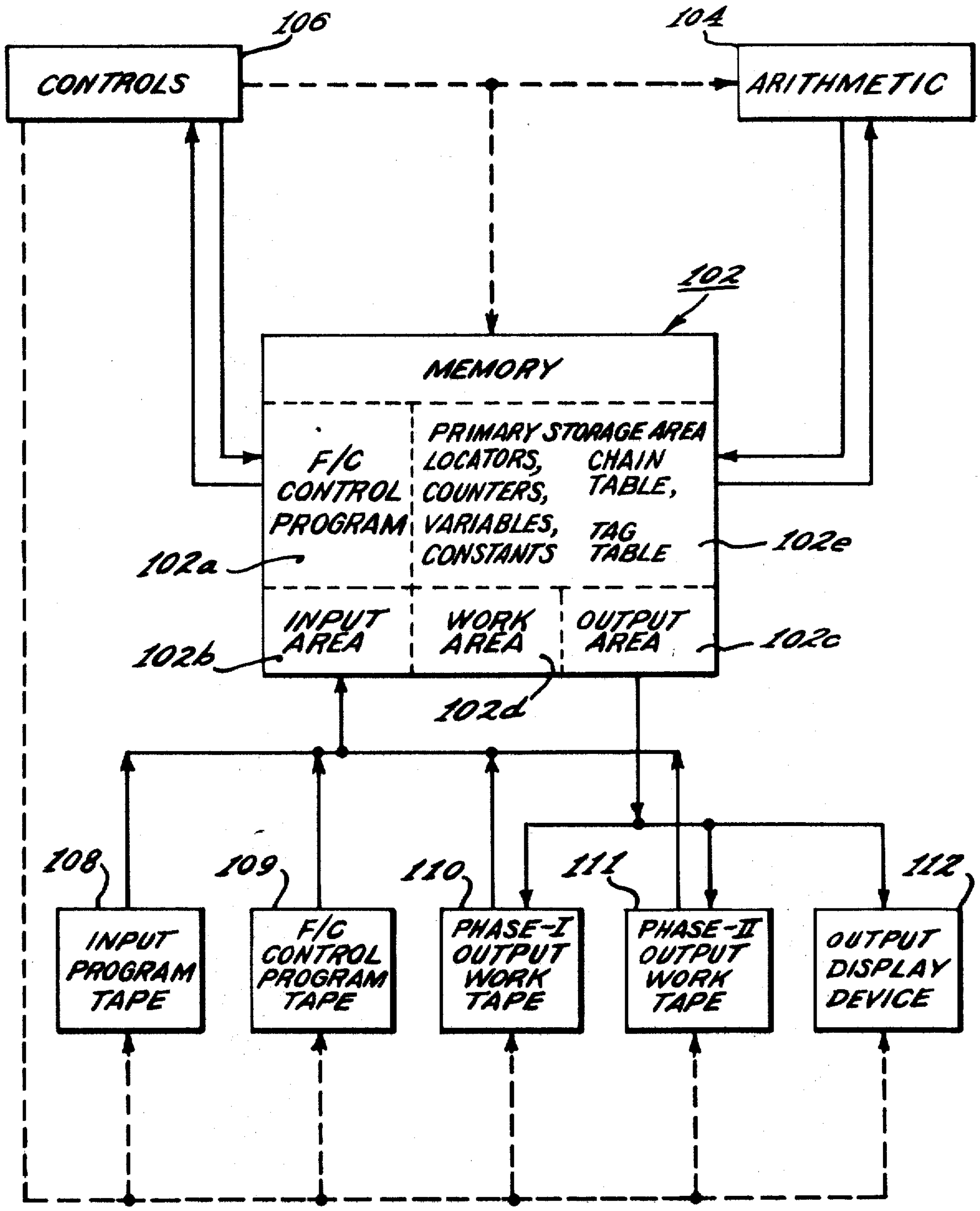
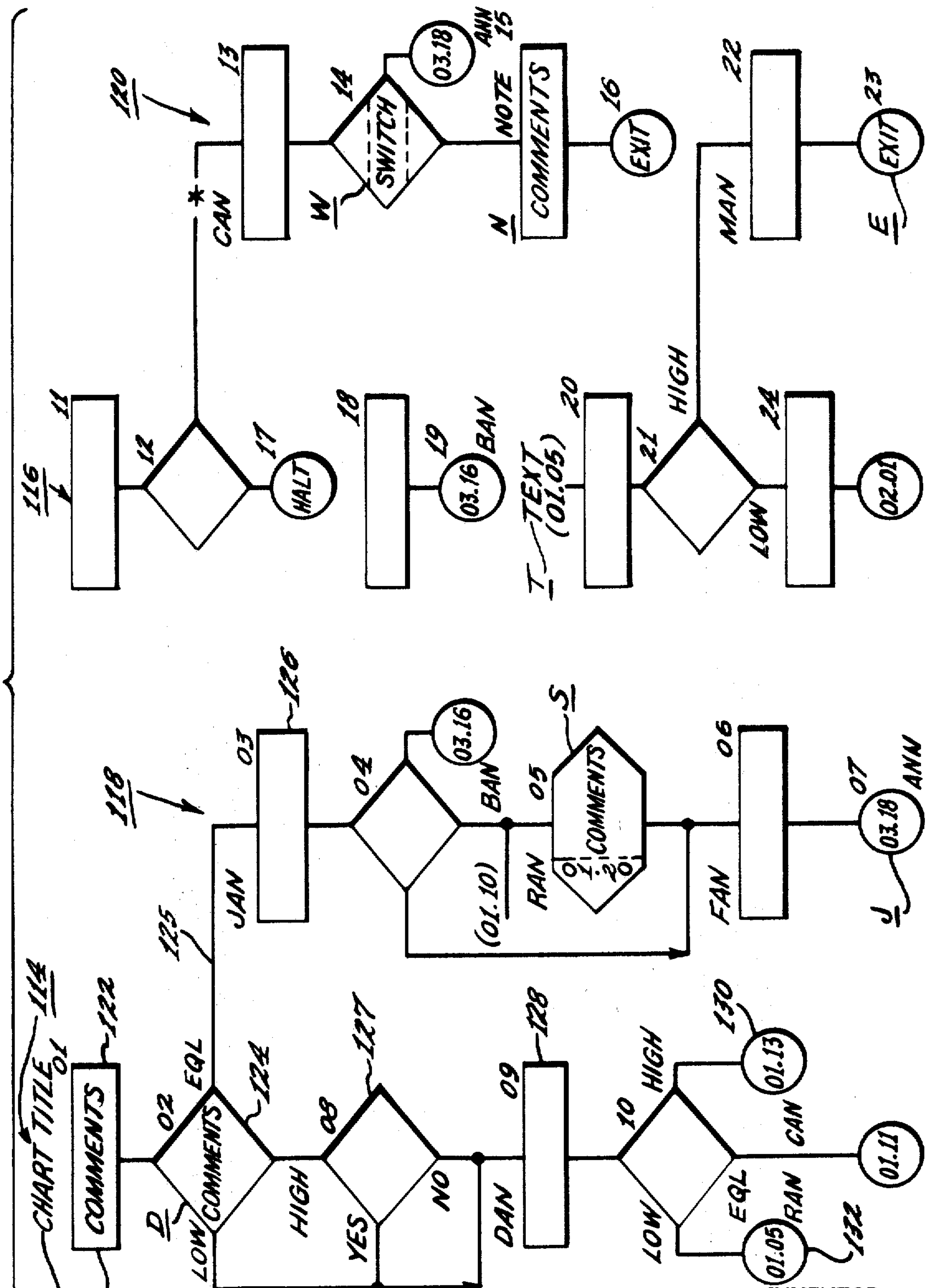


Fig. 1.

INVENTOR.
Martin A. Goetz
BY
Millman and Jacobs
ATTORNEYS.

Fig. 2.



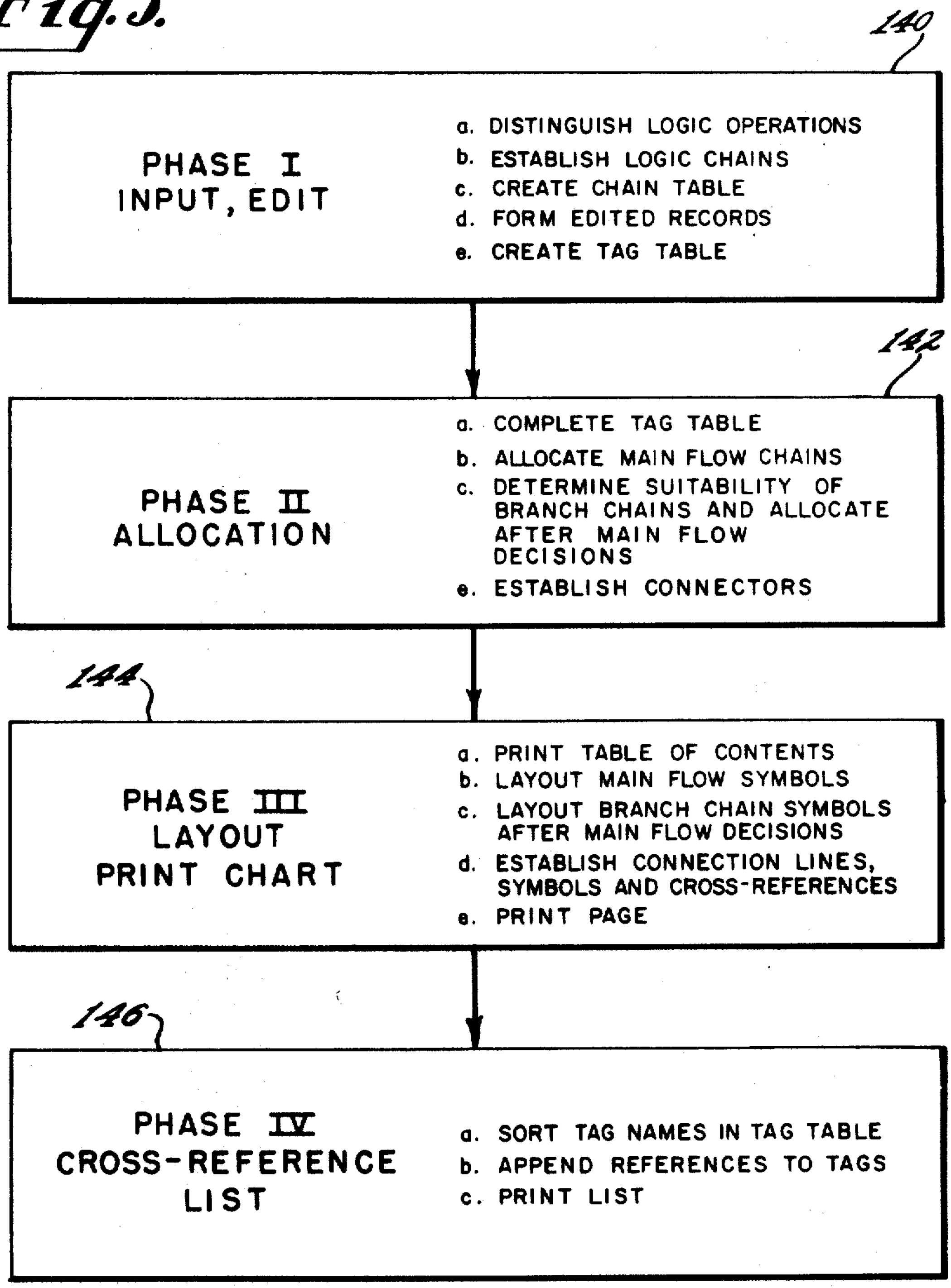
INVENTOR.
Martin A. Goetz

BY Millman and Jacobs

ATTORNEYS.

Oct. 6, 1970 M. A. GOETZ 3,533,086
AUTOMATIC SYSTEM FOR CONSTRUCTING
AND RECORDING DISPLAY CHARTS
Filed Dec. 24, 1968 38 Sheets-Sheet 3

Fig. 3.



INVENTOR.
Martin A. Goetz
BY
[signature]
ATTORNEYS.

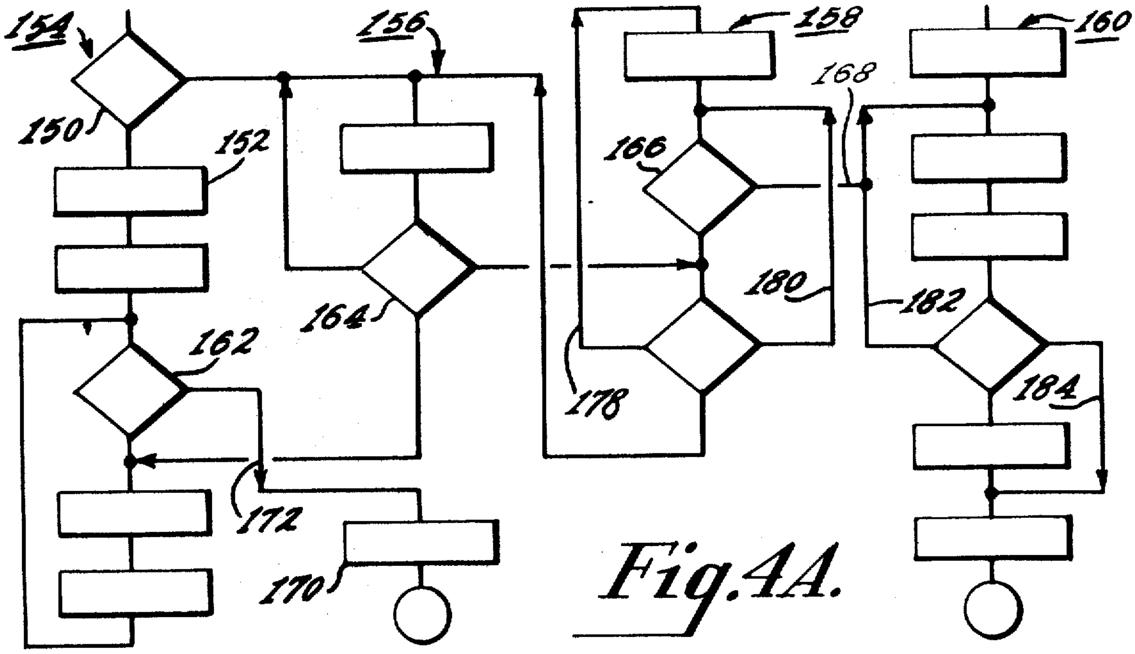


Fig.4A.

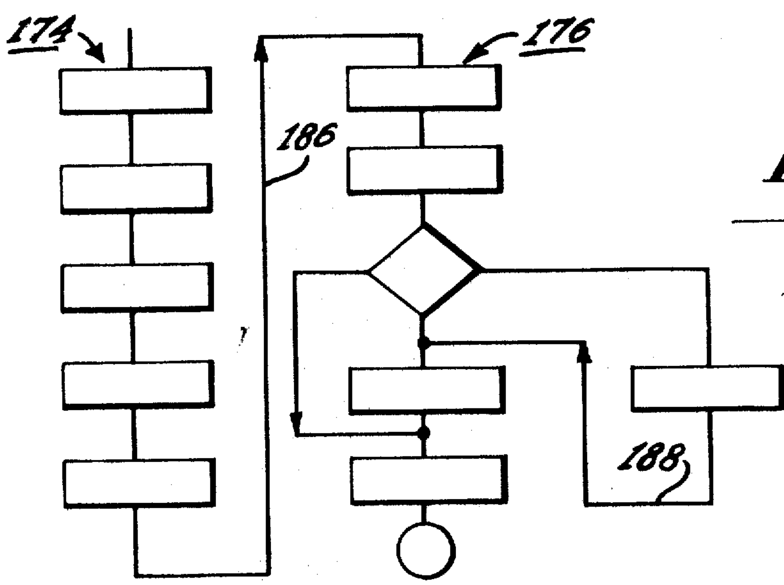


Fig.4B.

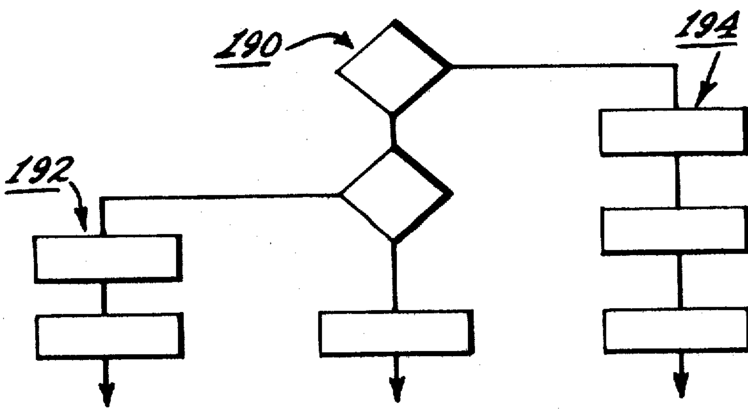


Fig.5.

INVENTOR.
Martin A. Goetz
BY
Williams and Jacobs
ATTORNEYS.

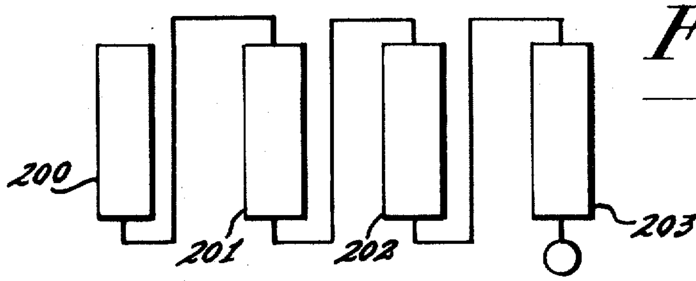


Fig. 6A.

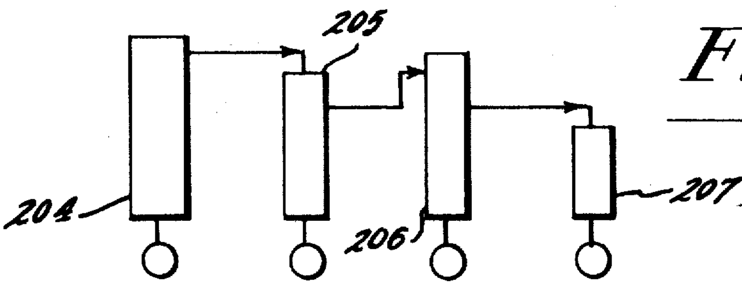


Fig. 6B.

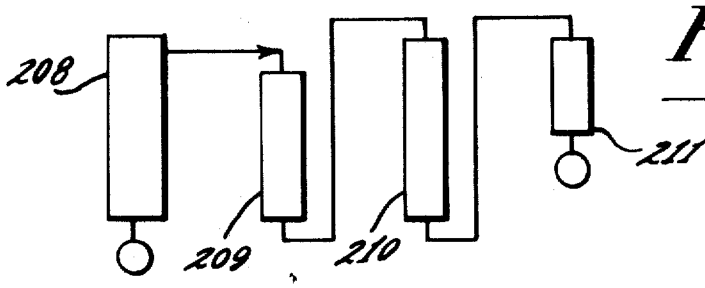


Fig. 6C.

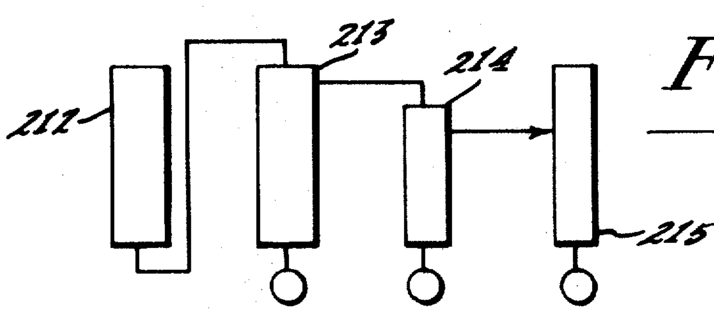


Fig. 6D.

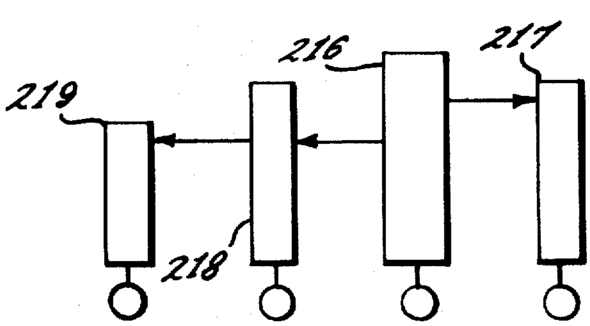


Fig. 6E.

INVENTOR.
Martin A. Goetz
BY
Williams and Jacobs
ATTORNEYS.

Fig. 7A.

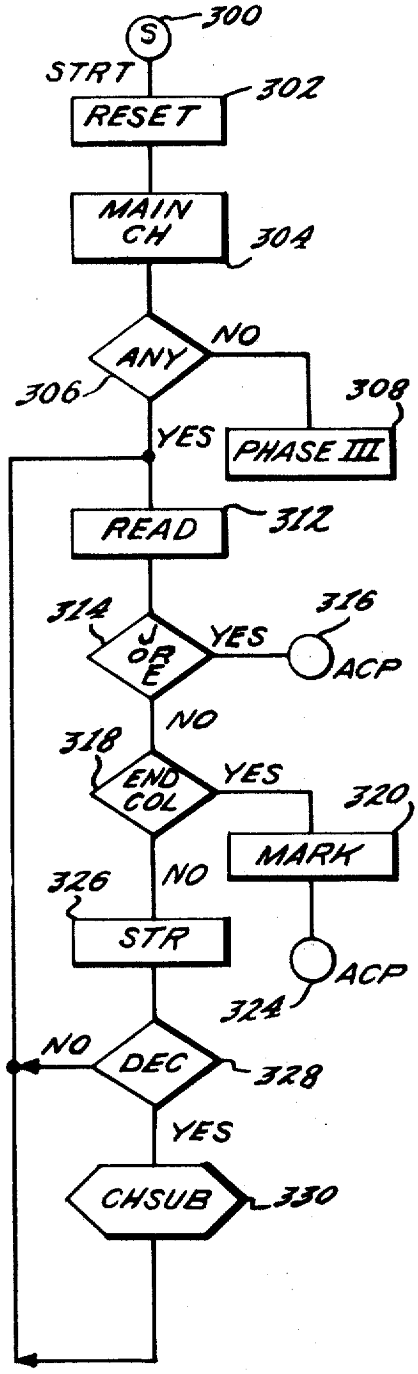
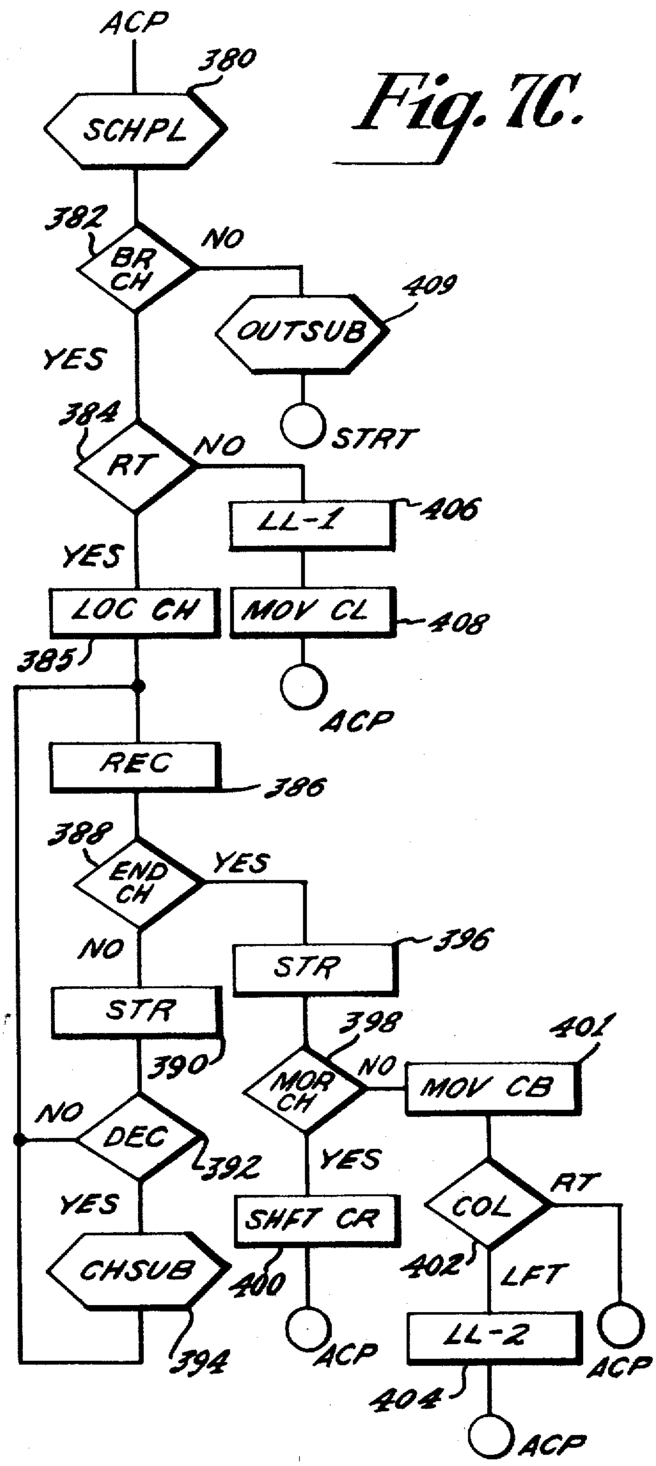
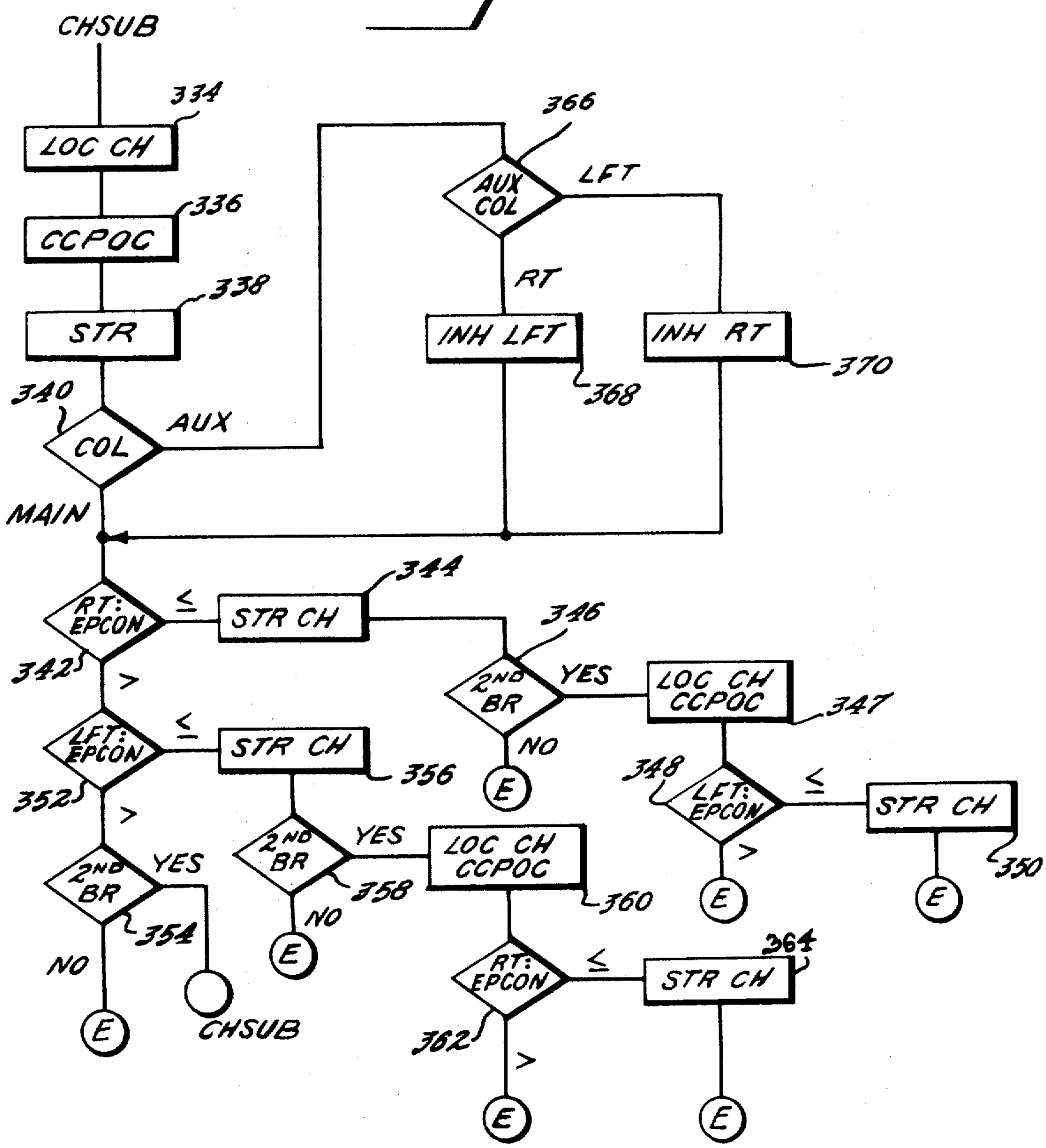


Fig. 7C.



INVENTOR.
Martin A. Goetz
BY
Millman and Jacobs
ATTORNEYS.

Fig. 7B.



INVENTOR.
Martin A. Goetz
BY
Millman and Jacobs
ATTORNEYS.

Fig. 7D.

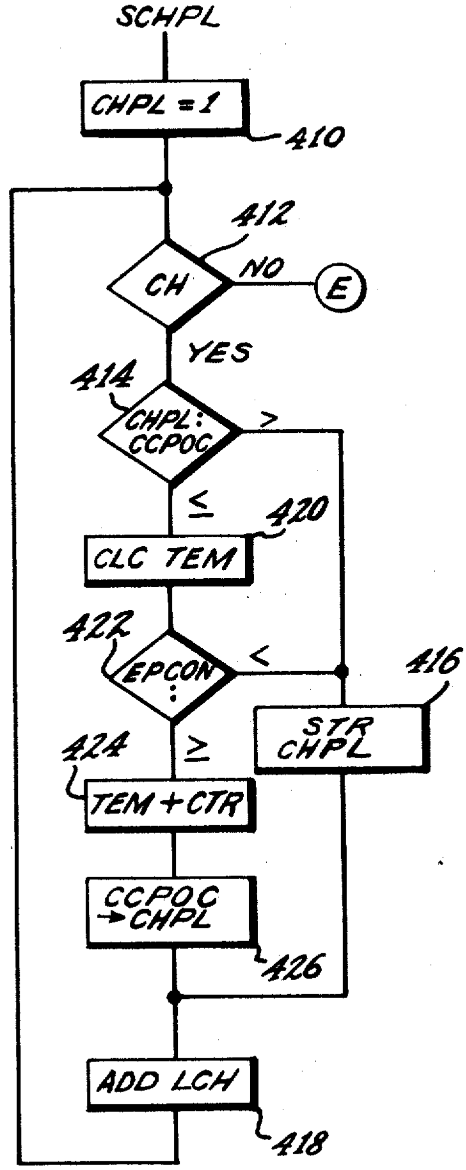
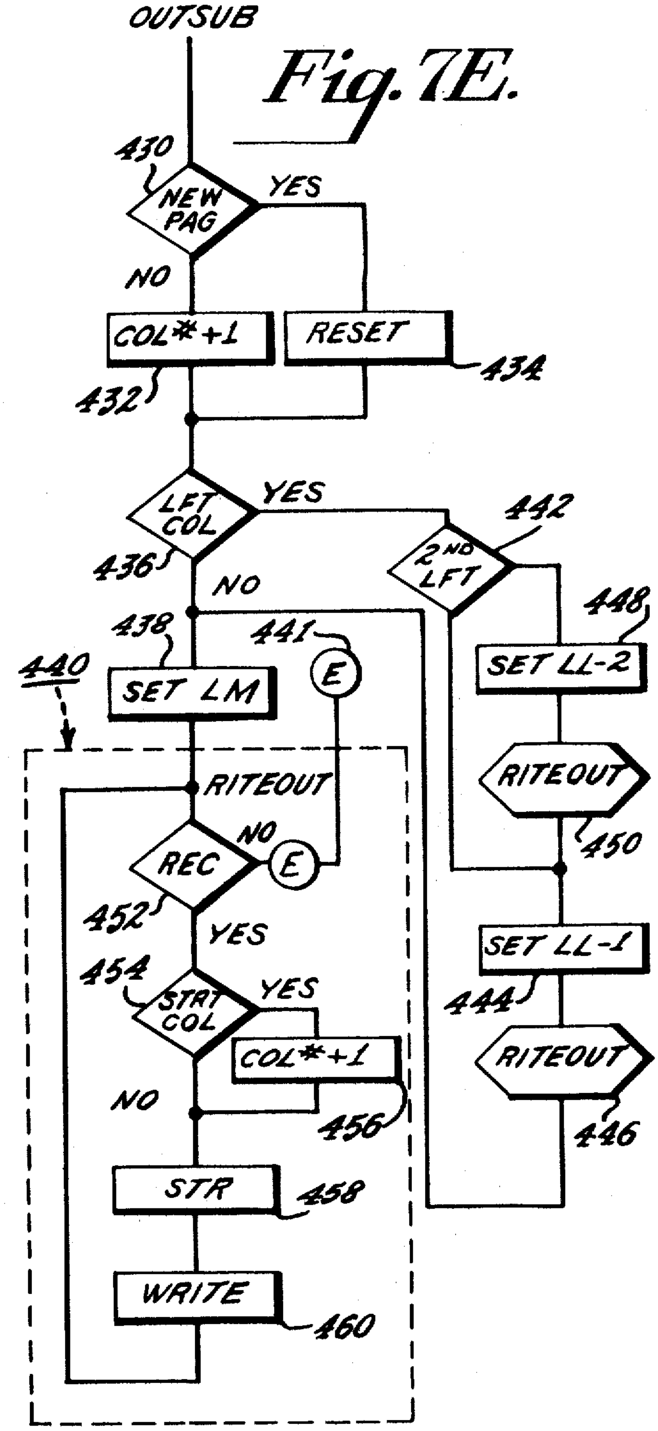


Fig. 7E.



INVENTOR.
Martin A. Goetz
BY
Millman and Jacobs
ATTORNEYS.

Fig.8/1

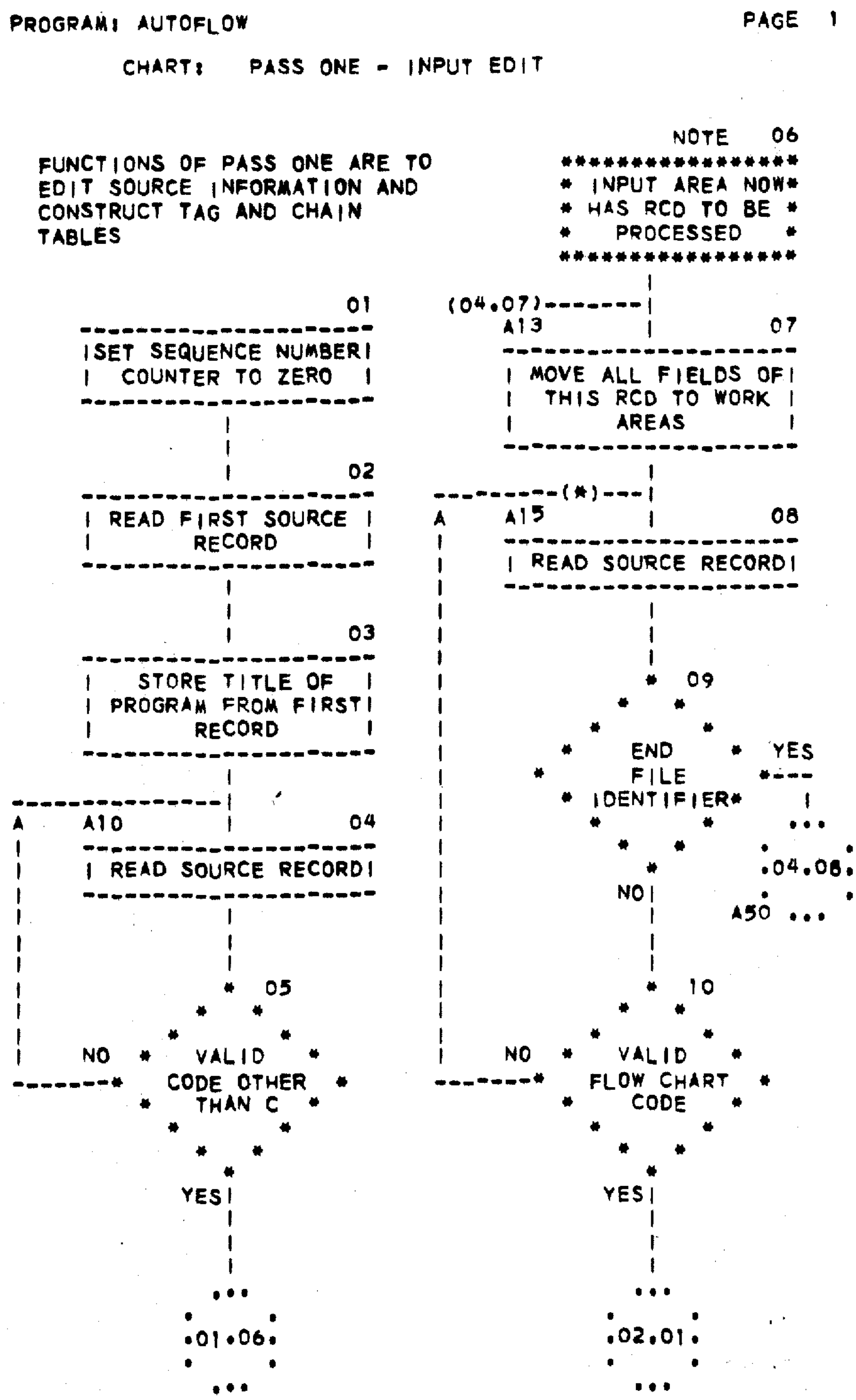
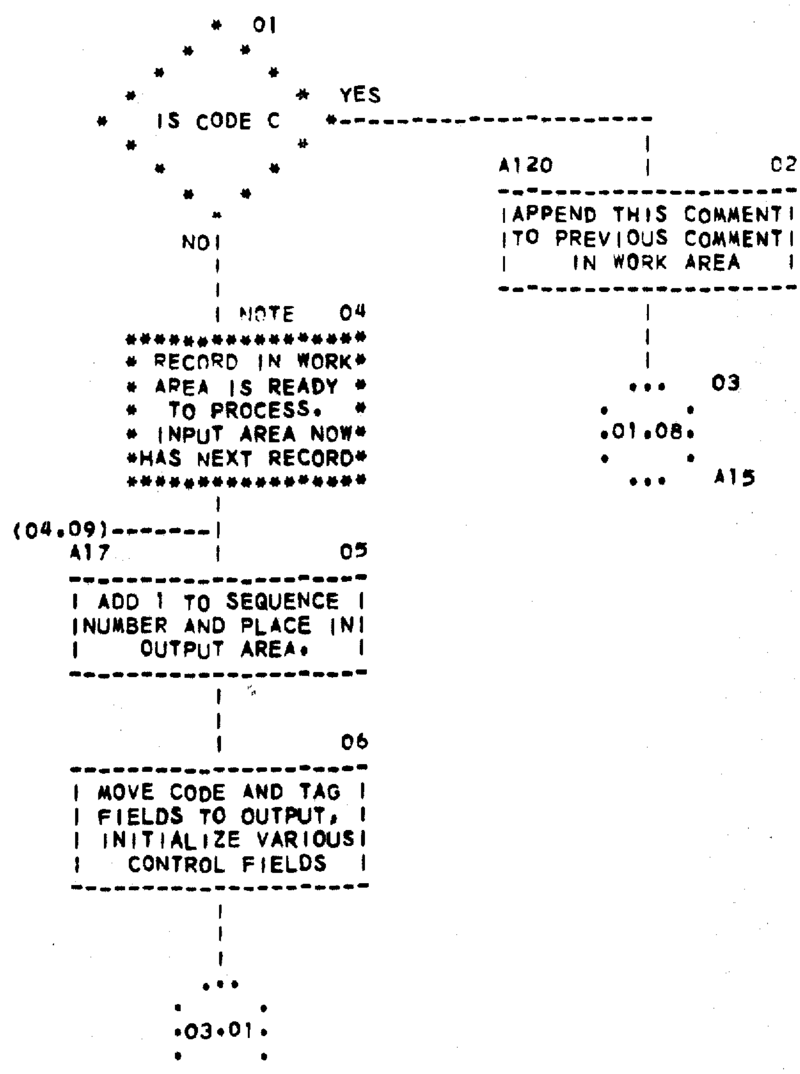

PROGRAM: AUTOFLOW
PAGE 1
CHART: PASS ONE - INPUT EDIT
FUNCTIONS OF PASS ONE ARE TO EDIT SOURCE INFORMATION AND CONSTRUCT TAG AND CHAIN TABLES
NOTE 06
INPUT AREA NOW HAS RCD TO BE PROCESSED
01
SET SEQUENCE NUMBER COUNTER TO ZERO
02
READ FIRST SOURCE RECORD
03
STORE TITLE OF PROGRAM FROM FIRST RECORD
A A10
04
READ SOURCE RECORD
05
NO
VALID CODE OTHER THAN C
YES
01.06
(04.07)
A13
07
MOVE ALL FIELDS OF THIS RCD TO WORK AREAS
(*)
A A15
08
READ SOURCE RECORD
09
END FILE IDENTIFIER
YES
04.08
A50
NO
10
NO
VALID FLOW CHART CODE
YES
02.01

Fig. 8/2

PROGRAM: AUTOFLOW PAGE 2

CHART: PASS ONE - INPUT EDIT



Oct. 6, 1970 M. A. GOETZ 3,533,086

AUTOMATIC SYSTEM FOR CONSTRUCTING AND RECORDING DISPLAY CHARTS

Filed Dec. 24, 1968 38 Sheets-Sheet 11

Fig. 8/3

PROGRAM: AUTOFLOW PAGE 3

CHART: PASS ONE - INPUT EDIT

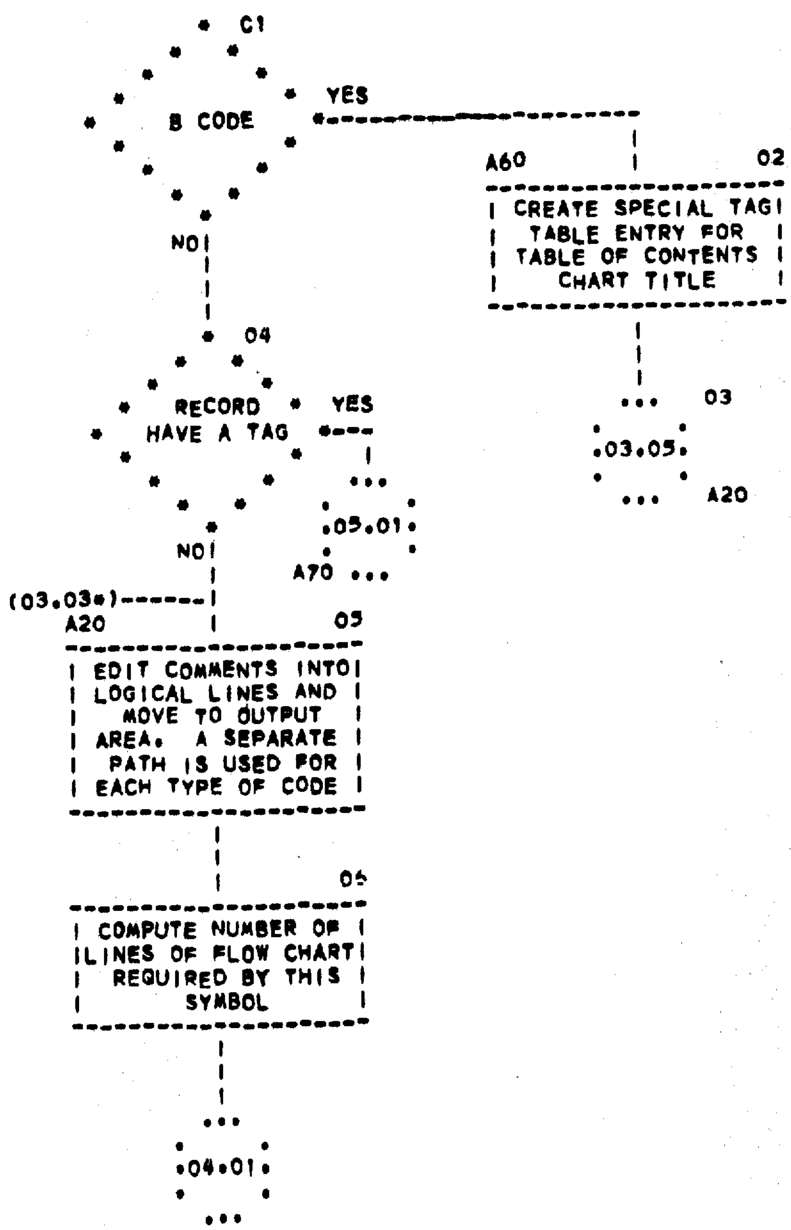


Fig.8/4

PROGRAM: AUTOFLOW PAGE 4

CHART: PASS ONE - INPUT EDIT

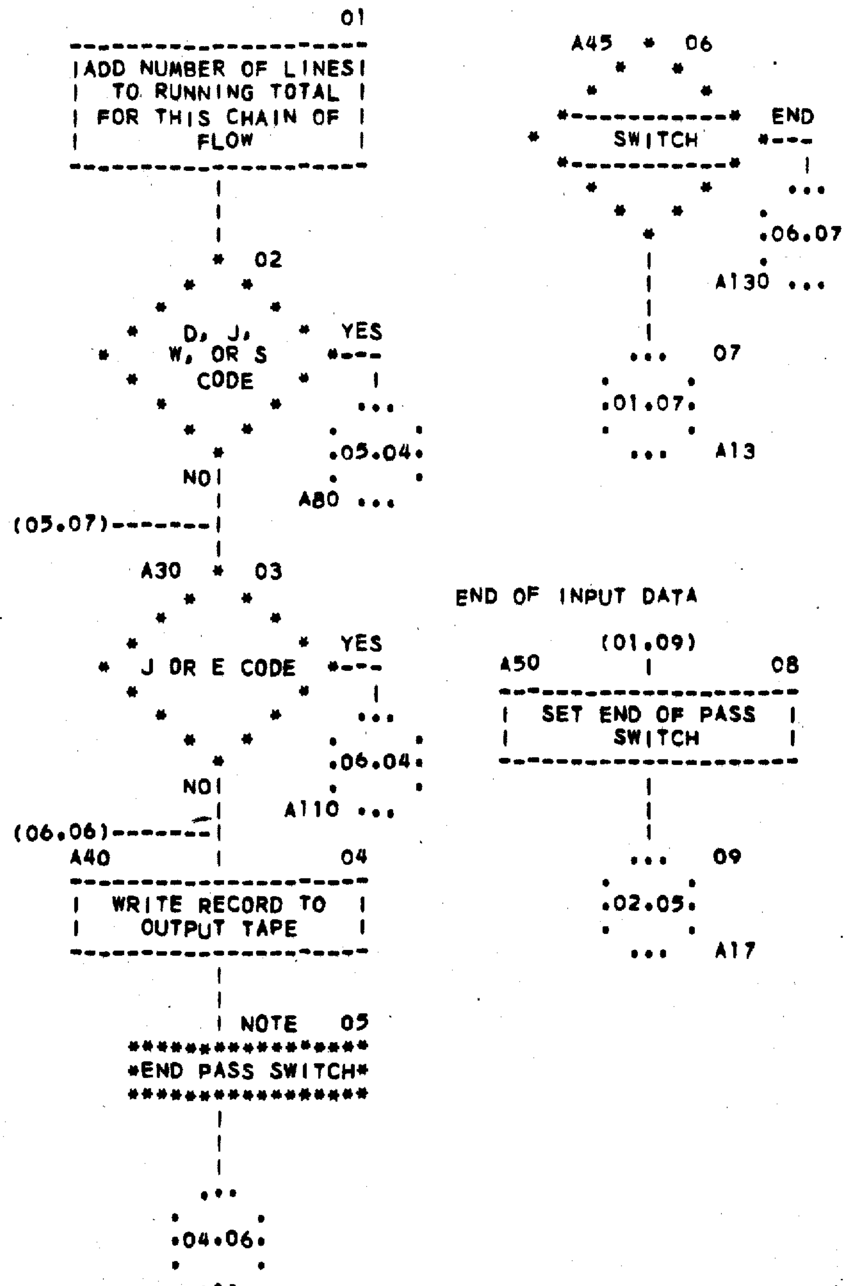


Fig. 8/5

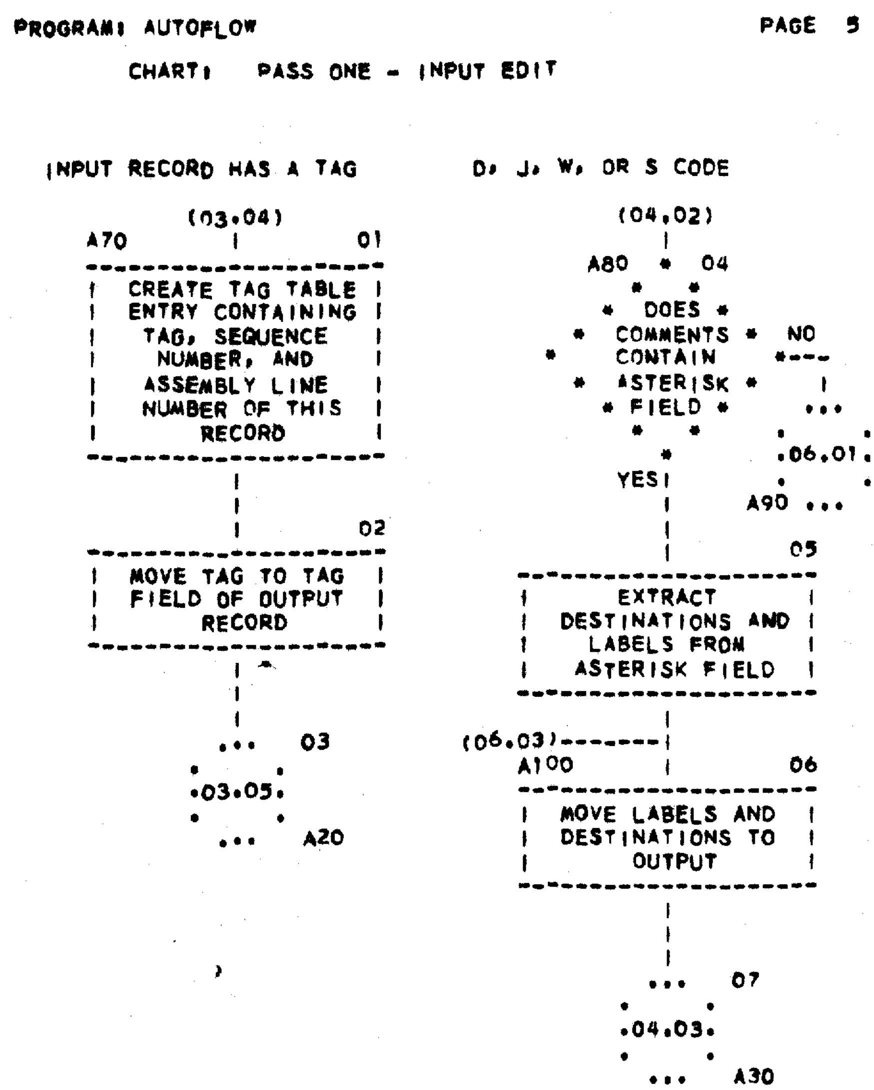

PROGRAM: AUTOFLOW
PAGE 5
CHART: PASS ONE - INPUT EDIT
INPUT RECORD HAS A TAG
D, J, W, OR S CODE
(03.04)
A70
01
CREATE TAG TABLE ENTRY CONTAINING TAG, SEQUENCE NUMBER, AND ASSEMBLY LINE NUMBER OF THIS RECORD
02
MOVE TAG TO TAG FIELD OF OUTPUT RECORD
03
.03.05.
A20
(04.02)
A80
04
DOES COMMENTS CONTAIN ASTERISK FIELD
NO
.06.01.
A90
YES
05
EXTRACT DESTINATIONS AND LABELS FROM ASTERISK FIELD
(06.03)
A100
06
MOVE LABELS AND DESTINATIONS TO OUTPUT
07
.04.03.
A30

Fig. 8/6

PROGRAM: AUTOFLOW PAGE 6

CHART: PASS ONE - INPUT EDIT

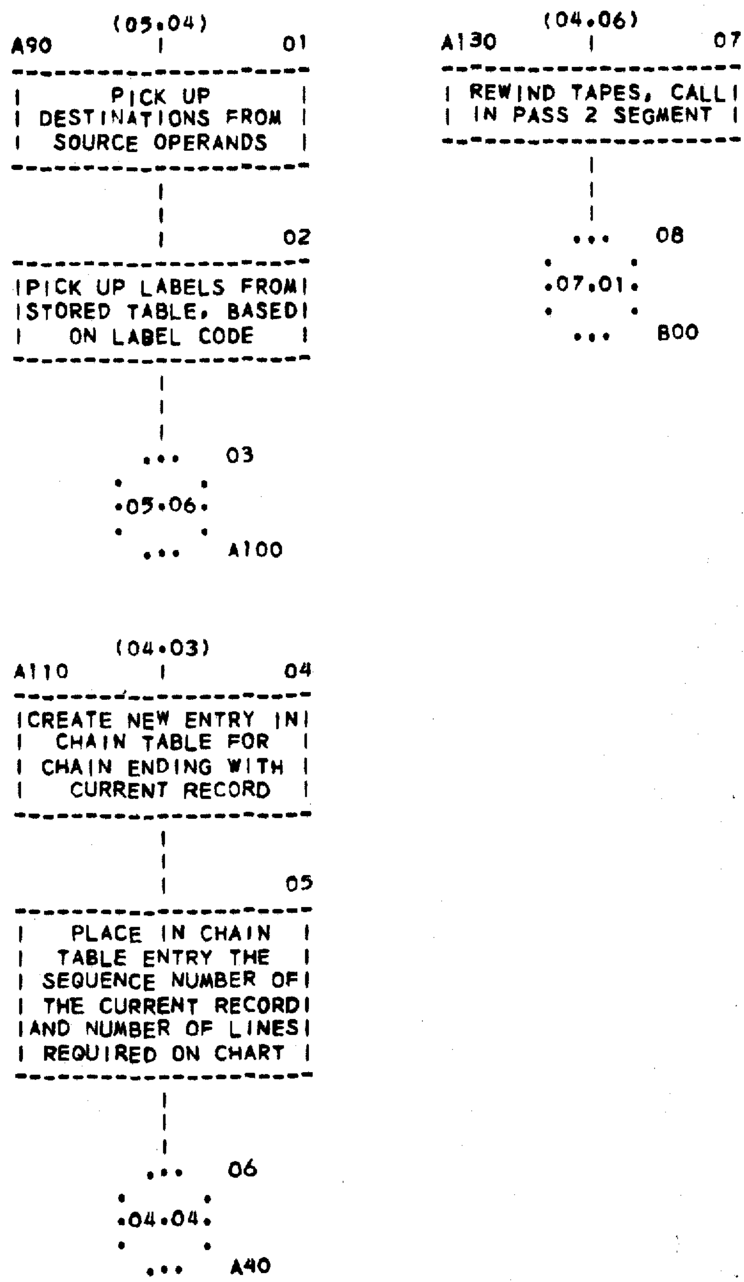


Fig. 8/7

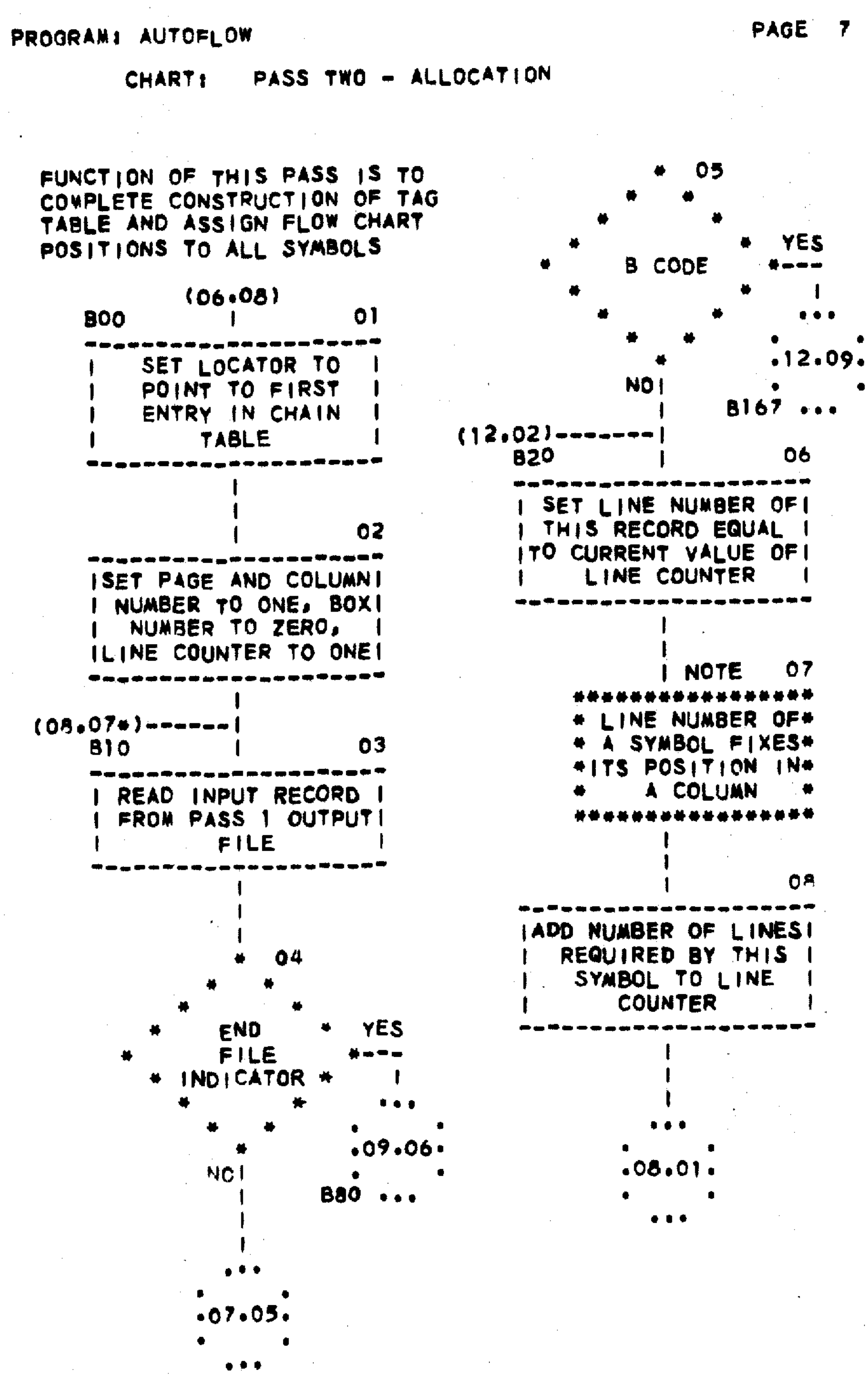

PROGRAM: AUTOFLOW
PAGE 7
CHART: PASS TWO - ALLOCATION
FUNCTION OF THIS PASS IS TO COMPLETE CONSTRUCTION OF TAG TABLE AND ASSIGN FLOW CHART POSITIONS TO ALL SYMBOLS
(06.08)
B00
01
SET LOCATOR TO POINT TO FIRST ENTRY IN CHAIN TABLE
02
SET PAGE AND COLUMN NUMBER TO ONE, BOX NUMBER TO ZERO, LINE COUNTER TO ONE
(08.07*)
B10
03
READ INPUT RECORD FROM PASS 1 OUTPUT FILE
04
END FILE INDICATOR
YES
.09.06.
B80
NO
.07.05.
05
B CODE
YES
.12.09.
B167
NO
(12.02)
B20
06
SET LINE NUMBER OF THIS RECORD EQUAL TO CURRENT VALUE OF LINE COUNTER
NOTE
07
LINE NUMBER OF A SYMBOL FIXES ITS POSITION IN A COLUMN
08
ADD NUMBER OF LINES REQUIRED BY THIS SYMBOL TO LINE COUNTER
.08.01.

Fig. 8/8

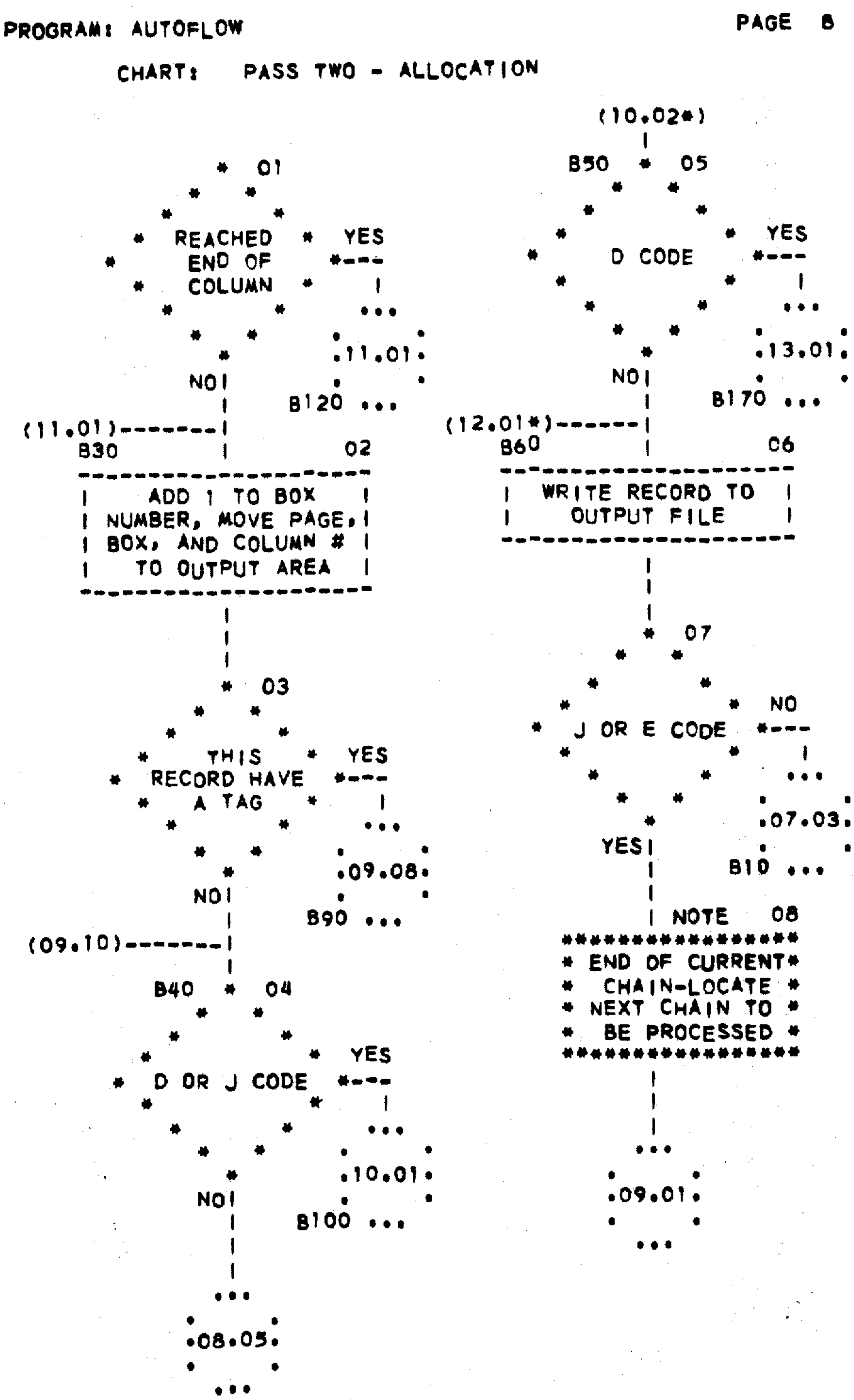

PROGRAM: AUTOFLOW
PAGE 8
CHART: PASS TWO - ALLOCATION
01
REACHED END OF COLUMN
YES
.11.01.
B120
NO
(11.01)
B30
02
ADD 1 TO BOX NUMBER, MOVE PAGE, BOX, AND COLUMN # TO OUTPUT AREA
03
THIS RECORD HAVE A TAG
YES
.09.08.
B90
NO
(09.10)
B40
04
D OR J CODE
YES
.10.01.
B100
NO
.08.05.
(10.02*)
B50
05
D CODE
YES
.13.01.
B170
NO
(12.01*)
B60
06
WRITE RECORD TO OUTPUT FILE
07
J OR E CODE
NO
.07.03.
B10
YES
NOTE
08
END OF CURRENT CHAIN-LOCATE NEXT CHAIN TO BE PROCESSED
.09.01.

Fig. 8/9

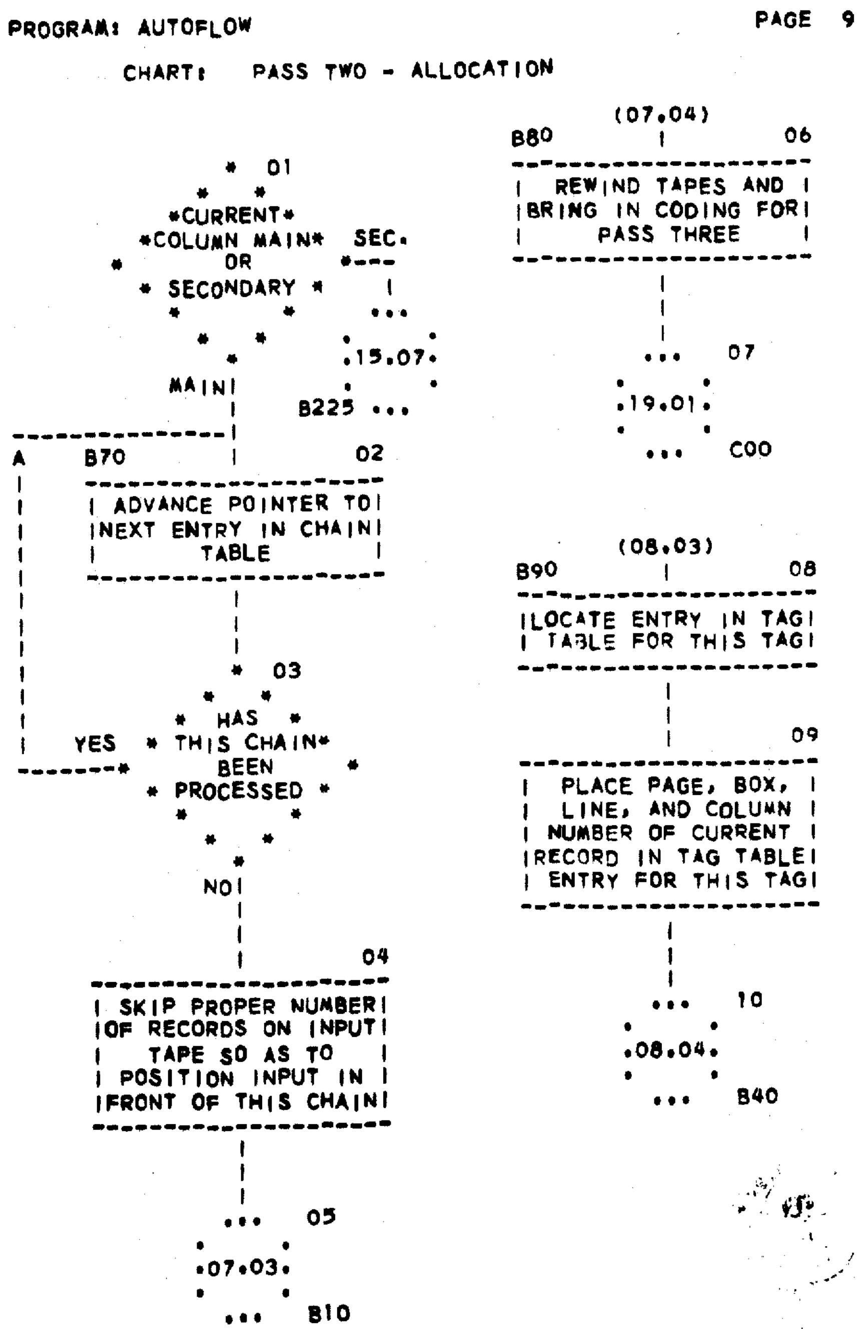

PROGRAM: AUTOFLOW
PAGE 9
CHART: PASS TWO - ALLOCATION
01
CURRENT COLUMN MAIN OR SECONDARY
SEC.
15.07
B225
MAIN
A
B70
02
ADVANCE POINTER TO NEXT ENTRY IN CHAIN TABLE
03
HAS THIS CHAIN BEEN PROCESSED
YES
NO
04
SKIP PROPER NUMBER OF RECORDS ON INPUT TAPE SO AS TO POSITION INPUT IN FRONT OF THIS CHAIN
05
07.03
B10
(07.04)
B80
06
REWIND TAPES AND BRING IN CODING FOR PASS THREE
07
19.01
C00
(08.03)
B90
08
LOCATE ENTRY IN TAG TABLE FOR THIS TAG
09
PLACE PAGE, BOX, LINE, AND COLUMN NUMBER OF CURRENT RECORD IN TAG TABLE ENTRY FOR THIS TAG
10
08.04
B40

Fig. 8/10

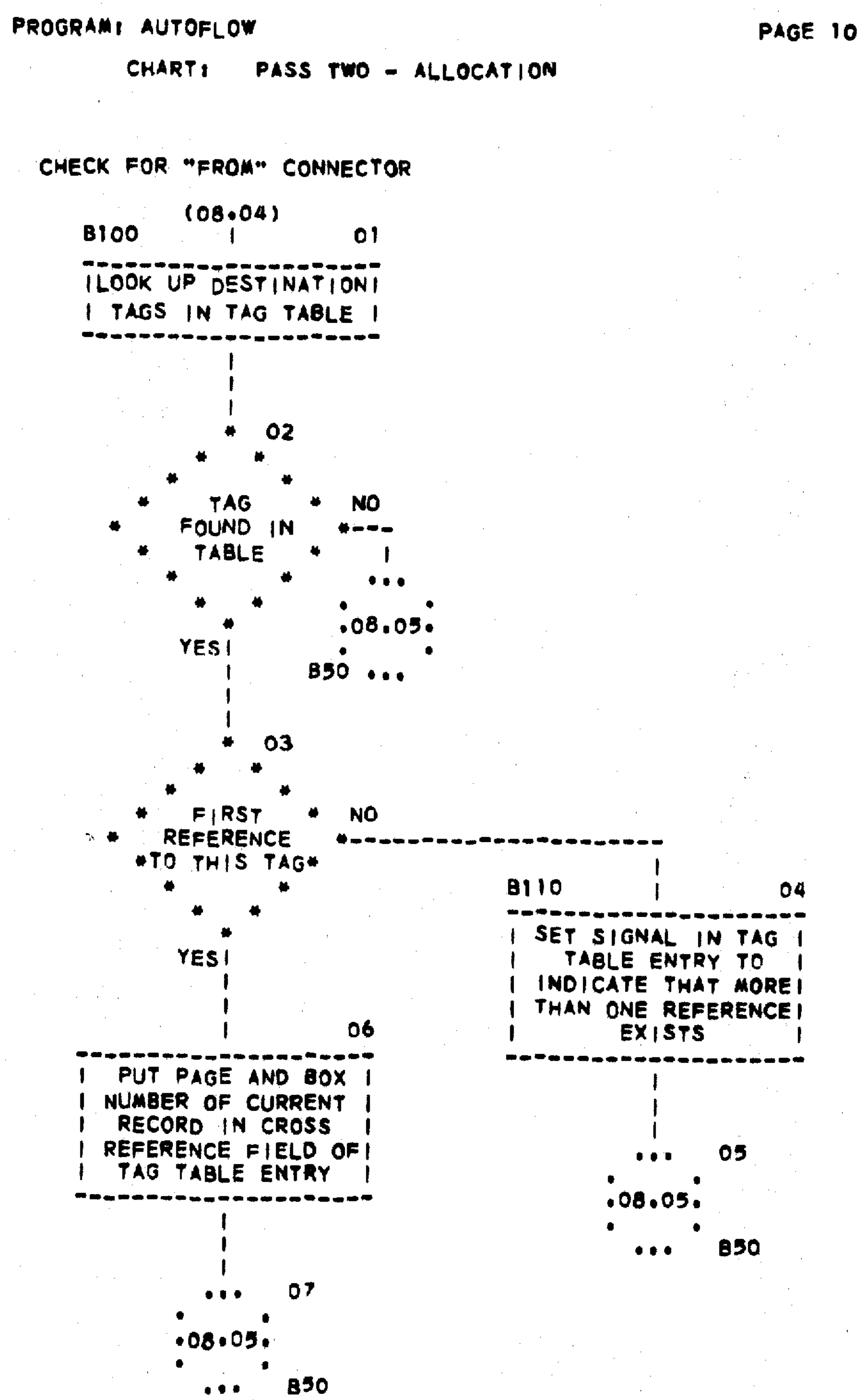

PROGRAM: AUTOFLOW
PAGE 10
CHART: PASS TWO - ALLOCATION
CHECK FOR "FROM" CONNECTOR
(08.04)
B100
01
LOOK UP DESTINATION TAGS IN TAG TABLE
02
TAG FOUND IN TABLE
NO
.08.05.
B50
YES
03
FIRST REFERENCE TO THIS TAG
NO
B110
04
SET SIGNAL IN TAG TABLE ENTRY TO INDICATE THAT MORE THAN ONE REFERENCE EXISTS
05
.08.05.
B50
YES
06
PUT PAGE AND BOX NUMBER OF CURRENT RECORD IN CROSS REFERENCE FIELD OF TAG TABLE ENTRY
07
.08.05.
B50

Fig. 8/11

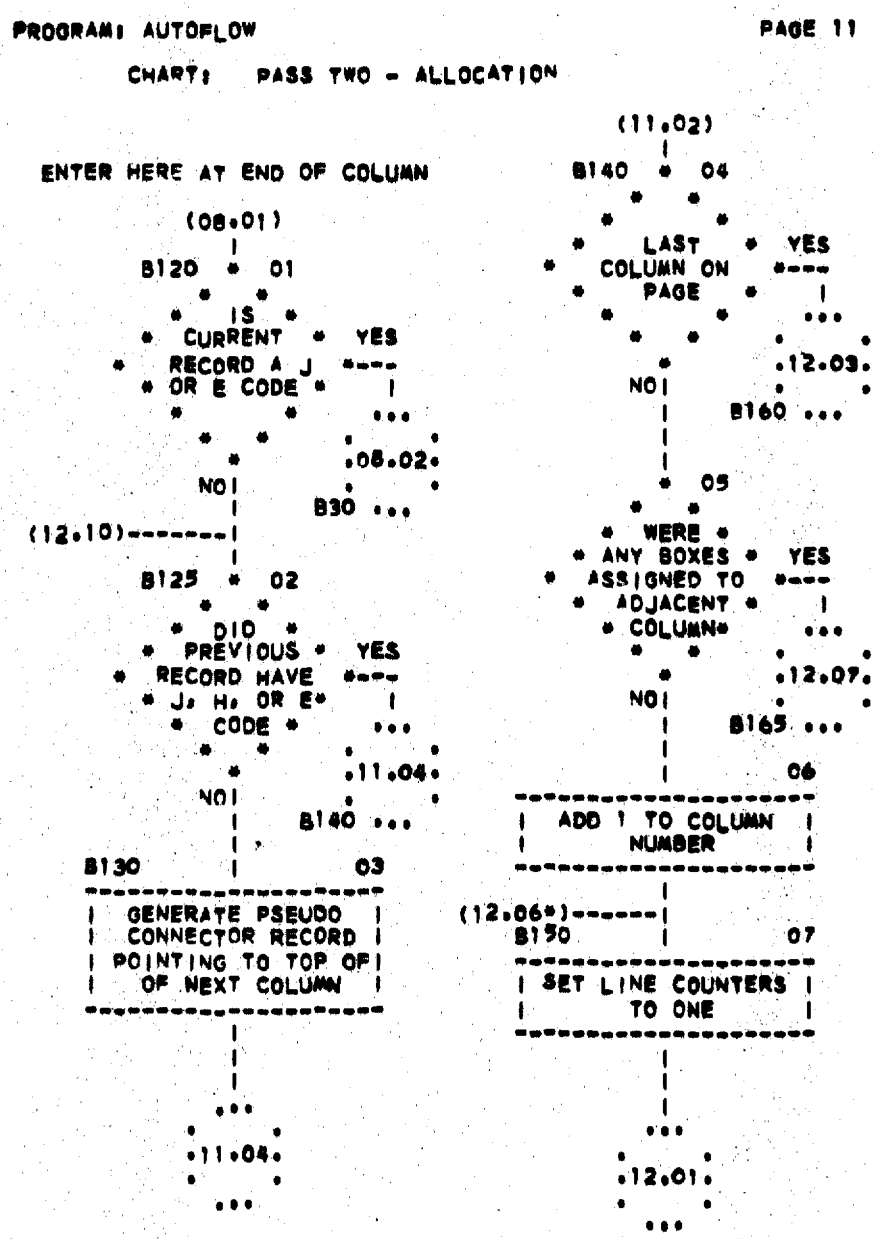

PROGRAM: AUTOFLOW
PAGE 11
CHART: PASS TWO - ALLOCATION
ENTER HERE AT END OF COLUMN
(08.01)
B120 01
IS CURRENT RECORD A J OR E CODE
YES
.08.02.
B30
NO
(12.10)
B125 02
DID PREVIOUS RECORD HAVE J, H, OR E CODE
YES
.11.04.
B140
NO
B130 03
GENERATE PSEUDO CONNECTOR RECORD POINTING TO TOP OF OF NEXT COLUMN
.11.04.
(11.02)
B140 04
LAST COLUMN ON PAGE
YES
.12.03.
B160
NO
05
WERE ANY BOXES ASSIGNED TO ADJACENT COLUMN
YES
.12.07.
B165
NO
06
ADD 1 TO COLUMN NUMBER
(12.06*)
B150 07
SET LINE COUNTERS TO ONE
.12.01.

