INTERNATIONAL BUSINESS MACHINES CORPORATION

CUSTOMER ENGINEERING EDUCATION DEPARTMENT

IBM EDUCATION CENTER

POUGHKEEPSIE, NEW YORK

STUDENT STUDY GUIDE

9IOCS PACKAGE PROGRAM

COURSE OBJECTIVES:

To present to the student a detailed understanding of the 709/7090 Input/Output Control System. In addition, he is exposed to techniques useful in the analysis of a program whether IBM's or the customer's. During the course of the investigation of this package, he is offered an opportunity to read program listings, thereby becoming familiar with methods employed by programmers to accomplish their objectives.

R. F. Buchan	3/16/62
Written by	Date
R H Eckhart	3-28-62
Reviewed by	Date
K. L. LeComb	3/28/62
Approved by	Date

REFERENCE MATERIAL

1. 709/7090 Input/Output Control System Reference Manual - Form #C28-6100-2

2. 9IOCS Flow Chart Package

STUDY GUIDE CONTENTS

9IOCS Package Program

Lecture Summaries

Assignments

Laboratory Projects

Supplemental Instruction Material

LEGEND

1.	R. M.	-	709/7090 Input/Output Control System Reference Manual
2.	S. G.	-	Contained within this study guide (9IOCS Package Program)
3.	F. C.	-	9IOCS Flow Chart Package

I. INTRODUCTION

Objective: To acquaint the student with the basic objectives of the course and to condition his thinking toward programming lines. The latter will be accomplished through a review of the standard flow charting symbols and the introduction of new symbols used in the course. Simple loops will be presented for charting by the students.

q A. COURSE OBJECTIVES

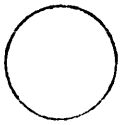
1. Exposure to techniques useful in the analysis of a program.
   a. Techniques used in class to analyze the internal structure of the program will offer useful suggestions as:
      (1) Use of charts to keep a progressing record of ACC, MQ, SR, IR, SI, etc. content
      (2) Use of programmer's remarks
      (3) Placing notes on listings to recall changes of data in cells, etc.
      (4) Use of flow charting

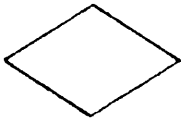
2. An extended opportunity to read program listings both with the aid of the instructor in class and individually during outside study.

3. Gain a detailed understanding of the 9IOCS package program.
   a. Definition
      (1) The 9IOCS package consists of a series of subroutines that, when employed in a customer program, offer a simplified method of attaining maximum utilization of all major components of the 709/7090 system.
   b. Basic concepts of Input/Output Control Systems
   c. Overall functions of 9IOCS followed by those of the individual routines.
   d. Programming of a sample job using this package.
   e. Theory associated with 9IOCS.
   f. Investigation of the program through the use of its listings and associated flow charts.

B. FLOW-CHARTING

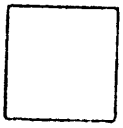
1. Basic Symbols

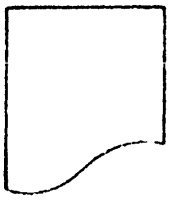
a.  Connector - Connect between one section of a flow chart and another.

b.  Decision - Result of some operation

c.  Program Step

d.  Halt

e.  Console Operation

f.  Document

g.  Input/Output Operation

2. New Symbols

a. Subroutine Symbol

(1) Purpose

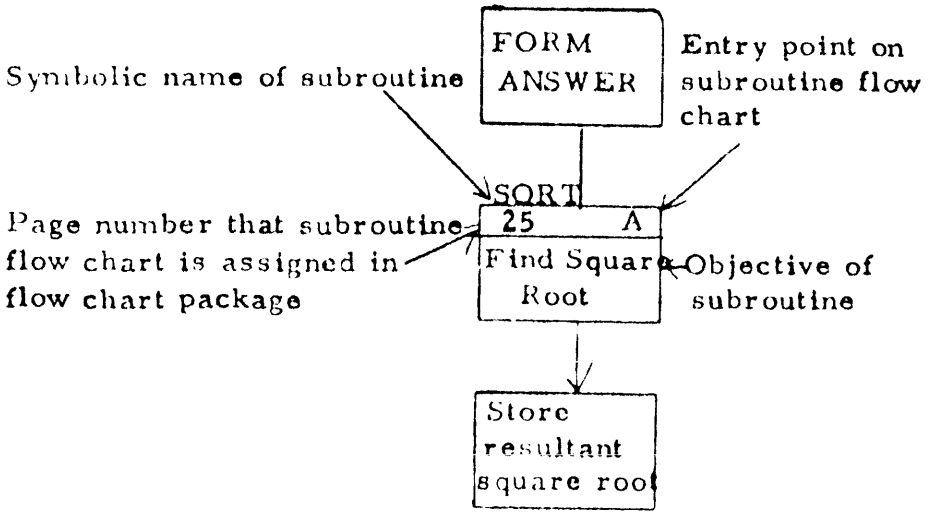
(a) Represent an entire subroutine with one block

(b) Makes flow-charting a common subroutine each time it is used unnecessary

(c) Common subroutine flow-charted once can be referred to as required by this symbol.

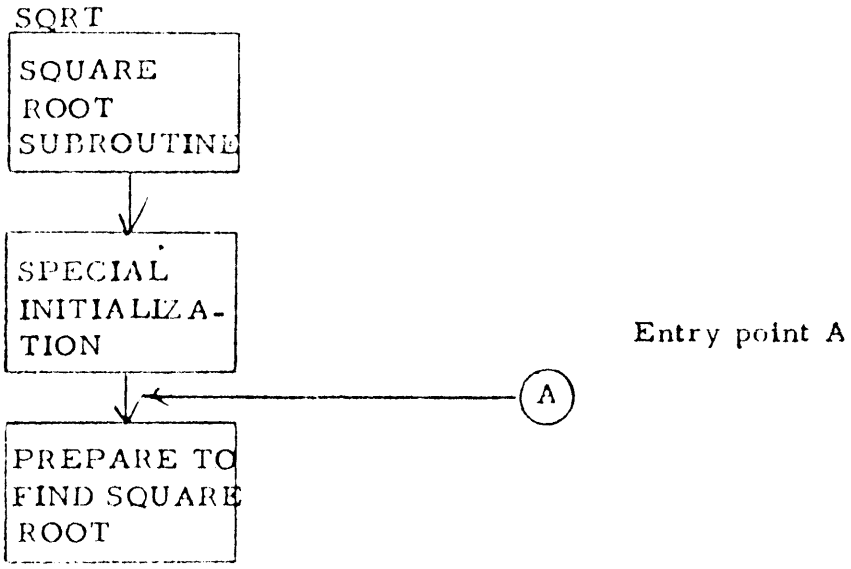
(2) An example of its use in main program follows

(a)



(3) An example of the subroutine flow-chart to tie in data illustrated in subroutine block follows

(a)



3. Student Practice

a. Students are to use the above symbols to flow-chart simple problems.

(1) Suggested Problems

(a) A + B x C = D Find A + B, print sum using subroutine "PRNT", then multiply sum by C, print result D using subroutine "PRNT", and stop. Assume "PRNT" is written and located on page 115. Use entry point Q.

(b) Read a record from Tape A1 and one from B1. Subroutine "COMP" should be used to compare record numbers. This subroutine has 3 returns to the main program (A1 = B1, A1 < B1, A1 > B1). If A1 = B1 stop at 000. If A1 < B1 stop at 001. If A1 > B1 stop at 002.

II. BASIC CONCEPTS OF INPUT/OUTPUT CONTROL SYSTEMS

Objective: To familiarize the student with the basic concepts of this type of programming package. Its advantages and basic principles of operation will be stressed.

A. BASIC SERIAL JOB

1. Definition of "Serial"
   a. All functions in the program are performed in a sequential fashion. Each major component of the computer system operates individually.

2. Following sample job is to be considered in class.
   a. Hypothetical Inventory Control Job

3. Develop flow chart as a group to perform the job serially. Reference

4. Calculate and record the amount of time it would take to process an "update" situation. Assume that Model 2 tape drives are to be used in low density (10.8 Ms/record access time - 67.2 us/character). In respect to tape time, computation time can be ignored.

B. TIMESAVING DEVICES

1. Hardware Improvements
   a. Engineering improvements are not to be considered in this course. Simultaneous I/O and compute, channel trap, etc. are some examples.

2. Programming Improvements
   a. Blocking
      (1) Defined
         (a) Many logical records within one physical tape record
      (2) Advantage
         (a) Tape access time now occurs once per block rather than once per record
      (3) Blocking Factor
         (a) Number of logical records within one physical record

(4) Proof of Advantage

(a) Assume a theoretical blocking factor of 50 for all input and output files. Refer to "Inventory Job" and note that 50 items could now be processed in the time it took one to be performed serially. Additional computing time is necessary but relatively small.

b. Buffering

(1) "Buffer"

(a) Intermediate storage area where input/output data is held till operation on it is required.

(2) Swing Buffering

(a) Defined - Two buffer areas used. One being filled while content of other is being used.

(b) Advantage - This, together with blocking, is an aid toward approaching our ideal situation of having all major units of the computer (CPU, I/O) operating continuously and simultaneously

(c) Depending on the relationship between compute and I/O time, it is still possible to become I/O bound (CPU waiting for data).

(I) Adding additional buffers to be filled in advance could relieve this situation. However, core capacity must be considered.

(3) Pool Buffering

(a) Defined - Useful when multiple files of varying activity are employed in a program. A buffer pool consists of a preset number of buffers. Any unused buffer is available to one of a preset number of files "attached" or assigned to the pool. Any of the files may use more than one buffer if necessary. The balance of the number of buffers within the pool assigned to each of the corresponding files is determined by the relationship of file activity. If the CPU demand

for data transfer to or from a particular file is high, many buffers will be "in use" by that file. The other files sharing the pool will employ as many of the remaining buffers as their activity demands. 9IOCS automatically monitors and adjusts this balance as required.

(b) Advantage - A maximum availability of buffers to a file when its activity is high with a minimum of storage allocation.

(c) Pool buffering is the basis of the 9IOCS package.

(d) Note the complicated programming necessary to maintain this type of operation. Through the use of 9IOCS, this operation becomes available to the customer at any time in a program through a simple calling sequence composed of a TSX followed by a series of control words. In addition, IOCS offers automatic redundancy recovery and end of file routines, together with a simplified method of operating on I/O data.

III. FUNCTIONS OF MINIMUM 9IOCS ROUTINES

Objectives: To point out the various functions of the individual 9IOCS routines

A. GENERAL

1. Method of calling routines - (TSX XX)

2. Definition of "Calling Sequences"
   a. TSX to subroutine and associated control words that follow
   b. State parameters

B. DEFINE

1. Pool Makeup — Ref.
   a. 2 pool control words per pool
   b. 2 buffer control words per buffer
   c. Formula to find pool size. M (N + 2)+2 = NUMBER OF CELLS
      M = number of buffers in pool
      N = number of cells in each

   Note: Do not go into content of control words at this time.

2. "Define" function
   a. Form pool

3. Consider "Define" calling sequence — Ref.

C. ATTACH

1. "File"
   a. Data associated with some I/O device
   b. Types of files
      (1) Input (I) - Each type I file requires one buffer from pool. One physical record is read in per buffer.
      (2) Output
         (a) Partial Output File (P) - Each type P file requires at least one buffer at all times. Buffer can be filled by individual logical records. When full, it will be written as physical record.

(b) Total Output File (T) - All type T files attached to a particular pool require one buffer at all times. Entire physical record must be placed in buffer then written immediately.

c. File grouping
   (1) Immediate
      (a) Regular Input/Output file
   (2) Reserve (Discuss briefly)
      (a) Special purpose files
      (b) Files specified in groups
      (c) Must specify number of buffers to be reserved per group
   (3) Internal (Discuss briefly)
      (a) Files contained entirely within core. No I/O unit associated with them.

Note: From this point, emphasis will be placed on Immediate files.

2. Functions of Attach
   a. Tie all files using a specific pool to the pool.
   b. In the case of Immediate files, "Open" or in other words, initiate I/O operations for that file.
      (1) Rewind
      (2) Set Density
      (3) For input files read in first physical record.

3. "Attach" calling sequence
   a. File List
      (1) List of files to be attached to the same pool
   b. Consider its calling sequence

Ref.

Ref.

4. Internal and Reserve files must be attached and then opened separately thru use of the Open routine.

D. READ

1. Function
   a. Transmit words from buffers to working storage.
   b. Locate words within buffers.
   c. Fill additional buffers as required.

2. Calling sequence
   a. Consider PZE words but do not dwell on EOB at this point. Ref.
   b. Note the function of the command list. The individual commands will be covered later.

E. WRITE

1. Function
   a. Transmit words from working storage to buffers.
   b. Locate empty cells within buffers.
   c. Write out buffers as required.

2. Calling sequence
   a. Consider PZE word but do not dwell on EOB at this point. Ref.
   b. Note that a command list is also used as part of this calling sequence.

F. COMMANDS

1. Function
   a. Control the transfer of words between buffers and working storage.
   b. Locate specific words within buffers.

2. Format Ref.
   a. Type of Control
      (1) C - Count Control
      (2) R - Buffer Control
      (3) S - Special Count Control
   b. Function to be performed at completion of command.
      (1) P - Proceed
      (2) T - Terminate
      (3) D - Terminate and Truncate

   c. N - Non-transmitting - will be covered in detail later.
   d. * - Indirect Addressing
   e. A - Address
   f. m - word count

3. Transmitting commands - Transmit data between buffers and working storage.
   a. Count Control - IOCP, IOCT, IOCD Ref.
   b. Buffer Control - IORP, IORT Ref.
   c. Special Count Control - IOSP, IOST Ref.
   d. Transfer and Continue Command List - TCH Ref.
      (1) Note that the exit from the IOCS routine is to the location after the first TCH command in the list.

4. Non-transmitting commands
   a. Functions
      (1) Locating - Means of processing information in buffers.
         (a) Determining the location in a read buffer of the first word in a group.
         (b) Finding buffer space in which to place information to be written.
      (2) Skipping
         (a) Omission of information "read" as input
         (b) Informing IOCS of the actual length of an output record already placed in the buffer via conventional programming.
      (3) Skipping or locating determined by condition of EOB switch in calling sequence.
         (a) EOB equals zero - signifies skipping
         (b) EOB is not equal to zero - signifies locating
   b. Conditions
      (1) No single command can locate words in more than one buffer.
      (2) If reading and the EOB switch is not equal to zero, words located are considered used when the file is next referenced by any IOCS routine. Therefore, all information located will be retained until then.
      (3) If reading and the EOB switch is equal to zero skipping has been indicated; therefore, it is not necessary to retain the information as above.

(4) In writing the same conditions hold true as for 1, 2, and 3 above except space is located rather than data.

Note: 2, 3, and 4 above would apply to non-n type commands only if an n-type command had been encountered earlier in the same command list, and acted on the same buffer. Otherwise, the condition of the EOB switch has no effect.

(5) Information or space located by a single IOCS command must be in sequential cells; therefore, the execution of a count command interrupted by the end of buffer condition is discontinued, and EOB itself is used as the exit of transfer address. An end of buffer condition is encountered when all the cells in the buffer are full and the command word count has not been satisfied.

c. N Type Commands

(1) Count control - IOCPN, IOCTN, IOCON Ref.

(2) Buffer Control - IORPN, IORTN Ref.

(3) Special Count Control - IOSPN, IOSTN Ref.

G. CLOSE

1. Function

a. Terminate activity on file

2. Calling sequence Ref.

a. General layout

b. Types of closing

(1) PZE - Rewind-unload, EOF written if output

(2) PTW - Rewind, EOF written if output

(3) MZE - No rewind, EOF written if output

(4) MON - No rewind, no EOF written

c. Closing more than 1 file

(1) Uses file list

IV. USING 9IOCS IN A PROGRAM

Objective: Setting up the theoretical inventory job discussed earlier will familiarize the student with the use of 9IOCS both in the coding and operational aspects of a job.

A. PHYSICAL ASPECTS

1. System Tape
   a. Contains IOCS routines
   b. Normally mounts on A1

2. Object Program
   a. As Card Input
      (1) Deck Layout — Ref.
         (a) Control Cards - Furnish job and file information
         (b) Loader
         (c) Binary Program
   b. Can also be tape input if desired. Layout on tape same as above.

B. OPERATIONAL ASPECTS

1. Preparing System Tape
   a. Items furnished by Applied Programming
      (1) Binary System Tape Writer Deck
      (2) Two file tape
         (a) File 1 - Symbolic version of IOCS, suitable for tape to card conversion.
         (b) File 2 - Assembly listing.
   b. Writing system tape
      (1) Ready Tape Writer
      (2) Tape drive number placed in address of keys
      (3) Load Cards

2. Running the Object Program
   a. Set up
      (1) System tape ready on A1
      (2) Input and Output Files mounted
      (3) Control cards and object program deck ready in card reader
      (4) Sense Switch 1 down signifies card input
      (5) Depress Load Tape

b. Visual Sequence of Operation
   (1) Portion of System Tape Loaded
   (2) Control Cards Read In
   (3) List of files to be mounted is printed
   (4) Program Stop - Operator Action Pause
   (5) Mount tapes if not mounted already
   (6) Press start
   (7) Remainder of System Tape loaded (IOCS routines)
   (8) System tape rewound
   (9) Object program loaded
   (10) Object program executed TSXing to IOCS routines in memory when required

C. PREPARING THE OBJECT PROGRAM DECK

1. A brief consideration of File Control Blocks is necessary at this time.
   a. Block of 12 cells per file
   b. Function is to keep track of file conditions.
   c. Blocks for each file must be grouped together in sequential locations. This area is referred to as the File Block.

2. Control Card Preparation
   a. Types of Control Cards
      (1) Job Card
         (a) Function - supply details of job
         (b) Must be first card of deck - 1 per deck
         (c) Layout - Consider fields of card
      (2) File Card
         (a) Function - supply details of a particular file for file control block
         (b) One per file. Cards for each file grouped together following "Job" card
         (c) Layout - Consider fields of card
      (3) Date Card
         (a) Function - supply job date
         (b) One per deck - follows "File" cards
         (c) Layout - Consider fields of card

      Note: As control card fields are considered unfamiliar terms will be encountered. Do not dwell in these areas at this time; they will be covered as the course progresses.

3. Object Deck Preparation
   a. The program must be now written and assembled.

D. WRITING THE PROGRAM

1. Review the "Inventory Job" flow chart to bring back to mind job objectives and flow. Ref.

2. Go through program listing dwelling on the function of each step. Emphasis need not be placed on CPU operations (number conversions, comparisons, etc.). Ref.

V. DETAILED INVESTIGATION OF THE 9IOCS PROGRAM

Objective: To offer to the student a detailed understanding of the programming used to allow the 9IOCS package perform the objectives covered.

The program listings and associated flow charts will be discussed both from a functional and coding viewpoint. Advanced theory of the internal workings of the program will be presented where necessary.

A. SYSTEM TAPE LAYOUT

1. Single file consisting of 6 records
   - a. First three read in on first select of system tape
     - (1) Record 1 - Self-loading IOCS loader
       - (a) Loads itself and rest of system tape as required
     - (2) Record 2 - I/O Executor
       - (a) Section of program which contains the routines and control areas (called Unit Control Blocks) that have direct communication with the I/O devices
     - (3) Record 3 - Preprocessor
       - (a) Routines which read in the Control Cards, interpret the information they supply and place data in the various control areas for future use.
   - b. Any number of last three read in on second and final select of system tape. They are selected as requested in Job card.
     - (1) Record 4 - Minimum IOCS
       - (a) Routines that initiate the execution of all IOCS subroutines discussed so far in course.
     - (2) Record 5 - Basic IOCS
       - (a) Will be read-in in addition to record 4 if an advanced operational IOCS is desired.
       - (b) Contains additional routines to be discussed later.
         Join, Copy, Stash, Rew, Wef, Bsr, Bsf, Ckpt

(3) Record 6 - Label IOCS

(a) Will be read-in in addition to records 4 & 5 if a full IOCS package is desired.

(b) Supplies additional routines to handle labeling operations of input/output files

Note: The larger the system used the more core area consumed.

B. SYSTEM TAPE WRITER

Ref. - Foil 50.06.24

1. Consider theory of operation of Writer. This is necessary for understanding loader operation.

a. When writing a system tape whenever a program break, other than that for a transfer card, is encountered in load addresses, a command (IOCP) is formed to load the previous sequence of instructions from the tape. This command is then inserted physically into the cell before that sequence.

b. When a transfer card is encountered an IORT is formed and placed in the cell located physically before the last sequence. A record is then written comprising all data read in since the initial start or last transfer card.

c. These commands will be used to load the program from tape.

d. A transfer card followed by a blank card will signal the end of the writing process. An end of file mark is written and tape rewound. A read check follows.

C. IOCS LOADER

1. Go through the loader listing step by step.

D. PREPROCESSOR

1. Consideration of the Unit Control Block is necessary at this time.

a. Block of 3 cells each, 1 per I/O device on system.

b. Contained in Executor section

c. Function is to keep track of operation of its associated I/O device. (Tape position, errors encountered, etc.).

d. Divorces the physical I/O device address from logical addresses used when writing the program. Reference to a logical unit will eventually cause program reference to a UCW which will then supply a physical I/O address.

2. Using appropriate flow-charts go through this portion of the program.

E. BUFFERING ROUTINES

1. Advanced IOCS theory of internal operation

a. Chaining

(1) Necessary to keep track of location of buffers in pool since each can be in different stages of use (filling, full, empty, etc.) by any of a number of files.

(2) Major chains Ref.

(a) Pool Chain--Circular type--used to keep track of buffers available for use in pool.

(b) Request Chain--First-last type--used to keep track of priority of buffers stacked to be filled by a particular I/O device in the case of input or written in the case of an output file.

(c) Sync Chain--First-last type--used to keep track of priority of full buffers stacked waiting to be emptied for an input file. Sync chain not used for output file.

(3) Other chains Ref.

(a) Regen Chain--First-last type--used to keep track of buffers being used by internal files.

(b) Hold Chain--used to keep track of buffers in hold status (referenced by "N" type command)

(c) Output File Chain--rather novel in that it keeps track of output files attached to a specific pool.

b. Advance of buffers through various chains Ref.
   (1) Input file
      (a) Location of empty buffer found in Pool Chain
      (b) Location of buffer placed in Request Chain to be filled
      (c) When full, location of buffer transferred to Sync Chain to wait until used
      (d) When its data is required, location of buffer is placed in Use status
      (e) When its data is used, location of buffer placed back in Pool Chain
   (2) Output file
      (a) Location of empty buffer found in Pool Chain
      (b) Location of buffer transferred to Use status where it remains till full
      (c) Location of buffer then transferred to Request Chain to be written out on output device
      (d) Location of buffer then transferred to Pool Chain to be used again

c. Various control blocks
   (1) Cover in detail the content of the three major control blocks used
      (a) pool control words
      (b) file control block
      (c) unit control block

2. Define routine
   a. Using appropriate flow-charts, listings and work sheets go through this subroutine. Ref.
   b. Cover the use of the communication region transfer vector.

3. Attach
   a. Using appropriate flow-charts, listings, and work sheets go through this subroutine.
      (1) Consider function of each subroutine on overall flow chart first.
      (2) Use listings and individual flow charts to cover each small routine in detail.

(3) Do not go into the details of the trapping sequence at this time.

Note: Adhere to conditions set forth on overall flow chart.

4. Read

a. Using appropriate flow-charts, listings, and work sheets go through this subroutine.

(1) Consider function of each subroutine on overall flow chart first.

(2) Use listings and individual flow charts to cover each small routine in detail.

(3) Using read trapping overall flow chart consider function of each subroutine in sequence.

(4) Use listings and individual flow charts to cover each small routine in the trapping sequence.

Note: Adhere to conditions set forth on overall flow charts.

5. Write

a. Using appropriate flow-charts, listings, and work sheets go through this subroutine.

(1) Consider function of each subroutine on overall flow chart first.

(2) Use listings and individual flow charts to cover each small routine in detail.

(3) Using write trapping overall flow chart consider function of each subroutine in sequence.

(4) Use listings and individual flow charts to cover each small routine in the trapping sequence.

Note: Adhere to conditions set forth on overall flow charts.

6. Close

a. Using appropriate flow-charts, listings, and work sheets go through this subroutine.

(1) Consider function of each subroutine on overall flow chart first.

(2) Use listings and individual flow charts to cover each small routine in detail.

Note: Adhere to conditions set forth on overall flow chart.

VI. BASIC IOCS ROUTINES

Objectives: To familiarize the student with the function of each of the additional routines of the Basic system.

A. JOIN

1. Function
   a. Connect two pools to form one of increased capacity

2. Restrictions
   a. Buffers of each pool must be of same size.
   b. Pool 2 may no longer be referred to in any other calling sequence.

3. Calling Sequence — Foil Ref. 50.06.17

B. COPY

1. Function
   a. Transfer data from an input to an output file without word transmission.

2. Restrictions
   a. Both files must be attached to the same pool.
   b. All words in the same buffer with and preceding the last word located are included in the output.
   c. All words in the same buffer subsequent to, the last word located are not included in the output; furthermore, they will behave as skipped when File 1 reading is resumed.
   d. If more than two buffers are involved, an intermediate buffer is included in the output only if at least one word within it has been located by a non-transmitting command.

3. Calling Sequence — Foil Ref. 50.06.22

C. STASH

1. Function
   a. Transfer information between an internal file and some other file.

2. Restrictions
   a. Both files must be attached to the same pool.
   b. 4 permissable types
      (1) Type 1 - File 1 is an internal output; File 2 is a regular output.
      (2) Type 2 - File 1 is a regular input; File 2 is an internal output.
      (3) Type 3 - File 1 is an internal input; File 2 is a regular output.
      (4) Type 4 - File 1 is an internal input; File 2 is an internal output.

3. Calling sequence — Ref.
   a. Consider NTS exit

D. WEF

1. Function
   a. Write an end of file mark

2. Calling sequence — Ref.

E. REW

1. Function
   a. Rewind the file

2. Calling Sequence — Ref.

F. BSR

1. Function
   a. Backspace 1 record

2. Calling Sequence — Ref.

G. BSF

1. Function
   a. Backspace specified number of files

2. Calling Sequence — Ref.

VII. ADDITIONAL FEATURES

Objectives: To acquaint the student with additional functions of 9IOCS.

A. LABELING

1. Provides a tape labeling system which can be employed if the use of labeled files is desired.

B. CHECKPOINT AND RESTART

1. Provide reference points to which a program may return for restart.

2. Provides a means to perform an operator initiated restart.

C. PRINTING MESSAGES ON-LINE

1. Provides a simplified method of printing messages on line through use of the MWR routine and its associated calling sequence.

ASSIGNMENT # 1

INTRODUCTION

Objective: To review the basic objectives of the course and reinforce the use of the flow chart symbols presented during the lecture.

A. Review the material presented by reading Part I of Lecture Summaries of the Study Guide.

B. Study Questions

1. List three techniques that can be used in the analysis of a program.

   a. _____

   b. _____

   c. _____

2. Name three advantages offered to the programmers by the use of 9IOCS.

   a. _____

   b. _____

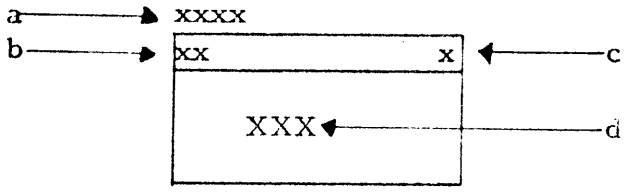
   c. _____

3. Why was the subroutine symbol adopted?

   _____

   _____

4. In the following illustration determine the purpose of each of the indicated areas.



5. Flow chart the following problem: $q = \frac{ax + b}{cx + d}$

Print the result using subroutine "PRINT" located on page 15 of the flow chart package. Entrance to the subroutine is at point D. Halt on return to the main program. Assume a, b, c, α and x are contained on separate input devices.

STUDY QUESTION ANSWERS

ASSIGNMENT #1

B.

1. a. Keep account of the changing conditions of all major registers and cells.

   b. Keeping notes on listings.

   c. Examining programmers' remarks.

   d. Use of flow charts.

2. a. Improved efficiency by minimizing the time utilized by C. P. U. waiting to receive or process I/O data.

   b. Simplified handling of I/O data.

   c. Automatic redundancy recovery routines.

   d. Built-in labeling, checkpoint, restart and message writer routines.

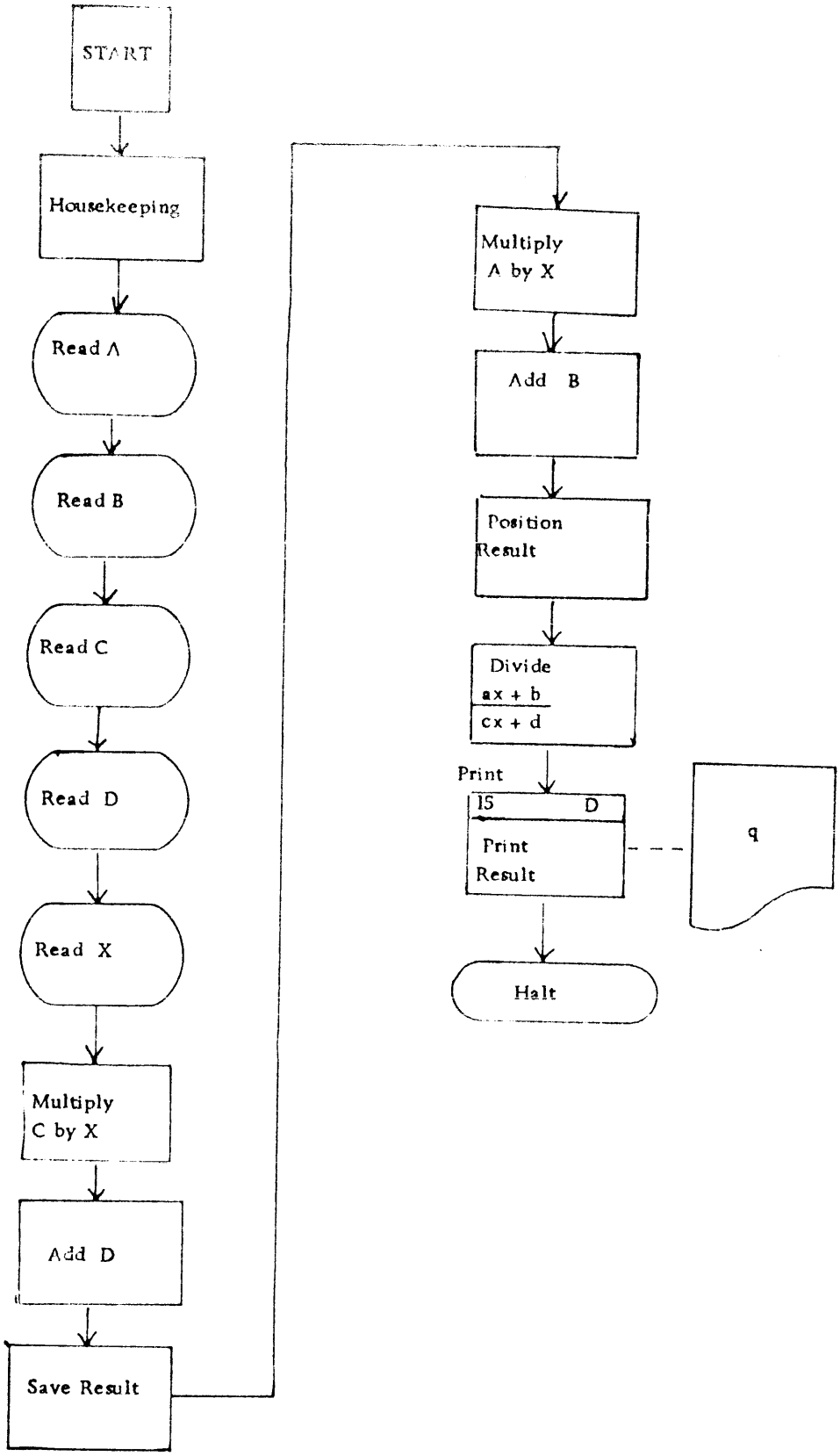
3. To provide a means of referring to a closed subroutine without the necessity of flow-charting it each time its use is encountered in a main program. Using this method it is only necessary to flow-chart the subroutine once.

4. a. Symbolic name of subroutine

   b. Page number

   c. Entry point

   d. Function


START
Housekeeping
Read A
Read B
Read C
Read D
Read X
Multiply C by X
Add D
Save Result
Multiply A by X
Add B
Position Result
Divide
ax + b
cx + d
Print
15 D
Print Result
q
Halt

ASSIGNMENT #2

BASIC CONCEPTS OF INPUT/OUTPUT CONTROL SYSTEMS

Objective: To review and reinforce the lecture on Basic Concepts of Input/Output Control Systems.

A. Review the material presented by reading Part II of Lecture Summaries of the Study Guide.

B. Study Questions

1. Name two programming improvements that can be used to improve I/O efficiency.

   a. _____

   b. _____

2. What is a buffer?

   _____

   _____

3. List two types of buffering.

   a. _____

   b. _____

4. What is a disadvantage of assigning a fixed number of buffers to each file of a job which involves varying file activity?

