File Number 1620-37
Form J26-5547-0

IBM Systems Reference Library

IBM 1620 Program Writing and Testing

This bulletin describes writing and testing techniques useful in programming the IBM 1620 Data Processing System. Included are dump, trace, and utility routines, and a check routine for flags and record marks.

This publication is a reprint of an earlier edition with the format changed to conform to that of the Systems Reference Library.

Address comments regarding this publication to:
IBM, Product Publications Department, San Jose, California

This bulletin discusses program writing and testing techniques that can assist the programmer to produce accurate, time-saving 1620 programs. It also presents a number of short programs that have proved useful in testing machine language programs on the IBM 1620. Five types of programs are given and discussed in the Appendix.

Writing the Program

Programs should be written with the realization that changes or modifications may be required to obtain a correct, working program. They should be written in segments, each of which performs a specific task. These segments should be separated by blank spaces for insertion of additional commands which subsequent testing may indicate to be necessary for a complete, working program.

Frequent use of subroutines is desirable, especially in programs containing numerical calculations. Subroutines or segments can often be advantageously checked out separately, even before other parts of the program are written. In addition, programs composed of subroutines or segments can be more easily extended or modified than programs that are not so divided. Although the length of a program can sometimes be reduced through the use of complex logic, such approaches should be minimized unless core storage space is at a premium. Ease of modification and ready comprehension are often more important than the length of the program.

Linkage to the subroutine should be kept to a minimum. Items of data required for the subroutine can be stored consecutively so that all items can be transferred to the subroutine by using only one storage address. Branch and Transmit, and Branch and Transmit Immediate (BT-27 and BTM-17) commands are useful for branching to subroutines, because either command causes data transfer and branching to the subroutine, and the setting up of the return to the next command in sequence. Otherwise, this threefold task requires three commands.

A program that involves much intermediate calculation before the final result is obtained may be interspersed with commands that cause a typeout (under sense switch control, if desired) of the intermediate data as these data are calculated. After program testing is completed, these commands can be eliminated.

Should an error occur, another useful technique is the inclusion of an independent routine to which a manual branch can be made. This routine causes certain areas to print out. The printout serves as a check on the contents of these areas. Printout, or Dump, routines appear in Appendix A.

Certain common errors peculiar to the program format may occasionally occur as data is composed for the program. Such errors can be checked for by the program, and provision can be made in the program for re-entry of a particular data record via the typewriter. The user is thus saved the necessity of creating a new input tape or card deck before continuing the program.

Many segments or subroutines can be shortened in one or both of the following ways: first, by reducing the length of each data field to the maximum needed for that item, rather than regarding the field length as fixed at an unvarying minimum. Programmers accustomed to fixed word machines may otherwise fail to take advantage of the variable length aspects of the 1620. Second, subroutines can be shortened by rewriting a routine of "X" commands as "X/N" or less commands and executing the reduced routine "N" times in a "tight loop" sequence. However, the increased execution time required for the tight loop operation must be weighed, in the light of time and space requirements, against the greater storage space needed by the "X command" routine.

The importance and usefulness of "desk checking" cannot be overemphasized. Not only does this process include a visual check of the program listing for obvious mistakes, but also manual run of an item of data through

the program. Desk checking can eliminate a surprising number of potential errors in a new program.

Data used for manual or machine check of a program should be carefully selected to ensure a comprehensive check of the numerous combinations that may exist in the logical flow of the program. It is usually advisable to begin checking with data that traces a relatively simple, straightforward, logical flow through the program. Succeeding data then can be selected to take increasingly complex paths so that each new item of data involves the use of a new subroutine or segment. A careful selection of data lessens the chance of a programming error being undetected during the preliminary data check.

To produce a program which is easy to test and is readily understandable to another programmer, the original programmer should generate the following items with his program, even before it is checked out on the 1620:

1. A general flow chart of the main logic of the program. This chart should be as machine independent as possible so that it can be understood by a programmer who is unfamiliar with the 1620.
2. Detailed flow charts of the segments and subroutines that appear as blocks in the main flow chart.
3. A table containing field addresses of the data (or data areas) and address locations of the segments and subroutines comprising the program.
4. The SPS (Symbolic Programming System) listing and machine language translation.
5. Sample data input and the expected resultant output.

   Most of the auxiliary programs available for use as subroutines (i.e., floating point subroutines) or as program testing aids (i.e., trace routines) are located initially in 1620 storage locations 402 to 3999 and 18500 to 19999. To avoid relocation of these prestored subroutines, storage locations 4000 to 18499 should be used first as new programs are written. Routines included in this bulletin are contained within the last 500 positions of core storage, except for the load routine (402 to 842) and the general trace routine (15000 to 17443).

Testing the Program on the 1620

Machine time on the 1620 may often be at a premium. When 1620 time is scheduled for program testing, the programmer's testing plan and the execution of this plan should be carefully determined in advance, so that machine time is not wasted and the next person scheduled is not delayed. Therefore, the programmer who has been assigned 1620 time should first decide how the program can best be tested. The following suggestions may be used as a guide.

1. Make an initial dump of the program to check that it is stored in the correct location and to provide a comparison for the dump usually performed when the test run of the program is complete. To produce a neat, readable listing, use a dump routine such as the one in Appendix B.
2. At minimum, check the critical flags (see FLAGS).
3. If possible, plan to check certain key subroutines or segments independently of the rest of the program. If it appears advisable to check a particular program in its entirety, replace certain B or BB commands with Halts to assure stopping of the program after certain segments are executed. While the program is halted, check the data at that point, execute the branch manually, proceed to the next Halt and again check, etc. Areas of the program that do not contain tight loops can be checked by stepping through a single command at a time.

Trace Routines

For accurate program testing, the programmer must at all times be able to identify and know the location of the program step, segment, or routine under execution at a particular time. If a storage location outside the defined program area is addressed while the program is under test, the programmer can with this knowledge retrace steps to the last known valid location tested and check to see why the program has gone awry. One way to secure this knowledge is to use a trace routine. Five types of trace routines are given in Appendix B. (The trace routines in Appendix B are operative on a 1620 with indirect addressing, only if the indirect addressing is not used in the P field of a branch command.) The general trace routine prints out all commands and the contents of the P and Q fields. Since the trace is time consuming, it should be used only when the programmer is at a loss to find the error in any other way and when a detailed check is necessary to determine the reasons for a computational subroutine producing incorrect results from correct data.

Although the general trace is invaluable in some instances, its use should be limited to short sections of the program and should be considered only as a last resort.

Four other trace routines that avoid the necessity for stepping through the program one command at a time or for manually tracing by placement of Halt commands in the program are given in Appendix B. Essentially, these traces print out only those branches executed by the program, and are therefore much faster than the general trace. These "fast trace" routines give an ex-

cellent trace of the logical flow of the program, and also indicate the last branch executed when a program error occurs. The similarities and differences in these four routines are summarized in Figure 1. They are:

1. Trace Routine for Branches in a Program

This routine prints out all Branch, BT, and BTM commands that are executed. Although it executes all BT and BTM subroutines,it does not trace them, but returns to the trace only on the execution of a Branch Back (BB-42) command.

2. Trace and Count Routine for Branches in a Program

This routine is similar to (1), but has the added feature that any branch executed more than once in sequence is printed out only once, together with the number of times it was executed. This routine is useful in any program that contains several "tight loop" subroutines.

3. Flow Trace

This routine traces BT and BTM subroutines and is useful when the subroutine is ended by a branch command rather than by a BB command.

4. Fast Trace

This routine, while similar to (3), also includes the counting feature found in (2), so that it is useful for programs containing tight loops. In addition, it results in a typeout of all Write Numerically (WN-38) commands and of the output record.

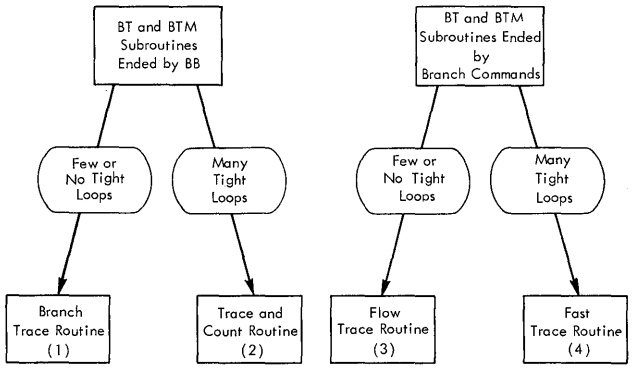

Figure 1. Uses of Various Trace Routines

Symbolic Programming System (SPS)

The attempt to write any 1620 program, except short ones, in machine language is cumbersome and confusing, even to an experienced programmer. Thus, SPS has been used extensively in the writing of 1620 programs. However, when paper tape input is used and when a program is changed many times before the final version is produced, manual translation of SPS may be more advantageous than use of the SPS processor.

Although manual translation may appear formidable to the beginner at first, he soon finds that with practice he can translate the SPS language almost as fast as he can write the numbers down. Manual translation of SPS may save time for the experienced programmer who is working on a 1620 with paper tape input; for the beginner, both manual and SPS processor translation should be attempted to gain practice.

Manual translation has this notable advantage: it familiarizes the programmer with machine language, and since he must work with machine language when he checks out his program on the 1620, he has accomplished two tasks at the same time – translation and machine language skill. Moreover, familiarity with the machine language equivalent usually permits him to check out the program more quickly.

The very qualities that lend versatility to the 1620 machine language may, at first, cause a programmer who is new to variable word length machines to regard machine language as intricate. This feeling usually disappears with added experience, but the programmer also discovers that certain 1620 programming techniques may be difficult to express in SPS. The "execute command" feature contained in the load program (Appendix C) is such a technique.

Once the program has been tested in machine language form, the programmer should check the SPS, which is an invaluable tool for relocating programs.

Loading and Timing

After a correct, working program has been evolved and thoroughly tested, a continuous record tape is useful for loading the 1620; however, such a tape is difficult to obtain for an unchecked program. Except for short programs, cutting a new tape each time corrections are to be made requires an excessive length of time to test a program.

Since even an experienced programmer averages only about 50 commands an hour in testing a program on the 1620, he will require 20 hours of machine testing time for a 1,000-command program. Normally, the programmer has only two hours on the machine at any one time. This period gives him more than enough information to make corrections and check out certain portions of the program. If longer intervals are attempted on the 1620, testing efficiency often decreases. The time considerations cited are averages and depend on the type of program and the experience of the programmer. A beginner may require more time.

