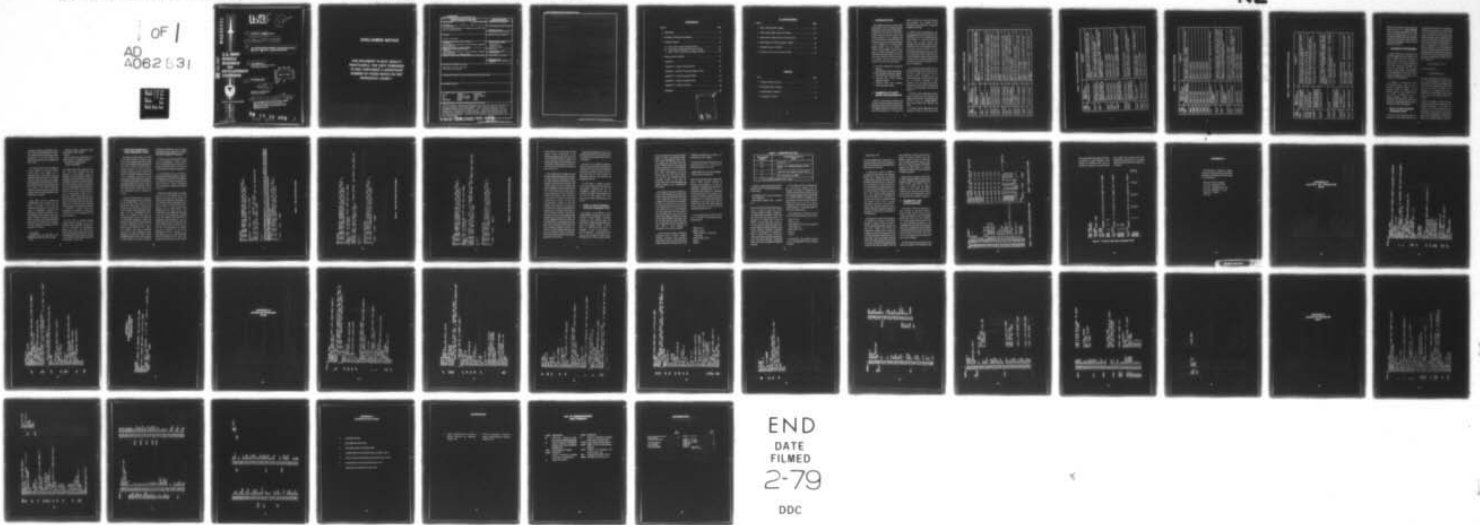


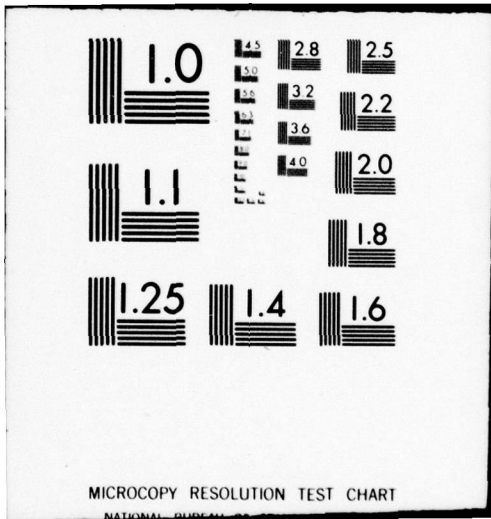
AD-A062 831 ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
SOFTWARE DEVELOPMENT FOR INTERFACING AN HP-21 MX WITH A TEKTRON--ETC(U)
SEP 78 J R MITCHELL

UNCLASSIFIED DRDMI-T-78-102

NL

1 of 1
AD
A062 831





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD A062831



LEVEL II

12

9 TECHNICAL REPORT T-78-102

14 DRDMI-T-78-102

6 SOFTWARE DEVELOPMENT FOR INTERFACING AN HP-21 MX WITH A TEKTRONIX 4051

10 Jerrel R. Mitchell
Guidance and Control Directorate
Technology Laboratory

DDC
RECEIVED
DEC 29 1978
F

11 29 September 1978

12 45 p.

Approved for Public Release;
Distribution Unlimited

16 LW 362303A214 17 02

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

78 12 26 089
393 427

mt

DDC FILE COPY

U.S. ARMY
MISSILE
RESEARCH
AND
DEVELOPMENT
COMMAND



Redstone Arsenal, Alabama 35809

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER T-78-102	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Development for Interfacing an HP-21 MX with a Tektronix 4051		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerrel R. Mitchell		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander US Army Missile Research and Development Command ATTN: DRDMI-TG Redstone Arsenal, Alabama 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1W362303A214 632303.11.21402
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Research and Development Command ATTN: DRDMI-TI Redstone Arsenal, Alabama 35809		12. REPORT DATE 29 September 1978
		13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) RTE-M FORTRAN ASSEMBLER HP CONTROL SYSTEM BINARY SYSTEM RTE GRAPHIC SYSTEM XFER TEKTRONIX RADAR TCOM		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In this report, software developments for fully utilizing the marriage between a Tektronix 4051 Graphic System, an HP-21 MX minicomputer, and an HP-9885M flexible disc are presented. First, software necessary for this combination is presented in several tables. Then three programs that were specifically tailored for transferring data between the three devices are presented and discussed. The use of each program is illustrated with one or more examples.		

DD FORM 1 JAN 73 1473

EDITION OF NOV 65 IS OBSOLETE

78-12-26-089

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

Section	Page
1. Introduction	3
2. Summary of Software Developments	3
3. Transfer Programs	8
A. Disc to Disc Transfer Program (XFER)	8
B. Disc-Tape Communications Program (TCOM)	10
C. Basic to Disc Communications Program (TKHP)	14
4. Summary and Conclusions	17
Appendix A	21
Appendix B—Listing of Program XFER	23
Appendix C—Example of Running Program XFER	27
Appendix D—Listing of Program TCOM	29
Appendix E—Listing of Program TKHP	39
Appendix F—Listing of File Types	45
References	47

ACCESSION For	
NTIS	<input checked="" type="checkbox"/>
DDC	<input type="checkbox"/>
DISPATCHING	<input type="checkbox"/>
REPRODUCTION OF THIS	
ORIGINAL	
A	23 DTC

ILLUSTRATIONS

Figure	Page
1 Disc to Tape Transfer Example	11
2 Tape to Disc Transfer When File Existed	12
3 Tape to Disc Transfer When File Did Not Exist	13
4 Basic Program for Showing Utility of TKHP	18
5 Example Execution of TKHP	18
6 Contents of Files Used to Illustrate TKHP	19

TABLES

Table	Page
1 Symbolic Program and Files	4
2 Relocatable Binary Programs	5
3 Absolute Binary Programs	6
4 Commands for TKHP	16

1. INTRODUCTION

The standard console for an HP-21 MX minicomputer with an HP-9885M flexible (floppy) disc is an HP-2644/45 data station. In fiscal year 1977 an HP-21 MX with a flexible disc was purchased by the Control Systems Branch of the Guidance and Control Directorate of MIRADCOM. The HP data station was not purchased because of the availability of a Tektronix 4051 Graphic System that can emulate a Tektronix 4012 computer terminal. In this report, software developments for fully utilizing the marriage between the HP-21 MX and the Tektronix 4051 are presented. It is assumed that the reader has some familiarity with the following manuals:

- RTE-M Programmer's Reference Manual
- HP FORTRAN Reference Manual
- RTE Assembler Reference Manual
- RTE-M Editor Reference Manual
- Tektronix 4051 Graphic System Reference Manual
- Tektronix 4051 Data Communications Interface Manual

2. SUMMARY OF SOFTWARE DEVELOPMENTS

Tables 1, 2, and 3 present a summary of all symbolic and binary programs generated by this author, along with some frequently used HP supplied programs and libraries. Each line of the tables gives a file name, the

label(s) of disc(s) on which the file resides, and comments. The comments provide information that should aid in using the programs in the files.

In Table 3, there are references to two operating systems, SYSGEN and SYSGN2. SYSGEN is a Type M-I operating system, and SYSGN2 is a Type M-II operating system. Both of these systems were generated for use with the Tektronix 4051, operating in the Terminal Mode [1]. Before entering Terminal Mode, several environmental parameters that control the Communication Interface of the 4051 must be set. Setting these parameters and entering Terminal Mode is easily accomplished by executing, on the 4051, the BASIC program given in Appendix A.

After entering Terminal Mode, either of the above operating systems can be "booted-in" from a floppy disc. There are two different boot-in procedures, one for systems beginning in track 0, sector 2, and one for systems beginning in other locations. (If no track and sector numbers are given on the disc label for an operating system, it can be assumed to start in track 0, sector 2.)

To boot-in systems beginning in track 0, sector 2, bits zero, six, nine, fourteen, and fifteen of the S-register to one (the other bits should be zero), i.e., 1100001001000001. Then, press STORE, IBL, PRESET, and RUN.

To boot-in systems beginning in other locations, set bits six, nine, fourteen, and

TABLE 1. SYMBOLIC PROGRAMS AND FILES

FILE NAME	DISC(S) ON WHICH LOCATED	COMMENTS
\$AB	FORTRAN and Program Development	Assembler routine for returning contents of A and B registers to a FORTRAN Program (See Program Reference Manual Pages 4-5).
ECHO	Reduced System Generation	Answer file for generating a TYPE II System.
ECHO1	Reduced System Generation	Echo file resulting from generation of an M-II System can be used as answer file in future generations. Is commented.
\$HELP	Assembler Development	Assembler routine for satisfying certain calls by & TKHP.
\$INSUB	Assembler Development	Assembler routine called by subroutine TRDS in TCOM.
\$IO13	Assembler Development	Assembler routine called by subroutines \$HELP, \$INSUB and \$OUTSB.
\$OUTSB	Assembler Development	Assembler routine called by subroutine TRTP in TCOM.
SCR1	FORTRAN Development	Scratch file used in compilation of FORTRAN Programs.
SNAP1	FORTRAN Assembler and Reduced Generation	Snap file for Type I Systems. Needed in relocating programs for Type I Systems.
SNAP2	M-II System	Same as SNAP1 except for Type II System.
\$TCOM	FORTRAN Development	FORTRAN main program and subroutines of TCOM (See Section 3. b. for usage).
\$TKHP	FORTRAN Development	FORTRAN main program and subroutine of TKHP (See Section 3. c. for usage.).
\$XFER	FORTRAN Development	FORTRAN source program of XFER. (See Section 3.A, for usage.)

