

S

24



Programming Manual

SOAP II

for the IBM 650 Data Processing System

**SOAP II — SYMBOLIC OPTIMAL ASSEMBLY PROGRAM
FOR THE IBM 650 DATA PROCESSING SYSTEM**

PART I: PROGRAMMER'S GUIDE

	<u>Page</u>
Introduction	1
Definition of Terms	1
Coding Form and Input Card Format	2
Specifying Program Area	2
Types of Addresses	4
Operation Codes	5
Numerical Data	6
Alphabetic Data	6
Signs	7
Comments Card	7
Example I	7
Example II	12
Indexing	12
Predefining Symbols	13
Assembly Capacity and Speed	14
Punching Availability Table	14
Order of Assembly Deck	15
Multifile Assembly	15
Machine Operator's Guide	15

APPENDIX

Heading	17
800X Instructions	18
Use of Library Programs	18
Writing Library Programs	19
Symbolic 650 Operation Codes	24
Summary of Pseudo Operations	29
Description of Output Cards	30
533 Control Panel	32
407 Control Panel	33

PART II: PROGRAM ANALYSIS

Introduction	34
Memory Allocation	34
Table Descriptions	34
Input Block	40
Output Block	41
Basic Flow Chart	42
Pseudo Operation Flow Charts	43
Relocate Flow Charts	45
Subroutines	47
Symbolic Listing	65
Availability Table	91
Seven-Words Per Card Listing	92

PART I: PROGRAMMER'S GUIDE

Introduction

The principal achievement of an assembly program is that it almost completely relieves the programmer of the problem of assigning actual storage locations to instructions or quantities manipulated by the program. If, in the course of programming, he wishes to refer to any location in storage which will contain some quantity used by the program, he gives this location a "name", preferably one of high mnemonic value, and refers to it by means of this name. Thus, salary might be stored in "WAGES", sin x in "SIN X". The assembly program assigns actual locations to these "names" and produces an absolute machine language program.

SOAP II is a symbolic assembly program for the IBM 650 Data Processing System. It is designed to assemble programs written for any array of equipment including tapes, printer, immediate access storage, indexing registers, floating point, disk storage and inquiry stations. The assembly program itself uses none of these features. It will operate on a basic 650 having an alphabetic device. Special features described in the sequel should render it equally useful for both scientific and commercial applications.

Definition of Terms

L = location

D = data (address)

I = instruction (address)

Tag = character used to specify an indexing register

FWA = first word address

LWA = last word address

OP = operation (code)

Block = a consecutive group of drum (or core) locations

Region = a block designated by an alphabetic character

Pseudo-instruction = an instruction which is never executed

Symbolizer part = left-most position of L, OP, D or I field

Absolute part = L, OP, D or I field with symbolizer part deleted

Available = subject to selection as an optimum drum location whenever the assembly program is given freedom of choice

Unavailable = not subject to the above selection

Reserve = to make unavailable

I. A. S. = Immediate Access (core) Storage

SOAP II Coding Form and Input Card Format

Figure 1 shows the coding form to be used when writing for SOAP II. Vertical dotted lines separate the symbolizer and absolute parts of the L, OP, D and I fields. Observe that the input card format is indicated at the top of the form. Punching in columns 1-40 and 73-80 is ignored except for column 2 which must not contain a Y(12) punch.

Instructions and data enter the assembly one word per card. The output is in the form of one word per card load cards.

Specifying Program Area on Drum

Prior to assembly, the entire drum is available to the program. It is invariably necessary, however, to prevent the program from occupying certain drum locations, e.g., input-output blocks, tables, data areas, etc. There are three pseudo operations which can be used to restrict the assembled program to any predesignated parts of the drum:

Block Reservation: BLR

This card contains two absolute drum addresses FWA, LWA ($FWA \leq LWA$) punched in the absolute part of the D and I fields, respectively. All locations from FWA to LWA (inclusive) are made unavailable.

For example,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS		TAG	INSTRUCTION ADDRESS		TAG	REMARKS					
					BLR	1500			1900								

will make drum locations (1500-1900) unavailable.

Block Availability: BLA

This card is exactly like a BLR card except that it makes all locations from FWA to LWA available. Note that it is generally redundant to put a BLA card at the beginning of an assembly since the entire drum is already available.

By way of illustration,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS		TAG	INSTRUCTION ADDRESS		TAG	REMARKS					
					BLA	1600			1730								

650 SOAP II CODING FORM

COL.	41	42	43	44-	-47	48	49	,50	51	52-	-55	56	57	58-	-61	62	63-	-72
T	I	S	T	S	LOCATION	OPERATION	DATA	T	INSTRUCTION	T	ACCUMULATOR	UPPER	LOWER	8003	8002			
T	P	E	N			CODE	ADDRESS	A	ADDRESS	G								
PROBLEM	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	REMARKS	_____	_____	_____	_____	_____	_____	_____
CODER	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	DATE	_____	_____	_____	_____	_____	_____	_____
PAGE	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	OF	_____	_____	_____	_____	_____	_____	_____

will make drum locations 1600-1730 available.

Regional Specification: REG

This card contains an alphabetic character, punched in the symbolizer part of the D field, to be associated with a block defined by the terminal drum addresses FWA, LWA. As in the case of a BLR card, all locations from FWA to LWA are made unavailable. If the programmer wishes to refer to locations within a region using "regional" addresses (see below), an REG card must precede the first use of such addresses.

The REG card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN		LOCATION			OPERATION CODE		DATA ADDRESS		T	INSTRUCTION ADDRESS	T				REMARKS	
						R	E	G	P	0027		0036					

defines the block 0027-0036 as region P and causes this block to be made unavailable.

Types of Addresses

Regional

A regional address has the form

| A |NNNN|

where "A" is any alphabetic character and NNNN is a four digit number. The regional address of the first word of region B, for example, is B0001. The twelfth word of region R is R0012.

Note:

1. The regional address "A"0000 is permitted. It will result in the absolute address (FWA-1).
2. Regional addresses whose absolute equivalents are larger than LWA are permitted. For example, if region R is the block 1951-1960, the regional address R0015 will assemble as 1965.
3. The use of a regional address without a defining REG card will result in — the assembled address being left blank.

Symbolic

Symbolic addresses have the form

| C₅ | C₄ C₃ C₂ C₁ |

where the C's may be any characters acceptable to the alphabetic or special

character devices subject to the following restrictions:

1. C₅ must not be blank.
2. An address meant to be symbolic must not have a regional form, i.e., an alphabetic character followed by four numerical characters.

Typical symbolic addresses are

X		N	ET
X	Y	G	ROSS
2	GS	T	AXES
L	OOP	B	ONDS
F	INIS	5	
R	EAD	F	IVE

Symbolic addresses are assigned optimum drum equivalents when initially encountered and maintain this equivalence throughout the assembly unless redefined. It is possible to preassign a value to a symbol (see EQU, SYN, pages 13, 14.)

Absolute

Absolute addresses have the form

 | NNNN |

where NNNN is a valid four digit machine address. The symbolizer part of an absolute address must be blank.

Drum locations corresponding to absolute addresses are not made unavailable during assembly. Thus, drum locations containing instructions or data specified by absolute addresses must be block reserved at the beginning of assembly.

Blank

Whenever the D or I address of an instruction refers to the location of the "next" instruction (or data) written on the coding form, this address and the location of the "next" instruction may be left blank provided this order is not altered when assembling. Blank addresses will be filled in optimally by the assembly program.

The D and I address of an instruction may both be blank if they both refer to the location of the next instruction written. An address should not be left blank in an instruction card unless the above meaning is specifically intended.

Operation Codes

650 operation codes may be written in either three character symbolic or two digit numerical form, e. g.,

R	AL
	65

Symbolic operation codes to be used with SOAP II are given in the Appendix.

Any card containing an illegal symbolic op code will be considered an instruction and processed accordingly. The assembled op will be left blank.

Numerical Data

Numerical data are written using the absolute part of the OP, D and I fields, e.g.,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE	DATA ADDRESS			T A G	INSTRUCTION ADDRESS			T A G	REMARKS				
		PI				31	4159		2654								
		ONE				00	0001		0000								
		FICA				00	0000		4200								

The symbolizer parts of these fields must be blank. Leading or trailing zeros must be punched in the card.

Alphabetic Data: ALF

The pseudo-op card ALF permits the programmer to enter alphabetic information into the 650 on load cards, up to five characters per card. Any characters acceptable to the alphabetic or special character devices are permissible, and are written in the D field. During assembly, alphabetic data are converted to their numerical equivalents and the latter are punched as a ten digit number in exactly the same manner as numerical data. For example,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE	DATA ADDRESS			T A G	INSTRUCTION ADDRESS			T A G	REMARKS				
		EDPM ALF IBMNY															

will cause the number [69, 62, 74, 75, 88] to be loaded into symbolic location EDPM.

Similarly,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE	DATA ADDRESS			T A G	INSTRUCTION ADDRESS			T A G	REMARKS				
		9010 ALF 705															

will load $[00, 97, 90, 95, 00]$ into (core) location 9010.

Note: The I address of an ALF card is not used by SOAP II. However, if it is blank, it will be filled in by the symbol "SOAP 2". This is merely an artifice employed to avoid erroneous error detection when listing output (See Appendix-407 Control Panel).

Signs

Any punch in column 42 signifies a negative instruction or negative data. For coding purposes, a (-) written in the sign column and punched as X (11) will suffice. A blank in column 42 is interpreted as positive.

Comments Card: Type 1

Every card processed by SOAP II may contain remarks not exceeding 10 characters in length. When longer remarks are desired, a comments card, which may contain up to 30 characters, may be used. Comments cards have no effect on the assembly. They are merely reproduced in the output deck. The following figure shows the format for comments cards. Note that the type (1) is punched in column 41.

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN		LOCATION		OPERATION CODE		DATA ADDRESS	T	INSTRUCTION ADDRESS	A	G		T	A	G		REMARKS
I		\leftarrow	C		O		M	M	E		N	T	S		\rightarrow		

Example.I

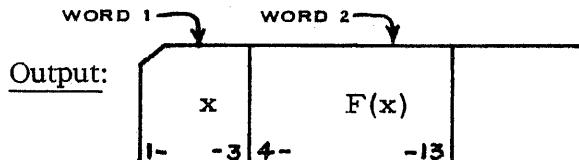
Prepare a table of

$$F(x) = Ax^2 + Bx + C$$

for

$$x = 1, 2, \dots, 100$$

assuming A, B, C integers and $|F(x)| < 10^{10}$.



Coding is given in Figure 2. The assembled program is shown in Figure 3.

$$F(x) = A_x + \beta x + C = (A_x + \beta) x + C$$

650 SOAP II CODING FORM

COL.	41	42	43	44-	-47	48	49	50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	S	LOCATION	OPERATION CODE		DATA ADDRESS	T	INSTRUCTION	A	G		REMARKS							
1	E	X AMP	LIE		1b CAL	C	ULATE	F	OF	X								DISTRIBUTOR 8001
1	BLR	1951		1960	READ AREA													
1	REG P	0027		0028	PUNCH AREA													
1	SET X	RAU	ONE	STX	SET X													
1	STX	STU	P0001			TO	I											
1	M PY	A				CALCULATE												
1	A LO	B				F												
1	R AU	8002					AX+B											
1	M PY	P0001						(AX+B)X										
1	A LO	C							(AX+B)X+C									
1	S TL	P0002																
1	P CH	P0001																
1	R AU	P0001																
1	S UP X	MAX																
1	N ZU				9999													
1	A UP	I 01		STX	STEP X													
1	ONE	00	0000															
1	X MAX	00	0000															
1	I 01	00	0000															

1 1
2 1
3 BLR 1951 1960 READ AREA

4 REG P0027 0028 PUNCH AREA

5 1

6 SETX RAU ONE STX SET X 0000 60 0003 0007
7 STX STU P0001 TO 1 0007 21 0027 0030
8 MPY A CALCULATE 0030 19 0033 0004
9 AL0 B F 0004 15 0057 0011
10 RAU 8002 0011 60 8002 0019
11 MPY P0001 0019 19 0027 0048
12 AL0 C 0048 15 0001 0005
13 STL P0002 0005 20 0028 0031
14 PCH P0001 PUNCH 0031 71 0027 0040
15 RAU P0001 IS X MAX 0040 60 0027 0081
16 SUP XMAX 0081 11 0034 0039
17 NZU 9999 0039 44 0043 9999
18 AUP 101 STX STEP X 0043 10 0046 0007
19 1
20 ONE 00 0000 0001 CONSTANTS 0003 00 0000 0001
21 XMAX 00 0000 0100 0034 00 0000 0100
22 101 00 0000 0101 0046 00 0000 0101

Figure 3 : Assembly of Example 1

650 SOAP II CODING FORM

COL. 41	42	43	44-	-47	48	49	50	51	52-	-55	56	57	58-	-61	62	63-	ACCUMULATOR		DISTRIBUTOR 8001
																	UPPER 8003	LOWER 8002	
-	E	X	AMP	L	E	b	CAL	C	ULATE		GROSS	PAY							
-																			
-	BLR	1951	1960	READ	AREA														
-	REG	P0027	0030	PUNCH	AREA														
-	BLR	0000	0009	RATE	TABLE														
-																			
1	READ	RCD	1950	READ	CARD														
1	LDD	1951		STORE															
1	STD	P0001		IDENT															
1	RAL	1952		GET															
1	STD	P0002		HOURLY															
1	ALO	MASK	8002	RATE															
1	8002	RAU	0009	CG															
CG	MPY	1953		CALCULATE															
CG	STD	P0003		GROSS															
CG	SRD	0001		ROUND															
CG	STL	P0004																	
CG	TPCH	P0001	READ	PUNCH															
CG	MASK	RAU	0000	CG	CONSTANT														
CG																			

1 1 EXAMPLE 2 CALCULATE GROSS PAY

2 1
3 BLR 1951 1960 READ AREA
4 REG P0027 0030 PUNCH AREA
5 BLR 0000 0009 RATE TABLE
6 1
7 READ RCD 1950 READ CARD 0050 70 1950 0013
8 LDD 1951 STORE 0013 69 1951 0054
9 STD P0001 IDENT 0054 24 0027 0080
10 RAL 1952 GET 0080 65 1952 0057
11 STD P0002 HOURLY 0057 24 0028 0031
12 ALO MASK 8002 RATE 0031 15 0034 8002
13 8002 RAU 0009 CG 8002 60 0009 0063
14 CG MPY 1953 CALCULATE 0063 19 1953 0024
15 STD P0003 GROSS PAY 0024 24 0029 0032
16 SRD 0001 ROUND 0032 31 0001 0039
17 STL P0004 PUNCH 0039 20 0030 0033
18 PCH P0001 READ 0033 71 0027 0050
19 1
20 MASK RAU 0000 CG CONSTANT 0034 60 0000 0063

Figure 5 : Assembly of Example 2

Example II

Given a rate code and number of hours worked, search a rate table and calculate gross pay.

Input:

IDENT.	RATE CODE	HOURS	
I - -5	X 6	XX-X	
		7- -9	

WORD 1 WORD 2 WORD 3

Word 1: 00000, xxxxx = Identification

Word 2: 00,000x,0000 = Rate Code

Word 3: 00 00000 xx.x = Hours

The rate code is a one digit number 0-9. The corresponding hourly rate (x.xx) is in the low three positions of locations 0000-0009.

Output:

WORD 1	WORD 2	WORD 3	WORD 4
↓	↓	↓	↓
IDENT.	RATE CODE	HOURS	GROSS
I - -5	X 6	XX-X	XXX-XX
		7- -9	10- -14

Coding is shown in Figure 4. The assembled program is given in Figure 5.

Indexing

When the D or I address of an instruction is to be modified by an indexing register, the corresponding TAG column is used to indicate the appropriate indexing register. Permissible tags are A, B, C or 1, 2, 3. The assembly program will automatically add the proper multiple of 2,000 or 200 to a tagged address.

For example, the following program will perform the vector addition $x_i + y_i = z_i$, $i = 1, 2, \dots, 100$. The x_i are stored in region X, y_i in region Y and z_i in region Z.

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS		TAG	INSTRUCTION ADDRESS		TAG	REMARKS					
					R SA	0100	A DD										
					ADD	R AL X	0101	A									
						A LO Y	0101	A									
						S TL Z	0101	A									
						A X A	0001										
						N Z A A	DD										

The following types of D or I addresses will not be indexed even if a tag is present:

1. Undefined symbolic
2. Blank
3. Not in the range 0000-1999 or 9000-9059.

Predefining Symbols

The following two pseudo operations may be used to preassign a value to a symbolic address:

Equivalence: EQU

The symbol written in the D address is assigned the equivalent of the expression written in the I address. The I address may be absolute, regional or symbolic. If the I address is regional or symbolic, it must have been previously defined.

The EQU card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS			TAG	INSTRUCTION ADDRESS			TAG	REMARKS			
		EQU TAPE 1				8010											

will cause the symbol TAPE 1 to be assigned the value 8010 whenever it subsequently appears during assembly.

The EQU card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS			TAG	INSTRUCTION ADDRESS			TAG	REMARKS			
		EQU CODE				R0002											

will cause the symbol CODE to be assigned the value R0002, i. e., the second word of region R. Region R must have been previously defined by an REG card.

Similarly, the EQU card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS			TAG	INSTRUCTION ADDRESS			TAG	REMARKS			
		EQU ALPHA				BETA											

will assign to ALPHA the value previously assigned to BETA.

Synonym: SYN

An SYN card is exactly like an EQU card except that the equivalent of the expression written in the I address must be a drum address, i.e., 0000 ≤ 'I' ≤ 1999. This drum location is made unavailable to the program. The I address may be absolute, regional or symbolic.

The SYN card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	S	I	G	LOCATION	OPERATION CODE		DATA ADDRESS	T	A	G	INSTRUCTION ADDRESS	T	A	G		REMARKS	
					SYN	ENTRY					1205						

will assign the value 1205 to the symbol ENTRY and reserve location 1205.

If the I address of an EQU or SYN card is undefined, the card will be ignored. EQU or SYN takes precedence over any previous definition.

Assembly Capacity and Speed

During assembly, all symbolic addresses and their equivalents are stored in a symbol table which can accommodate 400 symbols. Programs containing more than 400 symbolic addresses should be assembled in separate blocks. Non-overlap can be guaranteed by use of an availability punchout described in the next section.

Assembly progresses at the rate of 50 to 90 cards per minute. This will decrease somewhat if the symbol table becomes densely packed or if few drum locations are available for the remaining program.

Punching Availability Table: PAT

The availability table is a 200 word table used by SOAP II to "remember" those drum locations already used and those still remaining. The pseudo-op PAT can be used at any time during assembly to cause punching of this table in a highly readable form. The output will be fifty Type Y (12) cards, one for each dynamic drum level (00-49). These cards may be listed with the 407 control panel described in the Appendix. This listing will reveal at a glance all drum locations used by the program (0) and those remaining (1) for additions or corrections.*

The availability table may be loaded as input to an assembly thus restoring the

*See page 91.

availability status which existed at some point of a prior assembly.

Order of Assembly Deck

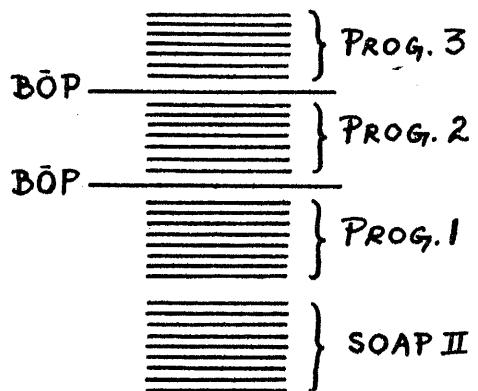
The assembly input deck should have the following order:

1. SOAP II
2. Deck to be assembled

Due to the one pass nature of the assembly, priority for the choice of optimal locations diminishes as the assembly progresses. Thus, frequently executed portions of the program should be placed toward the beginning of the assembly deck.

Multifile Assembly: BOP

Any number of independent programs may be assembled in one pass. Merely place a pseudo-op card BOP (Beginning of Program) between each program, i. e.,



Note that it is redundant to place a BOP card before the first of one or more programs to be assembled.

Machine Operator's Guide

1. 533 Read-Punch Unit
 - a. Insert SOAP II control panel
 - b. Ready read feed with assembly deck
 - c. Ready punch feed with blanks
2. 650 Console
 - a. Set programmed switch to STOP
 - b. Set half cycle switch to RUN

- c. Set control switch to RUN
 - d. Set display switch to DISTRIBUTOR
 - e. Set overflow switch to SENSE
 - f. Set error switch to STOP
- 3A. If SOAP II is being loaded, set (70, 1951, 9999) in storage entry switches.
- 3B. If SOAP II is already on the drum, do one of the following:
1. Set (00, 0000, 1000) in storage entry switches.
 2. Precede input with a BOP card and set (00, 0000, 1950) in storage entry switches.
4. Press computer-reset key
 5. Press program-start key
 6. When read hopper empties, press end of file key
 7. The availability table may be obtained manually by sending control to location 1900.

Programmed Stops

0111-Symbol table full; pressing program-start key will continue assembly.
Symbolic address which cannot be assigned an equivalent will be left blank in the output.

0222-Drum packed, i.e., no locations available for remaining program.
Pressing program-start key will continue assembly. Addresses which cannot be assigned will be left blank in the output.

There are no other programmed stops.

Program Decks

Copies of the condensed SOAP II program deck corresponding to the listing shown on pages 92-94 of this manual are available from

650 Program Librarian
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

APPENDIX

Heading: HED

Heading is used to avoid duplicity of symbols when several programs or several sections of a single program are to be assembled together. The need for heading is paramount if several persons have contributed to a program or when a program employs symbolic library routines.

Heading is accomplished in SOAP II by the automatic insertion of a heading character into the right-most position of symbols having this position blank. The heading character is punched in the symbolizer part of the D address of a HED card. Thus the symbol

| T | W Ø b b | (b = blank)

subsequent to the heading card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE	DATA ADDRESS		TAG	INSTRUCTION ADDRESS		TAG	REMARKS						
		H E D Z															

is processed by the assembly program as if it were actually the symbol

| T | W Ø b Z |

In order to make effective use of heading, it is necessary to agree before hand to restrict all symbols to four characters or less.* Prior to assembly, each of the various sections of the program is preceded by an appropriate HED card.

If within a section of the program headed by "A" it is desired to refer to the symbol (TAX bb) appearing in a section headed by F, simply use the symbol (TAX bF). The small b, for blank column, is necessary to prevent the card punch operator from closing up the symbol and thus taking the "F" out of the heading column.

*If a symbol is to be referred to by several headed parts of the program, then it is best that this symbol not be headed anywhere. This can be accomplished by making it a five-character symbol.

The effect of a heading card is terminated only by another heading card. If it is desired to terminate a heading card without introducing a new heading, use a HED card with a blank D address. Any character acceptable to the alphabetic or special character devices may be used for heading.

800X Instructions

When the D or I address of an instruction located in 800X (X = 0, 1, 2, 3, 5, 6, 7) is to be optimized by the assembly program, an instruction located in 800X should be written on the coding form. 800X instructions will be automatically bypassed when loading the assembled output.

Use of Library Programs

Within the framework of SOAP II, library programs are in either symbolic or relocatable form. Both of these forms are discussed in the following two sections.

Symbolic Library Programs

Symbolic library programs are written in exactly the same form as the main program. Except for the entry and exit symbols, all symbols are four characters or less. They may thus be conveniently headed. It is important to note that such programs do not incorporate a "guaranteed" optimization.

Relocatable Library Programs: REL

Whenever tight optimization is required, one should use relocatable library programs. These are coded in absolute and may be translated an amount specified by the programmer. This is accomplished by use of the pseudo operation REL (relocate). The number punched in the absolute D address specifies the amount of drum translation, Δ_D . Similarly, the absolute I address specifies the amount of core translation, Δ_C . A blank field is interpreted as zero. Thus

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
T Y P E	S I G N	LOCATION			OPERATION CODE	DATA ADDRESS			T A G	INSTRUCTION ADDRESS			T A G	REMARKS			
					REL		1200				0010						

would cause drum addresses in the relocatable program following it to be incremented by 1200 and core addresses by 10. The REL card

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G	REMARKS					
					REL	0600											

will increment subsequent drum addresses by 600 and core addresses by 0. Only positive translations are permitted.

Relocatable library programs must be assembled prior to the main program. The programmer need not block reserve the drum area used by a relocatable program. This is automatically performed by the assembly program.

The effect of an REL card is terminated only by another REL card.

If a translated drum location is greater than 1999, it is left blank in the output. The same applies to a translated core location greater than 9059.

Writing Library Programs

The following two sections are of primary interest to programmers responsible for writing SOAP II library routines.

Symbolic Library Programs

Entry and exit symbols should be five characters to avoid heading difficulties. All other symbols should be four characters or less to permit heading by the user.

Relocatable Library Programs

Relocatable programs are written in absolute and should use the lowest possible drum or core locations in the unrelocated state. The following rules apply specifically to these routines:

1. A relocatable card is indicated by a 2 punch in the type column (41).
2. Op codes may be symbolic or numerical
3. L may be drum or core.
4. D and I may be absolute or symbolic.*

*Symbolic D or I addresses may occasionally be required to refer to other routines.

5. Absolute addresses may be one of five varieties:

- a. Non-relocatable (Fixed): |F| NNNN|
- b. Drum relocatable: |I| NNNN| ; $0000 \leq NNNN \leq 1999$
- c. Core relocatable: |I| NNNN| ; $9000 \leq NNNN \leq 9059$

where NNNN is a four-digit number. In situations where D or I is relocatable but not in the range (b) or (c), the usage

- d. Drum relocatable: |D| NNNN|
- e. Core relocatable: |C| NNNN|

will eliminate possible ambiguity.

6. Absolute D or I addresses in the range 0000-1999 or 9000-9059 after possible relocation may be tagged for indexing by the assembly program.

7. If L, after possible relocation, is a drum address, it will be made unavailable. D or I drum addresses are not reserved.

Two pseudo operations, RBR and REQ are designed for use with relocatable programs.

Relocatable Block Reservation: RBR

The pseudo operation RBR is identical to BLR except that the drum addresses FWA, LWA are incremented by Δ_D prior to reservation. For example, if $\Delta_D = 1000$,

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN			LOCATION		OPERATION CODE			DATA ADDRESS	T A G		INSTRUCTION ADDRESS	T A G		REMARKS		
				R B R		0025			0050								

will cause drum locations (1025-1050) to be reserved.

The use of RBR is mainly for drum routines which use erasable blocks within the routine. Reservation of these blocks is easily accomplished with RBR cards thus avoiding the necessity of a large number of cards simply to reserve erasable areas.

Relocatable Equivalence: REQ

The pseudo operation REQ enables symbolic linkage of the main program with relocatable programs. REQ is similar to EQU except that the absolute I address is incremented by Δ_C or Δ_D prior to being assigned as the equivalent of the

symbol in the D address. The I address must be absolute, type (b) or (c) (see 5 above). Thus if $\Delta_C = 0020$,

COL.	41	42	43	44-	-47	48 49, 50	51	52-	-55 56	57	58-	-61 62	63-	-72
TYPE	SIGN		LOCATION		OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G		REMARKS
					R E Q	R A M A C			9 0 1 0					

will assign 9030 as the equivalent of the symbol "RAMAC".

