COMPUTER TECHNOLOGY
DR. JOHN MANIOTES
COMPUTER TECHNOLOGY DEPT.
PURDUE UNIVERSITY
CALUMET CAMPUS
HAMMOND, IN 46323

LESS (Scheduling Portion)

LESS (Scheduling Portion)

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

Direct Inquiries To: Ray N. Sauer
IBM Corporation
2601 South Main
Houston 2, Texas

CHANGES TO 1620 LESS PROGRAM FILE NO. 10.3.003 - November 3, 1961

On page 4 A in Table 2, there are two numbers in column TI (N) for the row where N=3. Instead of 7, this row should be 9.

On page 8 the sixth line from the bottom reads 7 - 11, etc. It should read 7 - 10 time duration of the job - D (I, J).

On page 10 the last line should read, "several type 2 and 3 error messages however."

Where this program is run with the overflow switch on stop, as the writeup specifies, it will stop at location 1390 when executing a compare instruction. To correct this change two instructions on page 19.

Location	Old Instruction	New Instruction
01390	C Test, Most	C Most, Test
01402	BH Error 4	BL Error 4

To make this change in the object program listing on page 24, change the first 24 columns of card number 20 (columns 79-80) from:

240326303240460242601100 to
240324003263470242601300.

All decks mailed after November 15, 1961, have been corrected.

A2

PART I - INTRODUCTION

Least Cost Estimating and Scheduling (LESS) refers to a management science technique for analyzing certain business projects. The three phases of this analysis are (1) planning, (2) scheduling, and (3) determining project cost to completion time relationships. The first two phases, commonly called arrow diagram planning and critical path scheduling, are also the basis of many similar business management methods such as the Navy's Program Evaluation and Review Technique (PERT) and the Air Force's Program Evaluation Procedure (PEP).

This report states the rules for constructing an arrow diagram, and describes an IBM 1620 (Card System) program for scheduling. Many improvements have been made over the 1620 (paper tape system) scheduling program (file 10.3.002). There are no restrictions on numbering of jobs (except all numbers are three digits) or on the order of input cards. For a 20K computer, the sum of jobs and nodes may be as high as 1672.

TABLE OF CONTENTS

A4

FILE NO. ABSTRACT

10.3.003 IBM 1620-LESS *LEAST-COST ESTIMATING AND SCHEDULING**SCHEDULING PORTION*-*CARD*
AVAILABLE 1ST QUARTER 1962

AUTHOR- RAY N. SAUER-IBM

DIRECT INQUIRIES TO
RAY N. SAUER
IBM
2601 SOUTH MAIN
HOUSTON 2, TEXAS

FOR A PROJECT THAT MAY BE DESCRIBED IN TERMS OF AN ARROW DIAGRAM OF ITS COMPONENT JOBS THIS PROGRAM FINDS THE MINIMUM PROJECT COMPLETION TIME. THE EARLIEST AND LATEST START AND FINISH TIMES FOR EACH JOB AND THE TOTAL AND FREE FLOAT TIME ARE CALCULATED. THE SUM OF NODES AND JOB ARROWS MAY BE AS HIGH AS 1672. STORAGE PROGRAM - 3275 DIGITS. 20K 1622 CARD READ PUNCH.

THIS PROGRAM AND ITS DOCUMENTATION WERE WRITTEN BY AN IBM EMPLOYEE. IT WAS DEVELOPED FOR A SPECIFIC PURPOSE AND SUBMITTED FOR GENERAL DISTRIBUTION TO INTERESTED PARTIES IN HOPE THAT IT MIGHT PROVE HELPFUL TO OTHER MEMBERS OF THE DATA PROCESSING COMMUNITY. THE PROGRAM AND ITS DOCUMENTATION ARE ESSENTIALLY IN THE AUTHORS ORIGINAL FORM. IBM SERVES AS THE DISTRIBUTION AGENCY IN SUPPLYING THIS PROGRAM. QUESTIONS CONCERNING THE USE OF THE PROGRAM SHOULD BE DIRECTED TO THE AUTHORS ATTENTION.

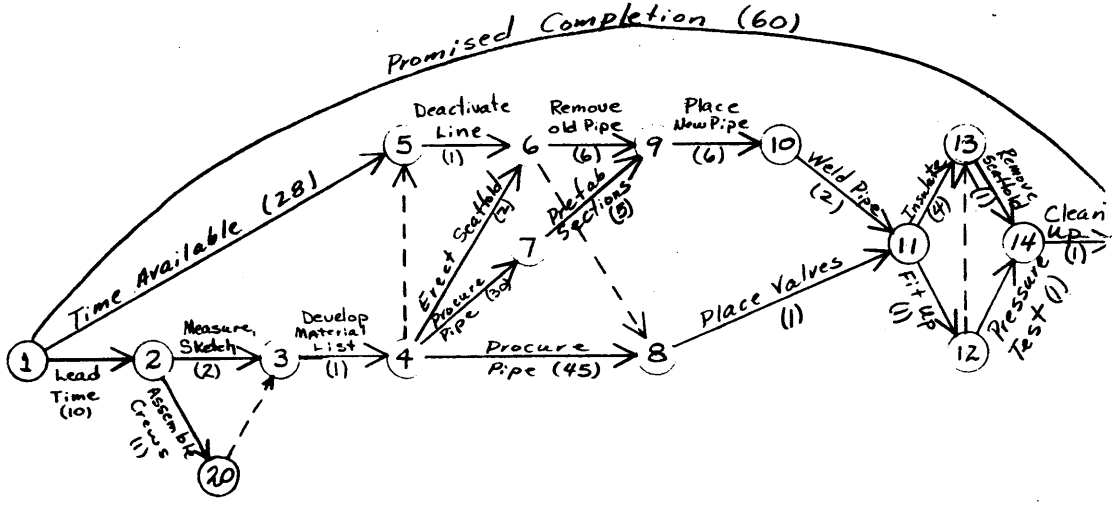


Figure 1 - Replace Pipe Line

PART II - ARROW DIAGRAM PLANNING

Fundamental to the technique being described is a graphical representation of any project by an arrow diagram which defines all jobs in the project and the order in which they must be done. Figure 1 is such a diagram, which represents the sequence of jobs necessary to replace a pipe line. This diagram will be used to illustrate several facts concerning arrow diagramming.

(1) Every job is represented by an arrow, and denoted by the numbers at the tail and head of the arrow. This set of numbers must be unique.
ex. Job (12, 14) is a pressure test.

(2) Jobs whose heads bear the same number as the tail of a given job must immediately preceed the given job.
ex. Job (11, 12) preceeds job (12, 14). That is (12, 14) cannot be started until (11, 12) is finished.

(3) Jobs whose tails bear the same number as the head of a given job must immediately succeed the given job.
ex. Job (14, 15) succeeds job (12, 14) and may not be started until (12, 14) and (13, 14) are finished.

(4) Jobs whose tails bear the same number may be done concurrently.
ex. Jobs (11, 12) and (11, 13) may be done concurrently.

(5) Dummy jobs (denoted by dotted line arrows) are inserted to complete the logic of an arrow diagram.
ex. Dummy (6, 8) shows that the jobs immediately preceeding job (8, 11) have heads numbered 6 as well as 8. That is jobs (5, 6), (4, 6), and (4, 8) preceed job (8, 11).

(6) Dummy jobs may also be introduced to satisfy rule 1. (All jobs must have a unique set of numbers.)

ex. Jobs (2, 3) and (2, 20) are concurrent jobs that must be complete before starting (3, 4). Since they could not both be called (2, 3), dummy (20, 3) was inserted.

The rules presented thus far allow descriptions of the technological sequence of jobs within a project. Actually, this planning should include a time estimate of each diagrammed job. A few additional rules will now be given that allow the injection of the time element.

(7) Every job has an estimated elapsed time associated with it. In the case of dummy jobs, this time is zero. This time may be used along with arrow head and tail to denote a job.
ex. The time estimated to complete the pressure test (12, 14, 1) is one day.

(8) In order to later calculate start and finish dates for each job, the first job is usually designated as lead time.
ex. Job (1, 2, 10) states that the project may begin on the 10th day of a particular calendar (or 10th hour of a clock). That is the first actual jobs (2, 3) and (2, 20) may begin on the 10th day.

(9) Time restraints on the execution of certain jobs may be described by the use of arrows with associated times.
ex. Restraint (1, 5, 28) means that the old pipe line must not be deactivated until the 28th day.

(10) Material delivery restraints do not always have to be tied to the calendar as in (9), but may be in elapsed time.
ex. Restraint (4, 7, 30) means that the pipe will be delivered 30 days after the completion of job (3, 4).

PART III - CRITICAL PATH SCHEDULING

The fact that scheduling has not yet been mentioned is a unique advantage of this technique - planning and scheduling are recognized as two separate functions. After completing the arrow diagram and estimating the duration of each job, a schedule (in the form of a detailed time table) is easily obtained by a few simple calculations. The following nomenclature is used.

I	Tail of a job, dummy, or restraint arrow.
J	Head of a job, dummy, or restraint arrow.
N	A Node. Either the head or tail of an arrow.
D (I,J)	Estimated elapsed time for job (I,J).
TI (N)	The earliest time that a job whose tail is N may start and assure minimum project completion time.
TJ (N)	The latest time that a job whose head is N may finish and assure minimum project completion time.
ES	Earliest start time. Same as TI (N)
EF	Earliest finish time.
LS	Latest Start time.
LF	Latest finish time. Same as TJ (N)
TF	Total float time. The length of time that the start of a job may be delayed without changing the minimum project completion time.
FF	Free float time. The length of time that the start of a job may be delayed without changing ES for another job.
λ	Minimum project completion time.

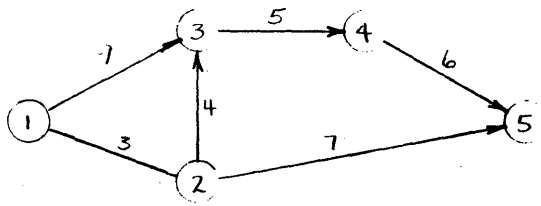


Figure 2

I	J	D(I,J)
1	2	3
1	3	9
2	3	4
2	5	7
3	4	5
4	5	6

Table 1A

ES	EF	LS	LF	TF	FF	
0	3	2	5	2	0	
0	9	0	9	0	0	*
3	7	5	9	2	2	
3	10	13	20	10	10	
9	14	9	14	0	0	*
14	20	14	20	0	0	*

Table 1B

N	TI(N)	TJ(N)
1	0	0 ~~7~~
2	3	~~13~~ 5
3	~~9~~ 7	9
4	14	14
5	~~10~~ 20	20

Table 2 —

The following steps are followed to calculate a time table for the project diagrammed in Figure 2.

(1) Place the planning results in a table like Table 1 A.

(2) Set up a row in Table 2 for each node in the diagram.

(3) Compute the TI (N) value in Table 2 by first setting TI (FIRST NODE) = 0 and then generating possible values of TI (J) = TI (I) + D (I,J). The largest such value of TI (J) is the correct value for a given node.

ex. TI (1) = 0

TI (2) $\overset{?}{=}$ TI (1) + D (1,2) = 0 + 3 = (3) = TI (2)

TI (3) $\overset{?}{=}$ TI (1) + D (1,3) = 0 + 9 = (9) = TI (3)

TI (3) $\overset{?}{=}$ TI (2) + D (2,3) = 3 + 4 = 7

TI (5) $\overset{?}{=}$ TI (2) + D (2,5) = 3 + 7 = 10

TI (4) $\overset{?}{=}$ TI (3) + D (3,4) = 9 + 5 = (14) = TI (4)

TI (5) $\overset{?}{=}$ TI (4) + D (4,5) = 14 + 6 = (20) = TI (5)

(4) The TI value for the end node will be the minimum completion time for the project.

ex. λ = TI (END) = TI (5) = 20

(5) Compute TJ (N) values by setting TJ (END) = λ and generating possible values of TJ (I) = TJ (J) - D (I,J). The smallest such value of TJ (I) is the correct value for a given node.

ex. TJ (END) = λ = TJ (5) = 20

TJ (4) $\overset{?}{=}$ TJ (5) - D (4,5) = 20 - 6 = 14, etc.

(6) With Table 2 complete, Table 1 B can be constructed by use of the following relationships.

ex. For job (1,2)

ES = TI (I) = TI (1) = 0

EF = ES + D (I,J) = 0 + 3 = 3

LF = TJ (J) = TJ (2) = 5

LS = LF - D (I,J) = 5 - 3 = 2

TF = LS - ES = 2 - 0 = 2

FF = TI (J) - EF = TI (2) - EF = 3-3 = 0

The longest chain of jobs through a project is termed the "critical path." The jobs along this path have zero total float times and are marked by an asterisk in Table 1 B. Any delay in the starting or completion of these jobs will delay completion of the project by a like amount of time. On the other hand some of the jobs are floaters and may be delayed a limited amount without effecting the project completion date.

The indexing character (column 27) is not used with the 141.

