RECOMP II USER'S PROGRAM NO. 1120

PROGRAM TITLE:	DATA PLOTTER, FLOATING POINT
PROGRAM CLASSIFICATION:	General
AUTHOR:	T. W. Lawhorn
PURPOSE:	To plot computed variables in graphical (rectangular) form using standard Recomp II equipment.
DATE:	3 August 1961

Published by

RECOMP User's Library

at

AUTONETICS INDUSTRIAL PRODUCTS
A DIVISION OF NORTH AMERICAN AVIATION, INC.
3400 E. 70th Street, Long Beach 5, Calif.

3 August 1961
T. W. Lawhorn
Dept. 3041-96 (Whittier)

PROGRAM TITLE: DATA PLOTTER, FLOATING POINT

1. Purpose: The purpose of this program is to plot computed variables in graphical (rectangular) form using standard RECOMP II equipment. For details of operation, one should consult the operations manual for this program.

2. Restrictions:

2.1 Other Equipment: While no other equipment is necessary, it is convenient to use graph paper instead of the usual output paper. The dimensions should be 10 squares to the inch across the carriage, and 6 squares to the inch through the roller for perfect correlation with the typewriter (this is not mandatory). RECOMP output format paper (Form 809-C-5) may be used but the ruling is frequently imperfect.

2.2 Subroutines: Two subroutines are required by the plotter program. The first of these converts mixed input quantities into normalized floating point numbers. No standard RECOMP subroutine exists for this application and the subroutine is therefore incorporated into the plotter. The standard floating to fixed point output routine AN-015 is required. This subroutine may be relocated with respect to the plotter or may be used in its standard location (0150_8-0427_8), provided the plotter is relocated so as not to conflict with it.

2.3 Data: Only three absolute restrictions exist on the plotter alone. The number of spaces used for the graph (B) must be an integral multiple of the number of tabs set into the carriage (N_T). Secondly,

all tabs must be of equal length. Thirdly, tabs must be longer than two spaces. The quantities L and R are used to set up the AN-015 calling sequence and thus all restrictions pertinent to AN-015 apply to the abcissa variable (I) in the plotter. The program has been designed to account for normal typewriter eccentricities, but the user would be well advised to examine his equipment for operations required here.

2.4 Computer Set-up: There are several special set-up requirements on this program. Tabs and margins must be adjusted to suit the plotting task under consideration. The user is referred to the operations manual for a detailed procedural description.

2.5 Manual Operations: Once the program has been initialized (see operations manual), no further manual operations are necessary (save for correcting paper drift in the typewriter).

3. Method: Basically, the program scales and plots one or two floating point "dependent" variables against a single fixed or variable increment "independent" variable. Due to the several operational modes, the reader is again referred to the operations manual for a more elaborate description.

3.1 Accuracy: The independent variable will be plotted to the nearest roller setting, leaving a maximum error of $\pm \frac{\text{Line Scale Value}}{2}$. The dependent variable(s) will be plotted to the nearest space, giving a maximum error of $\pm \frac{\text{Range of Plot}}{2\ (\text{number of spaces allotted})}$.

3.2 Range: This situation applies to the dependent variable(s). It is limited by input accuracy of mixed point input. No more than 11_{10} significant decimal digits in either side of the decimal point may be used. Round-off (binary) will be introducted after the fourth decimal digit after the decimal point.

4. Usage: See Operations Manual.

5. Coding Information: See attached block diagrams and coding forms.

5.1 Constants:

L	Value
0021	+ 39 @ B39
0022 - 0023	+ 1.0
0024 - 0025	+ 2.0
0026	+ 0
0076	+ 2 @ B39
0077	+ 1 @ B39
0167	+ 1 @ B17
0207	+ 3 @ B39
0216	+ 1 @ B16
0254	+ 1 @ B16
0256	+ 3 @ B17
0334	+ 1 @ B39
0337	+ 1 @ B17
0533	+ 1 @ B39

5.2 No erasable locations are available during plotter usage.

5.3 The average time required to plot one data point (one complete program cycle) is 4 seconds. The average time for two data points is ca. 5.6 seconds. When abcissa output is obtained,

the times given for AN-015, + .2 second, must be added to the above times. The above times are measured from situations using maximum resolution (B = 100 spaces) and 10 tabs. The operation time will decrease with B. It is not advisable to have tab lengths less than five spaces, except for small B.

6. Checkout:

6.1 Method: The program has been tested on over ten thousand data points, both functionally generated and contrived. Mechanical dissimilarities between the checkout equipment and the user's equipment may cause erratic performance, however. It is suggested the user familiarize himself with the plotter's requirements and test his equipment briefly.

Appendix A

GLOSSARY

A	Accumulator or A register
AV	Abcissa scale value
AV_i	Abcissa scale value (initial)
B	Number of allotted ordinate spaces
CODE i(i = 1,2)	Character code for 1st or 2nd D
D_i, D_i' (i = 1,2)	ith dependent variable
* $D_{CD} = B/N_T$	Spaces per tab
Δ_i (i = 1,2)	$\frac{D_i R}{B}$
D_{iLL}	Smallest plottable ordinate value of ith dependent variable
D_{iR}	Ordinate range ($D_{iUL} - D_{iLL}$) of ith dependent variable
I	Independent variable
Int.	Integer used by abcissa scale output with fixed increment I
L	Abcissa scale significant figures to left of decimal point
LSV	Line scale value - Abcissa increment per carriage return
N_T	Number of tabs
PC_i	Position counter. Numberical carriage position of plot point
PC_i'	General notation for PC_i
PC'	Present numerical carriage position (ordinate)
PI	Abcissa scale print interval
PI'	Abcissa scale print interval counter
R	Abcissa scale significant figures to right of decimal point
R_R	Remainder or R register
$S_i = D_{iLL} + \frac{D_i}{(i = 1,2)^2}$	Scaler for ith dependent variable
SF	Scale Factor for abcissa scale output with I increment fixed

Appendix A

T,Y,Z	Temporary storage
T_T	Tab tally. Counts used tabs.
X_p	Number of carriage spaces occupied by abcissa output

* Must be an integer

Program: Floating Point, Data Plotter Block Diagram

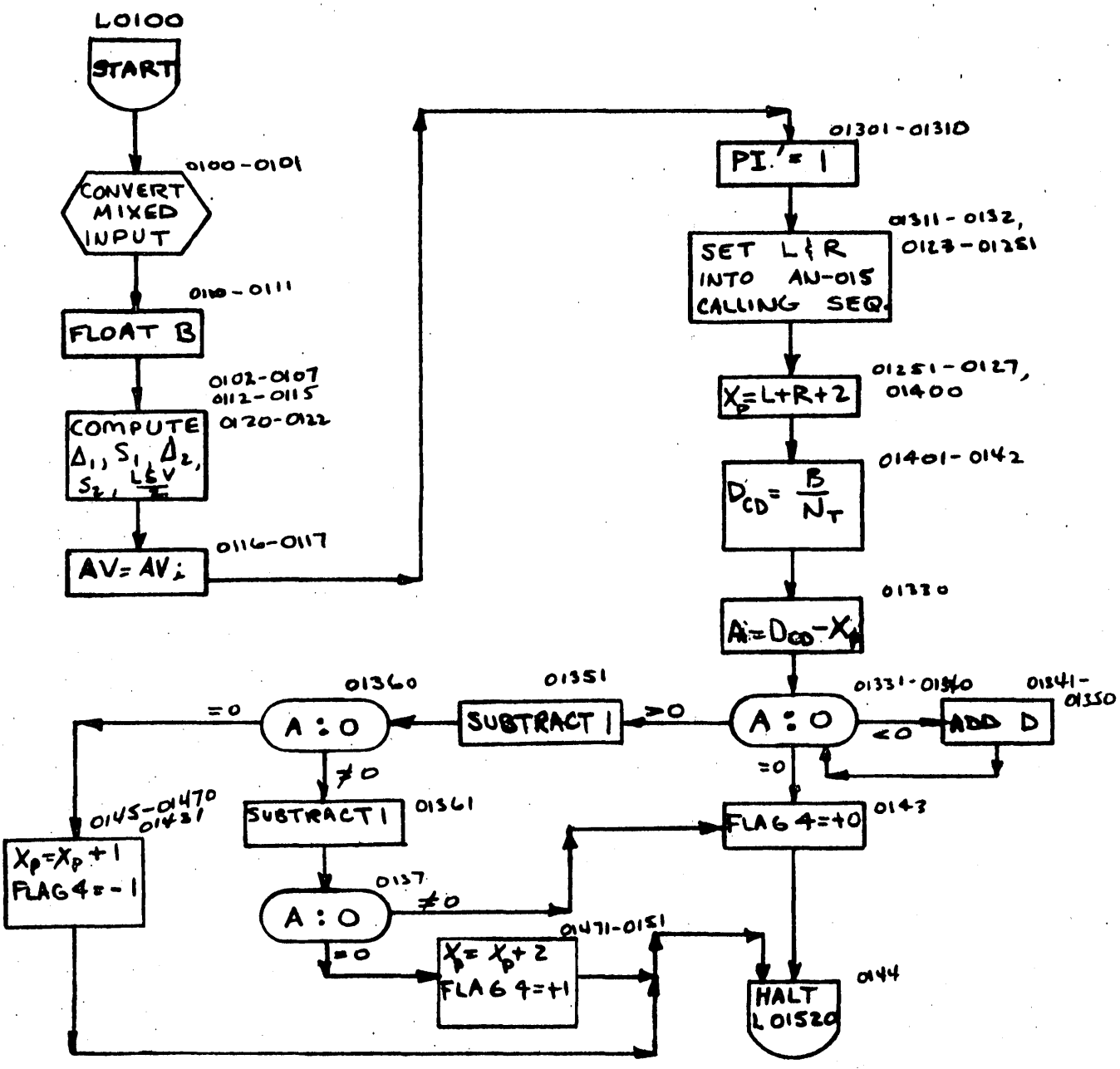
Flag Meaning:

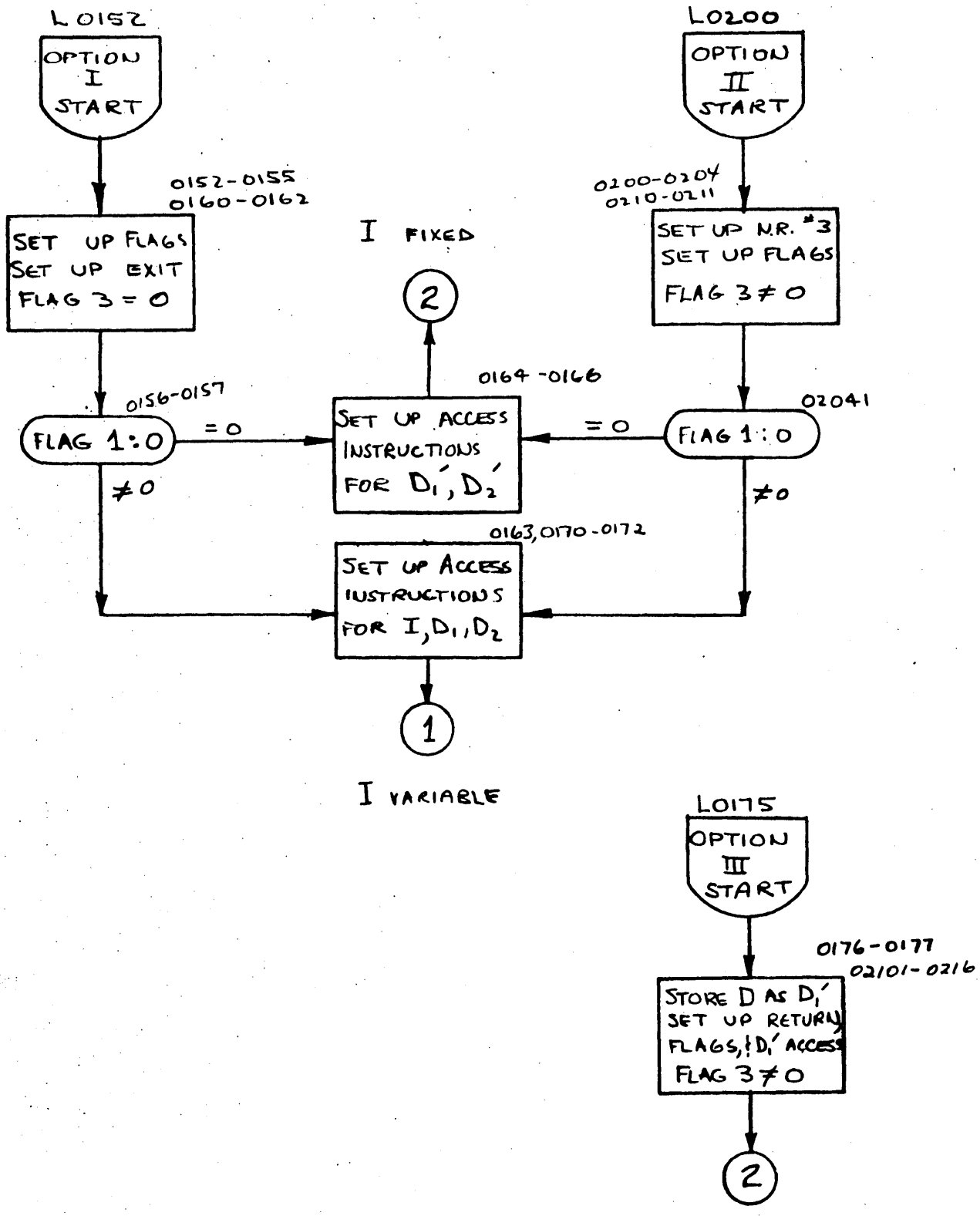
Flag 1 = X { = 0 fixed increment I
= 1 variable increment I