TABLE 2. RELOCATABLE BINARY PROGRAMS

FILE NAME	DISC(S) ON WHICH LOCATED	COMMENTS
%CAT	Assembler Development	Program that reads up to 128 characters from 4051. The read is terminated by <u>D</u> and the characters are returned to the 4051.
%F4.N	FORTRAN Development and Program Development	FORTRAN library supplied by HP. First library to be searched when relocating FORTRAN programs.
%MPP	FORTRAN Development Program Development and All Generation (Reduced, too)	File management package. Should be searched when any file management routine is being used. In FORTRAN, second library to be searched.
%HELP	Assembler Development	Relocatable version of \$HELP2.
%INSUB	Assembler Development	Relocatable version of \$INSUB.
%I013	Assembler Development	Relocatable version of \$I013.
%MSYLB	FORTRAN Development, Program Development and All Generation (Reduced, too)	System library. In FORTRAN relocation is second library to be searched if file management package is not needed.
%OUTSB	Assembler Development	Relocatable version of \$OUTSB.
%RLIB1	FORTRAN Development, Program Development and All Generation (Reduced, too)	Floating Point Library, Part 1. In FORTRAN relocation, is third library to be searched if FMP is not needed.
%RLIB2	(Same as %RLIB1)	Floating Point Library, Part 2. In FORTRAN relocation is last library to be searched.
%TKHP	FORTRAN Development	Relocatable version of \$TKHP2
%TCOM	FORTRAN Development	Relocatable version of \$TCOM.

TABLE 3. ABSOLUTE BINARY PROGRAMS

FILE NAME	DISC(S) ON WHICH LOCATED	COMMENTS
ASM	Assembler Development and Program Development	Main Program of Assembler. Relocated for Type I Systems. Automatically loads segments as needed.
ASMB1	(Same as ASM)	Segment of Assembler
ASMB2	(Same as ASM)	Segment of Assembler
ASMB3	(Same as ASM)	Segment of Assembler
ASMB4	(Same as ASM)	Segment of Assembler
ASMBD	(Same as ASM)	Segment of Assembler
ASMBX	(Same as ASM)	Segment of Assembler
CAT	FORTRAN Development and Assembler Development	Absolute version of %CAT. Relocated for Type I Systems.
DSKET	First Generation	Program for formatting discs. Relocated for Type I Systems. (See OVR33 Programming Manual for instructions on usage.)
EDIT	FORTRAN Development and Assembler Development	Editor to be run under control of Type I Systems located on FORTRAN and Assembler Development Discs.
EDIT2	M-II Development	Editor to be run under Type II System.
FTN	FORTRAN Development	Main Program of HP FORTRAN Compiler. Relocated for Type I Systems. Automatically loads segments as needed.
FTN1	(Same as FTN)	Segment of FORTRAN Compiler.
FTN2	(Same as FTN)	Segment of FORTRAN Compiler.

TABLE 3. ABSOLUTE BINARY PROGRAMS (Continued)

FILE NAME	DISC(S) ON WHICH LOCATED	COMMENTS
RELF	FORTRAN Development and M-II Development	Command file that can be used to place relocation commands which the relocating loader can transfer to.
RTL2	M-II Development	Relocating loader for the Type II System.
RTLD2 RTMGN	First Generation and Reduced Generation	Relocated system generation program for use with Type I systems. Can be used to generate other systems with 4051.
RTMLD	FORTRAN Development, Assembler Development and Reduced Generation	Relocating loader for use with the Type I Systems.
SGPRP	Program Development	Segmented Program Preparation Program. Runs under Type I Systems. After relocating segmented programs, this program should be run. (See Pages 7-33 of Program Reference Manual.)
SYSCP	SYSCP	Type I System with single executable program. (See section 3. A. of this report for details).
SYSGEN	FORTRAN Development, Assembler Development and Reduced Generation	Type I Operating Systems.
SYSGN2	M-II Development	Type II Operating Systems.
TCOM	FORTRAN Development	Absolute version of \$TCDM. Relocated for execution under Type I Systems.
TKHP	FORTRAN Development	Absolute version of \$TKHP. Relocated for execution under Type I System.

fifteen of the S-register to one (the other bits should be zero), i.e., 1100001001000000. Look up the octal equivalents of the track and sector numbers (see p. F-2 of the RTE-M System Generation Reference Manual). Add the track and sector octal equivalents and store in the B-register. Then, press IBL, PRESET, and RUN.

3. TRANSFER PROGRAMS

A major problem that resulted from the available equipment was the inability to be able to transfer files from one disc to another. Also, there was no means to backup files, i.e., by storing on alternate mass media. These two problems were solved by developing the necessary computer coding to allow for file transfers between different discs and for file transfers between the magnetic tape cassette unit on the 4051 and the flexible discs.

In addition to solving the above-mentioned problems, a goal was set to develop the necessary software to allow programs written in BASIC on the Tektronix 4051 to use the HP-9885M for mass storage. To achieve this goal, a monitor program (written partly in FORTRAN and partly in assembler) was written for interfacing BASIC program on the 4051 with files on the 9885M.

A. DISC TO DISC TRANSFER PROGRAM (XFER)

The disc to disc transfer program was developed to aid in transferring files from

one flexible disc to another. The underlying principle of the program is to transfer data from a file on one flexible disc to the memory of the HP-21 MX, mount a second flexible disc and transfer the data from the memory to a file on the second flexible disc. The computer coding to accomplish this is given in Appendix B. This program uses several routines from the File Management Package. All these routines are described in the Programmer's Reference Manual except DCMC [2]. This is the mount/dismount routine and is FORTRAN callable. The call for a dismount is

```
CALL DCMC (1, -lu, 0)
```

and for a mount it is

```
CALL DCMC (0, lu, 0)
```

where lu is the logical unit number of the disc. The last argument is the last track number of the disc; if zero, it defaults to that of the disc. If this argument is omitted, the disc will not mount. Actually, for a dismount the first argument needs only to differ from zero, and the disc number can be used in place of the negative logical unit number.

Because of limited computer memory there is a limit on the size of a file that can be transferred. Because of this limit two versions of the copying programs were developed. The first version can be used to transfer files whose lengths are less than 105 blocks (this is approximate). The absolute binary version of this program is XFER. It

is located on both the FORTRAN DEV. DISC and the ASSEMBLER DEV. DISC. The symbolic and relocatable binary reside, respectively, in the files \$XFER and %XFER and are located on the FORTRAN DEV. DISC.

The second version has been included as the lone program in a minimum TYPE I system. Files with lengths up to approximately 150 blocks can be transferred with this version. The TYPE I system in which this version is an integral part is located on the SYSCPY disc. To run this version, "boot-up" on this disc and type RU, XFER or ON, XFER. When the transfer is complete, another system must be booted-in before another program can be loaded and/or executed.

Using either version is simple and straightforward. But remember, *do not mount or dismount a disc until the programs give permission*. If you do, you might be unpleasantly surprised. (For example, you might find a directory on a disc changed.) As an example on the use of XFER, consider transferring a file called TEST on one disc to a file called TESTX on another disc. Assuming that the program has been loaded, the dialogue with the computer is as follows:

```
*ON, XFER
DISMOUNT & MOUNT (IF
DESIRED) AND GIVE FILE NAME
TEST
```

```
REMOVE DISK & MOUNT NEW
DISK & GIVE FILE NAME
TESTX
THE FILE TESTX IS CREATED. IT IS
TYPE 4 AND IS -1 BLOCKS LONG.
TRANSFER COMPLETED
XFER: STOP 0000
```

The italicized parts were supplied by the user. (The actual dialogue with the computer for this example is given in Appendix C.) The user has the freedom of loading and running XFER from one disc, mounting a second disc and transferring the file TEST to memory, and then mounting a third disc and transferring the contents of memory to file TESTX. In the case above, TESTX was not found on the disc; thus, a file was automatically created. The "-1 BLOCKS LONG" means that the exact number of blocks needed by the file will be used. If TESTX had existed, the file space defined for it would have been used and extents would have been added if needed. If it is desired to transfer several files sequentially, the program can simply be rerun (however, do not change discs until given permission).

If errors occur in opening, closing, reading, writing, creating, etc., files, the program will automatically terminate and print the appropriate file management negative error codes. The codes can be interpreted by referring to Section IX of the Programmer's Reference Manual. A typical error is "FMP-6," which usually means a file is not found or a disc is full.

B. DISC-TAPE COMMUNICATIONS PROGRAM (TCOM)

The purpose of this program is to allow file transfers between a magnetic tape cassette on the Tektronix 4051 and a flexible disc on the HP-9885M. This program can be used in lieu of XFER, or it can be used to transfer programs to tape for backup purposes. If used in lieu of XFER, the transfer will be slower because the transfer rate is basically controlled by the communications link between the computer and the 4051 (in particular, each character is transmitted serially as ten bits at a rate of 2400 baud). However, binary files of more than 750 blocks and ASCII files of more than 1500 blocks can be transferred by TCOM.

A listing of TCOM is given in Appendix D. The absolute binary version is located in the file TCOM that is located on the FORTRAN DEV. DISC. TCOM can be loaded and run under the control of the TYPE I systems on the FORTRAN DEV. DISC or on the ASSEMBLER DEV. DISC. The symbolic version and the relocatable binary versions of TCOM and its FORTRAN subroutines, TRDS and TRTP, are located in the files \$TCOM and %TCOM, respectively, which are also located on the FORTRAN DEV. DISC. The symbolic and binary relocatable versions of the two assembler routines required by TCOM are located on the ASSEMBLER DEV. DISC. The symbolics of these routines are stored in the files

\$OUTSB and \$INSUB, and the binary relocatables of these programs are stored in %OUTSB and %INSUB, respectively.

In order to use the Tektronix 4051 in the Tape Communications Mode, certain environmental parameters must be set [1]. A listing of a BASIC program for setting the required parameters is given in Appendix A.

As an illustration, the use of TCOM is demonstrated by three examples. Listings of the dialogue with TCOM and the dialogue with the BASIC interpreter of the 4051 are shown in *Figures 1, 2, and 3*. All lines less than ten characters in length were supplied by the user.

