ORCHARD 1-6171
OREGON 8-7161

THE RAMO-WOOLDRIDGE CORPORATION
8820 BELLANCA AVENUE
LOS ANGELES 45, CALIFORNIA

October 10, 1955

Recipients of the Ramo-Wooldridge Utility Routine Library:

The enclosed "Table of Contents" should replace the one currently in the notebook.

The enclosed description for FRI-0 is a revision and should replace the present FRI-0 description.

All thirteen pages of the section currently in the notebook entitled "General Description of ERA-1103 Operation Using the Service Routine Library" should be replaced by the enclosed section "The Utility Routine Library Handling Package".

The routines currently included as appendices to this section (that is, FRI-1, URT-0, and URT-1) should be replaced by the enclosed routines FRI-1, URT-3, and URT-1 respectively. The three new routines should be placed in their proper alphabetical position with the rest of the utility routines rather than as appendices to the "Utility Routine Library Handling Package".

The following enclosures are new additions to your library notebook:

CPO-1	Card Punch Output for Floating Point Numbers
DIE-0	Definite Integral Evaluation Routine
EXP-3	Floating Point Exponential Routine
LOG-2	Floating Point Natural Logarithm
MDP-1	Bioctal Memory Dump
NUI-3	Numerical Integration by the Gill Method
SIN-3	Floating Point Sine-Cosine Routine
SNI-1	Arcsine-Arcosine Routine
SNI-2	Floating Point Arcsine-Arcosine Routine
STT-0	Storage to Magnetic Tape Transfer
TST-0	Magnetic Tape to Storage Transfer

Walter F. Bauer

Walter F. Bauer, Head
Digital Computing Center
Computer Systems Division

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

UTILITY ROUTINE LIBRARY

Table of Contents

Interpolation and Curve Fitting Routines

Logarithmic, Exponential, and Rooting Routines

EXP-1	Approximation e^{-x} Routine
EXP-3	Floating Point Exponential Routine
LOG-2	Floating Point Natural Logarithm Routine
SQR-0	Square Root Routine

Matrix Routines

Statistical and Clerical Routines

Trigonometric Routines

SIN-0	Central Exchange Sine-Cosine Routine
SIN-1	Polynomial Multiply Sine-Cosine Routine
SIN-2	Small Angle Sine-Cosine Routine
SIN-3	Floating Point Sine-Cosine Routine
SNI-1	Arcsine-Arcosine Routine
SNI-2	Floating Point Arcsine-Arcosine Routine
TNI-0	Arctangent Routine
TNI-1	Floating Point Arctangent Routine

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Utility Routine Library Handling Package for Paper Tape Input

Normal Operation

During normal operation, the Service Routine Library is stored on the drum. In order to use one of the routines, control is transferred to one of the low-numbered drum addresses in the 40000b channel (see the list of "Service Routine Starting Addresses").

Details concerning the operation of these routines and their locations can be found in the write-ups.

MT Start

If, at any time, the library stored on MD is destroyed by a program, or because the drum interlace has been changed, or for some other reason, the entire library may be loaded onto MD from magnetic tape. Selecting MT Start and starting effects loading of the service routine library from MT zero. PAK is set to the FRI-0 starting address upon completion of the transfer. Selecting MT Start, setting $A_R > 0$, and starting effects loading of the service routine library and the assembly program and subroutines from MT zero. PAK is set to the CMP-0 starting address upon completion of the transfer.

Bootstrap

Since the Ferranti reader requires a programmed read in, it is necessary to "bootstrap" into the machine when no input routine is stored in memory. The procedure devised to load an input program involves the use of one binary card (since this method requires the fewest number of instructions to be loaded manually). It is necessary to key in manually only four words which perform the read in of one binary card (24 words) and transfer control to these 24 words.

This binary card contains a simplified Ferranti Input Routine (FRI-0) which then begins to read in the service routine library paper tape. This tape contains at its beginning the regular FRI-0 input routine and instructions transferring it to its proper location on MD. When FRI-0 has been loaded on MD control is transferred to it and FRI-0 reads in the remainder of the tape.

Following FRI-0 on the tape are the library, a Magnetic Tape to MD transfer routine (URT-1), and an MD to Magnetic Tape transfer routine (URT-3). When the complete paper tape has been read in the computer halts with the library loaded on MD. PAK is set to 40000 by an MD Start, the machine is started and URT-3 transfers the library and URT-1 to magnetic tape unit zero. The computer then halts with PAK set to 40001, the FRI-0 starting address.

Detailed Descriptions of Routines

Detailed descriptions, operating instructions, and codes for the routine mentioned above are included in the notebook.

Operating Instructions for the Bootstrap Procedure (Loads MD and MT zero with Service Routine Library)

1. Put the binary card deck in reader making sure that card reader is set for two fields and that all three switches on the reproducer are away from the card hoppers. Also be sure that there are cards in the punch hopper as the reader will not operate without them.

2. Put the library paper tape in the Ferranti Reader.

3. Position MT zero at the dead space immediately preceeding the first block. (Maintenance people will perform this function if requested).

4. Key in the following program:

00104	17 00000 00104	pick card
00105	17 00000 00105	read and pick card
00106	75 30030 00000	read one binary card
00107	76 10000 00000	then jump to 00000

5. START at 00104. The computer reads in one card containing the simplified Ferranti routine, then switches control to this routine which reads in FRI-0 and the necessary orders to transfer it to MD. After FRI-0 has been placed on MD, control is transferred to it and it reads in the library tape and the MT to MD and MT transfer routines. The computer then halts with the MS instruction 56 00000 40001.

6. Select MD Start

7. Set low order octal digit of Q to the address of the MT unit desired.

   To load tape zero, omit this step of the procedure.

8. START. The MD to MT routine loads the MT to MD routine, 40001b thru 40040b, 70000b thru 75777b, and 60000 thru 67777 onto MT zero. The computer then halts with the MS instruction 56 00000 40001, setting PAK to the FRI-0 starting address.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Card Punch Output for SNAP Numbers

Specifications

Identification Tag:	CPO-1
Type:	Subroutine
Assembly Routine Spec:	SUB 50490 20824 (see special Assembly Instructions)
Storage:	180 instructions, addresses 00K00 thru 07K32
	28 constants in program, addresses 02C00 thru 01C23
	43 words temporary storage required but not stored with the program
	208 words total program storage, addresses 00K00 thru 07K32, 02C00 thru 01C23
	10 words temporary storage pool used, addresses 00027b thru 00040b
	The constant pool is used by this routine
Program Entrances:	Addresses 00K02, 00K03
Program Exit:	Address 00K01
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Addresses 64372b thru 64711b
Machine Time:	0.5 seconds per card if both the entrance and the numbers are in ES

Coded by:	R. Summers	September 7, 1955
Code Checked by:	R. Summers	September 9, 1955
Machine Checked by:	R. Summers	September 14, 1955
Approved by:	W. Bauer	September 23, 1955

Description

This routine converts SNAP floating point numbers into floating decimal numbers and punches them onto cards. These cards may be used either for listing or for SNAP input.

Two entrances are provided; one requires a parameter word for each word to be punched and the other requires two parameter words for punching any number of words stored in consecutive cells.

The cards produced by this routine contain four numbers each. Each number consists of a ten-digit mantissa and a two-digit exponent. Each number has associated with it a five-digit decimal location number which may be used as an input address.

Assembly Instructions

Although only 208 instructions are assembled an additional 43 cells (bringing the required total to 251 cells) must be allowed for temporary storage.

This routine must operate from ES.

Programming Instructions I (to be followed when a parameter word is supplied for each cell to be dumped)

This routine may be assembled into a program by use of a "SUB" card as mentioned under "Assembly Instructions" above.

1. Enter the routine with an RJ instruction. Assuming that the routine was assigned to region 00K00 for assembly use the instruction RJ 00K01 00K02.

2. Enter the parameter words. If the RJ instruction was placed in cell n one parameter word for each cell to be dumped should be placed in cells n+1, n+2, n+3, etc.

   The parameter word is of the form

   00 M L.

   The u-address portion of the word (M) should contain the address of the number to be dumped. This may be any ES or drum address.

   The v-address portion of the word (L) should contain the location number. This should be the octal equivalent of the decimal location number that will be punched on the output card.

3. At the conclusion of the routine control will be transferred to the first cell (following cell n) in which the first octal digit is not a zero.

Alarm Conditions for "Programming Instructions I"

If (M) is not a true SNAP number (that is, if bit 27 of (M) is equal to zero) a partially blank card will be punched. "CPO-1" and the address of the RJ instructions used to enter the routine will be printed on the flexowriter.

At the time that the alarm halt occurs PAK will be set to the address of the cell immediately following the RJ instruction. It must be reset before starting again.

Programming Instructions II (to be followed when two parameter words are to be used to specify a dump of the contents of a number of consecutive cells)

This routine may be assembled into a program by use of a "SUB" card as mentioned under "Assembly Instructions" above.

1. Enter the routine with an RJ instruction. Assuming that the routine was assigned to region 00K00 for assembly use the instruction RJ 00K01 00K03.

2. Enter the two parameter words. If the RJ instruction was placed in cell n the parameter words should be placed as follows:

   n+1: 00 00000 p

   n+2: 00 M L

   M and L are defined as in item 2 of "Programming Instructions I".

   The numbers located in p consecutive cells (starting with address M) will be converted and punched. L will be advanced by one for each number.

3. At the conclusion of the routine control will be transferred to cell n+3.

Alarm Conditions for "Operating Instructions II"

If (M) is not a true SNAP number (that is, if bit 27 of (M) is equal to zero) a partially blank card will be punched. "CPO-1" and the address of cell n+2 will be printed on the flexowriter.

Starting at this time will cause a normal exit from the subroutine.

Last Card Punched

If the number of cells to be punched for a particular dump is not an even multiple of four the last card will contain blank fields to allow for the missing cells.

```
D                     01C00  01208
D                     02C00  01204
D                     00K00  01024
D                     01K00  01053
D                     02K00  01077
D                     03K00  01095
D                     05K00  01113
D                     06K00  01159
D                     07K00  01171
D                     00C00  00013
D                     00T00  01268
D                     01T00  00023
D                     41C00  50674
D                     42C00  50670
D                     40K00  50490
D                     41K00  50519
D                     42K00  50543
D                     43K00  50561
D                     44K00  50608
D                     45K00  50579
D                     04K00  01142
D                     46K00  50625
D                     47K00  50637
D                     1MG00  01232
D                     2MG00  01244
D                     3MG00  01256
D                     00D00  00460
40K00          37     75701  75702  B          ALARM EXIT
40K01          MJ     00000  00000             NORMAL EXIT
40K02          MJ     00000  01K00             ENTRANCE ONE
40K03          SP     00K01  00015             ENTRANCE TWO
40K04          TU     A0000  00K12
40K05          RA     00K01  01C04             SET EXIT
40K06          SS     00C03  00015             SET UP PAR
40K07          TU     A0000  00K09              WD ADDRESS
40K08          TP     00C00  01T09             ZERO TO F
40K09          TP     00000  Q0000             STORE PAR WD
40K10          QT     01C02  00T03             EXTRACT LOC
40K11          TU     Q0000  00K16             EXTRACT M
40K12          SP     00000  00000             4 VS N1
40K13          TJ     01C00  00K25             N1 MINUS 4
40K14          ST     01C00  01T08              TO N1
40K15          RP     30004  00K17             STORE 4
40K16          TP     00000  01T00              NUMBERS
40K17          RS     00T03  00C03
40K18          RP     10004  00K20             STORE 4
40K19          AT     00C03  01T04              LOCATIONS
40K20          RA     00T03  01C00             ADV LOC 4
40K21          RA     00K16  01C17             ADV M 4
```

```
40K22   RJ  07K24  02K00        RUN 1 CYCLE
40K23   SP  01T08  00000        TEST FOR END
40K24   ZJ  00K13  00K01         OF NUMBERS
40K25   TP  00C00  01T08        ZERO TO N1
40K26   TN  A0000  A0000        MINUS N1
40K27   AT  01C00  01T09        4 MINUS N1
40K28   MJ  00000  00K15         TO F
41K00   SP  00K01  00015
41K01   TU  A0000  01K05
41K02   TP  01C01  01T09        3 TO F
41K03   TV  00K19  01K08        SET INITIAL
41K04   TV  00K16  01K10         ADDRESSES
41K05   TP  00000  Q0000        PAR WD TO Q
41K06   QT  01C03  A0000        TEST FOR
41K07   ZJ  01K17  01K08        PARAMETER WD
41K08   QT  01C02  01T04        STORE LOC
41K09   TU  Q0000  01K10
41K10   TP  00000  01T00        STORE NUMBER
41K11   RA  01K05  00C02
41K12   RA  01K08  00C03
41K13   RA  01K10  00C03
41K14   IJ  01T09  01K05
41K15   RJ  07K24  02K00        RUN 1 CYCLE
41K16   MJ  00000  01K02        REPEAT
41K17   LQ  01K05  10021        SET UP EXIT
41K18   TV  Q0000  00K01
41K19   SP  01T09  00000        EXIT IF F
41K20   EJ  01C01  00K01         EQUALS ZERO
41K21   AT  00C03  01T09        ADD ONE TO F
41K22   TP  00C00  01T08        ZERO TO N1
41K23   MJ  00000  00K22
42K00   EF  3MG01  01C05        START CYCLE
42K01   RP  10036  02K03        CLEAR
42K02   TP  00C00  1MG00         CARD IMAGE
42K03   TP  01C01  00T01        SET W
42K04   TU  02C03  02K14        INITIAL N
42K05   TU  07K25  07K30         AND LOC ADR
42K06   TP  01C00  00T00        FLOATING ONE
42K07   TP  01C01  00T02        BREAK CNTR
42K08   TU  02K00  06K02        INITIAL CARD
42K09   TU  02C02  02C00         IMAGE
42K10   TU  02K00  07K00          ADDRESSES
42K11   RS  02K14  00C02        ADV NUMBER
42K12   TV  02K06  02C00
42K13   IJ  01T09  02K16        TEST F
42K14   TP  01T04  A0000
42K15   MJ  00000  03K00
42K16   QT  00C00  00C00        MAKE ROUTINE
```

```
42K17   TV  02K16  02C00          INOPERATIVE
43K00   TP  00C00  00T06        ZERO TO D
43K01   TV  02K16  07K00        SET UP
43K02   SJ  03K03  03K04         SIGN OF
43K03   TV  02K06  07K00          NUMBER
43K04   ZJ  03K05  04K21
43K05   TM  A0000  Q0000
43K06   QT  01C06  A0000        TEST FOR BIT
43K07   ZJ  03K08  03K15          IN 27
43K08   QT  01C07  00T04        M TO 33T34
43K09   LA  00T04  00008        SCALE TO 35
43K10   QT  01C08  A0000
43K11   SS  01C11  00008
43K12   TP  B0000  A0000
43K13   TP  A0000  00T05        B TO 33T35
43K14   SJ  05K00  04K00
43K15   RJ  07K24  07K20        PUNCH CARD
43K16   11  01C09  75756  BRB   TAG WORD
43K17   MJ  00000  00K00        TO ALARM
45K00   TM  A0000  A0000        TEST ABS B
45K01   TJ  01C19  05K07         AGAINST 34
45K02   RS  00T06  01C12        D MINUS 10
45K03   MP  01C23  00T04        MULT NUMBER
45K04   TP  B0000  00T04        BY 10 TO 10
45K05   RA  00T05  01C19        REDUCE ABS B
45K06   MJ  00000  05K00
45K07   MP  A0000  01C10        COMPUTE D
45K08   TP  B0000  Q0000
45K09   TN  B0000  A0000        COMPUTE
45K10   AT  00T06  00T06         TOTAL D
45K11   RA  00T05  00C05        SET UP
45K12   TV  A0000  05K27         REPEAT MULT
45K13   SP  Q0000  00015        SET UP
45K14   RJ  05K28  05K22         SHIFT INSTR
45K15   MP  A0000  01C13         MULTIPLY BY
45K16   TP  B0000  A0000         10 TO 10TH
45K17   MJ  00000  04K15
45K18   TP  A0000  Q0000
45K19   RS  00T06  00C03
45K20   MP  Q0000  01C12
45K21   MJ  00000  04K15
45K22   TU  A0000  05K24
45K23   SP  00C03  00000        COMPUTE 10
45K24   RP  00000  05K26
45K25   MP  A0000  01C12
45K26   MP  A0000  00T04        MULTIPLY AND
45K27   LA  A0000  00000         SHIFT
45K28   MJ  00000  00000
```

```
44K00   TJ  01C22  04K06      TEST B VS 33
44K01   RA  00T06  01C12      ADD 10 TO D
44K02   MP  00T04  01C20      MULT BY 2 TO
44K03   TP  B0000  00T04       33 OVER TEN
44K04   RS  00T05  01C22       TO 10TH AND
44K05   MJ  00000  04K00      REDUCE B
44K06   TV  A0000  05K27      SET UP SHIFT
44K07   MP  A0000  01C10      COMPUTE D
44K08   TP  B0000  00T05       AND STORE
44K09   RA  00T05  00C03
44K10   AT  00T06  00T06       REPEAT
44K11   SP  01C12  00000
44K12   SS  00T05  00015
44K13   RJ  05K28  05K22       TOTAL D
44K14   TP  B0000  A0000
44K15   TJ  01C21  05K18
44K16   TP  A0000  00T04
46K00   TP  00T06  A0000      SET UP SIGN
46K01   SJ  06K02  06K03       OF DECIMAL
46K02   CC  3MG01  00T00       EXPONENT
46K03   TM  00T06  A0000
46K04   DV  01C12  00T06      CONVERT D
46K05   TP  00019  Q0000       INTO
46K06   SP  A0000  00015       DECIAML
46K07   AT  02C00  06K08       DIGITS
46K08   CC  3MG02  00T00       AND
46K09   LA  00T00  00001       FORM
46K10   SP  00T06  00015       IMAGE
46K11   QJ  06K07  07K00
47K00   CC  3MG01  00T00      SET UP SIGN
47K01   TP  01C14  00T06       OF NUMBER
47K02   SP  00T04  00000
47K03   DV  01C12  00T04
47K04   SP  A0000  00015
47K05   AT  02C00  07K06
47K06   CC  3MG02  00T00
47K07   LQ  00T00  00001
47K08   IJ  00T02  07K14
47K09   TP  01C15  00T02
47K10   RS  06K02  01C16      MODIFY CARD
47K11   RS  07K00  01C16       FIELD IF
47K12   RS  02C00  01C16       NECESSARY
47K13   TP  00C03  00T00
47K14   IJ  00T06  07K02
47K15   RJ  07K15  07K16
47K16   RJ  07K15  07K30
47K17   RS  07K30  00C02      ADVANCE LOC
47K18   LQ  00T00  00002      2 BLANK COLS
```

```
47K19    IJ   00T01   02K11        LAST NUMBER
47K20    TP   01C18   00T00        SET ROW
47K21    RP   30003   07K23         COUNTER AND
47K22    TV   02C01   07K25         ADDRESSES
47K23    IJ   00T00   07K25        TEST END
47K24    MJ   00000   00000        EXIT
47K25    EW   00030   3MG11        PUNCH ONE
47K26    EW   10000   1MG11         ROW OF EACH
47K27    EW   10000   2MG11         FIELD
47K28    RP   20003   07K23        MODIFY ROWS
47K29    RS   07K25   00C03
47K30    TP   01T07   00T04        PREPARE TO
47K31    TP   01C00   00T06         CONVERT
47K32    MJ   00000   07K02         LOCATION
42C00    CC   3MG02   00T00
42C01    00   01T07   3MG11
42C02    00   3MG02   1MG11
42C03    00   01T04   2MG11
41C00    00   00000   00004   B
41C01    00   00000   00003   B
41C02    00   00000   77777   B    EXTRACTOR
41C03    70   00000   00000   B
41C04    00   00000   00002   B
41C05    00   00000   00112   B    EF CODE
41C06    00   04000   00000   B
41C07    00   07777   77777   B    M EXTRACTOR
41C08    77   70000   00000   B    B EXTRACTOR
41C09    16   15035   65204   B    TAG WORD
41C10    11   50404   65025   B    LOG 2 AT 35
41C11    20   00000   00000   B    BASE EXPON
41C12    00   00000   00012   B    TEN
41C13    11   24027   62000   B    TEN TO TENTH
41C14    00   00000   00011   B    NINE
41C15    00   00000   00033   B    BREAK CNTR
41C16    00   00014   00000   B    ROW CNTR
41C17    00   00004   00000   B
41C18    00   00000   00014   B    TWELVE
41C19    00   00000   00042   B    34
41C20    33   37157   73166   B
41C21    00   73465   45000   B
41C22    00   00000   00041   B    33
41C23    22   50057   44000   B
START         00D00
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Definite Integral Evaluation Routine

Specifications

Identification Tag:	DIE-0
Type:	Subroutine
Assembly Routine Spec:	SUB 49810 05804
Storage:	54 instructions, addresses 00P00 thru 00P07 00S00 thru 00S26 00N00 thru 00N18 4 constants in program, addresses 01C00 thru 01C03 58 words total program storage, addresses 00P00 thru 00P07 00S00 thru 00S26 00N00 thru 00N18 01C00 thru 01C03 10 words temporary storage pool used, addresses 00027b (00T00) thru 00040b (00T09) The constant pool is used by this routine
Drum Assignment:	Addresses 63122b thru 63213b
Program Entrance:	Address 00P02
Program Exit:	Address 00P01
Mode of Operation:	Fixed point

Coded by:	F. Meek	June 7, 1955
Code Checked by:	R. Bigelow	June 10, 1955
Machine Checked by:	F. Meek	July 7, 1955
Approved by:	W. Bauer	July 26, 1955

Description

Assuming that $y = f(x)$ is a tabulated function with equal increments in the argument $(x_0, x_1, x_2, \ldots, x_n)$, this routine will approximate the definite integral

$$\frac{1}{(x_n - x_0)} \int_{x_0}^{x_n} y dx$$

using either Simpson's rule or a modification of Simpson's rule.

The function values may be stored in ascending order of the argument in any block of consecutive storage cells.

At the time of entry into the subroutine the programmer must supply the value of n (that is, the number of intervals, one less than the number of points) and the address of the cell containing the first function value $y_0 = f(x_0)$.

The routine gives as a result an approximation to

$$\frac{1}{(x_n - x_0)} \int_{x_0}^{x_n} y dx$$

and the programmer must then multiply this result by $(x_n - x_0)$ to obtain the approximation to the integral itself.

Notation

$$I = \int_{x_0}^{x_n} f(x) \; dx, \quad J = \frac{1}{(x_n - x_0)} \; I.$$

I^* and J^* are approximations to I and J, respectively.

$\varepsilon = I^* - I.$

Range of y_i, J^*, and n

The only restriction on the y_i's is that they must be single precision fixed point numbers. The number of intervals n must be greater than one but can be arbitrarily large. The result J^* will be given scaled by the same amount that the y_i's were scaled.

In order to obtain the maximum significance for J^* the y_i's should be scaled as far to the left as possible.

Programming Instructions

Before entering the routine the function values y_i must be stored in ascending order of the arguments in consecutive storage cells.

1. Place $n \cdot 2^0$ in A.

2. Place the address of $y_0 = f(x_0)$ in Q_v

3. Enter the routine with the instruction RJ 00K01 00K02 (assuming 00K00 is the region that was assigned to the routine during assembly)

At the time of exit from the routine the result J* is left in A, scaled by the same amount that the y_i's were scaled. The numbers y_i have been left in their original state.

Mathematical Analysis

Let the equal increment of x be denoted by

$$h = x_i - x_{i-1} = \frac{x_n - x_0}{n} > 0, \; i = 1, 2, \ldots, n.$$

Suppose I is to be approximated by a quadrature formula of the form

$$I^* = h \sum_{i=0}^{n} c_i \, f(x_i) = \frac{x_n - x_0}{n} \sum_{i=0}^{n} c_i \, y_i$$

where the c_i are the appropriate coefficients, e.g., for the trapezoidal rule $c_0 = c_n = 1/2$ and $c_i = 1$ otherwise. Let

$$J^* = 1/n \sum_{i=0}^{n} c_i \, y_i .$$

Then

$$I^* = (x_n - x_0) \, J^*.$$

Notice that J* does not involve x, and therefore J* can be computed without regard to the scaling of x. For this reason J* rather than I* is obtained by the subroutine.

If n is even Simpson's rule is used throughout the interval (x_0, x_n). If n is odd Simpson's rule is used over the interval (x_0, x_{n-3}) and Newton's three-eighths rule is used over the interval (x_{n-3}, x_n),

therefore

$$J^* = 1/3n \, (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots + 4y_{n-1} + y_n) \text{ for n even, and}$$

$$J^* = 1/3n \, (y_0 + 4y_1 + 2y_2 + \ldots + 4y_{n-4} + y_{n-3}) +$$

$$+ 3/8n \, (y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n) \text{ for n odd.}$$

Error Analysis

Let α be some value in the closed interval (x_0, x_n) and let β and γ be values in the closed intervals (x_0, x_{n-3}) and (x_{n-3}, x_n) respectively.

If $\frac{d^4y}{dx^4}$ is continuous throughout the interval (x_0, x_n) and if n is even

$$\varepsilon = \frac{y^4(\alpha)\ (x_n - x_0)^5}{180n^4}, \text{ where } y^4(\alpha) = \left.\frac{d^4y}{dx^4}\right]_{x=\alpha}$$

If $\frac{d^4y}{dx^4}$ is continuous throughout the invertal (x_0, x_{n-3}) and exists throughout the interval (x_{n-3}, x_n) and if n is odd

$$\varepsilon = 1/n^5 \left[\frac{n-3}{180}\ y^4(\beta) + \frac{3}{80}\ y^4(\gamma)\right]\left[x_n - x_0\right]^5.$$

For the derivation of these quadrature formulas and their error terms, see Milne's Numerical Calculus, pp. 120 thru 124.

Machine Time

The time required for this subroutine is (2.25 + .62n) ms, $n \neq 3$. When n = 3 the time required is 2.73 ms.

Machine Checking

Two preliminary test cases were run:

1. $n = 99$, $y_i = -(2^{35}-1)$ for all i. The result obtained was -2^{35} (it should have been $-(2^{35}-1)$.
2. $n = 98$, $y_i = (2^{35}-1)$ for all i. The correct result, $(2^{35}-1)$, was obtained.

In addition, the following computations were performed:

1. SIN-0 was used to produce a table of sines and cosines for the arguments

   $y = \pi X/2 = (\pi/2)\cdot n\cdot 2^{-4}$, n = 0, 1, 2, . . . , 99.

2. Let $S = \int_0^b \cos y dy = \sin b$ and $C = \int_0^b \sin y dy = 1-\cos b$. DIE-0 was used to

   compute S* and C* for $b = (\pi/2)\cdot n\cdot 2^4$, n = 2, 3, 4, . . . 9 ,
   n = 10,15,20,25
   n = 30,40,50, . . . 90, and
   n = 99

3. For each b (or n) $\varepsilon_s = S^* - \sin b$ and $\varepsilon_c = C^* - 1 + \cos b$ were computed. The following tables resulted:

n	$\varepsilon_s \cdot 10^6$	$\varepsilon_c \cdot 10^6$
2	.101	.010
3	.338	.050
4	.197	.039
5	.422	.125
6	.287	.087
7	.490	.215
8	.365	.151
9	.540	.316
10	.429	.229
15	.560	.650
20	.477	.713
25	.200	1.056
30	.101	1.022
40	-.365	.881
50	-.506	.415
60	-.198	.039
70	.286	.087
80	.516	.516
90	.286	.945
99	-.338	.981

It was to be expected that, in general, the errors would be greater for n odd. The overall behavior of ε_s and ε_c is easily seen to be consistent with the fact that ε_s and ε_c represent the errors in integrating the cosine and the sine respectively. For example, ε_s is small for n = 30 because it is the error obtained in

$$\int_0^{\frac{15}{16}\pi} \cos y\,dy = \int_0^{\frac{1}{16}\pi} + \int_{\frac{1}{16}\pi}^{\frac{\pi}{2}} + \int_{\frac{\pi}{2}}^{\frac{15}{16}\pi}$$

$$= \int_0^{\frac{1}{16}\pi} + \int_{\frac{1}{16}\pi}^{\frac{\pi}{2}} - \int_{\frac{1}{16}\pi}^{\frac{\pi}{2}} = \int_0^{\frac{1}{16}\pi}$$

since $\int_{\frac{\pi}{2}}^{\frac{15\pi}{16}} = -\int_{\frac{1}{16}\pi}^{\frac{\pi}{2}}$

In fact, ε_s for n = 30 (b = (15/16)π) is exactly equal to ε_s for n = 2 (b = (1/16)π).

Evidently, all the errors ε_s and ε_c are less than their corresponding maximum estimates as computed by the formulas above under <u>Error estimate</u>. In particular, for n = 10 and 15 the following maximum error estimates were hand computed:

n	Max ε_s 10^6	Max ε_c 10^6
10	.508	.421
15	.740	.903

```
D                 00C00   00013          D
D                 00T00   00023           I       O
D                 01C00   01078            R       R
D                 00P00   01024             E       Y
D                 00S00   01032              C
D                 00N00   01059               T
D                 D1C00   49864          DRUM
D                 D0P00   49810            STORAGE
D                 D0S00   49818              DIREC-
D                 D0N00   49845                TORY
D0P00     MS      00000   00P00          NO ALRM EXIT
D0P01     MJ      00000   00000          NORMAL EXIT
D0P02     TP      A0000   00T01          ENTRY STORE
D0P03     TP      Q0000   00T02          N AND ADRS
D0P04     DV      01C00   00T03          OF Y0
D0P05     ZJ      00N00   00P06          IS N EVEN
D0P06     TV      00S22   00S19          YES EXIT TO
D0P07     TP      00T01   00T03          S20    N-BAR
D0S00     MP      01C01   00T01          EQUALS N
D0S01     TP      A0000   00T04          STORE 3N
D0S02     TP      00C03   00T05            STORE 1
D0S03     TP      01C02   00T06                  4
D0S04     TP      00C03   00T07             AND  1
D0S05     TP      00T03   A0000          INDEX IS ONE
D0S06     DV      01C00   00T08          HALF N-BAR
D0S07     RS      00T08   00C03          MINUS ONE
D0S08     TV      00T02   00S11          PRESTORE V
D0S09     SP      00C00   00000          CLEAR A
D0S10     RP      30003   00S12          FORM
D0S11     MA      00T05   00000                SUM
D0S12     TP      A0000   00T09          STORE
D0S13     LA      A0000   00036            PARTIAL
D0S14     TP      A0000   00T00                 SUMS
D0S15     IJ      00T08   00S23          IS INDEX NEG
D0S16     SP      00T09   00036          YES    STORE
D0S17     SA      00T00   00036          INTEGRAL TO
D0S18     DV      00T04   00T04          X N-BAR
D0S19     MJ      00000   00000          STORE INTE-
D0S20     TP      00T04   A0000          GRAL IN A
D0S21     MJ      00000   00P01          GO TO EXIT
D0S22     00      00000   00S20          DUMMY-SEE P6
D0S23     RA      00S11   01C00          MDFY V BY 2
D0S24     SP      00T09   00036          RESTORE PAR-
```

```
DOS25    SA  00T00  00036          TIAL SUMS
DOS26    MJ  00000  00S10          GO TO SIO
DON00    TV  00N18  00S19          N IS ODD EX-
DON01    TP  00T01  A0000          IT TO N5
DON02    ST  01C01  00T03          N-BAR IS N-3
DON03    ZJ  00S00  00N04          IS N 3
DON04    TP  A0000  00T04          YES CLEAR T4
DON05    LA  00T01  00003             STORE 8N
DON06    TP  01C01  00T06             STORE 3
DON07    TP  01C03  00T07                   9
DON08    TP  01C03  00T08                   9
DON09    TP  01C01  00T09               AND 3
DON10    RA  00T02  00T03          STORE ADRS
DON11    TV  00T02  00N14          OF YN-BAR
DON12    SP  00C00  00000             CLEAR A
DON13    RP  30004  00N15          FORM
DON14    MA  00T06  00000                 SUM
DON15    DV  00T01  00T06          STORE INTE-
DON16    RA  00T04  00T06          GRAL IN A
DON17    MJ  00000  00P01          GO TO EXIT
DON18    00  00000  00N05          DUMMY-SEE NO
D1C00    00  00000  00002          CONSTANTS 2
D1C01    00  00000  00003                    3
D1C02    00  00000  00004                    4
D1C03    00  00000  00009              AND   9
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Floating Point Exponential Routine

Specifications

Identification Tag:	EXP-3
Type:	Subroutine
Assembly Routine Spec:	SUB 50230 04715
Storage:	32 instructions, addresses 00E00 thru 00E31 01E00 thru 01E31 15 constants in program, addresses 02E00 thru 02E14 03E00 thru 03E14 47 words total program storage, addresses 00E00 thru 00E31 01E00 thru 01E31 02E00 thru 02E14 03E00 thru 03E14 2 words temporary storage pool used, addresses 00027b thru 00030b The constant pool is used by this routine
Program Entrance:	Address 00E02 (01E02)
Program Exit:	Address 00E01 (01E01)
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Addresses 63766b thru 64045b
Machine Time:	4.3 ms average, 5.46 ms maximum
Mode of Operation:	Floating point

Coded by:	M. Perry	July 27, 1955
Code Checked by:	R. Bigelow	July 27, 1955
Machine Checked by:	M. Perry	August 8, 1955
Approved by:	W. Bauer	August 10, 1955

Description

When supplied with an argument X in SNAP form this routine will compute the exponential (e^X) using a Rand Polynomial Approximation producing the answer in SNAP form.

Programming Instructions

This routine can be inserted into a program by CMP-0 by the use of a "SUB" card in the input deck.

1. Place the double length extension of X in the accumulator.

   X must be in SNAP form.

2. Return Jump to the subroutine. Assuming that the subroutine was assigned to region 00K00 for assembly, use the instruction RJ 00K01 00K02.

3. At the time of exit from the subroutine, the double length extension of e^X in SNAP form will be in the accumulator.

Error Analysis

For $X \leq 2$, the error in e^X is less than 2^{-26}, for $X > 2$, the error in e^X is less than $2^{-(26-E)}$ where E is the binary power of X.

Mathematical Analysis

1. $e^X = \left(2^{-129}\right) \cdot 2^{X(\log_2 e + 129)}$

2. Divide $(X \cdot \log_2 e + 129)$ into an integral part R and a fractional part S.

   By the necessary limitations on X, $R \geqslant 0$, $0 \geqslant S > 1$

3. $e^X = \left(2^{R-129}\right)\left(2^S\right)$

   $= \left(2^{R-128}\right)\left(\frac{2^S}{2}\right)$

4. 2^S is evaluated using the Rand Polynomial Approximation number 20. $(2^S = 10^{S \log_{10}}$

5. Since $0 \geqslant S > 1$, $1 \geqslant 2^S > 2$

   $1/2 \geqslant 2^S/2 > 1$

6. R-128 is the characteristic of the answer in floating notation, and $\frac{2^S}{2}$ is the mantissa.

Range of Variable

An alarm will result if X is greater than $.693 \times 2^7$. If X is less than $-.693 \times 2^7$, the answer will be zero, but no alarm will occur. Essentially, the exponential of any number can be found, provided that exponential can be expressed in SNAP form.

Alarm Conditions

An alarm print will occur if the variable falls outside the permissible range stated above. The flexowriter will print "alarm" and the address of the cell in the main program containing the RJ instruction which was used to enter EXP-3.

Pushing the start button after an alarm halt will transfer control to the exit of EXP-3.

```
D                    00E00  50230          EXP ROUTN 32
D                    01E00  01024          REL 2000  32
D                    02E00  50262          EXP CONST 15
D                    03E00  01056          REL 2000  15
    E00      37      75701  75702   B      ALARM EXIT
    E01      MJ      00000  00000          NORMAL EXIT
    E02      TP      A0000  Q0000
    E03      TM      A0000  A0000
    E04      TJ      03E00  01E07
    E05      SP      00013  00000
    E06      QJ      01E01  01E00          ZERO   ALARM
    E07      QT      03E01  00024          E-128
    E08      QT      03E02  00023          M
    E09      QJ      01E10  01E12          NEG
    E10      AT      03E01  00023
    E11      CC      00024  03E01
    E12      RS      00024  03E03          E-36
    E13      SJ      01E17  01E14          ANS 1
    E14      LA      A0000  00008
    E15      TV      B0000  01E16
    E16      LA      00023  00000          M FIXED
    E17      MP      B0000  03E04          LOG E BASE 2
    E18      LA      A0000  00010
    E19      AT      03E05  00024          E FINAL -128
    E20      SS      00024  00035
    E21      MP      A0000  03E06
    E22      TP      B0000  00023          ARG OF POLY
    E23      SP      03E07  00035          A-7
    E24      RP      20007  01E26          POLY
    E25      PM      03E08  00023            EVALUATION
    E26      MP      B0000  Q0000
    E27      SP      B0000  00028
    E28      TP      B0000  Q0000          M FINAL
    E30      RA      A0000  Q0000          PACKED
    E31      MJ      00000  01E01          OUT
   2E00      20      75426  00000   B      RANGE
   2E01      77      70000  00000   B      MASK
   2E02      00      07777  77777   B      MASK
   2E03      13      40000  00000   B      92
   2E04      27      05243  54511   B      LOG E BASE 2
   2E05              00000  00201   B      129
   2E06      11      50404  65025   B      LG 2 BASE 10
   2E07      00      00750  76227   B      A-7          34
   2E08      00      01235  60322   B      A-6          33
   2E09      00      10726  65710   B      A-5          33
   2E10      00      45263  67026   B      A-4          33
   2E11      02      02177  57751   B      A-3          33
   2E12      05      23242  73144   B      A-2          33
   2E13      11      15354  37452   B      A-1          33
   2E14      10      00000  00000   B      A-0          33
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Ferranti Input Routine

Specifications

Identification Tag:	FRI-0
Type:	Service routine (with a subroutine entrance provided)
Storage:	91 instructions, addresses 40001b and 75160b thru 75174b, 75177b thru 75256b, 75260b thru 75314b
	16 constants in program, addresses 75175b, 75176b, 75257b, 75333b, and 75315b thru 75330b
	2 words temporary storage used in program, addresses 75331b thru 75332b
	109 words total program storage, addresses 40001b and 75160b thru 75333b
	The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40001b
Program Entrance:	Address 75161b
Program Exit:	Address 75173b
Mode of Operation:	Fixed point

Coded by:	R. Beach	May 18, 1955
Code Checked by:	R. Summers	May 19, 1955
Machine Checked by:	R. Beach	August 4, 1955
Approved by:	W. Bauer	August 29, 1955

Description

I. General

This routine is designed to read, by means of the Ferranti reader, seven-level bioctal tape prepared as described below. The routine reads in paper tape at the full speed of the Ferranti with only short hesitation when a check or insert address is encountered.

If desired, the tape may contain a check sum to be tested for agreement with the computed sum of the data read-in. The routine will read data into any ES or MD cell although the reading of information into certain drum cells (as described in detail below) will result in abnormal operation.

The routine stores the contents of ES on MD at addresses 76000b through 77777b and then transfers itself to ES. It sums itself (in ES) and checks the sum against the correct sum (stored on MD).

The Ferranti reader is started in the free running mode and the routine proceeds to read tape and process the information contained on the tape in the same manner as does the ERA photoelectric reader (for exceptions, see II. 3 and 4).

Each word to be transferred to memory is summed as it is read in from tape. Words which are to be read into ES are first stored in the MD image of ES (76000b thru 77777b).

During operation all words are read into ES from the tape and a block transfer to MD is made when (1) ES has been filled with data (that is, when 924 words have been read in); (2) an insert address appears on the tape; or (3) the "end of tape" seven-level combination has been read in (see II. 4).

The reader is stopped before making the transfer and is started again after the transfer has been completed in the first two cases; in the last case, the reader is stopped, ES is restored from the MD image and control is transferred to the exit.

The reader is also halted when a check address appears on the tape. If no check sum test (see II. 3) is to be made after a successful check address test the reader is started immediately; if the check sum test is specified the reader is started after the test is made and the sum determined to be correct.

The routine does not prevent read in to addresses 40001b, 75160b thru 75333b, or 76000b thru 77777b. A tape specifying loading into these cells will ordinarily not be read in correctly.

II. Requirements for Tape Preparation

1. The first word on a tape must be an insert address.

2. Check addresses should be used, although FRI-0 will operate without them. A check address immediately following an insert address must be the same as the insert address.

3. For a check sum test the following four words must appear on the tape at the point where the sum is to be tested:

1) Insert address 75202b

2) High order 36 bits of check sum

3) Low order 36 bits of check sum

4) Check address 75204b

Operating Instructions (to be followed when the routine is used as a service routine)

1. Set PAK to 40001b and start.

2. Computation will halt with the MS instruction 56 00000 40001 at the completion of the read in.

Programming Instructions (to be followed when the routine is used as a subroutine)

1. Enter the routine with the RJ instruction 37 75173 75161.

2. Control is returned to the cell immediately following the RJ instruction as soon as an "end of block" punch is reached on the tape.

Alarm Conditions

1. No "end of tape" punch. This condition is indicated by the tape running completely out of the Ferranti reader. When such a condition occurs the operator should

   a. Master clear

   b. Set PAK to 00074b and start

   c. When computation halts (when a service entry was used) with the MS instruction 56 00000 40001 the machine will be returned to its original state and the data read from the tape will be properly stored.

      If a program entry was used control will be transferred to the proper cell in the main program.

2. FRI-0 not transferred to ES correctly. If ALR-1 prints "FRI·0 75165" and (A) and (Q), the sum of the program transferred to ES has failed to check. Starting at this point transfers FRI-0 to ES again.

   A second failure indicates that FRI-0 is not on the drum correctly and should be restored.

3. Check address failure. If ALR-1 prints "ALAR|C" and (A) and (Q), a check address has failed. In the alarm print (A_R) is the address of the next cell to be loaded and (Q) is the check address that was read in from paper tape.

   Starting at this time will cause the machine to ignor the failure and operation will continue normally.

4. Check sum failure. If ALR-1 prints "ALAR|M" and (A) and (Q), the check sum on the tape has failed to agree with the computed sum.

   Starting at this point will cause the routine to ignor the failure and to begin to read in the tape again.

If at any time (ES) need be restored from its image, starting at 40040b will transfer the image to ES and transfer control to the FRI-0 exit.

5. And "end of tape" (or "end of block") punch must be present on the tape to halt read in. This consists of seventh level punches in two consecutive frames on the tape at the point where the read in is to be stopped. This seventh level combination acts as a signal to FRI-0 to restore (ES) and stop the Ferranti reader. It is compatible with the ERA photoelectric reader in that it is an illegal combination which halts the ERA reader.

```
75161   00  00000  00000     FRI 0
75325   00  00000  00000       DIRECTORY
40000   00  00000  00000       CARDS
75160   00  00000  00000       CARDS
  117   00  00000  00000
  133   00  00000  00000
  136   00  00000  00000
  144   00  00000  00000
40001   45  00000  75160     SET ENTRANCE
75160   16  75302  75173     SET EXIT
75161   11  00000  76000     STORE ES
75162   11  75173  00000       AND LOAD
75163   75  31777  75165       FRI 0 INTO
75164   11  00001  76001       ES
75165   16  75306  40040     SET RESTORE
75166   75  30135  00001
75167   11  75177  00001
75170   75  31777  75172     RESTORE
75171   11  76001  00001       ES FROM
75172   11  76000  00000       IMAGE
75173   45  00000  00000     EXIT
75174   56  00000  40001
75175   00  00000  00042
75176   73  53553  76415
75177   31  00000  00000     COMPUTE
75200   75  20135  00004
75201   32  00001  00000       AND TEST
75202   34  75176  00044
75203   11  75257  75756
75204   43  75175  00011
75205   37  75701  75703
75206   45  00000  75166
75207   23  00136  00136     CLEAR SUM
75210   23  00137  00137       STORAGE
75211   11  00120  00061     SET UP
75212   17  00006  00014     START READER
75213   76  00000  10000     READ TO Q
75214   31  00140  00006     ASSEMBLE
75215   52  00121  00140       DATA
75216   31  00133  00001     OBTAIN
75217   52  00122  10000       SEVENTH
75220   51  00122  00133       LEVEL CODE
75221   43  00123  00061     IS THIS ED
75222   43  00124  00074     IS THIS IA
75223   43  00125  00031     IS THIS CA
75224   51  00126  20000     IS THIS END
75225   43  00126  00074       OF TAPE
75226   45  00000  00015     RET TO READ
```

75227	17	00005	00031	STOP READER
75230	11	00061	25016	
75231	36	00120	20000	ADDRESS
75232	21	20000	00134	TEST
75233	43	00127	00044	
75234	43	00140	00042	
75235	16	00032	75756	TAG TO ALARM
75236	11	00140	10000	TAPE CA TO Q
75237	37	75701	75703	
75240	43	00135	00046	
75241	45	00000	00014	SPECIFIED
75242	23	20000	20000	MOD CA
75243	45	00000	00036	
75244	31	00136	00044	TEST CHECK
75245	32	00137	00000	SUM
75246	34	00144	00000	
75247	34	00145	00000	
75250	34	00145	00044	
75251	43	00144	00011	
75252	54	20000	00044	SET UP ALARM
75253	32	00145	00000	PRINT
75254	16	00071	75756	TAG TO ALARM
75255	37	75701	75703	
75256	45	00000	00011	
75257	ED → 26	12144	23704	replaced if no cksum alarm by 11 140 144
75260	21	00061	00130	MOD ADRS
75261	31	00136	00044	ADD
75262	32	00137	00000	ASSEMBLED
75263	32	00140	00000	WORD TO
75264	11	20000	00137	COMPUTED
75265	54	20000	00044	SUM
75266	11	20000	00136	
75267	11	00061	25007	
75270	43	00117	00074	WITH DATA
75271	45	00000	00015	
75272	17	00005	00074	STOP READER
75273	31	00061	00000	SET UP NMBR
75274	34	00120	00017	OF WORDS
75275	35	00131	00103	TO TRNSFR
75276	11	00134	20000	DOES DATA GO
75277	42	00127	00111	IN ES
75300	16	20000	00104	SET UP TNSFR
75301	00	00000	01634	TRANSFER
75302	11	00144	75174	DATA TOMD
75303	11	00133	20000	IS THIS A
75304	43	00123	00113	DATA WORD ?
75305	43	00124	00115	IS THIS ~~CA~~ IA ?

75306	45	00000	75170	RET TO MD
75307	21	20000	00132	MOD STORAGE
75310	45	00000	00102	ADDRESS
75311	21	00134	75301	MOD INSERT
75312	45	00000	00013	ADDRESS
75313	16	00140	00134	INSERT
75314	45	00000	00013	ADDRESS
75315	11	00140	02000	00K00
75316	11	00140	00144	1
75317	00	00000	00077	2
75320	00	00000	17700	3
75321	00	00000	10100	4
75322	00	00000	11100	5
75323	00	00000	10500	6
75324	00	00000	00300	7
75325	00	00000	02000	8
75326	00	00000	00001	9
75327	75	30000	00105	10
75330	00	00000	76000	11
75331	00	00000	00000	7th level 00S00 CODE
75332	00	00000	00000	1 IA
75333	00	00000	75204	

75334 (136)
75335 (137)
336 (140)
337 (141)
340 (142)
341 (143)
342 (144)

CK SUM
STORAGES
word assembly

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Simplified Ferranti Input Routine for Bootstrap Procedure

Specifications

Identification Tag:	FRI-1
Type:	Service routine, but not available as part of service routine library
Storage:	17 instructions, addresses 00000b thru 00020b
	6 constants in program, addresses 00021b thru 00025b and 00027b
	1 word of temporary storage in program, address 00026b. Two words of temporary storage not in program, addresses 00030b and 00040b.
	The temporary and constant storage pools are not used by this routine.
Entrance:	00000b (automatic entrance from keyed in binary card read in routine).

Coded by:	R. Beach	April 1, 1955
Code Checked by:	R. Summers	April 2, 1955
Machine Checked by:	R. Beach	April 12, 1955
Approved by:	W. Bauer	August 23, 1955

Description

This routine is a simplification of FRI-0 which can be contained on one binary card. It is used, as part of the bootstrap procedure, to read FRI-0 into ES. The routine reads tape into ES recognizing only insert addresses, data words, and the "end of tape" seven-level combination. When it finds the "end of tape" combination it transfers control to the transfer instructions at 00050b thru 00053b. The transfer instructions stop the reader, clear cells 40001b thru 40040b, 70000b thru 75777b, and 60000b thru 67777b, transfer FRI-0 to its proper location on MD and then transfer control to FRI-0.

Operating Instructions

See the "Bootstrap Procedure" in the "Utility Routine Library Handling Package" description.

	45	00000	00001	ENTRANCE
1	17	00006	00001	START READER
2	11	00030	00040	ENTER DATA
3	21	00002	00021	MOD ADRS
4	76	00000	10000	READ TO Q
5	31	00030	00006	ASSEMBLE
6	52	00022	00030	DATA
7	31	00026	00001	OBTAIN
10	52	00023	10000	SEVENTH
11	51	00023	00026	LEVEL CODE
12	43	00024	00002	IS THIS ED
13	43	00025	00017	IS THIS IA
14	51	00027	20000	IS THIS END
15	43	00027	00050	OF BLOCK
16	45	00000	00004	RET TO READ
17	16	00030	00002	INSERT
20	45	00000	00004	ADDRESS
21	00	00000	00001	ADRS MODFIER
22	00	00000	00077	DATA MASK
23	00	00000	17700	7 LVL MASK
24	00	00000	10100	ED CODE
25	00	00000	11100	IA CODE
26	00	00000	00000	7TH LVL CODE
27	00	00000	00300	END CODE

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Floating Point Natural Logarithm Routine

Specifications

Identification Tag:	LOG-2
Type:	Subroutine
Assembly Routine Spec:	SUB 50190 03810
Storage:	28 instructions, addresses 0LN00 thru 0LN27 1LN00 thru 1LN27 10 constants in program, addresses 2LN00 thru 2LN09 3LN00 thru 3LN09 38 words total program storage, addresses 0LN00 thru 0LN27 1LN00 thru 1LN27 2LN00 thru 2LN09 3LN00 thru 3LN09 3 words temporary storage pool used, addresses 00027b thru 00031b The constant pool is used by this routine
Program Entrance:	Address 0LN02 (1LN02)
Program Exit:	Address 0LN01 (1LN01)
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Address 63716b thru 63763b
Machine Time:	3.37 ms average, 4.12 ms maximum
Mode of Operation:	Floating point

Coded by:	M. Perry	July 27, 1955
Code Checked by:	R. Bigelow	July 27, 1955
Machine Checked by:	M. Perry	August 8, 1955
Approved by:	W. Bauer	August 10, 1955

Description

When supplied with an argument X in SNAP form this routine will compute the natural logarithm of X using a Rand Polynomial Approximation, producing an answer in SNAP form.

Programming Instructions

This routine can be inserted into a program by CMP-0 by the use of a "SUB" card in the input deck.

1. Place the double length extension of X in the accumulator. X must be in SNAP form.
2. Return Jump to the subroutine. Assuming that the routine was assigned to region 00K00 for assembly, use the instruction RJ 00K01 00K02.
3. At the time of exit from this routine, the double length extension of ln X in SNAP form will be in the accumulator.

Error Analysis

The error in the result of this routine is less than 2^{-26}.

Mathematical Analysis

1. Let $X = M \cdot 2^e$

   Then $\ln X = (\ln 2)(\log_2 X)$
   $= (\ln 2) \log_2 (M \cdot 2^e)$
   $= (\ln 2)\ e-1 + \log_2(2M)$

2. $\text{Log}_2(2M)$ is found by evaluating the Rand Polynomial Approximation Number 42, with argument 2M.
3. Since $1/2 \leqslant M < 1, \quad 1 \leqslant 2M < 2$ and $\log_2(2M)$ is between 0 and 1.

Range of Variable

The logarithm of any number x, $(X>0)$ may be evaluated by this routine provided X can be expressed in SNAP form.

Alarm Conditions

An alarm print will occur if the variable falls outside the permissible range stated above. The flexowriter will print "alarm" and the address of the cell in the main program containing the RJ instruction which was used to enter LOG-2.

Pushing the start button after an alarm halt will transfer control to the exit of LOG-2.

```
D                0LN00  50190       LN ROUTIN 28
D                1LN00  01024       TO BE ALTERD
D                2LN00  50218       LN CONST  10
D                3LN00  01052       TO BE ALTERD
  LN00    37     75701  75702  B    ALARM EXIT
  LN01    MJ     00000  00000       NORMAL EXIT
  LN02    SJ     1LN00  1LN03       NORMAL ENTRY
  LN03    TP     A0000  Q0000
  LN04    QT     3LN00  00023       M
  LN05    RS     Q0000  00023       E-128
  LN06    ST     3LN01  00024       E-ONE HALF
  LN07    SP     00023  00001
  LN08    AT     3LN02  Q0000
  LN09    SS     3LN03  00034
  LN10    DV     Q0000  00023       ARG OF POLY
  LN11    MP     Q0000  Q0000
  LN12    TP     B0000  00025       ARG SQUARED
  LN13    SP     3LN04  00035       C-7
  LN14    RP     20003  1LN16
  LN15    PM     3LN05  00025       POLY
  LN16    PM     00024  00023         EVALUATION
  LN17    MP     B0000  3LN08       LN 2
  LN18    TP     B0000  A0000
  LN19    ZJ     1LN20  1LN01
  LN20    SF     A0000  00025       SCALE M
  LN21    LA     A0000  00027
  LN22    TP     B0000  00023       M FINAL
  LN23    SP     00025  00027
  LN24    AT     3LN09  Q0000       E-128 FINAL
  LN25    CC     00023  Q0000       PACK
  LN26    TP     A0000  A0000       EXTEND
  LN27    MJ     00000  1LN01       OUT
2LN00     00     07777  77777  B    MASK
2LN01     20     04000  00000  B
2LN02     00     13240  47463  B    SQ RT 2
2LN03     00     26501  17147  B    2 X RT 2
2LN04     15     71272  26456  B    C-7           35
2LN05     02     23066  04015  B    C-5           32
2LN06     00     75434  22311  B    C-3           30
2LN07     00     56125  07310  B    C-1           28
2LN08     13     05620  57740  B    LN 2          34
2LN09     10     00000  00000  B    64
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Bioctal Memory Dump

Specifications

Identification Tag:	MDP-1
Type:	Service routine (with a program entry available)
Storage:	157 instructions, addresses 40005b, 74703b thru 74754b 74760b thru 75141b
	17 constants in program, addresses 74755b thru 74757b 75142b thru 75157b
	174 words total program storage, addresses 74703b thru 75157b
	10 words temporary storage pool used, addresses 00027b thru 00040b
	The constant pool is used by this routine
Service Entrance:	Address 40005b
Program Entrance:	Address 74705b
Program Exit:	Address 74704b
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Addresses 74703b thru 75157b
Machine Time:	21 seconds per 100 words maximum
Mode of Operation:	Fixed point

Coded by:	W. Dixon	August 15, 1955
Code Checked by:	C. Koos	August 20, 1955
Machine Checked by:	W. Dixon	September 15, 1955
Approved by:	W. Bauer	September 19, 1955

Description

This routine will dump onto bioctal tape the contents of any specified number of consecutive storage cells. It feeds leader and trailer, inserts insert and check addresses and check sums, and may place a double seven-level punch (required to stop FRI-0) at the end of each dump.

The tape produced can be read back into the 1103 with the Ferranti reader (using FRI-0) or with the ERA photoelectric reader. However, when using the photoelectric reader one should note that the check sum will be read into cells 75702b and 75703b and that the double seven-level combination will stop the reader by causing a fault.

This routine stores (ES) in cells 76000b thru 77777b while operating and at its conclusion restores (ES), (A), (Q), and (F1) to their original state.

Operating Instructions (to be followed when the routine is used as a service routine)

1. Set PAK to 40005b and start.

2. Computation will halt with the MS instruction 56 00000 74730 and Q will be cleared.

3. Manually enter the parameter word in Q. Place the address of the first cell to be dumped in Q_u and the address of the last cell to be dumped in Q_v.

   The range of the dump may extend from ES addresses to drum addresses. For example, if the word 00 00200 40050b is placed in Q, the routine will dump cells 00200b through 01777b and 40000b through 40050b.

   If an FRI-0 stop code is to be punched on the tape following the dump place a 01 in the operation portion of Q. If no stop code is to be punched place an 00 in the operation portion of Q.

4. Start.

5. Computation will halt at the conclusion of the dump with the MS instruction 56 00000 74706.

6. For another dump one need only push the start button and return to step 2 above.

Programming Instructions (to be followed when the routine is used as a subroutine)

1. Enter the routine with the RJ instruction 37 74704 74705 B.

2. Enter the parameter word. If the RJ instruction is in cell n the parameter word (as described in operating instruction 3 above) must be placed in cell n + 1.

3. At the conclusion of the routine control will be transferred to the instruction in cell n + 2.

Alarm Conditions

An alarm print will occur if an unacceptable parameter word has been supplied to the routine. Any one of the following three conditions will produce an alarm print:

$$Q_u > Q_v$$

$$02000b \leq Q_u < 40000b$$

$$02000b \leq Q_v < 40000b$$

If the service entrance was used the Flexowriter will print "MDP-1 74705". The operator may push the start button at this time and return to operating instruction 2 for another dump.

If the program entrance was used the Flexowriter will print "MDP-1" and the address of the cell in the main program containing the unacceptable parameter word. Pushing the start button at this time will transfer control to the exit of MDP-1.

Manual Restore

If the operator wishes to stop a dump before its normal completion he may

1. Force stop while the punch is operating.

2. Set PAK to 40040b and start.

3. Computation will halt with an MS instruction in cell 74704b (the MDP-1 exit).

If the service entrance has been used the operator may push the start button and return to operating instruction 2 for another dump.

If the program entrance has been used pushing the start button will cause a normal exit from the routine.

```
75701   00 00000 00000   ALR 1          D
   15   00 00000 00000   CONST POOL
   27   00 00000 00000   TEMP POOL      C
40000   00 00000 00000   40000 ENTR     A
74703   00 00000 00000
74767   00 00000 00000
   41   00 00000 00000   ES PROG        M
75112   00 00000 00000
  164   00 00000 00000   ES PROG        P
75142   00 00000 00000
  214   00 00000 00000
   15   00 00000 00000   ZERO
   16   00 00000 00077   SIX BIT EXTR
   17   00 00001 00000   ADVANCE U
   20   00 00000 00001   ADVANCE V
   21   00 00001 00001   ADV U AND V
   22   00 00000 00110   DECIMAL 72
   23   52 52525 25252   ALTERNATOR
   24   77 00000 00000   OP CODE MASR
   25   00 07777 00000   N MASK
   26   00 00000 07777   NR MASK
   27   00 00000 00000   FIRST ADDRES
   30   00 00000 00000   LAST ADDRESS
   31   00 00000 00000   INSERT ADDRS
   32   00 00000 00000   CHECK ADDRES
   33   00 00000 00000   FA INDEX
   34   00 00000 00000   LA INDEX
   35   00 00000 00000   NBR OF WORDS
   36   00 00000 00000   WORD INDEX
   37   00 00000 00000   SUM HI
   40   00 00000 00000   SUM LOW
40005   45 00000 74706   TO ENTRANCE
74703   37 75701 75702   ALARM XT
74704   00 00000 00000   EXIT
74705   45 00000 74721   PROG ENTR
74706   11 74755 74704   SERV ENTR
74707   37 74720 74712   TO STORE A Q
74710   23 10000 10000   CLEAR Q
74711   56 00000 74761   ENTER PARAM
74712   11 20000 00030   STORE
74713   54 20000 00044       AR
74714   11 20000 00027          AL
74715   11 10000 00031             Q
74716   11 00000 76000               FI
74717   11 74705 00000   SET FI TO MJ
74720   45 00000 00000
74721   37 74720 74712   TO STORE A Q
74722   11 00024 10000   OP MASK
74723   53 74705 74704   SET XT TO MJ
74724   31 74704 00017   PARAM
74725   15 20000 74726     WORD
74726   11 00000 10000     TO Q
74727   45 00000 74760
```

```
74730   75  31777  74732        ES TO IMAGE
74731   11  00001  76001
74732   11  74757  40040        SET 40040
74733   75  30172  00041        PROG TO ES
74734   11  74767  00041        EXIT TO ES
74735   37  74754  74746        TO RESTORE
74736   11  74756  75756        TAG TO ALR
74737   45  00000  74703        TO ALARM XT
74740   37  74754  74746        TO RESTORE
74741   45  00000  74704        TO EXIT
74742   37  74754  74744        TO MS RSTR
74743   45  00000  74704        TO EXIT
74744   11  00024  10000        MASK TO Q
74745   53  74755  74704        MS TO EXIT
74746   75  31777  74750        R   R
74747   11  76001  00001         E   O
74750   31  00027  00044          S   U
74751   32  00030  00000           T   T
74752   11  00031  10000            O   I
74753   11  76000  00000             R   N
74754   45  00000  00000              E   E
74755   56  00000  74706        CONST
74756   07  22155  65204        TAG
74757   45  00000  74742        MAN RESTR
74760   21  74704  00020        INCRIM EXIT
74761   16  74766  75101        TEST FOR PU
74762   51  00024  20000          OF STOP
74763   47  74765  74764        CODE
74764   16  75103  75101        NO PUNCH
74765   45  00000  74730
74766   00  00000  00155        CONST
74767   75  10012  00043        CLEAR
74770   11  00015  00027          TEMP STORE
74771   75  00240  00045        PUNCH
74772   63  00000  00015             LEADER
74773   63  10000  00015        PU 7 LEVEL
74774   16  10000  00030        LAST ADRESS
74775   55  10000  00025          FIRST ADRS
74776   16  10000  00027          TO STORE
74777   11  00030  20000        LA TO A
75000   42  00027  74735        AGNST FA
75001   11  00027  20000        FA TO A
75002   42  00214  00056        AGNST 40000
75003   45  00000  00072        DRUM ONLY
75004   42  00215  00060        AGNST 2000
75005   45  00000  74735        TO ALR RSTR
75006   11  00030  20000        LA TO A
75007   42  00215  00074        AGNST 2000
75010   42  00214  74735        AGNST 40000
75011   11  00027  00031        FA TO IA
```

```
75012  11  00027  00033    FA TO FA NDX
75013  11  00216  00032    01777 TO CA
75014  11  00216  00034    01777 TO LAX
75015  11  00214  00027    40000 TO FA
75016  16  00226  00142    SET JUMP 1
75017  45  00000  00075
75020  75  30004  00077    FALA TO IACA
75021  11  00027  00031      TO FAX LAX
75022  37  00072  00072
75023  21  00033  00217    INCR FAX AND
75024  21  00034  00217    LAX BY 76000
75025  16  00227  00106    SET JUMP 2
75026  11  00034  20000
75027  36  00033  00036    SET WRD INDX
75030  35  00020  00035    SET NBR WRDS
75031  11  00220  20000    824 TO A
75032  42  00035  00107    AGST NBR WDS
75033  23  00035  00035    CLER NBR WDS
75034  45  00000  00114    JUMP 2
75035  11  00221  00036    SET WRD INDX
75036  45  00000  00106    TO JUMP 2
75037  16  00230  00106    SET JUMP 2
75040  11  00031  10000    IA TO Q
75041  37  00176  00172    TO PU IA
75042  11  00025  10000    MASK TO Q
75043  31  00036  00017    SET NUMBR OF
75044  35  00017  20000      REPEATS
75045  53  20000  00124      FOR SUM
75046  53  20000  00130      AND TRNSFR
75047  31  00033  00017    FA INDEX TO
75050  15  20000  00125      TP INSTRCT
75051  21  00033  00220    INCR FA INDX
75052  75  30000  00126    DATA TO
75053  11  00000  00275      ES
75054  31  00037  00044    SUM ROUTINE
75055  32  00040  00000      STARTS
75056  75  20000  00132        HERE
75057  32  00275  00000
75060  11  20000  00040
75061  54  20000  00044        AND
75062  11  20000  00037      ENDS HERE
75063  37  00171  00164    TO PU DATA
75064  11  00035  20000    ZJ ON NUMBER
75065  47  00100  00140      OF WORDS
75066  11  00032  10000    CA TO Q
75067  37  00204  00177    TO PU CA
75070  45  00000  00143    JUMP 1
75071  11  00222  10000    SUM IA TO Q
75072  37  00176  00172    TO PU SUM IA
75073  11  00020  00036    SET WRD INDX
```

75074	11	00037	00275	SUM TO		
75075	11	00040	00276	ES		
75076	37	00171	00164	TP PU SUM		
75077	11	00223	10000	SUM CA TO Q		
75100	37	00204	00177	TO PU SUM CA		
75101	75	00022	00155	PUNCH		
75102	63	00000	00015	LEADER		
75103	75	00002	00157	PUNCH FRI 0		
75104	63	10000	00015	STOP CODE		
75105	75	00240	00161	PUNCH		
75106	63	00000	00015	LEADER		
75107	45	00000	74740	TO RESTORE		
75110	16	00231	00142	RESTORE		
75111	45	00000	00072	JUMP 1		
75112	15	00230	00165	WRD TO Q		P
75113	11	00000	10000			U
75114	21	00165	00017	ICR INST		
75115	37	00213	00205	TO PUNCH		D
75116	41	00036	00165	WRD INDX		A
75117	45	00000	00000			T
75120	11	00020	00041	SET INDX		P
75121	37	00213	00206	TO PUNCH		U
75122	11	00020	00041	SET INDX		
75123	37	00213	00206	TO PUNCH		I
75124	45	00000	00000			A
75125	21	10000	00020	INCR CA		P
75126	11	00224	00041	SET INDX		U
75127	37	00213	00206	TO PUNCH		
75130	23	00041	00041	SET INDX		C
75131	37	00213	00206	TO PUNCH		A
75132	45	00000	00000			
75133	11	00225	00041	SET INDX		R
75134	55	10000	00006		P	O
75135	63	00000	10000		U	U
75136	41	00041	00206		N	T
75137	55	10000	00006		C	I
75140	63	10000	10000		H	N
75141	45	00000	00000			E
75142	00	00000	40000	40000		
75143	00	00000	02000	2000		
75144	00	00000	01777	1777B		
75145	00	00000	76000	76000		
75146	00	00000	01470			
75147	00	00000	01467			
75150	00	00000	75202	SUM IA		
75151	00	00000	75203	SUM CA		
75152	00	00000	00002	2B		
75153	00	00000	00004	4B		
75154	00	00000	00162	DUMY ADDRESS		
75155	00	00000	00111	DUMY ADDRESS		
75156	00	00275	00114	DUMY ADDRESS		
75157	00	00000	00143	DUMY ADDRESS		

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Gill Method Subroutine

Specifications

Identification Tag:	NUI-3
Type:	Subroutine
Assembly Storage Spec:	SUB 49880 07414
Storage:	59 instructions, addresses OGM00 thru OGM40 1GM00 thru 1GM17 15 constants in program, addresses OGC00 thru OGC14 74 words total program storage, addresses OGM00 thru OGM40 1GM00 thru 1GM17 OGC00 thru OGC14 10 words temporary storage pool used, addresses 00027b (OGT00) thru 00040b (OGT09)
Drum Assignment:	Addresses 63230b thru 63354b
Program Entrances:	Addresses OGM02, OGM03, and OGM04
Program Exit:	Address OGM01
Machine Time:	(10.3 n + 1.9) ms per point average, where n equals the number of equations in the system
Mode of Operation:	Fixed point
Coded by:	J. Carlson R. Douthitt M. Elmore R. Summers
Code Checked by:	M. Elmore — June 8, 1955
Machine Checked by:	M. Elmore — July 7, 1955
Approved by:	W. Bauer — July 22, 1955

Description

The Gill Method Subroutine integrates a system of first order, differential equations using a step-by-step process. Using the values of the variables at a point and the coding for computing the derivative of each of the dependent variables at that point, the Gill Method Subroutine produces the coordinates for the next point of the solution each time it is entered.

A special entrance sets up the subroutine for a particular system of equations, thus allowing the subroutine to solve concurrently several different systems in the same program.

The independent variable is incremented within the subroutine itself.

Notation

The system of equations to be solved is

$$\frac{dy_i}{dx} = f_i\,(x, y_1, y_2, \ldots, y_n), \; (i = 1, 2, \ldots, n).$$

q_i are intermediate values of the calculation (zero initially)

Δx is the increment of the independent variable x

h is the binary scaling power of x (i.e. $x \cdot 2^h$ is in the computer)

h-1 is the binary scaling power of Δx

m_i is the binary scaling power of y_i

f is the common difference between the scaling power of y_i and the scaling power of $\frac{dy_i}{dx}$ for each i.

$m_i - f$ is the binary scaling power of $\frac{dy_i}{dx}$

$L = 73 + f - h$

Programming and Operating Instructions

Assign the Gill Method Subroutine to some arbitrary region, say OGM00.

In order to solve a given system, the following array of variables, derivatives, intermediate values, and parameters should be assigned a region, say OGN00.

OGN00		L	
OGN01	00 OGN05 OGN06		
OGN02		n-1	
OGN03		Δx	scaled 2^{h-1}
OGN04		x	scaled 2^h
OGN05		$\frac{dy_1}{dx}$	scaled $m_1 - f$
OGN06		y_1	scaled m_1
OGN07		q_1	initially zero
OGN08		$\frac{dy_2}{dx}$	scaled $m_2 - f$
OGN09		y_2	scaled m_2
OGN10		q_2	initially zero

.
.
.

In addition, the coding for computing $\frac{dy_i}{dx}$ for all i, (i = 1, 2, . . . , n) should be assigned a region, say ODE00. This coding will use the values in region OGN00 to compute all $\frac{dy_i}{dx}$ as specified by the equations in the system and should place the results in the appropriate places in region OGN00. It should then exit to the Gill Method Subroutine with an MJ 00000 OGM04 (see below).

Assuming the Gill Method Subroutine is in region OGM00, the three entrances are OGM02, OGM03, and OGM04. The exit is OGM01.

The first entrance, OGM02, is used for setting up the Gill Method Subroutine only for the particular system to be solved. It is entered by an RJ command followed by a parameter word which specifies the location of the variables, and the location of the coding for calculating the derivatives:

```
RJ OGM01 OGM02
00 OGN00 ODE00
```

The second entrance, OGM03, is the entrance for producing a point of the solution. It is entered by an RJ command: RJ OGM01 OGM03. Entering using this command results in four passes through both the Gill Method Subroutine and the coding for computing the derivatives, and leaves in region OGN00 the new values of the variables, the derivatives at those values, and x advanced by Δx, ready for the next step.

The third entrance, OGM04, is the entrance from the coding for calculating the derivatives and is used on each of the four passes necessary for computing one point. As noted above, it is entered by an MJ command in the ODE00 region:

```
MJ 00000 OGM04
```

Mathematical Analysis

Theory. "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine" by S. Gill, published in Cambridge Philosophical Society Proceedings, Vol. 47, Part I, January 1951, should be consulted for a detailed analysis of the process on which the subroutine is based.

Suppose we know the point $(X, Y_1, Y_2, \ldots, Y_n)$ on the curve defined by the system of equations

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \ldots, y_n)$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \ldots, y_n)$$

$$\vdots$$

$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, \ldots, y_n)$$

The Gill Method is a process by which we can find the next point on the curve: i.e. the value of $y_1, y_2, \ldots, y_n$ for $x = X + h$.

The process can be better understood if the case where n=1 is first considered.

We have the point (X,Y) on the curve $\frac{dy}{dx} = f(x,y)$, and we want to find y at X + h; i.e. we want $k = \delta y$ such that $\left.\frac{dy}{dx}\right]_{X + h,\ Y + k} = f(X + h, Y + k)$.

We derive k by making four approximations and averaging them in a particular way.

First approximate the curve by a straight line through (X,Y) with the slope $\left.\frac{dy}{dx}\right]_{X,Y} = f(X,Y)$, and find a first approximation to k:

$$k_o = h \cdot f(X,Y)$$

Then we travel a fraction m of the way along this line to the point $(X + mh, Y + mk_o)$ and find $f(X + mh, Y + mk_o)$.

This gives us a new straight line through $(X + mh, Y + mk_o)$ with slope $f(X + mh, Y + mk_o)$, and we find

$$k_1 = h\, f(X + mh, Y + mk_o)$$

We now use k_o and k_1 to find a third point at which f is calculated: $(X + nh, Y + [n-r]\, k_o + rk_1)$.

$$k_2 = h\, f(X + nh, Y + [n-r]\, k_o + rk_1)$$

Similarly,

$$k_3 = h \cdot f\ (X + ph,\ Y + \left[p-s-t\right]\ k_o + sk_1 + tk_2$$

The weighted average of k_o, k_1, k_2, and k_3 is the desired $k = \delta y$:

$$\delta y = y\ (X + h) - y\ (X) = c_o k_o + c_1 k_1 + c_2 k_2 + c_3 k_3$$

where $c_o + c_1 + c_2 + c_3 = 1.$

For a system of equations, the same four steps given above are made for each equation and

$$\delta y_i = c_o k_{io} + c_1 k_{i1} + c_2 k_{i2} + c_3 k_{i3} \text{ where } c_o + c_1 + c_2 + c_3 = 1.$$

The above process is, for certain values of m, n, p, s, t, c_o, c_1, c_2, and c_3, the Runge-Kutta process. The Gill process was derived, with application to machine use in mind, by minimizing the number of storage cells required. For the Gill Method the above constants are

$m = 1/2$,	$r = 1 - \sqrt{1/2}$,	$c_o = 1/6$
$n = 1/2$,	$s = \ - \sqrt{1/2}$,	$c_1 = (1/3)\ (1 - \sqrt{1/2})$
$p = 1$,	$t = 1 + \sqrt{1/2}$,	$c_2 = (1/3)\ (1 + \sqrt{1/2})$
		$c_3 = 1/6$

The Gill process further systematizes the calculation so as to increase the accuracy and simplify the coding.

<u>The Subroutine</u> As used in the Gill Method Subroutine, the process is as follows:

1st pass:

Advance x by $(1/2)h$

$$k_{io} = h \cdot f_i\ (x, y_{10}, y_{20}, \ldots, y_{no})$$

$$r_{i1} = (1/2) k_{io} - q_{io}$$

$$q_{i1} = q_{io} + 3r_{i1} - (1/2) k_{io}$$

$$y_{i1} = y_{io} + r_{i1}$$

Calculate $f_i\ (x, y_{11}, y_{21}, \ldots, y_{n1})$ in programmer's own coding.

2nd pass:

$$k_{i1} = h\ f_i\ (x, y_{11}, y_{21}, \ldots, y_{n1})$$

$$r_{i2} = (1 - \sqrt{1/2})\ (k_{i1} - q_{i1})$$

$$q_{i2} = q_{i1} + 3r_{i2} - (1 - \sqrt{1/2})\ k_{i1}$$

$$y_{i2} = y_{i1} + r_{i2}$$

Calculate $f_i\ (x, y_{12}, y_{22}, \ldots, y_{n2})$ in programmer's own coding.

3rd pass:

Advance x by $(1/2)h$

$$k_{i2} = h \cdot f_1 \ (x, y_{12}, y_{22}, \ . \ . \ . \ , \ y_{n2})$$

$$r_{i3} = (1 + \sqrt{1/2}) \ (k_{i2} - q_{i2})$$

$$q_{i3} = q_{i2} + 3r_{i3} - (1 + \sqrt{1/2}) \ k_{i2}$$

$$y_{i3} = y_{i2} + r_{i3}$$

Calculate $f_i \ (x, y_{13}, y_{23}, \ . \ . \ . \ , \ y_{n3})$ in programmer's own coding.

4th pass:

$$k_{i3} = h \cdot f_i \ (x, y_{13}, y_{23}, \ . \ . \ . \ y_{n3})$$

$$r_{i4} = (1/6)(k_{i3} - 2q_{i3})$$

$$q_{i4} = q_{i3} - 3r_{i4} - (1/2)k_{i3}$$

$$y_{i4} = y_{i3} + r_{i4}$$

Calculate $f_i \ (x, y_{14}, y_{24}, \ . \ . \ . \ , \ y_{n4})$ in programmer's own coding.

Errors The paper by S. Gill mentioned previously includes a detailed analysis of errors, both truncation error and round-off error.

The expression for the truncation error in δy_i is too complicated to give here, but its dominating term, the author states, is

$$\frac{h^5}{-120} \sum_{\substack{o \\ j,k,l,m}}^{n} \left[f_j \frac{\partial f_k}{\partial y_j} \cdot \frac{\partial f_l}{\partial y_k} \cdot \frac{\partial f_m}{\partial y_l} \cdot \frac{\partial f_i}{\partial y_m} \right]_{x=X} \quad \text{where } y_o = x, \ f_o = 1,$$

and the truncation error in δy_i will be approximately this when the second partial derivatives are all close to zero. It is probably more useful to say merely that the truncation error is of the order of h^5.

The standard deviation in $y_i - (1/3)q_i$ over one step from all rounding off errors is (where f is the quantity mentioned in the section on notation)

$$1/6 \left[7/3 \left\{ 2^{-2f} + (1/16) h^2 \sum_j \left(\frac{\partial f_i}{\partial y_j} \right)^2_X \right\} \right]^{1/2} u, \quad u = \text{the value of one unit in the last digit of } y.$$

Machine Checking

A driver routine solved two systems of equations both separately and concurrently, using the Gill Method Subroutine. The two systems solved are given below to indicate accuracy and to serve as examples.

1. Equations

$$\left.\begin{aligned}\frac{dy_1}{dx} &= y_2 \\ \frac{dy_2}{dx} &= -y_1\end{aligned}\right\}$$ equivalent to the second order equation,

$$\frac{d^2y}{dx^2} + y = 0.$$

$\Delta x = .0872664626 = \pi/36 = 5^\circ$

Initial Conditions

At $x=0$, $y_1 = 0$ and $y_2 = 1$.

Solution

$y_1 = \sin x$

Accuracy

In a spot check of the results, the greatest absolute error observed was 1.5×10^{-6}. (For $x = 3.1415925696$, $y_1 = .0000015425$. However, $\sin x = .000000084$).

2. Equations

$$\left.\begin{aligned}\frac{dy_1}{dx} &= y_2 \\ \frac{dy_2}{dx} &= y_3 \\ \frac{dy_3}{dx} &= \frac{y_3 + 4x^2}{x}\end{aligned}\right\}$$ Equivalent to the third order equation

$$x\frac{d^3y}{dx^3} - \frac{d^2y}{dx^2} = 4x^2$$

$\Delta x = .1$

Initial Conditions

At $x = .1$, $y_1 = .000025$, $y_2 = .001$, $y_3 = .03$

Solution

$$y_1 = \frac{x^4}{3} - \frac{x^3}{60} + \frac{x}{6000} - \frac{1}{120{,}000}$$

Accuracy

In a spot check of the results, the greatest relative error observed was 3.4×10^{-6}. (For $x - .1999999975$, $y_1 = .00042499858$. However, the solution is actually .00042500002).

```
D                 GM      1024                                      2000   00  00000  00000
D                1GM      1065                                      2051   00  00000  00000
D                2GM     49880                                     63230   00  00000  00000
D                3GM     49921                                     63301   00  00000  00000
D                OGC      1083                                      2073   00  00000  00000
D                1GC     49939                                     63323   00  00000  00000
D                OGT        23                                        27   00  00000  00000
2GM00     FS                                                       63230   57  00000  00000
2GM01     MJ                                                       63231   45  00000  00000
2GM02     MJ             1GM00                TO SET UP            63232   45  00000  02051
2GM03     MJ             OGM36                ENTER                63233   45  00000  02044
2GM04     TP  OGT09      Q                    RE-ENTER             63234   11  00040  10000
2GM05     QJ  OGM06      OGM40                WHAT                 63235   44  02006  02050
2GM06     QJ  OGM38      OGM07                  PASS               63236   44  02046  02007
2GM07     TP  Q          OGT09                STORE CTR            63237   11  10000  00040
2GM08     RA  OGM10      OGC01                UP PASS              63240   21  02012  02074
2GM09     RP  30003      OGM11                STORE                63241   75  30003  02013
2GM10     TP             OGT05                  PASS CONS          63242   11  00000  00034
2GM11     TU             OGM15                RESET                63243   15  00000  02017
2GM12     TV             OGM31                  ADDRESSES          63244   16  00000  02037
2GM13     TP             OGT08                RESET N-1            63245   11  00000  00037
2GM14     RP  30003      OGM16                                     63246   75  30003  02020
2GM15     TP             OGT00                                     63247   11  00000  00027
2GM16     MP             OGT00                CALC K               63250   71  00000  00027
2GM17     LA  A                                 SHIFT L            63251   54  20000  00000
2GM18     TP  A          OGT03                STORE K              63252   11  20000  00032
2GM19     MP  OGT05      OGT03                AK                   63253   71  00034  00032
2GM20     MA  OGT06      OGT02                AK BQ                63254   72  00035  00031
2GM21     LA  A             38                                     63255   54  20000  00046
2GM22     TP  A          OGT04                STORE R              63256   11  20000  00033
2GM23     AT  A          A                    2R                   63257   35  20000  20000
2GM24     AT  OGT04      A                    3R                   63260   35  00033  20000
2GM25     LA  A             34                                     63261   54  20000  00042
2GM26     MA  OGT07      OGT03                3R CK                63262   72  00036  00032
2GM27     LA  A             38                ADD OLD Q            63263   54  20000  00046
2GM28     AT  OGT02      OGT02                EQUALS Q             63264   35  00031  00031
2GM29     RA  OGT01      OGT04                NEW Y                63265   21  00030  00033
2GM30     RP  30002      OGM32                STORE                63266   75  30002  02040
2GM31     TP  OGT01                             Y AND Q            63267   11  00030  00000
2GM32     RA  OGM15      OGC01                ADVANCE              63270   21  02017  02074
2GM33     RA  OGM31      OGC02                  ADDRESSES          63271   21  02037  02075
2GM34     IJ  OGT08      OGM14                CYCLE N              63272   41  00037  02016
2GM35     MJ                                                       63273   45  00000  00000
2GM36     TU  OGC00      OGM10                                     63274   15  02073  02012
2GM37     TP  OGM21      Q                                         63275   11  02025  10000
2GM38     RA                                                       63276   21  00000  00000
2GM39     MJ             OGM07                                     63277   45  00000  02007
2GM40     QJ  OGM01      OGM07                                     63300   44  02001  02007
3GM00     SP  OGM01      00015                                     63301   31  02001  00017
3GM01     TU  A          1GM02                                     63302   15  20000  02053
3GM02     TP             OGT00                                     63303   11  00000  00027
3GM03     TV  OGT00      OGM35                                     63304   16  00027  02043
3GM04     TU  OGT00      1GM05                                     63305   15  00027  02056
3GM05     TV             OGM17                                     63306   16  00000  02021
3GM06     RA  OGT00         15                                     63307   21  00027  00017
3GM07     RP  10003      1GM09                                     63310   75  10003  02062
3GM08     TU  OGT00      OGM11                                     63311   15  00027  02013
3GM09     RA  OGM13         15                                     63312   21  02015  00017
3GM10     AT     15      A                                         63313   35  00017  20000
3GM11     TU  A          OGM16                                     63314   15  20000  02020
3GM12     TU  A          OGM38                                     63315   15  20000  02046
3GM13     LQ  A             21                                     63316   55  20000  00025
3GM14     TV  Q          OGM38                                     63317   16  10000  02046
3GM15     RA  OGM38         15                                     63320   21  02046  00017
3GM16     RA  OGM01         16                                     63321   21  02001  00020
3GM17     MJ             OGM01                                     63322   45  00000  02001
1GC00           GC                                                 63323   00  02073  00000
1GC01               3          B                                   63324   00  00003  00000
1GC02                        3 B                                   63325   00  00000  00003
1GC03      5                 -    1    34   A1  SCALE   34         63326   10  00000  00000
1GC04-    01                           34  OB1          34         63327   57  77777  77777
1GC05-    05                 -    1    34  OC1          34         63330   67  77777  77777
1GC06     02  92893      21881-0  1    34   A2  SCALE   34         63331   04  53730  31460
1GC07-    02  92893      21881-0  1    34  OB2          34         63332   73  24047  46317
1GC08-    02  92893      21881-001     34  OC2  SCALE   34         63333   73  24047  46317
1GC09      1  70710      67812         34   A3          34         63334   33  24047  46320
1GC10-    01  70710      67812         34  OB3          34         63335   44  53730  31457
1GC11-    01  70710      67812         34  OC3          34         63336   44  53730  31457
1GC12      1  66666      66667-0  1    34   A4          34         63337   02  52525  25253
1GC13-    03  33333      33333-0  1    34  OB4          34         63340   72  52525  25252
1GC14-    05                 -    1    34  OC4          34         63341   67  77777  77777
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Floating Point Sine-Cosine Routine

Specifications

Identification Tag:	SIN-3
Type:	Subroutine
Assembly Routine Spec:	SUB 50075 05915
Storage:	44 instructions, addresses 00S00 thru 00S43 01S00 thru 01S43 15 constants in program, addresses 02S00 thru 02S14 03S00 thru 03S14 59 words total program storage, addresses 00S00 thru 00S43 01S00 thru 01S43 02S00 thru 02S14 03S00 thru 03S14 2 words temporary storage pool used, addresses 00027b thru 00030b The constant pool is used by this routine
Program Entrances:	00T02 (01T02) for sine, 00T04 (01T04) for cosine
Program Exit:	00T01 (01T01)
Drum Assignment:	Addresses 63533b thru 63625b
Machine Time:	3.9 ms average, 4.8 ms maximum
Mode of Operation:	Floating point

Coded by:	M. Perry	July 27, 1955
Code Checked by:	R. Bigelow	July 27, 1955
Machine Checked by:	M. Perry	August 8, 1955
Approved by:	W. Bauer	August 10, 1955

Description

When supplied with an argument X in SNAP form, this routine will evaluate Sine X or Cosine X (depending on which of the two entrances is used) using a Rand Polynomial Approximation, producing the answer in SNAP form.

Programming Instructions

This routine can be inserted into a program by CMP-0 by the use of a "SUB" card in the input deck.

1. Place the double length extension of X in the accumulator.

   X must be in radians and must be in SNAP form.

2. Return jump to the subroutine. Assuming that the subroutine was assigned to region 00K00 for assembly, use either the instruction RJ 00K01 00K02 for the sine, or the instruction RJ 00K01 00K04 for the cosine.

3. At the time of exit from the subroutine, the double length extension of sine X (or cosine X) in SNAP form will be in the accumulator.

Error Analysis

The error in the result of this routine is less than 2^{-26}; however, the significance of the sine (or cosine) cannot exceed the significance of the fractional part of X.

Mathematical Method

1. Let $y = (2/\pi)X$, then sine $X = \sin(\pi/2)(y)$

   $$\text{cosine } X = \text{sine } (\pi/2)(y + 1)$$

2. Divide y (or y + 1) into an integral part R, and a fractional part S.

3. R defines the quadrant into which X falls. Let R' be the two low order positions of R, since in binary notation, any other positions merely define a number of complete revolutions.

4. R' is a number one less than the number of the quadrant into which X falls.

5. S defines the displacement (in a position direction) within the quadrant indicated by R'.

6. Therefore, if R' = 00 Let Z = S first quadrant
   R' - 01 Let Z = (1-S) second quadrant
   R' = 10 Let Z = (-S) third quadrant
   R' = 11 Let Z = (1-S) fourth quadrant

7. Sine (or cosine) $X = \sin(\pi/2)Z$.

8. $\sin(\pi/2)X$ is approximated by the Rand Polynomial Approximation Number 16, using argument Z.

Range of Variable

No alarm condition is recognized by this routine. However, as X approaches $\pm\ 2^{27}$ the number of significant digits in Sine X (or Cosine X) approaches zero, and X merely defines a number of revolutions and does not significantly designate an angle.

```
D              00S00  50075          SIN ROUTN 44  63533  00  00000  00000
D              01S00  01024          TO BE ALTERD   2000  00  00000  00000
D              02S00  50119          SIN CONST 15  63607  00  00000  00000
D              03S00  01068          TO BE ALTERD   2054  00  00000  00000
    S00                              ALARM EXIT    63533  00  00000  00000
    S01    MJ  00000                 NORMAL EXIT   63534  45  00000  00000
    S02    TP  00013  01S42          SIN ENTRY     63535  11  00015  02052
    S03    MJ  00000  01S05                        63536  45  00000  02005
    S04    TP  03S00  01S42          COS ENTRY     63537  11  02054  02052
    S05    TP  A0000  Q0000                        63540  11  20000  10000
    S06    QT  03S01  00023          M             63541  51  02055  00027
    S07    QT  03S02  00024          E-128         63542  51  02056  00030
    S08    QJ  01S09  01S11          NEG           63543  44  02011  02013
    S09    CC  00024  03S02                        63544  27  00030  02056
    S10    RA  00023  03S02                        63545  21  00027  02056
    S11    LQ  00023  00008          M          35 63546  55  00027  00010
    S12    RS  00024  03S00          E             63547  23  00030  02054
    S13    TM  A0000  A0000          E ABS         63550  12  20000  20000
    S14    TJ  03S03  01S16                        63551  42  02057  02020
    S15    TP  00013  00023                        63552  11  00015  00027
    S16    LQ  00024  20009                        63553  55  00030  20011
    S17    AT  03S04  01S20                        63554  35  02060  02024
    S18    MP  03S05  00023          M IN QDRNTS   63555  71  02061  00027
    S19    TP  B0000  A0000                        63556  11  30000  20000
    S20    LA  A0000  00                           63557  54  20000  00000
    S21    AT  01S42  Q0000                        63560  35  02052  10000
    S22    QT  03S06  00023          Z             63561  51  02062  00027
    S23    TP  03S07  01S42                        63562  11  02063  02052
    S24    QJ  01S25  01S26                        63563  44  02031  02032
    S25    TP  03S08  01S42                        63564  11  02064  02052
    S26    QJ  01S27  01S29                        63565  44  02033  02035
    S27    SP  03S00  00000          1 MINUS Z     63566  31  02054  00000
    S28    ST  00023  00023          ARG OF POLY   63567  36  00027  00027
    S29    MP  00023  00023          ARG SQUARED   63570  71  00027  00027
    S30    TP  B0000  00024                     33 63571  11  30000  00030
    S31    SP  03S09  00035          C-9           63572  31  02065  00043
    S32    RP  20004  01S34                        63573  75  20004  02042
    S33    PM  03S10  00024          POLY          63574  24  02066  00030
    S34    MP  B0000  00023            EVALUATION  63575  71  30000  00027
    S35    TP  B0000  A0000                        63576  11  30000  20000
    S36    ZJ  01S37  01S01                        63577  47  02045  02001
    S37    SF  A0000  00024          SCALE M       63600  74  20000  00030
    S38    TP  A0000  00023                        63601  11  20000  00027
    S39    SP  00024  00000                        63602  31  00030  00000
    S40    SA  03S14  00035          E-128 FINAL   63603  32  02072  00043
    S41    SA  00023  00027          PACKED        63604  32  00027  00033
    S42    TP  B0000  A0000          EXTEND        63605  11  30000  20000
    S43    MJ  00000  01S01          OUT           63606  45  00000  02001
   2S00    20  00000  00000  B       1 OR 128      63607  20  00000  00000
   2S01    00  07777  77777  B       MASK          63610  00  07777  77777
   2S02    77  70000  00000  B       MASK          63611  77  70000  00000
   2S03    03  30000  00000  B       27            63612  03  30000  00000
   2S04    54  20000  04107  B       ADDRESS MOD   63613  54  20000  04107
   2S05    24  27630  15562  B       2 PI       35 63614  24  27630  15562
   2S06    17  77777  77777  B       MASK          63615  17  77777  77777
   2S07    11  30000  20000  B       POS INTO A    63616  11  30000  20000
   2S08    13  30000  20000  B       NEG INTO A    63617  13  30000  20000
   2S09    00  02366  57351  B       C-9        38 63620  00  02366  57351
   2S10    77  66333  14715  B       C-7        35 63621  77  66333  14715
   2S11    00  50632  12755  B       C-5        33 63622  00  50632  12755
   2S12    76  55242  07644  B       C-3        31 63623  76  55242  07644
   2S13    00  62207  73244  B       C-1        29 63624  00  62207  73244
   2S14    00  00000  00076  B       62            63625  00  00000  00076
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Arcsine-Arcosine Routine

Specifications

Identification Tag:	SNI-1
Type:	Subroutine
Assembly Routine Spec:	SUB 50410 08014
Storage:	66 instructions, addresses DSC00 (ASC00) thru DSC65 (ASC65)
	14 constants in program, addresses DSC66 (ASC66) thru DSC79 (ASC79)
	80 words total program storage, addresses DSC00 (ASC00) thru DSC79 (ASC79)
	7 words temporary storage pool used, addresses 00027b thru 00035b
	The constant pool is used by this routine
Program Entrances:	DSC03 for arcsine, DSC02 for arcosine
Program Exit:	DSC01
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Addresses 64252b thru 64371b
Machine Time:	6.0 ms average, 6.6 ms maximum machine time
Mode of Operation:	Fixed point

Coded by:	A. Franck (ERA)	May 14, 1955
Translated by:	D. Gantner	August 16, 1955
Machine Checked by:	T. Tack	August 25, 1955
Approved by:	W. Bauer	September 12, 1955

Description

This subroutine computes F(x) = arcsin x or F(x) = arcos x (depending on which of two entrances is used) by use of a polynomial approximation.

The routine was originally coded by Dr. A. Franck of ERA and has been adopted for use at The Ramo-Wooldridge Corporation.

Notation

X = sine or cosine of an angle F(x).

F(x) = the computed angle in radians whose sine or cosine is x.

A_J = Rand Constants (See Rand Sheet No. 39).

Programming Instructions

This routine can be assembled into a program by use of a "SUB" card in the input deck.

1. Place the argument scaled by 2^{33} (i.e. $X \cdot 2^{33}$) in A.

2. Enter the routine with an RJ instruction. If the routine was assigned to region 00K00 for assembly use the instruction

   RJ 00K01 00K03 for the arcsine, or

   RJ 00K01 00K02 for the arcosine.

3. At the time of exit from the routine the result $F(x) \cdot 2^{33}$ is left in A.

Alarm Conditions

If the |x| is greater than one an alarm exit will occur. The word "alarm" and the address of the cell in the main program containing the RJ instruction which was used to enter SNI-1 will be printed on the flexowriter.

Pushing the start button after the alarm halt will transfer control to the exit of SNI-1.

```
D                DSC00  50410
D                ASC00  01024
DSC00     37     75701  75702  B      ALARM EXIT
DSC01     MJ     00000  00000         ROUTINE EXIT
DSC02     MJ     00000  ASC57         ARCCOS  ENTRY
DSC03     MJ     00000  ASC60         ARCSIN  ENTRY
DSC04     TP     00013  00024
DSC05     TM     A0000  00025
DSC06     TJ     ASC66  ASC00
DSC07     TJ     ASC67  ASC09
DSC08     MJ     00000  ASC00
DSC09     TP     A0000  Q0000
DSC10     ZJ     ASC11  ASC52
DSC11     SJ     ASC12  ASC13
DSC12     TP     00016  00024
DSC13     TM     A0000  A0000
DSC14     EJ     ASC68  ASC55
DSC15     MP     00025  ASC69
DSC16     LA     A0000  00037
DSC17     AT     ASC70  00026
DSC18     MP     00025  00026
DSC19     LA     A0000  00037
DSC20     AT     ASC71  00026
DSC21     MP     00025  00026
DSC22     LA     A0000  00039
DSC23     AT     ASC72  00026
DSC24     MP     00025  00026
DSC25     LA     A0000  00038
DSC26     AT     ASC73  00026
DSC27     MP     00025  00026
DSC28     LA     A0000  00038
DSC29     AT     ASC74  00026
DSC30     MP     00025  00026
DSC31     LA     A0000  00038
DSC32     AT     ASC75  00026
DSC33     MP     00025  00026
DSC34     LA     A0000  00036
DSC35     AT     ASC76  00026
DSC36     TN     00025  A0000
DSC37     SA     ASC68  00002
DSC38     TP     ASC78  00027
DSC39     EJ     ASC78  ASC47
DSC40     TP     A0000  00028
DSC41     SP     00028  00034
DSC42     DV     00027  00029
DSC43     LA     00027  00071
DSC44     RS     Q0000  00027
DSC45     RA     00027  00029
```

DSC46	QJ	ASC41	ASC47	
DSC47	MP	A0000	00026	
DSC48	LA	A0000	00037	
DSC49	ST	ASC77	Q0000	
DSC50	IJ	00024	ASC52	
DSC51	TN	Q0000	Q0000	
DSC52	RS	Q0000	00023	
DSC53	DV	ASC79	A0000	
DSC54	MJ	00000	ASC01	
DSC55	TN	ASC77	Q0000	
DSC56	MJ	00000	ASC50	
DSC57	TP	ASC77	00023	
DSC58	TV	ASC65	ASC54	
DSC59	MJ	00000	ASC04	
DSC60	TP	00013	00023	
DSC61	TV	ASC64	ASC54	
DSC62	MJ	00000	ASC04	
DSC63	TN	A0000	A0000	
DSC64	MJ	00000	ASC01	
DSC65	00	00000	ASC63	
DSC66	67	77777	77777	B
DSC67	10	00000	00001	B
DSC68	10	00000	00000	B
DSC69	53	24135	20070	B
DSC70	33	24414	25535	B
DSC71	56	40071	51545	B
DSC72	37	50417	41233	B
DSC73	46	23706	66522	B
DSC74	26	61651	66073	B
DSC75	44	42003	30653	B
DSC76	31	10375	51633	B
DSC77	31	10375	52421	B
DSC78	37	77777	77777	B
DSC79	00	00000	00002	B

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Floating Point Arcsine-Arcosine Routine

Specifications

Identification Tag:	SNI-2
Type:	Subroutine
Assembly Routine Spec:	SUB 50349 06112
Storage:	49 instructions, addresses 0AS00 (1AS00) thru 0AS48 (1AS48)
	12 constants in program, addresses 2AS00 (3AS00) thru 2AS11 (3AS11)
	61 words total program storage, addresses 0AS00 (1AS00) thru 0AS48 (1AS48) 2AS00 (3AS00) thru 2AS11 (3AS11)
	3 words temporary storage pool used, addresses 00027b thru 00031b
	The constant pool is used by this routine
Program Entrances:	0AS02 for arcsine, 0AS03 for arcosine
Program Exit:	0AS01
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Address 64155b thru 64251b
Machine Time:	7.17 ms average, 8.74 ms maximum
Mode of Operation:	Floating point

Coded by:	M. Perry	August 25, 1955
Code Checked by:	R. Bigelow	August 28, 1955
Machine Checked by:	M. Perry	September 7, 1955
Approved by:	W. Bauer	September 12, 1955

Description

When supplied with an argument X in SNAP form, this routine will compute the arcsine or the arcosine of X (depending on which of two entrances was used) using a Rand Polynomial Approximation producing the answer in SNAP form.

Programming Instructions

This routine can be inserted into a program by CMP-0 by the use of a "SUB" card in the input deck.

1. Place the argument X in the accumulator. X must be in SNAP form.

2. Return Jump to the subroutine. Assuming that the subroutine was assigned to region 00K00 for assembly, use the instruction RJ 00K01 00K02 for arcsine, or RJ 00K01 00K03 for arcosine.

3. At the time of exit from the subroutine, the double length extension of arcsine X, or arcosine X, in SNAP form will be in the accumulator.

Error Analysis

The error in the result produced by this subroutine is less than 2^{-26}.

Mathematical Analysis

1. The Rand Polynomial Number 39 is evaluated using the absolute value of X as the argument. Designate the result as P(X).

2. The square root of 1 minus the absolute value of X is found using the square root subroutine within SNAP. Designate this results as R(X).

3. If X is positive, let $Y = P(X)R(X)$

   If X is negative, let $Y = \pi - P(X)R(X)$

4. Arcsine = X = $(\pi/2) - Y$

   Arcosine X = Y

5. This procedure places arcsine X in the first or fourth quadrant, and arcosine X in the first or second quadrant.

Range of Variable

An alarm will occur if the argument is outside the range -1 X 1. Any argument within this range will give results with the above stated accuracy.

Alarm Conditions

An alarm print will occur if the argument is outside the permissible range. The flexowriter will print "alarm" and the address of the cell in the main program containing the RJ instruction which was used to enter SNI-2.

Special Note

The SNAP floating point routine must be in electrostatic storage before this subroutine can be used, since the square root subroutine is used within this routine.

```
D                  00K00  00906      SNAP CONS
D                  00T00  00722      SNAP TEMP
D                  00C00  00754      SNAP EXIT
D                  SRT00  00927      SNAP SQ ROOT
D                  00P00  00877      SNAP CONS
D                  0AS00  50349      ARCSN RTN  49
D                  1AS00  01024      REL 2000   49
D                  2AS00  50398      ARCSN CNS  11
D                  3AS00  01073      REL 2000   11
  AS00      37     75701  75702  B   ALARM EXIT
  AS01      MJ                       NORMAL EXIT
  AS02      TP     3AS11  00013      ARC SIN ENTR
  AS03      TP     00013  00023      ARC COS ENTR
  AS04      TP     A0000  00024      X
  AS05      TM     A0000  A0000      X ABS
  AS06      TP     B0000  00013      CLEAR 13
  AS07      TP     B0000  00025      CLEAR 25
  AS08      ST     3AS08  Q0000      X-50
  AS09      TJ     00K14  1AS13      X EQUAL ZERO
  AS10      QT     00K03  00025      M
  AS11      SP     Q0000  00008
  AS12      TV     B0000  1AS13
  AS13      LA     00025  00000      M FIXED    33
  AS14      ST     3AS09  Q0000      X-1
  AS15      SF     Q0000  00T04
  AS16      SJ     1AS17  1AS00
  AS17      LA     A0000  00027
  AS18      TN     B0000  00T03      1-X FLOATED
  AS19      LA     00T04  00027      SF IN T04
  AS20      TP     1AS01  00C00
  AS21      RJ     00C00  SRT00      FIND ROOT
  AS22      TP     00P02  00C00      REPAIR SNAP
  AS23      SP     00T04  00000
  AS24      SS     SRT26  00008
  AS25      TV     B0000  1AS31
  AS26      SN     3AS00  00036      A-7
  AS27      RP     20007  1AS29      EVAL RAND
  AS28      PM     3AS01  00025      POLY       29
  AS29      MP     B0000  00T03                 56
  AS30      LA     A0000  00010                 66
  AS31      SP     B0000  00000
  AS32      TP     B0000  00025                 31
```

AS33	TP	00024	Q0000			X
AS34	QJ	1AS35	1AS37			X NEG
AS35	SP	3AS11	00001			PI
AS36	ST	00025	00025			
AS37	SP	00023	00000			
AS38	ZJ	1AS39	1AS40			
AS39	ST	00025	00025			ARC SIN
AS40	SF	00025	00023			
AS41	ZJ	1AS42	1AS01			
AS42	LA	A0000	00027			
AS43	TP	B0000	00025			M FINAL
AS44	SP	00023	00027			
AS45	AT	3AS10	Q0000			E FINAL
AS46	CC	00025	Q0000			PACK
AS47	TP	A0000	A0000			EXTEND
AS48	MJ	00000	1AS01			OUT
2AS00	01	26249	11000	003	42	A—7
2AS01	06	67009	01000	003	40	A—6
2AS02	01	70881	25600	002	38	A—5
2AS03	03	08918	81000	002	36	A—4
2AS04	05	01743	04600	002	34	A—3
2AS05	08	89789	87400	002	32	A—2
2AS06	02	14598	80160	001	30	A—1
2AS07	01	57079	63050		28	A 0
2AS08	06	20000	00000	B		50
2AS09	10	00000	00001	B		1 33
2AS10	07	40000	00000	B		60
2AS11	01	57079	63268		31	PI OVER 2
START		00000	00000			TAPE OUTPUT
START		00000	00000			TAPE OUTPUT

MEMO Y SUM
MEMO Y SUM

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

STORAGE TO MAGNETIC TAPE TRANSFER ROUTINE

Specifications

Identification Tag:	STT-0
Type:	Service routine (with a program entry available)
Storage:	97 instructions, addresses 40006b, 74530b thru 74667b 11 constants in program, addresses 74670b thru 74702b 108 words total program storage, addresses 40006b, 74530b thru 74702b The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40006b
Program Entrance:	Address 74532b
Program Exit:	Address 74544b
Alarm Exit:	The alarm exit is not used by this routine
Drum Assignment:	Addresses 74530b thru 74702b
Machine Time:	5.6 seconds for transfer of (ES)
Mode of Operation:	Fixed point

Coded by:	R. Beach	May 11, 1955
Code Checked by:	C. Koos	August 14, 1955
Machine Checked by:	C. Koos	August 20, 1955
Approved by:	W. Bauer	August 30, 1955

Description

This routine transfers information from the internal computer memory to magnetic tape where it will be stored until read back in again by TST-0.

A parameter word is used to specify

1. The location of data to be stored
2. The MT unit to be used for storage
3. Whether or not MT is to be rewound to its original position after storage
4. The address to which control is to be transferred when the data is read back by TST-0.

When using STT-0 as a subroutine the parameter word follows the RJ instruction used to enter the routine. When using STT-0 as a service routine the parameter word is manually entered in Q when the computer halts (after being started at the service entrance).

At the time of entry the routine stores (ES) on the drum, bootstraps itself into ES, stores (A) and (Q) and obtains the parameter word. At is conclusion the routine restores (ES), (A) and (Q) and transfers control to the exit instruction.

The routine stores one block of information in addition to the number of blocks necessary for storing the data, as follows:

1. The first half of the first block contains (Q), (A), the parameter word and twelve zero words.
2. The second half of the first block thru the first half of the last block inclusive contain the information to be stored.
3. The last half of the last block contains the sum of the data (that is, the double precision sum of the split extension of each word), the number of blocks transferred to tape, the starting and stopping addresses for the transfer, and eleven zero words.

Parameter Word

This parameter word is of the form BC DEEFF GGGGG, where B, C, D, E, F, and G are all octal digits.

B. The octal digit B determines whether (ES) is to be stored on MT. If $B = 0$ (ES) will be stored, if $B \neq 0$ (ES) will not be stored.

C. The octal digit C determines whether MT is to be rewound to its original position after the data has been transferred. If $C = 0$ the rewind will be executed, if $C \neq 0$ it will not be.

D. The octal digit D determines the MT unit on which the data is to be stored. MT units are specified by the same digits used in the standard 1103 MT commands.

E. The two octal digit number EE specifies the address of the first word to be transferred from internal storage to tape. This number is the integer part of the first address devided by 8^3. That is, (EE)(512) is the address of the first cell to be transferred.

F. The two octal digit number FF specifies the address of the last word to be transferred. As in E above this number must also be a multiple of 512. (FF)(512) is the address of the last word to be transferred.

G. The V-address portion of the parameter word (GGGGG) specifies the address to which PAK is to be set when the transferred information is read back to internal memory by TST-0.

As an example consider the parameter word 01 24246 00017B. This specifies a transfer of (ES) and the contents of cells 42000b thru 45777b with no rewinding after the transfer. PAK will be set to 00017b by TST-0 when the routine is read back to internal memory.

Operating Instructions (to be followed when the routine is used as a service routine)

1. Set PAK to 40006b and start.

2. Computation halts with the MS instruction 56 00000 00010.

3. Enter the parameter word in Q and start.

4. Computation halts when the transfer is completed, setting PAK to the address specified in the parameter word.

Programming Instructions (to be followed when the routine is to be used as a subroutine)

1. Enter the routine with the RJ instruction 37 74544 74532. If the RJ instruction is stored at address n the parameter word should be in address n + 1 and at its conclusion the routine will transfer control to the instruction in address n + 2.

Restore

To restore (ES), (A), and (Q) at any time before normal completion set PAK to 40040b and start.

The magnetic tape will be rewound at this time if the parameter word specifies a rewind.