Similarly, if $\Delta_D = 1300$,

COL.	41	42	43	44-	-47	48 49, 50	51	52-	-55 56	57	58-	-61 62	63-	-72
TYPE	SIGN		LOCATION		OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G		REMARKS
					R E Q	T Y P E 3			0 0 7 5					

will assign 1375 as the equivalent of the symbol "TYPE 3."

To illustrate a typical drum-drum* relocatable subroutine, the program given in Figure 6, using indexing registers, will evaluate the polynomial

$$F(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

where $n \geq 0$ and x, a_i are floating point numbers. The a_i 's are stored such that

$$L(a_i) = L(a_0) + i$$

The calling sequence is

COL.	41	42	43	44-	-47	48 49, 50	51	52-	-55 56	57	58-	-61 62	63-	-72
TYPE	SIGN		LOCATION		OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G		REMARKS
					R A L	C W								
					L D D	X								
					R A C	N E X T			P O L Y X					

where
 $CW = [00, \{L(a_0) + 2000\}, n]$ if the a_i are on the drum
 $[00, \{L(a_0) + 200\}, n]$ if the a_i are in core

The routine will exit to (drum location) NEXT with the result in the upper accumulator.

*Loaded onto and executed from the drum.

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G	REMARKS					
2		0009 STD		0013		0016			0016			SAVE X					
2		0016 LDD		0019		0011			0001			SET ADDR. OF a_0					
2		0011 SDA		0019		0010			n → I.A. A								
2		0001 RAAF 8002		0010		0019			()·X								
2		0010 FMP		0013		0002			()·X + $a_i \rightarrow ()$								
2		0002 NZA		0005		F 0000 C			IS n = 0								
2		0005 SXAF 0001		0010		0000			n-1 → n								
2		0013 OI F 0000		F 0000		0009			ERASEABLE (X)								
		REQ POLYX							DEFINE ENTRY								

Figure 6: Relocatable Drum-Drum Subroutine

The relocatable drum-core* subroutine given in Figure 7 will evaluate $K!$ where $K > 0$ and $K! < 10^{10}$. The calling sequence is

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION		OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G	REMARKS					
		RAL K															
		LDD EXIT		CALKF													

The result will be in the lower accumulator on exiting from the subroutine. Note that whereas seven core locations are required for execution, only six drum locations are needed to store the subroutine. Savings of this kind may usually be effected by placing erasable locations at the end of drum-core routines.

*Loaded onto the drum and executed in core.

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE		DATA ADDRESS		T A G	INSTRUCTION ADDRESS		T A G	REMARKS				
2		0000	S	TD			9006			9001			STORE EXIT				
2		0001	R	AA	F		8002			9002			K → I.A. A				
2		0002	S	X	A	F	0001			9003			(K-1) IN I.A. A				
2		0003	N	Z	A		9004			9006			IS (K-1)=0				
2		0004	R	AU	F		8002			9005			()→UA				
2		0005	M	PY	F		8005			9002			()(K-P)				
			R	EQ	C	ALKF				9000			DEFINE ENTRY				

Figure 7: Relocatable Drum-Core Subroutine

SYMBOLIC 650 OPERATION CODES

Numerical	Symbolic*	Operation
00	NOP	No Operation
01	HLT	Halt
02	UFA	Unnormalized Floating Add
03	RTC	Read Tape Check
04	RTN	Read Tape Numeric
05	RTA	Read Tape Alphanumeric
06	WTN	Write Tape Numeric
07	WTA	Write Tape Alphanumeric
08	LIB	Load I.A.S. Block
09	LDI	Load I.A.S.
10	AUP	Add Upper
11	SUP	Subtract Upper
12	-	(Not Used)
13	-	(Not Used)
14	DIV	Divide
15	ALO	Add Lower
16	SLO	Subtract Lower
17	AML	Add Magnitude to Lower
18	SML	Subtract Magnitude from Lower
19	MPY	Multiply

*All O's in symbolic OP codes are alphabetic O's, not numerical zeros.

Numerical	Symbolic	Operation
20	STL	Store Lower
21	STU	Store Upper
22	SDA	Store Data Address
23	SIA	Store Instruction Address
24	STD	Store Distributor
25	NTS	Branch No Tape Signal
26	BIN	Branch on Inquiry
27	SET	Set I.A.S. Ring
28	SIB	Store I.A.S. Block
29	STI	Store I.A.S.
30	SRT	Shift Right
31	SRD	Shift and Round
32	FAD	Floating Add
33	FSB	Floating Subtract
34	FDV	Floating Divide
35	SLT	Shift Left
36	SCT	Shift Left and Count
37	FAM	Floating Add Magnitude
38	FSM	Floating Subtract Magnitude
39	FMP	Floating Multiply

Numerical	Symbolic	Operation
40	NZA	Branch Non-Zero I.R. A
41	BMA	Branch Minus I.R. A
42	NZB	Branch Non-Zero I.R. B
43	BMB	Branch Minus I.R. B
44	NZU	Branch Non-Zero Upper
45	NZE	Branch Non-Zero
46	BMI	Branch Minus
47	BOV	Branch on Overflow
48	NZC	Branch Non-Zero I.R. C
49	BMC	Branch Minus I.R. C
50	AXA	Add I.R. A
51	SXA	Subtract I.R. A
52	AXB	Add I.R. B
53	SXB	Subtract I.R. B
54	NEF	Branch No End of File
55	RWD	Rewind
56	WTM	Write Tape Mark
57	BST	Backspace Tape
58	AXC	Add I.R. C
59	SXC	Subtract I.R. C

Numerical	Symbolic	Operation
60	RAU	Reset Add Upper
61	RSU	Reset Subtract Upper
62	-	(Not Used)
63	-	(Not Used)
64	DVR	Divide Reset Upper
65	RAL	Reset Add Lower
66	RSL	Reset Subtract Lower
67	RAM	Reset Add Magnitude into Lower
68	RSM	Reset Subtract Magnitude into Lower
69	LDD	Load Distributor
70*	RD1	Read Input Storage 1
71*	WR1	Write Output Storage 1
72	RC1	Read Conditional Input Storage 1
73	RD2	Read Input Storage 2
74	WR2	Write Output Storage 2
75	RC2	Read Conditional Input Storage 2
76	RD3	Read Input Storage 3
77	WR3	Write Output Storage 3
78	RC3	Read Conditional Input Storage 3
79	RPY	Reply

*SOAP II will also accept the symbolic op codes

70 RCD Read Card
 71 PCH Punch

Numerical	Symbolic	Operation
80	RAA	Reset Add I.R. A
81	RSA	Reset Subtract I.R. A
82	RAB	Reset Add I.R. B
83	RSB	Reset Subtract I.R. B
84	TLU	Table Lookup
85	SDS	Seek Disk Storage
86	RDS	Read Disk Storage
87	WDS	Write Disk Storage
88	RAC	Reset Add I.R. C
89	RSC	Reset Subtract I.R. C
90	BDO	Branch on 8 in Distributor 10
91	BD1	Branch on 8 in Distributor 1
92	BD2	"
93	BD3	"
94	BD4	"
95	BD5	"
96	BD6	"
97	BD7	"
98	BD8	"
99	BD9	Branch on 8 in Distributor 9

COL.	41	42	43	44-	-47	48	49, 50	51	52-	-55	56	57	58-	-61	62	63-	-72
TYPE	SIGN	LOCATION			OPERATION CODE	DATA ADDRESS		TAG	INSTRUCTION ADDRESS		TAG	REMARKS					
							B LR	FWA						LWA			
							B LA	FWA						LWA			
							R EG X	FWA						LWA	* = ALPHABETIC CHAR.		
							E QU S	YMBL						ANY			
							S YN S	YMBL						ANY			
							A LF X	XXXXX									
							B OP										
							H ED #							# = HEADING CHAR.			
							P AT										
							R EL	Δ D						Δ C			
							R BR	FWA						LWA			
							R EQ S	YMBL						NNNN			

Summary of Pseudo Operations

Y		CARD NO.	T S Y - LOCATION P G E N			OP	D	I	T A I G REPRODUCED	REMARKS
00,00000,8000	X X X X					INPUT				q 72 73
10	17 20		41 42 43	47	48-50 51	55	56 57	G1 G2 G3		72

Instructions, Data and ALF

Y		CARD NO.	LOCATION	Y	ASSEMBLED {X INSTRUCTION	Y	ASSEMBLED {X INSTRUCTION	Y	INPUT	REPRODUCED
69,1954,1953	X X X X	24, XXXX, 8000	XX, XXXX, XXXX							
10	17 20 21		30 31		40 41					72

800X Instructions

Y		CARD NO.	Y	ASSEMBLED {X INSTRUCTION	Y	ASSEMBLED {X INSTRUCTION	Y	INPUT	REPRODUCED	Y
69,1954,8000	X X X X	24, 800X, 8000	XX, XXXX, XXXX							
10	17 20 21		30 31		40 41					72

Availability Table: $00 \leq \alpha \leq 49$

Y		(X=0 or 1)	Y	Y	Y	Y	Y	Y	Y	Y
00, α , $\alpha+450$	X X X X X X X X	00, $\alpha+500, \alpha+950$	X X X X X X X X	00, $\alpha+1000, \alpha+1450$	X X X X X X X X	00, $\alpha+1500, \alpha+1950$	X X X X X X X X	X X X X X X X X	X X X X X X X X	
10,11	20 21		30 31	40 41	50 51	60 61	70 71	80 81		

DESCRIPTION OF OUTPUT CARDS

407 Control Panel

The 407 control panel shown in Figure 9 will detect the following situations in the assembled output:

Note

- 1 - Instruction * with blank L following an instruction with D and I both not blank.
- 2 - Instruction with L not blank following an instruction with either D or I blank.
- 3 - Instruction with blank L not preceded by an instruction.
- 4 - Non-instruction following an instruction with D or I blank.

Notes (1) and (2) always imply a coding error while (3) and (4) are warning signals to point out potential coding errors.

If the assembled L, OP, D or I fields of an instruction have been left blank, columns 75-78, respectively, will be x (11) punched. Correspondingly, the phrase

BLANK L, OP, D or I

will print to the right of the absolute instruction.

If an EQU or SYN card has been ignored due to an undefined I address or a packed symbol table, column 79 will contain a 9 punch causing the word "BYPAS" to print to the right of the Remarks field.

*Instruction → instruction or data.

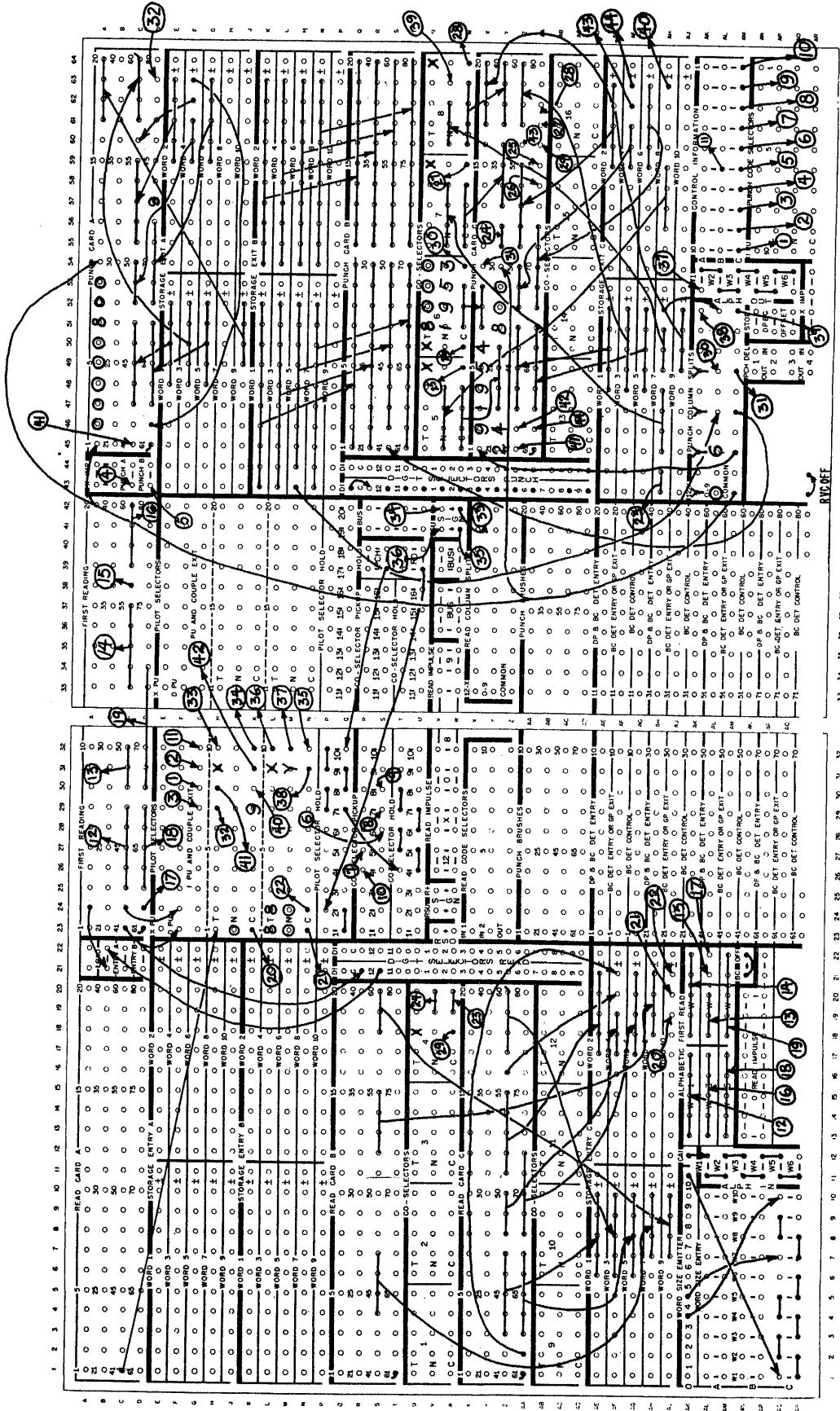
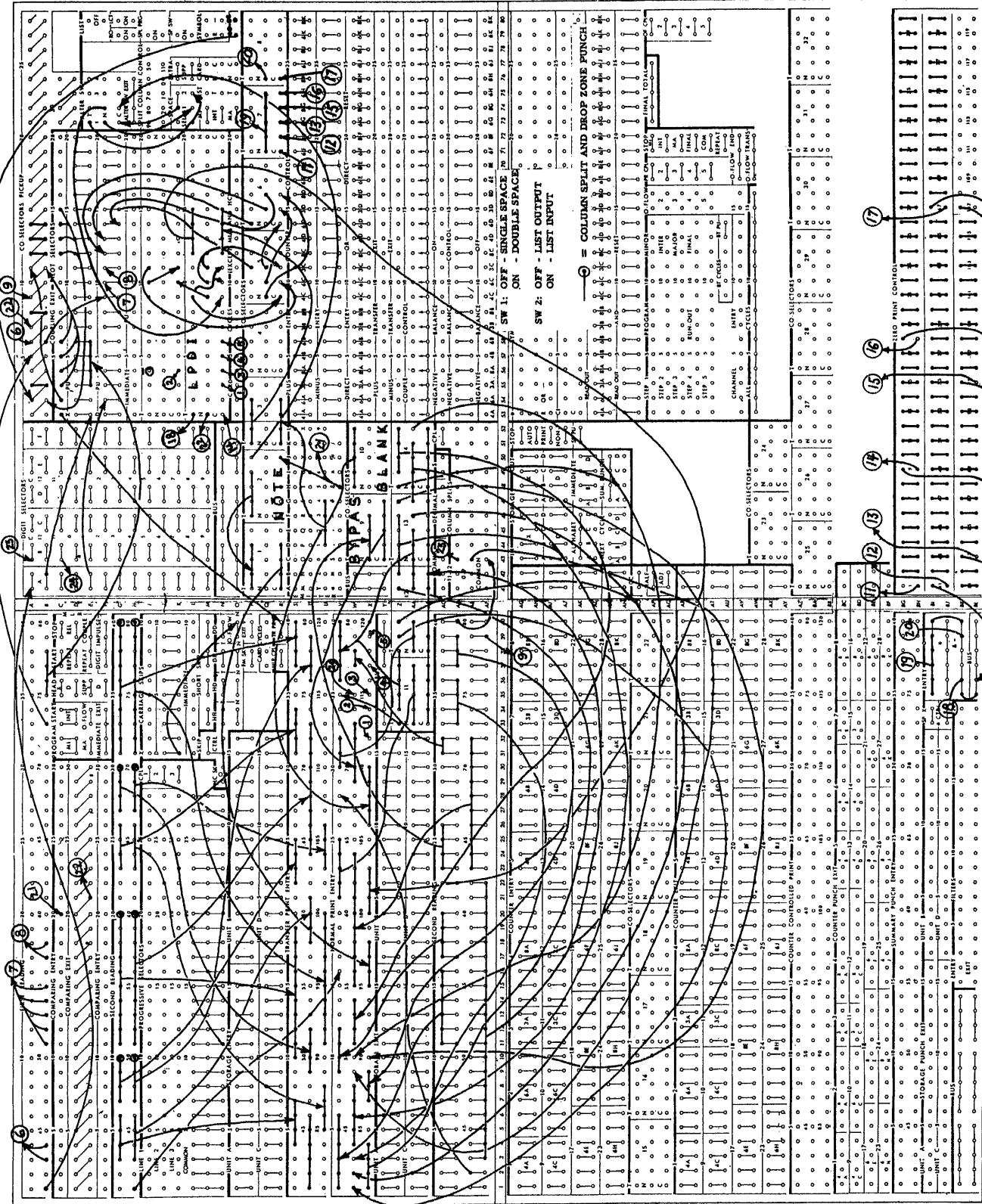


FIGURE 8: SOAP II 533 CONTROL PANEL

FIGURE 9: SOAP II 407 CONTROL PANEL



PART II : PROGRAM ANALYSIS

Introduction

The following is a description of the internal operation of SOAP II. An attempt has been made to present this description in a logical order beginning with allocation of memory followed by an explanation of the tables governing the assembly and outline of the various subroutines.

The symbolic assembly of SOAP II included at the end of this report was arrived at in the following manner:

1. SOAP II was coded symbolically for assembly by SOAP I.
2. Deck (1) was assembled by SOAP I, giving SOAP II in SOAP I form.
3. Deck (1) was translated into SOAP II symbolic form using the SOAP I to SOAP II Translator (N.Y.S.C.C. #334).
4. The symbolic SOAP II form of SOAP II (3) was assembled by Deck (2) giving SOAP II in SOAP II form.

The program is listed in logical order as opposed to the order in which it was assembled.

Memory Allocation

Figure 1 is a "snapshot" of the drum showing the location of the various tables used by the assembly program and the input-output and program blocks.

Table Descriptions

Symbol Table: 0000-0399

Every symbol processed by the assembly program is stored in the symbol table. An attempt is first made to store a symbol in location L_0 computed as follows:

Let $S = XXXXXXXXXX$ be the ten digit numerical representation of a five alphanumerical character symbol. Calculate

$$(1001001001) X (S) = YYYYYYYYYY(ZZZ)YYYYYYY$$

Then L_0 is taken to be the integral part of $.4(ZZZ)$. Thus $000 \leq L_0 \leq 399$. It may be seen that all of the digits in S contribute to the formation of L_0 . Thus different symbols will tend to give rise to different L_0 's, but the problem of duplicity, i.e., different symbols having the same L_0 , must be accounted for. If a symbol cannot be stored in L_0 (because another symbol has "gotten

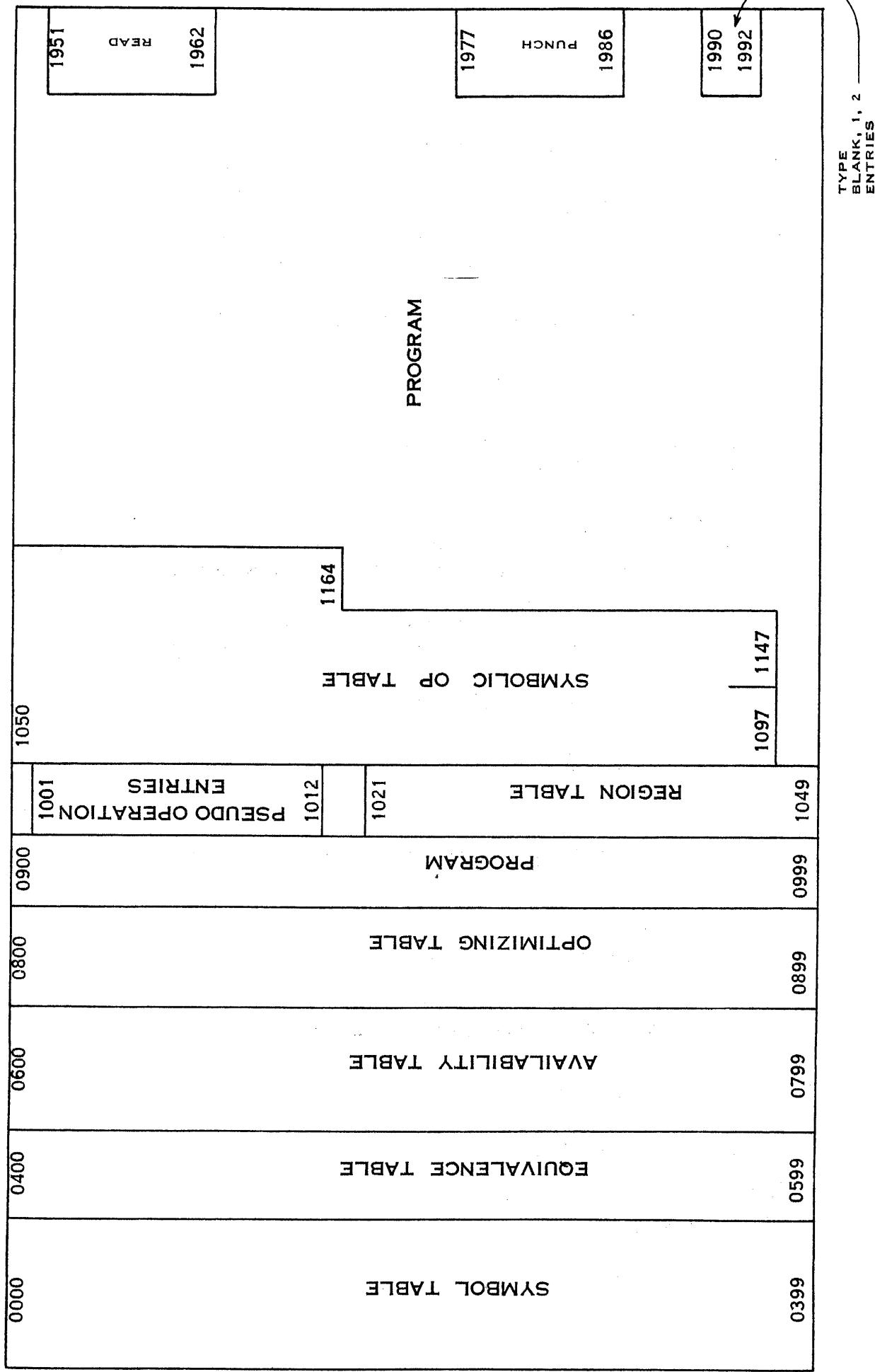


FIGURE 1: MEMORY ALLOCATION

there first"), an attempt is made to store it in $(L_0 + 1)$. If this too fails, we try $L_0 + 2$, $L_0 + 3$, ..., $L_0 + K$ where K is (hopefully) small.

Equivalence Table: 0400-0599

Whenever a symbol is stored in the symbol table, its four digit equivalent is stored in the equivalence table in the following manner:

Let L be the location in the symbol table containing the symbol. Let

$$\frac{L}{2} = Q + \frac{R}{2}$$

Then the equivalent is stored in the data address or instruction address part of $(0400 + Q)$ according as $R = 0$ or 1 .

Availability Table: 0600-0799

The availability table "remembers" drum locations used by the program being assembled and those still remaining. Each of the 200 words contains indicator digits ($0 \rightarrow$ unavailable, $1 \rightarrow$ available) for ten dynamically equivalent drum locations. If these digits are numbered from left to right, $P = 0, 1, 2, \dots, 9$, then the digit corresponding to a drum location L may be found as follows:

$$\text{Let } L = X_1 X_2 X_3 X_4. \text{ Let } 2L = Y_1 Y_2 Y_3 Y_4.$$

Then

$$A = 0600 + Y_1 + 2(Y_3 Y_4)$$

$$P = Y_2$$

where A is the availability table location whose P^{th} digit corresponds to L .

Example:

$$\begin{aligned} L &= 1238 \\ 2L &= 2476 \\ A &= 0600 + 2 + 2(76) = 0754 \\ P &= 4 \end{aligned}$$

Conversely, given A and P , one may compute L as follows:

$$\text{Let } \frac{A - 0600}{4} = Q + \frac{R}{4}$$

$$\text{Then } L = 500R + 50P + Q$$

Example: $A = 0754, P = 4$

$$\frac{0754 - 0600}{4} = \frac{154}{4} = 38 + \frac{2}{4}$$

$$L = 500 \times 2 + 50 \times 4 + 38 = 1238$$

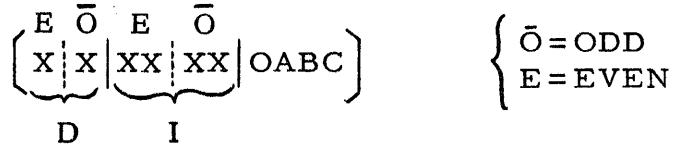
Optimizing Table: 0800-0899

Information required to optimize 650 instructions is stored in the optimizing table. Location (0800 + XX) contains the entry for numerical op code XX where $00 \leq XX \leq 99$. Thus the entry for RAL (65) is in 0865.

As there is no one rule by which all 650 instructions may be optimized, operations having the same or similar optimizing rules have been grouped as shown in Figure 2. The general optimizing rules for these groups are:

- 999: $I = D + S$ where $S = f(D)$
- 998: $I = L + S$ where $S \neq f(D)$
- 898: $I = L + S$ where $S = f(D)$, non-shift
- 988: $I = L + S$ shift other than SRD
- 888: $I = L + S$ SRD

Optimizing table entries have the following format:



where ABC is the group code given in Figure 2. Thus the entry for RAL is (3305040999).

GROUP CODE	OPERATIONS
999	UFA, AUP, SUP, DIV, ALO, SLO, AML, SML, MPY, STL, STU, SDA, SIA, STD, FAD, FSB, FDV, FAM, FSM, FMP, RAU, RSU, DVR, RAL, RSL, RAM, RSM, LDD, LIB, LDI, SIB, STI, TLU, RD1, WR1, RC1, RD2, WR2, RC2, RD3, WR3, RC3, RCD, PCH, RPY
998	NOP, HLT, RTC, RTN, RTA, WTN, WTA, NTS, BIN, SET, NZA, BMA, NZB, BMB, NZC, BMC, NZU, NZE, BMI, BOV, NEF, RWD, WTM, BST, BDO, BD1, , BD9, SDS, RDS, WDS
898	AXA, SXA, AXB, SXB, AXC, SXC, RAA, RSA, RAB, RSB, RAC, RSC
988	SRT, SLT, SCT
888	SRD

Figure 2: 650 Operation Groups

Pseudo Operation Entries: 1001-1012

The twelve pseudo operations have been (arbitrarily) numbered as follows:

#	Pseudo Operation
1	BOP
2	BLR
3	BLA
4	REG
5	EQU
6	SYN
7	ALF
8	PAT
9	HED
10	REL
11	REQ
12	RBR

The first instruction of a pseudo operation is in location (1000 + #). Thus the REL routine begins in 1010.