Fig. 8/12

PROGRAM: AUTOFLOW PAGE 12

CHART: PASS TWO - ALLOCATION

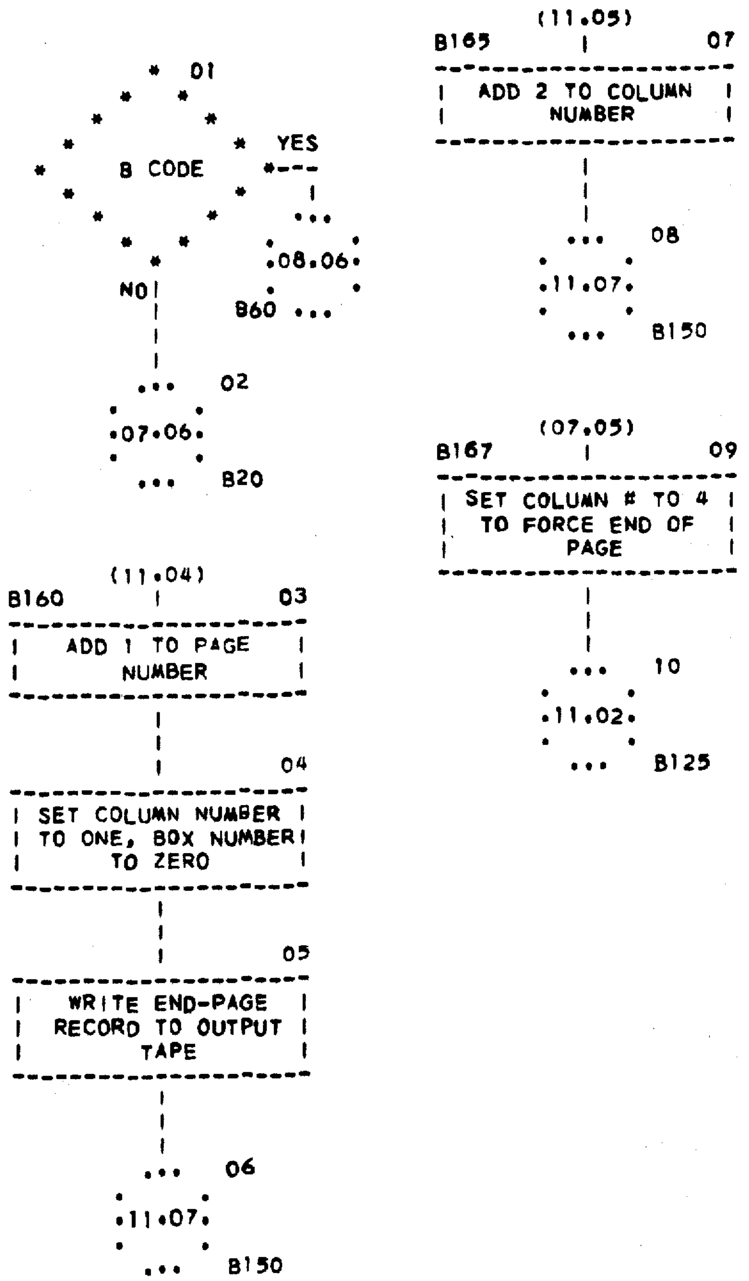


Fig. 8/13

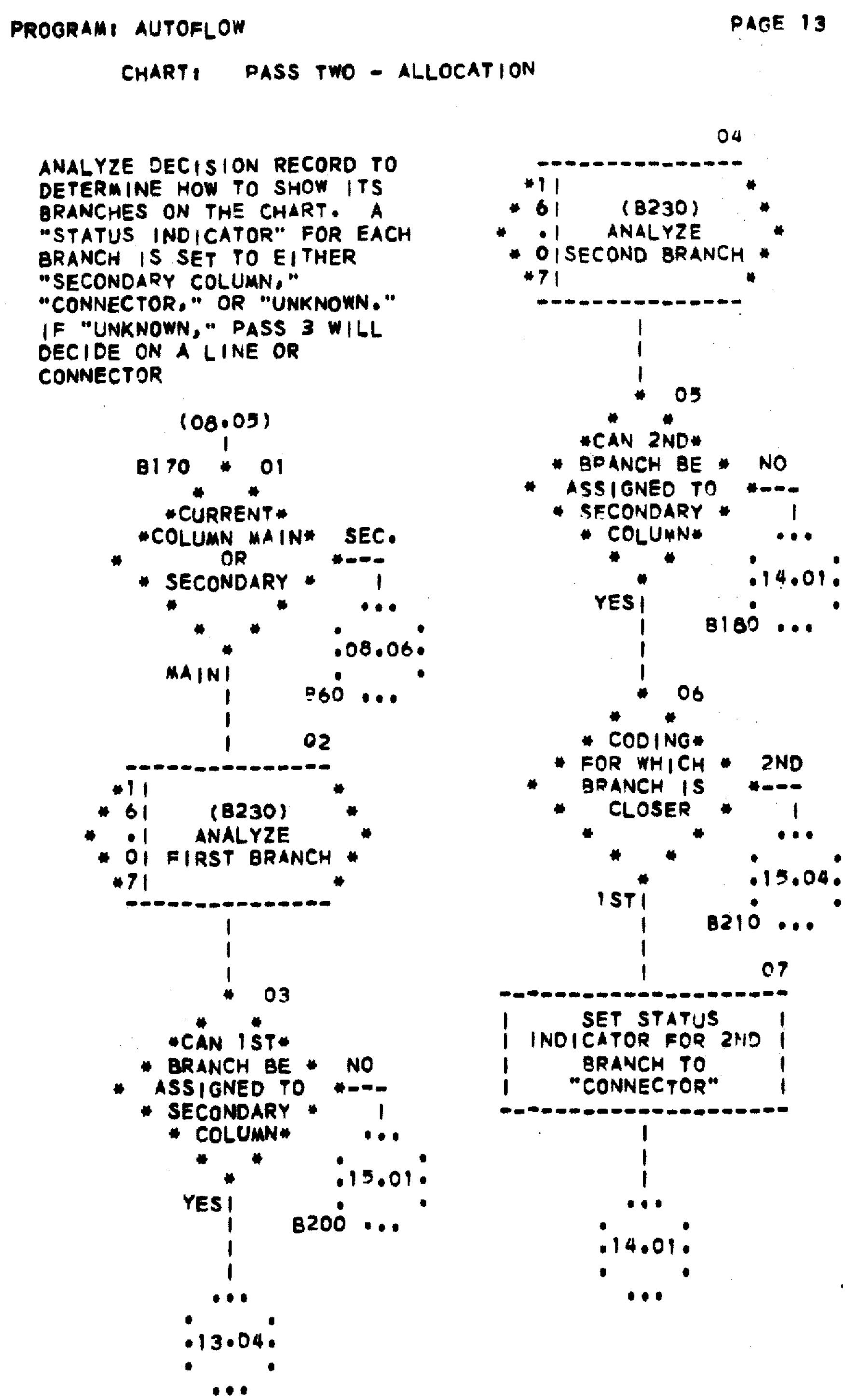

PROGRAM: AUTOFLOW
PAGE 13
CHART: PASS TWO - ALLOCATION
ANALYZE DECISION RECORD TO DETERMINE HOW TO SHOW ITS BRANCHES ON THE CHART. A "STATUS INDICATOR" FOR EACH BRANCH IS SET TO EITHER "SECONDARY COLUMN," "CONNECTOR," OR "UNKNOWN." IF "UNKNOWN," PASS 3 WILL DECIDE ON A LINE OR CONNECTOR
(08.05)
B170 01
CURRENT COLUMN MAIN OR SECONDARY
SEC.
08.06
B60
MAIN
02
16.07
(B230) ANALYZE FIRST BRANCH
03
CAN 1ST BRANCH BE ASSIGNED TO SECONDARY COLUMN
NO
15.01
B200
YES
13.04
04
16.07
(B230) ANALYZE SECOND BRANCH
05
CAN 2ND BRANCH BE ASSIGNED TO SECONDARY COLUMN
NO
14.01
B180
YES
06
CODING FOR WHICH BRANCH IS CLOSER
2ND
15.04
B210
1ST
07
SET STATUS INDICATOR FOR 2ND BRANCH TO "CONNECTOR"
14.01

Fig. 8/14

```
PROGRAM: AUTOFLOW                                        PAGE 14

      CHART:   PASS TWO - ALLOCATION

          (13.05)
  B180       |         01                                 06
  ---------------------               ---------------------
  |    SET STATUS     |               |  SET INDICATOR TO |
  | INDICATOR FOR 1ST |               |   SHOW THAT A     |
  |     BRANCH TO     |               |SECONDARY COLUMN IS|
  | "SECONDARY COL"   |               |   BEING FILLED    |
  ---------------------               ---------------------
             |                                  |
(15.06)------|                                  |
  B190       |         02                       |         07
  ---------------------               ---------------------
  |  WRITE DECISION   |               |STORE CURRENT VALUE|
  | RECORD TO OUTPUT  |               |  OF LINE COUNTER  |
  |       FILE        |               ---------------------
  ---------------------                         |
             |                                  |
             |                                  |         08
             |         03             ---------------------
  ---------------------               |REDUCE LINE COUNTER|
  |POSITION INPUT TAPE|               | BY 6 TO GET LINE #|
  | IN FRONT OF FIRST |               |  OF FIRST SYMBOL  |
  |  RECORD OF CHAIN  |               |    GOING INTO     |
  |   WHICH IS TO BE  |               |  SECONDARY COLUMN |
  |  ASSIGNED TO THE  |               ---------------------
  |  SECONDARY COLUMN |                         |
  ---------------------                         |
             |                                  | NOTE    09
             |                          *****************
             |         04               *   GO PROCESS  *
  ---------------------                 *   CODING FOR  *
  |SET SIGNAL IN CHAIN|                 *   SECONDARY   *
  | TABLE ENTRY THAT  |                 * COLUMN. FIRST *
  |THIS CHAIN HAS BEEN|                 *  J OR E CODE  *
  |     PROCESSED     |                 * ENDS CHAIN AND*
  ---------------------                 *PROGRAM RETURNS*
             |                          *  TO TAG B225  *
             |                          *****************
             |         05                       |
  ---------------------                         |
  | ADD ONE TO COLUMN |                         |
  |      NUMBER       |                        ...    10
  ---------------------                       .   .
             |                                .07.03.
             |                                .   .
             |                                 ...    B10
            ...
           .   .
           .14.06.
           .   .
            ...
```

Fig. 8/15

PROGRAM: AUTOFLOW PAGE 15

CHART: PASS TWO - ALLOCATION

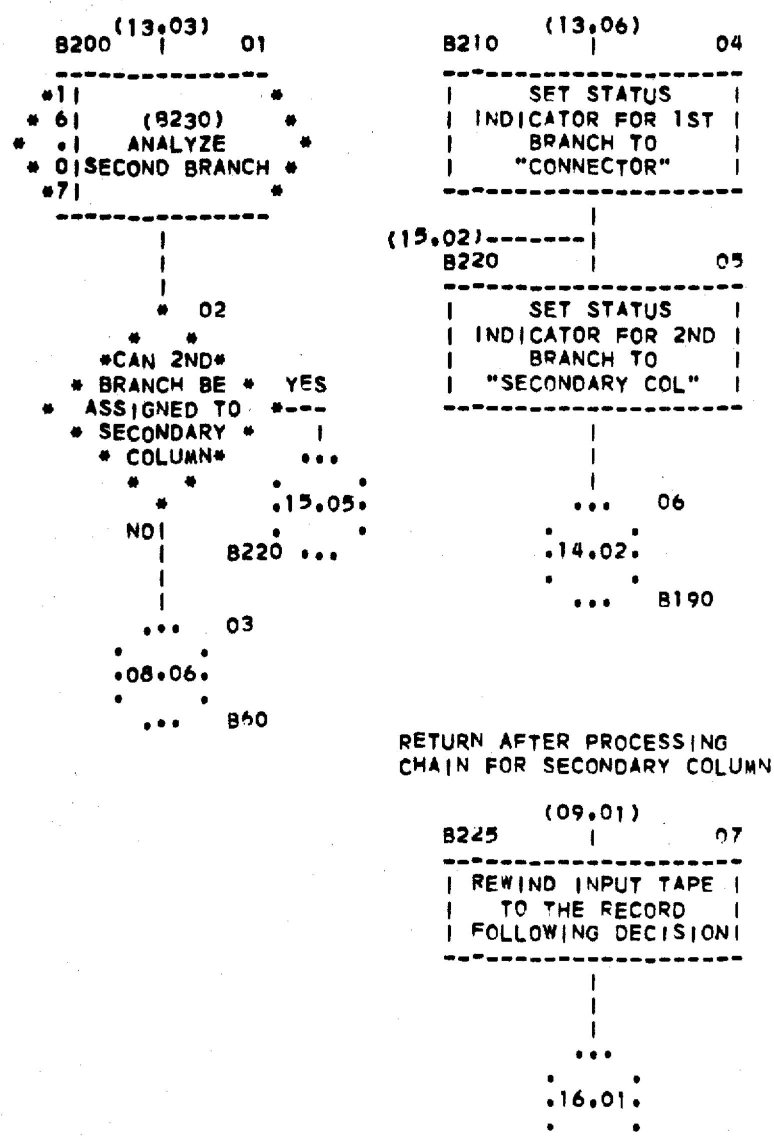


Fig. 8/16

PROGRAM: AUTOFLOW PAGE 16

CHART: PASS TWO - ALLOCATION

```
       NOTE    01
 ****************
 *  RETURN FROM  *
 *   PROCESSING  *
 *    CHAIN IN   *
 *   SECONDARY   *
 * COLUMN RECORD,*
 *    SO AS TO   *
 *    CONTINUE   *
 *   PROCESSING  *
 *CODING FOR MAIN*
 *     COLUMN    *
 ****************
         |
         |
         |          02
--------------------
|   RESTORE LINE   |
|  COUNTER TO ITS  |
|  VALUE PRIOR TO  |
|    PROCESSING    |
| SECONDARY CODING |
--------------------
         |
         |
         |          03
--------------------
| RESET INDICATOR TO|
|  SHOW THAT A MAIN |
|  COLUMN IS AGAIN  |
|   BEING FILLED    |
--------------------
         |
         |
         |          04
--------------------
|  REDUCE COLUMN   |
|  NUMBER BY ONE   |
--------------------
         |
         |
         | NOTE    05
 ****************
 * RETURN TO MAIN*
 *     COLUMN    *
 *   PROCESSING  *
 ****************
         |
         |
         |
        ...     06
      .     .
      .07.03.
      .     .
        ...     B10
```

SUBROUTINE TO ANALYZE A CHAIN OF CODE BRANCHING FROM A DECISION SYMBOL. PURPOSE OF THE ANALYSIS IS TO DETERMINE IF IT IS POSSIBLE TO ASSIGN THE CHAIN TO A SECONDARY COLUMN.

```
            (13.02*)
 B230          |           07
--------------------
|   SET EXIT FROM   |
|    SUBROUTINE     |
--------------------
               |
               |
               |           08
--------------------
|LOOK UP DESTINATION|
|  TAG IN TAG TABLE |
--------------------
               |
               |
               |
               *     09
             *   *
           *       *
         *    TAG    *    NO
       *   FOUND IN    *---
         *   TABLE   *    |
           *       *     ...
             *   *     .     .
               *       .17.02.
           YES |       .     .
               |    B250 ...
               |
               |
              ...
            .     .
            .17.01.
            .     .
              ...
```

Fig. 8/17

PROGRAM: AUTOFLOW PAGE 17

CHART: PASS TWO - ALLOCATION

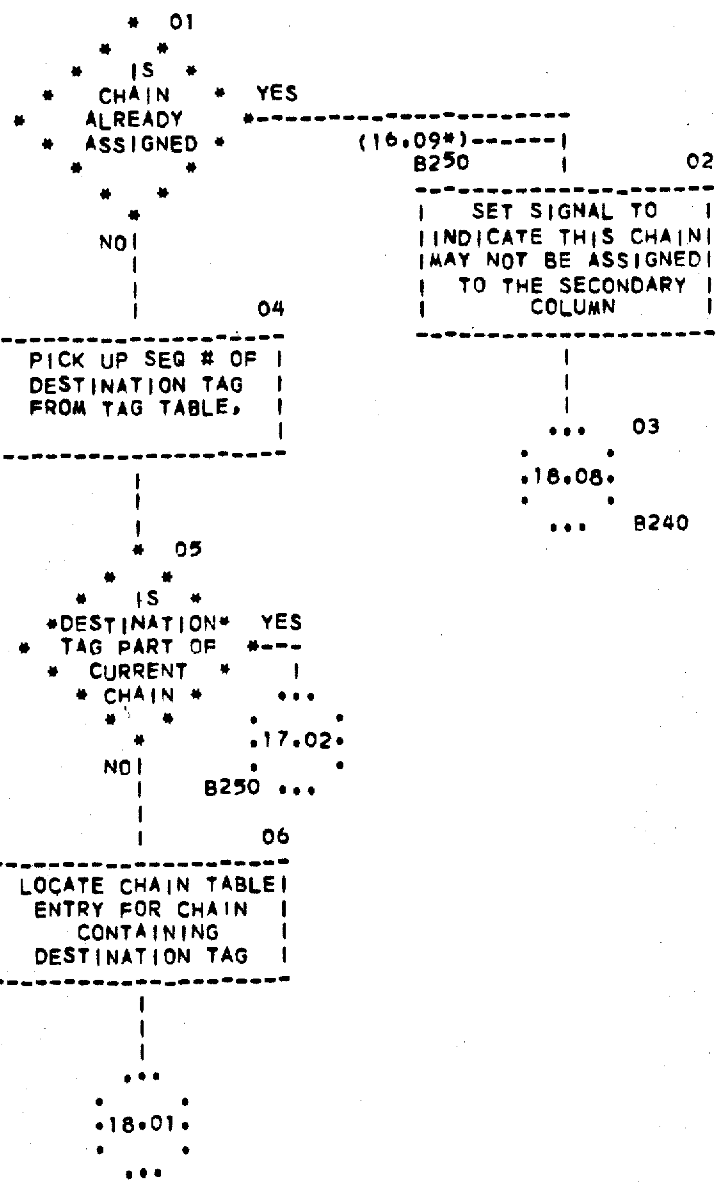


Fig. 8/18

PROGRAM: AUTOFLOW PAGE 18

CHART: PASS TWO - ALLOCATION

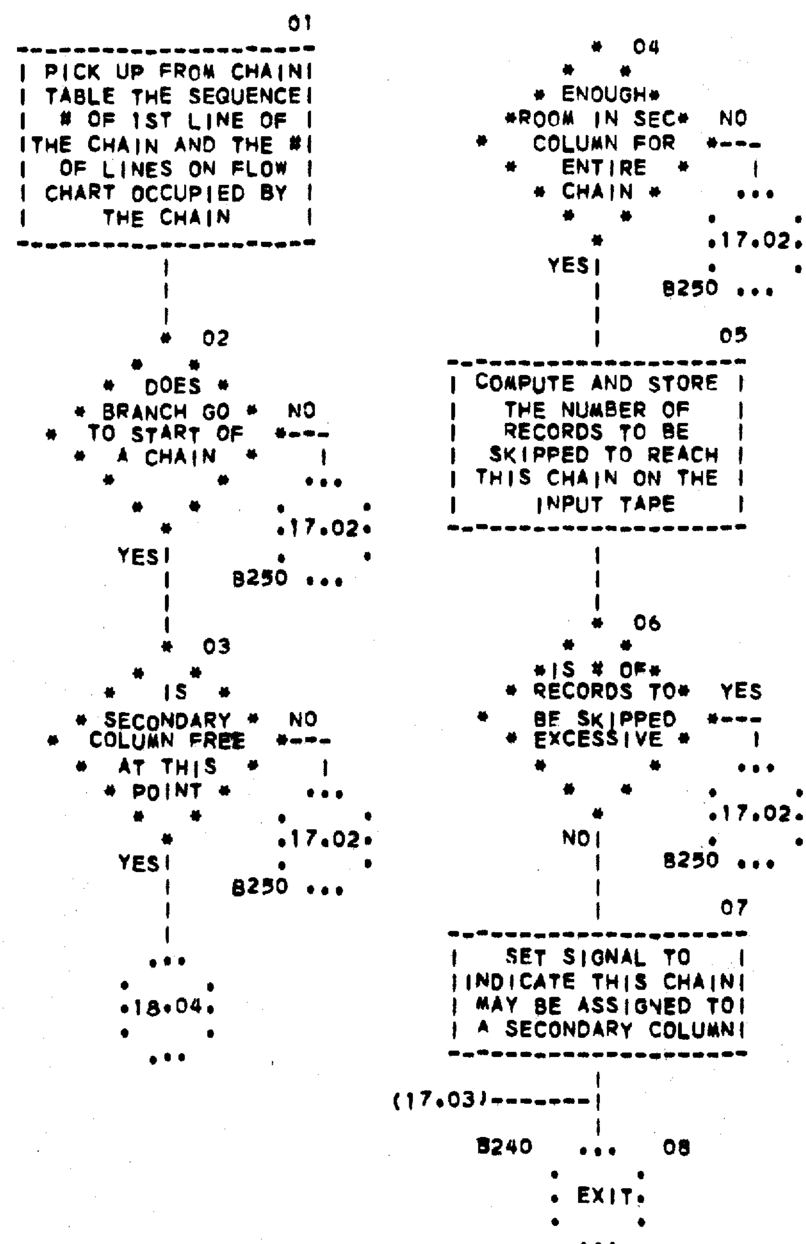


Fig. 8/19

PROGRAM: AUTOFLOW PAGE 19

CHART: PASS THREE - LAYOUT

THE FUNCTION OF THIS PASS IS TO PRODUCE THE ACTUAL FLOW CHARTS, USING INFORMATION DEVELOPED BY THE EARLIER PASSES. A PAGE OF CHART IS DEVELOPED AT A TIME, WITH THE ENTIRE PAGE BEING STORED IN MEMORY.

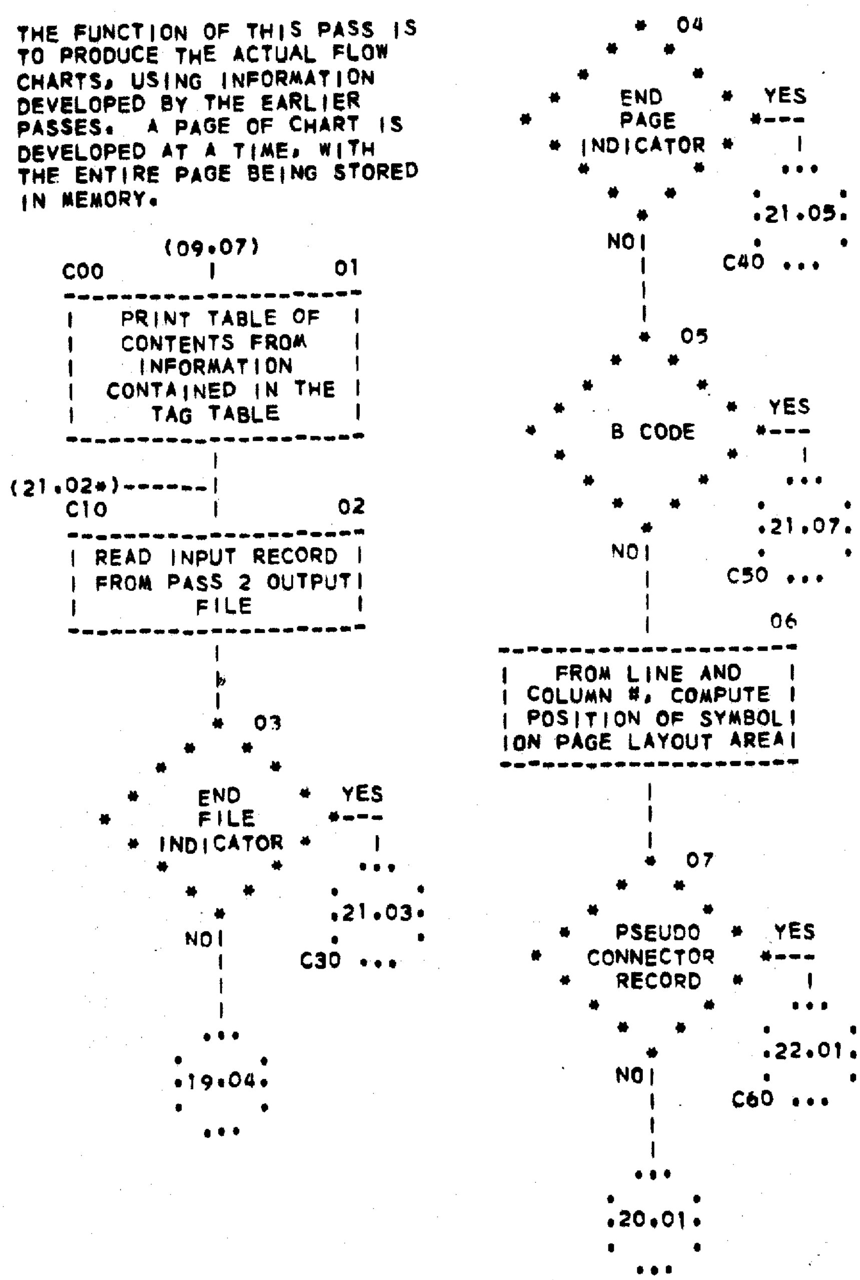


Fig. 8/20

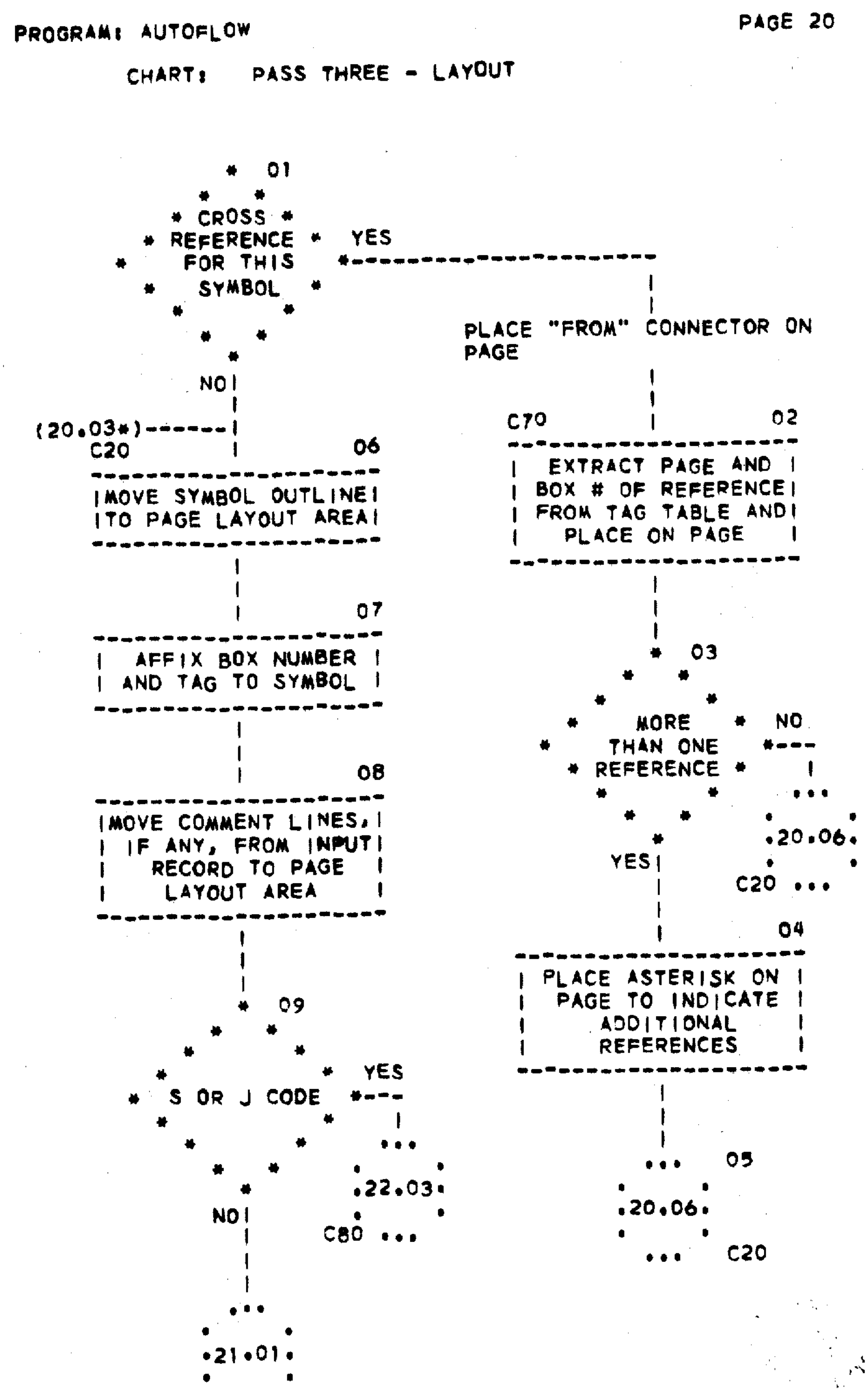

PROGRAM: AUTOFLOW
PAGE 20
CHART: PASS THREE - LAYOUT
01
CROSS REFERENCE FOR THIS SYMBOL
YES
NO
PLACE "FROM" CONNECTOR ON PAGE
(20.03*)
C20
06
MOVE SYMBOL OUTLINE TO PAGE LAYOUT AREA
C70
02
EXTRACT PAGE AND BOX # OF REFERENCE FROM TAG TABLE AND PLACE ON PAGE
07
AFFIX BOX NUMBER AND TAG TO SYMBOL
03
MORE THAN ONE REFERENCE
NO
20.06
C20
YES
08
MOVE COMMENT LINES, IF ANY, FROM INPUT RECORD TO PAGE LAYOUT AREA
04
PLACE ASTERISK ON PAGE TO INDICATE ADDITIONAL REFERENCES
09
S OR J CODE
YES
22.03
C80
NO
05
20.06
C20
21.01

Fig. 8/21

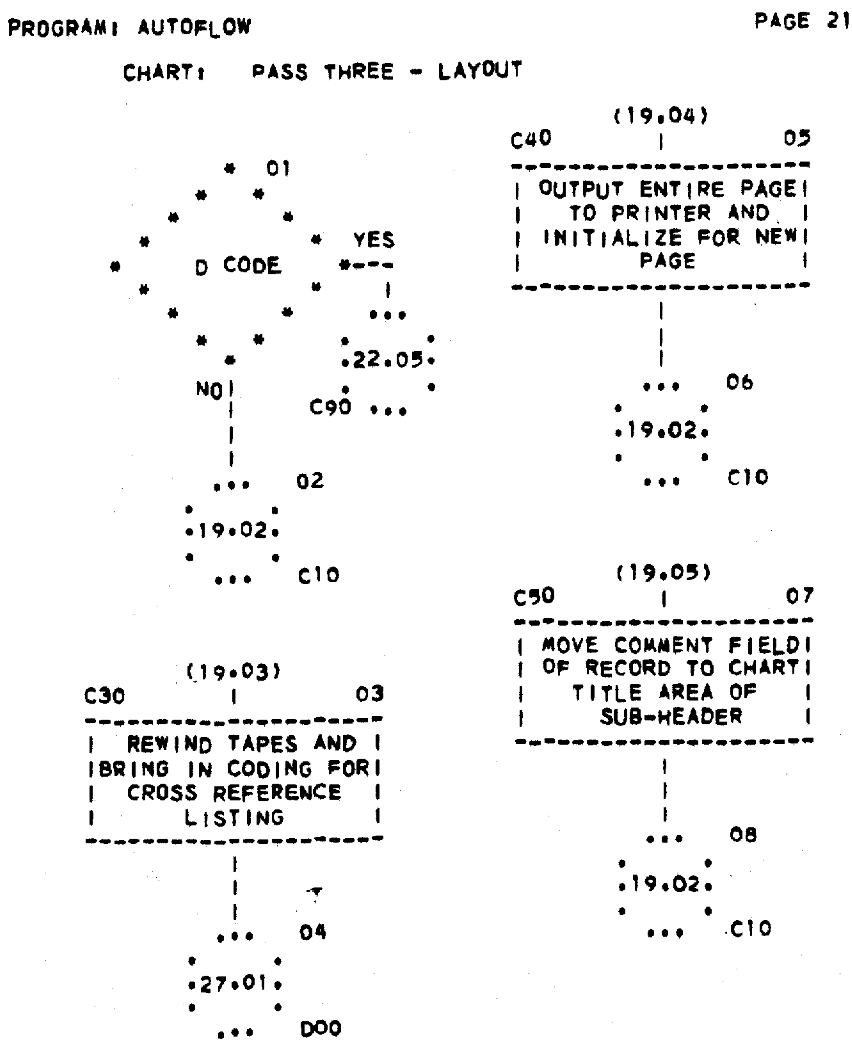

PROGRAM: AUTOFLOW
PAGE 21
CHART: PASS THREE - LAYOUT
01
D CODE
YES
22.05
C90
NO
02
19.02
C10
(19.04)
C40
05
OUTPUT ENTIRE PAGE TO PRINTER AND INITIALIZE FOR NEW PAGE
06
19.02
C10
(19.05)
C50
07
MOVE COMMENT FIELD OF RECORD TO CHART TITLE AREA OF SUB-HEADER
08
19.02
C10
(19.03)
C30
03
REWIND TAPES AND BRING IN CODING FOR CROSS REFERENCE LISTING
04
27.01
D00

Fig. 8/22

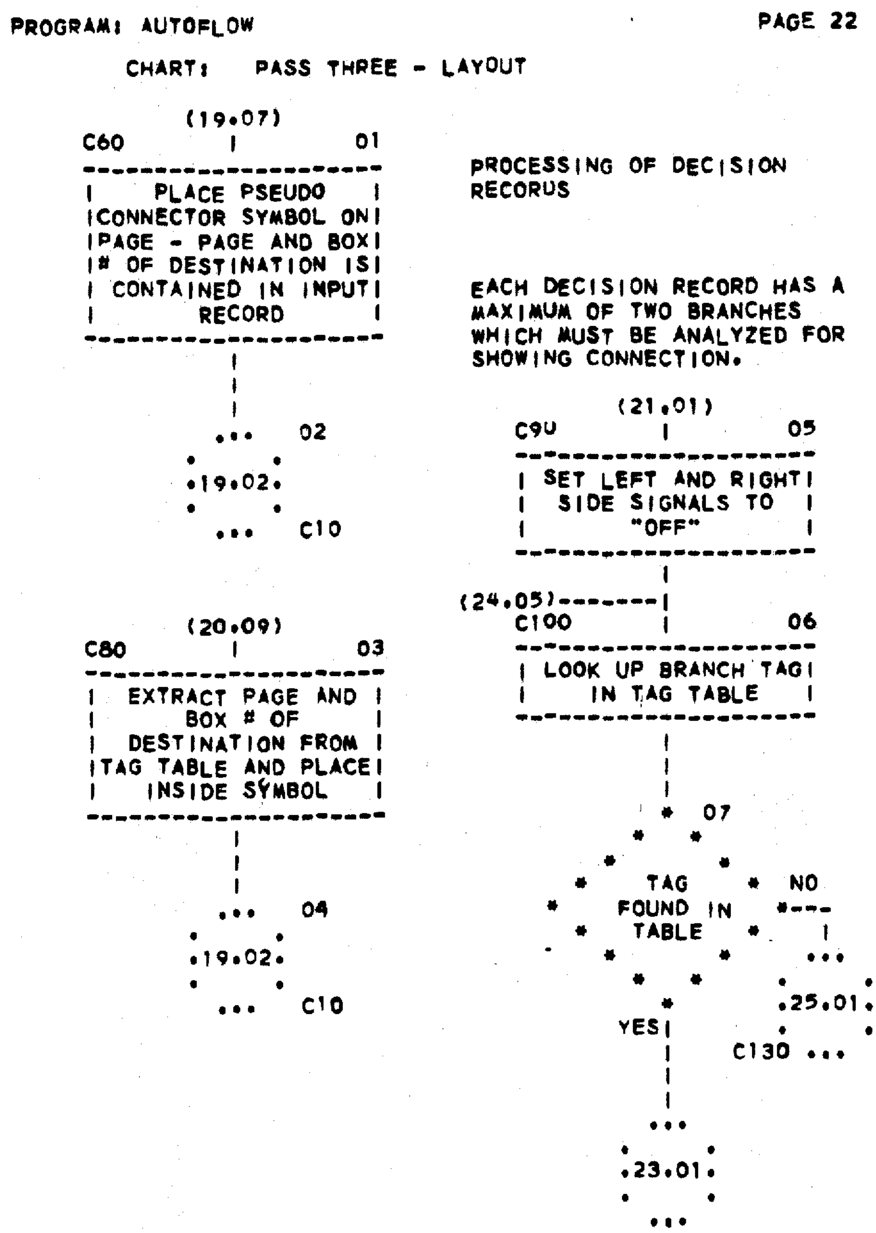

PROGRAM: AUTOFLOW
PAGE 22
CHART: PASS THREE - LAYOUT
(19.07)
C60
01
PLACE PSEUDO CONNECTOR SYMBOL ON PAGE - PAGE AND BOX # OF DESTINATION IS CONTAINED IN INPUT RECORD
PROCESSING OF DECISION RECORUS
EACH DECISION RECORD HAS A MAXIMUM OF TWO BRANCHES WHICH MUST BE ANALYZED FOR SHOWING CONNECTION.
02
19.02
C10
(21.01)
C9U
05
SET LEFT AND RIGHT SIDE SIGNALS TO "OFF"
(24.05)
C100
06
LOOK UP BRANCH TAG IN TAG TABLE
(20.09)
C80
03
EXTRACT PAGE AND BOX # OF DESTINATION FROM TAG TABLE AND PLACE INSIDE SYMBOL
04
19.02
C10
07
TAG FOUND IN TABLE
NO
YES
25.01
C130
23.01

Fig. 8/23

PROGRAM: AUTOFLOW PAGE 23

CHART: PASS THREE - LAYOUT

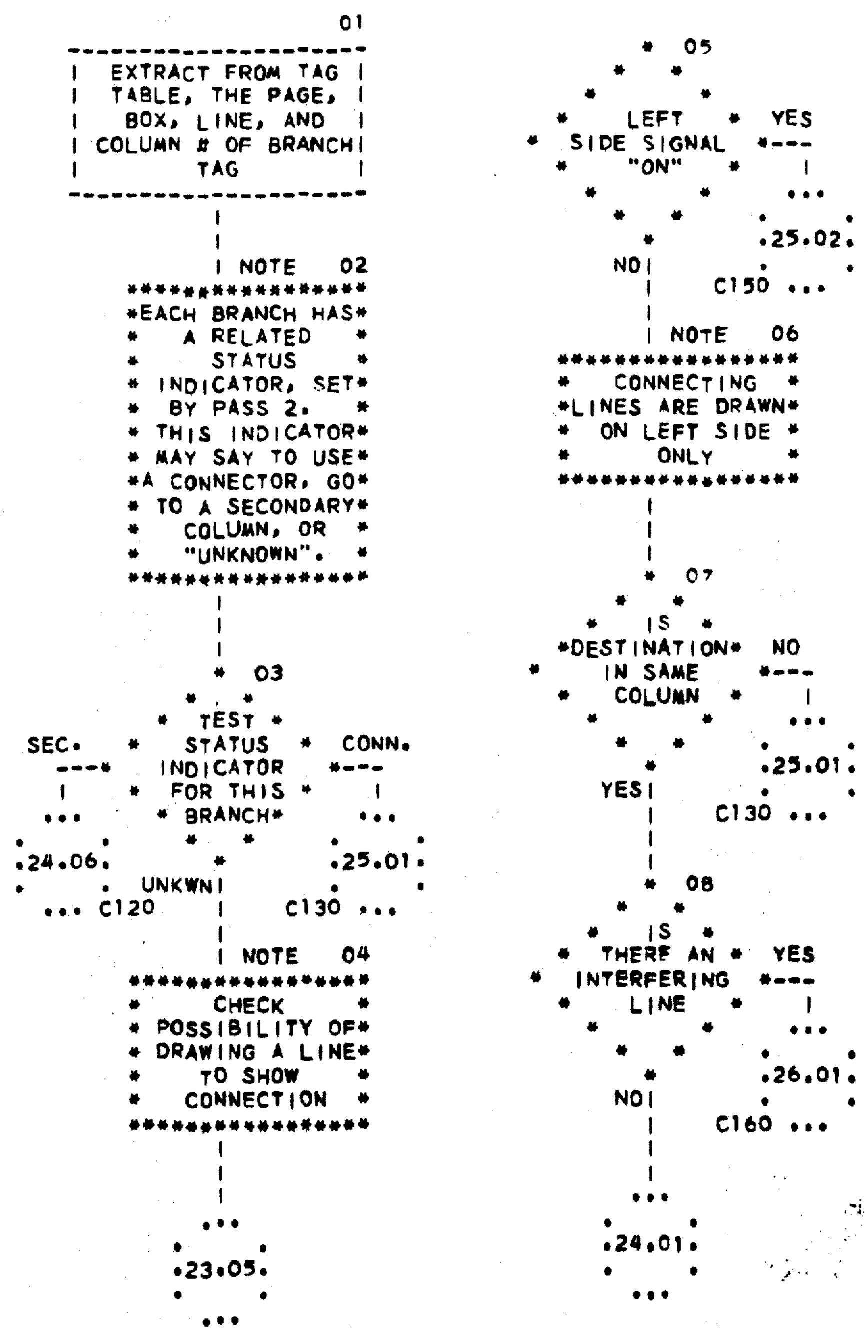


Fig. 8/24

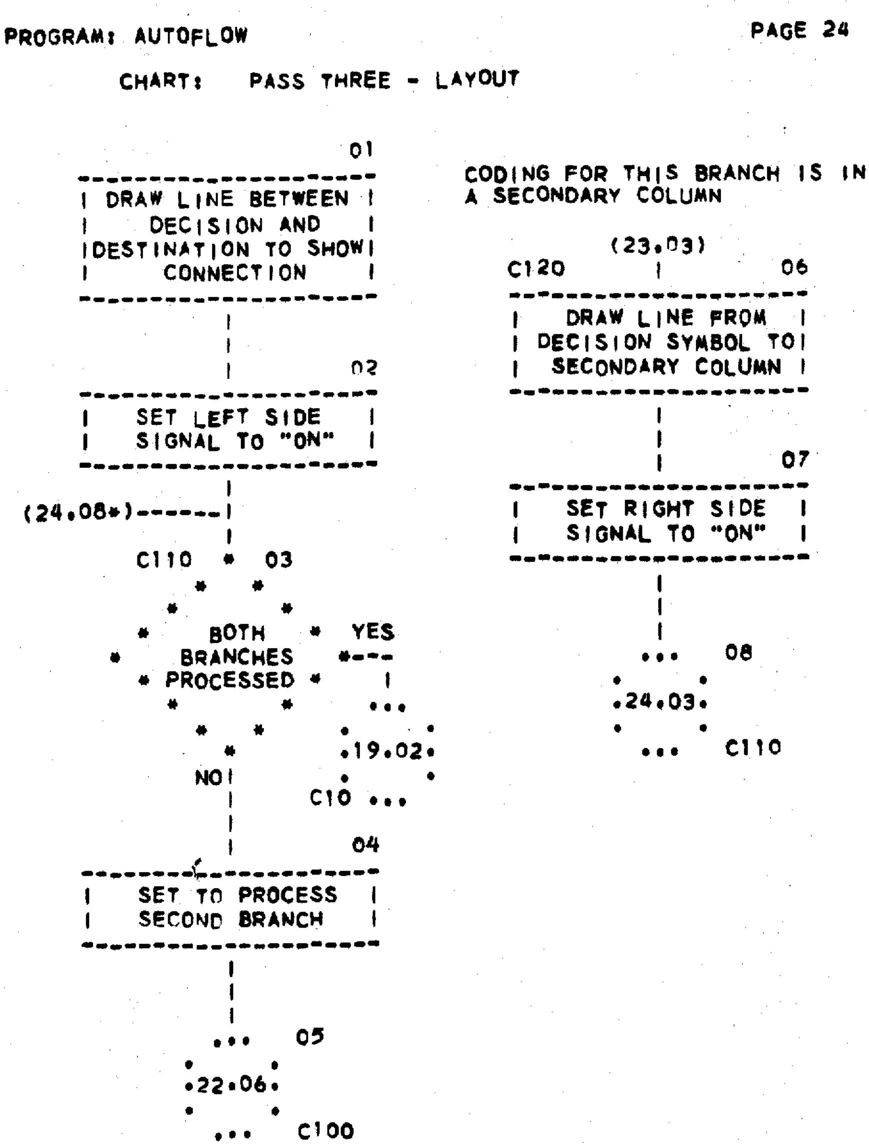

PROGRAM: AUTOFLOW
PAGE 24
CHART: PASS THREE - LAYOUT
01
DRAW LINE BETWEEN DECISION AND DESTINATION TO SHOW CONNECTION
02
SET LEFT SIDE SIGNAL TO "ON"
(24.08*)
C110
03
BOTH BRANCHES PROCESSED
YES
19.02
C10
NO
04
SET TO PROCESS SECOND BRANCH
05
22.06
C100
CODING FOR THIS BRANCH IS IN A SECONDARY COLUMN
(23.03)
C120
06
DRAW LINE FROM DECISION SYMBOL TO SECONDARY COLUMN
07
SET RIGHT SIDE SIGNAL TO "ON"
08
24.03
C110

Fig. 8/25

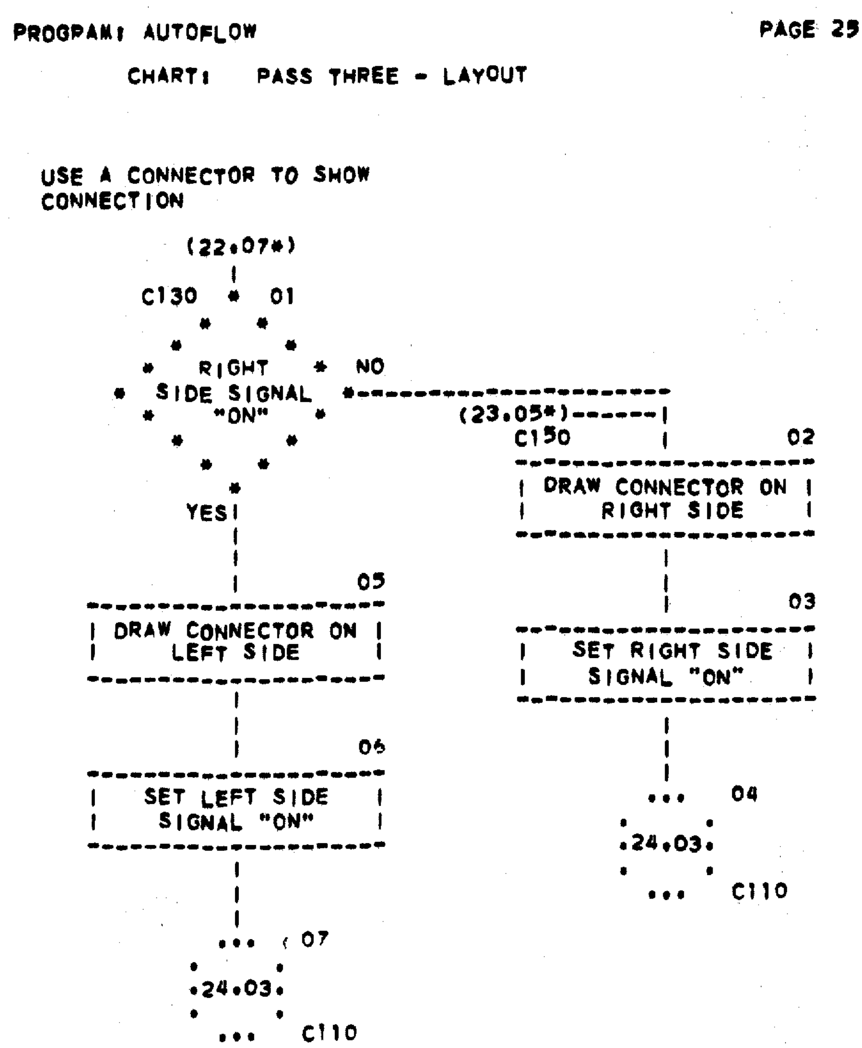

PROGRAM: AUTOFLOW
PAGE 25
CHART: PASS THREE - LAYOUT
USE A CONNECTOR TO SHOW CONNECTION
(22.07*)
C130
01
RIGHT SIDE SIGNAL "ON"
NO
(23.05*)
C150
02
DRAW CONNECTOR ON RIGHT SIDE
YES
05
DRAW CONNECTOR ON LEFT SIDE
03
SET RIGHT SIDE SIGNAL "ON"
06
SET LEFT SIDE SIGNAL "ON"
04
24.03
C110
07
24.03
C110

Fig. 8/26

PROGRAM: AUTOFLOW PAGE 26

CHART: PASS THREE - LAYOUT

INTERFERING LINE

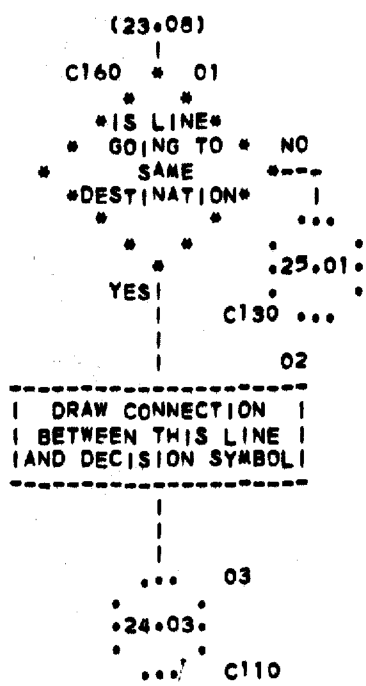


Fig. 8/27

```
PROGRAM: AUTOFLOW                                          PAGE 27

      CHART:   CROSS-REFERENCE LIST

                                   ------------(27.05*)
                                   A     D10       |       03
TWO INTERNAL TABLES ARE USED|      --------------------
IN PRODUCING THE CROSS      |      | READ RECORD FROM |
REFERENCE LISTING - THE TAG |      | PASS 3 INPUT FILE |
TABLE AND THE REFERENCE     |      --------------------
TABLE.  A TAG WILL HAVE ONE |                |
ENTRY IN THE REFERENCE TABLE|                |
FOR EACH REFERENCE TO IT.   |                |
EACH REFERENCE ENTRY        |                *  04
CONTAINS THE TABLE LOCATION |              *   *
OF THE NEXT ENTRY TO THE    |            *       *
SAME TAG.  THE FINAL        |          *    END    *  YES
REFERENCE TO A TAG HAS AN   |        *     FILE      *---
APPROPRIATE INDICATOR.      |          * INDICATOR *    |
                            |            *       *     ...
        (21.04)             |              *   *     .     .
   D00     |         01     |                *       .28.07.
   --------------------     |             NO|        .     .
   | CREATE AN ABRIDGED|    |               |       D30 ...
   | TAG TABLE FROM THE|    |               |
   |  MAIN TAG TABLE.  |    |               |
   |EACH ENTRY CONTAINS|    |               *  05
   |  THE TAG AND ITS  |    |             *   *
   | ASSIGNED PAGE AND |    |           *       *
   | BOX NUMBER.  ONLY |    |   NO    *   D, J,   *
   |   TAGS WHICH ARE  |    -------*   OR S CODE   *
   |   REFERENCED ARE  |             *           *
   | PLACED IN ABRIDGED|               *       *
   | TABLE - EACH MAIN |                 *   *
   | TABLE ENTRY HAS AN|                   *
   | INDICATOR TO SHOW |              YES|
   |       THIS.       |                 |
   --------------------       (28.06)----|
            |                   D20      |          06
            |                   --------------------
            |         02        |LOOK UP DESTINATION|
   --------------------         |TAG IN ABRIDGED TAG|
   |SORT ABRIDGED TABLE|        |       TABLE       |
   | INTO PAGE AND BOX |        ---------------------
   |  NUMBER SEQUENCE  |                 |
   --------------------                  |
            |                            |
            |                           ...
            |                         .     .
           ...                        .28.01.
         .     .                      .     .
         .27.03.                        ...
         .     .
           ...
```

Fig. 8/28

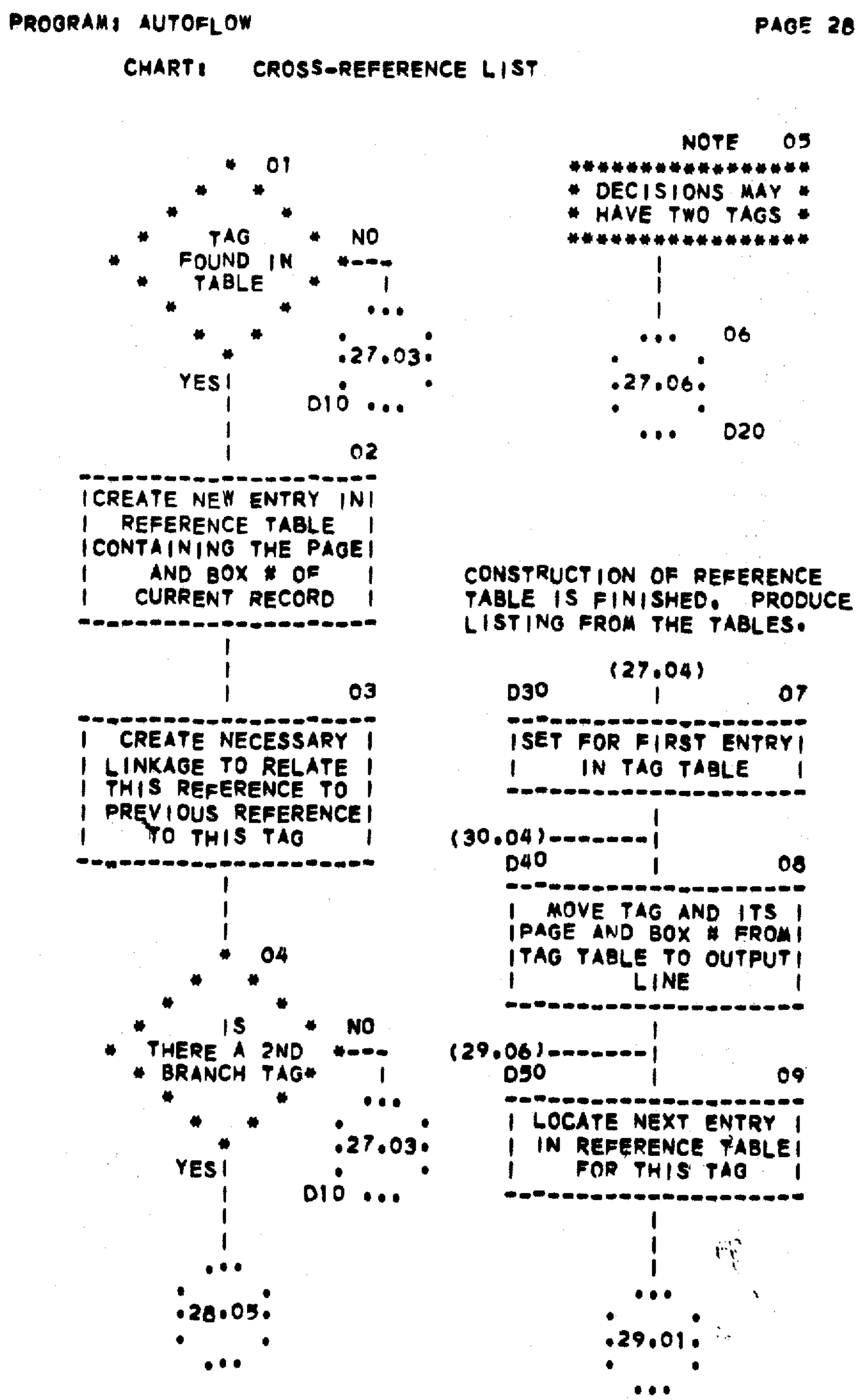

PROGRAM: AUTOFLOW
PAGE 28
CHART: CROSS-REFERENCE LIST
01
TAG FOUND IN TABLE
NO
27.03
D10
YES
02
CREATE NEW ENTRY IN REFERENCE TABLE CONTAINING THE PAGE AND BOX # OF CURRENT RECORD
03
CREATE NECESSARY LINKAGE TO RELATE THIS REFERENCE TO PREVIOUS REFERENCE TO THIS TAG
04
IS THERE A 2ND BRANCH TAG
NO
27.03
D10
YES
28.05
NOTE 05
DECISIONS MAY HAVE TWO TAGS
06
27.06
D20
CONSTRUCTION OF REFERENCE TABLE IS FINISHED. PRODUCE LISTING FROM THE TABLES.
(27.04)
D30
07
SET FOR FIRST ENTRY IN TAG TABLE
(30.04)
D40
08
MOVE TAG AND ITS PAGE AND BOX # FROM TAG TABLE TO OUTPUT LINE
(29.06)
D50
09
LOCATE NEXT ENTRY IN REFERENCE TABLE FOR THIS TAG
29.01

Fig. 8/29

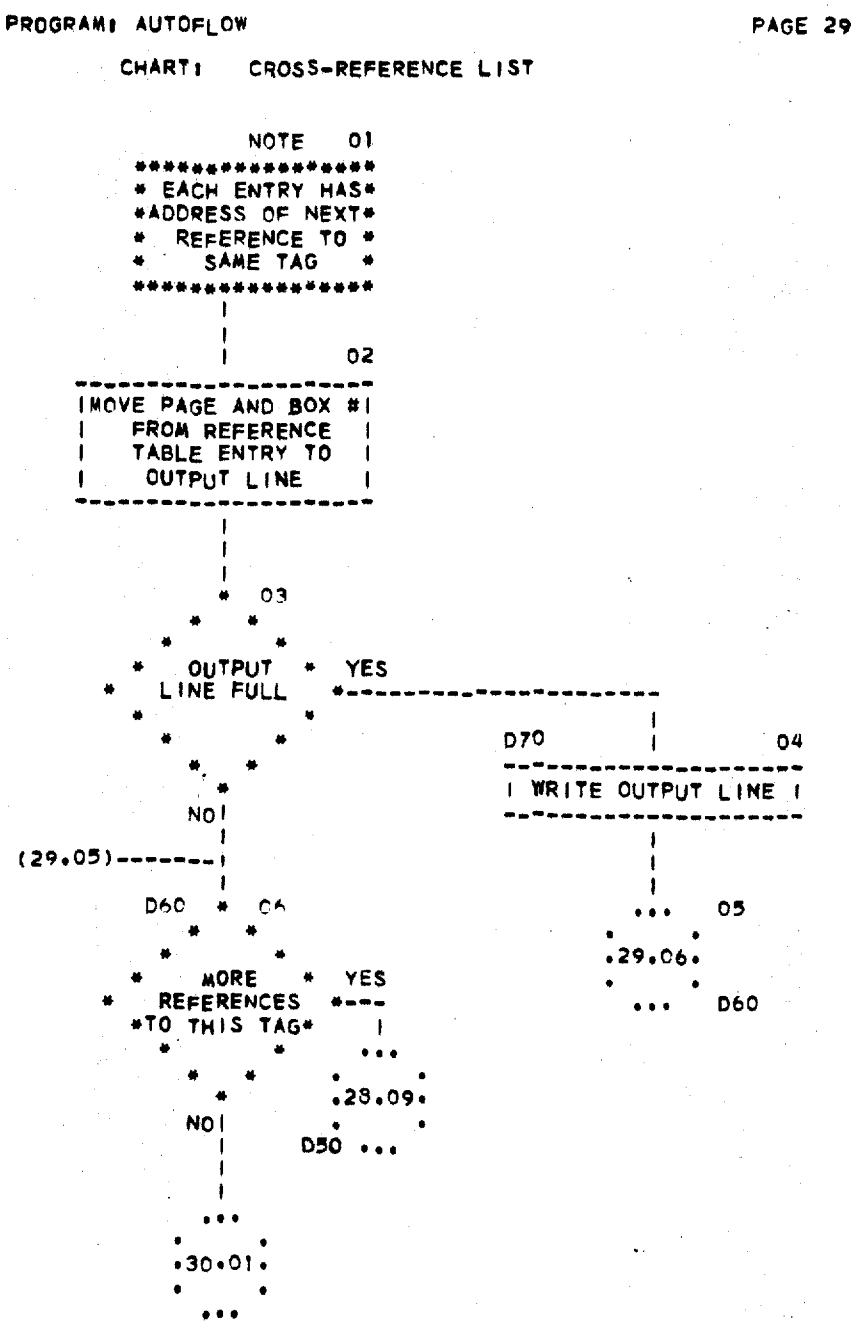
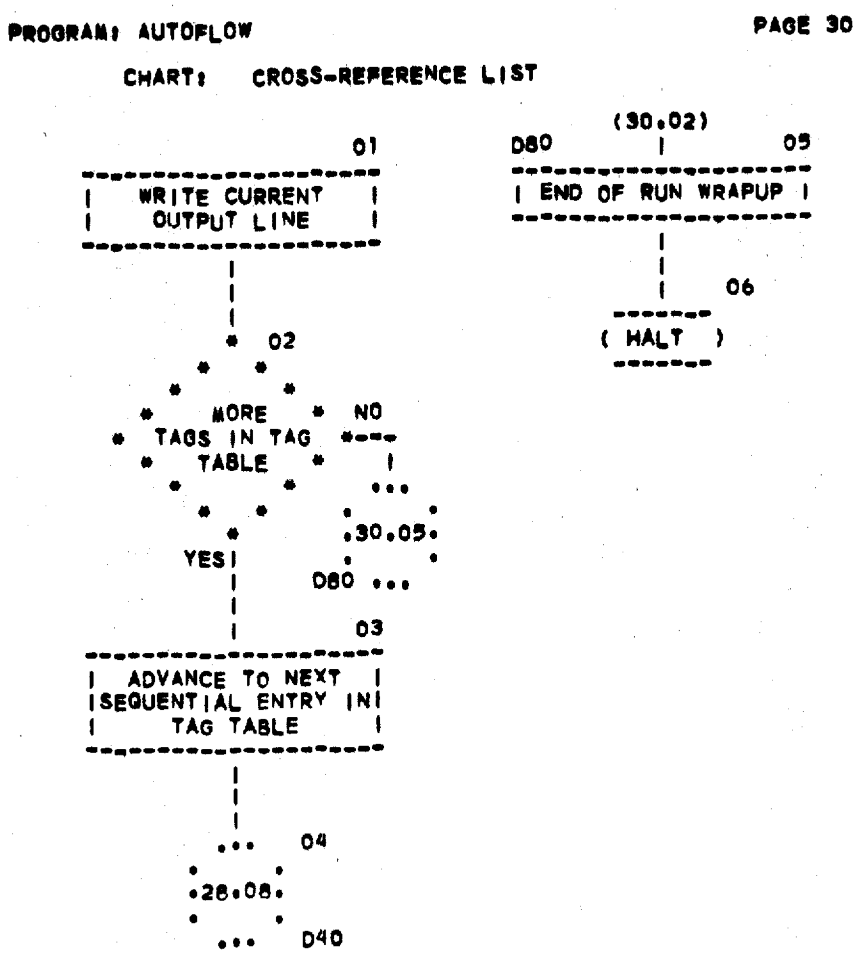

PROGRAM: AUTOFLOW
PAGE 29
CHART: CROSS-REFERENCE LIST
NOTE 01
EACH ENTRY HAS ADDRESS OF NEXT REFERENCE TO SAME TAG
02
MOVE PAGE AND BOX # FROM REFERENCE TABLE ENTRY TO OUTPUT LINE
03
OUTPUT LINE FULL
YES
D70
04
WRITE OUTPUT LINE
05
29.06
D60
NO
(29.05)
D60
06
MORE REFERENCES TO THIS TAG
YES
28.09
D50
NO
30.01

Fig. 8/30


PROGRAM: AUTOFLOW
PAGE 30
CHART: CROSS-REFERENCE LIST
01
WRITE CURRENT OUTPUT LINE
02
MORE TAGS IN TAG TABLE
NO
30.05
D80
YES
03
ADVANCE TO NEXT SEQUENTIAL ENTRY IN TAG TABLE
04
28.08
D40
(30.02)
D80
05
END OF RUN WRAPUP
06
(HALT)

United States Patent Office

3,533,086

Patented Oct. 6, 1970

1

3,533,086

AUTOMATIC SYSTEM FOR CONSTRUCTING AND RECORDING DISPLAY CHARTS

Martin A. Goetz, Princeton, N.J., assignor to Applied Data Research, Inc., a corporation of New Jersey

Continuation-in-part of application Ser. No. 512,113, Dec. 7, 1965. This application Dec. 24, 1968, Ser. No. 786,782

Int. Cl. G06f *9/06*

U.S. Cl. 340—172.5 45 Claims

ABSTRACT OF THE DISCLOSURE

A data processor system for automatically making two-dimensional flow charts forms chain sequences of the flow chart symbols and allocates the symbol chains in parent and branch and sub-branch sequences as clusters to successive flow chart pages.

BACKGROUND OF THE INVENTION

This invention relates to a system for automatically constructing and recording display charts and particularly flow charts representative of control systems for digital computers.

This application is a continuation-in-part of copending application Ser. No. 512,113, filed Dec. 7, 1965, now abandoned.

Computer programs that are used to control the sequential operations of digital computers are made up of sequences of hundreds or thousands of computer instructions or commands which have complex interrelationships. The relationships of these instruction sequences, whether presented in machine coding or in machine dependent or independent languages, are difficult to interpret, even when they are read by skilled programmers. For that reason the program is generally presented in the form of a flow chart, which graphically presents the logic flow of machine operation and enables the programmer and users of the program to more readily interpret and understand the program. When a programmer constructs a new program he may develop rough sketches of a flow chart prior to implementation of the program, but commonly, such sketches are an inadequate description of the final program that is implemented, which may incorporate numerous changes and revisions. Moreover, a draftsman is needed to convert the sketches to suitable drawings, and the drawings, in turn, should be checked to ensure that no errors have been made in the transcription. Due to the tediousness of making a good flow chart, the pressure of other duties, and changes in personnel, the flow chart documentation of a program by the programmer is often incomplete and inaccurate. Yet, without reliable flow chart documentation, skilled personnel who were not involved in the original design of the program have great difficulty in learning and understanding its construction and operation, and in developing modifications and variations of the program, as circumstances often require. In addition, as a program is updated or revised, procedures are needed for readily updating the flow chart documentation.

SUMMARY OF THE INVENTION

Accordingly, it is among the objects of this invention to provide a new and improved data processing system for automatically producing flow chart documentation of a computer program.

Another object is to provide a new and improved automatic flow chart documentation system for computer programs that automatically produces from a computer program a flow chart which is an accurate and informative graphical representation of the program.

2

Another object is to provide a new and improved automatic flow chart documentation system for computer programs that relieves the programmer of documentation chores and makes it possible to obtain documentation immediately upon the program being constructed.

Another object is to provide a new and improved automatic flow chart documentation system for computer programs which assists a programmer in debugging the programs that he constructs and in revising the program as may be required.

Another object is to provide a new and improved flow chart documentation system for computer programs by means of generally available digital computers.

Another object is to provide a new and improved method of operating digital computers to produce flow chart documentation of computer programs.

Another object is to provide a new and improved computer programming system for operating stored program computers to produce automatically flow chart documentation of other computer programs.

In accordance with an embodiment of this invention a computer program is provided for operating a stored program digital computer to perform the flow chart documentation of other computer programs. The computer program to be documented is in the form of groups or combinations of digital signals that are treated as data by the digital computer when it is operated in accordance with this invention. The digital computer operates on each of the successive groups of data signals representing the sequences of instructions or instruction groups of the program to be documented, and determines therefrom what type of instruction is represented and the length of the column display required for presenting each instruction as a diagrammatic block along a column of a flow chart. Chains of such blocks between successive transfer types of instructions are established and the length of the chain for display on the flow chart page is determined. Destination tags in the data blocks are identified, and a tag table is developed of those tags and their relations to the associated blocks. The locations of successive chains in a main path of the program are allocated to successive columns of the display pages. Branch instructions of those chains are handled specially by identifying the destination (or tagged) chain to which the program branches and, where adequate space is available for the tagged chain in a column adjacent to the column of the main path, allocating the tagged chain to the adjacent column. Chains are allocated in sequential order to the main flow except where branch instructions are encountered; for the latter, the main path allocation is interrupted to allocate the branch chains. The locations of the connecting paths between blocks in the same and adjacent columns and of connectors to blocks in non-adjacent columns of the same or different pages are established. A display is provided of the interconnected chains with connecting paths where possible to represent paths between blocks in the same and adjacent columns and connector symbols are drawn where such paths cannot be drawn.

In other embodiments, modified forms of the invention are used.

BRIEF DESCRIPTION OF THE DRAWING

The foregoing and other objects of this invention, the various features thereof, as well as the invention itself may be more fully understood from the following description when read together with the accompanying drawing, in which:

FIG. 1 is a schematic block diagram of a data processing system incorporating a control program for flow chart documentation in accordance with this invention;

FIG. 2 is a schematic block diagram of a fragment in outline form of the type of chart that is produced by means of this invention;

FIG. 3 is a general schematic block diagram of a computer program embodying this invention and used in the system of FIG. 1;

FIG. 4A and B are schematic block diagrams of outline fragments of a form of computer program flow chart produced in accordance with modifications of this invention;

FIG. 5 is a schematic block diagram of an outline fragment of another form of computer program flow chart produced in accordance with modifications of this invention;

FIGS. 6A, B, C, D and E are schematic graphical diagrams of flow chart patterns produced in accordance with another modification of this invention;

FIGS. 7A, B, C, D and E are a series of schematic flow chart diagrams that together illustrate graphically details of a modified form of computer program embodying this invention and used in the system of FIG. 1; and

FIGS. 8/1 to 8/30 are a series of schematic flow chart diagrams that together illustrate graphically details of a computer program shown in FIG. 3.