123	00	00000	00000
200	00	00000	00000
1200	00	00000	00000
137	00	00000	00000
74530	00	00000	00000
74545	00	00000	00000
40000	00	00000	00000
40006	45	00000	74530
74530	16	74666	74553
74531	45	00000	74533
74532	16	74701	74553
74533	11	00000	76000
74534	11	74553	00000
74535	75	31777	74537
74536	11	00001	76001
74537	75	30135	00001
74540	11	74546	00001
74541	75	31777	74543
74542	11	76001	00001
74543	11	76000	00000
74544	45	00000	00000
74545	56	00000	00000
74546	16	00135	40040
74547	11	10000	00137
74550	11	20000	00141
74551	54	20000	00044
74552	11	20000	00140
74553	45	00000	00000
74554	23	10000	10000
74555	56	00000	00011
74556	11	10000	00142
74557	16	10000	74545
74560	16	00012	74544
74561	45	00000	00025
74562	31	74544	00017
74563	15	20000	00022
74564	11	00011	20000
74565	42	00022	00022
74566	21	00022	00133
74567	11	00000	00142
74570	21	74544	00123
74571	11	10000	10000
74572	11	00124	10000
74573	53	00142	00057
74574	53	00142	00112
74575	53	00142	00121
74576	75	30020	00033
74577	23	00143	00143

74600	75	30020	00035
74601	11	00137	00200
74602	55	10000	00003
74603	51	00142	20000
74604	47	00041	00040
74605	16	00030	00120
74606	55	10000	00003
74607	51	00142	20000
74610	47	00065	00044
74611	11	00127	00146
74612	21	00143	00125
74613	75	31000	00050
74614	11	76000	00220
74615	31	00144	00044
74616	32	00145	00000
74617	75	21000	00054
74620	32	00220	00000
74621	11	20000	00145
74622	54	20000	00044
74623	11	20000	00144
74624	65	00020	00200
74625	75	30020	00062
74626	11	01200	00200
74627	21	00047	00126
74630	43	00146	00065
74631	45	00000	00045
74632	16	00070	00063
74633	55	00142	10003
74634	51	00130	00147
74635	42	00132	00100
74636	55	10000	00006
74637	51	00130	00146
74640	11	00111	00047
74641	15	20000	00047
74642	11	00047	00146
74643	15	00147	00047
74644	45	00000	00045
74645	21	00143	00131
74646	31	00144	00044
74647	32	00145	00000
74650	75	20005	00105
74651	32	00137	00000
74652	11	20000	00145
74653	54	20000	00044
74654	11	20000	00144
74655	75	30020	00112
74656	11	00143	00220

74657	65	00001	00200
74660	11	00132	10000
74661	53	00143	00121
74662	11	00137	10000
74663	31	00140	00044
74664	32	00141	00000
74665	45	00000	74541
74666	67	00000	00007
74667	45	00000	74541
74670	00	00000	00001
74671	00	70000	00000
74672	00	00020	00000
74673	00	01000	00000
74674	12	00000	00220
74675	00	77000	00000
74676	00	00001	00000
74677	00	07777	00000
74700	00	76000	00000
74701	00	00000	00015
74702	00	00000	00115

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Magnetic Tape to Storage Transfer Routine

Specifications

Identification Tag:	TST-0
Type:	Service routine (with a program entry available)
Storage:	95 instructions, addresses 40007b, 74340b thru 74475b
	8 constants in program, addresses 74476b thru 74505b
	103 words total program storage, addresses 40007b, 74340b thru 74505b
	The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40007
Program Entrance:	Address 74342
Program Exit:	Address 74354
Alarm Exit:	The alarm exit is used by this routine
Drum Assignment:	Addresses 74340b thru 74505b
Machine Time:	5.6 seconds for transfer of (ES)
Mode of Operation:	Fixed point

Coded by:	R. Beach	May 11, 1955
Code Checked by:	C. Koos	August 13, 1955
Machine Checked by:	C. Koos	August 21, 1955
Approved by:	W. Bauer	August 30, 1955

Description

This routine has been designed to read back into the internal computer memory the information transferred to MT by STT-0. A parameter word is used to tell the routine which MT contains the information to be transferred and the address to which PAK is to be set after the routine has finished operating.

When used as a subroutine, the parameter word follows the RJ instruction transferring control to TST-0. When used as a service routine the parameter word is entered in the Q register before activating the routine.

The routine stores (ES) on the drum, bootstraps itself into ES and reads in one block from magnetic tape. It examines the parameter word used by STT-0 and loads the specified portions of memory while computing the sum as the data is transferred. The sum is checked against the sum placed on MT by STT-0.

If the sum is correct, the parameter word is consulted to determine the address to which PAK is to be set and the proper address is placed in the exit instruction. The parameter word from STT-0 is checked to determine whether or not the MT is to be rewound after the transfer and a rewind command given if rewind was specified when STT-0 was used to store the data. (A) and (Q) are then set from values stored on MT, (ES) is restored, and control is transferred to the exit instruction.

Parameter Word

The form of the parameter word is OX YOOOO ZZZZZ, where X, Y, and Z are octal digits.

X. The octal digit X determines the cell to which control will be transferred to at the conclusion of the routine.

   If X = 0 control will be transferred to the address specified in the parameter word used for STT-0 when the data was transferred to magnetic tape. If X ≠ 0 control will be transferred to ZZZZZ.

Y. The octal digit Y determines which MT unit will be selected. MT units are specified by the same digits used in the standard 1103 MT commands.

Z. The V-address of the parameter word (ZZZZZ) specifies the address to which control will be transferred at the conclusion of the routine (see X above).

Operating Instructions (to be followed when TST-0 is used as a service routine)

1. Manually enter the parameter word in Q.

2. Set PAK to 40007b and start.

3. Computation will halt after a successful transfer with PAK set as specified (see "Parameter Word" above).

Programming Instructions

1. Use the RJ instruction 37 74354 74342B to enter TST-0. The cell immediately following the RJ instruction must contain the parameter word.

2. After successful transfer control will be transferred to the cell specified by the parameter word.

Alarm Conditions

If the sum test fails ALR-1 is entered and "TST-0 75777" is printed on the flexowriter.

Starting after the alarm halt causes a rewind of the tape and another transfer of the same data from MT.

Restore

If, at any time during its operation, TST-0 is interrupted (or after an alarm print), PAK set to 40040b and the machine started, the routine will

1. Rewind MT (if this had been specified)

2. Restore (ES), (A), and (Q)

3. Transfer control to the TST-0 exit instruction

121	00	00000	00000
200	00	00000	00000
1220	00	00000	00000
132	00	00000	00000
74340	00	00000	00000
74355	00	00000	00000
40000	00	00000	00000
40007	45	00000	74340
74340	16	74350	74347
74341	45	00000	74343
74342	16	74412	74347
74343	11	00000	76000
74344	11	74354	00000
74345	75	31777	74347
74346	11	00001	76001
74347	75	30130	00000
74350	11	74356	00001
74351	75	31777	74353
74352	11	76001	00001
74353	11	76000	00000
74354	45	00000	00000
74355	56	00000	00000
74356	11	10000	00132
74357	45	00000	00006
74360	42	00005	00005
74361	21	00005	00130
74362	11	00000	00132
74363	16	00066	40040
74364	11	00121	10000
74365	53	00132	00013
74366	53	00132	00024
74367	53	00132	00071
74370	64	00001	00200
74371	23	00133	00133
74372	23	00134	00134
74373	55	00121	10006
74374	51	00203	20000
74375	47	00043	00021
74376	16	00071	00035
74377	11	00122	00135
74400	16	00105	00041
74401	64	00020	00240
74402	31	00133	00044
74403	32	00134	00000
74404	75	21000	00031

74405	32	00220	00000
74406	11	20000	00134
74407	54	20000	00044
74410	11	20000	00133
74411	75	31000	00036
74412	11	00220	00115
74413	75	30020	00040
74414	11	01220	00220
74415	21	00035	00123
74416	43	00135	00106
74417	45	00000	00024
74420	16	00047	00041
74421	15	00030	00035
74422	55	00203	10030
74423	51	00124	00136
74424	42	00125	00056
74425	55	10000	00006
74426	51	00124	00135
74427	16	20000	00035
74430	11	00035	00135
74431	16	00136	00035
74432	45	00000	00024
74433	11	00126	10000
74434	53	00220	00071
74435	31	00133	00044
74436	32	00134	00000
74437	75	20004	00064
74440	32	00200	00000
74441	32	00220	00000
74442	34	00222	00044
74443	43	00221	00073
74444	11	00127	75756
74445	37	75701	75702
74446	67	00000	76000
74447	45	00000	00013
74450	55	00121	10003
74451	51	00132	20000
74452	47	00077	00076
74453	16	00203	00132
74454	11	74362	20000
74455	43	00005	00112
74456	16	00132	74354
74457	51	00203	20000
74460	47	00106	00104

74461	15	00071	00105
74462	67	00000	00043
74463	31	00201	00044
74464	32	00202	00000
74465	11	00200	10000
74466	45	00000	74351
74467	16	00132	74355
74470	16	00112	74354
74471	45	00000	00102
74472	31	74354	00017
74473	15	20000	00005
74474	11	00001	20000
74475	45	00000	00003
74476	00	70000	00000
74477	11	00221	00000
74500	00	00000	01000
74501	00	00000	77000
74502	00	00000	40000
74503	00	07777	00000
74504	01	24015	63704
74505	00	76000	00000

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Utility Routine Transfer-Magnetic Tape to Drum

Specifications

Identification Tag:	URT-1
Type:	Service routine (but not available as part of service routine library)
Storage:	45 instructions, addresses 00000b thru 00054b
	1 constant in program, address 00073b (remaining constants stored with instructions)
	The remainder of ES is used as temporary storage
	The constant and temporary storage pools are not used by this routine
Entrance:	MT Start
Machine Time:	Approximately 15 seconds for successful transfer of the service routine library only, or approximately 35 seconds for transfer if CMP-0 and the subroutine library are included.

Coded by:	R. Beach	April 1, 1955
Code Checked by:	R. Beach	April 2, 1955
Machine Checked by:	R. Beach	April 14, 1955
Approved by:	W. Bauer	August 23, 1955

Description

This routine is located in the first two blocks of magnetic tape unit zero and is specifically designed to transfer the library from magnetic tape to magnetic drum.

It operates in two different modes, the mode of operation having been selected when it was activated. Mode No. 1 loads addresses 40001b thru 40040b and 70000b thru 75777b only. Mode No. 2 loads these addresses and addresses 60000b thru 67777b.

This routine does not save the contents of ES since it is assumed that it will be used only when a complete reloading of the computer memory is necessary. An MT Start reads in the first 32 words of the routine and starts operation. The routine first reads in an additional 32 words from MT (remainder of the routine itself) and then checks its sum, which is stored at the end of the second block. In doing this, it also checks the sum of the service routine library which is stored in the second block.

After a successful sum check the routine reads in the 96 blocks needed to fill 70000b thru 75777b. Twenty-four blocks are read in at one time and transferred to MD, then read back into ES and summed. When all 96 blocks have been transferred the routine reads in one more block and transfers this into 40001 thru 40041, reads it back into ES, sums, and adds the sum of the sum of the 96 blocks previously transferred. This computed sum is then checked against the correct sum. If the sum checks, the mode of operation is determined.

If Mode No. 1 has been selected a rewind instruction is given and the computer halts with the MS instruction 56 00000 40001, setting PAK to the FRI-0 starting address.

If Mode No. 2 has been selected TST-0 is activated to read in Rawoop and the subroutine library. A rewind instruction is given and the computer then halts with the MS instruction 56 00000 40010, setting PAK to the CMP-0 starting address.

Operating Instructions

I. To transfer the service routine library only

1. Select MT Start

2. Change PCR (if necessary) to select the proper MT unit

3. Start. The routine loads 40001b thru 40041b and 70000b thru 75777b and halts with the MS instruction 56 00000 40001, setting PAK to the FRI-0 starting address. Successful transfer takes about 15 seconds.

II. To transfer the service routine library, CMP-0, and the subroutine library

1. Select MT Start

2. Change PCR (if necessary) to select the proper MT unit

3. Make (AR) greater than zero

4. Start. The routine loads 40001b thru 40040b, 70000b thru 75777b, and 60000b thru 67777b and halts with the MS instruction 56 00000 40010 setting PAK to the starting address of CMP-0. Successful transfer takes about 35 seconds.

Alarm Conditions

1. If the machine halts on a final stop almost immediately after an MT start the transfer routine is not in ES correctly.

   Select MT start and start for another transfer. If the second transfer is not successful revert to the bootstrap procedure to load the library.

2. If the flexowriter prints an "e" and the machine halts with the MS instruction 56 00000 00051 the sum of the library transferred to the drum is not correct.

   Select MT start and start for another transfer.

3. When operating in Mode No. 2 TST-0 is activated after address 40001b thru 40040b and 70000b thru 75777b have been loaded successfully. If the sum test fails while loading addresses 60000b thru 67777b, the alarm routine prints the tag word TST-0 and the address 75777b. Starting causes rewind and another MT transfer to addresses 60000b thru 67777b.

Warning

After a successful transfer the computer halts but MT is still rewinding to its original position. If a Master Clear is executed and the machine started a reference to the rewinding MT (before rewinding is complete) will cause trouble. If no Master Clear has been executed the machine will wait for the rewinding to be completed.

1700	45	00000	00001
1701	11	20000	00172
1702	64	00001	00040
1703	31	00000	00000
1704	75	20075	00006
1705	32	00001	00000
1706	34	00077	00044
1707	43	00076	00012
1710	67	00002	00003
1711	57	77777	77777
1712	75	30003	00014
1713	23	00175	00175
1714	16	00010	00175
1715	64	00030	00200
1716	75	31400	00020
1717	11	00200	70000
1720	31	00017	00017
1721	15	20000	00023
1722	75	31400	00024
1723	11	00000	00200
1724	21	00017	00073
1725	31	00176	00044
1726	75	21401	00030
1727	32	00177	00000
1730	11	20000	00177
1731	54	20000	00044
1732	11	20000	00176
1733	41	00175	00015
1734	64	00001	00200
1735	75	30040	00037
1736	11	00200	40001
1737	75	30040	00041
1740	11	40001	00200
1741	31	00176	00044
1742	75	20041	00044
1743	32	00177	00000
1744	34	00075	00044
1745	43	00074	00052
1746	61	00000	00016
1747	34	00074	00044
1750	67	00143	00000
1751	56	00000	00051
1752	41	00172	00055
1753	16	00036	00051
1754	45	00000	00050

1755	37	74354	74342	
1756	01	00000	00057	
1757	67	00344	00000	
1760	56	00000	40010	
1761	00	00000	00000	
1762	00	00000	00000	
1763	00	00000	00000	
1764	00	00000	00000	
1765	00	00000	00000	
1766	00	00000	00000	
1767	00	00000	00000	
1770	00	00000	00000	
1771	00	00000	00000	
1772	00	00000	00000	
1773	00	00000	01400	
1774	00	00000	00000	LIB SUM HI
1775	00	00000	00000	LIB SUM LO
1776	00	00000	00000	URT 1 SUM HI
1777	00	00000	00000	URT 1 SUM LO

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Utility Routine Transfer Drum to Magnetic Tape

Specifications

Identification Tag:	URT-3
Type:	Service routine
Storage:	108 instructions, addresses 40000b and 00050b thru 00222b
	8 constants in program, addresses 00223b thru 00232b
	All of ES is used for temporary storage but not included with the program
	116 words total program storage, addresses 40000b and 00050b thru 00232b
	The temporary and constant storage pools are not used by this routine
Machine Time:	100 seconds approximately
Mode of Operation:	Fixed point

Coded by:	R. Beach	August 1, 1955
Machine Checked by:	R. Beach	August 15, 1955
Approved by:	W. Bauer	August 23, 1955

Description

Upon being entered the routine first sets up all references to magnetic tape to correspond to the unit selected when URT-3 is activated.

The contents of cells 40001b thru 40040b and 70000b thru 75777b are then summed and the sum placed in 01774b and 01775b. The contents of cells 01700b thru 01775b are then summed and the sum placed in 01776b and 01777b. The information in cells 01700b, 70000b thru 75777b, and 40001b thru 40040b are then transferred to MT in that order.

The contents of cells 60002b thru 67777b are summed and the sum placed in 60000b and 60001b.

STT-0 is entered to dump the information in cells 60000b thru 67777b (the subroutine library consisting of Rawoop and the subroutines).

URT-3 computes the sum of all information placed on MT, rewinds MT to its original position and reads back the data from MT, summing as it reads.

If the sum is correct, a BM instruction is given to return MT to its original position and computation halts with PAK set to 40001b, the FRI-0 starting address.

Operating Instructions

1. Select MD Start.

2. Set the number of the MT to be loaded in the low order octal digit of Q.

3. Start.

   URT-3 transfers the complete library to MT and halts with the MS instruction 56 00000 40001 after a successful transfer.

Alarm Conditions

If the sum of data read back from MT is not correct the alarm routine is entered; the tag word URT-3 and the address 00067 are printed on the flexowriter. The sum of the data on MT appears in A.

Restarting at this time initiates another transfer of data.

Warning

1. It is advisable to position MT at the first block before loading so that the MT can be repositioned, manually if necessary.

2. After a successful transfer the machine halts but MT is still rewinding to its original position. If a master clear is executed and the machine started a reference to the rewinding MT (before the rewinding is complete) will cause trouble. If no master clear has been executed the machine will wait for the reversing to be completed.

```
60000   00  00000  00000
   50   00  00000  00000
   70   00  00000  00000
  223   00  00000  00000
  233   00  00000  00000
  240   00  00000  00000
 1700   00  00000  00000
40000   00  00000  00000
   50   54  10000  00033
   51   11  00230  10000
   52   53  20000  00120
   53   53  20000  00124
   54   53  20000  00127
   55   53  20000  00175
   56   53  20000  00176
   57   53  20000  00202
   60   53  20000  00213
   61   53  20000  01702
   62   53  20000  01710
   63   53  20000  01715
   64   53  20000  01734
   65   53  20000  01750
   66   53  20000  01756
   67   53  20000  01757
   70   11  00214  00000
   71   75  30040  00073        TRNSFR
   72   11  40001  00240
   73   31  00240  00000          FROM 40001
   74   15  00226  00102
   75   75  20037  00101
   76   32  00241  00000
   77   31  01774  00044        TRNSFR
  100   32  01775  00000          AND SUM
  101   75  31400  00103          FROM 70000
  102   11  00000  00300
  103   75  21400  00105
  104   32  00300  00000
  105   11  20000  01775
  106   54  20000  00044
  107   11  20000  01774
  110   21  00102  00223
  111   42  00224  00077
  112   31  01700  00000        SUM URT 1
  113   75  20075  00115          AND STORE
  114   32  01701  00000          ON MT
  115   11  20000  01777
  116   54  20000  00044
  117   11  20000  01776
```

120	65	00002	01700
121	15	00226	00123
122	75	31400	00124
123	11	00000	00300
124	65	00030	00300
125	21	00123	00223
126	42	00224	00122
127	65	00001	00240
130	23	60000	60000
131	23	60001	60001
132	15	00130	00136
133	31	60000	00044
134	32	60001	00000
135	75	31000	00137
136	11	00000	00300
137	75	21000	00141
140	32	00300	00000
141	11	20000	60001
142	54	20000	00044
143	11	20000	60000
144	21	00136	00225
145	42	00226	00133
146	31	60000	00044
147	32	00175	00000
150	32	60000	00000
151	32	60001	00000
152	32	60001	00000
153	32	00227	00000
154	11	20000	00234
155	34	20000	00044
156	11	20000	00233
157	32	01774	00000
160	32	01776	00044
161	32	00233	00000
162	75	20003	00164
163	32	01775	00000
164	32	00234	00000
165	32	00234	00000
166	32	00232	00000
167	32	01777	00000
170	11	20000	00234
171	54	20000	00044
172	11	20000	00233
173	23	10000	10000
174	37	74544	74532
175	11	06070	00000
176	67	00344	00215

177	23	10000	10000
200	55	00230	10030
201	11	10000	00235
202	64	00014	00300
203	75	20600	00205
204	32	00300	00000
205	11	00235	10000
206	44	00207	00201
207	16	00176	00214
210	34	00234	00044
211	43	00233	00213
212	16	00213	00214
213	67	00344	00216
214	45	00000	00000
215	56	00000	40001
216	54	20000	00044
217	32	00234	00000
220	11	00231	75756
221	37	75701	75702
222	45	00000	00070
223	00	01400	00000
224	11	76000	00300
225	00	01000	00000
226	11	70000	00300
227	00	00201	00000
230	00	70000	00000
231	34	12015	67004
232	12	50000	00220

February 17, 1955 No. 33
(Revised)

RAMO-WOOLDRIDGE ONE-PASS ASSEMBLY PROGRAM

FOR THE ERA-1103

By

Jules Mersel and Thomas Tack

The Ramo-Wooldridge one-pass assembly program (Rawoop) is designed to translate an 1103 program originally coded in symbolic, regional, decimal form to its final octal form.

The program will accept instructions with symbolic addresses and numerical data in binary or decimal form. It will cause subroutines to be appropriately assembled in the program. The result of assembling a program will be output in a form to facilitate program check-out and rapid program read-in of the translated data.

Input-Output

Punched cards are used as input for Rawoop. The punched card has one 1103 word plus remarks on it or the card contains an instruction to the assembly program and remarks.

Rawoop's output is both punched cards and punched paper tape.

The output card contains an exact duplicate of the corresponding input card in addition to the octal translation of the input card's information. The programmer can obtain a side by side listing of his untranslated program, remarks, and translated program by running the output deck of cards on associated equipment such as the IBM 402.

The punched paper tape is a seventh-level bioctal tape complete with insert and check addresses, and is sufficient for putting the translated program into the 1103 either by using the ERA photo-electric reader or by using a Ferranti tape reader with an appropriate read-in program.

Input and Output Cards

The input and and output cards are standard 80-column, 12-row cards.

Columns 1-5 are used for the symbolic address of the untranslated word.

Columns 7-10 are used for the 1103 instruction or psuedo instruction.

Columns 12-16 are for the u address.

Columns 18-22 are for the v address.

Columns 24-26 are for decimal scaling information.

Columns 28-30 are for binary scaling information.

Columns 32-43 are for alpha-numeric remarks.

THE RAMO-WOOLDRIDGE CORPORATION

PAGE ____ OF ____

PROBLEM (Figure 1)

ADDRESS	OP	U	V	± D	± B	COMMENTS
D		1C	512			THIS IS NOT
D		99Z	630			
D		12T	49010			A PROGRAM.
D		73L	183			
D		T	2			IT IS AN
1C	RP	3 19	1C 2			
1C 1	TN	12T 5	Q			EXAMPLE OF
1C 2	SP	99Z13	9			
1C 3	MJ	3	99Z 1			THE TYPE OF
1C 4	LA	12T 4	2 18			
1C 5	ZJ	99Z	99Z02			WORDS
1C 6	—1	23456	78901	4	18	
1C 7	3	123		— 6	22	DESCRIBED ON
1C 8	1	6793		8	— 10	
1C 9	TP	Q	A			THE FOLLOW-
1C10 FP	21	12T28	12T12			
1C11	RJ	73L 1	73L 2			ING PAGES.
1C12	SJ	99Z23	99Z30			
1C13	1	024		3	F	
1C14		77777		B		
73L SU	B	51018	234			
START		01C00				
Columns 1 — 5	7 — 10	12 — 16	18 — 22	24 - 26	28-30	32 — 43

CODED BY ____ CHECKED BY ____

Figure 1

In addition to the above columns, the output cards contain the translated information in columns 47-67.

On the input cards, zeros need not be punched.

Speed of Assembly

Due to the 1103's ability to read and punch cards simultaneously while it is punching paper tape, Rawoop takes only a few seconds more than the total card reading time to execute its entire translation. Errors do not necessitate complete re-assembly. Consequently, Rawoop is exceedingly economical in its use of machine time.

Symbolic Addresses

A five character form, in keeping with 1103 machine form, is used by Rawoop for symbolic addresses. The first three characters designate the region of the address while the last two characters are the sequence number of the address within its region.

For example, 18J00 is the zeroth address in region 18J and 00C19 is the nineteenth address in region 01C (01C00 is the first address in region 01C). In keeping with the one-pass nature of Rawoop, the sequence numbers are consecutive decimal numbers. The absolute address assigned to 01C19 is nineteen greater than the address assigned to 01C00. Thus, the address structure has a regional character.

As indicated above, the first two characters of the relative address or region are numeric and the third character is alphabetic.

The absolute address for region 000 (all zeros) has already been chosen in Rawoop to be zero. Consequently, 00029 would have as its octal translation, 00035 and absolute machine addresses up to 99 will be correctly translated. The assembly program recognizes the alphabetic letter "O" as different from the numerical "0"; to avoid confusion, the programmer will probably not want to use symbolic addresses involving the letter O.

A, B, and Q Addresses

The accumulator, the B register (accumulator bits $A_{70} - A_{35}$), and the Q register must be addressed by putting an A, B, or Q in the leftmost column of either the u or v fields. The remaining four columns may have no punches or zero punches.

The octal translations of A, B, and Q are 20000, 30000, and 10000 respectively.

Addresses Involving j, k, and n

The command structure of the 1103 is such that the u and v addresses at times contain numbers rather than machine addresses as is the case in the SPuk, RPjn, w and MJjv commands.

The representation of j,n has j as the first character and n as the last four characters. The quantities j and n are written as decimal numbers. Thus a j,n of 30199 is translated as 30307 (octal). No distinction is made by Rawoop between j,n addresses and j addresses. If the programmer desires to use the last four characters of a j address to store a number (not a relative address), he may do so knowing that these four digits will be treated as the n in j,n addresses.

In the 1103 address structure, k addresses indicate left circular shifts of from 0 to 127 places at most. However, since the 1103 internal hardware occasionally makes it desirable to have the first octal bit of a k address be a number other than zero, k addresses will be treated in the same manner as j,n addresses; for example, 20017 becomes 20021 (octal).

Void Addresses

Certain of the 1103 commands such as FS- -, BJj,n -- have ignored addresses associated with them. All such addresses are treated by Rawoop as if they were relative addresses and are available for the storage of pre-setting addresses. All zeros, of course, are translated into all zeros.

Directory Cards

In a one-pass assembly program, it is necessary that at the beginning of the program sufficient information is supplied to enable all symbolic addresses to be assigned absolute addresses.

Rawoop does this by means of directory cards. A directory card has a D punched in column 1, the base word of the region (e.g. 01C00) in the u address columns, and the absolute decimal address of the base word in the v address columns. For examples, see figure 1.

For purposes of assigning decimal addresses to the drum, the convention was adopted that octal address 40,000 on the drum has the decimal address 40,000. Thus, the drum addresses range from 40,000 — 56,383.

Rawoop can handle up to 60 directory cards in any one assembly.

Symbolic Addresses of Program Data

With the exception of the D cards and the START card, all the input cards have a relative address punched in columns 1-5. This address is the address of the word to the right and completely determines the memory location into which this word will be read by the output bioctal tape.

Commands

The 1103 alphabetic representation of the commands is used. These two letter combinations, such as RA, are entered into columns 9 and 10. All the standard 1103 commands are recognized by Rawoop and this recognition implies knowing whether the addresses associated with the command are of the u, v, the jn, v, or of the u,k types.

In addition to the standard 1103 commands, the special commands IP, PM, and MM are recognized. The PM, MM commands and the availability of the B register for addressing are modifications on the Ramo-Wooldridge 1103. PM is a "polynomial multiply" command for polynomial evaluation whose octal equivalent is 24; MM is the "modified multiply-add" (faster in operation than MA) whose octal equivalent is 25. None of these modifications are used in the operation of Rawoop; the program will operate on any 1103 with reproducer and high-speed punch.

IP commands are treated as if the command structure were IPuv. However, for users of interpretive programs such as the Convair Flip, a psuedo command is available. The command FPabuv, where a and b are octal digits and u and v are relative addresses assigned to the electrostatic storage, is translated into 14 abu'v'. u' and v' are 12 bit numbers occupying the right hand 24 bits of the translated word with ab to the left of them. As is shown in figure 1, the FP goes into columns 7 and 8; the ab goes into columns 9 and 10 on the input card.

It is sometimes desirable that a relative address be placed in a word that has zeros for its command code. To enable this, the command 00 (zeros) is recognized and translated into 00. The u and v addresses must be symbolic addresses. As usual, however, an address of five zeros is translated into five zeros.

Decimal Numbers

Decimal numbers are presented to Rawoop as normalized numbers times a power of ten. The programmer also states the binary scaling factor to be applied to the resulting rounded binary number. For example, -739.1 is presented as -7.391×10^2.

All decimal numbers are normalized so that their absolute value lies between 1 and 9.99999 99999.

The sign of the number is in column 9, the integral part in column 10, and the fractional part in columns 12-16 and 18-22. The power of 10 allowed is from -10 to +10. This exponent goes into columns 24-26. The desired binary scale factor goes into columns 28-30. For examples, in line 13, Figure 1, the number to be translated is 3.123×10^{-6} with a scale factor of 2^{22}.

In all cases a minus sign represents a negative number and a zero or no punch a positive number.

Floating decimal numbers are presented to Rawoop in the same manner as decimal numbers. However, instead of a binary scale factor being placed in columns 28-30, an F is punched in column 28. The converted floating decimal number is in the form used in the Convair Flip. That is, the leftmost 28 bits is a mantissa with the binary point just to the right of the sign bit and the rightmost eight bits is a signed exponent.

Octal Constants

Octal constants can be inserted into the program using Rawoop. The octal constant is entered into columns 9-10, 12-16, and 18-22. A "B" is punched into column 24 to signify that the number is in octal (binary) form. This feature implies that programs coded in usual machine language will be correctly assembled.

Start Card

The signal to Rawoop that all the cards have been received is a card with START punched in columns 1-5, and a relative address punched in columns 12-16.

This signal causes Rawoop to insert into the translated program in locations 00000 and 40000 a manual jump to the octal translation of the relative address. Hence, after the resulting punched paper tape is read into the 1103, a magnetic drum start will start the problem at the relative address indicated in the u columns.

Subroutines

Rawoop is designed to translate subroutines coded in octal relative to 01000 and stored on the drum, and include them into the main program. This is effected by one psuedo command, the "SUB" command.

Columns 1-5 of the SUB command contain the relative address assigned to the zeroth word of the subroutine by one of the D cards. Columns 7, 8, and 9 have the letters SUB. Columns 18, 19, and 20 have the number of words in the subroutine. Columns 21 and 22 have the number of constants at the end of the subroutine which are not to be translated. Columns 12-16 have the location of where the zeroth word of the subroutine is stored on the drum (this address is the decimal drum address). In operation, the information in these columns is simply copied from a subroutine specification list.

A listing of the translated subroutine is usually provided by the output cards. The punching of cards for this however, can be suppressed by depressing the j=3 jump button.

Memory Sums

Rawoop calculates the memory sum of all translated words including the 00000 and 40000 jump instructions. This sum is the double precision sum of the split extension of the translated words. The sum is both printed on the listing and is read by the output paper tape into octal addresses 67,776 and 67,777 of the drum. The high order value of the sum is in 67,776.

Ordering of Input Cards

Rawoop makes the following requirements on the ordering of the input cards:

1. All directory cards must come first. The directory cards, however, can have any order within themselves.

2. The START card must come last.

3. The SUB cards, if any, must immediately precede the START card. However, the SUB cards can have any order within themselves.

The cards actually giving the words of the program follow the directory cards. For the sake of minimizing the number of insert and check addresses on the output bioctal tape; the cards should be in order within their regions. However, a correct output tape will result no matter what order these incoming cards are in. If the cards were out of order, a convenient listing still can be obtained by re-ordering the output cards.

Operating Instructions

The following points are important in the operation of Rawoop:

1. The reproducer must be set for fields I and II only.

2. The input cards are to be placed face down with at least six blank cards following the START card.

3. Two cards are to be fed into the punch channel.

4. Both the typewriter and the high-speed punch must be turned on.

5. After reading the assembly program into the machine, an MD start is sufficient to start or re-start Rawoop. Rawoop checks its own memory sum at the beginning of the problem and gives a signal if the check discloses an error.

6. All cards are cleared out of the read and write channels at the end of the program.

If it is desired to suppress the output subroutine cards, jump 3 should be depressed.

Error Detection

Rawoop will stop and signal the reason in case of either of the following two occurrences: More than sixty D cards are entered or any card other than either a SUB card or the START card follows a SUB card.

All other errors, including a D card occurring in the main deck will not stop the machine. For each card with an error on it, "ERROR" will be printed on the typewriter. The contents of the erroneous card will be ignored and will not affect the memory sum. The corresponding output cards will have "error" punched on them in the place ordinarily used for the translated information.

The detected errors can be corrected by a second assembly of only the now corrected input cards, their associated D cards, and the START card. This will give a secondary input type with its own memory sum.

An Example

The following pages contain the listing of an alarm routine coded at Ramo-Wooldridge using the Rawoop notation and subsequently translated by Rawoop.

Note that there are two output "start" cards though there was only one input "start" card. The first of these is to put an MJ into 00000; the second allows a MD start at 40000.

Symbolic word N04 is an example of the use of the zero, zero, zero region to get a decimal integer less than 100 without using the usual decimal number format.

D		00L00 49300		3 DIRECTORY	62124	00 00000 00000
D		00N00 49333		CARDS FOR	62165	00 00000 00000
D		00Z00 49345		ALR-0	62201	00 00000 00000
Z00		37	B	FLEX 0	62201	00 00000 00037
Z01		52	B	FLEX 1	62202	00 00000 00052
Z02		74	B	FLEX 2	62203	00 00000 00074
Z03		70	B	FLEX 3	62204	00 00000 00070
Z04		64	B	FLEX 4	62205	00 00000 00064
Z05		62	B	FLEX 5	62206	00 00000 00062
Z06		66	B	FLEX 6	62207	00 00000 00066
Z07		72	B	FLEX 7	62210	00 00000 00072
Z08		60	B	FLEX 8	62211	00 00000 00060
Z09		33	B	FLEX 9	62212	00 00000 00033
Z10		45	B	FLEX CAR RTN	62213	00 00000 00045
Z11		4	B	FLEX SPACE	62214	00 00000 00004
Z12		57	B	FLEX SHFT DN	62215	00 00000 00057
Z13		47	B	FLEX SHFT UP	62216	00 00000 00047
Z14		51	B	FLEX TAB	62217	00 00000 00051
Z15		42	B	FLEX PERIOD	62220	00 00000 00042
Z16		56	B	FLEX MINUS	62221	00 00000 00056
L00	MJ	00000 00000		EXIT	62124	45 00000 00000
L01	TP	A	N 6	STORE AR	62125	11 20000 62173
L02	LA	A	36		62126	54 20000 00044
L03	TP	A	N 5	STORE AL	62127	11 20000 62172
L04	TP	Q	N 7	STORE Q	62130	11 10000 62174
L05	TP	00000	N 8	STORE F1	62131	11 00000 62175
L06	TP	L	00000	SET MJ IN F1	62132	11 62124 00000
L07	PR	00000	Z10	PR CAR RETRN	62133	61 00000 62213
L08	PR	00000	Z12	PR SHIFT DN	62134	61 00000 62215

L09	TU	L18	L20	RESET ADDRES	62135	15	62146	62150
L10	TP	N 3	N 9 .	SET INDEX 1	62136	11	62170	62176
L11	TP	N 2	N10	SET INDEX 2	62137	11	62167	62177
L12	TP	N11 Q		TAG TO Q	62140	11	62200	10000
L13	LQ Q		6		62141	55	10000	00006
L14	PR 00000 Q			PR TAG CHAR	62142	61	00000	10000
L15	IJ	N 9	L13	INDEX 1	62143	41	62176	62141
L16	RP	5	L18		62144	75	00005	62146
L17	PR 00000		Z11	PR 5 SPACES	62145	61	00000	62214
L18	TP	N 4	N 9	SET INDEX 1	62146	11	62171	62176
L19	RA	L20	N 1	ADV U OF N1	62147	21	62150	62166
L20	SP 00000		36		62150	31	00000	00044
L21	SS A		3		62151	34	20000	00003
L22	SA	N	00000	PLUS DUMMY	62152	32	62165	00000
L23	TP A		L24	TO N1	62153	11	20000	62154
L24	00 00000 00000			PR DIGIT	62154	00	00000	00000
L25	IJ	N 9	L21	INDEX 1	62155	41	62176	62151
L26	IJ	N10	L16	INDEX 2	62156	41	62177	62144
L27	SP	N 5	36	RESTORE AL	62157	31	62172	00044
L28	SA	N 6	00000	RESTORE AR	62160	32	62173	00000
L29	TP	N 7 Q		RESTORE Q	62161	11	62174	10000
L30	TP	N 8	00000	RESTORE F1	62161	11	62175	00000
L31	PR 00000		Z10	PR CAR RETRN	62163	61	00000	62213
L32	MJ 00000		L	STOP	62164	45	00000	62124
N00	PR 00000 00Z00			DUMMY	62165	61	00000	62201
N01		1	B	ADV U	62166	00	00001	00000
N02			2 B	DEC 2	62167	00	00000	00002
N03			5 B	DEC 5	62170	00	00000	00005
N04			11	DEC 11	62171	00	00000	00013

N05	00 00000 00000 B	AL	62172	00	00000	00000
N06	00 00000 00000 B	AR	62173	00	00000	00000
N07	00 00000 00000 B	Q	62174	00	00000	00000
N08	00 00000 00000 B	F1	62175	00	00000	00000
N09	00 00000 00000 B	INDEX 1	62176	00	00000	00000
N10	00 00000 000 0 B	INDEX 2	62177	00	00000	00000
N11	00 00000 00000 B	TAG	62200	00	00000	00000
START	00L01		00000	45	00000	62125
START	00L01		40000	45	00000	62125

DIVISION San Diego

MODEL All

REPORT ZM-490

DATE 8/4/54
revised 12/14/54

TITLE

FLIP

A Floating Point Subroutine System

for

The ERA 1103 Computer

PREPARED BY Charles J. Swift

GROUP Digital Computing Lab

REFERENCE

CHECKED BY Members of Digital Computing Laboratory

APPROVED BY

NO OF PAGES 57

NO OF DIAGRAMS

REVISIONS

NO	DATE	BY	CHANGE	PAGES AFFECTED

GD 241A

Notes on FLIP

1. Commands 40 and 44 now leave (R) unchanged.

2. (v) for command 42 may not be zero, although the exponent part (v_7 to v_0) may be zero.

3. In addition to cells 01477 to 01777 and those assigned to subroutine commands, Flip also occupies cells 00001 (entrance), 00040 to 00077 (constants) and 76000 to 76052 (alarm).

4. The 07 Command can be used to terminate a series. If

$$(x) = (R)$$

and $\left|\frac{(y)}{(x)}\right| < 2^{-27}$

this command will leave (A) = 0. An example of its use is this:-

01000	14	00	1400	1401	$x + \delta x \rightarrow x, R$ (add term to series)
01001	14	07	1401	1400	$[\delta x - x] + x \rightarrow R$ Did this term
01002	47	00500	01003		$\left|\frac{\delta x}{x}\right| : 2^{-27}$ alter the sum?

(1400) δx

(1401) x

A Floating Point Subroutine System Using A Packed Representation for the ERA 1103 Computer

(Flip) Revised 10/15/54

The system described here is intended to be indefinitely expanded to include card operations, transcendental functions etc. The interpreted instructions are 10 octal digits long (preceded by octal 14, the IP command). The basic system occupies ES cells 01477 to 01777 and 00001. If any subroutine operations are used, more space is needed. The whole system is located on the drum in locations 76000 to 77777. The basic operations in FLIP use temporaries 00002 to 00014. Subroutine operations use cells 00002 to 00037.

I Composition of Instructions

Instruction Code	6 bits	(i_{29} ... i_{24})
First Address, x	12 bits	(i_{23} ... i_{12})
Second Address, y	12 bits	(i_{11} ... i_0)

II Addresses

The last 10 bits of each address form the "basic" address. The first two bits, if ones, cause either or both of the two special counters b_1 and b_2 to be added to this basic address to form the "execution" address. The first bit causes b_1 to be added, the second bit causes b_2 to be added. The instruction is executed using this execution address which must not exceed 01777. All execution addresses refer to ES.

III Special Registers

In addition to R and Q, two other 36 bit registers, b_1 and b_2 are specially referred to in orders. In order to refer to R, Q, b_1 and b_2, in these orders, use addresses 1774, 1775, 1776 and 1777 respectively. The contents of 1774 and 1775 after any FLIP operation are respectively equal to (R) and (Q) preceding that operation. If a result is placed in both y and R, ($\mathcal{A}$) is not a double extension of (R); otherwise it is.

IV Contents of Registers

Except for b_1 and b_2, the contents of every register is interpreted as a number of the form $q \times 2^p$. q occupies the first 28 bits, p the last 8. Negative p's and q's are expressed by complements, the first digit of both p and q being a sign bit. The binary point of q follows the sign bit. Except for the special case $p = q = 0$, q is restricted to the values $1 > q \geq \frac{1}{2}$ and p is restricted to the range $|p| < 2^7$. For $p \geq 2^7$ an alarm print occurs. For $p \leq -2^7$ both p and q are set equal to zero.

V Loading

A loader routine is included in the FLIP system to transfer FLIP and such of its subroutines as are used, into ES. It uses ES cells 00000 to 00006, 00040 to 00121 and 01477 to 01777. FLIP operations are of two types, basic code operations and subroutine operations. The coder must specify which of the latter he is using and the memory locations of the subroutines. These are specified by parameter words in consecutive cells starting in cell 00122. The first two octal digits of each such word form the operation code. The last five octal digits form the address of the first cell to be occupied by the subroutine. A zero cell terminates the parameters.

The loader is entered at cell 76575 (or 40,000) and halts on completion. It then exits to cell 00010. The constant pool and cells 00000, and 00001 are taken care of by the loader. The use of any command codes not specified to the loader (except basic commands) or the use of command codes still unassigned will cause an alarm halt.

FLIP (revised 10/15/54)

VI Alarm Prints

If an out-of-range or non-existent result occurs, FLIP prints out ALARM aaaaa xxxxxxxxxxxx yyyyyyyyyyyy, where aaaaa is the main routine address,

xxxxxxxxxxxx is (x) in octal ,

yyyyyyyyyyyy is (y) in octal .

The 1103 will halt after the typeout; hit the start button to continue the main routine.

VII Operations

For most arithmetic operations the operation bits are divided into 4 groups; $a = i_{29}$, $b = i_{25}, i_{24}$, $c = i_{26}$, and $d = i_{28}, i_{27}$. These function as follows:

a	0	An arithmetic operation
	1	A logical operation
b	00	Use (y), (x) as operands and store the result in (R) and (y).
	01	Use (R), (x) as operands and store the result in (R) and (y).
	10	Use (y), (x) as operands and store the result in (R) only.
	11	Use (y), x) as operands, add the result to (R) and stor the result in (R) only.
c	0	Use (x) without change in the operation.
	1	Use -(x) in the operation instead of (x).
d	00	Perform the add operation.
	01	Perform a special polynomial order. (see list following)
	10	Perform the multiply operation
	11	Perform the divide operation.

(FLIP) (revised 10/15/54)

Code	Symbolic Operation	Name
00	y + x → y, R	Replace Add
01	R + x → y, R	Add and transmit
02	y + x → R	Add
03	R + y + x → R	Accumulate Add
04	y - x → y, R	Replace subtract
05	R - x → y, R	Subtract and transmit
06	y - x → R	Subtrast
07	R + y - x → R	Accumulate add and subtract
10	y + R · x → y, R	Replace add a product
11	(1 + x) · R → y, R	Increment and Multiply
12 }	y + R · x → R	Positive Polynomial
13 }		
14	y - R · x → Y, R	Replace add a negative product
15	(1 - x) · R → y, R	Decrement and Negative Multiply
16 }	y - R · x → R	Alternating Polynomial
17 }		
20	y · x → y, R	Replace multiply
21	R · x → y, R	Multiply and Transmit
22	y · x → R	Multiply
23	R + y · x → R	Accumulate Multiply
24	-y · x → y, R	Negative replace Multiply
25	-R · x → y, R	Negative multiply and transmit
26	-y · x → R	Negative Multiply
27	R - y · x → R	Accumulate negative Multiply
30	y ÷ x → y, R	Replace Divide

(FLIP) (revised 12/9/54)

31	$R \div x \rightarrow y, R$	Divide and transmit
32	$y \div x \rightarrow R$	Divide
33	$R + (y \div x) \rightarrow R$	Accumulate Divide
34	$-y \div x \rightarrow y, R$	Negative replace Divide
35	$-R \div x \rightarrow y, R$	Negative Divide and transmit
36	$-y \div x \rightarrow R$	Negative Divide
37	$R - (y \div x) \rightarrow R$	Accumulate negative divide

40 Counter b_1 jump
44 Counter b_2 jump

If (counter b_n) $<$ (x), add one to counter b_n and jump to y. If not, set counter b_n to zero and continue present sequence. (R) is unchanged.

Note: The contents of the counters are <u>not</u> floating numbers.

41 Absolute Value Threshold Jump If $|(x)| > |(R)|$, jump to y, otherwise continue present sequence. (R) is unchanged by operation. If x = -R a jump may occur.

45 Reverse Absolute Value Threshold Jump

If $|(x)| \leq |(R)|$ jump to y, otherwise continue present sequence. (R) is unchanged by operation. If x = -R a jump may not occur.

42 Fixed to Floating Call the last 8 bits of (y) <u>P</u> and let (x) (35) = q, where $|q| < 1$. Form the correctly normalized and packed representation of $q \times 2^P$ in R.

Note: (y) may <u>not</u> be <u>zero.</u>

51 Floating to Fixed Take q' and p' from the floating number

(FLIP) (revised 12/9/54)

51 (cont) Floating to Fixed

in R. Take p from the last 8 bits of (x). Store q (35) in y and R where $q = q' \times 2^{p'-p}$

B. Subroutine Operations
(see separate pages for details)

40 (basic code)

41 (basic code)

42 (basic code)

43 Convert Flexowriter input data.

44 (basic code)

45 (basic code)

50 Square Root

51 (basic code)

52 Print on Flexowriter

53 Punch Flexowriter Tape

54 Log_e N

56 Punch Cards

57 Read Cards

60 Sine

61 Cosine

62 Tan^{-1} x

63 Cotan^{-1} x

70 e^x

77 Trace

FLIP (revised 12/9/54)

Subroutine Specifications

FLIP subroutines are coded to start in cell 01000 and are modified by the assembly routine under control of the FLIP leader routine. Each subroutine is assigned a command code number.

I Command Code Parameter

If the subroutine is assigned command number OP (octal), a parameter is placed in drum location 76555 + 2 · OP. It has the form:-

m_1	D.A.	n	m_2
xx	xxxxx	xxx	0 x

where D.A. = Drum address of the subroutine

$64m_2 + m_1$ = number of cells occuppied

n = number of cells modified

II Input Information

At the entry to the subroutine, the temporaries contain this information:

Second Octal Digit of Command Code

	0	1	2	3	4	5	6	7
00005	(x)	(x)	(x)	(x)	-(x)	-(x)	-(x)	-(x)
00006	(y)	(R)	(y)	(y)	(y)	(R)	(y)	(y)
00007	0	0	0	(R)	0	0	0	(R)
ϕ (01734)	y	y	20000	20000	y	y	20000	20000

00004 Command code in 1st two octal digits.

00010 36 bit extension of exponent from (00005)

00011 36 bit extension of exponent from (00006)

00013 Execution addresses x and y in the u and v positions.

R 36 bit extension of exponent from (00005)

10000 Command code in last two octal digits.

Notes: If any argument was zero, it has been transformed to 00000000200 (octal) with the exponent 777777777600 (octal).

FLIP (revised 12/9/54)

III Exit Information

Subroutine exits are to 01607, 01721, 01734, 01735 or 76045.

a. Exit at 01607. Place zero in A and go to 01734 (see below)

b. Exit at 01721. Normalize, test, pack and store. In this case the subroutine should leave:

q (35) in 00005

p (0) in 00010

Where the desired result is q · 2^p. q and p may be any possible 36 bit numbers. The routine will normalize, test, pack into R, and go to 01734.

c. Exit at 01734, 01735 or 01737

01734	11 20000 [00000]	STORE RESULT	(see table above)
01735	45 10000 [01736]	TRACE?	(exits to trace if used)
01736	45 00000 [00000]	Exit	
01737	37 00000 76045	ALARM EXIT	

Notes:

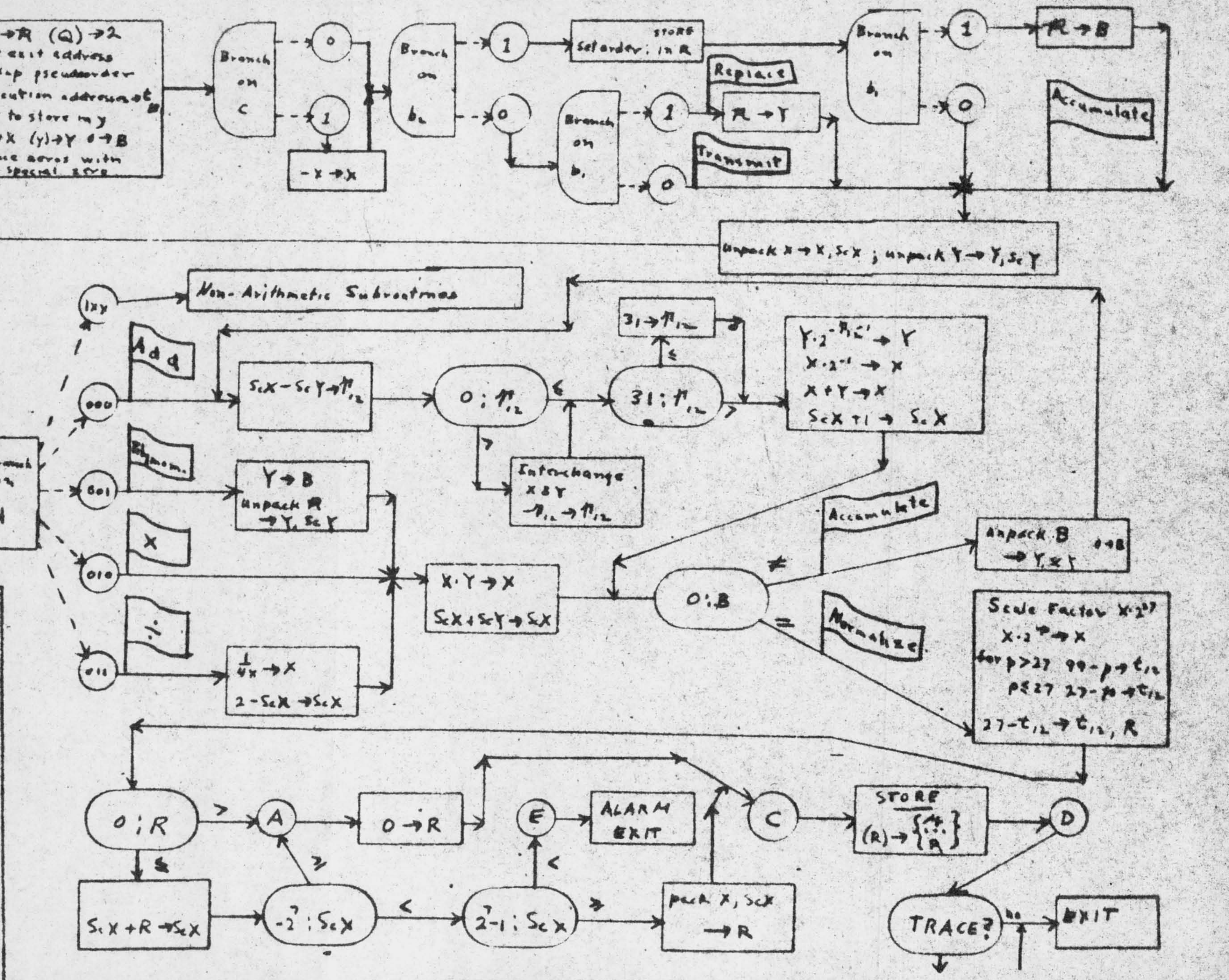
1. All references to exits above must be in unmodified orders.
2. Double duty subroutines (e.g. sine and cosine) are assigned two command codes and treated as two overlapping subroutines.

IV "Own Code" Subroutines

Frequently occurring subroutines can be coded as FLIP subroutines. Before using other FLIP commands in such subroutines, it is necessary to transfer ø (1736) and such other information as must be saved to new locations. FLIP basic orders do not use cells 00015 to 00037 or cell 00002. If the subroutine needs no modification or change of location, use 00 00001 00000 for its command code parameter.

ANALYSIS
PREPARED BY C. J. Swift
CHECKED BY
REVISED BY

FLOW CHART ARITHMETIC OPERATIONS



FLIP (revised 12/9/54)

FLIP STORAGE LOCATIONS

Subroutine Code Number	Name	Number of Words	Drum Address	HS Address
	Constants, Assembly Routine	72	77716	40
	Basic Code	193	76755	1477
	Subroutine Parameters	60	76661	1704
	Loader	61	76575	01537
50	Square Root	17	76553	(01000)
77	Trace	26	76521	(01000)
54	Log_e n	31	76462	(01000)
53	PUNCH	56	77262	(01000)
52	PRINT	2	77260	(01000)
	ALARM	42	76000	(76000)
43	Flexowriter Input Conversion	58	77352	(01000)
60	Sine	50	77454	(01000)
61	Cosine	2	77452	(01000)
70	e^x	18	76360	(01000)
	Read in Test	36	76414	76414
	Process Trace	59	77600	00117
62	Tan^{-1}	54	76300	(01000)
63	$Cotan^{-1}$	2	76356	(01000)

Loader

76575	01537	11	76655	00000	Preset F_1
76576	01540	75	30061	01615	Transfer loader to ES
76577	01541	11	76600	01542	
76600	01542	75	30062	01544	Transfer Constants and Assembly to ES
76601	01543	11	77716	00040	
76602	01544	11	00122	00003	Requisition parameter P_i
76603	01545	11	00003	00004	$P_i \rightarrow t_4$
76604	01546	31	00004	00017	ϕ $(P_i) \rightarrow \Theta$ (t_5)
76605	01547	11	20000	00005	
76606	01550	55	00003	00026	$p \cdot OP \rightarrow \Theta$ (t_3)
76607	01551	11	01621	10000	Mask $\longrightarrow$ q
76610	01552	53	00003	01554	Set order: "requisition subroutine parameters
76611	01553	75	30002	01555	Requisition Subroutine Parametes P_1, P_2
76612	01554	11	01[600]	00006	
76613	01555	15	00006	01576	Set order: "Transfer subroutine to ES"
76614	01556	16	00004	01576	
76615	01557	55	00006	00011	Place parameter for assembly routine
76616	01560	31	00005	00071	
76617	01561	52	01621	00122	
76620	01562	47	01563	01601	Is this the termination flag?
76621	01563	55	00006	00014	Set repeat order: "Transfer subroutine to ES"
76622	01564	11	01577	20000	
76623	01565	52	01621	01575	
76624	01566	16	00007	01571	Set subroutine references in Basic Code
76625	01567	55	00007	00025	
76626	01570	16	10000	01572	
76627	01571	16	00004	[30000]	

76630	01572	15 00005 [30000]	}
76631	01573	21 01544 00073	Step to next subroutine
76632	01574	21 01561 00074	
76633	01575	[77 77777 77777]	Transfer Subroutine to ES
76634	01576	11 [00000 00000]	
76635	01577	75 30000 01544	Prototype order for 01575
76636	01600	00 00000 00000	"Parameter for zero subroutine"
76637	01601	11 01620 01774	Preset exit of Loader
76640	01602	11 01617 00001	Preset F_2
76641	01603	11 01622 00002	obsolete order
76642	01604	37 00101 00100	Modify subroutines
76643	01605	75 10003 01607	Clear FLIP temporaries
76644	01606	11 00040 01775	
76645	01607	75 30275 01774	Transfer in Basic FLIP
76646	01610	11 76755 01477	
76647	01611	75 30222 01776	Obsolete
76650	01612	11 77027 01551	
76651	01613	75 30236 01776	
76652	01614	11 77014 01536	
76653	01615	75 30074 01542	Transfer Subroutine parameters to ES
76654	01616	11 76661 01704	
76655	01617	45 00000 01624	(F_2)
76656	01620	56 00000 00010	Exit order for Loader
76657	01621	00 00177 00000	Extractor
76660	01622	45 00000 76761	obsolete order

COMMAND CODE PARAMETERS

76661	01704	01	01731	00000	42
76662	01705	00	00003	00003	
76663	01706	72	77352	05000	43
76664	01707	00	77023	00003	
76665	01710	01	01731	00000	44
76666	01711	00	00003	00003	
76667	01712	01	01731	00000	45
76670	01713	00	00003	00003	
76671	01714	01	01731	00000	46
76672	01715	00	00003	77007	
76673	01716	01	01731	00000	47
76674	01717	00	77007	00003	
76675	01720	21	76553	02000	50
76676	01721	00	00003	77026	
76677	01722	24	76360	00500	51
76700	01723	00	00003	00003	
76701	01724	02	77450	00200	52
76702	01725	00	00003	77011	
76703	01726	70	77262	03400	53
76704	01727	00	77011	00003	
76705	01730	37	76462	02400	54
76706	01731	00	00003	77012	
76707	01732	01	01731	00000	55
76710	01733	00	77012	00003	
76711	01734	01	01731	00000	56
76712	01735	00	00003	77013	
76713	01736	01	01731	00000	57
76714	01737	00	77013	00003	
76715	01740	61	77454	04700	60

76716	01741	00	00003	76777	
76717	01742	02	77452	00200	61
76720	01743	00	76777	00003	
76721	01744	55	76300	03600	62
76722	01745	00	00003	77000	
76723	01746	02	76356	00200	63
76724	01747	00	77000	00003	
76725	01750	01	01731	00000	64
76726	01751	00	00003	77001	
76727	01752	01	01731	00000	65
76730	01753	00	77001	00003	
76731	01754	01	01731	00000	66
76732	01755	00	00003	77002	
76733	01756	01	01731	00000	67
76734	01757	00	77002	00003	
76735	01760	31	76360	01200	70
76736	01761	00	00003	77003	
76737	01762	01	01731	00000	71
76740	01763	00	77003	00003	
76741	01764	01	01731	00000	72
76742	01765	00	00003	77004	
76743	01766	01	01731	00000	73
76744	01767	00	77004	00003	
76745	01770	01	01731	00000	74
76746	01771	00	00003	77005	
76747	01772	01	01731	00000	75
76750	01773	00	77005	00003	
76751	01774	01	01731	00000	76
76752	01775	00	00003	77006	
76753	01776	32	76521	03100	77
76754	01777	00	00003	[illegible]	

Basic Code

76755	01477	27	00005	01772	-X → X
76756	01500	44	01501	01505	Branch on b_2
76757	01501	16	01712	01734	Set to store in R
76760	01502	44	01503	01666	Branch on b_1
76761	01503	11	01774	00007	R → B
76762	01504	45	00000	01666	jump
76763	01505	44	01665	01666	Branch on b_1
76764	01506	11	01511	20000	27 → R
76765	01507	36	00012	00012	p → R (if p > 27; p-72 → R)
76766	01510	46	01607	01726	0:R
76767	01511	00	00000	00033	27
76770	01512	44	01514	01513	SORTING
76771	01513	44	01516	01515	WHIFFLETREE
76772	01514	44	01520	01517	for NON-ARITHMETIC
76773	01515	44	01522	01521	ORDERS
76774	01516	44	01524	01523	
76775	01517	44	01526	01525	
76776	01520	44	01530	01527	
76777	01521	44	01737	01737	
77000	01522	44	01737	01737	
77001	01523	44	01737	01737	
77002	01524	44	01737	01737	
77003	01525	44	01737	01737	
77004	01526	44	01737	01737	
77005	01527	44	01737	01737	
77006	01530	44	01737	01737	
77007	01531	44	01737	01737	
77010	01532	44	01535	01534	

77011	01533	44	01737	01737	
77012	01534	44	01737	01737	
77013	01535	44	01737	01737	
77014	01536	44	01512	01537	
77015	01537	44	01546	01540	
77016	01540	44	01542	01541	
77017	01541	44	01545	01543	
77020	01542	44	01531	01544	
77021	01543	44	01571	01553	
77022	01544	44	01574	01560	
77023	01545	44	01737	01551	
77024	01546	44	01532	01547	
77025	01547	44	01533	01550	
77026	01550	44	01605	01737	
77027	01551	11	00011	00010	scY → scX 42 command
77030	01552	45	00000	01721	Jump to Normalize
77031	01553	21	01776	00074	b_1-1 → b_1,R 40 command
77032	01554	36	00005	20000	b_1-(x) → R
77033	01555	42	00074	01602	1 : b_1-(x) jump to "jump"
77034	01556	11	00040	01776	0 → b_1
77035	01557	45	00000	01603	jump to "no jump"
77036	01560	27	00005	01772	-x → x 44 command
77037	01561	21	01777	00074	b_2-1 → b_2,R
77040	01562	36	00005	20000	b_2-(x) → R
77041	01563	42	00074	01602	1: b_2-(x) jump to "jump"
77042	01564	11	00040	01777	0 → b_2
77043	01565	45	00000	01603	jump to "no jump"
77044	01566	16	01734	01736	} empty
77045	01567	45	00000	01735	} empty
77046	01570	45	00000	01735	} empty

77047	01571	16	01734	00012	Interchange y and Exit — 41 command
77050	01572	16	01736	01734	
77051	01573	16	00012	01736	
77052	01574	23	20000	00011	scX-scY → R — 45 command
77053	01575	46	01602	01576	If scY > scX, jump to 1602
77054	01576	47	01603	01577	If scX > scY, jump to 1603
77055	01577	12	00005	00005	|X| → X
77056	01600	12	00006	20000	|Y| → R
77057	01601	42	00005	01603	If |X| > |Y| jump to 1603
77060	01602	16	01734	01736	Change exit
77061	01603	11	01774	20000	R → R
77062	01604	45	00000	01735	Exit
77063	01605	23	20000	00011	p - p' → R — 51 command
77064	01606	42	00056	01611	jump to 1611
77065	01607	11	00040	20000	Clear answer
77066	01610	45	00000	01734	Exit
77067	01611	46	01737	01612	If p - p' < 0, jump to ALARM
77070	01612	13	20000	20000	p' - p → R
77071	01613	35	01773	01614	Set order: $o \cdot 2^{p-p'} \rightarrow q$
77072	01614	54	00006	00110	$q \cdot 2^{p-p'} \rightarrow q$
77073	01615	45	00000	01734	Exit
77074	01616	11	00006	00007	Y → B — "p" orders
77075	01617	31	01774	00034	unpack (R) → Y, scY
77076	01620	11	20000	00011	
77077	01621	54	00011	00054	
77100	01622	11	01774	00006	
77101	01623	45	00000	01744	Jump to Multiply
77102	01624	16	00000	01736	Store exit — ENTER
77103	01625	11	20000	01774	Store R → R
77104	01626	11	10000	01775	Store Q → Q

77105	01627	31 00000 00017	
77106	01630	35 01766 01631	Instruction → q
77107	01631	11 [00000] 10000	OP → t_4; b_1 indicators → t_5;
77110	01632	75 30004 01634	b_2 indicators → t_6; Addresses → t_7
77111	01633	51 01761 00004	
77112	01634	31 00007 00012	
77113	01635	72 00005 01777	Modify Basic Addresses
77114	01636	54 20000 00001	
77115	01637	72 00006 01776	
77116	01640	73 01765 10000	
77117	01641	16 10000 01652	
77120	01642	55 10000 00017	
77121	01643	11 10000 00013	Put Execution Addresses into t_{13}
77122	01644	54 20000 00075	Preset Pickup and Store orders
77123	01645	16 20000 01734	
77124	01646	16 20000 00013	
77125	01647	16 20000 01655	
77126	01650	75 10003 01652	0 → X, Y and B
77127	01651	11 00040 00005	
77130	01652	[21 00005 00000]	Pickup (x) → X
77131	01653	47 01655 01654	0:x
77132	01654	11 01770 00005	2^{-155} → X
77133	01655	[21 00006 00000]	Pickup (y) → Y
77134	01656	47 01660 01657	0:y
77135	01657	11 01770 00006	2^{-155} → Y
77136	01660	11 01774 20000	
77137	01661	47 01663 01662	0: R
77140	01662	11 01770 01774	2^{-155} → R
77141	01663	55 00004 00011	Branch on C

77142	01664	44	01477	01500	Branch on C
77143	01665	11	01774	00006	R → Y
77144	01666	31	00006	00034	} Unpack Y → Y, scY
77145	01667	11	20000	00011	
77146	01670	54	00011	00054	
77147	01671	31	00005	00034	} Unpack X → X, scX and R
77150	01672	11	20000	00010	
77151	01673	54	00010	00054	
77152	01674	55	00004	00041	} Branch on a and d
77153	01675	44	01536	01676	
77154	01676	44	01677	01700	
77155	01677	44	01740	01744	
77156	01700	44	01616	01701	
77157	01701	23	20000	00011	scX - scY → T_1, R Add
77160	01702	46	01703	01710	0: T
77161	01703	11	00005	00014	} Interchange X and Y; scY → scX ; -R → R
77162	01704	11	00006	00005	
77163	01705	11	00014	00006	
77164	01706	11	00011	00010	
77165	01707	13	20000	20000	
77166	01710	42	00045	01712	If 31 < R jump to 1712
77167	01711	11	00045	20000	31 → R
77170	01712	13	20000	20000	-R → R
77171	01713	35	01767	01714	Set next order
77172	01714	54	00006	00107	Y · $2^{-1-T_{12}}$ → Y
77173	01715	54	00005	00107	X · 2^{-1} → X
77174	01716	35	00006	00005	X + Y → X
77175	01717	21	00010	00074	scX + 1 → scX
77176	01720	45	00000	01750	Exit
77177	01721	11	00040	00012	0 → t_{12} Normalize

12/1/54

77200	01722	54 00005 00032	$2^{27}X \to A$
77201	01723	74 20000 00012	$X \cdot 2^p \to R$; $27 - p \to t_{12}$
77202	01724	11 20000 00005	store X
77203	01725	45 00000 01506	jump
77204	01726	23 00010 00012	scX - p → scX
77205	01727	42 01771 01607	
77206	01730	42 01770 01732	Check size of scX
77207	01731	45 00000 01737	
77210	01732	11 01772 10000	Pack answer in R
77211	01733	53 00005 00010	
77212	01734	11 20000 [00000]	Store result
77213	01735	45 10000 [01736]	Automonitor?
77214	01736	45 00000 [00000]	Exit
77215	01737	45 00000 76045	Alarm Exit
77216	01740	11 00041 20000	2-scx → scx (Divide)
77217	01741	36 00010 00010	
77220	01742	31 00066 00042	$\frac{1}{4x} \to x$
77221	01743	73 00005 00005	
77222	01744	71 00006 00005	Multiply
77223	01745	54 20000 00045	X · Y → X
77224	01746	11 20000 00005	
77225	01747	21 00010 00011	scX + scY → scX
77226	01750	11 00007 20000	If B = 0 jump to 1721
77227	01751	47 01752 01721	
77230	01752	11 20000 00006	
77231	01753	31 20000 00034	Unpack B → Y, scY
77232	01754	11 20000 00011	
77233	01755	54 00011 00054	
77234	01756	11 00010 20000	scX → R

77235	01757	11	00040	00007	0 $\rightarrow$ B
77236	01760	45	00000	01701	Jump to ADD
77237	01761	00	77000	00000	}
77240	01762	00	00200	02000	}
77241	01763	00	00400	04000	} Extractors
77242	01764	00	00177	71777	}
77243	01765	00	00400	00000	}
77244	01766	10	77777	10000	Prototype order for 01631
77245	01767	54	00006	00107	Prototype order for 01714
77246	01770	00	00000	00200	2^7
77247	01771	77	77777	77600	$1 - 2^7$
77250	01772	77	77777	77400	$(2^8 - 1)$
77251	01773	54	00006	00110	Prototype order for 01614

Alarm Print *

76000	76000	45	00000	30000
76001	76001	45	00000	76024
76002	76002	37	76030	76025
76003	76003	31	76000	00017
76004	76004	15	20000	76005
76005	76005	11	30000	76000
76006	76006	11	76043	00006
76007	76007	55	76042	00006
76010	76010	61	00000	76042
76011	76011	41	00006	76007
76012	76012	13	00074	20000
76013	76013	32	76000	00025
76014	76014	11	20000	10000
76015	76015	11	00044	00006
76016	76016	37	76041	76033
76017	76017	11	00005	10000
76020	76020	37	76041	76032
76021	76021	11	00004	10000
76022	76022	37	76041	76032
76023	76023	56	00000	76000
76024	76024	16	76031	76030
76025	76025	11	20000	00004
76026	76026	54	20000	00044
76027	76027	11	20000	00005
76030	76030	45	00000	30000
76031	76031	00	00000	76006
76032	76032	11	00057	00006
76033	76033	61	00000	00044
76034	76034	55	10000	00003
76035	76035	51	00067	20000
76036	76036	35	00042	76037
76037	76037	00	00000	00000
76040	76040	41	00006	76034
76041	76041	45	00000	30000
76042	76042	45	30113	01207
76043	76043	00	00000	00005
76044	76044	00	00000	00000

FLIP ALARM ADDITION

76045	76045	15	00013	76047	Set Orders
76046	76046	16	00013	76050	
76047	76047	31	[30000]	00044	(x) → L
76050	76050	27	20000	[30000]	(y) → R
76051	76051	16	01736	76000	Enter Alarm
76052	76052	45	00000	76001	

*For explanation of orders see Report ZM-491

INPUT SUM CHECK for FLIP

76414	76414	57 00000 00000	Halt
76415	76415	[00 00000 00000]	prestored sum
76416	76416	11 40000 00000	set F_1
76417	76417	23 10000 10000	clear A,Q
76420	76420	75 22000 76422	SUM
76421	76421	32 76000 00000	
76422	76422	36 76415 10000	Check sum
76423	76423	23 10000 76415	
76424	76424	47 76427 76425	
76425	76425	75 10012 76431	Print "FLIP OK"
76426	76426	61 00000 76434	
76427	76427	75 10012 76414	Print "NO GO"
76430	76430	61 00000 76446	
76431	76431	16 76433 40000	set order
76432	76432	56 00000 40000	Halt
76433	76433	00 00000 76575	address
76434	76434	00 00000 00045	FLEXOWRITER
76435	76435	00 00000 00047	CHARACTERS ↓
76436	76436	00 00000 00026	
76437	76437	00 00000 00011	
76440	76440	00 00000 00014	
76441	76441	00 00000 00015	
76442	76442	00 00000 00004	
76443	76443	00 00000 00003	
76444	76444	00 00000 00036	

76445	76445	00	00000	00057	
76446	76446	00	00000	00045	
76447	76447	00	00000	00047	
76450	76450	00	00000	00002	
76451	76451	00	00000	00006	
76452	76452	00	00000	00003	
76453	76453	00	00000	00004	
76454	76454	00	00000	00013	
76455	76455	00	00000	00003	
76456	76456	00	00000	00002	
76457	76457	00	00000	00057	
40000	40000	45	00000	76416	ENTRANCE

CONSTANTS and ASSEMBLY ROUTINE *

77716	00040	00	00000	00000
77717	00041	00	00000	00002
77720	00042	61	00000	00045
77721	00043	00	00000	00003
77722	00044	00	00000	00004
77723	00045	00	00000	00037
77724	00046	00	00000	00052
77725	00047	00	00000	00074
77726	00050	00	00000	00070
77727	00051	00	00000	00064
77730	00052	00	00000	00062
77731	00053	00	00000	00066
77732	00054	00	00000	00072
77733	00055	00	00000	00060
77734	00056	00	00000	00033
77735	00057	00	00000	00013
77736	00060	00	00000	00012
77737	00061	00	00000	00056
77740	00062	31	10375	52421
77741	00063	31	46314	63146
77742	00064	00	00000	00077
77743	00065	21	67643	24177
77744	00066	20	00000	00000
77745	00067	00	00000	00007
77746	00070	37	77777	77777
77747	00071	00	77777	00000
77750	00072	00	00000	77777
77751	00073	00	00001	00000
77752	00074	00	00000	00001
77753	00075	00	00001	00001
77754	00076	00	07777	07777
77755	00077	00	00000	00110
77756	00100	11	00122	20000
77757	00101	47	00102	00000
77760	00102	73	00073	00005
77761	00103	16	20000	00112
77762	00104	36	00121	00006
77763	00105	41	00005	00110
77764	00106	21	00100	00073
77765	00107	45	00000	00100
77766	00110	16	00112	00116
77767	00111	11	00040	00007
77770	00112	21	00007	00000
77771	00113	55	20000	00033
77772	00114	51	00075	10000
77773	00115	71	10000	00006
77774	00116	35	00007	00000
77775	00117	21	00112	00074
77776	00120	45	00000	00105
77777	00121	00	00000	01000

*For explanation of orders see Report ZM-491

FLIP

Flexowriter Input Conversion Routine

Command Code: 43

Number of Cells: 58

I Description

This command converts a two-word input representation of a number to the normalized FLIP form and stores the result in y, but not in R. The two words are (x) and (x + 1). When taken 6 bits at a time they are 12 flexowriter character codes. If the number represented is: -

$$N = q' \cdot 10^{p'};\ 0.1 \leq q' < 1$$

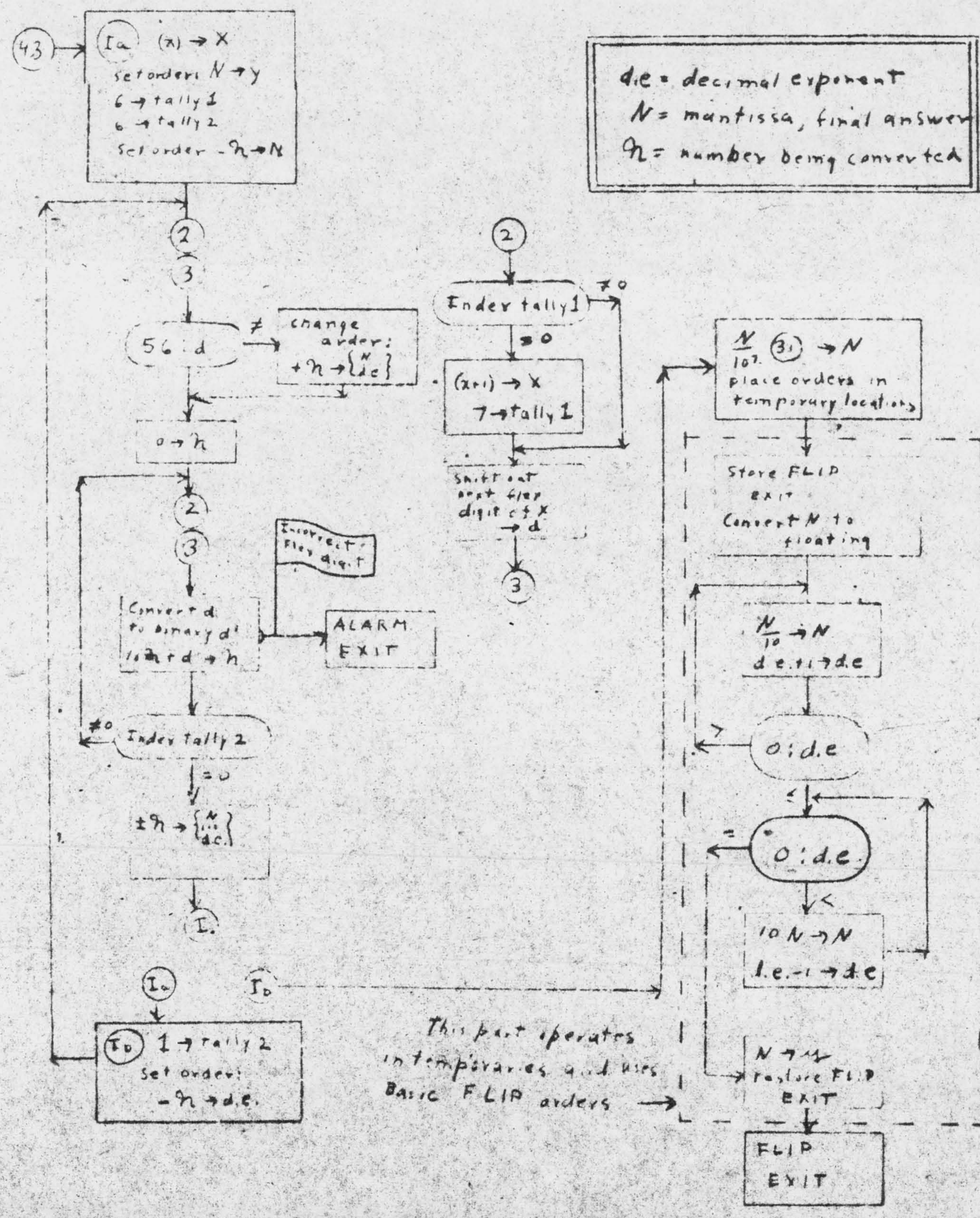
these flexowriter characters are: -

1	2	3	4	5	6	7	8	9	10	11	12
±	q	q	q	q	q	q	q	±	p	p	

The decimal point (not present) precedes character number 2. The sign positions are considered plus unless occupied by a minus sign (octal 56). The twelfth digit is immaterial and will usually be a carriage return.

If the number represented exceeds 2^{127} in absolute magnitude, or if any digit q or p is not a flexowriter numerical digit, an alarm halt will occur.

II Flow Chart



43 INPUT CONVERSION SUBROUTINE

77352	01000	16 01042 01025	set switch Ia
77353	01001	15 00013 01003	set orders
77354	01002	16 00013 01064	
77355	01003	11 30000 00004	$(x) \rightarrow x$
77356	01004	11 01051 00012	$6 \rightarrow$ tally 2
77357	01005	11 01051 00003	$6 \rightarrow$ tally 1
77360	01006	11 01046 01024	set order ł $- h \rightarrow N$
77361	01007	37 01041 01032	1st FLEX digit (sign)
77362	01010	43 00061 01012	Test for minus sign
77363	01011	23 01024 01050	charge order: $+h \rightarrow \left\{ \begin{matrix} N \\ --- \\ d.c. \end{matrix} \right\}$
77364	01012	23 00005 00005	$0 \rightarrow A, h$
77365	01013	37 01041 01032	one digit $\rightarrow A$
77366	01014	75 20013 76045	binary digit $\rightarrow d'$
77367	01015	43 00045 01016	
77370	01016	51 00064 00006	
77371	01017	11 00060 00007	
77372	01020	23 00007 00006	
77373	01021	71 00060 00005	$10\, h + d' \rightarrow h$
77374	01022	35 00007 00005	
77375	01023	41 00003 01013	Index tally 2
77376	01024	[13 00005 30000]	$\pm h \rightarrow \left\{ \begin{matrix} N \\ d.c. \end{matrix} \right\}$
77377	01025	37 01025 30000	I_b
77400	01026	54 00032 00037	$N \cdot 2^{31} \rightarrow A$
77401	01027	73 01070 00032	$N \cdot 2^{31} \div 10^7 \rightarrow N$
77402	01030	75 30016 00014	enter temporary orders
77403	01031	11 01052 00014	
77404	01032	41 00012 01037	Index tally 2 — Digit Subroutine
77405	01033	11 01003 20000	$(x+1) \rightarrow x$
77406	01034	35 00073 01035	
77407	01035	11 30000 00004	

77410	01036	11	00067	00012	7 → tally 2
77411	01037	55	00004	00006	shift one flex digit
77412	01040	51	00064	20000	x tract to A
77413	01041	45	00000	30000	exit
77414	01042	00	00000	01043	constant
77415	01043	11	00074	00003	1 → tally 1
77416	01044	11	01047	01024	set order - n → d.e.
77417	01045	45	00000	01007	jump
77420	01046	13	00005	00032	constants
77421	01047	13	00005	00033	
77422	01050	02	00000	00000	
77423	01051	00	00000	00006	
77424	00014	16	01736	00002	store exit address
77425	00015	14	42003	20031	FIXED → FLOATING
77426	00016	11	20000	00032	
77427	00017	14	30003	10032	N ÷ 10 → N
77430	00020	21	00033	00074	d.e. + 1 → d.e.
77431	00021	46	00017	00022	0:d.e.
77432	00022	43	00040	00026	0:d.e.
77433	00023	14	20003	10032	10N → N
77434	00024	23	00033	00074	d.e. -1 → d.e.
77435	00025	45	00000	00022	jump
77436	00026	11	00032	00000	store
77437	00027	16	00002	01736	exit
77440	00030	45	00000	01735	
77441	00031	24	00000	00004	10
77442	01070	00	00461	13200	10^7
76663	76663	72	77352	05000	loader parameters

Square Root Subroutine

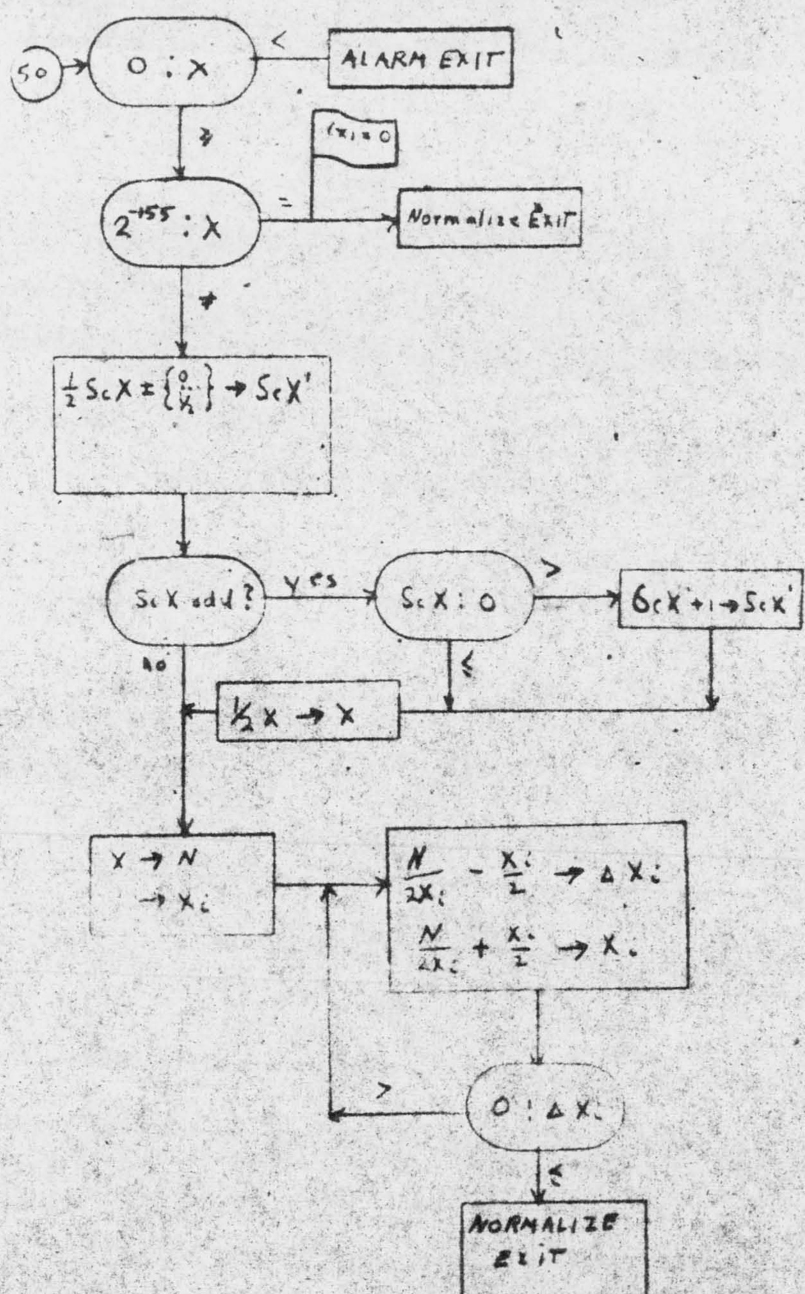
Command Code: 50

Number of Cells: 17

I Description

This command computes the square root of (x) and stores it in y and R with full accuracy. An Alarm Halt occurs for negative arguments.

II Flow Chart



50 SQUARE ROOT SUBROUTINE

76553	01000	11	00005	20000	X → A
76554	01001	46	76045	01002	x:0
76555	01002	42	00073	01020	2^{-155}:x (ie is (x) = 0?)
76556	01003	54	00010	00107	$\frac{1}{2}$ sc x ± {0, 2} → scx
76557	01004	43	20000	01010	scx odd?
76560	01005	46	01006	01007	scx:0
76561	01006	35	00074	00010	scx + 1 → scx
76562	01007	54	00005	00107	$\frac{1}{2}$ x → x
76563	01010	11	00005	00006	x → N
76564	01011	11	00070	00005	$1 - 2^{-35}$ → xi
76565	01012	31	00006	00042	$\frac{1}{2}$N (70) → A
76566	01013	73	00005	00004	$\frac{1}{2}$N ÷ Xi → Q, t_4 (35)
76567	01014	54	00005	00107	$\frac{1}{2}$ Xi → A, t_5 (35)
76570	01015	23	10000	00005	$\frac{1}{2}$N ÷ Xi − $\frac{1}{2}$Xi → Q = △ Xi
76571	01016	21	00005	00004	$\frac{1}{2}$N ÷ Xi + $\frac{1}{2}$Xi → Xi + 1
76572	01017	44	01012	01020	△ Xi:0
76573	01020	45	00000	01721	exit
76675	76675	21	76553	02000	loader parameter

FLIP

Print and Punch Subroutine

Print Command Code: 52

Punch Command Code: 53

Number of Cells: 56

I Description

These two commands convert (x) to a floating decimal form and print it on the flexowriter or punch a flexowriter tape. For these commands y is immaterial. If the Print subroutine is used, the Punch subroutine must be also specified to the loader. If the location of the first cell of the print subroutine is y, that of the punch subroutine must be y + 2.

Fifteen digits are printed or punched as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
±	q	.	q	q	q	q	q	q	sp	±	p	p	sp	sp

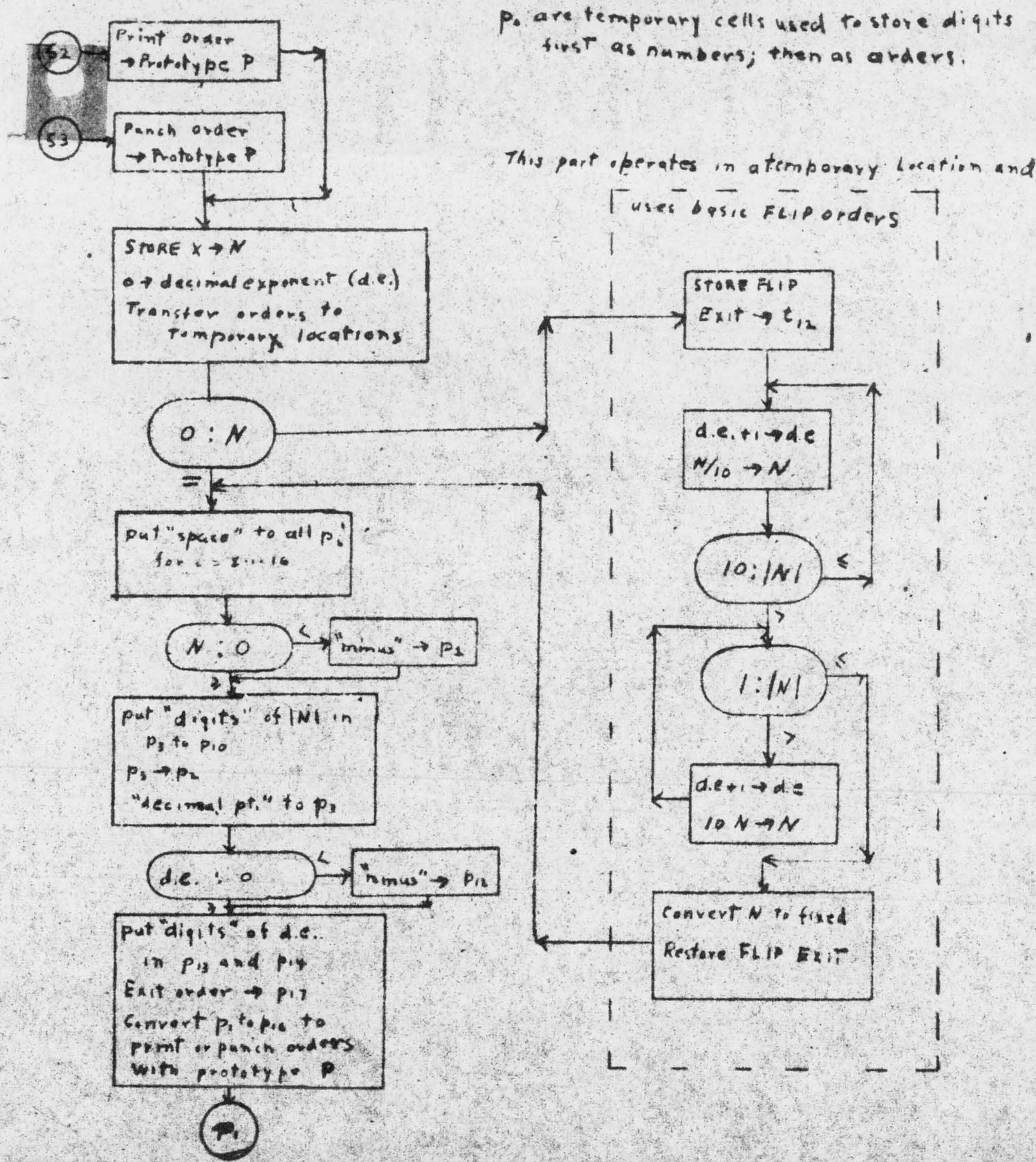
Where the number (in floating decimal form) is

$$n = q \cdot 10^{p}$$

No carriage return is included.

Print and Punch Subroutine

II Flow Chart



53 PUNCH SUBROUTINE

77262	01000	11	01067	00002	prototype $\rightarrow t_2$
77263	01001	11	00005	00035	$x \rightarrow t_{35}$
77264	01002	11	00040	00036	0 $\rightarrow$ decimal exponent (d.e.)
77265	01003	75	30016	01005	load temporary orders
77266	01004	11	01051	00015	
77267	01005	54	00005	20015	test X
77270	01006	43	20000	01010	
77271	01007	37	00027	00015	enter temporaries
77272	01010	75	10020	01012	put "space" to all digits
77273	01011	13	00074	00015	
77274	01012	71	00035	01041	$X \cdot 10^7$
77275	01013	46	01014	01015	adjust sign digit
77276	01014	11	01037	00015	
77277	01015	54	20000	00051	X·107 as integer
77300	01016	12	20000	20000	$\|x\|$
77301	01017	35	01050	20000	Round
77302	01020	73	01041	00016	1st digit
77303	01021	13	01040	00017	decimal point
77304	01022	75	30006	01024	Remaining digits
77305	01023	73	01042	00020	
77306	01024	11	01036	00037	set decimal point
77307	01025	11	00036	20000	adjust sign digit of d.e.
77310	01026	46	01027	01030	
77311	01027	11	01037	00027	
77312	01030	12	20000	20000	$\|d.e.\|$
77313	01031	73	01047	00030	1st digit d.e.
77314	01032	11	20000	00031	2nd digit d.e.
77315	01033	11	01063	00034	jump order
77316	01034	75	20017	00015	convert to orders
77317	01035	21	00015	00002	
77320	01036	00	00000	00042	decimal point

77321	01037	00	00000	00014	12.
77322	01040	00	00000	00006	6.
77323	01041	00	00461	13200	10^7
77324	01042	00	00036	41100	10^6
77325	01043	00	00003	03240	10^5
77326	01044	00	00000	23420	10^4
77327	01045	00	00000	01750	10^3
77330	01046	00	00000	00144	10^2
77331	01047	00	00000	00012	10
77332	01050	00	00000	00005	5
77333	00015	16	01736	00015	save exit address
77334	00016	21	00036	00074	d.e + 1 → d.e.
77335	00017	14	30003	20035	N ÷ 10 → N
77336	00020	14	45003	10016	10;N
77337	00021	14	45003	00025	1;N
77340	00022	23	00036	00074	d.e.-1 → d.e.
77341	00023	14	20003	20035	10N → N
77342	00024	45	00000	00021	jump
77343	00025	14	51003	20035	FLOATING → FIXED
77344	00026	16	00015	01736	restore exit
77345	00027	45	00000	01735	exit
77346	00030	37	77777	30000	1 (adjusted for rounding)
77347	00031	23	77777	75404	10 (adjusted for rounding)
77350	00032	24	00000	00004	10
77351	01067	63	00000	00045	prototype order
76703	76703	70	77262	03400	loader parameter

52 PRINT SUBROUTINE

77450	01000	11	00042	00002	set prototype
77451	01001	45	00000	01003	jump
76701	76701	02	77450	00200	loader parameter

FLIP

Logarithm Subroutine

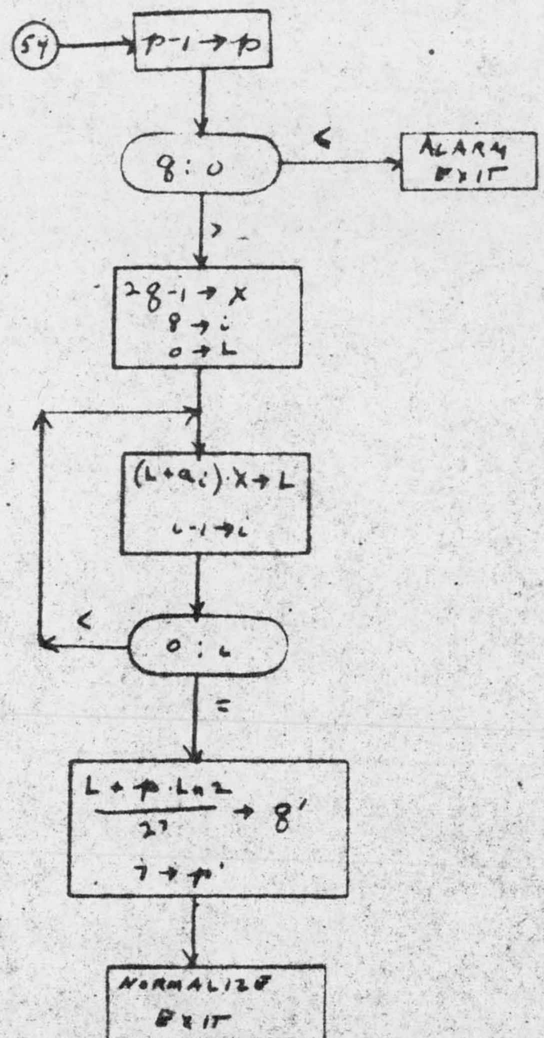
Command Code: 54

Number of Cells: 31

I **Description**

This command computes $\log_e(x)$ and stores the result in y and R. A polynomial approximation is used[(1)] which gives a maximum error of the order of $3 \cdot 10^{-8}$.

II **Flow Chart**



$$(x) = q \cdot 2^{p}$$

$$X = 2q - 1$$

$$Ln(x) = (p-1)\,Ln\,2 + Ln\,(X+1)$$

$$= q' \cdot 2^{p'}\ ;\ p' = 7$$

$$Ln\,(X+1) = a_1 X + a_2 X^2 + \cdots a_8 X^8$$

(1) See sheet 56, Approximations in Numerical Analysis, a publication of the Rand Corporation

54 LOGARITHM SUBROUTINE

76462	01000	36	00074	00010	p-1 → p
76463	01001	13	00005	20000	q:0
76464	01002	42	00066	01025	
76465	01003	36	00066	00006	2q-1 → x (35)
76466	01004	54	00006	00001	
76467	01005	11	01022	01007	8 → i
76470	01006	11	00040	00005	0 → L
76471	01007	21	00005	[30000]	L + a_i → L
76472	01010	71	20000	00006	x · L → L (35)
76473	01011	54	20000	00045	
76474	01012	11	20000	00005	
76475	01013	21	01007	00074	i-1 → i
76476	01014	42	01023	01007	0:i
76477	01015	54	00005	00101	L (28)
76500	01016	71	00010	01026	p.ln 2 → R (28)
76501	01017	35	00005	00005	L + p.ln 2 → q^1
76502	01020	11	00067	00010	7 → p^1
76503	01021	45	00000	01024	exit
76504	01022	21	00005	01027	prototypes
76505	01023	21	00005	01037	
76506	01024	45	00000	01721	normal exit
76507	01025	45	00000	01737	alarm exit
76510	01026	00	13056	20577	ln 2 (28)
76511	01027	77	62620	75765	a_8

76512	01030	01	11721	41642	a_7
76513	01031	74	74607	70746	a_6
76514	01032	05	27266	02203	a_5
76515	01033	70	22764	23456	a_4
76516	01034	12	47414	37545	a_3
76517	01035	60	00203	77320	a_2
76520	01036	37	77774	20006	a_1
76705	76705	37	76462	02400	loader parameter

FLIP

Sine and Cosine Subroutine

Sine Command Code: 60

Cosine Command Code: 61

Number of Cells: 51

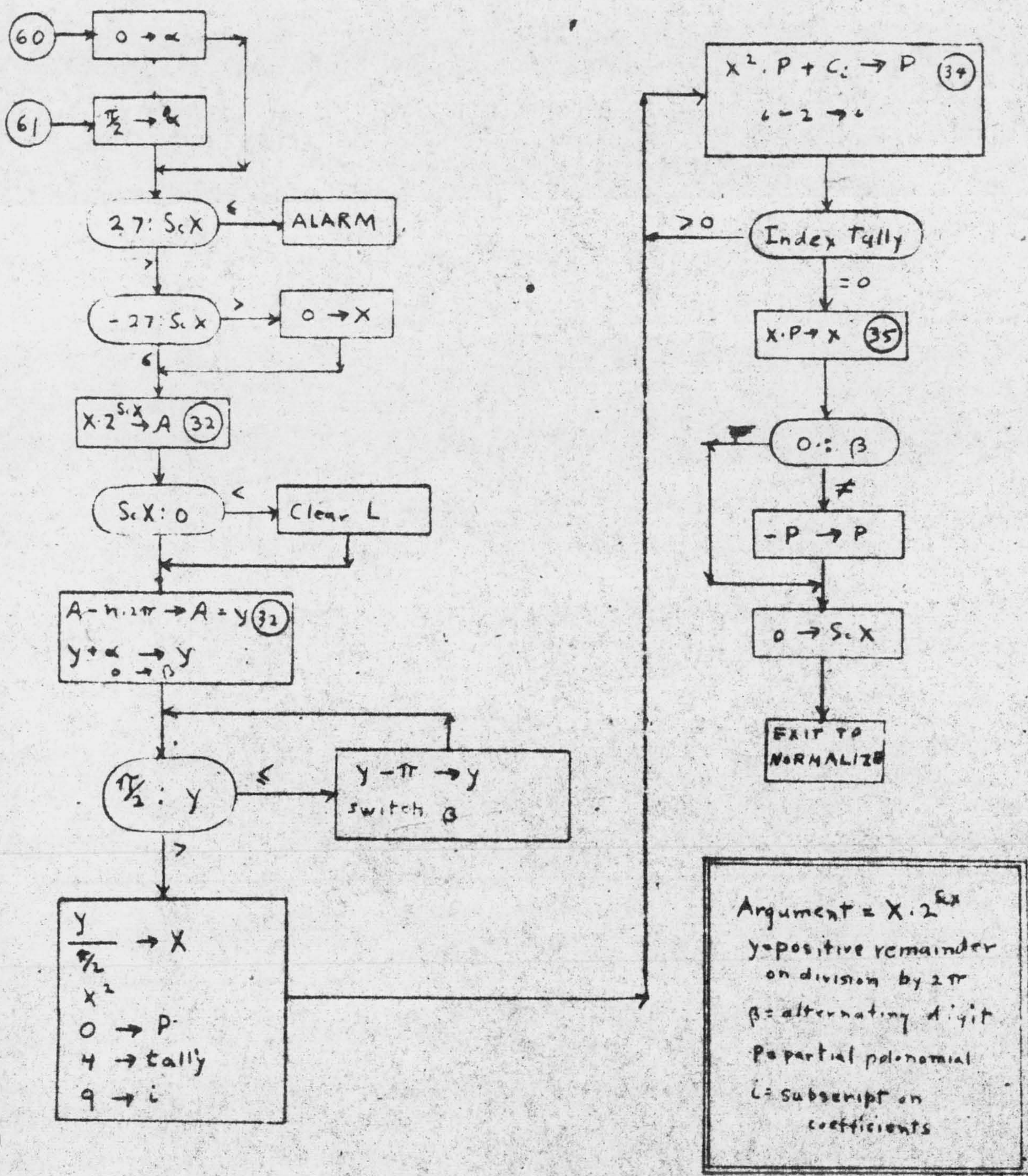
I Description

These two commands compute the sine or cosine of (x) and store it in y and R. A polynomial approximation is used[1] which gives a maximum error of the order of $-5 \cdot 10^{-9}$. For arguments so large that the roundoff error of the argument obscures the result, an alarm halt occurs. If the cosine subroutine is used, the sine subroutine must also be specified to the loader; and if the location of the first cell of the cosine subroutine is y, that of the sine subroutine must be y + 2.

(1) See sheet 14, "Approximations in Numerical Analysis", a publication of the Rand Corporation

Sine and Cosine Subroutine

II Flow Chart



61 COSINE SUBROUTINE

77452	01000	11 01054 00004	$\pi/2 \rightarrow t_4$
77453	01001	45 00000 01003	jump
76717	76717	02 77452 00200	loader parameter

60 SINE SUBROUTINE

77454	01000	11 00040 00004	$0 \rightarrow t_4$
77455	01001	23 00010 00056	scx-27 $\rightarrow$ scx
77456	01002	46 01003 76045	ALARM?
77457	01003	35 00053 20000	scx-27 + 54 $\rightarrow$ A
77460	01004	46 01006 01007	Zero Result?
77461	01005	00 00000 00036	30.
77462	01006	11 00040 00005	$0 \rightarrow$ x
77463	01007	36 01005 10000	scx-27 +54-30 $\rightarrow$ Q,A
77464	01010	35 01050 01011	} $x \cdot 2^{scx} \rightarrow x$
77465	01011	[11 00010 10000]	}
77466	01012	44 01013 01014	left shift?
77467	01013	11 20000 20000	$0 \rightarrow$ L
77470	01014	73 01060 10000	$x - n \cdot 2\pi \rightarrow A = Y$ (32)
77471	01015	35 00004 20000	$Y + \left\{ \begin{matrix} 0 \\ \pi/2 \end{matrix} \right\} \rightarrow Y$
77472	01016	11 00066 00004	+ sign $\rightarrow t_4$
77473	01017	42 01052 01023	$\pi/2 : Y$
77474	01020	55 00004 00001	change sign t_4
77475	01021	36 01051 20000	$Y - \pi \rightarrow Y$
77476	01022	45 01057 01017	jump
77477	01023	54 20000 00043	} $Y/2\pi \rightarrow x$ (35)
77500	01024	73 01052 00005	}
77501	01025	71 00005 10000	}
77502	01026	54 20000 00045	} $x^2 \rightarrow t_6$ (35)
77503	01027	11 20000 00006	}
77504	01030	11 00040 00007	$0 \rightarrow$ P

77505	01031	15	01022	01035	9 → i
77506	01032	11	00044	00010	4 → tally
77507	01033	71	00007	00006	
77510	01034	54	20000	00045	x.P + ci → P (34)
77511	01035	35	01060	00007	
77512	01036	23	01035	00073	step i
77513	01037	41	00010	01033	done?
77514	01040	71	00007	00005	
77515	01041	54	20000	00046	x.P → x (35)
77516	01042	11	20000	00005	
77517	01043	11	00004	10000	check sign t
77520	01044	44	01045	01046	
77521	01045	13	00005	00005	-x → x
77522	01046	11	00040	00010	0 → acx
77523	01047	45	00000	01721	exit
77524	01050	54	00005	24110	prototype
77525	01051	14	44176	65200	π (32)
77526	01052	06	22077	32504	π/2 (32)
77527	01053	31	10375	52202	c_1
77530	01054	65	52420	76452	c_3
77531	01055	01	21464	25731	c_5
77532	01056	77	73155	46346	c_7
77533	01057	00	00117	32757	c_9
77534	01060	31	10375	52421	2π (32)
76715	76715	61	77454	04700	loader parameter

FLIP

Arc Tangent and Arc Cotangent Subroutine

Arc Tangent Command Code: 62

Arc Cotangent Command Code: 63

Number of Cells: 47

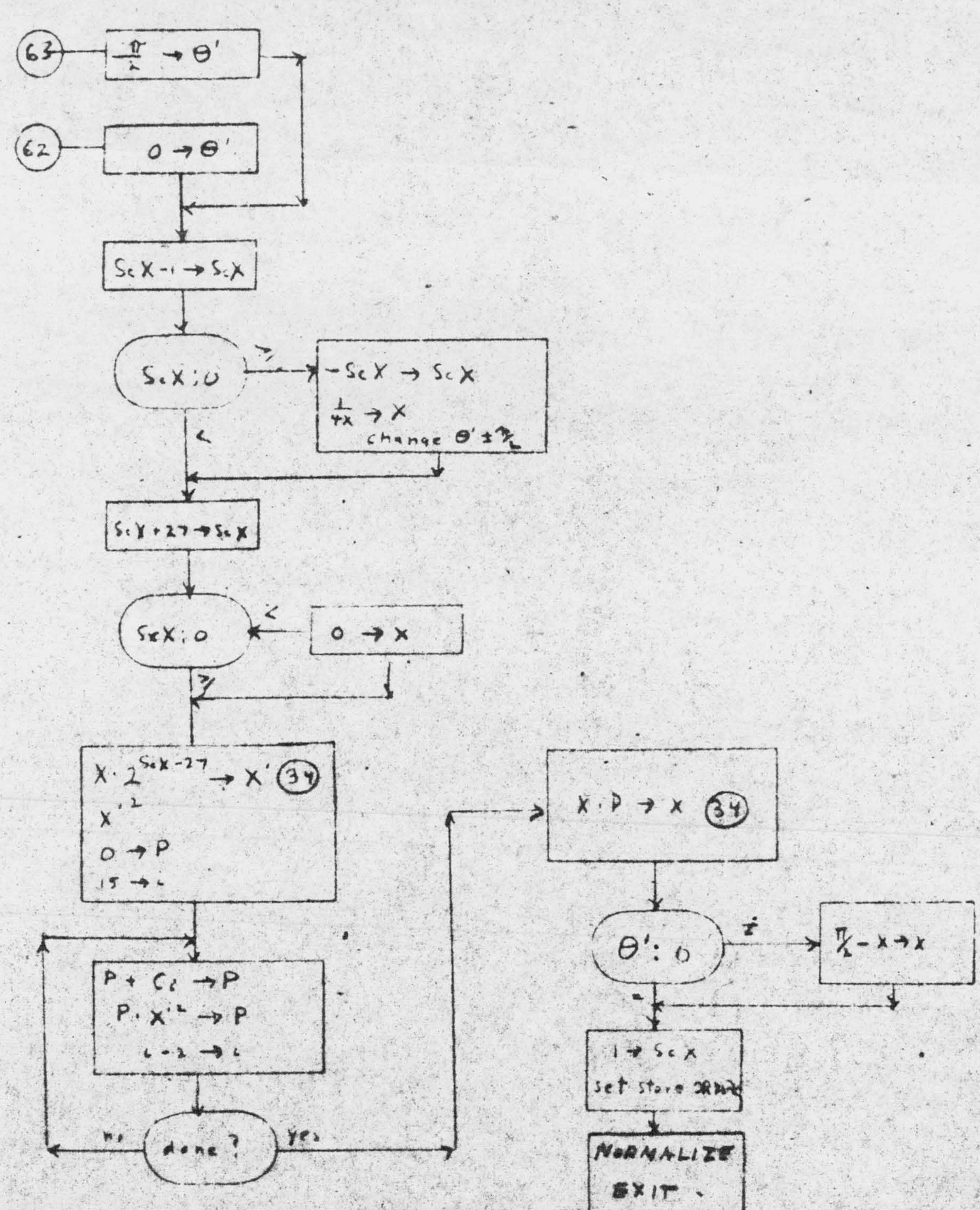
I Description

These two commands compute $\tan^{-1}X$ or $\cot^{-1}X$ and store the result in y and R. A polynomial approximation is used.(1) The error is of the order of $1/2 \cdot 10^{-7}$. If the arc cotangent subroutine is used, the arctangent subroutine must also be specified to the loader; and if the location of the first cell of the arctangent subroutine is y, that of the arc cotangent subroutine must be y - 2.