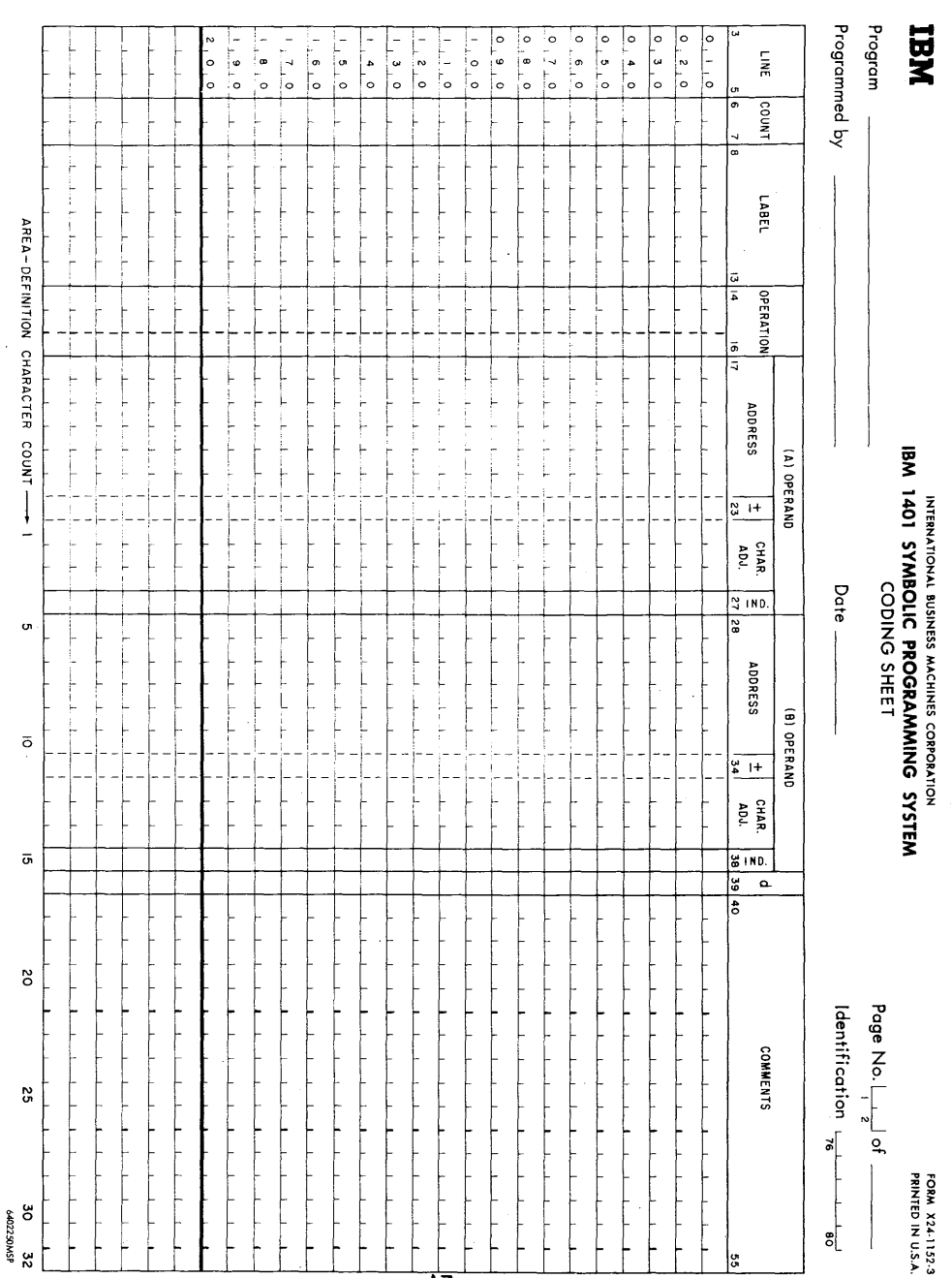
If the instruction requires a B-operand, its address is written in columns 28 through 37 in the same form as the A-operand.

When an instruction requires a d-character, the actual machine code is placed in column 39.

COMMENTS

Short comments may be placed in columns 40 through 55 of the instruction cards. Longer comments may be placed on "Comment Cards". These cards are identified by an asterisk in column 8. The remainder of the card, columns 9 through 55, is available for the comment.

A sample coding sheet is shown on the next page.



DECLARATIVES

Define Constant With Word Mark — DCW

The symbolic operation code DCW causes a constant to be loaded into storage and sets a word mark in the high-order (left most) position of the constant field. The number of characters in the constant field is specified in the Count portion of the coding sheet, (columns 6 and 7). The symbolic label by which the constant is referenced is placed in the Label area (columns 8 through 13). The code DCW is placed in columns 14 through 16. Column 17 must contain an asterisk to indicate to the assembler that it may choose the location of the constant field or else columns 17 through 20 must contain the desired storage location of the low order position (right most) of the constant field. The constant itself begins in column 24 and may extend through column 55 giving a maximum of 32 characters. If the constant is to be a signed number, the sign may be placed in column 23.

Define Constant — DC

The symbolic operation code DC causes a constant to be loaded into storage without a word mark. Otherwise, it is identical to the DCW.

Define Symbol — DS

The operation code DS causes the processor to assign equivalent addresses to labels or to assign storage for work areas. The DS differs from DC and DCW statements in that neither data nor word marks are loaded during assembly. The number of positions to be reserved in storage is specified in the Count portion of the coding sheet. If it is desired to refer symbolically to the low order position of the field reserved, then a label must be placed in the Label field. If the assembler is to assign the address, an asterisk must be placed in column 17 of the coding sheet. If it is desired to equate the label to an actual address, then that address is written beginning in column 17 and the Count field of the coding sheet is left blank. It is not possible to character adjust DS statements.

Define Symbolic Address — DSA

The DSA statement causes a three character machine language address which the assembler has assigned to a label to be stored as a constant when the program is loaded.

The number of characters need not be specified in the Count portion of the coding sheet since it is automatically assigned three storage positions by the processor. If it is desired to refer to the address of the address field, a symbol may be written in the Label portion of the coding sheet. Column 17 may contain an asterisk thus allowing the assembler to assign the storage positions or else columns 17 through 20 may contain the desired storage locations of the low order position for the address field. The symbol whose equivalent address is to be the address field is written beginning in column 28 of the B-operand.

CONTROL STATEMENTS

Origin — ORG

The ORG statement causes the assembler to assign addresses to the following instructions beginning at the location specified by the statement. The symbolic operation code ORG must be placed in the operation field and the absolute address at which storage assignment is to be made must be written in columns 17 through 20 of the coding sheet.

Execute — EX

The EX statement causes the computer to suspend loading of the object program and execute part of the program prior to continuing the loading process. The symbolic operation code EX must be placed in the operation field and the symbolic or actual address of the first instruction to be executed when the loading process is suspended must be placed in the A-operand portion of the coding sheet. The card containing the Execute statement must be inserted at the point in the source program where suspension of loading is desired in order to execute the preceeding portion.

End — END

The END statement is an indication to the assembler that the last card of the source program has been processed. The symbolic operation code END must be placed in the operation field and the address of the first instruction, either actual or symbolic, must be placed in the A-operand portion of the coding sheet.

SECTION 3

EXERCISES

Exercise 1

Write a program that will reproduce a card, that is, will read a card and punch a card identical to the one read.

Exercise 2

Write a program that will read a card and punch a card with the information from columns 1 - 40 of the card read in columns 41 -80 of the card punched and the information from columns 41 - 80 of the card read in columns 1 - 40 of the card punched.

Exercise 3

Write a program that will reproduce an entire deck of cards.

Exercise 4

Write a program that will read one card and will punch copy after copy of it until the machine is stopped by the operator.

Exercise 5

Write a program that will print a directory of telephone extensions from a deck of personnel cards. The cards and directory forms are as follows:

Card Columns	Field	Print Positions
1 - 18	Name	1 - 18
19	First Initial	20
20	Second Initial	22
21 - 60	Not used in this program	
61 - 64	Telephone Extension	28 - 31
65 - 80	Not used in this program	

Exercise 6

Write a program that will read cards containing numeric fields A, B, and C and will punch corresponding cards that contain fields A, B, C, and D, where D = A + B - C. The card columns are shown on the next page.

Field	Card Columns	Card
A	1 - 6	Input and Output
B	7 - 11	Input and Output
C	12 - 14	Input and Output
D	75 - 80	Output Only

Assume that no overflows will occur.

Exercise 7

Write a program that will check the sequence of employee numbers found in columns 75 - 80 of a deck of cards. The program should stop the machine if it finds any employee number that is not larger than the one in the previous card.

Exercise 8

Write a program that will punch consecutive numbers 001 through 015 in columns 78 - 80 of the first 15 blank cards in the punch hopper and stop automatically before punching a sixteenth card.

Exercise 9

Write a program that will calculate and punch D, where D = A + B - C (all values are positive). Provide for decimal alignment, rounding (half-adjustment), and over flow. The card columns and decimal form of each field is as follows:

Input Card	A	Col.	5 - 8	XXX.X
	B		9 - 12	XX.XX
	C		13 - 14	XX.
Output Card	D		7 - 10	XXXX.

Exercise 10

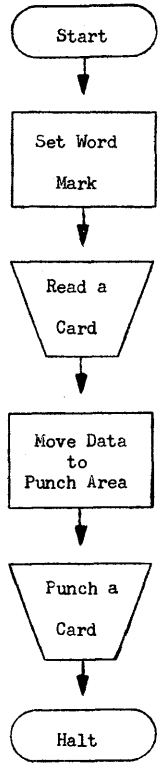
Write a program that will up-date a customer's charge account after a new purchase has been recorded. A new balance card is to be punched and a listing of each customer's name, new balance, and limit is to be printed. If the new balance exceeds the customer's limit the words OVER LIMIT are also to be printed on his entry. The card columns and print positions are as follows:

Filed	Input Card	Output Card	Listing
Name	1 - 20	1 - 20	11 - 30
Balance	21 - 30	21 - 30	35 - 44
Charge	31 - 40		
Limit	71 - 80	71 - 80	49 - 58
'OVER LIMIT'			63 - 72

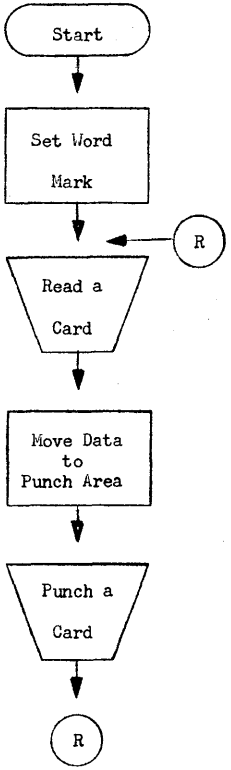
The Limit field is to be punched with leading zeros.