If the assumption is made that the programmer modifies and corrects a 1,000-command program at least ten times before evolving the final version, the task of

recompiling the program in SPS each time to include minor changes is obviously laborious and time-consuming. Cutting a completely new tape each time is almost as laborious. An efficient method of reducing time spent in producing tapes is to use a load program that contains a series of short records, each record containing one command and the address into which that command is loaded. To correct the program, a short tape containing the desired corrections is cut. This tape is loaded, following the loading of the main program tape, and replaces data previously loaded from the main tape.

A load program such as the one described is included in Appendix C to illustrate the general approach. Each programmer may prefer to write load programs more directly applicable to his own problem. The significant concept is that short data records can be loaded into addresses contained in the records in such a manner that the sequence of the records is unimportant. Such an approach can save considerable time and effort in testing a program.

Preparing a Patch Tape

One way to correct or alter a program in core storage is to incorporate all corrections into a "patch tape." The following patch procedure offers these advantages:

1. Any number of consecutive digits in core storage can be corrected as a group. The last character corrected can be a record mark.
2. The corrections can be loaded in any sequence.
3. The patch program uses only locations 00000 to 00091, which are normally free. No buffer area is required.
4. The program does not use the arithmetic tables.

Punching the Patch Tape

1. To punch the patch tape, load the paper tape punch, and punch feed codes in the leader. Press Reset and insert 360002400100 380002400200. Press Release and Start.

   Type: 360006200300
   260005900071
   260007800071
   160007$\bar{2}$03002
   250007119999
   $\bar{3}$60007$\bar{2}$$\bar{1}$9999‡

   If MAR indicates 00096, press Release and SIE. Then press Insert, Release, and Start.

   Type: 251999900071
   4900000‡

   If MAR indicates 00043, press Release and SIE. This entire procedure causes two records containing the patching program to be punched. Alternatively, the program can be duplicated from another tape.
2. To add corrections to the patch tape, press Reset and insert:

00000	360100000100	Type in the (following)
00012	440001201005	data (A . . . ., B . . . .,
00024	450002401010	C . . . .).
00036	380100000200	
00048	380101100200	
00060	4900000	

   NOTE: A read-in area other than 01000 can be used by changing the underlined addresses.
   Press Release and Start. The data (A. . .B. . . C. . .) can then be typed.

   If the console locks after program step 12 or 14 (owing to the fact that a flag or a record mark was omitted when the data was typed), press Stop, Insert, Release, and Start. Then retype the data and the routine should proceed correctly.

   Type: AAAAABBBBB‡ C . . . C‡
   AAAAA — the address of the left-hand digit to be corrected.
   $\bar{B}$BBBB — the address of the digit to the right of the correction. It is destroyed by a record mark and replaced. The flag is essential.
   C . . . C — the correction. A record mark is permissible only as the last character of the correction.

   If an error was made, press Release, Insert, Release, and Start. Retype the data (A, B, and C). If an error was not made, press Release and Start.
3. The final correction to be added should be a branch: 00000 $\bar{0}$0019‡ 48 00000 00000 49 XXXXX where XXXXX is the address of the first instruction of the program. Leave an unpunched "trailer" on the tape for subsequent corrections.

   To add corrections to the patch tape, duplicate the tape up to the point of the final branch correction and add the corrections and a new branch correction. When the program is completely tested, a single program tape may be punched to eliminate the patch tape.

Using the Patch Tape

Load the program. Before executing it, place the patch tape on the paper tape reader, press Reset, insert 360000000300, press Release and Start. When the cor-

rections are loaded, the program halts to permit a data tape to be loaded. The start key may be pressed to execute the branch to the corrected program.

Flags

The beginning 1620 programmer may experience some difficulty in the proper use of flags in the 1620 machine language. Neglecting to set a single flag may result in destroying a large area of data in core storage, often leaving the programmer with little idea as to the cause. If one or more of the following are executed before a run with a new program is attempted, errors may be discovered that can save reloading the program:

1. Instead of clearing core storage to zeros before loading the program, clear to flag zeros by using 31$\bar{0}$0000300002. This Transmit Record command limits the amount of transfer that may occur outside the program storage area if the setting of one or more flags has been neglected.
2. Check that the Q field of each immediate command contains a flag in one of the first four positions (Q_7 through Q_{10}) of the field. The routine in Appendix D accomplishes this check rapidly.
3. Set a flag in the first position (O_0) of every command, thus limiting the length of any field transfer that may occur within the program storage area. The routine in Appendix D leaves the first position of the command flagged so that an error in field transfer is easily recognized in a dump operation.
4. If a data area is used into which a field will be set at some intermediate program step, it is helpful to preset flags in this area. Thus, if the data is set incorrectly, a subsequent transfer field operation will not seriously harm the execution of the program. Because the above precautions – especially (1) and (4) – may occasionally result in the continuance of the program even when an error occurs, the error may not be detected immediately. Usually these errors can be pinpointed by examining the typeout of intermediate data, if such a typeout is available.

"Packing"

Where space is at a premium, the programmer may want to "pack" commands. Packing is possible with an unconditional Branch (B-49) command, where the last four digits of the Q field (Q_8 through Q_{11}) may be used to start the next command. It is also possible with the Branch Back (BB-42) command, where the last ten digits may be used. As these commands (especially B) are used frequently in most programs, considerable storage space can thus be saved.

It should be noted that the Q field of a Set Flag (SF-32) or Clear Flag (CF-33) command can be used to store any 5-digit field. The P field of a Control (K-34) command or the P and Q fields of a Halt (H-48) command can also be used for storage.

Commands used for initialization or setting of the program that are performed only once at the start of the program may be placed in areas reserved for data or for data manipulation. Alternatively, through use of a load program (Appendix C), initialization commands may be executed in sequence as they are loaded on top of one another from the program tape. Such a load program may be used to read in and read out alphameric statements without taking up any space needed for the program.

Branch and Transmit

A Branch and Transmit (BT-27) command not involving transfer of any fields uses

BT A, A – 1

to branch to A. The programmer should ensure that a flag is set at some location to the left of A–1. The greater the number of storage positions between the flag and location A–1, the more time is consumed in executing the BT command.

Program Indicators

When planning to create numerical indicators in the program to direct program branching, it is well to remember that any single digit can indicate six different possible branches, without the use of compare commands. Figure 2 shows a flow chart and partial program

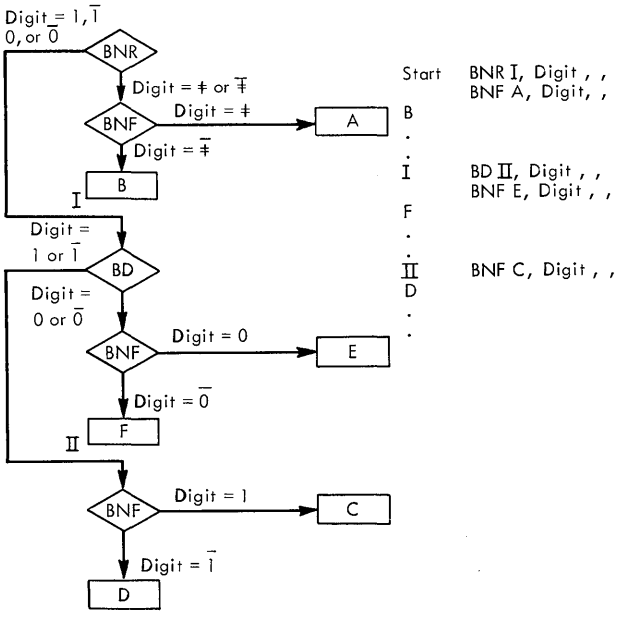

Figure 2. Six-way Branching Scheme

that illustrate a scheme for branching to A, B, C, D, E, or F respectively, depending on whether the indicator is ‡, $\bar{‡}$, 1, $\bar{1}$, 0, or $\bar{0}$, respectively.

The use of compare commands and 2-digit fields for the same scheme would require more commands and more execution time. However, record marks should be used sparingly in a program, as indicated under RECORD MARKS.

The three commands—Branch No Record Mark, Branch No Flag, and Branch on Digit (BNR, BNF, and BD)—are very useful in determining the path of the program. Where either branch commands or compare commands can be used successfully, the use of branch commands is normally more advantageous.

Parity Errors

The following program determines if any parity errors exist in storage:

00000	430001200000	(BD-43)
00012	110001100001	(AM-11)
00024	4900000	(B-49)

If a parity error is present, the program stops in the E cycle with an MBR-E or MBR-O check. When a parity error is detected, the MAR address is always even because this program searches core storage in an ascending order of addresses. However, if the MBR-O light is on, the parity error is in the following odd address. The end of the program occurs on a MAR check, with 20000 showing in the Memory Address Register. This program in no way affects anything in core storage.

Record Marks

The use of record marks in the middle of a program should generally be avoided; normally their use should be reserved for locations in the data area or at the end of the program. A special program is required to duplicate tapes containing record marks that are end-of-line punches on the tape.

One programmer may experience difficulty in running another's program. If the reason for the difficulty is not apparent, it is advisable to check the program for its dependence upon a record mark at address 00400, and if a record mark is required at that address, to check that it is present.

A MAR check results if an attempt is made to use a record mark as part of a numerical field and if it is used to perform a compare or numerical operation with that field. A MAR check also results from reading an address that has a record mark as an address digit. These are MAR checks that often result from an untested program.

A record mark appearing as the first character in the record causes the 1620 to fail to execute a Write Numerically (WN-38) or Write Alphamerically (WA-39) command. In this event, depress the Release and Start keys so that the 1620 continues with the next command.

MAR Check

If a MAR check occurs during an I cycle (the cycle is indicated by the instruction and execute lights), the address in the Memory Address Register is the address of one of the twelve digits of the command in which the MAR check occurred. However, if the MAR check occurs during an E cycle, the Memory Address Register usually contains the address of a digit in the tables or in either the P or Q field locations. To determine the address of the command being executed when the error occurred, the Instant Stop key should be depressed twice, following depression of the Reset key. The address of the command following the desired address then appears in the Memory Address Register.

Check on Table Loading

If a checked computational program has been properly loaded and fails to operate, it is always in order to check that the tables have been properly loaded into storage.

Program Testing Aids

Appendices A through E include a number of short programs useful in testing machine language programs on the IBM 1620 Data Processing System.

