EUR 2637.e
VOL. IV

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090

VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR

by

A.F.R. BROWN

1966

Joint Nuclear Research Center
Ispra Establishment - Italy

Scientific Information Processing Center - CETIS

LEGAL NOTICE

This document was prepared under the sponsorship of the Commission of the European Atomic Energy Community (EURATOM).

Neither the EURATOM Commission, its contractors nor any person acting on their behalf :

Make any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this document, or that the use of any information, apparatus, method, or process disclosed in this document may not infringe privately owned rights; or

Assume any liability with respect to the use of, or for damages resulting from the use of any information, apparatus, method or process disclosed in this document.

This report is on sale at the addresses listed on cover page 4

at the price of FF 12,50 FB 125,— DM 10,— Lit. 1 560 Fl. 9,—

When ordering, please quote the EUR number and the title, which are indicated on the cover of each report.

Printed by Guyot, s.a.
Brussels, January 1966

This document was reproduced on the basis of the best available copy.

EUR 2637.e

VOL. IV

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090
VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR
by A.F.R. BROWN

European Atomic Energy Community - EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Information Processing Center - CETIS
Brussels, January 1966 - 100 Pages - FB 125

In the field of non-numerical data processing it is often more profitable to use a medium-size computer instead of a big one. Compilation, however, may better be done on a bigger machine.

The four volumes of this report describe a symbolic programming language, its compiler for the IBM 7090 which produces IBM 1401 object programs, and a simulator permitting the execution of these programs on the IBM 7090.

EUR 2637.e

VOL. IV

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090
VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR
by A.F.R. BROWN

European Atomic Energy Community - EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Information Processing Center - CETIS
Brussels, January 1966 - 100 Pages - FB 125

In the field of non-numerical data processing it is often more profitable to use a medium-size computer instead of a big one. Compilation, however, may better be done on a bigger machine.

The four volumes of this report describe a symbolic programming language, its compiler for the IBM 7090 which produces IBM 1401 object programs, and a simulator permitting the execution of these programs on the IBM 7090.

EUR 2637.e

VOL. IV

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090
VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR
by A.F.R. BROWN

European Atomic Energy Community - EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Information Processing Center - CETIS
Brussels, January 1966 - 100 Pages - FB 125

In the field of non-numerical data processing it is often more profitable to use a medium-size computer instead of a big one. Compilation, however, may better be done on a bigger machine.

The four volumes of this report describe a symbolic programming language, its compiler for the IBM 7090 which produces IBM 1401 object programs, and a simulator permitting the execution of these programs on the IBM 7090.

On sale at
PRESSES ACADEMIQUES EUROPEENNES
98, Chaussée de Charleroi, Bruxelles 6

EUR 2637.e

VOL. IV

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090
VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR
by A.F.R. BROWN

European Atomic Energy Community - EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Scientific Information Processing Center - CETIS
Brussels, January 1966 - 100 Pages - FB 125

In the field of non-numerical data processing it is often more profitable to use a medium-size computer instead of a big one. Compilation, however, may better be done on a bigger machine.

The four volumes of this report describe a symbolic programming language, its compiler for the IBM 7090 which produces IBM 1401 object programs, and a simulator permitting the execution of these programs on the IBM 7090.

This, the last of the four volumes of this report, contains in its first part the flow charts representing the compiler system and the subroutines in the IBM 1401 program package that handle magnetic tape input and output. All these flow charts are described in volume 2 of the report.

The second part of the present report consists of the flow charts of the program, which simulates 1401 operations on the 7090. Comments on these flow charts are given in the third volume.

This, the last of the four volumes of this report, contains in its first part the flow charts representing the compiler system and the subroutines in the IBM 1401 program package that handle magnetic tape input and output. All these flow charts are described in volume 2 of the report.

The second part of the present report consists of the flow charts of the program, which simulates 1401 operations on the 7090. Comments on these flow charts are given in the third volume.

This, the last of the four volumes of this report, contains in its first part the flow charts representing the compiler system and the subroutines in the IBM 1401 program package that handle magnetic tape input and output. All these flow charts are described in volume 2 of the report.

The second part of the present report consists of the flow charts of the program, which simulates 1401 operations on the 7090. Comments on these flow charts are given in the third volume.

This, the last of the four volumes of this report, contains in its first part the flow charts representing the compiler system and the subroutines in the IBM 1401 program package that handle magnetic tape input and output. All these flow charts are described in volume 2 of the report.

The second part of the present report consists of the flow charts of the program, which simulates 1401 operations on the 7090. Comments on these flow charts are given in the third volume.

EUR 2637.e
VOL. IV

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

THE COMPILATION AND PROCESSING OF IBM 1401 PROGRAMS ON IBM 7090

VOL. IV : FLOW CHARTS OF THE COMPILER AND OF THE SIMULATOR

by

A.F.R. BROWN

1966

Joint Nuclear Research Center
Ispra Establishment - Italy

Scientific Information Processing Center - CETIS

TABLE OF CONTENTS

SUMMARY

In the field of non-numerical data processing it is often more profitable to use a medium-size computer instead of a big one. Compilation, however, may better be done on a bigger machine.

The four volumes of this report describe a symbolic programming language, its compiler for the IBM 7090 which produces IBM 1401 object programs, and a simulator permitting the execution of these programs on the IBM 7090.

This, the last of the four volumes of this report, contains in its first part the flow charts representing the compiler system and the subroutines in the IBM 1401 program package that handle magnetic tape input and output. All these flow charts are described in volume 2 of the report.

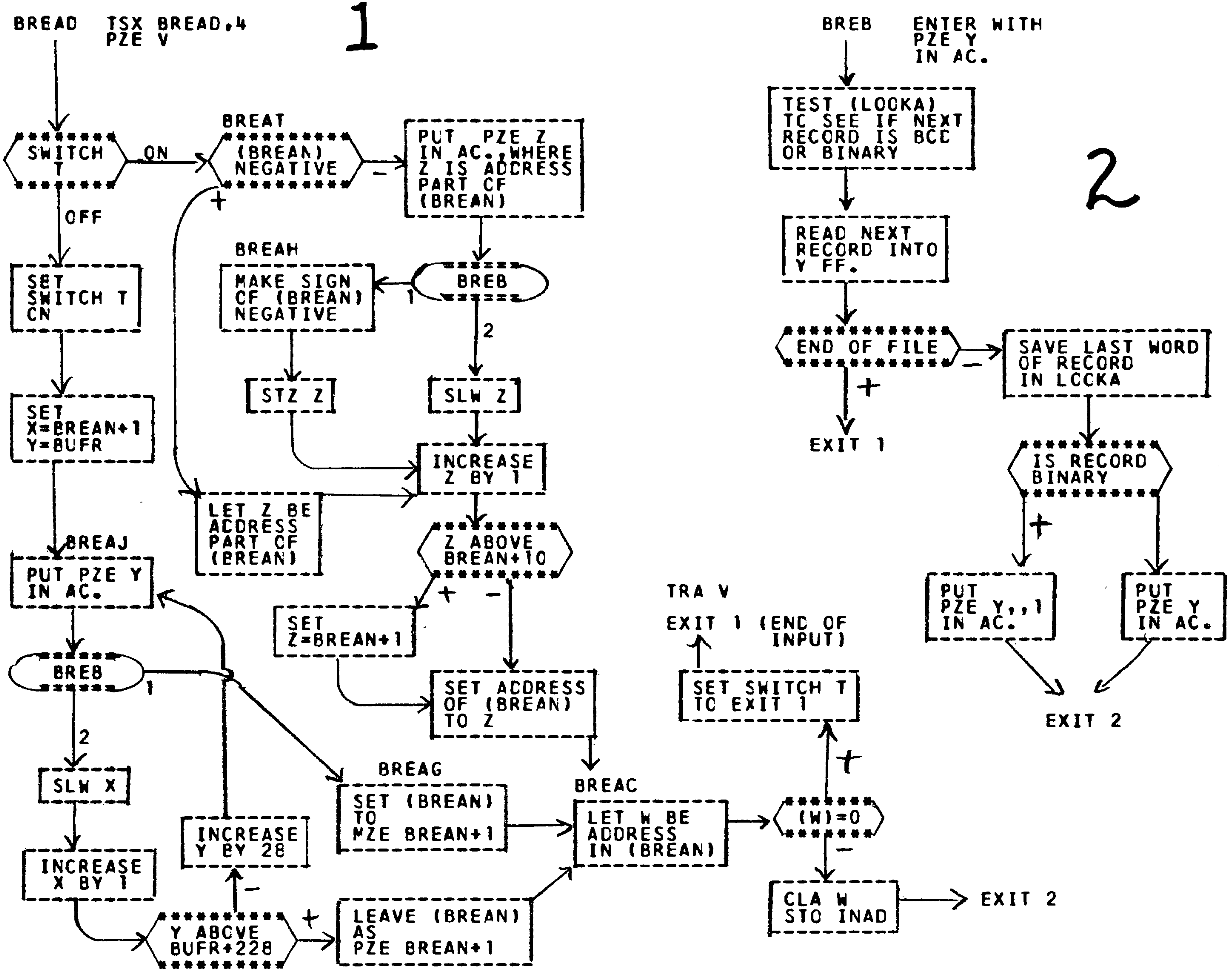
The second part of the present report consists of the flow charts of the program, which simulates 1401 operations on the 7090. Comments on these flow charts are given in the third volume.

The Flow Charts of the Compiler System

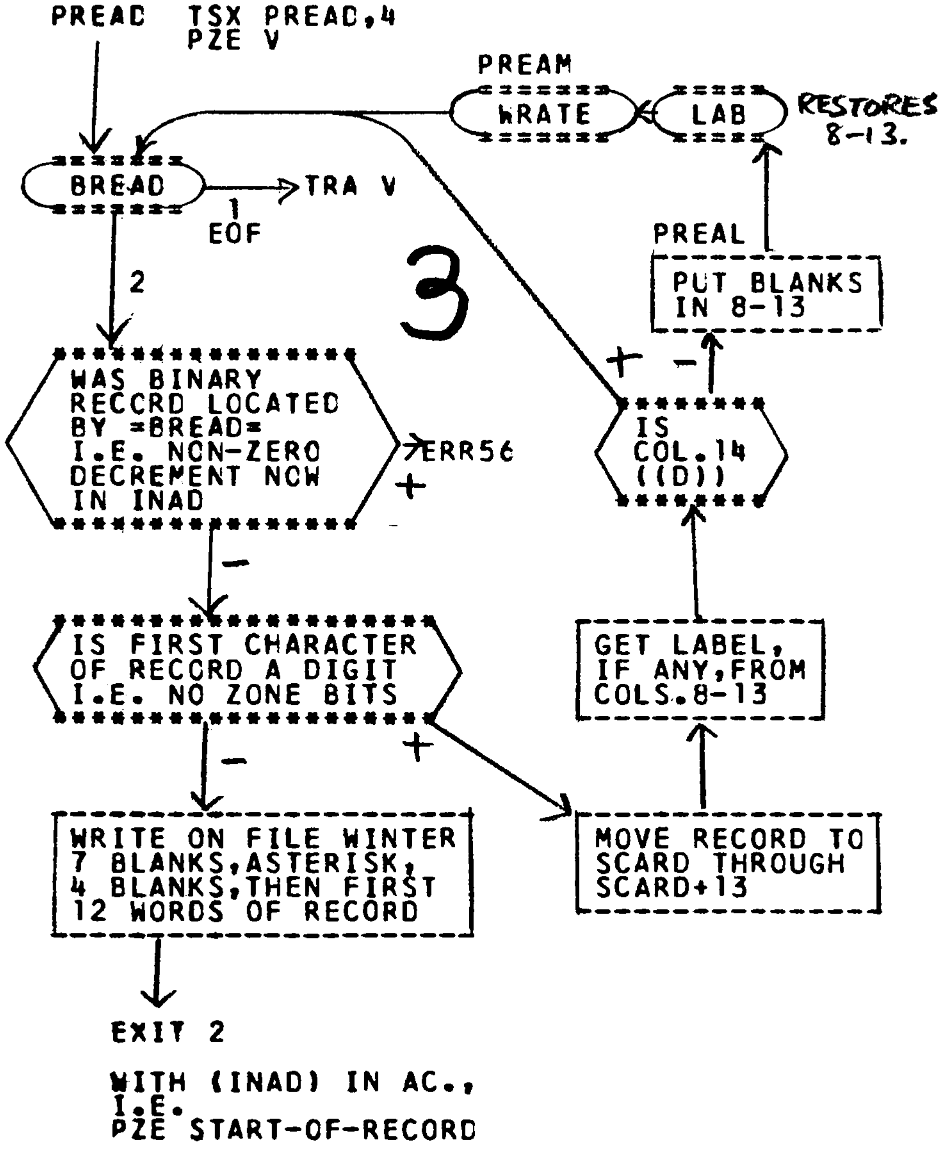
In order to be able to regard simultaneously the flow charts and the comments given on them in the second and third volumes, the flow charts are published separately in the present volume.

Pages 2-56 contain the flow charts explaining the IBM 7090 program which turns the statements of the compiler language, presented in the first volume of this report, into IBM 1401 instructions. Comments on these flow charts are to be found on pages 1-113 of the second volume of this report.

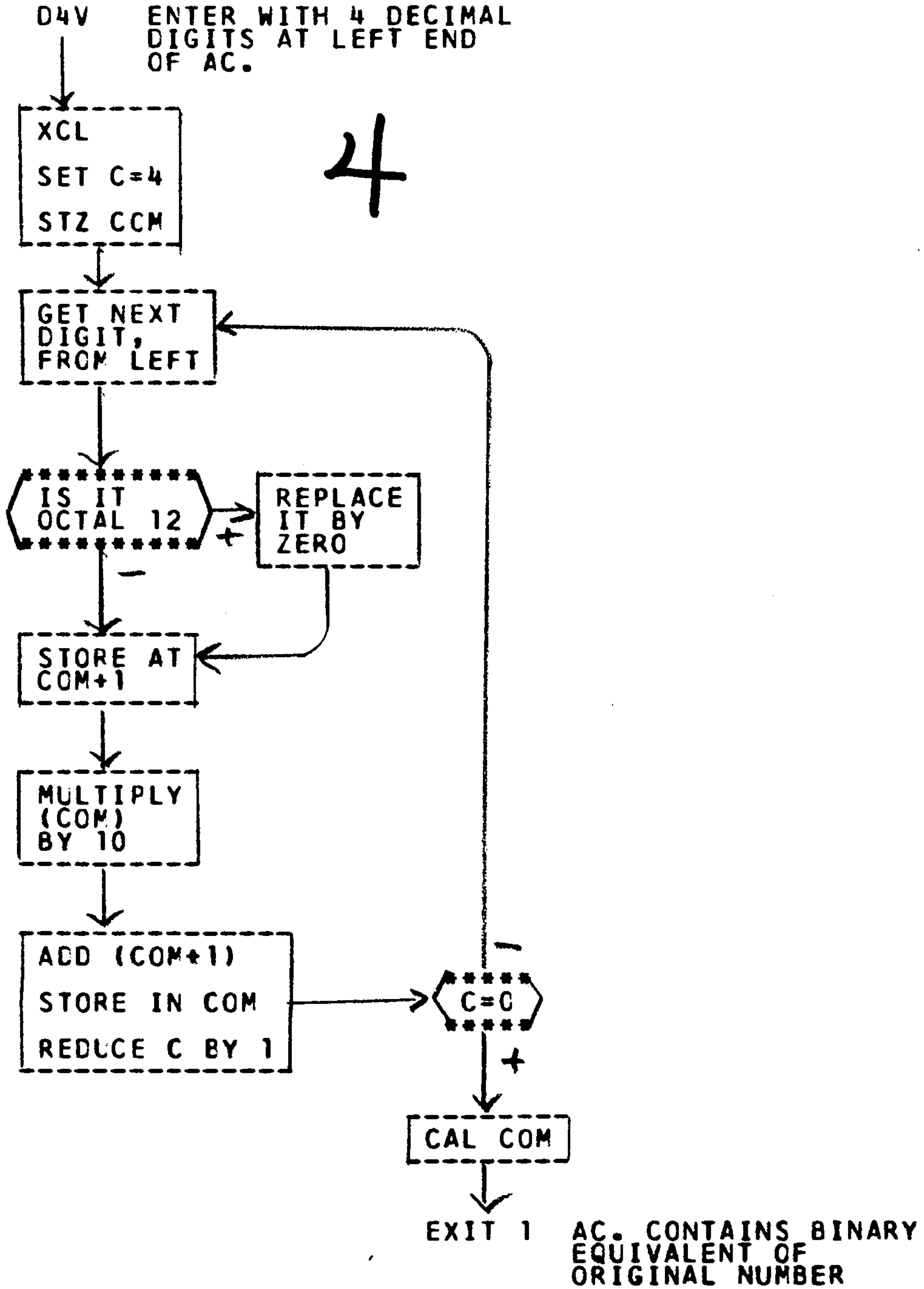
Manuscript received on October 27, 1965.

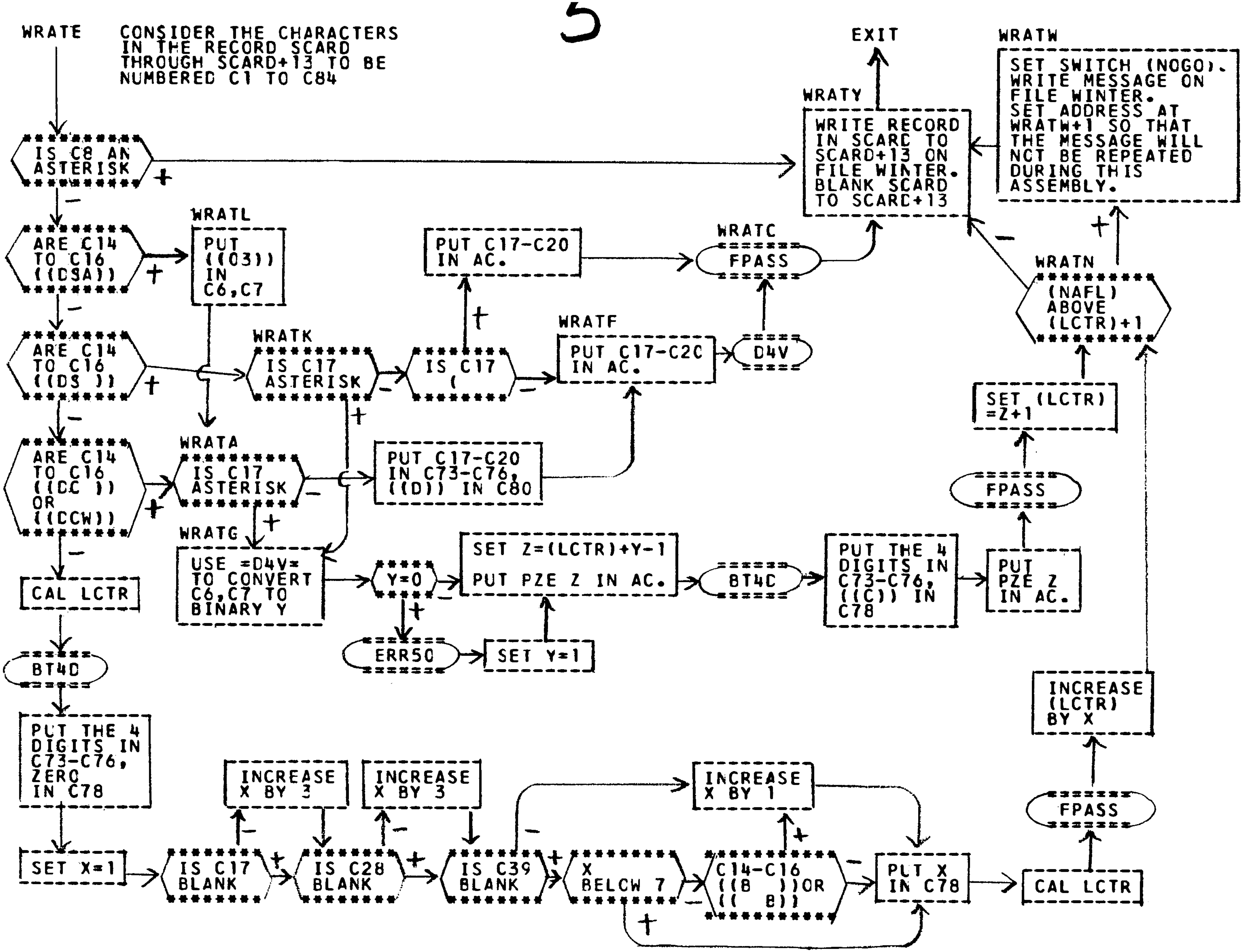

1
BREAD TSX BREAD,4
PZE V
SWITCH T
ON
OFF
SET SWITCH T CN
SET X=BREAN+1 Y=BUFR
BREAJ
PUT PZE Y IN AC.
BREB
1
2
SLW X
INCREASE X BY 1
Y ABCVE BUFR+228
+
-
INCREASE Y BY 28
LEAVE (BREAN) AS PZE BREAN+1
BREAT
(BREAN) NEGATIVE
+
-
PUT PZE Z IN AC.,WHERE Z IS ADDRESS PART CF (BREAN)
BREAH
MAKE SIGN CF (BREAN) NEGATIVE
BREB
1
2
STZ Z
SLW Z
INCREASE Z BY 1
LET Z BE ADDRESS PART CF (BREAN)
Z ABOVE BREAN+10
+
-
SET Z=BREAN+1
SET ADDRESS OF (BREAN) TO Z
BREAG
SET (BREAN) TO MZE BREAN+1
BREAC
LET W BE ADDRESS IN (BREAN)
(W)=0
+
-
SET SWITCH T TO EXIT 1
TRA V
EXIT 1 (END OF INPUT)
CLA W STO INAD
EXIT 2
2
BREB ENTER WITH PZE Y IN AC.
TEST (LOOKA) TC SEE IF NEXT RECORD IS BCC OR BINARY
READ NEXT RECORD INTO Y FF.
END OF FILE
+
-
EXIT 1
SAVE LAST WORD OF RECORD IN LCCKA
IS RECORD BINARY
+
PUT PZE Y,,1 IN AC.
PUT PZE Y IN AC.
EXIT 2

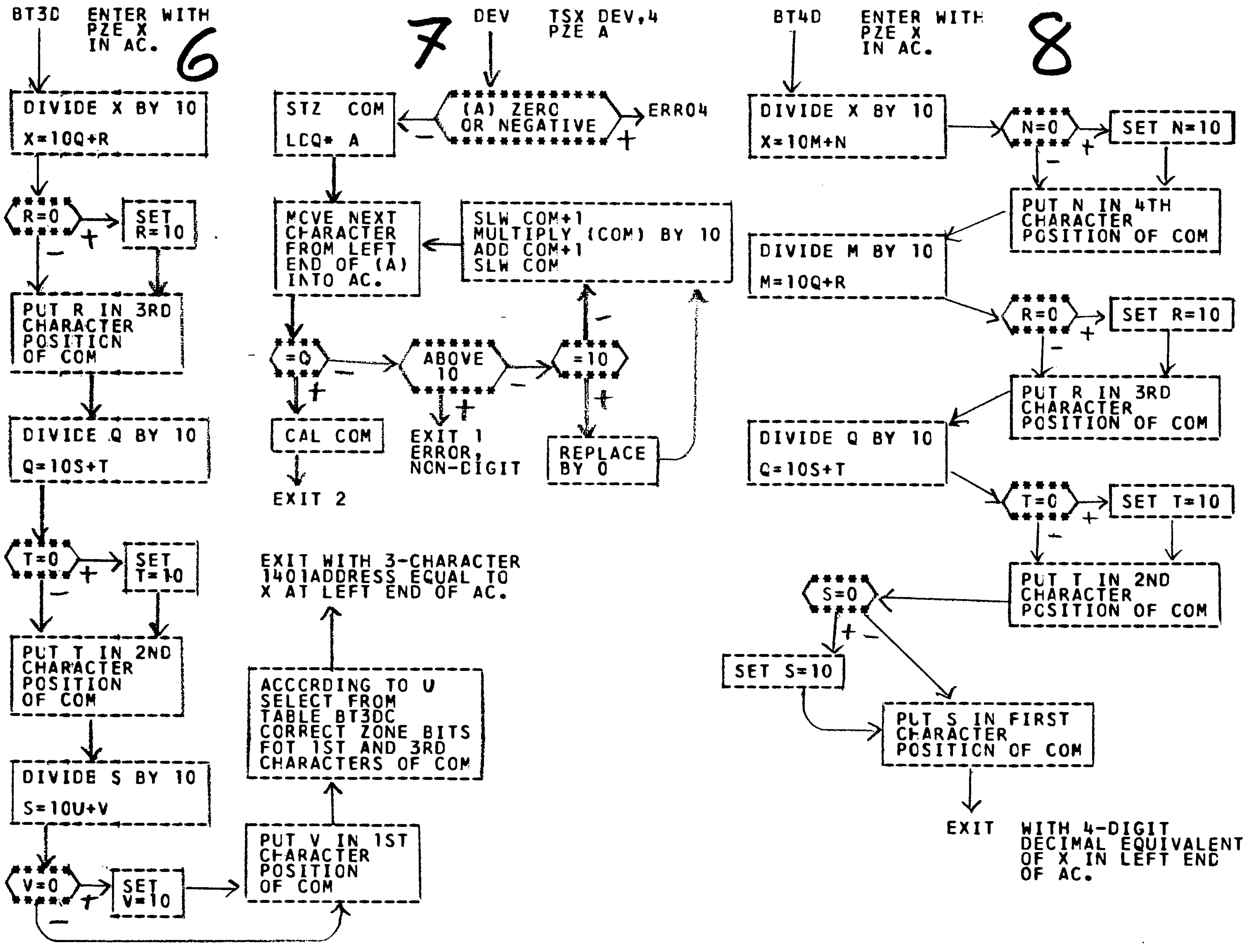
3


PREAD TSX PREAD,4
PZE V
PREAM
WRATE
LAB
RESTORES 8-13.
BREAD
TRA V
1 EOF
2
WAS BINARY RECORD LOCATED BY =BREAD= I.E. NON-ZERO DECREMENT NOW IN INAD
ERR56
+
-
IS FIRST CHARACTER OF RECORD A DIGIT I.E. NO ZONE BITS
-
+
WRITE ON FILE WINTER 7 BLANKS,ASTERISK, 4 BLANKS,THEN FIRST 12 WORDS OF RECORD
EXIT 2
WITH (INAD) IN AC., I.E. PZE START-OF-RECORD
MOVE RECORD TO SCARD THROUGH SCARD+13
GET LABEL, IF ANY,FROM COLS.8-13
IS COL.14 ((D))
+
-
PUT BLANKS IN 8-13
PREAL

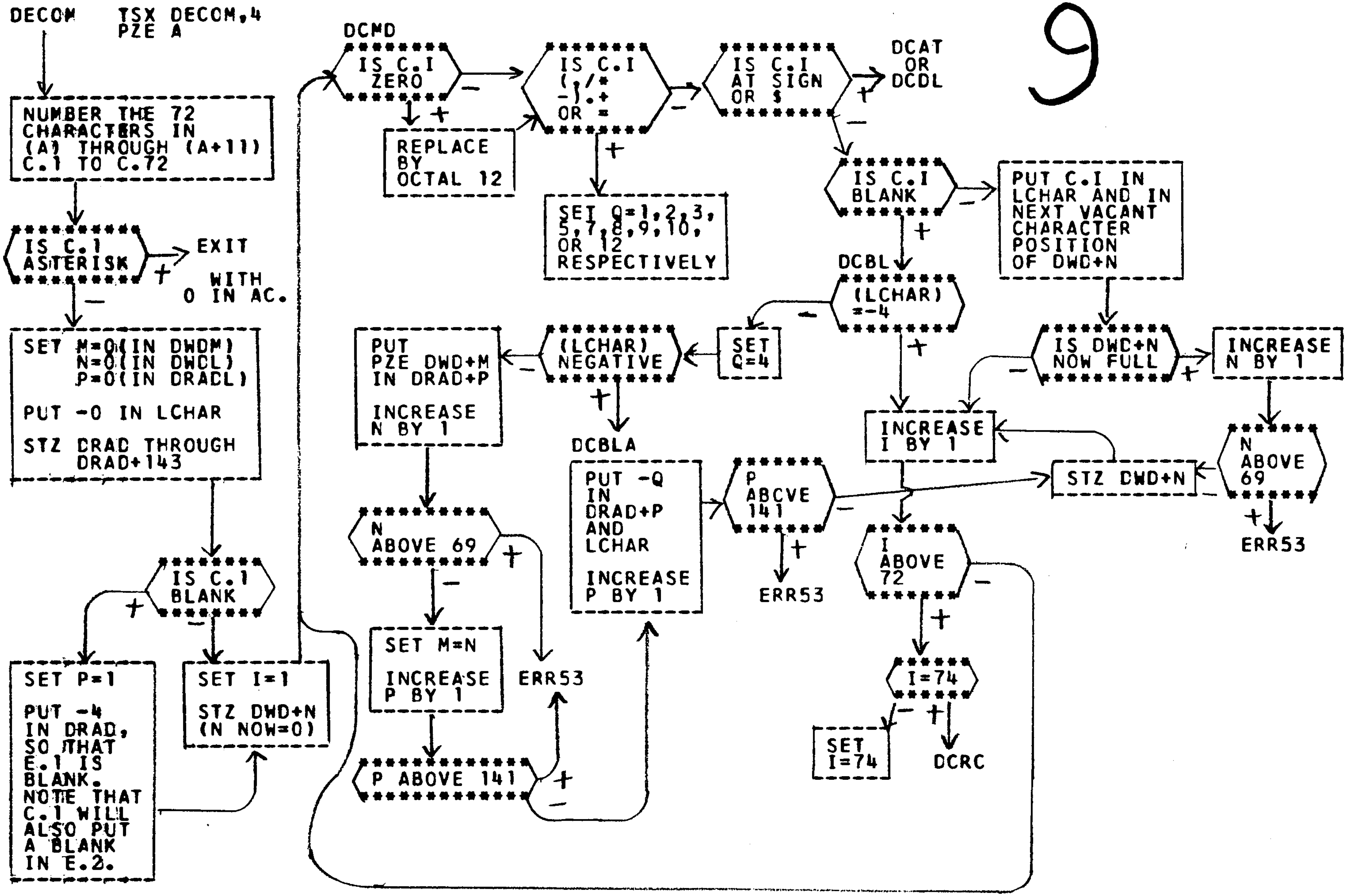
4


D4V ENTER WITH 4 DECIMAL DIGITS AT LEFT END OF AC.
XCL
SET C=4
STZ COM
GET NEXT DIGIT, FROM LEFT
IS IT OCTAL 12
+
REPLACE IT BY ZERO
-
STORE AT COM+1
MULTIPLY (COM) BY 10
ADD (COM+1)
STORE IN COM
REDUCE C BY 1
C=0
-
+
CAL COM
EXIT 1 AC. CONTAINS BINARY EQUIVALENT OF ORIGINAL NUMBER

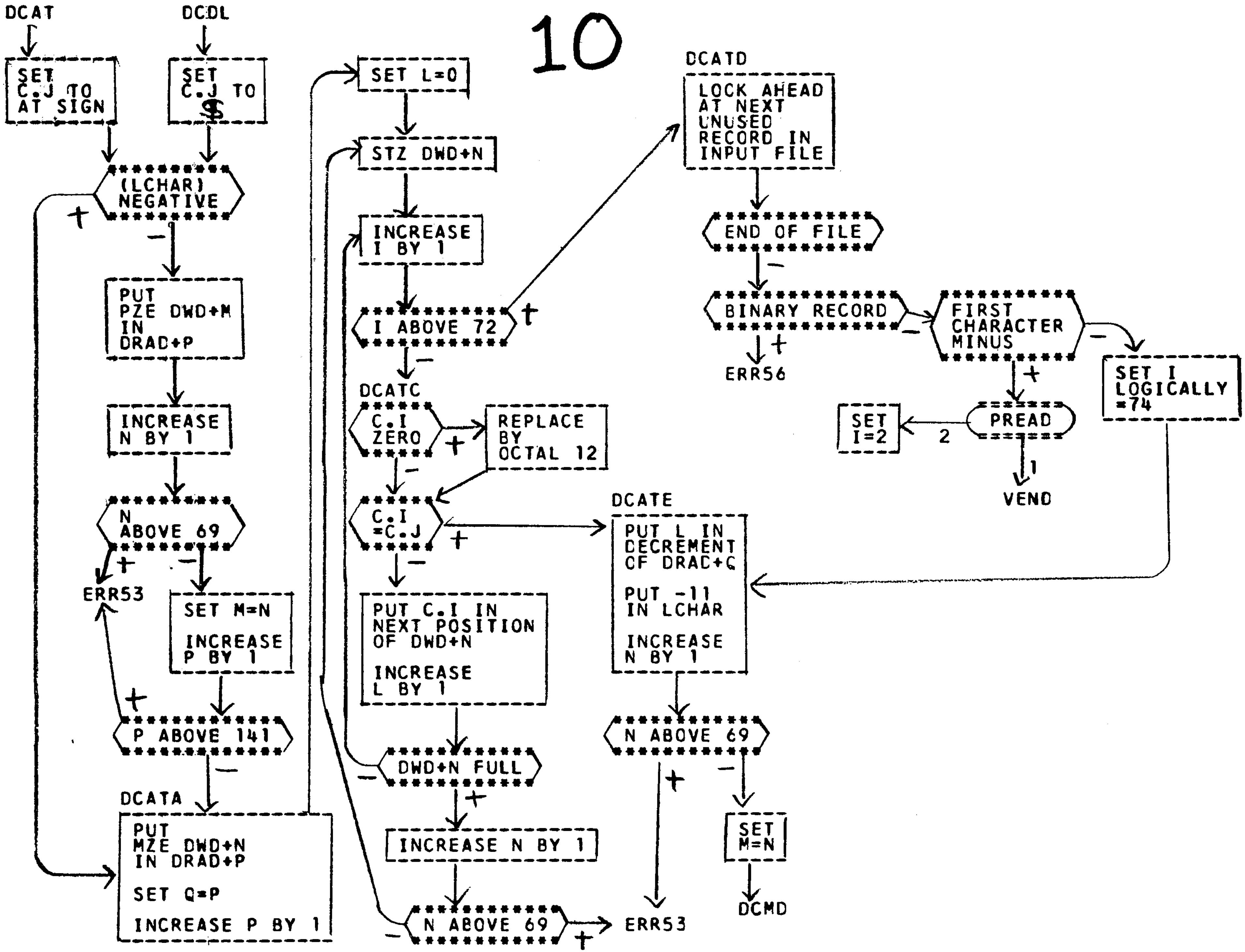

WRATE
CONSIDER THE CHARACTERS IN THE RECORD SCARD THROUGH SCARD+13 TO BE NUMBERED C1 TO C84
IS C8 AN ASTERISK
+
-
ARE C14 TO C16 ((DSA))
+
WRATL
PUT ((03)) IN C6,C7
-
ARE C14 TO C16 ((DS))
+
WRATK
IS C17 ASTERISK
-
+
IS C17 (
+
-
PUT C17-C20 IN AC.
WRATF
PUT C17-C2C IN AC.
D4V
WRATC
FPASS
-
ARE C14 TO C16 ((DC)) OR ((DCW))
+
WRATA
IS C17 ASTERISK
-
+
PUT C17-C20 IN C73-C76, ((D)) IN C80
WRATG
USE =D4V= TO CONVERT C6,C7 TO BINARY Y
Y=0
-
+
ERR50
SET Y=1
SET Z=(LCTR)+Y-1
PUT PZE Z IN AC.
BT4C
PUT THE 4 DIGITS IN C73-C76, ((C)) IN C78
PUT PZE Z IN AC.
FPASS
SET (LCTR) =Z+1
WRATN
(NAFL) ABOVE (LCTR)+1
-
+
EXIT
WRATY
WRITE RECORD IN SCARC TO SCARD+13 ON FILE WINTER. BLANK SCARD TO SCARC+13
WRATW
SET SWITCH (NOGO). WRITE MESSAGE ON FILE WINTER. SET ADDRESS AT WRATW+1 SO THAT THE MESSAGE WILL NCT BE REPEATED DURING THIS ASSEMBLY.
-
CAL LCTR
BT4D
PUT THE 4 DIGITS IN C73-C76, ZERO IN C78
SET X=1
IS C17 BLANK
-
INCREASE X BY 3
+
IS C28 BLANK
-
INCREASE X BY 3
+
IS C39 BLANK
-
+
X BELCW 7
+
-
C14-C16 ((B))OR ((B))
+
-
INCREASE X BY 1
PLT X IN C78
CAL LCTR
FPASS
INCREASE (LCTR) BY X


6
BT3D
ENTER WITH PZE X IN AC.
DIVIDE X BY 10
X=10Q+R
R=0
SET R=10
PUT R IN 3RD CHARACTER POSITICN OF COM
DIVICE Q BY 10
Q=10S+T
T=0
SET T=10
PUT T IN 2ND CHARACTER POSITION OF COM
DIVIDE S BY 10
S=10U+V
V=0
SET V=10
PUT V IN 1ST CHARACTER POSITION OF COM
ACCCRDING TO U SELECT FROM TABLE BT3DC CORRECT ZONE BITS FOT 1ST AND 3RD CHARACTERS OF COM
EXIT WITH 3-CHARACTER 1401ADDRESS EQUAL TO X AT LEFT END OF AC.
7
DEV
TSX DEV,4
PZE A
(A) ZERO OR NEGATIVE
ERR04
STZ COM
LDQ* A
MCVE NEXT CHARACTER FROM LEFT END OF (A) INTO AC.
SLW COM+1
MULTIPLY (COM) BY 10
ADD COM+1
SLW COM
=0
ABOVE 10
=10
CAL COM
EXIT 2
EXIT 1
ERROR,
NON-DIGIT
REPLACE BY 0
8
BT4D
ENTER WITH PZE X IN AC.
DIVIDE X BY 10
X=10M+N
N=0
SET N=10
PUT N IN 4TH CHARACTER POSITION OF COM
DIVIDE M BY 10
M=10Q+R
R=0
SET R=10
PUT R IN 3RD CHARACTER POSITION OF COM
DIVIDE Q BY 10
Q=10S+T
T=0
SET T=10
PUT T IN 2ND CHARACTER PCSITION OF COM
S=0
SET S=10
PLT S IN FIRST CHARACTER POSITION OF COM
EXIT
WITH 4-DIGIT DECIMAL EQUIVALENT OF X IN LEFT END OF AC.
+
-

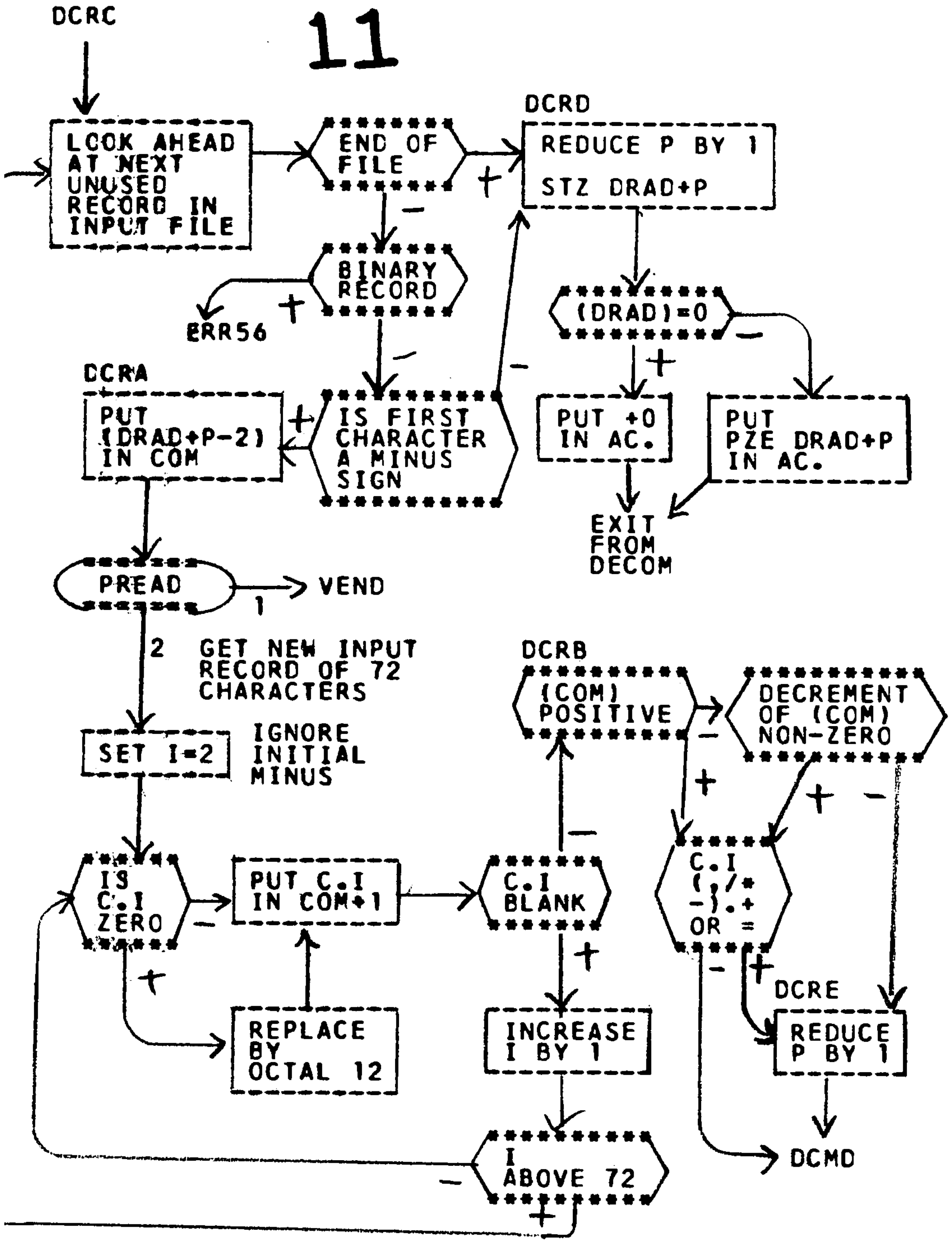
9


DECOM TSX DECOM,4
PZE A
NUMBER THE 72 CHARACTERS IN (A) THROUGH (A+11) C.1 TO C.72
IS C.1 ASTERISK
+
EXIT WITH 0 IN AC.
−
SET M=0(IN DWDM) N=0(IN DWCL) P=0(IN DRADL)
PUT -0 IN LCHAR
STZ DRAD THROUGH DRAD+143
IS C.1 BLANK
+
−
SET P=1
PUT -4 IN DRAD, SO THAT E.1 IS BLANK. NOTE THAT C.1 WILL ALSO PUT A BLANK IN E.2.
SET I=1
STZ DWD+N (N NOW=0)
DCMD
IS C.I ZERO
−
+
REPLACE BY OCTAL 12
IS C.I (,/* -).+ OR =
−
+
SET Q=1,2,3, 5,7,8,9,10, OR 12 RESPECTIVELY
IS C.I AT SIGN OR $
+
DCAT OR DCDL
−
IS C.I BLANK
−
PUT C.I IN LCHAR AND IN NEXT VACANT CHARACTER POSITION OF DWD+N
+
DCBL
(LCHAR) =-4
−
SET Q=4
(LCHAR) NEGATIVE
−
PUT PZE DWD+M IN DRAD+P
INCREASE N BY 1
+
DCBLA
PUT -Q IN DRAD+P AND LCHAR
INCREASE P BY 1
N ABOVE 69
+
−
ERR53
SET M=N
INCREASE P BY 1
P ABOVE 141
+
−
P ABCVE 141
−
+
ERR53
IS DWC+N NOW FULL
−
+
INCREASE N BY 1
INCREASE I BY 1
STZ DWD+N
N ABOVE 69
−
+
ERR53
I ABOVE 72
−
+
I=74
−
+
SET I=74
DCRC

10

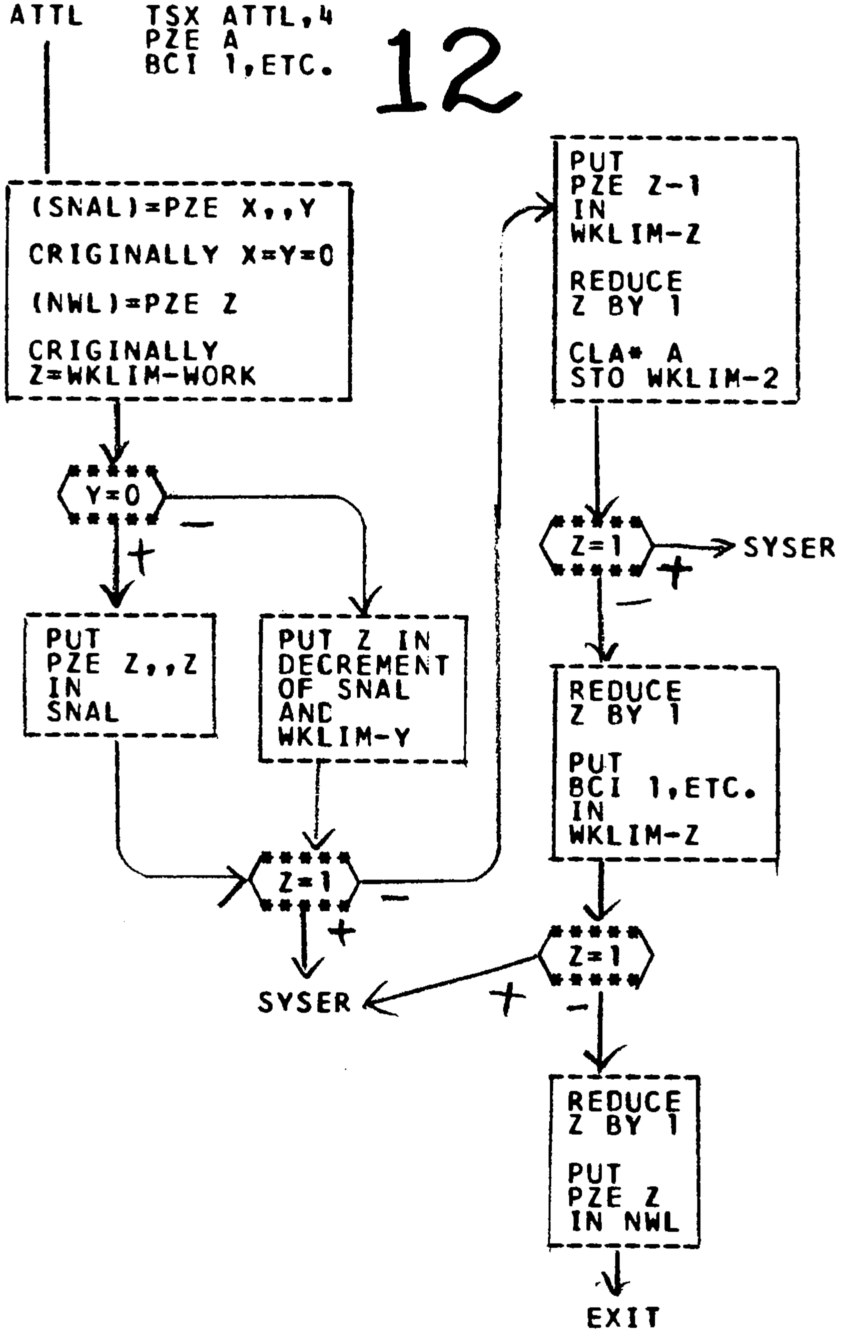

DCAT
SET C.J TO AT SIGN
DCDL
SET C.J TO $
(LCHAR) NEGATIVE
+
-
PUT PZE DWD+M IN DRAD+P
INCREASE N BY 1
N ABOVE 69
+
-
ERR53
SET M=N
INCREASE P BY 1
+
P ABOVE 141
-
DCATA
PUT MZE DWD+N IN DRAD+P
SET Q=P
INCREASE P BY 1
SET L=0
STZ DWD+N
INCREASE I BY 1
I ABOVE 72
+
-
DCATC
C.I ZERO
+
-
REPLACE BY OCTAL 12
C.I =C.J
+
-
PUT C.I IN NEXT POSITION OF DWD+N
INCREASE L BY 1
DWD+N FULL
-
+
INCREASE N BY 1
N ABOVE 69
-
+
ERR53
DCATD
LOCK AHEAD AT NEXT UNUSED RECORD IN INPUT FILE
END OF FILE
-
BINARY RECORD
+
ERR56
-
FIRST CHARACTER MINUS
-
+
SET I LOGICALLY =74
PREAD
2
SET I=2
1
VEND
DCATE
PUT L IN DECREMENT OF DRAD+Q
PUT -11 IN LCHAR
INCREASE N BY 1
N ABOVE 69
+
-
ERR53
SET M=N
DCMD

11



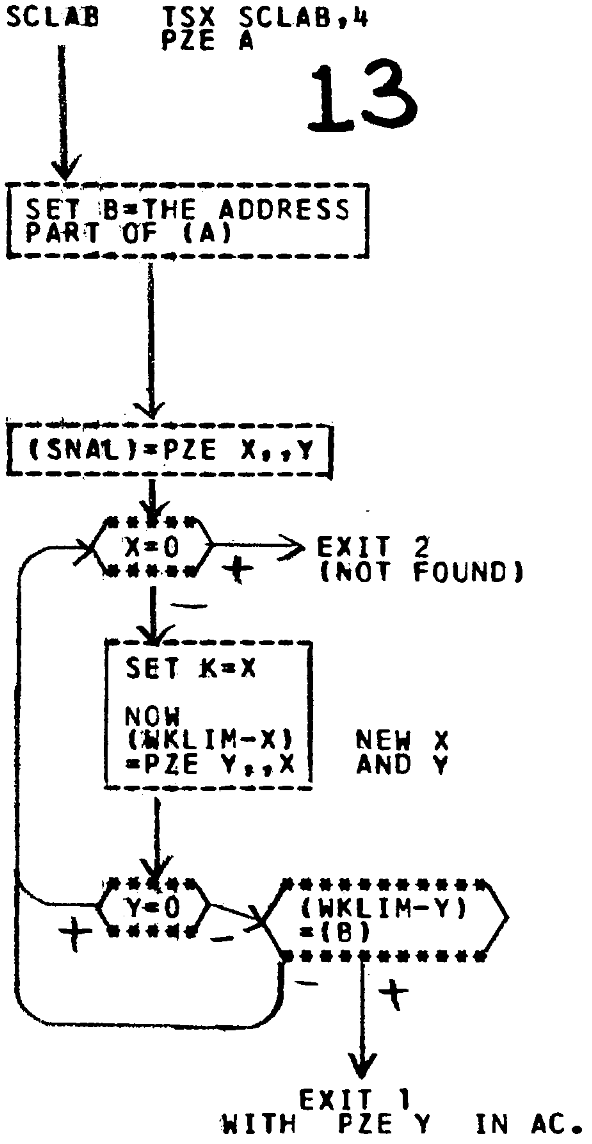
DCRC
LOOK AHEAD AT NEXT UNUSED RECORD IN INPUT FILE
END OF FILE
+
-
DCRD
REDUCE P BY 1
STZ DRAD+P
BINARY RECORD
+
ERR56
-
IS FIRST CHARACTER A MINUS SIGN
+
-
DCRA
PUT (DRAD+P-2) IN COM
(DRAD)=0
+
-
PUT +0 IN AC.
PUT PZE DRAD+P IN AC.
EXIT FROM DECOM
PREAD
1
VEND
2
GET NEW INPUT RECORD OF 72 CHARACTERS
SET I=2
IGNORE INITIAL MINUS
DCRB
(COM) POSITIVE
-
+
DECREMENT OF (COM) NON-ZERO
+
-
IS C.I ZERO
-
+
PUT C.I IN COM+1
REPLACE BY OCTAL 12
C.I BLANK
-
+
INCREASE I BY 1
I ABOVE 72
-
+
C.I (,/* -).+ OR =
-
+
DCRE
REDUCE P BY 1
DCMD

12

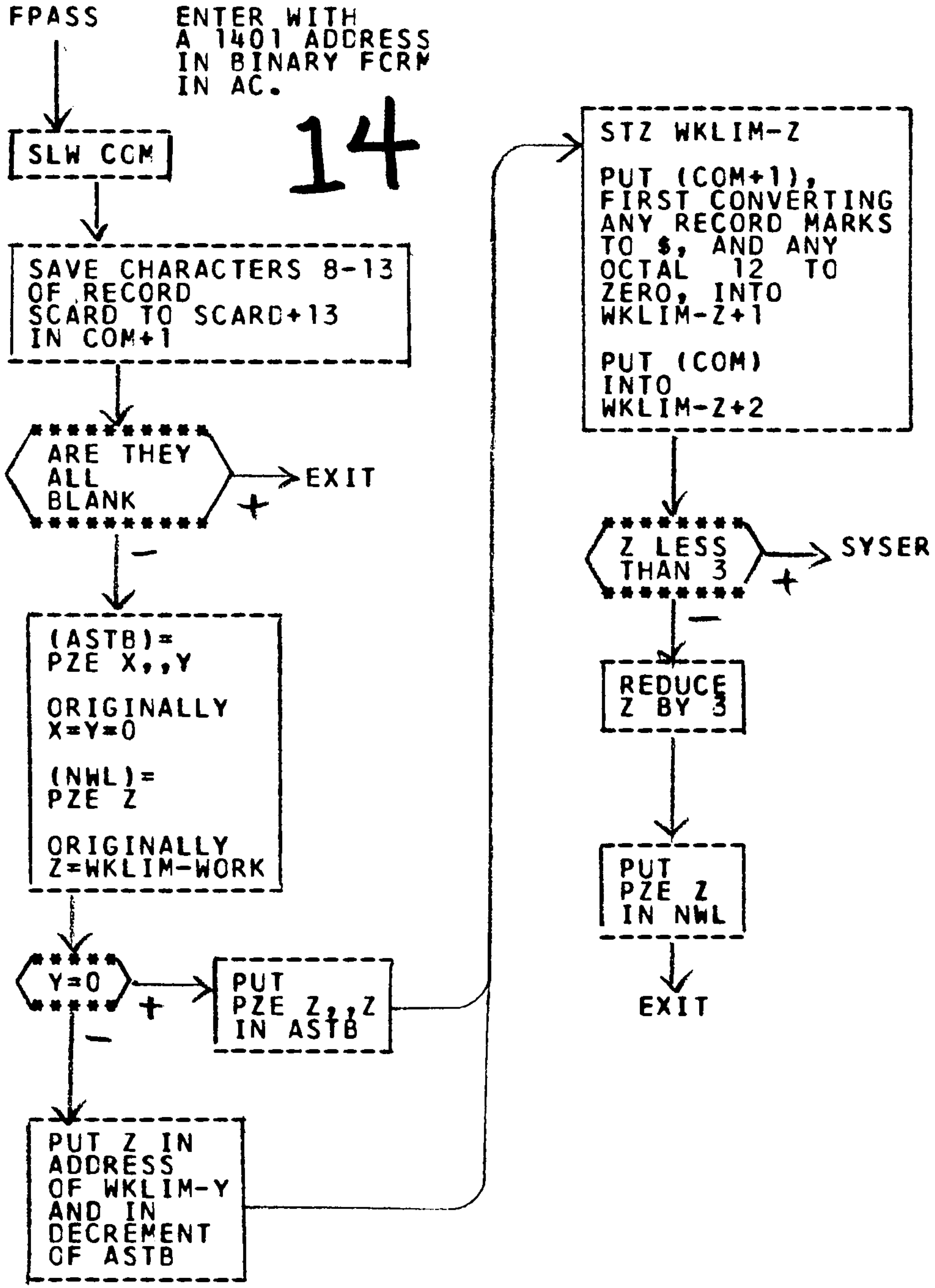


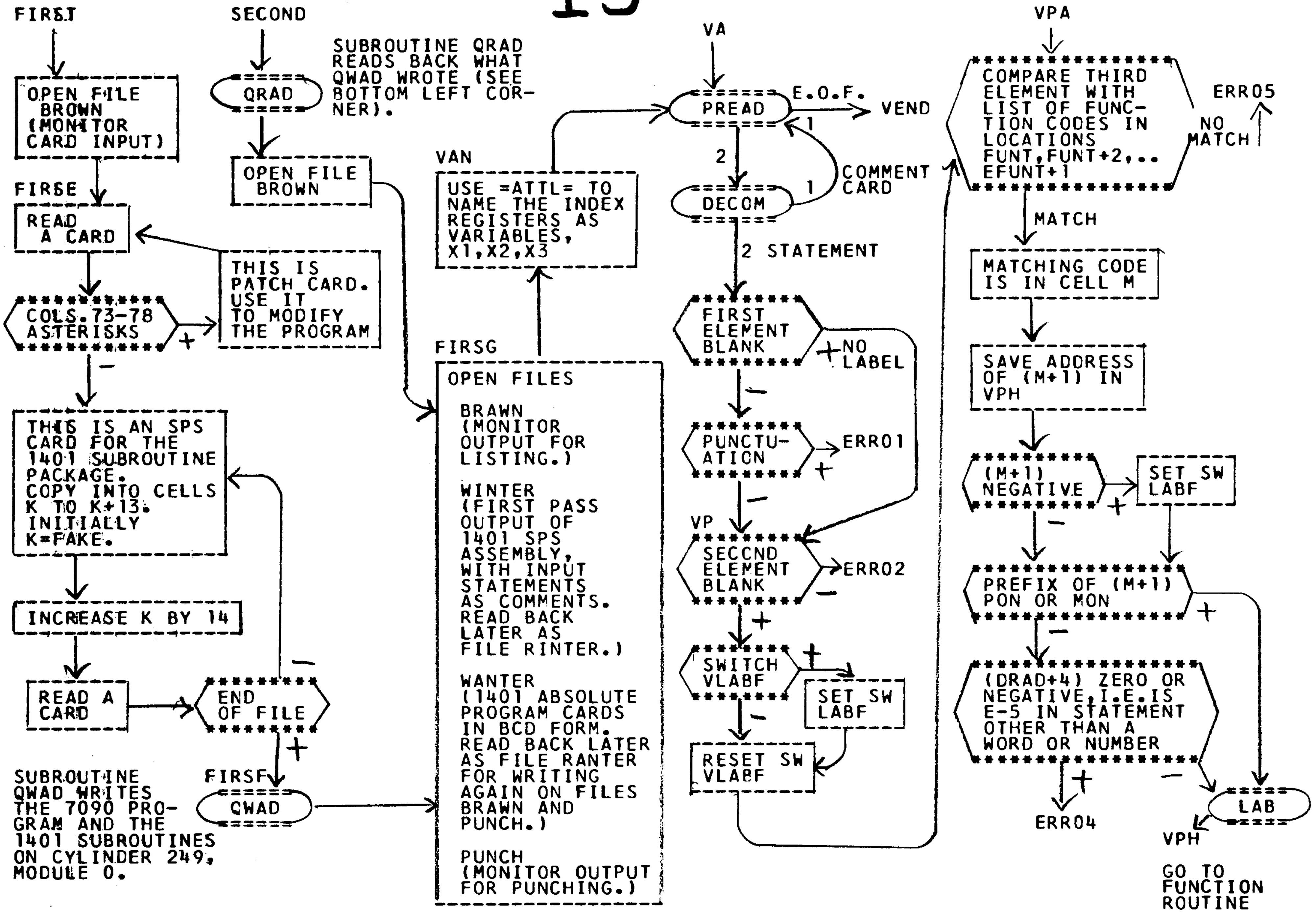
ATTL
TSX ATTL,4
PZE A
BCI 1,ETC.
(SNAL)=PZE X,,Y
CRIGINALLY X=Y=0
(NWL)=PZE Z
CRIGINALLY Z=WKLIM-WORK
Y=0
+
-
PUT PZE Z,,Z IN SNAL
PUT Z IN DECREMENT OF SNAL AND WKLIM-Y
Z=1
+
-
SYSER
PUT PZE Z-1 IN WKLIM-Z
REDUCE Z BY 1
CLA* A
STO WKLIM-2
Z=1
+
SYSER
-
REDUCE Z BY 1
PUT BCI 1,ETC. IN WKLIM-Z
Z=1
+
SYSER
-
REDUCE Z BY 1
PUT PZE Z IN NWL
EXIT

13


SCLAB
TSX SCLAB,4
PZE A
SET B=THE ADDRESS PART OF (A)
(SNAL)=PZE X,,Y
X=0
+
EXIT 2 (NOT FOUND)
−
SET K=X
NOW (WKLIM-X) =PZE Y,,X
NEW X AND Y
Y=0
+
−
(WKLIM-Y) =(B)
−
+
EXIT 1 WITH PZE Y IN AC.

14


FPASS
ENTER WITH A 1401 ADDRESS IN BINARY FCRM IN AC.
SLW CCM
SAVE CHARACTERS 8-13 OF RECORD SCARD TO SCARD+13 IN COM+1
ARE THEY ALL BLANK
+
EXIT
−
(ASTB)= PZE X,,Y
ORIGINALLY X=Y=0
(NWL)= PZE Z
ORIGINALLY Z=WKLIM-WORK
Y=0
+
PUT PZE Z,,Z IN ASTB
−
PUT Z IN ADDRESS OF WKLIM-Y AND IN DECREMENT OF ASTB
STZ WKLIM-Z
PUT (COM+1), FIRST CONVERTING ANY RECORD MARKS TO $, AND ANY OCTAL 12 TO ZERO, INTO WKLIM-Z+1
PUT (COM) INTO WKLIM-Z+2
Z LESS THAN 3
+
SYSER
−
REDUCE Z BY 3
PUT PZE Z IN NWL
EXIT


FIRST
OPEN FILE BROWN (MONITOR CARD INPUT)
FIRSE
READ A CARD
COLS.73-78 ASTERISKS
+
THIS IS PATCH CARD. USE IT TO MODIFY THE PROGRAM
−
THIS IS AN SPS CARD FOR THE 1401 SUBROUTINE PACKAGE. COPY INTO CELLS K TO K+13. INITIALLY K=FAKE.
INCREASE K BY 14
READ A CARD
END OF FILE
−
+
FIRSF
QWAD
SECOND
QRAD
OPEN FILE BROWN
VAN
USE =ATTL= TO NAME THE INDEX REGISTERS AS VARIABLES, X1,X2,X3
FIRSG
OPEN FILES
BRAWN (MONITOR OUTPUT FOR LISTING.)
WINTER (FIRST PASS OUTPUT OF 1401 SPS ASSEMBLY, WITH INPUT STATEMENTS AS COMMENTS. READ BACK LATER AS FILE RINTER.)
WANTER (1401 ABSOLUTE PROGRAM CARDS IN BCD FORM. READ BACK LATER AS FILE RANTER FOR WRITING AGAIN ON FILES BRAWN AND PUNCH.)
PUNCH (MONITOR OUTPUT FOR PUNCHING.)
VA
PREAD
E.O.F.
VEND
1
2
DECOM
1
COMMENT CARD
2 STATEMENT
FIRST ELEMENT BLANK
+ NO LABEL
−
PUNCTU-ATION
+
ERR01
−
VP
SECOND ELEMENT BLANK
−
ERR02
+
SWITCH VLABF
+
SET SW LABF
−
RESET SW VLABF
VPA
COMPARE THIRD ELEMENT WITH LIST OF FUNCTION CODES IN LOCATIONS FUNT,FUNT+2,.. EFUNT+1
NO MATCH
ERR05
MATCH
MATCHING CODE IS IN CELL M
SAVE ADDRESS OF (M+1) IN VPH
(M+1) NEGATIVE
+
SET SW LABF
−
PREFIX OF (M+1) PON OR MON
+
−
(DRAD+4) ZERO OR NEGATIVE, I.E. IS E-5 IN STATEMENT OTHER THAN A WORD OR NUMBER
+
ERR04
−
LAB
VPH
GO TO FUNCTION ROUTINE

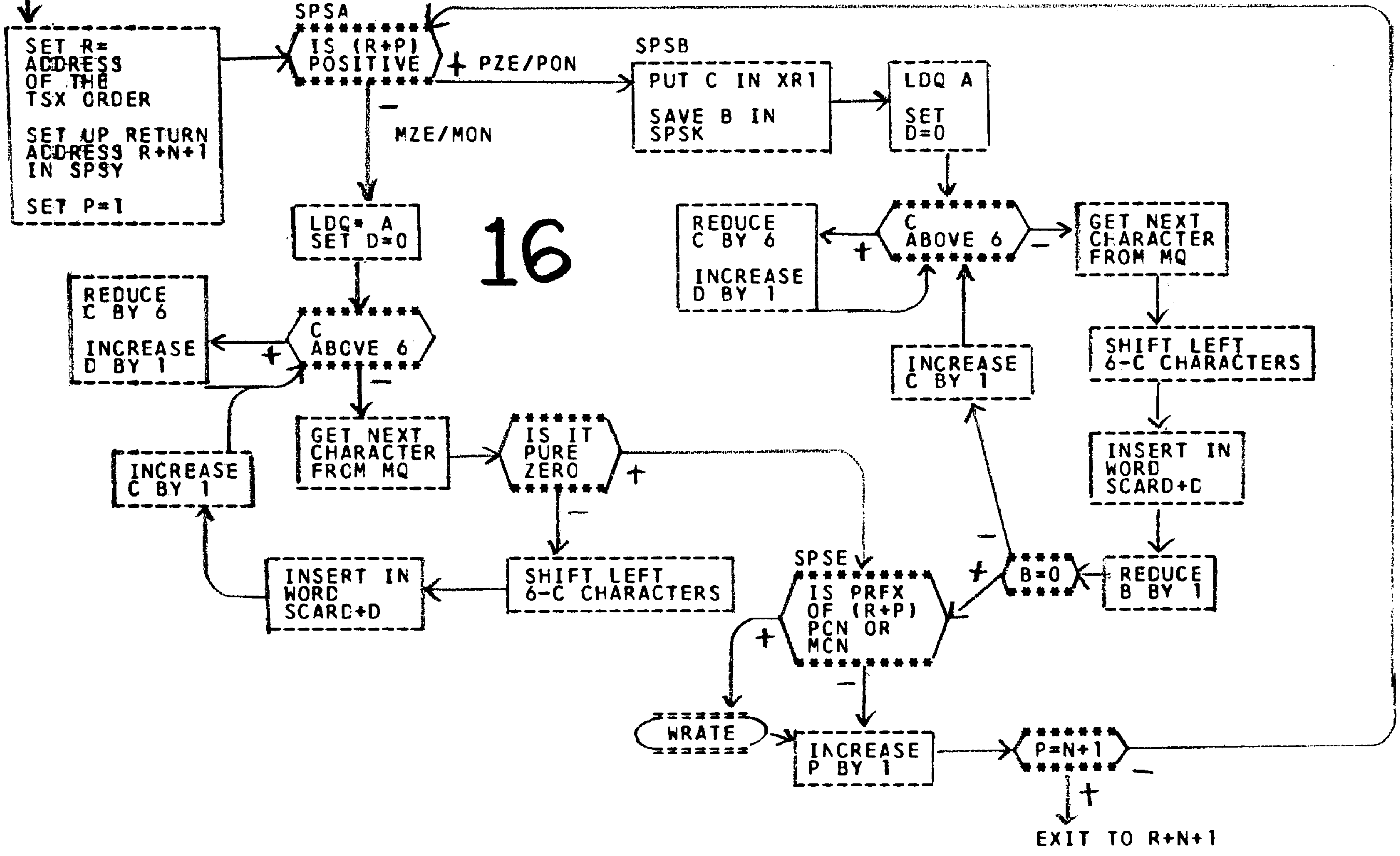
SUBROUTINE QRAD READS BACK WHAT QWAD WROTE (SEE BOTTOM LEFT CORNER).

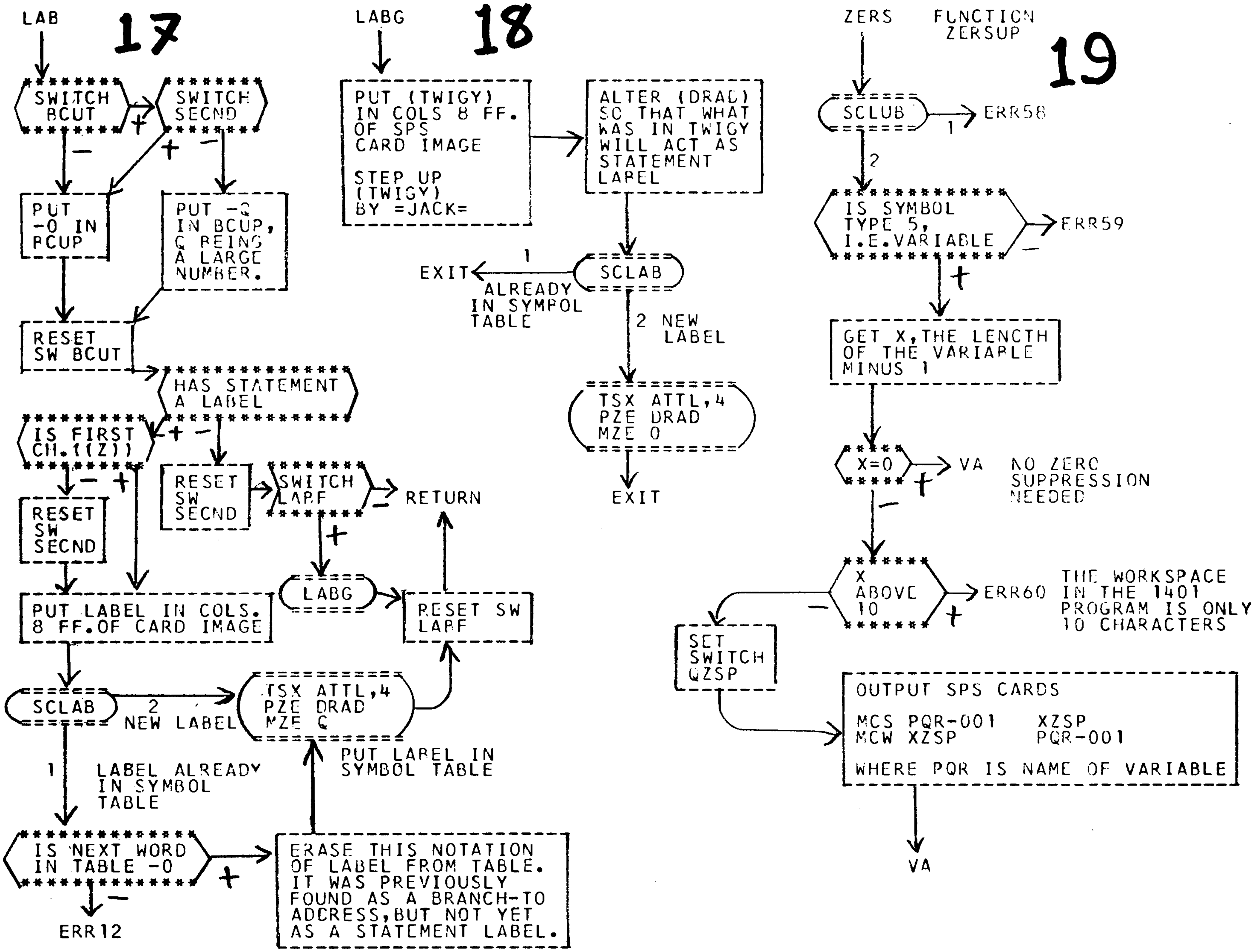
SUBROUTINE QWAD WRITES THE 7090 PROGRAM AND THE 1401 SUBROUTINES ON CYLINDER 249, MODULE 0.

SPS TSX SPS,4,N
FOLLOWED BY N WORDS OF THE FORM PZE/PON A,B,C OR MZE/MON A,,C. B MUST BE FROM 1 TO 6.

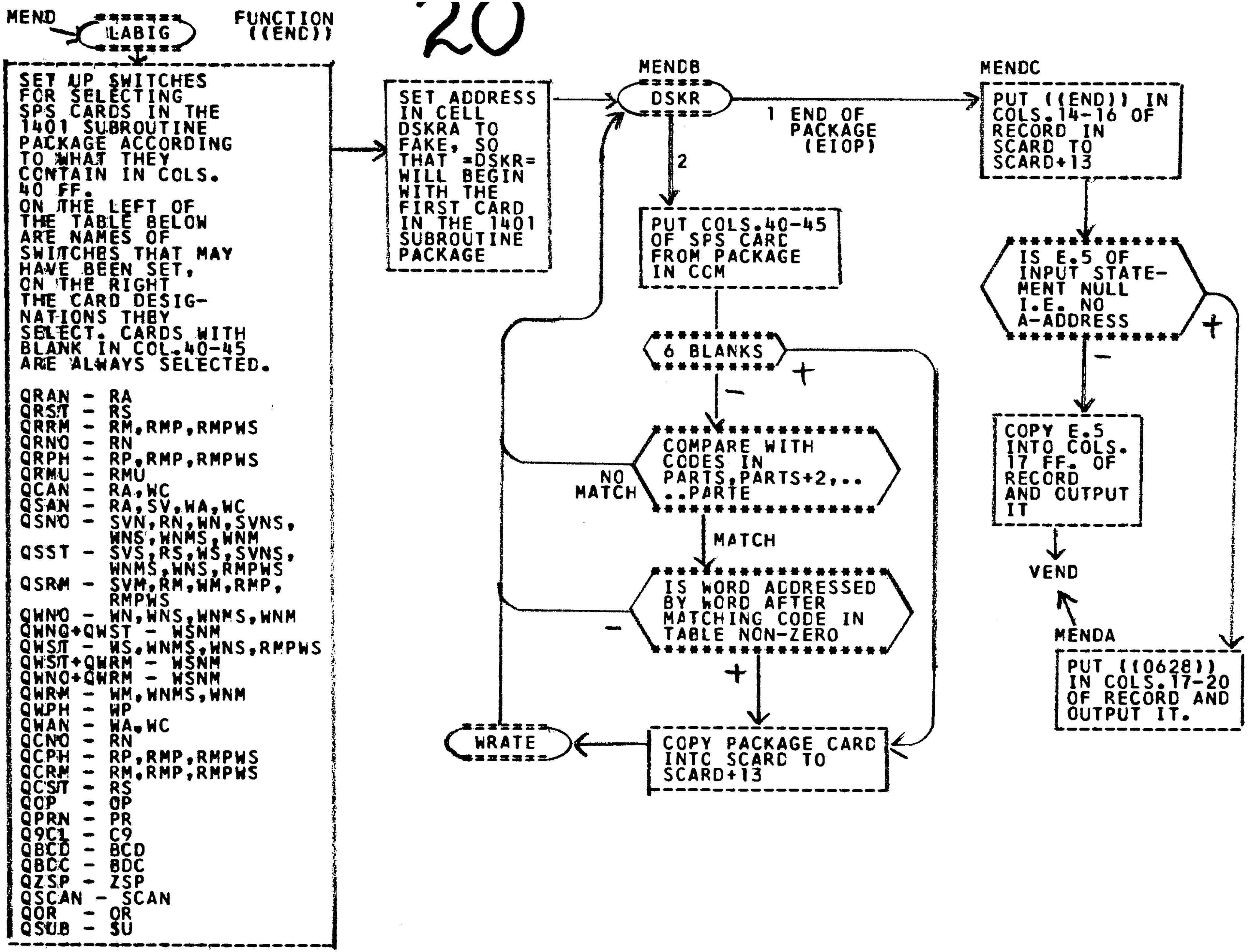
IF PZE/PON, LDQ A FETCHES A WORD WHOSE FIRST B CHARACTERS ARE TO BE PLACED IN COLUMNS C TO C+B-1 OF THE SPS CARD IMAGE IN SCARD THROUGH SCARD+13.

IF MZE/MON LDQ* A FETCHES THE WORD. THE EFFECTIVE B IS 6, OR IF LESS THAN 6 IS DETERMINED BY FINDING A PURE ZERO IN POSITION B+1.




17
LAB
SWITCH BCUT
SWITCH SECND
+
-
+
-
PUT -0 IN BCUP
PUT -Q IN BCUP, Q BEING A LARGE NUMBER.
RESET SW BCUT
HAS STATEMENT A LABEL
IS FIRST CH.1(Z))
+
-
-
+
RESET SW SECND
SWITCH LABF
-
RETURN
+
RESET SW SECND
LABG
RESET SW LABF
PUT LABEL IN COLS. 8 FF.OF CARD IMAGE
SCLAB
2 NEW LABEL
TSX ATTL,4
PZE DRAD
MZE Q
PUT LABEL IN SYMBOL TABLE
1 LABEL ALREADY IN SYMBOL TABLE
IS NEXT WORD IN TABLE -0
+
-
ERR12
ERASE THIS NOTATION OF LABEL FROM TABLE. IT WAS PREVIOUSLY FOUND AS A BRANCH-TO ADDRESS,BUT NOT YET AS A STATEMENT LABEL.
18
LABG
PUT (TWIGY) IN COLS 8 FF. OF SPS CARD IMAGE
STEP UP (TWIGY) BY =JACK=
ALTER (DRAD) SO THAT WHAT WAS IN TWIGY WILL ACT AS STATEMENT LABEL
SCLAB
1
EXIT
ALREADY IN SYMBOL TABLE
2 NEW LABEL
TSX ATTL,4
PZE DRAD
MZE 0
EXIT
ZERS
FUNCTION ZERSUP
19
SCLUB
1
ERR58
2
IS SYMBOL TYPE 5, I.E.VARIABLE
ERR59
-
+
GET X,THE LENGTH OF THE VARIABLE MINUS 1
X=0
VA
+
NO ZERO SUPPRESSION NEEDED
-
X ABOVE 10
ERR60
+
-
THE WORKSPACE IN THE 1401 PROGRAM IS ONLY 10 CHARACTERS
SET SWITCH QZSP
OUTPUT SPS CARDS
MCS PQR-001 XZSP
MCW XZSP PQR-001
WHERE PQR IS NAME OF VARIABLE
VA

20



SET UP SWITCHES FOR SELECTING SPS CARDS IN THE 1401 SUBROUTINE PACKAGE ACCORDING TO WHAT THEY CONTAIN IN COLS. 40 FF.
ON THE LEFT OF THE TABLE BELOW ARE NAMES OF SWITCHES THAT MAY HAVE BEEN SET, ON THE RIGHT THE CARD DESIGNATIONS THEY SELECT. CARDS WITH BLANK IN COL.40-45 ARE ALWAYS SELECTED.