(1) See sheet 13, "Approximations in Numerical Analysis", a publication of the Rand Corporation"

Arc Tangent and Arc Cotangent Subroutine

II Flow Chart



62 Arc Tangent Subroutine

76300	01000	11 00040 *0012	$0 \rightarrow \theta^1$
76301	01001	23 00010 00074	Scx-1 → Scx
76302	01002	46 01007 01003	0:Scx
76303	01003	13 00010 00010	-Scx → Scx
76304	01004	31 00066 00042	$\frac{1}{4x} \rightarrow x$
76305	01005	73 00005 00005	
76306	01006	27 00012 00062	reverse θ' ($\pi/2$ (34) or zero)
76307	01007	21 00010 01043	Scx + 27 → Scx
76310	01010	46 01011 01012	0:Scx
76311	01011	11 00040 20000	0 → Scx
76312	01012	35 01054 01013	$x.2^{Scx-27} \rightarrow x^1$ (34)
76313	01013	[54 00005 50055]	
76314	01014	71 00005 10000	$(x^1)^2$ (34)
76315	01015	54 20000 00046	
76316	01016	11 20000 00006	
76317	01017	11 00040 00010	0 → P
76320	01020	15 01033 01023	15 → i
76321	01021	⌈71 00010 00006	$P \cdot (x^1)^2 + C_i \rightarrow P$ (35)
76322	01022	54 20000 00046	
76323	01023	35 [01044] 00010	
76324	01024	21 01023 00073	i-2 → i
76325	0r025	42 01034 01021]	done?
76326	01026	71 00005 00010	$P \cdot x^1 \rightarrow x$ (34)
76327	01027	54 20000 00045	
76330	01030	11 20000 00005	
76331	01031	11 00012 20000	$0 : \theta^1$
76332	01032	47 01035 01040	

76333	01033	00	01044	00000	constants
76334	01034	35	01054	00000	
76335	01035	44	01036	01037	$0:x^1$
76336	01036	13	20000	20000	$-\theta^1 \rightarrow \theta^1$
76337	01037	36	00005	00005	$\theta^1 - \pi/2 \rightarrow \theta^1$
76340	01040	11	00074	00010	1 $\longrightarrow$ scx
76341	01041	16	00013	01734	set order
76342	01042	45	00000	01721	exit
76343	01043	00	00000	00033	27.
76344	01044	77	67545	00613	c_{15}
76345	01045	00	54613	12165	c_{13}
76346	01046	76	15376	17035	c_{11}
76347	01047	03	05357	57500	c_9
76350	01050	73	43116	35123	c_7
76351	01051	06	30402	45553	c_5
76352	01052	65	25317	10166	c_3
76353	01053	37	77777	23166	c_1
76354	01054	54	00005	00055	prototype order
76721	76721	55	76300	03600	loader parameter

63 ARC COTANGENT SUBROUTINE

76356	01000	11	00062	00012	$\pi/2$ (34) $\rightarrow \theta^1$
76357	01001	45	00000	01003	jump
76723	76723	02	76356	00200	loader parameter

FLIP

Exponential Subroutine

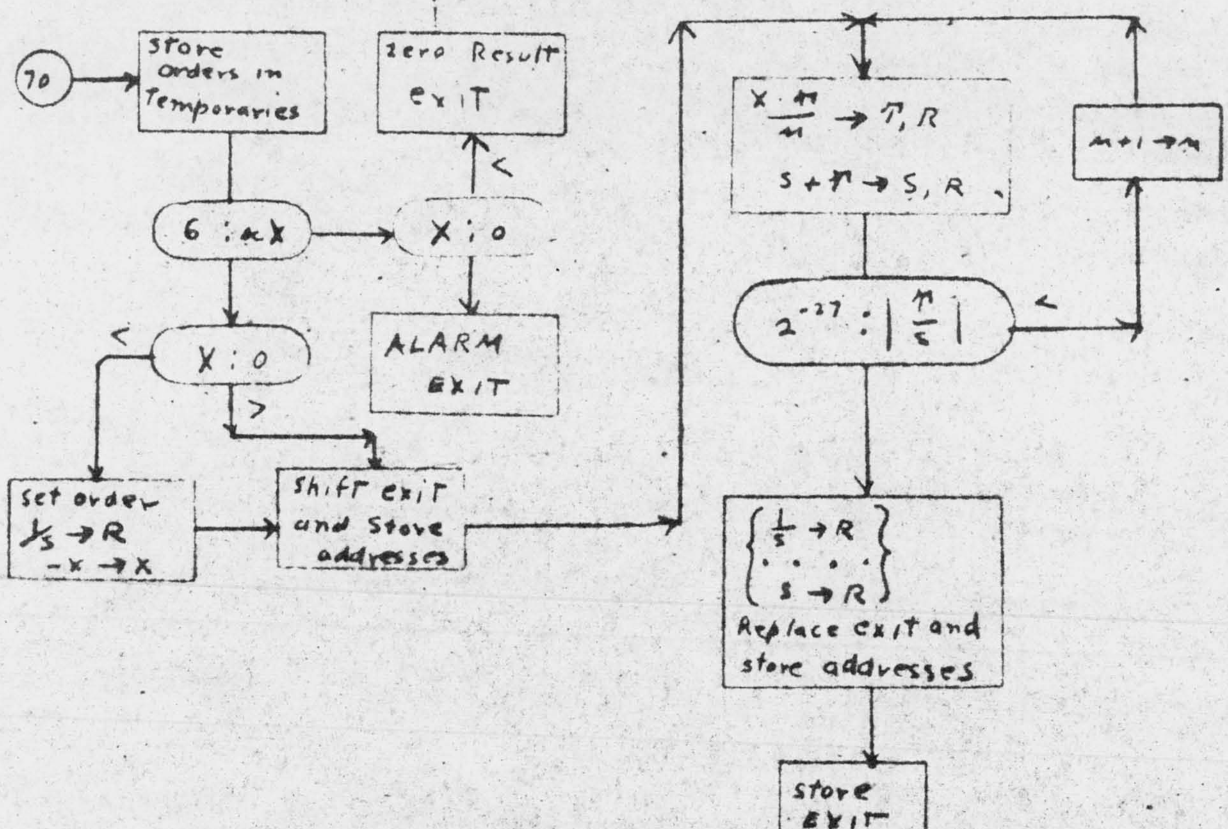
Command Code: 70

Number of Cells: 25

I **Description**

This command computes the exponential of (x) and stores it in y and R. Full accuracy is obtained by a power series. For values of (x) > 64, an alarm halt occurs. e^{-x} is obtained by computing e^{+x} and reciprecating.

II **Flow Chart**



70 EXPONENTIAL SUBROUTINE

76360	01000	11	00005	00037	$x \rightarrow t_{37}$
76361	01001	36	00067	10000	scx - 7 → Q
76362	01002	11	00005	20000	x → A
76363	01003	44	01004	01030	scx-7 : 0
76364	01004	75	30011	01006	set up temporary orders
76365	01005	11	01015	00015	
76366	01006	75	10004	01010	preset variables
76367	01007	11	01026	00026	
76370	01010	47	01011	01013	x:0
76371	01011	11	01027	00023	set order $\frac{1}{s}$ → R
76372	01012	27	00037	01772	-x → x
76373	01013	75	30003	00016	shift addresses
76374	01014	16	01734	00034	
76375	00015	[14	00003	10030	n + 1 → n
76376	00016	14	32003	00037	$\frac{x \cdot T}{n} \rightarrow T$, R
76377	00017	14	21002	60026	
76400	00020	14	00002	60027	s + T → S,R
76401	00021	14	07002	70026	$\lvert T/s \rvert : 2^{-27}$
76402	00022	47	00015	00023	
76403	00023	[11	00027	20000]	S → R
76404	00024	75	30003	01734	shift addresses, EXIT
76405	00025	16	00034	01734	
76406	01026	20	00000	00001	1
76407	01027	14	32002	70031	1/s → R
76410	01030	47	01607	76045	x:0
76735	76735	31	76360	01200	Loader parameter

FLIP

Trace Subroutine

Command Code: 77

Number of Cells: 26

I Description

The trace routine operates in two phases. Phase I operates concurrently with the running of the routine being tested and stores information on Magnetic Tape 2 (MT2). Phase II operates separately from the routine being tested. It reads the information which was stored by Phase I on MT2, processes it and punches a paper tape output. The content of ES will be automatically restored after this phase.

II Phase I

The trace subroutine must be specified to the loader. Its "command code" for this purpose is 77. It requires 26 cells. When loaded, it will operate whenever MJ1 is on. The MJ instruction is in cell 01735. This subroutine uses cells 74000 to 74041 as temporaries.

III Phase 2

This operation uses the ES and cell 40000 but will restore both when completed. Its operating instructions are: -

1. Set PAK to 77600. Press Start. The 1103 prints out "Rewind MT2" and halts.
2. After rewinding MT2, start. (if PAK was disturbed, set it to 40000). The routine will search the tape for the data, then process it one block at a time. The output is punched on paper tape. The end of data will be apparent when the routine searches MT2 without punching paper tape. Halt.
3. To continue the problem, set PAK to 40000 and start. The 1103 will restore ES and 40000 and halt with a 56 00000 40000 command.

IV Output

The trace routine output prints a 32 digit line for each FLIP instruction. If a jump occurs, either an erroneous line appears or no line at all. Some FLIP subroutine commands include FLIP basic commands. These will appear as extra lines before the FLIP subroutine command line. The lines have the form:

AAAA OP xxxx yyyy ± q.qqqqqq ± pp

where AAAA is the last four digits of the address of the instruction.

OP is the command code.

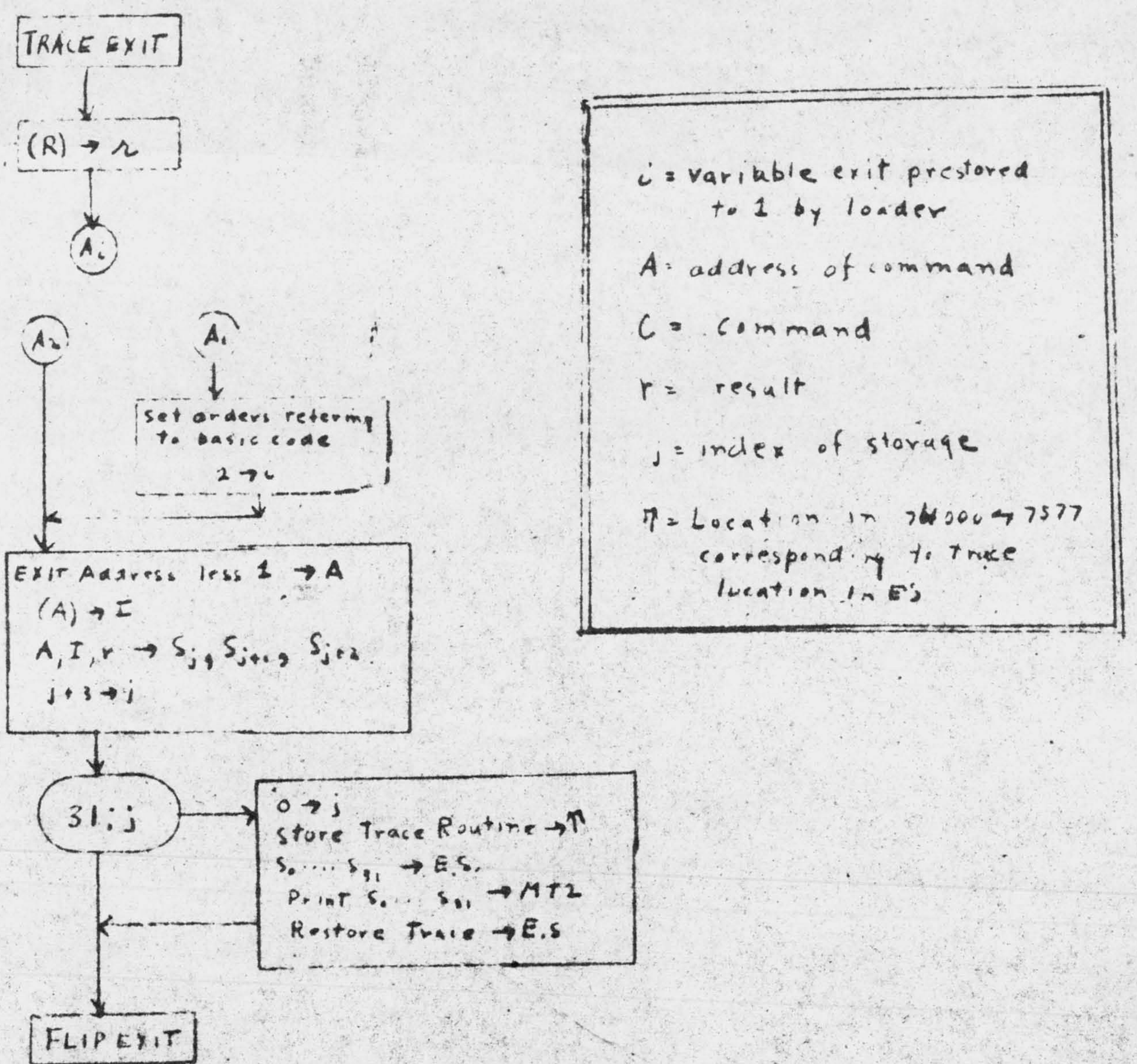
xxxx is the basic x address.

yyyy is the basic y address.

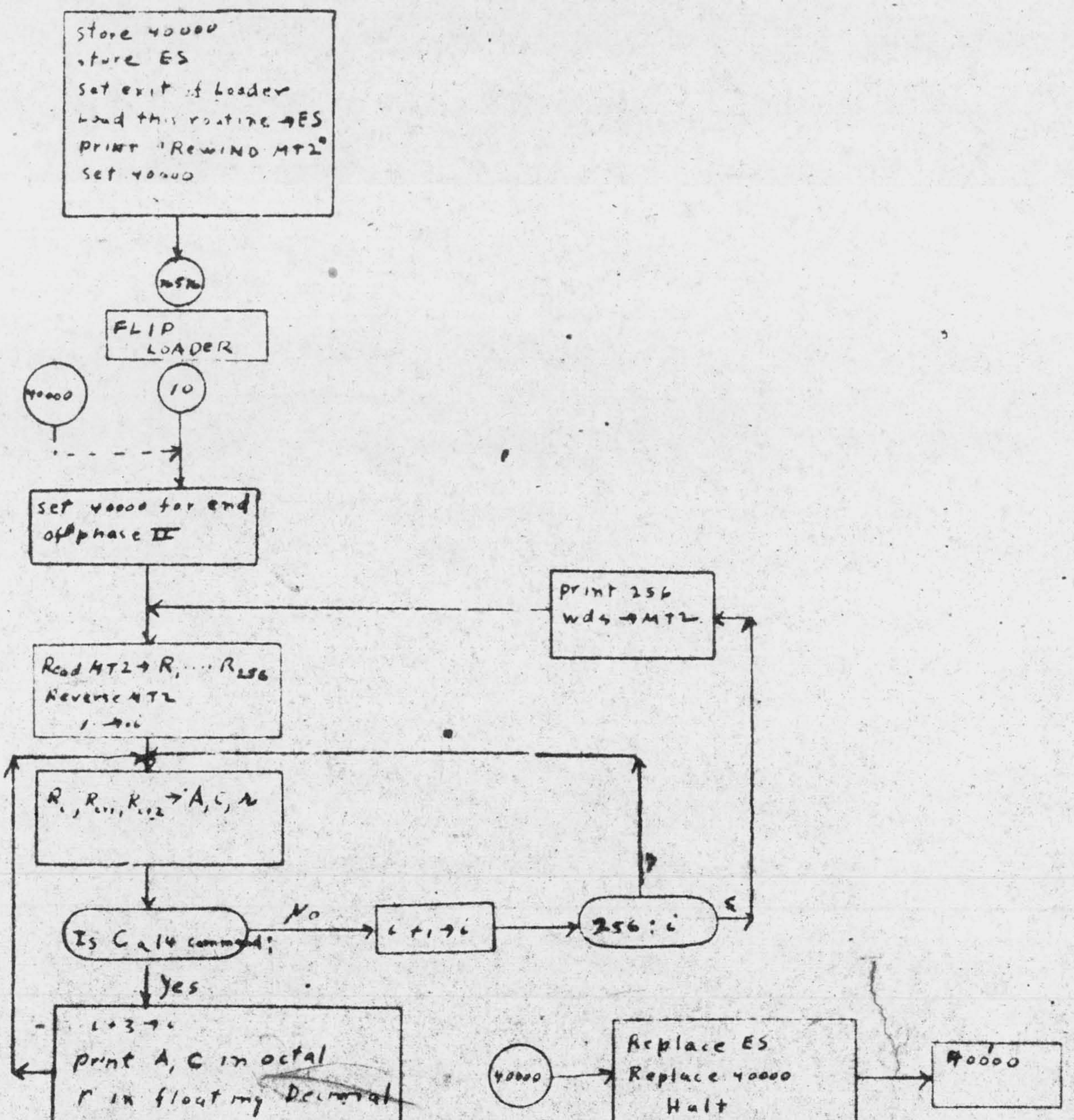
The result of the operation, in floating decimal form, is $q \cdot 10^p$ where $1 \leq q < 10$. Some of the FLIP subroutine commands do not leave their result in R. For these, the result, $q \cdot 10^p$ will be erroneous.

In order to avoid confusion when several problems are traced using the same magnetic tape, Phase II overwrites the trace information as it is processed.

V Flow Chart, Phase I



VI Flow Chart, Phase II



77 PHASE I of TRACE

76521	01000	37 01000 01023	2 → i
76522	01001	11 20000 01025	store R
76523	01002	31 01736 00017	
76524	01003	23 20000 00073	exit address → A
76525	01004	11 20000 01023	
76526	01005	15 20000 01006	(A) → I
76527	01006	11 30000 01024	
76530	01007	75 30003 01011	store A, I, R on drum
76531	01010	11 01023 74001	
76532	01011	21 01010 00043	step
76533	01012	42 01026 01030	done?
76534	01013	16 01027 01010	restore
76535	01014	75 30036 01016	place es on MD
76536	01015	11 01000 75000	
76537	01016	75 30036 75020	place information in ES
76540	01017	11 74001 01000	
76541	01020	65 20001 01000	print on M T 2
76542	01021	75 30036 01030	restore ES
76543	01022	11 75000 01000	
76544	01023	15 01031 01002	set order
76545	01024	75 10040 01001	clear temporaries
76546	01025	11 00040 74001	
76547	01026	11 01023 74037	constants
76550	01027	00 00000 74001	
76551	01030	11 01025 20000	restore R
76552	01031	45 01736 01736	exit
76753	76753	32 76521 03100	loader parameter

PHASE II of TRACE

77600	00117	11 40000 74000	store 40,000
77601	00120	75 31777 77673	store E.S.
77602	00121	11 00001 74001	
77603	00122	53 00000 01300	Loader parameters
77604	00123	00 00000 00000	
77605	00124	45 00000 00215	return from loader
77606	00125	37 00200 00201	print flex characters
77607	00126	45 12203 11406	Flex characters
77610	00127	22 04470 70157	
77611	00130	74 45000 00000	
77612	00131	00 00000 00000	
77613	00132	11 00124 40000	preset 40,000
77614	00133	45 00000 76575	enter loader
77615	00134	67 20010 00000	back tape
77616	00135	15 00172 00137	preset order
77617	00136	75 30003 00140	pickup A,I,R
77620	00137	11 [00700] 00020	
77621	00140	11 00021 20000	Test I
77622	00141	42 00170 00143	
77623	00142	42 00171 00146	
77624	00143	21 00137 00073	Step by one
77625	00144	42 00173 00136	done?
77626	00145	45 00000 00222	jump
77627	00146	21 00137 00174	Step by three
77630	00147	63 00000 00042	punch carriage return
77631	00150	55 00020 00011	shift A by 3 octal
77632	00151	37 00166 00157	punch A
77633	00152	55 00021 00006	shift C

77634	00153	11	00074	00006	1 → tally	
77635	00154	37	00166	00160	punch operation	
77636	00155	37	00166	00157	punch x	
77637	00156	45	00000	00175	jump	
77640	00157	11	00043	00006	3 → tally	SUBROUTINE FOR OCTAL PUNCH
77641	00160	⌈55	10000	00003	SHIFT	
77642	00161	51	00067	20000	X-TRACT DIGIT	
77643	00162	35	00167	00163	set order	
77644	00163	[00	00000	00000]	punch digit	
77645	00164	41	00006	00160⌋	Index	
77646	00165	63	00000	00044	punch space	
77647	00166	45	00000	00000	exit	
77650	00167	63	00000	00045	prototype order	
77651	00170	13	77777	77777	constants	
77652	00171	14	77777	77777	constants	
77653	00172	00	00700	00000	constants	
77654	00173	11	01275	00000	constants	
77655	00174	00	00003	00000	constants	
77656	00175	37	00166	00157	punch y	
77657	00176	14	53002	20000	punch r	
77660	00177	45	00000	00136	jump	
77661	00200	45	00000	[00000]	exit	SUBROUTINE FOR FLEX PRINT
77662	00201	⌈31	00200	00017	set order	
77663	00202	15	20000	00204	set order	
77664	00203	21	00200	00074	step	
77665	00204	31	00000	00044	Word → L	
77666	00205	47	00206	00200	done?	
77667	00206	⌈54	20000	00006	shift next digit	
77670	00207	61	00000	20000	punch	

77671	00210	27	00040	00040	clear R
77672	00211	47	00206	00201	done?
77673	00212	11	77605	00010	set exit of loader
77674	00213	75	30200	00125	put phase II into E.S.
77675	00214	11	77603	00122	
77676	00215	37	40000	00223	set 40,000 for restoration
77677	00216	75	31777	77701	restore ES
77700	00217	11	74001	00001	
77701	00220	11	74000	40000	restore 40,000
77702	00221	56	00000	40000	halt
77703	00222	65	20010	00300	print over old date
77704	00223	64	20010	00700	read new data
77705	00224	75	10400	00134	store zeros for printing
77706	00225	11	00040	00300	

A MAGNETIC TAPE INTERPRETIVE ROUTINE

"TAPEWORM"

Description

"Tapeworm" is a routine for controlling the operation of the Uniservos on the Univac Scientific, Model 1103A. It is an interpretive routine which permits a programmer to specify with a single Interpret instruction a read, write, move, or rewind operation. This routine includes provision for re-reading a block in case of a parity check failure and automatic selection of the writing mode according to the peripheral equipment for which the tape is intended.

Form of the Interpret Instruction

Use of the Interpret instruction provides a convenient means of referencing magnetic tape subroutines. Because of the availability of 10 octal digits in the instruction for the storage of parameters, any of the Uniservo operations may be specified with a single word. The form of the Interpret instruction for the operations of reading, writing, moving and rewinding is given below. Here each character represents an octal digit of the instruction.

a. IP OP -B UU SSSS for reading tape

b. IP OP △B UU SSSS for writing tape

c. IP OP -- UU BBBB for moving tape

d. IP OP △- UU ---- for rewinding tape

These Interpret instructions specify magnetic tape operations as follows:

a. Reading Tape: IP OP -B UU SSSS

IP = 14

OP = 01 For reading foreward

= 05 For reading backward

B = the number of blocks (not words) to be read

UU = the Uniservo involved

SSSS = the address in Rapid Access Storage of the first cell to be read into

b. Writing Tape: IP OP △B UU SSSS

IP = 14

OP = 03

△ = the code number as given below of the peripheral equipment for which the tape is intended

B = the number of blocks (not words) to be written

UU = the Uniservo involved

SSSS = the address in Rapid Access Storage of the first cell to be read from

The following code numbers are used for △:

△ = 0 1103 A internal

= 1 High Speed Printer

= 2 Uniprinter

= 3 Magnetic Tape-to-Punched-Card Converter

= 4 Unitape-to-Teletape Converter

= 5 Unitape Transmission

Tapes prepared for the Unitape-to-Teletape Converter may be used for Unitape Transmission and conversely. Any tapes may be read into the 1103A.

c. Moving Tape: IP OP — UU BBBB

IP = 14

OP = 02 for moving foreward

= 06 for moving backward

UU = the Uniservo involved

BBBB = the number of blocks to be moved

d. Rewinding: IP OP △- UU ----

IP = 14

OP = 04

△ = 1 if interlock is desired

= 0 if interlock is not desired

UU = the Uniservo involved

Digits denoted by a dash may be filled with anything. A maximum of seven blocks may be written or read with a single Interpret instruction but the tape may be moved as many as 7777 blocks. Note that the block number for moving is in a different position than the block number for reading or writing.

In reading backward, the first word goes into SSSS and the second into address SSSS - 1; in reading foreward the second word goes into address SSSS + 1.

Reread

After a block is read from magnetic tape, the contents of IOA are examined to see if a parity check failure has occurred. If there is no failure, the next block is read in without stopping the tape. If there is a failure, the reread subroutine is entered. This subroutine performs successive rereads of the block, reversing the direction of the read after each failure. In case the direction of the original read is foreward, the rereads are performed in the following sequence:

1. read backward, normal bias
2. read foreward, high bias
3. read backward, high bias
4. read foreward, low bias
5. read backward, low bias

After each reread, the parity check indicator is tested. If the reread has been successful, the tape is positioned at the end of the block, the bias returned to normal, and the next block read. If a reread fails, the next reread in the sequence is performed. Should the entire sequence be executed without a successful reading of the block, the computer is stopped. A restart of the computer will continue with the reading of the next block.

References to Tapeworm

When an Interpret instruction at address y is executed, y + 1 is inserted in the U-address of F_1 and the next instruction is obtained from F_2. F_1 must contain a manual Jump instruction which provides the exit from Tapeworm to the next instruction of the main program at address y + 1. F_2 must contain a manual Jump to the entrance of Tapeworm.

The first eight instructions of Tapeworm detect whether an Interpret instruction is referring to this magnetic tape interpretive routine or to some other interpretive routine used by the same program. With the Interpret instruction of the form

IP OP XXXX XXXX,

if $00 < OP < 07$ it is assumed the Interpret instruction refers to Tapeworm. Two exits are provided, one for OP = 00 and the other for $OP \geq 07$. If a program refers to interpretive routines other than Tapeworm, these exits may lead to the other interpretive routines. If no other interpretive routines are used, the first eight instructions of Tapeworm may be omitted.

Location

Tapeworm occupies 149_{10} cells of Rapid Access Storage. Of these, the parity check test and reread routine occupy 50_{10} cells; 8 cells are used to determine if an Interpret instruction refers to Tapeworm.

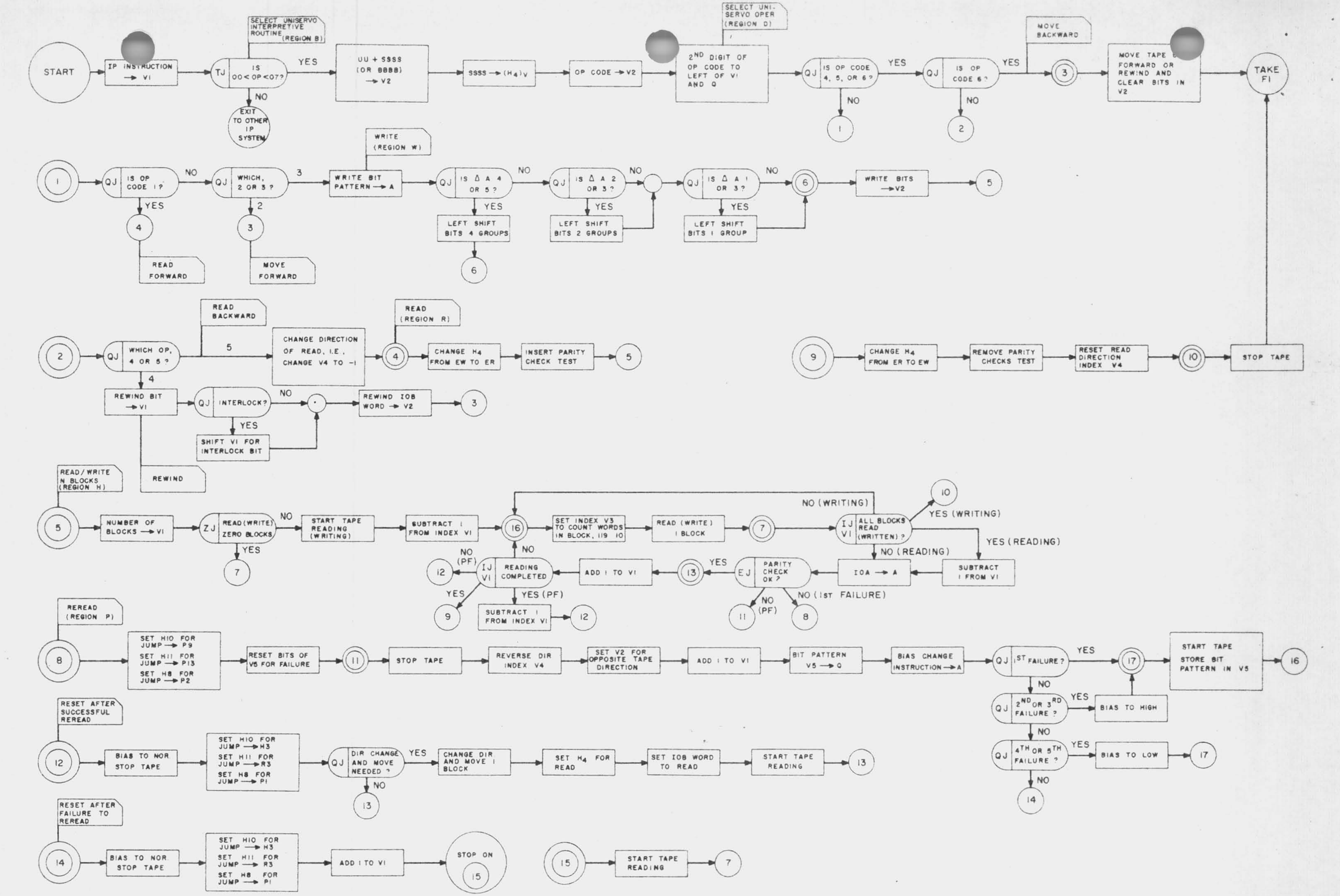
Speed

For a reading or writing operation there is approximately one millisecond between execution of the Interpret instruction which specifies the Uniservo operation and the External Function instruction which initiates the operation. This time is increased to 1.25 milliseconds if it is necessary to identify the interpretive routine being referred to.

Notes

If absolute addresses are assigned to Tapeworm, starting with b_1 = 01000, it is assembly modifiable.

In case Tapeworm is interrupted during operation, it must be again transferred into Rapid Access Storage in order to operate correctly the next time it is used. If Tapeworm stops after an unsuccessful reread, it will reset itself if and only if it is restarted without any changes in the computer controls.



OP CODES:

01 - READ FORWARD
02 - MOVE FORWARD
03 - WRITE
04 - REWIND
05 - READ BACK
06 - MOVE BACK

VALUES OF Δ:

0 1103A INTERNAL
1 HIGH SPEED PRINTER
2 UNIPRINTER
3 TAPE TO CARD CONVERTER
4 UNITAPE-TELETYPE CONVERTER
5 UNITAPE TRANSMISSION

ABBREVIATIONS:

F1 - ADDRESS 00000
F2 - ADDRESS 00001 (JUMP TO START OF THIS PROGRAM)
PF - PREVIOUS FAILURE
OP - OPERATION CODE
$(H_4)_V$ - CONTENTS OF V-ADDRESS AT ADDRESS H_4

KEY:

REMINDER FLAG
OPERATION
DECISION (JUMP)
5 CONNECTOR
5 REFERENCED CONNECTOR

MAGNETIC TAPE INTERPRETIVE ROUTINE

"TAPEWORM"

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS
Region B: Selects Uniservo Interpretive Routine	b1	31	0	17	IP instruction → A; left shift 15
		36	c1	a	subtract 1 from u-address
		15	a	b2	address of IP → u of b2
	b2	11	[y]	v1	IP instruction → v1
		11	v1	a	IP instruction → a
		42	c2	exit 1	Uniservo instruction? yes → d1
		42	c3	d1	too small → exit 1
		45		exit 2	too large → exit 2
Region D: Set Parameters; Select Uniservo Operation	d1	11	c4	q	mask → q
		53	v1	v2	uu + ssss (or bbbb) → v2
		11	c5	q	mask → q
		53	v2	h4	ssss → v address for EW or ER
		55	v1	34	Right shift IP by 8
		53	c6	v2	OP code → v2
		55	v1	21	2nd octal digit of OP code to left end
		44	d2	w1	Back or rewind? yes → d2 no → w1
	d2	44	d7	d3	Read back or rewind → d3 Move back → d7
	d3	44	r1	d4	Read back → r1 Rewind → d4

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS
	d4	11	c8	v1	Rewind digit → v1
		55	q	2	Interlock? yes → d5
		44	d5	d6	no → d6
	d5	55	v1	1	Set IOB_{22} -- IOB_{23} for rewind with interlock
	d6	55	c6 q	40	Insert uu
		53	v2	v1	
		31	c1	20	Rewind IOB word → v2
		35	v1	v2	
	d7	17		v2	Uniservo selection for move and rewind
		15	c9	v2	Clear digits in v2
		45	0	f1	Exit

<u>Region W:</u>

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS	
Set up to write	w1	44	w2	r2	Move foreward or write → w2 Read foreward → r2	
	w2	44	w3	d7	Write → w3 Move foreward → d7	
	w3	11	c10	a	Bit pattern → a	
		44	w4	w5	Separate Δ Δ = 4 or 5 → w4 Δ = 0 - 3 → w5	Shift write bits in a.
	w4	54	a	20	Left shift 16 (4 groups)	
		45		w9		
	w5	44	w6	w7	Δ = 2 or 3 → w6 Δ = 0 or 1 → w7	
	w6	54	a	10	Left shift 8 (2 groups)	

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION				REMARKS
	w7	44	w8		w9	$\Delta = 1 \rightarrow$ w8 $\Delta = 0 \rightarrow$ w9
	w8	54	a		4	Left shift 1 group
	w9	55	c6	q	2	Set mask for write bits
Region H:		53	a		v2	Write bits → v2
Read / Write	h1	55	v1	q	11	Number of blocks → v add. of q
		51	c7		v1	Number of blocks → v1
		47	h2		h5	Read (write) 0 blocks? yes → h5 no → h2
	h2	17			v2	Start tape writing (reading)
		23	v1		c9	Set index for 2nd IJ
	h3	11	c11		v3	Set index for 1st IJ
	h4	77 [7]	10000		[ssss]	Write (read) 1 block
		21	h4		v4	
		41	v3		h4	
	h5	41	v1		[h3]	n blocks done? no → h3 if writing h7 if reading
		23	v1		c9	
	h6	45			[r4]	yes → r4 if writing h7 if reading
	h7	76			a	Parity check OK?
	h8	43	c9		[p1]	Yes → h9 no → p1 if 1st failure p2 if previous failure
	h9	21	v1		c9	
	h10	41	v1		[h3]	All blocks read? no → h3 no previous failure p9 previous failure yes → h11

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS
	h11	45		[r3]	All blocks are read : no previous failure → r3 previous failure → p13
Region R:					
Set up to read	r1	13	v4	v4	Change direction of read
	r2	23	h4	c13	EW → ER
		21	h5	c14	Insert Parity check
		23	h6	c15	Set exit for Parity check
		45		h1	Go to read
	r3	21	h4	c13	ER → EW
		23	h5	c14	Remove Parity check
		21	h6	c15	(Remove Parity check)
		11	c9	v4	Reset direction index
	r4	17		c12	Stop tape
		45		f1	Exit
Region P:					
Parity-Error	p1	21	h10	c16	Set exits from ER
Reread		21	h11	c18	(Set exits from ER)
		21	h8	c17	(Set exits from ER)
		11	c19	v5	Reset bit formation
	p2	17		c12	Stop tape
		13	v4	v4	Change direction index
		21	h4	v4	
		55	v4	q 22	Line up direction bit with IOB

18

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS
		23	v2	q	Change tape direction
		21	v1	c9	Reset block counter to offset h10
		11	v5	q	Set bit sequence
		11	c20	a	Bias change instruction → a
		44	p3	p8	1 change bias → p3 0 not change bias → p8
	p3	44	p4	p6	1 low and stop → p4 0 high → p6
	p4	44	p11	p5	1 stop → p11 0 low → p5
	p5	35	c9	p7	Modify p7 for bias change
	p6	35	c9	p7	
	p7	17		(c21-1)	Change bias
	p8	17		v2	Start tape
		11	q	v5	Store parity check bit pattern
		45		h3	Reread
	p9	17		c23	Set bias to normal; stop
		23	h10	c16	Reset exits from ER
		23	h11	c18	
		23	h8	c17	
		11	v5	q	Direction change and move needed?
		44	p10	h9	Yes → p10; no → h9
	p10	13	v4	v4	Change direction
		55	v4	q 22	
		23	v2	q	
		21	v2	c24	Change $(v2)_{18}$ - $(v2)_{16}$ to a 2 or 6 for moving
		55	c4	q 14	Set mask in q

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION			REMARKS
		53	v2	v7	Move IOB word → v7
		17		v7	Move tape forward(back) 1 block
		71	v6	v4	Words index $\pm 121_{10}$ → a
		35	h4	h4	Set h4 for read
		23	v2	c24	Reset IOB word for read
		17		v2	Continue reading
		45		h9	Continue reading
	p11	17		c23	Set bias at normal and stop
		23	h10	c16	Reset ER exits
		23	h11	c18	Reset ER exits
		23	h8	c17	Reset ER exits
		21	v1	c9	Block counter = + 1
		56	0	p12	Stop
	p12	17		v2	Start reading
		45		h5	
	p13	23	v1	c9	Subtract 1 from index v1
		45		p9	

Constants

c1	1	u address modifier
c2	14 01000 00000	test constants for other IP instructions
c3	14 07000 00000	test constants for other IP instructions
c4	00 00001 77777	Masks
c5	00 00000 07777	Masks
c6	00 00036 00000	Masks

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION	REMARKS
	c7	00 00000 00007	
	c8	00 00200 00000	Rewind digit
	c9	00 00000 00001	v address modifier
	c10	00 00005 65706	Bit patterns for writing
	c11	00 00000 00167	Constant for index v3
	c12	02 00600 00000	Stop code
	c13	01 00000 00000	To change h4 from EW to ER
	c14	h7-h3	Address increment for h5
	c15	r4-h7	Address increment for h6
	c16	p9-h3	Address increment for h10
	c17	p2-p1	Address increment for h8
	c18	p13-r3	Address increment for h11
	c19	2^3 16000 00000	To reset v5
	c20	17 00000 c21-1	Bias change instruction
	c21	02 00001 70000	Bias code for high
	c22	02 00001 60000	Bias code for low
	c23	02 00001 50000	Bias code for normal
	c24	00 00002 00000	Forward - backward bit for v7
Variable Storage			
	v1	00 00000 00000	IP instruction; index; new digit
	v2	02 00000 00000	IOB word
	v3	00 00000 00167	Block index = 119_{10}
	v4	00 00000 00001	[- 1 read back] direction change

MAGNETIC TAPE INTERPRETIVE ROUTINE

Tapeworm

REGION	REGIONAL ADDRESS	INSTRUCTION	REMARKS
	v5	23 16000 00000	Bit formation for IOA fault
	v6	00 00000 00171	Block size + 1 = 121_{10}
	v7	02 00000 00001	IOB word for move 1 block

Talmadge ✓

15 May 1957

Dear USE Members:

There is an error in the Square Root Floating Point USE Routine issued by Remington Rand Univac in March 1956.

The error appears in the line labeled NOZERO. The correction to the line NOZERO should read; TP A Q, and the line NOZERO +1 should read; QT B A +129.

Previously the line NOZERO read; TP A Q +201. The first error here is that the 201 should have been written as decimal 129 instead of the octal 201. The second error is that with the V Address reading as it did, the routine in effect always gave a result with a I in the sign bit position. This is due to the 1103A Address assignment of 31,000 for Q.

This routine is presently being revised and rewritten and will be resubmitted to USE shortly.

Very truly yours,

W. E. McVicar

W. E. McVicar
Executive Secretary, USE
Remington Rand Univac
Univac Park
St. Paul 16, Minn.

WEM:lps

/ Talmadge

1. IDENTIFICATION — RRF4, SQUARE ROOT-FLOATING POINT

L. R. Turner, L. B. Kennedy - Revised 1 May 1957
Remington Rand Univac

2. PURPOSE

Given X, calculate $Y(X) = \sqrt{X}$, Floating Point

3. METHOD

a. Scaling of Argument: 1103A Floating Point Number.

b. Accuracy: $|Y(X) - \sqrt{X}| \leq 2^{-27}$ in mantissa.

c. Range of Argument: $2^{-128} \leq X < 2^{128}$

d. Derivation: an initial approximation followed by three iterations, $X_{K+1} = \frac{1}{2}(X_K + A/X_K)$. The line of coding, EVEN, is used by itself as the constant A in forming the estimate, $Y(X)_0 = \frac{A + X}{2}$; the u address of EVEN line of coding may have any value 00000 through 77777. The biased characteristic of 1 is part of the v address of line NOZERO+1.

4. USAGE

a. Calling Sequence

LOC.	OP.	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal	Return	

b. Control and Results

The argument, X, must be initially stored at location t+3 and the function Y(X) will be found at t+4, and in A_R.

c. Space Required

40 cells of instructions and constants

2 cells of working storage

d. Error Codes

The following error code is left in the accumulator on return through the error exit.

CODE	EXPLANATION
RRF4 · 2^{12}	X is negative

5. RESTRICTIONS

Argument must be in 1103A floating point number representation: from left to right sign bit, eight bit biased (200b) characteristic, and normalized twenty-seven bit mantissa. The initial contents of the accumulator and Q-register are not restored at the completion of this routine.

6. CODING INFORMATION

a. Constants

LOCATION	CONSTANT	EXPLANATION
CODE	54 54310 70000	Negative argument error code
MASKM	00 07777 77777	Mantissa mask
MASKC	77 60000 00000	Characteristic mask

b. Working Storage

2 cells labeled E through E + 1

c. Timing

2.9 ms. maximum

2.85 ms. average

PROBLEM RRF4

CODED BY Turner and Kennedy DATE 1 May '57 (Rev.)

ITEM NUMBER	LOC	OP	U	V	Comments	
,	,	SUB	, RRF4	, 40	,	$
,	,	TEMPS	, 2	, 0	,	$
,	,	INOUT	, 1	, 1	,	$
,	ENTRY ,	MJ	, 0	, START	,	$
,	ERROR ,	RJ	, DIAG + 2	, DIAG	, ERROR EXIT	$
,	EXIT ,	MJ	, 0	, FILL	, SUCCESS EXIT	$
,	Y ,	00	, FILL	, FILL	, FUNCTION	$
,	X ,	00	, FILL	, FILL	, ARGUMENT	$
,	START ,	TP	, X	, A	, ARGUMENT TO	$
,	,	TP	, A	, Y	, A AND Y	$
,	,	ZJ	, NOZERO	, EXIT	, ZERO TEST	$
,	NOZERO ,	TP	, A	, Q	, X TO Q	$
,	,	QT	, MASKM	, A + 129	, MASK MANTISSA	$
,	,	LTR	, 8	, E	, MANTISSA TO E	$
,	,	QJ	, NEG	, NONEG	, SIGN TEST	$
,	NEG ,	SP	, CODE	, 0	, ERROR CODE TO A	$
,	,	MJ	, 0	, ERROR	, TO ALARM EXIT	$
,	NONEG ,	QT	, MASKC	, A	, FORM	$
,	,	LTL	, 8	, A	, AND	$
,	,	SA	, NOZERO + 1	, 0	, STORE	$
,	,	LQ	, A	, 35	, CHARACTERISTIC	$
,	,	LTR	, 27	, Y	, IN Y	$
,	,	SJ	, ODD	, EVEN	, PARITY TEST	$
,	ODD ,	LQ	, E	, 1	,	$
,	EVEN ,	SP	, EVEN	, 1	, FORM AND	$

PROBLEM RRF4

CODED BY Turner and Kennedy DATE 1 May 1957 (Rev.)

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, SA	, E	, 0	, STORE	$
,		, LTL	, 31	, E + 1	, ESTIMATE	$
,		, SP	, E	, 28	,	$
,		, DV	, E + 1	, A	, 1ST ITERATION	$
,		, AT	, E + 1	, E + 1	,	$
,		, SP	, E	, 30	,	$
,		, DV	, E + 1	, A	, 2ND ITERATION	$
,		, AT	, E + 1	, E + 1	,	$
,		, SP	, E	, 32	,	$
,		, DV	, E + 1	, A	, 3RD ITERATION	$
,		, SA	, E + 1	, 0	,	$
,		, LTL	, 28	, E	, MANTISSA TO E	$
,		, TP	, MASKM	, Q	, MASK TO Q	$
,		, QS	, E	, Y	, PACK UP Y	$
,		, MJ	, 0	, EXIT	,	$
,	CODE	, B54	, 54310	, 70000	, ERROR CODE	$
,	MASKM	, B	, 7777	, 77777	, MANTISSA MASK	$
,	MASKC	, B77	, 60000	, 00000	, CHARACTERISTIC MASK	$
,		, ENDSUB	,	,	,	$

PROGRAM WRITE-UP

1. IDENTIFICATION SQUARE ROOT, STATED POINT

A. E. ROBERTS, JR., M. D. BERNICK-MARCH, 1956

REMINGTON-RAND UNIVAC

2. PURPOSE

Given x, this program calculates the square root of x.

3. METHOD

a. Accuracy: $\left| \sqrt{x} - F(X) \right| < 2^{-33}$

b. Range of argument: $0 \leq x \cdot 2^{33} < 2^{71}$

c. If $1 \leq x \leq 2$, then for suitable a, b, c and d,

$$y = a(x+d) + b - \frac{c}{(x+d)}$$ is an approximation of $\sqrt{x}$.

One application of the Newton-Raphson Formula with proper round-off bias gives essentially 33-bit accuracy.

4. USAGE

a. Calling Sequence

LOC.	OP.	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal	return	

b. Control and results

The argument is to be placed in the Accumulator and the function will be found in the Accumulator on return through the success exit. The argument is stored at t+3 and t+5 and the results at t+6 by the program.

c. Space Required

59 words including constants and working space.

d. Error Codes

These are left in the Accumulator on return through the error exit.

CODE	EXPLANATION
243512001001	X is negative

5. RESTRICTIONS

6. CODING INFORMATION

a. Constants: 10 locations

b. Working Storage: 4 locations

c. Timing: 3.00 ms max.
2.14 ms min.

SQUARE ROOT, STATED POINT

A. E. ROBERTS, JR., M. D. BERNICK-MARCH, 1956

SPERRY-RAND, INC.

LOCATION	OPERATION	u ADDRESS	v ADDRESS	EXPLANATION
ENTRY	MJ	Ø	START	Jump to the body of the program
ERROR	RJ	DIAG+2	DIAG	Error exit to diagnostic routine
EXIT	MJ	Ø	FILL	Success Exit
X	FILL	FILL	FILL	Most significant part of argument
X1	FILL	FILL	FILL	Least significant part of argument
Y	FILL	FILL	FILL	Function
START	LT	Ø	X	Store MSP of argument
	LT1	Ø	X1	Store L.S.P. of argument
	SJ	ERROR1	NONEG	Y Neg.
NONEG	ZJ	NOZERO	EXIT	Y Zero
NOZERO	SF	A	STORE	Scale Factor
	SP	A	Ø	$(A_L)=0;\ (A_R)=N;\ 2^{34} \leq N \leq 2^{35}$
	TP	A	STORE1	Store N
	SA	B	18	$(N+D) \div 2^{18}=F \rightarrow A_R$
	LT	Ø	STORE2	Store F
	MP	B1	STORE3	$A\cdot F \rightarrow A_R$
	AT	B2	STORE4	Store $A\cdot F+B$
	SN	B3	15	$-C\cdot 2^{15} \rightarrow A$
	DV	STORE2	A	$(-L\cdot 2^{15}) \div F = -G \rightarrow A_R$
	AT	STORE3	STORE	Store $AF+B-G=Y_1$
	SP	STORE1	32	$N\cdot 2^{32} \rightarrow A$
	SS	STORE2	Ø	$N\cdot 2^{32} - Y_1 \rightarrow A$

LOCATION	OPERATION	u ADDRESS	v ADDRESS	EXPLANATION
	DV	STORE2	A	$(N \cdot 2^{32} + Y_1) - 1 \rightarrow A_R$
	AT	STORE2	STORE2	Store $(N \cdot 2^{32} \div Y_1) - 1 + Y_1 = Y_2$
	LQ	STORE	35	$K_0 \rightarrow Q_{35}$; $K_6 \ldots K_1 \rightarrow Q_5 \ldots Q_0$
	QT	B6	A	$(K-K_0) \div 2 \rightarrow A_R$
	TV	A	RIFT	Set up V portion of shift
	TP	DONE	PROG	Set PROG for storage of result
	TP	DONE1	PROG1	Set PROG 1 for jump to exit
	TJ	B4	NOSCAR	$(K-K_0) \div 2 < 19$, jump
	TP	SCAR	PROG	Reset for scale and round
	TP	SCAR1	PROG1	
NOSCAR	QJ	RIFT	KEVEN	K odd, jump
KEVEN	MP	B5	STORE2	$2^{34} \sqrt{2} \cdot Y_2 \rightarrow A$
	SA	B7	1	$1/2 \sqrt{2} \cdot Y_2 + 1/2 \rightarrow A_R$
	LT	Ø	STORE1	$\approx 2^{17} \sqrt{N \cdot 2^{ko-1}}$ for k even
RIFT	SP	STORE2	FILL	$2^{17} \sqrt{N \cdot 2^{k-1}} \rightarrow A$
PROG	FILL	FILL	FILL	$K \leq 37$; exit
PROG1	FILL	FILL	FILL	$K > 37$; round & scale down
DONE	TP	A	Y	Store results
DONE1	MJ	0	EXIT	Jump to exit
SCAR	SA	B8	Ø	Instruction for round and scale down
SCAR1	LT	Ø	A	$(A_L) = 0$
ERROR1	SP	CODE	Ø	Error code into A_R
	MJ	Ø	ERROR	
B	B	264767031361		D=24, 290, 062, 513
B1	B	65324		A=27, 349

LOCATION	OPERATION	u ADDRESS	v ADDRESS	EXPLANATION
B2	B	114534644516		B=10, 291, 988, 814
B3	B	330657140271		C=29, 104, 062, 651
B4	B	23		**TJ constant**
B5	B	265011714640		24, 296, 004, 000
B6	B	77		Mask for scale factor
B7	B	200000000000		2^{34}
B8	B	377777777777		$2^{35} - 1$
CODE	B	243512001001		
STORE	Ø	Ø	FILL	
STORE1	FILL	FILL	FILL	
STORE2	FILL	FILL	FILL	
STORE3	FILL	FILL	FILL	

PROGRAM WRITE-UP

1. IDENTIFICATION SINE x, STATED POINT

A. E. ROBERTS, JR., M. D. BERNICK - MARCH, 1956

REMINGTON RAND UNIVAC

2. PURPOSE

Given x, this program computes Sine x.

3. METHOD

a. Accuracy

b. Range of argument: $|x| \cdot 2^{32} < 2^{71}$

c. Derivation is obtained from

$\sin \frac{\pi}{2} x = 1-2 \sin^2 \frac{\pi}{4} (x-1)$

Polynomial approximation for $\frac{\sin \frac{\pi}{4} n}{n}$ as a function of

$(-n)^2$ for $-1 \leq n \leq 1$ is derived from the Chebyshev expansion:

$$\sin \frac{\pi}{4} x = 2 \sum_{k=1}^{\sigma} (-1)^k J_{2k-1} \left(\frac{\pi}{4}\right) T_{2k-1}(x).$$

x is entered and multiplied by $\frac{2}{\pi}$ so that

$\frac{2}{\pi} x = y.$ The routine then computes

$\sin \frac{\pi}{2} y = \sin x.$

4. USAGE

a. Calling Sequence - Standard.

b. Control and results

The argument is to be placed in the accumulator; the function will be found in the accumulator upon completion of the routine. The routine also stores the argument in t+3 and t+4 and the function in t+5.

c. Space required: 54 words including constants and working storage.

d. Error codes - none.

5. RESTRICTIONS

6. CODING INFORMATION

a. Constants: 12 locations.

b. Working Storage: 1 location.

c. Timing

SINE x, STATED POINT

A. E. ROBERTS, JR. M. D. BERNICK - MARCH, 1956

REMINGTON RAND UNIVAC

LOCATION	OPERATION	u ADDRESS	v ADDRESS	EXPLANATION
ENTRY	MJ	Ø	START	Jump to body of program
ERROR	RJ	DIAG+2	DIAG	Error exit to diagnostic routine
EXIT	MJ	Ø	FILL	Success exit
X	FILL	FILL	FILL	Most significant part of argument
X1	FILL	FILL	FILL	Least significant part of argument
Y	FILL	FILL	FILL	Function
START	LT	Ø	X	Store M.S.P. of argument
	LT1	Ø	X1	Store L.S.P. of argument
	DV	B	Q	$R \cdot 2^{32} \rightarrow A;\ 0 \leq R < 2\pi$
	MP	A	B1	$R \cdot \frac{2}{\pi} \cdot 2^{67} = R_1 \cdot 2^{67} \rightarrow A$
	LT	3	Q	$R_1 \cdot 2^{34} \rightarrow Q$
	TP	PRO	PRO1	Set PRO1 for positive function
	QJ	NEG	POS	Test for positive or negative functio
NEG	TP	PRO2	PRO1	Reset PRO1 for negative function
POS	QT	B2	A	Extract $2^{35} \cdot (x \bmod 1) = 2^{35} x^* \rightarrow A_R$
	QJ	2OR4	1OR3	Test for quadrant
1OR3	SS	B3	Ø	$\left\lvert \sin \frac{\pi}{2} x \right\rvert = \cos \pi/2\ (x^*-1) \cdot (x^*-1) \cdot 2^{35} \rightarrow A$
2OR4	MP	A	Q	$2^{70}\ (x^*-1)^2 \rightarrow A$
	SA	B3	Ø	Round & scale down $\left[2^{34}(x^*-1)^2+1/2 \right]$
	LT	Ø	A	(cont.)
	TN	A	STORE	$-\left[2^{34}(x^*-1)^2+1/2 \right] = -N \rightarrow STORE$
	TU	PRO2	PRO3	Set u address of PRO3 for i=0

LOCATION	OPERATION	uADDRESS	vADDRESS	EXPLANATION
	TP	B4	Q	$A_4 \rightarrow Q$, $A_4 = P_0$
PR04	MP	Q	STORE	With P_i in Q, i=0,1,2,3; $-N \cdot P_i \rightarrow A$
	SA	B3	Ø	Round & scale down $\left[-N \cdot P_i \cdot 2^{-36}+1/2\right] \rightarrow A_R$
	LT	Ø	A	
PR03	AT	FILL	Q	$\left[A_{3-i}-N \cdot P_i \cdot 2^{-36}+1/2\right] = P_{i+1} \rightarrow Q$
	RA	PR03	B5	$i+1 \rightarrow i$ in PR03
	TJ	PR05	PR04	Test end point
	MP	Q	Q	$P_4 \approx (2^{35} \sin\frac{\pi}{4}y)\div$ y in $Q \cdot P_4^2 \rightarrow A$
	SA	B3	Ø	Round & scale down $2^{-36}P_4^2+1/2$ A_R
	LT	Ø	A	
	MP	A	STORE	$-\left[2^{-36}P_4^2+1/2\right] \cdot N \approx -2^{68} \sin^2 \frac{\pi}{4}y \rightarrow A$
	SS	B6	37	Round & scale down $\approx -2^{-33} \sin^2 \frac{\pi}{4}y \rightarrow A_R$
PR0	TN	A	A	$\approx 2^{32}\ 2 \sin^2 \frac{\pi}{4}y \rightarrow A$
	ST	B7	A	$\approx 2^{32}\ (2 \sin^2 \frac{\pi}{4}y-1) \rightarrow A$
PR01	FILL	FILL	FILL	For + function complement A
PR06	TP	A	Y	Store function
	MJ	Ø	EXIT	Jump to exit
PR02	MJ	B8	PR06	for - function, don't complement A
PR05	AT	B6	Q	End point test threshold.
B2	B	377777777774		Mask for $(Q_{34} \cdots Q_2)$
B3	B	400000000000		2^{35}
B4	B	12265046		A_4 = 2,714,150
B8	B	462621024		A_3 = 80,421,396
	B	12146566440		A_2 = 1,369,107,744
	B	122535716221		A_1 = 11,097,578,641

LOCATION	OPERATION	uADDRESS	vADDRESS	EXPLANATION
	B	311037552420		A_0 = 26,986,075,408
B6	B	200000000000		2^{34}
B7	B	40000000000		2^{32}
B5	B	100000		Advance of u
B	B	311037552421		$2\pi \times 2^{32}$
B1	B	242763015554		$2/_{\pi} \times 2^{35}$
STORE	FILL	FILL	FILL	

1. IDENTIFICATION

RRF2, ARCTAN X STATED POINT
P. Johnson, M. Bernick - Revised 15 May 1957
Remington Rand Univac

2. PURPOSE

Given X, Compute Y(X) = arctan X

3. METHOD

a. Accuracy: $|Y(X) - \arctan X| \leq 2^{-25}$

b. Range of Argument: $|X| \leq 1$

c. Scaling: $X \cdot 2^{33}$, $Y(X) \cdot 2^{33}$

d. Derivation: Y(X) is computed using the polynomial approximation

$Y(X) \approx \sum_{i=1}^{7} C_{2i+1} X^{2i+1}$ given in Rand Sheet No. 13.

4. USAGE

a. Calling Sequence

LOC.	OP.	U ADDR.	V ADDR.
r	RJ	t + 2	t
r+1	Normal	Return	

b. Control and Results

The argument, X, must be initially stored at t+4; the result, Y(X), will be found at t+3, and in A_R.

c. Space Required

54 cells of instructions and constants

1 cell of working storage

d. Error Codes

The following error codes are left in the accumulator on return through the error exit.

CODE	EXPLANATION
$RRF2 \cdot 2^{12} + 1$	$X < -1$
$RRF2 \cdot 2^{12} + 2$	$X > 1$

5. RESTRICTIONS

The argument must be in radians, within the stated range, and scaled 2^{33}.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
B	677777777777	Lower limit on X: -2^{33}
B1	100000000001	Upper limit on X: $2^{33} + 1$
B2	040000000000	2^{32} for rounding
B3	573120142744	$C_{15} \cdot 2^{42}$
B4	263054507277	$C_{13} \cdot 2^{40}$
B5	432774360726	$C_{11} \cdot 2^{39}$
B6	305357575005	$C_9 \cdot 2^{38}$
B7	561447164514	$C_7 \cdot 2^{37}$
B8	314201226657	$C_5 \cdot 2^{37}$
B9	525263620355	$C_3 \cdot 2^{36}$
B10	377777723167	$C_1 \cdot 2^{35}$
CODE	545431050001	Error code for $X < -1$
CODE 1	545431050002	Error code for $X > 1$

b. Working Space

1 cell labeled STORE

c. Timing

Average 3.19 mls.

Maximum 3.2 mls.

PROBLEM RRF2

CODED BY Johnson, Bernick DATE REV. 15 May '57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,	,	SUB ,	RRF2 ,	54 ,		$
,	,	TEMPS,	1 ,	0 ,		$
,	,	INOUT,	1 ,	1 ,		$
,	ENTRY ,	MJ ,	0 ,	START ,		$
,	ERROR ,	RJ ,	DIAG+2 ,	DIAG ,	ERROR EXIT	$
,	EXIT ,	MJ ,	0 ,	FILL ,	SUCCESS EXIT	$
,	Y ,	00 ,	FILL ,	FILL ,	FUNCTION	$
,	X ,	00 ,	FILL ,	FILL ,	ARGUMENT	$
,	START ,	TP ,	X ,	A ,	ARGUMENT TO A	$
,	,	TJ ,	B ,	ERROR 1 ,	TEST FOR	$
,	,	TJ ,	B1 ,	OK ,	ARGUMENT OUT	$
,	,	MJ ,	0 ,	ERROR 2 ,	OF RANGE	$
,	OK ,	MP ,	X ,	X ,	FORM ROUND AND	$
,	,	SA ,	B2 ,	3 ,	STORE X SQRD	$
,	,	LTL ,	0 ,	STORE ,	SCLD 33 IN TEMP	$
,	,	MP ,	B3 ,	STORE ,		$
,	,	LTL ,	1 ,	A ,	EVALUATE	$
,	,	AT ,	B4 ,	Q ,		$
,	,	MP ,	Q ,	STORE ,	POLYNOMIAL	$
,	,	LTL ,	2 ,	A ,		$
,	,	AT ,	B5 ,	Q ,	EXPRESSION	$
,	,	MP ,	Q ,	STORE ,		$
,	,	LTL ,	2 ,	A ,	FOR	$

PROBLEM RRF2

CODED BY Johnson, Bernick DATE Rev. 15May '57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
		AT	B6	Q		$
		MP	Q	STORE	ARCTAN X	$
		LTL	2	A		$
		AT	B7	Q	B_Y	$
		MP	Q	STORE		$
		LTL	3	A	NESTING	$
		AT	B8	Q		$
		MP	Q	STORE	METHOD	$
		LTL	2	A		$
		AT	B9	Q	.	$
		MP	Q	STORE		$
		LTL	2	A		$
		AT	B10	Q		$
		MP	Q	X		$
		LTL	1	A		$
		TP	A	Y	STORE FUNCTION	$
		MJ	0	EXIT	TO SUCCESS EXIT	$
	ERROR 1	SP	CODE	0	ERROR CODE TO A	$
		MJ	0	ERROR	TO ERROR EXIT	$
	ERROR 2	SP	CODE 1	0	ERROR CODE TO A	$
		MJ	0	ERROR	TO ERROR EXIT	$
	B	B67	77777	77777	LOWER LIMIT	$

PROBLEM RRF2

CODED BY Johnson, Bernick DATE REV. 15 May '57

ITEM NUMBER		LOC		OP		U		V		COMMENTS	
	,	B1	,	B10	,	00000	,	00001	,	UPPER LIMIT	$
	,	B2	,	B04	,	00000	,	00000	,	ROUND BIT	$
	,	B3	,	B57	,	31201	,	42744	,	C15 SCLD 42	$
	,	B4	,	B26	,	30545	,	07277	,	C13 SCLD 40	$
	,	B5	,	B43	,	27743	,	60726	,	C11 SCLD 39	$
	,	B6	,	B30	,	53575	,	75005	,	C9 SCLD 38	$
	,	B7	,	B56	,	14471	,	64514	,	C7 SCLD 37	$
	,	B8	,	B31	,	42012	,	26657	,	C5 SCLD 37	$
	,	B9	,	B52	,	52636	,	20355	,	C3 SCLD 36	$
	,	B10	,	B37	,	77777	,	23167	,	C1 SCLD 35	$
	,	CODE	,	B54	,	54310	,	50001	,	ERROR	$
	,	CODE 1	,	B54	,	54310	,	50002	,	CODES	$
	,		,	ENDSUB	,		,		,		$

Talmadge

THIS IS A USE PROGRAM

BAT001 PRINT - OFF-LINE - STATED DECIMAL DATA
by D I Cook, Boeing Airplane Co, Dec 1956

A PURPOSE

The purpose of this program is to prepare a magnetic tape on the 1103A computer to print a block of consecutive words of memory as stated decimal data on the Univac High Speed Printer. The information is written on Uniservo number two in the fixed block mode.

B METHOD

The data to be printed are scaled to proper binary fractions (with the binary point between the sign and the most significant bits) from the scaling information supplied by the programmer as control information. Repeated multiplication by ten is performed on the scaled word to obtain successive decimal digits. The decimal point is inserted in the appropriate position and a minus sign is inserted after the least significant digit if the word is negative. Zeros preceeding the most significant digit are not converted to XS3 code and, hence, are not printed by the high speed printer.

Each column to be printed may contain twelve characters including the decimal point and sign. If the word is too large to be printed in the form indicated by the scaling information the decimal point is shifted to the right and the low order digits are truncated. An eleven digit integer is printed without a decimal point.

A "Fast Feed I" symbol is inserted as the first character of the first blockette and every sixtieth blockette causing the data to be printed at 60 lines per page. A "Printer Stop" symbol is inserted as the first character of the blockette immediately following the last blockette to be printed in order to stop the printer after completing the printing.

C USAGE

1. The following instruction is written by the programmer to enter this program where "t" is the location of the first instruction of this program and "r" is a location in the main program:

Loc	Op	U	V	Explanation
r	RJ	t+2	t	Jump to the first instruction of this program.
r+1	--	-	-	Control is returned to this location following the successful execution of this program.

C USAGE (cont'd)

2. The following control data must be stored within this program prior to transferring control to location t.

Loc	Op	U	V	Explanation
t+3	N	L	M	Format Control. N is the number of columns to be printed where $0 \leq N \leq 8$. L is the location of the first data word to be printed. It is assumed that successive words are in consecutive locations. M is the number of lines to be printed and must be chosen such that $L+M \cdot N$ is a legal storage location.
t+4	-	D_1	B_1	Scale Indices. D_i is the number of decimal places to be printed to the right of the decimal point for all data in the ith column where $0 \leq D_i \leq 10$. B_i is the number of binary places to the right of the binary point for all data to be printed in the ith column where $0 \leq B_i \leq 35$
t+5	-	D_2	B_2	
t+6	-	D_3	B_3	
.		.	.	
.		.	.	
.		.	.	
t+11	-	D_8	B_8	

3. When using this program Uniservo number two must be ready to receive written information. The tape is not rewound before or after printing so that additional information may be written on the tape.

4. The following is the normal storage assignment for this program. It should be noted that this assignment may be altered by the compiler.

	Starting Location		Number of words	
Item	Dec	Oct	Dec	Oct
Total Program	t	t	301	455
Without Erasable Storage	t	t	168	250
Subject to Address Mod.	t	t	143	217
Program Constants	t+143	t+217	25	31
Erasable Storage	t+168	t+250	133	205

5. No error checks are made within this program and no error codes are used. Failure to comply with the restrictions imposed on the Format Control Word in location t+3 may result in an MCT fault.

C USAGE (cont'd)

6. The printed format produced by this program consists of N columns of 12 characters each with 3 blank spaces between columns. The 12 characters within a column consist of 10 (or less) decimal digits, a decimal point and a minus sign (if the number is negative) or a blank space (if the number is positive). Eleven digit integers are printed without a decimal point and the 12 characters then consist of 11 decimal digits and a minus sign or a blank space.

D RESTRICTIONS

1. This program will operate on the standard "minimum 1103A" computer as established by USE.

2. This program is self-contained and no other programs are used.

3. This program will print data contained anywhere in the addressable memory of the computer other than that space occupied by the program itself. The data are assumed to be in stated binary form in consecutive memory locations.

4. The tape produced by this program is in fixed block form in blockettes of 120 characters each and a density of 128 lines per inch suitable for printing on the Univac High Speed Printer.

5. The standard USE 120-120 printer board is used to print the data from the tape.

E CODING INFORMATION

1. This program uses 133 words of erasable storage. In normal usage this block of erasable storage begins with location $_{10}(t+168) = _{8}(t+250)$.

2. This program will write a printer tape at approximately 400 blockettes per minute.

BA-T001 PRINT - OFF-LINE - STATED DECIMAL DATA - FIXED BLOCK
THIS IS A USE PROGRAM

```
        ,B      ,+61                  , DOUBLE LENGTH CHECK SUM WITH
        ,B      ,+4716354C6515        ,    PROGRAM AT 1000
        ,X      ,+143.0               , INSTRUCTIONS SUBJECT TO ADDRESS MOD
        ,X      ,+25.0                , PROGRAM CONSTANTS
        ,X      ,+133.0               , ERASABLE STORAGE
        ,X      ,+0.0                 , NUMBER OF CONTROL WORDS
        ,X      ,+0.0                 , NUMBER OF PROGRAM RESULTS

ENTRY   ,MJ     ,0        ,2A         ,
ERROR   ,RJ     ,DIAG+2   ,DIAG       , ERROR EXIT
EXIT    ,MJ     ,0        ,FILL       , SUCCESS EXIT
CTRL    ,FILL   ,FILL     ,FILL       , CONTROL WORD
S0      ,FILL   ,FILL     ,FILL       , SCALE INDICES
S1      ,FILL   ,FILL     ,FILL       ,
S2      ,FILL   ,FILL     ,FILL       ,
S3      ,FILL   ,FILL     ,FILL       ,
S4      ,FILL   ,FILL     ,FILL       ,
S5      ,FILL   ,FILL     ,FILL       ,
S6      ,FILL   ,FILL     ,FILL       ,
S7      ,FILL   ,FILL     ,FILL       ,
2A      ,TU     ,CTRL     ,6A         , LOCATION OF FIRST WORD
        ,TU     ,CTRL     ,6E4        ,
        ,TP     ,ZERO     ,LINES      ,
        ,TV     ,CTRL     ,LINES      , NUMBER OF LINES TO PRINT
        ,TP     ,ZERO     ,PAGE       , START NEW PAGE
        ,IJ     ,LINES    ,4A         ,
        ,MJ     ,0        ,EXIT       , EXIT WHEN NO LINES TO PRINT
4A      ,RP1    ,120      ,4A2        ,
4A1     ,TP     ,ZERO     ,E          , CLEAR STORAGE FOR PRINT IMAGE
4A2     ,TV     ,4A1      ,FIRST      , SET LOCATION OF FIRST LINE OF SIX
        ,TP     ,V5       ,BLOCK      , PREPARE TO PRINT SIX LINES
5A      ,SP     ,CTRL     ,6          ,
        ,LT     ,0        ,COLS       , NUMBER OF COLUMNS TO PRINT PER LINE
        ,IJ     ,COLS     ,5A1        ,
        ,MJ     ,0        ,EXIT       , PRINT NO COLUMNS
5A1     ,TJ     ,V8       ,5A2        , TEST FOR MORE THAN 8 COLUMNS
        ,TP     ,V7       ,COLS       , PRINT 8 COLUMNS
5A2     ,TU     ,US0      ,6A1        , LOCATION OF FIRST SCALE INDEX
        ,TV     ,FIRST    ,6F1        ,
        ,TV     ,FIRST    ,6F2        ,
        ,TV     ,FIRST    ,6F4        ,
        ,TV     ,FIRST    ,6F5        ,
        ,TV     ,FIRST    ,6F6        ,
        ,RA     ,6F5      ,D0         ,
        ,RA     ,6F6      ,V2         ,
        ,RA     ,6F1      ,V3         ,
        ,RA     ,6F2      ,V4         ,
        ,TP     ,D0       ,ALTER      ,
```

CALC	Cook	12-56	REVISED	DATE	PRINT - OFF-LINE - STATED DECIMAL DATA FIXED BLOCK	
CHECK						
APR					BOEING AIRPLANE COMPANY SEATTLE 14, WASHINGTON	T001
APR						
CONTRACT NO.						PAGE 4

```
6A      ,TM     ,FILL    ,WORD     , GET NEXT WORD TO PRINT
        ,TP     ,ZERO    ,DEC      ,
        ,TP     ,ZERO    ,BIN      ,
6A1     ,SP     ,FILL    ,C        ,
        ,TU     ,A       ,DEC      , SAVE DECIMAL INDEX = D
        ,TV     ,A       ,BIN      , SAVE BINARY INDEX = B
        ,SP     ,DEC     ,21       ,
        ,LT     ,C       ,SCALE    , SAVE D FOR INDEXING
        ,TP     ,9A      ,6B       , PREPARE SHIFT COMMAND
        ,PS     ,6B      ,BIN      , U ADDRESS = (36-B)
        ,SP     ,D1      ,15       ,
        ,ST     ,DEC     ,DEC      ,
        ,AT     ,9A1     ,6B2      , U ADDRESS = (D10-D)
        ,TP     ,D1      ,BIN      , DO ELEVEN CHARACTERS
        ,LQ     ,DEC     ,21       ,
6B      ,SP     ,WORD    ,35       ,
        ,LT     ,C       ,UPPER    ,
        ,LT1    ,C       ,LOWER    ,
        ,TP     ,ZERO    ,G        ,
6B1     ,SP     ,UPPER   ,C        ,
6B2     ,TJ     ,D10     ,6C       ,
        ,RA     ,DEC     ,D0       , POWER OF TEN NOT LARGE ENOUGH
        ,RA     ,6B2     ,U1       ,
        ,IJ     ,SCALE   ,6B1      ,
        ,SP     ,UPPER   ,C        ,
        ,DV     ,D10     ,G        ,
        ,LT     ,C       ,UPPER    ,
        ,RA     ,Q       ,V3       ,
        ,TP     ,D2      ,DEC      , SUPPRESS PRINTING OF DECIMAL POINT
6C      ,SP     ,UPPER   ,36       ,
        ,SA     ,LOWER   ,71       ,
        ,TP     ,Q       ,LOWER    ,
6C1     ,TP     ,ZERO    ,UPPER    ,
        ,TU     ,6B2     ,6C2      , LOCATION OF DIVISOR
6C2     ,DV     ,FILL    ,G        ,
        ,RA     ,C       ,D0       , ROUND QUOTIENT
        ,CJ     ,6C3     ,6C4      , TEST FOR OVERFLOW
6C3     ,TP     ,V4      ,LOWER    , INSERT HIGH DIGIT OF 1
        ,IJ     ,BIN     ,6C4      ,
6C4     ,TP     ,Q       ,WORD     , SAVE PROPER FRACTION
6D      ,IJ     ,DEC     ,6F       ,
        ,TP     ,POINT   ,G        , INSERT POINT
        ,TP     ,D2      ,DEC      , SUPPRESS ANOTHER POINT
        ,MJ     ,C       ,6E1      ,
6E      ,SP     ,WORD    ,2        ,
        ,SA     ,WORD    ,1        , MULTIPLY BY TEN
        ,TP     ,A       ,WORD     ,
        ,LT     ,C       ,G        , SAVE DIGIT IN G
6E1     ,SP     ,UPPER   ,36       ,
        ,SA     ,LOWER   ,6        ,
```

CALC	Cook	12-56	REVISED	DATE	PRINT - OFF-LINE - STATED DECIMAL DATA FIXED BLOCK	
CHECK						
APR					BOEING AIRPLANE COMPANY	T001
APR					SEATTLE 14, WASHINGTON	
CONTRACT NO.						PAGE 5

```
        ,SA     ,0        ,C        ' ADD NEXT DIGIT
        ,ZJ     ,6E2      ,6E3      ' SUPPRESS ZEROS
6E2     ,AT     ,V3       ,LOWER    ' CONVERT TO XS3
        ,LT     ,C        ,UPPER    '
6E3     ,IJ     ,BIN      ,6D       ' DO ANOTHER DIGIT
        ,SP     ,UPPER    ,36       '
        ,SA     ,LOWER    ,6        '
6E4     ,TP     ,FILL     ,G        '
        ,QJ     ,6E5      ,6E6      '
6E5     ,SA     ,V2       ,C        ' INSERT MINUS SIGN
6E6     ,LT     ,C        ,UPPER    '
        ,TP     ,A        ,LOWER    '
6F      ,IJ     ,ALTER    ,6E3      '
6F1     ,TP     ,UPPER    ,FILL     ' MOVE CONVERTED WORD TO PRINT AREA
6F2     ,TP     ,LOWER    ,FILL     '
        ,RA     ,6F1      ,V5       '
        ,RA     ,6F2      ,V5       '
        ,RA     ,6F4      ,V5       '
        ,RA     ,6F5      ,V5       '
        ,RA     ,6F6      ,V5       '
        ,TP     ,D0       ,ALTER    '
        ,MJ     ,0        ,7A       '
6F3     ,SP     ,UPPER    ,18       '
6F4     ,LT     ,0        ,FILL     ' MOVE FIRST 3 DIGITS TO PRINT AREA
        ,SP     ,A        ,18       '
        ,SA     ,LOWER    ,18       '
6F5     ,LT     ,0        ,FILL     ' MOVE NEXT 6 DIGITS
6F6     ,TP     ,A        ,FILL     ' MOVE LAST 3 DIGITS
7A      ,RA     ,6F4      ,U1       ' LOCATION OF NEXT WORD
        ,TU     ,6F4      ,6A       '
        ,RA     ,6A1      ,U1       ' LOCATION OF NEXT INDEX
        ,IJ     ,COLS     ,6A       ' DO ANOTHER WORD
        ,RA     ,FIRST    ,V20      ' LOCATION OF NEXT LINE
        ,IJ     ,BLOCK    ,7A1      '
        ,MJ     ,0        ,8A       ' WRITE BLOCK ON TAPE
7A1     ,IJ     ,LINES    ,5A       ' DO ANOTHER LINE
FIRST   ,TP     ,STOP     ,FILL     ' STOP PRINTER
8A      ,IJ     ,PAGE     ,8A1      '
        ,RA     ,E        ,FFI      ' RESTORE PAPER
        ,TP     ,V9       ,PAGE     '
8A1     ,EF     ,0        ,WRITE    ' WRITE TAPE IN FIXED BLOCK MODE
        ,RP1    ,120      ,8A2      '
        ,EW1    ,C        ,E        '
8A2     ,IJ     ,LINES    ,4A       '
        ,TP     ,BLOCK    ,A        '
        ,ZJ     ,EXIT     ,8A3      '
8A3     ,EF     ,C        ,WRITE    ' WRITE TAPE IN FIXED BLOCK MODE
        ,EW1    ,C        ,STOP     ' STOP PRINTER
        ,RP     ,119      ,EXIT     '
        ,EW1    ,C        ,ZERO     '
```

CALC	Cook	12-56	REVISED	DATE	PRINT - OFF-LINE - STATED DECIMAL DATA FIXED BLOCK	
CHECK						
APR						
APR					BOEING AIRPLANE COMPANY SEATTLE 14, WASHINGTON	T001
CONTRACT NO.						PAGE 6

```
9A      ,SP     ,WORD     ,36     ,
9A1     ,TJ     ,D0       ,6C     ,
USO     ,TP     ,SO       ,SO     ,
ZERO    ,B      ,+0               ,
D0      ,B      ,+1               , POWERS OF TEN
D1      ,X      ,+10.0            ,
D2      ,X      ,+100.0           ,
D3      ,X      ,+1000.0          ,
D4      ,X      ,+10000.0         ,
D5      ,X      ,+100000.0        ,
D6      ,X      ,+1000000.0       ,
D7      ,X      ,+10000000.0      ,
D8      ,X      ,+100000000.0     ,
D9      ,X      ,+1000000000.0    ,
D10     ,X      ,+10000000000.0   ,
U1      ,B      ,+100000          ,
V2      ,B      ,+2               ,
V3      ,B      ,+3               ,
V4      ,B      ,+4               ,
V5      ,B      ,+5               ,
V7      ,B      ,+7               ,
V8      ,X      ,+8.0             ,
V9      ,X      ,+9.0             ,
V20     ,X      ,+20.0            ,
POINT   ,B      ,+17              ,
STOP    ,B      ,+600000000000    ,
EFI     ,B      ,+370000000000    ,
WRITE   ,B      ,+200646200000    ,
        ,ENDSUB,

                 ERASEABLE STORAGE

LINES   ,B      ,+0               , NUMBER OF LINES TO PRINT
PAGE    ,B      ,+0               , NUMBER OF LINES PER PAGE
BLOCK   ,B      ,+0               , NUMBER OF LINES PER BLOCK
COLS    ,B      ,+0               , NUMBER OF COLUMNS PER LINE
ALTER   ,B      ,+0               , FLIP - FLOP
WORD    ,B      ,+0               , WORD TO BE CONVERTED
DEC     ,B      ,+0               , NUMBER OF DIGITS TO LEFT OF POINT
BIN     ,B      ,+0               , NUMBER OF CHARACTERS PER WORD
SCALE   ,B      ,+0               , NUMBER OF DIGITS TO RIGHT OF POINT
UPPER   ,B      ,+0               , FIRST 6 CHARACTERS OF CONVERTED WORD
LOWER   ,B      ,+0               ,  LAST 6 CHARACTERS OF CONVERTED WORD
E       ,B      ,+0               , PRINT AREA - TEMPORARY STORAGE FOR
                                       LINE IMAGES - 120 WORDS
```

USE Subroutine

I. Identification

R-W EXP-2, Stated point Exponential, Malcolm Perry,

22 March 1956 - Ramo-Wooldridge Corporation

II. Purpose

Given $2^{35}x$ this routine computes $2^{35}\ e^{x}$.

III. Accuracy, Range, Method

a. Truncation error of the polynomial is $< 2^{-35}$.

b. Range. $-(\ln 2)\ (2^{35}+2^{-1}) < x < 34.5 \ln 2$

c. Method. The routine finds q, an integer such that

$$x = q\ (\ln 2) + r$$

where $|r| \leq \frac{\ln 2}{2}$

This gives $e^{x} = (e^{\ln 2})^{q} \cdot e^{r} = 2^{q} \cdot e^{r} = (2^{q+1}) \cdot \frac{(e^{r})}{2}$

Since the factor 2^{q+1} is easily applied by shifting, it is only necessary to calculate the quantity $e^{r}/2$. This is accomplished by a 7th order approximating polynomial where the domain of r is

$$-\frac{\ln 2}{2} \leq r \leq \frac{\ln 2}{2}$$

This polynomial was obtained with the aid of routine CVF-0. The coefficients of the polynomial are listed in the accompanying code listing. The maximum discrepancy between the function $e^{r}/2$ and the polynomial, in the interval stated above, is $.75 \times 2^{-35}$.

The error in the machine's approximation to e^{x} is bounded in all cases by

$$\left[(11.3 + .7\ |x|\ )\ e^{x} + 1\right] \cdot 2^{-35}$$

That is

$$\left|(A)_f - e^x \cdot 2^{35}\right| < (11.3 + .7\,|x|)\, e^x + 1$$

Most of the error is due to round-off within the routine. The actual error is usually less than the bound stated here.

IV. Usage

a. Calling sequence

LOC	OP	u	v
r	RJ	t+2	t+2
r+1	Normal return		

b. Control and Results

2^{35} x must be in A upon entry to the routine; 2^{35} e^x is left in A upon normal return.

c. Space required

28 instructions, 15 constants, 3 temporaries

d. Alarm conditions

If x falls in the interval

$$34.5\,(\ln 2) < x < (\ln 2)\,(2^{35} - 2^{-1})$$

then the alarm routine is entered. This alarm condition is equivalent to

$$e^x \geqslant 2^{34.5}$$

and, hence, in terms of the scaled result

$$e^x \cdot 2^{35} \geqslant 2^{69.5}$$

This value will nearly overflow in A and therefore becomes an upper limit.

For $x \leqslant -(\ln 2)\,(2^{35} + 2^{-1})$

or $x \geqslant (\ln 2)\,(2^{35} - 2^{-1})$

a divide overflow will occur at cell EX03 of the subroutine.

V. Except for the limits on x, there are no restrictions on the use of this routine.

VI. Coding Information

a. Constants

LOC	CONSTANT
+28	$(\frac{1}{2}\ln 2)2^{34}$
+29	$(\ln 2)2^{35}$
+38	36
+39	2^{15}
+40	6
+41	1

RW EXP-2
3-26-56

```
EX00   MJ 00000 0EX03
EX01   00 00000 00000
EX02   MJ 00000 00000
EX03   AT 1EX00 A0000        HLF LN 2
EX04   DV 1EX01 2EX00        LN 2  EXP 1
EX05   ST 1EX00 Q0000        ARG         35
EX06   TP 1EX02 2EX01        C 7         34
EX07   TU 0EX06 0EX12
EX08   TP 1EX12 2EX02        INDEX
EX09   RA 0EX12 1EX11
EX10   MP Q0000 2EX01                    69
EX11   LT 00001 A0000                    34
EX12   AT 00000 2EX01
EX13   IJ 2EX02 0EX09
EX14   RA 2EX00 1EX13
EX15   SJ 0EX22 0EX16
EX16   TJ 1EX10 0EX19
EX17   TP 1EX14 00000        ALARM TAG
EX18   MJ 00000 0EX01
EX19   TV A0000 0EX20
EX20   LA 2EX01 00000
EX21   MJ 00000 0EX02
EX22   SA 1EX10 00000        36
EX23   SJ 0EX26 0EX24
EX24   TV A0000 0EX25
EX25   LA 2EX01 00000        36 LESS EXP
EX26   LT 00000 A0000
EX27   MJ 00000 0EX02
```

1EX00	06	93147	18056	-01	34	HALF LN 2
1EX01	06	93147	18056	-01	35	LN 2
1EX02	01	99243	65600	-04	34	C7
1EX03	01	39485	76760	-03	34	C6
1EX04	08	33324	84740	-03	34	C5
1EX05	04	16662	18354	-02	34	C4
1EX06	01	66666	66994	-01	34	C3
1EX07	05	00000	01077	-01	34	C2
1EX08	09	99999	99997	-01	34	C1
1EX09	09	99999	99996	-01	34	CO
1EX10	00	00000	00036			
1EX11	00	00001	00000			
1EX12	00	00000	00006			
1EX13	00	00000	00001			
1EX14	00	00000	00000			ALARM TAG

USE Subroutine

I. Identification

R-W TNI-1, Floating point Arctangent, Malcolm Perry,

20 March 1956, Ramo-Wooldridge Corporation.

II. Purpose

Given x in floating point this routine computes arctan x in floating point.

III. Accuracy, Range, Method

a. The magnitude of the error is $< 2^{-25}$

b. Any x expressible in floating point is permitted.

c. Method. If $x \geq 1$, the identity arctan x = $\pi/2$ - arctan 1/x is used. Rand Polynomial 13 is evaluated using x or 1/x.

IV. Usage

a. Calling sequence

LOC	OP	u	v
r	RJ	t+2	t
r+1	Normal return		

b. Control and Results

x in floating point must be in A upon entry; arctan x in radians in floating point will be in A upon return.

c. Space required.

46 instructions, 18 constants, 5 temporaries

d. There are no alarm conditions.

V. Coding Information

a. Constants

LOC	CONSTANT
+46	$2^{34}\pi/2$
+55	00 07777 77777
+56	37 70000 00000
+60	6
+61	2^{27}
+62	2^{15}
+63	2^{16}

```
T00   MJ 00000 00T03        ENTRY
T01   00 00000 00000        ALARM EXIT
T02   MJ 00000 00000        NORMAL EXIT
T03   TP 00T45 00T44        ASSUME POS
T04   SJ 00T05 00T06
T05   TP 00T13 00T44        IF NEGATIVE
T06   TM A0000 Q0000
T07   QT 01T09 02T01        MANTISSA
T08   CC 02T00 A0000        CLEAR SUM
T09   QT 01T10 02T02        EXP PLUS 200
T10   LT 00009 A0000
T11   ST 01T11 Q0000        LESS 201
T12   SJ 00T16 00T13
T13   TN A0000 A0000
T14   TP 01T00 02T00        PI OVER 2
T15   TP 01T12 02T02        EXP 200
T16   SA 01T13 00000        PLUS 36
T17   SJ 00T27 00T18
T18   QJ 00T19 00T23
T19   TV A0000 00T20
T20   SP 02T01 00000
T21   LT 00000 Q0000        X FIXED   26
T22   MJ 00000 00T26
T23   TV A0000 00T24
T24   SN 01T17 00000
T25   DV 02T01 02T01        RECIP X   26
T26   MP Q0000 Q0000
T27   LT 00014 Q0000        SQUARED   30
```

```
T28   TP 01T01 02T04              C15
T29   TU 00T28 00T32
T30   TP 01T14 02T03
T31   RA 00T32 01T16
T32   LA 00000 20025              C          60
T33   MA Q0000 02T04
T34   LT 00006 02T04              PARTL ANS  30
T35   IJ 02T03 00T31
T36   SP 02T00 00023              PREV SUM
T37   MA 02T01 02T04              RAND SUM   57
T38   LT 00006 Q0000              FINL MANT  26
T39   LT 00001 A0000              FINL MANT  27
T40   TJ 01T15 00T43
T41   RA 02T02 01T15
T42   TP Q0000 A0000              FINL MANT  26
T43   CC A0000 02T02
T44   TN A0000 A0000              IF NEG
T45   MJ 00000 00T02              OUT
1T00  01 57079 63268          34  PI OVER 2
1T01  -4 05405 80000 -03      30  C15
1T02  02 18612 28800 -02      35  C13
1T03  -5 59098 86100 -02      35  C11
1T04  09 64200 44100 -02      35  C9
1T05  -1 39085 33510 -01      35  C7
1T06  01 99465 35990 -01      35  C5
1T07  -3 33298 56050 -01      35  C3
```

```
1T08   09 99999 33290 -01   35 C1
1T09   00 07777 77777 B        MANTISSA MSK
1T10   37 70000 00000 B        EXP MASK
1T11   00 00000 00201 B
1T12   20 00000 00000 B
1T13   00 00000 00044 B
1T14   00 00000 00006 B        INDEX
1T15   00 10000 00000 B
1T16   00 00001 00000 B
1T17   00 00002 00000 B
2T00   00 00000 00000          SUM
2T01   00 00000 00000          FLOATNG ARG
2T02   00 00000 00000          EXP
2T03   00 00000 00000          INDEX
2T04   00 00000 00000          TEMP
```

USE Subroutine

I. Identification

R-W EXP-3, Floating point exponential, Malcolm Perry,

23 March 1956, Ramo-Wooldridge Corporation

II. Purpose

Given x in floating point this routine computes e^x in floating point.

III. Accuracy, Range, Method

a. For $X \leq 2$, the error in e^x is less than 2^{-26}, for $X > 2$, the error in e^x is less than $2^{-(26-E)}$ where E is the binary power of X.

b. Range. $x < .693 \times 2^7$. If $x < -.693 \times 2^7$, the answer will be zero.

c. Method.

1. $e^x = (2^{-129}) \cdot 2^{x(\log_2 e+129)}$

2. Divide $(X \cdot \log_2 e + 129)$ into an integral part R and a fractional part S. By the necessary limitations on X, $R \geq 0$, $0 \leq S < 1$

3. $e^x = (2^{R-129})(2^S)$

   $= (2^{R-128})(\frac{2^S}{2})$

4. 2^S is evaluated using the Rand Polynomial Approximation number 20.($2^S = 10^{S\log_{10}2}$)

5. Since $0 \leq S < 1$ $,1 \leq 2^S < 2$

   $1/2 \leq 2^S/2 < 1$

6. R-128 is the characteristic of the answer in floating notation, and $\frac{2^S}{2}$ is the mantissa.

IV. Usage

a. Calling sequence

LOC	OP	u	v
r	RJ	t+2	t
r+1	Normal return		

b. Control and Results

The argument x in floating point must be in A upon entry; e^x in floating point is left in A upon normal return.

c. Space required

43 instructions, 19 constants, 3 temporaries

d. Alarm conditions

The alarm exit is used if x falls outside the permissible range.

V. No restrictions

VI. Coding Information

a. Constants

LOC	CONSTANT
+44	2^{35} ln2
+45	2^{34} $\frac{1}{2}$ln2
+53	2^{34}
+54	40 07777 77777b
+55	37 70000 00000b
+56	00 00000 00170b
+57	00 00000 00044b
+58	6
+59	2^{15}
+60	2^{27}
+61	43

```
E00   MJ 00000 00E03        ENTRY
E01   00 00000 00000        ALARM EXIT
E02   MJ 00000 00000        NORMAL EXIT
E03   TP A0000 Q0000
E04   QT 01E10 02E00        MANTISSA
E05   QT 01E11 02E01        EXPONENT
E06   QJ 00E07 00E09
E07   CC 02E00 01E11
E08   CC 02E01 01E11
E09   LT 00009 A0000
E10   ST 01E12 Q0000        E PLUS 8
E11   TJ 01E17 00E13
E12   MJ 00000 00E34
E13   SJ 00E14 00E16
E14   SA 01E13 00000
E15   SJ 00E23 00E16
E16   TV A0000 00E17
E17   LA 02E00 00000
E18   QJ 00E19 00E20
E19   LT 00000 A0000        MANTISSA   35
E20   AT 01E01 00000        HLF LN 2
E21   DV 01E00 02E01        LN 2
E22   ST 01E01 Q0000        HLF LN 2   35
E23   TP 01E02 02E00        C7         34
E24   TU 00E23 00E29
E25   TP 01E14 02E02        INDEX
E26   RA 00E29 01E15
E27   MP Q0000 02E00                   69
```

RW EXP-3
3-26-56

```
E28   LT 00001 A0000                      34
E29   AT 00000 02E00                      34
E30   IJ 02E02 00E26
E31   LA 02E01 20027
E32   AT 01E09 Q0000
E33   EJ A0000 00E36
E34   TP 01E18 00000           ALARM TAG
E35   MJ 00000 00E01
E36   SP 02E00 00028
E37   LT 00000 02E00                      27
E38   LT 00001 A0000                      28
E39   TJ 01E16 00E42
E40   RA Q0000 01E16
E41   TP 02E00 A0000
E42   AT Q0000 A0000           PACK
E43   QJ 00E34 00E02           CHECK EXP
1E00  06 93147 18056 -01   35 LN 2
1E01  06 93147 18056 -01   34 HLF LN 2
1E02  01 99243 65600 -04   34 C7
1E03  01 39485 76760 -03   34 C6
1E04  08 33324 84740 -03   34 C5
1E05  04 16662 18354 -02   34 C4
1E06  01 66666 66994 -01   34 C3
1E07  05 00000 01077 -01   34 C2
1E08  09 99999 99997 -01   34 C1
1E09  20 00000 00000 B        C0
```

RW EXP-3

3-26-56

1E10	40 07777 77777 B	MASK
1E11	37 70000 00000 B	MASK
1E12	00 00000 00170 B	
1E13	00 00000 00044 B	
1E14	00 00000 00006 B	
1E15	00 00001 00000 B	
1E16	00 10000 00000 B	
1E17	00 00000 00053 B	
1E18	00 00000 00000	ALARM TAG
2E00	00 00000 00000	MANTISSA
2E01	00 00000 00000	EXPONENT
2E02	00 00000 00000	INDEX

PROGRAM WRITE-UP

1. IDENTIFICATION ARCSINE X, STATED POINT

A. FRANCK, M.D. BERNICK - MARCH 1956

REMINGTON-RAND UNIVAC

2. PURPOSE

Given x, this program computes Arcsine x.

3. METHOD

a. Accuracy = $|\text{arcrin } x - F(x)| \leq 2^{-24}$

b. Range of argument: $|(x)| \leq 1$

c. F(x) = aresin x is computed in radians from the polynomial approximation

$$\arcsin x \approx \pi/2 - \sqrt{1-x} \sum_{i=o}^{7} a_i x^i.$$

The square root is computed by an approximation and one application of the Newton-Raphson Formula.

4. USAGE

a. Calling Sequence

Lou.	OP	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal	Return	

b. Control and Results

The argument is to be placed in the Accumulator and the function will be found in the Accumulator. The program also places the argument in t+3 and the function in t+4.

c. Space Required: 109 locations including constants and working storage.

d. Error Codes

CODE	EXPLANATION
301224060001	| x | > |

5. RESTRICTIONS

6. CODING INFORMATION

   a. Constants: 22 locations

   b. Working Storage: 7 locations

   c. Timing

ARCSINE x, STATED POINT
A. FRANCK, M.D. BERNICK - MARCH, 1956
REMINGTON-RAND UNIVAC

LOCATION.	OPERATION.	uADDRESS.	vADDRESS.	EXPLANATION
ENTRY	MJ	∅	START	Jump to body of program
ERROR	RJ	DIAG+2	DIAG	Error exit to diagnositc routine
EXIT	MJ	∅	FILL	Success exit
X	FILL	FILL	FILL	Argument
Y	FILL	FILL	FILL	Function
START	TP	A	X	Store argument
	TM	A	A	$\|x\|-2^{33} \rightarrow A$
	TJ	B21	PROG	
	MJ	∅	ERROR1	$\|x\| > 1$
PROG	TP	A	Y	
	ZJ	NOZERO	EXIT	$x=0$
NOZERO	TP	A	STORE	Store $\|x\|$
	TP	X	A	$x \rightarrow A$
	TP	B1	STORE1	Preset STORE1 for sign determination
	SJ	NEG	POS	
NEG	TP	B2	STORE1	Reset STORE1 for negative argument
	TM	A	A	$\|x\| \rightarrow A$
POS	EJ	B	XONE	$\|x\|=1$
	MP	STORE	B3	$a > x \cdot 2^{77}$
	LT	1	A	
	AT	B4	STORE2	$(STORE2)=(a_7 x+a_6) \cdot 2^{42}$
	MP	STORE	STORE2	
	LT	1	A	

LOCATION.	OPERATION.	uADDRESS.	vADDRESS.	EXPLANATION
	AT	B5	STORE2	$(a_7x^2+a_6x+a_5)\cdot 2^{40}$
	MP	STORE	STORE2	
	LT	3	A	
	AT	B6	STORE2	$(a_7x^3+\cdots+a_4)\cdot 2^{40}$
	MP	STORE	STORE2	
	LT	2	A	
	AT	B7	STORE2	$(a_7x^4P\cdots+a_3)\cdot 2^{39}$
	MP	STORE	STORE2	
	LT	2	A	
	AT	B8	STORE2	$(a_7x^5+\ldots+a_2)\cdot 2^{38}$
	MP	STORE	STORE2	
	LT	2	A	
	AT	B9	STORE2	$(a_7x6+\ldots+a_1)\cdot 2^{37}$
	MP	STORE	STORE2	
	LT	∅	A	
	AT	BID	STORE2	$(a_7x^7+\ldots+a_0)\cdot 2^{34}$
	TN	STORE	A	$(A)=-x\cdot 2^{33}$
	SA	B	∅	$(A)=(1-x)\cdot 2^{33}$
	SF	A	STORE3	Scale factor
	SP	A	∅	$(A1)=0; (A_R)=N; 2^{34}\leq N\leq 2^{35}$
	TP	A	STORE4	Store N
	SA	B11	18	$\left[(N+D)\right]\div 2^{18} = F\rightarrow A$
	LT	∅	STORE5	Store F
	MP	B12	STORE5	$A.F\rightarrow A_R$
	AT	B13	STORE6	Store A.F+B
	SN	B14	15	$-C\cdot 2^{15}\rightarrow A$

LOCATION.	OPERATION.	uADDRESS.	vADDRESS.	EXPLANATION
	DV	STORE5	A	$(-C\cdot 2^{15})\div F=-G$ A_R
	AT	STORE6	STORE5	Store $AF+B-G=Y_1$
	SP	STORE4	32	$N\cdot 2^{32}$ A
	SS	STORE5	∅	$N\cdot 2^{32}$ Y_1 A
	DV	STORE5	A	$(N\cdot 2^{32}\div Y_1)-1$ A_R
	AT	STORE5	STORE5	Store $(N\cdot 2^{32}\div Y_1)-1+Y_1=Y_2$
	LQ	STORE3	35	Ko Q35,K6...K1...Q5...Qo
	QT	B17	A	$(K-Ko)\div 2\rightarrow A_R$
	TV	A	RIFT	Set up V part of shift
	TP	DONE	PROG1	Set PROG1 for storage of results
	TJ	B15	NOSCAR	$(K-Ko)\div 2<19$, jump
	TP	SCAR	PROG1	Reset PROG1 for scale and round
NOSCAR	QJ	RIFT	KEVEN	K odd, jump
KEVEN	MP	B16	STORE5	$2^{34}\sqrt{2}\cdot Y_2\rightarrow A$
	SA	B18	1	$1/2\ \sqrt{2}\cdot Y_2+1/2\rightarrow A_R$
	LT	∅	STORE5	$\approx 2^{17}\sqrt{N\cdot 2^{ko-1}}$ for K even
RIFT	SP	STORE5	FILL	$2^{17}\sqrt{N\cdot 2^{k-1}}\rightarrow A$
PROG1	FILL	FILL	FILL	$K\leq 37$, exit; $K>37$, round & scale
	LT	∅	A	
DONE	MJ	∅	PROG2	
SCAR	SA	B19	∅	
PROG2	MP	A	STORE2	$1-x\ X(x)\cdot 2^{67}$
	LT	2	A	$1-x\ X(x)\cdot 2^{33}$
	ST	B20	Q	$(Q)=(c-\arcsin X)\cdot 2^{33}$
PROG3	IJ	STORE1	NEG1	x neg, jump
	TN	Q	A	
NEG1	TP	A	Y	

LOCATION.	OPERATION.	uADDRESS.	vADDRESS.	EXPLANATION
~~NEXT~~	MJ	Ø	EXIT	Jump to exit
XONE	TN	B20	Q	x=1;$- \frac{\pi}{2} \cdot 2^{33} \rightarrow Q$
	MJ	Ø	PROG3	
ERROR1	SP	CODE	A	Enter error code in A
	MJ	Ø	ERROR	Jump to error exit
B	B	100000000000		2^{33}
B1	B		Ø	
B2	B		1	
B3	B	532413520070		a_7 44
B4	B	332441425535		a_6 42
B5	B	564007151545		a_5 40
B6	B	370417441233		a_4 40
B7	B	462370666522		a_3 39
B8	B	266165166073		a_2 38
B9	B	444200330653		a_1 37
B10	B	311037551633		a_0 34
B11	B	264767031361		D=24,290,062,513
B12	B	65324		A=27,349
B13	B	114534644516		B-10,291,988,814
B14	B	330657140271		C=29,104,062,651
B15	B	23		TJ constant
B16	B	265011714640		24,296,004,000
B17	B	77		Masic for SF
B18	B	200000000000		2^{34}
B19	B	377777777777		$2^{35}-1$
B20	B	144417665210		$\frac{\pi}{2} \cdot 2^{33}$

LOCATION.	OPERATION.	uADDRESS.	vADDRESS.	EXPLANATION
B21	B	100000000001		
CODE .	B	301224060001		
STORE	FILL	FILL	FILL	
STORE1	FILL	FILL	FILL	
STORE2	FILL	FILL	FILL	
STORE3	FILL	FILL	FILL	
STORE4	FILL	FILL	FILL	
STORE5	FILL	FILL	FILL	
STORE6	FILL	FILL	FILL	

USE Subroutine

I. Identification: RW SIN-4, Floating Point Sine-Cosine Routine

Malcolm Perry, April, 1956

Ramo-Wooldridge Corporation
Los Angeles 45, California

II. Purpose:

Given x in floating point form, this program calculates sine x if the parameter word is zero or cosine x if the parameter word is non-zero.

III. Accuracy, Range, Method:

a. Sine x or cosine x is computed to 26 binary places of accuracy, or to as many correct bits as there are in the fractional portion of the argument, whichever is less.

When $x \geq 2^{27}$, this routine substitutes zero for the argument. The alarm exit is not used.

b. x may be any floating point number.

c. The Rand Polynomial Approximation Number 16 is used to calculate sin x or cos x.

1. Let $y = (2/\pi)x$, then sine $x = \sin(\pi/2)y$

   cosine $x = \sin(\pi/2)(y+1)$

2. Divide y (or y+1) into an integral part R, and a fractional part S.

3. R defines the quadrant into which x falls. Let R' be the two low order positions of R, since in binary notation, other positions merely define a number of complete revolutions.

4. R' is a number one less than the number of the quadrant into which x falls.

5. S defines the displacement (in a positive direction) within the quadrant indicated by R'.

6. Therefore, if R' = 00, Let = S first quadrant

   R' = 01, Let = (1-S) second quadrant

   R' = 10, Let = (-S) third quadrant

   R' = 11, Let = (1-S) fourth quadrant

7. Sine (or cosine) x = $\sin(\pi/2)z$

8. $(1/z) \sin (\pi/2)z$ is approximated by the Rand Polynomial Approximation Number 16, using argument z.

9. If $x < 1/2$, $(2/\pi)x$, which is in floating form, is substituted for z before doing step 10.

10. Multiply the approximation from item 8 by z giving the result, sin (or cosine x).

IV. Usage:

a. Calling Sequence

<u>LOC</u>	<u>OP</u>	<u>u</u>	<u>v</u>
r	RJ	t + 2	t
r	Normal return		

b. Control and Results

The argument x must be placed in the Accumulator and the parameter word (zero or non-zero) in t + 3. At the time of normal exit from the routine, sinx (or cos x) will be in the Accumulator.

c. Space Required

56 instructions, 12 constants, 4 temporaries.

d. Error Codes

The alarm exit is not used by this routine.

V. Restrictions:

The range of x as a floating point number is approximately $\pm 10^{38}$.

VI. Coding Information:

a. Constants

+62				2^{35} $2/\pi$
+63	17	77777	77777	
+64	00	00001	00000	
+65	00	00000	00200	128
+68	00	00000	00003	

```
00S00   MJ 00000 00S04        ENTRY
00S01   00 00000 00000        ALARM EXIT
00S02   MJ 00000 00000        NORMAL EXIT
00S03   00 00000 00000        PARAM WD
00S04   TU 00S37 00S54        SET EXIT POS
00S05   LT 00009 02S00        EXP PLUS 200
00S06   LQ A0000 00035        MANT SCAL 35
00S07   MP Q0000 01S05           SCALED 70
00S08   LT 00000 Q0000        QUADRANTS
00S09   TP Q0000 02S01           SCALED 34
00S10   TM 02S00 A0000        EXPONENT
00S11   ST 01S08 02S00        UNBIAS EXPON
00S12   SJ 00S13 00S20
00S13   SA 01S10 00000        ANGLE LESS
00S14   SJ 00S17 00S15         THAN HALF
00S15   TV A0000 00S16
00S16   LA Q0000 00000        MANTISSA
00S17   LT 00008 Q0000        RS SCALED 34
00S18   TP 00S03 A0000        PARAMETER WD
00S19   ZJ 00S26 00S36
00S20   TJ 01S10 00S22        ANGLE GREATR
00S21   CC Q0000 A0000         THAN HALF
00S22   TV A0000 00S23
00S23   LA Q0000 00000        RS SCALED 34
00S24   TP 00S03 A0000        PARAMETER WD
00S25   ZJ 00S26 00S28
00S26   TN Q0000 A0000        IF COSINE
00S27   AT 01S06 Q0000
00S28   QT 01S06 02S01        ARE Z FIXED
00S29   CC 02S00 A0000        EXP FINL ZRO
00S30   QJ 00S31 00S32
00S31   RS 00S54 01S07        QUADRNT 3OR4
00S32   QJ 00S33 00S35
00S33   TP 01S06 A0000        QUADRNT 2OR4
00S34   ST 02S01 02S01
00S35   TP 02S01 Q0000
00S36   MP Q0000 Q0000        Z SQUARED
00S37   LT 00002 02S02           SCALED 34
00S38   TP 01S00 Q0000        C9 SCALED 30
00S39   TP 01S11 02S03        INDEX
```

```
00S40     TU 00S38 00S44
00S41     RA 00S44 01S07                BUMP CONSTNT
00S42     MP Q0000 02S02                EVALUATE
00S43     LT 00002 A0000                 RAND
00S44     AT 00000 Q0000                ' POLYNOMIAL
00S45     IJ 02S03 00S41                   SCALED 30
00S46     MP Q0000 02S01                MULT BY Z 64
00S48     ZJ 00S49 00S02                EXIT TEST
00S49     SF A0000 02S03                FLOAT MNTSSA
00S50     LT 00028 Q0000
00S51     RA 02S00 02S03                PLUS SCL FCT
00S52     SA 01S09 00027                FINL EXPONNT
00S53     CC A0000 Q0000                PACK
00S54     RP 00000 00S02                ONCE FOR NEG
00S55     TN A0000 A0000                TWICE FR POS
00S47     LT 00006 A0000                FINL MANTSSA
01S00     01 51484 19000 -04    30      C-9
01S01     -4 67376 55700 -03    30      C-7
01S02     07 96896 79280 -02    30      C-5
01S03     -6 45963 71106 -01    30      C-3
01S04     01 57079 63185        30      C-1
01S05     06 36619 77225 -01    35      2 OVER PI
01S06     17 77777 77777 B              MASK
01S07     00 00001 00000 B
01S08     00 00000 00200 B              128
01S09     00 00000 00073 B              59
01S10     00 00000 00034 B              28
01S11     00 00000 00003 B              3
```

LOAD HUB TO AUTOMATIC PLOTTER CARD CONVERSION

This routine will retain the indicative information in the first field of a load hub card and convert the other seven fields from floating decimal form to fixed decimal form which can be read by the automatic plotter. Each output word will be of the form 000XX.XX000 and should not exceed 70.00.

The plotter does not distinguish between an X and Y punch on a load hub card, so a constant must be added to fields having negative values, to make all values positive. Each field is divided by a scaling factor so that the final output will be units of centimeters. Use the following formula to determine the scaling factors.

$$\frac{X + K}{M} = X_p$$

Given X

K = A Multiple of M such that K bounds largest negative number in the field.

M = Number of units of X desired per centimeter. M should be of the form 1×10^p, 2×10^p, or 5×10^p where p may equal any number. This facilitates easier reading of the graph.

X_p = Plotter values in centimeters.

I. Input (Use 533 Panel No. 2, 7, or 12).

- A. Load Card.
- B. FLAIR Deck (08 command is used) Decks 033.20 and 033.21.
- C. Conversion Deck No. 079.00 - 16 cards.
- D. Load Hub Card containing K in floating point for each field (except first field -- first field may be used for indicative material and is not used in the output).
- E. Load Hub Card containing M in floating point for each field (except first field -- not used in output).
- F. Load Hub Cards to be converted.

NOTE: Input must be in this order for routine to work.

II. Console Setting -- Standard Load Setting.

III. Output.

- A. Load Hub form -- fixed decimal in form 000XX.XX000 with Y punch in units position of each field.
- B. A one (1) is added to the left position in the first field, which still contains original indicative information, to identify it as a plotter card. The last seven fields are converted at a rate of approximately 25 cards per minute.