NOTE: In some cases the SPS has not been checked; however, the machine language version has been checked in all cases. The 407 printout is as follows: Columns 56 to 75 are numerical and a flag zero in these columns is represented by an eleven punch and appears as a hyphen (-). A flag 1 appears as J, flag 2 as K, etc. The record mark, which is represented by 0, 2, and 8 punches, appears as Z. Column 55 contains a record mark represented by an eleven and twelve punch and appears as a hyphen (-). The twelve punch or the plus sign appears as an &.

Appendix A

Dump Routine

```
J1010* PROGRAM PRINTOUT  ROUTINE TO TYPE 1620 PROGRAMS WITH 1 INSTRUCTION -
J1020* ON A LINE AND WITH SPACING BETWEEN THE FIELDS.  THE FIRST AND LAST -
J1030* ADDRESS IS SPECIFIED AND INSTRUCTIONS ARE ASSUMED TO BE IN STRAIGHT-
J1040* SEQUENCE WITH EACH 12 CONSECUTIVE DIGITS COMPRISING AN INSTRUCTION.-
J1050* IF PROGRAMS ARE PACKED, EACH BLOCK OF CONSECUTIVE STEPS MUST BE    -
J1060* PRINTED SEPARATELY. ENTRY TO THE PROGRAM IS MADE BY INSERTING       -
J1070* 49 START. AFTER DEPRESSING THE RELEASE AND START KEYS, A SERIES OF -
J1080* UP TO FIVE STARTING AND ENDING ADDRESSES MAY BE SPECIFIED WITH A   -
J1090* RECORD MARK FOLLOWING THE LAST SET AS FOLLOWS                       -
J1091*                                                                   -
J1100*                          XXXXX YYYYY                              -
J1110*                          XXXXX YYYYY                              -
J1120*                          XXXXX YYYYYRM                            -
J1130*                                                                   -
J1140* WHERE XXXXX IS THE FIRST ADDRESS IN A BLOCK TO BE PRINTED AND YYYYY-
J1150* IS THE LAST. THE RESULTS WILL APPEAR AS FOLLOWS                    -
J1160*                                                                   -
J1170*                          AAAAA   OO PPPPP QQQQQ                   -
J1180*                                                                   -
J1190* WHERE AAAAA IS THE ADDRESS OF THE INSTRUCTION, OO IS THE OP CODE,  -
J1200* PPPPP IS THE P FIELD, AND QQQQQ IS THE Q FIELD. THE PROGRAM TO BE  -
J1210* PRINTED MUST BE LOADED FIRST, INCLUDING TABLES, AND THEN THIS      -
J1220* PROGRAM FOLLOWS.  ONE TAB MUST BE SET 9 SPACES TO THE RIGHT OF THE -
J1230* LEFT MARGIN.                                                       -
J1240      DORG19500,,,                                -19500
J1250START RNTYINPUT-55,,, TYPE 1ST AND LAST ADDRESSES-19500   361986200100
J1260      BNR START&26,INPUT-55,, BRANCH ON RM        -19512   451952619862
J1270      H   ,,, HALT IF RECORD MARK IS PRESENT     -19524   480000000000
J1280      DORG*-9,,,                                  -19526
J1290SET   SF  INPUT-55,,, SET FLAG IN HIGH ORDER      -19526   321986200000
J2010      SF  INPUT-49,,, SET FLAG IN HIGH ORDER      -19538   321986800000
J2020BEGIN RCTY,,, RETURN CARRIAGE                     -19550   340000000102
J2030      TF  PRINTB-1,INPUT-51,, 1ST ADD TO PRINT    -19562   261994219866
J2040      WNTYPRINTB-5,,, PRINT INST ADDRESS          -19574   381993800100
J2050MOVE  TFM TD&6,PRINTB-12,, TO P FIELD OF TD INST  -19586   1619616J9931
J2060      TF  TD&11,INPUT-51,, TO Q FIELD OF TD INST  -19598   261962119866
J2070TD    TD  00000,00000,, MOVE INST TO PRINT AREA   -19610   250000000000
J2080      CM  TD&6,PRINTB-1,, COMP. TO 12TH POSITION  -19622   1419616J9942
J2090      BE  CONT,,, IF EQUAL GO TO CONT             -19634   461967801200
J2100      AM  TD&6,1,10, MODIFY P FIELD BY 1          -19646   111961600001
J2110      AM  TD&11,1,10, MODIFY Q FIELD BY 1         -19658   111962100001
```

```
J2120       B   TD,,, BRANCH BACK TO TD TO REPEAT      -19670   491961000000
J2130       DORG*-3,,,                                 -19678
J2140CONT   TR  PRINTA-12,PRINTB-12,, MOVE TO PRINTA   -19678   311991819931
J2150       TD  PRINTB-10,00400,, RM TO P2 OF PRINTB   -19690   251993300400
J2160       TD  PRINTA-5,00400,, RM TO Q7 OF PRINTA    -19702   251992500400
J2170       TBTY,,, TABULATE                           -19714   340000000108
J2180       WNTYPRINTB-12,,, PRINT OP CODE             -19726   381993100100
J2190       SPTY,,, SPACE                              -19738   340000000101
J2200       WNTYPRINTA-10,,, PRINT P FIELD             -19750   381992000100
J2210       SPTY,,, SPACE                              -19762   340000000101
J2220       WNTYPRINTB-5,,, PRINT Q FIELD              -19774   381993800100
J2230       C   INPUT-51,INPUT-45,, COMPARE TO LAST    -19786   241986619872
J2240       BNL NEXT,,, BRANCH TO NEXT BLOCK           -19798   461983001300
J2250       AM  INPUT-51,12,10, INC. INST. ADD. BY 12  -19810   1119866000J2
J2260       B   BEGIN,,, BRANCH UP TO REPEAT           -19822   491955000000
J3010       DORG*-3,,,                                 -19830
J3020NEXT   TR  INPUT-55,INPUT-44,, ADVANCE ADDRESSES  -19830   311986219873
J3021       RCTY,,,                                    -19842   340000000102
J3030       B   START&12,,, BRANCH TO START&12          -19854   491951200000
J3040       DORG*-3,,,                                 -19862
J3050* AREA DEFINITIONS                                -
J3060INPUT  DS  56,,,                                  -
J3070PRINTADC   13,0,,                                 -
J3080PRINTBDC   13,0@,,                                -19943   Z
J3090       DENDSTART,,,                               -
```

Dump and Simulate Routine

The following program dumps a program from core storage in the form:

AAAAA OO PPPPP QQQQQ.

In addition, the dump routine simulates the execution of B (49), BT (27), BTM (17), and BB (42) instructions. Therefore the dump routine, after dumping a branch command, continues dumping at the address specified by the P field of the command.

If sense switch 4 is on, it simulates any conditional branch that occurs. If sense switch 3 is on, the routine halts on a conditional branch in order to allow the operator time to consider the setting he wishes for sense switch 4.

Linkage to this routine is 1619559AAAAA
4919500
where AAAAA is the starting address.

The routine uses storage space from 19500 to 19999.

```
SPS         LISTING                                    ADDRESS COMMAND
91000       DORG19500,,,                               -
91010START  RCTY,,,RETURN CARRIAGE                     -19500   34       01 2
91020       TF  TYLO,TRAND&11,,,SET TYPE OUT           -19512   261996219559
91030       WNTYTYLO-4,,, TYPE ADDRESS                 -19524   3819958 01
91040       TFM TRAND&6,PRINTB-12,7, RESET LOOP        -19536   1619554J9986
91050TRAND  TD  ,,, TRANSFER COMMAND BY DIGITS         -19548   25
91060       AM  TRAND&6,01,10,STEP BY ONE              -19560   1119554    01
91070       AM  TRAND&11,01,10,                        -19572   1119559    01
91080       CM  TRAND&6,PRINT8,7,TEST FOR END          -19584   1419554J9998
91090       BNZ TRAND,,, LOOP TRANSFER OF DIGITS       -19596   4719548 12
91100       TR  PRINTA-10,PRINTB-10,,  MOVE RECORD     -19608   311997519988
91110       TD  PRINTA-5,PRINTB,, SET RMKS             -19620   251998019998
91120       TD  PRINTB-10,PRINTB,,                     -19632   251998819998
91130       TBTY,,, TABULATE                           -19644   34       01 8
91140       WNTYPRINTB-12,,, PRINT OPERATION CODE      -19656   3819986 01
91150       SPTY,,,                                    -19668   34       01 1
91160       WNTYPRINTA-10,,, PRINT P FIELD             -19680   3819975 01
91170       SPTY,,,                                    -19692   34       01 1
91180       WNTYPRINTB-5,,, PRINT Q FIELD              -19704   3819993 01
91190       SF  PRINTB-12,,,                           -19716   3219986
91200       CM  PRINTB-11,48,10,TEST FOR HALT &BRANCH  -19728   1419987    M8
```

```
91210       BP   BBB,,, BRANCH FOR 49 COMMAND              -19740   4619872 11
91220       BZ   START,,, HALT 48 COMMAND                  -19752   4619500 12
91230       CM   PRINTB-11,17,10, TEST FOR BTM             -19764   1419987   J7
91240       BZ   BTBTM,,,                                  -19776   4619916 12
91250       CM   PRINTB-11,27,10, TEST FOR BT              -19788   1419987   K7
91260       BZ   BTBTM,,,                                  -19800   4619916 12
92010       CM   PRINTB-11,42,10,TEST FOR BB OR OTHER B    -19812   1419987   M2
92020       BZ   BTBB,,,                                   -19824   4619936 12
92030       BN   START,,, CONTINUE                         -19836   4719500 13
92040       BC3 HALT,,, SENSE SWITCH 3 ON STOPS DUMP       -19848   4619956 03
92050       BNC4START,,,SENSE SWITCH 4 ON CON. BRANCH      -19860   4719500 04
92060BBB    SF   PRINTA-10,,,FLAGP FIELD                   -19872   3219975
92070       TF   TRAND&11,PRINTA-6,, SET NEW ADDRESS       -19884   261955919979
92080       RCTY,,, STORE INP FIELD RETURN FOR 42          -19896   3400000 01 2
92090STORE  DS   5,*-5,,                                   -
92100       B    START,,                                   -19908   4919500 2619
92110       DORG*-3,,,                                     -
92120BTBTM  TF   STORE,TRAND&11,, STORE ADDRESS FOR BB      -19916   261990319559
92130       B    BBB,,, EXECUTE 17 OR 27                   -19928   491987202619
92140       DORG*-3,,,                                     -
92150BTBB   TF   PRINTA-6,STORE,,SET STORED ADDRESS         -19936   261997919903
92160       B    BBB,,, EXECUTE 42                         -19948   491987204800
92170       DORG*-3,,,                                     -
92180HALT   H    ,,,                                       -19956   48      Z
92190       B    BBB-12,,,TEST TO BRANCH ON CONDITIONAL    -19968   491986000000
92200PRINTADC    11,0000000000@,*&6                        -19980   00000Z000000
92210PRINTBDC    13,000000000000@,,                        -19987   000000000002
92220TYLO   DC   5,000000,HALT&6,                          -
92230       DC   1,@,HALT&7,                               -
92240       DENDSTART,,,
```

Appendix B

General Trace

The general trace and the flow trace (No. 3) routines are available at the DP Library Services Department. Requests for the program tapes and accompanying documentation should be placed with the appropriate applied programming support group.

