AN/UYK-1 (TRW-130) PROGRAM ASSEMBLER

FLOW CHARTS AND SPECIAL INSTRUCTIONS

M250-2U38

Thompson Ramo Wooldridge Inc.
R W DIVISION
8433 Fallbrook Ave
Canoga Park, Calif

Oct 1962

AN/UYK-1 (TRW-130) PROGRAM ASSEMBLER

FLOW CHARTS AND SPECIAL INSTRUCTIONS

T A B L E OF C O N T E N T S

T A B L E O F C O N T E N T S, contd.

AN/UYK-1 (TRW-130) PROGRAM ASSEMBLER

FLOW CHARTS AND SPECIAL INSTRUCTIONS

INTRODUCTION

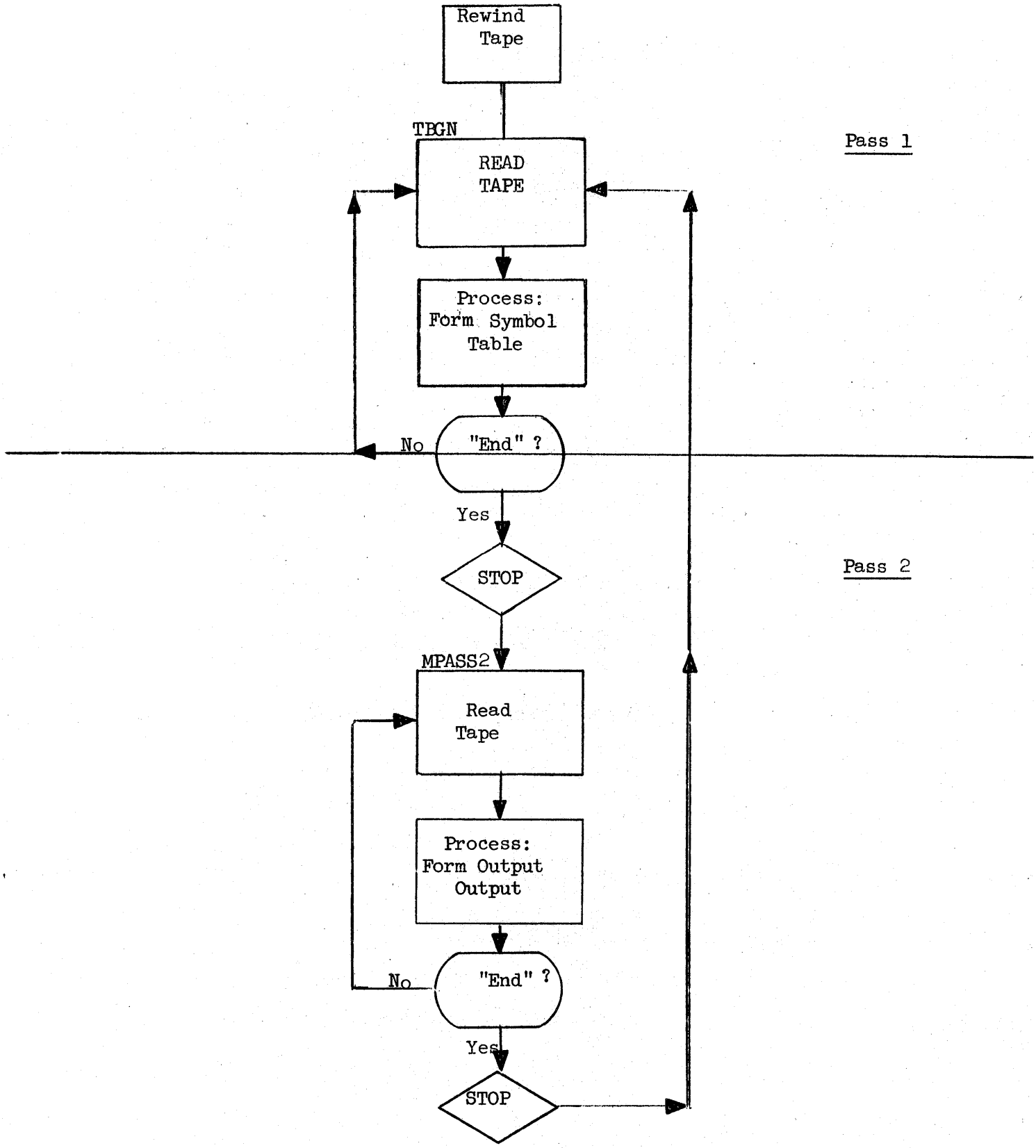
The flow charts are graphic representations of the processes involved in Logand Assembler program operations. The details for preparing and executing the program are given in the document AN/UYK-1 (TRW-130) Program Assembler for Paper Tape System (M250-2U20).

Included with this collection of flow charts are descriptions of the special lograms used to correct assembly errors, pack words, convert from one code to another, etc. (See pages 30 through 43, and 50).

In assembling the set of logands and parameters which comprise a program, the assembly first "sorts" into meaningful categories the information transmitted to memory. The tables necessary for this "filing and indexing" procedure are shown in pages 44 through 50.

As insurance against loss of information through failure (i.e. power failure, malfunction, etc.) certain memory locations are dedicated to counting the assembly order and preserving the current address and contents of command words. Programming rules for these and other conventions for the Assembly Program are shown at page 44.