In the accompanying drawing, corresponding parts are identified by similar reference characters throughout.

TABLE OF CONTENTS

GENERAL SYSTEM

In the embodiment of the invention in FIG. 1, a data processing system constructed in accordance with this invention is shown, and it includes a memory **102**, an arithmetic unit **104**, and a set of controls **106**. This system may be a suitable form of digital computer in which

data stored in the memory **102** is supplied under the direction of the controls **106** to the arithmetic unit **104** for processing and then returned to the memory to be stored at appropriate locations. The controls **106** determine the memory locations from which information is taken to be processed in the arithmetic unit **104** as well as the memory locations to which it is returned after processing, and also determine the timing of the flow of electrical signals representing the information. The controls **106** also determine the particular operations performed by the arithmetic unit as well as the time interrelations thereof. The flow of signals to and from the memory is indicated by solid lines, while the flow of control signals from the controls **106** to operate the memory **102** and the arithmetic unit **104** is shown by broken lines. In practice, various kinds of such control lines are required, and the details are omitted here since appropriate arrangements are well known in the art and they are unnecessary to an understanding of the invention.

The data processing system of FIG. 1 may be any of various well-known types of systems such as that employing a stored program in the form of signals stored in the memory and representing sequences of control instructions or commands which select the operations to be produced by the controls **106** as they are needed to perform the desired operations. Such a stored program is effectively a part or extension of the controls **106** and is commonly stored in the memory in a section set apart for that purpose. A section **102*a*** of the memory **102** is labeled to indicate that the control program for the documentation of a flow chart (F/C) is shown as part of that memory. Alternatively, the controls **106** may have a fixed program wired and/or built-in which may take the form of logic combinations of gates and other circuits to perform the proper sequence of operations that make up the logic of the program, all in accordance with techniques that are well-known in the art. The operations required of the flow chart documentation system of this invention are generally large in number and have complex interrelationships. Therefore, a stored program is the preferred form of control system for presently available computers, and an embodiment of such a program is described below.

The input portion of the data processing system is represented as a magnetic tape unit **108** operated by appropriate signals from the controls **106** to supply groups of combinatorial signals to the input area **102*b*** of the memory **102**, which input signals represent the program which is to be documented. That is, the data inputs of the system are themselves successive sections of another computer program, which are processed as data to provide a flow chart representation of the logic of the input program. Another magnetic tape unit **109**, operated by the controls, carries the signals that form the flow chart control program, and this program is read into the memory section **102*a*** when the computer is to be operated in accordance therewith. The entire F/C program may be read into memory section **102*a*** at the beginning; or, since the program is formed as a sequence of subdivisions or passes, the subdivisions may be separately read into memory as required by the program. Other magnetic tape units **110** and **111**, operated by the controls **106**, receive records from the memory output area **102*c*** and supply these records back to the input area **102*b*** thereof during different stages of operation of the system. In use, the tape units operate effectively as portions of the computer memory system. An output display device **112** is also operated by the controls **106** to produce a graphical display of the final flow chart produced by the system. This display device **112** may be a high speed printer (e.g. a line-at-a-time printer), a digital plotter recorder, a cathode ray tube display or recording system, or any other appropriate form of display or recording system. The display device may be operated on-line directly from the memory **102**, or the output from the memory may be written on to an appropriate tape and used to operate the display device off-line in any suitable manner.

FLOWCHART SYMBOLS AND FORMATS

FIG. 2 illustrates an outline of a flow chart that is produced in accordance with the system of this invention. The flow chart documentation program of this invention examines specific fields of each instruction line and other data sections of the program to be documented and produces a standardized flow chart by means of the display device **112**. In accordance with one form of the invention, the input program is in an assembly language and the flow chart that is produced is divided into four columns (the invention may be used to supply flow charts of any desired size having one or more columns). The first and third columns **114** and **116** may be used for depicting the main flow of the program and the other two columns **118** and **120** would then be reserved for branches from the main flow columns. However, as described below, it is preferred to have all four columns available for display of the main flow logic of the program and to use the next adjacent column for display of the logic that branches from a main flow column. Various types of graphical symbols are used to represent the different types of instruction or instruction groups of the program that are being documented. One such symbol is a rectangular box **122** representing a PROCESS; another symbol is a diamond-shaped box **124** that represents a DECISION or branch (that is, conditional transfer) instruction. Seven other different symbols are utilized and illustrated in FIG. 2 as is described hereinafter. Symbols are assigned numbers sequentially beginning with "01" on each page, and all cross-referencing to other symbols is in the form "XX.YY," where XX is the page number and YY the number of the symbol on that page, which provides for 99 symbols for each of 99 pages, and which can be readily modified for larger numbers if needed. Symbol sequence numbers are printed above and to the right of all symbols on the chart, as shown in FIG. 2, where box **122** carries the symbol sequence number "01." The symbols are numbered in sequence from top to bottom of a column, and, in the main flow columns, from the bottom (or exit) of one column to the top (or entry) of the next reading from left to right. Branches from a DECISION symbol are from the side corners of the diamond, and these branches may be connected to the entry point of another symbol in the same column (see line **123**) or to the entry of a symbol in the adjacent column (see line **125**). The symbols in a branch column are sequentially numbered (in one embodiment) starting from the number of the DECISION symbol from which it branches; and the symbols below that DECISION in the main flow column have numbers that continue after those of the branch column. Thus, the number of branch block **126** is "03" following that of DECISION **124**, and the number of block **127** is "08" following that of the last symbol in branch column **118**.

The documentation program in one embodiment, examines four fields of an input program presented in a fixed formal assembly language and produces the flow charts therefrom; these fields are the symbolic tag, the comments of the programmer interpreting each instruction or group of instructions, a special flow chart code located in a predetermined part of the Comments field, and operands (such as the operation code and certain addresses) that supply the destination tags or addresses of transfer and branch instructions. The following eleven flow chart codes are utilized in such an assembly language format for the system of FIG. 2:

"P"—PROCESS	"H"—HALT
"C"—CONTINUATION	"N"—NOTE
"S"—SUBROUTINE	"B"—CHART TITLE
"J"—JUMP	"D"—DECISION
"E"—EXIT	"W"—SWITCH

"T"—TEXT

Eight different chart symbols are used to represent the eight different classes of data processing, and "B" and "T" identify TITLE and TEXT that are to be printed on the chart. "C" is used to identify a continuation of a comment that started in a preceding record of the program. It has been found convenient to insert the F/C code after the Comments field separated therefrom by spaces. Due to the records being fixed in size, the number of characters usually being that of a punch card, successive records are used to carry extensive comments.

In FIG. 2 the PROCESS (or P-code) symbol is shown as a rectangular block, such as the block **122**, and the Comments portion of the corresponding instruction is inserted in the symbol in the manner illustrated in FIG. 2 and as shown in greater detail in FIG. 8 (the latter may be referred to as an illustration of a flow chart, in two-column form, produced by this invention). This PROCESS symbol is variable in column length depending on the length of comments. The TITLE represented by code-code-B is placed on the top of each page of the flow chart, as indicated at the top of column **114** of FIG. 2. The T-code for TEXT indicates that the textual material supplied in the assembly language is to be printed out without a special chart symbol (see column **116**). In column **114**, diamond-shaped symbol **124** represents a DECISION, and the accompanying comments are edited and inserted in that symbol as indicated. Either one or two branches of coding may be shown as coming from the side corners of the DECISION symbol. Labels are supplied to the branch paths from the DECISION symbol in accordance with a special code or in accordance with the Comments field of the input instruction. The documentation program of this invention determines how the lateral branches are to be depicted on the chart; they may be shown by a horizontal line **125** connecting to an adjacent column as indicated by the branch EQL from the box **124** to process box **126** in column **118**; or by a connecting line **123** from the LOW branch of box **124** down to the input of block **128** in column **114**; a third method (where such connecting lines cannot be drawn) is by means of a connector symbol in the form of a circle **130** connected to the DECISION branch and containing the cross-reference identification XX.YY of the flow chart symbol to the input of which it connects. Thus, in the case of connector **130**, the HIGH branch from the DECISION block-**10** goes to block-**13** on page 1, which is shown in column **120**.

Another symbol is that for SUBROUTINES (S) such as that shown by block-**05** in column **118**. This is a hexagon-shaped box in which the Comments field is written, and in a separate section in the left of the box in the cross-reference is given to the location of the details of the subroutine. As illustrated. SUBROUTINE-**05** is cross-referenced to **01.20**, which one can readily locate in column **116** on the same page. The cross-reference back to SUBROUTINE block **01.05** is shown at the input to box **01.20** so that the reader of the flow chart may readily determine the entry into that section of the program and interrelate the different positions thereof.

The JUMP code (J) is represented by a circle as indicated by block-**07** at the bottom of column **118**. The circle has at the lower right the tagged destination to which the program jumps, and contains within it the page and box number of that destination in the flow chart so that it can be readily located. The tag of the entry point of a block is at the upper-left of a block, as the tag, "JAN" for block **126**.

The E-code representing EXIT is illustrated by a circle that terminates a chain of blocks and contains the word "exit," as shown by circle-**23** in column **120**. Similarly, the H-code for HALT terminates a chain of coding as shown by circle-**17** in column **116**. The N-code, used for NOTES, is represented by a rectangular block (see block-**5**) which can be varied in column length to contain the associated comments, and which is offset with its left side indented to distinguish it from a PROCESS rectangle.

The W-code is a diamond-shaped symbol similar to a DECISION and it contains the word "switch." Its branches are handled in a similar fashion to the DECISION box as illustrated by the example of block-**14** in column **120**.

The destination tags where they are provided in the comments field of the input program (or otherwise in the operand codes of that program) are picked up and supplied to the transferring symbol, where it is not directly connected to the destination symbol. For example, block **130** is a connector symbol that has the destination tag "CAN" printed next to it, for that tag is the input of block-**13** in column **120**. "CAN" is also printed at the input of block-**13** and if the other blocks have input tags they are similarly printed. Cross-references to show the originating points for entries to blocks of the flow chart are by way of the page and block number of each symbol. An example is the cross-reference back to block **01.05** that is set forth in parentheses at the input to block-**20** in column **116**; another example is the tagged entry "RAN" to block-**05** that is shown in parentheses (namely, **01.10**), which cross-references back to the branch from connector **132** in column **114**. In addition where more than one cross-reference entry exists, rather than indicating all of the symbol locations, an asterisk is provided, as at the entry of block-**13** in column **120**, and a separate cross-reference table lists all such entries, as is discussed in further detail herein below.

In accordance with the documentation program of this invention, each chain of symbols is terminated by a JUMP or EXIT symbol. In addition, where space limitations do not permit the printing of a long chain of symbols until it terminates in that fashion, a special connector symbol is generated by the F/C program to set forth the continuity of the main flow of the program. Thus at the bottom of the column **114**, such a pseudo-connector symbol is shown as a circle containing the location (**01.11**) of the next instruction in the main path, which in the case illustrated is to block-**11** on the same page, namely, the first block in column **116**. Also at the bottom of column **116** a pseudo-connector circle is generated containing a cross-reference to the first block of page 2, for the next symbol of the main flow of the program.

PHASES OF FLOWCHARTING SYSTEM

FIG. 2 is referred to hereinafter to illustrate the development of the flow chart as the operations called for by the documentation program are sequentially performed. FIG. 3 is a block diagram of the four main phases of the F/C program. In the specific embodiment of the invention described hereinafter, these four phases correspond to four separate sections of the program and four passes of data. During Pass I, block **140**, the editing phase is performed in which the input data is accepted and edited, the column lengths of individual symbols are fixed, individual chains of logic are established and their lengths determined, and a Chain Table and the skeleton of a Tag Table are constructed. During Pass II, block **142** output records of Pass I are processed to complete the Tag Table; individual symbols are assigned locations on successive pages of the flow chart, and assigned to a specific part of a column within the page. Successive main flow chains are processed along with the branch chains. During Pass III, block **144**, the Tag Table is used to print a table of contents, the successive pages of the flow chart are laid out, connecting lines and symbols set out, and each page is printed. During an additional pass, Pass IV, block **146**, a table of cross-references is developed and printed out. The program of this invention is not limited in its form to any particular number of passes of the data; the particular number varies with the computer that is used and with the availability and division of memory space (say, in the random-access section of the memory as against drum or tape memory storage) in the computer

as well as with the complexity of the processing operations that can be performed by the computer and of the flow chart that is desired.

INPUT DATA

The input data for the assembly language program generally takes the form of successive records corresponding to the quantity of information that can be developed on an individual punch card. This record may have the information arranged in any prescribed order, and for one assembly language, it contains the following ordered fields of information:

Assembly line #; tag, if any, of current record; operation code; A-Address; index instruction code; B-Address; Comments.

The Comments field carries the programmer's interpretation of the data processing operation, so that the operation code is not needed and is not used for that purpose in the present embodiment; nor is the index instruction code and B-Address. However, for the purpose of picking up destination tags, if they are not carried by the Comments, the operation code and A and B-Addresses are examined. That is, the operation code is examined to determine if it is a conditional branch instruction, and if so, the A-Address is used for one branch destination and the B-Address for any second branch destination that may be involved. The Comments field may be used to carry any desired destination tags to which the current record transfers, which destinations are set off parenthetically at the beginning of that field by asterisks, e.g. as follows: *TAG*. In addition, the labels for the branches of DECISION symbols may be supplied with those destination tags in a special format described below. In addition, in accordance with the present embodiment of the documentation system, an additional field of flow chart code is provided; that is, one of the aforementioned eleven flow chart code characters; and, in the case of DECISION OF D-codes, an additional optional code character may be provided representing the different classes of labels for common DECISION branches, as follows:

"Y"—YES; NO
"Z"—NO; YES
"I"—HIGH; LOW
"L"—LOW; HIGH
"Q"—EQUAL; UNEQUAL
"U"—UNEQUAL; EQUAL
"#"—PLUS; MINUS
"—"—MINUS; PLUS
"3"—EQUAL; HIGH; LOW

Program Logic

FIG. 8 is a schematic flow chart diagram of the general logic of the documentation program; it consists of 30 sections or pages of flow chart, and FIG. 8 is numbered as FIG. 8/1 to FIG. 8/30 to identify those 30 flow chart pages. Each of these pages of FIG. 8 is presented in the form of flow chart that would be produced under the automatic control of the documentation program itself; and, in fact, that program was used to develop the flow chart of FIG. 8. Due to the size limitations of the patent drawings, only a quarter page of the high-speed-printer page was utilized, which afforded room only for two (of the available four) columns in width and a half-column in length. Thus, FIG. 8 illustrates an actual flow chart developed by the documentation program, but is much simplified in that less information is provided on a single page. Reference is made to FIG. 2 for a representation of the four-column format that is produced by a preferred embodiment of the documentation program; and FIG. 8 illustrates that a two-column format may also be produced with minor modifications of this program. Each of the pages of FIG. 8 has a program title and chart title together with a page number in the same fashion as is developed by the documentation program itself. As explained above, each of the blocks is referenced as XX.YY, where XX is the page number and YY is the block number thereon.

PASS I

In FIG. 8/1, following the program and chart titles and page number at the top, a body of text material sets forth the functions of Pass I as that of editing the source information and constructing Tag and Chain Tables. Block **01.01** is a PROCESS symbol whose Comment field indicates that its function is that of setting the record sequence number counter to zero. For simplicity of illustration, the flow diagram of FIG. 8 omits certain other preliminary operations such as those known as "housekeeping" operations, and those of calling in Pass I of the F/C program, clearance of memory areas where required, and entry of constants, and the like, which are routine in nature, and which would be readily apparent to those skilled in the art and are not needed for an understanding of the invention.

The main flow continues with the second block **01.02**, and the process of reading the first source record is performed. This operation of the computer may call for a series of detailed instructions, depending on the computer construction, by which the input tape **108** is controlled to operate momentarily, and the first available record of the input program is read and stored in a predetermined order so that its fields in a fixed format are placed in prescribed sections of the input area **102*b*** of the memory **102**. The first record that is read in may be assumed to be a special control record, which may be marked with a special code identification if desired, and which should contain the name of the program. This program title is stored, under the operation of block **01.03**, in a prescribed primary storage area of the memory for use by Pass III in composing and printing out each page of the program; then this first card is dropped.

The next available source record is read, block **01.04**, from the input tape **108** into the input area **102*b***; and the field for the F/C code is checked, block **01.05**, to determine if it is a valid F/C code other than "C." If the check indicates that the answer is NO, the program branches in a loop back to block **01.04** to read the next available input record into the input area **102*b***, and DECISION **01.05** checks the F/C code of that record in the same way. This loop is repeated until the answer to the check is YES, and the program then continues on the block **01.06** (as indicated by the reference in the pseudo-connector symbol at the end of the first column of FIG. 8/1). As indicated by NOTE **01.06**, the input memory area **01.06** now contains a source record that is to be processed, since it contains a valid F/C code other than "C." A C-code is not processed, where it does not follow another code type.

The main-flow logic of Pass I then begins with block **01.07**, and all of the fields of the first record to be processed are moved from the input area **102*b*** to corresponding sections of the work area **102*d***. That is, the following fields of that record from the memory input area **102*b*** are moved to individual memory work areas **102*d*** to which these fields are assigned: assembly-line #; tag; Comments field; F/C code; and, if a transfer instruction, the operand that contains the destination tag. Prior to moving the F/C code to its work area, the previous contents of that F/C code work area are transferred to a field of the primary storage area **102*e*** in memory, identified by the mnemonic LSTCD; this previous F/C code is used under a certain condition, as is explanied hereinafter. The next block **01.08** controls the reading of the next available source record into the input area **102*b***, which is now free to receive it since the previous record was moved to the work area **102*d***. The two memory areas **102*b*** and *d* each contains a record, which records are processed in the order they were received.

The next block, DECISION **01.08**, checks to see if the record in input area **102*b*** contains an identifier character EF indicating the end-of-a-file, which is commonly provided in all magnetic tape systems in one form or another. If the result of this test is that such an end-of-file identifier is in the record, the program branches to PROCESS block **04.08** (the last block of FIG. 8/4) which calls for the setting of an end-of-pass switch. After completion of the processing of the record that lies in the work area **102*d***, that switch is tested (block **04.07**) and if it is set, operations of block **06.07** are initiated for rewinding the input and output tapes **198** and **110** and calling in Pass II of the program; thereafter, by means of the JUMP instruction **06.08**, the main flow of the program is transferred to block **07.01**, the first block on FIG. 8/7, at the Pass II starts. However, if DECISION test **01.09** indicates that there is no end-of-file identifier in the record in input area **102*b***, the program steps to DECISION **01.10** to test if there is a valid F/C code in that field of the input record that sits in the input area **102*b*** of memory. If the answer is NO (that is, if it is an invalid character or a space), the program ignores that record and returns as a loop to PROCESS block **01.08**, and the next source record is read. This loop is repeated until the next record is found that contains a valid F/C code in the proper field of the input record. When it is found, the main flow of the program continues with DECISION **02.01** (FIG. 8/2) which determines if the F/C code of the record in the input area is a "C," which identifies a CONTINUATION record. If input area is a "C," the program branches to block **02.02** where the process is performed of appending the Comments field of that CONTINUATION record to the Comments field of the previous record lying in the memory work area. A CONTINUATION record serves only to carry continued lines of Comments of a previous record; it is not otherwise processed, and except for the Comments, it is not moved to the work area **102*d***. Thereafter, as indicated by JUMP **02.02**, the program then loops back again to block **01.08**, and the same process is performed on the next record. If it is another C-code, the Comments field of that input record is again appended to the Comments field of the previous record lying in memory work area **102*d***. This loop is repeated until another record is established in the memory input area which contains an F/C code other than "C" or space. At that itme, as indicated by NOTE **02.04**, memory work area **102*d*** contains all of the necessary information for processing the code record therein. The record in the input area **102*b*** is left there and it is processed after the record developed in the working area has been processed. A locator RHECOM is used to maintain the memory location of the right-hand-end (RHE) of the Comments work area, since this work area increases as additional CONTINUATION records are read and appended. This locator is a field of the primary storage area **102*e*** which contains the address in the work area of the RHE of the Comments. When the record in the work area is complete and ready to process, a control symbol is inserted into the memory location at the RHE of the Comments to mark the end of that record.

Blocks **02.05** and **02.06** begin the processing and they represent a number of processing operations that are performed to prepare for the output area **102*c*** to receive the record from the work area. The output record of Pass I has a certain format, which is set forth in Table I below. The same format is used for the output record of Pass II, and Table I indicates which Pass is used to fill each field. The beginning and end of each "line" of the Comments field is denoted by control symbols since the "lines" are of variable length, as explained below. A maximum size of the Comments field is arbitrarily set at about 15 lines of TEXT. The output memory area includes the following prescribed memory fields for receiving the characters that compose each record. The number of characters specified in the following table are those found suitable for particular embodiments; the number of characters may vary for other embodiments. Each output record assembled in the output area is written successively on work tape **110**, so that the latter contains all of the records in order at the end of Pass I; in turn, work tape **110** becomes the input to Pass II. Each output record is processed successively and completed during Pass II and then written out to the output file on tape **111**.

TABLE I.—OUTPUT RECORD

Field	No. CHAR	Filled in by—	Remarks
	1		Control Symbol.
No. Lines	3	Pass I	No. lines required on a F/C page to contain this record.
Column No	1	Pass II	Column No. on F/C page assigned to this symbol.
Line No	3	do	Line No. on F/C page at which symbol begins.
Page No	2	do	F/C page No. assigned to this symbol.
Box No	2	do	Box No. assigned to this symbol.
Code	1	Pass I	Type of symbol.
Tag	7	do	Blank if record has no tag.
"LHTC"	1	Pass II	Connection indicator for field "LHT".
"LHT"	6	Pass I	Contains destination tag on D, J, S codes—blank for other codes.
"RHTC"	1	Pass II	Connection indicator for field "RHT".
"RHT"	6	Pass I	Second destination tag on DECISIONS (may be blank).
SEQ No	4	do	Ascending Seq No.
Length	4	do	Length—lines of this record.
Comments	Variable	do	(Max. of 456 char.).
			Control Symbol.

As indicated by block **02.05**, the record-sequence-number counter is incremented by "1," and its new number is moved to the prescribed field of output area **102***c*. The processing continues with block **02.06**, and the F/C code is moved from the work area to the corresponding field of the output area. Also several memory fields are set to initial conditions by block **02.06**: Line Counter-A in primary storage **102***e* is set to 0; Comments Locator is set to the initial address (which is a constant) of the left-hand-end (LHE) of the Comments portion of the work area **102***d*, and Output Area Locator is set to the initial address of the LHE of the Comments portion of the output area **102***c*. All other output fields, other than the Comments field, are fixed and have predetermined addresses, and therefore, require no locators.

The main flow of the program steps to block **03.01** (as indicated by the pseudo-connector at the bottom of FIG. 8/2) where a test is made on the F/C code. If it is a B-code, the program branches to block **03.02** where construction of the Tag Table in primary storage is initiated for the first such B-code; and each succeeding B-code initiates a new section of the table for all tags following it. The entry in the Tag Table consists of insertion of the title of the chart, and upon printing out of the table of contents during Pass III this chart title is printed out as a heading for the tags associated with that chart. Thereafter, as indicated by JUMP **03.03**, the program jumps to block **03.05** where the editing operations are initiated.

If the test **03.01** determines that the F/C code is not a "B," the program steps along the main flow to DECISION **03.04** where a test is made for a tag in the record, which identifies the entry point of that record. If it does not have a tag the program steps to block **03.05** for the editing operations; if it does have a tag the program branches to block **05.01**, which controls the creation of a Tag Table entry. The Tag Table is begun by Pass I and completed by Pass II; it is used by Pass III and this remains as part of the primary storage area **102***e* of the

memory throughout the program. Each entry in the table consists of the following items:

TABLE II, TAG TABLE

(a) Tag
(b) Assembly line # of tag
(c) Sequence # of output record (from Pass I) containing this tag
(d) Page and box number assigned to tag
(e) Column and flow chart line number of tag
(f) Indicator to show cross-reference by other symbols, and page and box number of the first cross-reference symbol.

Item (b) of the Tag Table is used for information purposes in composing the table of contents during Pass III. Item (c) is used by Pass II in determining whether a DECSION branch can be processed and allocated to a secondary column (which is an adjacent column in the present embodiment). Item (d) is used for cross-reference connectors in Pass III. Item (e) is used by Pass III for drawing connecting lines on DECISION symbols. Item (f) is used by Pass III in placing "from-connectors" or cross-references on the charts. The entry of items (a), (b) and (c) is performed during Pass I; items (d), (e) and (f) are entered during Pass II.

Block **05.01** creates the Tag Table entry by inserting items (a), (b), and (c) of the record currently being processed. That is, it determines whether the record has a tag, and enters that tag as item (a), enters the assembly line # from the record in working storage as item (b); and enters the current reading of the sequence number counter as item (c). The sequence number counter was stepped forward by block **02.05** to establish the sequence number of the current record. Space is left for the additional items (d), (e) and (f) of the Tag Table to be added during Pass II, and a locator (TTLOC) of the RHE of the Tag Table is incremented appropriately so that the next Tag Table entry may be made at the proper location. In addition, a control field NUMTAG containing the number of items in the Tag Table is also incremented. The tags associated with C-codes do not have to be placed in the Tag Table and in fact it is not necessary to pick them up from the input area except where the preceding input record has a blank tag area. The operation of block **05.02** moves the tag field of the record in the work area also to the tag field of the output record. Thereafter, JUMP **05.03** transfers the program to block **03.05** for the editing process. It is seen from the logic flow after the test **03.01** for the B-code that this tag routine is bypassed for such B-code records, so that their tag fields are not examined either for the purpose of entry in the Tag Table or for transfer to the output record. The T-code records are similarly tested, and the logic flow therefrom also bypasses the Tag Table routine. Since B- and T-codes do not generate a flow chart symbol, any tags that they may have are not referenced on the flow chart.

After the tag processing has been completed or bypassed as required, three subroutines (EDLIN, PAREN, CHENT) are performed on the different types of records in the various ways described in detail below. These subroutines are represented in simplified form in FIG. 8. At block **03.05** the editing routine (EDLIN) of editing successive lines of the Comment field of the current record is entered. The operations that are performed are those of editing the Comments field into intelligible lines and moving them into the output field as summarized generally by the comment of block **03.05**. This operation varies with each of the code symbols and is described in detail below. As the editing is performed, the number of lines of flow chart required by each symbol is measured by block **03.06**; this number is fixed for the fixed format symbols such as D, S, W, E, H and J; the number varies for the P and N symbols as well as for the lines of text identified by the T-code.

The number of lines for the current symbol (from Line Counter A) is added to the running total (in Line Counter B) for the current chain of logic flow by block **04.01**. A chain is defined as all coding between successive J- or E-codes, and includes the first group of coding so terminated. Examples of chains are marked off by slant bars in the following sequences of codes; /BTCPCCNJ/ PSCPHPDCJ/PPCDE/.

Thus, block **04.01** produces a running total of the lines within any chain by means of Line Counter -B, which is cleared at the end of the chain, as noted below. The details of the EDLIN subroutine for the individual codes are discussed below.

A test is made for D-, W-, S- and J-codes by block **04.02**; and, if the current record contains such a code, a branch from the main flow is taken to a subroutine PAREN starting at block **05.04** and continuing through **06.02**. This subroutine is used to identify and extract from the Comments field, the destination tags and labels, if any, that are parenthetically included therein by asterisks, or to extract the destination tags from the input operands, which operations are described below for individual codes. After the PAREN routine or if the test of block **04.02** is negative, the program jumps to text **04.03**, which determines whether the current record contains a J- or E-code. If the answer to the test is "YES," the program branches to a subroutine CHENT of blocks **06.04** and **06.05**, which controls the construction of the Chain Table and the entry of new items in that table. The chain table contains the following three items:

TABLE III, CHAIN TABLE

(a) Sequence number of the Pass I output record at which the chain ends.
(b) Number of lines on a flow chart page required for the chain.
(c) Field for an indicator character to show that the chain has been processed during Pass II.

From the definition of a chain, every record in the output record file is necessarily part of some chain, which is identified in the Chain Table by the sequence number of the last record in that chain. The Chain Table entries have sequence numbers in ascending order, since they are created sequentially in Pass I as the records themselves are processed, and the records are numbered in ascending order. Block **06.05** transfers the current value of the record sequence counter to the Chain Table to establish the Chain Table identification of the current chain then ending, and also transfer the current value of Line Counter B, which is a direct measure of the number of lines in a column required on the flow chart page for the chain. Line Counter-B is cleared to "0," so that it is in condition to accumulate the lines for the next chain. This completes the operation of subroutine CHENT, and it exits back to block **04.04**. The record in the output memory area is not complete and a record terminating symbol is added to the RHE of the record, which is then written to the output file of work tape **110**.

At this point, the aforementioned end-pass switch is tested (NOTE **04.05** and DECISION **04.06**) to determine whether it is the end of a pass. As explained above, this switch would have been set by block **04.08** if the following record (now in the input area) contained an end-of-file (EF) identifier indicating that the input tape had been completely processed. By testing the switch at the end of each cycle, it is determined whether there are no further records in the input area to be processed, and whether the input tape and output work tape are to be rewound and Pass II called in. When an input record in the input area contains an EF identifier, the switch is set, the last record lying in the work area is processed, and then the switch is tested, which leads to the termination of Pass I. If the end-pass switch is not set, the program jumps, via block **04.07** to block **01.07** to move the fields of the record that is then in the input area **102*b*** to the work area **192*d***, which starts another cycle of processing. Block **01.08** reads the next source record into the input area, and it is tested for an end-of-file identifier (block **01.09**), a valid F/C code (block **01.10**) and a C-code (block **02.01**), all in the manner described above. If the new input record has a C-code, its Comment field is appended (block **02.02**) to that of the record in the work area, and the loop repeated until a record with a F/C code that is not a "C" is in the input area. Thereafter, the record in the work area is processed in the manner described, starting with block **02.05** and as indicated by NOTE **02.04**.

Processing of Individual Codes—Pass-I

The flow chart of FIG. 8 has been simplified and omits a number of detailed tests and paths of coding that are sufficiently outlined hereinafter for an understanding of the invention. Many of the codes do not enter the EDLIN subroutine (block **03.05**) since the Comments fields of such codes are not displayed (e.g. J- and E-codes). In addition, the PAREN subroutine is different for D- and J-codes, which differences are noted below in the detailed discussion thereof.

As noted above, each code is processed in detail by an individual set of instructions to perform functions peculiar to that code; two functions that all codes require are: (1) determine the setting of Line Counter A (the number of lines required by that particular symbol on a flow chart), which is inserted in the output record as "# lines" (see Table I), and (2) adding that number to the running total of Line Counter B to get the total number of lines required for the current chain, which is entered in the Chain Table. For the purpose of processing the individual codes, separate legs or paths of coding are provided, with each leg being a branch from an individual comparison test of the current code against one of a set of constants that respectively represent the codes that are employed.

J-code

The test **04.02** initiates the PAREN subroutine **05.04**, which is used to extract the destination tag from the asterisk field of the Comments. Block **05.04** determines whether the Comments field contains an asterisk field (located at the beginning of the Comments field) and if it does not, the program branches to block **06.01**. The later controls the picking up of the destination tag from the operand field of the input record which currently sits in the work area; the operation code of the input record is examined and the appropriate operand address is picked up. If the Comments do contain an asterisk field, block **05.05** develops a subroutine for extracting the destination tag from that field. If there is a label associated with that tag, it is ignored in the processing of J-codes, since labels are handled only in connection with D-codes. From whichever source the destination tag is obtained, block **05.06** moves it to field LHT of the output memory area, and the program jumps to a test **04.03** for the J-code, which leads to a branch to the CHENT subroutine **06.04** to create the Chain Table entry. The editing operation of block **03.05** is bypassed for J-codes (the Comments field is ignored completely for the processing of J-codes except for the asterisk-field search). Block **03.06** sets Line Counter-A to 10 lines (which is suitable for the fixed format symbol plus an extra space left between the Jump symbol and the next chain in the column); block **04.01** similarly increments Line Counter-B, which provides the information needed for the Chain Table entry to be made at blocks **06.04** and **06.05**. The record is then complete and can be written to the output tape **110** via block **04.04**.

H-code

Line Counter A is set to 10 lines covering the fixed format of this symbol, and Line Counter B i; incremented by the value of Line Counter A; the editing subroutine is bypassed. Thereafter, block **04.04** writes the record to the output tape. The H-code may be considered to be the end of a chain, if desired; and the Chain Table subroutine CHENT would be entered accordingly. However, preferably it is not so considered, since the computer operator may thereafter push a start button and the program would pick up the next line in the flow of coding. Accordingly, by not treating the H-code as the end of a chain, the next intended line of coding is naturally followed in the flow chart. The record is written to the output tape without the Comments field.

E-code

This code is processed in the same fashion as the H-code, except that it is considered the end of a chain and the Chain Table subroutine CHENT is entered and followed in the manner described above for J-codes.

P-code

Initially, the editing subroutine EDLIN represented by block **03.01** is entered to cary out the P-code editing. The coding supplies a constant for the length of line that is to be moved from the Comments field; for the P-code, this length is arbitrarily set for 19 characters as a maximum. An "intelligible" line is one that does not exceed the maximum and does not break up a word; hence, it may and usually will be somewhat smaller than the maximum length. EDLIN operates by finding the start of the Comments field which is supplied by the Comments Locator and then counting successive characters of the Comments field until that character is located which would mark the last character for the maximum length of the prescribed line. If this marked character is a space, then the intelligible line is exactly the length desired. If this character is not a space the subroutine steps back to the left until it does find a space, which signifies a word break; thus, the character immediately to the left of the space is the RHE of the line to be moved to the output Comments area. After the line is moved to the output area a control symbol is placed to its right as a line delimiter. The Output Area Locator is adjusted to the new RHE of the Comments field in the output area, and the Comments Locator for the work area is advanced to the first character of the next line to be moved out. A test is made before each line operation is performed to determine if there is an indicator for the end of the Comments field, which was placed there when the Comments were moved into the work area. If it is not, the logic recycles back to pick up a new line until eventually the end of the Comments area is found and the editing operation terminates. Line Counter A is stepped for each edited line of Comments that is moved to the output area; in addition, a count of 5 lines is added to allow for the fixed format of top and bottom symbol lines and a 3-line vertical connection to the next box. Line Counter B is similarly advanced.

A special case arises where the line length that is desired is too small to pick up even a single word; for example, where the word in the work area is 20 characters, and for a P-code the line length desired is only 19 characters. If this occurs, an artificial "word" is made by forcing a space into the twentieth character which results in a 19 character word and line, and permits the EDLIN loop to operate.

The P-code does not enter the PAREN subroutine; after the editing operation is complete the record is written to the output tape.

N-code

This code is handled in the same fashion as the P-code, except that each intelligible line is 15 characters long. Pass III arranges for the offset position of this symbol.

B-code

This code is handled without editing by EDLIN, and the Comments field is picked up in its entirety and moved to the corresponding output record area. The Line Counter A is set to zero, since the space for the chart title is allocated for each page, and it does not vary from page to page.

S-code

The PAREN subroutine **05.04** is entered from test **04.02** to place the destination tag in the output field LHT. If there is no asterisk field, the A-Address in the operand work area is used instead via block **06.01**. If asterisks were present, after the contents were moved via blocks **05.05** and **05.06**, the Comments Locator ends up pointing to the first character after the right-hand asterisk. If this character is a space, it is deleted by advancing the Locator to the right; and the Locator is so advanced until the first non-space character is found. A subroutine to perform this operation is also used for the D-code editing. Thereafter, the subroutine EDLIN (as described above for the P-code) is entered 5 separate times to move the Comments to the output area; the fixed odd-shaped format of the SUBROUTINE symbol (see, for example, block **13.02**) allows for 5 lines that are respectively 14, 15, 15, 15 and 14 characters long. After each call for a line, a test is made of an indicator (set by the subroutine itself) to see if all the Comments have been processed, and when the indicator is set the remaining calls are by-passed. No calls are made after the fifth one, so that any remaining Comments are dropped.

In processing S-codes, supplementary editing operations take place when there are only one or two lines to go in the symbol. Since the block is of a fixed column length and line processing is from top to bottom, Comments of one or two lines are moved down in order to center them within the box. Thus, if an end-of-Comments-area indicator is set after the first call on EDLIN (which indicates a single line), a subroutine arranges to shift this single line two positions to the right in the output Comments area, and each of the vacated spaces has an end-of-line symbol place therein. This anticipates the printing-layout operations of Pass III, which automatically moves the single line to be printed to the middle line of the box, so that it is centered in the block without additional coding. If the second EDLIN call produces an end-of-area indicator, the second line is shifted two lines to the right, the first line is shifted one line to the right, and end-of-line symbols are inserted between the two lines and in front of the first line, which have the effect in Pass III of moving the two desired lines into the second and fourth lines of the symbol to provide neat centering.

In the S-code processing, the Line Counter A is advanced 10 lines because of the fixed symbol format, Line Counter–B is correspondingly advanced and the record is written to the output tape.

D-code

The PAREN subroutine is entered via asterisk field test **05.04**. If there is an asterisk field, the subroutine is performed twice for the D-code, since two asterisk fields may be provided for the two lateral branches from a DECISION symbol. If only one asterisk field is present, the second entry to the subroutine has no effect. The destination tag extracted by the first call on PAREN is put in field LHT in the output area; and the destination tag extracted by the second call is placed in field RHT. If the second call produces no tag, the field RHT is cleared. If the first call on this subroutine produces no tag, then there is an input error, both LHT and RHT are cleared, and the DECISION symbol is printed out on the flow chart during Pass III with no lateral branches being indicated. The labels for the main flow branch and a single lateral branch from a DECISION symbol may be supplied by a special code that supplements the flow chart code, as noted above. Alternatively, these labels may be provided by means of the asterisk field in accordance with the following format:

LABEL*TAG,LABEL*TAG,LABEL*

The first label is the main-flow branch, the second label corresponds to the associated tag placed in LHT, and the third label corresponds to the associated tag placed in RHT. These three labels are extracted via block **05.05** and stored via block **05.06** in the first 15 characters of the output Comments area in the order given. Labels of more than 5 characters are truncated by the subroutine, and if a label is missing its corresponding output field is cleared to spaces. If there is no asterisk field, the destination tags are picked up via block **06.01** from the A- and B-Addresses of the input operands, and the labels are picked up via block **06.02** from a stored table of contents as determined by the special label code.

A subroutine is used which shifts the Comments Locator to the right, if necessary, to bypass any spaces between the end of the asterisk field and the first non-space character of the actual comment, in a manner similar to that described above for the S-code operation. The Output Area Locator is advanced 15 characters so that the first actual Comments line is laid down after the labels. Thus, on a DECISION symbol, the actual start of the Comments in the output area is the sixteenth character of the field.

A test is then made to see if the total number of characters in the Comments work area is 13 or less; 13 corresponds to the room available along the middle line of the DECISION box, whose size is arbitrarily set to permit six Comment lines of 7, 11, 13, 11, 7 and 3 characters, respectively. If the total number of Comments characters is 13 or less, two end-of-line symbols are inserted into the output area (representing blanks for the first two lines), an EDLIN call of 13 characters is made and all remaining calls on this subroutine are bypassed. If the Comments work area contains more than 13 characters, then successive EDLIN calls of 7, 11, 13, 11, 7 and 3 characters, respectively, are made. The logic operation is similar to that described above for the S-code. After each call, the end-of-area indicator is tested, and if set, all remaining calls are bypassed. If EDLIN indicates an intelligible line is impossible, a word is forced, as explained above, by inserting a space in the last character that fits. Due to the earlier test for a total of 13 characters, it is not possible for the first call of 7 characters to produce an end-of-area setting; however, if the second call produces such a setting, both lines are shifted on to the right in the output area, and an end-of-line symbol is inserted in front of the first line to produce a more attractive line spacing, as explained above in connection with the S-code. If the third call of 13 characters produces an end-of-area setting, no shifting is performed since otherwise the 13-character line that has already been moved might not fit into the smaller available space in a line below it. If the end-of-area setting is not reached after the final, sixth call, the Comments work area is simply truncated; if a large Comment is desired by the programmer, a NOTE can be used for that purpose.

After the EDLIN subroutine is complete, Line Counter A is advanced 13 lines corresponding to the fixed format used for the DECISION symbol. Line Counter B is similarly advanced, and the output area is written onto the output tape.

W-code

This code is initially routed down the path followed for the D-code to process the asterisks field in the associated PAREN subroutine. Immediately thereafter, a test is made to determine if it is a W or D-code, and if the former, it is processed down its own branch leg of logic. The remainder of the Comments field of the W-code is ignored, and instead, a constant is moved into th output Comments area beginning with the sixteenth character, and the code of the output record is changed to a D-code, so that it may be processed in that fashion from then on. The constant which is moved into the Comments area consists of the following: an end-of-line symbol, 11 minus signs, another end-of-line symbol, the word "SWITCH," another end-of-line symbol, 11 more minus signs, and two more end-of-line symbols to form the fixed format symbol shown as block **04.06**. Since the code is now changed to D, Passes II and III treat it as a D-code and the constant in the Comments field is so arranged that it is printed out as the desired SWITCH symbol. The Line Counters A and B are handled in the same fashion as in the D-code described above, and the output area is written out to the work tape.

T-code

In order to avoid ambiguity in the flow chart, the TEXT of T-codes is printed out without a special symbol only at the start of a chain of flow, for otherwise the T format would interfere with the appearance of the chart. Thus, if the T-code occurs other than at the start of a chain, it is converted to an N-code and processed as such in the manner described above to be printed out in a NOTE symbol. This is done by an initial test to see if the code of the previous record is a J, E, or B; this previous record code was saved by moving it into the memory filled area field LSTCD prior to transferring the current record from input to work area. If the T-code does come in at the start of the chain, it is processed by making successive calls on EDLIN until the end-of-area indicvator is set; the line requested is 30 characters long. Line Counter A is incremented for each such line, and after the last line, it is incremented by "3" to provide a space between the final line and the first part of the next F/C symbol to be printed in the column.

Summary of Pass–I

This first Pass examines successive records of the input program to be documented and extracts all the information needed to produce an F/C symbol from each record that contains an F/C code. Where the Comments field for an F/C symbol is greater than that carried by one record, the Comments fields of succeeding records (denoted by C-codes) are tacked to the previous record. A sequence number counter is incremented for each record and used to identify that record in the subsequent processing. Any tag that identifies the entry point of each input record is extracted and used to construct the skeleton of the Tag Table. The destination tages carried by records for the JUMP, SUBROUTINE, DECISION and SWITCH records are extracted as well as branching labels for the latter two symbols. Editing operations of the Comments fields of P-, N-, S-, D-, and T-codes are performed, and the number of lines along a column is determined for each symbol. In some cases, the symbols are a fixed format, and in other cases they are variable in foramt and their column length is determined by the number of Comment lines that have to be printed out.

Each unconditional transfer, i.e. J- and E-code, defines the end of a "chain" of coding, and a Chain Table is constructed that has an entry for each chain; each chain is formed as a sequence of symbols terminating with a JUMP or EXIT symbol and represents a section of program logic that is referenced on the flow chart to one or more other sections and that can be treated as a separable entity of logic for display on the flow chart. The Chain Table entries are identifiable by the sequence number of the last symbol of each chain and include the column length required for recording each chain on a page of the flow chart. The flow chart codes for HALT, SUBROUTINE, DECISION and SWITCH are not treated as chain terminating symbols, but rather as parts of a chain. The HALT symbol in some respects is like an unconditional transfer symbol in terminating a section of logic flow; however, it may be followed, as the program is performed, by the operator of the computer restarting the program, which would lead to the next record of the original program sequence being the one to be processed. Consequently, the logic flow from the HALT symbol, in effect, is an entry to the next symbol in the original sequence, and these symbols are preferably considered as part of the same section of logic and not separated on the flow chart. The SUBROUTINE symbol refers to a sub-section of logic which continues the main flow processing and does not terminate it, though the details of it are ordinarily separately reviewed and are therefore best left to a sepa- SWITCH symbols, each also have a branch that continues the main flow processing and are therefore incorporated as parts of the chain and not as branches from it.

PASS II

The flow chart for Pass II is shown in summary in FIGS. 8/7 to 8/18. As indicated in the text at the beginning of FIG. 8/7, the function of this pass is to complete the construction of the Tag Table and to assign all flow chart symbols to certain positions on the F/C pages. The input records for Pass II are supplied by the Pass I output work tape **110** and the output of Pass II is written on the second work tape **111**.

The sequential operation of Pass II is determined overall by the sequence of chains in the Chain Table, and certain chains are processed out of that sequence. Within any chain, successive records are generally processed in sequence as received, except when DECISION records are encountered; at that point the processing of the chain containing that DECISION is interrupted and the branch chains are investigated and (in the present embodiment) processed. As indicated in block **07.01**, a Chain Table Locator in the primary storage area is employed, which always points to (carries the address of) the Chain Table entry which is currently being processed along a main flow column. As the processing of each chain is completed, this Locator is advanced to the next chain in sequence.

Other initial operations performed in Pass II, block **07.02**, include that of setting a page-number (Page #) counter and a column (Col #) counter to "1," a box-number (Box #) counter to "0," and a Line Counter 1 (LNC–1) to "1."

The main loop of Pass–II then begins at block **07.03** with the first input record from work tape **110** being read into the memory area **102***b* (an additional memory work area is not required in this pass). A test is made, block **07.04**, for an end-of-file indicator, and if it is found, the program branches to **09.06** which controls the rewinding of the tapes **110** and **111** and the initiation of Pass–III. If it is not the end of the file, the sequence number of the current record is stored in field TPOS for later use; the program steps to a test **07.05** for a B-code, and if it is found, the program branches to block **12.09** where the end of a page is forced by setting Col # to "4" (assuming a 4-column page). Then the program transfers to an end-of-column subroutine ENDCOL, where a pseudo-connector record is developed. Since a B-code involves the development of a new chart title, a new page is normally started, which is the function of the ENDCOL subroutine. However, as explained below, in certain situations (as where it is the first B record of the chart) is is not necessary to start a new page and thereby needlessly skip a page; and the ENDCOL subroutine is essentially bypassed to block **08.06**, which writes the first B-code record to the output file. The program is then recycled via the test **08.07** back to block **07.03** to read the next input record from work tape **110**.

The next record which is not a B-code is processed via block **07.06** where the current value of LNC–1 (which is "1," for the first record of a page) is established as the Line # of the record being processed by moving it to LNC–3. As indicated in NOTE **07.07**, the Line # of a record is allocated as the Line # of the corresponding symbol to be printed on the F/C page and thereby fixes the position of the symbol in a column. Thus, in the example of the first record after the chart title, which may be text or some symbol, the Line # is set at 1. Block **07.08** extracts "# Lines" (i.e., the length) of the current record therefrom (see Table 1) and adds it to LNC–**1**. Thus, the previous LNC–**1** represents the ending line-number-plus-one of the previous symbol, and the new LNC–**1** becomes the starting line number of the following symbol; the current LNC–**3** is the beginning line number of the current symbol and is used as a temporary store of that number before it is moved to the output record.

The program steps to block **08.01** to compare the new value of LNC–**1** to an end-of-page constant EPCON, where EPCON is the total number of lines allowed in a column, that is the address of the last line (which is set to 106 for the high-speed printer page, and would vary for different types of recorder and display devices). If the end of a column has been reached, the program branches to block **11.01** for the ENDCOL subroutine; but if the new LNC–**1** is less than EPCON, there is room in the current column for the current symbol and the program steps to block **08.02**. Box # is incremented by **1**, and the Page #, Box # and Col # are moved to the output record in the output memory area.

Thereafter, test **08.03** determines if the input tag field of the current record contains a tag or is blank. If it contains a tag, the program branches to a block **09.08** to locate a Tag Table entry for that tag. The input parameter for this operation is a locator which points to the left-hand character of the tag field for which a search is desired. If this field contains a tag, the Tag Table is then searched by a straight series of compares beginning at the start of the table and running down until the tag entry is found. The starting address TBIN of the Tag Table was set during Pass I, and a counter NUMTAG was established during Pass I containing the number of entries in the Tag Table. Thus, NUMTAG tells the routine when it has exhausted the table as it makes its series of compares; upon exhaustion of the table if the Tag Table does not contain the desired tag an indicator is set to show this. The main output of this operation is the setting of a locator to the address of the left-hand character of the proper entry in the Tag Table; a subsidiary output is the setting of indicators to reflect "tag found," "no tag in field," or "Tag-Table entry missing." Block **09.09** places the page location data (Page #, Box #, Line # and Col #) of the current record in the Tag Table entry for the associated tag; this completes the basic structure of the Tag Table entry, and the supplementary data of cross-references to "from connectors" is subsequently added as explained below.

Modified logic is employed for T-code records; that is, Box # is not incremented and the Tag Table search is bypassed, because T-codes do not produce a symbol on the F/C page and, thus, do not carry box numbers or tags.

The Tag Table receives any "from connector" information in the current record; and this operation starts with a transfer back to a test **08.04** for a D- or J-code. If it is found, the program branches to block **10.01** where a cross-reference subroutine is performed in order to place Box # of the current record as a cross-reference in the Tag Table to any destination tags contained by the current record. Thus, block **10.01** obtains the destination tags (LHT and RHT) from the current record and looks those tags up in the Tag Table (in a manner similar to the above described subroutine of block **09.08**). A test **10.02** determines the results of the search; if no tags are found in the table the program returns to the next operation **08.05** of the main flow; but if the tags are found a test **10.03** determines whether this is the first reference to the tag. If it is, block **10.06** puts Page # and Box # of the current record in the cross-reference field of that Tag Table entry, and the program returns to the next main flow block **08.05**. If it is not the first reference, the program branches to block **10.04** which sets a signal in the Tag Table entry to indicate that there is more than one such cross-reference (which signal determines that an asterisk is to be printed at that entry point; see for example, the branch input to block **07.03**); the program returns to the next main path block **08.05**, to which it would pass if test **08.04** had proved negative. The cross-reference subroutine of **10.01** is entered twice for D-codes, since two such tags may be carried by such a record.

Upon return to the main logic path of the program the test **08.05** determines if the current record is a D-code. If so, the logic branches to block **13.01** for processing the DECISION record; if it is not a D-code, block **08.06** writes the record in the output memory area to the output tape **111**.

Thereafter, block **08.07** tests to determine if there is a J- or E-code in the current record, which codes indicate the end of a chain. If it is not the end of a chain, the program recycles back to block **07.03** to read in the next input record and repeat the processing loop described above. If it is the end of a chain, as indicated by NOTE **08.08**, the program proceeds to locate the next chain to be processed via test **09.01**, which determines whether the current record is being allocated to a main column or to a branch or secondary column. If the latter, the program branches to block **15.07**, which is described below; if the former, block **09.02** advances the Chain Table Locator to the next table entry, and test **09.03** determines from the indicator of that entry whether this chain has yet been processed. If so, the program loop continues until the test **09.03** finds an unprocessed chain. This loop is an important part of the processing system since by the very nature of the processing of DECISION branches, as explained below, it is possible that the next chain in sequence may have already been processed and assigned to a secondary column by the DECISION branch logic.

When the next unprocessed chain is found, its sequence number is used to locate the corresponding data record on the input work tape **110** via block **09.04**. It should be noted that the Chain Table entry for any chain contains the final sequence number of that chain. Thus, if the Chain Table Locator points to the chain that it is desired to process, the sequence number of the first record of that chain corresponds to the final sequence number of the previous chain plus 1. The tape position sequence field TPOS contains the sequence number of the last record that has been read; therefore by subtracting TPOS from the Chain Table entry sequence number, the difference corresponds to the number of input records that have to be skipped to get the desired record. Ordinarily, the next chain to be processed is the next physical chain on the tape; which is indicated if the sequence number of the Chain Table entry is the same as TPOS, and no further records have to be skipped. Thus, block **09.04** computes the difference between TPOS and the sequence number of the first record of the desired chain; it then proceeds to skip that number of records on the input tape **110** so as to position the tape at the first record of the desired chain. After the input tape **110** is so positioned a test is made to determine if there are 20 or more lines left in the current column being allocated. This determination is made by subtracting LNC–1 from EPCON (the total column length) and comparing the result with the constant 20. This test is of assistance in insuring good page format in that a new chain is not initiated near the bottom of the page instead the remainder of the column is left blank and the new chain is started at the top of the next column. If this test shows that there are less than 20 lines left in the column, then an end-of-a-column subroutine similar to ENDCOL is entered to determine which column is currently being processed and thereby begin a new column, before reading the first record of the new chain. Whichever direction is taken by the last mentioned test, the program recycles back to the start of the main processing path at **07.03** to read in the first record of the new chain and perform its processing.

ENDCOL Subroutine

If the aforementioned test **08.01** indicates that the end of a column has been reached, the program branches to perform a further test **11.01** to determine whether the current record is a J- or E-code. If it is such a code it can nevertheless be allocated to the current column since EPCON has its value chosen so that there is enough room at the end of every column to contain an additional 10 lines required for a J- or E-connector (or for a pseudo-connector). Consequently, if the test shows a J- or E-code, the logic is routed directly back to the main processing path at block **08.02** as if the test **08.01** against EPCON had gone the other way, since the rest of the processing beginning with block **08.02** can be properly performed on the current J- or E-record. The next record will then be directed by the test **08.01** into the ENDCOL subroutine to start a new column.

When the code is not a J- or E-, then the ENDCOL subroutine is entered at block **11.03** to arrange for the proper termination of the current column by the generation of a pseudo-connector record in the output file, and by the proper initiation of the new column which involves resetting the various indicators and locators. A pseudo-connector record is a short record, 13 characters in length, which goes to the output file and is used by Pass III to create a connector symbol at the bottom of each column that is not terminated by a JUMP or EXIT symbol.

TABLE IV.—PSEUDO-CONNECTOR RECORD

Field	# CHAR	Field	# CHAR
Control symbol	1	Line #	3
Letter "X"	1	Page #	2
Spaces	2	Box #	2
Column #	1	Control symbol	1

The letter "X" identifies the record as a pseudo-connector. Column # and Line # (LNC–3) fix the location of the symbol on the page. Box # and Page # are those of the currently-processed record and are printed inside the connector symbol to indicate the next symbol in the path of flow. If it is a connector for the bottom of a main column (other than the last) of a page, the connector symbol contains the current Page # and current Box # plus 1; if it is a connector for the bottom of the last main column on a page, it contains the current Page # plus 1 and a Box # of "01."

A pseudo-connector is not needed when the previous record was a J-, H- or E-code record terminating a chain, since the J-, H- or E-symbol satisfactorily terminates the column. A test 11.02 for this condition is made by examining the contents of LFTCD, where the previous record's F/C code was saved. If the test is negative, block **11.03** proceeds to generate the desired pseudo-connector record in the fashion explained above and write it to the output tape. The program continues with block **11.04**; if the test **11.02** indicates that the previous record was a J-, H- or E-code, this pseudo-connector operation **11.03** is bypassed and the program proceeds directly with the test **11.04**. The latter tests the Col # counter to determine whether it is set to a value of "4," if so, the current column is the last column on a page, and the program branches to block **12.03**, where the Page # is increased by 1; then to block **12.04**, where the Col # counter is set to "1" and the Box # to "0." Then block **12.05** writes an end-page record to the output tape **111**, which record comprises a control symbol that Pass III uses to determine when it has read in all of the records needed to create a page. With the end-page symbol a complete page of records has been written to tape **111**, and the program then jumps to block **11.07** where the line counters are set to "1." Thereafter, a test **12.01** determines whether the currently processed record is a B-code, and if it is the program branches to

)8.06 where the record is written to the output file, and he program proceeds to process the next record via blocks)8.07 and 07.03. If it is not a B-code, the program jumps o 07.06. Jump 12.02 transfers the program to block 07.06 :o repeat the initial processing of the line number of the :urrent input record in view of the resetting of the line :ounters at the start of this new column. Thereafter, the program proceeds in the manner described above.

If the test 11.04 indicates that the current column being allocated is not the last one on the page, the next test 11.05 determines if any symbols have been allocated to the next adjacent column, which condition can occur upon processing of branch chains from DECISION symbols, as explained below. Thus, if the adjacent column has been already allocated, the program branches to block 12.07, where "2" is added to the Col # counter. This has the effect of skipping the adjacent column to obtain the next column thereafter for the current allocation. If this new column number is greater than "4," the program operation is via blocks 12.03 to 12.05 to start a new page as described above. In any case, the program transfers back to block 11.07 to reset the line counters and start the processing at the top of a new column.

If the test 11.05 determines that the adjacent column has not yet been allocated, block 11.06 adds "1" to the Col # counter. Block 11.07 resets the line counters, and the processing continues at the top of a new column, in the manner described above.

B-code Processing

As previously described a test 07.05 for the B-code is made shortly after each record is read. When such a code is found, a new page is started by setting the Col # field to "4" and then entering subsoutine ENDCOL at block 12.09. This subroutine, via test 11.02, blocks 11.04, 12.03 to 12.05, 12.01 and 08.06, sets up a new page and returns control directly to the main path at the point of writing the record to the output.

There are two special cases where the ENDCOL subroutine is preferably not entered. The first is where this is the first record on the input file, which is normally a B record. To avoid skipping a blank page, a test is made of the sequence number of the B record, and ENDCOL is bypassed if it is "1." The second case is where B-symbol by chance is the start of a new page; that is, where the current symbol location is at the start of a new page, and the previous page was already properly terminated. This condition can be tested for by testing Box # for "00"; which number indicates that the current operation is at the top of a new page and that ENDCOL can be bypassed. Normally, a B-code should be preceded by a J- or E-code, which would properly terminate the previous page. If by error that should not occur the B-code will nevertheless start a new page, and a pseudo-connector for the previous page is generated in block 11.03.

DECISION Branch Processing

The aforementioned test 08.05 for a D-code, if affirmative, directs an immediate branch to the associated processing logic at block 12.01, and there-preceding is TEXT that sets forth the function of this processing. The DECISION record and associated data, are analyzed to determine how best to illustrate its branches on the flow chart. The processing includes the secondary-column subroutine SCOL, block 16.07, which, as the TEXT there-preceding indicates, analyzes a chain of code branching from a DECISION symbol in order to determine if it is possible to assign that chain to a secondary-column. When it is determined that such an assignment to the secondary-column should be made, the DECISION-branch logic functions (in this embodiment) as a control routine for processing directly to process that chain in the secondary-column.

There are two key indicators in the input records which must be set for all DECISION records: These are indicators LHTC and RHTC, which are used by Pass-III to determine how the page is to be laid out. LHTC refers to the destination tag located in field LHT, and RHTC performs a similar function for field RHT. Each of these indicators can have three possible values that correspond to the following courses of action, respectively: (a) This branch requires a connector symbol; (b) The coding for this branch is in the adjacent secondary-column; (c) The adjacent column is not being used for this branch; it is not known whether a connector is to be used or a line can be drawn to show the path of flow.

Of the above three possibilities, courses-*a* and *b* are definite; course-*c* indicates that the final result is unknown and is to be finally revolved by Pass III. Pass I, when it sets up its output record, sets both LHTC and RHTC to indicate course-*c*. Pass II may change them to course-*a* or *b*, or leave them set at course-*c*. Under certain conditions, subroutine SCOL will set the two indicators; under other conditions the DECISION-branch logic itself does the setting.

It should also be noted that subroutine SCOL determines the suitability of a chain for display in the adjacent secondary-column, and the final determination of so displaying that chain is made by the DECISION-branch logic itself. For a single branch DECISION, field LHT is used. For a two-branch DECISION, both LHT and RHT are used. The subroutine SCOL generates a "no good" signal if the testing field has no tag (thus, on a one-branch DECISION, the field RHT has no tag, and SCOL generates "no good").

The DECISION-branch logic is divided essentially into two sections. The first section starts at block 13.01 and determines (by means of SCOL) if either branch chain is suitable. The second second section (beginning at block 14.03) is a control routine which sets up a branch chain to be processed and starts the processing at block 07.03; after the branch chain is processed, the program control picks up processing of the main column of flow again from the point it was temporarily halted to handle the branch chain. In the event that no chain is allocated to the secondary-column, the second section of the DECISION-branch logic is not entered.

The first test 13.01 determines whether the current processing is taking place in a secondary column by checking an indicator EVOD which assumes one value for processing main columns of flow and another for secondary or branch columns. If EVOD indicates "secondary column," no further branching to subordinate columns takes place from the secondary-column (in this embodiment); LHTC and RHTC remain unchanged (Pass III determines whether to draw a connecting line or a connector symbol at the branch point), and the program branches back to block 08.06 to continue with the processing of the branch chain in that column. If EVOD indicates the current processing is in the main column, the program steps to the SCOL subroutine 13.02, and the latter symbol indicates that SCOL starts at block 16.07.

SCOL subroutine

The SCOL logic begins, block 16.07, with setting the exit from the subroutine back to the main-flow reentry point. This subroutine is entered from block 13.02, 13.04 or 15.01, and the reentry point in each case is the main-flow block immediately thereafter; namely, block 13.03, 13.05 or 15.02, respectively. The input parameter for this subroutine consists of a locator pointing to the left-hand character of the tag field in question (LHT for the first-branch analysis and RHT for the second). This locator also fixes the location of LHTC or RHTC, as the case may be, since the latter indicators are located one character to the left of their associated tag fields. The output of the subroutine is a signal stating whether or not the chain involved can be put in the adjacent secondary-column, namely "ok" or "no good," respectively.

Block 16.08 looks up the destination tag of the first branch LHT for an entry in the Tag Table (in the manner similar to the operation of block **09.08** described above). Test **16.09** checks if the record's tag field contained a tag, and if the tag could be found in the Tag Table; if either condition is negative, block **17.02** sets the indicator LHTC for a "connector" symbol, and the subroutine jumps to EXIT **18.08**.

If a tag and its corresponding Tag-Table entry are located, SCOL determines whether the chain identified by the tag is suitable for assignment to the adjacent secondary-column. All six of the following criteria must be satisfied or a "no good" signal is established, and LHTC is set to "connector" by block **17.02**.

(a) The record identified by the destination tag must not have been assigned. Test **17.01** checks this criterion by determining whether the Box # field in the current Tag Table entry is a blank; if so then the associated record has not yet been processed and allocated (for block **09.09** makes the Tag Table entry upon such allocation). If the Box # field is completed, then the destination has been allocated a place on the chart (this of course is the case for all DECISIONS which jump to an earlier section of the coding).

(b) The destination record must not be in the current chain. To check this criterion, block **17.04** extracts from the Tag Table (Table II) the Sequence # of the destination tag and places it in field TSQ. The Chain Table Locator points to the Chain Table entry for the current main-column chain and thereby the final Sequence # of the current chain can be extracted and compared wth the field TSQ, the Sequence # of the destination tag. The TSQ must be greater than the current-chain Sequence # or else the destination tag is in the current chain (since it has already been determined that it is not in a prior chain by test **17.01**). Thus, the result of this comparison supplies the answer to the test **17.05** of this criterion.

(c) The destination record must be at the start of a chain, rather than at some other point thereof (for this embodiment of the invention). This operation is performed by locating (block **17.06**) the chain that contains TSQ; beginning at the start of the Chain Table (Table III), the successive Sequence # fields thereof are compared with TSQ until the Chain Table entry is higher than TSQ. The chain of the tag is thereby located. To obtain the start of that chain, block **18.01** extracts the Sequence # of the previous Chain Table entry (which is that of the last record of the previous chain), and "1" is added to it to obtain the Sequence # of the first record of the chain containing the tag; the latter result is compared with TSQ (in block **18.2**) to determine if they are equal. If so, the destination record is the start of a chain. Block **18.01** also picks up the length of the chain containing the tag for subsequent use in this subroutine. An extra test has been found desirable to determine if the destination record is the second one of the chain; and, if so, whether the first record is TEXT or NOTE. If it is, the second record is then considered as satisfying this criterion; and the chain can be printed in the secondary-column with NOTE or TEXT at the beginning thereof.

(d) The adjacent secondary-column must be free at this branch point of the DECISION symbol (for this embodiment). This criterion determines that the adjacent column at the branch point has not been previously assigned to a chain coming down from another DECISION symbol in the same main column, but located above the current DECISION symbol. Block **18.03** determines this condition by comparing LNC–3 against LNC–2. LNC–3 is the column Line # at which the current DECISION symbol begins; and LNC–2 is a field completed by the DECISION-branch logic when the secondary-column is allocated and it represents the Line # at which the last chain in the secondary-column ends. If LNC–2 is greater than LNC–3, the adjacent column is occupied at the branch point and therefore not free.

(e) There must be enough room in the adjacent column to contain the entire chain (for this embodiment). The test **18.04** for this criterion is performed by taking LNC–1 (the ending line for the current DECISION symbol), subtracting "6" to get the Line # of the branch point of the DECISION symbol (which, by an arbitrary rule, is where branch chains should start on the flow chart), adding the number of lines in the destination chain (obtained by block **18.01**) and comparing the results with the constant EPCON (the column line-length). If the latter is smaller, there is insufficient room in the secondary-column for the entire chain.

(f) The number of records to be skipped on the work tape **110** to reach the desired chain must not be excessive. This criterion is checked via block **18.05** by taking TSQ, subtracting "1," and then subtracting the field TPOS (the Sequence # of the record that was last read). The result gives the number of records to be skipped to reach the chain in question, and if test **18.06** determines that this number is excessive then it would be too time-consuming to pick up the chain. This number would vary depending upon the apparatus and individual choice as to efficiency. For example, a skip of 40 records or more has been considered excessive for some purposes.

If all of the above criteria (a) to (f) are satisfied, the "ok" indicator is set by block **18.07**. If the chain is "ok," no further action is taken by SCOL. The subroutine exits via block **18.08** and returns to the main flow reentry point of the DECISION-branch logic (e.g. to block **13.03** after analysis of the first branch chain). Failure of any of the above six criteria results in the "no good" signal being set by block **17.02**.

If a chain is found to be "ok" then the number of records to be bypassed (as computed by block **18.05**) to reach the desired chain is preserved in an appropriate memory field, since it will be used by the succeeding DECISION-branch logic.

DECISION-branch logic continued

The "ok" and "no good" signals from SCOL are used in the test **13.03**; and if "ok" the program steps to subroutine **13.04**, which directs another entry into the SCOL subroutine for the second branch tag RHT. Upon completion of the second tag analysis by SCOL, the subroutine exits back to **13.05** to test if the second tag can be assigned to the secondary-column. The different possible combinations are handled as follows: If the first branch could not be assigned, as tested at block **13.03**, the program branches to **15.01** for the second-branch operation of SCOL. If that second branch likewise could not be assigned (test **15.02**), then the DECISION-branch logic exits by pumping back to the main flow at block **08.06**, the current DECISION record is written to the output tape **111**, and main-column processing continues. However, if the second branch is tested to be "ok" in block **15.02**, the program branches to block **15.05** to set the status indicator for the second branch to the secondary-column. The program then jumps to block **14.02**, which writes the current DECISION record to the output file and steps to block **14.03** for processing the second-branch chain. However, if the first branch tests "ok" in block **13.03**, while the second branch tests "no good" in block **13.05**, the program jumps to block **14.01** to set the status indicator for the first branch to the secondary-column. Thereafter, block **14.02** writes the current DECISION to the output file, and block **14.03** initiates the processing of the first branch.

Where both the first and second branches test "ok," a decision is made (in this embodiment) to process one of the branches and mark the other one for a "connector"; a criterion found to be suitable is that of determining which branch is the closer one, as shown by test **13.06**. The relative closeness is readily determined from the number of records that have to be skipped to reach each chain, as computed in block **18.05** of the SCOL subroutine. If the first branch is closer, the status indicator for the second branch is set to "connector" in block **13.07** and the status indicator for the first branch to the secondary-column by block **14.01**. If the second branch is the closer one, the program branches to block **15.04** which sets the status indicator for the first branch to "connector," and block **15.05** sets that of the second branch to the secondary-column, and the program jumps to **14.02** for writing the current DECISION to output. Appropriate coding is provided to preserve the information derived during the SCOL analyses for LHT, so that it is not lost when SCOL is repeated for RHT.

When a chain is finally chosen for the adjacent secondary-column, its corresponding indicator LHTC or RHTC is set to indicate this; this indicator for the other chain is said to use a "connector." This is done even though there may be only one branch in the DECISION, since the setting of RHTC for a nonexistent tag in RHT is ignored by Pass III.

Secondary-column processing

When block **14.02** writes the current DECISION to the output file, the current value of TPOS (the Sequence # of that DECISION record) is stored for later reference and reentry to the main column. Block **14.03** starts the second section of the DECISION-branch logic and uses the calculation (block **18.05**) of the number of records to be skipped on the input tape to position the tape in front of the first record of the branch chain which is to be assigned to the secondary column. An indicator is then set (**14.04**) in the Chain-Table entry for that branch chain that the latter has been processed; this prevents further processing of that chain later on in the operation of Pass–II when it would normally be picked up in turn.

The column number is advanced (**14.05**) by "1," so that it points to the adjacent column next in order. The indicator EVOD is set (**14.06**) to indicate that a secondary-column is being filled; this indicator is needed, because the processing of the secondary-column chain is via the main processing logic; and it is tested upon completion thereof for return to the DECISION-branch logic.

LNC–1 (which indicates the bottom line number of the DECISION symbol from which the branch occurs) is stored (**14.07**) so that it may be subsequently picked up upon return from the secondary-column processing. The new LNC–1 for the secondary chain is obtained (**14.08**) by subtracting "6" from the previous LNC–1, which has the effect of starting that secondary chain 6 lines above the bottom of the DECISION symbol from which it branches, which is at the branch point of the diamond-shaped symbol. Control is then transferred (**14.10**) to the main processing path at the point **07.03** where it starts processing a new record, the first in the branch chain. As indicated by NOTE **14.09** the main processing path allocates assignments for the secondary-column in its normal fashion, since LNC–1 and Col # have been appropriately set. DECISION symbols that are encountered in the secondary chain tend to move the logic into the DECISION-branch coding via the test at **08.05**, but the program is immediately returned to the main processing path by the test at **13.01** which initiates the branch logic. Eventually, a J- or E-code is found by the main processing path via the test **08.07**; the next test **09.01** finds the EVOD indicator set to "secondary," and the program branches to block **15.07**, which is effective to rewind the input tape to the record following the DECISION symbol from which the secondary-column chain branched. Since the SCOL subroutine measured the secondary chain and found that it would fit in the adjacent column, when the first J- or E-code is reached, the secondary-column chain terminates and it is proper to return to the main-column processing as indicated in NOTE **16.01**. Block **16.02** transfers the then current value of LNC–1 to LNC–2, so that the latter represents the last line assigned to the secondary-column. LNC–2 may be required by SCOL in the event that another DECISION occurs further down in the main column. The value of LNC–1 of the main column DECISION symbol, which was stored (at block **14.07**) upon entry into the secondary-column processing, is returned to **LNC–1** so that the main column processing begins where it left off prior to processing the branch. Block **16.03** resets EVOD to "main," and block **16.04** reduces the column number to restore it to its original main-column value. The indicators are all restored and the input tape is then at its proper place to continue the processing (NOTE **16.05**) of the main-column chain that was interrupted for the secondary-column branch. The main-column DECISION record that initiated the branch operation has been completely processed so that the input tape is positioned to the succeeding record by block **15.07**. For this purpose the TPOS of the DECISION record, which was stored is now subtracted from the current value of TPOS, which is the sequence number of the last record of the secondary-column chain. The result is the number of records that the input tape **110** must be backspaced for proper repositioning. After repositioning of the input tape, control goes back to **07.03** to read the next input record of the main column and continue the processing of the chain that was interrupted.

The foregoing operations of Pass II are performed iteratively on all of the records of each chain in the manner described. Successive chains are processed in order until a DECISION record calls for a branch chain from the main flow. At that time, the branch chain is analyzed to determine if it is suitable for allocation adjacent to the main flow chain, and if so, it is processed. After the last record of the input file is processed, the end-file indicator is detected (**07.04**) the output tape **111** is rewound (**09.06**) and Pass III is called in for operation.

Summary of Pass II

The primary function of Pass II is the allocation of flow chart locations to the symbols. The input is the output from Pass I. As each record is read, it is assigned a Box # and a Line # in the current column (the associated portions of the Tag Table are thereby completed). When the column is filled, a new column is started, until the rightmost column is reached. After this column is filled, a new page is started. DECISION records involve considerably more complicated processing than other types of records. Whenever possible, the F/C program attempts to place the coding that branches off from a DECISION in the column immediately to the right of the DECISION symbol, which is termed an adjacent or secondary column. However, before this can be done, a number of conditions must be satisfied by the branch chain, including the following:

(1) The section of coding that branches from the DECISION must be further down the input tape; otherwise it would have been allocated at an earlier point.

(2) The entire chain must fit in the adjacent column without overflowing the bottom of the column, to avoid breaking up a chain. (In other embodiments, branch chains that fit in larger page sections than a column may be used.)

When all of the above conditions have been satisfied, a chain branching from a DECISION symbol is considered eligible for assignment to an adjacent column. All of the information necessary to test the above conditions is contained in the Tag and Chain Tables and the record itself. In the event of a three-way DECISION, where there are two branches to be considered, it is possible that both branches will be eligible for the adjacent column; in this case, the chain closer to the current record is chosen. Each of the two destination tags in the record has an associated indicator which is set by Pass II to one of the three possible values—"adjacent column," "connector," or "unknown." If a chain is selected for the adjacent column, its corresponding indicator is so marked in the DECISION record. Chains not eligible for the adjacent column may be designated either "unknown" or "connector," depending on which of several conditions was not met. With a destination indicator of "unknown," Pass III attempts to draw a connecting line to show the path of flow instead of using a connector.

After a branch chain has been selected for assignment to the adjacent column, the input tape is advanced to the first record of this chain. An indicator is then set in the Chain-Table entry for this chain to show it has been processed; this is necessary to avoid reprocessing this same chain at a later time in a main column. The column number is advanced by "1," appropriate indicators and counters are set, and the program logic is then routed back to the same coding that processes records for the main columns; thereby common coding is used for processing chains in both main and adjacent columns. The first J- or E-code encountered while in the adjacent-column mode indicates the end of the branch chain; at this point the input tape is rewound back to the original DECISION record, and main-column processing continues where it left off.