Assembly

The general trace program requires 2443 locations of core storage. The small parameter table (Table I) used with the program requires additional storage. This storage varies, depending upon the number of parameters specified. The selective trace is supplied in symbolic form with comments and is completely relocatable.

Table I Small Parameters

Operation Code	*Trace Output*
A, AM, S, SM	Contents of P field after execution; contents of Q field.
M, MM	Multiplicand; multiplier; product.
C, CM	Contents of P field; contents of Q field.
TD, TDM, CF, SF	Digit at P after execution.
TF, TFM	Contents of P field after execution.
BT, BTM	Contents of (P–1) field after execution.
LD, LDM	Dividend positioned in fixed product area.
D, DM	Quotient; remainder; divisor.
TR	Record at P after execution.
K	No additional output.
DN, RN, RA, WN, WA	First record read from or punched on tape. (Typewriter I/O functions normally).
NOP	No additional output.
BB	No additional output.
BD, BNR, BNF	Digit at P.
BI, BNI, B	No additional output.

Operation

To use the general trace, the programmer should first load his object program, then the trace program (using 360000000300), and finally any data tape necessary. He then depresses Start and a branch occurs to the first instruction of the trace. This instruction is a RNTY and the programmer must type in the address of the starting location in his program, in the form $\overline{X}$XXXX. When he presses Release and Start, the second trace instruction is executed. This is also a RNTY and he must now type in the addresses of the instructions with which he wishes to start and stop tracing. For example, he may type in $\overline{A}$AAAA$\overline{B}$BBBB$\overline{C}$CCCC$\overline{D}$DDDD$\overline{E}$EEEE $\overline{F}$FFFF‡. The trace program interprets this to mean that when it reaches the instruction with address $\overline{A}$AAAA (or $\overline{C}$CCCC or $\overline{E}$EEEE), it is to begin tracing.

The trace continues until the corresponding instruction at $\overline{B}$BBBB (or $\overline{D}$DDDD or $\overline{F}$FFFF) is reached (but not executed). At this point the trace terminates. The trace starts again when the instruction at $\overline{A}$AAAA, $\overline{C}$CCCC, or $\overline{E}$EEEE is encountered, and terminates when the corresponding terminal parameter address ($\overline{B}$BBBB, $\overline{D}$DDDD, or $\overline{F}$FFFF respectively) is encountered. There is no limit to the number of parameter addresses that may be given, provided each trace starting address is followed immediately by its corresponding trace terminating address.

Note that if the trace starts at $\overline{A}$AAAA, it does not terminate until it reaches $\overline{B}$BBBB, even if it executes the instructions at $\overline{C}$CCCC, $\overline{D}$DDDD, etc. In other words, once the trace starts, it continues until it reaches the terminal address of the particular parameter set. Note, too, that each parameter address must be flagged in the high-order position and that the last parameter must be followed by a record mark (‡).

When in the nontrace mode, the general trace monitors each instruction of the object program (i.e., each object instruction executed under trace control) but there is no typeout. When tracing begins, the following is typed out for each object instruction: the address of the instruction, the operation code, the P address, and the Q address exactly as they appear in storage. However, if the instruction contains a record mark, only that part of the instruction up to, but not including the record mark, is typed. In addition, other information, as indicated in Table I, may be typed out.

Should the programmer wish to stop tracing, he must do so only when the program is in the tracing mode. He must depress Stop while the program is typing out the instruction address. He may then insert a branch to the address just typed out and his program then executes without trace control.

NOTE: The programmer must not terminate the trace during the execution of a subroutine that is linked to his program with either a BT or BTM instruction and a BB instruction. If he does, the BB will not execute properly, since the trace simulates, but does not execute, BT, BTM, or BB instructions.

If the programmer wishes to redefine the trace parameters while the program is executing, he may do so by stopping the trace in the manner described above. He must then insert a branch to the first instruction of the selective trace program. At the first RNTY, he enters the address of the instruction at which he halted the trace. At the second RNTY, he may enter the new trace parameters. When he presses Release and Start, the trace resumes at the point where it was interrupted, but now traces only the newly defined areas.

1. Trace Routine for Branches in a Program

This routine prints out all branches that are executed, along with their addresses. The routine is in locations 19500 to 19992. BT and BTM commands are printed out and executed but not traced. H commands halt the routine at 19788; to continue, press Start. The address of the command being traced is at 19987.

Linkage for the routine is 1619595@@@@@@
4919500

where @@@@@@ is the starting address.

To eliminate typing of BTM and BT commands, place a 41 at both 19668 and 19692. To modify for nontypeout of branches on a sense switch, 19 units of storage space must be used at some location. Type the address of this location into 19982. At this address, place 461950000X00
4919820

where X stands for the sense switch used. When the sense switch is on, the typing of branch commands is eliminated.

To eliminate BT and BTM typeout, and add sense switch control without adding space, type into 19656:

491970400000
461950000X00
4919820

and type into 19982, 19668.

The best place to stop the routine is during typing.

To leave trace on the execution of a particular branch command, place a flag in the 01 position of that branch command. The program then executes without tracing.

A Branch, BT, or BTM command that contains a flag in either P_3, P_4, or P_5 results in having the trace branch to the 4, 3, or 2 digit location defined by that flagged field. If the trace "hangs up," check the last branch command printed out for a flag in one of these P-field locations. A flag in the P_6 position of a branch command will cause an error as this is considered a negative number.

To eliminate the typeout of branches on a sense switch, use the following changes and additions:

```
T2160       B   SENSE,,,                                    -19980   4919840
T2191       DORG19480,,,                                    -
T2200SENSE  BC4 START,,,                                    -19480   491950000400
T2210       B   BTBTM&12,,,                                 -19492   4919820 1619

NUMERICAL LISTING FOR TRACE OF BRANCHES ROUTINE

1619535  -0047195360110032195 35  -0047195600120032195 3400000          J9500    2
261999119595161959 0J978825-     -     1119590   -11119595   -1          J9560    2
1419590J9800471958401200321978 8-     1419789   J74619808 12          J9620    2
1419789   K74619808 12  1419789   M24619900 11  1419736-1 -2           J9680    2
4419764195341419760-1 -1441978819535141978 4-2 -141                   J9740    2
4919500 1619898J97283819987 01  2519800  4003400000 01 13819          J9800    2
788 01  34      01 21519800    449-     3219790     1619898J           J9860    2
9500261965519794161979 4J99564919728 261959519655261979419655        J9920    2
4919820-    -                                                         J9980    2
```

```
SPS         LISTING                                          ADDRESS COMMAND
T1001       DORG19500,,,                                     -
T1010START  TFM IND&11,000,9,CLEAR FLAG IN INDICATORS        -19500   1619535  -00
T1020       BNH *&24,,,                                      -19512   471953601100
T1030IND    SF  IND&11,000,9,SET HP                          -19524   3219535  000
T1040       BNZ *&24,,,                                      -19536   471956001200
T1050       SF  IND&10,,,   SET EZ                           -19548   321953400000
T1060       TF  TYLO,TRAND&11,, STORE ADDRESS                -19560   261999119595
T1070       TFM *&18,EXECU,7,   RESET                        -19572   1619590J9788
T1080TRAND  TD  EXECU,,,TRANSMIT DIGIT BV DIGIT              -19584   250    0
T1090       AM  *-6,01,10,STEP BY 1                          -19596   1119590   -1
T1100       AM  *-13,01,10,STEP BY 1                         -19608   1119595   -1
T1110       CM  TRAND&6,EXECU&12,7,TEST TRANSFER             -19620   1419590J9800
T1120       BNZ TRAND,,,LOOP TO FINISH TRANSFER              -19632   471958401200
T1130PFIELDSF   EXECU,,,STORE P FIELD OF BRANCH IN Q         -19644   32197880
T1140       CM  EXECU&1,17,10,,                              -19656   1419789   J7
T1150       BZ  BTBTM,,,                                     -19668   461980801200
T1160       CM  EXECU&1,27,10,,                              -19680   1419789   K7
T1170       BZ  BTBTM,,,                                     -19692   461980801200
T1180       CM  EXECU&1,42,10,,                              -19704   1419789   M2
T1190       BH  BBB,,,                                       -19716   461990001100
T1200SETINDCM   *&8,01002,710,TURN OFFINDICATORS             -19728   1419736-1 -2
T1210       BNF *&24,IND&10,, CHECK EQUAL ZERO                -19740   441976419534
T1220       CM  *&8,01001,710,SET EQUAL ZERO                 -19752   1419760-1 -1
T1230       BNF *&24,IND&11,, CHECK HIGH POSITIVE            -19764   441978819535
T1240       CM  *&8,02001,710, SET HIGH POSITIVE             -19776   1419784-2 -1
T1250EXECU  NOP ,,, COMMAND PLACED HERE                      -19788   41
T1260       B   START,,,                                     -19800   491950001619
T2001       DORG*-3,,,                                       -
T2010BTBTM  TFM END&6,SETIND,7, BT OR BTM COMMAND            -19808   1619898J9728
T2020       WNTYTYLO-4,,,  TYPE OUT ADDRESS                  -19820   381998700100
T2030       TD  EXECU&12,400,, SET RECORD MARK               -19832   251980000400
T2040       SPTY,,,SPACE CARRIAGE                            -19844   340000000101
T2050       WNTYEXECU&12,,,TYPE OUT COMMAND                  -19856   381978800100
T2060       RCTY,,,RETURN CARRIAGE                           -19868   340000000102
T2070       TDM EXECUT&12,4,,REPLACE RECORD MARK             -19880   151980000004
T2080END    B   ,,,                                          -19892   490    03219
T2081       DORG*-3,,,                                       -
T2090BBB    SF  EXECU&2,,,                                   -19900   321979000000
T2100       TFM END&6,START,7,SET FOR BRANCH COMMAND         -19912   1619898J9500
T2110       TF  PFIELD&11,EXECU&6,,STOR P FIELD              -19924   261965519794
T2120       TFM EXECU&6,BBI,7,SET RETURN FOR BRANCH          -19936   1619794J9956
T2130       B   SETIND,,,                                    -19948   491972802619
```

```
T2131       DORG*-3,,,                                          -
T2140BBI    TF   TRAND&11,PFIELD&11,, SET NEW ADDRESS    -19956    261959519655
T2150       TF   EXECU&6,PFIELD&11,,                      -19968    261979419655
T2160       B    BTBTM&12,,,STORE ADDRESS IN Q            -19980    49198200
T2170TYLO   DS   5,*,,,                                   -19987    -0000Z
T2180       DC   1,@,*&1,                                 -
T2190       DENDSTART,,,
```

2. *Trace and Count Routine for Branches in a Program*

This routine traces a program and prints out the branches that are executed and their locations. It further prints out the number of times a branch is executed before another branch occurs. Therefore, a branch occurring 300 times is printed out only once with the number 00300 following the command. BT and BTM commands are executed but not traced. Overflow occurs when a branch is executed more than 99,9999 times, without other branches occurring.