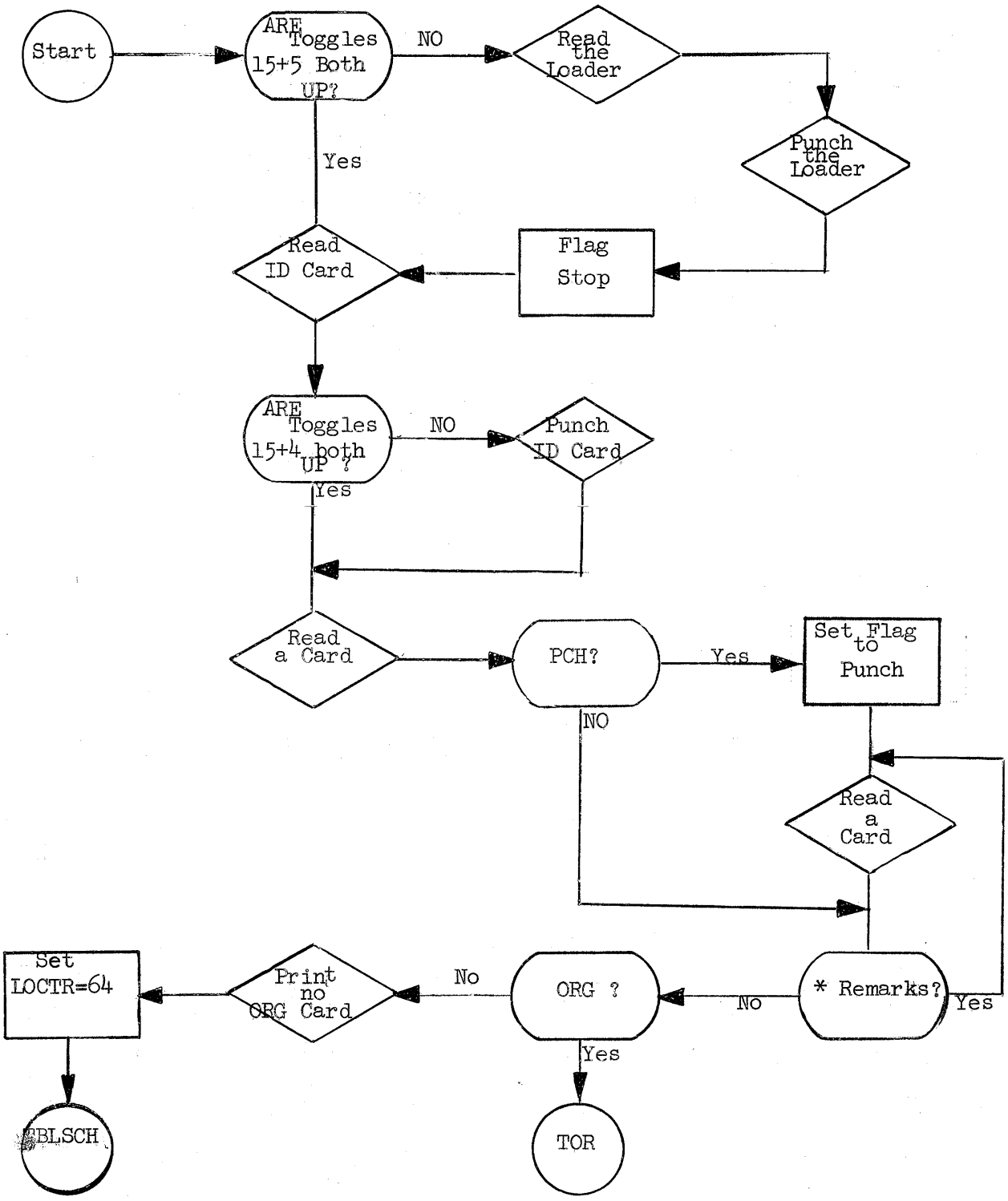
-o-

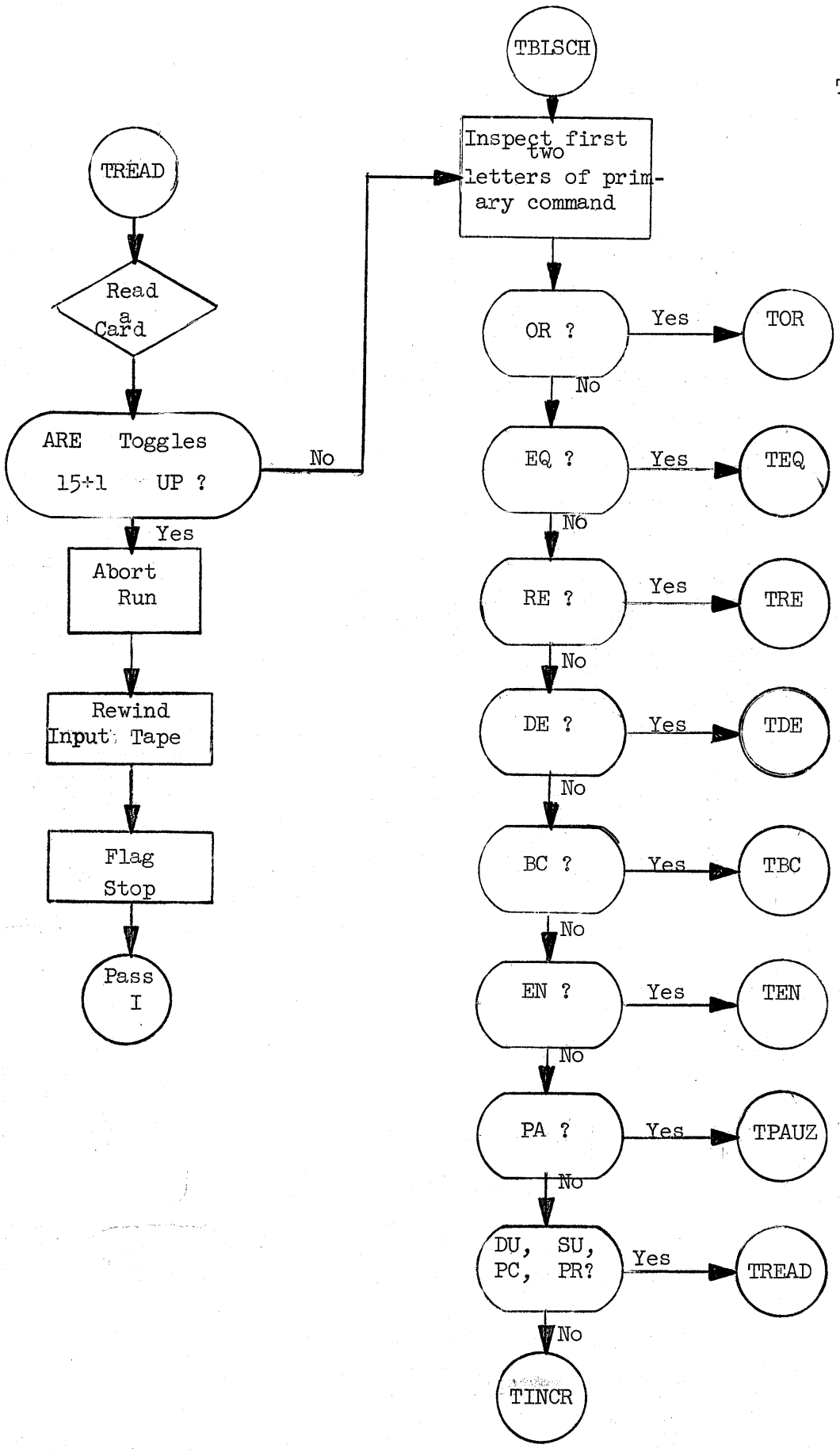


Block Diagram of Assembly Program

Pass 1

Initiation of Program Assembly





p 4

p 4

p 5

p 5

p 5

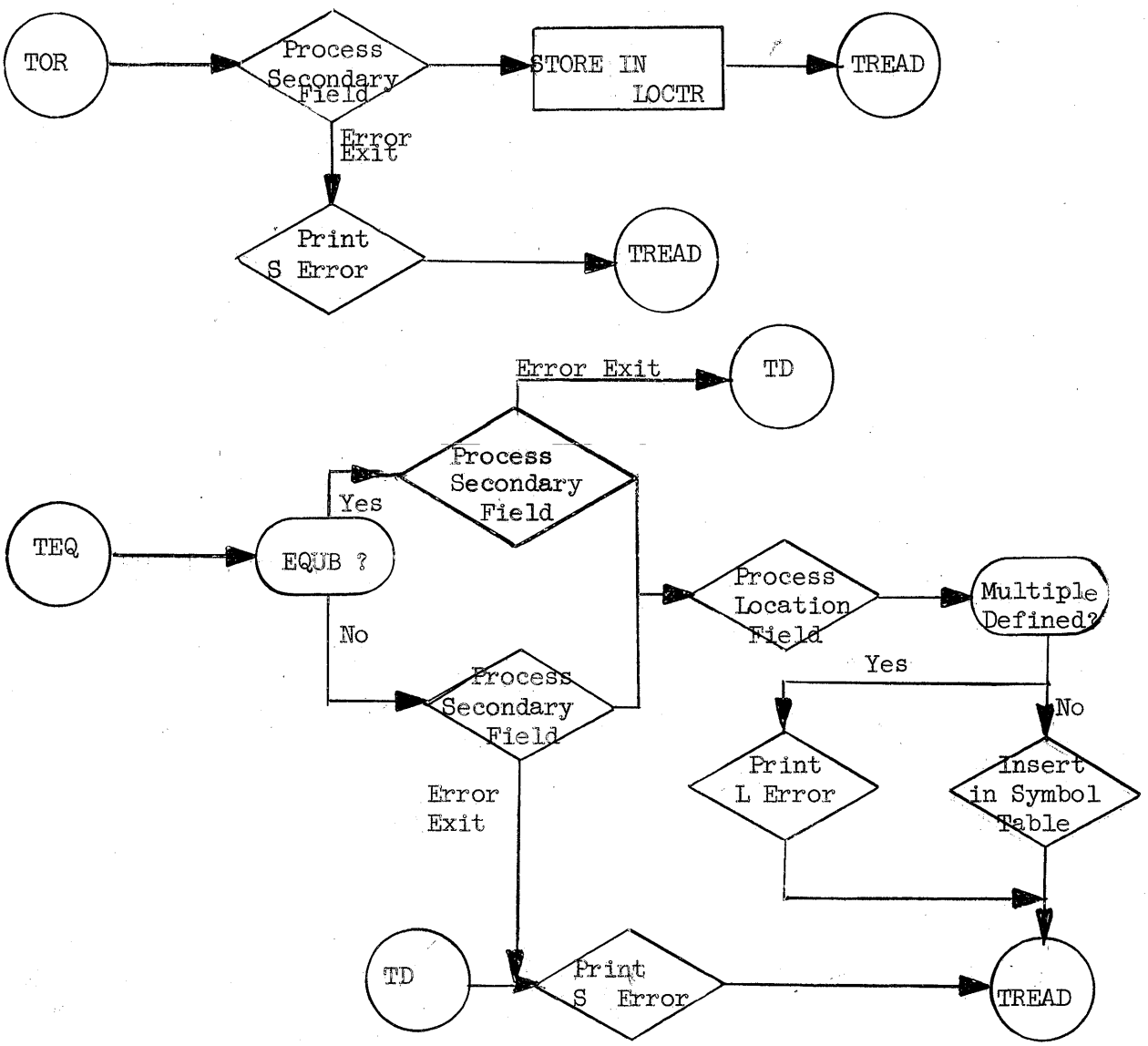
p 6

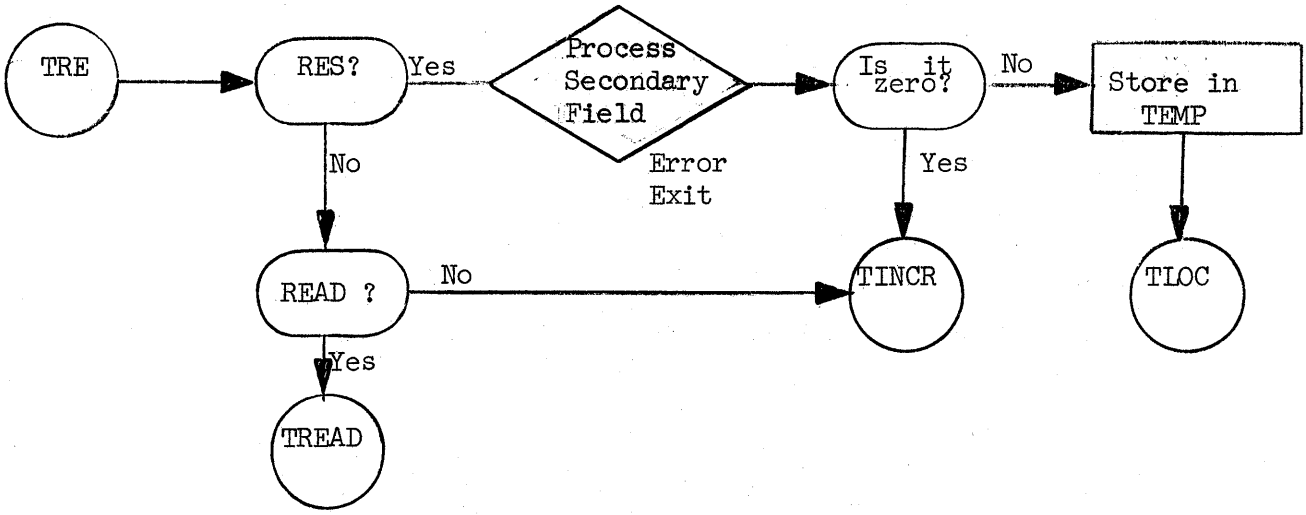
p 6

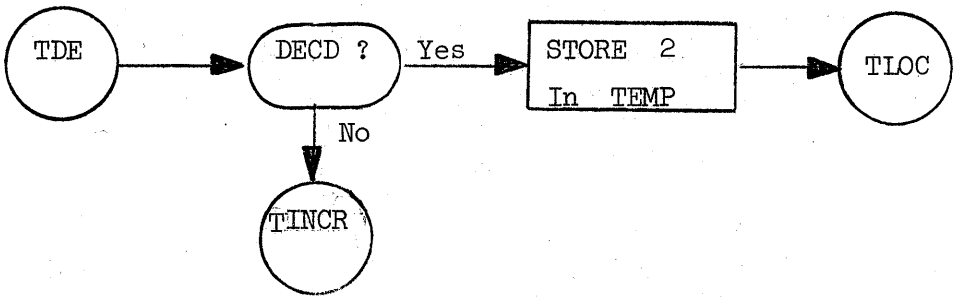
p 3

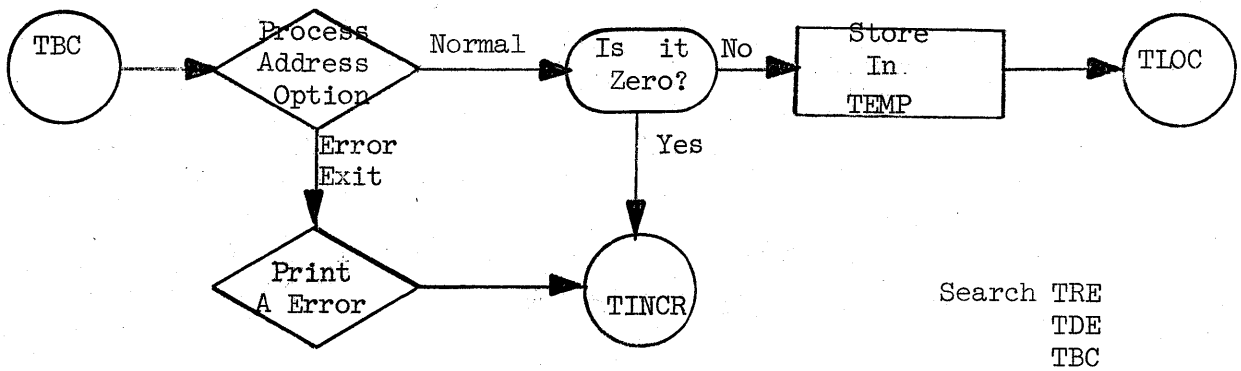
p 6

Search TOR
TBQ

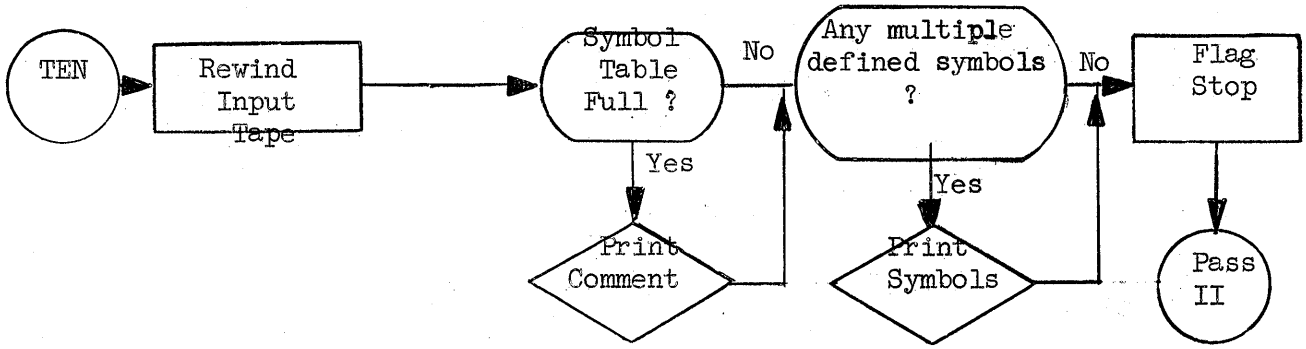


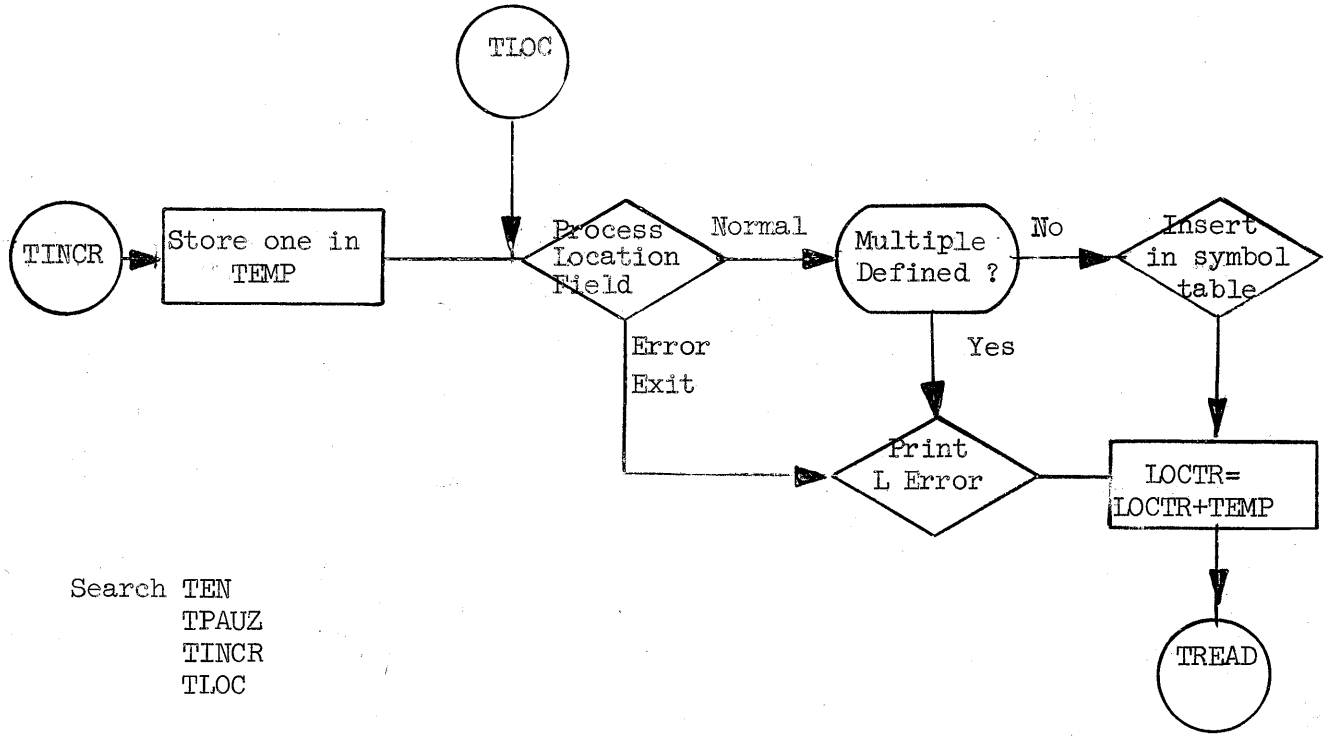






Search TRE
TDE
TBC

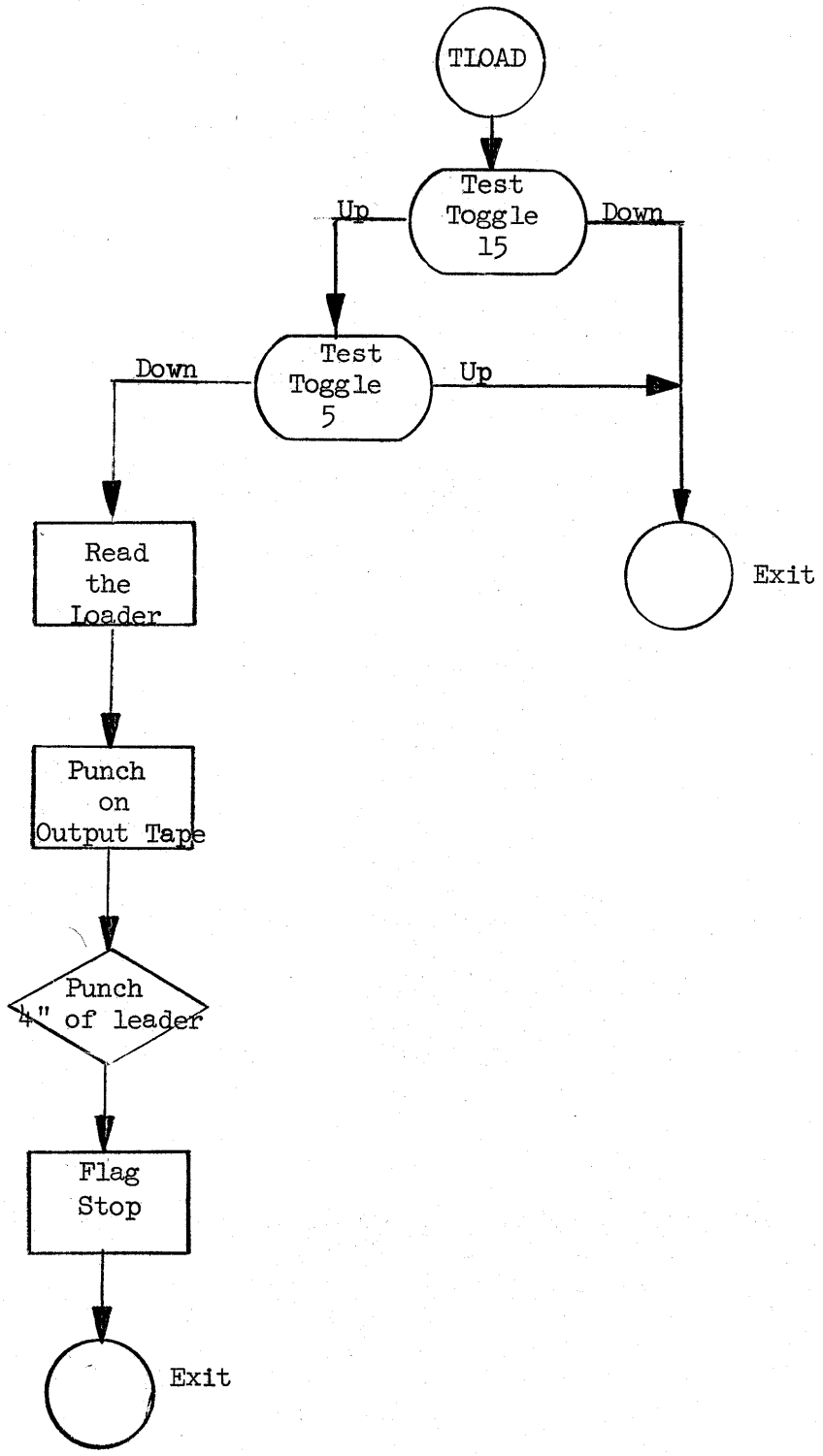




Search TEN
TPAUZ
TINCR
TLOC

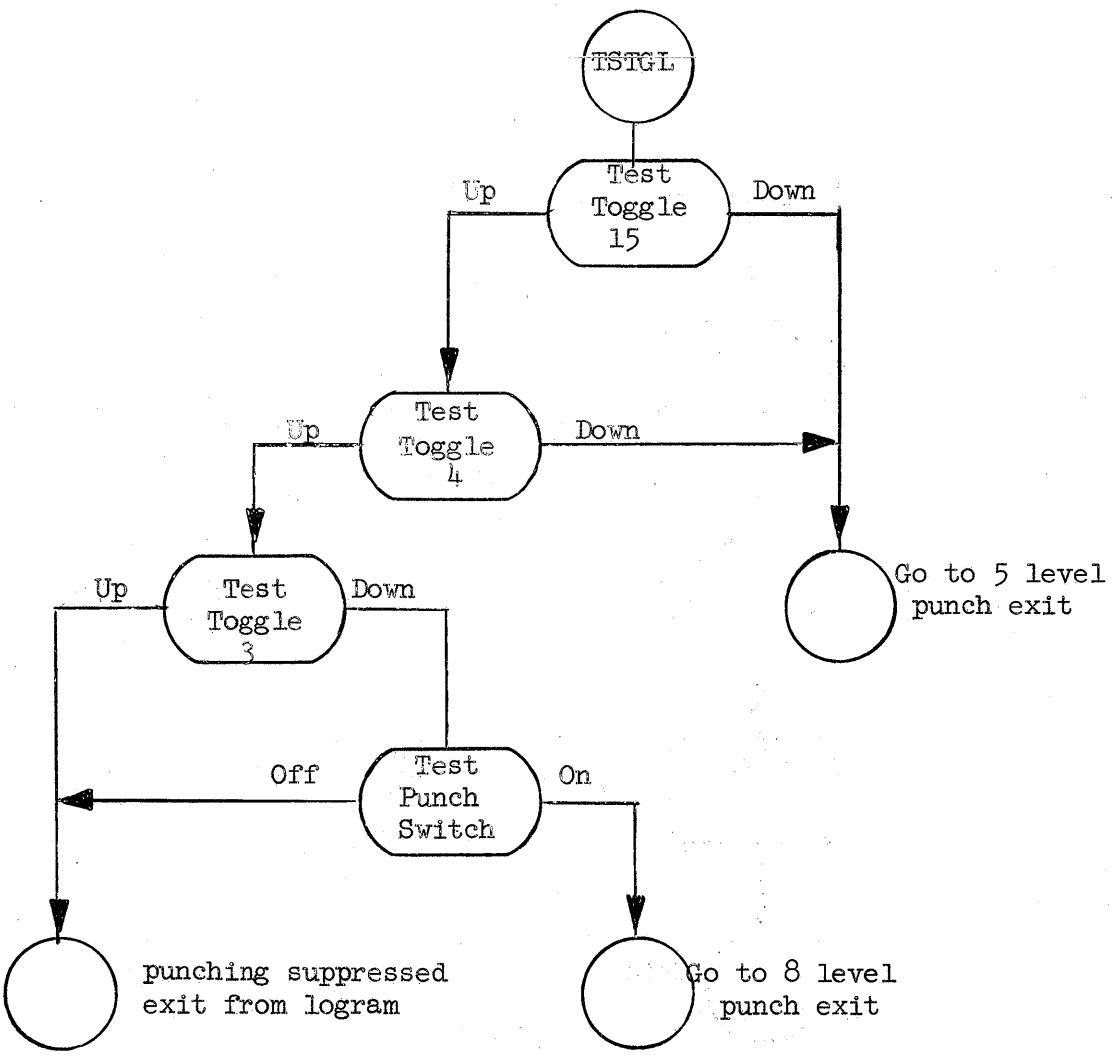
Logram: TLØAD (Read, Punch Loader)

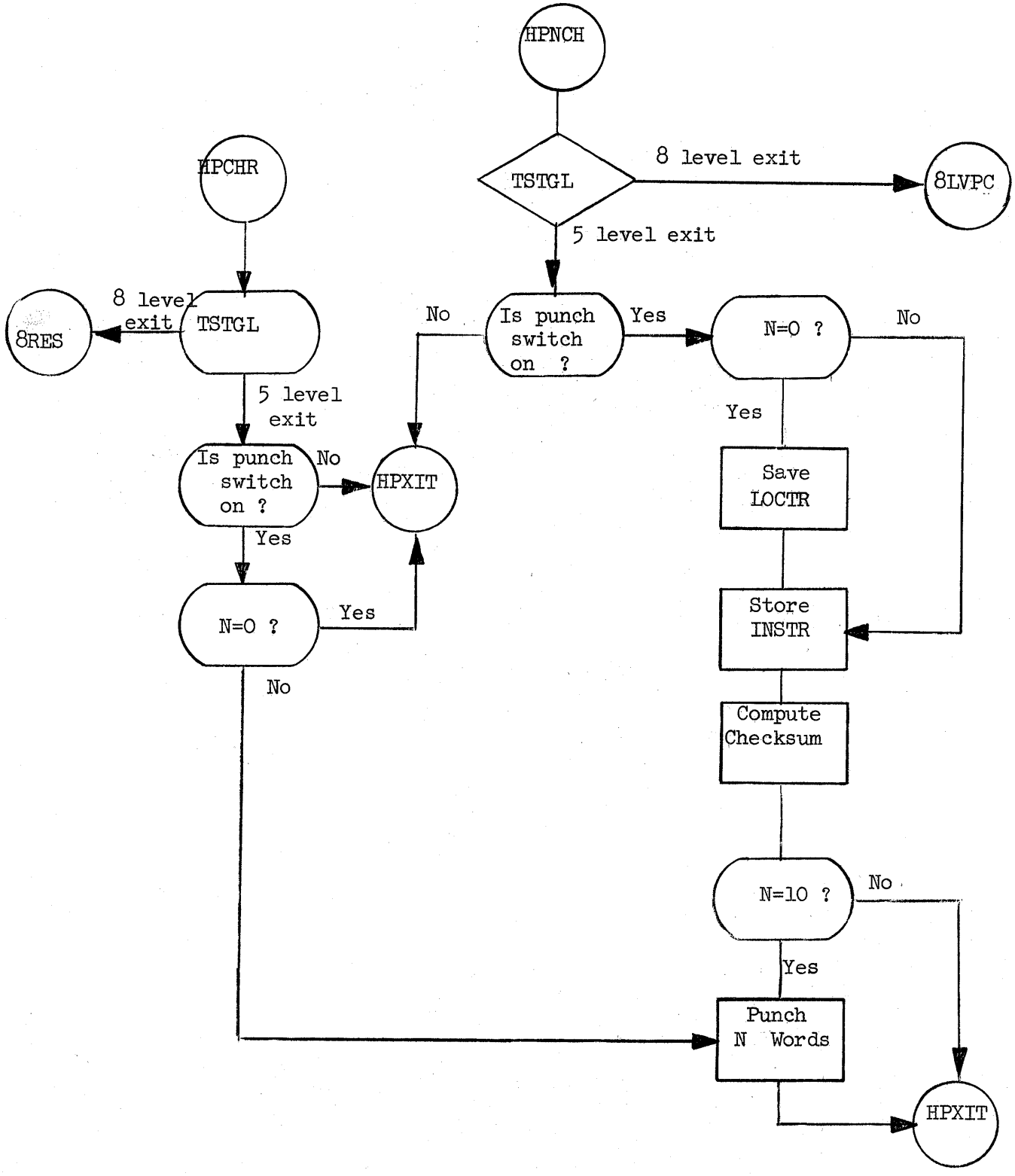
Function: To read, then punch the loader on the output tape.