Pass II also makes the necessary "from" connector entries in the Tag-Table to allow handling of cross-references by Pass III. Only the first such reference to any tag is noted, along with a signal if there is more than one.

PASS III

The function of Pass III is to form an entire flow chart page in memory, draw the necessary connecting lines from DECISION symbols, and produce a finished flow chart. The input consists of the output from Pass II on tape **111** and the Tag-Table located in memory. The output either goes to an on-line printer or to a tape for off-line printing. Pass III also produces the Table of Contents at the beginning of the flow chart.

Page layout

Throughout Pass III, a section (e.g. 15,000 locations) of memory is reserved (in this embodiment) for holding an entire F/C page internally; for computers having limited memory capacity, storage tapes or drum may be used to supplement the memory. This page-layout memory area is structured as contiguous "lines" of 120 characters each. The first line of each chart is represented by the first 120 characters of this area, the second line by the next 120 characters, etc. The location of any symbol on a page is given by its Column # and Line #. A subroutine **19.06** is used to convert these two factors into a memory address that represents the centerpoint of the first line of that symbol; it operates by multiplying the Line # by 120 and adding to the result one of four factors depending on the Column #. An index register is reserved for use as a locator; this index register always contains this base location on the page that the program is currently concerned with, and hereinafter it is referred to as the "Page Locator."

To "move" this Locator around on the page, "120" is added to the Page Locator, which moves it to the same position on the next lower line; subtracting 120 moves it to the same position on the line above. Moving to the right or left on the same line is accomplished by adding or subtracting the appropriate number of positions from the Page Locator. Since the Page Locator is an index register, indexing techniques may be used in place of actually modifying the Locator.

Every symbol has fixed dimensions for which constants are stored; that is, the horizontal dimensions are fixed for all symbols, and the vertical dimensions are fixed for some and variable for others (e.g. P- and N-codes). For all symbols, the location of the Box #, tag, and other related information is always located in a fixed position relative to the center-line of the column. Detailed specifications of these dimensions will be apparent from the flow chart of FIG. 8, which illustrates suitable values and conventions that are followed in printing the flow chart.

GENERAL DESCRIPTION OF PROCESSING

The first function performed by Pass III is to print out the Table of Contents (block **19.01**). All information present in this table is obtained from the Tag Table, which is in memory throughout the entire program. Printing of the Table of Contents involves moving the necessary information from the table to the page-layout memory area; that is, for each chart a list of the tags and the page and box numbers therefor. A tally NUMTAG set by Pass I determines when processing of the Tag Table to Table of Contents is completed. An entire page is constructed prior to writing anything out and as many pages as needed to contain all the tags are produced. It has been found suitable to list the information in two columns or sections on a page, with the entire left column of the output memory area being filled before any entries are made in the right column thereof.

After the Table of Contents is completed, production of the flow chart pages begins. A full page is processed at a time; nothing is printed until the entire page has been formed in the page layout area, at which time the entire page is printed. A control symbol on the input tape **111** designates the end of a page; this control symbol is developed in Pass II. Each input record is processed separately, and a new record is not read until all processing for the previous record is completed.

Immediately upon reading a record by block **19.02**, tests **19.03**, **19.04** and **19.05** are made respectively for end-file and end-page indicators and B-code, as discussed below. Thereafter in the program, a subroutine **19.06** computes the chart location of the F/C symbol for the current record. This chart location is stored in the Page Locator and is obtained from the Col # and Line # of the symbol in the input record. After determining the location of the symbol, a check **20.01** is made for a cross-reference and the proper "from connector" is generated (block **20.02**) if required. The record is then routed down a particular path, depending upon its F/C code. There is a separate page for each code. Each path performs the necessary layout for the particular symbol involved, puts in any connecting lines, box numbers, and tags necessary, and upon completion returns back to read and process the next record.

Any Comments text associated with a symbol is moved from the input record to the page layout area by means of a Move-Line subroutine **20.08**. In an input record, each Comments line is delimited by a control symbol, and a second control symbol is used to indicated the last Comments line within the record. Each individual code path calls upon this subroutine when necessary in order to move the text from the record to the page area. The Move-Line subroutine picks up the next line of text from the input record and places it on the page centered about the memory position given by the Page Locator. Upon entry to the subroutine, therefore, the Page Locator must point to the center of the field where the line is to be placed. It should be noted that Comment lines, as they exist in the input record, are usually less than the horizontal dimensions of the symbol, and this subroutine centers them so that there are equal margins on the right and the left.

The Move line subroutine **20.08** also performs additional functions:

(1) After moving each line, it increases the Page Locator by 120 characters, thereby automatically setting the Locator in proper position for the next line within the symbol.

(2) Adjusts an input record pointer so that the next call on the subroutine will move the next sequential Comment line of the record.

(3) Tests to see when it has moved the last line of an input record to the page area, and sets a signal when this condition is encountered.

When an End-of-Page symbol is located in an input record, block **19.04**, the entire page is written to a print

:ape, block **21.05**, or printed directly to the on-line printer **112**. The page layout area is then cleared to spaces and formation of the next page begins with the reading of the next input record.

An End-of-file symbol on the input file signifies that all input data for this program has been processed (blocks **19.03**, **21.03**; at this point Pass IV is called in.

Detailed Processing

As outlined above, every record (except B-codes) has the location of its symbol computed by means of a subroutine **19.06**, which location is stored in the Page Locator. Separate paths are then taken for each F/C code:

B-code

A B-code is the only F/C code which is not allocated to a column; its sole function is to supply a chart title. When a B-code is encountered (blocks **19.05**, **21.07**), its Comments field is stored in the sub-header area of the page memory area, where it remains until overlaid by the next B-code. No further processing is needed for B-codes.

J-code

A subroutine **20.06** is used to create an octagon of dots on the page, with the vertical connecting line pointing to the midpoint of the top line thereof. The Box # and tag (if any) of the record are placed outside the symbol in the proper memory locations (block **20.07**). The Move-Line subroutine **20.08** is bypassed for there are no Comments in J-records, and a test **20.09** leads to a branch **22.03** that locates in the Tag Table the destination of the JUMP record. The Page # and Box # of this destination are taken from the table and placed inside the symbol. If the destination tag is not found in the Tag Table, indicating an undefined tag, the center of the symbol is left blank. This completes the processing, and the next record is brought in (**19.02**).

E-code

The subroutine **20.06** creates the symbol in the page area. The word "EXIT" is then placed within the symbol and the Box # of the symbol and tag (if any) placed alongside the symbol (block **20.07**).

H-code

A HALT symbol is generated (**20.06**) with the word "HALT" inside of it. Tag, if any, and Box # are then placed on the page (**20.07**).

P-code

The Page Locator is first backed up one line (120 characters) and the tag (if any) and Box # placed on the page (**20.07**). The Locator is then advanced back to the top line of the symbol, and the top symbol line, consisting of a field of minus signs, is placed (**20.06**) on the page. The Locator is then advanced to the next line, the letter "I" is generated, one on the extreme left and one on the extrem right, to form a part of the vertical boundary lines of the symbol, and a call is made on the Move-Line subroutine **20.08** to move the first line of Comments text to the page. On return from the subroutine, a test is made to see if this was the last line. If not the last line, the logic is recycled back to where the vertical boundary line segments "I" are inserted, and the next parts thereof are inserted, and another call is made on the Move-Line subroutine. This cycle continues until the signal indicating "last line" is set, when the vertical boundaries are complete, as is the Comments area. Then, the bottom line of the symbol, consisting of a field of minus signs, is placed on the page. It should be noted that it is not necessary to adjust the Locator to the next line, since this is a function performed by the Move-Line subroutine.

N-code

This code is processed identically to the P-code, with the following exceptions:

(a) Asterisks are used for both horizontal and vertical boundaries.

(b) Since a NOTE symbol has its left side offset by two positions to the right, the Locator is incremented by two, prior to doing any processing. (Incrementing the Locator by two positions sets it to the horizontal center of the offset NOTE symbol.)

S-code

The Page Locator is backed up to the previous line, and tag and Box # placed on the page. The entire SUBROUTINE symbol is then moved to the page layout area; this symbol is stored in memory as a constant and is moved to the page area from the constant area in a series of moves controlled by a tally (**20.06**). After each line of the constant is moved, the Locator is incremented by 120 positions to bring it to the next line. After the entire symbol is on the page, a subroutine is used to put in the three "I" symbols forming the connecting line leading down to the next box. The Locator is then adjusted back to the second line of the symbol, which is the first line to receive any Comments text (the line directly underneath the upper horizontal boundary). Successive calls are made on the Move-Line subroutine (**20.08**), until the last-line indicator is found to have been set. Test **20.09** leads to extracting (**22.03**) the destination tag of the SUBROUTINE (identified in field LHT) from the Tag Table, and its corresponding Page # and Box # (from the Tag Table) are placed within the symbol. If the tag is undefined, its Page # and Box # are omitted from the symbol. It should be noted that printing of the destination tag in parentheses within the subroutine symbol is not handled by Pass III; this field is inserted as a regular Comments line by Pass I, and Pass III handles it merely as another line of Comments. Also, any vertical editing of lines, for better spacing, is controlled by Pass-I through the insertion of dummy control symbols representing blank lines, thereby effectively spacing the lines properly within the symbol.

T-code

A T-code generates no symbol, but merely results in the placing of text on a page. Successive calls are made on the Move-Line subroutine (**20.08**) until the last-line indicator is set.

Pseudo-Connectors

Pseudo-connectors are short records generated by Pass II to indicate a connection from the bottom of one column to the top of the next column. These records (Table IV) are of a different format from the other input records, and are identified by the letter "X" in a fixed position of the record. The only information contained within this record, in addition to its Column and Line #, are the Page # and Box # to which the connector is jumping. Pseudo-connectors cannot have tags, since they are generated internally, nor are they assigned box numbers. After the address is computed (19.06) and the symbol is generated on the page area, test 19.07 leads to block 22.01, where the destination Page # and Box # from the pseudo-connector record are placed within the symbol. It is always necessary to add "1" to the Box # before placing it inside the symbol. This is because Pass-II, when setting up the pseudo-connector record, uses a Box # which is one too low. There is no logical basis for this; it is purely a matter of convenience in the implementation of Pass II.

DECISION Records

The handling of DECISION records presents a far more difficult problem than other codes, primarily because of the many courses of action available on the branches from a DECISION. Each DECISION record has two fields, LHT and RHT, in which destination tags are stored, and two indicators LHTC and RHTC that indicate one of the following courses of action:

(a) This branch requires a connector.

(b) The coding from this branch is located in the adjacent column.

(c) Use a line to indicate connection, if possible; if this is impossible then use a connector.

It should be noted here that the mnemonics LHT and RHT do not refer to "left" or "right" side; one of the decisions to be made by Pass III is which side of the DECISION symbol will indicate a particular branch.

Rules Regarding Connection of Branches

In laying out lines from DECISION symbols, the program follows certain pre-defined rules. These are:

(a) Wherever possible, connection lines are used in place of connectors.

(b) Where a connector must be used, the connector is always placed on the right side of the DECISION symbol unless this side is already in use (either by another connector or by a line to an adjacent column) in which case it is placed on the left.

(c) Connection lines may only appear (in this embodiment) in a lane on the left side of the DECISION symbol (**23.04** et seq.).

(d) Connection lines in the same lane are allowed to go to different destination symbols so long at they do not overlap (**23.08**).

(e) Connection lines are drawn (in this embodiment) only if the destination is in the same column as the DECISION symbol (**23.07**).

Detailed Description of DECISION Processing

The outline of the DECISION diamond is first moved to the page. This outline is carried as a constant within the program and is moved (**20.06**) to the page area by a series of moves controlled by a tally. The main-flow branch label is then moved from the Comments field to its position on the page and a subroutine is used to drop a vertical connection line to where the next symbol will be. It should be noted that a DECISION symbol has a vertical connect line consisting of four elements, while other symbols use three elements; this is necessary in order to guarantee proper clearance between any connectors and following F/C symbols. The Page Locator is then moved back up to the top of the symbol and the tag and Box # placed on the page alongside the symbol. The Locator is then moved down two lines in place to receive the first line of text. Successive calls are made on the Move-Line subroutine **20.06** until an indicator shows that all lines have been moved. Overflow of the DECISION diamond is not possible at this point; if overflow did occur, the excess Comment was truncated by the Pass–I editing logic.

After the DECISION symbol is completely laid out on the page, with its related tag and Box #, the logic to examine the branches begins (**21.01, 22.05**). In processing the branches, the same physical coding is used for processing both fields LHT and RHT. The branch tag currently being processed is always located in field RHT. When the tag originally located in RHT is finished, LHT is moved into RHT for its processing. An indicator is used so that the logic knows when it has completed processing the second tag and can go fetch a new record.

It is possible for either or both of the tag fields, LHT and RHT, to be blank; for a two-way decision, field RHT is blank. Both tags may be blank due to an error condition in the source program; in this case a DECISION diamond is printed with no branches. If a field is found to be blank, it is bypassed: thus, the program does not have to formally distinguish between a two- and three-way decision since bypassing blank tag fields automatically handles the problem. The following indicators are used throughout the DECISION-branch logic:

(a) Two signals LSS and RSS tell the logic whether the left side and right side respectively of the DECISION symbol have been utilized.

(b) An undefined symbol indicator UNDS is set by the subroutine which searches the Tag Table. This indicates that a tag has not been found in the table.

(c) An indicator TAGTA tells the logic whether it is processing the first or second branch tag.

At the start **22.05** of branch processing, indicators LSS and RSS are set to OFF, indicating that both sides of the DECISION symbol may be available. Indicator TAGTA is set to indicate that the first branch is being processed. A check is then made to see if the indicator RHTC says to go to the adjacent column, and if so, the fields LHT and RHT are reversed, along with their related indicators and labels. This is necessary in order to insure that the RHT branch is always processed first (an arbitrary convention), and consequently the right side of the DECISION symbol (which leads to the adjacent column) is initially made available for it. At this point, branch processing begins. A test is first made for the presence of a tag in this field. If the tag field is blank, the logic is routed to a test **25.01** of indicator TAGTA, which is described below. A subroutine **22.06** locates the branch tag in the Tag Table; and if found (**22.07**) the location data for this tag is extracted and stored (**23.01**) in an index register. If the tag in question is not found, indicator UNDS is set to ON for later use and test **22.07** routes the program to the branch **25.01** for drawing a connector symbol.

When the tag data is obtained, one of three courses of action is taken depending on the status of the tag indicator (NOTE **23.02**), which is then tested (**23.03**). If RHTC indicates a connection to the secondary (adjacent) column, a horizontal line is extended (**24.06**) to the right an appropriate number of positions. A vertical connecting line at the right-hand end of the horizontal line is then dropped (two elements in length) to connect with the top symbol of the adjacent column. The label is then placed just above the horizontal line, and RSS is set (**24.07**) to ON, indicating that the right side of the DECISION has been utilized.

If RHTC indicates (**23.03**) that a connector is to be used, a test **25.01** is made to see if RSS is ON, if it is ON, then the connector must be drawn (**25.05**) on the left side of the DECISION symbol. If RSS if OFF, then the right side of the symbol is used (**25.02**). Depending on the status of RSS, the Locator is either advanced or retarded to the left or right side of the DECISION symbol. The label is put on the page and the connecting lines between the DECISION symbol and the connector are drawn in, as is the symbol itself. After the symbol is drawn, the undefined tag indicator UNDS is tested. If the tag data is undefined, then the tag itself is placed within the generated symbol. Otherwise, the Page # and Box # of the destination are picked up from the Tag Table and placed within the symbol. Either LSS or RSS is then set (**25.06** or **25.03**) to ON, depending on which side of the symbol the connector was drawn.

The third course of action to be taken is when field RHTC indicates (**23.03**) that a line should be used if possible (NOTE **23.04**). The program determines whether it is feasible to draw a line; if not feasible, then a Connector is used. First LSS is tested (**23.05**); if it is ON, then the left side of the DECISION symbol is all ready in use. Since connection lines may only be drawn to the left (NOTE **23.06**), a Connector must be used on the right side, and the program is routed down that logic path **25.02**. If LSS is OFF (test **23.05**), then a test is made of UNDS; if this indicator is ON, then a Connector is used and the logic is routed down the path **25.05** for left-side Connectors. However, if the tag data is available, a test **23.07** determines if the tag symbol is in he same column. That is, the Page # and Col # of the branch tag (obtained from the Tag Table) are compared with the Page # and Col # of the DECISION symbol. If they are not the same, then the program is routed through the Connector logic **25.01**. If the Page # and Col # of the branch tag match that of the DECISION symbol, then the branch is in the same column as the DECISION and a line may be feasible. A check is then made to see if the destination of the branch is above or below the DECISION symbol. This is done by comparing the Line # of the DECISION symbol with the Line # of the branch tag. If the destination is below the DECISION symbol, then a "down" line must be used. A check **23.08** is made of the left lane reserved for connecting lines; if it contains an "I" then there is already a line in that column. This line is then traced **26.01** back to its destination, to see if its destination is the same as the destination of the current DECISION. If so, then a simple horizontal connection is made **26.02** to the line that already exists. If the destination of the existing line is not the same, then a Connector must be used for current branch tag and the program is routed to that logic **25.01**. If the test **23.08** determines that the column reserved for vertical lines is unoccupied, a down-line is drawn in (**24.01**). The ending point of the down-line is known from the Line # of the destination tag (obtained from the Tag Table). Appropriate horizontal connecting lines are drawn on the page, the label is placed in its proper position, and LSS is set (**24.02**) to ON.

If the destination tag is above the DECISION symbol, a similar type of logic is followed, with the following exception: in testing for the presence of an existing vertical line, the down-line logic had only to test one location—that element of the vertical-line lane immediately below the DECISION branch point. For an up-line, however, every element of the lane between the DECISION symbol and the destination must be tested. Otherwise, an up-line might interfere with an earlier up-line placed further up towards the top of the page. An additional complication may also arise whereby an up-line may interfere with a "from" connector, which is discussed below.

As explained above, one of several courses of action will be taken for each branch, depending on the setting of RHTC and the feasibility of drawing a line. At the end of each path, return is made to a common point **24.03**, where the indicator TAGTA is tested to see if this is the second or first branch just completed. If the indicator shows that the second branch has been processed, then processing for the entire record is not complete and control is returned back to block **19.02** to fetch a new record. If only the first branch has been processed, then fields LHT and LHTC are moved (**24.04**) into fields RHT and RHTC, respectively. The label for LHT (first five positions of Comments field) is moved into the label area for RHT (second five positions of Comment field), UNDS is set to OFF, and the logic recycled (**24.05**) to begin the process **22.06** for field LHT, now located in the area previously reserved for RHT.

Cross-References

The information for inserting cross-references ("from" Connectors) is contained in the Tag Table, where it was placed by Pass II. For each tag entry in the Tag Table, five characters are reserved for dealing with cross-references. The first four characters contain the Page # and Box # of the first reference to that tag. The fifth character is a counter of the total number of references to that tag. After a record is read from the input file, a test **20.01** is performed for the presence of cross-reference (this test is made prior to splitting each code down its own branch). This test examines the tag field for presence of a tag, since the absence of a tag indicates that there is no cross-reference. If a tag is present, the tag is looked up in the Tag Table. If the tag is not found in the Table, then it is undefined and cross-references are not possible. If the tag is found in the Tag Table, a check is made for the presence of a cross-reference by testing the Tag Table field for the first cross-reference; if that field contains spaces, no cross-reference exists.

If a cross-reference does exist, then it is necessary to determine (**20.02**) the relative position of the current symbol on the page. There are three possibilities:

(a) Middle of a chain.
(b) Top of a chain in a main column.
(c) Top of a chain in a secondary column.

How a cross-reference is shown on the page depends on where the current symbol is located on the page. At this time, the Page Locator is pointing to the center of the top line of the current symbol. This Locator is now backed off to three lines above the top line of the current symbol and a test is made for the character present in this location. From the page design used, it follows that if this character is the letter "I," then the current symbol is in the middle of a chain; if a minus sign, it is at the top of a chain in a secondary column. If the character is a space, the current symbol is at the top of a chain in a main column.

If the current symbol is at the top of the chain in a main column, the cross-reference is placed on the page centered about the column's centerline. The reference placed on the page is extracted from the Tag-Table entry, and an asterisk is inserted if the Tag-Table indicator is set to show more than one cross-reference (**20.03**, **20.04**). If the current symbol is at the top of a chain in a secondary column, then it is not desirable to show the Page # and Box # of the first reference, since this is the DECISION symbol connected by a line to this point, and showing the Page # and Box # here would be redundant and possibly confusing. However, a check **20.03** is made for more than one reference and if there is more than one reference, an asterisk in parenthesis is placed to the left of the centerline. If there is only one reference, then nothing is placed on the second column of the page.

If the current symbol is located in the middle of a chain (either main or secondary column), a check is made to see if there is interference with an existing line. Cross-references for this case are always inserted to the left of the column centerline and are two lines above the first line of the symbol. If there is a vertical down-line coming into this point from a DECISION symbol above, then this horizontal line is already occupied. The check is made by positioning the Page Locator to two lines above and one space to the left and checking the resultant location for a minus sign. If there is no minus sign, then the line is free; the first cross-reference is placed on the page, along with an asterisk if the indicator in the Tag-Table entry is set for more than one. If there is a minus sign in that location, a connecting line is being drawn and there is no need to place the Page # and Box # on the line, since this usually is the same as the symbol from which the line is drawn. In this case, a test is made for more than one reference and, if found, an asterisk in parenthesis is placed on the line, if there is only one reference, then no action is taken. The asterisk which may be placed on this line becomes part of the horizontal line coming into the centerline and thus provides notice to the user that there is at least one more reference besides the one shown via the connecting line.

If a cross-reference is placed to the left of the centerline, in the situation where the symbol is the middle of a chain, it is still possible that the subsequent drawing of an "up" connecting line will erase it. This can happen because up-lines come from symbols which are further down in the column and have not yet been processed. It cannot happen with down-lines, since they must come from symbols above the current one and hence will have already been drawn. That part of the DECISION branch logic which places up-lines on the page tests for the presence of an asterisk prior to drawing the horizontal connection back to the centerline of the column. If an asterisk is present, its position is moved up over the connecting line. If an asterisk is not present, then the line is either free of interference or there is but a single reference, which must be the one for which the line is presently being drawn. In either case, the horizontal connection line back to the centerline can be put in without any complication.

When the end-of-page indicator is detected (**19.04**) in the last input data block, the F/C page is complete in the memory layout. Thereafter, the entire page is put out to the printer (**21.05**), and the next page is started in the same fashion as described above. When the last page has been printed out, test **19.03** detects an end-of-file indicator to initiate (**21.03**) the rewinding of the tapes and calling in of Pass IV.

PASS IV

Production of the Cross-Reference List is accomplished by a separate pass, following the completion of the last page of flow chart. Input to the Cross-Reference pass is the same tape **111** that served as input to Pass III; output consists of the Cross-Reference List, either to an on-line printer **112** or to a magnetic tape for off-line purposes.

Two tables, both kept entirely in memory, serve as the basis for producing the listing. These tables are designated:

(1) Abridged Tag Table
(2) Reference Table

Tag Table

The Tag Table used by this pass is an abridged version of the main Tag Table (Table II) used by the first three passes. Each entry of the Abridged Table consists of the following items:

(1) Name of Tag
(2) Page # and Box # assigned to this tag
(3) First reference (Page # and Box #) to this tag
(4) An indicator which tells whether there are more references to this tag
(5) Memory address of the Reference Table entry containing the next reference (if there are any more references).

At the start of the Cross-Reference pass, the main Tag Table is still in memory from the previous pass. The first job **27.01** is to set up the Abridged Tag Table from the main table. Since each entry of the Abridged Table is shorter than its corresponding entry in the main table, the same physical memory area may be used for the Abridged Table. Every tag in the main table has a notation as to whether that tag is referenced by another symbol (created by Pass II for the flow chart layout). Only those tags which have references to them are moved to the Abridged Table; all others are dropped.

Each entry of the Abridged Table has four characters reserved for the first reference to this entry. This information is already available from the main Tag Table. However, for ease of implementation, this information is not transferred between tables, but is dropped. The only information transferred between the two tables, therefore, is the name of the tag and the Page # and Box # of that tag. Room is reserved in each entry for the remaining three items, which are filled in later on in the pass.

After all appropriate entries from the main Tag Table have been transferred to the Abridged Table, the latter is internally sorted **27.02** into a Page # and Box sequence. Any one of several known sorting techniques may be used for this purpose.

Reference Table

Each entry in the Reference Table consists of three items:

(1) Identification of this reference (Page # and Box #).
(2) An indicator telling whether this is the final reference or whether there are additional reference.
(3) Memory address of the entry in the reference table containing the next reference for this tag (if another reference exists).

The three fields of each reference entry are identical in format to the final three items of each Abridged Tag Table entry. One entry is created in the Reference Table for each reference (after the first one) for any given tag. A chaining technique is used to connect these references back to the Abridged Table entry to which they refer. Thus for any given tag, the first reference is in the Abridged Table entry, and succeeding references are spread out throughout the Reference Table, with each reference giving the location of the next reference in the chain. The chain is ended when the indicator in a particular entry says that is the last reference.

It should be noted here that each Abridged Table entry contains the first reference to that tag. Consequently a tag has entries in the Reference Table only if there is more than one reference. This choice of format was made for reasons of efficiency; that is, for most flow charts, the great majority of tags have only one reference, and for these, the Cross-Referencing can be handled entirely within the Abridged Table itself eliminating the need for access to the Reference Table. Other techniques may be used to collect the cross-reference data which has been developed by the F/C program and to present it in a simple table.

Setting up Reference in the Tables

The input file **111** to Pass–III also serves as the input file to this pass. Only the records which represent J-, D- or S-codes need be processed, since they are the only codes which involve "jumps" to other locations; consequently all other records may be bypassed without any processing.

Each D-, J- or S-record contains three fields that are of interest to this pass. The first field is the Page # and Box # assigned to the symbol on the flow chart. The second and third fields are the destination tags to which a transfer is called for by these records. In the case of J- and S-codes, only one of the latter fields contains a tag; for a D-code, either one or both of the fields contains tags, depending on whether the decision has one or two branches. It should be noted that the input records do not contain the Page # and Box # of the destinations, but only the tags of the destinations.

Prior to reading any input records, a locator must be set up for the Reference Table. This locator always contains the current RHE-plus-1 of the Reference Table. Since successive Reference Table entries are constructed extending to the right in memory, the locator always contains the memory locations at which the next entry is to be created. The Reference Table immediately follows the Abridged Table in memory, and is placed in its initial condition prior to starting the input file. Therefore, as each new entry in the Reference Table is created, the locator is incremented by a fixed constant.

When a D-, J- or S-record is read via blocks **27.03**, **27.04**, **27.05**, the first destination tag is extracted **27.06** from the record. A search **28.01** is then made of the Abridged Tag Table to locate the tag entry; if an entry is not found for that tag, the destination is undefined and no cross-reference is made. Upon locating the Table entry for that tag, the first-reference field is examined; if it only contains spaces, the current record is the first reference to this tag. The Page # and Box # of the input record (i.e. of the D-, J- or S-symbol) is then placed in the "first reference" field and the indicator in he entry is set to "last reference" status, which only indicates that the reference just inserted is thus far the final link of the chain for this entry.

However, if the first-reference field of the entry is already filled, then an entry in the Reference Table is created **28.02** for the reference from the current record. To create a linkage to previous reference (**28.02**), the existing chain of references for this tag is traced down to its end via a simple loop. The indicator in the Abridged-Table entry is tested; if set to "last reference" a new entry in the Reference Table is created at the next available address set up by its locator by extending the Table to the right by the length of the new entry. This address of the new Reference Table entry is placed **26.03** in the Abridged Table entry of the tag and its indicator is set to "not last reference" status. The Page # and Box # of the D-, J-, or S-symbol are extracted from the input record and placed in the newly created entry, and the indicator of the new entry is set to "last reference" status. The Reference-Table locator is incremented so that it again contains the RHE of the table. On the other hand, if the Abridged Table entry is set to "not last reference," the address of the Reference Table entry containing the next reference is picked up from the Abridged Table entry, and the indicator of that Reference Table entry is tested. The latter entry is either the "last reference" or it in turn leads to the next reference. Eventually the "last reference" entry in the Reference Table is located, and a new Reference Table entry is created, and filled in at the next available address set up by the locator. This address is placed in the previous Reference-Table entry for this tag, and the indicator therefore is reset to "not last reference." The Reference-Table locator is incremented to supply the next available address for any new entry to be created.

The above process is repeated if the input record is for a D-code and contains a second destination tag (via test **28.04**, NOTE **28.05**, and blocks **28.06**, **27.06**). After processing the second tag, or after the first tag processing if the second tag is not present, a new input record is read **27.03** and the entire process recycled. This process continues until an end-file indicator is found **27.04** in the input; construction of the Tables is then complete, and all the information needed for the listing is now contained within the Tables. Accordingly, the program transfers to the output section **28.07** of this pass.

Output

Production of the Cross Reference List consists of combining and printing out the contents of the two internal tables. Each entry in the Abridged Tag Table produces at least one line on the listing. Additional lines are used if the number of references to a particular tag overflows the amount of room available on the first line. The Abridged Table entries are handled sucessively with a locator being set **28.07** to the initial entry. The tag name and its Pages # and Box # is moved **28.08** from the Abridged Table to the output area; the first-reference field of this Table entry is also moved to the output area. The indicator of the entry is then tested for "last reference"; if not the last reference, the address of the next reference in the Reference Table is picked up **28.09**, **29.01** from that entry. The Page # and Box # of the next-reference entry is moved **29.02** to the output, the indicator of the new reference entry is tested **29.06**, and the logic recycled to block **28.09** if "last reference" is not found. The process is then repeated to locate the next entry and extract the desired data therefrom. When the indicator of any link specifies "last reference" (test **29.06**), the cycle ends and the current output line is printed **30.01** or written to the listing tape. Appropriate locators and counters are maintained for controlling placement of the references in the output line. When a counter indicates **29.03** that the output line is filled, the line is printed **29.04** or written to tape and a new line begun by checking **29.06** for further references.

The above processing is repeated by checking **30.02** for more tags in the Abridged Table, advancing a locator (**30.03**) to the next entry thereof, and recycling via block **30.04** and **28.08** to repeat the process until all entries in the Abridged Tag Table have been processed. After processing the final Abridged Table entry, test **30.02** determines that the listing is complete, and the program transfers to block **30.05** to "wrapup" any housekeeping details, such as rewinding the tapes, and the operation terminates (**30.06**).

The Cross-Reference List affords a valuable body of information that assists in reading and studying the flow chart. That is, each entry point marked with a cross-reference is known to have but a single transfer into that point, except where it is marked with an asterisk. In the latter case, the Cross-Reference List provides, under the tag of the entry point, a complete list of all other such transfers, which makes it possible to determine various interrelationships of the documented program.

MODIFICATIONS OF THE INVENTION

By modifications of the flow chart documentation system of this invention other forms of flow charts may be produced, such as those having characteristics illustrated in the fragmentary charts of FIGS. 4A and B, 5 and 6. FIGS. 4A and B and 5 present diagrammatically the interrelationships of D-symbols, each represented by a diamond **150**, and the other types of symbols all represented, for simplicity, by a rectangle **152**, except for J-, E-, H-symbols and connectors which are represented by circles. A four-column chart is assumed by way of example.

These flow charts may have one or more of the following features:

(1) A branch chain from a main flow column may be presented in an adjacent secondary-column as described above, and in addition the branch chains from the secondary-column may also be illustrated in the next adjacent column. See FIG. 4A where column **154** contains the main flow, column **156** contains branch chains from DECISIONS **150** and **162**, column **158** contains a branch chain from DECISION **164** in column **156**, and column **160** contains a branch from DECISION **166** in column **158**.

(2) A branch chain may be entered at an intermediate point of that chain as well as from the first block of the branch chain and branch chains may be shifted up or down so that they fit in the available space. See FIG. 4A where branch chains **158** and **160** are entered at intermediate points, chain **160** is shifted up and chain **170** is shifted down.

(3) A branch chain need not be entered directly opposite the branch output of the DECISION in the main column; the branch connecting line may be formed as a combination of horizontal and vertical line segments so that the branch chain may be positioned in any suitable place within the adjacent column. See FIG. 4A, branch chain **160** and connecting line **168**, and branch column chain **170** and line **172**.

(4) If a branch chain is not provided in a column adjacent to the main column, that adjacent column may be used for the continuation of the main flow, and all four columns of a page may be used for the main flow where appropriate and where branch chains are not or cannot be illustrated. See FIG. 4B, columns **174** and **176**. Each column has two possible vertical-line lanes, one on each side of the symbol, to permit connection in the same or adjacent columns (e.g., the lanes for lines **178** and **180** of column **158**, and the lanes for lines **182** and **184** of column **160**). The vertical lines can be connected up or down in each path. Thereby, in a four-column chart, eight vertical-line lanes are available for appropriate interconnections (and the use of all eight is illustrated in FIG. 4A).

(5) Vertical and horizontal lines may cross, (e.g., lines **168** and **180** in FIG. 4A) but provision is made to try alternative non-crossing paths.

(6) Connecting lines may be drawn between any two of the four columns, and these connecting lines may be directed either from left to right or from right to left, and may be a combination of vertical and horizontal line segments (e.g., line **168** of FIG. 4A, and lines **186** and **188** of FIG. 4B).

(7) Unconditional transfers (jumps or exit instructions) are represented by a line being drawn wherever possible, either to the same or to another column on the page (e.g., lines **186** and **188** of FIG. 4B). Similarly, pseudo-connectors are avoided where connecting lines can be drawn to the same page.

(8) Branch chains are connected either to the left or to the right, or both, of the main column containing the DECISION symbol from which the branch or branches occur (e.g., in FIG. 5 the DECISIONS in main-flow column **190** have respective branch chains **192** and **194** that are presented in columns on opposite sides of main column). Thereby, any column may be used for the main flow or for branch chains.

(9) A branch chain is picked up and printed if it fits in the space remaining on a page, be it one or more columns.

(10) Each flow chart page may be developed as a cluster of chains, with any one column or columns containing the main-flow logic and the remaining column or columns containing the chains branching from the main-flow.

A form of flow chart incorporating the last two features is shown diagrammatically in FIG. 6, in which the main-flow column of logic is illustrated in a simplified fashion by a relatively wide strip, and branch chains by a narrower strip (so that they can be readily distinguished) and JUMPS and pseudo-connectors at the ends of columns by circles. The simplified diagrams of FIG. 6 indicate the branching of chains from DECISIONS of the main-flow logic and from DECISIONS of the branch chains; the various F/C symbols are omitted to illustrate the general nature of the flow chart configurations that are handled. The aforementioned features of FIGS 4 and 5 are applied in illustrating the "cluster" feature.

FIG. 6A illustrates the four columns **200, 201, 202, 203** used for the main-flow logic (where no branches from DECISIONS occur that would fit in the remaining space on the page). The successive columns are connected by lines; alternatively pseudo-connectors may be used to terminate each column. FIG. 6B illustrates a column **204** of main-flow logic, from a DECISION of which a branch chain **205** is connected; and from a DECISION of the latter a sub-branch chain **206** is connected; and another sub-branch chain **207** connects from a DECISION in branch chain **207**. Where only a single branch chain (e.g., chain **205**) develops from the main-flow logic **204** and does not itself develop additional branch chains, only the main-flow logic **204** and branch chain **205** are printed on the page. Thereafter, the next page continues initially with the development of the main-flow logic and with the processing of branch chains as DECISIONS arise (and in the manner described with respect to FIG. 8). FIG. 6B illustrates the facility of displaying sub-branch chains to the right of the main chain.

FIG. 6C illustrates in the first column the main-flow logic **208**, from a DECISION of which there is a branch chain that is a long one and has sections **209, 210, 211** in three remaining columns of the page. The single column of main-flow logic and the single branch chain make up the page. If the branch chain terminates in the second or third column, the page likewise terminates.

FIG. 6D shows two columns **212** and **213** of main-flow logic and a branch chain **214** from a DECISION in the second column **213**, as well as a second branch chain **215** from a DECISION of the first branch **214**. Where the second branch chain **215** is not suitable for presentation in the fourth column, the page terminates with branch chain **214** and presents a cluster of the three columns **212, 213,** and **214**. This cluster feature of the F/C program does not attempt to use all of the available page space, but rather it is constructed to display as much of the branch interrelationships of the program being documented as the page size limitations permit. For practical reasons, the page size limits the amount of information that is presented as a unit.

FIG. 6E shows a column **216** of main-flow logic with a branch chain **217** connected from the right side of a DECISION thereof, another branch chain **218** connected from the left side of a DECISION thereof, and a sub-branch chain **219** connected from the left side of a DECISION of the left branch chain **218**. FIG. 6E illustrates the cluster feature of presenting branch chains on either side, or both sides of the column containing the main-flow logic, and the feature of sub-branch chains being allocated to the left for versatility in the display of branch information in each cluster.

In implementing the feature of forming successive pages as "chain clusters," the F/C Control Program is constructed to start each page with a column of main-flow logic. Upon reaching a DECISION record, the F/C program branches in the manner described above for the first embodiment, and processes a branch chain from that DECISION. In the course of processing that chain (or a second chain from the same DECISION), further branches may be developed from DECISIONS within the branch chain. If these sub-branches can be presented on the same page, they are developed in the same fashion. The information regarding the space required for the first branch is already known before the second or succeeding sub-branches are processed; therefore, the available space for the second branch or for sub-branches is then known, and the F/C can determine whether the sub-branches are suitable to be placed on the same page or not.

The implementation of an automatic system for producing a flow-chart formed from clusters of chains may assume different forms including that of a three-phase program of the same general type as shown in FIG. 3 (the fourth, cross-reference phase is optional). The detailed program logic may be similar to that described above in connection with FIG. 8, with certain modifications of Pass I and II, as described hereinafter.

Pass I

This logic is the same as described above for FIGS. 8/1 to 8/6, with the following additions. Each chain is given a separate identifier (e.g., a sequence number) in addition to the one already provided in the chain table, and this chain identifier is used in the Tag Table to associate all Tags that are part of a particular chain. Thus, the first chain is so identified and set forth in the Tag Table, and all Tags occurring in the first chain are listed in the Tag Table under (or in association with) the first chain. Thereafter, each time a new entry is created in the Chain Table for a chain ten ending (block **06.04**, FIG. 8) a corresponding new entry is likewise created for the Tag Table for the following chain, under which all associated tags are listed. An additional field is also provided in the Tag Table, which is used to furnish the relative line position of a particular tag within its associated chain. This information is available from the cumulative count in Line Counter B (block **04.01**) and inserted in the field upon the creation of each Tag Table entry (block **05.01**). Thereafter, a search for a particular tag in the Tag Table supplies the identifier of the chain in which the tag lies together with its line position within that chain.

Pass II.—Main Chain Processing

The general logic for this portion of the processing is shown in FIGS. 7A to E; details will be apparent from the following description and from the foregoing of FIG. 8. Upon the start **300** of the program, block **302** operates to reset counters and work storage areas to their proper conditions. The input data is the output tape from Pass I, in the manner described above. Thereafter, block **304** locates the next main chain via a pointer in the Chain Table; the first chain is assumed to be a main chain, and succeeding chains are also assumed to be main chains until they are assigned as auxiliary chains. Test **306** determines if there are any more main chains; and if not, the program branches to block **308**, which operates to rewind the tapes and bring in Pass III. If there is another main chain, the program proceeds with block **312** to read the next symbol record from the located chain. Test **314** determines whether the symbol is a J or an E, and if so, the program branches to a connector **316** leading to the auxiliary chain processing, ACP, described below with respect to FIG. 7B. If the current record is not the end of a chain, test **318** determines if the current record is at the end of a main column (i.e., if EPCON is exceeded, see block **08.01** of FIG. 8); if so, the program branches via block **320** to ACP as indicated by connector **324**. Block **320** creates the appropriate pseudo-connector symbol used to identify the end of the column and the page block number to which the program connects from that point, and it provides a supplementary record in the Chain Table. Each chain when processed, be it a main chain or an auxiliary chain, is marked in the Chain Table as processed. Under the circumstances of block **320**, a main chain cannot be so marked; however, the main chain can be marked as partially processed, with a store of the return record number to which the program will go to continue the processing after the remainder of the current cluster is completed.

If the current record is not at the end of the main column, the program continues with block **326**, which stores the record in a storage in memory that receives in order all the records making up a page and which starts with address LM, and also steps a counter LC to provide the next address in the record storage area for receiving the succeeding record; it also steps Line Counter–I an amount corresponding to the number of lines in the current record to obtain the line number in the main-chain column of the succeeding record. Thereafter, test **328** determines if the current record is a DECISION, and if it is, the program steps to subroutine CHSUB **330**, which determines if the branch chains from that DECISION can be pulled and inserted in auxiliary (adjacent) column. An indicator is set to identify for CHSUB that branch chain is a transfer from a main chain to distinguish from transfers from another branch chain. After subroutine CHSUB, the program returns to the main-chain processing at block **312**. If the current record is not a DECISION, the branch from test **328** is also back to block **312** to read the next symbol record from the main chain and repeat the above-described process.

Successive records of the main chain are processed in this manner until test **314** finds an end-of-chain record or test **318** finds an end-of-column record. In either case the program branches to ACP. The main chain processing is not interrupted for auxiliary chain processing, though it is interrupted for CHSUB to determine whether an auxiliary chain is appropriate to be pulled; all branch chains are examined to determine their suitability for display as an auxiliary chain. The actual processing of such auxiliary chains that are found follows the completion of the main chain processing. The actual assignment of page and block numbers to the main and auxiliary chain records is performed after the branch chains have all been identified and pulled and their records stored in memory.

FIG. 7B illustrates the logic flow for the subroutine CHSUB, which determines whether a branch chain can be pulled and utilized in an auxiliary column. Initial block **334** operates to locate the chain name for the tag to which the program branched from a DECISION record. The tag may be at the start of a chain or anywhere within a chain, and a search is made of the Tag Table to locate the associated chain name which was stored during Pass I. The line position of the tag within the chain is also extracted from the Tag Table. Thereafter, block **336** performs a calculation to determine CPPOC, the preferred position of the chain in the auxiliary column. CPPOC is calculated by subtracting the line position of the tag in the chain from the line position of the DECISION record from which the branch takes place. If CPPOC is a negative value, it is reset to zero; this represents a condition of the tag having a line position lower down on the page than the branch-point of the DECISION and CPPOC cannot be assigned a useable line number. Thereafter, block **338** stores CPPOC in a temporary storage field until it is determined whether that chain can be pulled for an auxiliary column. In addition, three sets of stores in memory are provided to hold the names of branch chains that may be pulled for auxiliary columns together with other chain-locating information such as CPPOC; these temporary stores are identified as CR, CL, and CB, representing respectively branch chains to the right, those to the left, and sub-branch chains that branch from the left or right auxiliary chains currently being processed. Thus, the CB stores contain a buffer storage of sub-branch chains, which branch to the right of right auxiliary chains or to the left of left auxiliary chains, whichever is currently being processed, and which sub-branches are to be processed thereafter. Each set of these chain-locating records may contain an arbitrary maximum number, say 10, which indicates the maximum number of branch chains that may be actually utilized in a particular auxiliary column.

Thereafter, test **340** determines whether the branch chain being investigated is a branch from a main-column chain or from another auxiliary-column chain. If it is from a main column, test **342** then determines whether the length of the current branch chain plus the length of the other branch chains already assigned to CR–1 to 10 would be greater than the column length EPCON. If not greater, there is still room in the right auxiliary column for the current branch chain, and block **344** sets up the next chain-locating record CR and stores the name of the current chain, its length, sequence number, and CPPOC in the appropriate fields thereof. Thereafter, test **346** determines whether there is a second, unprocessed branch in the current DECISION record being processed; if not, processing exits from CHSUB. If there is a second branch, block **347** locates the chain and calculates CPPOC for the second branch and test **348** determines whether the length of this branch chain together with the combined lengths of the other left auxiliary chains already assigned are greater than EPCON. If EPCON is exceeded, the program exits from the subroutine; if it is not, then block **350** stores the chain information in the appropriate fields of the next one of the chain-locating records CL–1 to 10 for the left auxiliary column, and the program exits.

If test **342** indicates that the branch chain is too large for the right hand column, test **352** determines whether the branch chain is suitable to fit in the first auxiliary left hand column. If it does not fit, test **354** determines if there is a second, unprocessed branch chain from the current DECISION record, and if not, the program exits from the subroutine; however, if there is a second branch, an indicator is set for processing the second branch and the program recycles back to the start of CHSUB to process it in the same fashion as the first branch chain was processed (and the latter is identified as processed).

If test **352** indicates that the branch chain will fit in the left auxiliary column the program branches to block **358**, and the chain information is stored in the next left chain-locating record CL–1 to 10. Thereafter, test **360** checks to see if there is a second branch chain from the DECISION record, and if not, the program exits from the subroutine. If there is a second branch, block **360** locates the chain and CPPOC is computed and stored in the manner described above, and test **362** determines whether it will fit in the right auxiliary column (the left having already been preliminarily assigned); if not, the program exits from the subroutine, if it will fit, block **364** stores the chain information in the next right chain-locating record CR–1 to 10, and the program exits.

If the test **340** indicates that the chain is being pulled by an auxiliary column, test **366** determines whether it is from the right or left auxiliary column from the setting of an indicator. If from the right, block **368** inhibits the pulling of any chains to the left and inhibits pulling more than one branch chain from the current DECISION record; this inhibition of pulling to the left, once initiated, continues for the remainder of the sub-branches pulled from the current auxiliary column being processed. Block **368** also sets an indicator to store the information for sub-branch chains from the current right auxiliary column in buffer chain-locating records CB–1 to 10, and thereafter the program continues with the processing from test **342** in the manner described above. Similarly, if test **366** indicates that the chain is being pulled from the left auxiliary column, block **370** inhibits the pulling of any chains to the right column, inhibits the pulling of more than one branch chain from the current DECISION record, and sets up CB–1 to 10 to receive the information of sub-branch chains from the current left auxiliary column. Thereafter, the program continues with the processing of the chain via test **342**.

ACP

The processing of auxiliary chains, ACP (FIG. 7C), begins with a subroutine SCHPL **380** (FIG. 7D) for calculating the exact starting line position SCHPL of each branch chain from CPPOC (the preferred starting line position of the chain with the tag opposite the DECISION branch point as calculated by block **336**, FIG. 7B) and storing the new value in the field FCPPOC of the associated one of the records CR, CL, CB. Each time the program passes through SCHPL and performs the calculations, a counter is incremented so that its count represents the number of auxiliary columns processed thus far. Thereafter, test **382** determines whether any right or left branch chains remain to be processed (by examining the contents of CR–1 and CL–1, as explained below) and whether any of the columns available on a page for a cluster remain unassigned; and if so, test **384** determines whether they are right auxiliary chains. If so, block **385** locates the chain specified in CR–1 on the input tape so that it can be processed. In addition, an indicator is set for the CHSUB subroutine to identify any sub-branch chain as coming from a right auxiliary chain.

Thereafter, block **386** begins the processing of this auxiliary chain by reading the first symbol record thereof from the input tape. Test **388** determines if the current record is an end-of-chain record (J or E). If not, block **390** stores the record in the next available location in memory as indicated by the address LC (appropriate marker signals are provided at the beginning of the records for each auxiliary column), and thereafter LC is adjusted to indicate the next available memory location for the next record to be stored.

Test **392** determines whether the current record is a DECISION record, and if not, the program returns to block **386** to read the next record and process it in the manner described. If the current record is a DECISION record, subroutine CHSUB is entered (with an indicator set to identify that the branching is from an auxiliary column) to determine whether the sub-branch chain from the current auxiliary chain can be pulled. In the subroutine CHSUB (FIG. 7B) test **340** steers the program down the auxiliary processing section and test **366** determines whether it is a right or left auxiliary currently being processed to provide appropriate steering, in the manner described above. When CHSUB is processing an auxiliary column, the sub-branch chain information is stored in the records CB. In addition, five counters CTR–**1** to **5** are used by CHSUB to maintain cumulative counts of the chain lengths for the five possible auxiliary columns, the three right and two left columns, in order. The right or left column indicator identified which type of column and the number of successive auxiliary column passes through CHSUB determines which column in order is being processed. Upon existing from CHSUB, the program returns to block **386** to process the next record in the branch chain for the current auxiliary column.

This loop continues until test **388** indicates that an end-of-chain record has been reached and the program branches to block **396**, which proceeds to store the record in memory and adjusts the setting of address LC. Thereafter, test **398** determines whether there are any more chains in the current column; this test may be performed by determining whether CR–**2** contains any data. If there are more chains, block **400** shifts the contents of CR–**2** to **10** into CR–**1** to **9**, respectively, so that the previous contents of CR–**2** are stored in CR–**1**, CR–**3** in CR–**2**, and so on. Thereafter, this routine begins again at ACT to process the chain now specified in CR–**1** in the manner described. If test **398** indicates that there are no more CR chains to be processed, the program branches to block **401**, which moves the contents of CB–**2** to CB–**10** into CR–**1** to **9**, respectively; thereby, the chain-locating records for the sub-auxiliary column are moved into position to be processed. Test **402** then checks an indicator to determine whether the auxiliary column just processed was a left or a right auxiliary; if a right auxiliary, then the new sub-auxiliary column is also a right auxiliary column, and the program returns to ACP to start the auxiliary chain processing for that sub-auxiliary column. If the auxiliary column just processed was a left column, the program branches to block **404**, which sets a field LL–**2** to the current value of LC; LL–**2** contains the memory address of the first record of the second left auxiliary column, and the program returns to ACP to begin the processing of that second left auxiliary column. An indicator is set to inhibit entering the subroutine CHSUB for any branches from this second left column, assuming a maximum of two left auxiliary columns. Similarly, that inhibit indicator is also set when test **402** steers the program down the "right" branch the second time, since the program is then starting to process the third right auxiliary, which is assumed to be the maximum; when a third auxiliary is processed, there is room (in the assumed 4-column example of a page) only for the first left auxiliary, and accordingly the processing of the second left is inhibited.

The indicator for identifying whether the auxiliary column being processed is a right or a left is set initially during the first pass through test **384** of ACP, which determines first whether there are data in the contents in CR–**1**. If there are, then it is known that it is a right auxiliary; and it is processed first on a priority basis with the right column indicator being set. If no data is in CR–**1** it must be a left auxiliary (since test **382** had indicated that there is an auxiliary), and the program branches to block **406**, which sets the left column indicator and sets LL–**1** (the address of the first record of the first left auxiliary column) to the current value of LC. Thereafter block **408** moves CL–**2** to **10** into CR–**1** to **9**, respectively, so that the left auxiliary chain-locating records are in condition to be processed in the same manner as the right auxiliary chain-locating records, and the program returns to the start of ACP for processing. After the first left auxiliary is processed, test **402** steers the program via block **404** to process the record left auxiliary. After the left auxiliary columns are processed, test **382** finds both CR–**1** and CL–**1** empty of data, and steers the program to the output subroutine **409**, OUTSUB, from which it returns to the beginning of the pass at STRT, FIG. 7A, to process the next main chain and from the cluster therefrom.

In operation, the first right auxiliary column chains are processed initially by following the contents of the CR records, and then the second right auxiliary column, if any, is processed with the sub-branch chains from the first right auxiliary column using the records in CB, which are transferred to CR for the processing operations. Thereafter, the third right auxiliary column, if any, is processed using the sub-branch chains that were pulled from the second right auxiliary column; the records for the third right auxiliary column are set up initially in CB as the second right auxiliary column is being processed. After all of the right auxiliary columns are processed, if space permits, test **384** steers the program to processing of the left auxiliary chains via blocks **406** and **408**, with the chains for the first left auxiliary columns being located by means of CL records, which are relocated into CR. After the first left auxiliary column is completely processed, test **398** indicates that there are no more chains for that column, and the program branches to block **401** to move the chain records CB for the second left auxiliary column into CR. Test **402** steers the program to block **404** to set up the address of the first record of that second left auxiliary column, and the program then proceeds to process those records.

When the records are written out to memory, the address of the first record of the main chain is LM, and the address LC is then used for storage thereafter of the successive records of the main chain, followed by those of the first-right, the second-right, and the third-right auxiliary columns in that order. Thereafter, the first-left and second-left column records are stored in that order. The records of the different columns are separated by appropriate marker signals. LL–1 and LL–2 provide the starting addresses of the records for the two left columns.

The subroutine SCHPL, shown in detail in FIG. 7D, calculates for each chain in an auxiliary column, its exact starting line position and stores it in the field FCPPOC of CR–1 to 10. It starts with CPPOC, the preferred line position for the start of the chain, already stored in FCPPOC (block **338**), and the subroutine terminates with the exact position determined. In addition, the subroutine starts with the fields CTR–1, 2, 3, 4, 5 (developed by SCHUB) which contain respectively the total number of lines required by all of the chains in the first, second, and third right auxiliary columns and the first and second left auxiliary columns, in that order. The cumulative counts in the CTR fields are based on the packed lengths of the branch chains; i.e., it is assumed that the first branch chain in each auxiliary column starts at the first line and succeeding chains are positioned thereafter without extra spaces therebetween. This subroutine terminates with the CTR value adjusted to include any spaces inserted by readjustment of chains downward within the associated auxiliary column. Thus, SCHPL starts with the branch chains fitting in a column at least if they are moved up all the way, and proceeds to determine if they also fit when moved down to prepared positions.

Initially at the start of each column, block **410** sets CHPL equal to "1," and thereafter, each chain in the current auxiliary column is processed in order. Test **412** looks for the next branch chain in CR–1; if there is none, the subroutine exits. If a next chain is set up in CR–1, test **414** determines if the stored value of CHPL in working storage is greater than the value of the field FCPPOC of that branch chain; if it is, block **416** stores the value CHPL in FCPPOC as the exact line position for the chain. That is, if CHPL is the greater value, the starting line position is already far down in the column, and no further downward adjustment of the chain is desired; and CHPL is therefore used as the starting line position. Thereafter, block **418** adds the line length of the current chain to CHPL to obtain a new value of the latter, so that the initial CHPL value of "1" may be applied only to the first chain in a column. Thereafter, the subroutine returns to test **412** and exits. If test **414** indicates that CHPL is not greater than CPPOC, the program branches to determine if space is available in the column to move the chain down so as to use CPPOC for its starting line position. Only if the full space is available, will the chain be moved down. Block **420** calculates TEM, equal to the difference between CPPOC and CHPL, and representing the desired downward displacement of the branch chain in number of lines. Thereafter test **422** compares the constant EPCON (the column length in lines) with the quantity of TEM plus CTR for the current auxiliary column; if EPCON is the lesser, there is no room in the column for downward adjustment of the chain, CPPOC cannot be used and the program branches to block **416** to store CHPL in FCPPOC of the current chain and proceeded in the manner described above. If test **422** indicates that there is room in the column for adjustment, CPPOC remains unchanged in the CR record. The program continues with block **424**, and CTR for the current column is augmented by TEM and the new value of CTR is stored in its own field. Thereafter, block **426** moves the line number in FCPPOC to working storage for a new value of CHPL; since the field FCPPOC remains unchanged, the preferred starting line position is actually used for the chain. The program then continues with block **418**, where the value of CHPL in working storage is augmented by the current chain length to obtain a new value of CHPL for the next chain, and the program returns to test **412**.

The output subroutine OUTSUB is shown in FIG. 7E; it is entered after completion of the auxiliary chain processing and processes each cluster of columns by determining whether a new page is required and assigning the page, box, column and line numbers in each flow chart record in memory and writes the records to the output tape.

Initially, test **430** determines whether a new page is required; this test involves checking the number of unused columns, if any, on the current page (i.e., the columns required by the previous cluster or clusters) and comparing it with the number of columns required by the current cluster. If a new page is not required, block **432** adds "1" to the column number; if a new page is required, block **434** operates to add "1" to the page number and to set the column # to "0" and box number to "1" (the box numbers on each page are assigned sequentially in each column and in order from column to column starting with the column on the left). Thereafter, in either case, test **436** determines if there are any left auxiliary columns in the current cluster by checking the contents of LL–1 and 2; if not, the main column is the first column of the cluster and is displayed to the left on the page, with the other columns of the cluster to the right in order. Block **438** sets a pair of output pointers to LM (which is the start of the record area in the memory containing the first record of the main column chain) and to LC (the end of the last right auxiliary record) thereby bracketing the memory area of records to be processed. Thereafter, these bracketed records are computed and written to the output tape by the RITEOUT subroutine **440**, and upon exiting therefrom, the OUTSUB subroutine also exits as shown by connector **441**.

If there are left auxiliary columns, test **442** determines whether or not there is a second left auxiliary column; if not, block **444** sets the output pointers to start with LL–1 and to bracket the memory area of the records for the first left auxiliary; thereafter, the RITEOUT subroutine **446** processes those records, and upon exiting goes to block **438** to set up RITEOUT for the records of the remaining columns. If there is a second left auxiliary column, block **448** sets the output pointers to bracket the memory area for the corresponding records, and the RITEOUT subroutine **450** processes those records. Thereafter, the program passes to block **444** to initiate processing the first left auxiliary records, and so on. In this fashion the records are completed and written out starting with the column which should appear on the left on the flow chart.

In the RITEOUT subroutine (details are shown in block **440**), an initial test **452** determines if there are any more records in the bracketed memory area to be processed, and if not, the subroutine exits. If the bracketed memory area contains additional records, a test **454** determines whether the curent record is at the start of a column. If it is, block **456** adds "1" to the column number. Thereafter (or if test **454** proves negative), block **458** adds "1" to the box number and calculates the line number of each record by adding the line length of each record to its line number to obtain the line number of the succeeding record; main chains start at the first line and each branch chain starts at the line set in the field FCPPOC for that chain. Block **458** also stores the page, box, column and line numbers in the record. Block **460** thereafter writes the record to the output tape, and the subroutine returns to test **452** to repeat the loop for each succeeding record until all available records are processed.

After the OUTSUB subroutine is completed for all of the records of the cluster, the program returns to the beginning of the pass to process the next main chain and start the development of the next cluster. When all of the chains have been processed into clusters, Pass II is completed and Pass–III is started, in the manner described above. Various modifications may be made in Pass III as indicated above and also as discussed below. For example, a main chain may terminate before the end of a column leaving room for a small main chain thereafter; the documentation program is readily adapted to recognize this condition and to utilize the space efficiently by inserting the small main chain in the available main column space.

The flow chart documentation of a computer program in the form of clusters of chains (main and branch chains in main and auxiliary columns) enhances the two-dimensional character of the chart. It tends to supply the user with a greater amount of information about branches from the main-flow, since only one of four columns on a page is devoted to the main chain of a cluster, and the remaining columns display any of the branch chains that occur. Moreover, the sub-branching is also displayed, and as much sub-branching can be displayed as space permits; thereby, various loops and processing interrelationships and complexities in the documented program tend to be presented to the reader as he reviews each page of the chart. The main chain continues from column to column where branch chains do not occur or do not fit. The main chain also continues from page to page where it is of any substantial length, so that the main flow can be followed by flipping successive pages of the chart. But the intricacies of the documented program at any stage thereof generally occur at conditional transfer operations and they tend to be presented on a page displaying the cluster of branches from the main flow.

Various modifications may be made in the control system of this invention depending on the size and type of memory facilities provided by the computer. For example, where the computer has a large random-access memory (such as a core memory), the processing of the data may be made more efficient in the operation that requires searching for branch chains in the input tape during Phase II. The control system, as described above, only searches for those branch chains that do not exceed a specified length, say, not greater than one column. This search can be reduced by placing all potential branch chains (those which are less than the maximum length) on a separate magnetic tape during the Phase I operation; since these short chains are the only ones that may not be processed in their natural sequence on the input tape. The chain length, of course, is a variable that cannot be preset, and a special memory storage of the records making up a chain would be necessary in order to determine whether it was small enough to be a potential branch chain or not. Such a memory storage of records that would form one page column is provided and serves as a standard measuring unit for all chains that are developed. Those that fit within this memory area of a column length are placed in sequence on a separate output tape and serve as a search tape for potential branch chains. All other chains go on to another output tape in the usual fashion described above, and they are treated as main chains in the operation of the program. Such an arrangement would not result in any loss of processing time during Pass I, but would result in a substantial saving of time during Pass II, since the potential branch chains would be segregated and more readily searched, and the searching would not run through those chains that could not be branch chains. The order of chain processing is maintained and set forth in the chain table, which would indicate the tape that a particular chain was on. Thus, in picking up successive chains to be treated as main chains, either of the two tapes would function as a source, with the location of a particular chain on one of the two tapes being set forth in a chain table.

The searching of chains can also be reduced by filling the memory area that is available with as many potential branch chains as possible to eliminate the serial searching of these chains on the input tape. Another technique to reduce search time is to search for several chains at one time on the input tape with the search performed in the sequence of appearance of the chains. For example, the branch chains for the right auxiliary column can all be pulled out in one search when that auxiliary column is to be processed; thereby rewinding of the main tape separately for each branch chain would be avoided.

For computers having random-access disc or drum storage devices, the searching may be eliminated since the branch chain can be identified by its address on the disc or drum and retrieved directly in accordance with that address, which would be stored in the chain table instead of the sequence number.

Where all of the branch chains are separated initially, these chains may be preliminarily analyzed to determine whether a DECISION in a subsequent main chain (or auxiliary chain) refers to a tag in the branch chain. Thereby, it would be possible to pull such branch chains and allocate them to the subsequent DECISIONS as well as to preceding ones.

This invention is not restricted to the processing of small branch chains that fit in the remaining space in a column or on a flow chart page. The branch chains can be broken by pseudo-connectors (similarly as the main chains), and the Chain Table entry is used to identify the portion of the chain that has been processed and the portion that remains to be processed. Thereby, the remainder of the branch chain is picked up subsequently in the processing operation and displayed on an appropriate page of the chart identified in the pseudo-connector. Such a procedure enhances the two-dimensional character of the flow chart that is produced, since a large number of branch chains tends to be displayed on the same page as the associated DECISIONS. Even though only portions of the branch chains are shown on a page, the reader of the flow chart is generally given sufficient information to identify the type of operation performed in the branch chain and thereby given in a single page a greater amount of information about the interrelationships of the program both in the main flow and the various associated branches.

Where it is found desirable, the control system may be arranged to duplicate any or all branch chains in subsequent portions of the flow chart where the same branch chain was developed as a transfer from a preceding DECISION. Thereby, the user's reading of the flow chart is made more convenient, in that he does not have to move to different pages of the chart to identify the branch chains. In such a system, the second and subsequent displays of a branch chain are identified as duplicates, and the chain table maintains a record of first and subsequent displays of such branch chains. Such an arrangement may be a selective one under the control of the programmer in that a special code symbol may be provided to be inserted at the discretion of the programmer where he feels it desirable for branch chains to be duplicated. That is, each DECISION which would carry branch chains of special significance could be marked to present the branch chain as a duplicate (and in as much detail as desired). The program then operates with such a control signal to display the branch chain even though it may otherwise not be displayed at that particular location on the chart.

For small computers, those with memory sizes of appreciably less than 32,000 characters (e.g., 8,000) additional tape units (or random-access disc or drum storage) can be substituted to compensate for the smaller core memory. In Phase I, an additional tape is used to record the Chain Table entries. The Tag Table need not be developed during Phase I, but rather during Phase II; the tag information is extracted from the symbol record tape by means of an extra pass through all of the records (after the page, box, column and line number assignments are completed). In addition, the record for each symbol includes additional information, such as whether the symbol terminates a chain or not, and the format of the symbol to be printed stores with all of the printed elements at their various locations set forth in the record.

In Phase II, an additional tape is used to record the Tag Table entries, as indicated above; a tape is used to record the second-left auxiliary chains, if any, and a tape to record the main and any right auxiliary chains. Additional information in the form of a control record is added to the second-left auxiliary tape, which record sets forth the information pertaining to the number of columns required for each cluster and the distribution of the cluster over the different columns. This control record precedes each group of second-left auxiliary column records and defines the individual cluster. During Phase II, clusters are stored on the tapes prior to being assigned page, box and line numbers. The input to Phase II is the symbol record tape and the Chain Table tape from Phase I. During Phase II it has been found convenient to develop all cross-references in the form of a table which can be then utilized in the final tape that is constructed representing successive pages.

In Phase III, four tapes are provided to record each column of page on a separate tape, and a tape to describe the actual vertical and horizontal lines that are to be drawn on a page; the latter is placed on one of the column tapes preceding the associated column data, or it may be placed on a fifth tape. The small memory may only be large enough to hold a single record for each of the four columns and the matrix of required lines for the whole page. The page image is segmented into columns, and the columns and the matrix of line connections are merged by page and line during the printing process. Several passes of the data are required for drawing line connections on a page and for printing the page. Each conditional and unconditional branch is assigned page, box, column and line numbers during the Phase II allocation, so that all destination points are established for Phase III.

The first pass of Phase III reads the symbol record tape for a page and writes out a "line coordinate matrix" based on the destinations of symbol records for that page. The size of the matrix in characters, is the column length in lines multiplied by the number of vertical line lanes. For instance, if a vertical lane can be drawn on the left or right of a column and if each column is 150 lines long, and if there are 4 columns to a page, then a 1200 character matrix is required. A matrix character may represent any one of several "horizontal" or "point" conditions. For instances: it may specify a blank, an up arrow, a down arrow, a horizontal line pointing to the left, a horizontal line pointing to the right, a vertical line and others. This "matrix" describes line drawings on a page and is formed by examining each decision or branch record on the symbol record tape. Based on (1) the position of the decision or branch record and (2) its destination point, then (3) the matrix can be examined and the line drawings plotted within the matrix. For instance, if a decision in column **1**, line **25** of page **1** is to branch to column **2**, line **29** of page **1**, the broken line to the right, downward in the lane, and to the right may be drawn and can be described by five characters stored for lane **2**, as follows: one character (say, coded "A") for the horizontal line between the DECISION branch-point and the lane (formed of 3 dashes), three characters (each coded "1") for the three vertical lines in the lane running downward, and one character (coded "B") for the horizontal line from the lane to column **2** (formed of 5 dashes with an arrowhead). The number of combinations possible for any point in the matrix can be represented by one character (in different embodiments about 6 to 12 possibilities exist). In the above example, the characters A, 1, 1, 1, B would be stored in the matrix that represents lines **25** to **29** of lane **2**, of column **1**, of page **1**. As the records are read in page by page, they are written out on 4 tapes; each tape containing one column. The "line matrices" for all pages are written on a separate tape or they can be placed on one of the column tapes, with each page's matrix written before the column record for that page. The next pass of Phase–III reads in the matrix tape, reads in each column of the page, creates a line for output based on the matrix and the column information (as described above) and prints the output.

Another change resulting from the design of a system for a small computer is that Phase IV (the development of the cross-reference listing), is produced after Phase II and prior to the Phase III printing of the flow chart page, for convenience in developing cross-reference data for display on the flow chart.

The particular embodiment of the invention described above is directed to input data in the form of an assembly language program, a form that is quite commonly used. This invention is also applicable to programs written in a "machine-independent" language, i.e., a language which is not limited to a particular construction of a machine nor to its particular body of instructions. The "instructions" of a machine independent language are macro-instructions or statements that can be translated as a group of machine instructions. Where a program consists of such macro-statements presented in accordance with a consistent convention, they can be interpreted by a flow-chart control program constructed in accordance with this invention to produce a two-dimensional flow chart that is properly representative.

The documentation system of this invention can process higher-level languages such as COBOL, FORTRAN, JOVIAL and other languages by analyzing the source statement input. The analysis may be performed prior to Phase–I, say Phase–O, or performed concurrently with Phase–I. The source statements are analyzed without the programmer writing the special codes that are currently required where the input is in an assembly language input (e.g., the embodiment of FIG. 8) or the special codes may be used where desired as an alternative in each case and also to delete or combine various statements of the program. The source statements are analyzed to (1) determine the special code (i.e., the F/C symbol; e.g., a subroutine, a process, an unconditional transfer, a conditional transfer), and (2) determine the Comments to appear in the flow chart symbol. Depending on the amount of analysis desired, the Comment produced (1) may be dependent on the procedure statement itself or, (2) may be dependent on the procedure statement plus an analysis of the nouns (e.g., data fields) and their associated descriptors (e.g., file descriptions). In any event, the flow charting of this system is independent of the language used as input. For example, preprocessors to Phase-I analyze the higher-level language and then produce input to Phase-I of the system.

The source statements are analyzed to determine the corresponding special code or the flow chart symbol or symbols which are to be used to represent those statements (i.e., symbols such as subroutine, process, unconditional transfer or conditional transfer, or suitable modifications of those basic forms. This statement analysis may incorporate a well-developed technique used to translate higher level languages; that is, the translators for these languages (e.g., compilers or interpreters) provide well-known techniques for translating each language statement (operation) or data field into a symbol which would properly represent the operation and its relationship to other operations (e.g., tags) that may be involved in transfer operations. Thus, the state of the art is such as to permit the development of program processing for analyzing the statements of such languages into a form suitable for this documentation system to operate. The Comment field may vary depending on the amount of analysis of the language that may be desired and the amount of detail desired in the Comment field of the F/C symbol; thus, the Comment produced may be simply dependent on the procedure statement itself (e.g., a simple repetition of the verbal or algebraic statement making up the procedure statement) or it may be dependent upon the procedure statement together with an analysis of the nouns (e.g., the data fields) and their associated descriptors (e.g., the file descriptions for those fields). Suitable techniques for this purpose are likewise well-developed, and appropriate forms exist for different compilers.

Where the language statements involve complex logical conditions, standard techniques may be used to present the involved statement as the Comment itself. Alternatively, known compiler techniques may be used to break down the involved logical statement into its simple components so as to present each one as a conditional transfer (DECISION), whereby a logical sequence of conditional transfers is developed in the flow chart rather than the single involved statement.

By means of this documentation system, a higher level language program may be documented in a flow chart, and various ones of the features described above may be employed. That is, F/C symbols for the process blocks may vary in size with the requirements of the Comment field; chains of logical sequences may be developed in the manner described above with each chain terminating with an unconditional transfer; and conditional transfers and associated branch logic may be analyzed to develop a two-dimensional chart showing the branch chains from the DECISION symbols. Employing the technique described above for presenting a sequence of symbols on a flow chart as called for by the program to be documented, and upon reaching a conditional transfer, the branch chains are analyzed to determine the suitability of their being displayed in adjacent columns. Likewise, the cluster techniques would also be applicable so that sub-branch chains may be shown in two-dimensional configuration.

Thus, this invention provides mechanisms for automatically producing flow charts by means of a data processor operating on coded signals representing the instructions of a computer program to be documented. The mechanism for developing a two-dimensional flow chart pulls branch sequences of symbols for display on the same page as the main flow sequence from which the branch takes place. It processes the branch logic out of the order in which the logic appears in the original program in order to show the branch on the same page as the associated main flow logic. This mechanism involves the development of symbols for individual instructions or groups of instructions, and then the development of sequences of symbols and chains from the individual symbols. The mechanism further develops columns of symbols from the main flow symbols and from the branch chains and assembles the columns into flow chart pages of related symbols, with the relationships being shown by connecting lines or references to the page and box numbers of the symbols. A mechanism also pulls sub-branch chains that stem from branch chains, and displays them on the same page. The cluster mechanism uses the above mechanisms and, in addition, forms each page of flow chart from one or more chain clusters, where a cluster is developed as a column of main flow logic and as many branch and sub-branch chains associated therewith as may fit on the page. In one form of the invention, the main flow symbols are successively allocated along the column until a DECISION is processed, and then the branch chain therefrom is processed by determining whether it fits within the adjacent column. The allocation of the branch chain may take place before the allocation of the main column is completed, or it may take place after the main column is allocated. The allocation of symbols of a branch chain may similarly lead to a sub-branch chain upon reaching a DECISION in the branch chain, and the suitability of fit of the sub-branched chain can be determined at that point.

This invention also furnishes mechanisms for editing unnecessary details from the original program and to combine a group of instructions into a single symbol. Thereby, it permits a programmer to edit, simplify and explain the program and the various portions and sequences thereof by means of an informative flow chart without the labor of drawing the chart or of laying out the sections thereof. A mechanism scans the various parts of the COMMENTS field of each input instruction and extracts the pertinent parts as required: the explanatory COMMENTS; the code, if any, for the type of symbol to be displayed; the destination tags, if any, carried for branch and transfer instructions; and the code for DECISION labels, or the labels themselves. By means of the symbol codes, detailed program instructions that are unnecessary for an understanding of the program may be deleted (either by the absence of a code, or by applying a delete code thereto), and several instructions may be combined and displayed as a single symbol, which more clearly sets forth the overall function of the detailed instructions. For example, a group of 20 individual instructions of the program to be documented may be displayed by a single DECISION symbol fully representing the overall function of those 20 instructions, though none of them may be a DECISION instruction, or several of them may be subsidiary DECISIONS. The essential functions of the program, by this mechanism, are set forth in the flow chart with as much detail as the programmer may desire to show. The significance of the processing details may be incorporated in statements of COMMENTS, which can be set forth in a NOTE or TEXT. The COMMENTS field also permits the carrying of labels for DECISION symbols, since the branch conditions may not be readily apparent from the details of instructions that are edited. Likewise, the COMMENTS field serves as a vehicle for destination tags for branch and transfer instructions, since these tags may not be always available from the operands due to the editing process. The COMMENTS of explanation of the program may be as long as desired, and a mechanism appends continuation COMMENTS to preceding records and edits these COMMENTS so that variable size symbols for PROCESS, NOTE and TEXT can be drawn and set forth in the flow chart.

This invention may also be used to interpret assembly language programs without the use of special codes for the flow chart symbols. The instructions themselves may be interpreted to develop process, unconditional transfer, and branch instruction symbols. In addition, the commentary customarily provided in an assembly language program may be extracted to present NOTES and the contents of PROCESS symbols. Where the symbol codes are not used, the resulting flow chart may contain a good bit of detail that is not ordinarily required; however, some editing of the program can be obtained. For example, where a DECISION is formed by two or more individual instructions such as "compare" and "conditional transfer," appropriate techniques may be used for combining those instructions as a single flow chart symbol. Similarly, where a long string of PROCESS instructions occur, the individual instructions may be set forth in one PROCESS box with suitable separations between the individual instructions; thereby, a great deal of flow chart space is not wasted on the separations and connecting lines between PROCESS symbols. In addition, a programmer versed in a particular assembly language program may perform a small amount of editing by marking certain instructions, or groups thereof, with a "delete" code symbol to avoid unnecessary detail in the final flow chart.