This routine prints out a 23-digit line containing, in order, the address of the branch, the record mark, the branch command, and the number of times the branch was executed.

This routine is located at 19486 to 19999. The address of the command being traced is at 19981 to 19986. There are record marks at 19981 and 19987.

The best place to stop the trace routine is at any of the commands from 19486 to 19630, or at 19678, where the command being traced is executed. To stop while typing out, press the Stop key eleven times in order to ensure that the indicators on the machine are reset. Executing the command 271991419913 resets the indicators.

Linkage to this routine is 1619581@@@@@
4919486

where @@@@@ is the address of the start command.

To stop tracing on the execution of a particular branch command, place a flag in the 01 position of that branch command. The program then executes without tracing.

A Branch, BT, or BTM command that contains a flag in either P_3, P_4, or P_5 results in having the trace branch to the 4, 3, or 2 digit location defined by that flagged field. If the trace "hangs up," the last branch command printed out should be checked for a flag in one of these P-field locations. A flag in the P_6 position of a branch command will cause an error as this is considered a negative number.

To eliminate typeout on a sense switch, use the following changes and additions:

```
T2030       BNZ SENSE,,,CHECK FOR TYPE                  -19766    4719466 12
T2281       DORG19466,,,                                -
T2290SENSE  BC4 OUT&72,,,IF SENSE SWITCH 4 NO TYPE OUT  -19466    4619882 04
T2300       B   OUT,,, TYPE OUT                          -19478    4919810 1619

NUMERICAL LISTING FOR TRACE AND COUNT OF BRANCHES ROUTINE

1619521  -004719522 11  3219521  -004719546 12  3219520                J9486   3
261998619581161957 6J967825-     -    1119576   -11119581   -1          J9546   3
1419576J96904719570 12  3219678     1419679   M24619698 11             J9606   3
27199141991341          4919486 3219680-    2619994196841619            J9666   3
684J97424919666 261958119994241970919986 4719810 12  1119980          J9726   3
  -1271991419913 4919486 3819976 01  34     .01 23219685               J9786   3
261999919689261998919679351998 2 01  1619980-0001261970919986          J9846   3
4919790 1419922-1 -2441995019520 1419946-1 -14419974195211419           J9906   3
970-2 -142-0001Z     -                                                  J9966   3
```

```
  SPS         LISTING                                          ADDRESS COMMAND
  T1001       DORG19486,,,                                -
  T1010START  TFM IND&11,000,9, CLEAR FLAGS IN INDICATORS-19486    1619521  -00
  T1020       BNH *&24,,,                                 -19498    471952201100
  T1030IND    SF  IND&11,000,9, SET HIGH POSITIVE         -19510    3219521  000
  T1040       BNZ *&24,,,                                 -19522    471954601200
```

```
T1050       SF   IND&10,,, SET EQUAL ZERO                    -19534   3219520
T1060       TF   TYLO,TRAND&11,,STORE ADDRESS                -19546   261998619581
T1070       TFM *&18,EXECU,7, RESET TRANSFER AREA            -19558   1619576J9678
T1080TRAND  TD   EXCU,,, TRANSMIT DIGIT BY DIGIT             -19570   250    0
T1090       AM   *-6,01,10, STEP BY ONE                      -19582   1119576   -1
T1100       AM   *-13,01,10, STEP BY ONE                     -19594   1119581   -1
T1110       CM   TRAND&6,EXECU&12,7, TEST TRANSFER           -19606   1419576J9690
T1120       BNZ  TRAND,,, LOOP TO FINISH TRANSFER            -19618   471957001200
T1130       SF   EXECU,,,                                    -19630   3219678
T1140       CM   EXECU&1,42,10,,                            -19642   1419679   M2
T1150       BH   BBB,,, BRANCH IF BRANCH POSSIBLE            -19654   461969801100
T1160       BT   SETIND,SETIND-1,, SET INDICATORS           -19666   271991419913
T1170EXECU  NOP  ,,, COMMAND PLACED HERE                     -19678   41
T1180       B    START,,,                                   -19690   4919486 3219
T1181       DORG*-3,,,                                      -
T1190BBB    SF   EXECU&2,,, STAD STORED HERE                 -19698   32196800
T1200       TF   PFIELD,EXECU&6,, STORE P FIELD              -19710   261999419684
T1210       TFM  EXECU&6,BBBI,7, SET NEW BRANCH              -19722   1619684J9742
T1220       B    EXECU-12,,, SEE IF COMMAND BRANCHES         -19734   4919666 2619
T2001       DORG19742,,,                                    -
T2010BBI    TF   TRAND&11,PFIELD,,SET NEW ADDRESS           -19742   261958119994
T2020       C    STAD,TYLO,, CHECK OLD BRANCH                -19754   241970919986
T2030       BNZ  OUT,,, TYPE OUT IF THEY DO NOT COMPARE      -19766   471981001200
T2040       AM   COUNT,01,10,COUNT BRANCH                    -19778   1119980   -1
T2050       BT   SETIND,SETIND-1,,SET INDICATORS             -19790   271991419913
T2060       B    START,,,                                   -19802   4919486 3819
T2061       DORG*-3,,,                                      -
T20700UT    WNTYCOUNT-4,,,TYPE OUT NUMBER OF TIMES          -19810   3819976 01
T2080       RCTY,,,        BRANCH WAS EXECUTED              -19822   34      01 2
T2090       SF   EXECU&7,,,SET FLAG IN Q FIELD               -19834   3219685
T2100       TF   PFIELD&5,EXECU&11,, TRANSFER Q FIELD        -19846   261999919689
T2110       TF   PFIELD-5,EXECU&1,, TRANSFER OP CODE        -19858   261998919679
T2120       DN   19982,,,NEW BRANCH TYPED OUT                -19870   3519982 01
T2130       TFM  COUNT,00002,7,RESET COUNT                   -19882   1619980-0001
T2140       TF   STAD,TYLO,,SET ADDRESS OF NEW BRANCH        -19894   261970919986
T2150       B    OUT-20,,,                                  -19906   4919790 1419
T2151       DORG*-3,,,                                      -
T2160SETINDCM  *&8,01002,710, TURN OFFINDICATORS             -19914   1419922-1 -2
T2170       BNF  *&24,IND&10,, CHECK EQUAL ZERO              -19926   441995019520
T2180       CM   *&8,01001,710, SET EQUAL ZERO               -19938   1419946-1 -1
T2190       BNF  *&24,IND&11,,CHECK HIGH POSITIVE            -19950   441997419521
T2200       CM   *&8,02001,710,SET HIGH POSITIVE             -19962   1419970-2 -1
T2210       BB   ,,,     INDICATORS NOW RESET                -19974   42-0001Z
T2220COUNT  DC   5,00001,19980,                              -19986    Z
T2230PFIELDDS  5,19994,,  P FIELD OF BRANCH                  -
T2240TYLO   DS   5,19986,,  ADDRESS OF COMMAND               -
T2250       DC   1,@,19987,     RECORD MARK                  -
T2260       DC   1,@,19981,     RECORD MARK                  -
T2270STAD   DS   5,BBB&11,,                                  -
T2280       DENDSTART,,,
```

3. *Flow Trace*

Assembly

The flow trace program is supplied in symbolic form with comments included. It occupies 631 positions of core storage and may be assembled anywhere in storage.

Operation

To use the flow trace, the programmer should first load his object program, then load the assembled flow trace,

using a 360000000300 instruction. When the trace has been read in, any data required by the object program should be placed in the reader. Depress Start; this results in a branch to the first instruction of the trace. This instruction is RNTY; the programmer must type in $\overline{\text{XXXXX}}$, where this address is the starting address of his program. He then depresses Release and Start and his program executes under trace control. Whenever a branch actually occurs in the object program, the address of this branch instruction is typed out at the left margin, followed by the address to which it branched.

If, the programmer wishes for any reason to stop tracing his program, he must press the Stop key when the trace is typing out one of the branched-to addresses. He may then depress Insert and type in a branch to this address. When he depresses Release and Start, his program executes normally without trace control.

NOTE: The programmer must not discontinue the trace in the middle of a subroutine linked to his main program by a BT or BTM instruction and a BB instruction. If he does, the BB will execute improperly because the trace simulates, but does not execute BT, BTM, or BB instructions.

To resume tracing, the programmer must interrupt his program and branch to the RNTY instruction previously referenced.