EXAMPLE OF HOW TO DETERMINE SCALING FACTORS

1st Field	2nd	3rd	4th	5th	6th	7th	8th
Indicative Information							

Suppose field number 4 is to be plotted against field number 6. In field 4 are the X values and in field 6 are the Y values. The highest X value is 1379 and the largest negative X value is -17. The highest Y value is 120 and there are no negative Y values. The graph is to be plotted on the regular size "centimeter paper", which is 70 X 70 centimeters (notebook size is 20 cm. X 25 cm.).

$\frac{1379 + K}{M}$ = No more than 70 cm.

$\frac{1379 + 20}{20}$ = 69.95 cm.

Therefore: K = 20, M = 20 (for Field 4)

The X axis is moved up 1 cm.

$\frac{120 + K}{M}$ = No more than 70 cm.

$\frac{120 + 0}{2}$ = 60 cm.

Therefore: K = 0, M = 2 (for Field 6)

The Y axis does not move.

Coding Eugene W. Hubbard

Write-up Eugene W. Hubbard

Approved E C Dodge

EWH:ag

June 29, 1956

Revision of FLAIR

New 06 Command

Deck 033.24 consists of four cards and is now available to use in conjunction with FLAIR. This deck changes the 06 command from a NO-OP to a repeat command of the form "repeat the set of commands starting at α the number of times in β." The 06 command must be sequential to the last instruction of the set. At the completion of β loops, the next command is sequential to the 06 command.

The form of the instruction is 06 α N, where α is the address of the first instruction of the set to be repeated, and N is the number of times the loop is to be performed.

For example, if I is the address of the first instruction of the set, and L is the last, the address of the 06 command must be L + 1.

A	O	α	β
L+1	06	I	N

Thus, the set of commands from address I to L will be performed N times and the next instruction will be at L+2. The set of commands from I to L may contain both FLAIR and machine language commands, providing L is a FLAIR command, or a command to re-enter FLAIR at (L+1), where (L+1) is sequential to the last FLAIR command used.

If N is zero, the 06 command is a NO-OP, and the next instruction is sequential.

If the set of instructions I to L must be used again later in the program, their addresses may be modified by use of the 08 and/or 8 γ command so that the 06 command may be used again on the same set (even though it is not sequential before the modification).

This deck should be read into the machine following FLAIR so that the proper modification of FLAIR will be made. Storages 1200 thru 1216 are required for this command.

Coding of new 06 command:

A	O	D	I
1678	65	1615	1204
1204	16	1793	1205
1205	69	1203	1206
1206	22	1203	8001
1203	65	0000	1207
1207	16	1202	1208
1208	69	1615	1209
1209	22	1890	1210
1210	35	0006	1211
1211	11	8003	1212

Coding of new 06 command: (continued)

A	O	D	I
1212	45	1213	1200
1213	65	1202	1214
1214	15	1201	1215
1215	20	1202	1216
1216	60	1890	8003
1200	20	1202	1612
1201	00	0000	0001
1202	00	0000	0000

Coding W. O. Wootan

Write-up W. O. Wootan

Approved B C Dove

ag

650-SD-00008S

June 29, 1956

ADDITION TO FLAIR
New 09 Command

Deck 033.23 can be used in conjunction with FLAIR (Deck 033.20) and the FLAIR index registers (Deck 033.21) to provide a "transfer and set index" command.

The 09 α β command, if located in storage "n", sets the index register tagged by "α" to "n + 1" and transfers the control sequence to address "β"

This command makes an automatic return from subroutines possible.

Example: sets index "D" to 0111

0109	80	0008	0250
0110	09	8000	0251
0251	Hypothetical routine which always uses number stored in Position 0250		
.			
.			
.			
0259			
0260	00	0000	8000
0111	OP	α	β

This Routine uses drum positions 1217 to 1227 inclusive.

Coding:

A	O	D	I
1681	65	1867	1217
1222	35	3	1225
1218	22	1885	1222
1225	19	1642	1219
1219	35	2	1226
1226	15	1220	1227
1220	23	1209	1792
1227	22	1867	1221
1221	65	1615	1223
1223	30	4	1224
1224	69	8003	1867
1217	69	1885	1218

Coding Charles M. Wimberley

Write-up Charles M. Wimberley

Approved Bernard C. Dove

ag

SYMBOLIC LIBRARY OPERATIONS. TAPE NO. 05906 20 FEB 1957

D	LGT		45000	LOGIC TRACE				
D	FTR		44327	FAP TRACE				
D	DDD		40000	DRUM STARTS				
D	DDR		40026	CALL SEQS				
D	DAS		40172	INT PR				
D	DST		40203	COLD				
D	ST	40203	00000	START				
D	DRR		40234	TAPE				
D	RR	40203	00031	READ				
D	DBL		40256	BINARY				
D	BL	40203	00053	LOAD				
D	DMP		40224	BASIC CALLING				
D	MP	40203	00021	SEQUENCE				
D	DCN		40307	CONSTANTS FOR				
D	CN	40203	00104	OP PKG				
D	DBS		40326	BLOCK				
D	BS	40203	00123	STORE				
D	DMQ		40360	CHECKSUM				
D	MQ	40203	00155	ROUTINE				
D	DMM		40373	DRUM				
D	MM	40203	00170	READ				
D	DLF		40415	LOAD FAP				
D	LF	40203	00212	WITH BINARY				
D	F		01750	OPTION				
D	DZY		40441	PREPARE				
D	ZY	40203	00236	SERVICE LIB				
D	MAF		53500	MAFIA				
D	MHP		50000	MISHAP				
D	MCR		50002	MISCRAP				
D	DUM		40557	OCTAL DUMP			DUM	45600 RELOC
D	DTC		45652	DATA TAPE			DTC	44632 RELOC
D	TC		01774	CORRECTION				
D	DUP		41702	TAPE			DUP	44230 RELOC
D	UP		00037	DUPLICATE				
D	CAM		46144	TAPE			CAM	44172 RELOC
D	AM		00001	COMPARE				
D	PBT		45516	PRINT BIN TAPE			PBT	41325 RELOC
D	RBC		45574	READ BIN CDS			RBC	46300 RELOC
D	PBC		41176	PCH BIN CDS			PBC	46400 RELOC
D	RPQ		46426	READ PAPER TAPE			RPQ	44454 RELOC
D	DCI		46600	DRUM CORE IMAGE				
D	AB		00170	BUFFER				
D	FAP		41376	FAP				
D	BCW		00055	BKPT CONTROL WD				
D	FF		01420	BUFFER				
D	GF		01610	ANOTHER BUFFER				
DDD00	MJ		DDR00	RD SERVICE TPE	40000	45 00000 40026		
DDD01	MJ		DDD02	DRUM F1	40001	45 00000 40002		
DDD02	MS		DDD00	STOP	40002	56 00000 40000		
DDD03	MJ		DDR05	LOAD FAP	40003	45 00000 40033		
DDD04	MJ		DDR10	LOAD BINARY	40004	45 00000 40040		
DDD05	MJ		DDR15	LOAD MAFIA	40005	45 00000 40045		
DDD06	MJ		DDR20	ASSEMBLY	40006	45 00000 40052		
DDD07	MJ		DDR25	CORRECTION	40007	45 00000 40057		
DDD08	MJ		DDR30	MEMORY DUMP	40010	45 00000 40064		
DDD09	MJ		DDR35	LOGIC TRACE	40011	45 00000 40071		
DDD10	MJ		DDR40	FAP TRACE	40012	45 00000 40076		
DDD11	MJ		DDR45	NOT USED	40013	45 00000 40103		
DDD12	MJ		DDR50	NOT USED	40014	45 00000 40110		

```
DDD13   MJ          DDR55            DATA TAPE CORR       40015   45 00000 40115
DDD14   MJ          DDR60            TAPE DUPLICATE       40016   45 00000 40122
DDD15   MJ          DDR65            TAPE COMPARE         40017   45 00000 40127
DDD16   MJ          DDR70            PRINT BINARY TP      40020   45 00000 40134
DDD17   MJ          DDR75            READ BIN CDS         40021   45 00000 40141
DDD18   MJ          DDR80            PUNCH BIN CDS        40022   45 00000 40146
DDD19   MJ          DDR85            READ PAPER TAPE      40023   45 00000 40153
DDD20   MJ          DDR90            NOT USED             40024   45 00000 40160
DDD21   MJ          DDR95            PREPARE LIBRARY      40025   45 00000 40165
DDR00   RJ DMP00 DAS00               READ                 40026   37 40224 40172
DDR01   TP  ST01   ST01               SERVICE             40027   11 00001 00001
DDR02   00 00020 00000   B            TAPE                40030   00 00020 00000
DDR03                    B                                40031   00 00000 00000
DDR04      DST01         BRB                              40032   00 40204 00000
DDR05   RJ DMP00 DAS00               LOAD FAP             40033   37 40224 40172
DDR06   TP  LF00   LF00               WITH                40034   11 00212 00212
DDR07   00 00024 00000   B            BINARY              40035   00 00024 00000
DDR08                    B            OPTION              40036   00 00000 00000
DDR09      DLF00         BRB                              40037   00 40415 00000
DDR10   RJ DMP00 DAS00               LOAD ASSEMBLED       40040   37 40224 40172
DDR11   TP  BL00   BL00               BINARY TAPE         40041   11 00053 00053
DDR12   00 00031 00000   B            WITH STOP           40042   00 00031 00000
DDR13                    B            OPTION              40043   00 00000 00000
DDR14      DBL00         BRB                              40044   00 40256 00000
DDR15   RJ DMP00 DAS00               LOAD MAFIA           40045   37 40224 40172
DDR16   TP MAF00 MAF00                                    40046   11 53500 53500
DDR17   00 01500 00000   B                                40047   00 01500 00000
DDR18                    B                                40050   00 00000 00000
DDR19      MAF00         BRB                              40051   00 53500 00000
DDR20   RJ DMP00 DAS00               MISHAP               40052   37 40224 40172
DDR21   TP MHP00 MHP00                                    40053   11 50000 50000
DDR22   00 03477 00000   B                                40054   00 03477 00000
DDR23                    B                                40055   00 00000 00000
DDR24      MHP00         BRB                              40056   00 50000 00000
DDR25   RJ DMP00 DAS00               MISCRAP              40057   37 40224 40172
DDR26   TP MCR00 MCR00                                    40060   11 50002 50002
DDR27   00 03475 00000   B                                40061   00 03475 00000
DDR28                    B                                40062   00 00000 00000
DDR29      MCR00         BRB                              40063   00 50002 00000
DDR30   RJ DMP00 DAS00               MEMORY               40064   37 40224 40172
DDR31   TP DUM00 DUM00                DUMP                40065   11 40557 40557
DDR32   00 00425 00000   B                                40066   00 00425 00000
DDR33                    B                                40067   00 00000 00000
DDR34      DUM00         BRB                              40070   00 40557 00000
DDR35   RJ DMP00 DAS00               LOGIC                40071   37 40224 40172       REPL
DDR36   TP LGT00 LGT00                TRACE               40072   11 45000 45000       REPL
DDR37   00 00516 00000   B                                40073   00 00516 00000       REPL
DDR38                    B                                40074   00 00000 00000
DDR39   00 LGT00 00000   BRB                              40075   00 45000 00000       REPL
DDR40   RJ DMP00 DAS00               FAP                  40076   37 40224 40172       REPL
DDR41   TP FTR00 FTR00                TRACE               40077   11 44327 44327       REPL
DDR42   00 00451 00000   B                                40100   00 00451 00000       REPL
DDR43                    B                                40101   00 00000 00000
DDR44   00 FTR00 00000   BRB                              40102   00 44327 00000       REPL
DDR45   RJ DMP00 DAS00               NOT USED             40103   37 40224 40172
DDR46                    B                                40104   00 00000 00000
DDR47                    B                                40105   00 00000 00000
DDR48                    B                                40106   00 00000 00000
DDR49                    B                                40107   00 00000 00000
```

```
DDR50    RJ DMP00 DAS00            NOT USED          40110   37 40224 40172
DDR51                      B                         40111   00 00000 00000
DDR52                      B                         40112   00 00000 00000
DDR53                      B                         40113   00 00000 00000
DDR54                      B                         40114   00 00000 00000
DDR55    RJ DMP00 DAS00            DATA              40115   37 40224 40172
DDR56    TP DTC00   TC00            TAPE             40116   11 45652 01774
DDR57    00 00300 00000    B        CORRECTION       40117   00 00300 00000
DDR58                      B                         40120   00 00000 00000
DDR59       DTC00          BRB                       40121   00 45652 00000
DDR60    RJ DMP00 DAS00            TAPE              40122   37 40224 40172
DDR61    TP DUP00   UP00            DUPLICATE        40123   11 41702 00037
DDR62    00 00224 00000    B                         40124   00 00224 00000
DDR63                      B                         40125   00 00000 00000
DDR64       DUP00          BRB                       40126   00 41702 00000
DDR65    RJ DMP00 DAS00            TAPE              40127   37 40224 40172
DDR66    TP CAM00   AM00            COMPARE          40130   11 46144 00001
DDR67    00 00262 00000    B                         40131   00 00262 00000
DDR68                      B                         40132   00 00000 00000
DDR69       CAM00          BRB                       40133   00 46144 00000
DDR70    RJ DMP00 DAS00            PRINT             40134   37 40224 40172
DDR71    TP PBT00 PBT00             BINARY TAPE      40135   11 45516 45516
DDR72    00 00051 00000    B                         40136   00 00051 00000
DDR73                      B                         40137   00 00000 00000
DDR74       PBT00          BRB                       40140   00 45516 00000
DDR75    RJ DMP00 DAS00            READ FULL         40141   37 40224 40172
DDR76    TP RBC00 RBC00             BINARY           40142   11 45574 45574
DDR77    00 00100 00000    B        CARDS            40143   00 00100 00000
DDR78                      B                         40144   00 00000 00000
DDR79       RBC00          BRB                       40145   00 45574 00000
DDR80    RJ DMP00 DAS00            PUNCH             40146   37 40224 40172
DDR81    TP PBC00 PBC00             BINARY           40147   11 41176 41176
DDR82    00 00177 00000    B        CARDS            40150   00 00177 00000
DDR83                      B                         40151   00 00000 00000
DDR84       PBC00          BRB                       40152   00 41176 00000
DDR85    RJ DMP00 DAS00            READ              40153   37 40224 40172
DDR86    TP RPQ00 RPQ00             BIOCTAL          40154   11 46426 46426
DDR87    00 00070 00000    B        PAPER TAPE       40155   00 00070 00000
DDR88                      B                         40156   00 00000 00000
DDR89       RPQ00          BRB                       40157   00 46426 00000
DDR90    RJ DMP00 DAS00            NOT USED          40160   37 40224 40172
DDR91                      B                         40161   00 00000 00000
DDR92                      B                         40162   00 00000 00000
DDR93                      B                         40163   00 00000 00000
DDR94                      B                         40164   00 00000 00000
DDR95    RJ DMP00 DAS00            PREPARE           40165   37 40224 40172
DDR96    TP  ZY00   ZY00            SERVICE          40166   11 00236 00236
DDR97    00 00114 00000    B        LIBRARY          40167   00 00114 00000
DDR98                      B        FROM DRUM        40170   00 00000 00000
DDR99       DZY00          BRB                       40171   00 40441 00000
DAS00    TP        DCI00           SAVE F1 CORE      40172   11 00000 46600
DAS01    TP Q      DMP05           SAVE Q            40173   11 31000 40231
DAS02    LT        DMP06           SAVE A LEFT       40174   22 00000 40232
DAS03    TP A      DMP07           SAVE A RIGHT      40175   11 32000 40233
DAS04    TP DST00                  RESET F1 CORE     40176   11 40203 00000
DAS05    RP 30239 DAS07            SAVE CORE         40177   75 30357 40201
DAS06    TP   ST01 DCI01            IMAGE            40200   11 00001 46601
DAS07    RP 30239   MM00           PROGRAM           40201   75 30357 00170
DAS08    TP DST01   ST01            TO CORE          40202   11 40204 00001
```

```
ST00    MJ          ST01                CORE F1                 40203    45 00000 00001
ST01    TV  RR12    ST15                SET COLD EXIT           40204    16 00045 00017
ST02    TV  RR02    CN11                SET STORE EXIT          40205    16 00033 00117
ST03    TP Q        A                   LIB TPE NO TO A         40206    11 31000 32000
ST04    ZJ  ST06    ST05                IF ZERO                 40207    47 00006 00005
ST05    LQ  CN12 31022                   SET EQUAL ONE          40210    55 00120 31026
ST06    RJ  RR10    RR00                READ BLOCK              40211    37 00043 00031
ST07    TP  AB00 A                      FIRST WD                40212    11 00170 32000
ST08    EJ  CN06    ST10                TST END TPE             40213    43 00112 00012
ST09    MJ          BS00                TO STORE AND RD         40214    45 00000 00123
ST10    17 30357    CN00    BBR         REWIND LIB TPE          40215    17 30357 00104
ST11    TU  ST10    ST15                SET RPT                 40216    15 00012 00017
ST12    TP  MP05 Q                      RESTORE Q               40217    11 00026 31000
ST13    SP  MP06       36                A LEFT AND             40220    31 00027 00044
ST14    SA  MP07                         A RIGHT                40221    32 00030 00000
ST15    RP 30000 DDR00                  REST ADJUSTED           40222    75 30000 40026
ST16    TP DCI01    ST01                 CORE AND EXIT          40223    11 46601 00001
RR00    SP Q           12               TU NUMBER TO A          40234    31 31000 00014
RR01    LQ  CN09 31008                  MASK TO Q               40235    55 00115 31010
RR02    RP 10004    RR04                SET REWIND MOVE         40236    75 10004 00035
RR03    QS A        CN00                 BKSPCE READ            40237    53 32000 00104
RR04    EF          CN04                SET BIAS NORMAL         40240    17 00000 00110
RR05    TP  CN05    MP00                SET BIAS INDEX          40241    11 00111 00021
RR06    EF          CN03                READ ONE                40242    17 00000 00107
RR07    RP 10120    RR09                 BLOCK INTO             40243    75 10170 00042
RR08    ER 10000    AB00                 BUFFER                 40244    76 10000 00170
RR09    ER          A                   IOA TO A                40245    76 00000 32000
RR10    ZJ  RR11                        TST FOR ERROR           40246    47 00044 00000
RR11    IJ  MP00    RR13                TST BIAS                40247    41 00021 00046
RR12    57 00000 DDD02      BBR         STOP-BAD TAPE           40250    57 00000 40002
RR13    EF          CN02                BACKSPACE               40251    17 00000 00106
RR14    SP A           12               SET UP                  40252    31 32000 00014
RR15    AT  CN04    MP03                 AND                    40253    35 00110 00024
RR16    EF          MP03                 CHANGE BIAS            40254    17 00000 00024
RR17    MJ          RR06                REREAD                  40255    45 00000 00037
BL00    RJ  RR10    RR00                READ BLOCK              40256    37 00043 00031
BL01    TP  CN07    MP04                SET INDEX TO 5          40257    11 00113 00025
BL02    TU  ST07    BL03                TEST                    40260    15 00007 00056
BL03    TP          A                    END OF                 40261    11 00000 32000
BL04    EJ  CN06    BL08                 HIGH SPEED             40262    43 00112 00063
BL05    RA  BL03    CN13                 PRINTER                40263    21 00056 00121
BL06    IJ  MP04    BL03                 PART OF TAPE           40264    41 00025 00056
BL07    MJ          RR04                READ ANOTHER            40265    45 00000 00035
BL08    TV  RR12    ST15                SET EXIT STOP           40266    16 00045 00017
BL09    TV  BL07    CN11                SET STORE EXIT          40267    16 00062 00117
BL10    RJ  RR10    BL06                READ BINARY             40270    37 00043 00061
BL11    TP  AB00 Q                      TEST END                40271    11 00170 31000
BL12    QJ  BL18    BL13                 BINARY                 40272    44 00075 00070
BL13    RP 20120    BL15                FORM AND                40273    75 20170 00072
BL14    SA  AB00                         SAVE                   40274    32 00170 00000
BL15    SA  MP04                         RUNNING                40275    32 00025 00000
BL16    TP A        MP04                 CHECKSUM               40276    11 32000 00025
BL17    MJ          BS00                STORE AND READ          40277    45 00000 00123
BL18    EF          CN01                MOVE FWD 1 BLK          40300    17 00000 00105
BL19    TP  MP05 Q                      TPE NO TO Q             40301    11 00026 31000
BL20    QJ  BL22    BL21                SET EXIT                40302    44 00101 00100
BL21    TV  AB00    ST15                 IF NO STOP             40303    16 00170 00017
BL22    SP  MP04                        FINAL CKSUM             40304    31 00025 00000
BL23    SS  AB01                        TEST FOR                40305    34 00171 00000
```

```
BL24     ZJ  MQ10  ST11           CHECKSUM ERROR      40306   47 00167 00013
MP00                             LOC CALL SEQ         40224   00 00000 00000
MP01     TP  RR00  RR00          TP STORE EXEC        40225   11 00031 00031
MP02     00 00161 00000   B      BLANK NO WDS BL      40226   00 00161 00000
MP03                      B      CK SUM LOWER         40227   00 00000 00000
MP04     00 DRR00 00000   BRB    BL DM LOC CKSUP      40230   00 40234 00000
MP05              60000   B      Q                    40231   00 00000 60000
MP06                      B      A LEFT               40232   00 00000 00000
MP07                      B      A RIGHT              40233   00 00000 00000
CN00     02 00200 10000   B      REWIND               40307   02 00200 10000
CN01     02 00004 10001   B      MOVE FWD             40310   02 00004 10001
CN02     02 00014 10001   B      MOVE BACK            40311   02 00014 10001
CN03     02 00602 10000   B      READ FWD             40312   02 00602 10000
CN04     02 00001 50000   B      NORMAL BIAS          40313   02 00001 50000
CN05     00 00000 00003   B      3                    40314   00 00000 00003
CN06     60 60606 06060   B      INDICATOR            40315   60 60606 06060
CN07     00 00000 00005   B      5                    40316   00 00000 00005
CN08     TP       DC100          OV TRM DUMMY         40317   11 00000 46600
CN09     00 00000 00360   B      240                  40320   00 00000 00360
CN10     RP 30000  BS24          NORM RPT DUMMY       40321   75 30000 00153
CN11     RP 30000                OVLP RPT DUMMY       40322   75 30000 00000
CN12              40000   B      CORE TEST            40323   00 00000 40000
CN13           24                20 S15               40324   00 00030 00000
CN14     RP 20000  MQ07          CKSUM DUMMY          40325   75 20000 00164
BS00     TU  BL23  MP01          SET STORE LOC        40326   15 00102 00022
BS01     TV  AB00  MP01          SET EXEC LOC         40327   16 00170 00022
BS02     TU  AB00  MP02          SET NO WORDS         40330   15 00170 00023
BS03     SP  AB00    21          EXEC LOC             40331   31 00170 00025
BS04     LQ A        15           TO A AND Q          40332   55 32000 00017
BS05     TU  MP01  CN08          SET OV TRM DMMY      40333   15 00022 00114
BS06     AT  CN08  BS25          SET OV TRM           40334   35 00114 00154
BS07     TP  CN09 A              240 MINUS EX         40335   11 00115 32000
BS08     ST Q      MP03           LOC EQ OVLP         40336   36 31000 00024
BS09     SJ  BS10  BS11          IF NEGATIVE          40337   46 00135 00136
BS10     RS  MP03 A               SET ZERO            40340   23 00024 32000
BS11     55  MP03 32017   BRB    POS OV Q AND A       40341   55 00024 32017
BS12     AT  MP03 A              OV IN U V OF A       40342   35 00024 32000
BS13     AT  MP01  BS23          SET NORM TRM         40343   35 00022 00152
BS14     TN Q      A             NO WDS MINUS         40344   13 31000 32000
BS15     AT  MP02 A               OV IN U OF A        40345   35 00023 32000
BS16     SJ  BS17  BS19          IF NEGATIVE          40346   46 00144 00146
BS17     RS A      A              SET ZERO            40347   23 32000 32000
BS18     TP  MP02 Q               N TO Q              40350   11 00023 31000
BS19     AT  CN10  BS22          SET NORM RPT         40351   35 00116 00151
BS20     TP Q      A             SET OVERLAP          40352   11 31000 32000
BS21     AT  CN11  BS24           REPEAT              40353   35 00117 00153
BS22                      B      NORMAL               40354   00 00000 00000
BS23                      B       TRM                 40355   00 00000 00000
BS24                      B      OVERLAP              40356   00 00000 00000
BS25                      B       TRM                 40357   00 00000 00000
MQ00     TU        MP04          ZERO LOCATION        40360   15 00000 00025
MQ01     LQ  MP02 32000          WDS TO Q AND A       40361   55 00023 32000
MQ02     AT  CN14  MQ05          SET RPT              40362   35 00122 00162
MQ03     TU  MP01  MQ06          SET ADD              40363   15 00022 00163
MQ04     RS A      A             CLEAR A              40364   23 32000 32000
MQ05     75 20161  MQ07   BBR    FORM                 40365   75 20161 00164
MQ06     SA  RR00                 CHECKSUM            40366   32 00031 00000
MQ07     SS  MP03    36          SUBTRACT             40367   34 00024 00044
MQ08     SS  MP04                 CORRECT SUM         40370   34 00025 00000
```

```
MQ09   ZJ  MQ10  MM01         EXIT IF ZERO          40371   47 00167 00171
MQ10   MS        ST02         ERROR STOP            40372   56 00000 00002
MM00   RJ  MQ09  MQ00         CKSUM DMM+            40373   37 00166 00155
MM01   55  MP00 31017   BRB   POS LOC CALL          40374   55 00021 31017
MM02   TU Q      MM04         OBTAIN CALL SEQ       40375   15 31000 00174
MM03   RP 30004  MM05          FOR CURRENT          40376   75 30004 00175
MM04   TP        MP01          ROUTINE              40377   11 00000 00022
MM05   IJ  MP00  MM06         CALL MINUS ONE        40400   41 00021 00176
MM06   TV A      ST15         SET ERROR EXIT        40401   16 32000 00017
MM07   RJ  MQ09  MQ00         CKSM ROUTINE          40402   37 00166 00155
MM08   TV  MP01  ST15         SUCCESS EXIT          40403   16 00022 00017
MM09   SP A        06         OBTAIN                40404   31 32000 00006
MM10   SP A        15          STORAGE              40405   31 32000 00017
MM11   LT        MP03          LOCATION             40406   22 00000 00024
MM12   LQ A        15         EX LOC TO A Q         40407   55 32000 00017
MM13   EJ  MP03  MM16         NO TRM IF EQ          40410   43 00024 00210
MM14   RJ  CN11  BS05         OBTAIN ROUTINE        40411   37 00117 00130
MM15   MJ        ST11         EXIT                  40412   45 00000 00013
MM16   ST  CN12 A             IF IN CORE            40413   36 00120 32000
MM17   SJ  ST12  ST11          DO NOT REPLACE       40414   46 00014 00013
LF00   RP 30018   F02         TRANSFER              40415   75 30022 01752
LF01   TP  LF02   F02          ROUTINE              40416   11 00214 01752
LF02   TV  RR12   F17         SET FAP ONLY XT       40417   16 00045 01771
LF03   TP A       F02         SAVE BCW              40420   11 32000 01752
LF04   TP Q      A            IF TAPE NUMBER        40421   11 31000 32000
LF05   ZJ   F06   F08          NOT ZERO             40422   47 01756 01760
LF06   RJ  BL24  BL00          LOAD BINARY          40423   37 00103 00053
LF07   TV  ST15   F17          SET EXIT             40424   16 00017 01771
LF08   75 31607   F10   BBR   BRING FAP             40425   75 31607 01762
LF09   TP FAP01  ST01          OFF DRUM             40426   11 41377 00001
LF10   75 21607   F12   BBR   FORM                  40427   75 21607 01764
LF11   SA  ST01                CHECKSUM             40430   32 00001 00000
LF12   SS   F18    36         SUBTRACT              40431   34 01772 00044
LF13   SS   F19                CORRECT SUM          40432   34 01773 00000
LF14   ZJ   F15   F16         TEST FOR              40433   47 01767 01770
LF15   MS         F08          CHECKSUM ERROR       40434   56 00000 01760
LF16   TP   F02 BCW00         SET BCW               40435   11 01752 00055
LF17   MJ                     EXIT                  40436   45 00000 00000
LF18                    B     CKSUM     LOWER       40437   00 00000 00000
LF19                    B      FOR FAP  UPPER       40440   00 00000 00000
ZY00   RJ  ZY64  ZY50         CKSUM BASIC OPS       40441   37 00336 00320
ZY01   RS A      A            CLEAR A               40442   23 32000 32000
ZY02   75 21607  ZY04   BBR   FORM FAP              40443   75 21607 00242
ZY03   SA FAP01                CHECKSUM             40444   32 41377 00000
ZY04   TP A     DLF18          AND                  40445   11 32000 40437
ZY05   LT       DLF19          STORE                40446   22 00000 40440
ZY06   TP  ZY71  MP00         SET INDEX             40447   11 00345 00021
ZY07   RJ  ZY64  ZY49         TO CKSUM LOOP         40450   37 00336 00317
ZY08   IJ  MP00  ZY50         TST END CKSUMS        40451   41 00021 00320
ZY09   RP 10004  ZY11         SET TAPE              40452   75 10004 00251
ZY10   TV  ZY68  CN00          OPERATIONS           40453   16 00342 00104
ZY11   EF        CN00         REWIND TU2            40454   17 00000 00104
ZY12   TP DMP05 Q             END-40000 OCT         40455   11 40231 31000
ZY13   RS Q      CN12          EQ NO WDS            40456   23 31000 00120
ZY14   DV  ZY66  MP06         BY 119 EQ BLKCT       40457   73 00340 00027
ZY15   LT 10015  MP07         OVFL CT               40460   22 10017 00030
ZY16   RP 30120  ZY18         COLD START            40461   75 30170 00260
ZY17   TP DST00  FF00          TO BUFFER            40462   11 40203 01420
ZY18   TV  ZY34  RR08         SET READ BUFFER       40463   16 00300 00041
```

```
 ZY19   RJ   ZY47  ZY37             WRITE BLOCK         40464   37 00315 00303
 ZY20   TP   ZY67 A                 119 S15 TO A        40465   11 00341 32000
 ZY21   AT   CN12  FF00             SET FIRST WORD      40466   35 00120 01420
 ZY22   RJ   ZY47  ZY25             SET LOOP            40467   37 00315 00267
 ZY23   RA   ZY36  ZY67             BUMP DRUM READ      40470   21 00302 00341
 ZY24   RA   FF00  ZY66             BUMP LOCATION       40471   21 01420 00340
 ZY25   IJ   MP06  ZY35             TEST NO BLOCKS      40472   41 00027 00301
 ZY26   TP   MP07 A                 OVERFLOW TO A       40473   11 00030 32000
 ZY27   ZJ   ZY28  ZY30             EXIT IF ZERO        40474   47 00272 00274
 ZY28   TU A       FF00             SET WORD COUNT      40475   15 32000 01420
 ZY29   RJ   ZY47  ZY35             WRITE BLOCK         40476   37 00315 00301
 ZY30   EF         ZY68             WRITE               40477   17 00000 00342
 ZY31   RP    120  ZY33              END                40500   75 00170 00277
 ZY32   EW 10000   CN06              OF FILE            40501   77 10000 00112
 ZY33   EF         CN00             REWIND              40502   17 00000 00104
 ZY34   FS DDR02   GF00             END PROGRAM         40503   57 40030 01610
 ZY35   RP 30119   ZY37             BLOCK               40504   75 30167 00303
 ZY36   11 40000   FF01    BBR       FROM DRUM          40505   11 40000 01421
 ZY37   EF         ZY68             WRITE               40506   17 00000 00342
 ZY38   RP 10120   ZY40              BLOCK              40507   75 10170 00306
 ZY39   EW 10000   FF00              ON TAPE            40510   77 10000 01420
 ZY40   EF         CN02             BACKSPACE           40511   17 00000 00106
 ZY41   RJ   RR10  RR04             READ BACK           40512   37 00043 00035
 ZY42   RP 30120   ZY44             TAPE MINUS          40513   75 30170 00312
 ZY43   RS   GF00  FF00              ORIGINAL           40514   23 01610 01420
 ZY44   RS A       A                ZERO ACCUM          40515   23 32000 32000
 ZY45   RP 20120   ZY47             FORM                40516   75 20170 00315
 ZY46   SA   GF00                    SUM                40517   32 01610 00000
 ZY47   ZJ   ZY48                   TST FOR ERROR       40520   47 00316 00000
 ZY48   MS         ZY37             TAPE STOP           40521   56 00000 00303
 ZY49   TU   ZY34  ZY51             SET FIRST WORD      40522   15 00300 00321
 ZY50   RP 30003   ZY52             PARAMETERS          40523   75 30003 00322
 ZY51   TP DMP02   MP02              TO CORE            40524   11 40226 00023
 ZY52   TU   MP04  ZY57             SAVE LOC            40525   15 00025 00327
 ZY53   TP   MP02 A                 SET                 40526   11 00023 32000
 ZY54   AT   ZY69  ZY56              REPEAT             40527   35 00343 00326
 ZY55   RS A       A                ZERO ACCUM          40530   23 32000 32000
 ZY56                      B        FORM                40531   00 00000 00000
 ZY57   SA                           CHECKSUM           40532   32 00000 00000
 ZY58   LT         MP04             STORE UPPER         40533   22 00000 00025
 ZY59   TP A       MP03             STORE LOWER         40534   11 32000 00024
 ZY60   TU   ZY57  MP04             RESET LOCATION      40535   15 00327 00025
 ZY61   LQ   ZY51 31021             POSITION            40536   55 00321 31025
 ZY62   TV Q       ZY65             SET DRUM STORE      40537   16 31000 00337
 ZY63   RA   ZY51  ZY70             BUMP LOC            40540   21 00321 00344
 ZY64   RP 30003                    CHECKSUM            40541   75 30003 00000
 ZY65   TP   MP02                    TO DRUM            40542   11 00023 00000
 ZY66               167    B        119                 40543   00 00000 00167
 ZY67         167          B        119 S15             40544   00 00167 00000
 ZY68   02 00606  20001    B        WRITE FORWARD       40545   02 00606 20001
 ZY69   RP 20000   ZY58             CKSUM DUMMY         40546   75 20000 00330
 ZY70          05          B        CKSUM BUMP          40547   00 00005 00000
 ZY71                23    B        19                  40550   00 00000 00023
 ZY72   RP 30239   ZY00             INITIAL LIBRARY     40551   75 30357 00236      REPL
 ZY73   TP DST01   ST01              PREPARATION        40552   11 40204 00001
START      DZY72                                        40551
```


31350

DATA TAPE CORRECTOR.   TAPE # 07153

```
D      DLF              40415                                                          DLF        40351 RELOC
D      F                 1750
D      CS               46144
D      BF               02300
D      CR     43656      1774
D      VC     43656      2004
D      EF     43656      2052
D      HZ     43656      2111
D      CD     43656      2152
D      TR     43656      2223
D      MM     43656      2241
D      TS               00000
D      AB               01610
D      CP               00056
  CR00     RP 30013    CR02          BRING FAP              45652    75 30015 01776
  CR01     TP DLF07     F07           OFF DRUM AND          45653    11 40424 01757
  CR02     RJ    F17    F08           CHECKSUM              45654    37 01771 01760
  CR03     EF          MM06          REWIND TU2             45655    17 00000 02247
  CR04     RP 30003    CR06          CLEAR                  45656    75 30003 02002
  CR05     RS I   01 I   01          INDICES                45657    23 00010 00010
  CR06     TP Q         A            TO CARD                45660    11 31000 32000
  CR07     ZJ  CD00    TR00            OR TAPE READ         45661    47 02152 02223
  VC00     RP 30002    VC02          OBTAIN L               45662    75 30002 02006
  VC01     TP          BF00           AND W                 45663    11 00000 02300
  VC02     LA  BF01      05                                 45664    54 02301 00005
  VC03     LT 00000    BF02          W OR W-1 DIV 2         45665    22 00000 02302
  VC04     SP  BF02      02          TIMES FIVE             45666    31 02302 00002
  VC05     AT  BF02    BF02           EQUAL DEL             45667    35 02302 02302
  VC06     TP  BF01 Q                IF W IS                45670    11 02301 31000
  VC07     QJ  VC08    VC09            EVEN                 45671    44 02014 02015
  VC08     AT  CP05    BF02           ADD 2                 45672    35 00063 02302
  VC09     TN  MM00 Q                MASK TO Q              45673    13 02241 31000
  VC10     QT  BF00 Q                ZS3 L TO Q             45674    51 02300 31000
  VC11     TP  TS07    BF00          ZERO SUM               45675    11 00007 02300
  VC12     TP  CP07 I    04          SET INDEX 4            45676    11 00065 00013
  VC13     SP  BF00      02          SUM                    45677    31 02300 00002
  VC14     SA  BF00      01           TIMES                 45700    32 02300 00001
  VC15     TP A        BF00           TEN                   45701    11 32000 02300
  VC16     LQ Q          06          POSITION               45702    55 31000 00006
  VC17     QT  MM00 A                NEXT DIGIT             45703    51 02241 32000
  VC18     ST  CP06 A                -3                     45704    36 00064 32000
  VC19     SJ  VC21    VC20           SUPPRESS SPACE        45705    46 02031 02030
  VC20     AT  BF00    BF00           ADD TO SUM            45706    35 02300 02300
  VC21     IJ I   04   VC13           TEST END SUM          45707    41 00013 02021
  VC22     RS  BF00    CP04          L-1                    45710    23 02300 00062
  VC23     DV  CP09    BF00           DIV BY 6              45711    73 00067 02300
  VC24     MP A        MM01           FRACT TIMES 20        45712    71 32000 02242
  VC25     AT  MM02 A                 +INITIAL              45713    35 02243 32000
  VC26     AT  BF02 A                 +DEL                  45714    35 02302 32000
  VC27     TV A        BF01           EQ BUFFER ADD         45715    16 32000 02301
  VC28     RA  BF00    CP04           BLOCK NUMBER          45716    21 02300 00062
  VC29     LQ  VC01   31021           SET                   45717    55 02005 31025
  VC30     TV Q        VC33           UP                    45720    16 31000 02045
  VC31     RA  VC01    MM09          BUMP ADDRESS           45721    21 02005 02252
  VC32     RP 30002    VC36           AND                   45722    75 30002 02050
  VC33     TP  BF00                   STORE                 45723    11 02300 00000
  VC34     TU  EF12    VC01          SET FIRST STORE        45724    15 02066 02005
  VC35     TP I   01 I   02          SET IND TO N           45725    11 00010 00011
  VC36     IJ I   02   VC00          TST END LOOP           45726    41 00011 02004
```

```
VC37    MJ          HZ00         TO CORRECT          45727   45 00000 02111
EF00    RJ   EF29   EF29         RESTORE AND         45730   37 02107 02107
EF01  FP74    210   AB00          WRITE LAST BL      45731   14 74210 01610
EF02    TP   MM03 Q              TYPE                45732   11 02244 31000
EF03    RJ   EF19   EF14          TAPE               45733   37 02075 02070
EF04    TP I   02 A               IF ALL CORRECT     45734   11 00011 32000
EF05    ZJ   EF06   EF08           IONS HAVE BEE     45735   47 02060 02062
EF06    TP   MM04 Q                MADE TYPE GOO     45736   11 02245 31000
EF07    MJ          EF09           OTHERWISE         45737   45 00000 02063
EF08    TP   MM05 Q                TYPE              45740   11 02246 31000
EF09    RJ   EF19   EF15           SHORT             45741   37 02075 02071
EF10    PR          CP13         CARRIAGE RETN       45742   61 00000 00073
EF11    RP 10003    EF13         REWIND              45743   75 10003 02067
EF12    EF   BF08   MM06          TAPES              45744   17 02310 02247
EF13    MS 00000    CR04         AND STOP            45745   56 00000 02000
EF14    PR          CP14         SHIFT UP            45746   61 00000 00074
EF15    TP   CP08 I   05         SET LOOP            45747   11 00066 00014
EF16    LQ Q          06         TYPE                45750   55 31000 00006
EF17    PR          Q             DIGIT              45751   61 00000 31000
EF18    IJ I   05   EF16          TEST               45752   41 00014 02072
EF19    MJ                       AND BACK            45753   45 00000 00000
EF20    RJ   EF29   EF21         SET FINAL MODE      45754   37 02107 02077
EF21  FP74    210   AB00         WRITE BLOCK         45755   14 74210 01610
EF22  FP64    310   AB00         READ NEXT           45756   14 64310 01610
EF23    RP 20120    EF25          SUBTRACT           45757   75 20170 02103
EF24    23   AB00 01521   BRB     SIX-OHS            45760   23 01610 01521
EF25    TP   TS07 A              ZERO A              45761   11 00007 32000
EF26    RP 20120    EF28         FORM                45762   75 20170 02106
EF27    SA   AB00 00000           SUM                45763   32 01610 00000
EF28    ZJ   EF29   EF00         TEST END FILE       45764   47 02107 02052
EF29    RP 20120                  RESTORE            45765   75 20170 00000
EF30    21   AB00 01521   BRB     BLOCK              45766   21 01610 01521
HZ00    TP I   01 I   02         SET INDEX           45767   11 00010 00011
HZ01    RJ   EF29   EF22         SET READ            45770   37 02107 02100
HZ02    RA I   03   CP04         BUMP BLOCK CT       45771   21 00012 00062
HZ03    TP I   01 I   06         SET INDEX TO N      45772   11 00010 00015
HZ04    TU   EF12   HZ05         SET FIRST WD        45773   15 02066 02116
HZ05    TP            A          BLOCK NO            45774   11 00000 32000
HZ06    EJ I   03   HZ10         IS THIS IN B6       45775   43 00012 02123
HZ07    RA   HZ05   MM09         BUMP WD             45776   21 02116 02252
HZ08    IJ I   06   HZ05         TST BLOCK FILL      45777   41 00015 02116
HZ09    MJ 00000    EF21         TO WRITE            46000   45 00000 02077
HZ10    TU   HZ05   HZ12         SET UP              46001   15 02116 02125
HZ11    RP 30004    HZ13          AND OBTAIN         46002   75 30004 02126
HZ12    TP          BF00          WORDS              46003   11 00000 02300
HZ13    LQ   HZ12 31021          DELETE              46004   55 02125 31025
HZ14    TV Q        HZ15          FROM               46005   16 31000 02130
HZ15    TP   TS07                 TABLE              46006   11 00007 00000
HZ16    TV   BF01   HZ30         SET FIRST WORD      46007   16 02301 02147
HZ17    TP   BF01 Q              IF ODD              46010   11 02301 31000
HZ18    QJ   HZ21   HZ19           SET RPT 2         46011   44 02136 02134
HZ19    TU   VC00   HZ29          AND                46012   15 02004 02146
HZ20    MJ 00000    HZ29          STORE              46013   45 00000 02146
HZ21    TU   CR04   HZ29         IF EVEN SET RP3     46014   15 02000 02146
HZ22    SP   BF03     18         SPLIT LAST          46015   31 02303 00022
HZ23    AT   MM10   BF04         +BLANKS             46016   35 02253 02304
HZ24    LT 00000    BF03         SAVE                46017   22 00000 02303
HZ25    SP   BF02     18         SPLIT FIRST         46020   31 02302 00022
HZ26    AT   BF03   BF03         +PART               46021   35 02303 02303
```

```
HZ27    LT 00000 A            TOP                46022   22 00000 32000
HZ28    AT  MM11  BF02        +BLANKS            46023   35 02254 02302
HZ29    RP        HZ31        STORE AND          46024   75 00000 02150
HZ30    TP  BF02               TST INDEX         46025   11 02302 00000
HZ31    RS I  02  CP04        IF LAST CORR       46026   23 00011 00062
HZ32    ZJ  HZ03  EF20         TO FINAL MODE     46027   47 02114 02076
CD00    TV  MM18  CD32        SET FIRST STORE    46030   16 02263 02212
CD01    EF        MM12        START CD RDR       46031   17 00000 02255
CD02    TP  MM13 I 05         ROW IND TO 8       46032   11 02256 00014
CD03    RJ  CD31  CD24        TO RD DIGITS       46033   37 02211 02202
CD04    TP  CP05 I 05         ROW INDEX TO 2     46034   11 00063 00014
CD05    RJ  CD31  CD33        TO READ ZONE       46035   37 02211 02213
CD06    SP  BF04 00030         IF FIRST          46036   31 02304 00036
CD07    LT        A             WORD IS START    46037   22 00000 32000
CD08    EJ  MM14  VC34          DO NOT READ      46040   43 02257 02046
CD09    RA I  01  CP04        BUMP LINE CT       46041   21 00010 00062
CD10    RA  CD32  CP07        BUMP STORE         46042   21 02212 00065
CD11    MJ 00000  CD01        TO READ            46043   45 00000 02153
CD12    RP 30008  CD14        CLEAR WORKING      46044   75 30010 02170
CD13    RS  BF00  BF00         STORAGE           46045   23 02300 02300
CD14    ER 00000 A            READ               46046   76 00000 32000
CD15    ER 10000  AB00         CARD              46047   76 10000 01610
CD16    ER 10000 A             ROW               46050   76 10000 32000
CD17    RS  CD25  MM15        SET FOR ROW SUM    46051   23 02203 02260
CD18    TP  CP06 I  06        WD IND TO 3        46052   11 00064 00015
CD19    SP A        06        ENTER SUM          46053   31 32000 00006
CD20    LQ  AB00    01        POSIT COLUMN       46054   55 01610 00001
CD21    QA  CP04 A            ADD 1 IF PCH       46055   52 00062 32000
CD22    LQ  MM16    01        TEST END           46056   55 02261 00001
CD23    QJ  CD19               COMPUTER WD       46057   44 02175 00000
CD24    RJ  CD23  CD12        SET FOR DIGIT      46060   37 02201 02166
CD25    AT  BF04  BF04        ROW SUM            46061   35 02304 02304
CD26    RA  CD25  MM17        BUMP SUM ADD       46062   21 02203 02262
CD27    IJ I  06  CD19        IS ROW DONE        46063   41 00015 02175
CD28    RP 30004  CD30        TOTAL              46064   75 30004 02210
CD29    RA  BF04  BF00         SUM               46065   21 02304 02300
CD30    IJ I  05  CD14        IS CD DONE         46066   41 00014 02170
CD31    RP 30004              STORE              46067   75 30004 00000
CD32    TP  BF04               CARD              46070   11 02304 00000
CD33    RJ  CD23  CD14        SET FOR ZONE       46071   37 02201 02170
CD34    TU  CD25  CD36        SET LOGICAL MP     46072   15 02203 02216
CD35    TP A      Q           WORD TO Q          46073   11 32000 31000
CD36    QT        AB01        ONES COMMON        46074   51 00000 01611
CD37    RS Q      AB01         ONES NOT COMMO    46075   23 31000 01611
CD38    SP  AB01    04        ONES COM TO 20     46076   31 01611 00004
CD39    AT Q      A           +OTHERS            46077   35 31000 32000
CD40    MJ        CD25        TO SUM             46100   45 00000 02203
TR00    TP  MM18  TR05        SET FIRST STORE    46101   11 02263 02230
TR01    TU  CD20  TR05         AND BUFFER        46102   15 02176 02230
TR02  FP64   410  AB00        READ BLOCK         46103   14 64410 01610
TR03    TP  CP08 I  05        SET INDEX TO 5     46104   11 00066 00014
TR04    RP 30004  TR06        TRANSFER           46105   75 30004 02231
TR05                    B      BLOCKETTE         46106   00 00000 00000
TR06    TU  TR05  TR07        IF FIRST           46107   15 02230 02232
TR07    SP          30         WORD IS           46110   31 00000 00036
TR08    LT        A            START             46111   22 00000 32000
TR09    EJ  MM14  VC34         DO NOT READ       46112   43 02257 02046
TR10    RA  TR05  MM19        BUMP TRANSFER      46113   21 02230 02264
TR11    RA I  01  CP04        BUMP LINE COUNT    46114   21 00010 00062
```

```
      TR12     IJ I  05  TR04                TST END BLOCK       46115    41 00014 02227
      TR13     MJ        TR01                READ AGAIN          46116    45 00000 02224
      MM00                 77      B         MASK                46117    00 00000 00077
      MM01                 24      B         20:0                46120    00 00000 00024
      MM02     00 00000 01576      B          ADDRESS DUMMY      46121    00 00000 01576
      MM03     01 30152 00404      B         TAPEBB              46122    01 30152 00404
      MM04     24 05031 20104      B         SHORTB              46123    24 05031 20104
      MM05     13 03032 20404      B         GOODBB              46124    13 03032 20404
      MM06     02 00200 20000      B         REWIND 2            46125    02 00200 20000
      MM07     02 00200 30000      B                3            46126    02 00200 30000
      MM08     02 00200 40000      B                4            46127    02 00200 40000
      MM09     00 00004 00000      B         4 IN U              46130    00 00004 00000
      MM10     00 00000 10101      B         SPACES BOTTOM       46131    00 00000 10101
      MM11     01 01010 00000      B         SPACES TOP          46132    01 01010 00000
      MM12     40 00000 00005      B         START CD RDR        46133    40 00000 00005
      MM13                 10      B         8:0                 46134    00 00000 00010
      MM14     00 65662 45466      B          XS3 START          46135    00 65662 45466
      MM15     00 00004 00004      B         4IN U AND V         46136    00 00004 00004
      MM16     37 37373 73737      B         LOOP TESTER         46137    37 37373 73737
      MM17     00 00001 00001      B         1 IN U AND V        46140    00 00001 00001
      MM18     TP            BF08            DUMMY               46141    11 00000 02310
      MM19     00 00024 00004      B         ADV WD              46142    00 00024 00004
      MM20                 30      B         24:0                46143    00 00000 00030
      CS00     RP 30192  CR00                PROGRAM             46144    75 30300 01774
      CS01     TP  CS02  CR00                 TO CORE            46145    11 46146 01774
     START          CS00                                         46144
```

INDEX OF LIBRARY PROGRAMS

01 AUG 56

KEY TO CLASSIFICATION

B1	ELEMENTARY FUNCTIONS - TRIGONOMETRIC
B3	ELEMENTARY FUNCTIONS - EXPONENTIAL AND LOGARITHMIC
B4	ELEMENTARY FUNCTIONS - ROOTS AND POWERS
E1	APPROXIMATIONS - TABLE LOOK-UP AND INTERPOLATION
F0	MATRICES, VECTORS, AND SIMULTANEOUS LINEAR EQUATIONS
I1	INPUT - BINARY
I4	INPUT - COMPOSITE
J4	OUTPUT - COMPOSITE
K0	INFORMATION TRANSFER
L1	EXECUTIVE ROUTINES - ASSEMBLY
N0	DEBUGGING ROUTINES - TRAP MEMORY PRINT
N1	DEBUGGING ROUTINES - TRACING
N2	DEBUGGING ROUTINES - MEMORY DUMP
Z0	ALL OTHERS

INDEX OF TITLES

B1	UA	ATN1	0	ARC TANGENT SUBROUTINE
B1	UA	S+C1	0	SINE AND COSINE SUBROUTINE
B3	UA	EXP1	0	EXPONENTIAL SUBROUTINE
B3	UA	LN 2	0	NATURAL LOGARITHM SUBROUTINE
B4	UA	SQR3	0	SQUARE ROOT SUBROUTINE
B4	UA	SQR4	0	SQUARE ROOT SUBROUTINE
E1	UA	BPE1	0	BIVARIATE POLYNOMIAL EVALUATION SUBROUTINE
E1	UA	BPE3	0	BIVARIATE TABLE INTERPOLATION
F0	UA	INV1	0	MATRIX INVERSION
I1	UA	CSB1	0	ABSOLUTE BINARY LOADER
I1	UA	CSB2	0	ABSOLUTE BINARY LOADER
I4	UA	DBC1	0	DECIMAL, OCTAL, BCD LOADER
J4	UA	BDC1	0	GENERALIZED PRINT PROGRAM
K0	UA	CCB1	0	BINARY CHECK SUM CORRECTOR
K0	UA	CSH2	0	READ BCD TAPE OR ON-LINE CARD READER
K0	UA	CTH1	0	OFF-LINE CARD READER SIMULATOR
K0	UA	RWD2	0	READ-WRITE DRUM PROGRAM
K0	UA	SPH1	0	BCD OUTPUT PROGRAM
K0	UA	STH1	0	BCD TAPE WRITING PROGRAM
K0	UA	TCH1	0	OFF-LINE PUNCH SIMULATOR
K0	UA	TPH1	0	OFF-LINE PRINTER SIMULATOR
K0	UA	TSB3	0	LOAD BINARY CARD IMAGES FROM TAPE
K0	UA	TSM2	0	READ TAPE WITH REDUNDANCY CHECKING
L1	UA	SAP1	0	SHARE ASSEMBLER
N0	UA	SPM1	0	TRAP DECIMAL MEMORY PRINT
N1	UA	SPO2	0	FLOW TRACE
N2	UA	SPO1	0	CONTROL PANEL PRINT AND OCTAL MEMORY PRINT - (SCOOP)
Z0	UA	OTM1	0	TAPE END-OF-FILE AND/OR REWIND

ZO UA OTM2 0 TAPE REWIND CONTROL
ZO UA OTM3 0 WRITE END-OF-FILE AND/OR REWIND N TAPES
ZO UA OTM4 0 TAPE REWIND CONTROL
ZO UA OTM5 0 BEGINNING-OF-JOB PRINTER TAPE MARK
ZO UA OTM6 0 PRINTER OUTPUT TAPE POSITIONER
ZO UA OTM7 0 BEGINNING-OF-JOB PRINTER TAPE MARK
ZO UA OTM8 0 BEGINNING-OF-JOB PUNCH TAPE MARK
ZO UA OTM9 0 PUNCH OUTPUT TAPE POSITIONER
ZO UA OTMA 0 WRITE JOB TITLE AND BEGINNING-OF-JOB PRINTER TAPE MARK
ZO UA PCS1 0 PUNCH DRUM CHECK SUM VERIFIER
ZO UA RWD1 0 READ-WRITE DRUM
ZO UA VCS1 0 VERIFY DRUM CHECK SUM
ZO UA ZCS1 0 CLEAR CORE STORAGE AND MAIN FRAME
ZO UA ZCS2 0 SET CORE STORAGE TO ZERO
ZO UA ZDR1 0 CLEAR N DRUMS

INDEX OF PROGRAMS

ELEMENTARY FUNCTIONS - TRIGONOMETRIC

B1 UA ATN1 0 ARC TANGENT SUBROUTINE
B1 UA ATN1 1 COMPUTES PRINCIPAL VALUE OF THE ARC TANGENT. RESULT IS GIVEN
B1 UA ATN1 2 IN RADIANS. USES RAND POLYNOMIAL. ROUTINE REQUIRES 39 CELLS
B1 UA ATN1 3 PLUS 3 COMMON. TIMING ABOUT 4.2 MS.

B1 UA S+C1 0 SINE AND COSINE SUBROUTINE
B1 UA S+C1 1 COMPUTES THE SINE OR COSINE OF AN ANGLE GIVEN IN RADIANS.
B1 UA S+C1 2 USES RAND POLYNOMIAL. ROUTINE REQUIRES 61 CELLS PLUS 2
B1 UA S+C1 3 COMMON. TIMING ABOUT 3 MS.

ELEMENTARY FUNCTIONS - EXPONENTIAL AND LOGARITHMIC

B3 UA EXP1 0 EXPONENTIAL SUBROUTINE
B3 UA EXP1 1 COMPUTES E TO THE X FOR ALL X LESS THAN 72. USES ECONOMIZED
B3 UA EXP1 2 POLYNOMIAL. ROUTINE REQUIRES 45 CELLS PLUS 2 COMMON. TIMING
B3 UA EXP1 3 ABOUT 3.1 MS.

B3 UA LN 2 0 NATURAL LOGARITHM SUBROUTINE
B3 UA LN 2 1 COMPUTES THE NATURAL LOGARITHM OF THE ABSOLUTE VALUE OF X.
B3 UA LN 2 2 USES A RAND POLYNOMIAL. ROUTINE REQUIRES 55 CELLS PLUS 3
B3 UA LN 2 3 COMMON. TIMING ABOUT 4 MS.

ELEMENTARY FUNCTIONS - ROOTS AND POWERS

```
B4 UA SQR3 0  SQUARE ROOT SUBROUTINE
B4 UA SQR3 1  COMPUTES THE SQUARE ROOT OF THE ABSOLUTE VALUE OF X.  ROUTINE
B4 UA SQR3 2  REQUIRES 22 CELLS PLUS 4 COMMON.  TIMING 1.4 MS.

B4 UA SQR4 0  SQUARE ROOT SUBROUTINE
B4 UA SQR4 1  COMPUTES THE SQUARE ROOT OF THE ABSOLUTE VALUE OF X.  AN
B4 UA SQR4 2  ERROR RETURN RESULTS IF THE RADICAND IS NEGATIVE.  ROUTINE
B4 UA SQR4 3  REQUIRES 25 CELLS PLUS 4 COMMON.  TIMING 1.5 MS.
```

APPROXIMATIONS - TABLE LOOK-UP AND INTERPOLATION

```
E1 UA BPE1 0  BIVARIATE POLYNOMIAL EVALUATION SUBROUTINE
E1 UA BPE1 1  EVALUATES A BIVARIATE POLYNOMIAL OF TOTAL DEGREE N.  ROUTINE
E1 UA BPE1 2  REQUIRES 36 CELLS PLUS 4 COMMON.  EACH POLYNOMIAL REQUIRES
E1 UA BPE1 3  ((N+2)(N+1)/2)+5 CELLS FOR COEFFICIENT STORAGE.  NO OUT-OF-
E1 UA BPE1 4  RANGE TEST IS PERFORMED.  TIMING 0.18N*N+0.72N+1.056 MS.

E1 UA BPE3 0  BIVARIATE TABLE INTERPOLATION
E1 UA BPE3 1  SELECTS A SUITABLE 3 BY 3 ARRAY OF FUNCTION VALUES FROM AN
E1 UA BPE3 2  M BY N ARRAY IN CORE.  PERFORMS BIVARIATE INTERPOLATION IN
E1 UA BPE3 3  THIS 3 BY 3 ARRAY BY FITTING AND EVALUATING 3 QUADRATICS IN
E1 UA BPE3 4  Y, FOLLOWED BY ONE IN X.  AN OUT-OF-RANGE ERROR RETURN IS
E1 UA BPE3 5  PROVIDED.  ROUTINE REQUIRES 161 CELLS PLUS 12 COMMON.  VALUES
E1 UA BPE3 6  OF X, Y, AND Z REQUIRE M+N+M*N CELLS.  TIMING AVERAGES ABOUT
E1 UA BPE3 7  16 MS FOR A 10 BY 10 ARRAY.
```

MATRICES, VECTORS, AND SIMULTANEOUS LINEAR EQUATIONS

```
F0 UA INV1 0  MATRIX INVERSION
F0 UA INV1 1  INVERTS A MATRIX STORED IN CORE STORAGE.  USES AN ELIMINATION
F0 UA INV1 2  METHOD.  THE STARRING ELEMENT IS THE LARGEST IN THE COLUMN,
F0 UA INV1 3  BUT THE COLUMNS ARE USED IN ORDER FROM LEFT TO RIGHT.  THE
F0 UA INV1 4  ORIGINAL MATRIX IS DESTROYED, AND IS REPLACED IN STORAGE BY
F0 UA INV1 5  THE INVERSE.  THE ROUTINE REQUIRES 171 CELLS PLUS 2N+8
F0 UA INV1 6  COMMON.  A 61 BY 61 MATRIX CAN BE INVERTED IN A 4096 WORD
F0 UA INV1 7  MACHINE IN ABOUT 100 SECONDS.
```

INPUT - BINARY

```
I1 UA CSB1 0  ABSOLUTE BINARY LOADER
I1 UA CSB1 1  LOADS SHARE STANDARD NON-RELOCATABLE BINARY INFORMATION CARDS
I1 UA CSB1 2  AND TRANSFER CARDS.  ALSO LOADS PROGRAM CORRECTIONS PUNCHED
I1 UA CSB1 3  IN ROWS 8 TO 12 OF TRANSFER CARDS.  PROGRAM IS SELF-LOADING
I1 UA CSB1 4  AND USES CELLS 0 TO 71.

I1 UA CSB2 0  ABSOLUTE BINARY LOADER
I1 UA CSB2 1  LOADS SHARE STANDARD NON-RELOCATABLE BINARY INFORMATION CARDS
I1 UA CSB2 2  AND TRANSFER CARDS.  ALSO LOADS PROGRAM CORRECTIONS PUNCHED
I1 UA CSB2 3  IN ROWS 8 TO 12 OF TRANSFER CARDS.  PROGRAM IS NOT SELF-
I1 UA CSB2 4  LOADING.  PROGRAM REQUIRES 54 CELLS PLUS 3 COMMON.
```

INPUT - COMPOSITE

```
I4 UA DBC1 0  DECIMAL, OCTAL, BCD LOADER
I4 UA DBC1 1  USED WITH UA TSM 2 OR UA CSH 2.  CONTROLS TAPE PROGRAM UA TSM
I4 UA DBC1 2  2 OR TAPE OR CARD PROGRAM UA CSH 2 TO READ BCD INFORMATION
I4 UA DBC1 3  INTO CORE.  CONVERTS THIS INFORMATION TO BINARY, - FIXED OR
I4 UA DBC1 4  FLOATING DECIMAL NUMBERS BEING CONVERTED TO FIXED OR FLOATING
I4 UA DBC1 5  BINARY NUMBERS, AND DECIMAL OR OCTAL INTEGERS BEING CONVERTED
I4 UA DBC1 6  TO BINARY INTEGERS.  ALSO READS AND STORES HOLLERITH LABELS,
I4 UA DBC1 7  COMMENTS, ETC.  INPUT CARD FORMAT IS VARIABLE.  LOADING MAY
I4 UA DBC1 8  BE CONTROLLED BY TRANSFER CARDS.  ROUTINE REQUIRES 372 CELLS
I4 UA DBC1 9  PLUS 24 COMMON.
```

OUTPUT - COMPOSITE

```
J4 UA BDC1 0  GENERALIZED PRINT PROGRAM
J4 UA BDC1 1  USED WITH UA STH 1 OR UA SPH 1.  CONVERTS FLOATING BINARY
J4 UA BDC1 2  NUMBERS TO FIXED OR FLOATING BINARY CODED DECIMAL NUMBERS.
J4 UA BDC1 3  CONVERTS BINARY INTEGERS TO BINARY CODED DECIMAL INTEGERS.
J4 UA BDC1 4  ARRANGES THIS BINARY CODED DECIMAL INFORMATION IN ANY ARBI-
J4 UA BDC1 5  TRARY FORMAT, INTERSPERSING IT AS SPECIFIED WITH HOLLERITH
J4 UA BDC1 6  LABELS, COMMENTS, ETC.  CONTROLS TAPE PROGRAM UA STH 1 OR
J4 UA BDC1 7  TAPE AND/OR PRINTER PROGRAM UA SPH 1 TO PRODUCE THE FINAL
J4 UA BDC1 8  PRINTED RESULT.  ROUTINE REQUIRES 402 CELLS PLUS 60 COMMON.
```

INFORMATION TRANSFER

```
KO UA CCB1 0   BINARY CHECK SUM CORRECTOR
KO UA CCB1 1   READS, ONE AT A TIME, ANY NUMBER OF NON-RELOCATABLE BINARY
KO UA CCB1 2   CARDS, RECOMPUTES THE CARD CHECK SUMS, AND PUNCHES OUT
KO UA CCB1 3   CORRECTED CARDS.

KO UA CSH2 0   READ BCD TAPE OR ON-LINE CARD READER
KO UA CSH2 1   READS EITHER BCD TAPE (WITH REDUNDANCY CHECKING) OR HOLLERITH
KO UA CSH2 2   PUNCHED CARDS, AS DETERMINED BY SENSE SWITCH.  INFORMATION
KO UA CSH2 3   READ IS STORED IN CORE IN BCD FORM.  ROUTINE REQUIRES 167
KO UA CSH2 4   CELLS PLUS 9 COMMON.

KO UA CTH1 0   OFF-LINE CARD READER SIMULATOR
KO UA CTH1 1   SIMULATES THE OFF-LINE CARD-TO-TAPE MACHINE.  THE TAPE IS NOT
KO UA CTH1 2   CHECKED FOR WRITE ERRORS.

KO UA RWD2 0   READ-WRITE DRUM PROGRAM
KO UA RWD2 1   WRITES A BLOCK OF INFORMATION FROM CORE TO DRUM, OR READS A
KO UA RWD2 2   BLOCK OF INFORMATION FROM DRUM TO CORE.  READING IS CHECKED
KO UA RWD2 3   BY AN ACL CHECK SUM FORMED AND RECORDED DURING WRITING.
KO UA RWD2 4   ROUTINE REQUIRES 30 CELLS PLUS 3 COMMON.

KO UA SPH1 0   BCD OUTPUT PROGRAM
KO UA SPH1 1   WRITES A BCD RECORD ON TAPE AND/OR PRINTS IT ON THE ON-LINE
KO UA SPH1 2   PRINTER, AS DETERMINED BY SENSE SWITCHES.  ROUTINE REQUIRES
KO UA SPH1 3   109 CELLS PLUS 33 COMMON.

KO UA STH1 0   BCD TAPE WRITING PROGRAM
KO UA STH1 1   WRITES A BCD RECORD OF ARBITRARY LENGTH ON TAPE.  ROUTINE
KO UA STH1 2   REQUIRES 15 CELLS, NO COMMON.

KO UA TCH1 0   OFF-LINE PUNCH SIMULATOR
KO UA TCH1 1   SIMULATES THE OFF-LINE TAPE-TO-CARD MACHINE.  72 COLUMNS ARE
KO UA TCH1 2   PUNCHED PER CARD.  THE TAPE IS REDUNDANCY-CHECKED, BUT THE
KO UA TCH1 3   CARDS ARE NOT CHECKED AT THE PUNCH BRUSH STATION.

KO UA TPH1 0   OFF-LINE PRINTER SIMULATOR
KO UA TPH1 1   SIMULATES THE TAPE-TO-OFF-LINE-PRINTER MACHINE.  THE TAPE IS
KO UA TPH1 2   REDUNDANCY-CHECKED, BUT NO PRINT WHEEL ECHO CHECKING IS
KO UA TPH1 3   PERFORMED.  THE CARRIAGE CONTROL (PROGRAM DEVICE) IS NOT
KO UA TPH1 4   SIMULATED.  PRINTER OPERATES AT HALF SPEED FOR RECORDS
KO UA TPH1 5   CONTAINING MORE THAN 72 CHARACTERS.

KO UA TSB3 0   LOAD BINARY CARD IMAGES FROM TAPE
KO UA TSB3 1   READS BINARY CARD IMAGES CONTAINING INSTRUCTIONS AND/OR DATA
KO UA TSB3 2   FROM TAPE INTO CORE AND, UPON ENCOUNTERING THE IMAGE OF A
KO UA TSB3 3   TRANSFER CARD, INITIATES THE EXECUTION OF THE PROGRAM THUS
KO UA TSB3 4   LOADED.  ONLY THE IMAGES ON TAPE OF SHARE STANDARD BINARY
KO UA TSB3 5   INFORMATION CARDS AND TRANSFER CARDS CAN BE HANDLED.

KO UA TSM2 0   READ TAPE WITH REDUNDANCY CHECKING
KO UA TSM2 1   READS, WITH REDUNDANCY CHECKING, ANY TAPE WRITTEN BY OR FOR
KO UA TSM2 2   THE 704.  THE ENTIRE RECORD MUST BE READ AND STORED.  ROUTINE
KO UA TSM2 3   REQUIRES 26 CELLS, NO COMMON.
```

EXECUTIVE ROUTINES - ASSEMBLY

```
L1 UA SAP1 0  SHARE ASSEMBLER
L1 UA SAP1 1  ASSEMBLES PROGRAMS WRITTEN IN SYMBOLIC FORM.  INPUT AND OUT-
L1 UA SAP1 2  PUT MAY BE EITHER OFF-LINE OR ON.  PRINTED OUTPUT INCLUDES
L1 UA SAP1 3  THE GIVEN PROGRAM IN SYMBOLIC AND THE ASSEMBLED PROGRAM IN
L1 UA SAP1 4  OCTAL.  OUTPUT IS ALSO PUNCHED ON NON-RELOCATABLE BINARY
L1 UA SAP1 5  CARDS, ON RELOCATABLE BINARY CARDS, ON BINARY CARDS CONTAIN-
L1 UA SAP1 6  ING 24 WORDS EACH, OR IT MAY BE WRITTEN ON TAPE IN NON-
L1 UA SAP1 7  RELOCATABLE BINARY CARD IMAGE FORM.  DECIMAL, OCTAL AND
L1 UA SAP1 8  HOLLERITH DATA MAY BE USED.  A LIBRARY OF STANDARD ROUTINES
L1 UA SAP1 9  IS AVAILABLE ON TAPE.  ADDRESS ARITHMETIC MAY BE PERFORMED.
```

DEBUGGING ROUTINES - TRAP MEMORY PRINT

```
NO UA SPM1 0  TRAP DECIMAL MEMORY PRINT
NO UA SPM1 1  PRINTS, IN FLOATING DECIMAL, OFF-LINE AND/OR ON-LINE, THE
NO UA SPM1 2  CONTENTS OF ANY NUMBER OF BLOCKS OF CORE STORAGE.  PRINTING
NO UA SPM1 3  MAY BE PERFORMED DURING THE EXECUTION OF THE PROGRAM, WITHOUT
NO UA SPM1 4  OTHERWISE AFFECTING THE ACTION OF THE PROGRAM IN ANY WAY.
NO UA SPM1 5  PRINTING IS SPECIFIED BY CONTROL CARDS, EACH BLOCK BEING
NO UA SPM1 6  PRINTED WHEN A SELECTED INSTRUCTION HAS BEEN EXECUTED A
NO UA SPM1 7  DESIGNATED NUMBER OF TIMES.  PRINTING MAY ALSO BE PERFORMED
NO UA SPM1 8  AFTER THE PROGRAM HAS STOPPED.  THE ROUTINE IS USUALLY STORED
NO UA SPM1 9  ON A DRUM AND READ INTO CORE STORAGE CELLS 0-511 WHEN NEEDED.
```

DEBUGGING ROUTINES - TRACING

```
N1 UA SPO2 0  FLOW TRACE
N1 UA SPO2 1  PRINTS THE INSTRUCTION, ITS LOCATION, AND THE CONTENTS OF AC
N1 UA SPO2 2  MQ, AND INDEX REGISTERS, EACH TIME A TRANSFER IS EXECUTED.
N1 UA SPO2 3  TRACING IS CONTINUOUS, BUT OUTPUT PRINTING STARTS ONLY AFTER
N1 UA SPO2 4  A DESIGNATED TRANSFER HAS BEEN EXECUTED A SPECIFIED NUMBER O
N1 UA SPO2 5  TIMES, AND IS THEN UNDER SENSE SWITCH CONTROL.  OUTPUT (IN
N1 UA SPO2 6  OCTAL) MAY BE OBTAINED OFF-LINE AND/OR ON-LINE.  THE ROUTINE
N1 UA SPO2 7  IS USUALLY STORED ON A DRUM AND READ INTO CORE STORAGE CELLS
N1 UA SPO2 8  0-457 WHEN NEEDED.
```

DEBUGGING ROUTINES - MEMORY DUMP

```
N2 UA SP01 0   CONTROL PANEL PRINT AND OCTAL MEMORY PRINT - (SCOOP)
N2 UA SP01 1   PRINTS ALL CONTROL PANEL INFORMATION EXCEPT C(MQ) AND
N2 UA SP01 2   C(PROGRAM COUNTER). PRINTS, IN OCTAL, THE CONTENTS OF UP TO
N2 UA SP01 3   24 CORE STORAGE REGIONS. OUTPUT MAY BE OBTAINED OFF-LINE
N2 UA SP01 4   AND/OR ON-LINE. THE ROUTINE IS USUALLY STORED ON A DRUM AND
N2 UA SP01 5   READ INTO CORE STORAGE CELLS 0-476 WHEN NEEDED.
```

ALL OTHERS

```
Z0 UA OTM1 0   TAPE END-OF-FILE AND/OR REWIND
Z0 UA OTM1 1   WRITES END-OF-FILE AND/OR REWINDS ANY ONE OF THE TEN TAPES
Z0 UA OTM1 2   UNDER CONTROL OF THE SENSE SWITCHES. SELF-LOADING.

Z0 UA OTM2 0   TAPE REWIND CONTROL
Z0 UA OTM2 1   TESTS BITS 1-10 OF WORD IN 7 RIGHT ROW AND REWINDS TAPES IF
Z0 UA OTM2 2   CORRESPONDING BITS ARE PRESENT. LOADED BY UA CSB 1.

Z0 UA OTM3 0   WRITE END-OF-FILE AND/OR REWIND N TAPES
Z0 UA OTM3 1   WRITES END-OF-FILE AND/OR REWINDS ANY ONE OR MORE OF THE TEN
Z0 UA OTM3 2   TAPES UNDER CONTROL OF THE SENSE SWITCHES. SELF-LOADING.

Z0 UA OTM4 0   TAPE REWIND CONTROL
Z0 UA OTM4 1   TESTS BITS 1-10 OF WORD IN 7 RIGHT ROW AND REWINDS TAPES IF
Z0 UA OTM4 2   CORRESPONDING BITS ARE PRESENT. SELF-LOADING.

Z0 UA OTM5 0   BEGINNING-OF-JOB PRINTER TAPE MARK
Z0 UA OTM5 1   WRITES THE PRINTER BEGINNING-OF-JOB MARK ON LOGICAL TAPE 2.
Z0 UA OTM5 2   SELF-LOADING.

Z0 UA OTM6 0   PRINTER OUTPUT TAPE POSITIONER
Z0 UA OTM6 1   POSITIONS THE PRINTER OUTPUT TAPE AT THE M-TH 1-EJECT
Z0 UA OTM6 2   FOLLOWING THE N-TH 9-EJECT. SELF-LOADING. M AND N ARE
Z0 UA OTM6 3   PUNCHED IN THE 12 RIGHT ROW.

Z0 UA OTM7 0   BEGINNING-OF-JOB PRINTER TAPE MARK
Z0 UA OTM7 1   WRITES THE PRINTER BEGINNING-OF-JOB MARK ON LOGICAL TAPE 2.
Z0 UA OTM7 2   LOADED BY UA CSB 1.

Z0 UA OTM8 0   BEGINNING-OF-JOB PUNCH TAPE MARK
Z0 UA OTM8 1   WRITES THE PUNCH BEGINNING-OF-JOB MARK ON LOGICAL TAPE 4.
Z0 UA OTM8 2   SELF-LOADING.

Z0 UA OTM9 0   PUNCH OUTPUT TAPE POSITIONER
Z0 UA OTM9 1   POSITIONS THE PUNCH OUTPUT TAPE AT THE N-TH BEGINNING-OF-JOB
Z0 UA OTM9 2   MARK. SELF-LOADING. N IS PUNCHED IN THE 12 RIGHT ROW.

Z0 UA OTMA 0   WRITE JOB TITLE AND BEGINNING-OF-JOB PRINTER TAPE MARK
Z0 UA OTMA 1   WRITES ANY SPECIFIED SINGLE LINE JOB TITLE AND THE PRINTER
Z0 UA OTMA 2   BEGINNING-OF-JOB MARK ON LOGICAL TAPE 2. SELF-LOADING.

Z0 UA PCS1 0   PUNCH DRUM CHECK SUM VERIFIER
Z0 UA PCS1 1   COMPUTES THE ACL CHECK SUM OF N CONSECUTIVE DRUMS AND PUNCHES
Z0 UA PCS1 2   THIS CHECK SUM INTO AN APPROPRIATE VERSION OF UA VCS 1.

Z0 UA RWD1 0   READ-WRITE DRUM
Z0 UA RWD1 1   PRODUCES A WRITE-DRUM CARD (UA CSB CORRECTION/TRANSFER TYPE)
Z0 UA RWD1 2   AND A READ-DRUM CARD (SELF-LOADING). THE WRITE-DRUM CARD
```

```
ZO UA RWD1 3  WRITES A BLOCK OF INFORMATION FROM CORE TO DRUM.  THE
ZO UA RWD1 4  READ-DRUM CARD READS THE BLOCK BACK INTO CORE, VERIFIES ITS
ZO UA RWD1 5  ACL CHECK SUM, AND TRANSFERS TO THE DESIGNATED INSTRUCTION.

ZO UA VCS1 0  VERIFY DRUM CHECK SUM
ZO UA VCS1 1  THIS ROUTINE, USED IN CONJUNCTION WITH UA PCS 1, PRODUCES A
ZO UA VCS1 2  ONE-CARD SELF-LOADING PROGRAM WHICH MAY THEN BE USED TO
ZO UA VCS1 3  VERIFY THE ACCURACY OF INFORMATION STORED ON N CONSECUTIVE
ZO UA VCS1 4  DRUMS.  VERIFICATION IS BY MEANS OF ACL CHECK SUMS.

ZO UA ZCS1 0  CLEAR CORE STORAGE AND MAIN FRAME
ZO UA ZCS1 1  RESETS THE DIVIDE CHECK, TAPE CHECK AND MQ OVERFLOW INDI-
ZO UA ZCS1 2  CATORS, TURNS OFF THE SENSE LIGHTS, CLEARS INDEX REGISTERS 1
ZO UA ZCS1 3  AND 2, AND CLEARS ALL OF CORE EXCEPT 12-71.  LOADED BY BINARY
ZO UA ZCS1 4  LOADER UA CSB 1.

ZO UA ZCS2 0  SET CORE STORAGE TO ZERO
ZO UA ZCS2 1  RESETS THE TRAP MODE, AC OVERFLOW, MQ OVERFLOW, TAPE CHECK
ZO UA ZCS2 2  AND DIVIDE CHECK INDICATORS, TURNS OFF THE SENSE LIGHTS,
ZO UA ZCS2 3  CLEARS ALL OF CORE EXCEPT 4-23, AND CLEARS THE ACCUMULATOR
ZO UA ZCS2 4  AND ALL INDEX REGISTERS.  SELF-LOADING.

ZO UA ZDR1 0  CLEAR N DRUMS
ZO UA ZDR1 1  PRODUCES A CORRECTION/TRANSFER CARD WHICH, WHEN LOADED BY UA
ZO UA ZDR1 2  CSB 1, CLEARS ANY N CONSECUTIVE DRUMS.
```

115 FORMULAS WERE INVOLVED

1223 INSTRUCTIONS WERE GENERATED

COMPILATION TOOK BETWEEN TWO AND THREE MINUTES

```
  A119=A31*(A58*A56/A51)*(A36-A39-A36*A51/A56)
  B119=A32*(A58*A56/A51)*(A37-A40-A37*A51/A56)
  C119=A29*(A58*A56/A51)*(D1-A51/A56)
  D119=A30*A58*A56/A51
0 E119=(A58*A56/A51)*((A19*A3)*(A36-A39-A36*A51/A56)+(A20*A32)*
1 (A37-A40-A37*A51/A56)-(A23*A29)*(D1-A51/A56)+A24*A30)
  P120=A57*A59/A52
  A120=A31*P120*(D1-A52/A57)
  B120=A32*P120
  C120=A29*P120*(A42*(D1-A52/A57)+A25*A33*A38)
  D120=A30*P120*(A43-A46-A43*A52/A57+A41*A25*A33)
  E120=A33*P120
  F120=A44-A47+A24*A30*A41-A52*A44/A57+A38*A23*A29
0 H120=P120*(-A19*A31*(D1-A52/A57)+A20*A32+A23*A29*A42*(D1-A52/
1 A57)+A30*A24*(A43-A46-A43*A52/A57)+A25*A33*(A44-A47+A24*A30*
2 A41+A38*A23*A29-A52*A44/A57))
  A123=A67*A79*A80/A81
  B123=(A13+A15)*A79*A67/A81
  C123=A31*A64/A77
  D123=A32*A65/A77
  G123=A30*A66/A77
  P123=A34/A77
  Q123=A35*A34/A77
0 R123=((-A79*A78*A67-A18*A80*A67*A79)/A81+(A94*A35*A34+A19*A31
1 *A64+A20*A32*A65+A23*A29*A63+A30*A24*A66-A28*A34)/A77)
  F124=A51*A77
  A124=A31*(A36-A39)/F124
  B124=A32*(A37-A40)/F124
  C124=A29/F124
  D124=A30/F124
0 E124=(A19*A31*(A36-A39)+A20*A32*(A37-A40)-A23*A29+A24*A30)/
1 F124
  G125=A77*A82
  A125=A29/G125
  B125=A31*A36/G125
  E125=A53/A82
  A126=A31/R126
  B126=A32/R126
  C126=(A42+A33*A38*A25)*A29/R126
  F126=A29*A38
  G126=A30*A41
0 P126=(A32*A20-A19*A31+A23*A29*A42+A24*A30*(A43-A46)+A33*A25*(
1 A44-A47+A38*A23*A29+A24*A30*A41))/R126
  G127=A77*A83
  A127=A31/G127
  B127=A42*A29/G127
  C127=A30*A43/G127
  D127=A33*A44/G127
  E127=A60/A83
  F127=(A19*A31-A42*A29*A29-A30*A24*A43-A33*A44*A25)/G127
  P128=A77*A83
  A128=A33/P128
  B128=A29*A48/P128
```

```
        C128=A31*A41/P128
        D128=A32*A50/P128
        E128=A30*A62/P128
        F128=A54/A83
0       G128=(A25*A33-A23*A29*A48-A19*A31*A49-A20*A32*A50-A24*A30*A62
1       )/P128
        A129=A84*A72
        C129=A84*(A72*A74+A90)
        A130=A72*A85
        C130=A85*(A72*A74+A90)
        A141=D 7*A71*A79*A88*A89
        B141=D 7*A79*A88*A89
        C141=D 7*A26*A71*A79*A88*A89
        A1=I1
        A2A=I2S
        B2=I2
        A3S=I3S
        B3=I3
        A5=I5
        A6S=I6S
        B6=I6
        A7S=I7S
        B7=I7
        A8S=I8S
        B8=I8
        A9=I9
        A10=I10
        A11=I11
        A12=I12
        A2=A2S*A91+B2
        A3=A3S*A95+B3
        A6=A6S+A92+B6
        A7=A7S*A96+B7
        A8=A8S*A93+B8
        Z119=A2*A119+A3*B119-A6*C119+A7*D119-E119
0       Z120=-A2*A120+A3*B120+A6*C120+A7*D120+A8*E120*(F120-A6*A29*
1       A38-A7*A30*A41)-H120
        A121=A67*((A100-A78)*A79)/A81
0       Z123=A1*A123+(A13+A15)*B123-A2*C123-A3*D123-A6*F123-A7*G123
1       +A12*P123-A10*Q123+R123
        Z124=A2*A124+A3*B124-A6*C124+A7*D124-E124
        Z125=A6*A125-A2*B125-A3*C125+D125+Z124*E125+Z123*F125
0       Z126=A8*A33/(A52*A77)*((E126)-(F126)*A6-A7*G126)-P126-A2*
1       A126+A3*B126+A6*C126+A7*D126
        Z127=A2*A127-A6*B127-A7*C127-A8*D127+Z126*E127-F127
        Z128=A8*A128-A6*B128-A2*C128-A3*D128-A7*E128+Z126*F128-G128
        Z129=A10*A129+Z119-C129
        Z130=-A10*A130+Z120+C130
        X=COSF(Z129)
        Y=SINF(Z129)
        Z131=Z127*X-Z128*Y
        Z132=Z128*X+Z127*Y
        Z133=Z124*X-Z123*Y
        Z134=Z123*X+Z124*Y
        Z141=A9*A141+(A13+A14)*B141-C141
        A135=((A5-A22)*A70+A13+A15)/((A9-A26)*A71+A13+A14)
        A139=D 7*A71*A79
        B139=D 7*A79
        C139=D 7*A71*A79*A26
0       Z139=A9*A137*Z136*Z136*A139+(A13+A14)*A137*Z136*Z136*B139-
```

```
 1         A137*Z136*Z136*C139
           A143=(A11*A73+A13+A15-A27*A73)/(A9*A71+A13+A14-A26*A27)
           Z136=A135*SL+B
           X136=Z136*ALP+BET
           X143=A143*ALP+BET
 0         A137=C0+X136*(C1+X136*(C2+X136*(C3+X136*(C4+X136*(C5+X136*
 1         C6)))))
 0         Z143=C0+X143*(C1+X143*(C2+X143*(C3+X143*(C4+X143*(C5+X143*
 1         C6)))))
```

```
        REM         CODING BY ALGEBRAIC TRANSLATION SECTION OF FORTRAN
        REM        Z = -A + (B-C*D)*(E*SINF(B-C*D))*(F-C*D)/(R0 + X*(R1 + X*(R2
      1 REM               + X*(R3 + X*R4/Z4) /Z3) /Z2) /Z1)
1.      CLA X
        FDP Z4
        FMP R4
        FAD R3
        FDP Z3
        FMP X
        FAD R2
        FDP Z2
        FMP X
        FAD R1
        FDP Z1
        FMP X
        FAD R0
        STO 1S+8           DENOMINATOR
        LDQ C
        FMP D
        STO 1S+7           C * D
        CHS
        FAD F
        STO 1S+6           F - C * D
        CLA B
        FSB 1S+7
        STO 1S+5           B - C * D
        SXD 1S+4,4
        TSX SIN,4
        LXD 1S+4,4
        STO 1S+4           sin (B - C * D)
        LDQ E
        FMP 1S+4
        STO 1S+3           E sin (B - C * D)
        LDQ 1S+3
        FMP 1S+5
        FDP 1S+8
        FMP 1S+6
        FSB A
        STO Z
1
```

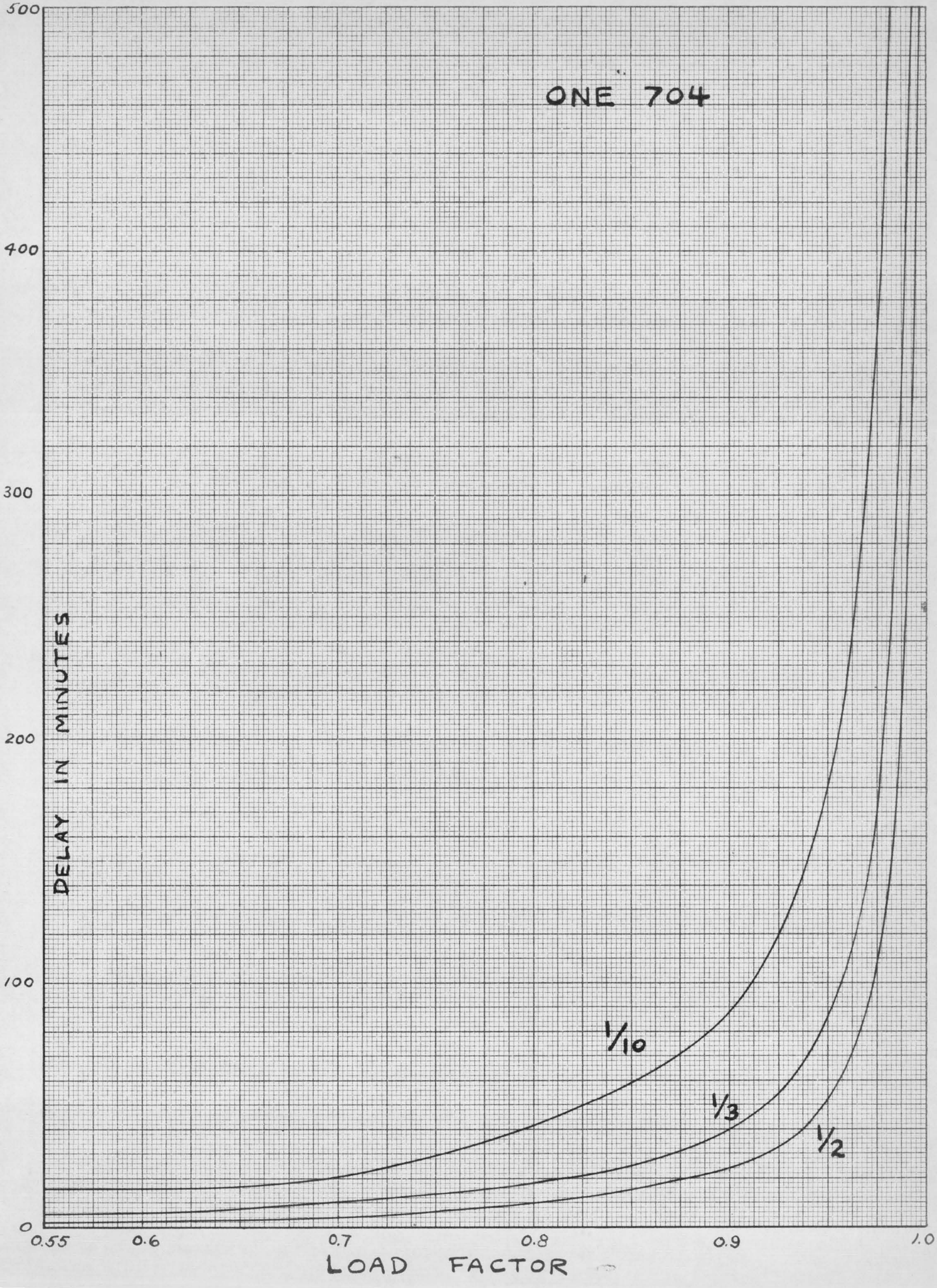
Stacking Analysis

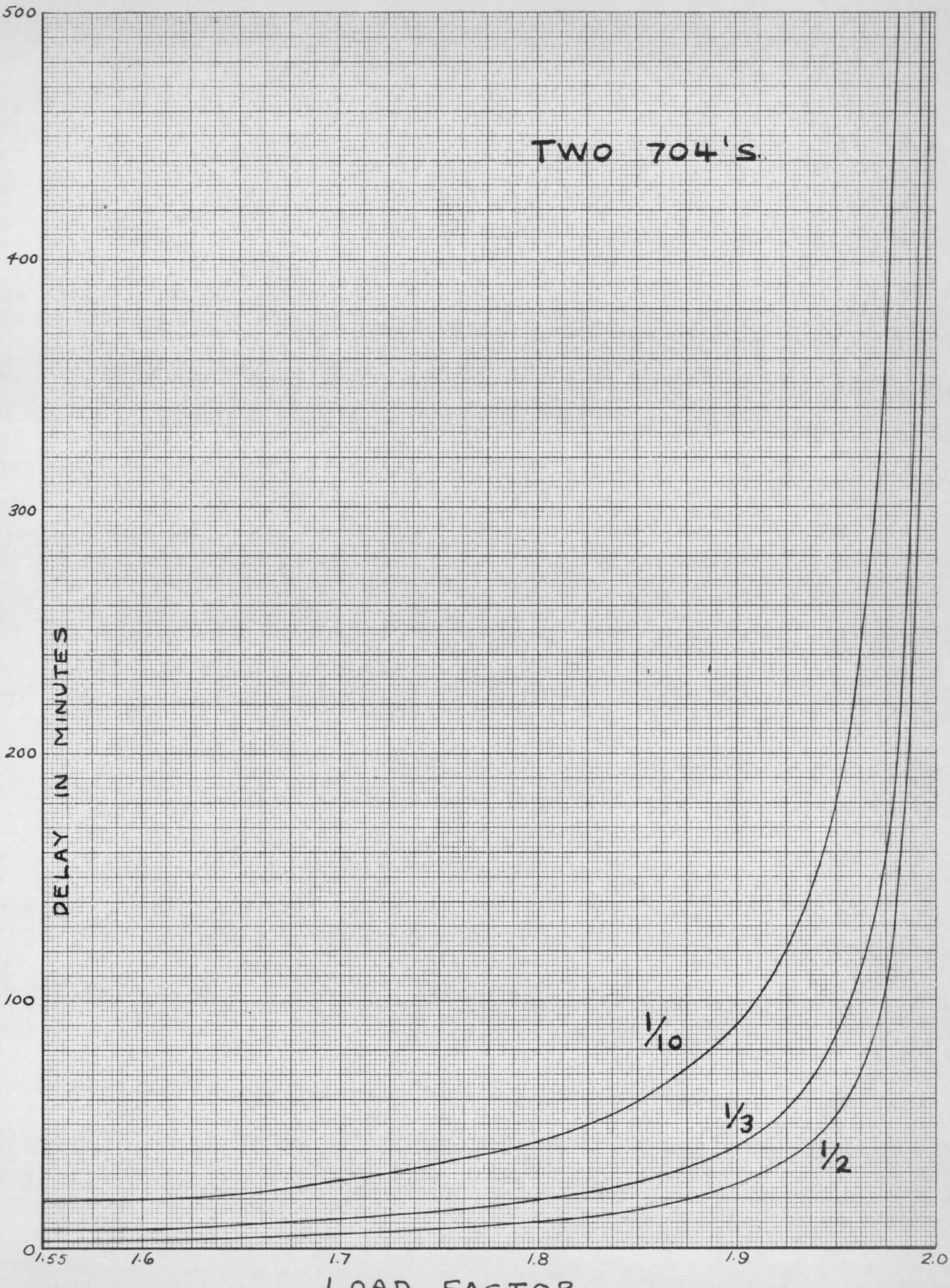
Reference:

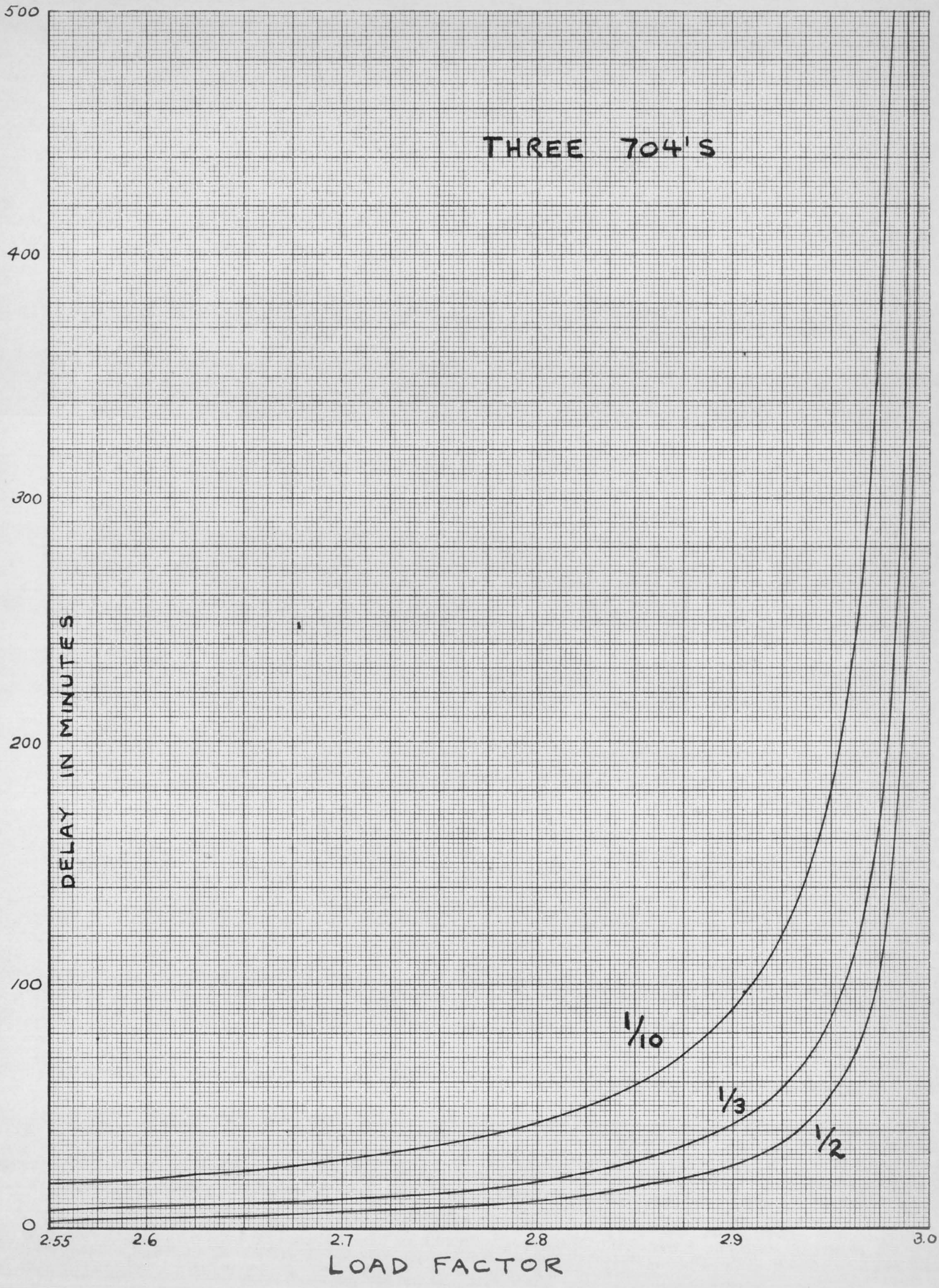
T. C. Fry - "Probability and Its Engineering Uses" -
Chapter X - D. Van Nostrand Co.

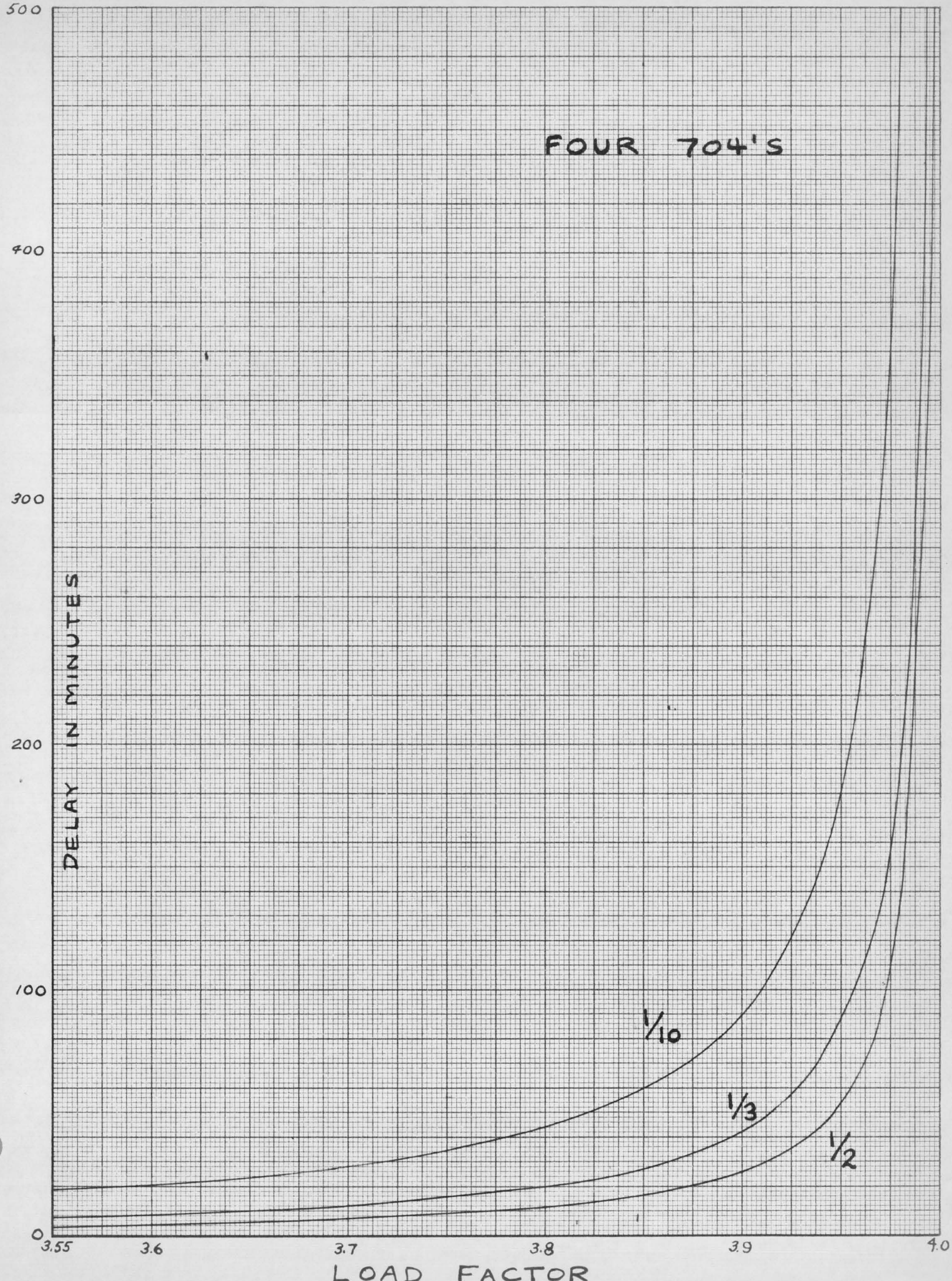
Assumptions:

1. Runs average 4 minutes each.
2. Distribution of length of runs is approximately exponential.
3. System is in statistical equilibrium; i.e., the probability of its being in any specified condition is independent of the time at which it is examined.
4. Machine change-over time between one run and the next is negligible.
5. The number of job sources is large compared to the number of machines.
6. Any job can be handled on any machine.
7. All machines are in use whenever there is enough work available.
8. Any run which cannot be handled immediately is held until a machine becomes available. It is then handled at once. If several runs are being held, they will be serviced in the order in which they originate.


ONE 704
DELAY IN MINUTES
LOAD FACTOR
500
400
300
200
100
0
0.55
0.6
0.7
0.8
0.9
1.0
1/10
1/3
1/2


TWO 704'S.
500
400
300
200
100
0
DELAY IN MINUTES
1/10
1/3
1/2
1.55
1.6
1.7
1.8
1.9
2.0
LOAD FACTOR


THREE 704'S
DELAY IN MINUTES
500
400
300
200
100
0
2.55
2.6
2.7
2.8
2.9
3.0
LOAD FACTOR
1/10
1/3
1/2


FOUR 704'S
DELAY IN MINUTES
LOAD FACTOR
500
400
300
200
100
0
3.55
3.6
3.7
3.8
3.9
4.0
1/10
1/3
1/2

Tabulation of Typical Results

Percentage of Idle Time				Delay in Minutes		
One 704	Two 704's	Three 704's	Four 704's	Average	One out of three	One out of ten
20.	10.	6.67	5.	11	19	43
10.	5.	3.33	2.5	25	41	89
5.	2.5	1.67	1.25	52	85	181
2.5	1.25	0.83	0.62	108	173	365
1.25	0.62	0.42	0.31	219	349	734

Note that these results may easily be modified for cases where the average length of run is greater than or less than 4 minutes since, to a close approximation, the delay is directly proportional to the average length of run.

OR2
6 January 1958

DECIMAL CARD READ ROUTINE
Compilable: Minimum 1103A
Operations Research Office-Sumner
Revision of CV131-Hauser and Gerkin

A. PURPOSE

This routine reads decimal numbers from cards and converts them into the appropriate binary numbers, sealed as desired by the programmer.

B. USAGE

1. LEADING LINES

Sub	OR2	142
Temps	0	0
Inout	1	0

2. INPUT

The input consists of two basic parts. The actual input to the routine is a control word having the following composition:

AB OPPPP ODDDD (octal)

A: First octal digit, controls picking of cards in either channel of the Bull reproducer.

A = 0 Do not pick
A = 1 Pick a read card
A = 2 Pick a punch card
A = 3 Pick both read and punch cards

B: Second octal digit, controls the reading operation.

B = 0 Do not read
B = 1 Read a card

OPPPP: Magnetic core address of the first parameter word.

ODDDD: Magnetic core address where the number from the first card field is to be stored.

This control word supplies the subroutine with the location of the parameter words which have the following composition:

FF SSBBL LRRZZ (octal)

FF: Flag for final parameter word

FF = 77 (octal) for final parameter word
FF = 00 otherwise

1043

SS: Binary scaling factor (number of bits to the right of the binary point) of converted number.

BB: Number of blank or unused card columns to the left of the field. (See definition of field under D.)

LL: Number of columns (digit positions) to the left of the decimal point.

RR: Number of remaining columns in the field, exclusive of sign (number of decimal digits to the right of the decimal point plus one for the decimal point).

ZZ: Not used.

Range of parameters:

DECIMAL	OCTAL
$00 \leq SS \leq 35$	$00 \leq SS \leq 43$
$00 \leq BB \leq 63$	$00 \leq BB \leq 77$
$00 \leq LL \leq 10$	$00 \leq LL \leq 12$
$00 \leq RR \leq 11$	$00 \leq RR \leq 13$
$01 \leq LL + RR \leq 11$	$01 \leq LL + RR \leq 13$

These parameter words should be stored as constants by the programmer in consecutive order. To read a card containing n fields (numbers) n consecutive parameters must be stored beginning at the address specified by the u-address of the control word. The nth parameter should contain 77 (octal) as the first two digits.

3. RECOVERY PROCEDURE

In case of a card machine failure or an accidental stop in the middle of a card cycle, the current card may be reread: reposition the cards, set (PAK) = 00000, and Start.

4. NUMBER OF TAGS USED = 23 (decimal)

5. RUNNING TIME

The programmer has 14 ms. computing time available between references to the card routine. If this time is regarded as a maximum, the Bull reader is virtually in a state of free run, and can attain a reading speed of approximately 120 cards per minute.

C. RESTRICTIONS

The only restrictions are those caused by the parameter ranges.

D. ADDITIONAL PROGRAMMING AND OPERATING INFORMATION

This card routine operates on a two-cycle basis, making more efficient use of the card cycle time than previous routines. As a result, as many as forty fields may be read from each card without causing any timing difficulties. Two basic operations are performed during the 18-point card cycle. The first five points (about 140 ms.) are used to decode the control words and to perform the final conversion of information read during the previous read cycle. The remainder of the card cycle is used to read information from the present card and convert this information into binary coded decimal form. The binary coded decimal information is then converted to binary and scaled during the first part of the next read cycle. Thus, although it takes two card cycles to complete the operation of reading and converting, the net effect is conversion of one card each card cycle.

A field consists of a number of consecutive card columns. The last column of a field is reserved for the sign of the decimal number stored in that field. An 11-punch signifies a negative number, no punch (blank column) signifies a positive number. A combination 12, 3, and 8 punch in one column may be used to represent a decimal point.

Fields need not be adjacent--there may be unused columns, punched or unpunched, between them--nor need they be alike in size.

The read side of the Bull must be cycled once before reading. The following coding is an example of a routine to read <u>n</u> cards, each of which contains <u>p</u> decimal numbers.

TAG	OPS	U	V	Comments
	OR2	Cycle		Pick a read card
	OR2	Read		Read (only) first card
Set	TP	K	IJC	Set counter for n -1
	OR2	Read - 1		Read and convert
	RA	Read - 1	Step	Advance data storage
	IJ	IJC	Set - 1	Check counter
	MS			
Cycle	10			Control word
Read	11			" "
	11	Param	Data	" "
K			n - 1	No. of cards, minus one
IJC	X			Counter for <u>n</u> reads
Step			p	Constant
Param	B	1st. parameter word		Parameter words
	.	.	.	
	.	.	.	
	.	.	.	
	B	pth. parameter word		
Data	Reserv	pn	pn	Region at which data is to be stored

Note that the above coding assumes the format on each of the <u>n</u> cards to be the same. In other words, any particular field must have the same size and location on all of the <u>n</u> cards, or else a new set of <u>p</u> parameter words must be used for each card.

The information read from the card is stored within the card routine in coded decimal form. Thus, if the subroutine is destroyed between card cycles, this information will not be converted on the following card cycle.