A mechanism is also provided to illustrate on the flow chart the allocation data (page and box number) of all branch and transfer instructions which are the origin symbols for a particular tagged symbol on the chart. Thereby, the chart furnishes cross references to all of the originating points from which a particular branch or other chain stems. The cross reference list furnishes a full listing of such originating points. Connecting lines can be drawn between symbols on the same page if conflicts do not exist; the program mechanisms attempt to draw lines in the available lanes starting from one coordinate point of a page and attempting to go to the other. If such connecting lines cannot be drawn, then cross references are set forth on the chart.

Any of various types of output devices may be used to develop a record of the flow chart. The record may be a printed page made by a line-at-a-time printer or a digital plotter printer, or a momentary display record or a permanent photographic record made by a cathode ray tube or similar display device. The record may also be a magnetic tape recording of the corresponding page format.

The page format may take various forms: for example, the arrays of main flow and branch chain symbols may be set forth in parallel columns (or rows) interrelated in a two-dimensional chart by means of connecting lines or cross references; alternatively, main flow columns and branch rows (and sub-branch columns) may be used to develop the two-dimensional chart, as may any other suitable arrangement of symbol arrays. This invention is not limited to a fixed format of a columnar page; for example, with a cathode ray tube display, the magnification may be varied to permit different sizes of columns and individual symbols. The magnification for different sections of a column or of a page can be varied so that a low magnification can be used to fit a long chain in a column, and the magnification can be varied to determine the spacing between the symbols or columns to achieve the most suitable presentation for an individual page.

As described above, this invention may be used to develop flow charts from input programs written in any desired program language, including assembly and higher order languages. The invention is also adaptable to various types of computers including those having large and small memory capacities.

As previously noted, the stored-program embodiment of this flowcharting invention described above is preferred in that it is comparatively less expensive to construct than a fixed-program or hardware embodiment in the present state of the art. A stored-program processor such as that of FIGS. 8/1–8/30, or the more specifically detailed RCA 501 program noted below, enables one to modify, enlarge or simplify the system or any part thereof without rewiring the circuitry and changing any other hardware portion of the system.

The flowcharting processor of FIGS. 8/1–8/30 is not dependent on any particular form of computer. It may be used as the basis for providing a stored program for any of a number of specific general-purpose computers that are available, and the RCA 501 program set forth below is one such example; other computers for which such programs have been provided include the IBM 360, 1401 and 7090, Honeywell 200 and the RCA Spectra.

A general-purpose computer, as an elementary information-transformation machine or apparatus, has a built-in capability limited to the execution of basic instructions such as add, subtract, compare, branch, etc. The stored-program embodiment of this invention converts the general-purpose computer into a special-purpose or extended machine having a unique operation sequence or process. Thereby, the programmed computer takes on the character of the flowcharting processor (program) that controls and directs the operation of the computer's hardware processor. For example, the flowcharting program for the RCA 501 set forth below converts the RCA 501 computer into a flowcharting computer machine during the time it is controlling and directing the computer. In that general-purpose computer, and generally in others, the logic and control circuits for performing the various instructions of the machine are time shared. The control signals of a stored program embodiment of this invention specify the particular instruction or instruction combination to be carried out at each instant in a specific and interrelated sequence. Thus the program's signal combinations physically initiate the operation of various ones of the computer's logic and control circuits, and direct the activation and deactivation of the computer's circuits and devices in certain sequences and relationships physically determined by the signal combinations of the program.

The interrelations of the circuits, their operations, and the control signals of the stored flowcharting program retermine the unique character of the computer as a flowcharting machine when it is so programmed. The flowcharting program acts as a control mechanism for the general-purpose computer to establish the configuration of machine operations that form the process of this invention. The flowcharting program as a control mechanism also determines a particular machine configuration, which is uniquely established while that program is in control; that is, the aforementioned mechanisms for automatically producing flow charts are established by various sequencies of particular instructions and by various combinations of those sequences. The machine system for making two-dimensional flow charts incorporates, for example, the mechanism for pulling branch sequences of symbols for display on the same page as the main flow sequence; the mechanisms for developing symbols, and the sequences of the symbols; the allocation mechanisms; the mechanism for pulling the sub-branch chains; the cluster mechanisms; the editing mechanisms; the layout mechanisms. These mechanisms are combined to form a special-purpose mechane, which in an illustrated embodiment is a stored-program controlled general-purpose computer.

This invention may also be embodied in various forms of fixed or wired program embodiments. For example, a program form of this invention, similar to the RCA 501 program noted below, may be established in a read-only type of memory for use in those computers employing such memories for the control program. This "firmware" embodiment of the invention is constructed and operates in the same way, in all material respects, as the stored-program embodiment described above; the term "firmware" indicates the relatively permanent character of the sequence control formed by the program in a read-only memory as contrasted to the "software" embodiment of the program temporarily written in a read-write memory.

Another form of construction of the invention may employ a read-only memory to establish a macro-instruction embodiment of the invention; the physical removal of one read-only memory with its program and the replacement by another changes the computer's control mechanism. Each of the different blocks of FIGS. 8/1–8/30 may be separately constructed and identified as a macro-instruction. That is, each such block represents a sequence of basic machine operations that are required to carry out the overall opeartion of each block as shown in the drawing and as described in the specification. In addition, each block represents the control mechanism or macro-instruction for directing that operational sequence. It will also be apparent from the detailed RCA 501 program, noted below, how the macro-instruction for each such block may be constructed. The computer so constructed has a macro-instruction set composed of the different types of blocks of FIGS. 8/1–8/30; each of these macro-instructions controls the computer to perform the appropriate combination and sequence of basic machine operations in accordance with the corresponding sequence of basic instructions that make up the macro-instruction. The macro-instructions of the computer so constructed with a read-only memory are specific to this flowcharting invention, and the computer having such a macro-instruction set has a unique flowcharting configuration. The control program for sequencing the macro-instructions calls for each of the individual macro-instructions by an appropriate identification code using standard computer techniques, and the corresponding sequence of basic instructions is supplied from the read-only memory. The macro-instruction sequence is shown in FIGS. 8/1–8/30 and set forth in the associated description.

In a fully wired embodiment of the invention, each of the blocks of FIGS. 8/1–8/30 may be constructed as a separate unit of logic circuitry, which unit includes And/Or logic circuits, and flip-flop registers and switches that are interconnected to perform the block's logic. For example, each such block may be composed of those portions of the arithmetic unit 104 and control unit 106 that are required to carry out the corresponding machine operations in association with the common memory 102 to which all such units have access as required and with the input-output devices. The machine operations corresponding to the instructions of the RCA 501 program noted below may be used for the sequence of operations composing each such block, and the corresponding RCA 501 logic circuits or any other suitable form may be used therefor. Each such block has an input connection at which it receives an activation signal from a preceding block. Each block has at least one output connection, and in the case of decision blocks, two or three outputs. These output connections of each block are connected to the input connections of other blocks in the manner shown in FIGS. 8/1–8/30. The circuitry of each block is normally quiescent, and upon receiving an input activation signal, it proceeds to perform its sequence of basic machine operations including memory fetches and stores. Upon completing the sequence, each block generates an output signal that is transferred to the succeeding block as the latter's activation signal, and the latter block proceeds to perform its operations. In the case of a decision block, but a single one of its output connections receives the signal for activating the next block depending upon the results of the comparison or decision operation that is performed. Thereby, a single block is active at any instant, the sequence of active blocks is in accordance with the description of FIGS. 8/1–8/30.

In this way, a hardware embodiment may be constructed of a serially operating computer having the operational and control relationships shown in FIGS. 8/1–8/30. The latter are schematic block diagrams of different portions of such a flow-charting computer with the inputs and outputs interconnected in the manner illustrated. Other hardware embodiments may be constructed. For example, each of the RCA 501 instructions may be provided by a separate unit of logic; in addition, a wired sequence control unit is connected to all of said logic units and establishes their operating sequence in accordance with the RCA 501 program noted below. Such a sequence control unit is responsive to the completion of the operation of each unit for initiating the operation of the succeeding one, and similarly responsive to the various alternative results of the operation of a decision unit for initiating the operation of the proper one of the associated succeeding units. Thus, equivalent software, firmware and hardware embodiments of this invention may be constructed in accordance with its principles.

Various other modifications of this invention will be apparent to those skilled in the art from the above descriptions of illustrative forms of this invention. The appended claims are intended to cover such modifications as are encompassed by the scope and spirit of this invention.

Appended hereto is a print-out of a complete program of one form of this invention known as the "Autoflow" Documentation System. This print-out is written in an assembly language known as the "RCA 501 EZCODE Assembly System, and also in machine language produced by an assembly of the former. This program, when assembled into machine language for the RCA 501 Electronic Data Processing System, directs the operation of that computing machine in accordance with the invention and particularly with the form of the invention described above in connection with FIG. 8. A detailed flow chart of that form of the invention, was produced automatically by an RCA 501 computing machine using the appended assembly language program as input data and using that program in machine language form to direct the operation of the machine. Copies of the appended program and of the aforementioned flow chart have been deposited with the U.S. Register of Copyrights.

