFC,X     SAVE THE FUNCTION CODE
        JSR   DFM        ASK DFM TO CLOSE THE FILE
        BEQ   CLSOK     BRANCH IF THE FILE WAS SUCESSFULLY
                        CLOSED
        JSR   ZTYPDE     TELL THE MONITOR TO TYPE THE DISC
                        ERROR CODE
        JSR   CDFM       ASK DFM TO CLOSE ANY OPEN FILES
        JMP   ZWARMS     RESTART THE MONITOR

CLSOK   EQU    *        AT THIS POINT THE FILE IS CLOSED

```

HOW TO WRITE (CREATE) A FILE

WRITING A FILE FOLLOWS THE IDENTICAL SEQUENCE OF OPERATIONS AS READING FROM A FILE WITH THE FOLLOWING EXCEPTIONS:

- 1) WHEN OPENING THE FILE, THE FUNCTION CODE IS "QSO4W" TO OPEN THE FILE FOR WRITE.
 - 2) "QSWRIT" IS USED IN PLACE OF "QSREAD" WHEN PERPARING THE FILE TO BE WRITTEN.
 - 3) WHEN DATA IS TO BE WRITTEN TO THE FILE, THE DATA MUST BE IN THE A REGISTER WHEN DFM IS CALLED.
 - 4) WHEN CLOSING THE FILE, "QSWC" USED INSTEAD OF "QSRC".
- (SEE THE APPENDICES FOR THE FUNCTION CODE VALUES)

BFD-68 SYSTEM HARDWARE

PREFACE

THIS MANUAL DESCRIBES THE CHARACTERISTICS OF THE SSB MINIFLOPPY INTERFACE. THIS INTERFACE ALLOWS USERS OF THE SWTPC 6800 MICRO-COMPUTER SYSTEM TO EASILY INTERFACE UP TO THREE S.A. 400 MINIFLOPPY DISKETTE STORAGE DRIVES.

BOOT AND I/O ROUTINES

TO FACILITATE DISC I/O A ROM HAS BEEN PROVIDED ON THE DISC INTERFACE BOARD. THE ROM CONTAINS ALL NECESSARY I/O ROUTINES TO READ, WRITE, SEEK, STEP AND RESTORE THE DISC DRIVES. IN ADDITION, A DISC BOOT ROUTINE HAS BEEN INCLUDED IN THE ROM TO ALLOW EASY ACCESS TO ANY SYSTEM STORED ON A DISC.

DISC CONTROL

DISC CONTROL IS ACHIEVED BY THE USE OF THE WESTERN DIGITAL FD1771B-01 FLOPPY DISC CONTROLLER IC. THIS IC CONTROLS READ/WRITE FORMAT, HEAD STEP, SEEKING, AND CHECKS THE WRITE PROTECT STATUS OF THE DISC. THE READ/WRITE FORMAT CAN BE PROGRAMMED TO MANY FORMATS INCLUDING IBM 3740 FORMAT. CRC GENERATION AND CHECKING ARE ALSO PERFORMED WITHIN THE FD1771B-01 IC. ADDITIONAL INFORMATION ON PROGRAMMING THE FLOPPY DISC CONTROLLER CHIP MAY BE FOUND IN THE WESTERN DIGITAL FD1771B-01 PRODUCT GUIDE.

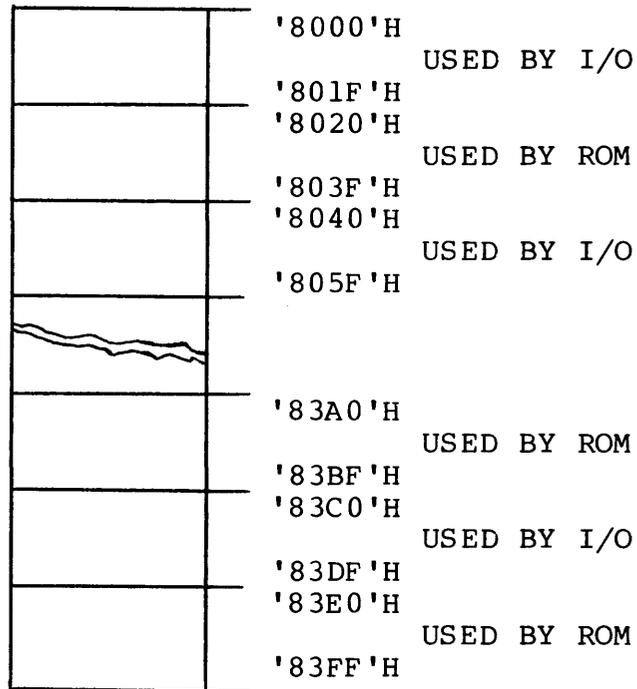
FD1771B-01 CONTROL

ALL COMMUNICATIONS BETWEEN THE HOST SYSTEM AND THE FD1771B-01 IC ARE THROUGH A 6820 PIA. CONTROL LINE FUNCTIONS OF THE FD1771B-01 ARE HANDLED BY THE A-PORT AND OTHER PROGRAMMABLE CONTROL INPUT/OUTPUT PINS OF THE PIA. DATA TO AND FROM THE FLOPPY DISC CHIP IS TRANSMITTED THROUGH THE B-PORT OF THE PIA.

ROM

THE ROM LOCATED AT U17 PROVIDES 512X8 BITS OF INFORMATION FOR THE USER. THE ROM ADDRESSING IS DECODED TO USE ALL UNUSED ADDRESSES IN THE I/O PAGE BETWEEN '8020'H AND '83FF'H. THE 9328 DECODER U23 PROVIDES HIGH ORDER ADDRESS DECODING. USE OF UNUSED AREAS OF THE I/O PAGE IS ACHIEVED BY REQUIRING ADDRESS BIT 5 TO BE TRUE TO ENABLE THE ROM. ADDRESS BITS 0-4 OF THE ROM ARE DRIVEN BY ADDRESS LINES 0-4. ADDRESS BITS 5-8 ARE DRIVEN BY ADDRESS LINES 6-9. THE FIGURE BELOW ILLUSTRATES THE INTERLEAVING OF I/O AND ROM ADDRESSES.

ROM MEMORY MAP



THE ABOVE FIGURE SHOWS I/O MEMORY ILLUSTRATED PAGES OF 32 WORDS. WHEN BIT 5 IS A 0 THAT 32 WORD PAGE IS USED AS I/O LOCATIONS. WHEN BIT 5 IS A 1 THAT PAGE IS DECODED BY THE ROM AS ONE OF ITS ADDRESSES AND THE PARTICULAR BYTE OF INFORMATION REQUIRED IS PLACED ON THE SYSTEM DATA BUSS.

DISC INTERFACE

ADDRESS DECODING FOR THE 6820 PIA IS PROVIDED FOR BY NAND GATE U22 AND THE 9328 DECODER, U16. THE 74LS30 8-INPUT NAND GATE U22 GENERATES THE NON-PROGRAMMABLE ADDRESS DECODING OF THE ADDRESS BITS A5-A12 AT U22-8 TO THE CS2- INPUT AT U4 PIN 23. REGISTER SELECTION WITHIN THE PIA IS CONTROLLED BY ADDRESS LINES A0 AND A1 AT U4 PINS 36 AND 35.