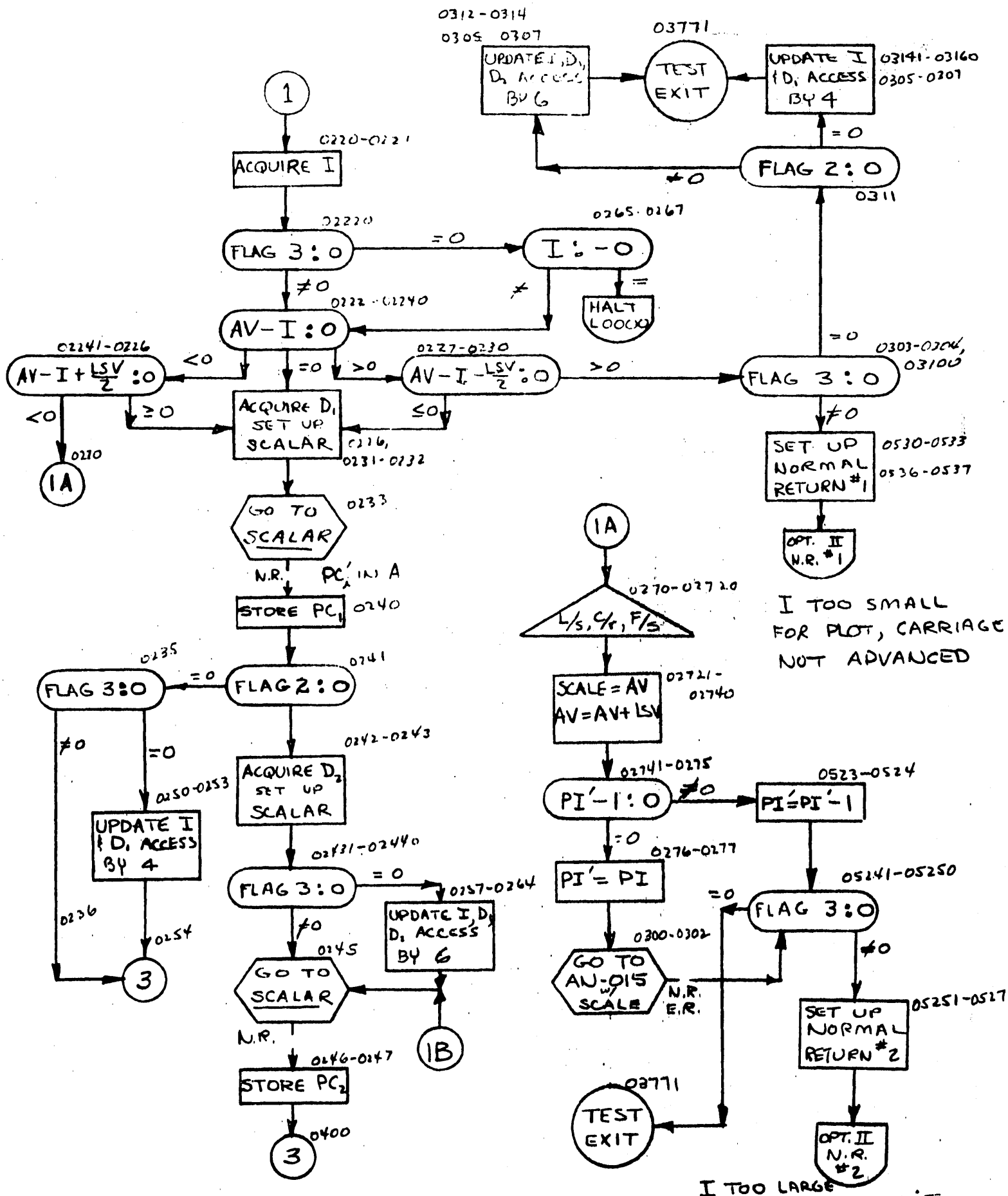
Flag 2 = Y { = 0 single D
= 1 D_1 and D_2

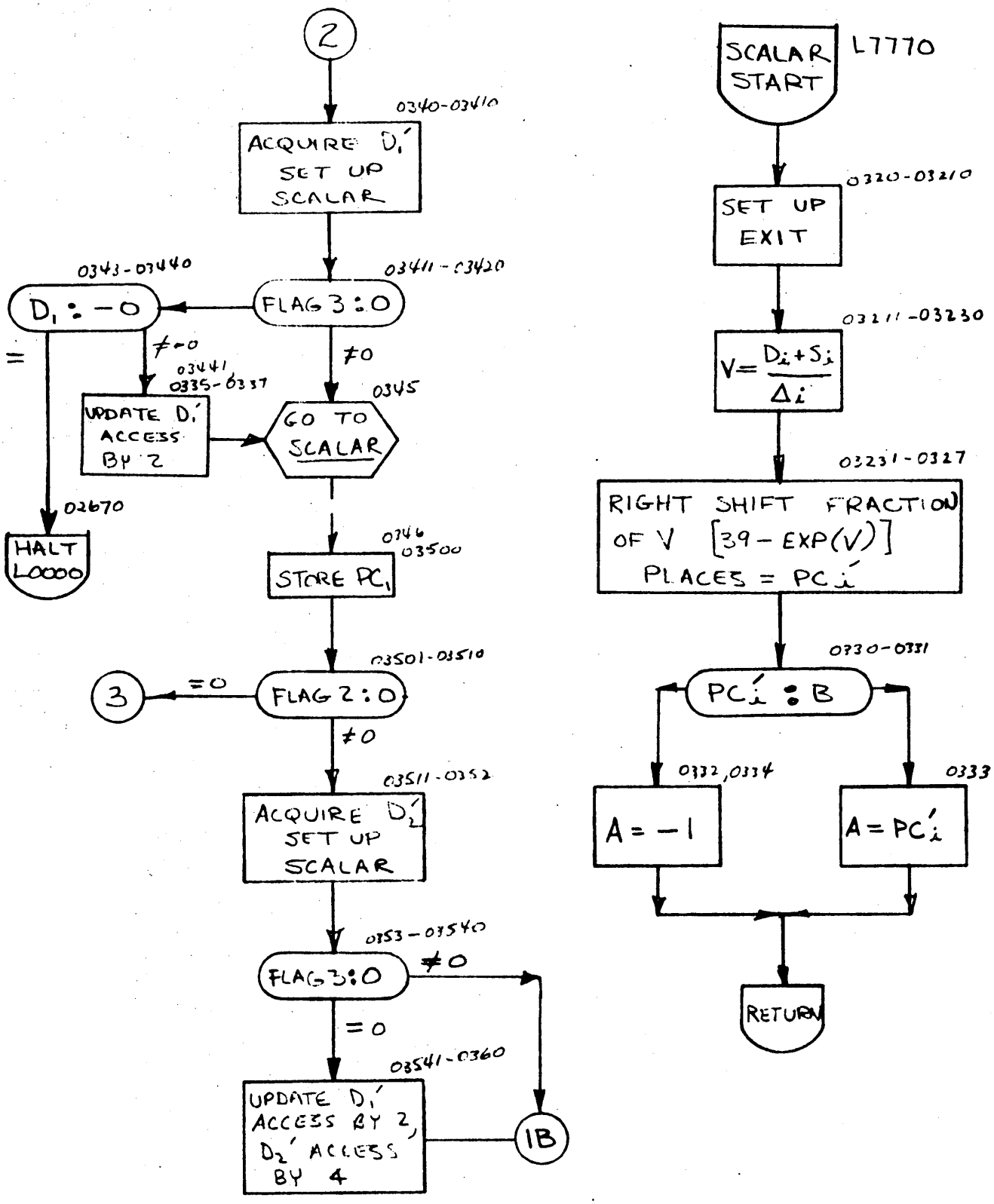
Flag 3 { = 0 fixed program
≠ 0 subroutine

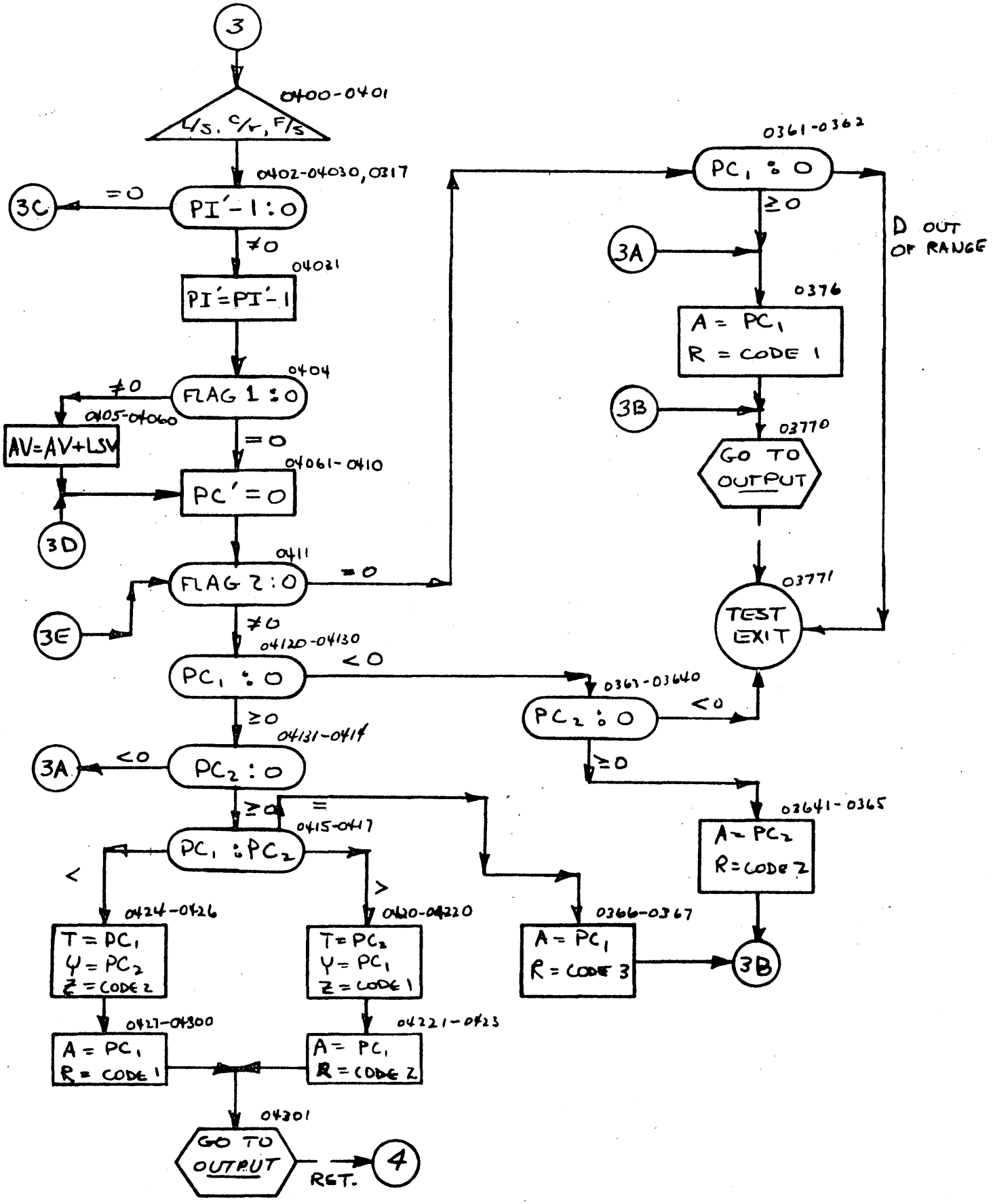
"Initialize"