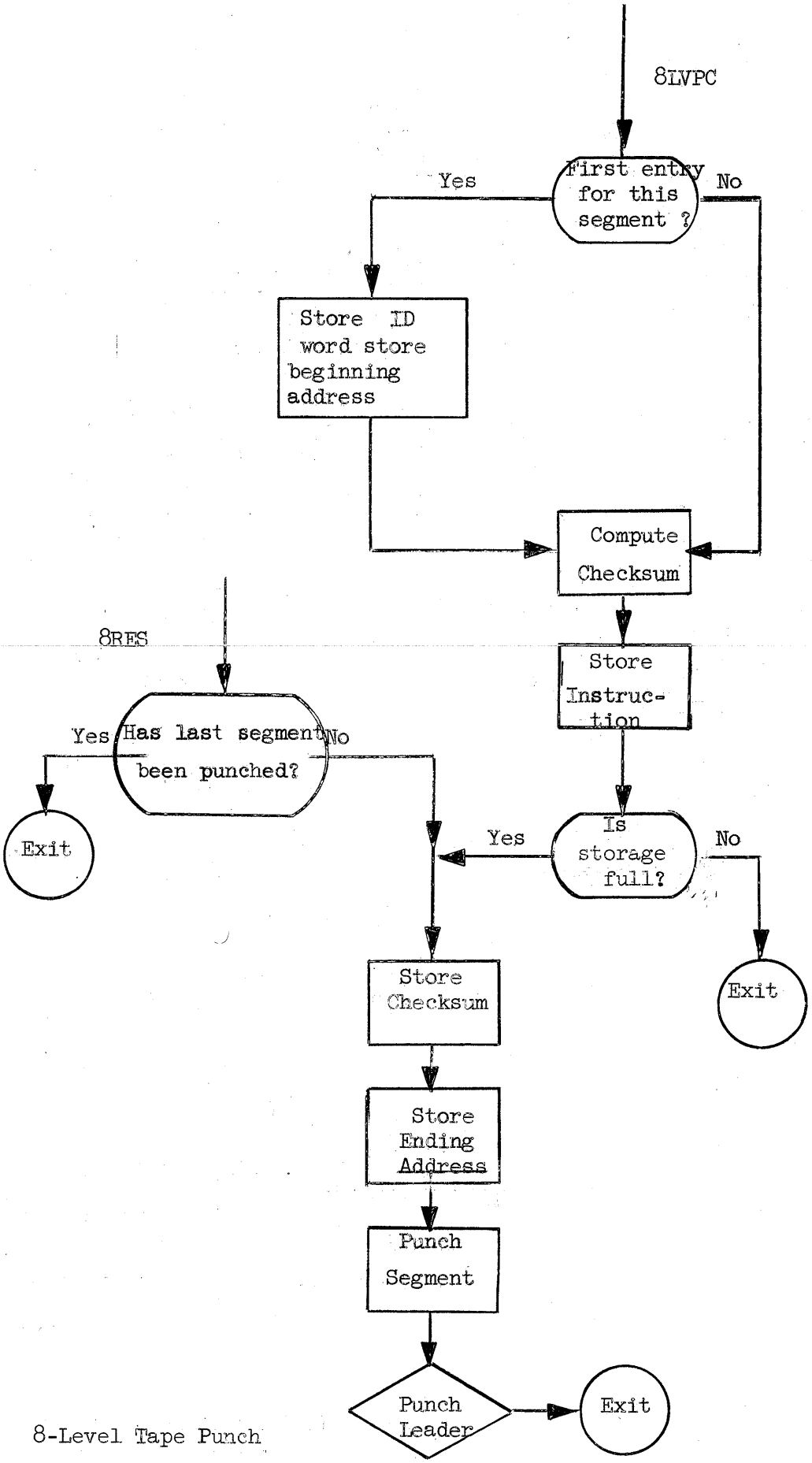
Subroutine TSTGL (Test for Tape Level Type)

Used by HPNCH, HPCHR and MPCHX to test toggle switches 15 and 4 to determine whether 5 or 8 level tape is expected. If 8 level tape is called for, it further tests to see if punching has been suppressed either by toggle switch settings or by the absence of a PCH card.

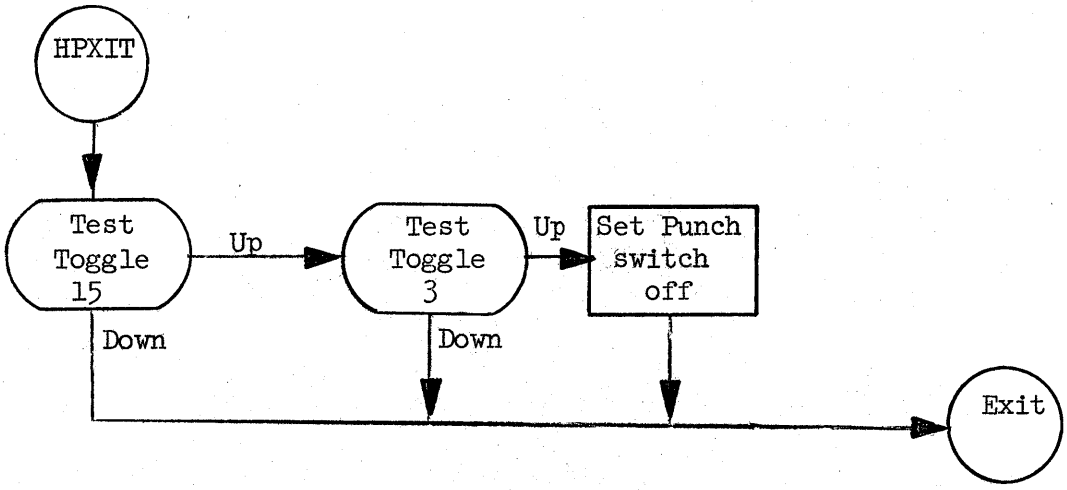




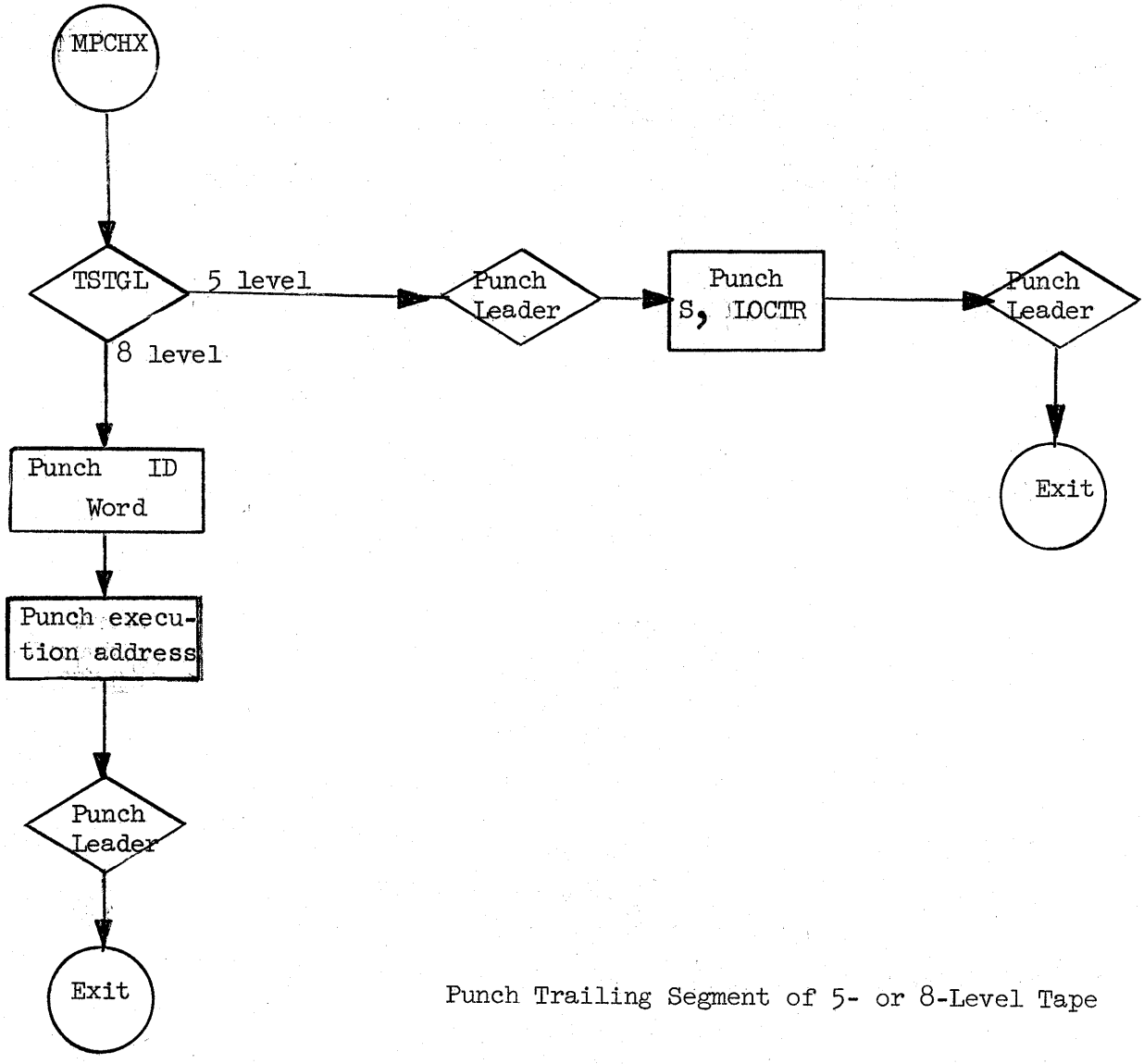
Test for Punch (5 level)



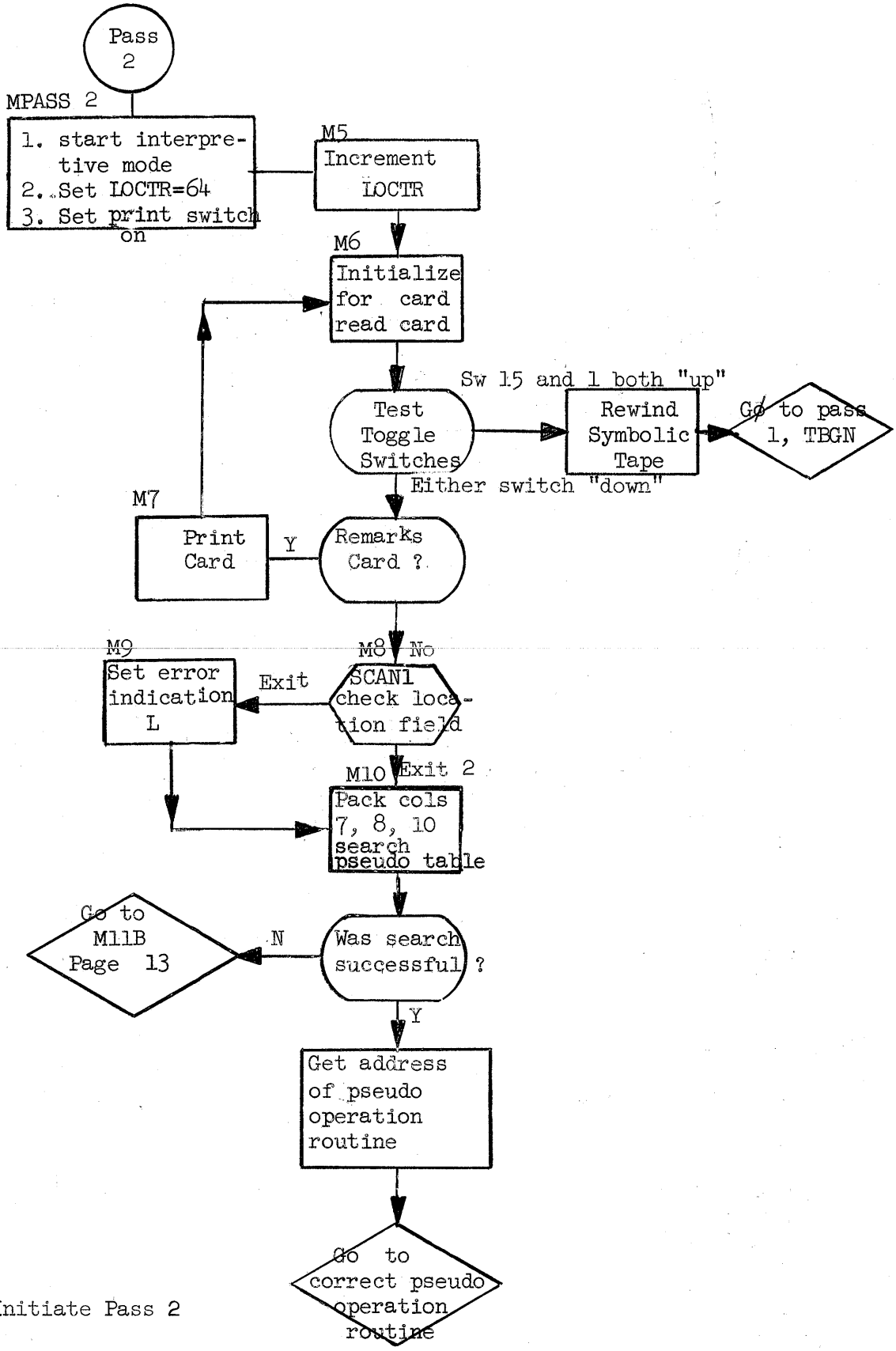
8-Level Tape Punch



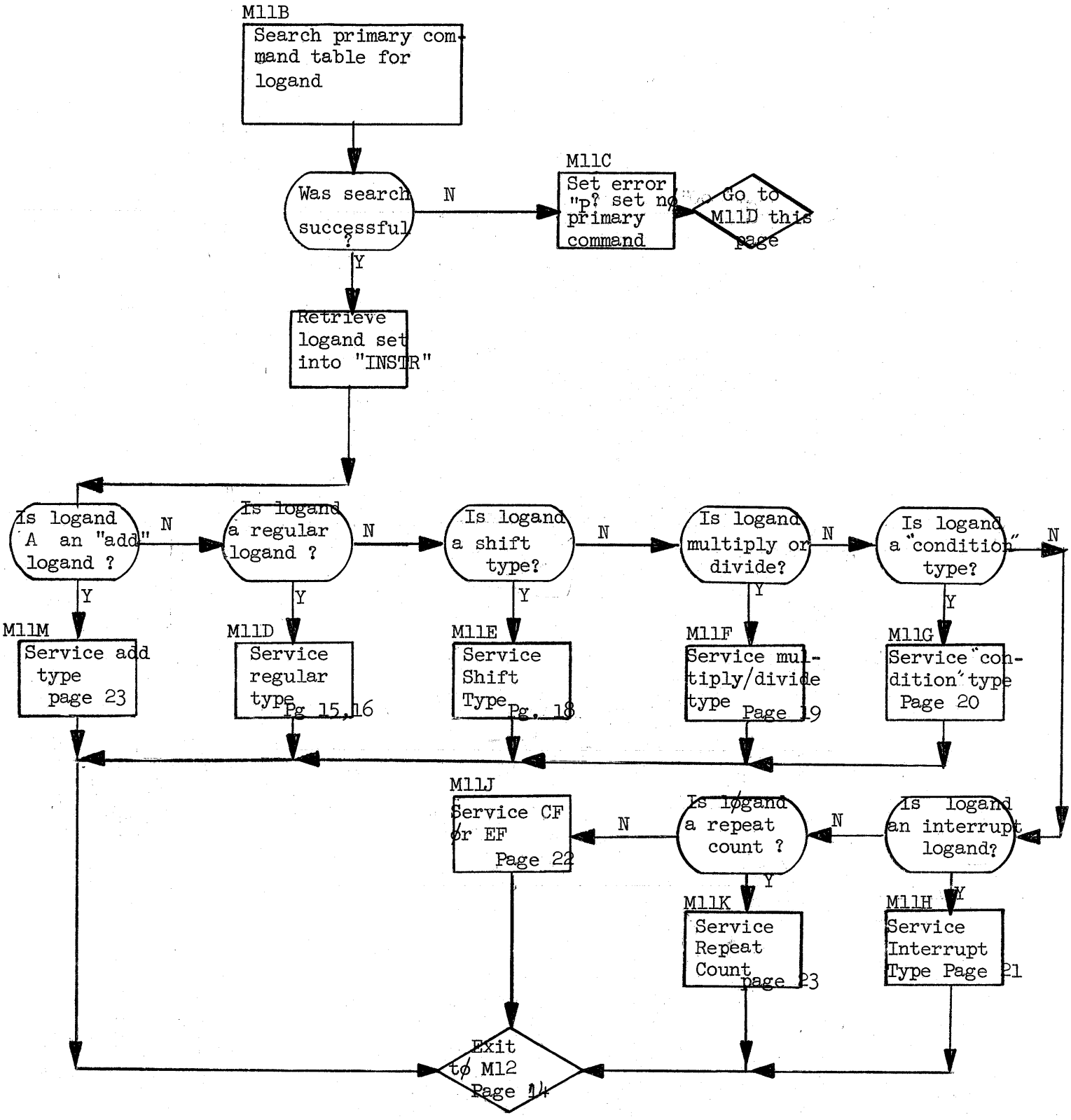
Punch "OFF"



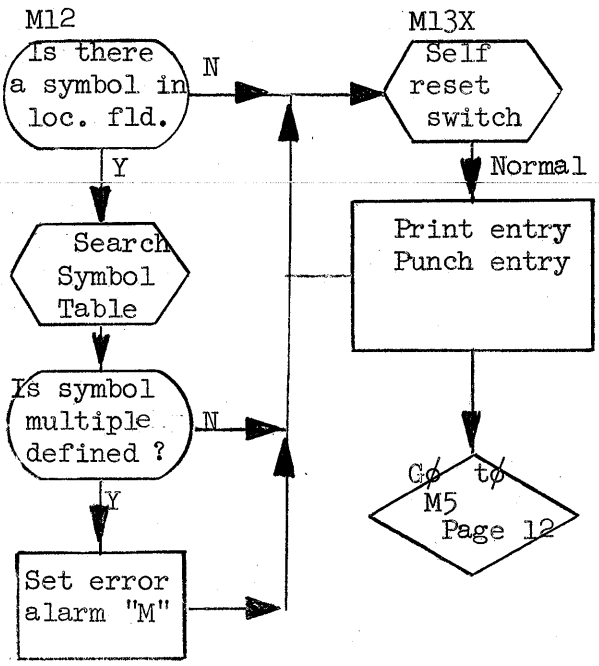
Punch Trailing Segment of 5- or 8-Level Tape