THE PIA DATA INPUTS (U4, 26-33) ARE TIED DIRECTLY TO THE THE NEGATIVE TRUE SYSTEM DATA BUSS. CARE MUST BE EXERCISED IN PROGRAMMING THE PIA AS THE PIA EXPECTS TO SEE POSITIVE TRUE DATA AT THE SYSTEM PORT. ALL CONTROL FUNCTIONS SENT TO THE PIA SHOULD FIRST BE COMPLEMENTED BY THE CONTROLLING PROGRAM. ALL CONTROL INFORMATION RECEIVED FROM THE PIA SHOULD BE INTERPRETED AS NEGATIVE TRUE DATA BY THE RECEIVING PROGRAM. THE DAL- LINES OF THE FD1771B-01 IC ARE TIED TO THE B-PORT OF THE PIA AT U11(10-17). DATA TO AND FROM THE FD1771B-01 ON THE DAL LINES IS DEFINED AS NEGATIVE TRUE DATA. AS DATA IS NOT INVERTED THROUGH THE PIA, IT IS NOT NECESSARY TO INVERT DATA FROM OR TO THE FD1771B-01 AS IS REQUIRED WITH THE CONTROL REGISTERS OF THE PIA. DATA EXCHANGE BETWEEN THE PIA AND FD1771B-01 ARE CONTROLLED BY THE FOLLOWING CONTROL LINES:

WRITE ENABLE

CB2 OF THE PIA IS USED TO STROBE INFORMATION FROM THE 6820 INTO THE FD1771B-01. CRB BITS 3-5 SHOULD BE PROGRAMMED TO '101'. IN THIS MODE CB2 IS CLEARED ON THE POSITIVE TRANSITION OF THE FIRST 'E' PULSE FOLLOWING A WRITE 'B' DATA REGISTER OPERATION AND SET HIGH ON THE POSITIVE TRANSITION OF THE NEXT 'E' PULSE. DURING WRITE OPERATIONS THE READ FLIP-FLOP, U15, MUST BE DISABLED BY PROGRAMMING A-PORT BITS 6 AND 7 TO 1 AND 0 RESPECTIVELY.

REGISTER SELECT

A-PORT BITS 0 AND 1 DRIVE THE REGISTER SELECT LINES OF THE FLOPPY DISC CONTROLLER CHIP. PA0 DRIVES ADDRESS LINE A0 AND PA1 DRIVES ADDRESS LINE A1.

CHIP SELECT

A-PORT BIT 2 (U4-4) DRIVES THE CHIP SELECT INPUT OF THE FD1771B-01. WHEN PA2 IS 0 THE FLOPPY DISC CHIP WILL BE ENABLED.

READ ENABLE

READING INFORMATION FROM THE FDC IS CONTROLLED BY ITS READ ENABLE INPUT AT U11-4. THIS INPUT IS DRIVEN BY THE READ ENABLE FLIP-FLOP U15-6. FLIP-FLOP U15 IS USED IN TWO MODES. MODE ONE IS USED WHEN READING CONTROL REGISTERS IN THE FD1771B-01. MODE TWO IS USED WHEN READING DATA FROM THE DISC. IN MODE ONE PA6 (U4-8) IS PROGRAMMED TO '0'. PA6 LOW FORCES U15-6 TO THE LOW STATE THEREBY ENABLING INFORMATION FROM SELECTED REGISTER ONTO THE DAL LINES (U11,7-14). THIS INFORMATION CAN THEN BE READ IN ON THE B-PORT OF THE PIA. IN MODE TWO PA6 AND PA7 (U4-8,9) ARE PROGRAMMED TO '1'S. WHEN DRQ (U11-38) COMES TRUE AND PA7 IS TRUE FLIP-FLOP U15 WILL BE SET ON THE POSITIVE TRANSITION ON THE 1 MHZ CLOCK (U15-12). DRQ IS ALSO TIED TO THE CA2 INPUT OF THE PIA (U4-39). THIS PROGRAMMABLE PIN IS PROGRAMMED AS AN INPUT TRIGGERED BY A POSITIVE TRANSITION (CRA BITS 3-5 =110). ON THE POSITIVE TRANSITION OF DRQ, IRQA AT U4-38 WILL BE SET LOW (IRQA REFLECTS THE STATUS OF THE BIT SET BY THE POSITIVE TRANSITION AT INPUT CA2(U4-39)). THE IRQA OUTPUT IS USED TO DRIVE THE K INPUT FOR READ ENABLE FLIP-FLOP U15. AS LONG AS IRQA REMAINS ACTIVE LOW U15 WILL NOT BE ALLOWED TO RESET. U15 SETTING CAUSES A BYTE OF DATA FROM THE FLOPPY DISC TO BE PLACED ON THE DAL LINES (U11-7,14). WHEN THE PROCESSOR DETECTS CRA BIT 6 SET IT CAN THEN READ THE BYTE OF DATA ON THE DAL LINES THROUGH THE B-PORT DATA REGISTER. AFTER THE PROCESSOR HAS READ THE DATA FROM THE B-PORT THE FLAG IN THE A-PORT CONTROL REGISTER IS CLEARED BY READING THE A-PORT DATA REGISTER. READING THE A-PORT DATA REGISTER ALSO DEACTIVATES IRQA. THIS ALLOWS U15 TO RESET THEREBY PREPARING THE FLOPPY DISC CHIP FOR THE NEXT BYTE OF DATA FROM THE DISC.

HEAD LOAD TIMING

HEAD LOAD TIME OF THE S.A. 400 MINIFLOPPY IS APPROXIMATELY 75 MILLISECONDS. THE 9602 ONESHOT (U14) PROVIDES THE DELAY SIGNAL REQUIRED FOR PROPER OPERATION OF THE FD1771B-01. THE TIME DELAY GENERATED PREVENTS THE FLOPPY DISC CONTROLLER FROM READING OR WRITING BEFORE THE HEAD HAS HAD TIME TO SETTLE.

DISC INTERFACE SIGNALS

DISC SELECT

DURING OPERATION ONE OF THREE DISCS MAY BE SELECTED AT ANY ONE TIME. DISC SELECT IS CONTROLLED BY PIA A-PORT BITS 3, 4 AND 5. PA6 PROGRAMMED TO A '1' SELECTS DISC 0. PA4 PROGRAMMED TO A '1' SELECTS DISC 1 AND PA5 PROGRAMMED TO A '1' SELECTS DISC 2. THE PIA DISC SELECT LINES ARE BUFFERED BY THE 7438 NAND BUFFER LOCATED AT U3. U3 PROVIDES THE REQUIRED DRIVE CAPABILITY NEEDED TO DRIVE THE DISC INTERFACE BUSS. IN ORDER FOR A DISC TO BE SELECTED BOTH HEAD LOAD FROM THE FDC AND ONE OF THE DISC SELECT BITS FROM THE PIA MUST BE ACTIVE. HEAD LOAD FROM THE FD1771B-01 IS BUFFERED BY BUFFER-INVERTER U9(1,2) AS THE FDC OUTPUTS WILL DRIVE ONLY ONE NORMAL TTL LOAD.

MOTOR ON

THE MOTOR ON FUNCTION HAS BEEN MODIFIED FROM THE ORIGINAL CONTROLLER. THE MOTOR IS NOW TURNED ON AS SOON AS THE DISC IS SELECTED AND WILL STAY ON AS LONG AS THE DISC SYSTEM IS ACCESSED. U10 IS WIRED AS A RETRIGGERABLE ONE-SHOT AND HAS A PERIOD OF APPROXIMATELY 30 SECONDS. AFTER THE LAST HEAD LOAD, U9-2 GOES HIGH WHICH REVERSE BIASES D1 ALLOWING THE 555 TIMER TO TIME OUT. U3-3 IS A BUFFERED OUTPUT USED TO DRIVE THE MON- LINE OF THE SA400 DRIVE.

ONE-SHOT U14 IS USED TO PROVIDE A MOTOR START DELAY AND A HEAD LOAD DELAY. THE MOTOR START DELAY IS DISABLED BY U10-3 AFTER THE MOTOR IS STARTED. THIS ALLOWS FASTER ACCESS TO THE DISC.

WRITE DATA

THE WRITE DATA SIGNAL AT U11-31 IS BUFFERED BY U2-1. WRITE DATA ON THE DISC INTERFACE BUSS IS NEGATIVE TRUE DATA.