BLOCK DIAGRAMS

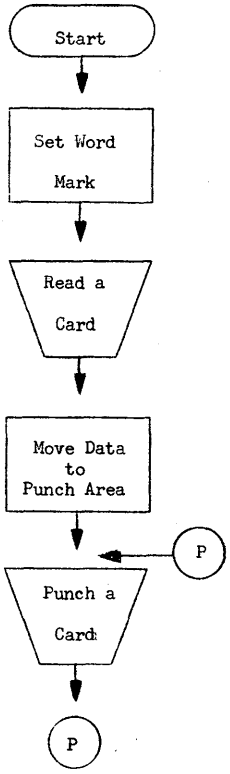
Exercises 1 and 2



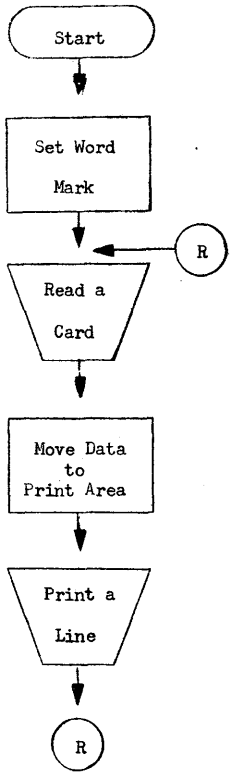
Exercise 3



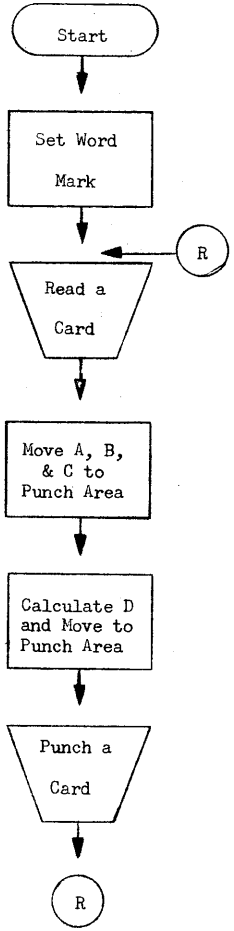
Exercise 4



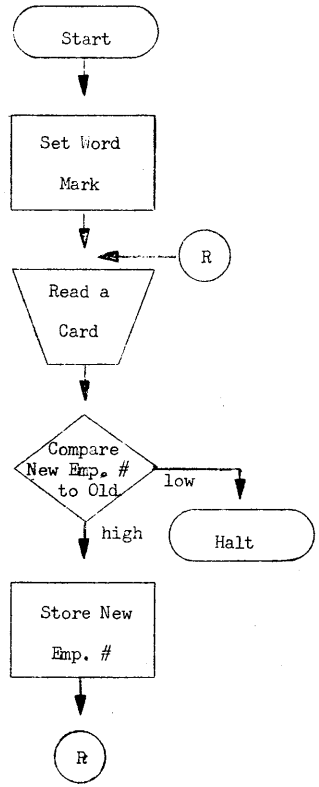
Exercise 5



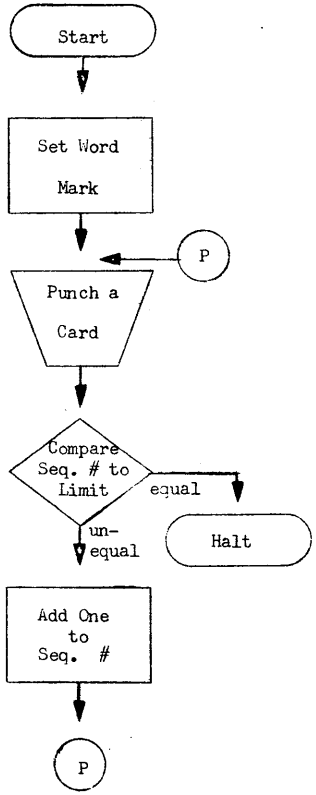
Exercise 6



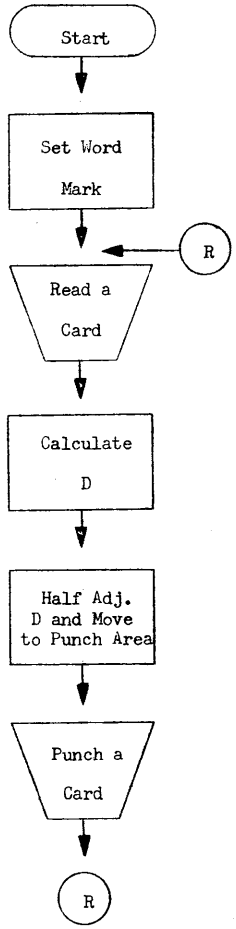
Exercise 7



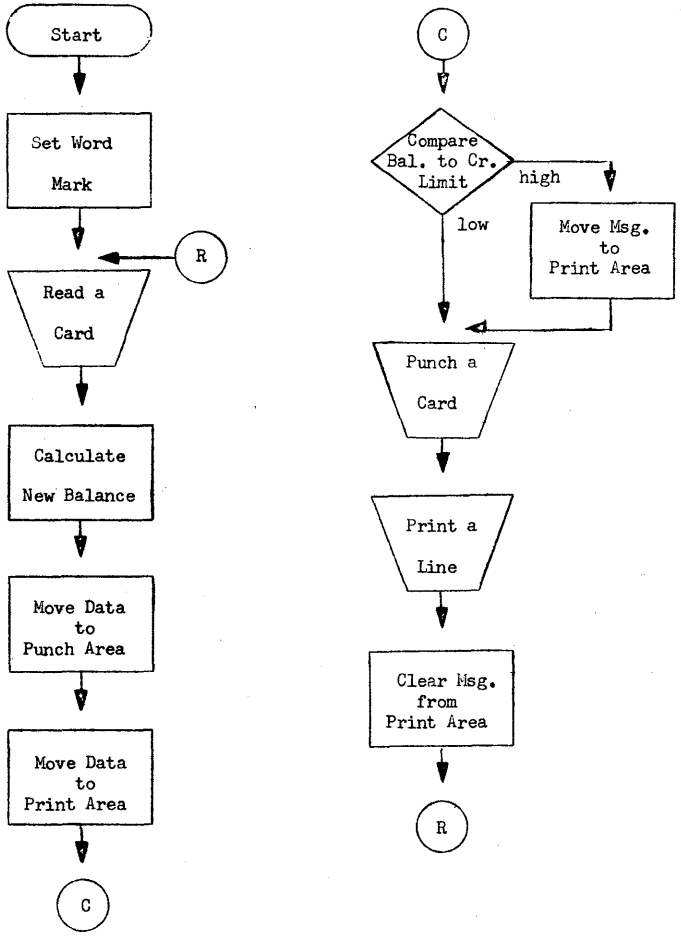
Exercise 8



Exercise 9



Exercise 10



SOLUTIONS TO EXERCISES

EXERCISE 1

```
PG LIN    LABEL   OP   A-OPERAND   B-OPERAND   COMMENTS
01 010    START   SW   0001                    DEFINE 80 POS FL
01 020            R                            READ ONE CARD
01 030            MCW  0080        0180        MOVE TO PCH
01 040            P                            PUNCH ONE CARD
01 050            H                            HALT
01 060            NOP                          PROVIDE WM
01 070            END  START
```

EXERCISE 2

```
02 010    START   SW   0001                    DEFINE FIRST FLD
02 020            SW   0041                    DEFINE SECOND FL
02 030            R                            READ ONE CARD
02 040            MCW  0040        0180        MOVE TO PCH AREA
02 050            MCW  0080        0140        MOVE TO PCH AREA
02 060            P                            PUNCH A CARD
02 070            H                            HALT
02 080            NOP                          PROVIDE WM
02 090            END  START
```

EXERCISE 3

```
03 010    START   SW   0001                    DEFINE FIELD
03 020    READ    R                            READ A CARD
03 030            MCW  0080        0180        MOVE TO PCH AREA
03 040            P                            PUNCH A CARD
03 050            B    READ                    BRANCH TO READ
03 060            END  START
```

EXERCISE 4

```
PG LIN    LABEL   OP  A-OPERAND   B-OPERAND   COMMENTS

04 010    START   SW  0001                    DEFINE FIELD
04 020            R               READ CARD
04 030            MCW 0080        0180        MOVE TO PCH AREA
04 040    PUNCH   P               PUNCH
04 050            B   PUNCH       REPEAT PUNCH
04 060            END START
```

EXERCISE 5

```
05 010    START   SW  0001                    DEFINE FIELDS
05 020            SW  0019
05 030            SW  0020
05 040            SW  0061
05 050    READ    R               READ CARD
05 060            MCW 0018        0218        ASSEMBLE LINE
05 070            MCW 0019        0220
05 080            MCW 0020        0222
05 090            MCW 0064        0231
05 100            W               PRINT A LINE
05 110            B   READ        RETURN TO READ
05 120            END START
```

EXERCISE 6

```
06 010    START   SW  0001                    DEFINE FIELDS
06 020            SW  0007
06 030            SW  0012
06 040    READ    R               READ CARD
06 050            MCW 0006        0106        MOVE INPUT TO
06 060            MCW 0011        0111        PUNCH AREA
06 070            MCW 0014        0114
06 080            S   0014        0006        A-C
06 090            A   0011        0006        A+B-C
06 100            MCW 0006        0180        MOVE D TO PCH AR
06 110            P               PUNCH CARD
06 120            B   READ        LOOP
06 130            END START
```


EXERCISE 7

```
PG LIN    LABEL   OP  A-OPERAND   B-OPERAND   COMMENTS

07 010    BEGIN   SW  0075                    DEFINE EMPNO FLD
07 020    READ    R               READ CARD
07 030            C   STORE       0080        COMP WITH LST CD
07 040            B   LOOP                    ULOOP IF OK
07 050            H               HALT
07 060    LOOP    MCW 0080        STORE
07 070            B   READ
07 080  6 STORE   DCW *
07 090            END BEGIN
```

EXERCISE 8

```
08 010    FIRST   SW  0178                    DEFINE FLD
08 020    PUNCH   P               PUNCH
08 030            C   0180        LIMIT       TEST FOR LIMIT
08 040            B   HALT                    S
08 050            A   ONE         0180        STEP SEQ NO
08 060            B   PUNCH                   LOOP
08 070    HALT    H               HALT
08 080  3 LIMIT   DCW *       015
08 090  1 ONE     DCW *       1
08 100  3         DCW 0180    001
08 110            END FIRST
```

EXERCISE 9

```
09 010    START   SW  0005                    DEFINE FIELDS
09 020            SW  0009
09 030            SW  0013
09 040    READ    R               READ CARD
09 050            A   0008        ACCUM - 1   A
09 060            A   0012        ACCUM       A+B
09 070            S   0014        ACCUM - 2   A+B-C
09 080            A   FIVE        ACCUM - 1   HALF ADJUST
09 090            MCW ACCUM - 2 0110          MOVE TO D
09 100            P
09 110            MCW ZEROS       ACCUM       CLEAR ACCUM
09 120            B   READ
09 130  6 ACCUM   DCW *       000000
09 140  6 ZEROS   DCW *       000000
09 150  1 FIVE    DCW *       5
09 160            END START
```

EXERCISE 10

```
PG LIN     LABEL   OP   A-OPERAND   B-OPERAND   COMMENTS

10 010     START   SW   0001                    DEFINE FIELDS
10 020             SW   0021
10 030             SW   0031
10 040             SW   0071
10 050     READ    R                            READ CARD
10 060             A    0040        0030        CALC NEW BAL
10 070             MCW  0020        0120        MOVE TO PCH
10 080             MCW  0030        0130
10 090             MCW  0080        0180
10 100             MCW  0020        0230        MOVE TO PRINT
10 110             MCW  0030        0244
10 120             MCW  0080        0258
10 130             C    0030        0080        TEST FOR HI BAL
10 140             B    OVER                   T
10 150     PUNCH   P                            PUNCH
10 160             W                            WRITE
10 170             MCW  BLANK       0272        CLEAR MSG
10 180             B    READ
10 190     OVER    MCW  MSG         0272        INSERT MSG
10 200             B    PUNCH
10 210 10  BLANK   DCW  *
10 220 10  MSG     DCW  *           OVER LIMIT
10 230             END  START
```


SUBROUTINES

The following subroutines written in 141 language were contributed by Mr. Wilson T. Price of Merritt College, Oakland, California. In preparing these routines, simplicity of arithmetic method, compatability with the 1401, and compatability with each other were primary considerations. Speed of operation was deemed the least important feature since students write 141 programs as learming experience and not for production runs.

THE MULTIPLY SUBROUTINE

TITLE: Multiply

MNEMONIC: MULT

PURPOSE: To provide the capability of multiplying a number containing up to 8 digits by a second number containing up to 8 digits to form a product up to 16 digits in length.

STORAGE REQUIREMENTS:

Multiplicand	(MULTD)	081 through 089
Multiplier	(MULTR)	091 through 099
Product	(PROD)	181 through 196
Additional work areas		197 through 200
Program		100 additional locations as assigned by assembler

LINKAGE: Move the multiplicand of m digits to MULTD. This field will then occupy storage positions (090 - m) through 089. Move the multiplier of n digits to MULTR. This field will then occupy storage positions (100 - n) through 099. Move the return Branch instruction to MULTX + 3. Branch to MULT. The linkage is illustrated below:

```
          MCW (Multiplicand)    MULTD
          MCW (Multiplier)      MULTR
          MCW RETURN - 1        MULTX + 3
          B   MULT
          B   RETURN
RETURN    (next instruction in program)
```

After completion of the operation, the product of m + n digits will be in PROD. Both the multiplicand and multiplier remain in their respective areas.

WORD MARKS: Word marks are placed in locations 081, 091, and 181 with DCW's during assembly and care must be exercised that they are not cleared during execution of the main program.

CLEARING: Initially all three work areas will be zero, further clearing is left to the programmer. Blanking or zeroing of the multiplicand and multiplier areas will only be necessary if the new values contain fewer digits than the previous quantities which utilized these areas. Zeroing of the product accumulator will always be necessary unless it is desired to sum products.

SCALING: Decimal alignment is the responsibility of the programmer. The number of decimal places in the product is equal to the sum of the number of decimal places in the multiplicand and the multiplier.

MULTIPLY SUBROUTINE

```
PG LIN    LABEL  OP  A-OPERAND      B-OPERAND      COMMENTS
M1 010    MULT   MCW M16            M3   + 3
M1 020           MCW M17            M6   + 6
M1 030    M3     MCW MULTR - 7      M19  - 1
M1 040    M4     C   M19    - 1     M18  - 1
M1 050           B   M9                          U
M1 060    M6     A   MULTD          PROD - 7
M1 070           S   M18            M19
M1 080           B   M4
M1 090    M9     SW  M3     + 1     M6   + 4
M1 100           A   M18    - 1     M3   + 3
M1 110           A   M18    - 1     M6   + 6
M1 120           CW  M3     + 1     M6   + 4
M1 130           C   M3     + 3     M16  - 2
M1 140           B   M3                          /
M1 150    MULTX  B   0000
M1 160 03 M16    DCW *      092
M1 170 02 M17    DCW *      89
M1 180 02 M18    DCW 0198   10
M1 190 02 M19    DCW 0200   00
M1 200 09 MULTD  DCW 0089   000000000
M1 210 09 MULTR  DCW 0099   000000000
M1 220 16 PROD   DCW 0196   0000000000000000
```

THE DIVIDE SUBROUTINE

TITLE: Divide

MNEMONIC: DIV

PURPOSE: To provide the capability of dividing a number containing up to 16 digits by a second number containing up to 8 digits to form a quotient of up to 8 digits.

STORAGE REQUIREMENTS:

Dividend	(DIVD)	181 through 196
Divisor	(DIVR)	081 through 089
Quotient	(QUOT)	091 through 099
Program		154 additional locations as assigned by assembler

LINKAGE: Move the dividend of m digits to DIVD. This field will then occupy storage positions (197 - m) through 196. Move the divisor of n digits to DIVR. This field will then occupy storage positions (090 - n) through 089. Move the return Branch instruction to DIVX + 3. Branch to DIV.

```
          MCW (Dividend)   DIVD
          MCW (Divisor)    DIVR
          MCW RETURN - 1   DIVX + 3
          B   DIV
          B   RETURN
RETURN    (next instruction in program)
```

After completion of the operation, the quotient will be located at QUOT and the remainder at DIVD. The divisor remains in DIVR but the dividend is lost.

WORD MARKS: Word marks are placed in locations 081, 091, and 181 with DCW's during assembly and care must be taken that they are not cleared during execution of the main program.

CLEARING: Initially all three work areas will contain zeroes, further clearing is left to the programmer. Zeroing of the dividend and divisor areas will be necessary if new values contain fewer digits then previous quantities which utilized these areas. The high order position (081) of the divisor must contain zero. Zeroing of the quotient accumulator will always be necessary unless it is desired to sum quotients.

SCALING: Decimal alignment is the responsibilty of the programmer. The rules to follow are:listed on the next page.

1. Multiply dividend and divisor by the appropriate power of ten to clear decimals from divisor.
2. Muliply dividend and expected quotient by the same power of ten to obtain greater accuracy.
3. Upper eight digits (181 through 188) of dividend must be less than divisor.

The following examples illustrate scaling in the divide subroutine:

$$\frac{38}{1.2} = \frac{380}{12}$$

		Number		Location of low order position
1.	Before division	380		DIVD
		12		DIVR
	After division	31		QUOT
		8	(remainder)	DIVD
2.	Before division	380^0		DIVD
		12		DIVR
	After division	31^6		QUOT
		0^8	(remainder)	DIVD
3.	Before division	380^00		DIVD
		12		DIVR
	After division	31^66		QUOT
		0^08	(remainder)	DIVD

DIVIDE SUBROUTINE

```
PG LIN     LABEL  OP   A-OPERAND      B-OPERAND      COMMENTS
D1 010     DIV    MCW  D24            D7    + 3
D1 020            MCW  D25            D11   + 6
D1 030            MCW  D24            D13   + 6
D1 040            C    DIVR           DIVD  - 8
D1 050            B    D7                              T
D1 060            H    DIVX
D1 070     D7     MCW  DIVD  - 7      D26   - 1
D1 080     D8     C    D26   - 1      DIVR
D1 090            B    D13                             U
D1 100            S    DIVR  + 1      D26
D1 110     D11    A    D24   - 2      QUOT  - 7
D1 120            B    D8
D1 130     D13    MCW  D26   - 1      DIVD  - 7
D1 140            SW   D7    + 1      D11   + 6
D1 150            SW   D13   + 4
D1 160            C    D24            D11   + 6
D1 170            A    D24   - 2      D7    + 3
D1 180            A    D24   - 2      D11   + 6
D1 190            A    D24   - 2      D13   + 6
D1 200            CW   D7    + 1      D11   + 6
D2 010            CW   D13   + 4
D2 020            B    D7                              /
D2 030     DIVX   B    0000
D2 040 03  D24    DCW  *       189
D2 050 01  D25    DCW  *       2
D2 060 10  D26    DCW  *       0000000000
D2 070 09  DIVR   DCW  0089    000000000
D2 080 09  QUOT   DCW  0099    000000000
D2 090 16  DIVD   DCW  0196    0000000000000000
```


THE SUPPRESS ZERO SUBROUTINE

TITLE: Suppress Zero

MNEMONIC: SUPZR

PURPOSE: Given a numeric field of 9 digits or fewer, to suppress leading zeroes (that is change high order zeroes to blanks).