   _____

   _____

5. How does pool buffering overcome this?

   _____

   _____

STUDY QUESTION ANSWERS

ASSIGNMENT #2

B.

1. a. Blocking

   b. Buffering

2. Intermediate storage area where input or output data is held until operated on.

3. a. Swing

   b. Pool

4. At one time file A might require the use of one buffer while at others it might require a maximum of six. Assuming file B's demands are identical to that of file A but 180^{o} out of phase, a fixed buffer system would require the use of a minimum of 12 buffers to guarantee maximum efficiency in respect to time. Core storage area would be sacrificed.

5. Assuming the use of pool buffering rather than fixed in answer 4 above, a maximum of 7 buffers shared by both files need be alotted. When file A's activity is low it would use one buffer from the pool while file B would have six at its disposal; thereby, being able to satisfy its greatest demand. We have therefore achieved maximum timing efficiency with a minimum of storage allocation.

ASSIGNMENT #3

FUNCTIONS OF MINIMUM 9IOCS ROUTINES

Objective: To assign reading material which will review and reinforce the lecture on Functions of Minimum 9IOCS Routines.

A. Review the material presented by reading Part III of Lecture Summaries of the Study Guide.

B. Read and study pages

ASSIGNMENT #4

USING 9IOCS IN A PROGRAM

Objective: To assign reading material which will review and reinforce the lecture on Using 9IOCS In a Program.

A. Review the material presented by reading Part IV of Lecture Summaries of the Study Guide.

B. Read and study pages through of the 9IOCS Reference Manual.

C. Go through the sample program listing written in class and justify in your own mind the purpose of each step.

ASSIGNMENT #5

DETAILED INVESTIGATION OF THE 9IOCS PROGRAM

Objective: To offer suggestions toward reviewing and reinforcing the lecture on Detailed Investigation of the 9IOCS Program.

A. Review the material presented by reading Part V of Lecture Summaries of the Study Guide.

B. Using appropriate flow charts, listings, and work sheets review the areas of the program as they are presented in class.

C. Study Questions

1. a. How many records are present on the 9IOCS system tape?

_____

b. Name them.

_____ _____
_____ _____
_____ _____
_____ _____

2. What is the purpose of the Unit Control Blocks?

_____
_____

3. What is the purpose of the File Control Blocks?

_____
_____

4. List the sequence of progression through the various chains for an input file (Non-n type commands executed)

_____
_____
_____
_____
_____

5. List the sequence of progression through the various chains for an output file (non-n type commands executed)

_____
_____
_____
_____

STUDY QUESTION ANSWERS

ASSIGNMENT #5

C.

1. a. Six

   b. Loader, Executor, Preprocessor, Minimum, Basic, Label

2. Keep account of the conditions associated with a particular unit.

3. Keep account of the conditions associated with a particular file.

4. Pool, request, sync, use, pool

5. Pool, Use, Request, Pool.

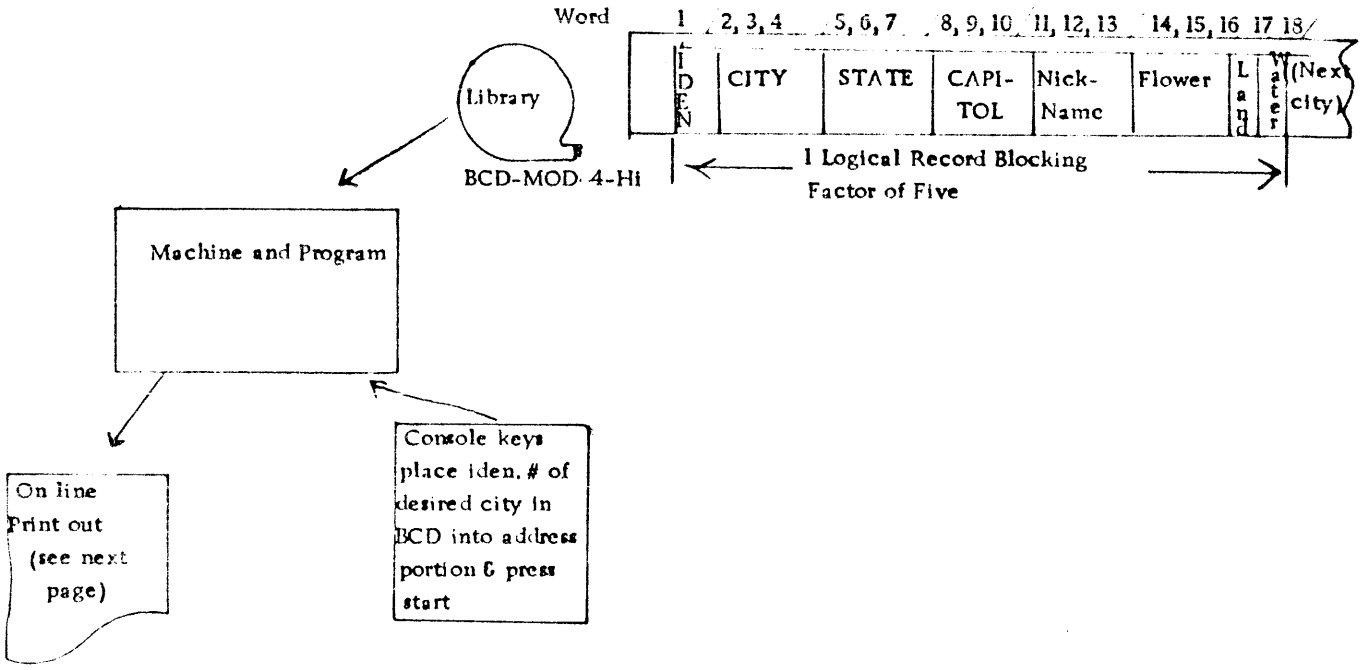
ASSIGNMENT #6

BASIC IOCS ROUTINES

Objective: To review the basic IOCS routines and offer a means of gaining experience in the writing of a program using 9IOCS.

A. Review the material presented by reading Part VI of Lecture Summaries of the Study Guide.

B. Write a program using 9IOCS routines, including Message Writer, to produce a demonstration printout. A sample printout and job specifications follow.



OCTOBER 21,1961
HI,I AM THE 7090 AT THE DEPARTMENT OF EDUCATION - POUGHKEEPSIE,N.Y.

YOU HAVE SELECTED BIRMINGHAM ,ALABAMA FOR YOUR INQUIRY.

SOME GENERAL STATISTICS
THE AREA OF THE STATE IS 51609 SQUARE MILES.
LAND AREA 51078 SQUARE MILES.
WATER AREA 531 SQUARE MILES.
CALLED COTTON STATE ,ITS CAPITOL IS MONTGOMERY
THE STATE FLOWER IS THE GOLDENROD

ASSIGNMENT #7

ADDITIONAL FEATURES

Objective: To offer suggestions toward reviewing and reinforcing the lecture on Additional Features.

A. Review the material presented by reading part VII of Lecture Summaries of the Study Guide.

B. Read the following sections of the 9IOCS Reference Manual.

LABORATORY PROJECT #1

RUNNING A PROGRAM USING 9IOCS

Objective: To furnish the student with an operator's view of the running of a program utilizing 9IOCS for Input/Output control.

A. Procedure

1. Run program written in class, observing flow and printouts, etc.

2. Misplace control cards and observe diagnostic printouts.

3. Display or dump pool control words, file control blocks and unit control blocks. Observe their formation at various points in the program.

LABORATORY PROJECT #2

DIAGNOSING 9IOCS PROGRAM FAILURES

Objective: To offer to the student the opportunity to gain experience using the 9IOCS program as a diagnostic on the 7090 system.

A. Procedure

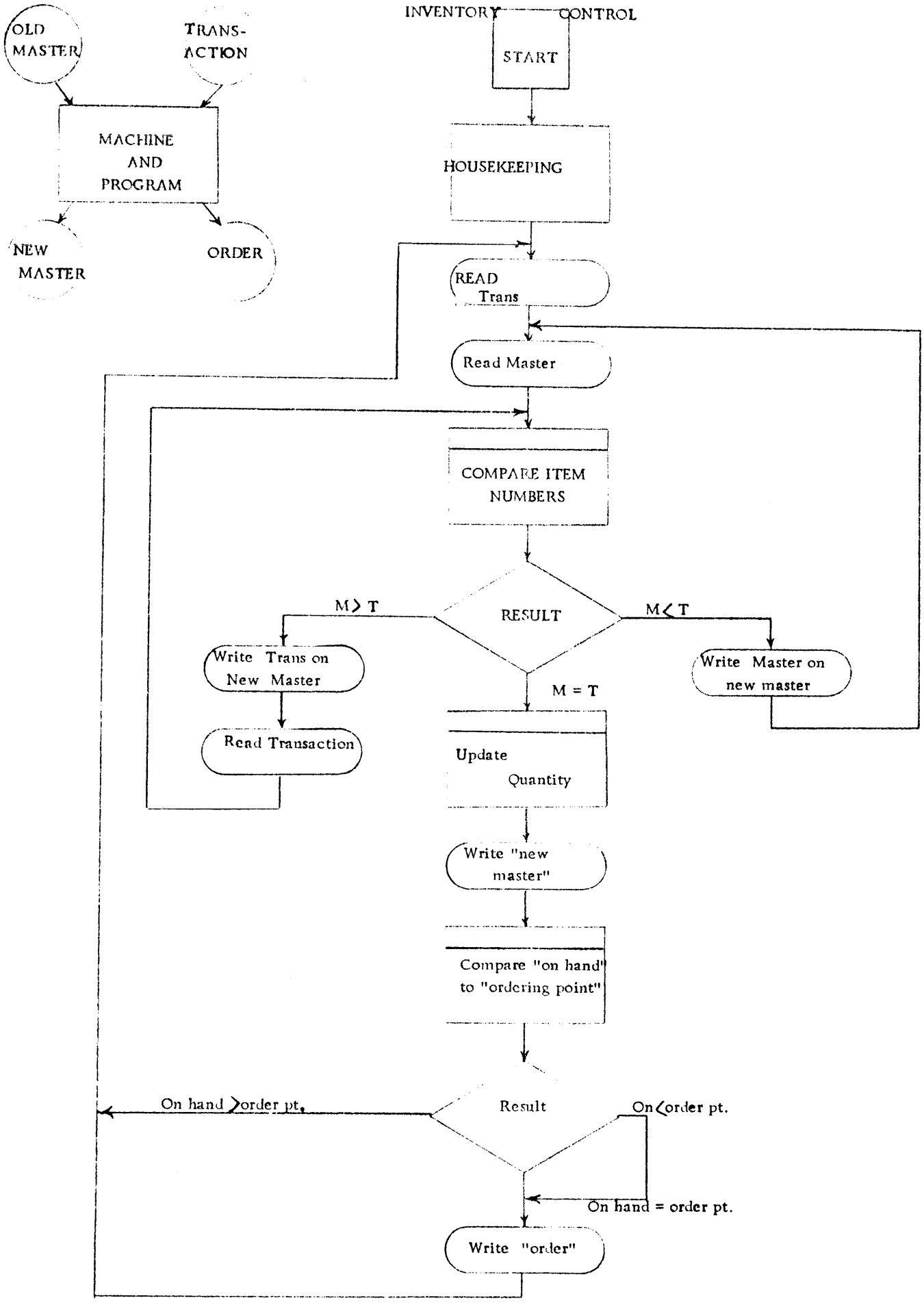
The lab instructor will place a bug patch card in the 9IOCS System Tape Writer deck. Write a system tape and then using it execute the sample job. With the aid of the flow charts and listings investigate the resultant failure and determine the location of the program step causing the problem.

I. SAMPLE PROGRAM

A. Attached is an example program using 9IOCS. For purposes of illustration a hypothetical inventory control job is considered. Two input tapes are involved in the operation. The first is an "Old Master", the second a "Transaction". The format of these and the two output tapes is given in the attached as Tape Layout. The outputs consist of a "New Master" and an "Order" file. The overall objective of the operation follows:

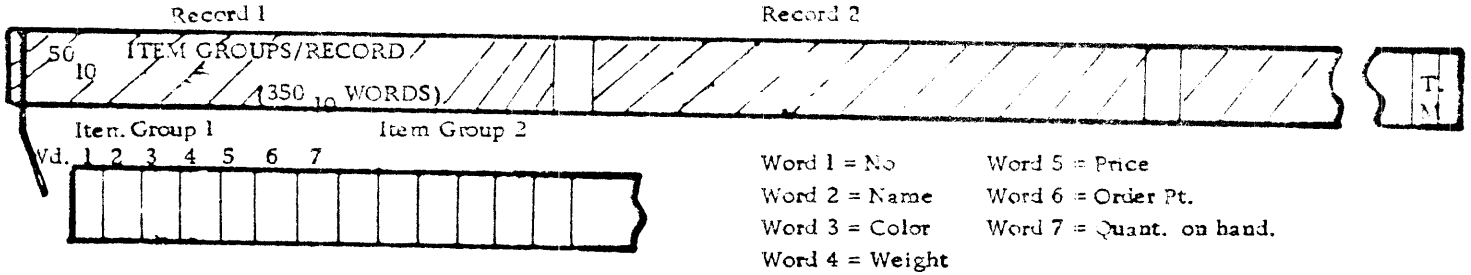
1. Compare Old Master and Transaction item numbers.

   a. If Old Master item number is greater than Transaction item number this is an insertion. Form New Master item group from Transaction.
   b. If Old Master item number is smaller than Transaction item number, no change is to be made to this Old Master item group. Transfer Old Master item group to New Master.
   c. If Old Master item number equals Transaction item number, update quantity on hand. This operation consists of the addition of Transaction items received to Old Master quantity on hand. The quantity in the Transaction items sold field is then subtracted from the sum to form the new quantity on hand. Transfer the Old Master item group with new quantity on hand inserted to the New Master. Handling of the Order file remains. The calculated quantity on hand is compared to the Old Master ordering point. If it is equal to or less than that specified in ordering point an Order item group is formed and transferred to the Order tape. A logic diagram and assembly listing of the program is attached.


OLD MASTER
TRANS-ACTION
MACHINE AND PROGRAM
NEW MASTER
ORDER
INVENTORY CONTROL
START
HOUSEKEEPING
READ Trans
Read Master
COMPARE ITEM NUMBERS
RESULT
M > T
M < T
M = T
Write Trans on New Master
Read Transaction
Write Master on new master
Update Quantity
Write "new master"
Compare "on hand" to "ordering point"
Result
On hand > order pt.
On < order pt.
On hand = order pt.
Write "order"

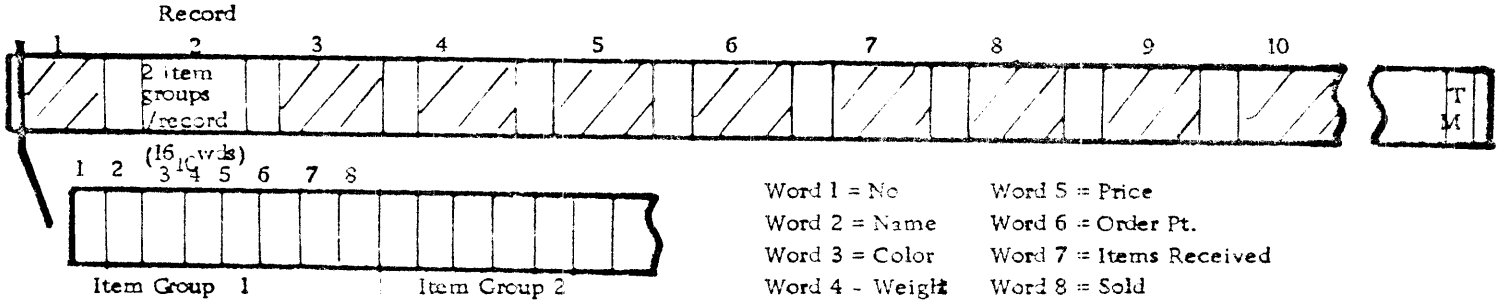
TAPE LAYOUT

OLD MASTER & NEW MASTER



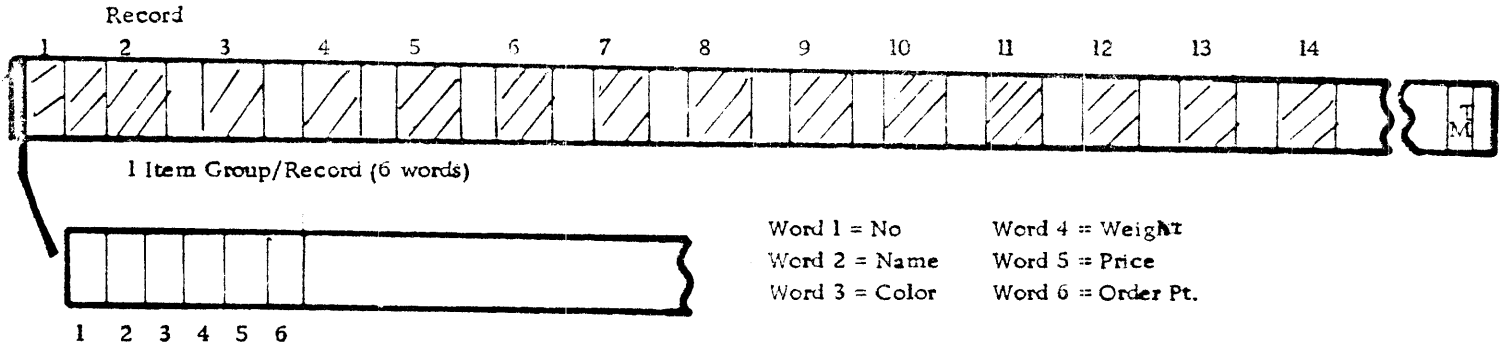
Word 1 = No
Word 2 = Name
Word 3 = Color
Word 4 = Weight
Word 5 = Price
Word 6 = Order Pt.
Word 7 = Quant. on hand.

TRANSACTION



Word 1 = No
Word 2 = Name
Word 3 = Color
Word 4 = Weight
Word 5 = Price
Word 6 = Order Pt.
Word 7 = Items Received
Word 8 = Sold

ORDER



Word 1 = No
Word 2 = Name
Word 3 = Color
Word 4 = Weight
Word 5 = Price
Word 6 = Order Pt.

NOTE: All tapes BCD

```
                    10000           ORG     4096        BEYOND IOCS
                    00044   IOCS    EQU     36
                    00050   DEFINE  EQU     IOCS+4
                    00054   ATTACH  EQU     IOCS+8
                    00056   CLOSE   EQU     IOCS+10
                    00060   OPEN    EQU     IOCS+12
                    00062   READ    EQU     IOCS+14
                    00064   WRITE   EQU     IOCS+16
                    10000   LAPOOL  BSS     5000
                    21610   SMPOOL  BSS     5000
                    33420   TNYPOO  BSS     5000
                    45230   OLDMST  BSS     12          OLD MASTER FILE CONTROL BLOCK
                    45244   NEWMST  BSS     12          NEW MASTER FILE CONTROL BLOCK
                    45260   TRNACT  BSS     12          TRANSACTION FILE CONTROL BLOCK
                    45274   ORDER   BSS     12          ORDER FILE CONTROL BLOCK
45310  0 00000 0 45230  LIST1   PZE     OLDMST      OLD MASTER FILE IS IMMEDIATE
45311  0 00000 0 45244          PZE     NEWMST      NEW MASTER FILE IS IMMEDIATE
45312  0 00000 0 45260  LIST2   PZE     TRNACT      TRANSACTION FILE IS IMMEDIATE
45313  0 00000 0 45274  LIST3   PZE     ORDER       ORDER FILE IS IMMEDIATE
45314  0074 00 4 00050  START   TSX     DEFINE,4    FORM BUFFER POOL
45315  0 00000 0 10000          PZE     LAPOOL      POOL FOR MASTER FILES IS A LARGE POOL
45316  0 00536 0 00012          PZE     10,,350           CONSISTING OF TEN 350 WORD BUFFE
45317  0074 00 4 00050          TSX     DEFINE,4    FORM BUFFER POOL
45320  0 00000 0 21610          PZE     SMPOOL      POOL FOR TRANSACTION IS SMALL
45321  0 00020 0 00012          PZE     10,,16            CONSISTING OF TEN 16 WORD BUFFER
45322  0074 00 4 00050          TSX     DEFINE,4    DEFINE TINY POOL
45323  0 00000 0 33420          PZE     TNYPOO            AS CONSISTING OF
45324  0 00006 0 00012          PZE     10,,6             TEN SIX WORD BUFFERS
45325  0074 00 4 00054          TSX     ATTACH,4    ATTACH MASTER FILES TO LARGE POOL AND
45326  0 00000 0 10000          PZE     LAPOOL            OPEN THEM SINCE THEY ARE IMMEDIA
45327  0 00002 0 45310          PZE     LIST1,,2          2 FILES OF LIST 1
45330  0074 00 4 00054          TSX     ATTACH,4    ATTACH TRANSACTION FILE TO SMALL POOL
45331  0 00000 0 21610          PZE     SMPOOL            POOL AND OPEN IT
45332  0 00001 0 45312          PZE     LIST2,,1    1 FILE  OF LIST 2
45333  0074 04 0 00054          TSX     ATTACH,,4   ATTACH ORDER FILE TO TINY POOL
45334  0 00000 0 33420          PZE     TNYPOO
45335  0 00001 0 45313          PZE     LIST3,,1
45336  0600 00 0 45521          STZ     SW4         INITIALIZE SWITCHES
45337  0600 00 0 45520          STZ     SW3
45340  0600 00 0 45517          STZ     SW2
45341  0600 00 0 45516          STZ     SW1
```

50.03.48

```
45342  0074 00 4 00062  RDTRNS TSX      READ,4
45343  0 45350 0 45260           PZE      TRNACT,,TIS+1
45344  0 45515 0 45474           PZE      TREOF,,ERR
45345 -0000 06 2 00000  TIN      IOCPN    **,,6
45346 -0000 01 2 00000  TIR      IOCPA    **,,1
45347 -1 00001 2 00000  TIS      IOCTN    **,,1
45350  0520 00 0 45516           ZET      SW1
45351  0020 00 0 45364           TRA      COITNU
45352  0520 00 0 45517           ZET      SW2
45353  0020 00 0 45466           TRA      DUP
45354  0074 00 4 00062  RDMSTR TSX      READ,4
45355  0 45362 0 45230           PZE      OLDMST,,OMQOH+1
45356  0 45515 0 45462           PZE      MSTEOF,,ERR
45357 -0000 05 2 00000  OMIN   IOCPN    **,,5
45360 -0000 01 2 00000  OMQP   IOCPA    **,,1
```

READ

E O F AND ERROR ROUTINE LOCATION
LOCATE ITEM NUMBER OF TRANSACTION
LOCATE ITEMS RECEIVED OF TRANSACTION
LOCATE ITEMS SOLD OF TRANSACTION RECO
NEW TRANSACTION REQUIRED
NO
YES DID OLD MASTER EOF OCCUR
YES
NO READ

LOCATE ITEM NUMBE OLD MASTER
Locate ordering point old master


```
45361 -1 00001 2 00000  OMQOH  IOCTN    **,,1
45362  0520 00 0 45521           ZET      SW4
45363  0020 00 0 45502           TRA      DUP2
45364  0774 00 2 00000  COITNU AXT      0,2
45365  0560 60 0 45357           LDQ*     OMIN
45366  0774 00 2 00044           AXT      36,2
45367  0074 00 1 45525           TSX      BCDBIN,1
45370  0602 00 0 45522           SLW      TEMP1
45371  0774 00 2 00000           AXT      0,2
45372  0560 60 0 45345           LDQ*     TIN
45373  0774 00 2 00044           AXT      36,2
45374  0074 00 1 45525           TSX      BCDBIN,1
45375  0774 00 2 00000           AXT      0,2
45376 -0340 00 0 45522           LAS      TEMP1
45377  0020 00 0 45406           TRA      MSTRSW
45400  0020 00 0 45412           TRA      EQUAL
45401  0074 00 4 00064           TSX      WRITE,4
45402  0 00000 0 45244           PZE      NEWMST,,0
45403 -1 00007 4 45345           IOCT     TIN,4,7
45404  0634 00 4 45516           SXA      SW1,4
45405  0020 00 0 45342           TRA      RDTRNS
```

LOCATE QUANTITY ON HAND OLD MAST
DID TRANSACTION EOF OCCUR
YES
NO
EXTRACT OLD MASTER ITEM NUMBER
INITIALIZE FOR CONVERSION
CONVERT
CONVERTED NUMBER IN ACCUMULATOR ON EX

EXTRACT TRANSACTION ITEM NUMBER

CONVERT

OLD MASTER SMALLER

OLD MASTER GREATER
WRITE TRANSACTION
ON NEW MASTER
TURN ON SWITCH ONE
TRY AGAIN

```
45406   0074 00 4 00064  MSTRSM TSX     WRITE,4        MASTER IS SMALLER THAN TRANSACTION
45407   0 00000 0 45244          PZE     NEWMST,,0      WRITE THIS RECORD ON NEW MASTER
45410  -1 00007 4 45357          IOCT    OMIN,4,7
45411   0020 00 0 45354          TRA     RDMSTR         TRY AGAIN
45412   0774 00 2 00000  EQUAL    AXT     0,2
45413   0560 60 0 45361          LDQ*    OMQOH          EXTRACT OLD MASTER QUANTITY ON HAND
45414   0774 00 2 00044          AXT     36,2           INITIALIZE FOR CONVERSION
45415   0074 00 1 45525          TSX     BCDBIN,1       CONVERT IT
45416   0602 00 0 45523          SLW     TEMP2
45417   0774 00 2 00000          AXT     0,2
45420   0560 60 0 45346          LDQ*    TIR            EXTRACT TRANSACTION ITEMS RECEIVED
45421   0774 00 2 00044          AXT     36,2
45422   0074 00 1 45525          TSX     BCDBIN,1       CONVERT IT
45423   0401 00 0 45523          ADM     TEMP2          ADD ITEMS RECEIVED TO QUANTITY ON HA
45424   0601 00 0 45523          STO     TEMP2
45425   0774 00 2 00000          AXT     0,2
45426   0560 60 0 45347          LDQ*    TIS            EXTRACT TRANSACTION ITEMS SOLD
45427   0774 00 2 00044          AXT     36,2
45430   0074 00 1 45525          TSX     BCDBIN,1       CONVERT IT
45431   0601 00 0 45524          STO     TEMP3
45432   0500 00 0 45523          CLA     TEMP2          SUBTRACT ITEMS SOLD FROMSUM FORMING
45433  -0400 00 0 45524          SBM     TEMP3               NEW QUANTITY ON HAND
45434   0601 00 0 45524          STO     TEMP3
45435   0774 00 2 00005          AXT     5,2
45436   0074 00 1 45544          TSX     BINBCD,1
45437   0774 00 2 00000          AXT     0,2
45440  -0600 60 0 45361          STQ*    OMQOH
45441   0074 00 4 00064          TSX     WRITE,4        WRITE UPDATED RECORD ON
45442   0 00000 0 45244          PZE     NEWMST,,0           NEW MASTER
45443  -1 00007 4 45357          IOCT    OMIN,4,7
45444   0774 00 2 00000          AXT     0,2
45445   0560 60 0 45360          LDQ*    OMOP           EXTRACT OLD MASTER ORDERING POINT
45446   0774 00 2 00044          AXT     36,2
45447   0074 00 1 45525          TSX     BCDBIN,1       CONVERT IT
45450  -0340 00 0 45524          LAS     TEMP3
```

```
45451  0020 00 0 45455          TRA     ORDR
45452  0020 00 0 45455          TRA     ORDR
45453  0600 00 0 45516          STZ     SW1
45454  0020 00 0 45342          TRA     RDTRNS
45455  0074 00 4 00064  ORDR    TSX     WRITE,4
45456  0 00000 0 45274          PZE     ORDERP,,0
45457 -1 00006 4 45345          IOCT    TI..,4,6
45460  0600 00 0 45516          STZ     SW1
45461  0020 00 0 45342          TRA     RDTRNS
45462  0074 00 4 00056  MSTEOF  TSX     CLOSE,4
45463  2 00000 0 45230          PTW     OLDMST
45464  0520 00 0 45521          ZET     SW4
45465  0020 00 0 45506          TRA     CLSTR
45466  0074 00 4 00064  DUP     TSX     WRITE,4
45467  0 00000 0 45244          PZE     NEWMST,,0
45470 -1 00007 4 45345          IOCT    TI..,4,7
45471  0634 00 4 45517          SXA     SW2,4
45472  0634 00 4 45520          SXA     SW3,4
45473  0020 00 0 45342          TRA     RDTRNS
45474  0520 00 0 45520  TREOF   ZET     SW3
45475  0020 00 0 45506          TRA     CLSTR
45476  0520 00 0 45516          ZET     SW1
45477  0020 00 0 45502          TRA     DUP2
45500  0634 00 4 45521          SXA     SW4,4
45501  0020 00 0 45354          TRA     RDMSTR
45502  0074 00 4 00064  DUP2    TSX     WRITE,4
45503  0 00000 0 45244          PZE     NEWMST,,0
45504 -1 00007 4 45357          IOCT    OR..,4,7
45505  0020 00 0 45500          TRA     DUP2-2
45506 -0074 00 4 00056  CLSTR   TSX     CLOSE,4
45507  2 00000 0 45260          PTW     TRNACT
45510  0074 00 4 00056          TSX     CLOSE,4
45511  2 00000 0 45244          PTW     NEWMST
45512  0074 00 4 00056          TSX     CLOSE,4
45513  2 00000 0 45274          PTW     ORDER

45514  0020 00 0 00044  STOP    TRA     IOCS
```

```
        OLD MASTER ORDERING POINT GREATER TH
        EQUAL
        OLD MASTER ORDERING POINT LESS THAN
              QUANTITY ON HAND        REPEAT
        WRITE ORDER



        OLD MASTER E O F
              REWIND  IT
              DID TRANSACTION EOF OCCUR
              YES
NO      COMPLETE BY DUPING TRANSACTIONS
              ON NEW MASTER UNTIL ALL DONE



        TRANSACTION EOF  — DUPING
              YES
NO      HAS LAST OLD MASTER BEEN WRITTEN
        NO
        YES,READ OLD MASTER

        COMPLETE BY DUPING OLD MASTER
        ON NEW MASTER UNTIL DONE


        CLOSE TRANSACTION
              REWIND  IT
        CLOSE NEW MASTER
              REWIND IT
        CLOSE ORDER
              REWIND IT
        Job Finished
```

```
45515  0000 00 0 00000  ERR     HTR                          ERROR  STOP
45516 +0000000000000    SW1     OCT     0
45517 +0000000000000    SW2     OCT     0
45520 +0000000000000    SW3     OCT     0
45521 +0000000000000    SW4     OCT     0
45522 +0000000000000    TEMP1   OCT     0
45523 +0000000000000    TEMP2   OCT     0
45524 +0000000000000    TEMP3   OCT     0
45525  0600 00 0 45541  BCDBIN  STZ     BNOO
45526 -0754 00 0 00000          PXD     0,0
45527 -0763 00 0 00006          LGL     6
45530  0361 00 0 45541          ACL     BNOO
45531 -2 00006 2 45540          TIX     EXIT,2,6
45532 -0600 00 0 45542          STQ     BNOO+1
45533 -0130 00 0 00000          XCL
45534  0200 00 0 45543          MPY     TEN
45535 -0600 00 0 45541          STQ     BNOO
45536  0560 00 0 45542          LDQ     BNOO+1
45537  0020 00 0 45527          TRA     BCDBIN+2
45540  0020 00 1 00001  EXIT    TRA     1,1
```



```
                45541  BNOO    BSS     2
45543 +0000000000012    TEN     DEC     10
45544  0560 00 0 45524  BINBCD  CLA     TEMP3
45545  0760 00 0 00000          CLM
45546  0560 00 0 45524          LDQ     TEMP3
45547  0600 00 0 45563          STZ     TEMP4
45550  0220 00 0 45543          DVH     TEN
45551 -0600 00 0 45541          STQ     BNOO
45552  0560 00 0 45563          LDQ     TEMP4
45553 -0763 00 0 00006          LGR     6
45554 -0600 00 0 45563          STQ     TEMP4
45555  0560 00 0 45541          LDQ     BNOO
45556  2 00001 2 45550          TIX     BINBCD+1,2,1
```

50.03.52

```
45557  0560 00 0 45563          LDQ     TEMP4
45560 -0765 00 0 00006          LGR     6
45561 -0600 00 0 45563          STO     TEMP4
45562  0020 00 1 00001          TRA     1,1
45563 +0000000000000    TEMP4   OCT     0
                 45314          END     START
```