QRAN	RA
QRST	RS
QRRM	RM,RMP,RMPWS
QRNO	RN
QRPH	RP,RMP,RMPWS
QRMU	RMU
QCAN	RA,WC
QSAN	RA,SV,WA,WC
QSNO	SVN,RN,WN,SVNS, WNS,WNMS,WNM
QSST	SVS,RS,WS,SVNS, WNMS,WNS,RMPWS
QSRM	SVM,RM,WM,RMP, RMPWS
QWNO	WN,WNS,WNMS,WNM
QWNO+QWST	WSNM
QWST	WS,WNMS,WNS,RMPWS
QWST+QWRM	WSNM
QWNO+QWRM	WSNM
QWRM	WM,WNMS,WNM
QWPH	WP
QWAN	WA,WC
QCNO	RN
QCPH	RP,RMP,RMPWS
QCRM	RM,RMP,RMPWS
QCST	RS
QOP	OP
QPRN	PR
Q9CL	C9
QBCD	BCD
QBDC	BDC
QZSP	ZSP
QSCAN	SCAN
QOR	OR
QSUB	SU

SET ADDRESS IN CELL DSKRA TO FAKE, SO THAT =DSKR= WILL BEGIN WITH THE FIRST CARD IN THE 1401 SUBROUTINE PACKAGE

PUT COLS.40-45 OF SPS CARD FROM PACKAGE IN CCM

COMPARE WITH CODES IN PARTS,PARTS+2,.. ..PARTE

IS WORD ADDRESSED BY WORD AFTER MATCHING CODE IN TABLE NON-ZERO

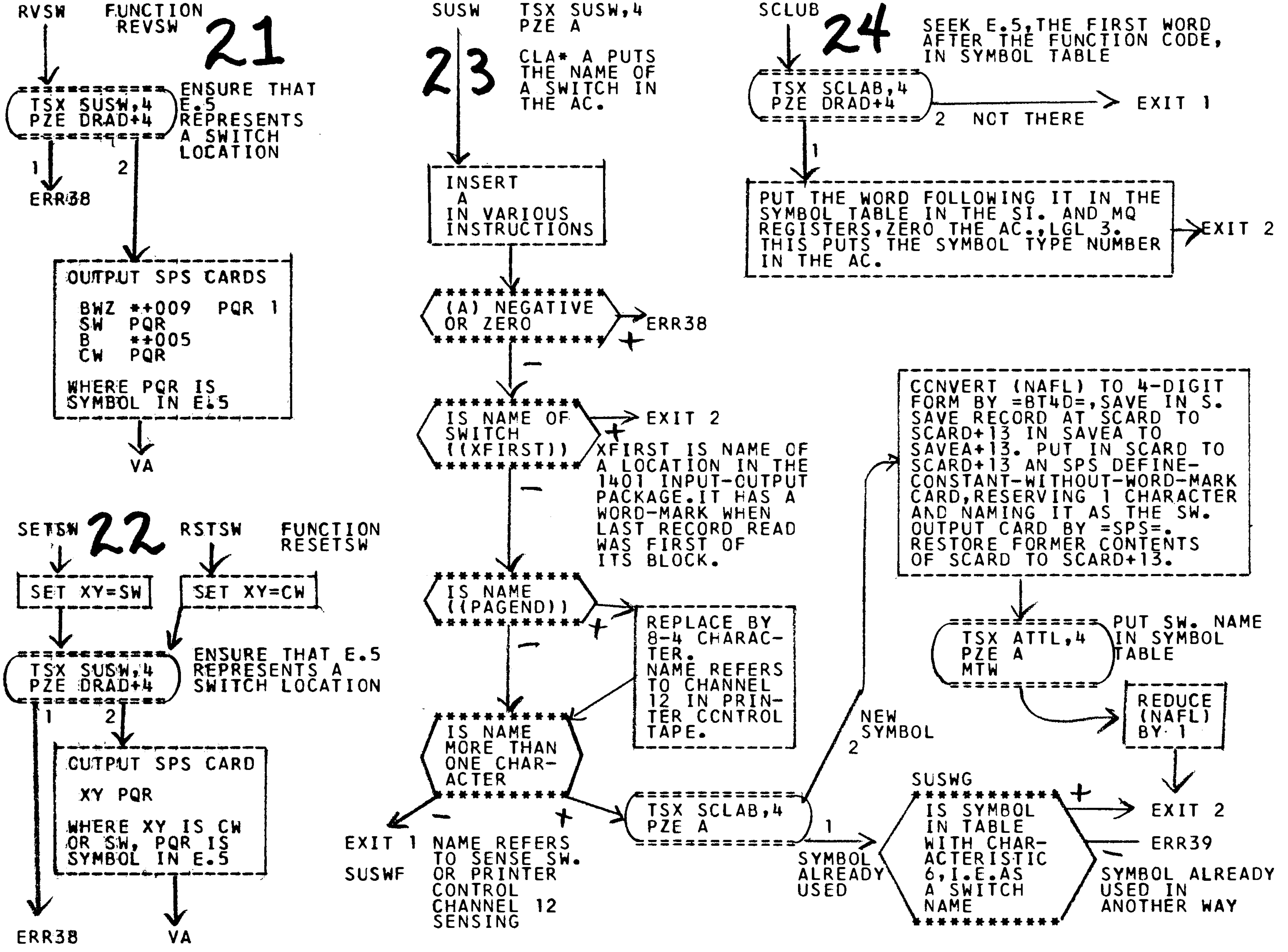
COPY PACKAGE CARD INTO SCARD TO SCARD+13

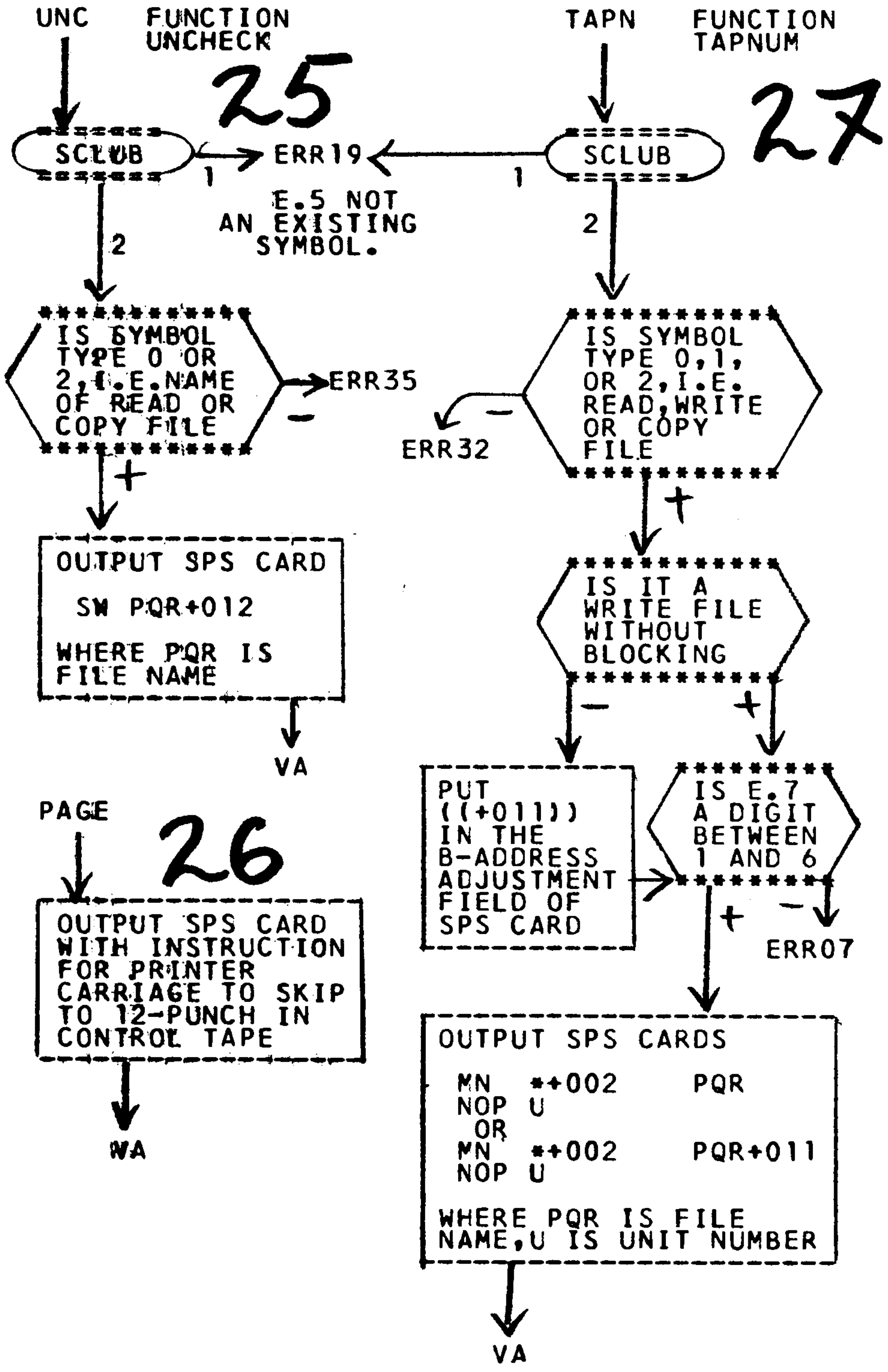
PUT ((END)) IN COLS.14-16 OF RECORD IN SCARD TO SCARD+13

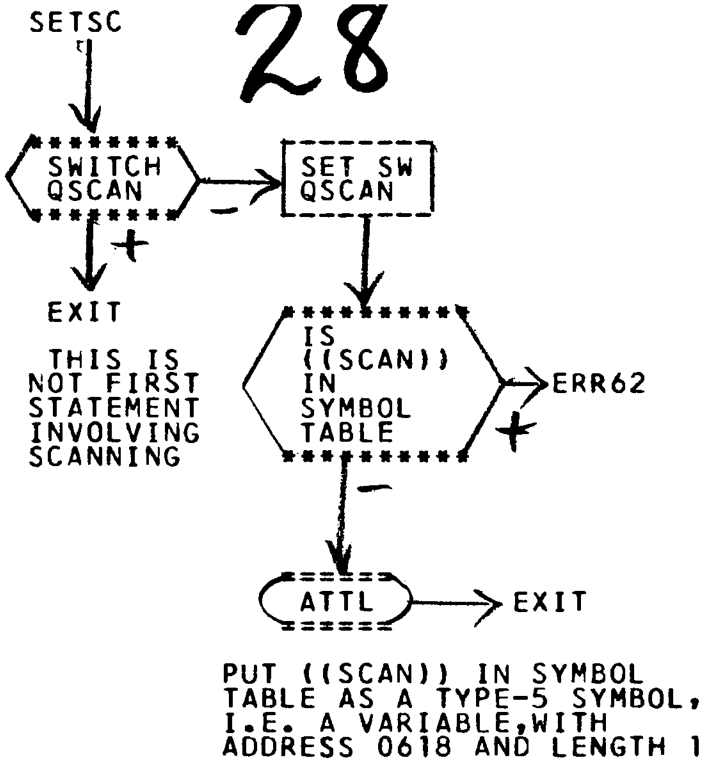
IS E.5 OF INPUT STATEMENT NULL I.E. NO A-ADDRESS

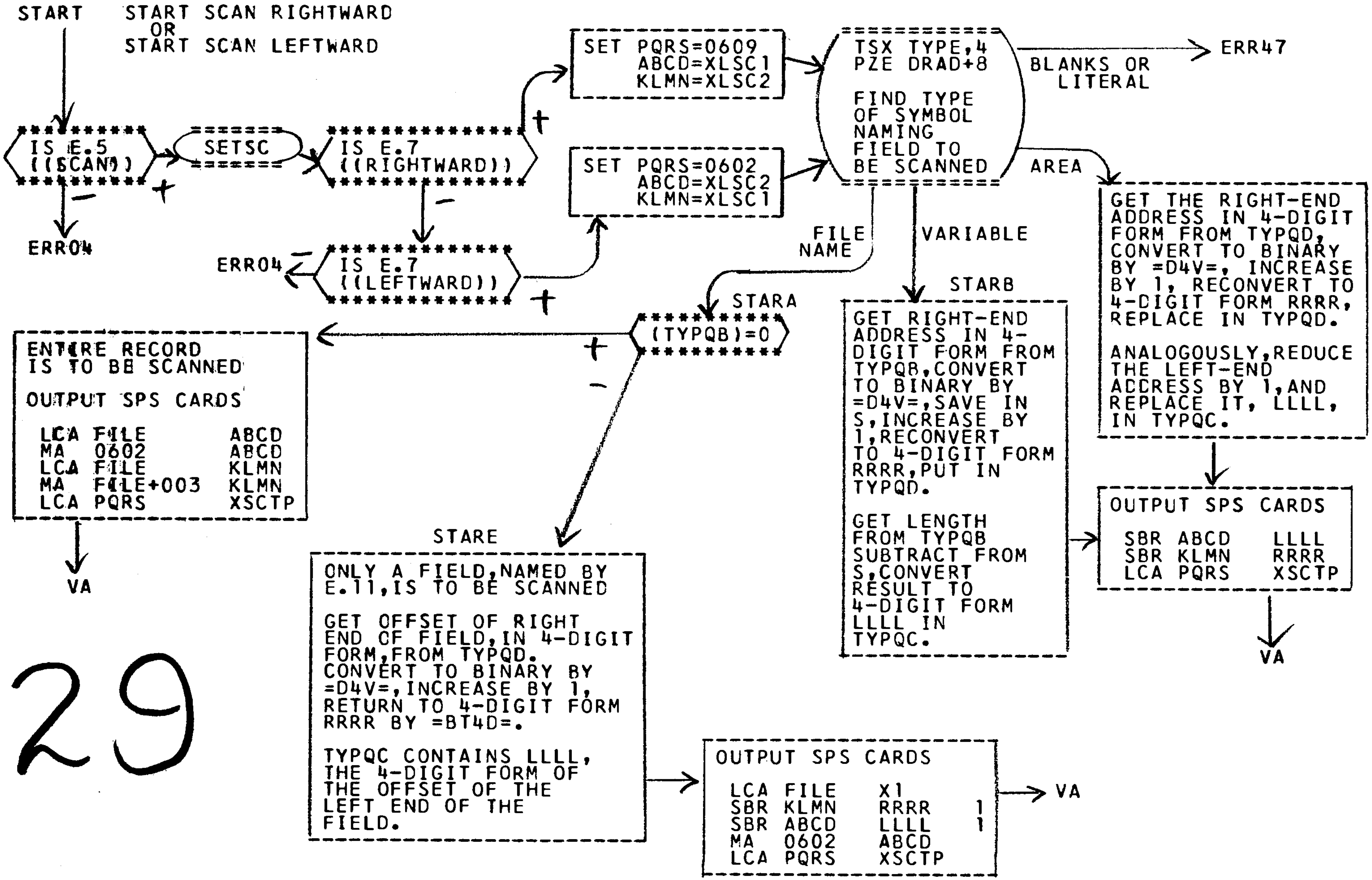
COPY E.5 INTO COLS. 17 FF. OF RECORD AND OUTPUT IT

PUT ((0628)) IN COLS.17-20 OF RECORD AND OUTPUT IT.


21
RVSW
FUNCTION REVSW
TSX SUSW,4
PZE DRAD+4
ENSURE THAT E.5 REPRESENTS A SWITCH LOCATION
1
2
ERR38
OUTPUT SPS CARDS
BWZ *+009 PQR 1
SW PQR
B *+005
CW PQR
WHERE PQR IS SYMBOL IN E.5
VA
22
SETSW
RSTSW
FUNCTION RESETSW
SET XY=SW
SET XY=CW
TSX SUSW,4
PZE DRAD+4
ENSURE THAT E.5 REPRESENTS A SWITCH LOCATION
1
2
OUTPUT SPS CARD
XY PQR
WHERE XY IS CW OR SW, PQR IS SYMBOL IN E.5
ERR38
VA
23
SUSW
TSX SUSW,4
PZE A
CLA* A PUTS THE NAME OF A SWITCH IN THE AC.
INSERT A IN VARIOUS INSTRUCTIONS
(A) NEGATIVE OR ZERO
ERR38
+
−
IS NAME OF SWITCH ((XFIRST))
EXIT 2
+
−
XFIRST IS NAME OF A LOCATION IN THE 1401 INPUT-OUTPUT PACKAGE.IT HAS A WORD-MARK WHEN LAST RECORD READ WAS FIRST OF ITS BLOCK.
IS NAME ((PAGEND))
+
−
REPLACE BY 8-4 CHARACTER. NAME REFERS TO CHANNEL 12 IN PRINTER CCNTROL TAPE.
IS NAME MORE THAN ONE CHARACTER
−
+
EXIT 1
SUSWF
NAME REFERS TO SENSE SW. OR PRINTER CONTROL CHANNEL 12 SENSING
TSX SCLAB,4
PZE A
1
SYMBOL ALREADY USED
2
NEW SYMBOL
CCNVERT (NAFL) TO 4-DIGIT FORM BY =BT4D=,SAVE IN S. SAVE RECORD AT SCARD TO SCARD+13 IN SAVEA TO SAVEA+13. PUT IN SCARD TO SCARD+13 AN SPS DEFINE-CONSTANT-WITHOUT-WORD-MARK CARD,RESERVING 1 CHARACTER AND NAMING IT AS THE SW. OUTPUT CARD BY =SPS=. RESTORE FORMER CONTENTS OF SCARD TO SCARD+13.
TSX ATTL,4
PZE A
MTW
PUT SW. NAME IN SYMBOL TABLE
REDUCE (NAFL) BY 1
EXIT 2
SUSWG
IS SYMBOL IN TABLE WITH CHARACTERISTIC 6,I.E.AS A SWITCH NAME
+
EXIT 2
−
ERR39
SYMBOL ALREADY USED IN ANOTHER WAY
24
SCLUB
SEEK E.5,THE FIRST WORD AFTER THE FUNCTION CODE, IN SYMBOL TABLE
TSX SCLAB,4
PZE DRAD+4
2 NOT THERE
EXIT 1
1
PUT THE WORD FOLLOWING IT IN THE SYMBOL TABLE IN THE SI. AND MQ REGISTERS,ZERO THE AC.,LGL 3. THIS PUTS THE SYMBOL TYPE NUMBER IN THE AC.
EXIT 2

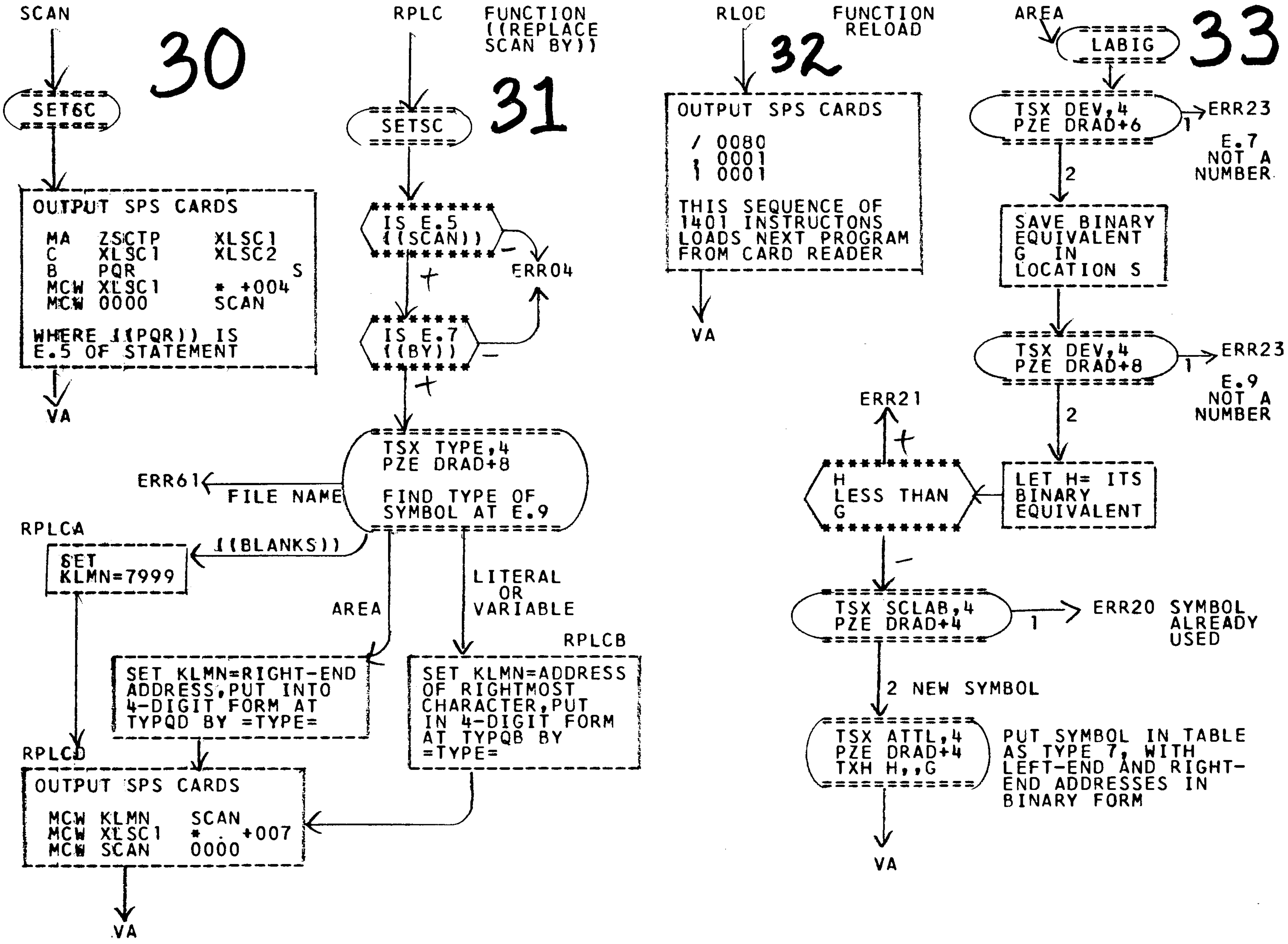

UNC
FUNCTION UNCHECK
25
SCLUB
1
ERR19
E.5 NOT AN EXISTING SYMBOL.
2
IS SYMBOL TYPE 0 OR 2,I.E.NAME OF READ OR COPY FILE
ERR35
-
+
OUTPUT SPS CARD
SW PQR+012
WHERE PQR IS FILE NAME
VA
PAGE
26
OUTPUT SPS CARD WITH INSTRUCTION FOR PRINTER CARRIAGE TO SKIP TO 12-PUNCH IN CONTROL TAPE
WA
TAPN
FUNCTION TAPNUM
27
SCLUB
1
2
IS SYMBOL TYPE 0,1, OR 2,I.E. READ,WRITE OR COPY FILE
-
ERR32
+
IS IT A WRITE FILE WITHOUT BLOCKING
-
+
PUT ((+011)) IN THE B-ADDRESS ADJUSTMENT FIELD OF SPS CARD
IS E.7 A DIGIT BETWEEN 1 AND 6
+
-
ERR07
OUTPUT SPS CARDS
MN *+002 PQR
NOP U
OR
MN *+002 PQR+011
NOP U
WHERE PQR IS FILE NAME,U IS UNIT NUMBER
VA


SETSC
28
SWITCH QSCAN
-
SET SW QSCAN
+
EXIT
THIS IS NOT FIRST STATEMENT INVOLVING SCANNING
IS ((SCAN)) IN SYMBOL TABLE
ERR62
+
-
ATTL
EXIT
PUT ((SCAN)) IN SYMBOL TABLE AS A TYPE-5 SYMBOL, I.E. A VARIABLE,WITH ADDRESS 0618 AND LENGTH 1


START
START SCAN RIGHTWARD
OR
START SCAN LEFTWARD
IS E.5 ((SCAN))
ERR04
SETSC
IS E.7 ((RIGHTWARD))
IS E.7 ((LEFTWARD))
ERR04
SET PQRS=0609
ABCD=XLSC1
KLMN=XLSC2
SET PQRS=0602
ABCD=XLSC2
KLMN=XLSC1
TSX TYPE,4
PZE DRAD+8
FIND TYPE OF SYMBOL NAMING FIELD TO BE SCANNED
BLANKS OR LITERAL
ERR47
AREA
FILE NAME
VARIABLE
STARA
(TYPQB)=0
STARB
GET RIGHT-END ADDRESS IN 4-DIGIT FORM FROM TYPQB,CONVERT TO BINARY BY =D4V=,SAVE IN S,INCREASE BY 1,RECONVERT TO 4-DIGIT FORM RRRR,PUT IN TYPQD.
GET LENGTH FROM TYPQB SUBTRACT FROM S,CONVERT RESULT TO 4-DIGIT FORM LLLL IN TYPQC.
GET THE RIGHT-END ADDRESS IN 4-DIGIT FORM FROM TYPQD, CONVERT TO BINARY BY =D4V=, INCREASE BY 1, RECONVERT TO 4-DIGIT FORM RRRR, REPLACE IN TYPQD.
ANALOGOUSLY,REDUCE THE LEFT-END ADDRESS BY 1,AND REPLACE IT, LLLL, IN TYPQC.
OUTPUT SPS CARDS
SBR ABCD LLLL
SBR KLMN RRRR
LCA PQRS XSCTP
VA
ENTIRE RECORD IS TO BE SCANNED
OUTPUT SPS CARDS
LCA FILE ABCD
MA 0602 ABCD
LCA FILE KLMN
MA FILE+003 KLMN
LCA PQRS XSCTP
VA
STARE
ONLY A FIELD,NAMED BY E.11,IS TO BE SCANNED
GET OFFSET OF RIGHT END OF FIELD,IN 4-DIGIT FORM,FROM TYPQD. CONVERT TO BINARY BY =D4V=,INCREASE BY 1, RETURN TO 4-DIGIT FORM RRRR BY =BT4D=.
TYPQC CONTAINS LLLL, THE 4-DIGIT FORM OF THE OFFSET OF THE LEFT END OF THE FIELD.
OUTPUT SPS CARDS
LCA FILE X1
SBR KLMN RRRR 1
SBR ABCD LLLL 1
MA 0602 ABCD
LCA PQRS XSCTP
VA
+
-

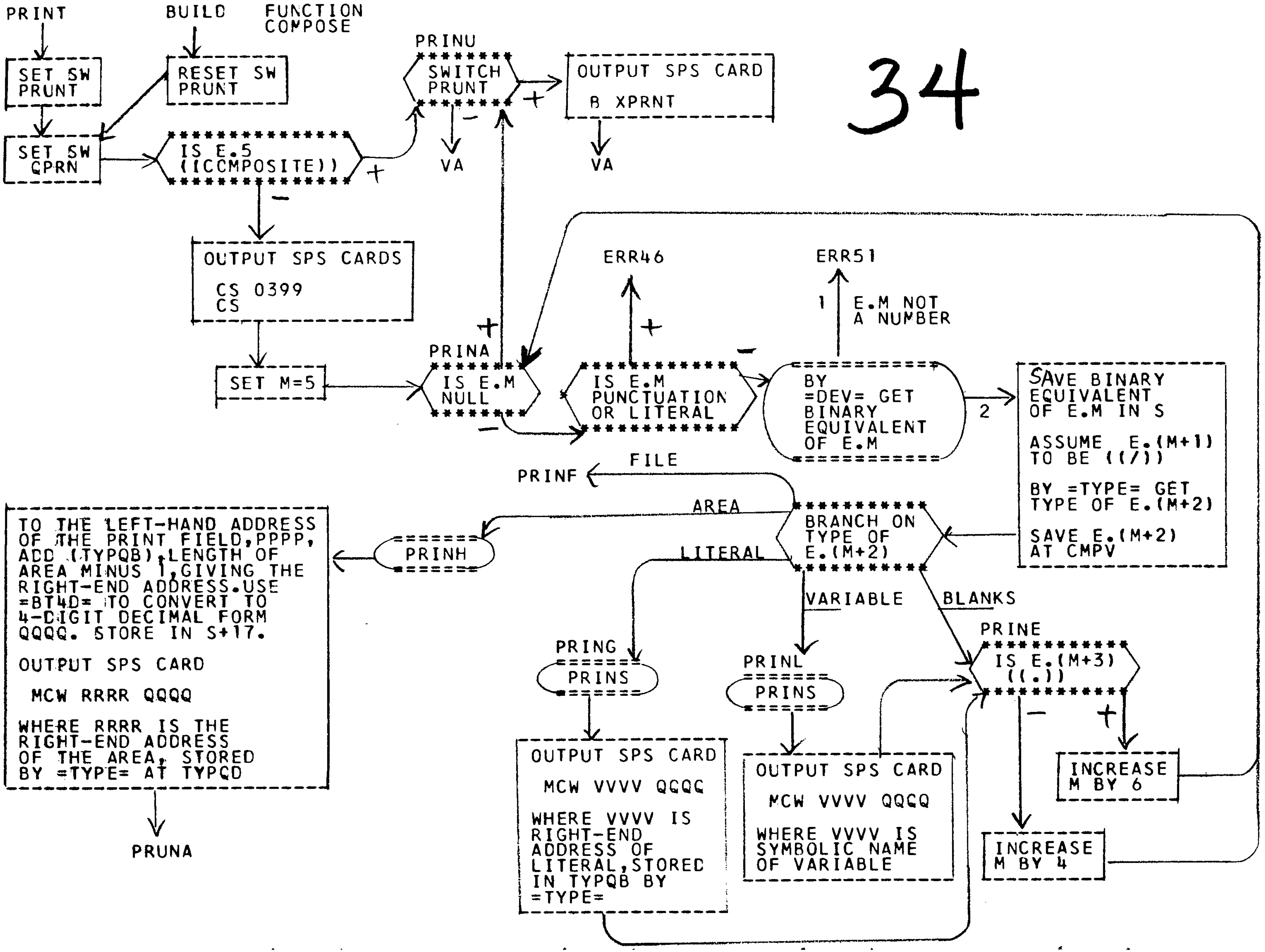
29

IF THE LEFT END OF THE FIELD TO SCAN IS AT X AND THE RIGHT END AT Y,THEN FOR A RIGHTWARD SCAN,INITIALLY XLSC1,XLSC2 AND XSCTP MUST CONTAIN X-1,Y+1,AND 001 (I.E. +1). FOR A RIGHTWARD SCAN THEY MUST INITIALLY CONTAIN,RESPECTIVELY, Y+1,X-1,AND I9I (I.E. -1).

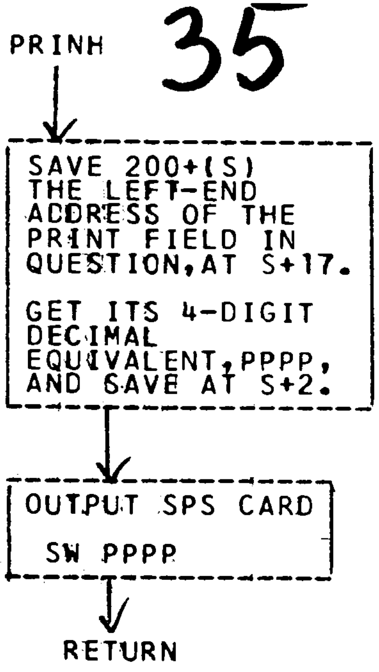


30
SCAN
SETSC
OUTPUT SPS CARDS
MA ZSCTP XLSC1
C XLSC1 XLSC2
B PQR S
MCW XLSC1 * +004
MCW 0000 SCAN
WHERE ((PQR)) IS
E.5 OF STATEMENT
VA
31
RPLC
FUNCTION ((REPLACE SCAN BY))
SETSC
IS E.5 ((SCAN))
+
-
ERR04
IS E.7 ((BY))
+
-
TSX TYPE,4
PZE DRAD+8
FIND TYPE OF SYMBOL AT E.9
ERR61
FILE NAME
((BLANKS))
RPLCA
SET KLMN=7999
AREA
LITERAL OR VARIABLE
SET KLMN=RIGHT-END ADDRESS,PUT INTO 4-DIGIT FORM AT TYPQD BY =TYPE=
RPLCB
SET KLMN=ADDRESS OF RIGHTMOST CHARACTER,PUT IN 4-DIGIT FORM AT TYPQB BY =TYPE=
RPLCD
OUTPUT SPS CARDS
MCW KLMN SCAN
MCW XLSC1 * +007
MCW SCAN 0000
VA
32
RLOC
FUNCTION RELOAD
OUTPUT SPS CARDS
/ 0080
, 0001
1 0001
THIS SEQUENCE OF 1401 INSTRUCTONS LOADS NEXT PROGRAM FROM CARD READER
VA
33
AREA
LABIG
TSX DEV,4
PZE DRAD+6
1
ERR23
E.7 NOT A NUMBER
2
SAVE BINARY EQUIVALENT G IN LOCATION S
TSX DEV,4
PZE DRAD+8
1
ERR23
E.9 NOT A NUMBER
2
LET H= ITS BINARY EQUIVALENT
H LESS THAN G
+
ERR21
-
TSX SCLAB,4
PZE DRAD+4
1
ERR20 SYMBOL ALREADY USED
2 NEW SYMBOL
TSX ATTL,4
PZE DRAD+4
TXH H,,G
PUT SYMBOL IN TABLE AS TYPE 7, WITH LEFT-END AND RIGHT-END ADDRESSES IN BINARY FORM
VA

34


PRINT
BUILD
FUNCTION COMPOSE
SET SW PRUNT
RESET SW PRUNT
SET SW QPRN
IS E.5 ((COMPOSITE))
+
-
PRINU
SWITCH PRUNT
+
-
VA
OUTPUT SPS CARD
B XPRNT
VA
OUTPUT SPS CARDS
CS 0399
CS
SET M=5
PRINA
IS E.M NULL
+
-
IS E.M PUNCTUATION OR LITERAL
+
-
ERR46
ERR51
1
E.M NOT A NUMBER
BY =DEV= GET BINARY EQUIVALENT OF E.M
2
SAVE BINARY EQUIVALENT OF E.M IN S
ASSUME E.(M+1) TO BE ((/))
BY =TYPE= GET TYPE OF E.(M+2)
SAVE E.(M+2) AT CMPV
PRINF
FILE
AREA
BRANCH ON TYPE OF E.(M+2)
LITERAL
VARIABLE
BLANKS
PRINH
TO THE LEFT-HAND ADDRESS OF THE PRINT FIELD,PPPP, ADD (TYPQB),LENGTH OF AREA MINUS 1,GIVING THE RIGHT-END ADDRESS.USE =BT4D= TO CONVERT TO 4-DIGIT DECIMAL FORM QQQQ. STORE IN S+17.
OUTPUT SPS CARD
MCW RRRR QQQQ
WHERE RRRR IS THE RIGHT-END ADDRESS OF THE AREA, STORED BY =TYPE= AT TYPQD
PRUNA
PRING
PRINS
OUTPUT SPS CARD
MCW VVVV QQQC
WHERE VVVV IS RIGHT-END ADDRESS OF LITERAL,STORED IN TYPQB BY =TYPE=
PRINL
PRINS
OUTPUT SPS CARD
MCW VVVV QQCQ
WHERE VVVV IS SYMBOLIC NAME OF VARIABLE
PRINE
IS E.(M+3) ((.))
-
+
INCREASE M BY 6
INCREASE M BY 4

35



PRINH

SAVE 200+(S)
THE LEFT-END
ADDRESS OF THE
PRINT FIELD IN
QUESTION,AT S+17.

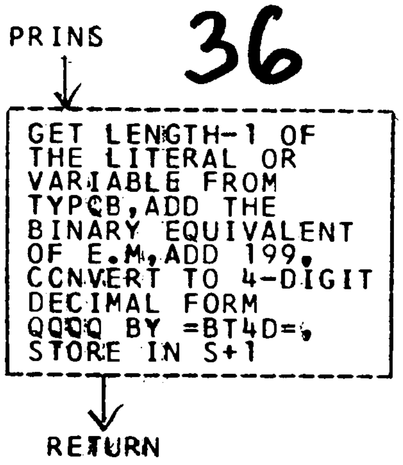
GET ITS 4-DIGIT
DECIMAL
EQUIVALENT,PPPP,
AND SAVE AT S+2.

OUTPUT SPS CARD

SW PPPP

RETURN

36

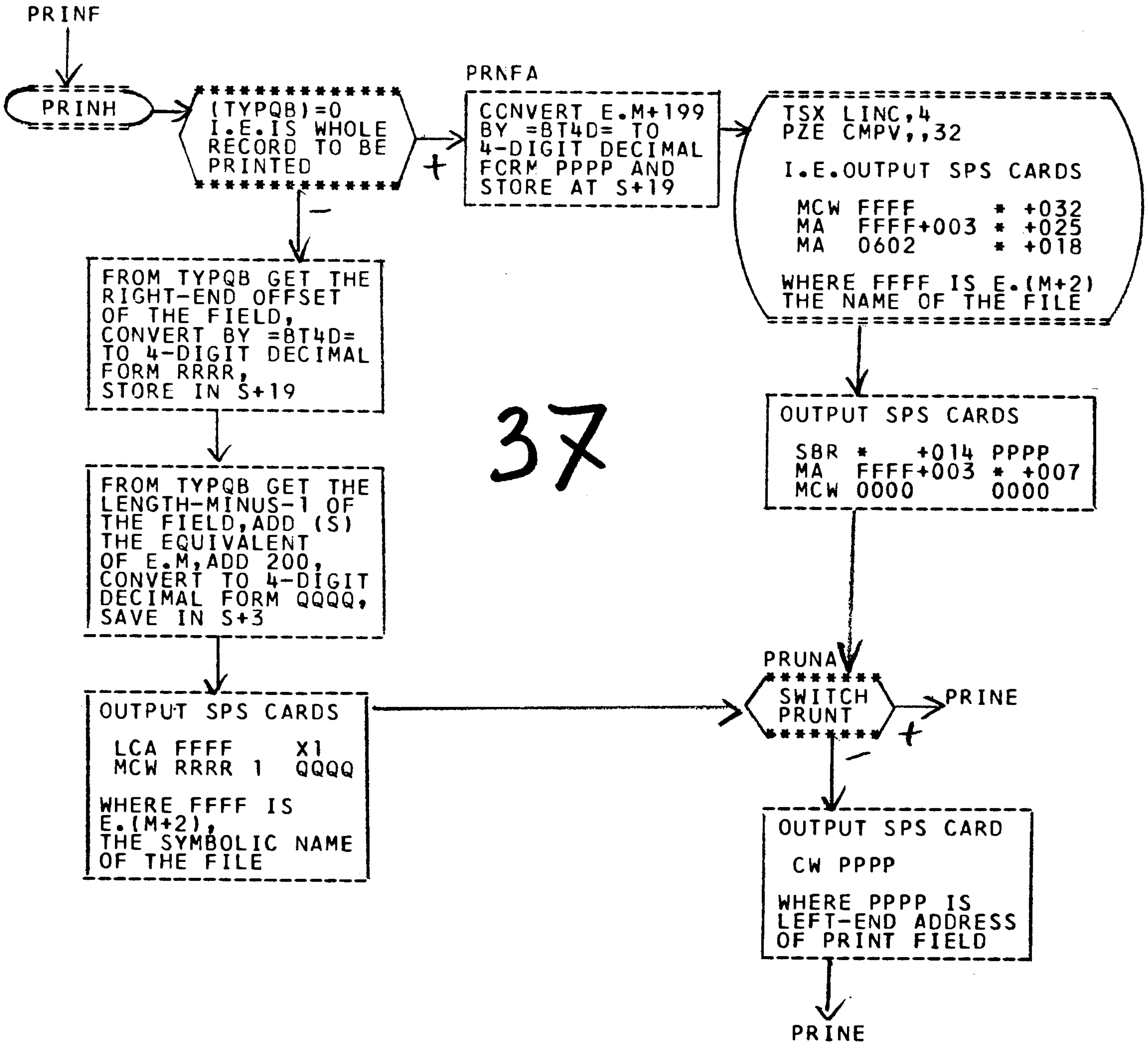


PRINS

GET LENGTH-1 OF
THE LITERAL OR
VARIABLE FROM
TYPQB,ADD THE
BINARY EQUIVALENT
OF E.M,ADD 199,
CCNVERT TO 4-DIGIT
DECIMAL FORM
QQQQ BY =BT4D=,
STORE IN S+1

RETURN

37



PRINF

PRINH

(TYPQB)=0
I.E.IS WHOLE
RECORD TO BE
PRINTED

PRNFA

CCNVERT E.M+199
BY =BT4D= TO
4-DIGIT DECIMAL
FCRM PPPP AND
STORE AT S+19

```
TSX LINC,4
PZE CMPV,,32

I.E.OUTPUT SPS CARDS

 MCW FFFF       * +032
 MA  FFFF+003   * +025
 MA  0602       * +018

WHERE FFFF IS E.(M+2)
THE NAME OF THE FILE
```

FROM TYPQB GET THE
RIGHT-END OFFSET
OF THE FIELD,
CONVERT BY =BT4D=
TO 4-DIGIT DECIMAL
FORM RRRR,
STORE IN S+19

FROM TYPQB GET THE
LENGTH-MINUS-1 OF
THE FIELD,ADD (S)
THE EQUIVALENT
OF E.M,ADD 200,
CONVERT TO 4-DIGIT
DECIMAL FORM QQQQ,
SAVE IN S+3

```
OUTPUT SPS CARDS

 LCA FFFF       X1
 MCW RRRR 1     QQQQ

WHERE FFFF IS
E.(M+2),
THE SYMBOLIC NAME
OF THE FILE
```

```
OUTPUT SPS CARDS

 SBR *     +014 PPPP
 MA  FFFF+003 * +007
 MCW 0000       0000
```

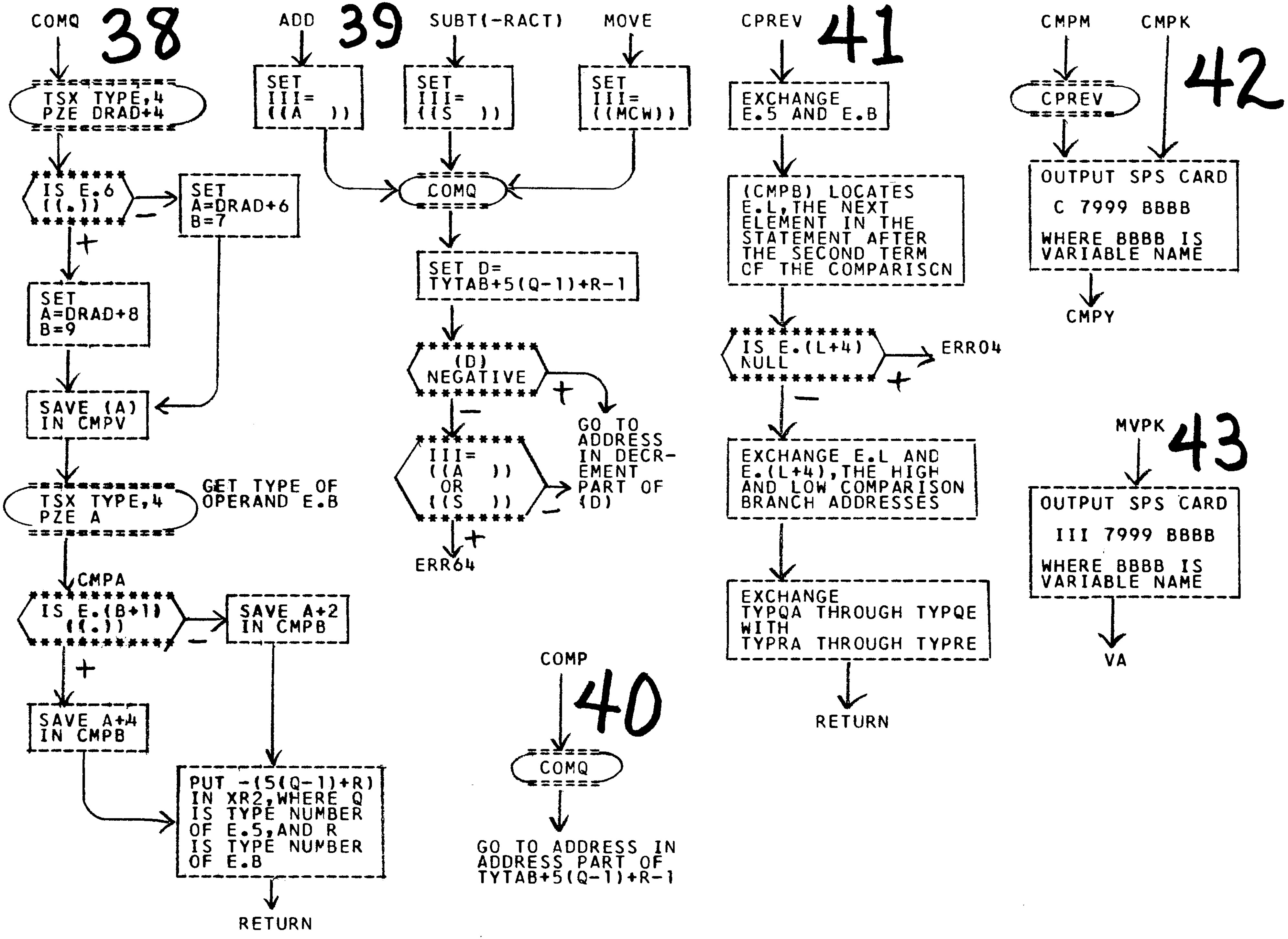
PRUNA

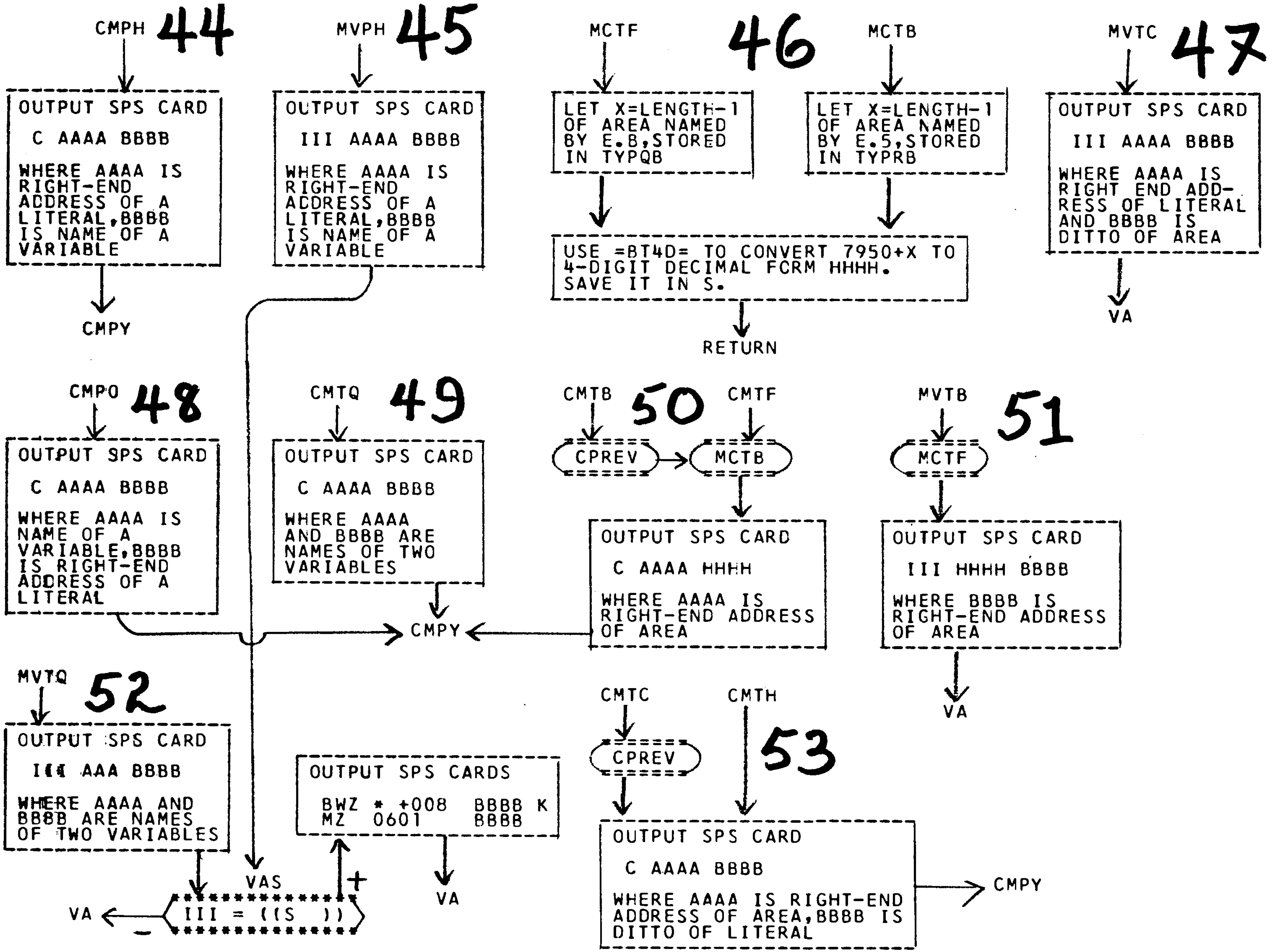
SWITCH
PRUNT

PRINE

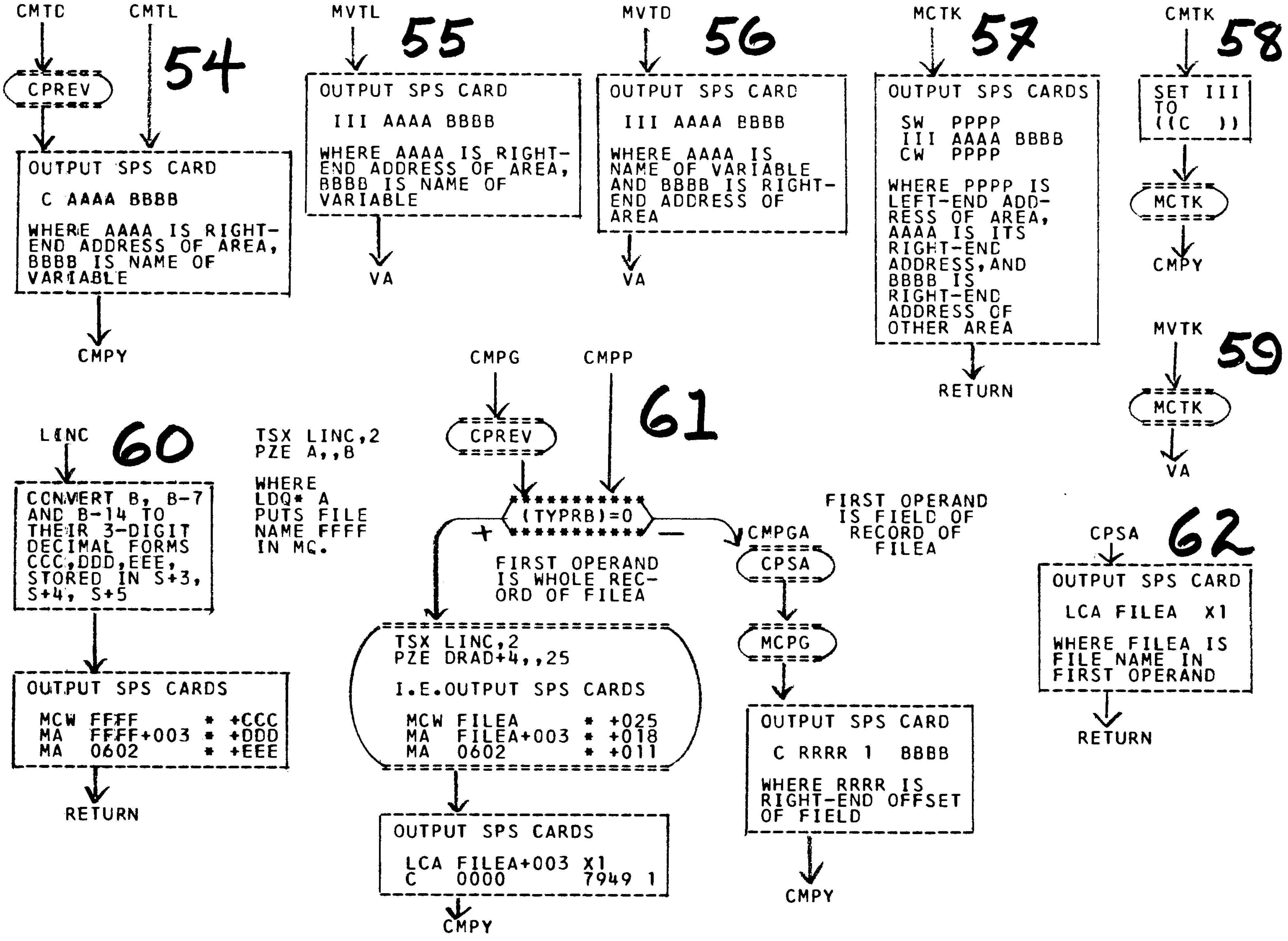
```
OUTPUT SPS CARD

 CW PPPP

WHERE PPPP IS
LEFT-END ADDRESS
OF PRINT FIELD
```

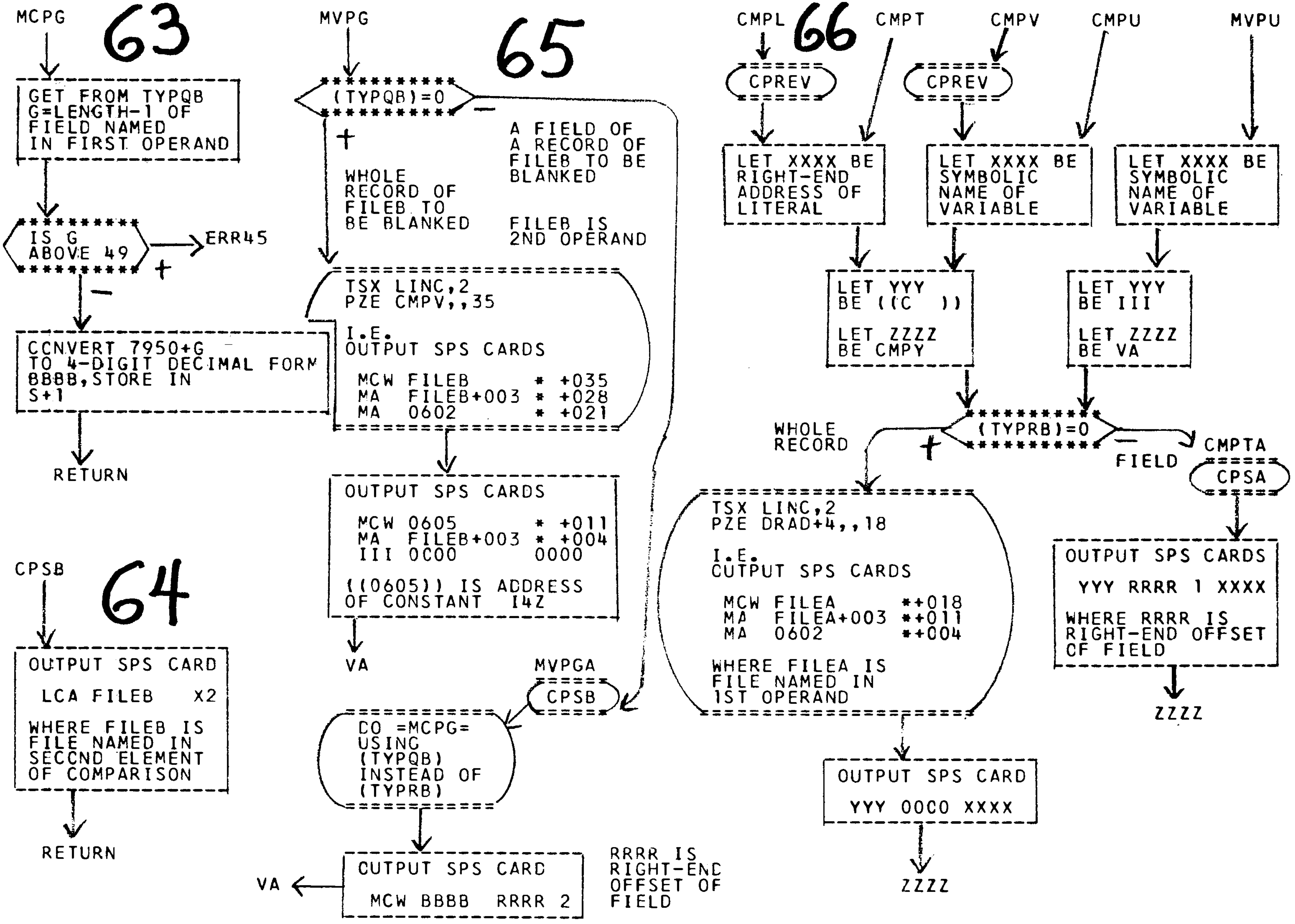
PRINE

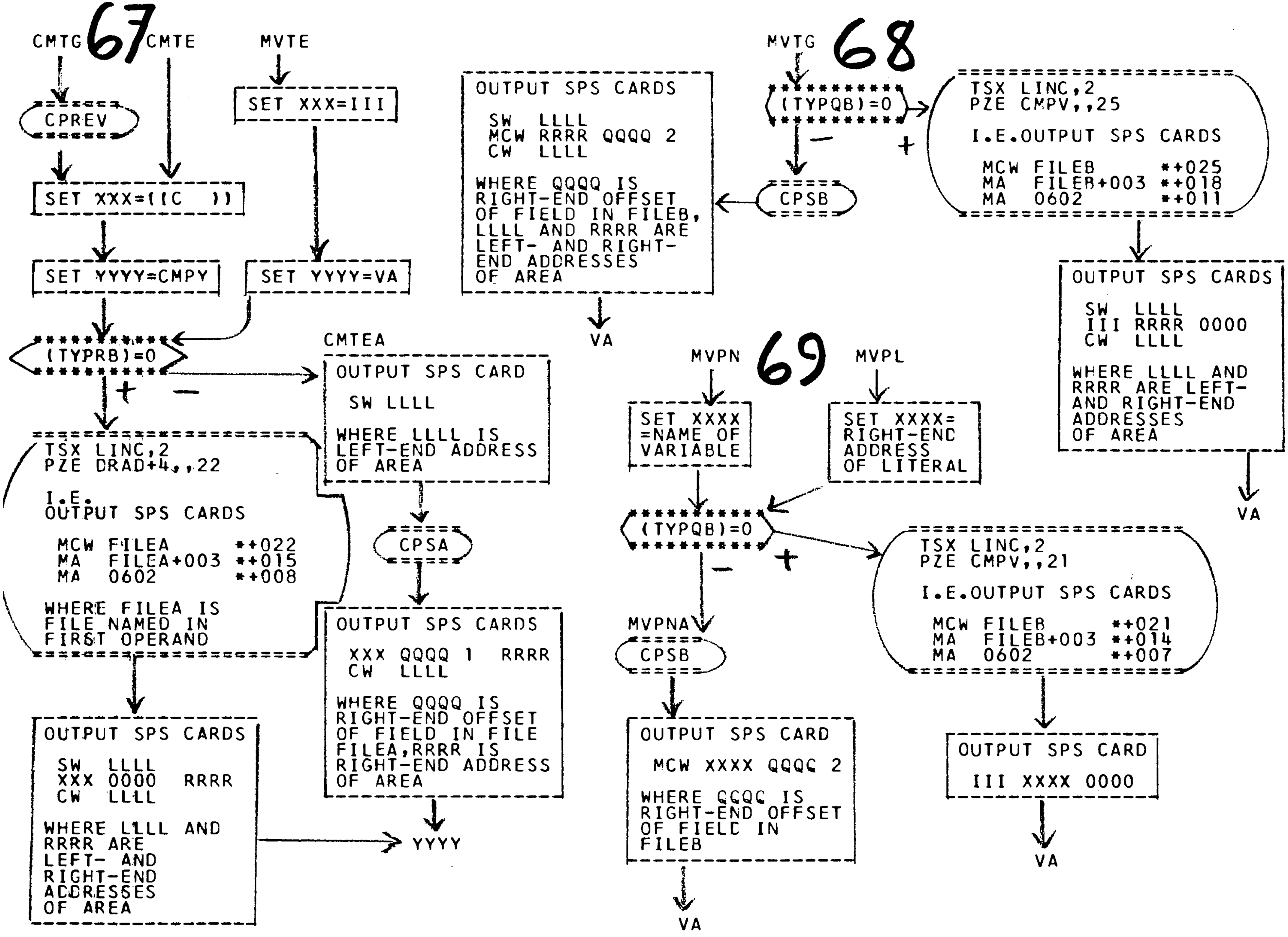

38
COMQ
TSX TYPE,4
PZE DRAD+4
IS E.6
((.))
−
SET
A=DRAD+6
B=7
+
SET
A=DRAD+8
B=9
SAVE (A)
IN CMPV
TSX TYPE,4
PZE A
GET TYPE OF
OPERAND E.B
CMPA
IS E.(B+1)
((.))
−
SAVE A+2
IN CMPB
+
SAVE A+4
IN CMPB
PUT −(5(Q−1)+R)
IN XR2,WHERE Q
IS TYPE NUMBER
OF E.5,AND R
IS TYPE NUMBER
OF E.B
RETURN
39
ADD
SET
III=
((A))
SUBT(−RACT)
SET
III=
((S))
MOVE
SET
III=
((MCW))
COMQ
SET D=
TYTAB+5(Q−1)+R−1
(D)
NEGATIVE
+
GO TO
ADDRESS
IN DECR-
EMENT
PART OF
(D)
−
III=
((A))
OR
((S))
−
+
ERR64
40
COMP
COMQ
GO TO ADDRESS IN
ADDRESS PART OF
TYTAB+5(Q−1)+R−1
41
CPREV
EXCHANGE
E.5 AND E.B
(CMPB) LOCATES
E.L,THE NEXT
ELEMENT IN THE
STATEMENT AFTER
THE SECOND TERM
OF THE COMPARISCN
IS E.(L+4)
NULL
+
ERR04
−
EXCHANGE E.L AND
E.(L+4),THE HIGH
AND LOW COMPARISON
BRANCH ADDRESSES
EXCHANGE
TYPQA THROUGH TYPQE
WITH
TYPRA THROUGH TYPRE
RETURN
42
CMPM
CMPK
CPREV
OUTPUT SPS CARD
C 7999 BBBB
WHERE BBBB IS
VARIABLE NAME
CMPY
43
MVPK
OUTPUT SPS CARD
III 7999 BBBB
WHERE BBBB IS
VARIABLE NAME
VA

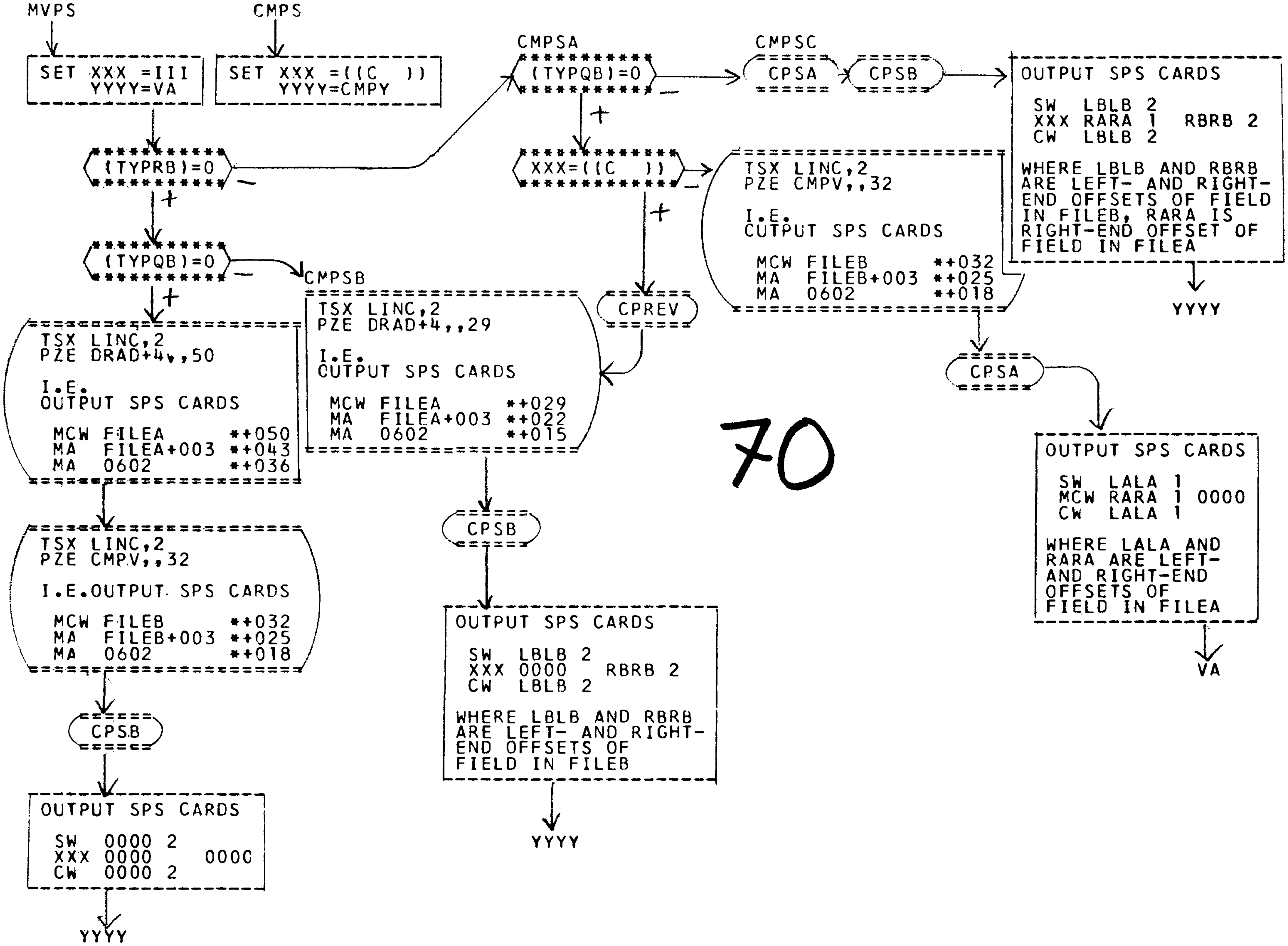


CMPH
44
OUTPUT SPS CARD
C AAAA BBBB
WHERE AAAA IS RIGHT-END ADDRESS OF A LITERAL,BBBB IS NAME OF A VARIABLE
CMPY
MVPH
45
OUTPUT SPS CARD
III AAAA BBBB
WHERE AAAA IS RIGHT-END ADDRESS OF A LITERAL,BBBB IS NAME OF A VARIABLE
MCTF
46
MCTB
LET X=LENGTH-1 OF AREA NAMED BY E.B,STORED IN TYPQB
LET X=LENGTH-1 OF AREA NAMED BY E.5,STORED IN TYPRB
USE =BT4D= TO CONVERT 7950+X TO 4-DIGIT DECIMAL FORM HHHH. SAVE IT IN S.
RETURN
MVTC
47
OUTPUT SPS CARD
III AAAA BBBB
WHERE AAAA IS RIGHT END ADDRESS OF LITERAL AND BBBB IS DITTO OF AREA
VA
CMPO
48
OUTPUT SPS CARD
C AAAA BBBB
WHERE AAAA IS NAME OF A VARIABLE,BBBB IS RIGHT-END ADDRESS OF A LITERAL
CMTQ
49
OUTPUT SPS CARD
C AAAA BBBB
WHERE AAAA AND BBBB ARE NAMES OF TWO VARIABLES
CMPY
CMTB
50
CMTF
CPREV
MCTB
OUTPUT SPS CARD
C AAAA HHHH
WHERE AAAA IS RIGHT-END ADDRESS OF AREA
MVTB
51
MCTF
OUTPUT SPS CARD
III HHHH BBBB
WHERE BBBB IS RIGHT-END ADDRESS OF AREA
VA
MVTQ
52
OUTPUT SPS CARD
III AAA BBBB
WHERE AAAA AND BBBB ARE NAMES OF TWO VARIABLES
VAS
VA
III = ((S))
−
+
OUTPUT SPS CARDS
BWZ * +008 BBBB K
MZ 0601 BBBB
VA
CMTC
CMTH
CPREV
53
OUTPUT SPS CARD
C AAAA BBBB
WHERE AAAA IS RIGHT-END ADDRESS OF AREA,BBBB IS DITTO OF LITERAL
CMPY


CMTC
CMTL
54
CPREV
OUTPUT SPS CARD
C AAAA BBBB
WHERE AAAA IS RIGHT-END ADDRESS OF AREA, BBBB IS NAME OF VARIABLE
CMPY
MVTL
55
OUTPUT SPS CARD
III AAAA BBBB
WHERE AAAA IS RIGHT-END ADDRESS OF AREA, BBBB IS NAME OF VARIABLE
VA
MVTD
56
OUTPUT SPS CARD
III AAAA BBBB
WHERE AAAA IS NAME OF VARIABLE AND BBBB IS RIGHT-END ADDRESS OF AREA
VA
MCTK
57
OUTPUT SPS CARDS
SW PPPP
III AAAA BBBB
CW PPPP
WHERE PPPP IS LEFT-END ADDRESS OF AREA, AAAA IS ITS RIGHT-END ADDRESS, AND BBBB IS RIGHT-END ADDRESS OF OTHER AREA
RETURN
CMTK
58
SET III TO ((C))
MCTK
CMPY
MVTK
59
MCTK
VA
LINC
60
CONVERT B, B-7 AND B-14 TO THEIR 3-DIGIT DECIMAL FORMS CCC,DDD,EEE, STORED IN S+3, S+4, S+5
OUTPUT SPS CARDS
MCW FFFF * +CCC
MA FFFF+003 * +DDD
MA 0602 * +EEE
RETURN
TSX LINC,2
PZE A,,B
WHERE LDQ* A PUTS FILE NAME FFFF IN MQ.
CMPG
CMPP
61
CPREV
(TYPRB)=0
+
−
FIRST OPERAND IS WHOLE RECORD OF FILEA
TSX LINC,2
PZE DRAD+4,,25
I.E.OUTPUT SPS CARDS
MCW FILEA * +025
MA FILEA+003 * +018
MA 0602 * +011
OUTPUT SPS CARDS
LCA FILEA+003 X1
C 0000 7949 1
CMPY
CMPGA
FIRST OPERAND IS FIELD OF RECORD OF FILEA
CPSA
MCPG
OUTPUT SPS CARD
C RRRR 1 BBBB
WHERE RRRR IS RIGHT-END OFFSET OF FIELD
CMPY
CPSA
62
OUTPUT SPS CARD
LCA FILEA X1
WHERE FILEA IS FILE NAME IN FIRST OPERAND
RETURN


63
MCPG
GET FROM TYPQB G=LENGTH-1 OF FIELD NAMED IN FIRST OPERAND
IS G ABOVE 49
+
ERR45
-
CCNVERT 7950+G TO 4-DIGIT DECIMAL FORM BBBB,STORE IN S+1
RETURN
64
CPSB
OUTPUT SPS CARD
LCA FILEB X2
WHERE FILEB IS FILE NAMED IN SECCND ELEMENT OF COMPARISON
RETURN
65
MVPG
(TYPQB)=0
-
A FIELD OF A RECORD OF FILEB TO BE BLANKED
FILEB IS 2ND OPERAND
+
WHOLE RECORD OF FILEB TO BE BLANKED
TSX LINC,2
PZE CMPV,,35
I.E.
OUTPUT SPS CARDS
MCW FILEB * +035
MA FILEB+003 * +028
MA 0602 * +021
OUTPUT SPS CARDS
MCW 0605 * +011
MA FILEB+003 * +004
III 0C00 0000
((0605)) IS ADDRESS OF CONSTANT I4Z
VA
MVPGA
CPSB
CO =MCPG= USING (TYPQB) INSTEAD OF (TYPRB)
CUTPUT SPS CARD
MCW BBBB RRRR 2
VA
RRRR IS RIGHT-ENC OFFSET OF FIELD
66
CMPL
CPREV
LET XXXX BE RIGHT-END ADDRESS OF LITERAL
CMPT
CMPV
CPREV
LET XXXX BE SYMBOLIC NAME OF VARIABLE
CMPU
MVPU
LET XXXX BE SYMBOLIC NAME OF VARIABLE
LET YYY BE ((C))
LET ZZZZ BE CMPY
LET YYY BE III
LET ZZZZ BE VA
(TYPRB)=0
WHOLE RECORD
+
-
FIELD
CMPTA
CPSA
TSX LINC,2
PZE DRAD+4,,18
I.E.
CUTPUT SPS CARDS
MCW FILEA *+018
MA FILEA+003 *+011
MA 0602 *+004
WHERE FILEA IS FILE NAMED IN 1ST OPERAND
OUTPUT SPS CARD
YYY 00C0 XXXX
ZZZZ
OUTPUT SPS CARDS
YYY RRRR 1 XXXX
WHERE RRRR IS RIGHT-END OFFSET OF FIELD
ZZZZ


67
CMTG
CMTE
MVTE
CPREV
SET XXX=III
SET XXX=((C))
SET YYYY=CMPY
SET YYYY=VA
(TYPRB)=0
+
-
CMTEA
OUTPUT SPS CARD
SW LLLL
WHERE LLLL IS LEFT-END ADDRESS OF AREA
TSX LINC,2
PZE DRAD+4,,22
I.E. OUTPUT SPS CARDS
MCW FILEA *+022
MA FILEA+003 *+015
MA 0602 *+008
WHERE FILEA IS FILE NAMED IN FIRST OPERAND
CPSA
OUTPUT SPS CARDS
XXX QQQQ 1 RRRR
CW LLLL
WHERE QQQQ IS RIGHT-END OFFSET OF FIELD IN FILE FILEA,RRRR IS RIGHT-END ADDRESS OF AREA
OUTPUT SPS CARDS
SW LLLL
XXX 0000 RRRR
CW LLLL
WHERE LLLL AND RRRR ARE LEFT- AND RIGHT-END ADDRESSES OF AREA
YYYY
OUTPUT SPS CARDS
SW LLLL
MCW RRRR QQQQ 2
CW LLLL
WHERE QQQQ IS RIGHT-END OFFSET OF FIELD IN FILEB, LLLL AND RRRR ARE LEFT- AND RIGHT-END ADDRESSES OF AREA
VA
68
MVTG
(TYPQB)=0
-
+
CPSB
TSX LINC,2
PZE CMPV,,25
I.E.OUTPUT SPS CARDS
MCW FILEB *+025
MA FILEB+003 *+018
MA 0602 *+011
OUTPUT SPS CARDS
SW LLLL
III RRRR 0000
CW LLLL
WHERE LLLL AND RRRR ARE LEFT- AND RIGHT-END ADDRESSES OF AREA
VA
69
MVPN
MVPL
SET XXXX =NAME OF VARIABLE
SET XXXX= RIGHT-END ADDRESS OF LITERAL
(TYPQB)=0
-
+
MVPNA
CPSB
OUTPUT SPS CARD
MCW XXXX QQQQ 2
WHERE QQQQ IS RIGHT-END OFFSET OF FIELD IN FILEB
VA
TSX LINC,2
PZE CMPV,,21
I.E.OUTPUT SPS CARDS
MCW FILEB *+021
MA FILEB+003 *+014
MA 0602 *+007
OUTPUT SPS CARD
III XXXX 0000
VA



```
SET XXX =III
    YYYY=VA
```

```
SET XXX =((C  ))
    YYYY=CMPY
```

```
TSX LINC,2
PZE DRAD+4,,50

I.E.
OUTPUT SPS CARDS

 MCW FILEA      *+050
 MA  FILEA+003  *+043
 MA  0602       *+036
```

```
TSX LINC,2
PZE CMPV,,32

I.E.OUTPUT SPS CARDS

 MCW FILEB      *+032
 MA  FILEB+003  *+025
 MA  0602       *+018
```

```
OUTPUT SPS CARDS

 SW  0000 2
 XXX 0000      000C
 CW  0000 2
```

```
TSX LINC,2
PZE DRAD+4,,29

I.E.
OUTPUT SPS CARDS

 MCW FILEA      *+029
 MA  FILEA+003  *+022
 MA  0602       *+015
```

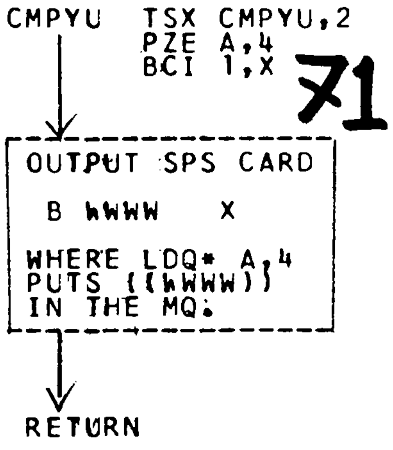
```
OUTPUT SPS CARDS

 SW  LBLB 2
 XXX 0000   RBRB 2
 CW  LBLB 2

WHERE LBLB AND RBRB
ARE LEFT- AND RIGHT-
END OFFSETS OF
FIELD IN FILEB
```

```
TSX LINC,2
PZE CMPV,,32

I.E.
OUTPUT SPS CARDS

 MCW FILEB      *+032
 MA  FILEB+003  *+025
 MA  0602       *+018
```

```
OUTPUT SPS CARDS

 SW  LBLB 2
 XXX RARA 1  RBRB 2
 CW  LBLB 2

WHERE LBLB AND RBRB
ARE LEFT- AND RIGHT-
END OFFSETS OF FIELD
IN FILEB, RARA IS
RIGHT-END OFFSET OF
FIELD IN FILEA
```

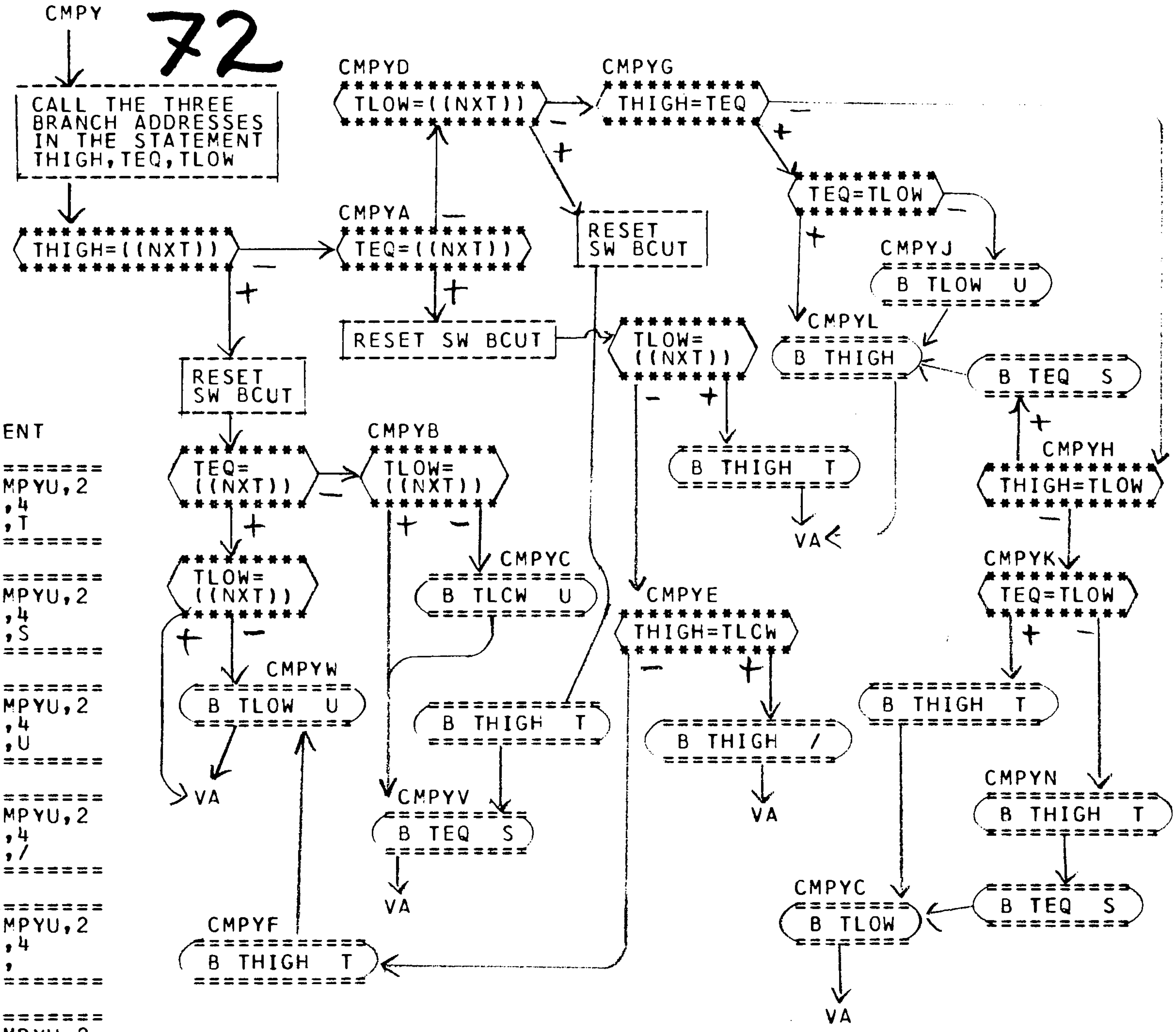
```
OUTPUT SPS CARDS

 SW  LALA 1
 MCW RARA 1 0000
 CW  LALA 1

WHERE LALA AND
RARA ARE LEFT-
AND RIGHT-END
OFFSETS OF
FIELD IN FILEA
```

70

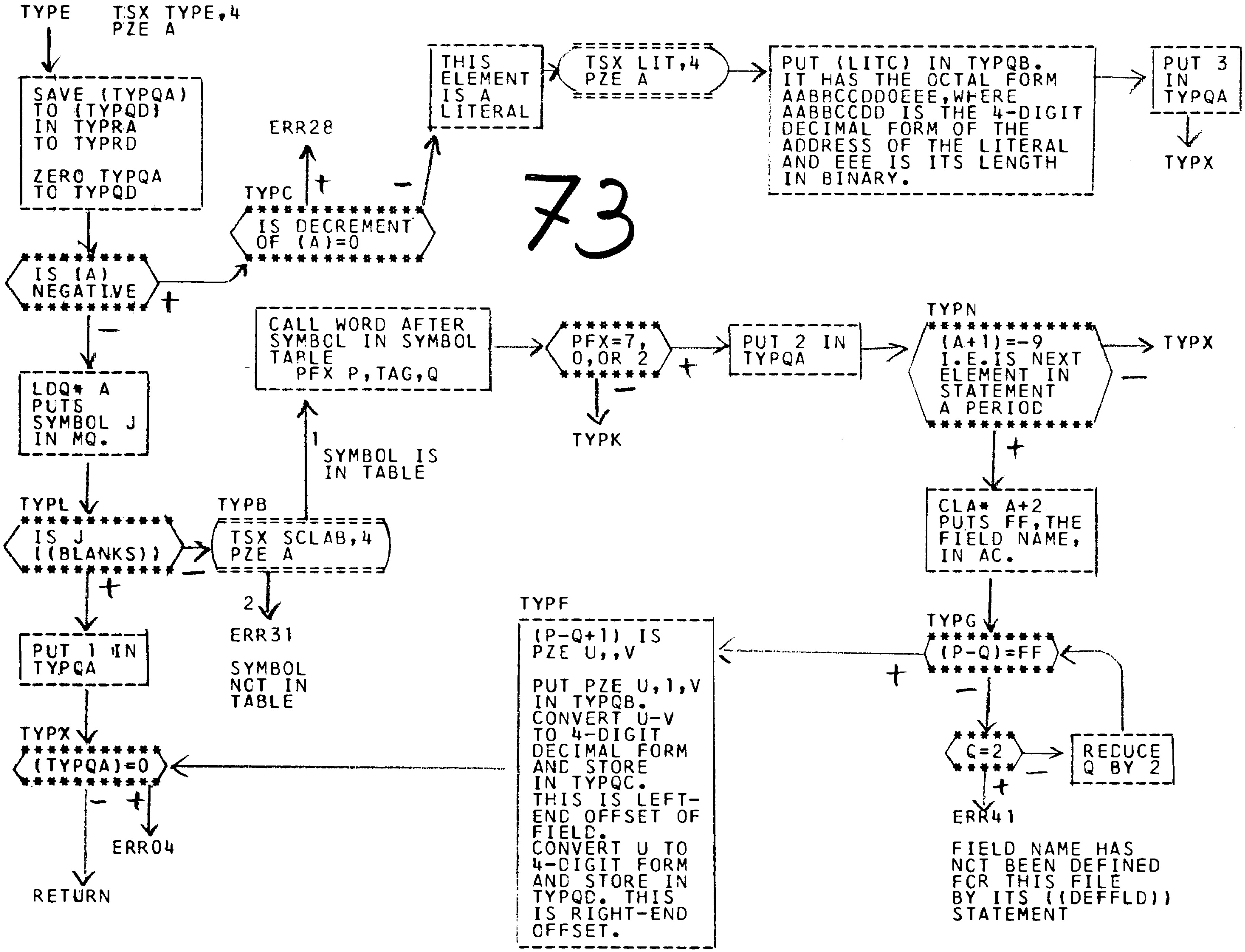


IN THE FLOW CHART ON THE RIGHT,

	REPRESENT
B THIGH T	TSX CMPYU,2 PZE 0,4 BCI 1,T
B TEQ S	TSX CMPYU,2 PZE 2,4 BCI 1,S
B TLOW U	TSX CMPYU,2 PZE 4,4 BCI 1,U
B THIGH /	TSX CMPYU,2 PZE 0,4 BCI 1,/
B THIGH	TSX CMPYU,2 PZE 0,4 BCI 1,
B TLOW	TSX CMPYU,2 PZE 4,4 BCI 1,



73



TYPE
TSX TYPE,4
PZE A
SAVE (TYPQA) TO (TYPQD) IN TYPRA TO TYPRD
ZERO TYPQA TO TYPQD
IS (A) NEGATIVE
+
-
LDQ* A PUTS SYMBOL J IN MQ.
TYPL
IS J ((BLANKS))
+
-
PUT 1 IN TYPQA
TYPX
(TYPQA)=0
-
+
ERR04
RETURN
TYPC
IS DECREMENT OF (A)=0
+
-
ERR28
THIS ELEMENT IS A LITERAL
TSX LIT,4
PZE A
PUT (LITC) IN TYPQB. IT HAS THE OCTAL FORM AABBCCDDOEEE,WHERE AABBCCDD IS THE 4-DIGIT DECIMAL FORM OF THE ADDRESS OF THE LITERAL AND EEE IS ITS LENGTH IN BINARY.
PUT 3 IN TYPQA
TYPX
TYPB
TSX SCLAB,4
PZE A
1
SYMBOL IS IN TABLE
2
ERR31
SYMBOL NCT IN TABLE
CALL WORD AFTER SYMBCL IN SYMBOL TABLE
PFX P,TAG,Q
PFX=7, 0,OR 2
+
-
TYPK
PUT 2 IN TYPQA
TYPN
(A+1)=-9 I.E.IS NEXT ELEMENT IN STATEMENT A PERIOD
-
TYPX
+
CLA* A+2 PUTS FF,THE FIELD NAME, IN AC.
TYPG
(P-Q)=FF
+
-
Q=2
-
+
REDUCE Q BY 2
ERR41
FIELD NAME HAS NCT BEEN DEFINED FCR THIS FILE BY ITS ((DEFFLD)) STATEMENT
TYPF
(P-Q+1) IS PZE U,,V
PUT PZE U,1,V IN TYPQB. CONVERT U-V TO 4-DIGIT DECIMAL FORM AND STORE IN TYPQC. THIS IS LEFT-END OFFSET OF FIELD. CONVERT U TO 4-DIGIT FORM AND STORE IN TYPQD. THIS IS RIGHT-END OFFSET.