ASSEMBLY LANGUAGE AND MACHINE LANGUAGE PROGRAM—RCA 501

```
NAME  AUTOFLOW                                    INDEX    4000C501                  DATE 111665    PAGE 001
   PROGRAM BLOCK  000         MSC  002510   LSC  002607
   PROGRAM BLOCK  001         MSC  003000   LSC  005457
   PROGRAM BLOCK  002         MSC  004170   LSC  020157
   PROGRAM BLOCK  003         MSC  004170   LSC  011777
   PROGRAM BLOCK  004         MSC  004430   LSC  011357
   PROGRAM BLOCK  005         MSC  007450   LSC  022167
   PROGRAM BLOCK  006         MSC  003000   LSC  003017
   PROGRAM BLOCK  007         MSC  004170   LSC  011047
   PROGRAM BLOCK  008         MSC  002510   LSC  002527

NAME  AUTOFLOW                                    INDEX    4000C501                  DATE 111665    PAGE 002
   PROGRAM BLOCK  000         STARTS AT 002510
  002510 40 646356 45  535666
  002520 01 010101 01  010101
  002530 01 010101 01  010101
  002540 01 010101 01  010101
  002550 01 010101 01  010101
  002560 00 002510 00  000000
  002570 00 000001 0C  010011
  002600 00 000000 72  000000

NAME  AUTOFLOW                                    INDEX    4000C501           SEG 01  DATE 111665    PAGE 003
   HSM   OP    A  N   B       SEQ #  TAG      OP  A ADDRESS             N  B ADDRESS          COMMENTS
                              000010          PID 40000501                 003000377777       AUTOFLOW
```

```
NAME  AUTOFLOW                                          INDEX     4000/501        SEG 01  DATE 111665     PAGE 004
  HSM    OP     A    N     B      SEQ #  TAG      OP  A ADDRESS            N   B ADDRESS             COMMENTS
                                  000020          SEG 01                         003000                HOUSEKEEPING
003000 71 004170 00 000000        000030 LOAD1    TC  START                                            BEGIN PROGRAM
003010                            000040 OPAR     DAR 5,0,0                                            OUTPUT AREA FOR
                                  000050          SLC OPAR,1                                           PASS 1, I/P AND O/P
                                  000060          RMK                                                  FOR PASSES 2 AND 3
003011                            000070 LINES    DAR 3                                                # LINES ON CHART
003014                            000080 ADDR     DAR 4                                                LOCATION OF BOX
003020                            000090 BOXNO    DAR 4                                                BOX ADDRESS
003024                            000100 CODE     DAR 1                                                PROCESS CODE
003025                            000110 TAG      DAR 7                                                TAG
003034                            000120 LHTC     DAR 1                                                LEFT CONNECTION
003035                            000130 LHT      DAR 7                                                LEFT DEST TAG
003044                            000140 RHTC     DAR 1                                                RIGHT CONNECTION
003045                            000150 RHT      DAR 7                                                RIGHT TEST TAG
003054                            000160 OPSQ     DAR 4                                                SEQ # OF RCD
003060                            000170 OPLTH    DAR 4                                                LENGTH OF RCD
003064                            000180 COMM     DAR 4*6                                              COMMENTS FIELD
003775                            000190 TEMP1    DAR 3,1                                              TEMPORARY
004001                            000200 TEMP2    DAR 3,1                                                STORAGE AREAS
004004                            000210 TEMP3    DAR 7
004014 01 000034 01 010101        000220 TENTL    DAC 000034                                           LENGTH OF ENTRY
                                  000230          RMK                                                  IN TAG TABLE
                                  000240 TENTLX   EQL 000033                                           LNTH-1 OF TABLE ENTRY
004020 01 000000 01 010101        000250 NUMTAG   DAC 000000                                           # TAGS IN TABLE
004024 01 060430 01 010101        000260 TTLOC    DAC TTAR                                             CURRENT MSC TABLE
004030 01 060430 01 010101        000270 TBIN     DAC TTAR                                             MSC TAG TABLE
004034 01 000000 01 010101        000280 NUMCH    DAC 000000                                           # OF CHAINS
004040 01 024020 01 010101        000290 CHLOC    DAC CHTAB                                            CURR MSC CH TABLE
004044 01 024020 01 010101        000300 CHIN     DAC CHTAB                                            MSC CHAIN TABLE
004050 01 000000 01 010101        000310 SEQNO    DAC 000000                                           SEQ # COUNTER
004054                            000320 TT1      DAR 1                                                INPUT TAPE
004055                            000330 TT2      DAR 1                                                WORK TAPE 1
004056                            000340 TT3      DAR 1                                                WORK TAPE 2
004057                            000350 TT4      DAR 1                                                WORK TAPE 3
004060                            000360 LSTCD    CON 1
       51
004064 01 003065 01 010101        000370 OPCOM    DAC COMM,1                                           MSC PLUS ONE OF
                                  000380          RMK                                                   O/P COMM AREA
004070 01 003010 01 010101        000390 AOPAR    DAC OPAR                                             ADDRESS OF O/P
004074 01 003035 01 010101        000400 ALHT     DAC LHT                                              ADDRESSES OF
004100 01 003045 01 010101        000410 ARHT     DAC RHT                                              TAG AREAS IN
004104 01 003025 01 010101        000420 ATAG     DAC TAG                                              OUTPUT AREAS
004110 01 003014 01 010101        000430 AADDR    DAC ADDR
004114 01 000152 01 010101        000440 EPCON    DAC 000152                                           # CHAR IN COLUMN
004120                            000450 TITLE    DAR 30
004156                            000460 INDEX    DAR 8                                                PASS ONE            R
                                  000470          RMK                                                  START OF PROGRAM    T
                                  000480          RMK
004170 22 001210 00 004205        000490 START    OCT 001210               * 1*5                       REWIND INPUT        P
004200 17 000000 00 000000        000500          RWD
004210 71 004240 00 010000        000510          TC  * 3                  BPO                         IF OFF-LINE OUTPUT, P
004220 22 001240 00 004235        000520          OCT 001240               * 1*5                       REWIND PRINT TAPE   C
004230 17 000000 00 000000        000530          RWD
004240 22 001220 00 004255        000540          OCT 001220               * 1*5
004250 17 000000 00 000000        000550          RWD                                                  RWD PASS 1 O/P  P
004260 22 001210 00 004054        000560 NEX      OCT 001210               TT1                         SET UP INPUT TRUNK
004270 22 001220 00 004055        000570          OCT 001220               TT2                         W/T 1
004300 22 001230 00 004056        000580          OCT 001230               TT3                         W/T 2
```

```
NAME  AUTOFLOW                                     INDEX     4000^501        SEG 01  DATE 111665      PAGE 005
  HSM    OP    A    N     B         SEQ #  TAG     OP  A ADDRESS          N   B ADDRESS               COMMENTS
 004310 22 001240 00 004057         000590         OCT 001240                   TT4                     O/P IF OFF LINE
 004320 75 000000 00 010000         000600         CSG                          010000                  CLOSE GATE           P
 004330 71 004350 00 400000         000610 NEX1    TC  NEX2                     400000                  B.P. 5 ON           D
 004340 71 004750 00 000000         000620         TC  RPT                                              READ PAPER TAPE      S
                                    000630         RMK                                                  END FILE NO.EOJ,YES. D
                                    000640         RMK                                                  IF EF, GO TO EOJ
 004350 22 004054 00 004365         000650 NEX2    OCT TT1                      . 1,5                   READ FIRST SOURCE    P
 004360 14 023200 00 000000         000660         LRF IPAR1                                            RECORD OF PROGRAM    C
 004370 72 005426 00 600000         000670         SET #73#                     $T                      WAS AN ED READ       D
 004400 43 023200 00 023200         000680         SC  IPAR1                    IPAR1
 004410 61 004430 00 004430         000690         CTC NEX3                     NEX3
 004420 71 004640 00 000000         000700         TC  NEX7                                                                  2
 004430 71 004470 00 400000         000710 NEX3    TC  NEX4                     400000                  B.P. 5 ON           D
 004440 72 023521 00 600000         000720         SET IPAR2,9                  $T                      COMPARE PROGRAM NO.  D
 004450 43 023225 00 023234         000730         SC  IPAR1,21                 IPAR1,28                TO PAPER TAPE        C
 004460 61 004610 00 004560         000740         CTC NEX6                     NEX5                                         3
 004470 22 005427 00 023235         000750 NEX4    OCT #75#                     IPAR1,29                TYPE OUT NUMBER      P
 004500 12 023225 00 770000         000760         LW  IPAR1,21                 770000                  OF PROGRAM          C
                                    000770         RIS START                 02 STPAS1                  BRING IN SEG 2       P
 004510 72 001167 00 600000                        SET 001167                   $T
 004520 24 004540 00 004557                        STC . 2                      . 3,7
 004530 71 000340 00 000000                        TC  000340
 004540 00 004170 00 000000                        00  START
 004550 00 000002 00 005770                        00  02                       STPAS1
                                    000780         RMK                                                  BEGIN FLOW CHART     N
                                    000790         RMK                                                  FOR THIS PROGRAM     C
                                    000800         RMK                                                  .STPAS1.             J
 004560 22 004054 00 004575         000810 NEX5    OCT TT1                      . 1,5                   UNWIND AN EF ON      P
 004570 06 720001 00 000000         000820         UWS 720001                                           INPUT FILE           C
 004600 71 004350 00 000000         000830         TC  NEX2                                                                  J
                                    000840         RMK                                                  ERROR ON P/T         T
 004610 12 023510 00 770000         000850 NEX6    LW  IPAR2                    770000                  TYPE ERROR MESSAGE   P
 004620 12 004710 00 770000         000860         LW  NEX8                     770000                  AND BAD RECORD       C
 004630 71 004330 00 000000         000870         TC  NEX1                                             TRY AGAIN            J
                                    000880         RMK                                                  ED ON INPUT SOURCE   T
 004640 71 005240 00 400000         000890 NEX7    TC  EOJ                      400000                  B.P. 5 ON           D
                                    000900         RMK                                                  ILLOGICAL.IF BPS     N
                                    000910         RMK                                                  ON, SHOULD NOT       C
                                    000920         RMK                                                  REACH ED ON I/P      C
 004650 12 023510 00 770000         000930         LW  IPAR2                    770000                  READ REST OF         P
 004660 12 004710 00 770000         000940         LW  NEX8                     770000                  P/T AND TYPE OUT     C
 004670 71 004750 00 000000         000950         TC  RPT                                              AS ERRORS            C
 004700 71 004650 00 000000         000960         TC  *-3                                              .EOJ.                J
 004710                             000970 NEX8    CON 12                       PROGRAM NOT
          576156466140540155566301
 004724                             000980         CON 14                       FOUND ON INPUT
          455664554301565501505557646
 004742                             000990         CON 1                        75
                                    001000         RMK                                                  READ PAPER TAPE      T
                                    001010         RMK                                                  MESSAGE.TEST FOR     C
                                    001020         RMK                                                  VALIDITY AND EF      C
 004750 25 000243 00 005100         001030 RPT     TCT $ST#                     RPTX                    SET EXIT             P
 004760 14 023510 00 770000         001040 RPT1    LRF IPAR2                    770000                  READ PAPER TAPE      F
 004770 72 000223 00 600000         001050         SET 000223                   $T                      WAS AN EF READ      D
 005000 43 004761 00 004763         001060         SC RPT1,1                    RPT1,3
 005010 61 005030 00 005030         001070         ETC *2                       *2
 005020 71 005240 00 000000         001080         TC  EOJ                                                                   7
 005030 72 005431 00 600000         001090         SET #7574#                   $T                      IS MESSAGE VALID     C
```

```
NAME  AUTOFLOW                                    INDEX    40000501       SEG.01  DATE 111665     PAGE 006
  HSM    OP    A     N     B       SEQ #  TAG     OP  A ADDRESS       N    B ADDRESS             COMMENTS
 005040 43 023511 00 023511      001100          SC   IPAR2,1              IPAR2,1
 005050 61 005110 00 005110      001110          CTC RPT2                  RPT2
 005060 43 023522 00 023522      001120          SC   IPAR2,10             IPAR2,10
 005070 61 005110 00 005110      001130          CTC RPT2                  RPT2
 005100 71 777777 00 000000      001140 RPTX     TC   777777                                       F
 005110 12 023510 00 770000      001150 RPT2     L,   IFAR2                770000     TYPE OUT MESSAGE   P
 005120 12 005150 00 770000      001160          LW   RPT3                 770000     AND EXPLANATION    C
 005130 01 000000 00 000000      001170          PES                                  HALT               H
 005140 71 004760 00 000000      001180          TC   RPT1                            BACK TO READ       J
 005150                          001190 RPT3     CON 18                    INVALID MESSAGE.
      50556540535043015444626240464416010l
 005172                          001200          CON 18                    PUT CORRECTION IN
      57646301425661614442635056550150550l
 005214                          001210          CON 17                    REAVER, HIT START
      6l44404344613501475063016263406163
 005235                          001220          CON 1                     75
                                 001230          RHK                                  END OF JOB         T
 005240 22 004054 00 005255      001240 EOJ      OCT TT1                   • 1,5
 005250 17 000000 00 000000      001250          RWD                                  REWIND INPUT       P
 005260 71 005330 00 010000      001260          TC   EOJ1                 BPO        IS OUTPUT OFF-LINE C
 005270 22 004057 00 005305      001270          OCT TT4                   • 1,5      WRITE ED TO O/P    P
 005300 12 005426 00 000000      001280          LW   #73#
 005310 22 004057 00 005325      001290          OCT TT4                   • 1,5      REWIND O/P         P
 005320 17 000000 00 000000      001300          RWD
 005330 12 005432 00 770000      001310 EOJ1     L*   *END OF RUN/*        770000     TYPE END OF RUN    P
 005340 76 000000 00 000000      001320          ST   6FOR                            HALT               H
 023000                          001330 LAYFR    DAR 120,0                 END
                                 001340          RHK                                  FRONT OF PAGE AREA
 023200                          001350 LAYOUT   DAR 15000,100             END
                                 001360          SLC LAYOUT
 023200                          001370 IPAR1    DAR 200,0                 END
 023510                          001380 IPAR2    DAR 200,0                 END
                                 001390          SLC IPAR2,R 1
 024020                          001400 CHTAB    DAR 3200,0                END
                                 001410          SLC LAYOUT,R 1
 024044                          001420 #0000    CON. 22,0                 END
      01010101010101010101010101010101010101010101
 005400                                 #0001    CON 22,0                  0000000000000000000000
      23232323232323232323232323232323232323232323
 005426                                 #0002    CON 01               *    73
 005427                                 #0003    CON 01               *    75
 005430                                 #0004    CON 02               *    7574
 005432                                 #0005    CON 11                    END OF RUN/
      4455430156450161645575
 005450 00 000000 00 000001                      CON 8,10             B    0000000000000001
```

```
NAME  AUTOFLOW                                    INDEX      40000501          SEG 02  DATE 111665      PAGE 007
  HSM    OP      A    N     B         SEQ #  TAG     OP   A ADDRESS            N   B ADDRESS              COMMENTS
                                      001430         SEG 02                        START
004170                                001440         CON 5
         0101010101
004200                                001450 WACOM   DAR 450,0
005104 01 004201 01 010101            001460 COMAD   DAC WACOM,1                                          COMMENTS WORK AREA
005110                                001470 HDTAG   DAR 7                                                MSC COMMENTS
005117                                001480 HDLAB   DAR 5                                                RESULT OF PAREN
005124 01 000000 01 010101            001490 LNCNT   DAC 00,0000
005130 01 000000 01 010101            001500 RHECOM  DAC 000000                                           LINES IN CURR CHN
                                      001510         RMK                                                  STORAGE FOR RHE
005134 01 023200 01 010101            001520 STOR1   DAC IFAR1                                            OF COMMENTS
005140 01 023510 01 010101            001530 STOR2   DAC IPAR2                                            USED BY PASS 1
005144                                001540 IPCOM   DAR 76,0                                               READ ROUTINE
                                      001550         RMK   INPUT AREAS FOR FI EL DS-MOVED FROM READ A REA BY INPUT SUB
005260                                001560 IP      DAR 24,0
                                      001570         SLC IP
005260                                001580 IPTAG   DAR 7                                                TAG
005267                                001590 IPCODE  DAR 1                                                F/C CODE
005270                                001600 IPLN    DAR 6                                                LINE NUMBER
005276                                001610 IPAD    DAR 7                                                A ADDRESS
                                      001620         SLC IP,R 1
                                      001630         RMK   STORAGE AREAS COR RE SPONDING TO ABOVE IN PUT AREAS
005310                                001640 WA      DAR 24,0
                                      001650         SLC WA
005310                                001660 WATAG   DAR 7                                                TAG
005317                                001670 WACODE  DAR 1                                                F/C CODE
005320                                001680 ASLN    DAR 6                                                LINE NUMBER
005326                                001690 ADWA    DAR 7                                                A ADDRESS
                                      001700         SLC WA,R 1
005340                                001710 WALAB   DAR 1                                                STORAGE-LABEL CODE
005341                                001720 WADCA   DAR 7                                                STORAGE FOR ADDRESSES
005350                                001730 WADCB   DAR 7                                                ON DECISIONS
005360                                001740 LBCD    CON 11,0                      YZILQUFO#-3
        7071505360644556361426
005374                                001750 LBCD1   CON 1,0                       3
        26
005375                                001760         DAR 4
                                      001770         RMK TABLE CONTAINING LAB EL  CODES AND ASSOCIATE D LABELS
                                      001780         RMK FIRST CHARACTER OF E AC H.LINE IS CODE, REST  IS 3 LABELS
005404                                001790 LBTB    DAR 224,0
                                      001800         SLC LBTB
005404                                001810         CON 16                        YYES  NO
        7070446201015556010101010101
005424                                001820         CON 16                        ZNO   YES
        7155560101017044620101010101C101
005444                                001830         CON 16                        IHIGH LOW
        50475046470153566601010101010101
005464                                001840         CON 16                        LLOW  HIGH
        53535666010147504647010101010101
005504                                001850         CON 16                        OEQUALUNEQL
        60446064405364554465301010101 01
005524                                001860         CON 16                        UUNEQLEQLAL
        6464554460534460644053 01-101 0101
005544                                001870         CON 16                        FOFF  ON
        45564545010156550101010101010101
005564                                001880         CON 16                        OON   OFF
        56565501010156454501010101010101
005604                                001890         CON 16                        #PLUS MINUS
```

```
NAME  AUTOFLOW                                       INDEX    4000'501      SEQ 02  DATE 111665     PAGE 008
  HSM    OP     A     N     B      SEQ #  TAG      OP  A ADDRESS          N   B ADDRESS             COMMENTS
         36575364620154505564620101010101
 005624                             001900          CON 16                   -MINUSPLUS
         14545055616257536462010101010101
 005644                             001910          CON 16                   3EQUALHIGH LOW
         26446064405347504647015356660101
                                    001920          SLC LBTB,R 1
 005744 01 005404 01 010101         001930 LBTBX    DAC LBTB
 005750                             001940 VALCD    CON 11                   BCDEHJNPSTW
         4142434447515557626366
 005770 22 004054 00 014535         001950 STPAS1   OCT TT1                  INP11,5               PUT TAPE NUMBERS      P
 006000 22 004055 00 010011         001960          OCT TT2                  FP7,1                 IN I/O INSTRUCTION    C
                                    001970          RMK                                            STORE TITLE FROM      P
                                    001980          RMK                                            PID LINE              C
 006010 75 000000 00 010000         001990          CSG                      C10000                CLOSE GATE
 006020 22 004054 00 000151         002000          OCT TT1                  000151                SET TAPE SLOT
 006030 14 023200 05 000000         002010          LRF IPAR1           05                         READ NEXT RECORD  P
 006040 72 004155 00 600000         002020          SET TITLE,R              $T                    STORE TITLE FROM  P
 006050 24 023302 00 023337         002030          STC IPAR1,66             IPAR1,95              SECOND LINE          C
 006060 72 004165 00 600000         002040          SET INDEX,R              $T                    STORE INDEX #         F
 006070 24 023225 00 023234         002050          STC IPAR1,21             IPAR1,28
 006100 14 023200 05 000000         002060          LRF IPAR1           05                         READ AREA 1
 006110 75 000000 00 000000         002070          CSG                      000000                OPEN GATE
 006120 71 014300 00 000000         002080 RD1      TC  INP1                                       PICK UP FIRST RECORD  S
 006130 72 006137 00 600042         002090          SET *,7                  600042                IS CODE A C           D
 006140 43 005267 00 005267         002100          SC  IPCODE               IPCODE
 006150 61 006170 00 006170         002110          CTC FP1                  FP1
 006160 71 006120 00 000000         002120          TC  RD1                                                              Z
                                    002130          RMK                                            NOW HAVE FIRST        N
                                    002140          RMK                                            RCD TO BE PROC-       C
                                    002150          RMK                                            ESSED IN I/P AREA     C
                                    002160          RMK                                            START MAIN LOOP       N
 006170 25 005104 00 000153         002170 FP1      TCT COMAD                $5                    INITIALIZE FOR        P
 006200 25 004064 00 000173         002180          TCT OFCOM                $7                      NEW RCD             C
 006210 22 003024 00 004060         002190          OCT CODE                 LSTCD                 SAVE OLD CODE         P
 006220 36 003010 00 003773         002200          SCT OPAR                 OPAR,R                CLEAR O/P AREAS       P
 006230 22 017756 00 003010         002210          OCT #76#                 OPAR                  PUT IN ST MESS
 006240 22 017757 00 004200         002220          OCT #74#                 WACOM
 006250 72 005337 00 600000         002230          SET WAAR                 $T                    MOVE TAG, OPERANDS,   P
 006260 26 005260 00 005307         002240          STT IP                   IP,R                   LINE #, COMMENTS     C
 006270 36 005260 00 005307         002250          SCT IP                   IP,R                   FROM INPUT AREA      C
 006300 71 015730 00 000000         002260          TC  MVCOM                                       TO WORK AREAS        C
 006310 71 014300 00 000000         002270 FP2      TC  INP1                                       READ NEXT I/P RECORD  S
 006320 72 006327 00 600042         002280 FP3      SET *,7                  600042                CODE EQUAL TO C       D
 006330 43 005267 00 005267         002290          SC  IPCODE               IPCODE
 006340 61 006500 00 006500         002300          CTC FP4                  FP4
 006350 71 015730 00 000000         002310          TC  MVCOM                                      APPEND THIS COMMENT   P
                                    002320          RMK                                            TO BACK OF PREVIOUS   C
 006360 72 005310 00 600000         002330          SET WATAG                $T                    ONE IN COMMENT W/A    C
 006370 43 017760 00 017760         002340          SC  #01#                 #01#                  IF TAG FIELD IS       P
 006400 61 006310 00 006310         002350          CTC FP2                  FP2                   BLANK, MOVE TAG OF    C
 006410 72 005316 00 600000         002360          SET WATAG,R              $T                    C LINE TO TAG   C
 006420 24 005260 00 005266         002370          STC IPTAG                IPTAG,R               WORK AREA    C
 006430 72 005325 00 600000         002380          SET ASLN,R               $T
 006440 24 005270 00 005275         002390          STC IPLN                 IPLN,R                MOVE LINE # OF        P
 006450 34 005260 00 005266         002400          SCC IPTAG                IPTAG,R                C LINE TO WORK AREA  C
 006460 34 005270 00 005275         002410          SCC IPLN                 IPLN,R
 006470 71 006310 00 000000         002420          TC  FP2                                        READ NEW RECORD       J
 006500 22 005317 00 003024         002430 FP4      OCT WACODE               CODE                  MOVE CODE TO O/P      P
```

```
NAME  AUTOFLOW                                           INDEX    40000501        SEG 02  DATE 111665      PAGE 009
  HSM    OP    A     N    B       SEQ #   TAG      OP   A ADDRESS          N    B ADDRESS           COMMENTS
006510 44 004053 00 017763       002440            TCA  SEQNO,3                 #000001#,1          ADD 1 TO SEQUENCE #   P
006520 25 004050 00 003054       002450            TCT  SEQNO                   OPSQ                  AND MOVE TO OUTPUT  C
006530 22 017764 05 777777       002460            OCT  #77#                  5 777777              77 TO RHE COMM FIELD
006540 45 000153 00 017767       002470            TCS  $5                      #000002#,1          STORE RHE OF
006550 25 000153 00 005130       002480            TCT  $5                      RHECOM               COMMENTS FIELD
006560 72 005107 00 600000       002490            SET  COMAD,R                 $T                  IS COMMENTS FIELD
006570 43 000151 00 000153       002500            SC   000151                  000153              COMPLETELY EMPTY
006600 61 006640 00 006610       002510            CTC  * 4                     * 1
006610 72 000004 50 600000       002520            SET  000004               50 $T                  CREATE DUMMY
006620 24 017770 00 017773       002530            STC  #74010177#              #74010177#          COMMENT AND
006630 44 005133 00 017777       002540            TCA  RHECOM,R                #000003#,1          ADJUST RHE
006640 25 005104 00 000153       002550            TCT  COMAD                   $5                  LHE COMM TO AMS
006650 22 005317 00 006661       002560            OCT  WACODE                  * 1,1               IF THIS IS A B        P
006660 72 770001 00 600000       002570            SET  770001                  $T                  CODE, PUT TITLE       C
006670 31 020000 00 020001       002580            LNS  *BT*                    *BT*                IN TAG TABLE FOR      C
006700 61 007240 00 007030       002590            CTC  FP5                     FP4A                TABLE OF CONTENTS     C
                                 002600            RMK                                              IF T, BYPASS TAG
                                 002610            RMK                                              ROUTINE COMPLETELY
                                 002620            RMK                                              T CODE   NO=FP5,YES=  D
006710 25 004024 00 000133       002630            TCT  TTLOC                   $3
006720 22 017764 03 000001       002640            OCT  #77#                 03 000001              MOVE 77 TO MSC OF
                                 002650            RMK                                              TAG AREA IN TABLE
006730 72 000033 30 600000       002660            SET  TENTLX               30 $T                  MOVE 26 CHAR
006740 24 004201 00 004232       002670            STC  WACOM,1                 WACOM,26            OF TITLE TO TABLE
006750 72 770001 00 600000       002680            SET  770001                  $T                  IF TITLE LESS
006760 31 000002 33 000033       002690            LNS  000002               33 TENTLX              THAN 26 CHAR
006770 61 007000 00 007010       002700            CTC  * 1                     * 2                 CLEAR EXCESS
007000 34 000001 23 000033       002710            SCC  000001               23 TENTLX              CHARACTERS MOVED
007010 44 004027 00 004017       002720            TCA  TTLOC,R                 TENTL,R             INCR RHE TABLE
007020 44 004023 00 017763       002730            TCA  NUMTAG,R                #000001#,1          ADD TO NUM TAGS
007030 72 005310 00 600000       002740   FP4A     SET  WATAG                   $T                  IS THERE A TAG        D
007040 43 020002 00 020002       002750            SC   #02#                    #02#
007050 61 007240 00 007060       002760            CTC  FP5                     * 1                 PRP = NO TAG
                                 002770            RMK                                              MAKE TAG TABLE ENTRY  N
007060 72 003033 00 600000       002780            SET  TAG,R                   $T                  MOVE TAG FIELD        P
007070 24 005310 00 005316       002790            STC  WATAG                   WATAG,R               TO OUTPUT           C
007100 44 004023 00 017763       002800            TCA  NUMTAG,R                #000001#,1          ADD TO TAG COUNTER    P
007110 25 004024 00 000133       002810            TCT  TTLOC                   $3
007120 72 000007 30 600000       002820            SET  000007               30 $T                  MOVE TAG TO           P
007130 24 005310 00 005316       002830            STC  WATAG                   WATAG,R               TAG TABLE           C
007140 25 004050 03 000013       002840            TCT  SEQNO                03 000013              SEQ # TO TABLE        P
007150 36 000014 33 000023       002850            SCT  000014               33 000023              CLEAR ADDR FIELD      P
007160 72 000027 30 600000       002860            SET  000027               30 $T                  MOVE ASSEMBLY LINE    P
007170 24 005322 00 005325       002870            STC  ASLN,2                  ASLN,5              NUMBER TO TABLE       C
007200 22 005321 03 000010       002880            OCT  ASLN,1               03 000010
007210 22 017761 03 000000       002890            OCT  #00#                 03 000000
007220 36 000030 33 000033       002900            SCT  000030               33 000033
007230 44 004027 00 004017       002910            TCA  TTLOC,R                 TENTL,R             INCREMENT RHE TABLE   P
                                 002920            RMK  SWITCH BELOW IS TFR  ON  FIRST RCD OF EVERY CHAIN=SET BY CHENT   N
007240 72 014030 00 004000       002930   FP5      SET  FRC                     004000              SWITCH    =FRC,TFR=   W
007250 72 020003 00 600000       002940   FP5A     SET  #51#                    $T                  J CODE                D
007260 43 005317 00 005317       002950            SC   WACODE                  WACODE
007270 61 007310 00 007310       002960            CTC  * 2                     * 2
007300 71 010050 00 000000       002970            TC   JCODE
007310 72 020004 00 600000       002980            SET  #41#                    $T                                        Z
007320 43 005317 00 005317       002990            SC   WACODE                  WACODE              K CODE                C
007330 61 007350 00 007350       003000            CTC  * 2                     * 2
007340 71 010640 00 000000       003010            TC   BCODE                                                             Z
```

```
NAME  AUTOFLOW                                  INDEX    4000C501      SEG 02  DATE 111665     PAGE 010
 HSW   OP    A    V    B     SEQ #  TAG      OP  A ADDRESS          N   E ADDRESS           COMMENTS
007350 72 020005 00 600000   003020          SET #43)                   $T                  D CODE              D
007360 43 005317 00 005317   003030          SC  WACODE                 WACODE
007370 61 007410 00 007410   003040          CTC * 2                    * 2
007400 71 011510 00 000000   003050          TC  DCODE                                                          Z
007410 72 020006 00 600000   003060          SET #44)                   $T                  E CODE              D
007420 43 005317 00 005317   003070          SC  WACODE                 WACODE
007430 61 007450 00 007450   003080          CTC * 2                    * 2
007440 71 010370 00 000000   003090          TC  ECODE                                                          Z
007450 72 020007 00 600000   003100          SET #47)                   $T                  H CODE              D
007460 43 005317 00 005317   003110          SC  WACODE                 WACODE
007470 61 007510 00 007510   003120          CTC * 2                    * 2
007500 71 010210 00 000000   003130          TC  HCODE                                                          Z
007510 72 020010 00 600000   003140          SET #55)                   $T                  N CODE              D
007520 43 005317 00 005317   003150          SC  WACODE                 WACODE
007530 61 007550 00 007550   003160          CTC * 2                    * 2
007540 71 010540 00 000000   003170          TC  NCODE                                                          Z
007550 72 020011 00 600000   003180          SET #57)                   $T                  P CODE              D
007560 43 005317 00 005317   003190          SC  WACODE                 WACODE
007570 61 007610 00 007610   003200          CTC * 2                    * 2
007600 71 010440 00 000000   003210          TC  PCODE                                                          Z
007610 72 020012 00 600000   003220          SET #-2#                   $T                  S CODE              D
007620 43 005317 00 005317   003230          SC  WACODE                 WACODE
007630 61 007650 00 007650   003240          CTC * 2                    * 2
007640 71 010740 00 000000   003250          TC  SCODE                                                          Z
007650 72 020013 00 600000   003260          SET #66#                   $T                  W CODE              D
007660 43 005317 00 005317   003270          SC  WACODE                 WACODE
007670 61 007710 00 007710   003280          CTC * 2                    * 2
007700 71 011510 00 000000   003290          TC  DCODE                                                          Z
007710 72 020014 00 600000   003300          SET #63#                   $T                  T CODE              D
007720 43 005317 00 005317   003310          SC  WACODE                 WACODE
007730 61 010030 00 010030   003320          CTC FP8                    FP8                 V
007740 71 013530 00 000000   003330          TC  TCODE                                                          J
                             003340          RMK                                            INDIVIDUAL CODES    T
                             003350          RMK                                            RETURN TO THIS POINT C
007750 25 000173 00 003060   003360 FP6      TCT $7                     OPLTH               COMPUTE RCD LENGTH, P
007760 45 003063 00 004073   003370          TCS OPLTH,R                AOPAR,R             PUT IN O/P          C
007770 44 003063 00 017763   003380          TCA OPLTH,R                #000001#,1
010000 22 020015 07 000001   003390          OCT #75#               07  000001              EM TO O/P AREA
010010 72 000000 00 600000   003400 FP7      SET 000000                 $T                  WRITE RCD TO        P
010020 11 003010 07 000001   003410          SSW OPAR               07  000001                OUTPUT FILE       C
                             003420          RMK                                            END PASS SWITCH     N
010030 72 014130 00 004000   003430 FP8      SET ENDP1                  004000              SWITCH *ENDP1*END*  W
                             003440          RMK                                            PROCESS NEXT RECORD N
010040 71 006170 00 000000   003450          TC  FP1                                        RECYCLE             J
                             003460          RMK                                            PROCESS JUMP CODE   T
010050 71 016210 00 000000   003470 JCODE    TC  PAREN                                      CHECK FOR ASTERISKS S
010060 61 010120 00 010120   003480          CTC * 4                    * 4                 WAS TAG PRESENT
010070 72 003043 00 600000   003490          SET LHT,R                  $T                  IF NO ASTERISKS, USE P
010100 24 005326 00 005334   003500          STC ADWA                   ADWA,R              A ADDRESS AS TAG    C
010110 71 010140 00 000000   003510          TC  * 3
010120 72 003043 00 600000   003520          SET LHT,R                  $T                  MOVE DESTINATION.TAG P
010130 24 005110 00 005116   003530          STC HDTAG                  HDTAG,R               TO OUTPUT         C
010140 25 020023 00 003011   003540          TCT #000012#,1             LINES               SET # LINES TO 10   P
010150 44 005127 00 020027   003550          TCA LNCNT,R                #000010#,1          ADD TO LINE CTR     P
010160 71 016730 00 000000   003560          TC  CHENT                                      CHAIN TABLE ENTRY   S
010170 45 000173 00 017767   003570          TCS $7                     #000002#,1          ANY IS RHE RCD
010200 71 007750 00 000000   003580          TC  FP6                                        GO TO WRITE         J
                             003590          RMK                                            PROCESS H CODE      T
```

```
NAME  AUTOFLOW                                INDEX    40000501        SEG 02  DATE 111665      PAGE 011
  HSM    OP    A      N    B       SEQ #  TAG     OP   A ADDRESS       N   B ADDRESS        COMMENTS
 010210 25 020027 00 003011     003600 HCODE   TCT #C00010#,1              LINES            SET # LINES TO 10     P
 010220 44 005127 00 020027     003610         TCA LNCNT,R                 #000010#,1       ADD TO LINE CTR       P
 010230 72 020030 00 600000     003620         SET "J"                     $T               CODE OF NEXT RCD      D
 010240 43 005267 00 005267     003630         SC  IPCODE                  IPCODE             EQUAL TO J          C
 010250 61 010300 00 010300     003640         CTC H1                      H1
 010260 45 003013 00 017777     003650         TCS LINES,R                 #C00003#,1       REDUCE # LINES BY 3,  P
 010270 22 005267 00 003045     003660         OCT IPCODE                  RHT              PUT J IN FIELD RHT    C
                                003670         RMK                                            FOR USE BY PASS 3   C
 010300 45 000173 00 017767     003680 H1      TCS $7                      #000002#,1       AM7 IS RHE OF RCD
 010310 22 005267 00 010321     003690         OCT IPCODE                  * 1,1            IF NEXT CODE IS R OR  P
 010320 72 770001 00 600000     003700         SET 770001                  $T                 T, SET # LINES IN   C
 010330 31 020000 00 020001     003710         LNS "BT"                    "BT"               CURRENT CHAIN TO A  C
 010340 61 010350 00 010360     003720         CTC * 1                     * 2                HIGH # TO AVOID     C
 010350 44 005127 00 020033     003730         TCA LNCNT,R                 #001000#,1         USE OF SEC. COLUMN  C
 010360 71 007750 00 000000     003740         TC  FP6                                      GO TO WRITE           J
                                003750         RMK                                          PROCESS E CODE        T
 010370 25 020023 00 003011     003760 ECODE   TCT #C00012#,1              LINES            SET # LINES TO 10     P
 010400 44 005127 00 020027     003770         TCA LNCNT,R                 #000010#,1       ADD TO LINE CTR       P
 010410 71 016730 00 000000     003780         TC  CHENT                                    CHAIN TABLE ENTRY     S
 010420 45 000173 00 017767     003790         TCS $7                      #000002#,1       AM7 IS RHE RCD
 010430 71 007750 00 000000     003800         TC  FP6                                      GO TO WRITE           J
                                003810         RMK                                          PROCESS P CODE        T
 010440 25 020037 00 003011     003820 PCODE   TCT #000005#,1              LINES            SET # LINES TO 5      P
 010450 71 017060 00 000023     003830 P1      TC  EDLIN                   000023           EDIT LINE OF COMMENT  S
 010460 61 010510 00 010450     003840         CTC P2                      P1               END OF COMMENT FIELD  D
                                003850         RMK                                               YES=P1,NO=  C
 010470 44 005127 00 003013     003860         TCA LNCNT,R                 LINES,R          ADD TO LINE CNTR      P
 010500 71 007750 00 000000     003870         TC  FP6                                      GO TO WRITE           J
                                003880         RMK                                          WORD TOO BIG TO
                                003890         RMK                                            FIT ON LINE
 010510 25 010457 00 010527     003900 P2      TCT P1,7                    * 1,7            CREATE ARTIFICIAL
 010520 22 017760 05 777777     003910         OCT #01#                 05 777777            WORD WITH SPACE      C
 010530 71 010450 00 000000     003920         TC  P1
                                003930         RMK                                          PROCESS N CODE        T
 010540 25 020037 00 003011     003940 NCODE   TCT #000005#,1              LINES            SET # LINES TO 5      P
 010550 71 017060 00 000017     003950 N1      TC  EDLIN                   000017           EDIT LINE OF COMMENT  S
 010560 61 010610 00 010550     003960         CTC N2                      N1               END OF COMMENT FIELD  D
                                003970         RMK                                               YES=N1,NO=  C
 010570 44 005127 00 003013     003980         TCA LNCNT,R                 LINES,R          ADD TO LINE CNTR      P
 010600 71 007750 00 000000     003990         TC  FP6                                      GO TO WRITE           J
                                004000         RMK                                          WORD TOO BIG TO
                                004010         RMK                                            FIT ON LINE
 010610 25 010557 00 010627     004020 N2      TCT N1,7                    * 1,7            CREATE ARTIFICIAL
 010620 22 017760 05 777777     004030         OCT #01#                 05 777777           WORD WITH SPACE       C
 010630 71 010550 00 000000     004040         TC  N1
                                004050         RMK                                          PROCESS B CODE        T
 010640 25 020043 00 003011     004060 BCODE   TCT #000000#,1              LINES            ZERO TO # LINES       P
 010650 25 005130 00 000133     004070         TCT RHFCOM                  $3               SET UP ADDRESS        P
 010660 45 000133 00 000153     004080         TCS $3                      $5                 FOR MOVE            C
 010670 44 000173 00 000133     004090         TCA $7                      $3
 010700 44 000133 00 000153     004100         TCA $3                      $5
 010710 72 000000 70 600000     004110         SET 000000               70 $T               MOVE COMMENTS TO      P
 010720 24 777777 53 000000     004120         STC 777777               53 000000           OUTPUT AREA           C
 010730 71 007750 00 000000     004130         TC  FP6                                      GO TO WRITE           J
                                004140         RMK                                          PROCESS S CODE        T
 010740 71 016210 00 000000     004150 SCODE   TC  PAREN                                    PICK UP DESTINATION   P
 010750 61 011010 00 011010     004160         CTC * 4                     * 4                FROM * FIELD.       C
 010760 72 003043 00 600000     004170         SET LIT,R                   $T                 USE A ADDRESS       C
```

```
NAME  AUTOFLOW                       INDEX     40000501     SEG 02  DATE 111665     PAGE 012
  HSM   OP    A    N    B    SEQ #  TAG   OP  A ADDRESS    N  B ADDRESS           COMMENTS
010770 24 005326 00 005334   004180       STC A.WA              ADWA,R               IF NO * FIELD        C
011000 71 011030 00 000000   004190       TC  * 3
011010 72 003043 00 600000   004200 S1    SET LHT,R             $T                 MOVE TAG TO            P
011020 24 005110 00 005116   004210       STC HDTAG             HDTAG,R            LHT AREA               C
                             004220       RMK                                      NOW PICK UP            N
                             004230       RMK                                      COMMENTS TO PUT        C
                             004240       RMK                                      IN BOX                 C
011030 71 017440 00 000000   004250       TC  S.SP                                 SKIP ANY SPACES TO
                             004260       RMK                                      RIGHT OF * FIELD
011040 72 003074 00 600000   004270       SET COMM,8            $T                 MOVE DESTINATION   P
011050 24 003035 00 003043   004280       STC LHT               LHT,R              TAG TO COMMENTS   C
011060 22 017757 00 003064   004290       OCT #74#              COMM               AREA AND ENCLOSE  C
011070 22 020044 00 003065   004300       OCT #-3#              COMM,1             IN BRACKETS   C
011100 72 010001 00 600000   004310       SET 010001            $T
011110 31 003064 00 003075   004320       LNS COMM              COMM,9
011120 25 000223 00 000173   004330       TCT $STA              $7
011130 22 020045 07 000001   004340       OCT #.4#           07 000001
011140 44 000173 00 017777   004350       TCA $7                #000003#,1
011150 71 017060 00 000014   004360       TC  EDLIN             000014             EDIT LINE OF COMMENT   $
011160 61 011430 00 011320   004370       CTC S6                S25                END OF COMMENT FIELD   D
                             004380       RMK                                              YES*S25,NO*    C
011170 72 740002 00 600000   004390       SET 740002            $T                 INSERT BLANK LINE      P
011200 31 003064 07 000000   004400       LNS COMM           07 000000               AT TOP FOR BETTER    C
011210 72 000001 70 600000   004410       SET 000001         70 $T                  VERTICAL ALIGNMENT   C
011220 24 000001 27 000000   004420       STC 000001         27 000000
011230 44 000173 00 017763   004430 S1A   TCA $7                #000001#,1
011240 72 000001 70 600000   004440 S1B   SET 000001         70 $T                 INSERT BLANK LINE      P
011250 24 003064 07 000000   004450       STC COMM           07 000000               AT TOP FOR BETTER    C
011260 44 000173 00 017763   004460       TCA $7                #000001#,1          VERTICAL ALIGNMENT  C
011270 25 020023 00 003011   004470 S2    TCT #000012#,1        LINES              SET # LINES TO 10   P
011300 44 005127 00 003013   004480       TCA LNCNT,R           LINES,R            ADD TO LINE CTR        P
011310 71 007750 00 000000   004490       TC  FP6                                                         J
011320 71 017060 00 000015   004500 S25   TC  EDLIN             000015             EDIT LINE OF COMMENT   $
011330 61 011430 00 011350   004510       CTC S6                S3                 END OF COMMENT FIELD   D
011340 71 011240 00 000000   004520       TC  S1B                                                         Z
011350 71 017060 00 000015   004530 S3    TC  EDLIN             000015             EDIT LINE OF COMMENT   $
011360 61 011430 00 011400   004540       CTC S6                S5                 END OF COMMENT FIELD   D
011370 71 011270 00 000000   004550       TC  S2                                                          Z
011400 71 017060 00 000014   004560 S5    TC  EDLIN             000014             EDIT LINE OF COMMENT   $
011410 61 011430 00 011470   004570       CTC S6                $7                 END OF COMMENT FIELD   D
                             004580       RMK                                                YES*$7,NO*   C
011420 71 011270 00 000000   004590       TC  S2                                                          J
                             004600       RMK                                      NO ROOM FOR 1 WORD
011430 25 000243 00 000133   004610 S6    TCT $STP              $3                 CREATE ARTIFICIAL
011440 25 777767 30 011457   004620       TCT 777767         30 * 1,7               WORD AND SET
011450 22 017760 05 777777   004630       OCT #(1#           05 777777              TO TRY AGAIN
011460 71 777760 30 000000   004640       TC  777760         30                    EXIT
                             004650       RMK                                      NOT ENOUGH ROOM        T
011470 45 000173 00 017767   004660 S7    TCS $7                #000002#,1         SET ANY TO RHE         P
                             004670       RMK                                      IGNORE REST            N
                             004680       RMK                                        OF COMMENTS          C
011500 71 011270 00 000000   004690       TC  S2                                                          J
                             004700       RMK                                      PROCESS D CODE         T
011510 25 005130 00 011537   004710 DCODE TCT RHECOM            * 2,7              LOCATE FIRST STAR      P
011520 72 150001 00 600000   004720       SET 150001            $T                 IN COMMENTS FIELD      C
011530 31 000000 50 777777   004730       LNS 000000         50 777777
011540 61 012770 00 012770   004740       CTC D7                D7                 WAS STAR FOUND         D
                             004750       RMK                                      ASTERISK NOTATION      N
```

```
NAME  AUTOFLOW                                        INDEX     40000501            SEG 02  DATE 111665       PAGE 013
  HSM    OP    A     N     B      SEQ #  TAG       OP  A ADDRESS               N   B ADDRESS                 COMMENTS
 011550 25 000223 00 000153     004760           TCT $STA                        $5                        ANS TO *
 011560 25 000153 00 000113     004770           TCT $5                          $1                        AM1 TO *-1           P
 011570 44 000153 00 017763     004780           TCA $5                          #000001#,1
 011600 22 017760 00 004200     004790           OCT #01#                        WACOM
                                004800           RMK                                                       CLEAR 1SS FROM FRONT
 011610 45 000113 00 017763     004810           TCS $1                          #000001#,1                OF COMMENTS W/A
 011620 72 000001 10 600000     004820           SET 000001                   10 $T                        FIND FIRST SPACE     P
 011630 43 017760 00 017760     004830           SC  #01#                        #01#                      TO LEFT OF STAR      C
 011640 61 011610 00 011610     004840           CTC *-3                         *-3
 011650 72 000015 70 600000     004850           SET 000015                   70 $T                        MOVE 5 CHARACTERS    P
 011660 24 000002 11 000006     004860           STC 000002                   11 000006                    TO LAB AREA IN O/P  C
 011670 34 000001 15 777777     004870           SCC 000001                   15 777777                    CLEAR LABEL FIELD IN P
 011700 72 150001 00 600000     004880           SET 150001                      $T                        COMMENTS AREA        C
 011710 31 000011 77 000015     004890           LNS 000011                   77 000015                    CLEAR OUT ANY CRAP   P
 011720 61 011730 00 011740     004900           CTC * 1                         * 2                       IN CASE LABEL IS     C
 011730 34 000001 27 000015     004910           SCC 000001                   27 000015                    LESS THAN 5 CHAR   C
 011740 25 000113 00 004001     004920           TCT $1                          TEMP2                     SAVE AM1
 011750 22 017757 00 004200     004930           OCT #74#                        WACOM                     REPLACE 1SS AT FRONT
                                004940           RMK                                                       OF COMMENTS W/A
 011760 71 016210 00 000000     004950 D1A       TC  PAREN                                                 EXTRACT 1ST TAG AND  S
                                004960           RMK                                                        LABEL IN STAR FIELD C
 011770 72 003043 00 600000     004970           SET LHT,R                       $T                        TAG GOES TO          P
 012000 24 005110 00 005116     004980           STC HDTAG                       HDTAG,R                     OUTPUT AREA        C
 012010 72 000003 70 600000     004990           SET 000003                   70 $T                        LABEL GOES TO        P
 012020 24 005117 00 005123     005000           STC HDLAB                       HDLAB,R                    OUTPUT AREA         C
                                005010           RMK                                                       IF NO TAG OR LABEL,  N
                                005020           RMK                                                       SPACES ARE MOVED     C
                                005030           RMK                                                       TO OUTPUT AREA       C
 012030 71 016210 00 000000     005040           TC  PAREN                                                 EXTRACT 2ND TAG AND  S
                                005050           RMK                                                        LABEL IN STAR FIELD C
 012040 72 003053 00 600000     005060           SET RHT,R                       $T                        MOVE 2ND TAG AND     P
 012050 24 005110 00 005116     005070           STC HDTAG                       HDTAG,R                     LABEL TO O/P AREA  C
 012060 72 000010 70 600000     005080           SET 000010                   70 $T
 012070 24 005117 00 005123     005090           STC HDLAB                       HDLAB,R
 012100 72 005133 00 600000     005100           SET RHECOM,R                    $T                        IF ASTERISK FIELD
 012110 43 000151 00 000153     005110           SC  000151                      000153                    WAS AT END OF
 012120 61 012130 00 012170     005120           CTC * 1                         D1                        COMMFIELD, ADJUST
 012130 25 004001 00 000113     005130           TCT TEMP2                       $1                        TEMP2 TO RHE COMM FIELD
 012140 25 000113 00 005130     005140           TCT $1                          RHECOM                    RHE OF FIELD TO
 012150 22 017764 01 000001     005150           OCT #77#                      1 000001                    LEFT OF LABEL
 012160 25 005104 00 000153     005160           TCT COMAD                       $5                        SET LHE COMM FIELD
                                005170           RMK IF ASTERISK FIELD WAS    AT FRONT, NO ADJUSTMENTS NEEDED
 012170 44 000173 00 020053     005180 D1        TCA $7                          #000017#,1                SET AM7 TO POINT  P
                                005190           RMK                                                       TO COMMENTS FIELD    C
 012200 72 020013 00 600000     005200           SET #66#                        $T                        * CODE               D
 012210 43 003024 00 003024     005210           SC  CODE                        CODE
 012220 61 012240 00 012240     005220           CTC * 2                         * 2
 012230 71 013420 00 000000     005230           TC  ACODE
 012240 71 017440 00 000000     005240           TC  S,SP                                                                       Z
                                005250           RMK                                                       SKIP ANY SPACES TO
 012250 25 005130 00 000133     005260           TCT RHECOM                      $3                        RIGHT OF * FIELD
 012260 45 000133 00 000153     005270           TCS $3                          $5                        LESS THAN 13         D
 012270 45 000133 00 020057     005280           TCS $3                          #000014#,1                 COMMENT CHARACTERS  C
 012300 61 012410 00 012310     005290           CTC D4                          * 1
                                005300           RMK                                                       # CHAR IS LESS       N
                                005310           RMK                                                       THAN 13              C
 012310 72 000000 70 600000     005320           SET 000000                   70 $T                        PUT IN 2 1SS FOR     P
 012320 24 020060 00 020061     005330           STC #7474#                      #7474#                    2 BLANK LINES        C
```

```
NAME  AUTOFLOW                                    INDEX    4000'501    SEG F2  DATE 111665    PAGE 014
 HSM    OP    A     N    B      SEQ #  TAG    OP.  A ADDRESS        N  B ADDRESS          COMMENTS
012330 44 000173 00 017767   005340         TCA $7                    #000002#,1         ADJUST AMT              P
012340 71 017060 00 000015   005350         TC  EDLIN                 000015             PICK UP SINGLE LINE     S
012350 61 000240 00 000240   005360         CTC 000240                000240                OF COMMENT           C
                             005370         RMK                                          PRP,PRN ILLOGICAL
012360 25 020067 00 003011   005380 D2      TCT #000015#,1            LINES              SET # LINES TO 13       P
012370 44 005127 00 003013   005390         TCA LNCNT,R               LINES,R            ADD TO LINE COUNTER     P
012400 71 007750 00 000000   005400         TC  FP6                                      GO TO WRITE             J
012410 71 017060 00 000007   005410 D4      TC  EDLIN                 000007             EDIT LINE OF COMMENT    S
012420 61 012730 00 012520   005420         CTC D6                    D4B                END OF COMMENT FIELD    D
                             005430         RMK                                                 YES*D4B,NO*      C
012430 72 000001 70 600000   005440         SET 000001             70 $T                 SHIFT LINES DOWN        P
012440 24 003103 07 000000   005450         STC COMM,15            07 000000             BY ONE FOR BETTER     C
012450 44 000173 00 017763   005460         TCA $7                    #000001#,1         VERT ALIGNMENT          C
012460 72 000001 70 600000   005470 D4A     SET 000001             70 $T                 SHIFT LINES DOWN        P
012470 24 003103 07 000000   005480         STC COMM,15            07 000000             BY ONE FOR BETTER     C
012500 44 000173 00 017763   005490         TCA $7                    #000001#,1         VERT ALIGNMENT          C
012510 71 012360 00 000000   005500         TC  D2                                       J
012520 71 017060 00 000013   005510 D4B     TC  EDLIN                 000013             EDIT LINE OF COMMENT    S
012530 61 012730 00 012550   005520         CTC D6                    * 2                END OF COMMENT FIELD    D
012540 71 012460 00 000000   005530         TC  D4A                                                              Z
012550 71 017060 00 000013   005540         TC  EDLIN                 000013             EDIT LINE OF COMMENT    S
012560 61 012730 00 012600   005550         CTC D6                    * 2                END OF COMMENT FIELD    D
012570 71 012460 00 000000   005560         TC  D4A                                                              Z
012600 71 017060 00 000013   005570         TC  EDLIN                 000013             EDIT LINE OF COMMENT    S
012610 61 012730 00 012630   005580         CTC D6                    * 2                END OF COMMENT FIELD    D
012620 71 012360 00 000000   005590         TC  D2                                                               Z
012630 71 017060 00 000007   005600         TC  EDLIN                 000007             EDIT LINE OF COMMENT    S
012640 61 012730 00 012660   005610         CTC D6                    * 2                END OF COMMENT FIELD    D
012650 71 012360 00 000000   005620         TC  D2                                                               Z
012660 71 017060 00 000003   005630         TC  EDLIN                 000003             EDIT LINE OF COMMENT    S
012670 61 012710 00 012710   005640         CTC D5                    D5                 END OF COMMENT FIELD    D
012700 71 012360 00 000000   005650         TC  D2                                       FINISHED                J
                             005660         RMK                                          NOT ENOUGH ROOM         T
                             005670         RMK                                          IN THE DIAMOND          C
012710 44 000173 00 017767   005680 D5      TCA $7                    #000002#,1         FORCE END OF FIELD      P
                             005690         RMK                                          DROP REST OF COMMENT    N
012720 71 012360 00 000000   005700         TC  D2                                                               J
                             005710         RMK                                          LINE TOO SMALL FOR
                             005720         RMK                                          EVEN ONE WORD
012730 25 000243 00 000133   005730 D6      TCT $STP                  $3                 CREATE ARTIFICIAL
012740 25 777767 30 012757   005740         TCT 777767             30 * 1,7              WORD AND TRY
012750 22 017760 05 777777   005750         OCT #1#                05 777777             AGAIN
012760 71 777760 30 000000   005760         TC  777760             30                                            F
                             005770         RMK                                          USE LABEL CODE TABLE    T
012770 25 005744 00 000113   005780 D7      TCT LNTBX                 $1                 API TO MSC OF TABLE     P
013000 72 000000 10 600000   005790 D8      SET 000000             10 $T                 COMPARE LABEL CODE      D
013010 43 005340 00 005340   005800         SC  WALAB                 WALAB                AGAINST TABLE ENTRY   C
013020 61 013400 00 013400   005810         CTC D12                   D12                                        O
013030 72 000015 70 600000   005820         SET 000015             70 $T                 MOVE LABELS             P
013040 24 000001 11 000005   005830         STC 000001             11 000005             FROM TABLE TO           C
013050 72 000010 70 600000   005840         SET 000010             70 $T                 OUTPUT AREA             C
013060 24 000006 11 000017   005850         STC 000006             11 000017
013070 22 005340 00 013101   005860         OCT WALAB                 * 1,1              THREE WAY DECISION      D
013100 72 770001 00 600000   005870         SET 770001                $T
013110 31 005374 00 005374   005880         LNS LBCD1                 LBCD1,R
013120 61 013130 00 013270   005890         CTC * 1                   C10
                             005900         RMK THREE WAY DECISION      N
013130 72 003043 00 600000   005910         SET LNT,R                 $T                 MOVE A AND B ADDRESS    P
```

```
NAME  AUTOFLO,                                    INDEX    40000501         SEG 02  DATE 111665     PAGE 015
  HSV    OP   A      R    B      SEQ #  TAG       OP  A ADDRESS          N  B ADDRESS               COMMENTS
 013140 24 005341 00 005347   005920              STC WADCA                 WADCA,R                 STORAGE AREAS TO O/P  C
 013150 72 003053 00 600000   005930              SET RHT,R                 $T
 013160 24 005350 00 005356   005940              STC WADCB                 WADCB,R
 013170 72 020071 00 600000   005950 D9           SET #1515#                $T                      IF ANY OF THE         P
 013200 43 003035 00 003035   005960              SC  LHT                   LHT                     DESTINATION TAGS      C
 013210 61 013230 00 013230   005970              CTC * 2                   * 2                     HAVE EZCODE *         C
 013220 34 003035 00 003043   005980              SCC LHT                   LHT,R                   CONVENTION, CLEAR     C
 013230 43 003045 00 003045   005990              SC  R-T                   RHT                     THEM TO SPACES        C
 013240 61 013260 00 013260   006000              CTC * 2                   * 2
 013250 34 003045 00 003053   006010              SCC R-T                   RHT,R
 013260 71 012170 00 000000   006020              TC  D1                                                                  J
                              006030              RMK     TWO-WAY DECISION   T
 013270 72 020070 00 600000   006040 D10          SET #15#                  $T                      A ADDRESS HAVE        D
 013300 43 005341 00 005341   006050              SC  WADCA                 WADCA                   ASTERISK NOTATION     C
 013310 61 013350 00 013350   006060              CTC D11                   D11
 013320 72 003043 00 600000   006070              SET LHT,R                 $T                      USE B ADDRESS         P
 013330 24 005350 00 005356   006080              STC WADCB                 WADCB,R                 AS DESTINATION        C
 013340 71 013170 00 000000   006090              TC  D9                                                                  J
 013350 72 003043 00 600000   006100 D11          SET LHT,R                 $T                      USE A ADDRESS         P
 013360 24 005341 00 005347   006110              STC WADCA                 WADCA,R                 AS DESTINATION        C
 013370 71 013170 00 000000   006120              TC  D9                                            J
 013400 44 000113 00 020077   006130 D12          TCA $1                    #000020#,1              AM1 TO NEXT ENTRY     P
 013410 71 013000 00 000000   006140              TC  DM                                            RECYCLE               J
                              006150              RMK                                               PROCESS W CODE        T
 013420 44 000173 00 020103   006160 WCODE        TCA $7                    #000036#,1              ADJUST AM7 TO RHE     P
                              006170              RMK                                               OF O/P COMM FIELD     C
 013430 72 000000 70 600000   006180              SET 000000             70 $T                      MOVE IN CONSTANT      P
 013440 24 013470 00 013527   006190              STC S,CON1                S,CON4,R                FOR COMMENT           C
 013450 22 020005 00 003024   006200              OCT #43#                  CODE                    CHANGE CODE TO D      P
                              006210              RMK                                               PROCESS AS D CODE     N
 013460 71 012360 00 000000   006220              TC  D2                                                                  J
 013470                       006230 SWCON1       CON 2                   # 7474
 013472                       006240 SWCON2       CON 11                    -----------
            1414141414141414141414
 013505                       006250 SWCON3       CON 8                   # 7462665063424774
 013515                       006260 SWCON4       CON 11                    -----------
            1414141414141414141414
                              006270              RMK                                               PROCESS T CODE        T
 013530 25 017777 00 003011   006280 TCODE        TCT #000003#,1            LINES                   SET # LINES TO 3      P
                              006290              RMK                                               FOR VERT MARGIN       C
 013540 22 004060 00 013551   006300              OCT LSTCD                 * 1,1                   IS PREVIOUS CODE      D
 013550 72 770001 00 600000   006310              SET 770001                $T                        J, H, E, D, T       C
 013560 31 020104 00 020110   006320              LNS #5147444163           #5147444163#
 013570 61 013600 00 013750   006330              CTC * 1                   T3
 013600 25 005130 00 000113   006340              TCT R,ECOM                $1                      CHANGE ANY *    P
 013610 72 150001 00 600000   006350              SET 150001                $T                      IN COMMENTS WORK  C
 013620 31 004200 01 000000   006360              LNS W,COM              *1 000000                  AREA TO 77 OCTAL  C
 013630 61 013660 00 013660   006370              CTC T1                    T1                      TO FAKE EDLIN SUB  C
 013640 22 017764 02 000001   006380              OCT #77#               *2 000001                  INTO ENDING LINE AT   C
 013650 71 013610 00 000000   006390              TC  *-4                                           THAT POINT        C
 013660 71 017060 00 000034   006400 T1           TC  EDLIN                 000034                  EDIT LINE OF TEXT     S
 013670 61 013660 00 013660   006410              CTC T1                    T1                      END FIELD I'DIC SET   D
 013700 72 005133 00 600000   006420              SET RHECOM,R              $T
 013710 43 000151 00 000153   006430              SC  0 0151                000153                  REAL END OF           D
 013720 61 013730 00 013770   006440              CTC * 1                   T4                         COMMENT FIELD      C
 013730 44 005127 00 003013   006450              TCA LNC,T,R               LINES,R                 ADD TO LINE COUNTER   P
 013740 71 007750 00 000000   006460              TC  FF6                                                                 J
                              006470              RMK                                               PREV CODE ACT SHEET   N
```

```
NAME  AUTOFLOW                                   INDEX    4000'501        SEG 02  DATE 111665    PAGE 016
  MSM   OP    A    M    B        SEQ #  TAG     OP  A ADDRESS            B ADDRESS              COMMENTS
013750 22 020010 00 003024      006480 T3      OCT #551                  CODE                   CHANGE CODE TO N    P
                                006490         RMK                                              PROCESS AS NOTE     N
                                006500         RMK                                              INSTEAD OF TEXT     C
013760 71 010540 00 000000      006510         TC  NCODE                                                            J
013770 44 000153 00 017767      006520 T4      TCA $5               #000002#,1                  SET AMS TO NEXT LINE P
014000 22 017757 05 777777      006530         OCT #74#          05 777777                      ISS TO LEFT OF FIELD
014010 44 000173 00 017767      006540         TCA $7               #000002#,1                  AMT TO NEXT LINE  P
014020 71 013660 00 000000      006550         TC  T1                                           RECYCLE    J
                                006560         RMK  FIRST RECORD OF EACH  CHAIN                 N
014030 22 007247 00 007245      006570 FRC     OCT FP5,7            FP5,5                       REST SWITCH FP5     P
014040 22 003024 00 014051      006580         OCT CODE             * 1,1                       IF RECORD IS N      P
014050 72 770001 00 600000      006590         SET 770001           $T                          OR T CODE, SET      C
014060 31 020111 00 020112      006600         LNS *NT*             *NT*                        INDICATOR IN CHAIN  C
014070 61 014100 00 007250      006610         CTC * 1              FP5A                        TABLE SO PASS 3     C
014100 25 004040 00 000133      006620         TCT CHLOC            $3                          KNOWS 1ST RCD OF    C
014110 22 020113 03 000004      006630         OCT #40#          03 000004                      CHAIN IS N OR T     C
014120 71 007250 00 000000      006640         TC  FP5A                                         J
                                006650         RMK                                              END OF PASS 1       T
014130 22 004055 00 014145      006660 ENDP1   OCT TT2              * 1,5                       WRITE EF            P
014140 12 014147 00 000072      006670         LW  *,7              000072
014150 22 004055 00 014165      006680         OCT TT2              * 1,5                       WRITE ED            P
014160 12 014167 00 000073      006690         LW  *,7              000073
014170 22 004055 00 014205      006700         OCT TT2              * 1,5                       REWIND              P
014200 17 000000 00 000000      006710         RWD
014210 25 004040 00 000133      006720         TCT CHLOC            $3                          PUT 00 IN RHE PLUS 1
014220 22 017761 03 000000      006730         OCT #00#          03 000000                      OF CHAIN TABLE
                                006740         RIS START         03 STPAS2                      BRING IN SEG 02     P
014230 72 001167 00 600000                     SET 001167           $T
014240 24 014260 00 014277                     STC * 2              * 3,7
014250 71 000340 00 000000                     TC  000340
014260 00 004170 00 000000                     00  START
014270 00 000003 00 004600                     00  03               STPAS2
                                006750         RMK                                              *STPAS2*            J
                                006760         RMK                                              PASS 1 SUBROUTINES  R
                                006770         RMK                                              PASS 1 READ RTN     T
                                006780         RMK                                              GET NEXT RECORD WITH T
                                006790         RMK                                              CODE OTHER THAN     C
                                006800         RMK                                              SPACE, MOVE TAG,    C
                                006810         RMK                                              CODE, OPERAND, AND  C
                                006820         RMK                                              COMMENTS TO WK AREAS C
014300 25 000243 00 015330      006830 INP1    TCT $STP             INP1X                       SET EXIT            P
014310 25 005140 00 000113      006840 INP1A   TCT STOR2            $1                          AM1 TO MSC OLD RCD  P
014320 72 015470 00 000000      006850 INP1B   SET INP14                                        *INP14,TFR*     N
                                006860         RMK ABOVE SWITCH IS USED TO PICK UP ADDRESSES ON DECISIONS    N
                                006870         RMK      WHEN LABEL CODE IS OMITTED     C
014330 25 005140 00 014530      006880         TCT STOR2            INP11                       REVERSE             P
014340 25 005134 00 005140      006890         TCT STOR1            STOR2                       ALTERNATE AREAS     C
014350 25 014530 00 005134      006900         TCT INP11            STOR1
014360 25 005140 00 000113      006910         TCT STOR2            $1                          MSC RCD TO AM1      P
014370 62 014370 00 014370      006920         SSM *                *                           SENSE BINO MODE
014400 72 000000 10 600000      006930         SET 000000        10 $T                          END OF FILE         D
014410 43 020114 00 020114      006940         SC  #72#             #72#
014420 61 014530 00 014530      006950         CTC INP11            INP11
014430 22 010036 00 010035      006960         OCT FP8,6            FP8,5                       SET END PASS SWITCH P
014440 45 004053 00 020043      006970         TCS SEQNO,R          #000000#,1                  ANY PASS 1 O/P  D
014450 61 006500 00 006500      006980         CTC FP4              FP4                                             Z
                                006990         RMK                                              BRING BACK IN       P
                                007000         RIS LOAD1         01 NEX                           CONTROL SEGMENT   C
```

```
NAME  AUTOFLOW                                      INDEX     40000501       SEG 02  DATE 111665     PAGE 017
  HSM   OP    A     N    B       SEQ #  TAG     OP   A ADDRESS           N   B ADDRESS         COMMENTS
014460 72 001167 00 600000                      SET  001167                  $T
014470 24 014510 00 014527                      STC  * 2                     * 3,7
014500 71 000340 00 000000                      TC   000340
014510 00 003000 00 000000                      00   LOAD1
014520 00 000001 11 004260                      00   01                      NEX
                                 007010         RMK                                            *NEX*  J
014530 14 000000 00 000000       007020 INP11   LRF                                          READ SOURCE RECORD   P
014540 72 000011 10 600000       007030 INP12   SET  000011              10  $T              IS TAG FIELD BLANK  D
014550 43 020002 00 020002       007040         SC   #02#                    #02#
014560 61 014630 00 014570       007050         CTC  INP12A                  * 1             Z
014570 72 005266 00 600000       007060         SET  IPTAG,R                 $T              PICK UP TAG          P
014600 24 000011 11 000017       007070         STC  000011              11  000017          FROM SOURCE INPUT    C
014610 72 005275 00 600000       007080         SET  IPLN,R                  $T              MOVE ASSY LINE #     P
014620 24 000007 11 000007       007090         STC  000002              11  000007          TO STORAGE           C
014630 72 005304 00 600000       007100 INP12A  SET  IPAD,R                  $T              MOVE A ADDRESS       P
014640 24 000025 11 000033       007110         STC  000025              11  000033          TO STORAGE           C
014650 36 005144 00 005257       007120         SCT  IPCOM                   IPCOM,R
014660 72 000023 10 600000       007130         SET  000023              10  $T              IS THIS A RMK LINE   D
014670 43 020115 00 020117       007140         SC   *RMK*                   *RMK*
014700 61 015120 00 015120       007150         CTC  INP12C                  INP12C
014710 72 005257 00 600000       007160         SET  IPCOM,R                 $T              MOVE AND RIGHT       P
014720 35 000102 11 000137       007170         SCR  000102              11  000137          JUSTIFY ALL FIELDS   C
014730 61 014740 00 014750       007180         CTC  * 1                     * 2             OF RMK LINE          C
014740 24 017760 00 017760       007190         STC  #01#                    #01#
014750 35 000055 11 000100       007200         SCR  000055              11  000100
014760 61 014770 00 015000       007210         CTC  * 1                     * 2
014770 24 017760 00 017760       007220         STC  #01#                    #01#
015000 35 000052 11 000053       007230         SCR  000052              11  000053
015010 61 015020 00 015030       007240         CTC  * 1                     * 2
015020 24 017760 00 017760       007250         STC  #01#                    #01#
015030 35 000025 11 000050       007260         SCR  000025              11  000050
015040 34 005260 00 005266       007270         SCC  IPTAG                   IPTAG,R
015050 34 005276 00 005304       007280         SCC  IPAD                    IPAD,R
015060 72 005257 00 600000       007290         SET  IPCOM,R                 $T              WAS ENTIRE LINE      D
015070 43 017760 00 017760       007300         SC   #01#                    #01#            A NULL ONE           C
015100 61 015150 00 015150       007310         CTC  INP12B                  INP12B
015110 71 014310 00 000000       007320         TC   INP1A                                   BACK FOR NEW RCD     J
                                 007330         RMK                                          NOT REMARK LINE      N
015120 72 005257 00 600000       007340 INP12C  SET  IPCOM,R                 $T              MOVE COMMENTS        P
015130 35 000102 11 000137       007350         SCR  000102              11  000137          AND RIGHT JUSTIFY    C
015140 61 015150 00 014310       007360         CTC  * 1                     INP1A
015150 72 005256 00 600000       007370 INP12B  SET  IPCOM,R-1               $T              LAST CHARACTER       D
015160 43 020120 00 020121       007380         SC   #0101#                  #0101#            SEPERATED BY       C
015170 61 014310 00 014310       007390         CTC  INP1A                   INP1A             TWO SPACES         C
015200 22 005257 00 015211       007400         OCT  IPCOM,R                 * 1,1           LAST CHARACTER       D
015210 72 770001 00 600000       007410         SET  770001                  $T               A VALID FLOW CHART  C
015220 31 005750 00 005762       007420         LNS  VALCD                   VALCD,R            CODE              C
015230 61 015240 00 015360       007430         CTC  * 1                     INP13
015240 22 005257 00 005267       007440         OCT  IPCOM,R                 IPCODE          MOVE TO CODE AREA    F
015250 34 005257 00 005257       007450         SCC  IPCOM,R                 IPCOM,R         CLEAR LAST CHARACTER P
015260 22 005267 00 015271       007460         OCT  IPCODE                  * 1,1           TEST CODE FOR  D
015270 72 770001 00 600000       007470         SET  770001                  $T              D OR C         C
015300 31 020122 00 020123       007480         LNS  *DC*                    *DC*            D=INPIX,C=INP13A,NO=  C
015310 61 015330 00 015340       007490         CTC  INPIX                   INP13A
015320 22 020124 00 014320       007500         OCT  #71#                    INP1B           SET SWITCH INP1B     P
015330 71 777777 00 000000       007510 INPIX   TC   777777                                                       E
015340 22 020114 00 014320       007520 INP13A  OCT  #72#                    INP1B           RESET SWITCH INP1B   P
015350 71 015330 00 000000       007530         TC   INPIX                                                        J
```

```
NAME  AUTOFLOW                                          INDEX    4000-501         SEG 02  DATE 111665      PAGE C18
  HSM    OP     A     N     B          SEQ #  TAG      OP  A ADDRESS            N   B ADDRESS              COMMENTS
                                       007540          RMK                                                 NOT FLOW CHART CODE    N
 015360 22 005257 00 015371           007550 INP13    OCT IPCOM,R                  * 1+1                   IS THIS A VALID        C
 015370 72 770001 00 600000           007560          SET 770001                   $T                      LABEL CODE             C
 015400 31 005360 00 005372           007570          LNS LBCD                     LBCD,R
 015410 61 015420 00 014310           007580          CTC * 1                      INP1A                   PR-GET NEW RCD
 015420 22 005257 00 005340           007590          OCT IPCOM,R                  WALAB                   STORE LABEL CODE   P
 015430 71 015640 00 000000           007600          TC  INP15                                            PICK UP ADDRESSES      S
 015440 22 020125 00 005267           007610          OCT "C"                      IPCODE                  CHANGE CODE TO C       P
 015450 34 005257 00 005257           007620          SCC IPCOM,R                  IPCOM,R                 CLEAR LABEL CODE
 015460 71 015330 00 000000           007630          TC  INPIX                                            TO EXIT                J
                                       007640          RMK CODING BELOW IS TO       PICK UP ADDRESSES IN    ABSENCE OF LABEL CODE
                                                                                                                      T
 015470 25 000243 00 015630           007650 INP14    TCT $STP                     INP14X                  SET EXIT   P
 015500 72 020136 00 600000           007660          SET "TA TC CTC"              $T                      IS OP CODE A CTC,      D
 015510 43 000021 11 000023           007670          SC  000021                11 000023                  TC, OR TALLY           C
 015520 61 015540 00 015540           007680          CTC * 2                      * 2
 015530 71 015610 00 000000           007690          TC  INP14A
 015540 43 000021 11 000023           007700          SC  000021                11 000023
 015550 61 015570 00 015570           007710          CTC * 2                      * 2
 015560 71 015610 00 000000           007720          TC  INP14A
 015570 43 000021 11 000023           007730          SC  000021                11 000023
 015600 61 015630 00 015630           007740          CTC INP14X                   INP14X                  Y
 015610 22 020137 00 005340           007750 INP14A   OCT "Y"                      WALAB                   ASSUME Y CODE   P
 015620 71 015640 00 000000           007760          TC  INP15                                            PICK UP ADDRESSES   S
 015630 71 777777 00 000000           007770 INP14X   TC  777777                                           EXIT    E
                                       007780          RMK SUBROUTINE TO PICK UP    ADDRESSES    T
 015640 25 000243 00 015720           007790 INP15    TCT $STP                     INP15X                  SET EXIT               P
 015650 72 005347 00 600000           007800          SET WADCA,R                  $T                      PICK UP A ADDRESS   P
 015660 24 000025 11 000033           007810          STC 000025                11 000033
 015670 72 005356 00 600000           007820          SET WADCB,R                  $T                      PICK UP B ADDRESS   P
 015700 24 000055 11 000063           007830          STC 000055                11 000063
 015710 22 020114 00 014320           007840          OCT #720                     INP1B                   RESET SWITCH INP1B     P
 015720 71 777777 00 000000           007850 INP15X   TC  777777                                           EXIT                   E
                                       007860          RMK                                                  MOVE COMMENT FROM      T
                                       007870          RMK                                                  INPUT TO W/A AND       C
                                       007880          RMK                                                  ADJUST AMS             C
 015730 25 000243 00 016170           007890 MVCOM    TCT $STP                     MVCOMX                  SET EXIT               P
 015740 72 005257 00 600000           007900 MV1      SET IPCOM,R                  $T                      RIGHT JUSTIFY          P
 015750 35 005144 00 005257           007910          SCR IPCON                    IPCOM,R                  COMMENTS FIELD        C
 015760 61 015770 00 016170           007920          CTC * 1                      MVCOMX                  FIELD HAVE ANYTHING    D
 015770 73 000133 00 600000           007930          STR $3                       $T                      STORE T AS LHE         P
 016000 44 000133 00 017763           007940          TCA $3                       #000001#,I              LOCATE FIRST           P
 016010 72 000000 30 600000           007950          SET 000000                30 $T                       NON-SPACE CHARACTER   C
 016020 43 017760 00 017760           007960          SC  #01#                     #01#                     FROM LEFT END         C
 016030 61 016050 00 016050           007970          CTC * 2                      * 2
 016040 71 016000 00 000000           007980          TC  *-4
 016050 25 015740 00 016070           007990          TCT MV1                      MV2                     GET LENGTH OF FIELD    P
 016060 45 016073 00 000133           008000          TCS MV2,3                    $3                      AND SET T TO RHE       C
 016070 72 777777 50 600000           008010 MV2      SET 777777                50 $T                      OF RECEIVING AREA      C
 016100 73 016203 00 600000           008020          STR MV3,R                    $T                      WILL COMM W/A          D
 016110 45 016203 00 016207           008030          TCS MV3,R                    MV4,R                   OVERFLOW               C
 016120 61 016170 00 016130           008040          CTC MVCOMX                   * 1                                            Z
 016130 24 000000 30 005257           008050          STC 000000                30 IPCOM,R                 MOVE TO COMM W/A       P
 016140 44 000153 00 016073           008060          TCA $5                       MV2,3                   CREATE SPACE TO        P
 016150 44 000153 00 017767           008070          TCA $5                       #000002#,I              RIGHT OF FIELD AND     C
 016160 34 777777 55 777777           008080          SCC 777777                55 777777                  ADJUST AMS TO MSC      C
                                       008090          RMK                                                  OF NEXT FIELD          C
 016170 71 777777 00 000000           008100 MVCOMX   TC  777777                                           EXIT                   E
```

```
NAME  AUTOFLOW                                           INDEX    40000501          SEG 02  DATE 111665        PAGE 019
  HSK    OP    A     N     B      SEQ #  TAG      OP  A ADDRESS                N   B ADDRESS            COMMENTS
 016201                           008110 MV3      DAR 3,1
 016204 01 005101 01 010101       008120 MV4      DAC MACOM,R
                                  008130          RMK                                                 EXTRACT TAG AND      T
                                  008140          RMK                                                 LABEL FROM           C
                                  008150          RMK                                                 COMMENTS FIELD,      C
                                  008160          RMK                                                 PRP=LABEL PRESENT    C
                                  008170          RMK                                                 PRZ=NO TAG           C
                                  008180          RMK                                                 PRN=NO LABEL         C
 016210 25 000243 00 016650       008190 PAREN    TCT $STP                         PARENX             SET EXIT             P
 016220 34 005117 00 005123       008200          SCC HDLAB                        HDLAB,R            CLEAR HOLD AREAS     P
 016230 34 005110 00 005116       008210          SCC HDTAG                        HDTAG,R
 016240 25 000173 00 003775       008220          TCT $7                           TEMP1              PRESERVE AMT         P
 016250 25 005130 00 000173       008230          TCT RHECOM                       $7                 RHE COMM TO AMT      P
 016260 72 000173 00 600000       008240          SET 000173                       $T                 COMMENT FIELD        D
 016270 43 000151 00 000153       008250          SC  000151                       000153               EXHAUSTED          C
 016300 61 016310 00 016320       008260          CTC *,1                          * 2
 016310 71 016660 00 000000       008270          TC  PAR3                                                               Z
 016320 72 150001 00 600000       008280          SET 150001                       $T                 LOCATE RIGHT END     P
 016330 31 000001 57 000000       008290          LNS 000001                  57   000000              OF ASTERISK FIELD   C
 016340 61 016350 00 016660       008300          CTC * 1                          PAR3               RIGHT END FOUND      D
 016350 25 000223 00 000113       008310          TCT $STA                         $1                 RHE TO AM1           P
 016360 72 350001 00 600000       008320          SET 350001                       $T                 LOCATE COMMA         P
 016370 31 000001 51 000001       008330          LNS 000001                  51   000001             WITHIN FIELD     C
 016400 61 016700 00 016700       008340          CTC PAR4                         PAR4               WAS COMMA FOUND      D
 016410 25 000223 00 000173       008350          TCT $STA                         $7
 016420 22 020045 00 016643       008360          OCT #04#                         PAR2,3             SET SIGNAL FOR PRP   P
 016430 72 005123 00 600000       008370          SET HDLAB,R                      $T                 MOVE LABEL TO        P
 016440 24 000002 77 000006       008380          STC 000002                  77   000006             HOLD AREA, LEFT      C
 016450 72 150001 00 600000       008390          SET 150001                       $T                 JUSTIFY, AND         C
 016460 31 005117 00 005123       008400          LNS HDLAB                        HDLAB,R            CLEAR TO RIGHT       C
 016470 61 016500 00 016510       008410          CTC * 1                          * 2
 016500 34 000001 20 005123       008420          SCC 000001                  20   HDLAB,R
 016510 72 005116 00 600000       008430 PAR1     SET HDTAG,R                      $T                 MOVE TAG TO          P
 016520 24 000001 55 000007       008440          STC 000001                  55   000007             HOLD AREA, LEFT      C
 016530 72 350001 00 600000       008450          SET 350001                       $T                 JUSTIFY, AND         C
 016540 31 005110 00 005116       008460          LNS HDTAG                        HDTAG,R            CLEAR TO RIGHT       C
 016550 61 016560 00 016570       008470          CTC * 1                          * 2
 016560 34 000001 20 005116       008480          SCC 000001                  20   HDTAG,R
 016570 34 000000 51 000001       008490          SCC 000000                  51   000001             CLEAR ASTERISK FIELD P
 016600 25 000113 00 000153       008500          TCT $1                           $5                 ADJUST AMS TO NEW    P
 016610 44 000153 00 017763       008510          TCA $5                           $000001#$1         LHE OF COMM FIELD    C
 016620 22 017757 05 777777       008520          OCT #74#                    05   777777             ISS TO MSC COMM      P
 016630 25 003775 00 000173       008530          TCT TEMP1                        $7                 RESTORE AMT          P
 016640 72 000077 00 100000       008540 PAR2     SET 000077                       $PRI               SET PRIS             P
 016650 71 777777 00 000000       008550 PARENX   TC  777777                                          EXIT                 E
                                  008560          RMK                                                 NO ASTERISK FIELD    T
 016660 22 020002 00 016643       008570 PAR3     OCT #02#                         PAR2,3             SET SIGNAL FOR PRZ   P
 016670 71 016630 00 000000       008580          TC  PAR2-1                                                  *PAR2*   J
                                  008590          RMK                                                 NO LABEL IN FIELD    T
 016700 22 017760 00 016643       008600 PAR4     OCT #01#                         PAR2,3             SET SIGNAL FOR PRN   P
 016710 22 020140 01 000001       008610          OCT #35#                    01   000001             SET COMMA OVER STAR
 016720 71 016510 00 000000       008620          TC  PAR1                                                                 J
                                  008630          RMK                                                 ENTER NEW CHAIN      T
                                  008640          RMK                                                 IN CHAIN TABLE       C
 016730 25 000243 00 017050       008650 CHENT    TCT $STP                         CHENTX             SET EXIT             P
 016740 25 004040 00 000133       008660          TCT CHLOC                        $3                 PICK UP LSC OF TABLE P
 016750 25 004050 03 000003       008670          TCT SEQNO                   03   000003             SEQ # TO TABLE       P
 016760 25 005124 03 000007       008680          TCT LCNT                    03   000007             # LINES TO TABLE     P
```

```
NAME  AUTOFLOW                                        INDEX     4000 501        SEG 02  DATE 111665    PAGE 020
  HSM    OP     A    M    B        SEQ #  TAG      OP  A ADDRESS            N   B ADDRESS           COMMENTS
 016770 22 017761 03 000000        008690          OCT #103               03 000000             SET IND TO GO            P
 017000 22 017761 03 000014        008700          OCT #103               03 000014             1ST RCD INDIC OF  P
                                   008710          RMK                                          NEXT CHAIN TO NOP  C
 017010 44 004043 00 020027        008720          TCA CHLOC,R               #000010#,1         ADVANCE LSC OF TABLE     P
 017020 44 004037 00 017763        008730          TCA NUMCH,R               #000001#,1         ADD TO CHAIN COUNTER     P
 017030 25 020043 00 005124        008740          TCT #000000#,1            LNCNT              RESET LINE CTR           P
 017040 22 007246 00 007245        008750          OCT FP5,6                 FP5,5              SET SWITCH FP5           P
 017050 71 777777 00 000000        008760 CHENTX   TC  777777                                   EXIT                     E
                                   008770          RMK                                          EDIT LINE                T
                                   008780          RMK                                          SUBROUTINE-MOVE A        C
                                   008790          RMK                                          LINE FROM I/P TO         C
                                   008800          RMK                                          O/P.  B ADDRESS          C
                                   008810          RMK                                          OF TC CONTAINS           C
                                   008820          RMK                                          NO OF CHAR IN LINE       C
                                   008830          RMK                                          . PRZ-END OF FIELD       C
                                   008840          RMK                                          PRN-NOT END FIELD        C
                                   008850          RMK                                          PRP-WORD TOO BIG         C
 017060 25 000243 00 000113        008860 EDLIN    TCT $STP                  $1                 STP TO AM1               P
 017070 25 777777 10 000133        008870          TCT 777777             10 $3                 PICK UP LINE LENGTH      P
 017100 44 000133 00 000153        008880          TCA $3                    $5                 AM3 TO MAX RIGHT END     P
 017110 72 770001 00 600000        008890          SET 770001                $T                 END MESSAGE WITHIN       D
 017120 31 000000 53 000000        008900          LNS 000000             53 000000             LIMITS OF LINE           C
 017130 61 017140 00 017250        008910          CTC * 1                   ED4
                                   008920          RMK                                          LAST LINE OF COMMENT     N
 017140 25 000223 00 000133        008930 ED1      TCT $STA                  $3                 COMPUTE TRUE LENGTH      P
 017150 45 000133 00 000153        008940          TCS $3                    $5                   OF FINAL LINE          C
 017160 44 000153 00 000133        008950          TCA $5                    $3                 SET RIGHT AND LEFT       P
 017170 44 000173 00 000133        008960          TCA $7                    $3                   LIMITS OF LINE         C
 017200 21 000000 57 000000        008970          IT  000000             57 000000             MOVE LINE TO O/P         P
 017210 22 020002 00 017233        008980          OCT #02#                  ED3,3              SET SIGNAL FOR PRZ       C
 017220 44 003013 00 017763        008990 ED2      TCA LINES,R               #000001#,1         ADD TO LINE CTR          P
 017230 72 000077 00 100000        009000 ED3      SET 000077                $PRI               SET PRI INDICATOR        P
 017240 71 000000 10 000000        009010 EDLINX   TC  000000             10                    EXIT                     F
                                   009020          RMK                                          NOT LAST LINE            N
 017250 25 777777 10 017263        009030 ED4      TCT 777777             10 * 1,3              SET LOCATION OF          P
 017260 72 770001 50 600000        009040          SET 770001             50 $T                 RIGHTMOST CHARACTER      C
 017270 43 017760 00 017760        009050          SC  #01#                  #01#               IS CHARACTER A SPACE     D
 017300 61 017410 00 017410        009060          CTC ED5                   ED5
 017310 73 000153 00 600000        009070          STR $5                    $T
 017320 22 017757 05 000001        009080          OCT #74#               05 000001             ISS OVERLAYS SPACE       P
 017330 44 000153 00 017767        009090          TCA $5                    #000002#,1         SET AMS AND AM7 TO       P
 017340 44 000173 00 017263        009100          TCA $7                    ED4 1,3            TO LINE BOUNDARIES       C
 017350 21 777776 57 777777        009110          IT  777776             57 777777             MOVE LINE TO O/P         P
 017360 44 000173 00 017763        009120          TCA $7                    #000001#,1
 017370 22 017760 00 017233        009130          OCT #01#                  ED3,3              SET SIGNAL FOR PRN       P
 017400 71 017220 00 000000        009140          TC  ED2                                                               J
 017410 66 017260 00 017263        009150 ED5      TA  ED4 1                 ED4 1,3            SHIFT LEFT BY 1 CHAR     P
                                   009160          RMK                                                   *ED4*           J
 017420 22 020045 00 017233        009170 ED6      OCT #C4#                  ED3,3              SET PRP
 017430 71 017230 00 000000        009180          TC  ED3
                                   009190          RMK                                          SKIP EXTRA SPACES
                                   009200          RMK                                          TO RIGHT OF PAREN
                                   009210          RMK                                          FIELD IN COMMENTS
 017440 25 000243 00 017530        009220 SKS#     TCT $STP                  SKSPX              SET EXIT
 017450 72 000000 50 600000        009230 SKSP1    SET 000000             50 $T
 017460 43 017760 00 017760        009240          SC  #01#                  #01#               CURRENT LINE
 017470 61 017520 00 017520        009250          CTC SKSP2                 SKSP2              A SPACE
 017500 44 000153 00 017763        009260          TCA $5                    #000001#,1         ADVANCE LNE BY ONE
```

```
NAME  AUTOFLOW                                          INDEX     40000501            SEG 02  DATE 111665      PAGE 021
  HSM     OP    A      N.    B        SEQ #  TAG        OP  A ADDRESS            N    B ADDRESS*              COMMENTS
 017510 71 017450 0C 000000        009270             TC  SKSP1                                             RECYCLE
 017520 22 017757 05 777777        009280 SKSP2       OCT #74#                   ,5 777777                  PUT IN FIELD MARK
 017530 71 777777 00 000000        009290 SKSPX       TC  777777                                            EXIT
 017540                            009300             DAR 96                                                FOR PATCHES
 017700                                   #0006       CON 22,0
         01010101010101010101010101010101010101010101
 017730                                   #0007       CON 22,0                        0000000000000000000000
         23232323232323232323232323232323232323232323
 017756                                   #0008       CON 01                     #    76
 017757                                   #0009       CON 01                     #    74
 017760                                   #0010       CON 01                     #    01
 017761                                   #0011       CON 03,1                   #    000001
 017764                                   #0012       CON 01                     #    77
 017765                                   #0013       CON 03,1                   #    000002
 017770                                   #0014       CON 04                     #    74010177
 017775                                   #0015       CON 03,1                   #    000003
 020000                                   #0016       CON 02                          BT
         4163
 020002                                   #0017       CON 01                     #    02
 020003                                   #0018       CON 01                     #    51
 020004                                   #0019       CON 01                     #    41
 020005                                   #0020       CON 01                     #    43
 020006                                   #0021       CON 01                     #    44
 020007                                   #0022       CON 01                     #    47
 020010                                   #0023       CON 01                     #    55
 020011                                   #0024       CON 01                     #    57
 020012                                   #0025       CON 01                     #    62
 020013                                   #0026       CON 01                     #    66
 020014                                   #0027       CON 01                     #    63
 020015                                   #0028       CON 01                     #    75
 020021                                   #0029       CON 03,1                   #    000012
 020025                                   #0030       CON 03,1                   #    000010
 020030                                   #0031       CON 01                          J
         51
 020031                                   #0032       CON 03,1                   #    001000
 020035                                   #0033       CON 03,1                   #    000005
 020041                                   #0034       CON 03,1                   #    000000
 020044                                   #0035       CON 01                     #    03
 020045                                   #0036       CON 01                     #    04
 020051                                   #0037       CON 03,1                   #    000017
 020055                                   #0038       CON 03,1                   #    000014
 020060                                   #0039       CON 02                     #    7474
 020065                                   #0040       CON 03,1                   #    000015
 020070                                   #0041       CON 02                          1515
 020075                                   #0042       CON 03,1                   #    000020
 020101                                   #0043       CON 03,1                   #    000036
 020104                                   #0044       CON 05                     #    5147444163
 020111                                   #0045       CON 02                          %T
         5563
 020113                                   #0046       CON 01                     #    40
 020114                                   #0047       CON 01                          72
 020115                                   #0048       CON 03                          #HK
         615452
 020120                                   #0049       CON 02                     #    0101
 020122                                   #0050       CON 02                          DC
         4342
 020124                                   #0051       CON 01                          71
 020125                                   #0052       CON 01                          C
```

```
NAME  AUTOFLOW                                          INDEX     4000 501            SEG 02  DATE 111665      PAGE 022
  HSM     OP    A      N.    B        SEQ #  TAG        OP  A ADDRESS            N    B ADDRESS               COMMENTS
         42
 020126                                   #0053       CON 09                          TA TC CTG
         634001634201426342
 020137                                   #0054       CON 01                          Y
         70
 020140                                   #0055       CON 01                          35
 020150 00 000000 0C 000002                           CON 8,1                         0000000000000002
```

```
NAME  AUTOFLOW                               INDEX     40000501         SEG 03  DATE 111665     PAGE 023
  HSM    OP      A     N      B       SEQ #  TAG        OP  A ADDRESS           N   B ADDRESS       COMMENTS
                                      009310           SEG 03                        START           PASS TWO                 R
                                      009320           RMK                                           SUBROUTINE TO            T
                                      009330           RMK                                           LOCATE GIVEN             C
                                      009340           RMK                                           TAG IN TAG TABLE.        C
                                      009350           RMK                                           *PRZ SET IF OK,           C
                                      009360           RMK                                           PRN IF NO TAG.           C
                                      009370           RMK                                           PRP IF NO ENTRY          C
                                      009380           RMK                                           I/P=AMS AT MSC TAG
                                      009390           RMK                                           O/P=AMS TO TBL LOC       C
004170 25 000243 00 004240            009400 LOCTAG    TCT $STP                      LOCTAGX         SET EXIT                 P
004200 72 000000 30 600001            009410           SET 000000               30 600001           -IS FIELD EMPTY          D
004210 43 004207 00 004207            009420           SC  *=1,7                     *=1,7
004220 61 004250 00 004250            009430           CTC LOC1                      LOC1
004230 72 000001 00 100000            009440           SET 000001                    $PRI            NO TAG = SET PRN         P
004240 71 777777 00 000000            009450 LOCTAGX   TC  777777                                    EXIT                     F
004250 72 004012 00 600000            009460 LOC1      SET TEMP3,R                   $T              MOVE TAG TO              P
004260 24 000000 33 000006            009470           STC 000000               33 000006              WORK AREA              C
004270 25 004020 00 004001            009480           TCT NUMTAG                    TEMP2           MOVE TALLY TO W/A         P
004300 25 004030 00 000153            009490           TCT TAIN                      $5              AMS TO START TABLE       P
004310 66 004340 00 004001            009500 LOC2      TA  LOC3                      TEMP2           END OF TAG TABLE          D
004320 72 000004 00 100000            009510           SET 000004                    $PRI            NO ENTRY=SET PRP         P
004330 71 004240 00 000000            009520           TC  LOCTAGX                                   TO EXIT                  J
004340 72 000007 50 600000            009530 LOC3      SET 000007               50 $T              COMPARE TAG TO            D
004350 43 004004 00 004012            009540           SC  TEMP3                     TEMP3,R           THIS ENTRY             C
004360 61 004410 00 004410            009550           CTC LOC4                      LOC4                                   O
                                      009560           RMK                                           TAG FOUND                R
004370 72 000002 00 100000            009570           SET 000002                    $PRI            SET PRZ                  P
004400 71 004240 00 000000            009580           TC  LOCTAGX                                   TO EXIT                   J
004410 44 000153 00 004017            009590 LOC4      TCA $5                        TENTL,R         INCREMENT AMS            P
004420 71 004310 00 000000            009600           TC  LOC2                                                                J
004430                                009610 LOAD4     CON 8,10                      &LOAD AD
       7453564043014043
004440                                009620 PACN      CON 4                    #  74232401
004444                                009630 BXCN      CON 4                    #  74232301
004450 01 023200 01 010101            009640 STOR3     DAC IPAR1                                     USED AS STORAGE
004454 01 023510 01 010101            009650 STOR4     DAC IPAR2                                       IN READ RTN
004460 01 000001 01 010101            009660 COLNO     DAC 000001                                    COLUMN #
004464 01 000000 01 010101            009670 TPOS      DAC 000000                                    SEQ # OF LAST RCD
004471                                009680 TSO       DAR 3,1                                       DEST TAG SEQ #
004474 01 000001 01 010101            009690 LNC1      DAC 000001                                    ADDRESS=COL1
004500 01 000001 01 010101            009700 LNC2      DAC 000001                                    ADDRESS=COL2
004504 01 777777 01 010101            009710 LNC3      DAC 777777                                    TEMP STORAGE TO
                                      009720           RMK                                           HOLD ADDRESS
004510 01 000006 01 010101            009730 DECAD     DAC 000006                                    USED IN LOCATING
                                      009740           RMK                                           COLUMN 2 ADDRESS
004515                                009750 STOR5     DAR 3,1                                       STORAGE FOR INFO
004521                                009760 STOR6     DAR 3,1                                       RELATING TO = GAPS
004525                                009770 STOR7     DAR 3,1
004531                                009780 STOR8     DAR 3,1
004535                                009790 TPOS1     DAR 3,1                                       STORAGE FOR TPOS
004541                                009800 SPST1     DAR 3,1
004545                                009810 SPST2     DAR 3,1
004550                                009820 EVOD      CON 1                         00
                                      009830           RMK                                           40 IF EVEN COL
004554                                009840 PSRCD     DAR 1,20                                      PSEUDO COMM RCD
                                      009850           RMK                                           BEGIN PASS 2             R
004600 22 004055 00 004615            009860 STPAS2    OCT TT2                       * 1,5           REWIND INPUT AND         P
004610 17 000000 00 000000            009870           RWD                                           OUTPUT TAPES             C
```

```
NAME  AUTOFLOW                                          INDEX     4000-501          SEG 03  DATE 111665    PAGE 024
  HSM    OP    A     N    B       SEQ #   TAG    OP   A ADDRESS            N   B ADDRESS       COMMENTS
004620 22 004056 00 004635      009880          OCT  TT3                        * 1,5
004630 17 000000 00 000000      009890          RND
004640 25 004044 00 000113      009900          TCT  CHI*                       $1              SET AMI TO START        P
                                009910          RMK                                             OF CHAIN TABLE          C
004650 22 011672 00 004554      009920          OCT  #76#                       PSRCD           SET UP PSEUDO
004660 22 011673 00 004570      009930          OCT  #75#                       PSRCD,12        CONN RECORD
004670 22 011674 00 004555      009940          OCT  #67#                       PSRCD,1         X IN 2ND POSITION
                                009950          RMK                                             MEANS PSEUDO CONN
004700 75 000000 00 010000      009960          CSG                             010000
                                009970          RMK                                             BEGIN MAIN LOOP         N
004710 22 004055 00 004725      009980  SP1     OCT  TT2                        * 1,5           READ RECORD             P
004720 15 003010 00 000000      009990          BRF  OFAR
004730 62 004730 00 004730      010000          SSM  *                          *
004740 72 003010 00 600000      010010          SET  OFAR                       $T              END FILE                D
004750 43 011675 00 011675      010020          SC   #72#                       #72#
004760 61 000240 00 005000      010030          CTC  000240                     * 2
004770 71 007240 00 000000      010040          TC   ENDPZ
005000 25 003054 00 004464      010050          TCT  OPSQ                       TPOS            STORE SEG #             Z P
005010 72 011676 00 600000      010060          SET  #41#                       $T              B CODE                  D
005020 43 003024 00 003024      010070          SC   CODE                       CODE
005030 61 005050 00 005050      010080          CTC  * 2                        * 2
005040 71 006730 00 000000      010090          TC   SP40
005050 25 004474 00 004504      010100          TCT  LNC1                       LNC3            SET SYMBOL ADDRESS      Z P
005060 44 004477 00 003013      010110  SP1A    TCA  LNC1,R                     LINES,R         ADD TO LINE CTR         P
005070 72 004477 00 600000      010120          SET  LNC1,R                     $T              ROOM IN COLUMN          D
005100 43 004115 00 004117      010130          SC   EPCON,R=2                  EPCON,R           FOR THIS RECORD       C
005110 61 005120 00 005500      010140          CTC  * 1                        SP10
005120 72 011677 00 600000      010150          SET  "T"                        $T              T CODE                  D
005130 43 003024 00 003024      010160          SC   CODE                       CODE
005140 61 005160 00 005160      010170          CTC  * 2                        * 2
005150 71 005270 00 000000      010180          TC   SP4
005160 51 004447 00 011701      010190  SP2     DA   BXCNTR                     #7424#          INCREASE BOX #          Z P
005170 71 010300 00 000000      010200          TC   BOP                                        MOVE BOX # TO O/P       P
005200 25 004104 00 000133      010210          TCT  ATAG                       $3              ADDR OF TAG TO AM3      P
005210 71 004170 00 000000      010220          TC   LOCTAG                                     FIND TAG IN TABLE       $
005220 61 005270 00 005270      010230          CTC  SP4                        SP4             ENTRY FOUND IN TABLE    D
005230 72 000017 50 600000      010240  SP3     SET  000017                50   $T              PUT BOX # IN            P
005240 26 003020 00 003023      010250          STT  BOXNO                      BOXNO,R           TAG TABLE             C
005250 25 004504 05 000023      010260          TCT  LNC3                  05   000023          PUT F/C ADDRESS         P
005260 22 004463 05 000020      010270          OCT  COLNO,R               05   000020          IN TAG TABLE            C
005270 25 004504 00 003017      010280  SP4     TCT  LNC3                       ADDR,R          PUT F/C ADDRESS         P
005300 22 004463 00 003014      010290          OCT  COLNO,R                    ADDR              IN RECORD             C
005310 22 003024 00 005321      010300          OCT  CODE                       * 1,1           D OR J CODE             D
005320 72 000001 00 600000      010310          SET  000001                     $T
005330 31 011702 00 011704      010320          LNS  "DJS"                      "DJS"
005340 61 005350 00 005360      010330          CTC  * 1                        SP5
005350 71 007050 00 000000      010340          TC   DJ
005360 72 011705 00 600000      010350  SP5     SET  #43#                       $T              DECISION                Z D
005370 43 003024 00 003024      010360          SC   CODE                       CODE
005400 61 005420 00 005420      010370          CTC  * 2                        * 2
005410 71 006000 00 000000      010380          TC   SP20
005420 71 010360 00 000000      010390  SP8     TC   OPTZ                                       WRITE RECORD TO O/P     Z $
005430 22 003024 00 005441      010400  SP9     OCT  CODE                       * 1,1           J OR E CODE             D
005440 72 770001 00 600000      010410          SET  770001                     $T
005450 31 011706 00 011707      010420          LNS  #5144#                     #5144#
005460 61 005470 00 004710      010430          CTC  * 1                        SP1
005470 71 005610 00 000000      010440          TC   SP12                                                               J
                                010450          RMK                                             IF NOT A JUMP, GO
```

```
NAME  AUTOFLO*                                 INDEX    4000*50*        SEG 03  DATE 111665    PAGE 025
  HSM   OP    A    *     *      SEQ #  TAG     OP  A ADDRESS               B ADDRESS          .COMMENTS
                                010460         RMK                                             PROCESS NEXT RCD
                                010470         RMK                                             CURRENT RCD WILL    C
                                010480         RMK                                             NOT FIT IN COL      C
 005500 22 003024 00 005511     010490 SP1A    OCT CODE                   * 1,1                J, H, OR E CODE     C
 005510 72 770001 00 600000     010500         SET 770001                 $T
 005520 31 011710 00 011712     010510         LNS #514744#               #514744#
 005530 61 005540 00 005550     010520         CTC * 1                    * 2                  PRN = NOT J, H, E
 005540 71 005160 00 000000     010530         TC  SP2                                                             Z
 005550 22 004550 00 005565     010540         OCT EVOD                     1,5                IF EVEN COLUMN, GO
 005560 72 000240 00 000000     010550         SET 000240
 005570 71 007470 00 000000     010560 SP11    TC  E DCOL                                      END OF COLUMN CODING S
 005600 71 005060 00 000000     010570         TC  SP1A                                                            J
                                010580         RMK                                             J OR E CODE         T
 005610 22 004550 00 005625     010590 SP1Z    OCT EVOD                     *5                 IS THIS AN EVEN      D
 005620 72 006600 00 000000     010600         SET SP34                                        COLUMN  NO*SP34,YES* C
                                010610         RMK                                             FALL THRU IF ODD
                                010620         RMK                                             POSITION TAPE FOR   N
                                010630         RMK                                             NEXT MAIN CHAIN     C
 005630 44 000113 00 011717     010640 SP125   TCA $*                     #000010#,1           ADVANCE TO NEXT      P
                                010650         RMK                                             ENTRY IN CHAIN TABLE C
 005640 22 000000 10 005655     010660         OCT 000000              1C * 1,5                THIS CHAIN BEEN USED D
 005650 72 005630 00 000000     010670         SET SP125                                            NO*SP125,YES*  C
 005660 25 777773 10 005710     010680         TCT 777773              10 SP127                COMPUTE # GAPS      P
 005670 45 005713 00 004467     010690         TCS SP127,3                TPOS,R               TO THIS CHAIN FROM  C
 005700 22 004055 00 005715     010700         OCT TT2                    * 1,5                PRESENT POSITION    C
 005710 06 777777 00 000000     010710 SP127   UNS 777777                 000000               POSITION I/P TAPE   P
                                010720         RMK                                             SEE IF IT PAYS      N
                                010730         RMK                                             TO PUT NEW CHAIN    C
                                010740         RMK                                             IN THIS COLUMN      C
 005720 25 004114 00 003775     010750 SP13    TCT EFCON                  TEMP1                MORE THAN 20 LINES  D
 005730 45 003777 00 004477     010760         TCS TEMP1,R                LOC1,R                 LEFT IN COLUMN    C
 005740 45 003777 00 011723     010770         TCS TEMP1,R                #000024#,1
 005750 61 004710 00 005760     010780         CTC SP1                    SP14
 005760 71 007470 00 000000     010790 SP14    TC  E DCOL                                      END OF COLUMN CODING ZS
 005770 71 004710 00 000000     010800         TC  SP1                                         GET NEW RECORD      J
                                010810         RMK                                             BEGIN HANDLING OF   T
                                010820         RMK                                             DECISION RECORDS    C
 006000 22 004550 00 006015     010830 SP20    OCT EVO                    * 1,5                ADJACENT COLUMN      D
 006010 72 005420 00 000000     010840         SET SP8                    000000                                    Z
 006020 72 011724 00 600000     010850         SET #*3#                   $T                   COLUMN 4             D
 006030 43 004463 00 004463     010860         SC  COL*0,R                COL*0,R
 006040 61 005420 00 006050     010870         CTC SP8                    * 1                                       Z
                                010880         RMK IF ALLREADY IN ADJ     COL, OR COL 4, MUST  NOT USE ADJ COL NOW  N
 006050 25 004074 00 000133     010890         TCT ALHT                   $3                   SET AM3 TO 1SC OF    P
                                010900         RMK                                             1ST DESTINATION TAG  C
 006060 71 010470 00 000000     010910         TC  SPOL                                        DETERMINE STATUS OF  S
                                010920         RMK                                             1ST DESTINATION TAG  C
 006070 61 006350 00 006350     010930         CTC SP27                   SP27                 1ST TAG SUITABLE FOR D
                                010940         RMK                                             ASSIGNMENT TO COL 2  C
 006100 25 004515 00 004521     010950 SP21    TCT STOR5                  STOR6                PRESERVE # GAPS      P
 006110 25 000173 00 004531     010960         TCT $7                     STOR8                TO REACH LHT AND     C
                                010970         RMK                                             LOCATION IN CHTBLE   C
 006120 25 004100 00 000133     010980         TCT ALHT                   $3                   SET AM3 TO 1SC OF    P
                                010990         RMK                                             2ND DESTINATION TAG  C
 006130 71 010470 00 000000     011000         TC  SPOL                                        DETERMINE STATUS OF  S
                                011010         RMK                                             2ND DESTINATION TAG  C
 006140 61 006420 00 006420     011020         CTC SP27                   SP29                 2ND TAG SUITABLE FOR D
                                011030         RMK                                             ASSIGNMENT TO COL2   C
```

```
NAME  AUTOFLOW                                 INDEX     4000'501        SEQ 03  DATE 111665   PAGE 026
  HSM   OP     A    M     B     SEQ #  TAG     OP  A ADDRESS        N   B ADDRESS          COMMENTS
                                011040         RMK                                         BOTH TAGS SUITABLE   N
                                011050         RMK                                           FOR COL TWO.       C
                                011060         RMK                                         FIND CLOSEST OF      C
                                011070         RMK                                           THE TWO CHAINS     C
006150 72 004523 00 600000      011080 SP22    SET STOR6,R              $T                 WHICH TAG IS CLOSER  D
006160 43 004515 00 004517      011090         SC  STOR5                STOP5,R                    1ST=SP25,2ND= C
006170 61 006200 00 006270      011100         CTC * 1                  SP25               PR=RHT IS CLOSER
                                011110         RMK                                         USE TAG 1 FOR COL 2  N
006200 25 004100 00 000133      011120 SP23    TCT A.HT                 $3                 SET 2ND TAG FOR      P
006210 22 011725 03 777777      011130         OCT #C2#             03  777777
006220 25 004521 00 004525      011140         TCT STOR6                STOR7                CONNECTOR          C
006230 25 004074 00 000133      011150 SP24    TCT ALHT                 $3                 SET 1ST TAG FOR      P
006240 22 011726 03 777777      011160         OCT #C4#             03  777777               COLUMN 2           C
006250 25 004531 00 000173      011170         TCT STOR8                $7                 SET LOC IN CHTBLE    P
006260 71 006440 00 000000      011180         TC  SP30                                                         J
                                011190         RMK                                         USE TAG 2 FOR COL 2  N
006270 25 004074 00 000133      011200 SP25    TCT ALHT                 $3                 SET 1ST TAG FOR      P
006300 22 011725 03 777777      011210         OCT #C2#             03  777777
006310 25 004515 00 004525      011220         TCT STOR5                STOR7                CONNECTOR          C
006320 25 004100 00 000133      011230 SP26    TCT ARHT                 $3                 SET 2ND TAG FOR      P
006330 22 011726 03 777777      011240         OCT #C4#             03  777777               COLUMN 2           C
006340 71 006440 00 000000      011250         TC  SP30                                                         J
                                011260         RMK                                         1ST TAG NOT SUITABLE N
                                011270         RMK                                         FOR COL 2, TRY 2ND   C
006350 25 004100 00 000133      011280 SP27    TCT ARHT                 $3                 MSC 2ND TAG TO AM3   P
006360 71 010470 00 000000      011290         TC  SCOL                                    DETERMINE STATUS OF  S
                                011300         RMK                                         2ND DESTINATION TAG  C
006370 61 005420 00 005420      011310         CTC SP8                  SP8                2ND TAG SUITABLE FOR D
                                011320         RMK                                         ASSIGNMENT TO COL 2  C
                                011330         RMK                                         USE TAG 2 FOR COL 2  N
006400 25 004515 00 004525      011340 SP28    TCT STOR5                STOR7              SET # GAPS           P
006410 71 006320 00 000000      011350         TC  SP26                                                         J
                                011360         RMK                                         USE TAG 2 FOR COL 2  N
006420 25 004521 00 004525      011370 SP29    TCT STOR6                STOR7              SET # OF GAPS        P
006430 71 006230 00 000000      011380         TC  SP24                                                         J
                                011390         RMK                                         LOCATE, PROCESS      T
                                011400         RMK                                         COMING FOR COL 2.    C
                                011410         RMK                                         AM3 HAS MSC OF       C
                                011420         RMK                                         DEST TAG,AM7 HAS     C
                                011430         RMK                                         LOC IN CHAIN TABLE   C
006440 25 004464 00 004535      011440 SP30    TCT TPOS                 TPOS1              STORE SEQ # OF       P
                                011450         RMK                                         LAST RECORD READ     C
006450 71 010360 00 000000      011460         TC  OPT2                                    WRITE DECISION RCD   P
                                011470         RMK                                           TO OUTPUT FILE     C
006460 25 004525 00 006500      011480         TCT STOR7                SP31               PUT # GAPS INTO      P
006470 22 004055 00 006505      011490         OCT TT2                  * 1,5                TAPE INSTRUCTION   C
006500 06 777777 00 000000      011500 SP31    UNS 777777               000000             UNWIND GAPS TO       P
                                011510         RMK                                         REACH CHAIN          C
006510 22 011727 07 000000      011520         OCT #40#             07  000000             SET INDICATOR IN     P
                                011530         RMK                                         TABLE THAT THIS      C
                                011540         RMK                                         CHAIN IS PROCESSED   C
006520 25 004460 00 004541      011550         TCT COLNO                SPST1              STORE MAIN COLUMN #  P
006530 44 004463 00 011733      011560         TCA COLNO,R              #000001#,I         SET COL # TO         P
                                011570         RMK                                         EITHER 2 OR 4        C
006540 25 004474 00 004545      011580 SP32    TCT LOC1                 SPST2              STORE CURRENT ADDR,  P
                                011590         RMK                                           IN MAIN COLUMN     C
006550 45 004477 00 004513      011600         TCS LOC1,R               DECAD,R            SET ADDRESS FOR      P
                                011610         RMK                                         EVEN COLUMN TO 3     C
```

```
NAME  AUTOFLOW                                  INDEX     40000503            SEG 03  DATE 111665      PAGE 027
  HSM    OP     A    N     B       SEQ #  TAG     OP  A ADDRESS          N   B ADDRESS             COMMENTS
                                   011620         RMK                                              LINES BELOW MIDDLE    C
                                   011630         RMK                                              OF DECISION SYMBOL    C
 006560 22 011727 00 004550        011640         OCT #40#                   EVOC                  SET EVEN/ODD          P
                                   011650         RMK                                              INDICATOR TO EVEN     C
                                   011660         RMK                                              GO TO MAIN PATH       N
                                   011670         RMK                                              AND PROCESS CHAIN     C
                                   011680         RMK                                              FOR COLUMN TWO        C
 006570 71 004710 00 000000        011690 SP33    TC  SP1                                                                J
                                   011700         RMK                                              RETURN HERE FROM      T
                                   011710         RMK                                              PROCESSING COLUMN     C
                                   011720         RMK                                              TWO CHAIN-VIA SP12    C
 006600 22 011715 00 004550        011730 SP34    OCT #00#                   EVOD                  RESET EVEN/ODD        P
                                   011740         RMK                                              INDICATOR TO ODD      C
 006610 25 004474 00 004500        011750         TCT LNC1                   LNC2                  STORE CURRENT         P
                                   011760         RMK                                              ADDRESS OF EVEN       C
                                   011770         RMK                                              COL FOR LATER TEST    C
 006620 25 004545 00 004474        011780         TCT SPST2                  LNC1                  RESTORE ADDRESS        P
                                   011790         RMK                                                OF MAIN COLUMN      UC
 006630 25 004541 00 004460        011800         TCT SPST1                  COLNO                 RESTORE MAIN          P
                                   011810         RMK                                              COLUMN NUMBER         C
                                   011820         RMK                                              NOW REPOSIT TAPE      N
                                   011830         RMK                                              TO WHERE IT WAS       C
                                   011840         RMK                                              PRIOR TO PROCESSNG    C
                                   011850         RMK                                              EVEN COLUMN           C
 006640 45 004467 00 004537        011860 SP35    TCS TPOS,R                 TPOS1,R               COMPUTE # GAPS        P
 006650 61 006670 00 006660        011870         CTC * 2                    * 1
 006660 01 000000 00 000000        011880         PES                                              ILLOGICAL
 006670 25 004464 00 006710        011890         TCT TPOS                   SP36                  # GAPS TO REWIND
 006700 22 004055 00 006715        011900         OCT TT2                    * 1,5                 RWD GAPS TO           P
 006710 16 777777 00 000000        011910 SP36    RWS 777777                 000000                  REPOSITION          C
                                   011920         RMK                                              NOW RETURN AND        N
                                   011930         RMK                                              PROCESS NEXT RCD      C
                                   011940         RMK                                              OF MAIN FLOW          C
 006720 71 004710 00 000000        011950         TC  SP1                                                                J
                                   011960         RMK                                              CODE B RECORD         T
 006730 72 011737 00 600000        011970 SP40    SET #000002#,1             $T                    IS THIS FIRST CARD
                                                                                                                          D
 006740 43 003055 00 003057        011980         SC  OPS0,R-2               OPS0,R
 006750 61 006760 00 007040        011990         CTC * 1                    SP41                                        Z
 006760 72 004446 00 600000        012000         SET BXCN,Z                 $T                    START OF NEW PAGE     D
 006770 43 011740 00 011741        012010         SC  #2323#                 #2323#
 007000 61 007020 00 007020        012020         CTC * 2                    * 2
 007010 71 007040 00 000000        012030         TC  SP41                                                               Z
                                   012040         RMK                                              FORCE NEW PAGE        N
 007020 25 004474 00 004504        012050         TCT LNC1                   LNC3                  SET ADDRESS FOR      P
                                   012060         RMK                                              PSEUDO CONNECTOR      C
 007030 71 007470 00 000000        012070         TC  ENDCOL                                       END OF COLUMN CODING  S
 007040 71 005420 00 000000        012080 SP41    TC  SPB                                          WRITE B RECORD        J
                                   012090         RMK                                              ON D OR J CODE, PUT   T
                                   012100         RMK                                              BOX # OF JUMP IN      C
                                   012110         RMK                                              TAG TABLE ENTRY OF    C
                                   012120         RMK                                              DESTINATION TAGS      C
                                   012130         RMK                                              FOR FROM CONNECTORS    C
 007050 25 004074 00 000133        012140 DJ      TCT ALH+                   $3                    SET FOR 1ST TAG       P
 007060 25 011733 00 007230        012150         TCT #000001#,1             DJ3                   SET LOOP TALLY        P
 007070 71 004170 00 000000        012160 DJ1     TC  LOCTAG                                       LOOK UP DEST TAG      S
 007100 61 007200 00 007200        012170         CTC DJ2                    DJ2                   TAG FOUND IN TABLE     O
 007110 72 011742 00 600000        012180         SET #1#                    $T                    ADD 1 TO COUNTER      P
```

```
NAME  AUTOFLOW                                   INDEX     4000-501          SEG 03  DATE 111665     PAGE 028
  HSM   OP    A     M    B      SEQ #  TAG       OP  A ADDRESS              M  B ADDRESS          COMMENTS
007120 41 000000 55 000000   012190             BA  000000                  55 000000          OF CROSS REFRENCES   C
007130 72 000000 50 600000   012200             SET 000000                  50 $T .            FIRST REFERENCE       D
007140 43 011742 00 011742   012210             SC  #01#                       #01#              TO THIS POINT       C
007150 61 007160 00 007200   012220             CTC * 1                        DJ2                      YES*DJ2,NO*  C
007160 72 000033 50 600000   012230             SET 000033                  50 $T              BOX # OF CURRENT  P
007170 24 003020 00 003023   012240             STC BOXNO                      BOXNO,R         RECORD TO TABLE      C
007200 25 004100 00 000133   012250 DJ2.        TCT AHMT                       $3              SET FOR 2ND TAG      P
007210 66 007070 00 007230   012260             TA  DJ1                        DJ3             BOTH TAGS PROCESSED  D
007220 71 005360 00 000000   012270             TC  SP5                                        RETURN TO MAIN PATH  J
007230 01 000000 01 010101   012280 DJ3         DAC 000000                                     TALLY FIELD
                             012290             RMK                                            END OF PASS 2        .
007240 72 000000 00 000000   012300 ENDP2       IGN
007250 22 004550 00 007265   012310             OCT EVOD                       * 1,5           IF EVEN COLUMN
007260 72 000240 00 000000   012320             SET 000240                                       ILLOGICAL HALT
007270 72 011742 00 600000   012330             SET #01#                       $T              BYPASS WRITING FINAL
007300 43 004445 00 004446   012340             SC  BXCN,1                     BXCN,2           EF IF BOX # IS 00
007310 61 007320 00 007340   012350             CTC * 1                        * 3
007320 22 004056 00 007335   012360             OCT TT3                        * 1,5           WRITE EF FOR LAST    P
007330 12 011675 00 770000   012370             LW  #72#                       770000           PAGE OF CHART       C
007340 22 004056 00 007355   012380             OCT TT3                        * 1,5           REWIND OUTPUT        P
007350 12 007357 00 000073   012390             LW  *,7                        000073
007360 22 004056 00 007375   012400             OCT TT3                        * 1,5
007370 17 000000 00 000000   012410             RWD
007400 22 004055 00 007415   012420             OCT TT2                        * 1,5
007410 17 000000 00 000000   012430             RWD                                            REWIND INPUT         P
                             012440             RIS LOAD4                   04 STPAS3          READ SEG 04          P
007420 72 001167 00 600000                      SET 001167                     $T
007430 24 007450 00 007467                      STC * 2                        * 3,7
007440 71 000340 00 000000                      TC  000340
007450 00 004430 00 000000                      00  LOAD4
007460 00 000004 00 007450                      00  04                         STPAS3
                             012450             RMK                                            *STPAS3*             J
                             012460             RMK                                            PASS 2 SUBROUTINES   R
                             012470             RMK                                            CODING FOR END       T
                             012480             RMK                                              OF COLUMN          C
007470 25 000243 00 007710   012490 ENDCOL      TCT $STP                       ENDCOLX         SET EXIT             P
007500 25 004504 00 004563   012500             TCT LNC3                       PSRCD,7         PUT ADDRESS IN       P
007510 22 004463 00 004560   012510             OCT COLNO,R                    PSRCD,4         PSEUDO CONN RCD      C
007520 72 011743 00 600000   012520             SET *P*                        $T              IS CURRENT RECORD   D
007530 43 003024 00 003024   012530             SC  CODE                       CODE            A B CODE            C
007540 61 007560 00 007560   012540             CTC * 2                        * 2
007550 71 007720 00 000000   012550             TC  ENDCOL2                                                          Z
007560 72 011724 00 600000   012560             SET #03#                       $T              LAST COLUMN ON PAGE  D
007570 43 004463 00 004463   012570             SC  COLNO,R                    COLNO,R
007600 61 007720 00 007610   012580             CTC ENDCOL2                    * 1                                   Z
007610 72 011733 00 600000   012590             SET #000001#,1                 $T              ADJACENT COLUMN HAVE D
007620 43 004501 00 004503   012600             SC  LNC2,R-2                   LNC2,R            HAVE ANY BOXES     C
007630 61 010010 00 010010   012610             CTC ENDCOL3                    ENDCOL3                               Z
007640 44 004463 00 011733   012620             TCA COLNO,R                    #000001#,1      SHIFT TO NEXT COLUMN P
007650 71 010060 00 000000   012630 ENDCOLA     TC  PSCN                                       CREATE PSEUDO CONN.  S
007660 25 011733 00 004474   012640 ENDCOL1     TCT #000001#,1                 LNC1            SET COL ADDRESS      P
007670 25 004474 00 004500   012650             TCT LNC1                       LNC2            TO TOP OF PAGE       C
007700 25 004474 00 004504   012660             TCT LNC1                       LNC3
007710 71 777777 00 000000   012670 ENDCOLX     TC  777777                                     EXIT                 E
                             012680             RMK                                            START NEW PAGE        T
007720 25 011733 00 004460   012690 ENDCOL2     TCT #000001#,1                 COLNO           SET FOR COL 1        P
007730 51 004443 00 011701   012700             DA  PACN,R                     #7424#          ADD TO PAGE NO       P
007740 21 011746 00 004446   012710             1T  #742323#                   BXCN,2          SET BOX # TO ZERO    P
```

```
NAME  AUTOFLOW                                    INDEX     40000501          SEG 03  DATE 111665         PAGE 029
  HSM    OP    A      N    B       SEQ #  TAG     OP   A ADDRESS            N   B ADDRESS               COMMENTS
 007750 71 010060 00 000000       012720          TC   PSCN                                             CREATE PSEUDO CONN    S
 007760 22 004056 0C 007775       012730          OCT  TT3                      * 1,5                   WRITE EF TO           P
 007770 12 007777 00 000072       012740          LW   *,7                      00007Z                    OUTPUT TAPE         C
 010000 71 007660 0C 000000       012750          TC   ENDCOL1                                                                J
 010010 72 011725 00 600000       012760 ENDCOL3  SET  #,2#                     $T                      IS THIS COLUMN 3      D
 010020 43 004463 CC 004463       012770          SC   COLNO,R                  COLNO,R                 IF COL 3, NEW PAGE
 010030 61 007720 0C 010040       012780          CTC  ENDCOL2                  * 1                                           Z
 010040 44 004463 0C 011737       012790          TCA  COLNO,R                  #000002#,1              SHIFT 2 COLUMNS OVER  P
 010050 71 007650 00 000000       012800          TC   ENDCOLA                                                                J
                                  012810          RMK                                                   GENERATE PSEUDO       T
                                  012820          RMK                                                   CONNECTOR RECORD      C
 010060 25 000243 00 010270       012830 PSCN     TCT  $STP                     PSCNX                   SET EXIT              P
 010070 22 003024 0C 010101       012840          OCT  CODE                     * 1,1                   CURRENT OR PREV RCD   D
 010100 72 770001 00 600000       012850          SET  770001                   $T                      EQUAL TO J H E T      C
 010110 31 011747 0C 011752       012860          LNS  "JHET"                   "JHET"
 010120 61 010130 0C 010140       012870          CTC  * 1                      * 2
 010130 71 010270 0C 000000       012880          TC   PSCNX                                                                  Z
 010140 22 004060 00 010151       012890          OCT  LSTCD                    * 1,1                   DO NOT CREATE         N
 010150 72 770001 00 600000       012900          SET  770001                   $T                      PSEUDO-CONNECTOR      C
 010160 31 011747 0C 011752       012910          LNS  "JHET"                   "JHET"                  IF JUMP, HALT,        C
 010170 61 010200 0C 010210       012920          CTC  * 1                      * 2                     EXIT, OR TEXT         C
 010200 71 010270 00 000000       012930          TC   PSCNX
 010210 72 004565 0C 600000       012940          SET  PSRCD,9                  $T                      MOVE PAGE AND BOX     P
 010220 24 004441 0C 004442       012950          STC  PACN,1                   PACN,2                  TO PSEUDO CONN RCD    C
 010230 72 004567 00 600000       012960          SET  PSRCD,11                 $T                      BOX # IS ONE LESS     N
 010240 24 004445 0C 004446       012970          STC  BXCN,1                   BXCN,2                  THAN IT SHOULD BE     C
 010250 22 004056 0C 010265       012980          OCT  TT3                      * 1,5                   WRITE PSEUDO CONN     P
 010260 12 004554 0C 000000       012990          LW   PSRCD                    000000                  RCD TO OUTPUT         C
 010270 71 777777 00 000000       013000 PSCNX    TC   777777                                           EXIT                  F
                                  013010          RMK                                                   BOX # TO O/P
                                  013020          RMK                                                   MOVE PAGE # AND
                                  013030          RMK                                                   BOX # TO O/P AREA
 010300 25 000243 0C 010350       013040 BOP      TCT  $STP                     BOPX                    SET EXIT
 010310 22 004441 0C 003020       013050          OCT  PACN,1                   BOXNO                   MOVE PAGE # TO
 010320 22 004442 0C 003021       013060          OCT  PACN,2                   BOXNO,1                   OUTPUT AREA
 010330 22 004445 0C 003022       013070          OCT  BXCN,1                   BOXNO,2                 MOVE BOX # TO
 010340 22 004446 0C 003023       013080          OCT  BXCN,2                   BOXNO,3                   OUTPUT AREA
 010350 71 777777 00 000000       013090 BOPX     TC   777777                                           EXIT
                                  013100          RMK                                                   PASS 2 WRITE RTN      T
 010360 25 000243 0C 010460       013110 OPT2     TCT  $STP                     OPT2X                   SET EXIT              P
 010370 22 004550 0C 010405       013120          OCT  EVOD                     * 1,5                   IF THIS IS AN ODD     P
 010400 72 010420 0C 000000       013130          SET  * 2                                              COLUMN, STORE CODE    C
 010410 22 003024 CC 004C60       013140          OCT  CODE                     LSTCD                   -IT MAY BE NEEDED     C
                                  013150          RMK                                                   BY PSCN ROUTINE       C
 010420 25 003060 00 000133       013160          TCT  OPLTH                    $3                      RCD LGTH TO AM3       P
 010430 22 004056 00 010441       013170          OCT  TT3                      * 1,1
 010440 72 000000 0C 600000       013180          SET  0,0000                   $T                      WRITE RECORD          P
 010450 11 003010 03 003010       013190          SSX  OPAR                  C3 OPAR                      TO O/P FILE         C
 010460 71 777777 00 000000       013200 OPT2X    TC   777777                                           EXIT                  F
                                  013210          RMK                                                   SUBROUTINE TO         T
                                  013220          RMK                                                   DETERMINE STATUS      C
                                  013230          RMK                                                   OF TAG-INSERT         C
                                  013240          RMK                                                   NOTATION IN RCD       C
                                  013250          RMK                                                   PRZ-USE COL 2.        C
                                  013260          RMK                                                   PRN-NO COL 2.         C
                                  013270          RMK                                                   AM3 IS MSC OF TAG     C
 010470 25 000243 0C 011140       013280 SCOL     TCT  $STP                     SCOLX                   SET EXIT              P
 010500 71 004170 0C 000000       013290          TC   LOCTAG                                           FIND TAG IN TABLE     S
```

```
NAME  AUTOFLOW                              IMDIX    4000 501        SEG 03  DATE 111665     PAGE 030
 MSM   OP    A    M    B     SEQ #  TAG     OP  A ADDRESS,           N  B ADDRESS           COMMENTS
 010510 61 011170 00 011150  013300         CTC SCOL13                  SCOL11              RESULT OF TABLE       D
                             013310         RMK                                             SEARCH  FOUND=SCOL13  C
                             013320         RMK                                             ,UNDEF=SCOL11,NOTAG=  C
 010520 72 000017 50 600000  013330 SCOL1   SET 0.0017               50 $T                  IS TAG ALLREADY       D
 010530 43 011742 00 011742  013340         SC  #01#                    #01#                  ASSIGNED            C
 010540 61 011150 00 011150  013350         CTC SCOL11                  SCOL11                                    Z
 010550 25 000013 50 004471  013360 SCOL2   TCT 0.0013               50 TSQ                 GET SEQ # OF TAG      P
                             013370         RMK                                               FROM CHAIN TABLE    C
 010560 72 000003 10 600000  013380         SET 0.0003               10 $T                  IS TAG LOCATED        D
 010570 43 004471 00 004473  013390         SC  TSQ                     TSQ,R                 IN CURRENT CHAIN    C
 010600 61 010620 00 011150  013400         CTC SCOL3                   SCOL11
 010610 71 011150 00 000000  013410         TC  SCOL11                                                            Z
                             013420         RMK                                             GET CHAIN OF          N
                             013430         RMK                                             DESTINATION TAG       C
 010620 25 000113 00 000173  013440 SCOL3   TCT $1                      $7                  ADV. TO NEXT ENTRY    P
 010630 44 000173 00 011717  013450 SCOL4   TCA $7                      #000010#,1            IN CHAIN TABLE      C
 010640 72 000003 70 600000  013460         SET 0.0003               70 $T                  TAG PART OF THIS      D
 010650 43 004471 00 004473  013470         SC  TSQ                     TSQ,R                CHAIN TABLE ENTRY    C
 010660 61 010630 00 010670  013480         CTC SCOL4                   * 1
 010670 25 777773 70 004001  013490 SCOL5   TCT 777773               70 TEMP2               GET START OF THIS     P
 010700 44 004003 00 011733  013500         TCA TEMP2,R                 #000001#,1            CHAIN FROM TABLE    C
 010710 72 004003 00 600000  013510         SET TEMP2,R                 $T                  IS TAG THE START      D
 010720 43 004471 00 004473  013520         SC  TSQ                     TSQ,R                 OF A CHAIN          C
 010730 61 011210 00 011170  013530         CTC SCOL14                  SCOL13
 010740 25 004504 00 004001  013540 SCOL6   TCT LNC3                    TEMP2               IS COLUMN 2 FREE      D
 010750 44 004003 00 011757  013550         TCA TEMP2,R                 #000007#,1            AT THIS POINT       C
 010760 45 004003 00 004503  013560         TCS TEMP2,R                 LNC2,R
 010770 61 011000 00 011150  013570         CTC * 1                     SCOL11
 011000 25 004474 00 004001  013580 SCOL7   TCT LNC1                    TEMP2               ENOUGH ROOM IN COL 2  D
 011010 45 004003 00 004513  013590         TCS TEMP2,R                 DECAD,R              TO HOLD ENTIRE CHAIN
                                                                                                       C
 011020 44 004003 07 000007  013600         TCA TEMP2,R              07 000007
 011030 45 004003 00 011737  013610         TCS TEMP2,R                 #000002#,1
 011040 45 004003 00 004117  013620         TCS TEMP2,R                 EPCON,R              ENTIRE CHAIN         C
 011050 61 011150 00 011060  013630         CTC SCOL11                  * 1
 011060 25 777773 70 004515  013640 SCOL8   TCT 777773               70 STOR5               COMPUTE # GAPS        P
 011070 45 004517 00 004467  013650         TCS STOR5,R                 TPOS,R              TO REACH CHAIN        C
 011100 25 004517 00 004001  013660 SCOL9   TCT STOR5,R                 TEMP2               # GAPS TO REACH       D
 011110 45 004003 00 011763  013670         TCS TEMP2,R                 #000144#,1            CHAIN TOO GREAT     C
 011120 61 011170 00 011130  013680         CTC SCOL13                  * 1
                             013690         RMK                                             CHAIN OK FOR ADJ COL  N
 011130 72 000002 00 100000  013700 SCOL10  SET 000002                  $PRI                SET PRZ               P
 011140 71 777777 00 000000  013710 SCOLX   TC  777777                                      EXIT                  E
                             013720         RMK                                             CHAIN NOT SUITED      N
                             013730         RMK                                             FOR ADJACENT COLUMN   C
 011150 72 000001 00 100000  013740 SCOL11  SET 000001                  $PRI                SET PRN               P
 011160 71 011140 00 000000  013750         TC  SCOLX                                       TO EXIT               J
 011170 22 011725 03 777777  013760 SCOL13  OCT #02#                 03 777777              SET INDICATOR         P
                             013770         RMK                                               FOR CONNECTOR       C
 011200 71 011150 00 000000  013780         TC  SCOL11                                                            J
 011210 25 004001 00 011300  013790 SCOL14  TCT TEMP2                   SCOL15              IS TAG SECOND         D
 011220 44 011303 00 011733  013800         TCA SCOL15,R                #000001#,1          RECORD OF CHAIN       C
 011230 45 011303 00 004473  013810         TCS SCOL15,R                TSQ,R
 011240 61 011170 00 011170  013820         CTC SCOL13                  SCOL13              NO
 011250 22 000004 70 011265  013830         OCT 000004               70 *, 1,5              IS 1ST RCD N OR T     D
 011260 72 010740 00 000000  013840         SET SCOL6                                       NO=SCOL6,YES=         C
 011270 71 011170 00 000000  013850         TC  SCOL13                                      NO GOOD FOR COL 2     J
 011300 01 000000 01 010101  013860 SCOL15  DAC 000000                                      WORK AREA
```

```
NAME  AUTOFLOW                                        INDEX     40000501        SEG 03  DATE 111665     PAGE 031
  HSM   OP     A    N    B          SEQ #  TAG         OP   A ADDRESS            N    B ADDRESS              CONTENTS
011304                              013870             DAR 2 0                                              FOR PATCHES
011614                                     #0056       CON 2,,0
        0101010101010101010101010101010101010101010101010101
011644                                     #0057       CON 22,,                     000000000000000000000000
        2323232323232323232323232323232323232323232323232323
011672                                     #0058       CON 01                .    76
011673                                     #0059       CON 01                #    75
011674                                     #0060       CON 01                #    67
011675                                     #0061       CON 01                .    72
011676                                     #0062       CON 01                #    41