II. CARD LAYOUT TO ILLUSTRATE SYSTEM TAPE WRITER OPERATION

BLA	NK
TRA BVOO	IOCP 5,,1
TCH 4	TCOX 2
NOP	TCOX 2
1,,6	CK sum

CARD - IOCB0110

BLA	NK
TCH 4	
6,,1	Ck. Sum

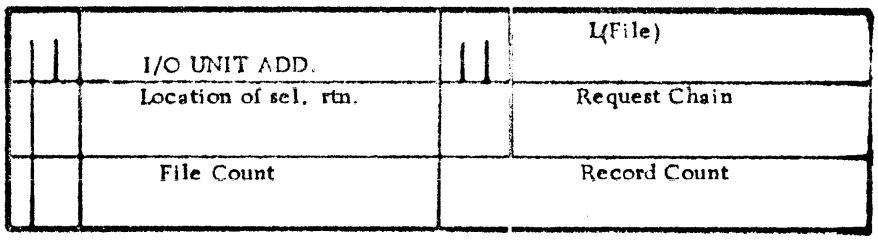
CARD - IOCB0120

INSTRUCTIONS	
AXT 4,2	SXA BRECT,4
77634,,26	Ck. Sum

CARD - IOCB0130

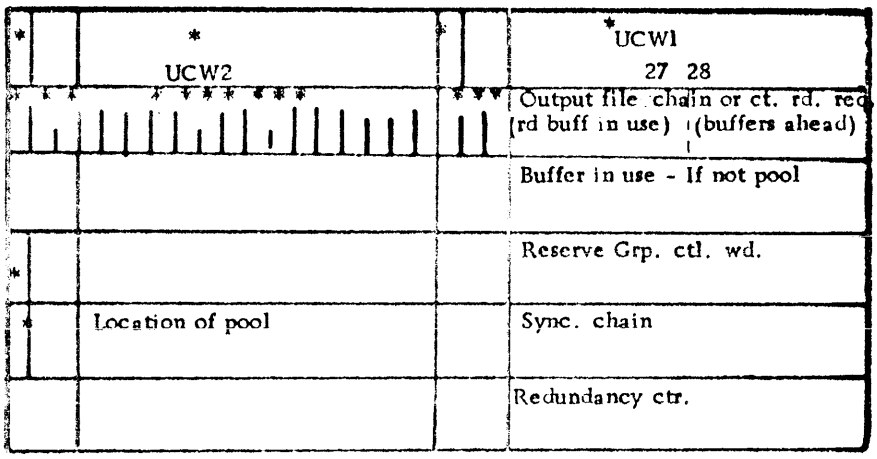
(lst 3 cards of loader)

UCW

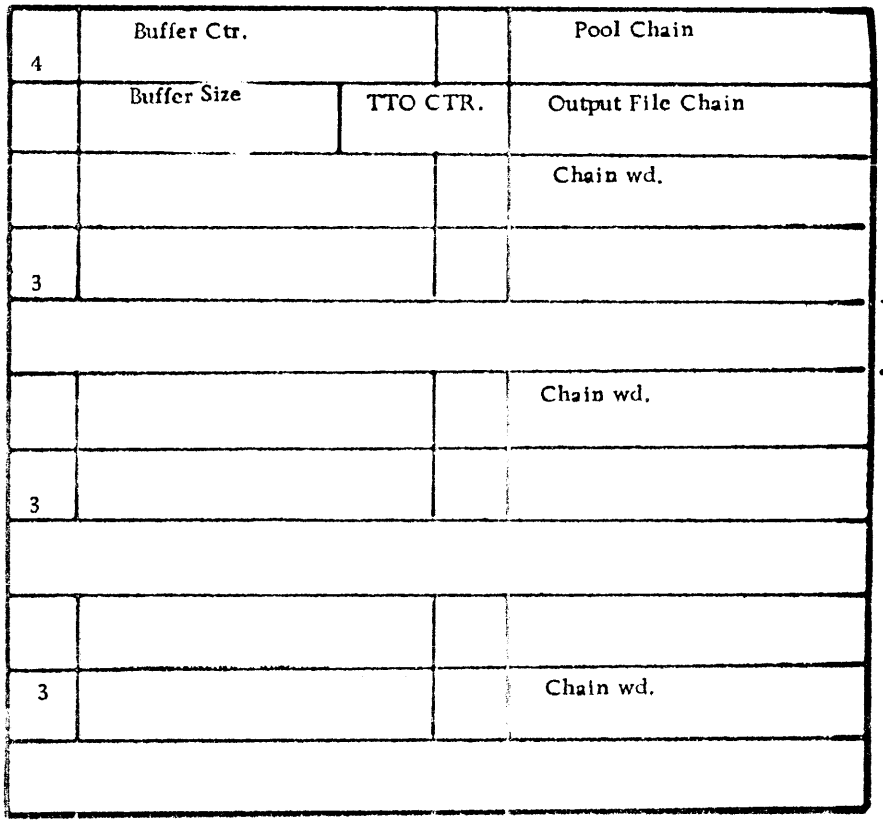


Regen. chain for Int. File

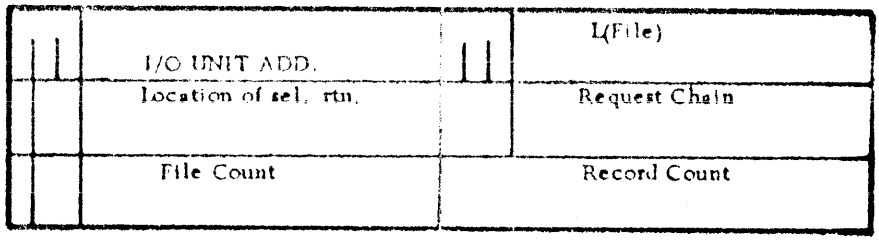
FILE



POOL

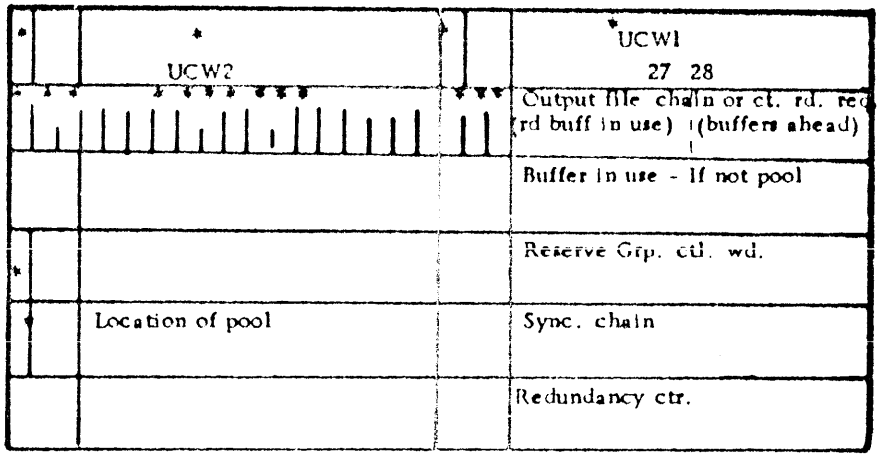


UCW

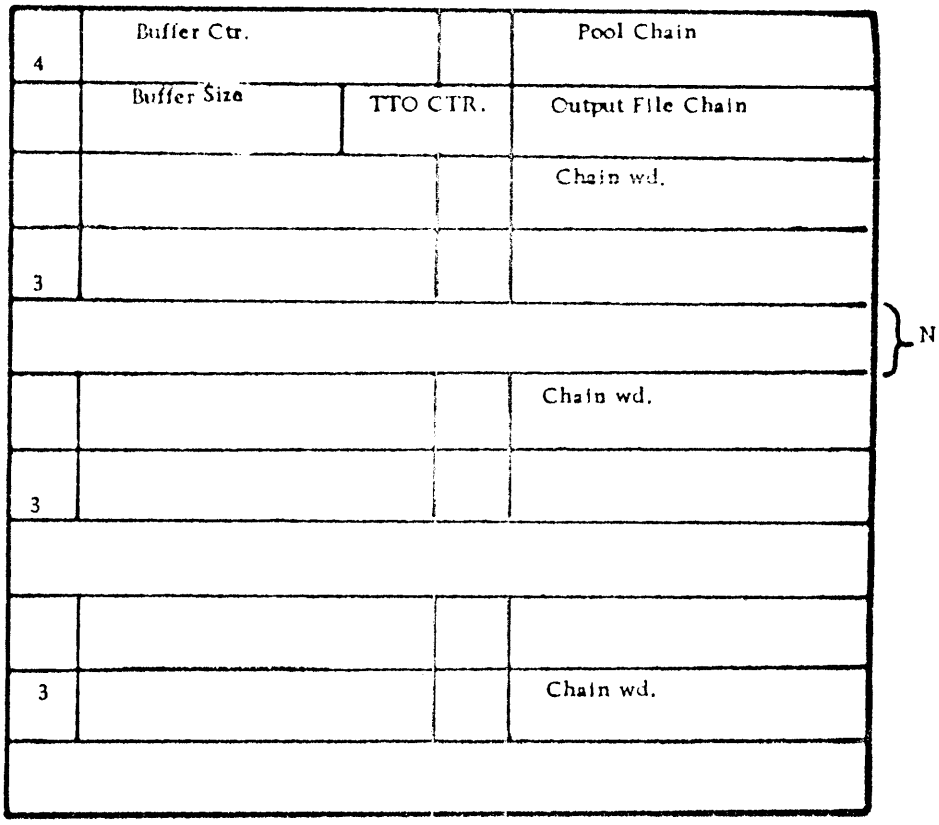


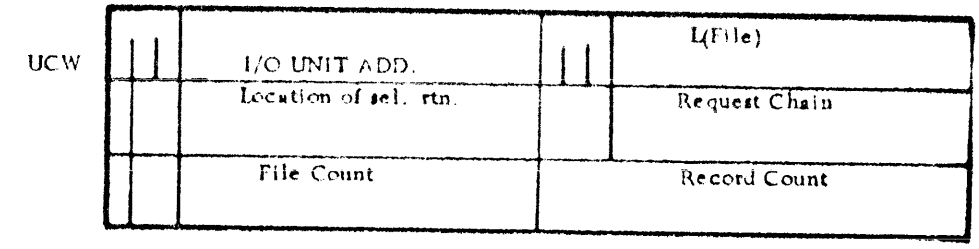
FILE

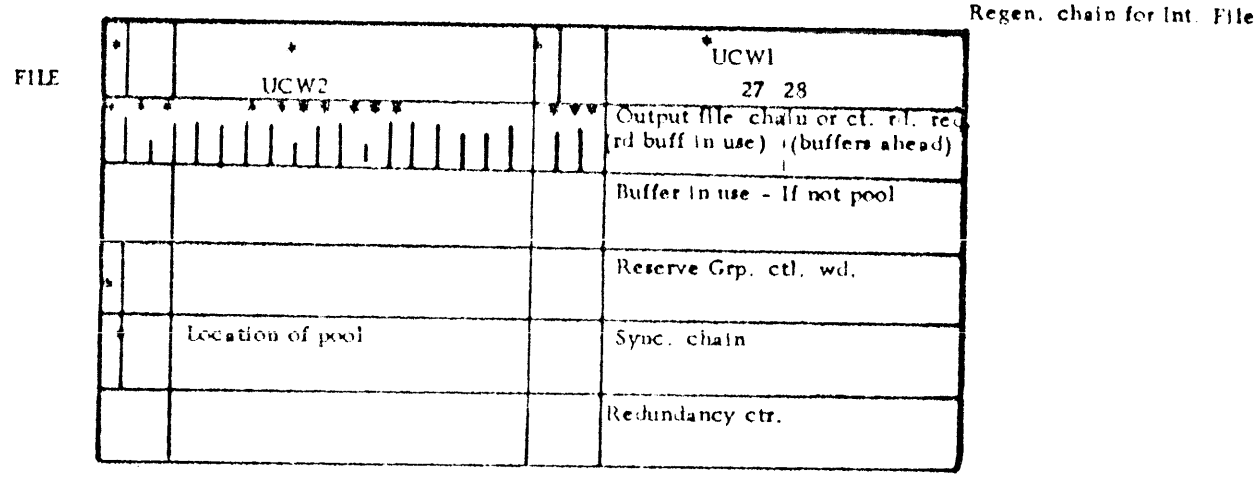
Regen. chain for Int. File

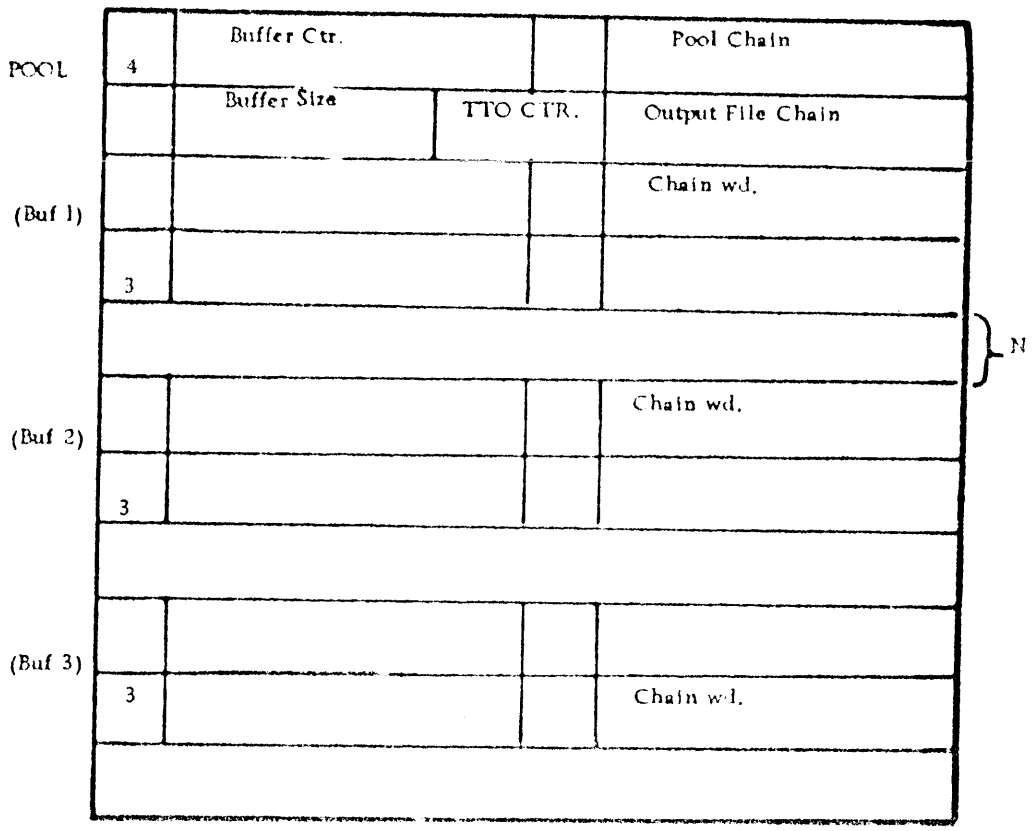


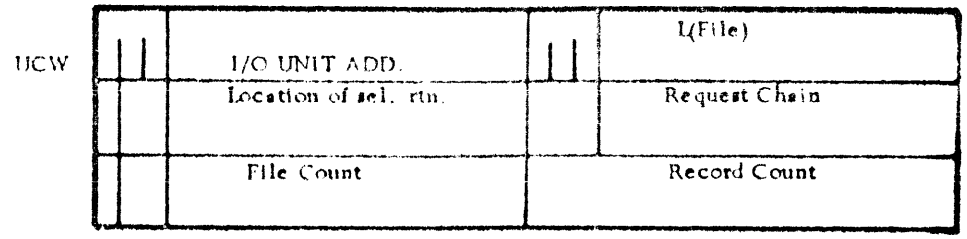
POOL

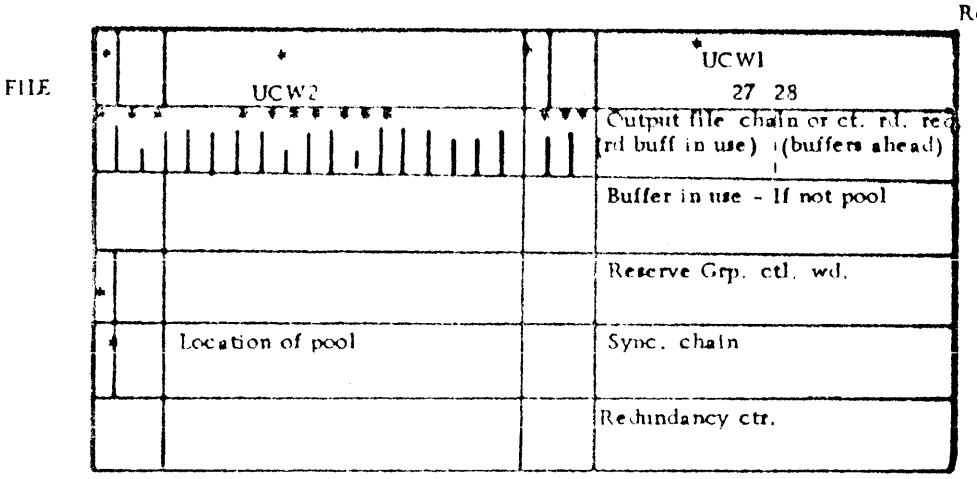


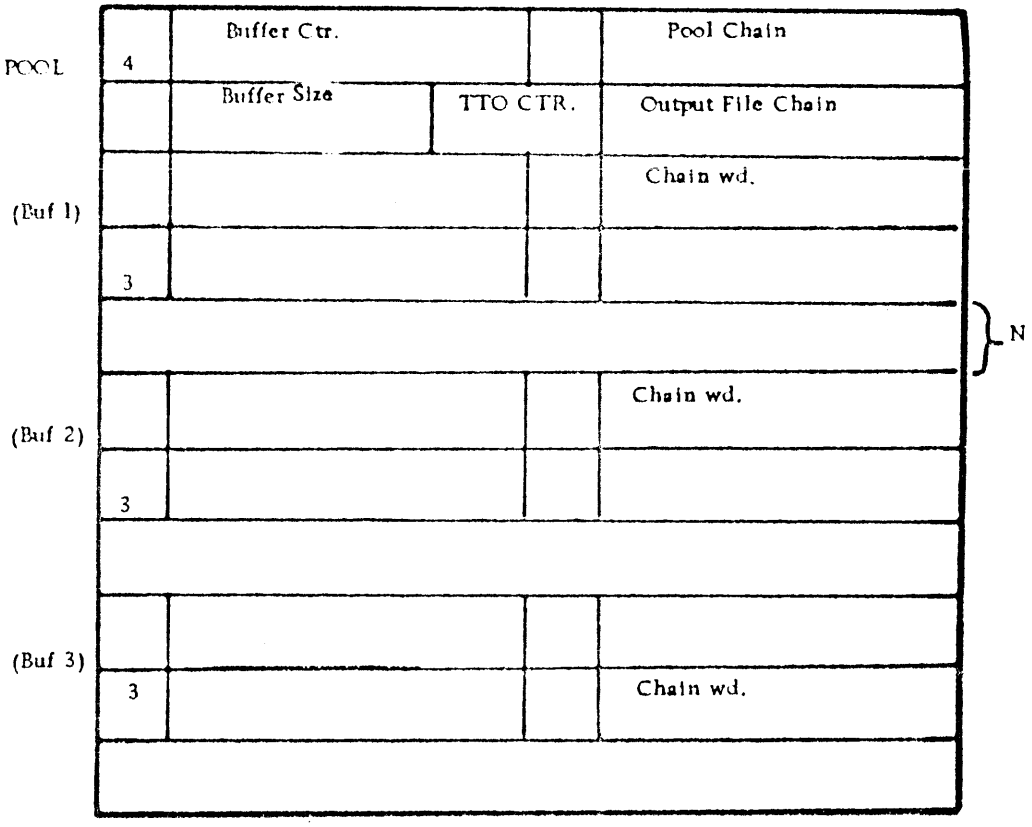

UCW
I/O UNIT ADD.
L(File)
Location of sel. rtn.
Request Chain
File Count
Record Count

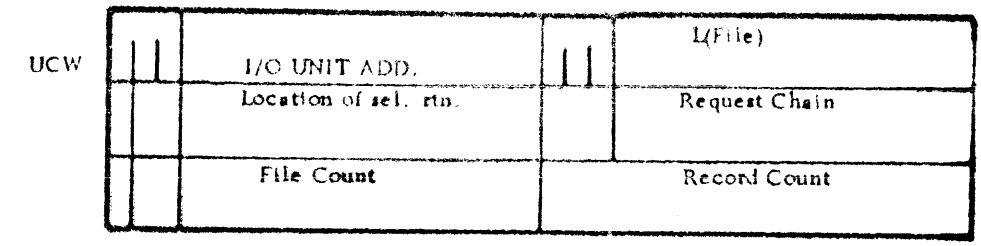

Regen. chain for Int. File
FILE
UCW2
UCW1
27 28
Output file chain or ct. rd. req
rd buff in use)
(buffers ahead)
Buffer in use - If not pool
Reserve Grp. ctl. wd.
Location of pool
Sync. chain
Redundancy ctr.

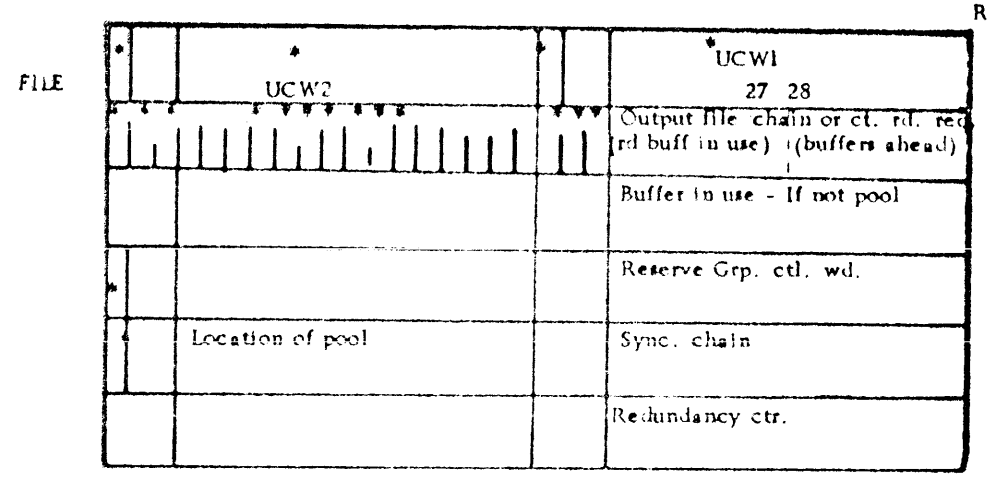

POOL
4
Buffer Ctr.
Pool Chain
Buffer Size
TTO CTR.
Output File Chain
(Buf 1)
Chain wd.
3
N
(Buf 2)
Chain wd.
3
(Buf 3)
3
Chain wd.

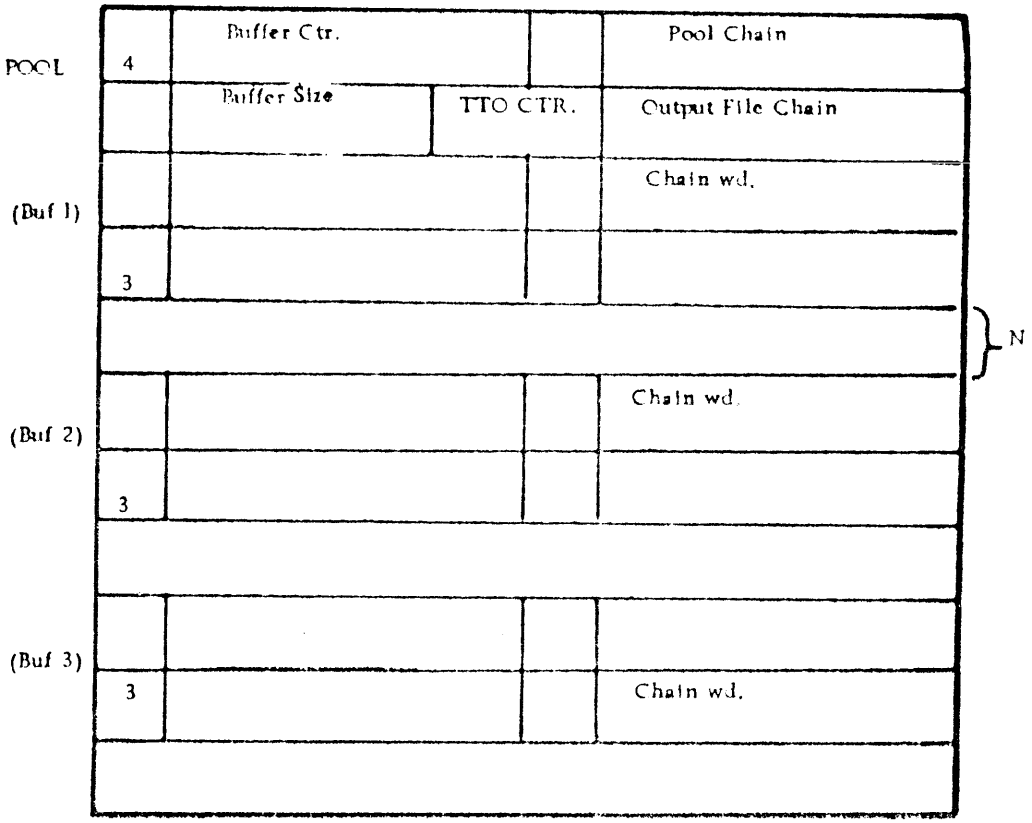

UCW
I/O UNIT ADD.
I(File)
Location of sel. rtn.
Request Chain
File Count
Record Count

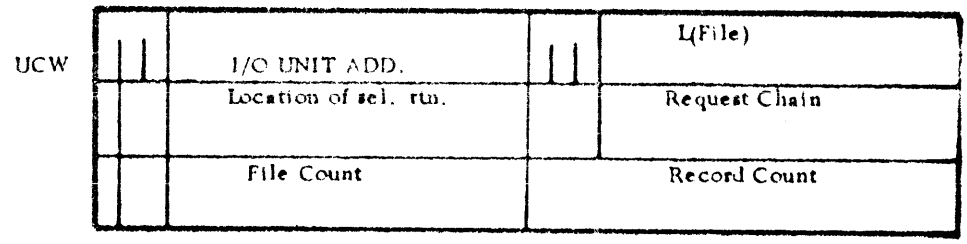

Regen. chain for Int. File
FILE
UCW2
UCW1
27 28
Output file chain or ct. rd. req
(rd buff in use)
(buffers ahead)
Buffer in use - If not pool
Reserve Grp. ctl. wd.
Location of pool
Sync. chain
Redundancy ctr.

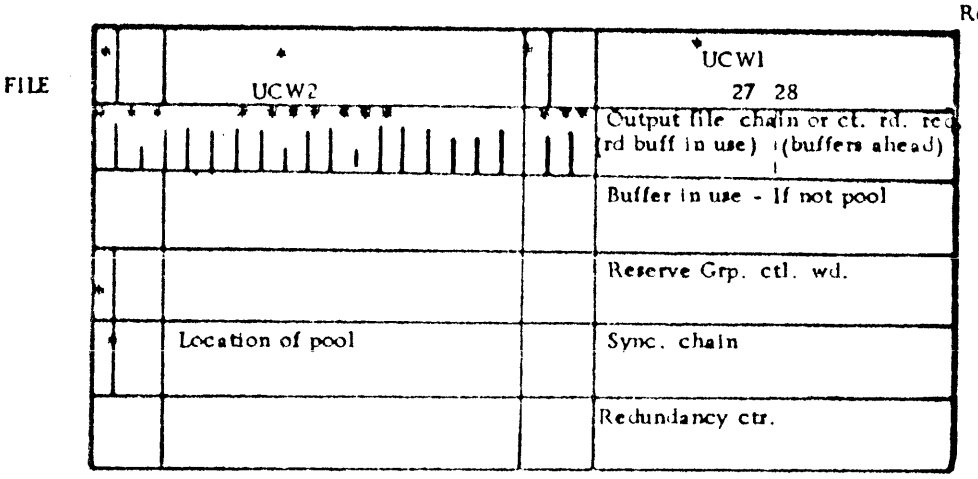

POOL
4
Buffer Ctr.
Pool Chain
Buffer Size
TTO CTR.
Output File Chain
(Buf 1)
Chain wd.
3
N
(Buf 2)
Chain wd.
3
(Buf 3)
3
Chain wd.

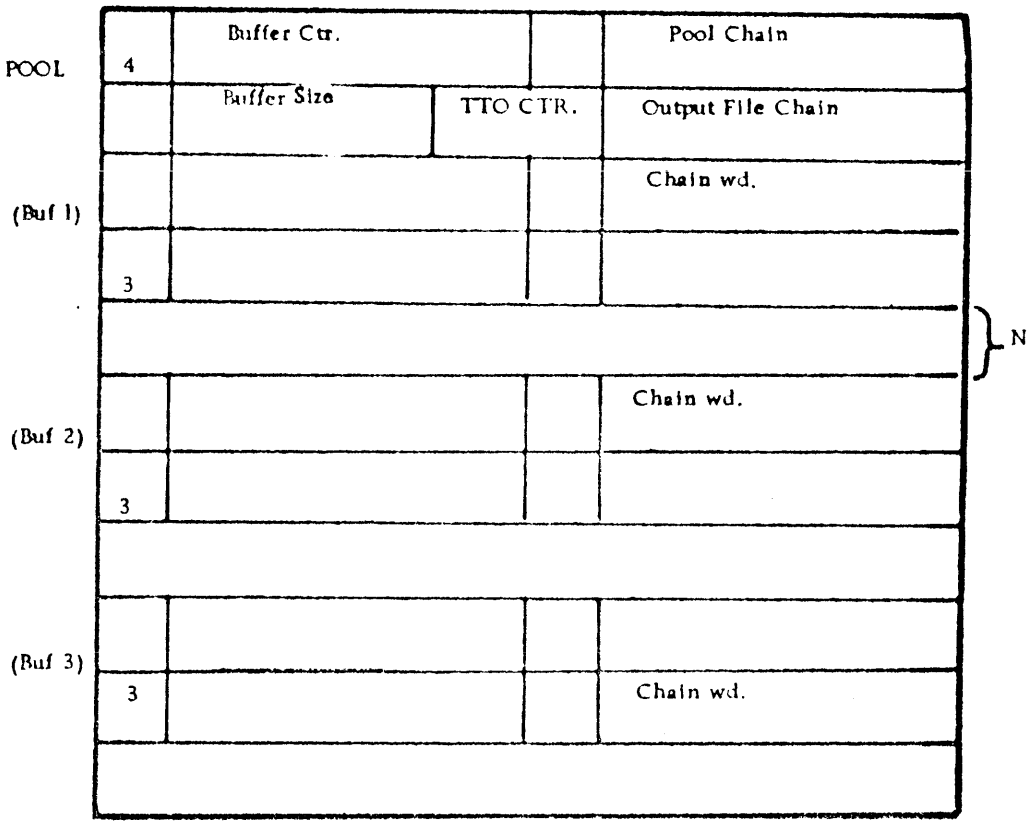

UCW
I/O UNIT ADD.
L(File)
Location of sel. rtn.
Request Chain
File Count
Record Count


Regen. chain for Int. File
FILE
UCW2
UCW1
27 28
Output file chain or ct. rd. req
rd buff in use
(buffers ahead)
Buffer in use - If not pool
Reserve Grp. ctl. wd.
Location of pool
Sync. chain
Redundancy ctr.

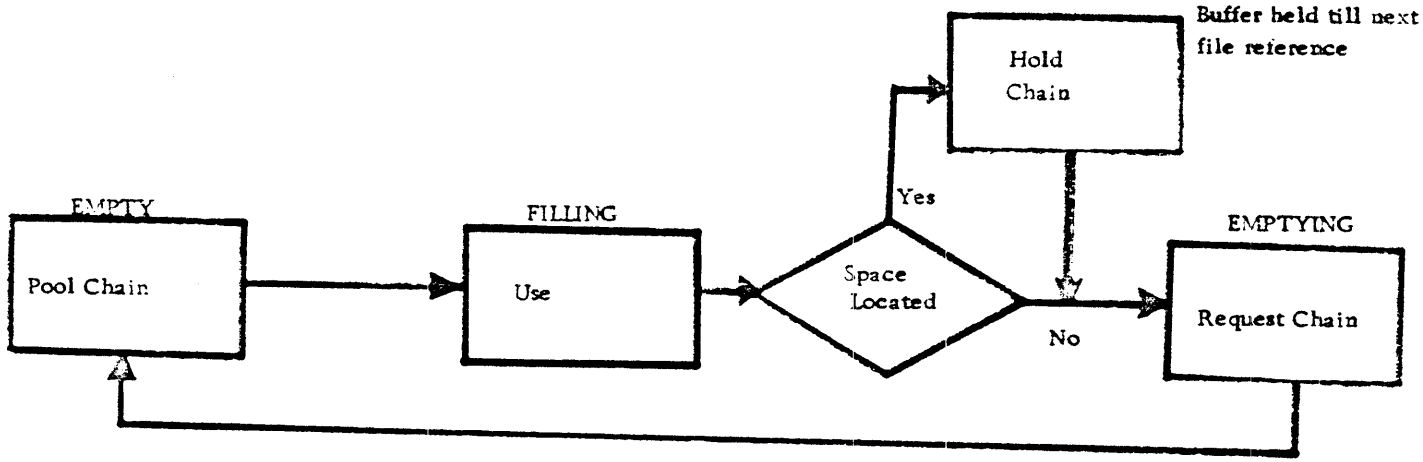

POOL
4
Buffer Ctr.
Pool Chain
Buffer Size
TTO CTR.
Output File Chain
(Buf 1)
Chain wd.
3
N
(Buf 2)
Chain wd.
3
(Buf 3)
3
Chain wd.