In the first example, the goal is to transfer the file CAT to file number four on the magnetic tape. First, the user initiates the execution of TCOM (the assumption here is that TCOM was previously loaded). Next, the user types +1 and presses return to indicate a disc to tape transfer. After mounting the disc with the file to be transferred, the user supplies the file name (in this case, CAT) along with a carriage return. The computer prints three lines of information and instructions. The user selects a tape file by pressing the shift key and the FIND FILE key; then he types 4. The tape is then positioned to file number four. (It is assumed that the DATA COMMUNICATIONS OVERLAY has been placed over the USER DEFINED KEYS on the 4051.) He then types -1 (no carriage return) and presses the DATA


```

*RU, TCOM
IF YOU ARE SENDING INFORMATION TO THE TAPE TYPE +1
IF YOU ARE SENDING INFORMATION FROM THE TAPE TYPE -1
IF YOU WANT TO TERMINATE THIS SESSION TYPE 0
-1
MOUNT DISC TO RECEIVE; GIVE TYPE AND FILE NAME
FORMAT IS (11,1X,3A2)
? CAT2
PREPARE TAPE: WHEN READY TYPE -1, PRESS RETURN AND PRESS DATA SEND KEY
TERMINAL MUST BE IN PROMPT MODE

File 4
-1
TRANSFER COMPLETED
IF YOU ARE SENDING INFORMATION TO THE TAPE TYPE +1
IF YOU ARE SENDING INFORMATION FROM THE TAPE TYPE -1
IF YOU WANT TO TERMINATE THIS SESSION TYPE 0
0
TCOM : STOP 0000

```

Figure 2. Tape to disc transfer when file existed.

```
*RU, TCOM
IF YOU ARE SENDING INFORMATION TO THE TAPE TYPE +1
IF YOU ARE SENDING INFORMATION FROM THE TAPE TYPE -1
IF YOU WANT TO TERMINATE THIS SESSION TYPE 0
-1
MOUNT DISC TO RECEIVE; GIVE TYPE AND FILE NAME
FORMAT IS (11, 1X, 3A2)
? CAT3
FILE NOT FOUND. FILE, CAT3 IS CREATED AS A TYPE: 7
PREPARE TAPE: WHEN READY TYPE -1, PRESS RETURN AND PRESS DATA SEND KEY
TERMINAL MUST BE IN PROMPT MODE
```

```
File 4
-1
```

```
TRANSFER COMPLETED
IF YOU ARE SENDING INFORMATION TO THE TAPE TYPE +1
IF YOU ARE SENDING INFORMATION FROM THE TAPE TYPE -1
0
TCOM : STOP 0000
```

Figure 3. Tape to disc transfer when file did not exist.

RECEIVE key. At this point the transfer begins. Some overprinting of the information going to tape will occur on the screen. When the transfer is complete, the program will return to the initial point, and the transfer of other files can be initiated or execution can be terminated. The latter was done in this case.

The examples shown in *Figures 2 and 3* illustrate transfer from tape files to disc files. It is easily seen that the dialogue is similar. One major difference between disc to tape and tape to disc transfers is that tape to disc transfers require the user to supply a file type APPENDIX F and a file name. When a file is transferred from disc to tape, the user must remember the file type if he plans to transfer the file back to a disc. Type 4 files are assumed to be binary. Since the data communications interface of the 4051 assumes ASCII data only, the transfer of disc binary files to tape is accomplished by coding each 16 bit binary word into four ASCII characters. (Because of this, binary files require twice as much magnetic tape storage as disc storage.) Thus, when files that are designated as binary are transferred from tape to disc, a decoding process takes place. On the other hand, ASCII files (type 4) are transferred unaltered. Similarly, it is desired to transfer magnetic tapes files that have been generated with the Tektronix BASIC interpreter to disc files; they must be generated as ASCII files.

The goal of the example in *Figure 2* is to transfer the tape file 4 to the disc file CAT2. It is known that the information in file 4 is coded binary.

The goal of the example presented in *Figure 3* is to transfer file 4 on the tape to the type 4 file, CAT3, on the disc. The difference between the examples of *Figures 2 and 3* is that CAT3 did not exist; however, as indicated in *Figure 3*, it was created as the appropriate type file.

As with XFER, if TCOM encounters errors in accessing disc files, the program is automatically terminated and the file management negative error codes are displayed. In addition, if the magnetic tape unit on the 4051 detects errors, the appropriate message will be printed on the screen.

C. BASIC TO DISC COMMUNICATIONS PROGRAM (TKHP)

The purpose of the BASIC to Disc Communications Program is to provide an interface between BASIC programs executing on the Tektronix 4051 and flexible discs residing in the HP-9885M disc drive. The program, TKHP, is written partly in FORTRAN and partly in assembler. A list of the program is given in Appendix E. The absolute binary version of the program is located on the FORTRAN DEV. DISC in the file TKHP.

It can be run under the control of the Type I systems on either the FORTRAN DEV. DISC or the ASSEMBLER DEV. DISC. The symbolic and relocatable binary files for the FORTRAN part of the program are located, respectively, in the files \$TKHP and %TKHP, located on the FORTRAN DEV. DISC. The assembler part of the program has three entry points, HELP, INP, and OUT. The symbolic and binary relocatable versions of this part are located respectively, in the files \$HELP and %HELP located on the ASSEMBLER DEV. DISC.

The principle of operation of TKHP is as follows. After starting the execution of the program, the user is asked to give the names of up to eight files that are to be made available for I/O with the Tektronix BASIC Interpreter. (At this point the user can also mount a different disc.) He can use any or all of the eight; however, he must remember which file number goes with which name. If any of the files do not exist, they will be created with lengths of 20 blocks (extents will be added if needed). The program tells the user which files are created, which files are not being used, and how many files are open. Then a "ready" indicator is transmitted to the screen of the 4051.

At this point the program is waiting for instructions from a BASIC program running on the 4051. These instructions must come in the form of ASCII character strings, followed by a carriage return. Table

4 defines the instruction set, where n is a numerical from one to eight.

For a BASIC program to input a record of ASCII data from file ? on the disc, the following sequence can be used:

```
PRINT @40: "I?" (? Any file 1 through 8)
INPUT @ 40: (Variable list).
```

(Note: On input and output, variable lists cannot contain matrices; elements of matrices are acceptable.) To avoid the possibility of losing data there should be no statements separating these two. It is assumed that the variable list is compatible with the record that is forthcoming, i.e., the number of variables is equal to the number of numbers, etc. (A number has the normal definition for free field input with the BASIC INPUT statement.) By knowing the format of the data, the input can also be made under format control.

For the reading of matrix values (A) from disc to terminal, the following sequence should be used:

```
FOR I=1 to P
FOR J=1 to Q
PRINT @40: "I?" (? Any file 1
through 8)
INPUT @ 40: A (I,J)
NEXT J
NEXT I
```

TABLE 4. COMMANDS FOR TKHP

CHARACTER STRING	INTERPRETATION BY TKHP
Rn	Rewind file n . *
On	Output the following logical record to N.
In	Send the next logical record from N .
E	End execution of TKHP.
* If n is omitted, a default to file 1 occurs.	

In order to output a record from a BASIC program to file "?," the following sequence must be used:

```
PRINT@40: "O?" (? any file 1 through 8)
```

```
INPUT@40: P$
```

```
PRINT@40: USING XXX: (variable list).
```

The first statements tell TKHP to prepare to receive a record of data. The second statement forces the BASIC program to wait until TKHP is prepared to receive the data (P\$ can be any target variable. The character P is actually what is read in TKHP). The third statement outputs the record to the disc. In this statement XXX is used to denote the statement number of the format statement. Although a formatted output is not required, it is advisable in order to "pack" the data on the disc. Unformatted output can waste valuable disc space with blank characters. Outputting data to any of the other eight files should be obvious.

For each block of data to be read on the disc, the previous sequence must be repeated. Any number of variables can be printed from the terminal to the disc as long as the image statement reflects the number [e.g., IMAGE 4(3D)] of variables in the string (four in this case) and the variable list defines them (e.g., P, Q, R, S) with a carriage return after the last. Inherent in the TKHP program is a maximum number of 128 in any variable string.

For transferring a P by Q matrix A to the disc, the following sequence should be used:

```
FOR I=1 to P
FOR J=1 to Q
PRINT@40: "O?" (? any file 1 through 8)
INPUT@40: P$
PRINT@40: A(I,J)
NEXT J
NEXT I
```

Rewinding file ? using a BASIC program can be accomplished with the following statement:

PRINT @ 40: "R?".

If a file is being used for a scratch file, i.e., input and output in the same program, it should be rewound before inputting from it after output has occurred.

To illustrate the utility of TKHP in providing the link between the Tektronix BASIC and a flexible disc, the BASIC program shown in *Figure 4* was written. The program reads values for X, Y, and Z from file 1 and values for W and U from file 2. Five computations are made using X, Y, Z, W, and U, producing values for P, Q, R, V, and S. Then P, Q, R, V, and S are printed on file 3. This is repeated while reading two records from files 1 and 2. Then, file 1 is rewound, and the above is repeated except that records 1 and 2 of file 1 are used, respectively, with records 3 and 4 of file 2. A total of four records is printed on file 3. Finally, the BASIC program terminates the execution of TKHP and returns the 4051 to Terminal Mode.

The sequence of events that occur prior to and during the execution of the BASIC program in *Figure 4* is shown in *Figure 5*. TKHP is loaded into memory and run. The files TEST, TESTX, and TESTY are assigned the numbers one, two, and three, respectively. One or more blanks are entered for the other five files to indicate they are not being used. Then, the computer indicates that TESTY was not found on the disc; thus, it is created. The files not being used and the number of files open are printed. Then, the

ready indication is received. The user then presses the return to BASIC key. The BASIC program in *Figure 4*, which had been previously loaded into memory of the 4051, is run. Upon completion of the BASIC program, the 4051 returns to Terminal Mode and the indication of the completion of TKHP is printed.

Figure 6 shows the session with the HP-21 MX for aborting TKHP, loading the file manager (FMGR) and inspecting the files TEST, TESTX, and TESTY. Prior to execution of TKHP and the BASIC program, the files TEST and TESTX existed with the contents shown. However, the file TESTY was created and its contents were generated by the BASIC program.