Region Table: 1021-1049

Origins of the twenty-six possible regions are stored in the region table. The instruction address part of location (0960 + XX) contains the first word address (FWA) of the region having XX as the numerical representation of its alphabetic designation. Thus the origin of region F is in 1026 (= 0960 + 66).

The region table is initialized to (8000000000). Processing of an REG card with FWA = YYYY converts the appropriate entry to (900000YYYY).

Symbolic Op Table: 1050-1164 except 1098-1099 and 1148-1149

All pseudo operation codes and symbolic 650 operation codes are stored in the symbolic op table. The table is in ascending order on the six high digits of each entry representing the numerical representation of the alphanumerical op code.

The table entry for RAL, for example, is (7961730065) while that for REL is (7965731010). Thus the I address part of pseudo operation entries is the location of the first instruction of the pseudo routine while the two low order digits of 650 symbolic op code entries are the corresponding numerical op.

Type Entries: 1990-1992

The first instruction for types "blank", 1 and 2 is in location 1990, 1991, 1992 respectively.

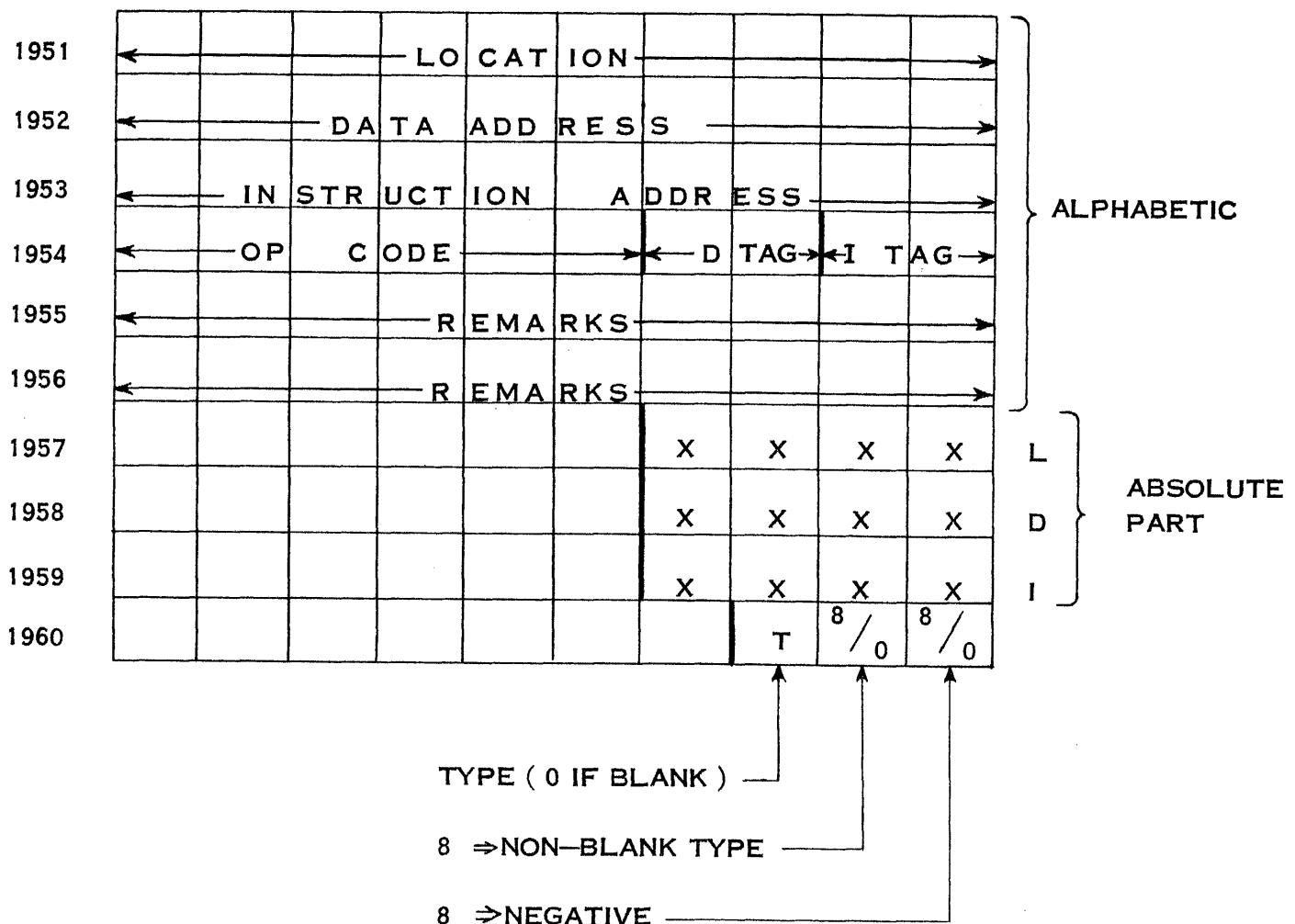


FIGURE 3: INPUT BLOCK : 1951—1960

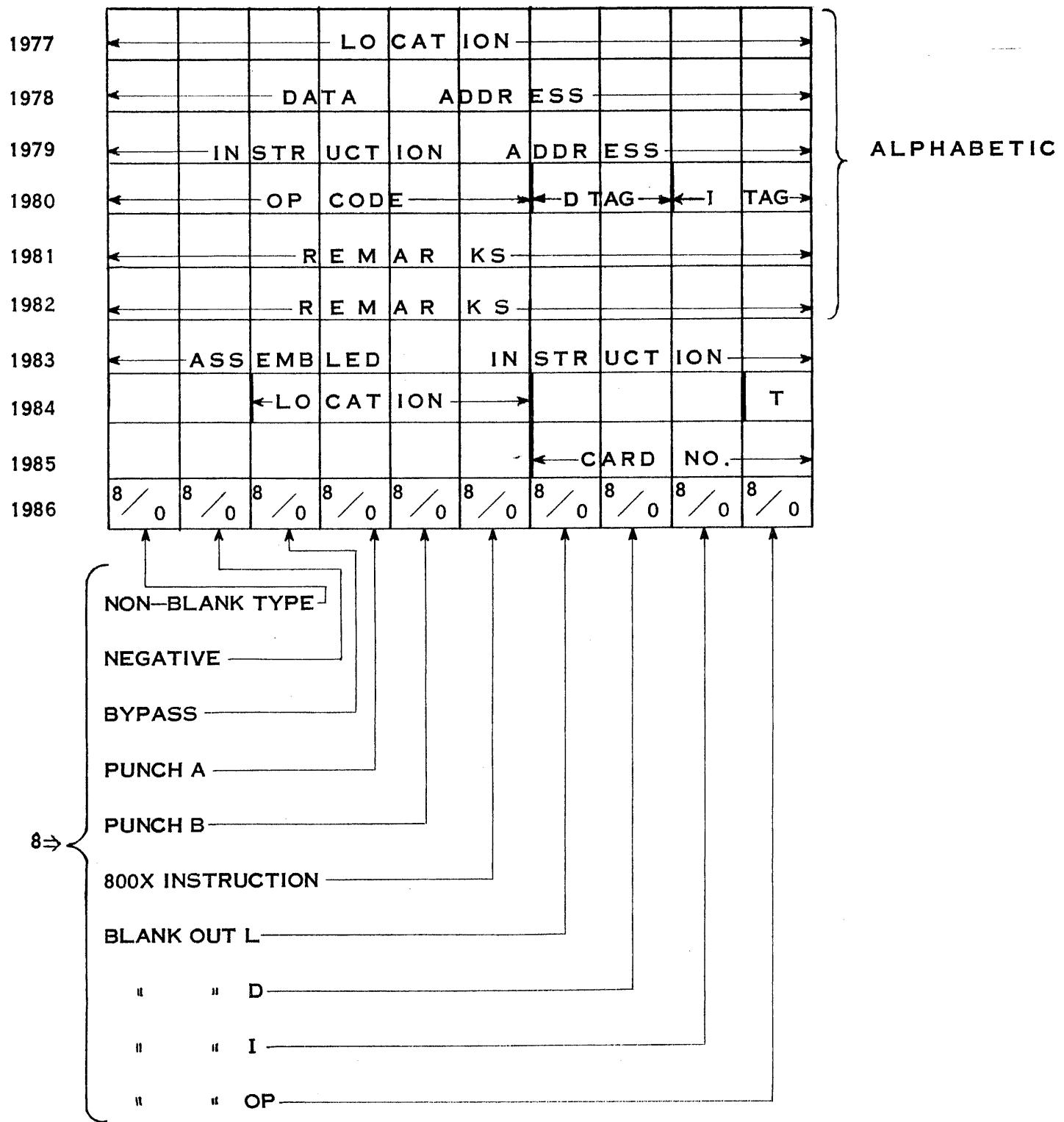


FIGURE 4 : OUTPUT BLOCK : 1977 – 1986

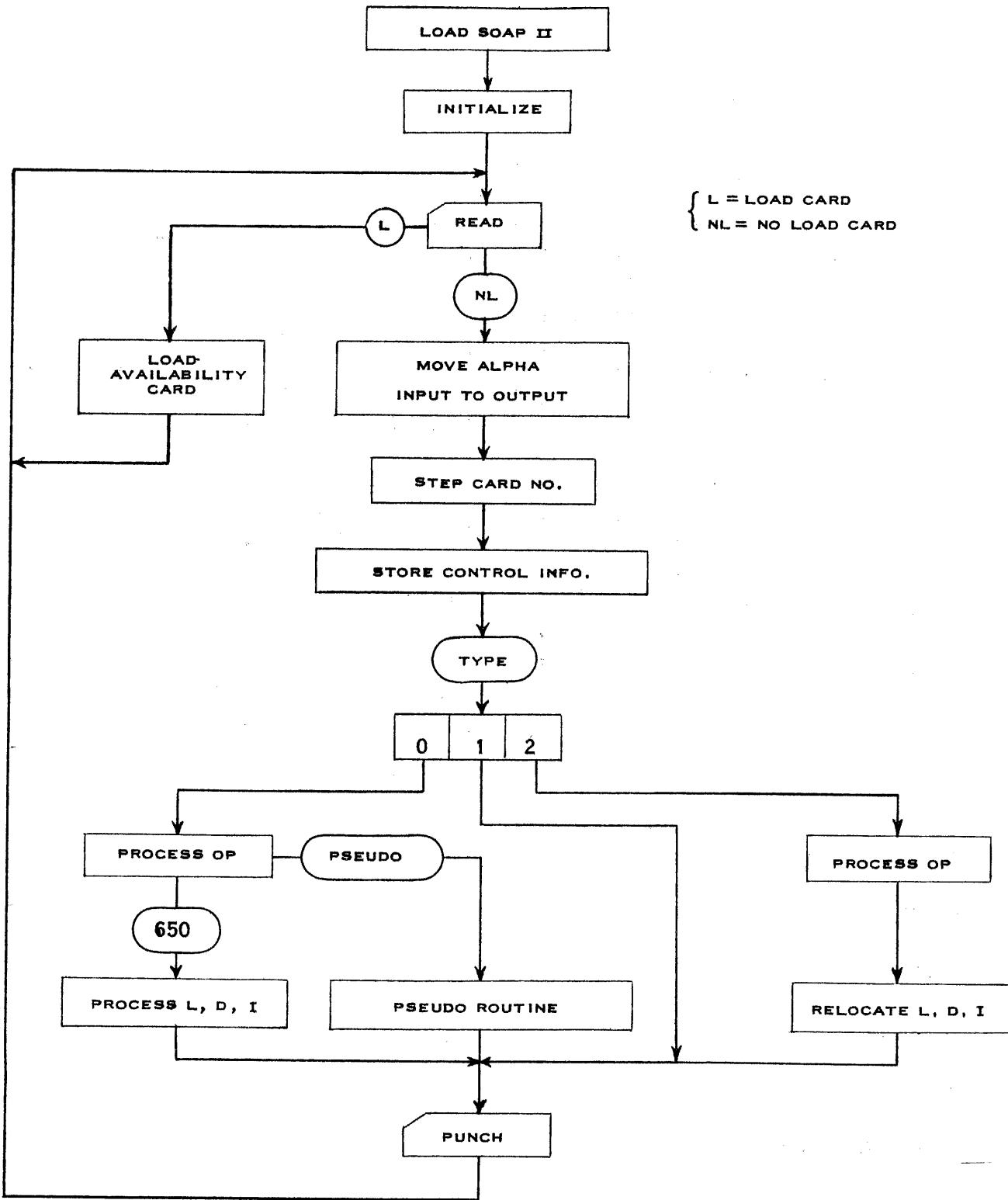
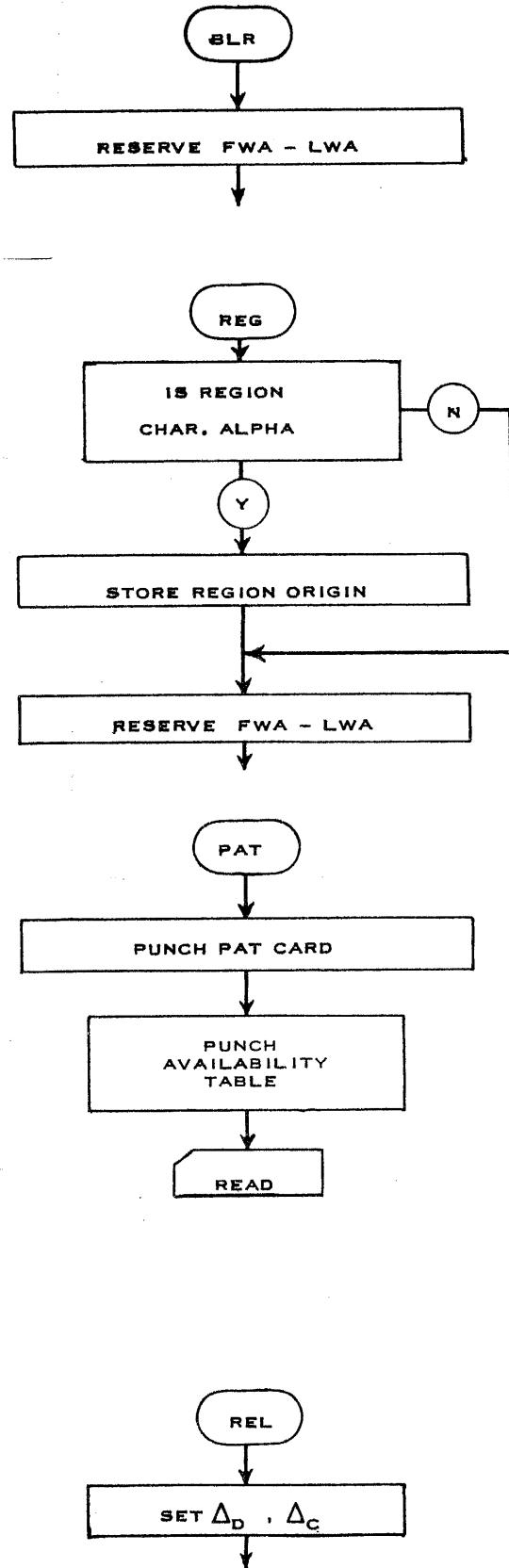
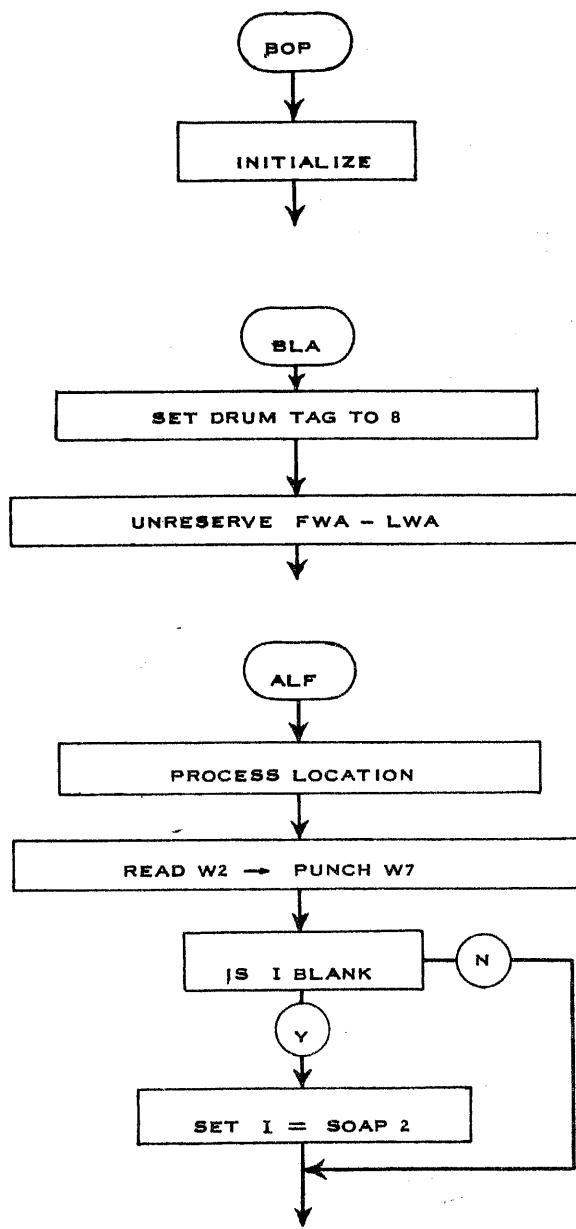
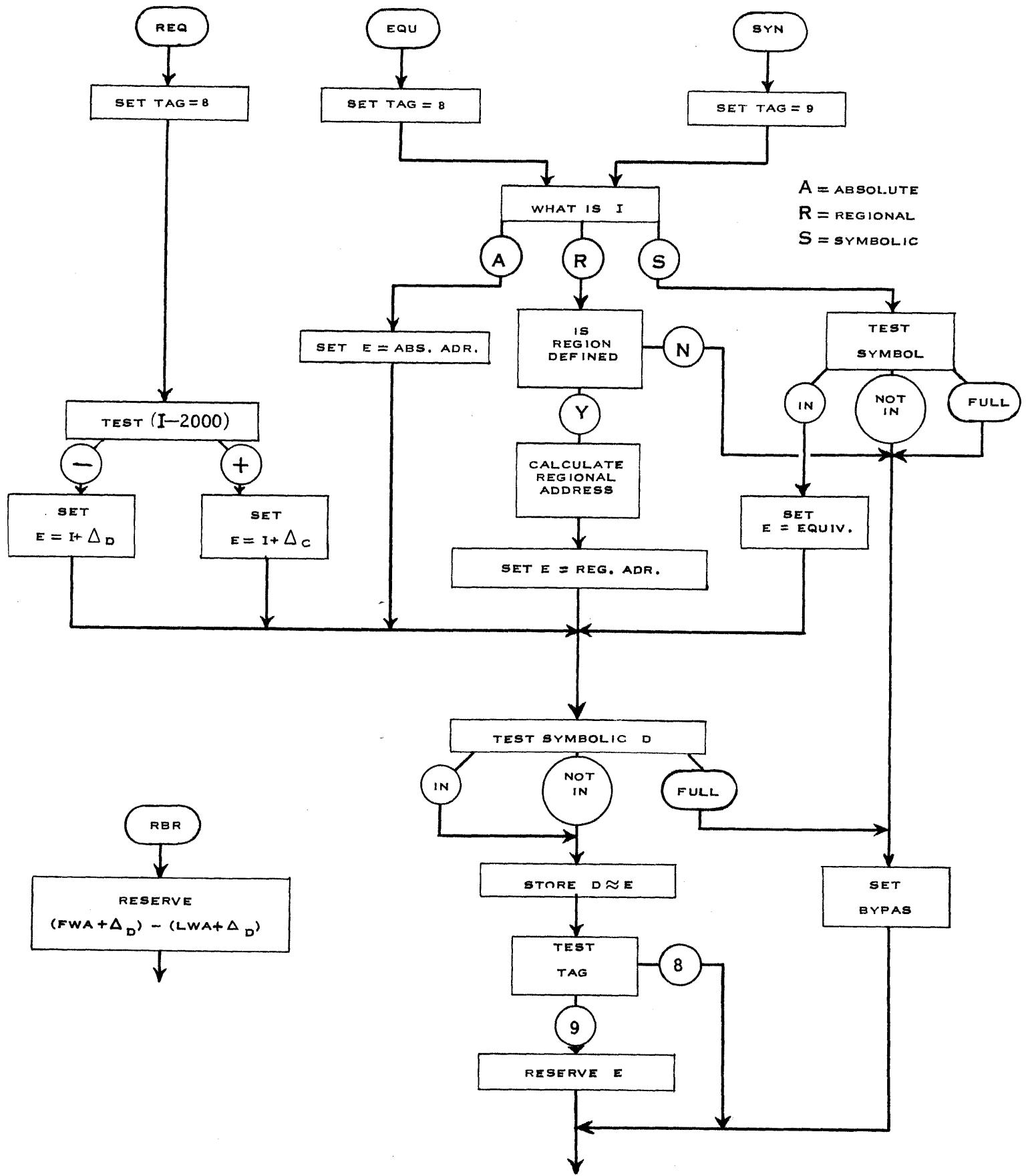


FIGURE 5: BASIC FLOW CHART

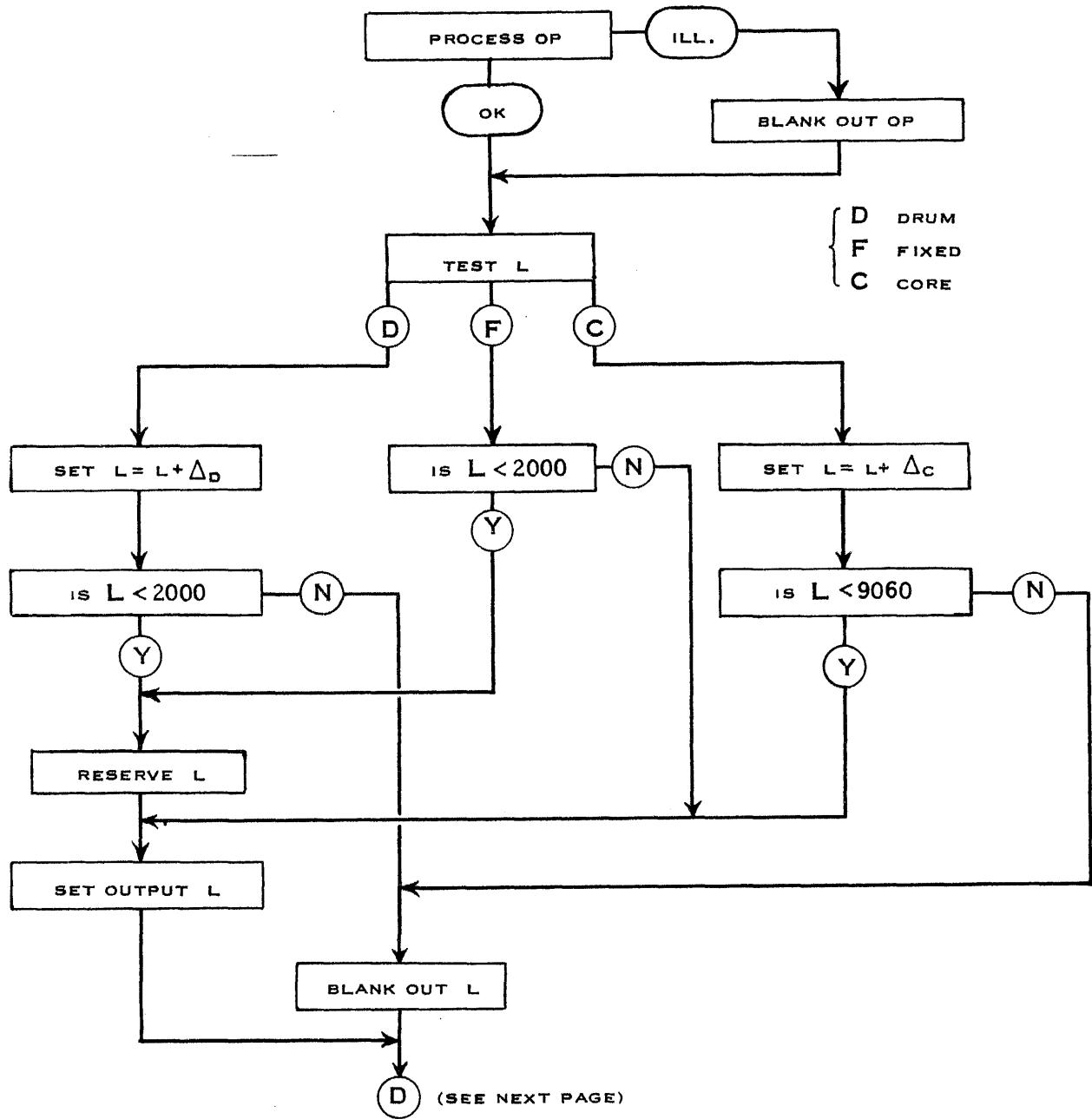
PSEUDO-OPERATION FLOW CHARTS



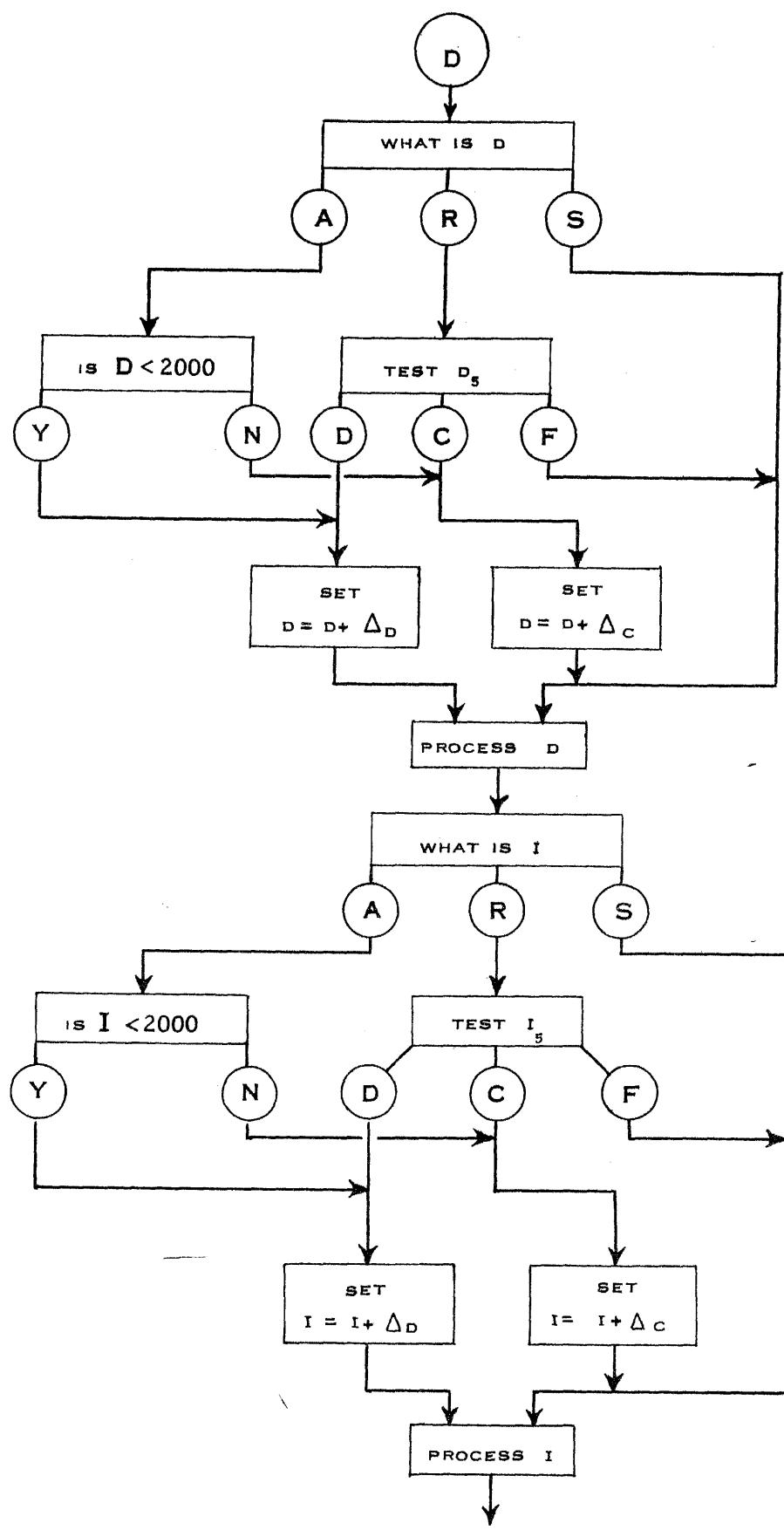
NOTE: EXCEPT FOR PAT, THESE ROUTINES TERMINATE AT "PUNCH"



RELOCATABLE ROUTINE



L MAY BE 1) |F| NNNN | FIXED DRUM OR CORE
 2) | | NNNN | RELOCATABLE DRUM OR CORE



(D₅ = SYMBOLIZER
PART OF D)

Subroutines

SOAP II contains twenty-one subroutines. The following is a description of each subroutine giving

- A. Functional description
- B. Entry conditions and entry point
- C. Exit conditions (where applicable)
- D. Flow chart (where applicable)

1. Store K in all locations from FWA to LWA

Entry: UA (K) Go to SUBR1
 LA(00, FWA, LWA)
 DIST (exit)

2. Reserve or unreserve N sequential drum locations beginning at FWA.

Entry: UA (00, 0000, FWA) Go to {SUB2R - Reserve
 LA (00, 0000, N-1) SUB2U - Unreserve
 DIST (exit)

3. Process op code and exit to K or (K + 1).

Entry: DIST (00, 0000, K) Go to SUBR3

Exit: K: 1. Legal 650 op; numerical op stored in punch word 7 and optimizing table entry stored in "OPTIM".
 2. Illegal symbolic op; "OPTIM" set to (5505050999) and punch control word set to blank out assembled op.