UCW
I/O UNIT ADD.
L(File)
Location of sel. rtn.
Request Chain
File Count
Record Count

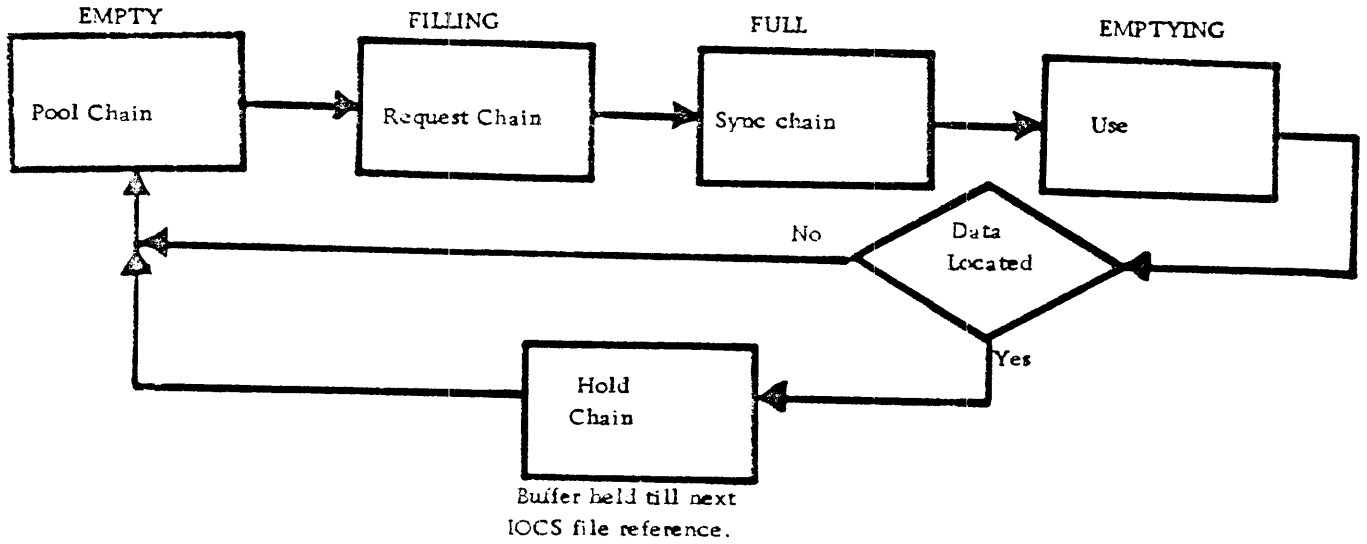

Regen. chain for Int. File
FILE
UCW2
UCW1
27 28
Output file chain or ct. rd. req
(rd buff in use)
(buffers ahead)
Buffer in use - If not pool
Reserve Grp. ctl. wd.
Location of pool
Sync. chain
Redundancy ctr.


POOL
4
Buffer Ctr.
Pool Chain
Buffer Size
TTO CTR.
Output File Chain
(Buf 1)
Chain wd.
3
N
(Buf 2)
Chain wd.
3
(Buf 3)
3
Chain wd.

OUTPUT

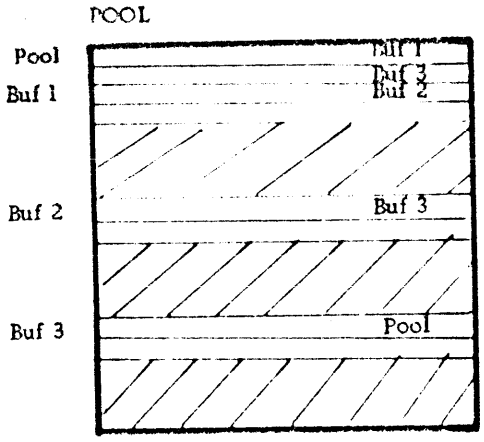


INPUT

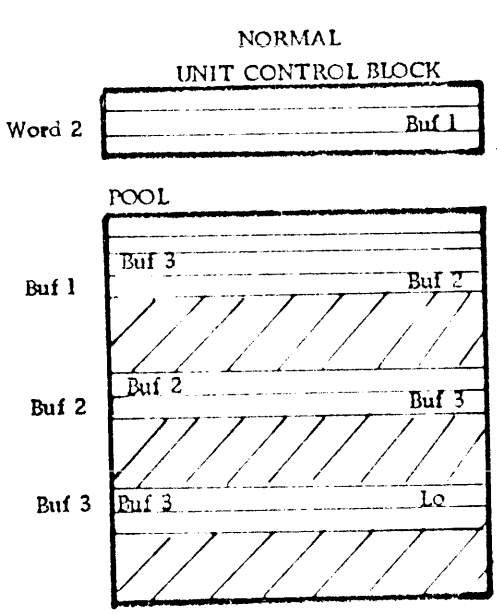


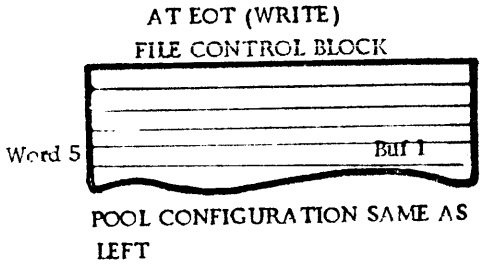
TRANSFER OF BUFFER LOCATIONS BETWEEN CHAINS

POOL CHAIN

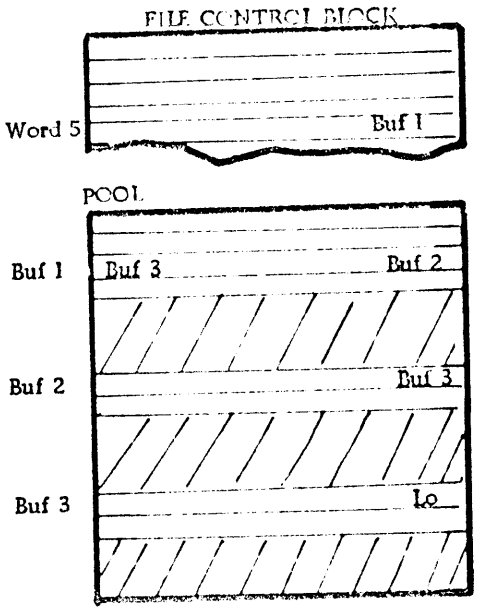


REQUEST CHAIN

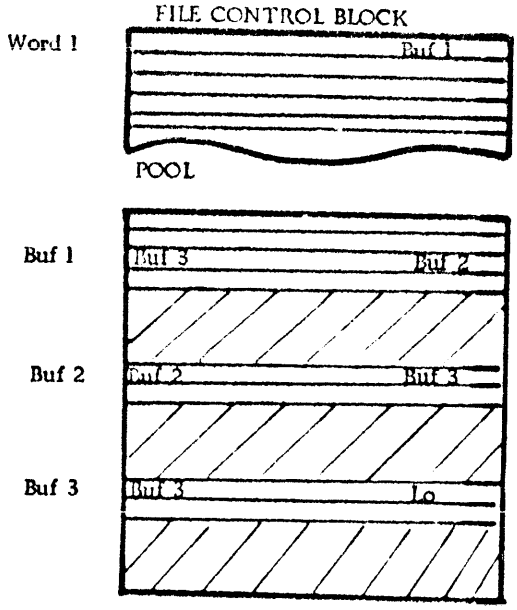




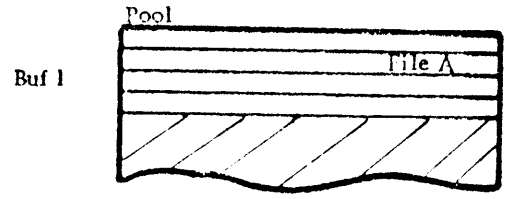
SYNC CHAIN

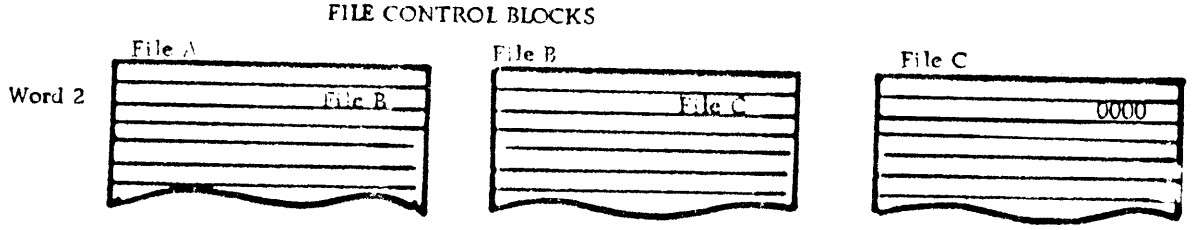


REGEN CHAIN
(INTERNAL FILE)



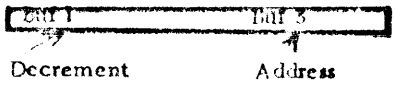
OUTPUT FILE CHAIN

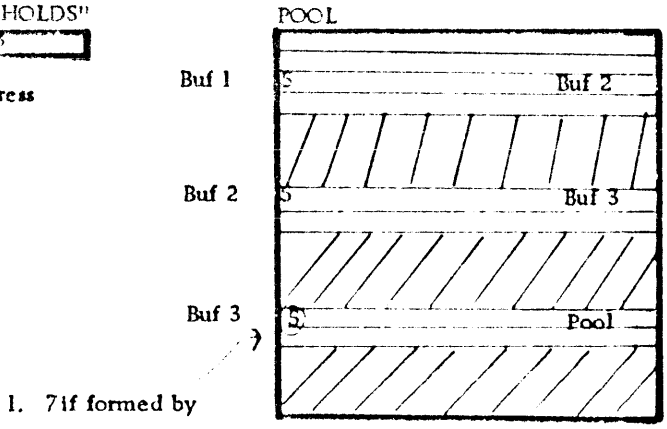




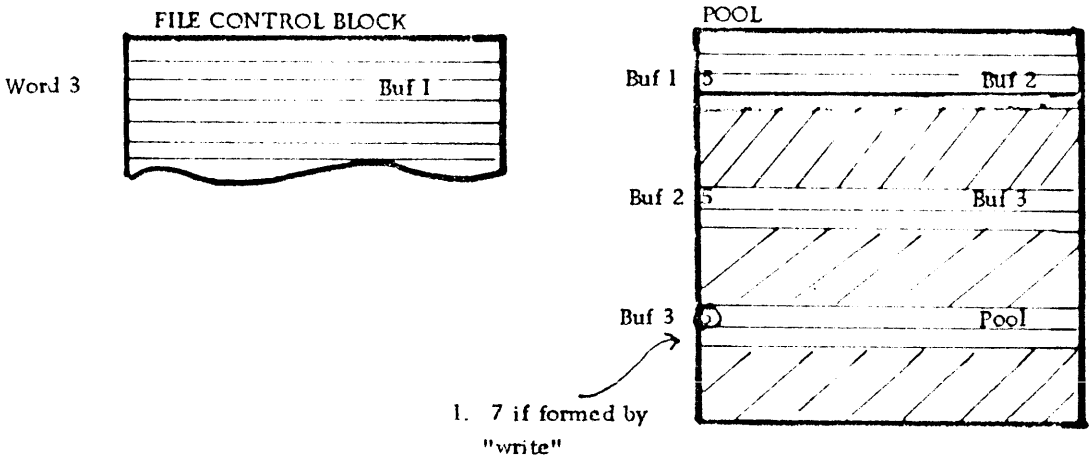
HOLD CHAIN PRIOR TO EXIT FROM READ OR WRITE ROUTINE

(05146) STORAGE LOCATION "HOLDS"





HOLD CHAIN AFTER EXIT



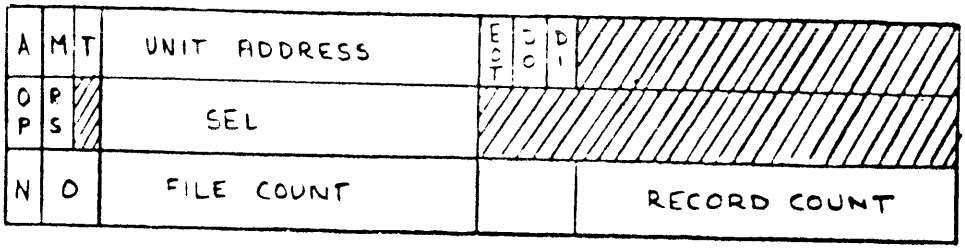
IOCS Operation	IOCS Subroutine	COMMENTS	Buffer put in request	Buffer put in use	Buffers in use (FCBW2)	Buffers ahead (FCBW2	Buffers synced (synct)	Rush (SI 13)	GIVUP (SI 14)	Rush history (SI 17)	Buffer Control
ATTACH	OPEN	Initiate filling 1st buffer									
(Assume a time lapse not suitable - to fill the buffer)											
READ	READ	GIVUP On?									Build up reserve
IORP	SYNCR	Any buffer ready to put in use? Request exist? Last buffer synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers ahead? GIVUP On? Rush occur?									
IORT	SYNCR	Any buffer ready to put in use? Request exist? Last buffer synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers ahead? GIVUP On? Rush occur?									
	RDPRI	Buffers ahead>synced by this read (synct)? Reinitiate later on? Rush history on?									
(Assume a time lapse suitable to satisfy all requests)											
READ	READ	GIVUP On?									Maintain Reserve
IORP	SYNCR	Any buffer ready to put in use? Last buffer synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers ahead? GIVUP On? Rush occur?									
IORT	SYNCR	Any buffer ready to put in use? Last buffer synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers ahead? GIVUP On? Rush occur?									
	RDPRI	Buffers ahead > synced by this read (synct)? Reinitiate later on ? Rush history on?									
(Assume a time lapse suitable to satisfy all requests)											

IOCS Operation	IOCS Subroutine	COMMENTS	Buffer put in request	Buffer put in Use	Buffers in use (FCBW2)	Buffers Ahead (FCBW3)	Buffers Synced (synct)	Rush (SII3)	GIVUP (SII4)	Rush History (SII7)	Buffer Control
Read IORT	Read	GIVUP On?									Reduce Reserve
	SYNCR	Any buffer ready to put in use? Last buffer synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers Ahead? GIVUP On? Rush occur?									
	RDPRI	Buffers ahead > synced by this read Reinitiate later on?									
Read IORT	Read	GIVUP on?									Maintain Reserve
	SYNCR	Any buffer ready to put in use? Last synced?									
	POOLRD	Reduce count of buffers in use.									
	CKGIV	Buffers ahead? GIVUP On? Rush occur?									
	RDPRI	Buffers Ahead > synced by this read (synct)? Reinitiate later on? Rush history on?									
Read IORT	Read	GIVUP On?									Maintain Reserve
	SYNCR	Any buffer ready to put in use? Last synced?									
	POOLRD	Reduce count of buffers in use									
	CKGIV	Buffers ahead? GIVUP On? Rush Occur?									
	RDPRI	Buffers ahead> synced by this read (synct)? Reinitiate later on? Rush history on?									

VI. TRAP SUPERVISOR WRITEUP

Unit Control Block

Each I/O device which is part of the standard configuration is represented in IOEX by a 3-word Unit Control Block (UCB). The format of this block is as follows:



Word 1 A - Unit availability flag:

0 - unit has been made available for use.
1 - unit is not available. (The test for this condition is made in the ACTIV routine and in the non-data select routine, NDSEL).

M - {Unit attachment flag (A = 1)
{Restart repositioning control flag (A = 0)

Unit attachment flag —

0 - unit is connected to this channel and can be made available.
1 - unit is not attached to the channel.

Restart repositioning control flag:

0 - unit will be repositioned when restarting.
1 - unit will not be repositioned when restarting.

T - Unit type (for tapes only):

0 - Model II 729
1 - Model IV 729

Unit address - address of the I/O unit (if the unit is a tape, the address is the BCD mode address, e.g., 1201).

EOT - End-of-tape flag:
0 - no end-of-tape has occurred on this unit.
1 - an end-of-tape has occurred while writing on this unit.

D0 - Tape density at the load point:
0 - low density.
1 - high density.

D1 - Density at the current tape position:
0 - low density.
1 - high density.

Word 2 OP - Select Type:
0 - read.
1 - write.

RS - Permanent redundancy message control:
0 - If a permanent read redundancy occurs, a message is printed.
1 - if a permanent read redundancy occurs, error message is by-passed.

SEL - Location of the data select routine

Word 3 N - Noise record flag (reading only):
0 - no noise records have been detected while reading.
1 - one or more noise records have been detected while reading.

File count - Number of file marks written on or read from this tape.

Record count - Number of records which have been written on or read from the current file.

Notes:

a. The EOT flag and noise record flag are turned off only when the tape is returned to the rewound position.

b. The record count is complemented when backspacing from an end-of-file. A backspace which repositions in front of a file mark gives a record count (18-35) of 777777_8, for example. When writing occurs from such a position, the two low-order tag bits are cleared, to prevent a spurious increase in the file count.

Data Transmission and Priorities

The initiation of data transmission operations for a unit, and the maintenance of any request queue for the unit, is the responsibility of the user of IOEX. These functions must be provided in a subroutine, SEL, which is entered twice for each data transmission operation which results in a trap. The calling sequence is:

```
TSX    SEL,4
Return 1
Return 2
Return 3
```

For either entry:

C(IR1) = - (channel index: 0 = A, 1 = B, etc.)
A_{21-35} = address of the UCB

Trapping is disabled before entry to SEL.

A. Select Entry.

If the sign of the accumulator is plus on entry to SEL, the routine must initiate a data select operation which will cause a trap. Return 1 is always used with SEL^{+}. The sign of Word 2 of the UCB must be set according to the operation (+ for RDS, - for WRS).

If the operation is RDS, redundancy checking can be suppressed by

```
STZ    RCTX,1
```

B. Posting Entry.

If the sign of the accumulator is negative on entry to SEL, a trap has occurred as the result of the previous select for the unit. The following information is furnished:

1. Sense indicators:
   S - Noise record flag
   1 - the record was not an apparent noise record.
   0 - the record was an apparent noise record.

1 - End-of-file (read) or end-of-tape (write).
0 - no end-of-file or end-of-tape.
1 - end-of-file or end-of-tape.

2 - Permanent redundancy (read only)
0 - no permanent redundancy.
1 - permanent redundancy.

Note: Bit S is on in all cases except when a noise record is detected.

2. Cell COMM - results of a store channel for the operation.

3 URRX, 1 table - redundancy counts:

PZE N1,,N2

N1 - number of recovery entries to SEL^{+}.
N2 - 0 - no permanent redundancy (while reading) or no erase areas written.
1 - permanent redundancy (reading) or one or more erase areas (writing). If a permanent redundancy, N1 = 0.

The normal return from SEL^{-} is Return 1. The exit conditions are described fully in connection with redundancy recovery.

Note: If EOT is detected while writing, the EOT bit is set on in the UCB, and the EOT indication is given on each subsequent entry to SEL^{-} for the unit until the tape is rewound.

The tables RCTX, 1 and URRX, 1 can be indirectly addressed by means of the communication cells RCTX1 and URRX1, respectively.

When an end-of-file indication is given to SEL^{-}, the tape position has already been adjusted. The communication cell LTPOS contains the tape position prior to this adjustment. A redundancy indication cannot occur together with an end-of-file indication.

Design of SEL Routines

1. SEL must not destroy the contents of IR1. IR2 and the sense indicators need not be saved by SEL.

2. UCB Word 2 should never be cleared by SEL^{+} - a posting entry should always be provided. SEL^{-} should clear Word 2 if there are no more requests for the unit.

3. SEL must not modify the unit address in UCB Word 1.

Use of the UCB

Word 2 of the UCB being non-zero indicates that the unit has something to do. This cell must be set and reset by the user of IOEX.

Normally, any direct reference to a UCB at other than trap (SEL) time should be trap-protected by the sequence.

```
ENB     L(0)
.
.
.
ENB*    TRAPX
```

Bit positions 21-35 of UCB Word 1 and Bit 2, 18-35 of UCB Word 2 are available to the user of IOEX.

Activating a channel and/or assigning priority.

A channel which is dormant can be activated, or a unit can be given top priority on a channel, by use of the routine ACTIV. The calling sequence is:

```
TSX     ACTIV,4
P       A,T
Return
```

Where A,T is the location of a cell which contains

```
OP      UCB,X,Y
```

UCB is the location of the Unit Control Block (OP,X and Y are ignored by IOEX).

P = PZE: activates a dormant channel. The specified unit is given priority on the dormant channel. If the channel is active, the entry has no effect.

P = MZE: the specified unit is given top priority on the channel, whether active or dormant. If the channel is dormant, it is activated. If ACTIV is entered at non-trap time, there is a hold until a trap occurs on the channel. Note: at non-trap time, if a priority entry is made for a dormant channel, this channel is held until the completion of the operation which activates the channel.

A transfer to ACTIV+1 skips validity testing for the unit (availability bit).

Priority determination

The selection of the unit on a channel to be activated is made as follows:

1. The channel priority cell is examined. If the cell is not zero, its address is interpreted as the location of the UCB for the unit to be activated next. The second word of this UCB is tested:

   a. non-zero: do SEL^+ for the unit.
   b. zero: do (2).

2. If the priority cell is zero, the channel is scanned for a waiting unit. If one is found, the location of its UCB is placed in the priority cell, and SEL^+ is entered for the unit. If no unit is waiting the channel is allowed to go dormant.

Once activity begins on some unit of a given channel, that unit continues to retain its priority until all of its waiting operations have been completed, unless a priority entry is made to ACTIV for some other unit on the channel.

Non-Data Selects

Non-Data selects are executed by the sequence

```
TSX     NDATA,4
PZE     A,T,NDS
```

Where A,T is the location of a cell of the form

```
OP      UCB,X,Y
```

UCB is the location of the Unit Control Block for the unit (OP,X and Y are ignored).

NDS =	NOP -
0	
1	SDNL
2	SDNH
3	REW
4	RUN
5	BSR
6	BSF
7	WEF

Comments:

1. The user must insure that non-data selects are properly synchronized with any data selects for the unit. At non-trap time, this normally means that the unit must be free --- i.e., the 2nd word of the UCB must be zero.

2. Non-data selects other than WEF are simply stacked into the channel without checking to see whether the channel is free or not.

3. Non-data selects for card equipment are ignored.

4. Transfer to NDATA+1 skips validity testing of the unit.

5. Backspace record across the previous file mark complements the record count in bits 18-35.

6. WEF causes the channel to be freed, after which the following sequence is executed:

```
WEFX
TCOX *
TRCX
ETTX
```

Recovery is attempted if a redundancy occurs while writing an end-of-file mark: the tape is backspaced, and the file mark is rewritten and checked, as often as necessary. After each group of 25 recovery attempts, a message is printed. If the EOT condition is detected following the writing of any redundant file mark except the first, an error halt is made. If the EOT condition is detected after a non-redundant end-of-file, a return to 2,4 is made. The normal return for WEF is 3,4.

Redundancy Recovery

Writing

The redundancy trapping mode is used for write operations. "Disable redundancy checking" control is not available. If the first attempt to write a record produces a redundancy trap, the following procedure is used:

a. The tape is backspaced one record.
b. An erase area is written. If this operation produces a redundancy check, an operator note is printed.

c. The record is rewritten (i.e., SEL+ is entered) and checked for redundancy.
d. Steps a - c are repeated until the record is either written correctly, or until the EOT condition is sensed. After each group of 25 erase areas, an operator message is written.

Note: IOEX stops when the EOT condition occurs during redundancy recovery. However, there is always at least one erase - rewrite sequence attempted and if the rewrite is successful, the stop does not occur.

If, following the successful writing of a record, IOEX determines that the apparent record length was less than 3 words, an entry is made to SEL- with the noise record condition indicated Two exits are available for this condition:

Return 1	the record is accepted.
Return 2	the record is accepted, and an operator note is printed, indicating that a short record has been written.

The record count is increased before entry to SEL-, it is not increased before rewrite entries to SEL+.

On each rewrite entry to SEL+ during redundancy recovery, URRX, 1 has the following configuration:

```
PZE    N,,1
```

where N is the number of consecutive erase areas which have been written. Following a successful redundancy recovery, at the entry to SEL-, URRX, 1 has the same configuration. N in this case is the total number of erase areas written on this recovery.

Reading

The redundancy trapping mode is not used during read operations.

If a redundancy occurs as the result of a reading operation, the following steps are taken:

1. RCTX, 1 is tested. If redundancy recovery is suppressed, SEL- is entered with no redundancy indication. URRX, 1 is not adjusted for this redundancy. The record count is updated.

2. If the record was an apparent noise record, SEL- is entered with the noise record indication (the record count has been increased by 1, but URRX, 1 has

not been adjusted). The action taken by IOEX after return from SEL⁻ depends upon the return:

Return 1 - IOEX assumes that the record has been accepted despite the redundancy. No messages are printed.

Return 2 - The redundant record is considered noise, and is discarded. An operator message is printed, and the noise record bit is set on in the UCB. The record count is reduced by 1, URRX, 1 is cleared, and SEL⁺ is re-entered to read the following record.

Return 3 - the normal redundancy recovery procedure is entered (step 3 below).

3. For normal redundancy recovery, the following steps take place:

   a. If the redundancy count is URRX, 1 is 1, a tape cleaner action is taken, provided the following conditions are satisfied:
      1) The noise record bit is off.
      2) There are at least two prior records in the current file.

      The tape cleaner action consists of backspacing over the redundant record and the two prior records, giving two dummy reads, and then entering SEL⁺ to re-read the redundant record. If either of the two conditions stated above is not fulfilled, the tape cleaner action is not carried out, and a single backspace over the redundant record is made.

   b. When SEL⁺ is entered to re-read the redundant record, the record count has not been increased, and URRX, 1 indicates the total number of previous attempts to read the record.

   c. If the redundancy count in URRX, 1 is 10, a permanent read redundancy is assumed. The record count is increased, an operator message is printed, PZE 0, , 1 is stored in URRX, 1 and SEL⁻ is entered with the permanent redundancy indication.

      Note: printing of the permanent redundancy message can be suppressed by setting the corresponding control bit on in UCW Word 2. The redundancy indication is still given, but URRX, 1 is not modified.

Apparent Noise Records.

After each successfully completed write operation, and after a redundant read operation, a test is made for an apparent short (noise) record. This test is made as follows:

1. The decrement of the Store Channel word is reduced by 1 to obtain the location of the last IO command executed.

2. The address of this IO command is subtracted from the address of the Store Channel word to obtain an apparent word count. If this word count is less than 3, IOEX gives the noise record indication to SEL⁻.

Obviously, there are sequences of IO commands which will produce this noise record condition, even when the true record length is greater than 2 words.

Note: no test is made for use of the indirect addressing feature of IO commands; if this feature is used, a short record will normally not be detected.

Freeing a Channel

It is sometimes necessary to perform some I/O operation immediately, interrupting the normal sequence of operations on a channel. This can be done with the subroutine FRCHX. The calling sequence for this routine is

```
TSX    FRCHX,4
PZE    UCB,,TRP
```

Where UCB is the location of a Unit Control Block on the channel to be freed (normally the unit to be used) and TRP is the location of the exit point from the routine which transfers to FRCHX.

Rules:

1. All trapping should be disabled before using FRCHX by giving

```
ENB    L(0)
```

2. The subroutine exit point specified in the FRCHX calling sequence must be of the general form

```
TRA    **
```

with the address preset on entry to the routine.

3. On exit from FRCHX, all trapping remains disabled. The desired I/O operation must be carried out without using data channel trapping.

4. The user is responsible for testing and turning off any indicators (e.g., EOT, redundancy) which may be turned on by the operation.

5 FRCHX may be entered either at non-trap or at trap time. If at non-trap time, trapping is enabled at the subroutine exit point. If at trap time, trapping is enabled at the exit from IOEX corresponding to the uncompleted trap.

6. If FRCHX is entered when the referenced channel is trapped, or dormant, or already freed, the subroutine takes no action.

7. If the operation following FRCHX changes a tape position, the corresponding adjustment should be made to UCB Word 3.

Reset and Load Channel Table

The SEL+ routine can make use of the IOEX table of Reset and Load Channel instructions. This should be done indirectly through the communication cell RCHXI:

```
RCHXI   PZE     RCHX,1
```

The form of the table itself is:

```
RCHX    RCHA  **
        RCHB  **
          .
          .
          .
```

On-Line Messages

Messages can be printed on-line, using the subroutine MWR. The calling sequence for this routine is

```
TSX     MWR,4
PZE     N
P       L1,T1,M1+512*SPR1
P       L2,T2,M2+512*SPR2
.
.
.
P       LN,TN,MN+512*SPRN
```

Where N = number of following entries in the calling sequence. P,L,T,M: M words (six characters each) beginning in location L,T are converted to hollerith and placed in the line image for printing. If P = PZE, the image is taken to be complete and the line is printed. If P = MZE, this line is considered incomplete, and the L,T,M of the next calling sequence entry are used to continue building the image, beginning with the next print position to the right.

SPR: if P = PZE, the sense exit SPR is activated after the line is printed. (A SPR appearing in a word with P = MZE is ignored). To activate an exit before printing the first line, an entry should be made in the calling sequence of the form

```
PZE     **,,5i2*SPR
```

This will print a blank line, followed by activation of the hub SPR.

MWR can be used either at trap time or at non-trap time. Printing is immediate; that is, FRCHX is used, and the printing operation has been started before the return from MWR.