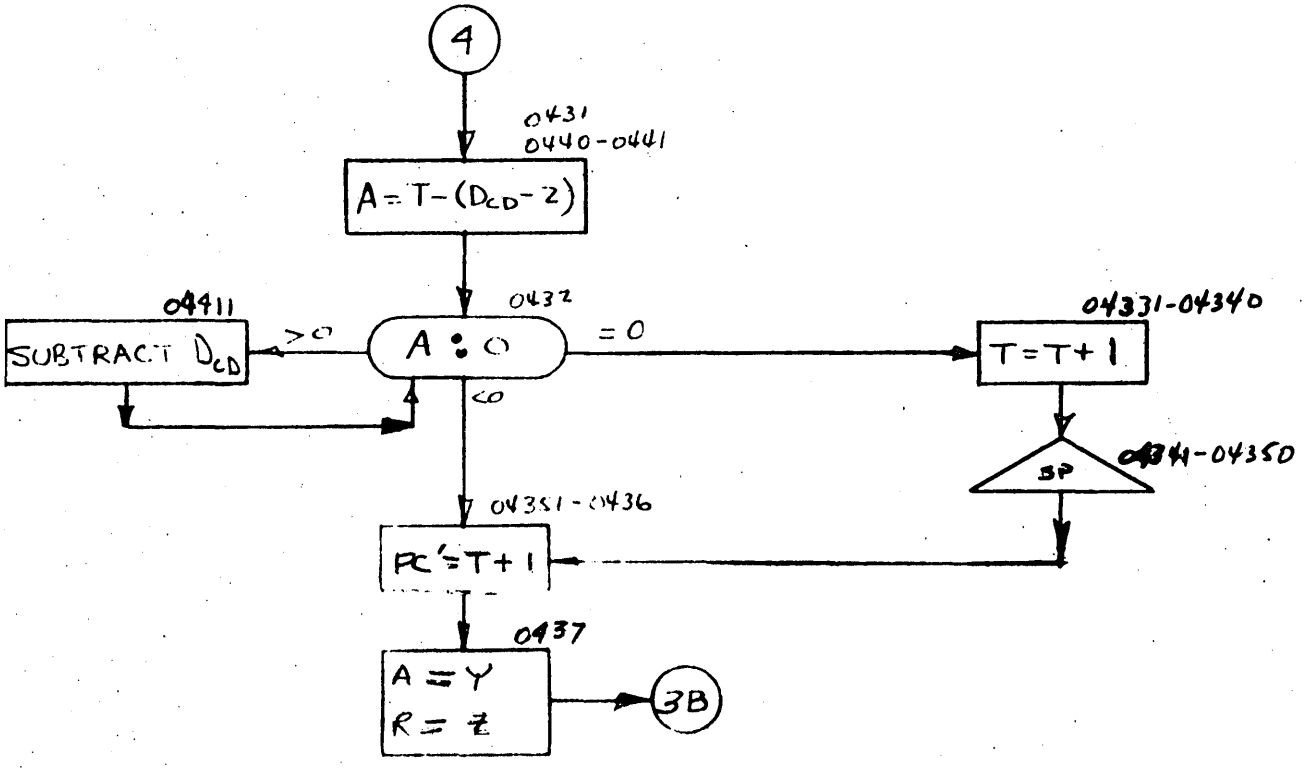


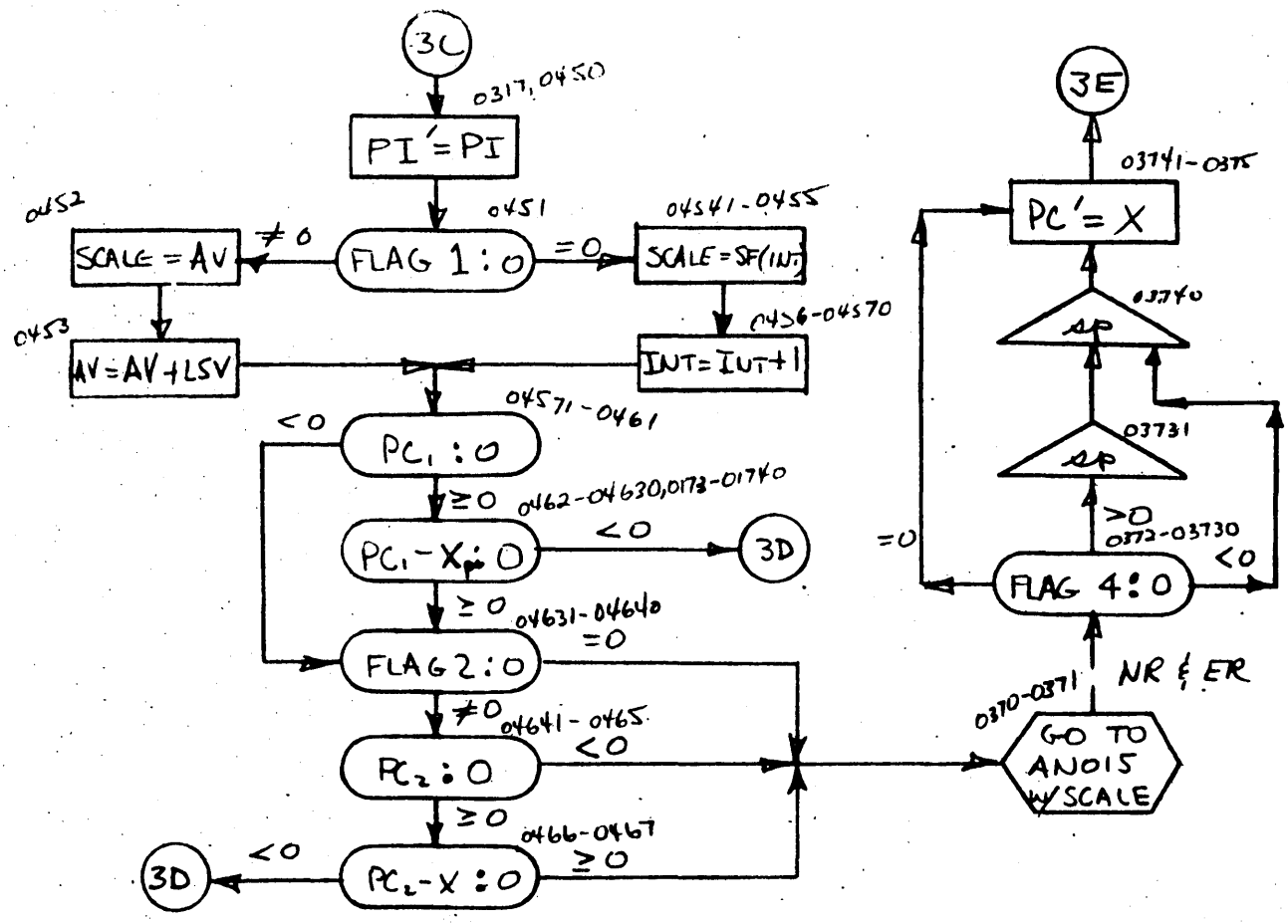

L0152
OPTION I START
0152-0155
0160-0162
SET UP FLAGS
SET UP EXIT
FLAG 3 = 0
I FIXED
L0200
OPTION II START
0200-0204
0210-0211
SET UP N.R. #3
SET UP FLAGS
FLAG 3 ≠ 0
2
0164-0166
0156-0157
FLAG 1:0
= 0
SET UP ACCESS INSTRUCTIONS FOR D1', D2'
= 0
02041
FLAG 1:0
≠ 0
≠ 0
0163, 0170-0172
SET UP ACCESS INSTRUCTIONS FOR I, D1, D2
1
I VARIABLE
L0175
OPTION III START
0176-0177
02101-0216
STORE D AS D1'
SET UP RETURN FLAGS, ! D1' ACCESS
FLAG 3 ≠ 0
2

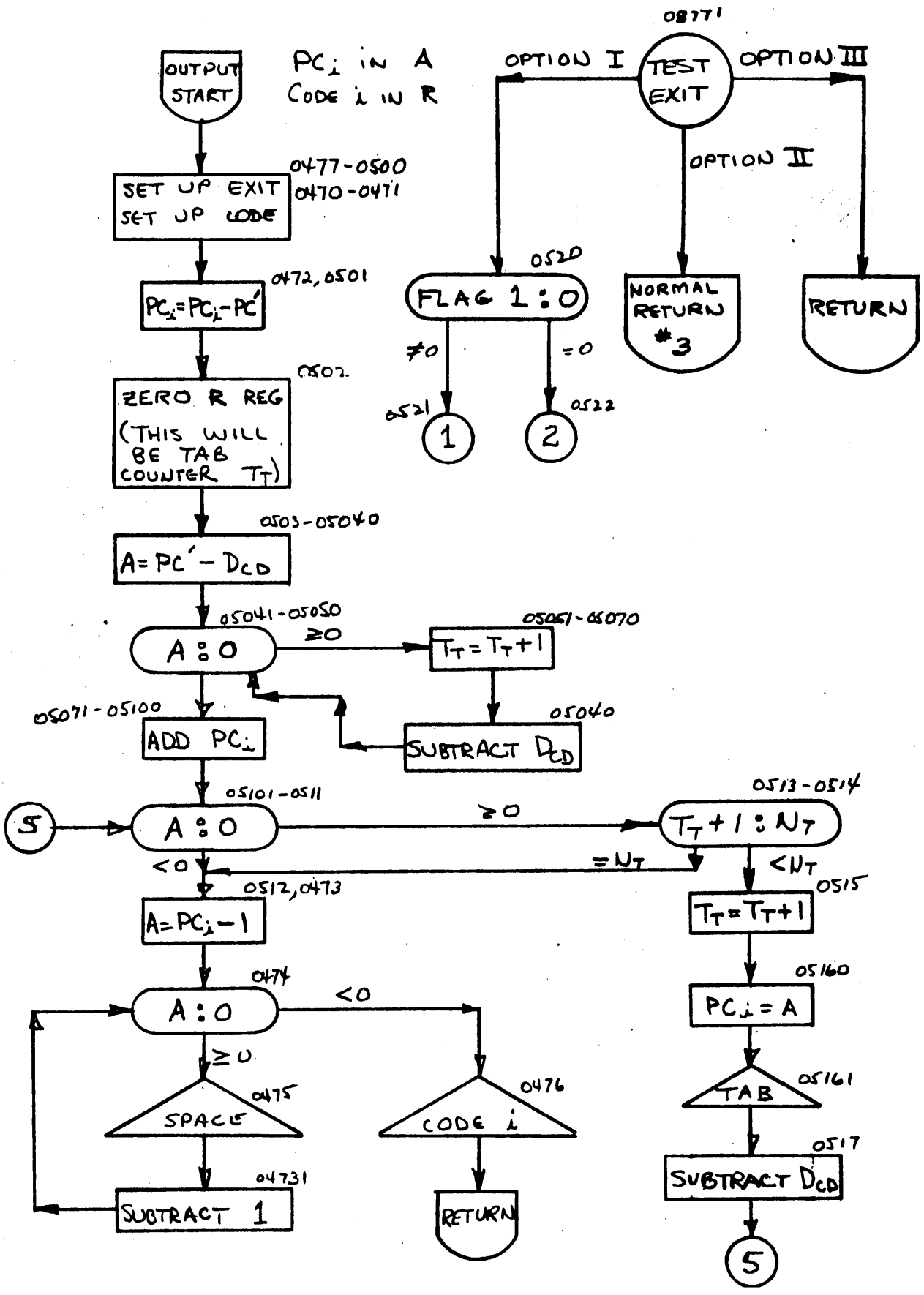

0312-0314
0305 0307
UPDATE I, D1, D2 ACCESS BY 6
03771
TEST EXIT
UPDATE I & D1 ACCESS BY 4
03141-03160
0305-0307
1
0220-0221
ACQUIRE I
FLAG 2 : 0
=0
≠0
0311
02220
FLAG 3 : 0
=0
0265-0267
I : -0
≠
=
HALT LOOOXO
≠0
0222-02240
AV-I : 0
02241-0226
AV-I+LSV/2 : 0
<0
=0
>0
0227-0230
AV-I-LSV/2 : 0
>0
FLAG 3 : 0
0303-0304, 03100
=0
≠0
<0
≥0
≤0
ACQUIRE D1 SET UP SCALAR
0226, 0231-0232
SET UP NORMAL RETURN #1
0530-0533
0536-0537
0210
1A
GO TO SCALAR
0233
N.R.
PC1 IN A
OPT. II N.R. #1
STORE PC1
0240
I TOO SMALL FOR PLOT, CARRIAGE NOT ADVANCED
0235
FLAG 3:0
=0
FLAG 2:0
0241
1A
0270-02720
L/S, C/R, F/S
≠0
=0
0242-0243
ACQUIRE D2 SET UP SCALAR
SCALE = AV
AV = AV + LSV
02721-02740
0250-0253
UPDATE I & D1 ACCESS BY 4
02741-0275
PI'-1 : 0
≠0
0523-0524
PI = PI'-1
02431-02440
=0
0276-0277
PI' = PI
FLAG 3 : 0
=0
0257-0264
UPDATE I, D1, D2 ACCESS BY 6
05241-05250
FLAG 3 : 0
≠0
0236
0254
0245
0300-0302
GO TO AN-015 w/ SCALE
N.R.
E.R.
GO TO SCALAR
N.R.
1B
0246-0247
STORE PC2
05251-0527
SET UP NORMAL RETURN #2
3
0400
03771
TEST EXIT
OPT. II N.R. #2
I TOO LARGE FOR PLOT - CARRIAGE ADVANCED


2
0340-03410
ACQUIRE D1'
SET UP
SCALAR
0343-03440
03411-03420
D1 : -0
FLAG 3 : 0
=
≠-0
≠0
03441
0335-0337
0345
UPDATE D1'
ACCESS
BY 2
GO TO
SCALAR
02670
HALT
L0000
0346
03500
STORE PC1
03501-03510
3
=0
FLAG 2 : 0
≠0
03511-0352
ACQUIRE D2'
SET UP
SCALAR
0353-03540
FLAG 3 : 0
≠0
=0
03541-0360
UPDATE D1'
ACCESS BY 2,
D2' ACCESS
BY 4
1B
SCALAR
START
L7770
0320-03210
SET UP
EXIT
03211-03230
V = Di+Si / Δi
03231-0327
RIGHT SHIFT FRACTION
OF V [39 - EXP(V)]
PLACES = PCi'
0330-0331
PCi' : B
0332,0334
0333
A = -1
A = PCi'
RETURN


3
0400-0401
L/S. C/r, F/S
0402-04030, 0317
PI'-1:0
=0
3C
≠0
04021
PI'=PI'-1
0404
FLAG 1:0
≠0
0405-04060
AV=AV+LSV
=0
04061-0410
PC'=0
3D
0411
FLAG 2:0
=0
3E
≠0
04120-04130
PC1 : 0
<0
0361-0362
PC1 : 0
≥0
3A
D OUT OF RANGE
0376
A = PC1
R = CODE 1
3B
03770
GO TO OUTPUT
03771
TEST EXIT
0367-03640
PC2 : 0
<0
≥0
04131-0414
PC2 : 0
<0
3A
≥0
=
0415-0417
PC1 : PC2
<
>
02641-0365
A = PC2
R = CODE 2
0366-0367
A = PC1
R = CODE 3
3B
0424-0426
T = PC1
Y = PC2
Z = CODE 2
0420-04220
T = PC2
Y = PC1
Z = CODE 1
0427-04300
A = PC1
R = CODE 1
04221-0423
A = PC1
R = CODE 2
04301
GO TO OUTPUT
RET.
4


4
0431
0440-0441
A = T-(D_CD-2)
04411
SUBTRACT D_CD
>0
0432
A : 0
=0
04331-04340
T=T+1
<0
04341-04350
SP
04351-0436
PC'=T+1
0437
A = Y
R = Z
3B


3C
0317, 0450
PI'=PI
0452
SCALE = AV
≠0
0451
FLAG 1 : 0
=0
04541-0455
SCALE = SF(INT)
0453
AV = AV + LSV
0456-04570
INT = INT+1
04571-0461
<0
PC1 : 0
≥0
0462-04630, 0173-01740
PC1 - X : 0
<0
3D
≥0
04631-04640
=0
FLAG 2 : 0
≠0
04641-0465
<0
PC2 : 0
≥0
0466-0467
≥0
PC2 - X : 0
<0
3D
3E
03741-0375
PC' = X
03740
SP
03731
SP
>0
0372-03730
<0
=0
FLAG 4 : 0
0370-0371
NR & ER
GO TO ANOIS W/SCALE


OUTPUT START
PC$_i$ IN A
CODE i IN R
0477-0500
0470-0471
SET UP EXIT
SET UP CODE
0472, 0501
PC$_i$=PC$_i$-PC′
0502
ZERO R REG
(THIS WILL BE TAB COUNTER T_T)
0503-05040
A=PC′ − D_{CD}
05041-05050
A : 0
≥0
05051-05070
T_T = T_T+1
05040
SUBTRACT D_{CD}
05071-05100
ADD PC$_i$
05101-0511
A : 0
<0
0512, 0473
A=PC$_i$−1
0474
A : 0
<0
≥0
0475
SPACE
04731
SUBTRACT 1
0476
CODE i
RETURN
5
0513-0514
T_T+1 : N_T
≥0
=N_T
<N_T
0515
T_T=T_T+1
05160
PC$_i$ = A
05161
TAB
0517
SUBTRACT D_{CD}
5
08771
TEST EXIT
OPTION I
OPTION II
OPTION III
0520
FLAG 1 : 0
≠0
=0
0521
1
0522
2
NORMAL RETURN #3
RETURN