STORAGE REQUIREMENTS:

Work area	(SZARG)	091 through 099
Program		82 additional locations as assigned by assembler

LINKAGE: Move the numeric field of m digits to SZARG. The field will then occupy storage positions (100 - m) through 099. For example, a three digit field would occupy positions 097 through 099. Move the return Branch instruction to SUPZRX + 3. Branch to SUPZR.

```
          MCW (Argument)   SZARG
          MCW RETURN - 1   SUPZRX + 3
          B   SUPZR
          B   RETURN
RETURN    (next instruction in program)
```

After completion of the operation, the field with leading zeroes suppressed will remain in its original location. If the entire field is zero, then one zero will remain.

WORD MARKS: A word mark is set at location 091 during processing by the assembler. If cleared during execution of the main program it should be reset.

CLEARING: Initially the work area will be zero, further clearing is left to the programmer. Zeroing will always be necessary if the new field contains fewer digits than the previous quantity which utilized this area.

SUPPRESS ZERO SUBROUTINE

```
PG LIN     LABEL   OP   A-OPERAND        B-OPERAND       COMMENTS
S1 010     SUPZR   MCW  SZ15             SZ3   +  3
S1 020             MCW  SZ15             SZ5   +  6
S1 030     SZ3     C    SZARG -  8  SZ13  -  1
S1 040             B    SUPZRX                          T
S1 050     SZ5     MCW  SZ14             SZARG -  8
S1 060             SW   SZ3   +  1  SZ5   +  4
S1 070             A    SZ13             SZ3   +  3
S1 080             A    SZ13             SZ5   +  6
S1 090             CW   SZ3   +  1  SZ5   +  4
S1 100             C    SZ3   +  3  SZ15  -  1
S1 110             B    SZ3                             /
S1 120     SUPZRX  B    0000
S1 130 02  SZ13    DCW  *       01
S1 140 01  SZ14    DCW  *
S1 150 02  SZ15    DCW  *       91
S1 160 09  SZARG   DCW  0099    000000000
```


THE EDIT SUBROUTINE

TITLE: Edit

MNEMONIC: EDIT

PURPOSE: To provide the capability to edit a field of up to 8 digits consisting of dollars and cents. Leading zeroes are suppressed and a decimal point, a comma (if needed) and a floating dollar sign are placed in appropriate positions of the field.

STORAGE REQUIREMENTS:

Input field	(EDIN)	081 through 089
Output field	(EDOUT)	181 through 191
Program		127 additional locations as assigned by assembler

LINKAGE: Move the field of m digits to be edited to EDIN. This field will then occupy positions (090 - m) through 089. Move the return Branch instruction to EDITX + 3. Branch to EDIT.

```
          MCW (Argument)   EDIN
          MCW RETURN - 1   EDITX + 3
          B   EDIT
          B   RETURN
RETURN    (next instruction in program)
```

After completion of the operation, the edited field will be located at EDOUT. The original field remains in EDIN.

WROD MARKS: Word marks are placed in locations 081 and 191 with DCW's during assembly and care must be taken that they are not cleared during execution of the main program.

CLEARING: Initially both work areas will be zero, further clearing is left up to the programmer. Zeroing of the input area (EDIN) will be necessary if the new argument contains fewer digits than previous quantities which utilized this area. The output area (EDOUT) is self clearing.

SCALING: Quantities which are edited must consist of a dollar and cent amount. The following examples illustrate scaling in the edit subroutine:

Input field	Output field
12345678	$123,456.78
12345	$123.45
123	$1.23
12	$0.12

EDIT SUBROUTINE

```
PG LIN     LABEL   OP   A-OPERAND      B-OPERAND      COMMENTS
E1 010     EDIT    MCW  ED20           ED9   + 3
E1 020             MCW  ED20           ED11  + 6
E1 030             MCW  EDIN           EDOUT
E1 040             MCW  ED21           EDOUT - 2
E1 050             MCW  EDIN   - 2     EDOUT - 3
E1 060             MCW  ED21   - 1     EDOUT - 6
E1 070             MCW  EDIN   - 5     EDOUT - 7
E1 080             MCW  ED21   - 2     EDOUT - 10
E1 090     ED9     C    EDOUT - 9      ED19  - 1
E1 100             B    EDITX                    T
E1 110     ED11    MCW  ED21   - 2     EDOUT - 9
E1 120             SW   ED9    + 1     ED11  + 4
E1 130             A    ED19           ED9   + 3
E1 140             A    ED19           ED11  + 6
E1 150             CW   ED9    + 1     ED11  + 4
E1 160             C    ED9    + 3     ED20  - 1
E1 170             B    ED9                      /
E1 180     EDITX   B    0000
E1 190 02  ED19    DCW  *        01
E1 200 02  ED20    DCW  *        82
E1 210 04  ED21    DCW  *         $,.
E1 220 11  EDOUT   DCW  0191     00000000000
E1 230 09  EDIN    DCW  0089     000000000
```


OPERATING PROCEDURES

Four versions of the 141 SPS Assembler and the 141 Simulator are available in order to permit maximum utilization of the computer hardware. These are identified as:

Non-Monitor Versions

Version A - Basic 1620
Version B - 1620 with 1443 Printer

Monitor Versions

Version C - 1620 with 1311 Disk Storage Drive and indirect addressing
Version D - 1620 with 1443 Printer, 1311 Disk Storage Drive, and indirect addressing

Letters preceeding each procedure statement below identify the versions to which they apply.

141 SPS ASSEMBLER

Prepare Console

Version	
A C	1) Set left typewriter margin at 10 and right margin at 95.
A B C D	2) Set Parity Switch and I/O Switch to STOP.
A B C D	3) Set O'Flow Switch to PROGRAM.
C D	4) Set Disk Switch to PROGRAM.
A B C D	5) Set Program Switches 1 and 2 according to the options listed below.

Assemble SPS Programs

Version		
A B	1)	Place the 141 SPS Assembler deck in the reader hopper in the 9-edge face-down position.
C D	2)	Place the following Monitor cards in the reader hopper: "COLD START", ≠ ≠ JOB, and ≠ ≠ XEQ 141SPS.
A B C D	3)	Place SPS source program decks in the reader hopper. Any number of programs may be stacked for assembly. The last card of each deck must be an END statement.
A B C D	4)	With the machine in MANUAL mode, press the LOAD key on the 1622 Reader-Punch unit.

Program Switch Options

Version		
A B C D	1)	Switch 1 and 2 off - Object deck will be punched and program will be listed.
A B C D	2)	Switch 1 off and Switch 2 on - Object deck will be punched but program listing will be suppressed except for incorrect statements. A program listing can be prepared from the object program cards on an IBM 407 Accounting Machine. This option will greatly reduce assembly time for versions A and C.
A B C D	3)	Switch 1 on and Switch 2 off - Object deck will be suppressed and program will be listed on the console typewriter (or printer).
A B C D	4)	Switch 1 and 2 on - Object deck and program listing will be suppressed. This combination can be used as an edit run. Programs from an entire class can quickly be scanned for errors with only incorrect statements being listed. The particular op-code or address that is erroneous will appear as the symbol =. For easy recognition, be sure that the source cards are numbered in columns 1 through 5 and that the IDENTIFICATION field, columns 76 through 80, is punched.

Long Programs

Version		
A B	1)	An SPS assembly is a two pass operation but the 141 SPS assembler only requires that the cards be fed through once if the number of cards in the source program does not exceed 100. This reduces the amount of card handling and permits the stacking of of programs. If the number of cards in a source program is greater than 100, images of the first 100 cards are held in storage and copies of the remaining cards are punched for a second pass. These cards are removed from the PUNCH stacker and placed in the READ hopper at the end of PASS I. Only those statements in excess of 100 need be processed twice.
C D	2)	Images of the source cards are stored on the disk and therefore the length of the program does not effect the operating procedures.

141 SIMULATOR

Prepare Console

Version		
A C	1)	Set left margin at 10 and right margin at 95.
A B C D	2)	Set Parity Switch to STOP.
A B C D	3)	Set O'Flow Switch to PROGRAM.
C D	4)	Set Disk Switch to PROGRAM.
A B C D	5)	Set Program Switches 1,2,3, and 4 according to the options listed at the end of this section.

Load Simualtor

Version		
A B	1)	Place 141 Simulator deck in the reader hopper in the 9-edge face-down position.
C D	2)	Place the following Monitor cards in the reader hopper: "COLD START", ≠ ≠ JOB, and ≠ ≠ XEQ 141SIM
A B C D	3)	With the machine in MANUAL mode, press the LOAD key on the 1622 Reader-Punch unit. When the Simulator is loaded the typewriter will automatically begin typing a list of the functions that the simulator will perform and the request words that will initiate these functions.

Functions Performed	Request by Typing
Load Program From Card Reader	LOAD
Clear 141 Storage	CLEAR
Alter Storage From Typewriter	ALTER
Dump Contents of 141 Storage	DUMP
Begin Execution of Program	EXECUTE
Return to 1620 Monitor	EXIT (C & D only)

Select the Desired Function

Each function, except EXIT, is available in all versions.

a) The typewriter will type the words REQUESTED FUNCTION IS and then stop.
b) The operator then types the word LOAD, CLEAR, ALTER, DUMP, EXECUTE or EXIT and presses the RELEASE and START keys on the console or the RS key on the typewriter.
c) If a function runs to completion the simulator will automatically request the next function. If the function is interrupted by turning on Program Switch 1, the operator may return to the request statement by pressing, in order, the RESET, INSERT, RELEASE, and START keys on the console.

The LOAD Function

Programs that have been assembled by SPS can be loaded with this function.

a) Place the SPS object deck, including the two clear storage cards and the bootstrap card, in the hopper.
b) Type the request word LOAD and press the RELEASE and START keys.
c) Press READER START, if necessary.

The CLEAR Function

The 141 storage can be cleared (set to blanks) with this function.

a) Type the request word CLEAR and press the RELEASE and START keys.
b) When the clearing operation is completed the typewriter will request the next function.

The ALTER Function

Instructions and data, including word marks, in the 141 storage can be altered with this function. This may be used for debugging a program or entering complete small demonstration programs directly in machine language.

a) Type the request word ALTER and press the RELEASE and START keys.
b) The typewriter will type BEGINNING AT.
c) Type the three digit 141 location at which the alteration is desired and press the RELEASE and START keys.
d) The typewriter will repeat this location to verify it.
e) Type the instructions and data in machine language, disregarding word marks. This is the only instance where the operator will have to use the typewriter shift key. For all other entries the typewriter will automatically be in the proper alphabetic or numeric shift. At any convenient place, at least one character before the end of the line, cease typing and press the RELEASE and START keys.

f) The typewriter carriage will return for a second line. This line will indicate the presence or absence of word marks. If the character above requires a word mark type a 1, if it does not, strike the space bar. Continue to type 1's and spaces until the carriage has moved across the entire line above. In the first position after completion of the word mark line, type a record mark, and then press the RELEASE and START keys.
g) The typewriter will now type the address of the next storage location that will be altered if steps c) and f) are repeated.
h) When altering is completed press, in order, the RESET, INSERT, RELEASE, and START keys. The EXECUTE function can be used to start the program.

The DUMP Function

When a 141 program is stopped either by a programmed halt or by an error condition, it is desirable to be able to "DUMP" the Instruction Register (I-REG), the Operation Register (OP-REG) and the storage. The DUMP function will list the contents of the I-REG, which will be the address of the next character to be accessed, the contents of the OP-REG, which is the operation code of the last instruction to be executed, and the contents of the 141 storage as it stood when the program stopped.

a) Type the request word DUMP and press the RELEASE and START keys.
b) When the entire storage is dumped the typewriter will request the next function.

The EXECUTE Function

Execution of 141 programs can be started with this function.

a) Type the request word EXECUTE and press the RELEASE and START keys.
b) The typewriter will type BEGINNING AT.
c) Type the three-digit 141 location of the first instruction to be executed and press the RELEASE and START keys.

The EXIT Function

In versions C and D this function returns control to the 1620 Monitor.

a) Type the request word EXIT.
b) Press the RELEASE and START keys.

Program Switch Options

a) Program Switch 1 - Turning Program Switch 1 on will cause the program to halt at the end of the execution of the current 141 instruction. The operator may either press START to continue with the next 141 instruction or he may press RESET, INSERT, RELEASE and START to request a new function.

b) Program Switch 2 - When Program Switch 2 is off the DUMP function will use the typewriter or printer. When it is on the DUMP function will use the card punch. These cards can be listed on an IBM 407 Accounting Machine.

c) Program Switch 3 - Cards punched by the DUMP function can be reloaded with the LOAD function with Program Switch 3 on. With Program Switch 3 off SPS self-loading cards can be loaded.

d) Program Switch 4 - If Program Switch 4 is on at the time the simulator is loaded the typing of the list of functions will be omitted.