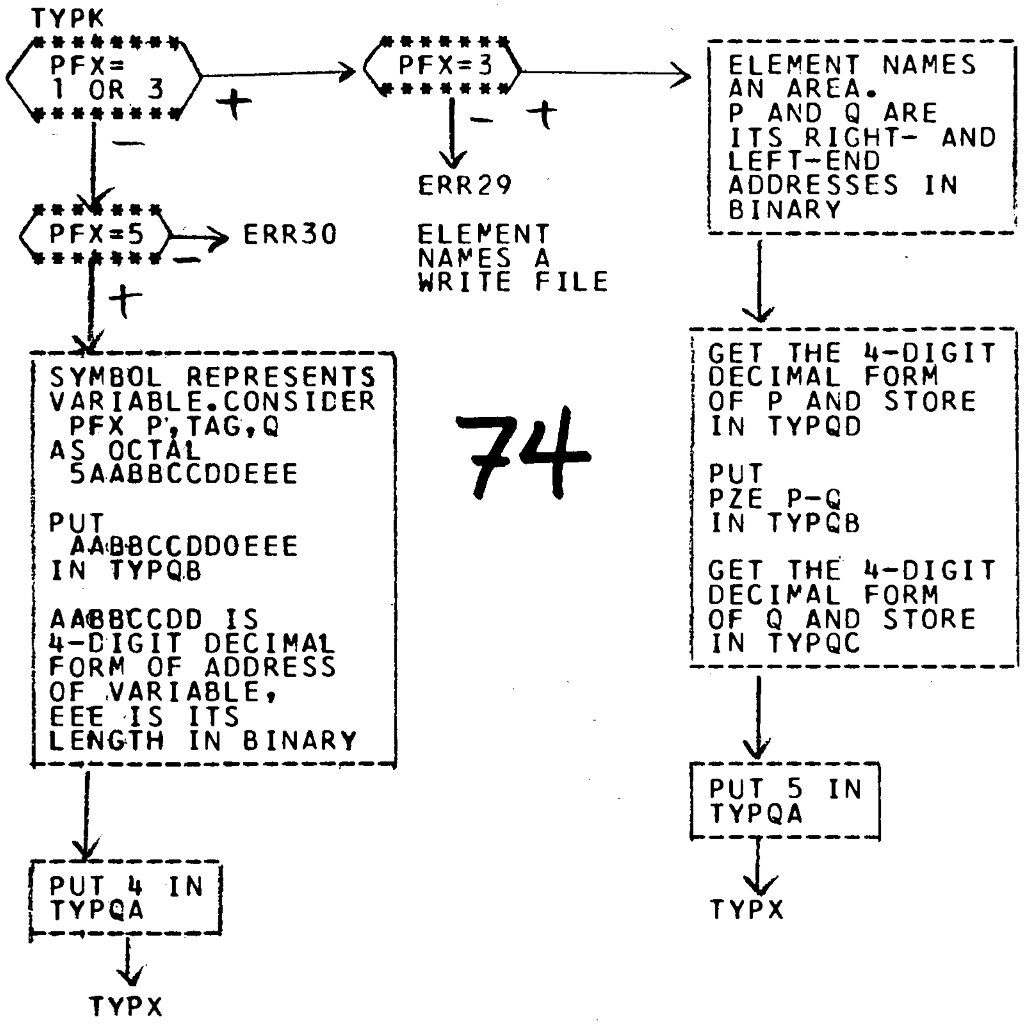
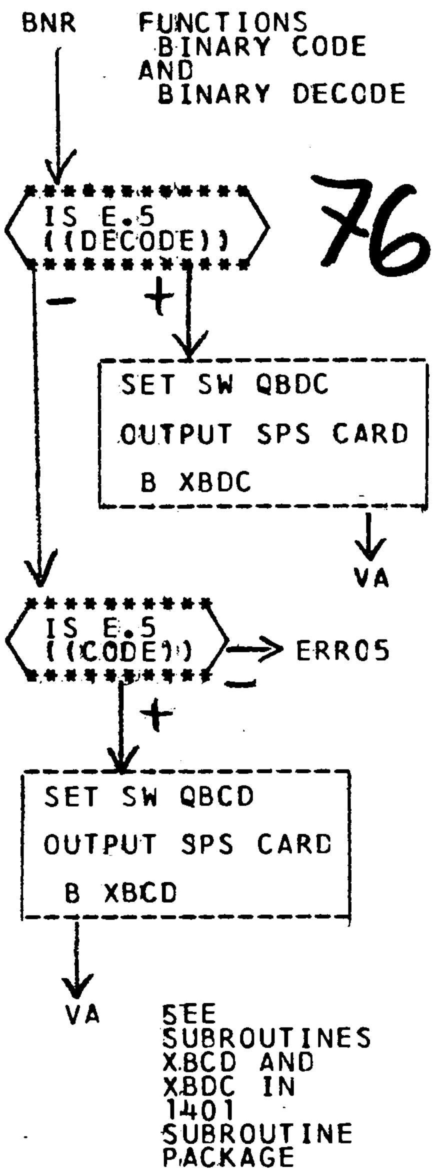

TYPK
PFX=
1 OR 3
+
−
PFX=3
− +
ERR29
ELEMENT NAMES A WRITE FILE
ELEMENT NAMES AN AREA. P AND Q ARE ITS RIGHT- AND LEFT-END ADDRESSES IN BINARY
PFX=5
ERR30
−
+
SYMBOL REPRESENTS VARIABLE.CONSIDER PFX P,TAG,Q AS OCTAL 5AABBCCDDEEE
PUT AABBCCDD0EEE IN TYPQB
AABBCCDD IS 4-DIGIT DECIMAL FORM OF ADDRESS OF VARIABLE, EEE IS ITS LENGTH IN BINARY
74
GET THE 4-DIGIT DECIMAL FORM OF P AND STORE IN TYPQD
PUT PZE P-Q IN TYPQB
GET THE 4-DIGIT DECIMAL FORM OF Q AND STORE IN TYPQC
PUT 4 IN TYPQA
TYPX
PUT 5 IN TYPQA
TYPX

TABLE OF OUTPUTS FROM =TYPE= 75

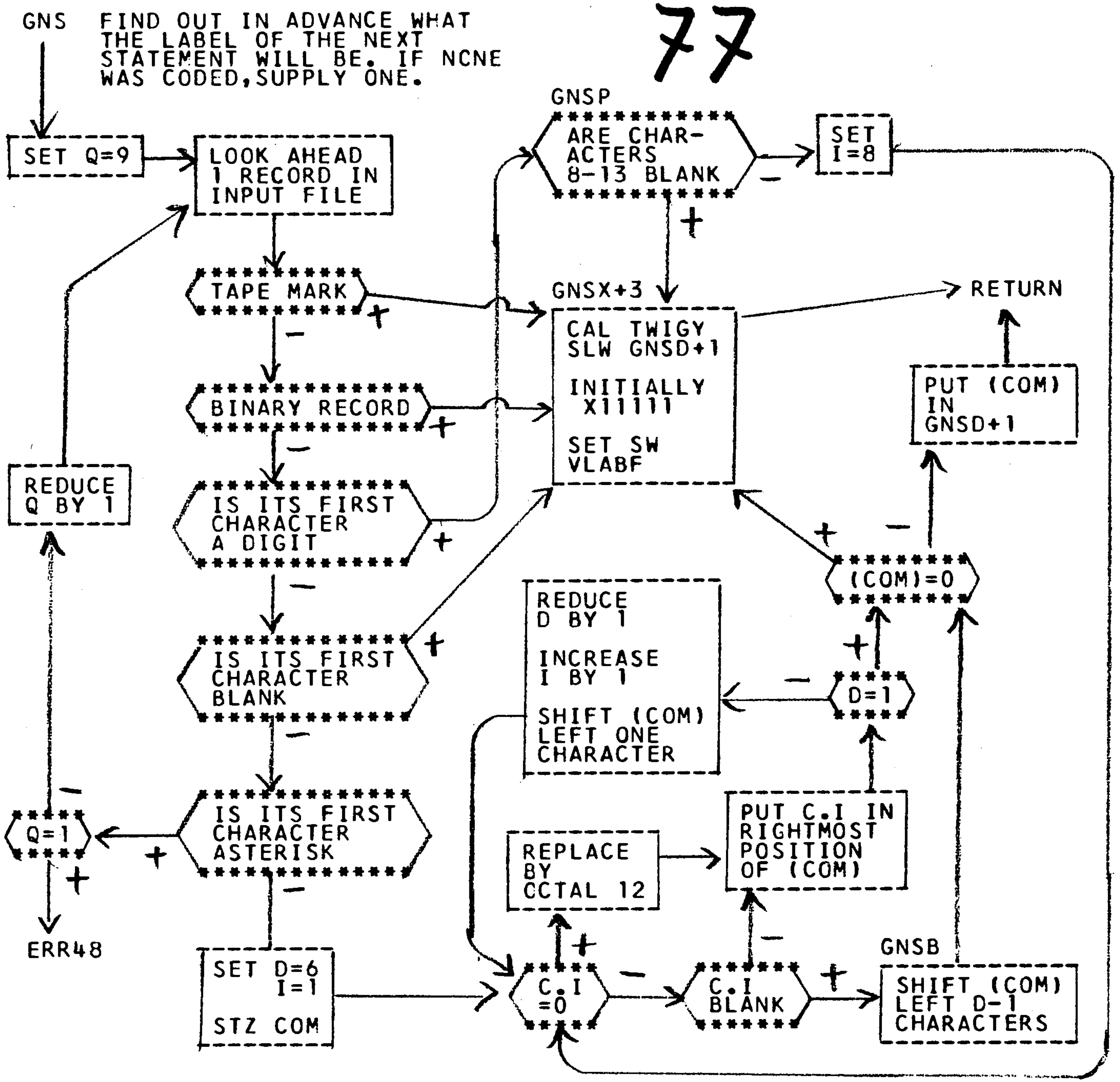
SYMBCL IS	TYPQA	TYPQB	TYPCC	TYPQD
((BLANKS))	1			
READ,COPY OR SAVE FILE NAME ALCNE	2	0		
DITTC WITH FIELC NAME	2	PZE RB,,LG	LDLD	RDRD
LITERAL	3	RCRD-LGG		
VARIABLE	4	RCRD-LGG		
AREA	5	PZE LG	LDLD	RDRD

LG	IS	LENGTH-1 OF FIELD,BINARY
LGG	IS	LENGTH IN BINARY OF LITERAL OR VARIABLE
RB	IS	RIGHT-END OFFSET OF FIELC,IN BINARY
LDLD	IS	LEFT-END OFFSET OF FIELD, OR LEFT-END ADDRESS OF AREA,IN 4-DIGIT DECIMAL
RDRD	IS	RIGHT-END DITTO

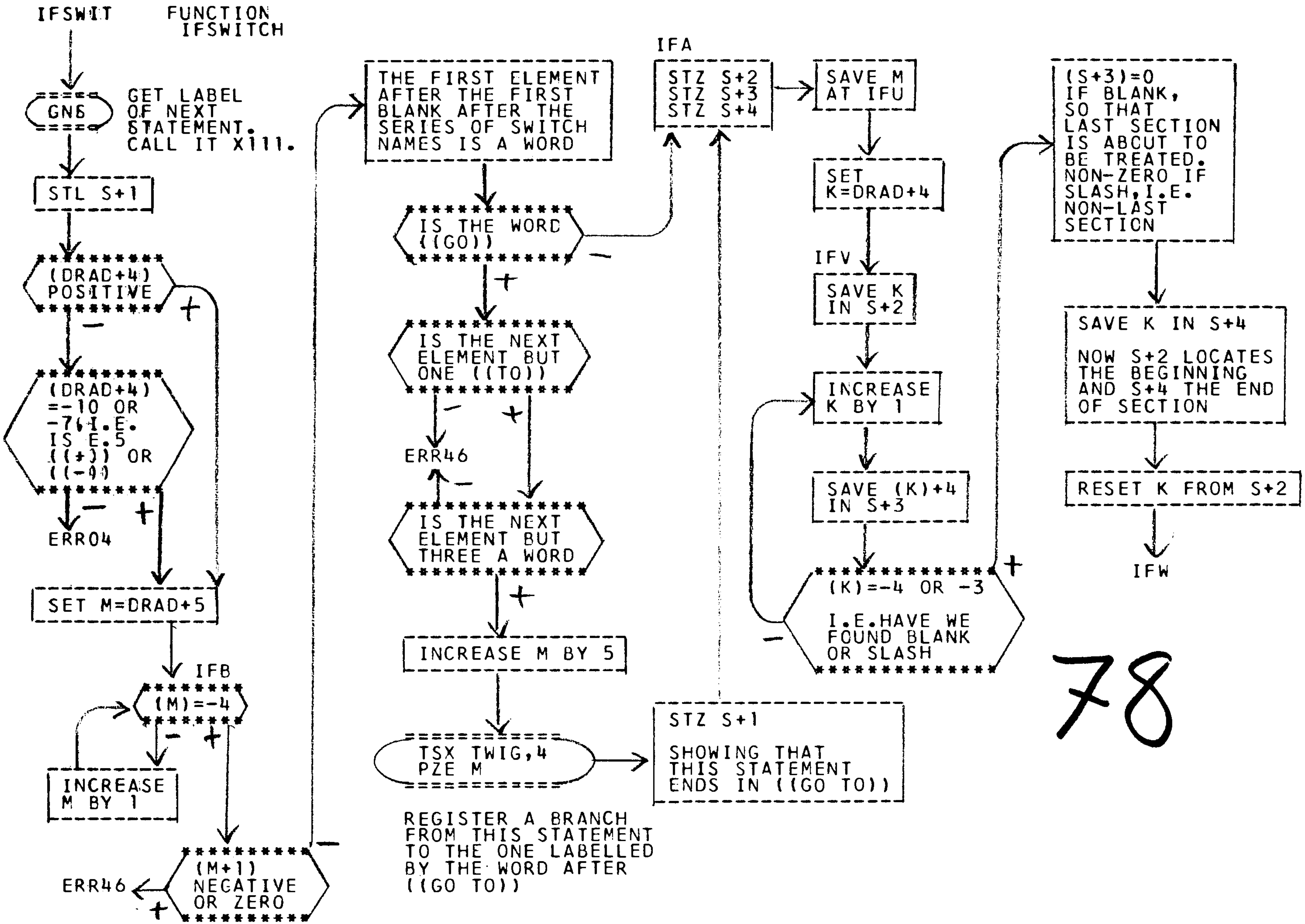


BNR FUNCTIONS BINARY CODE AND BINARY DECODE

SEE SUBROUTINES XBCD AND XBDC IN 1401 SUBROUTINE PACKAGE

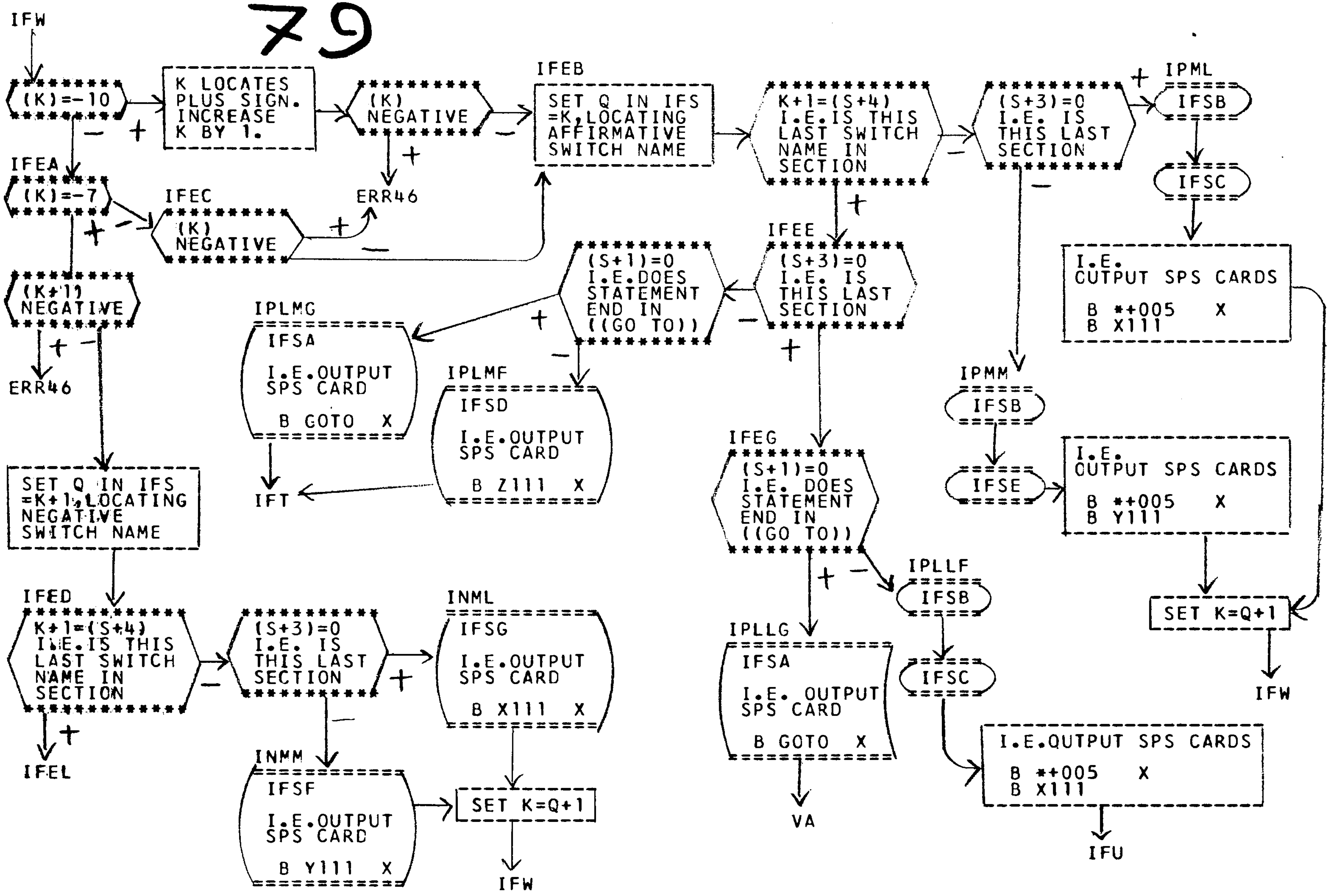


GNS FIND OUT IN ADVANCE WHAT THE LABEL OF THE NEXT STATEMENT WILL BE. IF NCNE WAS CODED, SUPPLY ONE.

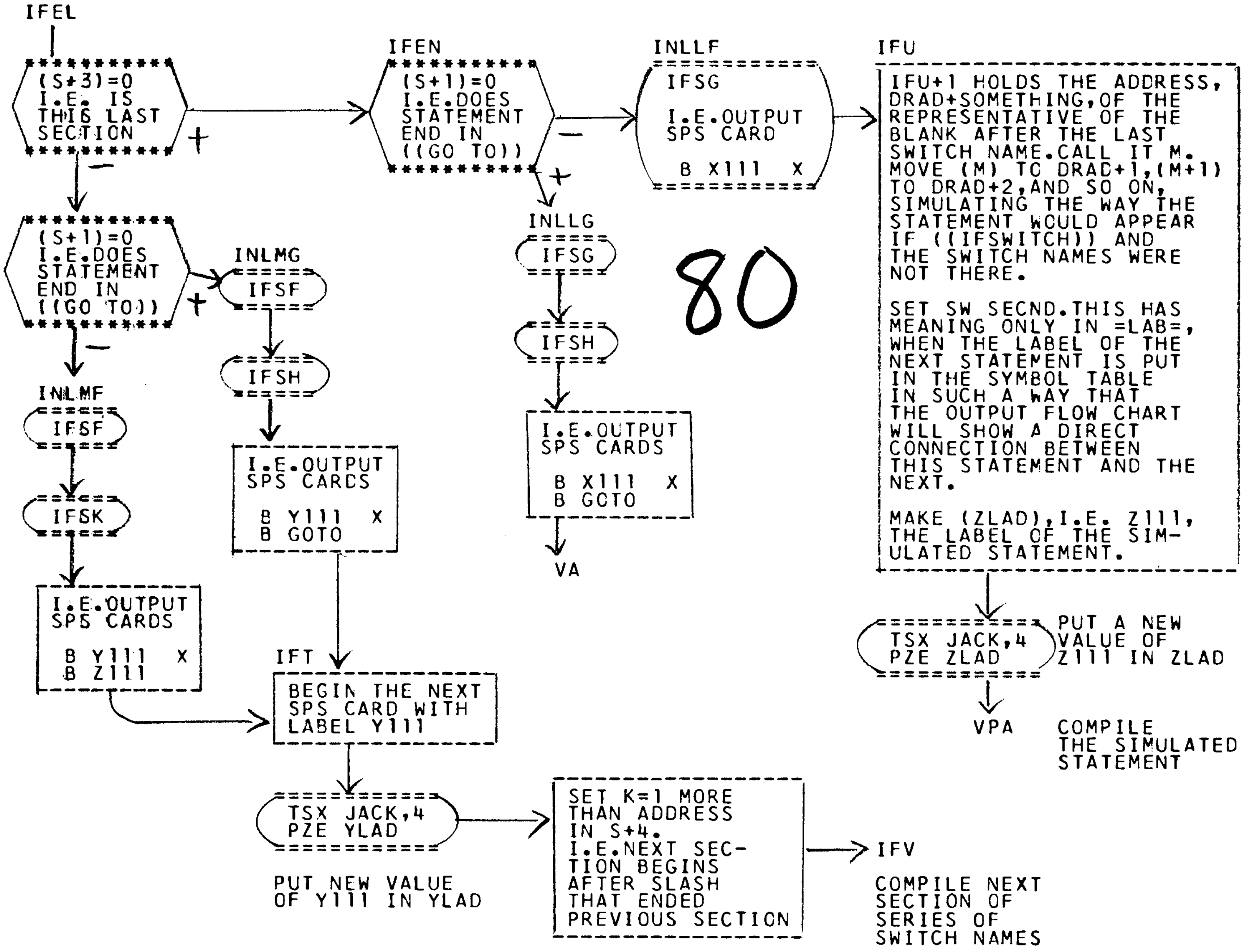

IFSWIT FUNCTION IFSWITCH
GN6
GET LABEL OF NEXT STATEMENT. CALL IT X111.
STL S+1
(DRAD+4) POSITIVE
+
-
(DRAD+4) =-10 OR -7 I.E. IS E.5 ((+)) OR ((-))
-
+
ERR04
SET M=DRAD+5
IFB
(M)=-4
-
+
INCREASE M BY 1
(M+1) NEGATIVE OR ZERO
+
ERR46
-
THE FIRST ELEMENT AFTER THE FIRST BLANK AFTER THE SERIES OF SWITCH NAMES IS A WORD
IS THE WORD ((GO))
-
+
IS THE NEXT ELEMENT BUT ONE ((TO))
-
+
ERR46
-
IS THE NEXT ELEMENT BUT THREE A WORD
+
INCREASE M BY 5
TSX TWIG,4 PZE M
REGISTER A BRANCH FROM THIS STATEMENT TO THE ONE LABELLED BY THE WORD AFTER ((GO TO))
STZ S+1
SHOWING THAT THIS STATEMENT ENDS IN ((GO TO))
IFA
STZ S+2 STZ S+3 STZ S+4
SAVE M AT IFU
SET K=DRAD+4
IFV
SAVE K IN S+2
INCREASE K BY 1
SAVE (K)+4 IN S+3
(K)=-4 OR -3
I.E.HAVE WE FOUND BLANK OR SLASH
-
+
(S+3)=0 IF BLANK, SO THAT LAST SECTION IS ABCUT TO BE TREATED. NON-ZERO IF SLASH,I.E. NON-LAST SECTION
SAVE K IN S+4
NOW S+2 LOCATES THE BEGINNING AND S+4 THE END OF SECTION
RESET K FROM S+2
IFW

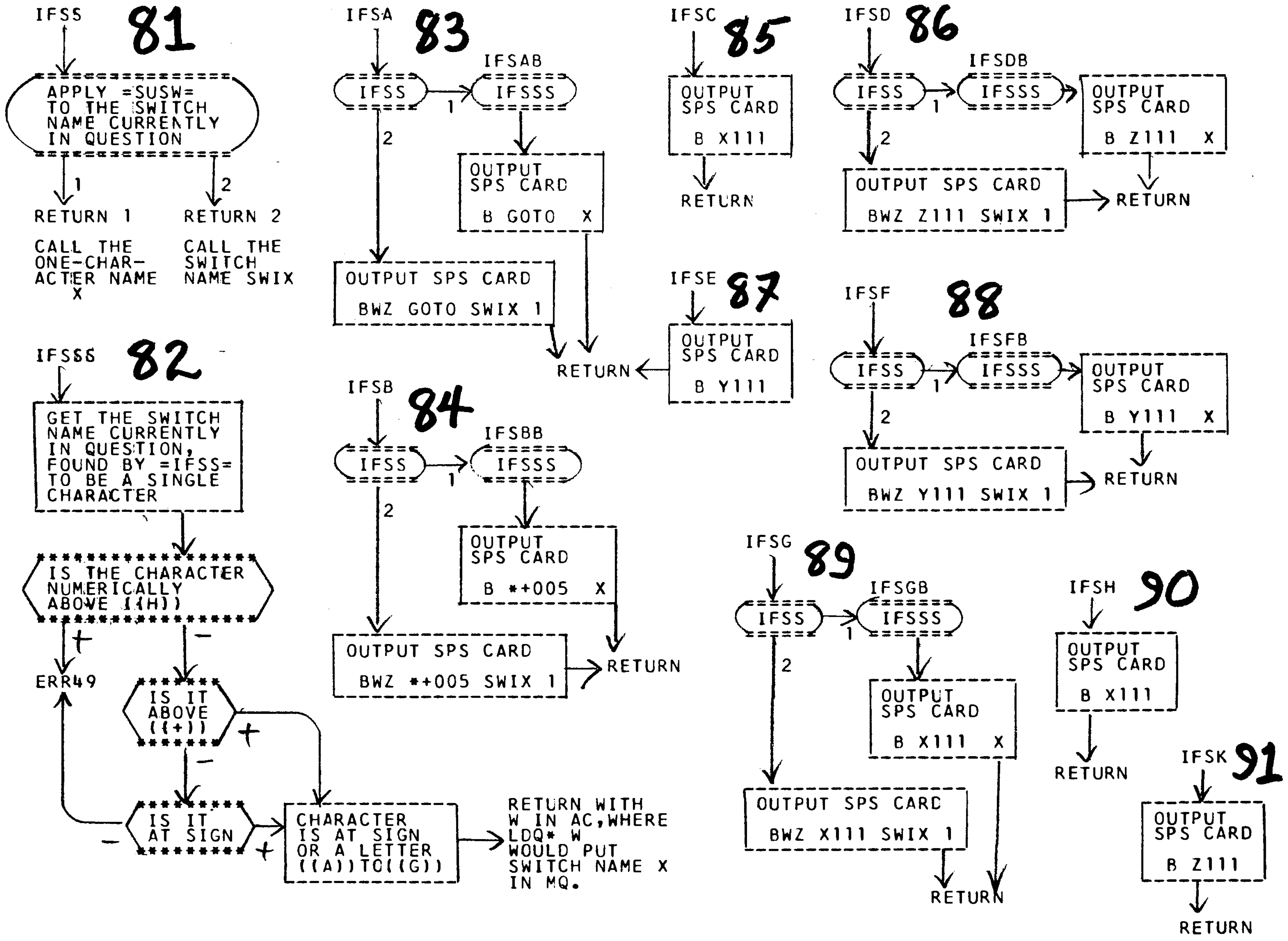
78

79

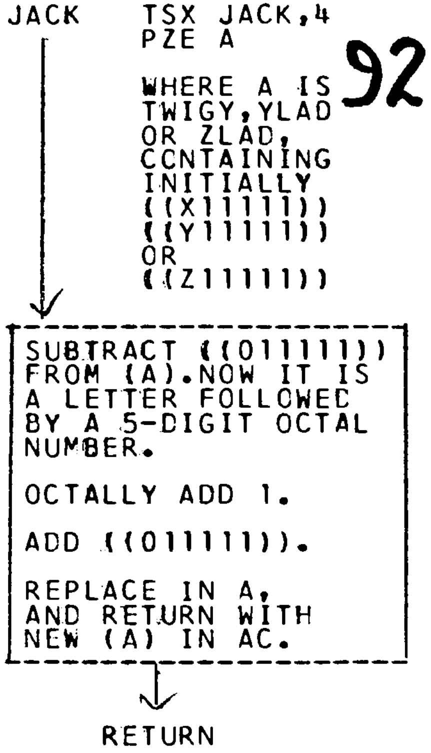


IFW
(K)=-10
K LOCATES PLUS SIGN. INCREASE K BY 1.
(K) NEGATIVE
ERR46
IFEB
SET Q IN IFS =K,LOCATING AFFIRMATIVE SWITCH NAME
K+1=(S+4) I.E.IS THIS LAST SWITCH NAME IN SECTION
(S+3)=0 I.E. IS THIS LAST SECTION
IPML
IFSB
IFSC
I.E. OUTPUT SPS CARDS B *+005 X B X111
IFEA
(K)=-7
IFEC
(K) NEGATIVE
ERR46
(K+1) NEGATIVE
ERR46
IFEE
(S+3)=0 I.E. IS THIS LAST SECTION
(S+1)=0 I.E.DOES STATEMENT END IN ((GO TO))
IPLMG
IFSA I.E.OUTPUT SPS CARD B GOTO X
IFT
IPLMF
IFSD I.E.OUTPUT SPS CARD B Z111 X
IPMM
IFSB
IFSE
I.E. OUTPUT SPS CARDS B *+005 X B Y111
SET K=Q+1
IFW
SET Q IN IFS =K+1,LOCATING NEGATIVE SWITCH NAME
IFEG
(S+1)=0 I.E. DOES STATEMENT END IN ((GO TO))
IPLLF
IFSB
IFSC
I.E.OUTPUT SPS CARDS B *+005 X B X111
IFU
IPLLG
IFSA I.E. OUTPUT SPS CARD B GOTO X
VA
IFED
K+1=(S+4) I.E.IS THIS LAST SWITCH NAME IN SECTION
IFEL
(S+3)=0 I.E. IS THIS LAST SECTION
INML
IFSG I.E.OUTPUT SPS CARD B X111 X
INMM
IFSF I.E.OUTPUT SPS CARD B Y111 X
SET K=Q+1
IFW
+
-

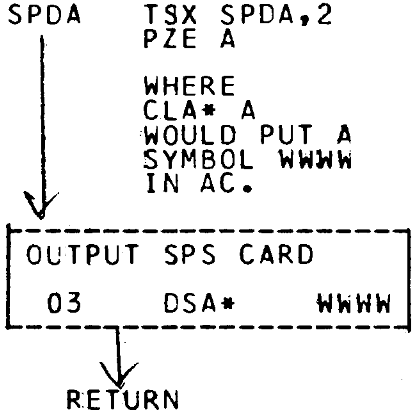

IFEL
(S+3)=0
I.E. IS
THIS LAST
SECTION
+
−
(S+1)=0
I.E.DOES
STATEMENT
END IN
((GO TO))
+
−
INLMF
IFSF
IFSK
I.E.OUTPUT
SPS CARDS
B Y111 X
B Z111
INLMG
IFSF
IFSH
I.E.OUTPUT
SPS CARDS
B Y111 X
B GOTO
IFT
BEGIN THE NEXT
SPS CARD WITH
LABEL Y111
TSX JACK,4
PZE YLAD
PUT NEW VALUE
OF Y111 IN YLAD
SET K=1 MORE
THAN ADDRESS
IN S+4.
I.E.NEXT SEC-
TION BEGINS
AFTER SLASH
THAT ENDED
PREVIOUS SECTION
IFV
COMPILE NEXT
SECTION OF
SERIES OF
SWITCH NAMES
IFEN
(S+1)=0
I.E.DOES
STATEMENT
END IN
((GO TO))
−
+
INLLG
IFSG
IFSH
I.E.OUTPUT
SPS CARDS
B X111 X
B GOTO
VA
INLLF
IFSG
I.E.OUTPUT
SPS CARD
B X111 X
IFU
IFU+1 HOLDS THE ADDRESS,
DRAD+SOMETHING,OF THE
REPRESENTATIVE OF THE
BLANK AFTER THE LAST
SWITCH NAME.CALL IT M.
MOVE (M) TO DRAD+1,(M+1)
TO DRAD+2,AND SO ON,
SIMULATING THE WAY THE
STATEMENT WOULD APPEAR
IF ((IFSWITCH)) AND
THE SWITCH NAMES WERE
NOT THERE.
SET SW SECND.THIS HAS
MEANING ONLY IN =LAB=,
WHEN THE LABEL OF THE
NEXT STATEMENT IS PUT
IN THE SYMBOL TABLE
IN SUCH A WAY THAT
THE OUTPUT FLOW CHART
WILL SHOW A DIRECT
CONNECTION BETWEEN
THIS STATEMENT AND THE
NEXT.
MAKE (ZLAD),I.E. Z111,
THE LABEL OF THE SIM-
ULATED STATEMENT.
TSX JACK,4
PZE ZLAD
PUT A NEW
VALUE OF
Z111 IN ZLAD
VPA
COMPILE
THE SIMULATED
STATEMENT
80


IFSS
81
APPLY =SUSW= TO THE SWITCH NAME CURRENTLY IN QUESTION
1
2
RETURN 1
RETURN 2
CALL THE ONE-CHAR-ACTER NAME X
CALL THE SWITCH NAME SWIX
IFSSS
82
GET THE SWITCH NAME CURRENTLY IN QUESTION, FOUND BY =IFSS= TO BE A SINGLE CHARACTER
IS THE CHARACTER NUMERICALLY ABOVE ((H))
+
-
ERR49
IS IT ABOVE ((+))
+
-
IS IT AT SIGN
-
+
CHARACTER IS AT SIGN OR A LETTER ((A))TO((G))
RETURN WITH W IN AC,WHERE LDQ* W WOULD PUT SWITCH NAME X IN MQ.
IFSA
83
IFSS
1
2
IFSAB
IFSSS
OUTPUT SPS CARD B GOTO X
OUTPUT SPS CARD BWZ GOTO SWIX 1
RETURN
IFSB
84
IFSS
1
2
IFSBB
IFSSS
OUTPUT SPS CARD B *+005 X
OUTPUT SPS CARD BWZ *+005 SWIX 1
RETURN
IFSC
85
OUTPUT SPS CARD B X111
RETURN
IFSE
87
OUTPUT SPS CARD B Y111
IFSD
86
IFSS
1
2
IFSDB
IFSSS
OUTPUT SPS CARD B Z111 X
OUTPUT SPS CARD BWZ Z111 SWIX 1
RETURN
IFSF
88
IFSS
1
2
IFSFB
IFSSS
OUTPUT SPS CARD B Y111 X
OUTPUT SPS CARD BWZ Y111 SWIX 1
RETURN
IFSG
89
IFSS
1
2
IFSGB
IFSSS
OUTPUT SPS CARD B X111 X
OUTPUT SPS CARD BWZ X111 SWIX 1
RETURN
IFSH
90
OUTPUT SPS CARD B X111
RETURN
IFSK
91
OUTPUT SPS CARD B Z111
RETURN

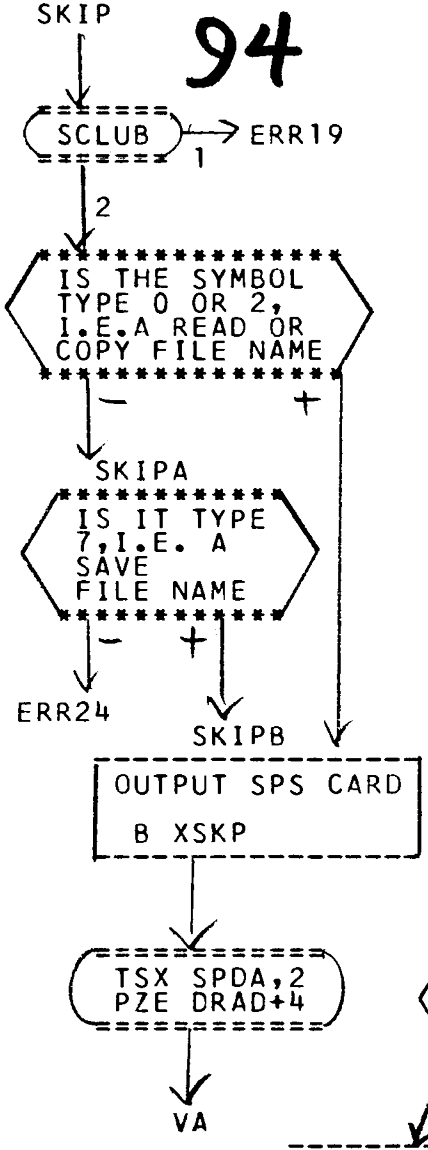
92


JACK
TSX JACK,4
PZE A
WHERE A IS TWIGY,YLAD OR ZLAD, CCNTAINING INITIALLY ((X11111)) ((Y11111)) OR ((Z11111))
SUBTRACT ((011111)) FROM (A).NOW IT IS A LETTER FOLLOWEC BY A 5-DIGIT OCTAL NUMBER.
OCTALLY ADD 1.
ADD ((011111)).
REPLACE IN A, AND RETURN WITH NEW (A) IN AC.
RETURN

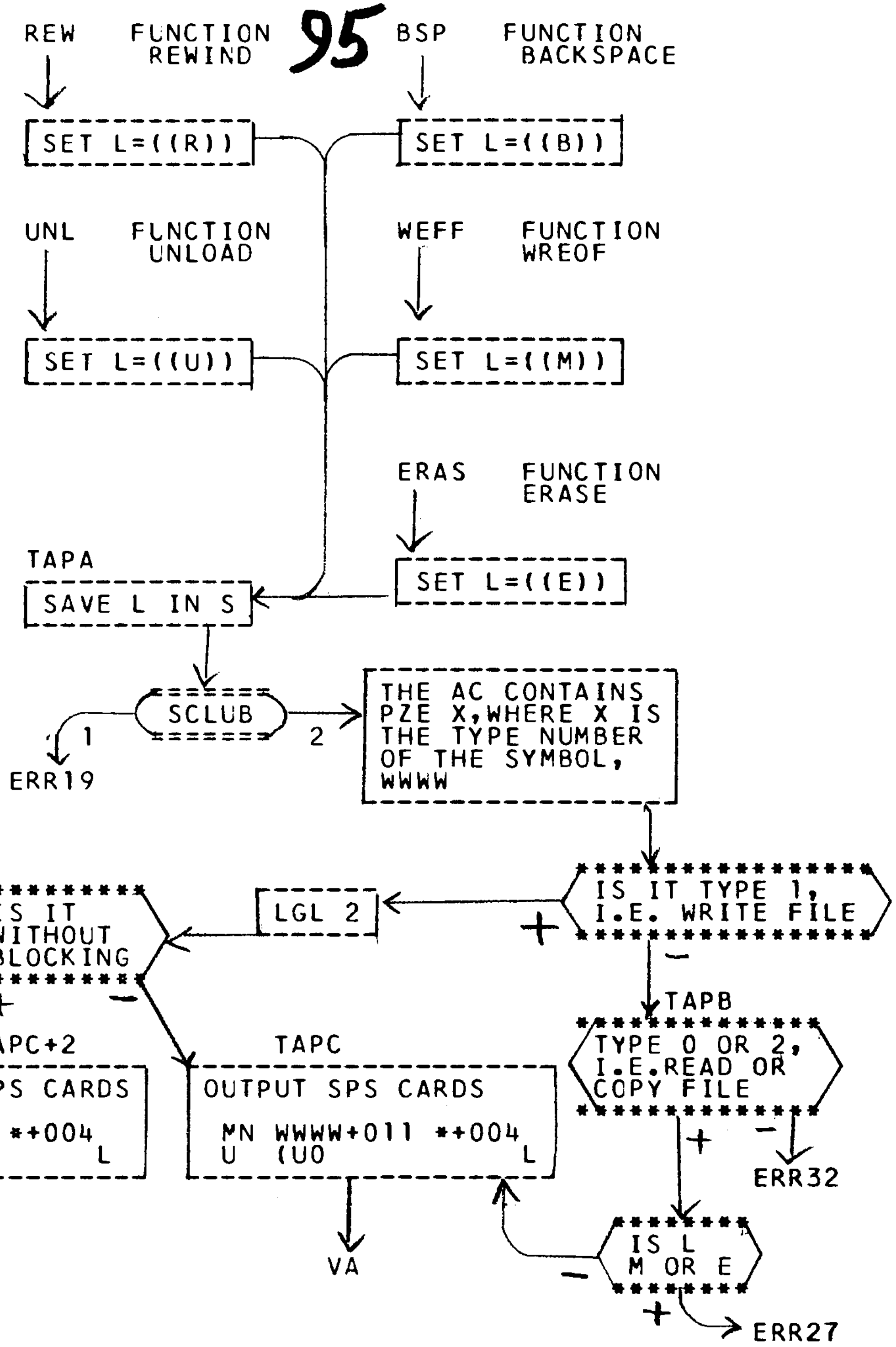
93

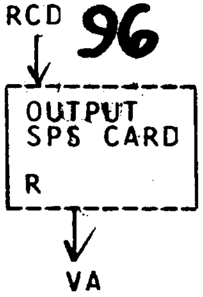

SPDA
TSX SPDA,2
PZE A
WHERE CLA* A WOULD PUT A SYMBOL WWWW IN AC.
OUTPUT SPS CARD
03 DSA* WWWW
RETURN

94

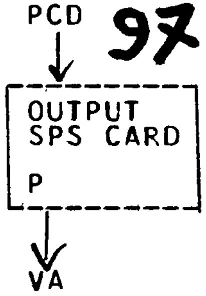

SKIP
SCLUB
ERR19
1
2
IS THE SYMBOL TYPE 0 OR 2, I.E.A READ OR COPY FILE NAME
−
+
SKIPA
IS IT TYPE 7,I.E. A SAVE FILE NAME
−
+
ERR24
SKIPB
OUTPUT SPS CARD
B XSKP
TSX SPDA,2
PZE DRAD+4
VA

95

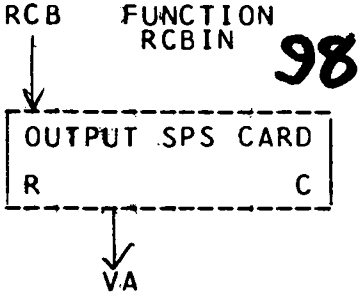

REW
FUNCTION REWIND
SET L=((R))
BSP
FUNCTION BACKSPACE
SET L=((B))
UNL
FUNCTION UNLOAD
SET L=((U))
WEFF
FUNCTION WREOF
SET L=((M))
ERAS
FUNCTION ERASE
SET L=((E))
TAPA
SAVE L IN S
SCLUB
1
2
ERR19
THE AC CONTAINS PZE X,WHERE X IS THE TYPE NUMBER OF THE SYMBOL, WWWW
IS IT TYPE 1, I.E. WRITE FILE
+
−
LGL 2
IS IT WITHOUT BLOCKING
+
−
TAPC+2
OUTPUT SPS CARDS
MN WWWW **+004
U (U0 L
VA
TAPC
OUTPUT SPS CARDS
MN WWWW+011 **+004
U (U0 L
VA
TAPB
TYPE 0 OR 2, I.E.READ OR COPY FILE
+
−
ERR32
IS L M OR E
−
+
ERR27



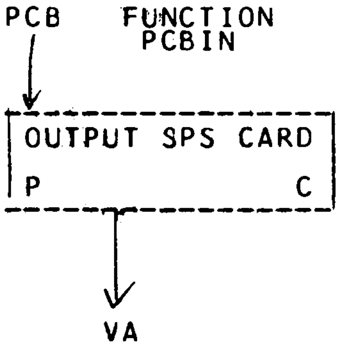
96



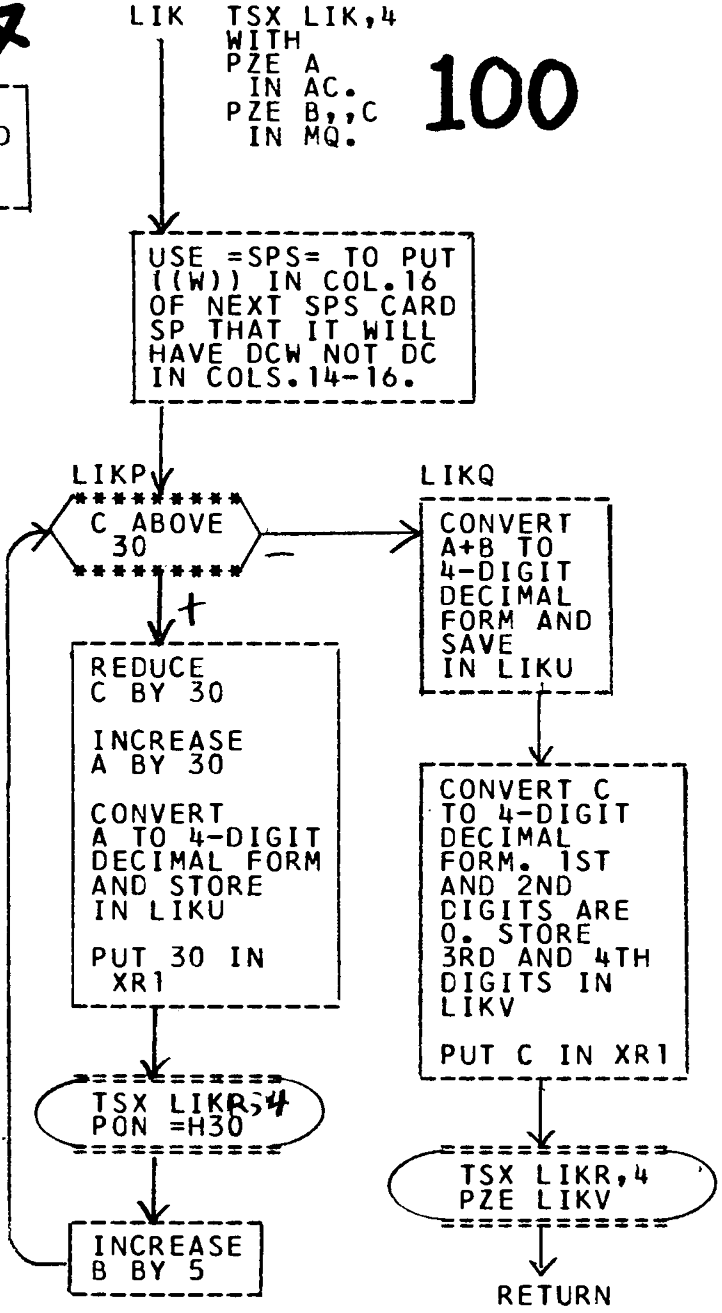
97



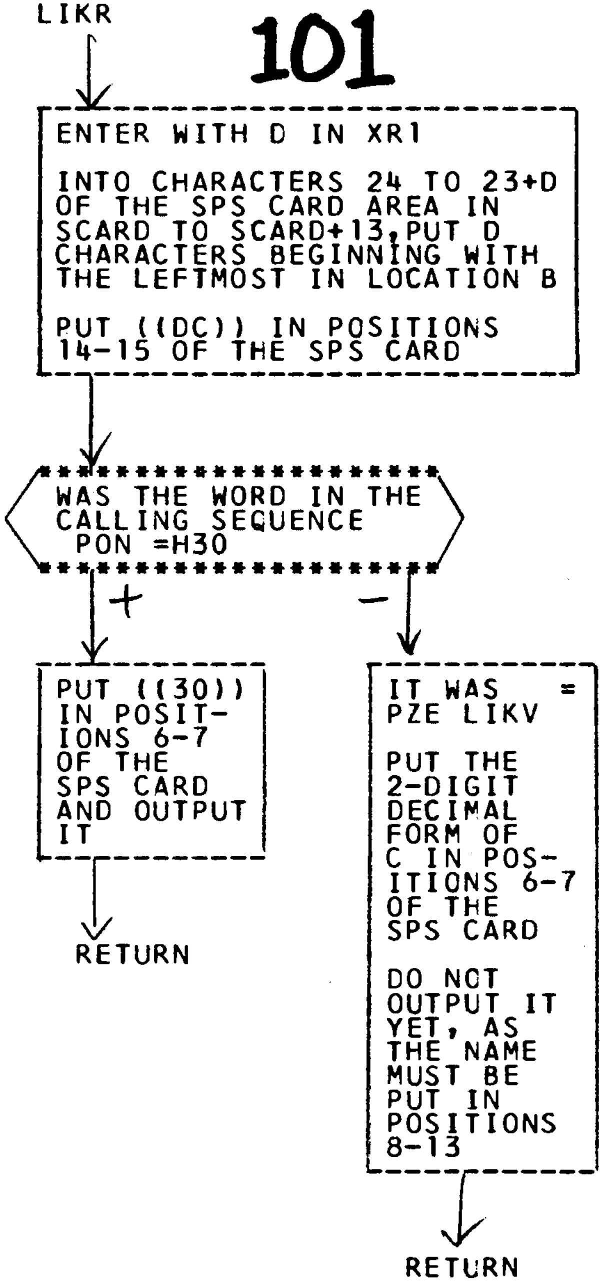
98



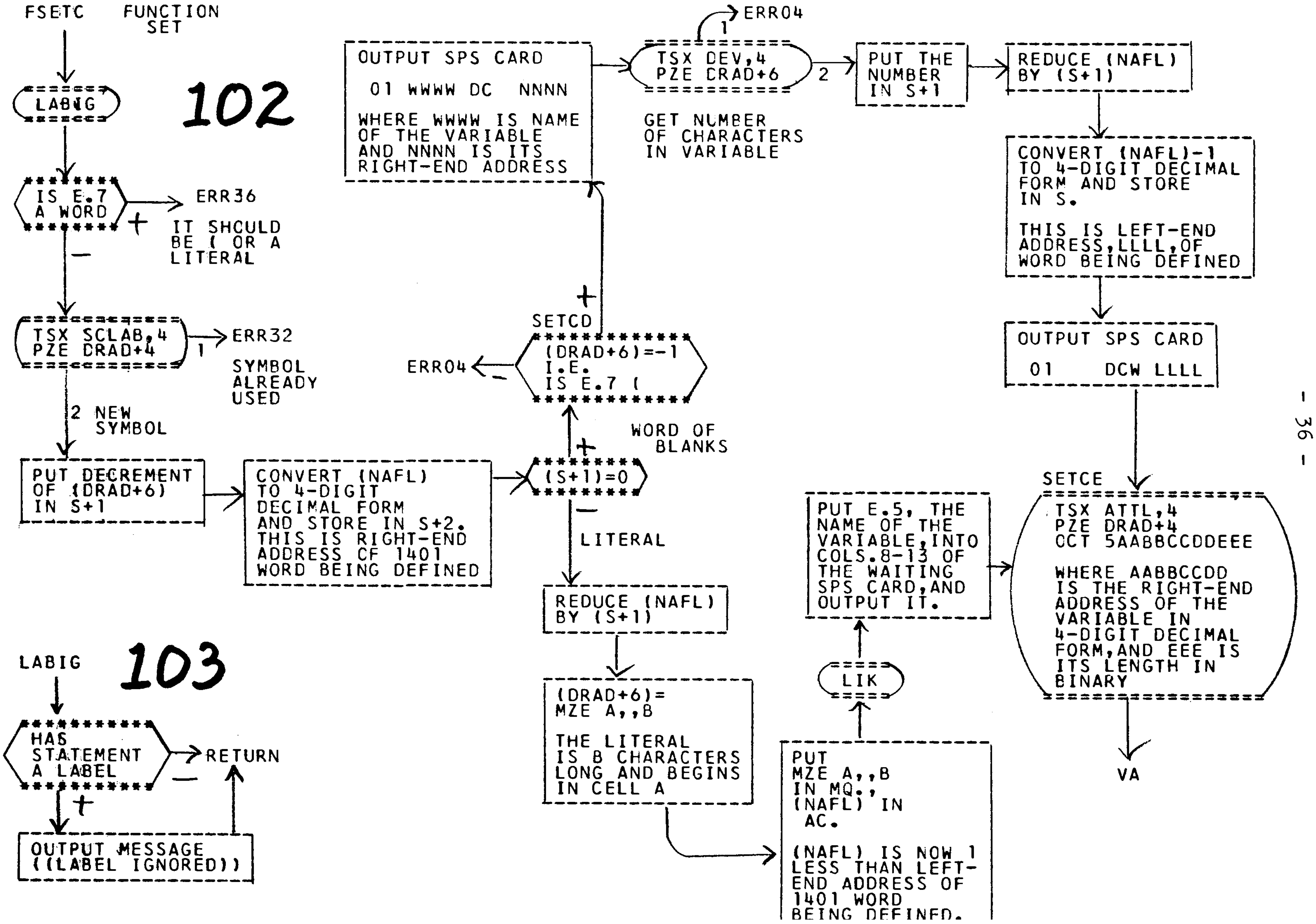
99

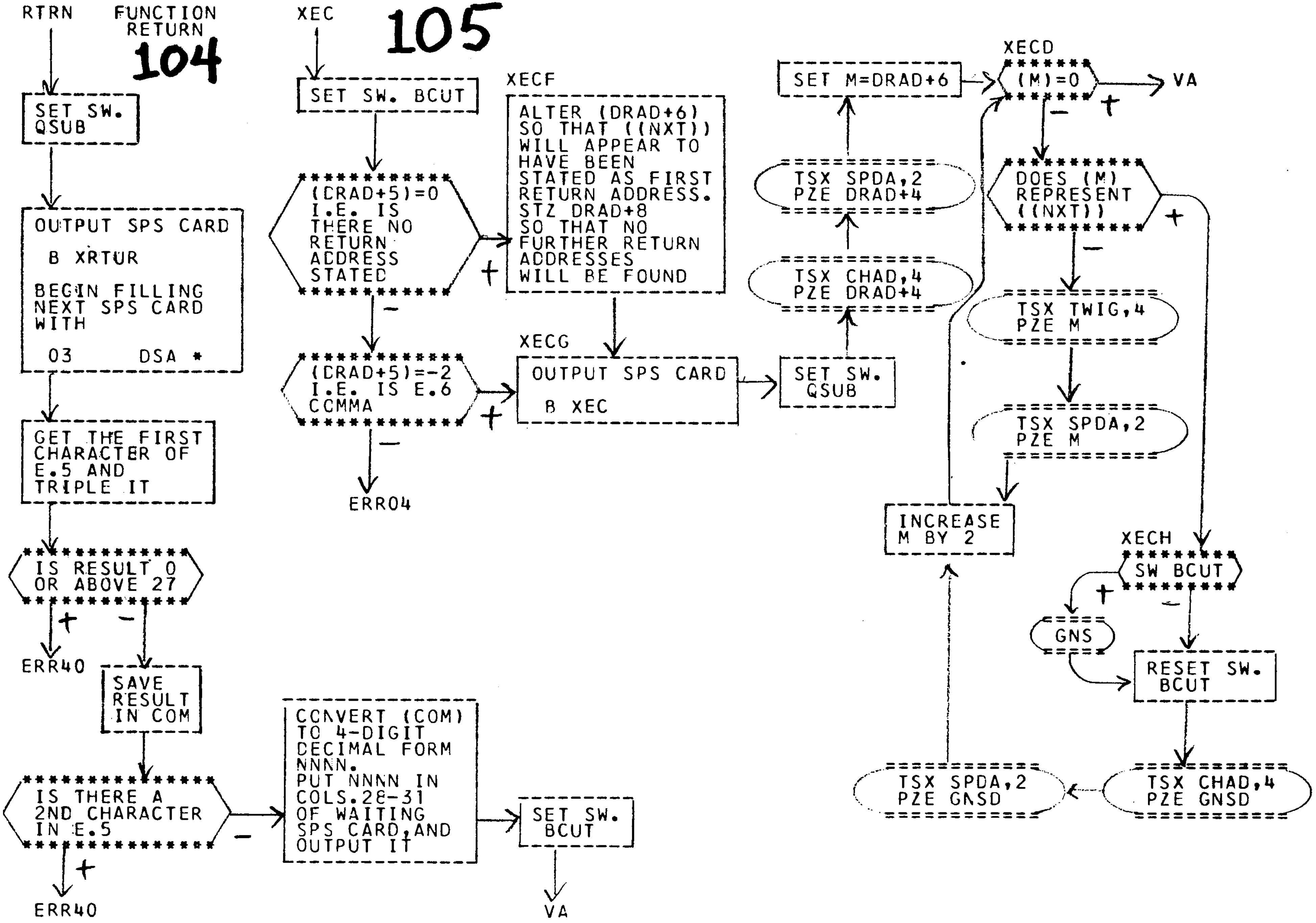

LIK
TSX LIK,4 WITH PZE A IN AC. PZE B,,C IN MQ.
USE =SPS= TO PUT ((W)) IN COL.16 OF NEXT SPS CARD SP THAT IT WILL HAVE DCW NOT DC IN COLS.14-16.
LIKP
C ABOVE 30
+
-
REDUCE C BY 30
INCREASE A BY 30
CONVERT A TO 4-DIGIT DECIMAL FORM AND STORE IN LIKU
PUT 30 IN XR1
TSX LIKR,4
PON =H30
INCREASE B BY 5
LIKQ
CONVERT A+B TO 4-DIGIT DECIMAL FORM AND SAVE IN LIKU
CONVERT C TO 4-DIGIT DECIMAL FORM. 1ST AND 2ND DIGITS ARE 0. STORE 3RD AND 4TH DIGITS IN LIKV
PUT C IN XR1
TSX LIKR,4
PZE LIKV
RETURN

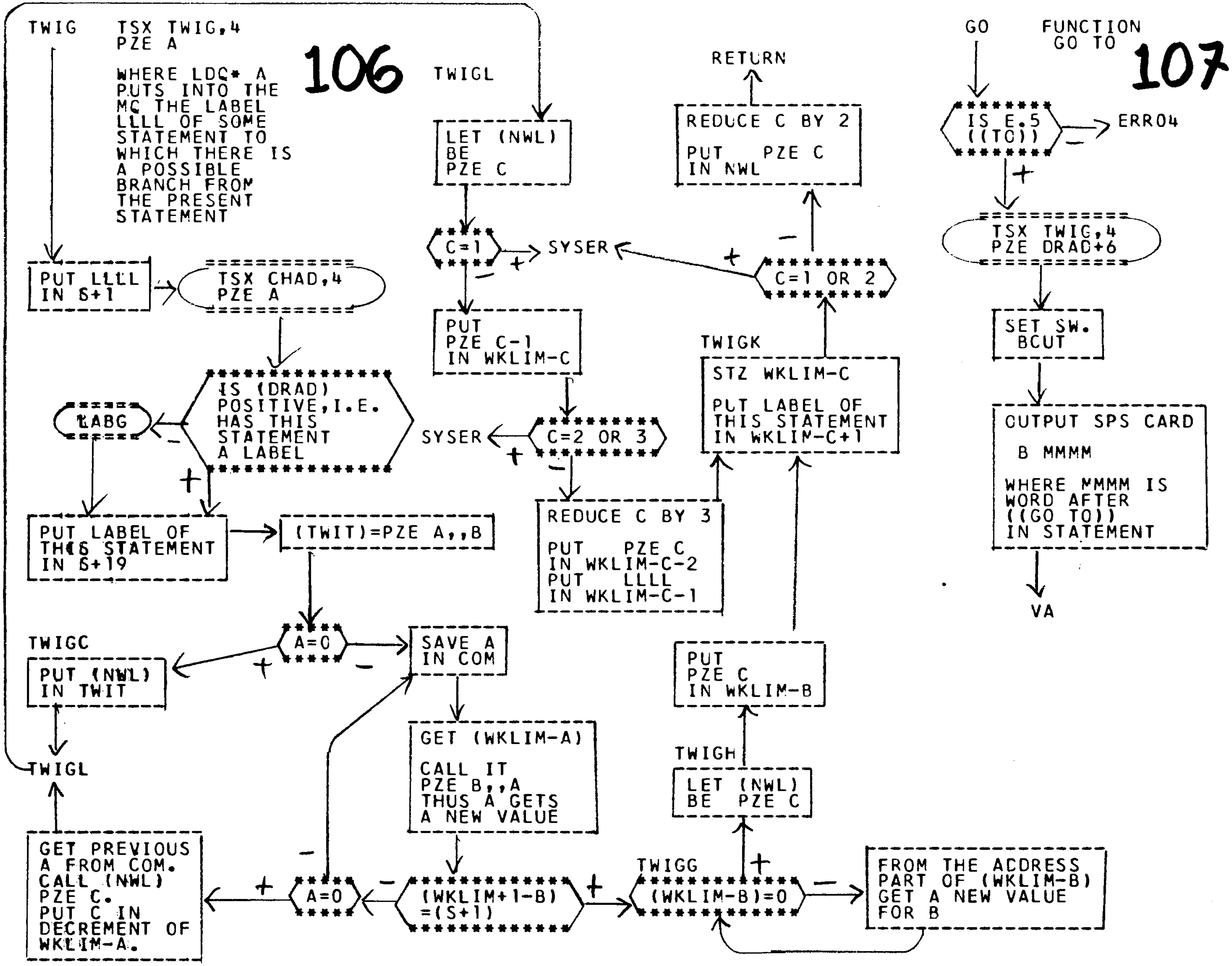
100

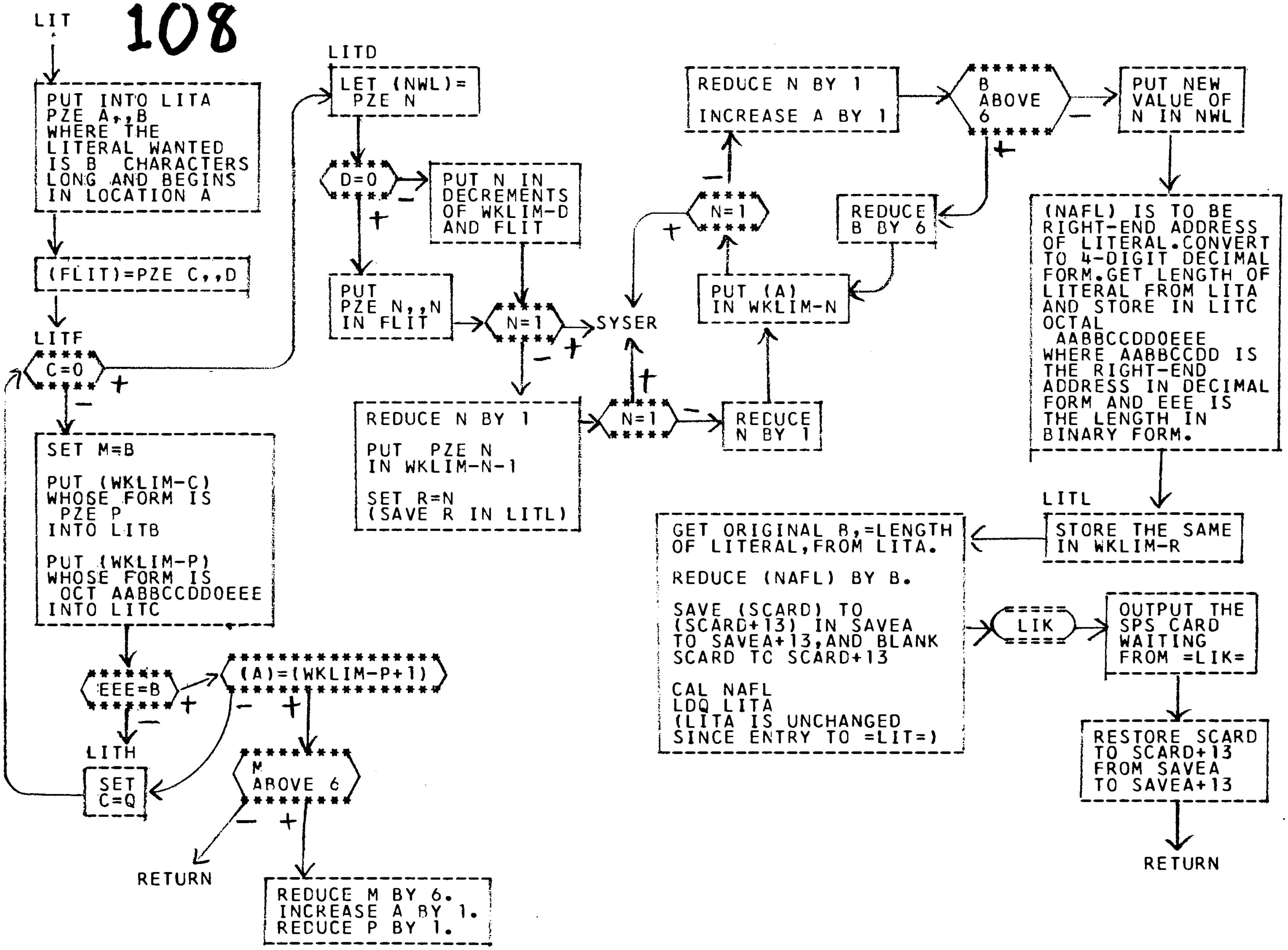

LIKR
ENTER WITH D IN XR1
INTO CHARACTERS 24 TO 23+D OF THE SPS CARD AREA IN SCARD TO SCARD+13,PUT D CHARACTERS BEGINNING WITH THE LEFTMOST IN LOCATION B
PUT ((DC)) IN POSITIONS 14-15 OF THE SPS CARD
WAS THE WORD IN THE CALLING SEQUENCE PON =H30
+
-
PUT ((30)) IN POSITIONS 6-7 OF THE SPS CARD AND OUTPUT IT
RETURN
IT WAS = PZE LIKV
PUT THE 2-DIGIT DECIMAL FORM OF C IN POSITIONS 6-7 OF THE SPS CARD
DO NOT OUTPUT IT YET, AS THE NAME MUST BE PUT IN POSITIONS 8-13
RETURN