DATA PLOTTER, FLOATING POINT, SYMBOLIC LISTING

```
0100.0
 + TRA 0540.0   + CLA 0006.0
 + CLA 0000.0   + CLA 0000.0
 + CTV 0103.0   + TRA 7770.0
 + FDV 7760.0   + FST 7764.0
 + FDV 0024.0   + FST 7770.0
 + FCS 7766.0   + FAD 7770.0
 + FST 7766.0   + CFL 0040.0
 + CTV 0113.0   + TRA 7770.0

0110.0
 + FCA 0020.0   + CTL 0000.0
 + FST 7762.0   + FNM 0000.0
 + FST 7760.0   + FCA 7764.0
 + FAD 7770.0   + FST 7766.0
 + FCA 0000.0   + FDV 0024.0
 + FST 0065.0   + CFL 0050.0
 + FCA 0002.0   + FST 0063.0
 + CTL 0123.0   + TRA 7760.0

0120.0
 + FCA 0010.0   + FDV 7760.0
 + FST 7764.0   + FDV 0024.0
 + FST 7770.0   + FCS 0012.0
 + CLA 7775.0   + ALS 0025.0
 + ADD 7760.0   + STA 0301.0
 + STA 0371.0   + CLA 7774.0
 + ADD 7775.0   + CTV 0133.0
 + CTL 0017.0   + ADD 7765.0

0130.0
 + CTV 0014.0   + CLA 7773.0
 + STO 0067.0   + CLA 7774.0
 + ALS 0033.0   + STO 7760.0
 + SUB 7761.0   + TZE 0143.0
 + TPL 7775.1   + ADD 7760.0
 + TRA 7773.1   + SUB 7763.0
 + TZE 0145.0   + SUB 7763.0
 + TZE 0147.1   + TRA 0143.0

0140.0
 + STO 7761.0   + CLA 7760.0
 + XAR 0000.0   + CLA 7766.0
 + DIV 7767.0   + STO 7760.0
 + CLA 7766.0   + STO 7767.0
 + CFL 0022.0   + HTR 0152.0
 + CLA 7761.0   + ADD 7763.0
 + STO 7761.0   + CLS 7763.0
 + TRA 0143.1   + CLA 7761.0

0150.0
 + ADD 7765.0   + STO 7761.0
 + CLA 7763.0   + TRA 0143.1
 + CTL 0153.0   + TRA 7762.0
 + ARS 0024.0   + STA 7770.1
 + FCA 7760.0   + STA 0377.1
 + ARS 0024.0   + STO 7772.0
 + CLA 7770.0   + CFV 0060.0
 + CTV 0163.0   + TZE 7774.0

0160.0
 + CLA 0000.0   + CLA 0520.0
 + CLA 0000.0   + CLA 0000.0
 + CTV 0060.0   + STA 7771.1
 + CTL 0220.0   + TRA 7760.0
 + XAR 0000.0   + STA 0340.0
 + ADD 7777.0   + STA 0352.0
 + CTV 0335.0   + TRA 7770.0
 + CLA 0002.0   - CLA 0000.0

0170.0
 + XAR 0000.0   + STA 0220.0
 + ADD 7777.0   + STA 0231.0
 + ADD 7777.0   + STA 0242.0
 + CLA 0026.0   + CTV 0410.0
 + TRA 7770.0   - CLA 0000.0
 + FST 0040.0   + SAX 7760.0
 + CTL 0207.0   + CTV 0060.0
 + ADD 0023.0   + TRA 7760.1
```

```
0200.0
 + CTV 0060.0   + SAX 7771.0
 + CTL 0202.0   + TRA 7760.0
 + STA 7770.1   + CFV 0060.0
 + CTV 0163.0   + STO 7760.0
 + ALS 0024.0   + TZE 7766.0
 + CLA 7760.0   + TRA 7770.1
 + CLA 7760.0   + TRA 7774.1
 + CLA 0000.0   - CLA 0001.1

0210.0
 + ADD 7767.0   + STA 0377.1
 + STA 7772.1   + XAR 0000.0
 + CLA 7766.0   + STA 0340.0
 + STA 7771.1   + STA 7770.1
 + CFV 0060.0   + CTV 0335.0
 + TRA 7770.0   - CLA 0000.0
 + CLA 0040.0   + CLA 0000.0
 - CLA 0000.0   - CLA 0000.0

0220.0
 + FCA 0000.0   + CTV 0060.0
 + FST 7760.0   + CLA 7772.0
 + TZE 0265.0   + FCA 0063.0
 + FSB 7760.0   + TZE 7766.0
 + TPL 7767.0   + FAD 7775.0
 + TZE 7766.0   + TMI 0270.0
 + CTV 0230.0   + TRA 7771.0
 + FSB 7775.0   + CTV 0230.0

0230.0
 + TZE 7771.0   + TPL 0303.0
 + FCA 0000.0   + CTL 0040.0
 + FST 7760.0   + TRA 0233.0
 + CTV 0320.0   + TRA 7770.0
 + CTV 0235.0   + TRA 7770.0
 + CLA 7762.0   + TZE 7777.0
 + CTL 0400.0   + TRA 7760.0
 + CTL 0250.0   + TRA 7760.0

0240.0
 + CTL 0060.0   + STO 0033.0
 + CLA 7761.0   + TZE 7775.0
 + FCA 0000.0   + CTL 0050.0
 + FST 7760.0   + CLA 0062.0
 + TZE 0257.0   + TRA 0245.0
 + CTV 0320.0   + TRA 7770.0
 + STO 0034.0   + CTL 0400.0
 + TRA 7760.0   - CLA 0000.0

0250.0
 + CLA 0220.0   + ADD 7764.0
 + STA 0220.0   + CLA 0231.0
 + ADD 7764.0   + STA 0231.0
 + CTL 0400.0   + TRA 7760.0
 + CLA 0004.0   - CLA 0000.0
 - CLA 0000.0   - CLA 0000.0
 + CLA 0006.0   - CLA 0000.0
 + CTV 0255.0   + TRA 7770.0

0260.0
 + CLA 0220.0   + ADD 7776.0
 + STA 0220.0   + CLA 0231.0
 + ADD 7776.0   + STA 0231.0
 + CLA 0242.0   + ADD 7776.0
 + STA 0242.0   + TRA 0245.0
 + CLA 7760.0   + TPL 7762.1
 + TZE 0267.0   + TRA 7762.1
 + HTR 0000.0   - CLA 0000.0

0270.0
 + CTL 0270.0   + TRA 7761.0
 + TYC 0037.0   + TYC 0010.0
 + TYC 0033.0   + FCA 0063.0
 + FST 0070.0   + FAD 0000.0
 + FST 0063.0   + CLA 0067.0
 + SUB 0023.0   + TZE 7766.1
 + TRA 0523.0   + CLA 0016.0
 + STO 0067.0   + TRA 0300.0
```

```
0300.0
 + FCA 0070.0   + TRA 0000.0
 + CLA 0000.0   + ARS 0000.0
 + CTL 0524.0   + TRA 7764.1
 + CTL 0305.0   + CTV 0060.0
 + CLA 7772.0   + TRA 7760.0
 + ADD 7760.0   + STA 0231.0
 + CLA 0220.0   + ADD 7760.0
 + STA 0220.0   + TRA 0377.1

0310.0
 + TZE 7761.0   + TRA 0530.0
 + CLA 7771.0   + TZE 0315.0
 + CLA 0256.0   + STO 7760.0
 + CLA 0242.0   + ADD 7760.0
 + STA 0242.0   + CLA 0231.0
 + CLA 0254.0   + STO 7760.0
 + TRA 7764.1   - CLA 0000.0
 + CTL 0450.0   + TRA 7760.0

0320.0
 + SAX 7770.0   + ADD 0023.0
 + STA 0332.1   + FCA 7760.0
 + FAD 7766.0   + FDV 7764.0
 + FST 7760.0   + CLA 7763.0
 + SUB 7761.0   + ALS 0025.0
 + STA 7776.0   + CLA 7760.0
 + ARS 0000.0   + STO 7760.0
 + CTV 0330.0   + TRA 7770.0

0330.0
 + SUB 7762.0   + TZE 7773.0
 + TMI 7773.0   + ARS 0000.0
 + CLS 7774.0   + TRA 0000.0
 + CLA 7760.0   + TRA 7772.1
 + CLA 0000.0   - CLA 0000.1
 + ADD 7777.0   + STA 0340.0
 + TRA 0345.0   - CLA 0000.0
 + CLA 0002.0   - CLA 0000.0

0340.0
 + FCA 0000.0   + CTL 0040.0
 + FST 7760.0   + CLA 0062.0
 + TZE 7773.0   + TRA 0345.0
 + CLA 7760.0   + TPL 7774.1
 + TZE 0267.0   + CLA 7770.0
 + CTV 0320.0   + TRA 7770.0
 + CTV 0347.0   + TRA 7770.0
 + CTL 0400.0   + TRA 7760.0

0350.0
 + STO 0033.0   + CLA 0061.0
 + TZE 7777.0   + CTL 0050.0
 + FCA 0000.0   + FST 7760.0
 + CLA 0062.0   + TZE 7774.1
 + TRA 0245.0   + CTV 0352.0
 + CLA 0340.0   + ADD 0337.0
 + STA 0340.0   + CLA 7772.0
 + ADD 0254.0   + TRA 7770.0

0360.0
 + STA 0352.0   + TRA 0245.0
 + CLA 7763.0   + TZE 0376.0
 + TMI 0377.1   + TRA 0376.0
 + CLA 7764.0   + TZE 0364.1
 + TMI 0377.1   + FCA 7765.0
 + CLA 7764.0   + TRA 0377.0
 + FCA 7766.0   + CLA 7763.0
 + TRA 0377.0   - CLA 0000.0

0370.0
 + FCA 0070.0   + TRA 0000.0
 + CLA 0000.0   + ARS 0000.0
 + CLA 0027.0   + TZE 0374.1
 + TMI 0374.0   + TYC 0004.0
 + TYC 0004.0   + CLA 0031.0
 + CTV 0410.0   + TRA 7770.0
 + FCA 7764.0   + CLA 7763.0
 + TRA 0477.0   + TRA 0000.0
```

```
0400.0
 + TYC 0037.0  + TYC 0010.0
 + TYC 0033.0  + CTV 0060.0
 + CLA 7777.0  + SUB 0023.0
 + TZE 0317.0  + STO 7777.0
 + CLA 7770.0  + TZE 7766.1
 + FCA 7773.0  + FAD 0000.0
 + FST 7773.0  + CLA 0026.0
 + CFV 0060.0  + CTV 0410.0

0410.0
 + STO 0032.0  + CTL 0030.0
 + CLA 0061.0  + TZE 0361.0
 + CLA 7763.0  + TZE 7773.1
 + TMI 0363.0  + CLS 7764.0
 + TZE 7775.0  + TPL 0376.0
 + ADD 7763.0  + TZE 0366.0
 + CTL 0420.0  + TMI 7764.0
 + CTV 0030.0  + TRA 7760.0

0420.0
 + CLA 7774.0  + STO 0072.0
 + FCA 7774.0  + CLA 7773.0
 + FST 0073.0  + FCA 7775.0
 + CLA 7774.0  + TRA 0430.1
 + CTV 0030.0  + CLA 7773.0
 + STO 0072.0  + FCA 7775.0
 + CLA 7774.0  + FST 0073.0
 + FCA 7774.0  + TRA 0430.0

0430.0
 + CLA 7773.0  + TRA 0477.0
 + CTL 0432.0  + TRA 7760.0
 + TZE 7763.1  + TMI 7765.1
 + TRA 7761.1  + CLA 7772.0
 + ADD 7777.0  + TYC 0004.0
 + TRA 7766.0  + CLA 7772.0
 + ADD 7777.0  + STO 0032.0
 + FCA 7773.0  + TRA 0377.0

0440.0
 + CTV 0070.0  + CLA 7772.0
 + ADD 7776.0  + SUB 0030.0
 - CLA 0000.0  - CLA 0000.0
 - CLA 0000.0  - CLA 0000.0
 - CLA 0000.0  - CLA 0000.0
 - CLA 0000.0  - CLA 0000.0
 - CLA 0000.0  - CLA 0000.0
 - CLA 0000.0  - CLA 0000.0

0450.0
 + CLA 0016.0  + STO 0067.0
 + CLA 7770.0  + TZE 7764.1
 + FCA 7773.0  + FST 0070.0
 + FAD 0000.0  + FST 0063.0
 + TRA 7767.1  + FCA 0000.0
 + FMP 0002.0  + FST 0070.0
 + FCA 0002.0  + FAD 0022.0
 + FST 0002.0  + CTV 0460.0

0460.0
 + CTL 0030.0  + CLA 7763.0
 + TZE 7772.0  + TMI 7773.1
 + SUB 7761.0  + TZE 7773.1
 + TMI 0173.0  + CLA 0061.0
 + TZE 0370.0  + CLA 7764.0
 + TZE 7776.0  + TMI 0370.0
 + SUB 7761.0  + TZE 0370.0
 + TMI 0173.0  + TRA 0370.0

0470.0
 + ADD 0023.0  + STA 7776.1
 + XAR 0000.0  + STA 7776.0
 + CLA 7760.0  + TRA 7761.0
 + CTL 0020.0  + SUB 7763.0
 + TZE 7775.0  + TMI 7776.0
 + TYC 0004.0  + TRA 7773.1
 + TYC 0000.0  + TRA 0000.0
 + CTL 0500.0  + SAX 7760.0
```

```
0500.0
 + CTV 0470.0  + TRA 7770.0
 + SUB 0032.0  + STO 7760.0
 + FCA 0025.0  + CFV 0470.0
 + CTV 0023.0  + CLA 7772.0
 + SUB 7770.0  + TZE 7765.1
 + TMI 7767.1  + XAR 0000.0
 + ADD 7773.0  + XAR 0000.0
 + TRA 7764.0  + CTV 0510.0

0510.0
 + ADD 7760.0  + TZE 7773.0
 + TPL 7773.0  + ARS 0000.0
 + CLA 7760.0  + CTV 0470.0
 + XAR 0000.0  + ADD 0023.0
 + SUB 0017.0  + TZE 7772.0
 + ADD 0017.0  + XAR 0000.0
 + STO 7760.0  + TYC 0010.0
 + SUB 0030.0  + TRA 7770.1

0520.0
 + CLA 0060.0  + TZE 0522.0
 + CTL 0220.0  + TRA 7760.0
 + CTV 0335.0  + TRA 7770.0
 + CTL 0524.0  + TRA 7764.0
 + STO 0067.0  + CLA 0062.0
 + TZE 0377.1  + CLA 0377.0
 + SUB 7763.0  + STA 7767.1
 + ARS 0000.0  + TRA 0000.0

0530.0
 + CTL 0524.0  + TRA 7761.0
 + CLA 0377.0  + SUB 7763.0
 + TRA 7766.0  - CLA 0000.0
 + CLA 0000.0  - CLA 0000.1
```

MIXED TO FLOATING POINT INPUT UTILITY ROUTINE

```
0540.0
 + SAX 7760.0   + CTL 0552.0
 + CTV 0562.0   + STA 0543.1
 + ADD 7763.0   + STA 7772.1
 + ARS 0000.0   + FCA 0000.0
 + EXT 0551.0   + TZE 7772.1
 + ARS 0001.0   + STO 7760.0
 + XAR 0000.0   + STA 7764.1
 + ARS 0024.0   + STA 7761.1

0550.0
 + TRA 7761.1   - CLA 0000.0
 + CLA 0000.0   - CLA 7777.0
 + FST 7774.0   + FCA 7773.0
 + XAR 0000.0   + FNM 0002.0
 + FAD 7775.0   + FST 0000.0
 + FCA 7760.0   + SUB 7777.0
 + TZE 7772.1   + STO 7760.0
 + XAR 0000.0   + ADD 7763.0

0560.0
 + CLA 0000.0   - CLA 0000.0
 + CLA 0000.0   + FCA 0000.0
 + TRA 7761.1   + TRA 0000.0
 + CLA 0000.0   - CLA 0023.1
 + CLA 0000.0   - CLA 0000.0
 + CLA 0000.0   - CLA 0000.0
 + CLA 0000.0   - CLA 0000.0
 + CLA 0000.0   - CLA 0000.1

0570.0
 + STA 7761.1   + CLA 7764.0
 + ADD 7763.0   + STA 7764.1
```

CALLING SEQUENCE

```
SLL

TRA  0540.0    PZE  XXXX.0
PZE  YYYY.0    PZE  ZZZZ.0
NORMAL RET.
```

WHERE

XXXX.0 IS OCTAL NUMBER OF MIXED NUMBERS TO BE CONVERTED,

YYYY.0 IS THE OCTAL ORIGIN OF THE RAW MIXED NOS.,

ZZZZ.0 IS THE OCTAL ORIGIN OF THE DESIRED STORAGE FOR THE FLOATED NOS.