WRITE GATE

THE WRITE GATE SIGNAL AT U11-30 IS BUFFERED BY U2-4. WRITE GATE ALONG WITH WRITE DATA CONTROLS WRITING OF DATA TO THE SELECTED DISC.

STEP

THE STEP PULSE AT U11-15 IS BUFFERED TO THE DISC INTERFACE BUSS BY U2-9. THE STEP OUTPUT PROVIDES THE STEP INSTRUCTION TO THE DISC AT A CONTROLLED RATE. THE STEP RATE IS PROGRAMMED BY THE USER. FOR THE S.A. 400 THE STEP RATE SHOULD BE PROGRAMMED TO 40 MSEC PER STEP. FOR ADDITIONAL INFORMATION ON STEP RATES SEE THE WESTERN DIGITAL FD1771B-01 PRODUCT GUIDE.

DIRECTION

THE DIRECTION OUTPUT OF THE FD1771B-01 (U11-16) IS BUFFERED BY U2-12. FOR STEP-IN (TOWARDS THE DISC HUB) THE DIRECTION LINE WILL BE HIGH. FOR STEP-OUT THE DIRECTION LINE WILL BE LOW (THIS LEVEL WILL BE REVERSED ON THE BUSS).

TRACK 00

THE TRACK 00 STATUS OF THE SELECTED DISC IS BUFFERED BY U1-2 AND IS ANDED TO FKTR00-. THIS SIGNAL IS GENERATED BY A ONE-SHOT WHICH IS TRIGGERED BY THE POWER-ON/RESET PULSE GENERATED BY U21-7. THIS SIGNAL "FAKES" THE WD1771-1 INTO THINKING IT IS ON TRACK 00. THIS ALLOWS THE SYSTEM TO RESPOND FASTER AFTER A RESET.

WRITE PROTECT

WRITE PROTECT IS BUFFERED BY U1-5. RESISTOR R5 PROVIDES THE REQUIRED PULLUP. THE WRITE PROTECT LINE REFLECTS THE STATUS OF THE CURRENTLY ACTIVE DISC. IF THE WRITE PROTECT HOLE IN THE DISC IS COVERED THE WRITE PROTECT LINE WILL BE LOW. THE BUFFERED WRITE PROTECT LINE U1-6 DRIVES U11-36. BEFORE DOING ANY WRITE OPERATIONS THE WRITE PROTECT LINE IS SAMPLED. IF THE LINE IS LOW THE WRITE OPERATIONS WILL BE ABORTED BY THE CONTROLLER CHIP.

READ DATA

READ DATA- IS BUFFERED BY NAND GATE U19-(4,5). IT IS THEN SENT THROUGH A ONE-SHOT TO SHAPE THE SIGNAL. FROM HERE THERE ARE TWO OPTIONS: OPTION 1) IS TO USE THE EXTERNAL DATA SEPARATOR, AND OPTION 2) IS TO USE THE DATA SEPARATOR INTERNAL TO THE FD1771-1. THE FDC BOARD IS CONFIGURED TO OPERATE WITH THE EXTERNAL DATA SEPARATOR. THE INTERNAL SEPARATOR MAY BE SELECTED BY CUTTING THE TRACES BETWEEN PADS B,C, AND D AND INSERTING JUMPERS TO THE ALTERNATE PADS. THE DIFFERENCE BETWEEN THE TWO SEPARATORS IS ONE OF RESOLUTION; THE EXTERNAL SEPARATOR IS BETTER AT REJECTING JITTER FROM AN SA400, AND THUS IT IS RECOMMENDED THAT THE EXTERNAL DATA SEPARATOR BE USED.

THE EXTERNAL DATA SEPARATOR CONSISTS OF IC'S U8, U9, U12, U18, U19, AND U20. THE SEPARATOR WORKS BY GENERATING "WINDOWS" THROUGH WHICH DATA AND CLOCK PULSES ARE GATED FROM READ DATA- TO THE FD1771-1. THE SYNCHRONIZATION OF THE SEPARATOR TO THE INCOMING DATA IS DONE BY RETRIGGERABLE ONE-SHOTS U8, U18, ASSOCIATED GATES, AND FLIPFLOPS.

INDEX PULSE

THE INDEX PULSE IS BUFFERED BY AND GATE U1-13. PULLUP IS PROVIDED BY RESISTOR R6. THE BUFFERED INDEX PULSE SIGNAL AT U11-11 DRIVES U11-35. THIS SIGNAL PROVIDES SYNCHRONIZATION INFORMATION FOR THE FLOPPY DISC CHIP.

PREVENTATIVE MAINTENANCE

ACCORDING TO SHUGART ASSOCIATES THERE IS NO PREVENTATIVE MAINTENANCE REQUIRED FOR THE S.A. 400 MINIFLOPPY. IF ADDITIONAL INFORMATION IS DESIRED A COPY OF THE SHUGART SERVICE MANUAL MAY BE PURCHASED FROM SMOKE SIGNAL BROADCASTING. THERE ARE NO ADJUSTMENTS REQUIRED ON THE SSB MINIFLOPPY INTERFACE.

COOLING REQUIREMENTS

THE BFD-68 DISC INTERFACE HAS NO SPECIAL COOLING REQUIREMENTS, OTHER THAN PROVIDING AIR SPACE ON ALL FOUR SIDES OF THE CABINET. SHOULD THE USER DECIDE TO ADD A FAN, ONE MAY BE ORDERED FROM SMOKE SIGNAL BROADCASTING OR THE UNIT MAY BE SENT IN FOR MODIFICATION. CAUTION: 120 VOLTS IS PRESENT INSIDE THE CABINET. REMOVE ALL POWER TO THE UNIT BEFORE OPENING THE CABINET AND INSTALLING A FAN.

INSTALLING ADDITIONAL DRIVES

ADDITIONAL SHUGART SA400 DRIVES MAY BE INSTALLED IN THE FIELD. TO INSTALL A SECOND OR THIRD DRIVE, PROCEED AS FOLLOWS:

- 1) LOCATE THE DRIVE SELECT JUMPERS LOCATED IN A DIP SOCKET ON THE TOP CORNER OF THE BOARD ON THE SA400 DRIVE.