101

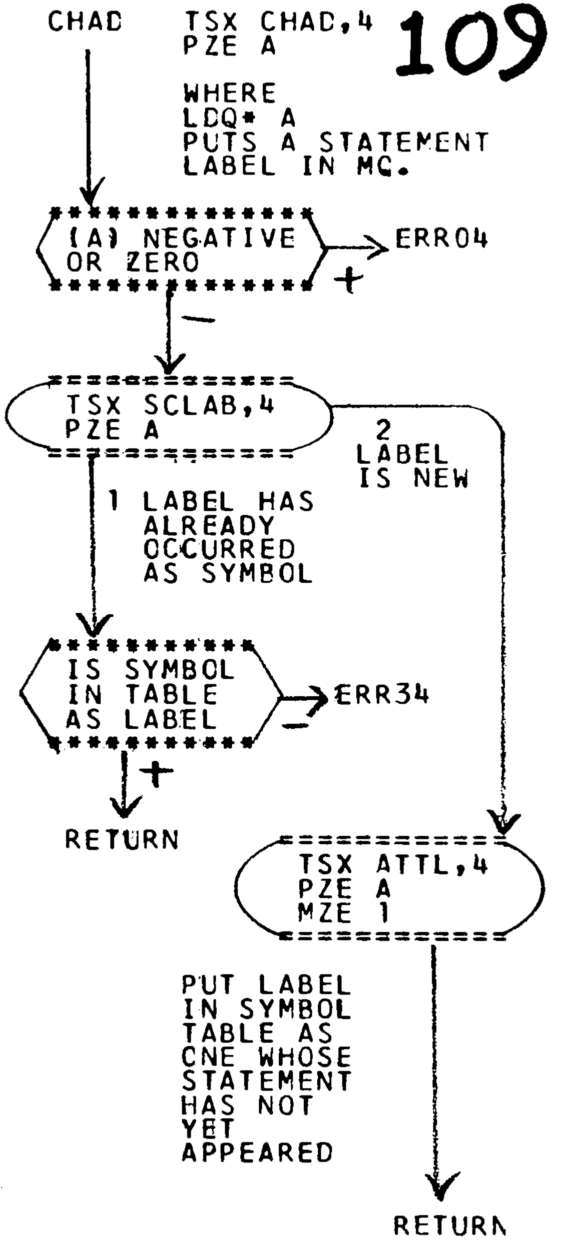

FSETC FUNCTION SET
102
LABIG
IS E.7 A WORD
+
ERR36
IT SHOULD BE (OR A LITERAL
−
TSX SCLAB,4
PZE DRAD+4
1
ERR32
SYMBOL ALREADY USED
2 NEW SYMBOL
PUT DECREMENT OF (DRAD+6) IN S+1
CONVERT (NAFL) TO 4-DIGIT DECIMAL FORM AND STORE IN S+2. THIS IS RIGHT-END ADDRESS CF 1401 WORD BEING DEFINED
(S+1)=0
+
WORD OF BLANKS
−
LITERAL
SETCD
(DRAD+6)=-1 I.E. IS E.7 (
ERR04
−
+
OUTPUT SPS CARD
01 WWWW DC NNNN
WHERE WWWW IS NAME OF THE VARIABLE AND NNNN IS ITS RIGHT-END ADDRESS
TSX DEV,4
PZE CRAD+6
1
ERR04
2
GET NUMBER OF CHARACTERS IN VARIABLE
PUT THE NUMBER IN S+1
REDUCE (NAFL) BY (S+1)
CONVERT (NAFL)-1 TO 4-DIGIT DECIMAL FORM AND STORE IN S.
THIS IS LEFT-END ADDRESS,LLLL,OF WORD BEING DEFINED
OUTPUT SPS CARD
01 DCW LLLL
REDUCE (NAFL) BY (S+1)
(DRAD+6)= MZE A,,B
THE LITERAL IS B CHARACTERS LONG AND BEGINS IN CELL A
PUT MZE A,,B IN MQ., (NAFL) IN AC.
(NAFL) IS NOW 1 LESS THAN LEFT-END ADDRESS OF 1401 WORD BEING DEFINED.
LIK
PUT E.5, THE NAME OF THE VARIABLE,INTO COLS.8-13 OF THE WAITING SPS CARD,AND OUTPUT IT.
SETCE
TSX ATTL,4
PZE DRAD+4
OCT 5AABBCCDDEEE
WHERE AABBCCDD IS THE RIGHT-END ADDRESS OF THE VARIABLE IN 4-DIGIT DECIMAL FORM,AND EEE IS ITS LENGTH IN BINARY
VA
LABIG
103
HAS STATEMENT A LABEL
−
RETURN
+
OUTPUT MESSAGE ((LABEL IGNORED))


RTRN
FUNCTION RETURN
104
SET SW. QSUB
OUTPUT SPS CARD
B XRTUR
BEGIN FILLING NEXT SPS CARD WITH
03 DSA *
GET THE FIRST CHARACTER OF E.5 AND TRIPLE IT
IS RESULT 0 OR ABOVE 27
+
-
ERR40
SAVE RESULT IN COM
IS THERE A 2ND CHARACTER IN E.5
+
-
ERR40
CONVERT (COM) TO 4-DIGIT DECIMAL FORM NNNN. PUT NNNN IN COLS.28-31 OF WAITING SPS CARD,AND OUTPUT IT
SET SW. BCUT
VA
XEC
105
SET SW. BCUT
(DRAD+5)=0 I.E. IS THERE NO RETURN ADDRESS STATED
+
-
XECF
ALTER (DRAD+6) SO THAT ((NXT)) WILL APPEAR TO HAVE BEEN STATED AS FIRST RETURN ADDRESS. STZ DRAD+8 SO THAT NO FURTHER RETURN ADDRESSES WILL BE FOUND
(DRAD+5)=-2 I.E. IS E.6 COMMA
+
-
ERR04
XECG
OUTPUT SPS CARD
B XEC
SET SW. QSUB
TSX CHAD,4 PZE DRAD+4
TSX SPDA,2 PZE DRAD+4
SET M=DRAD+6
XECD
(M)=0
+
-
VA
DOES (M) REPRESENT ((NXT))
+
-
TSX TWIG,4 PZE M
TSX SPDA,2 PZE M
INCREASE M BY 2
XECH
SW BCUT
+
-
GNS
RESET SW. BCUT
TSX CHAD,4 PZE GNSD
TSX SPDA,2 PZE GNSD

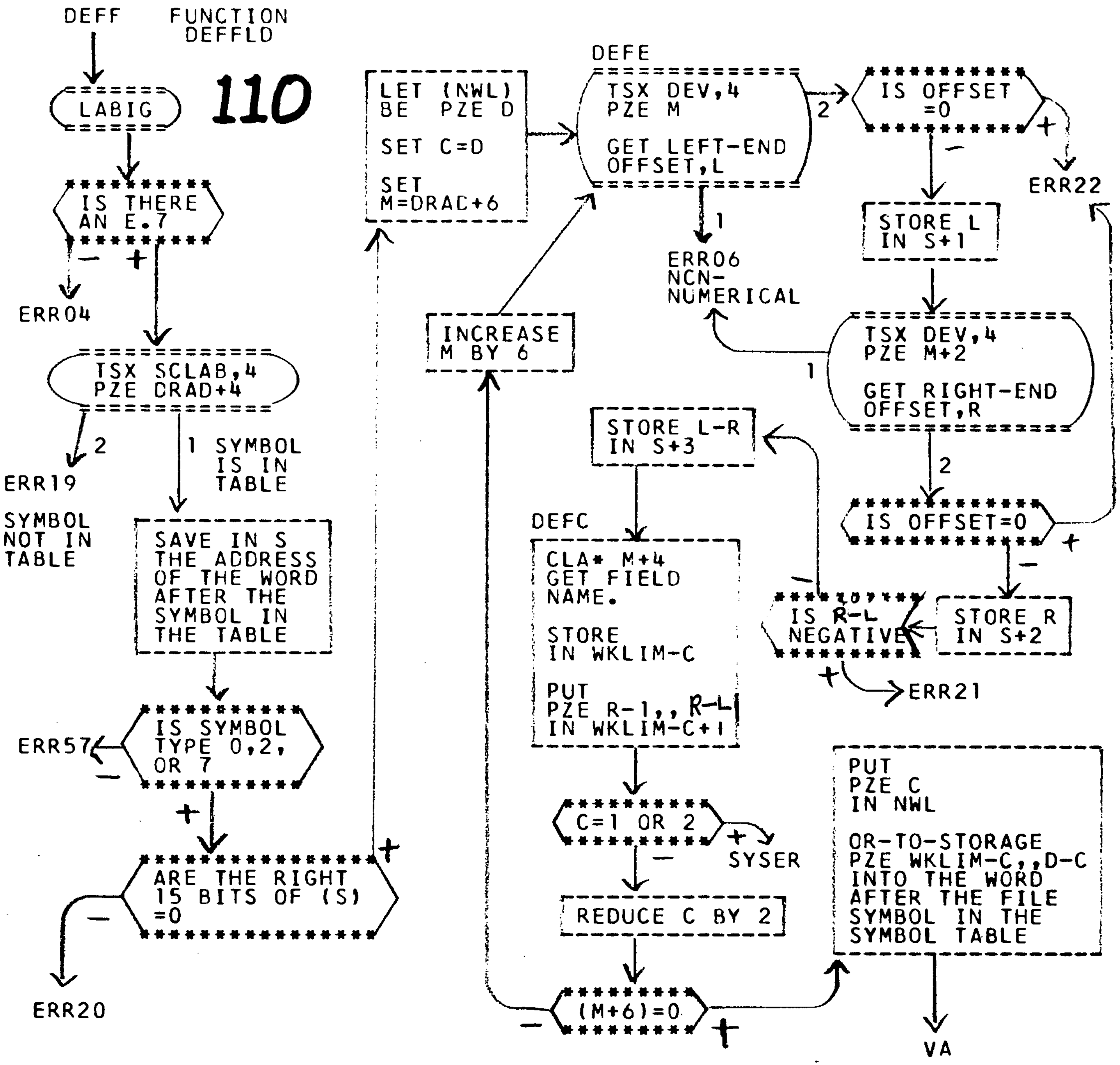

TWIG
TSX TWIG,4
PZE A
WHERE LDQ* A PUTS INTO THE MQ THE LABEL LLLL OF SOME STATEMENT TO WHICH THERE IS A POSSIBLE BRANCH FROM THE PRESENT STATEMENT
106
PUT LLLL IN S+1
TSX CHAD,4
PZE A
IS (DRAD) POSITIVE,I.E. HAS THIS STATEMENT A LABEL
LABG
-
+
PUT LABEL OF THIS STATEMENT IN S+19
(TWIT)=PZE A,,B
A=0
+
-
TWIGC
PUT (NWL) IN TWIT
SAVE A IN COM
GET (WKLIM-A)
CALL IT PZE B,,A THUS A GETS A NEW VALUE
TWIGL
GET PREVIOUS A FROM COM. CALL (NWL) PZE C. PUT C IN DECREMENT OF WKLIM-A.
A=0
+
-
(WKLIM+1-B) =(S+1)
-
+
TWIGG
(WKLIM-B)=0
+
-
FROM THE ADDRESS PART OF (WKLIM-B) GET A NEW VALUE FOR B
TWIGL
LET (NWL) BE PZE C
C=1
+
-
SYSER
PUT PZE C-1 IN WKLIM-C
C=2 OR 3
SYSER
+
-
REDUCE C BY 3
PUT PZE C IN WKLIM-C-2
PUT LLLL IN WKLIM-C-1
RETURN
REDUCE C BY 2
PUT PZE C IN NWL
-
+
C=1 OR 2
TWIGK
STZ WKLIM-C
PUT LABEL OF THIS STATEMENT IN WKLIM-C+1
PUT PZE C IN WKLIM-B
TWIGH
LET (NWL) BE PZE C
GO
FUNCTION GO TO
107
IS E.5 ((TO))
-
ERR04
+
TSX TWIG,4
PZE DRAD+6
SET SW. BCUT
OUTPUT SPS CARD
B MMMM
WHERE MMMM IS WORD AFTER ((GO TO)) IN STATEMENT
VA

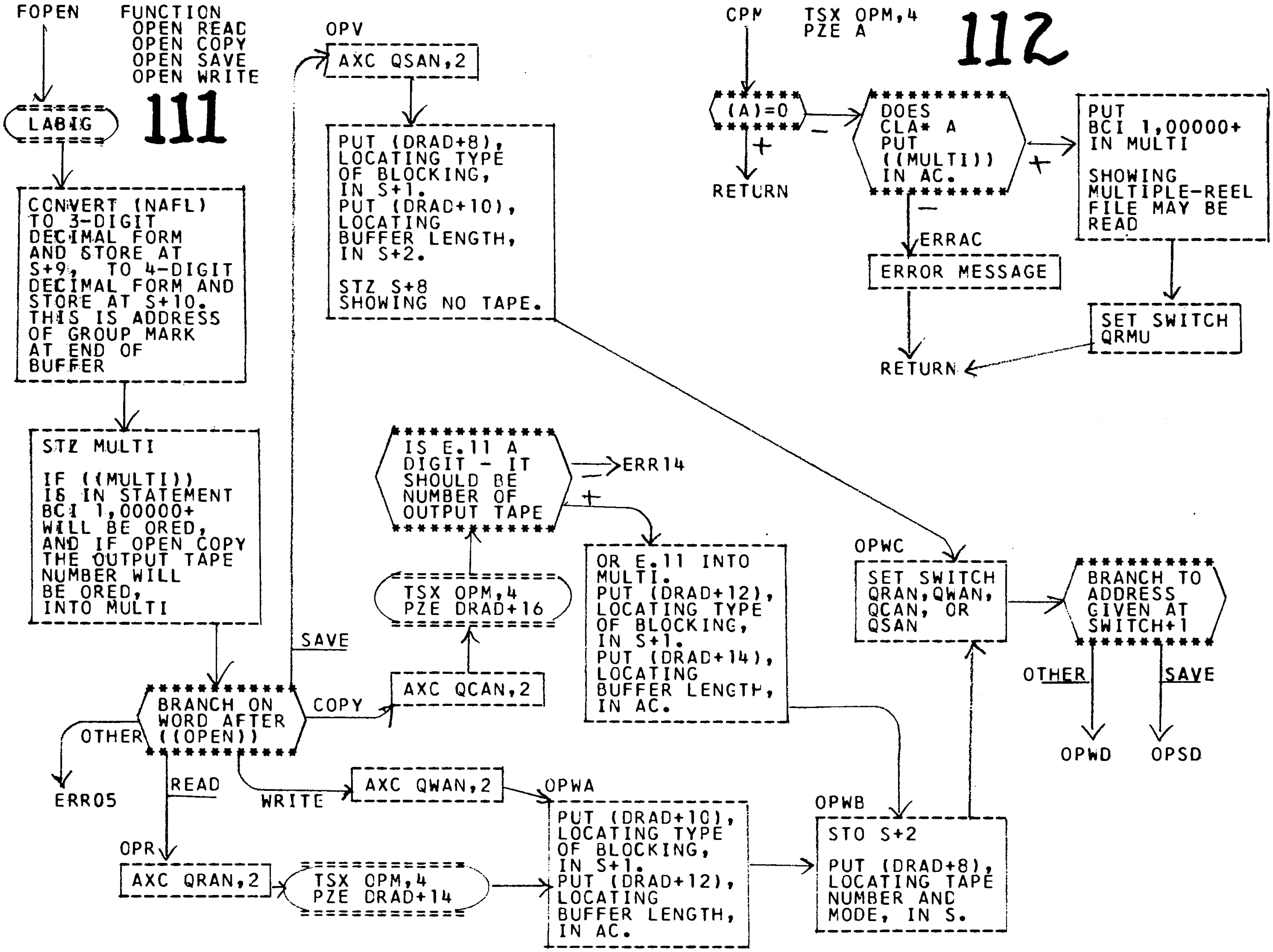

LIT
PUT INTO LITA PZE A,,B WHERE THE LITERAL WANTED IS B CHARACTERS LONG AND BEGINS IN LOCATION A
(FLIT)=PZE C,,D
LITF
C=0
+
−
SET M=B
PUT (WKLIM-C) WHOSE FORM IS PZE P INTO LITB
PUT (WKLIM-P) WHOSE FORM IS OCT AABBCCDD0EEE INTO LITC
EEE=B
+
−
LITH
SET C=Q
(A)=(WKLIM-P+1)
−
+
M ABOVE 6
−
+
RETURN
REDUCE M BY 6. INCREASE A BY 1. REDUCE P BY 1.
LITD
LET (NWL)= PZE N
D=0
−
+
PUT N IN DECREMENTS OF WKLIM-D AND FLIT
PUT PZE N,,N IN FLIT
N=1
+
−
SYSER
REDUCE N BY 1
PUT PZE N IN WKLIM-N-1
SET R=N (SAVE R IN LITL)
N=1
+
−
REDUCE N BY 1
PUT (A) IN WKLIM-N
N=1
+
−
REDUCE N BY 1
INCREASE A BY 1
B ABOVE 6
−
+
REDUCE B BY 6
PUT NEW VALUE OF N IN NWL
(NAFL) IS TO BE RIGHT-END ADDRESS OF LITERAL.CONVERT TO 4-DIGIT DECIMAL FORM.GET LENGTH OF LITERAL FROM LITA AND STORE IN LITC OCTAL AABBCCDD0EEE WHERE AABBCCDD IS THE RIGHT-END ADDRESS IN DECIMAL FORM AND EEE IS THE LENGTH IN BINARY FORM.
LITL
STORE THE SAME IN WKLIM-R
GET ORIGINAL B,=LENGTH OF LITERAL,FROM LITA.
REDUCE (NAFL) BY B.
SAVE (SCARD) TO (SCARD+13) IN SAVEA TO SAVEA+13,AND BLANK SCARD TO SCARD+13
CAL NAFL
LDQ LITA
(LITA IS UNCHANGED SINCE ENTRY TO =LIT=)
LIK
OUTPUT THE SPS CARD WAITING FROM =LIK=
RESTORE SCARD TO SCARD+13 FROM SAVEA TO SAVEA+13
RETURN

109

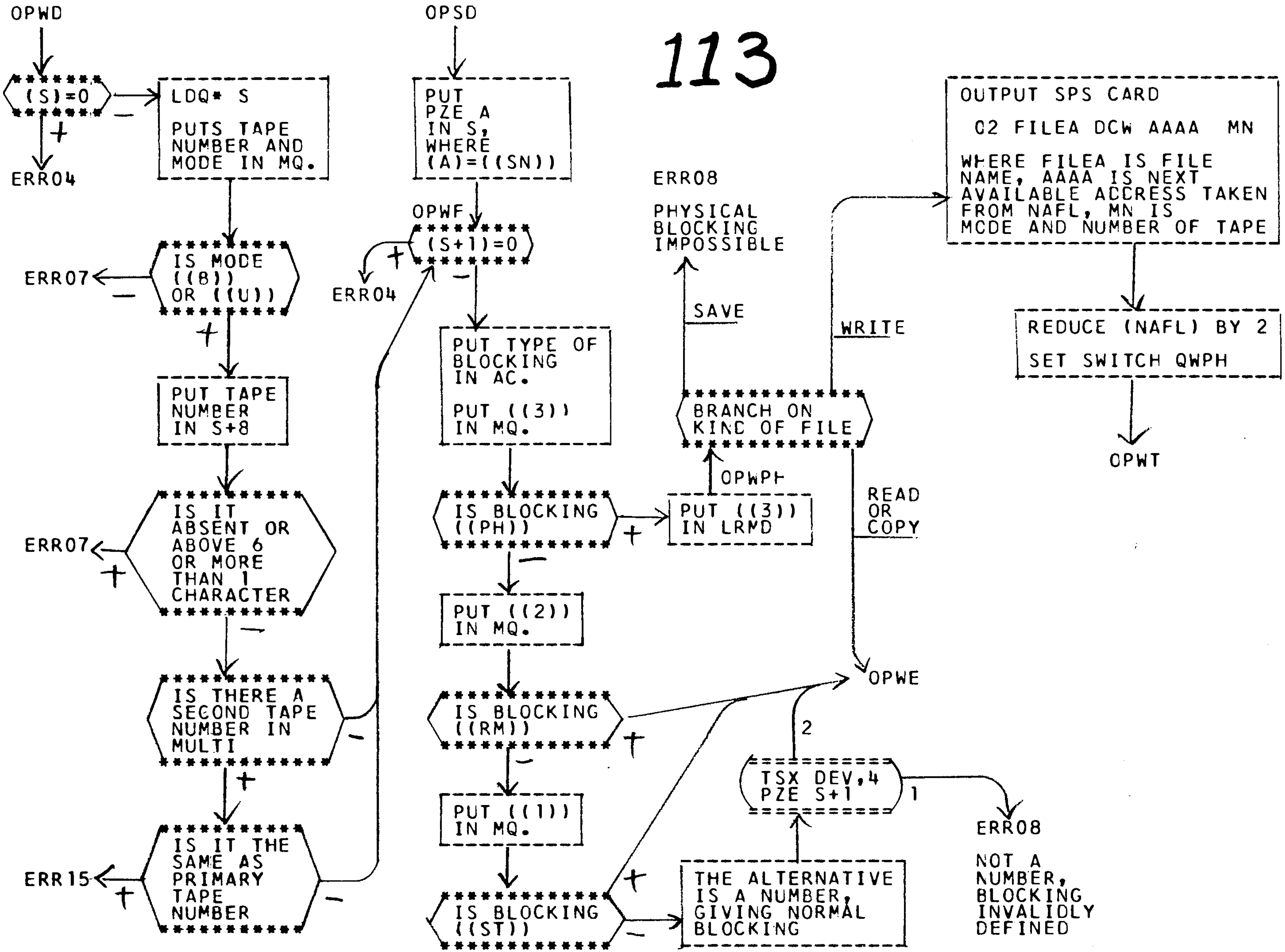

CHAD
TSX CHAD,4
PZE A
WHERE
LDQ* A
PUTS A STATEMENT
LABEL IN MQ.
(A) NEGATIVE OR ZERO
+
ERR04
−
TSX SCLAB,4
PZE A
2 LABEL IS NEW
1 LABEL HAS ALREADY OCCURRED AS SYMBOL
IS SYMBOL IN TABLE AS LABEL
−
ERR34
+
RETURN
TSX ATTL,4
PZE A
MZE 1
PUT LABEL IN SYMBOL TABLE AS ONE WHOSE STATEMENT HAS NOT YET APPEARED
RETURN

110

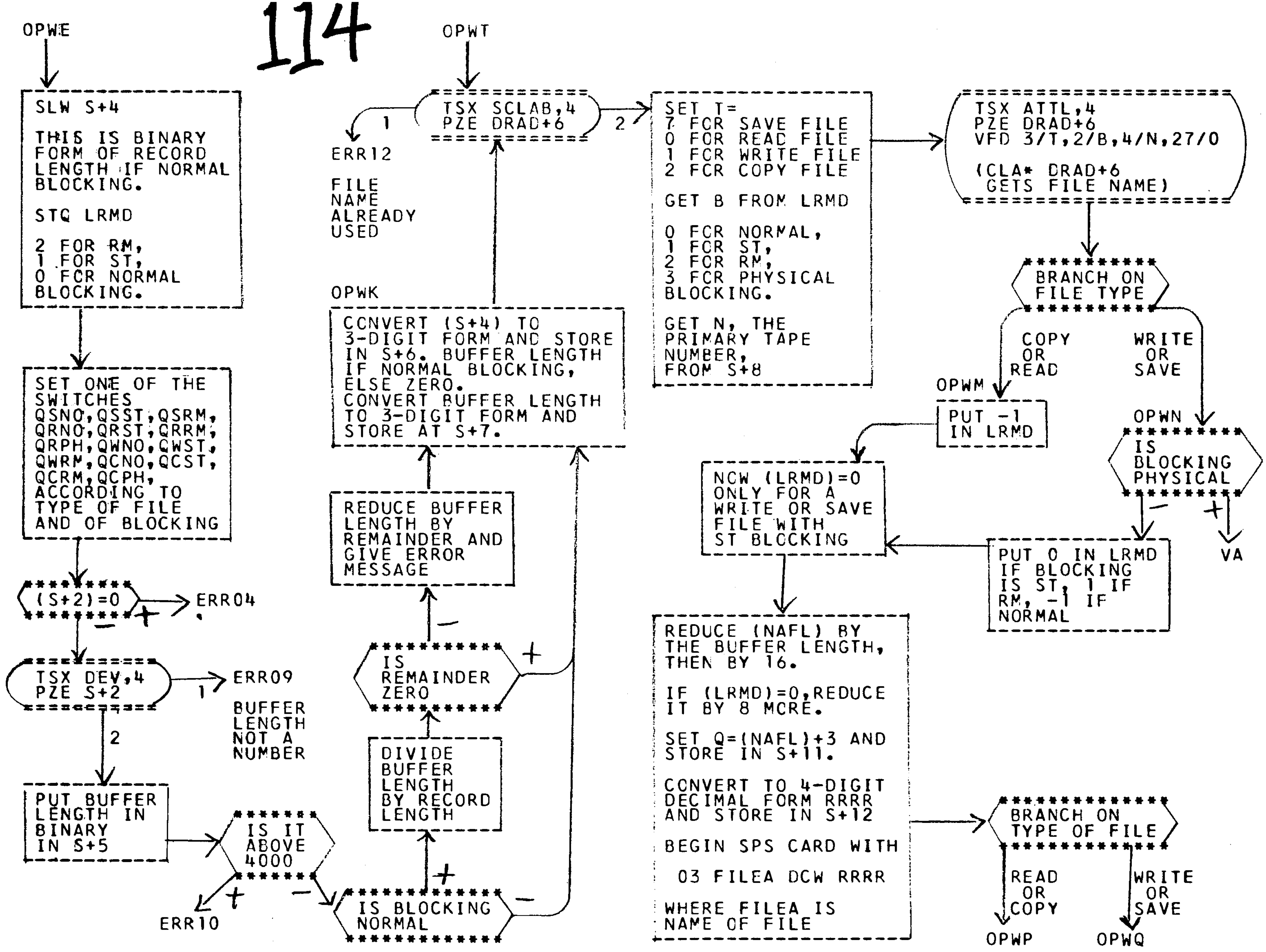

DEFF
FUNCTION DEFFLD
LABIG
IS THERE AN E.7
−
+
ERR04
TSX SCLAB,4
PZE DRAD+4
2
ERR19
SYMBOL NOT IN TABLE
1 SYMBOL IS IN TABLE
SAVE IN S THE ADDRESS OF THE WORD AFTER THE SYMBOL IN THE TABLE
IS SYMBOL TYPE 0,2, OR 7
−
ERR57
+
ARE THE RIGHT 15 BITS OF (S) =0
−
ERR20
+
LET (NWL) BE PZE D
SET C=D
SET M=DRAD+6
DEFE
TSX DEV,4
PZE M
GET LEFT-END OFFSET,L
2
1
ERR06 NON-NUMERICAL
IS OFFSET =0
+
ERR22
−
STORE L IN S+1
TSX DEV,4
PZE M+2
GET RIGHT-END OFFSET,R
1
2
IS OFFSET=0
+
−
STORE R IN S+2
IS R-L NEGATIVE
−
+
ERR21
STORE L-R IN S+3
DEFC
CLA* M+4
GET FIELD NAME.
STORE IN WKLIM-C
PUT
PZE R-1,,R-L
IN WKLIM-C+1
C=1 OR 2
+
SYSER
−
REDUCE C BY 2
(M+6)=0
−
+
INCREASE M BY 6
PUT
PZE C
IN NWL
OR-TO-STORAGE
PZE WKLIM-C,,D-C
INTO THE WORD AFTER THE FILE SYMBOL IN THE SYMBOL TABLE
VA

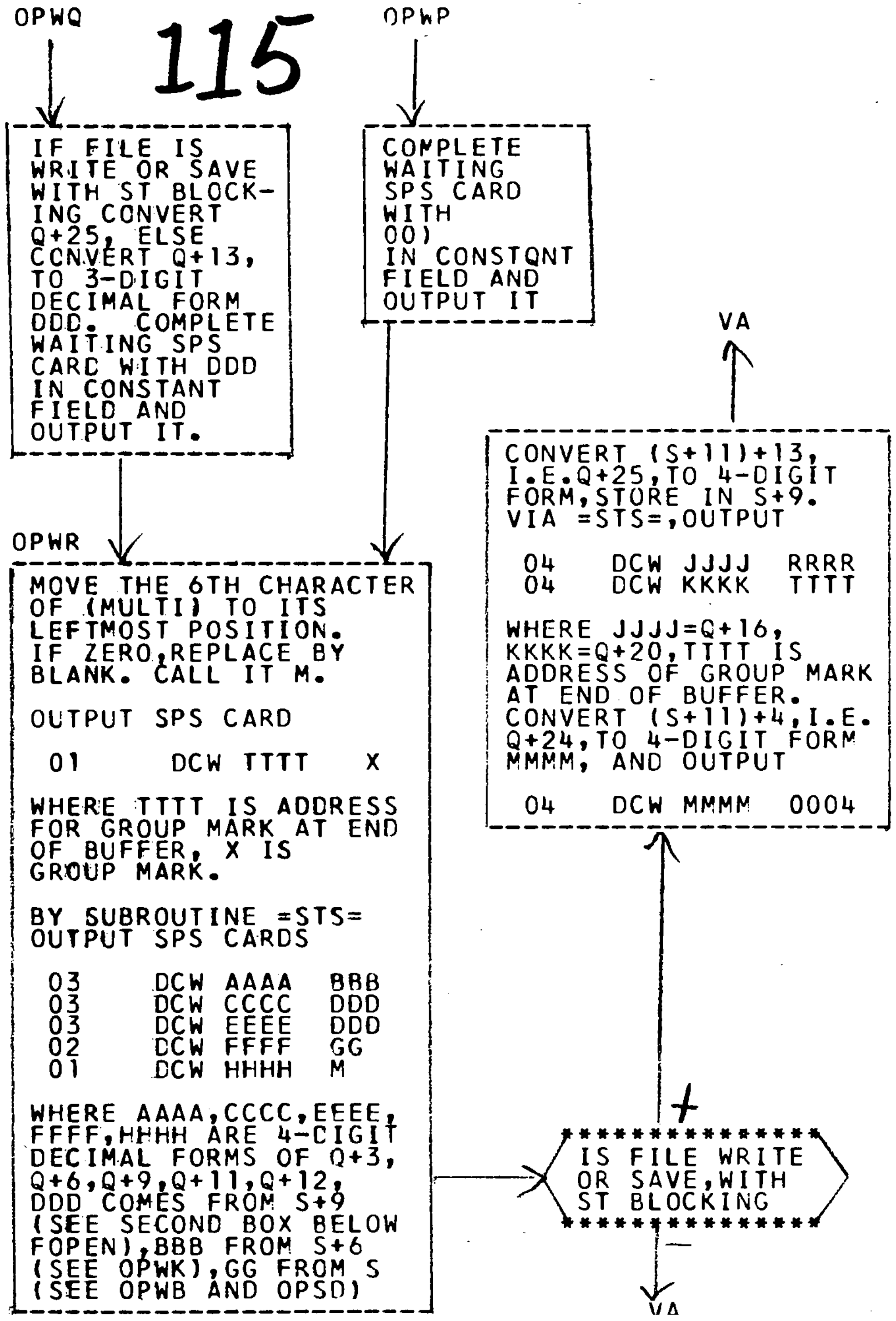

FOPEN
FUNCTION
OPEN READ
OPEN COPY
OPEN SAVE
OPEN WRITE
111
LABIG
CONVERT (NAFL) TO 3-DIGIT DECIMAL FORM AND STORE AT S+9, TO 4-DIGIT DECIMAL FORM AND STORE AT S+10. THIS IS ADDRESS OF GROUP MARK AT END OF BUFFER
STZ MULTI
IF ((MULTI)) IS IN STATEMENT BCI 1,00000+ WILL BE ORED, AND IF OPEN COPY THE OUTPUT TAPE NUMBER WILL BE ORED, INTO MULTI
BRANCH ON WORD AFTER ((OPEN))
OTHER
ERR05
READ
OPR
AXC QRAN,2
TSX OPM,4
PZE DRAD+14
WRITE
AXC QWAN,2
COPY
AXC QCAN,2
TSX OPM,4
PZE DRAD+16
IS E.11 A DIGIT - IT SHOULD BE NUMBER OF OUTPUT TAPE
—
ERR14
+
OR E.11 INTO MULTI. PUT (DRAD+12), LOCATING TYPE OF BLOCKING, IN S+1. PUT (DRAD+14), LOCATING BUFFER LENGTH, IN AC.
SAVE
OPV
AXC QSAN,2
PUT (DRAD+8), LOCATING TYPE OF BLOCKING, IN S+1. PUT (DRAD+10), LOCATING BUFFER LENGTH, IN S+2.
STZ S+8 SHOWING NO TAPE.
OPWA
PUT (DRAD+10), LOCATING TYPE OF BLOCKING, IN S+1. PUT (DRAD+12), LOCATING BUFFER LENGTH, IN AC.
OPWB
STO S+2
PUT (DRAD+8), LOCATING TAPE NUMBER AND MODE, IN S.
OPWC
SET SWITCH QRAN,QWAN, QCAN, OR QSAN
BRANCH TO ADDRESS GIVEN AT SWITCH+1
OTHER
SAVE
OPWD
OPSD
CPM
TSX OPM,4
PZE A
112
(A)=0
+
RETURN
—
DOES CLA* A PUT ((MULTI)) IN AC.
—
ERRAC
ERROR MESSAGE
RETURN
+
PUT BCI 1,00000+ IN MULTI
SHOWING MULTIPLE-REEL FILE MAY BE READ
SET SWITCH QRMU

113


OPWD
(S)=0
+
−
ERR04
LDQ* S
PUTS TAPE NUMBER AND MODE IN MQ.
IS MODE ((B)) OR ((U))
−
ERR07
+
PUT TAPE NUMBER IN S+8
IS IT ABSENT OR ABOVE 6 OR MORE THAN 1 CHARACTER
+
ERR07
−
IS THERE A SECOND TAPE NUMBER IN MULTI
−
+
IS IT THE SAME AS PRIMARY TAPE NUMBER
+
ERR15
−
OPSD
PUT PZE A IN S, WHERE (A)=((SN))
OPWF
(S+1)=0
+
ERR04
−
PUT TYPE OF BLOCKING IN AC.
PUT ((3)) IN MQ.
IS BLOCKING ((PH))
+
PUT ((3)) IN LRMD
OPWPH
−
PUT ((2)) IN MQ.
IS BLOCKING ((RM))
+
−
PUT ((1)) IN MQ.
IS BLOCKING ((ST))
+
−
THE ALTERNATIVE IS A NUMBER, GIVING NORMAL BLOCKING
TSX DEV,4
PZE S+1
1
ERR08
NOT A NUMBER, BLOCKING INVALIDLY DEFINED
2
OPWE
BRANCH ON KIND OF FILE
SAVE
ERR08
PHYSICAL BLOCKING IMPOSSIBLE
WRITE
READ OR COPY
OUTPUT SPS CARD
02 FILEA DCW AAAA MN
WHERE FILEA IS FILE NAME, AAAA IS NEXT AVAILABLE ADCRESS TAKEN FROM NAFL, MN IS MODE AND NUMBER OF TAPE
REDUCE (NAFL) BY 2
SET SWITCH QWPH
OPWT

114


OPWE
SLW S+4
THIS IS BINARY FORM OF RECORD LENGTH IF NORMAL BLOCKING.
STQ LRMD
2 FOR RM, 1 FOR ST, 0 FOR NORMAL BLOCKING.
SET ONE OF THE SWITCHES QSNO,QSST,QSRM, QRNO,QRST,QRRM, QRPH,QWNO,QWST, QWRM,QCNO,QCST, QCRM,QCPH, ACCORDING TO TYPE OF FILE AND OF BLOCKING
(S+2)=0
+
-
ERR04
TSX DEV,4
PZE S+2
1
ERR09
BUFFER LENGTH NOT A NUMBER
2
PUT BUFFER LENGTH IN BINARY IN S+5
IS IT ABOVE 4000
+
ERR10
-
IS BLOCKING NORMAL
+
-
DIVIDE BUFFER LENGTH BY RECORD LENGTH
IS REMAINDER ZERO
+
-
REDUCE BUFFER LENGTH BY REMAINDER AND GIVE ERROR MESSAGE
OPWK
CONVERT (S+4) TO 3-DIGIT FORM AND STORE IN S+6. BUFFER LENGTH IF NORMAL BLOCKING, ELSE ZERO. CONVERT BUFFER LENGTH TO 3-DIGIT FORM AND STORE AT S+7.
OPWT
TSX SCLAB,4
PZE DRAD+6
1
ERR12
FILE NAME ALREADY USED
2
SET T= 7 FOR SAVE FILE 0 FOR READ FILE 1 FOR WRITE FILE 2 FOR COPY FILE
GET B FROM LRMD
0 FOR NORMAL, 1 FOR ST, 2 FOR RM, 3 FOR PHYSICAL BLOCKING.
GET N, THE PRIMARY TAPE NUMBER, FROM S+8
TSX ATTL,4
PZE DRAD+6
VFD 3/T,2/B,4/N,27/0
(CLA* DRAD+6 GETS FILE NAME)
BRANCH ON FILE TYPE
COPY OR READ
WRITE OR SAVE
OPWM
PUT -1 IN LRMD
OPWN
IS BLOCKING PHYSICAL
-
+
VA
PUT 0 IN LRMD IF BLOCKING IS ST, 1 IF RM, -1 IF NORMAL
NOW (LRMD)=0 ONLY FOR A WRITE OR SAVE FILE WITH ST BLOCKING
REDUCE (NAFL) BY THE BUFFER LENGTH, THEN BY 16.
IF (LRMD)=0, REDUCE IT BY 8 MORE.
SET Q=(NAFL)+3 AND STORE IN S+11.
CONVERT TO 4-DIGIT DECIMAL FORM RRRR AND STORE IN S+12
BEGIN SPS CARD WITH
03 FILEA DCW RRRR
WHERE FILEA IS NAME OF FILE
BRANCH ON TYPE OF FILE
READ OR COPY
WRITE OR SAVE
OPWP
OPWQ



IF FILE IS WRITE OR SAVE WITH ST BLOCKING CONVERT Q+25, ELSE CONVERT Q+13, TO 3-DIGIT DECIMAL FORM DDD. COMPLETE WAITING SPS CARD WITH DDD IN CONSTANT FIELD AND OUTPUT IT.

COMPLETE WAITING SPS CARD WITH 00) IN CONSTQNT FIELD AND OUTPUT IT

MOVE THE 6TH CHARACTER OF (MULTI) TO ITS LEFTMOST POSITION. IF ZERO,REPLACE BY BLANK. CALL IT M.

OUTPUT SPS CARD

```
 01       DCW TTTT    X
```

WHERE TTTT IS ADDRESS FOR GROUP MARK AT END OF BUFFER, X IS GROUP MARK.

BY SUBROUTINE =STS= OUTPUT SPS CARDS

```
 03     DCW AAAA   BBB
 03     DCW CCCC   DDD
 03     DCW EEEE   DDD
 02     DCW FFFF   GG
 01     DCW HHHH   M
```

WHERE AAAA,CCCC,EEEE, FFFF,HHHH ARE 4-DIGIT DECIMAL FORMS OF Q+3, Q+6,Q+9,Q+12, DDD COMES FROM S+9 (SEE SECOND BOX BELOW FOPEN),BBB FROM S+6 (SEE OPWK),GG FROM S (SEE OPWB AND OPSD)

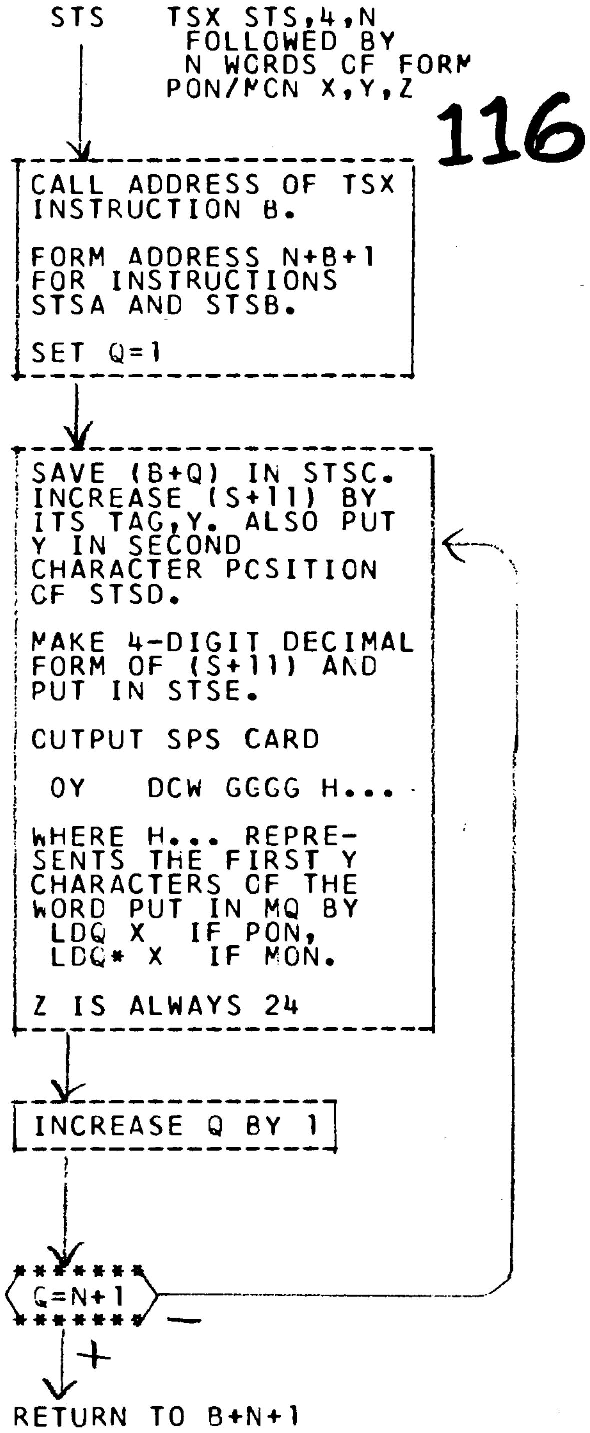
IS FILE WRITE OR SAVE,WITH ST BLOCKING

CONVERT (S+11)+13, I.E.Q+25,TO 4-DIGIT FORM,STORE IN S+9. VIA =STS=,OUTPUT

```
 04    DCW JJJJ   RRRR
 04    DCW KKKK   TTTT
```

WHERE JJJJ=Q+16, KKKK=Q+20,TTTT IS ADDRESS OF GROUP MARK AT END OF BUFFER. CONVERT (S+11)+4,I.E. Q+24,TO 4-DIGIT FORM MMMM, AND OUTPUT

```
 04    DCW MMMM   0004
```



TSX STS,4,N
FOLLOWED BY
N WORDS OF FORM
PON/MON X,Y,Z

CALL ADDRESS OF TSX INSTRUCTION B.

FORM ADDRESS N+B+1 FOR INSTRUCTIONS STSA AND STSB.

SET Q=1

SAVE (B+Q) IN STSC. INCREASE (S+11) BY ITS TAG,Y. ALSO PUT Y IN SECOND CHARACTER POSITION OF STSD.

MAKE 4-DIGIT DECIMAL FORM OF (S+11) AND PUT IN STSE.

OUTPUT SPS CARD

```
 0Y     DCW GGGG H...
```

WHERE H... REPRESENTS THE FIRST Y CHARACTERS OF THE WORD PUT IN MQ BY
 LDQ X IF PON,
 LDQ* X IF MON.

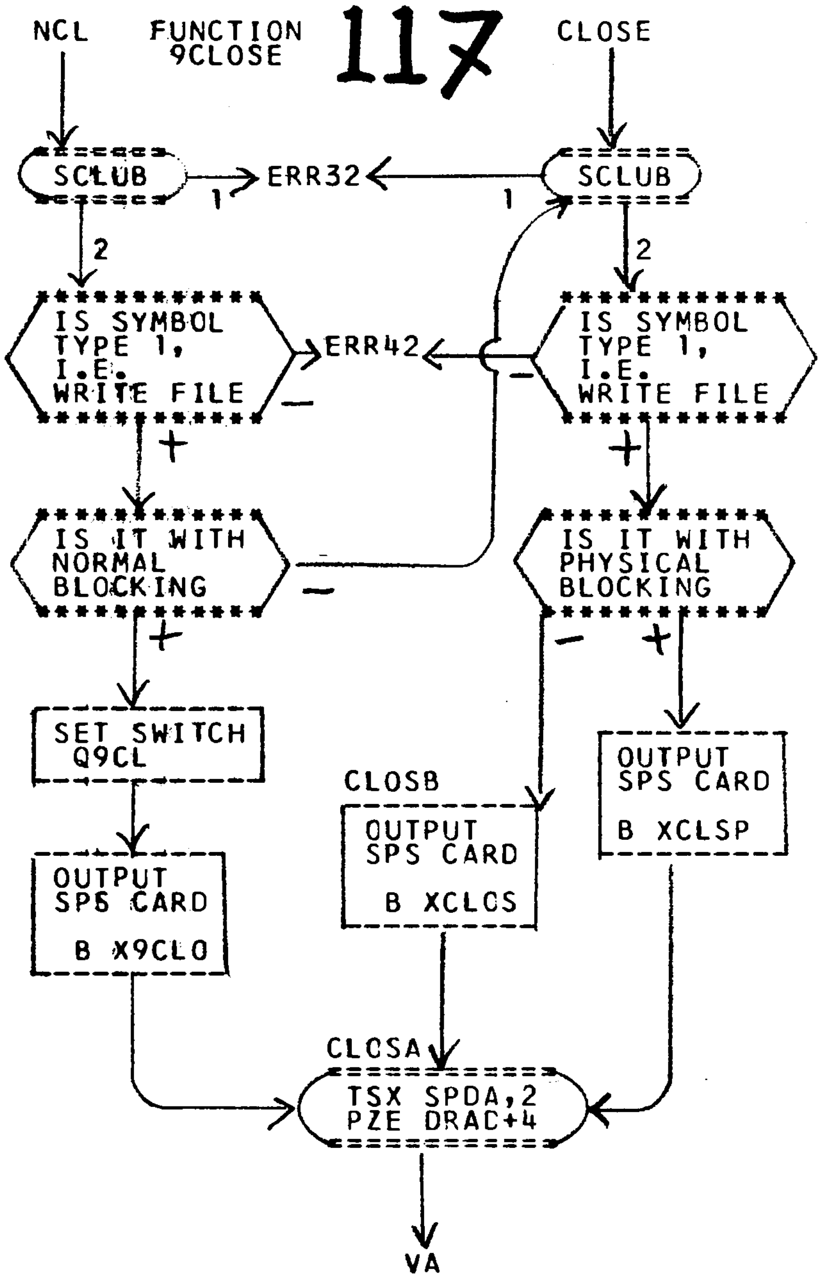
Z IS ALWAYS 24

INCREASE Q BY 1

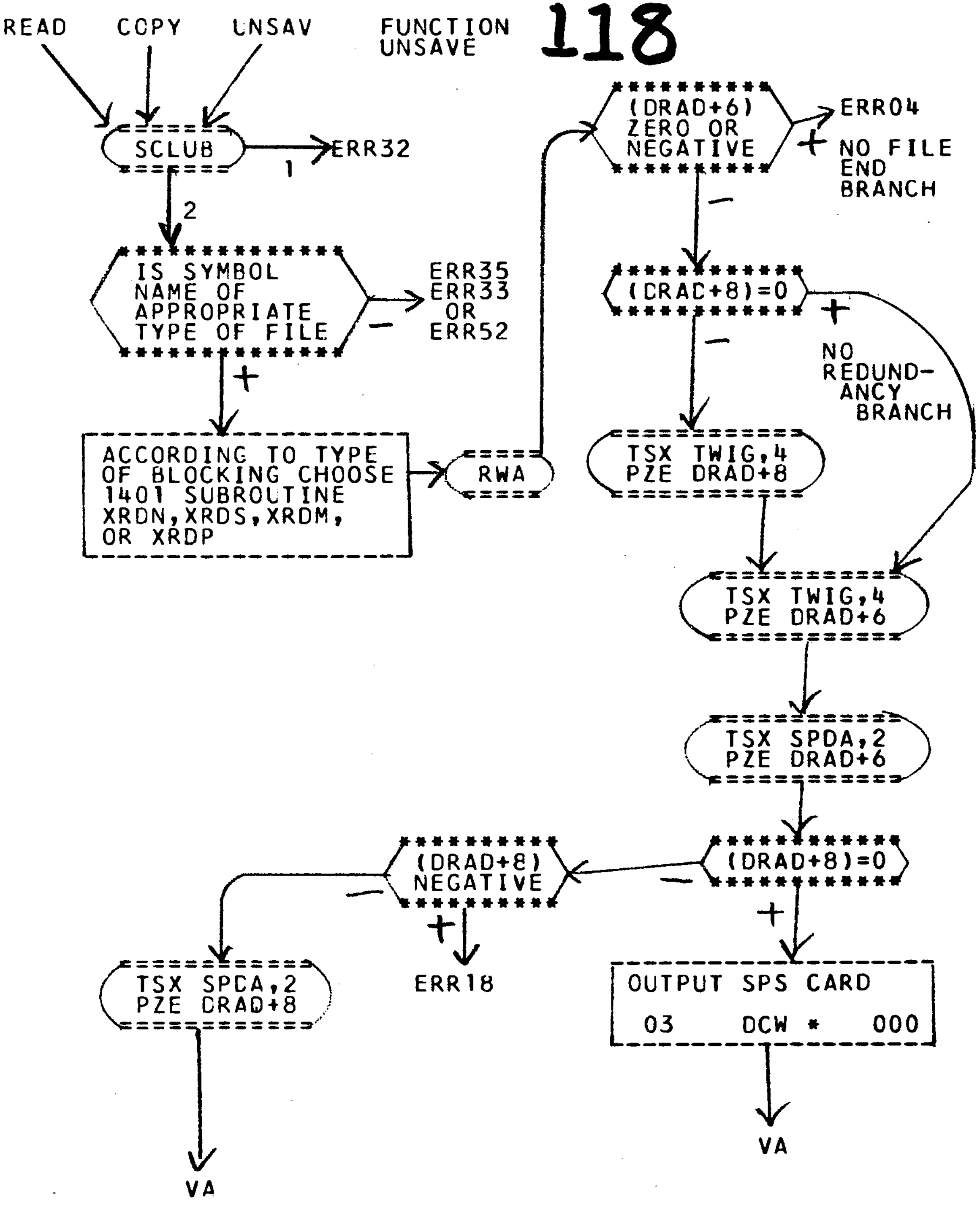
Q=N+1

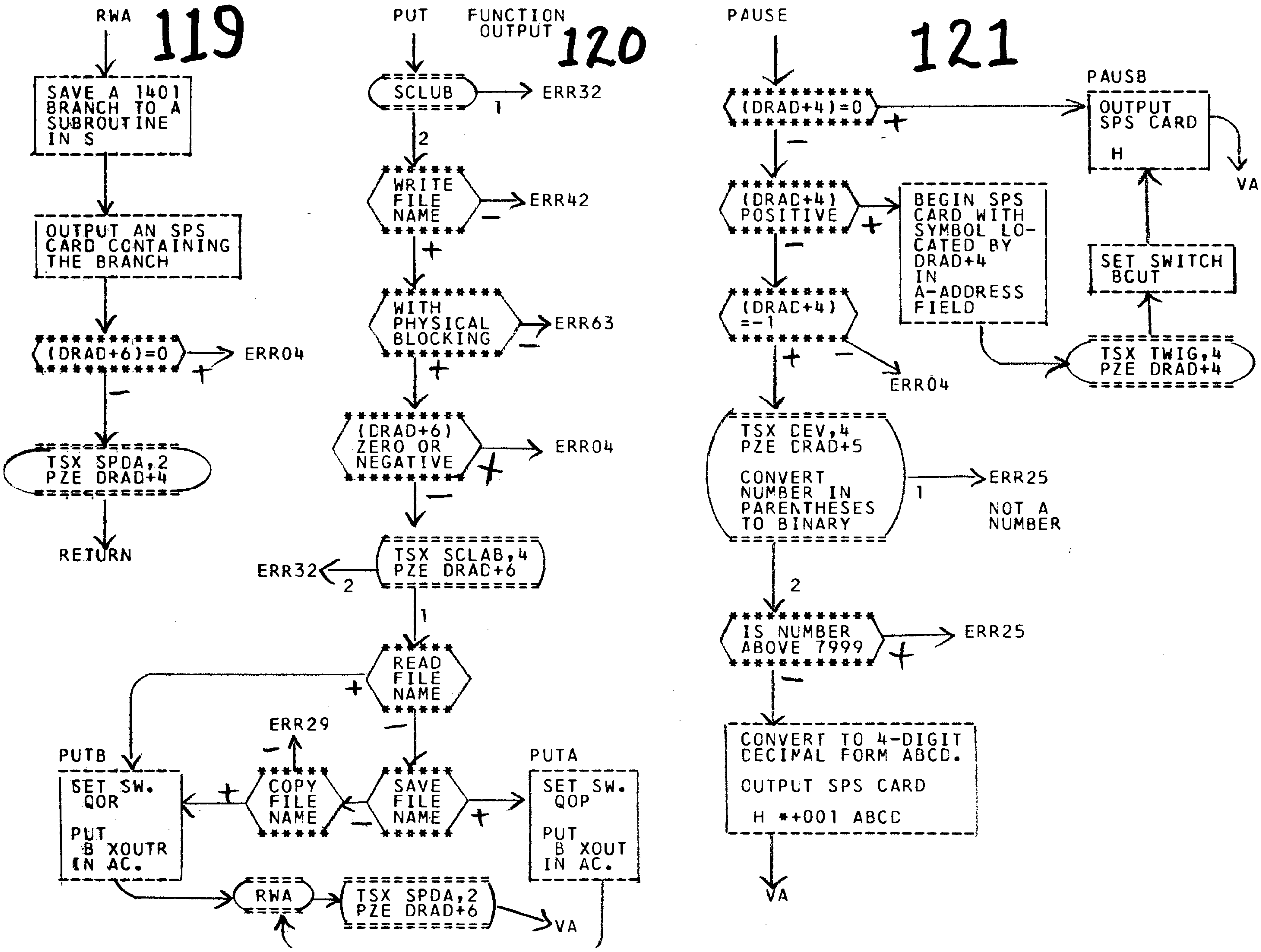
RETURN TO B+N+1

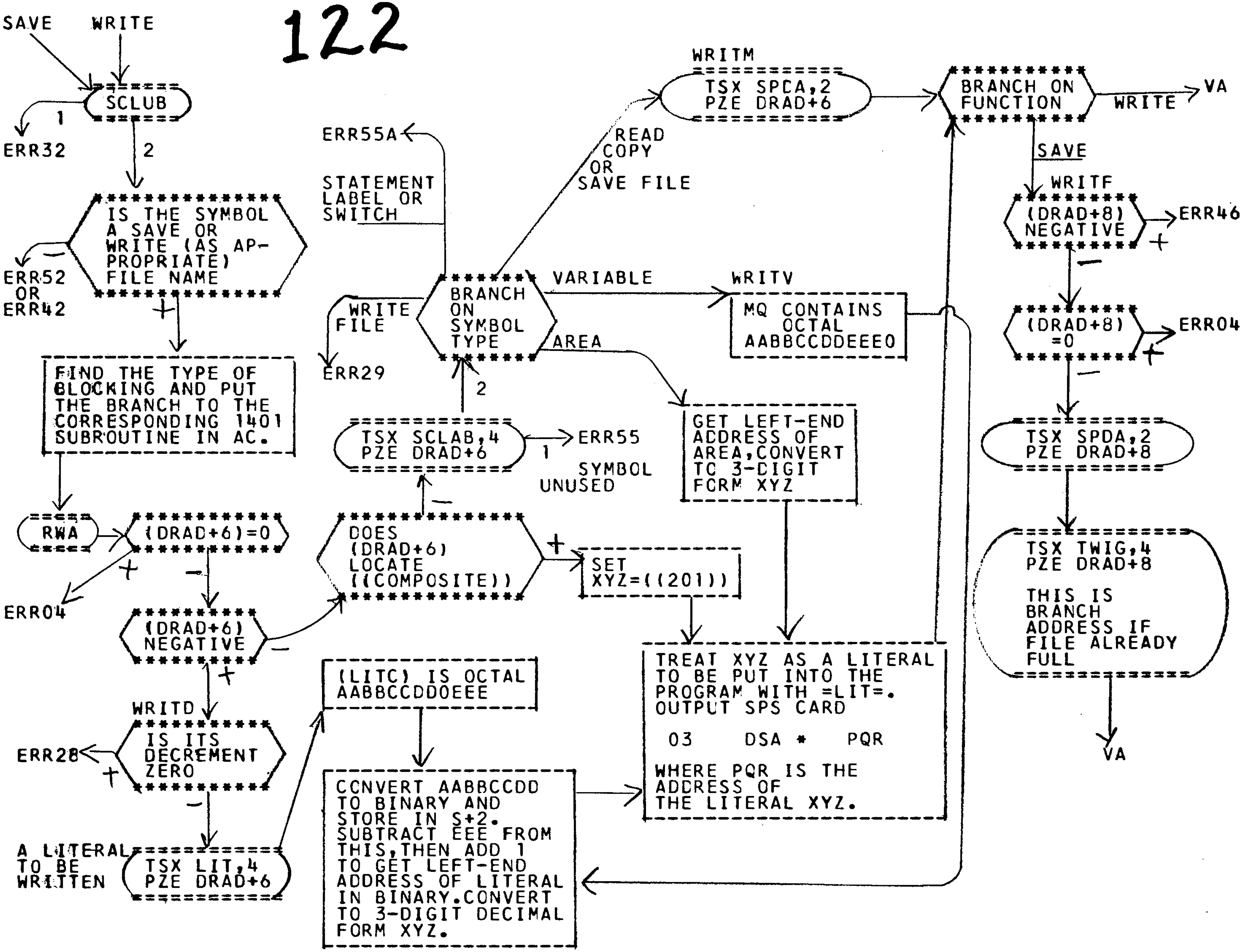
FUNCTION 9CLOSE 117

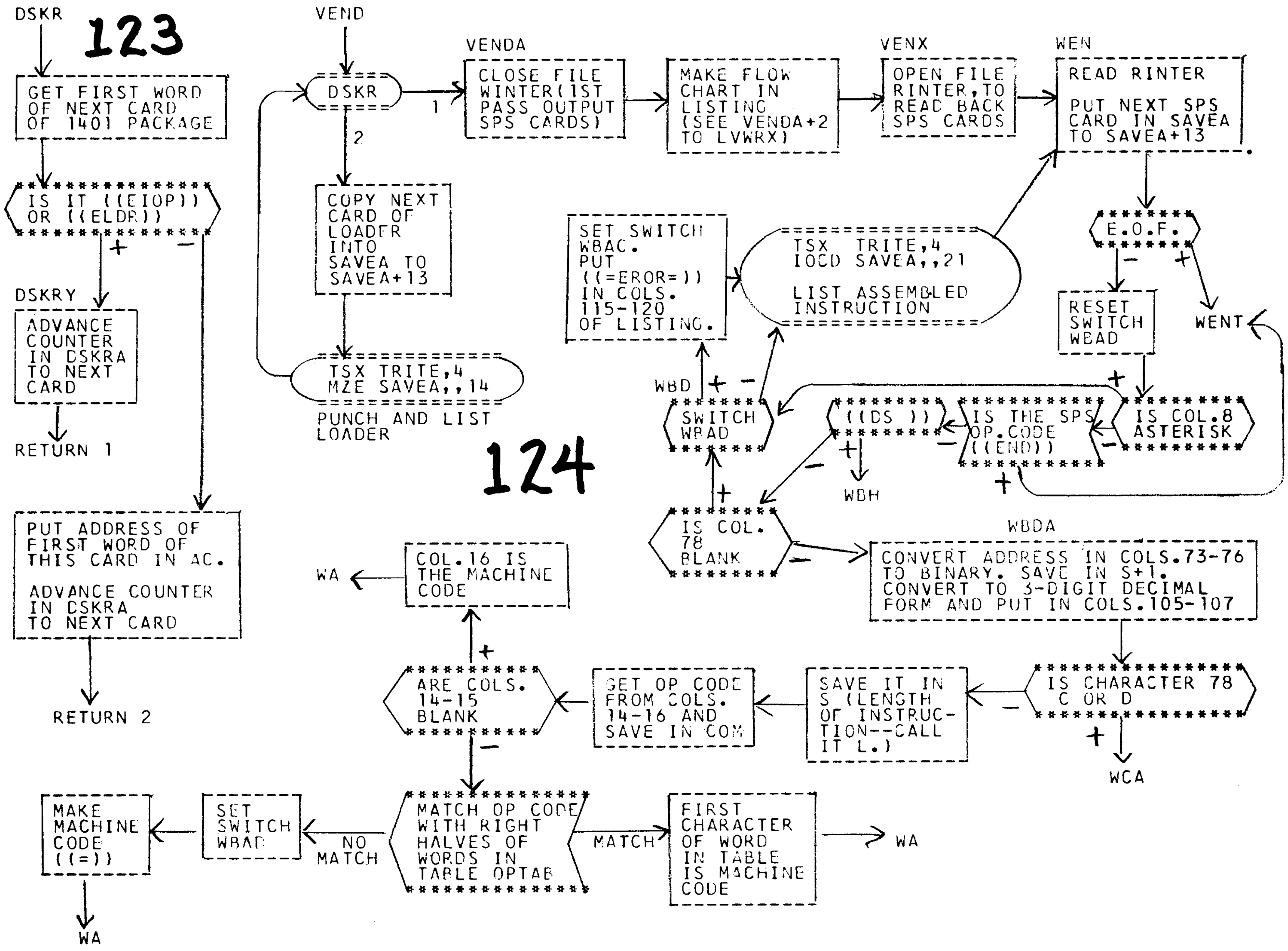


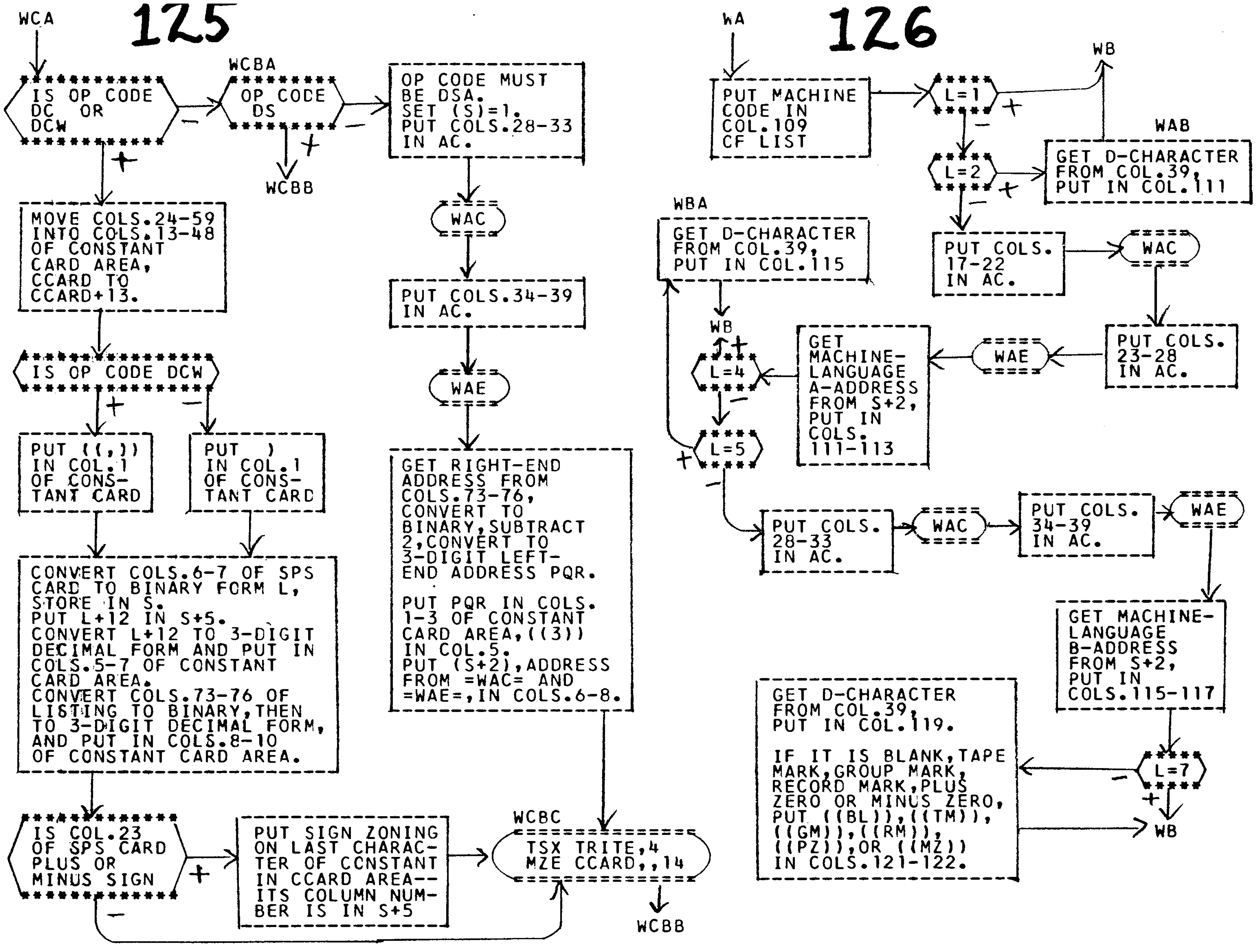
FUNCTION UNSAVE 118

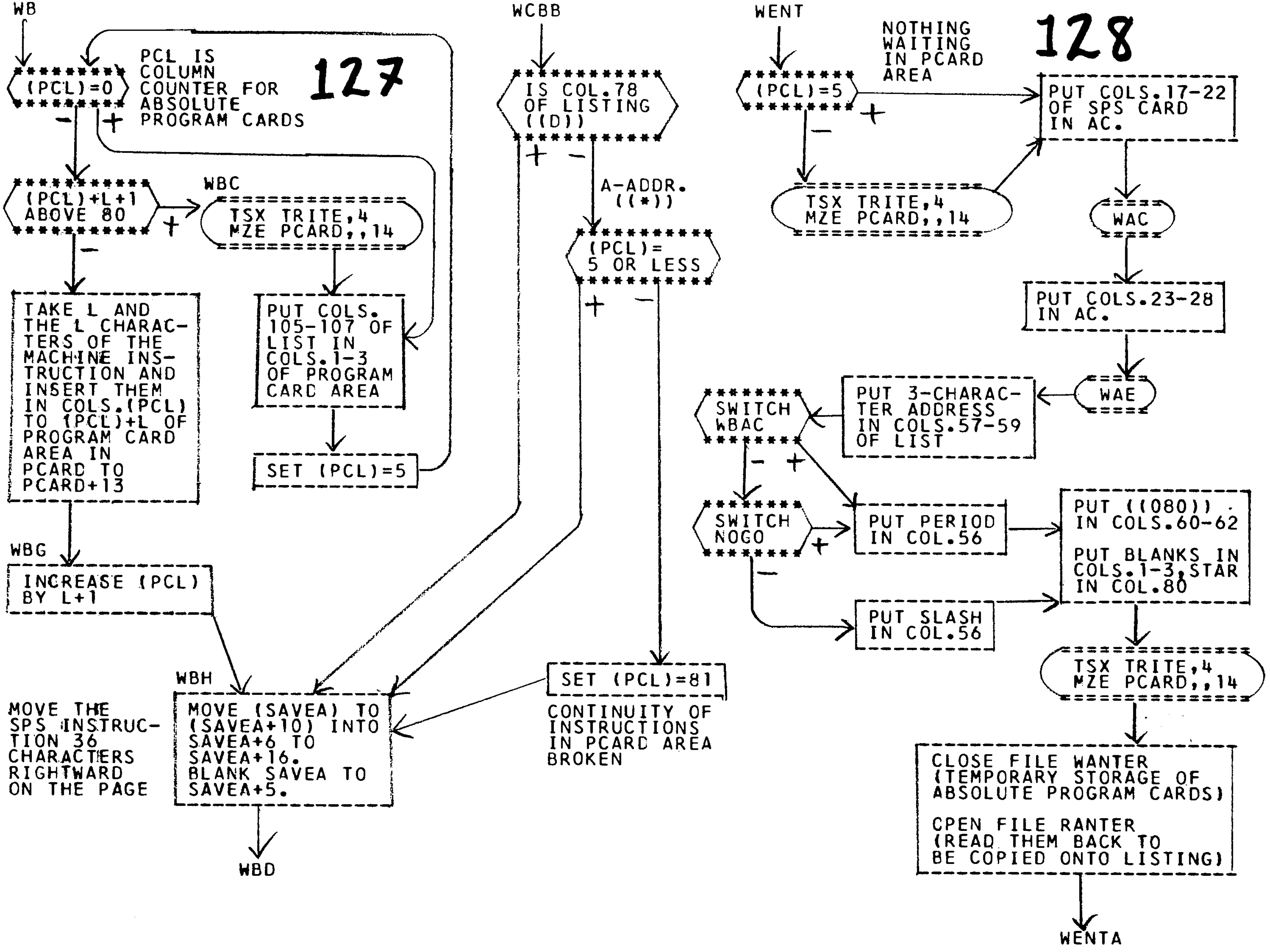



119
RWA
SAVE A 1401 BRANCH TO A SUBROUTINE IN S
OUTPUT AN SPS CARD CONTAINING THE BRANCH
(DRAD+6)=0
+
ERR04
-
TSX SPDA,2
PZE DRAD+4
RETURN
120
PUT
FUNCTION OUTPUT
SCLUB
1
ERR32
2
WRITE FILE NAME
-
ERR42
+
WITH PHYSICAL BLOCKING
-
ERR63
+
(DRAD+6) ZERO OR NEGATIVE
+
ERR04
-
TSX SCLAB,4
PZE DRAD+6
2
ERR32
1
READ FILE NAME
+
-
SAVE FILE NAME
+
PUTA
SET SW. QOP
PUT B XOUT IN AC.
-
COPY FILE NAME
-
ERR29
+
PUTB
SET SW. QOR
PUT B XOUTR IN AC.
RWA
TSX SPDA,2
PZE DRAD+6
VA
121
PAUSE
(DRAD+4)=0
+
PAUSB
OUTPUT SPS CARD
H
VA
-
(DRAD+4) POSITIVE
+
BEGIN SPS CARD WITH SYMBOL LOCATED BY DRAD+4 IN A-ADDRESS FIELD
TSX TWIG,4
PZE DRAD+4
SET SWITCH BCUT
-
(DRAC+4) =-1
-
ERR04
+
TSX DEV,4
PZE DRAD+5
CONVERT NUMBER IN PARENTHESES TO BINARY
1
ERR25
NOT A NUMBER
2
IS NUMBER ABOVE 7999
+
ERR25
-
CONVERT TO 4-DIGIT DECIMAL FORM ABCD.
OUTPUT SPS CARD
H *+001 ABCD
VA

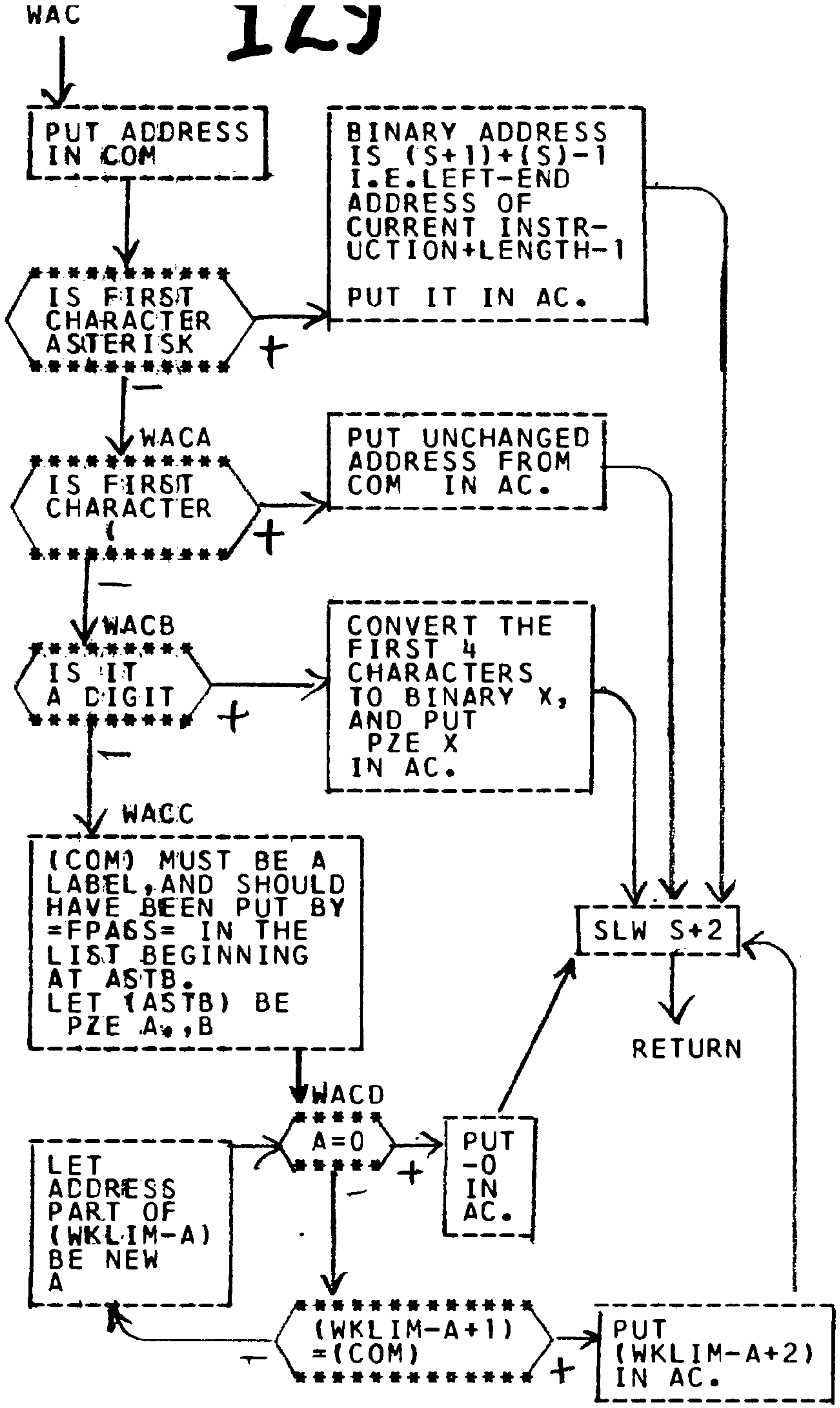

SAVE
WRITE
SCLUB
1
ERR32
2
IS THE SYMBOL A SAVE OR WRITE (AS APPROPRIATE) FILE NAME
−
ERR52 OR ERR42
+
FIND THE TYPE OF BLOCKING AND PUT THE BRANCH TO THE CORRESPONDING 1401 SUBROUTINE IN AC.
RWA
(DRAD+6)=0
+
ERR04
−
(DRAD+6) NEGATIVE
−
+
WRITD
IS ITS DECREMENT ZERO
ERR28
+
−
A LITERAL TO BE WRITTEN
TSX LIT,4
PZE DRAD+6
ERR55A
STATEMENT LABEL OR SWITCH
WRITE FILE
ERR29
BRANCH ON SYMBOL TYPE
2
TSX SCLAB,4
PZE DRAD+6
1
ERR55
SYMBOL UNUSED
−
DOES (DRAD+6) LOCATE ((COMPOSITE))
+
SET XYZ=((201))
(LITC) IS OCTAL AABBCCDDOEEE
CONVERT AABBCCDD TO BINARY AND STORE IN S+2. SUBTRACT EEE FROM THIS,THEN ADD 1 TO GET LEFT-END ADDRESS OF LITERAL IN BINARY.CONVERT TO 3-DIGIT DECIMAL FORM XYZ.
READ COPY OR SAVE FILE
WRITM
TSX SPCA,2
PZE DRAD+6
VARIABLE
WRITV
MQ CONTAINS OCTAL AABBCCDDEEE0
AREA
GET LEFT-END ADDRESS OF AREA,CONVERT TO 3-DIGIT FORM XYZ
TREAT XYZ AS A LITERAL TO BE PUT INTO THE PROGRAM WITH =LIT=. OUTPUT SPS CARD
03 DSA * PQR
WHERE PQR IS THE ADDRESS OF THE LITERAL XYZ.
BRANCH ON FUNCTION
WRITE
VA
SAVE
WRITF
(DRAD+8) NEGATIVE
ERR46
+
−
(DRAD+8) =0
ERR04
+
−
TSX SPDA,2
PZE DRAD+8
TSX TWIG,4
PZE DRAD+8
THIS IS BRANCH ADDRESS IF FILE ALREADY FULL
VA


123
DSKR
GET FIRST WORD OF NEXT CARD OF 1401 PACKAGE
IS IT ((EIOP)) OR ((ELDR))
+
-
DSKRY
ADVANCE COUNTER IN DSKRA TO NEXT CARD
RETURN 1
PUT ADDRESS OF FIRST WORD OF THIS CARD IN AC.
ADVANCE COUNTER IN DSKRA TO NEXT CARD
RETURN 2
124
VEND
DSKR
1
2
COPY NEXT CARD OF LOADER INTO SAVEA TO SAVEA+13
TSX TRITE,4
MZE SAVEA,,14
PUNCH AND LIST LOADER
VENDA
CLOSE FILE WINTER(1ST PASS OUTPUT SPS CARDS)
MAKE FLOW CHART IN LISTING (SEE VENDA+2 TO LVWRX)
VENX
OPEN FILE RINTER,TO READ BACK SPS CARDS
WEN
READ RINTER
PUT NEXT SPS CARD IN SAVEA TO SAVEA+13
E.O.F.
-
+
RESET SWITCH WBAD
WENT
+
IS COL.8 ASTERISK
-
IS THE SPS OP.CODE ((END))
+
-
((DS))
+
WBH
-
SET SWITCH WBAC.
PUT ((=EROR=)) IN COLS. 115-120 OF LISTING.
TSX TRITE,4
IOCD SAVEA,,21
LIST ASSEMBLED INSTRUCTION
WBD
+
-
SWITCH WBAD
+
IS COL. 78 BLANK
-
WBDA
CONVERT ADDRESS IN COLS.73-76 TO BINARY. SAVE IN S+1.
CONVERT TO 3-DIGIT DECIMAL FORM AND PUT IN COLS.105-107
IS CHARACTER 78 C OR D
+
WCA
-
SAVE IT IN S (LENGTH OF INSTRUC-TION--CALL IT L.)
GET OP CODE FROM COLS. 14-16 AND SAVE IN COM
ARE COLS. 14-15 BLANK
+
COL.16 IS THE MACHINE CODE
WA
-
MATCH OP CODE WITH RIGHT HALVES OF WORDS IN TABLE OPTAB
MATCH
FIRST CHARACTER OF WORD IN TABLE IS MACHINE CODE
WA
NO MATCH
SET SWITCH WBAD
MAKE MACHINE CODE ((=))
WA

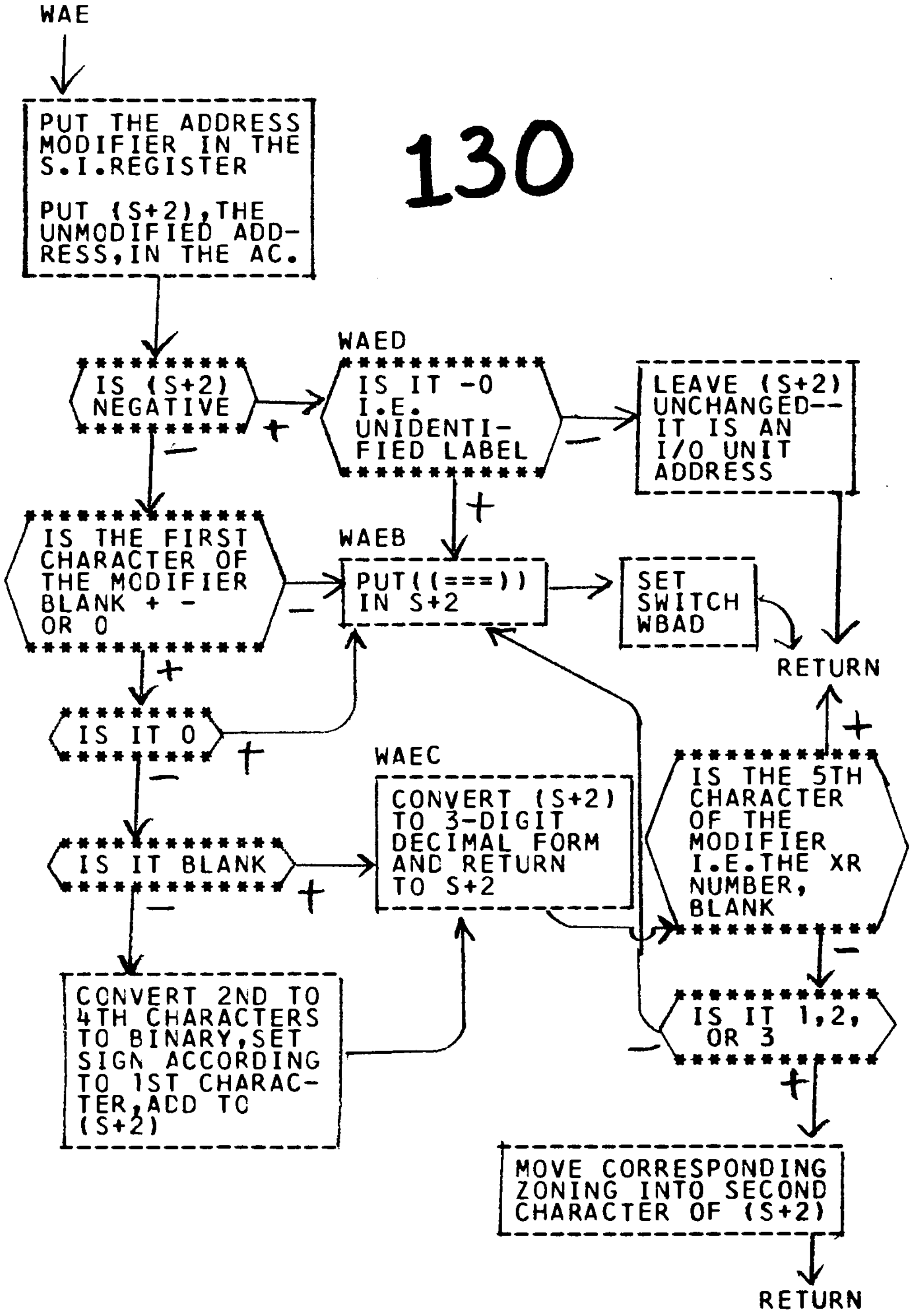

125
WCA
IS OP CODE DC OR DCW
+
-
WCBA
OP CODE DS
+
-
WCBB
OP CODE MUST BE DSA. SET (S)=1. PUT COLS.28-33 IN AC.
WAC
PUT COLS.34-39 IN AC.
WAE
GET RIGHT-END ADDRESS FROM COLS.73-76, CONVERT TO BINARY,SUBTRACT 2,CONVERT TO 3-DIGIT LEFT-END ADDRESS PQR.
PUT PQR IN COLS. 1-3 OF CONSTANT CARD AREA,((3)) IN COL.5. PUT (S+2),ADDRESS FROM =WAC= AND =WAE=,IN COLS.6-8.
MOVE COLS.24-59 INTO COLS.13-48 OF CONSTANT CARD AREA, CCARD TO CCARD+13.
IS OP CODE DCW
+
-
PUT ((,)) IN COL.1 OF CONSTANT CARD
PUT) IN COL.1 OF CONSTANT CARD
CONVERT COLS.6-7 OF SPS CARD TO BINARY FORM L, STORE IN S. PUT L+12 IN S+5. CONVERT L+12 TO 3-DIGIT DECIMAL FORM AND PUT IN COLS.5-7 OF CONSTANT CARD AREA. CONVERT COLS.73-76 OF LISTING TO BINARY,THEN TO 3-DIGIT DECIMAL FORM, AND PUT IN COLS.8-10 OF CONSTANT CARD AREA.
IS COL.23 OF SPS CARD PLUS OR MINUS SIGN
+
-
PUT SIGN ZONING ON LAST CHARACTER OF CONSTANT IN CCARD AREA-- ITS COLUMN NUMBER IS IN S+5
WCBC
TSX TRITE,4
MZE CCARD,,14
WCBB
126
WA
PUT MACHINE CODE IN COL.109 CF LIST
L=1
+
-
WB
L=2
+
-
WAB
GET D-CHARACTER FROM COL.39, PUT IN COL.111
PUT COLS. 17-22 IN AC.
WAC
PUT COLS. 23-28 IN AC.
WAE
GET MACHINE-LANGUAGE A-ADDRESS FROM S+2, PUT IN COLS. 111-113
WBA
GET D-CHARACTER FROM COL.39, PUT IN COL.115
WB
L=4
+
-
L=5
+
-
PUT COLS. 28-33 IN AC.
WAC
PUT COLS. 34-39 IN AC.
WAE
GET MACHINE-LANGUAGE B-ADDRESS FROM S+2, PUT IN COLS.115-117
L=7
-
+
WB
GET D-CHARACTER FROM COL.39, PUT IN COL.119.
IF IT IS BLANK,TAPE MARK,GROUP MARK, RECORD MARK,PLUS ZERO OR MINUS ZERO, PUT ((BL)),((TM)), ((GM)),((RM)), ((PZ)),OR ((MZ)) IN COLS.121-122.