OPERATIONS MANUAL FOR DATA PLOTTER

1. General: This data plotter will rapidly convert computed data to graphical form using standard RECOMP II peripheral equipment. One or two dependent variables are plotted across the carriage (ordinate) against a single independent variable (through the roller = abcissa). The independent variable may be incremented (external to the plotter) by fixed amounts or by variable amounts. Since the difference allows for plotter simplification, this information must be given to the program.

   This program is particularly useful for plotting system output versus system input in simulations, etc., or for merely plotting functional variations in a computed quantity. It is also useful for preparation of document illustrations since no visible grid is necessary for accurate plotting. Since the same piece of graph paper can be re-circulated through the typewriter, many variables can be plotted on the same graph.

   Appendix A contains symbology used in this document.

2. Usage: In order to enhance the value of the plotter, several modes of operation have been incorporated into the program. Provision is made to utilize the program as an addressable subroutine so that on-line data plotting may be carried out during the prime program's operation. Provision is also made to use the plotter as a fixed program which will plot blocks of data automatically sorted from a given block of main memory locations. Data points are then limited by the capacity of the memory space allotted.

   2.1 Relocation: Before relocation, the user must make several decisions. First, he must decide where to locate the plotter, occupying 571_8

location, and AN-015, which occupies 260_8 locations. None of these locations may coincide. Lastly, the user must decide the origin of the blocks of data to be plotted. This is necessary when using the plotter as an object program. With these decisions made, the relocation process may be undertaken.

2.1.1 Relocation Procedure:

1. Enter the relocatable plotter program tape. It will set the location counter to 0161.0_8.
2. From the console, enter the following:
   a. C+00 XXXX.0+00 0000.0E, where XXXX.0 is the origin of the block assigned to data storage for the plotter. Remember, the data must be in normalized floating point form. See Section 2.4.1 below for complete details on this point.
   b. L0300.0E

      C+30 00700+57 YYY0.0E, where YYY0.0 is the origin of AN-015 as decided above.
   c. L0370.0E

      C+30 00700+57 YYY0.0E

      You are now ready to relocate.[1]

      NOTE: Step (1) below may be performed at this point if desired.
   d. L7724.0E

      C+00 00000-00ZZZ0.0ES, where ZZZ0.0 is the relocated origin of the plotter.

[1] The relocation program supplied is modified to punch the new tape in alpha-numeric format and to self-correct punching errors. If the operator's RECOMP has not been modified for this operation, enter AN-004 before performing step (d).

The new plotter tape will be punched and a normal halt will be obtained upon completion with the location counter at L7725.0. If duplicate tapes are desired, the start button may be depressed, after a leader has been run out on the tape punch. If step (d) is performed incorrectly, an error return to L7724.0 will be obtained and step (d) should be repeated (see restrictions on AN-004).

e. Relocate AN-015. This step may be eliminated if prepared tapes of AN-015 are to be used.

2.2 Initialization: Regardless of the mode of operation to be used, certain preparations are necessary for plotter operation. These operations set the scales to be used, set abcissa increment information, inform the computer of typewriter tab settings, etc. The procedure for initialization is as follows:

a. Set the carriage return/tab select toggle switch to tab position. This switch is located under the typewriter ribbon cover on the right-hand front edge of the opening. (Tab position means toggle toward roller.)

b. Turn tape punch power "off" and set "Computer-Manual" select switch to manual.

c. Determine number of spaces to be allotted for ordinate (dependent) scale. This number (B) may not exceed 100 and on some typewriters 90 - 95 spaces is the maximum available. The larger B is, the greater the graphical resolution over a given range will be.

d. Set the left hand margin to the position desired. It must be extreme left if B $>$90. Otherwise, the left margin and B will "locate" the graph on the paper. The aesthetic aspects of the graph's appearance and location can be determined in this manner to suit the user's taste.

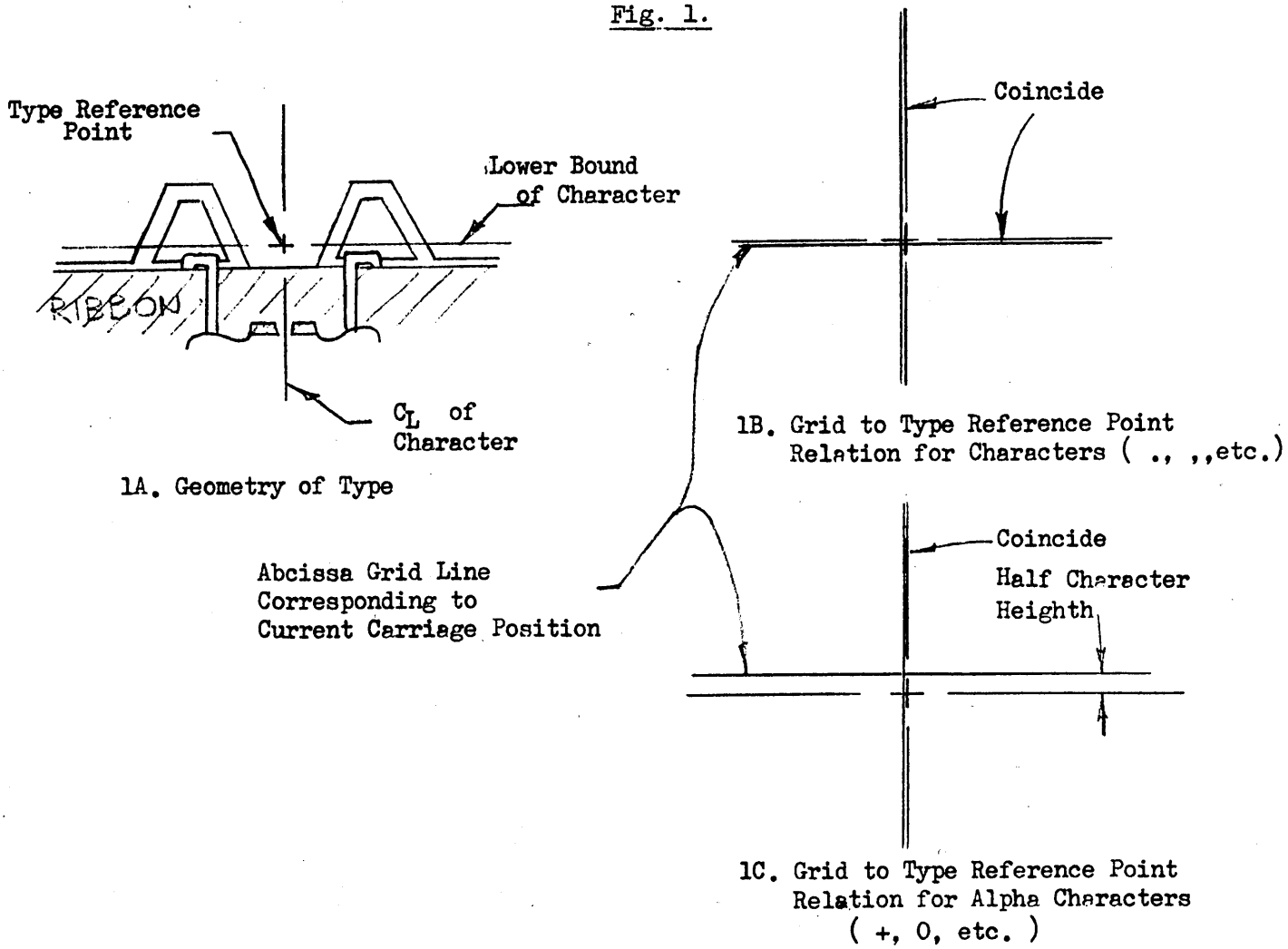
e. Set the right hand margin extreme right. This will avoid later mechanical difficulties.

f. Set tab stops. It is wise to type a carriage return and then type tabs, clearing all stops, until the carriage hits the right margin. Type a second carriage return, and then type the appropriate number of spaces chosen, setting tabs after each set of spaces. There is no need to set the final tab at the uppermost ordinate position. Due to mechanical restrictions, the plotter may not tab into the upper ordinate value. (For maximum resolution (B = 100), no tab setting exists.) The purpose of the tabs is to allow more rapid access to a given carriage position. The number of tabs used is left to the user's discretion. However, two restrictions exist. First, the number of sapces (B) divided by the number of tabs (N_T) must be an integer (obviously N_T cannot be zero). Secondly, all tab lengths must be of equal numbers of spaces, and tabs of less than three spaces are not used. This latter necessity stems from the mechanical limitation that a tab stop located within two spaces of the current carriage position will be ignored if a tab is typed, thereby giving a plotting error equal to $\frac{B}{N_T}$ spaces for second plot points (in two variable plottings). In the above

discussions, it is noted that "spaces" is used in describing B and N_T rather than type points. Actually, a character may be typed at either end of a space, so that B + 1 character locations will be available for an allotted B spaces.

g. Type a carriage return and enter the paper selected. Adjust the paper in the roller such that the characters typed will fall in the proper grid location. Always aline the least ordinate value grid line with the carriage return position. Move carriage back and forth to check horizontal (ordinate) type tracking of the grid. Experimentation by the user with his equipment is suggested to gain experience for this operation. Fig. 1 illustrates operation (g).

Fig. 1.



1A. Geometry of Type

1B. Grid to Type Reference Point Relation for Characters (., ,,etc.)

1C. Grid to Type Reference Point Relation for Alpha Characters (+, O, etc.)

h. Enter ordinate scales. The user may enter the ordinate scale(s) across the carriage by manual typewriter operation. When two variables are being plotted, the ordinate scales need not be the same. The location of the ordinate scales is irrelevant to the plotter. However, upon completion of this operation, the carriage should be advanced (or rolled backwards) until the roller position is one carriage return away from the first abcissa plot value. The plotter will type a carriage return before plotting the first point.

i. Return the "Computer-Manual" select switch to "Computer." The tape punch may now be operated at the user's discretion.

These steps have completed typewriter set-up for plotting. After a little practice, the user will find he performs these steps rapidly. For repeated use of the plotter, only steps b, g, h, and i above may need be repeated. All the following steps (j and k) must be repeated, however.

j. From the console, set

LOOOO.OE.

In sequential locations, enter the following mixed decimal numbers.

1. LSV or SF. If fixed increment I is to be used, this number (SF) will correspond to the abcissa increment between consecutive abcissa scale outputs. If variable increment I is to be used, this number (LSV) will correspond to the abcissa increment between consecutive abcissa divisions (i.e., between consecutive carriage return).

2. Int. or AV_i. If fixed increment I is to be used, this number (Int.) will be multiplied by SF to give the initial abcissa scale output, and will be incremented by one each scale output. If variable increment I is to be used, this number (AV_i.) will set the initial abcissa scale value, and will be updated by ISV each carriage return.

3. D_{1R}. The algebraic range of the ordinate scale assigned to the first dependent variable.

   Ex. Plot sine ω t vs t. $R = 1-(-1) = 2$.

   NOTE: Both D_{1R} and D_{2R} should be positive; i.e., upper limit minus lower limit.

4. D_{1LL}. The algebraically smallest value (of the first dependent variable) which is to be plotted; i.e., the lowest ordinate scale value.

5. D_{2R}. Same as D_{1R} (in 3 above) except applies to second dependent variable. Does not need to equal D_{1R}. If D_2 is not used, set to zero.

6. D_{2LL}. Same as D_{1LL} (in 4 above) except applies to second dependent variable. Need not equal D_{1LL}. If D_2 is not used, set to zero.

The location counter should now indicate L0014.0. Enter the following fixed point numbers at a binary scale of 39.

7. L. The number of significant decimal digits to the left of the decimal point desired for abcissa scale output. (See restrictions on AN-015.)

8. R. Same as L (in 7 above) except to right of decimal point.

9. PI. Number of abcissa units (carriage returns) between consecutive abcissa scale outputs.

10. N_T. Number of tabs set into typewriter. Count the last tab even if no tab setting was entered. (See step (f) above.)

11. B. Number of spaces allotted to ordinate scale.

The location counter should indicate L0021.0.

k. Set L0100.0. Start. In one second a halt at L0152.0 will be obtained. This completes the necessary initialization of the plotter. It can now plot in one of its modes. The user has a choice over the characters to be typed, however, and may elect to change them at this point.

l. Character modification. As programmed, a decimal point will be typed for D_1, a comma for D_2, and a colon for $D_1 = D_2$. These character codes are located in L0035.0 through L0037.0 respectively. To change, set the location to the desired code and enter C+00 00XX.0-00 0000.0E, where XX is the octal representation of the character desired. These codes may be obtained from page 48 of the RECOMP Manual. It is advisable not to mix codes such as + and . since they would not fall in correlative grid positions (see Fig. 1b and 1c).

2.3 Operational Modes: The plotter may be used to plot data in any of the following modes.

1. Fixed increment I, single D

2. Fixed increment I, D_1 and D_2

3. Variable increment I, single D

4. Variable increment I, D_1 and D_2

2.4 Operational Options: Three options are provided for the user.

2.4.1 Fixed program plotter, Option I.

2.4.1.1 General: In this option, a block of data whose origin is XXXX.0 (see 2.1.1--step 2 above) is plotted with the plotter program operating automatically. Plotting will continue automatically until a value D_1 = -0 (modes 1 and 2) or I = -0 (modes 3 and 4) is encountered. An automatic normal halt at I0000.0 will be obtained when this occurs.[2]

2.4.1.2 Compiler: It is usually laborious to compile the data in ordered form for this mode of operation, so an on-line compiler, using the tape punch, is available for accomplishing this task. Appendix B explains the compiler and its usage.

2.4.1.3 Procedure: (Assumes initialization is complete.)

1. Enter data from compiler or other source.

2. Set I0152.0E.

3. Depress C+00 000X.0+00 000Y.0 (do not enter),

   where $X = \begin{cases} 0 & \text{fixed increment I} \\ 1 & \text{variable increment I} \end{cases}$

   and $Y = \begin{cases} 0 & \text{single D} \\ 1 & D_1 \text{ and } D_2 \end{cases}$

4. Start: Program will plot until an appropriate negative zero is encountered (see Section 2.4.1.1).

[2] The program may be interrupted manually during operations by setting the pre-set stop switch to M0377.1 for purposes of paper change or grid realinement. No reinitialization is necessary after such operations.

2.4.2 Addressable Subroutine, Option II.

2.4.2.1 General: In this option, the main (user's) program addresses the plotter as a subroutine and supplies mode information and data locations. The plotter operates normally for a single cycle but then returns control to the main program. The plotter cannot stop operations in any way during this option.