011677                                     #0063       CON 01                     T
        63
011700                                     #0064       CON 02                #    7424
011702                                     #0065       CON 03                     DJS
        435162
011705                                     #0066       CON 01                #    43
011706                                     #0067       CON 02                #    5144
011710                                     #0068       CON 03                #    514744
011715                                     #0069       CON 03,1              #    000010
011721                                     #0070       CON 03,1              #    000024
011724                                     #0071       CON 01                #    03
011725                                     #0072       CON 01                #    02
011726                                     #0073       CON 01                #    04
011727                                     #0074       CON 01                #    40
011731                                     #0075       CON 03,1              #    000001
011735                                     #0076       CON 03,1              #    000002
011740                                     #0077       CON 02                #    2323
011742                                     #0078       CON 01                #    01
011743                                     #0079       CON 01                     B
        41
011744                                     #0080       CON 03                     742323
011747                                     #0081       CON 04                     JHET
        51474463
011755                                     #0082       CON 03,1              #    000007
011761                                     #0083       CON 03,1              #    000144
011770 00 000000 00 000003                             CON 8,10              #    0000000000000003
```

```
NAME  AUTOFLOW                                          INDEX     4000^501         SEG 04  DATE 111665     PAGE 032
  HSM   OP     A   N    B      SEQ #  TAG        OP  A ADDRESS               N    B ADDRESS          COMMENTS
                               013680            SEG 04                            LOAD4             PASS THREE
004430                         013690            DAR 1                                               USED FOR ST MESS
004600                         013700 HDR1       DAR 120,200                                         MAIN HEADER
                               013910    .       SLC HDR1
004600                         013920 HDR11      CON 7                       #    57615646614054
004607                         013930            CON 1                       #    06
                               013940            SLC HDR1,53
004665                         013950 HDR12      CON 17                           TABLE OF CONTENTS
        634041634401564501425655634455636 2
                               013960            SLC HDR1,88
004730                         013970 HDR13      CON 4                            DATE
        43406344
                               013980            SLC HDR1,R-8
004757                         013990 HDR14      CON 4                            PAGE
        57404644
                               014000            SLC HDR1,R 1
004770                         014010            CON 2                       #    7475
005000                         014020 HDR2       DAR 120,200                                         SUB HEADER
                               014030            SLC HDR2,8
005010                         014040 HDR21      CON 6                       #    424740616306
                               014050            SLC HDR2,R 1
005170                         014060            CON 2                       #    7475
005172                         014070  .         DAR 1                                               USED FOR ST MESS
005200                         014080 HDR3       DAR 120,200                                         T OF C HEADER
                               014090            SLC HDR3,16
005220                         014100 HDR31      CON 16                           TAG    PAGE  BOX
        63404601010101574046440101415667
005240                         014110 HDR32      CON 14                              ASSY LINE #
        0101014062627001535055440136
                               014120            SLC HDR3,76
005314                         014130            CON 16                           TAG    PAGE  BOX
        63404601010101574046440101415667
005334                         014140            CON 14                              ASSY LINE #
        0101014062627001535055440136
                               014150            SLC HDR3,R 1
005370                         014160            CON 2                       #    7475
005372                         014170            CON 1                            &
        74
005400                         014180 PRAR       DAR 120,200                                         AREA FROM WHICH
                               014190            RMK                                                 ACTUAL PRINT OR
                               014200            RMK                                                 WRITE IS DONE
005570                         014210            CON 1                       #    75
005574 01 023200 01 010101     014220 ADDLAY     DAC LAYOUT                                          MSC OF PAGE
                               014230            RMK                                                   LAYOUT AREA
005600                         014240 PCON1      CON 12                           ------------
        141414141414141414141414
005614                         014250 PCON2      CON 9                            ---------
        141414141414141414
                               014260            RMK                                                  OF PROCESS BOX
005625                         014270 PCON3      CON 1                            1
        50
005626                         014280 ONE        CON 2                       #    7424
005630 01 000000 01 010101     014290 OCTZR      DAC 000000
005634 01 000006 01 010101     014300 OCTSX      DAC 000006
005640 01 000001 01 010101     014310 OCTON      DAC 000001
005644 01 000170 01 010101     014320 LNTH       DAC 000170                                          # CHAR PER LINE
                               014330            RMK                                                   ON PAGE LAYOUT
005651                         014340 TAL1       DAR 3,1                                             MISC TALLIES
```

```
NAME  AUTOFLOW                              INDEX     40000501       SEG 04  DATE 111665     PAGE 033
  MSM    OP     A    N    B       SEQ #  TAG      OP  A ADDRESS          N   B ADDRESS          COMMENTS
 005655                           014350 TAL2     DAR 3,1
 005661                           014360 TAL3     DAR 3,1
 005665                           014370 TAL4     DAR 3,1
 005670                           014380 VTCON    CON 1                      1
        50
                                  014390          RMK                                            VERTICAL LINE
 005671                           014400 HZCON    CON 12                     ------------
        141414141414141414141414
 005705                           014410          CON 12                     ------------
        141414141414141414141414
 005724 01 000016 01 010101       014420 CNLN     DAC 000016                                    # CHARACTERS FROM
                                  014430          RMK                                           LEFT SIDE OF COL
                                  014440          RMK                                           TO CENTERLINE
 005730 01 000170 01 010101       014450 NMBL     DAC 000170                                    # LINES PER PAGE
                                  014460 BPO      EQL 010000                                    OFF-LINE BREAKPNT
 005734                           014470 PNO      CON 6                  #   017423232401
 005742                           014480 PCMES1   CON 4                  #   76010101
 005746                           014490 PCMES2   CON 4                  #   01012075
 005752                           014500 CRC1     CON 7                  #   01011616160101
 005761                           014510          CON 7                  #   16010101010116
 005770                           014520          CON 7                  #   16010101010116
 005777                           014530          CON 7                  #   16010101010116
 006006                           014540          CON 7                  #   01011616160101
 006020 01 000005 01 010101       014550 CRC2     DAC 000005                                    # LINES TO CIRCLE
 006024 01 000007 01 010101       014560 CRC3     DAC 000007                                    LENGTH OF LINE
 006030                           014570 NCON1    CON 12                     ************
        151515151515151515151515
 006044                           014580 NCON2    CON 5                      *****
        1515151515
 006051                           014590 NCON3    CON 1                      *
        15
 006052                           014600 NCON4    CON 4                      NOTE
        55566344
 006056                           014610 HCON1    CON 7                      -------
        14141414141414
                                  014620          RMK                                            LINES OF HALT
 006065                           014630 HCON2    CON 4                      HALT
        47405363
 006071                           014640 SCON     CON 7                  #   01010114141414
 006100                           014650          CON 7                  #   14141414141414
 006107                           014660          CON 7                  #   14141414010101
 006116                           014670          CON 7                  #   01011501500101
 006125                           014680          CON 7                  #   01010101010101
 006134                           014690          CON 7                  #   01010101150101
 006143                           014700          CON 7                  #   01150101500101
 006152                           014710          CON 7                  #   01010101010101
 006161                           014720          CON 7                  #   01010101011501
 006170                           014730          CON 7                  #   15010101500101
 006177                           014740          CON 7                  #   01010101010101
 006206                           014750          CON 7                  #   01010101010115
 006215                           014760          CON 7                  #   01150101500101
 006224                           014770          CON 7                  #   01010101010101
 006233                           014780          CON 7                  #   01010101011501
 006242                           014790          CON 7                  #   01011501500101
 006251                           014800          CON 7                  #   01010101010101
 006260                           014810          CON 7                  #   01010101150101
 006267                           014820          CON 7                  #   01010114141414
 006276                           014830          CON 7                      14141414141414
```

```
NAME  AUTOFLOW                                   INDEX    4000*501         SEG 04  DATE 111665      PAGE 034
  HSM   OP     A        B        SEQ #  TAG      OP  A ADDRESS          N  B ADDRESS            COMMENTS
006305                           014840          CON 7
006314 01 000006 01 010101       014850 SCO'I    DAC 0:0006             .  14141414C10101
                                 014860          RMK                                            # LINES-1 IS
                                 014870          RMK                                            SUBROUTINE SYMBOL
                                 014880          RMK                                            OUTPUT ROUTINE       P
                                 014890          RMK                                            SUBROUTINE TO        T
                                 014900          RMK                                            PRINT OR WRITE       C
                                 014910          RMK                                            OUTPUT.  IF BPO      C
                                 014920          RMK                                            IS ON, WRITE         C
006320 25 000243 00 007060       014930 ENPG     TCT $STP                  ENPG4                TO OFF-LINE TAPE     C
006330 72 004765 00 600000       014940          SET H.R14,R 3             $T                   SET EXIT             P
006340 24 005736 00 005740       014950          STC P.O,2                 PNO,4                MOVE PAGE # TO HDR   P
006350 32 004763 00 004765       014960          ZS  HDR14,R 1             HDR14,R 3
006360 51 005741 00 005627       014970          DA  P.O,R                 ONE,R                ADD 1 TO PAGE #      P
006370 71 007070 00 010000       014980          TC  E.PG5                 BPO                  OFF-LINE              D
006400 22 004057 00 006415       014990          OCT TT4                   * 1,5
006410 12 004577 00 000000       015000          LW  HDR1,-1                                    WRITE MAIN AND       P
006420 22 004057 00 006435       015010          OCT TT4                   * 1,5                  SUB HEADERS        C
006430 12 005177 00 000000       015020 ENPG1A   LW  HDR3,-1                                    A ADD VARIABLE
006440 25 004114 00 005651       015030 ENPG1    TCT EPCON                 TAL1                 SET TALLY TO         P
006450 44 005653 00 005637       015040          TCA TAL1,R                OCTSX,R              # LINES ON PAGE      C
006460 25 005630 00 000133       015050          TCT OCTZR                 $3                   INITIALIZE AM3       P
                                 015060          RMK                                            ADJUST TALLY TO      P
                                 015070          RMK                                            AVOID PRINTING ANY   C
                                 015080          RMK                                            BLANK LINES AT       C
                                 015090          RMK                                            BOTTOM OF PAGE       C
006470 44 000133 00 005647       015100          TCA $3                    LNTH,R               SET AM3 TO BOTTOM
006500 66 006470 00 005651       015110          TA  *-1                   TAL1                 LINE OF PAGE
006510 25 004114 00 005651       015120          TCT EPCON                 TAL1                 RESET TALLY TO
006520 44 005653 00 005637       015130          TCA TAL1,R                OCTSX,R              # LINES ON PAGE
006530 72 023367 30 600000       015140 ENPG1B   SET LAYOUT,119         30 $T                   TEST LINE FOR
006540 43 007250 00 007437       015150          SC  SPACES                SPACES,R             ALL SPACES
006550 61 006620 00 006620       015160          CTC ENPG1C                ENPG1C
006560 66 006600 00 005651       015170          TA  * 2                   TAL1                 REDUCE TALLY
006570 71 006620 00 000000       015180          TC  ENPG1C                                     THIS IS BLANK PAGE
006600 45 000133 00 005647       015190          TCS $3                    LNTH,R               SET FOR LINE ABOVE
006610 71 006530 00 000000       015200          TC  ENPG1B                                     RECYCLE
006620 25 005630 00 000133       015210 ENPG1C   TCT OCTZR                 $3                   INITIALIZE AM3
006630 44 005653 00 005643       015220          TCA TAL1,R                OCTON,R              ADJUST TALLY
006640 72 005653 00 600000       015230          SET TAL1,R                $T                   IS AN EXTRA P/C
006650 43 007441 00 007443       015240          SC  PCTAL,R-2             PCTAL,R              REQUIRED TO GIVE
006660 61 006670 00 006700       015250          CTC * 1                   * 2                  A TWO-PAGE CHART
006670 22 007037 00 007035       015260          OCT ENPG3B,7              ENPG3B,5
006700 72 005567 00 600000       015270 ENPG2    SET PRAR,R                $T                   SET EXTRA P/C SW
006710 26 023200 33 023367       015280          STT LAYOUT             33 LAYOUT,119           MOVE LINE TO         P
006720 71 007140 00 010000       015290          TC  ENPG6                 BPO                  OUTPUT AREA          C
006730 22 004057 00 006745       015300          OCT TT4                   * 1,5                OFF-LINE              D
006740 12 005377 00 000000       015310          LW  PRAR,-1                                    WRITE LINE           P
006750 44 000133 00 005647       015320 ENPG3    TCA $3                    LNTH,R                TO PRINT TAPE       C
006760 66 006700 00 005651       015330          TA  ENPG2                 TAL1                 SET FOR NEXT LINE    P
006770 36 023200 00 060427       015340          SCT LAYOUT                LAYOUT,R             END OF PAGE           D
007000 71 007210 00 010000       015350 ENPG3A   TC  ENPG7                 BPO                  CLEAR PAGE AREA      P
007010 22 004057 00 007025       015360          OCT TT4                   * 1,5                PRINT OR WRITE       P
007020 12 005742 00 000000       015370          LW  PCMES1                                       PAGE CHANGE        C
007030 72 007060 00 404000       015380 ENPG3B   SET ENPG4                 404000               FALL THRU IF EXTRA
                                 015390          RMK                                            P/C IS REQUIRED
                                 015400          RMK                                            TO GIVE TWO-PAGE
007040 22 007036 00 007035       015410          OCT ENPG3B,6              ENPG3B,5             CHART
```

```
NAME  AUTOFLOW                                        INDEX    40000501         SEG 04  DATE 111665      PAGE 035
  HSM    OP    A     N     B       SEQ #  TAG      OP  A ADDRESS          N   B ADDRESS            COMMENTS
007050 71 007000 00 000000       015420            TC  ENPG3A
007060 71 777777 00 000000       015430 ENPG4      TC  777777                                     EXIT                    F
007070 02 004600 00 000000       015440 ENPG5      PR  HDR1                                       PRINT MAIN AND          P
007100 03 000002 00 000000       015450            PA  000002                                       SUB HEADERS           C
007110 02 005200 00 000000       015460 ENPG5A     PR  HDR3                                       A ADD VARIABLE
007120 03 000002 00 000000       015470            PA  000002
007130 71 006440 00 000000       015480            TC  ENPG1                                                              J
007140 75 000000 00 000000       015490 ENPG6      CSG                      000000                OPEN GATE               P
007150 02 005400 00 000000       015500            PR  PRAR                                       PRINT LINE AND          P.
007160 03 000001 00 000000       015510            PA  000001                                     PAPER ADVANCE           C
007170 75 000000 00 010000       015520            CSG                      010000                CLOSE GATE              P
007200 71 006750 00 000000       015530            TC  ENPG3                                                              J
007210 75 000000 00 000000       015540 ENPG7      CSG                      000000
007220 03 020000 00 000000       015550            PA  020000                                     PAGE CHANGE
007230 75 000000 00 010000       015560            CSG                      010000
007240 71 007030 00 000000       015570            TC  ENPG3B
007250                           015580 SPACES     DAR 120,0                                      AREA TO TEST SPACES
007440 01 000075 01 010101       015590 PCTAL      DAC 000075                                     CONSTANT FOR TEST
                                 015600            RMK                                            OF EXTRA P/C
007450                           015610 LOADS      DAR 8,10                                       LOAD PY-SEG 5
                                 015620            SLC LOADS
                                 015630            RMK                                            PRODUCE TABLE           B
                                 015640            RMK                                            OF CONTENTS             C
007450 36 007250 00 007437       015650 STPAS3     SCT SPACES               SPACES,R              SET SPACES CONSTANT
007460 72 004620 00 600000       015660            SET HDR11,R'10           ST                    PUT INDEX-SERIAL #      P
007470 24 004156 00 004165       015670            STC INDEX                INDEX,R               IN MAIN HEADER          C
007500 72 004661 00 600000       015680            SET HDR11,R 43           ST                    PUT TITLE IN            P
007510 24 004120 00 004155       015690            STC TITLE                TITLE,R               MAIN HEADER             C
007520 22 000271 00 004735       015700            OCT 000271               HDR13,R 2             MOVE DATE FROM          P
007530 22 000272 00 004736       015710            OCT 000272               HDR13,R 3             STANDARD LOCATION       C
007540 22 011326 00 004737       015720            OCT #22#                 HDR13,R 4             TO HEADER               C
007550 22 000273 00 004740       015730            OCT 000273               HDR13,R 5
007560 22 000274 00 004741       015740            OCT 000274               HDR13,R 6
007570 22 011326 00 004742       015750            OCT #22#                 HDR13,R 7
007600 22 000275 00 004743       015760            OCT 000275               HDR13,R 8
007610 22 000276 00 004744       015770            OCT 000276               HDR13,R 9
007620 71 007720 00 010000       015780            TC  ST31                 BPO                   OFF-LINE
007630 22 011327 00 004577       015790            OCT #76#                 HDR1,-1               PUT IN START
007640 22 011327 00 004777       015800            OCT #76#                 HDR2,-1                 MESSAGES
007650 22 011327 00 005177       015810            OCT #76#                 HDR3,-1
007660 22 011327 00 005377       015820            OCT #76#                 PRAR,-1
007670 22 004057 00 007705       015830            OCT TT4                  * 1,5                 WRITE P/C MESSAGE
007700 12 005742 00 000000       015840            LW  PCMES1
007710 71 007730 00 000000       015850            TC  ST32                                       J
007720 03 020000 00 000000       015860 ST31       PA  020000                                     PAPER ADVANCE  P
007730 22 004056 00 007745       015870 ST32       OCT TT3                  * 1,5                 RWD I/P TAPE            P
007740 17 000000 00 000000       015880            RWD
007750 36 023000 00 040427       015890            SCT LAYFR                LAYOUT,R              CLEAR PAGE AREA  P
                                 015900            RMK                                            BEGIN CODING TO         N
                                 015910            RMK                                            PRINT TABLE OF          C
                                 015920            RMK                                            CONTENTS                C
007760 25 004020 00 005665       015930 TGCN1      TCT NMTAG                TALA                  SET TALLY TC # TAG      P
007770 25 004030 00 000153       015940            TCT T 1%                 $5                    SET AMS TO 'SC          P
                                 015950            RMK                                            OF TAG TABLE            C
                                 015960            RMK                                            INITIALIZE FOR          N
                                 015970            RMK                                            NEW PAGE                C
010000 25 005574 00 000173       015980 TGCN2      TCT ALDLAY               $7                    SET AMT TO TOP          P
                                 015990            RMK                                            OF COLUMN 1             C
```

```
NAME  AUTOFLO.                                        INDEX    40000501         SEG 04  DATE 111665    PAGE 036
 HSM   OP     A     M     B       SEQ #   TAG        OP   A ADDRESS          N    B ADDRESS         COMMENTS.
010010 22 010257 00 010255        016000             OCT  T C 8,7                 TBC 8,5           SET SWITCH TBCN,R        P
                                  016010             RMK                                              TO FALL THROUGH        C
010020 25 004114 00 005655        016020  TBCN3      TCT  EPCON                   TAL2              SET LINE TALLY           F
010030 66 010050 00 005665        016030  TBCN4      TA   TACN5                   TAL4              END OF TAG TABLE         D
010040 71 010550 00 000000        016040             TC   TBCN13                                                             Z
010050 72 000001 50 600000        016050  TBCN5      SET  000001             50   $T                IS THIS ENTRY            D
010060 43 011330 00 011330        016060             SC   #77#                    #77R                A CHART TITLE          C
010070 61 010110 00 010110        016070             CTC  * 2                     * 2
010100 71 010340 00 000000        016080             TC   TBCN10                                    Z
010110 72 000026 70 600000        016090  TBCN6      SET  000026             70   $T                MOVE TAG TO PAGE         P
010120 24 000001 55 000007        016100             STC  000001             55   000007
010130 72 000032 70 600000        016110             SET  000032             70   $T                MOVE PAGE, BOX, AND      P
010140 24 000014 55 000015        016120             STC  000014             55   000015              AND ASSEMBLY LINE      C
010150 72 000037 70 600000        016130             SET  000037             70   $T                  NUMBER TO PAGE       C
010160 24 000016 55 000017        016140             STC  000016             55   000017
010170 72 000052 70 600000        016150             SET  000052             70   $T
010200 24 000024 55 000027        016160             STC  000024             55   000027
010210 22 000010 57 000046        016170             OCT  000010             57   000046
010220 44 000173 00 005647        016180  TBCN7      TCA  $7                      LNTH,R            ADJUST AMY TO.           P
                                  016190             RMK                                            NEXT LINE OF PAGE        C
010230 44 000153 00 004017        016200             TCA  $5                      TENTL,R           ADJUST AMS TO            P
                                  016210             RMK                                            NEXT TABLE ENTRY         C
010240 66 010030 00 005655        016220             TA   TBCN4                   TAL2              END OF COLUMN            D
                                  016230             RMK                                            SWITCH BELOW IS          N
                                  016240             RMK                                            SET TO TFR IF            C
                                  016250             RMK                                            THIS IS COLUMN 2         C
010250 72 010320 00 004000        016260  TBCN8      SET  TBCN9                   004000            *TBCN9,TFR.              W
010260 25 005574 00 000173        016270             TCT  ADDLAY                  $7                SET AMY TO TOP           P
010270 44 000173 00 011333        016280             TCA  $7                      #000074#,1        OF COLUMN TWO            C
010300 22 010256 00 010255        016290             OCT  TBCN8,6                 TBCN8,5           SET SWITCH TO TFR        P
010310 71 010020 00 000000        016300             TC   TBCN3                                                              J
010320 71 006320 00 000000        016310  TBCN9      TC   ENPG                                      PRINT OR WRITE           S
                                  016320             RMK                                            ENTIRE PAGE              C
                                  016330             RMK                                            START NEW PAGE           N
010330 71 010000 00 000000        016340             TC   TBCN2                                                              J
                                  016350             RMK                                            CHART TITLE ENTRY  T
010340 72 005657 00 600000        016360  TBCN10     SET  TAL2,R                  $T                MORE THAN 6 LINES        D
010350 43 011334 00 011336        016370             SC   #000006#                #000006#           LEFT IN COLUMN          C
010360 61 010530 00 010370        016380             CTC  TBCN12                  * 1
010370 72 005657 00 600000        016390             SET  TAL2,R                  $T                IF THIS IS TOP
010400 43 004115 00 004117        016400             SC   EPCON,R=Z               EPCON,R           OF COLUMN, DO
010410 61 010430 00 010430        016410             CTC  * 2                     * 2               NOT LEAVE
010420 71 010450 00 000000        016420             TC   * 3                                       BLANK LINE
010430 44 000173 00 005647        016430             TCA  $7                      LNTH,R            ADVANCE AMY TO           P
                                  016440             RMK                                            LEAVE BLANK LINE         C
010440 66 010450 00 005657        016450             TA   * 1                     TAL2,R
010450 72 000014 70 600000        016460             SET  000014             70   $T                MOVE CHART TITLE         P
010460 24 010520 00 010524        016470             STC  TBCN11                  TBCN11,R            TO PAGE                C
010470 72 000045 70 600000        016480             SET  000045             70   $T
010500 24 000002 55 000027        016490             STC  000002             55   000027
010510 71 010220 00 000000        016500             TC   TBCN7                                     J
010520                            016510  TBCN11     CON  5                       CHART
      4247406163
010530 44 005667 00 011343        016520  TBCN12     TCA  TAL4,R                  #000001#,1        ADD 1 TO TAG TALLY       P
                                  016530             RMK                                            FORCE END COLUMN         N
010540 71 010250 00 000000        016540             TC   TBCN8                                                              J
                                  016550             RMK                                            END TABLE OF CONTENTS
010550 71 006320 00 000000        016560  TBCN13     TC   ENPG                                      PRINT OR WRITE           S
```

```
NAME  AUTOFLOW                                          INDEX      40000501      SEG 04  DATE 111665     PAGE 037
  HSM    OP     A     N     B       SEQ #  TAG        OP  A ADDRESS           N   B ADDRESS          COMMENTS
                                    016570            RMK                                            FINAL PAGE             C
 010560 21 011347 00 005740         016580            IT  *74232324*              PNO,R=1            SET PAGE NUMBER        P
 010570 34 004665 00 004705         016590            SCC HDR12                   HDR12,R            BACK TO ONE            C
 010600 72 004704 00 600000         016600            SET HDR12,R=1               ST                 CHANGE TITLE OF        P
 010610 24 010710 00 010725         016610            STC TBCN14                  TBCN14,R           MAIN HEADER            C
 010620 25 010730 00 006430         016620            TCT TBCN15                  ENPG1A             MODIFY ENPG SUBRTN     P
 010630 25 010734 00 007110         016630            TCT TBCN16                  ENPG5A             TO PRINT DIFFERENT     C
                                    016640            RMK                                            SUBHEADER              C
                                    016650            RIS LOADS                C5 TP1                LOAD SEGMENT 5         P
 010640 72 001167 00 600000                           SET 001167                  ST
 010650 24 010670 00 010707                           STC * 2                     * 3,7
 010660 71 000340 00 000000                           TC  000340
 010670 00 007450 00 000000                           00  LOADS
 010700 00 000005 00 007450                           00  05                      TP1
                                    016660            RMK                                            *TP1*                  J
 010710                             016670 TBCN14     CON 14                      501 FLOW CHART
        3023240145535666014247406163
 010730 01 004777 01 010101         016680 TBCN15     DAC HDR2,=1
 010734 01 005000 01 010101         016690 TBCN16     DAC HDR2                                       FOR PATCHES
 010740                             016700            DAR 2,0
 011250                                    #0084      CON 22,0
        0101010101010101010101010101010101010101010101
 011300                                    #0085      CON 22,0                    0000000000000000000000
        232323232323232323232323232323232323232323232323
 011326                                    #0086      CON 01                   *  22
 011327                                    #0087      CON 01                   *  76
 011330                                    #0088      CON 01                   *  77
 011331                                    #0089      CON 03,1                 *  000074
 011334                                    #0090      CON 03                   -  000006
 011341                                    #0091      CON 03,1                 *  000001
 011344                                    #0092      CON 04                   *  74232324
 011350 00 000000 00 000004                           CON 8,10                 B  0000000000000004
```

```
NAME  AUTOFLOW                                     INDIX     4000C501       SEG 05  DATE 111665     PAGE 038
  MSH    OP     A    M      B      SEQ N  TAG    OP   A ADDRESS           N   B ADDRESS              COMMENTS
                                   016710        SEG 05                       LOADS                  PASS 3
                                   016720        RMK                                                 PASS THREE           P
                                   016730        RMK                                                 BEGIN PASS 3         T
 007450 22 004056 00 007465        016740 TP1    OCT TT3                      * 1,5
 007460 15 003010 00 000000        016750        BRF OPAR                                            READ RECORD          P
 007470 72 022026 00 600000        016760        SET #72#                     $T                     END FILE              D
 007500 43 003010 00 003010        016770        SC  OPAR                     OPAR
 007510 61 007540 00 000240        016780        CTC TP1A                     000240
 007520 71 006320 00 000000        016790        TC  ENPG                                            TO END               S
                                   016800        RMK                                                 PAGE SUBROUTINE      C
 007530 71 007450 00 000000        016810        TC  TP1                                             NEW PAGE             J
 007540 72 022027 00 600000        016820 TP1A   SET #73#                     $T                     END DATA           D
 007550 43 003010 00 003010        016830        SC  OPAR                     OPAR
 007560 61 007600 00 007600        016840        CTC * 2                      * 2
 007570 71 013120 00 000000        016850        TC  ENDP3                                           Z
 007600 72 022030 00 600000        016860 TP2    SET #67#                     $T                     PSEUDO CONNECTOR      D
 007610 43 003011 00 003011        016870        SC  OPAR,1                   OPAR,1                   RECORD            C
 007620 61 007640 00 007640        016880        CTC * 2                      * 2
 007630 71 012730 00 000000        016890        TC  PSD                                             Z
 007640 25 004064 00 000113        016900 TP3    TCT OPCOM                    $1                     SET AM1 TO MSC       P
 007650 45 000113 00 022033        016910        TCS $1                       E000001#,1             OF COMMENTS FIELD    C
 007660 72 022034 00 600000        016920        SET #41#                     $T
 007670 43 003024 00 003024        016930        SC  CODE                     CODE                   B CODE                D
 007700 61 007720 00 007720        016940        CTC * 2                      * 2
 007710 71 011150 00 000000        016950        TC  BCD                                             Z
 007720 25 004110 00 000133        016960 TP4    TCT AADDR                    $3
 007730 71 013540 00 000000        016970        TC  GETAD                                           COMPUTE LOCATION OF
                                                                                                               S
                                   016980        RMK                                                  SYMBOL ON PAGE       C
 007740 72 022035 00 600000        016990        SET #61#                     $T                     RECORD HAVE A TAG    D
 007750 43 003025 00 003025        017000        SC  TAG                      TAG
 007760 61 020060 00 020060        017010        CTC FR                       FR
 007770 72 022036 00 600000        017020 TP4A   SET #57#                     $T                                          Z
 010000 72 022036 00 600000        017030        SET #57#                     $T                     P CODE                D
 010010 43 003024 00 003024        017040        SC  CODE                     CODE
 010020 61 010040 00 010040        017050        CTC * 2                      * 2
 010030 71 010420 00 000000        017060        TC  PCD
 010040 72 022037 00 600000        017070        SET #51#                     $T                     J CODE                Z D
 010050 43 003024 00 003024        017080        SC  CODE                     CODE
 010060 61 010100 00 010100        017090        CTC * 2                      * 2
 010070 71 010630 00 000000        017100        TC  JCD
 010100 72 022040 00 600000        017110        SET #44#                     $T                     E CODE                Z D
 010110 43 003024 00 003024        017120        SC  CODE                     CODE
 010120 61 010140 00 010140        017130        CTC * 2                      * 2
 010130 71 011050 00 000000        017140        TC  ECD
 010140 72 022041 00 600000        017150        SET #55#                     $T                     N CODE                Z D
 010150 43 003024 00 003024        017160        SC  CODE                     CODE
 010160 61 010200 00 010200        017170        CTC * 2                      * 2
 010170 71 011310 00 000000        017180        TC  NCD
 010200 72 022042 00 600000        017190        SET #47#                     $T                     H CODE                Z D
 010210 43 003024 00 003024        017200        SC  CODE                     CODE
 010220 61 010240 00 010240        017210        CTC * 2                      * 2
 010230 71 011570 00 000000        017220        TC  HCD
 010240 72 022043 00 600000        017230        SET #62#                     $T                     S CODE                Z D
 010250 43 003024 00 003024        017240        SC  CODE                     CODE
 010260 61 010300 00 010300        017250        CTC * 2                      * 2
 010270 71 012040 00 000000        017260        TC  SCD
 010300 72 022044 00 600000        017270        SET #43#                     $T                     D CODE                Z D
```

```
NAME  AUTOFLOW                                      INDEX     40000301         SEG 05  DATE 111665        PAGE 039
   HSM   OP     A     N    B        SEQ #  TAG      OP  A ADDRESS          N    B ADDRESS                  COMMENTS
010310 43 003024 00 003024        017280           SC  CODE                     CODE
010320 61 010340 00 010340        017290           CTC * 2                      * 2
010330 71 014670 00 000000        017300           TC  DCD                                                            Z
010340 72 022045 00 600000        017310           SET #63#                     $T                T,CODE              D
010350 43 003024 00 003024        017320           SC  CODE                     CODE
010360 61 010400 00 010400        017330           CTC * 2                      * 2
010370 71 012500 00 000000        017340           TC  TCD                                                            Z
                                  017350           RMK                                            ILLOGICAL CONDITION N
010400 01 000000 00 000000        017360           PES                                            ILLEGAL CODE        H
010410 71 007450 00 000000        017370           TC  TPI                                        RESTART
                                  017380           RMK                                            PROCESS P CODE      T
010420 45 000173 00 005647        017390 PCD       TCS $7                       LNTH,R            BACK UP AMT TO      P
                                  017400           RMK                                            LINE ABOVE          C
010430 72 777774 70 600000        017410           SET 777774              70  $T                 PUT TAG ON LINE     P
010440 24 003025 00 003033        017420           STC TAG                      TAG,R             ABOVE SYMBOL        C
010450 22 003022 07 000011        017430           OCT BOXNO,2             07  000011             PUT BOX # ON LINE   P
010460 22 003023 07 000012        017440           OCT BOXNO,3             07  000012               ABOVE SYMBOL      C
010470 44 000173 00 005647        017450           TCA $7                       LNTH,R            PUT UPPER LINE      P
010500 72 000012 70 600000        017460           SET 000012              70  $T                 OF PROCESS BOX      C
010510 24 005600 00 005624        017470           STC PCON1                    PCON2,R           ON PAGE LAYOUT      C
010520 44 000173 00 005647        017480           TCA $7                       LNTH,R            INCREMENT TO        P
                                  017490           RMK                                            GET TO NEXT LINE    C
010530 22 005625 07 000012        017500 PCD1      OCT PCON3               07  000012             PUT IN SIDES OF     P
010540 22 005625 07 777766        017510           OCT PCON3               07  777766              PROCESS BOX        C
010550 71 013710 00 000000        017520           TC  HVLN                                       LINE OF TEXT TO PAGE S
010560 61 010530 00 010530        017530           CTC PCD1                     PCD1              END OF COMMENT FIELD D
010570 72 000012 70 600000        017540 PCD2      SET 000012              70  $T                 PUT LOWER LINE      P
010600 24 005600 00 005624        017550           STC PCON1                    PCON2,R           OF PROCESS BOX      C
                                  017560           RMK                                            ON PAGE LAYOUT      C
010610 71 014320 00 000000        017570           TC  VTLN                                       PUT IN VERTICAL     P
                                  017580           RMK                                              CONNECT LINE      C
010620 71 007450 00 000000        017590           TC  TPI                                        GET NEW RECORD      J
                                  017600           RMK                                            PROCESS J CODE      T
010630 22 003022 07 000005        017610 JCD       OCT BOXNO,2             07  000005             PUT BOX NUMBER      P
010640 22 003023 07 000006        017620           OCT BOXNO,3             07  000006             ON PAGE             C
010650 72 777773 70 600000        017630           SET 777773              70  $T                 PUT TAG OF JUMP     P
010660 35 003025 00 003033        017640           SCR TAG                      TAG,R             ON PAGE             C
010670 71 013370 00 000000        017650           TC  GENCR                                      DRAW CIRCLE ON PAGE S
010700 25 004074 00 000133        017660           TCT ALMT                     $3                MSC OF DESTINATION  P
                                  017670           RMK                                              TAG TO AM3        C
010710 71 004170 00 000000        017680           TC  LOCTAG                                     FIND TAG IN TABLE   S
010720 61 011000 00 011000        017690           CTC JCD1                     JCD1              TAG FOUND IN TABLE  D
010730 22 000014 57 777776        017700           OCT 000014              57  777776             PUT PAGE AND BOX #  P
010740 22 000015 57 777777        017710           OCT 000015              57  777777             OF DESTINATION TAG  C
010750 22 022046 07 000000        017720           OCT #16#                07  000000             INSIDE CIRCLE       C
010760 22 000016 57 000001        017730           OCT 000016              57  000001
010770 22 000017 57 000002        017740           OCT 000017              57  000002
011000 44 000173 00 005647        017750 JCD1      TCA $7                       LNTH,R            ADJUST AMT TO       P
011010 44 000173 00 005647        017760           TCA $7                       LNTH,R            BOTTOM OF CIRCLE    C
011020 72 000013 70 600000        017770           SET 000013              70  $T                 PUT DESTINATION     P
011030 24 003035 00 003043        017780           STC L.T                      LHT,R              TAG ON PAGE        C
011040 71 007450 00 000000        017790           TC  T.1                                        GET NEW RECORD      J
                                  017800           RMK                                            PROCESS E CODE      T
011050 72 777773 70 600000        017810 ECD       SET 777773              70  $T                 PUT TAG OF EXIT     P
011060 35 003025 00 003033        017820           SCR TAG                      TAG,R             ON PAGE             C
011070 22 003022 07 000005        017830           OCT BOXNO,2             07  000005             PUT BOX NUMBER      P
011100 22 003023 07 000006        017840           OCT BOXNO,3             07  000006             OF PAGE             C
011110 71 013370 00 000000        017850           TC  GENCR                                      DRAW CIRCLE ON PAGE S
```

```
NAME  AUTOFLOW                                        INDEX     40000501        SEG 05  DATE 111665    PAGE 040
 HSM    OP    A     N    B       SEQ #  TAG         OP - A ADDRESS              N  B ADDRESS          COMMENTS
011120 72 000002 70 600000     017860              SET 0.0002                  70 $T                 PUT EXIT             P
011130 24 022047 00 022052     017870              STC #44675063#                 #44675063#         INSIDE CIRCLE        C
011140 71 007450 00 000000     017880              TC  TP1                                           GET NEW RECORD       J
                               017890              RMK                                               PROCESS B CODE       T
011150 25 003060 00 000133     017900 BCD          TCT 0.LTH                      $3                 COMPUTE LENGTH OF    P
011160 44 000133 00 004073     017910              TCA $3                         AOPAR,R            COMMENTS FIELD       C
011170 45 000133 00 004067     017920              TCS $3                         OPCOM,R
011200 61 011220 00 011210     017930              CTC * 2                        * 1                BLOCK NAME PRESENT   D
011210 71 011270 00 000000     017940              TC  BCD2                                                               Y
011220 72 005020 30 600000     017950              SET HDR21,R 3               30 $T                 MOVE NEW BLOCK       P
011230 24 003064 03 003064     017960              STC COMM                    03 COMM               NAME TO HEADER 2     C
011240 34 005020 00 005020     017970              SCC HDR21,R 3                  HDR21,R 3          CLEAR ISS
011250 34 005021 30 005167     017980              SCC HDR21,R 4               30 HDR2,R             CLEAR REST OF LINE   P
011260 71 007450 00 000000     017990 BCD1         TC  TP1                                           GET NEW RECORD       J
011270 34 005020 00 005167     018000 BCD2         SCC HDR21,R 3                  HDR2,R             CLEAR WHOLE LINE     P
011300 71 011260 00 000000     018010              TC  BCD1                                                               J
                               018020              RMK                                               PROCESS N CODE       T
011310 44 000173 00 022057     018030 NCD          TCA $7                         #000002#,I         ADJUST AMT TO        P
                               018040              RMK                                               CENTERLINE OF NOTE   C
011320 45 000173 00 005647     018050              TCS $7                         LNTH,R             BACK UP AMT TO       P
                               018060              RMK                                                 LINE ABOVE         C
011330 72 000003 70 600000     018070              SET 000003                  70 $T                 PUT NOTE ABOVE       P
011340 24 006052 00 006055     018080              STC NCON4                      NCON4,R              TOP LINE           C
011350 22 003022 07 000007     018090              OCT BOXNO,2                 07 000007             PUT BOX NUMBER       P
011360 22 003023 07 000010     018100              OCT BOXNO,3                 07 000010             ON PAGE              C
011370 72 777775 70 600000     018110              SET 777775                  70 $T                 PUT TAG ON PAGE      P
011400 24 003025 00 003032     018120              STC TAG                        TAG,R-1
011410 44 000173 00 005647     018130              TCA $7                         LNTH,R             AMT TO NEXT LINE     P
011420 72 000010 70 600000     018140              SET 000010                  70 $T                 PUT IN TOP LINE      P
011430 24 006030 00 006050     018150              STC NCON1                      NCON2,R              OF NOTE SYMBOL     C
011440 44 000173 00 005647     018160              TCA $7                         LNTH,R             AMT TO NEXT LINE     P
011450 22 006051 07 000010     018170 NCD1         OCT NCON3                   07 000010             PUT IN SIDE OF       P
011460 22 006051 07 777770     018180              OCT NCON3                   07 777770             NOTE SYMBOL          C
011470 71 013710 00 000000     018190              TC  MVLN                                          LINE OF TEXT TO PAGE $
011500 61 011450 00 011450     018200              CTC NCD1                       NCD1               END OF COMMENT FIELD D
011510 72 000010 70 600000     018210              SET 000010                  70 $T                 PUT IN BOTTOM LINE   P
011520 24 006030 00 006050     018220              STC NCON1                      NCON2,R              OF NOTE SYMBOL     C
011530 45 000173 00 022057     018230              TCS $7                         #000002#,I         ADJUST AMT BACK TO   P
                               018240              RMK                                               COLUMN CENTERLINE    C
011540 71 014320 00 000000     018250              TC  VTLN                                          PUT IN VERTICAL      P
                               018260              RMK                                                CONNECT LINE        C
011550 45 000173 00 022057     018270              TCS $7                         #000002#,I
011560 71 007450 00 000000     018280              TC  TP1                                           GET NEW RECORD       J
                               018290              RMK                                               PROCESS H CODE       T
011570 45 000173 00 005647     018300 HCD          TCS $7                         LNTH,R             SET AMT TO LINE      P
                               018310              RMK                                               ABOVE SYMBOL         C
011600 22 003022 07 000005     018320              OCT BOXNO,2                 07 000005             PUT BOX # ON PAGE    P
011610 22 003023 07 000006     018330              OCT BOXNO,3                 07 000006
011620 72 777773 70 600000     018340              SET 777773                  70 $T                 PUT TAG ON PAGE      P
011630 35 003025 00 003033     018350              SCR TAG                        TAG,R
011640 44 000173 00 005647     018360              TCA $7                         LNTH,R             ADJUST AMT TO TOP    P
                               018370              RMK                                               LINE OF SYMBOL       C
011650 72 000003 70 600000     018380              SET 000003                  70 $T                 PUT IN TOP LINE      P
011660 24 006056 00 006064     018390              STC HCON1                      HCON1,R              OF HALT SYMBOL     C
011670 44 000173 00 005647     018400              TCA $7                         LNTH,R             ADJUST AMT TO        P
                               018410              RMK                                                 NEXT LINE          C
011700 22 022060 07 777774     018420              OCT #03#                    07 777774             PUT IN ENDS OF       P
011710 22 022061 07 000004     018430              OCT #04#                    07 000004               HALT SYMBOL        C
```

```
NAME  AUTOFLOW                                                INDEX    40000501          SEG 05  DATE 111665      PAGE 041
  HSM    OP     A     N     B        SEQ #   TAG       OP   A ADDRESS               N    B ADDRESS                COMMENTS
 011720 72 000001 70 600000       018440           SET 000001              70  $T                  PUT HALT IN          P
 011730 24 006065 00 006070       018450           STC HCON2                   HCON2,R               CENTER             C
 011740 44 000173 00 005647       018460           TCA $7                      LNTH,R              ADJUST AMT TO        P
                                  018470           RMK                                               NEXT LINE          C
 011750 72 000003 70 600000       018480           SET 000003              70  $T                  PUT IN BOTTOM LINE   P
 011760 24 006056 00 006064       018490           STC HCON1                   HCON1,R              OF HALT SYMBOL      C
 011770 72 022062 00 600000       018500           SET "J"                     $T                  IS 1ST CHAR OF        D
 012000 43 003045 00 003045       018510           SC  RHT                     RHT                 FIELD RHT A J         C
 012010 61 012030 00 012030       018520           CTC HCD1                    HCD1
                                  018530           RMK                                             NEXT RCD IS A JUMP   N
 012020 71 014320 00 000000       018540           TC  VTLN                                        DROP VERT DOWN LINE  P
 012030 71 007450 00 000000       018550 HCD1      TC  TP1                                         GET NEW RECORD       J
                                  018560           RMK                                             PROCESS $ CODE       T
 012040 45 000173 00 005647       018570 SCD       TCS $7                      LNTH,R              BACK UP TO           P
                                  018580           RMK                                                LINE ABOVE        C
 012050 22 003022 07 000006       018590           OCT BOXNO,2             07  000006              PUT BOX # ON PAGE     P
 012060 22 003023 07 000007       018600           OCT BOXNO,3             07  000007
 012070 72 777777 70 600000       018610           SET 777777              70  $T                  PUT TAG ON PAGE      P
 012100 24 003025 00 003033       018620           STC TAG                     TAG,R
 012110 44 000173 00 005647       018630           TCA $7                      LNTH,R              ADJUST AMT BACK      P
                                  018640           RMK                                             TO TOP LINE OF SYM   C
 012120 25 000173 00 005655       018650           TCT $7                      TAL2                PRESERVE AMT
 012130 25 022067 00 000133       018660           TCT #000000#,1              $3                  INITIALIZE AND
 012140 25 006314 00 005651       018670           TCT SCON1                   TAL1                # LINES TO ALLY
 012150 72 000012 70 600000       018680 SCD1      SET 000012              70  $T                  MOVE LINE OF FORM    P
 012160 24 006071 33 006115       018690           STC SCON                33  SCON,20               TO PAGE LAYOUT     C
 012170 44 000133 00 022073       018700           TCA $3                      #000025#,1          STEP ADDRESSES       P
 012200 44 000173 00 005647       018710           TCA $7                      LNTH,R                TO NEXT LINE       C
 012210 66 012150 00 005651       018720           TA  SCD1                    TAL1                ALL LINES MOVED
                                                                                                                        D
 012220 45 000173 00 005647       018730           TCS $7                      LNTH,R              BACK UP TO BOTTOM
                                  018740           RMK                                             LINE OF SYMBOL
 012230 71 014320 00 000000       018750           TC  VTLN                                        PUT IN VERTICAL      P
                                  018760           RMK                                               CONNECT LINE       C
 012240 25 005655 00 000173       018770           TCT TAL2                    $7                  SET AMT TO TOP       P
                                  018780           RMK                                               OF SYMBOL          C
 012250 44 000173 00 005647       018790           TCA $7                      LNTH,R              ADJUST TO FIRST      P
                                  018800           RMK                                               LINE OF TEXT       C
 012260 44 000173 00 022033       018810           TCA $7                      #C00001#,1          SHIFT CENTERLINE     P
                                  018820           RMK                                             ONE TO RIGHT         C
 012270 71 013710 00 000000       018830 SCD2      TC  MVLN                                        LINE OF TEXT TO PAGE  S
 012300 61 012270 00 012270       018840           CTC SCD2                    SCD2                END OF COMMENT FIELD  D
 012310 25 004074 00 000133       018850           TCT ALHT                    $3
 012320 71 004170 00 000000       018860           TC  LPCTAG                                      LOOK UP TITLE         S
                                  018870           RMK                                               IN TAG TABLE       C
 012330 61 012470 00 012470       018880           CTC SCD3                    SCD3                TITLE FOUND IN TABLE  D
 012340 25 005655 00 000173       018890           TCT TAL2                    $7                  SET AMT TO 2ND
 012350 44 000173 00 005647       018900           TCA $7                      LNTH,R              LINE OF SYMBOL
 012360 22 000014 57 777771       018910           OCT 000014              57  777771              MOVE PAGE AND        P
 012370 44 000173 00 005647       018920           TCA $7                      LNTH,R              BOX # OF             C
 012400 22 000015 57 777771       018930           OCT 000015              57  777771                SUBROUTINE         C
 012410 44 000173 00 005647       018940           TCA $7                      LNTH,R                TO SYMBOL          C
 012420 22 022046 07 777771       018950           OCT #16=                07  777771
 012430 44 000173 00 005647       018960           TCA $7                      LNTH,R
 012440 22 000016 57 777771       018970           OCT 000016              57  777771
 012450 44 000173 00 005647       018980           TCA $7                      LNTH,R
 012460 22 000017 57 777771       018990           OCT 000017              57  777771
 012470 71 007450 00 000000       019000 SCD3      TC  TP1                                         GET NEW RECORD        J
```

```
NAME  AUTOFLOA*                                   INDEX     40000501          SEG 05  DATE 111665     PAGE 042
 HSM    OP     A   N     B       SEQ #   TAG     OP  A ADDRESS              N   B ADDRESS            COMMENTS
                                 019010          RMK
012500 25 004064 00 012553       019020  TCD     TCT OFCOM                      TCD2,3               PROCESS T CODE           T
012510 45 012553 00 022033       019030          TCS TCD2,3                     #000001#,1           SET START OF             P
012520 25 000173 00 012643       019040          TCT $7                         TCD4,3               COMMENTS FIELD           C
012530 45 000173 00 022077       019050          TCS $7                         #000016#,1           SAVE AM7.
                                 019060          RMK                                                 ADJUST AM7 TO            P
012540 25 000173 00 012720       019070  TCD1    TCT $7                         TXLST                FIRST LINE OF COL        C
                                 019080          RMK                                                 SET MSC OF NEXT          P
012550 27 777777 00 012720       019090  TCD2    RD  777777                     TXLST                  LINE TO BE MOVED       C
                                 019100          RMK                                                 MOVE ONE LINE OF         P
012560 25 000223 00 012553       019110          TCT $STA                       TCD2,3               TEXT TO PAGE             C
                                 019120          RMK                                                 STORE STA AS LHE         P
012570 34 000000 77 000000       019130          SCC 0,0000                 77  000000               OF NEXT LINE             C
                                 019140          RMK                                                 CLEAR ISS FROM
012600 61 012630 00 012630       019150          CTC TCD3                       TCD3                 FRONT OF LINE
012610 44 000173 00 005647       019160          TCA $7                         LNTH,R               ANY MORE LINES           D
                                 019170          RMK                                                 ADJUST AM7 TO            P
012620 71 012540 00 000000       019180          TC  TCD1                                            NEXT LINE ON PAGE        C
012630 45 012643 00 005647       019190  TCD3    TCS TCD4,3                     LNTH,R               RECYCLE                  J
012640 72 777777 00 600000       019200  TCD4    SET 777777                     $T                   IS THIS AN ADJACENT      D
012650 43 022100 00 022100       019210          SC  "1"                        "1"                    COLUMN                 C
012660 61 007450 00 007450       019220          CTC TP1                        TP1
012670 44 000173 00 022077       019230          TCA $7                         #000016#,1           DROP A VERTICAL LINE     P
012700 71 014320 00 000000       019240          TC  VTLN                                              TO START OF CHAIN      C
012710 71 007450 00 000000       019250          TC  TP1                                                J
012720                           019260  TXLST   CON 8,0                        017777770 1000000
                                 019270          RMK                                                 PROCESS PSEUDO           T
                                 019280          RMK                                                    CONNECTOR             C
012730 25 004070 00 000133       019290  PSD     TCT AOPAR                      $3                   SET AM3 TO ADDRESS
012740 44 000133 00 022103       019300          TCA $3                         #000004#,1           FIELD WITHIN RCD
012750 71 013540 00 000000       019310          TC  GFTAD                                           COMPUTE LOCATION OF
                                                                                                                $
                                 019320          RMK                                                  SYMBOL ON PAGE          C
012760 71 013370 00 000000       019330          TC  GENCR                                           DRAW CIRCLE ON PAGE      $
012770 72 013113 00 600000       019340          SET PSD1,3                     $T                   MOVE PAGE AND
013000 26 003020 00 003023       019350          STT OPAR,8                     OPAR,11              BOX # TO W/A
013010 22 013110 07 777776       019360          OCT PSD1                   07  777776               MOVE PAGE # TO           P
013020 22 013111 07 777777       019370          OCT PSD1,1                 07  777777               INSIDE CIRCLE            C
013030 22 022046 07 000000       019380          OCT #16#                   07  000000
013040 22 022104 00 013111       019390          OCT #74#                       PSD1,1               ADD ONE TO               P
013050 51 013114 00 022106       019400          DA  PSD1,4                     #7424#               BOX NUMBER               C
013060 22 013112 07 000001       019410          OCT PSD1,2                 07  000001               MOVE BOX # TO            P
013070 22 013113 07 000002       019420          OCT PSD1,3                 07  000002               INSIDE CIRCLE            C
013100 71 007450 00 000000       019430          TC  TP1                                             GET NEW RECORD           J
013110                           019440  PSD1    CON 5,0                    #   0101010101
                                 019450          RMK                                                 END OF PASS 3            T
013120 22 004056 00 013135       019460  ENDP3   OCT TT3                        * 1,5                RND I/P TAPE             P
013130 17 000000 00 000000       019470          RWD
013140 71 013220 00 040000       019480          TC  ENDP30                     040000               CROSS REFERENCE          D
                                 019490          RMK                                                 LISTING REQUESTED        C
                                 019500          RIS START                  07  START                BRING IN SEGMENT 7   P
013150 72 001167 00 600000                       SET 001167                     $T
013160 24 013200 00 013217                       STC * 2                        * 3,7
013170 71 000340 00 000000                       TC  000340
013200 00 004170 00 000000                       00  START
013210 00 000007 00 004170                       00  07                         START
                                 019510          RMK                                                 *CI*                     J
                                 019520          RMK RETURN HERE AT END         OF CROSS REFERENCE   LISTING                  N
```

```
NAME  AUTOFLOW                                         INDEX      4000#S01        SEG 05  DATE 111665       PAGE 043
 HSM     OP     A     N     B           SEQ #  TAG     OP   A ADDRESS          N  B ADDRESS                 COMMENTS
 013220  71  013340  00  010000         019530 ENDP3C  TC   ENDP32                BPO                       OFF LINE LISTING       D
 013230  22  004057  00  013245         019540         OCT  T14                   • 1,5                     WRITE PAGE  P
 013240  12  013360  00  000000         019550         LW   PGCHMS                                          CHANGE MESSAGE   C
 013250  22  004057  00  013265         019560         OCT  T14                   • 1,5                     WRITE END FILE  P
 013260  12  013267  00  000072         019570         LW   •,7                   00007Z                      TO OUTPUT TAPE       C
                                        019580 ENDP31  RIS  LOADI              01 NEX                       CALL IN CONTROL        P
 013270  72  001167  00  600000                ENDP31  SET  001167                $T
 013300  24  013320  00  013337                        STC  • 2                   • 3,7
 013310  71  000340  00  000000                        TC   000340
 013320  00  003000  00  000000                        00   LOADI
 013330  00  000001  11  004260                        00   01                    NEX
                                        019590         RMK                                                  •NEX•   J
 013340  03  020000  00  000000         019600 ENDP32  PA   020000                                          PAGE CHANGE  P
 013350  71  013270  00  000000         019610         TC   ENDP31                                            J
 013360                                 019620 PGCHMS  CON  8                  ≠  760101010101012075
                                        019630         RMK                                                  PASS 3 SUBROUTINES     R
                                        019640         RMK                                                  CREATE CIRCLE ON       T
                                        019650         RMK                                                    PAGE LAYOUT          C
                                        019660         RMK                                                  AM7 POINTS TO          T
                                        019670         RMK                                                    MIDPOINT OF TOP      C
                                        019680         RMK                                                  OF CRCLE=CHANGED       C
                                        019690         RMK                                                  TO CENTER OF CIRCL     C
 013370  25  000243  00  013460         019700 GENCR   TCT  $STP                  GENCRX                    SET EXIT               P
 013400  25  006020  00  005651         019710         TCT  CRC2                  TALI                      # LINES TO TALLY       P
 013410  25  022067  00  000133         019720         TCT  #000000#•1            $3                        INITIALIZE AM3
 013420  66  013470  00  005651         019730 GENCR1  TA   GENCR2                TALI                      ALL LINES MOVED         D
 013430  45  000173  00  005647         019740         TCS  $7                    LNTH,R                    BACK OFF AM7 TO        P
 013440  45  000173  00  005647         019750         TCS  $7                    LNTH,R                    CENTER OF CIRCLE       C
 013450  45  000173  00  005647         019760         TCS  $7                    LNTH,R
 013460  71  777777  00  000000         019770 GENCRX  TC   777777                                          EXIT                   F
 013470  72  000003  70  600000         019780 GENCR2  SET  000003             70 $T                        MOVE FORM LINE TO      P
 013500  24  005752  33  005760         019790         STC  CRC1               33 CRC1,R                      PAGE LAYOUT          C
 013510  44  000133  00  006027         019800         TCA  $3                    CRC3,R                    SET ADDRESSES FOR      P
 013520  44  000173  00  005647         019810         TCA  $7                    LNTH,R                      NEXT LINE            C
 013530  71  013420  00  000000         019820         TC   GENCR1                                          RECYCLE                 J
                                        019830         RMK                                                  LOCATE, FROM AN        T
                                        019840         RMK                                                  ADDRESS, PROPER        C
                                        019850         RMK                                                  POSITION ON PAGE.      C
                                        019860         RMK                                                  AM3 HAS MSC OF         C
                                        019870         RMK                                                  COL #=ADDR FIELD.      C
                                        019880         RMK                                                  AM7 GETS ANSWER        C
 013540  25  000243  00  013640         019890 GETAD   TCT  $STP                  GETADX                    SET EXIT               P
 013550  25  000003  30  005651         019900         TCT  000003             30 TALI                      LINE # TO STORAGE      P
 013560  25  005574  00  000173         019910         TCT  ADDLAY                $7                        SET AM7 TO START OF     P
                                        019920         RMK                                                    PAGE LAYOUT AREA      C
 013570  44  000173  00  005647         019930         TCA  $7                    LNTH,R                    BUMP AM7 TO GIVE       P
                                        019940         RMK                                                  EXTRA TOP LINE         C
 013600  66  013650  00  005651         019950 GET1    TA   GET3                  TALI                      TALLY ON LINE #         D
                                        019960         RMK                                                           ZERO•GET3•  C
                                        019970         RMK                                                  AM5 NOW HAS MSC        N
                                        019980         RMK                                                  OF LINE ON LAYOUT      C
                                        019990         RMK                                                  NOW ADJUST AM7 TO      N
                                        020000         RMK                                                  START OF COLUMN        C
 013610  22  000000  30  005653         020010         OCT  000003             30 TALI,R                    COLUMN # TO TALLY      P
 013620  45  000173  00  022113         020020         TCS  $7                    #000017##1
 013630  66  013670  00  005651         020030 GET2    TA   GET4                  TALI                      TALLY ON COLUMN #       D
                                        020040         RMK                                                           ZERO•GET4•  C
 013640  71  777777  00  000000         020050 GETADX  TC   777777                                          EXIT                   F
```

```
NAME  AUTOFLOA                            INDEX    4000 501                REG 05  DATE 111665    PAGE 044
  RSM    OP     A     *.    B       SEQ #  TAG     OP  A ADDRESS        N  P ADDRESS           COMMENTS
 013650 44 000173 00 005647       020060 GET3    TCA $7                    LNTH,R              STEP AM7 BY 120     F
 013660 71 013600 00 000000       020070         TC  GET1                                                          J
 013670 44 000173 00 022117       020080 GET4    TCA $7                    #000036#,1          STEP AM7 BY 30      .P
 013700 71 013630 00 000000       020090         TC  GET2                                                          J
                                  020100         RMK                                           MOVE COMMENT LINE   T
                                  020110         RMK                                           FROM I/P RCL TO     C
                                  020120         RMK                                           PAGE LAYOUT. AM7    C
                                  020130         RMK                                           SET TO CENTER       C
                                  020140         RMK                                           OF AREA ON PAGE.    C
                                  020150         RMK                                           AM1 IS MSC-1 OF     C
                                  020160         RMK                                           LINE TO BE MOVED.   C
                                  020170         RMK                                           AM7 ADJUSTED TO     C
                                  020180         RMK                                           NEXT LINE AT EXIT   C
 013710 25 000243 00 014170       020190 MVLN    TCT $STP                  MVLNX               SET EXIT            P
 013720 25 003060 00 000133       020200         TCT OPLTH                 $3                  RCD LGTH TO AM3
 013730 72 740001 00 600000       020210         SET 740001                $T.                 LOCATE NEXT ISS     P
 013740 31 000001 13 003010       020220         LNS 000001             13 OPAP                  IN COMM FIELD      C.
 013750 25 000223 00 000133       020230         TCT $STA                  $3                  RHE LINE TO AM3     P
 013760 61 000240 00 014200       020240         CTC 0,0240                MVLN6               LAST LINE IN FIELD   D
                                  020250         RMK                                                  NO.MVLN6.YES. C
 013770 22 022035 00 014163       020260         OCT #.1#                  MVLN5,3             SET FOR PRN         P
 014000 45 000133 00 000113       020270 MVLN1   TCS $3                    $1                  GET LENGTH OF LINE  P
 014010 61 014030 00 000240       020280         CTC MVLN2                 000240              NULL LINE            D
 014020 71 014140 00 000000       020290         TC  MVLN4                                     Z
 014030 25 000133 00 005661       020300 MVLN2   TCT $3                    TAL3                STORE AM3
 014040 25 000133 00 014057       020310         TCT $3                    . 1/7               DIVIDE LENGTH BY Z
 014050 71 014220 00 000000       020320         TC  OCTDV
 014060 25 014057 00 014113       020330         TCT .-1/7                 MVLN3,3             SET MOVE ADDRSS
 014070 25 005661 00 000133       020340         TCT TAL3                  $3                  RESTORE AM3
 014100 44 000133 00 000113       020350         TCA $3                    $1                  RHE LINE TO AM3
 014110 72 777777 70 600000       020360 MVLN3   SET 777777             70 $T                  MOVE LINE TO        P
 014120 24 000001 13 000000       020370         STC 000001             13 000000                PAGE LAYOUT        C
 014130 25 000133 00 000113       020380         TCT $3                    $1                  SET AM1 TO MSC-1    P
 014140 44 000113 00 022033       020390 MVLN4   TCA $1                    #000001#,1          OF NEXT I/P LINE    C
 014150 44 000173 00 005647       020400         TCA $7                    LNTH,R              SET AM7 FOR NEXT    P
                                  020410         RMK                                           LINE ON PAGE        C
 014160 72 000077 00 100000       020420 MVLN5   SET 0,0077                $PRI                SET PRIS            P
 014170 71 777777 00 000000       020430 MVLNX   TC  777777                                    EXIT                E
 014200 22 022120 00 014163       020440 MVLN6   OCT #02#                  MVLN5,3             SET FOR PRZ         P
 014210 71 014000 00 000000       020450         TC  MVLN1                                                          J
                                  020460         RMK                                           OCTAL DIVIDE BY 2.
                                  020470         RMK                                           B ADD OF TC IS
                                  020480         RMK                                           INPUT AND ANSWER
                                  020490         RMK                                           IF ODD #, LOWER
                                  020500         RMK                                           VALUE IS USED
 014220 25 000243 00 000133       020510 OCTDV   TCT $STP                  $3                  PICK UP STP
 014230 25 777777 30 014311       020520         TCT 777777             30 OCT3                PICK UP I/P #
 014240 25 022067 03 777777       020530         TCT #000000#,1         03 777777              SET ANSWER TO ZERO
 014250 45 014313 00 022057       020540 OCT1    TCS OCT3,R                #000002#,1          SUBTRACT 02
 014260 61 014270 03 000000       020550         CTC . 1                03 000000              IF NEGATIVE, EXIT
 014270 44 777777 30 022033       020560         TCA 777777             30 #000001#,1          ADD 1 TO ANSWER
 014300 71 014250 00 000000       020570 OCT2    TC  OCT1
 014311                           020580 OCT3    DAR 3,1
                                  020590         RMK                                           DROP VERTICAL LINE
                                  020600         RMK                                           FROM CURRENT
                                  020610         RMK                                           SYMBOL TO CONNECT
                                  020620         RMK                                           TO NEXT SYMBOL,
                                  020630         RMK                                           AM7 SET ONE LINE
```

```
NAME  AUTOFLOW                                 INDEX     40000501        SEG 05  DATE 111665    PAGE 045
  HSM    OP    A     N    B       SEQ #  TAG   OP  A ADDRESS         N   B ADDRESS          COMMENTS
                                  020640       RMK                                             ABOVE START
014320 25 000243 00 014370        020650 VTLN  TCT $STP                  VTLNX              SET EXIT
014330 25 022057 00 005651        020660       TCT #000002#,1            TALI               SET TALLY
014340 44 000173 00 005647        020670 VTLN1 TCA $7                    LNTH,R             ADVANCE ONE LINE
014350 22 005670 07 000000        020680       OCT VTCON             07  000000             PUT IN VERY CHAR
014360 66 014340 00 005651        020690       TA  VTLN1                 TALI
014370 71 777777 00 000000        020700 VTLNX TC  777777                                   EXIT
                                  020710       RMK                                          FOLLOWING AREAS
                                  020720       RMK                                          USED IN PROCESSING
                                  020730       RMK                                          DECISIONS
014401                            020740 TAGTA DAR 3,1                                      TALLY IN DEC LEG
014404                            020750 UNDS  DAR 1                                        UNDEF TAB INDICAT
014405                            020760 RSS   DAR 1                                        RIGHT SIDE SIGNAL
014406                            020770 LSS   DAR 1                                        LEFT SIDE SIGNAL
014411                            020780 SV7   DAR 3,1                                      SAVE AREA-AM7
014415                            020790 VTAL  DAR 3,1                                      MISC TALLY
014420                            020800 TPXX  DAR 8,0                                      TEMP STORAGE
014430                            020810 DCON  CON 7                 #   01010101010101
014437                            020820       CON 7                 #   01150101010101
014446                            020830       CON 3                 #   010101
014451                            020840       CON 7                 #   01010101010115
014460                            020850       CON 7                 #   01010115010101
014467                            020860       CON 3                 #   010101
014472                            020870       CON 7                 #   01010101150101
014501                            020880       CON 7                 #   01010101011501
014510                            020890       CON 3                 #   010101
014513                            020900       CON 7                 #   01011501010101
014522                            020910       CON 7                 #   01010101010101
014531                            020920       CON 3                 #   150101
014534                            020930       CON 7                 #   15010101010101
014543                            020940       CON 7                 #   01010101010101
014552                            020950       CON 3                 #   010115
014555                            020960       CON 7                 #   01011501010101
014564                            020970       CON 7                 #   01010101010101
014573                            020980       CON 3                 #   150101
014576                            020990       CON 7                 #   01010101150101
014605                            021000       CON 7                 #   01010101011501
014614                            021010       CON 3                 #   010101
014617                            021020       CON 7                 #   01010101010115
014626                            021030       CON 7                 #   01010115010101
014635                            021040       CON 3                 #   010101
014640                            021050       CON 7                 #   01010101010101
014647                            021060       CON 7                 #   01150101010101
014656                            021070       CON 3                 #   010101
014664 01 000010 01 010101        021080 DCON1 DAC 0-0010                                   # LINES-1 IN FORM
                                  021090       RMK                                          DECISION PROCESSING   R
                                  021100       RMK                                          PROCESS D CODE        T
014670 25 000173 00 014411        021110 DCD   TCT $7                    SV7                PRESERVE AM7
014700 25 022067 00 000133        021120       TCT #000000#,1            $3                 INITIALIZE AM3
014710 25 014664 00 005651        021130       TCT DCON1                 TALI               # LINES TO TALLY      P
014720 72 000010 70 600000        021140 DCD1  SET 000010            70  $T                 MOVE LINE OF FORM     P
014730 24 014430 33 014450        021150       STC DCON              33  DCON,16             TO PAGE LAYOUT AREA  C
014740 44 000133 00 022123        021160       TCA $3                    #000021#,1         STEP ADDRESSES        P
014750 44 000173 00 005647        021170       TCA $7                    LNTH,R             TO NEXT LINE          C
014760 66 014720 00 005651        021180       TA  DCD1                  TALI               ALL FORM LINES MOVED  D
014770 72 777777 70 600000        021190       SET 777777            70  $T                 PUT LABEL ON          P
015000 35 003076 00 003102        021200       SCP COMM,10               COMM,14            EQUAL BRANCH          C
015010 45 000173 00 005647        021210       TCS $7                    LNTH,R             ADJUST AM7 TO LAST    P
```

```
NAME  AUTOFLOW                          INDEX    4000-501            SEG 05  DATE 111665       PAGE 046
  HSN    OP    A    N    B     SEQ #  TAG    OP.  A ADDRESS      N   B ADDRESS           COMMENTS
                               021220        RMK                                            LINE OF SYMBOL      C
015020 71 014320 00 000000     021230        TC   VTLN                                   PUT IN VERTICAL      P
015030 44 000173 00 005647     021240        TCA  $7                 LNTH,R              CONNECT LINE         C
015040 22 005670 07 000000     021250        OCT  VTCON          07  000000
015050 25 014411 00 000173     021260        TCT  SV7                $7                  SET AM7 BACK TO      P
                               021270        RMK                                         TOP OF SYMBOL        C
015060 72 777775 70 600000     021280        SET  777775         70  $T                  PUT TAG ON PAGE      P
015070 35 003025 00 003033     021290        SCP  TAG                TAG,R
015100 72 000004 70 600000     021300        SET  000004         70  $T                  PUT BOX # ON PAGE    P
015110 24 003022 00 003023     021310        STC  BOXNO,2            BOXNO,3
015120 44 000173 00 005647     021320        TCA  $7                 LNTH,R              ADJUST TO FIRST      P
015130 44 000173 00 005647     021330        TCA  $7                 LNTH,R              LINE OF TEXT         C
015140 44 000113 00 022113     021340        TCA  $1                 #000017#,1          SET AM1 TO START   P
                               021350        RMK                                           OF COMMENT FIELD   C
015150 71 013710 00 000000     021360 DCD2   TC   MVLN                                   LINE OF TEXT TO PAGE S
015160 61 015150 00 015150     021370        CTC  DCD2               DCD2                END OF COMMENT FIELD D
                               021380        RMK                                         BEGIN LOGIC TO       N
                               021390        RMK                                         EXAMINE BRANCHES     C
                               021400        RMK                                         AND INSERT LINES     C
                               021410        RMK                                         AND CONNECTORS       C
015170 22 022031 00 014406     021420 DCD3   OCT  #00#               LSS                 SET LEFT AND RIGHT   P
015200 22 022031 00 014405     021430        OCT  #00#               RSS                   SIDE SIGNALS TO OFF
                                                                                                               C
015210 25 022033 00 014401     021440        TCT  #000001#,1         TAGTA               SET TAG TALLY TO 1
015220 72 022061 00 600000     021450        SET  #04#               $T                  IF THE LEFT HAND     P
015230 43 003034 00 003034     021460        SC   LHTC               LHTC                DESTINATION TAG IS   C
015240 61 015410 00 015410     021470        CTC  DCD4               DCD4                INDICATED FOR        C
015250 72 014427 00 600000     021480        SET  TPXX,R             $T                  ADJACENT COLUMN,     C
015260 26 003034 00 003043     021490        STT  LHTC               LHT,R               REVERSE POSITIONS    C
015270 72 003043 00 600000     021500        SET  LHT,R              $T                  OF LEFT AND RIGHT    C
015300 26 003044 00 003053     021510        STT  RHTC               RHT,R               DEST TAGS.  THIS     C
015310 72 003053 00 600000     021520        SET  RHT,R              $T                  INSURES ADJ COLUMN   C
015320 26 014420 00 014427     021530        STT  TPXX               TPXX,R              TAG IS ALWAYS        C
015330 72 014424 00 600000     021540        SET  TPXX,4             $T                  PROCESS FIRST        C
015340 24 003064 00 003070     021550        STC  COMM               COMM,4
015350 72 003070 00 600000     021560        SET  COMM,4             $T
015360 24 003071 00 003075     021570        STC  COMM,5             COMM,9
015370 72 003075 00 600000     021580        SET  COMM,9             $T
015400 24 014420 00 014424     021590        STC  TPXX               TPXX,4
015410 25 014411 00 000173     021600 DCD4   TCT  SV7                $7                  SET AM7 TO ONE       P
015420 44 000173 00 005647     021610        TCA  $7                 LNTH,R              LINE ABOVE           C
015430 44 000173 00 005647     021620        TCA  $7                 LNTH,R              CENTER OF SYMBOL     C
015440 44 000173 00 005647     021630        TCA  $7                 LNTH,R
015450 25 004100 00 000133     021640        TCT  ARHT               $3                  SET AM3 TO MSC OF    P
                               021650        RMK                                         DESTINATION TAG      C
015460 71 004170 00 000000     021660        TC   LOCTAG                                 FIND TAG IN TABLE    S
015470 61 016440 00 016110     021670 DCD5   CTC  DCD14              DCD9                TEST RESULT OF TABLE D
                               021680        RMK                                         SEARCH   FOUND=DCD14 C
                               021690        RMK                                         ,UNDEF=DCD9,NOTAG=  C
015500 22 022031 00 014404     021700        OCT  #00#               UNDS                SET UNDEFINED TAG    P
                               021710        RMK                                         INDICATOR TO OFF     C
015510 22 003044 00 015523     021720 DCD6   OCT  RHTC               * 1,3               TEST BRANCH INDIC.   D
015520 72 000077 00 100000     021730        SET  000077             $PRI                CONN=DCD13,SEC       C
015530 61 016300 00 016460     021740        CTC  DCD13              DCD15               =DCD15,UNKWN=        C
                               021750        RMK                                         PRP=ADJ COLUMN       N
                               021760        RMK                                         PRN=UNKNOWN          C
                               021770        RMK                                         PR2=CONNECTOR        C
                               021780        RMK                                         INDICATOR SAYS TO    N
```

```
NAME  AUTOFLOW                                   INDEX    40000501        SEG 05  DATE 111665   PAGE 047
  HSM    OP    A    N    B           SEQ #  TAG   OP  A ADDRESS         N   B ADDRESS           COMMENTS
                                     021790       RMK                                           DROP CONNECTOR          C
015540 22 014405 00 015555           021800 DCD7  OCT R&S                   * 1,5               YES=DCD11,NO= RIGHT     D
015550 72 016200 00 000000           021810       SET DCD11                 000000              SIDE AVAILABLE          C
                                     021820       RMK                                           RIGHT CONNECTOR.-       N
015560 22 022031 00 016035           021830       OCT #00#                  DCD8B,5             SET DEST TAG SWITCH,    
015570 44 000173 00 022127           021840       TCA $7                    #000013#,1          SET AMT TO RIGHT        P
                                     021850       RMK                                           SIDE OF DEC SYMBOL      C
015600 72 000002 70 600000           021860       SET 000002            70  $T                  PUT LABEL ON            P
015610 24 003071 00 003075           021870       STC COMM,5                COMM,9              PAGE                    C
015620 44 000173 00 005647           021880       TCA $7                    LNTH,R              ADJUST AMT TO           P
                                     021890       RMK                                           VERTICAL MIDDLE         C
015630 72 000000 70 600000           021900       SET 000000            70  $T                  PUT IN HORIZONTAL       P
015640 24 005671 00 005673           021910       STC HZCON                 HZCON,2             LINE TO RIGHT           C
015650 22 022130 00 014405           021920       OCT #40#                  RSS                 SET RIGHT SIDE          P
                                     021930       RMK                                           INDICATOR TO ON         C
                                     021940       RMK                                           CODING BELOW IS         N
                                     021950       RMK                                           FOR CONNECTORS          C
                                     021960       RMK                                           OF EITHER SIDE          C
015660 44 000173 00 005647           021970 DCD8  TCA $7                    LNTH,R              PUT IN VERTICAL         P
015670 22 005670 07 000000           021980       OCT VTCON             07  000000              LINE TO CONNECTOR       C
015700 44 000173 00 005647           021990       TCA $7                    LNTH,R              ADVANCE AMT TO          
                                     022000       RMK                                           TOP LINE OF CONN        
015710 71 013370 00 000000           022010       TC  GENCR                                     DRAW CIRCLE ON PAGE     S
015720 22 014404 00 015735           022020       OCT UNDS                  * 1,5               UNDEF INDICATOR ON      D
015730 72 016010 00 000000           022030       SET DCD8A                                       NO=DCD8A,YES=         C
015740 22 000014 57 777776           022040       OCT 000014            57  777776              PUT PAGE AND BOX        P
015750 22 000015 57 777777           022050       OCT 000015            57  777777              INSIDE CIRCLE *         C
015760 22 022046 07 000000           022060       OCT #164              07  000000              AMS SET TO ENTRY        C
015770 22 000016 57 000001           022070       OCT 000016            57  000001              IN TAG TABLE BY         C
016000 22 000017 57 000002           022080       OCT 000017            57  000002              LOCTAG SUBROUTINE       C
016010 44 000173 00 005647           022090 DCD8A TCA $7                    LNTH,R              DROP AMT TO BOTTOM      P
016020 44 000173 00 005647           022100       TCA $7                    LNTH,R                 OF CIRCLE            C
016030 72 016070 00 000000           022110 DCD8B SET * 4                                       LEFT=RIGHT SIDE SWITCH
016040 72 777775 70 600000           022120       SET 777775            70  $T                  MOVE DESTINATION TAG    P
016050 35 003045 00 003053           022130       SCR RHT                   RHT,R               TO SIDE OF CONNECTOR    C
016060 71 016110 00 000000           022140       TC  DCD9
016070 72 000011 70 600000           022150       SET OCD11             70  $T
016100 24 003045 00 003053           022160       STC RHT                   RHT,R
016110 66 016130 00 014401           022170 DCD9  TA  DCD10                 TAGTA               HAVE BOTH TAG FIELDS    D
                                     022180       RMK                                           BEEN PROCESSED          C
016120 71 007450 00 000000           022190       TC  TP1                                       GET NEXT RECORD  J
                                     022200       RMK                                           PROCESS 2ND TAG         N
016130 72 003053 00 600000           022210 DCD10 SET RHT,R                 $T                  MOVE TAG AND LABEL      P
016140 26 003034 00 003043           022220       STT LHTC                  LHT,R               OF 2ND BRANCH           C
016150 72 003075 00 600000           022230       SET COMM,9                $T                  TO OVERLAY FIELDS       C
016160 24 003064 00 003070           022240       STC COMM                  COMM,4              OF FIRST BRANCH         C
016170 71 015410 00 000000           022250       TC  DCD4                                                              J
                                     022260       RMK                                           LEFT CONNECTOR          N
016200 22 022130 00 016035           022270 DCD11 OCT #40#                  DCD8B,5             SET DEST TAG SWITCH
016210 45 000173 00 022127           022280       TCS $7                    #000013#,1          SET AMT TO LEFT         P
                                     022290       RMK                                           SIDE OF DEC SYMBOL      C
016220 72 000002 70 600000           022300       SET 000002            70  $T                  PUT LABEL               P
016230 24 003071 00 003075           022310       STC COMM,5                COMM,9              ON PAGE                 C
016240 44 000173 00 005647           022320       TCA $7                    LNTH,R              ADJUST AMT TO           P
                                     022330       RMK                                           VERTICAL MIDDLE         C
016250 72 000002 70 600000           022340       SET 000002            70  $T                  PUT IN HORIZONTAL       P
016260 24 005671 00 005673           022350       STC HZCON                 HZCON,2             LINE TO LEFT            C
                                     022360       RMK                                           GENERATE CONNECTOR      N
```

```
NAME AUTOFLOW                                             INDEX - 4000-501              SEG 05  DATE 111665      PAGE 045
  HSM   OP    A    N    B         SEQ #  TAG    OP  A ADDRESS                  N  B ADDRESS              COMMENTS
016270 71 015660 00 000000     022370         TC   DCD4
                               022380         RMK                                  DCD4                  INDICATOR SAYS USE       J N
                               022390         RMK                                                        SECONDARY COLUMN          C
016300 72 000015 70 600000     022400 DCD13   SET 000015                       70 $T                     PUT LABEL O'              P
016310 24 003071 00 003075     022410         STC COMN,5                          COMN,9                RIGHT SIDE OF SYM         C
016320 44 000173 00 005647     022420         TCA $7                              LNTH,R                AMT TO MIDDLE LINE        P
016330 44 000173 00 022117     022430         TCA $7                              #000036#,1            SET AMT TO CENTER         P
                               022440         RMK                                                        OF ADJACENT COLUMN        C
016340 72 000000 70 600000     022450         SET 000000                       70 $T                     MOVE HORIZONTAL           P
016350 24 005671 00 005716     022460         STC HZCON                           HZCON,21              LINE TO PAGE              C
016360 44 000173 00 005647     022470         TCA $7                              LNTH,R                DROP VERTICAL             P
016370 22 005670 07 000000     022480         OCT VTCON                        07 000000                LINE TO CONNECT           C
016400 44 000173 00 005647     022490         TCA $7                              LNTH,R                TO 1ST BOX OF CHAIN       C
016410 22 005670 07 000000     022500         OCT VTCON                        07 000000
016420 22 022130 00 014405     022510         OCT #40#                            RSS                   SET RIGHT SIDE            P
                               022520         RMK                                                        INDICATOR TO ON           C
016430 71 016110 00 000000     022530         TC  DCD9                                                                             J
                               022540         RMK                                 DCD5                  UNDEFINED TAG
016440 22 022130 00 014404     022550 DCD14   OCT #40#                            UNDS                  SET UNDEFINED             P
                               022560         RMK                                                        INDICATOR TO ON           C
                               022570         RMK                                                        CREATE CONNECTOR          N
016450 71 015540 00 000000     022580         TC  DCD7                                                                             J
                               022590         RMK                                                        DECIDE WHETHER IT         T
                               022600         RMK                                                        IS POSSIBLE TO DRAW       C
                               022610         RMK                                                        A LINE OR MUST A          C
                               022620         RMK                                                        CONNECTOR BE USED         C
016460 22 014406 00 016475     022630 DCD15   OCT L&S                             * 1,5                 LEFT SIDE SIGNAL ON       D
016470 72 015540 00 000000     022640         SET DCD7                                                       NO=DCD7,YES=      C
                               022650         RMK                                                        IF ON, USE CONNECTOR      N
016500 72 000015 50 600000     022660 DCD16   SET 000015                       50 $T                     IS DESTINATION ON         D
016510 43 005737 00 005740     022670         SC  PNO,3                           PNO,4                  ON THIS PAGE             C
016520 61 015540 00 015540     022680         CTC DCD7                            DCD7
016530 72 000020 50 600000     022690 DCD17   SET 000020                       50 $T                     COLUMN OF DEST. SAME      D
016540 43 003014 00 003014     022700         SC  ADDR                            ADDR                  AS COLUMN OF DEC BOX      C
016550 61 015540 00 015540     022710         CTC DCD7                            DCD7                  SAME AS COLUMN            C
016560 72 000023 50 600000     022720         SET 000023                       50 $T                     IS DEST ABOVE OR          D
016570 43 003015 00 003017     022730         SC  ADDR,1                          ADDR,3                BELOW DECISION SYMB.      C
016600 61 017300 00 016620     022740         CTC DCD23                           * 2
016610 71 015540 00 000000     022750         TC  DCD7                                                     BELOW=DCD23,ABOVE=      C
016620 72 777761 70 600000     022760 DCD18   SET 777761                       70 $T                     IS THERE A LINE           D
016630 43 022035 00 022035     022770         SC  #01#                            #01#                    IN THE WAY              C
016640 61 017630 00 017630     022780         CTC DCD29                           DCD29
                               022790         RMK BRANCH IS SUITABLE              FOR VERTICAL LINE     FOR VERTICAL LINE         N
016650 72 777767 70 600000     022800 DCD19   SET 777767                       70 $T                     PUT LABEL                 P
016660 35 003071 00 003075     022810         SCR COMN,5                          COMN,9                ABOVE LINE                C
016670 44 000173 00 005647     022820         TCA $7                              LNTH,R                ADJUST AMT TO             P
                               022830         RMK                                                        MIDDLE OF SYMBOL          C
016700 45 000173 00 022113     022840         TCS $7                              #000017#,1            ADJUST AMT TO COL         P
                               022850         RMK                                                        USED FOR VERT LINE        C
016710 72 000006 70 600000     022860         SET 000006                       70 $T                     PUT IN HORIZ LINE         P
016720 24 005671 00 005677     022870         STC HZCON                           HZCON,6               EXTENDING TO LEFT         C
                               022880         RMK                                                        TO JOIN VERT LINE         C
                               022890         RMK                                                        NOW COMPUTE LENGTH        N
                               022900         RMK                                                        OF VERTICAL LINE          C
                               022910         RMK                                                        NEEDED TO CONNECT         C
                               022920         RMK                                                        TO DESTINATION            C
016730 25 000023 50 014415     022930 DCD20   TCT 000023                       50 VTAL                   COMPUTE # LINES           P
016740 45 014417 00 003017     022940         TCS VTAL,2                          ADDR,3                BETWEEN MIDDLE OF         C
```

```
NAME  AUTOFLO*                                INDEX    40000501        SEG 05  DATE 111665    PAGE 049
 HSM    OP   A     N   B        SEQ #  TAG     OP  A ADDRESS        N   B ADDRESS         COMMENTS
016750 45 014417 00 022133     022950          TCS VTAL,2               *000005*,1        DEC SYMBOL AND TOP    C
                               022960          RMK                                        OF DESTINATION SYM    C
016760 45 014417 00 022057     022970          TCS VTAL,2               *000002*,1        ADJUST COUNT TO       P
                               022980          RMK                                        TWO LINES ABOVE       C
                               022990          RMK                                        TOP OF DEST SYMBOL.   C
016770 44 000173 00 005647     023000 DCD21    TCA $7                   LNTH,R            AM7 TO NEXT LINE      P
017000 66 017260 00 014415     023010          TA  DCD22                VTAL              IS LINE COMPLETE       D
017010 45 000173 00 005647     023020          TCS $7                   LNTH,R            BACK UP ONE LINE      P
017020 22 022134 07 000000     023030          OCT *V*               07 000000            AND PUT IN LETTER     C
017030 44 000173 00 005647     023040          TCA $7                   LNTH,R            V FOR DOWN ARROW      C
                               023050          RMK FOLLOWING LOGIC IS C OM MON TO BOTH UP AND D OWN LINES  T
017040 22 022130 00 014406     023060 DCD21A   OCT *40*                 LSS               SET LEFT SIDE         P
                               023070          RMK                                        INDICATOR TO ON       C
                               023080          RMK       CHECK FOR IN TE RFERENCE BETWEEN HOR IZONTAL
                               023090          RMK                  LI NE  AND FROM CONNECTORS
017050 72 000017 70 600000     023100          SET 000017            70 $T                IS THERE A TOP        D
017060 43 022046 00 022046     023110          SC  *16*                 *16*              FROM CONN IN WAY      C
017070 61 017130 00 017130     023120          CTR DCD21B               DCD21B
017100 72 000013 70 600000     023130          SET 000013            70 $T                PUT IN HORIZ LINE     P
017110 24 005671 00 005704     023140          STR HZCON                HZCON,11          BACK TO CENTER        C
017120 71 016110 00 000000     023150          TC  DCD9                                   J
017130 72 150001 00 600000     023160 DCD21B   SET 150001               $T                IS THERE A STAR       D
017140 31 000000 77 000016     023170          LNS 000000            77 000016            IN THE WAY            C
017150 61 017160 00 017170     023180          CTR * 1                  DCD21C            PRN-STAR IN WAY
017160 22 017217 00 017215     023190          OCT DCD21D,7             DCD21D,5          SET SW TO PUT IN *     P
017170 72 000016 70 600000     023200 DCD21C   SET 000016            70 $T                PUT IN HORIZ LINE     P
017200 24 005671 00 005707     023210          STR HZCON                HZCON,14          BACK TO CENTER        C
017210 72 017250 00 404000     023220 DCD21D   SET DCD21E               404000            *DCD21E,TFR*          W
                               023230          RMK SWITCH IS NOP IF STA R  WAS IN WAY OF LINE  N
017220 22 017216 00 017215     023240          OCT DCD21D,6             DCD21D,5          SET SWITCH TO TFR     P
017230 72 000013 70 600000     023250          SET 000013            70 $T                PUT IN * AND PAREN    P
017240 24 022135 00 022137     023260          STR *031504*             *031504*
017250 71 016110 00 000000     023270 DCD21E   TC  DCD9                                                         J
017260 22 005670 07 000000     023280 DCD22    OCT VTCON             07 000000            PUT IN ELEMENT OF     P
                               023290          RMK                                        VERTICAL LINE         C
017270 71 016770 00 000000     023300          TC  DCD21                                                        J
                               023310          RMK   TRY TO PUT IN LINE  G OING UP     T
017300 25 003017 00 014415     023320 DCD23    TCT ADDR,3               VTAL              COMPUTE TALLY QTY     P
017310 45 014417 05 000023     023330          TCS VTAL,2            05 000023            FOR # CHAR IN LINE    C
017320 44 014417 00 022133     023340          TCA VTAL,2               *000005*,1
017330 25 014415 00 017621     023350          TCT VTAL                 DCD28             PRESERVE TALLY QTY    P
017340 25 000173 00 000133     023360          TCT $7                   $3                AM7 TO AM3            P
017350 72 777761 30 600000     023370 DCD24    SET 777761            30 $T                IS ANY PART OF         D
017360 43 022035 00 022035     023380          SC  *01*                 *01*              VERTICAL COLUMN       C
017370 61 015540 00 015540     023390          CTR DCD7                 DCD7              IN USE                C
017400 66 017560 00 014415     023400          TA  DCD26                VTAL              TALLY ON # LINES
                               023410          RMK                      VERTICAL LINE CAN BE  USED              N
017410 25 017621 00 014415     023420          TCT DCD28                VTAL              RESTORE TALLY QTY     P
017420 72 777767 70 600000     023430          SET 777767            70 $T                PUT LABEL ON PAGE  P
017430 35 003071 00 003075     023440          SCR COMM,5               COMM,9
017440 44 000173 00 005647     023450          TCA $7                   LNTH,R            AM7 TO MID OF SYM     P
017450 45 000173 00 022113     023460          TCS $7                   *000017*,1        ADJUST AM7 TO COL     P
                               023470          RMK                                        USED FOR VERT LINE    C
017460 72 000006 70 600000     023480          SET 000006            70 $T                PUT IN HORIZ LINE     P
017470 24 005671 00 005677     023490          STR HZCON                HZCON,6           EXTENDING TO LEFT     C
017500 45 000173 00 005647     023500 DCD25    TCS $7                   LNTH,R            AM7 TO LINE ABOVE     F
017510 66 017600 00 014415     023510          TA  DCD27                VTAL              IS LINE COMPLETE       D
017520 44 000173 00 005647     023520          TCA $7                   LNTH,R               YES*DCD27,NO*   C
```

```
NAME  AUTOFLOW                                  IBM X    4000 501          SEG 05  DATE 111665       PAGE C50
 LSM    OP     A      B      SEQ #  TAG    OP   A ADDRESS                 B ADDRESS                COMMENTS
 017530 22 022130 07 000000  023530        OCT #.04                     7 000000                   PUT IN LETTER A        P
 017540 45 000173 00 005647  023540        TCS $7                         LNTH,R                   FOR AN UP ARROW  C
 017550 71 017040 00 000000  023550        TC  DCD21A                                                                     J
 017560 45 000133 00 005647  023560 DCD26  TCS $3                         LNTH,R                   AM3 TO LINE ABOVE
 017570 71 017350 00 000000  023570        TC  DCD24
 017600 22 005670 07 000000  023580 DCD27  OCT VYCON                    7 000000                   PUT IN ELEMENT         P
                             023590        RMK                                                     OF VERTICAL LINE       C
 017610 71 017500 00 000000  023600        TC  DCD25                                                                      J
 017621                      023610 DCD28  DAR 3,1                                                 TALLY STORAGE
                             023620        RMK CHECK IF LI    AY IS A DOWN LINE GOING TO            SAME PLACE
 017630 25 003017 00 020053  023630 DCD29  TCT ADD,3                      DCD32,R
 017640 44 020053 00 022143  023640        TCA DCD32,R                    #000006#,1               SET TO SCAN VERT  P
 017650 25 000173 00 000133  023650        TCT $7                         $3                       LINE FOR UP ARROW  C
 017660 44 020053 00 022033  023660 DCD30  TCA DCD32,R                    #000001#,1
 017670 44 000133 00 005647  023670        TCA $3                         LNTH,R                   STEP DOWN 1 LINE  P
 017700 72 022144 00 600000  023680        SET #50#                       $T                       COMPARE CHARACTER  D
 017710 43 777761 33 777761  023690        SC  777761                  33 777761                   TO OCTAL 50  C
 017720 61 017740 00 015540  023700        CTC DCD31                      DCD7                                            J
                             023710        RMK LOW MEANS LINE IS NO       GOOD
 017730 71 017660 00 000000  023720        TC  DCD30                                               RECYCLE           J
                             023730        RMK DOWN ARROW FOUND  N
 017740 72 000023 50 600000  023740 DCD31  SET 000023                  50 $T                       IS VERT LINE GOING  D
 017750 43 020051 00 020053  023750        SC  DCD32                      DCD32,R                  TO SAME DESTINATION  C
 017760 61 015540 00 015540  023760        CTC DCD7                       DCD7
 017770 72 777767 70 600000  023770        SET 777767                  70 $T                       PUT LABEL ON PAGE  P
 020000 35 003071 00 003075  023780        SCR COMM,5                     COMM,9
 020010 44 000173 00 005647  023790        TCA $7                         LNTH,R
 020020 72 777767 70 600000  023800        SET 777767                  70 $T                       CONNECT DEC SYMBOL  P
 020030 24 005671 00 005676  023810        STC HZCON                      HZCON,5                  TO VERTICAL LINE  C
 020040 71 016110 00 000000  023820        TC  DCD9                                                                       J
 020051                      023830 DCD32  DAR 3,1
                             023840        RMK                                                     FROM CONNECTORS        B
                             023850        RMK PUT FROM CONNECTORS  ON  PAGE
 020060 25 004104 00 000133  023860 FR     TCT ATAG                       $3
 020070 71 004170 00 000000  023870        TC  LOCTAG                                              LOCATE ENTRY FOR       S
                             023880        RMK                                                     TAG IN TABLE           C
 020100 61 007770 00 007770  023890        CTC TP4A                       TP4,                     FIND ENTRY IN TABLE    D
 020110 72 000033 50 600000  023900        SET 000033                  50 $T                       CROSS REFERENCE      D
 020120 43 022035 00 022035  023910        SC  #01#                       #01#                      FOR THIS SYMBOL       C
 020130 61 020150 00 020150  023920        CTC * 2                        * 2                      CROSS-REF PRESENT
 020140 71 007770 00 000000  023930        TC  TP4A                                                Y
 020150 45 000173 00 005647  023940        TCS $7                         LNTH,R                   BACK OFF TO TWO        P
 020160 45 000173 00 005647  023950        TCS $7                         LNTH,R                   LINES ABOVE SYMBOL   C
                             023960        RMK CHECK FOR REDUNDANT        REFERENCE - WILL OC      CUR AT TOP ADJ CHAIN   N
                             023970        RMK IF REF IS REDUNDANT        DO NOT SHOW ON PAGE                             N
 020170 72 000031 50 600000  023980        SET 000031                  50 $T                       PAGE # OF RCD EOL TO   D
 020200 43 003020 00 003021  023990        SC  BOXNO                      BOXNO,1                    PAGE # OF REF        C
 020210 61 020370 00 020370  024000        CTC FR1A                       FR1A
 020220 21 022150 00 021123  024010        IT  #74010101#                 FR6,3
 020230 72 021122 00 600000  024020        SET FR6,2                      $T
 020240 24 003022 00 003023  024030        STC BOXNO,2                    BOXNO,3
 020250 72 021126 00 600000  024040        SET FR7,2                      $T                       SUBTRACT BOX # OF      P
 020260 24 000032 55 000033  024050        STC 000032                  55 000033                     REF FROM BOX # OF    C
 020270 52 021123 00 021126  024060        DS  FR6,3                      FR7,2                      THIS RECORD          C
 020300 61 020310 00 020370  024070        CTC * 1                        FR1A
 020310 52 021123 00 022152  024080        DS  FR6,3                      #7425#                   COMPARE DIFF VS 2      D
 020320 61 020370 00 020560  024090        CTC FR1A                       FR2C                     2=FR1A,MORE=FR2C,1=    C
 020330 72 777420 70 600000  024100        SET 777420                  70 $T                       PRN = REDUNDANT
```

```
NAME  AUTOFLOW                                      INDEX    40000301       SEG 05  DATE 111665     PAGE 081
  HSM    OP     A     N     B       SEQ #  TAG      OP  A ADDRESS          N  B ADDRESS            COMMENTS
 020340 43 022153 00 022153        024110           SC  "."                   "."                  IS BOX ABOVE A NOTE     D
 020350 61 020370 00 020370        024120           CTC FR1A                  FR1A                 FR2 - REDUNDANT
 020360 71 020560 00 000000        024130           TC  FR2C                                                               Z
 020370 72 000000 70 600000        024140 FR1A      SET 000000             70 $T                   MIDDLE OR TOP OF        D
 020400 43 022035 00 022035        024150           SC  #01#                  #01#                   MAIN COLUMN           C
 020410 61 020530 00 020530        024160           CTC FR2                   FR2                       TOP=FR2,MID=       C
 020420 45 000173 00 022103        024170           TCS $7                    #000004#,1           ADJUST AM7 TO GIVE      P
                                   024180           RMK                                            CENTERED REFERENCE      C
 020430 71 020730 00 000000        024190           TC  FR3                                        PLACE REFERENCE         P
                                   024200           RMK                                            AT TOP OF SYMBOL        C
 020440 44 000173 00 022103        024210           TCA $7                    #000004#,1           RESTORE AM7             P
 020450 44 000173 00 005647        024220           TCA $7                    LNTH,R               PUT IN VERTICAL         P
 020460 22 005670 07 000000        024230           OCT VTCON              07 000000               LINE CHARACTER          C
 020470 71 020510 00 000000        024240           TC  * 2
 020500 44 000173 00 005647        024250 FR1       TCA $7                    LNTH,R               ADVANCE AM7 BACK TO     P
 020510 44 000173 00 005647        024260           TCA $7                    LNTH,R               TOP LINE OF SYMBOL      C
 020520 71 007770 00 000000        024270           TC  TP4A                                       RETURN                  J
                                   024280           RMK PUT FROM CONNECTOR  ON SIDE OF COLUMN                              N
 020530 72 777777 70 600000        024290 FR2       SET 777777             70 $T                   IS A                    D
 020540 43 022154 00 022154        024300           SC  #14#                  #14#                 HORIZONTAL LINE         C
 020550 61 020640 00 020640        024310           CTC FR2B                  FR2B                 IN THE WAY              C
                                   024320           RMK                                            HORIZ LINE IN WAY       N
 020560 72 000000 50 600000        024330 FR2C      SET 000000             50 $T                   IF MORE THAN       P
 020570 43 022120 00 022120        024340           SC  #02#                  #02#                 ONE REF TO THIS         C
 020600 61 020500 00 020610        024350           CTC FR1                   * 1                  POINT, PUT IN           C
 020610 72 777774 70 600000        024360           SET 777774             70 $T                   PARENS AND STAR         C
 020620 24 022135 00 022137        024370           STC #031504#              #031504#
 020630 71 020500 00 000000        024380           TC  FR1
 020640 72 777777 70 600000        024390 FR2B      SET 777777             70 $T                   PUT IN HORIZONTAL       P
 020650 24 020720 00 020727        024400           STC FR2A                  FR2A,R               LINE TO LEFT            C
 020660 45 000173 00 022113        024410           TCS $7                    #000017#,1           ADJUST AM7 TO LEFT      P
                                   024420           RMK                                            OF CENTERLINE           C
 020670 71 020730 00 000000        024430           TC  FR3                                        PUT REFERENCE TO        P
                                   024440           RMK                                            LEFT OF CENTERLINE      C
 020700 44 000173 00 022113        024450           TCA $7                    #000017#,1           RESET AM7               P
 020710 71 020500 00 000000        024460           TC  FR1                                                                J
 020720                            024470 FR2A      CON 8,0                   --------
         14141414141414141
                                   024480           RMK SUBROUTINE TO PLACE  REFERENCE ON PAGE.
                                   024490           RMK A-7 IS ONE TO LEFT OF  SPOT FOR LEFT PAREN
 020730 25 000243 00 021070        024500 FR3       TCT $STP.                 FR4                  SET EXIT
 020740 22 022060 07 000001        024510           OCT #(3#               07 000001
 020750 22 000030 57 000002        024520           OCT 000030             57 000002               PAGE #
 020760 22 000031 57 000003        024530           OCT 000031             57 000003
 020770 22 022046 07 000004        024540           OCT #16#               07 000004
 021000 22 000032 57 000005        024550           OCT 000032             57 000005               BOX #
 021010 22 000033 57 000006        024560           OCT 000033             57 000006
 021020 72 000000 50 600000        024570           SET 000000             50 $T                   MORE THAN ONE
 021030 43 022120 00 022120        024580           SC  #02#                  #02#                 REFERENCE
 021040 61 021100 00 021050        024590           CTC FR5                   * 1
 021050 72 022155 07 000007        024600           OCT #15#               07 000007               PUT IN STAR
 021060 22 022061 07 000010        024610           OCT #(4#               07 000010
 021070 71 777777 00 000000        024620 FR4       TC  777777                                     EXIT
 021100 22 022061 07 000007        024630 FR5       OCT # 4#               07 000007
 021110 71 021070 00 000000        024640           TC  FR4
 021120                            024650 FR6       DAR 4
 021124                            024660 FR7       CON 3                     740101
 021127                            024670           DAR 410                                        FOR PATCHES
```

```
NAME  AUTOFLOW                                        INDEX     40000501        SEG 05  DATE 111665      PAGE 052
  HSM    OP     A    N     B         SEQ #  TAG       OP  A ADDRESS           N   B ADDRESS              COMMENTS
 021750                                     #0093     CON 22,0
          01010101010101010101010101010101010101010101
 022000                                     #0094     CON 22,0                    0000000000000000000000
          23232323232323232323232323232323232323232323
 022026                                     #0095     CON 01                  #   72
 022027                                     #0096     CON 01                  #   73
 022030                                     #0097     CON 01                  #   67
 022031                                     #0098     CON 03,1                #   000001
 022034                                     #0099     CON 01                  #   41
 022035                                     #0100     CON 01                  #   01
 022036                                     #0101     CON 01                  #   57
 022037                                     #0102     CON 01                  #   51
 022040                                     #0103     CON 01                  #   44
 022041                                     #0104     CON 01                  #   55
 022042                                     #0105     CON 01                  #   47
 022043                                     #0106     CON 01                  #   62
 022044                                     #0107     CON 01                  #   43
 022045                                     #0108     CON 01                  #   63
 022046                                     #0109     CON 01                  #   16
 022047                                     #0110     CON 04                  #   44675063
 022055                                     #0111     CON 03,1                #   000002
 022060                                     #0112     CON 01                  #   03
 022061                                     #0113     CON 01                  #   04
 022062                                     #0114     CON 01                      J
          51
 022065                                     #0115     CON 03,1                #   000000
 022071                                     #0116     CON 03,1                #   000025
 022075                                     #0117     CON 03,1                #   000016
 022100                                     #0118     CON 01                      I
          50
 022101                                     #0119     CON 03,1                #   000004
 022104                                     #0120     CON 01                  #   74
 022105                                     #0121     CON 02                  #   7424
 022111                                     #0122     CON 03,1                #   000017
 022115                                     #0123     CON 03,1                #   000036
 022120                                     #0124     CON 01                  #   02
 022121                                     #0125     CON 03,1                #   000021
 022125                                     #0126     CON 03,1                #   000013
 022130                                     #0127     CON 01                  #   40
 022131                                     #0128     CON 03,1                #   000005
 022134                                     #0129     CON 01                      V
          65
 022135                                     #0130     CON 03                  #   031504
 022141                                     #0131     CON 03,1                #   000006
 022144                                     #0132     CON 01                  #   50
 022145                                     #0133     CON 04                  #   74010101
 022151                                     #0134     CON 02                  #   7425
 022153                                     #0135     CON 01                      •
          15
 022154                                     #0136     CON 01                  #   14
 022155                                     #0137     CON 01                  #   15
 022160 00 000000 00 000005                           CON 8,10                B   0000000000000005

NAME  AUTOFLOW                                        INDEX     40000501        SEG 06  DATE 111665      PAGE 053
  HSM    OP     A    N     B         SEQ #  TAG       OP  A ADDRESS           N   B ADDRESS              COMMENTS
                                     024680           SEG 06                      OSLOADI                SOURCE UPDATE PROGRAM
 003000                              024690           DAR 8                                              DUMMY BLOCK
 003010 00 000000 00 000006                           CON 8,10                B   0000000000000006
```

```
NAME  AUTOFLOW                                    INDEX    4000'50%      SEG 07  DATE 111665      PAGE 054
 MST   CP    A          B       SEQ #   TAG    OP  A ADDRESS,           N  B ADDRESS         COMMENTS
                                024700         SEG 07                       0$START
                                024710         HED D
                                024720         RMK                                             CROSS REFERENCE LIST    R
                                024730         RMK R,DUCE TAG TABLE TO     N ABBREVIATED VERSION CONTAINING TAG, PAGE  T
                                024740         RMK #, BOX#, FIRST REF,       AND ADDRESS OF NEXT REFERENCE,            C
004170 25 004120 00 003775      024750 C1      TCT NUMTAG                    TEMP1             SET TALLY TO NUMBER     P
004200 45 003777 00 010773      024760         TCS TEMP1,R                   #000001#,$        OF TABLE ENTRIES-1      C
004210 25 010777 00 004020      024770         TCT # 000000#,$               NUMTAG            SET # TAGS CNTR TO 0    P
004220 25 004030 00 000113      024780         TCT T-1"                      $1                SET AM1 TO MSC          P
                                024790         RMK                                             OF OLD TABLE            C
004230 25 007464 00 000133      024800         TCT RTMSC                     $3                SET AM3 TO MSC OF       P
                                024810         RMK                                             NEW TABLE               C
004240 72 000001 10 600000      024820 C2      SET 000001                 10 $T                THIS ENTRY A            D
004250 43 011000 00 011000      024830         SC  #76#                      #76#                CHART TITLE           C
004260 61 004270 00 004410      024840         CTC * 1                       C2A                                     Z
004270 72 000031 10 600000      024850         SET 000031                 10 $T                THIS ENTRY HAVE A       D
004300 43 011001 00 011001      024860         SC  #02#                      #02#                CROSS REFERENCE       C
004310 61 004410 00 004320      024870         CTC C2A                       * 1
004320 36 000000 33 000023      024880         SCT 000000                 33 000023            MOVE TAG, PAGE, AND     P
004330 72 000007 30 600000      024890         SET 000007                 30 $T                BOX # TO NEW TABLE      C
004340 26 000001 11 000007      024900         STT 000001                 11 000007
004350 72 000013 30 600000      024910         SET 000013                 30 $T
004360 26 000014 11 000017      024920         STT 000014                 11 000017
004370 44 004023 00 010773      024930         TCA NUMTAG,R                  #000001#,$        ADD 1 TO # TAGS CNTR    P
004400 44 000133 00 007453      024940         TCA $3                        RTL,R             ADJUST AM3 TO NEXT      P
                                024950         RMK                                             ENTRY IN NEW TABLE      C
004410 44 000113 00 007457      024960 C2A     TCA $1                        RTL,R             ADJUST AM1 TO NEXT      P
                                024970         RMK                                             ENTRY IN OLD TABLE      C
004420 66 004240 00 003775      024980         TA  C2                        TEMP1             END OF TABLE            D
                                024990         RMK CONSTRUCTION OF NEW       TAG TABLE COMPLETED                       N
                                025000         RMK SORT TAG TABLE INTO       SEQUENCE BY PAGE AN BOX #                 N
004430 22 011002 00 004640      025010 C3      OCT #72#                      C3C               SET INDICATOR TO 72     P
004440 25 007464 00 000173      025020         TCT RTMSC                     $7                SET AM7 TO MSC TABLE    P
004450 25 004020 00 003775      025030         TCT NUMTAG                    TEMP1             SET TALLY TO # TAGS     P
004460 45 003777 00 011007      025040         TCS TEMP1,R                   #000002#,$        MINUS ONE               C
004470 61 004500 00 007210      025050         CTC * 1                       ENDCC             IF LESS THAN 1, END     
004500 72 000013 70 600000      025060 C3A     SET 000013                 70 $T                COMPARE ENTRY VS        D
004510 43 000034 77 000037      025070         SC  000034                 77 000037              NEXT ENTRY    .       C
004520 61 004630 00 004540      025080         CTC C3B                       * 2                     HIGH=C3B,LOW=   C
004530 01 000000 00 000000      025090         PES                                             ILLOGICAL
004540 22 011010 00 004640      025100         OCT #71#                      C3C               SET INDICATOR TO 71     P
004550 72 004723 00 600000      025110         SET C3E,R                     $T                REVERSE ENTRIES         P
004560 26 000000 77 000023      025120         STT 000000                 77 000023              IN TABLE              C
004570 72 000023 70 600000      025130         SET 000023                 70 $T
004600 26 000024 77 000047      025140         STT 000024                 77 000047
004610 72 000047 70 600000      025150         SET 000047                 70 $T
004620 26 004700 00 004723      025160         STT C3E                       C3E,R
004630 66 004660 00 003775      025170 C3B     TA  C3D                       TEMP1             END OF ITERATION        D
                                025180         RMK                                             IS TABLE IN             D
004640 72 004430 00 600000      025190 C3C     SET C3                        $T                PROPER SEQUENCE         Y
                                025200         RMK                                             SORT IS FINISHED        N
004650 71 004750 00 000000      025210         TC  C4                                                                  J
004660 44 000173 00 007453      025220 C3D     TCA $7                        RTL,R             ADVANCE TO NEXT         P
                                025230         RMK                                             ENTRY IN TABLE          C
004670 71 004500 00 000000      025240         TC  C3A                                                                 J
004700                          025250 C3E     DAR 20,0                                        TEMPORARY STORAGE
004724                          025260         DAR 20                                          AVAIL IF EXPANSION
004750 22 004056 00 004765      025270 C4      OCT TT3                       * 1,5             REWIND O/P TAPE         P
```

```
NAME  AUTOFLOW                                        INDEX     40000501              SEG 07  DATE 111665        PAGE 055
  HSM    OP    A     N    B         SEQ #  TAG        OP  A ADDRESS              N    B ADDRESS               COMMENTS
004760 17 000000 00 000000          025280            RMD                                                      &FROM PASS 3

004770 22 000271 00 007740          025290            OCT 000271                      HDC11,6                                    C
005000 22 000272 00 007741          025300            OCT 000272                      HDC11,7                  MOVE DATE TO HEADER   P
005010 22 011011 00 007742          025310            OCT #22#                        HDC11,8
005020 22 000273 00 007743          025320            OCT 000273                      HDC11,9
005030 22 000274 00 007744          025330            OCT 000274                      HDC11,10
005040 22 011011 00 007745          025340            OCT #22#                        HDC11,11
005050 22 000275 00 007746          025350            OCT 000275                      HDC11,12
005060 22 000276 00 007747          025360            OCT 000276                      HDC11,13
005070 75 000000 00 010000          025370            CSG                             010000
005100 72 007620 00 600000          025380            SET HDC1,16                     $T
005110 24 004156 00 004165          025390            STC INDEX                       INDEX,R                  MOVE INDEX # TO HDR   P
005120 72 007661 00 600000          025400            SET HDC1,49                     $T
005130 24 004120 00 004155          025410            STC TITLE                       TITLE,R                  MOVE TITLE TO HDR     P
005140 22 004056 00 005155          025420 C5         OCT TT3                         * 1,5                    READ RECORD           P
005150 15 003010 00 000000          025430            BRF OPAR
005160 72 003010 00 600000          025440            SET OPAR                        $T
005170 43 011012 00 011012          025450            SC  #73#                        #73#                     TEST FIRST CHARACTER  D
005200 61 005140 00 005220          025460            CTC C5                          * 2
005210 71 006060 00 000000          025470            TC  C20                                                   SM,C5,EF,C20,ED,     C
                                    025480            RMK EF SYMBOL IS A PAGE         MARKER AND MUST BE       BYPASSED             N
005220 22 003024 00 005231          025490            OCT CODE                        * 1,1                    IS CODE D $ OR J      D
005230 72 770001 00 600000          025500            SET 770001                      $T
005240 31 011013 00 011015          025510            LNS "DJS"                       "DJS"
005250 61 005260 00 005140          025520            CTC * 1                         C5
005260 25 006050 00 000153          025530            TCT C16                         $5                       SET AMS TO MSC OF     P
                                    025540            RMK                                                      FIRST BRANCH TAG      C
005270 71 005520 00 000000          025550            TC  C10                                                  OPERATE ON FIRST      S
                                    025560            RMK                                                      BRANCH TAG            C
005300 25 006054 00 000153          025570            TCT C17                         $5                       SET AMS TO MSC OF     P
                                    025580            RMK                                                      SECOND BRANCH TAG     C
005310 71 005520 00 000000          025590            TC  C10                                                  OPERATE ON SECOND     S
                                    025600            RMK                                                      BRANCH TAG            C
005320 71 005140 00 000000          025610 C6         TC  C5                                                   GET NEW RECORD        J
                                    025620            RMK SUBROUTINE TO FIND          TAG IN TABLE.  TAG       IS STORED IN FNTG5.   T
                                    025630            RMK PRP IS SET IF TAG           IS NOT FOUND IN          TABLE.                C
                                    025640            RMK AM7 IS SET TO MSC           OF TABLE ENTRY FOR       TAG                   C
005330 25 000243 00 005400          025650 FNTG       TCT $STP                        FNTGX                    SET EXIT              P
005340 25 004020 00 005501          025660            TCT NUMTAG                      FNTG4                    # ENTRIES TO TALLY    P
005350 25 007464 00 000173          025670            TCT RTMSC                       $7                       SET MSC OF TABLE      P
005360 66 005410 00 005501          025680 FNTG1      TA  FNTG2                       FNTG4                    END OF TABLE          D
                                    025690            RMK                                                      ENTRY NOT FOUND       N
005370 72 000004 00 100000          025700            SET 000004                      $PRI                     SET PRP FOR NO FIND   P
005400 71 777777 00 000000          025710 FNTGX      TC  777777                                               EXIT                  E
005410 72 000007 70 600000          025720 FNTG2      SET 000007                   70 $T                       TAG VS TABLE ENTRY    D
005420 43 005505 00 005513          025730            SC  FNTG5,1                     FNTG5,R
005430 61 005460 00 005460          025740            CTC FNTG3                       FNTG3
005440 72 000002 00 100000          025750            SET 000002                      $PRI                     SET PRZ             G P
005450 71 005400 00 000000          025760            TC  FNTGX                                                TO EXIT               J
005460 44 000173 00 007453          025770 FNTG3      TCA $7                          RTL,R                    MOVE TO NEXT ENTRY    P
005470 71 005360 00 000000          025780            TC  FNTG1                                                RECYCLE               J
005501                              025790 FNTG4      DAR 3,1
005504                              025800 FNTG5      DAR 8                                                    STORAGE-TAG
                                    025810            RMK SUBROUTINE TO PLACE      4  CROSS REFERENCE IN       THE TABLE.            T
                                    025820            RMK AM5 HOLDS MSC OF            BRANCH TAG. AM3 IS       ALWAYS POINTING       C
                                    025830            RMK TO NEXT AVAILABLE           LOCATION IN TABLE.                             C
005520 25 000243 00 005560          025840 C10        TCT $STP                        C11                      SET EXIT              P
```

```
NAME  AUTOFLOW                                  IDEX    4000F501      SEG 07  DATE 111665    PAGE 056
 MSM   OP    A    C    B      SEQ #  TAG    OP  A ADDRESS           N  B ADDRESS        COMMENTS
005530 72 000001 50 600000   025850         SET 000001              50 $T               IS BRANCH TAG           D
005540 43 011016 00 011017   025860         SC  #-101#                 #0101#             ALL SPACES            C
005550 61 005570 00 005570   025870         CTC C12                    C12
005560 71 777777 00 000000   025880 C11     TC  777777                                  EXIT                    E
005570 72 005513 00 600000   025890 C12     SET F TG5,R                $T               MOVE TAG TO             P
005600 26 000000 55 000006   025900         STT 000000              55 000006              WORK AREA            C
005610 71 005330 00 000000   025910         TC  FNTG                                    FIND TAG IN TABLE       S
005620 61 005560 00 005560   025920         CTC C11                    C11              TAG FOUND               D
005630 44 000173 00 007473   025930         TCA $7                     DFF1,R           ADJUST AM7 TO FIRST     P
                             025940         RMK                                         REF FOR THIS TAG        C
005640 72 000000 70 600000   025950 C13     SET 000000              70 $T               IS FIRST REFERENCE      D
005650 43 011016 00 011016   025960         SC  #-1#                   #01#               AREA BLANK            C
005660 61 005720 00 005720   025970         CTC C14                    C14
005670 72 000003 70 600000   025980         SET 000003              70 $T               FILL IN FIRST           P
005700 26 003020 00 003023   025990         STT BOXNO                  BOXNO,R            REFERENCE BOX         C
005710 71 005560 00 000000   026000         TC  C11                                                             J
005720 22 000004 70 005735   026010 C14     OCT 000004              70 0 1,5            LAST REFERENCE          D
005730 72 006030 00 600000   026020         SET C15                    $T               FOR THIS TAG            Y
                             026030         RMK HAVE REACHED LAST      REFERENCE FOR THIS TAG
005740 36 000000 33 000007   026040         SCT 000000              33 000007           SET UP NEW LINK         P
005750 72 000003 30 600000   026050         SET 000003              30 $T               FOR THIS REFERENCE      C
005760 26 003020 00 003023   026060         STT BOXNO                  BOXNO,R
005770 25 000133 07 000007   026070         TCT $3                  07 000007           PUT ADDRESS OF NEW      P
006000 22 011020 07 000004   026080         OCT #40#                07 000004           ENTRY IN OLD ENTRY      C
006010 44 000133 00 007463   026090         TCA $3                     RTLA,R           INCREMENT AM3 FOR       P
                             026100         RMK                                            NEXT ENTRY           C
006020 71 005560 00 000000   026110         TC  C11                                                             J
006030 25 000007 70 000173   026120 C15     TCT 000007              70 $7               SET ADDRESS OF NEXT     P
                             026130         RMK                                         REFERENCE IN AM7        C
006040 71 005720 00 000000   026140         TC  C14                                                             J
006050 01 003035 01 010101   026150 C16     DAC LHT
006054 01 003045 01 010101   026160 C17     DAC RHT
                             026170         RMK CONSTRUCTION OF TBL  EI COMPLETE.  NOW BEGI  N PRINTING OF TABLE.   T
006060 22 004056 00 006075   026180 C20     OCT TT3                    0 1,5            REWIND TAPE             P
006070 17 000000 00 000000   026190         RWD                                         A
006100 25 004020 00 003775   026200         TCT NUMTAG                 TEMP1            SET TALLY TO NUMBER     P
006110 45 003777 00 010773   026210         TCS TEMP1,R                #000001#-1       OF TABLE ENTRIES-1      C
006120 25 007464 00 000133   026220         TCT RTMSC                  $3               MSC OF TABLE TO AM3     P
006130 36 010200 00 010367   026230         SCT CPLN                   CPLN,R           CLEAR PRINT LINE     P
                             026240         RMK START PROCESSING FOR   NEXT TAG IN TABLE                         N
006140 25 000133 00 000173   026250 C21     TCT $3                     $7               SET AM7 TO START OF  P
                             026260         RMK                                         ENTRY FOR THIS TAG   C
006150 25 010374 00 000153   026270         TCT CPLN1                  $5               SET AM5 TO MSC          P
                             026280         RMK                                            OF PRINT LINE        C
006160 72 000002 50 600000   026290         SET 000002              50 $T               MOVE PAGE #, BOX #,     P
006170 24 000010 77 000011   026300         STC 000010              77 000011           AND TAG TO PRINT LNE    C
006200 72 000006 50 600000   026310         SET 000006              50 $T
006210 24 000012 77 000013   026320         STC 000012              77 000013
006220 72 000021 50 600000   026330         SET 000021              50 $T
006230 24 000001 77 000007   026340         STC 000001              77 000007
006240 44 000173 00 007473   026350         TCA $7                     DFF1,R           SET AM7 TO 1ST REF      P
006250 44 000153 00 011023   026360         TCA $5                     #000030#-1       ADJUST AM5 TO FIRST     P
                             026370         RMK                                         REF ON PRINT LINE       C
006260 66 006310 00 006710   026380 C22     TA  C23                    OUT4             PRINT LINE COMPLETED    D
006270 71 006500 00 000000   026390         TC  OUT                                     PUT OUT LINE            S
006300 71 006260 00 000000   026400         TC  C22
006310 22 000000 75 000000   026410 C23     OCT 000000              75 000000           MOVE THIS REFERENCE     P
006320 22 000001 75 000001   026420         OCT 000001              75 000001            - TO OUTPUT AREA       C
```

```
NAME  AUTOFLOW                                        INDEX     40000501         SEG 07  DATE 111665     PAGE 097
 HSN    OP    A    N    B        SEQ #  TAG      OP   A ADDRESS                N  B ADDRESS            COMMENTS
 006330 22 011024 05 000002      026430          OCT  #16#                     05 000002
 006340 22 000002 75 000003      026440          OCT  000002                   75 000003
 006350 22 000003 75 000004      026450          OCT  000003                   75 000004
 006360 44 000153 00 011027      026460          TCA  $5                          #000010#/1          ADVANCE AMS TO NEXT     P
                                 026470          RMK                                                  O/P AREA FIELD          C
 006370 22 000004 70 006405      026480          OCT  000004                   70 * 1/5               IS THERE ANOTHER        D
 006400 72 006440 00 600000      026490          SET  C24                         $T                     REFERENCE            Z
 006410 71 006500 00 000000      026500          TC   OUT                                             PUT OUT LINE            S
 006420 66 006460 00 003775      026510          TA   C25                         TEMP1               WAS THIS LAST TAG       D
                                 026520          RMK                                                  END OF LISTING          N
 006430 71 007210 00 000000      026530          TC   ENDCC                                                                   J
 006440 25 000007 70 000173      026540 C24      TCT  000007                   70 $7                  PICK UP ADDRESS OF      P
                                 026550          RMK                                                    NEXT REFERENCE        C
 006450 71 006260 00 000000      026560          TC   C22                                                                     J
 006460 44 000133 00 007453      026570 C25      TCA  $3                          RTL,R               ADVANCE AMS TO NEXT     P
                                 026580          RMK                                                  TAG ENTRY IN TABLE      C
 006470 71 006140 00 000000      026590          TC   C21                                                                     J
                                 026600          RMK  WRITE LINE TO OUTPU      T  AND INITIALIZE FOR  NEXT LINE               T
 006500 25 000243 00 006650      026610 OUT      TCT  $STP                        OUTX                SET EXIT                P
 006510 25 010374 00 000153      026620          TCT  CPLN1                       $5                  RESET AMS TO FIRST    D
 006520 44 000153 00 011023      026630          TCA  $5                          #000030#,1          O/P AREA FIELD          C
 006530 72 000001 50 600000      026640          SET  000001                   50 $T                  ANY REFERENCES          D
 006540 43 011001 00-011001      026650          SC   #02#                        #02#                ON THIS LINE            C
 006550 61 006630 00 006560      026660          CTC  OUT2                        * 1
 006560 66 006600 00 006720      026670          TA   OUT1                        OUTS
 006570 71 006730 00 000000      026680          TC   CHP                                             END OF PAGE             D
 006600 71 006660 00 010000      026690 OUT1     TC   OUT3                        010000              CHANGE PAGE ROUTINE     S
 006610 22 004057 00 006625      026700          OCT  TT4                         * 1/5               B. P. 0 ON              D
 006620 12 010177 00 000000      026710          LW   CPLN/-1                                         WRITE LINE TO           P
 006630 36 010200 00 010367      026720 OUT2     SCT  CPLN                        CPLN,R              OFF-LINE LIST TAPE    C
                                                                                                      CLEAR PRINT LINE        P
 006640 25 006714 00 006710      026730          TCT  OUT4A                       OUT4                RESET TALLY             P
 006650 71 777777 00 000000      026740 OUTX     TC   777777                                          EXIT                    E
 006660 02 010200 00 000000      026750 OUT3     PR   CPLN                                            PRINT LINE              P
 006670 03 000002 00 000000      026760          PA   000002
 006700 71 006630 00 000000      026770          TC   OUT2                                                                    J
 006710 01 000014 01 010101      026780 OUT4     DAC  000014                                          TALLY-# ENTRIES ON LINE
 006714 01 000014 01 010101      026790 OUT4A    DAC  000014                                          INITIALIZER FOR ABOVE
 006720 01 000000 01 010101      026800 OUT5     DAC  000000                                          TALLY-# LINES ON PAGE
 006724 01 000031 01 010101      026810 OUT5A    DAC  000031                                          INITIALIZER FOR ABOVE
                                 026820          RMK                                                  END OF PAGE ROUTINE     T
 006730 25 000243 00 007100      026830 CHP      TCT  $STP                        CHPX                SET EXIT                P
 006740 51 007204 00 011031      026840          DA   CHP4,R                      #7424#              ADD 1 TO PAGE #         P
 006750 72 007746 00 600000      026850          SET  HDC12,6                     $T                  MOVE TO HEADER          P
 006760 24 007202 00 007203      026860          STC  CHP4,R-2                    CHP4,R-1
 006770 32 007745 00 007766      026870          ZS   HDC12,5                     HDC12,6
 007000 71 007110 00 010000      026880          TC   CHP2                        010000
 007010 22 004057 00 007025      026890          OCT  TT4                         * 1/5               B. P. 0 ON              D
 007020 12 007170 00 000000      026900          LW   CHP3                                            WRITE PAGE CHANGE       P
 007030 22 004057 00 007045      026910          OCT  TT4                         * 1/5                 MESSAGE             C
 007040 12 007577 00 000000      026920          LW   HDC1/-1                                         WRITE MAIN AND          P
 007050 22 004057 00 007065      026930          OCT  TT4                         * 1/5                 SUB HEADERS         C
 007060 12 007777 00 000000      026940          LW   HDC2/-1                                         &
 007070 25 006724 00 006720      026950 CHP1     TCT  OUT5A                       OUT5                RESET LINE TALLY        P
 007100 71 777777 00 000000      026960 CHPX     TC   777777                                          EXIT                    F
 007110 03 020000 00 000000      026970 CHP2     PA   020000                                          ON/LINE PAGE CHANGE     P
 007120 02 007600 00 000000      026980          PR   HDC1                                            PRINT MAIN AND          P
 007130 03 000002 00 000003      026990          PA   000002                                          SUB HEADERS             C
 007140 02 010000 00 000000      027000          PR   HDC2
```

```
NAME  AUTOFLOW                                       INDEX     40000501        SEG 07  DATE 111665       PAGE 058
  HSM   OP     A    M     B        SEQ #   TAG       OP  A ADDRESS          N   B ADDRESS              COMMENTS
007150 03 000002 00 000000        027010             PA  000002
007160 71 007070 00 000000        027020             TC  C,P1                                                                 J
007170                            027030 CHP3        CON 08                     7601010101012075
007200                            027040 CHP4        CON 05                 #   0174232301
                                  027050             RMK                        END OF CROSS             REFERENCE LISTING    T
007210 71 007360 00 000000        027 60 ENDCC       TC  ENDCCA                                          PAGE CHANGE       P
007220 72 011032 00 600000        027070             SET #23P                   $T                       IF AN EVEN # OF PAGE P
007230 42 007203 00 007203        027080             BS  CHP4,3                 CHP4,3                   WERE WRITTEN FOR THE C
007240 22 007203 00 007255        027090             OCT CHP4,3                 • 1,5                    CROSS REF LISTING   C
007250 75 000000 00 000000        027100             CSG                                                 GIVE AN EXTRA P/C    C
007260 65 007270 00 007300        027110             SSG • 1                    • 2                      FOR EDITING PURPOSES C
007270 71 007360 00 000000        027120             TC  ENDCCA
007300 75 000000 00 010000        027130             CSG                        010000
                                  027140             RIS LOADS              05  ENDP30                   BRING BACK SEGMENT 5 P
007310 72 001167 00 600000                           SET 001167                 $T
007320 24 007340 00 007357                           STC • 2                    • 3,7
007330 71 000340 00 000000                           TC  000340
007340 00 007450 00 000000                           OC  LOADS
007350 00 000005 00 013220                           OO  05                     ENDP30
                                  027150             RMK RETURN TO CODING AT    END OF PASS 3                                 N
                                  027160             RMK                                                      •ENDP30•        J
007360 25 000243 00 007440        027170 ENDCCA      TCT $STP                   ENDCCB                   PAGE CHANGE
007370 71 007430 00 010000        027180             TC  • 4                    010000                       SUBROUTINE
007400 22 004057 00 007415        027190             OCT TT4                    • 1,5
007410 12 007170 00 000000        027200             LW  CHP3
007420 71 007440 00 000000        027210             TC  • 2
007430 03 020000 00 000000        027220             PA  020000
007440 71 777777 00 000000        027230 ENDCCB      TC  777777
007450 01 000024 01 010101        027240 RTL         DAC 000024                                          LENGTH NEW TABLE ENTRY
007454 01 000034 01 010101        027250 MTL         DAC 000034                                          LENGTH OLD TABLE ENTRY
007460 01 000010 01 010101        027260 RTLA        DAC 000010                                          LENGTH OF EXTRA ENTRIES
011050                            027270 RT          DAR 8,0                    END SEG
007464 01 011050 01 010101        027280 RTMSC       DAC RT                                              MSC OF TABLE
007470 01 000014 01 010101        027290 DFF1        DAC 000014
007474                            027300             DAR 1
007600                            027310 HDC1        DAR 120,200                                         MAIN HEADER
                                  027320             SLC HDC1,-1            #   76
007577                            027330             CON 1                      PROGRAM
007600                            027340             CON 07
        57615646614054
007607                            027350             CON 01                 #   06
                                  027360             SLC HDC1,50
007662                            027370             CON 15                     CROSS REFERENCE
        4261566262C1614445446144554244
007701                            027380             CON 8                       LISTING
        0153506263505546
                                  027390             SLC HDC1,90
007732                            027400 HDC11       CON 04                     DATE
        43406344
                                  027410             SLC HDC1,112
007760                            027420 HDC12       CON 04                     PAGE
        57404644
                                  027430             SLC HDC1,R 1
007770                            027440             CON 02                 #   7475
007772                            027450             DAR 1
010000                            027460 HDC2        DAR 120,200                                         SUB HEADER
                                  027470             SLC HDC2,-1
007777                            027480             CON 01                 #   76
```

```
NAME  AUTOFLOW                                        INDEX     40000501          SEG 07  DATE 111665      PAGE 059
  HSM    OP     A    N     B         SEQ #  TAG        OP  A ADDRESS          N   B ADDRESS                  COMMENTS
 010000                              027490            CON 15                     PAGE BOX    TAG
         5740464401415667010101016 34046
 010017                              027500            DAR 13
 010034                              027510            CON 1?                     REFERENCES
         61444544614455424462
                                     027520            SLC HNC2,R 1
 010170                              027530            CON 2                  =   7475
 010172                              027540            DAR 1
 010200                              027550 CPLN       DAR 120,200
                                     027560            SLC CPLN,-1                                         OUTPUT AREA
 010177                              027570            CON 01                 #   76
                                     027580            SLC CPLN,R 1
 010370                              027590            CON 02
 010374 01 010200 01 010101          027600 CPLN1      DAC CPLN               #   7475
 010400                              027610            DAR 200
 010710                                     #0138      CON 22,0                                            FOR PATCHES
        01010101010101010101010101010101010101010101
 010740                                     #0139      CON 22,0                   0000000000000000000000
        23232323232323232323232323232323232323232323
 010771                                     #0140      CON 03,1               #   000001
 010775                                     #0141      CON 03,1               #   000000
 011000                                     #0142      CON 01                 #   76
 011001                                     #0143      CON 01                 #   02
 011002                                     #0144      CON 01                 #   72
 011005                                     #0145      CON 03,1               #   000002
 011010                                     #0146      CON 01                 #   71
 011011                                     #0147      CON 01                 #   22
 011012                                     #0148      CON 01                 #   73
 011013                                     #0149      CON 03                     DJS
        435162
 011016                                     #0150      CON 02                 #   0101
 011020                                     #0151      CON 01                 #   40
 011021                                     #0152      CON 03,1               #   000030
 011024                                     #0153      CON 01                 #   16
 011025                                     #0154      CON 03,1               #   000010
 011030                                     #0155      CON 02                 #   7424
 011032                                     #0156      CON 01                 #   23
 011040 00 000000 00 000007                            CON 8,1?               #   0000000000000007
```