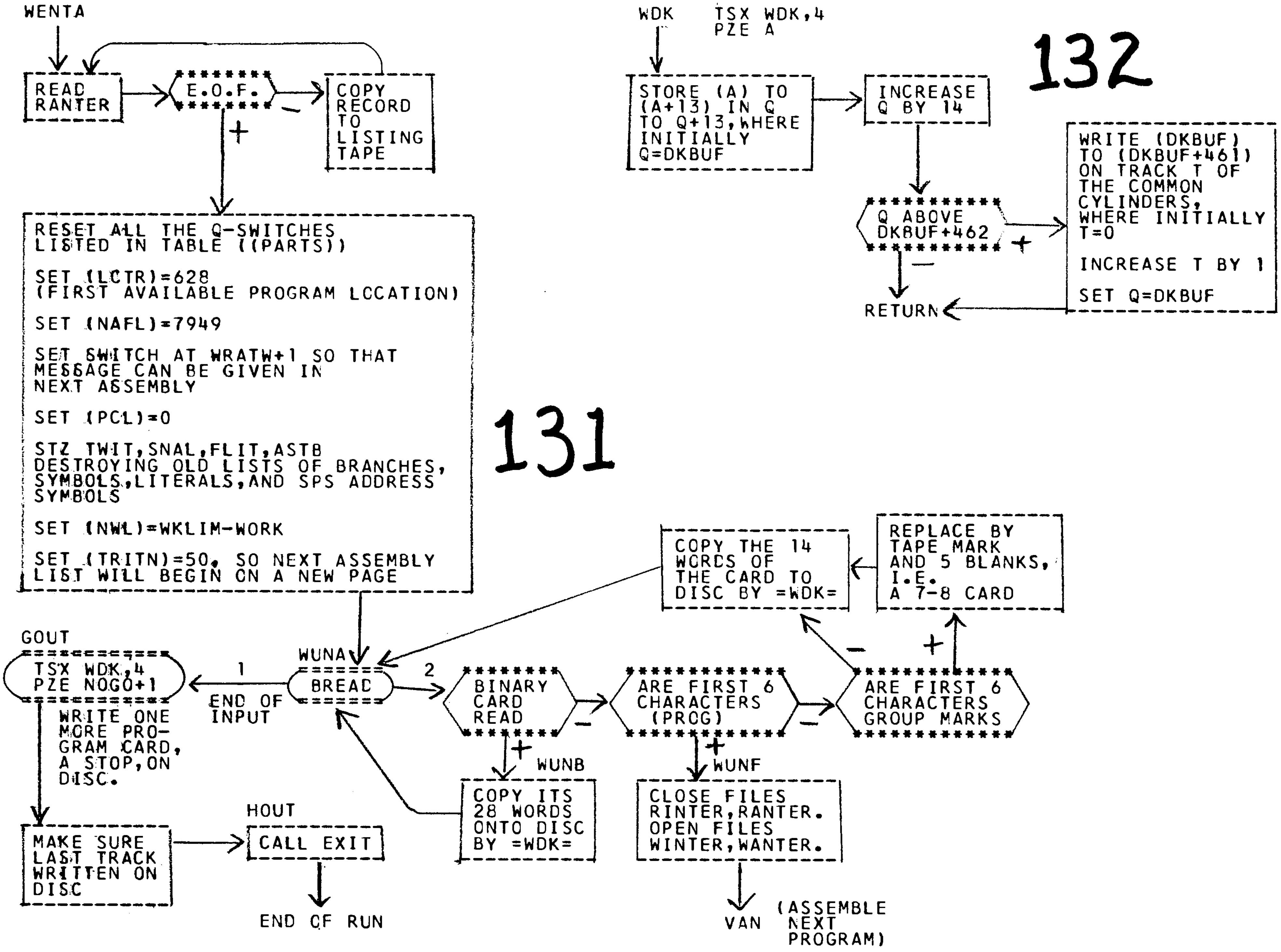

127
WB
(PCL)=0
PCL IS COLUMN COUNTER FOR ABSOLUTE PROGRAM CARDS
- +
(PCL)+L+1 ABOVE 80
+ -
WBC
TSX TRITE,4
MZE PCARD,,14
TAKE L AND THE L CHARACTERS OF THE MACHINE INSTRUCTION AND INSERT THEM IN COLS.(PCL) TO (PCL)+L OF PROGRAM CARD AREA IN PCARD TO PCARD+13
PUT COLS. 105-107 OF LIST IN COLS.1-3 OF PROGRAM CARD AREA
SET (PCL)=5
WBG
INCREASE (PCL) BY L+1
WCBB
IS COL.78 OF LISTING ((D))
+ -
A-ADDR. ((*))
(PCL)= 5 OR LESS
+ -
SET (PCL)=81
CONTINUITY OF INSTRUCTIONS IN PCARD AREA BROKEN
WBH
MOVE (SAVEA) TO (SAVEA+10) INTO SAVEA+6 TO SAVEA+16. BLANK SAVEA TO SAVEA+5.
MOVE THE SPS INSTRUCTION 36 CHARACTERS RIGHTWARD ON THE PAGE
WBD
128
WENT
(PCL)=5
NOTHING WAITING IN PCARD AREA
+ -
TSX TRITE,4
MZE PCARD,,14
PUT COLS.17-22 OF SPS CARD IN AC.
WAC
PUT COLS.23-28 IN AC.
WAE
PUT 3-CHARACTER ADDRESS IN COLS.57-59 OF LIST
SWITCH WBAC
- +
SWITCH NOGO
+ -
PUT PERIOD IN COL.56
PUT SLASH IN COL.56
PUT ((080)) IN COLS.60-62
PUT BLANKS IN COLS.1-3,STAR IN COL.80
TSX TRITE,4
MZE PCARD,,14
CLOSE FILE WANTER (TEMPORARY STORAGE OF ABSOLUTE PROGRAM CARDS)
OPEN FILE RANTER (READ THEM BACK TO BE COPIED ONTO LISTING)
WENTA

129


WAC
PUT ADDRESS IN COM
IS FIRST CHARACTER ASTERISK
+
BINARY ADDRESS IS (S+1)+(S)-1 I.E.LEFT-END ADDRESS OF CURRENT INSTRUCTION+LENGTH-1
PUT IT IN AC.
−
WACA
IS FIRST CHARACTER (
+
PUT UNCHANGED ADDRESS FROM COM IN AC.
−
WACB
IS IT A DIGIT
+
CONVERT THE FIRST 4 CHARACTERS TO BINARY X, AND PUT PZE X IN AC.
−
WACC
(COM) MUST BE A LABEL,AND SHOULD HAVE BEEN PUT BY =FPASS= IN THE LIST BEGINNING AT ASTB. LET (ASTB) BE PZE A,,B
SLW S+2
RETURN
WACD
A=0
+
PUT -0 IN AC.
−
LET ADDRESS PART OF (WKLIM-A) BE NEW A
(WKLIM-A+1) =(COM)
−
+
PUT (WKLIM-A+2) IN AC.

130


WAE
PUT THE ADDRESS MODIFIER IN THE S.I.REGISTER
PUT (S+2),THE UNMODIFIED ADDRESS,IN THE AC.
IS (S+2) NEGATIVE
+
WAED
IS IT -0 I.E. UNIDENTIFIED LABEL
−
LEAVE (S+2) UNCHANGED--IT IS AN I/O UNIT ADDRESS
+
−
IS THE FIRST CHARACTER OF THE MODIFIER BLANK + - OR 0
−
WAEB
PUT((===)) IN S+2
SET SWITCH WBAD
RETURN
+
IS IT 0
+
−
WAEC
IS IT BLANK
+
CONVERT (S+2) TO 3-DIGIT DECIMAL FORM AND RETURN TO S+2
−
CONVERT 2ND TO 4TH CHARACTERS TO BINARY,SET SIGN ACCORDING TO 1ST CHARACTER,ADD TO (S+2)
IS THE 5TH CHARACTER OF THE MODIFIER I.E.THE XR NUMBER, BLANK
+
−
IS IT 1,2, OR 3
−
+
MOVE CORRESPONDING ZONING INTO SECOND CHARACTER OF (S+2)
RETURN



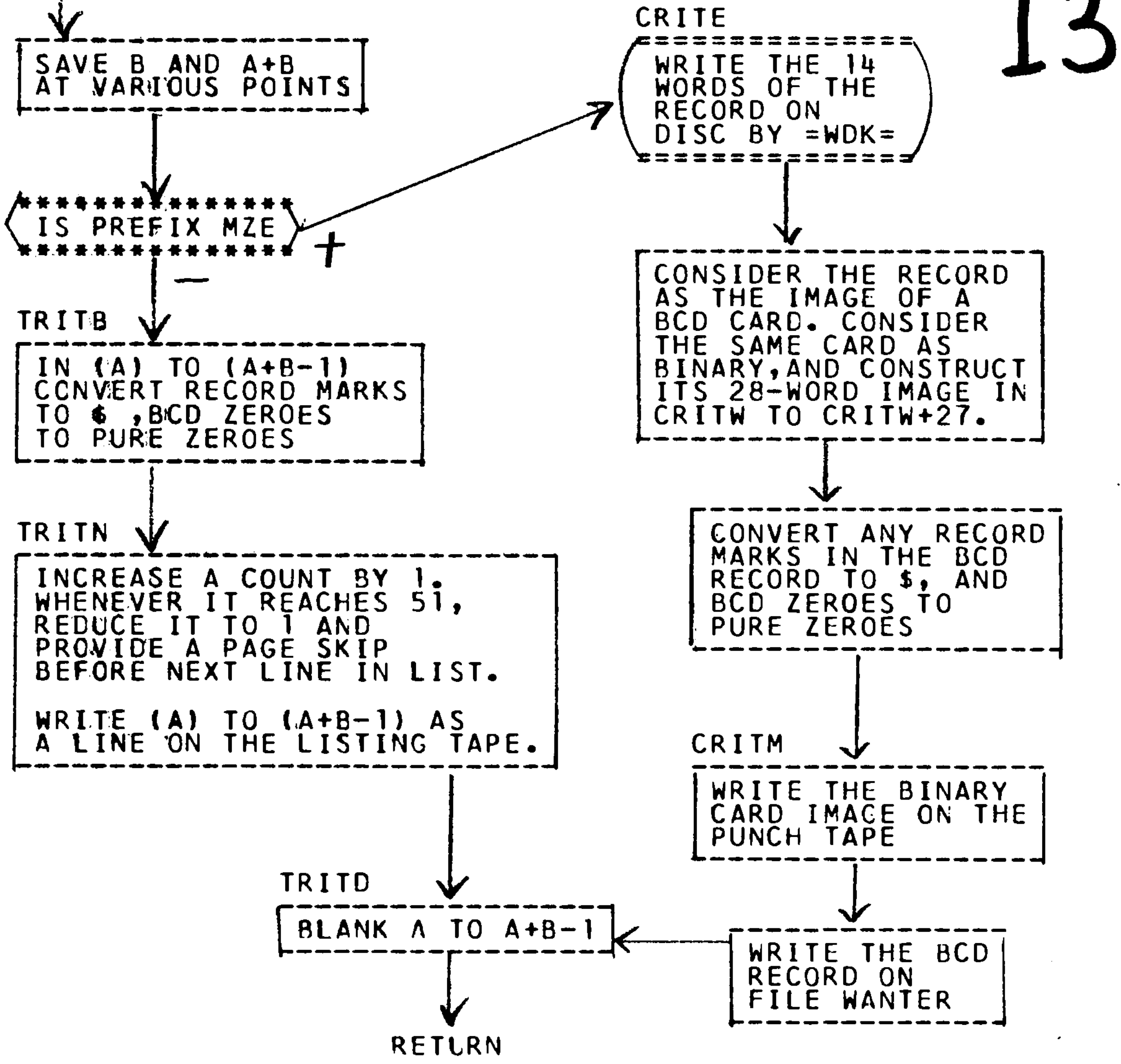
WENTA
READ RANTER
E.O.F.
+
−
COPY RECORD TO LISTING TAPE
RESET ALL THE Q-SWITCHES LISTED IN TABLE ((PARTS))
SET (LCTR)=628 (FIRST AVAILABLE PROGRAM LOCATION)
SET (NAFL)=7949
SET SWITCH AT WRATW+1 SO THAT MESSAGE CAN BE GIVEN IN NEXT ASSEMBLY
SET (PCL)=0
STZ TWIT,SNAL,FLIT,ASTB DESTROYING OLD LISTS OF BRANCHES, SYMBOLS,LITERALS,AND SPS ADDRESS SYMBOLS
SET (NWL)=WKLIM-WORK
SET (TRITN)=50, SO NEXT ASSEMBLY LIST WILL BEGIN ON A NEW PAGE
GOUT
TSX WDK,4
PZE NOGO+1
1
END OF INPUT
WUNA
BREAD
2
WRITE ONE MORE PROGRAM CARD, A STOP,ON DISC.
MAKE SURE LAST TRACK WRITTEN ON DISC
HOUT
CALL EXIT
END OF RUN
BINARY CARD READ
+
WUNB
−
COPY ITS 28 WORDS ONTO DISC BY =WDK=
ARE FIRST 6 CHARACTERS (PROG)
+
WUNF
−
CLOSE FILES RINTER,RANTER. OPEN FILES WINTER,WANTER.
VAN
(ASSEMBLE NEXT PROGRAM)
ARE FIRST 6 CHARACTERS GROUP MARKS
−
+
COPY THE 14 WORDS OF THE CARD TO DISC BY =WDK=
REPLACE BY TAPE MARK AND 5 BLANKS, I.E. A 7-8 CARD
WDK
TSX WDK,4
PZE A
STORE (A) TO (A+13) IN Q TO Q+13,WHERE INITIALLY Q=DKBUF
INCREASE Q BY 14
Q ABOVE DKBUF+462
+
−
RETURN
WRITE (DKBUF) TO (DKBUF+461) ON TRACK T OF THE COMMON CYLINDERS, WHERE INITIALLY T=0
INCREASE T BY 1
SET Q=DKBUF

131

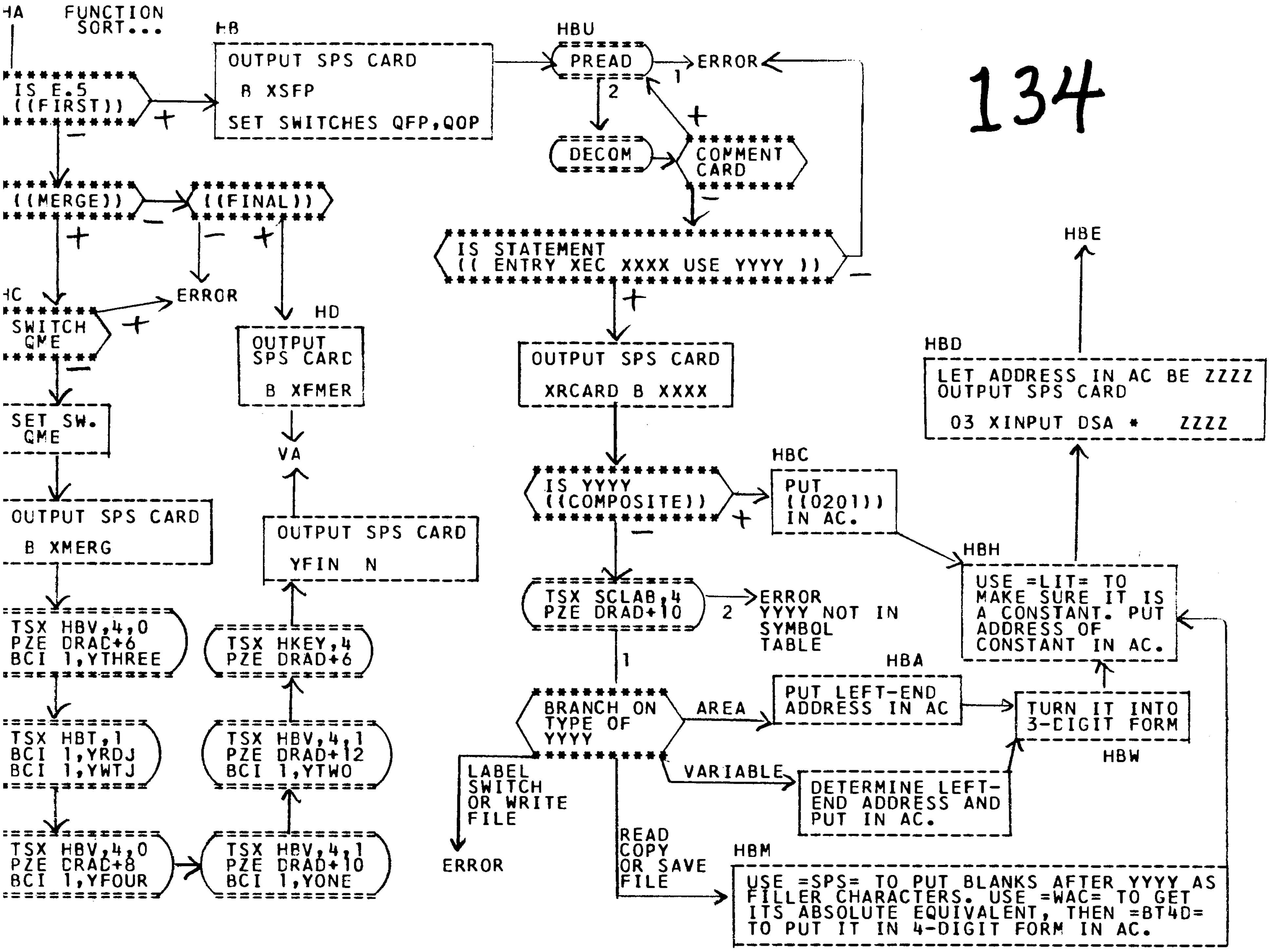
132

```
TRITE    TSX TRITE,4
         IOCD/MZE A,,B

         IF PREFIX IS IOCD, WRITE (A) TO (A+B-1)
         ON LISTING TAPE.

         IF MZE, WRITE (A) TO (A+B-1) ON FILE
         WANTER,FOR LATER COPYING TO LISTING TAPE.
         ALSO WRITE (A) TO (A+13) ON PUNCH TAPE
         AND ON DISC.
```

133

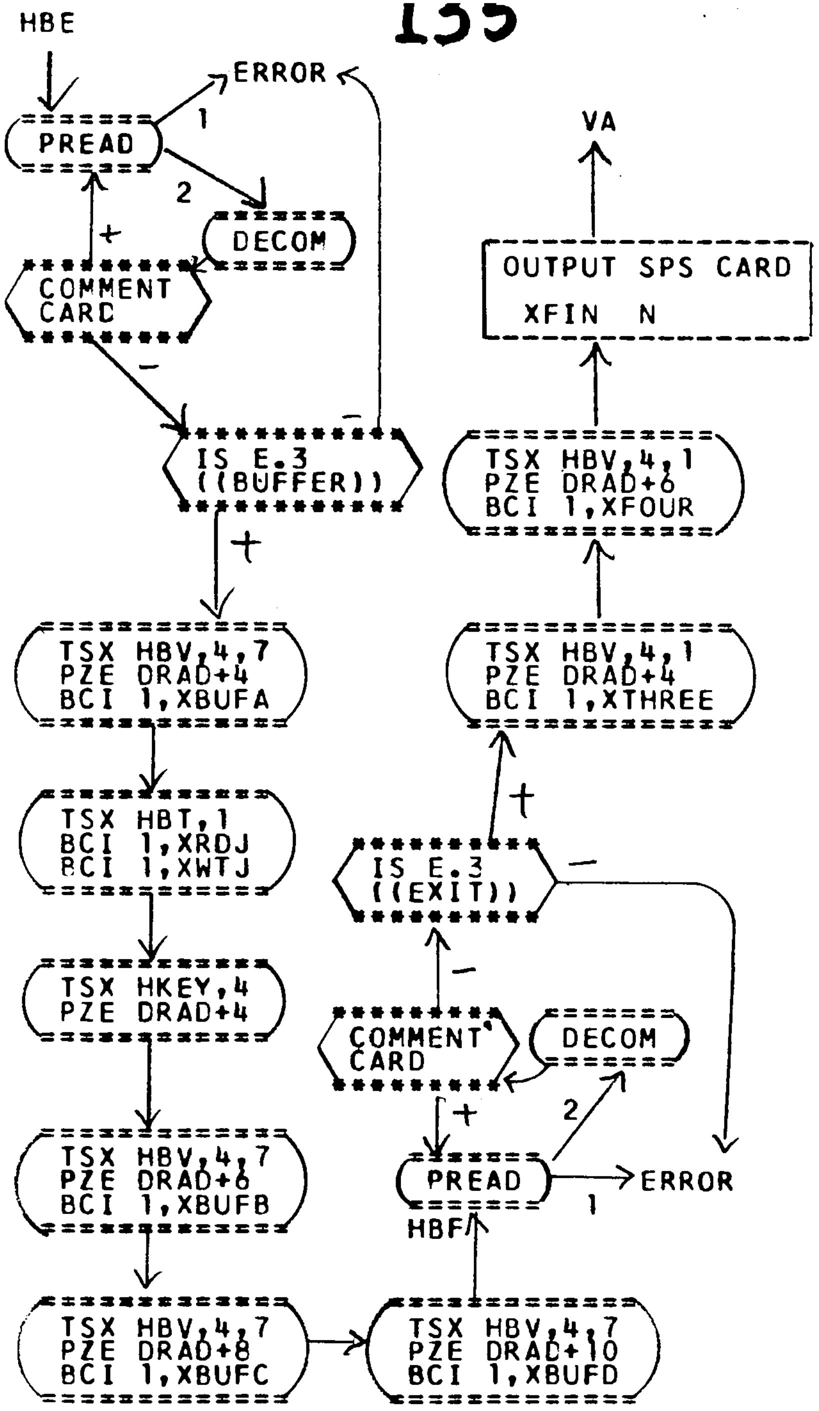

SAVE B AND A+B AT VARIOUS POINTS
IS PREFIX MZE
+
−
CRITE
WRITE THE 14 WORDS OF THE RECORD ON DISC BY =WDK=
TRITB
IN (A) TO (A+B-1) CONVERT RECORD MARKS TO $,BCD ZEROES TO PURE ZEROES
TRITN
INCREASE A COUNT BY 1. WHENEVER IT REACHES 51, REDUCE IT TO 1 AND PROVIDE A PAGE SKIP BEFORE NEXT LINE IN LIST.
WRITE (A) TO (A+B-1) AS A LINE ON THE LISTING TAPE.
CONSIDER THE RECORD AS THE IMAGE OF A BCD CARD. CONSIDER THE SAME CARD AS BINARY,AND CONSTRUCT ITS 28-WORD IMAGE IN CRITW TO CRITW+27.
CONVERT ANY RECORD MARKS IN THE BCD RECORD TO $, AND BCD ZEROES TO PURE ZEROES
CRITM
WRITE THE BINARY CARD IMAGE ON THE PUNCH TAPE
WRITE THE BCD RECORD ON FILE WANTER
TRITD
BLANK A TO A+B-1
RETURN

134



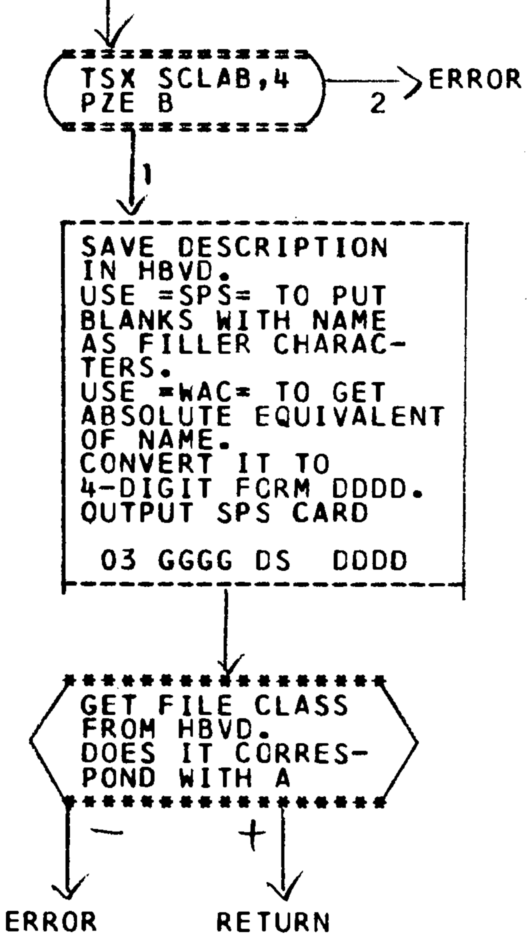
HA
FUNCTION SORT...
IS E.5 ((FIRST))
+
−
HB
OUTPUT SPS CARD
B XSFP
SET SWITCHES QFP,QOP
HBU
PREAD
1
ERROR
2
DECOM
COMMENT CARD
+
−
IS STATEMENT ((ENTRY XEC XXXX USE YYYY))
−
+
((MERGE))
−
((FINAL))
+
−
+
ERROR
HC
SWITCH QME
+
ERROR
−
SET SW. QME
OUTPUT SPS CARD
B XMERG
TSX HBV,4,0
PZE DRAC+6
BCI 1,YTHREE
TSX HBT,1
BCI 1,YRDJ
BCI 1,YWTJ
TSX HBV,4,0
PZE DRAC+8
BCI 1,YFOUR
HD
OUTPUT SPS CARD
B XFMER
VA
OUTPUT SPS CARD
YFIN N
TSX HKEY,4
PZE DRAD+6
TSX HBV,4,1
PZE DRAD+12
BCI 1,YTWO
TSX HBV,4,1
PZE DRAD+10
BCI 1,YONE
OUTPUT SPS CARD
XRCARD B XXXX
IS YYYY ((COMPOSITE))
+
−
HBC
PUT ((0201)) IN AC.
TSX SCLAB,4
PZE DRAD+10
2
ERROR
YYYY NOT IN SYMBOL TABLE
1
BRANCH ON TYPE OF YYYY
AREA
VARIABLE
LABEL SWITCH OR WRITE FILE
ERROR
READ COPY OR SAVE FILE
HBA
PUT LEFT-END ADDRESS IN AC
DETERMINE LEFT-END ADDRESS AND PUT IN AC.
TURN IT INTO 3-DIGIT FORM
HBW
HBM
USE =SPS= TO PUT BLANKS AFTER YYYY AS FILLER CHARACTERS. USE =WAC= TO GET ITS ABSOLUTE EQUIVALENT, THEN =BT4D= TO PUT IT IN 4-DIGIT FORM IN AC.
HBH
USE =LIT= TO MAKE SURE IT IS A CONSTANT. PUT ADDRESS OF CONSTANT IN AC.
HBD
LET ADDRESS IN AC BE ZZZZ
OUTPUT SPS CARD
03 XINPUT DSA * ZZZZ
HBE

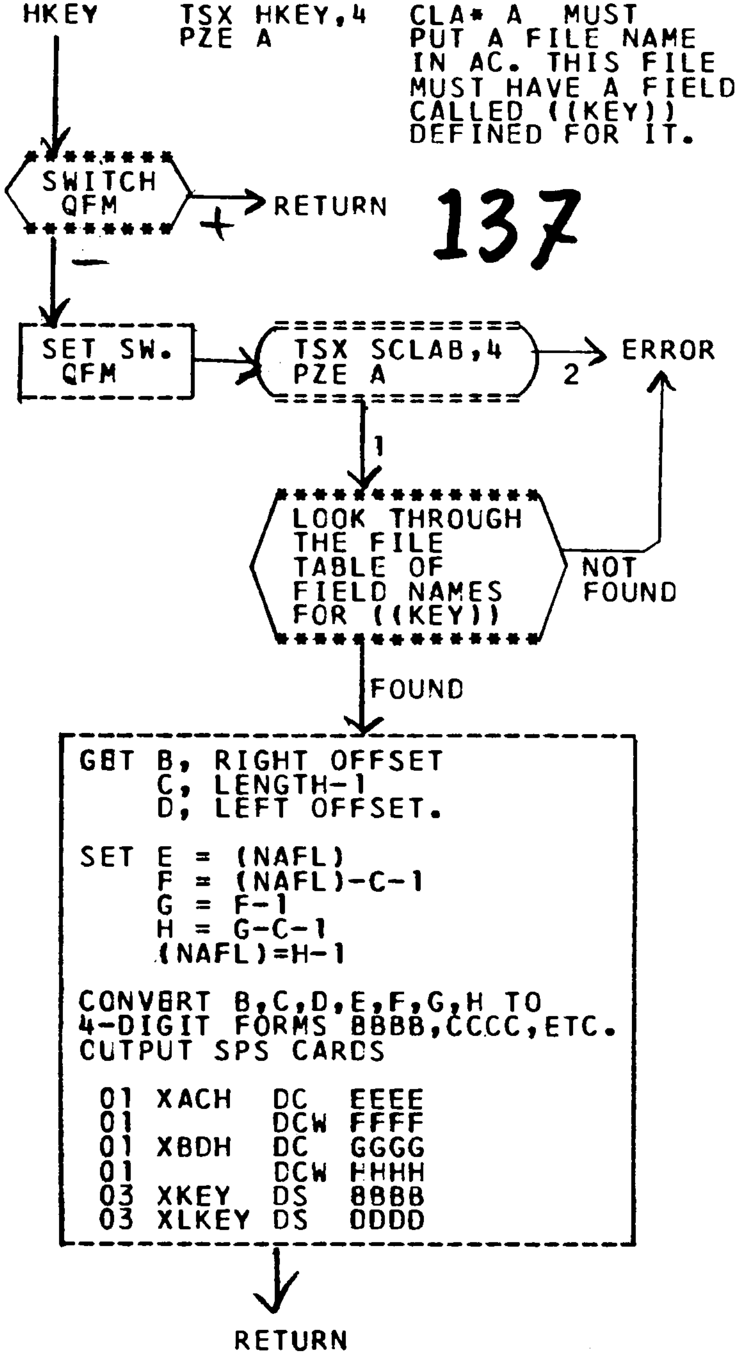
135



```
HBV    TSX HBV,4,A    CLA* B  MUST PUT A FILE NAME
       PZE B          GGGG IN AC., READ FILE IF A=0,
       BCI 1,FFFF     WRITE FILE IF A=1, SAVE FILE
                      IF A=7. FFFF MUST BE EQUATED
                      WITH FILE ADDRESS.
```

136


TSX SCLAB,4
PZE B
2
ERROR
1
SAVE DESCRIPTION IN HBVD.
USE =SPS= TO PUT BLANKS WITH NAME AS FILLER CHARACTERS.
USE =WAC= TO GET ABSOLUTE EQUIVALENT OF NAME.
CONVERT IT TO 4-DIGIT FORM DDDD.
OUTPUT SPS CARD
03 GGGG DS DDDD
GET FILE CLASS FROM HBVD. DOES IT CORRESPOND WITH A
-
+
ERROR
RETURN



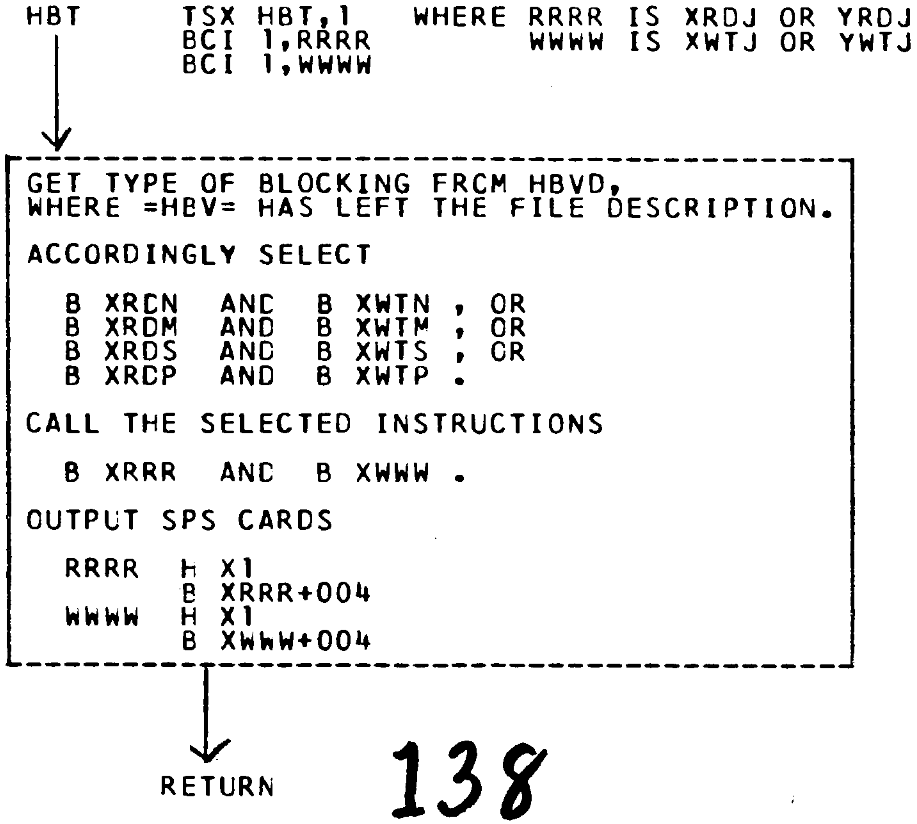
137

```
HKEY      TSX HKEY,4    CLA* A   MUST
          PZE A         PUT A FILE NAME
                        IN AC. THIS FILE
                        MUST HAVE A FIELD
                        CALLED ((KEY))
                        DEFINED FOR IT.
```

```
TSX SCLAB,4
PZE A
```

```
LOOK THROUGH
THE FILE
TABLE OF
FIELD NAMES
FOR ((KEY))
```

```
GBT B, RIGHT OFFSET
    C, LENGTH-1
    D, LEFT OFFSET.

SET E = (NAFL)
    F = (NAFL)-C-1
    G = F-1
    H = G-C-1
    (NAFL)=H-1

CONVBRT B,C,D,E,F,G,H TO
4-DIGIT FORMS BBBB,CCCC,ETC.
CUTPUT SPS CARDS

 01 XACH   DC   EEEE
 01        DCW  FFFF
 01 XBDH   DC   GGGG
 01        DCW  HHHH
 03 XKEY   DS   BBBB
 03 XLKEY  DS   DDDD
```



138

```
HBT       TSX HBT,1     WHERE RRRR IS XRDJ OR YRDJ
          BCI 1,RRRR          WWWW IS XWTJ OR YWTJ
          BCI 1,WWWW
```

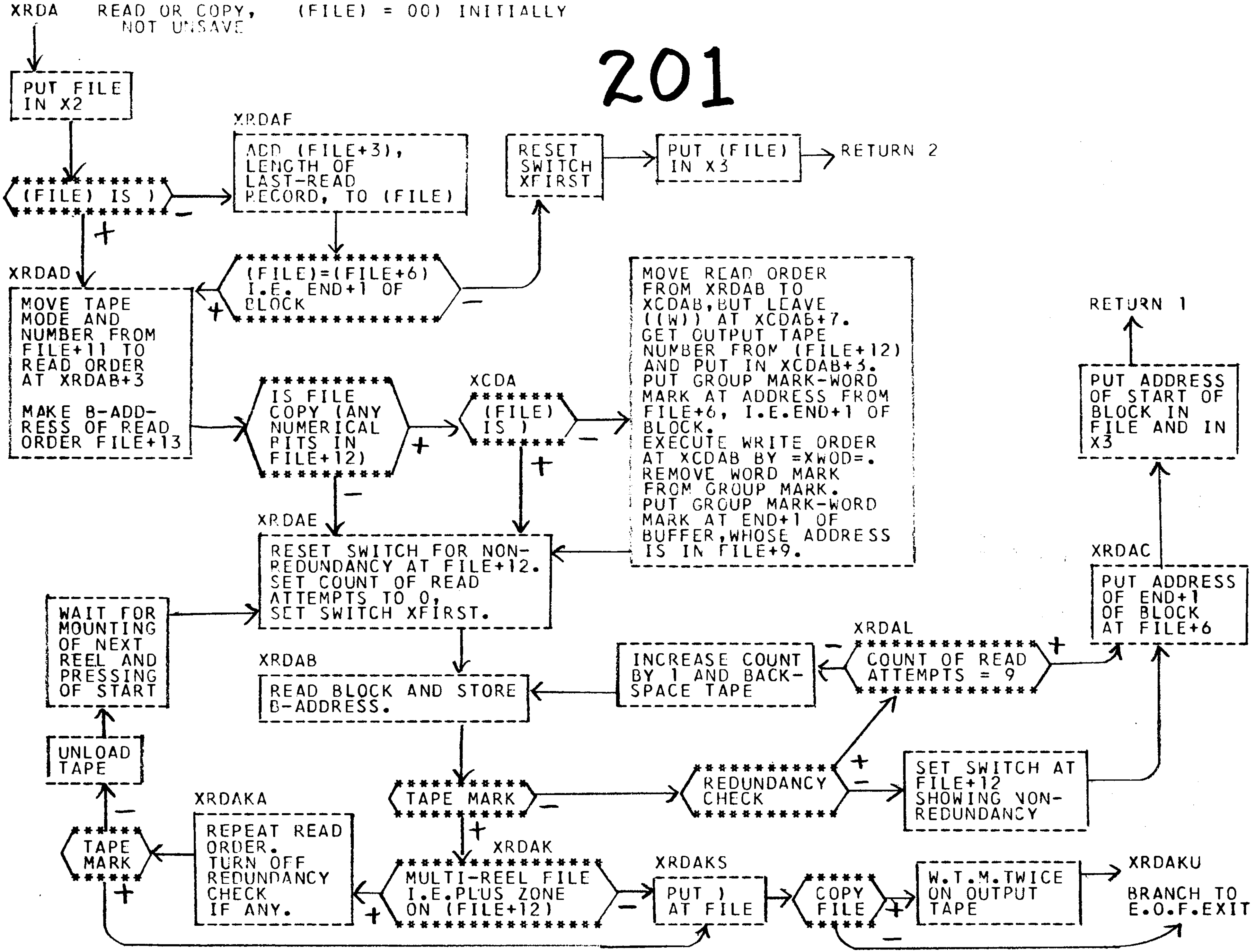
```
GET TYPE OF BLOCKING FRCM HBVD,
WHERE =HBV= HAS LEFT THE FILE DESCRIPTION.

ACCORDINGLY SELECT

   B XRCN  ANC  B XWTN , OR
   B XRDM  AND  B XWTM , OR
   B XRDS  ANC  B XWTS , OR
   B XRDP  AND  B XWTP .

CALL THE SELECTED INSTRUCTIONS

   B XRRR  ANC  B XWWW .

OUTPUT SPS CARDS

   RRRR  H X1
         B XRRR+004
   WWWW  H X1
         B XWWW+004
```

Flow Charts of some 1401 Subroutines

On pages 58-66 one finds the flow charts of the subroutines in the 1401 program package that handle magnetic tape input and output, and of the analogous storing and fetching subroutines called by the SAVE and UNSAVE functions.

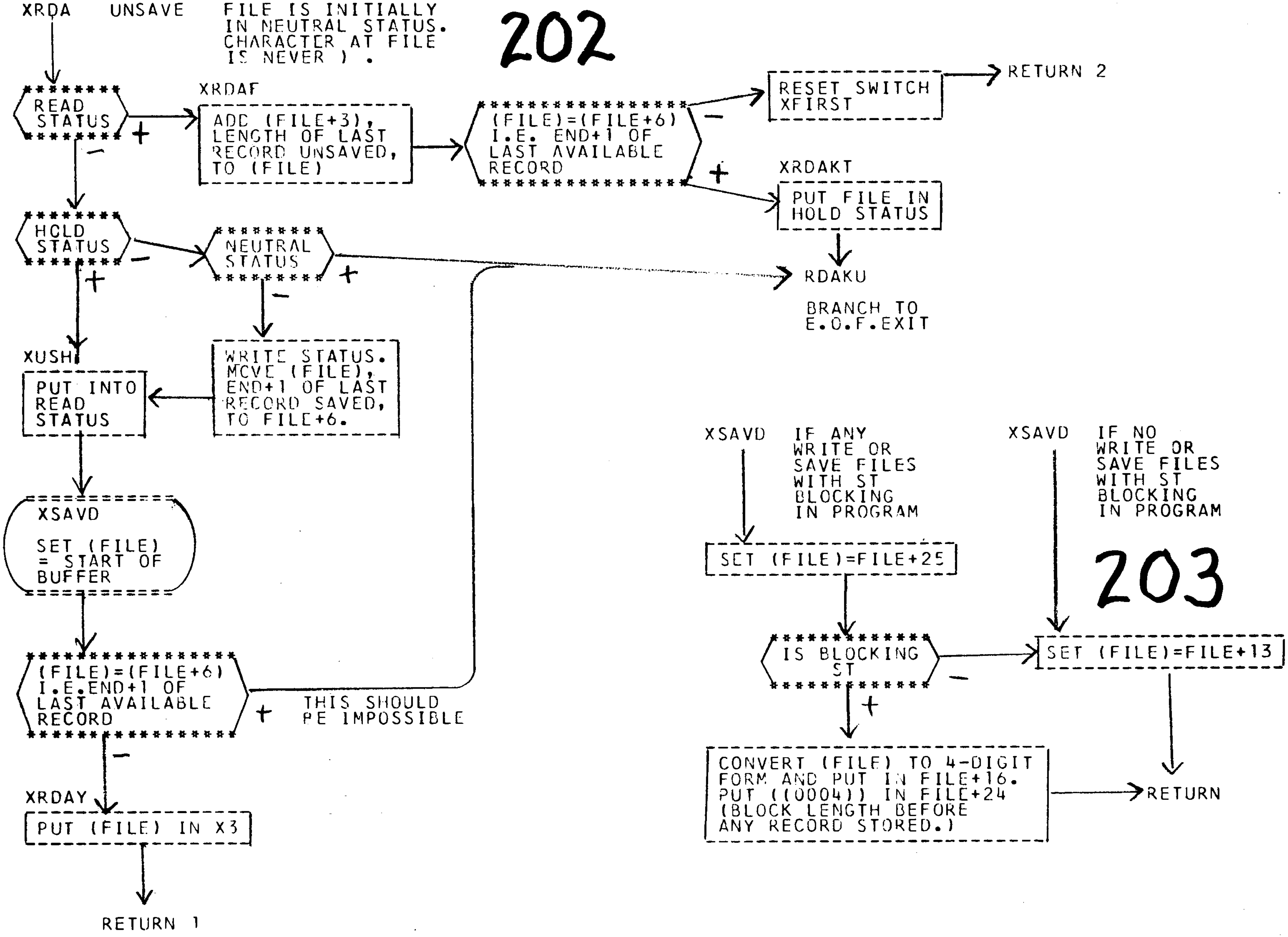
Comments on all these flow charts are given on pages 114-129 of the second volume.

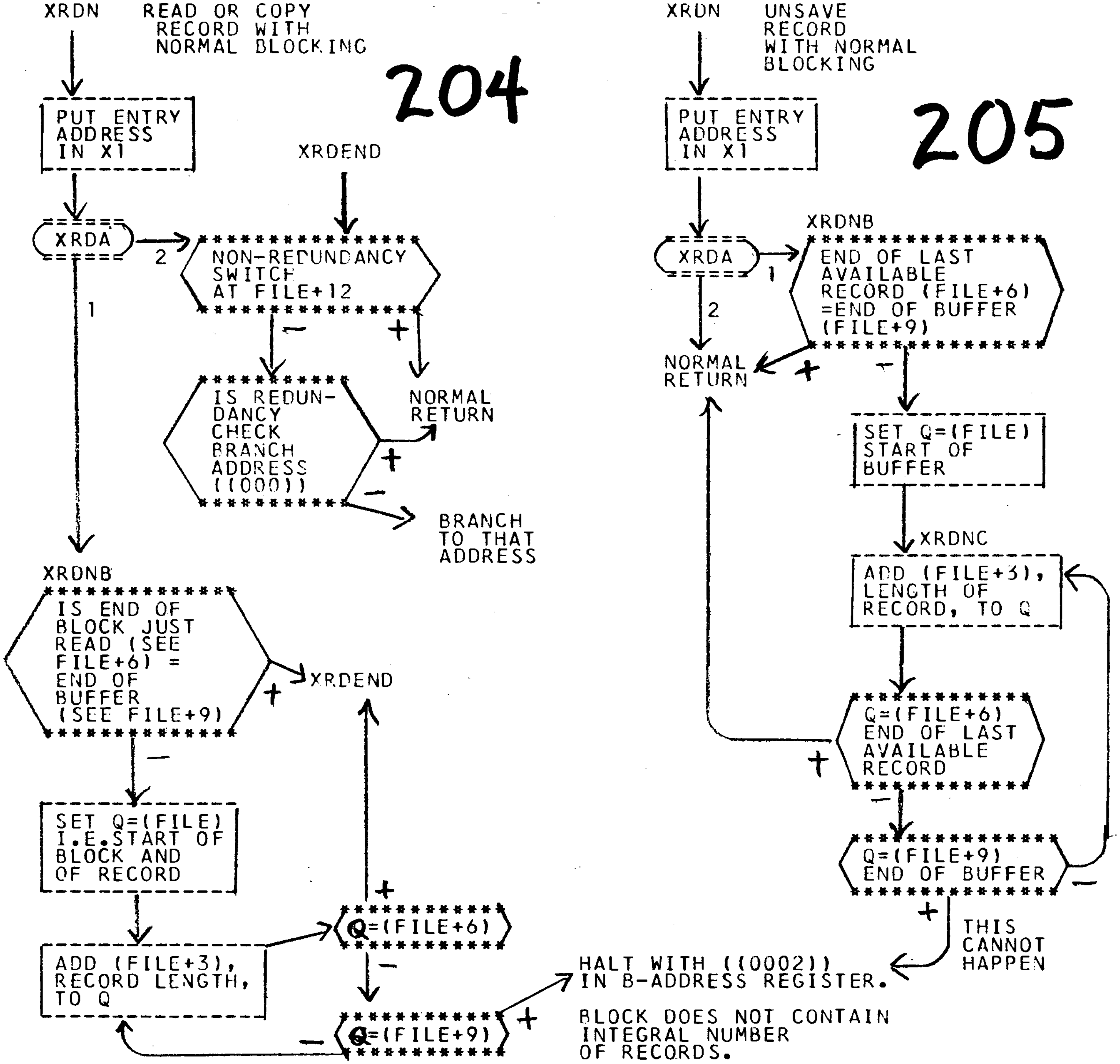
201

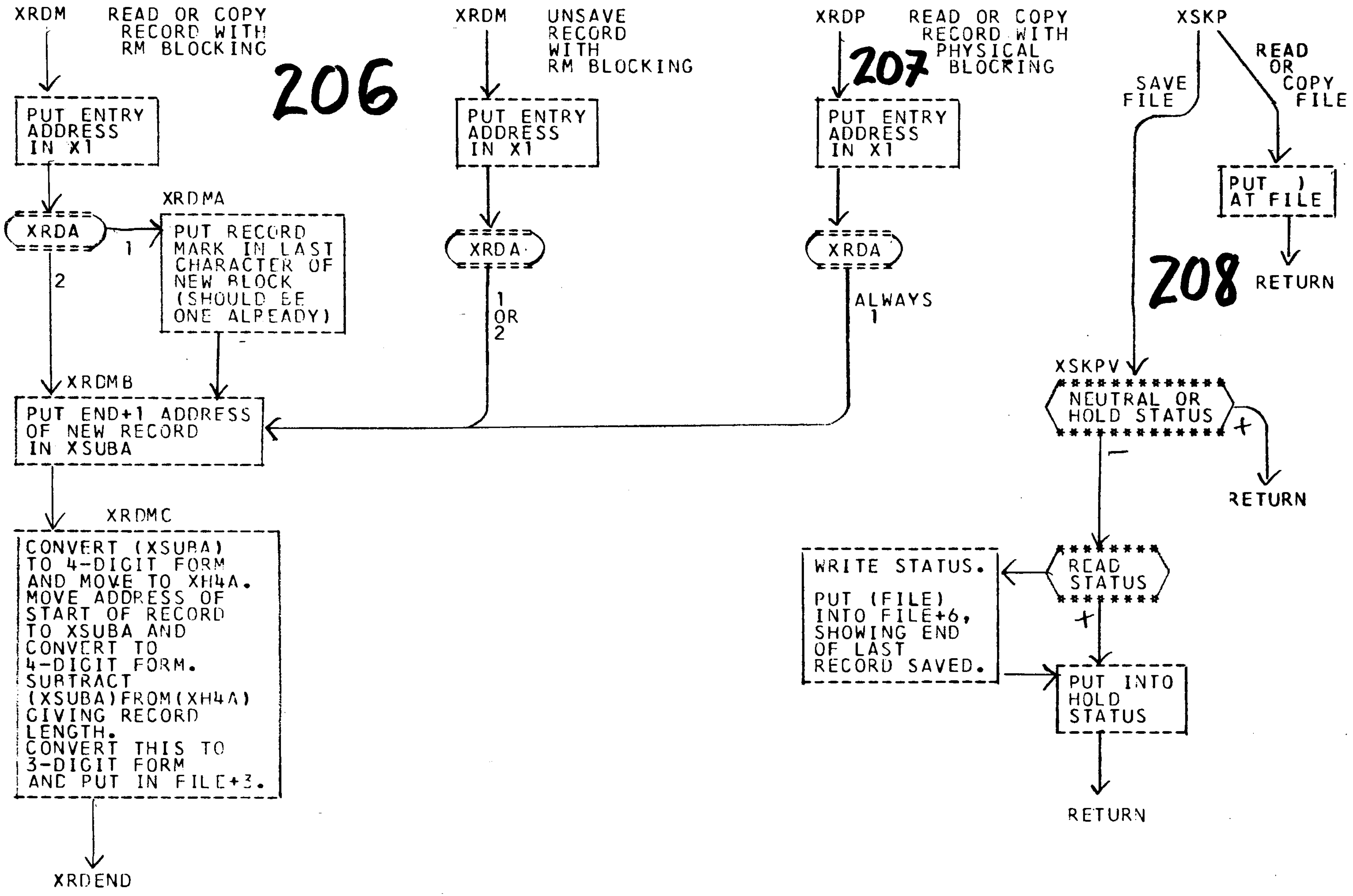

XRDA READ OR COPY, NOT UNSAVE
(FILE) = 00) INITIALLY
PUT FILE IN X2
(FILE) IS)
+
-
XRDAF
ADD (FILE+3), LENGTH OF LAST-READ RECORD, TO (FILE)
RESET SWITCH XFIRST
PUT (FILE) IN X3
RETURN 2
XRDAD
MOVE TAPE MODE AND NUMBER FROM FILE+11 TO READ ORDER AT XRDAB+3
MAKE B-ADDRESS OF READ ORDER FILE+13
(FILE)=(FILE+6) I.E. END+1 OF BLOCK
+
-
MOVE READ ORDER FROM XRDAB TO XCDAB,BUT LEAVE ((W)) AT XCDAB+7. GET OUTPUT TAPE NUMBER FROM (FILE+12) AND PUT IN XCDAB+3. PUT GROUP MARK-WORD MARK AT ADDRESS FROM FILE+6, I.E.END+1 OF BLOCK. EXECUTE WRITE ORDER AT XCDAB BY =XWOD=. REMOVE WORD MARK FROM GROUP MARK. PUT GROUP MARK-WORD MARK AT END+1 OF BUFFER,WHOSE ADDRESS IS IN FILE+9.
RETURN 1
PUT ADDRESS OF START OF BLOCK IN FILE AND IN X3
IS FILE COPY (ANY NUMERICAL BITS IN FILE+12)
+
-
XCDA
(FILE) IS)
-
+
XRDAE
RESET SWITCH FOR NON-REDUNDANCY AT FILE+12. SET COUNT OF READ ATTEMPTS TO 0, SET SWITCH XFIRST.
XRDAC
PUT ADDRESS OF END+1 OF BLOCK AT FILE+6
WAIT FOR MOUNTING OF NEXT REEL AND PRESSING OF START
XRDAB
READ BLOCK AND STORE B-ADDRESS.
INCREASE COUNT BY 1 AND BACK-SPACE TAPE
XRDAL
COUNT OF READ ATTEMPTS = 9
-
+
UNLOAD TAPE
TAPE MARK
-
+
REDUNDANCY CHECK
+
-
SET SWITCH AT FILE+12 SHOWING NON-REDUNDANCY
XRDAKA
REPEAT READ ORDER. TURN OFF REDUNDANCY CHECK IF ANY.
TAPE MARK
-
+
XRDAK
MULTI-REEL FILE I.E.PLUS ZONE ON (FILE+12)
+
-
XRDAKS
PUT) AT FILE
COPY FILE
+
-
W.T.M.TWICE ON OUTPUT TAPE
XRDAKU
BRANCH TO E.O.F.EXIT

202

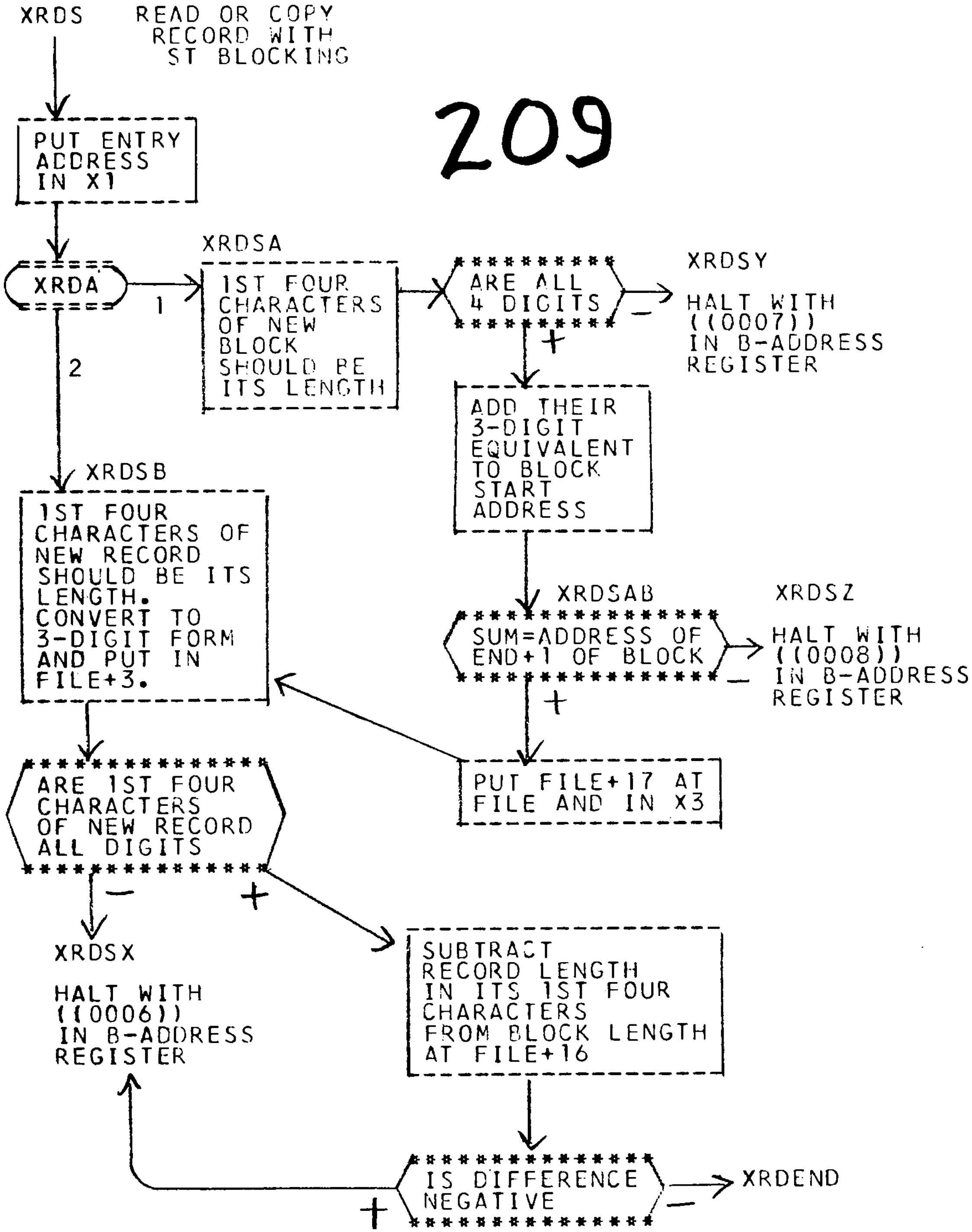
203


XRDA
UNSAVE
FILE IS INITIALLY IN NEUTRAL STATUS. CHARACTER AT FILE IS NEVER) .
READ STATUS
+
-
XRDAF
ADD (FILE+3), LENGTH OF LAST RECORD UNSAVED, TO (FILE)
(FILE)=(FILE+6) I.E. END+1 OF LAST AVAILABLE RECORD
-
+
RESET SWITCH XFIRST
RETURN 2
XRDAKT
PUT FILE IN HOLD STATUS
RDAKU
BRANCH TO E.O.F.EXIT
HOLD STATUS
-
+
NEUTRAL STATUS
+
-
XUSHI
PUT INTO READ STATUS
WRITE STATUS. MOVE (FILE), END+1 OF LAST RECORD SAVED, TO FILE+6.
XSAVD
SET (FILE) = START OF BUFFER
(FILE)=(FILE+6) I.E.END+1 OF LAST AVAILABLE RECORD
+
THIS SHOULD BE IMPOSSIBLE
-
XRDAY
PUT (FILE) IN X3
RETURN 1
XSAVD
IF ANY WRITE OR SAVE FILES WITH ST BLOCKING IN PROGRAM
SET (FILE)=FILE+25
IS BLOCKING ST
-
+
CONVERT (FILE) TO 4-DIGIT FORM AND PUT IN FILE+16. PUT ((0004)) IN FILE+24 (BLOCK LENGTH BEFORE ANY RECORD STORED.)
RETURN
XSAVD
IF NO WRITE OR SAVE FILES WITH ST BLOCKING IN PROGRAM
SET (FILE)=FILE+13

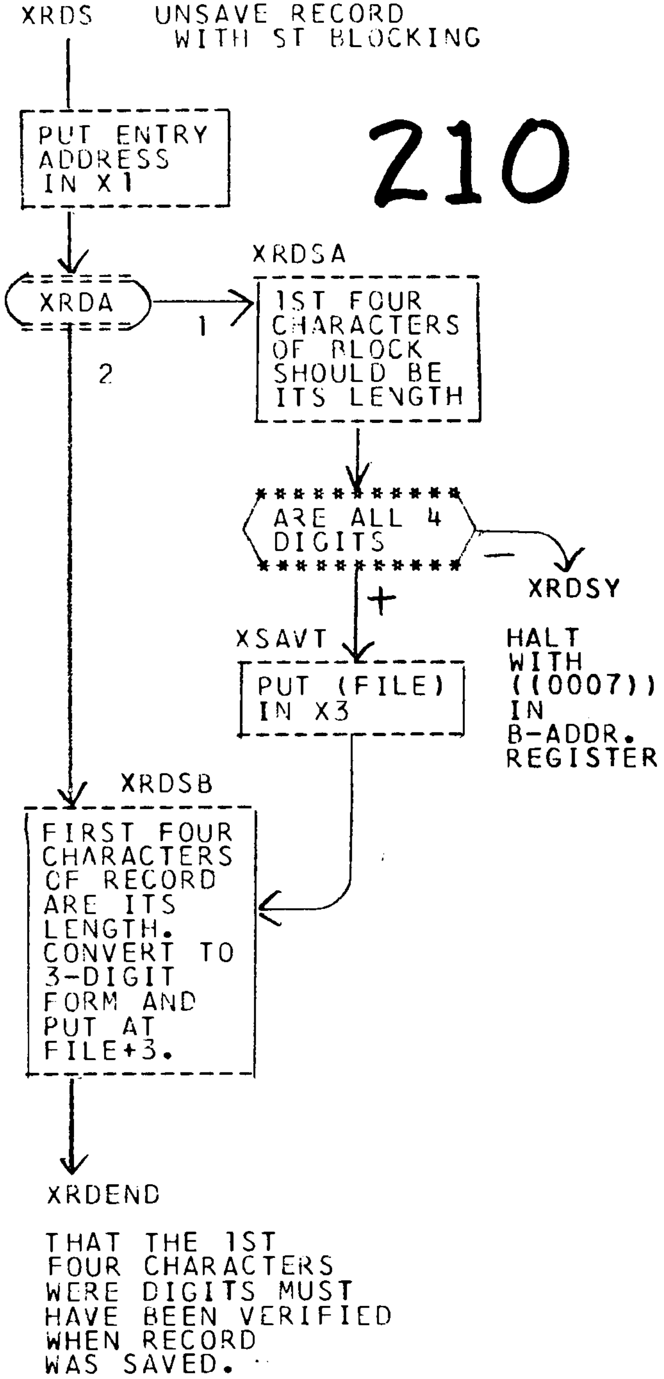

204
XRDN
READ OR COPY RECORD WITH NORMAL BLOCKING
PUT ENTRY ADDRESS IN X1
XRDA
1
2
XRDEND
NON-REDUNDANCY SWITCH AT FILE+12
-
+
IS REDUNDANCY CHECK BRANCH ADDRESS ((000))
NORMAL RETURN
+
-
BRANCH TO THAT ADDRESS
XRDNB
IS END OF BLOCK JUST READ (SEE FILE+6) = END OF BUFFER (SEE FILE+9)
+
XRDEND
-
SET Q=(FILE) I.E.START OF BLOCK AND OF RECORD
ADD (FILE+3), RECORD LENGTH, TO Q
+
Q=(FILE+6)
-
Q=(FILE+9)
-
+
HALT WITH ((0002)) IN B-ADDRESS REGISTER.
BLOCK DOES NOT CONTAIN INTEGRAL NUMBER OF RECORDS.
205
XRDN
UNSAVE RECORD WITH NORMAL BLOCKING
PUT ENTRY ADDRESS IN X1
XRDA
1
2
XRDNB
END OF LAST AVAILABLE RECORD (FILE+6) =END OF BUFFER (FILE+9)
NORMAL RETURN
+
-
SET Q=(FILE) START OF BUFFER
XRDNC
ADD (FILE+3), LENGTH OF RECORD, TO Q
Q=(FILE+6) END OF LAST AVAILABLE RECORD
+
-
Q=(FILE+9) END OF BUFFER
-
+
THIS CANNOT HAPPEN


XRDM
READ OR COPY RECORD WITH RM BLOCKING
206
PUT ENTRY ADDRESS IN X1
XRDA
1
2
XRDMA
PUT RECORD MARK IN LAST CHARACTER OF NEW BLOCK (SHOULD EE ONE ALPEADY)
XRDMB
PUT END+1 ADDRESS OF NEW RECORD IN XSUBA
XRDMC
CONVERT (XSUBA) TO 4-DIGIT FORM AND MOVE TO XH4A. MOVE ADDRESS OF START OF RECORD TO XSUBA AND CONVERT TO 4-DIGIT FORM. SUBTRACT (XSUBA) FROM (XH4A) GIVING RECORD LENGTH. CONVERT THIS TO 3-DIGIT FORM AND PUT IN FILE+3.
XRDEND
XRDM
UNSAVE RECORD WITH RM BLOCKING
PUT ENTRY ADDRESS IN X1
XRDA
1 OR 2
XRDP
READ OR COPY RECORD WITH PHYSICAL BLOCKING
207
PUT ENTRY ADDRESS IN X1
XRDA
ALWAYS 1
XSKP
SAVE FILE
READ OR COPY FILE
PUT) AT FILE
RETURN
208
XSKPV
NEUTRAL OR HOLD STATUS
+
RETURN
−
READ STATUS
WRITE STATUS.
PUT (FILE) INTO FILE+6, SHOWING END OF LAST RECORD SAVED.
+
PUT INTO HOLD STATUS
RETURN

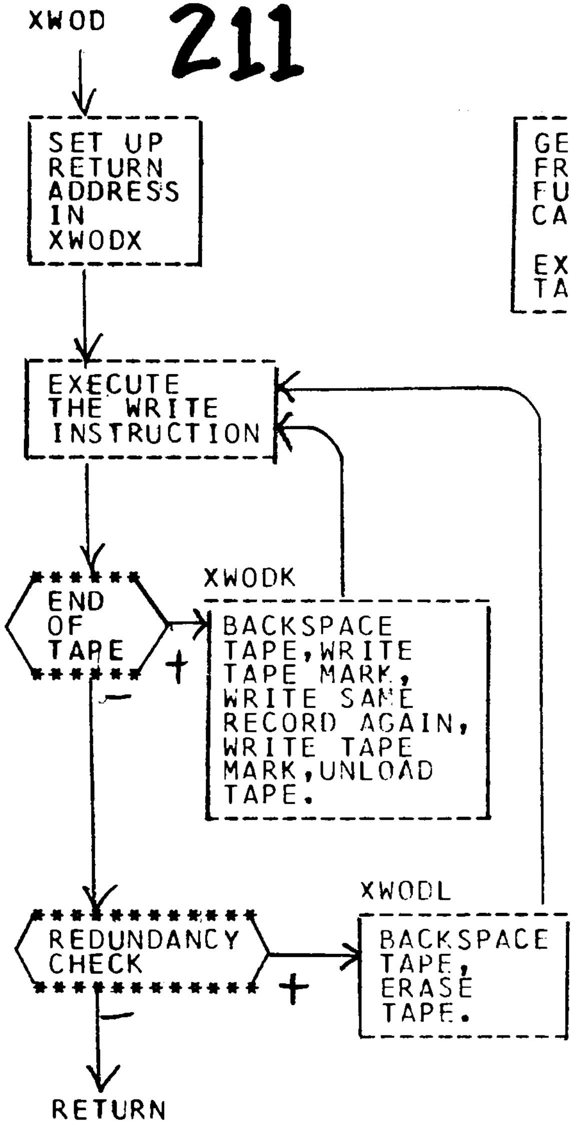
209


XRDS
READ OR COPY RECORD WITH ST BLOCKING
PUT ENTRY ADDRESS IN X1
XRDA
1
XRDSA
1ST FOUR CHARACTERS OF NEW BLOCK SHOULD BE ITS LENGTH
ARE ALL 4 DIGITS
−
XRDSY
HALT WITH ((0007)) IN B-ADDRESS REGISTER
+
ADD THEIR 3-DIGIT EQUIVALENT TO BLOCK START ADDRESS
XRDSAB
SUM=ADDRESS OF END+1 OF BLOCK
XRDSZ
HALT WITH ((0008)) IN B-ADDRESS REGISTER
−
+
PUT FILE+17 AT FILE AND IN X3
2
XRDSB
1ST FOUR CHARACTERS OF NEW RECORD SHOULD BE ITS LENGTH. CONVERT TO 3-DIGIT FORM AND PUT IN FILE+3.
ARE 1ST FOUR CHARACTERS OF NEW RECORD ALL DIGITS
−
+
XRDSX
HALT WITH ((0006)) IN B-ADDRESS REGISTER
SUBTRACT RECORD LENGTH IN ITS 1ST FOUR CHARACTERS FROM BLOCK LENGTH AT FILE+16
IS DIFFERENCE NEGATIVE
+
−
XRDEND

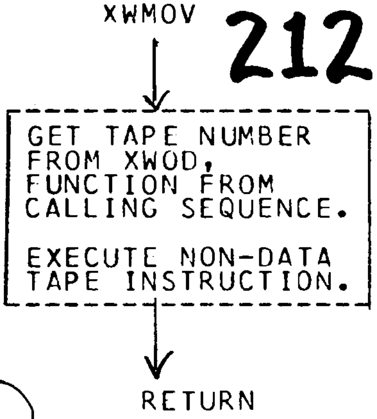
210


XRDS
UNSAVE RECORD WITH ST BLOCKING
PUT ENTRY ADDRESS IN X1
XRDA
1
XRDSA
1ST FOUR CHARACTERS OF BLOCK SHOULD BE ITS LENGTH
ARE ALL 4 DIGITS
−
XRDSY
HALT WITH ((0007)) IN B-ADDR. REGISTER
+
XSAVT
PUT (FILE) IN X3
2
XRDSB
FIRST FOUR CHARACTERS OF RECORD ARE ITS LENGTH. CONVERT TO 3-DIGIT FORM AND PUT AT FILE+3.
XRDEND
THAT THE 1ST FOUR CHARACTERS WERE DIGITS MUST HAVE BEEN VERIFIED WHEN RECORD WAS SAVED.