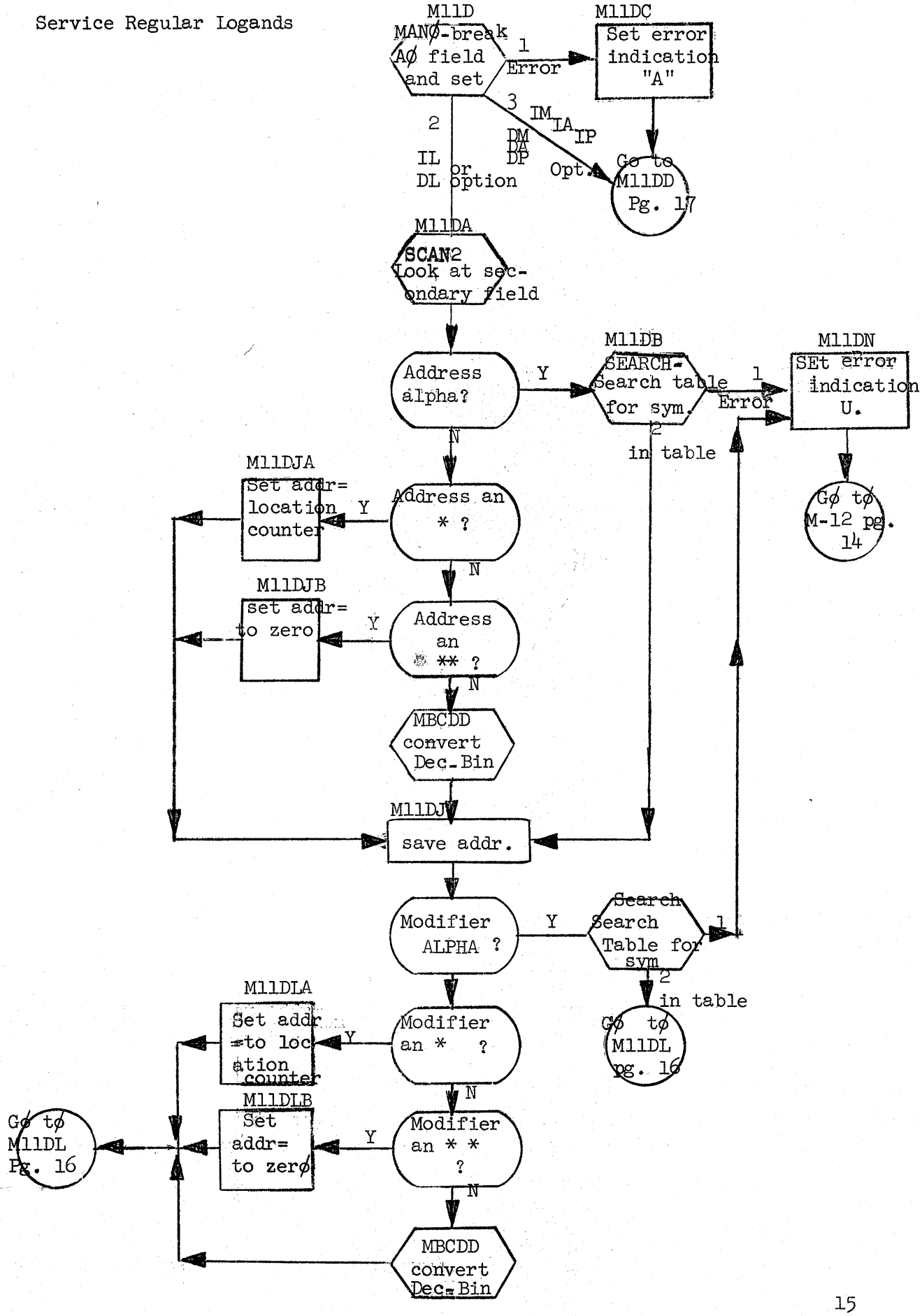
Initiate Pass 2

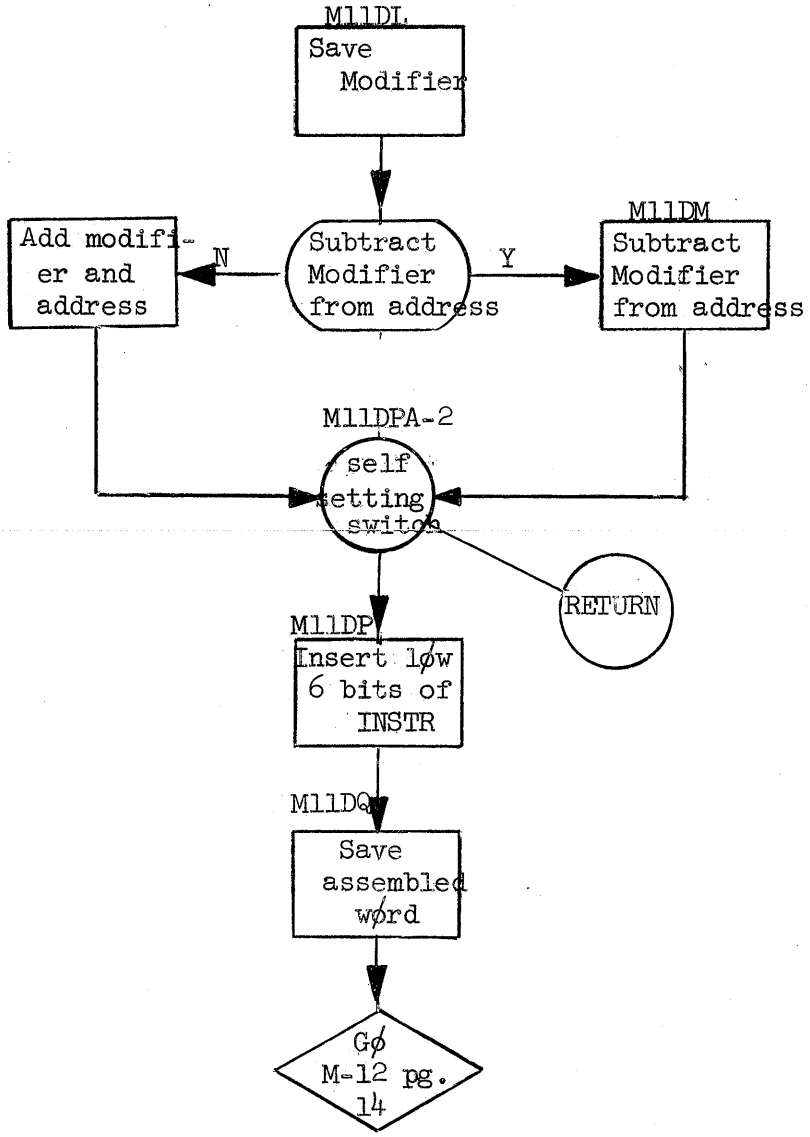


Logand Search



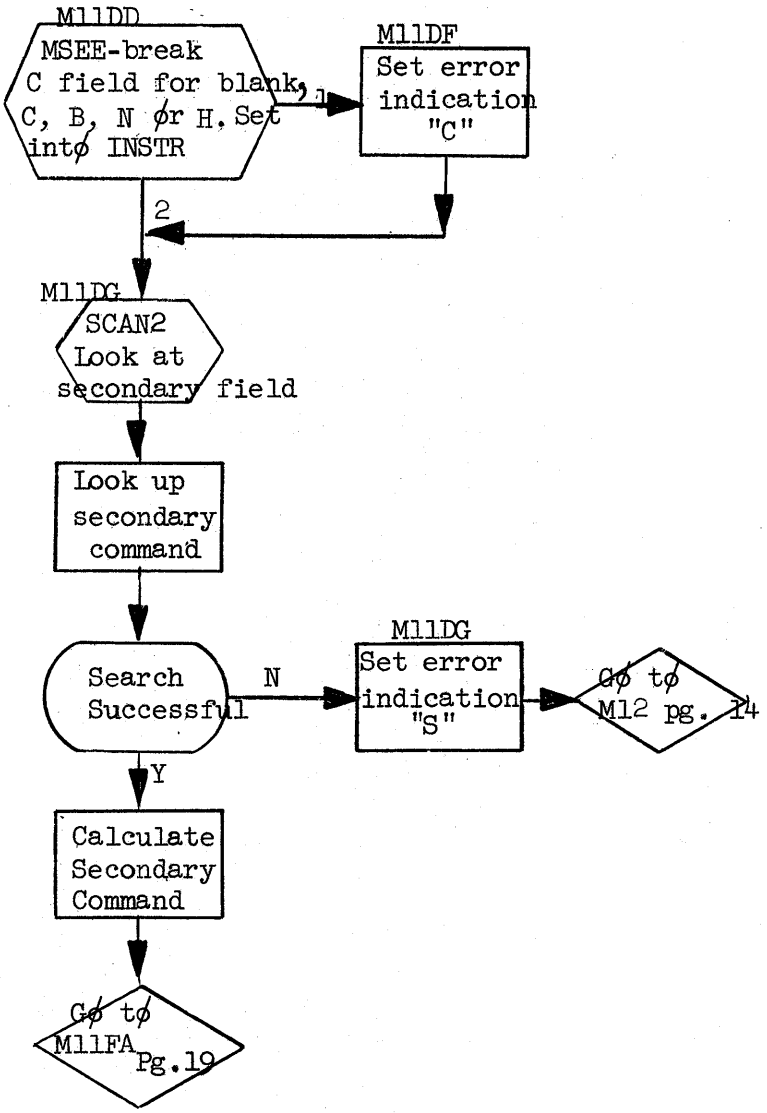
Examine Location Field


M11D
MANØ-break
AØ field
and set
1
Error
M11DC
Set error
indication
"A"
3
IM IA IP
DM DA DP
Opt.
Go to
M11DD
Pg. 17
2
IL or
DL option
M11DA
SCAN2
Look at sec-
ondary field
Address
alpha?
Y
M11DB
SEARCH-
Search table
for sym.
1
Error
M11DN
SEt error
indication
U.
2
in table
Gø tø
M-12 pg.
14
N
M11DJA
Set addr=
location
counter
Y
Address an
* ?
N
M11DJB
set addr=
to zero
Y
Address
an
** ?
N
MBCDD
convert
Dec-Bin
M11DJ
save addr.
Modifier
ALPHA ?
Y
Search
Search
Table for
sym
1
2
in table
Gø tø
M11DL
pg. 16
M11DLA
Set addr
=to loc
ation
counter
Y
Modifier
an * ?
N
M11DLB
Set
addr=
to zerø
Y
Modifier
an * *
?
N
MBCDD
convert
Dec-Bin
Gø tø
M11DL
Pg. 16

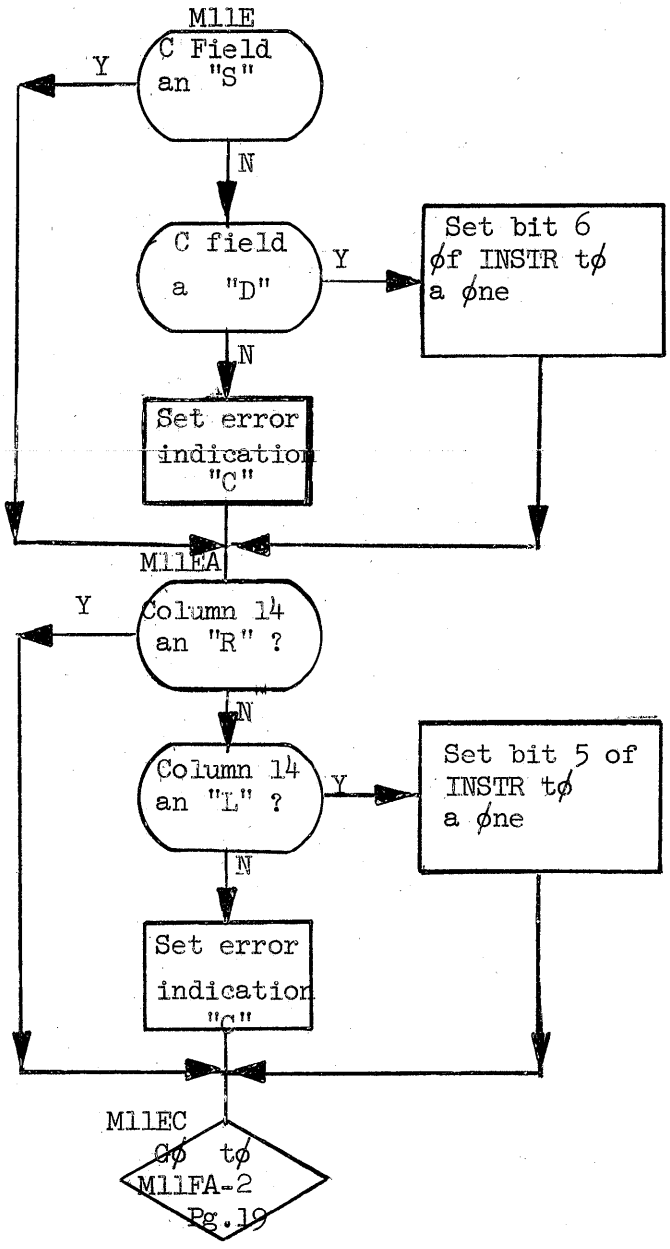


Service Modifier

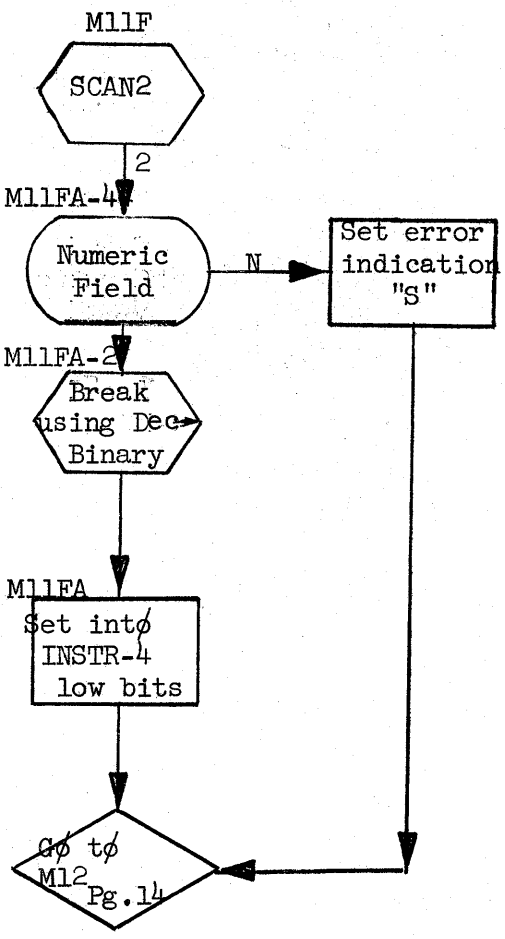
Service regular logands except IL or DL address options.



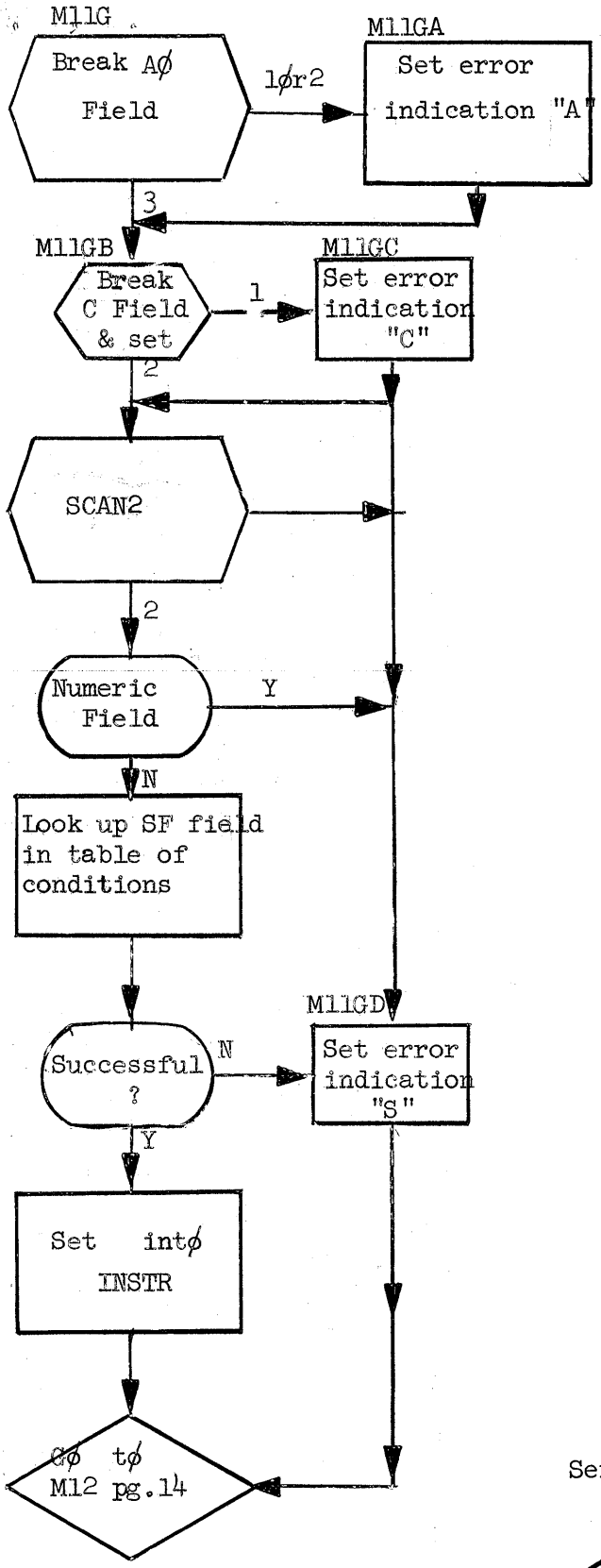
Service shift logands.



Service MP, MS, DV.



Service MV, BØ, BI, BR, SK, SR, TB.

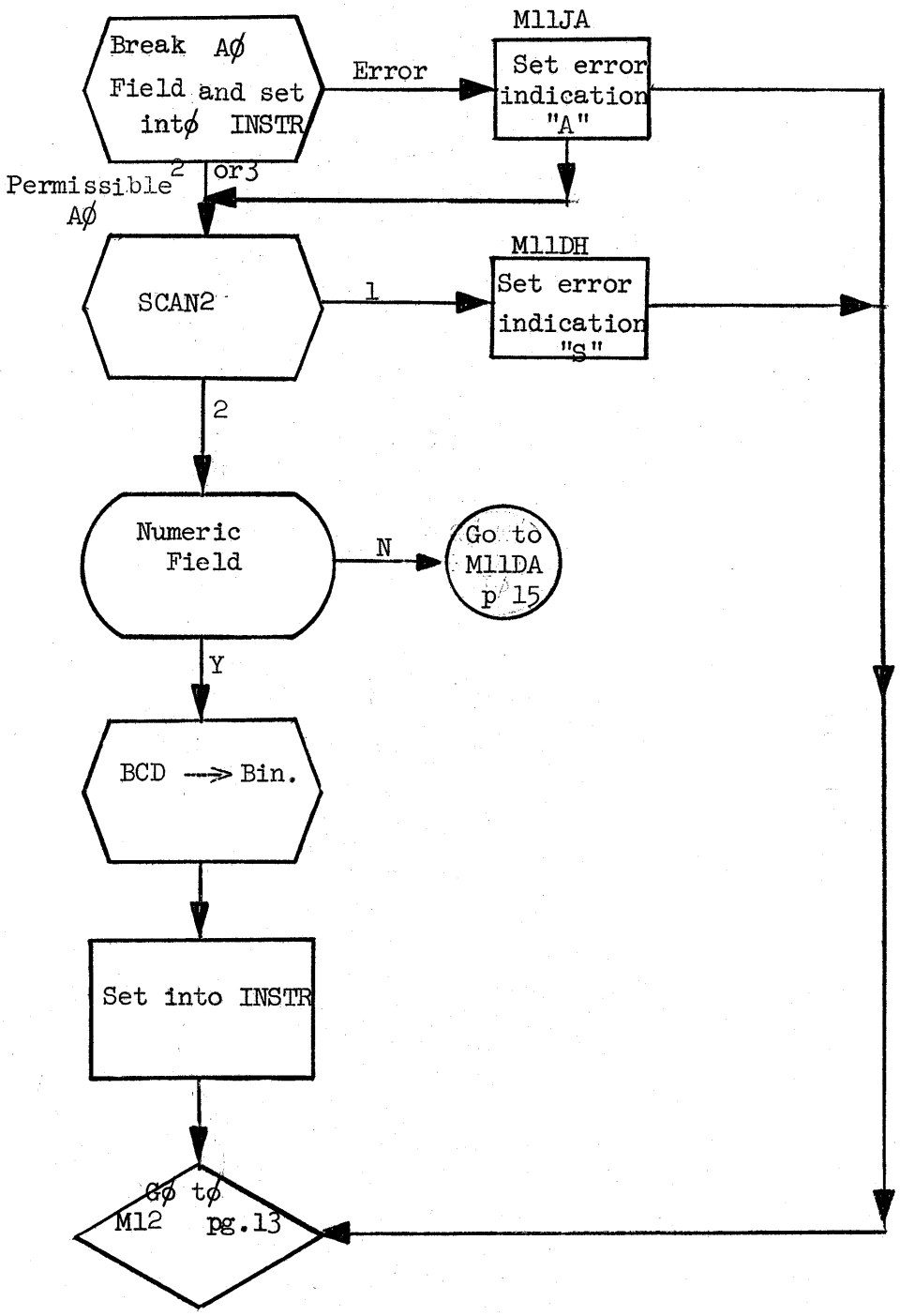


Service IT or TM

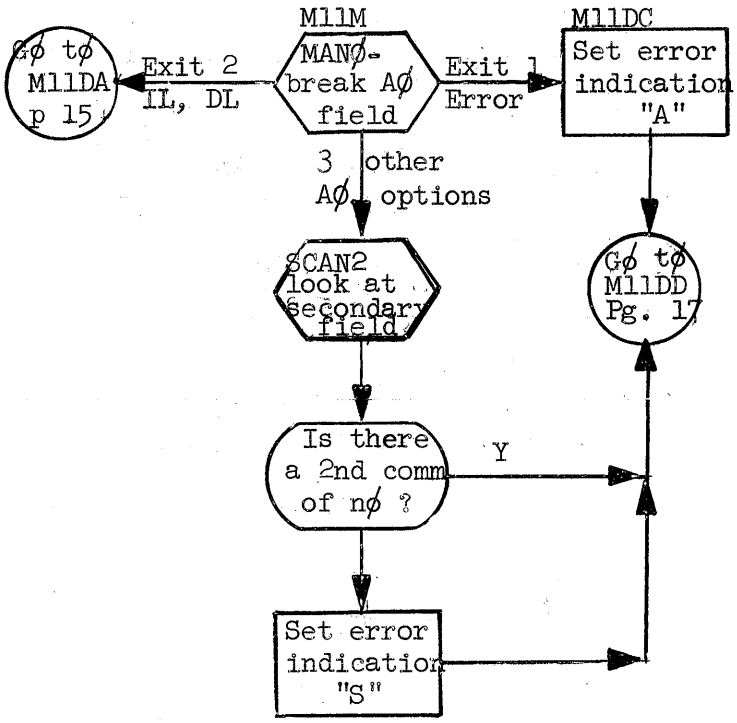
Go to M11DA p 15

Service EF or CF.