K + 1: Pseudo op (routine begins in 10XX)
 UA (00, 0000, 10XX)
 LA (irrelevant)

4. Determine whether address is absolute, regional or symbolic and accordingly exit to K, (K + 1), or (K + 2).

Entry: UA (alpha. address) Go to SUBR4
 LA (clear)
 DIST (00, 0000, K)

Exit: K: Absolute
 K + 1: Regional } UA (alpha.address)
 K + 2: Symbolic } LA (clear)

Note: If address is symbolic, it is automatically headed and stored in "HSYMB." Upon exiting, the UA also contains the headed symbol.

5. Test absolute address A and exit to K, (K + 1), (K + 2) or (K + 3).

Entry: UA (irrelevant) Go to SUBR5
 LA (00, 0000, A)
 DIST (00, 0000, K)

Exit: K: A < 2000
 K + 1: A = 800X
 K + 2: 9000 ≤ A ≤ 9059 }
 K + 3: Other } UA (clear)
 LA (00, 0000, A)

6. Test symbol and exit to K, K + 1, or K + 2.

Entry: UA (symbol) Go to SUBR6
 LA (clear)
 DIST (00, 0000, K)

Exit: K: In table }
 Equivalent in I address
 part of "EQUIV" }
 K + 1: Not in table }
 K + 2: Not in table and symbol table packed. } UA (symbol)
 LA (00, L*, 0000)
 "LSYMB" (00, L*, 0000)

Note: L* = symbol table location containing symbol on exit K or location in symbol table where symbol may be stored on exit (K + 1).

7. Store symbol in symbol table L* location and equivalent E in equivalence table.

Entry: UA (symbol) Go to SUBR7
 LA (00, L*, E)
 DIST (exit)

8. Obtain equivalent E of symbol stored in symbol table location L*.

Entry: UA (clear) Go to SUBR8
 LA (XX, L*, XXXX)
 DIST (exit)

Exit: UA (clear)
 LA (00, L*, 0000)
 "EQUIV" (00, 0000, E)

9. Calculate regional address A and exit to K or (K + 1).

Entry: UA (alpha. regional address) Go to SUBR9
 LA (clear)
 DIST (00, 00 $\alpha\beta$, K); $\alpha\beta = \begin{cases} 90 - L \\ 88 - D \\ 89 - I \end{cases}$

Exit: K: UA (clear)
 LA (00, 0000, A)
K + 1 : Region undefined or A negative.

10. Set "PUNCH A" 8 in punch control word and execute punch instruction.

Entry: UA (exit) Go to SUB10
 LA (clear)

11. Calculate optimum dynamic address and find and reserve "best" drum location L; exit to K or (K + 1).

Entry: UA (clear) Go to SUB11
 LA (00, 00 $\alpha\beta$, K); $\alpha\beta = \begin{cases} 90 - L \\ 88 - D \\ 89 - I \end{cases}$

Exit: K: UA (clear)
 LA (00, 0000, L)
K + 1: Drum packed

12. Set 8 in punch control word to blank out assembled location.

Entry: UA (exit) Go to SUB12
 LA (clear)

13. Calculate optimum dynamic address XX.

Entry: DIST (00, 000 β , K) Go to SUB13
 $\beta = \begin{cases} 8-D \\ 9-I \end{cases}$

Exit: K: UA (clear)
 LA (00, 0000, 00XX) $00 \leq XX \leq 49$

14. Punch availability table.

Entry: DIST (exit)

Go to SUB14

15. Head symbol.

Entry: UA (symbol)
LA (clear)
DIST (exit)

Go to SUB15

Exit: UA (headed symbol)
LA (clear)

16. Initialize assembly.

Entry: DIST (exit)

Go to SUB16

17. Calculate drum equivalent E of 800X

Entry: UA (exit)

Go to $\begin{cases} \text{SB17D} & - D \text{ equivalent} \\ \text{SB17I} & - I \text{ equivalent} \end{cases}$

Exit: UA (clear)
LA (00, 0000, E)

18. Index address A giving A*.

Entry: UA (irrelevant)
LA (00, 0000, A)
DIST (00, 00 α , β , K)

Go to INDEX

$$\alpha = \begin{cases} 8-\text{Drum} \\ 9-\text{Core} \end{cases}, \quad \beta = \begin{cases} 8-D \\ 9-I \end{cases}$$

Exit: UA (irrelevant)
LA (00, 0000, A*)

19. Process location.

Entry: DIST (exit)

Go to PROCL

20. Process data address.

Entry: DIST (exit)

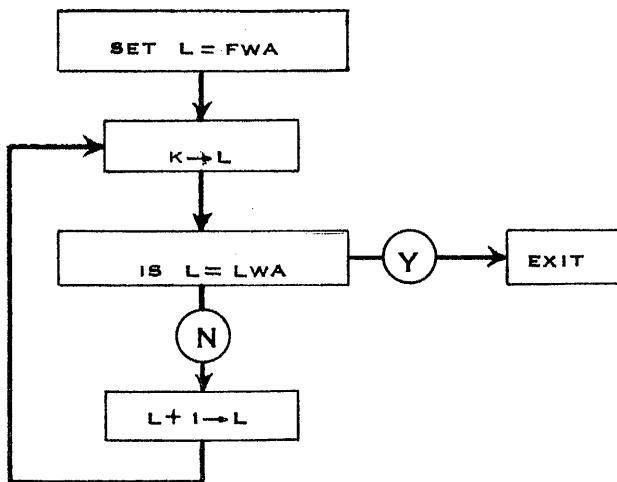
Go to PROCD

21. Process Instruction address.

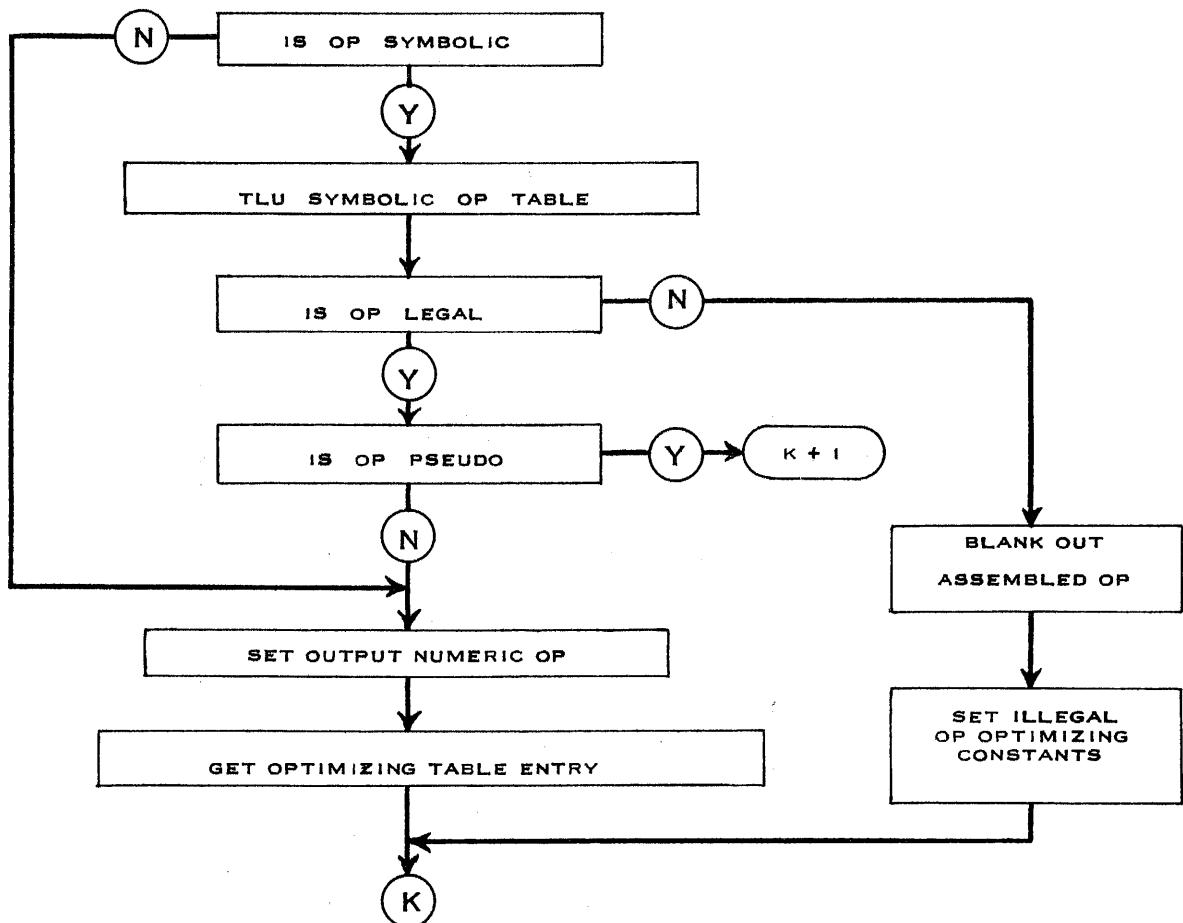
Entry: DIST (exit)

Go to PROCI

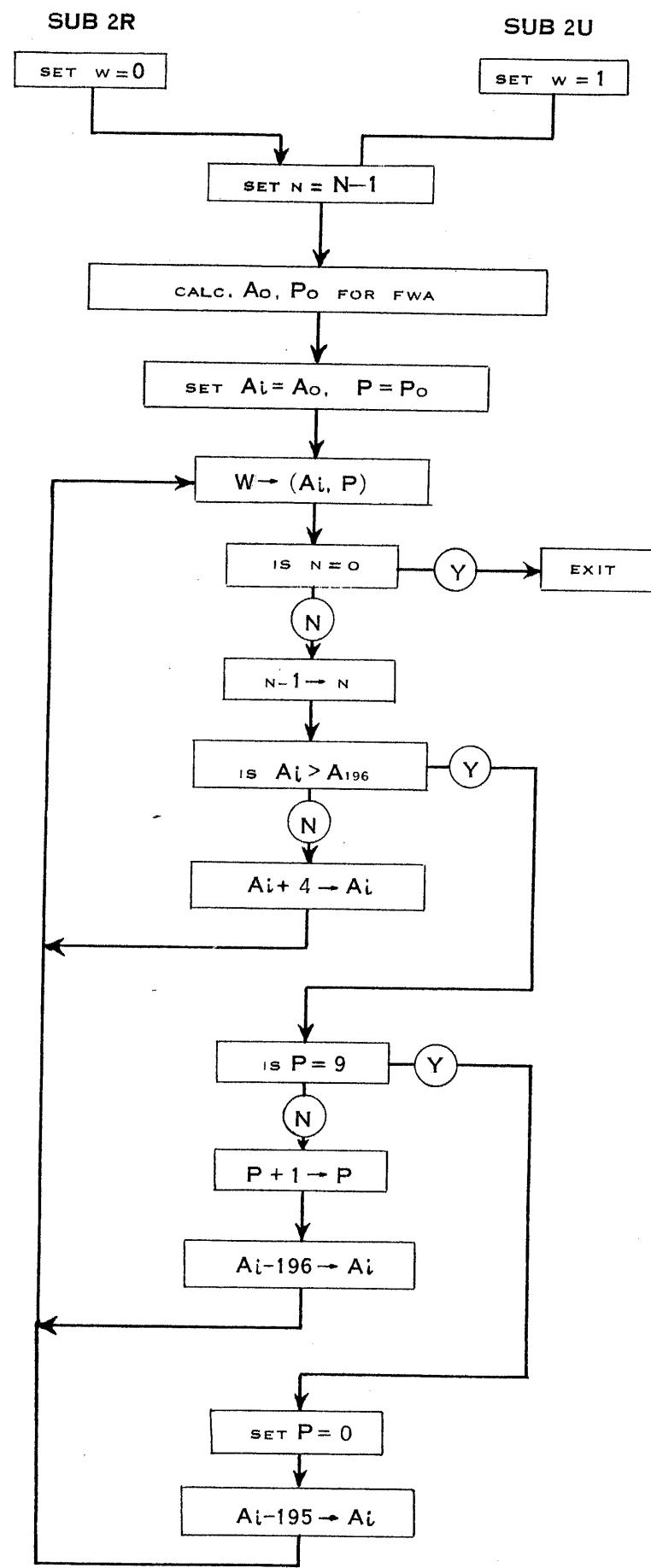
SUBROUTINE FLOW CHARTS



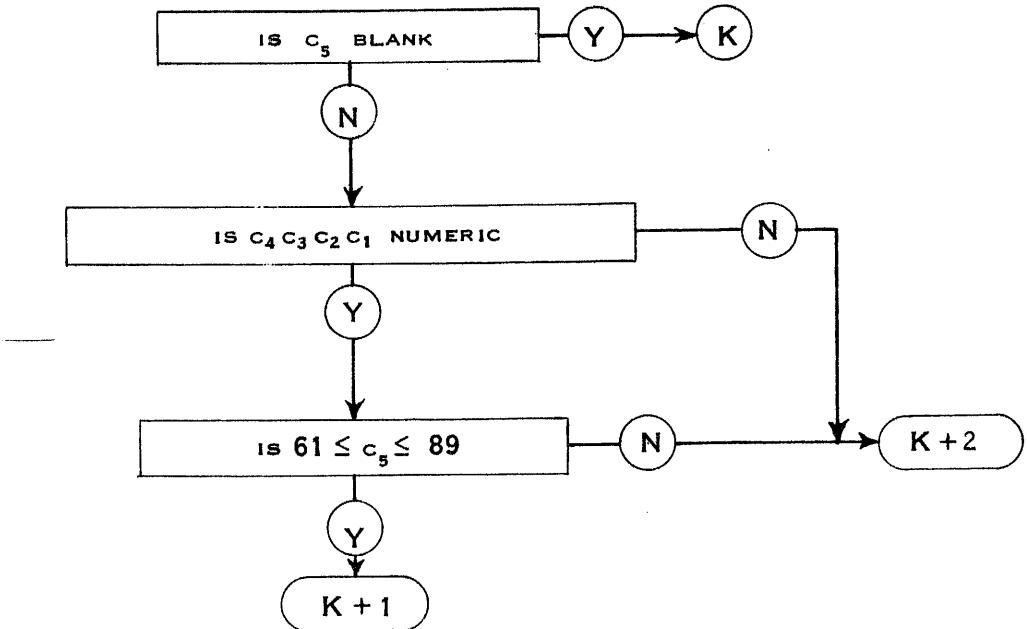
SUBROUTINE 1: STORE K IN FWA – LWA



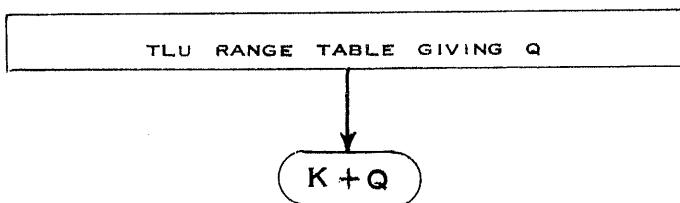
SUBROUTINE 3: PROCESS OP CODE



SUBROUTINE 2: RESERVE OR UNRESERVE N WORDS STARTING AT FWA

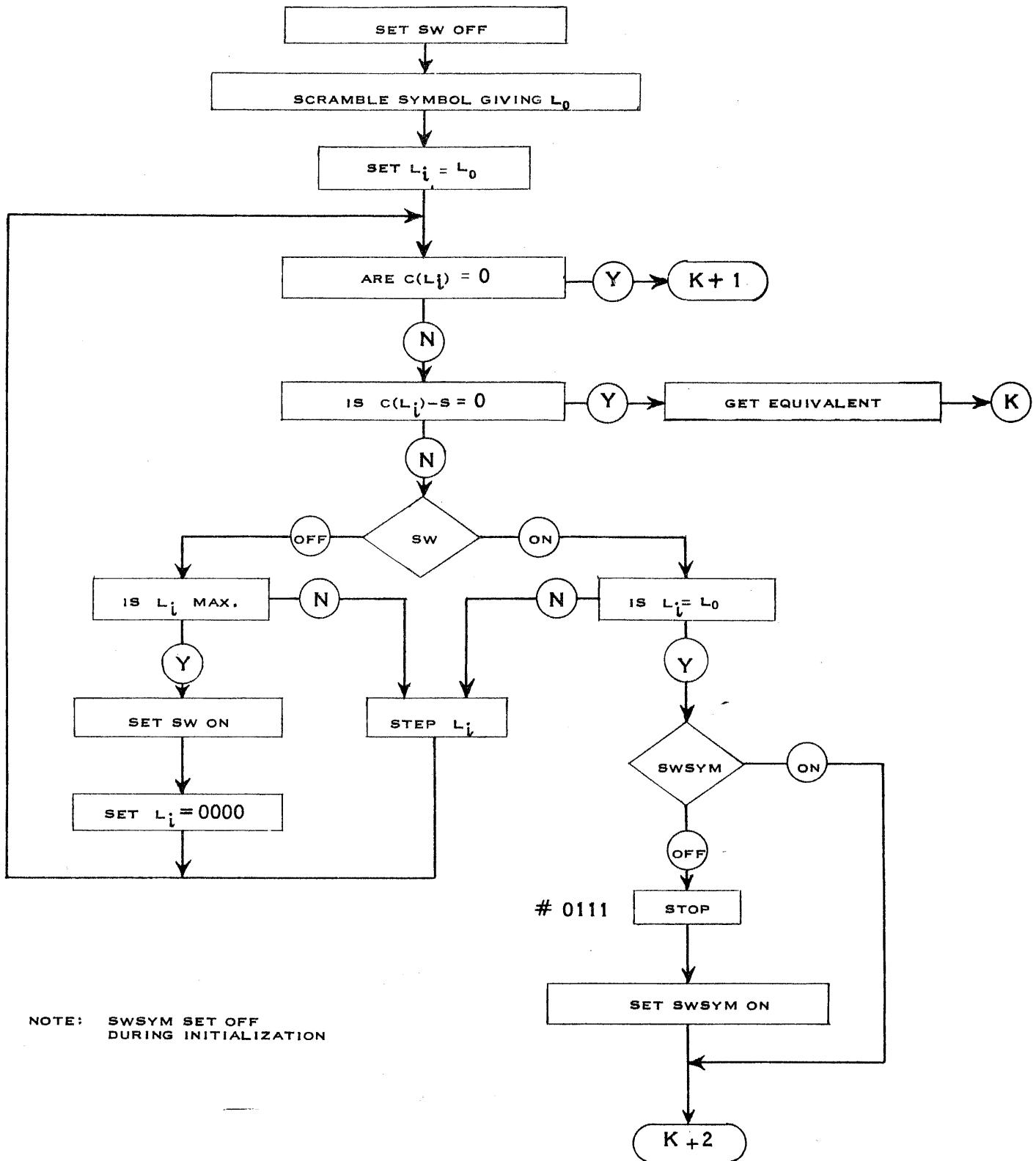


SUBROUTINE 4: ADDRESS TEST

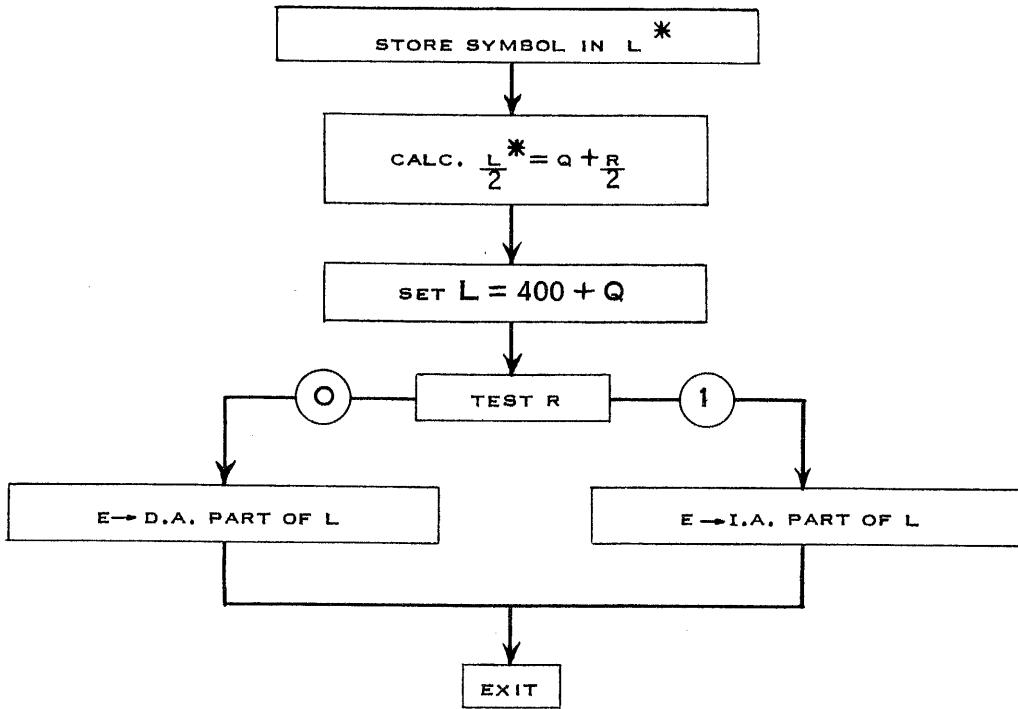


Q	RANGE
0	$0000 \leq A \leq 1999$
1	$\{ 8000 \leq A \leq 8003$ $8005 \leq A \leq 8007$
2	$9000 \leq A \leq 9059$
3	OTHER