Flow Trace: Sample Assembly

```
       DORG 18000                        18000
* READ STARTING ADDRESS INTO TRACE
BEGIN  RNTY STRT&31                      18000 36 18103 00100
* STORE HP AND EZ TRIGGER SETTINGS
STTRIG TDM  SET&9                        18012 15 18357 00000
       BE   STRT                         18024 46 18072 01200
       TDM  SET&9,1                      18036 15 18357 00001
       BH   STRT                         18048 46 18072 01100
       SF   SET&9                        18060 32 18357 00000
* SAVE ADDRESS OF INSTRUCTION BEING TRACED
STRT   TF   WRT1&6,*&35                  18072 26 18434 18107
       TFM  *&18,1                       18084 16 18102 J8360
* MOVE INSTRUCTION INTO TRACE PROGRAM
       TD                                18096 25 00000 00000
       AM   *-1,1                        18108 11 18107 -0001
       AM   *-18,1                       18120 11 18102 -0001
* TEST FOR WRAP AROUND MEMORY
       CM   STRT&35,20000                18132 14 18107 K0000
       BL   *&24                         18144 47 18168 01300
TFM    STRT&35                           18156 16 18107 -0000
       CM   STRT&30,1&12                 18168 14 18102 J8372
       BL   STRT&24                      18180 47 18096 01300
* SET AND CLEAR FLAGS ON INSTRUCTION
       SF   I                            18192 32 18360 00000
       CF   I&1                          18204 33 18361 00000
       SF   I&2                          18216 32 18362 00000
       CF   I&3                          18228 33 18363 00000
       CF   I&4                          18240 33 18364 00000
       CF   I&5                          18252 33 18365 00000
       CF   I&6                          18264 33 18366 00000
* TEST FOR BRANCH INSTRUCTIONS
       CM   I&1,42,10                    18276 14 18361 000M2
       BL   TWO                          18288 47 18504 01300
* HANDLE BRANCH BACK
       BE   ONE                          18300 46 18472 01200
* SET UP TRAP FOR EXECUTED BRANCHES
       TF   REPL&11,I&6                  18312 26 18391 18366
       TFM  I&6,REPL                     18324 16 18366 J8380
       TFM  I&18,STRT                    18336 16 18378 J8072
* SET HP AND EZ TRIGGERS
SET    AM   *&9,,810                     18348 11 18357 0-0-0
```

```
* LOCATION OF OBJECT INSTRUCTION IN TRACE
I       NOP                              18360 41 00000 00000
        B                                18372 49 00000 00000
        DORG *-3                         18380
* GIVE TRACE BRANCHED TO ADDRESS
REPL    TFM  STRT&35                     18380 16 18107 -0000
        RCTY                             18392 34 00000 00102
* WRITE OUT ADDRESS OF EXECUTED BRANCH
        WNTY WRT1&2                      18404 38 18430 00100
        TF   WRT1&6,STRT&35              18416 26 18434 18107
WRT1    SPTY                             18428 34 00000 00101
        DC   1,@,WRT1&7                  18435 00001    Z
        SPTY                             18440 34 00000 00101
* WRITE OUT ADDRESS OF INSTRUCTION BRANCHED TO
        WNTY WRT1&2                      18452 38 18430 00100
        B    STRT                        18464 49 18072 00000
        DORG *-3                         18472
* SIMULATE BRANCH BACK
ONE     TFM  STRT&35                     18472 16 18107 -0000
        TFM  *-1,90000                   18484 16 18483 R0000
        B    REPL&12                     18496 49 18392 00000
        DORG *-3                         18504
* ARRANGE TO STORE TRIGGER SETTINGS
TWO     TFM  1&18,STTRIG                 18504 16 18378 J8012
* TEST FOR BTM
        CM   I&1,17,10                   18516 14 18361 000J7
        BE   THREE                       18528 46 18564 01200
* TEST FOR BT
        CM   I&1,27,10                   18540 14 18361 000K7
* EXECUTE NON-BT,BTM INSTRUCTIONS
        BNE  SET                         18552 47 18348 01200
* SAVE RETURN ADDRESS
THREE   TF   ONE&11,STRT&35              18564 26 18483 18107
*SIMULATE BRANCH
        TF   STRT&35,I&6                 18576 26 18107 18366
        SM   I&6,1                       18588 12 18366 -0001
* ARRANGE TO TRANSMIT FIELD
        TDM  I&1,6                       18600 15 18361 00006
        TFM  I&18,REPL&12                18612 16 18378 J8392
        B    I                           18624 49 18360 00000
        DEND BEGIN                       18000
END OF PASS II
```

4. *Fast Trace*

Whenever a branch is executed, this trace prints out:

AAAAA PPPPP CCC,

where AAAAA is the address of the branch command, PPPPP is the P field of the command, and CCC is the number of times the branch was executed without another branch command occurring. CCC is printed out, only if it is greater than 1.

If a WN is encountered, the following is printed out:

38PPPPP XXXXXXX . . . XXXXX,

where XXXXX is the record at PPPPP. In other words, all output in the program is typed out on the typewriter.

Linkage to this routine is 1616095@@@@@
4916000

where @@@@@ is the starting address.

If the P field of a branch command is negative, the routine "hangs up" with RR in the operation code. The routine also does not work if P_3, P_4, or P_5 of a Branch command is flagged. The trace routine also simulates BT, BTM, and BB commands. However, it does not print out when these commands are executed, but continues tracing the program in the BT or BTM subroutine.

The routine is located from 16000 to 16744, with record marks at 16744, 16738, and 16599. See Figure 3 for a flow chart of this program.

```
SPS         LISTING                                                ADDRESS COMMAND
01000       DORG16000,,,
01010START  TDM SET&11,2,,RECORD INDICATORS                        16000   1516287    2
01020       BNH *&24,,,                                            16012   4716036 11
01030       TDM SET&11,0,,SET HP                                   16024   1516287   10
01040       BNZ *&24,,,                                            16036   4716060 12
01050       TDM SET&11,1,,SET EZ                                   16048   1516287    1
01060BEGIN  TF  TYLO,TRAND&11,,STORE ADDRESS OF COMMAND            16060   261673716095
01070       TFM *&18,EXECU,7, RESET TRANSFER AREA                  16072   1616090J6288
01080TRAND  TD  ,,, TRANSMIT DIGIT BY DIGIT                        16084   25
01090       AM  TRAND&6,01,10, STEP BY ONE                         16096   1116090   01
01100       AM  TRAND&11,01,10, STEP BY ONE                        16108   1116095   01
01110       CM  TRAND&6,EXECU&12,7, TEST TRANSFER                  16120   1416090J6300
01120       BNZ TRAND,,, LOOP TO FINISH                            16132   4716084 12
01130       CF  EXECU&1,,,  STAD IN Q FIELD                        16144   33162890   0
01140       SF  EXECU,,,                                           16156   3216288
01150       CM  EXECU&1,17,10, TEST FOR BTM                        16168   1416289   J7
01160       BE  BBTM,,,                                            16180   4616328 12
01170       CM  EXECU&1,27,10, TEST FOR BT                         16192   1416289   K7
01180       BE  BBT,,,                                             16204   4616308 12
01190       CM  EXECU&1,38,10, TEST FOR WN                         16216   1416289   L8
01200       BE  OUTPUT,,,                                          16228   4616416 12
01210       CM  EXECU&1,42,10, TEST FOR BB                         16240   1416289   M2
01220       BE  BBACK,,,                                           16252   4616396 12
01230       BH  BRANCH,,, B,BD,BNR,BNF,BI,BNI,H,                   16264   4616588 11
02010SET    CM  *&8,01002,710, RESET HP AND EZ                     16276   141628401 02
02020EXECU  NOP ,,, COMMAND STORED HERE                            16288   41
02030       B   START,,, CYCLE TRACE ROUTINE                       16300   4916000 1616
02040       DORG*-3,,,
02050BBT    TFM EXECU&1,26,10 SIMULATE BT                          16308   1616289   K6
02060       B   *&20,,,                                            16320   4916340 1616
02070       DORG*-3,,,
02080BBTM   TFM EXECU&1,16,10, SIMULATE BTM                        16328   1616289   J6
02090       SF  EXECU&2,,,                                         16340   32162900   0
02100STBB   DS  5,*,, STORE RETURN FOR
02110       TF  STBB,TRAND&11,,BB COMMAND                          16352   261635116095
02120       TF  TRAND&11,EXECU&6,, EXECUTE BRANCH                  16364   261609516294
02130       SM  EXECU&6,01,10,                                     16376   1216294   01
02140       B   SET,,,                                            16388   4916276 2616
02150       DORG*-3,,,
02160BBACK  TF  TRAND&11,STBB,,SIMULATE BB                         16396   261609516351
02170       B   BEGIN,,,                                          16408   4916060 1516
02180       DORG*-3,,,
02190OUTPUTTDM EXECU&9,1,, SET FOR TYPEWRITER                      16416   1516297    1
02200       TD  EXECU&7,TYLO&1,,SET RECORD MARK                    16428   251629516738
02210       RCTY,,,                                                16440   34       01 2
02220       WNTYEXECU,,,                                           16452   3816288 01
02230       SPTY,,,                                                16464   34       01 1
02240       B   SET,,,                                             16476   4916276 3400
03010       DORG*-3,,,
03020TYPE   SPTY,,,                                                16484   34       01 1
03030       WNTYCOUNT-2,,,NUMBER OF TIMES                          16496   3816596 01
03040TY1    RCTY,,,                                                16508   34       01 2
03050       WNTYTYLO-4,,, ADDRESS OF BRANCH                        16520   3816733 01
03060       SPTY,,,                                                16532   34       01 1
03070       WNTYPFB-4,,, PFIELD BRANCH                             16544   3816739 01
03080       TF  STAD,TYLO,, STORE NEW BRANCH ADDRESS               16556   261615516737
03090       TFM COUNT,001,9, RESET COUNT                           16568   1616598  001
03100       B   BEGIN,,,                                           16580   4916060 3216
03110       DORG*-3,,,
03120BRANCHSF  EXECU&2,,,COUNT IN Q FIELD                          16588   3216290 001Z
03130       TF  PFB,EXECU&6,, STORE PFIELD                         16600   261674316294
03140       TFM EXECU&6,BBB,7, SET NEW PFIELD                      16612   1616294J6632
```

```
03150        B    SET,,,                                  16624    4916276 2616
03160        DORG*-3,,,
03170BBB     TF   TRAND*11,PFB,, SIMULATE BRANCH          16632    261609516743
03180        BNF  *&14,PFB,,TEST IF PFIELD NEGATIVE       16644    441665816743
03190        DC   2,-99,, OR IF INDIRECT ADDRESS          16656    RR2416155167
03200        C    STAD,TYLO,,TEST IF OLD BRANCH           16658    241615516737
03210        BNZ  BBBI,,,IS THE SAME AS NEW ONE           16670    4716702 12
03220        AM   COUNT,01,10, STEP COUNT BY ONE          16682    1116598   01
03230        B    BEGIN,,,                                16694    4916060 1416
04010        DORG*-3,,,
04020BBBI    CM   COUNT,001,9 IF BRANCH ONLY              16702    1416598  001
04030        BZ   TY1,,, EXECUTED ONE TIME                16714    4616508 12
04040        B    TYPE,,, DO NOT TYPE COUNT               16726    49164840   0
04050TYLO    DS   5,*,,                                   16738    Z0   0Z
04060        DC   1,@,,
04070PFB     DS   5,,,
04080        DC   1,@,,
04090COUNTDS      3,BRANCH&10,,
04100        DC   1,@,BRANCH&11,,
04110STAD    DS   5,TRAND&71,,
04120        DENDSTART,,,
```

Fast Trace Tape Listing

The following is a listing of the tape used to load the fast trace routine. Loading halts on a 48 operation code; if Start is depressed, the typewriter is then ready to accept a 5-digit address to start the trace. Release and Start are then depressed.

```
41000000000036000260030036000000030049000120000Z
16000Z
151628700002471603601100151628700000471606001200151628700001
261673716095161609OJ62882500000-00001116090000-11116095000-1
1416090J6300471608401200L316289-0000L216288000001416289000J7
461632801200141628900OK7461630801200141628900OL8461641601200
1416289000M24616396012004616588011001416284-10-2410000000000
491600001616289000K6491634001616289000J63216290-000026163511
609526160951629412162940OO-14916276026160951635149160600151 6
297000012516295167383400000001023816288001003400000001014916
276034000000010138165960010034000000010238167330010034000000
010138167390010026161551673716165980O-014916060032162900-01Z
16600Z
261674316294161629 4J6632491627602616095167434416658167 43RR24
161551673747167020120011165980O0-149160600141659800-01461650
8012004916484-0000Z
16739Z
-0000Z
00000Z
3616091001003216091000004916000000004800000000004900000000005
```

Load tape with 360000000300.