/hat is claimed is:

. For use in a system for automatically controlling ›mputer having a storage, a processor, an output unit, control apparatus for controlling the operation of . processor, storage and output unit to perform ıences of operations on blocks of data in the form :oded digital signals stored in said storage and repre- :ing successive instructions of various types of a com- ›r program including process, unconditional transfer conditional branch instructions; means for directing operation of said control apparatus to process said a blocks sequentially and to produce a record of a v chart representative of said program, the method said directing means comprising:

›rocessing said data blocks to identify the types of instructions represented thereby and to establish flow chart symbols therefor;

.llocating successive ones of said symbols forming a main flow and branch sequences of the computer program as arrays in sections of each of successive flow chart pages;

dentifying branch sequences of the computer program associated with branch instructions of said main flow sequences and initiating said allocating of successive symbols of said branch sequences to other sections of the associated flow chart pages to form a symbol array with said main flow array; and producing a record of said symbol arrays in successive flow chart pages in accordance with the allocation thereof including producing chart indications of the relationships of branch and main flow symbols.

2. The directing means having the method as recited claim 1 wherein the identifying of said branch ɪuences includes identifying sub-branch sequences of ; computer program that are associated with branch ›tructions of said branch sequences and allocating suc- ›ssive symbols of said sub-branch sequences to other :ctions of the associated flow chart page to form a mbol array with the branch symbol array.

3. The directing means having the method as recited claim 2 wherein said allocating includes forming ısters of main flow and branch sequences with each ıster being formed as a sequence of main flow symbols ıd available sequences of branch and sub-branch sym- ›ls associated with branch instructions of said main ›w and branch sequences, respectively, and allocating id clusters successively to said chart pages.

4. The directing means having the method as recited . claim 1 and further comprising:

determining the amount of page space required for each of said symbols;

identifying chains of said symbols occurring between successive transfer instructions and determining the amount of page space required for the symbols of each chain;

and wherein said branch sequence identifying includes determining in connection with each branch instruction of the main flow whether the page space required for the symbols of the associated branch chain fits within the space of an unallocated section of the associated chart page.

5. The directing means having the method as recited ı claim 2 and further comprising:

determining the amount of page space required for each of said symbols;

identifying chains of said symbols occurring between successive transfer instructions and determining the amount of page space required for the symbols of each chain;

wherein said branch sequence identifying further includes determining in connection with each branch instruction of the main flow and branch chains whether the page space required for the symbols of the associated branch and sub-branch chain, respectively, fits within the space of an unallocated section of the associated chart page.

6. The directing means having the method as recited in claim 4 wherein said allocating includes, upon allocating a branch instruction symbol, determining the fit of a branch chain.

7. The directing means having the method as recited in claim 6 wherein said branch symbol allocating is performed upon the allocating of a branch instruction of the main flow sequence, and said main flow symbol allocating continues upon the completion of the branch chain allocation.

8. The directing means having the method as recited in claim 6 wherein said branch symbol allocating is performed upon the completion of a page section of main flow allocating.

9. The directing means having the method as recited in claim 6 wherein said data block processing includes scanning a predetermined field of each of said data blocks to establish different flow chart symbols therefor or to reject said data block, respectively, in accordance with different coded signal groups in said field, and extracting coded signal portions from said field for certain types of instructions; and wherein said record producing includes means for producing indications on said chart representative of the extracted signal portions.

10. For use in a system for automatically controlling a computer having a storage, a processor, an output unit, and control apparatus for controlling the operation of said processor, storage and output unit to perform sequences of operations on blocks of data in the form of coded digital signals in a certain format stored in said storage and representing successive instructions of various types of a computer program including process, unconditional transfer and conditional branch instructions; a method for directing the operation of said control apparatus to process said data blocks sequentially and to produce a record of a flow chart representative of said program, said method comprising:

processing said data blocks successively included scanning predetermined fields of each of said data blocks to establish a flow chart symbol therefor or to reject said data block, respectively, in accordance with different coded signal groups in said fields;

allocating successive ones of said symbols forming a sequence of the computer program as an array and allocating each of successive ones of said sequences as an array in a section of each of successive flow chart pages;

and producing a record of said symbol arrays in successive flow chart pages in accordance with the allocation thereof.

11. The method as recited in claim 10 wherein said scanning of said data block field includes establishing different flow chart symbols in accordance with different coded signal groups in said field.

12. The method as recited in claim 10 wherein said scanning of said data block field includes extracting from said data block field a coded signal portion representing an instructional commentary and extracting therefrom a coded signal portion representing a supplementary commentary to append it to the extracted portion of a preceding data block, respectively, in accordance with different coded signal groups in said field; and wherein said record producing includes producing symbols with representations of said commentary portions set forth therein and of variable size in accordance with the lengths of the commentary portions.

13. The method as recited in claim 10 wherein said scanning of said data block field includes identifying the types of instrutcions represented by said data blocks and extracting from said data block field for transfer and branch instructions; a coded signal portion identifying the transferee instruction; and wherein said record producing includes producing from said identifying portions

indications of the relationships of the transferor and transferee symbols.

14. The method as recited in claim **10** wherein said scanning of said data block field includes identifying the types of instructions represented by said data blocks and extracting from said data block field for branch instructions a coded signal portion identifying the operational conditions of the branch instruction; and wherein said record producing includes producing from said identifying portion labels of the branch conditions at the associated branch instruction symbol.

15. For use in a system for automatically controlling a computer having a storage, a processor, an output unit, and control apparatus for controlling the operation of said processor, storage and output unit to perform sequences of operations on blocks of data in the form of coded digital signals stored in said storage and representing successive instructions of various types of a computer program including process, unconditional transfer and conditional branch instructions; the method of directing the operation of said control apparatus to process said data blocks sequentially and to produce a record of a two-dimensional flow chart representative of said program, said method comprising:

processing said data blocks to identify the types of instructions represented thereby and to establish flow chart symbols therefor;

forming sequences of successive related symbols and allocating some of said sequences as arrays in sections of successive flow chart pages;

identifying branch sequences associated with branch instructions of others of said sequences and allocating said branch sequences to other sections of the associated flow chart pages to form two-dimensional arrays;

and producing a record of said symbol arrays in successive flow chart pages in accordance with the allocation thereof including producing chart indications of the relationships of branch sequences to the associated branch instruction symbols.

16. The method as recited in claim **15** wherein said sequence forming and allocating allocates the symbols of said sequences in an order corresponding to that of said data blocks and said branch sequence identifying and allocating includes allocating said branch sequence symbols subsequent to the allocation of and on the same page as the associated branch instruction.

17. In a data processing system for producing flow charts of a computer program and having a storage, a processor, an input and an output unit, and control apparatus for controlling the operation of said processor, storage and input and output units to perform sequences of operations on input blocks of data in the form of coded combinations of digital signals stored in said storage and representing successive operations of various types employed in a computer program, including conditional branch instructions;

said control apparatus being operative with said processor and storage for determining the size of page space to display flow chart symbolic representations of said data blocks and of one-dimensional arrays of said symbolic representations in relation to subdivisions of a flow chart page, and operative with said processor and storage for allocating successive ones of said symbolic representations of a one-dimensional array thereof to sections of a flow chart page, and operative with said processor and output unit for producing an output record of pages of said one-dimensional flow chart arrays;

control means for directing the operation of said data processing system to produce an output record of a two dimensional flow chart representative of said computer program, the method of said flow chart control means comprising:

initiating operation of said allocating to allocate one-dimensional branch sequence arrays having symbolic representations for data blocks referenced by said branch instruction data blocks of parent sequences to unallocated sections of the same pages as the parent one-dimensional arrays that include said branch instruction representations;

producing an output record of successive flow chart pages in accordance with said allocations of said one-dimensional sequence arrays and said one-dimensional branch sequence arrays with chart indications of the relationships of the symbolic representations of the branch sequence arrays to those of the associated branch data block representations in the parent arrays on the same pages; whereby clusters of one-dimensional arrays and associated one-dimensional branch arrays are formed each for display on a page to provide a two-dimensional flow chart display.

18. The method as recited in claim **17** wherein said initiating of said allocating includes initiating allocating of one-dimensional sub-branch sequence arrays having symbolic representations for data blocks referenced by said branch instruction data blocks of parent branch sequences to unallocated sections of the same pages as the parent branch arrays that include said branch instruction representations; and producing records of said pages in accordance with the allocations of said sub-branch arrays and in relation to parent arrays on the same pages; whereby clusters may be formed with sub-branch arrays.

19. For use in a data processing system having a storage, a processor, an input and an output unit, and control apparatus for controlling the operation of said processor, storage and input and output units to perform sequences of operations on input blocks of data in the form of coded combinations of digital signals stored in said storage and representing successive operations of various types of a program including branch operations,

an automatic control method for directing the operation of said data processing system to process said data blocks sequentially and to produce a record of a two-dimensional flow chart representative of said program for reproduction on a rectangular page of a known size in a plurality of similar rectangular columns, said automatic control method comprising:

processing said data blocks to establish flow chart symbolic representations thereof for display within a width corresponding to that of one of said columns, including processing sequences of said data blocks of varying lengths to form one-dimensional arrays of said symbolic representations for display within said column width;

allocating to the columns of said flow chart pages said one-dimensional arrays for parent sequences of said program, including identifying branch ones of said one-dimensional arrays for data block sequences having operations referenced by branch operations of said parent sequences and having lengths that fit within unallocated portions of said columns on the same pages as said parent arrays and allocating the identified branch arrays to said unallocated column portions; and producing an output record of successive flow chart pages in accordance with the allocations of said parent and branch arrays with chart indications of the relations of associated parent and branch arrays on the same pages; whereby clusters of parent and branch arrays are formed for a two-dimensional flow chart display.

20. An automatic control method for a data processing system as recited in claim **19**, wherein said allocating includes identifying sub-branch ones of said one-dimensional arrays for data block sequences having operations

:renced by branch operations of said branch sequences . having lengths that fit within unallocated portions of l columns on the same pages as said branch arrays and ıcating the identified sub-branch arrays to said un- ıcated column portions; and said output record pro- ing includes producing records of said pages in ac- dance with the allocations of said sub-branch arrays h chart indications of the relations of associated branch l sub-branch arrays on the same pages.

1. An automatic control method as recited in claim wherein said data block sequence processing to form -dimensional arrays includes determining the column gth of each of said arrays in terms of lines of a page ımn for display of the symbolic representations there- and determining the line length of each of said sym- ic representations from the associated data blocks.

2. An automatic control method as recited in claim wherein said data block sequence processing further ludes storing in said storage a table of identifiers for l data block sequences including an identifier for the ımn lengths determined for the associated one-dimen- ıal arrays; and said allocating includes determining unallocated column lengths, locating within said ·age table said identifiers of branch sequences and ıparing the column lengths thereof with the unallo- ed lengths.

:3. An automatic control method as recited in claim wherein said sequence processing includes identifying h of said data blocks representative of operations of :ertain type and generating said sequence identifiers each group of sequential data blocks between suc- sive ones of said certain type, whereby data block uences of varied lengths are automatically formed in ordance with the character of the program.

:4. An automatic control method as recited in claim wherein said allocating and said column length com- ing include allocating a portion of a one-dimensional ay to one column and a remaining portion of the array a succeeding column, and generating and allocating a ıbolic representation of a connective relation between l array portions, whereby an array that exceeds the illocated column length may be broken in portions display in separate columns.

:5. An automatic control method as recited in claim wherein said table storing includes storing a table of identifiers of data blocks in said branch sequences erred to by branch operations of said parent sequences; aid allocating further includes supplying to said tag table identifiers of page allocations of the symbolic representations in said branch arrays for the tag data blocks;

ınd said output record producing includes generating and allocating symbolic representations of connective relations between the associated referring and tag symbolic representations in the parent and branch arrays.

:6. An automatic control method as recited in claim wherein said connective relation generating includes ıerating connector line symbols where the referring l tag symbolic representations are on the same page l in suitable relation, and generating and allocating ınector symbols at the parent referring symbolic rep- entations and recording thereat identifiers of the page ocations of the associated tag symbolic representations oranch arrays.

:7. An automatic control method as recited in claim wherein said allocating further includes supplying to l tag table identifiers of page allocations of parent erring symbolic representations associated with tag ıbolic representations;

ınd said output record producing further includes producing a record for each tag data block of page allocations of all associated referring symbolic representations.

:8. For use in a data processing system having a storage, a processor, an input and an output unit, and control apparatus for controlling the operation of said processor, storage and input and output units to perform sequences of operations on input blocks of data in the form of coded combinations of digital signals stored in said storage and representing successive operations of various types of a program including branch operations;

an automatic control system for directing the operation of said data processing system to process said data blocks sequentially and to produce a record of a two dimensional flow chart representative of said program for reproduction on a page of a known size, the method of said automatic control system comprising:

processing said data blocks to identify the types of operations represented thereby and to establish flow chart symbols therefor;

identifying sequences of said data blocks and storing in said storage a table of identifiers of said data block sequences, said identifying including determining the size of a one-dimensional array of the symbols for each of said data block sequences in relation to subdivisions of a flow chart page;

allocating successive ones of said symbols for parent and branch sequences of the program as parent and branch-sequence arrays, respectively, in sections of each of successive flow chart pages, including locating within said storage table branch sequences of the program associated with branch operations of said parent sequences and of sizes that fit within unallocated sections of the associated pages and initiating the operation of said allocating means to allocate successive symbols of the associated branch sequence arrays to other sections of the associated flow chart pages to form a two-dimensional array with said parent array;

producing a record of said symbol arrays in successive flow chart pages in accordance with the allocations thereof including producing chart indications of the relationships of branch and parent symbols;

whereby flow chart records are produced with parent and branch arrays on the same pages and corresponding to parent and branch sequences of data blocks in different sequential locations in said program.

29. The method of an automatic control system for a data processing system as recited in claim **28**, wherein said sequence identifying includes generating said sequence identifiers for each group of sequential data blocks between successive ones of a certain type of operation, whereby data block sequences are automatically formed in accordance with the program.

30. In a method of controlling a data processing system to produce flow charts of a computer program, said system having a storage, a processor, an input and an output unit, and control apparatus for controlling the operation of said processor, storage and input and output units to perform sequences of operations on input blocks of data in the form of coded combinations of digital signals stored in said storage and representing successive instructions of various types of a computer program, including transfer instructions;

wherein the size of page space for reproducing flow chart symbolic representations of said data blocks and of sequences of said symbolic representations is determined in relation to subdivisions of said flow chart page, successive ones of said symbolic representations of any array thereof are allocated to sections of said flow chart pages, and a record of said pages of arrays is produced by means of said output unit;

the method of directing the operation of said data processing system to produce a record of a two-dimensional flow chart representative of said computer program, said method comprising:

forming chain sequences of said data blocks;

allocating chain sequences of said data block representations to certain sections of said pages;

allocating chain sequences of said data block representations associated with transfer instructions of previously allocated chain sequences to unallocated sections of the same pages;

and producing a record of successive flow chart pages in accordance with said chain sequence allocations with chart indications of the relationships of the transfer chain sequences to the associated transfer instruction data block representations.

31. The method set forth in claim **30**, wherein said step of forming chain sequences includes determining the page space size thereof in relation to said page subdivisions; and wherein said step of allocating transfer chain sequences includes the step of first determining whether the page sizes thereof are within those of the unallocated page sections.

32. The method set forth in claim **30**, wherein said transfer instructions of the previously allocated chain sequences are conditional transfer instructions; and

wherein said step of forming chain sequences includes forming chain sequences between successive unconditional transfer instructions.

33. The method set forth in claim **32**, wherein said step of forming chain sequences includes forming a plurality of shorter chain sequences from a longer chain sequence; and identifying on said pages the sequential relationships of the shorter chain sequences.

34. A data processor method of automatically producing two-dimensional flow charts of operations of a computer program, each of said program operations being represented by an input block of data in the form of coded combinations of digital signals; said method being performed with a digital computer and comprising:

forming signal records of sequence chains of flow chart symbols, each of said chains being representative of a different sequence of said data blocks;

allocating some of said sequence chains together with associated branch path ones of said sequence chains to portions of the same flow chart pages;

and identifying interrelationships of said sequence chains and the associated branch path sequences.

35. A data processor method as set forth in claim **34**, wherein said step of chain allocating includes the step of allocating sub-branch path sequence chains to portions of the same flow chart pages to which associated branch path sequence chains are allocated.

36. A data processor method of automatically producing two-dimensional flow charts of operations of a computer program, each of said program operations being represented by an input block of data in the form of coded combinations of digital signals; said method being performed with a digital computer and comprising:

forming signal records of sequence chains of flow chart symbols, each of said chains being representative of a different sequence of said data blocks;

allocating clusters of interrelated ones of said sequence chains to flow chart pages; and identifying interrelationships of said sequence chains in each of said clusters.

37. A data processor method as set forth in claim **36**, wherein said cluster allocating step includes allocating a cluster to each flow chart page; and assembling clusters from said sequence chains associated in the same data block sequence chains and in branch and sub-branch sequence chains.

38. A data processor method as set forth in claim **37**, wherein said cluster assembling step includes forming a plurality of shorter sequence chains from a longer sequence chain.

39. A data processor method as set forth in claim **38**, wherein said cluster assembling step includes assembling a cluster from at least one sequence chain and at least one associated branch sequence chain of combined page space within the space size of said chart pages.

40. A data processor method as recited in claim **36**, wherein said program operations include transfer operations; and the step of forming records of sequence chains includes forming sequence chains as a sequence of flow chart symbols between successive transfer operations.

41. A data processor method as recited in claim **36**, and further comprising producing from said signal records a display of said flow chart symbols on successive rectangular pages each comprising a plurality of rectangular columns with each column being of a width suitable for display of various ones of said symbols;

and wherein said step of allocating clusters of sequence chains includes allocating branch path ones of said interrelated sequence chains having lengths that fit within unallocated portions of said columns on the same pages.

42. A data processor method as recited in claim **37**, and further comprising producing from said signal records a display of said flow chart symbols on successive rectangular pages each comprising a plurality of rectangular columns with each column being of a width suitable for display of various ones of said symbols;

and wherein said step of assembling clusters includes incorporating in a cluster a sequence chain and associated branch and sub-branch chains having lengths that fit within portions of said columns on the same page.

43. A data processor method as recited in claim **34**, wherein said program operations include transfer operations; and the step of forming records of sequence chains includes forming sequence chains as a sequence of flow chart symbols between successive transfer operations.

44. A data processor method as recited in claim **34**, and further comprising producing from said signal records a display of said flow chart symbols on successive rectangular pages each comprising a plurality of rectangular columns with each column being of a width suitable for display of various ones of said symbols;

and wherein said step of allocating sequence chains includes allocating those of said associated branch path sequence chains having lengths that fit within unallocated portions of said columns on the same pages.

45. A data processor method as recited in claim **35** and further comprising producing from said signal records a display of said flow chart symbols on successive rectangular pages each comprising a plurality of rectangular columns with each column being of a width suitable for display of various ones of said symbols;

said step of allocating branch and sub-branch sequence chains including the allocating of those of said associated branch and sub-branch chains having lengths that fit within unallocated portions of said columns on the same page.

References Cited

Krider, L., A Flow Analysis Algorithm, Journal of the Association for Computing Machinery, vol. II, No. 4, October 1964, pp. 429–436.

IBM 7070/7074 Autochart Programming System, file No. 7070/7074–48, Form C28–6772–1, IBM Corp., February 1964.

Scott, A. E., Automatic Preparation of Flow Chart Listings, Journal of the Association for Computing Machinery, January 1958, pp. 57–66.

(Other references on following page)

References Cited

:nuth, D. E., Computer Drawn Flow Charts, Communications of the ACM, September 1963, pp. 555–563.

\nderson, H. E., Automated Plotting of Flow Charts a Small Computer, Communications of the ACM, uary 1965, pp. 38–39.

;ant, W. T., Flow Outlining—A Substitute for Flow ırting, Communications of the ACM, November 1959, 17.

;aalbach, C. P. and Sapovchak, B. J., The Flow Chart ıgram, Proceedings of 4th Annual Meeting of UAIDE, .ober 1965, pp. XXII-1—XXII-8.

Haibt, Lois M., A Program to Draw Multilevel Flow Charts, Proceedings of the Western Joint Computer Conference, 1959, pp. 131–137.

Hain, G. and Hain, K., Automatic Flow Chart Design, Proceedings of ACM Conference, August 1965, pp. 513–523.

Hain, G. and Hain, K., A General Purpose Automatic Flow Charter, Proceedings of 4th Annual Meeting of UAIDE, October 1965, pp. IV–1—IV–11.

RAULFE B. ZACHE, Primary Examiner