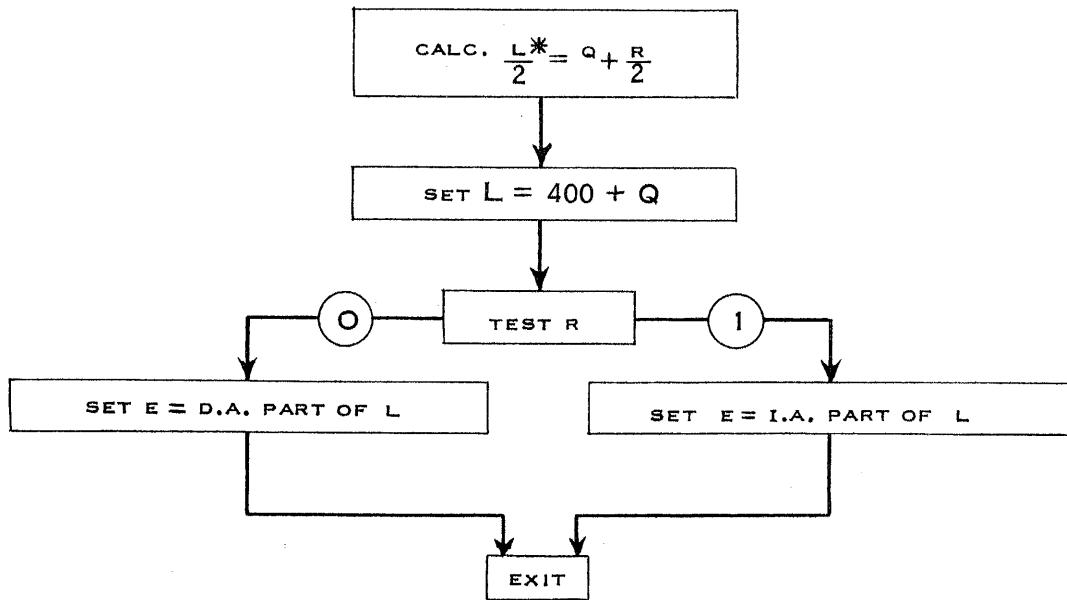
SUBROUTINE 5: ADDRESS RANGE TEST



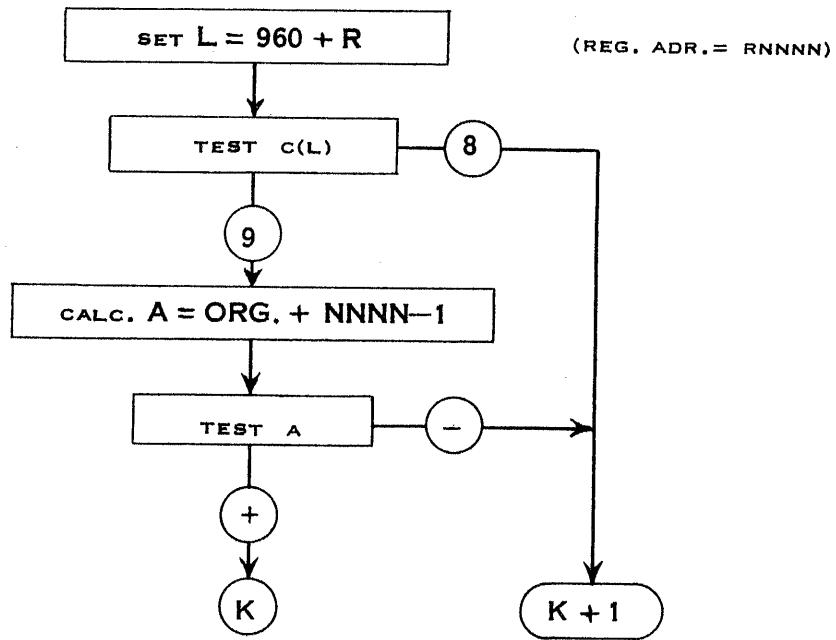
SUBROUTINE 6: SYMBOL TEST



SUBROUTINE 7: STORE SYMBOL AND EQUIVALENT



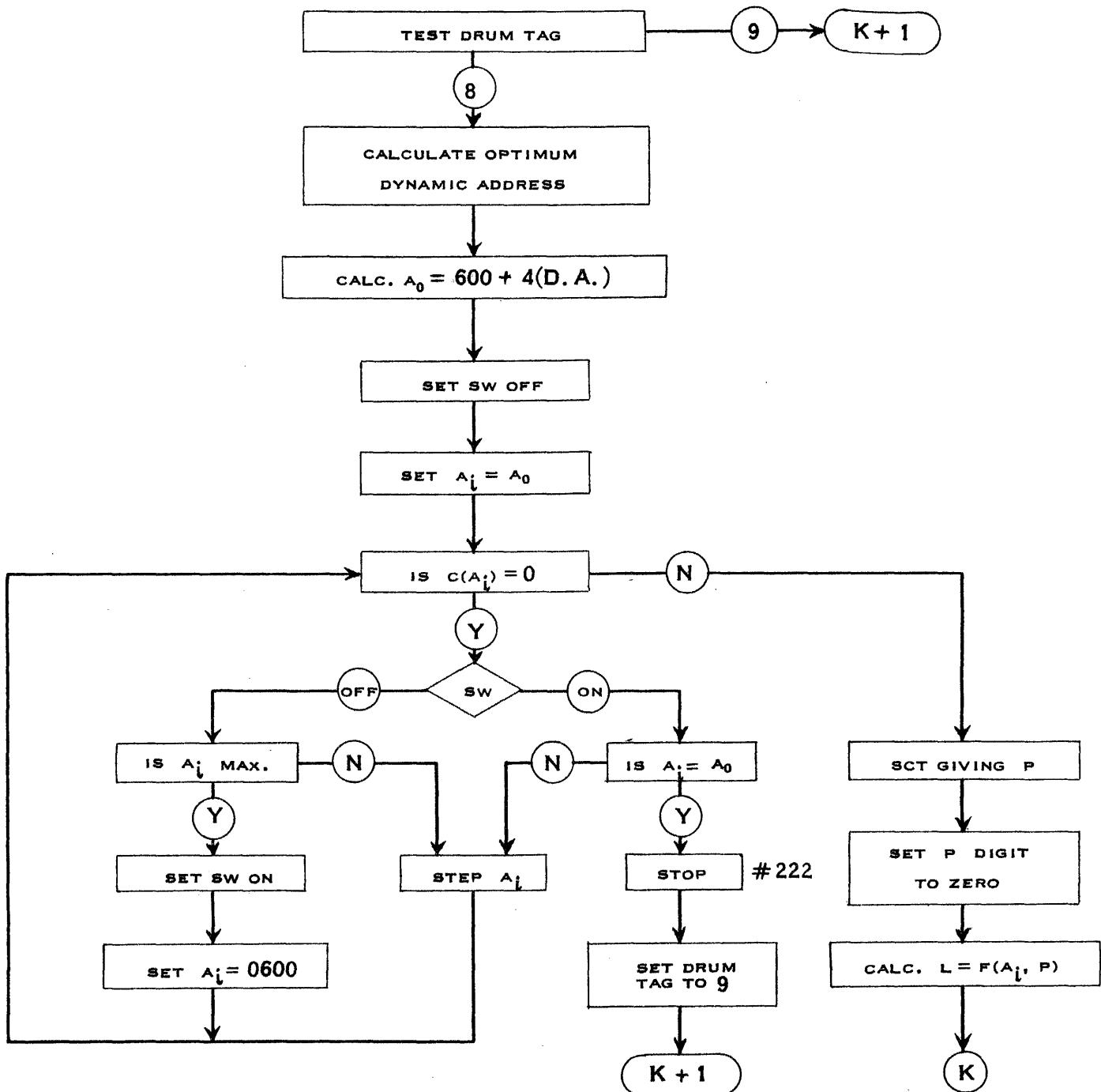
SUBROUTINE 8: OBTAIN SYMBOLIC EQUIVALENT



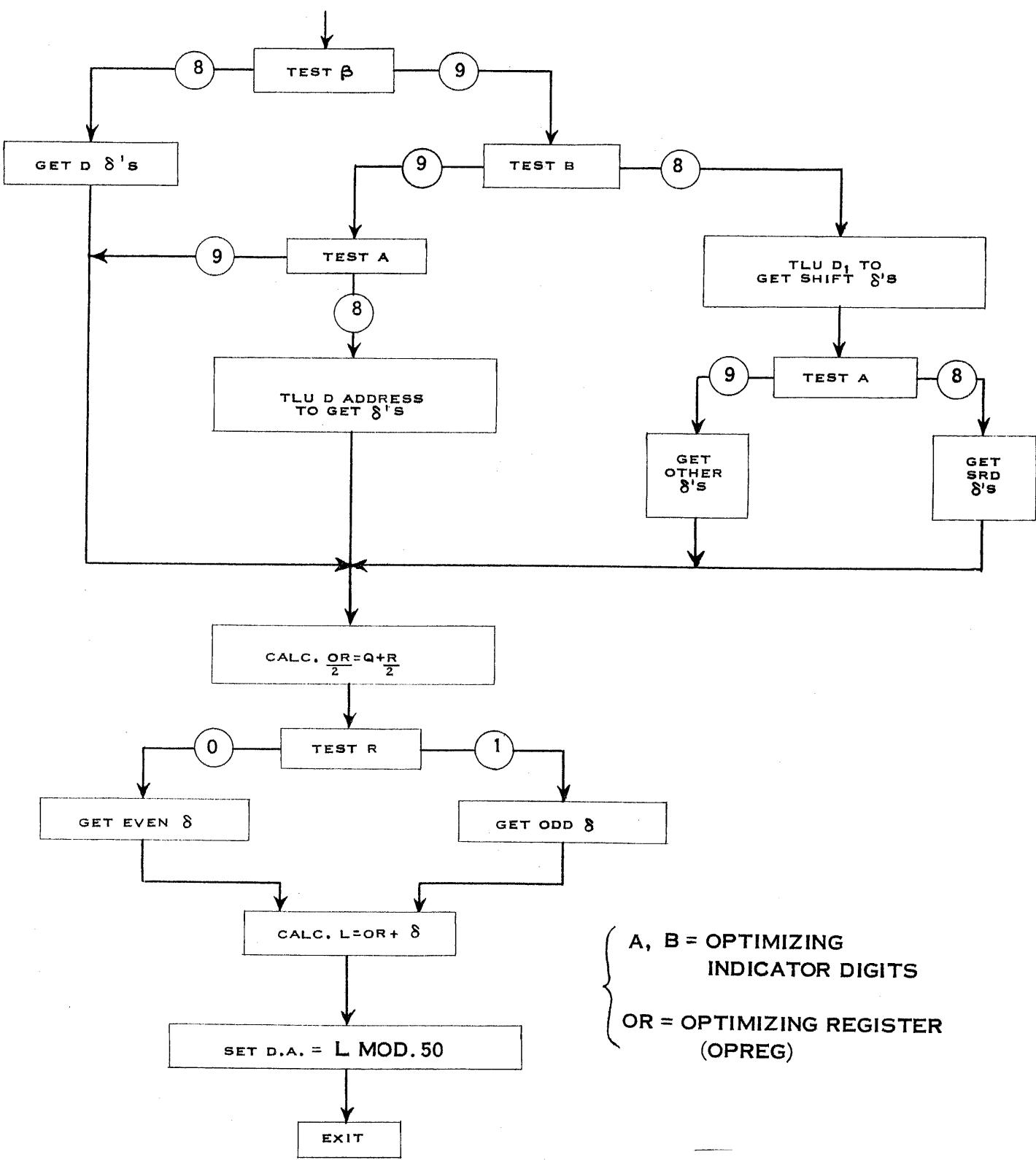
SUBROUTINE 9 : CALCULATE REGIONAL ADDRESS

SUBROUTINE 10 : (SEE PROGRAM LISTING)

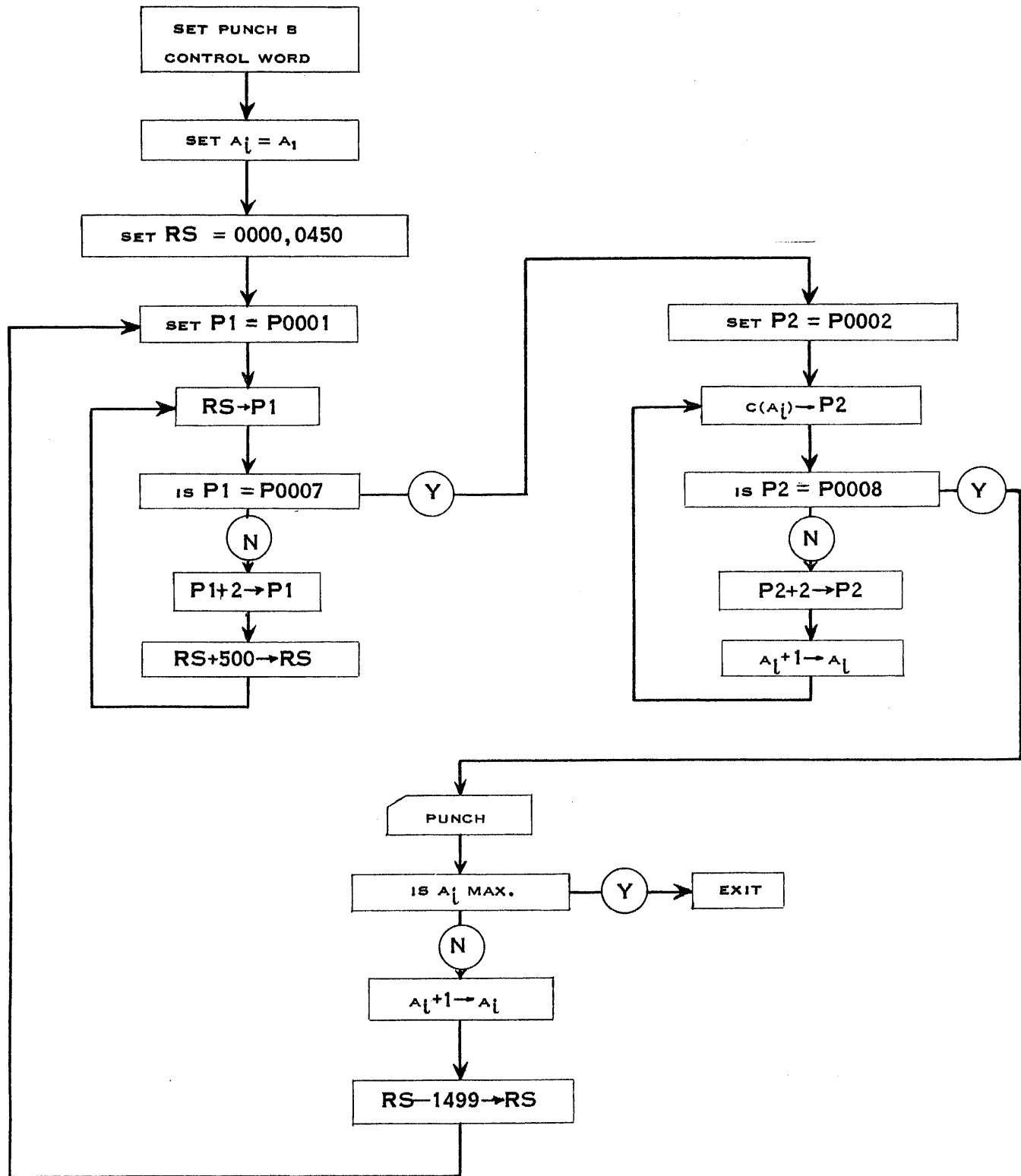
SUBROUTINE 12 : (SEE PROGRAM LISTING)



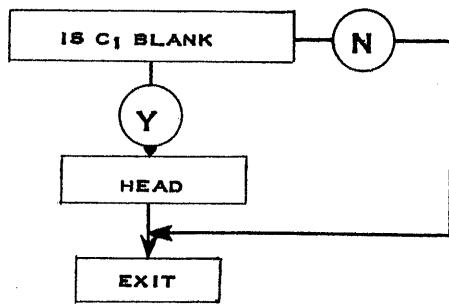
SUBROUTINE 11: FIND AND RESERVE BEST DRUM LOCATION



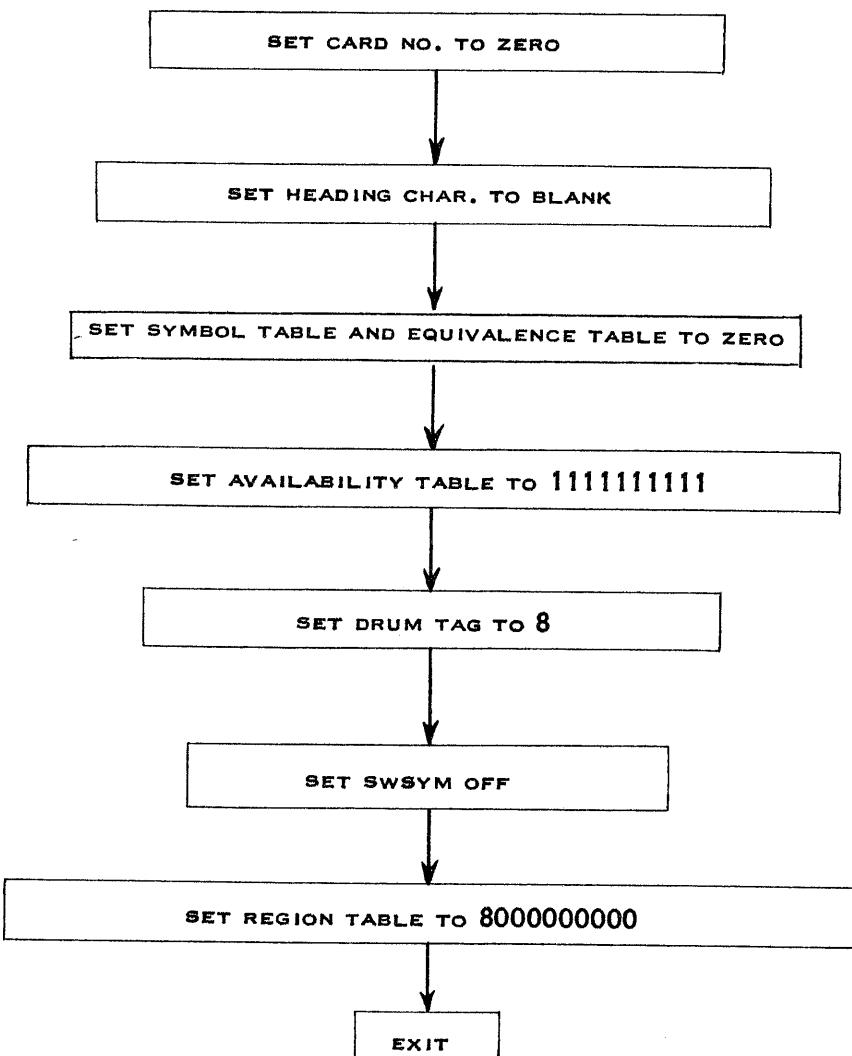
SUBROUTINE 13: CALCULATE OPTIMUM DYNAMIC ADDRESS



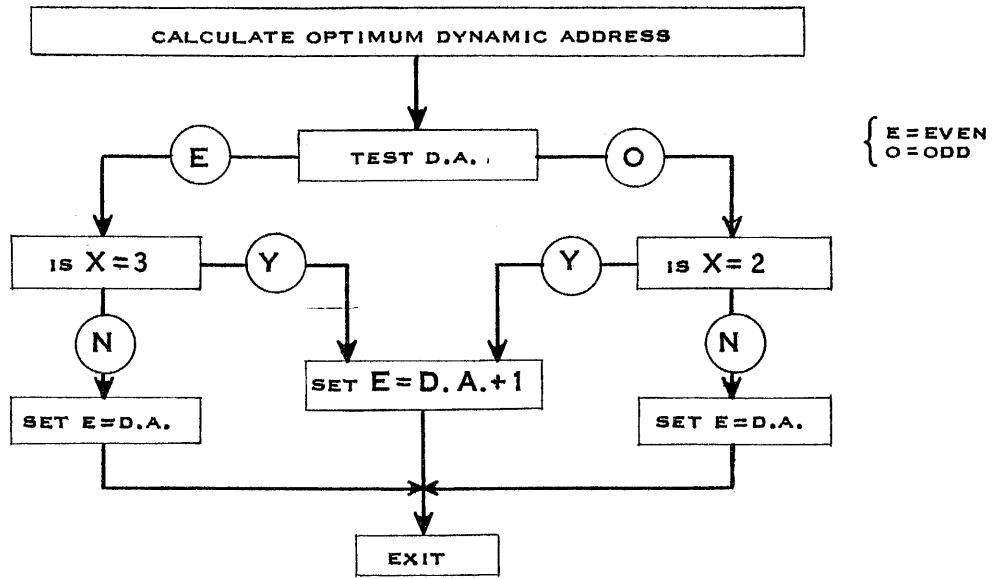
SUBROUTINE 14: PUNCH AVAILABILITY TABLE



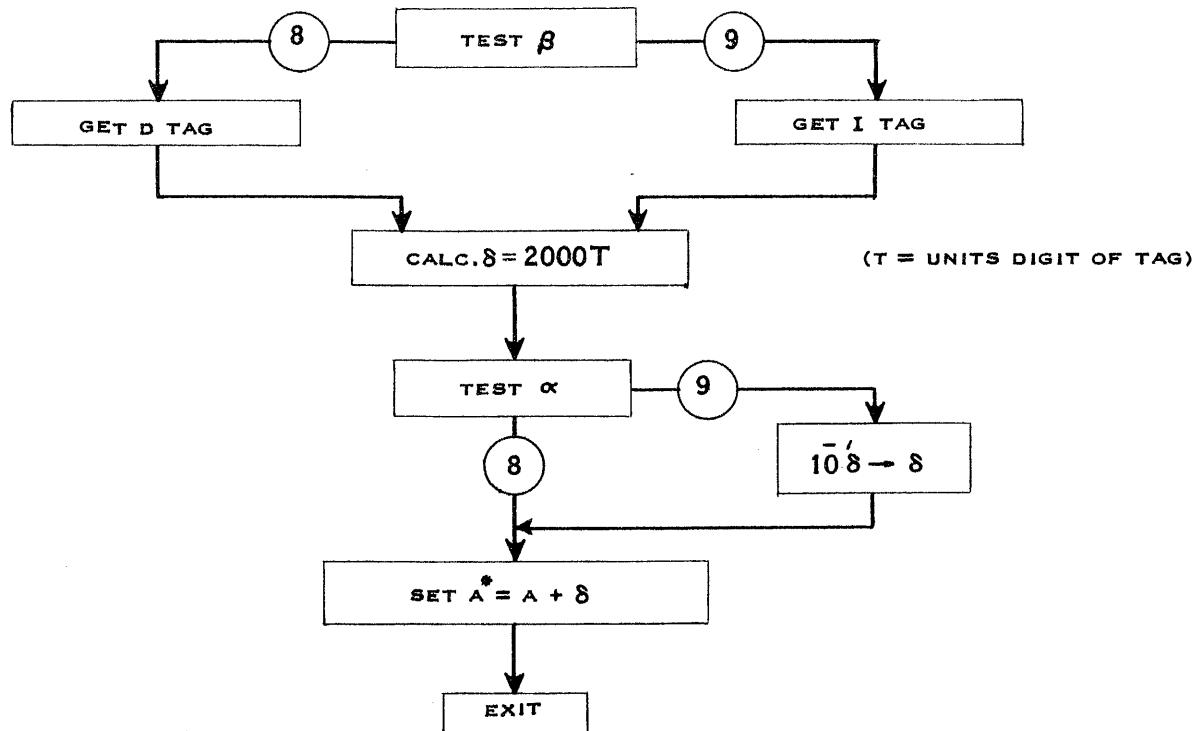
SUBROUTINE 15: HEADING



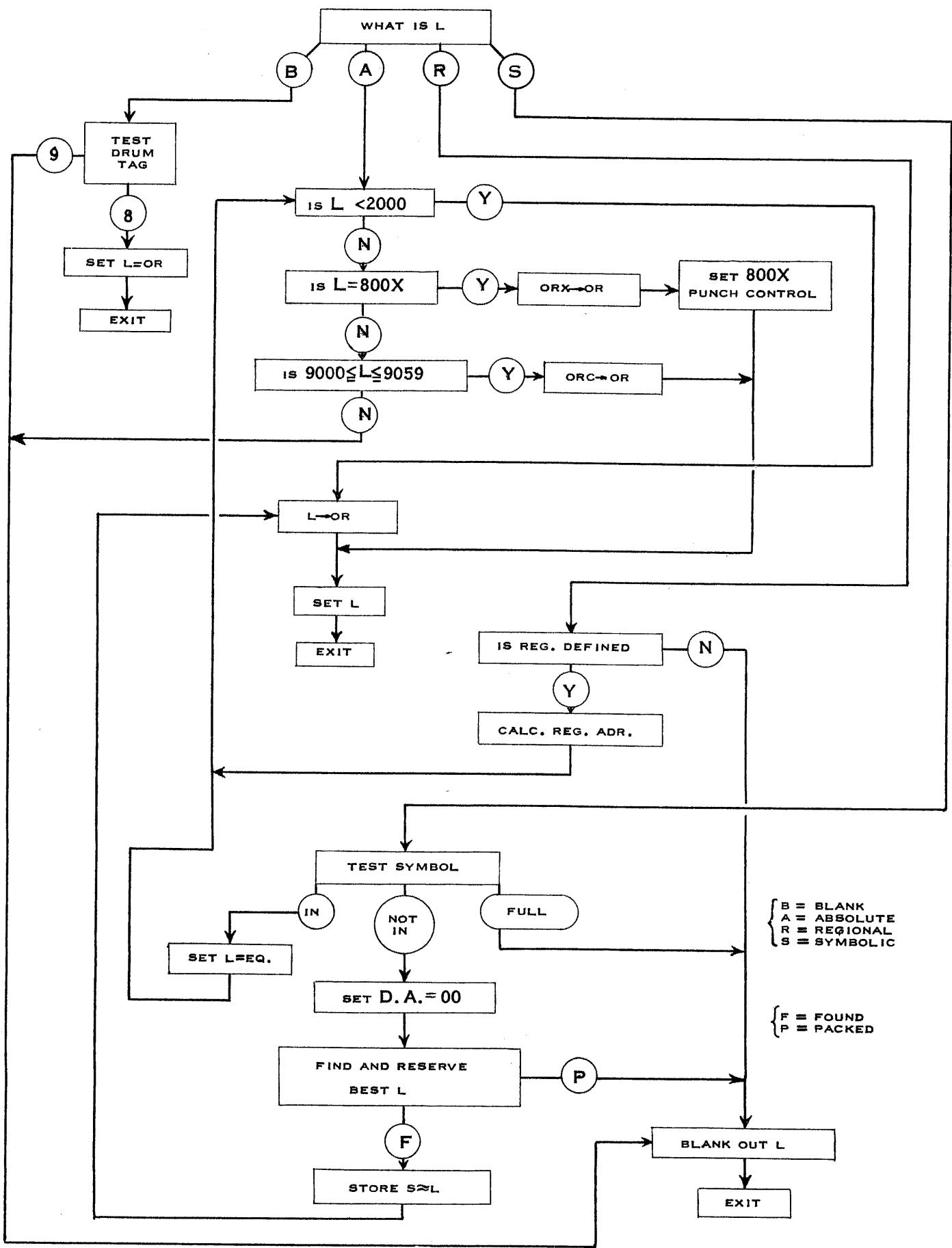
SUBROUTINE 16: INITIALIZATION



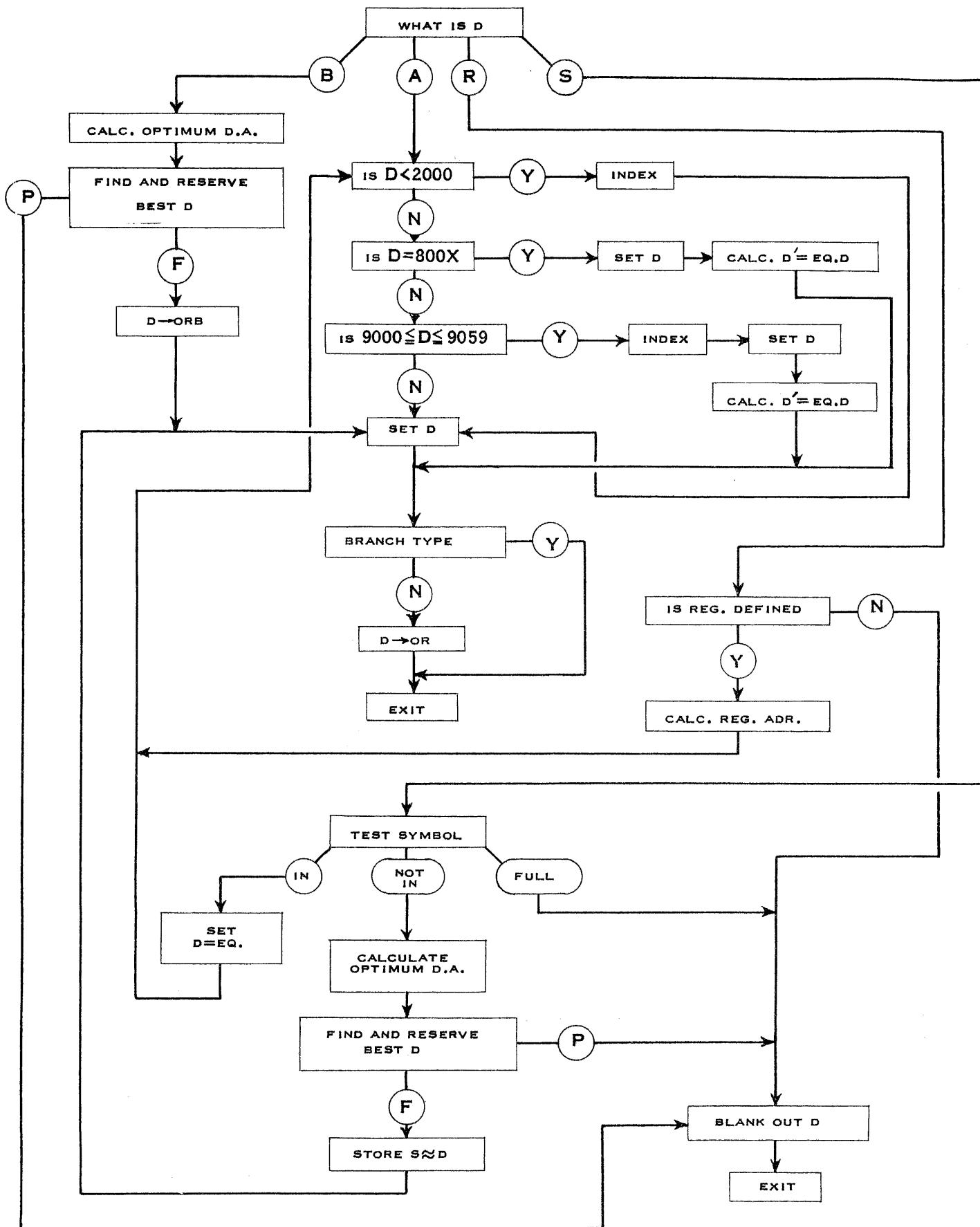
SUBROUTINE 17: CALCULATE 800X DRUM EQUIVALENT