VII. MAP OF KEY CELLS IN IOCS-2 Channels 10 Tapes each

Location (octal)	Name	Content & Use	
0 - 11		Not used	
12		0 XXXXX 0 XXXXX	Ch. A. Trap
13		TTR 313	
14		1 XXXXX 0 XXXXX	Ch. B. Trap
15		TTR 311	
16		2 XXXXX 0 XXXXX	Ch. C. Trap
17		TTR 311	
20		3 XXXXX 0 XXXXX	Ch. D. Trap
21		TTR 311	
22		4 XXXXX 0 XXXXX	Ch. E. Trap
23		TTR 311	
24		5 XXXXX 0 XXXXX	Ch. F. Trap
25		TTR 311	
26		6 XXXXX 0 XXXXX	Ch. G. Trap
27		TTR 311	
30		7 XXXXX 0 XXXXX	Ch. H. Trap
31		TTR 311	
32-43		Not used	
44	IOCS	TTRJOBEND	
45	STOPD	HTR 45 IOCS Stop	
46	PAUSED	HPR 77777 IOCS Pause	
160	TRAPX	21-35 = L(ENB Cell)	
161	TRAP	0 00XXX 0 00377	Normal Enable Cell (redundancy bits on if ch. is writing)
162	COMM	Result of last store channel	
204-206	UEA11	Unit Control Words Ch. A. Card Reader	
207-211	UEA12	Unit Control Words Ch. A. Punch	
212-214	UEA13	Unit Control Words CH. A. Printer	
215-217	UEA1	Unit Control Words Ch. A. Tape 1201	
220-222	UEA2	Unit Control Words Ch. A. Tape 1202	
223-225	UEA3	Unit Control Words Ch. A. Tape 1203	
226-230	UEA4	Unit Control Words Ch. A. Tape 1204	
231-233	UEA5	Unit Control Words Ch. A. Tape 1205	
234-236	UEA6	Unit Control Words Ch. A. Tape 1206	
237-241	UEA7	Unit Control Words Ch. A. Tape 1207	
242-244	UEA8	Unit Control Words Ch. A. Tape 1210	
245-247	UEA9	Unit Control Words Ch. A. Tape 1211	
250-252	UEA10	Unit Control Words Ch. A. Tape 1212	
253-255	UEB1	Unit Control Words Ch. B. Tape 2201	
256-260	UEB2	Unit Control Words Ch. B. Tape 2202	

261-263	UEB3	Unit Control Words Ch. B. Tape 2203
264-266	UEB4	Unit Control Words Ch. B. Tape 2204
267-271	UEB5	Unit Control Words Ch. B. Tape 2205
272-274	UEB6	Unit Control Words Ch. B. Tape 2206
275-277	UEB7	Unit Control Words Ch. B. Tape 2207
300-302	UEB8	Unit Control Words Ch. B. Tape 2210
303-305	UEB9	Unit Control Words Ch. B. Tape 2211
306-310	UEB10	Unit Control Words Ch. B. Tape 2212
315	CHXSP	Ch. A. Priority Cell Bits 21-35= UCW with priority
316	CHXSP+1	Ch. B. Priority Cell Bits 21-35= UCW with priority
317	CHXAC	Ch. A. Activity Cell Bits S-2= 3 if waiting for trap Bits 21-35= L (active UCW)
320	CHXAC+1	Ch. B. Activity Cell Bits S-2= 3 if waiting for trap Bits 21-35= L(active UCW)
331	URRX	Ch. A. } Redundancy count cells
332	URRX+1	Ch. B. } 3-17 = if reading, 1 if perm. redundancy if writing, # of erases this record 21-35= # of entries to redundancy routine current record
337	RCHX	RCHA a ; a = L (last IO command)
340	RCHX+1	RCHB b ; b= L(last IO command)
353	SELX-7	WEF ** } last non-data selects executed
354	SELX-6	BSF **
355	SELX-5	BSR **
356	SELX-4	RUN **
357	SELX-3	REW **
360	SELX-2	SDNH **
361	SELX-1	SDNL **
412	EXIT	TTR ** exit from trap routine
5016	SELCT	RDS ** or last data select WRS
5017	SELCT+1	XEC, RCHX, 1 RCHX for last select IR1 = - (Channel-1)

IBM

709 / 7090

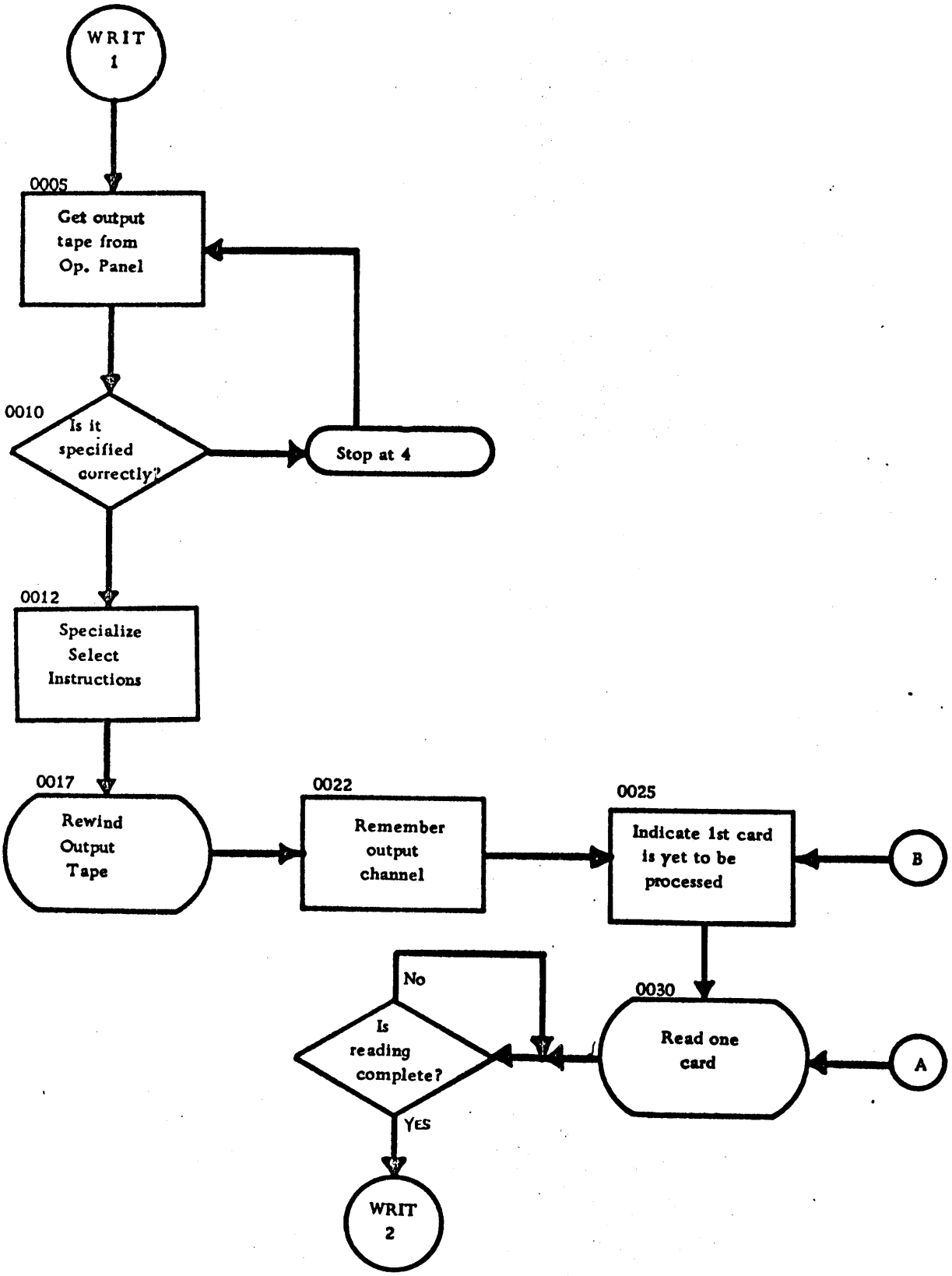
INPUT/OUTPUT CONTROL SYSTEM.

FLOW CHARTS

PREPARED BY
CUSTOMER ENGINEERING EDUCATION
POUGHKEEPSIE, NEW YORK.

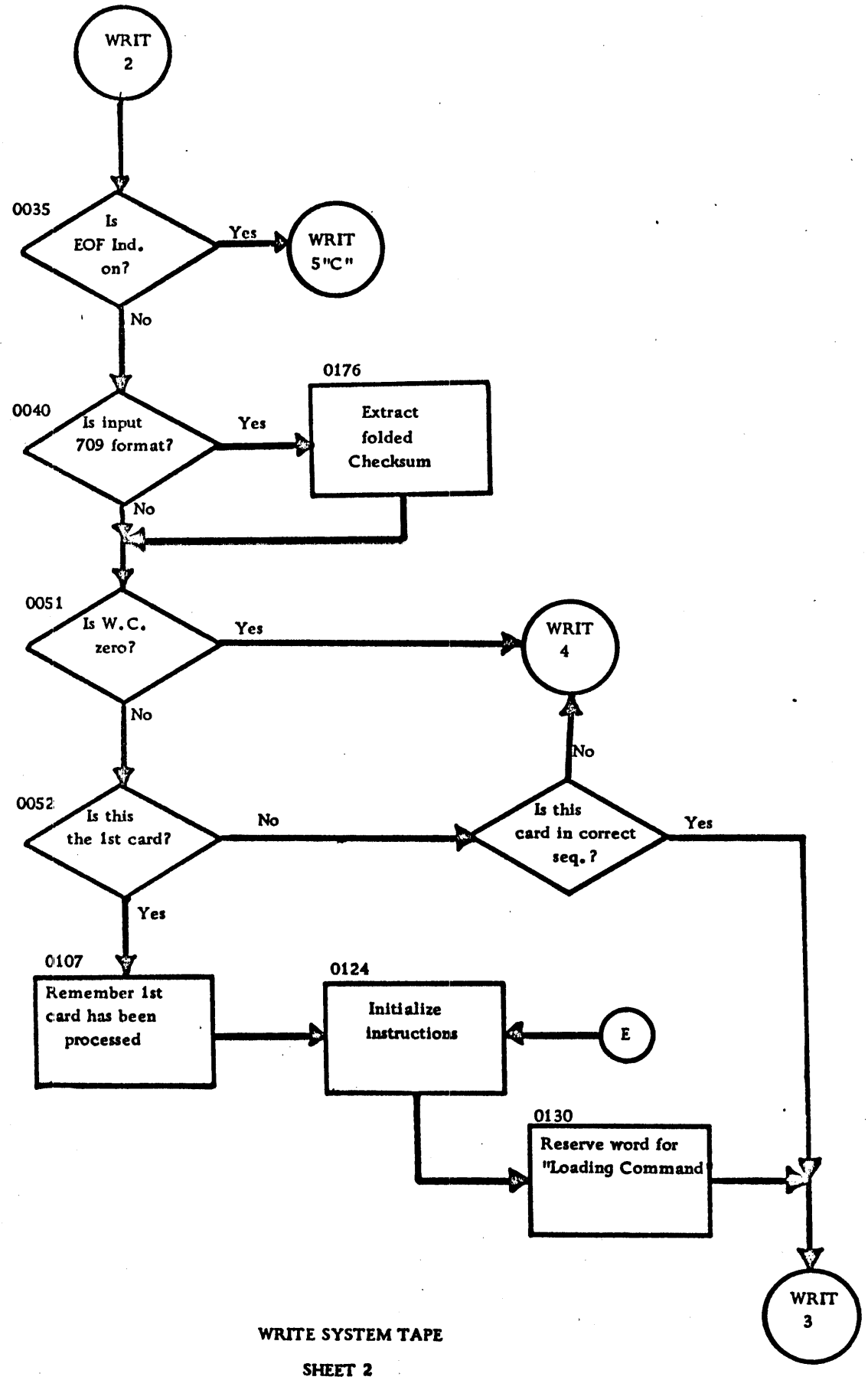
TABLE OF CONTENTS

* IOCS Routines



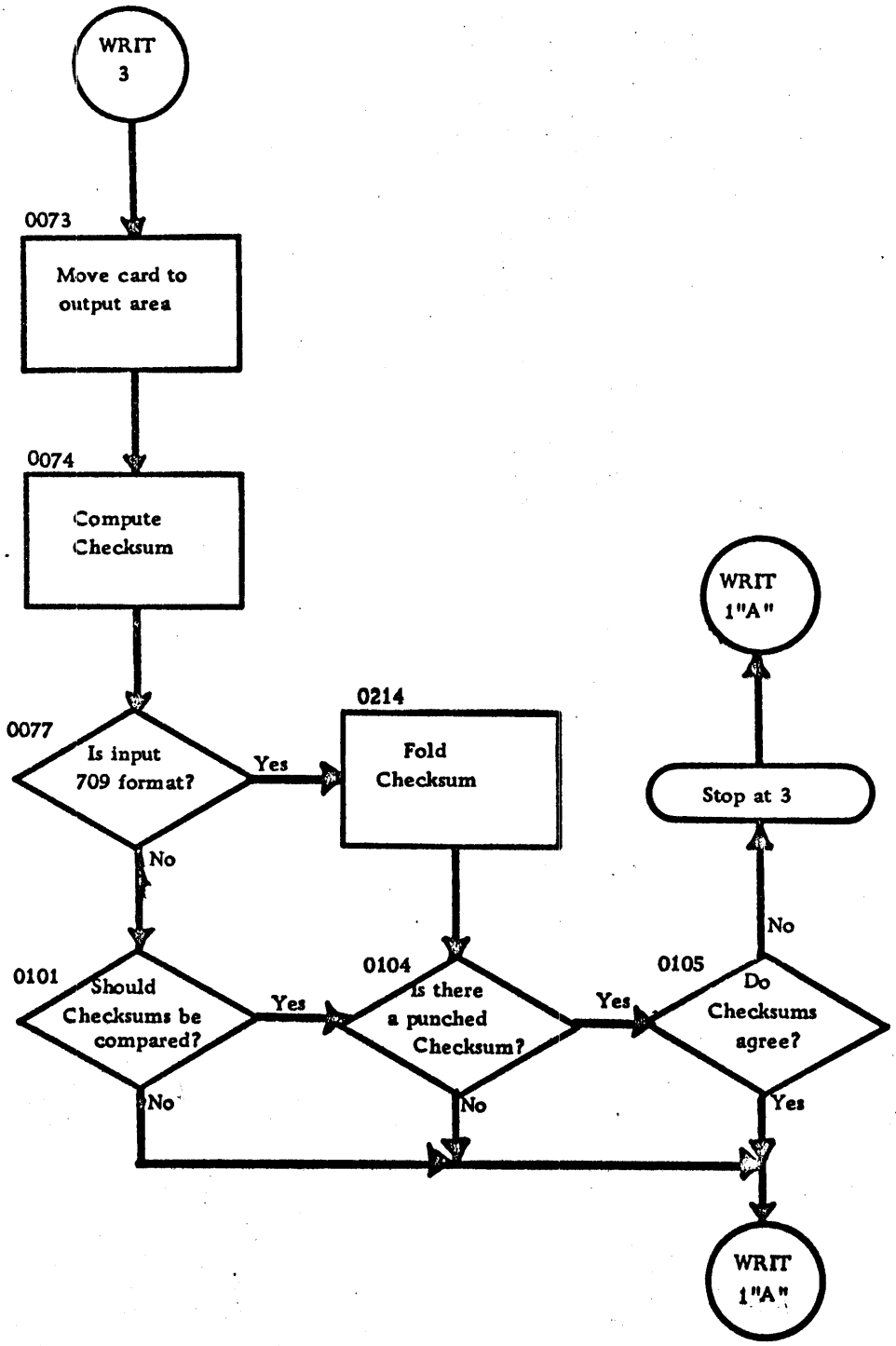
WRITE SYSTEM TAPE

SHEET 1



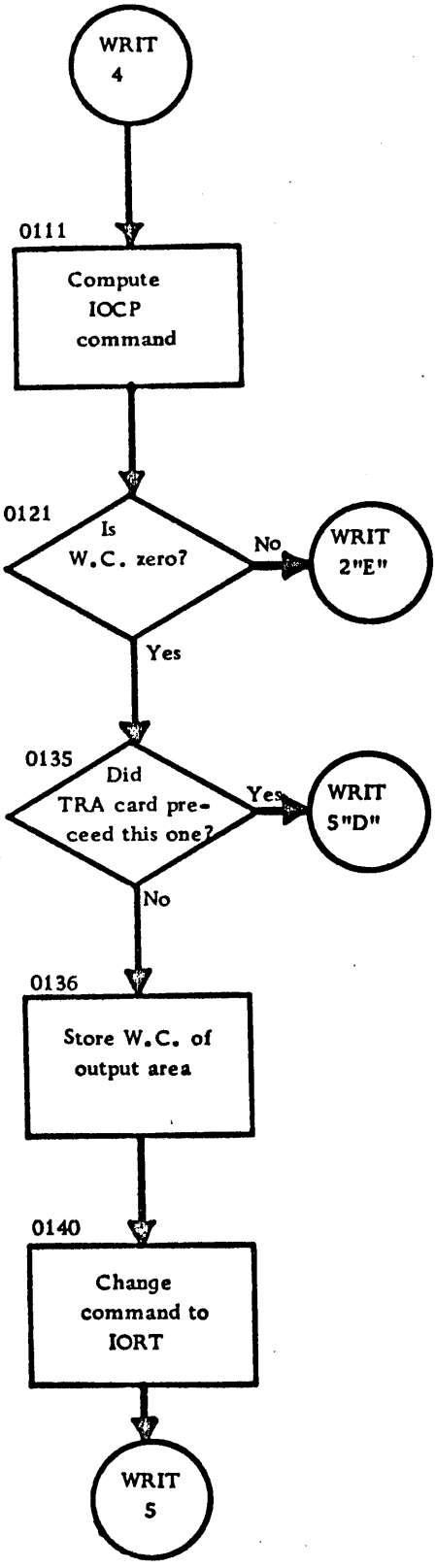
WRITE SYSTEM TAPE

SHEET 2



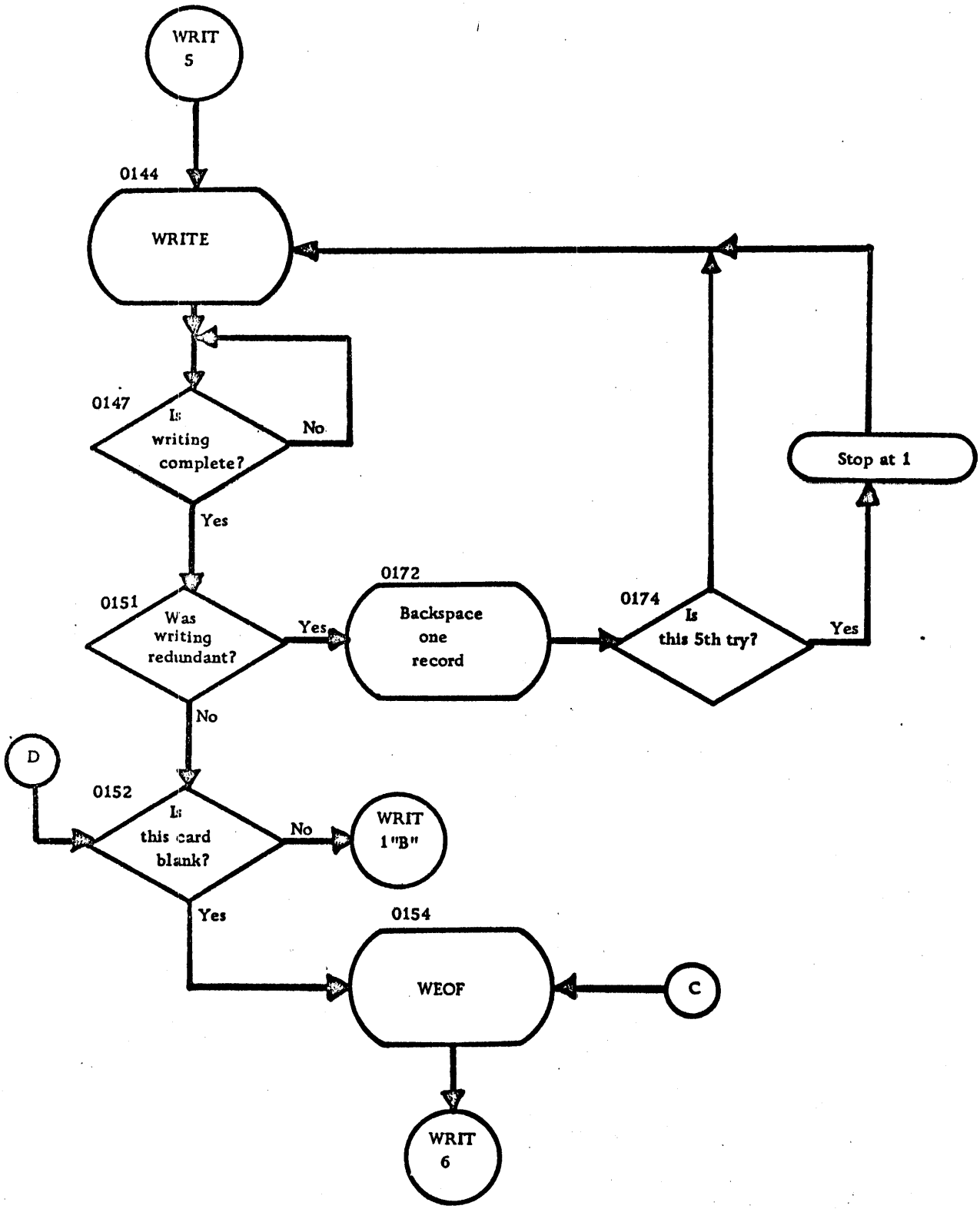
WRITE SYSTEM TAPE

SHEET 3



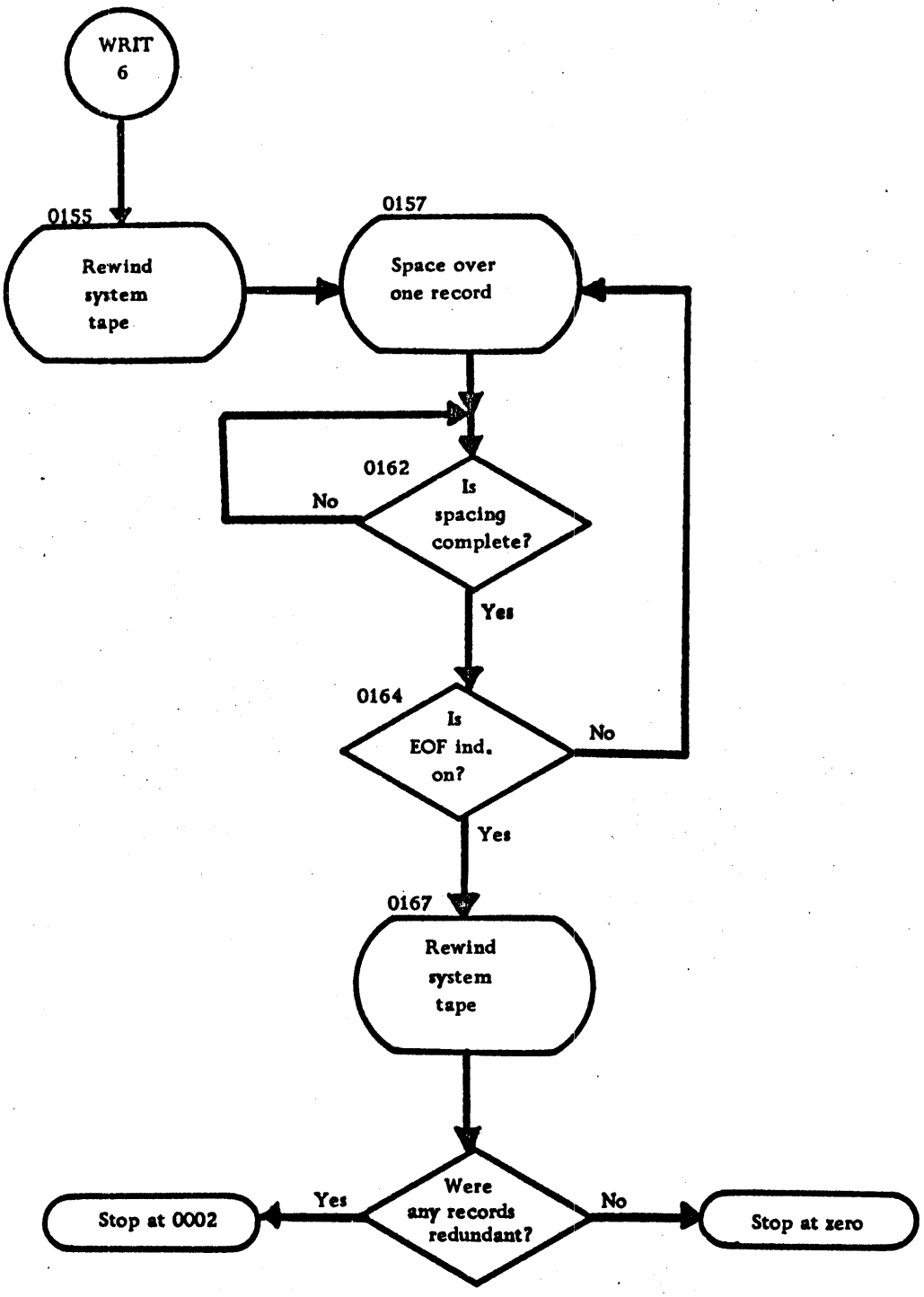
WRITE SYSTEM TAPE

SHEET 4



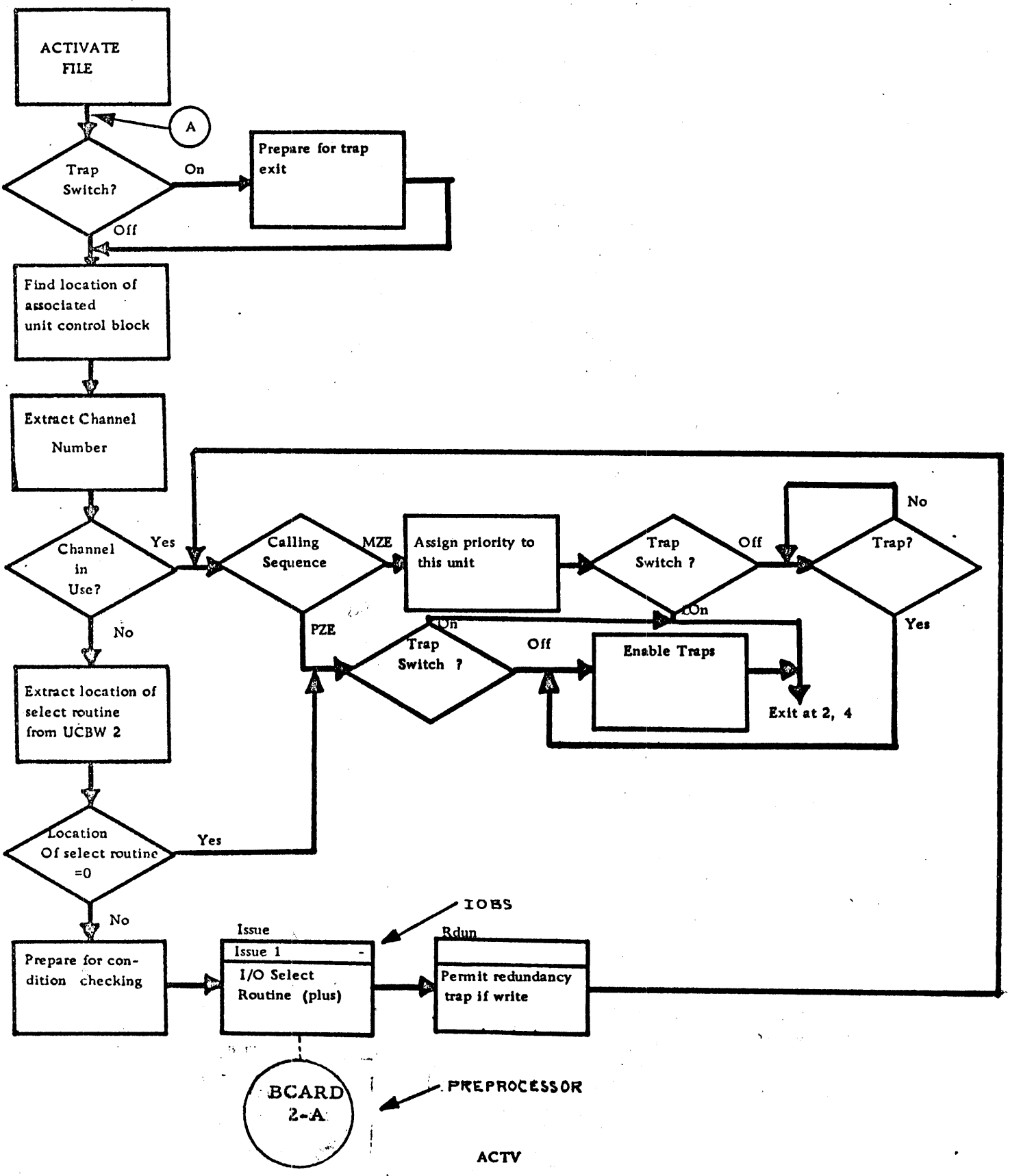
WRITE SYSTEM TAPE

SHEET 5

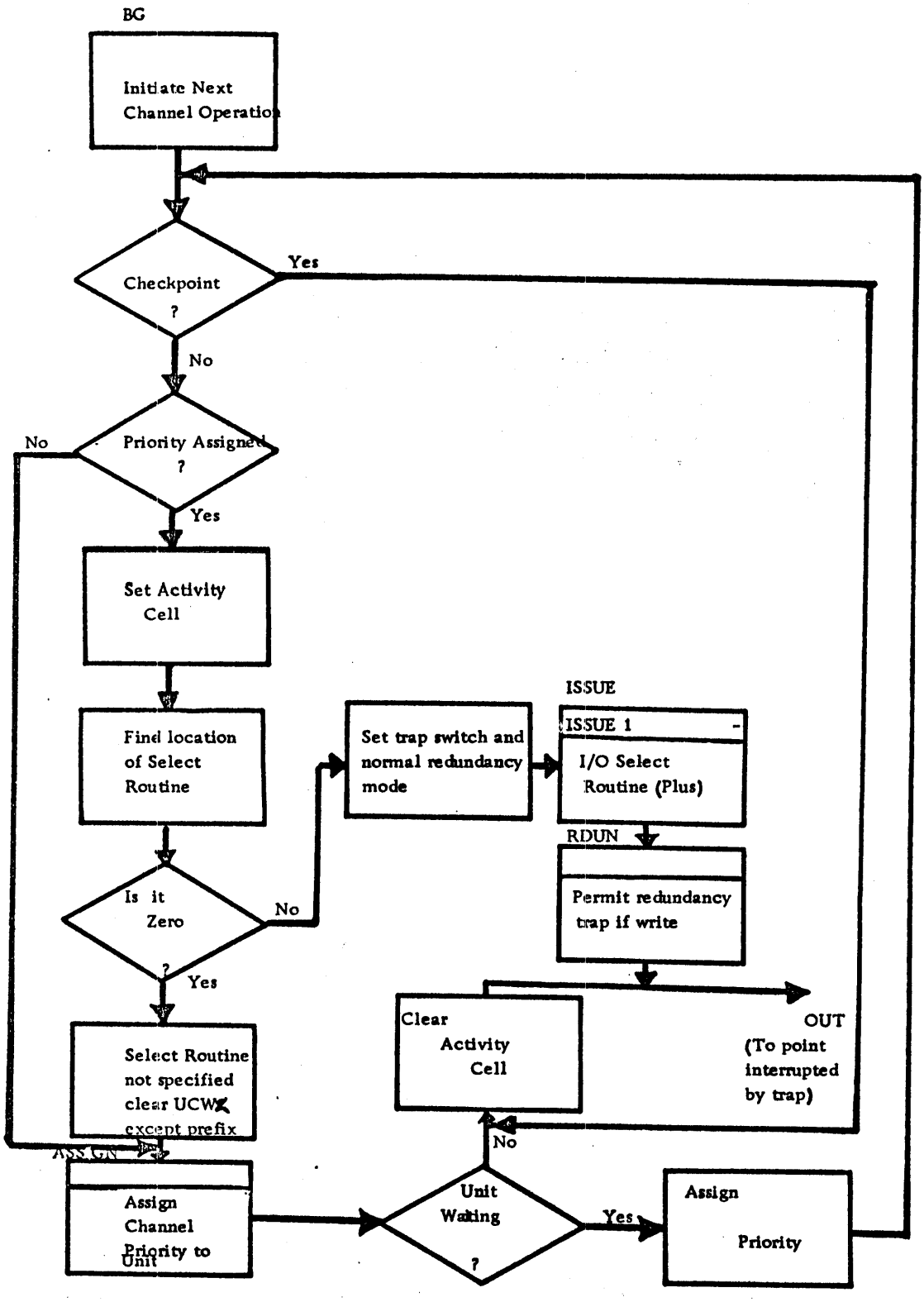


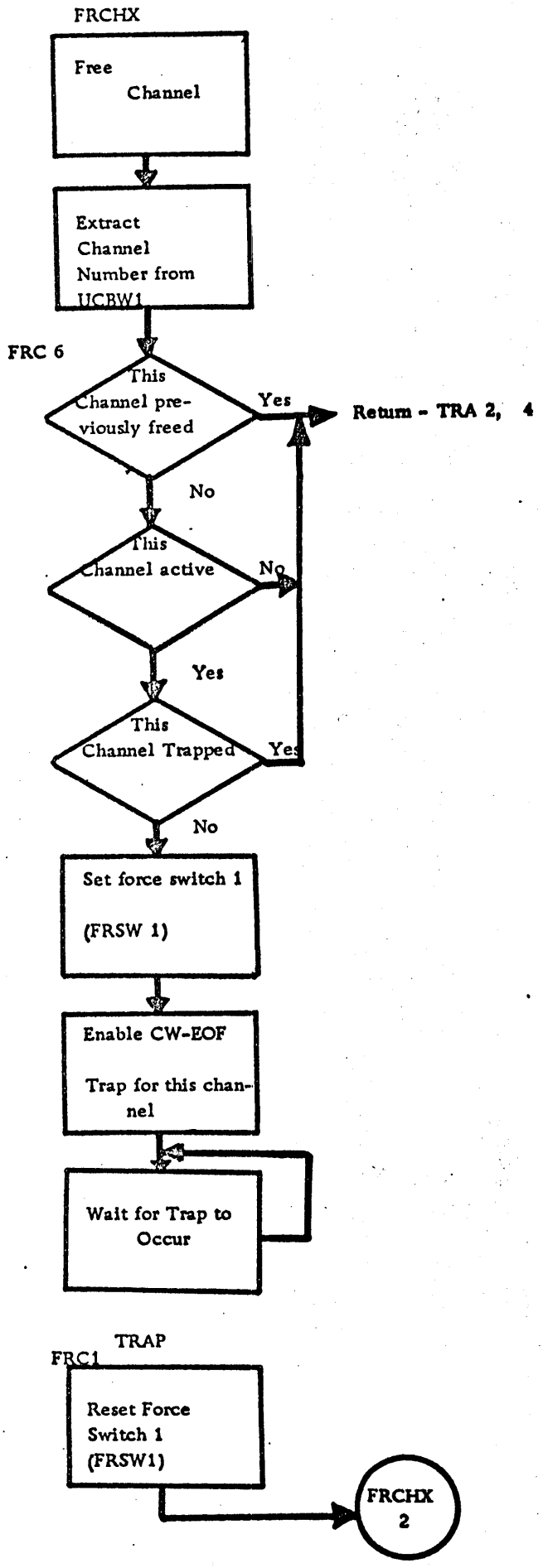
WRITE SYSTEM TAPE

SHEET 6

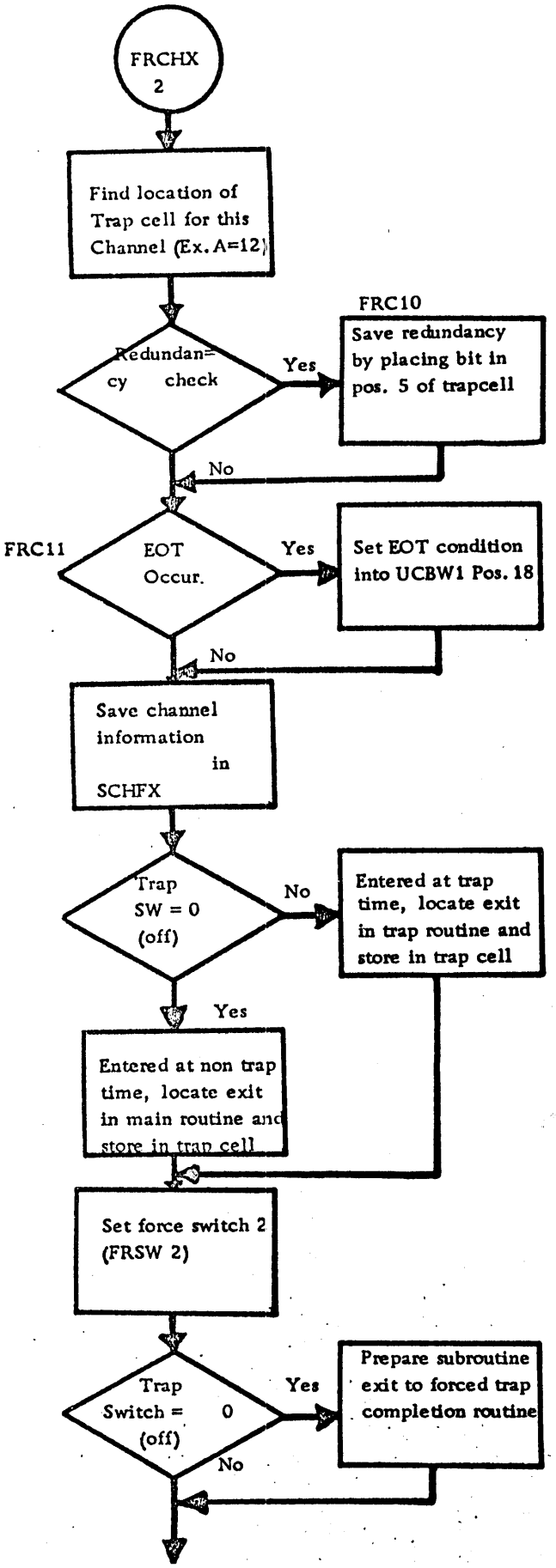

ACTIVATE FILE
A
Trap Switch?
On
Prepare for trap exit
Off
Find location of associated unit control block
Extract Channel Number
Channel in Use?
Yes
No
Calling Sequence
MZE
PZE
Assign priority to this unit
Trap Switch ?
Off
On
No
Trap?
Yes
Trap Switch ?
On
Off
Enable Traps
Exit at 2, 4
Extract location of select routine from UCBW 2
Location Of select routine =0
Yes
No
Prepare for condition checking
Issue
Issue 1
I/O Select Routine (plus)
IOBS
Rdun
Permit redundancy trap if write
BCARD 2-A
PREPROCESSOR