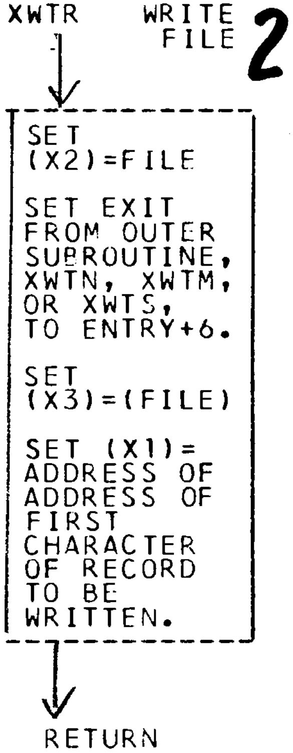
211


XWOD
SET UP RETURN ADDRESS IN XWODX
EXECUTE THE WRITE INSTRUCTION
END OF TAPE
+
−
XWODK
BACKSPACE TAPE, WRITE TAPE MARK, WRITE SAME RECORD AGAIN, WRITE TAPE MARK, UNLOAD TAPE.
REDUNDANCY CHECK
+
−
XWODL
BACKSPACE TAPE, ERASE TAPE.
RETURN

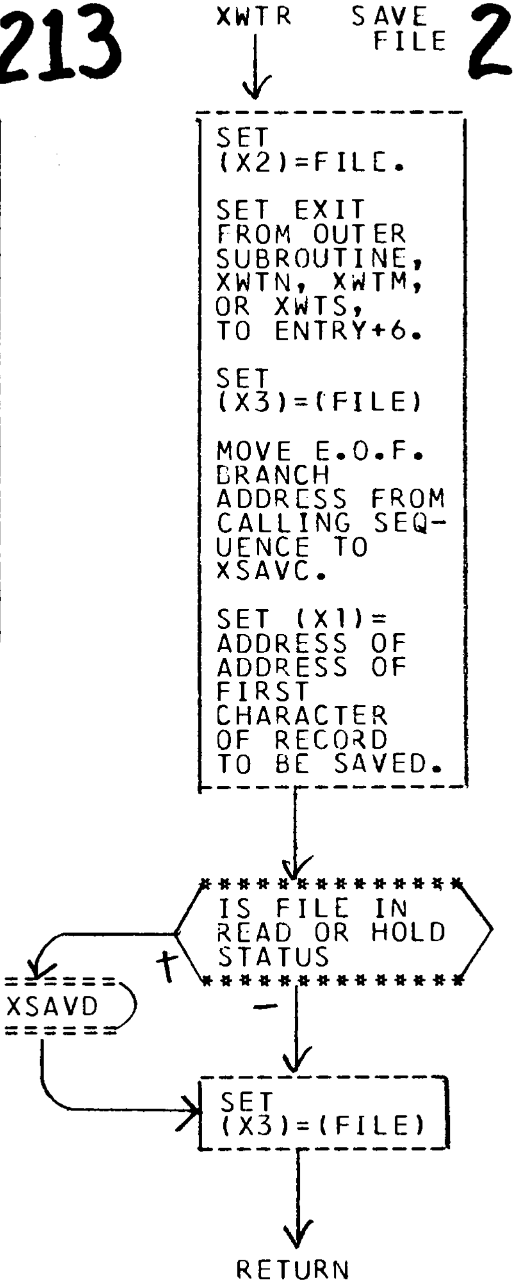
212

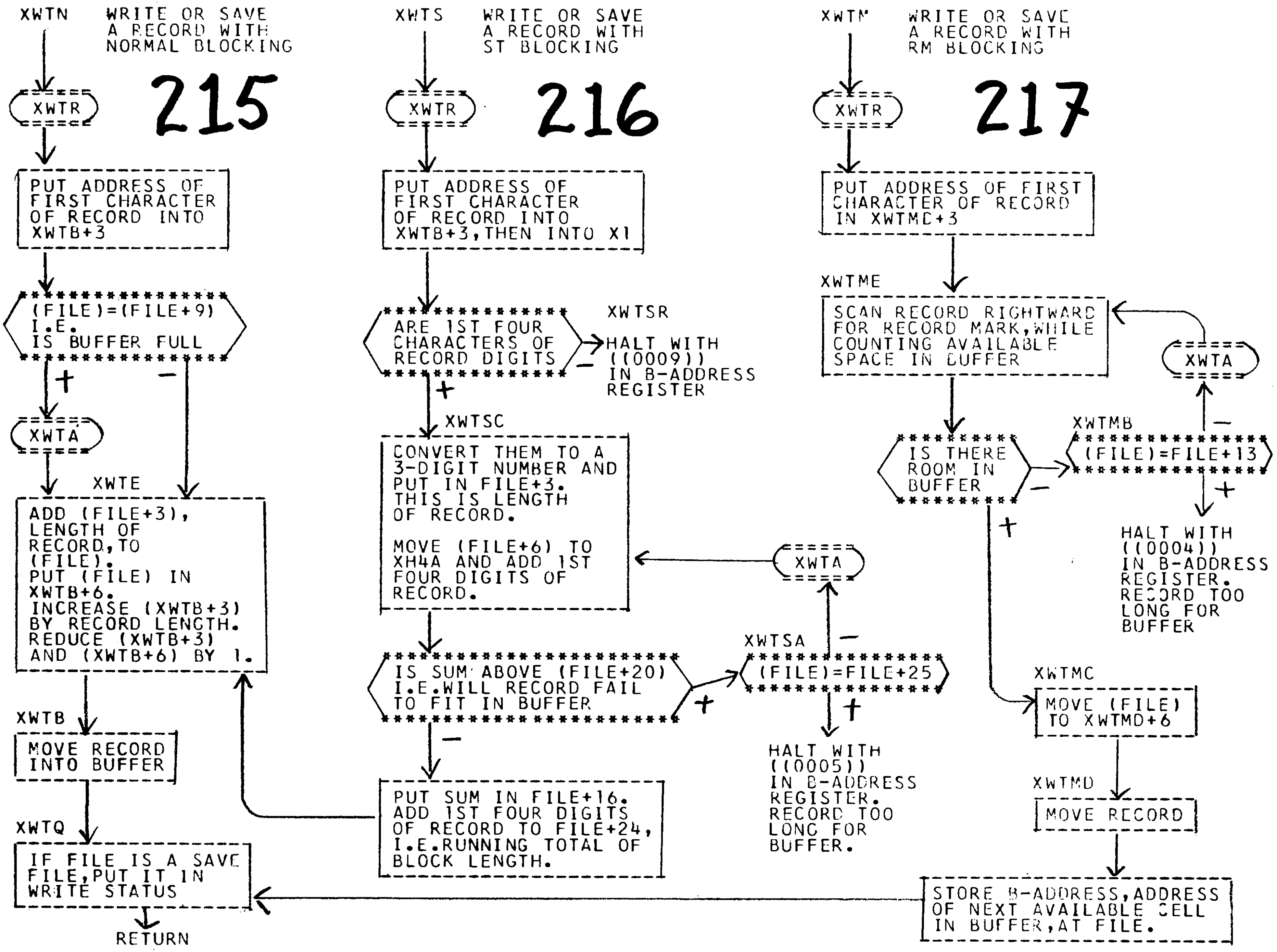

XWMOV
GET TAPE NUMBER FROM XWOD, FUNCTION FROM CALLING SEQUENCE.
EXECUTE NON-DATA TAPE INSTRUCTION.
RETURN

213


XWTR
WRITE FILE
SET (X2)=FILE
SET EXIT FROM OUTER SUBROUTINE, XWTN, XWTM, OR XWTS, TO ENTRY+6.
SET (X3)=(FILE)
SET (X1)= ADDRESS OF ADDRESS OF FIRST CHARACTER OF RECORD TO BE WRITTEN.
RETURN

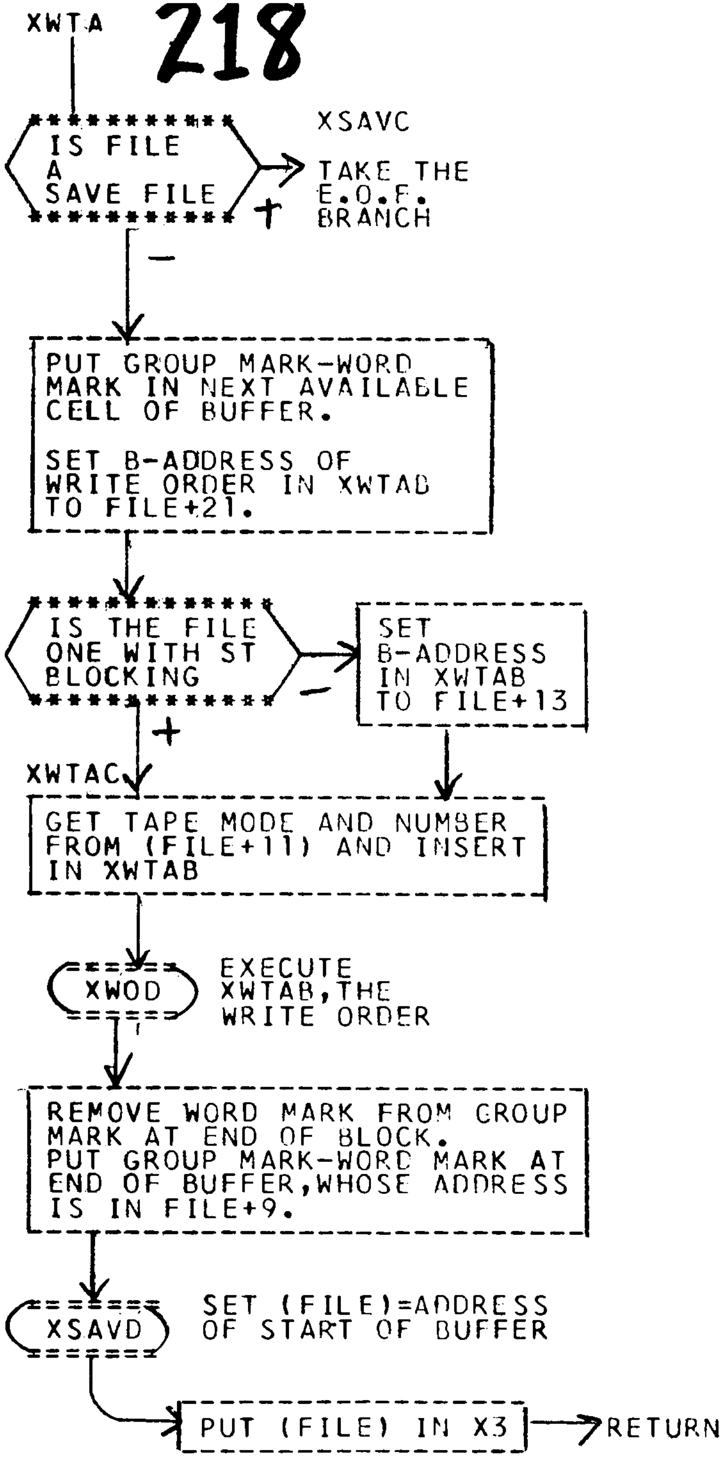
214


XWTR
SAVE FILE
SET (X2)=FILE.
SET EXIT FROM OUTER SUBROUTINE, XWTN, XWTM, OR XWTS, TO ENTRY+6.
SET (X3)=(FILE)
MOVE E.O.F. BRANCH ADDRESS FROM CALLING SEQUENCE TO XSAVC.
SET (X1)= ADDRESS OF ADDRESS OF FIRST CHARACTER OF RECORD TO BE SAVED.
IS FILE IN READ OR HOLD STATUS
+
−
XSAVD
SET (X3)=(FILE)
RETURN

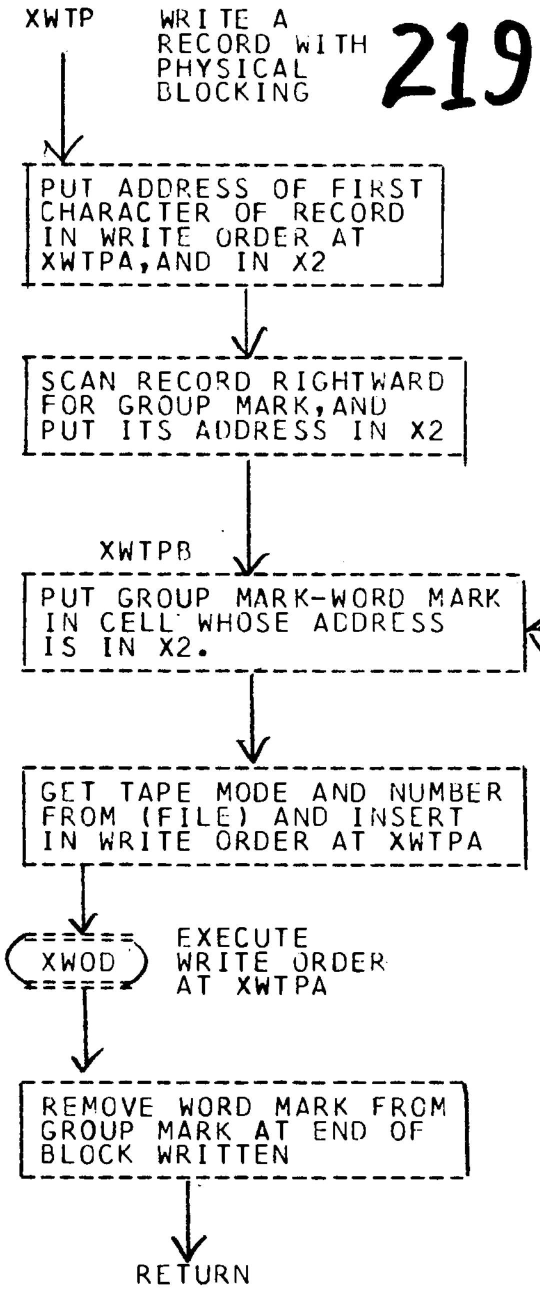


XWTN
WRITE OR SAVE A RECORD WITH NORMAL BLOCKING
215
XWTR
PUT ADDRESS OF FIRST CHARACTER OF RECORD INTO XWTB+3
(FILE)=(FILE+9) I.E. IS BUFFER FULL
+
-
XWTA
XWTE
ADD (FILE+3), LENGTH OF RECORD, TO (FILE). PUT (FILE) IN XWTB+6. INCREASE (XWTB+3) BY RECORD LENGTH. REDUCE (XWTB+3) AND (XWTB+6) BY 1.
XWTB
MOVE RECORD INTO BUFFER
XWTQ
IF FILE IS A SAVE FILE, PUT IT IN WRITE STATUS
RETURN
XWTS
WRITE OR SAVE A RECORD WITH ST BLOCKING
216
XWTR
PUT ADDRESS OF FIRST CHARACTER OF RECORD INTO XWTB+3, THEN INTO X1
ARE 1ST FOUR CHARACTERS OF RECORD DIGITS
-
+
XWTSR
HALT WITH ((0009)) IN B-ADDRESS REGISTER
XWTSC
CONVERT THEM TO A 3-DIGIT NUMBER AND PUT IN FILE+3. THIS IS LENGTH OF RECORD.
MOVE (FILE+6) TO XH4A AND ADD 1ST FOUR DIGITS OF RECORD.
XWTA
IS SUM ABOVE (FILE+20) I.E. WILL RECORD FAIL TO FIT IN BUFFER
+
-
XWTSA
(FILE)=FILE+25
-
+
HALT WITH ((0005)) IN B-ADDRESS REGISTER. RECORD TOO LONG FOR BUFFER.
PUT SUM IN FILE+16. ADD 1ST FOUR DIGITS OF RECORD TO FILE+24, I.E. RUNNING TOTAL OF BLOCK LENGTH.
XWTM
WRITE OR SAVE A RECORD WITH RM BLOCKING
217
XWTR
PUT ADDRESS OF FIRST CHARACTER OF RECORD IN XWTMC+3
XWTME
SCAN RECORD RIGHTWARD FOR RECORD MARK, WHILE COUNTING AVAILABLE SPACE IN BUFFER
XWTA
IS THERE ROOM IN BUFFER
-
+
XWTMB
(FILE)=FILE+13
-
+
HALT WITH ((0004)) IN B-ADDRESS REGISTER. RECORD TOO LONG FOR BUFFER
XWTMC
MOVE (FILE) TO XWTMD+6
XWTMD
MOVE RECORD
STORE B-ADDRESS, ADDRESS OF NEXT AVAILABLE CELL IN BUFFER, AT FILE.

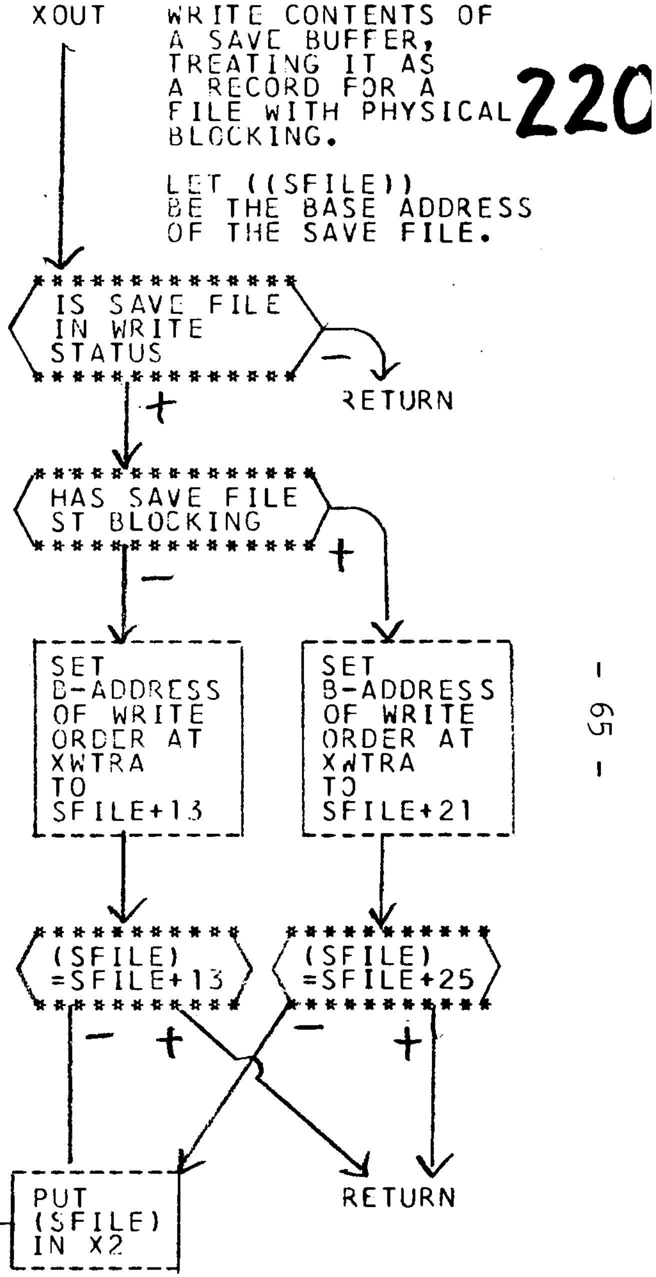
218

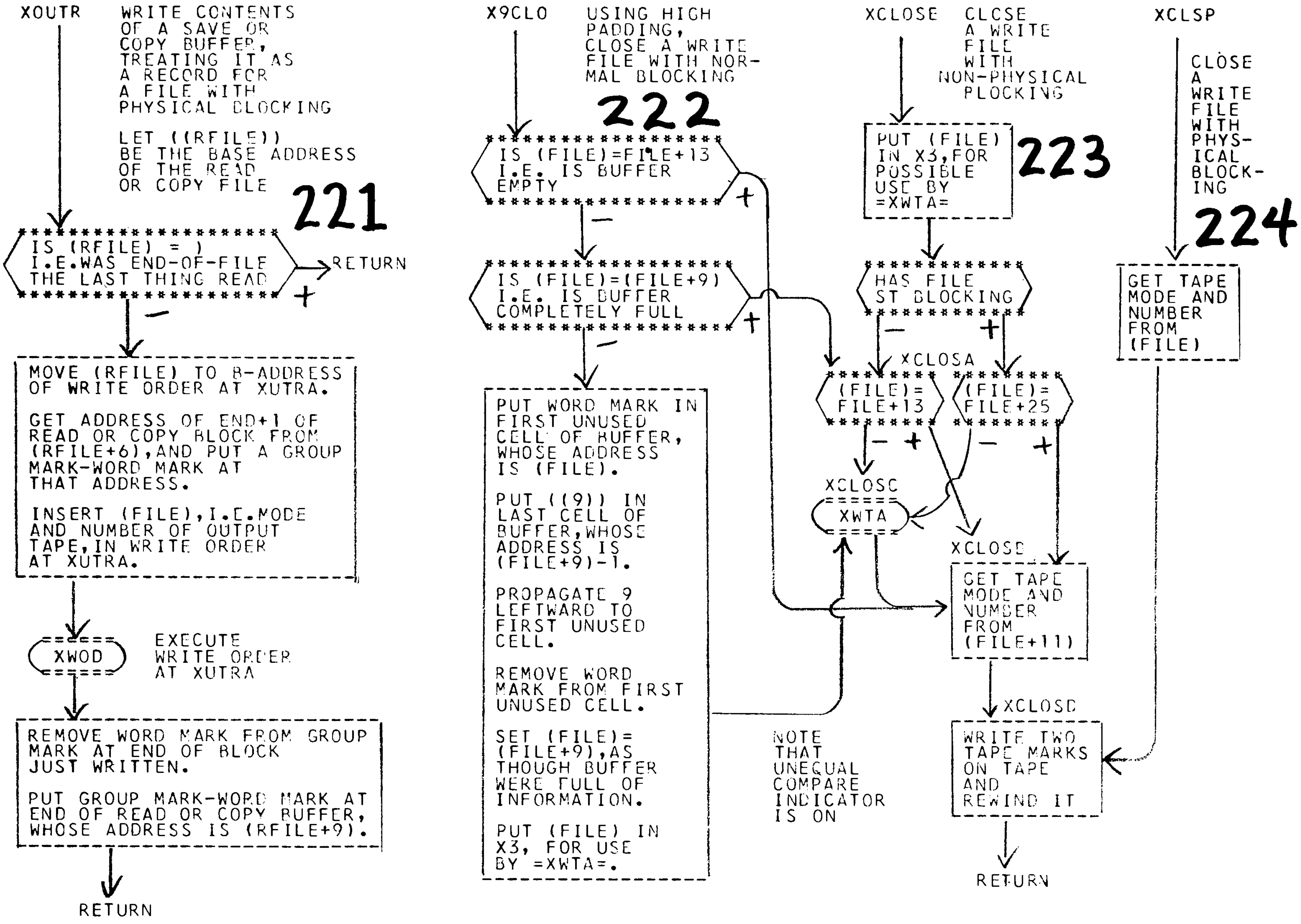

XWTA
IS FILE A SAVE FILE
+
XSAVC
TAKE THE E.O.F. BRANCH
−
PUT GROUP MARK-WORD MARK IN NEXT AVAILABLE CELL OF BUFFER.
SET B-ADDRESS OF WRITE ORDER IN XWTAB TO FILE+21.
IS THE FILE ONE WITH ST BLOCKING
−
SET B-ADDRESS IN XWTAB TO FILE+13
+
XWTAC
GET TAPE MODE AND NUMBER FROM (FILE+11) AND INSERT IN XWTAB
XWOD
EXECUTE XWTAB,THE WRITE ORDER
REMOVE WORD MARK FROM GROUP MARK AT END OF BLOCK. PUT GROUP MARK-WORD MARK AT END OF BUFFER,WHOSE ADDRESS IS IN FILE+9.
XSAVD
SET (FILE)=ADDRESS OF START OF BUFFER
PUT (FILE) IN X3
RETURN

219


XWTP
WRITE A RECORD WITH PHYSICAL BLOCKING
PUT ADDRESS OF FIRST CHARACTER OF RECORD IN WRITE ORDER AT XWTPA,AND IN X2
SCAN RECORD RIGHTWARD FOR GROUP MARK,AND PUT ITS ADDRESS IN X2
XWTPB
PUT GROUP MARK-WORD MARK IN CELL WHOSE ADDRESS IS IN X2.
GET TAPE MODE AND NUMBER FROM (FILE) AND INSERT IN WRITE ORDER AT XWTPA
XWOD
EXECUTE WRITE ORDER AT XWTPA
REMOVE WORD MARK FROM GROUP MARK AT END OF BLOCK WRITTEN
RETURN

220

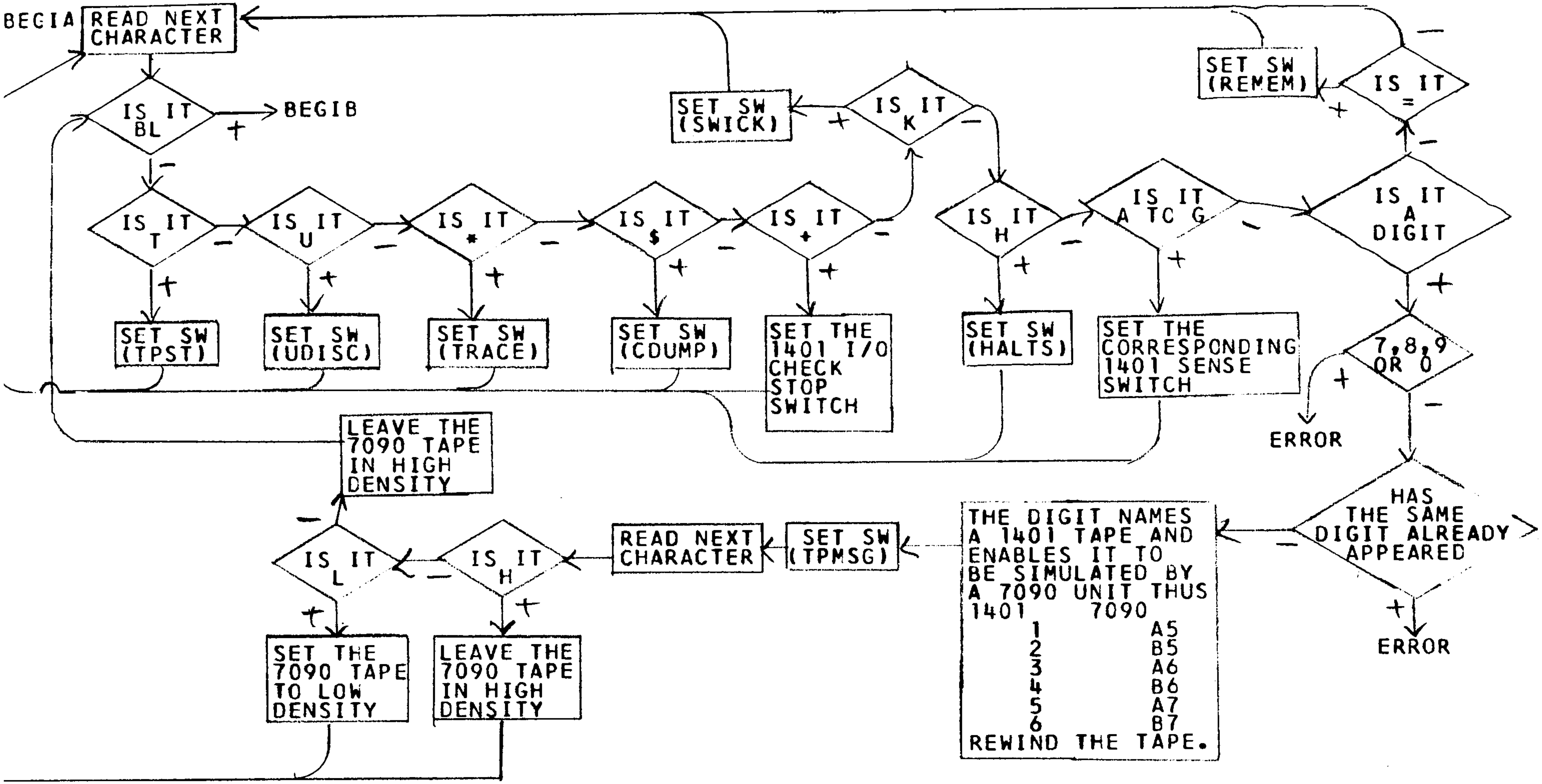

XOUT
WRITE CONTENTS OF A SAVE BUFFER, TREATING IT AS A RECORD FOR A FILE WITH PHYSICAL BLOCKING.
LET ((SFILE)) BE THE BASE ADDRESS OF THE SAVE FILE.
IS SAVE FILE IN WRITE STATUS
−
RETURN
+
HAS SAVE FILE ST BLOCKING
−
+
SET B-ADDRESS OF WRITE ORDER AT XWTRA TO SFILE+13
SET B-ADDRESS OF WRITE ORDER AT XWTRA TO SFILE+21
(SFILE) =SFILE+13
(SFILE) =SFILE+25
−
+
−
+
PUT (SFILE) IN X2
RETURN



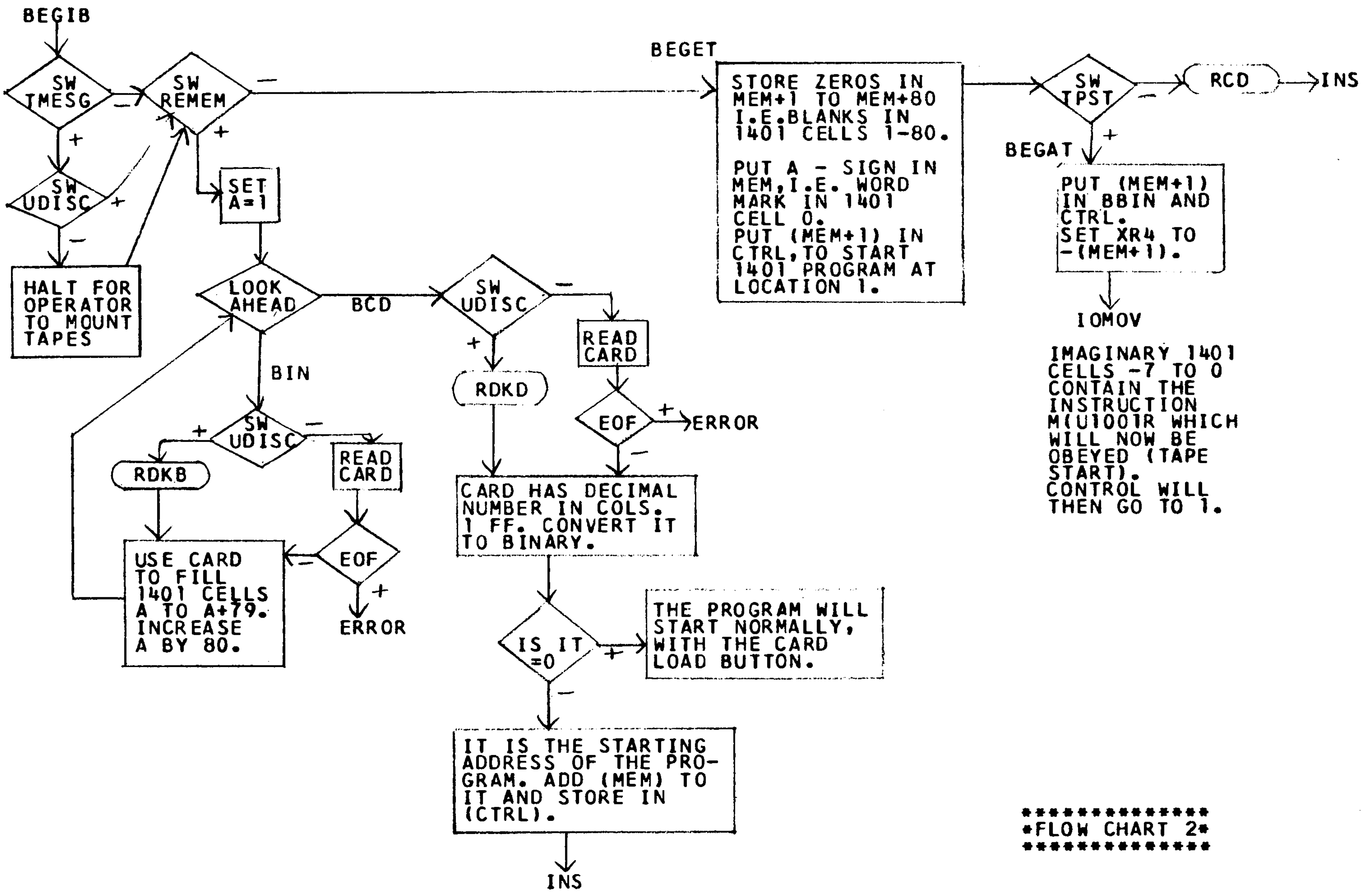
XOUTR
WRITE CONTENTS OF A SAVE OR COPY BUFFER, TREATING IT AS A RECORD FOR A FILE WITH PHYSICAL BLOCKING
LET ((RFILE)) BE THE BASE ADDRESS OF THE READ OR COPY FILE
221
IS (RFILE) =) I.E.WAS END-OF-FILE THE LAST THING READ
RETURN
+
−
MOVE (RFILE) TO B-ADDRESS OF WRITE ORDER AT XUTRA.
GET ADDRESS OF END+1 OF READ OR COPY BLOCK FROM (RFILE+6),AND PUT A GROUP MARK-WORD MARK AT THAT ADDRESS.
INSERT (FILE),I.E.MODE AND NUMBER OF OUTPUT TAPE,IN WRITE ORDER AT XUTRA.
XWOD
EXECUTE WRITE ORDER AT XUTRA
REMOVE WORD MARK FROM GROUP MARK AT END OF BLOCK JUST WRITTEN.
PUT GROUP MARK-WORD MARK AT END OF READ OR COPY BUFFER, WHOSE ADDRESS IS (RFILE+9).
RETURN
X9CLO
USING HIGH PADDING, CLOSE A WRITE FILE WITH NORMAL BLOCKING
222
IS (FILE)=FILE+13 I.E. IS BUFFER EMPTY
+
−
IS (FILE)=(FILE+9) I.E. IS BUFFER COMPLETELY FULL
+
−
PUT WORD MARK IN FIRST UNUSED CELL OF BUFFER, WHOSE ADDRESS IS (FILE).
PUT ((9)) IN LAST CELL OF BUFFER,WHOSE ADDRESS IS (FILE+9)-1.
PROPAGATE 9 LEFTWARD TO FIRST UNUSED CELL.
REMOVE WORD MARK FROM FIRST UNUSED CELL.
SET (FILE)= (FILE+9),AS THOUGH BUFFER WERE FULL OF INFORMATION.
PUT (FILE) IN X3, FOR USE BY =XWTA=.
XCLOSE
CLOSE A WRITE FILE WITH NON-PHYSICAL BLOCKING
223
PUT (FILE) IN X3,FOR POSSIBLE USE BY =XWTA=
HAS FILE ST BLOCKING
−
+
XCLOSA
(FILE)= FILE+13
− +
(FILE)= FILE+25
− +
XCLOSC
XWTA
XCLOSD
GET TAPE MODE AND NUMBER FROM (FILE+11)
XCLOSD
WRITE TWO TAPE MARKS ON TAPE AND REWIND IT
RETURN
NOTE THAT UNEQUAL COMPARE INDICATOR IS ON
XCLSP
CLOSE A WRITE FILE WITH PHYSICAL BLOCKING
224
GET TAPE MODE AND NUMBER FROM (FILE)

The Flow Charts of the Simulator

The following pages contain the flow charts of the 7090 program which simulates the 1401 operations. Comments on these flow charts are given on pages 1-49 of the third volume.

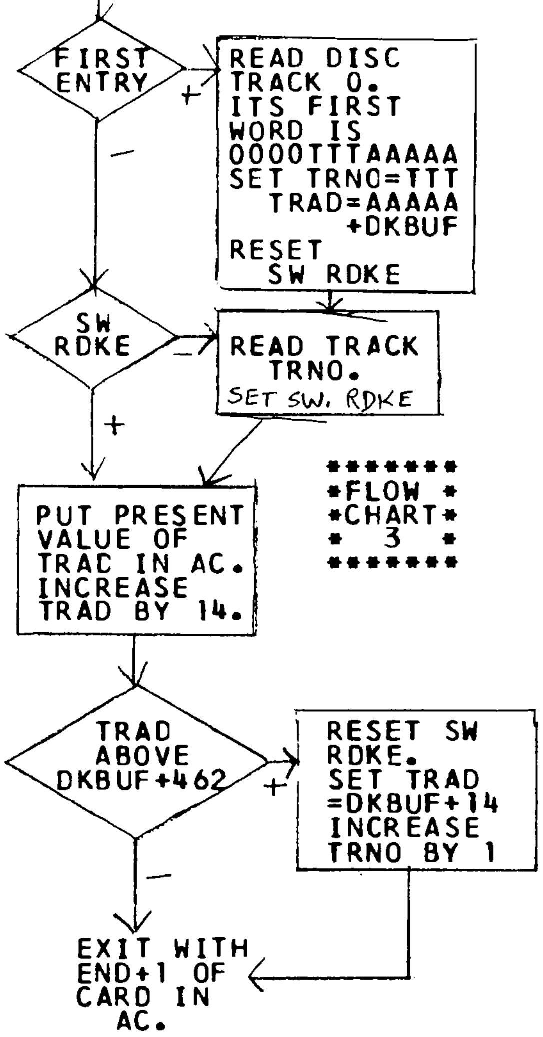

BEGIA
READ NEXT CHARACTER
IS IT BL
+
BEGIB
-
IS IT T
+
SET SW (TPST)
-
IS IT U
+
SET SW (UDISC)
-
IS IT *
+
SET SW (TRACE)
-
IS IT $
+
SET SW (CDUMP)
-
IS IT +
+
SET THE 1401 I/O CHECK STOP SWITCH
-
IS IT K
+
SET SW (SWICK)
-
IS IT H
+
SET SW (HALTS)
-
IS IT A TC G
+
SET THE CORRESPONDING 1401 SENSE SWITCH
-
IS IT A DIGIT
-
IS IT =
+
SET SW (REMEM)
-
+
7,8,9 OR 0
+
ERROR
-
HAS THE SAME DIGIT ALREADY APPEARED
+
ERROR
-
THE DIGIT NAMES A 1401 TAPE AND ENABLES IT TO BE SIMULATED BY A 7090 UNIT THUS
1401 7090
1 A5
2 B5
3 A6
4 B6
5 A7
6 B7
REWIND THE TAPE.
SET SW (TPMSG)
READ NEXT CHARACTER
IS IT H
+
LEAVE THE 7090 TAPE IN HIGH DENSITY
-
IS IT L
+
SET THE 7090 TAPE TO LOW DENSITY
-
LEAVE THE 7090 TAPE IN HIGH DENSITY

****
FLOW CHART 1
****


BEGIB
SW TMESG
SW UDISC
HALT FOR OPERATOR TO MOUNT TAPES
SW REMEM
SET A=1
LOOK AHEAD
BCD
BIN
SW UDISC
RDKB
READ CARD
EOF
ERROR
USE CARD TO FILL 1401 CELLS A TO A+79. INCREASE A BY 80.
SW UDISC
RDKD
READ CARD
EOF
ERROR
CARD HAS DECIMAL NUMBER IN COLS. 1 FF. CONVERT IT TO BINARY.
IS IT =0
THE PROGRAM WILL START NORMALLY, WITH THE CARD LOAD BUTTON.
IT IS THE STARTING ADDRESS OF THE PROGRAM. ADD (MEM) TO IT AND STORE IN (CTRL).
INS
BEGET
STORE ZEROS IN MEM+1 TO MEM+80 I.E.BLANKS IN 1401 CELLS 1-80.
PUT A - SIGN IN MEM,I.E. WORD MARK IN 1401 CELL 0. PUT (MEM+1) IN CTRL,TO START 1401 PROGRAM AT LOCATION 1.
SW TPST
RCD
INS
BEGAT
PUT (MEM+1) IN BBIN AND CTRL. SET XR4 TO -(MEM+1).
IOMOV
IMAGINARY 1401 CELLS -7 TO 0 CONTAIN THE INSTRUCTION M(U1001R WHICH WILL NOW BE OBEYED (TAPE START). CONTROL WILL THEN GO TO 1.
+
-

****
FLOW CHART 2
****

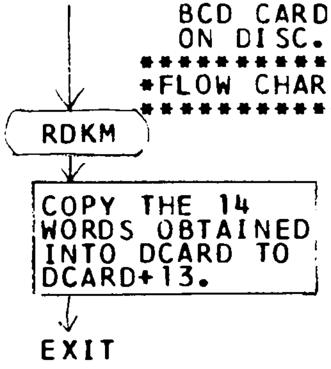
RDKM - GET NEXT 14 WORDS FROM DISC.


FIRST ENTRY
+
−
READ DISC TRACK 0. ITS FIRST WORD IS 0000TTTAAAAA SET TRNO=TTT TRAD=AAAAA +DKBUF RESET SW RDKE
SW RDKE
−
READ TRACK TRNO. SET SW. RDKE
+
PUT PRESENT VALUE OF TRAD IN AC. INCREASE TRAD BY 14.
TRAD ABOVE DKBUF+462
+
RESET SW RDKE. SET TRAD =DKBUF+14 INCREASE TRNO BY 1
−
EXIT WITH END+1 OF CARD IN AC.

****
*FLOW *
CHART
* 3 *
****

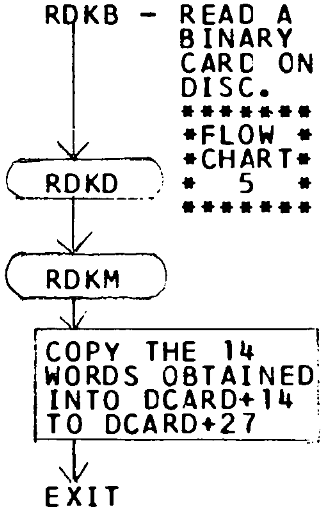
RDKD - READ A BCD CARD ON DISC.

****
FLOW CHART 4
****



RDKB - READ A BINARY CARD ON DISC.

****
*FLOW *
CHART
* 5 *
****

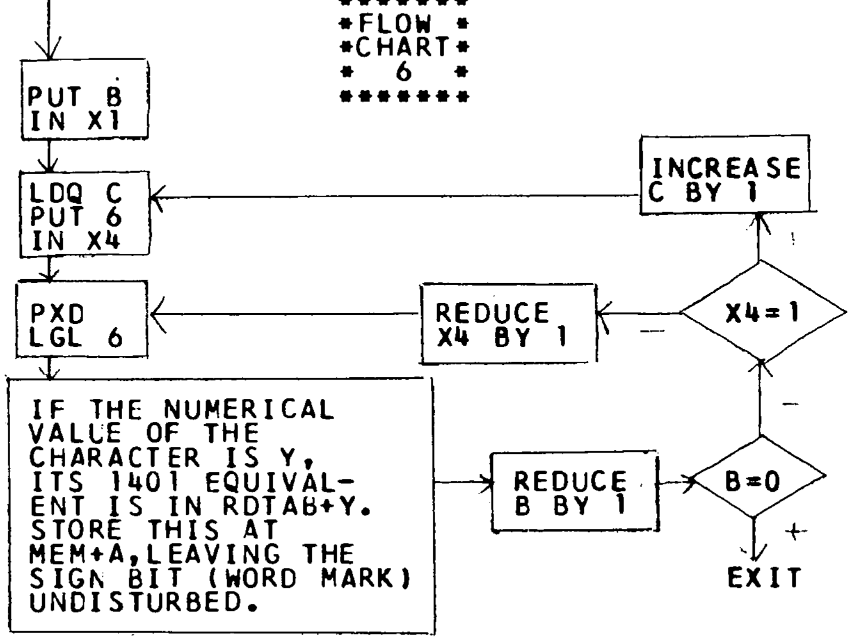


NOW 28 CONSECUTIVE WORDS HAVE BEEN COPIED INTO DCARD TO DCARD+27.

```
NPD -TSX NPD,4
     PZE MEM+A,,B
     PZE C
```

A BCD RECORD BEGINS AT THE LEFTMOST CHARACTER OF 7090 CELL C, AND IS B CHARACTERS LONG. STORE THE CHARACTERS IN 1401 FORM IN 1401 LOCATICNS A TO A+B-1, NOT DISTURBING WORD MARKS.

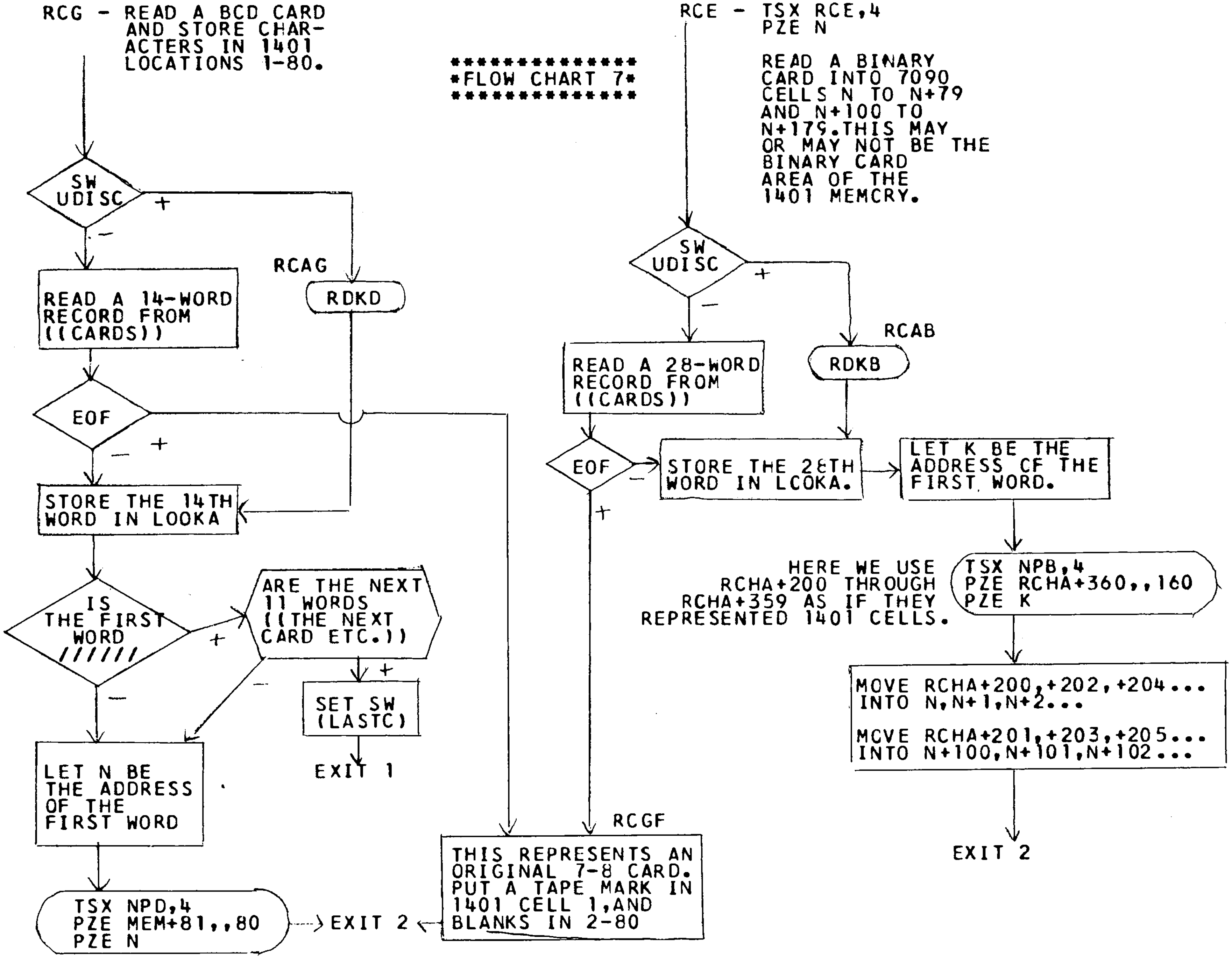
****
*FLOW *
CHART
* 6 *
****


PUT B IN X1
LDQ C PUT 6 IN X4
PXD LGL 6
IF THE NUMERICAL VALUE OF THE CHARACTER IS Y, ITS 1401 EQUIVALENT IS IN RDTAB+Y. STORE THIS AT MEM+A,LEAVING THE SIGN BIT (WORD MARK) UNDISTURBED.
REDUCE B BY 1
B=0
+
EXIT
−
X4=1
−
REDUCE X4 BY 1
INCREASE C BY 1

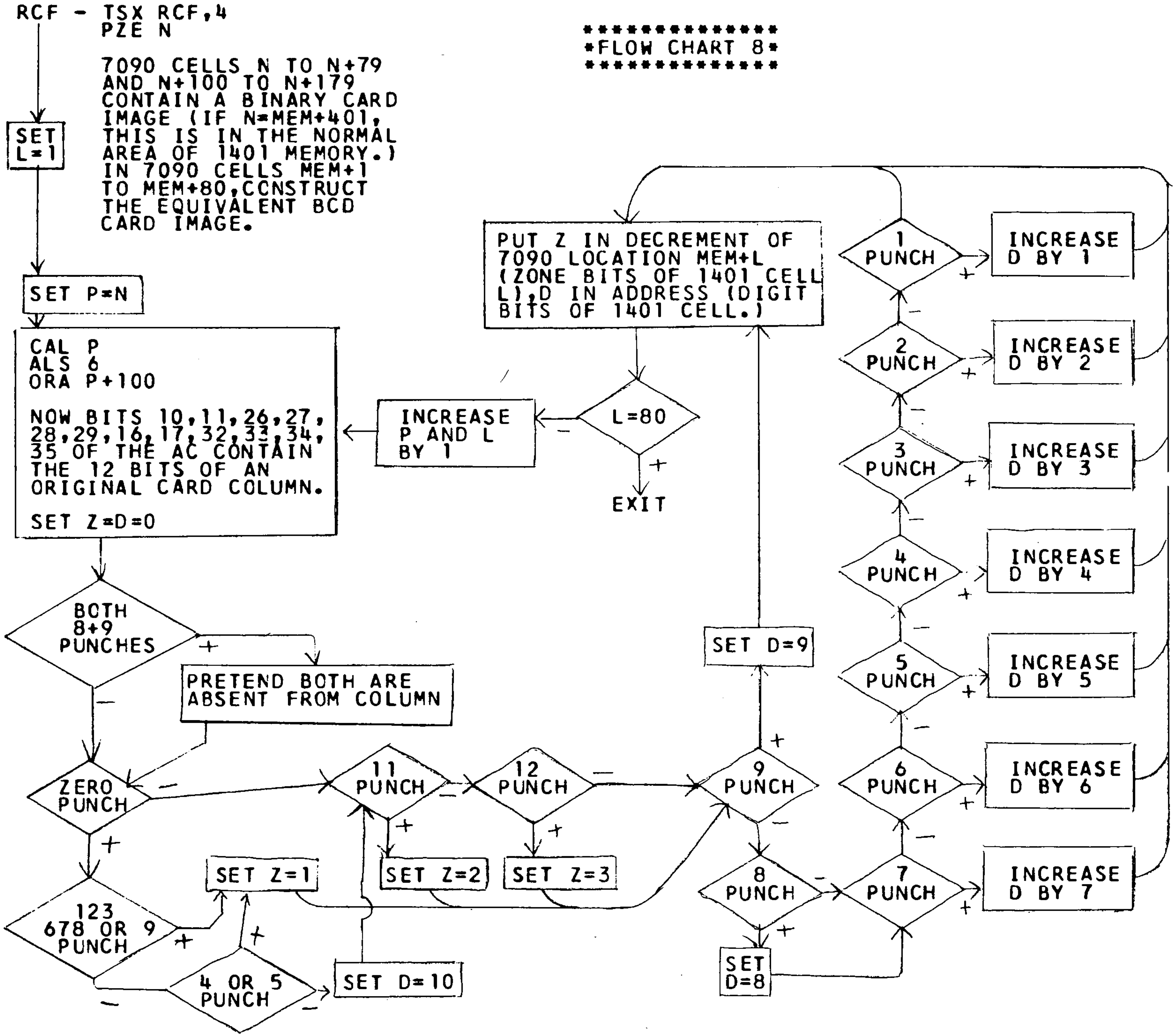
NPB IS THE SAME AS NPD,EXCEPT THAT THE RECORD WAS READ FROM 7090 TAPE IN THE BINARY MODE. THE CONVERSION TABLE BEGINNING AT RCTAB IS NOT NEEDED. THEREFORE THIS CHANGE --

```
    BCD                BINARY
 PXD                PXD
 LGL 6              LGL 2
 PAC ,2             ALS 14
 CAL RDTAB,2        LGL 4
```

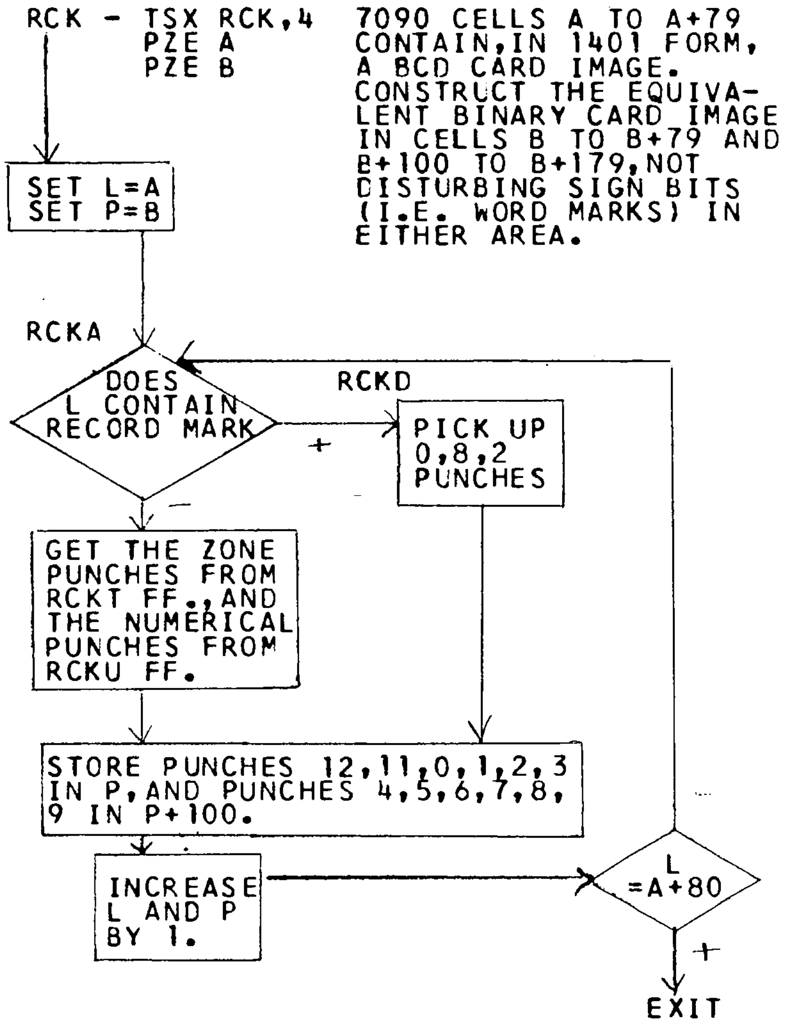
FLOW CHART 7


RCG - READ A BCD CARD AND STORE CHARACTERS IN 1401 LOCATIONS 1-80.
SW UDISC
+
-
READ A 14-WORD RECORD FROM ((CARDS))
RCAG
RDKD
EOF
-
+
STORE THE 14TH WORD IN LOOKA
IS THE FIRST WORD //////
+
-
ARE THE NEXT 11 WORDS ((THE NEXT CARD ETC.))
+
-
SET SW (LASTC)
EXIT 1
LET N BE THE ADDRESS OF THE FIRST WORD
TSX NPD,4
PZE MEM+81,,80
PZE N
EXIT 2
THIS REPRESENTS AN ORIGINAL 7-8 CARD. PUT A TAPE MARK IN 1401 CELL 1,AND BLANKS IN 2-80
RCGF
RCE - TSX RCE,4
PZE N
READ A BINARY CARD INTO 7090 CELLS N TO N+79 AND N+100 TO N+179.THIS MAY OR MAY NOT BE THE BINARY CARD AREA OF THE 1401 MEMCRY.
SW UDISC
+
-
READ A 28-WORD RECORD FROM ((CARDS))
RCAB
RDKB
EOF
-
+
STORE THE 28TH WORD IN LCOKA.
LET K BE THE ADDRESS CF THE FIRST WORD.
HERE WE USE RCHA+200 THROUGH RCHA+359 AS IF THEY REPRESENTED 1401 CELLS.
TSX NPB,4
PZE RCHA+360,,160
PZE K
MOVE RCHA+200,+202,+204...
INTO N,N+1,N+2...
MCVE RCHA+201,+203,+205...
INTO N+100,N+101,N+102...
EXIT 2

FLOW CHART 8

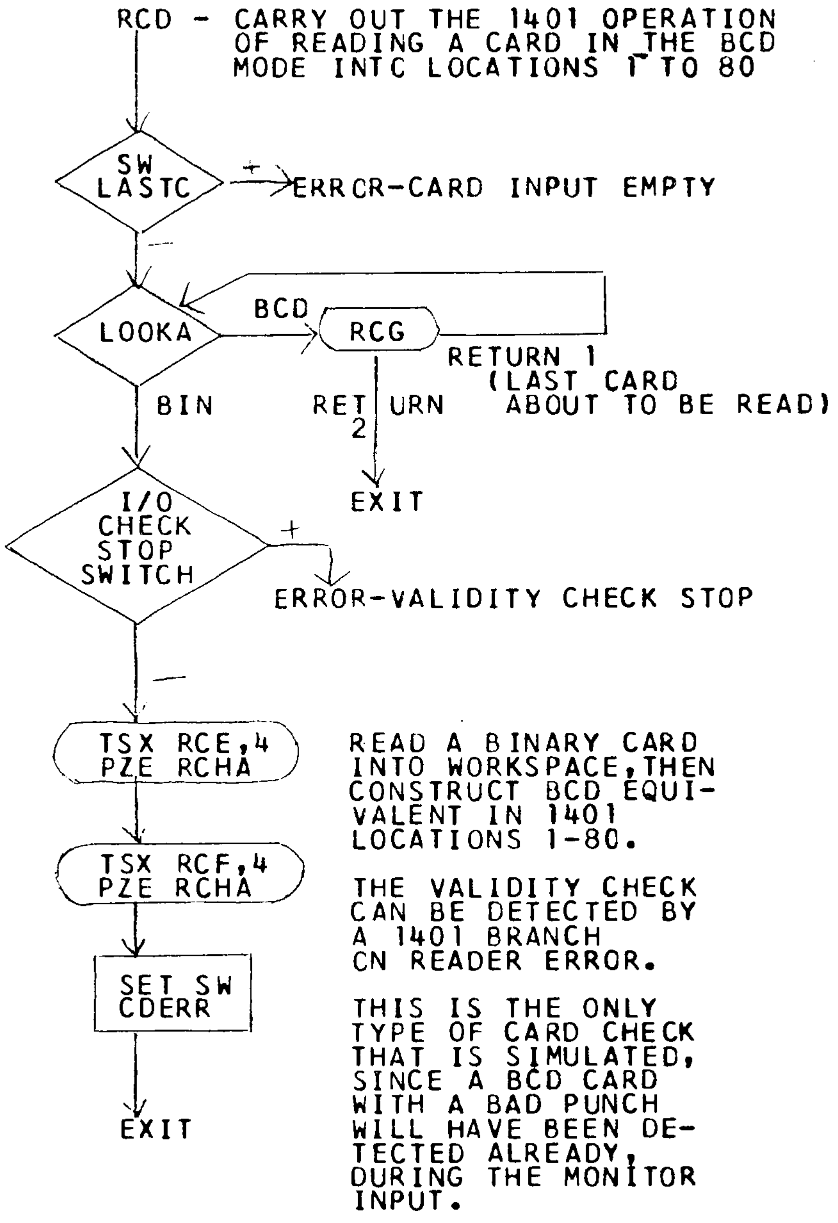


RCF - TSX RCF,4
PZE N
7090 CELLS N TO N+79 AND N+100 TO N+179 CONTAIN A BINARY CARD IMAGE (IF N=MEM+401, THIS IS IN THE NORMAL AREA OF 1401 MEMORY.) IN 7090 CELLS MEM+1 TO MEM+80,CCNSTRUCT THE EQUIVALENT BCD CARD IMAGE.
SET L=1
SET P=N
CAL P
ALS 6
ORA P+100
NOW BITS 10,11,26,27,28,29,16,17,32,33,34,35 OF THE AC CONTAIN THE 12 BITS OF AN ORIGINAL CARD COLUMN.
SET Z=D=0
BCTH 8+9 PUNCHES
+
−
PRETEND BOTH ARE ABSENT FROM COLUMN
ZERO PUNCH
−
+
123 678 OR 9 PUNCH
+
−
4 OR 5 PUNCH
+
−
SET Z=1
SET D=10
11 PUNCH
−
+
SET Z=2
12 PUNCH
−
+
SET Z=3
9 PUNCH
+
−
SET D=9
8 PUNCH
−
+
SET D=8
7 PUNCH
6 PUNCH
5 PUNCH
4 PUNCH
3 PUNCH
2 PUNCH
1 PUNCH
+
−
INCREASE D BY 1
INCREASE D BY 2
INCREASE D BY 3
INCREASE D BY 4
INCREASE D BY 5
INCREASE D BY 6
INCREASE D BY 7
PUT Z IN DECREMENT OF 7090 LOCATION MEM+L (ZONE BITS OF 1401 CELL L),D IN ADDRESS (DIGIT BITS OF 1401 CELL.)
L=80
−
+
EXIT
INCREASE P AND L BY 1


RCK - TSX RCK,4
PZE A
PZE B
SET L=A
SET P=B
RCKA
DOES L CONTAIN RECORD MARK
+
−
RCKD
PICK UP 0,8,2 PUNCHES
GET THE ZONE PUNCHES FROM RCKT FF.,AND THE NUMERICAL PUNCHES FROM RCKU FF.
STORE PUNCHES 12,11,0,1,2,3 IN P,AND PUNCHES 4,5,6,7,8, 9 IN P+100.
INCREASE L AND P BY 1.
L =A+80
+
EXIT

7090 CELLS A TO A+79 CONTAIN,IN 1401 FORM, A BCD CARD IMAGE. CONSTRUCT THE EQUIVALENT BINARY CARD IMAGE IN CELLS B TO B+79 AND B+100 TO B+179,NOT DISTURBING SIGN BITS (I.E. WORD MARKS) IN EITHER AREA.

****
FLOW CHART 9
****



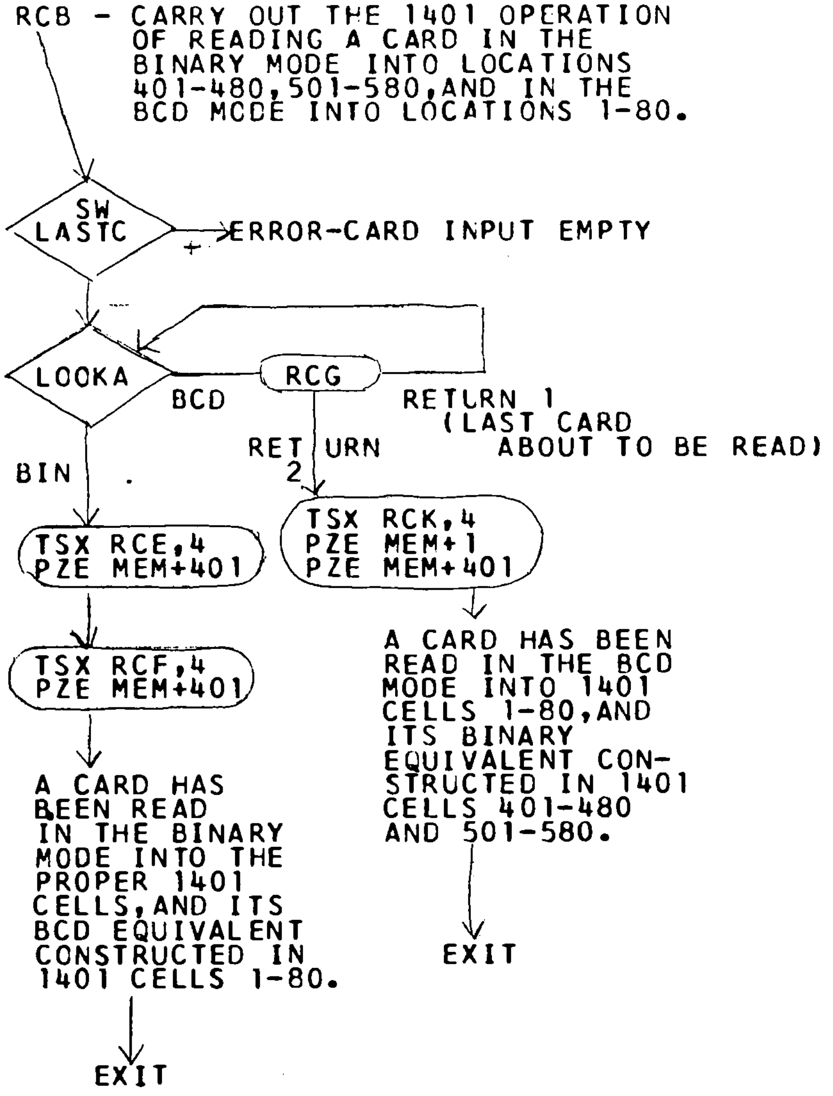
RCD - CARRY OUT THE 1401 OPERATION OF READING A CARD IN THE BCD MODE INTC LOCATIONS 1 TO 80

READ A BINARY CARD INTO WORKSPACE,THEN CONSTRUCT BCD EQUIVALENT IN 1401 LOCATIONS 1-80.

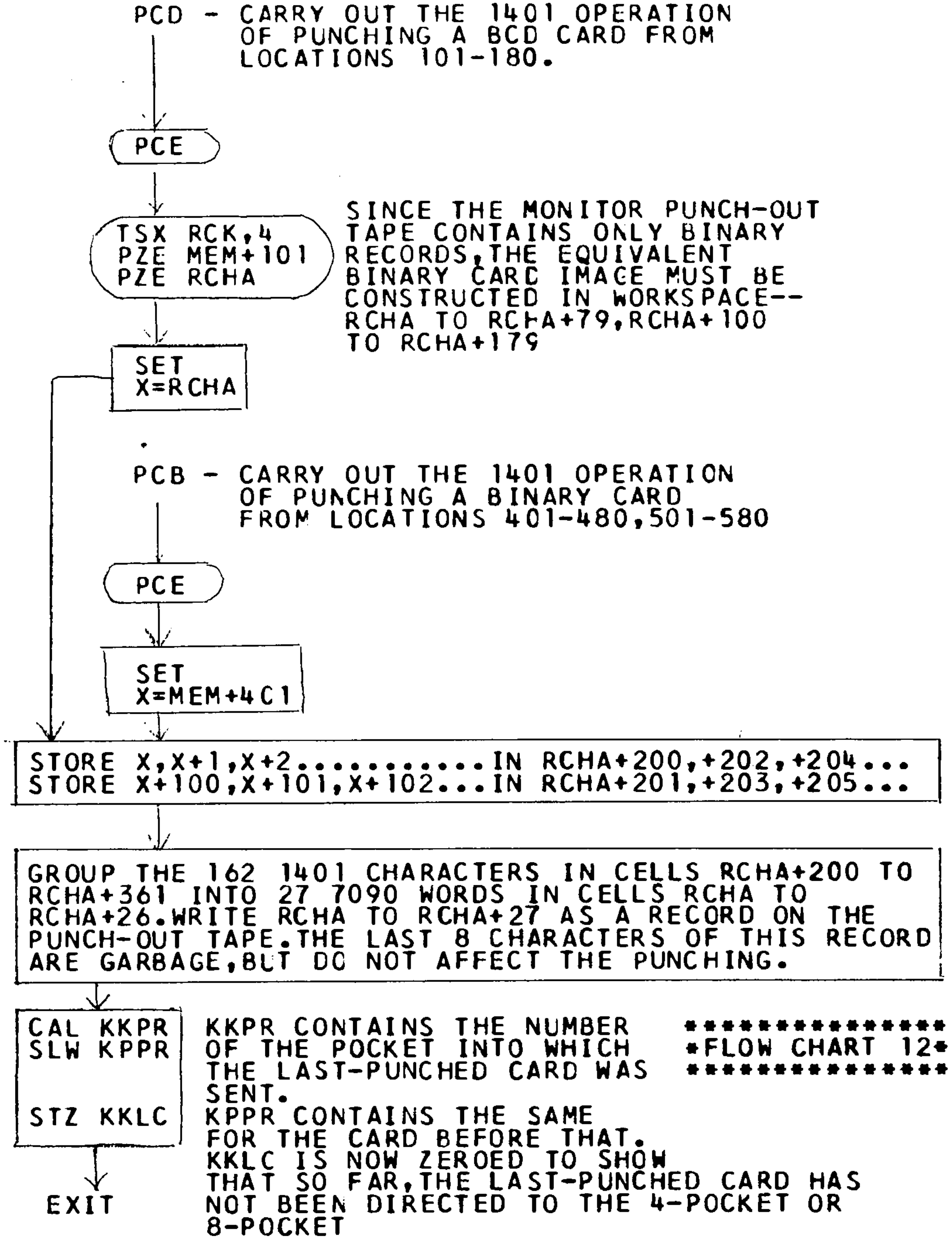
THE VALIDITY CHECK CAN BE DETECTED BY A 1401 BRANCH CN READER ERROR.

THIS IS THE ONLY TYPE OF CARD CHECK THAT IS SIMULATED, SINCE A BCD CARD WITH A BAD PUNCH WILL HAVE BEEN DETECTED ALREADY, DURING THE MONITOR INPUT.

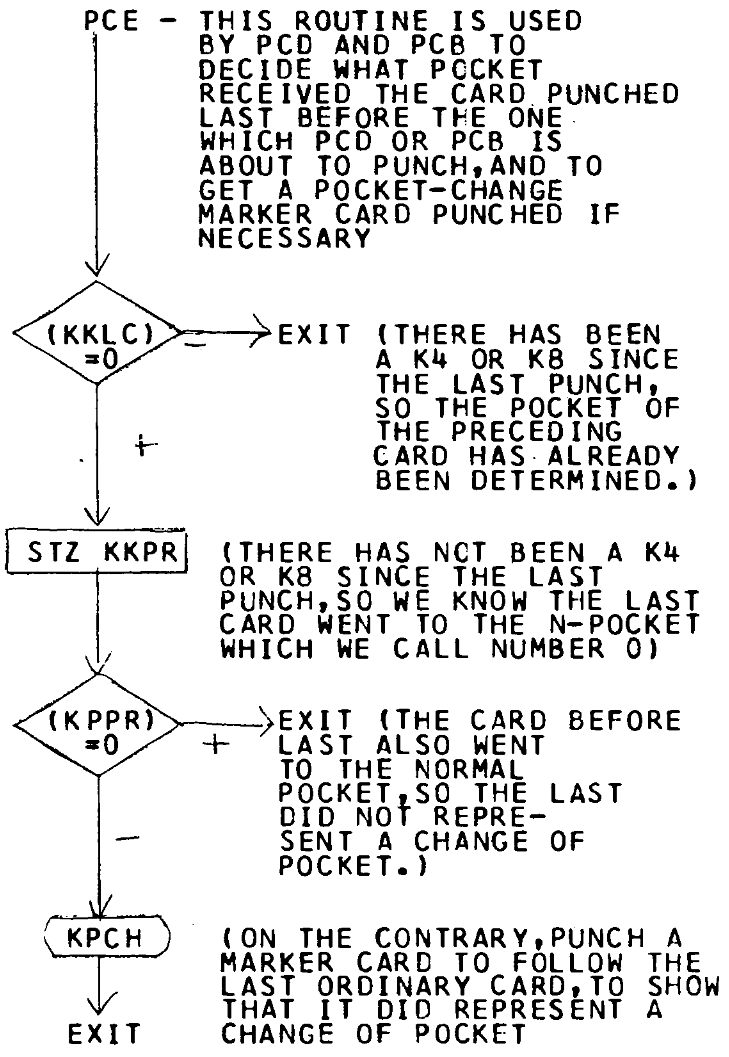
****
FLCW CHART 10
****


RCB - CARRY OUT THE 1401 OPERATION OF READING A CARD IN THE BINARY MODE INTO LOCATIONS 401-480,501-580,AND IN THE BCD MODE INTO LOCATIONS 1-80.
SW LASTC
+
ERROR-CARD INPUT EMPTY
LOOKA
BCD
RCG
RETURN 1 (LAST CARD ABOUT TO BE READ)
RET URN 2
BIN
TSX RCE,4
PZE MEM+401
TSX RCK,4
PZE MEM+1
PZE MEM+401
TSX RCF,4
PZE MEM+401
A CARD HAS BEEN READ IN THE BCD MODE INTO 1401 CELLS 1-80,AND ITS BINARY EQUIVALENT CONSTRUCTED IN 1401 CELLS 401-480 AND 501-580.
A CARD HAS BEEN READ IN THE BINARY MODE INTO THE PROPER 1401 CELLS,AND ITS BCD EQUIVALENT CONSTRUCTED IN 1401 CELLS 1-80.
EXIT
EXIT

****
FLOW CHART 11
****


PCD - CARRY OUT THE 1401 OPERATION OF PUNCHING A BCD CARD FROM LOCATIONS 101-180.
PCE
TSX RCK,4
PZE MEM+101
PZE RCHA
SINCE THE MONITOR PUNCH-OUT TAPE CONTAINS ONLY BINARY RECORDS,THE EQUIVALENT BINARY CARD IMAGE MUST BE CONSTRUCTED IN WORKSPACE--RCHA TO RCHA+79,RCHA+100 TO RCHA+179
SET X=RCHA
PCB - CARRY OUT THE 1401 OPERATION OF PUNCHING A BINARY CARD FROM LOCATIONS 401-480,501-580
PCE
SET X=MEM+4C1
STORE X,X+1,X+2...........IN RCHA+200,+202,+204...
STORE X+100,X+101,X+102...IN RCHA+201,+203,+205...
GROUP THE 162 1401 CHARACTERS IN CELLS RCHA+200 TO RCHA+361 INTO 27 7090 WORDS IN CELLS RCHA TO RCHA+26.WRITE RCHA TO RCHA+27 AS A RECORD ON THE PUNCH-OUT TAPE.THE LAST 8 CHARACTERS OF THIS RECORD ARE GARBAGE,BUT DO NOT AFFECT THE PUNCHING.
CAL KKPR
SLW KPPR
STZ KKLC
KKPR CONTAINS THE NUMBER OF THE POCKET INTO WHICH THE LAST-PUNCHED CARD WAS SENT.
KPPR CONTAINS THE SAME FOR THE CARD BEFORE THAT.
KKLC IS NOW ZEROED TO SHOW THAT SO FAR,THE LAST-PUNCHED CARD HAS NOT BEEN DIRECTED TO THE 4-POCKET OR 8-POCKET
EXIT

****
FLOW CHART 12
****



PCE - THIS ROUTINE IS USED BY PCD AND PCB TO DECIDE WHAT POCKET RECEIVED THE CARD PUNCHED LAST BEFORE THE ONE WHICH PCD OR PCB IS ABOUT TO PUNCH,AND TO GET A POCKET-CHANGE MARKER CARD PUNCHED IF NECESSARY

EXIT (THERE HAS BEEN A K4 OR K8 SINCE THE LAST PUNCH, SO THE POCKET OF THE PRECEDING CARD HAS ALREADY BEEN DETERMINED.)

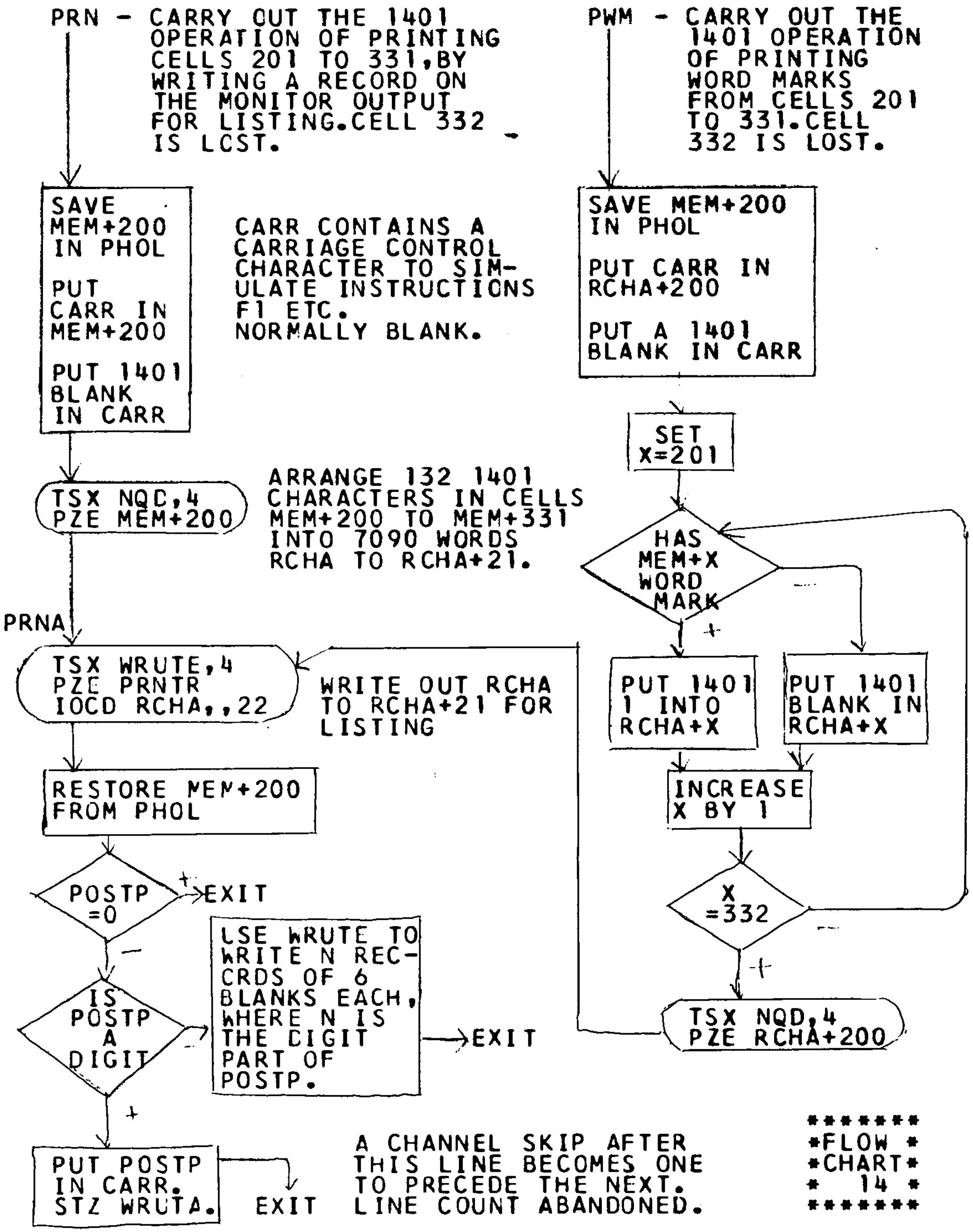
(THERE HAS NCT BEEN A K4 OR K8 SINCE THE LAST PUNCH,SO WE KNOW THE LAST CARD WENT TO THE N-POCKET WHICH WE CALL NUMBER 0)

EXIT (THE CARD BEFORE LAST ALSO WENT TO THE NORMAL POCKET,SO THE LAST DID NOT REPRESENT A CHANGE OF POCKET.)