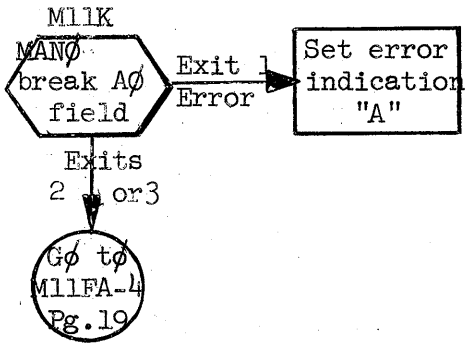
(see p 33)



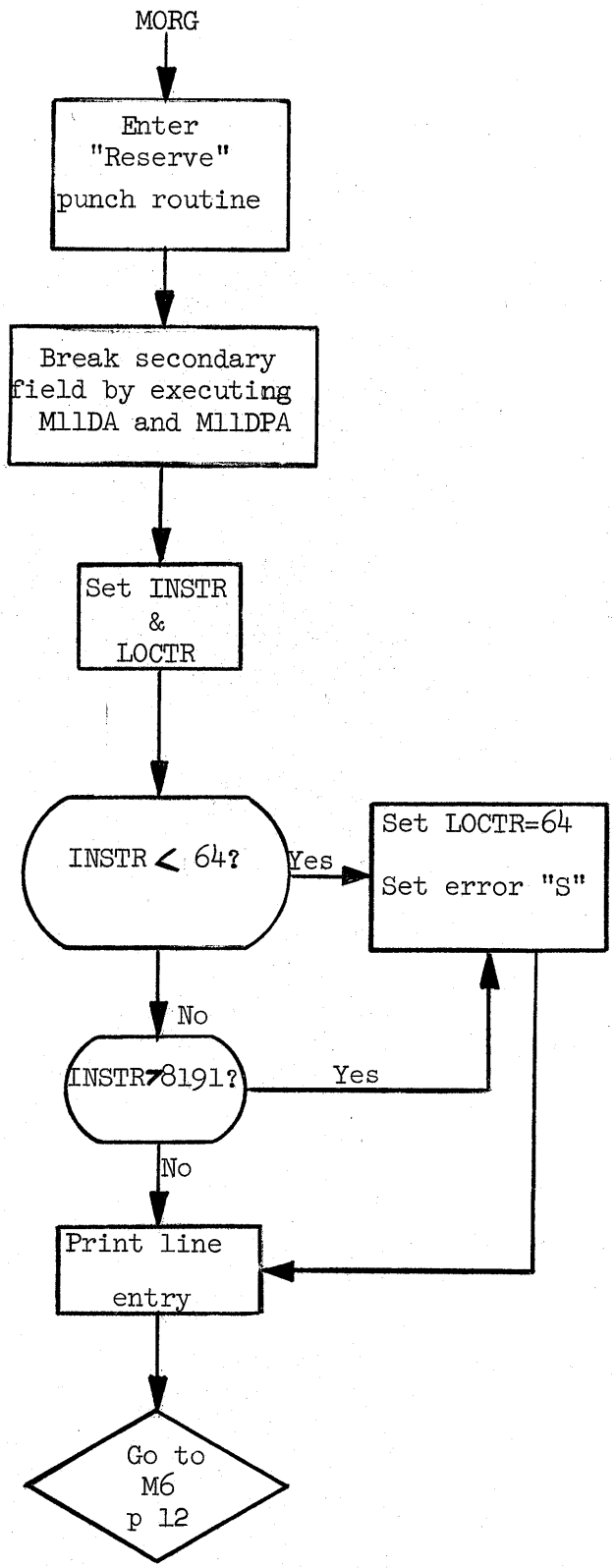
Service ADD type logands.

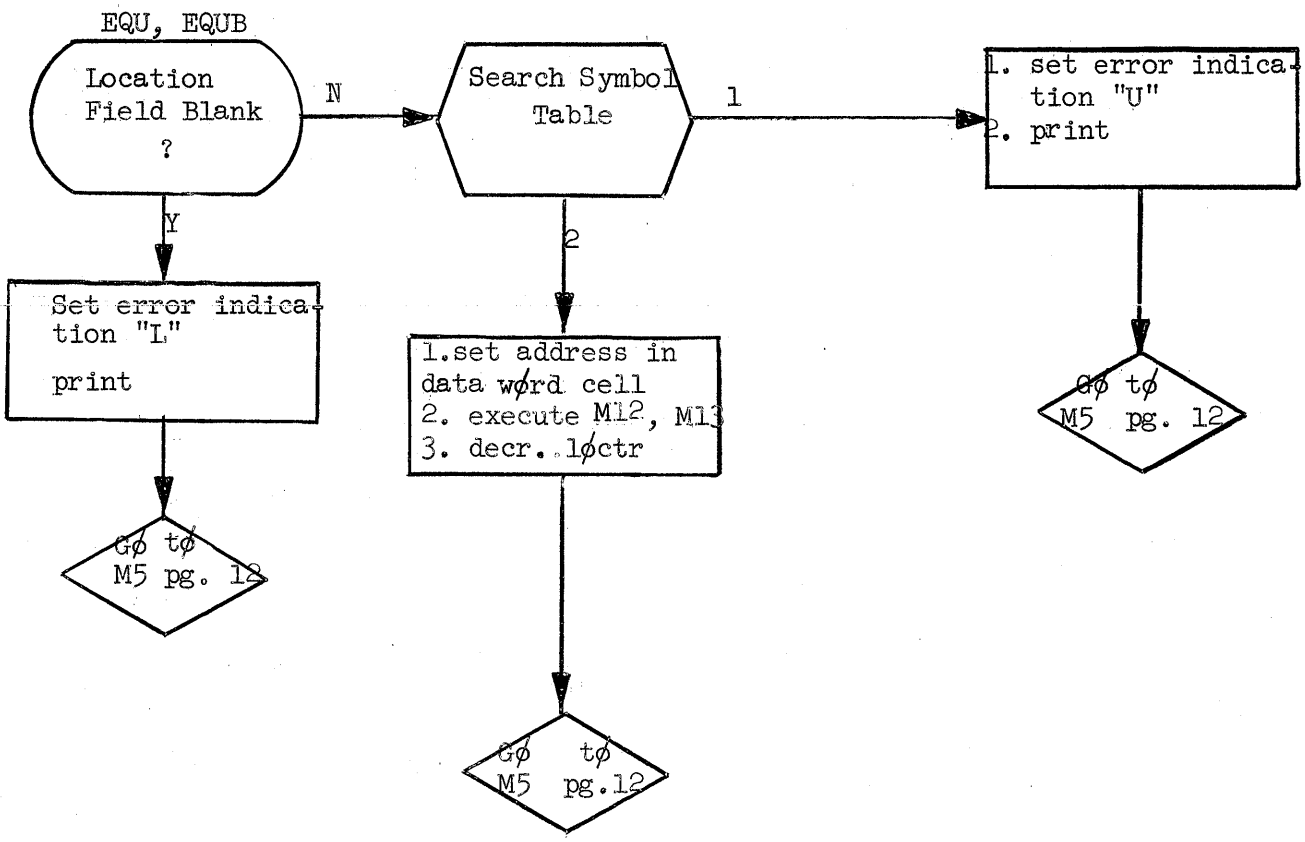


Service RC Logand.



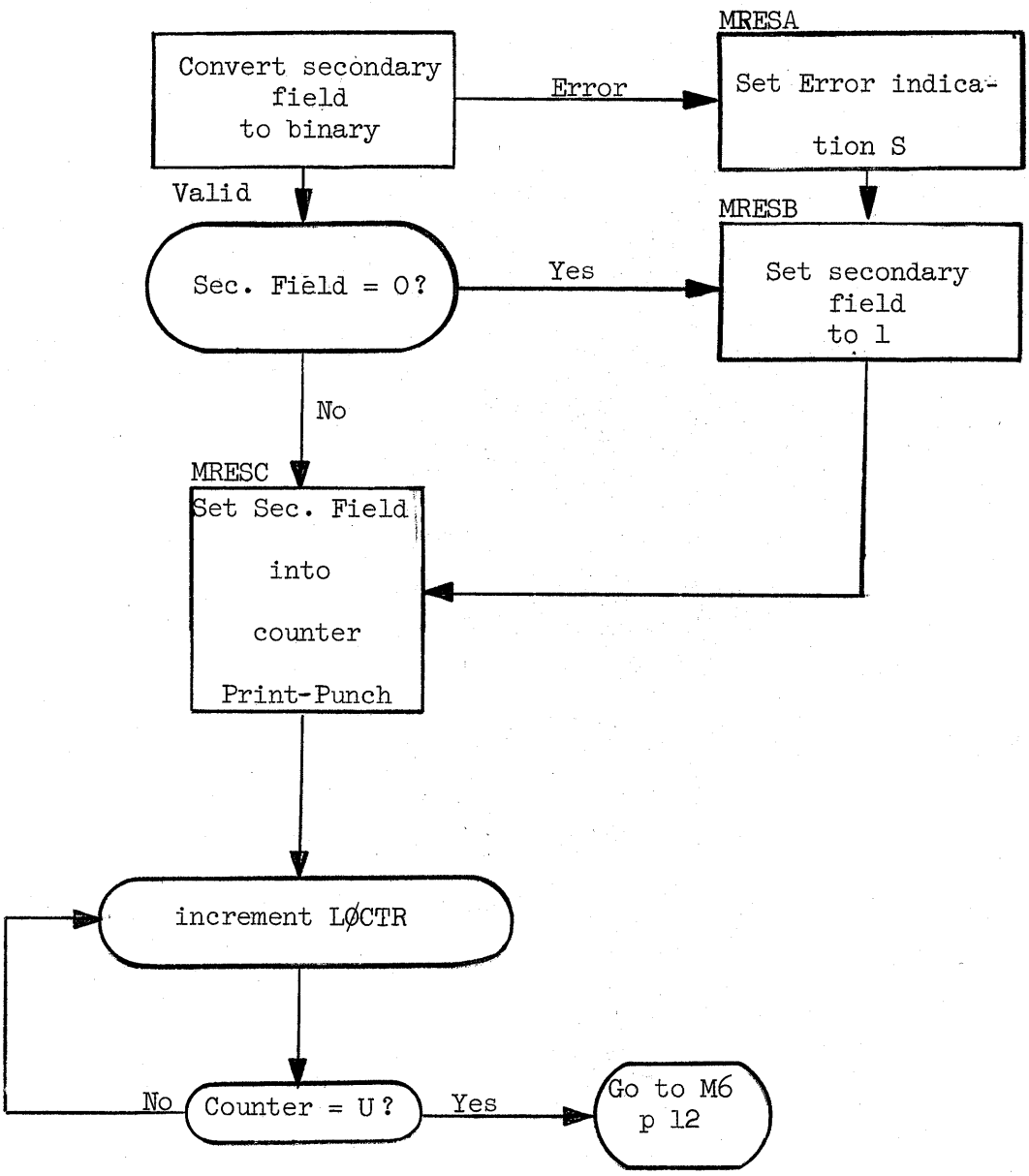
MORG = Prepare to Enter Pass 2



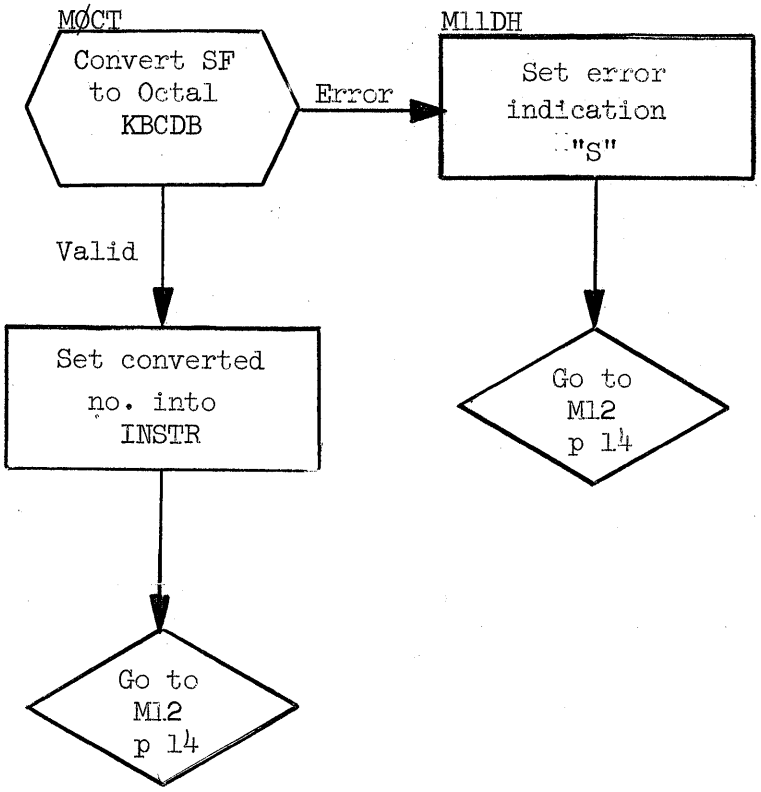


Place Symbol (EQU) or Octal Integer (EQUB) in Symbol Table

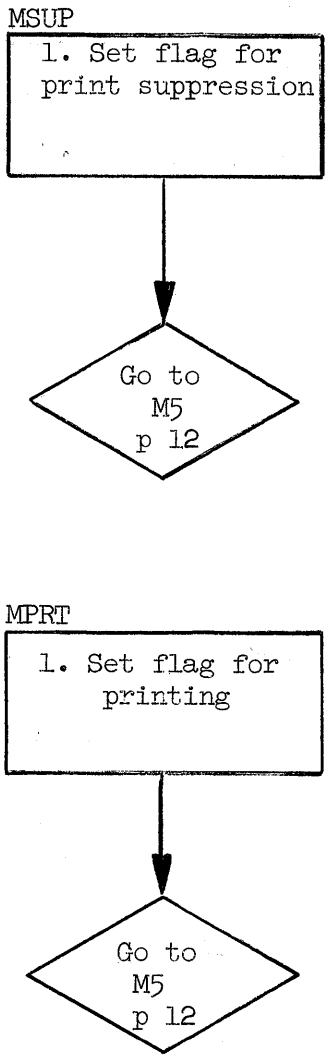
MRES



Reserve a Block of Cells
Pseudo operation RES

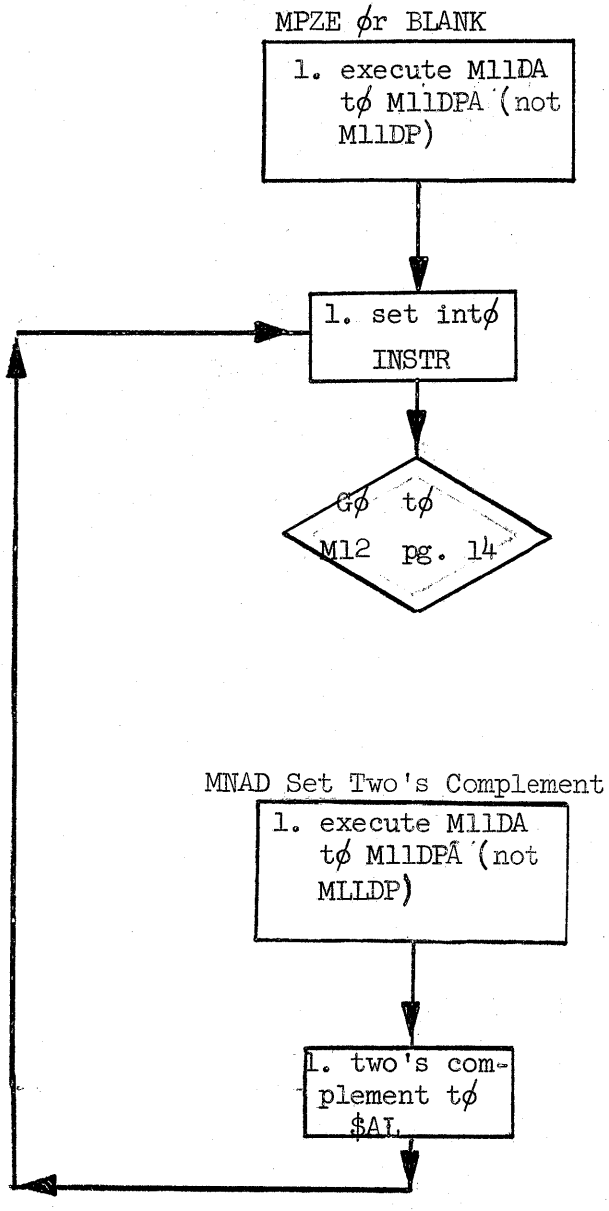


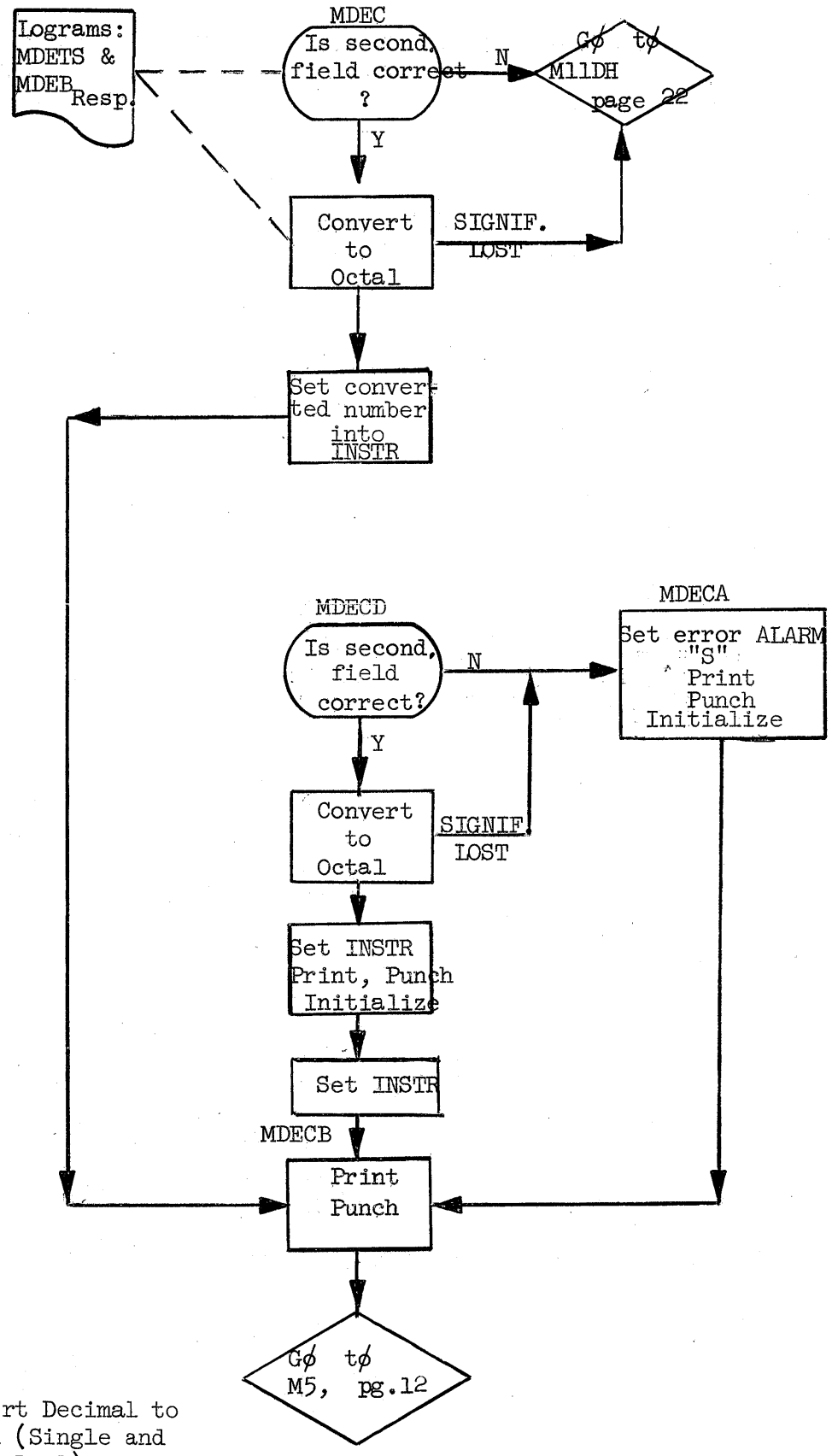
Convert Decimal Integer to Octal
Pseudo Operation ØCT