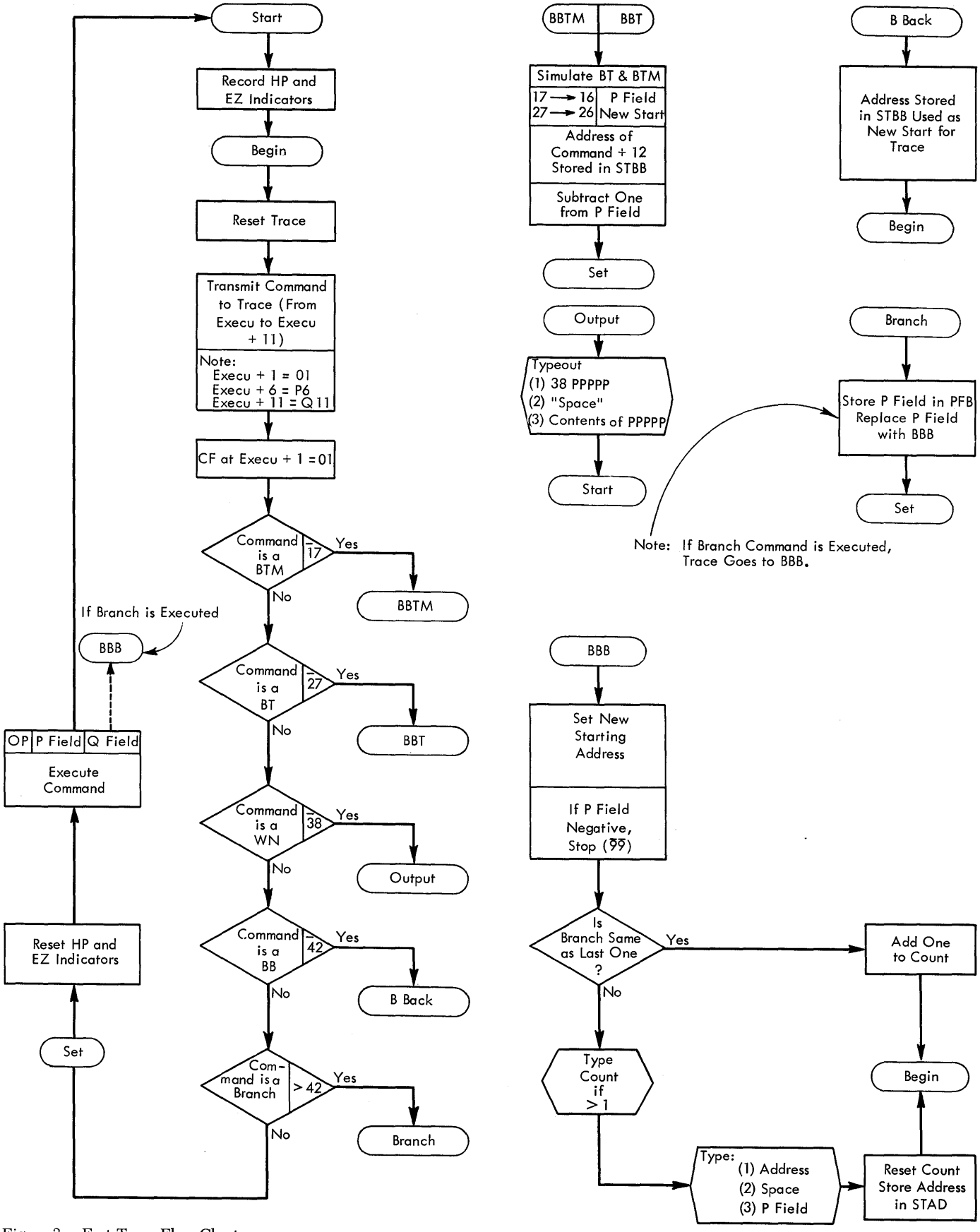

Figure 3. Fast Trace Flow Chart

Appendix C

Three Utility Programs

Three frequently used programs are:

	Program	*Location*
1.	Load Routine	From 402 to 616
2.	Punchout Routine (compatible with Load Routine)	From 718 to 842
3.	Modification and Typeout Routine	From 618 to 716

Load Routine

This routine loads a 21-digit record; the record mark is the twenty-first digit. The first 5 digits are the address of the command to be loaded; the next 3 digits are an indicator to the load program; the next 12 digits are the command or numerical information to be loaded into the specified address and higher locations. The twenty-first digit is not loaded. If the indicator area contains a numerical 000 (or blanks) then the command is loaded. However, it is possible to execute the command instead of loading it. It is also possible to type out the record.

```
SPS          LISTING                                          ADDRESS COMMAND
L1001        DORG00402,,,                                    -00
L1010LOAD    RNPTEXAM,,, READ IN RECORD                      -00402   36  590 03
L1020        BD  TYPE,EXAM&5,, BRANCH TO TYPE                -00414   43  542  595
L1030        BD  EXECUT,EXAM&6,,BRANCH TO EXECUTE            -00426   43  574  596
L1040        SF  EXAM,,,                                     -00438   32  590
L1050        TF  TRAN&6,EXAM&4,, SET ADDRESS                 -00450   26  468  594
L1060TRAN    TD  ,EXAM&8,7,TRANSMIT DIGIT BY DIGIT           -00462   25-     -0598
L1070        AM  TRAN&6,01,10,STEP TRANSMIT                  -00474   11  468   -1
L1080        AM  TRAN&11,01,10,                              -00486   11  473   -1
L1090        CM  TRAN&11,EXAM&20,7,TEST FOR END              -00498   14  473-0610
L1100        BN  TRAN,,, LOOP TILL FINISHED                  -00510   47  462 13
L1110        TFM TRAN&11,EXAM&8,7, RESET LOAD ROUTINE        -00522   16  473-0598
L1120        B   LOAD,,, START OVER                          -00534   49  402 34
L1121        DORG*-3,,,                                      -00
L1130TYPE    RCTY,,,                                         -00542   34      01 2
L1140        WNTYEXAM,,,TYPE OUT RECORD                      -00554   38  590 01
L1150        B   LOAD&24,,,CONTINUE ROUTINE                  -00566   49  426 16
L1151        DORG*-3,,,                                      -00
L1160EXECUTTFM EXAM&21,49,10, SET BY REMOVING RMK            -00574   16  611   M9
L1170        NOP ,,,EXAM EQUALS 590                          -00586   41
L1180        NOP ,,, COMMAND PLACED HERE                     -00598   41
L1190        B   LOAD,,, START OVER                          -00610   49  402 16
L1200EXAM    DS  2,EXECUT&16,,                               -00
L1210        DENDLOAD,,,                                     -00
```

Indicator Code	*Result*
000	Load information from last 12 digits
100	Type out and load
010	Execute command (not loaded)
110	Type out and execute (not loaded)

EXAMPLES

120120003400000001O2@	This loads 340000000102 into 12012 to 12023
000000103600100003O0@	The command 360010000300 is executed and the record following on the tape goes into 00100. By this method, continuous records such as tables may be loaded.
000001103600100003O0@	Same as above except that this record is typed out.
000000104800000000O0@	This halts loading of tape.
000000101511111000O@	This loads a record mark into 11111.
000000104912012000O0@	This executes a branch to 12012.

Modification and Typeout Routine

To enter numerical data into core storage from the typewriter, branch to 00638. The typewriter is then ready to receive the 5-digit address of the location which is to be changed. After typing in the address, press Release and Start. The typewriter then spaces twice and is ready to receive the information to be put

into that address. After this information has been typed in, press Release and Start. The carriage is then returned and the typewriter is ready for a new address to be typed in.

To type out numerical data beginning at some address, branch to 00618. The same procedure described above occurs, except that the data is typed out rather than entered.

```
SPS         LISTING                                          ADDRESS COMMAND
MT001       DORG00618,,,                                    -00
MT010TYP    TFM MT&1,38,10, SET FOR TYPE OUT                -00618   16  699   L8
MT020       B   *&20,,,                                     -00630   49  650 16
MT021       DORG*-3,,,                                      -00
MT030MODIFYTFM MT&1,36,10,SET FOR TYPE IN MODIFY            -00638   16  699   L6
MT040       RCTY,,,                                         -00650   34      01 2
MT050       RNTYMT&3,,,  SET (TYPE IN) ADDRESS              -00662   36  700 01
MT060       SPTY,,,                                         -00674   34      01 1
MT070       SPTY,,,                                         -00686   34      01 1
MT080MT     NOP ,00100, TYPE IN OR OUT HERE                 -00698   41      01
MT090       B   MODIFY&12,,,LOOP ROUTINE                   -00710   49  650 16
MT100       DENDTYP,,,
```

Punchout Routine

To enter this routine, branch to 00718. The routine first punches out the load routine and the other two utility programs. The typewriter is then ready to receive the address of the record to be punched out. After this 5-digit address has been entered, press Release and Start. The routine punches out the load command for the record and then the record itself. The carriage then returns and the typewriter is ready to receive a new address. To punch out a record without having the load program punched out, branch to 00742.