Special Notes

a) Restarting Programs - 141 programs can be stopped, dumped, and later restarted by the following procedure:
   1) Stop the program by turning Program Switch 1 on.
   2) Dump the program on cards using the DUMP function with Program Switch 2 on.
   3) Later re-load the program using the LOAD function with Program Switch 3 on.

b) Console Lights - When a 141 program is stopped by a program halt, an error halt, or by turning on Program Switch 1, the operation code of the instruction just completed can be determined from the DIGIT AND BRANCH lights on the console. The 1620 display can be converted to a 141 operation code by using the following table:

DIGIT AND BRANCH	141 OP-CODE	SPS OP-CODE	DIGIT AND BRANCH	141 OP-CODE	SPS OP-CODE
03	.	H	53	L	LCA
04	¤ or)	CW	54	M	MCW
21	/	CS	55	N	NOP
23	,	SW	62	S	S
41	A	A	71	1	R
42	B	B	72	2	W
43	C	C	74	4	P

The address of the next instruction to be executed can be determined by pressing the DISPLAY MAR key with the MEMORY ADDRESS REGISTER SELECTOR rotated to the OR-2 position. The 141 address of the next instruction will be displayed by the lights of the MEMORY ADDRESS REGISTER.

c) Loading Machine Language Programs - Machine language programs can be loaded either by typing them under the control of the ALTER function or by key punching them in the Card Dump format and loading them using the LOAD function with Program Switch 3 on.

Card Dump Format - Cards in this format must be sequentially numbered with the odd numbered cards containing the program and data characters and the even numbered cards containing the word marks.

COLUMNS	ODD	EVEN
L - 2	Card Number	Card Number
4 - 6	Blank	Blank except for last card
9 - 11	Load address	Blank
20 - 69	Program or Data	1's for word marks

In an odd numbered card, up to fifty characters to be loaded are punched starting in column 20. In columns 9 through 11 is punched the address of the location in storage where the character in column 20 is to be stored. In columns 20 through 69 of an even numbered card are punched 1's for the word marks associated with the characters in columns 20 through 69 of the previous card. In columns 4 through 6 of the last card (even numbered) is punched the address at which execution is to begin.

d) Monitor END OF JOB cards - In versions C and D, ‡ ‡ ‡ ‡ END OF JOB cards may be used to facilitate continuous operation. In an SPS Assembly, if the last source program deck is followed by an END OF JOB card control is automatically returned to the 1620 Monitor and the next program, such as the 141 Simulator, can be called into storage for execution.

During the execution of a 141 program using the 141 Simulator, an END OF JOB card following the data cards will automatically cause a return to request a new function. This may be any 141 Simulator function, including the EXIT function which will return control to the 1620 Monitor.

e) 1443 Carriage Control - In versions B and D, no provisions are made for control of the 1443 printer carriage except for an automatic detection of a channel 12 punch which will skip the paper form to the channel 1 position.

```
00010*                                                          A
00020*
00030*
00040* INITIALIZATION AND STORE PROGRAM ROUTINE
00050*
00060 ASMBLY TFM   CDCNT,0
00070        TF    11,INIT+11
00080        BLC   *+12
00090        TFM   ERRCNT,0
00100        TFM   MOD+30,STORE
00110        TFM   ICTR,0333,8
00120        TFM   MADDR+6,LABEL-15
00130        TFM   MLABEL+6,LABEL-18
00140        TDM   OVERSW,0
00150        CF    IDENT-1
00160 LC     BLC   NOEND
00170        RACD  LAREA
00180        AM    CDCNT,1,10
00190        C     END+4,LAREA+30
00200        BE    MOD
00210        C     AST,LAREA+14
00220        BE    MOD
00230        C     CCTL,LAREA+30
00240        BE    MOD
00250        C     CEX,LAREA+30
00260        BE    MOD
00270        C     CDCW-2,LAREA+28
00280        BE    DCDSR
00290        C     CDSA-2,LAREA+28
00300        BE    DCDSR
00310        C     CORG,LAREA+30
00320        BE    ORGR
00330        C     CB,LAREA+30
00340        BNE   *+60
00350        C     BLANK,LAREA+64
00360        BE    *+36
00370        TFM   CNT,8,9
00380        B     REPL
00390        TFM   CNT,0,9
00400        BD    INCR,LAREA+75
00410        BD    INCR,LAREA+76
00420        B     *+24
00430 INCR   AM    CNT,1,10
00440        C     BLANK,LAREA+64
00450        BNE   *+60
00460        C     BLANK,LAREA+42
00470        BNE   *+60
00480        AM    CNT,1,10
00490        B     REPL
00500        AM    CNT,7,10
00510        B     REPL
00520        AM    CNT,4,10
00530 REPL   TD    LAREA+12,CNT
00540        TDM   LAREA+11,7
00550        TD    LAREA+10,CNT-1
00560        C     BLANK,LAREA+24
```

```
00570          BE    REPLIM
00580          TF    LOC,ICTR
00590          BTM   LTABLE
00600 REPLIM   A     ICTR,CNT
00610          BD    MOD,OVERSW
00620          BD    OVERR,ICTR-3
00630 MOD      CM    CDCNT,100
00640          BH    *+48
00650          TF    0,LAREA+108
00660          AM    *-6,110,9
00670          B     *+24
00680          WACD  LAREA
00690          C     END+4,LAREA+30
00700          BE    PASS2
00710          B     LC
00720          DC    5,0
00730 LTABLE   AM    MLABEL+6,15,10
00740          AM    MADDR+6,15,10
00750          CM    MADDR+6,LABEL+15*90
00760          BNL   LBLERR
00770          SF    LAREA+13
00780 MLABEL   TF    0,LAREA+24
00790          SF    LOC-2
00800 MADDR    TF    0,LOC
00810          CF    LAREA+13
00820          BB
00830 LBLERR   RCTY
00840          WATY  LBLMSG
00850          B     OVERR+36
00860 LBLMSG   DAC   18,LABEL TABLE FULL.@,
00870 ORGR     TD    ICTR,LAREA+38
00880          TD    ICTR-1,LAREA+36
00890          TD    ICTR-2,LAREA+34
00900          B     MOD
00910 DCDSR    TD    CNT,LAREA+12
00920          TD    CNT-1,LAREA+10
00930          C     CDSA,LAREA+30
00940          BNE   *+48
00950          TF    CNT,C3
00960          TFM   LAREA+12,0073,8
00970          CF    LAREA+9
00980          C     AST,LAREA+32
00990          BNE   ABSLT
01000          A     ICTR,CNT
01010          C     BLANK,LAREA+24
01020          BE    REPLIM+12
01030          TF    LOC,ICTR
01040          SM    LOC,1,10
01050          BTM   LTABLE
01060          B     REPLIM+12
01070 ABSLT    TD    LOC,LAREA+38
01080          TD    LOC-1,LAREA+36
01090          TD    LOC-2,LAREA+34
01100          BTM   LTABLE
01110          B     MOD
01120 NOEND    RCTY
```