ACTV

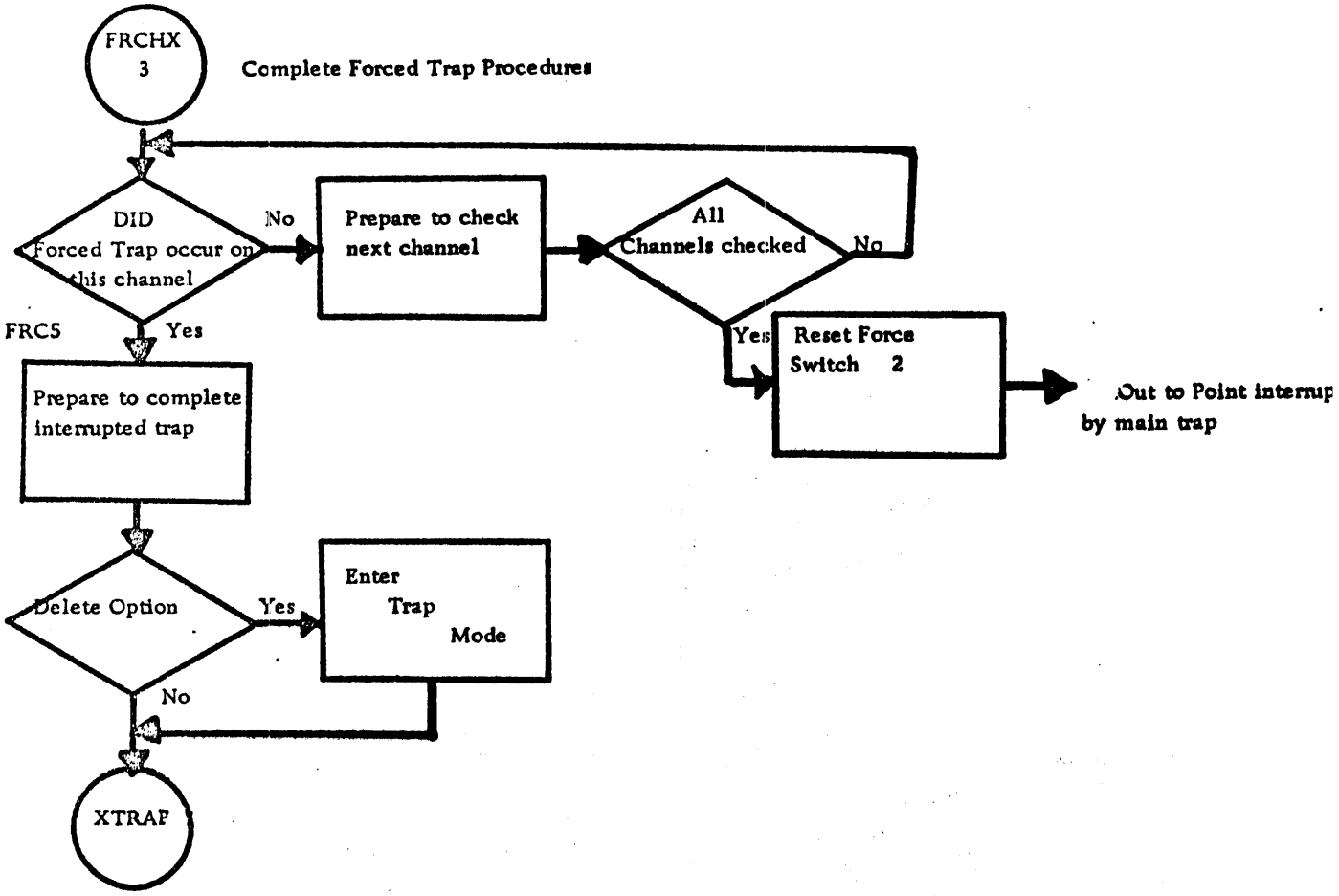

BG
Initiate Next Channel Operation
Checkpoint ?
Yes
No
Priority Assigned ?
No
Yes
Set Activity Cell
Find location of Select Routine
Is it Zero ?
No
Yes
Set trap switch and normal redundancy mode
ISSUE
ISSUE 1
I/O Select Routine (Plus)
RDUN
Permit redundancy trap if write
Select Routine not specified clear UCW except prefix
ASSIGN
Assign Channel Priority to Unit
Unit Waiting ?
No
Yes
Clear Activity Cell
Assign Priority
OUT
(To point interrupted by trap)

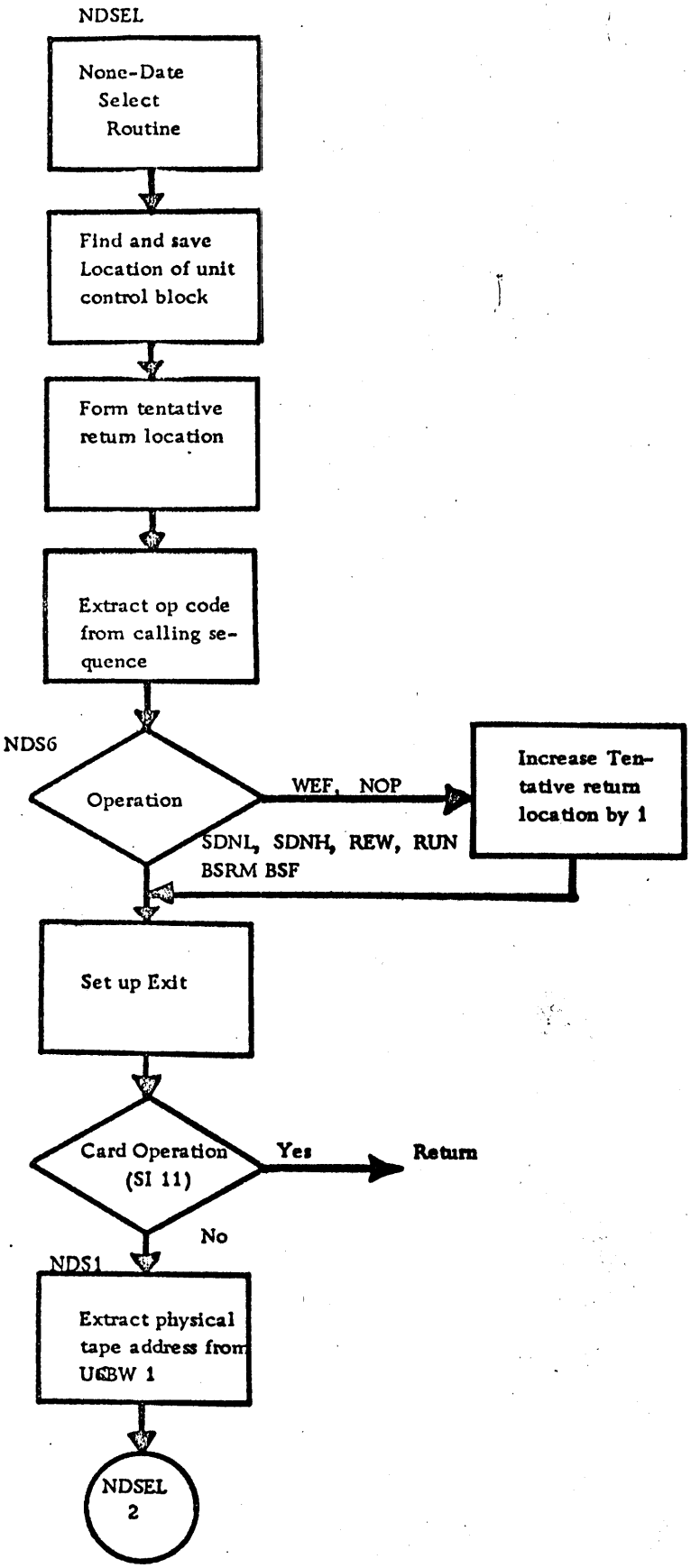

FRCHX
Free Channel
Extract Channel Number from UCBW1
FRC 6
This Channel previously freed
Yes
Return - TRA 2, 4
No
This Channel active
No
Yes
This Channel Trapped
Yes
No
Set force switch 1 (FRSW 1)
Enable CW-EOF Trap for this channel
Wait for Trap to Occur
TRAP
FRC1
Reset Force Switch 1 (FRSW1)
FRCHX 2

FRCHX 1

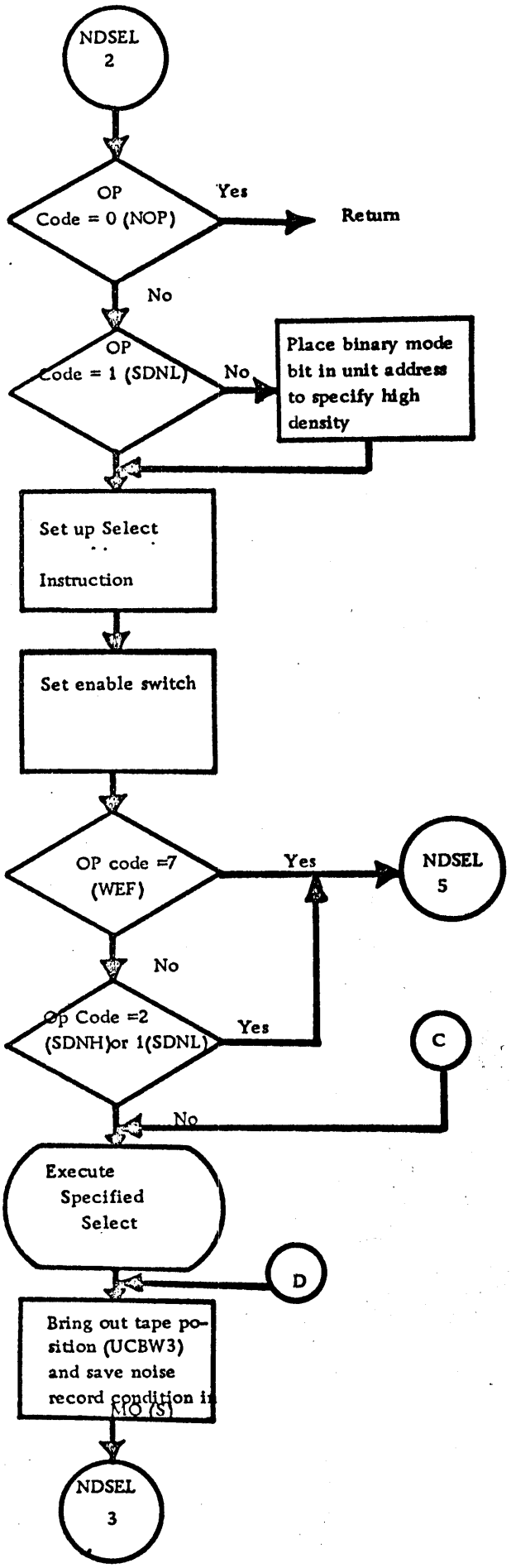


Return - TRA 2, 4

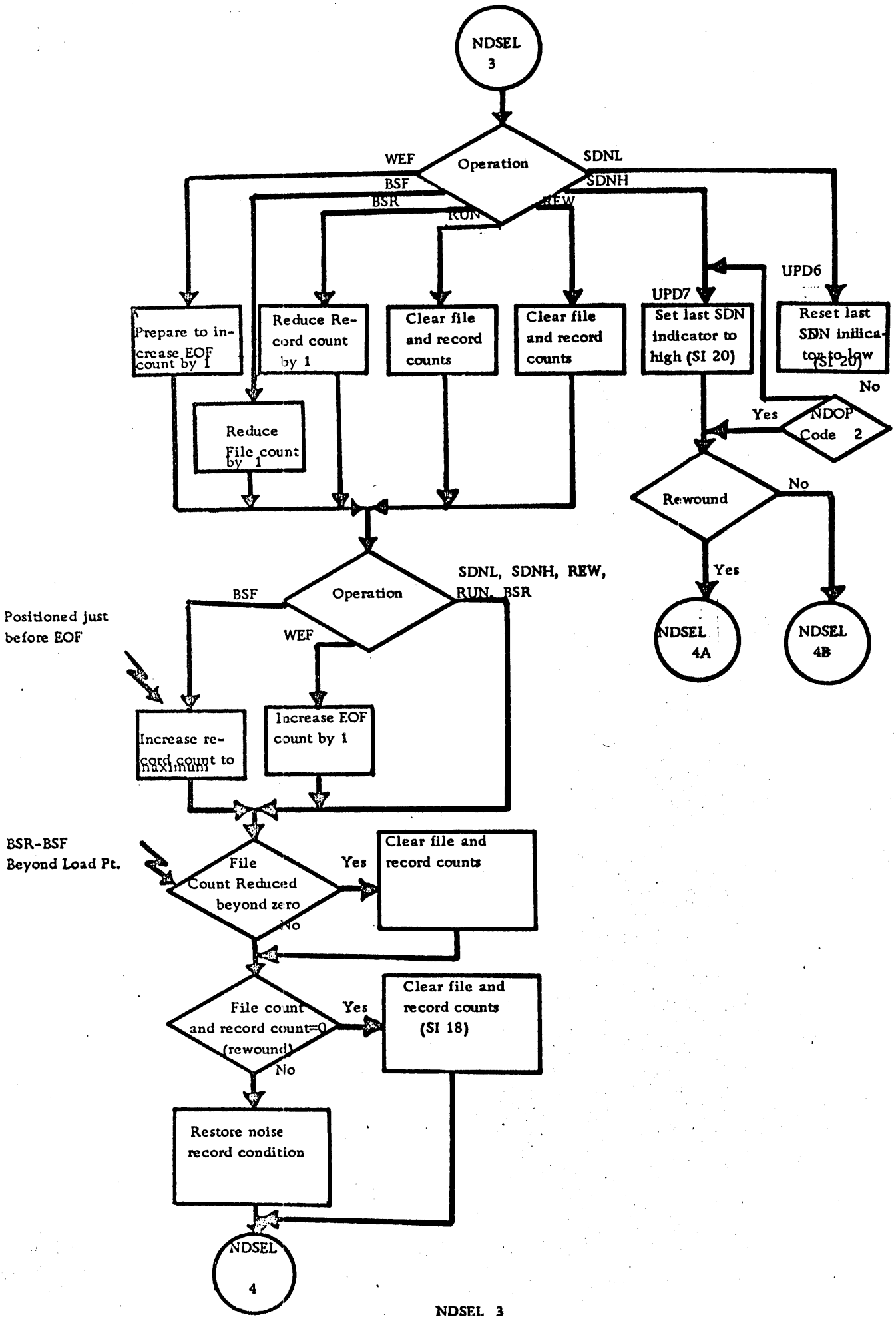

FRCHX
3
Complete Forced Trap Procedures
DID
Forced Trap occur on
this channel
No
Prepare to check
next channel
All
Channels checked
No
Yes
FRC5
Yes
Reset Force
Switch 2
Out to Point interrup
by main trap
Prepare to complete
interrupted trap
Delete Option
Yes
Enter
Trap
Mode
No
XTRAP


NDSEL
None-Date Select Routine
Find and save Location of unit control block
Form tentative return location
Extract op code from calling sequence
NDS6
Operation
WEF, NOP
SDNL, SDNH, REW, RUN BSRM BSF
Increase Tentative return location by 1
Set up Exit
Card Operation (SI 11)
Yes
Return
No
NDS1
Extract physical tape address from UCBW 1
NDSEL 2

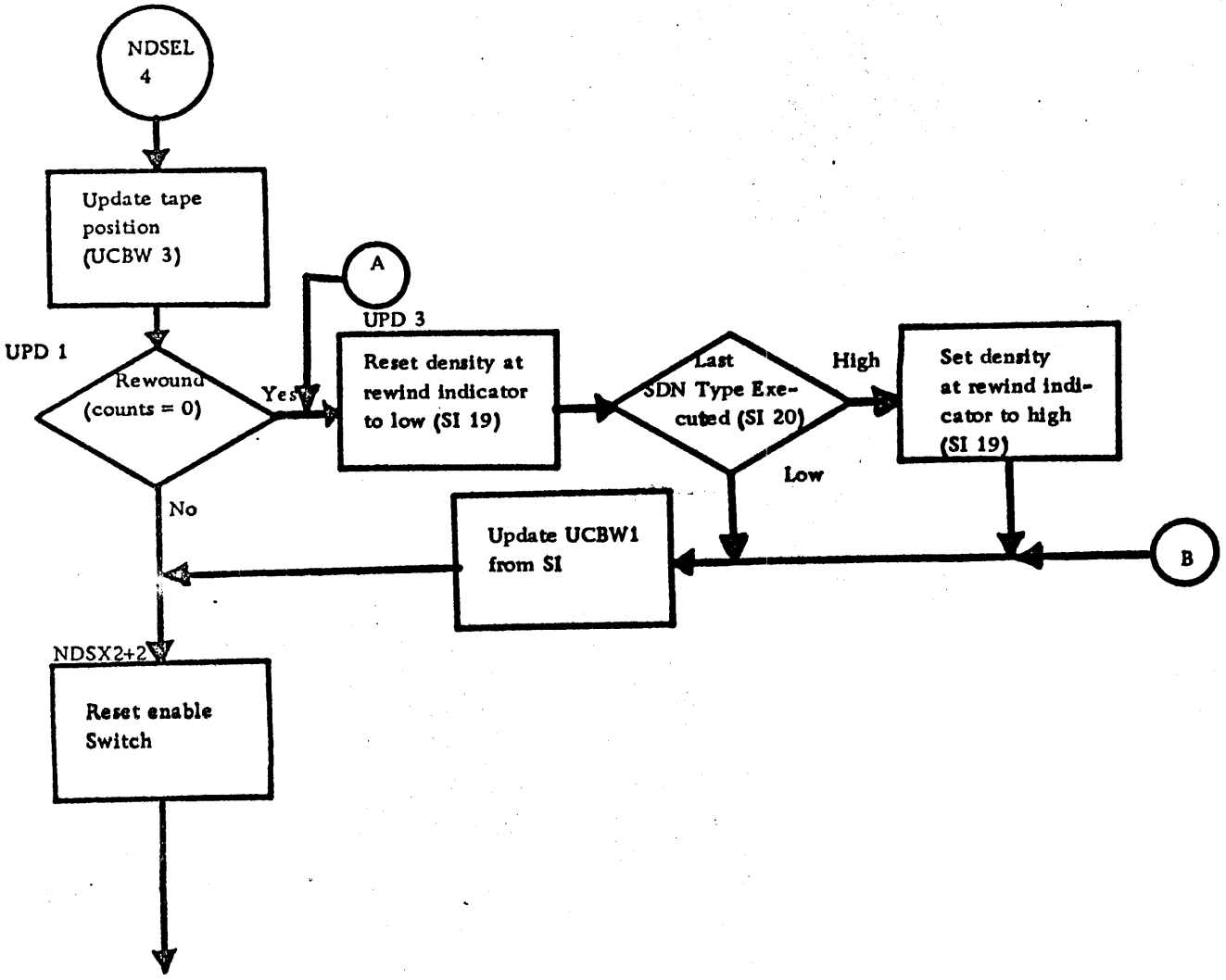
NDSEL 1


NDSEL 2
OP Code = 0 (NOP)
Yes
Return
No
OP Code = 1 (SDNL)
No
Place binary mode bit in unit address to specify high density
Set up Select Instruction
Set enable switch
OP code =7 (WEF)
Yes
NDSEL 5
No
Op Code =2 (SDNH)or 1(SDNL)
Yes
C
No
Execute Specified Select
D
Bring out tape position (UCBW3) and save noise record condition in MQ (S)
NDSEL 3

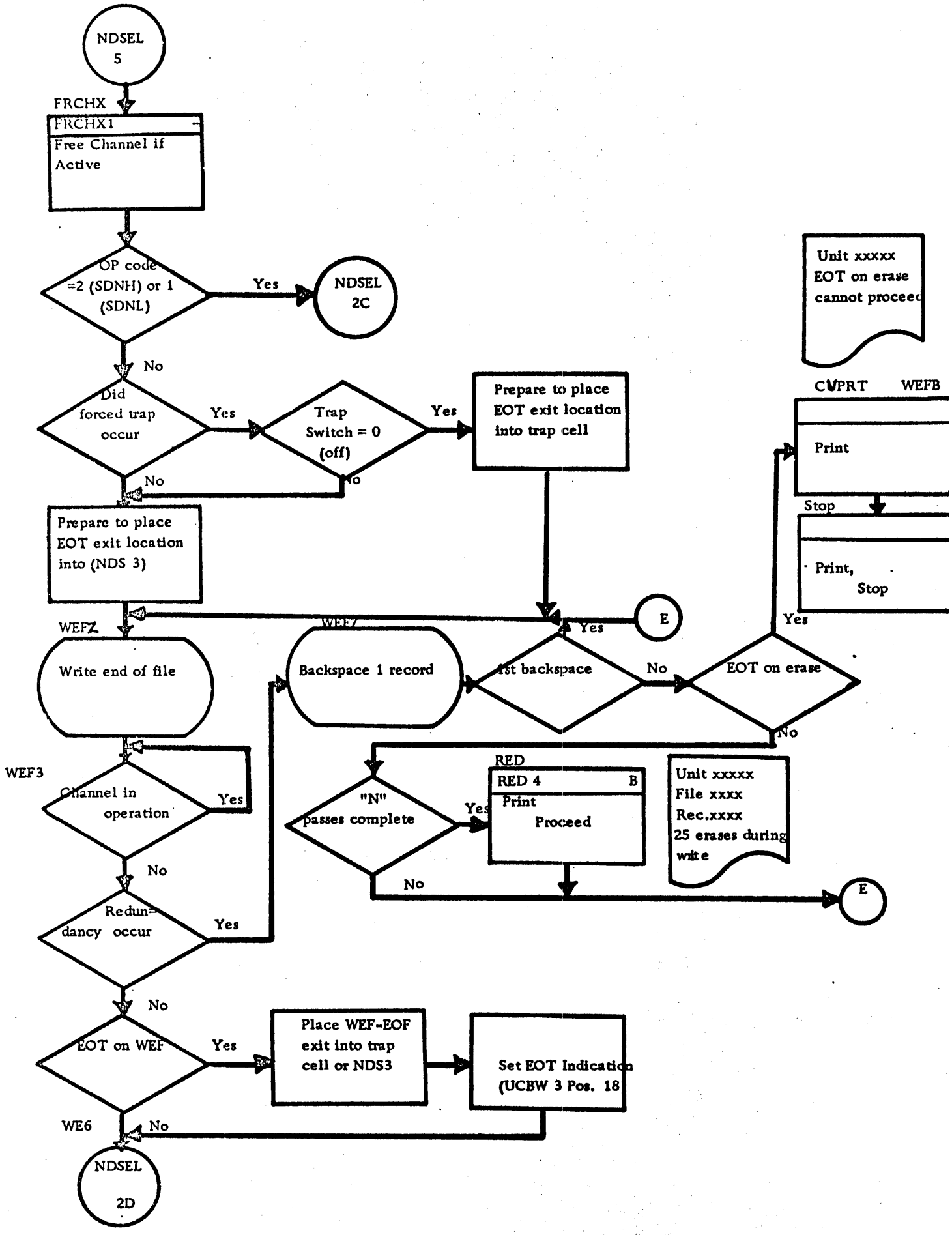
NDSEL 2


NDSEL 3
Operation
WEF
BSF
BSR
RUN
REW
SDNL
SDNH
Prepare to increase EOF count by 1
Reduce Record count by 1
Clear file and record counts
Clear file and record counts
UPD7
Set last SDN indicator to high (SI 20)
UPD6
Reset last SDN indicator to low (SI 20)
No
Yes
NDOP Code 2
Reduce File count by 1
Rewound
No
Yes
NDSEL 4A
NDSEL 4B
SDNL, SDNH, REW, RUN, BSR
Operation
BSF
WEF
Positioned just before EOF
Increase record count to maximum
Increase EOF count by 1
BSR-BSF Beyond Load Pt.
File Count Reduced beyond zero
Yes
No
Clear file and record counts
File count and record count=0 (rewound)
Yes
No
Clear file and record counts (SI 18)
Restore noise record condition
NDSEL 4

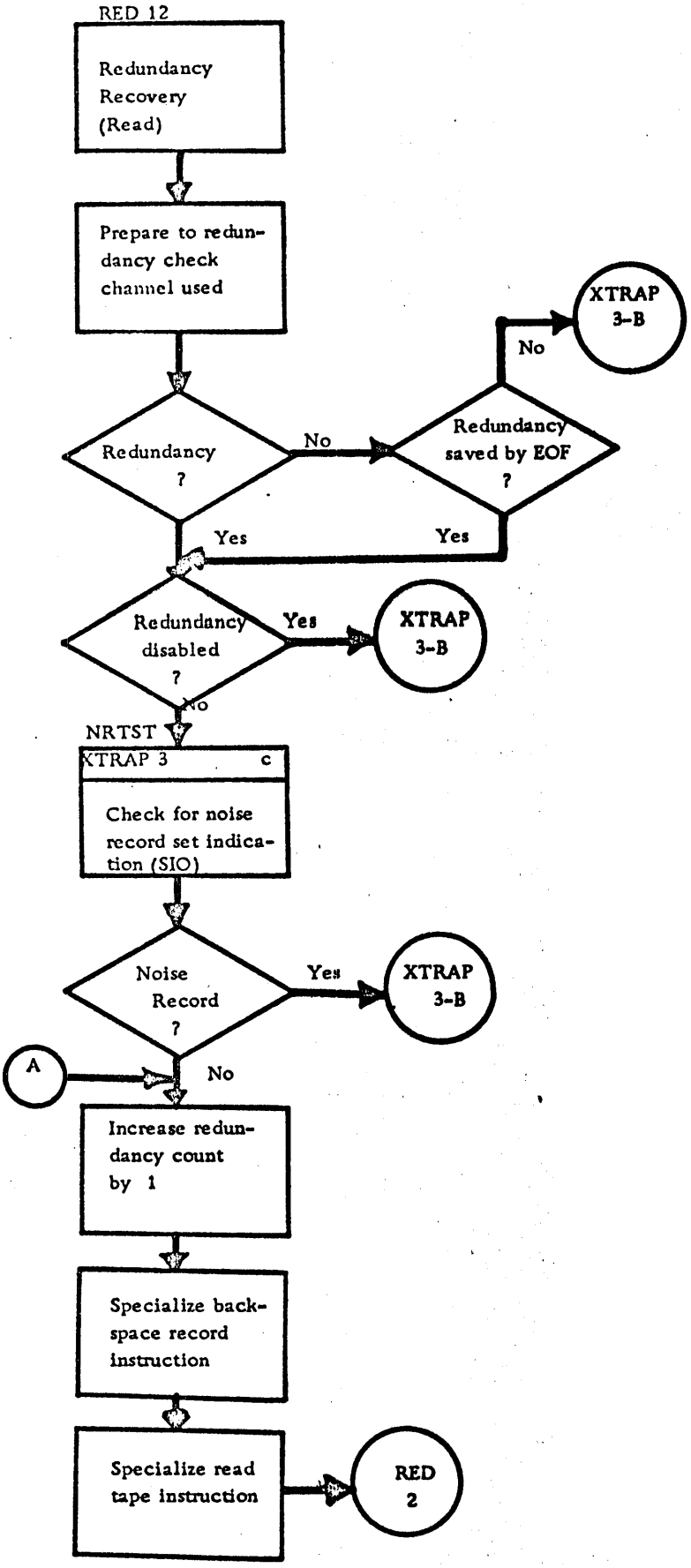
NDSEL 3



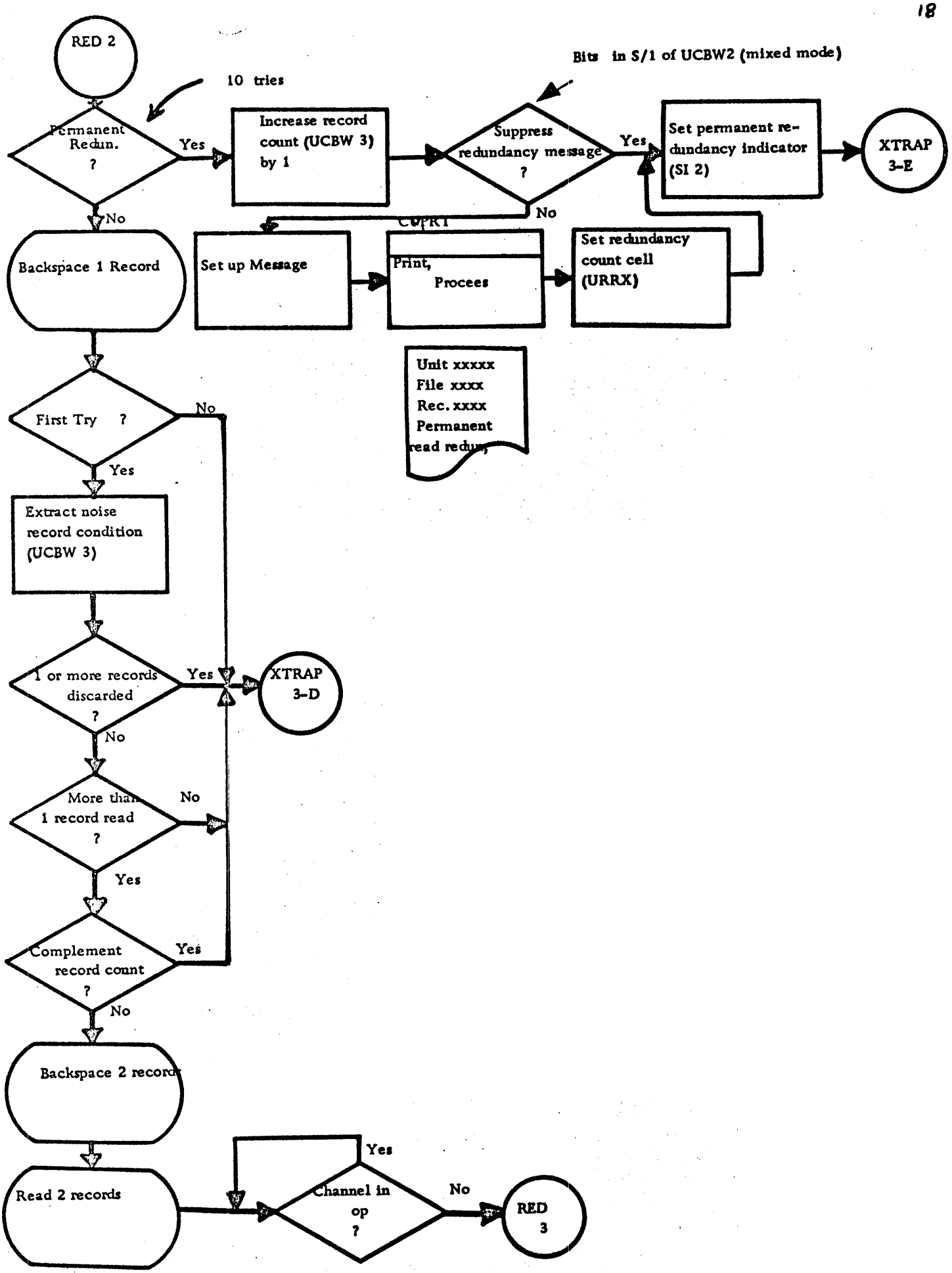
Return to main routine or forced trap completion (see FRCHX 3)