- 2) THE JUMPERS ARE CUT AS SHOWN IN THE TABLE BELOW:

DRIVE	CUT	REMOVE
0	MX, DS2, DS3	
1	MX, DS1, DS3	RESISTOR PACK 760-3-R150 OHM
2	MX, DS1, DS2	RESISTOR PACK 760-3-R150 OHM

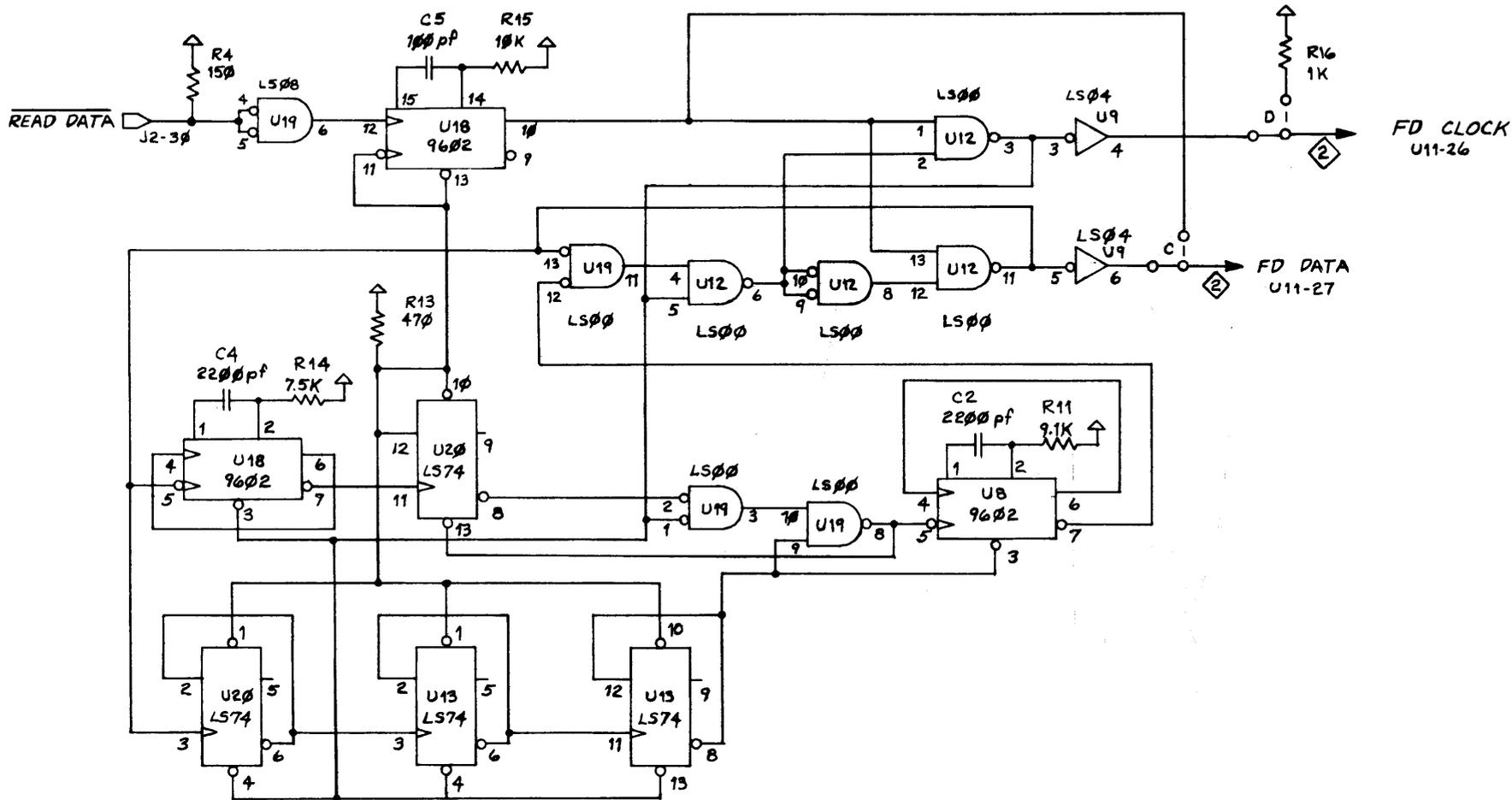
NOTE: THE RESISTOR PACK IS REMOVED FROM ALL DRIVES EXCEPT THE DRIVE WHICH IS ON THE END OF THE RIBBON CABLE CONNECTING TO THE CONTROLLER (NORMALLY THIS IS DRIVE ZERO).

DISKETTE REQUIREMENTS

THE SMOKE SIGNAL BROADCASTING BFD-68 USES THE STANDARD SIZE MINIFLOPPY MEDIA WITH ONE INDEX HOLE. FOR MAXIMUM FLEXIBILITY IN ADAPTING OUR SYSTEM TO SPECIAL USER REQUIREMENTS, WE USE A SOFT-SECTORED DISC FORMAT. THUS, DISKETTES DESIGNED FOR THE SPECIALIZED REQUIREMENTS OF HARD-SECTORED SYSTEMS SUCH AS THE NORTHSTAR WHICH USE MULTIPLE INDEX HOLES WILL NOT WORK WITH THE BFD-68. IF YOU INADVERTENTLY TRY TO FORMAT A MULTIPLE INDEX HOLE DISKETTE, THE FORMATTING PROGRAM WILL REPORT A VERY LARGE NUMBER OF "BAD SECTORS".

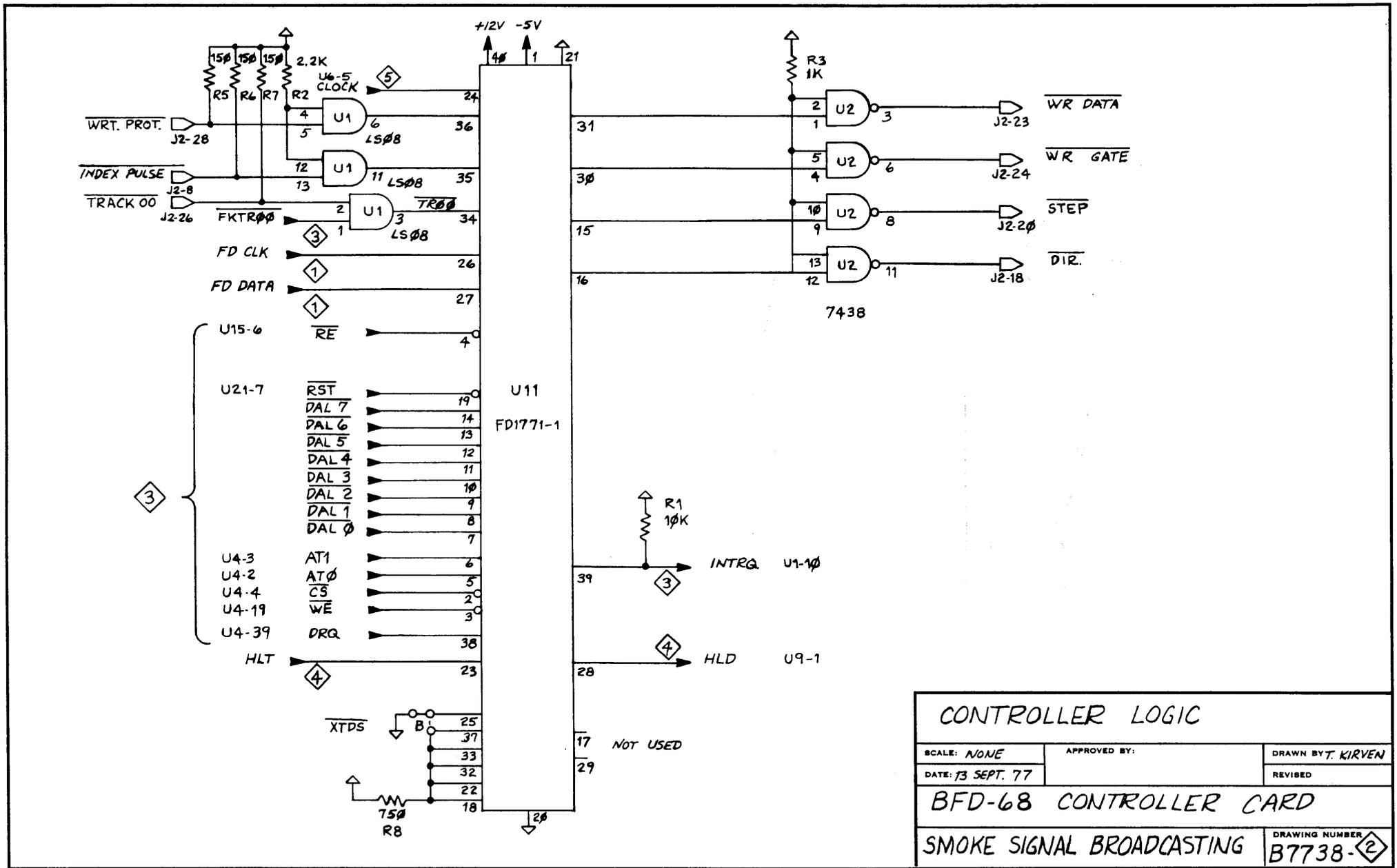
DELAY ON POWER ON

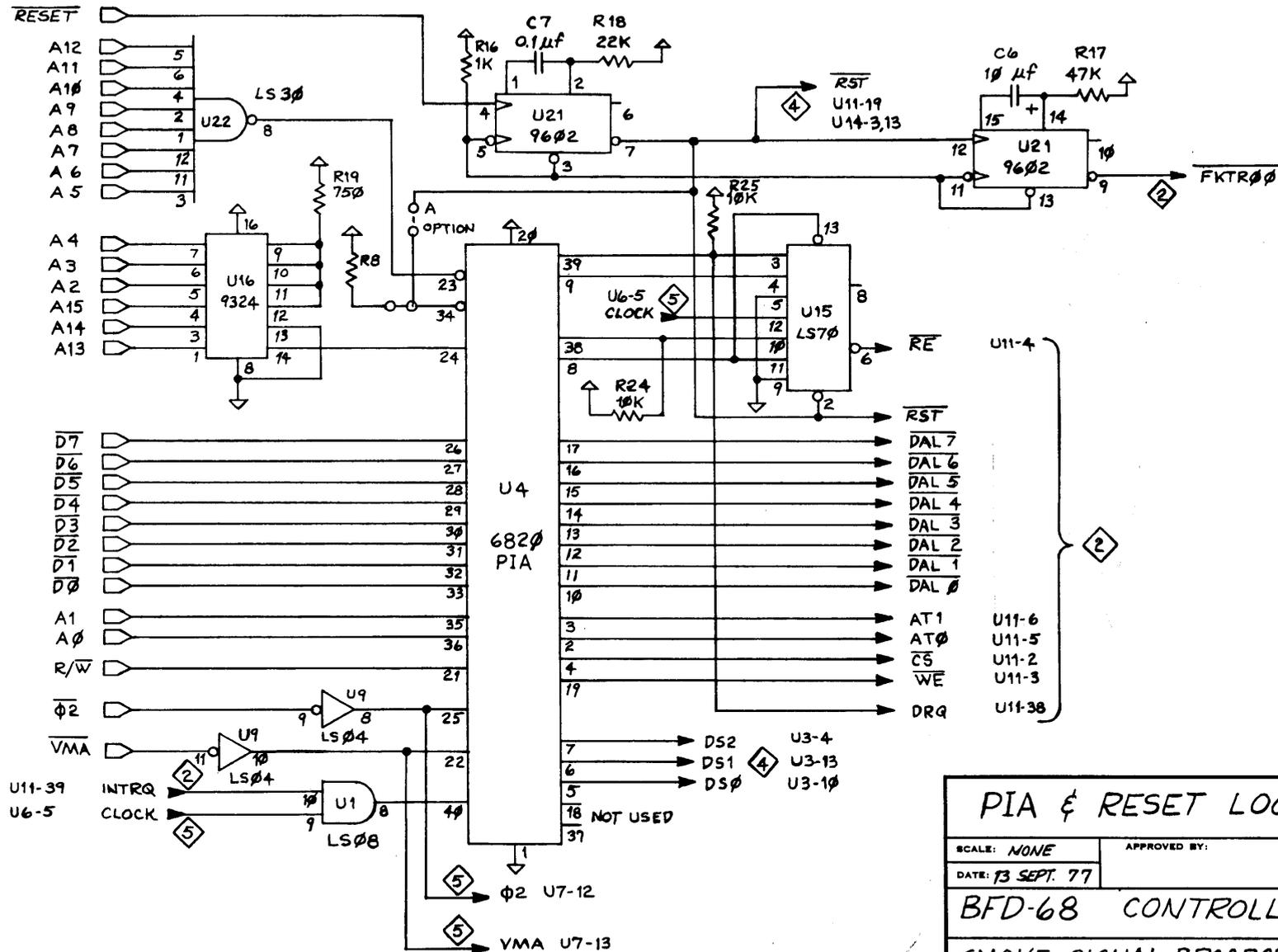