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,SUB	,OR2	,142	,	$
,	,TEMPS	,0	,0	,	$
,	,INOUT	,1	,0	,	$
,	,MJ	,	,P+1	,ENTRANCE	$
,	,ALARM	,	,	,ALARM EXIT	$
,EXIT	,MJ	,	,FILL	,NORMAL EXIT	$
,P	,X	,	,	,CONTROL WORD	$
,	,TP	,BODY-2	,Q	,TEST CONTROL	$
,	,QT	,P	,A	, WORD FOR	$
,	,ZJ	,BODY	,BODY-1	, SWITCH	$
,	,07	,	,	, SETTING	$
,	,RJ	,PICK+1	,PICK+1	,SET SWITCH	$
,BODY	,TP	,P	,PAR	,SET UP	$
,	,RJ	,GO	,GO	, ENTRANCE	$
,PAR	,X	,	,	,	$
,	,MJ	,	,EXIT	,	$
,GO	,MP	,K	,FILL	,	$
,	,TU	,A	,ROW+4	,STORE PARAMETER WORD	$
,	,TV	,A	,NEG+2	,STORE DATA WORD	$
,	,LQ	,A	,3	,	$
,	,SP	,MASK+4	,30	,	$
,	,QA	,MASK-1	,A	,EXTRACT PICK CODES	$
,	,SA	,A	,1	,	$
,	,LQ	,Q	,2	,	$
,	,QJ	,OUT+1	,PICK	,READ?	$
,PICK	,EF	,	,A	,START CARD CYCLE	$
,	,RJ	,PICK+1	,PICK+2	,CONVERSION SWITCH	$
,	,RJ	,PICK+2	,PICK+3	,READ SWITCH	$

1047

```
TAG      OPS      U ADDRESS   V ADDRESS   COMMENTS

,        ,SP      ,GO         ,           ,SET                    $
,        ,AT      ,K-1        ,OUT        ,  EXIT                 $
,OUT     ,X       ,           ,           ,ROW WORD 3             $
,        ,SA      ,K          ,           ,ADD READ CODE          $
,        ,RJ      ,PICK+2     ,PICK       ,SET TO READ            $
,        ,RS      ,A          ,A          ,                       $
,        ,RP1     ,10         ,OUT+6      ,CLEAR                  $
,        ,TP      ,A          ,MASK+5     ,  MATRIX               $
,        ,TV      ,K-2        ,           ,SET RERUN              $
,        ,TV      ,READ+6     ,DIGIT      ,                       $
,READ    ,ER0     ,           ,OUT        ,READ                   $
,        ,ER1     ,           ,Q          ,  ONE                  $
,        ,ER1     ,           ,R WORD     ,    ROW                $
,        ,RJ      ,READ+3     ,READ+4     ,LAST ROW SWITCH        $
,        ,TP      ,AWAY+2     ,ADD        ,                       $
,        ,TP      ,MASK-1     ,INDEX      ,                       $
,        ,SP      ,K-1        ,9          ,                       $
,        ,SA      ,A          ,3          ,                       $
,BIT     ,QJ      ,BIT+1      ,BIT+2      ,BIT IN THIS COLUMN?    $
,        ,SA      ,DIGIT      ,           ,                       $
,        ,SJ      ,BIT+3      ,READ+7     ,REACHED SENTINEL?      $
,        ,SP      ,A          ,           ,                       $
,ADD     ,X       ,           ,           ,                       $
,        ,RA      ,ADD        ,MASK+2     ,STEP MATRIX STORE      $
,        ,IJ      ,INDEX      ,READ+6     ,ROW WORD EXHAUSTED?    $
,        ,RJ      ,ADD+3      ,ADD+4      ,SWITCH                 $
,        ,LQ      ,RWORD      ,           ,                       $
,        ,RJ      ,ADD+3      ,READ+5     ,                       $
```

1048

```
TAG      OPS     U ADDRESS   V ADDRESS    COMMENTS

,        ,LQ     ,OUT        ,28          ,                          $
,        ,RJ     ,ADD+3      ,READ+6      ,                          $
,        ,RJ     ,ADD+8      ,ADD+9       ,SWITCH FOR SIGN ROW       $
,        ,IJ     ,DIGIT      ,READ        ,                          $
,        ,RJ     ,ADD+8      ,OUT+7       ,                          $
,ROW     ,RJ     ,READ+3     ,READ        ,READ LAST ROW             $
,        ,RJ     ,PICK+1     ,PICK+3      ,SET CONVERSION SWITCH     $
,        ,TU     ,AWAY+2     ,STEP+1      ,                          $
,        ,RJ     ,NEG+5      ,ROW+4       ,SET CONVERSION REPEAT     $
,        ,LQ     ,FILL       ,            ,                          $
,LAST    ,QJ     ,LAST+1     ,LAST+2      ,LAST FIELD?               $
,        ,TV     ,PICK+1     ,NEG+5       ,SET CONVERSION EXIT       $
,        ,LQ     ,Q          ,11          ,                          $
,        ,QT     ,MASK       ,A           ,SET                       $
,        ,TV     ,A          ,REM+3       ,  SHIFT                   $
,        ,LQ     ,Q          ,6           ,                          $
,        ,QT     ,MASK       ,INDEX       ,                          $
,        ,LQ     ,Q          ,6           ,                          $
,        ,QT     ,MASK       ,RWORD       ,                          $
,        ,LQ     ,Q          ,6           ,                          $
,        ,QT     ,MASK       ,OUT         ,                          $
,        ,SA     ,A          ,14          ,                          $
,SET     ,AT     ,AWAY+1     ,SET+1       ,SET NEXT INSTRUCTION      $
,        ,X      ,           ,            ,                          $
,        ,IJ     ,INDEX      ,NEG+6       ,                          $
,        ,RJ     ,AWAY       ,SET+4       ,                          $
,        ,IJ     ,RWORD      ,STEP-2      ,                          $
,        ,RJ     ,STEP+3     ,REM         ,SHIFT DECIMAL POINT       $
```

1049

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,RJ	,AWAY	,REM	,CONVERT	$
,REM	,IJ	,OUT	,STEP-2	, REMAINING TERMS	$
,	,TP	,SET+1	,A	,	$
,	,LT1	,71	,Q	,	$
,	,SP	,INDEX	,FILL	,	$
,	,SA	,Q	,	,ADD ROUNDING TERM	$
,	,DV	,SET+1	,SET+1	,	$
,	,RJ	,STEP+3	,STEP-2	,SHIFT SIGN	$
,	,QT	,MASK+4	,A	,	$
,NEG	,ZJ	,NEG+1	,NEG+2	,SIGN NEGATIVE?	$
,	,TN	,SET+1	,SET+1	,	$
,	,TP	,SET+1	,FILL	,STORE RESULT	$
,	,RA	,ROW+4	,MASK+1	,	$
,	,RA	,NEG+2	,K	,	$
,	,MJ	,	,FILL	,	$
,	,TV	,AWAY+3	,STEP+3	,	$
,	,IJ	,DIGIT	,STEP+2	,MATRIX WORD EXHAUSTED?	$
,	,TP	,MASK+4	,DIGIT	,	$
,STEP	,RA	,STEP+1	,MASK+1	,	$
,	,TP	,FILL	,MASK+5	,TRANSFER NEW MATRIX WORD	$
,	,LQ	,MASK+5	,4	,	$
,	,RJ	,STEP+3	,STEP+4	,SWITCH	$
,	,SP	,INDEX	,2	,	$
,	,SA	,INDEX	,1	,	$
,	,QA	,MASK+3	,INDEX	,	$
,AWAY	,MJ	,	,FILL	,CONVERSION EXIT	$
,	,TP	,K	,SET+1	,PRESET	$
,	,AT	,MASK+5	,MASK+5	,PRESET	$

1050

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,	,	,SET+2	,	$
,	,	,	,GO	,	$
,	,LA	,	,	,	$
,K	,X	,	,1	,	$
,	,X	,	,1	,TABLE	$
,N	,X	,	,10	,	$
,	,X	,	,100	, POWERS	$
,	,X	,	,1000	,	$
,	,X	,	,10000	,	$
,	,X	,	,100000	,	$
,	,X	,1	,000000	, OF	$
,	,X	,10	,000000	,	$
,	,X	,100	,000000	,	$
,	,X	,1000	,000000	,	$
,	,X	,10000	,000000	, TEN	$
,	,X	,	,3	,	$
,MASK	,	,	,77)B	,EXTRACTOR	$
,	,	,1	,0	,U ADVANCE	$
,	,	,1	,1	,U AND V ADVANCE	$
,	,	,	,17)B	,	$
,	,	,	,10)B	,	$
,	,RESERV	,9	,9	,MATRIX WORD	$
,INDEX	,X	,	,	,INDEX	$
,DIGIT	,X	,	,	,DIGIT	$
,RWORD	,X	,	,	,ROW WORD 2	$
,	,ENDSUB	,	,	,	$

1051

OR3
6 January 1958

DECIMAL CARD PUNCH ROUTINE
Compilable: Minimum 1103A
Operations Research Office-Sumner
Revision of CV130-Hauser and Gerkin

A. PURPOSE

This routine converts binary numbers into decimal, and punches them on cards, complete with sign and decimal point.

B. USAGE

1. LEADING LINES

Sub	OR3	173
Temps	0	0
Inout	1	0

2. INPUT

The input to this routine consists of a control word and a series of parameter words. The programmer must supply the routine with the control word, which in turn supplies the routine with the location of the parameter words and the location of the data which is to be punched. The control word has the following composition:

AB OPPPP ODDDD (octal)

A: First octal digit, controls picking of cards in either channel of the Bull Reproducer.

A = 0 Do not pick
A = 1 Pick a read card
A = 2 Pick a punch card

B: Second octal digit, controls the punching operation.

B = 0 Do not punch
B = 2 Punch a card

OPPPP: Magnetic core address of the first parameter word.

ODDDD: Magnetic core address of the first data word.

The parameter words (one for each data word) have the following compositions:

FF = 77 (octal) for final parameter word.
FF = 00 otherwise

SS: Binary scaling factor (number of bits to the right of the binary point).

BB: Number of blank or unused columns between previous field (see definition of field under D), or edge of card, and present field.

LL: Number of digit positions to the left of the decimal point.

RR: Number of remaining columns in the field exclusive of sign. (RR = 00 - no decimal point and no decimal fraction.)

ZZ: Flag for zero suppression.

ZZ = 77 (octal) for zero suppression
ZZ = 00 for no zero suppression. Only zeros in the integer part are suppressed. A zero immediately preceding the decimal point is not suppressed.

The total size of a field is: LL + RR + 1

Range of Parameters:

DECIMAL	OCTAL
$00 \leq SS \leq 35$	$00 \leq SS \leq 43$
$00 \leq BB \leq 63$	$00 \leq BB \leq 77$
$00 \leq LL \leq 10$	$00 \leq LL \leq 12$
$00 \leq RR \leq 11$	$00 \leq RR \leq 13$
$01 \leq LL - RR \leq 11$	$01 \leq LL - RR \leq 13$

The parameter words, one for each field, must be stored consecutively, starting at some magnetic core memory location OPPPP. There must be an equal number of consecutive words starting with some magnetic core memory location ODDDD, filled with data for the punch routine.

3. RECOVERY PROCEDURE

In the event of a card machine failure or an accidental stop in the middle of a card cycle, the current card may be punched again. Set (PAK) = 00000, and Start.

4. NUMBER OF TAGS USED = 18 (decimal)

5. RUNNING TIME

The programmer has 17 ms. computing time available between references to the card routine. If this time is regarded as a maximum, the Bull punch is virtually in a state of free run, and can attain a punching speed of approximately 120 cards per minute.

53

C. RESTRICTIONS

The only restrictions are those caused by the parameter ranges.

D. ADDITIONAL PROGRAMMING AND OPERATING INFORMATION

Numbers are rounded to the specified number of decimal digits after the decimal point before punching takes place. A divide fault occurs if an insufficient number of card columns is allowed for the integer portion of a field.

Punching takes place at the third card station in the punch channel. Therefore, two punch cards must be advanced before punching can take place. This may be done by referencing the card routine twice, supplying the appropriate control word. (See composition of control words.)

A field consists of a number of consecutive card columns. The last column of the field is reserved for the sign of the decimal number stored in that field. An 11-punch signifies a negative number, no punch (blank column) signifies a positive number. A combination 12, 3, and 8 punch in one column represents a decimal point. Fields need not be adjacent--there may be unused columns, punched or unpunched, between them--nor need they be alike in size.

It is possible to convert and punch as many as forty fields in a card and have 17 ms. computing time available between references to the punch routine.

```
TAG      OPS      U ADDRESS    V ADDRESS    COMMENTS
,        ,SUB     ,OR3         ,173         ,                          $
,        ,TEMPS   ,0           ,0           ,                          $
,        ,INOUT   ,1           ,0           ,                          $
,        ,MJ      ,            ,BODY        ,ENTRANCE                  $
,        ,ALARM   ,            ,            ,ALARM EXIT                $
,EXIT    ,MJ      ,            ,FILL        ,NORMAL EXIT               $
,P       ,X       ,            ,            ,CONTROL WORD              $
,BODY    ,TP      ,P           ,PAR         ,SET                       $
,        ,RJ      ,GO          ,GO          ,  UP                      $
,PAR     ,X       ,            ,            ,   ENTRANCE               $
,        ,MJ      ,            ,EXIT        ,                          $
,GO      ,MP      ,K+6         ,FILL        ,                          $
,        ,TU      ,A           ,LAST-1      ,SET PARAMETER PICKUP      $
,        ,LQ      ,A           ,3           ,                          $
,        ,SA      ,A           ,11          ,                          $
,        ,TU      ,A           ,STORE+7     ,SET DATA PICKUP           $
,        ,SP      ,K+3         ,32          ,                          $
,        ,QA      ,IMAGE-5     ,A           ,EXTRACT PICK CODES        $
,        ,SA      ,A           ,1           ,                          $
,        ,QJ      ,BULL-1      ,BULL-1      ,                          $
,        ,QJ      ,PUNCH-1     ,BULL        ,PUNCH?                    $
,BULL    ,EF      ,            ,A           ,START BULL                $
,        ,SP      ,GO          ,            ,SET                       $
,        ,AT      ,K+5         ,BULL+3      ,  EXIT                    $
,        ,MS      ,            ,BULL+3      ,EXIT                      $
,        ,SA      ,K+3         ,            ,ADD PUNCH CODE            $
,PUNCH   ,EF      ,            ,A           ,START BULL                $
,        ,CC      ,IMAGE       ,A           ,                          $
```

1055

```
TAG      OPS    U ADDRESS     V ADDRESS     COMMENTS

,        ,RPV   ,35           ,PUNCH+4      ,                           $
,        ,TP    ,A            ,IMAGE+1      ,                           $
,        ,TV    ,K            ,             ,SET EMERGENCY RERUN        $
,        ,TP    ,K+2          ,BULL+3       ,                           $
,        ,TP    ,PRESET       ,ALL+2        ,                           $
,        ,RJ    ,ALL+4        ,LAST-1       ,SET SWITCH                 $
,        ,LQ    ,FILL         ,             ,                           $
,LAST    ,QJ    ,NEG+4        ,LAST+1       ,LAST FIELD?                $
,        ,LQ    ,Q            ,11           ,                           $
,        ,QT    ,IMAGE-4      ,A            ,                           $
,        ,SN    ,A            ,             ,                           $
,        ,AT    ,PRESET+5     ,STORE+8      ,SET UP SHIFT ORDER         $
,        ,LQ    ,Q            ,6            ,                           $
,        ,QT    ,IMAGE-4      ,STEP+5       ,                           $
,        ,LQ    ,Q            ,6            ,                           $
,        ,QT    ,IMAGE-4      ,ALL+3        ,                           $
,        ,SA    ,A            ,14           ,                           $
,SET     ,AT    ,PRESET+6     ,SET+1        ,SET NEXT INSTRUCTION       $
,        ,X     ,             ,             ,                           $
,        ,LQ    ,Q            ,6            ,                           $
,        ,QT    ,IMAGE-4      ,SET+1        ,                           $
,        ,SA    ,A            ,14           ,                           $
,        ,AT    ,PRESET+7     ,STORE        ,SET NEXT INSTRUCTION       $
,STORE   ,X     ,             ,             ,                           $
,        ,QJ    ,NEG+1        ,STORE+2      ,ZERO SUPPRESSION?          $
,        ,TV    ,NOZERO       ,PRESET-1     ,                           $
,        ,RJ    ,NOZERO+8     ,STORE+4      ,                           $
,        ,IJ    ,STEP+5       ,NOZERO+3     ,                           $
```

1056

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,SP	,IMAGE-3	,19	,	$
,	,DV	,STORE	,Q	,	$
,	,TM	,FILL	,STEP+5	,	$
,	,X	,	,	,	$
,	,SA	,Q	,	,ADD ROUNDING TERM	$
,	,DV	,STEP+6	,A	,	$
,	,AT	,A	,STORE	,	$
,	,RJ	,NOZERO+8	,DEC-1	,L TIMES THROUGH	$
,	,IJ	,ALL+3	,NOZERO+9	, CONVERSION LOOP	$
,DEC	,TV	,K-1	,NOZERO+8	,STORE DECIMAL POINT	$
,	,IJ	,SET+1	,STEP+2	,	$
,	,IJ	,SET+1	,NOZERO+9	,	$
,	,TU	,STORE+7	,NEG-1	,	$
,	,RA	,LAST-1	,IMAGE-3	,STEP PARAMETER	$
,	,RA	,STORE+7	,IMAGE-3	,	$
,	,TV	,ALL+4	,NOZERO+8	,SET EXIT	$
,	,SN	,IMAGE-3	,	,	$
,	,LQ	,FILL	,	,	$
,NEG	,QJ	,NOZERO+1	,NOZERO+3	,NEGATIVE?	$
,	,RJ	,PRESET-1	,STORE+3	,	$
,	,EJ	,ALL+3	,NOZERO	,	$
,	,MJ	,	,NOZERO+3	,	$
,	,RJ	,ALL+4	,LAST+1	,	$
,	,RP3	,3	,ALL+1	,SET UP	$
,	,TP	,PRESET+2	,ALL+2	, PUNCH ORDERS	$
,ALL	,EJ	,PRESET+3	,BULL+1	,	$
,	,TV	,K	,	,SET EMERGENCY RERUN	$
,	,X	,	,	,PUNCH	$

1057

```
TAG       OPS     U ADDRESS    V ADDRESS    COMMENTS
,         ,X      ,            ,            ,    ONE                  $
,         ,X      ,            ,            ,        ROW              $
,STEP     ,RP2    ,3           ,ALL         ,STEP                     $
,         ,RS     ,ALL+2       ,K+6         ,    PUNCH ORDERS         $
,         ,SP     ,IMAGE-2     ,            ,                         $
,         ,AT     ,ALL+2       ,STEP+5      ,                         $
,         ,AT     ,K+4         ,STEP+6      ,                         $
,         ,X      ,            ,            ,                         $
,         ,X      ,            ,            ,                         $
,         ,SN     ,IMAGE-1     ,            ,                         $
,NOZERO   ,RJ     ,PRESET-1    ,NOZERO+1    ,SET NO ZERO SUPPRESSION  $
,         ,AT     ,ALL+2       ,NOZERO+2    ,                         $
,         ,X      ,            ,            ,                         $
,         ,LQ     ,BULL+3      ,35          ,                         $
,         ,QJ     ,NOZERO+5    ,NOZERO+8    ,ADVANCE TO NEW CARD FIELD? $
,         ,RA     ,ALL+2       ,K+1         ,                         $
,         ,TJ     ,PRESET+1    ,NOZERO+8    ,THIRD CARD FIELD?        $
,         ,LQ     ,BULL+3      ,8           ,                         $
,         ,MJ     ,            ,FILL        ,                         $
,         ,SP     ,STORE       ,2           ,                         $
,         ,SA     ,STORE       ,1           ,                         $
,         ,TP     ,A           ,STORE       ,STORE FRACTIONAL PART    $
,         ,SS     ,A           ,51          ,                         $
,         ,ZJ     ,NOZERO      ,FILL        ,                         $
,PRESET   ,CC     ,IMAGE+2     ,BULL+3      ,P                        $
,         ,CC     ,IMAGE+24    ,BULL+3      , R                       $
,         ,EWO    ,            ,IMAGE+35    ,  E                      $
,         ,EW1    ,            ,IMAGE+11        S                     $
```

1058

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,EW1	,	,IMAGE+23	, E	$
,	,SP	,STEP+5	,35	, T	$
,	,TP	,K+7	,STEP+6	, S	$
,	,TP	,K+6	,STORE	,	$
,	,	,	,DEC+2	,	$
,K	,	,	,GO	,	$
,	,	,12	,	,U ADVANCE	$
,	,B	,400000	,000000	,	$
,	,	,	,2	,	$
,	,	,5	,	,	$
,	,LA	,	,	,	$
,	,B	,	,1	,	$
,	,B	,	,1	,TABLE	$
,	,X	,	,10	,	$
,	,X	,	,100	,	$
,	,X	,	,1000	, POWERS	$
,	,X	,	,10000	,	$
,	,X	,	,100000	,	$
,	,X	,	,1000000	, OF	$
,	,X	,	,10000000	,	$
,	,X	,1	,00000000	,	$
,	,X	,10	,00000000	,	$
,	,X	,100	,00000000	, TEN	$
,	,X	,	,3	,	$
,	,B	,	,77	,EXTRACTOR	$
,	,	,1	,	,	$
,	,	,3	,	,	$
,	,	,2	,	,	$

1059

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,IMAGE	,RESERV	,36	,36	,IMAGE REGION	$
,	,ENDSUB	,	,	,	$

1060

OR4
6 January 1958

COLUMN HEADING ROUTINE
Compilable: Minimum 1103A
Operations Research Office-Sumner

A. PURPOSE

This routine allows the programmer to include alpha-numeric information, titles, and headings in any punched card output.

B. USAGE

1. LEADING LINES

Sub	OR4	75
Temps	0	0
Inout	1	0

2. INPUT

The input to this routine is a single parameter word having the form:

AO 00000 BBBBB (octal)

A = 4 Read and store one card image
A = 0 Punch one stored card image

BBBBB: The first storage address (drum or core) of a card image. If a card is to be read and stored, BBBBB is the first of 36 (decimal) storage locations which will be used by the image. If a stored image is to be punched, BBBBB is the first address at which the image is to be found.

3. RECOVERY PROCEDURE

There are only two ways in which a card fault can occur while this routine is operating:

(a) Erasure of the manual jump command at position 00000 before entering the column heading routine.

(b) Failure of the external equipment in picking a read or punch card. In either case, the routine may be restarted at the normal entrance after the cards have been repositioned.

4. NUMBER OF TAGS USED = 2 (decimal)

5. RUNNING TIME

The time required for one use of the routine is approximately the same as the time required for one complete read or punch cycle of the Bull reproducer. Therefore, it is possible to achieve virtual free run (120 cards per minute) if the programming between successive references to the routine is kept to a minimum.

1061

C. RESTRICTIONS

The storage positions at which a card image is to be stored should not coincide with any address within the column heading routine itself, since this would result in writing over part of the routine. Similarly, the programmer should not enter the read routine without specifying a storage address, since the routine will then store the image beginning at address 00000, erasing the manual jump at this position.

D. ADDITIONAL PROGRAMMING AND OPERATING INFORMATION

Since this routine does not attempt any conversion, but merely holds a card image, it is possible to punch anything whatsoever on the input cards. The output cards are limited by the Bull reproducer's inability to perform dependably if more than 160 punches per card, or 60 punches in any one row, are required. Since there are no real limitations on the input-output format, it is possible to include any number of special codes to control the tabulating equipment when the output cards are printed.

```
TAG       OPS       U ADDRESS     V ADDRESS     COMMENTS

,         ,SUB      ,OR4          ,76           ,                       $
,         ,TEMPS    ,0            ,0            ,                       $
,         ,INOUT    ,1            ,0            ,                       $
,         ,MJ       ,             ,BODY         ,NORMAL ENTRANCE        $
,         ,ALARM    ,             ,             ,                       $
,         ,MJ       ,             ,FILL         ,NORMAL EXIT            $
,         ,X        ,             ,             ,                       $
,BODY     ,RPO      ,1            ,BODY+2       ,                       $
,         ,TP       ,BODY-1       ,A            ,                       $
,         ,SJ       ,BODY+17      ,BODY+3       ,                       $
,         ,LA       ,A            ,15           ,PUNCH ENTRANCE         $
,         ,TU       ,A            ,BODY+6       ,                       $
,         ,RP3      ,36           ,BODY+7       ,OBTAIN IMAGE           $
,         ,TP       ,FILL         ,IJC+5        ,                       $
,         ,TP       ,IJC-1        ,IJC          ,SET IJC                $
,         ,EF       ,             ,IJC+4        ,PICK PUNCH             $
,         ,EWO      ,             ,IJC+5        ,PUNCH                  $
,         ,EW1      ,             ,IJC+6        ,ONE                    $
,         ,EW1      ,             ,IJC+7        ,ROW                    $
,         ,RP2      ,3            ,BODY+14      ,ADVANCE                $
,         ,RA       ,BODY+9       ,IJC+1        ,   IMAGE               $
,         ,IJ       ,IJC          ,BODY+9       ,CARD FINISHED?         $
,         ,RPU      ,3            ,BODY-2       ,RESET, EXIT            $
,         ,RS       ,BODY+9       ,IJC+2        ,                       $
,         ,TV       ,A            ,BODY+29      ,                       $
,         ,TP       ,IJC-1        ,IJC          ,                       $
,         ,EF       ,             IJC+3         ,PICK READ              $
,         ,ERO      ,             ,IJC+5        ,READ                   $
```

1063

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,ER1	,	,IJC+6	, ONE	$
,	,ER1	,	,IJC+7	, ROW	$
,	,RP2	,3	,BODY+25	,ADVANCE	$
,	,RA	,BODY+20	,IJC+1	, IMAGE	$
,	,IJ	,IJC	,BODY+20	,CARD FINISHED?	$
,	,RP2	,3	,BODY+28	,RESET	$
,	,RS	,BODY+20	,IJC+2	,	$
,	,RP3	,36	,BODY-2	,STORE IMAGE, EXIT	$
,	,TP	,IJC+5	,FILL	,	$
,	,X	,	,11	,N-1 ROWS	$
,IJC	,X	,	,	,ROW COUNTER	$
,	,X	,	,3	,ROW ADVANCE	$
,	,X	,	,36	,RESET CONSTANT	$
,	,B	,400000	,000005	,PICK READ CODE	$
,	,B	,400000	,000012	,PICK PUNCH CODE	$
,	,RESERV	,36	,36	,IMAGE REGION	$
,	,ENDSUB	,	,	,	$

OR5
6 January 1958

OCTAL TYPEWRITER OUTPUT ROUTINE
Compilable: Minimum 1103A
Operations Research Office-Summer

A. PURPOSE

This routine types on the console flexowriter the octal contents of any storage address, with the following options:

(a) Twelve digit octal type-out, no spacing, no zero suppression.

(b) Octal type-out with zero suppression, no spacing.

(c) Twelve digit octal type-out, with spacing between the ops., u, and v portions of the word, no zero suppression.

B. USAGE

1. LEADING LINES

Sub	OR5	55
Temps	0	0
Inout	1	0

2. INPUT

The input to this routine consists of a single parameter word which has the following composition:

AO BBBBB CCCCC (octal)

A = 0 no zero suppression
0 = 4 zero suppression (assumes that BBBBB = 0)

BBBBB: Any bits at all within the u-portion of the parameter will cause the routine to space between the ops., u, and v portion of the word. BBBBB = 0 assumes no spacing desired.

CCCCC: The storage address at which the desired octal number is to be found.

3. RECOVERY PROCEDURE

1065 No recovery procedure is necessary, since the only possible fault which could occur would be one due to computer malfunction or typewriter malfunction, either of which would be just cause for running maintenance tests.

4. NUMBER OF TAGS USED = 3 (decimal)

5. RUNNING TIME

The speed of the routine is limited only by the operating speed of the console flexowriter.

C. <u>RESTRICTIONS</u>

It is not possible to type out the contents of the Accumulator, but it is possible to type out the contents of the Q-register if so desired.

D. <u>ADDITIONAL PROGRAMMING AND OPERATING INFORMATION</u>

This routine is not intended to be used as a flexowriter dump of consecutive storage addresses, and therefore, must be used one time for each machine word to be typed out. For this same reason, the address of the output word is not included in the type-out.

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,SUB	,OR5	,55	,	$
,	,TEMPS	,	,	,	$
,	,INOUT	,1	,0	,	$
,	,MJ	,	,P+1	,ENTRANCE	$
,	,ALARM	,	,	,ALARM	$
,	,MJ	,	,FILL	,EXIT	$
,P	,X	,	,	,PARAMETER	$
,	,TP	,P	,A	,TEST FOR	$
,	,SJ	,IJC+1	,P+3	,ZERO SUPPRESSION	$
,	,TU	,A	,PRINT-2	,NO SUPPRESSION	$
,	,RS	,A	,A	,TEST FOR	$
,	,EJ	,PRINT-2	,IJC+2	,SPACING	$
,	,SP	,P	,15	,SHIFT PARAMETER	$
,	,TU	,A	,P+9	,OBTAIN ADDRESS	$
,	,TP	,IJC-2	,IJC	,SET IJC	$
,	,TP	,FILL	,PRINT-3	,OBTAIN NUMBER	$
,	,SP	,PRINT-3	,	,NO. TO A	$
,	,LT	,39	,PRINT-3	,ISOLATE ONE DIGIT	$
,	,TP	,A	,A	,CLEAR A LEFT	$
,	,MJ	,	,P+14	,SWITCH	$
,	,AT	,PRINT	,P+15	,LOAD PRINT ORDER	$
,	,X	,	,	,PRINT	$
,	,IJ	,IJC	,P+10	,CHECK IJC	$
,	,PR	,	,IJC-1	,SPACE	$
,	,TP	,IJC-1	,IJC	,SET IJC	$
,	,RJ	,IJC-9	,P+10	,SWITCH	$
,	,TP	,IJC-1	,IJC	,SET IJC	$
,	,RJ	,IJC-9	,P+10	,SWITCH	$

1067

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,TV	,PRINT-1	,IJC-9	,RESET	$
,	,TV	,PRINT-1	,IJC-7	,RESET	$
,	,MJ	,	,P-1	,GO TO EXIT	$
,	,B	,	,13	,NO. OF DIGITS	$
,	,B	,	,1	,	$
,	,B	,	,4	,	$
,IJC	,X	,	,	,INDEX JUMP COUNTER	$
,	,TP	,PRINT-5	,P+13	,SET SWITCH	$
,	,TP	,IJC-3	,IJC	,SET IJC	$
,	,SP	,P	,15	,SHIFT PARAMETER	$
,	,TU	,A	,P+9	,OBTAIN ADDRESS	$
,	,RJ	,IJC-9	,P+9	,RE-ENTER	$
,	,TP	,PRINT-4	,P+13	,RESET SWITCH	$
,	,MJ	,	,IJC-6	,RETURN	$
,	,TP	,PRINT-4	,P+13	,RESET SWITCH	$
,	,MJ	,	,P+14	,RETURN	$
,	,ZJ	,PRINT-7	,P+16	,SWITCH MODIFIER	$
,	,MJ	,	,P+14	,SWITCH RESET	$
,	,X	,	,	,	$
,	,X	,	,	,	$
,	,	,	,P+10	,RESET	$
,PRINT	,PR	,	,PRINT+1	,PRINT ORDER	$
,	,B	,	,37	,F	$
,	,B	,	,52	,L	$
,	,B	,	,74	,E	$
,	,B	,	,70	,X C	$
,	,B	,	,64	, O	$
,	,B	,	,62	, D	$

1068

TAG	OPS	U ADDRESS	V ADDRESS	COMMENTS	
,	,B	,	,66	, E	$
,	,B	,	,72	, S	$
,	,ENDSUB	,	,	,	$

1069

Talmadge

1. IDENTIFICATION ML T003, OCTAL TAPE WRITE

ROGER SKINNER - 5 MARCH 1957

LOCKHEED MISSILES DIVISION

2. PURPOSE

To write on a designated tape in octal format, n words from consecutive core or drum locations.

3. METHOD

This program writes the necessary number of blocks as determined by n in XS-3 code with 3 blanks following each word, or 48 words per block. No instructions are executed between external write commands during the writing of each block.

4. USAGE

a. Calling Sequence

LOC.	OP.	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal Return		

b. Control and Results

A parameter word with the following form must be placed at t+3.

aa bbbbb ccccc

a) Uniservo to be selected.

b) Number of words to be written.

c) Initial core or drum location.

c. Space Required

74 cells of instructions.

128 cells of erasable labeled COMMON through COMMON+127.

d. Error Codes

The parameter word is left in the Q-Register on return through the error exit. The following error code is left in the Accumulator on return through the error exit.

CODE	EXPLANATION
$T003 \cdot 2^{12}+1$	Core overflow in attempting to fetch and write n words.

e. Tape Format

The tape is written at 128 lines density and 1.0 inch block and blockette spacing, 8 words per blockette.

5. RESTRICTIONS

If the initial core address and n are such that this routine attempts to fetch a word at a location one greater than the last legal core address, the routine will write the number of words up to and including the last core address and then exit through the error exit.

6. CODING INFORMATION

a. Constants

LOCATION	CONSTANT	EXPLANATION
VMASK	000000077777	V address mask.
TMASK	000000170000	Tape indicator mask.
BLANK	010101010101	XS-3 code for blank word.
XS3TS	377737773777	XS-3 conversion test.
ODDEV	525252525252	Odd-even indicator.
LBKS	010101000000	Blanks in left half of word.
RBKS	000000010101	Blanks in right half of word.
CODE1	660303060001	Error code for core overflow.
U1	000000100000	1 in U-address.
N1	000000000001	One.
N3	000000000003	Three.
N5	000000000005	Five.
N7	000000000007	Seven.
N47	000000000057	Forty-seven.
N48	000000000060	Forty-eight.

b. Working Storage

128 cells labeled COMMON through COMMON+127.

c. Timing

Unknown.

```
  LOC      OP     U-ADDR         V-ADDR        EXPLANATION

 ENTRY     MJ                    START
 ERROR     RJ     DIAG+    2     DIAG
  EXIT     MJ                    FILL
     Y   FILL     FILL           FILL
 START     SP     Y              15
           TU     A              WORD          SET INITIAL ADDR
           LQ     Y              31025B
           QT     VMASK          COMMON        NO. WORDS
           LA     Y              32066B        POSITION T
           TP     TMASK          Q
           QS     A              WRITE         SET WRITE
 CYCLE     RS     COMMON         N48
           TP     N47            COMMON+  1    SET WORD INDEX
           SJ     NTEST          STBK-    1    TEST FOR FULL BLK
 NTEST     SA     N48            0
           SJ     EXIT           NZERO         TEST FOR END
 NZERO     ZJ     NPART          EXIT
 NPART     ST     N1             COMMON+  1    SET WORD INDEX
           RPV    120            WORD-    1
  STBK     TP     BLANK          COMMON+  8    SET IMAGE TO BKS
           TV     STBK           STORE         1ST BUFFER ADDR
  WORD     TP     FILL           COMMON+  2    CURRENT WORD
           TN     N3             A             -3 TO A
  CONV     SA     N3             6             XS-3 POSITIONED
           LQ     COMMON+  2     3             POSITION AND
           QA     N7             A              ADD DIGIT
           LQ     XS3TS          1             LOOP
           QJ     CONV           PTST           TEST
  PTST     AT     N3             COMMON+  7    STORE RIGHT
           LT     0              COMMON+  6    STORE LEFT
           LQ     ODDEV          1             ODD-EVEN IND
           QJ     EVEN           ODD           EVEN OR ODD
   ODD     RPB    2              NEWV          FIRST AND
           TP     COMMON+  6     COMMON+  3     SECOND
  EVEN     LA     COMMON+  6     18            SPLIT WORD
           LT     0              A
           AT     LBKS           COMMON+  5    BBBXXX THIRD
           LA     COMMON+  7     18            SPLIT WORD
           LT     0              A
           AT     COMMON+  6     COMMON+  6    FOURTH
           RA     COMMON+  7     RBKS          XXXBBB FIFTH
           RPB    5              BUMP          STORE
 STORE     TP     COMMON+  3     FILL           IN BUFFER
  BUMP     RA     STORE          N5            BUMP STORE
  NEWV     RA     WORD           U1            BUMP WORD FETCH
           EJ     OVER           BEXIT         TEST FOR OVFL
           IJ     COMMON+  1     WORD          NO. WORDS TEST
 LASTW     LQ     ODDEV          31001B
           QJ     ODDWD          WRBK           ODD NO. OF WORDS
 ODDWD     LQ     ODDEV          1             RESET IND
           TV     STORE          LSST
           RPB    2              WRBK          STORE
```

LOC	OP	U-ADDR	V-ADDR	EXPLANATION
LSST	TP	COMMON+ 3	FILL	LAST WORD
WRBK	EF		WRITE	START TAPE
	RPV	120	CYCLE	COPY AND
	EWB		COMMON+ 8	RETURN
BEXIT	TP	ODDEV	COMMON	PREPARE EXIT
	MJ		LASTW	RETURN TO WRITE
OVER	TP	10000B	COMMON+ 2	OVERFLOW TEST
WRITE	B		020064600001	WRITE DUMMY
VMASK	B		77777	VMASK
TMASK	B		170000	TAPE IND MASK
BLANK	B		10101010101	ALL BLANKS
XS3TS	B		377737773777	XS-3 CONV. TEST
ODDEV	B		525252525252	ODD-EVEN TEST
LBKS	B		10101000000	LEFT 3 BLANKS
RBKS	B		10101	RT 3 BLANKS
U1	B		100000	1 IN U ADDR
N1	B		1	ONE
N3	B		3	THREE
N5	B		5	FIVE
N7	B		7	SEVEN
N47	B		57	FORTY-SEVEN
N48	B		60	FORTY-EIGHT
	END			

Talmadge

1. IDENTIFICATION ML T001, OCTAL TAPE READ

Roger Skinner - Revised - 6 March 1957
Lockheed Missile Systems Division

2. PURPOSE

To read and store into consecutive core or drum locations, n words from a designated octal tape.

3. METHOD

a. This program reads the necessary number of blocks as determined by n from the designated uniservo. The blanks between words are removed during reading of each block. The conversion out of XS-3 and storing of the converted information is accomplished between blocks.

b. Four attempts will be made to read each block successfully in event of parity check. Two attempts will be made at normal bias, one at high bias, and one at low bias.

4. USAGE

a. Calling Sequence

LOC.	OP.	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal Return		

b. Control and Results

A parameter word with the following form must be placed at t+3.

aa bbbbb ccccc

a) Uniservo to be selected.

b) Number of octal words to be read and stored.

c) Initial storage location

c. Space Required

86 cells of instructions.

101 cells of erasable labeled COMMON through COMMON+100.

d. Error Codes

The parameter word is left in the Q-Register on return through the error exit. The diagnostic routine is entered immediately, and no attempt is made to space the tape forward to correspond to the number of blocks indicated by n. The following error codes are left in the Accumulator on return through the error exit.

CODE	EXPLANATION
$T001 \cdot 2^{12}+1$	Four unsuccessful attempts to read block.
$T001 \cdot 2^{12}+2$	Core overflow while storing information.

e. Tape Format

The octal tape read by this routine consists of blocks containing 48 octal words in XS-3 code, each followed by 3 blanks.

5. RESTRICTIONS

a. Information stored in such a way to overflow the last core address will cause immediate exit to the diagnostic routine.

b. If other than 3 blanks separate the octal words on tape, then the leading binary bit of the first of these 3 characters must not be a binary one.

6. CODING INFORMATION

a. Constants

LOCATION	CONSTANT	EXPLANATION
VMASK	000000077777	V address mask.
TMASK	000000170000	Tape indicator mask.
BIAS	020000150000	Normal bias.
THREES	030303030303	Used in XS-3 Conversion.
CODE1	660303040001	T001 x $2^{12}+1$
CODE2	660303040002	T001 x $2^{12}+2$
U2	000000200000	2 in U address
N1	000000000001	One
N3	000000000003	Three
N4	000000000004	Four
N5	000000000005	Five
N7	000000000007	Seven
N23	000000000027	Twenty-three
N48	000000000060	Forty-eight

b. Working Storage

101 cells labeled COMMON through COMMON+100.

c. Timing

Unknown

LOC	OP	U-ADDR	V-ADDR	EXPLANATION
ENTRY	MJ	0	START	
ERROR	RJ	DIAG+ 2	DIAG	
EXIT	MJ	0	FILL	
Y	FILL	FILL	FILL	
START	TV	Y	FSTOR	SET INITIAL ADD
	LQ	Y	31025B	
	QT	VMASK	COMMON	NO WORDS
	LA	Y	32066B	POSITION T
	TP	TMASK	Q	
	QS	A	READ	SET READ
	QS	A	BKSP	SET BK SPACE
CYCLE	RS	COMMON	N48	
	TP	N48	COMMON+ 1	SET INDEX
	SJ	TEST	SBIAS	TST FOR FULL BK
TEST	SA	N48	0	
	SJ	EXIT	ZERO	
ZERO	ZJ	PART	EXIT	
PART	TP	A	COMMON+ 1	SET INDEX
SBIAS	EF	0	BIAS	SET NORMAL BIAS
	TP	N3	COMMON+ 2	SET BIAS INDEX
RDBK	EF		READ	READ ONE BLOCK
	TV	PART	STORE	SET FIRST STORE
	TP	N23	COMMON+ 3	BLK INDEX TO 23
	IJ	COMMON+ 1	COPY	
	RPB	4	COPY	
STORE	TP	COMMON+ 97	FILL	STORE FOUR WORDS
COPY	RPV	4	POSIT	
	ERB	0	COMMON+ 97	READ FOUR WORDS
POSIT	LA	COMMON+ 99	18	POSITION THIRD
	LA	COMMON+100	18	SPLIT FOURTH
	LT	0	A	
	AT	COMMON+ 99	COMMON+ 99	THIRD STANDARD
	ERB	0	A	FIFTH FROM TAPE
	LT	18	A	POSITION
	AT	COMMON+100	COMMON+100	FOURTH STANDARD
	RA	STORE	N4	BUMP STORE
	IJ	COMMON+ 3	STORE- 1	
	ERA		A	IOA TO A
	ZJ	FAULT	OK	ERROR CHECK
FAULT	IJ	COMMON+ 2	CHNGE	TEST BIAS
	SP	CODE1	0	TAPE ERROR CODE
RETRN	TP	Y	Q	PARAMETER WORD
	MJ	0	ERROR	GO TO ALARM EXIT
CHNGE	SP	A	12	
	AT	BIAS	COMMON+ 4	
	EF	0	COMMON+ 4	CHANGE BIAS
	EF	0	BKSP	BACKSPACE
	RA	COMMON+ 1	N1	
	MJ	0	RDBK	REREAD
OK	TU	SHIFT	OK+ 2	SET FIRST WORD
	RPB	2	SUBST- 2	
FETCH	TP	FILL	COMMON+ 5	CURRENT WORD

LOC	OP	U-ADDR		V-ADDR		EXPLANATION
	TP	N1		COMMON+	3	
	RPU	2		SUBST+	2	
SUBST	RS	COMMON+	5	THREES		SUBSTRACT THREES
SEC	TP	COMMON+	6	COMMON+	5	SECOND HALF
	TP	N5		COMMON+	4	INDEX 1 TO 5
SHIFT	LQ	COMMON+	5	6		
	SP	COMMON+	2	3		POSITION PREVIOUS
	QA	N7		COMMON+	2	DIGIT
	IJ	COMMON+	4	SHIFT		TEST END HALF
	IJ	COMMON+	3	SEC		TEST END WORD
FSTOR	TP	COMMON+	2	FILL		STORE FINAL WORD
	RA	FSTOR		N1		BUMP STORE
	EJ	OVERD		OVER		OVERFLOW TEST
	RA	FETCH		U2		BUMP WORD FETCH
	IJ	COMMON+	1	FETCH-	1	TEST END BLK
	MJ	0		CYCLE		RETURN TO NEXT BLK
OVER	SP	CODE2		0		OVERFLOW CODE
	MJ	0		RETRN		
OVERD	TP	COMMON+	2	10000B		OVERFLOW DUMMY
READ	02	602B		0		DUMMY READ
BKSP	02	614B		1		DUMMY BACKSPACE
BIAS	02	1B		50000B		NORMAL BIAS
VMASK	B			77777		V ADDR MASK
TMASK	B			170000		TAPE IND MASK
N1	B			1		
N3	B			3		
N4	B			4		
N5	B			5		
N7	B			7		
N23	B			27		
N48	B			60		
U2	B			200000		2 IN U ADDR
THREES	B			30303030303		XS-3 DUMMY
CODE1	B			660303040001		TAPE ERROR CODE
CODE2	B			660303040002		OVERFLOW CODE
	END					

1. IDENTIFICATION — ML T002, FLOATING DECIMAL TAPE READ
Roger Skinner - Revised 6 March 1957
Lockheed Missile Systems Division

2. PURPOSE

To read and store into consecutive core or drum locations in floating binary form, n words from a designated floating decimal tape.

3. METHOD

a. This program reads the necessary number of blocks as determined by n from the designated uniservo. The blanks between words are removed during reading of each block. The conversion and storing of information is accomplished between blocks. An exponential method is used in the conversion from floating decimal to floating binary.

b. The accuracy of the conversion is to within a binary one in the 27th bit of the fraction in all cases. The conversion of all integers less than 10^7 is exact.

c. Four attempts will be made to read each block successfully in event of parity check. Two attempts will be made at normal bias, one at high bias, and one at low bias.

4. USAGE

a. Calling Sequence

LOC.	OP.	u ADDR.	v ADDR.
r	RJ	t+2	t
r+1	Normal Return		

b. Control and Results

A parameter word with the following form must be placed at t+3.

aa bbbbb ccccc

a) Uniservo to be selected.

b) Number of floating decimal words to be read and stored.

c) Initial storage location.

c. Space Required

176 cells of instructions

103 cells of erasable.

d. Error codes

The parameter word is left in the Q-Register on return through the error exit. In the case of tape failure or core overflow, the diagnostic routine is entered immediately, and no attempt is made to space the tape forward to correspond to the number of blocks indicated by n. In the case of power overflow, the largest possible number is substituted for the offending number and the diagnostic routine is not entered until completion of the n words called for by the parameter word. In case of power underflow, the offending number is set to zero, and there is no error indication. The following error codes are left in the Accumulator on return through the error exit.

CODE	EXPLANATION
$T002 \cdot 2^{12}+1$	Four unsuccessful attempts to read one block.
$T002 \cdot 2^{12}+2$	Core overflow while storing information.
$T002 \cdot 2^{12}+3$	Power overflow during floating conversion.

e. Tape Format

The floating decimal tape read by this routine consists of blocks containing 48 floating decimal numbers in XS-3 code, each followed by 3 blanks. A number is represented by 12 characters in the following order: Sign of number; 8 digit fraction; sign of power; 2 digit power.

5. RESTRICTIONS

a. Information stored in such a way to overflow the last core address will cause immediate exit to the diagnostic routine.

b. If other than 3 blanks separate the octal words on tape, then the leading binary bit of the first of these 3 characters must not be a binary one.

6. CODING INFORMATION

a. Constants

LOCATION	CONSTANT	EXPLANATION
OVDUM	353100010000	Overflow dummy word
WMASK	037777777777	Conversion mask
TMASK	000000170000	Tape indicator mask.
MASK1	000000000077	XS-3 conversion mask
VMASK	000000077777	Normal Bias
NEMAX	400000000000	Conversion constant
EXP1	035440262675	Conversion coefficient
EXP2	600000171150	Conversion coefficient
EXP3	270524354513	Conversion coefficient
LGB2T	324464741134	Log to base 2 of 10, *33
U2	000000200000	2 in U address
ETAB	200000000000	2 to 0, *34
ETAB+1	213453407440	2 to .125, *34
ETAB+2	230157701214	2 to .25, *34
ETAB+3	245775532516	2 to .375, *34
ETAB+4	265011714640	2 to .5, *34
ETAB+5	305316250212	2 to .625, *34
ETAB+6	327211763126	2 to .75, *34
ETAB+7	352601433477	2 to .875, *34
CODE1	660303050001	Tape error code
BIAS	020000150000	Normal Bias

6. CODING INFORMATION (Continued)

a. Constants (Continued)

LOCATION	CONSTANT	EXPLANATION
CODE2	660303050002	Core overflow code
CODE3	660303050003	Power overflow code.
NZ	000000000000	Zero
N1	000000000001	One.
N2	000000000002	Two
N3	000000000003	Three
N4	000000000004	Four
N5	000000000005	Five
N8	000000000010	Eight
N23	000000000027	Twenty-three.
N48	000000000060	Forty-eight
N129	000000000201	One hundred twenty-nine
N256	000000000400	One hundred fifty-six.

b. Working Storage

103 cells labeled COMMON through COMMON+102

c. Timing

Unknown

LOC	OP	U-ADDR	V-ADDR	EXPLANATION
ENTRY	MJ	0	START	
ERROR	RJ	DIAG+ 2	DIAG	
EXIT	MJ	0	FILL	
Y	FILL	FILL	FILL	
START	TV	Y	PSTOR	SET INITIAL ADD
	TP	PZ	OVIND	
	LQ	Y	31025B	
	QT	VMASK	COMMON	NO WORDS
	LA	Y	32066B	POSITION T
	TP	TMASK	Q	
	QS	A	READ	SET READ
	QS	A	BKSP	SET BK SPACE
CYCLE	RS	COMMON	N48	
	TP	N48	COMMON+ 1	SET INDEX
	SJ	TEST	SBIAS	TST FOR FULL BK
TEST	SA	N48	0	
	SJ	BEXIT	ZERO	
ZERO	ZJ	PART	BEXIT	
PART	TP	A	COMMON+ 1	SET INDEX
SBIAS	EF	0	BIAS	SET BIAS NORMAL
	TP	N3	COMMON+ 2	SET BIAS INDEX
RDBK	EF	0	READ	READ ONE BLOCK
	TV	RDBK+ 2	STORE	SET FIRST STORE
	TP	N23	COMMON+ 3	BLK INDEX TO 23
	IJ	COMMON+ 1	COPY	
	RPB	4	COPY	
STORE	TP	COMMON+ 99	FILL	STORE FOUR WORDS
COPY	RPV	4	POSIT	
	ERB	0	COMMON+ 99	READ FOUR WORDS
POSIT	LA	COMMON+101	18	POSITION THIRD
	LA	COMMON+102	18	SPLIT FOURTH
	LT	0	A	
	AT	COMMON+101	COMMON+101	THIRD STANDARD
	ERB	0	A	FIFTH FROM TAPE
	LT	18	A	POSITON
	AT	COMMON+102	COMMON+102	FOURTH STANDARD
	RA	STORE	N4	BUMP STORE
	IJ	COMMON+ 3	STORE- 1	
	ERA	0	A	IOA TO A
	ZJ	FAULT	OK	ERROR CHECK
FAULT	IJ	COMMON+ 2	CHNGE	TEST BIAS
	SP	CODE1	0	TAPE ERROR CODE
RETRN	TP	Y	Q	PARAMETER WORD
	MJ	0	ERROR	GO TO ALARM EXIT
CHNGE	SP	A	12	
	AT	BIAS	COMMON+ 4	
	EF	0	COMMON+ 4	CHANGE BIAS
	EF	0	BKSP	BACKSPACE
	RA	COMMON+ 1	N1	
	MJ	0	RDBK	REREAD
OK	TU	SPF	WORD	SET FIRST WORD
	RPB	2	SETST	CURRENT

```
LOC     OP    U-ADDR       V-ADDR       EXPLANATION

WORD    TP    FILL         COMMON+ 7     WORD
SETST   TV    MUPL+   2    STOR         FIRST WORD
SPF     SP    COMMON+ 7    6
        TP    A            Q            M TO Q
        LT    1            COMMON+ 2    SIGN M DOUBLED
        TP    N4           COMMON+ 3    INDEX TO 4
        RJ    RET          CLEAR        PART OF M
        TM    COMMON+ 4    COMMON+ 4
        TP    N2           COMMON+ 3    INDEX TO 2
        TP    COMMON+ 8    Q
        RJ    RET          CLEAR+  1    OBTAIN M
        SP    Q            6
        LT    0            COMMON+ 2    SIGN OF POWER
        TP    A            Q
        RA    STOR         N1           BUMP STORE
        TP    N1           COMMON+ 3    INDEX TO 1
        RJ    RET          CLEAR        OBTAIN POWER
        RS    COMMON+ 6    N8
        MP    LGB2T        COMMON+ 6
        LT    3            COMMON+ 6    INTEGER
        LQ    A            35           POSITION FRACT
        SJ    NEGAT        PST          IF NEGATIVE
NEGAT   RS    COMMON+ 6    N1           INT - 1
        SP    ETAB         1            FRACTION
        AT    Q            Q            PLUS 1
PST     SP    Q            4            2EX ENTRY
        LT    0            A            FIRST 3 BITS
        AT    MPDUM        MUPL         SET TABLE MPY
        QT    WMASK        Q
        MP    Q            EXP1
        LT    0            A
        AT    EXP2         COMMON+ 3
        MP    Q            COMMON+ 3
        LT    0            A            PREV 34
        AT    EXP3         COMMON+ 3
        SP    Q            33
        DV    COMMON+ 3    A
        AT    ETAB         Q
MUPL    FILL  FILL         FILL
        LT    2            A
        MP    A            COMMON+ 5
        ZJ    NZERO        ZEXIT        EXIT IF ZERO
NZERO   SF    A            SFACT        NORMALIZE
        TP    A            A            POSITION
        LT    28           COMMON+ 5     MANTISSA
        RA    COMMON+ 6    N129         INT 129
        AT    SFACT        COMMON+ 6
        SJ    ZEXIT        OVTST        IF NEGATIVE
ZEXIT   ST    A            Q             SET FOR ZERO
        MJ    0            PSTOR         AND STORE
OVTST   TJ    N256         PST3
        TP    N1           OVIND
```

FORM #
MSD 8520

LOC	OP	U-ADDR		V-ADDR		EXPLANATION
	TN	NEMAX		A		MAX TO A
	TP	COMMON+	5	Q		MANT TO Q
	QJ	CHSGN		CHSGN+	1	IF NEG
CHSGN	TN	A		A		CHANGE SIGN
	TP	N5		Q		FIVE TO Q
	MJ	0		STST		AND EXIT
PST3	LTR	27		Q		POSITION CHAR
	TP	COMMON+	5	A		MANT TO A
STST	SJ	ADJC		PSTOR		IF NEG
ADJC	TN	Q		Q		ADJUST CHAR
PSTOR	AT	Q		FILL		PACK AND STORE
	RA	PSTOR		N1		BUMP STORE
	EJ	OVDUM		OVER		
	RA	WORD		U2		BUMP WORD FETCH
	IJ	COMMON+	1	WORD-	1	TEST FOR END
	MJ	0		CYCLE		RET FOR NEXT BK
CLEAR	TP	NZ		COMMON+	4	CLEAR SUM
POLY	LQ	Q		6		
	SP	COMMON+	4	2		SUM
	SA	COMMON+	4	1		TIMES TEN
	QA	MASK1		A		DIGIT
	ST	N3		COMMON+	4	-XS 3
	IJ	COMMON+	3	POLY		
	RS	COMMON+	2	N2		TEST
	ZJ	STOR		STOR-	1	SIGN
	TN	COMMON+	4	COMMON+	4	
STOR	TP	COMMON+	4	FILL		STORE
RET	MJ	0		FILL		
OVER	SP	CODE2		0		
	MJ	0		RETRN		
BEXIT	SP	OVIND		0		
	ZJ	SIND		EXIT		
SIND	SP	CODE3		0		
	MJ	0		RETRN		
BKSP	02	614B		1		
READ	02	602B		0		
MPDUM	MP	Q		ETAB		MULP DUMMY
SFACT	B			0		SCALE FACTOR
OVIND	B			0		OVERFLOW IND
NZ	B			0		
N1	B			1		
N2	B			2		
N3	B			3		
N4	B			4		
N5	B			5		
N8	B			10		
N23	B			27		
N48	B			60		
N129	B			201		
N256	B			400		
OVDUM	B			353100010000		OVERFLOW IND
WMASK	B			37777777777		CONVERSION MASK

LOC	OP	U-ADDR	V-ADDR	EXPLANATION
TMASK	B		170000	TAPE IND MASK
MASK1	B		77	XS-3 MASK
VMASK	B		77777	V ADDR MASK
BIAS	B		20000150000	NORMAL BIAS
NEMAX	B		400000000000	CONVERSION CONSTAN
EXP1	B		35440262675	CONV COEFFICIENT
EXP2	B		600000171150	CONV COEFFICIENT
EXP3	B		270524354513	LOG BS 2 OF E
LGB2T	B		324464741134	LOG BS 2 OF 10
U2	B		200000	2 IN U ADDR
CODE1	B		660303050001	TAPE ERROR CODE
CODE2	B		660303050002	CORE OVFL CODE
CODE3	B		660303050003	POWER OVFL CODE
ETAB	B		200000000000	2 TO 0
	B		213453407440	2 TO .125
	B		230157701214	2 TO .25
	B		245775532516	2 TO .375
	B		265011714640	2 TO .5
	B		305316250212	2 TO .625
	B		327211763126	2 TO .75
	B		352601433477	2 TO .875
	END			

Talmadge

1. IDENTIFICATION

MLC001 Numerical Card Data Input
Frank Bozgeh, April 2, 1957
Lockheed Missile Systems Division

2. PURPOSE

To read and store into core or drum locations, from variable field cards, both stated point and floating point decimal numbers.

3. METHOD

This program reads "N" number of variable field cards and stores the converted number in specified locations.

The conversion and storing of the information is accomplished between card reads.

4. USAGE

A. Calling Sequence

Loc	Op	u	v
r	RJ	t+2	t
r+1	Normal Return		

B. Control and Results
A parameter word in octal notation following the form 00 bbbbb ccccc, must be placed in location t+3 where:
bbbbb) Number of cards to be read
ccccc) Relative address. This may be zero.

C. Space Required
280 cells of instructions
66 cells of constants
42 temporary cells

D. Card Format:
1. The variable field card input uses only the first 72 columns.
A number that would require punching beyond the first 72 columns may be continued in column 1 of the succeeding card if there are 72 non blank columns in the first card. The reading of a card is terminated by any of the following:

a. A blank column found anywhere in the first 72 columns.
b. When 72 non blank columns have been read.
c. By the letter "C" found anywhere in the first 72 columns.

2. Loading of the Input is terminated in one of two ways, whichever occurs first:

   a. When the letter "C" appears on a card, or,
   b. When "N" cards, as specified in the "u" address of the parameter word, have been read.

3. Decimal Storage Addresses in the form of, LXXXXX, may appear anywhere on any card. The number immediately following such an address will be stored in the cell whose address is formed by adding to the LXXXXX address, the relative address in the v portion of the parameter word. The following numbers will be stored in consecutive cells until another address appears.

4. A comma separates all fields of the variable field card input. Complete numbers may be duplicated into consecutive cells by successive commas.

5. Floating point numbers are written in the form ±XXXXXXXXX±FXX. Plus signs need not be specified. No decimal point may be written in a floating point number, it is assumed that the decimal point precedes the left most written digit. The fractional part of the floating point number may be less than but cannot exceed nine decimal digits.

6. Stated point numbers are specified by a sign followed by not more than eleven decimal digits. Plus signs need not be written. A decimal point may appear anywhere in a stated point number.

7. Scale factors are associated with each stated point number. The decimal scaling is in the form D±XX and the binary scaling is in the form BXX. The binary scaling is actually the location of the binary point obtained by counting from the right, starting with the bit zero of the word as it appears in the machine. The plus sign need not be written in the decimal scaling factor, a minus sign is not allowed in the binary scaling factor.

   The decimal scaling factor must not exceed ±11. A minus zero will be stored in place of the number if the scaling factor is too large.

   After once being specified the decimal and/or binary scaling of a number is duplicated in following stated point numbers until new scale factors are encountered. Zero scaling factors are specified by writing only the associated alphabetical character.

E. Error Codes

1. The parameter word is left in the Q-Register and the following error codes are left in the accumulator on return through the error exit.

Code	Explanation
$C001 \cdot 2^{12}+1$	Core overflow while storing information
$C001 \cdot 2^{12}+2$	Power overflow during floating conversion

2

5. RESTRICTIONS

A. The parameter word is expressed in octal notation.

B. Decimal scaling factors cannot exceed eleven.

C. Negative binary scaling is not permissible.

D. Scaling factors are duplicated.

6. CODING INFORMATION

A. Constants.

Location	Constant	Explanation
L(0)	000000000000	Zero
L(1)	000000000001	One
CN2	000000000020	Twenty
CN3	000000100000	Mod u address
CN4	000000000024	Comma
CN5	000000000045	B
CN6	000000000047	D
CN7	000000000051	F
CN8	000000000026	L
CN9	000000000046	C
CN10	000000000003	Zero XS-3
CN11	000000000004	One XS-3
CN12	000000000005	Two XS-3
CN13	000000000006	Three XS-3
CN14	000000000007	Four XS-3
CN15	000000000010	Five XS-3
CN16	000000000011	Six XS-3
CN17	000000000012	Seven XS-3
CN18	000000000013	Eight XS-3
CN19	000000000014	Nine XS-3
CN20	000000000061	Point

CN21	525252525252	Alternator
CN22	00 CN1900000	CN19 in u address
CN23	00 CN2400000	CN24 in u address
CN24	00IDI00IDI00	Left Row Dummy
CN25	00 CN2600000	CN26 in u address
CN26	00ICI00ICI00	Right Row Dummy
CN27	000000000077	Mask
CN28	000000000020	Minus Sign
CN29	400000000000	Indicator
CN30	000000000015	A number test
CN31	00 CN3200000	CN32 in u address
CN32	000000000001	10^0
CN33	000000000012	10^1
CN34	000000000144	10^2
CN35	000000001750	10^3
CN36	000000023420	10^4
CN37	000000303240	10^5
CN38	000003641100	10^6
CN39	000046113200	10^7
CN40	000575360400	10^8
CN41	000734654500	10^9
CN42	112402762000	10^{11}
CN43	324464741134	Log 10 Base 2
CN44	200000000000	1 scaled 34
CN45	037777777777	Mask
CN46	MPQ00000CN47	Dummy for T.L.U.
CN47	200000000000	2 to o/8 S-34
CN48	213453407440	2 to 1/8 S-34
CN49	230157701214	2 to 2/8 S-34

CN50	245775532516	2 to 3/8 S-34
CN51	265011714640	2 to 4/8 S-34
CN52	305316250212	2 to 5/8 S-34
CN53	327211763126	2 to 6/8 S-34
CN54	352601433477	2 to 7/8 S-34
CN55	035440262675	A2 Scaled 36
CN56	600000171150	A1 Scaled 35
CN57	270524354513	Log E Base 2 S-34
CN58	000000000201	Floating Pt power
CN59	000000000400	Test for power overflow
CN60	400000000005	EF-Read cards
CN61	000000000040	Plus sign
CN62	777777777777	Power overflow number
CN63	310303040001	Core overflow code
CN64	310303040002	Power overflow code

B. Working Storage

30 Cells

C. Timing

Unknown

```
ENTRY  ,  MJ   ,  ,START,$
ERROR  ,  RJ   ,  DIAG+2,DIAG,ERROR EXIT$
EXIT   ,  MJ   ,  ,FILL,$
Y      ,  00   ,  FILL,FILL,$
START  ,  SP   ,  Y,,CONTROL TO ACC$
       ,  TV   ,  A,TM5,REL LOCATION TO TM5$
       ,  TV   ,  A,STORE,REL LOCATION TO STORE$
       ,  LT   ,  21,IDX4,SET INDEX 4 TO N CARDS$
       ,  TP   ,  L(0),TM6,SET M EQ TO ZEROS$
       ,  TP   ,  L(0),TM7,SET S EQ TO ZEROS$
       ,  TP   ,  L(B1),TM8,SET T EQ TO 1$
TSTDX4,   IJ   ,  IDX4,RECRD,TEST INDEX 4$
       ,  MJ   ,  ,EXIT,EXIT$
RECRD  ,  EF   ,  ,CN60,EF CARD READ$
       ,  TP   ,  CN21,ALTN,SET ALTERNATOR$
SEDX1  ,  TP   ,  CN16,IDX1,SET INDEX 1$
       ,  TU   ,  CN22,DIGAD,SET DIGIT TO NINE$
       ,  RPV  ,  14)B,READ1,$
       ,  TP   ,  L(0),IDI00,CLEAR TEMP STORAGES$
READ1  ,  ER   ,  00000,A,READ JUNK$
       ,  ERB  ,  ,LEROW,READ LEFT ROW$
       ,  ERB  ,  ,RTROW,READ RIGHT ROW$
       ,  TP   ,  LEROW,Q,LEFT ROW TO Q$
       ,  TU   ,  CN23,MOD1,$
SEDX2  ,  TP   ,  CN12,IDX2,SET INDEX 2$
MOD1   ,  TU   ,  CN24,MOD2,$
SEDX3  ,  TP   ,  CN12,IDX3,SET INDEX 3$
       ,  TP   ,  L(0),TEE,SET T EQ TO ZEROS$
TASL1  ,  QJ   ,  YES,NO,TEST AND SHIFT LEFT$
NO     ,  LA   ,  TEE,6,SHIFT T LEFT SIX$
       ,  MJ   ,  ,TSTDX3,$
YES    ,  SP   ,  TEE,6,$
DIGAD  ,  AT   ,  CN19,TEE,ADD IN DIGIT$
TSTDX3,   IJ   ,  IDX3,TASL1,TEST INDEX 3$
MOD2   ,  RA   ,  IDI00,TEE,$
       ,  RA   ,  MOD2,CN3,$
TSTDX2,   IJ   ,  IDX2,SEDX3,TEST INDEX 2$
       ,  TP   ,  ALTN,Q,ALTERNATER TO Q$
       ,  QJ   ,  STOAL,MODIG,TEST ALT$
STOAL  ,  TP   ,  Q,ALTN,$
       ,  TU   ,  CN25,MOD1,$
       ,  TP   ,  RTROW,Q,RIGHT ROW TO Q$
       ,  MJ   ,  ,SEDX2,RETURN FOR RIGHT ROW$
MODIG  ,  RS   ,  DIGAD,CN3,MODIFY DIGIT$
       ,  TP   ,  Q,ALTN,$
TSTDX1,   IJ   ,  IDX1,READ1,TEST INDEX1$
       ,  TP   ,  L(1),IDX1,SET INDEX 1 TO 2$
       ,  TP   ,  CN2,TM12,SET DIGIT FOR X$
READ2  ,  ER   ,  00000,A,READ JUNK$
       ,  ERB  ,  ,LEROW,READ LEFT ROW$
       ,  ERB  ,  ,RTROW,READ RIGHT ROW$
       ,  TP   ,  LEROW,REG,LEFT ROW TO R$
       ,  TU   ,  CN23,SUADD,SET TO PICK D$
SEDX2A,   TP   ,  CN12,IDX2,SET INDEX 2 TO 6$
SUADD  ,  SP   ,  CN24,,SET UP ADD$
       ,  TV   ,  A,TOB,PRESET COMM$
       ,  TU   ,  A,BOT,SET UP B PICKUP$
SEDX3A,   TP   ,  CN12,IDX3,SET INDEX 3ATO 6$
BOT    ,  TP   ,  IDI00,TEE,B TO T$
```

```
RTOQ     ,  TP   , REG,Q,R TO Q$
         ,  QJ   , REST1,REST2,TEST AND SHIFT$
REST1    ,  TP   , Q,REG,RESTORE R$
         ,  LQ   , TEE,32006)B,$
         ,  AT   , TM12,TEE,ADD IN X CHAR$
         ,  MJ   , ,TSDX3A,$
REST2    ,  TP   , Q,REG,RESTORE R$
         ,  LQ   , TEE,6,SHIFT T LEFT 6$
TSDX3A,     IJ   , IDX3,RTOQ,TEST INDEX 3A$
TOB      ,  TP   , TEE,IDI00,T TO B$
         ,  RA   , TOB,L(B1),MODIFY K$
         ,  RA   , BOT,CN3,$
         ,  IJ   , IDX2,SEDX3A,TEST INDEX 2A$
         ,  TP   , ALTN,Q,$
         ,  QJ   , RSAL1,RSAL2,$
RSAL1    ,  TP   , Q,ALTN,$
         ,  TU   , CN25,SUADD,SET TO PICK C$
         ,  TP   , RTROW,REG,RIGHT ROW TO R$
         ,  MJ   , ,SEDX2A,RETURN FOR RIGHT ROW$
RSAL2    ,  TP   , Q,ALTN,$
         ,  LQ   , TM12,1,MODIFY X TO Y CHAR$
         ,  IJ   , IDX1,READ2,TEST INDEX 2$
         ,  TU   , CN23,SEUAD,$
SEDXR1,     TP   , L(1),IDX1,SET RT INDEX 1$
SEDXR2,     TP   , CN12,IDX2,SET RT INDEX 2$
SEUAD    ,  SP   , FILL,15,SET UP U ADDRESS$
         ,  TU   , A,NTQL6,$
SEDXR3,     TP   , CN12,IDX3,SET RT INDEX 3$
NTQL6    ,  LQ   , FILL,6,NUMBER TO Q LEFT 6$
         ,  QT   , CN27,A,MASK 6 LSD TO A$
         ,  TP   , TM27,TM28,PREVIOUS XS3 DIGIT$
         ,  TP   , A,TM27,CURRENT XS3 DIGIT$
         ,  ZJ   , NOBLK,BLANK,ACC EQ ZERO$
BLANK    ,  MJ   , ,TSTDX4,BLANK NEXT CARD$
NOBLK    ,  EJ   , CN61,PLUS,A PLUS SIGN$
         ,  EJ   , CN28,MINUS,A MINUS SIGN$
         ,  TJ   , CN30,ANO,A NUMBER$
         ,  EJ   , CN8,ANL,ANL$
         ,  EJ   , CN7,ANF,ANF$
         ,  EJ   , CN6,ADEE,AD$
         ,  EJ   , CN5,ABEE,AB$
         ,  EJ   , CN20,APT,A POINT$
         ,  EJ   , CN4,ACOM,A COMMA$
         ,  EJ   , CN9,EXIT,AC-EXIT$
         ,  RJ   , ,ERROR,ERROR$
PLUS     ,  TP   , L(0),SIGIND,CLEAR SIGN IND$
         ,  MJ   , ,RTDX3,$
MINUS    ,  TP   , CN29,SIGIND,SETSIGN IND MINUS$
         ,  MJ   , ,RTDX3,$
ANO      ,  ST   , CN10,Q,NUMBER MINUS 3$
         ,  SP   , TM7,2,STO ACC$
         ,  SA   , TM7,1,MULT BY 10$
         ,  AT   , Q,TM7,ADD IN DIGIT$
         ,  RA   , TM6,TM8,M PLUS T TO M$
         ,  MJ   , ,RTDX3,$
ANL      ,  TP   , CN29,LIND,SET L INDICATOR$
CLEARS,     TP   , L(0),TM7,CLEAR S$
         ,  MJ   , ,RTDX3,$
ANF      ,  TP   , CN29,FIND,SET F INDICATOR$
```

7

```
        , TP  , TM7,NFLOT,S TO N FLOATS$
CLEAT   , TP  , L(O),TM8,CLEAR T$
        , TP  , L(O),NSIGN,$
        , TP  , SIGIND,Q,SIGN IND TO Q$
        , QJ  , SENSIG,CLEARS,TEST SIGN IND$
SENSIG, , TP  , CN29,NSIGN,SET NO SIGN IND$
CLESIG, , TP  , L(O),SIGIND,CLEAR SIGN IND$
        , MJ  , ,CLEARS,$
ADEE    , TP  , L(O),DSCF,CLEAR D SCALE FACTOR$
        , TP  , CN29,DIND,SET D INDICATOR$
STONS   , TP  , TM7,NSTAT,S TO N STATED$
        , MJ  , ,CLEAT,$
ABEE    , TP  , CN29,BIND,SET B INDICATOR$
        , TP  , L(O),BSCF,CLEAR B SCALE FACTOR$
        , TP  , DIND,Q,D INDICATOR TO Q$
        , QJ  , STOD,STONS,$
STOD    , TP  , TM7,DSCF,S TO D SCALE FACTOR$
        , TP  , L(O),DIND,CLEAR D INDICATOR$
        , TP  , SIGIND,Q,SIGN IND TO Q$
        , QJ  , NEGED,CLEARS,TEST SIGN IND$
NEGED   , TN  , DSCF,DSCF,NEG D SCALE FACTOR$
        , MJ  , ,CLESIG,$
APT     , TP  , L(O),TM6,SET M EQ TO ZERO$
        , TP  , CN29,PIND,SET PT INDICATOR$
RTDX3   , IJ  , IDX3,NTQL6,TEST RT INDEX 3$
        , RA  , NTQL6,CN3,MODIFY K$
        , IJ  , IDX2,SEDXR3,TEST INDEX RT2$
        , TU  , CN25,SEUAD,CHANGE SEUAD$
        , IJ  , IDX1,SEDXR2,TEST INDEX RT1$
        , MJ  , ,TSTDX4,RETURN TO MAIN ROUTINE$
ACOM    , EJ  , TM28,STORE,COMMA BY COMMA$
        , TP  , LIND,Q,L IND TO Q$
        , QJ  , LSET,FINDQ,TEST L IND$
LSET    , TP  , TM7,A,S TO ACC$
        , AT  , TM5,A,ADD IN REL LOC$
        , TV  , A,STORE,MODIFY STORE INSTRUCTION$
CLEAM   , TP  , L(O),TM6,CLEAR M$
        , TP  , L(O),TM7,CLEAR S$
        , TP  , L(O),LIND,CLEAR L IND$
        , TP  , L(B1),TM8,SET EQ TO 1$
        , MJ  , ,RTDX3,$
FINDQ   , TP  , FIND,Q,FIND TO Q$
        , QJ  , FSET,DEEQ,TEST F INDICATOR$
FSET    , TP  , TM7,FLOAT,S TO F$
        , TP  , L(O),FIND,CLEAR F INDICATOR$
        , TP  , SIGIND,Q,SIGN INDICATOR TO Q$
        , QJ  , NEGF,FLOP,TEST SIGN INDICATOR$
NEGF    , TN  , FLOAT,FLOAT,NEGATIVE F$
        , TP  , L(O),SIGIND,CLEAR SIGNIND$
        , MJ  , ,FLOP,TO FLOATING PT CONV.$
DEEQ    , TP  , DIND,Q,DINDICATOR TO Q$
        , QJ  , DSET,BEEQ,TEST D INDICATOR$
DSET    , TP  , L(O),DIND,CLEAR D INDICATOR$
        , TP  , SIGIND,Q,SIGN INDICATOR$
        , QJ  , NEGD,STOD2,TEST DSF SIGN$
NEGD    , TN  , TM7,DSCF,MINUS DSF$
        , TP  , L(O),SIGIND,CLEAR SIGN INDICATOR$
        , MJ  , ,STAP,TO SP CONVERSION$
STOD2   , TP  , TM7,DSCF,S TO DSF$
```

```
        ,  MJ  ,  ,STAP,TO SP CONVERSION$
BEEQ    ,  TP  ,  BIND,Q,B INDICATOR$
        ,  QJ  ,  BSET,STONS2,TEST B INDICATOR$
BSET    ,  TP  ,  TM7,BSCF,S TO BSF$
        ,  TP  ,  L(0),BIND,CLEAR B INDICATOR$
        ,  MJ  ,  ,STAP,TO SP CONVERSION$
STONS2, TP  ,  TM7,NSTAT,S TO N STATED$
        ,  TP  ,  SIGIND,Q,SIGN INDICATOR$
        ,  QJ  ,  NEGN,STAP,TEST N SIGN$
NEGN    ,  TP  ,  CN29,NSIGN,SET NEG NUMBER IND$
        ,  TP  ,  L(0),SIGIND,CLEAR SIGN INDICATOR$
STAP    ,  TM  ,  DSCF,A,D TO A$
        ,  SS  ,  CN19,,D MINUS 12$
        ,  SJ  ,  DOK,TOOBIG,TEST SIZE OF DSCF$
DOK     ,  TP  ,  DSCF,A,$
        ,  MJ  ,  ,PTOQ,$
TOOBIG, TP  ,  CN62,RESULT,DSF TOO LARGE$
        ,  MJ  ,  ,STORE,$
PTOQ    ,  TP  ,  PIND,Q,PT INDICATOR TO Q$
        ,  QJ  ,  PSET,PNSET,$
PNSET   ,  TP  ,  DSCF,TEE,$
        ,  MJ  ,  ,SPCON,CONVERSION$
PSET    ,  TP  ,  L(0),PIND,CLEAR PT INDICATOR$
        ,  ST  ,  TM6,TEE,D MINUS M$
        ,  MJ  ,  ,SPCON,CONVERSION$
STORE   ,  TP  ,  RESULT,FILL,STORE NUMBER$
        ,  RA  ,  STORE,L(B1),BUMB LOCATION$
        ,  EJ  ,  CN65,COFLO,TEST CORE OVERFLOW$
        ,  MJ  ,  ,CLEAM,RETURN$
COFLO   ,  SP  ,  CN63,0,CORE OVERFLOW TO A$
RETURN, TP  ,  Y,Q,PARAMETER TO Q$
        ,  MJ  ,  ,ERROR,TO ERROR EXIT$
SPCON   ,  TP  ,  TEE,A,D TO A$
        ,  ZJ  ,  DENZO,DEZO,D EQ ZERO$
DENZO   ,  SJ  ,  DELZO,DEGRO,D LESS THAN ZERO$
DELZO   ,  SN  ,  TEE,15,ABSOLUTE D$
        ,  SA  ,  CN31,,ADD TABLE ORIG$
        ,  TU  ,  A,DIVIG,SET UP DIVIDE$
        ,  TV  ,  BSCF,SHIFT,SET UP SHIFT$
        ,  SP  ,  NSTAT,,N TO ACC$
DIVIG   ,  DV  ,  FILL,REG,DIVIDE BY 10D$
        ,  SP  ,  A,35,SHIFT REMAINDER$
        ,  TU  ,  DIVIG,DIVREM,SETUP REMAIN DIVIDE$
DIVREM, DV  ,  FILL,TM29,REMAINDER BY 10$
        ,  SP  ,  REG,35,INTEGER TO A$
        ,  SA  ,  Q,37,FRACTION TO A$
SHIFT   ,  LA  ,  A,FILL,RESULT IN A$
        ,  MJ  ,  ,RESIGN,$
DEZO    ,  TV  ,  BSCF,SHIFT2,SET UP SHIFT$
SHIFT2, SP  ,  NSTAT,FILL,SHIFT N$
        ,  MJ  ,  ,RESIGN,$
DEGRO   ,  TV  ,  BSCF,SHIFT3,$
        ,  SP  ,  TEE,15,D TO U$
        ,  SA  ,  CN31,,ADD TABLE ORIG$
        ,  TU  ,  A,MPYN,SET UP MPY$
MPYN    ,  MP  ,  FILL,NSTAT,MPY 10D NST$
SHIFT3, LA  ,  A,FILL,SHIFT PRODUCT$
RESIGN, TP  ,  NSIGN,Q,N SIGN TO Q$
        ,  QJ  ,  NNEG,NPOS,TEST INDICATOR$
```

9

FORM # MSD 8520

```
NNEG    ,  TN   ,  A,RESULT,NEGATIVE N$
        ,  MJ   ,  ,STORE,TO STORE$
NPOS    ,  TP   ,  A,RESULT,POSITIVE N$
        ,  MJ   ,  ,STORE,TO STORE$
FLOP    ,  TP   ,  FLOAT,A,F TO A$
        ,  ST   ,  TM6,Q,F MINUS M TO Q$
        ,  MP   ,  Q,CN43,E LOG 10$
        ,  LT   ,  3,LEROW,SAVE INTEGER$
        ,  LQ   ,  A,35,FRACTION TO A$
        ,  SJ   ,  EXNEG,EXPOS,EXPONET NEG$
EXNEG   ,  RS   ,  LEROW,L(B1),INTEGER PLUS 1$
        ,  SP   ,  CN29,,1SCALED 35$
        ,  AT   ,  Q,Q,GET POS COMP$
EXPOS   ,  SP   ,  Q,4,$
        ,  LT   ,  ,A,$
        ,  AT   ,  CN46,FPMY,TLU SETUP$
        ,  QT   ,  CN45,Q,MASK ALL BUT 3$
        ,  MP   ,  Q,CN55,$
        ,  LT   ,  ,A,$
        ,  AT   ,  CN56,RTROW,$
        ,  MP   ,  Q,RTROW,$
        ,  LT   ,  ,A,$
        ,  AT   ,  CN57,RTROW,$
        ,  SP   ,  Q,33,$
        ,  DV   ,  RTROW,A,$
        ,  AT   ,  CN44,Q,$
FPMY    ,  MP   ,  Q,FILL,$
        ,  LT   ,  2,A,$
        ,  MP   ,  A,NFLOT,$
        ,  SF   ,  A,TM12,$
        ,  TP   ,  A,REG,$
        ,  RA   ,  LEROW,CN58,$
        ,  AT   ,  TM12,LEROW,$
        ,  SJ   ,  ZETOA,OFLO,$
ZETOA   ,  TP   ,  L(0),A,ZERO TO ACC$
        ,  MJ   ,  ,REGITA,$
OFLO    ,  TJ   ,  CN59,SLEROW,OVERFLOW$
        ,  SP   ,  CN64,0,POWER OFLO TO A$
        ,  MJ   ,  ,RETURN,TO ERROR RETURN$
SLEROW, LQ      ,  LEROW,27,$
REGITA, TP      ,  REG,A,$
        ,  LT   ,  28,A,$
        ,  AT   ,  Q,A,$
        ,  MJ   ,  ,RESIGN,$
CN2     ,  B    ,  20,,TWENTY$
CN3     ,  B    ,  0000001,00000,$
CN4     ,  B    ,  24,,COMMA$
CN5     ,  B    ,  45,,B$
CN6     ,  B    ,  47,,D$
CN7     ,  B    ,  51,,F$
CN8     ,  B    ,  26,,L$
CN9     ,  B    ,  46,,C$
CN10    ,  B    ,  3,,0$
CN11    ,  B    ,  4,,ONE XS3$
CN12    ,  B    ,  5,,TWO XS3$
CN13    ,  B    ,  6,,THREE XS3$
CN14    ,  B    ,  7,,FOUR XS3$
CN15    ,  B    ,  10,,FIVE XS3$
CN16    ,  B    ,  11,,SIX XS3$
```

```
CN17  ,  B       ,  12,,SEVEN XS3$
CN18  ,  B       ,  13,,EIGHT XS3$
CN19  ,  B       ,  14,,NINE XS3$
CN20  ,  B       ,  61,,PT$
CN21  ,  B       ,  5252525,25252,ALTERNATOR$
CN22  ,          ,  CN19,,$
CN23  ,          ,  CN24,,$
CN24  ,          ,  IDI00,IDI00,$
CN25  ,          ,  CN26,,$
CN26  ,          ,  ICI00,ICI00,$
CN27  ,  B       ,  77,,MASK$
CN28  ,  B       ,  20,,NEGATIVE SIGN$
CN29  ,  B       ,  4000000,00000,INDICATOR$
CN30  ,  B       ,  15,,$
CN31  ,          ,  CN32,,$
CN32  ,  B       ,  1,,TEN TO ZERO$
CN33  ,  B       ,  12,,TEN TO ONE$
CN34  ,  B       ,  144,,TEN TO TWO$
CN35  ,  B       ,  1750,,TEN TO THREE$
CN36  ,  B       ,  23420,,TEN TO FOUR$
CN37  ,  B       ,  303240,,TEN TO FIVE$
CN38  ,  B       ,  3641100,,TEN TO SIX$
CN39  ,  B       ,  4611320,0,TEN TO SEVEN$
CN40  ,  B       ,  5753604,00,TEN TO EIGHT$
CN41  ,  B       ,  73465,45000,TEN TO NINE$
CN42  ,  B       ,  1124027,62000,TEN TO ELEVEN$
CN43  ,  B       ,  3244647,41134,LOG 10 TO BASE 2$
CN44  ,  B       ,  2000000,00000,1 SCALED 34$
CN45  ,  B       ,  0377777,77777,MASK$
CN46  ,  MP      ,  Q,CN47,DUMMY FOR TLU$
CN47  ,  B       ,  2000000,00000,2 TO 0-8 S34$
CN48  ,  B       ,  2134534,07440,2 TO 1-8 S34$
CN49  ,  B       ,  2301577,01214,2 TO 2-8 S34$
CN50  ,  B       ,  2457755,32516,2 TO 3-8 S34$
CN51  ,  B       ,  2650117,14640,2 TO 4-8 S34$
CN52  ,  B       ,  3053162,50212,2 TO 5-8 S34$
CN53  ,  B       ,  3272117,63126,2 TO 6-8 S34$
CN54  ,  B       ,  3526014,33477,2 TO 7-8 S34$
CN55  ,  B       ,  0354402,62675,A2 SCALED 36$
CN56  ,  B       ,  6000001,71150,A1 SCALED 35$
CN57  ,  B       ,  2705243,54513,LOG E BASE 2 S34$
CN58  ,  B       ,  201,,12.9$
CN59  ,  B       ,  400,,$
CN60  ,  B       ,  4000000,00005,EF READ CARDS$
CN61  ,  B       ,  40,,POSITIVE SIGN$
CN62  ,  B       ,  7777777,77777,$
CN63  ,  B       ,  31030,3040001,CORE OFLO CODE$
CN64  ,  B       ,  31030,3040002,POWER OFLO CODE$
CN65  ,  TP      ,  RESULT,10000)B,CORE OFLO IND$
IDI00 ,  RESERV, 6,6,$
ICI00 ,  RESERV, 6,6,$
DIAG  ,  RESERV, 3,3,$
      ,  END     ,  ,,$
```

FORM # MSD 8520

Talmadge

A. IDENTIFICATION.

ML D001 OCTAL DUMP
T. H. DEWEY
16 APRIL 1957
LOCKHEED MISSILE SYSTEMS DIVISION

B. PURPOSE.

To dump core or drum on any designated uniservo.

C. METHOD.

The program prepares an XS-3 tape, suitable for listing on the off-line high speed printer, which contains the contents of memory from A to B, where A and B are specified in a parameter word. The program preserves core by making use of a core image on drum. A 264 word buffer region is constructed, pulling out 6 words per blockette and replenishing the buffer when necessary.

A sentinel block is appended to the end of each dump consisting of a printer stop symbol repeated 720 times. Replace the constant at c+17 to substitute any other sentinel.

D. USAGE.

1. The dump was programmed with the intention of being initiated by a drum start. Modifications have been made in order to initiate it by a normal USE entry, to wit:

```
r        RJ     t+2        t            to dump
r+1         normal return
```

2. There are five parameter words as follows:

(t+3) = (Q)

(t+4) = (A_L)

(t+5) = (A_R)

(t+6) = 00 aaaaa bbbbb

where aaaaa is starting address
and bbbbb is last address to be dumped;

D. USAGE. (continued)

If (t+6) = 0 then entire core is dumped.

(t+7) = 00 00000 0000T
where T designates uniservo for dumping;

If (t+7) = 0 then uniservo 5 is used.

3. Space required

234 words of instructions and constants on drum
158 words of instructions in core
65 words of erasable and constants in core
636 words of image on drum

4. Tape format

Each block of tape contains the contents of 36 consecutive locations. In addition a heading block precedes each dump containing the contents of Q, A_L, A_R, CORE F_1, F_2, F_3, and the status of the manual jump switches.

E. RESTRICTIONS.

1. When specifying dump limits A and B must be either both core or both drum addresses. No continuous dumping from core to drum or drum to core is allowed.

2. The MJ1 switch is used to determine whether or not more than one dump is to be put on a given uniservo. If the MJ1 switch is on, then after the program has dumped from A to B the tape is left positioned to receive another dump. If the MJ1 switch is off the tape is rewound. In either case core is restored, if necessary, and control is returned to the programmer.

3. That part of drum designated by the programmer to be the drum core image cannot be dumped, since the lower part of core is immediately placed in the image when entering the dump program. This restriction can be avoided by a suitable modification.

F. CODING INFORMATION.

1. Numerical constants

22 cells labeled c thru c + 21

2. Alphabetic heading constants

31 cells labeled:

M thru M + 5
H thru H + 14
H2 thru H2 + 9

3. Buffer storage

264 cells labeled z thru z + 263

4. Tape write image

120 cells labeled IM thru IM + 119

5. Working storage

49 cells labeled:

IND thru IND +7
W thru W +6
V thru V +13
L thru L +19

6. Timing

Unknown

```
DCI     ,  EQUALS,  46600)B,,          DEFINE IMAGE
DIAG    ,  EQUALS,  50000)B,,          DEFINE ERROR ROUTINE
        ,  SETLOC,  45600)B,45600)B,
T       ,  MJ     , 0,START,          ENTRY
        ,  RJ     , DIAG+2,DIAG,
        ,  MJ     , 0,FILL,           EXIT
        ,  00     , FILL,FILL,          Q
        ,  00     , FILL,FILL,          AL
        ,  00     , FILL,FILL,          AR
        ,  00     , FILL,FILL,          DUMP LIMITS
        ,  00     , FILL,FILL,          UNISERVO
START   ,  TP     , 0,DCI,            PRESERVE F1
        ,  RPB    , 636,START+2,          PRESERVE CORE
        ,  TP     , 1,DCI+1,
        ,  RPB    , 220,DUMP,           PROGRAM TO CORE
        ,  TP     , RETURN+2,1,
RETURN, ,  RPB    , 636,T+2,            RESTORE CORE
        ,  TP     , DCI+1,1,
        ,  SETLOC,  1,RETURN+2,
DUMP    ,  RPB    , 3,HD3,            PICK F1 F2 F3
HD2     ,  TP     , DCI,W+3,
HD3     ,  RPB    , 3,HD5,            PICK Q AL AR
        ,  TP     , T+3,W,
HD5     ,  RJ     , KF,KA,            CONVERT
        ,  RJ     , BL,BL,            BLANK OUT IMAGE
        ,  RPB    , 15,HD9,
        ,  TP     , H,IM+3,            HEDING TO IMAGE
HD9     ,  RPB    , 10,HD11,            JUST 10 MORE BLANKS
        ,  TP     , H2,IM+90,
HD11    ,  TP     , C+21,IM,            HOME PAPER
        ,  TP     , V+2,IM+22,            AL L
        ,  TP     , V+3,IM+23,            AL R
        ,  SP     , V+4,66,
        ,  AT     , C+15,IM+24,            AR TOP
        ,  SP     , V+4,36,
        ,  SA     , V+5,30,
        ,  LTL    , ,IM+25,            AR MIDDLE
        ,  SP     , V+5,,
        ,  AT     , C+14,IM+26,            AR BOT
HD21    ,  TP     , V,IM+28,            QL
        ,  TP     , V+1,IM+29,            QR
        ,  TP     , V+6,IM+110,            F1L
        ,  TP     , V+7,IM+111,            F1 R
        ,  TP     , V+8,IM+114,            F2 L
        ,  TP     , V+9,IM+115,            F2 R
        ,  TP     , V+10,IM+118,            F3 L
        ,  TP     , V+11,IM+119,            F3 R
        ,  RPB    , 3,HD31,
        ,  TP     , M,IM+35,            SWITCHES ON
HD31    ,  MJ1    , 0,HD33,
        ,  TP     , M+3,IM+35,            SWITCH 1 OFF
HD33    ,  MJ2    , 0,HD35,
        ,  TP     , M+4,IM+36,            SWITCH 2 OFF
HD35    ,  MJ3    , 0,HD37,
        ,  TP     , M+5,IM+37,            SWITCH 3 OFF
HD37    ,  TP     , T+6,Q,            PICK UP DUMP LIMITS
        ,  TP     , T+7,A,            AND UNISERVO
        ,  LA     , A,12,            TEST TAPE NUMBER
        ,  ZJ     , HD42,HD41,
```

4

FORM #
MSD 8520

```
HD41  , SP  , C+4,12,            OUTPUT ON 5
HD42  , AT  , YA+5,YA+5,         SET REWIND
      , AT  , YA+4,YA+4,         SET WRITE
      , RJ  , YA+3,YA,           TO BLOCK WRITE
      , SP  , Q,21,              PICK UP START
      , LTL , ,IND,
      , SP  , A,15,              PICK UP STOP
      , LTL , ,IND+1,
      , ZJ  , DP6,DP5,
DP5   , TP  , C+10,IND+1,        SET 7777 TO STOP
DP6   , RPV , 3,DP8,             SET BLOK AND
      , TP  , C,IND+2,           BUFFER TEST
DP8   , TU  , HD2,IND+2,         SET IMAGE REF
DP9   , RJ  , BL,BL,             BLANK OUT IMAGE
      , TV  , BL+1,DM3,
DP11  , SP  , IND+1,,
      , ST  , IND,IND+5,         STOP-START
      , SJ  , DM12,DP14,         TEST FOR END
DP14  , TP  , DUM+1,DP34,        SET BASIC
      , TP  , DUM+2,DP36+1,      BUFFER
      , TU  , DUM,DP36,          TRANSFER
      , SP  , IND,15,            TEST OVERLAP
DP18  , TU  , A,DP36+1,
      , TJ  , C+9,DP21,          ENTIRE TRANSFER
      , MJ  , ,DP34-1,
DP21  , SA  , IND+2,,            TEST PARTIAL
      , TU  , A,DP34+1,          TRANSFER
      , TJ  , DUM+3,DP18,
      , SS  , DUM+3,,            COMPUTE OVERLAP
      , TN  , A,A,
      , AT  , C+19,Q,
      , AT  , DP34,DP34,
      , SN  , Q,,
      , AT  , DP36,DP36,
      , SP  , Q,57,
      , SA  , Q,,
      , AT  , DP36+1,DP36+1,
      , IJ  , IND+4,DP40,
DP34  , RPB , ,DP36,             FIRST TRANSFER
      , TP  , ,Z,
DP36  , RPB , 264,DP38,          SECOND TRANSFER
      , TP  , ,Z,
DP38  , TU  , DP55,DP41,
      , TP  , C+7,IND+4,
DP40  , RPB , 6,DP42,            6 WORDS TO W.S.
DP41  , TP  , Z,W,
DP42  , RA  , DP40+1,C+13,       BUMP PICK
      , SP  , C,,                TEST ALL ZEROS
      , RPU , 6,DP47,
DP46  , SA  , W,,
DP47  , ZJ  , DP47+1,DM6,
      , RJ  , KF,KA,             CONVERT
      , TP  , C+4,IND+7,         SET INDEX
      , TU  , HD21,DP53,
      , TV  , DUM+5,DP63,
DP52  , RPB , 2,DP54,
DP53  , TP  , V,Z,               PICK XS-3 WORD
DP54  , RA  , DP53,C+12,         BUMP PICK
DP55  , SP  , Z,,
```

12

FORM # MSD 8520

```
          , LTL   , 12,M+2,          STORE OPERATION
          , SP    , A,24,
          , SA    , Z+1,,
          , LTL   , 6,M+3,          STORE U
          , SP    , A,,
          , LTL   , 30,M+4,          STORE V
          , RPB   , 3,DP64,
DP63      , TP    , M+2,L+2,          WORDS TO IMAGE
DP64      , RA    , DP63,C+3,
          , IJ    , IND+7,DP52,
          , TP    , H+2,L+1,          SET BLANK WORD
          , SP    , V+13,,
          , AT    , C+16,L,          STORE ORIGIN
          , RPB   , 20,DM4,
DM3       , TP    , L,IM,          LINE TO IMAGE
DM4       , RA    , DM3,C+8,          BUMP BLOK STORE
          , RA    , IND+3,C+1,          BUMP LINE STORE
DM6       , RA    , IND,C+5,          BUMP START
          , SP    , IND+3,,
          , TJ    , C+5,DP11,          TEST FULL IMAGE
          , RJ    , YA+3,YA,          WRITE BLOK
          , TP    , C,IND+3,          RESET BLOK TEST
          , MJ    , ,DP9,          TO NEXT 6
DM12      , SP    , IND+3,,
          , ZJ    , DM14,DM15,          TEST IMAGE
DM14      , RJ    , YA+3,YA,          WRITE LAST BLOK
DM15      , SP    , T+4,36,          RESTORE REGISTERS
          , SA    , T+5,,
          , TP    , T+3,Q,
          , MJ1   , ,DM23,          TEST FOR MORE
          , EF    , ,YA+4,
          , RPN   , 120,DM22,          WRITE END OF FILE
          , EWB   , ,C+17,
DM22      , EF    , ,YA+5,          REWIND
DM23      , MJ    , ,RETURN,          EXIT
KA        , TP    , IND,W+6,          ORIGIN TO W.S.
          , TP    , C+5,IND+6,          SET LOOP
          , TU    , DP46,KB,
          , TV    , DUM,KC,
          , TV    , DUM+4,KD,
          , TN    , C+3,A,          PRESET A
          , SA    , C+3,6,
KB        , LQ    , FILL,3,
          , QA    , C+6,A,          NEXT DIGIT IN A
          , LQ    , C+20,1,
          , QJ    , KB-1,KC,
KC        , AT    , C+3,V+1,
KD        , LTL   , ,V,          STORE XS-3 WORD
          , RA    , KB,C+11,          BUMP LOOP
          , RA    , KC,C+2,
          , RA    , KD,C+2,
          , IJ    , IND+6,KB-2,
KF        , MJ    , ,FILL,          EXIT
YA        , EF    , ,YA+4,          WRITE BLOK
          , RPV   , 120,YA+3,
          , EWB   , ,IM,
          , MJ    , ,FILL,          EXIT
          , 00    , 446)B,0,          WRITE ONE BLOK
          , 02    , 200)B,0,          REWIND
```

FORM # MSD 8520

```
DUM   , RPB  , 264,V+1,              DUMP DUMMIES
      , RPB  , 0,DP36,
      , TP   , 0,Z,
      , 00   , DCI+373,0,
      , TP   , 0,V,
      , TP   , ,L+2,
M     , B    , 010101015150,,             SWITCH
      , B    , 010101515001,,             INFORMATION
      , B    , 010151500101,,             PRINT OUT
      , B    , 010101513131,,
      , B    , 010151313101,,
      , B    , 015131310101,,
H     , B    , 012426266747,,             HEADING
      , B    , 674624665154,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 015302543032,,
      , B    , 346566305401,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101474404,,
      , B    , 010147440501,,
      , B    , 014744060101,,
H2    , B    , 010101310401,,
      , B    , 265154300101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101310501,,
      , B    , 265154300101,,
      , B    , 010101010101,,
      , B    , 010101010101,,
      , B    , 010101310601,,
      , B    , 265154300101,,
C     ,      , ,,
      ,      , ,1,
      ,      , ,2,
      ,      , ,3,
      ,      , ,5,
      ,      , ,6,
      ,      , ,7,
      ,      , ,53)B,
      ,      , ,24)B,
      ,      , 1175)B,,
      ,      , ,7777)B,
      ,      , 1,,
      ,      , 2,,
      ,      , 6,,
      ,      , 01010)B,10101)B,
      , 01   , ,,
      , 54   , ,,
      , B    , 606060606060,,
      , 57   , 01010)B,10101)B,
      ,      , 410)B,,
      , 37   , 77377)B,73777)B,
      , 37   , 01010)B,10101)B,
```

14

FORM # MSD 8520

```
BL      ,  RPV     , 120,FILL,           BLANK OUT
        ,  TP      , H+2,IM,            IMAGE
IND     ,  RESERV, 8,8,
W       ,  RESERV, 7,7,
V       ,  RESERV, 14,14,
L       ,  RESERV, 20,20,
IM      ,  RESERV, 120,120,
Z       ,  RESERV, 264,264,
        ,  END     , ,,
```

1. IDENTIFICATION

RRF3, SIN X STATED POINT
A. E. Roberts, M. D. Bernick - REVISED 1 June 195?
Remington Rand Univac

2. PURPOSE

Given X, computes $Y(X) = \sin X$

3. METHOD

a. Accuracy: $|Y(X) - \sin X| \leq 2^{-30}$

b. Range of Argument: $|X| < \pi \cdot 2^{36}$ radians

c. Scaling: $X \cdot 2^{32}$, $Y(X) \cdot 2^{32}$

d. Derivation: obtained from the relation $\sin(\pi/2)Z = 1 - 2\sin^2 \frac{\pi}{4}(Z - 1)$

Polynomial approximation for $\frac{\sin(\pi/4)N}{N}$ as a function of $(-N)^2$ for $-1 \leq N \leq 1$ is derived from the Chebyshev expansion:

$$\sin \pi/4\, N = 2 \sum_{k=1}^{\infty} (-1)^k \; J_{2k-1}(\pi/4) \cdot T_{2k-1}(N).$$

See Central Exchange Routine RR-24

The argument, X, is entered and multiplied by $2/\pi$ so that $(2/\pi)X = Z$. The routine then computes $\sin(\pi/2)Z = \sin X$.

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t + 2	t
r+1	Normal	Return	

b. Control and Results

The most significant 36 bits of the argument, X, must be initially stored in cell t + 4, and the least significant 36 bits of X stored in t + 5. The function Y(X) will be found in t + 3, and in the accumulator at the completion of the routine.

c. Space Required

54 cells of instructions and constants

1 cell of working storage

d. Error Codes

None

5. RESTRICTIONS

The argument, X, must be scaled 2^{32} and within the stated range.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
2VPI	24 27630 15554	$(2/\pi) \cdot 2^{35}$
MASK	37 77777 77774	Modulus
N1	40 00000 00000	$1 \cdot 2^{35}$
N2	20 00000 00000	$1 \cdot 2^{34}$
N3	04 00000 00000	$1 \cdot 2^{32}$
UAD	00 00001 00000	U-Address Advance
A4	00 00122 65046	Coefficients in
A4+1	00 04626 21024	Polynomial
A4+2	01 21465 66440	Expansion of
A4+3	12 25357 16221	$\sin(\pi/4 \; N)/N$
A4+4	31 10375 52420	
2PI	31 10375 52421	$2\pi \cdot 2^{32}$

b. Working Storage

1 cell labeled STORE

c. Timing

Maximum 4.26 mls.

REMINGTON RAND UNIVAC

PROBLEM RRF3

CODED BY Roberts, Bernick DATE REV. June 1957

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, SUB	, RRF3	, 54	,	$
,		, TEMPS	, 1	, 0	,	$
,		, INOUT	, 2	, 1	,	$
,	ENTRY	, MJ	, 0	, START	,	$
,	ERROR	, ALARM	,	,	, NOT USED	
,	EXIT	, MJ	, 0	, FILL	,	$
,	Y	, 00	, FILL	, FILL	, FUNCTION Y	$
,	XM	, 00	, FILL	, FILL	, ARGUMENT X(MSP)	$
,	XL	, 00	, FILL	, FILL	, ARGUMENT X(LSP)	$
,	START	, SP	, XM	, 36	, X TO A	$
,		, SA	, XL	, 0	,	$
,		, DV	, 2PI	, Q	, FCTR OUT 2 PI	$
,		, MP	, A	, 2VPI	, FORM Z SCLD 34	$
,		, LTL	, 3	, Q	, IN Q	$
,		, TP	, NEGT	, NEGT + 2	, FUNCTION PLUS	$
,		, QJ	, MINUS	, PLUS	, TEST SIGN OF Y	$
,	MINUS	, TP	, RELC 1	, NEGT + 2	, FUNCTION MINUS	$
,	PLUS	, QT	, MASK	, A	, W = Z MOD 1 SCLD 35	$
,		, QJ	, EVEN	, ODD	, TEST QUAD NUMBER	$
,	ODD	, SS	, N1	, 0	, FORM W-1 IN R	$
,	EVEN	, MP	, A	, Q	, W - 1 SQRD IN A	$
,		, SA	, N1	, 0	, ROUND SCALE 34	$
,		, LTL	, 0	, A	, IN A = N	$

PROBLEM RRF3

CODED BY Roberts, Bernick DATE REV. 6-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,	,	TN	, A	, STORE	, -N TO STORE	$
,	,	TU	, RELC3	, NEST+3	, FORM SINE OF	$
,	,	TP	, A4	, Q	, PI(W-1) OVER	$
,	NEST ,	MP	, Q	, STORE	, 4(W-1)	$
,	,	SA	, N1	, 0	, SCLD 35	$
,	,	LTL	, 0	, A	, IN A	$
,	,	AT	, FILL	, Q	, = S	$
,	,	RA	, NEST+3	, UAD	,	$
,	,	TJ	, RELC2	, NEST	, TEST END POINT	$
,	,	MP	, Q	, Q	, SQUARE S	$
,	,	SA	, N1	, 0	, ROUND SCALE 34	$
,	,	LTL	, 0	, A	, IN A	$
,	,	MP	, A	, STORE	, ROUND SCALE 32	$
,	,	SS	, N2	, 37	, 2S SQRD TIMES(-N)	$
,	NEGT ,	TN	, A	, A	, MINUS 1	$
,	,	ST	, N3	, A	, IN A	$
,	,	00	, FILL	, FILL	, -A TO A FOR Y POS	$
,	,	TP	, A	, Y	, STORE FUNCTION	$
,	,	MJ	, 0	, EXIT	, TO EXIT	$
,	RELC1 ,	MJ	, 0	, NEGT+3	, RELATIVE	$
,	RELC2 ,	AT	, A4+5	, Q	, CONSTANTS FOR	$
,	RELC3 ,	00	, A4+1	, 00000	, TEST AND PRESET	$

PROBLEM RRF3

CODED BY Roberts, Bernick DATE REV. 6-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,	2VPI	, B24	, 27630	, 15554	, 2 OVER PI SCLD 35	$
,	MASK	, B37	, 77777	, 77774	, MODULUS	$
,	N1	, B40	, 00000	, 00000	, 1 SCLD 35	$
,	N2	, B20	, 00000	, 00000	, 1 SCLD 34	$
,	N3	, B04	, 00000	, 00000	, 1 SCLD 32	$
,	UAD	, B	, 1	, 00000	, U ADVANCE	$
,	A4	, B	, 122	, 65046	, CONSTANTS	$
,		, B	, 4626	, 21024	, FOR	$
,		, B01	, 21465	, 66440	, POLYNOMIAL	$
,		, B12	, 25357	, 16221	, EXPANSION	$
,		, B31	, 10375	, 52420	,	$
,	2PI	, B31	, 10375	, 52421	, 2PI SCLD 32	$
,		, ENDSUB	,	,	,	$

1. IDENTIFICATION

RRF9, SQUARE ROOT STATED POINT
A. E. Roberts, Jr., M. D. Bernick
REVISED JUNE 20, 1957
Remington Rand Univac

2. PURPOSE

Given X, compute $Y(X) = \sqrt{X}$

3. METHOD

a. Accuracy: 33 significant bits

b. Range of Argument: $0 \leq X \cdot 2^{33} < 2^{71}$

c. Scaling: $X \cdot 2^{33}$, $Y(X) \cdot 2^{33}$

d. Derivation: for $1 \leq n \leq 2$, then for suitable, a, b, c and d $y = a(n+d) + b - \frac{c}{n+d}$ is an approximation of $\sqrt{n}$. One application of the Newton-Raphson Formula (with proper round-off bias, as error of Formula is always positive) then gives $\sqrt{n}$ with essentially 33 bit accuracy.