COMPUTER
TECHNOLOGY

```
01130          WATY ENDMSG
01140          RCTY
01150          H
01160          B    LC+12
01170 OVERR    TDM  OVERSW,1
01180          RCTY
01190          WATY OVMSG
01200          RCTY
01210          WATY LAREA
01220          RCTY
01230          B    MOD
01240 LAREA    DAC  50,                                                    ,
01250          DS   10
01260 LDIN     DAC  20,L       1056          ,
01270 IDENT    DAC  7,       @,
01280 ADDRAR   DC   5,@,
01290 ICTR     DC   4,0
01300 BLANK    DC   12,0
01310 LOC      DC   4,0
01320 CNT      DC   3,0
01330 LABEL    DSB  15,90
01340 CDCNT    DC   5,0
01350          DC   1,@
01360 ERRCNT   DC   5,0
01370          DC   1,@
01380 STORE    DSB  110,100,9109
01390 END      DAC  3,END,
01400 ENDMSG   DAC  48,END CARD MISSING. LOAD END CARD AND PUSH START.@,
01410 OVMSG    DAC  22,PROGRAM EXCEEDS CORE.@,
01420*
01430* PASS2
01440*
01450 PASS2    BD   ASMBLY,OVERSW
01460          TFM  ICTR,0332,8
01470          CM   CDCNT,100
01480          BNH  *+60
01490          RCTY
01500          WATY P2MSG
01510          RCTY
01520          H
01530          TFM  CDCNT,0
01540          TFM  PULIM+11,STORE
01550          SF   LAREA+149
01560          TF   CS1+158,IDENT+8
01570          TF   CS2+158,IDENT+8
01580          TF   BS+158,IDENT+8
01590          BC2  PCS
01600          RCTY
01610          WATY CS1
01620          RCTY
01630          WATY CS2
01640          RCTY
01650          WATY BS
01660          RCTY
01670          RCTY
01680 PCS      BC1  LOOP2
```

```
01690          WACD CS1
01700          WACD CS2
01710          WACD BS
01720 LOOP2    CM   CDCNT,100
01730          BL   *+48
01740          RACD LAREA
01750          TF   IDENT+8,BS+158
01760          B    *+24
01770 PULIM    TF   LAREA+108,0
01780          TF   LDIN+38,CLDIN+38
01790          AM   CDCNT,1,10
01800          TD   CNT,LAREA+12
01810          TD   CNT-1,LAREA+10
01820          TDM  ERRSW,0
01830          C    AST,LAREA+14
01840          BE   ORGR2+48
01850          C    END+4,LAREA+30
01860          BE   ENDCD
01870          C    CORG,LAREA+30
01880          BE   ORGR2
01890          C    CCTL,LAREA+30
01900          BE   ORGR2+48
01910          C    CEX,LAREA+30
01920          BE   EXR2
01930          TF   ADDRAR-1,ICTR
01940          AM   ADDRAR-1,1,10
01950          A    ICTR,CNT
01960          TFM  LDIN+11,70707
01970          TD   LDIN+12,ICTR
01980          TD   LDIN+10,ICTR-1
01990          TD   LDIN+8,ICTR-2
02000          C    CDCW,LAREA+30
02010          BE   DCWR2
02020          C    CDC,LAREA+30
02030          BE   DCWR2-12
02040          C    CDS,LAREA+30
02050          BE   DSR
02060          C    CDSA,LAREA+30
02070          BE   DSAR
02080          SF   LAREA+10
02090          TF   WA,CNT
02100          AM   WA,66,10
02110          TFM  LDIN+5,70707
02120          TD   LDIN+6,WA
02130          TD   LDIN+4,WA-1
02140          TD   LDIN+2,WA-2
02150          BTM  TABLE,0
02160          C    C8,CNT
02170          BNE  *+84
02180          TFM  DMOD+6,LDIN+36
02190          TFM  DMOD+18,LDIN+35
02200          BTM  DMOD,0
02210          BTM  BADD,0
02220          BTM  AADD,0
02230          B    TESTSW
02240          C    C7,CNT
```

```
02250           BNE   *+48
02260           BTM   BADD
02270           BTM   AADD
02280           B     TESTSW
02290           C     C5,CNT
02300           BNE   *+60
02310           TFM   DMOD+6,LDIN+30
02320           TFM   DMOD+18,LDIN+29
02330           BTM   DMOD
02340           B     *-84
02350           C     C4,CNT
02360           BE    *-108
02370           C     C2,CNT
02380           BNE   TESTSW
02390           TFM   DMOD+6,LDIN+24
02400           TFM   DMOD+18,LDIN+23
02410           BTM   DMOD,0
02420           B     TESTSW
02430           TFM   LDIN,54,10
02440 DCWR2     C     TT,CNT
02450           BL    *+36
02460           C     BLANK-10,CNT
02470           BL    *+72
02480           AM    ERRCNT,1,10
02490           TDM   ERRSW,1
02500           TF    LDIN+12,LBS
02510           TF    LDIN+6,LBS
02520           B     TESTSW
02530           C     BSIGN,LAREA+44
02540           BNE   MINUS+12
02550           SF    LAREA+10
02560           TFM   MINUS+6,LAREA+43
02570           A     MINUS+6,CNT
02580           A     MINUS+6,CNT
02590 MINUS     TDM   0,5
02600           TFM   T24,23,9
02610           A     T24,CNT
02620           TF    LDIN+6,ZERO
02630           TD    LDIN+6,T24
02640           TD    LDIN+4,T24-1
02650           TD    LDIN+2,T24-2
02660           C     AST,LAREA+32
02670           BE    AAA
02680           S     ICTR,CNT
02690           TD    LDIN+12,LAREA+38
02700           TD    LDIN+10,LAREA+36
02710           TD    LDIN+8,LAREA+34
02720 AAA       TD    ADDRAR-1,LDIN+12
02730           TD    ADDRAR-2,LDIN+10
02740           TD    ADDRAR-3,LDIN+8
02750           B     TESTSW
02760 DSR       TFM   LDIN,55,10
02770           TF    LDIN+28,LDIN+12
02780           TF    LDIN+12,BRRD
02790           TFM   LDIN+22,70,10
02800           C     AST,LAREA+32
```

```
02810           BE    *+60
02820           S     ICTR,CNT
02830           SF    LAREA+31
02840           TF    LDIN+28,LAREA+38
02850           CF    LAREA+31
02860           TD    ADDRAR-1,LDIN+28
02870           TD    ADDRAR-2,LDIN+26
02880           TD    ADDRAR-3,LDIN+24
02890           B     TESTSW
02900 DSAR      TFM   LDIN+6,7276,8
02910           BTM   BADD
02920           CF    LDIN+30
02930           TF    LAREA+50,LDIN+34
02940           CF    LAREA+45
02950           TF    LDIN+34,BLANK-6
02960           B     MINUS+72
02970           DC    5,0
02980 TABLE     C     CMCW,LAREA+30
02990           BE    INM
03000           C     CR,LAREA+30
03010           BE    INM+24
03020           C     CW,LAREA+30
03030           BE    INM+48
03040           C     CP,LAREA+30
03050           BE    INM+72
03060           C     CSW,LAREA+30
03070           BE    INM+96
03080           C     CCW,LAREA+30
03090           BE    INM+120
03100           C     CA,LAREA+30
03110           BE    INM+144
03120           C     CS,LAREA+30
03130           BE    INS
03140           C     CC,LAREA+30
03150           BE    INS+24
03160           C     CH,LAREA+30
03170           BE    INS+48
03180           C     CB,LAREA+30
03190           BE    INS+72
03200           C     CCS,LAREA+30
03210           BE    INS+96
03220           C     CLCA,LAREA+30
03230           BE    INS+120
03240           C     CNOP,LAREA+30
03250           BE    INS+144
03260           TF    LDIN+22,LBS-4
03270           B     INLBS+12
03280 INM       TFM   LDIN+22,54,10
03290           BB
03300           TFM   LDIN+22,71,10
03310           BB
03320           TFM   LDIN+22,72,10
03330           BB
03340           TFM   LDIN+22,74,10
03350           BB
03360           TFM   LDIN+22,23,10
```

```
03370          BB
03380          TFM   LDIN+22,04,10
03390          BB
03400          TFM   LDIN+22,41,10
03410          BB
03420 INS      TFM   LDIN+22,62,10
03430          BB
03440          TFM   LDIN+22,43,10
03450          BB
03460          TFM   LDIN+22,03,10
03470          BB
03480          TFM   LDIN+22,42,10
03490          BB
03500          TFM   LDIN+22,21,10
03510          BB
03520          TFM   LDIN+22,53,10
03530          BB
03540          TFM   LDIN+22,55,10
03550          BB
03560* DMOD ROUTINE
03570          DC    5,0
03580 DMOD     TD    LDIN+36,LAREA+76
03590          TD    LDIN+35,LAREA+75
03600          BB
03610*B ADDRESS ROUTINE
03620          DC    5,0
03630 BADD     BD    *+36,LAREA+54
03640          BD    *+24,LAREA+53
03650          B     INLBS
03660          C     S9,LAREA+54
03670          BL    BINACT
03680          C     AST,LAREA+54
03690          BNE   *+72
03700          TF    LDIN+34,ZERO
03710          TD    LDIN+34,ICTR
03720          TD    LDIN+32,ICTR-1
03730          TD    LDIN+30,ICTR-2
03740          B     BCADJ
03750          TFM   LEXIT+6,BCADJ
03760          TF    LDIN+34,ZERO
03770          TFM   LOOK+23,LAREA+64
03780          TFM   XX+6,LDIN+34
03790          B     LOOK
03800 BCADJ    C     BLANK-6,LAREA+74
03810          BNE   ADJB
03820          BB
03830 INLBS    TF    LDIN+34,LBS
03840          AM    ERRCNT,1,10
03850          TDM   ERRSW,1
03860          BB
03870 BINACT   TF    LDIN+34,ZERO
03880          TD    LDIN+34,LAREA+60
03890          TD    LDIN+32,LAREA+58
03900          TD    LDIN+30,LAREA+56
03910          B     BCADJ
03920 ADJB     TD    WA1,LAREA+72
```

```
03930           TD    WA1-1,LAREA+70
03940           TD    WA1-2,LAREA+68
03950           SF    WA1-2
03960           TD    WA2,LDIN+34
03970           TD    WA2-1,LDIN+32
03980           TD    WA2-2,LDIN+30
03990           SF    WA2-2
04000           C     BSIGN,LAREA+66
04010           BNE   *+36
04020           S     WA2,WA1
04030           B     *+24
04040           A     WA2,WA1
04050           CF    WA2
04060           TD    LDIN+34,WA2
04070           TD    LDIN+32,WA2-1
04080           TD    LDIN+30,WA2-2
04090           BB
04100*A ADDRESS ROUTINE
04110           DC    5,0
04120 AADD      BD    *+36,LAREA+32
04130           BD    *+24,LAREA+31
04140           B     INLBSA
04150           C     S9,LAREA+32
04160           BL    AINACT
04170           C     AST,LAREA+32
04180           BNE   *+72
04190           TF    LDIN+28,ZERO
04200           TD    LDIN+28,ICTR
04210           TD    LDIN+26,ICTR-1
04220           TD    LDIN+24,ICTR-2
04230           B     ACADJ
04240           TFM   LEXIT+6,ACADJ
04250           TF    LDIN+28,ZERO
04260           TFM   LOOK+23,LAREA+42
04270           TFM   XX+6,LDIN+28
04280           B     LOOK
04290 ACADJ     C     BLANK-6,LAREA+52
04300           BNE   ADJA
04310           BB
04320 INLBSA    TF    LDIN+28,LBS
04330           B     INLBS+12
04340 AINACT    TF    LDIN+28,ZERO
04350           TD    LDIN+28,LAREA+38
04360           TD    LDIN+26,LAREA+36
04370           TD    LDIN+24,LAREA+34
04380           B     ACADJ
04390 ADJA      TD    WA1,LAREA+50
04400           TD    WA1-1,LAREA+48
04410           TD    WA1-2,LAREA+46
04420           SF    WA1-2
04430           TD    WA2,LDIN+28
04440           TD    WA2-1,LDIN+26
04450           TD    WA2-2,LDIN+24
04460           SF    WA2-2
04470           C     BSIGN,LAREA+44
04480           BNE   *+36
```

```
04490           S     WA2,WA1
04500           B     *+24
04510           A     WA2,WA1
04520           CF    WA2
04530           TD    LDIN+28,WA2
04540           TD    LDIN+26,WA2-1
04550           TD    LDIN+24,WA2-2
04560           BB
04570* LABEL TABLE LOOK UP
04580 LOOK      TFM   *+18,LABEL-3
04590           C     0,0
04600           BE    MVADDR
04610           C     MLABEL+6,LOOK+18
04620           BE    INSLB
04630           AM    LOOK+18,15,10
04640           B     LOOK+12
04650 MVADDR    TF    XX+11,LOOK+18
04660           AM    XX+11,3,10
04670           TF    XX+23,XX+11
04680 XX        TD    0,0
04690           BNF   *+24
04700 LEXIT     B     0
04710           SM    XX+6,2,10
04720           SM    XX+11,1,10
04730           SM    XX+23,1,10
04740           B     XX
04750 INSLB     TDM   ERRSW,1
04760           AM    ERRCNT,1,10
04770           TF    *+18,XX+6
04780           TF    0,LBS
04790           B     LEXIT
04800 TESTSW    CF    ADDRAR-3
04810           BD    PRINT,ERRSW
04820           BNC2  PRINT
04830           BC1   *+24
04840           WACD  LAREA
04850           AM    PULIM+11,110,9
04860           B     LOOP2
04870 ENDCD     TF    LDIN+12,ENDC
04880           BTM   AADD
04890           TF    LDIN+6,LDIN+28
04900           TF    LDIN+28,BLANK
04910           TF    LDIN+16,BLANK-8
04920           BD    *+24,ERRSW
04930           BC2   *+36
04940           RCTY
04950           WATY  LAREA
04960           BC1   *+24
04970           WACD  LAREA
04980           RCTY
04990           RCTY
05000           SPTY
05010           WNTY  CDCNT-2
05020           WATY  CNTMSG
05030           WNTY  ERRCNT-2
05040           WATY  ERRMSG
```

```
05050           RCTY
05060           BNLC  ASMBLY
05070           H
05080  INIT     B     ASMBLY,0,0
05090  ORGR2    TD    ICTR,LAREA+38
05100           TD    ICTR-1,LAREA+36
05110           TD    ICTR-2,LAREA+34
05120           SM    ICTR,1,10
05130           TFM   LDIN,55,10
05140           TF    LDIN+12,BRRD
05150           TF    ADDRAR-1,BLANK-8
05160           B     TESTSW
05170  EXR2     BTM   AADD
05180           TF    LDIN+6,LDIN+28
05190           TF    LDIN+28,BLANK-6
05200           TFM   LDIN,42,10
05210           TF    LDIN+12,BLANK-6
05220           B     TESTSW
05230  PRINT    WATY  LAREA
05240           C     BLANK-9,ADDRAR-1
05250           BE    *+24
05260           WNTY  ADDRAR-3
05270           RCTY
05280           B     TESTSW+36
05290  CS1      DAC   50,,008015,022026,030034,041,045,053,0570731026       ,
05300           DAC   31,                                          @,
05310  CS2      DAC   50,L072116,110106,105117B101/999,027A074028)027B00102,
05320           DAC   31,70B026/0991,001/00111710      @,
05330  BS       DAC   50,,008015,022029,056063/056029                       ,
05340           DAC   31,     ,0240671056              @,
05350  CLDIN    DAC   20,L0010561056          ,
05360  AST      DAC   1,*,
05370  P2MSG    DAC   46,PLACE CARDS PUNCHED IN READ FEED. PUSH START.@,
05380  CNTMSG   DAC   9, CARDS  @,
05390  ERRMSG   DAC   8, ERRORS@,
05400  ENDC     DC    14,21707070707870
05410  BRRD     DC    8,71707576
05420  ZERO     DC    6,707070
05430  CMCW     DC    6,544366
05440  CR       DC    6,590000
05450  CW       DC    6,660000
05460  CP       DC    6,570000
05470  CSW      DC    6,626600
05480  CCW      DC    6,436600
05490  CA       DC    6,410000
05500  CS       DC    6,620000
05510  CC       DC    6,430000
05520  CH       DC    6,480000
05530  CB       DC    6,420000
05540  CCS      DC    6,436200
05550  CLCA     DC    6,534341
05560  CNOP     DC    6,555657
05570  CDCW     DC    6,444366
05580  CDSA     DC    6,446241
05590  CDC      DC    6,444300
05600  CDS      DC    6,446200
```

```
05610 CORG    DC    6,565947
05620 CCTL    DC    6,436353
05630 CEX     DC    6,456700
05640 LBS     DC    6,333333
05650 TT      DC    3,32
05660 T24     DC    3,23
05670 S9      DC    2,69
05680 BSIGN   DC    2,20
05690 WA      DC    3,0
05700 C8      DC    3,8
05710 C7      DC    3,7
05720 C5      DC    3,5
05730 C4      DC    3,4
05740 C3      DC    3,3
05750 C2      DC    3,2
05760 WA1     DC    3,0
05770 WA2     DC    3,0
05780 OVERSW  DC    1,0
05790 ERRSW   DC    1,0
05800         DEND  ASMBLY
```

SYMBOL TABLE
1401 SPS-A

ASMBLY	00402	LC	00522	INCR	00846	REPL	00966	REPLIM	01050
MOD	01086	LTABLE	01200	MLABEL	01260	MADDR	01284	LBLERR	01320
LBLMSG	01357	ORGR	01392	DCDSR	01440	ABSLT	01632	NOEND	01692
OVERR	01752	LAREA	01837	LDIN	01947	IDENT	01987	ADDRAR	02004
ICTR	02008	BLANK	02020	LOC	02024	CNT	02027	LABEL	02042
CDCNT	03382	ERRCNT	03388	STORE	09109	END	03391	ENDMSG	03397
OVMSG	03493	PASS2	03536	PCS	03812	LOOP2	03860	PULIM	03920
DCWR2	04724	MINUS	04904	AAA	05060	DSR	05108	DSAR	05276
TABLE	05366	INM	05726	INS	05894	DMOD	06068	BADD	06110
BCADJ	06314	INLBS	06350	BINACT	06398	ADJB	06458	AADD	06680
ACADJ	06884	INLBSA	06920	AINACT	06944	ADJA	07004	LOOK	07220
MVADDR	07304	XX	07340	LEXIT	07364	INSLB	07424	TESTSW	07484
ENDCD	07568	INIT	07820	ORGR2	07832	EXR2	07928	PRINT	08000
CS1	08073	CS2	08235	BS	08397	CLDIN	08559	AST	08599
P2MSG	08601	CNTMSG	08693	ERRMSG	08711	ENDC	08739	BRRD	08747
ZERO	08753	CMCW	08759	CR	08765	CW	08771	CP	08777
CSW	08783	CCW	08789	CA	08795	CS	08801	CC	08807
CH	08813	CB	08819	CCS	08825	CLCA	08831	CNOP	08837
CDCW	08843	CDSA	08849	CDC	08855	CDS	08861	CORG	08867
CCTL	08873	CEX	08879	LBS	08885	TT	08888	T24	08891
S9	08893	BSIGN	08895	WA	08898	C8	08901	C7	08904
C5	08907	C4	08910	C3	08913	C2	08916	WA1	08919
WA2	08922	OVERSW	08923	ERRSW	08924				