THERE IS A DELAY OF SEVERAL SECONDS WHEN FIRST BOOTING THE SYSTEM AFTER APPLYING POWER TO THE HOST COMPUTER ON A TWO OR THREE DRIVE SYSTEM. THIS IS A FUNCTION OF THE LOGIC INTERFACE TO THE CONTROLLER CHIP AND OCCURS ONLY ON TWO OR THREE DRIVE SYSTEMS.



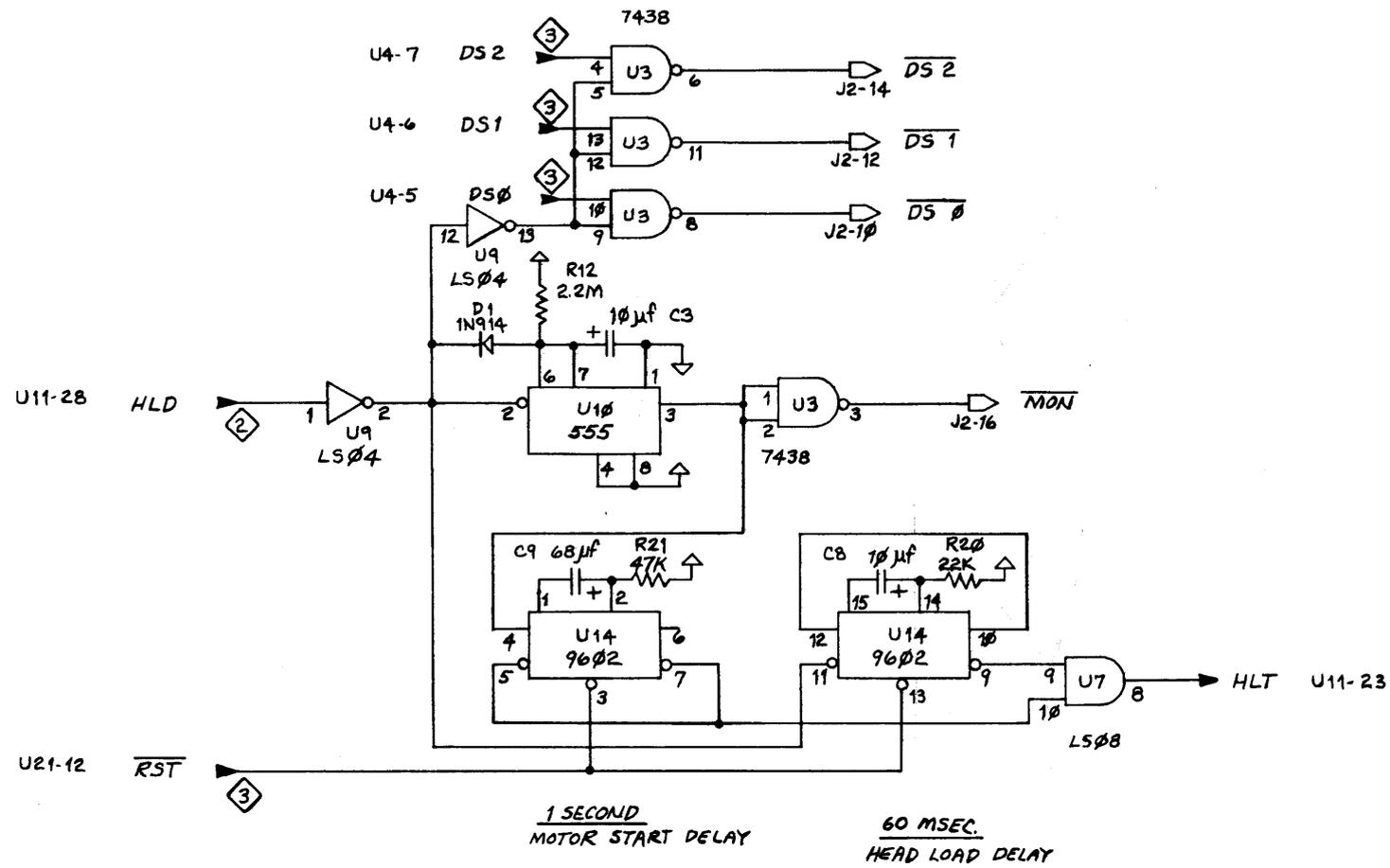
CLOCK SEPARATOR LOGIC

SCALE: NONE	APPROVED BY:	DRAWN BY T. KIRVEN
DATE: 13 SEPT. 77		REVISED
BFD-68 CONTROLLER CARD		
SMOKE SIGNAL BROADCASTING		DRAWING NUMBER B7738-1





PIA & RESET LOGIC		
SCALE: NONE	APPROVED BY:	DRAWN BY: T. KIRVEN
DATE: 13 SEPT. 77		REVISED
BFD-68 CONTROLLER CARD		
SMOKE SIGNAL BROADCASTING		DRAWING NUMBER B7738-3

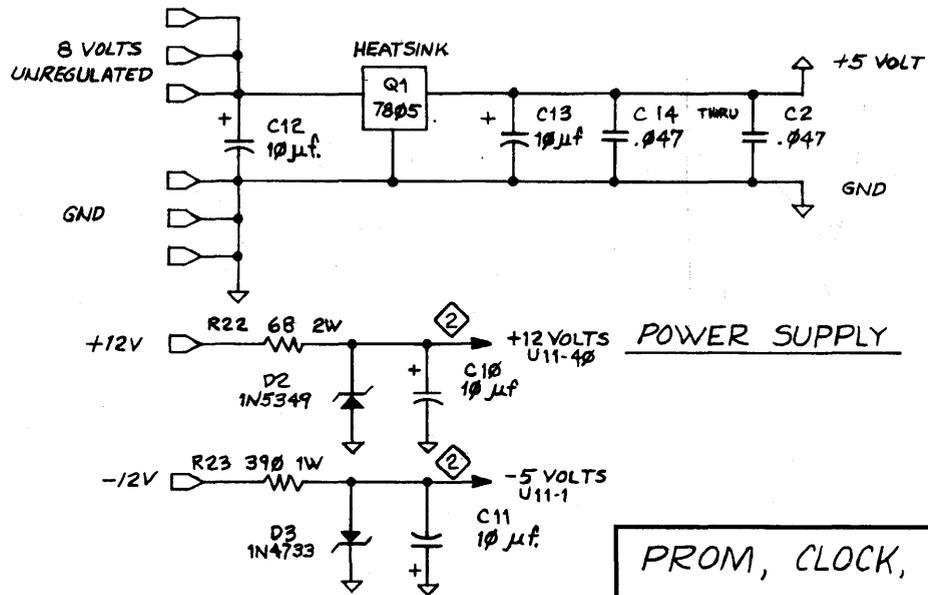
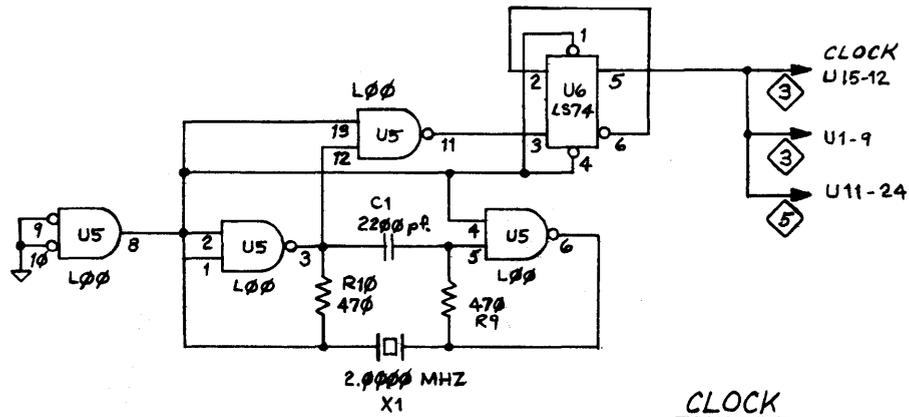
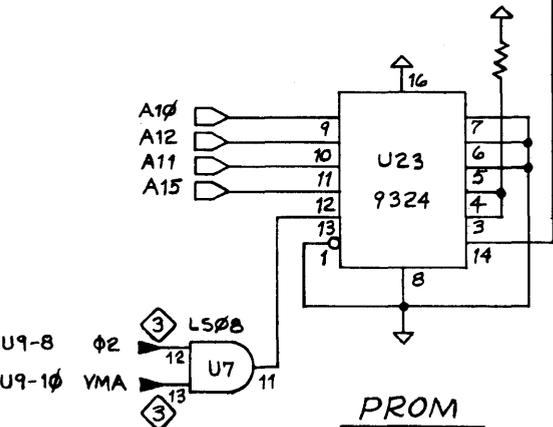
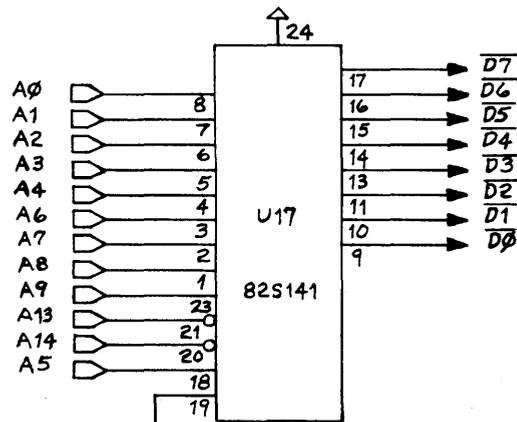


1 SECOND
MOTOR START DELAY

60 MSEC.
HEAD LOAD DELAY

DELAY LOGIC

SCALE: NONE	APPROVED BY:	DRAWN BY: KIRVEN
DATE: 13 SEPT. 77		REVISED
BFD-68 CONTROLLER CARD		
SMOKE SIGNAL BROADCASTING		DRAWING NUMBER B7738-4



PROM, CLOCK, POWER SUPPLY

SCALE: NONE	APPROVED BY:	DRAWN BY T. KIRVEN
DATE: 13 SEPT 77		REVISED
BFD-68 CONTROLLER CARD		
SMOKE SIGNAL BROADCASTING		DRAWING NUMBER B-7738-5

GENERAL INFORMATION