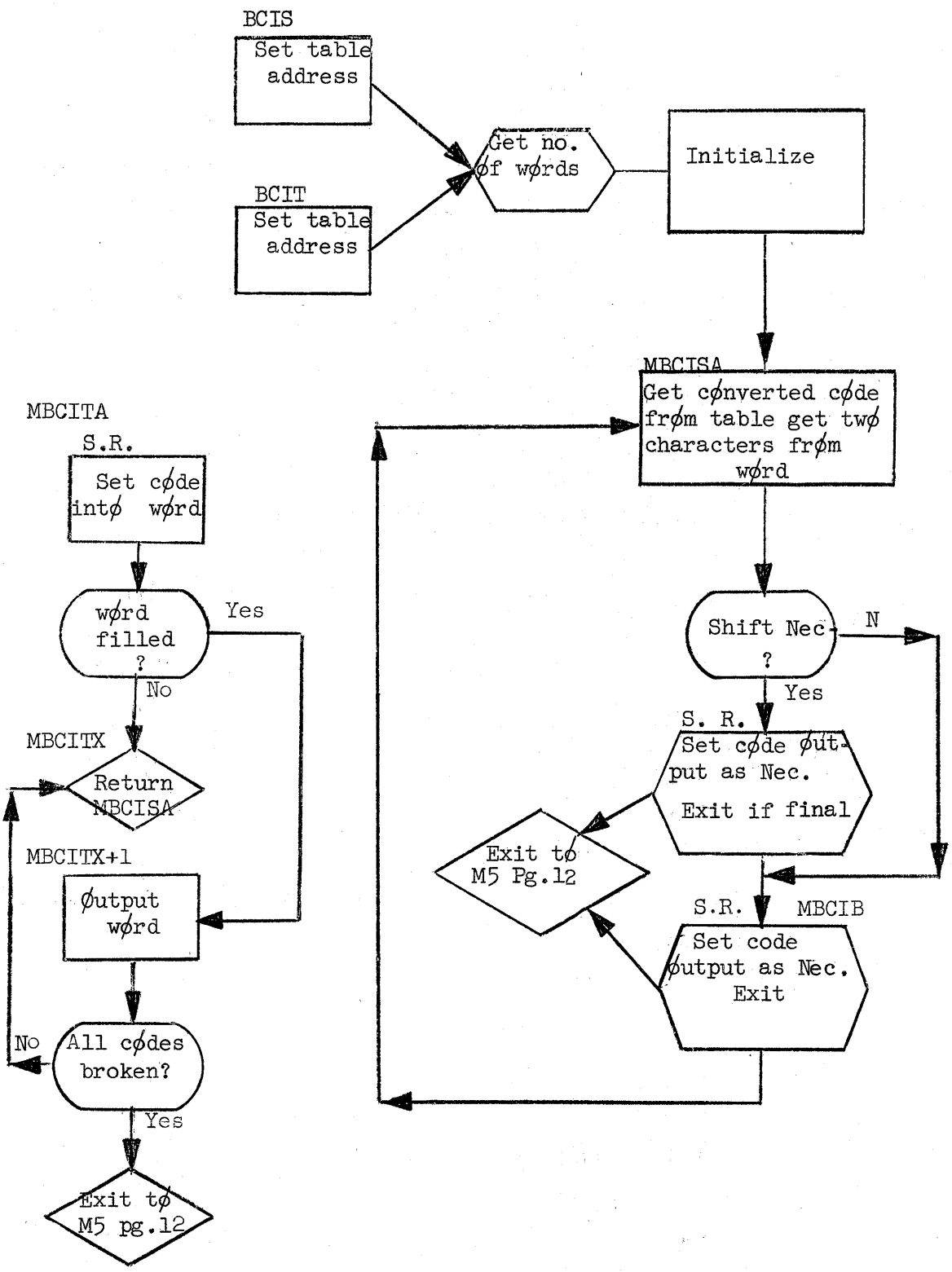
Print under Program
Control (SUP) (PRT)

Set Value in Secondary Field (PZE or BLANK)

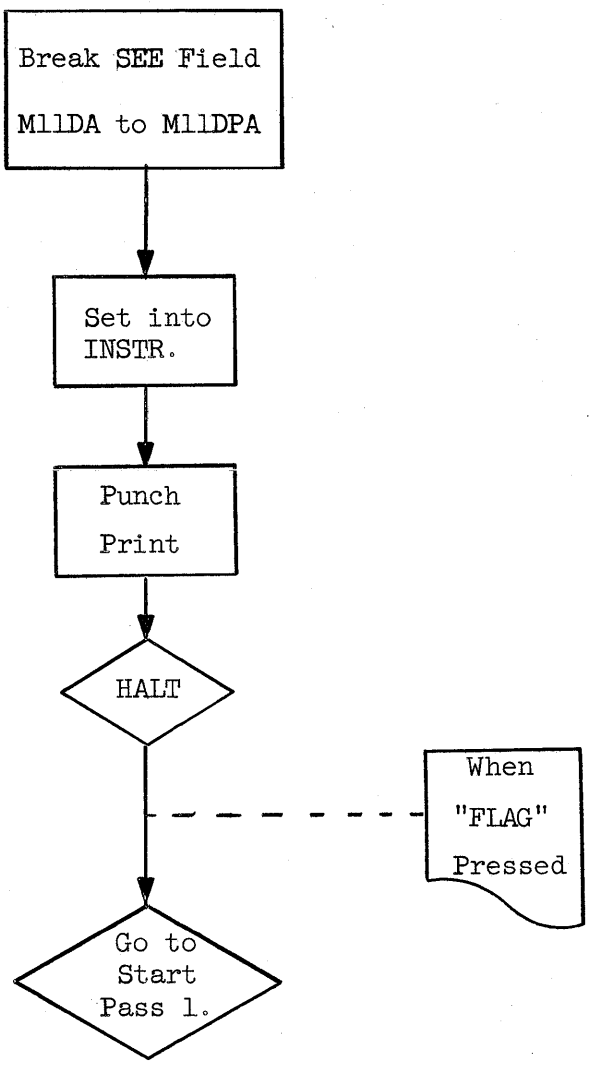




Convert Decimal to Octal (Single and Dual)


BCIS
Set table address
BCIT
Set table address
Get no. of words
Initialize
MBCISA
Get converted code from table get two characters from word
MBCITA
S.R.
Set code into word
word filled ?
Yes
No
MBCITX
Return MBCISA
MBCITX+1
Output word
No
All codes broken?
Yes
Exit to M5 pg.12
Shift Nec ?
N
Yes
S. R.
Set code output as Nec.
Exit if final
Exit to M5 Pg.12
S.R.
MBCIB
Set code output as Nec.
Exit

MEND



END Program

Corrections Lograms for Program Assembly Program

Logram: MPKPC Pack Code

Function: Pack code (bits 1 to 6) of columns 7 and 8 and code in bits 1 to 3 of column 10 into bits 10-15, 4-9 and 1-3, respectively.

Logram: MSETER Relocate Multiple Address Listing

Function: Place the contents of $AL into (MERR) and alter the placing of $AL for the next time thru. That is, add one to the cell addressing (MERR).

Logram: SEARCH

Function: Search the symbol table for the contents of $AL, $AR, $AT. If the symbol is in the table, flag the table entry as governed by the word found in (MSERE+1). If (MSERE+1) = 40000, flag entry with a minus sign. If (MSERE+1) = 00000, leave entry as is.

Logram: MØCØV Convert Octal Integer

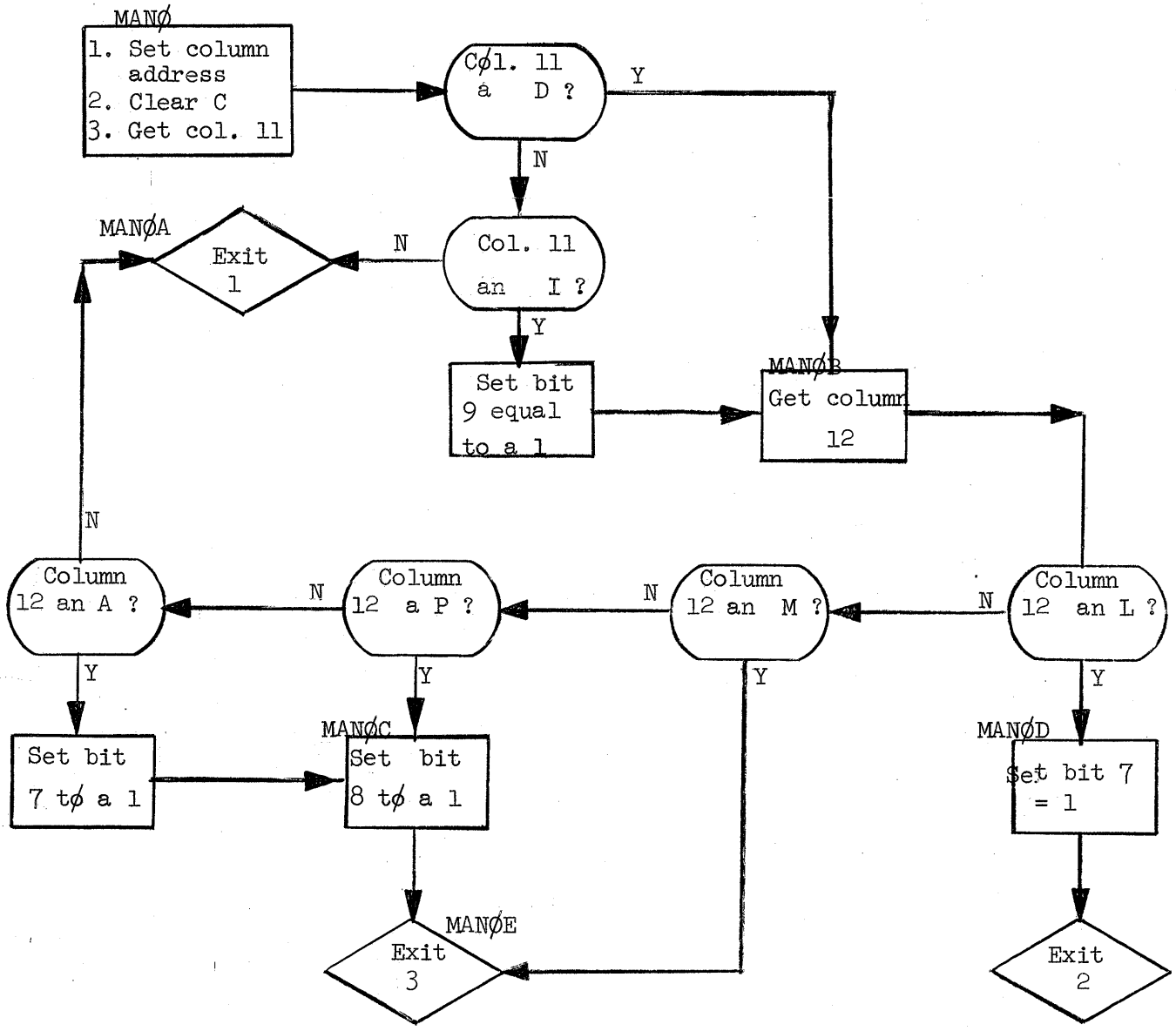
Function: Convert the code found by using ($AL) as an address to a binary number. The codes so found are assumed to be octal numbers. EXIT 1 if a non-numeric or 8 or 9 character occurs before a blank. EXIT 2 when a blank occurs. ($AL) = converted number at time of exit.

Logram: MBCDD Convert Decimal Integer

Function: The same as MØCØV except the number is converted as if it were a decimal number. Exit 1 does not occur for an 8 or 9.

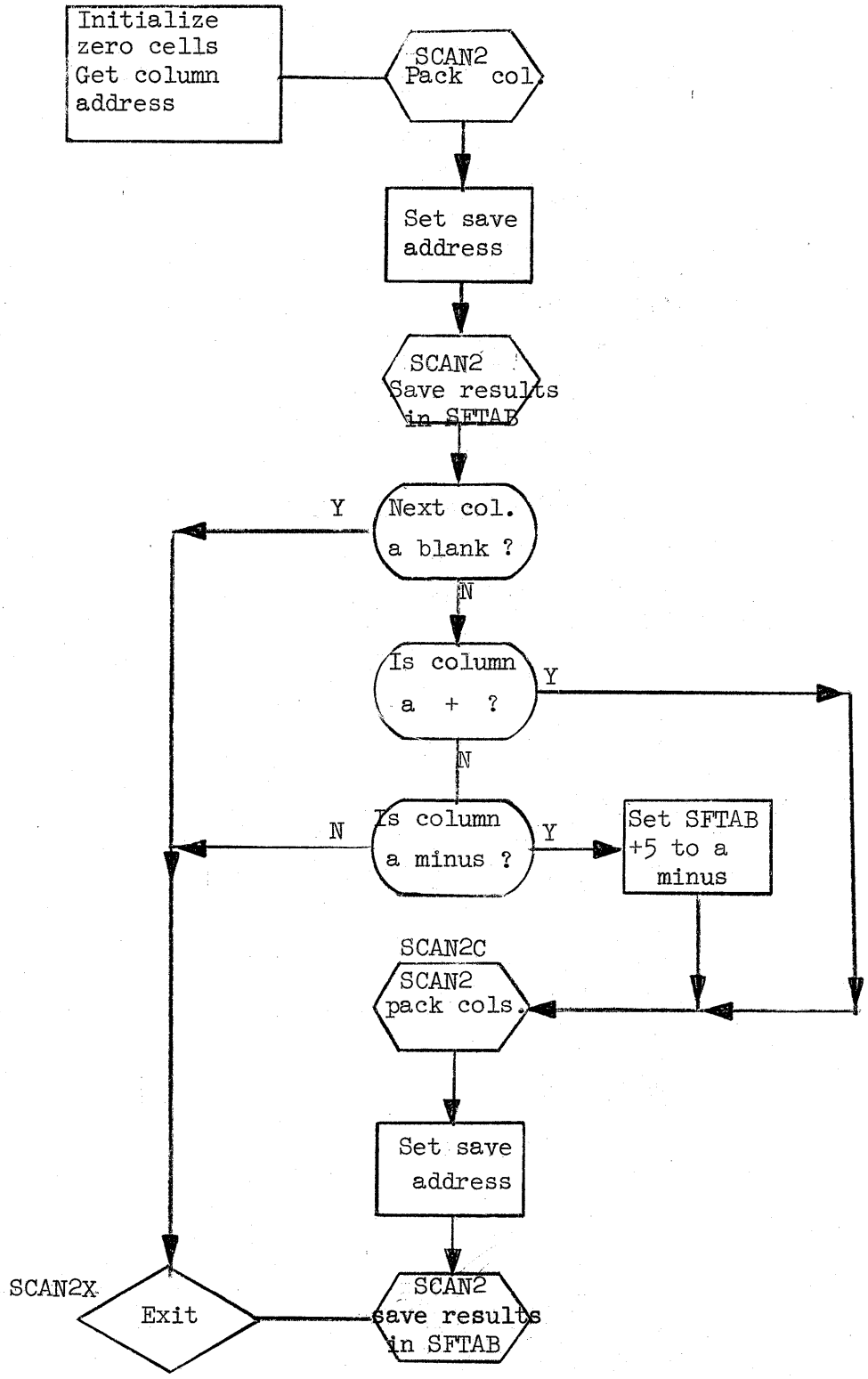
Logram: MANØ Examine Address Option field

Function: Examine address option field, columns 11 and 12. If address option legal set cøde and execute cell addressed by P+2 if IL or DL option. Execute P+3 if any other option. If illegal execute cell addressed by P+1.



Logram: SCAN2 Examine Secondary Field (see p 45 for Table)

Function: Examine secondary field. Set (SFTAB to SFTAB+9) [See construction of this table] Exit at conclusion of routine.



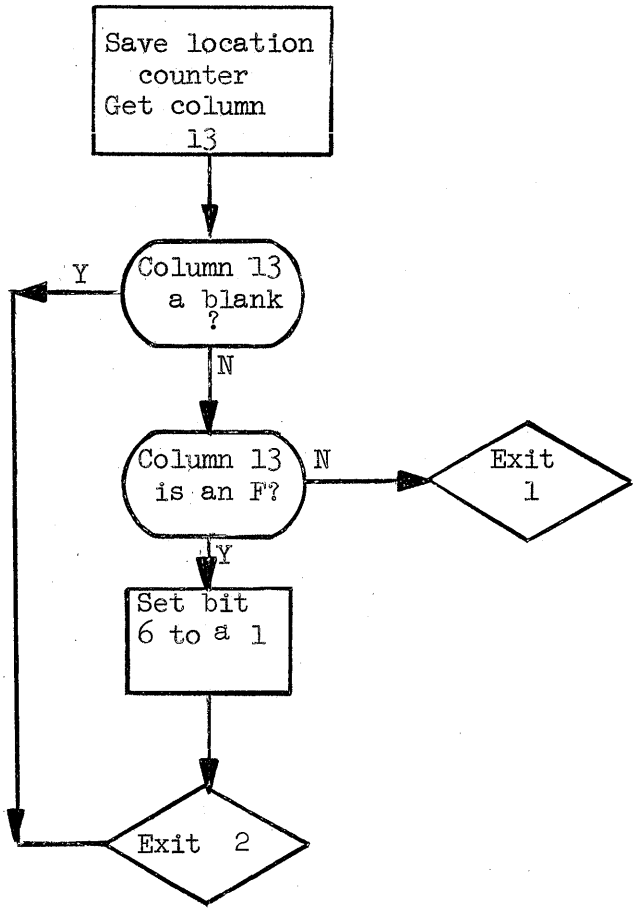
Logram: MSEA Examine Flag Field

Function: Examine C field, column 13 for a blank or an F. Set bit 6 to a 1 if F.

If blank, leave as is.

Execute a jump to the cell addressed by P+1 if not blank or an F.

Execute P+2 if either blank or an F.