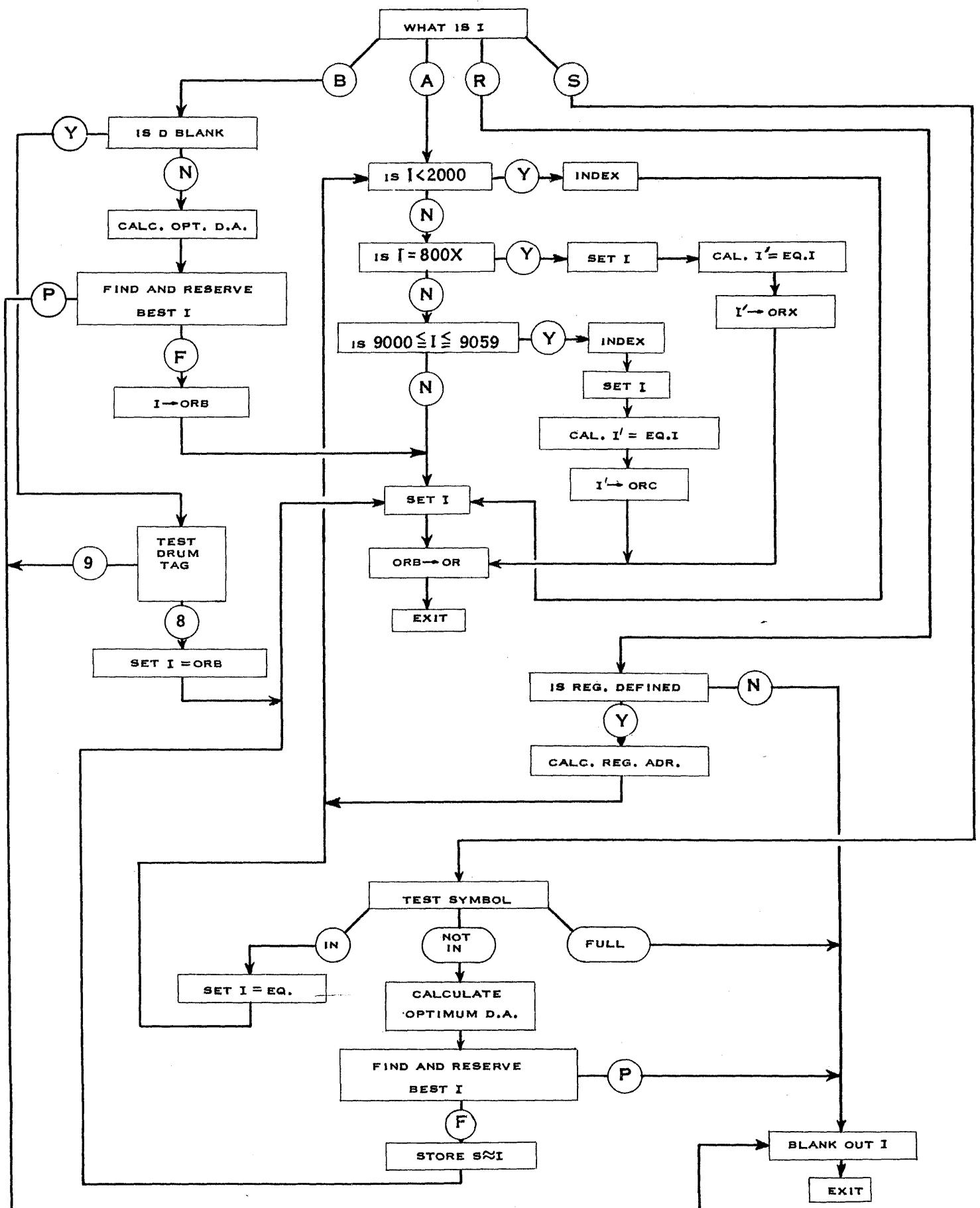
SUBROUTINE 18: INDEX ADDRESS



SUBROUTINE 19: PROCESS LOCATION



SUBROUTINE 20: PROCESS DATA ADDRESS



SUBROUTINE 21: PROCESS INSTRUCTION ADDRESS

SOAP 2
 SYMBOLIC OPTIMAL ASSEMBLY PROG
 FOR THE
 IBM 650
 DATA PROCESSING SYSTEM

REG	S0000	0399	SYMBOL TAB
REG	E0400	0599	EQUIV TABL
REG	A0600	0799	AVAIL TABL
REG	N0800	0899	NUM OP TAB
BLR	1000	1000	ENTRY
BLR	1001	1012	PSEUDO ENT
BLR	1021	1049	REG TABLE
REG	C1050	1164	SYM OP TAB
BLA	1098	1099	
BLA	1148	1149	
BLR	1900	1900	MANUAL PAT
SYN	READC	1950	
BLR	1951	1962	READ AREA
REG	T1990	1992	TYPE
REG	P1977	1986	PUNCH AREA
BLR	1998	1999	
EQU	XXXX1	1961	
EQU	XXXX2	1962	
BLR	1563	1570	TYPE2
BLA	1565	1565	
BLA	1569	1569	
REG	B0900	0901	SUB 2
REG	H0902	0904	TYPE 2
REG	M0905	0907	
REG	Q0908	0909	
REG	R1200	1207	SUB 5
REG	U1250	1259	SUB 13
REG	V0910	0912	EQU SYN
REG	X0913	0914	TYPE 0
REG	Y0915	0916	EQU SYN
REG	Z0917	0919	
REG	W0920	0922	
BLR	0923	0968	MULTIPLE R
REG	I1300	1307	SUB 13

CONTROL PROGRAM

1000	LDD	READC	SUB16	ENTRY	1000	69	1950	1353
READC	RCD	1999	1998	READ CARD	1950	70	1999	1998
1998	LDD	1951		TRANSFER	1998	69	1951	1354
	STD	P0001		ALPHABETIC	1354	24	1977	0980
	LDD	1952		INPUT	0980	69	1952	1355
	STD	P0002		TO	1355	24	1978	0981

	LDD	1953		OUTPUT	0981	69	1953	1356
	STD	P0003			1356	24	1979	0982
	LDD	1954			0982	69	1954	1357
	STD	P0004			1357	24	1980	0983
	LDD	1955			0983	69	1955	1208
	STD	P0005			1208	24	1981	0984
	LDD	1956			0984	69	1956	1209
	STD	P0006			1209	24	1982	0985
	RAL	P0009		STEP	0985	65	1985	0989
	ALO	IIXXX		CARD	0989	15	0992	0997
	STL	P0009		NUMBER	0997	20	1985	0988
	RAU	1960		STORE	0988	60	1960	1015
	SRT	0002		CONTROL	1015	30	0002	0971
	STU	P0008		INFO	0971	21	1984	0987
	AUP	TORG			0987	10	0990	0995
8003	STL	P0010	8003	TRANSFER	0995	20	1986	8003
	NOP	0000	9999	TO TYPE	8003	00	0000	9999
PUNCH	PCH	P0001	READC	PUNCH READ	1350	71	1977	1950
SETCC	RAU	READC	SUB10	SUDO EXIT	1400	60	1950	1405
1900	LDD	READC	SUB14	MANUAL PAT	1900	69	1950	1403
T0001	LDD	X	SUBR3	TYPE 0	1990	69	0993	0996
X0001	LDD		PROCL	650	0913	69	1016	0969
	LDD		PROCD	COMMAND OR	1016	69	1019	0972
	LDD	PUNCH	PROCI	CONSTANTS	1019	69	1350	1453
X0002	NOP	0000	8003	PSEUDO OP	0914	00	0000	8003
T0002	RAU	READC	SUB10	TYPE 1 COM	1991	60	1950	1405
X	00	0000	X0001		0993	00	0000	0913
	RELOCATE ROUTINE							
T0003	HED	T						
Q0002	LDD	Q	SUBR3	PROCESS OP	1992	69	1195	0996
Q0001	NOP	0000	ILLOP	ILLEGAL OP	0909	00	0000	1013
	RAL	1951		IS L FIXED	0908	65	1951	1455
	SLT	0002			1455	35	0002	1211
	NZU		REL		1211	44	1165	1166
	RAU	1957		FIXED L	1165	60	1957	1261
	SUP	2000I		DRUM CORE	1261	11	1014	1169
	BMI		FC		1169	46	1172	0973
	AUP	8001	RES	FIXED DRUM	1172	10	8001	0979
FC	RAL	1957	SETL	FIXED CORE	0973	65	1957	1311
SR	RAL	XXXX1	SETL	SET L	1450	65	1961	1311
SETL	SLT	0004			1311	35	0004	1171
	LDD	P0008			1171	69	1984	1187
	SDA	P0008	PROD		1187	22	1984	1237

REL	RAU	1957		DRUM CORE	1166	60	1957	1361
	SUP	2000I			1361	11	1014	1219
	BMI		RC		1219	46	1222	1173
	AUP	DDIFF		RELOCATE	1222	10	0975	1179
	BMI		BL	DRUM ADDR	1179	46	1182	1183
	AUP	2000I	RES		1182	10	1014	0979
RES	STU	XXXX1			0979	21	1961	1214
	LDD	SR	SUB2R	RESERVE L	1214	69	1450	1503
RC	RAL	1957		RELOCATE	1173	65	1957	1411
	ALO	CDIFF		CORE	1411	15	1264	1269
	SLO	9060			1269	16	1272	0977
	BMI		BL		0977	46	1180	1183
	ALO	8001	SETL		1180	15	8001	1311
BL	RAU	PROD	SUB12	BLANK L	1183	60	1237	0991
	REG	J0944	0944	D TYPE 2				
	REG	G0941	0941					
PROD	LDD	PROI		SET D EXIT	1237	69	1190	1193
	STD	EXITX			1193	24	1196	0999
	RAU	1952		WHAT IS D	0999	60	1952	1407
	LDD	H	SUBR4		1407	69	1210	1213
H0001	RAL	1958		RELOCATE	0902	65	1958	1263
	SLO	2000I		D OR C	1263	16	1014	1319
	BMI	1564	1563		1319	46	1564	1563
H0002	SRT	0008		C D OR F	0903	30	0008	1221
	AUP	1500	8003		1221	10	0974	8003
1563	RAL	CDIFF	CDD	C	1563	65	1264	1369
1564	RAL	DDIFF	CDD	D	1564	65	0975	1369
1566	RAL	1958	J0001	F	1566	65	1958	0944
CDD	ALO	1958	J0001		1369	15	1958	0944
H0003	NOP	0000	G0003	SYMBOLIC	0904	00	0000	0943
	REG	J0960	0960	I				
	REG	F0955	0955					
PROI	LDD	PUNCH		SET I EXIT	1190	69	1350	1553
	STD	EXITX			1553	24	1196	1099
	RAU	1953		WHAT IS I	1099	60	1953	1457
	LDD	M	SUBR4		1457	69	1260	1213
M0001	RAL	1959		RELOCATE	0905	65	1959	1313
	SLO	2000I		D OR C	1313	16	1014	1419
	BMI	1568	1567		1419	46	1568	1567
M0002	SRT	0008		C D OR F	0906	30	0008	1175
	AUP	1504	8003		1175	10	0978	8003
1567	RAL	CDIFF	CDI	C	1567	65	1264	1469
1568	RAL	DDIFF	CDI	D	1568	65	0975	1469
1570	RAL	1959	J0001	F	1570	65	1959	0960
CDI	ALO	1959	J0001		1469	15	1959	0960
M0003	NOP	0000	F0003	SYMBOLIC	0907	00	0000	0957
Q	00	0000	Q0001	CONSTANTS	1195	00	0000	0908
9060	00	0000	9060		1272	00	0000	9060
H	00	0000	H0001		1210	00	0000	0902
1500	00	0000	1500		0974	00	0000	1500

M	00	0000	M0001		1260	00	0000	0905
1504	00	0000	1504		0978	00	0000	1504
XXXX1	01	0000	XXXX1	ERASEABLE	1961	01	0000	1961
	HED							

BOP ROUTINE

1001	LDD	SUB16	INITIALIZE	1001	69	1404	1353
	RAU	READC	PUNCH	1404	60	1950	1405

BLR ROUTINE

1002	RAL	1959	BLR	RESERVE	1002	65	1959	1363
BLR	SLO	1958		FWA TO	1363	16	1958	1413
	AUP	8001		LWA	1413	10	8001	1271
	LDD	SETCC	SUB2R		1271	69	1400	1503

RBR ROUTINE

1012	RAU	DDIFF			1012	60	0975	1229
	ALO	1959	BLR		1229	15	1959	1363

BLA ROUTINE

1003	LDD	H8XXX	SET DRUM	1003	69	1406	1309	
	STD	DRUMT	TAG TO 8	1309	24	1212	1215	
	RAL	1959	UNRESERVE	1215	65	1959	1463	
	SLO	1958	FWA	1463	16	1958	1513	
	AUP	8001	TO	1513	10	8001	1321	
	LDD	SETCC	SUB2U	LWA	1321	69	1400	1603

REG ROUTINE

	HED	R						
1004	RAL	1952		IS	1004	65	1952	1507
	SRT	0008		REGION	1507	30	0008	1225
	SLO	90XXX		CHARACTER	1225	16	1178	1233
	BMI		1002	ALPHABETIC	1233	46	0986	1002
	ALO	29XXX			0986	15	1189	1243
	BMI	1002			1243	46	1002	1197
	SLT	0004		STORE	1197	35	0004	1557
	ALO	ST		REGION	1557	15	1310	1265
	AUP	1958		ORIGIN	1265	10	1958	1613
8002	AUP	H9XXX	8002		1613	10	1216	8002
	STU	9999	1002		8002	21	9999	1002
ST	STU	1021	1002	CONSTANTS	1310	21	1021	1002
	HED							

EQU REQ AND SYN ROUTINE

1005	HED	Z						
1006	LDD	H8XXX	BOTH	EQU	1005	69	1406	1359
BOTH	LDD	H9XXX	BOTH	SYN	1006	69	1216	1359
	STD	TAG			1359	24	1262	1315

	RAU	1953		WHAT IS I	1315	60	1953	1607
Z0001	LDD	Z	SUBR4		1607	69	1360	1213
Z0002	LDD	Y	SUBR9	ABSOLUTE	0917	65	1959	0915
Y0001	STL	E	TD	REGIONAL	0918	69	1371	1174
Y0002	RAL	P0010	BP	REG ERROR	0915	20	1519	1322
Z0003	LDD	W	SUBR6	SYMBOLIC	0916	65	1986	1191
W0001	RAL	EQUIV	Y0001	S DEFINED	0919	69	1372	1275
W0002	RAL	P0010	BP	S UNDEFIND	0920	65	1223	0915
W0003	RAL	P0010	BP	S TAB FULL	0921	65	1986	1191
					0922	65	1986	1191
TD	RAU	1952		HEAD D	1322	60	1952	1657
	LDD		SUB15	AND EQUATE	1657	69	1410	1663
	LDD	V	SUBR6	TO E	1410	69	1713	1275
V0003	RAL	P0010	BP	S TAB FULL	0912	65	1986	1191
V0001	ALO	E	SD		0910	15	1519	1273
V0002	ALO	E	SD		0911	15	1519	1273
SD	LDD	TT	SUBR7		1273	69	0976	1279
TT	LDD	TAG		TEST TAG	0976	69	1262	1365
	BDO	SETCC			1365	90	1400	0970
	RAU	E		RESERVE E	0970	60	1519	1323
	LDD	SETCC	SUB2R		1323	69	1400	1503
1011	LDD	H8XXX		REQ	1011	69	1406	1409
	STD	TAG-		ROUTINE	1409	24	1262	1415
	RAL	1959		IS I	1415	65	1959	1763
	SLO	2000I		DRUM O COR	1763	16	1014	1569
	BMI	D			1569	46	1422	1373
	ALO	CDIFF	CD		1373	15	1264	1619
D	ALO	DDIFF	CD		1422	15	0975	1619
CD	ALO	2000I	Y0001		1619	15	1014	0915
Z	00	0000	Z0001	CONSTANTS	1360	00	0000	0917
Y	00	0089	Y0001		1371	00	0089	0915
W	00	0000	W0001		1372	00	0000	0920
V	00	0000	V0001		1713	00	0000	0910
88	00	8800	0000		1500	00	8800	0000
BP	ALO	88		BY PASS	1191	15	1500	1505
	STL	P0010	PUNCH		1505	20	1986	1350
TAG	01	0000	TAG	ERASEABLE	1262	01	0000	1262
E	01	0000	E		1519	01	0000	1519
	HED							

ALF ROUTINE

1007	LDD		PROCL	PROCES LOC	1007	69	1460	0969
	LDD	1952			1460	69	1952	1555
	STD	P0007			1555	24	1983	1186
	RAL	1953		REPLACE	1186	65	1953	1707
	NZE	PUNCH		BLANK I	1707	45	1350	1461
	LDD	SOAP2		BY	1461	69	1314	1017
	STD	P0003	PUNCH	SOAP2	1017	24	1979	1350

SOAP2	ALF	SOAP2	SOAP2	CONSTANT	1314	82	7661	7792
PAT ROUTINE								
1008	RAU LDD	READC	SUB10 SUB14		1008 1511	60 69	1511 1950	1405 1403
HED ROUTINE								
1009	RAL SRT STL	1952 0008 0000H		STORE HEADING CHARACTER	1009 1757 1325	65 30 20	1952 0008 1329	1757 1325 1400
REL ROUTINE								
1010	HED RAL NZE RAL STL RAL NZE RAL STL	M 1952 SDD 1958 SDD DDIFF 1953 SCD 1959 SCD CDIFF		SET DRUM AND CORE DELTAS	1010 1807 1510 1561 1228 1857 1560 1611	65 45 65 20 65 45 65 20	1952 1510 1958 1561 1953 1560 1959 1264	1807 1561 1561 1228 1857 1611 1611 1400
DDIFF CDIFF	01 01	0000 0000	DDIFF CDIFF	ERASEABLE	0975 1264	01 01	0000 0000	0975 1264
LOAD AVAILABILITY TABLE								
1999	HED RAL SRT SLT ALO ALO ALO AUP LDD STD SUP NZU AUP ALO	L 1951 0004 0004 8002 8002 C1 C2 9995 9994 TW READC		CALC LOC W2 LOAD TABLE	1999 1605 1465 1375 1283 1241 1149 8003 8002 1247 1655 1459 1167	65 30 35 15 15 15 10 69 24 11 44 10 15	1951 0004 0004 8002 8002 1241 1352 9995 9994 1550 1459 1312 1020	1605 1465 1375 1283 1241 1149 8003 8002 1247 1655 1459 1167 8003
C1 C2 TW TWP2	STD LDD LDD LDD	A0001 1952 1958 1960	TUP 8002 8002 8002	CONSTANTS	0994 1352 1550 1312	24 69 69 69	0600 1952 1958 1960	1247 8002 8002 8002
HED								

OPTIMIZING TABLE

N0001	00	0404	0998	NOP	0800	00	0404	0998
N0002	00	0404	0998	HLT	0801	00	0404	0998
N0003	33	2322	0999	UFA	0802	33	2322	0999
N0004	00	0505	0998	RTC	0803	00	0505	0998
N0005	00	0505	0998	RTN	0804	00	0505	0998
N0006	00	0505	0998	RTA	0805	00	0505	0998
N0007	00	0505	0998	WTN	0806	00	0505	0998
N0008	00	0505	0998	WTA	0807	00	0505	0998
N0009	33	1212	0999	LIB	0808	33	1212	0999
N0010	33	0202	0999	LDI	0809	33	0202	0999
N0011	33	0504	0999	AUP	0810	33	0504	0999
N0012	33	0504	0999	SUP	0811	33	0504	0999
N0013	99	9999	9999	NOT USED	0812	99	9999	9999
N0014	99	9999	9999	NOT USED	0813	99	9999	9999
N0015	33	1110	0999	DIV	0814	33	1110	0999
N0016	33	0504	0999	ALO	0815	33	0504	0999
N0017	33	0504	0999	SLO	0816	33	0504	0999
N0018	33	0504	0999	AML	0817	33	0504	0999
N0019	33	0504	0999	SML	0818	33	0504	0999
N0020	33	2120	0999	MPY	0819	33	2120	0999
N0021	54	0303	0999	STL	0820	54	0303	0999
N0022	45	0303	0999	STU	0821	45	0303	0999
N0023	34	0303	0999	SDA	0822	34	0303	0999
N0024	34	0303	0999	SIA	0823	34	0303	0999
N0025	33	0303	0999	STD	0824	33	0303	0999
N0026	44	0505	0998	NTS	0825	44	0505	0998
N0027	00	0505	0998	BIN	0826	00	0505	0998
N0028	00	0505	0998	SET	0827	00	0505	0998
N0029	33	1212	0999	SIB	0828	33	1212	0999
N0030	33	0202	0999	STI	0829	33	0202	0999
N0031	00	0000	0988	SRT	0830	00	0000	0988
N0032	00	0000	0888	SRD	0831	00	0000	0888
N0033	33	2726	0999	FAD	0832	33	2726	0999
N0034	33	2726	0999	FSB	0833	33	2726	0999
N0035	33	0000	0999	FDV	0834	33	0000	0999
N0036	00	0000	0988	SLT	0835	00	0000	0988
N0037	00	0000	0988	SCT	0836	00	0000	0988
N0038	33	2726	0999	FAM	0837	33	2726	0999
N0039	33	2726	0999	FSM	0838	33	2726	0999
N0040	33	0000	0999	FMP	0839	33	0000	0999
N0041	33	0404	0998	NZA	0840	33	0404	0998
N0042	33	0404	0998	BMA	0841	33	0404	0998
N0043	33	0404	0998	NZB	0842	33	0404	0998
N0044	33	0404	0998	BMB	0843	33	0404	0998
N0045	34	0405	0998	NZU	0844	34	0405	0998
N0046	43	0504	0998	NZE	0845	43	0504	0998
N0047	33	0404	0998	BMI	0846	33	0404	0998
N0048	33	0505	0998	BOV	0847	33	0505	0998
N0049	33	0404	0998	NZC	0848	33	0404	0998
N0050	33	0404	0998	BMC	0849	33	0404	0998
N0051	00	0000	0898	AXA	0850	00	0000	0898
N0052	00	0000	0898	SXA	0851	00	0000	0898
N0053	00	0000	0898	AXB	0852	00	0000	0898

N0054	00	0000	0898	SXB	0853	00	0000	0898
N0055	44	0505	0998	NEF	0854	44	0505	0998
N0056	00	0505	0998	RWD	0855	00	0505	0998
N0057	00	0505	0998	WTM	0856	00	0505	0998
N0058	00	0505	0998	BST	0857	00	0505	0998
N0059	00	0000	0898	AXC	0858	00	0000	0898
N0060	00	0000	0898	SXC	0859	00	0000	0898
N0061	33	0504	0999	RAU	0860	33	0504	0999
N0062	33	0504	0999	RSU	0861	33	0504	0999
N0063	99	9999	9999	NOT USED	0862	99	9999	9999
N0064	99	9999	9999	NOT USED	0863	99	9999	9999
N0065	33	1110	0999	DVR	0864	33	1110	0999
N0066	33	0504	0999	RAL	0865	33	0504	0999
N0067	33	0504	0999	RSL	0866	33	0504	0999
N0068	33	0504	0999	RAM	0867	33	0504	0999
N0069	33	0504	0999	RSM	0868	33	0504	0999
N0070	33	0303	0999	LDD	0869	33	0303	0999
N0071	00	0000	0999	RD1	0870	00	0000	0999
N0072	00	0000	0999	WR1	0871	00	0000	0999
N0073	00	0000	0999	RC1	0872	00	0000	0999
N0074	00	0000	0999	RD2	0873	00	0000	0999
N0075	00	0000	0999	WR2	0874	00	0000	0999
N0076	00	0000	0999	RC2	0875	00	0000	0999
N0077	00	0000	0999	RD3	0876	00	0000	0999
N0078	00	0000	0999	WR3	0877	00	0000	0999
N0079	00	0000	0999	RC3	0878	00	0000	0999
N0080	00	0505	0999	RPY	0879	00	0505	0999
N0081	00	0000	0898	RAA	0880	00	0000	0898
N0082	00	0000	0898	RSA	0881	00	0000	0898
N0083	00	0000	0898	RAB	0882	00	0000	0898
N0084	00	0000	0898	RSB	0883	00	0000	0898
N0085	33	0506	0999	TLU	0884	33	0506	0999
N0086	00	0606	0998	SDS	0885	00	0606	0998
N0087	00	0606	0998	RDS	0886	00	0606	0998
N0088	00	0606	0998	WDS	0887	00	0606	0998
N0089	00	0000	0898	RAC	0888	00	0000	0898
N0090	00	0000	0898	RSC	0889	00	0000	0898
N0091	44	0505	0998	BDO	0890	44	0505	0998
N0092	33	0505	0998	BD1	0891	33	0505	0998
N0093	33	0505	0998	BD2	0892	33	0505	0998
N0094	33	0505	0998	BD3	0893	33	0505	0998
N0095	33	0505	0998	BD4	0894	33	0505	0998
N0096	33	0505	0998	BD5	0895	33	0505	0998
N0097	33	0505	0998	BD6	0896	33	0505	0998
N0098	33	0505	0998	BD7	0897	33	0505	0998
N0099	33	0505	0998	BD8	0898	33	0505	0998
N0100	44	0505	0998	BD9	0899	44	0505	0998