LIMITED WARRANTY

SMOKE SIGNAL BROADCASTING GUARANTEES THE BFD-68 HARDWARE FOR A PERIOD OF 90 DAYS FROM DATE OF PURCHASE. SMOKE SIGNAL BROADCASTING WILL, AT ITS OPTION, REPAIR OR REPLACE ANY BFD-68 SYSTEM WITH A HARDWARE DEFECT RETURNED TO IT POSTPAID WITHIN 90 DAYS FROM DATE OF PURCHASE; PROVIDED THAT, IN ITS OPINION, THE DEFECT WAS NOT CAUSED BY IMPROPER HANDLING OR IMPROPER CONNECTION TO THE HOST COMPUTER OR A MALFUNCTION OF THE HOST COMPUTER. THE LIABILITY OF SMOKE SIGNAL BROADCASTING IS SPECIFICALLY LIMITED TO REPAIR OR REPLACEMENT OF THE BFD-68 HARDWARE AND SHALL NOT EXTEND TO INCIDENTAL DAMAGES SUFFERED BY THE USER; NOR, SHALL SMOKE SIGNAL BROADCASTING BE LIABLE FOR ANY REPRESENTATION AS TO THE SUITABILITY OF THE BFD-68 TO ANY PARTICULAR USER APPLICATION UNLESS SUCH REPRESENTATION IS IN WRITING AND SIGNED BY AN OFFICER OF SMOKE SIGNAL BROADCASTING.

WHILE IN THE INTEREST OF GOOD CUSTOMER RELATIONS, SMOKE SIGNAL BROADCASTING WILL ATTEMPT TO CORRECT ANY SOFTWARE ERRORS BROUGHT TO ITS ATTENTION, THE SOFTWARE IS PROVIDED WITHOUT WARRANTY.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED.