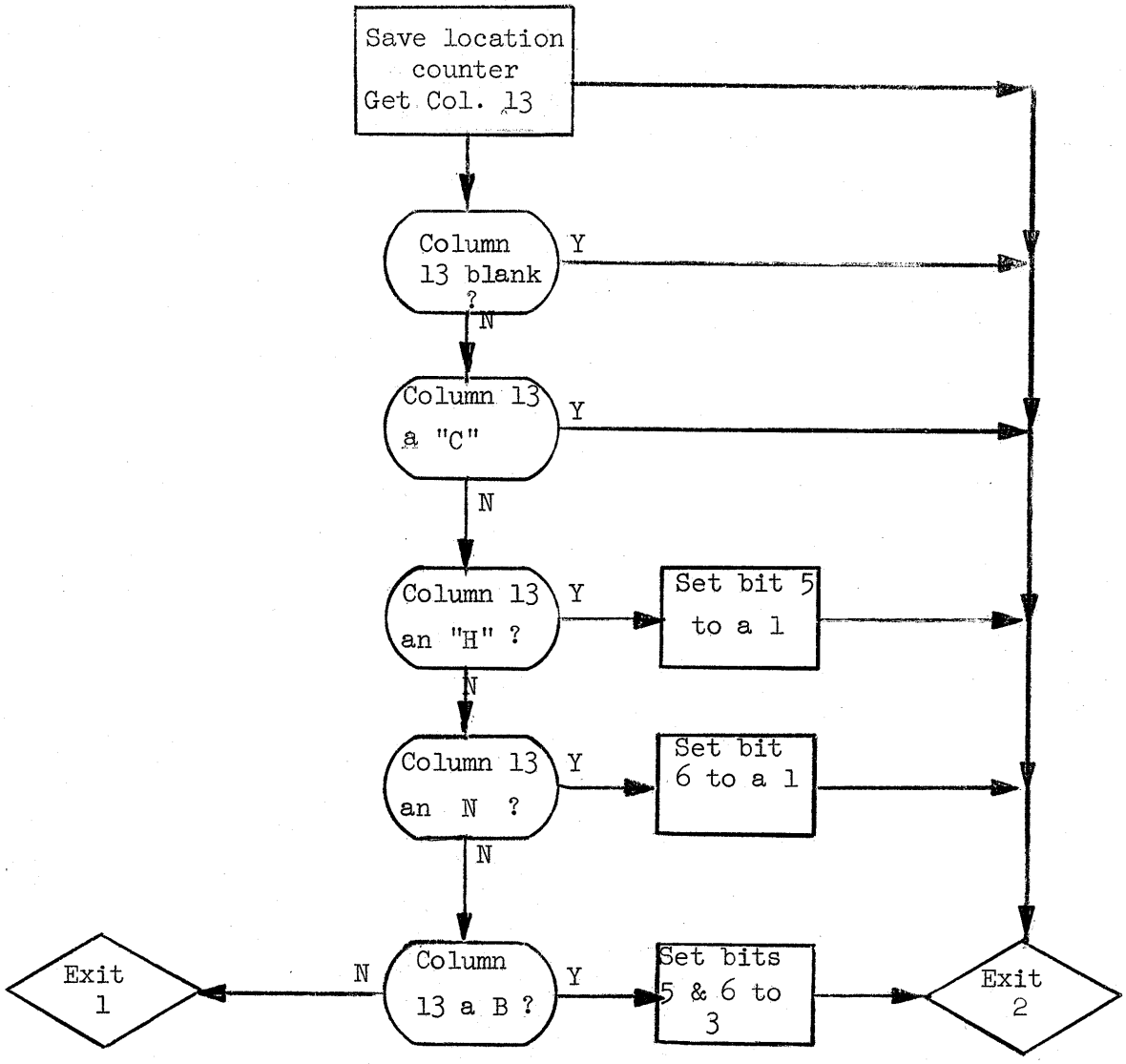


Logram: MSEE Set Control Field

Function: Examine C field, Column 13, for a blank, C, N, B or H. Set code into (INSTR) as follows:

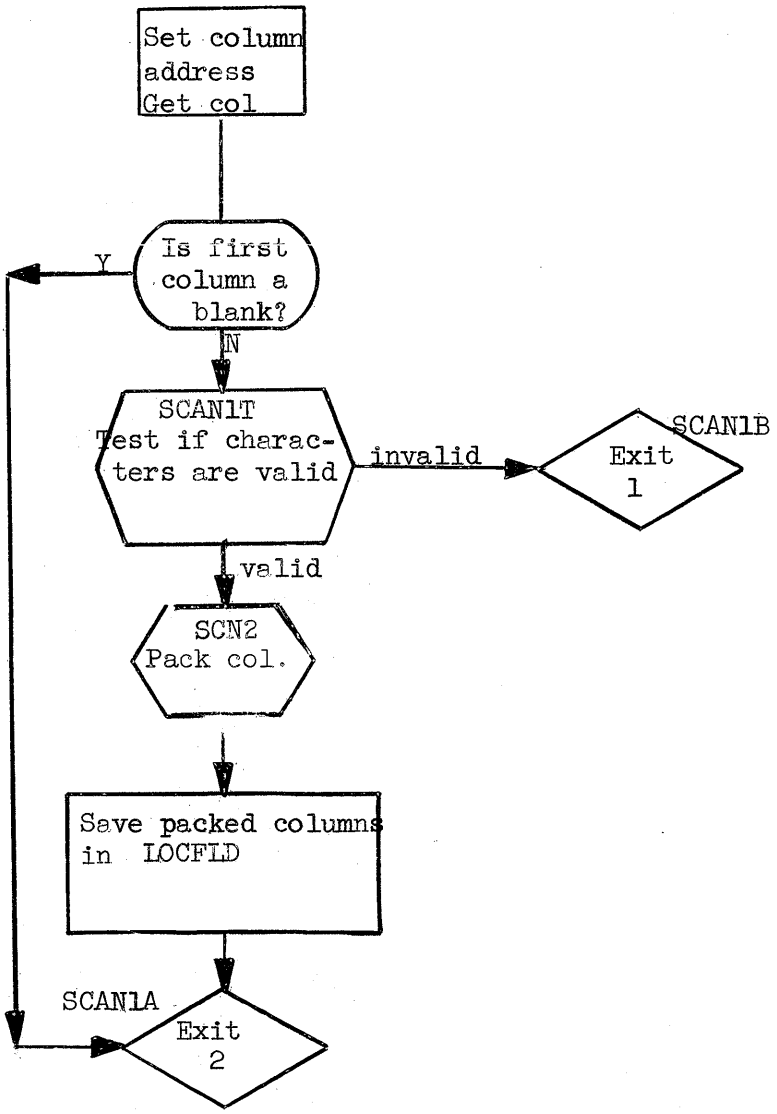
Control field	BIT 5&6
blank	0
C	0
H	1
N	2
B	3

Execute a jump to the cell addressed by P+1 if none of the above.Execute the cell P+2 if any of above.



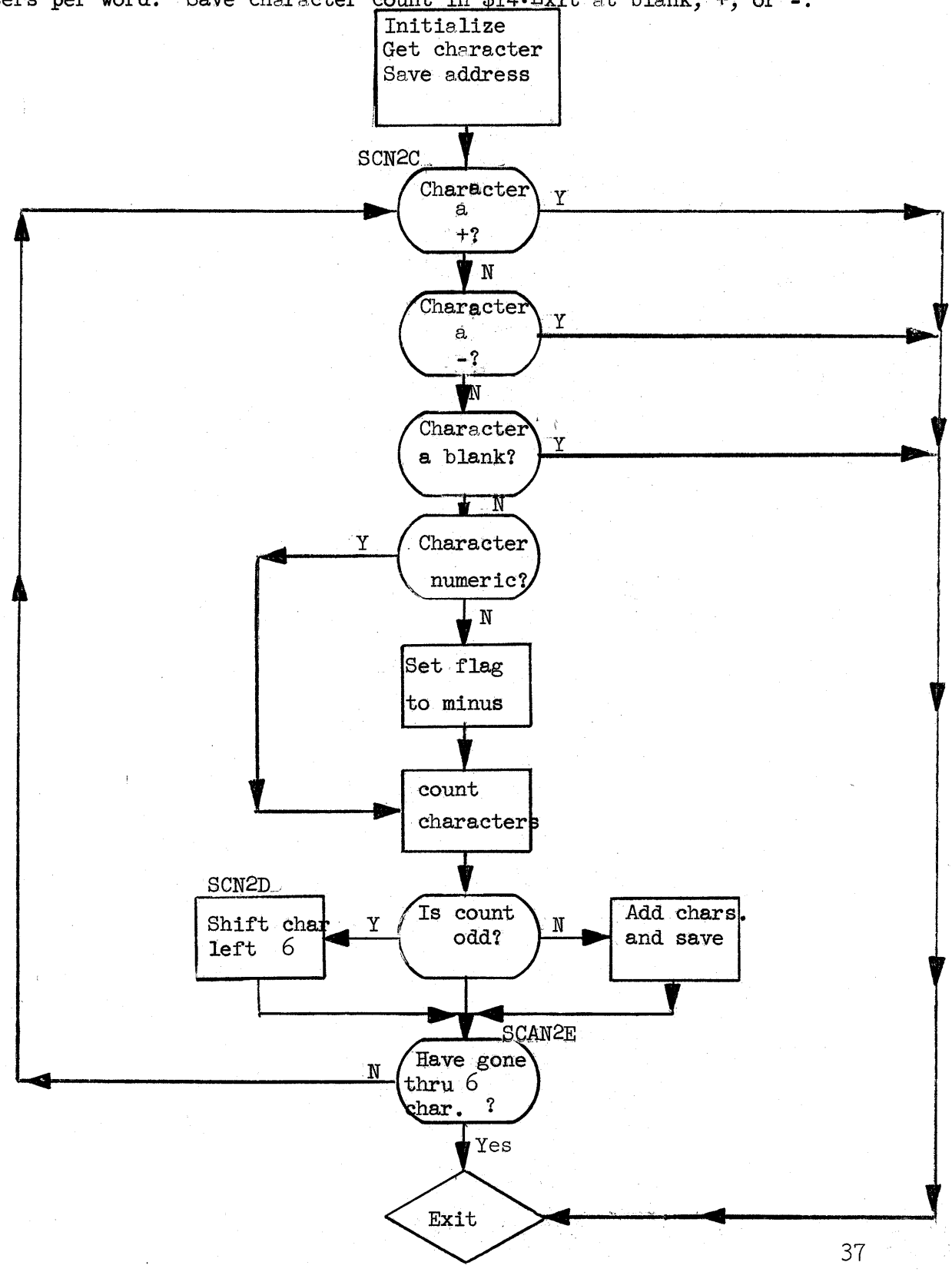
Logram: SCAN1 Examine Location Field

Function: Examine the location field for a valid symbol. Pack the symbol into "LOCFLD" if valid. Exit to cell addressed by P+1 if invalid symbol. Execute cell P+2 if valid.



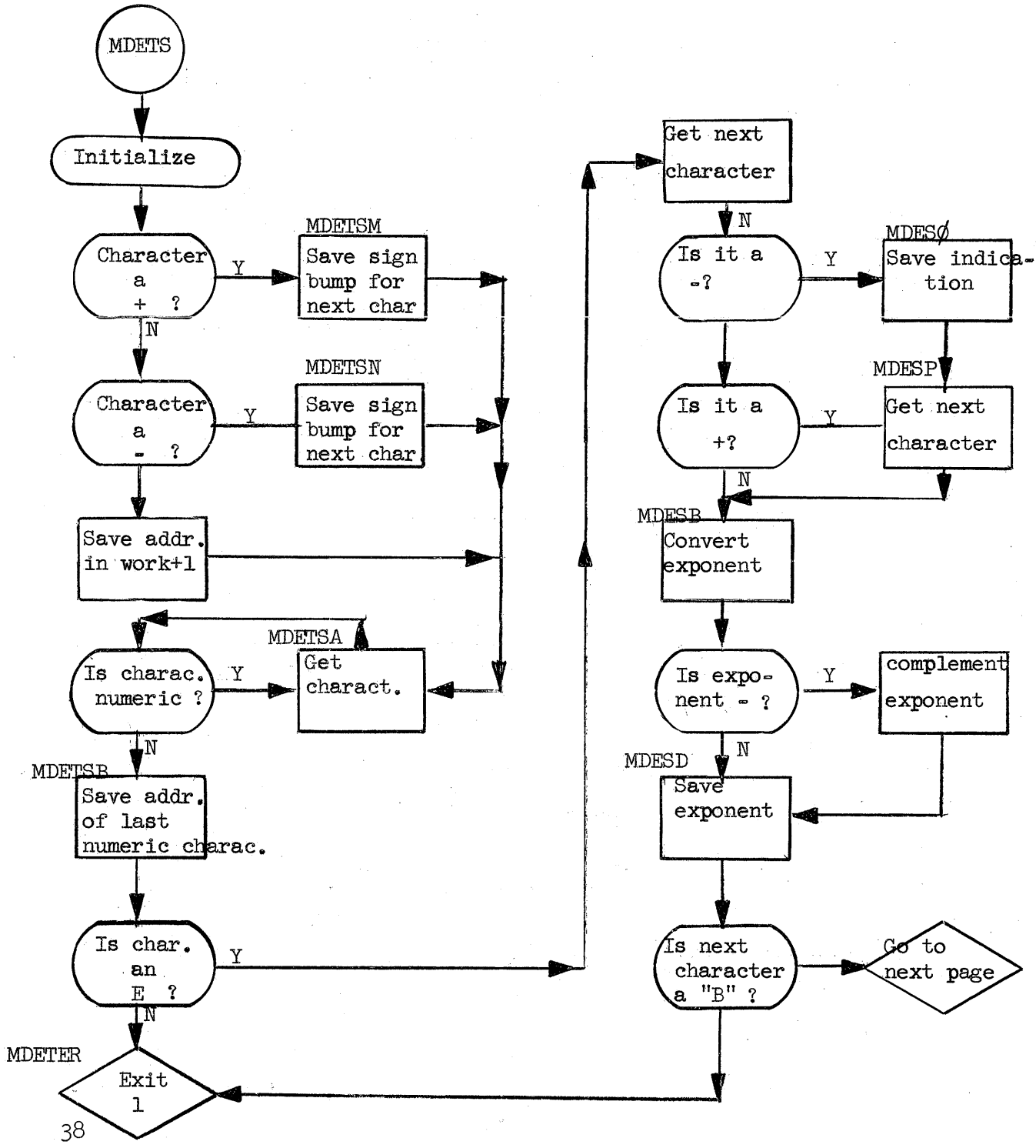
Subroutine: SCN2 Pack Location Field

Function: Pack columns addressed initially by $T1 into $T5, $T6, $T7, two characters per word. Save character count in $T4. Exit at blank, +, or -.

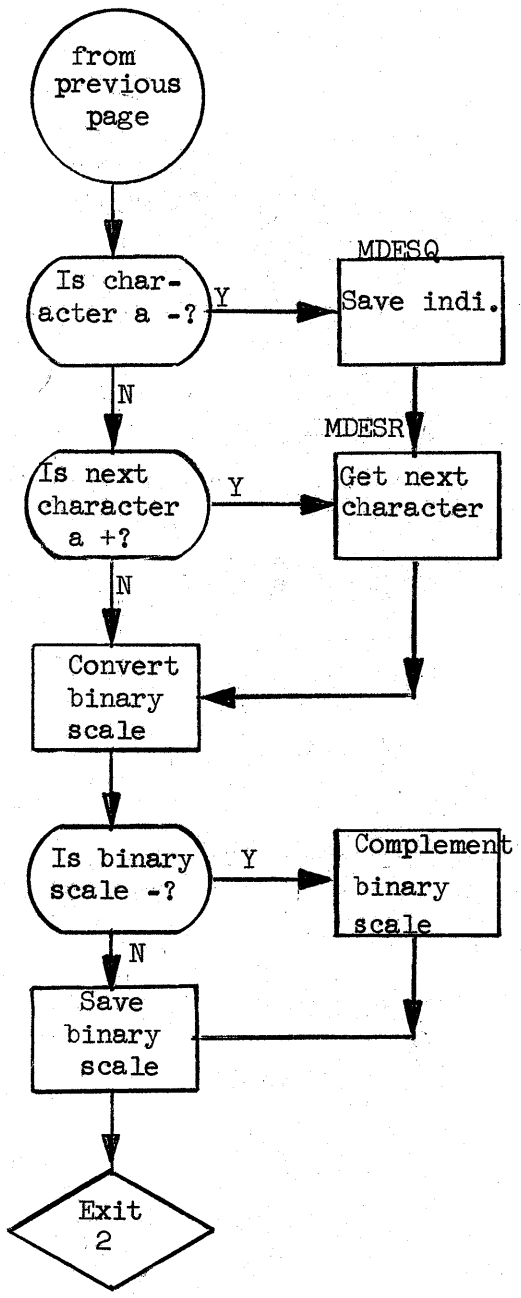


Iogram: MDETS Convert Decimal

Function: Examines secondary field for "DEC" and "DECD" pseudo-operations. Sets address of lst numeric character into (wǿrk+1), the decimal scale into (wǿrk), the address of the last numeric character to be converted into (work+2), the binary scale into (wǿrk+3) and the sign of the converted number into (wǿrk+4). Exit 1 if any illegality, exit 2 if all is copacetic.