SYMBOLIC OP TABLE

C0001	61	7366	1007	ALF	1050	61	7366	1007
C0002	61	7376	0015	ALO	1051	61	7376	0015
C0003	61	7473	0017	AML	1052	61	7473	0017
C0004	61	8477	0010	AUP	1053	61	8477	0010
C0005	61	8761	0050	AXA	1054	61	8761	0050

C0006	61	8762	0052	AXB	1055	61	8762	0052
C0007	61	8763	0058	AXC	1056	61	8763	0058
C0008	62	6476	0090	BDO	1057	62	6476	0090
C0009	62	6491	0091	BD1	1058	62	6491	0091
C0010	62	6492	0092	BD2	1059	62	6492	0092
C0011	62	6493	0093	BD3	1060	62	6493	0093
C0012	62	6494	0094	BD4	1061	62	6494	0094
C0013	62	6495	0095	BD5	1062	62	6495	0095
C0014	62	6496	0096	BD6	1063	62	6496	0096
C0015	62	6497	0097	BD7	1064	62	6497	0097
C0016	62	6498	0098	BD8	1065	62	6498	0098
C0017	62	6499	0099	BD9	1066	62	6499	0099
C0018	62	6975	0026	BIN	1067	62	6975	0026
C0019	62	7361	1003	BLA	1068	62	7361	1003
C0020	62	7379	1002	BLR	1069	62	7379	1002
C0021	62	7461	0041	BMA	1070	62	7461	0041
C0022	62	7462	0043	BMB	1071	62	7462	0043
C0023	62	7463	0049	BMC	1072	62	7463	0049
C0024	62	7469	0046	BMI	1073	62	7469	0046
C0025	62	7677	1001	BOP	1074	62	7677	1001
C0026	62	7685	0047	BOV	1075	62	7685	0047
C0027	62	8283	0057	BST	1076	62	8283	0057
C0028	64	6985	0014	DIV	1077	64	6985	0014
C0029	64	8579	0064	DVR	1078	64	8579	0064
C0030	65	7884	1005	EQU	1079	65	7884	1005
C0031	66	6164	0032	FAD	1080	66	6164	0032
C0032	66	6174	0037	FAM	1081	66	6174	0037
C0033	66	6485	0034	FDV	1082	66	6485	0034
C0034	66	7477	0039	FMP	1083	66	7477	0039
C0035	66	8262	0033	FSB	1084	66	8262	0033
C0036	66	8274	0038	FSM	1085	66	8274	0038
C0037	68	6564	1009	HED	1086	68	6564	1009
C0038	68	7383	0001	HLT	1087	68	7383	0001
C0039	73	6464	0069	LDD	1088	73	6464	0069
C0040	73	6469	0009	LDI	1089	73	6469	0009
C0041	73	6962	0008	LIB	1090	73	6962	0008
C0042	74	7788	0019	MPY	1091	74	7788	0019
C0043	75	6566	0054	NEF	1092	75	6566	0054
C0044	75	7677	0000	NOP	1093	75	7677	0000
C0045	75	8382	0025	NTS	1094	75	8382	0025
C0046	75	8961	0040	NZA	1095	75	8961	0040
C0047	75	8962	0042	NZB	1096	75	8962	0042
C0048	75	8963	0048	NZC	1097	75	8963	0048
C0051	75	8965	0045	NZE	1100	75	8965	0045
C0052	75	8984	0044	NZU	1101	75	8984	0044
C0053	77	6183	1008	PAT	1102	77	6183	1008
C0054	77	6368	0071	PCH	1103	77	6368	0071
C0055	79	6161	0080	RAA	1104	79	6161	0080
C0056	79	6162	0082	RAB	1105	79	6162	0082
C0057	79	6163	0088	RAC	1106	79	6163	0088
C0058	79	6173	0065	RAL	1107	79	6173	0065
C0059	79	6174	0067	RAM	1108	79	6174	0067
C0060	79	6184	0060	RAU	1109	79	6184	0060
C0061	79	6279	1012	RBR	1110	79	6279	1012
C0062	79	6364	0070	RCD	1111	79	6364	0070

C0063	79	6391	0072	RC1	1112	79	6391	0072
C0064	79	6392	0075	RC2	1113	79	6392	0075
C0065	79	6393	0078	RC3	1114	79	6393	0078
C0066	79	6482	0086	RDS	1115	79	6482	0086
C0067	79	6491	0070	RD1	1116	79	6491	0070
C0068	79	6492	0073	RD2	1117	79	6492	0073
C0069	79	6493	0076	RD3	1118	79	6493	0076
C0070	79	6567	1004	REG	1119	79	6567	1004
C0071	79	6573	1010	REL	1120	79	6573	1010
C0072	79	6578	1011	REQ	1121	79	6578	1011
C0073	79	7788	0079	RPY	1122	79	7788	0079
C0074	79	8261	0081	RSA	1123	79	8261	0081
C0075	79	8262	0083	RSB	1124	79	8262	0083
C0076	79	8263	0089	RSC	1125	79	8263	0089
C0077	79	8273	0066	RSL	1126	79	8273	0066
C0078	79	8274	0068	RSM	1127	79	8274	0068
C0079	79	8284	0061	RSU	1128	79	8284	0061
C0080	79	8361	0005	RTA	1129	79	8361	0005
C0081	79	8363	0003	RTC	1130	79	8363	0003
C0082	79	8375	0004	RTN	1131	79	8375	0004
C0083	79	8664	0055	RWD	1132	79	8664	0055
C0084	82	6383	0036	SCT	1133	82	6383	0036
C0085	82	6461	0022	SDA	1134	82	6461	0022
C0086	82	6482	0085	SDS	1135	82	6482	0085
C0087	82	6583	0027	SET	1136	82	6583	0027
C0088	82	6961	0023	SIA	1137	82	6961	0023
C0089	82	6962	0028	SIB	1138	82	6962	0028
C0090	82	7376	0016	SLO	1139	82	7376	0016
C0091	82	7383	0035	SLT	1140	82	7383	0035
C0092	82	7473	0018	SML	1141	82	7473	0018
C0093	82	7964	0031	SRD	1142	82	7964	0031
C0094	82	7983	0030	SRT	1143	82	7983	0030
C0095	82	8364	0024	STD	1144	82	8364	0024
C0096	82	8369	0029	STI	1145	82	8369	0029
C0097	82	8373	0020	STL	1146	82	8373	0020
C0098	82	8384	0021	STU	1147	82	8384	0021
C0101	82	8477	0011	SUP	1150	82	8477	0011
C0102	82	8761	0051	SXA	1151	82	8761	0051
C0103	82	8762	0053	SXB	1152	82	8762	0053
C0104	82	8763	0059	SXC	1153	82	8763	0059
C0105	82	8875	1006	SYN	1154	82	8875	1006
C0106	83	7384	0084	TLU	1155	83	7384	0084
C0107	84	6661	0002	UFA	1156	84	6661	0002
C0108	86	6482	0087	WDS	1157	86	6482	0087
C0109	86	7991	0071	WR1	1158	86	7991	0071
C0110	86	7992	0074	WR2	1159	86	7992	0074
C0111	86	7993	0077	WR3	1160	86	7993	0077
C0112	86	8361	0007	WTA	1161	86	8361	0007
C0113	86	8374	0056	WTM	1162	86	8374	0056
C0114	86	8375	0006	WTN	1163	86	8375	0006
C0115	99	9999	0001	TABLE END	1164	99	9999	0001

SUB 1 STORE K IN FWA TO LWA

	HED	1						
SUBR1	STD	EXITX	STORE EXIT	1600	24	1196	1199	
	STU	XXXX1	STORE K	1199	21	1961	1364	
	LDD	FWA	SET	1364	69	1217	1170	
	SDA	FWA	FWA	1170	22	1217	1220	
	SLT	0004	LWA	1220	35	0004	1181	
	SDA	XXXX2	ADDRESSES	1181	22	1962	1515	
	RAU	FWA		1515	60	1217	1421	
	ALO	XXXX1	8003		15	1961	8003	
8003	STL	9999	NEXT	STORE K	8003	20	9999	1402
NEXT	SUP	XXXX2		END OF	1402	11	1962	1267
	NZU		EXITX	LOOP TEST	1267	44	1471	1196
	AUP	8001			1471	10	8001	1379
	AUP	1DXXX	8003	STEP	1379	10	1020	8003
FWA	STL	9998	NEXT	CONSTANTS	1217	20	9998	1402
XXXX2	01	0000	XXXX2	ERASEABLE	1962	01	0000	1962
	HED							

SUB 2 RESERVE UNRESERVE

	HED	2						
SUB2R	STD	EXITY	RESERVE	1503	24	1813	1316	
	LDD	Z	ENTRY	1316	69	1769	1472	
SUB2U	STD	EXITY	UNRESERVE	1603	24	1813	1366	
	LDD	I	RU	1366	69	1565	1472	
RU	STD	W	ENTRY	1472	24	1246	1299	
	STL	N	SET W	1299	20	1653	1556	
	SLO	8001	STORE N	1556	16	8001	1913	
	AUP	8003	CALC AO	1913	10	8003	1571	
	SRT	0003	AND PO	1571	30	0003	1479	
	STU	X		1479	21	1284	1287	
	SUP	8001		1287	11	8001	1245	
	SLT	0001		1245	35	0001	1451	
	STU	P		1451	21	1606	1609	
	SUP	8001		1609	11	8001	1367	
	SLT	0002		1367	35	0002	1473	
	AUP	8003		1473	10	8003	1281	
	AUP	X		1281	10	1284	1239	
	ALO	P		1239	15	1606	1711	
	SLT	0004		1711	35	0004	1621	
	ALO	C1		1621	15	1506	1761	
	STL	B0002		1761	20	0901	1554	
	LDD	C2		1554	69	1188	1291	
	SDA	SPR		1291	22	1231	1334	
	AUP	C3	SU	1334	10	1337	1423	
SU	STU	SA		1423	21	1401	1604	
	AUP	C4	8003		1604	10	1558	8003
8003	RAL	9992	B0002		8003	65	9992	0901
B0002	SLT	0009	XX		0901	35	0009	1521
XX	STU	UH			1521	21	1176	1429
	SLT	0001			1429	35	0001	1185
	RAL	8002			1185	65	8002	1293

	AUP	W			1293	10	1246	1351
	SRT	0001			1351	30	0001	1907
	AUP	UH	SPR		1907	10	1176	1231
SPR	SRT	0009	SA		1231	30	0009	1401
SA	STL	9983	B0001		1401	20	9983	0900
B0001	RAL	N		IS N ZERO	0900	65	1653	1308
	NZE		EXITY		1308	45	1362	1813
	SLO	1I		REDUCE N	1362	16	1565	1669
	STL	N		BY 1	1669	20	1653	1456
	RAU	SA		IS AI	1456	60	1401	1705
	SUP	A197		IN LAST	1705	11	1358	1863
	BMI		TP	BLOCK OF	1863	46	1266	1317
	AUP	A201	SU	4	1266	10	1719	1423
TP	RAL	SPR		IS P 9	1317	65	1231	1235
	SLO	C2			1235	16	1188	1343
	NZE		ZP		1343	45	1296	1297
	ALO	C5		STEP P	1296	15	1249	1703
	LDD	C1			1703	69	1506	1509
	SDA	B0002			1509	22	0901	1454
	STL	SPR			1454	20	1231	1184
	RAU	SA		REDUCE AI	1184	60	1401	1755
	SUP	C6	SU	BY 196	1755	11	1408	1423
ZP	LDD	C1		SET P	1297	69	1506	1559
	STD	B0002		TO ZERO	1559	24	0901	1504
	LDD	C7			1504	69	1458	1661
	STD	SPR			1661	24	1231	1234
	RAU	SA		REDUCE AI	1234	60	1401	1805
	SUP	C8	SU	BY 195	1805	11	1508	1423
C1	SLT	0000	XX	CONSTANTS	1506	35	0000	1521
C2	SRT	0009	SA		1188	30	0009	1401
C3	STL	A0001	B0001		1337	20	0600	0900
C4	45	0000	0001		1558	45	0000	0001
1I	00	0000	0001		1565	00	0000	0001
A197	STL	A0197	B0001		1358	20	0796	0900
A201	STL	A0201	B0001		1719	20	0800	0900
C5	SRT	0010	SA		1249	30	0010	1401
C6	00	0196	0000		1408	00	0196	0000
C7	SRT	0000	SA		1458	30	0000	1401
C8	00	0195	0000		1508	00	0195	0000
Z	00	0000	0000		1769	00	0000	0000
W	01	0000	W	ERASEABLE	1246	01	0000	1246
N	01	0000	N		1653	01	0000	1653
B0002	01	0000	B0002		0901	01	0000	0901
SPR	01	0000	SPR		1231	01	0000	1231
SA	01	0000	SA		1401	01	0000	1401
UH	01	0000	UH		1176	01	0000	1176
X	01	0000	X		1284	01	0000	1284
	HED							

SUB 3 PROCESS OP CODE

SUBR3	STD	EXITX		STORE EXIT	0996	24	1196	1349
	RAL	1954			1349	65	1954	1659
	SRT	0004		IS OP	1659	30	0004	1819
	NZE		ILLOP	BLANK	1819	45	1522	1013
	SLT	0006		IS OP	1522	35	0006	1387
	NZU		NUM	NUMERIC	1387	44	1341	1192
	SRT	0002			1341	30	0002	1347
	STL	XXXX1		SEARCH	1347	20	1961	1414
	SLO	8001		SYMBOLIC	1414	16	8001	1671
	TLU	C0001		OP TABLE	1671	84	1050	1855
	ALO	1	8002		1855	15	1608	8002
8002	RAL	9997	NEXT	IS	8002	65	9997	1501
NEXT	SLO	XXXX1		SYM OP	1501	16	1961	1615
	SLT	0006		LEGAL	1615	35	0006	1529
	NZU	ILLOP			1529	44	1013	1384
	SLT	0001		TEST FOR	1384	35	0001	1391
	NZU	PSEU		PSEUDO OP	1391	44	1295	1346
SNO	SLT	0001	SNO	STORE NUM	1346	35	0001	1753
	STL	P0007		650 OP	1753	20	1983	1236
	SRT	0004		GET OPTIM	1236	30	0004	1397
	ALO	3	8002	CONSTANTS	1397	15	1650	8002
8002	LDD	9996	NEX		8002	69	9996	1399
NEX	STD	OPTIM	EXITX		1399	24	1452	1196
NUM	SLO	4		FORM	1192	16	1345	1449
	SLT	0002		NUMERIC	1449	35	0002	1905
	SLO	8002		OP	1905	16	8002	1963
	SRT	0001		CODE	1963	30	0001	1869
	ALO	8001	SNO		1869	15	8001	1753
ILLOP	RAL	P0010		ILLEGAL OP	1013	65	1986	1441
	ALO	5		SET 8	1441	15	1194	1499
	STL	P0010		FOR BLANK	1499	20	1986	1289
	LDD	6	NEX	OP	1289	69	1242	1399
PSEU	SLT	0003		PSEUDO	1295	35	0003	1803
	ALO	EXITX		OP	1803	15	1196	1551
	ALO	1IXXX	8002		1551	15	0992	8002
1	RAL	0000	NEXT	CONSTANTS	1608	65	0000	1501
2	10	0000	0000		1700	10	0000	0000
3	LDD	N0001	NEX		1650	69	0800	1399
4	90	9000	0000		1345	90	9000	0000
5	00	0000	0008		1194	00	0000	0008
6	55	0505	0999		1242	55	0505	0999
	HED							

SUB 4 WHAT IS ADDRESS

SUBR4	HED	4						
	STD	EXITZ		STORE EXIT	1213	24	1416	1919
	STU	XXXX1		STORE ADDR	1919	21	1961	1464
	SRT	0008		IS C5	1464	30	0008	1333
	NZU		EXITZ	BLANK	1333	44	1437	1416
	STU	XXXX2		SAVE C5	1437	21	1962	1665

	RAU	8002		ARE LOW 4	1665	60	8002	1523
	AUP	90XXX		CHARACTERS	1523	10	1178	1383
	BOV	LOOP	LOOP	NUMERIC	1383	47	1286	1286
LOOP	AUP	H1			1286	10	1339	1393
	BOV		SYM		1393	47	1396	0998
	SLT	0002			1396	35	0002	1853
	NZU	LOOP			1853	44	1286	1658
	RAL	XXXX2		LOW 4 NUM	1658	65	1962	1417
	SLO	90XXX		IS C5	1417	16	1178	1433
	BMI		SYM	ALPHABETIC	1433	46	1336	0998
	ALO	29XXX			1336	15	1189	1443
	BMI	SYM			1443	46	0998	1447
	RAL	1IXXX		REGIONAL	1447	65	0992	1497
	AUP	XXXX1	AEX		1497	10	1961	1715
AEX	ALO	EXITZ			1715	15	1416	1721
	SLO	8002	8001		1721	16	8002	8001
SYM	RAU	XXXX1		HED SYMBOL	0998	60	1961	1765
	LDD		SUB15		1765	69	1018	1663
	ALO	2IXXX		STEP EXIT	1018	15	1771	1425
	STU	HSYMB	AEX		1425	21	1230	1715
H1	10	0000	0000		1339	10	0000	0000
HSYMB	01	0000	HSYMB	ERASEABLE	1230	01	0000	1230
	HED							

SUB 5 TEST ABSOLUTE ADDRESS

	HED	5						
SUBR5	STD	EXITZ		STORE EXIT	1750	24	1416	1969
	STL	XXXX1		STORE A	1969	20	1961	1514
	SLT	0004			1514	35	0004	1475
	SLO	8002			1475	16	8002	1483
	TLU	R0001		TLU	1483	84	1200	1656
	ALO	GET	8002	RANGE	1656	15	1709	8002
8002	RAU	9972	X	TABLE	8002	60	9972	1177
X	SLT	0006			1177	35	0006	1491
	SRT	0006			1491	30	0006	1706
	AUP	EXITZ			1706	10	1416	1821
	ALO	XXXX1			1821	15	1961	1815
	SUP	8003	8001		1815	11	8003	8001
GET	RAU	0000	X		1709	60	0000	1177
R0001	00	1999	0000	RANGE	1200	00	1999	0000
R0002	00	7999	0003	TABLE	1201	00	7999	0003
R0003	00	8003	0001		1202	00	8003	0001
R0004	00	8004	0003		1203	00	8004	0003
R0005	00	8007	0001		1204	00	8007	0001
R0006	00	8999	0003		1205	00	8999	0003
R0007	00	9059	0002		1206	00	9059	0002
R0008	99	9999	0003		1207	99	9999	0003
	HED							

SUB 6 SYMBOL TEST ROUTINE

	HED	6						
SUBR6	STD	EXITZ		STORE EXIT	1275	24	1416	1320
	STU	S		STORE SYMB	1320	21	1601	1654

	LDD	SWOF		SET SWITCH	1654	69	1758	1861	
	STD	SW		OFF	1861	24	1610	1764	
	MPY	SC		SCRAMBLE	1764	19	1517	1238	
	RAL	8002		SYMBOL	1238	65	8002	1547	
	SLT	0003		GIVING	1547	35	0003	1806	
	MPY	CF		LO	1806	19	1759	1330	
	SRT	0001			1330	30	0001	1587	
	SLT	0004		SET	1587	35	0004	1597	
	ALO	L1P1		TEST	1597	15	1850	1856	
	STL	LOP1		WORD	1856	20	1865	1168	
	SLO	1DXXX			1168	16	1020	1525	
	STL	LO	SLI	SAVE LO	1525	20	1903	1487	
SLI	STL	LI	8001	INIT LI	1487	20	1614	8001	
8001	RAL	9990	A	IS CONT OF	8001	65	9990	1395	
A	NZE		UND	LI ZERO	1395	45	1098	1549	
	SLO	S		IS CONT OF	1098	16	1601	1756	
	NZE	SW	DEF	LI SYMB	1756	45	1610	1811	
SW	RAL	LI	OFF	IS LI	1610	65	1614	1270	
OFF	SLO	LMAX		MAXIMUM	1270	16	1573	1227	
	NZE		MAX		1227	45	1280	1331	
	ALO	LMP1	SLI	STEP LI	1280	15	1533	1487	
ON	SLO	LO		IS LI EQUA	1800	16	1903	1708	
	NZE		SWSYM	TO LO	1708	45	1412	1664	
	ALO	LOP1	SLI	STEP LI	1412	15	1865	1487	
MAX	LDD	SWON		SET	1331	69	1434	1537	
	STD	SW		SWITCH ON	1537	24	1610	1714	
	RAL	L1	SLI	ZERO LI	1714	65	1467	1487	
UND	RAL	EXITZ		UNDEFINED	1549	65	1416	1871	
	ALO	1IXXX		STEP	1871	15	0992	1647	
	STL	EXITZ		EXIT	1647	20	1416	1370	
	RAL	LI			1370	65	1614	1420	
	SLO	L1			1420	16	1467	1921	
	STL	LSYMB	ASU		1921	20	1575	1278	
DEF	RAL	LI		DEFINED	1811	65	1614	1470	
	LDD	ASU	SUBR8	GET	1470	69	1278	1381	
ASU	AUP	S	EXITZ	EQUIVALENT	1278	10	1601	1416	
SWSYM	HLT	0111	SYMOF	S TAB FULL	1664	01	0111	1218	
SYMOF	LDD	SYMON		SET SW SYM	1218	69	1971	1224	
	STD	SWSYM	SYMON	ON	1224	24	1664	1971	
SYMON	RAL	EXITZ			1971	65	1416	1572	
	ALO	2IXXX	8002		1572	15	1771	8002	
SWOF	RAL	LI	OFF	CONSTANTS	1758	65	1614	1270	
SWON	RAL	LI	ON		1434	65	1614	1800	
SC	10	0100	1001			1517	10	0100	1001
CF	00	0000	0004			1759	00	0000	0004
L1	RAL	0000	A			1467	65	0000	1395
L1P1	RAL	0001	A			1850	65	0001	1395
LMAX	RAL	E0000	A			1573	65	0399	1395
LMP1	RAL	E0001	A			1533	65	0400	1395
S	01	0000	S	ERASEABLE	1601	01	0000	1601	
SW	01	0000	SW		1610	01	0000	1610	

LO	01	0000	LO		1903	01	0000	1903
LI	01	0000	LI		1614	01	0000	1614
LSYMB	01	0000	LSYMB	HED	1575	01	0000	1575

SUB 7 STORE SYMBOL AND EQUIV

	HED	7						
SUBR7	STD	EXITZ		STORE EXIT	1279	24	1416	1520
	LDD	SS			1520	69	1623	1226
	SDA	XXXX2	8001	STORE	1226	22	1962	8001
8001	STU	9988	A	SYMBOL	8001	21	9988	1541
A	SUP	8001			1541	11	8001	1599
	STL	XXXX1		SAVE E	1599	20	1961	1814
	SRT	0004			1814	30	0004	1625
	DIV	2IXXX		CALCULATE	1625	14	1771	1232
	SLT	0004		LOCATION	1232	35	0004	1493
	ALO	C1		OF	1493	15	1446	1651
	LDD	C2		E	1651	69	1704	1808
	SDA	SEL		AND	1808	22	1911	1864
	LDD	C3		STORE	1864	69	1617	1620
	SDA	SER			1620	22	1673	1276
	STL	OBEE			1276	20	1431	1484
	SLO	8001			1484	16	8001	1591
	ALO	XXXX1	OBEE		1591	15	1961	1431
OBEE	LDD	9986	B		1431	69	9986	1389
B	NZU	SER			1389	44	1673	1244
	SLT	0004	SEL		1244	35	0004	1911
SS	STU	9987	A	CONSTANTS	1623	21	9987	1541
C1	LDD	E0001	B		1446	69	0400	1389
C2	SDA	9985	EXITZ		1704	22	9985	1416
C3	SIA	9984	EXITZ		1617	23	9984	1416
OBEE	O1	0000	OBEE	ERASEABLE	1431	01	0000	1431
SEL	O1	0000	SEL		1911	01	0000	1911
SER	O1	0000	SER		1673	01	0000	1673
	HED							

SUB 8 GIVEN SYM LOC GET EQUIV

	HED	8						
SUBR8	STD	EXIT		STORE EXIT	1381	24	1534	1637
	LDD	8003			1637	69	8003	1294
	SDA	LSYMB		STORE L	1294	22	1575	1328
	RAL	8001			1328	65	8001	1285
	DIV	2DXXX			1285	14	1288	1148
	SLT	0004			1148	35	0004	1809
	ALO	E1	8002		1809	15	1462	8002
8002	LDD	9975	TR		8002	69	9975	1378
TR	NZU	LH			1378	44	1481	1282
	RAM	8001	LR	GET E RH	1481	67	8001	1439
LH	RAM	8001		GET E LH	1282	67	8001	1489
	SRT	0004	LR		1489	30	0004	1439
LR	LDD	8003			1439	69	8003	1496
	SIA	EQUIV			1496	23	1223	1326
	RAL	LSYMB	EXIT		1326	65	1575	1534

E1	LDD	E0001	TR	CONSTANT	1462	69	0400	1378
EXIT	01	0000	EXIT	ERASEABLE	1534	01	0000	1534
EQUIV	01	0000	EQUIV		1223	01	0000	1223
	HED							
SUB 9 CALCULATE REG ADDRESS								
	HED	9						
SUBR9	STD	EXITZ		STORE EXIT	1174	24	1416	1670
	SRT	0008			1670	30	0008	1539
	RAL	8003			1539	65	8003	1697
	SLT	0004			1697	35	0004	1858
8002	ALO	C	8002	IS	1858	15	1512	8002
NEX	RAL	9991	NEX	REGION	8002	65	9991	1445
	BDO	ILL		DEFINED	1445	90	1649	1701
	SLT	0001			1701	35	0001	1908
	RAL	8002			1908	65	8002	1667
	SRT	0001			1667	30	0001	1723
	LDD	EXITZ			1723	69	1416	1720
	BD6	L			1720	96	1773	1675
	BD5	D	I		1773	95	1376	1428
L	ALO	1957	ALL		1675	15	1957	1562
D	ALO	1958	ALL		1376	15	1958	1562
I	ALO	1959	ALL		1428	15	1959	1562
ALL	SLO	IIXXX			1562	16	0992	1747
-ILL	BMI	ILL	EXITZ		1747	46	1649	1416
	RAL	EXITZ		UNDEFINED	1649	65	1416	1622
	ALO	IIXXX	8002	OR ILLEG	1622	15	0992	8002
C	RAL	0960	NEX	CONSTANT	1512	65	0960	1445
	HED							
SUB 10 SET CC 8 AND PUNCH								
SUB10	ALO	P0010			1405	15	1986	1641
	ALO	84TH			1641	15	1344	1699
	STL	P0010			1699	20	1986	1589
	PCH	P0001	8003		1589	71	1977	8003
84TH	00	0800	0000	CONSTANT	1344	00	0800	0000
SUB 11 FIND AND RESERVE BEST								
	HED	O						
SUB11	LDD	DRUMT		TEST	1801	69	1212	1965
	BDO		SEX	DRUM TAG	1965	90	1770	1820
	STL	EXIT		IS ADR L	1770	20	1725	1478
	BD6	DI			1478	96	1531	1633
DI	RAL	8003	SSW	L	1633	65	8003	1741
	LDD	F		D OR I	1531	69	1634	1787
	SDA	XXXX1	SUB13	FIND OPTIM	1787	22	1961	1964
GDA	SLT	0004		DYNAM ADR	1851	35	0004	1762
	ALO	8002			1762	15	8002	1672
	ALO	8002	SSW		1672	15	8002	1741
SSW	LDD	OFF		SET SW OFF	1741	69	1444	1797
	STD	SW			1797	24	1584	1837