A

```
00010*
00020*
00030*
00040* INITIALIZER ROUTINE
00050*
00060 BEGIN   TR    19998,ASK+41
00070         TF    11,PRELD+11
00080         SF    17982
00090         BC4   INITZR
00100         RCTY
00110         WATY  HEADG
00120         RCTY
00130         BTM   WRT,WORD
00140         BTM   WRT,WORD+10
00150         BTM   WRT,WORD+22
00160         BTM   WRT,WORD+34
00170         BTM   WRT,WORD+44
00180         B     INITZR
00190 WRT     BC4   INITZR
00200         RCTY
00210         WATY  FUNCT,,2
00220         BC4   INITZR
00230         TF    *+18,WRT-1
00240         WATY  0
00250         AM    WRT+30,80,10
00260         BB
00270 INITZR  RCTY
00280         RCTY
00290         WATY  ASK
00300         RATY  TESTL
00310         SF    TESTL-1
00320         C     TESTL+6,WORD+6
00330         BE    START
00340         C     TESTL+8,WORD+18
00350         BE    CLEAR
00360         C     TESTL+8,WORD+30
00370         BE    ALTER
00380         C     TESTL+6,WORD+40
00390         BE    DSTART
00400         C     TESTL+12,WORD+56
00410         BE    INBRCH
00420         WATY  INERR
00430         RCTY
00440         RCTY
00450         B     INITZR
00460 INBRCH  WATY  BGMSG
00470         RNTY  TESTL-1
00480         TD    17985,TESTL-1
00490         TD    17987,TESTL
00500         TD    17989,TESTL+1
00510         RCTY
00520         RCTY
00530         SF    17990
00540         B     B
00550 TESTL   DAC   10,LOAD      ,
00560 HEADG   DAC   36,FUNCTIONS PERFORMED      ,
```

```
00570          DAC   18,REQUEST BY TYPING@,
00580 FUNCT    DAC   40,    LOAD PROGRAM FROM CARD READER        @,
00590          DAC   40,    CLEAR 141 STORAGE                    @,
00600          DAC   40,    ALTER STORAGE FROM TYPEWRITER        @,
00610          DAC   40,    DUMP CONTENTS OF 141 STORAGE         @,
00620          DAC   40,    BEGIN EXECUTION OF PROGRAM           @,
00630 WORD     DAC   5,LOAD@,
00640          DAC   6,CLEAR@,
00650          DAC   6,ALTER@,
00660          DAC   5,DUMP@,
00670          DAC   8,EXECUTE@,
00680 ASK      DAC   23,REQUESTED FUNCTION IS @,
00690 BGMSG    DAC   15, BEGINNING AT @,
00700 INERR    DAC   24,  INVALID REQUEST WORD.@,
00710*
00720* LOADER ROUTINE
00730*
00740 START    RCTY
00750          RCTY
00760          BC3   LDUMP
00770          TF    18161,BLANKS
00780          TF    18141,BLANKS
00790          TF    18121,BLANKS
00800          TF    18101,BLANKS
00810          TF    18081,BLANKS
00820          TF    18061,BLANKS
00830          TF    18041,BLANKS
00840          TF    18021,BLANKS-1
00850          RACD  18003
00860          TFM   FTEST+11,18002
00870          B     NEXTIN
00880*
00890* INSTRUCTION ACCESS ROUTINE
00900*
00910 NEXTIN   BNC1  *+60
00920          BTM   CVTREG,0,10
00930          TF    *+35,17983
00940          TF    *+18,IREG-1
00950          H     0,0
00960          AM    FTEST+11,2,10
00970          BT    TESTHI,FTEST+11
00980 FTEST    BNF   *+48,0,7
00990          TF    *+23,FTEST+11
01000          TF    17984,0
01010          B     TABLE
01020          AM    FTEST+11,6,10
01030          BT    TESTHI,FTEST+11
01040          TF    *+23,FTEST+11
01050          TF    17990,0
01060          TF    *+23,FTEST+11
01070          BNF   *+24,0
01080          B     TABLE
01090          CM    17983,42,10
01100          BNE   *+72
01110          BD    *+60,17990
01120          TF    *+35,FTEST+11
```

```
01130          AM    *+23,1,10
01140          BD    *+24,0
01150          B     B+12
01160          AM    FTEST+11,2,10
01170          BT    TESTHI,FTEST+11
01180          TF    *+23,FTEST+11
01190          BNF   *+48,0
01200          TF    *+23,FTEST+11
01210          TF    17992,0
01220          B     TABLE+288
01230          AM    FTEST+11,4,10
01240          BT    TESTHI,FTEST+11
01250          TF    *+23,FTEST+11
01260          TF    17996,0
01270          TF    *+23,FTEST+11
01280          BNF   *+24,0
01290          B     TABLE+96
01300          CM    17983,23,10
01310          BE    SW
01320          CM    17983,21,10
01330          BE    CS-60
01340          AM    FTEST+11,2,10
01350          BT    TESTHI,FTEST+11
01360          TF    *+23,FTEST+11
01370          BNF   *-36,0
01380          TF    *+23,FTEST+11
01390          TF    17998,0
01400          B     TABLE+288
01410* TEST FOR WRAP-AROUND OFF HIGH END OF CORE.
01420          DC    5,0
01430 TESTHI   CM    *-1,20000
01440          BNL   *+24
01450          BB
01460          RCTY
01470          WATY HIMSG
01480          RCTY
01490          H
01500          B     DSTART
01510 HIMSG    DAC   47,HI LIMIT OF CORE EXCEEDED. PUSH START TO DUMP.@,
01520*
01530* TABLE SEARCH FOR OPERATIONAL SUBROUTINE
01540*   TABLE ORDER - R,W,P,H,SW,A,S,CS,CW,MCW,C,LCA,B,NOP.
01550*
01560 TABLE    CM    17983,71,10
01570          BE    R
01580          CM    17983,72,10
01590          BE    W
01600          CM    17983,74,10
01610          BE    P
01620          CM    17983,03,10
01630          BE    H
01640          CM    17983,23,10
01650          BE    SW
01660          CM    17983,41,10
01670          BE    A
01680          CM    17983,62,10
```

```
01690          BE    S
01700          CM    17983,21,10
01710          BE    CS-84
01720          CM    17983,04,10
01730          BE    CW
01740          CM    17983,54,10
01750          BE    MCW
01760          CM    17983,43,10
01770          BE    C
01780          CM    17983,53,10
01790          BE    LCA
01800          CM    17983,42,10
01810          BE    B
01820          CM    17983,55,10
01830          BE    NEXTIN
01840* INVALID OP CODE ROUTINE
01850 ERROR1   RCTY
01860          WATY  OPMSG
01870          RCTY
01880          B     CORLIM+36
01890 OPMSG    DAC   41,INVALID INSTRUCTION. PUSH START TO DUMP.@,
01900*
01910* OPERATIONAL SUBROUTINES
01920*
01930* WRITE SUBROUTINE
01940 W        TFM   *+23,18561
01950          C     ZEROES-38,0
01960          BNE   RE
01970          SM    W+23,2,10
01980          CM    W+23,18401
01990          BNE   W+12
02000          B     SECL
02010 RE       AM    W+23,2,10
02020          TF    *+47,W+23
02030          TF    *+42,W+23
02040          TF    *+54,W+23
02050          TD    *+47,0
02060          TD    0,400
02070          WATY  18403
02080          TDM   0,0
02090 SECL     RCTY
02100          TD    *+59,18562
02110          BV    *+12
02120          SF    18562
02130          C     18601,ZEROES
02140          TDM   18562,0
02150          BNE   *+36
02160          BV    *+24
02170          B     B-24
02180          TD    *+47,18603
02190          TD    18603,400
02200          WATY  18563
02210          TDM   18603,0
02220          RCTY
02230          B     B-24
02240 ZEROES   DC    40,0
```

```
02250* READ A CARD  SUBROUTINE
02260 R       RACD 18003
02270         B    B-24
02280* PUNCH A CARD SUBROUTINE
02290 P       WACD 18203
02300         B    B-24
02310* HALT SUBROUTINE
02320 H       BTM  CVTREG,0,10
02330         TF   *+35,17983
02340         TF   *+18,IREG-1
02350         H    0,0
02360         B    B-24
02370* SET WORD MARK SUBROUTINE
02380 SW      BTM  CONVTA
02390         TF   *+30,17989
02400         SM   *+18,1,10
02410         SF   0
02420         BNF  *+24,17990
02430         B    NEXTIN
02440         BTM  CONVTB
02450         TF   *+30,17995
02460         SM   *+18,1,10
02470         SF   0
02480         B    NEXTIN
02490* CLEAR WORD MARK SUBROUTINE
02500 CW      BTM  CONVTA
02510         TF   *+30,17989
02520         SM   *+18,1,10
02530         CF   0
02540         BNF  *+24,17990
02550         B    NEXTIN
02560         BTM  CONVTB
02570         TF   *+30,17995
02580         SM   *+18,1,10
02590         CF   0
02600         B    NEXTIN
02610* MOVE CHARACTER TO A OR B FIELD WORD MARK SUBROUTINE
02620 MCW     BTM  CONVTA
02630         TF   MOVE+11,17989
02640         TF   MOVE+23,17989
02650         SM   MOVE+23,1,10
02660         BTM  CONVTB
02670         TF   MOVE+6,17995
02680         TF   MOVE+18,17995
02690         SM   MOVE+18,1,10
02700         TF   *+23,MOVE+18
02710         BNF  MOVE,0
02720         TDM  SFCF+1,2
02730         TDM  MOVE+25,9
02740 MOVE    TD   0,0
02750         TD   0,0
02760         NOP  SFCF-24
02770         TF   *+23,MOVE+18
02780         BNF  SFCF+24,0
02790         TDM  SFCF+1,3
02800         TDM  MOVE+25,1
```

```
02810          TF    *+18,MOVE+18
02820 SFCF     SF    0,0
02830          B     NEXTIN
02840          SM    MOVE+6,2,10
02850          SM    MOVE+11,2,10
02860          SM    MOVE+18,2,10
02870          SM    MOVE+23,2,10
02880          CM    MOVE+18,18000
02890          BL    CORLIM
02900          CM    MOVE+23,18000
02910          BNL   MOVE-48
02920 CORLIM   RCTY
02930          WATY  CORMSG
02940          RCTY
02950          BTM   CVTREG,0,10
02960          TF    *+35,17983
02970          TF    *+18,IREG-1
02980          H     0,0
02990          B     DSTART
03000 CORMSG   DAC   48,LOW LIMIT OF CORE EXCEEDED. PUSH START TO DUMP.@,
03010* COMPARE SUBROUTINE
03020 C        BTM   CONVTA
03030          TF    *+47,17989
03040          BTM   CONVTB
03050          TF    *+18,17995
03060          C
03070          BNH   *+36
03080 HIGH     SF    HIGH
03090          B     *+24
03100          CF    HIGH
03110          BNE   *+36
03120 EQUAL    SF    EQUAL
03130          B     *+24
03140          CF    EQUAL
03150          B     NEXTIN
03160* BRANCH SUBROUTINE
03170          BNF   B+12,17984
03180          B     NEXTIN
03190 B        BNF   DMOD,17990
03200          BTM   CONVTA
03210          TF    FTEST+11,17989
03220          SM    FTEST+11,1,10
03230          TF    *+23,FTEST+11
03240          BNF   ERROR1
03250          B     NEXTIN
03260 DMOD     BNF   BCE,17992
03270          SF    17990
03280          CM    17991,21,10
03290          BE    SLASH
03300          CM    17991,62,10
03310          BE    SAME
03320          CM    17991,63,10
03330          BE    TINY
03340          CM    17991,64,10
03350          BE    UPPER
03360          B     ERROR1
```

```
03370 SLASH   BNF   B+12,EQUAL
03380         B     NEXTIN
03390 SAME    BNF   NEXTIN,EQUAL
03400         B     B+12
03410 TINY    BNF   *+24,EQUAL
03420         B     NEXTIN
03430         BNF   B+12,HIGH
03440         B     NEXTIN
03450 UPPER   BNF   *+24,EQUAL
03460         B     NEXTIN
03470         BNF   NEXTIN,HIGH
03480         B     B+12
03490 BCE     SF    17996
03500         BTM   CONVTB
03510         TF    *+23,17995
03520         C     17997,0
03530         BE    B+12
03540         B     NEXTIN
03550* ADD SUBROUTINE
03560 A       TFM   ADD+1,21,10
03570         B     *+24
03580* SUBTRACT SUBROUTINE
03590 S       TFM   ADD+1,22,10
03600         BTM   CONVTA
03610         BNF   *+36,17990
03620         TF    17995,17989
03630         B     *+24
03640         BTM   CONVTB
03650         TFM   STRIPA+6,FIELDA-1
03660         TF    STRIPA+11,17989
03670         TF    STRIPA+35,17989
03680         SM    STRIPA+35,1,10
03690         TF    TSIGNA+11,STRIPA+35
03700         TFM   STRIPB+6,FIELDB-1
03710         TF    STRIPB+11,17995
03720         TF    STRIPB+35,17995
03730         SM    STRIPB+35,1,10
03740         TF    TSIGNB+11,STRIPB+35
03750         TF    SN-25,STRIPB+35
03760         TF    SN+6,STRIPB+35
03770         TFM   SN+47,FIELDB-2
03780         TF    SN-6,17995
03790         TF    SN+42,17995
03800         SM    SN+42,2,10
03810         TF    SN+59,17995
03820         SM    SN+59,3,10
03830         TF    SN+102,SN+59
03840 TSIGNB  TD    *+22,0
03850         CM    *+9,5000,8
03860         BE    STRIPB-12
03870         TF    *+23,17995
03880         C     *+22,0
03890         BE    STRIPB-12,,9
03900         TDM   FIELDB,0
03910         B     *+24
03920         TDM   FIELDB,0,11
```

```
03930 STRIPB TD    0,0
03940        SM    STRIPB+6,1,10
03950        BNF   *+60
03960        TF    POSCNT,17995
03970        S     POSCNT,STRIPB+11
03980        AM    POSCNT,1,10
03990        B     TSIGNA-24
04000        SM    STRIPB+11,2,10
04010        SM    STRIPB+35,2,10
04020        CM    STRIPB+6,FIELDB-33
04030        BE    ERROR2
04040        B     STRIPB
04050        TF    *+18,STRIPB+6
04060        TDM   0,0,11
04070 TSIGNA TD    *+22,0
04080        CM    *+9,5000,8
04090        BE    STRIPA-12
04100        TF    *+23,17989
04110        C     *+22,0
04120        BE    STRIPA-12,,9
04130        TDM   FIELDA,0
04140        B     *+24
04150        TDM   FIELDA,0,11
04160 STRIPA TD    0,0
04170        SM    POSCNT,1,10
04180        BNF   *+24
04190        B     ADD-24
04200        CM    POSCNT,0
04210        BE    ADD-24
04220        SM    STRIPA+6,1,10
04230        SM    STRIPA+11,2,10
04240        SM    STRIPA+35,2,10
04250        B     STRIPA
04260        TF    *+18,STRIPA+6
04270        SF    0,0
04280 ADD    H     FIELDB,FIELDA
04290        BNF   *+36,FIELDB
04300        TDM   SN+11,5
04310        B     *+24
04320        TDM   SN+11,7
04330        BNF   *+24,0
04340        SF    SN+11
04350        TD    0,FIELDB-1
04360 SN     TDM   0,0
04370        BNF   *+24,SN+11
04380        B     NEXTIN
04390        TD    0,0
04400        BNF   *+48,0
04410        TF    *+18,*+42
04420        TDM   0,7,11
04430        B     NEXTIN
04440        TDM   0,7
04450        SM    SN+42,2,10
04460        SM    SN+47,1,10
04470        SM    SN+59,2,10
04480        SM    SN+102,2,10
```

```
04490          B     SN+36
04500 ERROR2   RCTY
04510          WATY  AMSG
04520          RCTY
04530          B     CORLIM+36
04540 AMSG     DAC   47,B-FIELD OF ADD OR SUB INSTR OVER 32 POSITIONS. ,
04550          DAC   20,PUSH START TO DUMP.@,
04560 POSCNT   DC    5,0
04570 FIELDA   DS    33,
04580 FIELDB   DS    34,
04590* CLEAR STORAGE SUBROUTINE
04600          BNF   *+24,17990
04610          B     CS
04620          BTM   CONVTA
04630          TF    FTEST+11,17989
04640          SM    FTEST+11,1,10
04650          SF    17990
04660          TF    17989,17995
04670 CS       TFM   CS+210,18000
04680          TD    CS+248,17985
04690          A     CS+208,CS+248
04700          A     CS+208,CS+248
04710          TFM   CS+234,18000
04720          TD    CS+249,17987
04730          A     CS+233,CS+249
04740          A     CS+233,CS+249
04750          TFM   CS+191,BLANKS-19
04760          TD    CS+248,17989
04770          A     CS+191,CS+248
04780          A     CS+191,CS+248
04790          TD    CS+248,17987
04800          BTM   CONVTA
04810          TF    CS+186,17989
04820          TF    0,0
04830          BD    CS+228,CS+248
04840          CF    0,0
04850          B     NEXTIN
04860          TF    0,BLANKS
04870          SM    *+8,1,710
04880          SM    CS+234,20,10
04890          B     CS+192
04900* LOAD CHARACTERS TO A-FIELD WORD MARK SUBROUTINE
04910 LCA      BTM   CONVTA,0
04920          TF    LCA+59,17989
04930          BTM   CONVTB
04940          TF    LCA+54,17995
04950          TF    0,0
04960          B     NEXTIN
04970* CONVERT A SUBROUTINE TO CONVERT FROM 141 TO 1620 ADDRESSING
04980          DC    5,0
04990 CONVTA   TD    17988,17987
05000          TD    17987,17985
05010          TFM   17986,0,10
05020          A     17989,17989
05030          AM    17989,18001
05040          BB
```

```
05050* CONVERT B SUBROUTINE TO CONVERT FROM 141 TO 1620 ADDRESSING
05060          DC    5,0
05070 CONVTB   TD    17994,17993
05080          TD    17993,17991
05090          TFM   17992,0,10
05100          A     17995,17995
05110          AM    17995,18001
05120          BB
05130*
05140* CLEAR ROUTINE
05150*
05160 CLEAR    RCTY
05170          RCTY
05180          TFM   CLEAR+42,19999
05190          TF    19999,BLANKS,2
05200          SM    CLEAR+42,20,10
05210          CM    CLEAR+42,17999
05220          BNE   CLEAR+36
05230 PRELD    B     INITZR,,0
05240 BLANKS   DC    21,0
05250*
05260* DUMP ROUTINE
05270*
05280 DSTART   RCTY
05290          RCTY
05300          TF    OPREG+10,17983
05310          BTM   CVTREG,0,10
05320          RCTY
05330          WATY  TITLE
05340          RCTY
05350          SPTY
05360          WNTY  IREG-3
05370          WATY  OPREG
05380          RCTY
05390          RCTY
05400          CF    BLNKS-49
05410          CF    BLNKS-99
05420          TFM   CARDNO,0,10
05430          TFM   ADDR1,0,9
05440          TFM   ADDR2,49,9
05450          TFM   SAVC+11,18101
05460          TFM   INSRM+6,18101
05470          TFM   IN+18,18101
05480          TFM   INSRM+23,18000
05490 SAVC     TD    IN+23,0
05500 INSRM    TD    0,400
05510          TR    BANDA+37,0
05520          AM    CARDNO,01,10
05530          TD    BANDA,CARDNO-1
05540          TD    BANDA+2,CARDNO
05550          AM    CARDNO,01,10
05560          TD    BANDB,CARDNO-1
05570          TD    BANDB+2,CARDNO
05580          TDM   BANDA+137,0
05590          TD    BANDA+18,ADDR1-1
05600          TD    BANDA+16,ADDR1-2
```

```
05610           TD    BANDA+30,ADDR2-1
05620           TD    BANDA+28,ADDR2-2
05630           BNC2  *+48
05640           TDM   BANDA+138,0
05650           WACD  BANDA
05660           B     PWM
05670           BNC1  *+24
05680           H
05690 TYPE      TFM   *+23,BANDA+136
05700           C     ZEROES-38,0
05710           BNE   *+36
05720           SM    TYPE+23,2,10
05730           B     TYPE+12
05740           AM    TYPE+23,2,10
05750           TF    *+30,TYPE+23
05760           TF    *+42,TYPE+23
05770           TD    0,400
05780           WATY  BANDA+16
05790           TDM   0,0
05800           RCTY
05810 PWM       TF    BANDB+138,BLNKS
05820           TFM   TEST5+11,BANDA+37
05830           TFM   TEST5+18,BANDB+38
05840           TFM   INSRM2+6,BANDB+16
05850 TEST5     BNF   INCR,0,27
05860           TFM   0,71,10
05870           TF    INSRM2+6,*-6
05880 INCR      AM    TEST5+11,2,10
05890           AM    TEST5+18,2
05900           CM    TEST5+11,BANDA+137
05910           BNE   TEST5
05920 WRITE     BNC2  *+120
05930           CM    CARDNO,40,10
05940           BNE   *+60
05950           TFM   BANDB+9,70707
05960           TD    BANDB+10,IREG-1
05970           TD    BANDB+8,IREG-2
05980           TD    BANDB+6,IREG-3
05990           WACD  BANDB
06000           TF    BANDB+10,ZEROES-34
06010           B     *+60
06020           AM    INSRM2+6,2,10
06030 INSRM2    TD    0,400
06040           WATY  BANDB+16
06050           RCTY
06060           BD    OUT,SWENDD
06070 IN        AM    SAVC+11,100,9
06080           TDM   0,0
06090           TF    INSRM+6,SAVC+11
06100           TF    IN+18,SAVC+11
06110           AM    INSRM+23,100,9
06120           AM    ADDR1,50,10
06130           AM    ADDR2,50,10
06140           CM    SAVC+11,20001
06150           BNE   SAVC
06160           TD    1,400
```

```
06170          TDM   SWENDD,1
06180          TR    BANDA+37,19900
06190          B     INSRM+24
06200 OUT      TDM   1,9
06210          TDM   SWENDD,0
06220          B     INITZR
06230 BANDA    DAC   50,01       000 - 049                          ,
06240          DAC   30,                                  ,
06250 BANDB    DAC   50,02                                          ,
06260          DAC   30,                                  ,
06270          DC    22,0
06280          DC    50,0
06290 BLNKS    DC    50,0
06300 SWENDD   DC    1,0
06310 ADDR1    DC    3,0
06320 ADDR2    DC    3,49
06330 CARDNO   DC    2,0
06340* PRINT REGISTERS SUBROUTINE
06350 TITLE    DAC   14,I-REG  OP-REG@,
06360 IREG     DC    6,@,
06370 OPREG    DAC   7,      @,
06380 DIV      DC    6,0
06390          DC    5,0
06400 CVTREG   TF    IREG-1,FTEST+11
06410          SM    IREG-1,18000
06420          TF    DIV-1,IREG-1
06430          S     DIV,IREG-1
06440          S     DIV,IREG-1
06450          S     DIV,IREG-1
06460          S     DIV,IREG-1
06470          S     DIV,IREG-1
06480          TF    IREG-1,DIV-1
06490          BB
06500*
06510* ALTER ROUNTINE AND
06520* LOAD DUMP CARDS ROUTINE
06530*
06540 ALTER    WATY  BGMSG
06550          RNTY  TESTL-1
06560          SF    TESTL-1
06570          TF    FIRST+2,TESTL+1
06580          TDM   ALTSW,1
06590 NEXTL    RCTY
06600          RCTY
06610          CF    FIRST
06620          WNTY  FIRST
06630          SF    FIRST
06640          RCTY
06650          TFM   READ1+6,18001
06660          A     READ1+6,FIRST+2
06670          A     READ1+6,FIRST+2
06680 READ1    RATY  0
06690          RCTY
06700          RNTY  WMS+19
06710          TF    STFLG+6,READ1+6
06720          SM    STFLG+6,1,10
```

```
06730 COMMON TFM   TDIG+11,WMS+19
06740        TFM   TRM+11,WMS+19
06750 TRM    BNR   *+24,0
06760        B     RM
06770 TDIG   BD    *+36,0
06780        TDM   STFLG+1,3
06790        B     *+24
06800        TDM   STFLG+1,2
06810 STFLG  SF    0
06820        AM    STFLG+6,2,10
06830        AM    TDIG+11,1,10
06840        AM    TRM+11,1,10
06850        CM    TDIG+11,WMS+119
06860        BNE   TRM
06870 RM     SM    TDIG+11,WMS+19
06880        SF    TDIG+9
06890        A     FIRST+2,TDIG+11
06900        BD    NEXTL,ALTSW
06910        BD    EXEC,WMS+3
06920 LDUMP  SF    BANDC+16
06930        RACD  BANDC
06940        TD    BANDC+138,400
06950        BD    *+24,BANDC+19
06960        B     CDERR
06970        TD    FIRST+2,BANDC+20
06980        TD    FIRST+1,BANDC+18
06990        TD    FIRST,BANDC+16
07000        TFM   TR+6,18000
07010        A     TR+6,FIRST+2
07020        A     TR+6,FIRST+2
07030 TR     TR    0,BANDC+37
07040        RNCD  WMS
07050        TD    WMS+69,400
07060        BD    CDERR,WMS+16
07070        TF    STFLG+6,TR+6
07080        SF    WMS
07090        CM    WMS+1,40,10
07100        BE    *+48
07110        TF    *+30,TR+6
07120        AM    *+18,101,9
07130        TDM   0,0
07140        TDM   ALTSW,0
07150        B     COMMON
07160 EXEC   TD    17985,WMS+3
07170        TD    17987,WMS+4
07180        TD    17989,WMS+5
07190        SF    17990
07200        TFM   1,49,10
07210        B     B
07220 CDERR  WATY  CDMSG
07230        RCTY
07240        H
07250        B     START
07260 ALTSW  DC    1,0
07270 CDMSG  DAC   38,SEQUENCE ERROR. PUSH START TO RE-LOAD@,
07280 FIRST  DSC   4,000@,
```

```
07290 WMS     DSS  120
07300 BANDC   DAC  50,
07310         DAC  30,
07320         DEND BEGIN
```