NDSEL 5
FRCHX
FRCHX1
Free Channel if Active
OP code =2 (SDNH) or 1 (SDNL)
Yes
NDSEL 2C
No
Did forced trap occur
Yes
Trap Switch = 0 (off)
Yes
No
No
Prepare to place EOT exit location into trap cell
Prepare to place EOT exit location into (NDS 3)
E
WEFZ
Write end of file
WEF2
Backspace 1 record
1st backspace
No
EOT on erase
Yes
No
Yes
Unit xxxxx EOT on erase cannot proceed
CWPRT
WEFB
Print
Stop
Print, Stop
WEF3
Channel in operation
Yes
No
Redun= dancy occur
Yes
No
RED
"N" passes complete
Yes
RED 4 B
Print
Proceed
No
Unit xxxxx
File xxxx
Rec.xxxx
25 erases during write
E
EOT on WEF
Yes
Place WEF-EOF exit into trap cell or NDS3
Set EOT Indication (UCBW 3 Pos. 18
WE6
No
NDSEL 2D

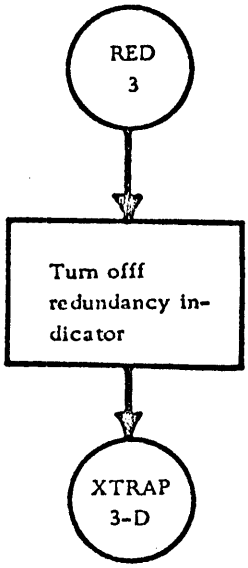
NDSEL 5

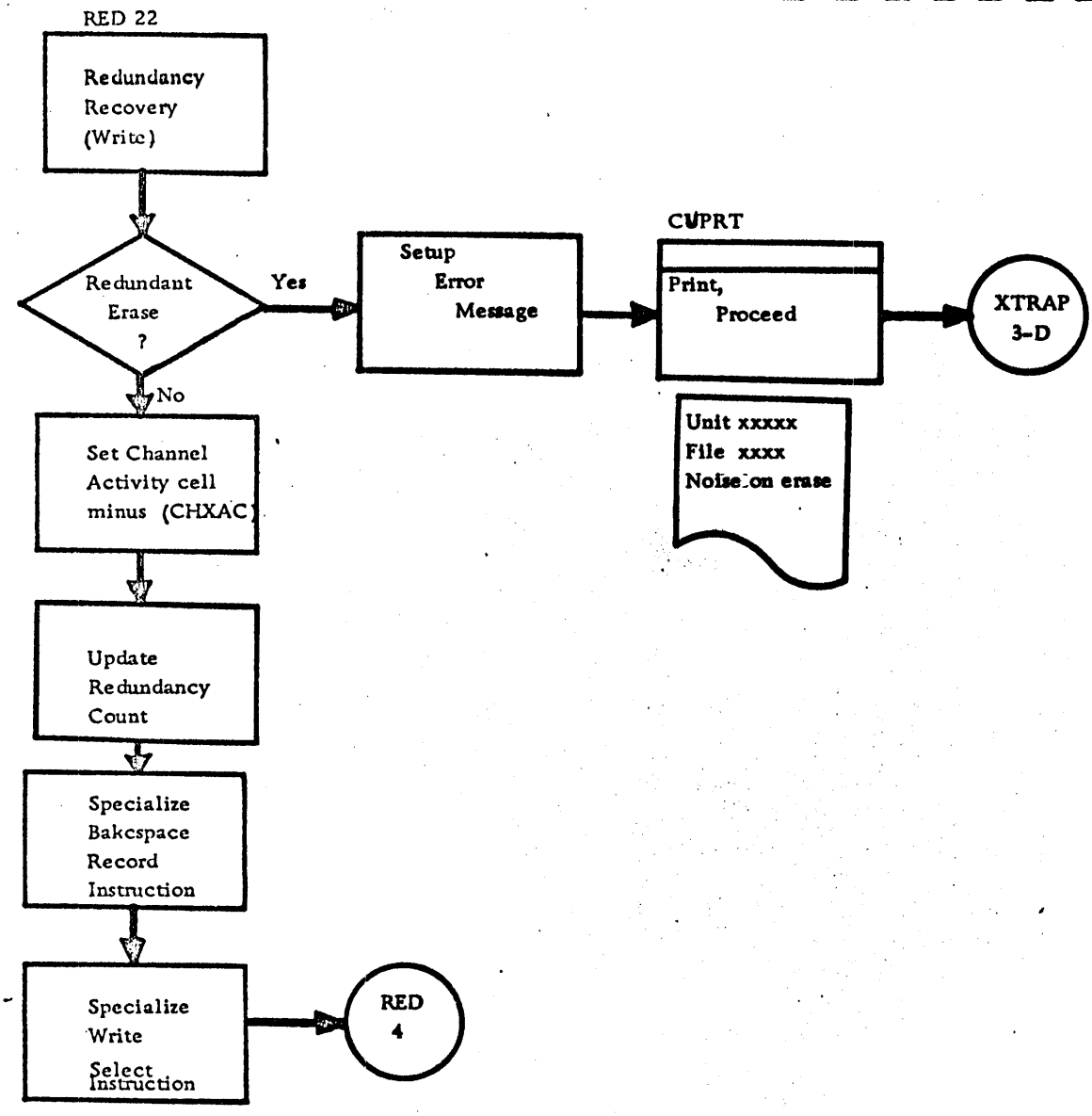

RED 12
Redundancy Recovery (Read)
Prepare to redundancy check channel used
Redundancy ?
No
Redundancy saved by EOF ?
No
XTRAP 3-B
Yes
Yes
Redundancy disabled ?
Yes
XTRAP 3-B
No
NRTST
XTRAP 3 c
Check for noise record set indication (SIO)
Noise Record ?
Yes
XTRAP 3-B
A
No
Increase redundancy count by 1
Specialize backspace record instruction
Specialize read tape instruction
RED 2

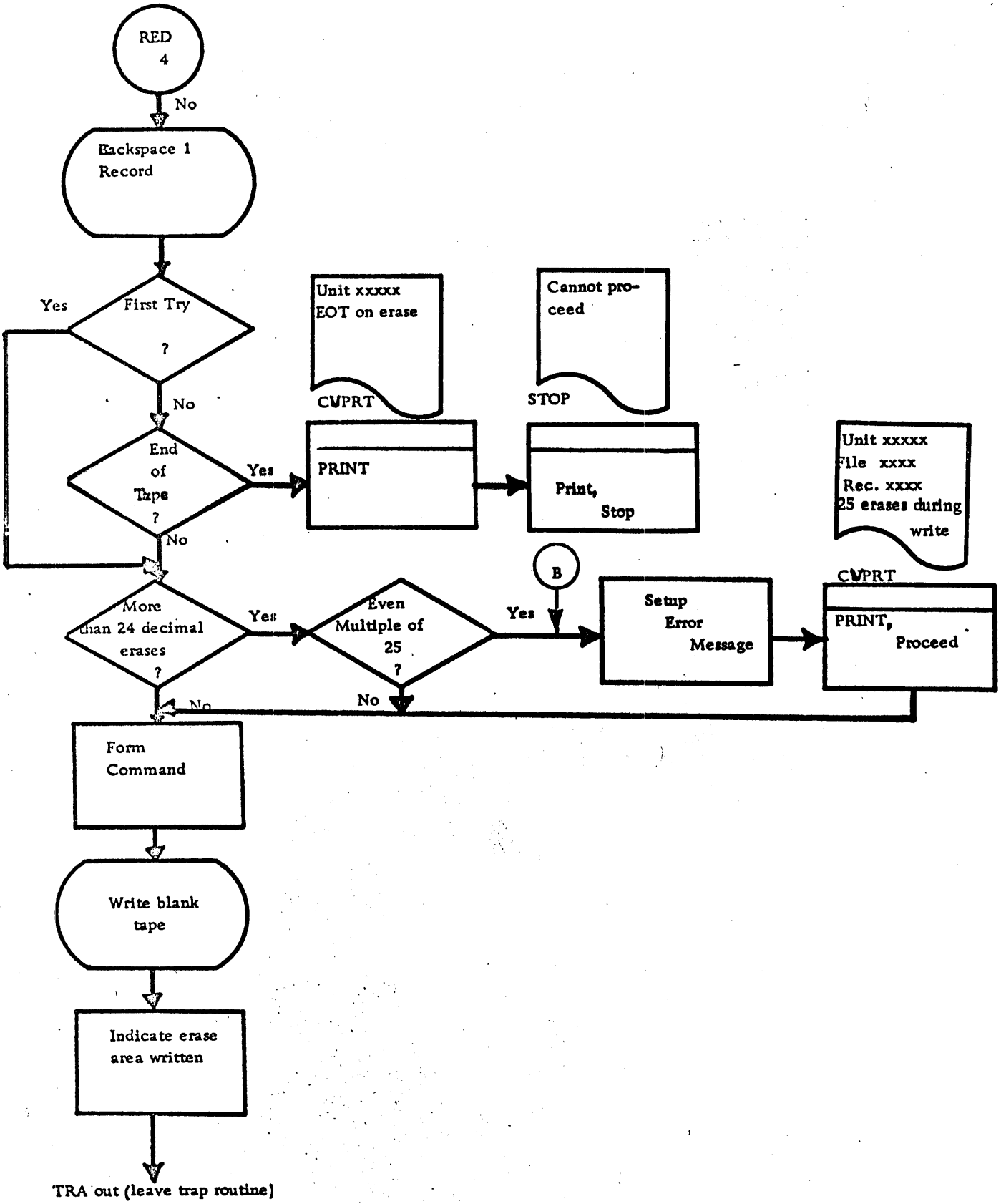
RED 1

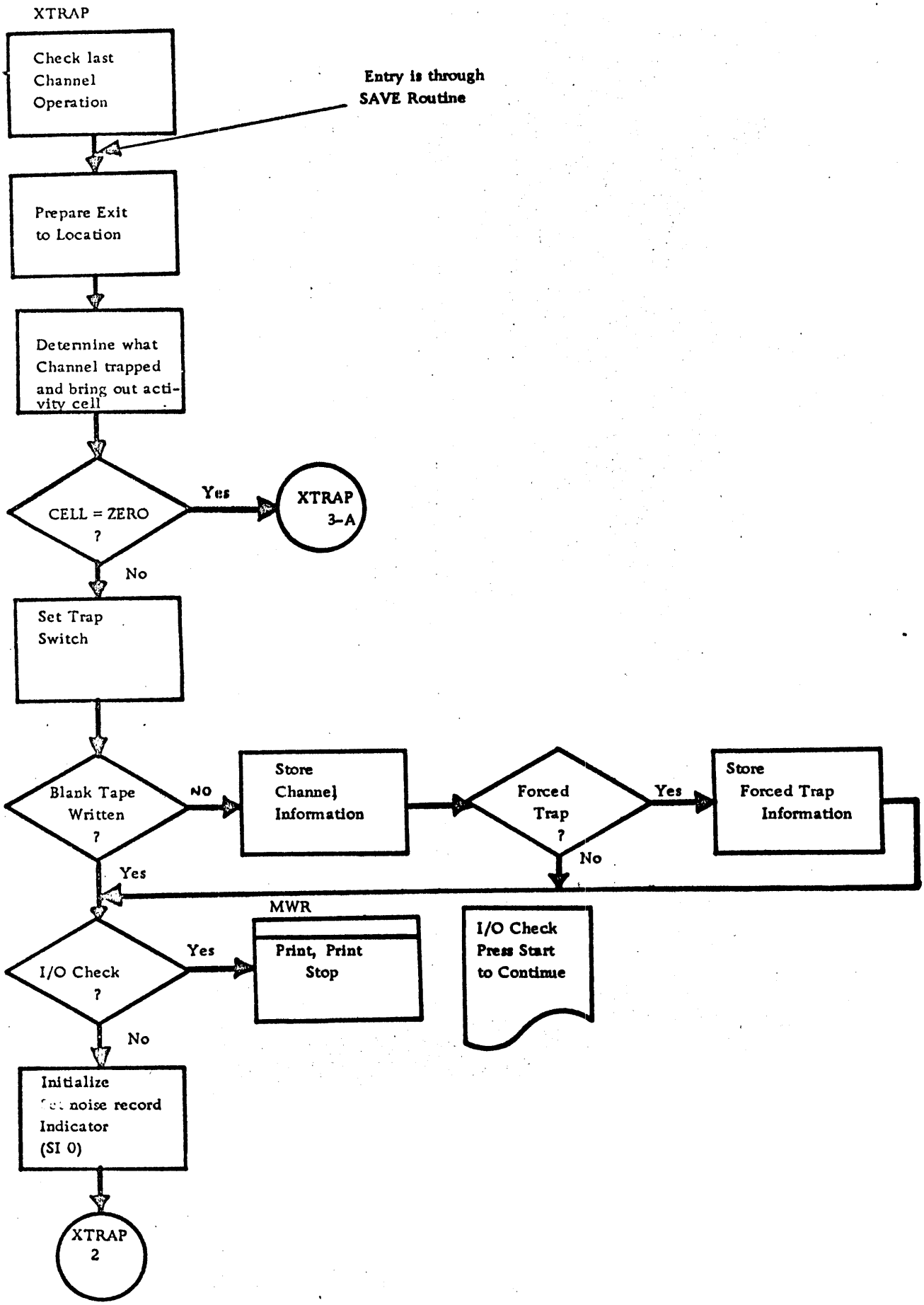

RED 2
10 tries
Bits in S/1 of UCBW2 (mixed mode)
Permanent Redun. ?
Yes
Increase record count (UCBW 3) by 1
Suppress redundancy message ?
Yes
Set permanent redundancy indicator (SI 2)
XTRAP 3-E
No
No
Backspace 1 Record
Set up Message
CWPRT
Print, Procees
Set redundancy count cell (URRX)
Unit xxxxx
File xxxx
Rec. xxxx
Permanent
First Try ?
No
Yes
Extract noise record condition (UCBW 3)
1 or more records discarded ?
Yes
XTRAP 3-D
No
More than 1 record read ?
No
Yes
Complement record count ?
Yes
No
Backspace 2 records
Read 2 records
Yes
Channel in op ?
No
RED 3

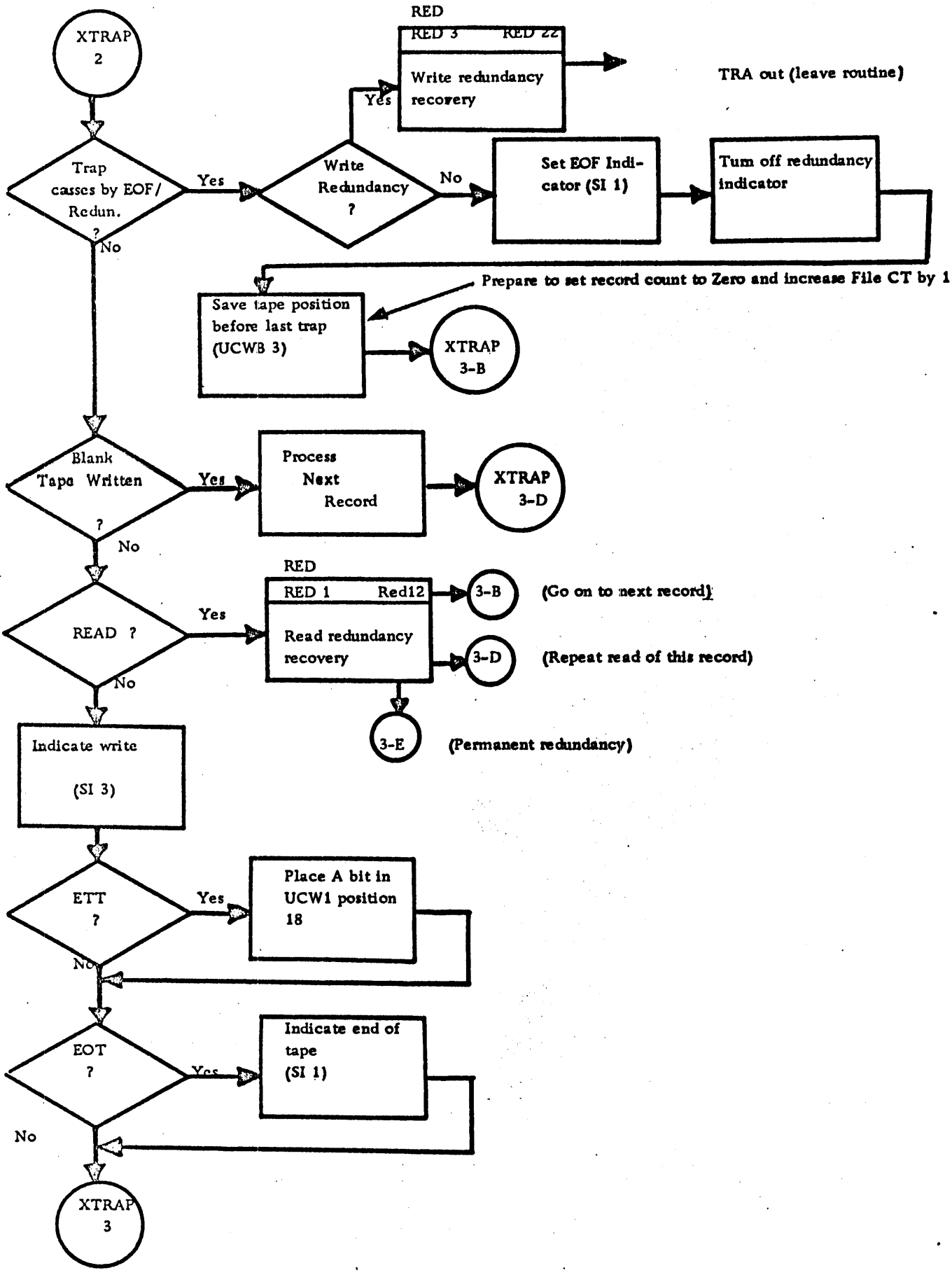
RED 2

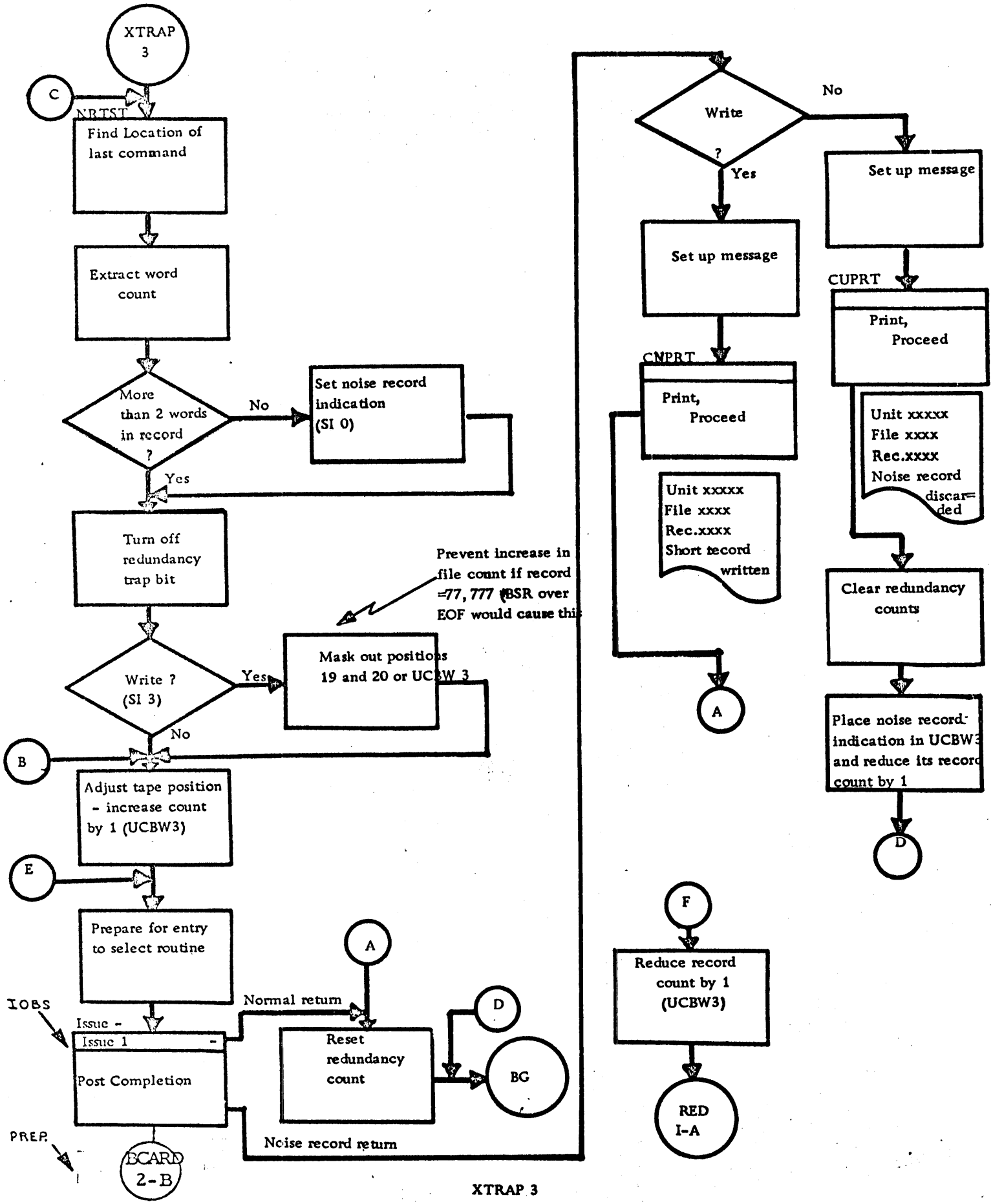

RED
3
Turn offf
redundancy in-
dicator
XTRAP
3-D


RED 22
Redundancy
Recovery
(Write)
Redundant
Erase
?
Yes
No
Setup
Error
Message
CUPRT
Print,
Proceed
XTRAP
3-D
Unit xxxxx
File xxxx
Noise on erase
Set Channel
Activity cell
minus (CHXAC)
Update
Redundancy
Count
Specialize
Bakcspace
Record
Instruction
Specialize
Write
Select
Instruction
RED
4

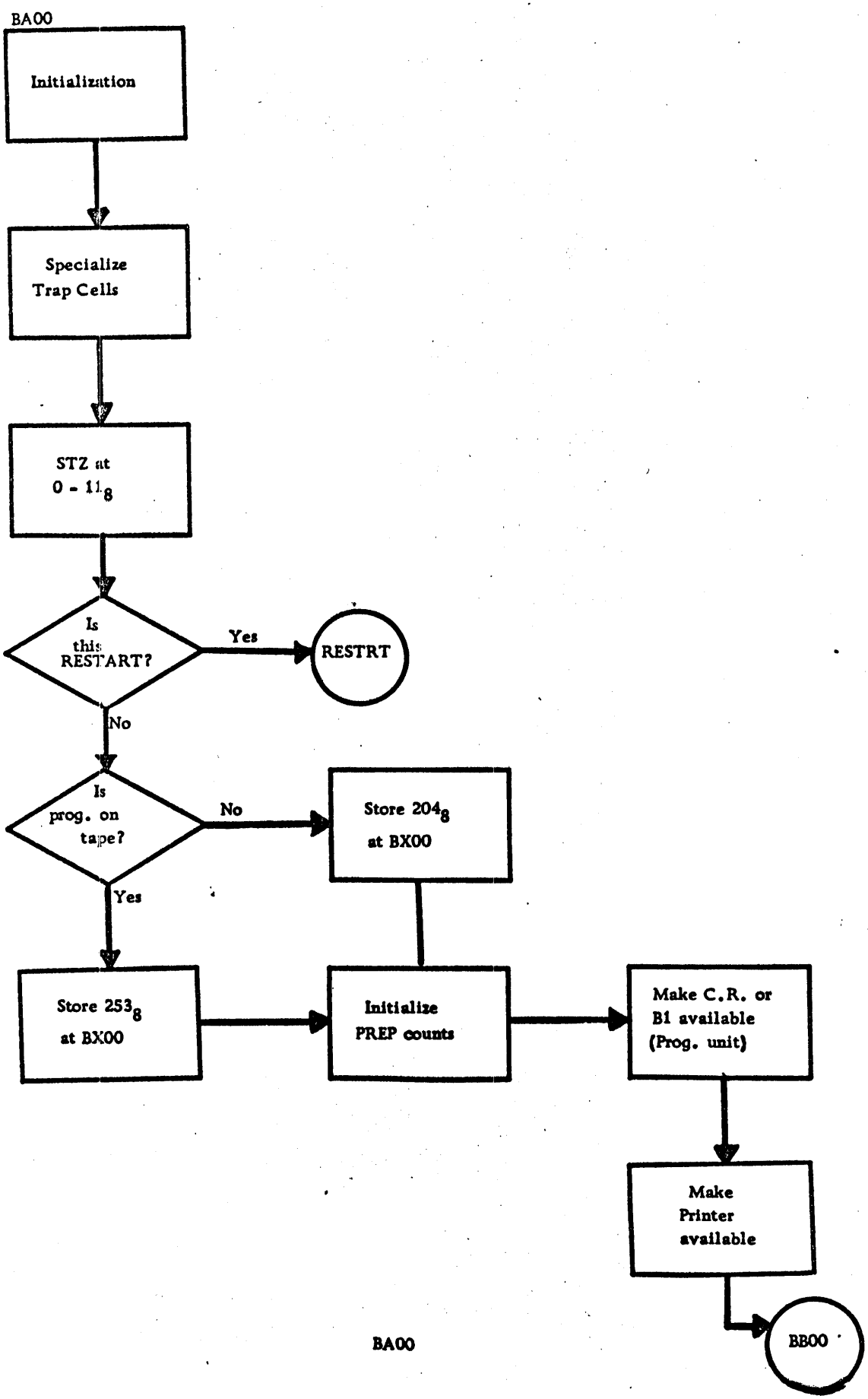

RED 4
No
Backspace 1 Record
First Try ?
Yes
No
Unit xxxxx EOT on erase
CWPRT
Cannot proceed
STOP
End of Tape ?
Yes
PRINT
Print, Stop
Unit xxxxx File xxxx Rec. xxxx 25 erases during write
CWPRT
No
More than 24 decimal erases ?
Yes
Even Multiple of 25 ?
Yes
B
Setup Error Message
PRINT, Proceed
No
No
Form Command
Write blank tape
Indicate erase area written
TRA out (leave trap routine)


XTRAP
Check last Channel Operation
Entry is through SAVE Routine
Prepare Exit to Location
Determine what Channel trapped and bring out activity cell
CELL = ZERO ?
Yes
XTRAP 3-A
No
Set Trap Switch
Blank Tape Written ?
NO
Store Channel Information
Forced Trap ?
Yes
Store Forced Trap Information
No
Yes
MWR
I/O Check ?
Yes
Print, Print Stop
I/O Check Press Start to Continue
No
Initialize [illegible] noise record Indicator (SI 0)
XTRAP 2

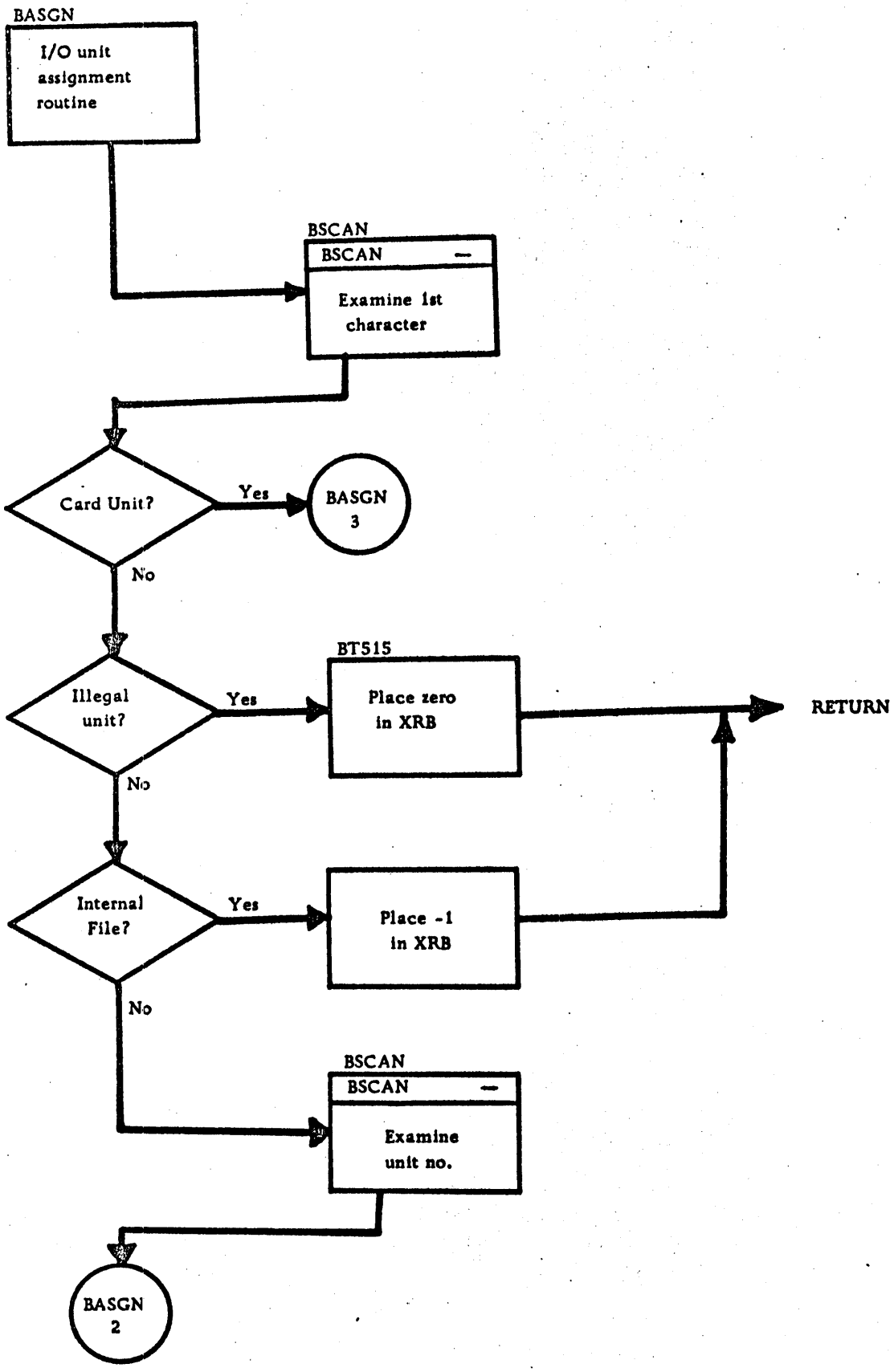

XTRAP 2
Trap cases by EOF/ Redun. ?
Yes
No
Write Redundancy ?
Yes
No
RED
RED 3 RED 22
Write redundancy recovery
TRA out (leave routine)
Set EOF Indicator (SI 1)
Turn off redundancy indicator
Prepare to set record count to Zero and increase File CT by 1
Save tape position before last trap (UCWB 3)
XTRAP 3-B
Blank Tape Written ?
Yes
No
Process Next Record
XTRAP 3-D
READ ?
Yes
No
RED
RED 1 Red12
Read redundancy recovery
3-B
(Go on to next record)
3-D
(Repeat read of this record)
3-E
(Permanent redundancy)
Indicate write (SI 3)
ETT ?
Yes
No
Place A bit in UCW1 position 18
EOT ?
Yes
No
Indicate end of tape (SI 1)
XTRAP 3


XTRAP 3
C
NRTST
Find Location of last command
Extract word count
More than 2 words in record ?
No
Set noise record indication (SI 0)
Yes
Turn off redundancy trap bit
Prevent increase in file count if record =77,777 (BSR over EOF would cause this
Write ? (SI 3)
Yes
Mask out positions 19 and 20 or UCBW 3
No
B
Adjust tape position - increase count by 1 (UCBW3)
E
Prepare for entry to select routine
IOBS
Issue -
Issue 1 -
Post Completion
PREP
BCARD 2-B
A
Normal return
Reset redundancy count
D
BG
Noise record return
Write ?
No
Yes
Set up message
CNPRT
Print, Proceed
Unit xxxxx
File xxxx
Rec.xxxx
Short record written
A
Set up message
CUPRT
Print, Proceed
Unit xxxxx
File xxxx
Rec.xxxx
Noise record discarded
Clear redundancy counts
Place noise record indication in UCBW3 and reduce its record count by 1
D
F
Reduce record count by 1 (UCBW3)
RED I-A