(ON THE CONTRARY,PUNCH A MARKER CARD TO FOLLOW THE LAST ORDINARY CARD,TO SHOW THAT IT DID REPRESENT A CHANGE OF POCKET

****
FLOW CHART 13
****

KPCH - WRITE,ON THE OUTPUT TAPE FOR PUNCHING,WHAT WILL BE A MARKER CARD FOR RECOGNITION ON SIGHT,WITH A BORDER OF PUNCHED HOLES,AND N , 4 ,OR 8 IN THE MIDDLE ACCCRDING TO WHETHER CELL KKPR NOW CONTAINS 0,4,OR 8.



PRN - CARRY OUT THE 1401 OPERATION OF PRINTING CELLS 201 TO 331,BY WRITING A RECORD ON THE MONITOR OUTPUT FOR LISTING.CELL 332 IS LCST.

CARR CONTAINS A CARRIAGE CONTROL CHARACTER TO SIMULATE INSTRUCTIONS F1 ETC. NORMALLY BLANK.

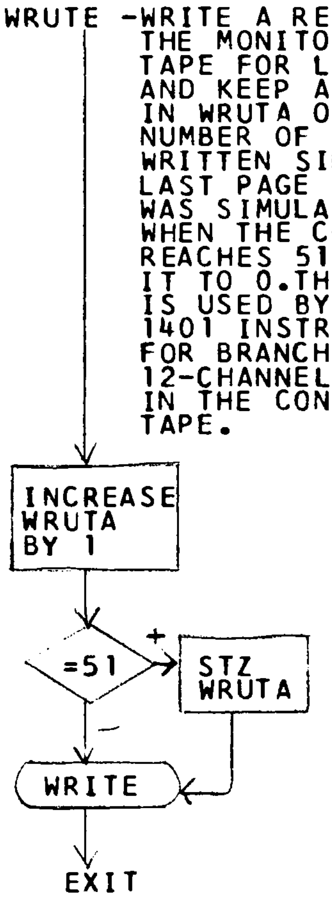
ARRANGE 132 1401 CHARACTERS IN CELLS MEM+200 TO MEM+331 INTO 7090 WORDS RCHA TO RCHA+21.

WRITE OUT RCHA TO RCHA+21 FOR LISTING

PWM - CARRY OUT THE 1401 OPERATION OF PRINTING WORD MARKS FROM CELLS 201 TO 331.CELL 332 IS LOST.

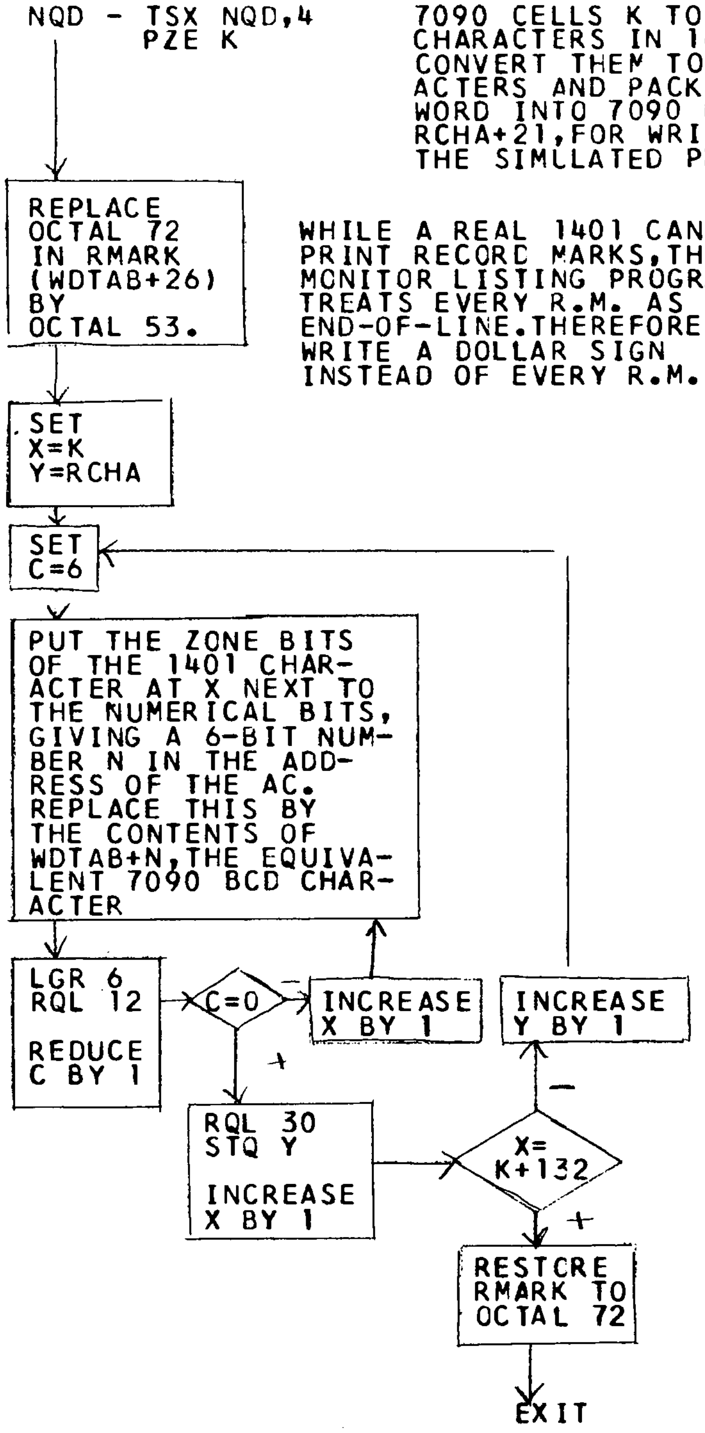
A CHANNEL SKIP AFTER THIS LINE BECOMES ONE TO PRECEDE THE NEXT. LINE COUNT ABANDONED.

****
*FLOW *
CHART
* 14 *
****



WRUTE -WRITE A RECORD ON THE MONITOR OUTPUT TAPE FOR LISTING AND KEEP A COUNT IN WRUTA OF THE NUMBER OF RECORDS WRITTEN SINCE THE LAST PAGE SKIP WAS SIMULATED. WHEN THE COUNT REACHES 51,RETURN IT TO 0.THIS COUNT IS USED BY THE 1401 INSTRUCTION FOR BRANCHING ON 12-CHANNEL PUNCH IN THE CONTROL TAPE.

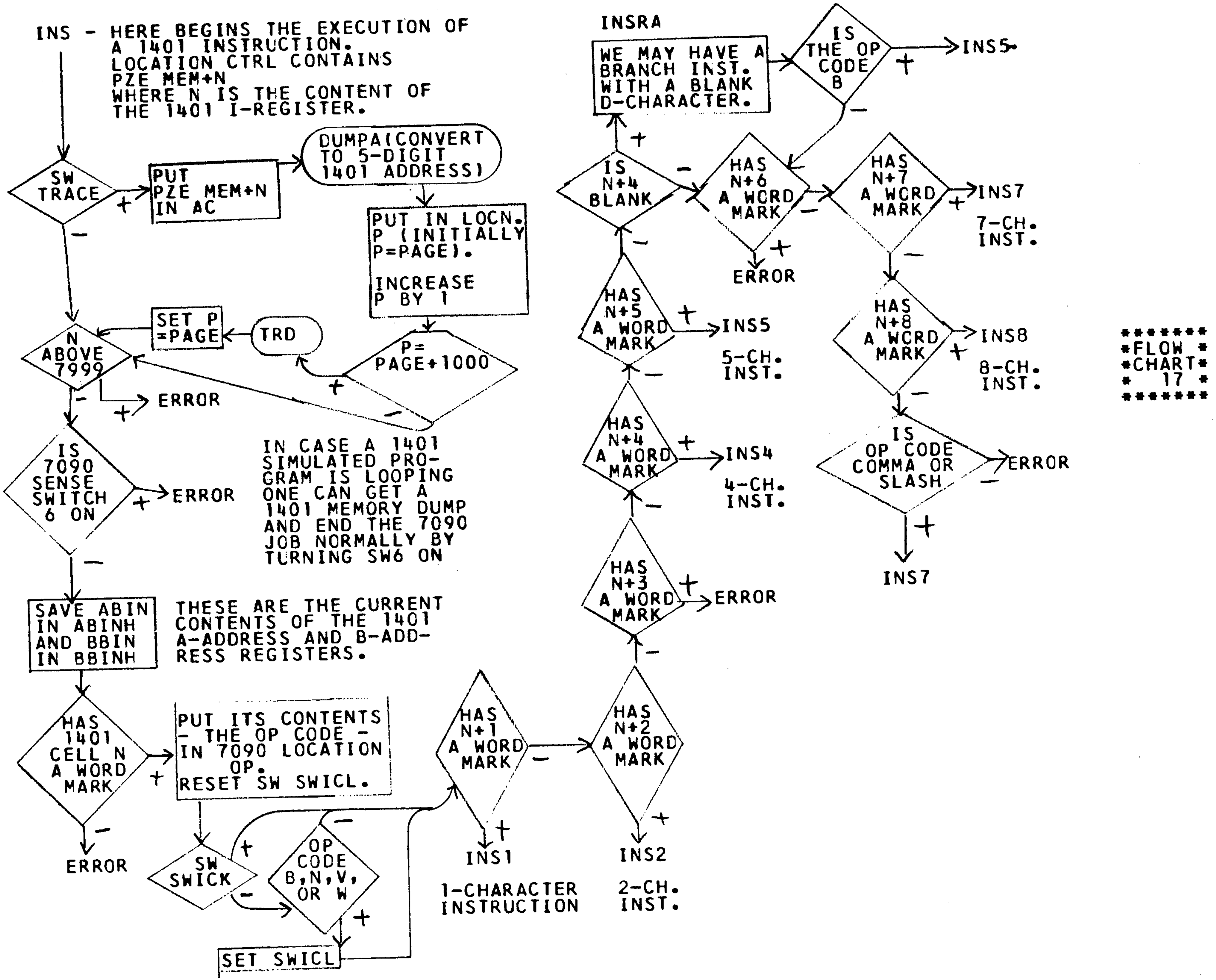
****
FLOW CHART 15
****

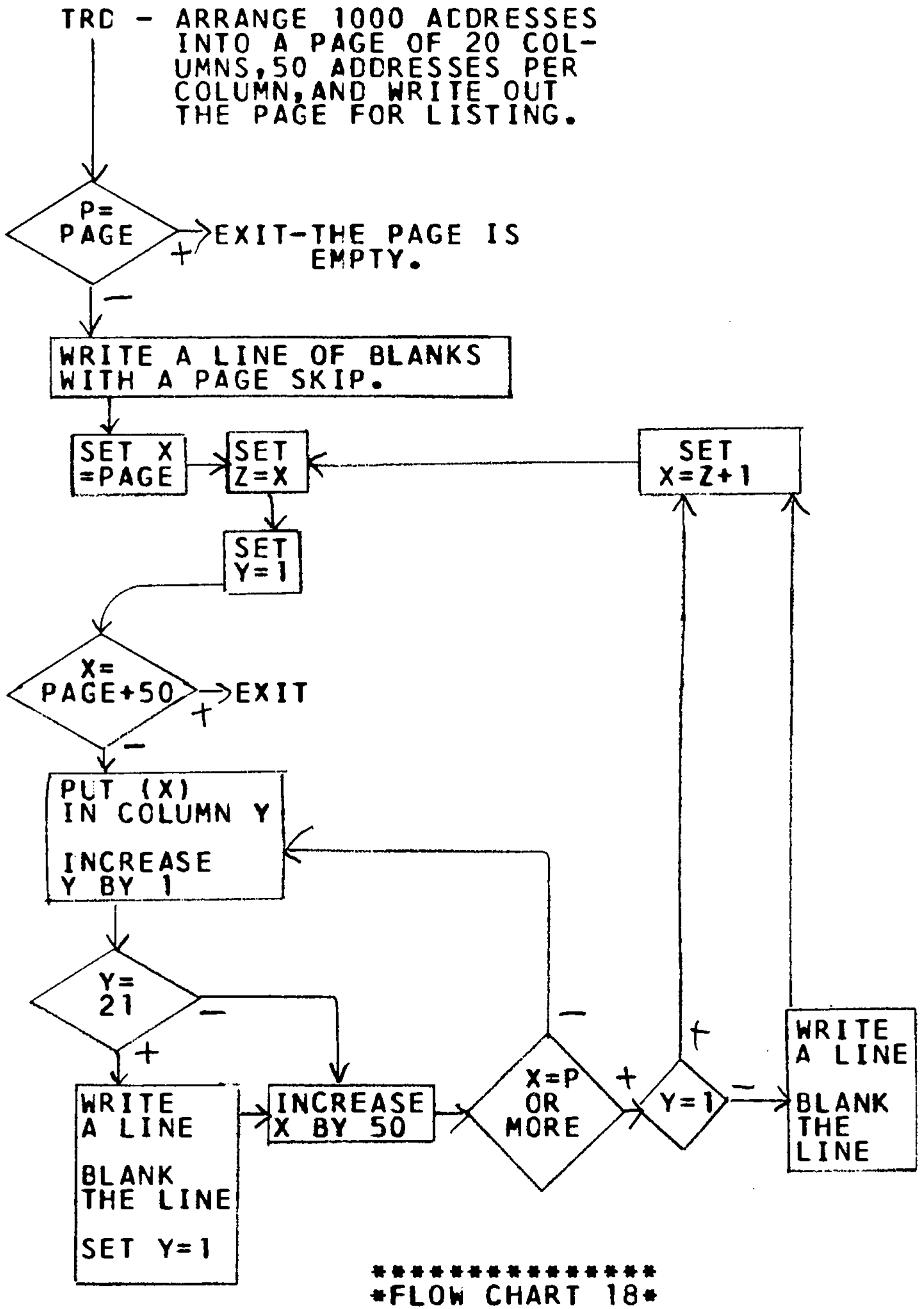

NQD - TSX NQD,4
PZE K
REPLACE OCTAL 72 IN RMARK (WDTAB+26) BY OCTAL 53.
SET X=K Y=RCHA
SET C=6
PUT THE ZONE BITS OF THE 1401 CHARACTER AT X NEXT TO THE NUMERICAL BITS, GIVING A 6-BIT NUMBER N IN THE ADDRESS OF THE AC. REPLACE THIS BY THE CONTENTS OF WDTAB+N,THE EQUIVALENT 7090 BCD CHARACTER
LGR 6 RQL 12 REDUCE C BY 1
C=0
−
INCREASE X BY 1
+
RQL 30 STQ Y INCREASE X BY 1
X= K+132
−
INCREASE Y BY 1
+
RESTCRE RMARK TO OCTAL 72
EXIT

7090 CELLS K TO K+131 CCNTAIN CHARACTERS IN 1401 FORM. CONVERT THEM TO 7090 BCD CHARACTERS AND PACK THEM 6 PER WORD INTO 7090 CELLS RCHA TO RCHA+21,FOR WRITING AS PART OF THE SIMULATED PRINTER OUTPUT

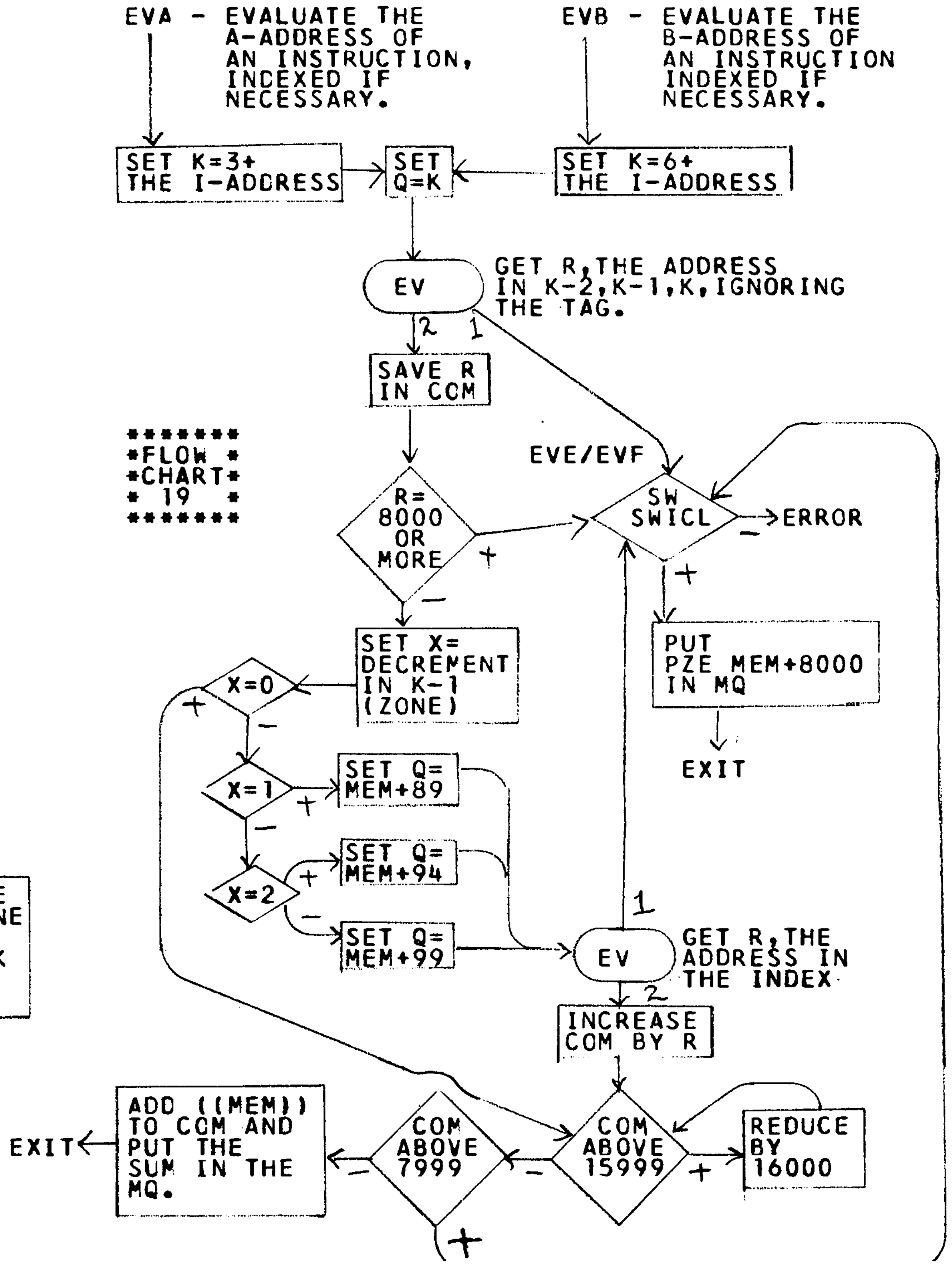
WHILE A REAL 1401 CAN PRINT RECORC MARKS,THE MCNITOR LISTING PROGRAM TREATS EVERY R.M. AS AN END-OF-LINE.THEREFORE WRITE A DOLLAR SIGN INSTEAD OF EVERY R.M.

****
FLOW CHART 16
****

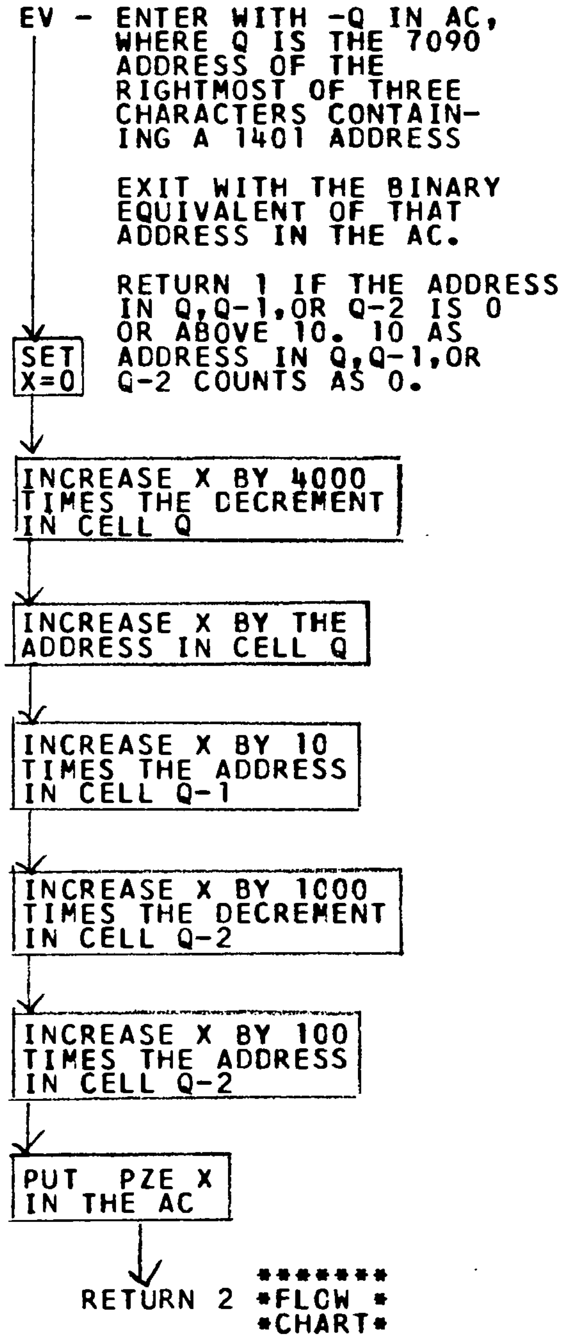

INS - HERE BEGINS THE EXECUTION OF A 1401 INSTRUCTION. LOCATION CTRL CONTAINS PZE MEM+N WHERE N IS THE CONTENT OF THE 1401 I-REGISTER.
SW TRACE
+
PUT PZE MEM+N IN AC
DUMPA(CONVERT TO 5-DIGIT 1401 ADDRESS)
PUT IN LOCN. P (INITIALLY P=PAGE). INCREASE P BY 1
P= PAGE+1000
+
TRD
SET P =PAGE
-
-
N ABOVE 7999
+
ERROR
-
IS 7090 SENSE SWITCH 6 ON
+
ERROR
-
IN CASE A 1401 SIMULATED PROGRAM IS LOOPING ONE CAN GET A 1401 MEMORY DUMP AND END THE 7090 JOB NORMALLY BY TURNING SW6 ON
SAVE ABIN IN ABINH AND BBIN IN BBINH
THESE ARE THE CURRENT CONTENTS OF THE 1401 A-ADDRESS AND B-ADDRESS REGISTERS.
HAS 1401 CELL N A WORD MARK
-
ERROR
+
PUT ITS CONTENTS - THE OP CODE - IN 7090 LOCATION OP. RESET SW SWICL.
SW SWICK
+
-
OP CODE B,N,V, OR W
-
+
SET SWICL
HAS N+1 A WORD MARK
+
INS1
1-CHARACTER INSTRUCTION
-
HAS N+2 A WORD MARK
+
INS2
2-CH. INST.
-
HAS N+3 A WORD MARK
+
ERROR
-
HAS N+4 A WORD MARK
+
INS4
4-CH. INST.
-
HAS N+5 A WORD MARK
+
INS5
5-CH. INST.
-
IS N+4 BLANK
+
INSRA
WE MAY HAVE A BRANCH INST. WITH A BLANK D-CHARACTER.
IS THE OP CODE B
+
INS5
-
-
HAS N+6 A WORD MARK
+
ERROR
-
HAS N+7 A WORD MARK
+
INS7
7-CH. INST.
-
HAS N+8 A WORD MARK
+
INS8
8-CH. INST.
-
IS OP CODE COMMA OR SLASH
-
ERROR
+
INS7
****
*FLOW *
CHART
* 17 *
****



FLOW CHART 18


EVA - EVALUATE THE A-ADDRESS OF AN INSTRUCTION, INDEXED IF NECESSARY.
EVB - EVALUATE THE B-ADDRESS OF AN INSTRUCTION INDEXED IF NECESSARY.
SET K=3+ THE I-ADDRESS
SET Q=K
SET K=6+ THE I-ADDRESS
EV
GET R,THE ADDRESS IN K-2,K-1,K,IGNORING THE TAG.
2
1
SAVE R IN COM
EVE/EVF
R= 8000 OR MORE
+
−
SW SWICL
−
ERROR
+
PUT PZE MEM+8000 IN MQ
EXIT
SET X= DECREMENT IN K-1 (ZONE)
X=0
+
−
X=1
+
SET Q= MEM+89
−
X=2
+
SET Q= MEM+94
−
SET Q= MEM+99
1
EV
GET R,THE ADDRESS IN THE INDEX
2
INCREASE COM BY R
COM ABOVE 15999
+
REDUCE BY 16000
−
COM ABOVE 7999
−
+
ADD ((MEM)) TO COM AND PUT THE SUM IN THE MQ.
EXIT

FLOW CHART 19



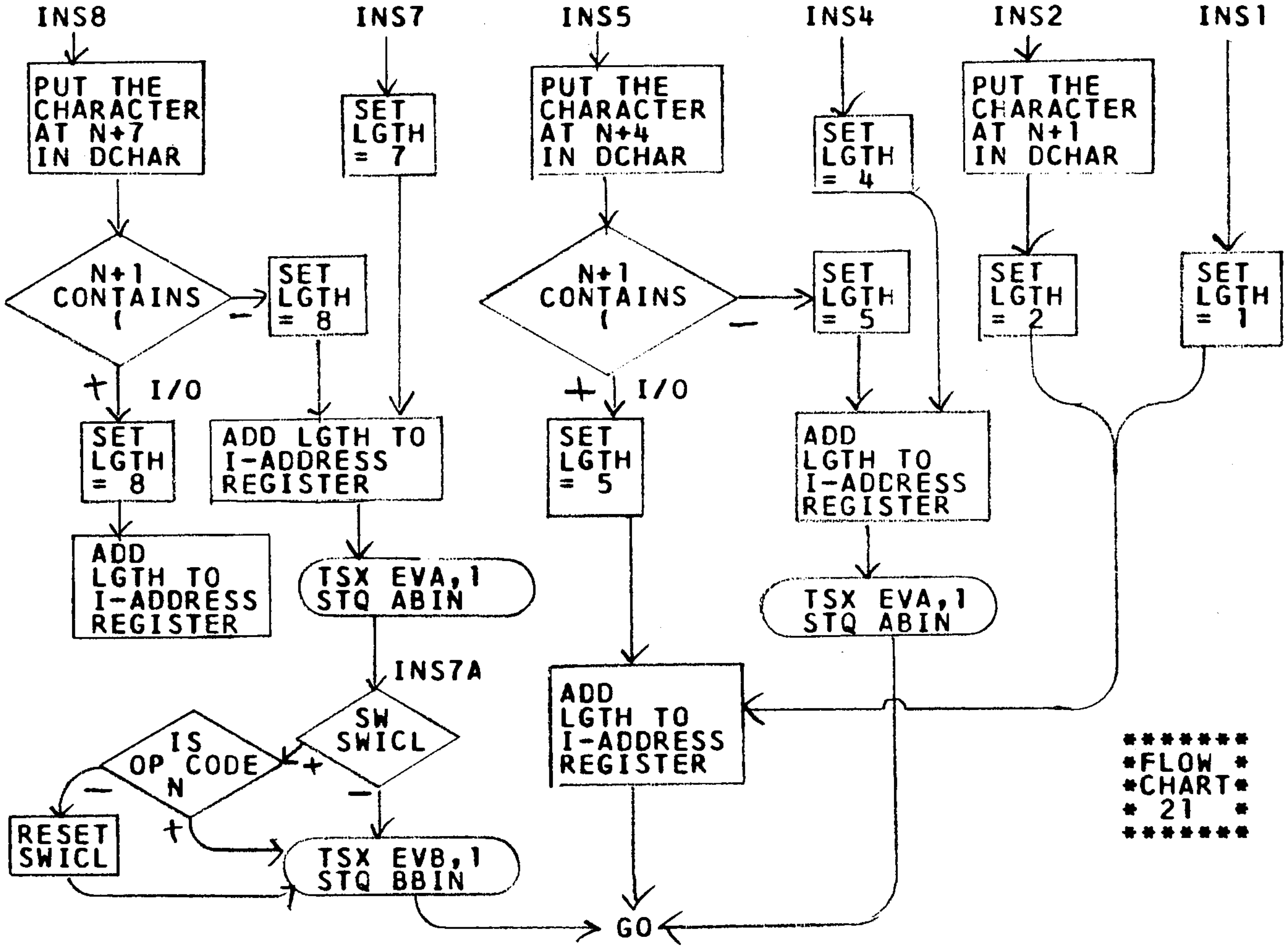
EV - ENTER WITH -Q IN AC, WHERE Q IS THE 7090 ADDRESS OF THE RIGHTMOST OF THREE CHARACTERS CONTAINING A 1401 ADDRESS

EXIT WITH THE BINARY EQUIVALENT OF THAT ADDRESS IN THE AC.

RETURN 1 IF THE ADDRESS IN Q,Q-1,OR Q-2 IS 0 OR ABOVE 10. 10 AS ADDRESS IN Q,Q-1,OR Q-2 COUNTS AS 0.

EV
SET X=0
INCREASE X BY 4000 TIMES THE DECREMENT IN CELL Q
INCREASE X BY THE ADDRESS IN CELL Q
INCREASE X BY 10 TIMES THE ADDRESS IN CELL Q-1
INCREASE X BY 1000 TIMES THE DECREMENT IN CELL Q-2
INCREASE X BY 100 TIMES THE ADDRESS IN CELL Q-2
PUT PZE X IN THE AC
RETURN 2

****
*FLOW *
CHART
* 20 *
****



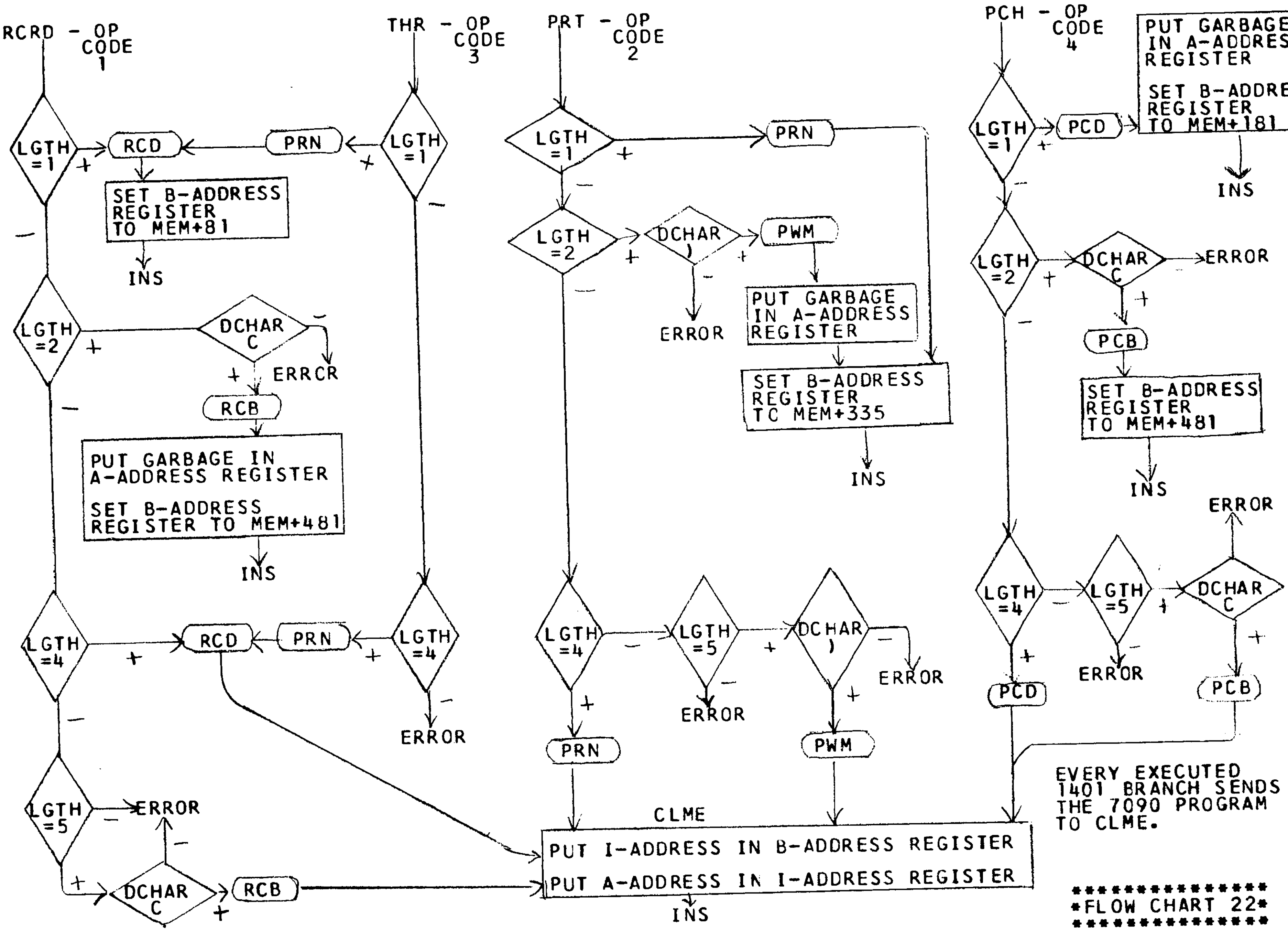
****
*FLOW *
CHART
* 21 *
****

LAC BBIN,2 - COMPLEMENT OF 7090 VERSION OF B-ADDRESS

LAC ABIN,1 - COMPLEMENT OF 7090 VERSION OF A-ADDRESS

BRANCH TO ROUTINE FOR 1401 FUNCTION SPECIFIED BY OP CODE (SEE TABLES GOZ,GO1,GO2,GO3 IN PROGRAM LISTING)

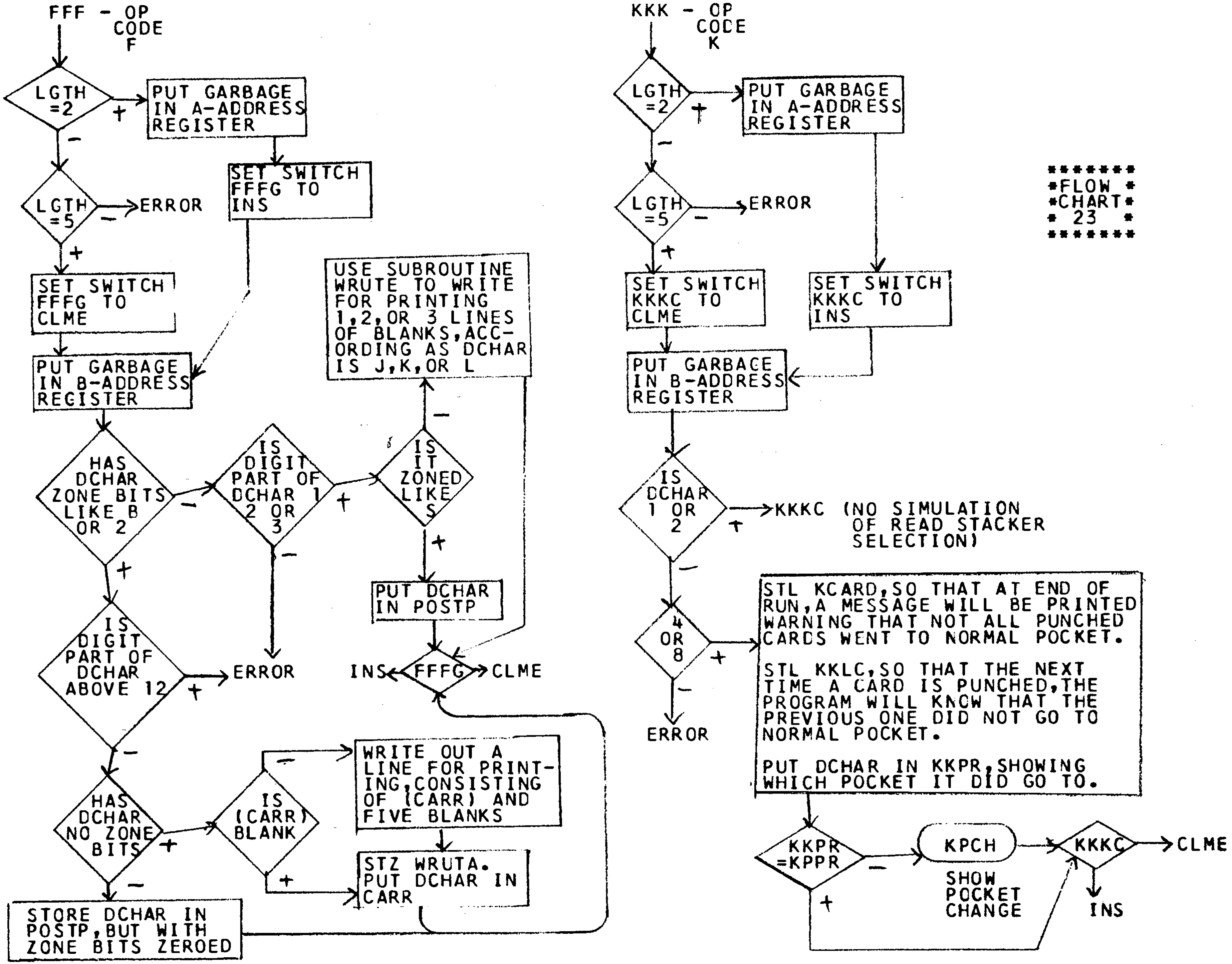
ERROR, IF OP CODE IS G I J O R T X $ * + - 0 RECORD MARK OR BLANK, OR HAS A NUMERICAL PART ABOVE 12.



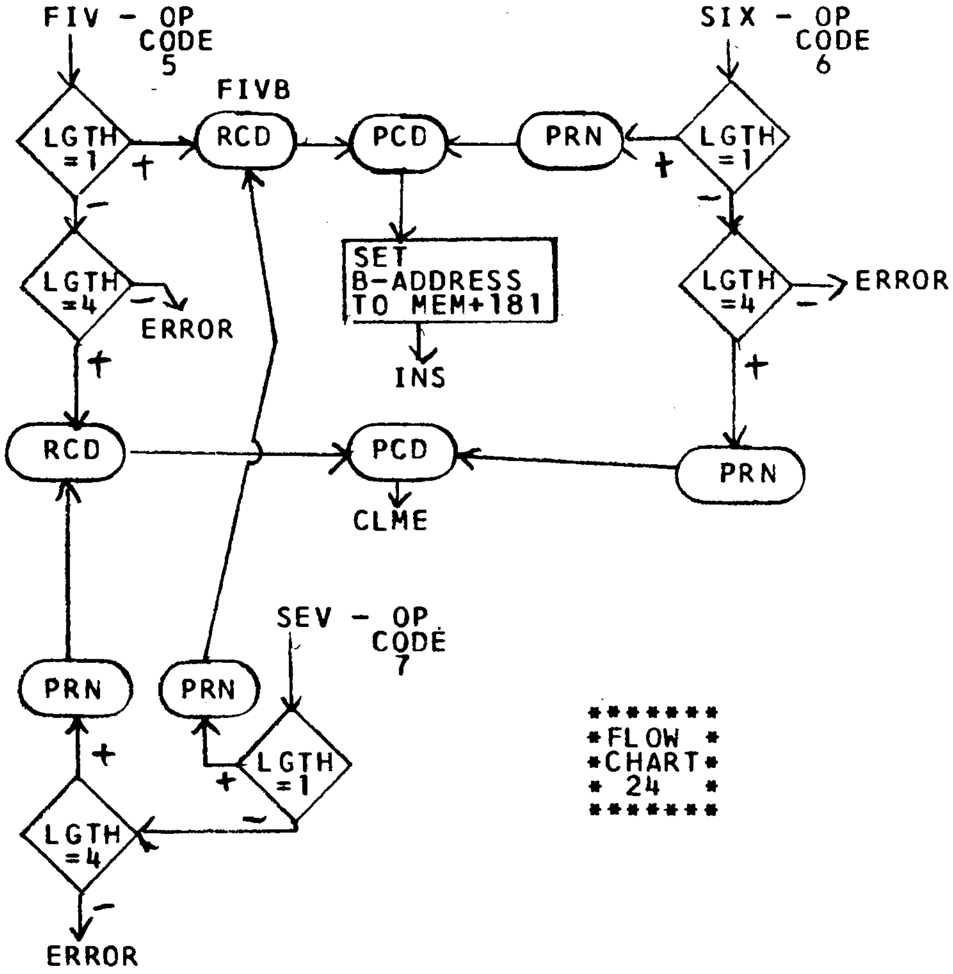
RCRD - OP CODE 1
THR - OP CODE 3
PRT - OP CODE 2
PCH - OP CODE 4
LGTH =1
LGTH =2
LGTH =4
LGTH =5
DCHAR C
DCHAR)
RCD
PRN
RCB
PWM
PCD
PCB
SET B-ADDRESS REGISTER TO MEM+81
PUT GARBAGE IN A-ADDRESS REGISTER
SET B-ADDRESS REGISTER TO MEM+481
SET B-ADDRESS REGISTER TO MEM+335
PUT GARBAGE IN A-ADDRESS REGISTER
SET B-ADDRESS REGISTER TO MEM+181
SET B-ADDRESS REGISTER TO MEM+481
INS
ERROR
ERRCR
CLME
PUT I-ADDRESS IN B-ADDRESS REGISTER
PUT A-ADDRESS IN I-ADDRESS REGISTER
+
-

EVERY EXECUTED 1401 BRANCH SENDS THE 7090 PROGRAM TO CLME.

****
FLOW CHART 22
****


FFF - OP CODE F
LGTH =2
+
PUT GARBAGE IN A-ADDRESS REGISTER
-
SET SWITCH FFFG TO INS
LGTH =5
-
ERROR
+
SET SWITCH FFFG TO CLME
PUT GARBAGE IN B-ADDRESS REGISTER
USE SUBROUTINE WRUTE TO WRITE FOR PRINTING 1,2,OR 3 LINES OF BLANKS,ACCORDING AS DCHAR IS J,K,OR L
HAS DCHAR ZONE BITS LIKE B OR 2
-
IS DIGIT PART OF DCHAR 1 2 OR 3
+
IS IT ZONED LIKE S
-
-
+
+
PUT DCHAR IN POSTP
IS DIGIT PART OF DCHAR ABOVE 12
+
ERROR
ERROR
INS
FFFG
CLME
-
WRITE OUT A LINE FOR PRINTING,CONSISTING OF (CARR) AND FIVE BLANKS
-
HAS DCHAR NO ZONE BITS
+
IS (CARR) BLANK
+
STZ WRUTA. PUT DCHAR IN CARR
-
STORE DCHAR IN POSTP,BUT WITH ZONE BITS ZEROED
KKK - OP CODE K
LGTH =2
+
PUT GARBAGE IN A-ADDRESS REGISTER
-
LGTH =5
-
ERROR
+
SET SWITCH KKKC TO CLME
SET SWITCH KKKC TO INS
PUT GARBACE IN B-ADDRESS REGISTER
IS DCHAR 1 OR 2
+
KKKC (NO SIMULATION OF READ STACKER SELECTION)
-
4 OR 8
+
STL KCARD,SO THAT AT END OF RUN,A MESSAGE WILL BE PRINTED WARNING THAT NOT ALL PUNCHED CARDS WENT TO NORMAL POCKET.
STL KKLC,SO THAT THE NEXT TIME A CARD IS PUNCHED,THE PROGRAM WILL KNOW THAT THE PREVIOUS ONE DID NOT GO TO NORMAL POCKET.
PUT DCHAR IN KKPR,SHOWING WHICH POCKET IT DID GO TO.
-
ERROR
KKPR =KPPR
-
KPCH
SHOW POCKET CHANGE
KKKC
CLME
INS
+

FLOW CHART 23



****
*FLOW *
CHART
* 24 *
****

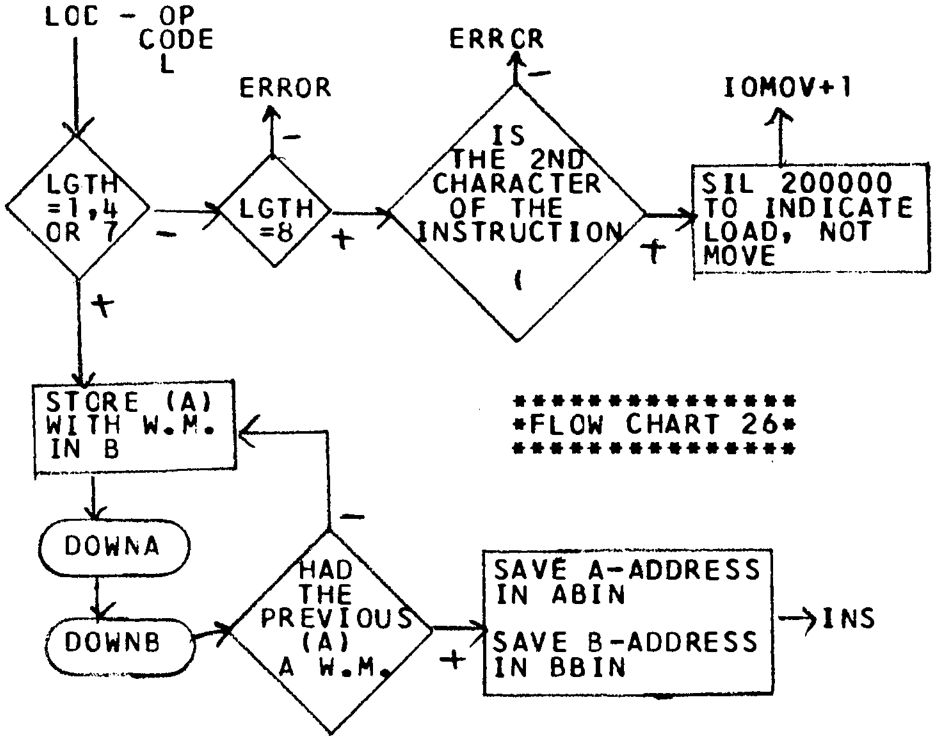
SRF SPF - OP CODES 8 AND 9

INS

THESE FUNCTIONS HAVE NO EFFECT EXCEPT TO SPEED UP AN ACTUAL 1401, OR TO HALT IT IF TIMING IS VIOLATED. AS THE SIMULATOR DOES NOT KEEP TRACK OF 1401 TIMING,THE FORMER EFFECT IS IGNORED,AND THE LATTER ONE, UNFORTUNATELY,CANNOT BE SIMULATED.

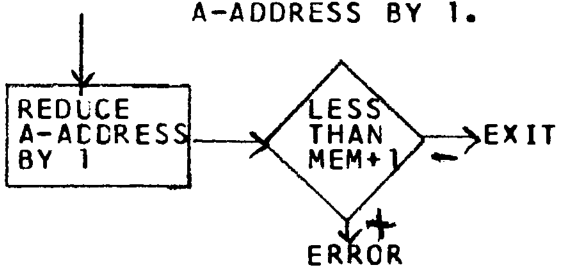
BTD - ENTER WITH A BINARY NUMBER LESS THAN 16000 IN THE AC ADDRESS. IN BTDU,BTDV, AND BTDW CONSTRUCT 3 1401 CHARACTERS THAT FORM AN ADDRESS EQUAL TO THE ORIGINAL NUMBER,BTDU BEING THE HIGH-ORDER CHARACTER.

****
FLOW CHART 25
****



****
FLOW CHART 26
****

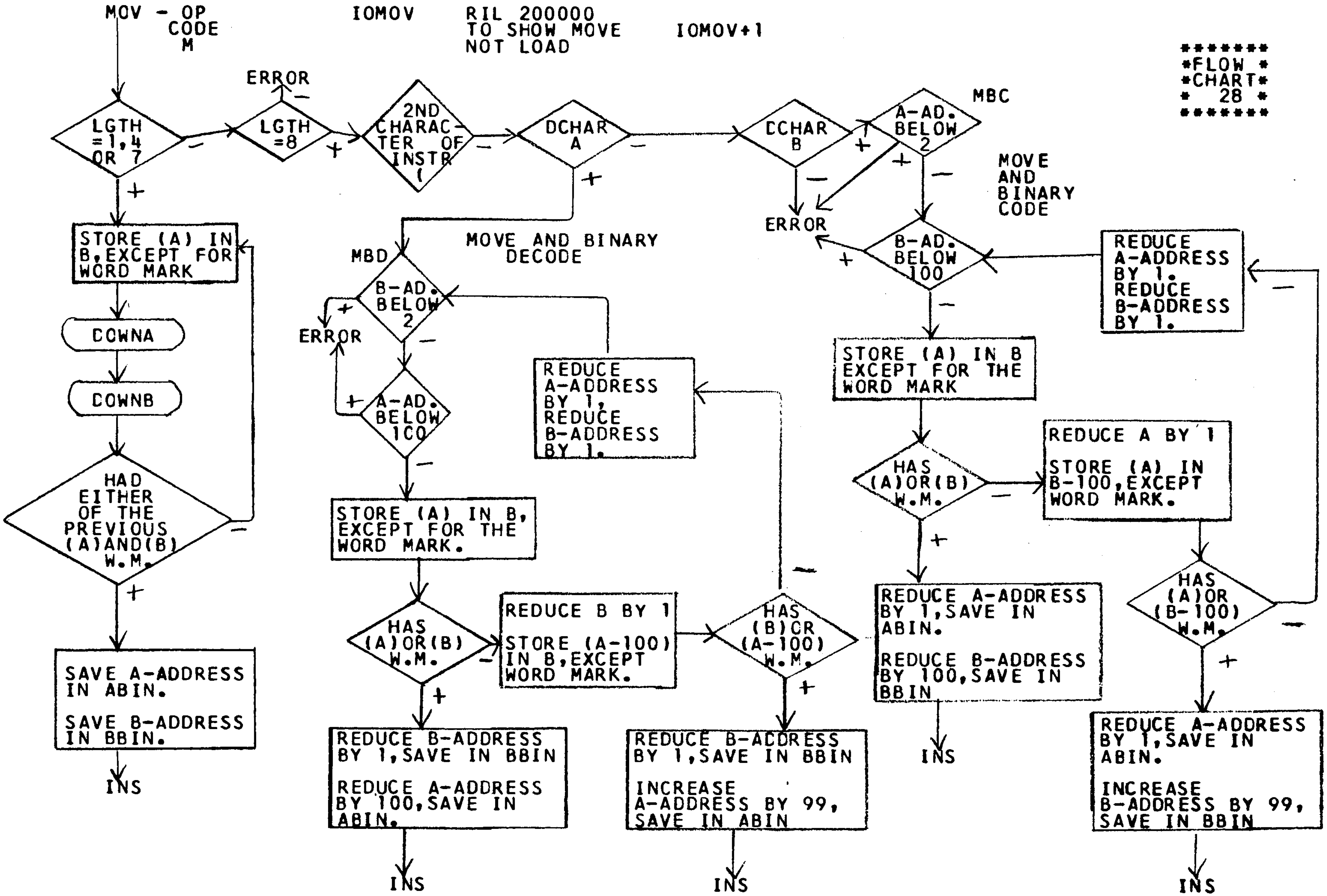
DOWNA - REDUCE THE EFFECTIVE A-ADDRESS BY 1.



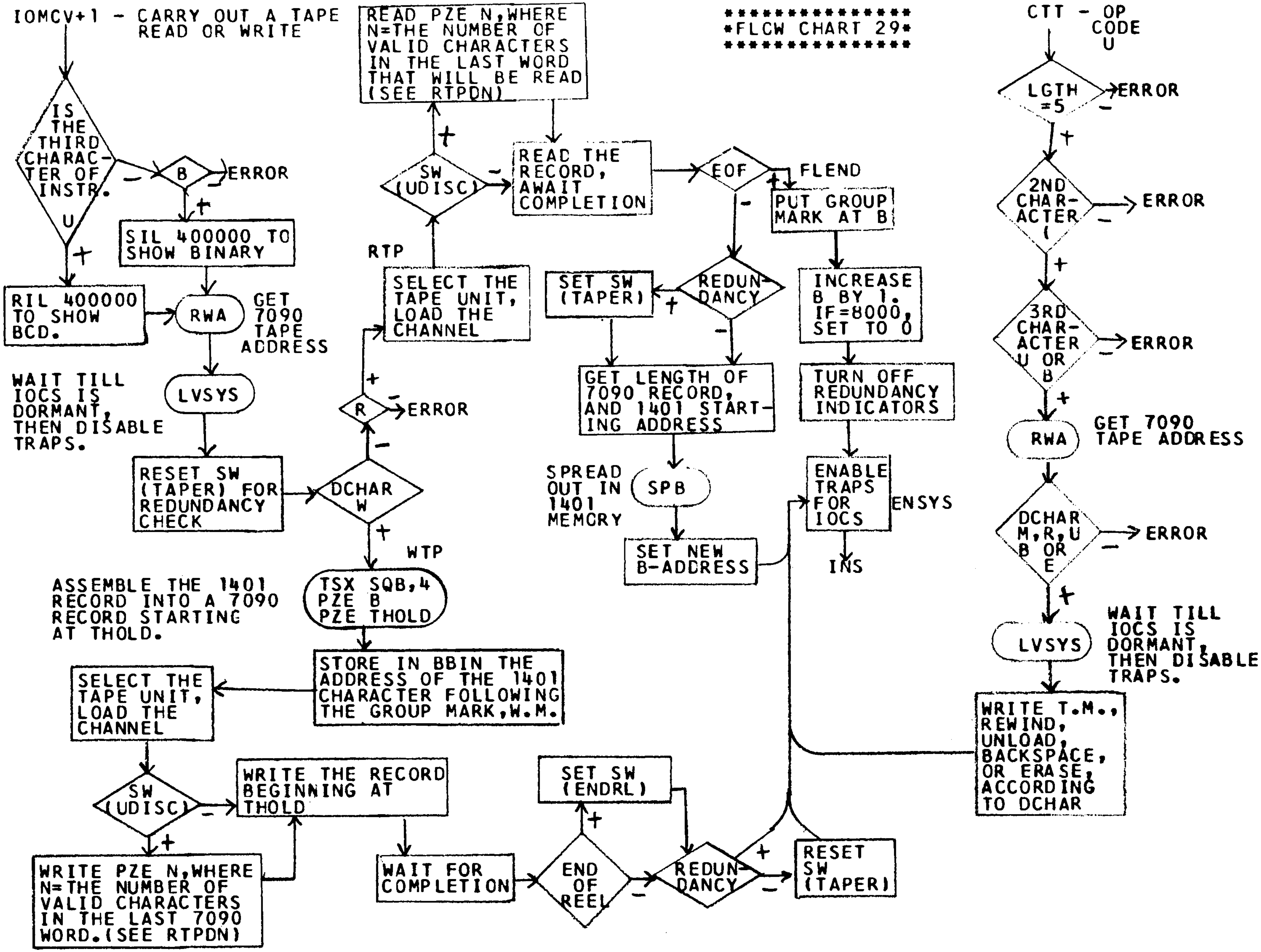
DOWNB IS EXACTLY PARALLEL

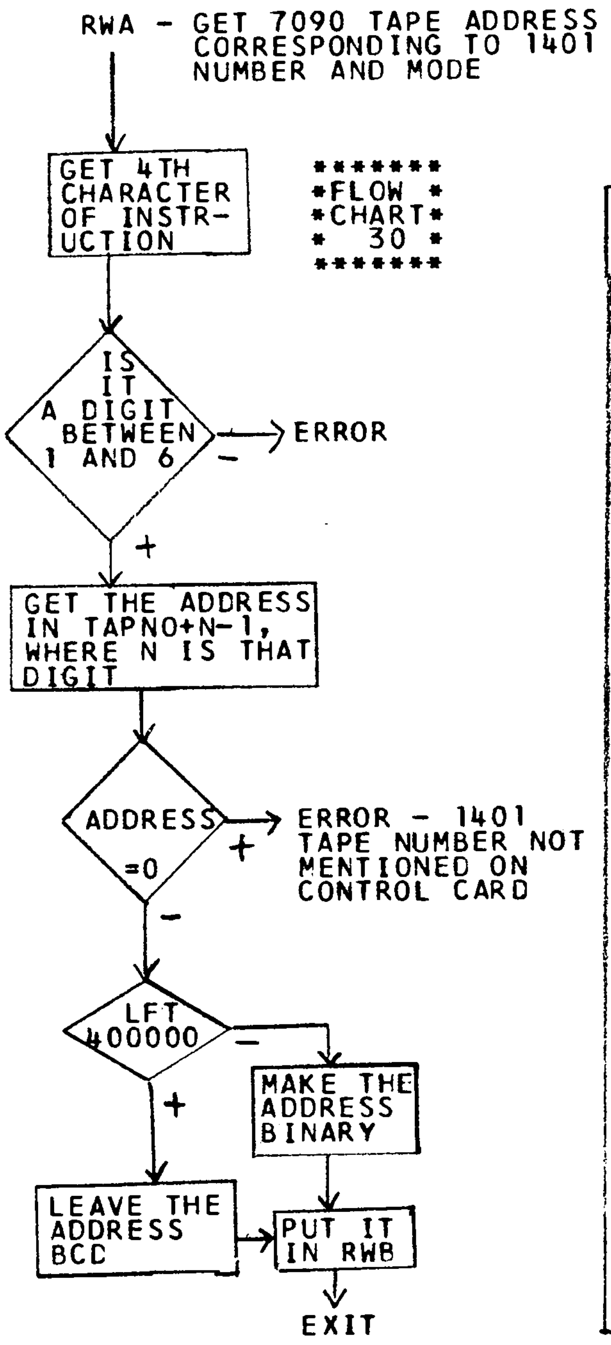
****
*FLOW *
CHART
* 27 *
****

****
*FLOW *
CHART
* 28 *
****


MOV - OP CODE M
IOMOV
RIL 200000 TO SHOW MOVE NOT LOAD
IOMOV+1
LGTH =1,4 OR 7
ERROR
LGTH =8
2ND CHARACTER OF INSTR (
DCHAR A
CCHAR B
MBC
A-AD. BELOW 2
MOVE AND BINARY CODE
ERROR
STORE (A) IN B, EXCEPT FOR WORD MARK
COWNA
COWNB
HAD EITHER OF THE PREVIOUS (A)AND(B) W.M.
SAVE A-ADDRESS IN ABIN.
SAVE B-ADDRESS IN BBIN.
INS
MBD
MOVE AND BINARY DECODE
B-AD. BELOW 2
ERROR
A-AD. BELOW 1CO
STORE (A) IN B, EXCEPT FOR THE WORD MARK.
HAS (A)OR(B) W.M.
REDUCE B-ADDRESS BY 1, SAVE IN BBIN
REDUCE A-ADDRESS BY 100, SAVE IN ABIN.
INS
REDUCE A-ADDRESS BY 1, REDUCE B-ADDRESS BY 1.
REDUCE B BY 1
STORE (A-100) IN B, EXCEPT WORD MARK.
HAS (B)CR (A-100) W.M.
REDUCE B-ADDRESS BY 1, SAVE IN BBIN
INCREASE A-ADDRESS BY 99, SAVE IN ABIN
INS
B-AD. BELOW 100
REDUCE A-ADDRESS BY 1. REDUCE B-ADDRESS BY 1.
STORE (A) IN B EXCEPT FOR THE WORD MARK
HAS (A)OR(B) W.M.
REDUCE A BY 1
STORE (A) IN B-100, EXCEPT WORD MARK.
REDUCE A-ADDRESS BY 1, SAVE IN ABIN.
REDUCE B-ADDRESS BY 100, SAVE IN BBIN
INS
HAS (A)OR (B-100) W.M.
REDUCE A-ADDRESS BY 1, SAVE IN ABIN.
INCREASE B-ADDRESS BY 99, SAVE IN BBIN
INS
+
-

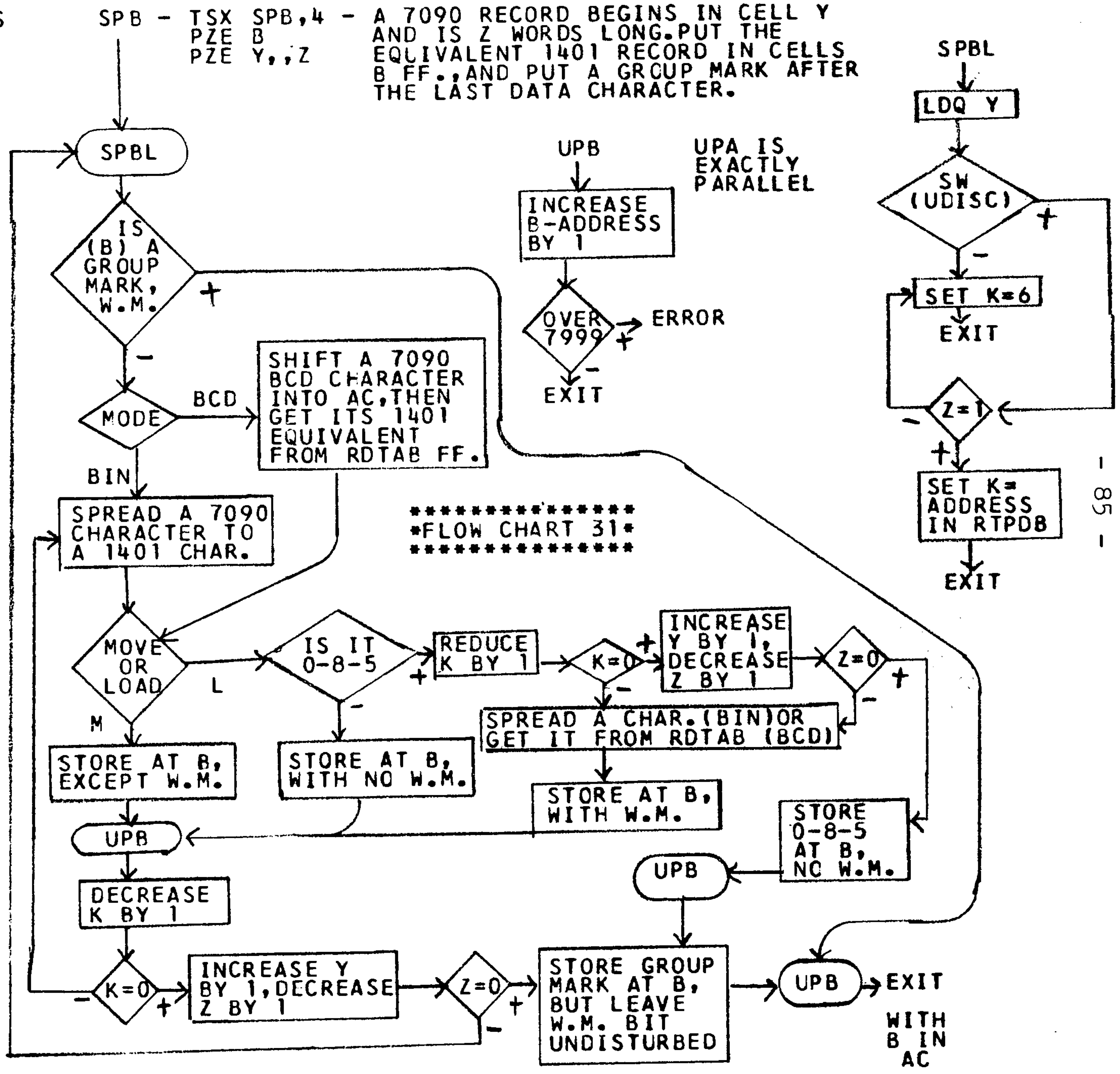
****
FLCW CHART 29
****


IOMCV+1 - CARRY OUT A TAPE READ OR WRITE
IS THE THIRD CHARACTER OF INSTR. U
B
ERROR
SIL 400000 TO SHOW BINARY
RIL 400000 TO SHOW BCD.
RWA
GET 7090 TAPE ADDRESS
LVSYS
WAIT TILL IOCS IS DORMANT, THEN DISABLE TRAPS.
RESET SW (TAPER) FOR REDUNDANCY CHECK
DCHAR W
R
ERROR
RTP
SELECT THE TAPE UNIT, LOAD THE CHANNEL
SW (UDISC)
READ PZE N,WHERE N=THE NUMBER OF VALID CHARACTERS IN THE LAST WORD THAT WILL BE READ (SEE RTPDN)
READ THE RECORD, AWAIT COMPLETION
EOF
FLEND
PUT GROUP MARK AT B
INCREASE B BY 1. IF=8000, SET TO 0
TURN OFF REDUNDANCY INDICATORS
ENABLE TRAPS FOR IOCS
ENSYS
INS
REDUNDANCY
SET SW (TAPER)
GET LENGTH OF 7090 RECORD, AND 1401 STARTING ADDRESS
SPREAD OUT IN 1401 MEMORY
SPB
SET NEW B-ADDRESS
WTP
TSX SQB,4
PZE B
PZE THOLD
ASSEMBLE THE 1401 RECORD INTO A 7090 RECORD STARTING AT THOLD.
STORE IN BBIN THE ADDRESS OF THE 1401 CHARACTER FOLLOWING THE GROUP MARK,W.M.
SELECT THE TAPE UNIT, LOAD THE CHANNEL
SW (UDISC)
WRITE PZE N,WHERE N=THE NUMBER OF VALID CHARACTERS IN THE LAST 7090 WORD.(SEE RTPDN)
WRITE THE RECORD BEGINNING AT THOLD
WAIT FOR COMPLETION
END OF REEL
SET SW (ENDRL)
REDUNDANCY
RESET SW (TAPER)
CTT - OP CODE U
LGTH =5
ERROR
2ND CHARACTER (
ERROR
3RD CHARACTER U OR B
ERROR
RWA
GET 7090 TAPE ADDRESS
DCHAR M,R,U B OR E
ERROR
LVSYS
WAIT TILL IOCS IS DORMANT, THEN DISABLE TRAPS.
WRITE T.M., REWIND, UNLOAD, BACKSPACE, OR ERASE, ACCORDING TO DCHAR
+
-



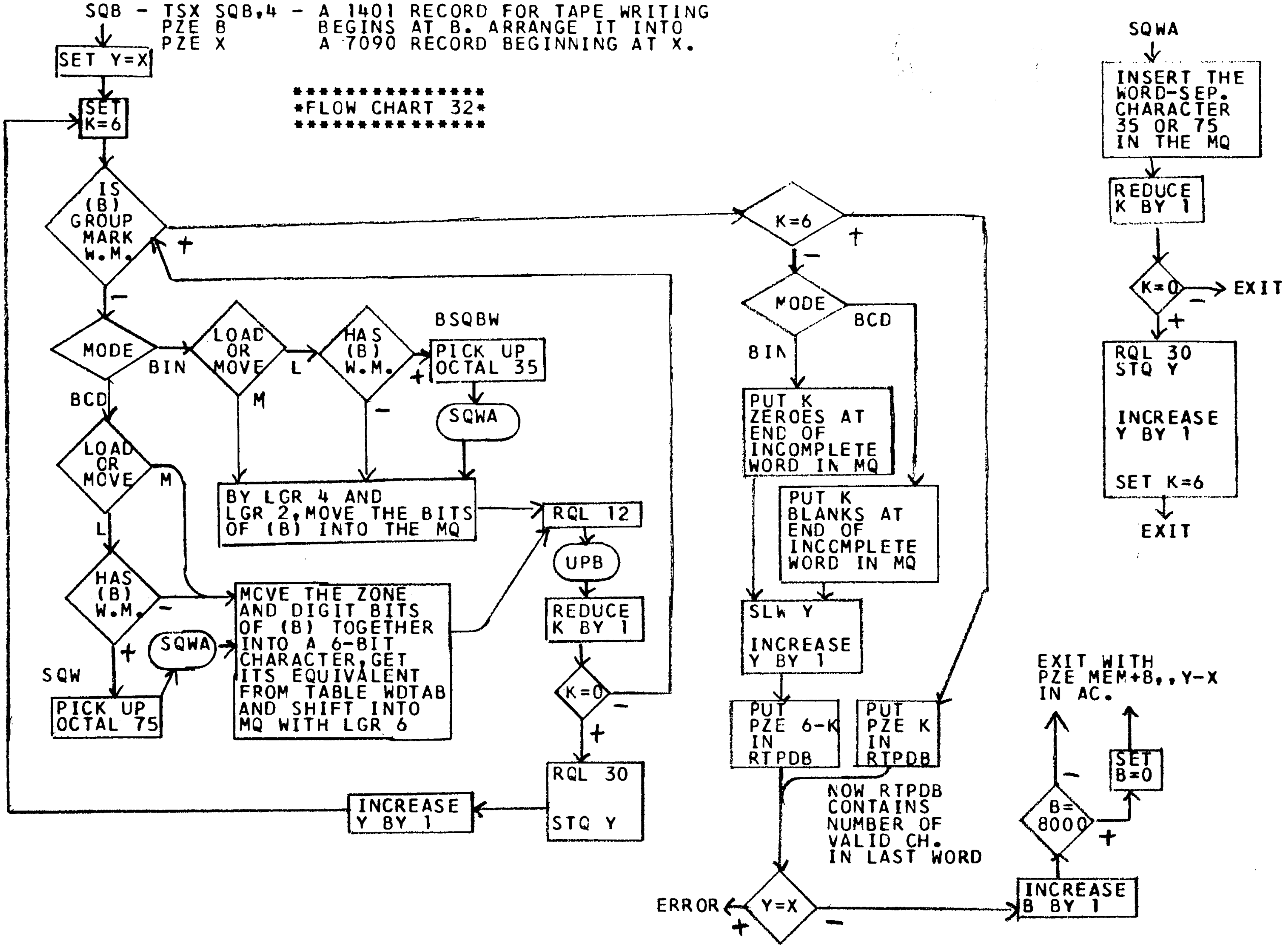
RWA - GET 7090 TAPE ADDRESS CORRESPONDING TO 1401 NUMBER AND MODE
GET 4TH CHARACTER OF INSTR-UCTION
IS IT A DIGIT BETWEEN 1 AND 6
-
ERROR
+
GET THE ADDRESS IN TAPNO+N-1, WHERE N IS THAT DIGIT
ADDRESS =0
+
ERROR - 1401 TAPE NUMBER NOT MENTIONED ON CONTROL CARD
-
LFT 400000
-
+
MAKE THE ADDRESS BINARY
LEAVE THE ADDRESS BCD
PUT IT IN RWB
EXIT

****
*FLOW *
CHART
* 30 *
****



SPB - TSX SPB,4 - A 7090 RECORD BEGINS IN CELL Y
PZE B
PZE Y,,Z
AND IS Z WORDS LONG.PUT THE EQUIVALENT 1401 RECORD IN CELLS B FF.,AND PUT A GROUP MARK AFTER THE LAST DATA CHARACTER.
SPBL
IS (B) A GROUP MARK, W.M.
+
-
MODE
BCD
SHIFT A 7090 BCD CHARACTER INTO AC,THEN GET ITS 1401 EQUIVALENT FROM RDTAB FF.
BIN
SPREAD A 7090 CHARACTER TO A 1401 CHAR.
MOVE OR LOAD
M
L
STORE AT B, EXCEPT W.M.
IS IT 0-8-5
+
-
STORE AT B, WITH NO W.M.
REDUCE K BY 1
K=0
+
-
INCREASE Y BY 1, DECREASE Z BY 1
Z=0
+
-
SPREAD A CHAR.(BIN)OR GET IT FROM RDTAB (BCD)
STORE AT B, WITH W.M.
STORE 0-8-5 AT B, NO W.M.
UPB
UPB
DECREASE K BY 1
K=0
-
+
INCREASE Y BY 1,DECREASE Z BY 1
Z=0
+
-
STORE GROUP MARK AT B, BUT LEAVE W.M. BIT UNDISTURBED
UPB
EXIT
WITH B IN AC
UPB
INCREASE B-ADDRESS BY 1
OVER 7999
+
ERROR
-
EXIT
UPA IS EXACTLY PARALLEL
SPBL
LDQ Y
SW (UDISC)
+
-
SET K=6
EXIT
Z=1
-
+
SET K= ADDRESS IN RTPDB
EXIT