2.4.2.2 Procedure and Calling Sequence: The calling sequence is as follows:

SLL or SLR

```
FCA      CODE
TRA      0200.0
N.R.     1
N.R.     2
N.R.     3
```

where CODE must be in the main program in the format.

```
CODE    +00 0000.0-00 000Y.0
CODE+1  +00 ZZZZ.0-00 000X.0
```

where X and Y are as in 2.4.1.3 above and ZZZZ.0 is the octal address of the first quantity to be plotted. If X = 0, Y = 0 see Option III below.

If Y $\neq$ 0, ZZZZ.0 will be the address of I. D_1 (and then D_2, if X $\neq$ 0) will follow I consecutively. All variables must be floating point numbers! If Y = 0, ZZZZ.0 will be the address of D_1, and D_2 will follow consecutively.

If Y = 0, normal return #3 will always be used. If Y ≠ 0, then I will be checked against carriage roller position value (abscissa value = AV) and if I is too small for AV, then normal return #1 will be used and no carriage advance will occur. If I is too great for AV, then normal return #2 will be used and the carriage, and AV, will be advanced one increment (ISV).

2.4.3 Addressable Subroutine, Option III:

2.4.3.1 General: This option provides simplification of main program and plotter wherever subroutine usage is desired and X = Y = 0. The main program addresses the plotter as in Option II but supplies only D, which is duly graphed by the plotter. Again, the plotter can not stop operations.

2.4.3.2 Procedure and Calling Sequence: The calling sequence is as follows:

```
SLL or SLR
FCA    D
TRA    0175.0
N.R.
```

The "FCA D" instruction may be replaced by any sequence of operations resulting in D being in the A and R registers in normalized floating point form.

2.4.4 General Comment: At this point, the user might note that by combination of Options I and II, as many as four variables may be plotted from any one given main program operation. The user might plot two variables using Option II and compile two more using a complier routine. By reinitialization and use of Option I, the other two variables may be plotted.

The adroit programmer may conceive many more ways to utilize the plotter. For example, six or more variables could be plotted from a single main program operation by use of special compilers. Option II may be used in place of Option I by use of a short program to sort the data and address the plotter. Whenever re-tracing a graph to add variables, and the abcissa (I) scale has not changed, it is advisable to suppress abcissa scale outputs by entering PI as a large number. If the I scale has changed, some attention should be given to the PI values used so that the scales are not superimposed upon one another when typed out.

The user may question the apparent difficulty associated with obtaining typed output data while the plotter is operating. This may still be done by skillful format management. Remember that the carriage is not returned after plotting a point(s), so that the user may confine his graph to the left side of the paper and, by tabbing and spacing, move the carriage to a position useable for numerical output. The plotter leaves the carriage positioned after the last point plotted, and this point may be anywhere on the graph. If Y = 0, then PC_1 will denote the typewriter position. If PC_1 is negative, the carriage has not moved (is at position 0). These numbers (PC_1 and PC_2) are located in 0033.0 and 0034.0 at a binary scale of 39.

If $Y \neq 0$, then the carriage location can be found from the larger of PC_1 and PC_2, If PC_1 and PC_2 are both negative, then the carriage is at position zero. The user can tab or space until his format location is reached. (Use caution when tabbing!) Tabs may be counted from

carriage position and B. Output routines used should not type any carriage returns. If N.R. #1 from Option II is used, the carriage has not been returned.

3. Method: The actual method used is variable depending on the mode setting and option used. Two important facets are common, however.

3.1 Scaling: The floating point D_i is adjusted to the range stated (D_{iR}) and rounded to the nearest space. It is then converted, by division by Δ_i and shifting, to a fixed point integer at binary 39. If the number lies outside the region bounded by 0 and B, it is set to -1 and the plotter ignores the point. Otherwise, the fixed number becomes the carriage position at which the correct character will be typed.

3.2 Output: By tabbing and spacing the correct location is reached and the appropriate character typed. The program is in control of the carriage position at all times. The program will not tab into the maximum ordinate value since no tab setting exists mechanically when $B > 95$.

3.3 Independent Variable I: When I varies by a fixed increment, the carriage return before output automatically "updates" I and the plotter is not concerned with the actual I. When I varies by a variable increment, however, the I obtained from the main program or data block is checked for the condition $AV-\frac{LSV}{2} \leq I \leq AV+\frac{LSV}{2}$. If the condition is satisfied, the dependent variables are plotted. If $I < AV-\frac{LSV}{2}$, the carriage is not advanced and no plotting occurs. If $I > AV+\frac{LSV}{2}$ the carriage is advanced and no plotting occurs.

3.4 Conclusion: The remainder of the program is concerned with mode and option management and compensating mechanical eccentricities of the typewriter. For a detailed understanding consult the attached block diagram and coding form (Appendix C).

Appendix A

GLOSSARY

A	Accumulator or A register
AV	Abcissa scale value
AV_i	Abcissa scale value (initial)
B	Number of allotted ordinate spaces
CODE i(i = 1,2)	Character code for 1st or 2nd D
D_i, D_i' (i = 1,2)	ith dependent variable
* $D_{CD} = B/N_T$	Spaces per tab
Δ_i (i = 1,2)	$\frac{D_i R}{B}$
D_{iLL}	Smallest plottable ordinate value of ith dependent variable
D_{iR}	Ordinate range ($D_{iUL} - D_{iLL}$) of ith dependent variable
I	Independent variable
Int.	Integer used by abcissa scale output with fixed increment I
L	Abcissa scale significant figures to left of decimal point
LSV	Line scale value - Abcissa increment per carriage return
N_T	Number of tabs
PC_i	Position counter. Numberical carriage position of plot point
PC_i'	General notation for PC_i
PC'	Present numerical carriage position (ordinate)
PI	Abcissa scale print interval
PI'	Abcissa scale print interval counter
R	Abcissa scale significant figures to right of decimal point
R_R	Remainder or R register
$S_i = D_{iLL} + \frac{D_i}{(i = 1,2)^2}$	Scaler for ith dependent variable
SF	Scale Factor for abcissa scale output with I increment fixed

T,Y,Z	Temporary storage
T_T	Tab tally. Counts used tabs.
X_P	Number of carriage spaces occupied by abcissa output

* Must be an integer

Appendix B

COMPILER

The relocatable compiler routine has been written for use with the main program of the user to allow on-line compilation of computed data. The compiler is addressed as a subroutine and given data locations and mode information.

It will punch a paper tape having XXXX.0 as an origin location. All succeeding data are entered consecutively. After completion of the main program (in that portion using compilers) the routine is addressed slightly differently and enters the -0 necessary to stop the plotter in Option I. If this last step is not performed, a single location must be manually set to +0 to allow further use of the compiler.

B.1 Compiler I: General compiler, floating point.

B.1.1 Calling Sequence:

```
SLL or SLR
CLA       CODE
TRA       COMPILER I
N.R.
```

Where CODE is in the main program in the form

CODE + 00 BBBB.0 + 00 000A.0

Where BBBB.0 is octal location of first data word (data in consecutive locations) and A is number of data words to be output; i.e., X + Y + 1 (see Section 2.4.1.3) in the usual case.

For final output, use calling sequence

```
CLS       CODE
TRA       COMPILER I
N.R.
```

Appendix B

B.1.2 Relocation: Relocation may be accomplished by the following procedure:

1. Enter the tape. The location counter will be set to 0035.0_8.
2. Enter C + CLA 0000.0 - CLA XXXX.0E.
3. Set L 7724.0_8E.
4. C + 00 0000.0 - 00 DDDO.0E, where DDDO.0 is the octal address of the relocated subroutine.
5. Start. Normal halt at L 7725.0.

B.1.3 Reuse of Compiler: If the final output calling sequence is used, the compiler may be restarted without further change. If not, set

L 0007.0E

N + 0 E.

The compiler is now reset. This operation ensures a location code being punched on the tape.

Program: Floating Point, Data Plotter Block Diagram

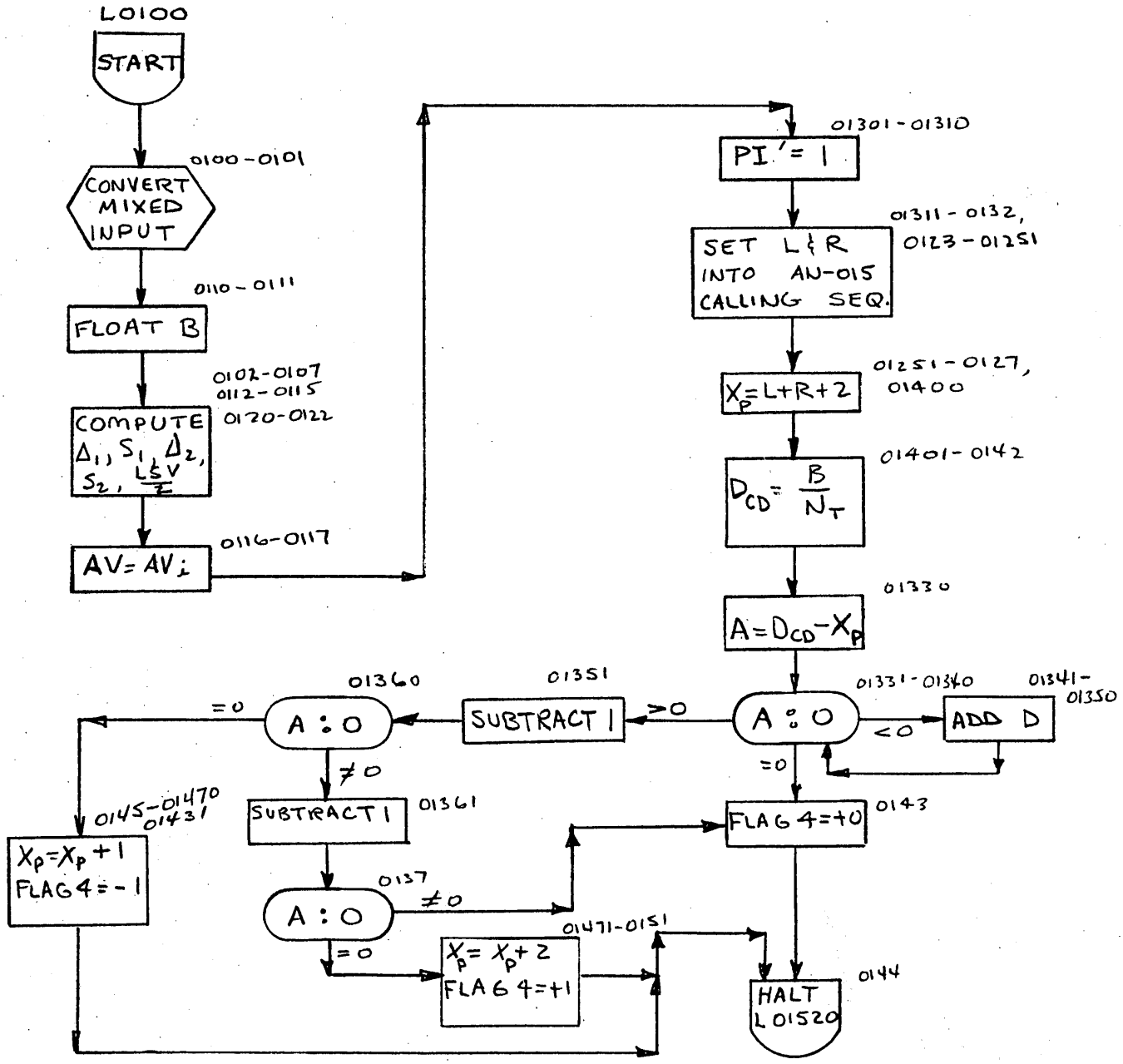
Flag Meaning:

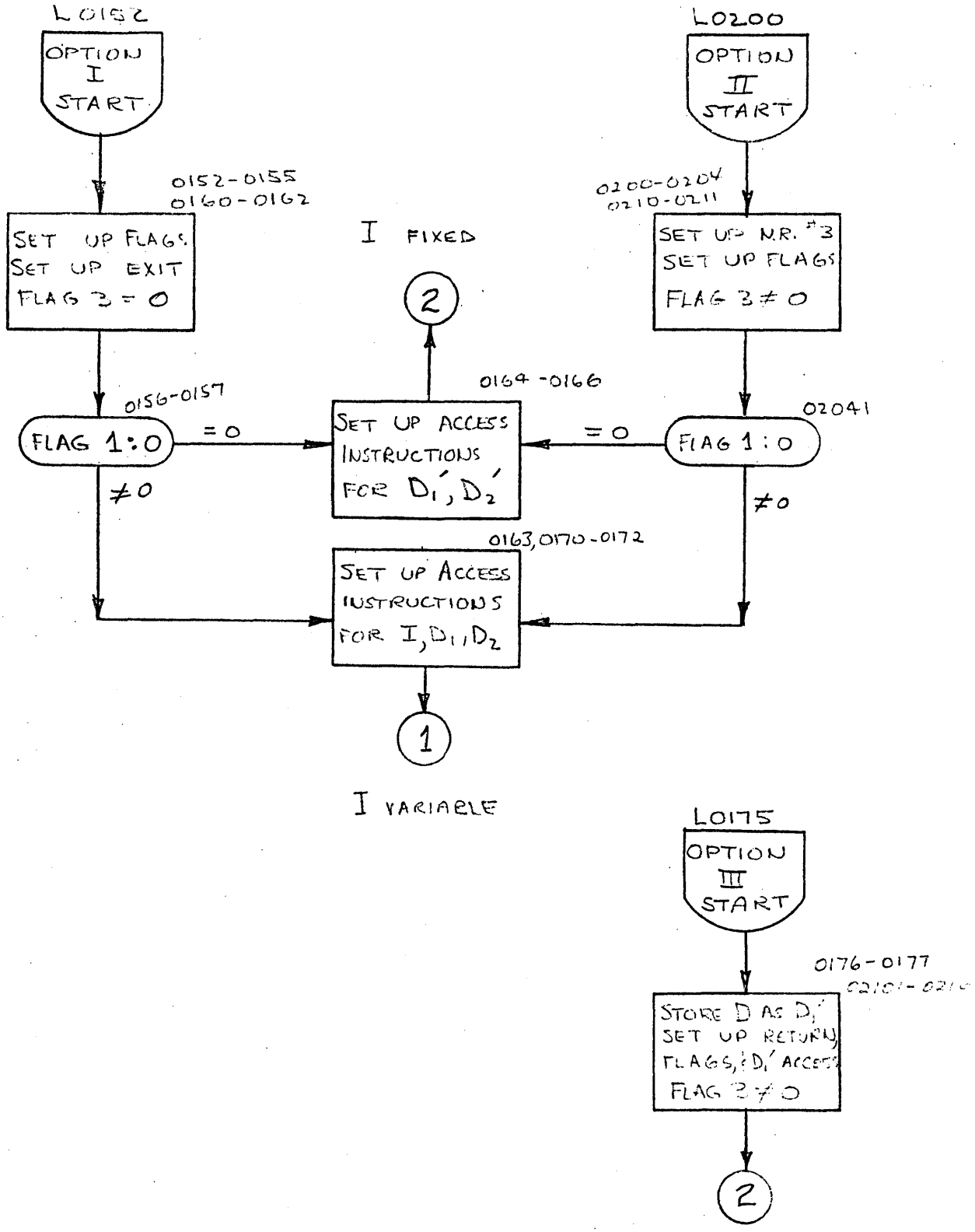
Flag 1 = X { = 0 fixed increment I; = 1 variable increment I