SOFTWARE LICENSE

THE PURCHASER OF A BFD-68 SYSTEM PURCHASES, IN ADDITION TO THE HARDWARE, A LICENSE FOR THE LIMITED USE OF THE DOS-68 SOFTWARE SUPPLIED WITH THE SYSTEM. THIS LICENSE ALLOWS THE PURCHASER TO USE THE SOFTWARE ON ANY DISC SYSTEM MANUFACTURED BY SMOKE SIGNAL BROADCASTING AND TO MAKE COPIES OF THE SOFTWARE FOR USE ON ANY DISC SYSTEM MANUFACTURED BY SMOKE SIGNAL BROADCASTING. USE OF THE SOFTWARE ON ANY OTHER DISC SYSTEM OR THE COPYING OF THE SOFTWARE FOR ANY OTHER USE IS A VIOLATION OF THIS LICENSE UNLESS SPECIFIC WRITTEN APPROVAL FOR OTHER USES HAS BEEN OBTAINED FROM AN OFFICER OF SMOKE SIGNAL BROADCASTING.

SYSTEM ACCESSORIES

THE FOLLOWING ACCESSORIES FOR THE BFD-68 ARE AVAILABLE FROM SMOKE SIGNAL BROADCASTING. NORMALLY, THESE ITEMS ARE KEPT IN STOCK AND ARE AVAILABLE FOR IMMEDIATE DELIVERY.

SHUGART SA400 SERVICE MANUAL	\$ 8.00
SHUGART SA400 OEM MANUAL	\$ 5.00
ADDITIONAL DISK DRIVE	\$ 360.00
COOLING FAN	\$ 18.00
BLANK DISKETTE	\$ 5.95
DISKETTE WITH DOS-68 LOCATED AT \$7000	\$ 6.95
DISKETTE WITH DOS-68 LOCATED AT \$D000	\$ 9.95
DOS-68 PROGRAM SOURCE LISTINGS (TRANSIENT COMMAND AND MONITOR LISTINGS. DFM LISTING NOT AVAILABLE)	\$ 30.00
BASIC COMPILER THIS BASIC IS A COMPREHENSIVE BUSINESS ORIENTED BASIC FOR SERIOUS BASIC USERS	\$ 325.00
BASIC COMPILER MANUAL	\$ 10.00
SE-1 TEXT EDITING SYSTEM (DISKETTE)	\$ 29.00
SA-1 MNEMONIC ASSEMBLER (DISKETTE)	\$ 29.00
SE-1/ SA-1 EDITOR/ASSEMBLER COMBINATION (DISKETTE)	\$ 53.00
TD-1 TRACE-DISASSEMBLER (CASSETTE) (ADD \$5.95 FOR DISKETTE)	\$ 19.95
SG-1 SOURCE GENERATOR (CASSETTE) (ADD \$5.95 FOR DISKETTE) SG-1 IS A DISASSEMBLER WHICH RECONSTRUCTS SOURCE FILES WHICH CAN BE DIRECTLY ASSEMBLED OR EDITED.	\$ 24.95

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

USER GROUP INFORMATION

SMOKE SIGNAL BROADCASTING OPERATES A 6800 PROGRAM USER'S GROUP. PURCHASE OF A BFD-68 AND RETURN OF THE WARRANTEE REGISTRATION FORM ENTITLES THE USER TO A ONE YEAR MEMBERSHIP TO THE USER'S GROUP. THE PURPOSE OF THE GROUP IS TO PROVIDE A LOW-COST PROGRAM EXCHANGE SERVICE TO GROUP MEMBERS. WE DO NOT INTEND THE USER'S GROUP TO BECOME A PROFIT CENTER FOR SMOKE SIGNAL BROADCASTING, HOWEVER, WE WILL ATTEMPT TO RECOVER THE DIRECT EXPENSES OF PROGRAM DUPLICATION, ADVERTISING OF THE USER'S GROUP AND OF EMPLOYEES ASSIGNED TO USER GROUP PROJECTS.

TO HELP US MEET THE GOAL OF A LOW COST PROGRAM EXCHANGE SERVICE, WE WOULD APPRECIATE THE CONTRIBUTION OF ALL TYPES OF PROGRAMS FOR 6800 BASED SYSTEMS - NOT NECESSARILY DISC BASED SYSTEMS.

WE ARE PARTICULARLY INTERESTED IN ADDITIONAL TRANSIENT COMMANDS FOR THE BFD-68. IF EVERYONE WILL SHARE WITH US THE PROGRAMS THEY HAVE CREATED TO MAKE THE OPERATION OF THE BFD-68 MORE CONVENIENT TO THEM, IT WILL QUICKLY ENHANCE THE VALUE OF ALL OUR SYSTEMS.

IT IS HOPED THAT SHORTER PROGRAMS, 500 BYTES OR SO, WILL BE CONTRIBUTED WITHOUT CHARGE. FOR LONGER PROGRAMS, WHERE THE CONTRIBUTOR NEEDS TO RECOVER SOME OF HIS DEVELOPMENT COSTS, A ROYALTY WILL BE PAID. LARGE GENERAL PURPOSE PROGRAMS (BASIC, FORTRAN, EDITORS, ETC) WILL BE EXTENSIVELY ADVERTISED TO INSURE WIDE DISTRIBUTION, LOW COST TO THE USER AND REASONABLE COMPENSATION TO THE PROGRAM AUTHOR.

WE BELIEVE THAT THE BFD-68 IS BY FAR THE BEST DISC SYSTEM AVAILABLE TO THE MICROCOMPUTER USER TODAY. YOUR SUPPORT OF THE USER'S GROUP WILL ENABLE US TO PROVIDE EVOLUTIONARY CHANGES TO THE SYSTEM THAT WILL KEEP IT THE LEADER IN MICROCOMPUTER DISC SYSTEMS.

ACCESSORY ORDER FORM

NAME _____
 ADDRESS _____
 CITY _____ STATE _____ ZIP _____
 PHONE _____

PLEASE SEND ME THE FOLLOWING BFD-68 ACCESSORIES:

QUANTITY	ITEM	AMOUNT
	SHUGART SA400 SERVICE MANUAL @ \$8.00	_____
	SHUGART SA400 OEM MANUAL @ \$5.00	_____
	ADDITIONAL DISC DRIVES @ \$360.00	_____
	COOLING FAN @ \$18.00	_____
	BLANK DISKETTES @ \$5.95	_____
	DISKETTES WITH DOS-68 AT \$7000 @ \$6.95	_____
	DISKETTES WITH DOS-68 AT \$D000 @ \$9.95	_____
	SE-1 TEXT EDITING SYSTEM @ \$29.00 (DISKETTE)	_____
	SA-1 MNEMONIC ASSEMBLER @ \$29.00 (DISKETTE)	_____
	SE-1/SA-1 EDITOR/ASSEMBLER COMBINATION @ \$53.00	_____
	TD-1 TRACE-DISASSEMBLER @ \$19.95 (ADD \$5.95 FOR DISC)	_____
	SG-1 SOURCE GENERATOR @ \$24.95 (ADD \$5.95 FOR DISC)	_____
	BASIC COMPILER @ \$325.00	_____
	BASIC COMPILER MANUAL @ \$10.00	_____
	CALIFORNIA RESIDENTS 6% SALES TAX	_____
	TOTAL	_____

_____ CHECK ENCLOSED _____ BILL MY MASTERCARD CARD

_____ BILL MY VISA CARD

CARD NUMBER _____ EXPIRATION DATE _____

SIGNATURE _____

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

BASIC PATCHES

LOAD SWTPC VERSION 2.0 BASIC INTO YOUR COMPUTER FROM CASSETTE TAPE. EXAMINE MEMORY LOCATION 014E. IF YOUR VERSION OF BASIC IS THE SAME AS OURS, IT SHOULD CONTAIN 1E AND LOCATION 014F SHOULD CONTAIN AF.