	ALO	A1P1		GET TABLE	1837	15	1290	1495
	STL	BP1		START ADDR	1495	20	1915	1268
	SLO	1DXXX			1268	16	1020	1775
	STL	AO	SAI	SAVE AO	1775	20	1804	1612
SAI	STL	AI	8001		1612	20	1687	8001
8001	RAU	9975	TA	IS A LOCAT	8001	60	9975	1579
TA	NZU	YES	SW	AVAILABLE	1579	44	1583	1584
SW	RAU	AI	SWOF		1584	60	1687	1691
SWOF	SLO	AMAX		IS AI MAX	1691	16	1394	1749
	NZE		MAX		1749	45	1502	1754
	ALO	AMP1	SAI	STEP AI	1502	15	1906	1612
SWON	SLO	AO		IS AI EQUA	1751	16	1804	1859
	NZE		FULL	TO AO	1859	45	1662	1914
	ALO	BP1	SAI	STEP AI	1662	15	1915	1612
MAX	LDD	ON		SET SW ON	1754	69	1909	1712
	STD	SW			1712	24	1584	1737
	RAL	A1	SAI	SET AI TO	1737	65	1240	1612
FULL	HLT	0222		DRUM PAKED	1914	01	0222	1318
	LDD	H9XXX		SET DRUM	1318	69	1216	1870
	STD	DRUMT		TAG TO 9	1870	24	1212	1466
	RAL	EXIT		STEP EXIT	1466	65	1725	1820
SEX	ALO	1IXXX	8002		1820	15	0992	8002
YES	SCT	0000		RESERVE	1583	36	0000	1660
	SLT	0001		LOCATION	1660	35	0001	1717
	SRT	0001		FOUND	1717	30	0001	1823
	STL	XXXX2		SAVE P-	1823	20	1962	1516
	SUP	8003			1516	11	8003	1873
	SLT	0004			1873	35	0004	1683
	AUP	8001			1683	10	8001	1791
	ALO	SRP	8002		1791	15	1494	8002
8002	SRT	0009	SCA	NULIFY SCT	8002	30	0009	1722
SCA	RAU	8003			1722	60	8003	1629
	ALO	AI			1629	15	1687	1841
	LDD	ST			1841	69	1544	1847
	SDA	ERAS	8001		1847	22	1901	8001
8001	STU	9972	CA		8001	21	9972	1825
CA	SUP	8001		CALCULATE	1825	11	8001	1733
	SLO	A1		LOCATION	1733	16	1240	1545
	SRT	0004		FOUND	1545	30	0004	1710
	DIV	4I			1710	14	1616	1426
	MPY	500			1426	19	1679	1552
	ALO	8003			1552	15	8003	1760
	SUP	8001			1760	11	8001	1767
	AUP	XXXX2			1767	10	1962	1817
	MPY	50			1817	19	1920	1340
	ALO	8003			1340	15	8003	1897
	SUP	8001	EXIT		1897	11	8001	1725
F	00	0000	GDA	CONSTANTS	1634	00	0000	1851
OFF	RAL	AI	SWOF		1444	65	1687	1691
A1P1	RAU	A0002	TA		1290	60	0601	1579
AMAX	RAU	A0200	TA		1394	60	0799	1579
AMP1	RAU	A0201	TA		1906	60	0800	1579
ON	RAL	AI	SWON		1909	65	1687	1751

A1	RAU	A0001	TA		1240	60	0600	1579
SRP	SRT	0000	SCA		1494	30	0000	1722
ST	STU	9999	CA		1544	21	9999	1825
4I	00	0000	0004		1616	00	0000	0004
500	00	0000	0500		1679	00	0000	0500
50	00	0000	0050		1920	00	0000	0050
FON	RAL	8001	STEPX		1602	65	8001	1810
AO	01	0000	AO	ERASEABLE	1804	01	0000	1804
AI	01	0000	AI		1687	01	0000	1687
BP1	01	0000	BP1		1915	01	0000	1915
EXIT	01	0000	EXIT		1725	01	0000	1725
	HED							

SUB 12 SET BLANK L 8

SUB12	ALO	P0010			0991	15	1986	1891
	ALO	87TH			1891	15	1594	1799
	STL	P0010	8003		1799	20	1986	8003
87TH	00	0000	8000	CONSTANT	1594	00	0000	8000

SUB 13 CAL OPTIMUM DYNAMIC ADR

SUB13	HED	Y						
	STD	EXITZ						
	BD5		I	STORE EXIT	1964	24	1416	1970
	RAL	OPTIM		D OR I	1970	95	1923	1875
	SLT	0001		D	1923	65	1452	1860
	SLO	8002			1860	35	0001	1867
	SLT	0001			1867	16	8002	1925
	ALO	8001			1925	35	0001	1581
	SRT	0003	SEO		1581	15	8001	1639
I	RAL	OPTIM		I	1639	30	0003	1947
	BD2	SHOP		SHIFT TEST	1875	65	1452	1910
	BD3	XAS		XAS TEST	1910	92	1666	1716
	SLT	0002	SEO		1716	93	1772	1822
SEO	STL	XXXX1		SAVE EV OD	1822	35	0002	1947
	RAL	OPREG		IS BASE	1947	20	1961	1766
	DIV	2IXXX		EVEN OR OD	1766	65	1872	1277
	RAU	8003			1277	14	1771	1332
	ALO	XXXX1			1332	60	8003	1689
	NZU		R8		1689	15	1961	1816
	SLT	0002			1816	44	1922	1972
	RAL	8002	R8		1922	35	0002	1729
R8	SRT	0008		GET BASE	1729	65	8002	1972
	ALO	OPREG		PLUS DELTA	1972	30	0008	1941
	DIV	50		MOD 50	1941	15	1872	1327
	RAL	8003	EXITZ		1327	14	1380	1390
					1390	65	8003	1416
SHOP	RAL	P0007		TLU UNITS	1666	65	1983	1887
	SRT	0004		DIGIT OF	1887	30	0004	1997
	SLT	0009		D ADDRESS	1997	35	0009	1917
	SLO	8002			1917	16	8002	1975
	TLU	U0001			1975	84	1250	1812
	ALO	GU	8002		1812	15	1866	8002

8002	RAL	9969	TOP		8002	65	9969	1973
TOP	SLT	0002			1973	35	0002	1779
	LDD	OPTIM		IS OP SRD	1779	69	1452	1862
	BD3		SEO		1862	93	1916	1947
	SLT	0004	SEO	SRD	1916	35	0004	1947
XAS	RAL	P0007		TLU	1772	65	1983	1937
	SRT	0004		D ADDRESS	1937	30	0004	1198
	SLT	0006			1198	35	0006	1966
	SLO	8002			1966	16	8002	1476
	TLU	I0001			1476	84	1300	1912
	ALO	GXD	8002		1912	15	1967	8002
8002	RAL	9968	NXT		8002	65	9968	1274
NXT	SLT	0004	SEO		1274	35	0004	1947
U0001	00	2322	2524	SHIFT	1250	00	2322	2524
U0002	10	0706	0706	OPTIMIZING	1251	10	0706	0706
U0003	20	0706	0908	TABLE	1252	20	0706	0908
U0004	30	0908	1110		1253	30	0908	1110
U0005	40	1110	1312		1254	40	1110	1312
U0006	50	1312	1514		1255	50	1312	1514
U0007	60	1514	1716		1256	60	1514	1716
U0008	70	1716	1918		1257	70	1716	1918
U0009	80	1918	2120		1258	80	1918	2120
U0010	90	2120	2322		1259	90	2120	2322
I0001	19	9906	0600	X ACCUM	1300	19	9906	0600
I0002	79	9907	0700	ADD SUB	1301	79	9907	0700
I0003	80	0008	0800	TABLE	1302	80	0008	0800
I0004	80	0106	0600		1303	80	0106	0600
I0005	80	0209	0800		1304	80	0209	0800
I0006	80	0308	0900		1305	80	0308	0900
I0007	90	5908	0800		1306	90	5908	0800
I0008	99	9909	0900		1307	99	9909	0900
GU	RAL	0000	TOP	CONSTANTS	1866	65	0000	1973
GXD	RAL	0000	NXT		1967	65	0000	1274
50	00	0000	0050		1380	00	0000	0050
	HED							

SUB 14 PUNCH AVAIL TABLE

SUB14	HED	P						
	STD	EXITX		STORE EXIT	1403	24	1196	1849
	LDD	85TH		SET PUNCH	1849	69	1652	1368
	STD	P0010		8	1368	24	1986	1739
	LDD	A1		INIT AI	1739	69	1292	1595
	STD	XXXX1			1595	24	1961	1418
	RAL	RS1	LOOP	INIT RS	1418	65	1324	1829
LOOP	AUP	C1	8001		1829	10	1382	8001
8001	STL	9999	NEXT	SET LOCAT	8001	20	9999	1702
NEXT	SUP	TW1			1702	11	1468	1374
	NZU		SRS		1374	44	1377	1528
	AUP	C2			1377	10	1430	1335
	ALO	C3	8003		1335	15	1338	8003

SRS	STL	XXXX2		SAVE RS	1528	20	1962	1518
	RAU	C4			1518	60	1424	1879
	ALO	XXXX1	8002		1879	15	1961	8002
8002	LDD	9998	8003	STORE	8002	69	9998	8003
8003	STD	9997	TP	AV TABLE	8003	24	9997	1752
TP	SUP	TW2			1752	11	1618	1474
	NZU		PUN		1474	44	1427	1578
	AUP	C5			1427	10	1480	1385
	ALO	C6	8002		1385	15	1388	8002
PUN	PCH	P0001			1578	71	1977	1477
	SLO	AIM		IS AI MAX	1477	16	1530	1435
	NZE		EXITX		1435	45	1438	1196
	ALO	AMP1		STEP AI	1438	15	1342	1248
	STL	XXXX1			1248	20	1961	1668
	RAL	XXXX2			1668	65	1962	1718
	SLO	C7	LOOP		1718	16	1524	1829
C1	STL	P0001	NEXT	CONSTANTS	1382	20	1977	1702
C2	STL	P0009	NEXT		1430	20	1985	1702
C3	00	0500	0500		1338	00	0500	0500
C4	STD	P0002	TP		1424	24	1978	1752
C5	STD	P0010	TP		1480	24	1986	1752
C6	00	0001	0000		1388	00	0001	0000
C7	00	1499	1499		1524	00	1499	1499
TW1	STL	P0007	NEXT		1468	20	1983	1702
TW2	STD	P0008	TP		1618	24	1984	1752
85TH	00	0080	0000		1652	00	0080	0000
A1	LDD	A0001	8003		1292	69	0600	8003
RS1	00	0000	0450		1324	00	0000	0450
AIM	LDD	A0200	8003		1530	69	0799	8003
AMP1	LDD	A0201	8003		1342	69	0800	8003
	HED							

SUB 15 HEADING ROUTINE

SUB15	HED	H						
	STD	EXITY		SAVE EXIT	1663	24	1813	1768
	LDD	8003		IS C1	1768	69	8003	1574
	SLT	0008		BLANK	1574	35	0008	1543
	NZU	DH			1543	44	1298	1348
	RAU	8001		HEAD	1348	60	8001	1818
	AUP	0000H	EXITY		1818	10	1329	1813
DH	RAU	8001	EXITY	DONT HEAD	1298	60	8001	1813
	HED							

SUB 16 INITIALIZATION

SUB16	STD	EXITY		STORE EXIT	1353	24	1813	1868
	RAL	I1			1868	65	1624	1929
	STU	P0009		ZERO CARD	1929	21	1985	1488
	STD	0000H		ZERO HEAD	1488	24	1329	1432
	LDD		SUBR1	ZERO SYM T	1432	69	1485	1600
	RAL	I2		MAKE DRUM	1485	65	1538	1593
	AUP	I3		AVAILABLE	1593	10	1546	1802
	LDD		SUBR1		1802	69	1918	1600

LDD	H8XXX		SET DRUM	1918	69	1406	1968	
STD	DRUMT		TAG TO 8	1968	24	1212	1674	
LDD	I4		SET SWSYM	1674	69	1527	1580 *	
STD	SWSYM		OFF	1580	24	1664	1724	
RAL	I5		UNDEFINE	1724	65	1577	1631	
AUP	H8XXX		REGIONS	1631	10	1406	1774	
LDD	EXITY	SUBR1		1774	69	1813	1600	
I1	00	S0001	E0200	CONSTANTS	1624	00	0000	0599
I2	00	A0001	A0200		1538	00	0600	0799
I3	11	1111	1111		1546	11	1111	1111
I4	01	0111	SYMOF		1527	01	0111	1218
I5	00	1021	1049		1577	00	1021	1049
PACOF	BD6	FINDI	FINDL		1852	96	1824	1874

SUB 17 CALC 800X EQUIVALENT

	HED	X						
SB17D	LDD	DEQ	DI	D EQUIV	1902	69	1924	1627
SB17I	LDD	IEQ	DI	I EQUIV	1854	69	1974	1627
DI	STD	XXXX1		SAVE TAG	1627	24	1961	1526
	STU	EXITY		STORE EXIT	1526	21	1813	1576
	STL	XXXX2		STORE 800X	1576	20	1962	1626
	LDD	XXXX1	SUB13	GET OPTIM	1626	69	1961	1964
SDA	STL	XXXX1		DYNAM ADDR	1904	20	1961	1676
	DIV	2IXXX		IS DY ADDR	1676	14	1771	1482
	NZU	OD	EV	EVEN OR OD	1482	44	1535	1386
OD	RAL	8002	EO		1535	65	1588	1643
EV	RAL	8003	EO		1386	65	1789	1643
EO	SLO	XXXX2			1643	16	1962	1726
	NZE		ADD1		1726	45	1630	1681
	RAL	XXXX1	EXITY		1630	65	1961	1813
ADD1	RAL	XXXX1			1681	65	1961	1776
	ALO	1IXXX	EXITY		1776	15	0992	1813
DEQ	00	0008	SDA	CONSTANTS	1924	00	0008	1904
IEQ	00	0009	SDA		1974	00	0009	1904
8002	00	0000	8002		1588	00	0000	8002
8003	00	0000	8003		1789	00	0000	8003
	HED							

SUB 18 INDEX ADDRESS

	HED	W						
INDEX	STD	EXITY		STORE EXIT	1826	24	1813	1876
	STL	XXXX1		STORE ADDR	1876	20	1961	1926
	RAU	1954			1926	60	1954	1976
	SLT	0007			1976	35	0007	1693
	LDD	EXITY			1693	69	1813	1677
	BD5	D	D OR I		1677	95	1680	1532
D	SLT	0002	D		1532	35	0002	1680
	SRT	0009			1680	30	0009	1727
	MPY	2000I			1727	19	1014	1684
	LDD	EXITY			1684	69	1813	1777
	BD6	DS	DRUM O COR		1777	96	1730	1582

DS	SRT ALO	0001 XXXX1	DS EXITY	CORE	1582 1730	30 15	0001 1961	1730 1813
EXITY	01 HED	0000	EXITY	ERASEABLE	1813	01	0000	1813

SUB 19 PROCESS LOCATION

	REG	D0923	0923	PROC L				
	REG	G0925	0925					
	REG	J0928	0928					
	REG	K0931	0931					
	REG	L0933	0933					
	HED	A						
PROCL	STD	EXITX		STORE EXIT	0969	24	1196	1899
	RAU	1951		IS L BLANK	1899	60	1951	1827
	NZU	NB			1827	44	1731	1632
	LDD	DRUMT		BLANK TEST	1632	69	1212	1877
	BDO	B	D0002	DRUM TAG	1877	90	1781	0924
B	RAL	OPREG	SETL	BLANK	1781	65	1872	1927
SETL	SLT	0004		SET L	1927	35	0004	1987
	LDD	P0008		TO OR	1987	69	1984	1638
	SDA	P0008	EXITX		1638	22	1984	1196
NB	LDD	G	SUBR4	WHAT IS L	1731	69	1734	1213
G0001	RAL	1957	K0001	ABSOLUTE	0925	65	1957	0931
G0002	LDD	K	SUBR9	REGIONAL	0926	69	1780	1174
G0003	LDD	J	SUBR6	SYMBOLIC	0927	69	1830	1275
K0001	LDD	L	SUBR5	TEST RANGE	0931	69	1784	1750
K0002	RAU	87TH	X	REG ERROR	0932	60	1585	1839
X	AUP	P0010		SET BLANK	1839	10	1986	1392
	STU	P0010	SETL	L8	1392	21	1986	1927
L0001	STL	OPREG	SETL	DRUM ADDR	0933	20	1872	1927
L0002	AUP	86TH	800X	800X	0934	10	1688	1743
L0003	LDD	ORCEQ	89X	CORE ADDR	0935	69	1738	1442
L0004	RAU	87TH	X	OTHER	0936	60	1585	1839
800X	LDD	ORXEQ	89X		1743	69	1596	1442
89X	STD	OPREG	X		1442	24	1872	1839
J0001	RAL	EQUIV	K0001	SYM DEFIND	0928	65	1223	0931
J0002	RAL	D	SUB11	SYM UNDEF	0929	65	1682	1801
J0003	RAU	87TH	X	S TAB FULL	0930	60	1585	1839
D0001	STL	OPREG	SS	S EQ FOUND	0923	20	1872	1628
D0002	RAU	87TH	X	DRUM PAKED	0924	60	1585	1839
SS	AUP	HSYMB		STORE	1628	10	1230	1635
	ALO	LSYMB		SYMBOL	1635	15	1575	1880
	LDD	B	SUBR7		1880	69	1781	1279
G	00	0000	G0001	CONSTANTS	1734	00	0000	0925
K	00	0090	K0001		1780	00	0090	0931
J	00	0000	J0001		1830	00	0000	0928
L	00	0000	L0001		1784	00	0000	0933
D	00	0090	D0001		1682	00	0090	0923
86TH	00	0008	0000		1688	00	0008	0000
87TH	00	0000	8000		1585	00	0000	8000
EXITX	01	0000	EXITX	ERASEABLE	1196	01	0000	1196

HED

SUB 20 PROCESS DATA ADDRESS

	REG	D0937	0937	PROC D				
	REG	F0939	0939					
	REG	G0941	0941					
	REG	J0944	0944					
	REG	K0946	0946					
	REG	L0949	0949					
	HED	B						
PROCD	STD	EXITX		STORE EXIT	0972	24	1196	1949
	RAU	1952		IS D BLANK	1949	60	1952	1678
	NZU	NB			1678	44	1831	1732
	RAL	D	SUB11	FIND	1732	65	1685	1801
D0001	STL	ORBAL	L0004	OPTIMUM D	0937	20	1492	0952
D0002	RAU	88TH	X	DRUM PAKED	0938	60	1542	1398
BT	LDD	OPTIM		IS OP	1728	69	1452	1778
	BD1	EXITX		A BRANCH	1778	91	1196	1783
	STL	OPREG	EXITX	D TO OR	1783	20	1872	1196
X	AUP	P0010		SET BLANK	1398	10	1986	1592
	STU	P0010	EXITX		1592	21	1986	1196
NB	LDD	G	SUBR4	WHAT IS D	1831	69	1834	1213
G0001	RAL	1958	J0001	ABSOLUTE	0941	65	1958	0944
G0002	LDD	J	SUBR9	REGIONAL	0942	69	1645	1174
G0003	LDD	K	SUBR6	SYMBOLIC	0943	69	1646	1275
J0001	LDD	L	SUBR5	TEST RANGE	0944	69	1448	1750
J0002	RAU	88TH	X	REG ERROR	0945	60	1542	1398
L0001	LDD	AXD	INDEX	DRUM ADDR	0949	69	1828	1826
L0002	SLT	0004	800X	800X	0950	35	0004	1878
L0003	LDD	AXC	INDEX	CORE ADDR	0951	69	1928	1826
L0004	SLT	0004	SD	OTHER	0952	35	0004	1930
800X	LDD	P0007		SET 800X D	1878	69	1983	1436
	SDA	P0007			1436	22	1983	1486
	SRT	0004			1486	30	0004	1498
	AUP	XEQ	SB17D	GET 800X E	1498	10	1881	1902
SCD	SLT	0004		SET CORE D	1931	35	0004	1642
	LDD	P0007			1642	69	1983	1536
	SDA	P0007			1536	22	1983	1586
	SRT	0004			1586	30	0004	1548
	LDD	CEQ	SUB13	GET CORE E	1548	69	1782	1964
SSEQ	STL	ORCEQ	BT	SAVE CORE	1832	20	1738	1728
K0001	RAL	EQUIV	J0001	SYM DEFIND	0946	65	1223	0944
K0002	RAL	F	SUB11	SYM UNDEF	0947	65	1882	1801
K0003	RAU	88TH	X	S TAB FULL	0948	60	1542	1398
F0001	STL	EQ	SS	SAVE EQ	0939	20	1793	1696
F0002	RAU	88TH	X	DRUM PAKED	0940	60	1542	1398
SS	AUP	HSYMB		STORE	1696	10	1230	1735
	ALO	LSYMB		SYMBOL	1735	15	1575	1932
	LDD		SUBR7		1932	69	1785	1279
	RAL	EQ	L0004		1785	65	1793	0952
SD	LDD	P0007		SET D ADDR	1930	69	1983	1636
	SDA	P0007			1636	22	1983	1686
	SRT	0004	BT		1686	30	0004	1728

D	00	0088	D0001	CONSTANTS	1685	00	0088	0937
G	00	0000	G0001		1834	00	0000	0941
J	00	0088	J0001		1645	00	0088	0944
L	00	0000	L0001		1448	00	0000	0949
F	00	0088	F0001		1882	00	0088	0939
K	00	0000	K0001		1646	00	0000	0946
AXD	00	0088	D0001		1828	00	0088	0937
AXC	00	0098	SCD		1928	00	0098	1931
XEQ	STL	ORXEQ	BT		1881	20	1596	1728
CEQ	00	0008	SCEQ		1782	00	0008	1832
88TH	00	0000	0800		1542	00	0000	0800
EQ	01	0000	EQ	ERASEABLE	1793	01	0000	1793
		HED						

SUB 21 PROCESS INSTR ADDRESS

REG	D0953	0953	PROC I						
REG	F0955	0955							
REG	G0958	0958							
REG	J0960	0960							
REG	K0962	0962							
REG	L0965	0965							
	HED	C							
PROCI	STD	EXITX	STORE EXIT	1453	24	1196	1833		
	RAU	1953	IS I BLANK	1833	60	1953	1883		
	NZU	NB		1883	44	1788	1838		
	RAU	1952	IS D BLANK	1838	60	1952	1933		
	NZU	DNB		1933	44	1888	1938		
	LDD	DRUMT	D I BLANK	1938	69	1212	1884		
	BDO		TEST DRUM	1884	90	1988	0959		
	RAL	ORBAL	L0004	TAG	1988	65	1492	0968	
DNB	RAL	D	SUB11	FIND BEST	1888	65	1692	1801	
D0001	STL	ORBAL	L0004	FOUND	0953	20	1492	0968	
D0002	RAL	P0010	X	DRUM PAKED	0954	65	1986	1742	
X	ALO	89TH			1742	15	1695	1934	
	STL	P0010	EXITX		1934	20	1986	1196	
NB	LDD	F	SUBR4	WHAT IS I	1788	69	1792	1213	
F0001	RAL	1959	J0001	ABSOLUTE	0955	65	1959	0960	
F0002	LDD	J	SUBR9	REGIONAL	0956	69	1835	1174	
F0003	LDD	K	SUBR6	SYMBOLIC	0957	69	1885	1275	
J0001	LDD	L	SUBR5	TEST RANGE	0960	69	1935	1750	
J0002	RAL	P0010	X	REG ERROR	0961	65	1986	1742	
L0001	LDD	AXD	INDEX	DRUM ADDR	0965	69	1736	1826	
L0002	LDD	P0007	800X	800X	0966	69	1983	1786	
L0003	LDD	AXC	INDEX	CORE ADDR	0967	69	1836	1826	
L0004	LDD	P0007		OTHER	0968	69	1983	1886	
	SIA	P0007	MOR		1886	23	1983	1936	
800X	SIA	P0007			1786	23	1983	1889	
	AUP	XEQ	SB17I	GET 800X E	1889	10	1842	1854	
90XX	LDD	P0007		SET CORE	1939	69	1983	1989	
	SIA	P0007		ADDRESS	1989	23	1983	1440	
	LDD	CEQ	SUB13	GET CORE E	1440	69	1843	1964	
K0001	RAL	EQUIV	J0001	SYM DEFIND	0962	65	1223	0960	

K0002	RAL	G	SUB11	SYM UNDEF	0963	65	1490	1801
K0003	RAL	P0010	X	S TAB FULL	0964	65	1986	1742
G0001	LDD	P0007	SI	FOUND BEST	0958	69	1983	1540
G0002	RAL	P0010	X	DRUM PAKED	0959	65	1986	1742
SI	SIA	P0007			1540	23	1983	1590
	AUP	HSYMB		STORE	1590	10	1230	1640
	ALO	LSYMB		SYMBOL	1640	15	1575	1690
MOR	LDD	MOR	SUBR7		1690	69	1936	1279
	LDD	ORBAL		SET OR	1936	69	1492	1745
	STD	OPREG	EXITX	TO ORB	1745	24	1872	1196
SCEQ	STL	ORCEQ	MOR	SAVE CORE	1740	20	1738	1936
D	00	0089	D0001	CONSTANTS	1692	00	0089	0953
F	00	0000	F0001		1792	00	0000	0955
J	00	0089	J0001		1835	00	0089	0960
K	00	0000	K0001		1885	00	0000	0962
L	00	0000	L0001		1935	00	0000	0965
G	00	0089	G0001		1490	00	0089	0958
89TH	00	0000	0080		1695	00	0000	0080
AXD	00	0089	L0004		1736	00	0089	0968
AXC	00	0099	90XX		1836	00	0099	1939
XEQ	STL	ORXEQ	MOR		1842	20	1596	1936
CEQ	00	0009	SCEQ		1843	00	0009	1740
	HED							
1IXXX	00	0000	0001	CONSTANTS	0992	00	0000	0001
1DXXX	00	0001	0000		1020	00	0001	0000
2IXXX	00	0000	0002		1771	00	0000	0002
2DXXX	00	0002	0000		1288	00	0002	0000
H8XXX	80	0000	0000		1406	80	0000	0000
H9XXX	90	0000	0000		1216	90	0000	0000
90XXX	00	0000	0090		1178	00	0000	0090
29XXX	00	0000	0029		1189	00	0000	0029
2000I	00	0000	2000	CONSTANT	1014	00	0000	2000
TORG	00	0000	T0001		0990	00	0000	1990

PAT

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