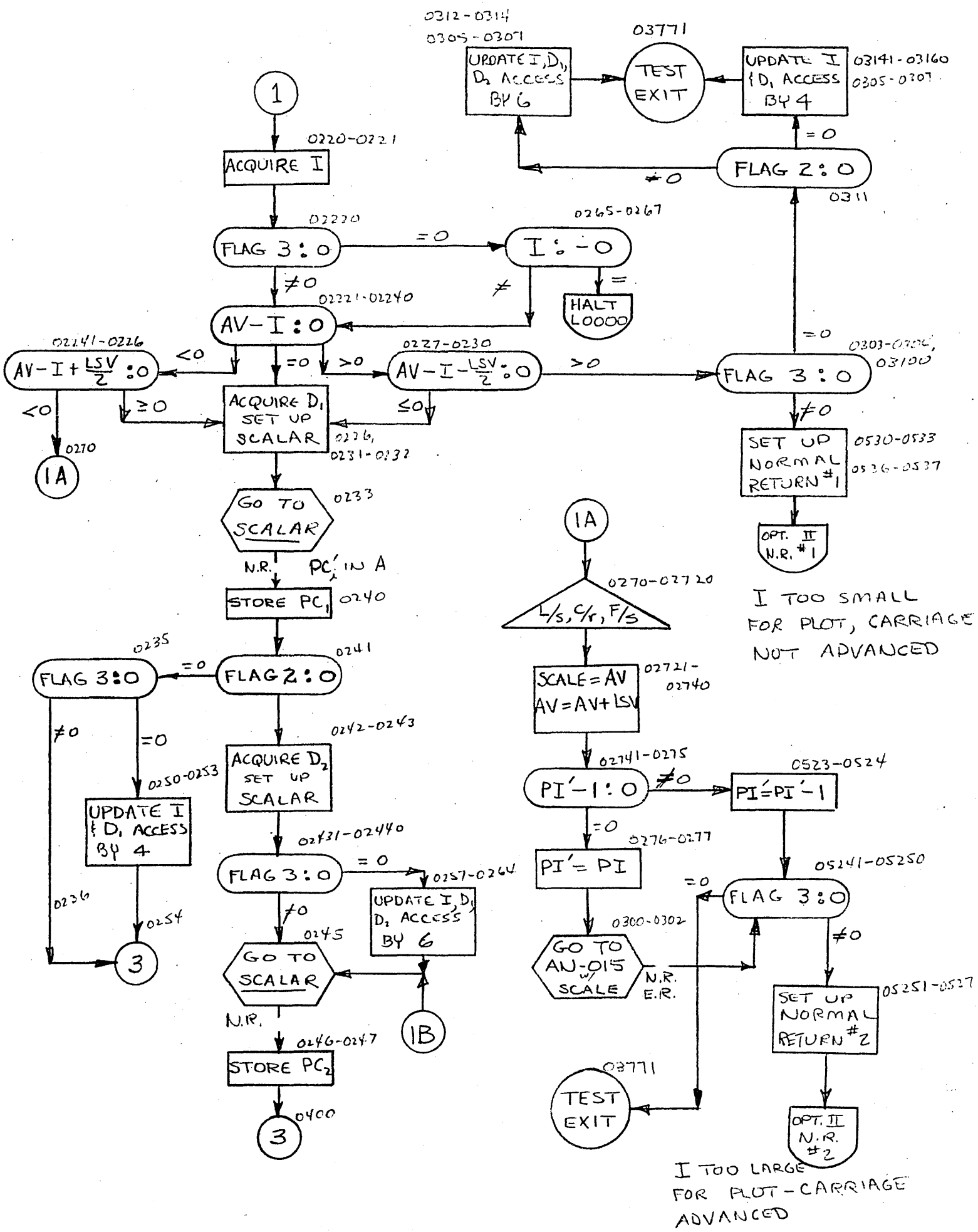
Flag 2 = Y { = 0 single D; = 1 D_1 and D_2

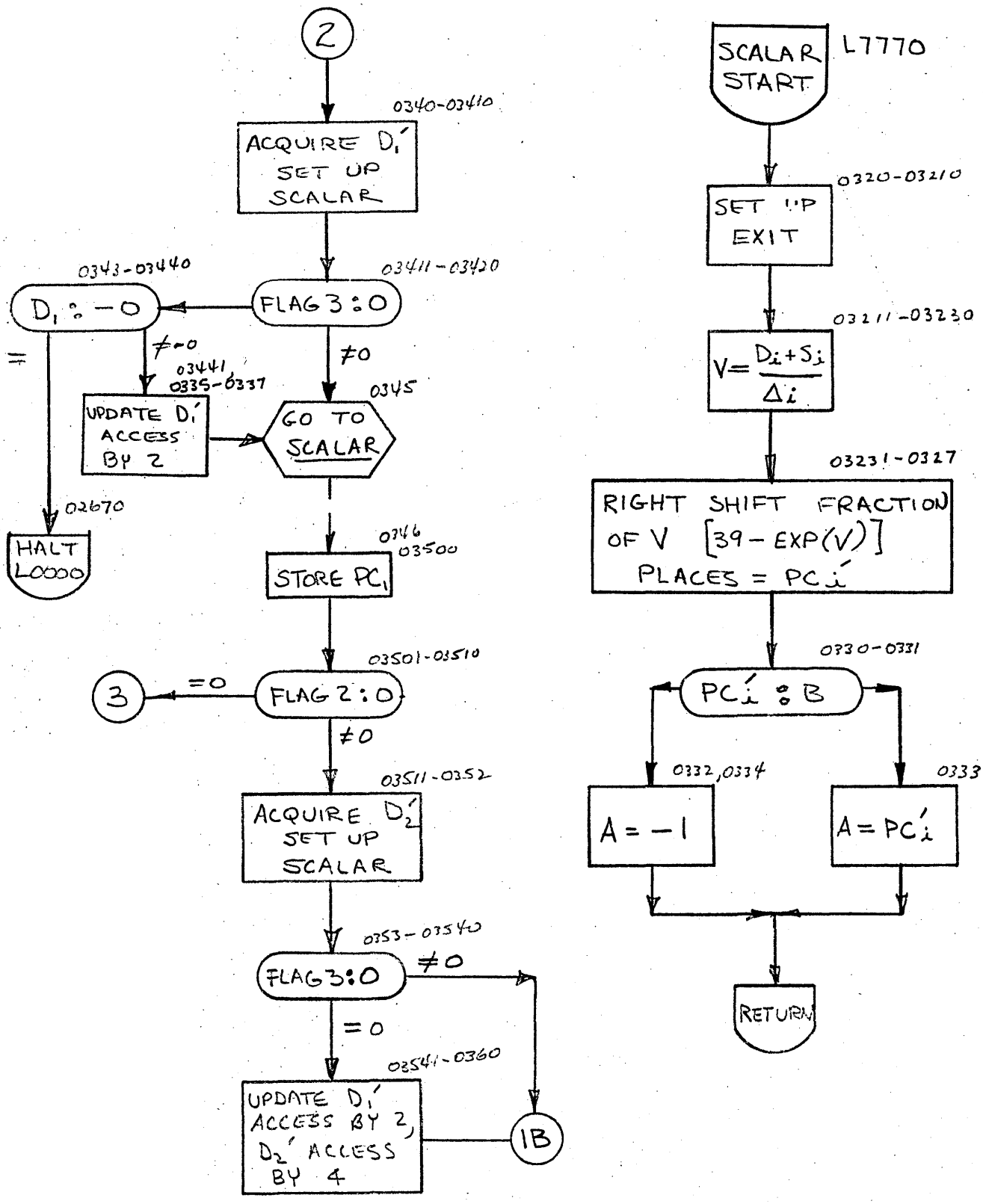
Flag 3 { = 0 fixed program; ≠ 0 subroutine

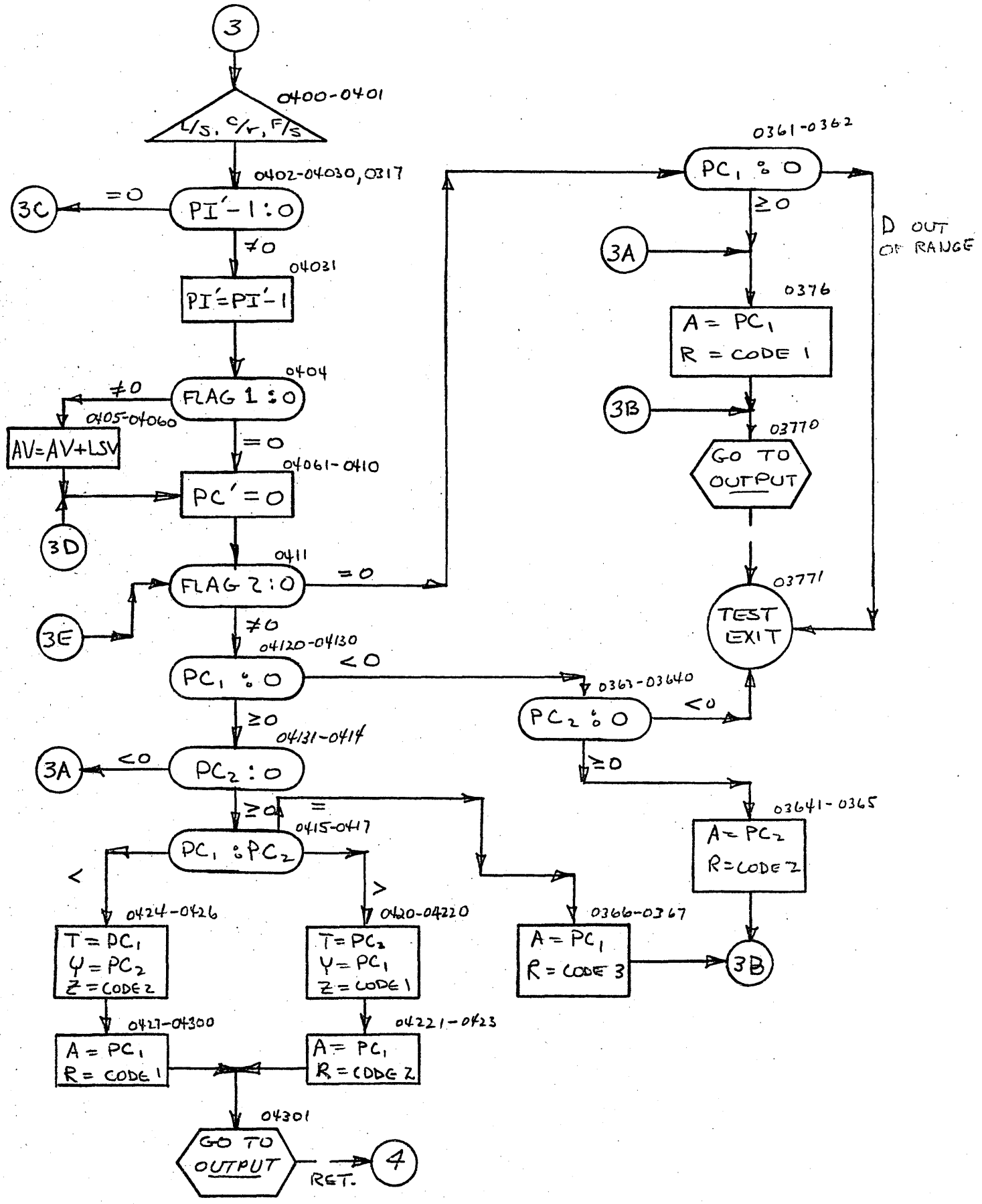
"Initialize"