4. SUMMARY AND CONCLUSIONS

In this report, software developments for fully utilizing the marriage between a Tektronix 4051 Graphic System, an HP-21 MX minicomputer, and an HP-9885M flexible disc drive have been presented. First, tables summarizing the software that has been specifically developed for this combination by this author were presented. Then, three programs that were developed as aids in data transfer between the devices were presented and discussed. The use of each of the programs was illustrated with one or more examples.

Software developments presented in this report have added to the flexibility of the

```

80 CALL "CMSET"
100 A$="I"
110 B$="O"
120 E$="E"
130 R$="R"
150 I1=1
160 I2=2
165 I3=3
165 FOR J=1 TO 2
170 FOR I=1 TO 2
180 PRINT @40:A$,I1.
190 INPUT @40:Y,Y;Z
192 PRINT @40:A$,I2
194 INPUT @40:W,U
200 P=X+Y
210 Q=X*Y+Z+W
220 R=X+Y+W+Z+U
230 U=2*X+3*W+4*U
240 S=X-Y+Z-W+U
250 PRINT @40:B$,I3
260 INPUT @40:P$
270 PRINT @40: USING 280:P,R,Q,S,U
280 IMAGE 5(100)
290 NEXT I
292 PRINT @40:R$,I1
298 NEXT J
299 PRINT @40:E$
300 CALL "TERMIN"
310 END

```

Figure 4. BASIC program for showing utility of TKHP.

```

*RU, TKHP2
MOUNT DISC TO SEND AND/OR RECEIVE
GIVE NAME OF NO. 1 FILE
TEST
GIVE NAME OF NO. 2 FILE
TESTX
GIVE NAME OF NO. 3 FILE
TESTY
GIVE NAME OF NO. 4 FILE
GIVE NAME OF NO. 5 FILE
GIVE NAME OF NO. 6 FILE
GIVE NAME OF NO. 7 FILE
GIVE NAME OF NO. 8 FILE
FILE TESTY NOT FOUND. IT IS CREATED
THERE IS NO NO. 4 FILE
THERE IS NO NO. 5 FILE
THERE IS NO NO. 6 FILE
THERE IS NO NO. 7 FILE
3 FILES ARE OPEN * * *
* * * READY * * *
RUN
TKHP2 : STOP 0000

```

Figure 5. Example execution of TKHP.

above combination of equipment. However, the addition of other equipment would enhance the flexibility even more. In particular, the addition of another flexible

disc, another 32K of memory, of an HP-2644/45 data station and of a line printer would more than double the capability of the system.

```

*OF, TKHP :8
TKHP, ABORTED

*LO, FMGR
APLDR: DONE- FMGR

*ON, FMGR
:LI, TEST
TEST T=00004 IS ON CR32760 USING 00001 BLKS R=0000
0001 1 4 2 5
0002 6 2 3

:LI, TESTX
TESTX T=00004 IS ON CR32760 USING 00001 BLKS R=0000
0001 8 10
0002 1 7
0003 2 15
0004 10 25

:LI, TESTY
TESTY T=00004 IS ON CR32760 USING 00020 BLKS R=0004
0001 5 28 44 4
0002 9 13 15 13
0003 5 27 14 15
0004 8 46 42 22
:
55
43
142

```

Figure 6. Contents of files used to illustrate TKHP.

APPENDIX A

The following is a listing of a BASIC program for setting certain environmental parameters and putting the Tektronix 4051 in Terminal Mode.

```
100 CALL "RATE", 2400,5,0
110 CALL "MARGIN", 0,0,0
120 CALL "TSTRIN",@ "J", "D"
130 CALL "PROMPT", 1,0,"R"
140 CALL "TERMN"
150 END
```



```

FTN,L,A
PROGRAM XFER
DIMENSION IDCB(144),IBUF(18000),NAME(3),LEN(999),ISIZE(2)
IDCB(10)=0
CALL DCMC(1,-2,0)
WRITE(1,4)
FORMAT("DISMOUNT & MOUNT (IF DESIRED) AND GIVE FILE NAME")
4 READ(1,5) (NAME(I),I=1,3)
5 FORMAT(3A2)
CALL DCMC(0,2,0)
CALL OPEN(IDCB,IERR,NAME,0,0,0,144)
12 IF(IERR)12,14
10 WRITE(1,10)IERR
FORMAT("ERROR: FMP ",I3)
STOP
14 I=1
I TYPE=IERR
ISIZE(2)=0
J=1
KTOT=0
15 CALL READF(IDCB,IERR,IBUF(I),128,LEN(J))
16 IF(LEN(J))20,16
I=LEN(J)+1+KTOT
22 IF(I TYPE-3)22,23
23 ISIZE(2)=LEN(J)
KTOT=KTOT+LEN(J)
J=J+1
IF(KTOT-18000)15,18
18 WRITE(1,19)
19 FORMAT("IBUF IS TOO SMALL")
STOP
CONTINUE
CALL CLOSE(IDCB)
J=J-1
20

```

```

25      ISIZE(1)=-1
        CALL DCMC(1,-2,0)
        WRITE(1,25)
        FORMAT('REMOVE DISK & MOUNT NEW DISK & GIVE FILE NAME')
        READ(1,5) (NAME(I),I=1,3)
        CALL DCMC(0,2,0)
        CALL OPEN(IDCIB,IERR,NAME,0,0,0,144)
        IF(IERR)31,40
31      WRITE(1,35)(NAME(I),I=1,3),ITYPE,ISIZE(1)
35      FORMAT('THE FILE ",3A2," IS CREATED.", " IT IS TYPE ",I2," AND "
1      " IS ",I3," BLOCKS LONG.")
        CALL CREAT(IDCIB,IERR,NAME,ISIZE,ITYPE,0,0,144)
        CONTINUE
        I=1
        DO 50 K=1,J
        CALL WRITF(IDCIB,IERR,IBUF(I),LEN(K))
        IF(IERR)43,45
43      WRITE(1,10) IERR
        STOP
        I=LEN(K)+1
        CONTINUE
50      CALL LOCF(IDCIB,IERR,IREF,IPB,IOFF,JSEC)
        ITRUN=JSEC/2-IPB-1
        CALL CLOSE(IDCIB,IERR,ITRUN)
        WRITE(1,55)
        FORMAT('TRANSFER COMPLETED')
55      STOP
        END
        END$
EOF

```

**APPENDIX C
EXAMPLE OF RUNNING
PROGRAM XFER**

```
#ON,XFER  
DISMOUNT & MOUNT (IF DESIRED) AND GIVE FILE NAME  
TEST  
REMOVE DISK & MOUNT NEW DISK & GIVE FILE NAME  
TESTX  
THE FILE TESTX IS CREATED. IT IS TYPE 4 AND IS -1 BLOCKS LONG.  
TRANSFER COMPLETED  
XFER : STOP 0000
```

**APPENDIX D
LISTING OF PROGRAM
TCOM**

```

14  *
15  *
16  *
17  *
18  *
19  *
20  *
21  *
22  *
23  *
24  *
25  *
26  *
27  *
28  *
29  *
30  *
31  *
32  *
33  *
34  *
35  *
36  *
37  *
38  *
39  *
40  *
41  *
42  *
43  *
44  *
45  *
46  *
47  *
48  *
49  *
50  *
51  *
52  *
53  *
54  *
55  *
56  *
57  *
58  *
59  *
60  *
61  *
62  *
63  *
64  *
65  *
66  *
67  *
68  *
69  *
70  *
71  *
72  *
73  *
74  *
75  *
76  *
77  *
78  *
79  *
80  *
81  *
82  *
83  *
84  *
85  *
86  *
87  *
88  *
89  *
90  *
91  *
92  *
93  *
94  *
95  *
96  *
97  *
98  *
99  *
100  *

```

```

FTN,L,T PROGRAM TCOM
DIMENSION IDCB(1296),IBUF1(128),IBUF2(258),NAME(3)
COMMON N2,IBUF2,KSKIP
WRITE(1,10)
FORMAT("IF YOU ARE SENDING INFORMATION TO THE TAPE TYPE +1"
1 /"IF YOU ARE SENDING INFORMATION FROM THE TAPE TYPE -1"
2 /"IF YOU WANT TO TERMINATE THIS SESSION TYPE 0")
READ(1,15) KX
FORMAT(15)
IF(KX)30,40,20
CALL TRTP(IDCB,IBUF1,IBUF2,NAME)
GO TO 5
CALL TRDS(IDCB,IBUF1,IBUF2,NAME)
GO TO 5
STOP
END
SUBROUTINE TRTP(IDCB,IBUF1,NAME)
DIMENSION IDCB(1296),IBUF1(128),IBUF2(258),NAME(3)
COMMON N2,IBUF2,KSKIP
IDCB(10)=0
CALL DCMC(1,-2,0)
WRITE(1,4)
FORMAT("DISMOUNT & MOUNT AND GIVE FILE NAME")
READ(1,5) (NAME(I),I=1,3)
FORMAT(3A2)
CALL DCMC(0,2,0)
CALL OPEN(IDCB,IERR,NAME,0,0,-2,1296)
ITYPE=IERR
IF(IERR)12,14
WRITE(1,13)IERR
FORMAT("ERROR: FMP",I3)
STOP
K=0

```

```

15 KOUNT=0
16 KOUNT=KOUNT+1
17 CALL READF(IDC8, IERR, IBUF1, 128, N)
18 IF(N>80, 16
19 IF(KOUNT-2>20, 30
20 WRITE(1, 25) ITYPE
21 FORMAT("THE FILE IS TYPE", I2, " PREPARE THE TAPE", /
22 "WHEN YOU ARE READY, TYPE -1 AND PRESS THE DATA RECEIVE KEY"
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

```

51 CONTINUE
   CALL OUTPT
   IF(N2)81,15
80 N2=N
   GO TO 51
81 CONTINUE
   DO 82 J=1,3
   K=0
82 DO 82 I=1,20000
   K=K+1
   CALL CLOSE(IDC B)
   WRITE(1,90)
   FORMAT(/"TRANSFER TO TAPE COMPLETED")
   RETURN
   END
   SUBROUTINE TRDS(IDC B,IBUF1,NAME)
   DIMENSION IDC B(1296),IBUF1(128),IBUF2(258),NAME(3)
   COMMON K,IBUF2,KSKIP
   IDC B(10)=0
   KSKIP=1
   CALL DCMC(1,-2,0)
   WRITE(1,4)
   FORMAT("MOUNT DISC TO RECEIVE; GIVE TYPE AND FILE NAME"
5
1 /"FORMAT IS (11,1X,3A2)")
   READ(1,5) ITYPE,(NAME(I),I=1,3)
   FORMAT(11,1X,3A2)
   CALL DCMC(0,2,0)
   CALL OPEN(IDC B,IERR,NAME,0,0,-2,1296)
   IF(IERR)12,14
   WRITE(1,13) (NAME(I),I=1,3),ITYPE
   FORMAT("FILE NOT FOUND. FILE, ",3A2," IS CREATED AS A TYPE:",I2,
12
13
14

```