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t + 2	t
r + 1	Normal	Return	

b. Control and Results

The 36 most significant bits of the argument, X, must be initially stored at t + 5, and the 36 least significant bits of X at t + 6. The 36 most significant and least significant bits of the function, Y, will be found at t + 3 and t + 4 respectively upon completion of the routine. The 72 bits of the function, Y, will also be found in the Accumulator upon completion of the routine.

c. Space Required

55 cells of instructions and constants

4 cells of working space

d. Error Codes

The following error code is left in the accumulator on return through the error exit:

CODE	EXPLANATION
RRF9 $\cdot 2^{12}$	$X < 0$

5. RESTRICTIONS

The argument, X, must be scaled 2^{33} and must be non-negative.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
C	26 47670 31361	D=24,291,062,513
C1	00 00000 65324	A=27,348
C2	11 45346 44516	B=10,291,988,814
C3	33 06571 40273	C=29,104,062,651
C4	00 00000 00023	19
C5	26 50117 14640	$\sqrt{2} \cdot 2^{34}$ (Rounded)
C6	00 00000 00077	Shift Count Mask
C7	20 00000 00000	$1 \cdot 2^{34}$
C8	37 77777 77777	Round Factor
CODE	54 54311 40000	Error Code for $X < 0$

b. Working Space

4 cells labeled STORE thru STORE+3

c. Timing

Approximate maximum of 2.92 mls for $X = 1 \cdot 2^{-33}$

PROBLEM RRF9

CODED BY Roberts, Bernick DATE REV. 6-20-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		SUB	RRF9	55		$
,		TEMPS	4	0		$
,		INOUT	2	2		$
,	ENTRY	MJ	0	START		$
,	ERROR	ALARM				$
,	EXIT	MJ	0	FILL		$
,	YM	00	FILL	FILL	FUNCTION	$
,	YL	00	FILL	FILL	SCLD 33	$
,	XM	00	FILL	FILL	ARGUMENT	$
,	XL	00	FILL	FILL	SCLD 33	$
,	START	SP	XM	36	ARGUMENT	$
,		SA	XL	0	TO A	$
,		SJ	NEG	NONEG	TEST SIGN OF X	$
,	NEG	SP	CODE	0	X NEG	$
,		MJ	0	ERROR	GO TO ALARM	$
,	NONEG	ZJ	NOZERO	KODD+3	TEST X = Y = 0	$
,	NOZERO	SF	A	STORE	K TO V OF STORE	$
,		SP	A	0	(A) = N	$
,		TP	A	STORE+1	N TO STORE	$
,		SA	C	18	N+D SCLD -18 = F	$
,		LTL	0	STORE+2	STORE F	$
,		MP	C1	STORE+2	FORM AF	$
,		AT	C2	STORE+3	STORE AF+B	$
,		SN	C3	15	-C SCLD 15 TO A	$
,		DV	STORE+2	A	FORM -G=C OVR F	$
,		AT	STORE+3	STORE+2	Y(1)=SQRT N SCLD 16	$

PROBLEM RRF9

CODED BY Roberts, Bernick DATE REV.6-20-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
,		, SP	, STORE+1	, 32	, N SCLD 32 TO A $
,		, SS	, STORE+2	, 0	, N SCLD 32 -Y(1) $
,		, DV	, STORE+2	, A	, (N OVR Y(1))+Y(1)-1=Y(2) $
,		, AT	, STORE+2	, STORE+2	, Y(2)=SQRT N SCLD 17 $
,		, LQ	, STORE	, 35	, K(0) TO Q(35) $
,		, QT	, C6	, A	, (K-K(0)) OVR 2 TO A $
,		, TV	, A	, KODD	, SET SHIFT=(K-1) OVR 2 $
,		, TP	, INSTC	, KODD+1	, SET FOR K LESS OR=37 $
,		, TJ	, C4	, KTEST	, IS 19 GRTR (K-K(0)) OVR 2
,		, TP	, INSTC+1	, KODD+1	, SET FOR K GRTR 37 $
,	KTEST	, QJ	, KODD	, KEVEN	, TEST PARITY OF K $
,	KEVEN	, MP	, C5	, STORE+2	, FORM (SQRT 2 TIMES Y(2) $
,		, SA	, C7	, 1	, +1) OVR 2 FOR $
,		, LTL	, 0	, STORE+2	, K EVEN IN A $
,	KODD	, SP	, STORE+2	, FILL	, SQRT(N SCLD (K-1)) = Y $
,		, 00	, FILL	, FILL	, IF K GRTR 37 SCL -36 $
,		, LTL	, 0	, A	, CLEAR A(L) TO 0 $
,		, LTL	, 0	, YM	, STORE $
,		, LTR	, 0	, YL	, Y SCLD 33 $
,		, MJ	, 0	, EXIT	, EXIT $
,	INSTC	, MJ	, 0	, KODD+3	, INSTRUCTIONAL $
,		, SA	, C8	, 0	, CONSTANTS $
,	C	, B26	, 47670	, 31361	, D=24 291 062 513 $
,	C1	, B	,	, 65324	, A=27 348 $
,	C2	, B11	, 45346	, 44516	, B=10 291 988 814 $

PROBLEM RRF9

CODED BY Roberts, Bernick DATE REV. 6-20-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
	, C3	, B33	, 06571	, 40273	, C=29 104 062 651	$
	, C4	, B	,	, 23	, DECIMAL 19	$
	, C5	, B26	, 50117	, 14640	, SQRT 2 SCLD 34	$
	, C6	, B	,	, 77	, SHIFT COUNT MASK	$
	, C7	, B20	, 00000	, 00000	, 1 SCLD 34	$
	, C8	, B37	, 77777	, 77777	, ROUND FACTOR	$
	, CODE	, B54	, 54311	, 40000	, ERROR CODE	$
	,	, ENDSUB	,	,	,	$

1. IDENTIFICATION RRF6, ARCCOS X STATED POINT

A. Franck, R. Van Hilst 15 June 1957

Remington Rand Univac

2. PURPOSE

Given X, compute Y(X) = arccos X in radians

3. METHOD

a. Accuracy: $|Y(X) - \arccos X| \leq 2^{-26}$

b. Range of Argument: $|X| \leq 1$

c. Scaling: $X \cdot 2^{33}$, $Y(X) \cdot 2^{33}$

d. Derivation: The function arcsin X is computed in radians from the polynomial approximation $\frac{\pi}{2} - \sqrt{1 - X}\ X(X)$ where $X = \sum_{i=o}^{7} a_i x^i$ (See Rand Sheet #39.), then arccos $X = -\arcsin X + \frac{\pi}{2}$.

The square root is obtained as follows: for $1 \leq N \leq 2$. Then for suitable A, B, C, D: Y(1) = A(N+D)+B-C/(N+D) is an approximation to $\sqrt{N}$ with relative error not in excess of .0000172; one application of the Newton Raphson Formula gives Y(2). (See RRF9, SQUARE ROOT STATED POINT.)

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	VADDR
r	RJ	t + 2	t
r+1	Normal	Return	

b. Control and Results

The argument, X, must be initially stored at t+4; the result, Y(X), will be found at t+3 upon successful completion of the routine.

c. Space Required

98 cells of instructions and constants

7 cells of working space.

d. Error Codes

The following error code is left in the Accumulator on return through the error exit:

CODE	EXPLANATION
RRF6 $\cdot 2^{12}$	$\|X\|>1$

5. RESTRICTIONS

The argument must be within the stated range and scaled 2^{33}.

6. CODING INFORMATION

a. Constants

LOC	CONSTANTS	EXPLANATION
C	10 00000 00000	$1 \cdot 2^{33}$
C1	00 00000 00000	Zero
C2	00 00000 00001	One
C3	53 24135 20070	$a_7 \cdot 2^{44}$
C4	33 24414 25535	$a_6 \cdot 2^{42}$
C5	56 40071 51545	$a_5 \cdot 2^{40}$
C6	37 50417 41234	$a_4 \cdot 2^{40}$
C7	46 23706 66522	$a_3 \cdot 2^{39}$
C8	26 61651 66073	$a_2 \cdot 2^{38}$
C9	44 42003 30653	$a_1 \cdot 2^{37}$
C10	31 10375 51633	$a_0 \cdot 2^{34}$
C11	37 77777 77777	Round Constant
C12	14 44176 65211	$\frac{\pi}{2} \cdot 2^{33}$
C13	10 00000 00001	$1 \cdot 2^{33} + 1$
C14	26 47670 31361	D = 24,291,062,513
C15	00 00000 65324	A = 27,348
C16	11 45346 44516	B = 10,291,988,814
C17	33 06571 40273	C = 29,104,062,651
C18	00 00000 00077	Shift Count Mask
C19	26 50117 14640	$\sqrt{2} \circ 2^{34}$ (Rounded)
C20	20 00000 00000	$1 \circ 2^{34}$
CODE	54 54311 10000	Error Code

b. Work Space

7 cells labeled STORE thru STORE+6

c. Timing

Approximate maximum of 6.75 mls. for $X = -(1-1 \cdot 2^{-33})$.

PROBLEM RRF6

CODED BY Franck, VanHilst DATE 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, SUB	, RRF6	, 98	,	$
,		, TEMPS	, 7	, 0	,	$
,		, INOUT	, 1	, 1	,	$
,	ENTRY	, MJ	, 0	, START	,	$
,	ERROR	, ALARM	,	,	,	$
,	EXIT	, MJ	, 0	, FILL	,	$
,	Y	, 00	, FILL	, FILL	, FUNCTION SCLD 33	$
,	X	, 00	, FILL	, FILL	, ARGUMENT SCLD 33	$
,	START	, TM	, X	, A	, ABS X TO A	$
,		, TJ	, C13	, PROG	, CHCK FOR X GRTR 1	$
,		, SP	, CODE	, 0	, ERROR CODE TO A	$
,		, MJ	, 0	, ERROR	, GO TO ERROR EXIT	$
,	PROG	, TP	, C12	, Y	, Y = PI ovr 2	$
,		, ZJ	, NOZERO	, EXIT	, FOR X=0	$
,	NOZERO	, TP	, A	, STORE	, STORE ABS X	$
,		, TP	, X	, A	, X TO A	$
,		, TP	, C1	, STORE+1	, SET FOR POS X	$
,		, SJ	, MINUS	, PLUS	, TEST SIGN OF X	$
,	MINUS	, TP	, C2	, STORE+1	, SET FOR NEG X	$
,		, TM	, A	, A	, ABS X TO A	$
,	PLUS	, EJ	, C	, XONE	, TEST X = 1	$
,		, MP	, STORE	, C3	, EVALUATE	$
,		, LTL	, 1	, A	, POLYNOMIAL	$
,		, AT	, C4	, STORE+2	, EXPRESSION	$

PROBLEM RRF6

CODED BY Franck, VanHilst DATE 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
,	,	MP	, STORE	, STORE+2	, IN $
,	,	LTL	, 1	, A	, APPROXIMATION $
,	,	AT	, C5	, STORE+2	, OF ARCSIN X $
,	,	MP	, STORE	, STORE+2	, ACCUMULATE $
,	,	LTL	, 3	, A	, IN $
,	,	AT	, C6	, STORE+2	, STORE+2 SCLD 34 $
,	,	MP	, STORE	, STORE+2	, $
,	,	LTL	, 2	, A	, $
,	,	AT	, C7	, STORE+2	, $
,	,	MP	, STORE	, STORE+2	, $
,	,	LTL	, 2	, A	, $
,	,	AT	, C8	, STORE+2	, $
,	,	MP	, STORE	, STORE+2	, $
,	,	LTL	, 2	, A	, $
,	,	AT	, C9	, STORE+2	, $
,	,	MP	, STORE	, STORE+2	, $
,	,	LTL	, 0	, A	, $
,	,	AT	, C10	, STORE+2	, X(X) TO STORE+2 $
,	,	TN	, STORE	, A	, FORM (1-X) SCLD 33$
,	,	SA	, C	, 0	, IN A $
,	,	SF	, A	, STORE+3	, K TO STORE+3 $
,	,	SP	, A	, 0	, (R) = N SCLD 34 $
,	,	TP	, A	, STORE+4	, STORE N $
,	,	SA	, C14	, 18	, FORM AND STORE $
,	,	LTL	, 0	, STORE+5	, (N+D) SCLD -18=F $

PROBLEM RRF6

CODED BY Franck.VanHilst DATE 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
,		, MP	, C15	, STORE+5	, A x F TO STORE $
,		, AT	, C16	, STORE+6	, A x F+B TO STORE $
,		, SN	, C17	, 15	, -C SCLD 15 TO A $
,		, DV	, STORE+5	, A	, -C OVER F = -G $
,		, AT	, STORE+6	, STORE+5	, Y(1) APPRX SQRT N SCLD 16 $
,		, SP	, STORE+4	, 32	, N SCLD 32 TO A $
,		, SS	, STORE+5	, 0	, N - Y (1) $
,		, DV	, STORE+5	, A	, (N ovr Y(1)) -1+Y(1) $
,		, AT	, STORE+5	, STORE+5	, =Y(2)=SQRT N SCLD 17 $
,		, LQ	, STORE+3	, 35	, K(0) IN Q(35) $
,		, QT	, C18	, A	, K-K(0) OVR 2 IN A $
,		, TV	, A	, KODD	, SET SCALE FOR SQRT N $
,		, QJ	, KODD	, KEVEN	, TEST PARITY OF K $
,	KEVEN	, MP	, C19	, STORE+5	, (Y(2)+1) ovr 2 $
,		, SA	, C20	, 1	, SCLD 17 EQLS $
,		, LTL	, 0	, STORE+5	, SQRT (N SCLD K-1) $
,	KODD	, SP	, STORE+5	, FILL	, IN A $
,		, SA	, C11	, 0	, RND SCL SQRT (1-X) $
,		, LTL	, 0	, A	, SCLD 33 IN A $
,		, MP	, A	, STORE+2	, FORM - ARCSIN X $
,		, LTL	, 2	, A	, SCLD 33 $
,		, ST	, C12	, Q	, IN Q $

PROBLEM RRF6

CODED BY Franck, Van Hilst DATE 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
	, TEST	, IJ	, STORE+1	, NEGX	, FOR X NEG $
	,	, TN	, Q	, Q	, Y IN 2ND QUAD $
	, NEGX	, RS	, Q	, C12	,-ARCOS X EQUALS $
	,	, TN	, A	, A	, ARCSINX - PI OVR 2$
	,	, TP	, A	, Y	, STORE FUNCTION $
	,	, MJ	, 0	, EXIT	, SCLD 33 $
	, XONE	, TN	, C12	, Q	, FOR X = 1 $
	,	, MJ	, 0	, TEST	, $
	, C	, B10	, 00000	, 00000	, 1 SCLD 33 $
	, C1	, B	,	, 0	, ZERO $
	, C2	, B	,	, 1	, ONE $
	, C3	, B53	, 24135	, 20070	, A(7) SCLD 44 $
	, C4	, B33	, 24414	, 25535	, A(6) SCLD 42 $
	, C5	, B56	, 40071	, 51545	, A(5) SCLD 40 $
	, C6	, B37	, 50417	, 41234	, A(4) SCLD 40 $
	, C7	, B46	, 23706	, 66522	, A(3) SCLD 39 $
	, C8	, B26	, 61651	, 66073	, A(2) SCLD 38 $
	, C9	, B44	, 42003	, 30653	, A(1) SCLD 37 $
	, C10	, B31	, 10375	, 51633	, A(0) SCLD 34 $
	, C11	, B37	, 77777	, 77777	, ROUND CONSTANT $
	, C12	, B14	, 44176	, 65211	, PI over 2 SCLD 33 $
	, C13	, B10	, 00000	, 00001	, LIMIT ON X $
	, C14	, B26	, 47670	, 31361	, D=24 291 062 513 $

PROBLEM RRF6

CODED BY Franck, VanHilst DATE 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
	, C15	, B	,	, 65324	, A = 27 348 $
	, C16	, B11	, 45346	, 44516	, B = 10 291 988 814$
	, C17	, B33	, 06571	, 40273	, C = 29 104 062 651 $
	, C18	, B	,	, 77	, K MASK $
	, C19	, B26	, 50117	, 14640	, SQRT 2 RND SCLD 34$
	, C20	, B20	, 00000	, 00000	, 1 SCLD 34 $
	, CODE	, B54	, 54311	, 10000	, ERROR CODE $
	,	, ENDSUB	,	,	, $

1. IDENTIFICATION

RRF1, COS X STATED POINT
A. E. Roberts, R. Van Hilst, 1 June 1957
Remington Rand Univac

2. PURPOSE

Given X, computes $Y(X) = \sin (X + \frac{\pi}{2}) = \cos X$

3. METHOD

a. Accuracy: $|Y(X) - \cos X| \leq 2^{-30}$

b. Range of Argument: $|X + \frac{\pi}{2}| < \pi \cdot 2^{36}$ radians

c. Scaling: $X \cdot 2^{32}$, $Y(X) \cdot 2^{32}$

d. Derivation: obtained from the relation $\sin \frac{\pi}{2} Z = 1 - 2 \sin^2 \frac{\pi}{4} (Z - 1)$ and a polynomial expansion for $\frac{\sin \frac{\pi}{4} N}{N}$. [See RRF3 - Sin X]

The argument X is incremented by $\frac{\pi}{2}$; then $2/\pi \ (X + \pi/2) = Z$ is formed and the routine computes $\sin(\pi/2) Z = \cos X$.

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t + 2	t
r + 1	Normal	Return	

b. Control and Results

The most significant 36 bits of the argument, X, must be initially stored in cells t + 4, and the least significant 36 bits of X stored in t + 5. The function Y(X) will be found in t +3, and in the accumulator at the completion of the routine.

c. Space Required

56 cells of instructions and constants

1 cell of working storage

d. Error Codes

None

5. RESTRICTIONS

The argument, X, must be scaled 2^{32} and within the stated range.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
PIV2	06 22077 32504	$\pi/2 \cdot 2^{32}$
2VPI	24 27630 15554	$2/\pi \cdot 2^{35}$
MASK	37 77777 77774	Modulus
N1	40 00000 00000	$1 \cdot 2^{35}$
N2	20 00000 00000	$1 \cdot 2^{34}$
N3	04 00000 00000	$1 \cdot 2^{32}$
UAD	00 00001 00000	U-Address Advance
A4	00 00122 65046	Coefficients in
A4+1	00 04626 21024	Polynomial
A4+2	01 21465 66440	Expansion of
A4+3	12 25357 16221	sin(π/4 N)/N
A4+4	31 10375 52420	
2PI	31 10375 52421	$2\pi \cdot 2^{32}$

b. Working Storage

1 cell labeled STORE

c. Timing

Maximum 4.30 mls.

PROBLEM RRF1

CODED BY Roberts, Van Hilst DATE 6/1/57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, SUB	, RRF1	, 56	,	$
,		, TEMPS	, 1	, 0	,	$
,		, INOUT	, 2	, 1	,	$
,	ENTRY	, MJ	, 0	, START	,	$
,	ERROR	, ALARM	,	,	, NOT USED	$
,	EXIT	, MJ	, 0	, FILL	,	$
,	Y	, 00	, FILL	, FILL	, FUNCTION Y	$
,	XM	, 00	, FILL	, FILL	, ARGUMENT X(MSP)	$
,	XL	, 00	, FILL	, FILL	, ARGUMENT X(LSP)	$
,	START	, SP	, XM	, 36	, X TO A	$
,		, SA	, XL	, 0	,	$
,		, AT	, PIV2	, A	, ADD PI OVER 2	$
,		, DV	, 2PI	, Q	, FCTR OUT 2 PI	$
,		, MP	, A	, 2VPI	, FORM Z SCLD 34	$
,		, LTL	, 3	, Q	, IN Q	$
,		, TP	, NEGT	, NEGT + 2	, FUNCTION PLUS	$
,		, QJ	, MINUS	, PLUS	, TEST SIGN OF Y	$
,	MINUS	, TP	, RELC1	, NEGT + 2	, FUNCTION MINUS	$
,	PLUS	, QT	, MASK	, A	, W+Z MOD 1 SCLD 35	$
,		, QJ	, EVEN	, ODD	, TEST QUAD NUMBER	$
,	ODD	, SS	, N1	, 0	, FORM W-1 IN R	$
,	EVEN	, MP	, A	, Q	, W-1 SQRD IN A	$
,		, SA	, N1	, 0	, ROUND SCALE 34	$
,		, LTL	, 0	, A	, IN A = N	$

PROBLEM RRF1

CODED BY Roberts, Van Hilst DATE 6/1/57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, TN	, A	, STORE	, - N TO STORE	$
,		, TU	, RELC 3	, NEST + 3	, FORM SINE OF	$
,		, TP	, A4	, Q	, PI(W-1) OVER	$
,	NEST	, MP	, Q	, STORE	, 4(W-1) SCLD	$
,		, SA	, N1	, 0	, 35 IN A	$
,		, LTL	, 0	, A	, = S	$
,		, AT	, FILL	, Q	,	$
,		, RA	, NEST + 3	, UAD	,	$
,		, TJ	, RELC 2	, NEST	, TEST END POINT	$
,		, MP	, Q	, Q	, SQUARE S	$
,		, SA	, N1	, 0	, ROUND SCALE 34	$
,		, LTL	, 0	, A	, IN A	$
,		, MP	, A	, STORE	, ROUND AND SCL 32	$
,		, SS	, N2	, 37	, 2S SQRD TIMES (-N)	$
,	NEGT	, TN	, A	, A	, MINUS 1	$
,		, ST	, N3	, A	, IN A	$
,		, 00	, FILL	, FILL	, -A TO A FOR Y POS	$
,		, TP	, A	, Y	, STORE FUNCTION	$
,		, MJ	, 0	, EXIT	, TO EXIT	$
,	RELC 1	, MJ	, 0	, NEGT + 3	, RELATIVE	$
,	RELC 2	, AT	, A4 + 5	, Q	, CONSTANTS FOR	$
,	RELC 3	, 00	, A4 + 1	, 00000	, TEST AND PRESET	$
,	PIV2	, B06	, 22077	, 32504	, PI OVER 2 SCLD 32	$
,	2VPI	, B24	, 27630	, 15554	, 2 OVER PI SCLD 35	$
,	MASK	, B37	, 77777	, 77774	, MODULUS	$

PROBLEM RRF1

CODED BY Roberts, Van Hilst DATE 6/1/57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,	N1	, B40	, 00000	, 00000	, 1 SCLD 35	$
,	N2	, B20	, 00000	, 00000	, 1 SCLD 34	$
,	N3	, B04	, 00000	, 00000	, 1 SCLD 32	$
,	UAD	, B	, 1	, 00000	, U ADVANCE	$
,	A4	, B	, 122	, 65046	, CONSTANTS	$
,		, B	, 4626	, 21024	, FOR	$
,		, B01	, 21465	, 66440	, POLYNOMIAL	$
,		, B12	, 25357	, 16221	, EXPANSION	$
,		, B31	, 10375	, 52420	,	$
,	2PI	, B31	, 10375	, 52421	, 2PI SCLD 32	$
,		, ENDSUB	,	,	,	$

1. IDENTIFICATION

RRF5, ARCSIN X STATED POINT
A. Franck, M. D. Bernick, REVISED 15 June 1957
Remington Rand Univac

2. PURPOSE

Given X, compute Y(X) = arcsin X in radians.

3. METHOD

a. Accuracy: $|Y(X) - \arcsin X| \leq 2^{-26}$

b. Range of Argument: $|X| \leq 1$

c. Scaling: $X \cdot 2^{33}$, $Y(X) \cdot 2^{33}$

d. Derivation: Y(X) is computed in radians from the polynomial approximation $Y(X) \approx \frac{\pi}{2} - \sqrt{1 - X}\ \chi(X)$, where $\chi = \sum_{i=0}^{7} a_i X^i$. (See Rand Sheet #39.) The square root is obtained as follows: for $1 \leq N \leq 2$, then for suitable A, B, C, D: $Y(1) = A(N+D)+B-C/(N+D)$ is an approximation to $\sqrt{N}$ with relative error not in excess of .0000172; one application of the Newton Raphson Formula gives Y(2). (See RRF9, SQUARE ROOT STATED POINT.)

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t+2	t
r+1	Normal	Return	

b. Control and Results

The argument, X, must be initially stored at t+4; the result, Y(X), will be found at t+3 upon successful completion of the routine.

c. Space Required

96 cells of instructions and constants

7 cells of working space

d. Error Codes

The following error code is left in the Accumulator on return through the error exit:

CODE	EXPLANATION
$RRF5 \cdot 2^{12}$	$|X|>1$

5. RESTRICTIONS

The argument must be within the stated range and scaled 2^{33}.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
C	10 00000 00000	$1 \cdot 2^{33}$
C1	00 00000 00000	Zero
C2	00 00000 00001	$1 \cdot 2^{0}$
C3	53 24135 20070	$a_7 \cdot 2^{44}$
C4	33 24414 25535	$a_6 \cdot 2^{42}$
C5	56 40071 51545	$a_5 \cdot 2^{40}$
C6	37 50417 41234	$a_4 \cdot 2^{40}$
C7	46 23706 66522	$a_3 \cdot 2^{39}$
C8	26 61651 66073	$a_2 \cdot 2^{38}$
C9	44 42003 30653	$a_1 \cdot 2^{37}$
C10	31 10375 51633	$a_0 \cdot 2^{34}$
C11	37 77777 77777	Round Constant
C12	14 44176 65211	$\frac{\pi}{2} \cdot 2^{33}$
C13	10 00000 00001	$1 \cdot 2^{33} + 1$
C14	26 47670 31361	D=24,291,062,513
C15	00 00000 65324	A=27,348
C16	11 45346 44516	B=10,291,988,814
C17	33 06571 40273	C=29,104,062,651
C18	00 00000 00077	Shift Count Mask
C19	26 50117 14640	$\sqrt{2} \cdot 2^{34}$ (rounded)
C20	20 00000 00000	$1 \cdot 2^{34}$
CODE	54 54311 00000	Error Code

b. Work Space

7 cells labeled STORE thru STORE + 6

c. Timing

Approximate maximum of 6.66 mls. for $X = -(1 - 1 \cdot 2^{-33})$.

PROBLEM RRF5

CODED BY Franck, Bernick DATE REV. 6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
	,	, SUB	, RRF5	, 96	,	$
	,	, TEMPS	, 7	, 0	,	$
	,	, INOUT	, 1	, 1	,	$
	, ENTRY	, MJ	, 0	, START	,	$
	, ERROR	, ALARM	,	,	,	$
	, EXIT	, MJ	, 0	, FILL	,	$
	, Y	, 00	, FILL	, FILL	, FUNCTION SCLD 33	$
	, X	, 00	, FILL	, FILL	, ARGUMENT SCLD 33	$
	, START	, TM	, X	, A	, ABS X TO A	$
	,	, TJ	, C13	, PROG	, CHK FOR X GRTR 1	$
	,	, SP	, CODE	, 0	, ERROR CODE TO A	$
	,	, MJ	, 0	, ERROR	, GO TO ERROR EXIT	$
	, PROG	, TP	, A	, Y	, X=Y IF X=0	$
	,	, ZJ	, NOZERO	, EXIT	, TEST X = 0	$
	, NOZERO	, TP	, A	, STORE	, ABS X TO STORE	$
	,	, TP	, X	, A	, X TO A	$
	,	, TP	, C1	, STORE+1	, SET FOR POS Y	$
	,	, SJ	, MINUS	, PLUS	, TEST SIGN OF X	$
	, MINUS	, TP	, C2	, STORE+1	, SET FOR NEG Y	$
	,	, TM	, A	, A	, ABS X TO A	$
	, PLUS	, EJ	, C	, XONE	, TEST X = 1	$
	,	, MP	, STORE	, C3	, EVALUATE	$
	,	, LTL	, 1	, A	, POLYNOMIAL	$
	,	, AT	, C4	, STORE+2	, EXPRESSION	$
	,	, MP	, STORE	, STORE+2	, IN	$

ITEM NUMBER	LOC	OP	U	V	COMMENTS
,	,	LTL ,	1 ,	A ,	APPROXIMATION $
,	,	AT ,	C5 ,	STORE+2 ,	OF Y $
,	,	MP ,	STORE ,	STORE+2 ,	ACCUMULATE $
,	,	LTL ,	3 ,	A ,	IN $
,	,	AT ,	C6 ,	STORE+2 ,	STORE+2 SCLD 34 $
,	,	MP ,	STORE ,	STORE+2 ,	$
,	,	LTL ,	2 ,	A ,	$
,	,	AT ,	C7 ,	STORE+2 ,	$
,	,	MP ,	STORE ,	STORE+2 ,	$
,	,	LTL ,	2 ,	A ,	$
,	,	AT ,	C8 ,	STORE+2 ,	$
,	,	MP ,	STORE ,	STORE+2 ,	$
,	,	LTL ,	2 ,	A ,	$
,	,	AT ,	C9 ,	STORE+2 ,	$
,	,	MP ,	STORE ,	STORE+2 ,	$
,	,	LTL ,	0 ,	A ,	$
,	,	AT ,	C10 ,	STORE+2 ,	X(X) TO STORE+2 $
,	,	TN ,	STORE ,	A ,	FORM (1-X) SCLD 33$
,	,	SA ,	C ,	0 ,	IN A $
,	,	SF ,	A ,	STORE+3 ,	K TO STORE+3 $
,	,	SP ,	A ,	0 ,	(R) = N SCLD 34 $
,	,	TP ,	A ,	STORE+4 ,	STORE N $
,	,	SA ,	C14 ,	18 ,	FORM AND STORE $
,	,	LTL ,	0 ,	STORE+5 ,	(N+D) SCLD -18=F $
,	,	MP ,	C15 ,	STORE+5 ,	A X F TO STORE $

PROBLEM RRF5

CODED BY Franck.Bernick DATE REV.6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS
,		, AT	, C16	, STORE+6	, AxF+B TO STORE $
,		, SN	, C17	, 15	, -C SCLD 15 TO A $
,		, DV	, STORE+5	, A	, -C OVER F = -Q $
,		, AT	, STORE+6	, STORE+5	, Y(1) APRX SQRTN SLD 16 $
,		, SP	, STORE+4	, 32	, N SCLD 32 TO A $
,		, SS	, STORE+5	, 0	, N - Y(1) $
,		, DV	, STORE+5	, A	, (N ovr Y(1))-1+Y(1)$
,		, AT	, STORE+5	, STORE+5	, =Y(2)=SQRT N SCLD 17 $
,		, LQ	, STORE+3	, 35	, K(0) in Q(35) $
,		, QT	, C18	, A	, K-K(0) OVER 2 IN A$
,		, TV	, A	, KODD	, SET SCALE FOR SQRT N $
,		, QJ	, KODD	, KEVEN	, TEST PARITY OF K $
,	KEVEN	, MP	, C19	, STORE+5	, (Y(2)+1) OVR 2 $
,		, SA	, C20	, 1	, SCLD 17 EQUALS $
,		, LTL	, 0	, STORE+5	, SQRT (N SLD K-1) $
,	KODD	, SP	, STORE+5	, FILL	, IN A $
,		, SA	, C11	, 0	, RND SCL SQRT (1-X) $
,		, LTL	, 0	, A	, SCLD 33 IN A $
,		, MP	, A	, STORE+2	, X(X) TIMES SQRT (1-X) $
,		, LTL	, 2	, A	, SCLD 33 IN A $
,		, ST	, C12	, Q	, -ARCSINX SCLD 33 IN Q $
,	TEST	, IJ	, STORE+1	, NEGY	, TEST FOR NEG Y $
,		, TN	, Q	, Q	, Y POS $
,	NEGY	, TP	, Q	, Y	, STORE FUNCTION $
,		, MJ	, 0	, EXIT	, $

PROBLEM RRF5

CODED BY Franck,Bernick DATE REV.6-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
	, XONE	, TN	, C12	, Q	, FOR X = 1	$
	,	, MJ	, 0	, TEST	,	$
	, C	, B10	, 00000	, 00000	, 1 SCLD 33	$
	, C1	, B	,	, 0	, ZERO	$
	, C2	, B	,	, 1	, ONE	$
	, C3	, B53	, 24135	, 20070	, A(7) SCLD 44	$
	, C4	, B33	, 24414	, 25535	, A(6) SCLD 42	$
	, C5	, B56	, 40071	, 51545	, A(5) SCLD 40	$
	, C6	, B37	, 50417	, 41234	, A(4) SCLD 40	$
	, C7	, B46	, 23706	, 66522	, A(3) SCLD 39	$
	, C8	, B26	, 61651	, 66073	, A(2) SCLD 38	$
	, C9	, B44	, 42003	, 30653	, A(1) SCLD 37	$
	, C10	, B31	, 10375	, 51633	, A(0) SCLD 34	$
	, C11	, B37	, 77777	, 77777	, ROUND CONSTANT	$
	, C12	, B14	, 44176	, 65211	, PI OVER 2 SCLD 33	$
	, C13	, B10	, 00000	, 00001	, LIMIT ON X	$
	, C14	, B26	, 47670	, 31361	, D=24 291 062 513	$
	, C15	, B	,	, 65324	, A = 27 348	$
	, C16	, B11	, 45346	, 44516	, B=10 291 988 814	$
	, C17	, B33	, 06571	, 40273	, C=29 104 062 651	$
	, C18	, B	,	, 77	, K MASK	$
	, C19	, B26	, 50117	, 14640	, SQRT 2 RND SCLD 34	$
	, C20	, B20	, 00000	, 00000	, 1 SCLD 34	$
	, CODE	, B54	, 54311	, 00000	, ERROR CODE	$
	,	, ENDSUB	,	,	,	$

A. Identification

CE 1002 Tape Handler*

Harold Dahlbeck Sperry Rand
Linnea Laure' Corps of Engineers
1 June 1957

B. PURPOSE

This program provides the facilities to read, write, move or rewind a tape on any or all Uniservos in variations of mode, spacing and direction.

It provides for the preparation of tapes to be read by several of the peripheral equipments.

Provision is also made for the storage of data in the memory in inverse or direct order, and may be read into memory in various modes by indicating the number of blocks and/or amount of working storage to be filled.

The routine can keep an account of the relative position of all tapes on the Uniservos in terms of blocks from the leader position, as well as to establish conditions for writing a full tape. It will also warn the user of undesirable orders, such as reading or moving backward more blocks than it has moved forward. When a tape reading fault occurs the routine will automatically attempt rereads of that block in various directions and at different bias levels.

C. METHOD

The program takes as its control the interpretation of two input parameters which initiate all tape actions. These parameters function as pseudo tape commands.

* This routine follows the set of specifications set forth in November 1956 by Capt. Roger Bate and George Toal of the Corps of Engineers.

There are two counters kept for each servo:

1. Block counter R________ advances on all read, write, and move orders in the forward direction; decreases on all read and move backward.
2. Tape length counter N________ increased each time a block is written onto the tape.

These two counters should be set to zero whenever a tape action is initiated from the beginning of the tape.

D. USAGE

1. The routine was programmed to be used as a straight subroutine, or as a USE subroutine.

   It is entered by means of the sequence

r	RJ	t + 2 t	(t is starting address of routine)
r + 1	normal return		

2. There are two input parameters which must be pre-inserted

P1	at	t + 3
P2	at	t + 4

3. Before using this routine, one should insure the proper setting of the following preset parameters for a given installation.

1.	NCON	determines tape length
2.	SERVV	number of servos plus one
3.	UL	upper address limit
4.	LL	lower address limit

Manual control of the routine is also possible by means of a small additional routine known as Test 1.

E. CODING INFORMATION

1. Tape Handler has 653 words.

2. The Tape Handler inserts a 45 command into 17777.

3. There are three print routines which can be used independent of the tape handler.

   a. Print Non-Negative decimal integer K.

   ```
   SP    K      0
   RJ    PEXIT  PRINT
   ```

   b. PRINT FLEX-CODE Word K

   ```
   SP    K      42
   RJ    W3     W1
   ```

   c. PRINT OCTAL Word K

   ```
   SP    K       39
   RJ    PREXIT  PRWORD
   ```

   These tags are internal to the Tape Handler.

<u>Parameters</u>

The two parameter words have the octal form:

P1:

OPERATION C D	B = Number of Blocks W = Number of words	M = 1st storage location

P2

OPERATION Read Write	L = Length of Block or Length of storage bin	Not Used	Operation A B	Servo S S

Every reference to the routine must be prefaced by the insertion of these two words: Every pair must obey the following general restrictions.

1. $1 \leq SS <$ No. of servos plus one.
2. $LL \leq M \leq UL$
3. $LL \leq M + L \leq UL$

Care should be taken to select the proper parameters to effect the required action; thus, a brief discussion of how the routine translates the parameters is in order.

OP.A is first translated. This option permits the programmer to set the servo counters to zero in order to begin a fresh tape with zero readings and hence keep a meaningful block and tape length count. Special heed should be given to this option in order to avoid ambiguous interpretation of the counters which are automatically varied with the tape action.

OP.A.

0	Has no effect on servo counters
1	Sets R = 0 for servo SS
2	Sets R = 0 for all servos
3	Sets N = 0 for servo SS
4	Sets N = 0 for all servos
5	Sets N = 0 & R = 0 for servo SS
6	Sets N = 0 & R = 0 for all servos
7	Sets switch to bypass test for full output tape when writing. Resets on next entry to this routine.

OP.B is next translated and selects the required action.

0	No action. Exits from routine setting Q to zero.
1	Sets Read switch (after selecting tape code and increment) Then it decodes OP.

READ.

2	Sets write switch (after selecting tape code and increment) then decodes OP.
	Write.
3	Moves B blocks forward or backward. Exits from routine setting Q to zero.
4	Rewind servo SS.
5	Rewind servo SS with interlock.
6	Rewind all servos.
7	Rewind all servos with interlock.

After the selection of a rewind, both servo counters are set to zero for the appropriate servo. Q is set to zero and normal exit is made.

OP.C initiates no tape action by itself; serves only to select the necessary EF code word as well as to set the required switches in the routine; e.g., tape direction.

READ OPTION

0	Variable Forward
1	Variable Backward
2,6,7	Fixed Forward
3	Fixed Backward
4	HSP Forward
5	HSP Backward

WRITE OPTION

0 or 1	Variable
2 or 3	Fixed
4 or 5	HSP
6	Tape to Card (Fixed Block 128 - .1" - 2.4")
7	Uniprinter II (Fixed Block 50 - 0" - 2.4")

Options 6 and 7 should be read by using options 2 and 3. HSP options reads a true machine variable n word block, whereas writing with the HSP option writes a 20 word block.

The direction of address incrementing in memory is determined from OP. D.

OP. D.

0 = 3 = 4 = 5 = 6 = 7	Sets increment of plus one for forward direction.
1	Sets increment of minus one for backward direction.
2	Sets increment of zero (One location for all words).

The WRITE options are translated only if the write switch was set (OP. B)

0 Write W words from M. OP.C selects writing mode.

(0=2=3=5=7) a - Fixed mode assumes 120 word blocks.

b - HSP mode assumes 20 word blocks.

c - Variable mode writes an (L + 3) word block. The BLI is automatically generated from L and added during the write.

If W is not an even multiple of 120, 20 or L, respectively, the final block will be padded with tape sentinels to give a full block.

If W is equal to zero, a stopper block is written onto tape.

1 Write B blocks, from M. Variable mode only. Assumes that each block in memory has one BLI followed by n data words. (This implies that M contains a BLI.) Before writing each block, a test is made to insure that the first word of each block contains a BLI. L will be reduced by n + 1 for each block to prevent

exceeding the initial L. Failure to meet either of these conditions causes an alarm printout.

4 Write B blocks from M. OP. C selects writing mode

a - Fixed mode assumes 120 word blocks

b - HSP mode assumes 20 word blocks

c - Variable mode writes an $(L+3)$ word block. The BLI is automatically generated from L and added during the write.

Before starting this option, a test is made to see whether the number of words as given by blocks times size will satisfy $LL \leq M + W \leq UL$

If W is equal to zero, a stopper block is written onto tape.

6 Write all blocks contained within L words from M. Variable mode only. Assumes that <u>each</u> block in memory has one BLI followed by n data words. (This implies that M contains a BLI). The first word following the last block must be a bin sentinel. Starting at M, blocks will be successively written on tape until one of the following conditions occur:

1. Bin Sentinel Found — normal exit
2. Last block exceeds reduced L — Alarm print
3. Proper location does not contain BLI — Alarm print

The READ Options are translated only if the read switch was set (O^P.B.)

0 Read B blocks into M. OP.C selects reading mode.

(0=2=3=6) a - Fixed mode reads in 120 word blocks.

b - HSP mode reads in N word blocks.

c - Variable mode does not insert BLI or sum check in memory.

1 Read B blocks into M. Variable mode only.

Insert one BLI (but not Sum Check) into memory as first word of block followed by n data words.

For options 1 and Oc, L = maximum storage allowed and is reduced as each <u>variable block</u> is read in. Should the remaining space be insufficient to store the last block; reading is stopped, the last block is not read in and the tape will be repositioned. This latter action will cause an alarm print.

4 Read Blocks as needed to fill up L words.

On variable mode, no BLI or sum check is inserted.

5 Read Blocks as needed to fill up L words. Variable mode only.

Inserts one BLI (but not sum check) into memory as first word of block followed by n data words.

For options 4 and 5, (L - 1) = maximum storage allowed. A special end sentinel (07 77777 77777) is added as the next word after the last valid word stored. W = longest variable block expected. Should the remaining space be insufficient, (as determined by the incoming BLI) the last block will not be read in and the tape is repositioned. Also, if W is larger than the reduced L, no further attempt will be made to read.

7 Check Read Tape OP. C selects mode.

At any desired time, this option may be invoked to

1. Print out

   CHRD

   Servo No.

   Block NO.

2. Read backward from tapes in the mode selected <u>that many blocks</u> without reading the words into memory.

3. Rewind the tape on that servo without interlock.

The necessary parameter words for check reading are:

Fixed	30,00001,M
	70,02000,001SS
HSP	50,00001, M
	70,02000,00 1 SS
Variable	10,00001, M
	70,02000,00 1 SS

ALARM EXIT AND ERROR PRINTOUT

Since the Tape Handler is normally referenced and controlled from external routines by means of the two parameter words, it was felt necessary to incorporate into the routine itself a method which guards against the insertion of improper parameter words as well as to monitor the dynamic results of the initiated tape actions. To facilitate this aim, it was required that normal exit from the routine would always be accompanied by the insertion of an appropriate bit pattern into the Q-Register, which could then be interrogated further by an external diagnostic routine which checks the bits in Q immediately following the exit from the Tape Handler.

There are only three normal exits from the tape handler:

1. When this routine exits with no unexpected results, $Q_{35}=0$.
2. Tape End Sentinel Found $Q_{35}=Q_{34}=1$
   - a - Test is made for variable and fixed mode.
   - b - The block <u>was</u> read into memory.
   - c - The tape is <u>not</u> repositioned or rewound.
   - d - Exit is preceded by a printout.
3. Output Tape Full $Q_{35}=Q_{33}= 1$
   - a - <u>After</u> each block is written, the tape length counter is increased by an amount dependent on block size and spacing.
   - b - Unless the bypass option was selected (A = 7), a test is made to see if the last accumulated sum exceeded the tape length constant NCON. If NCON was exceeded, a stopper block is automatically written onto tape and the block count increased by one. HSP option does not receive a stopper block.

c - Exit is preceded by a printout.

Before each exit the accumulator will be filled with a 77 00000 xxxxx where xxxxx will be the initial address of the current block entry should the read order be interrupted by an alarm condition.

On an exit print out, the format will consist of the following six lines:

Explanation		Example
1. Normal Exit Line		45 00000 01234
2. P1		130 00001 04000
3. P2		00 02000 00103
4. Error (or exit)	No.	Exit 34
5. Servo	No.	Servo 3
6. Block	No.	Block 63

This printout should enable the programmer to discover the cause of the printout.

In the example given above, one can readily obtain the following information:

1. The tape handler was last referenced from address 01233.
2. No servo counters were set to zero.
3. The action called for was a read backward one fixed block from servo 3 into the location 04000 and incrementing the address forward in memory.
4. The exit number states that a tape sentinel was discovered in block 63.

SUMMARY OF FLEXO-WRITER PRINTOUTS

NO	EXPLANATION	TEST CONDITION	ACTION TAKEN
34	End Sentinel Found	1. Variable mode: Found stopper block 2. Fixed mode: First word was tape sentinel	Print Exit
33	End of Tape	NCON $<$ N *after* writing last block	1. Automatically writes stopper block (except in HSP) 2. Advances block counter 3. Print and Exit
32	Mod 6	Last failure to read block due to Mod 6	Print and Cease *
31	Unrockable Parity	Last failure to read block due to parity	Print and Cease *
30	Unrockable Sum Check	Last failure to read block due to sum check	Print and Cease *
29	Possible 720	MACHINE HANGS UP on 720 or sprocket	Set PAK to 17777 to print and exit from routine.

* Halts on easily identified MS 00123, XXXXX. Push start button to attempt another reread of block.

9	Illegal Servo No.	$0 < SS < SERVV$	Print and Cease
8	Insufficient Space	$LL < M < UL$	Print and
	Sign depends upon	$LL < M \not< L < UL$	Cease
	direction in memory.	$LL < M \not< W < UL$	
	($\not<$ = forward)		
7	Move Back Neg	$N < 0$	Print and
	Read Back Neg		Cease
6	L reduced to point where (L - Block Size) < 0		Print and
	read or write not possible		Cease
5	Not Block Length Ind. 00 XXXXX XXXXX		Print and
			Cease

Note: Recovery is possible from errors 9, 8, 7, and 5 by

1. Correcting Parameters P1 and P2 manually.
2. Setting PAK to T. H. starting address and push START.

GLOSSARY

FIXED BLOCK	The standard fixed block length is 120 words. The block and blockette spacing depend on the option selected.
VARIABLE BLOCK	The block contains n +3 words where the first and last words contain the Block Length Indicators, and immediately preceding the last word is the word containing the Sum Check.
BLOCK LENGTH INDICATOR (BLI)	One octal word appearing as 00 XXXXX YYYYY where X = Y = the number of words in a variable block excluding the B. L. I.'s and the Sum Check.
SUM CHECK	One word which equals the least 36 bits of the algebraic summation of all the words within the block excluding the B. L. I. 's and the Sum Check itself.
HSP BLOCK	The true machine variable block consisting of n words. However, since this option was intended for use with the High Speed Printer, the writing of these blocks was restricted to 20 words. This restriction can be removed easily.
SENTINELS	Tape Sentinels: 74 74747 47474 which prints as ZZZZZZ Basket Sentinel: 07 77777 77777 (used internally with the tape handler routine).
STOPPER BLOCK	1. Fixed block length option, This block contains 120 words of tape sentinels. 2. Variable block length option. This block consists of only one word, the tape sentinel.
SS	Servo Number (Two octal digits)
M	Initial storage location from which words are written or read into memory.

B	Number of blocks (Four octal digits)
W	Number of Words (Five octal digits)
R	Block Counter for servo
N	Tape Length Counter for Servo
L	Has various meanings depending on use. In general, on writing variable blocks, it is the number of words in the block. On reading, it determines the amount of space available.
SERVV	Preset constant equal to one more than the number of servos existing at a particular installation.
UL	Preset upper limit in memory beyond which reading and writing is Prohibited.
LL	Preset lower limit in memory below which reading and writing is Prohibited.
N CON	Preset constant used to limit writing on tapes beyond a desired length. Measured in tenths of frames. Currently calculated from the formula: (1500)' 90% X 12" X 1280 = 20,736,000 tenths of frames at 128/inch.

RWBFJN, Floating Point Bessel Function J_n Subroutine

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: September 16, 1957

A. Purpose:

This subroutine calculates $J_n(x)$ for arbitrary x and integer n. Floating point arithmetic is used.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

SUB,	RWBFJN,	171
TEMPS,	5,	3
INOUT,	2,	1

2. Input

First word - x in floating point.

Second word - n in fixed point scaled at 2^0.

3. Output

First word - $J_n(x)$ in floating point.

4. Space required

Length of subroutine - 171 words.

Temporary storage in compiled region - 8 words.

Other routines used - RWSQF1, RWCNF4

Other temporary storage - None

5. Error codes - The alarm exit is not used

C. Restrictions and Coding Information:

The routine uses 1103A built-in floating point arithmetic. x must be a standard floating point number and n must be an integer.

D. Timing:

If $|x| > 3$ and $|n| = 0$ or 1, t = 15 m. s.

If $|x| \geq 3$ and $|x| \geq |n| > 1$ then t = $25 + |n|$ m. s.

If $|x| < |n|$ or $|x| < 3$ then t = $15 + 1.2\,|n|$ m. s.

E. Mathematical Method:

1. Formulas.

(a) For J_o and J_1 when $|x| \geq 3$ asympotic formulas are used. See E. E. Allen, Analytical Approximations , Mathematical Tables and Other Aids to Computation, Vol. 6, October, 1954, pp. 240-1.

(b) For $J_n(x)$ when $|x| > |n| > 1$ and $|x| \geq 3$ the recursion formula

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x)$$

is used for which the starting values $J_o(x)$ and $J_1(x)$ are computed as in (a).

(c) For $J_n(x)$ when $|x| < |n|$ or $|x| < 3$ then a method described in NN85* is employed.

2. Error.

The routine was checked out by computing $J_n(x)$ for $x = 0\,(1)\,100$ and $n = 0(1)\,50$. In this range the error never exceeded 10^{-6}.

---

*Numerical Note No. 85, "Computations of Bessel Functions," Digital Computing Center, The Ramo-Wooldridge Corporation.

```
,     ,     SUB,RWBFJN,171,     $
,     ,   INOUT,2,1,$
  ,       ,TEMPS  ,5          ,3          ,     $
 ,        ,MJ     ,           ,START      ,     $
 ,        ,ALARM  ,           ,           ,     $
 ,EXIT    ,MJ     ,           ,FILL       ,     $
 ,ANS     ,       ,           ,           ,     $
 ,ARG1    ,       ,           ,           ,     $
 ,ARG2    ,       ,           ,           ,     $
 ,START   ,TM     ,ARG1       ,X          ,     $
 ,        ,TM     ,ARG2       ,NX         ,     $
 ,        ,TP     ,NX         ,NF         ,     $
 ,        ,NP     ,NF         ,K1         ,     $
 ,        ,TP     ,X          ,A          ,     $
 ,        ,TJ     ,K2         ,SMALL      ,     $
 ,        ,TJ     ,K3         ,M          ,     $
 ,        ,TJ     ,NF         ,M          ,     $
 ,        ,TP     ,NX         ,A          ,     $
 ,        ,TJ     ,K4         ,J          ,     $
 ,        ,RJ     ,J13        ,J          ,     $
 ,        ,TP     ,Q          ,T1         ,     $
 ,        ,RJ     ,J13        ,J0         ,     $
 ,        ,TN     ,Q          ,T0         ,     $
 ,        ,RS     ,NX         ,K4         ,     $
 ,        ,FD     ,K7         ,X          ,     $
 ,        ,TP     ,Q          ,C1         ,     $
 ,        ,MJ     ,           ,L1         ,     $
 ,L       ,TN     ,T1         ,T0         ,     $
 ,        ,TP     ,Q          ,T1         ,     $
 ,        ,FA     ,NF         ,C1         ,     $
 ,L1      ,TP     ,Q          ,NF         ,     $
 ,        ,FP     ,T1         ,T0         ,     $
  ,       ,IJ     ,NX         ,L          ,     $
  ,SIGN   ,TP     ,Q          ,ANS        ,     $
  ,       ,TM     ,ARG2       ,Q          ,     $
  ,       ,LQ     ,Q          ,35         ,     $
  ,       ,QJ     ,S1         ,EXIT       ,     $
  ,S1     ,TP     ,ARG1       ,Q          ,     $
  ,       ,TP     ,ARG2       ,A          ,     $
  ,       ,SJ     ,S2         ,S3         ,     $
  ,S2     ,QJ     ,EXIT       ,S4         ,     $
  ,S3     ,QJ     ,S4         ,EXIT       ,     $
  ,S4     ,TN     ,ANS        ,ANS        ,     $
  ,       ,MJ     ,           ,EXIT       ,     $
 ,SMALL   ,TP     ,NX         ,A          ,     $
 ,        ,ZJ     ,M10        ,S5         ,     $
 ,S5      ,TP     ,K17        ,ANS        ,     $
 ,        ,MJ     ,           ,EXIT       ,     $
 ,M       ,FM     ,X          ,K8         ,     $
 ,        ,TP     ,Q          ,A          ,     $
 ,        ,TJ     ,NF         ,M1         ,     $
 ,        ,TP     ,Q          ,NF         ,     $
 ,        ,UP     ,NF         ,T0         ,     $
 ,        ,LQ     ,T0         ,24         ,     $
```

```
,       ,RS     ,T0     ,K9     ,       $
,       ,SJ     ,M1     ,M2     ,       $
,M2     ,TU     ,A      ,M3     ,       $
,       ,TP     ,NF     ,A      ,       $
,M3     ,LT     ,FILL   ,NF     ,       $
,M12    ,RA     ,NF     ,K10    ,       $
,       ,ST     ,K14    ,C1     ,       $
 ,       ,NP     ,NF        ,K13    ,   $
,       ,TP     ,K5     ,SUMT   ,       $
,       ,FD     ,K7     ,X      ,       $
,       ,TP     ,Q      ,C3     ,       $
,       ,TP     ,K11    ,ALT    ,       $
,       ,TP     ,C1     ,Q      ,       $
,       ,LQ     ,Q      ,35     ,       $
,       ,QJ     ,M4     ,M5     ,       $
,M4     ,LQ     ,ALT    ,1      ,       $
,M5     ,TP     ,K5     ,T0     ,       $
,       ,TP     ,K12    ,T1     ,       $
,       ,FD     ,NF     ,X      ,       $
,       ,RS     ,NX     ,K14    ,       $
,       ,MJ     ,       ,M7     ,       $
,M11    ,TP     ,T1     ,T0     ,       $
,       ,TP     ,Q      ,T1     ,       $
,       ,EJ     ,NX     ,ST0    ,       $
,M6     ,FS     ,NF     ,C3     ,       $
,M7     ,TP     ,Q      ,NF     ,       $
,       ,LQ     ,ALT    ,1      ,       $
,       ,QJ     ,SUM    ,M8     ,       $
,M8     ,TM     ,T1     ,A      ,       $
,       ,TJ     ,K15    ,M9     ,       $
,       ,MJ     ,       ,M10    ,       $
,M9     ,FM     ,T1     ,NF     ,       $
,       ,FS     ,Q      ,T0     ,       $
,       ,IJ     ,C1     ,M11    ,       $
,       ,EJ     ,NX     ,ST01   ,       $
,M13    ,FI     ,K7     ,SUMT   ,       $
,       ,TP     ,Q      ,SUMT   ,       $
,       ,FD     ,ANS    ,SUMT   ,       $
,       ,MJ     ,       ,SIGN   ,       $
,ST0    ,TP     ,T1     ,ANS    ,       $
 ,       ,MJ     ,          ,M6     ,   $
,SUM    ,FA     ,SUMT   ,T1     ,       $
,       ,TP     ,Q      ,SUMT   ,       $
,       ,MJ     ,       ,M8     ,       $
,M10    ,TP     ,K5     ,ANS    ,       $
,       ,MJ     ,       ,EXIT   ,       $
,M1     ,TP     ,NX     ,NF     ,       $
,       ,MJ     ,       ,M12    ,       $
,ST01   ,TP     ,Q      ,ANS    ,       $
,       ,MJ     ,       ,M13    ,       $
,J      ,RWSQF1,X      ,SQ     ,       $
,       ,FD     ,K3     ,X      ,
,       ,TP     ,Q      ,SQ#4   ,
,       ,TP     ,NX     ,A      ,
```

```
,       ,EJ     ,K5      ,J0       ,     $
,J1     ,TP     ,CON3    ,Q        ,     $
,       ,RPV    ,6       ,J11      ,     $
,       ,FP     ,SQ#4    ,CON3#1   ,     $
,J11    ,TP     ,Q       ,T0       ,     $
,       ,TP     ,CON4    ,Q        ,     $
,       ,RPV    ,6       ,J12      ,     $
,       ,FP     ,SQ#4    ,CON4#1   ,     $
,J0     ,TP     ,CON1    ,Q        ,     $
,       ,RPV    ,6       ,J01      ,     $
,       ,FP     ,SQ#4    ,CON1#1   ,     $
,J01    ,TP     ,Q       ,T0       ,     $
,       ,TP     ,CON2    ,Q        ,     $
,       ,RPV    ,6       ,J12      ,     $
,       ,FP     ,SQ#4    ,CON2#1   ,     $
,J12    ,FA     ,Q       ,X        ,     $
,       ,RWCNF4,Q        ,CN       ,     $
,       ,FD     ,A       ,SQ#3     ,     $
,       ,FM     ,Q       ,T0       ,     $
,J13    ,RJ     ,J13     ,J13#1    ,     $
,       ,MJ     ,        ,SIGN     ,     $
,K1     ,23     ,30000↑B ,         ,     $
,K2     ,F      ,1.0D-7  ,         ,     $
,K3     ,F      ,3.0     ,         ,     $
,K4     ,       ,        ,2        ,     $
,K5     ,       ,        ,0        ,     $
,K7     ,F      ,2.0     ,         ,     $
,K8     ,F      ,1.5     ,         ,     $
,K9     ,       ,167↑B   ,         ,     $
,K10    ,       ,        ,10       ,     $
,K11    ,52     ,52525↑B ,25252↑B  ,     $
,K12    ,       ,14000↑B ,         ,     $
,K13    ,23     ,40000↑B ,         ,     $
,K14    ,00     ,        ,1        ,     $
,K15    ,33     ,0000↑B  ,         ,     $
,K17    ,F      ,1.0     ,         ,     $
,CON1   ,F      ,.0001   ,4476     ,     $
,       ,F      ,-.0007  ,2805     ,     $
,       ,F      ,.0013   ,7237     ,     $
,       ,F      ,-.0000  ,9512     ,     $
,       ,F      ,-.0055  ,2740     ,     $
,       ,F      ,-.0000  ,0077     ,     $
,       ,F      ,.7978   ,8456     ,     $
,CON2   ,F      ,.0001   ,3558     ,     $
,       ,F      ,-.0002  ,9333     ,     $
,       ,F      ,-.0005  ,4125     ,     $
,       ,F      ,.0026   ,2573     ,     $
,       ,F      ,-.0000  ,3954     ,     $
,       ,F      ,-.0416  ,6397     ,     $
,       ,F      ,-.7853  ,9816     ,     $
,CON3   ,F      ,-.0002  ,0033     ,     $
,       ,F      ,.0011   ,3653     ,     $
,       ,F      ,-.0024  ,9511     ,     $
,       ,F      ,.0001   ,7105     ,
```

```
,       ,F      ,.0165    ,9667     ,                                    $
,       ,F      ,.0000    ,0156     ,                                    $
,       ,F      ,.7978    ,8456     ,                                    $
,CON4   ,F      ,-.0002   ,9166     ,                                    $
,       ,F      ,.0007    ,9824     ,                                    $
,       ,F      ,.0007    ,4348     ,                                    $
,       ,F      ,-.0063   ,7879     ,                                    $
,       ,F      ,.0000    ,5650     ,                                    $
,       ,F      ,.1249    ,9612     ,                                    $
 ,       ,F      ,-2.35619449,          ,                                $
 ,SUMT   ,EQUALS,CN#3       ,          ,                                 $
 ,C1     ,EQUALS,CN#4       ,          ,                                 $
 ,C3     ,EQUALS,SQ#3       ,          ,                                 $
 ,ALT    ,EQUALS,SQ#4       ,          ,                                 $
,       ,ENDSUB,          ,         ,                                    $
```

RWRTI 1, Complex, Floating Point Zeros of Arbitrary Functions

Author: Werner L. Frank, The Ramo-Wooldridge Corporation

Date: August 1, 1957

A. Purpose:

This subroutine employs an iterative procedure for finding zeros of arbitrary functions of the form f(z) = 0 by a method due to D. Muller. The programmer must provide an auxiliary routine which computes the value of the function f(z) for any argument z presented to it by this subroutine. A number of output options are available in addition to options in the starting procedure. The computation is performed in complex floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point package.

SUB,	RWRTI 1,	464
TEMPS,	1 ,	0 ,
INOUT,	2 ,	4 ,

2. Input

First word: XO, AUXLOC, N

Second word: YO, OOOOO , ANSWER

where AUXLOC = Address of the location of the auxiliary routine. This routine must be coded so that the first and second cells are the exit and entry respectively. That is, RWRTI 1 must be able to enter the auxiliary with RJ AUXLOC, AUXLOC + 1. The argument z will be located in the first and second of the result cells and the value of the function f(z) is to be placed in the third and fourth of the result cells of the subroutine.

N = Number of roots to be sought

ANSWER = Region of 2N cells where the roots are located upon leaving the subroutine. Also, estimates of roots, if available, are placed here prior to initiating the subroutine. Otherwise this region must contain all zeros.

X = Output option (see below)

Y = Iteration option

Y = 0, Normal mode. All roots are found by N applications of the iterative procedure.

Y = 4, Special mode. Whenever

a complex root is located its conjugate is automatically also accepted as a root.

3. Output (used by the auxiliary routine)

First word - z_R

Second word - z_I

Third word - $f_R(z)$

Fourth word - $f_I(z)$

where z_R = real part of z

z_I = imaginary part of z

$f_R(z)$ = real part of f(z)

$f_I(z)$ = imaginary part of f(z)

Depending upon the option chosen the roots which are found are output on magnetic tape by employing the output processor system. In addition the roots are also stored internally as indicated above. The options available are selected by choice of the octal digit X in the first parameter cell:

X = 4 All successive iterants and associated functional values are output with 6 column, floating point format. The order is

$$z\ ,\ f(z),\ f_r(z)$$

X = 0 Only the final iteration and not the intermediate iterations are output. The ordering is as above.

(See Mathematical Method for exceptions.)

An identification number for each line of printing is printed on the left margin and two line skips are made between roots. A page eject is made prior to leaving the subroutine.

X = 2 No output is written. The output processor is not activated.

In all three cases the results will be located in the region reserved for them. A program constant m, limits the number of iterations to 100. At that time the current iterant is accepted as a root and the process continues. This constant is located in word 425 (C 9) of the subroutine.

4. Space required

Length of Subroutine: 464 cells

Temporary storage in compiled region: 1 cell

Other subroutines used: RWP 1
RWF 1
Auxiliary routine

Other temporary storage: 2N cells

5. Error codes - The alarm exit is not used. It is possible to obtain machine exponent overflow alarm when the floating point number range is violated. It is advised in such instances that the problem be rescaled if possible.

C. Restrictions and Coding Information

Numerical data must be in standard USE complex floating point representation. The program is self-contained, performing its own complex arithmetic and calling for the output subroutine RWP 1 and RWF1. The 1103A built-in floating point package is used. The auxiliary program must not destroy the argument z in the first and second words of the output cells.

D. Timing

The timing is largely dependent on the amount of computation necessary to evaluate f(z). Computing time for one iteration, excluding output and the evaluation of f(z), is approximately 45 milliseconds.

E. Mathematical Method

In [1] D. Muller discusses an iterative procedure for finding real and complex roots of a polynomial equation whose coefficients may be complex. Selecting three arbitrary points z_1, z_2 and z_3 as starting values, one chooses z_4, the next approximation to the root, as one of the zeros of the second-degree polynomial which passes through the functional values $F(z_1)$, $F(z_2)$ and $F(z_3)$. The iteration continues by dropping the first point and repeating the quadratic fit for the points z_2, z_3 and z_4 and the associated functional values. The process halts when successive iterants pass a suitable convergence test.

The formal process described above is applied in this routine to any equation of the form $F(z) = 0$. The proof of convergence in the small, which was given by Muller for polynomial equations, also holds for general equations of the form $F(z) = 0$, where $F(z)$ is analytic in the neighborhoods of its roots. Although no proof of convergence in the large is available, to our knowledge the method has never failed on any of these general problems which have been attempted at The Ramo-Wooldridge Corporation.

1. Muller, D. "A method for solving algebraic equations using an automatic computer," Mathematical Tables and other Aids to Computation, Vol. X, Oct. 1956 pp. 208-215

The explicit formulas used in this computation are:

$$z_{i+3} = z_{i+2} + (z_{i+2} - z_{i+1})\, d_{i+3},$$

where

$$d_{i+3} = \frac{-2F(z_{i+2})\,(1+d_{i+2})}{b_{i+2} \pm \left\{ b_{i+2}^2 - 4F(z_{i+2})\, d_{i+2}(1+d_{i+2}) \left[F(z_i) d_{i+2} - F(z_{i+1})(1+d_{i+2}) + F(z_{i+2}) \right] \right\}^{1/2}}$$

and

$$b_{i+2} = F(z_i) d_{i+2}^2 - F(z_{i+1})\,(1+d_{i+2})^2 + F(z_{i+2})\,(1+2d_{i+2}).$$

The sign in the denominator is chosen to give d_{i+3} the smaller magnitude. Coupled with one of the optional starting procedures where $z_1 = -1$, $z_2 = 1$ and $z_3 = 0$ the roots are found in an approximately ascending order of magnitude.

After one root is obtained the subroutine proceeds with the effective function $F_1(z)$ where

$$F_1(z) = \frac{F(z)}{z - z_1}$$

and in general we consider

$$F_r(z) = \frac{F(z)}{\prod_{i=1}^{r} (z - z_i)} \qquad r = 1, 2, \ldots, N-1$$

This procedure works readily for functions having simple roots. On the other hand, if a function possesses multiple roots, ζ then $F_r(z)$ is indeterminate when z approaches a ζ which may already have been found. However, even in this case the process has never failed. In fact, roots of multiplicity six or more have been found successfully. This is primarily due to the fact that multiple roots are found to much less accuracy than simple roots and behave, in effect, like clustered roots.

The structure of this subroutine is given in the flow diagram below. A number of details are noted here.

Starting procedure - In finding the i^{th} root the 2i and 2i+1 cell of the answer region are inspected and one of the following options is selected:

(a) Contents zero - starting values are -1, 1, 0

(b) Contents $z \neq 0$ - starting values are z and $z(1 \pm \epsilon)$ where $\epsilon = 10^{-3}$ in this routine. The quantity ϵ can be changed by the programmer by suitable altering the 31^{st} and 33^{rd} word of this subroutine.

It is important to note that in the case of bunched roots it is possible that one is not led to the root closest to an approximation which may have been used to start the iteration. While convergence to some root will occur, this may invalidate the effectiveness of some other guess which may follow.

Output The processor output system is employed to:

(a) Output all iterations for a root with the information: identification, z_i, $F(z_i)$ and $F_r(z_i)$

(b) Output only the finally converged to values z, $F(z)$ and $F_r(z)$ and the identification

(c) No output.

In all three instances the i^{th} root z is placed in the 2i and 2i+1 cells of the answer section destroying the estimate which may have been there.

Convergence Several convergence tests are required to accomodate all the special cases arising. The iteration continues until

(a) $$\left| \frac{z_{i+1} - z_i}{z_{i+1}} \right| < \epsilon_1$$

or (b) $$|F(z_i)| < \epsilon_2 \text{ and } |F_r(z_i)| < \epsilon_2$$

Test (b) is necessary in order to stop the iteration if the lower bound (2^{-128}) of the floating point number range is reached and to prevent discrepancies in the values $F_r(z_i)$ and $F(z_i)$ due to the effect of $\prod_{i=1}^{r} (z-z_i)$ which may be very large or very small. In this routine $\epsilon_2 = 10^{-20}$ has been found sufficient for most purposes. The programmer can change this quantity by altering word 428 (C12) of this subroutine.

In (a) one actually tests

$$\max \exp(z_{i+1}) - \max \exp(z_{i+1} - z_i) > \epsilon_1^*$$

where max exp(z) is the larger of the two exponents associated with the real and imaginary component of z. In this routine $\epsilon_1^* = 20$ which yields approximately 6 or more places of accuracy in the roots. The quantity ϵ_1^* is contained in word 423(C7) of this subroutine and is also used in similar tests as described below.

Conjugate option - At the choice of the programmer the conjugate of a located complex root can be admitted as a root without any further iterations. The following criteria must be passed in order for $z = \alpha + i\beta$ to be considered a complex number:

(a) $\exp(\beta) - \exp(\alpha) < \epsilon_1^*$

and (b) $|\beta| \leq \epsilon_3$

In this routine $\epsilon_3 = 10^{-6}$ and is contained in word 427(C11) of the subroutine.

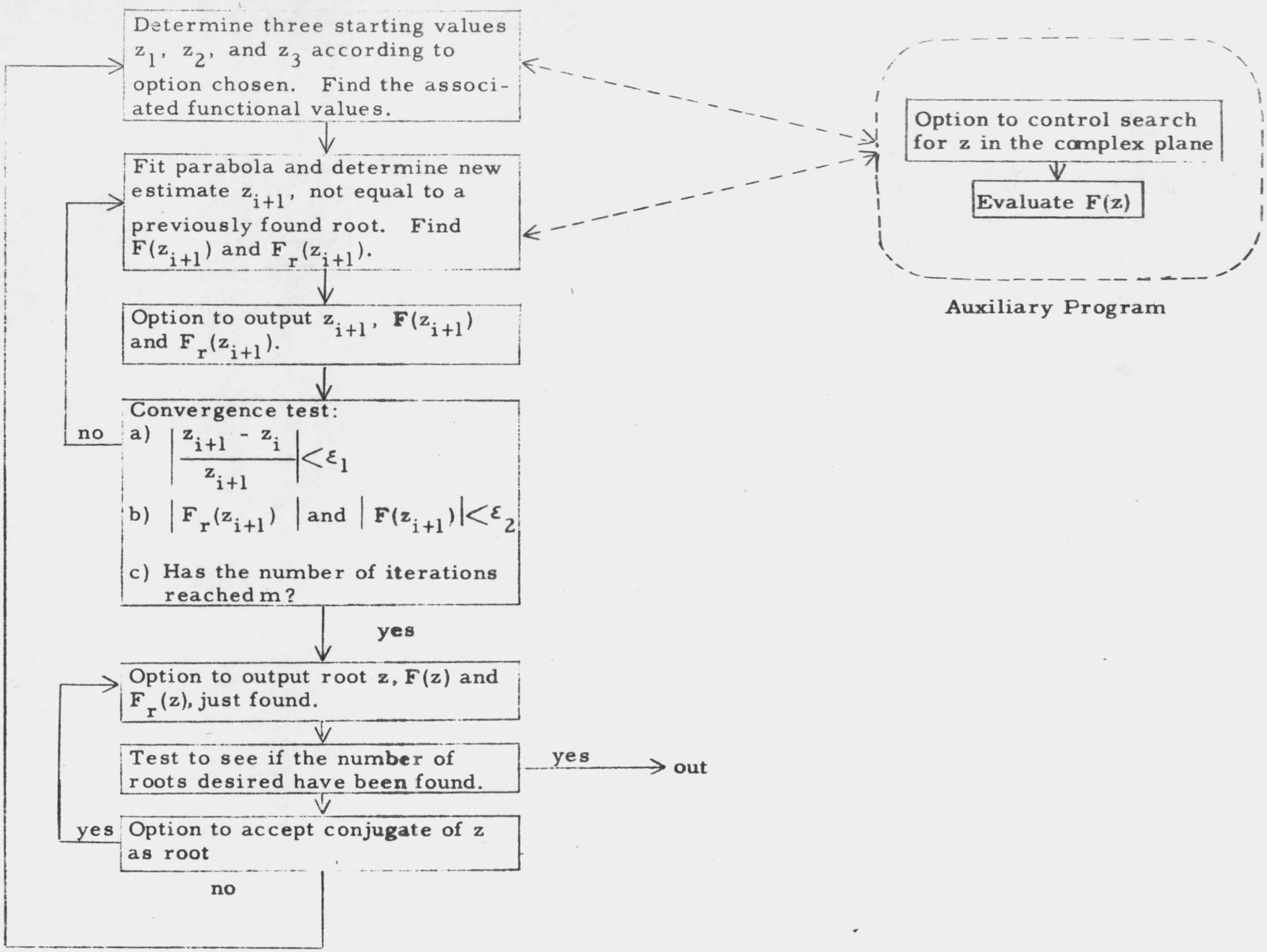
Special devices

a. In forming the product $\prod_{i=1}^{r} (z-z_i)$ each factor is tested to see that no previously found root is approached again. The test employed is max $\exp(z)$ - max $\exp(z-z_i) > \epsilon_1^*$ which also serves to prevent the product from becoming too small. In the event this test is violated an arbitrary change on z is imposed by adding ϵ (previously defined) to the real part of z. In this event the output includes only z and F(z) while $F_r(z)$ is replaced by zero.

b. After each iteration a test is made to prevent undue growth in the magnitude of the values of the function $F_r(z_i)$ and $F_r(z_{i+1})$. If these values do fluctuate a new iterant $z_i^* = \frac{(z_i+z_{i+1})}{2}$ is chosen. One line skip indicates that z_i^* was formed.

Auxiliary program. This program evaluates F(z) for a given z determined by the subroutine. This coding can include a control option limiting the region of search for a root to a well defined local portion of the complex plane by properly constraining successive iterants.

The applicability of this method to general root finding problems is noteworthy for a number of reasons. The procedure is completely general, requiring no knowledge of the location of the roots nor any special starting process. Furthermore, complex roots are obtained with the same ease as are real roots. Since the iteration requires only the evaluation of the function, and never the value of the derivative, the scheme is useful in problems where the evaluation of the derivative is very difficult. While multiple roots do not present any computational difficulties, they are obtained with less accuracy than simple roots. Also, the rate of convergence is considerably reduced in the case of multiple or clustered roots.



Flow Chart for Arbitrary Root Finding Substitute

```
,       ,SUB    ,RWRTI1  ,464      ,                  $
,       ,TEMPS  ,1       ,0        ,                  $
,       ,INOUT  ,2       ,4        ,                  $
,ENTRY  ,MJ     ,        ,START    ,                  $
,ALARM  ,ALARM  ,        ,         ,                  $
,EXIT   ,MJ     ,        ,         ,                  $
,ARG    ,       ,        ,         ,                  $
,ARG1   ,       ,        ,         ,                  $
,DET    ,       ,        ,         ,                  $
,DET1   ,       ,        ,         ,                  $
,PAR1   ,       ,        ,         ,                  $
,PAR2   ,       ,        ,         ,                  $
,START  ,TV     ,PAR2    ,N6       ,                  $
,       ,SP     ,PAR2    ,15       ,                  $
,       ,TU     ,A       ,M3       ,                  $
,       ,TP     ,C3      ,T#25     ,                  $
,       ,TV     ,PAR1    ,T#25     ,                  $
,       ,RS     ,T#25    ,C8       ,N-1               $
,       ,TU     ,PAR1    ,IN       ,                  $
,       ,SP     ,PAR1    ,57       ,                  $
,       ,AT     ,C8      ,A        ,                  $
,       ,TV     ,A       ,IN       ,                  $
,       ,TN     ,C8      ,T#23     ,SET COUNTER       $
,       ,TP     ,IN      ,INM      ,                  $
,AGAIN  ,TP     ,C3      ,T#22     ,CLEAR COUNTER     $
,       ,TP     ,C8      ,T#26     ,                  $
,       ,RPV    ,14      ,M2       ,CLEAR TEMPORARY   $
,       ,TP     ,C3      ,T        ,SECTION           $
,M2     ,RP3    ,2       ,M4       ,DETERMINE         $
,M3     ,TP     ,FILL    ,T#6      ,STARTING OPTION   $
,M4     ,TP     ,T#6     ,A        ,                  $
,       ,ZJ     ,OPT2    ,M5       ,                  $
,M5     ,EJ     ,T#7     ,OPT1     ,                  $
,OPT2   ,FM     ,C2      ,T#6      ,                  $
,       ,TP     ,Q       ,T#12     ,                  $
,       ,FM     ,C2      ,T#7      ,                  $
,       ,TP     ,Q       ,T#13     ,EX                $
,       ,FA     ,T#12    ,T#6      ,                  $
,       ,TP     ,Q       ,ARG      ,                  $
,       ,FA     ,T#13    ,T#7      ,X%1#E↑            $
,       ,TP     ,Q       ,ARG1     ,                  $
,       ,RJ     ,OUT     ,IN       ,F OF X.           $
,       ,TP     ,ANS     ,T        ,                  $
,       ,TP     ,ANS1    ,T#1      ,                  $
,       ,FS     ,T#6     ,T#12     ,                  $
,       ,TP     ,Q       ,ARG      ,                  $
,       ,FS     ,T#7     ,T#13     ,X%1-E↑            $
,       ,TP     ,Q       ,ARG1     ,                  $
,       ,RJ     ,OUT     ,IN       ,F OF X2           $
,       ,TP     ,ANS     ,T#2      ,                  $
```

```
,       ,TP    ,ANS1    ,T#3     ,            $
,       ,MJ    ,        ,M8      ,            $
,OPT1   ,TN    ,C1      ,T#12    ,            $
,       ,TN    ,C1      ,ARG     ,-1          $
,       ,TP    ,C3      ,ARG1    ,            $
,       ,RJ    ,OUT     ,IN      ,F OF -1     $
,       ,TP    ,ANS     ,T       ,            $
,       ,TP    ,ANS1    ,T#1     ,            $
,       ,TP    ,C1      ,ARG     ,            $
,       ,TP    ,C3      ,ARG1    ,1           $
,       ,RJ    ,OUT     ,IN      ,F OF 1      $
,       ,TP    ,ANS     ,T#2     ,            $
,       ,TP    ,ANS1    ,T#3     ,            $
,M8     ,TP    ,T#6     ,ARG     ,X OR ZERO   $
,       ,TP    ,T#7     ,ARG1    ,            $
,       ,RJ    ,OUT     ,IN      ,            $
,       ,TP    ,ANS     ,T#4     ,            $
,       ,TP    ,ANS1    ,T#5     ,            $
,       ,TP    ,ARG     ,T#6     ,            $
,       ,TP    ,ARG1    ,T#7     ,            $
,       ,TN    ,C4      ,T#8     ,            $
,       ,TP    ,C4      ,T#10    ,            $
,REPEAT,TN     ,T#9     ,Q       ,            $
,       ,FP    ,T#1     ,T#4     ,            $
,       ,FI    ,T#8     ,T       ,            $
,       ,TP    ,Q       ,T#14    ,            $
,       ,TP    ,T#5     ,Q       ,            $
,       ,FI    ,T#9     ,T       ,            $
,       ,FI    ,T#8     ,T#1     ,            $
,       ,TP    ,Q       ,T#15    ,            $
,       ,FM    ,T#9     ,T#15    ,            $
,       ,TN    ,Q       ,Q       ,            $
,       ,FI    ,T#8     ,T#14    ,            $
,       ,TP    ,Q       ,T#16    ,            $
,       ,FM    ,T#8     ,T#15    ,            $
,       ,FI    ,T#9     ,T#14    ,            $
,       ,TP    ,Q       ,T#17    ,            $
,       ,FM    ,T#11    ,T#3     ,            $
,       ,TN    ,Q       ,Q       ,            $
,       ,FI    ,T#10    ,T#2     ,            $
,       ,TN    ,Q       ,T#18    ,            $
,       ,FA    ,T#18    ,T#14    ,            $
,       ,FM    ,Q       ,C6      ,            $
,       ,TP    ,Q       ,T#14    ,            $
,       ,FM    ,T#10    ,T#3     ,            $
,       ,FI    ,T#11    ,T#2     ,            $
,       ,TN    ,Q       ,T#19    ,            $
,       ,FA    ,T#19    ,T#15    ,            $
,       ,FM    ,Q       ,C6      ,            $
,       ,TP    ,Q       ,T#15    ,2C          $
```

```
,       ,FA   ,T#18    ,T#4     ,      $
,       ,TP   ,Q       ,T#18    ,      $
,       ,FA   ,T#19    ,T#5     ,      $
,       ,TP   ,Q       ,T#19    ,      $
,       ,TN   ,T#19    ,Q       ,      $
,       ,FP   ,T#11    ,T#16    ,      $
,       ,FI   ,T#18    ,T#10    ,      $
,       ,TP   ,Q       ,T#16    ,      $
,       ,TP   ,T#17    ,Q       ,      $
,       ,FI   ,T#18    ,T#11    ,      $
,       ,FI   ,T#19    ,T#10    ,      $
,       ,TP   ,Q       ,T#17    ,B     $
,       ,FM   ,T#17    ,T#17    ,      $
,       ,TN   ,Q       ,Q       ,      $
,       ,FI   ,T#16    ,T#16    ,      $
,       ,TP   ,Q       ,T#18    ,      $
,       ,FM   ,T#17    ,T#16    ,      $
,       ,FM   ,Q       ,C6      ,      $
,       ,TP   ,Q       ,T#19    ,B2    $
,       ,FM   ,T#9     ,T#15    ,      $
,       ,TN   ,Q       ,Q       ,      $
,       ,FI   ,T#8     ,T#14    ,      $
,       ,TP   ,Q       ,T#20    ,      $
,       ,FM   ,T#8     ,T#15    ,      $
,       ,FI   ,T#9     ,T#14    ,      $
,       ,TP   ,Q       ,T#15    ,      $
,       ,TP   ,T#20    ,T#14    ,      $
,       ,FM   ,T#11    ,T#5     ,      $
,       ,TN   ,Q       ,Q       ,      $
,       ,FI   ,T#10    ,T#4     ,      $
,       ,FM   ,Q       ,C6      ,      $
,       ,TN   ,Q       ,T#20    ,      $
,       ,FM   ,T#10    ,T#5     ,      $
,       ,FI   ,T#4     ,T#11    ,      $
,       ,FM   ,Q       ,C6      ,      $
,       ,TN   ,Q       ,T#21    ,      $
,       ,TN   ,T#21    ,Q       ,      $
,       ,FP   ,T#15    ,T#18    ,      $
,       ,FI   ,T#20    ,T#14    ,      $
,       ,TP   ,Q       ,SARG#1  ,      $
,       ,TP   ,T#19    ,Q       ,      $
,       ,FI   ,T#21    ,T#14    ,      $
,       ,FI   ,T#20    ,T#15    ,      $
,       ,TP   ,Q       ,SARG#2  ,      $
,       ,RJ   ,NBODY   ,SBODY   ,      $
,       ,FM   ,ANS     ,T#16    ,      $
,       ,FI   ,ANS#1   ,T#17    ,      $
,       ,TP   ,Q       ,A       ,      $
,       ,SJ   ,M15     ,M17     ,      $
,M15    ,TN   ,ANS     ,ANS     ,      $
```

```
,      ,TN  ,ANS#1     ,ANS#1    ,      $
,M17   ,FA  ,ANS       ,T#16     ,      $
,      ,TP  ,Q         ,SARG#2   ,      $
,      ,FA  ,ANS#1     ,T#17     ,      $
,      ,TP  ,Q         ,SARG#3   ,      $
,      ,TP  ,T#20      ,SARG     ,      $
,      ,TP  ,T#21      ,SARG#1   ,      $
,      ,TP  ,SARG#2    ,A        ,      $
,      ,ZJ  ,M20       ,M18      ,      $
,M18   ,TP  ,SARG#3    ,A        ,      $
,      ,ZJ  ,M20       ,M19      ,      $
,M19   ,TP  ,C1        ,SARG#2   ,      $
,M20   ,RJ  ,NOBODY    ,BODY     ,      $
,      ,TP  ,ANS       ,T#8      ,      $
,      ,TP  ,ANS#1     ,T#9      ,      $
,      ,FM  ,T#9       ,T#13     ,      $
,      ,TN  ,Q         ,Q        ,      $
,      ,FI  ,T#8       ,T#12     ,      $
,      ,TP  ,Q         ,T#20     ,      $
,      ,FA  ,Q         ,T#6      ,      $
,      ,TP  ,Q         ,ARG      ,      $
,      ,FM  ,T#9       ,T#12     ,      $
,      ,FI  ,T#8       ,T#13     ,      $
,      ,TP  ,Q         ,T#13     ,      $
,      ,FA  ,Q         ,T#7      ,      $
,      ,TP  ,Q         ,ARG1     ,      $
,      ,TP  ,T#20      ,T#12     ,      $
,M20A  ,RJ  ,OUT       ,IN       ,      $
,      ,TM  ,T#4       ,Q        ,      $
,      ,TM  ,T#5       ,T#14     ,      $
,      ,FA  ,Q         ,T#14     ,      $
,      ,TM  ,Q         ,T#14     ,      $
,      ,TM  ,ANS       ,Q        ,      $
,      ,TM  ,ANS1      ,T#15     ,      $
,      ,FA  ,Q         ,T#15     ,      $
,      ,TM  ,Q         ,A        ,      $
,      ,ST  ,T#14      ,A        ,      $
,      ,TJ  ,SNEXT2#1  ,N-2      ,      $
,      ,FM  ,T#8       ,C4       ,      $
,      ,TP  ,Q         ,T#8      ,      $
,      ,FM  ,T#9       ,C4       ,      $
,      ,TP  ,Q         ,T#9      ,      $
,      ,FM  ,T#12      ,C4       ,      $
,      ,TP  ,Q         ,T#12     ,      $
,      ,FM  ,T#13      ,C4       ,      $
,      ,TP  ,Q         ,T#13     ,      $
,      ,FA  ,T#6       ,T#12     ,      $
,      ,TP  ,Q         ,ARG      ,      $
,      ,FA  ,T#7       ,T#13     ,      $
,      ,TP  ,Q         ,ARG1     ,
```

```
,       ,TP    ,PAR1    ,A       ,                        $
,       ,SJ    ,5N      ,M20A    ,                        $
,5N     ,RWF1  ,C8      ,        ,                        $
,       ,MJ    ,        ,M20A    ,                        $
,       ,RP3   ,4       ,N       ,                        $
,       ,TP    ,T#2     ,T       ,                        $
,N      ,TP    ,ANS     ,T#4     ,                        $
,       ,TP    ,ANS#1   ,T#5     ,                        $
,       ,TP    ,ARG     ,T#6     ,                        $
,       ,TP    ,ARG1    ,T#7     ,                        $
,       ,FA    ,T#8     ,C1      ,                        $
,       ,TP    ,Q       ,T#10    ,                        $
,       ,TP    ,T#9     ,T#11    ,                        $
,       ,MJ    ,        ,REPEAT  ,                        $
,IN     ,RJ    ,FILL    ,FILL    ,ENTRY TO AUXLIARY       $
,       ,TP    ,DET     ,ANS     ,                        $
,       ,TP    ,DET1    ,ANS#1   ,                        $
,       ,TP    ,T#23    ,A       ,                        $
,       ,SJ    ,SKIP    ,N15     ,                        $
,N15    ,TP    ,T#23    ,T#24    ,SET INDEX               $
,       ,TP    ,C1      ,T#20    ,SET ONE                 $
,       ,TP    ,C3      ,T#21    ,                        $
,       ,SP    ,PAR2    ,15      ,                        $
,       ,TU    ,A       ,N17     ,                        $
,N16    ,RP3   ,2       ,N18     ,                        $
,N17    ,TP    ,FILL    ,T#16    ,                        $
,N18    ,FS    ,ARG     ,T#16    ,                        $
,       ,TP    ,Q       ,T#16    ,                        $
,       ,TM    ,Q       ,T#14    ,                        $
,       ,FS    ,ARG1    ,T#17    ,                        $
,       ,TP    ,Q       ,T#17    ,X-X                     $
,       ,TM    ,Q       ,A       ,                        $
,       ,TJ    ,T#14    ,N19     ,                        $
,       ,ZJ    ,N19-1   ,N21-2   ,                        $
,       ,TM    ,T#17    ,T#14    ,                        $
,N19    ,TM    ,ARG     ,T#15    ,                        $
,       ,TM    ,ARG1    ,A       ,                        $
,       ,TJ    ,T#15    ,N20     ,                        $
,       ,ZJ    ,N19A    ,N19B    ,                        $
,N19A   ,TM    ,ARG1    ,T#15    ,                        $
,N20    ,FD    ,T#14    ,T#15    ,                        $
,       ,TP    ,Q       ,T#14    ,                        $
,N19B   ,TP    ,C11     ,A       ,                        $
,       ,TJ    ,T#14    ,CONTIN  ,                        $
,       ,RPV   ,2       ,N21     ,                        $
,       ,TP    ,C3      ,ANS     ,                        $
,N21    ,RJ    ,OUT2    ,OUT1    ,                        $
,       ,MJ    ,        ,BUMPX   ,
,CONTIN,FM     ,T#21    ,T#17    ,
,       ,TN    ,Q       ,Q       ,
```

```
,        ,FI      ,T#20      ,T#16      ,                    $
,        ,TP      ,Q         ,T#14      ,                    $
,        ,FM      ,T#20      ,T#17      ,                    $
,        ,FI      ,T#21      ,T#16      ,                    $
,        ,TP      ,Q         ,T#21      ,                    $
,        ,TP      ,T#14      ,T#20      ,                    $
,        ,RA      ,N17       ,C5        ,                    $
,        ,IJ      ,T#24      ,N16       ,                    $
,        ,TP      ,DET       ,SARG      ,                    $
,        ,TP      ,DET1      ,SARG#1    ,                    $
,        ,TP      ,T#20      ,SARG#2    ,                    $
,        ,TP      ,T#21      ,SARG#3    ,                    $
,        ,RJ      ,NOBODY    ,BODY      ,                    $
,        ,TM      ,ANS       ,A         ,                    $
,        ,TJ      ,C12       ,NQ1       ,                    $
,N25     ,TM      ,T#12      ,T#14      ,                    $
,        ,TM      ,T#13      ,A         ,                    $
,        ,TJ      ,T#14      ,N1        ,                    $
,        ,TM      ,T#13      ,T#14      ,                    $
,N1      ,TM      ,T#6       ,T#15      ,                    $
,        ,TM      ,T#7       ,A         ,                    $
,        ,TJ      ,T#15      ,N2        ,                    $
,        ,TM      ,T#7       ,T#15      ,                    $
,N2      ,RS      ,T#15      ,T#14      ,F-E                 $
,N3      ,TJ      ,C7        ,N30       ,                    $
,        ,MJ      ,          ,CONVG     ,                    $
,N30     ,RJ      ,OUT2      ,OUT1      ,                    $
,OUT     ,MJ      ,          ,FILL      ,                    $
,NQ1     ,TM      ,ANS#1     ,A         ,                    $
,        ,TJ      ,C12       ,SKIP      ,                    $
,        ,MJ      ,          ,N25       ,                    $
,SKIP    ,TM      ,DET       ,A         ,                    $
,        ,TJ      ,C12       ,NQ2       ,                    $
,        ,MJ      ,          ,N25       ,                    $
,NQ2     ,TM      ,DET1      ,A         ,                    $
,        ,TJ      ,C12       ,CONVG     ,                    $
,        ,MJ      ,          ,N25       ,                    $
,OUT1    ,RA      ,T#22      ,C8        ,                    $
,        ,EJ      ,C9        ,CONVG     ,                    $
,        ,TP      ,PAR1      ,A         ,                    $
,        ,SJ      ,PU        ,OUT2      ,                    $
,PU      ,RWF1    ,C13       ,          ,                    $
,        ,TP      ,T#23      ,A         ,                    $
,        ,AT      ,C10       ,A         ,                    $
,        ,RWP1    ,A         ,          ,                    $
,        ,RWP1    ,ARG       ,          ,                    $
,        ,RWP1    ,ARG1      ,          ,                    $
,        ,RWP1    ,DET       ,          ,                    $
,        ,RWP1    ,DET1      ,          ,                    $
,        ,RWP1    ,ANS       ,          ,                    $
```

```
,       ,RWP1   ,ANS#1   ,        ,                $
,OUT2   ,MJ     ,        ,FILL    ,                $
,BUMPX  ,FA     ,ARG     ,C2      ,                $
,       ,TP     ,Q       ,ARG     ,                $
,       ,MJ     ,        ,IN      ,                $
,CONVG  ,RA     ,M3      ,C5      ,                $
,       ,RP3    ,2       ,N7      ,STORE RESULT    $
,N6     ,TP     ,ARG     ,FILL    ,                $
,N7     ,RA     ,N6      ,C10     ,                $
,       ,TP     ,PAR1    ,A       ,                $
,       ,TJ     ,N6      ,N9      ,                $
,       ,MJ     ,        ,N8      ,                $
,N9     ,RJ     ,OUT2    ,PU      ,                $
,       ,RWF1   ,C10     ,        ,SKIP A LINE     $
,N8     ,RA     ,T#23    ,C8      ,                $
,       ,EJ     ,T#25    ,EXIT1   ,                $
,       ,TP     ,PAR2    ,A       ,                $
,       ,SJ     ,N10     ,AGAIN   ,                $
,N10    ,TM     ,ARG1    ,T#14    ,                $
,       ,TM     ,ARG     ,A       ,                $
,       ,ST     ,T#14    ,A       ,                $
,       ,TJ     ,C7      ,N10#5   ,                $
,       ,MJ     ,        ,AGAIN   ,                $
,       ,TM     ,ARG1    ,A       ,                $
,       ,TJ     ,C11     ,AGAIN   ,                $
,CONJUG ,TP     ,T#26    ,A       ,                $
,       ,SJ     ,AGAIN   ,N11     ,                $
,N11    ,TN     ,C8      ,T#26    ,                $
,       ,TN     ,ARG1    ,ARG1    ,                $
,INM    ,00     ,FILL    ,FILL    ,                $
,       ,RPV    ,2       ,CONVG   ,                $
,       ,TP     ,C3      ,ANS     ,                $
,SBODY  ,TM     ,SARG#1  ,TEMP    ,                $
,       ,TM     ,SARG#2  ,A       ,                $
,       ,TJ     ,TEMP    ,SCASE2  ,                $
,       ,ZJ     ,SNEXT1  ,SZERO   ,                $
,SNEXT1 ,TP     ,A       ,ANS     ,                $
,       ,FD     ,TEMP    ,ANS     ,                $
,SNEXT2 ,TP     ,Q       ,ANS#1   ,                $
,       ,FP     ,ANS#1   ,C1      ,                $
,       ,RWSQF1 ,Q       ,SQ      ,                $
,       ,TP     ,A       ,Q       ,                $
,       ,FP     ,ANS     ,TEMP    ,                $
,       ,RS     ,Q       ,SK2     ,                $
,       ,SJ     ,SZERO   ,SNEXT3  ,                $
,SNEXT3 ,RWSQF1 ,A       ,SQ      ,                $
,       ,TP     ,SARG#1  ,A       ,                $
,       ,SJ     ,SNEXT4  ,SNEXT8  ,                $
,SNEXT4 ,TP     ,SARG#2  ,A       ,                $
,       ,SJ     ,SNEXT5  ,SNEXT6  ,                $
```

```
,SNEXT5,TN     ,SQ#3     ,ANS#1     ,                $
,      ,MJ     ,         ,SNEXT7    ,                $
,SNEXT6,TP     ,SQ#3     ,ANS#1     ,                $
,SNEXT7,RA     ,SQ#3     ,SK2       ,                $
,      ,FD     ,SARG#2   ,SQ#3      ,                $
,      ,TM     ,Q        ,ANS       ,                $
,      ,MJ     ,         ,NBODY     ,                $
,SNEXT8,TP     ,SQ#3     ,ANS       ,                $
,      ,RA     ,SQ#3     ,SK2       ,                $
,      ,FD     ,SARG#2   ,SQ#3      ,                $
,      ,TP     ,Q        ,ANS#1     ,                $
,      ,MJ     ,         ,NBODY     ,                $
,SCASE2,TP     ,TEMP     ,ANS       ,                $
,SNEXT9,FD     ,SARG#2   ,SARG#1    ,                $
,      ,MJ     ,         ,SNEXT2    ,                $
,SZERO ,RS     ,ANS1     ,A         ,                $
,      ,TP     ,A        ,ANS#1     ,                $
,NBODY ,MJ     ,         ,FILL      ,                $
,BODY  ,TM     ,SARG#2   ,A         ,                $
,      ,TM     ,SARG#3   ,TEMP      ,                $
,      ,TJ     ,TEMP     ,CASE2     ,                $
,      ,FD     ,SARG#3   ,SARG#2    ,                $
,      ,TP     ,Q        ,TEMP      ,                $
,      ,FP     ,TEMP     ,C1        ,                $
,      ,FM     ,Q        ,SARG#2    ,                $
,NEXT1 ,TP     ,Q        ,ANS#1     ,                $
,      ,TP     ,TEMP     ,Q         ,                $
,      ,FP     ,SARG#1   ,SARG      ,                $
,      ,FD     ,Q        ,ANS#1     ,                $
,      ,TP     ,Q        ,ANS       ,                $
,      ,TN     ,TEMP     ,Q         ,                $
,      ,FP     ,SARG     ,SARG#1    ,                $
,      ,FD     ,Q        ,ANS#1     ,                $
,      ,TP     ,Q        ,ANS#1     ,                $
,      ,MJ     ,         ,NOBODY    ,                $
,CASE2 ,FD     ,SARG#2   ,SARG#3    ,                $
,      ,TP     ,Q        ,TEMP      ,                $
,      ,FP     ,TEMP     ,C1        ,                $
,      ,FM     ,Q        ,SARG#3    ,                $
,NEXT2 ,TP     ,Q        ,ANS#1     ,                $
,      ,TP     ,SARG     ,Q         ,                $
,      ,FP     ,TEMP     ,SARG#1    ,                $
,      ,FD     ,Q        ,ANS#1     ,                $
,      ,TP     ,Q        ,ANS       ,                $
,      ,TN     ,SARG#1   ,Q         ,                $
,      ,FP     ,TEMP     ,SARG      ,                $
,      ,FD     ,Q        ,ANS#1     ,                $
,      ,TN     ,Q        ,ANS#1     ,                $
,NOBODY,MJ     ,         ,FILL      ,                $
,EXIT1 ,TP     ,PAR1     ,A         ,                $
```

```
,       ,TJ     ,N6      ,EXIT2   ,                      $
,       ,MJ     ,        ,EXIT    ,                      $
,EXIT2  ,RWF1   ,C14     ,        ,SKIP PAGE             $
,       ,RWF1   ,C3      ,        ,OUTPUT BIN            $
,       ,MJ     ,        ,EXIT    ,                      $
,C1     ,F      ,1       ,        ,                      $
,C2     ,F      ,.001    ,        ,E                     $
,C3     ,       ,        ,        ,ZERO                  $
,C4     ,F      ,.5      ,        ,                      $
,C5     ,       ,2       ,        ,                      $
,C6     ,F      ,2       ,        ,                      $
,C7     ,02     ,40000↑B ,        ,CONVERGENCE FACTOR    $
,C8     ,       ,        ,1       ,                      $
,C9     ,       ,        ,100     ,                      $
,C10    ,       ,        ,2       ,                      $
,C11    ,F      ,.000001 ,        ,                      $
,C12    ,F      ,1D-20   ,        ,10-20                 $
,C13    ,       ,        ,56      ,                      $
,C14    ,       ,        ,11      ,                      $
,SK2    ,00     ,10000↑B ,        ,                      $
,ANS    ,       ,        ,        ,                      $
,ANS1   ,       ,        ,        ,                      $
,SARG   ,RESERV,4        ,4       ,                      $
,T      ,RESERV,27       ,27      ,                      $
,       ,ENDSUB,         ,        ,                      $
```

RWBFF0, Floating Point Bessel Function Subroutine

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: August 25, 1957

A. Purpose:

This subroutine calculates $J_o(x)$ or $Y_o(x)$ using floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

SUB,	RWBFF0,	97
TEMPS,	0 ,	0
INOUT,	2 ,	2

2. Input

First word: x, in floating point.

Second word: α

If α is > 0, $J_o(x)$ is calculated.

If α is $= 0$, $J_o(x)$ and $Y_o(x)$ are calculated.

If α is < 0, $Y_o(x)$ is calculated

3. Output

First word - $J_o(x)$, if calculated.

Second word - $Y_o(x)$, if calculated.

4. Space required

Length of subroutine - 97 words

Temporary storage in compiled region - none

Other routines used - RWLNF2, RWCNF4, RWSQF1.

Other temporary storage - none

5. Error codes

If $x < 0$ and $Y_0(x)$ is to be calculated, the routine goes to the alarm exit. If $x = 0$ and $Y_0(x)$ is to be calculated, the routine goes to the RWLNF2 alarm exit.

C. Restrictions and Coding Information:

The routine uses 1103A built-in floating point arithmetic. If $Y_0(x)$ is to be calculated, x must be positive. The input data is not destroyed. For full accuracy the round flip-flop should be set to 0.

D. Timing:

	J_0	Y_0	J_0 and Y_0
$\|x\| \geq 3$	13	13	18
$\|x\| < 3$	4	11	11

time in milliseconds.

E. Mathematical Method:

1. Formulas

Rational approximations are used. See E. E. Allen, Analytical Approximations, Mathematical Tables and Other Aids to Computation, Vol. 6, October, 1954, pp. 240-1.

2. Error

$$|E(J_0)| < 6.10^{-8}$$

$$|E(Y_0)| < \begin{cases} 6.10^{-8} & \text{if } |Y_0(x)| \leq 1 \\ 6.10^{-8}|Y_0| & \text{if } |Y_0(x)| > 1 \end{cases}$$

The above error bounds were obtained by calculating $J_o(x)$ and $Y_o(x)$ for x in the following ranges:

.01(.01)1

1(.1)10

10(1)100

and compared for accuracy with the British Association for Advancement of Science Tables of Bessel Functions.

```
,       ,SUB    ,RWBFFO    ,97       ,    $
,       ,TEMPS  ,0         ,0        ,    $
,       ,INOUT  ,2         ,2        ,    $
,       ,MJ     ,          ,BODY     ,    $
,ALARM  ,ALARM  ,          ,         ,    $
,EXIT   ,MJ     ,          ,FILL     ,    $
,ANS1   ,       ,          ,         ,    $
,ANS2   ,       ,          ,         ,    $
,ARG1   ,       ,          ,         ,    $
,ARG2   ,       ,          ,         ,    $
,BODY   ,TM     ,ARG1      ,A        ,    $
,       ,TJ     ,K1        ,SMALL    ,    $
,       ,TP     ,A         ,ANS1     ,    $
,       ,RWSQF1,A          ,SQ       ,    $
,       ,FD     ,K1        ,ANS1     ,    $
,       ,TP     ,Q         ,ANS2     ,    $
,       ,TP     ,CON1      ,Q        ,    $
,       ,RPV    ,6         ,NEXT1    ,    $
,       ,FP     ,ANS2      ,CON1#1   ,    $
,NEXT1  ,FD     ,Q         ,SQ#3     ,    $
,       ,TP     ,Q         ,SQ#3     ,    $
,       ,TP     ,CON2      ,Q        ,    $
,       ,RPV    ,6         ,NEXT2    ,    $
,       ,FP     ,ANS2      ,CON2#1   ,    $
,NEXT2  ,FS     ,Q         ,ANS1     ,    $
,       ,TP     ,Q         ,ANS2     ,    $
,       ,TP     ,ARG2      ,A        ,    $
,       ,TJ     ,K2        ,NEXT3    ,    $
,       ,RWCNF4,Q          ,CN       ,    $
,       ,FM     ,A         ,SQ#3     ,    $
,       ,TP     ,Q         ,ANS1     ,    $
,       ,TP     ,K2        ,A        ,    $
,       ,TJ     ,ARG2      ,EXIT     ,    $
,NEXT3  ,TP     ,ARG1      ,A        ,    $
,       ,SJ     ,ALARM     ,NEXT4    ,    $
,NEXT4  ,FA     ,K3        ,ANS2     ,    $
,       ,RWCNF4,Q          ,CN       ,    $
,       ,FM     ,A         ,SQ#3     ,    $
,       ,TP     ,Q         ,ANS2     ,    $
,       ,MJ     ,          ,EXIT     ,    $
,SMALL  ,FM     ,ARG1      ,ARG1     ,    $
,       ,FD     ,Q         ,K4       ,    $
,       ,TP     ,Q         ,ANS2     ,    $
,       ,TP     ,CON3      ,Q        ,    $
,       ,RPV    ,6         ,NEXT5    ,    $
,       ,FP     ,ANS2      ,CON3#1   ,    $
,NEXT5  ,TP     ,Q         ,ANS1     ,    $
,       ,TP     ,K2        ,A        ,    $
,       ,TJ     ,ARG2      ,EXIT     ,    $
,       ,TP     ,ARG1      ,A        ,    $
,       ,RS     ,A         ,K6       ,    $
,       ,SJ     ,ALARM     ,NEXT6    ,    $
,NEXT6  ,RWLNF2,A          ,LN       ,    $
,       ,FM     ,A         ,ANS1     ,    $
```

```
,       ,FM     ,Q          ,K5       ,   $
,       ,TP     ,Q          ,LN#3     ,   $
,       ,TP     ,CON4       ,Q        ,   $
,       ,RPV    ,6          ,NEXT7    ,   $
,       ,FP     ,ANS2       ,CON4#1   ,   $
,NEXT7  ,FA     ,Q          ,LN#3     ,   $
,       ,TP     ,Q          ,ANS2     ,   $
,       ,MJ     ,           ,EXIT     ,   $
,CON1   ,F      ,.0001      ,4476     ,   $
,       ,F      ,-.0007     ,2805     ,   $
,       ,F      ,.0013      ,7237     ,   $
,       ,F      ,-.00009    ,512      ,   $
,       ,F      ,-.0055     ,2740     ,   $
,       ,F      ,-.0000     ,0077     ,   $
,       ,F      ,.7978      ,8456     ,   $
,CON2   ,F      ,-.0001     ,3558     ,   $
,       ,F      ,.0002      ,9333     ,   $
,       ,F      ,.0005      ,4125     ,   $
,       ,F      ,-.0026     ,2573     ,   $
,       ,F      ,.0000      ,3954     ,   $
,       ,F      ,.04166     ,397      ,   $
,       ,F      ,.7853      ,9816     ,   $
,CON3   ,F      ,.0002      ,100      ,   $
,       ,F      ,-.003      ,9444     ,   $
,       ,F      ,.044       ,4479     ,   $
,       ,F      ,-.316      ,3866     ,   $
,       ,F      ,1.265      ,6208     ,   $
,       ,F      ,-2.2499    ,997      ,   $
,       ,F      ,1.0        ,         ,   $
,CON4   ,F      ,-.0002     ,4846     ,   $
,       ,F      ,.0042      ,7916     ,   $
,       ,F      ,-.0426     ,1214     ,   $
,       ,F      ,.2530      ,0117     ,   $
,       ,F      ,-.7435     ,0384     ,   $
,       ,F      ,.6055      ,9366     ,   $
,       ,F      ,.3674      ,6691     ,   $
,K1     ,F      ,3.0        ,         ,   $
,K2     ,       ,           ,         ,   $
,K3     ,F      ,1.5707     ,9633     ,   $
,K4     ,F      ,9.0        ,         ,   $
,K5     ,F      ,.6366      ,1978     ,   $
,K6     ,00     ,10000↑B    ,         ,   $
,       ,ENDSUB,            ,         ,   $
```

RWBFF1, Floating Point Bessel Function Subroutine

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: August 25, 1957

A. Purpose:

This subroutine calculates J_1 (x) or Y_1 (x) using floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

```
        SUB, RWBFF1, 98
      TEMPS,   0   ,  0
      INOUT,   2   ,  2
```

2. Input

First word: x, in floating point

Second word: α

If α is >0, J_1 (x) is calculated.

If α is = 0 , J_1 (x) and Y_1 (x) are calculated.

If α is =<0, Y_1 (x) is calculated.

3. Output

First word: J_1 (x), if calculated.

Second word: Y_1 (x), if calculated.

4. Space required:

Length of subroutine - 97 words

Temporary storage in compiled region - none

Other routines used - RWLNF2, RWCNF4, RWSQF1

Other temporary storage - none

5. Error codes

If $x \leqslant -3$ and $Y_1(x)$ is to be calculated the routine goes to the alarm exit. If $-3 < x \leqslant 0$, and $Y_1(x)$ is to be calculated the routine goes to the RWLNF2 alarm exit.

C. Restrictions and Coding Information:

The routine uses 1103A built-in floating point arithmetic. If $Y_0(x)$ is to be calculated, x must be positive. The input data is not destroyed. For full accuracy the round flip-flop should be set to 0.

D. Timing:

	J_1	Y_1	J_1 and Y_1
$\lvert x\rvert \geqslant 3$	13	13	18
$\lvert x\rvert < 3$	5	12	12

time in milliseconds.

E. Mathematical Method:

1. Formulas

Rational approximations are used. See E. E. Allen, Analytical Approximations, MTAC, Vol. 6, October, 1954, pp. 240-1.