SYMBOL TABLE
141 SIM-A

BEGIN	00402	WRT	00558	INITZR	00654	INBRCH	00882	TESTL	00991
HEADG	01011	FUNCT	01119	WORD	01519	ASK	01579	BGMSG	01625
INERR	01655	START	01702	NEXTIN	01870	FTEST	01954	TESTHI	02476
HIMSG	02573	TABLE	02666	ERROR1	03002	OPMSG	03051	W	03132
RE	03216	SECL	03312	ZEROES	03531	R	03532	P	03556
H	03580	SW	03640	CW	03772	MCW	03904	MOVE	04048
SFCF	04144	CORLIM	04264	CORMSG	04361	C	04456	HIGH	04528
EQUAL	04576	B	04648	DMOD	04732	SLASH	04864	SAME	04888
TINY	04912	UPPER	04960	BCE	05008	A	05080	S	05104
TSIGNB	05404	STRIPB	05512	TSIGNA	05680	STRIPA	05788	ADD	05932
SN	06028	ERROR2	06196	AMSG	06245	POSCNT	06382	FIELDA	06415
FIELDB	06449	CS	06534	LCA	06810	CONVTA	06888	CONVTB	06966
CLEAR	07038	PRELD	07122	BLANKS	07154	DSTART	07156	SAVC	07408
INSRM	07420	TYPE	07648	PWM	07792	TEST5	07840	INCR	07876
WRITE	07924	INSRM2	08056	IN	08104	OUT	08260	BANDA	08297
BANDB	08457	BLNKS	08737	SWENDD	08738	ADDR1	08741	ADDR2	08744
CARDNO	08746	TITLE	08749	IREG	08781	OPREG	08783	DIV	08801
CVTREG	08808	ALTER	08928	NEXTL	08988	READ1	09096	COMMON	09156
TRM	09180	TDIG	09204	STFLG	09252	RM	09324	LDUMP	09384
TR	09516	EXEC	09672	CDERR	09744	ALTSW	09792	CDMSG	09795
FIRST	09870	WMS	09874	BANDC	09995				

SYMBOL TABLE
141 SIM-A

BEGIN	00402	WRT	00558	INITZR	00654	INBRCH	00882	TESTL	00991		
HEADG	01011	FUNCT	01119	WORD	01519	ASK	01579	BGMSG	01625		
INERR	01655	START	01702	NEXTIN	01870	FTEST	01954	TESTHI	02476		
HIMSG	02573	TABLE	02666	ERROR1	03002	OPMSG	03051	W	03132		
RE	03216	SECL	03312	ZEROES	03531	R	03532	P	03556		
H	03580	SW	03640	CW	03772	MCW	03904	MOVE	04048		
SFCF	04144	CORLIM	04264	CORMSG	04361	C	04456	HIGH	04528		
EQUAL	04576	B	04648	DMOD	04732	SLASH	04864	SAME	04888		
TINY	04912	UPPER	04960	BCE	05008	A	05080	S	05104		
TSIGNB	05404	STRIPB	05512	TSIGNA	05680	STRIPA	05788	ADD	05932		
SN	06028	ERROR2	06196	AMSG	06245	POSCNT	06382	FIELDA	06415		
FIELDB	06449	CS	06534	LCA	06810	CONVTA	06888	CONVTB	06966		
CLEAR	07038	PRELD	07122	BLANKS	07154	DSTART	07156	SAVC	07408		
INSRM	07420	TYPE	07648	PWM	07792	TEST5	07840	INCR	07876		
WRITE	07924	INSRM2	08056	IN	08104	OUT	08260	BANDA	08297		
BANDB	08457	BLNKS	08737	SWENDD	08738	ADDR1	08741	ADDR2	08744		
CARDNO	08746	TITLE	08749	IREG	08781	OPREG	08783	DIV	08801		
CVTREG	08808	ALTER	08928	NEXTL	08988	READ1	09096	COMMON	09156		
TRM	09180	TDIG	09204	STFLG	09252	RM	09324	LDUMP	09384		
TR	09516	EXEC	09672	CDERR	09744	ALTSW	09792	CDMSG	09795		
FIRST	09870	WMS	09874	BANDC	09995						