****
FLOW CHART 31
****

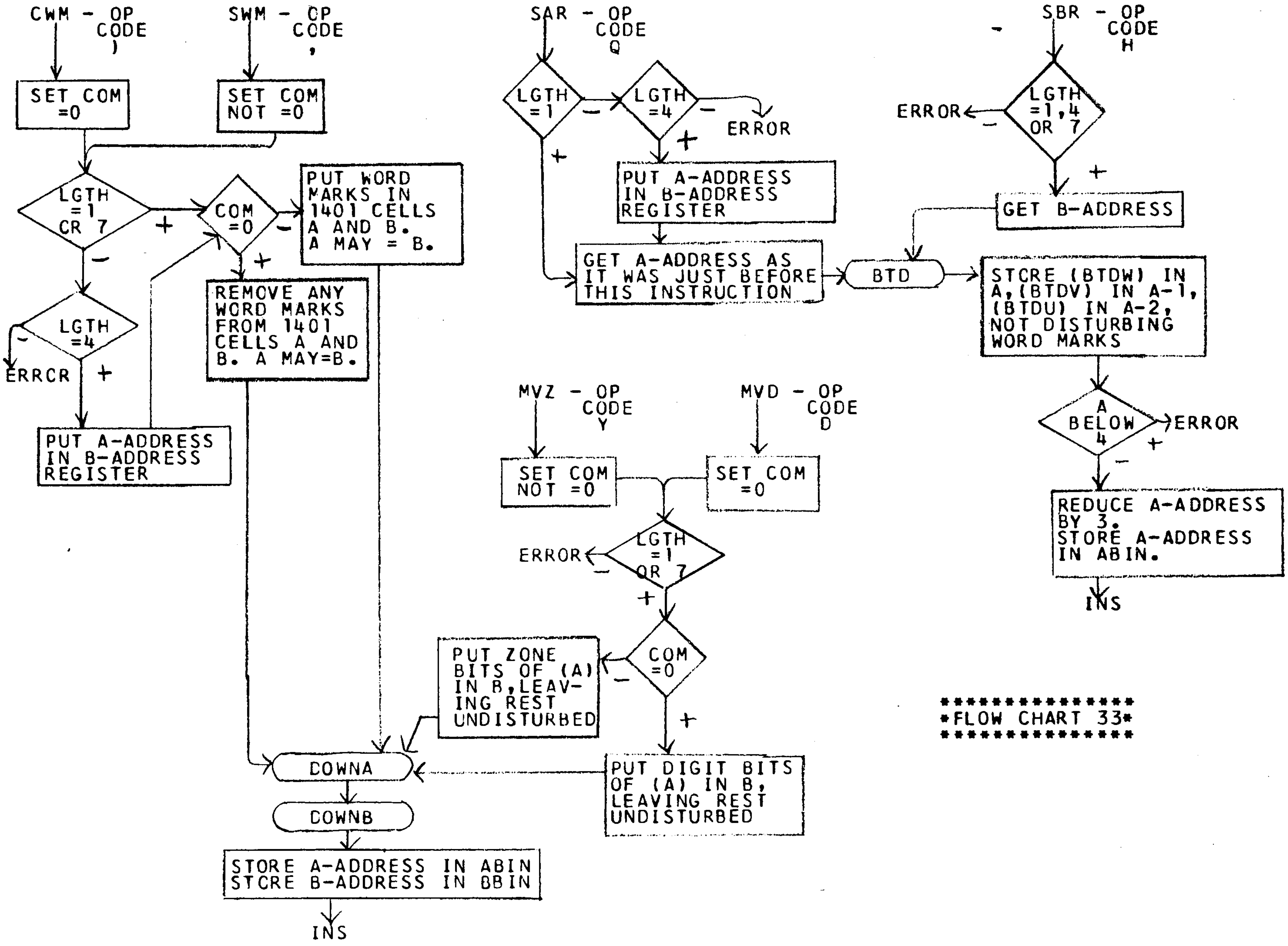


****

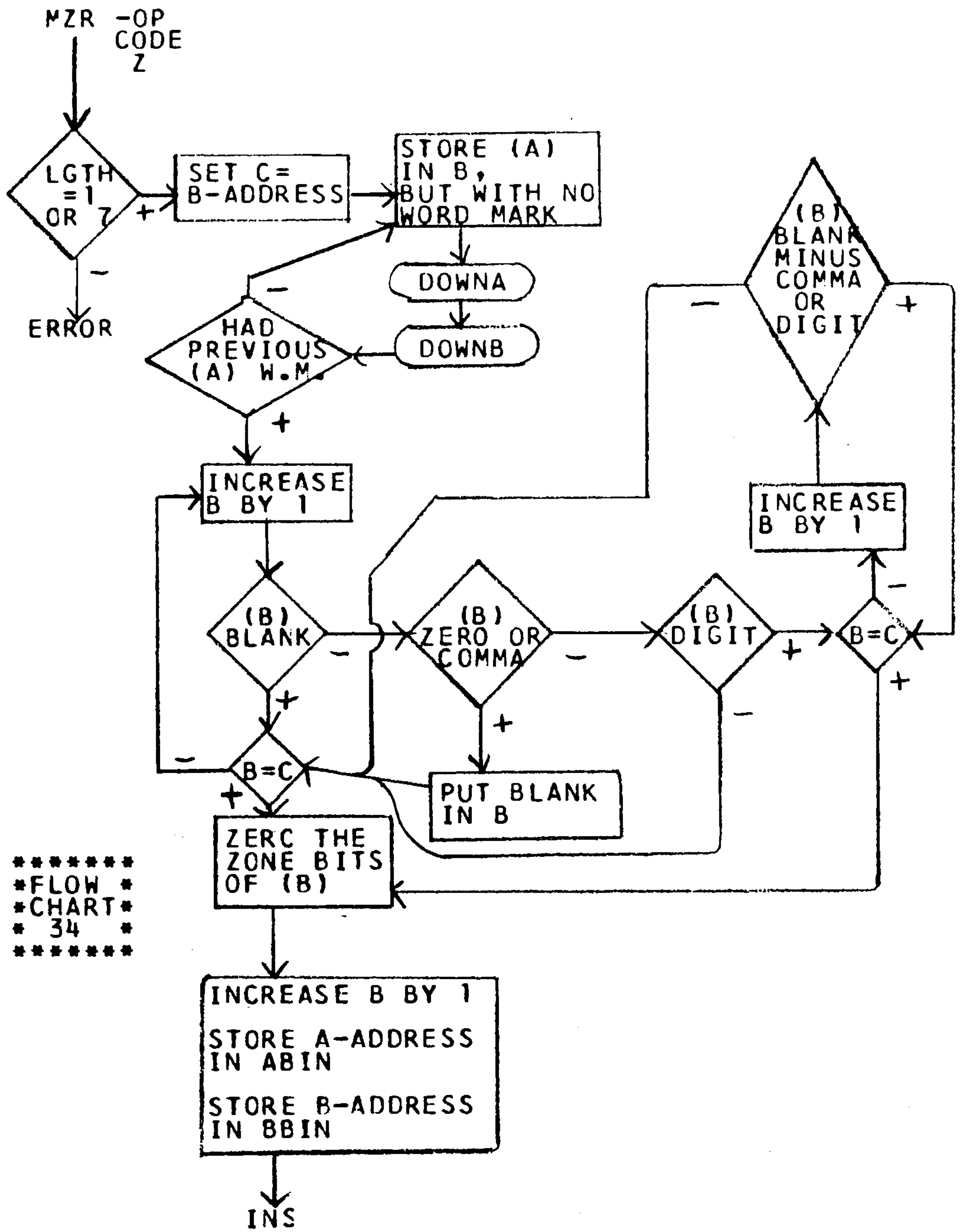
FLOW CHART 32

****

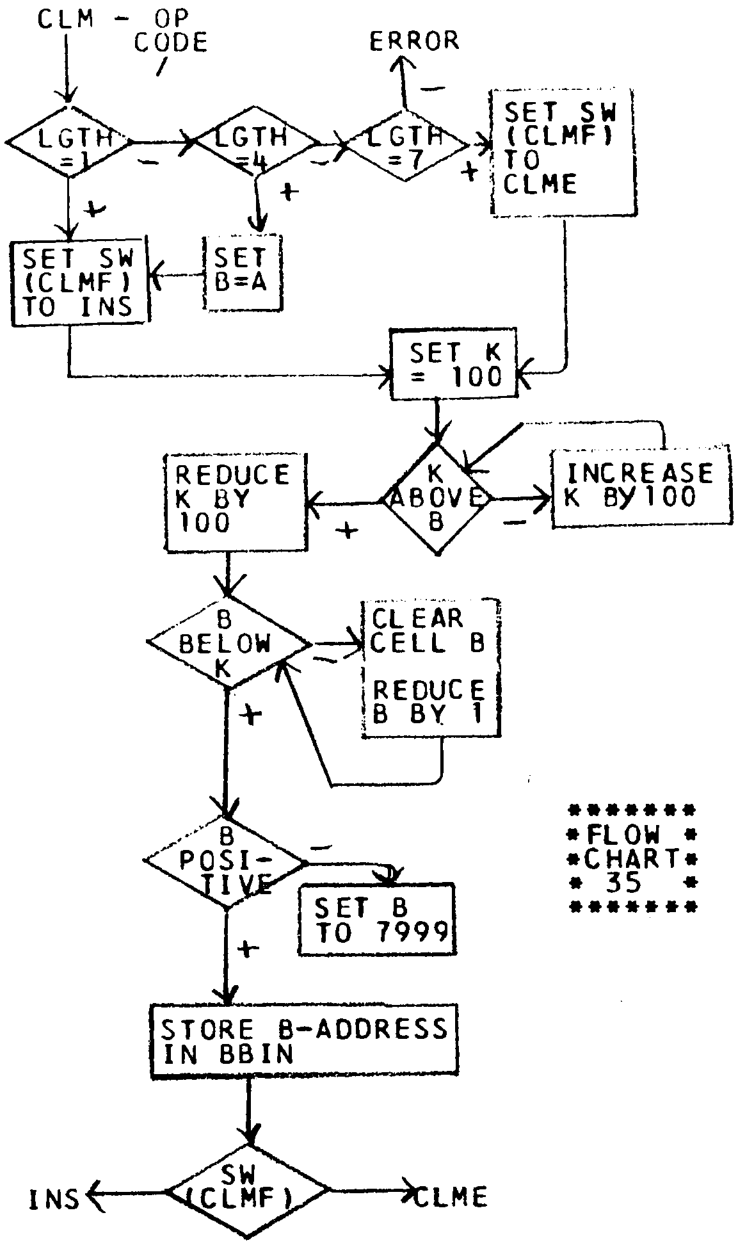
SQB - TSX SQB,4 - A 1401 RECORD FOR TAPE WRITING
PZE B BEGINS AT B. ARRANGE IT INTO
PZE X A 7090 RECORD BEGINNING AT X.
SET Y=X
SET K=6
IS (B) GROUP MARK W.M.
+
-
MODE
BIN
BCD
LOAD OR MOVE
L
M
HAS (B) W.M.
+
-
BSQBW
PICK UP OCTAL 35
SQWA
BY LGR 4 AND LGR 2, MOVE THE BITS OF (B) INTO THE MQ
LOAD OR MOVE
M
L
HAS (B) W.M.
-
+
SQW
PICK UP OCTAL 75
SQWA
MOVE THE ZONE AND DIGIT BITS OF (B) TOGETHER INTO A 6-BIT CHARACTER, GET ITS EQUIVALENT FROM TABLE WDTAB AND SHIFT INTO MQ WITH LGR 6
RQL 12
UPB
REDUCE K BY 1
K=0
-
+
RQL 30
STQ Y
INCREASE Y BY 1
K=6
+
-
MODE
BCD
BIN
PUT K ZEROES AT END OF INCOMPLETE WORD IN MQ
PUT K BLANKS AT END OF INCOMPLETE WORD IN MQ
SLW Y
INCREASE Y BY 1
PUT PZE 6-K IN RTPDB
PUT PZE K IN RTPDB
NOW RTPDB CONTAINS NUMBER OF VALID CH. IN LAST WORD
Y=X
ERROR
+
-
INCREASE B BY 1
B=8000
-
+
SET B=0
EXIT WITH PZE MEM+B,,Y-X IN AC.
SQWA
INSERT THE WORD-SEP. CHARACTER 35 OR 75 IN THE MQ
REDUCE K BY 1
K=0
EXIT
-
+
RQL 30
STQ Y
INCREASE Y BY 1
SET K=6
EXIT


CWM - OP CODE)
SET COM =0
SWM - OP CODE ,
SET COM NOT =0
LGTH =1 OR 7
+
-
COM =0
+
-
PUT WORD MARKS IN 1401 CELLS A AND B. A MAY = B.
REMOVE ANY WORD MARKS FROM 1401 CELLS A AND B. A MAY=B.
LGTH =4
-
ERRCR
+
PUT A-ADDRESS IN B-ADDRESS REGISTER
SAR - OP CODE Q
LGTH =1
-
+
LGTH =4
-
ERROR
+
PUT A-ADDRESS IN B-ADDRESS REGISTER
GET A-ADDRESS AS IT WAS JUST BEFORE THIS INSTRUCTION
BTD
SBR - OP CODE H
LGTH =1,4 OR 7
-
ERROR
+
GET B-ADDRESS
STCRE (BTDW) IN A, (BTDV) IN A-1, (BTDU) IN A-2, NOT DISTURBING WORD MARKS
A BELOW 4
+
ERROR
-
REDUCE A-ADDRESS BY 3. STORE A-ADDRESS IN ABIN.
INS
MVZ - OP CODE Y
SET COM NOT =0
MVD - OP CODE D
SET COM =0
LGTH =1 OR 7
-
ERROR
+
COM =0
-
PUT ZONE BITS OF (A) IN B, LEAV-ING REST UNDISTURBED
+
PUT DIGIT BITS OF (A) IN B, LEAVING REST UNDISTURBED
DOWNA
DOWNB
STORE A-ADDRESS IN ABIN
STCRE B-ADDRESS IN BBIN
INS

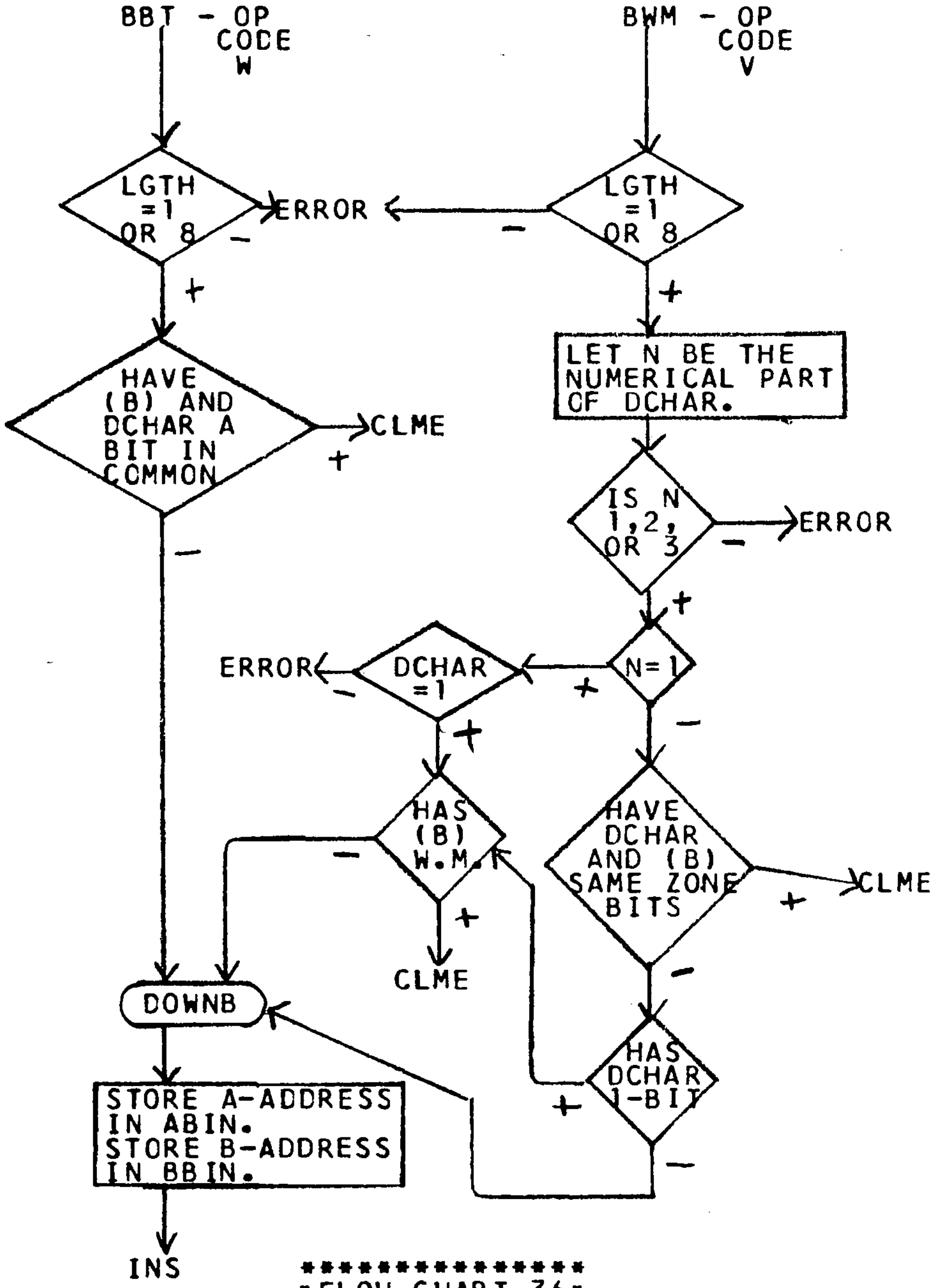
****
FLOW CHART 33
****



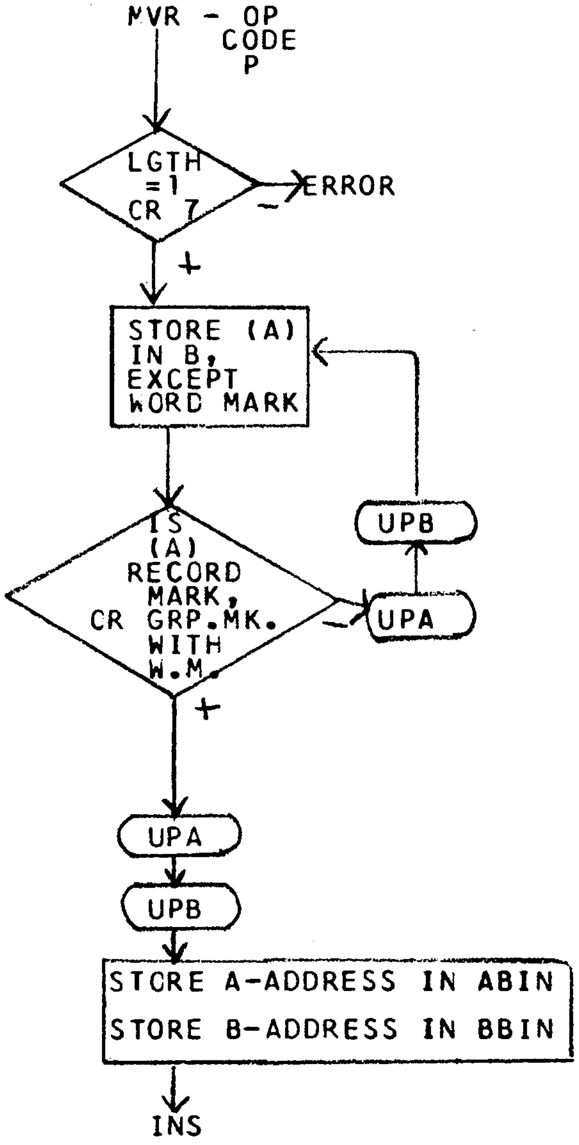
****
*FLOW *
CHART
* 34 *
****



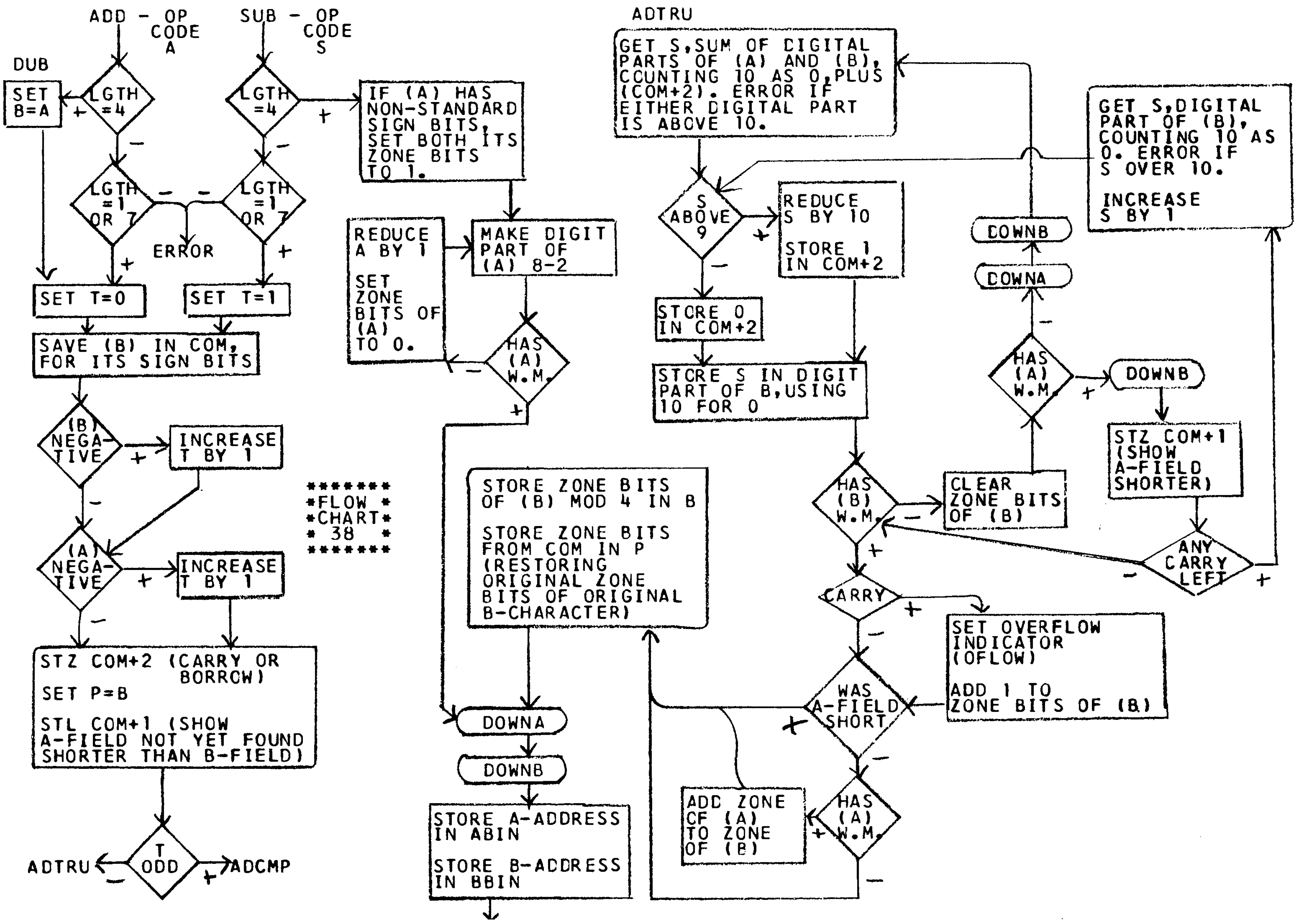
****
*FLOW *
CHART
* 35 *
****


BBT - OP CODE W
LGTH =1 OR 8
ERROR
+
-
HAVE (B) AND DCHAR A BIT IN COMMON
CLME
BWM - OP CODE V
LET N BE THE NUMERICAL PART OF DCHAR.
IS N 1,2, OR 3
ERROR
N=1
DCHAR =1
HAS (B) W.M.
CLME
HAVE DCHAR AND (B) SAME ZONE BITS
CLME
HAS DCHAR 1-BIT
DOWNB
STORE A-ADDRESS IN ABIN. STORE B-ADDRESS IN BBIN.
INS

****
FLOW CHART 36
****

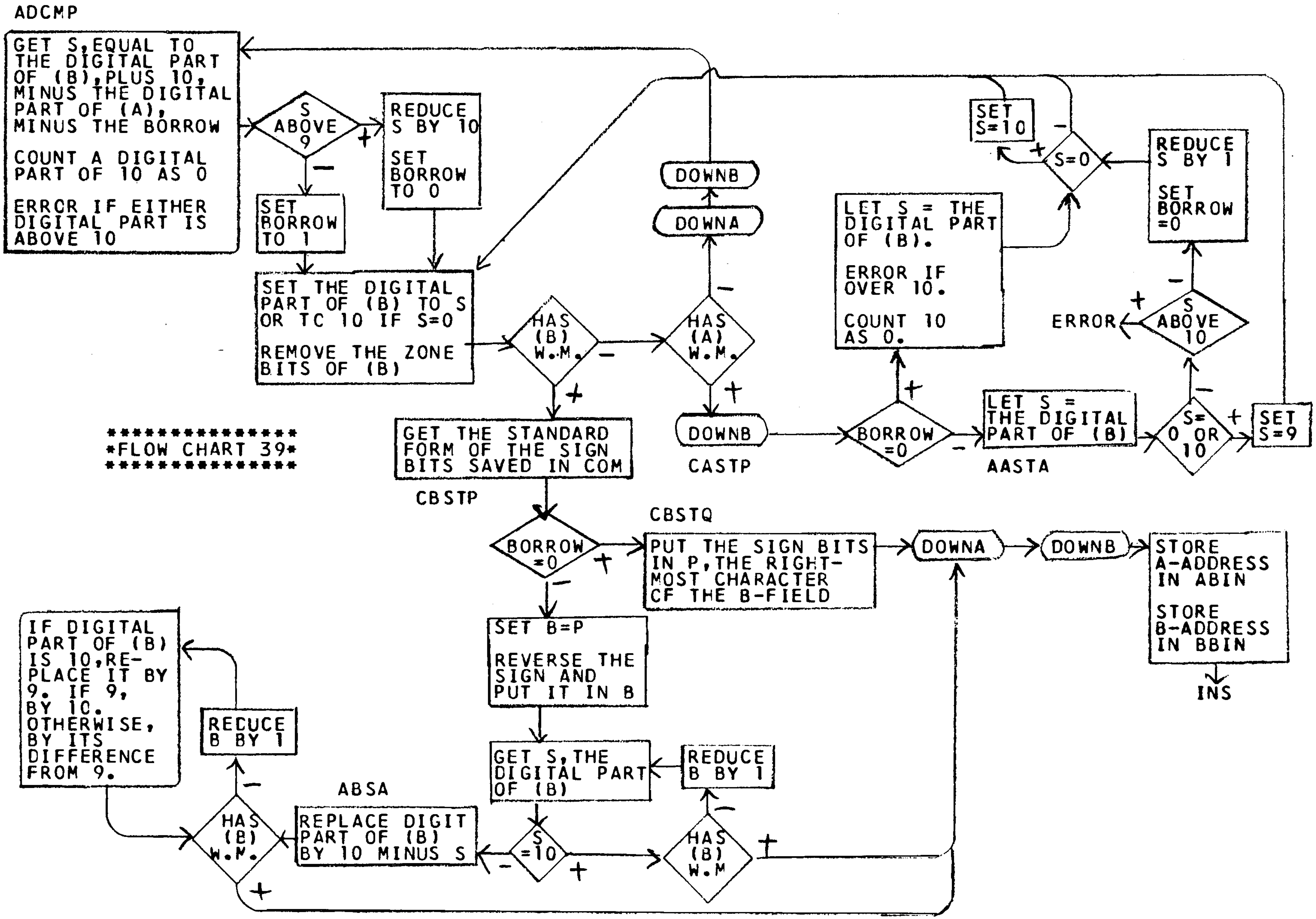


****
FLOW CHART 37
****

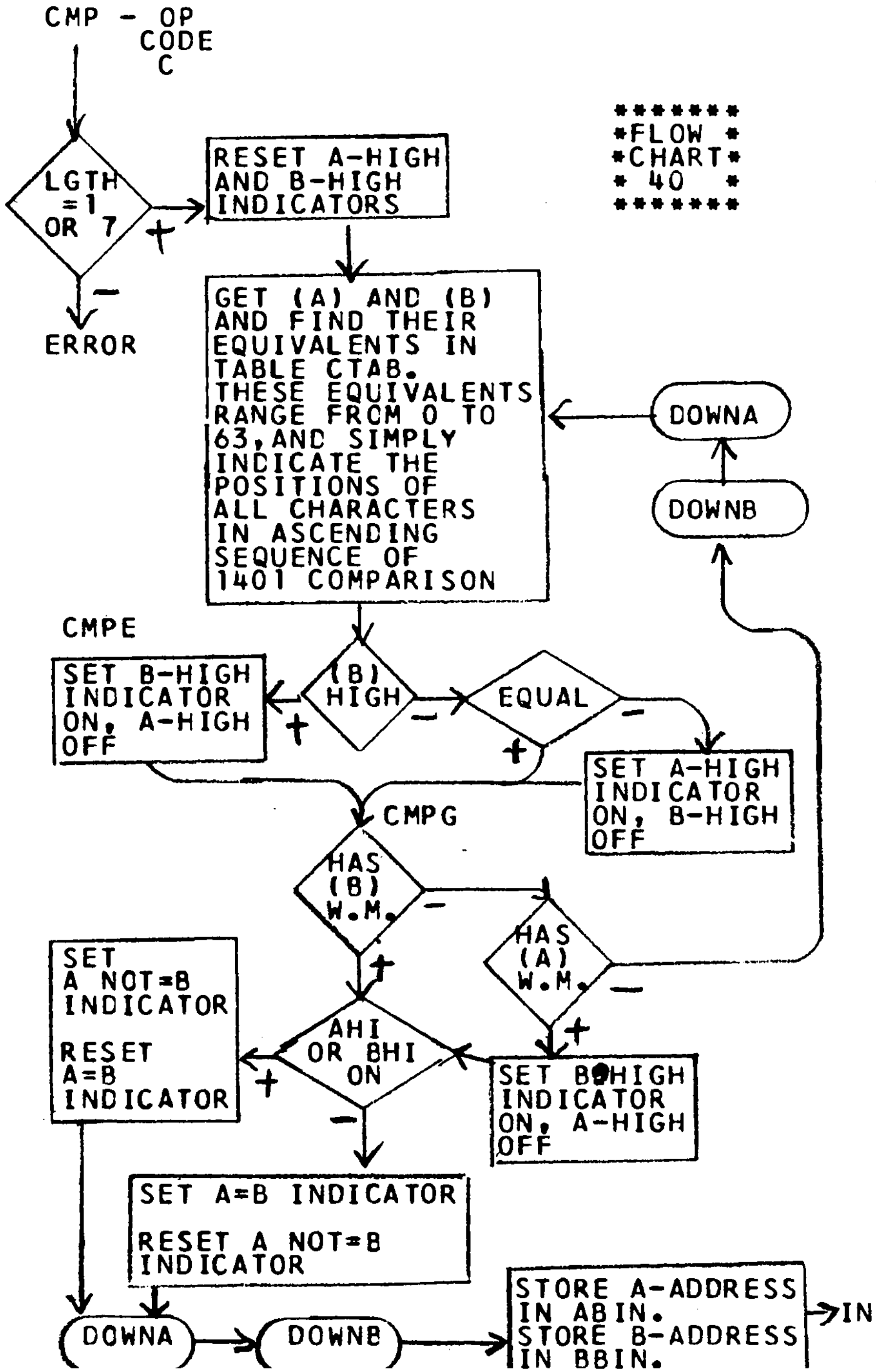


ADD - OP CODE A
SUB - OP CODE S
DUB
SET B=A
LGTH =4
+
-
LGTH =1 OR 7
ERROR
+
SET T=0
SET T=1
SAVE (B) IN COM, FOR ITS SIGN BITS
(B) NEGATIVE
INCREASE T BY 1
(A) NEGATIVE
INCREASE T BY 1
STZ COM+2 (CARRY OR BORROW)
SET P=B
STL COM+1 (SHOW A-FIELD NOT YET FOUND SHORTER THAN B-FIELD)
T ODD
ADTRU
ADCMP
IF (A) HAS NON-STANDARD SIGN BITS, SET BOTH ITS ZONE BITS TO 1.
REDUCE A BY 1
SET ZONE BITS OF (A) TO 0.
MAKE DIGIT PART OF (A) 8-2
HAS (A) W.M.
STORE ZONE BITS OF (B) MOD 4 IN B
STORE ZONE BITS FROM COM IN P (RESTORING ORIGINAL ZONE BITS OF ORIGINAL B-CHARACTER)
DOWNA
DOWNB
STORE A-ADDRESS IN ABIN
STORE B-ADDRESS IN BBIN
ADTRU
GET S, SUM OF DIGITAL PARTS OF (A) AND (B), COUNTING 10 AS 0, PLUS (COM+2). ERROR IF EITHER DIGITAL PART IS ABOVE 10.
S ABOVE 9
REDUCE S BY 10
STORE 1 IN COM+2
STORE 0 IN COM+2
STORE S IN DIGIT PART OF B, USING 10 FOR 0
HAS (B) W.M.
CLEAR ZONE BITS OF (B)
HAS (A) W.M.
DOWNA
DOWNB
DOWNB
STZ COM+1 (SHOW A-FIELD SHORTER)
ANY CARRY LEFT
GET S, DIGITAL PART OF (B), COUNTING 10 AS 0. ERROR IF S OVER 10.
INCREASE S BY 1
CARRY
SET OVERFLOW INDICATOR (OFLOW)
ADD 1 TO ZONE BITS OF (B)
WAS A-FIELD SHORT
HAS (A) W.M.
ADD ZONE OF (A) TO ZONE OF (B)

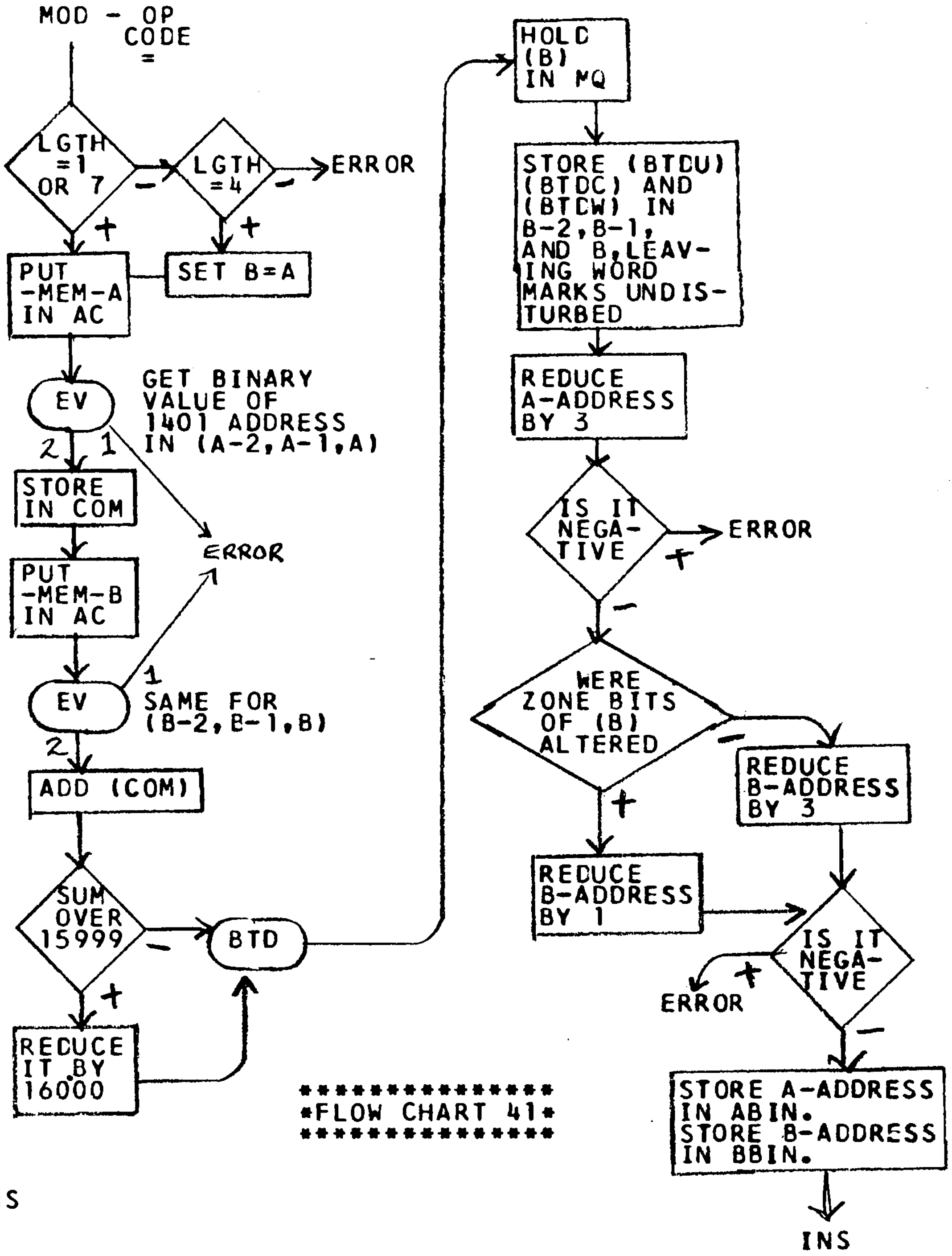
FLOW CHART 38


ADCMP
GET S,EQUAL TO THE DIGITAL PART OF (B),PLUS 10, MINUS THE DIGITAL PART OF (A), MINUS THE BORROW
COUNT A DIGITAL PART OF 10 AS 0
ERROR IF EITHER DIGITAL PART IS ABOVE 10
S ABOVE 9
+
−
REDUCE S BY 10
SET BORROW TO 0
SET BORROW TO 1
SET THE DIGITAL PART OF (B) TO S OR TC 10 IF S=0
REMOVE THE ZONE BITS OF (B)
HAS (B) W.M.
−
+
HAS (A) W.M.
−
+
DOWNA
DOWNB
DOWNB
CASTP
BORROW =0
+
−
LET S = THE DIGITAL PART OF (B).
ERROR IF OVER 10.
COUNT 10 AS 0.
LET S = THE DIGITAL PART OF (B)
AASTA
S= 0 OR 10
+
−
SET S=9
S ABOVE 10
+
−
ERROR
REDUCE S BY 1
SET BORROW =0
S=0
+
−
SET S=10
GET THE STANDARD FORM OF THE SIGN BITS SAVED IN COM
CBSTP
BORROW =0
+
−
CBSTQ
PUT THE SIGN BITS IN P,THE RIGHT-MOST CHARACTER CF THE B-FIELD
DOWNA
DOWNB
STORE A-ADDRESS IN ABIN
STORE B-ADDRESS IN BBIN
INS
SET B=P
REVERSE THE SIGN AND PUT IT IN B
GET S,THE DIGITAL PART OF (B)
REDUCE B BY 1
S =10
+
−
HAS (B) W.M
+
−
ABSA
REPLACE DIGIT PART OF (B) BY 10 MINUS S
HAS (B) W.M.
+
−
REDUCE B BY 1
IF DIGITAL PART OF (B) IS 10,RE-PLACE IT BY 9. IF 9, BY 1C. OTHERWISE, BY ITS DIFFERENCE FROM 9.

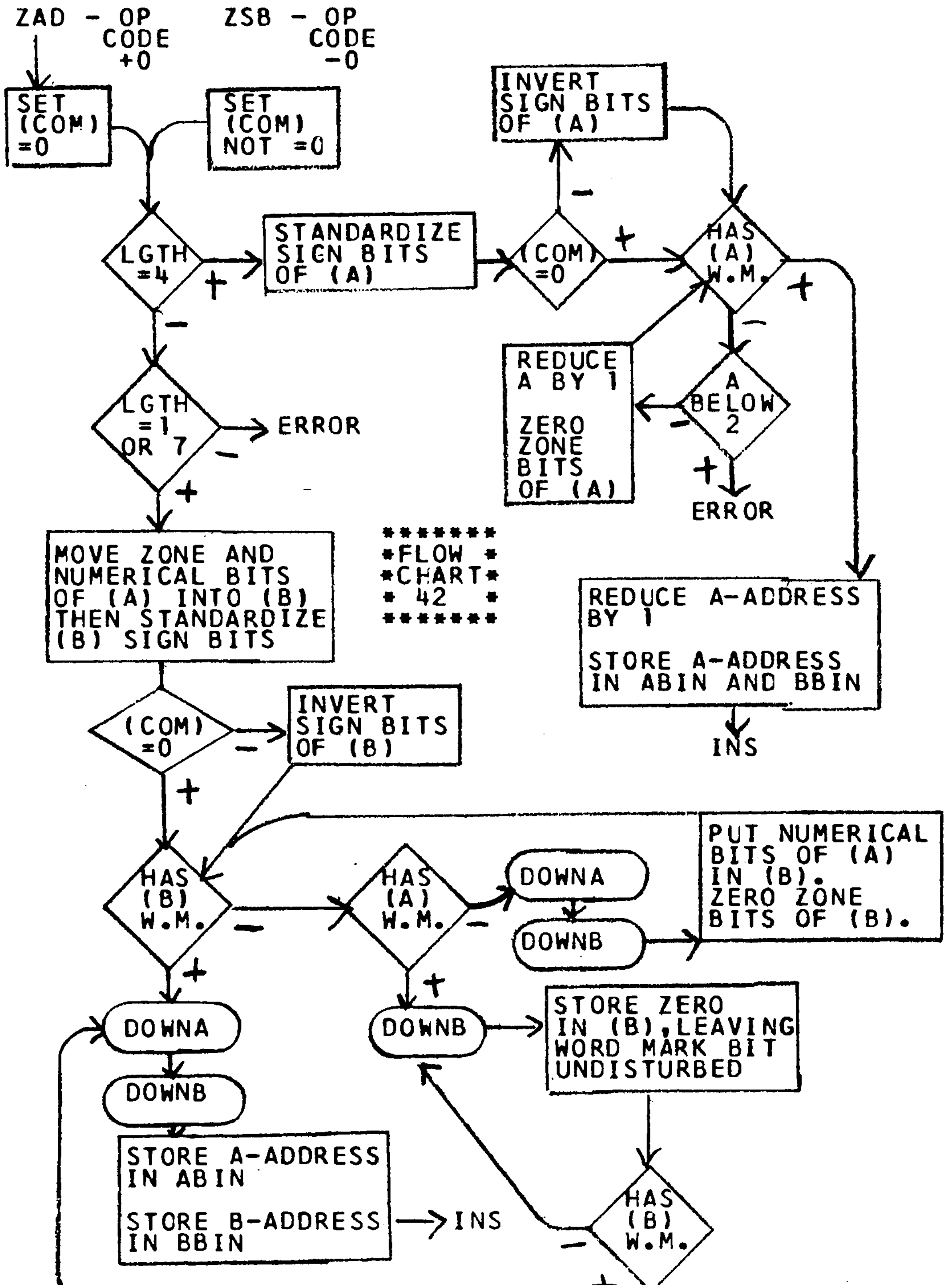
****
FLOW CHART 39
****


CMP - OP CODE C
LGTH =1 OR 7
+
-
ERROR
RESET A-HIGH AND B-HIGH INDICATORS
GET (A) AND (B) AND FIND THEIR EQUIVALENTS IN TABLE CTAB. THESE EQUIVALENTS RANGE FROM 0 TO 63, AND SIMPLY INDICATE THE POSITIONS OF ALL CHARACTERS IN ASCENDING SEQUENCE OF 1401 COMPARISON
DOWNA
DOWNB
CMPE
SET B-HIGH INDICATOR ON, A-HIGH OFF
(B) HIGH
+
-
EQUAL
+
-
SET A-HIGH INDICATOR ON, B-HIGH OFF
CMPG
HAS (B) W.M.
-
+
HAS (A) W.M.
-
+
SET A NOT=B INDICATOR
RESET A=B INDICATOR
AHI OR BHI ON
+
-
SET B-HIGH INDICATOR ON, A-HIGH OFF
SET A=B INDICATOR
RESET A NOT=B INDICATOR
DOWNA
DOWNB
STORE A-ADDRESS IN ABIN. STORE B-ADDRESS IN BBIN.
INS

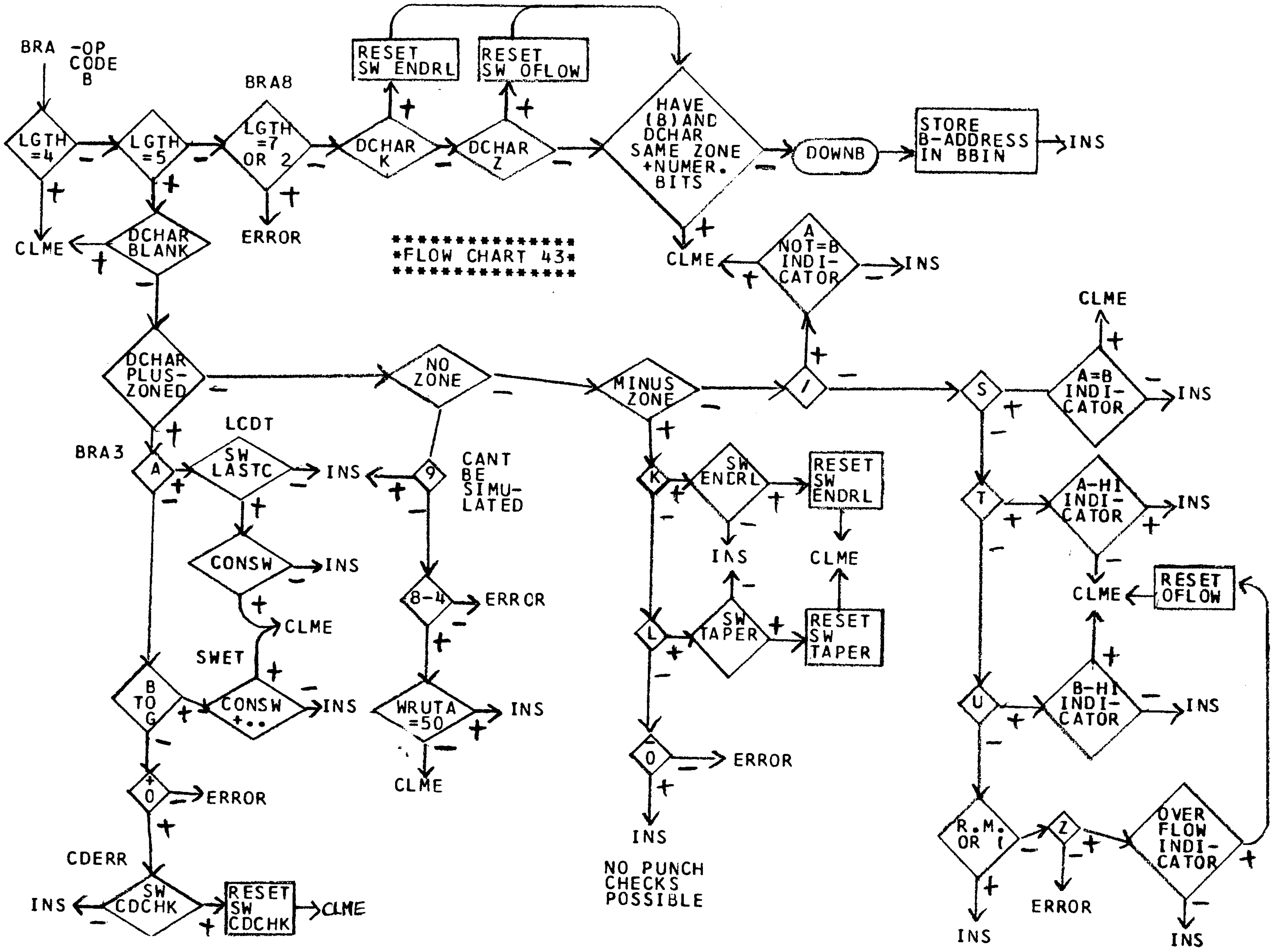
****
*FLOW *
CHART
* 40 *
****



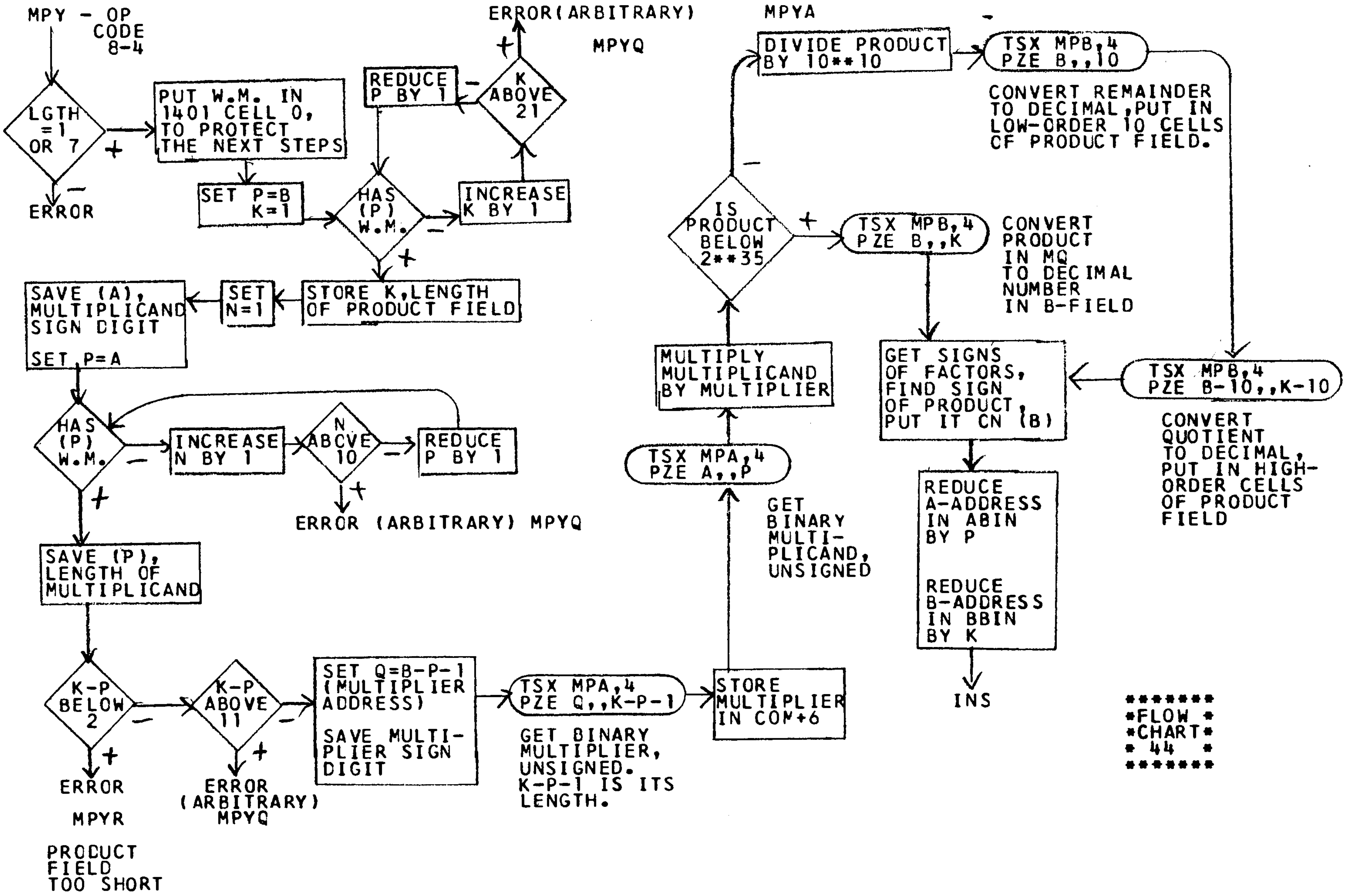
****
FLOW CHART 41
****



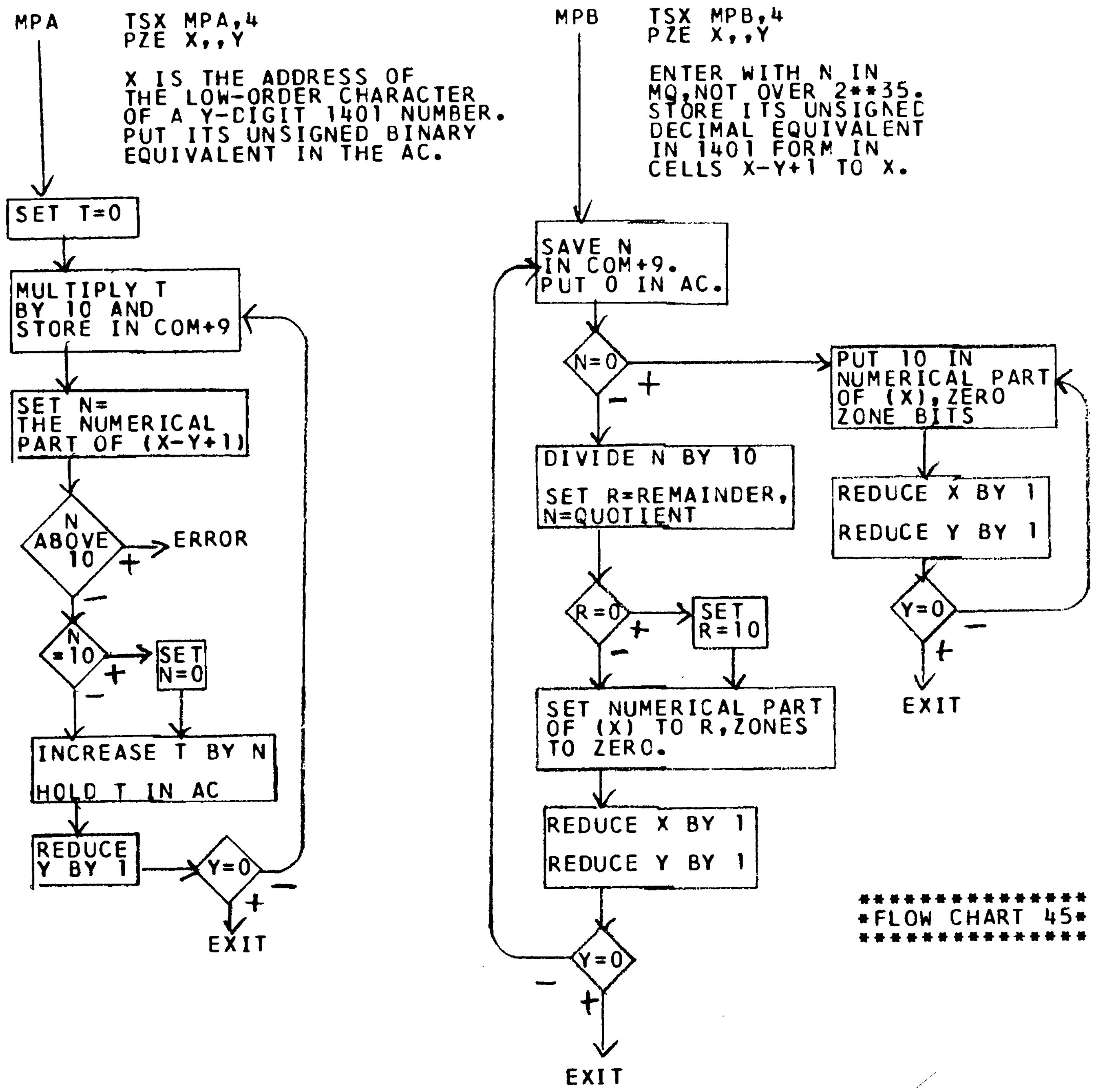
****
*FLOW *
CHART
* 42 *
****



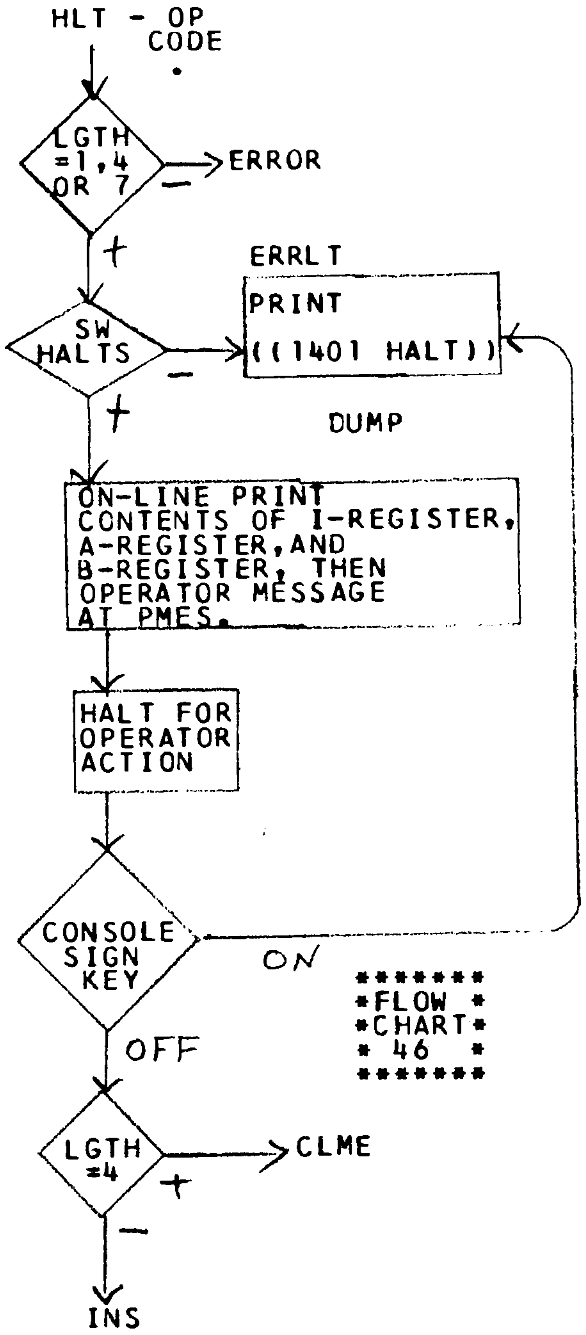
FLOW CHART 43


MPY - OP CODE 8-4
LGTH =1 OR 7
+
-
ERROR
PUT W.M. IN 1401 CELL 0, TO PROTECT THE NEXT STEPS
SET P=B K=1
HAS (P) W.M.
+
-
INCREASE K BY 1
K ABOVE 21
+
-
ERROR(ARBITRARY)
MPYQ
REDUCE P BY 1
STORE K,LENGTH OF PRODUCT FIELD
SET N=1
SAVE (A), MULTIPLICAND SIGN DIGIT
SET P=A
HAS (P) W.M.
-
+
INCREASE N BY 1
N ABOVE 10
-
+
REDUCE P BY 1
ERROR (ARBITRARY) MPYQ
SAVE (P), LENGTH OF MULTIPLICAND
K-P BELOW 2
+
-
ERROR
MPYR
PRODUCT FIELD TOO SHORT
K-P ABOVE 11
+
-
ERROR (ARBITRARY) MPYQ
SET Q=B-P-1 (MULTIPLIER ADDRESS)
SAVE MULTIPLIER SIGN DIGIT
TSX MPA,4 PZE Q,,K-P-1
GET BINARY MULTIPLIER, UNSIGNED. K-P-1 IS ITS LENGTH.
STORE MULTIPLIER IN COM+6
TSX MPA,4 PZE A,,P
GET BINARY MULTI-PLICAND, UNSIGNED
MULTIPLY MULTIPLICAND BY MULTIPLIER
IS PRODUCT BELOW 2**35
+
-
MPYA
DIVIDE PRODUCT BY 10**10
TSX MPB,4 PZE B,,10
CONVERT REMAINDER TO DECIMAL,PUT IN LOW-ORDER 10 CELLS OF PRODUCT FIELD.
TSX MPB,4 PZE B-10,,K-10
CONVERT QUOTIENT TO DECIMAL, PUT IN HIGH-ORDER CELLS OF PRODUCT FIELD
TSX MPB,4 PZE B,,K
CONVERT PRODUCT IN MQ TO DECIMAL NUMBER IN B-FIELD
GET SIGNS OF FACTORS, FIND SIGN OF PRODUCT, PUT IT ON (B)
REDUCE A-ADDRESS IN ABIN BY P
REDUCE B-ADDRESS IN BBIN BY K
INS

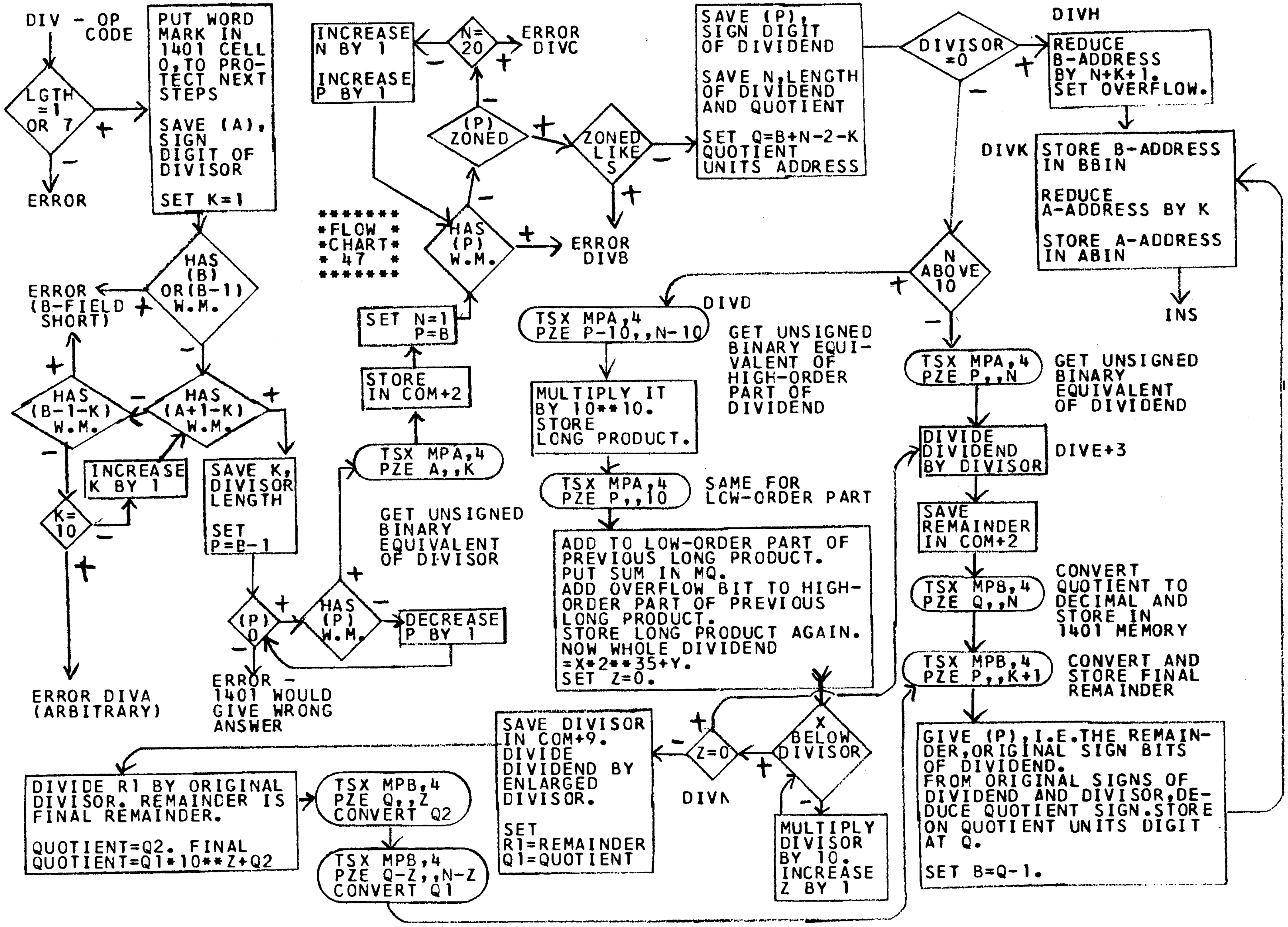
****
*FLOW *
CHART
* 44 *
****

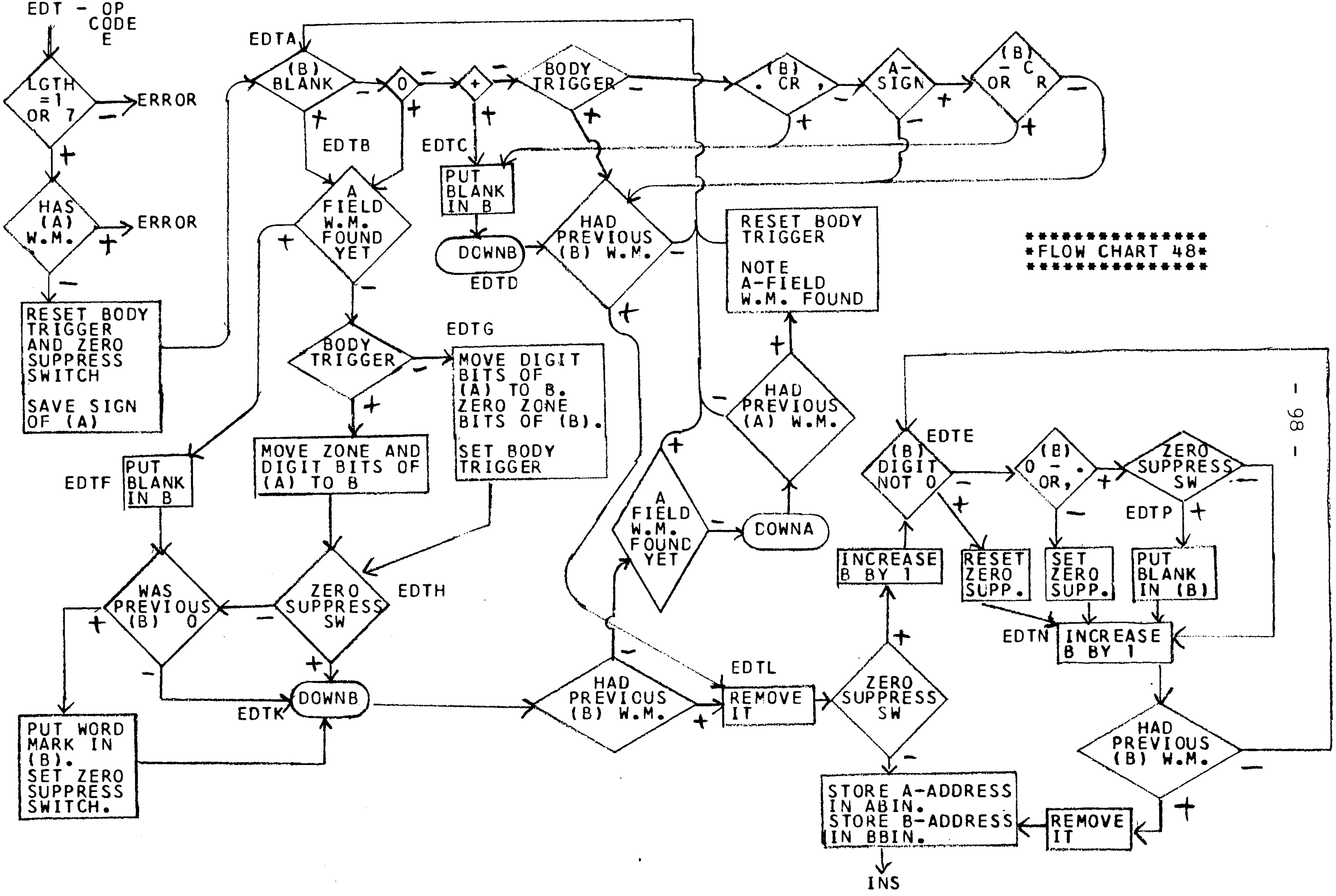

MPA
TSX MPA,4
PZE X,,Y
X IS THE ADDRESS OF THE LOW-ORDER CHARACTER OF A Y-DIGIT 1401 NUMBER. PUT ITS UNSIGNED BINARY EQUIVALENT IN THE AC.
SET T=0
MULTIPLY T BY 10 AND STORE IN COM+9
SET N= THE NUMERICAL PART OF (X-Y+1)
N ABOVE 10
+
ERROR
-
N =10
+
SET N=0
-
INCREASE T BY N
HOLD T IN AC
REDUCE Y BY 1
Y=0
-
+
EXIT
MPB
TSX MPB,4
PZE X,,Y
ENTER WITH N IN MQ,NOT OVER 2**35. STORE ITS UNSIGNED DECIMAL EQUIVALENT IN 1401 FORM IN CELLS X-Y+1 TO X.
SAVE N IN COM+9. PUT 0 IN AC.
N=0
+
-
PUT 10 IN NUMERICAL PART OF (X),ZERO ZONE BITS
REDUCE X BY 1
REDUCE Y BY 1
Y=0
-
+
EXIT
DIVIDE N BY 10
SET R=REMAINDER, N=QUOTIENT
R=0
+
SET R=10
-
SET NUMERICAL PART OF (X) TO R,ZONES TO ZERO.
REDUCE X BY 1
REDUCE Y BY 1
Y=0
-
+
EXIT

****
FLOW CHART 45
****

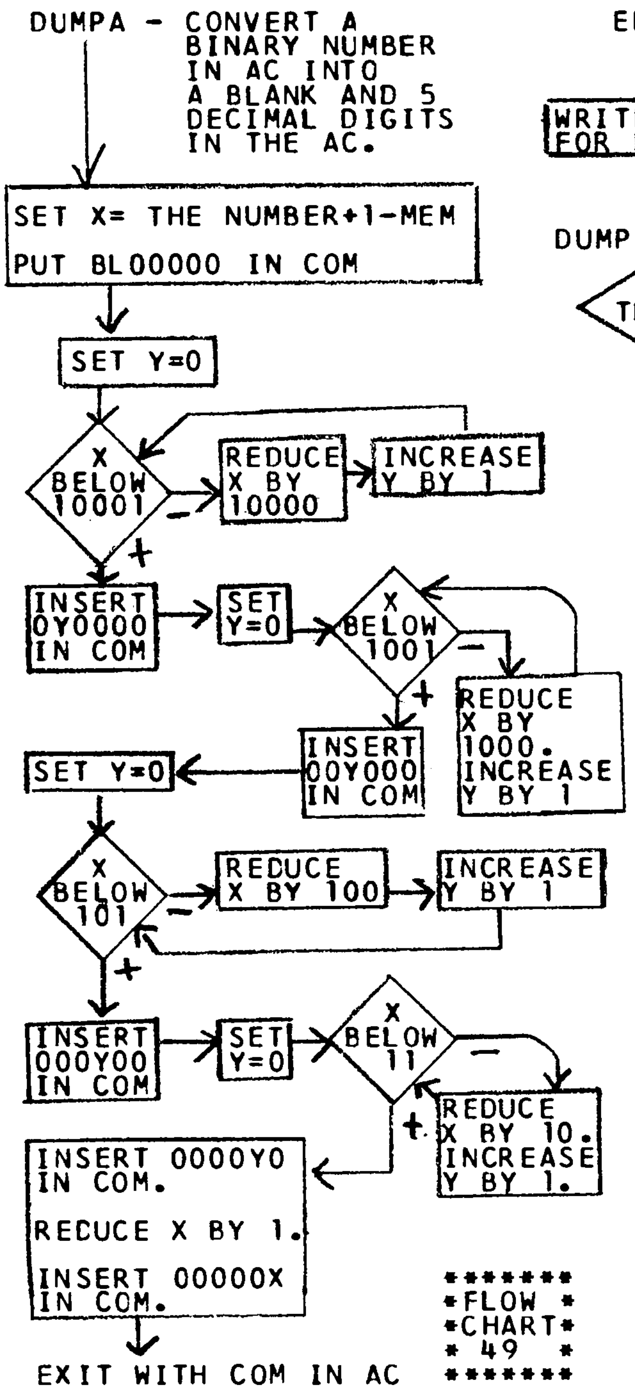


****
*FLOW *
CHART
* 46 *
****

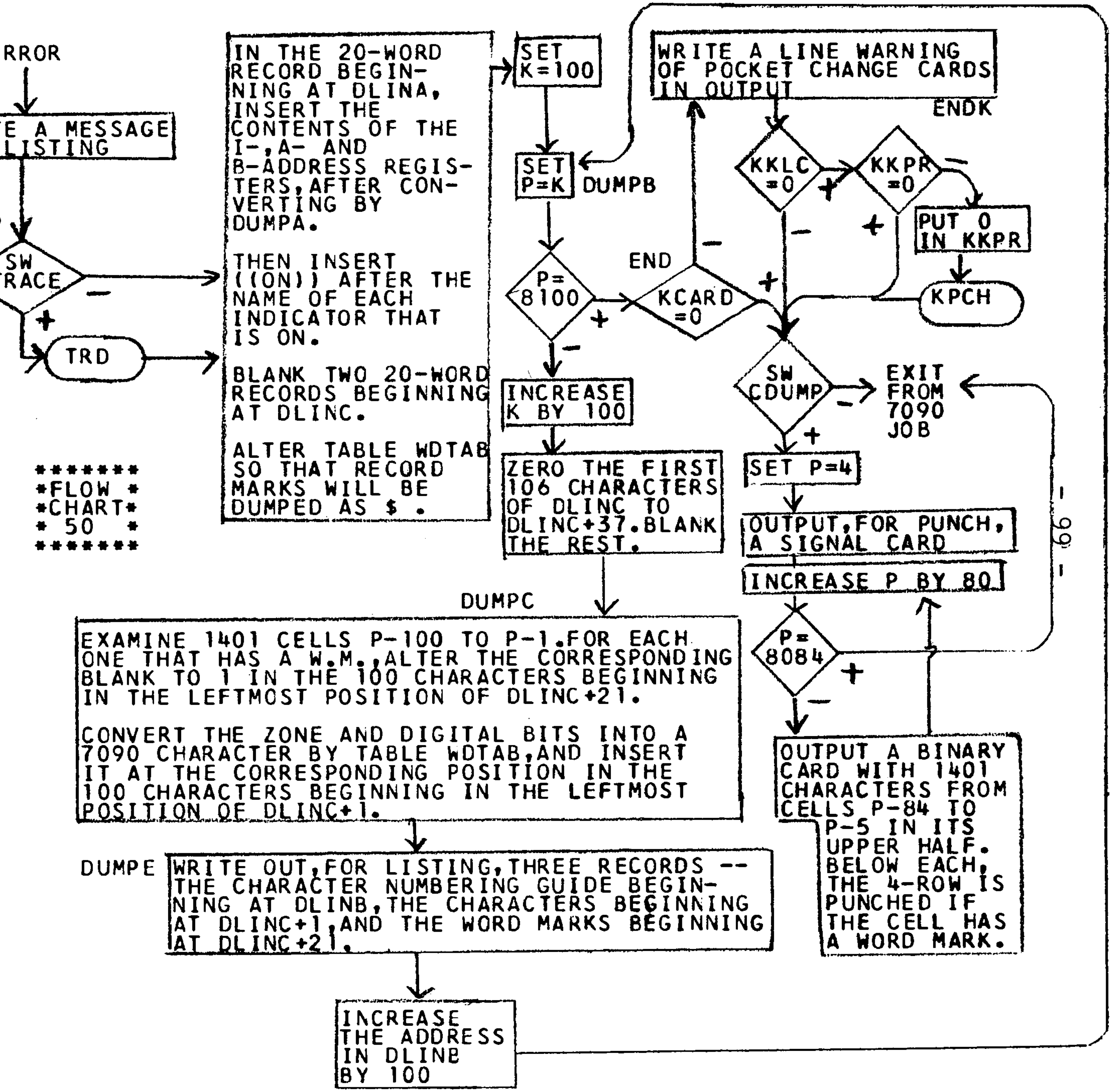

DIV - OP CODE
LGTH =1 OR 7
+
-
ERROR
PUT WORD MARK IN 1401 CELL 0,TO PROTECT NEXT STEPS
SAVE (A), SIGN DIGIT OF DIVISOR
SET K=1
HAS (B) OR(B-1) W.M.
ERROR (B-FIELD SHORT)
+
-
HAS (B-1-K) W.M.
HAS (A+1-K) W.M.
INCREASE K BY 1
K=10
ERROR DIVA (ARBITRARY)
SAVE K, DIVISOR LENGTH
SET P=B-1
(P) 0
HAS (P) W.M.
DECREASE P BY 1
ERROR - 1401 WOULD GIVE WRONG ANSWER
TSX MPA,4 PZE A,,K
GET UNSIGNED BINARY EQUIVALENT OF DIVISOR
STORE IN COM+2
SET N=1 P=B
****
*FLOW *
CHART
* 47 *
****
HAS (P) W.M.
ERROR DIVB
(P) ZONED
INCREASE N BY 1
INCREASE P BY 1
N= 20
ERROR DIVC
ZONED LIKE S
SAVE (P), SIGN DIGIT OF DIVIDEND
SAVE N,LENGTH OF DIVIDEND AND QUOTIENT
SET Q=B+N-2-K QUOTIENT UNITS ADDRESS
DIVISOR =0
DIVH
REDUCE B-ADDRESS BY N+K+1. SET OVERFLOW.
DIVK
STORE B-ADDRESS IN BBIN
REDUCE A-ADDRESS BY K
STORE A-ADDRESS IN ABIN
INS
N ABOVE 10
DIVD
TSX MPA,4 PZE P-10,,N-10
GET UNSIGNED BINARY EQUIVALENT OF HIGH-ORDER PART OF DIVIDEND
MULTIPLY IT BY 10**10. STORE LONG PRODUCT.
TSX MPA,4 PZE P,,10
SAME FOR LOW-ORDER PART
ADD TO LOW-ORDER PART OF PREVIOUS LONG PRODUCT. PUT SUM IN MQ. ADD OVERFLOW BIT TO HIGH-ORDER PART OF PREVIOUS LONG PRODUCT. STORE LONG PRODUCT AGAIN. NOW WHOLE DIVIDEND =X*2**35+Y. SET Z=0.
TSX MPA,4 PZE P,,N
GET UNSIGNED BINARY EQUIVALENT OF DIVIDEND
DIVIDE DIVIDEND BY DIVISOR
DIVE+3
SAVE REMAINDER IN COM+2
TSX MPB,4 PZE Q,,N
CONVERT QUOTIENT TO DECIMAL AND STORE IN 1401 MEMORY
TSX MPB,4 PZE P,,K+1
CONVERT AND STORE FINAL REMAINDER
GIVE (P),I.E.THE REMAINDER,ORIGINAL SIGN BITS OF DIVIDEND. FROM ORIGINAL SIGNS OF DIVIDEND AND DIVISOR,DEDUCE QUOTIENT SIGN.STORE ON QUOTIENT UNITS DIGIT AT Q.
SET B=Q-1.
X BELOW DIVISOR
MULTIPLY DIVISOR BY 10. INCREASE Z BY 1
Z=0
DIVN
SAVE DIVISOR IN COM+9. DIVIDE DIVIDEND BY ENLARGED DIVISOR.
SET R1=REMAINDER Q1=QUOTIENT
DIVIDE R1 BY ORIGINAL DIVISOR. REMAINDER IS FINAL REMAINDER.
QUOTIENT=Q2. FINAL QUOTIENT=Q1*10**Z+Q2
TSX MPB,4 PZE Q,,Z CONVERT Q2
TSX MPB,4 PZE Q-Z,,N-Z CONVERT Q1



****
FLOW CHART 48
****



****
*FLOW *
CHART
* 49 *
****


ERROR
WRITE A MESSAGE FOR LISTING
DUMP
SW TRACE
-
+
TRD
IN THE 20-WORD RECORD BEGINNING AT DLINA, INSERT THE CONTENTS OF THE I-,A- AND B-ADDRESS REGISTERS,AFTER CONVERTING BY DUMPA.
THEN INSERT ((ON)) AFTER THE NAME OF EACH INDICATOR THAT IS ON.
BLANK TWO 20-WORD RECORDS BEGINNING AT DLINC.
ALTER TABLE WDTAB SO THAT RECORD MARKS WILL BE DUMPED AS $.
SET K=100
SET P=K
DUMPB
P= 8100
INCREASE K BY 100
ZERO THE FIRST 106 CHARACTERS OF DLINC TO DLINC+37.BLANK THE REST.
DUMPC
EXAMINE 1401 CELLS P-100 TO P-1.FOR EACH ONE THAT HAS A W.M.,ALTER THE CORRESPONDING BLANK TO 1 IN THE 100 CHARACTERS BEGINNING IN THE LEFTMOST POSITION OF DLINC+21.
CONVERT THE ZONE AND DIGITAL BITS INTO A 7090 CHARACTER BY TABLE WDTAB,AND INSERT IT AT THE CORRESPONDING POSITION IN THE 100 CHARACTERS BEGINNING IN THE LEFTMOST POSITION OF DLINC+1.
DUMPE
WRITE OUT,FOR LISTING,THREE RECORDS -- THE CHARACTER NUMBERING GUIDE BEGINNING AT DLINB,THE CHARACTERS BEGINNING AT DLINC+1,AND THE WORD MARKS BEGINNING AT DLINC+21.
INCREASE THE ADDRESS IN DLINB BY 100
END
KCARD =0
WRITE A LINE WARNING OF POCKET CHANGE CARDS IN OUTPUT
ENDK
KKLC =0
KKPR =0
PUT 0 IN KKPR
KPCH
SW CDUMP
EXIT FROM 7090 JOB
SET P=4
OUTPUT,FOR PUNCH, A SIGNAL CARD
INCREASE P BY 80
P= 8084
OUTPUT A BINARY CARD WITH 1401 CHARACTERS FROM CELLS P-84 TO P-5 IN ITS UPPER HALF. BELOW EACH, THE 4-ROW IS PUNCHED IF THE CELL HAS A WORD MARK.
+
-

****
*FLOW *
CHART
* 50 *
****

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel

SALES OFFICES

All Euratom reports are on sale at the offices listed below, at the prices given on the back of the cover (when ordering, specify clearly the EUR number and the title of the report, which are shown on the cover).

PRESSES ACADEMIQUES EUROPEENNES
98, Chaussée de Charleroi, Bruxelles 6

Banque de la Société Générale - Bruxelles
compte N° 964.558,

Banque Belgo Congolaise - Bruxelles
compte N° 2444.141,

Compte chèque postal - Bruxelles - N° 167.37,

Belgian American Bank and Trust Company - New York
compte No. 22.186,

Lloyds Bank (Europe) Ltd. - 10 Moorgate, London E.C.2,

Postcheckkonto - Köln - Nr. 160.861.

OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
2, place de Metz, Luxembourg (Compte chèque postal N° 191-90)

BELGIQUE — BELGIË
MONITEUR BELGE
40-42, rue de Louvain - Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - Brussel

DEUTSCHLAND
BUNDESANZEIGER
Postfach - Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE
DES PUBLICATIONS DES
COMMUNAUTES EUROPEENNES
26, rue Desaix - Paris 15e

GRAND-DUCHE DE LUXEMBOURG
OFFICE CENTRAL DE VENTE
DES PUBLICATIONS DES
COMMUNAUTES EUROPEENNES
9, rue Goethe - Luxembourg

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - Roma

NEDERLAND
STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

EURATOM — C.I.D.
51-53, rue Belliard
Bruxelles (Belgique)