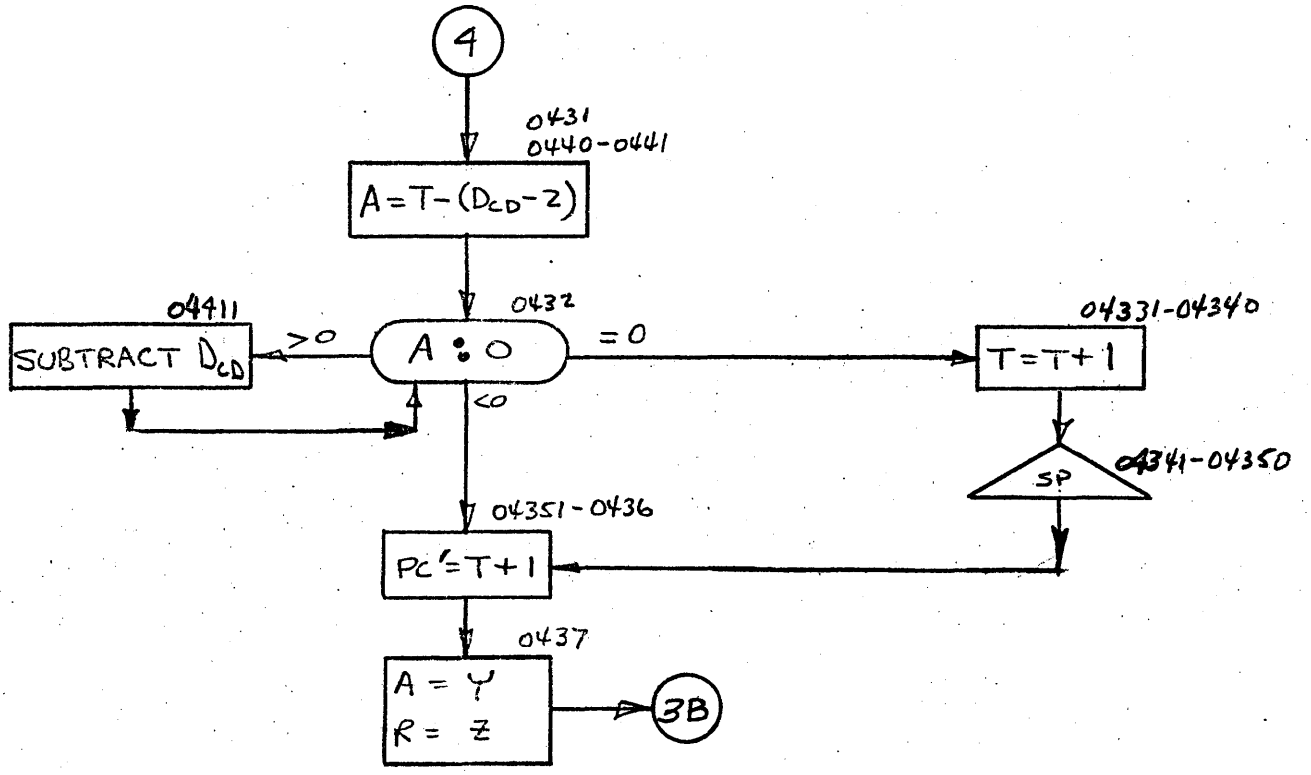


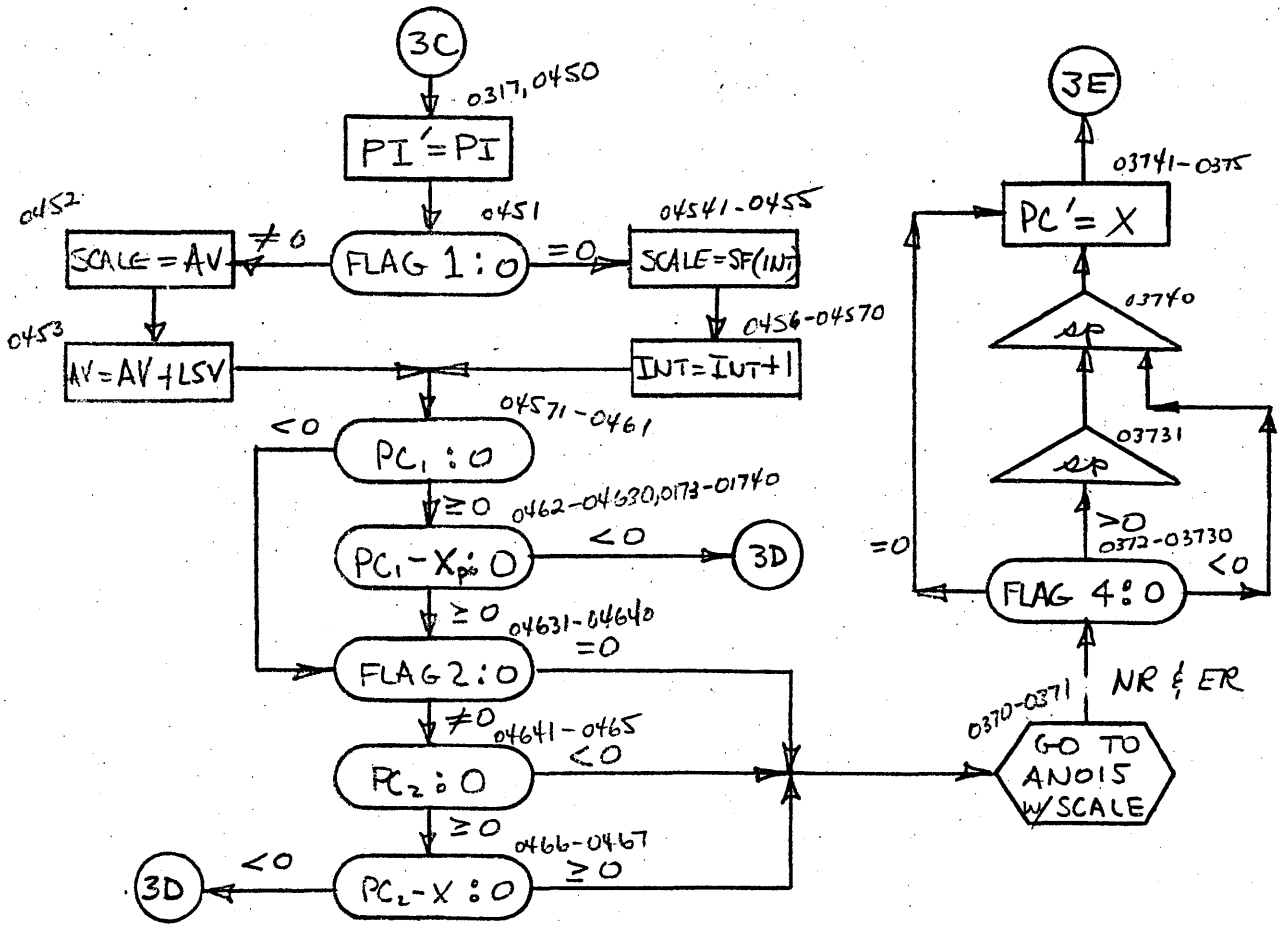

L0152
OPTION I START
0152-0155
0160-0162
SET UP FLAGS
SET UP EXIT
FLAG 3 = 0
0156-0157
FLAG 1:0
=0
≠0
I FIXED
2
0164-0166
SET UP ACCESS INSTRUCTIONS FOR D1', D2'
0163, 0170-0172
SET UP ACCESS INSTRUCTIONS FOR I, D1, D2
1
I VARIABLE
L0200
OPTION II START
0200-0204
0210-0211
SET UP M.R. #3
SET UP FLAGS
FLAG 3 ≠ 0
02041
FLAG 1:0
=0
≠0
L0175
OPTION III START
0176-0177
STORE D AS D1'
SET UP RETURN
FLAGS, & D1' ACCESS
FLAG 3 ≠ 0
2

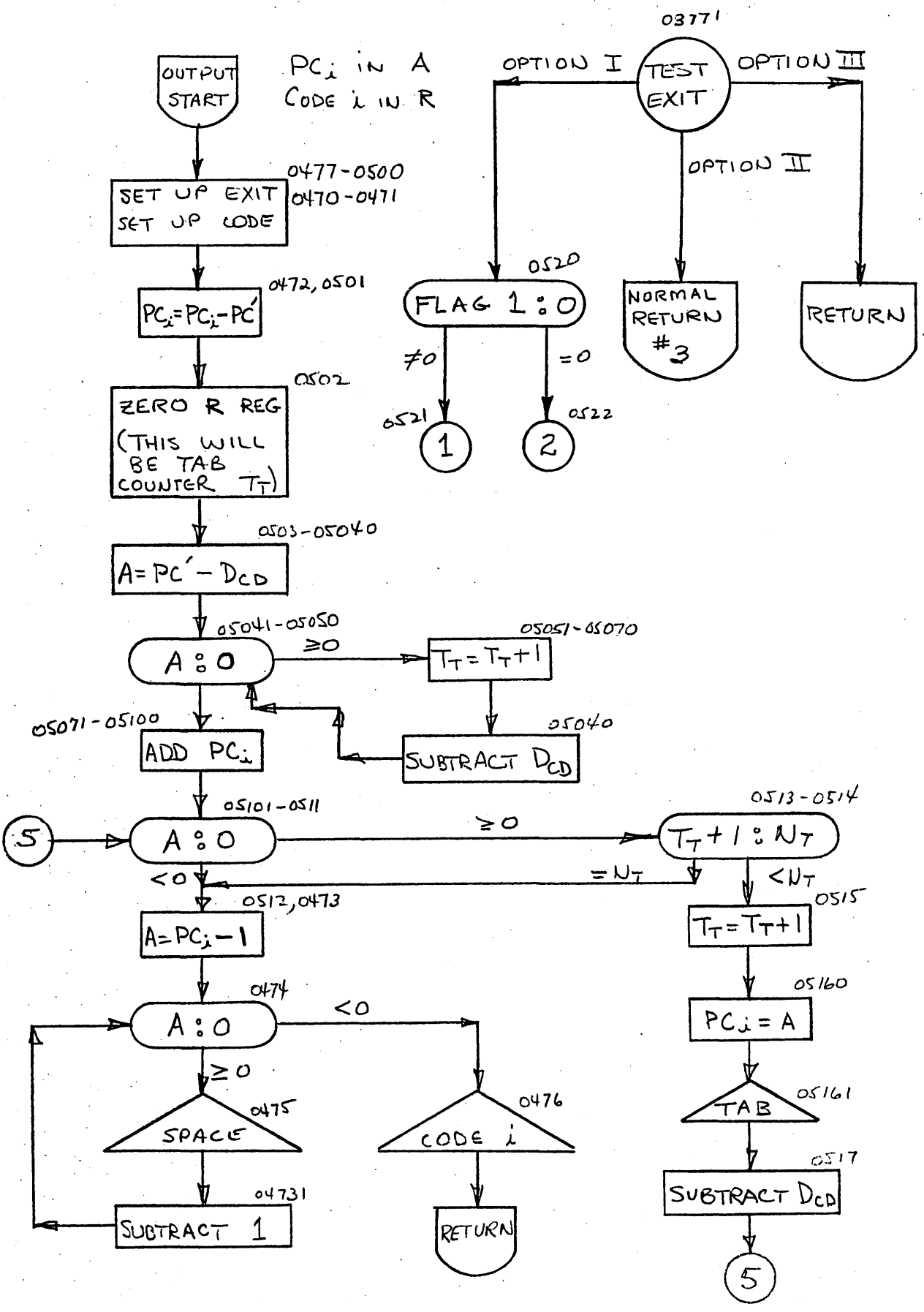

0312-0314
0305-0307
03771
UPDATE I, D1, D2 ACCESS BY 6
TEST EXIT
UPDATE I & D1 ACCESS BY 4
03141-03160
0305-0307
1
0220-0221
ACQUIRE I
=0
FLAG 2:0
0311
≠0
02220
0265-0267
FLAG 3:0
=0
I:-0
=
HALT L0000
≠0
≠
02221-02240
AV-I:0
02241-0226
AV-I+LSV/2 :0
<0
=0
>0
0227-0230
AV-I-LSV/2 :0
>0
=0
0303-0306, 03100
FLAG 3:0
<0
≥0
≤0
ACQUIRE D1 SET UP SCALAR
0226, 0231-0232
≠0
SET UP NORMAL RETURN #1
0530-0533
0536-0537
0270
1A
OPT. II N.R. #1
GO TO SCALAR
0233
N.R.
PC1 IN A
STORE PC1
0240
I TOO SMALL FOR PLOT, CARRIAGE NOT ADVANCED
1A
0270-02720
L/S, C/R, F/S
0235
FLAG 3:0
=0
FLAG 2:0
0241
SCALE=AV
AV=AV+LSV
02721-02740
≠0
=0
0242-0243
ACQUIRE D2 SET UP SCALAR
02741-0275
PI'-1:0
≠0
0523-0524
PI=PI'-1
0250-0253
UPDATE I & D1 ACCESS BY 4
=0
0276-0277
PI'=PI
05241-05250
FLAG 3:0
02431-02440
FLAG 3:0
=0
0257-0264
UPDATE I, D1, D2 ACCESS BY 6
0300-0302
GO TO AN-015 w/ SCALE
N.R.
E.R.
0236
0254
≠0
0245
GO TO SCALAR
N.R.
1B
0246-0247
STORE PC2
03771
TEST EXIT
SET UP NORMAL RETURN #2
05251-0527
3
0400
OPT. II N.R. #2
I TOO LARGE FOR PLOT - CARRIAGE ADVANCED


2
0340-03410
ACQUIRE D1'
SET UP
SCALAR
0343-03440
03411-03420
D1 :: -0
FLAG 3 : 0
=
≠-0
≠0
03441,
0335-0337
0345
UPDATE D1'
ACCESS
BY 2
GO TO
SCALAR
02670
HALT
L0000
0346
03500
STORE PC1
03501-03510
3
=0
FLAG 2 : 0
≠0
03511-0352
ACQUIRE D2'
SET UP
SCALAR
0353-03540
FLAG 3 : 0
≠0
=0
03541-0360
UPDATE D1'
ACCESS BY 2,
D2' ACCESS
BY 4
1B
SCALAR
START
L7770
0320-03210
SET UP
EXIT
03211-03230
V = (Di + Si) / Δi
03231-0327
RIGHT SHIFT FRACTION
OF V [39 - EXP(V)]
PLACES = PCi'
0330-0331
PCi' : B
0332,0334
A = -1
0333
A = PCi'
RETURN


3
0400-0401
L/S, C/R, F/S
0402-04030, 0317
PI'-1:0
=0
3C
≠0
04031
PI'=PI'-1
0404
FLAG 1:0
≠0
0405-04060
AV=AV+LSV
=0
04061-0410
PC'=0
3D
0411
FLAG 2:0
=0
3E
≠0
04120-04130
PC1 : 0
<0
≥0
04131-0414
PC2 : 0
<0
3A
≥0
=
0415-0417
PC1 : PC2
<
>
0424-0426
T=PC1
Y=PC2
Z=CODE 2
0427-04300
A=PC1
R=CODE 1
0420-04220
T=PC2
Y=PC1
Z=CODE 1
04221-0423
A=PC1
R=CODE 2
04301
GO TO OUTPUT
RET.
4
0361-0362
PC1 : 0
≥0
D OUT OF RANGE
3A
0376
A=PC1
R=CODE 1
3B
03770
GO TO OUTPUT
03771
TEST EXIT
0363-03640
PC2 : 0
<0
≥0
03641-0365
A=PC2
R=CODE 2
0366-0367
A=PC1
R=CODE 3
3B


4
0431
0440-0441
A=T-(DCD-2)
04411
SUBTRACT DCD
>0
0432
A : 0
=0
04331-04340
T=T+1
<0
04341-04350
SP
04351-0436
PC'=T+1
0437
A = Y
R = Z
3B


3C
0317, 0450
PI'=PI
0452
SCALE=AV
≠0
0451
FLAG 1 : 0
=0
04541-0455
SCALE=SF(INT)
0453
AV=AV+LSV
0456-04570
INT=INT+1
04571-0461
<0
PC1 : 0
≥0
0462-04630, 0173-01740
PC1-Xp : 0
<0
3D
≥0
04631-04640
=0
FLAG 2 : 0
≠0
04641-0465
<0
PC2 : 0
≥0
0466-0467
≥0
PC2-X : 0
<0
3D
3E
03741-0375
PC'=X
03740
SP
03731
SP
>0
0372-03730
<0
FLAG 4 : 0
=0
0370-0371
NR & ER
GO TO ANO15 W/SCALE


OUTPUT START
PC_i IN A
CODE i IN R
0477-0500
0470-0471
SET UP EXIT
SET UP CODE
0472, 0501
PC_i = PC_i - PC'
0502
ZERO R REG
(THIS WILL BE TAB COUNTER T_T)
0503-05040
A = PC' - D_CD
05041-05050
A : 0
≥0
05051-05070
T_T = T_T + 1
05071-05100
ADD PC_i
05040
SUBTRACT D_CD
05101-0511
5
A : 0
<0
0512, 0473
A = PC_i - 1
0474
A : 0
<0
≥0
0475
SPACE
04731
SUBTRACT 1
0476
CODE i
RETURN
03771
TEST EXIT
OPTION I
OPTION II
OPTION III
0520
FLAG 1 : 0
≠0
=0
0521
1
0522
2
NORMAL RETURN #3
RETURN
0513-0514
T_T + 1 : N_T
=N_T
<N_T
0515
T_T = T_T + 1
05160
PC_i = A
05161
TAB
0517
SUBTRACT D_CD
5