To load a tape made this way, use 360040200300
490040200000.

```
SPS         LISTING                                          ADDRESS COMMAND
P1001       DORG00718,,,                                    -00
P1010PUNCH TFM EXAM&21,49,10, REMOVE RECORD MARK            -00718   16  611   M9
P1020       WNPTLOAD,,, PUNCH OUT LOAD ROUTINE              -00730   38  402 02
P1030PUN    RCTY,,,                                         -00742   34      01 2
P1040       RNTYPI&10,,,TYPE IN ADDRESS OF PUNCH OUT        -00754   36  832 01
P1050       SF  PI&10,,,                                    -00766   32  832
P1060       WNPTPI,,,PUNCH OUT INSTRUCTION FOR LOADING      -00778   38  822 02
P1070       TF  *&18,PI&14,,                        RECORD  -00790   26  808  836
P1080       WNPT,,,  PUNCH OUT RECORD                       -00802   38-      02
P1090       B   PUN,,,LOOP FOR NEXT RECORD TO PUNCH         -00814   49  742 0000
P1100PI     DS  2,*-3,,                                     -00826   011 36     0
P1110       DC  21,000000110360000000300@,*&17,             -00838   0300Z
P1120       DENDPUNCH,,,
```

```
NUMERICAL LISTING COLUMNS 1 TO 60 FOR LOAD, MODIFY, AND PUNCH ROUTINES

36  590 03  43  542  59543  574  59632  590     26  468  594          -0402   1
25-    -059811  468   -111  473   -114  473-061047  462 13           -0462   1
16  473-059849  402 34      01 238  590 01  49  426 16  611          -0522   1
  M941          41          49  402 16  699   L849  650 16           -0582   1
699   L634      01 236  700 01  34      01 134      01 141           -0642   1
    01  49  650 16  611   M938  402 02  34      01 236  832          -0702   1
01  32  832     38  822 02  26  808  83638-     02  49  742          -0762   1
0000011 36-    00300-                                               -0822   1
```

Appendix D

Check Routine for Flags and Record Marks

This routine checks to see if there is a flag in the first four positions of the Q field of any immediate command. If there is no flag, it prints out the address of the command. The commands are assumed to be in intervals of 12; any operation code less than 19 and not equal to 15 is considered to need a flag.

Linkage to flag routine is 1619518@@@@@
1619763 # # # # #
4919500

where @@@@@ is the address of the start command and # # # # # is the address of the last command.

To have the addresses of all the record marks in core storage printed out, branch to 19810.

These two routines occupy locations 19500 to 19931.

```
SPS         LISTING                                          ADDRESS COMMAND
F1001       DORG19500,,,                                     -
F1010START  WATYCOM,,, NO FLAG AT PRINTOUT                   -19500   3919785 01
F1020       SF  ,,, BEGINING ADDRESS IN P FIELD              -19512   32-
F1030       TF  OUT,START&18,, SET ADDRESS                   -19524   261978219518
F1040       AM  START&18,01,10, STEP ONE                     -19536   1119518   -1
F1050       TF  F1&6,START&18,,                              -19548   261959019518
F1060       TF  F2&6,START&18,, OPERATION FIELD              -19560   261961419518
F1070       AM  START&18,10,10, STEP TO Q11                  -19572   1119518   J0
F1080F1     CM  ,15,10,  TEST FOR TDM COMMAND                -19584   14-       J5
F1090       BZ  SET,,, STEP TO NEXT COMMAND                  -19596   4619740 12
F1100F2     CM  ,18,10, TEST FOR NON IMMEDIATE               -19608   14-       J8
F1110       BH  SET,,, STEP FOR NEXT COMMAND                 -19620   4619740 11
F1120       TF  F3&11,F1&6,, TRANSFER 01                     -19632   261966719590
F1130       AM  F3&11,06,10,, STEP TO Q7 POSITION            -19644   1119667   -6
F1140F3     BNF *&24,,, TEST FOR FLAG                        -19656   4419680-
F1150       B   SET,,, IF FLAG BRANCH FOR NEXT COMMAND        -19668   4919740
F1160       AM  F3&11,01,10, NO FLAG STEP Q POSITION         -19680   1119667   -1
F1170       C   START&18,F3&11,, TEST FOR Q11                -19692   241951819667
F1180       BP  F3,,,LOOP IF NOT Q11                         -19704   4619656 11
F1190       WNTYOUT-4,,,NO FLAG IN Q7 TO Q10                 -19716   3819778 01
F1200       SPTY,,,                                          -19728   34      01 1
F1210SET    AM  START&18,01,10, STEP TO NEXT COMMAND         -19740   1119518   -1
F1220       CM  START&18,,, TEST FOR END IN Q FIELD          -19752   1419518-
F1230       BNP START&12,,,CONTINUE                          -19764   4719512 11
F1240       H   ,,,                                          -19776   48-    Z5556
F1250OUT    DS  5,*-5,,   ADDRESS LOCATION                   -19788   004653414700
F1260       DC  1,@,*-4,                                     -19800   4163000Z
F1270COM    DAC 12,NO FLAG AT @,*-2,                         -
F1280       DENDSTART,,,


R1001       DORG19810,,,                                     -
R1010RSTARTWATYWRI,,,      RMS AT                            -19810   3919927 01
R1020       BNR *&48,00000,7, STEP NO RM                     -19822   4519870-0000
R1030       TF  ADD,*-1,,  SET ADDRESS OF RM                 -19834   261992419833
R1040       WNTYADD-4,,,           TYPE OUT ADDRESS          -19846   3819920 01
R1050       SPTY,,,                                          -19858   34      01 1
R1060       AM  RSTART&23,01,10, STEP LOCATION BY 1          -19870   1119833   -1
R1070       CM  RSTART&23,20000,7, TEST FOR END              -19882   1419833K0000
R1080       BNZ RSTART&12,,,    LOOP                         -19894   4719822 12
R1090       TDM RSTART&20,0,11,         RESET                -19906   1519829    -
R1100       H   ,,,                                          -19918   48     Z5954
```

```
R1110ADD    DS  5,*-5,, ADDRESS LOCATION                -19930   620041630002
R1120       DC  1,@,*-4,  RECORD MARK                    -
R1130WRI    DAC 8,RMS AT @,*-2,                          -
R1140       DENDSTART,,,

NUMERICAL LISTING FOR FLAG AND RECORD MARK CHECK ROUTINE

3919785 01  32-          2619782195181119518   -1261959019518          J9500    4
2619614195181119518   J014         J54619740 12  14-        J8         J9560    4
4619740 11  2619667195901119667   -64419680-    4919740                J9620    4
1119667   -12419518196674619656 11  3819778 01  34      01 1          J9680    4
1119518   -11419518-    4719512 11  48-    Z5556004653414700           J9740    4
41630002  3919927 01  4519870-000026199241983338199 20 01  34          J9800    4
      01 11119833   -11419833K00004719822 12  1519829    -48          J9860    4
     Z595462004163000-                                                 J9920    4
```

Appendix E

For Card Input-Output

The following two cards, loaded at 00000, set all unflagged zeros, occurring after 400, to blank spaces for a card dump.

```
41          41          3600000 05
26   18   4715         @11   47   -143   24-040126   71   4744          49   24
```

The following card loads a deck that has been dumped from core storage by a 350000000400. Replace first card in deck with this one.

```
36   80 05  36  160 05  36  240 05  36-0320 05  11   42   Q049   36      LLLLLL
```

The following 12-card dump routine dumps 60 characters per card with the address contained in columns 71 to 75. The dumping begins at 00380. A load routine including the tables is punched out first, so that the resulting deck is ready to be reloaded at any time. Place cards in hopper and press Load. The program uses 00000 to 00091 plus a buffer area at 19840 to 19919.

This routine starts dumping at the address located in the Q field of the first command on the last card. The address must be flagged.

```
3619840 05  3819840 04  11   47   -114   44-7 -04700000 12  3600000 05
3619840 05  26   301991931      198404900000
0000000000001020304000204060800030609021004080216100500151020060218142007042-100
1128200806142230090817263000000000050607080900121416181518112427202428223632-175
5203530454036324844553249465360484654627544536271801234567891234567892-23456-250
789-J3456789-JK456789-JKL56789-JKLM6789-JKLMN789-JKLMNO89-JKLMNOP9-JKLMNOPQ-325
3619840 05                          41          41          36   12 05      -000
26   301991425      J984011   30   -111   35   -143   24   3416   35J98404900000
3619840 05  16    6-00004900000                                          4900000
                                                                @@@@@@@@@@@@@@@@@@@@
2619914   2349   60-0380                                        36   12 05
25J9840-038011   18   -111   23   -143   12   1738J9840 04  16   18J98404900000
```

THE FOLLOWING CARDS ARE THE SAME AS THE ABOVE EXCEPT THAT THEY USE A BUFFER LOCATED FROM 00840 TO 00919.

```
3600840 05  38  840 04  11   47   -114   44-7 -04700000 12  3600000 05
36  840 05  26   30  91931       8404900000
0000000000001020304000204060800030609021004080216100500151020060218142007042-100
1128200806142230090817263000000000050607080900121416181518112427202428223632-175
5203530454036324844553249465360484654627544536271801234567891234567892-23456-250
789-J3456789-JK456789-JKL56789-JKLM6789-JKLMN789-JKLMNO89-JKLMNOP9-JKLMNOPQ-325
36  840 05                          41          41          36   12 05      -000
26   30  91425      -084011   30   -111   35   -143   24   3416   35-08404900000
36  840 05  16    6-00004900000                                          4900000
                                                                @@@@@@@@@@@@@@@@@@@@
26  914   2349   60-0380                                        36   12 05
25-0840J946011   18   -111   23   -143   12   1738J9840 04  16   18-08404900000
```

THE FOLLOWING CARDS PLACED ON THE BACK OF A DECK DUMPED BY THE ABOVE ROUTINE WILL ALLOW THE USER TO TYPE OUT COMMENTS AND THEN HALT AND PROCEED TO THE START OF THE PROGRAM WHEN THE START KEY IS PRESSED.

```
11   30   -111   35   -143   24   343600000 05  4900000                -0036
3719841 05  3919841 01  34      01 23600000 05  4900000
ALPHAMERIC    COMMENT ONE@
3719841 05  3919841 01  34      01 23600000 05  4900000
ALPHAMERIC    COMMENT TWO@
3719841 05  3919841 01  34      01 23600000 05  4900000
ALPHAMERIC    COMMENT THREE@         ETC.
48          49START                           4900000
```

Note that each comment has a standard card preceding it; therefore, there is no limit to the number of comments to be read into location 19841 and then typed out. Location 19841 may be changed by the user. Use only the first and last card for no typeout.

IBM 1620 Printed in U. S. A. J26-5547-0

IBM®
International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York