2. Error:

$$|E(J_1)| < 2.10^{-8}$$

$$|E(Y_1)| < \begin{cases} 5.10^{-8} & \text{if } |Y_1(x)| \leqslant 1 \\ 5.10^{-8}\,|Y_1(x)| & \text{if } |Y_1(x)| > 1 \end{cases}$$

The above error bounds were obtained by calculating $J_1(x)$ and $Y_1(x)$ for x in the following ranges:

.01 (.01) 1

1 (.1) 10

10 (1) 100

and compared for accuracy with the British Association for Advancement of Science, Tables of Bessel Functions.

```
,       ,SUB    ,RWBFF1   ,98        ,        $
,       ,INOUT  ,2        ,2         ,        $
,       ,TEMPS  ,0        ,0         ,        $
,       ,MJ     ,         ,START     ,        $
,ALARM  ,ALARM  ,         ,          ,        $
,EXIT   ,MJ     ,         ,FILL      ,        $
,ANS1   ,       ,         ,          ,        $
,ANS2   ,       ,         ,          ,        $
,ARG1   ,       ,         ,          ,        $
,ARG2   ,       ,         ,          ,        $
,START  ,TM     ,ARG1     ,A         ,        $
,       ,TJ     ,K1       ,SMALL     ,        $
,       ,RWSQF1,A         ,SQ        ,        $
,       ,FD     ,K1       ,SQ#4      ,        $
,       ,TP     ,Q        ,LN#3      ,        $
,       ,FP     ,CON1     ,CON1#1    ,        $
,       ,RPV    ,5        ,N1        ,        $
,       ,FP     ,LN#3     ,CON1#2    ,        $
,N1     ,FD     ,Q        ,SQ#3      ,        $
,       ,TP     ,Q        ,LN#4      ,        $
,       ,TP     ,CON2     ,Q         ,        $
,       ,RPV    ,6        ,N2        ,        $
,       ,FP     ,LN#3     ,CON2#1    ,        $
,N2     ,FA     ,Q        ,SQ#4      ,        $
,       ,TP     ,Q        ,SQ#3      ,        $
,       ,TP     ,ARG2     ,A         ,        $
,       ,SJ     ,N7       ,N3        ,        $
,N3     ,FS     ,Q        ,K2        ,        $
,       ,RWCNF4,Q         ,CN        ,        $
,       ,FM     ,A        ,LN#4      ,        $
,       ,TP     ,ARG1     ,A         ,        $
,       ,SJ     ,N4       ,N5        ,        $
,N4     ,TN     ,Q        ,Q         ,        $
,N5     ,TP     ,Q        ,ANS1      ,        $
,N6     ,TP     ,ARG2     ,A         ,        $
,       ,ZJ     ,EXIT     ,N7        ,        $
,N7     ,RWCNF4,SQ#3      ,CN        ,        $
,       ,FM     ,A        ,LN#4      ,        $
,       ,TN     ,Q        ,ANS2      ,        $
,       ,TP     ,ARG1     ,A         ,        $
,       ,SJ     ,ALARM    ,EXIT      ,        $
 ,SMALL,FD,A,K1,     $
,       ,TP     ,Q        ,CN#3      ,        $
,       ,FM     ,Q        ,CN#3      ,        $
,       ,TP     ,Q        ,CN#3      ,        $
,       ,FP     ,CON3     ,CON3#1    ,        $
,       ,RPV    ,5        ,N8        ,        $
,       ,FP     ,CN#3     ,CON3#2    ,        $
,N8     ,FM     ,Q        ,ARG1      ,        $
,       ,TP     ,Q        ,ANS1      ,        $
,       ,RS     ,A        ,A         ,        $
,       ,TJ     ,ARG2     ,EXIT      ,        $
,       ,TP     ,ARG1     ,A         ,        $
,       ,RS     ,A        ,K3        ,        $
```

```
,       ,RWLNF2,A         ,LN        ,                                        $
,       ,FM     ,A        ,ANS1      ,                                        $
,       ,FM     ,Q        ,K4        ,                                        $
,       ,TP     ,Q        ,LN#3      ,                                        $
,       ,TP     ,CON4     ,Q         ,                                        $
,       ,RPV    ,6        ,N9        ,                                        $
,       ,FP     ,CN#3     ,CON4#1    ,                                        $
,N9     ,FD     ,Q        ,ARG1      ,                                        $
,       ,FA     ,Q        ,LN#3      ,                                        $
,       ,TP     ,Q        ,ANS2      ,                                        $
,       ,MJ     ,         ,EXIT      ,                                        $
,CON1   ,F      ,-.0002   ,0033      ,                                        $
,       ,F      ,#.0011   ,3653      ,                                        $
,       ,F      ,-.0024   ,9511      ,                                        $
,       ,F      ,.0001    ,7105      ,                                        $
,       ,F      ,.0165    ,9667      ,                                        $
,       ,F      ,.0000    ,0156      ,                                        $
,       ,F      ,.7978    ,8456      ,                                        $
,CON2   ,F      ,-.0002   ,9166      ,                                        $
,       ,F      ,#.0007   ,9824      ,                                        $
,       ,F      ,#.0007   ,4348      ,                                        $
,       ,F      ,-.0063   ,7879      ,                                        $
,       ,F      ,#.0000   ,5650      ,                                        $
,       ,F      ,#.1249   ,9612      ,                                        $
,       ,F      ,-.7853   ,9816      ,                                        $
,CON3   ,F      ,.0000    ,1109      ,                                        $
,       ,F      ,-.0003   ,1761      ,                                        $
,       ,F      ,.0044    ,3319      ,                                        $
,       ,F      ,-.0395   ,4289      ,                                        $
,       ,F      ,.2109    ,3573      ,                                        $
,       ,F      ,-.5624   ,9985      ,                                        $
,       ,F      ,.5000    ,0000      ,                                        $
,CON4   ,F      ,.0027    ,873       ,                                        $
,       ,F      ,-.0400   ,976       ,                                        $
,       ,F      ,.3123    ,951       ,                                        $
,       ,F      ,-1.316   ,4827      ,                                        $
,       ,F      ,2.168    ,2709      ,                                        $
,       ,F      ,.221     ,2091      ,                                        $
,       ,F      ,-.6366   ,198       ,                                        $
,K1     ,F      ,3.0      ,          ,                                        $
,K2     ,F      ,1.57079  ,63268     ,                                        $
,K3     ,00     ,10000↑B  ,          ,                                        $
,K4     ,F      ,.6366    ,1977      ,                                        $
,       ,ENDSUB,          ,          ,                                        $
```

RWGQF0, Gaussian Quadrature Subroutine

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: August 15, 1957

A. Purpose:

This subroutine evaluates the definite integral, $\int_a^b f(x)dx$, using 16 point Gaussian Quadrature. Single precision floating point arithmetic is used.

B. Usage:

1. Specifications. Standard USE subroutine using built in floating point.

```
       SUB, RWGQF0, 73
     TEMPS,    0  , 0
     INOUT,    3  , 1
```

2. Input

First word - 00, n, AUX

Second word - a, Lower limit in floating point.

Third word - b, Upper limit in floating point.

n is the number of subintervals of the basic interval of integration. In each of these subintervals a 16 point Gaussian Quadrature will be performed. AUX is the address of an auxiliary routine which calculates the integrand f(x). The argument x will be in Q upon entrance to the auxiliary routine. When the auxiliary routine returns control to RWGQF0, f(x) should be in Q. Entrance to the auxiliary routine is made with the command RJ, AUX, AUX+1. Therefore, the first and second words of the auxiliary routine must be the exit and the entrance respectively. x is in floating point and f(x) must be in floating point.

3. Output

First word - $\int_a^b f(x)dx$. The result is also in Q.

4. Space required:

Length of subroutine - 73 cells

Temporary storage in compiled region - none

Other routines used - auxiliary routine prepared by programmer

Other temporary storage - none

5. Error codes - The alarm exit is not used.

C. Restrictions and Coding Information:

n must be greater than, or equal to one. Since machine floating point is used the programmer must insure that f(x) and

$\int_a^b f(x)dx$ is within the floating point range.

D. Timing:

2.8 + n(17.5 + 16t) milliseconds where t is the time of a single functional evaluation in milliseconds and n is defined above.

E. Mathematical Method:

$$\int_a^b f(x)dx \cong \sum_{i=1}^{n} \sum_{k=1}^{16} \left(\frac{b-a}{2n}\right) c_k \cdot f\left[d_k \left(\frac{b-a}{2n}\right) + \left(\frac{b-a}{2n}\right)(2i-1) + a\right]$$

Heuristically the integration is equivalent to fitting n 31st degree polynomials end-to-end to the given integrand, f(x), and integrating them exactly. The $\{d_k\}$ are the roots of $P_{16}(x)$, the Legendre polynomial of degree 16, and the $\{c_k\}$ are the corresponding weights. A bound for the error of this integration formula is very difficult to calculate. Analytically the error, E, is given by:

$$E = \frac{f^{(32)}(\xi)}{(32!)K^2}$$

where $$K = \frac{n(32!)\sqrt{33}}{[16!]^2 \left(\frac{b-a}{n}\right)^{n+1/2}}$$ and

ξ is some point in (a, b). See [1] and [2]. For another method of error estimation applicable to analytic functions see [3]. If the function does not have a 32nd derivative, the error analysis becomes more difficult.

This routine may be used for m dimensional integration by placing m copies of the subroutine in memory. For example suppose that the double integral,

$\int_a^b \int_c^d f(x, y)dx\, dy$ is desired. This may be written as

$\int_a^b \left[\int_c^d f(x, y)\, dx\right] dy$. Now let $g(y) = \int_c^d f(x, y)dx$. Then the integral becomes $\int_a^b g(y)dy$. The evaluation of g(y) in the auxiliary routine requires a second copy of RWGQF0 in order to evaluate for a given y, $\int_c^d f(x, y)dx$. This method may be extended to higher dimensions.

Some numerical examples of quadrature using this routine follow:

A. $I(x) = \int_0^x \cos t\, dt = \sin x$

x	I(x)=sin x	N=1	N=2	N=4
10.0	-.54402111	-.54402108	-.54402113	-.54402110
20.0	.91294525	.91294508	.91294540	.91294520
30.0	.98803162	-.98803180	-.98803157	-.98803157
40.0	.74511316	.74467116	.74511254	.74511290
50.0	-.26237486	-.52019490	-.26237506	-.26237451

N=8	N=16	N=32
-.54402111	-.54402110	-.54402111
.91294526	.91294526	.91294522
-.98803159	-.98803163	-.98803156
.74511269	.74511316	.74511318
-.26237416	-.26237488	-.26237475

The first column is x. The second column is sin x evaluated by the sine routine. The remaining columns are the results of numerical integration using RWGQF0 for N = 1, 2, 4, 8, 16, and 32, respectively.

The error for n = 32 is due to rounding. Note that the results for N = 1 are very good up to x = 30, and degrade very rapidly after that.

B. $I(x) = 2\sqrt{x} = \int_0^x \frac{dt}{\sqrt{t}}$

x	$I(x)=2\sqrt{x}$	N=1	N=2	N=4
10.0	6.32455	6.15767	6.20655	6.24112
20.0	8.94427	8.70827	8.77739	8.82627
30.0	10.9545	10.6654	10.7501	10.8099
40.0	12.6491	12.3153	12.4131	12.4822
50.0	14..1421	13.7690	13.8783	13.9556

N=8	N=16	N=32
6.26555	6.28284	6.29505
8.86083	8.88527	8.90255
10.8523	10.8822	10.9034
12.5311	12.5657	12.5901
14.0102	14.0488	14.0762

The first column is x. The second column is $2\sqrt{x}$ evaluated using the square-root routine. The remaining columns are the results of numerical integration using RWGQF0 for N = 1, 2, 4, 8, 16, and 32, respectively. These results were rounded to six places because the accuracy is so low that the last two places are not significant. This example indicates clearly the problems involved in integrating a function with a singularity at the end points.

[1] Hildebrand, Introduction to Numerical Analysis, McGraw-Hill, 1956. pp. 312-367

[2] Lowan, Davids, and Levenson, Tables of the Zeroes of the Legendre Polynomials of Order 1-16 and the Weight Coefficients for Gauss' Mechanical Quadrature Formula, Bull. American Math. Society, Vol. 48, No. 10, pp. 739-743, October, 1942.

[3] Davis and Rabinowitz, On the Estimation of Quadrature Errors for Analytic Functions, MTAC, October, 1954, pp. 193-203.

```
,       ,SUB    ,RWGQF0    ,73       ,     $
,       ,INOUT  ,3         ,1        ,     $
,       ,MJ     ,          ,START    ,     $
,       ,ALARM  ,          ,         ,     $
,EXIT   ,MJ     ,          ,FILL     ,     $
,ANS    ,       ,          ,         ,     $
,ARG    ,RESERV,3          ,3        ,     $
,START  ,TV     ,ARG       ,AUX1     ,     $
,       ,SP     ,ARG       ,15       ,     $
,       ,TU     ,A         ,AUX1     ,     $
,       ,LT0    ,6         ,TEMP#1   ,     $
,       ,LT0    ,0         ,TEMP     ,     $
,       ,RA     ,AUX1      ,K1       ,     $
,       ,TP     ,A         ,AUX2     ,     $
,       ,RS     ,ANS       ,A        ,     $
,       ,RS     ,TEMP      ,K1       ,     $
,       ,NP     ,TEMP#1    ,K2       ,     $
,       ,FS     ,ARG#2     ,ARG#1    ,     $
,       ,FD     ,Q         ,TEMP#1   ,     $
,       ,TP     ,Q         ,TEMP#2   ,     $
,       ,FD     ,Q         ,K3       ,     $
,       ,TP     ,Q         ,TEMP#3   ,     $
,       ,FA     ,Q         ,ARG#1    ,     $
,LOOP1  ,TP     ,Q         ,TEMP#4   ,     $
,       ,TP     ,K4        ,TRANS    ,     $
,LOOP2  ,RPB    ,2         ,NEXT     ,     $
,TRANS  ,TP     ,CON       ,TEMP#5   ,     $
,NEXT   ,FM     ,TEMP#5    ,TEMP#3   ,     $
,       ,TP     ,Q         ,TEMP#5   ,     $
,       ,FA     ,Q         ,TEMP#4   ,     $
,AUX1   ,RJ     ,FILL      ,FILL     ,     $
,       ,TP     ,Q         ,TEMP#1   ,     $
,       ,FS     ,TEMP#4    ,TEMP#5   ,     $
,AUX2   ,RJ     ,FILL      ,FILL     ,     $
,       ,FA     ,Q         ,TEMP#1   ,     $
,       ,FP     ,TEMP#6    ,ANS      ,     $
,       ,TP     ,Q         ,ANS      ,     $
,       ,RA     ,TRANS     ,K5       ,     $
,       ,TJ     ,K6        ,LOOP2    ,     $
,       ,FA     ,TEMP#4    ,TEMP#2   ,     $
,       ,IJ     ,TEMP      ,LOOP1    ,     $
,       ,FM     ,ANS       ,TEMP#3   ,     $
,       ,TP     ,Q         ,ANS      ,     $
,       ,MJ     ,          ,EXIT     ,     $
,K1     ,00     ,          ,1        ,     $
,K2     ,23     ,30000↑B   ,         ,     $
,K3     ,F      ,2.0       ,         ,     $
,K4     ,TP     ,CON       ,TEMP#5   ,     $
,K5     ,00     ,00002↑B   ,         ,     $
,K6     ,TP     ,CON#15    ,TEMP#5   ,     $
,CON    ,F      ,0.09501250,9837637  ,     $
,       ,F      ,0.18945061,0455069  ,     $
,       ,F      ,0.28160355,0779259  ,     $
,       ,F      ,0.18260341,5044924  ,     $
```

```
,        ,F      ,0.458016777,657227  ,                                      $
,        ,F      ,0.169156519,395003  ,                                      $
,        ,F      ,0.6178762,44402644  ,                                      $
,        ,F      ,0.14959598,8816577  ,                                      $
,        ,F      ,0.75540440,8355003  ,                                      $
,        ,F      ,0.12462897,1255534  ,                                      $
,        ,F      ,0.865631202,387832  ,                                      $
,        ,F      ,0.0951585,11682493  ,                                      $
,        ,F      ,0.9445750,23073233  ,                                      $
,        ,F      ,0.0622535,23938648  ,                                      $
,        ,F      ,0.9894009,34991650  ,                                      $
,        ,F      ,0.0271524,59411754  ,                                      $
,TEMP    ,RESERV,7         ,7         ,                                      $
,        ,ENDSUB,          ,          ,                                      $
```

RWCSQ 1, Complex, Floating Point Square Root

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: August 15, 1957

A. Purpose:

This subroutine evaluates the square root of a complex floating point number, a + bi.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

SUB,	RWCSQ 1,	47,
TEMPS,	1 ,	2,
INOUT,	2 ,	2,

2. Input

First word - a

Second word - b

where a and b are the real and imaginary parts, respectively, of the given complex number.

3. Output

First word - x

Second word - y

where x and y are the real and imaginary parts, respectively, of the complex square root.

4. Space required

   Length of subroutine - 47 cells

   Temporary storage in compiled region - 1 cell

   Subsidiary routine - RWSQF 1

5. Error codes - The alarm exit is not used.

C. Restrictions and Coding Information:

Data must be in standard USE complex, floating point representation. The 1103A built in floating point package is used. RWSQF1 is used as a subsidiary subroutine. The input data is not destroyed.

D. Timing:

8.4 milliseconds per square root, which includes the evaluation of two real square roots.

E. Mathematical Method:

Given the number a + bi, and its square root x + iy. Then let $r = \max(|a|, |b|)$, $s = \min(|a|, |b|)$ and define

$$N = \sqrt{\frac{1}{2}\left(r\sqrt{1 + \left(\frac{s}{r}\right)^2} + |a|\right)}.$$

If $a \geq 0$, then $x = N$, $y = \frac{b}{2x}$. If $a < 0$, then $y = (\mathrm{sgn}\,b)N$, and $x = \frac{b}{2y}$ where $\mathrm{sgn}\,b = \begin{cases} 1 \text{ if } b \geq 0 \\ -1 \text{ if } b < 0 \end{cases}$. The square root, x + iy, is in the right half-plane except when a = 0, in which case it is on the non-negative imaginary axis. For the particular case $a = b = 0$, the routine gives $x = y = 0$. Zero is also given as an answer if $a = 0$ and $|b| < 2^{-128}$ where a and b are the real and imaginary parts of the given complex number.

```
,        ,SUB    ,RWCSQ1   ,47       ,          $
,        ,TEMPS  ,1        ,2        ,          $
,        ,INOUT  ,2        ,2        ,          $
,        ,MJ     ,         ,BODY     ,          $
,ALARM   ,ALARM  ,         ,         ,          $
,EXIT    ,MJ     ,         ,FILL     ,          $
,ANS1    ,       ,         ,         ,          $
,ANS2    ,       ,         ,         ,          $
,ARG1    ,       ,         ,         ,          $
,ARG2    ,       ,         ,         ,          $
,BODY    ,TM     ,ARG1     ,TEMP     ,          $
,        ,TM     ,ARG2     ,A        ,          $
,        ,TJ     ,TEMP     ,CASE2    ,          $
,        ,ZJ     ,NEXT1    ,ZERO     ,          $
,NEXT1   ,TP     ,A        ,ANS1     ,          $
,        ,FD     ,TEMP     ,ANS1     ,          $
,NEXT2   ,TP     ,Q        ,ANS2     ,          $
,        ,FP     ,ANS2     ,K1       ,          $
,        ,RWSQF1,Q         ,SQ       ,          $
,        ,TP     ,A        ,Q        ,          $
,        ,FP     ,ANS1     ,TEMP     ,          $
,        ,RS     ,Q        ,K2       ,          $
,        ,SJ     ,ZERO     ,NEXT3    ,          $
,NEXT3   ,RWSQF1,A         ,SQ       ,          $
,        ,TP     ,ARG1     ,A        ,          $
,        ,SJ     ,NEXT4    ,NEXT8    ,          $
,NEXT4   ,TP     ,ARG2     ,A        ,          $
,        ,SJ     ,NEXT5    ,NEXT6    ,          $
,NEXT5   ,TN     ,SQ#3     ,ANS2     ,          $
,        ,MJ     ,         ,NEXT7    ,          $
,NEXT6   ,TP     ,SQ#3     ,ANS2     ,          $
,NEXT7   ,RA     ,SQ#3     ,K2       ,          $
,        ,FD     ,ARG2     ,SQ#3     ,          $
,        ,TM     ,Q        ,ANS1     ,          $
,        ,MJ     ,         ,EXIT     ,          $
,NEXT8   ,TP     ,SQ#3     ,ANS1     ,          $
,        ,RA     ,SQ#3     ,K2       ,          $
,        ,FD     ,ARG2     ,SQ#3     ,          $
,        ,TP     ,Q        ,ANS2     ,          $
,        ,MJ     ,         ,EXIT     ,          $
,CASE2   ,TP     ,TEMP     ,ANS1     ,          $
,NEXT9   ,FD     ,ARG2     ,ARG1     ,          $
,        ,MJ     ,         ,NEXT2    ,          $
,ZERO    ,RS     ,ANS1     ,A        ,          $
,        ,TP     ,A        ,ANS2     ,          $
,        ,MJ     ,         ,EXIT     ,          $
,K1      ,F      ,1.0000   ,         ,          $
,K2      ,00     ,10000↑B  ,         ,          $
,        ,ENDSUB,          ,         ,          $
```

RWMVM2, Complex, Floating Point Matrix-Vector Multiplication

Programmed by: F. W. Blackwell, The Ramo-Wooldridge Corporation

Date: August 20, 1957

A. Purpose:

This subroutine computes the product of the square matrix $A - \alpha I$ and the vector x. The computation is performed in complex floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

```
SUB    , RWMVM2, 86
TEMPS,    5    , 0
 INOUT,   4    , 0
```

2. Input.

First word: α_R

Second word: α_I

Third word: 00, A, x

Fourth word: 00, Y, N

where α_R = floating point, real component of α

α_I = floating point, imaginary component of α

A = address of the first cell in the region of $2n^2$ cells where the matrix A is stored, row by row. A may be either in core storage or on the magnetic drum. It is assumed throughout this subroutine that each element is stored in two consecutive cells, the real part immediately preceding the imaginary part.

X = address of the first cell in a region of 4n cells, the first 2n of which contain the vector X, and the second 2n of which are successively used for temporary storage of the rows of the matrix. This region should be in core storage for efficient operation.

Y = address of the first cell in the region of 2n cells where the resultant vector Y is to be stored.

N = order of the matrix A

3. Output.

The vector Y which equals $(A - \alpha I) X$ is stored in a region of 2n cells beginning at location Y, as indicated above.

4. Space required.

Length of subroutine: 86 cells

Temporary storage in compiled region: 5 cells

Other routines used: none

Other temporary storage: data, results, and temporary storage occupy an additional $2n^2 + 6n$ cells, $2n^2$ of which (the matrix) may be stored on the magnetic drum.

5. Error Codes.

The alarm exit is not used by this subroutine.

C. Restrictions and Coding Information

Data must in standard USE floating point representation. The program is self-contained, performing its own complex arithmetic. The 1103A built-in floating point is used. The subroutine does not destroy the input data, the matrix, or the vector. If everything (including the matrix) is in core storage of 4096 cells, the limitation on the order of the matrix A is $N \leqslant 43$. Since the matrix may be stored on the drum, N may be as large as 90.

D. Timing

$2.5n^2 + 21.5n + 1.8$ milliseconds. If the matrix is on the drum, add 17n milliseconds to this time.

E. Mathematical Method

The matrix is brought in a row at a time to temporary storage and the complex inner product of this row and the given vector is computed to form an element of the resultant vector. Representative times obtained by trial on the 1103A for matrices having random elements with a range of approximately 10^4 and stored on the drum are as follows:

Order of Matrix	Time in Seconds
10	0.37
15	1.01
20	1.37
25	1.74
30	3.08
50	6.88
80	19.14

```
,       ,SUB    ,RWMVM2  ,86     ,                                    $
,       ,TEMPS  ,5       ,0      ,                                    $
,       ,INOUT  ,4       ,0      ,                                    $
,       ,MJ     ,        ,BODY   ,                                    $
,       ,ALARM  ,        ,       ,                                    $
,EXIT   ,MJ     ,        ,FILL   ,                                    $
,ARG    ,RESERV,4        ,4      ,                                    $
,BODY   ,TP     ,C3      ,T3     ,                                    $
,       ,TV     ,ARG#3   ,T3     ,N IN T3 %TEMPORARY↑                 $
,       ,TP     ,T3      ,T4     ,N IN T4 %TEMPORARY↑                 $
,       ,RS     ,T3      ,C4     ,N-1 IN T3                           $
,       ,SP     ,T4      ,1      ,2N IN A                             $
,       ,LQ     ,T4      ,16     ,2N IN U ADDRESS OF T4               $
,       ,RA     ,A       ,ARG#2  ,X#2N IN V ADDRESS OF A              $
,       ,TV     ,A       ,R8     ,                                    $
,       ,TV     ,A       ,R10    ,                                    $
,       ,TV     ,A       ,R2     ,                                    $
,       ,TV     ,A       ,R5     ,                                    $
,       ,RA     ,A       ,C4     ,X#2N#1 IN V ADDRESS OF A            $
,       ,TV     ,A       ,R12    ,                                    $
,       ,TV     ,A       ,R1     ,                                    $
,       ,TV     ,A       ,R4     ,                                    $
,       ,LA     ,A       ,15     ,X#2N#1 IN U ADDRESS OF A            $
,       ,TU     ,A       ,R11    ,                                    $
,       ,RS     ,A       ,C6     ,X#2N IN U ADDRESS OF A              $
,       ,TU     ,A       ,R9     ,                                    $
,       ,RS     ,A       ,T4     ,X IN U ADDRESS OF A                 $
,       ,TU     ,A       ,R2     ,                                    $
,       ,TU     ,A       ,R4     ,                                    $
,       ,RA     ,A       ,C6     ,X#1 IN U ADDRESS OF A               $
,       ,TU     ,A       ,R1     ,                                    $
,       ,TU     ,A       ,R5     ,                                    $
,       ,TU     ,ARG#2   ,R8     ,                                    $
,       ,SP     ,ARG#3   ,57     ,Y IN V ADDRESS OF A                 $
,       ,TV     ,A       ,R6     ,                                    $
,       ,RA     ,A       ,C4     ,Y#1 IN V ADDRESS OF A               $
,       ,TV     ,A       ,R7     ,                                    $
,       ,TP     ,T4      ,T5     ,                                    $
,       ,LQ     ,T5      ,21     ,                                    $
,       ,TU     ,T4      ,T5     ,2N---2N IN T5                       $
,       ,RA     ,R13     ,T4     ,SET REPEAT INSTRUCTION              $
,       ,TP     ,T3      ,T2     ,SET INDEX FOR LARGE LOOP            $
,R13    ,RP3    ,0       ,R9     ,FILL 3--2N IN U                     $
,R8     ,TP     ,FILL    ,FILL   ,FILL M IN U, X#2N IN V              $
,R9     ,FS     ,FILL    ,ARG    ,FILL X#2N IN U                      $
,R10    ,TP     ,Q       ,FILL   ,FILL X#2N IN V                      $
,R11    ,FS     ,FILL    ,ARG#1  ,FILL X#2N#1 IN U                    $
,R12    ,TP     ,Q       ,FILL   ,FILL X#2N#1 IN V                    $
,       ,TP     ,T3      ,T1     ,SET INDEX FOR SMALL LOOP 1          $
,       ,TP     ,C3      ,Q      ,CLEAR Q                             $
,R3     ,TN     ,Q       ,Q      ,                                    $
,R1     ,FI     ,FILL    ,FILL   ,FILL X#1 IN U, X#2N#1 IN V          $
,       ,TN     ,Q       ,Q      ,                                    $
,R2     ,FI     ,FILL    ,FILL   ,FILL X IN U, X#2N IN V              $
```

```
,      ,RA     ,R1     ,C1     ,INCREASE U BY 2, V BY 2         $
,      ,RA     ,R2     ,C1     ,INCREASE U BY 2, V BY 2         $
,      ,IJ     ,T1     ,R3     ,TEST END OF SMALL LOOP 1        $
,R6    ,TP     ,Q      ,FILL   ,FILL Y IN V                     $
,      ,RA     ,R6     ,C2     ,INCREASE V BY 2                 $
,      ,RS     ,R1     ,T5     ,RESET U AND V                   $
,      ,RS     ,R2     ,T5     ,RESET U AND V                   $
,      ,TP     ,T3     ,T1     ,SET INDEX FOR SMALL LOOP 2      $
,      ,TP     ,C3     ,Q      ,CLEAR Q                         $
,R4    ,FI     ,FILL   ,FILL   ,FILL X IN U, X#2N#1 IN V        $
,R5    ,FI     ,FILL   ,FILL   ,FILL X#1 IN U, X#2N IN V        $
,      ,RA     ,R4     ,C1     ,INCREASE U BY 2, V BY 2         $
,      ,RA     ,R5     ,C1     ,INCREASE U BY 2, V BY 2         $
,      ,IJ     ,T1     ,R4     ,TEST END OF SMALL LOOP 2        $
,R7    ,TP     ,Q      ,FILL   ,FILL Y#1 IN V                   $
,      ,RA     ,R7     ,C2     ,INCREASE V BY 2                 $
,      ,RS     ,R4     ,T5     ,RESET U AND V                   $
,      ,RS     ,R5     ,T5     ,RESET U AND V                   $
,      ,RA     ,R8     ,T4     ,INCREASE U BY 2N                $
,      ,RA     ,R9     ,C5     ,INCREASE U BY 2                 $
,      ,RA     ,R10    ,C2     ,INCREASE V BY 2                 $
,      ,RA     ,R11    ,C5     ,INCREASE U BY 2                 $
,      ,RA     ,R12    ,C2     ,INCREASE V BY 2                 $
,      ,IJ     ,T2     ,R13    ,TEST END OF LARGE LOOP          $
,      ,RS     ,R13    ,T4     ,RESET REPEAT INSTRUCTION        $
,      ,MJ     ,       ,EXIT   ,                                $
,C1    ,0      ,2      ,2      ,CONSTANTS                       $
,C2    ,0      ,0      ,2      ,                                $
,C3    ,0      ,0      ,0      ,                                $
,C4    ,0      ,0      ,1      ,                                $
,C5    ,0      ,2      ,0      ,                                $
,C6    ,0      ,1      ,0      ,                                $
,      ,ENDSUB,        ,       ,                                $
```

RWDET5, Complex, Floating Point Determinant Evaluation for Tri-diagonal Matrices

Programmed by: Werner L. Frank, The Ramo-Wooldridge Corporation

Date: August 15, 1957

A. Purpose:

This subroutine evaluates the determinant of the matrix $A - \lambda I$, where A is a tri-diagonal matrix of order N. A tri-diagonal matrix, or Jacobi matrix is defined to be one for which $a_{ij} = 0$ for $|i - j| > 1$. For the special case $\lambda = 0$ one obtains the determinant of A. The computation is performed in complex floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

```
     SUB,   RWDET 5, 59,
   TEMPS,   7      , 0 ,
   INOUT,   3      , 2 ,
```

2. Input

First word - λ_R

Second word - λ_I

Third word - 00, MATLOC, N

where λ_R = floating point, real component of λ

λ_I = floating point, imaginary component of λ

N = order of the Matrix A

MATLOC = address of the first cell of a region of 4N-2 cells where the matrix A is stored. Only the elements a_{ij} of A, $|i - j| \leq 1$ are stored. They are arranged in the order of a_{ii} $(i = 1 \ldots N)$ in the first 2N cells and the products $a_{i,\,i+1} \cdot a_{i+1,\,i}$ $(i = 1 \ldots N-1)$ in the next 2(N-1) cells.

3. Output

First word - floating point, real component of $|A - \lambda I|$

Second word - floating point, imaginary component of $|A - \lambda I|$

4. Space required

Length of subroutine - 59 cells

Temporary storage in compiled region - 7 cells

Other temporary storage - (4N-2) cells

5. Error codes - Alarm exit is not used

C. Restrictions and Coding Information:

Data must be in standard USE complex floating point representation. The program is self contained performing its own complex arithmetic. The 1103A built-in floating point package is used. The subroutine does not destroy the input data. For core storage of 4096 cells, the limitations on the order of the matrix A is $N \leq 1000$.

D. Timing:

(6 N + 1) milliseconds where N is the order of the matrix

E. Mathematical Method:

Given the tri-diagonal matrix A

$$A = \begin{bmatrix} a_1 & b_1 & 0 & 0 & . & . & . & 0 \\ c_1 & a_2 & b_2 & 0 & . & . & . & 0 \\ 0 & c_2 & a_3 & b_3 & & . & . & 0 \\ . & & & & & & & . \\ . & & & & & & & . \\ . & & & & & & & . \\ 0 & & & & & c_{n-2} & a_{n-1} & b_{n-1} \\ 0 & & & & & 0 & c_{n-1} & a_n \end{bmatrix}$$

then the determinant $P_n(\lambda)$ of $(A - \lambda I)$ is obtained by application of the recursion formula

$$P_i(\lambda) = (a_i - \lambda) P_{i-1}(\lambda) - b_{i-1} \cdot c_{i-1} \cdot P_{i-2}(\lambda) \quad i = 1, 2, \ldots, n$$

where $P_{-1} = 0$ and $P_o = 1$

The routine assumes that the products $b_i \cdot c_i$ have been formed prior to entry and are stored in order immediately following the a_i.

```
    ,          ,SUB     ,RWDET5    ,59        ,                              $
,,TEMPS,7,0,$
    ,          ,INOUT   ,3         ,2         ,                              $
    ,ENTRY     ,MJ      ,          ,STAR      ,                              $
    ,ALARM     ,ALARM   ,          ,          ,                              $
,EXIT,MJ,,,$
    ,RES       ,RESERV  ,2         ,2         ,                              $
    ,PAR       ,RESERV  ,3         ,3         ,                              $
    ,STAR      ,TU      ,PAR#2     ,NEX       ,                              $
    ,          ,TU      ,PAR#2     ,NEX1      ,                              $
    ,          ,SP      ,PAR#2     ,16        ,                              $
,,ST,TWO,A,$
    ,          ,AT      ,PAR#2     ,A         ,                              $
    ,          ,TU      ,A         ,NEX2      ,                              $
    ,          ,TU      ,A         ,NEX3      ,                              $
    ,          ,TU      ,A         ,NEX4      ,                              $
    ,          ,TU      ,A         ,NEX5      ,                              $
    ,          ,RA      ,NEX1      ,ONEB      ,                              $
    ,          ,RA      ,NEX2      ,ONEB      ,                              $
    ,          ,RA      ,NEX4      ,ONEB      ,                              $
,,RP1,7,NEX-3,$
    ,          ,TP      ,ZERO      ,P         ,                              $
    ,          ,TP      ,ONE       ,P#2       ,                              $
    ,          ,TV      ,PAR#2     ,P#6       ,                              $
    ,          ,MJ      ,          ,END       ,                              $
    ,NEX       ,FS      ,FILL      ,PAR       ,                              $
    ,          ,TP      ,Q         ,P#4       ,                              $
    ,NEX1      ,FS      ,FILL      ,PAR#1     ,                              $
    ,          ,TP      ,Q         ,P#5       ,                              $
    ,NEX2      ,FM      ,FILL      ,P#1       ,                              $
    ,          ,TN      ,Q         ,Q         ,                              $
    ,NEX3      ,FI      ,FILL      ,P         ,                              $
    ,          ,TN      ,Q         ,RES       ,                              $
    ,          ,TN      ,P#5       ,Q         ,                              $
    ,          ,FP      ,P#3       ,RES       ,                              $
    ,          ,FI      ,P#4       ,P#2       ,                              $
    ,          ,TP      ,Q         ,RES       ,                              $
    ,NEX4      ,FM      ,FILL      ,P         ,                              $
    ,NEX5      ,FI      ,FILL      ,P#1       ,                              $
    ,          ,TN      ,Q         ,Q         ,                              $
    ,          ,FI      ,P#4       ,P#3       ,                              $
    ,          ,FI      ,P#5       ,P#2       ,                              $
    ,          ,TP      ,Q         ,RES#1     ,                              $
    ,          ,TP      ,P#2       ,P         ,                              $
    ,          ,TP      ,P#3       ,P#1       ,                              $
    ,          ,TP      ,RES       ,P#2       ,                              $
    ,          ,TP      ,RES#1     ,P#3       ,                              $
    ,          ,RA      ,NEX       ,TWO       ,                              $
    ,          ,RA      ,NEX1      ,TWO       ,                              $
    ,          ,RA      ,NEX2      ,TWO       ,                              $
    ,          ,RA      ,NEX3      ,TWO       ,                              $
    ,          ,RA      ,NEX4      ,TWO       ,                              $
    ,          ,RA      ,NEX5      ,TWO       ,                              $
    ,END       ,IJ      ,P#6       ,NEX       ,                              $
```

```
,       ,MJ     ,           ,EXIT      ,                                             $
,ZERO   ,00     ,           ,          ,                                             $
,ONE    ,F      ,1          ,          ,                                             $
,TWO    ,       ,2          ,          ,                                             $
,ONEB   ,       ,1          ,          ,                                             $
,       ,ENDSUB,            ,          ,                                             $
```

RWCDV1, Complex Floating Point Division

Programmed by: David G. Cantor, The Ramo-Wooldridge Corporation

Date: September 1, 1957

A. Purpose:

Given the complex floating point numbers a + bi and c + di this subroutine calculates

$$p + qi = \frac{a+bi}{c+di}$$

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

SUB,	RWCDV1,	41
TEMPS,	1 ,	0
INOUT,	4 ,	2

2. Input

First word - a

Second word - b

Third word - c

Fourth word - d

a and b are the real and imaginary parts of the numerator, respectively; c and d are the real and imaginary parts of the denominator, respectively.

3. Output

   First word - p

   Second word - q

   p and q are the real and imaginary parts of the denominator, respectively.

4. Space required

   Length of subroutine - 41

   Temporary storage in compiled region - 1

   Other routines used - none

   Other temporary storage - none

5. Error codes - Alarm exit is not used.

C. Restrictions and Coding Information:

Data must be in standard USE complex floating point representation. Built-in 1103A floating point arithmetic is used. If the real or imaginary part of the answer is larger than 2^{127}, a division fault or characteristic overflow fault will occur.

D. Timing:

4 milliseconds

E. Mathematical Method:

If $|c| \geq |d|$ then $p+qi = \dfrac{a+b\left(\frac{d}{c}\right)}{c\left(1+\left(\frac{d}{c}\right)^2\right)} + i\,\dfrac{b-a\left(\frac{d}{c}\right)}{c\left(1+\left(\frac{d}{c}\right)^2\right)}$

If $|c| \leq |d|$ then $p+qi = \dfrac{a\left(\frac{c}{d}\right)+b}{d\left(1+\left(\frac{c}{d}\right)^2\right)} + i\,\dfrac{b\left(\frac{c}{d}\right)-a}{d\left(1+\left(\frac{c}{d}\right)^2\right)}$

```
,       ,SUB    ,RWCDV1  ,41      ,         $
,       ,TEMPS  ,1       ,0       ,         $
,       ,INOUT  ,4       ,2       ,         $
,       ,MJ     ,        ,BODY    ,         $
,ALARM  ,ALARM  ,        ,        ,         $
,EXIT   ,MJ     ,        ,FILL    ,         $
,ANS    ,RESERV,2        ,2       ,         $
,ARG    ,RESERV,4        ,4       ,         $
,BODY   ,TM     ,ARG#2   ,A       ,         $
,       ,TM     ,ARG#3   ,TEMP    ,         $
,       ,TJ     ,TEMP    ,CASE2   ,         $
,       ,FD     ,ARG#3   ,ARG#2   ,         $
,       ,TP     ,Q       ,TEMP    ,         $
,       ,FP     ,TEMP    ,K       ,         $
,       ,FM     ,Q       ,ARG#2   ,         $
,NEXT1  ,TP     ,Q       ,ANS#1   ,         $
,       ,TP     ,TEMP    ,Q       ,         $
,       ,FP     ,ARG#1   ,ARG     ,         $
,       ,FD     ,Q       ,ANS#1   ,         $
,       ,TP     ,Q       ,ANS     ,         $
,       ,TN     ,TEMP    ,Q       ,         $
,       ,FP     ,ARG     ,ARG#1   ,         $
,       ,FD     ,Q       ,ANS#1   ,         $
,       ,TP     ,Q       ,ANS#1   ,         $
,       ,MJ     ,        ,EXIT    ,         $
,CASE2  ,FD     ,ARG#2   ,ARG#3   ,         $
,       ,TP     ,Q       ,TEMP    ,         $
,       ,FP     ,TEMP    ,K       ,         $
,       ,FM     ,Q       ,ARG#3   ,         $
,NEXT2  ,TP     ,Q       ,ANS#1   ,         $
,       ,TP     ,ARG     ,Q       ,         $
,       ,FP     ,TEMP    ,ARG#1   ,         $
,       ,FD     ,Q       ,ANS#1   ,         $
,       ,TP     ,Q       ,ANS     ,         $
,       ,TN     ,ARG#1   ,Q       ,         $
,       ,FP     ,TEMP    ,ARG     ,         $
,       ,FD     ,Q       ,ANS#1   ,         $
,       ,TN     ,Q       ,ANS#1   ,         $
,       ,MJ     ,        ,EXIT    ,         $
,K      ,F      ,1.0000  ,000000  ,         $
,       ,ENDSUB,         ,        ,         $
```

RWDET4, Complex, Floating Point Determinant Evaluation for Nearly Triangular Matrices

Programmed by: Werner L. Frank, The Ramo-Wooldridge Corporation

Date: August 1, 1957

A. Purpose:

This subroutine evaluates the determinant of the matrix $A - \lambda I$, where A is nearly triangular, square matrix of order N. By a nearly triangular matrix is meant a matrix for which $a_{ij} = 0$ if $i - j > 1$. For the special case $\lambda = 0$ one obtains the determinant of A. The computation is performed in complex floating point arithmetic.

B. Usage:

1. Specifications. Standard USE subroutine using built-in floating point.

SUB,	RWDET4,	199,
TEMPS,	1 ,	0 ,
INOUT,	4 ,	2 ,

2. Input

First word	-	λ_R
Second word	-	λ_I
Third word	-	00, MATLOC, N
Fourth word	-	00, TEMLOC, 00000

where λ_R = floating point, real component of λ

λ_I = floating point, imaginary component of λ

N = order of the matrix A

MATLOC = address of the first cell in the region where the matrix A is stored, row by row. Zero elements, a_{ij} for $i - j > 1$ are not stored. This region of $N^2 + 3N-2$ cells should be in core storage for efficient operation.

TEMLOC = the address of the first cell in a region of temporary storage of length 2N. This region should be in core storage for efficient operation.

3. Output

First word - floating point, real component of $|A - \lambda I|$

Second word - floating point, imaginary component of $|A - \lambda I|$

4. Space required

Length of subroutine - 199 cells

Temporary storage in compiled region - 1 cell

Other temporary storage - $(N^2 + 5N-2)$ cells

5. Error codes - Alarm exit not used

C. <u>Restrictions and Coding Information:</u>

Data must be in standard USE complex floating point representation. The program is self contained performing its own complex arithmetic. The 1103A built-in floating point package is used. The subroutine does not destroy the input data nor the matrix A. For core storage of 4096 cells, the limitation on the order of the matrix A is $N \leq 60$.

D. Timing:

$$(1.5n^2 + 6n - 4) \text{ milliseconds}$$

E. Mathematical Method:

Elementary row operations are performed on the matrix $A - \lambda I$ reducing it to an upper triangular matrix $\overline{A}$. Before eliminating, the magnitude of leading elements of two rows which are to be linearly combined are compared and the element of largest modulus becomes the pivotal point. The product of the diagonal elements of $\overline{A}$ is the value of $|A - \lambda I|$.

```
,,SUB,RWDET4,198,$
,,TEMPS,1,0,$
      ,       ,INOUT  ,4        ,2        ,                    $
      ,ENT    ,MJ     ,         ,S        ,ENTRANCE            $
      ,ALARM  ,ALARM  ,         ,         ,ALARM EXIT          $
      ,EXIT   ,MJ     ,         ,         ,NORMAL EXIT         $
      ,R1     ,       ,         ,         ,RESULT              $
      ,R2     ,       ,         ,         ,CELLS               $
      ,PAR1   ,       ,         ,         ,LAMBDA              $
      ,PAR2   ,       ,         ,         ,                    $
      ,PAR3   ,       ,         ,         ,MATRIX LOC, N       $
      ,PAR4   ,       ,         ,         ,TEM LOC             $
      ,S      ,TU     ,PAR4     ,P3       ,                    $
      ,       ,TU     ,PAR4     ,P14      ,                    $
      ,       ,TU     ,PAR4     ,P19      ,                    $
      ,       ,TU     ,PAR4     ,P4       ,                    $
      ,       ,RA     ,P4       ,CON7     ,                    $
      ,       ,SP     ,PAR4     ,57       ,                    $
      ,       ,TV     ,A        ,PA       ,                    $
      ,       ,TV     ,A        ,P16      ,                    $
      ,       ,TV     ,A        ,PROD1    ,                    $
      ,       ,TV     ,A        ,PROD11   ,                    $
      ,       ,TV     ,A        ,PROD12   ,                    $
      ,       ,RA     ,PROD1    ,CON5     ,                    $
      ,       ,TV     ,PROD1    ,PROD13   ,                    $
      ,       ,RPV    ,10       ,S1       ,                    $
      ,       ,TP     ,CON3     ,TEM      ,                    $
      ,S1     ,TP     ,CON4     ,R1       ,                    $
      ,       ,TP     ,CON3     ,R2       ,                    $
      ,       ,TP     ,CON6     ,Q        ,                    $
      ,       ,SP     ,PAR3     ,16       ,2N                  $
      ,       ,TU     ,A        ,P23      ,                    $
      ,       ,QS     ,A        ,P        ,                    $
      ,       ,TU     ,PAR3     ,TEM#2    ,                    $
      ,       ,RA     ,P23      ,TEM#2    ,                    $
      ,       ,TU     ,PAR3     ,PA       ,                    $
      ,       ,TV     ,PAR3     ,TEM      ,                    $
      ,       ,RS     ,TEM      ,CON2     ,N-2                 $
,P,RP3,,PA#1,$
      ,PA     ,TP     ,FILL     ,FILL#T   ,                    $
      ,       ,TU     ,PAR4     ,C1R      ,                    $
      ,       ,TV     ,PA       ,C2R      ,                    $
      ,       ,TU     ,PAR4     ,C3R      ,                    $
      ,       ,TV     ,PA       ,C4R      ,                    $
      ,C1R    ,FS     ,FILL     ,PAR1     ,                    $
      ,C2R    ,TP     ,Q        ,FILL     ,                    $
      ,       ,RA     ,C3R      ,CON7     ,                    $
      ,       ,RA     ,C4R      ,CON5     ,                    $
      ,C3R    ,FS     ,FILL     ,PAR2     ,                    $
      ,C4R    ,TP     ,Q        ,FILL     ,                    $
      ,P1     ,TP     ,TEM      ,TEM#9    ,                    $
      ,       ,TV     ,CON8     ,P23      ,                    $
      ,       ,TV     ,CON8     ,P25      ,                    $
      ,       ,TV     ,PROD11   ,P27      ,                    $
```

```
        ,       ,TV     ,PROD1     ,P29      ,                                   $
        ,       ,TU     ,P23       ,P5       ,                                   $
        ,       ,TU     ,P23       ,P5A      ,                                   $
        ,       ,RA     ,P5A       ,CON7     ,                                   $
        ,       ,TU     ,P23       ,P16      ,                                   $
        ,       ,TU     ,P23       ,P15      ,                                   $
        ,       ,TU     ,P23       ,NOINT    ,                                   $
        ,       ,TU     ,P3        ,P25      ,                                   $
        ,P3     ,TM     ,FILL#T    ,TEM#2    ,DETERMINE                          $
        ,P4     ,TM     ,FILL#T#1  ,A        ,PIVOT                              $
        ,       ,TJ     ,TEM#2     ,P5       ,                                   $
        ,       ,TP     ,A         ,TEM#2    ,                                   $
        ,P5     ,TM     ,FILL#L#2N,TEM#3     ,                                   $
        ,P5A    ,TM     ,FILL#L#2N#1,A       ,                                   $
        ,       ,TJ     ,TEM#3     ,P8       ,                                   $
        ,       ,ZJ     ,P7        ,P6       ,                                   $
        ,P6     ,EJ     ,TEM#2     ,OUT      ,                                   $
        ,P7     ,TM     ,A         ,TEM#3    ,                                   $
        ,P8     ,RS     ,TEM#2     ,TEM#3    ,                                   $
,,SJ,INT,NOINT-1,$
        ,INT    ,TN     ,R1        ,R1       ,CHANGE SIGN OF                     $
        ,       ,TN     ,R2        ,R2       ,DET                                $
        ,       ,RP3    ,2         ,P14A     ,                                   $
        ,P14    ,TP     ,FILL#T    ,ARG      ,                                   $
        ,P14A   ,RP3    ,2         ,P15A     ,                                   $
        ,P15    ,TP     ,FILL#L#2N,ARG#2     ,                                   $
        ,P15A   ,RJ     ,NOBODY    ,BODY     ,DIVIDE                             $
        ,       ,RP3    ,2         ,P15B     ,                                   $
        ,       ,TP     ,ANS       ,TEM#2    ,FORM K                             $
        ,P15B   ,RA     ,P23       ,CON2     ,                                   $
        ,       ,RP3    ,2         ,P20      ,                                   $
        ,P16    ,TP     ,FILL#L#2N,FILL#T    ,                                   $
        ,       ,RP3    ,2         ,P18      ,                                   $
        ,NOINT  ,TP     ,FILL#L#2N,ARG       ,                                   $
        ,P18    ,RP3    ,2         ,P19A     ,                                   $
        ,P19    ,TP     ,FILL#T    ,ARG#2    ,                                   $
        ,P19A   ,RJ     ,NOBODY    ,BODY     ,DIVIDE                             $
        ,       ,RP3    ,2         ,P19B     ,                                   $
        ,       ,TP     ,ANS       ,TEM#2    ,FORM K                             $
        ,P19B   ,RA     ,P25       ,CON2     ,                                   $
        ,P20    ,RJ     ,PROD2     ,PROD1    ,                                   $
        ,CORR   ,RA     ,P23       ,CON1     ,                                   $
        ,       ,RA     ,P25       ,CON1     ,                                   $
        ,       ,TU     ,P23       ,CORR5    ,                                   $
        ,       ,TV     ,P23       ,CORR6    ,                                   $
        ,       ,TU     ,P23       ,CORR7    ,                                   $
        ,       ,TV     ,P23       ,CORR8    ,                                   $
        ,CORR5  ,FS     ,FILL      ,PAR1     ,                                   $
        ,CORR6  ,TP     ,Q         ,FILL     ,                                   $
        ,       ,RA     ,CORR7     ,CON7     ,                                   $
        ,       ,RA     ,CORR8     ,CON5     ,                                   $
        ,CORR7  ,FS     ,FILL      ,PAR2     ,                                   $
        ,CORR8  ,TP     ,Q         ,FILL     ,                                   $
,,MJ,,P24,$
```

```
    ,P22     ,RP3    ,2         ,P24       ,                 $
    ,P23     ,TP     ,FILL#L#2N,FILL      ,TEM#4            $
    ,P24     ,RP3    ,2         ,P26       ,                 $
    ,P25     ,TP     ,FILL#T    ,FILL      ,TEM#6            $
    ,P26     ,TN     ,TEM#6     ,Q         ,                 $
    ,        ,FP     ,TEM#2     ,TEM#4     ,                 $
    ,        ,FI     ,TEM#7     ,TEM#3     ,                 $
    ,P27     ,TP     ,Q         ,FILL#T    ,                 $
    ,        ,TN     ,TEM#5     ,Q         ,                 $
    ,        ,FI     ,TEM#2     ,TEM#7     ,                 $
    ,        ,FI     ,TEM#3     ,TEM#6     ,                 $
    ,P29     ,TN     ,Q         ,FILL#T#1  ,                 $
    ,        ,RA     ,P27       ,CON2      ,                 $
    ,        ,RA     ,P29       ,CON2      ,                 $
    ,P30     ,RA     ,P23       ,CON1      ,                 $
    ,        ,RA     ,P25       ,CON1      ,                 $
    ,        ,IJ     ,TEM#9     ,P22       ,                 $
    ,        ,IJ     ,TEM       ,P1        ,                 $
    ,        ,RJ     ,PROD2     ,PROD1     ,                 $
    ,        ,MJ     ,          ,EXIT      ,                 $
    ,OUT     ,TP     ,CON3      ,R1        ,                 $
    ,        ,TP     ,CON3      ,R2        ,                 $
    ,        ,MJ     ,          ,EXIT      ,                 $
    ,PROD1   ,FM     ,R2        ,FILL#T#1  ,                 $
    ,        ,TN     ,Q         ,Q         ,                 $
    ,PROD11,FI       ,R1        ,FILL#T    ,                 $
    ,        ,TP     ,Q         ,TEM#8     ,                 $
    ,PROD12,FM       ,R2        ,FILL#T    ,                 $
    ,PROD13,FI       ,R1        ,FILL#T#1  ,                 $
    ,        ,TP     ,Q         ,R2        ,                 $
    ,        ,TP     ,TEM#8     ,R1        ,                 $
    ,PROD2   ,MJ     ,          ,FILL      ,                 $
    ,CON1    ,       ,2         ,          ,                 $
    ,CON2    ,       ,          ,2         ,                 $
    ,CON3    ,00     ,          ,          ,                 $
    ,CON4    ,F      ,1         ,          ,                 $
    ,CON5    ,       ,          ,1         ,                 $
,CON6,,7777↑B,,$
    ,CON7    ,       ,1         ,          ,                 $
    ,CON8    ,       ,          ,TEM#4     ,                 $
    ,TEM     ,RESERV,10         ,10        ,                 $
    ,L       ,EQUALS,0          ,          ,                 $
    ,T       ,EQUALS,0          ,          ,                 $
    ,2N      ,EQUALS,0          ,          ,                 $
    ,ANS     ,RESERV,2          ,2         ,                 $
    ,ARG     ,RESERV,4          ,4         ,                 $
    ,BODY    ,TM     ,ARG#2     ,A         ,                 $
    ,        ,TM     ,ARG#3     ,TEMP      ,                 $
    ,        ,TJ     ,TEMP      ,CASE2     ,                 $
    ,        ,FD     ,ARG#3     ,ARG#2     ,                 $
    ,        ,TP     ,Q         ,TEMP      ,                 $
    ,        ,FP     ,TEMP      ,K         ,                 $
    ,        ,FM     ,Q         ,ARG#2     ,                 $
    ,        ,TP     ,Q         ,A         ,                 $
```

```
       ,        ,ZJ     ,NEXT1     ,ALARM     ,                    $
       ,NEXT1   ,TP     ,Q         ,ANS#1     ,                    $
       ,        ,TP     ,TEMP      ,Q         ,                    $
       ,        ,FP     ,ARG#1     ,ARG       ,                    $
       ,        ,FD     ,Q         ,ANS#1     ,                    $
       ,        ,TP     ,Q         ,ANS       ,                    $
       ,        ,TN     ,TEMP      ,Q         ,                    $
       ,        ,FP     ,ARG       ,ARG#1     ,                    $
       ,        ,FD     ,Q         ,ANS#1     ,                    $
       ,        ,TP     ,Q         ,ANS#1     ,                    $
,,MJ,,NOBODY,$
       ,CASE2   ,FD     ,ARG#2     ,ARG#3     ,                    $
       ,        ,TP     ,Q         ,TEMP      ,                    $
       ,        ,FP     ,TEMP      ,K         ,                    $
       ,        ,FM     ,Q         ,ARG#3     ,                    $
       ,        ,TP     ,Q         ,A         ,                    $
       ,        ,ZJ     ,NEXT2     ,ALARM     ,                    $
       ,NEXT2   ,TP     ,Q         ,ANS#1     ,                    $
       ,        ,TP     ,ARG       ,Q         ,                    $
       ,        ,FP     ,TEMP      ,ARG#1     ,                    $
       ,        ,FD     ,Q         ,ANS#1     ,                    $
       ,        ,TP     ,Q         ,ANS       ,                    $
       ,        ,TN     ,ARG#1     ,Q         ,                    $
       ,        ,FP     ,TEMP      ,ARG       ,                    $
       ,        ,FD     ,Q         ,ANS#1     ,                    $
       ,        ,TN     ,Q         ,ANS#1     ,                    $
       ,NOBODY  ,MJ     ,          ,FILL      ,                    $
       ,K       ,F      ,1.0000    ,000000    ,                    $
,,ENDSUB,,,$
```

A. Identification

1. Title: High Speed Printer Edit Routine, WF0001

2. Authors: R. Graham, W. Bauer
   Date: 1 July 1957

3. Installation: Wright Field

B. Purpose

This routine edits alphanumeric information from core or drum memory and prepares a magnetic tape on any designated servo suitable for listing on the off-line high speed printer.

C. Method

Any of a number of conversion or translation routines may be used to produce the alphanumeric characters for the Edit Routine from binary data.

This routine requires that one argument word be transferred into it by the calling sequence and that a parameter list be available. One entry into the Edit Routine is sufficient to produce any number of lines (blockettes) N, less than 1000, of identical format. Each column in the group of N lines requires two descriptive words in the parameter list. The edited information is recorded on tape in blocks of 720 hexabit characters, each block thus containing six blockettes of 120 characters.

D. Usage

1. Calling Sequence

Loc	Op	U-addr	V-addr
r-1	TP	Arg	Edit + 3
r	RJ	Edit + 2	Edit
r+1	Normal Return		

2. Control Data

a. Argument

The argument word which is transferred into the Edit Routine by the calling sequence is made up as follows:

Loc	Op	U-addr	V-addr
Arg	X X F E	XX XXX T N	XXXXX P

P is the address of the first word of the parameter list.

N (decimal) is the number of lines of identical format to be produced. ($1 \leq N \leq 999$)

T (decimal) designates the servo unit on which output is to be recorded. $(1 \leq T \leq 10)$

F and E (octal) are printer control digits.

If F = 1,2,3, or 4, the corresponding Fast Feed I, II, III, or IV is placed in the first of the N blockettes.

If $E_{32} = 1$, a multiline symbol is placed in each of the N blockettes.

If $E_{31} = 1$, a printer breakpoint symbol is placed in the first of the N blockettes.

If $E_{30} = 1$, a printer stop symbol is placed in the last of the N blockettes.

When the argument specifies printer control symbols, the total number of characters available for blockette construction is 120 minus the number of symbols used.

b. Parameters

The parameter list, which is stored beginning at address P, contains a pair of parameter words for each column and is followed by a pair of control words P_1 and P_2 to signal the end of the list. P_1 is <u>always</u> zero. If P_2 is zero, a partially filled block is not written at the end of N blockettes which permits an accumulation of blockettes; however, a full block (six blockettes) is <u>always</u> written on magnetic tape. If P_2 is not zero, a partially filled block is written at the end of N blockettes with each unused blockette translated as a blank line of output.

Each pair of words in the parameter list is made up as follows:

Loc	Op	U-addr	V-addr
Param	XX D	XXXXX M	XXX XX W S
Param + 1	--	-----	XXXXX C

W (decimal) is the number of characters allotted to the column. $(1 \leq W \leq 120)$

S (decimal) is the number of spaces to precede the column. $(0 \leq S \leq 99)$

M is the address of the first word of data to appear in this column.

D (octal) is the increment to be added to M to obtain the addresses of data for succeeding lines of this column. The original parameter list is never altered.

C is the first address occupied by the conversion routine which is required for data in this column.

The remainder of the second word of the pair may contain additional information required by the particular conversion routine, such as scaling, or the first address of additional information required. The precise form of this word will be specified by the particular conversion routine.

The Edit Routine generates a calling sequence which transfers this pair of parameter words into the particular conversion routine with the address C replaced by the address (K) at which the conversion routine is to store W results.

The results from the conversion routine will be a series of six-bit excess three characters. On exit from the conversion routine Q_{35} will be zero if these characters are packed six to a word. Q_{35} will be one if these characters are stored one to a word in the rightmost six bits, with the leftmost thirty bits being zero.

3. Space Required

327 memory cells

4. Error Codes

(A_L) = Code

If code = 1, Servo is greater than 10, or equal to zero.

= 2, N is zero.

= 3, P is zero.

= 4, C is zero.

= 5, the Blockette is to be greater than 120 characters.

(A_R) = Argument

(Q) = WF0001 in Flex Code.

5. Format Generated

The format is controlled entirely by the argument and parameter list.

E. Restrictions

1. A change in servo designation causes a partially filled block (if it exists) to be recorded on the old servo.

2. If P_2 is zero and N is not an even multiple of 6, a "clean-up" pass must be executed to write the partially filled block.

3. Other Program Required

One or more subsidiary routines are necessary to process data for editing.

F. Coding Information

1. Constants and their locations

   a. Alphanumeric

      1K through 1K6

   b. Numeric

      2K through 2K27

   c. Internal temporary storage

      3K through 3K4

2. Erasable output storage

   P through P126

° MEANS $
MEANS +
/ MEANS -

```
,     ,SUB    ,WF0001,327     ,°
,     ,TEMPS  ,127   ,0       ,
,     ,INOUT  ,1     ,0       ,
,B    ,MJ0    ,0     ,C       ,ENTRY°
,P1   ,ALARM  ,      ,        ,EXIT ALARM°
,B2   ,MJ0    ,0     ,FILL    ,EXIT NORMAL°
,PARAM,       ,      ,        ,
,B3   ,RESERV,120    ,120     ,BIN
,C    ,TP     ,1K    ,F       ,PRESETS/SWITCH 1°
,     ,TP     ,1K1   ,F3      ,SWITCH 2°
,     ,TP     ,1K2   ,F14     ,SWITCH 3°
,     ,TP     ,2K    ,P#126   ,SET LOOP COUNT TO ZERO
,     ,TV     ,C3    ,3K      ,SET STORE DATA LOCATION °
,     ,RP1    ,120   ,C1      ,°
,     ,TP     ,2K1   ,P       ,STORE SPACES IN BLOCKETTE°
,C1   ,55     ,PARAM,Q#3      ,SHIFT ARGUMENT
,     ,QT     ,2K7   ,P#120   ,
,     ,ZJ     ,C2    ,C4      ,TEST FOR FAST FEED SYMBOL°
,C2   ,RA     ,P#120,1K6      ,YES
,     ,LA     ,A     ,15      ,°
,     ,TU     ,A     ,C3      ,°
,C3   ,TP     ,FILL  ,P       ,SET FF SYMBOL°
,     ,RA     ,3K    ,2K1     ,°
,C4   ,QJ     ,C5    ,C6      ,TEST FOR MULTILINE SYMBOL°
,C5   ,RS     ,F     ,2K1     ,YES SET SWITCH1°
,     ,TV     ,3K    ,F1      ,SET ML LOCATION°
,     ,RA     ,3K    ,2K1     ,°
,C6   ,QJ     ,C7    ,C9      ,TEST FOR BREAKPOINT SYMBOL°
,C7   ,TV     ,3K    ,C8      ,YES°
,C8   ,TP     ,2K14  ,FILL    ,SET BP SYMBOL°
,     ,RA     ,3K    ,2K1     ,°
,C9   ,QJ     ,C10   ,C11     ,TEST FOR PRINTER STOP SYMBOL°
,C10  ,RS     ,F3    ,2K1     ,YES SET SWITCH 2°
,     ,TV     ,3K    ,F4      ,SET PS LOCATION°
,     ,RA     ,3K    ,2K1     ,°
,C11  ,LQ     ,Q     ,9       ,SHIFT ARGUMENT°
,     ,QT     ,2K16  ,P#120   ,EXTRACT P
,     ,ZJ     ,C14   ,C12     ,DOES P EQUAL ZERO°
,C12  ,SP     ,2K3   ,36      ,YES LOAD 3 CODE AND°
,C13  ,SA     ,PARAM,0        ,LOAD ARGUMENT INTO A
,     ,TP     ,2K27  ,Q       ,°
,     ,MJ0    ,0     ,B1      ,GO TO ALARM EXIT TO PRINT A°
,C14  ,LQ     ,Q     ,6       ,SHIFT ARGUMENT°
,     ,QT     ,2K17  ,A       ,°
,     ,DV     ,2K18  ,P#121   ,STORE SERVO NUMBER
,     ,ZJ     ,C16   ,C15     ,DOES NUMBER OF LINES EQUAL ZERO°
,C15  ,SP     ,2K2   ,36      ,YES LOAD 2 CODE°
,     ,MJ0    ,0     ,C13     ,°
,C16  ,TP     ,A     ,P#122   ,STORE N
```

```
,     ,TP   ,2K8  ,A      ,°
,     ,TJ   ,P#121,C18    ,IS SERVO GREATER THAN 10
,C17  ,TP   ,P#121,A      ,NO
,     ,ZJ   ,C19  ,C18    ,DOES SERVO EQUAL ZERO°
,C18  ,SP   ,2K1  ,36     ,SERVO IN ERROR/LOAD 1 CODE°
,     ,MJ0  ,0    ,C13    ,°
,C19  ,EJ   ,3K1  ,E      ,DOES NEW SERVO EQUAL OLD°
,C20  ,TP   ,3K1  ,A      ,NO°
,     ,ZJ   ,D0   ,D2     ,TEST FOR INITIAL SETUP
,D0   ,TP   ,3K2  ,A      ,NO
,     ,ZJ   ,D1   ,D3     ,IS THERE ANY DATA IN BLOCK°
,D1   ,EF   ,0    ,3K8    ,YES°
,     ,RP1  ,120  ,D2     ,WRITE°
,     ,EW1  ,0    ,B3     ,BLOCK AND°
,D2   ,RP1  ,120  ,D3     ,FILL BLOCK°
,     ,TP   ,2K20 ,B3     ,WITH CONDENSED SPACES°
,D3   ,TU   ,1K5  ,F9     ,°
,     ,TV   ,1K5  ,F12    ,°
,     ,TP   ,2K   ,3K2    ,SET BLOCKETTE COUNT TO ZERO°
,D4   ,RJ   ,D4   ,D5     ,°
,D5   ,TP   ,P#121,3K1    ,STORE NEW SERVO NUMBER
,     ,EA   ,P#121,A#12   ,
,     ,AT   ,2K19 ,3K3    ,PREPARE NEW MT COMMAND°
,E    ,TP   ,3K   ,P#121  ,SET STORE DATA LOCATION
,     ,TJ   ,P#122,E1     ,DOES N EQUAL ZERO
,     ,MJ0  ,0    ,B2     ,YES GO TO NORMAL EXIT°
,E1   ,TU   ,P#120,E3     ,SET E3
,E2   ,RP3  ,2    ,E4     ,°
,E3   ,TP   ,FILL ,P#123  ,OBTAIN PAIR OF PARAMETER WORDS
,E4   ,RA   ,E3   ,2K23   ,°
,     ,TP   ,P#123,A      ,
,     ,ZJ   ,E5   ,F      ,DOES PARAMETER EQUAL ZERO°
,E5   ,TP   ,2K17 ,Q      ,NO°
,     ,QT   ,P#124,A      ,EXTRACT C
,     ,ZJ   ,E7   ,E6     ,DOES C EQUAL ZERO°
,E6   ,SP   ,2K4  ,36     ,YES LOAD 4 CODE°
,     ,MJ0  ,0    ,C13    ,°
,E7   ,TV   ,A    ,E12    ,SET C IN RJ°
,     ,LA   ,A    ,15     ,°
,     ,TU   ,A    ,E12    ,MODIFY RJ ORDER°
,     ,RA   ,E12  ,2K23   ,FOR SUBRTN ENTRY°
,     ,TV   ,E12  ,E11    ,SET UP RP TO°
,     ,RA   ,E11  ,2K3    ,STORE PARAMETERS°
,     ,QT   ,P#123,A      ,EXTRACT WS
,     ,DV   ,2K26 ,P#125  ,STORE W
,     ,RA   ,A    ,P#121  ,MODIFY DATA
,     ,TV   ,A    ,P#124  ,STORAGE LOCATION BY S
,     ,RA   ,A    ,P#125  ,ADD W
,     ,TJ   ,1K4  ,F9     ,IS BLOCKETTE TOO LARGE°
```

```
,E8  ,SP   ,2K5  ,36      ,YES LOAD 5 CODE°
,    ,MJ0  ,0    ,C13     ,°
,E9  ,TV   ,A    ,P#121   ,STORE NEXT DATA
,    ,IJ   ,A    ,E10     ,STORAGE LOCATION°
,E1  ,TV   ,A    ,E22     ,SETUP ORDER TO UNPACK°
,    ,55   ,P#123,0#6     ,
,    ,QT   ,2K21 ,A       ,EXTRACT D°
,    ,MP   ,A    ,P#126   ,LOOP COUNT TIMES D
,    ,LA   ,A    ,15      ,°
,    ,AT   ,P#123,P#123   ,MODIFY M
,    ,RP3  ,2    ,E12     ,°
,E11 ,TP   ,P#123,FILL    ,STORE PARAMETER PAIR
,E12 ,RJ   ,FILL ,FILL    ,GO TO SUBROUTINE°
,    ,QJ   ,F2   ,E13     ,IS DATA PACKED°
,E13 ,TV   ,2K   ,E16     ,YES°
,    ,TP   ,2K   ,3K4     ,INDEX TO ZERO°
,    ,RS   ,P#124,2K1     ,
,    ,TP   ,P#125,A       ,W TO A
,    ,DV   ,2K6  ,P#123   ,NUMBER OF LOCATIONS
,    ,ZJ   ,E14  ,E15     ,IS THERE A REMAINDER°
,E14 ,TP   ,A    ,3K4     ,YES SET INDEX°
,    ,MP   ,A    ,2K6     ,°
,    ,TV   ,A    ,E16     ,SET LQ FOR SPECIAL SHIFT°
,    ,RA   ,P#123,2K1     ,INCREASE LOCATION NO
,E15 ,RA   ,P#124,P#123   ,DETERMINE LAST LOCATION
,    ,LA   ,A    ,15      ,°
,    ,TU   ,A    ,E16     ,SET LQ°
,E16 ,LQ   ,FILL ,FILL    ,SHIFT°
,    ,TU   ,E16  ,E18     ,°
,    ,TP   ,3K4  ,A       ,°
,    ,ZJ   ,E19  ,E17     ,TEST FOR REMAINDER°
,E17 ,TP   ,2K6  ,3K4     ,NO SET 6 INDEX°
,E18 ,SP   ,FILL ,0       ,°
,    ,TP   ,A    ,Q       ,WORD TO Q°
,E19 ,RS   ,E18  ,2K24    ,°
,E20 ,IJ   ,3K4  ,E21     ,°
,    ,MJ0  ,0    ,E17     ,°
,E21 ,IJ   ,P#125,E22     ,INDEX ON W
,    ,MJ0  ,0    ,F2      ,°
,E22 ,QT   ,2K21 ,FILL    ,EXTRACT TWO CHARACTERS°
,    ,RS   ,E22  ,2K1     ,°
,    ,LQ   ,Q    ,30      ,°
,    ,MJ0  ,0    ,E20     ,°
,F   ,MJ0  ,0    ,F2      ,SWITCH 1°
,F1  ,TP   ,2K13 ,FILL    ,SET ML SYMBOL°
,F2  ,RA   ,3K2  ,2K1     ,INCREASE BLOCKETTE COUNT°
,    ,RA   ,P#126,2K1     ,INCREASE LOOP COUNT
,    ,TP   ,P#122,A       ,
,    ,ZJ   ,F7   ,F3      ,DOES N EQUAL ZERO°
```

```
,F3  ,MJ0  ,0     ,F5     ,YES SWITCH 2°
,F4  ,TP   ,2K15  ,FILL   ,SET PS SYMBOL°
,F5  ,TP   ,P#124 ,A      ,
,    ,ZJ   ,F6    ,F7     ,DOES FINAL PARAMETER EQUAL ZERO°
,F6  ,RS   ,F14   ,2K1    ,NO SET SWITCH 3°
,F7  ,TU   ,1K3   ,F10    ,CONDENSE BLOCKETTE°
,    ,TP   ,2K21  ,0      ,AND STORE IN BLOCK°
,    ,TP   ,2K25  ,P#123  ,
,F8  ,TP   ,2K5   ,P#124  ,
,F9  ,SP   ,FILL  ,6      ,°
,F10 ,QA   ,FILL  ,FILL   ,°
,    ,RA   ,F10   ,2K24   ,°
,    ,IJ   ,P#124 ,F9     ,
,    ,RA   ,F9    ,2K24   ,°
,    ,RA   ,F10   ,2K1    ,°
,F11 ,IJ   ,P#123 ,F8     ,
,F12 ,TP   ,3K2   ,A      ,°
,    ,EJ   ,2K6   ,F13    ,IS BLOCK FULL°
,    ,MJ0  ,0     ,F14    ,NO°
,F13 ,RJ   ,D4    ,D1     ,YES.WRITE BLOCK°
,F14 ,MJ0  ,0     ,F16    ,SWITCH 3°
,F15 ,RJ   ,D4    ,D0     ,WRITE BLOCK
,F16 ,RP1  ,120   ,E      ,FILL BLOCKETTE°
,    ,TP   ,2K1   ,P      ,WITH SPACES°
,1K  ,MJ0  ,0     ,F2     ,°
,1K1 ,MJ0  ,0     ,F5     ,°
,1K2 ,MJ0  ,0     ,F16    ,°
,1K3 ,     ,P     ,       ,°
,1K4 ,     ,      ,P#121  ,
,1K5 ,     ,B3    ,B3     ,°
,1K6 ,     ,      ,2K8    ,°
,2K  ,B    ,      ,0      ,ZERO°
,2K1 ,B    ,      ,1      ,ONE°
,2K2 ,B    ,      ,2      ,TWO°
,2K3 ,B    ,      ,3      ,THREE°
,2K4 ,B    ,      ,4      ,FOUR°
,2K5 ,B    ,      ,5      ,FIVE°
,2K6 ,B    ,      ,6      ,SIX°
,2K7 ,B    ,      ,7      ,SEVEN°
,2K8 ,B    ,      ,12     ,INDEX°
,2K9 ,B    ,      ,37     ,FAST FEED/FF 1°
,2K10,B    ,      ,42     ,FF 2°
,2K11,B    ,      ,57     ,FF 3°
,2K12,B    ,      ,76     ,FF 4°
,2K13,B    ,      ,20     ,MULTILINE/ML°
,2K14,B    ,      ,61     ,BREAKPOINT BP°
,2K15,B    ,      ,60     ,PRINTER STOP°
,2K16,B    ,77777,00000   ,MASK U
,2K17,B    ,      ,77777  ,MASK V°
```

```
,2K18,B       ,     ,01750  ,DIVISOR°
,2K19,B02     ,00746,00000  ,OFF LINE WRITE BLOCK°
,2K20,B01     ,01010,10101  ,CONDENSED SPACES°
,2K21,B       ,     ,77     ,6 BIT MASK°
,2K22,B       ,     ,777    ,9 BIT MASK°
,2K23,B       ,    2,00000  ,ADVANCE U°
,2K24,B       ,    1,00000  ,ADVANCE U°
,2K25,B       ,     ,23     ,INDEX°
,2K26,B       ,     ,00144  ,DIVISOR°
,2K27,B31     ,26373,73752  ,°
,3K  ,B       ,     ,0      ,STORE DATA LOCATION°
,3K1 ,B       ,     ,0      ,SERVO NUMBER°
,3K2 ,B       ,     ,0      ,BLOCKETTE COUNT°
,3K3 ,B       ,     ,0      ,TAPE COMMAND°
,3K4 ,        ,     ,0      ,INDEX°
,     ,ENDSUB,      ,       ,°
```

USE Letter No: PC-4 December 11, 1957

To: USE Policy Committee, Publications Committee, Installation Heads.

Enclosed herewith is the coding only for USE routines WF 0001, WF 0002, WF0003, WF0004 and WF0005. The original routines, previously distributed, were somewhat illegible. WF was kind enough to provide us with another master.

Please note that line C18(+2) in WF 0004 was incomplete in the original copy which was distributed.

Dirk de Vries
Executive Secretary, USE
Remington Rand Univac,
Univac Park,
St. Paul 16,
MINNESOTA.

DdV:diw

```
,         ,SUB      ,WF0001   ,327      ,                                    $
,         ,TEMPS    ,127      ,0        ,                                    $
,         ,IN OUT   ,1        ,0        ,                                    $
,B        ,MJO      ,0        ,C        ,ENTRY                               $
,B1       ,ALARM    ,         ,       O ,EXIT ALARM                          $
,B2       ,MJO      ,0        ,FILL     ,EXIT NORMAL                         $
,PARAM    ,         ,         ,         ,                                    $
,B3       ,RESERV   ,120      ,120      ,BIN                                 $
,C        ,TP       ,1K       ,F        ,PRESETS/SWITCH 1                    $
,         ,TP       ,1K1      ,F3       ,SWITCH 2                            $
,         ,TP       ,1K2      ,F14      ,SWITCH 3                            $
,         ,TP       ,2K       ,P+126    ,SET LOOP COUNT TO ZERO              $
,         ,TV       ,C3       ,3K       ,SET STORE DATA LOCATION             $
,         ,RP1      ,120      ,C1       ,                                    $
,         ,TP       ,2K1      ,P        ,STORE SPACES IN BLOCKETTE           $
,C1       ,55       ,PARAM    ,Q+3      ,SHIFT ARGUMENT                      $
,         ,QT       ,2K7      ,P+120    ,                                    $
,         ,ZJ       ,C2       ,C4       ,TEST FOR FAST FEED SYMBOL           $
,C2       ,RA       ,P+120    ,1K6      ,YES                                 $
,         ,LA       ,A        ,15       ,                                    $
,         ,TU       ,A        ,C3       ,                                    $
,C3       ,TP       ,FILL     ,P        ,SET FF SYMBOL                       $
,         ,RA       ,3K       ,2K1      ,                                    $
,C4       ,QJ       ,C5       ,C6       ,TEST FOR MULTILINE SYMBOL           $
,C5       ,RS       ,F        ,2K1      ,YES SET SWITCH 1                    $
,         ,TV       ,3K       ,F1       ,SET ML LOCATION                     $
,         ,RA       ,3K       ,2K1      ,                                    $
,C6       ,QJ       ,C7       ,C9       ,TEST FOR BREAKPOINT SYMBOL          $
,C7       ,TV       ,3K       ,C8       ,YES                                 $
,C8       ,TP       ,2K14     ,FILL     ,SET BP SYMBOL                       $
,         ,RA       ,3K       ,2K1      ,                                    $
,C9       ,QJ       ,C10      ,C11      ,TEST FOR PRINTER STOP SYMBOL        $
,C10      ,RS       ,F3       ,2K1      ,YES SET SWITCH 2                    $
,         ,TV       ,3K       ,F4       ,SET PS LOCATION                     $
,         ,RA       ,3K       ,2K1      ,                                    $
,C11      ,LQ       ,Q        ,9        ,SHIFT ARGUMENT                      $
,         ,QT       ,2K16     ,P+120    ,EXTRACT P                           $
,         ,ZJ       ,C14      ,C12      ,DOES P EQUAL ZERO                   $
,C12      ,SP       ,2K3      ,36       ,YES LOAD 3 CODE AND                 $
,C13      ,SA       ,PARAM    ,0        ,LOAD ARGUMENT INTO A                $
,         ,TP       ,2K27     ,Q        ,                                    $
,         ,MJO      ,0        ,B1       ,GO TO ALARM EXIT TO PRINT A         $
,C14      ,LQ       ,Q        ,6        ,SHIFT ARGUMENT                      $
,         ,QT       ,2K17     ,A        ,                                    $
,         ,DV       ,2K18     ,P+121    ,STORE SERVO NUMBER                  $
,         ,ZJ       ,C16      ,C15      ,DOES NUMBER OF LINES EQUAL ZERO     $
,C15      ,SP       ,2K2      ,36       ,YES LOAD 2 CODE                     $
,         ,MJO      ,0        ,C13      ,                                    $
,C16      ,TP       ,A        ,P+122    ,STORE N                             $
,         ,TP       ,2K8      ,A        ,                                    $
,         ,TJ       ,P+121    ,C18      ,IS SERVO GREATER THAN 10            $
,C17      ,TP       ,P+121    ,A        ,NO                                  $
```

```
,     ,ZJ    ,C19     ,C18     ,DOES SERVO EQUAL ZERO              $
,C18  ,SP    ,2K1     ,36      ,SERVO IN ERROR/LOAD 1 CODE         $
,     ,MJO   ,0       ,C13     ,                                   $
,C19  ,EJ    ,3K1     ,E       ,DOES NEW SERVO EQUAL OLD           $
,C20  ,TP    ,3K1     ,A       ,NO                                 $
,     ,ZJ    ,DO      ,D2      ,TEST FOR INITIAL SETUP             $
,DO   ,TP    ,3K2     ,A       ,NO                                 $
,     ,ZJ    ,D1      ,D3      ,IS THERE ANY DATA IN BLOCK         $
,D1   ,EF    ,0       ,3K3     ,YES                                $
,     ,RP1   ,120     ,D2      ,WRITE                              $
,     ,EW1   ,0       ,B3      ,BLOCK AND                          $
,D2   ,RPI   ,120     ,D3      ,FILL BLOCK                         $
,     ,TP    ,2K20    ,B3      ,WITH CONDENSED SPACES              $
,D3   ,TU    ,1K5     ,F9      ,                                   $
,     ,TV    ,1K5     ,F10     ,                                   $
,     ,TP    ,2K      ,3K2     ,SET BLOCKETTE COUNT TO ZERO        $
,D4   ,RJ    ,D4      ,D5      ,                                   $
,D5   ,TP    ,P+121   ,3K1     ,STORE NEW SERVO NUMBER             $
,     ,54    ,P+121   ,A+12    ,                                   $
,     ,AT    ,2K19,   ,3K3     ,PREPARE NEW MT COMMAND             $
,E    ,TP    ,3K      ,P+121   ,SET STORE DATA LOCATION            $
,     ,IJ    ,P+122   ,E1      ,DOES N EQUAL ZERO                  $
,     ,MJO   ,0       ,B2      ,YES GO TO NORMAL EXIT              $
,E1   ,TU    ,P+120   ,E3      ,SET E3                             $
,E2   ,RP3   ,2       ,E4      ,                                   $
,E3   ,TP    ,FILL    ,P+123   ,OBTAIN PAIR OF PARAMETER WORDS     $
,E4   ,RA    ,E3      ,2K23    ,                                   $
,     ,TP    ,P+123   ,A       ,                                   $
,     ,ZJ    ,E5      ,F       ,DOES PARAMETER EQUAL ZERO          $
,E5   ,TP    ,2K17    ,Q       ,NO                                 $
,     ,QT    ,P+124   ,A       ,EXTRACT C                          $
,     ,ZJ    ,E7      ,E6      ,DOES C EQUAL ZERO                  $
,E6   ,SP    ,2K4     ,36      ,YES LOAD 4 CODE                    $
,     ,MJO   ,0       ,C13     ,                                   $
,E7   ,TV    ,A       ,E12     ,SET C IN RJ                        $
,     ,LA    ,A       ,15      ,                                   $
,     ,TU    ,A       ,E12     ,MODIFY RJ ORDER                    $
,     ,RA    ,E12     ,2K23    ,FOR SUBRTN ENTRY                   $
,     ,TV    ,E12     ,E11     ,SET UP RP TO                       $
,     ,RA    ,E11     ,2K3     ,STORE PARAMETERS                   $
,     ,QT    ,P+123   ,A       ,EXTRACT WS                         $
,     ,DV    ,2K26    ,P+125   ,STORE W                            $
,     ,RA    ,A       ,P+121   ,MODIFY DATA                        $
,     ,TV    ,A       ,P+124   ,STORAGE LOCATION BY S              $
,     ,RA    ,A       ,P+125   ,ADD W                              $
,     ,TJ    ,1K4     ,E9      ,IS BLOCKETTE TOO LARGE             $
,E8   ,SP    ,2K5     ,36      ,YES LOAD 5 CODE                    $
,     ,MJO   ,0       ,C13     ,                                   $
,E9   ,TV    ,A       ,P+121   ,STORE NEXT DATA                    $
,     ,IJ    ,A       ,E10     ,STORAGE LOCATION                   $
,E10  ,TV    ,A       ,E22     ,SETUP ORDER TO UNPACK              $
,     ,55    ,P+123   ,Q+6     ,                                   $
,     ,QT    ,2K21    ,A       ,EXTRACT D                          $
```

```
,     ,MP    ,A     ,P+126 ,LOOP COUNT TIMES D                $
,     ,LA    ,A     ,15    ,                                  $
,     ,AT    ,P+123 ,P+123 ,MODIFY M                          $
,     ,RP3   ,2     ,E12   ,                                  $
,E11  ,TP    ,P+123 ,FILL  ,STORE PARAMETER PAIR              $
,E12  ,RJ    ,FILL  ,FILL  ,GO TO SUBROUTINE                  $
,     ,QJ    ,E2    ,E13   ,IS DATA PACKED                    $
,E13  ,TV    ,2K    ,E16   ,YES                               $
,     ,TP    ,2K    ,3K4   ,INDEX TO ZERO                     $
,     ,RS    ,P+124 ,2K1   ,                                  $
,     ,TP    ,P+125 ,A     ,W TO A                            $
,     ,DV    ,2K6   ,P+123 ,NUMBER OF LOCATIONS               $
,     ,ZJ    ,E14   ,E15   ,IS THERE A REMAINDER              $
,E14  ,TP    ,A     ,3K4   ,YES SET INDEX                     $
,     ,MP    ,A     ,2K6   ,                                  $
,     ,TV    ,A     ,E16   ,SET LQ FOR SPECIAL SHIFT          $
,     ,RA    ,P+123 ,2K1   ,INCREASE LOCATION NO              $
,E15  ,RA    ,P+124 ,P+123 ,DETERMINE LAST LOCATION           $
,     ,LA    ,A     ,15    ,                                  $
,     ,TU    ,A     ,E16   ,SET LQ                            $
,E16  ,LQ    ,FILL  ,FILL  ,SHIFT                             $
,     ,TU    ,E16   ,E18   ,                                  $
,     ,TP    ,3K4   ,A     ,                                  $
,     ,ZJ    ,E19   ,E17   ,TEST FOR REMAINDER                $
,E17  ,TP    ,2K6   ,3K4   ,NO SET 6 INDEX                    $
,E18  ,SP    ,FILL  ,0     ,                                  $
,     ,TP    ,A     ,Q     ,WORD TO Q                         $
,E19  ,RS    ,E18   ,2K24  ,                                  $
,E20  ,IJ    ,3K4   ,E21   ,                                  $
,     ,MJO   ,0     ,E17   ,                                  $
,E21  ,IJ    ,P+125 ,E22   ,INDEX ON W                        $
,     ,MJO   ,0     ,E2    ,                                  $
,E22  ,QT    ,2K21  ,FILL  ,EXTRACT TWO CHARACTERS            $
,     ,RS    ,E22   ,2K1   ,                                  $
,     ,LQ    ,Q     ,30    ,                                  $
,     ,MJO   ,0     ,E20   ,                                  $
,F    ,MJO   ,0     ,F2    ,SWITCH 1                          $
,F1   ,TP    ,2K13  ,FILL  ,SET ML SYMBOL                     $
,F2   ,RA    ,3K2   ,2K1   ,INCREASE BLOCKETTE COUNT          $
,     ,RA    ,P+126 ,2K1   ,INCREASE LOOP COUNT               $
,     ,TP    ,P+122 ,A     ,                                  $
,     ,ZJ    ,F7    ,F3    ,DOES N EQUAL ZERO                 $
,F3   ,MJO   ,0     ,F5    ,YES SWITCH 2                      $
,F4   ,TP    ,2K15  ,FILL  ,SET PS SYMBOL                     $
,F5   ,TP    ,P+124 ,A     ,                                  $
,     ,ZJ    ,F6    ,F7    ,DOES FINAL PARAMETER EQUAL ZERO   $
,F6   ,RS    ,F14   ,2K1   ,NO SET SWITCH 3                   $
,F7   ,TU    ,1K3   ,F10   ,CONDENSE BLOCKETTE                $
,     ,TP    ,2K21  ,Q     ,AND STORE IN BLOCK                $
,     ,TP    ,2K25  ,P+123 ,                                  $
,F8   ,TP    ,2K5   ,P+124 ,                                  $
,F9   ,SP    ,FILL  ,6     ,                                  $
,F10  ,QA    ,FILL  ,FILL  ,                                  $
,     ,RA    ,F10   ,2K24  ,                                  $
,     ,IJ    ,P+124 ,F9    ,                                  $
,     ,RA    ,F9    ,2K24  ,                                  $
```

```
,       ,RA      ,F10     ,2K1      ,                        $
,F11    ,IJ      ,P+123   ,F8       ,                        $
,F12    ,TP      ,3K2     ,A        ,                        $
,       ,EJ      ,2K6     ,F13      ,IS BLOCK FULL           $
,       ,MJO     ,O       ,F14      ,NO                      $
,F13    ,RJ      ,D4      ,D1       ,YES WRITE BLOCK         $
,F14    ,MJO     ,0       ,F16      ,SWITCH 3                $
,F15    ,RJ      ,D4      ,DO       ,WRITE BLOCK             $
,F16    ,RP1     ,120     ,E        ,FILL BLOCKETTE          $
,       ,TP      ,2K1     ,P        ,WITH SPACES             $
,1K     ,MJO     ,0       ,F2       ,                        $
,1K1    ,MJO     ,0       ,F5       ,                        $
,1K2    ,MJO     ,0       ,F16      ,                        $
,1K3    ,        ,P       ,         ,                        $
,1K4    ,        ,        ,P+121    ,                        $
,1K5    ,        ,B3      ,B3       ,                        $
,1K6    ,        ,        ,2K8      ,                        $
,2K     ,B       ,        ,0        ,ZERO                    $
,2K1    ,B       ,        ,1        ,ONE                     $
,2K2    ,B       ,        ,2        ,TWO                     $
,2K3    ,B       ,        ,3        ,THREE                   $
,2K4    ,B       ,        ,4        ,FOUR                    $
,2K5    ,B       ,        ,5        ,FIVE                    $
,2K6    ,B       ,        ,6        ,SIX                     $
,2K7    ,B       ,        ,7        ,SEVEN                   $
,2K8    ,B       ,        ,12       ,INDEX                   $
,2K9    ,B       ,        ,37       ,FAST FEED - FF 1        $
,2K10   ,B       ,        ,42       ,FF 2                    $
,2K11   ,B       ,        ,57       ,FF 3                    $
,2K12   ,B       ,        ,76       ,FF 4                    $
,2K13   ,B       ,        ,20       ,MULTILINE - ML          $
,2K14   ,B       ,        ,61       ,BREAKPOINT BP           $
,2K15   ,B       ,        ,60       ,PRINTER STOP            $
,2K16   ,B       ,77777   ,00000    ,MASK U                  $
,2K17   ,B       ,        ,77777    ,MASK V                  $
,2K18   ,B       ,        ,01750    ,DIVISOR                 $
,2K19   ,B02     ,00746   ,00000    ,OFF LINE WRITE BLOCK    $
,2K20   ,B01     ,01010   ,10101    ,CONDENSED SPACES        $
,2K21   ,B       ,        ,77       ,6 BIT MASK              $
,2K22   ,B       ,        ,777      ,9 BIT MASK              $
,2K23   ,B       ,      2 ,00000    ,ADVANCE U               $
,2K24   ,B       ,      1 ,00000    ,ADVANCE U               $
,2K25   ,B       ,        ,23       ,INDEX                   $
,2K26   ,B       ,        ,00144    ,DIVISOR                 $
,2K27   ,B31     ,26373   ,73752    ,                        $
,3K     ,B       ,        ,0        ,STORE DATA LOCATION     $
,3K1    ,B       ,        ,0        ,SERVO NUMBER            $
,3K2    ,B       ,        ,0        ,BLOCKETTE COUNT         $
,3K3    ,B       ,        ,0        ,TAPE COMMAND            $
,3K4    ,        ,        ,0        ,INDEX                   $
,       ,ENDSUB  ,        ,         ,
```

```
,      ,SUB     ,WF0002  ,61     ,                          $
,      ,TEMPS   ,15      ,127    ,                          $
,      ,IN OUT  ,2       ,0      ,                          $
,B     ,MJO     ,0       ,C      ,ENTRY                     $
,B1    ,ALARM   ,        ,       ,EXIT ALARM                $
,B2    ,MJO     ,0       ,FILL   ,EXIT NORMAL               $
,B3    ,B       ,        ,0      ,                          $
,B4    ,B       ,        ,0      ,                          $
,C     ,TP      ,K10     ,Q      ,                          $
,      ,QT      ,B3      ,A      ,EXTRACT WS                $
,      ,DV      ,K9      ,A      ,W TO A                    $
,      ,TJ      ,K8      ,C2     ,IS W TOO LARGE            $
,C1    ,SP      ,B3      ,36     ,YES                       $
,      ,SA      ,B4      ,0      ,LOAD PARAMETERS           $
,      ,TP      ,K12     ,Q      ,AND CODE                  $
,      ,MJO     ,0       ,B1     ,GO TO ALARM EXIT          $
,C2    ,LA      ,A       ,15     ,                          $
,      ,AT      ,K       ,C12    ,SET RP                    $
,      ,TV      ,B4      ,C13    ,                          $
,      ,RP1     ,14      ,C3     ,                          $
,      ,TN      ,K2      ,P+1    ,STORE - 2                 $
,C3    ,TV      ,C3-1    ,C9     ,SET STORE DIGIT LOC       $
,      ,LQ      ,B4      ,Q+21   ,                          $
,      ,QT      ,K3      ,A      ,EXTRACT CODE              $
,      ,EJ      ,K1      ,C14    ,ADDRESS                   $
,C4    ,EJ      ,K2      ,C15    ,SPECIAL FORMAT            $
,C5    ,TP      ,K11     ,P      ,SET 11 INDEX              $
,C6    ,TU      ,B3      ,C7     ,                          $
,C7    ,SP      ,FILL    ,0      ,OBTAIN WORD               $
,      ,TP      ,A       ,Q      ,WORD TO Q                 $
,C8    ,LQ      ,Q       ,3      ,                          $
,C9    ,QT      ,K6      ,FILL   ,STORE DIGIT               $
,      ,RA      ,C9      ,K1     ,                          $
,      ,IJ      ,P       ,C8     ,                          $
,C10   ,RJ      ,C10     ,C11    ,                          $
,C11   ,TP      ,C9      ,Q      ,UNPACKED INDICATOR TO Q   $
,      ,RP2     ,14      ,C12    ,                          $
,      ,RA      ,P+1     ,K3     ,CONVERT TO XS-3           $
,C12   ,RP3     ,        ,B2     ,                          $
,C13   ,TP      ,P+1     ,FILL   ,TRANSFER W RESULTS        $
,C14   ,TP      ,K4      ,P      ,SET 4 INDEX               $
,      ,LQ      ,B3      ,Q+6    ,                          $
,      ,MJO     ,0       ,C8     ,                          $
,C15   ,TP      ,K1      ,P      ,SET 1 INDEX               $
,      ,RJ      ,C10     ,C6     ,                          $
,      ,RA      ,C9      ,K1     ,SKIP FOR SPACE            $
,      ,TP      ,K4      ,P      ,SET 4 INDEX               $
,      ,RJ      ,C10     ,C8     ,                          $
,      ,RA      ,C9      ,K1     ,SKIP FOR SPACE            $
,      ,TP      ,K4      ,P      ,SET 4 INDEX               $
,      ,MJO     ,0       ,C8     ,                          $
,K     ,RP3     ,0       ,B2     ,                          $
,K1    ,B       ,        ,1      ,
```

```
,K2     ,B      ,       ,2      ,        $
,K3     ,B      ,       ,3      ,        $
,K4     ,B      ,       ,4      ,        $
,K5     ,B      ,       ,5      ,        $
,K6     ,B      ,       ,7      ,        $
,K7     ,B      ,       ,16     ,        $
,K8     ,B      ,       ,17     ,        $
,K9     ,B      ,       ,144    ,        $
,K10    ,B      ,       ,77777  ,        $
,K11    ,B      ,       ,13     ,        $
,K12    ,B31    ,26373  ,73774  ,        $
,       ,ENDSUB ,       ,       ,        $
```

```
,       ,SUB     ,WF0003  ,104      ,                        $
,       ,TEMPS   ,16      ,127      ,                        $
,       ,IN OUT  ,2       ,0        ,                        $
,B      ,MJO     ,0       ,C        ,ENTRY                   $
,B1     ,ALARM   ,      0 ,       0 ,EXIT ALARM              $
,B2     ,MJO     ,0       ,FILL     ,EXIT NORMAL             $
,B3     ,B       ,        ,0        ,PARAMETER               $
,B4     ,B       ,        ,0        ,CELLS                   $
,C      ,TP      ,K7      ,Q        ,MASK TO Q               $
,       ,QT      ,B3      ,A        ,EXTRACT WS              $
,       ,DV      ,K18     ,P+13     ,SAVE W                  $
,       ,TP      ,K3      ,A        ,                        $
,       ,TJ      ,P+13    ,C23      ,IS W TOO LARGE          $
,C1     ,LA      ,P+13    ,15       ,NO                      $
,       ,AT      ,J       ,C21      ,SET TRANSFER            $
,       ,TP      ,K       ,P+14     ,                        $
,       ,RP1     ,13      ,C1+5     ,                        $
,       ,TN      ,K1      ,P        ,                        $
,       ,TV      ,B4      ,C22      ,                        $
,       ,LQ      ,B4      ,Q+21     ,SHIFT SS                $
,       ,QT      ,K7      ,P+13     ,SAVE SS                 $
,       ,TP      ,K5      ,A        ,                        $
,       ,TJ      ,P+13    ,C23      ,IS SS TOO LARGE         $
,C2     ,ST      ,P+13    ,P-13     ,NO                      $
,       ,ZJ      ,C3      ,C4       ,IS SS 35                $
,C3     ,MP      ,A       ,K6       ,NO                      $
,       ,SA      ,K8      ,0        ,                        $
,       ,TU      ,A       ,P+14     ,SAVE I DIGITS           $
,C4     ,TP      ,K9      ,A        ,                        $
,       ,ST      ,P+14    ,P+15     ,SAVE F DIGITS           $
,       ,AT      ,J2      ,C11      ,SET DIVIDE              $
,       ,TP      ,J1      ,C10      ,PLACE RP                $
,       ,RA      ,C10     ,P+14     ,SET N                   $
,       ,LQ      ,P+15    ,21       ,SHIFT F DIGITS          $
,       ,TP      ,J3      ,A        ,                        $
,       ,SA      ,P+14    ,57       ,                        $
,       ,TV      ,A       ,C5       ,                        $
,C5     ,TP      ,K4      ,FILL     ,SET POINT               $
,       ,TV      ,A       ,C19      ,SET F LOCATION          $
,       ,RA      ,C19     ,K20      ,                        $
,       ,TV      ,P+13    ,C8       ,SET LA                  $
,       ,RA      ,C8      ,K20      ,SET K                   $
,       ,TU      ,B3      ,C6       ,                        $
,C6     ,TP      ,FILL    ,A        ,OBTAIN WORD             $
,       ,SJ      ,C7      ,C8       ,IS WORD NEGATIVE        $
,C7     ,TN      ,K20     ,P        ,YES SET SIGN            $
,       ,TM      ,A       ,A        ,                        $
,C8     ,LA      ,A       ,FILL     ,SHIFT WORD              $
,       ,TP      ,A       ,P+14     ,SAVE F PART             $
,       ,ZJ      ,C9      ,C15      ,IS WORD ZERO            $
,C9     ,SS      ,A       ,36       ,NO                      $
,C10    ,RP3     ,0       ,C12      ,                        $
,C11    ,DV      ,K10     ,P+1      ,EXTRACT INTEGRAL DIGITS $
,C12    ,TU      ,J3      ,C13      ,                        $
```

```
,        ,TV      ,C11     ,C14     ,                                $
,C13     ,TP      ,FILL    ,A       ,SUPPRESS                        $
,        ,ZJ      ,C17     ,C14     ,HIGH                            $
,C14     ,TN      ,K1      ,FILL    ,ORDER                           $
,        ,RA      ,C13     ,K8      ,ZEROS                           $
,        ,RA      ,C14     ,K20     ,                                $
,        ,MJO     ,0       ,C13     ,                                $
,C15     ,TV      ,C5      ,C16     ,ZERO WORD                       $
,        ,RS      ,C16     ,K20     ,                                $
,C16     ,TP      ,K       ,FILL    ,                                $
,C17     ,IJ      ,P+15    ,C18     ,                                $
,        ,MJO     ,0       ,C20     ,                                $
,C18     ,SP      ,P+14    ,2       ,                                $
,        ,SA      ,P+14    ,1       ,EXTRACT                         $
,        ,TP      ,A       ,P+14    ,                                $
,C19     ,LTO     ,0       ,FILL    ,                                $
,        ,RA      ,C19     ,K20     ,                                $
,        ,MJO     ,0       ,C17     ,                                $
,C20     ,TP      ,C11     ,Q       ,UNPACKED INDICATOR TO Q         $
,        ,RP2     ,13      ,C21     ,CONVERT TO                      $
,        ,RA      ,P       ,K2      ,XS-THREE                        $
,C21     ,RP      ,FILL    ,FILL    ,TRANSFER                        $
,C22     ,TP      ,P       ,FILL    ,W DIGITS                        $
,C23     ,TP      ,K21     ,Q       ,SETUP                           $
,        ,SP      ,B3      ,36      ,FOR                             $
,        ,SA      ,B4      ,0       ,ALARM                           $
,        ,MJO     ,0       ,B1      ,EXIT                            $
,J       ,RP3     ,0       ,B2      ,                                $
,J1      ,RP3     ,0       ,C12     ,                                $
,J2      ,DV      ,K10     ,P+1     ,                                $
,J3      ,        ,P+1     ,        ,                                $
,K       ,B       ,        ,0       ,                                $
,K1      ,B       ,        ,2       ,                                $
,K2      ,B       ,        ,3       ,                                $
,K3      ,B       ,        ,15      ,                                $
,K4      ,B       ,        ,17      ,                                $
,K5      ,B       ,        ,43      ,                                $
,K6      ,B       ,        ,23210   ,                                $
,K7      ,B       ,        ,77777   ,                                $
,K8      ,B       ,      1 ,00000   ,                                $
,K9      ,B       ,     13 ,00000   ,                                $
,K10     ,B11     ,24027   ,62000   ,                                $
,K11     ,B       ,73465   ,45000   ,                                $
,K12     ,B       ,5753    ,60400   ,                                $
,K13     ,B       ,461     ,13200   ,                                $
,K14     ,B       ,36      ,41100   ,                                $
,K15     ,B       ,3       ,03240   ,                                $
,K16     ,B       ,        ,23420   ,                                $
,K17     ,B       ,        ,1750    ,                                $
,K18     ,B       ,        ,144     ,                                $
,K19     ,B       ,        ,12      ,                                $
,K20     ,B       ,        ,1       ,                                $
,K21     ,B31     ,26373   ,73770   ,                                $
,        ,END SUB ,        ,        ,                                $
```

```
,     ,SUB     ,WF0004  ,200     ,                         $
,     ,TEMPS   ,19      ,127     ,                         $
,     ,IN OUT  ,2       ,0       ,                         $
,B    ,MJO     ,0       ,C       ,ENTRY                    $
,B1   ,ALARM   ,        ,        ,EXIT ALARM               $
,B2   ,MJO     ,0       ,FILL    ,EXIT NORMAL              $
,B3   ,B       ,        ,0       ,                         $
,B4   ,B       ,        ,0       ,                         $
,C    ,TP      ,K14     ,Q       ,V MASK                   $
,     ,QT      ,B3      ,A       ,EXTRACT WS               $
,     ,DV      ,K12     ,P       ,STORE W                  $
,     ,TP      ,K8      ,A       ,                         $
,     ,TJ      ,P       ,C27     ,IS W TOO LARGE           $
,C1   ,TP      ,P       ,A       ,NO                       $
,     ,TJ      ,K6      ,C27     ,IS W TOO SMALL           $
,C2   ,ST      ,K5      ,P+1     ,NO                       $
,     ,AT      ,J1      ,C23     ,SET EXPONENT SHIFT       $
,     ,SP      ,P       ,15      ,                         $
,     ,AT      ,J2      ,C25     ,                         $
,     ,TV      ,J3      ,C15     ,SETUP FOR ROUNDING       $
,     ,RA      ,C15     ,P+1     ,                         $
,     ,RP1     ,13      ,C3      ,                         $
,     ,TN      ,K2      ,P+5     ,                         $
,C3   ,TU      ,B3      ,C4      ,                         $
,C4   ,TP      ,FILL    ,A       ,OBTAIN WORD              $
,     ,TP      ,A       ,P       ,AND STORE                $
,     ,ZJ      ,C6      ,C5      ,IS WORD ZERO             $
,C5   ,TP      ,A       ,P+6     ,YES STORE ZERO           $
,     ,MJO     ,0       ,C21+4   ,                         $
,C6   ,TM      ,A       ,Q       ,MAG WORD TO Q            $
,     ,QT      ,K17     ,P+1     ,WORD CHARACTERISTIC      $
,     ,QT      ,K18     ,P+2     ,WORD MANTISSA            $
,     ,LQ      ,P+1     ,9       ,SHIFT CHARACTERISTIC     $
,     ,TJ      ,K15     ,C21+5   ,IS WORD NORMALIZED       $
,C7   ,TP      ,P       ,A       ,WORD TO A                $
,     ,SJ      ,C8      ,C9      ,IS WORD NEGATIVE         $
,C8   ,TN      ,K1      ,P+5     ,YES SET SIGN             $
,     ,TM      ,A       ,A       ,                         $
,C9   ,TP      ,K16     ,P       ,10 TO MINUS 38           $
,     ,RP2     ,77      ,C10     ,                         $
,     ,TJ      ,1K      ,C11     ,                         $
,C10  ,SP      ,1K+77   ,45      ,HIGHEST POWER OF 10      $
,     ,TP      ,A       ,P+3     ,STORE CHARACTERISTIC     $
,     ,SS      ,A       ,27      ,                         $
,     ,TP      ,A       ,P+4     ,STORE MANTISSA           $
,     ,TP      ,K11     ,P       ,10 to 38                 $
,     ,MJO     ,0       ,C13     ,                         $
,C11  ,RS      ,Q       ,K13     ,J,N-R MINUS 20114        $
,     ,RS      ,P       ,Q       ,                         $
,     ,TP      ,J       ,A       ,ADD OF FIRST POWER OF 10 $
,     ,SS      ,Q       ,15      ,LOCATE                   $
,     ,TU      ,A       ,C12     ,APPROPRIATE POWER        $
,C12  ,TP      ,FILL    ,Q       ,OF TEN                   $
```

```
,       ,QT     ,K17    ,P+3     ,STORE CHARACTERISTIC          $
,       ,QT     ,K18    ,P+4     ,STORE MANTISSA                $
,       ,LQ     ,P+3    ,9       ,SHIFT CHARACTERISTIC          $
,C13    ,RS     ,P+1    ,P+3     ,                              $
,       ,RA     ,P+1    ,K10     ,ADD 35                        $
,       ,TV     ,A      ,C14     ,SET SCALING                   $
,C14    ,SP     ,P+2    ,FILL    ,SHIFT WORD MANTISSA           $
,       ,DV     ,P+4    ,P+2     ,DIVIDE BY POWER 10 MANTISSA   $
,C15    ,RA     ,P+2    ,FILL    ,ROUND                         $
,       ,EJ     ,P+2    ,C17     ,IS THERE OVERFLOW             $
,C16    ,RA     ,P      ,K1      ,YES INCREASE EXP              $
,       ,TP     ,K19    ,P+2     ,1 SCALED 35                   $
,C17    ,TP     ,K6     ,P+1     ,SET SEVEN INDEX               $
,       ,LQ     ,P+2    ,1       ,SHIFT FRACTION                $
,       ,TV     ,J1     ,C19     ,SET STORAGE LOCATION          $
,C18    ,SP     ,P+2    ,2       ,MULTIPLY                      $
,       ,SA     ,P+2    ,1       ,BY TEN                        $
,       ,TP     ,A      ,P+2     ,                              $
,C19    ,LTO    ,0      ,FILL    ,                              $
,       ,RA     ,C19    ,K1      ,                              $
,       ,IJ     ,P+1    ,C18     ,8 DIGITS                      $
,       ,TP     ,P+7    ,P+6     ,                              $
,       ,RS     ,P      ,K1      ,EXP MINUS 1                   $
,       ,ZJ     ,C20    ,C21+4   ,IS EXP ZERO                   $
,C20    ,SJ     ,C21    ,C21+2   ,NO IS EXP NEGATIVE            $
,C21    ,TN     ,K1     ,P+15    ,YES SET SIGN OF EXPONENT      $
,       ,TM     ,A      ,A       ,                              $
,       ,DV     ,K7     ,P+16    ,FIRST DIGIT OF EXPONENT       $
,       ,TP     ,A      ,P+17    ,SECOND DIGIT OF EXPONENT      $
,       ,TP     ,K9     ,P+7     ,SET POINT                     $
,       ,RP2    ,13     ,C22     ,CONVERT                       $
,       ,RA     ,P+5    ,K3      ,TO XS-3                       $
,C22    ,RP3    ,3      ,C24     ,SHIFT                         $
,C23    ,TP     ,P+15   ,P+7     ,SIGN AND EXP                  $
,C24    ,TV     ,B4     ,C26     ,SET TRANSFER                  $
,       ,TP     ,C22    ,Q       ,UNPACKED INDICATOR TO Q       $
,C25    ,RP3    ,0      ,B2      ,TRANSFER                      $
,C26    ,TP     ,P+5    ,FILL    ,W DIGITS                      $
,C27    ,SP     ,B3     ,36      ,LOAD                          $
,       ,SA     ,B4     ,0       ,ARGUMENTS                     $
,       ,TP     ,K4     ,Q       ,AND RTN CODE                  $
,       ,MJO    ,0      ,B1      ,GO TO ALARM EXIT              $
,J      ,       ,       ,1K      ,                              $
,J1     ,TP     ,P+15   ,P+7     ,                              $
,J2     ,RP3    ,0      ,B2      ,                              $
,J3     ,       ,       ,K18     ,                              $
,K1     ,B      ,       ,1       ,                              $
,K2     ,B      ,       ,2       ,                              $
,K3     ,B      ,       ,3       ,                              $
,K4     ,B31    ,26373  ,73764   ,WF0004                        $
,K5     ,B      ,       ,5       ,                              $
,K6     ,B      ,       ,7       ,                              $
,K7     ,B      ,       ,12      ,                              $
```

,K8	,B	,	,15	,	$
,K9	,B	,	,17	,	$
,K10	,B	,	,43	,	$
,K11	,B	,	,47	,	$
,K12	,B	,	,144	,	$
,K13	,B	,	,20114	,	$
,K14	,B	,	,77777	,	$
,K15	,B	,4000	,00000	,	$
,K16	,B77	,77777	,77731	,	$
,K17	,B37	,70000	,00000	,	$
,K18	,B40	,07777	,77777	,	$
,K19	,B3	,14631	,46554	,	$
,K20	,B	,12172	,70245	,	$
,K21	,B	,1014	,22336	,	$
,K22	,B	,64	,33343	,	$
,K23	,B	,5	,17427	,	$
,K24	,B	,	,41434	,	$
,K25	,B	,	,3266	,	$
,K26	,B	,	,254	,	$
,1K	,B00	,26634	,37347	,	$
,	,B00	,64201	,63521	,	$
,	,B01	,15242	,20445	,	$
,	,B01	,46512	,64556	,	$
,	,B02	,04116	,60745	,	$
,	,B02	,35142	,35136	,	$
,	,B02	,66373	,04365	,	$
,	,B03	,24034	,72631	,	$
,	,B03	,55044	,11400	,	$
,	,B04	,06255	,13700	,	$
,	,B04	,37730	,36660	,	$
,	,B04	,74747	,23216	,	$
,	,B05	,26141	,10061	,	$
,	,B05	,57571	,32075	,	$
,	,B06	,14653	,70246	,	$
,	,B06	,46026	,66320	,	$
,	,B06	,77434	,44004	,	$
,	,B07	,34561	,66403	,	$
,	,B07	,65716	,24103	,	$
,	,B10	,17301	,71124	,	$
,	,B10	,54471	,13564	,	$
,	,B11	,05607	,36522	,	$
,	,B11	,37151	,26246	,	$
,	,B11	,74401	,65750	,	$
,	,B12	,25502	,23342	,	$
,	,B12	,57022	,70232	,	$
,	,B13	,14313	,63140	,	$
,	,B13	,45376	,57770	,	$
,	,B13	,76676	,33766	,	$
,	,B14	,34227	,01372	,	$
,	,B14	,65274	,61671	,	$
,	,B15	,16553	,76247	,	$
,	,B15	,54143	,36750	,	$
,	,B16	,05174	,26542	,	$
,	,B16	,36433	,34273	,	$

```
,          ,B16       ,74061     ,11565     ,                                                       $
,          ,B17       ,25075     ,34122     ,                                                       $
,          ,B17       ,56314     ,63146     ,                                                       $
,          ,B20       ,14000     ,00000     ,                                                       $
,          ,B20       ,45000     ,00000     ,                                                       $
,          ,B20       ,76200     ,00000     ,                                                       $
,          ,B21       ,27640     ,00000     ,                                                       $
,          ,B21       ,64704     ,00000     ,                                                       $
,          ,B22       ,16065     ,00000     ,                                                       $
,          ,B22       ,47502     ,20000     ,                                                       $
,          ,B23       ,04611     ,32000     ,                                                       $
,          ,B23       ,35753     ,60400     ,                                                       $
,          ,B23       ,67346     ,54500     ,                                                       $
,          ,B24       ,24520     ,13710     ,                                                       $
,          ,B24       ,55644     ,16672     ,                                                       $
,          ,B25       ,07215     ,22450     ,                                                       $
,          ,B25       ,44430     ,23471     ,                                                       $
,          ,B25       ,75536     ,30410     ,                                                       $
,          ,B26       ,27065     ,76512     ,                                                       $
,          ,B26       ,64341     ,57116     ,                                                       $
,          ,B27       ,15432     ,12741     ,                                                       $
,          ,B27       ,46740     ,55532     ,                                                       $
,          ,B30       ,04254     ,34430     ,                                                       $
,          ,B30       ,35327     ,43536     ,                                                       $
,          ,B30       ,66615     ,34466     ,                                                       $
,          ,B31       ,24170     ,31702     ,                                                       $
,          ,B31       ,55226     ,40262     ,                                                       $
,          ,B32       ,06474     ,10336     ,                                                       $
,          ,B32       ,44105     ,45213     ,                                                       $
,          ,B32       ,75126     ,76456     ,                                                       $
,          ,B33       ,26354     ,56171     ,                                                       $
,          ,B33       ,64023     ,74714     ,                                                       $
,          ,B34       ,15030     ,74077     ,                                                       $
,          ,B34       ,46237     ,13116     ,                                                       $
,          ,B34       ,77706     ,75742     ,                                                       $
,          ,B35       ,34734     ,26555     ,                                                       $
,          ,B35       ,66123     ,34311     ,                                                       $
,          ,B36       ,17550     ,23373     ,                                                       $
,          ,B36       ,54641     ,14135     ,                                                       $
,          ,B37       ,06011     ,37164     ,                                                       $
,          ,B37       ,37413     ,67021     ,                                                       $
,          ,B37       ,74547     ,32313     ,                                                       $
,          ,B40       ,25701     ,20775     ,                                                       $
,          ,END SUB   ,          ,          ,                                                       $
```

```
,       ,SUB     ,WF0005   ,24       ,                                $
,       ,TEMPS   ,1        ,127      ,                                $
,       ,IN OUT  ,2        ,0        ,                                $
,B      ,MJO     ,0        ,C        ,ENTRY                           $
,B1     ,ALARM   ,        0,        0,EXIT ALARM                      $
,B2     ,MJO     ,0        ,FILL     ,EXIT NORMAL                     $
,B3     ,B       ,         ,0        ,PARAMETER                       $
,B4     ,B       ,         ,0        ,CELLS                           $
,C      ,TU      ,K4       ,C4       ,PRESETS                         $
,       ,TU      ,B3       ,C5       ,                                $
,       ,TP      ,K3       ,Q        ,                                $
,       ,QT      ,B3       ,A        ,                                $
,       ,DV      ,K2       ,A        ,OBTAIN W IN A                   $
,       ,DV      ,K1       ,P        ,DIVIDE BY 6                     $
,C1     ,ZJ      ,C2       ,C3       ,IS THERE A REMAINDER            $
,C2     ,RA      ,P        ,K        ,YES INCREASE QUOTIENT           $
,C3     ,LA      ,P        ,15       ,NO                              $
,       ,AT      ,C4       ,C4       ,FORM N OF REPEAT                $
,       ,TV      ,B4       ,C5       ,                                $
,       ,TP      ,C        ,Q        ,PACKED INDICATOR TO Q           $
,C4     ,RP3     ,         ,B2       ,                                $
,C5     ,TP      ,FILL     ,FILL     ,STORE DATA IN EDIT ROUTINE      $
,K      ,B       ,         ,1        ,                                $
,K1     ,B       ,         ,6        ,                                $
,K2     ,B       ,         ,144      ,                                $
,K3     ,B       ,         ,77777    ,                                $
,K4     ,B       ,30000    ,00000    ,                                $
,       ,END SUB ,         ,         ,                                $
```

1. IDENTIFICATION

APSD - SPECTRAL DENSITY
Harry Shaw - October 15, 1957
Applied Physics Laboratory

2. PURPOSE

Given M + 1 lags (see APAC - AUTOCORRELATION) RH, H = 0(1)M, compute M + 1 values UJ, J = 0(1)M, of what is essentially the spectral density function.