```

14 CALL CREAT(IDCB, IERR, NAME, -1, ITYPE, 0, 0, 1296)
15 CONTINUE
15 WRITE(1, 15)
15 FORMAT("PREPARE TAPE: WHEN READY TYPE -1, PRESS RETURN "
1 " AND PRESS DATA SEND KEY"/"TERMINAL MUST BE IN PROMPT MODE">
99 1 READ(1, 99) IDOG
99 FORMAT(15)
20 IF(IDOG) 20, 11
20 K=0
25 CALL INSUB
25 IF(K) 80, 25
35 CONTINUE
35 IF(ITYPE-4) 39, 35, 39
37 K2=K/2
37 DO 37 I=1, K2
37 IBUF1(I)=IBUF2(I)
39 GO TO 51
39 K2=K/4
39 K=0
39 DO 50 I=1, K2
39 K=K+1
39 IY= I AND( IBUF2(K), 37477B)
39 K=K+1
39 IZ= I AND( IBUF2(K), 10060B)
39 IZ= IZ * 4
39 IBUF2(K)= I AND( IBUF2(K), 1B)
39 IF( IBUF2(K) > 41, 42, 41
41 IZ= I OR( IZ, 100000B)
42 IBUF1(I)= I OR( IY, IZ)
50 CONTINUE
51 CALL WRITE( IDCB, IERR, IBUF1, K2)
51 IF( IERR) 55, 20
55 WRITE(1, 56) IERR
56 FORMAT("ERROR: FMP-", I3)

```