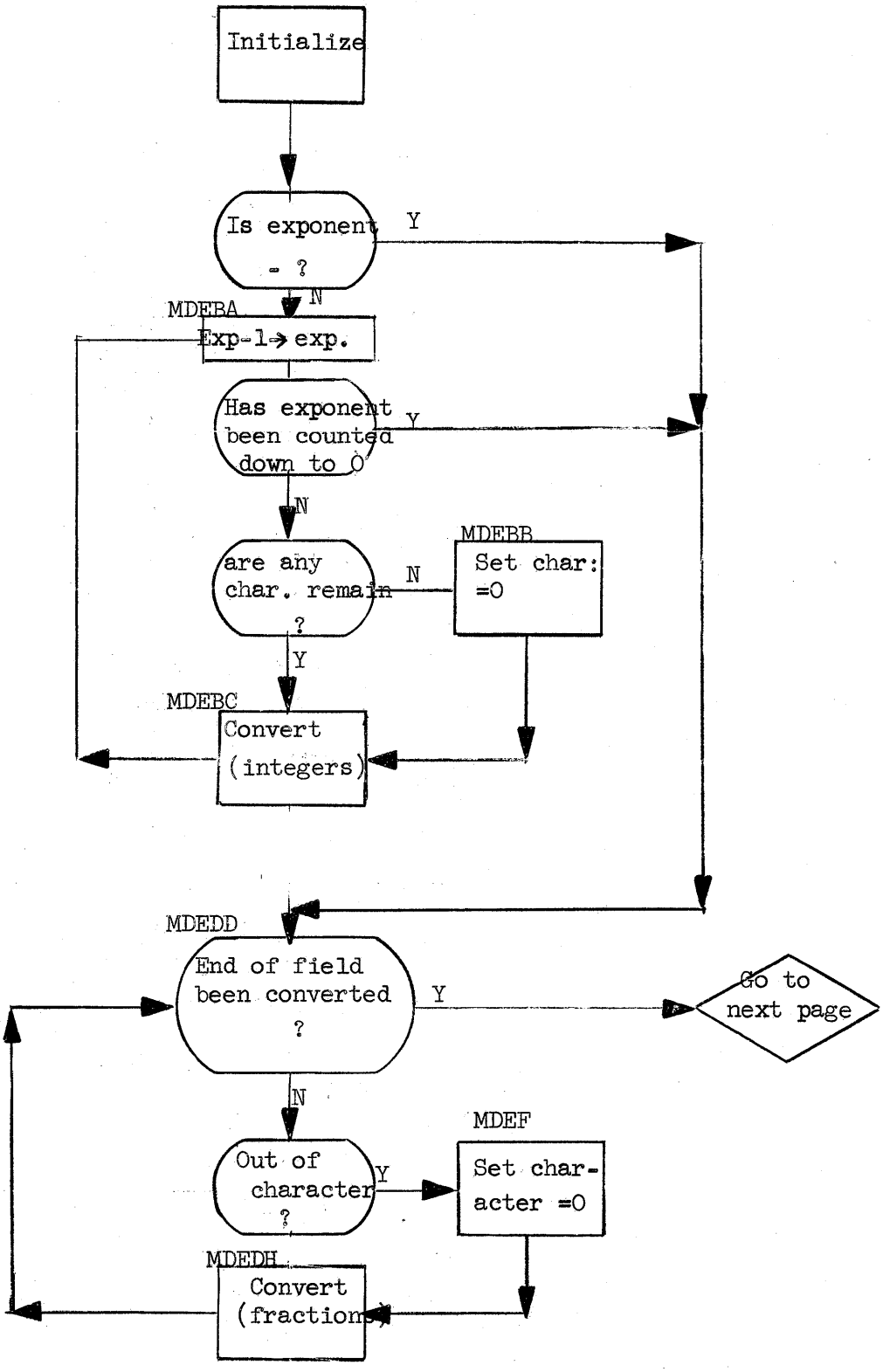


Logram: MDETS, continued

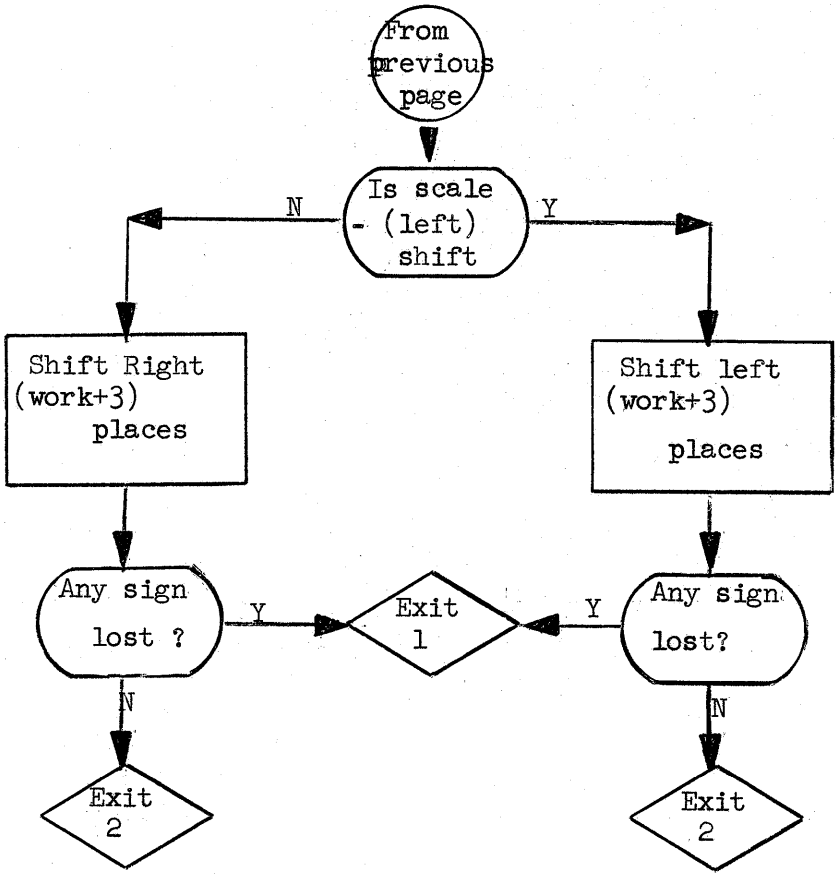


Logram: MDEB Convert SF to Binary, Scale

Function: Convert the numbers in the secondary field to binary and scale. The secondary field is defined by the output of the logram "MDETS".

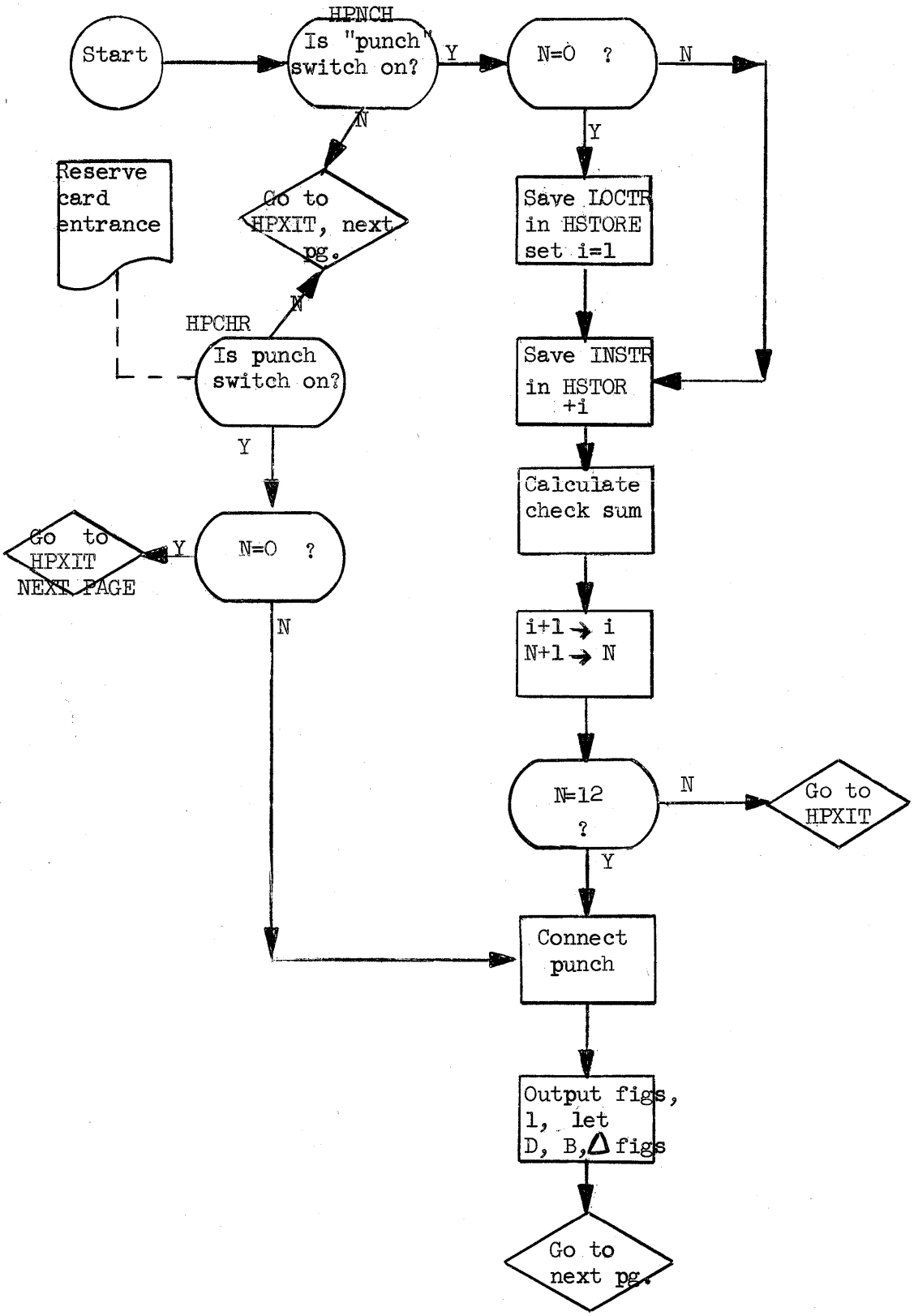


Logram: MDEB, continued

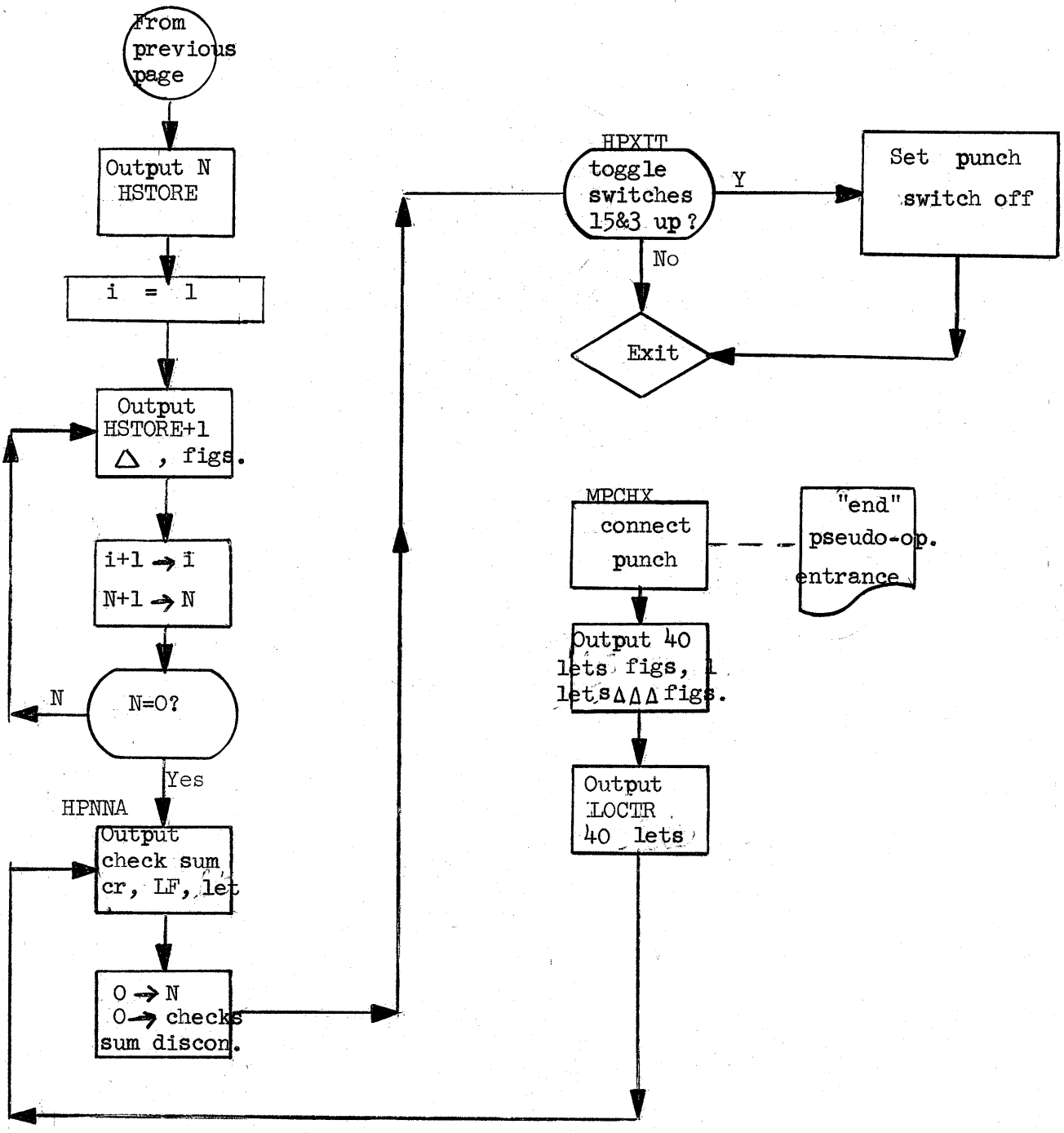


Logram: HPNCH, HPCHR, MPCHX. Punch Tape

Function: To punch a tape suitable for reading.



Logram: HPNCH, HPCHR, MPCHX, continued



Location Table: LOCTAB

This table is loaded by SCAN1. The table consists of 3 words. The words contain the characters found in card columns 1 to 6 inclusive. The three words are zeroed, then filled with none-blank characters until a blank is encountered or all six characters are filled into the table.

15	12	6	1
	c 1	c 2	
	c 3	c 4	
	c 5	c 6	

Ci, $i \leq 6$ is the character in column i.

Location Counter: LOCTR

The location counter for the assembler will be located in the scratchpad area. The mnemonic tag is "LOCTR".

Data Word: INSTR

Each word as it is assembled will be stored into a scratchpad cell tagged "INSTR".

Current Symbol Table Address: SYSLST

The location into which the last symbol was stored is located at mnemonic "SYSLST".

Length of Symbol Table (N)

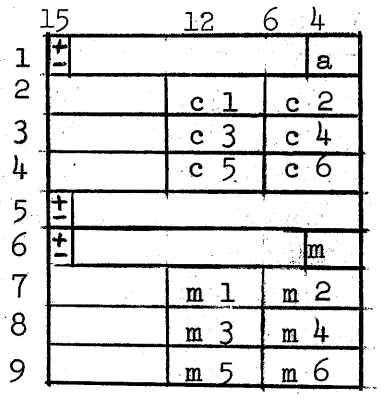
Assembled with the assembly program will be a parameter specifying the length of the symbol table It will be assembled as:

N EQU some number

The symbol table is stored starting at mnemonic "SYMTAB".

Secondary Field Table (SFTAB)

This a nine word table consisting of the address in the secondary field, an operation, and a modifier. The table is constructed as follows:



Word 1: a($\leq$ 6) right adjusted, represents the number of characters following.
If bit 15 = 0 (+) the characters are all numeric.
If bit 15 = 1 (-) the characters are alpha-numeric.

Word 2, 3, 4: are the characters of the address in the secondary field. The 6-a characters at the end are zeros (blanks).

Word 5: If bit 15 is 0 (+), the converted characters in words 2, 3, and 4 are to be be added to the converted characters in words 7, 8, 9. If bit 15 is 1 (-), a subtraction is made instead of an add.

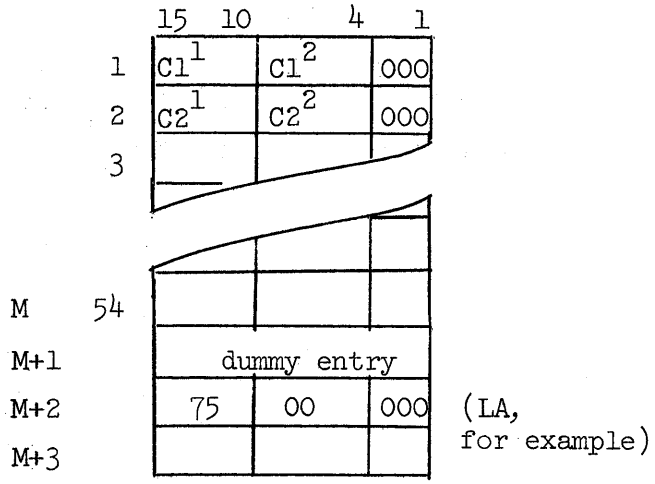
Word 6: This word is similar to word 1 except it gives the character count in words 7, 8 and 9.

Primary Command Table

Mnemonic: MPC

The primary command table is built by first assembling in sequential order the mnemonic code for each of the (54) Primary Commands.

M represents the length of the sequence of entries. The two characters (now in Hollerith code) representing each Primary Command occupy bits 10-15 and 4-9, respectively, in word i of the table. Bits 1-3 are zeros.



At M+1 is a dummy entry for the table lookup logram. At M+2 the instruction code (in binary) for Table Entry 1 is placed in bits 15-10; the associated address option is assembled in bits 9-4.

The instruction code for the i^{th} entry will always be at word M+(i+1).

The address option for every instruction code assembled is DM (00), with these exceptions:

Primary Command	Address Option
MV SR TB MH	IP
BO BI	IA
TM IT	IL
CF	DL

This assembly scheme makes it unnecessary to examine the address option field if a single address option is allowable. In BR and SK instructions, bit 5 is always a 1.

Secondary Command Table

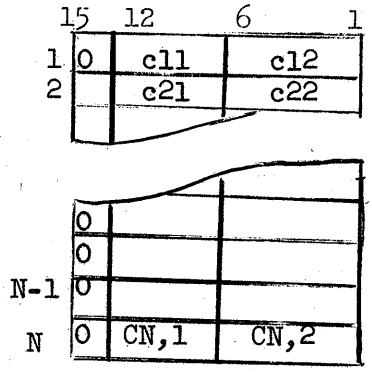
Mnemonic: MSC

The secondary commands are listed in the table in the numerical order of their machine codes, i.e. 00, 01, ... 17. (octal). Thus, the machine code for the command is obtained by subtracting the address of the first table entry from the address of the desired table entry. For this reason, the order in the table is very material.

The two mnemonic characters representing each of the 16 secondary commands occupy columns 14 and 15 of the card. The mnemonic equivalent, in Hollerith code, of the command is stored left to right, viz: A = 21, T = 63, stored as 02163 for secondary command AT.

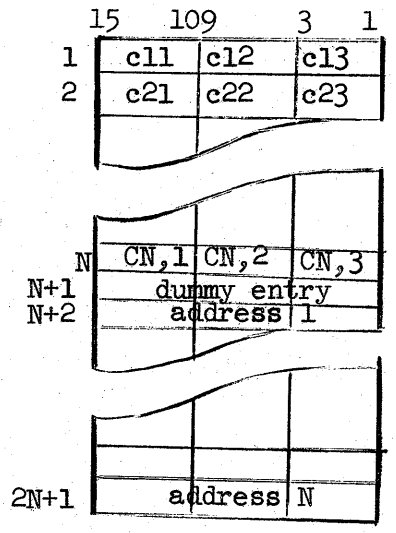
Machine Code	Address	15 12 6 1	Secondary Command
00	1	0 45 46	NO
01	2	0 21 31	AI
02	3	0 43 47	LP
03	4	0 21 43	AL
04	5	0 21 63	AT
.			
.			
.			
17	16		CH

Condition Tables for BR, SK, MV, SR, TB and MH Commands: Mnemonic: MCØND



The construction of this table is similar to the construction of the secondary command table. Cil represents the first character of the mnemonic for the i^{th} condition and Ci2 represents the second character. The i^{th} word contains the mnemonic code for the condition whose machine code equals i. Thus by subtracting the starting address of the table from the address of the appropriate mnemonic one gets the machine code for the condition.

Table for Pseudo-Operation Jump Addresses. Mnemonic: MPSUE



The construction of this table is similar to the construction of the table of primary commands. Ci1, occupying bits 10-15 of the i^{th} word, is the first character of a psuedo-operation and is the code generated by column 7 of the input card. Ci2, occupying bits 4-9, of the i^{th} word, is the second character of a pseudo-operation and is the code generated by column 8 of the input card. Ci3, occupying bits 1-3, of the i^{th} word, is the low order 3 bits of the code generated by column 10 of the input card. Word N+(i+1) of the table is an address at which the pseudo-operation is acted upon. By adding N+1 to the address of the mnemonic one finds an address at which the pseudo - operation is serviced.

Example: Suppose there is only one pseudo-operation called BCIT. The corresponding table entries would be: MCØND → 22 | 23 | 3 / 00 | 00 | 0 / X

N+1=2. Adding 2 to MCOND one gets (MCOND+2)=X. At address X would be further instructions on servicing, such as converting from one BCI to another BCI code.

Table for Conversion to Teletype Code

Mnemonic: MTELE (see

Conversion from card code to teletype code is made by comparisons in the Conversion Table. The teletype code for i is in the low order five bits (bits 1 to 5). Bits 7 to 11 contain the code indicating the shift necessary for the code. For example, suppose i=40 (octal), the card code for "minus". The contents of MTELE+40 is 03330_8. Writing this in binary, we have

0 0 0 0 1 1 0 1 1 0 1 1 0 0 0

teletype code for figure shift — teletype code for minus

Table for Conversion to Soroban Code

Mnemonic: MSORO

The same construction is used in this table as in that for Teletype conversion, except that the Soroban code is used.

An example of Soroban conversion, using 40_8, would yield MSOR+40_8 = 03713_8. In binary it would appear thus:

0 0 0 0 1 1 1 1 1 0 0 1 0 1 1

Soroban code for upper case — Soroban code for minus