3. METHOD

a. The method of Tukey (see The Sampling Theory of Power Spectra Estimates, Symposium on Applications of Autocorrelation Analysis to Physical Problems, Woods Hole, Mass., 1949) is used to compute

$$UJ \doteq \frac{\varphi(J/2M\Delta\tau)}{2M\Delta\tau}$$

where φ is the spectral density function

$$\varphi(f) = 4\int_0^\infty R(t)\cos 2\pi f\tau \, d\tau.$$

Since

$$\sum_{J=0}^{M} UJ \sim R0$$

furnishes a convenient check, the sum of the UJ is also computed.

b. Inputs RH are integers scaled zero and can be computed by APAC (if obtained in some other way care should be taken that $|RH| < 2^{29}$). Outputs are integers scaled zero.

4. USAGE

a. Calling Sequence

LOC	OP	U	V
r	RJ	t + 2	t
r + 1	Normal	Return	

b. Control and Results

Enter with the code word 00 OMMMM WWWWW in the accumulator, and lags RH in locations W + H.

Exit with UJ in locations W + M + 1 + J, ΣUJ in location W + 2M + 2, and locations W-1 thru W + M unchanged.

c. Space Required

172 cells of instructions and constants

13 cells of temporary storage

Block of 3M + 3 locations beginning at core address W-1 (includes data mean and lag storage)

d. Error Codes

None

5. RESTRICTIONS

See 3.b.

6. TIMING

Approximately M^2 mls.

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,SUB	,APSD	,172	,LEADING LINE	$
	,	,TEMPS	,13	,0	,TEMPS	$
	,T	,MJO	,	,P100	,ENTRANCE	$
	,	,ALARM	,	,	,ALARM EXIT	$
	,	,MJO	,	,	,NORMAL EXIT	$
	,P100	,TV	,A	,U101	,GENERATE	$
	,	,LT	,25)B	,U100	,V - ADDRESSES	$
	,	,SP	,U101	,	,	$
	,	,AT	,K1	,U300	,	$
	,	,AT	,U100	,U301	,	$
	,	,AT	,K1	,U302	,	$
	,	,AT	,U100	,U303	,	$
	,	,AT	,K1	,U304	,	$
	,	,AT	,K1	,U103	,	$
	,	,SP	,U302	,	,	$
	,	,AT	,K1	,U305	,	$
	,	,TV	,U302	,P205	,SET	$
	,	,TV	,U300	,P215	,V - ADDRESSES	$
	,	,TV	,U303	,P226	,	$
	,	,TV	,U300	,P228	,	$
	,	,TV	,U304	,P264	,	$
	,	,TV	,U304	,P302	,	$
	,	,TV	,U305	,P307	,	$
	,	,TV	,U301	,P320	,	$
	,	,TV	,U304	,P321	,	$
	,	,TV	,U301	,P344	,	$
	,	,TV	,U303	,P363	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,SP	,U100	,	,	$
	,	,ST	,U210	,U102	,	$
	,	,LQ	,U300	,17)B	,GENERATE	$
	,	,LQ	,U301	,17)B	,U - ADDRESSES	$
	,	,LQ	,U302	,17)B	,	$
	,	,LQ	,U305	,17)B	,	$
	,	,SP	,U101	,17)B	,	$
	,	,TP	,A	,U104	,	$
	,	,SP	,U100	,17)B	,	$
	,	,AT	,U104	,U304	,	$
	,	,LQ	,U303	,17)B	,	$
	,	,RS	,U303	,K2	,	$
	,	,TU	,U104	,P200	,SET	$
	,	,TU	,U104	,P222	,U - ADDRESSES	$
	,	,TU	,U104	,P304	,	$
	,	,TU	,U304	,P164	,	$
	,P164	,TP	,FILL	,U104	,	$
	,	,TU	,U304	,P201	,	$
	,	,TU	,U300	,P204	,	$
	,	,TU	,U300	,P215	,	$
	,	,TU	,U304	,P219	,	$
	,	,TU	,U304	,P221	,	$
	,	,TU	,U300	,P225	,	$
	,	,TU	,U300	,P228	,	$
	,	,TU	,U300	,P302	,	$
	,	,TU	,U305	,P320	,	$
	,	,TU	,U303	,P321	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,TU	,U301	,P340	,	$
	,	,TU	,U305	,P341	,	$
	,	,TU	,U302	,P343	,	$
	,	,TU	,U301	,P362	,	$
	,	,SP	,U100	,17)B	,SET	$
	,	,AT	,U208	,A	,REPEATS	$
	,	,TU	,A	,P203	,	$
	,	,TU	,A	,P224	,	$
	,	,TU	,A	,P361	,	$
	,	,AT	,U209	,A	,	$
	,	,TU	,A	,P301	,	$
	,	,RS	,P203	,K2	,	$
	,	,RS	,P224	,K2	,	$
	,	,RA	,P361	,K2	,	$
	,	,RS	,P301	,K2	,	$
	,P200	,TP	,FILL	,A	,LO (SO)	$
	,P201	,AT	,FILL	,A	,TO	$
	,	,LT	,43)B	,A	,W + M + 2	$
	,P203	,RP	,FILL	,P205	,	$
	,P204	,AT	,FILL	,A	,	$
	,P205	,DV	,U100	,FILL	,	$
	,	,SP	,U100	,43)B	,LM (SO)	$
	,	,TP	,A	,Q	,TO	$
	,	,LT	,	,A	,W + 2M + 2	$
	,	,ST	,K1	,U101	,	$
	,	,TP	,U101	,U400	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,P215	,TN	,FILL	,FILL	,	$
	,	,RA	,P215	,U200	,	$
	,	,IJ	,U400	,P215	,	$
	,	,QJ	,P219	,P221	,	$
	,P219	,TN	,FILL	,A	,	$
	,	,MJ	,	,P222	,	$
	,P221	,TP	,FILL	,A	,	$
	,P222	,AT	,FILL	,A	,	$
	,	,LT	,43)B	,A	,	$
	,P224	,RP	,FILL	,P226	,	$
	,P225	,AT	,FILL	,A	,	$
	,P226	,DV	,U100	,FILL	,	$
	,	,TP	,U101	,U400	,	$
	,P228	,TN	,FILL	,FILL	,	$
	,	,RA	,P228	,U200	,	$
	,	,IJ	,U400	,P228	,	$
	,	,SP	,U205	,	,COS PI/M (S30)	$
	,	,DV	,U100	,Q	,	$
	,	,MP	,Q	,Q	,	$
	,	,LT	,6	,U300	,	$
	,	,MP	,U300	,U300	,	$
	,	,LT	,6	,U301	,	$
	,	,MP	,U300	,U301	,	$
	,	,LT	,6	,U302	,	$
	,	,SP	,U201	,36)B	,	$
	,	,MA	,U300	,U202	,	$
	,	,MA	,U301	,U203	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,MA	,U302	,U204	,	$
	,	,LT	,6	,U302	,INITIALIZE	$
	,	,LT	,1	,U300	,2 COS PI/M	$
	,	,TP	,U201	,U301	,	$
	,	,SP	,U100	,	,SET OUTSIDE LOOP	$
	,	,ST	,K1	,U400	,FOR M-1 PASSES	$
	,P260	,IJ	,U400	,P262	,M-1 PASSES	$
	,	,MJ	,	,P320	,TEST	$
	,P262	,MP	,U300	,U301	,YH1 (S30)	$
	,	,LT	,6)B	,A	,	$
	,P264	,ST	,U302	,FILL	,	$
	,	,TP	,U301	,U302	,H + 1 TO H	$
	,	,TP	,A	,U301	,	$
	,	,TP	,A	,U304	,INITIALIZE	$
	,	,TP	,U201	,U305	,	$
	,	,LT	,10001)B	,U303	,2 YH1	$
	,	,TP	,U102	,U401	,SET M-2 PASSES	$
	,	,TV	,U103	,P282	,SET STORE YHP	$
	,P280	,MP	,U303	,U304	,YHP TO W + 2M + 2 + P	$
	,	,LT	,6)B	,A	,	$
	,P282	,ST	,U305	,FILL	,	$
	,	,TP	,U304	,U305	,P + 1 TO P	$
	,	,TP	,A	,U304	,	$
	,	,RA	,P282	,K1	,	$
	,	,IJ	,U401	,P280	,M-2 PASSES TEST	$
	,	,SP	,K0	,	,LH TO W + M + 2 + H	$
	,P301	,RP	,FILL	,P303	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,P302	,MA	,FILL	,FILL	,	$
	,P303	,LT	,7)B	,A	,	$
	,P304	,AT	,FILL	,A	,	$
	,	,TN	,U104	,U104	,	$
	,	,AT	,U104	,A	,	$
	,P307	,DV	,U100	,FILL	,	$
	,	,RA	,P307	,K1	,STEP LH STORE	$
	,	,MJ	,	,P260	,JUMP TO TEST	$
	,P320	,TP	,FILL	,FILL	,L1 TOL,-1	$
	,P321	,TP	,FILL	,FILL	,	$
	,	,TP	,U100	,U400	,SET + M + 1 PASSES	$
	,P340	,TP	,FILL	,A	,UJ TO W + M + 1 + J	$
	,P341	,AT	,FILL	,A	,	$
	,	,MP	,A	,U206	,	$
	,P343	,MA	,FILL	,U207	,	$
	,P344	,LT	,1)B	,FILL	,	$
	,	,RA	,P340	,K2	,	$
	,P341	,RA	,P341	,K2	,	$
	,	,RA	,P343	,K2	,	$
	,	,RA	,P344	,K1	,	$
	,	,IJ	,U400	,P340	,M + 1 PASSES TEST	$
	,	,TP	,KO	,A	,SUM UJ	$
	,P361	,RP	,FILL	,P363	,TO	$
	,P362	,AT	,FILL	,A	,W + 2M + 2	$
	,P363	,TP	,A	,FILL	,	$
	,	,MJ	,	,T + 2	, JUMP TO EXIT	$
	,U200	,00	,2	,2	, U UP 2 V UP P	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,U201	,01	,	,	,1 (S30)	$
	,U202	,77	,37777)B	,77777)B	,-½ (S30)	$
	,U203	,00	,2525)B	,25252)B	,¼ (S30)	$
	,U204	,77	,77722)B	,37223)B	,- 1/6 (S30)	$
	,U205	,03	,11037)B	,55242)B	,PI (S30)	$
	,U206	,07	,27024)B	,36561)B	,.23 (S35)	$
	,U207	,21	,21727)B	,2437)B	,.54 (S35)	$
	,U208	,00	,20000)B	,	,FOR	$
	,U209	,00	,10000)B	,	,PRESETS	$
	,U210	,00	,	,	,	$
	,K0	,00	,	,	,ZERO	$
	,K1	,00	,	,1	,V - ADVANCE	$
	,K2	,00	,1	,	,U - ADVANCE	$
	,	,ENDSUB	,	,	,	$

1. IDENTIFICATION

APAC - AUTOCORRELATION
Harry Shaw - October 15, 1957
Applied Physics Laboratory

2. PURPOSE

Given N data XJ, J = 1(1)N, compute data mean $\overline{X}$ and M + 1 lags RH, H = 0(1)M.

Talmadge

3. METHOD

a. $$R(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_0^T Y(t)\, Y(t+\tau)\, d\tau$$

is approximated by

$$R(H\Delta t) \doteq RH = \frac{1}{N-H} \sum_{K=1}^{N-H} YK.Y_{\overline{K+H}}$$

where

$YJ = XJ - \overline{X}$, J = 1(1)N.

b. Inputs XJ are integers scaled zero such that $|XJ| \leq 9999$.

Outputs are integers scaled zero.

4. USAGE

a. Calling Sequence

LOC	OP	U	V
r	RJ	t + 2	t
r + 1	Normal	Return	

b. Control and Results

Enter with the code word 00 OMMMM WWWWW in the accumulator, the code word 00 NNNNN [00000 - Core Mode / VVVVV - Drum Mode] in the Q-register, and the data XJ in locations W + J-1 for the Core Mode or V + J-1 for the Drum Mode.

Exit with the data mean $\overline{X}$ in location W-1 and the lags RH in locations W + H. The code word 00 OMMMM WWWWW is left in the accumulator (See APSD - Spectral Density).

c. Space Required

160 cells of instructions and constants

13 cells of temporary storage

Core Mode - block of N + M + 2 locations beginning at core address W-1

Drum Mode - block of N locations beginning at drum address V and block of 3M + 2 locations beginning at core address W-1.

d. Error Codes

None

5. RESTRICTIONS

See 3.b.

6. TIMING

Approximately 1/2 MN μs.

ITEM NO.	LOC.	OP	U	V	COMMENTS	
	,	, SUB	, APAC	, 160	, LEADING LINE	$
	,	, TEMPS	, 13	, 0	, TEMPS	$
	, T	, MJO	,	, P100	, ENTRANCE	$
	,	, ALARM	,	,	, ALARM EXIT	$
	,	, MJO	,	,	, NORMAL EXIT	$
	, P100	, TP	, A	, U100	, SAVE 1ST CD WD	$
	,	, TV	, A	, U101	, W (SO)	$
	,	, TV	, Q	, U103	, V (SO)	$
	,	, TU	, A	, U106	, M (S15)	$
	,	, TU	, Q	, U108	, N (S15)	$
	,	, LA	, A	, 17)B	, W (S15)	$
	,	, TU	, A	, U102	,	$
	,	, LQ	, Q	, 17)B	, V (S15)	$
	,	, TU	, Q	, U104	,	$
	,	, LT	, 6	, U105	, M (SO)	$
	,	, LQ	, Q	, 6	, N (SO)	$
	,	, TV	, Q	, U107	,	$
	,	, TU	, U102	, P401	, PRESET INST	$
	,	, TU	, U102	, P406	, COMMON TO	$
	,	, TV	, U101	, P401	, BOTH MODES	$
	,	, TV	, U101	, P501	,	$
	,	, RS	, P401	, K1	,	$
	,	, TP	, U109	, Q	,	$
	,	, QS	, U106	, P500	,	$
	,	, RA	, P500	, K2	,	$
	,	, SP	, U103	,	, MODE TEST	$
	,	, ZJ	, P300	, P200	,	$

ITEM NO.	LOC.	OP	U	V	COMMENTS	
	,P200	,QS	,U108	,P251	, CORE MODE PRESET	$
	,	,QS	,U108	,P400	,	$
	,	,QS	,U108	,P405	,	$
	,	,QS	,P500	,P254	,	$
	,	,TP	,M400	,P414	,	$
	,	,TU	,U102	,P408	,	$
	,	,RA	,P408	,U108	,	$
	,	,TU	,U102	,P501	,	$
	,	,RA	,P501	,U108	,	$
	,	,TV	,U101	,P252	,	$
	,	,TV	,P401	,P253	,	$
	,	,TV	,U101	,P255	,	$
	,	,RA	,P255	,U107	,	$
	,	,TV	,M500	,P410	,	$
	,	,TP	,KO	,A	, CORE MODE	$
	,P251	,RP	,10000)B	,P253	,	$
	,P252	,RA	,A	,FILL	,	$
	,P253	,DV	,U107	,FILL	,	$
	,P254	,RP	,10000)B	,P256	,	$
	,P255	,TP	,KO	,FILL	,	$
	,P256	,MJ	,	,P400	, JUMP TO E	$
	,P300	,TP	,U107	,A	, DRUM MODE PRESET	$
	,	,DV	,U105	,U300	,	$
	,	,LT	,10017)B	,U301	,	$
	,	,TP	,U109	,Q	,	$
	,	,QS	,A	,P356	,	$
	,	,QS	,A	,P358	,	$

ITEM NO.	LOC	OP	U	V		COMMENTS
	,	,QS	,A	, P601	,	$
	,	,RA	,P601	,U106	,	$
	,	,QS	,U106	,P351	,	$
	,	,QS	,U106	,P353	,	$
	,	,QS	,U106	,P361	,	$
	,	,RA	,P361	,K2	,	$
	,	,SP	,U106	,1	,	$
	,	,QS	,A	,P369	,	$
	,	,TV	,U101	,P350	,	$
	,	,TV	,U101	,P352	,	$
	,	,TV	,U101	,P354	,	$
	,	,TV	,U101	,P357	,	$
	,	,TV	,U101	,P359	,	$
	,	,TV	,U101	,P360	,	$
	,	,TV	,U101	,P370	,	$
	,	,TV	,U101	,P602	,	$
	,	,RS	,P350	,K1	,	$
	,	,RS	,P360	,K1	,	$
	,	,SP	,U105	,1	,	$
	,	,RA	,A	,U101	,	$
	,	,TV	,A	,P362	,	$
	,	,TU	,U104	,P352	,	$
	,	,TU	,U104	,P370	,	$
	,	,TU	,U102	,P354	,	$
	,	,TU	,U102	,P359	,	$
	,	,RS	,P354	,K2	,	$
	,	,RS	,P359	,K2	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,TP	,U108	,A	,	$
	,	,RS	,A	,U301	,	$
	,	,RA	,A	,U104	,	$
	,	,TU	,A	,P357	,	$
	,	,RS	,A	,U106	,	$
	,	,TU	,A	,P602	,	$
	,	,RS	,U300	,K1	,	$
	,	,TP	,U300	,U301	,	$
	,	,RS	,U301	,K1	,	$
	,P350	,TP	,K0	,FILL	, DRUM MODE	$
	,P351	,RP	,30000)B	,P353	,	$
	,P352	,TP	,FILL	,FILL	,	$
	,P353	,RP	,10000)B	,P355	,	$
	,P354	,RA	,FILL	,FILL	,	
	,P355	,IJ	,U300	,P375	,	$
	,	,TU	,P359	,P3552	,	$
	,P3552	,TP	,FILL	,A	,	$
	,P356	,RP	,30000)B	,P358	,	$
	,P357	,TP	,FILL	,FILL	,	$
	,P358	,RP	,10000)B	,P360	,	$
	,P359	,RA	,FILL	,FILL	,	$
	,P360	,DV	,U107	,FILL	,	$
	,P361	,RP	,10000)B	,P363	,	$
	,P362	,TP	,K0	,FILL	,	$
	,P363	,TP	,U109	,Q	,	$
	,	,QS	,U106	,P405	,	$
	,	,SP	,U106	,1	,	$
	,	,QS	,A	,P400	,	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,	,TP	,P415	,P414	,	$
	,	,TV	,M600	,P410	,	$
	,P369	,RP	,30000)B	,P371	,	$
	,P370	,TP	,FILL	,FILL	,	$
	,P371	,SP	,U106	,1	,	$
	,	,RA	,A	,U102	,	$
	,	,TU	,A	,P408	,	$
	,	,MJ	,	,P400	,JUMP TO E	$
	,P375	,RA	,P352	,U106	,	$
	,	,MJ	,	,P351	,LOOP	$
	,P400	,RP	,20000)B	,P402	,E1	$
	,P401	,RS	,FILL	,FILL	,	$
	,P402	,TP	,U107	,U401	,	$
	,	,TV	,U101	,P406	,	$
	,	,TP	,U105	,U400	,	$
	,P4041	,TP	,KO	,A	,	$
	,P405	,RP	,30000)B	,P407	,	$
	,P406	,MA	,FILL	,FILL	,	$
	,P407	,DV	,U401	,Q	,	$
	,P408	,RA	,FILL	,Q	,	$
	,	,IJ	,U400	,P411	,	$
	,P410	,MJ	,	,FILL	, JUMP TO E	$
	,P411	,RS	,U401	,K1	,	$
	,	,RA	,P406	,K1	,	$
	,[illegible]	,RA	,P408	,K2	,	$
	,P414	,RS	,P405	,K2	,	$
	,P415	,MJ	,	,P404 1	, LOOP	$

ITEM NO.	LOC	OP	U	V	COMMENTS	
	,P500	,RP	,30000)B	,P502	,E2	$
	,P501	,TP	,FILL	,FILL	,	$
	,P502	,TP	,U100	,A	,	$
	,	,MJ	,	,T + 2	,JUMP TO EXIT	$
	,P600	,IJ	,U301	,P613	,E3	$
	,P601	,RP	,30000)B	,P603	,	$
	,P602	,TP	,FILL	,FILL	,	$
	,P603	,TP	,U109	,Q	,	$
	,	,QS	,P601	,P400	,	$
	,	,TU	,P601	,P405	,	$
	,	,TP	,M400	,P414	,	$
	,	,SP	,U106	,1	,	$
	,	,RA	,A	,U102	,	$
	,	,TU	,A	,P408	,	$
	,	,TU	,A	,P501	,	$
	,	,TV	,M500	,P410	,	$
	,	,MJ	,	,P400	,JUMP TO E	$
	,P613	,RA	,P370	,U106	,	$
	,	,MJ	,	,P369	,JUMP TO E	$
	,M400	,RS	,P405	,K2	,PRESET	$
	,M500	,00	,	,P500	,E	$
	,M600	,00	,	,P600	,E	$
	,U109	,00	,7777)B	,	,MASK	$
	,K0	,00	,	,	,ZERO	$
	,K1	,00	,	,1	,V - ADVANCE	$
	,K2	,00	, 1	,	,U - ADVANCE	$
	,	,ENDSUB	,	,	,	$

1103-A Subroutine

1. Identification: HOUSE1
   Use Compiler Tape Dump
   Keipert, Tantzen, July 57
   Holloman Air Development Center

2. Purpose: Punch USE compiler side-by-side output on cards for listing on the IBM 407. Useful if the high speed printer is down or for installations having none.

3. Method: The first 80 characters of each blockette are punched on a first card, the next 36 characters are punched in columns 37-72 on a second card. The second card contains also a control punch to facilitate printing all 116 characters on one line. The last 4 characters of each blockette are ignored. The punching of the second card is optional.

4. Usage: The routine is a service routine. For purposes of distribution through USE it has been written as a standard subroutine, with one exception, the control words are in A and Q only. In the subroutine form the usage is:

   Place Tape Unit number in A_{0-4}
   Set Q_{35} = 0 for punching 2 cards per blockette
   Q_{35} = 1 for punching 1 card per blockette
   Then RJ HOUSE1+2 HOUSE1
   No manual cycling of Bull necessary.

5. Restrictions:

   5.1. Control Punch. The second card will have a punch in row 12, column 7. The column may be changed easily by taking a different constant C+7.

   5.2. Special Characters. There are 10 characters on the 407, which will not be identical for different machines. The compiler tape has 6 characters which have to be correlated to 6 of the 10 hole combinations. This is done by 6 bioctal special codes SC through SC+5. As coded now, the correspondence is:

Characters	Holes
plus	12
period	12-8-3
parenthesis close	12-8-4
comma	0-8-3
space	none
parenthesis open	0-8-4

If your 407 is different, substitute other codes according to the following correspondence table.

Holes wanted,	Special octal code.
12	01
11	02
8-3	15
8-4	16
12-8-3	35
12-8-4	36
11-8-3	55
11-8-4	56
0-8-3	75
0-8-4	76

The end of line character is always punched as 11-8-3.

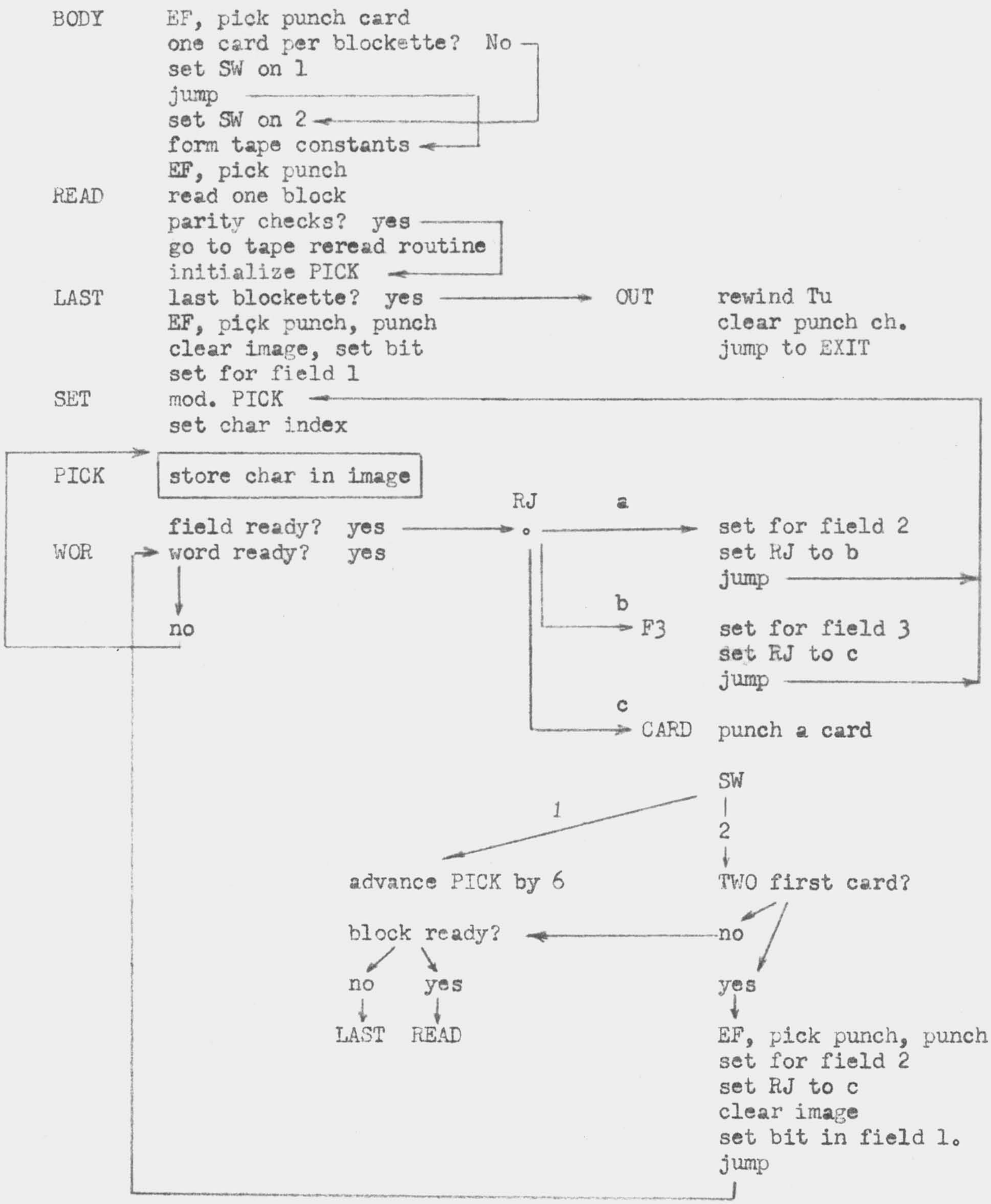
5.3. Wiring board. Use a standard 80-80 board. If wanted, use control punch on second card to print columns 37-72 of second card into paper columns 81-116.

5.4. Parity error. If a parity error is found, the routine will automatically go to the parity error routine HOSP11 (see Useful Note #13). This is accomplished by the three commands PAR+2,3, and 4. If this routine is not available at your installation, replace those 3 commands by something else.

6. Coding Information:

Space required in core to operate - 323 cells. Cards are punched at a rate of 110 cards/min. Routine is machine checked.

FLOW CHART


BODY
EF, pick punch card
one card per blockette? No
set SW on 1
jump
set SW on 2
form tape constants
EF, pick punch
READ
read one block
parity checks? yes
go to tape reread routine
initialize PICK
LAST
last blockette? yes
OUT
rewind Tu
clear punch ch.
jump to EXIT
EF, pick punch, punch
clear image, set bit
set for field 1
SET
mod. PICK
set char index
PICK
store char in image
RJ
a
field ready? yes
set for field 2
set RJ to b
jump
WOR
word ready? yes
no
b
F3
set for field 3
set RJ to c
jump
c
CARD
punch a card
SW
1
2
advance PICK by 6
TWO first card?
block ready?
no
no
yes
yes
LAST
READ
EF, pick punch, punch
set for field 2
set RJ to c
clear image
set bit in field 1.
jump

FLOW CHART

DETAIL: STORE CHARACTER IN IMAGE

```
          Set cmd for row 8
                  for row 4
                  for zone
                  for digits
PICK      shift for next character
          mask character out
          shift bit one right
          blank?  yes
TEST      test for special codes
          change if necessary
REGION    code < 20B?  yes
          modify PUN
          code - 20B

PUN       bit to zone
          code < 15B?  yes
          bit to row 8
          code = 16B?  yes
          set for bit to row 3
SPECIL    bit to row 4 or 3
          jump
          set cmd to store bit
NORMAL    bit to row 12 - 9
```

LOC	OP	U ADDR		V ADDR		REMARKS
	B		20			HOUSE1
	X		126			PUNCH USE COMPILER
	X		31			SIDE-BY-SIDEOUTPUT
	X		166			ON CARDS
		0				KEIPERT,TANTZEN
		0				JULY1957
START	MJ	0		BODY		ENTRY
		FILL		FILL		NOT USED
	MJ	0		FILL		EXIT
BODY	EF	W-	1	R+	1	PP
	QJ	BODY+	2	BODY+	4	ONE OR TWO CARDS?
	TV	EST+	1	SW		SET FOR 1 CARD
	MJ			READ-	4	
	TV	EST+	2	SW		SET FOR 2 CARDS
	LA	A		12		FORM
	AT	R+	6	T+	4	TAPE
	AT	R		T		CONSTANTS
	EF	0		R+	1	PP
READ	EF	0		T		READ
	RPV	120		PAR		ONE BLOCK
	ERB	0		W		
PAR	ERA	0		A		
	ZJ	PAR+	2	PAR+	5	PARITY?
	TP	READ		HOSP11+	3	GO
	TP	READ+	1	HOSP11+	4	TO
	RJ	HOSP11+	2	HOSP11		REREAD
	TU	BODY		PICK		INITIALIZE
LAST	TU	PICK		LAST+	2	
	RA	LAST+	2	C+	3	
	TP	FILL		A		
	EJ	R+	4	OUT		LAST BLOCKETTE?
	EF	0		R+	2	PP,P
CL	RPV	39		CL+	2	CLEAR
	TP	C+	1	CR		IMAGE
	TP	C+	2	T+	1	SET BIT
	TP	CON		T+	2	SET FIELD 1 ADDR
	TP	C+	2	IND		1ST CARD IND
SET	RA	PICK		C+	3	
	TP	C1+	2	IND+	1	CHAR IND
	TU	T+	2	PUN+	3	
	RA	PUN+	3	C+	3	SET FOR ROW 8
	AT	C+	4	SPECIL		FOR ROW 4
	AT	C+	5	PUN		ZONE
	TP	A		NORMAL		DIGIT
PICK	LQ	FILL		6		NEXT CHAR
	QT	R+	3	T+	3	MASK
	LQ	T+	1	35		SHIFT BIT 1 RIGHT
	ZJ	TEST		NORMAL+	1	BLANK? NO, YES
TEST	EJ	CE		CH		CHECK
	EJ	CE+	1	CH1		

```
LOC       OP        U ADDR            V ADDR      REMARKS

          EJ     CE+      2    CH2                FOR
          EJ     CE+      3    CH3
          EJ     CE+      4    CH5                SPECIAL
          EJ      C+      2    CH4                CODES
          MJ       0         REGION
CH        TP     CE+      5       T+         3    REPL 63B BY 01B
          MJ       0         REGION
CH1       TP     CE+      6       T+         3    REPL 22B BY 35B
          MJ       0         REGION
CH2       TP     CE+      7       T+         3    REPL 73B BY 36B
          MJ       0         REGION
CH3       TP     CE+      8       T+         3    REPL 21B BY 75B
          MJ       0         REGION
CH4       TP     CE+      9       T+         3    REPL 01 BY 00B
          MJ       0         NORMAL+         1
CH5       TP     CE+     10       T+         3    REPL 17B BY 76B
REGION    TP      T+      3        A
          TJ      C1             PUN              LESS 20B? YES
          RS     PUN              C+         3    MODIFY ZONE
          RS      T+      3       C1              CODE-20B
          MJ       0         REGION+         1
PUN       CC    FILL              T+         1    BIT TO ZONE
          TP      T+      3        A
          TJ     C1+      1  NORMAL-         2    NORMAL CODE? YES
          CC    FILL              T+         1    BIT TO ROW 8
          TP     C1+      1        A              15B TO A
          TJ      T+      3  SPECIL               CODE 16B? YES
          RA  SPECIL              C+         3    SET FOR ROW 3
SPECIL    CC    FILL              T+         1    BIT TO ROW 4 OR 3
          MJ       0         NORMAL+         1
         LTR      15              T+         3    SET
          RS  NORMAL              T+         3    FOR ROW
NORMAL    CC    FILL              T+         1    BIT TO ROW
          LQ       Q              35              TEST
          QJ NORMAL+      3      WOR              FIELD DONE? YES,NO
          RJ NORMAL+      3  NORMAL+         4
          RA      T+      2      C1+         4    SET FOR FIELD 2
          TV    CON+      1  NORMAL+         3    SET RJ TO B
          MJ       0             SET
F3        RA      T+      2      C1+         4    SET FOR FIELD 3
          LQ      T+      1        8
          TV    CON+      2  NORMAL+         3    SET RJ TO C
          MJ       0             SET
WOR       IJ    IND+      1     SET+         2    WORD READY? NO
          MJ       0             SET
CARD     RPB       3           CARD+         2    SET UP WRITE
          TV    CON+      3    WRITE              CMDS
          TP     C1+      3        Q              ROW INDEX = 11
WRITE    EWA       0            FILL              WRITE
         EWB       0            FILL              NEXT
```

```
LOC      OP           U ADDR           V ADDR      REMARKS

         EWB       0               FILL           ROW
         RPU       3               WRITE+    5
         RA     WRITE                 C+     2    MODIFY
         IJ        Q               WRITE          CARD READ? NO
SW       MJ        0               FILL           JUMP IF 2 CARDS
         RA     PICK                  C+     6    ADVANCE PICK BY 6
         EJ      EST               READ           BLOCK READY? YES
         MJ        0               LAST
TWO      IJ      IND                TWO+     3    FIRST CARD? YES
         TP     PICK                  A           PICK CMD TO A
         MJ        0                 SW+     2
         EF                           R+     2    PP,P
         TP      CON                  T+     2    SET FOR
         RA       T+      2          C1+     4    FIELD 2
         TV     CON+      2      NORMAL+     3    SET RJ TO C
         RPV      39                OUT-     2    CLEAR
         TP       C+      1           CR          IMAGE
         TP       C+      7          CR+    11    PLACE CONTROL BIT
         MJ        0                 WOR
OUT      EF                           T+     4    REWIND TAPE
         RP        3                OUT+     3
         EF                           R+     5    CLEAR PUNCH
         MJ                       START+     2    JUMP TO EXIT
CON               CR
                                      F3
                                   CARD
                                     CR+    26
                                      CR
                                     CR+    13
EST      LQ       W+    119            6
                                     SW+     1
                                     TWO
R               402B
         40                          10B          PP
         40                          12B          PP,P
                                     77B          MASK
         61   61616B              16161B          END MARK
         40        0                   0          CYCLE
          2   00200B                   0          DUMMY
C                                    120
                                                  ZERO
                                       1          COMPIL CODE SPACE
                   1
                   4
                   7
                   6
              40000B                   0          CONTROL PUNCH
C1                                   20B
                                     15B
                                       5
```

```
LOC         OP          U ADDR          V ADDR   REMARKS

                                   13B
                      15B
  CE                               63B           COMPILER CODE+
                                   22B           PERIOD
                                   43B           PAREN CLOSE
                                   21B           COMMA
                                   17B           PAREN OPEN
                                     1           6 SPECIAL    (PLUS)
                                   35B           CODES       (PERIOD)
                                   36B           DETERMINED  (PAR CL
                                   75B           BY LAYOUT OF (COMMA
                                                 INDIVIDUAL  (SPACE)
                                   76B           IBM 407     (PAR OP
   T RESERV            5             5
 IND RESERV            2             2           INDICES
   W RESERV          120           120           BLOCK SPACE
  CR RESERV           39            39           CARD IMAGE SPACE
          END
```

1. IDENTIFICATION

RRF010 - NATURAL LOG X FLOATING POINT
L. M. Johnson - 15 August 1957
Remington Rand Univac

2. PURPOSE

Given X, compute Y(X) = lnX in floating point, using stated point arithmetic.

3. METHOD

a. Accuracy: $|Y(X) - \ln(X)| \leq 2^{-27}$ in a 27 bit mantissa.

b. Range of Argument: $X>0$, $2^{-129} \leq X < 2^{127}$

c. Scaling: 1103AF floating point binary number representation.

d. Derivation: obtained from the relation $\ln X = (\ln 2)(\log_2 X)$; $\log_2 X$ is approximated by the expression $\frac{1}{2} + \sum_{i=1}^{4} C_{2i-1}\left(\frac{X - \sqrt{2}}{X + \sqrt{2}}\right)^{2i-1}$ where $1 \leq X \leq 2$.

(See Rand Sheet Number 42, and RRF011 - Natural Log X Stated Point.)

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t+2	t
r+1	Normal	Return	

b. Control and Results

The argument, X, in floating point form must be initially stored at t+4; the function Y(X) will be found in floating point form at t+3 upon successful completion of the routine. Y(X) is also left in A_R.

c. Space Required

78 cells of instructions and constants
3 cells of working storage

d. Error Codes

The following error code in flex is left in the Q-register upon return through the error exit

CODE	EXPLANATION
RRF010	$X \leq 0$

5. RESTRICTIONS

The argument must be within the stated range and must be in packed, normalized floating binary form.

6. CODING INFORMATION

a. Constants

LOC	CONSTANTS	EXPLANATION
C	00 00000 00000	Zero
C1	55 20236 31500	$\sqrt{2} \cdot 2^{35}$
C2	00 00000 00002	Two
C3	20 00000 00000	$1 \cdot 2^{34}$
C4	00 00001 00000	U-Address Advance
C5	15 71272 26456	$C_7 \cdot 2^{35}$
C6	22 34660 40144	$C_5 \cdot 2^{35}$
C7	36 61611 14432	$C_3 \cdot 2^{35}$
C8	34 25216 61765	$(C_1-2) \cdot 2^{35}$
C9	26 13441 37677	$Ln2 \cdot 2^{35}$
C10	00 00000 00200	Characteristic Bias
C11	00 00000 00046	38_{10}
C12	00 00000 00110	72_{10}
CODE	12 12263 75237	Error Code (Flex)

b. Working Space

3 cells labeled STORE thru STORE+2

c. Timing

Maximum of 5.70 mls.

PROBLEM RRF010

CODED BY L.JOHNSON DATE 8-15-57

```
ITEM NUMBER   LOC      OP       U          V          COMMENTS

    ,         , SUB    , RRF010   , 78       , LNX                                    $
    ,         , INOUT  , 1        , 1        , FLOATING POINT                         $
    ,         , TEMPS  , 3        , 0        ,                                        $
    , ENTRY   , MJ     , 0        , START    ,                                        $
    , ERROR   , ALARM  ,          ,          ,                                        $
    , EXIT    , MJ     , 0        , FILL     ,                                        $
    , Y       , 00     , FILL     , FILL     , FUNCTION FP                            $
    , X       , 00     , FILL     , FILL     , ARGUMENT FP                            $
    , START   , TP     , C        , A        , TEST FOR X                             $
    ,         , TJ     , X        , START+4, LESS-EQUAL ZERO                          $
    ,         , TP     , CODE     , Q        ,                                        $
    ,         , MJ     , 0        , ERROR    ,                                        $
    ,         , TP     , X        , A        , STORE                                  $
    ,         , LTL    , 9        , STORE    , CHARACTERISTIC                         $
    ,         , LTR    , 0        , Q        , AND MANTISSA (=M)                      $
    ,         , SP     , Q        , 0        , FORM AND STORE                         $
    ,         , SA     , C1       , 0        , (2M+SQUARE ROOT 2)                     $
    ,         , SA     , C2       , 0        , ROUNDED SCALED 33                      $
    ,         , LTL    , 34       , STORE+1, IN TEMP                                  $
    ,         , SP     , Q        , 0        , FORM (2M-SQUARE ROOT 2)                $
    ,         , SS     , C1       , 33       , SCALED 68 IN A                         $
    ,         , DV     , STORE+1  , STORE+2, FORM SQUARE OF Z=                        $
    ,         , MP     , Q        , Q        , ((2M-SQUARE ROOT 2)/(2M+SQUARE ROOT 2)) $
    ,         , SA     , C3       , 0        , ROUND SCALED                           $
    ,         , LTL    , 1        , STORE+1, 35 IN TEMP                               $
    ,         , TU     , SET      , NEST+3   ,                                        $
    ,         , TP     , C5       , Q        , C(7) TO Q                              $
    , NEST    , MP     , Q        , STORE+1, PARTIAL EVALUATION                       $
    ,         , SA     , C3       , 0        , OF POLYNOMIAL (=P)                     $
    ,         , LTL    , 1        , A        , EXPRESSION                             $
    ,         , AT     , FILL     , Q        , IN APPRX                               $
    ,         , RA     , NEST+3   , C4       , OF LNX                                 $
    ,         , TJ     , TEST     , NEST     , SCALED 35 IN Q                         $
    ,         , MP     , Q        , STORE+2, FORM ROUND STORE                         $
    ,         , RJ     , ROUND+4  , ROUND    , (P-2Z)                                 $
    ,         , LTL    , 1        , A        , SCALED 35 IN A                         $
    ,         , MP     , A        , C9       , FORM ROUND STORE                       $
    ,         , RJ     , ROUND+4  , ROUND    , (P-2Z) LN2 SCALD                       $
```

PROBLEM RRF010

CODED BY L. JOHNSON DATE 8-15-57

```
ITEM NUMBER   LOC      OP      U           V          COMMENTS
             ,        , LTL   , 1        , STORE+1, 35 IN TEMP              $
             ,        , LA    , STORE+2  , 1      , FORM (2Z-1/2) LN2       $
             ,        , ST    , C3       , STORE+2, SCLD 35                 $
             ,        , MP    , STORE+2  , C9     , IN TEMP                 $
             ,        , RJ    , ROUND+4  , ROUND  ,                         $
             ,        , LTL   , 1        , STORE+2,                         $
             ,        , RS    , STORE    , C10    , UNBIAS CHAR (=C)        $
             ,        , MP    , A        , C9     , FORM C(LN2) SCLD 35     $
             ,        , AT    , STORE+2  , A      , FORM (P+C-1/2) LN2      $
             ,        , AT    , STORE+1  , A      , SCLD 35 IN A            $
             ,        , TP    , C        , STORE  , FORM                    $
             ,        , SF    , A        , STORE  , NORMALIZED              $
             ,        , LTL   , 28       , Y      , MANTISSA                $
             ,        , LTL   , 0        , A      , IN Y AND A              $
             ,        , ZJ    , NTZER    , EXIT   ,                         $
             , NTZER  , SP    , STORE    , 0      ,                         $
             ,        , TJ    , C11      , YES    , TEST 38 GRTR K          $
             ,        , ST    , C12      , A      ,                         $
             , YES    , AT    , C10      , A      , ADD BIAS TO CHAR        $
             ,        , LTR   , 27       , Q      ,                         $
             ,        , CC    , Y        , Q      , PACK UP Y(X) FP         $
             ,        , MJ    , 0        , EXIT   , EXIT                    $
             , ROUND  , SJ    , ROUND+1  , ROUND+3,                         $
             ,        , ST    , C3       , A      ,                         $
             ,        , MJ    , 0        , ROUND+4,                         $
             ,        , AT    , C3       , A      ,                         $
             ,        , MJ    , 0        , FILL   ,                         $
             , SET    , 00    , C6       , FILL   ,                         $
             , TEST   , AT    , C6+3     , Q      ,                         $
             , C      , B     ,          ,       0, ZERO                    $
             , C1     , B55   , 20236    , 31500  , SQUARE ROOT 2 SCLD 35   $
             , C2     , B     ,          ,       2, ROUND BIT               $
             , C3     , B20   , 00000    , 00000  , 1 SCLD 34               $
             , C4     , B     ,        1 , 00000  , U ADD ADVANCE           $
             , C5     , B15   , 71272    , 26456  , C(7) SCALED 35          $
             , C6     , B22   , 34660    , 40144  , C(5) SCALED 35          $
```

PROBLEM RRF010

CODED BY L. JOHNSON DATE 8-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
	, C7	, B36	, 61611	, 14432	, C(3) SCALED 35	$
	, C8	, B34	, 25216	, 61765	, C(1)-2 SCALED 35	$
	, C9	, B26	, 13441	, 37677	, LN 2 SCALED 35	$
	, C10	, B	,	, 200	, DEC 128	$
	, C11	, B	,	, 46	, DEC 38	$
	, C12	, B	,	, 110	, DEC 72	$
	, CODE	, B12	, 12263	, 75237	, ERROR CODE	$
	,	, ENDSUB	,	,	,	$

A. Identification

1. Title: Single Precision Floating Point Conversion, WF0004.

2. Authors: W. Bauer and R. Graham
   Date: 1 July 1957

3. Installation: Wright Field

B. Purpose

This routine converts a single length floating point binary word from core or drum memory to its floating decimal equivalent and then to excess-three characters. WF0004 is to be used primarily as a subsidiary of the High Speed Printer Edit Routine, WF0001.

C. Usage

1. Calling Sequence

Loc	Op	U-addr	V-addr
r	RP3	2	r+2
r+1	TP	Parameter	C+3
r+2	RJ	C+2	C
r+3	Normal	Return	

2. Control Data

Two parameters are required for each word to be converted. They are as follows:

Loc	Op	U-addr	V-addr
Param	XX D	XXXXX M	XXX XX W S
Param+1	--	-----	XXXXX K

W (decimal) is the number of characters to be stored starting at address K. $(7 \leq W \leq 13)$

S See WF0001, para D2(b).

M is the address of the first word to be converted.

D See WF0001, para D2(b).

K is the first address of the cells in which the converted word is stored. This data is a series of six-bit excess-three characters stored one to a word in the rightmost six bits, the leftmost thirty bits being zero.

3. Space Required

200 memory cells

4. Error Codes

An alarm occurs if:

a. $W < 7$

b. $W > 13$

In either case the alarm is annotated as follows:

(A_L) = first parameter word
(A_R) = second parameter word
(Q) = WF0004 in flex code

5. Format Generated

W controls the number of output characters as follows:

W	Number of Characters
13	(-)X.XXXXXXX(-)XX
12	(-)X.XXXXXX(-)XX
.	.
.	.
.	.
7	(-)X.X(-)XX

D. Restrictions

1. The word to be converted must conform to the 1108A definition of a packed floating point number.

2. An unnormalized number is translated as a series of space codes.

3. Rounding to n digits is accomplished by adding 5 in the n+1 place.

E. Coding Information

1. Constants and their Locations

a. Alphanumeric

J through J3

b. Numeric

K1 through K26
1K through 1K115

2. Erasable Output Locations

P through P17

° MEANS $
MEANS +
/ MEANS -

```
,     ,SUB    ,WF0004,200      ,°
,     ,TEMPS  ,19    ,J27      ,°
,     ,INOUT  ,2     ,0        ,°
,B    ,MJO    ,0     ,C        ,ENTRY°
,B1   ,ALARM  ,      ,         ,EXIT ALARM°
,B2   ,MJO    ,0     ,FILL     ,EXIT NORMAL°
,B3   ,B      ,      ,0        ,°
,B4   ,B      ,      ,0        ,°
,C    ,TP     ,K14   ,Q        ,V MASK°
,     ,QT     ,B3    ,A        ,EXTRACT WS°
,     ,DV     ,K12   ,P        ,STORE W°
,     ,TP     ,K8    ,A        ,°
,     ,TJ     ,P     ,C27      ,IS W TOO LARGE°
,C1   ,TP     ,P     ,A        ,NO°
,     ,TJ     ,K6    ,C27      ,IS W TOO SMALL°
,C2   ,ST     ,K8    ,P#1      ,NO°
,     ,AT     ,J1    ,C23      ,SET EXPONENT SHIFT°
,     ,SP     ,P     ,15       ,°
,     ,AT     ,J2    ,C25      ,°
,     ,TV     ,J3    ,C15      ,SETUP FOR ROUNDIPO°
,     ,RA     ,C15   ,P#1      ,°
,     ,RP1    ,13    ,C3       ,°
,     ,TN     ,K2    ,P#5      ,°
,C3   ,TU     ,B3    ,C4       ,°
,C4   ,TP     ,FILL  ,A        ,OBTAIN WORD°
,     ,TP     ,A     ,P        ,AND STORE°
,     ,ZJ     ,C6    ,C5       ,IS WORD ZERO°
,C5   ,TP     ,A     ,P#6      ,YES STORE ZERO°
,     ,MJO    ,0     ,C21#4    ,°
,C6   ,TM     ,A     ,Q        ,MAG WORD TO Q°
,     ,QT     ,K17   ,P#1      ,WORD CHARACTERISTIC°
,     ,QT     ,K18   ,P#2      ,WORD MANTISSA°
,     ,LQ     ,P#1   ,9        ,SHIFT CHARACTERISTIC°
,     ,TJ     ,K15   ,C21#5    ,IS WORD NORMALIZED°
,C7   ,TP     ,P     ,A        ,WORD TO A°
,     ,SJ     ,C8    ,C9       ,IS WORD NEGATIVE°
,C8   ,TN     ,K1    ,P#5      ,YES SET SIGN°
,     ,TM     ,A     ,A        ,°
,C9   ,TP     ,K16   ,R        ,10 TO MINUS 38°
,     ,RP2    ,77    ,C10      ,°
,     ,TJ     ,1K    ,C11      ,°
,C10  ,SP     ,1K#77 ,45       ,HIGHEST POWER OF 10°
,     ,TP     ,A     ,P#3      ,STORE CHARACTERISTIC°
,     ,SS     ,A     ,27       ,°
,     ,TP     ,A     ,P#4      ,STORE MANTISSA°
,     ,TP     ,K11   ,P        ,10 TO 38°
,     ,MJO    ,0     ,C13      ,°
,C11  ,RS     ,Q     ,K13      ,J,N/R MINUS 20114°
,     ,RS     ,P     ,0        ,°
```

```
,     ,TP    ,J     ,A       ,ADD OF FIRST POWER OF 10°
,     ,SS    ,0     ,15      ,LOCATE°
,     ,TU    ,A     ,C12     ,APPROPRIATE POWER°
,C12,TP      ,FILL  ,0       ,OF TEN°
,     ,QT    ,K17   ,P#3     ,STORE CHARACTERISTIC°
,     ,QT    ,K18   ,P#4     ,STORE MANTISSA°
,     ,LQ    ,P#3   ,9       ,SHIFT CHARACTERISTIC°
,C13,RS      ,P#1   ,P#3     ,°
,     ,RA    ,P#1   ,K10     ,ADD 35°
,     ,TV    ,A     ,C14     ,SET SCALING°
,C14,SP      ,P#2   ,FILL    ,SHIFT WORD MANTISSA°
,     ,DV    ,P#4   ,P#2     ,DIVIDE BY POWER 10 MANTISSA°
,C15,RA      ,P#2   ,FILL    ,ROUND°
,     ,EJ    ,P#2   ,C17     ,IS THERE OVERFLOW°
,C16,RA      ,P     ,K1      ,YES INCREASE EXP°
,     ,TP    ,K19   ,P#2     ,1 SCALED 35°
,C17,TP      ,K6    ,P#1     ,SET SEVEN INDEX°
,     ,LQ    ,P#2   ,1       ,SHIFT FRACTION°
,     ,TV    ,J1    ,C19     ,SET STORAGE LOCATION°
,C18,SP      ,P#7   ,2       ,MULTIPLY°
,     ,SA    ,P#7   ,1       ,BY TEN°
,     ,TP    ,A     ,P#      ,°
,C19,LTO     ,0     ,FILL    ,°
,     ,RA    ,C19   ,K1      ,°
,     ,IJ    ,P#1   ,C18     ,8 DIGITS°
,     ,TP    ,P#7   ,P#6     ,°
,     ,RS    ,P     ,K1      ,EXP MINUS 1°
,     ,ZJ    ,C20   ,C21#4   ,IS EXP ZERO°
,C20,SJ      ,C21   ,C21#2   ,NO IS EXP NEGATIVE°
,C21,TN      ,K1    ,P#15    ,YES SET SIGN OF EXPONENT°
,     ,TM    ,A     ,A       ,°
,     ,DV    ,K7    ,P#16    ,FIRST DIGIT OF EXPONENT°
,     ,TP    ,A     ,P#17    ,SECOND DIGIT OF EXPONENT°
,     ,TP    ,K9    ,P#7     ,SET POINT°
,     ,RP2   ,13    ,C22     ,CONVERT°
,     ,RA    ,P#5   ,K3      ,TO XS/3°
,C22,RP3     ,9     ,C24     ,SHIFT°
,C23,TP      ,P#15  ,P#7     ,SIGN AND EXP°
,C24,TV      ,B4    ,C26     ,SET TRANSFER°
,     ,TP    ,C22   ,0       ,UNPACKED INDICATOR TO 0°
,C25,RP3     ,0     ,B2      ,TRANSFER°
,C26,TP      ,P#5   ,FILL    ,N DIGITS°
,C27,SP      ,B3    ,36      ,LOAD°
,     ,SA    ,B4    ,0       ,ARGUMENTS°
,     ,TP    ,K4    ,0       ,AND RTN CODE°
,     ,MJ0   ,0     ,B1      ,GO TO ALARM EXIT°
,J    ,      ,      ,1K      ,°
,J1   ,TP    ,P#15  ,P#7     ,°
,J2   ,RP3   ,0     ,B2      ,°
```

```
,J3  ,          ,        ,K18     ,0
,K1  ,B         ,        ,1       ,0
,K2  ,B         ,        ,2       ,0
,K3  ,B         ,        ,3       ,0
,K4  ,B31       ,26373   ,73764   ,WF00040
,K5  ,B         ,        ,5       ,0
,K6  ,B         ,        ,7       ,0
,K7  ,B         ,        ,12      ,0
,K8  ,B         ,        ,15      ,0
,K9  ,B         ,        ,17      ,0
,K10 ,B         ,        ,43      ,0
,K11 ,B         ,        ,47      ,0
,K12 ,B         ,        ,144     ,0
,K13 ,B         ,        ,20114   ,0
,K14 ,B         ,        ,77777   ,0
,K15 ,B         ,4000    ,00000   ,0
,K16 ,B77       ,77777   ,77731   ,0
,K17 ,B37       ,70000   ,00000   ,0
,K18 ,B40       ,07777   ,77777   ,0
,K19 ,B3        ,14631   ,46554   ,0
,K20 ,B         ,12172   ,70245   ,0
,K21 ,B         ,1014    ,22336   ,0
,K22 ,B         ,64      ,33343   ,0
,K23 ,B         ,5       ,17427   ,0
,K24 ,B         ,        ,41434   ,0
,K25 ,B         ,        ,3266    ,0
,K26 ,B         ,        ,254     ,0
,1K  ,B00       ,26634   ,37347   ,0
,    ,B00       ,64201   ,63521   ,0
,    ,B01       ,15242   ,20445   ,0
,    ,B01       ,46512   ,64556   ,0
,    ,B02       ,04116   ,60745   ,0
,    ,B02       ,35142   ,35136   ,0
,    ,B02       ,66373   ,04365   ,0
,    ,B03       ,24034   ,72631   ,0
,    ,B03       ,55044   ,11400   ,0
,    ,B04       ,06255   ,13700   ,0
,    ,B04       ,37730   ,36660   ,0
,    ,B04       ,74747   ,23216   ,0
,    ,B05       ,26141   ,10061   ,0
,    ,B05       ,57571   ,32075   ,0
,    ,B06       ,14653   ,70246   ,0
,    ,B06       ,46026   ,66320   ,0
,    ,B06       ,77434   ,44004   ,0
,    ,B07       ,34561   ,66403   ,0
,    ,B07       ,65716   ,24103   ,0
,    ,B10       ,17301   ,71124   ,0
,    ,B10       ,54471   ,13564   ,0
,    ,B11       ,05607   ,36522   ,0
```

,	,B11	,37151	,26246	,0
,	,B11	,74401	,65750	,0
,	,B12	,25502	,23342	,0
,	,B12	,57022	,70232	,0
,	,B13	,14813	,63140	,0
,	,B13	,45376	,57770	,0
,	,B13	,76676	,33766	,0
,	,B14	,34727	,01372	,0
,	,B14	,65274	,61671	,0
,	,B15	,16553	,76247	,0
,	,B15	,54143	,36750	,0
,	,B16	,05174	,26542	,0
,	,B16	,36433	,34273	,0
,	,B16	,74061	,11565	,0
,	,B17	,25075	,34122	,0
,	,B17	,56314	,63146	,0
,	,B20	,14000	,00000	,0
,	,B20	,45000	,00000	,0
,	,B20	,76200	,00000	,0
,	,B21	,27640	,00000	,0
,	,B21	,64704	,00000	,0
,	,B22	,16065	,00000	,0
,	,B22	,47502	,20000	,0
,	,B23	,04611	,32000	,0
,	,B23	,35753	,60400	,0
,	,B23	,67346	,54500	,0
,	,B24	,24520	,13710	,0
,	,B24	,55644	,16672	,0
,	,B25	,07215	,22450	,0
,	,B25	,44430	,28471	,0
,	,B25	,75536	,30410	,0
,	,B26	,27065	,76512	,0
,	,B26	,64341	,57116	,0
,	,B27	,15432	,12741	,0
,	,B27	,46740	,55532	,0
,	,B30	,04254	,34430	,0
,	,B30	,35827	,43536	,0
,	,B30	,66615	,34466	,0
,	,B31	,24170	,31702	,0
,	,B31	,55226	,40262	,0
,	,B32	,06474	,10336	,0
,	,B32	,44105	,45213	,0
,	,B32	,75126	,76456	,0
,	,B33	,26354	,56171	,0
,	,B33	,64023	,74714	,0
,	,B34	,15030	,74077	,0
,	,B34	,46237	,13116	,0
,	,B34	,77706	,75742	,0
,	,B35	,34734	,26555	,0

```
,    ,B35    ,66123 ,34311   ,°
,    ,B36    ,17550 ,23373   ,°
,    ,B36    ,54641 ,14185   ,°
,    ,B37    ,06011 ,37164   ,°
,    ,B37    ,37413 ,67021   ,°
,    ,B37    ,74547 ,32312   ,°
,    ,B40    ,25701 ,20775   ,°
,    ,ENDSUB,       ,        ,°
```

A. Identification

1. Title: Single Precision Stated Point Conversion, WF0003

2. Authors: W. Bauer and R. Graham
   Date: 1 July 1957

3. Installation: Wright Field

B. Purpose

This routine converts a single length stated point binary word from core or drum memory to its decimal equivalent and then to excess-three characters. WF0003 is to be used primarily as a subsidiary of the High Speed Printer Edit Routine, WF0001.

C. Usage

1. Calling Sequence

Loc	Op	U-addr	V-addr
r	RP3	2	r+2
r+1	TP	Parameter	C+3
r+2	RJ	C+2	C
r+3	Normal	Return	

2. Control Data

Two parameters are required for each word to be converted. They are as follows:

Loc	Op	U-addr	V-addr
Param	XX D	XXXXX M	XXX XX W S
Param+1	--	---XX J	XXXXX K

W (decimal) is the number of characters to be stored starting at address K. $(1 \leq W \leq 13)$

S See WF0001, para D2(b).

M is the address of the first word to be converted.

D See WF0001, para D2(b).

K is the first address of the cells in which the converted word is stored. This data is a series of six-bit, excess-three characters stored one to a word in the rightmost six bits, the leftmost thirty bits being zero.

J (decimal) is the binary scaling of the word to be converted. $(0 \leq J \leq 35)$.

3. Space Required

104 memory cells

4. Error Codes

The alarm conditions are as follows:

a. W > 13

b. J > 35

Both are annotated in the same manner, namely:

(A_L) = first parameter word
(A_R) = second parameter word
(Q) = WF0003 in flex code

D. <u>Restrictions</u>

Output will be truncated if W is less than 13; however, a word is never rounded.

E. <u>Coding Information</u>

1. Constants and their Locations

a. Alphanumeric

J through J3

b. Numeric

K through K21

2. Erasable Output Locations

P through P15

WF0003
Page 1 of 3 pages°

° MEANS $
MEANS +
/ MEANS -

```
,     ,SUB    ,WF0003 ,104    ,°
,     ,TEMP5  ,16     ,127    ,°
,     ,INOUT  ,2      ,0      ,°
,B    ,MJ0    ,0      ,C      ,ENTRY°
,B1   ,ALARM  ,      °,      °,EXIT ALARM°
,B2   ,MJ0    ,0      ,FILL   ,EXIT NORMAL°
,B3   ,B      ,       ,0      ,PARAMETER°
,B4   ,B      ,       ,0      ,CELLS°
,C    ,TP     ,K7     ,Q      ,MASK TO Q°
,     ,QT     ,B3     ,A      ,EXTRACT WS°
,     ,DV     ,K18    ,P#13   ,SAVE W°
,     ,TP     ,K8     ,A      ,°
,     ,TJ     ,P#13   ,C23    ,IS W TOO LARGE°
,C1   ,LA     ,P#13   ,15     ,NO°
,     ,AT     ,J      ,C21    ,SET TRANSFER°
,     ,TP     ,K      ,P#14   ,°
,     ,RPJ    ,13     ,C1X5   ,°
,     ,TN     ,K1     ,P      ,°
,     ,TV     ,B4     ,C22    ,°
,     ,LQ     ,B4     ,P#21   ,SHIFT SS°
,     ,QT     ,K7     ,P#13   ,SAVE SS°
,     ,TP     ,K8     ,A      ,°
,     ,TJ     ,P#13   ,C23    ,IS SS TOO LARGE°
,C2   ,ST     ,P#13   ,P#13   ,NO°
,     ,ZJ     ,C3     ,C4     ,IS SS 35°
,C3   ,MP     ,A      ,K6     ,NO°
,     ,SA     ,K8     ,0      ,°
,     ,TH     ,A      ,P#14   ,SAVE I DIGITS°
,C4   ,TP     ,K9     ,A      ,°
,     ,ST     ,P#14   ,P#15   ,SAVE F DIGITS°
,     ,AT     ,J2     ,C11    ,SET DIVIDE°
,     ,TP     ,J1     ,C10    ,PLACE RP°
,     ,RA     ,C10    ,P#14   ,SET N°
,     ,LQ     ,P#15   ,21     ,SHIFT F DIGITS°
,     ,TP     ,J3     ,A      ,°
,     ,SA     ,P#14   ,57     ,°
,     ,TV     ,A      ,C5     ,°
,C5   ,TP     ,K4     ,FILL   ,SET POINT°
,     ,TV     ,A      ,C19    ,SET F LOCATION°
,     ,RA     ,C19    ,K20    ,°
,     ,TV     ,P#13   ,C8     ,SET LA°
,     ,RA     ,C8     ,K20    ,SET K°
,     ,TU     ,B3     ,C6     ,°
,C6   ,TP     ,FILL   ,A      ,OBTAIN WORD°
,     ,SJ     ,C7     ,C8     ,IS WORD NEGATIVE°
,C7   ,TN     ,K20    ,P      ,YES SET SIGN°
,     ,TM     ,A      ,A      ,°
,C8   ,LA     ,A      ,FILL   ,SHIFT WORD°
,     ,TP     ,A      ,P#14   ,SAVE F PART°
```

WF0003


```
,      ,ZJ    ,C9      ,C15      ,IS WORD ZERO°
,C9    ,SS    ,A       ,36       ,NO°
,C10   ,RP3   ,C       ,C12      ,°
,C11   ,DV    ,K10     ,P#1      ,EXTRACT INTEGRAL DIGITS°
,C12   ,TH    ,J3      ,C13      ,°
,      ,TV    ,C11     ,C14      ,°
,C13   ,TP    ,FILL    ,A        ,SUPPRESS°
,      ,ZJ    ,C17     ,C14      ,HIGH°
,C14   ,TN    ,K1      ,FILL     ,ORDER°
,      ,RA    ,C13     ,K8       ,ZEROS°
,      ,RA    ,C14     ,K20      ,°
,      ,MJ0   ,0       ,C13      ,°
,C15   ,TV    ,C5      ,C16      ,ZERO WORD°
,      ,RS    ,C16     ,K20      ,°
,C16   ,TP    ,K       ,FILL     ,°
,C17   ,IJ    ,P#15    ,C18      ,°
,      ,MJ0   ,0       ,C20      ,°
,C18   ,SP    ,P#14    ,2        ,°
,      ,SA    ,P#14    ,1        ,EXTRACT°
,      ,TP    ,A       ,P#14     ,°
,C19   ,LTC   ,0       ,FILL     ,°
,      ,RA    ,C19     ,K20      ,°
,      ,MJ0   ,0       ,C17      ,°
,C20   ,TP    ,C11     ,0        ,UNPACKED INDICATOR TO 0°
,      ,RP2   ,13      ,C21      ,CONVERT TO°
,      ,RA    ,P       ,K2       ,XS/THREE°
,C21   ,RP    ,FILL    ,FILL     ,TRANSFER°
,C22   ,TP    ,P       ,FILL     ,N DIGITS°
,C23   ,TP    ,K21     ,0        ,SETUP°
,      ,SP    ,B3      ,36       ,FOR°
,      ,SA    ,B4      ,0        ,ALARM°
,      ,MJ0   ,0       ,B1       ,EXIT°
,J     ,RP3   ,0       ,B2       ,°
,J1    ,RP3   ,0       ,C12      ,°
,J2    ,DV    ,K10     ,P#1      ,°
,J3    ,      ,P#1     ,         ,°
,K     ,B     ,        ,0        ,°
,K1    ,B     ,        ,2        ,°
,K2    ,B     ,        ,3        ,°
,K3    ,B     ,        ,15       ,°
,K4    ,B     ,        ,17       ,°
,K5    ,B     ,        ,43       ,°
,K6    ,B     ,        ,23210    ,°
,K7    ,B     ,        ,77777    ,°
,K8    ,B     ,       1,00000    ,°
,K9    ,B     ,      13,00000    ,°
,K10   ,B11   ,24027   ,62000    ,°
,K11   ,B     ,73465   ,45000    ,°
,K12   ,B     ,5753    ,60400    ,°
```

```
,K13  ,B      ,461      ,13200   ,0
,K14  ,B      ,36       ,41100   ,0
,K15  ,B      ,3        ,03240   ,0
,K16  ,B      ,         ,23420   ,0
,K17  ,B      ,         ,1750    ,0
,K18  ,B      ,         ,144     ,0
,K19  ,B      ,         ,12      ,0
,K20  ,B      ,         ,1       ,0
,K21  ,B31    ,26373    ,73770   ,0
,     ,ENDSUB,          ,        ,0
```

A. Identification

1. Title: Octal Conversion, WF0002

2. Authors: W. Bauer and R. Graham
   Date: 1 July 1957

3. Installation: Wright Field

B. Purpose

This routine converts binary data from core or drum memory to its octal equivalent and then to excess-three characters. WF0002 is to be used primarily as a subsidiary of the High Speed Printer Edit Routine, WF0001.

C. Usage

1. Calling Sequence

Loc	Op	U-addr	V-addr
r	RP3	2	r+2
r+1	TP	Parameter	c+3
r+2	RJ	c+2	c
r+3	Normal	Return	

2. Control Data

Two parameters are required for each word to be converted. They are as follows:

Loc	Op	U-addr	V-addr
Param	XX D	XXXXX M	XXX XX W S
Param+1	--	----X B	XXXXX K

W (decimal) is the number of characters, including spaces, to be stored at address K. $(1 \leq W \leq 14)$.

S See WF0001, para D2(b)

M is the address of the first word to be converted.

K is the first address of the cells in which the converted word is stored. This data is a series of six-bit, excess-three characters stored one to a word in the rightmost six bits, the leftmost thirty bits being zero.

B is a format control word.
If B = 0, (M) is translated: XXXXXXXXXXXX__.
If B = 1, the address M is translated: XXXXX_________.
If B = 2, (M) is translated: XX_XXXXX_XXXXX.

3. Space Required

61 memory cells

4. Error Codes

When W is greater than 14, an alarm occurs.

(A_L) = first parameter word
(A_R) = second parameter word
(Q) = WF0002 in flex code

D. Restrictions

Output will be truncated if:

1. B = 0, and W is less than 12

2. B = 1, and W is less than 5

3. B = 2, and W is less than 14.

E. Coding Information

1. Constants and their Locations

   a. Alphanumeric

      K

   b. Numeric

      K1 through K12

2. Erasable Output Storage

   P through P14

° MEANS $
MEANS +
/ MEANS -

```
,    ,SUB   ,WF0002,61      ,°
,    ,TEMPS ,15    ,127     ,°
,    ,INOUT ,2     ,0       ,°
,B   ,MJO   ,0     ,C       ,ENTRY°
,B1  ,ALARM ,      ,        ,EXIT ALARM°
,B2  ,MJO   ,0     ,FILL    ,EXIT NORMAL°
,B3  ,B     ,      ,0       ,°
,B4  ,B     ,      ,0       ,°
,C   ,TP    ,K10   ,Q       ,°
,    ,QT    ,B3    ,A       ,EXTRACT WS°
,    ,DV    ,K9    ,A       ,W TO A°
,    ,TJ    ,K8    ,C2      ,IS W TOO LARGE°
,C1  ,SP    ,B3    ,36      ,YES°
,    ,SA    ,B4    ,C       ,LOAD PARAMETERS°
,    ,TP    ,K12   ,Q       ,AND CODE°
,    ,MJO   ,0     ,B1      ,GO TO ALARM EXIT°
,C2  ,LA    ,A     ,15      ,°
,    ,AT    ,K     ,C12     ,SET RP°
,    ,TV    ,B4    ,C13     ,°
,    ,RP1   ,14    ,C3      ,°
,    ,TN    ,K2    ,P#1     ,STORE /2°
,C3  ,TV    ,C3/1  ,C9      ,SET STORE DIGIT LOC°
,    ,LQ    ,B4    ,Q#21    ,°
,    ,QT    ,K3    ,A       ,EXTRACT CODE°
,    ,EJ    ,K1    ,C14     ,ADDRESS°
,C4  ,EJ    ,K2    ,C15     ,SPECIAL FORMAT°
,C5  ,TP    ,K11   ,P       ,SET 11 INDEX°
,C6  ,TU    ,B3    ,C7      ,°
,C7  ,SP    ,FILL  ,0       ,OBTAIN WORD°
,    ,TP    ,A     ,Q       ,WORD TO Q°
,C8  ,LQ    ,Q     ,3       ,°
,C9  ,QT    ,K6    ,FILL    ,STORE DIGIT°
,    ,RA    ,C9    ,K1      ,°
,    ,IJ    ,P     ,C8      ,°
,C10 ,RJ    ,C10   ,C11     ,°
,C11 ,TP    ,C9    ,Q       ,UNPACKED INDICATOR TO Q°
,    ,RP2   ,14    ,C12     ,°
,    ,RA    ,P#1   ,K3      ,CONVERT TO XS/3°
,C12 ,RP3   ,      ,B2      ,°
,C13 ,TP    ,P#1   ,FILL    ,TRANSFER # RESULTS°
,C14 ,TP    ,K4    ,P       ,SET 4 INDEX°
,    ,LQ    ,B3    ,Q#6     ,°
,    ,MJO   ,0     ,C8      ,°
,C15 ,TP    ,K1    ,P       ,SET 1 INDEX°
,    ,RJ    ,C10   ,C6      ,°
,    ,RA    ,C9    ,K1      ,SKIP FOR SPACE°
,    ,TP    ,K4    ,P       ,SET 4 INDEX°
,    ,RJ    ,C10   ,C8      ,°
,    ,RA    ,C9    ,K1      ,SKIP FOR SPACE°
```

```
,    ,TP   ,K4    ,P     ,SET 4 INDEX°
,    ,MJ0  ,0     ,C8    ,°
,K   ,RP3  ,0     ,R2    ,°
,K1  ,B    ,      ,1     ,°
,K2  ,B    ,      ,2     ,°
,K3  ,B    ,      ,3     ,°
,K4  ,B    ,      ,4     ,°
,K5  ,B    ,      ,5     ,°
,K6  ,B    ,      ,7     ,°
,K7  ,B    ,      ,16    ,°
,K8  ,B    ,      ,17    ,°
,K9  ,B    ,      ,144   ,°
,K10 ,B    ,      ,77777 ,°
,K11 ,B    ,      ,13    ,°
,K12 ,R01  ,26373 ,73774 ,°
,    ,ENDSUB,     ,      ,°
```

16

A. Identification

1. Title: Alphanumeric Conversion, WF0005

2. Authors: W. Bauer and R. Graham
   Date: 1 July 1957

3. Installation: Wright Field

B. Purpose

This routine assembles prestored, prepacked excess-three characters from core or drum memory for editing. WF0005 is to be used primarily as a subsidiary of the High Speed Printer Edit Routine, WF0001.

C. Usage

1. Calling Sequence

Loc	Op	U-addr	V-addr
r	RP3	2	r+2
r+1	TP	Parameter	c+8
r+2	RJ	c+2	c
r+3	Normal	Return	

2. Control Data

Two parameters are required for each group of characters to be assembled. They are as follows:

Loc	Op	U-addr	V-addr
Param	XX D	XXXXX M	XXX XX W S
Param+1	--	-----	XXXXX K

W (decimal) is the number of characters, including spaces to be stored starting at address K. $(1 \le W \le 120)$.

S See WF0001, para D2(b).

M is the address of the first six characters to be assembled.

D See WF0001, para D2(b).

K is the first address of the cells in which the assembled data is stored. This data is a series of six-bit, excess-three characters packed six to a word.

3. Space Required

24 memory cells

D. Coding Information

1. Constants and their Locations

   K through K4

2. Erasable Output Locations

   P

° MEANS $

```
,      ,SUB    ,WF0005 ,24      ,°
,      ,TEMPS  ,1      ,127     ,°
,      ,INOUT  ,2      ,0       ,°
,B     ,MJ0    ,0      ,C       ,ENTRY°
,B1    ,ALARM  ,      0,       0,EXIT ALARM°
,B2    ,MJ0    ,0      ,FILL    ,EXIT NORMAL°
,B3    ,B      ,       ,0       ,PARAMETER°
,B4    ,B      ,       ,0       ,CELLS°
,C     ,TU     ,K4     ,C4      ,PRESETS°
,      ,TU     ,B3     ,C5      ,°
,      ,TP     ,K3     ,Q       ,°
,      ,QT     ,B3     ,A       ,°
,      ,DV     ,K2     ,A       ,OBTAIN W IN A°
,      ,DV     ,K1     ,P       ,DIVIDE BY 6°
,C1    ,ZJ     ,C2     ,C3      ,IS THERE A REMAINDER°
,C2    ,RA     ,P      ,K       ,YES INCREASE QUOTIENT°
,C3    ,LA     ,P      ,15      ,NO°
,      ,AT     ,C4     ,C4      ,FORM N OF REPEAT°
,      ,TV     ,B4     ,C5      ,°
,      ,TP     ,C      ,Q       ,PACKED INDICATOR TO Q°
,C4    ,RP3    ,       ,B2      ,°
,C5    ,TP     ,FILL   ,FILL    ,STORE DATA IN EDIT ROUTINE°
,K     ,B      ,       ,1       ,°
,K1    ,B      ,       ,6       ,°
,K2    ,B      ,       ,144     ,°
,K3    ,B      ,       ,77777   ,°
,K4    ,B      ,30000  ,00000   ,°
,      ,ENDSUB ,       ,        ,°
```

1. IDENTIFICATION

RRF008 ARCCOS X FLOATING POINT
L. Krak, R. Van Hilst 15 July 1957
Remington Rand Univac

2. PURPOSE

Given X compute Y(X) = arccosx in radians

3. METHOD

a. Accuracy: $|Y(X) - \text{arccos}x| \leq 2^{-26}$ in a normalized 27 bit mantissa.

b. Range of Argument: $|X| \leq 1$; for $0 \leq |X| < 2^{-32}$, Y(X) is set equal to $\frac{\pi}{2}$.

c. Scaling: 1103AF packed, 8-27, floating point number representation.

d. Derivation: X is converted to stated point scaled 2^{33}; the function, arcsin X is computed; then arccos x = $\frac{\pi}{2}$ - arcsin x (See RRF6, Arccos x Stated Point), and this stated point value is converted to floating point representation for Y(X).

4. USAGE

a. Calling Sequence

LOC	OP	U ADDR	V ADDR
r	RJ	t+2	t
r+1	Normal	Return	

b. Control and Results

The floating point representation of the argument, X, must be initially stored at t+4, the floating point representation of the function, Y(X), will be found at t+3 upon successful completion of the routine.

c. Space required

122 cells of instructions and constants.

7 cells of working space.

d. Error Codes

The following error code is left in the Q-register on return through the error exit:

CODE	EXPLANATION
RRF008	$\lvert X \rvert > 1$

5. RESTRICTIONS

The argument must be in 1103AF normalized, packed floating point number representation and must be within the stated range.

6. CODING INFORMATION

a. Constants

LOC	CONSTANT	EXPLANATION
C	20 14000 00001	FP $(1+1 \cdot 2^{-27})$
C1	00 00000 00000	Zero
C2	14 14000 00000	FP 2^{-32}
C3	00 00000 00001	One
C4	20 14000 00000	FP 1
C5	37 70000 00000	Characteristic Mask
C6	00 00000 00126	Characteristic Unbias Constant
C7	53 24135 20070	$a_7 \cdot 2^{44}$
C8	33 24414 25535	$a_6 \cdot 2^{42}$
C9	56 40071 51545	$a_5 \cdot 2^{40}$
C10	37 50417 41234	$a_4 \cdot 2^{40}$
C11	46 23706 66522	$a_3 \cdot 2^{39}$
C12	26 61651 66073	$a_2 \cdot 2^{38}$
C13	44 42003 30653	$a_1 \cdot 2^{37}$
C14	31 10375 51633	$a_0 \cdot 2^{34}$
C15	10 00000 00000	$1 \cdot 2^{33}$
C16	26 47670 31361	Constants
C17	00 00000 65324	For
C18	11 45346 44516	Square Root
C19	33 06571 40273	Routine
C20	00 00000 00077	Mask
C21	26 50117 14640	$\sqrt{2} \cdot 2^{34}$
C22	20 00000 00000	$1 \cdot 2^{34}$
C23	37 77777 77777	Round Factor
C24	14 44176 65211	$\frac{\pi}{2} \cdot 2^{33}$
C25	07 20000 00000	Characteristic Bias I
C26	20 16220 77325	FP $\frac{\pi}{2}$

(cont)

LOC	CONSTANT	EXPLANATION
C27	11 00000 00000	Characteristic Bias II
CODE	12 12263 73760	Error Code

b. Work Space

7 cells labeled STORE thru STORE+6

c. Timing

Approximate maximum of 7.22 mls. for $X= -(1-1.2^{-27})$

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, SUB	, ERF008	, 122	,	$
,		, TEMPS	, 7	, 0	,	$
,		, INOUT	, 1	, 1	,	$
,	ENTRY	, MJ	, 0	, START	,	$
,	ERROR	, ALARM	,	,	,	$
,	EXIT	, MJ	, 0	, FILL	,	$
,	Y	, 00	, FILL	, FILL	, FUNCTION	$
,	X	, 00	, FILL	, FILL	, ARGUMENT	$
,	START	, TM	, X	, A	, ABS X TO A	$
,		, TJ	, C	, PROG	, CHK FOR X GRTR 1	$
,		, SP	, CODE	, Q	, GO TO	$
,		, MJ	, 0	, ERROR	, ERROR EXIT	$
,	PROG	, TP	, C26	, Y	, FOR SMALL X AND	$
,		, TJ	, C2	, EXIT	, X = 0 Y = PI OVR 2	$
,		, TP	, A	, STORE	, STORE ABS X	$
,		, TP	, X	, A	,	$
,		, TP	, C1	, STORE+1	, SET FOR POS X	$
,		, SJ	, MINUS	, PLUS	, TEST SIGN OF X	$
,	MINUS	, TP	, C3	, STORE+1	, SET FOR NEG X	$
,		, TM	, A	, A	, ABS X TO A	$
,	PLUS	, EJ	, C4	, XONE	, TEST X=1	$
,		, TP	, C5	, Q	, MASK TO Q	$
,		, QT	, A	, Q	, CHAR TO Q	$
,		, CC	, STORE	, Q	, STORE MANTISSA	$
,		, LQ	, Q	, 9	, FORM SHIFT COUNT	$
,		, RS	, Q	, C6	, IN Q	$

PROBLEM RRFC08

CODED BY KRAK, VAN HILST DATE 7-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, TV	, Q	, SHIFT	, K = CHAR -86	$
,	SHIFT	, SP	, STORE	, FILL	, SHIFT MANT K LEFT	$
,		, LTL	, 0	, STORE	, ABS X SCLD 33 TO STORE	$
,		, MP	, STORE	, C7	,	$
,		, LTL	, 1	, A	, EVALUATE	$
,		, AT	, C8	, STORE+2	, POLYNOMIAL	$
,		, MP	, STORE	, STORE+2	, EXPRESSION	$
,		, LTL	, 1	, A	, IN	$
,		, AT	, C9	, STORE+2	, APPROXIMATION	$
,		, MP	, STORE	, STORE+2	, OF ARCSINX	$
,		, LTL	, 3	, A	, ACCUMULATE	$
,		, AT	, C10	, STORE+2	, IN	$
,		, MP	, STORE	, STORE+2	, STORE+2	$
,		, LTL	, 2	, A	, SCLD 34	$
,		, AT	, C11	, STORE+2	,	$
,		, MP	, STORE	, STORE+2	,	$
,		, LTL	, 2	, A	,	$
,		, AT	, C12	, STORE+2	,	$
,		, MP	, STORE	, STORE+2	,	$
,		, LTL	, 2	, A	,	$
,		, AT	, C13	, STORE+2	,	$
,		, MP	, STORE	, STORE+2	,	$
,		, LTL	, 0	, A	,	$
,		, AT	, C14	, STORE+2	, X(X) TO STORE+2	$
,		, TN	, STORE	, A	, FORM (1-X)	$
,		, SA	, C15	, 0	, SCLD 33 IN A	$
,		, SF	, A	, STORE+3	, K TO STORE+3	$

PROBLEM RRF008
CODED BY KRAK, VAN HILST DATE 7-15-57

```
ITEM NUMBER     LOC       OP      U           V           COMMENTS

           ,          ,  TP   ,  A       ,  STORE+4   ,  STORE N                       $
           ,          ,  SA   ,  C16     ,  18        ,  FORM AND STORE                $
           ,          ,  LTL  ,  0       ,  STORE+5   ,  (N+D)SCLD -18=F               $
           ,          ,  MP   ,  C17     ,  STORE+5   ,  AF TO STORE                   $
           ,          ,  AT   ,  C18     ,  STORE+6   ,  AF+B                          $
           ,          ,  SN   ,  C19     ,  15        ,  -C SCLD 15 TO A               $
           ,          ,  DV   ,  STORE+5,,  A         ,  -C OVR F=-G                   $
           ,          ,  AT   ,  STORE+6,   STORE+5   ,  Y(1) APPRX SQRT N SCLD 16$
           ,          ,  SP   ,  STORE+4,   32        ,  N SCLD 32 TO A                $
           ,          ,  SS   ,  STORE+5,,  0         ,  N - Y(1)                      $
           ,          ,  DV   ,  STORE+5,   A         ,  (N OVR Y(1)) -1+Y(1)          $
           ,          ,  AT   ,  STORE+5,   STORE+5   ,  =Y(2) = SQRT N SCLD 17        $
           ,          ,  LQ   ,  STORE+3,   35        ,  K(0) IN Q (35)                $
           ,          ,  QT   ,  C20     ,  A         ,  K-K(0) OVR 2 IN A             $
           ,          ,  TV   ,  A       ,  KODD      ,  SET SCALE FOR SQRT N          $
           ,          ,  QJ   ,  KODD    ,  KEVEN     ,  TEST PARITY OF K              $
           ,  KEVEN   ,  MP   ,  C21     ,  STORE+5   ,                                $
           ,          ,  SA   ,  C22     ,  1         ,                                $
           ,          ,  LTL  ,  0       ,  STORE+5   ,                                $
           ,  KODD    ,  SP   ,  STORE+5,   FILL      ,                                $
           ,          ,  SA   ,  C23     ,  0         ,  RND SCL SQRT (1-X)            $
           ,          ,  LTL  ,  0       ,  A         ,  SCLD 33 IN A                  $
           ,          ,  MP   ,  A       ,  STORE+2,  ,  -ARCSIN X                     $
           ,          ,  LTL  ,  2       ,  A         ,  SCLD 33                       $
           ,          ,  ST   ,  C24     ,  Q         ,  IN Q                          $
           ,  TEST    ,  IJ   ,  STORE+1,   NEGX      ,  FOR X NEG                     $
           ,          ,  TN   ,  Q       ,  Q         ,  Y IN 2ND QUAD                 $
           ,  NEGX    ,  RS   ,  Q       ,  C24       ,  -ARCCOSX                      $
           ,          ,  TN   ,  A       ,  A         ,  =ARCSINX - PI OVR 2           $
```

PROBLEM RRF008

CODED BY KRAK, VAN HILST DATE 7-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,		, TP	, C1	, STORE+2	, CLEAR STORE+2	$
,		, SF	, A	, STORE+2	,	$
,		, ZJ	, NOZERO	, ZERO	, TEST FOR MANT=0	$
,	ZERO	, TP	, C1	, Y	,	$
,		, MJ	, 0	, EXIT	,	$
,	NOZERO	, LTL	, 28	, Y	, STORE MANTISSA	$
,		, SP	, STORE+2	, 27	, K SCLD 27 TO A	$
,		, ZJ	,KZERO+1	, KZERO	, TEST K=0	$
,	KZERO	, TP	, C27	, A	, SET A=72 SCLD 27	$
,		, AT	, C25	, STORE+2	, FORM CHARACTERISTIC	$
,		, CC	, Y	, STORE+2	, PACK UP RESULT	$
,		, MJ	, 0	, EXIT	,	$
,	XONE	, TN	, C24	, Q	, FOR X=1	$
,		, MJ	, 0	, TEST	,	$
,	C	, B201	, 4000	, 00001	, UPPER LIMIT ON X	$
,	C1	, B	,	, 0	, ZERO	$
,	C2	, B141	, 4000	, 00000	, LOWER LIMIT ON X	$
,	C3	, B	,	, 1	, ONE	$
,	C4	, B201	, 4000	, 00000	, FP 1	$
,	C5	, B377	, 0000	, 00000	, CHARACTERISTIC MASK	$
,	C6	, B	,	, 126	, CHAR UNBIAS CONST	$
,	C7	, B53	, 24135	, 20070	, A(7) SCLD 44	$
,	C8	, B33	, 24414	, 25535	, A(6) SCLD 42	$
,	C9	, B56	, 40071	, 51545	, A(5) SCLD 40	$
,	C10	, B37	, 50417	, 41234	, A(4) SCLD 40	$
,	C11	, B46	, 23706	, 66522	, A(3) SCLD 39	$
,	C12	, B26	, 61651	, 66073	, A(2) SCLD 38	$

REMINGTON RAND UNIVAC

PROBLEM RRF008

CODED BY KRAK, VAN HILST DATE 7-15-57

ITEM NUMBER	LOC	OP	U	V	COMMENTS	
,	C13	, B44	, 42003	, 30653	, A(1) SCLD 37	$
,	C14	, B31	, 10375	, 51633	, A(0) SCLD 34	$
,	C15	, B10	, 00000	, 00000	, 1 SCLD 33	$
,	C16	, B26	, 47670	, 31361	, CONSTANTS	$
,	C17	, B	,	, 65324	, FOR	$
,	C18	, B11	, 45346	, 44516	, SQRT	$
,	C19	, B33	, 06571	, 40273	, ROUTINE	$
,	C20	, B	,	, 77	, K MASK	$
,	C21	, B26	, 50117	, 14640	, SQRT 2 SCLD 34	$
,	C22	, B20	, 00000	, 00000	, 1 SCLD 34	$
,	C23	, B37	, 77777	, 77777	, ROUND FACTOR	$
,	C24	, B14	, 44176	, 65211	, PI OVR 2 SCLD 33	$
,	C25	, B072	, 0000	, 00000	, CHARACTER BIAS	$
,	C26	, B201	, 6220	, 77325	, FP PI OVR 2	$
,	C27	, B110	, 0000	, 00000	,	$
,	CODE	, B12	, 12263	, 73760	, ERROR CODE	$
,		, ENDSUB	,	,	,	$

Talmadge

USE Letter No: PC - 9

13 January, 1958

To: USE Policy Committee, Publications Committee, Installation Heads

Attached herewith is USEful Note #17, Sort Routine. We are circulating this program as a USEful Note inasmuch as it has not been machine checked. The routine has performed successfully on Remington Rand's 1103 at St. Paul but time on an 1103A has not as yet been available for final check-out. Your experience with this routine would be appreciated.

The attached material is excluded from the normal page number serializing.

Dirk de Vries

Dirk de Vries
Executive Secretary, USE
Remington Rand Univac,
Univac Park,
St. Paul 16,
Minnesota.

962

DdV:diw

Enc.

13 January 1958

USEful Note #17

SUBJECT: Sort Routine

CONTRIBUTOR: Remington Rand

1. IDENTIFICATION

RR UN 17 - Sort Routine
V. Benda, R. Russell - 19 November 1957
Remington Rand Univac

2. PURPOSE

To sort a possible 4094_{10} consecutive computer words, or portions thereof, in ascending or descending order and with a possible 10_{10} groups of related consecutively stored words to be arranged in order in respect to the sorted data.

3. METHOD

A group of cells are sorted in either ascending or descending order by transmitting the data involved into magnetic core negative or positive respectively and use of a series of repeated threshold jumps. Groups of N related cells may then be arranged in the same respective order as the sorted group by placing the data into the same magnetic core locations as the original data and transferring back to magnetic drum in the same order as the original sorted data.

Four control words supplied to the routine designate the order of sorting, magnetic drum locations of data to be sorted, magnetic core locations to be used as temporary storage during sorting, magnetic drum locations for data after a completed sort, magnetic drum temporary storage for relative locations of sorted data in reference to magnetic core, and the extractor, if desired, designating the portion of the computer word to be sorted.

Ten other possible control words may be supplied designating the magnetic drum locations of data to be arranged in respect to the original sorted data and the final magnetic drum locations for storage upon completion of arrangement.

4. USAGE

a. Calling sequence - standard

b. Control Data

The following octal control words must be stored at t 3 through t 16 prior to entering the routine:

LOC	OP	U ADDR	V ADDR	EXPLANATION
t 3	UU	XXXXX	YYYYY	UU_8 - Code for order of sort, 13-Ascending, 11-Descending; $XXXXX_8$ - First MD address of data to be sorted; $YYYYY_8$ - Last MD address of data to be sorted.
t 4	00	MMMMM	ZZZZZ	$MMMMM_8$ - First MC address for temporary storage of data while being sorted, must equal number of data plus one. $ZZZZZ_8$ - First MD address for final storage of data after sort.

t 5	00	00000	RRRRR	$RRRRR_8$ - First MD address for temporary storage of relative locations of data while being sorted, must equal number of data plus one.
t 6	WW	WWWWW	WWWWW	WW WWWWW $WWWWW_8$ - Extractor * to designate which bits of word to be sorted. If no extractor is desired set equal to zero.
t 7	SS	VVVVV	NNNNN	SS_8 - Code for last parameter, 77 - last parameter, 00 - otherwise; $VVVVV_8$ - First MD address of data to be arranged in respect to original sorted data. $NNNNN_8$ - First MD address of final storage for data after arrangement in respect to original sorted data.

*Upon completion of the sort with the use of an extractor the original data will be stored, sorted in respect to the extracted bits.

c. Space required

165 cells of instructions and constants
266 cells of working storage
2(n 1) cells of designated working storage. N equals number of cells of data to be sorted.

d. Error codes

"ERROR" is printed.

5. RESTRICTIONS

a. This routine will operate on any 1103A as established by USE but size of MC limits the number of data to be sorted.

b. This routine is self contained.

c. This routine requires that the temporary storage beginning at $MMMMM_8$ be in MC, thereby limiting the number of pieces of data to be sorted to the (number of locations in MC) minus 657_8. If so desired, this temporary storage could be located in the MD but the timing will be increased by a multiple of approximately 500.

d. An extractor (t 6) must not be used when it is known that the data will contain negative numbers.

e. The octal numbers 40 00000 00000 and 37 77777 77777 must not be contained in the data to be sorted as these numbers are used by the routine during the sorting procedure.

f. No internal check is made by the routine for over-lapping of regions.

6. CODING INFORMATION

a. Working storage
266 cells labeled TEMPS through TEMPS 265

b. Constants
20 cells labeled CON through CON 19

ITEM NO.	TAG	OPERATION	U ADDRESS	V ADDRESS	COMMENTS	
	,	, SUB	, SORT	, 165	,	$
	,	, TEMPS	, 266	, 0	,	$
	,	, INOUT	, 14	, 0	,	$
	, ENTRY	, MJ	, 0	, START	,	$
	, ERROR	, ALARM	,	,	,	$
	, EXIT	, MJ	, 0	, FILL	,	$
	, PAR	, RESERV	, 14	, 14	,	$
	, START	, PR3	, 3	, START+2	, PAR.TO	$
	, 1	, TP	, PAR	, TEMPS	, TEMP	$
	, 2	, PR2	, 3	, START+4	,	$
	, 3	, LQ	, TEMPS	, 17	, SHIFT TO U	$
	, 4	, TP	, PAR	, Q	,	$
	, 5	, TP	, PAR+1	, A	,	$
	, 6	, LQ	, Q	, 25	, SHIFT TO V	$
	, 7	, LT	, 25	, A	,	$
	, 8	, TV	, Q	, TEMPS	,	$
	, 9	, TV	, A	, TEMPS+1	,	$
	, ST	, TU	, PAR	, DA 1	,	$
	, 1	, TP	, CON	, Q	, EXT. TO Q	$
	, 2	, QT	, PAR	, TEMPS+4	,	$
	, 3	, QT	, TEMPS	, Q	,	$
	, 4	, RS	, TEMPS+4	, Q	,	$
	, 5	, AT	, CON+4	, TEMPS+4	,	$
	, 6	, SA	, CON+5	, 17	, SET UP RP	$
	, 7	, TU	, A	, DA	,	$
	, 8	, TV	, TEMPS+1	, DA+1	,	$
	, DA	, RP	, FILL	, DA+2	,	$
	, 1	, TP	, FILL	, FILL	, DATA TO MC	$
	,2	, SP	, DA 1	, 0	,	$
	, 3	, SA	, TEMPS+4	, 0	,	$
	, 4	, TV	, A	, DA+5	,	$
	, 5	, TP	, CON+7	, FILL	, STORE SEN.	$
	, K	, TP	, TEMPS+4	, A	, NO OF DATA TO A	$
	, 1	, LT	, 35	, TEMPS+6	, A/128 TO K	$
	, 2	, LQ	, A	, 7	,	$
	, 3	, TP	, Q	, TEMPS+7	, REM TO K 1	$
	, EX	, TP	, PAR+3	, A	, EXT. TO A	$
	, 1	, ZJ	, EX+2	, WS	,	$
	, 2	, TP	, A	, Q	,	$
	, 3	, TU	, DA	, EX+6	, SET RP	$
	, 4	, TU	, PAR+1	, EX+7	,	$
	, 5	, TV	, TEMPS+1	, EX+7	,	$
	, 6	, RP	, FILL	, WS	, MASK OUT	$
	, 7	, QT	, FILL	, FILL	, WITH EXT.	$
	, WS	, TP	, PAR	, A	,	$
	, 1	, TJ	, CON+8	, WS+8	,	$
	, 2	, TP	, A	, WS+7	,	$
	, 3	, TU	, DA	, WS+6	,	$
	, 4	, TU	, PAR+1	, WS+7	,	$
	, 5	, TV	, DA+1	, WS+7	,	$
	, 6	, RP	, FILL	, WS+8	,	$
	, 7	, 0	, FILL	, FILL	,	$
	, 8	, TP	, CON+9	, A	,	$
	, MA	, RP1	, 128	, MA+2	,	$

c. Timing

Depending on number of pieces of data and arrangement the timing will vary considerably. For 7776_8 pieces of data it will be approximately 2 minutes.

ITEM NO.	TAG	OPERATION	U ADDRESS	V ADDRESS	COMMENTS	
	, LB	, TV	, MA+33	, CON+16	,	$
	, 1	, SP	, CON+16	, 0	,	$
	, 2	, AT	, TEMPS+7	, LB+3	, SET NI	$
	, 3	, 0	, FILL	, FILL	, STORE JUMP OUT	$
	, 4	, SP	, TEMPS+7	, 0	,	$
	, 5	, SA	, CON+5	, 17	, SET RP	$
	, 6	, TU	, A	, MS+32	,	$
	, 7	, TU	, A	, TEMPS+8	,	$
	, 8	, RS	, TEMPS+3	, A	, CLEAR INDEX A	$
	, 9	, MJ	, 0	, MA+32	,	$
	, RL	, RP	, FILL	, RL+2	, BRING IN	$
	, 1	, TP	, FILL	, FILL	, DATA	$
	, 2	, TP	, CON+17	, TEMPS+137	,	$
	, 3	, RP3	, 128	, TEMPS+9	, BRING IN 128	$
	, 4	, TP	, FILL	, TEMPS+9	, REL. LOC	$
	, 5	, RA	, RL+4	, CON+19	,	$
	, 6	, RP3	, 128	, RL+8	, STORE TRUE	$
	, 7	, TP	, TEMPS+138	, FILL	, VALUES ON MD	$
	, 8	, RA	, RL+7	, CON-14	,	$
	, 9	, RJ	, RL+9	, RL+10	,	$
	, 10	, MJ	, 0	, RL+3	,	$
	, 11	, TU	, TEMPS+8	, RL+6	,	$
	, 12	, RJ	, RL+9	, RL+6	,	$
	, 13	, TU	, RL+3	, RL+6	,	$
	, 14	, MJ	, 0	, FILL	, LOOP EXIT	$
	, CON	, B	, 0	, 77777	,	$
	, 1	, B	, 0	, 2	,	$
	, 2	, B20	, 12120	, 31245	,	$
	, 3	, B	, 1	, 0	,	$
	, 4	, B	, 0	, 1	,	$
	, 5	, B	, 0	, 30000	,	$
	, 6	, B40	, 0	, 0	,	$
	, 7	, B37	, 77777	, 77777	,	$
	, 8	, TM	, PAR+4	, 0	,	$
	, 9	, TP	, FILL	, TEMPS+137	,	$
	, 10	, TP	, FILL	, A	,	$
	, 11	, TJ	, FILL	, MA+14	,	$
	, 12	, B	, 0	, 177	,	$
	, 13	, B	, 0	, 27777	,	$
	, 14	, B	, 0	, 200	,	$
	, 15	, B	, 0	, 11	,	$
	, 16	, TP	, CON+18	, FILL	,	$
	, 17	, MJ	, 0	, RL+5	,	$
	, 18	, MJ	, 0	, RL+11	,	$
	, 19	, B	, 200	, 0	,	$
	, ER	, TP	, CON+2	, Q	,	$
	, 1	, MJ	, 0	, ERROR	, TO ALARM EXIT	$

ITEM NO.	TAG	OPERATION	U ADDRESS	V ADDRESS	COMMENTS
	, 1	, AT	, CON+4	, TEMPS+11,	, $
	, 2	, TU	, PAR+1	, CON+10	, $
	,3	, TU	, PAR+1	, CON+11	, $
	, 4	, RA	, CON+11	, CON+3	, $
	, 5	, TV	, PAR+2	, MA+33	, $
	, 6	, TP	, TEMPS+6	, TEMPS+3	, LOAD INDEX A $
	, 7	, TP	, CON+12	, TEMPS+5	, 127 TO INDEX B $
	, 8	, TV	, MA+1	, MA+27	, $
	, 9	, TP	, CON+10	, MA+11	, $
	, 10	, TP	, CON+11	, MA+13	, $
	, 11	, TP	, FILL	, A	, 1ST VALUE TO A $
	, 12	, RP2	, 4095	, ER	, TEST FOR $
	, 13	, TJ	, FILL	, MA+14	, LARGEST VALUE $
	, 14	, TP	, CON+13	, A	, $
	, 15	, SS	, Q	, 17	, $
	, 16	, AT	, MA+11	, MA+17	, $
	, 17	, 0	, FILL	, FILL	, $
	, 18	, EJ	, CON+7	, MA+23	, E UAL SENTINEL $
	, 19	, TP	, MA+17	, MA+11	, $
	, 20	, TU	, MA+17	, MA+13	, $
	, 21	, RA	, MA+13	, CON+3	, SET TJ $
	, 22	, MJ	, 0	, MA+11	, $
	, 23	, TP	, MA+11	, MA+24	, $
	, 24	, 0	, FILL	, FILL	, $
	, 25	, EJ	, CON+6	, LB	, EQUAL FILLER $
	, 26	, LQ	, MA+11	, 25	, SHIFT TO V $
	, 27	, TU	, MA+24	, FILL	, STORE REL. LOC $
	, 28	, TV	, MA+11	, MA+29	, $
	, 29	, TP	, CON+6	, FILL	, SET EQUAL TO FILLER $
	, 30	, RA	, MA+27	, CON+4	, $
	, 31	, IJ	, TEMPS+5	, MA+9	, INDEX B EQUAL 0(127) $
	, 32	, RP3	, 128	, MA+34	, $
	, 33	, TP	, TEMPS+9	, FILL	, STORE REL LOC ON MD $
	, 34	, RA	, MA+33	, CON+14	, $
	, 35	, IJ	, TEMPS+3	, MA+7	, INDEX A EQUAL 0 $
	, 36	, TU	, RL+3	, MA+32	, $
	, 37	, TU	, DA	, RL	, $
	, 38	, TP	, DA+1	, RL+1	, $
	, 39	, TU	, TEMPS+2	, RL+4	, $
	, 40	, TV	, PAR+1	, RL+7	, $
	, 41	, RJ	, RL+14	, RL	, $
	, BL	, TP	, CON+15	, TEMPS+3	, 9 TO INDEX A $
	, 1	, TP	, PAR+4	, A	, PAR TO A $
	, 2	, ZJ	, BL+3	, EXIT	, EQUAL 0 $
	, 3	, TU	, TEMPS+2	, RL+4	, $
	, 4	, TU	, A	, RL+1	, $
	, 5	, TV	, A	, RL+7	, $
	, 6	, SJ	, BL+7	, BL+8	, CHECK FOR LAST PAR. $
	, 7	, RS	, TEMPS+3	, A	, CLEAR INDX A $
	, 8	, RJ	, RL+14	, RL	, $
	, 9	, RA	, BL+1	, CON+3	, $
	, 10	, IJ	, TEMPS+3	, BL+1	, INDEX A EQUAL 0(9) $
	, 11	, TU	, CON+8	, BL+1	, $
	, 12	, MJ	, 0	, EXIT	, TO EXIT $