```

28      L0BMN1( ERRORS) = 0
29      MAT(1,1) = 1
30      DO 82 J=1,3
31      K=0
32      DO 82 I=1,20000
33      K=K+1
34      WRITE(1,85)
35      FORMAT(/,"TRANSFER COMPLETED")
36      CALL LOCF(IDCB, IERR, IREC, IRB, IOFF, JSEC)
37      ITRUN=JSEC/2-IRB-1
38      CALL CLOSE(IDCB, IERR, ITRUN)
39      RETURN
40      END
41      END$
42
43      DO 10 I=1,10
44      I=I+1
45      DO 10 J=1,10
46      J=J+1
47      DO 10 K=1,10
48      K=K+1
49      DO 10 L=1,10
50      L=L+1
51      DO 10 M=1,10
52      M=M+1
53      DO 10 N=1,10
54      N=N+1
55      DO 10 O=1,10
56      O=O+1
57      DO 10 P=1,10
58      P=P+1
59      DO 10 Q=1,10
60      Q=Q+1
61      DO 10 R=1,10
62      R=R+1
63      DO 10 S=1,10
64      S=S+1
65      DO 10 T=1,10
66      T=T+1
67      DO 10 U=1,10
68      U=U+1
69      DO 10 V=1,10
70      V=V+1
71      DO 10 W=1,10
72      W=W+1
73      DO 10 X=1,10
74      X=X+1
75      DO 10 Y=1,10
76      Y=Y+1
77      DO 10 Z=1,10
78      Z=Z+1
79      DO 10 AA=1,10
80      AA=AA+1
81      DO 10 AB=1,10
82      AB=AB+1
83      DO 10 AC=1,10
84      AC=AC+1
85      DO 10 AD=1,10
86      AD=AD+1
87      DO 10 AE=1,10
88      AE=AE+1
89      DO 10 AF=1,10
90      AF=AF+1
91      DO 10 AG=1,10
92      AG=AG+1
93      DO 10 AH=1,10
94      AH=AH+1
95      DO 10 AI=1,10
96      AI=AI+1
97      DO 10 AJ=1,10
98      AJ=AJ+1
99      DO 10 AK=1,10
100     AK=AK+1
101     DO 10 AL=1,10
102     AL=AL+1
103     DO 10 AM=1,10
104     AM=AM+1
105     DO 10 AN=1,10
106     AN=AN+1
107     DO 10 AO=1,10
108     AO=AO+1
109     DO 10 AP=1,10
110     AP=AP+1
111     DO 10 AQ=1,10
112     AQ=AQ+1
113     DO 10 AR=1,10
114     AR=AR+1
115     DO 10 AS=1,10
116     AS=AS+1
117     DO 10 AT=1,10
118     AT=AT+1
119     DO 10 AU=1,10
120     AU=AU+1
121     DO 10 AV=1,10
122     AV=AV+1
123     DO 10 AW=1,10
124     AW=AW+1
125     DO 10 AX=1,10
126     AX=AX+1
127     DO 10 AY=1,10
128     AY=AY+1
129     DO 10 AZ=1,10
130     AZ=AZ+1
131     DO 10 BA=1,10
132     BA=BA+1
133     DO 10 BB=1,10
134     BB=BB+1
135     DO 10 BC=1,10
136     BC=BC+1
137     DO 10 BD=1,10
138     BD=BD+1
139     DO 10 BE=1,10
140     BE=BE+1
141     DO 10 BF=1,10
142     BF=BF+1
143     DO 10 BG=1,10
144     BG=BG+1
145     DO 10 BH=1,10
146     BH=BH+1
147     DO 10 BI=1,10
148     BI=BI+1
149     DO 10 BJ=1,10
150     BJ=BJ+1
151     DO 10 BK=1,10
152     BK=BK+1
153     DO 10 BL=1,10
154     BL=BL+1
155     DO 10 BM=1,10
156     BM=BM+1
157     DO 10 BN=1,10
158     BN=BN+1
159     DO 10 BO=1,10
160     BO=BO+1
161     DO 10 BP=1,10
162     BP=BP+1
163     DO 10 BQ=1,10
164     BQ=BQ+1
165     DO 10 BR=1,10
166     BR=BR+1
167     DO 10 BS=1,10
168     BS=BS+1
169     DO 10 BT=1,10
170     BT=BT+1
171     DO 10 BU=1,10
172     BU=BU+1
173     DO 10 BV=1,10
174     BV=BV+1
175     DO 10 BW=1,10
176     BW=BW+1
177     DO 10 BX=1,10
178     BX=BX+1
179     DO 10 BY=1,10
180     BY=BY+1
181     DO 10 BZ=1,10
182     BZ=BZ+1
183     DO 10 CA=1,10
184     CA=CA+1
185     DO 10 CB=1,10
186     CB=CB+1
187     DO 10 CC=1,10
188     CC=CC+1
189     DO 10 CD=1,10
190     CD=CD+1
191     DO 10 CE=1,10
192     CE=CE+1
193     DO 10 CF=1,10
194     CF=CF+1
195     DO 10 CG=1,10
196     CG=CG+1
197     DO 10 CH=1,10
198     CH=CH+1
199     DO 10 CI=1,10
200     CI=CI+1
201     DO 10 CJ=1,10
202     CJ=CJ+1
203     DO 10 CK=1,10
204     CK=CK+1
205     DO 10 CL=1,10
206     CL=CL+1
207     DO 10 CM=1,10
208     CM=CM+1
209     DO 10 CN=1,10
210     CN=CN+1
211     DO 10 CO=1,10
212     CO=CO+1
213     DO 10 CP=1,10
214     CP=CP+1
215     DO 10 CQ=1,10
216     CQ=CQ+1
217     DO 10 CR=1,10
218     CR=CR+1
219     DO 10 CS=1,10
220     CS=CS+1
221     DO 10 CT=1,10
222     CT=CT+1
223     DO 10 CU=1,10
224     CU=CU+1
225     DO 10 CV=1,10
226     CV=CV+1
227     DO 10 CW=1,10
228     CW=CW+1
229     DO 10 CX=1,10
230     CX=CX+1
231     DO 10 CY=1,10
232     CY=CY+1
233     DO 10 CZ=1,10
234     CZ=CZ+1
235     DO 10 DA=1,10
236     DA=DA+1
237     DO 10 DB=1,10
238     DB=DB+1
239     DO 10 DC=1,10
240     DC=DC+1
241     DO 10 DD=1,10
242     DD=DD+1
243     DO 10 DE=1,10
244     DE=DE+1
245     DO 10 DF=1,10
246     DF=DF+1
247     DO 10 DG=1,10
248     DG=DG+1
249     DO 10 DH=1,10
250     DH=DH+1
251     DO 10 DI=1,10
252     DI=DI+1
253     DO 10 DJ=1,10
254     DJ=DJ+1
255     DO 10 DK=1,10
256     DK=DK+1
257     DO 10 DL=1,10
258     DL=DL+1
259     DO 10 DM=1,10
260     DM=DM+1
261     DO 10 DN=1,10
262     DN=DN+1
263     DO 10 DO=1,10
264     DO=DO+1
265     DO 10 DP=1,10
266     DP=DP+1
267     DO 10 DQ=1,10
268     DQ=DQ+1
269     DO 10 DR=1,10
270     DR=DR+1
271     DO 10 DS=1,10
272     DS=DS+1
273     DO 10 DT=1,10
274     DT=DT+1
275     DO 10 DU=1,10
276     DU=DU+1
277     DO 10 DV=1,10
278     DV=DV+1
279     DO 10 DW=1,10
280     DW=DW+1
281     DO 10 DX=1,10
282     DX=DX+1
283     DO 10 DY=1,10
284     DY=DY+1
285     DO 10 DZ=1,10
286     DZ=DZ+1
287     DO 10 EA=1,10
288     EA=EA+1
289     DO 10 EB=1,10
290     EB=EB+1
291     DO 10 EC=1,10
292     EC=EC+1
293     DO 10 ED=1,10
294     ED=ED+1
295     DO 10 EE=1,10
296     EE=EE+1
297     DO 10 EF=1,10
298     EF=EF+1
299     DO 10 EG=1,10
300     EG=EG+1
301     DO 10 EH=1,10
302     EH=EH+1
303     DO 10 EI=1,10
304     EI=EI+1
305     DO 10 EJ=1,10
306     EJ=EJ+1
307     DO 10 EK=1,10
308     EK=EK+1
309     DO 10 EL=1,10
310     EL=EL+1
311     DO 10 EM=1,10
312     EM=EM+1
313     DO 10 EN=1,10
314     EN=EN+1
315     DO 10 EO=1,10
316     EO=EO+1
317     DO 10 EP=1,10
318     EP=EP+1
319     DO 10 EQ=1,10
320     EQ=EQ+1
321     DO 10 ER=1,10
322     ER=ER+1
323     DO 10 ES=1,10
324     ES=ES+1
325     DO 10 ET=1,10
326     ET=ET+1
327     DO 10 EU=1,10
328     EU=EU+1
329     DO 10 EV=1,10
330     EV=EV+1
331     DO 10 EW=1,10
332     EW=EW+1
333     DO 10 EX=1,10
334     EX=EX+1
335     DO 10 EY=1,10
336     EY=EY+1
337     DO 10 EZ=1,10
338     EZ=EZ+1
339     DO 10 FA=1,10
340     FA=FA+1
341     DO 10 FB=1,10
342     FB=FB+1
343     DO 10 FC=1,10
344     FC=FC+1
345     DO 10 FD=1,10
346     FD=FD+1
347     DO 10 FE=1,10
348     FE=FE+1
349     DO 10 FF=1,10
350     FF=FF+1
351     DO 10 FG=1,10
352     FG=FG+1
353     DO 10 FH=1,10
354     FH=FH+1
355     DO 10 FI=1,10
356     FI=FI+1
357     DO 10 FJ=1,10
358     FJ=FJ+1
359     DO 10 FK=1,10
360     FK=FK+1
361     DO 10 FL=1,10
362     FL=FL+1
363     DO 10 FM=1,10
364     FM=FM+1
365     DO 10 FN=1,10
366     FN=FN+1
367     DO 10 FO=1,10
368     FO=FO+1
369     DO 10 FP=1,10
370     FP=FP+1
371     DO 10 FQ=1,10
372     FQ=FQ+1
373     DO 10 FR=1,10
374     FR=FR+1
375     DO 10 FS=1,10
376     FS=FS+1
377     DO 10 FT=1,10
378     FT=FT+1
379     DO 10 FU=1,10
380     FU=FU+1
381     DO 10 FV=1,10
382     FV=FV+1
383     DO 10 FW=1,10
384     FW=FW+1
385     DO 10 FX=1,10
386     FX=FX+1
387     DO 10 FY=1,10
388     FY=FY+1
389     DO 10 FZ=1,10
390     FZ=FZ+1
391     DO 10 GA=1,10
392     GA=GA+1
393     DO 10 GB=1,10
394     GB=GB+1
395     DO 10 GC=1,10
396     GC=GC+1
397     DO 10 GD=1,10
398     GD=GD+1
399     DO 10 GE=1,10
400     GE=GE+1
401     DO 10 GF=1,10
402     GF=GF+1
403     DO 10 GG=1,10
404     GG=GG+1
405     DO 10 GH=1,10
406     GH=GH+1
407     DO 10 GI=1,10
408     GI=GI+1
409     DO 10 GJ=1,10
410     GJ=GJ+1
411     DO 10 GK=1,10
412     GK=GK+1
413     DO 10 GL=1,10
414     GL=GL+1
415     DO 10 GM=1,10
416     GM=GM+1
417     DO 10 GN=1,10
418     GN=GN+1
419     DO 10 GO=1,10
420     GO=GO+1
421     DO 10 GP=1,10
422     GP=GP+1
423     DO 10 GQ=1,10
424     GQ=GQ+1
425     DO 10 GR=1,10
426     GR=GR+1
427     DO 10 GS=1,10
428     GS=GS+1
429     DO 10 GT=1,10
430     GT=GT+1
431     DO 10 GU=1,10
432     GU=GU+1
433     DO 10 GV=1,10
434     GV=GV+1
435     DO 10 GW=1,10
436     GW=GW+1
437     DO 10 GX=1,10
438     GX=GX+1
439     DO 10 GY=1,10
440     GY=GY+1
441     DO 10 GZ=1,10
442     GZ=GZ+1
443     DO 10 HA=1,10
444     HA=HA+1
445     DO 10 HB=1,10
446     HB=HB+1
447     DO 10 HC=1,10
448     HC=HC+1
449     DO 10 HD=1,10
450     HD=HD+1
451     DO 10 HE=1,10
452     HE=HE+1
453     DO 10 HF=1,10
454     HF=HF+1
455     DO 10 HG=1,10
456     HG=HG+1
457     DO 10 HH=1,10
458     HH=HH+1
459     DO 10 HI=1,10
460     HI=HI+1
461     DO 10 HJ=1,10
462     HJ=HJ+1
463     DO 10 HK=1,10
464     HK=HK+1
465     DO 10 HL=1,10
466     HL=HL+1
467     DO 10 HM=1,10
468     HM=HM+1
469     DO 10 HN=1,10
470     HN=HN+1
471     DO 10 HO=1,10
472     HO=HO+1
473     DO 10 HP=1,10
474     HP=HP+1
475     DO 10 HQ=1,10
476     HQ=HQ+1
477     DO 10 HR=1,10
478     HR=HR+1
479     DO 10 HS=1,10
480     HS=HS+1
481     DO 10 HT=1,10
482     HT=HT+1
483     DO 10 HU=1,10
484     HU=HU+1
485     DO 10 HV=1,10
486     HV=HV+1
487     DO 10 HW=1,10
488     HW=HW+1
489     DO 10 HX=1,10
490     HX=HX+1
491     DO 10 HY=1,10
492     HY=HY+1
493     DO 10 HZ=1,10
494     HZ=HZ+1
495     DO 10 IA=1,10
496     IA=IA+1
497     DO 10 IB=1,10
498     IB=IB+1
499     DO 10 IC=1,10
500     IC=IC+1
501     DO 10 ID=1,10
502     ID=ID+1
503     DO 10 IE=1,10
504     IE=IE+1
505     DO 10 IF=1,10
506     IF=IF+1
507     DO 10 IG=1,10
508     IG=IG+1
509     DO 10 IH=1,10
510     IH=IH+1
511     DO 10 II=1,10
512     II=II+1
513     DO 10 IJ=1,10
514     IJ=IJ+1
515     DO 10 IK=1,10
516     IK=IK+1
517     DO 10 IL=1,10
518     IL=IL+1