TYPE "GET,BASIC.PAT". SAVE THE NEW FILE CREATED IN MEMORY TO DISC BY TYPING SAVE,BASIC,100,~~FECA~~,100. OF COURSE, YOU MAY USE A DIFFERENT FILE NAME THAN BASIC IF YOU WISH. YOU MAY WISH TO ADD ",\$" TO THE SAVE INFORMATION GIVEN. THIS WAY YOU MAY RELOAD BASIC BY JUST TYPING THE COMMAND "BASIC" INSTEAD OF "RUN,BASIC".

THE PATCHES ALLOW YOU TO LOAD AND SAVE PROGRAMS WRITTEN IN BASIC. IF YOU TYPE "LOAD" OR "SAVE" FOLLOWED BY A PORT NUMBER, THE PROGRAM WILL WORK AS IT DID PREVIOUSLY. IF YOU TYPE "SAVE OR "LOAD" FOLLOWED BY A CARRIAGE RETURN, THE SYSTEM WILL ASK FOR A FILENAME AND THEN SAVE OR LOAD TO OR FROM THAT FILE. NOTE THAT THE FILES CREATED ARE NOT COMPATIBLE WITH THE FILES CREATED BY THE CO-RESIDENT EDITOR AND ASSEMBLER. YOU MAY NOT LOAD A BASIC FILE INTO THE EDITOR FOR EDITING. THIS IS A FUNCTION OF THE ORIGINAL PROGRAMS - NOT THE PATCHES.

CORES PATCHES

LOAD THE SWTPC CO-RESIDENT EDITOR AND ASSEMBLER FROM CASSETTE TAPE. OUR VERSION CONTAINED A 1B7C IN LOCATIONS 00FE AND 00FF. TYPE "GET, CORES.PAT". NOW TYPE "SAVE,CORES,FE,1D7A,100. TO RUN, TYPE "RUN, CORES". CORES WILL OPERATE IN THE SAME FASHION AS BEFORE EXCEPT THAT WHEN "SAVE" OR "LOAD" IS TYPED FOLLOWED BY A CARRIAGE RETURN, THE SYSTEM WILL PROMPT WITH "D OR OTHER?". IF YOU WISH TO SAVE OR LOAD FROM THE DISC, TYPE "D". ANY OTHER CHARACTER WILL CAUSE THE UNIT TO SAVE OR LOAD FROM CASSETTE. IF YOU TYPE "D", THE SYSTEM WILL ASK FOR A FILENAME AND THEN SAVE OR LOAD TO THAT FILE.

WHEN RUNNING THE ASSEMBLER, THE PROGRAM WILL ASK FOR AN OUTPUT FILENAME AND WILL CREATE A FILE ON DISC IN HEX FORMAT. THE SOURCE PROGRAM MUST CONTAIN AN "OPT O" STATEMENT OR AN OUTPUT FILE WILL NOT BE CREATED. WHEN YOU WANT TO LOAD THAT FILE INTO MEMORY, TYPE "GETH,FILENAME". THEN EXIT TO MIKBUG AND GO TO THE BEGINNING OF THE PROGRAM IN THE NORMAL MIKBUG MANNER. REMEMBER, THE FILE CREATED BY THE ASSEMBLER IS IN HEX FORMAT AND TAKES TWICE THE NORMAL AMOUNT OF DISC SPACE. IF THE ASSEMBLED PROGRAM IS A FINISHED PROGRAM THAT YOU EXPECT TO KEEP AND USE OFTEN, YOU MAY WISH TO LOAD IT INTO MEMORY WITH THE GETH COMMAND AND THEN SAVE IT IN BINARY FORM USING THE SAVE COMMAND.

DFM68 FUNCTION CODES

CODE NUMBER	CODE NAME	MEANING
0	QFREE	REPORT AMOUNT OF FREE SPACE ON A DISC
1	QSO4W	OPEN A FILE FOR WRITE (CREATE A FILE)
2	QSWRIT	WRITE DATA TO A FILE
3	QSWC	CLOSE A FILE OPEN FOR WRITE
4	QSO4R	OPEN A FILE FOR READING
5	QSREAD	READ FROM A FILE
6	QSRC	CLOSE A READ FILE
7	QDEL	DELETE A FILE
8	QREN	RENAME A FILE
9	QAPP	APPEND TWO FILES
10	QDIRI	OPEN A DISC DIRECTORY
11	QDIRT	RETRIEVE A FILE NAME FROM THE DIRECTORY
12-17		(RESERVED)
18	QSSR	SINGLE SECTOR READ
19	QSSW	SINGLE SECTOR WRITE

DFM ERROR CODES

ERROR NUMBER	ERROR NAME	MEANING
\$01	EIFC	INVALID DFM FUNCTION CODE
\$02	EFE	FILE EXISTS
\$03	EFIB	MASTER FILE DIRECTORY ERROR
\$04	EFB	FILE IS IN USE
\$05	ENSF	NO SUCH FILE EXISTS
\$06	EEOF	END OF FILE
\$07	EDF	DISC FULL
\$08	EIF	INVALID FILE CONTROL BLOCK (FCB) ADDRESS
\$09	EIFN	ILLEGAL FILE NAME
\$0A	EFS	FILE STATUS ERROR
\$0B	EITS	INVALID TRACK OR SECTOR NUMBES!GOR SINGLE SECTOR I/O
\$0C	EIUN	INVALID UNIT NUMBER (ONLY 0,1, AND 2 ALLOWED)
\$0D		(UNUSED)
\$0E	EDR	DISC READ ERROR
\$0F	EDW	DISC WRITE ERROR

MONITOR CLOSE COMMAND ERROR CODES

ERROR NUMBER	MEANING
01XX	ILLEGAL FILE TYPE. IE. FILE STATUS IS NOT READ OR WITE XX = FILE STATUS CODE
02XX	(UNUSED)
03XX	FILE CLOSING ERROR XX = CLOSING ERROR CODE (SEE DFM ERROR CODES)
04XX	READ ERROR ON DISC CONTROL BLOCK XX = ERROR STATUS FROM THE WD 1771-1 CHIP
05XX	WRITE ERROR ON DISC CONTROL BLOCK XX = ERROR STATUS FROM THE WD 1771-1 CHIP

USING SD BASIC

We suggest that before you do anything, make a backup copy of the diskette and put it aside to be used in case of any accidental destruction.

When using SD BASIC, we suggest that you put all the DOS facilities, SDBAS, SDASM, and SDRUN on drive 1, and that drive 0 contain only a diskette with your BASIC program on it (otherwise, you won't have any room; SD BASIC is not recommended for single drive systems).

To use the BASIC, one must first generate the source file using EDIT (or any similar editor which does not write line numbers to the output file nor which outputs extraneous control characters). Don't confuse the editor's line numbers with BASIC line numbers; the editor's line numbers disappear when you exit from the editor. Let's say you built a file and called it TEST.B3.

To run the compiler, type:

```
RUN 1:SDBAS
```

When the compiler asks:

```
INPUT FILE =
```

respond with:

```
TEST.B3
```

All the DOS68 operating conventions for line input are valid here, including control O for character deletion and control X for line deletion. An I/O ERROR 000053 means "no such file". All the error numbers can be found in the error summary in the BASIC manual. Do a LIST to determine the correct file name and repeat the sequence. In response to:

```
OUTPUT FILE =
```

use a temporary file (we use JUNK). The compiler will delete any old version of junk and create a new version. The compiler's output goes into the output file. During the compilation process, all compile-time errors will be printed on the console. When the compiler is finished it will say:

```
COMPILATION COMPLETE
```

If there were any compilation errors, go back and re-edit the BASIC program, and then re-compile.

If there were no compilation errors, then you are ready for next step. Here you need to assemble the output of the compiler. To run the assembler, type:

```
RUN 1:SDASM
```