519     DO 10 IM=1,10
520     IM=IM+1
521     DO 10 IN=1,10
522     IN=IN+1
523     DO 10 IO=1,10
524     IO=IO+1
525     DO 10 IP=1,10
526     IP=IP+1
527     DO 10 IQ=1,10
528     IQ=IQ+1
529     DO 10 IR=1,10
530     IR=IR+1
531     DO 10 IS=1,10
532     IS=IS+1
533     DO 10 IT=1,10
534     IT=IT+1
535     DO 10 IU=1,10
536     IU=IU+1
537     DO 10 IV=1,10
538     IV=IV+1
539     DO 10 IW=1,10
540     IW=IW+1
541     DO 10 IX=1,10
542     IX=IX+1
543     DO 10 IY=1,10
544     IY=IY+1
545     DO 10 IZ=1,10
546     IZ=IZ+1
547     DO 10 JA=1,10
548     JA=JA+1
549     DO 10 JB=1,10
550     JB=JB+1
551     DO 10 JC=1,10
552     JC=JC+1
553     DO 10 JD=1,10
554     JD=JD+1
555     DO 10 JE=1,10
556     JE=JE+1
557     DO 10 JF=1,10
558     JF=JF+1
559     DO 10 JG=1,10
560     JG=JG+1
561     DO 10 JH=1,10
562     JH=JH+1
563     DO 10 JI=1,10
564     JI=JI+1
565     DO 10 JJ=1,10
566     JJ=JJ+1
567     DO 10 JK=1,10
568     JK=JK+1
569     DO 10 JL=1,10
570     JL=JL+1
571     DO 10 JM=1,10
572     JM=JM+1
573     DO 10 JN=1,10
574     JN=JN+1
575     DO 10 JO=1,10
576     JO=JO+1
577     DO 10 JP=1,10
578     JP=JP+1
579     DO 10 JQ=1,10
580     JQ=JQ+1
581     DO 10 JR=1,10
582     JR=JR+1
583     DO 10 JS=1,10
584     JS=JS+1
585     DO 10 JT=1,10
586     JT=JT+1
587     DO 10 JU=1,10
588     JU=JU+1
589     DO 10 JV=1,10
590     JV=JV+1
591     DO 10 JW=1,10
592     JW=JW+1
593     DO 10 JX=1,10
594     JX=JX+1
595     DO 10 JY=1,10
596     JY=JY+1
597     DO 10 JZ=1,10
598     JZ=JZ+1
599     DO 10 KA=1,10
600     KA=KA+1
601     DO 10 KB=1,10
602     KB=KB+1
603     DO 10 KC=1,10
604     KC=KC+1
605     DO 10 KD=1,10
606     KD=KD+1
607     DO 10 KE=1,10
608     KE=KE+1
609     DO 10 KF=1,10
610     KF=KF+1
611     DO 10 KG=1,10
612     KG=KG+1
613     DO 10 KH=1,10
614     KH=KH+1
615     DO 10 KI=1,10
616     KI=KI+1
617     DO 10 KJ=1,10
618     KJ=KJ+1
619     DO 10 KK=1,10
620     KK=KK+1
621     DO 10 KL=1,10
622     KL=KL+1
623     DO 10 KM=1,10
624     KM=KM+1
625     DO 10 KN=1,10
626     KN=KN+1
627     DO 10 KO=1,10
628     KO=KO+1
629     DO 10 KP=1,10
630     KP=KP+1
631     DO 10 KQ=1,10
632     KQ=KQ+1
633     DO 10 KR=1,10
634     KR=KR+1
635     DO 10 KS=1,10
636     KS=KS+1
637     DO 10 KT=1,10
638     KT=KT+1
639     DO 10 KU=1,10
640     KU=KU+1
641     DO 10 KV=1,10
642     KV=KV+1
643     DO 10 KW=1,10
644     KW=KW+1
645     DO 10 KX=1,10
646     KX=KX+1
647     DO 10 KY=1,10
648     KY=KY+1
649     DO 10 KZ=1,10
650     KZ=KZ+1
651     DO 10 LA=1,10
652     LA=LA+1
653     DO 10 LB=1,10
654     LB=LB+1
655     DO 10 LC=1,10
656     LC=LC+1
657     DO 10 LD=1,10
658     LD=LD+1
659     DO 10 LE=1,10
660     LE=LE+1
661     DO 10 LF=1,10
662     LF=LF+1
663     DO 10 LG=1,10
664     LG=LG+1
665     DO 10 LH=1,10
666     LH=LH+1
667     DO 10 LI=1,10
668     LI=LI+1
669     DO 10 LJ=1,10
670     LJ=LJ+1
671     DO 10 LK=1,10
672     LK=LK+1
673     DO 10 LL=1,10
674     LL=LL+1
675     DO 10 LM=1,10
676     LM=LM+1
677     DO 10 LN=1,10
678     LN=LN+1
679     DO 10 LO=1,10
680     LO=LO+1
681     DO 10 LP=1,10
682     LP=LP+1
683     DO 10 LQ=1,10
684     LQ=LQ+1
685     DO 10 LR=1,10
686     LR=LR+1
687     DO 10 LS=1,10
688     LS=LS+1
689     DO 10 LT=1,10
690     LT=LT+1
691     DO 10 LU=1,10
692     LU=LU+1
693     DO 10 LV=1,10
694     LV=LV+1
695     DO 10 LW=1,10
696     LW=LW+1
697     DO 10 LX=1,10
698     LX=LX+1
699     DO 10 LY=1,10
700     LY=LY+1
701     DO 10 LZ=1,10
702     LZ=LZ+1
703     DO 10 MA=1,10
704     MA=MA+1
705     DO 10 MB=1,10
706     MB=MB+1
707     DO 10 MC=1,10
708     MC=MC+1
709     DO 10 MD=1,10
710     MD=MD+1
711     DO 10 ME=1,10
712     ME=ME+1
713     DO 10 MF=1,10
714     MF=MF+1
715     DO 10 MG=1,10
716     MG=MG+1
717     DO 10 MH=1,10
718     MH=MH+1
719     DO 10 MI=1,10
720     MI=MI+1
721     DO 10 MJ=1,10
722     MJ=MJ+1
723     DO 10 MK=1,10
724     MK=MK+1
725     DO 10 ML=1,10
726     ML=ML+1
727     DO 10 MM=1,10
728     MM=MM+1
729     DO 10 MN=1,10
730     MN=MN+1
731     DO 10 MO=1,10
732     MO=MO+1
733     DO 10 MP=1,10
734     MP=MP+1
735     DO 10 MQ=1,10
736     MQ=MQ+1
737     DO 10 MR=1,10
738     MR=MR+1
739     DO 10 MS=1,10
740     MS=MS+1
741     DO 10 MT=1,10
742     MT=MT+1
743     DO 10 MU=1,10
744     MU=MU+1
745     DO 10 MV=1,10
746     MV=MV+1
747     DO 10 MW=1,10
748     MW=MW+1
749     DO 10 MX=1,10
750     MX=MX+1
751     DO 10 MY=1,10
752     MY=MY+1
753     DO 10 MZ=1,10
754     MZ=MZ+1
755     DO 10 NA=1,10
756     NA=NA+1
757     DO 10 NB=1,10
758     NB=NB+1
759     DO 10 NC=1,10
760     NC=NC+1
761     DO 10 ND=1,10
762     ND=ND+1
763     DO 10 NE=1,10
764     NE=NE+1
765     DO 10 NF=1,10
766     NF=NF+1
767     DO 10 NG=1,10
768     NG=NG+1
769     DO 10 NH=1,10
770     NH=NH+1
771     DO 10 NI=1,10
772     NI=NI+1
773     DO 10 NJ=1,10
774     NJ=NJ+1
775     DO 10 NK=1,10
776     NK=NK+1
777     DO 10 NL=1,10
778     NL=NL+1
779     DO 10 NM=1,10
780     NM=NM+1
781     DO 10 NN=1,10
782     NN=NN+1
783     DO 10 NO=1,10
784     NO=NO+1
785     DO 10 NP=1,10
786     NP=NP+1
787     DO 10 NQ=1,10
788     NQ=NQ+1
789     DO 10 NR=1,10
790     NR=NR+1
791     DO 10 NS=1,10
792     NS=NS+1
793     DO 10 NT=1,10
794     NT=NT+1
795     DO 10 NU=1,10
796     NU=NU+1
797     DO 10 NV=1,10
798     NV=NV+1
799     DO 10 NW=1,10
800     NW=NW+1
801     DO 10 NX=1,10
802     NX=NX+1
803     DO 10 NY=1,10
804     NY=NY+1
805     DO 10 NZ=1,10
806     NZ=NZ+1
807     DO 10 OA=1,10
808     OA=OA+1
809     DO 10 OB=1,10
810     OB=OB+1
811     DO 10 OC=1,10
812     OC=OC+1
813     DO 10 OD=1,10
814     OD=OD+1
815     DO 10 OE=1,10
816     OE=OE+1
817     DO 10 OF=1,10
818     OF=OF+1
819     DO 10 OG=1,10
820     OG=OG+1
821     DO 10 OH=1,10
822     OH=OH+1
823     DO 10 OI=1,10
824     OI=OI+1
825     DO 10 OJ=1,10
826     OJ=OJ+1
827     DO 10 OK=1,10
828     OK=OK+1
829     DO 10 OL=1,10
830     OL=OL+1
831     DO 10 OM=1,10
832     OM=OM+1
833     DO 10 ON=1,10
834     ON=ON+1
835     DO 10 OO=1,10
836     OO=OO+1
837     DO 10 OP=1,10
838     OP=OP+1
839     DO 10 OQ=1,10
840     OQ=OQ+1
841     DO 10 OR=1,10
842     OR=OR+1
843     DO 10 OS=1,10
844     OS=OS+1
845     DO 10 OT=1,10
846     OT=OT+1
847     DO 10 OU=1,10
848     OU=OU+1
849     DO 10 OV=1,10
850     OV=OV+1
851     DO 10 OW=1,10
852     OW=OW+1
853     DO 10 OX=1,10
854     OX=OX+1
855     DO 10 OY=1,10
856     OY=OY+1
857     DO 10 OZ=1,10
858     OZ=OZ+1
859     DO 10 PA=1,10
860     PA=PA+1
861     DO 10 PB=1,10
862     PB=PB+1
863     DO 10 PC=1,10
864     PC=PC+1
865     DO 10 PD=1,10
866     PD=PD+1
867     DO 10 PE=1,10
868     PE=PE+1
869     DO 10 PF=1,10
870     PF=PF+1
871     DO 10 PG=1,10
872     PG=PG+1
873     DO 10 PH=1,10
874     PH=PH+1
875     DO 10 PI=1,10
876     PI=PI+1
877     DO 10 PJ=1,10
878     PJ=PJ+1
879     DO 10 PK=1,10
880     PK=PK+1
881     DO 10 PL=1,10
882     PL=PL+1
883     DO 10 PM=1,10
884     PM=PM+1
885     DO 10 PN=1,10
886     PN=PN+1
887     DO 10 PO=1,10
888     PO=PO+1
889     DO 10 PP=1,10
890     PP=PP+1
891     DO 10 PQ=1,10
892     PQ=PQ+1
893     DO 10 PR=1,10
894     PR=PR+1
895     DO 10 PS=1,10
896     PS=PS+1
897     DO 10 PT=1,10
898     PT=PT+1
899     DO 10 PU=1,10
900     PU=PU+1
901     DO 10 PV=1,10
902     PV=PV+1
903     DO 10 PW=1,10
904     PW=PW+1
905     DO 10 PX=1,10
906     PX=PX+1
907     DO 10 PY=1,10
908     PY=PY+1
909     DO 10 PZ=1,10
910     PZ=PZ+1
911     DO 10 QA=1,10
912     QA=QA+1
913     DO 10 QB=1,10
914     QB=QB+1
915     DO 10 QC=1,10
916     QC=QC+1
917     DO 10 QD=1,10
918     QD=QD+1
919     DO 10 QE=1,10
920     QE=QE+1
921     DO 10 QF=1,10
922     QF=QF+1
923     DO 10 QG=1,10
924     QG=QG+1
925     DO 10 QH=1,10
926     QH=QH+1
927     DO 10 QI=1,10
928     QI=QI+1
929     DO 10 QJ=1,10
930     QJ=QJ+1
931     DO 10 QK=1,10
932     QK=QK+1
933     DO 10 QL=1,10
934     QL=QL+1
935     DO 10 QM=1,10
936     QM=QM+1
937     DO 10 QN=1,10
938     QN=QN+1
939     DO 10 QO=1,10
940     QO=QO+1
941     DO 10 QP=1,10
942     QP=QP+1
943     DO 10 QQ=1,10
944     QQ=QQ+1
945     DO 10 QR=1,10
946     QR=QR+1
947     DO 10 QS=1,10
948     QS=QS+1
949     DO 10 QT=1,10
950     QT=QT+1
951     DO 10 QU=1,10
952     QU=QU+1
953     DO 10 QV=1,10
954     QV=QV+1
955     DO 10 QW=1,10
956     QW=QW+1
957     DO 10 QX=1,10
958     QX=QX+1
959     DO 10 QY=1,10
960     QY=QY+1
961     DO 10 QZ=1,10
962     QZ=QZ+1
963     DO 10 RA=1,10
964     RA=RA+1
965     DO 10 RB=1,10
966     RB=RB+1
967     DO 10 RC=1,10
968     RC=RC+1
969     DO 10 RD=1,10
970     RD=RD+1
971     DO 10 RE=1,10
972     RE=RE+1
973     DO 10 RF=1,10
974     RF=RF+1
975     DO 10 RG=1,10
976     RG=RG+1
977     DO 10 RH=1,10
978     RH=RH+1
979     DO 10 RI=1,10
980     RI=RI+1
981     DO 10 RJ=1,10
982     RJ=RJ+1
983     DO 10 RK=1,10
984     RK=RK+1
985     DO 10 RL=1,10
986     RL=RL+1
987     DO 10 RM=1,10
988     RM=RM+1
989     DO 10 RN=1,10
990     RN=RN+1
991     DO 10 RO=1,10
992     RO=RO+1
993     DO 10 RP=1,10
994     RP=RP+1
995     DO 10 RQ=1,10
996     RQ=RQ+1
997     DO 10 RR=1,10
998     RR=RR+1
999     DO 10 RS=1,10
1000    RS=RS+1
1001    DO 10 RT=1,10
1002    RT=RT+1
1003    DO 10 RU=1,10
1004    RU=RU+1
1005    DO 10 RV=1,10
1006    RV=RV+1
1007    DO 10 RW=1,10
1008    RW=RW+1
1009    DO
```

```

ASMB,L,T      OUTPUT,7
NAM          OUTPUT
EXT         I013
COM         N2,IBUF2(258)
EQU         13B
SC          NOP
OUTPUT     LDA     N2
ALS        CAX    SC
CAX        LIA    *2
LIA        SSA    *2
JMP        CLF   00
CLF        LDA    CNW2
LDA        OTA   SC
OTA        LDA    DC2
LDA        JSB   I013
JSB        CXA   SC
CXA        SSA    CFILE
SSA        JMP   ADR2
ADR2       LDB   SC
LDB        LIA   SC
LIA        SSA    *2
SSA        JMP   CNW2
*2         LDA   SC
LDA        OTA   SC
OTA        LBT   I013
LBT        JSB   I013
JSB        DSX   UP1
DSX        JMP   SC
JMP        LIA   SC
LIA        SSA    *2
SSA        JMP   *2
*2         UP1
UP1        ADR2
ADR2       DC2
DC2        DC4
DC4        EOT
EOT        CNW2
CNW2       EOF
EOF

```

```

LDA CNW2
OTA SC
LDA DC4
JSB I013
CLF SC
STF 00
JMP OUTPUT,I
LIA SC
SSA *2
JMP CNW2
LDA SC
OTA EOT
LDA I013
JSB SC
CLF 00
JMP OUTPUT,I
DBL IBUF2
OCT 22
OCT 24
OCT 4
OCT 120000
END

```

```

ASMB, L, T
INSUB, 7
NAM INSUB
EXT I013
COM K, IBUF2(258), KSKIP
EQU 13B
SC
INSUB
NOP
LIA SC
SSA *2
JMP 00
CLF KSKIP
LDA DXN1
SSA NEG1
JMP KSKIP
LDA CNW1
STA SC
LDA I013
OTB JSB
LIA SC
NOP
LDA CNW2
OTA SC
LDA DC2
JSB I013
LDA CNW2
OTA SC
LDA DC2
JSB I013
LDA CNW2
OTA SC
LDA DC4
JSB I013
LDX =D0

```

```

LOAD I.F.C. REG.
TEST BUSY BIT
IF ZERO, SKIP THIS ONE
OF INTERRUPT

```

```

LOAD A WITH OUT-WORD
SEND TO I.F.C.
LOAD ROPEN CHAR.
SEND TO TEK. TERM.

```

```

LOAD A WITH PROMPT CHAR.

```

```

LOAD A WITH RCLOSE CHAR.

```

```

LOAD X WITH ZERO

```

LOAD B WITH BYTE ADDR. OF IBUF2
 LOAD A WITH IN-WORD

IS CHAR. TOPEN?
 YES! JUMP.
 NO! IS IT TCLOSE?
 YES! JUMP.

NO! COUNT CHAR. IN BYTE OF IBUF2.
 STORE CHAR. IN BYTE OF IBUF2.
 JUMP AND READ ANOTHER.
 SET END OF FILE FLAG.

STORE CHAR. COUNT IN K

ON INTERRUPT
 RETURN
 DEFN. BYTE ADDR. OF IBUF2
 INPUT WORD
 OUTPUT WORD
 CTRL R
 CTRL M
 CTRL T
 CTRL D

```

RET1  LD8  ADR1
      LDA  CNW1
      JSB  IO13
      LIA  SC
      CPA  NUL
      JMP  UP1
      CPA  DC3
      JMP  DWN3
      JMP  DWN1
      LDA  CNW1
      JSB  IO13
      LIA  SC
      CPA  EOT
      JMP  DWN2
      CPA  LF
      JMP  RET1
      ISX
      SBT  RET1
      JMP  =D-1
      LDA  K
      STA  DWN4
      JMP  DWN4
      STX  K
      CLF  SC
      STF  00
      JMP  INSUB, I
      DBL  IBUF2
      OCT  140000
      OCT  120000
      OCT  0
      OCT  22
      OCT  15
      OCT  24
      OCT  4
      ADR1
      CNW1
      CNW2
      NUL
      DC2
      DC3
      DC4
      EOT
  
```

EOI
DC9
DC9
DC9
HRT
CH
CH
HOL
DMH4
DMH3
DMH5
DMH1
b1
LTH

LF
HEG1
OCT 12
DEC -1
END
EOF

CLMF D
CLMF T
CLMF M
CLMF
ON
IN
DEN
BATCH
ON
SIOSE
SET
SIOSE
NOI
SET
NOI
FOND
FOND

LINE FEED

**APPENDIX E
LISTING OF PROGRAM
TKHP**

455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

FTN,L,T
PROGRAM TKHP2
DIMENSION IDCBC(528,8),IBUF(128),NAME(3,8),KD(8)
COMMON KSKIP,K,L,IBUF
DO 1 I=1,8
IDCB(10,I)=0
KD(I)=0
CALL DCNC(1,-2,0)
WRITE(1,2)
FORMAT("MOUNT DISC TO SEND AND/OR RECEIVE")
DO 4 J=1,8
WRITE(1,3) J
FORMAT("GIVE NAME OF NO. ",I1," FILE")
READ(1,5)(NAME(I,J),I=1,3)
FORMAT(3A2)
CALL DCNC(0,2,0)
DO 438 J=1,8
IF(NAME(1,J)-400)430,432,430
IF(NAME(1,J)-200)408)434,432,434
WRITE(1,433) J
FORMAT("THERE IS NO NO. ",I1," FILE")
KD(J)=-1
GO TO 438
CALL OPEN(IDCBC(1,J),IERR,NAME(1,J),0,0,-2,528)
IF(IERR)455,438
WRITE(1,456)(NAME(I,J),I=1,3)
FORMAT("FILE ",3A2," NOT FOUND. IT IS CREATED.")
CALL CREAT(IDCBC(1,J),IERR,NAME(1,J),20,4,0,0,528)
IF(IERR)40,438
CONTINUE
KIN=8
DO 482 I=1,8
KIN=KIN + KD(I)
WRITE(1,484)KIN

```

```

DO 10 I=1,M
L=0
DO 10 J=1,M
L=L+1
RETURN
END
END$

```

10

EOF

```

484 FORMAT(I2," FILES ARE OPEN")
23 WRITE(1,24)
24 FORMAT(" * * * READY * * * ")
CALL DELAY(4,20000)
KOK=0
L=1
CALL HELP
IF(KOK)33,34
33 N=K/2
CALL WRITE(IDCB(1,LO), IERR, IBUF, N)
IF(IERR)40,35
35 KOK=0
34 IF(KSKIP)40,36,29
29 IF(KSKIP-2)32,31
31 CALL PWNOF(IDCB(1,L))
GO TO 25
32 CALL READ(IDCB(1,L), IERR, IBUF, 128, N)
IF(N)40,28
28 K=2*N
CALL OUT
GO TO 25
36 CONTINUE
KOK=-1
LO=L
CALL INP
GO TO 25
CONTINUE
DO 50 J=1,9
IF(KD(J))50,42
42 CALL CLOSE(IDCB(1,J))
CONTINUE
STOP
END
SUBROUTINE DELAY(M,N)

```



```

STB L=D1
ADB SIX
CPA L=D1
STB L=SEVEN
ADB L=D1
CPA EIGHT
STB L=CR
CPA DWNG
JMP DWN4
JMP CLF SC
STF HELP,I
JMP NDB ADR
LDB N2
LDX SC
LIA SC
SSA *-2
JMP CLF CNW2
LDA SC
LBT JSB IO13
DSX UP2
JMP LDA CNW2
LDA SC
LDA CR
JSB IO13
CLF SC

```

DWNG

OUT

UP2

```

INP
STF 00
JMP OUT,I
NOP SC
LIA SC
SSA *-2
JMP CLF CNW2
LDA SC
LDA PCH
JSB IO13
LDA CNW2
LDA SC
LDA CR
JSB IO13
LDB ADR
LDX ZERO
LDA CNW
JSB IO13
LIA SC
CPA CR
JMP DWNS
ISX UP3
SBT UP3
JMP CXA ZERO
CPA UP3
JMP LDA ZERO
SBT ISX
ISX N2
STX SC
CLF SC
STF 00

```

INP

UP3

DWNS

```

JMP INP,I
END
EOF

```

APPENDIX F LISTING OF FILE TYPES

- 0** **non-flexible disc file**
- 1** **fixed length 128-word record**
- 2** **fixed length records; user defines length**
- 3** **variable length record, sequential access, automatic extents**
- 4** **ASCII code and source programs (otherwise like type 3 files)**
- 5** **relocatable binary code (otherwise like type 3 files)**
- 7** **absolute binary (otherwise like type 3 files)**

REFERENCES

1. *Data Communications Interface Manual*, Tektronix, Inc., Beaverton, Oregon, 1976.
2. *RTE-M Programmer's Reference Manual*, Hewlett-Packard, Cupertino, California, 1977.

LIST OF ABBREVIATIONS AND SYMBOLS

ASSM ASSEMBLER

@ Control @ . Obtain by pressing control and @simultaneously

D Control D character. Obtained by pressing control and D character simultaneously

FMP File Management Package

FORT. FORTRAN

J Control J. Obtained by pressing control and J simultaneously

lu Logical unit number

PROG. PROGRAM

R Control R. Obtained by pressing control and R simultaneously

TCOM Tape Communications Program

TKHP BASIC to Disc Communications Program

XFER Program for transferring files from one disc to disc

4051 Tektronix 4051 graphic system

9885M HP-9885M flexible disc drive

DISTRIBUTION

	No. of Copies		No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12	DRSMI-LP. Mr. Voigt	1
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, Illinois 60616	1	DRDMI-TG. Mr. Huff DRDMI-T. Dr. Kobler DRDMI-TGC, Mr. Griffith DRDMI-TBD DRDMI-TB DRDMI-TI (Record Set) (Reference Copy)	1 1 40 3 3 1 1