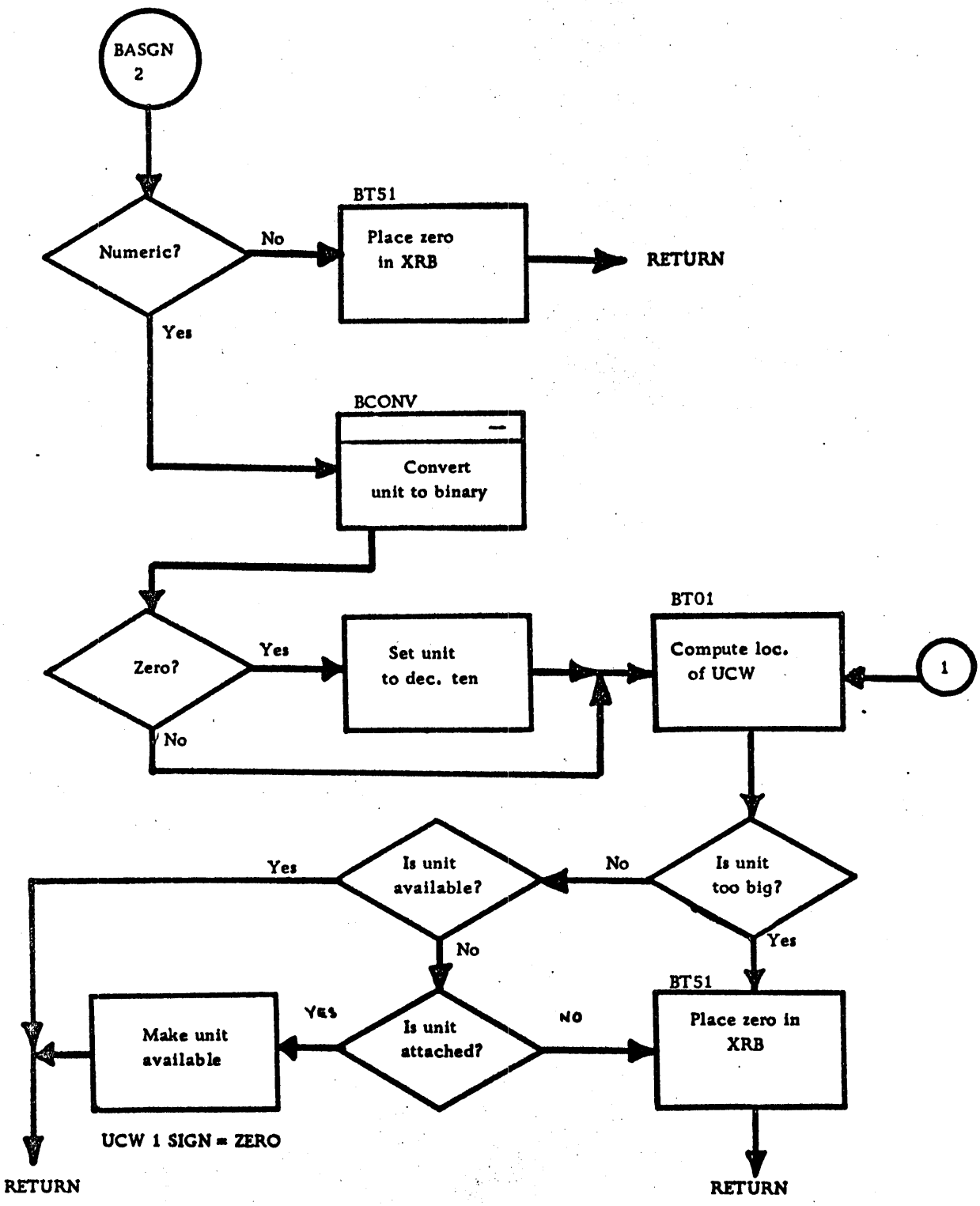
XTRAP 3


BA00
Initialization
Specialize Trap Cells
STZ at 0 - 11$_8$
Is this RESTART?
Yes
RESTRT
No
Is prog. on tape?
No
Store 204$_8$ at BX00
Yes
Store 253$_8$ at BX00
Initialize PREP counts
Make C.R. or B1 available (Prog. unit)
Make Printer available
BB00

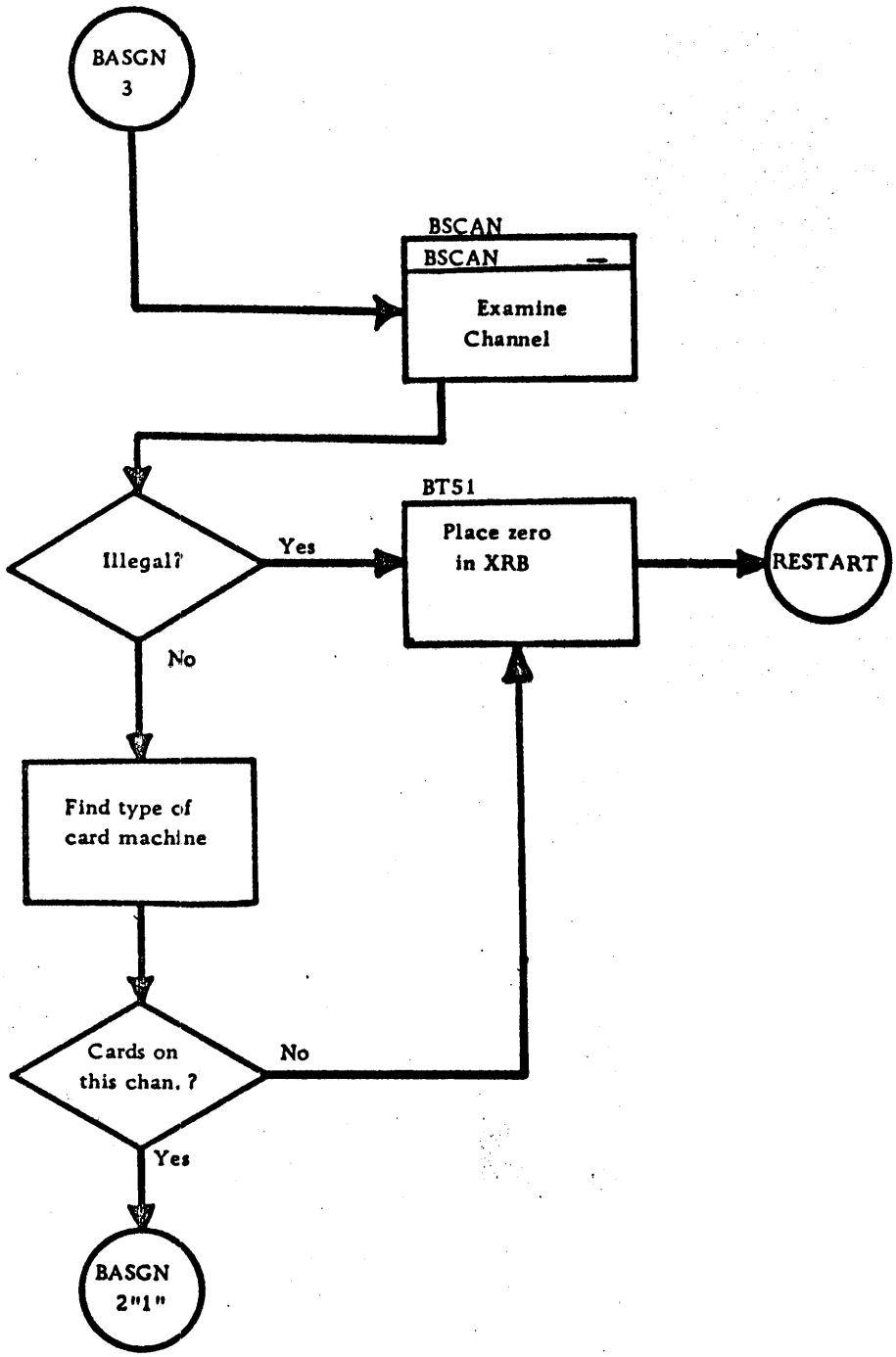
BA00


BASGN
I/O unit assignment routine
BSCAN
BSCAN —
Examine 1st character
Card Unit?
Yes
BASGN 3
No
BT515
Illegal unit?
Yes
Place zero in XRB
RETURN
No
Internal File?
Yes
Place -1 in XRB
No
BSCAN
BSCAN —
Examine unit no.
BASGN 2

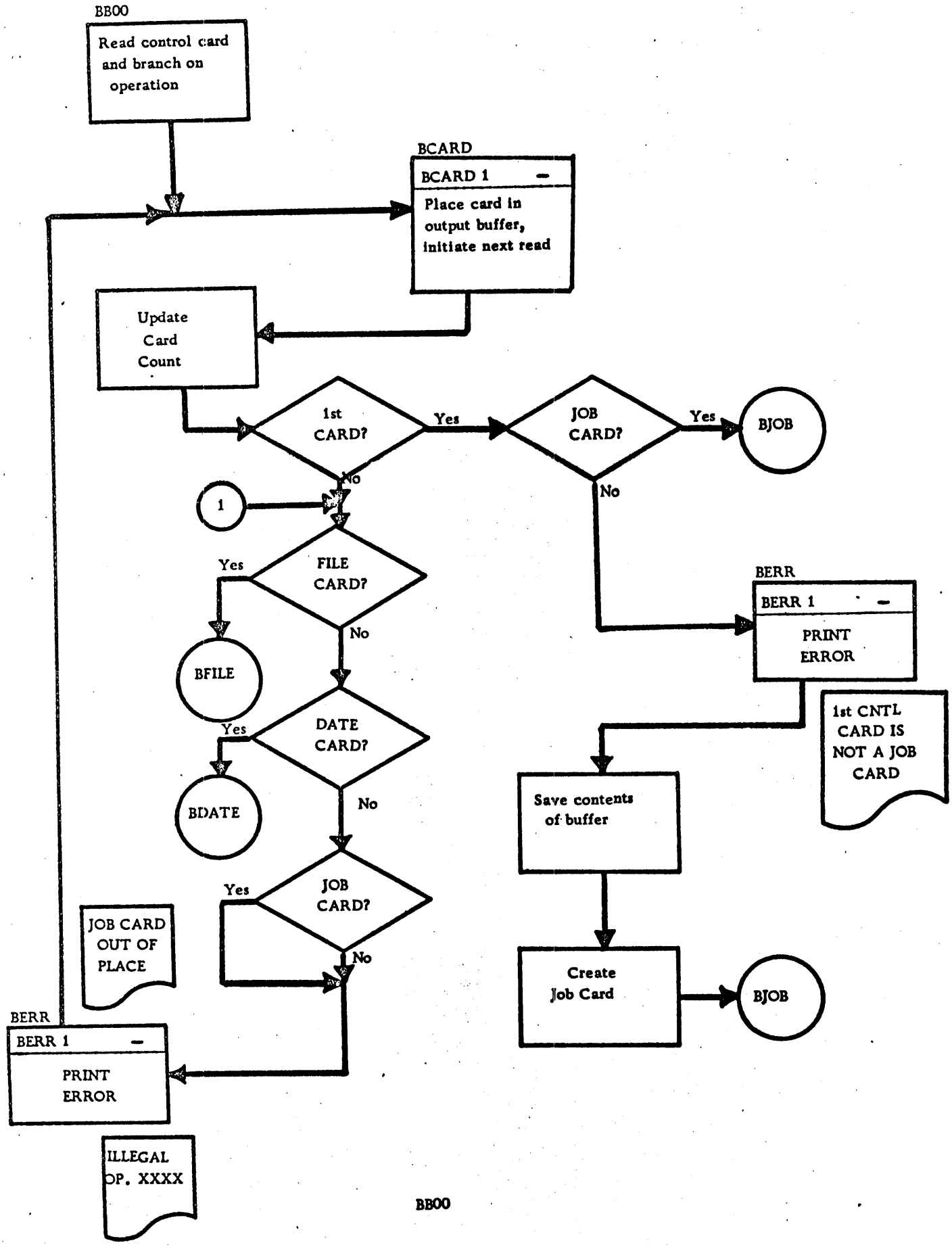
BASGN 1

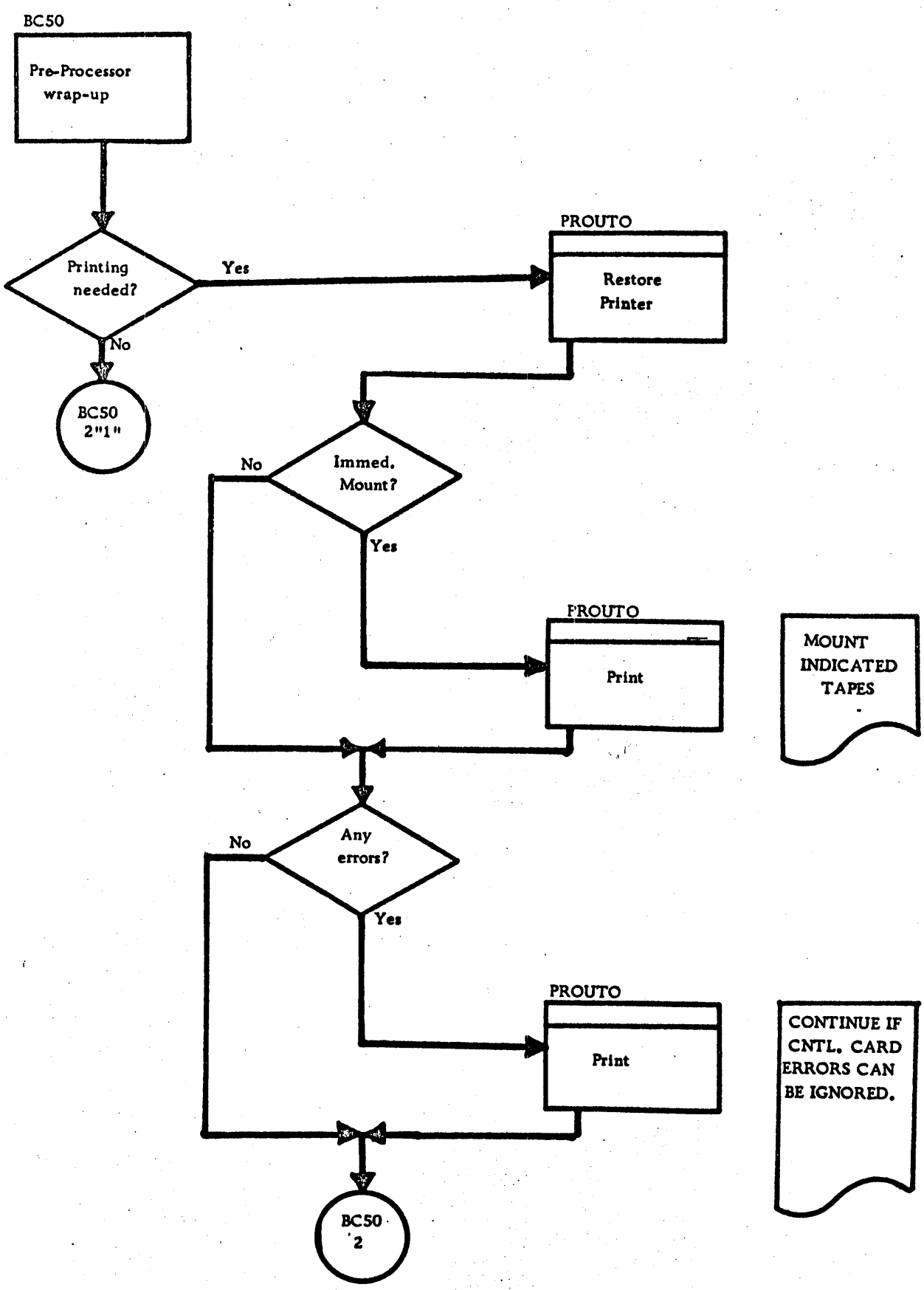

BASGN
2
Numeric?
No
BT51
Place zero
in XRB
RETURN
Yes
BCONV
Convert
unit to binary
Zero?
Yes
Set unit
to dec. ten
BT01
Compute loc.
of UCW
1
No
Is unit
too big?
No
Is unit
available?
Yes
Yes
No
BT51
Place zero in
XRB
NO
Is unit
attached?
YES
Make unit
available
UCW 1 SIGN = ZERO
RETURN
RETURN

BASGN 2

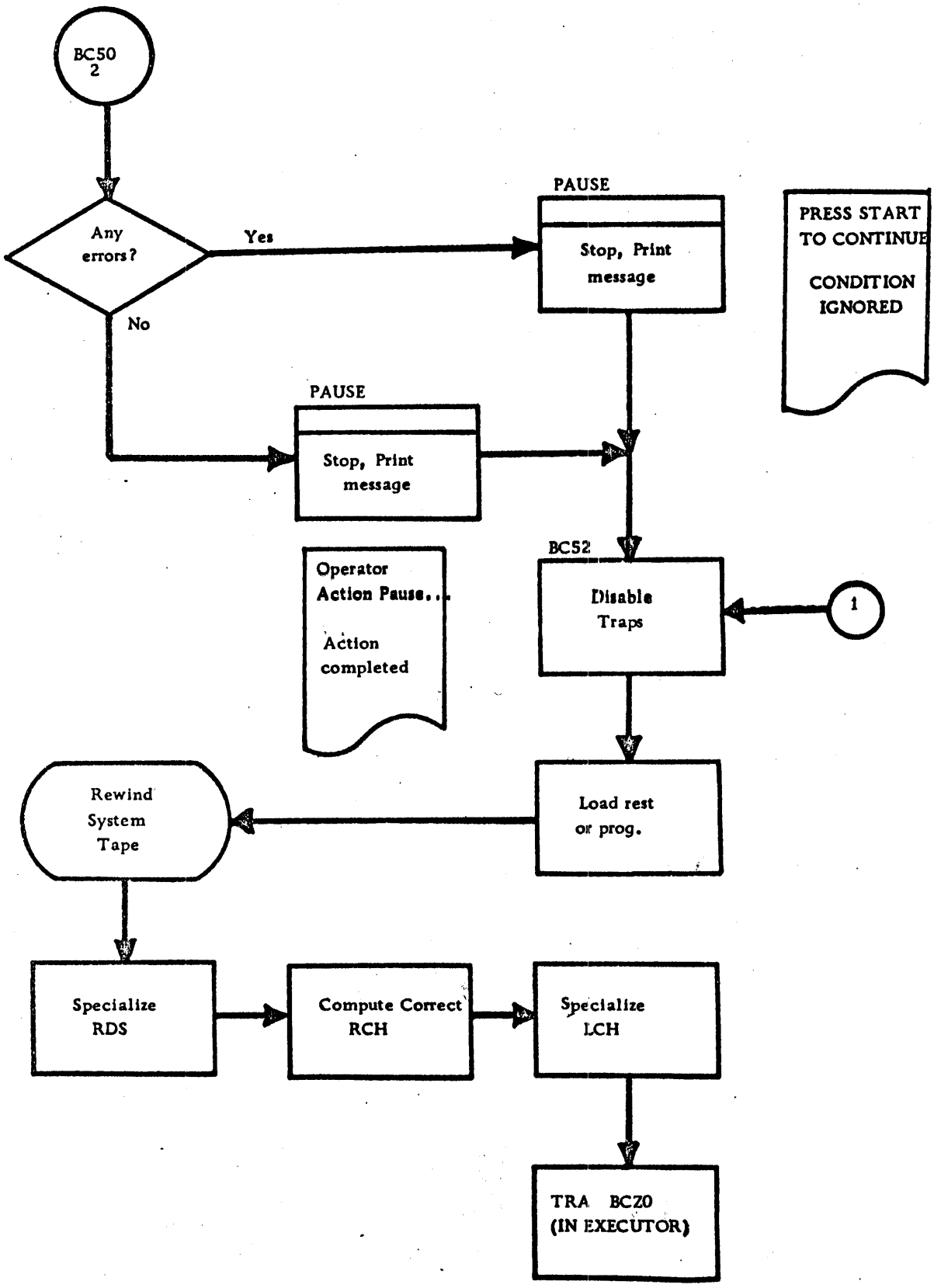

BASGN
3
BSCAN
BSCAN
Examine
Channel
Illegal?
Yes
BT51
Place zero
in XRB
RESTART
No
Find type of
card machine
Cards on
this chan. ?
No
Yes
BASGN
2"1"

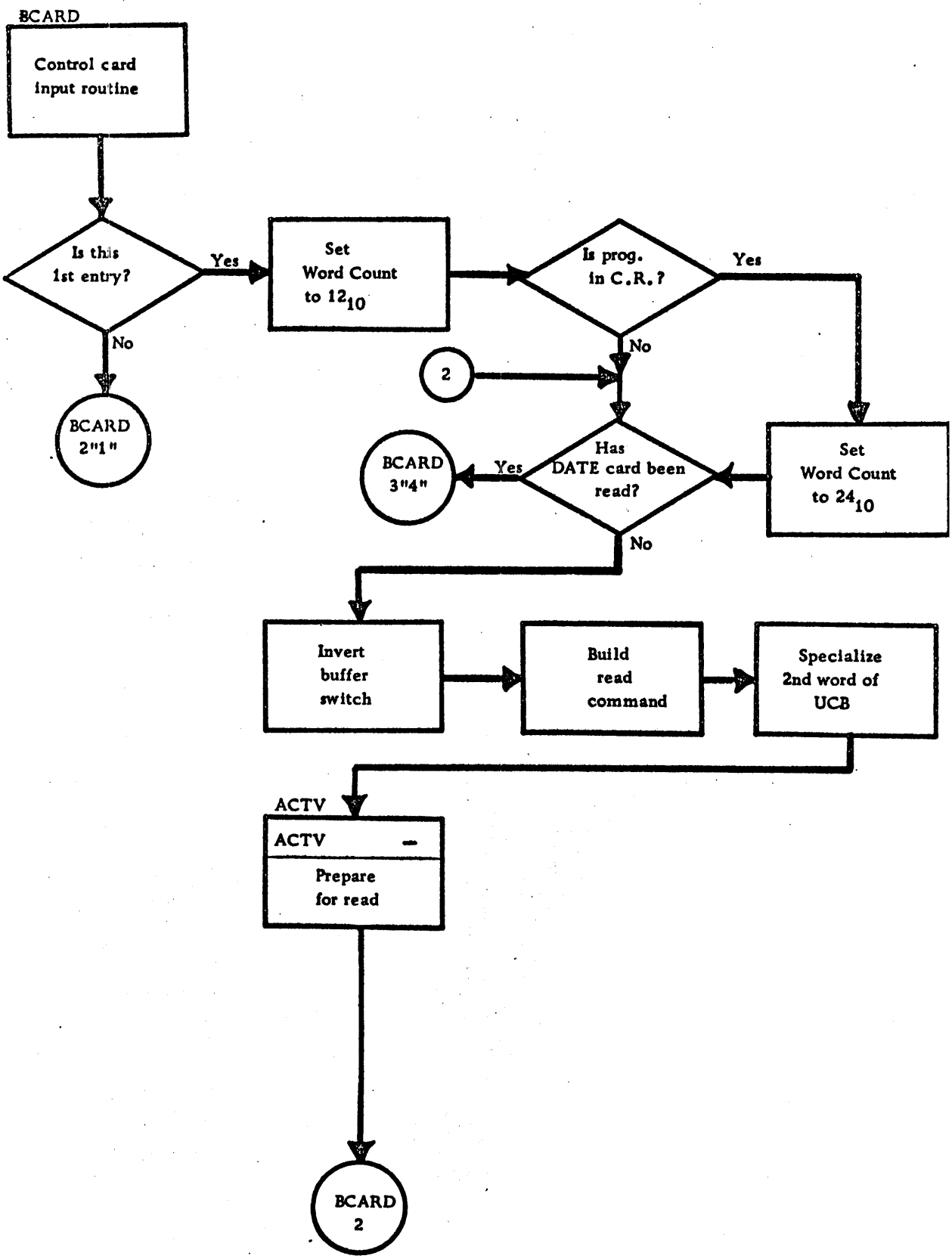
BASGN 3


BB00
Read control card and branch on operation
BCARD
BCARD 1 –
Place card in output buffer, initiate next read
Update Card Count
1st CARD?
Yes
JOB CARD?
Yes
BJOB
No
1
No
Yes
FILE CARD?
BFILE
No
Yes
DATE CARD?
BDATE
No
Yes
JOB CARD?
No
JOB CARD OUT OF PLACE
BERR
BERR 1 –
PRINT ERROR
ILLEGAL OP. XXXX
BERR
BERR 1 –
PRINT ERROR
1st CNTL CARD IS NOT A JOB CARD
Save contents of buffer
Create Job Card
BJOB

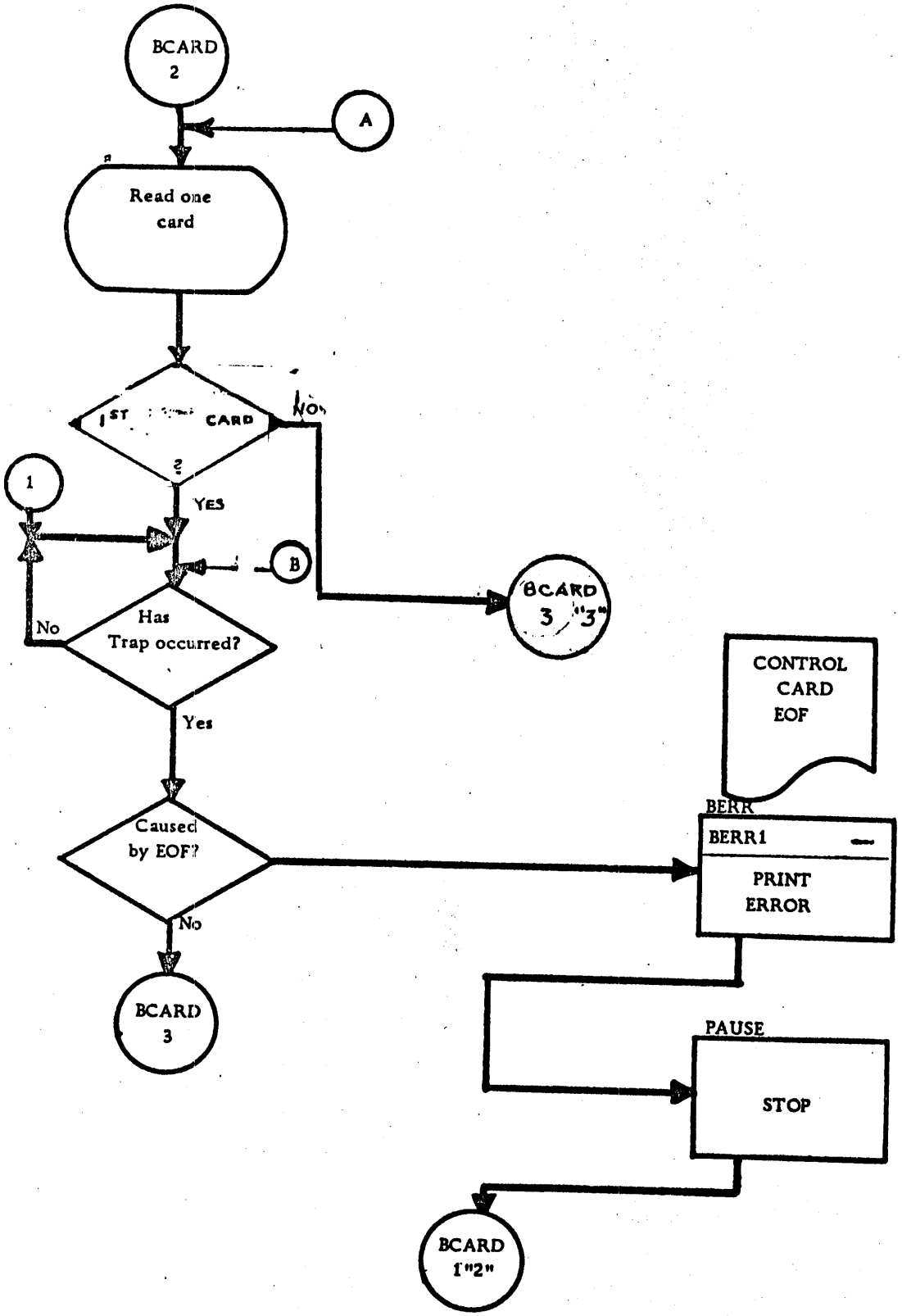

BC50
Pre-Processor wrap-up
Printing needed?
Yes
No
PROUTO
Restore Printer
BC50 2"1"
No
Immed. Mount?
Yes
PROUTO
Print
MOUNT INDICATED TAPES
No
Any errors?
Yes
PROUTO
Print
CONTINUE IF CNTL. CARD ERRORS CAN BE IGNORED.
BC50 2

BC50 1

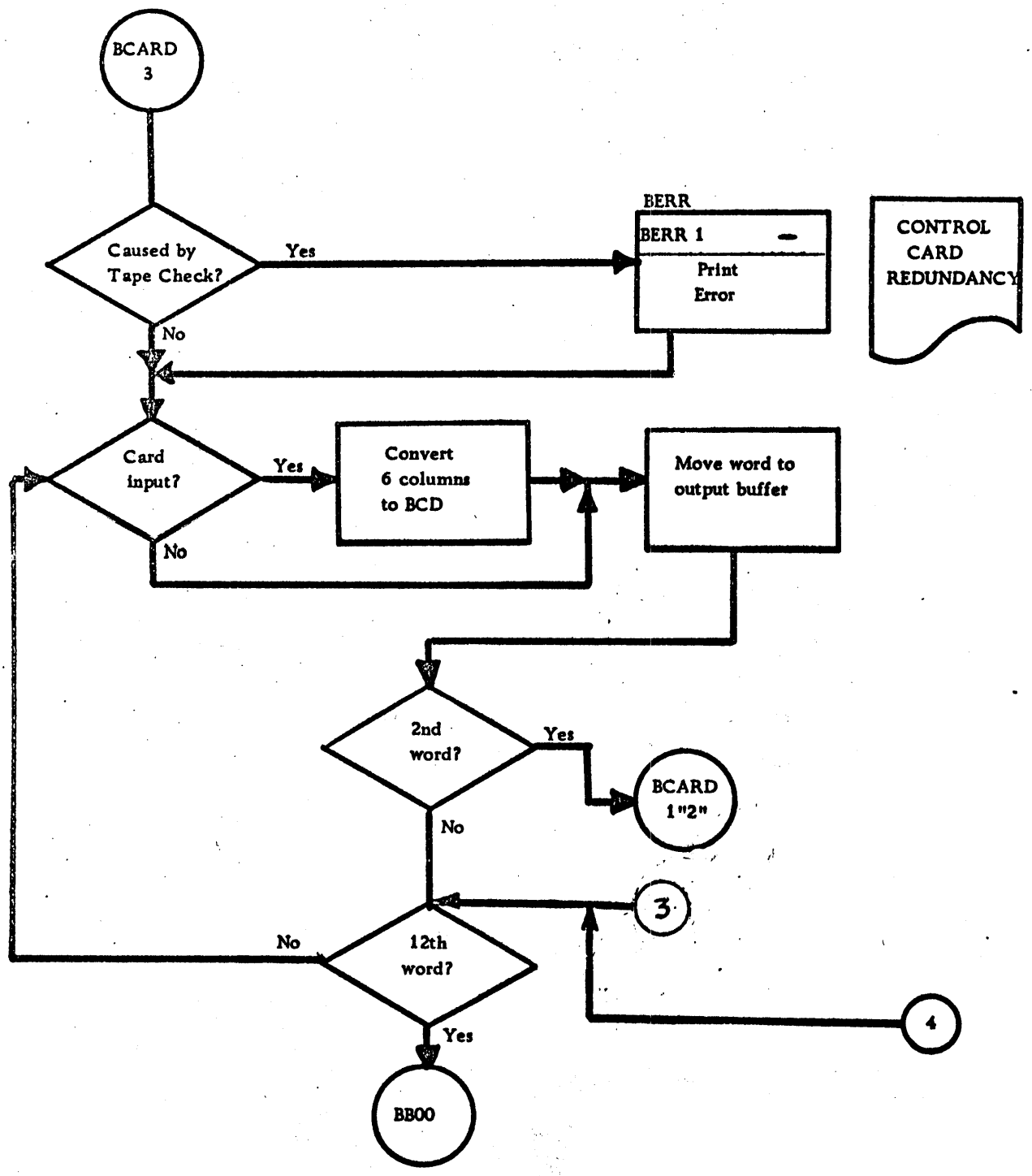

BC50
2
Any
errors?
Yes
No
PAUSE
Stop, Print
message
PRESS START
TO CONTINUE
CONDITION
IGNORED
PAUSE
Stop, Print
message
Operator
Action Pause...
Action
completed
BC52
Disable
Traps
1
Rewind
System
Tape
Load rest
or prog.
Specialize
RDS
Compute Correct
RCH
Specialize
LCH
TRA BCZ0
(IN EXECUTOR)


BCARD
Control card input routine
Is this 1st entry?
Yes
No
Set Word Count to 12_{10}
Is prog. in C.R.?
Yes
No
2
BCARD 2"1"
BCARD 3"4"
Yes
Has DATE card been read?
Set Word Count to 24_{10}
No
Invert buffer switch
Build read command
Specialize 2nd word of UCB
ACTV
ACTV
Prepare for read
BCARD 2

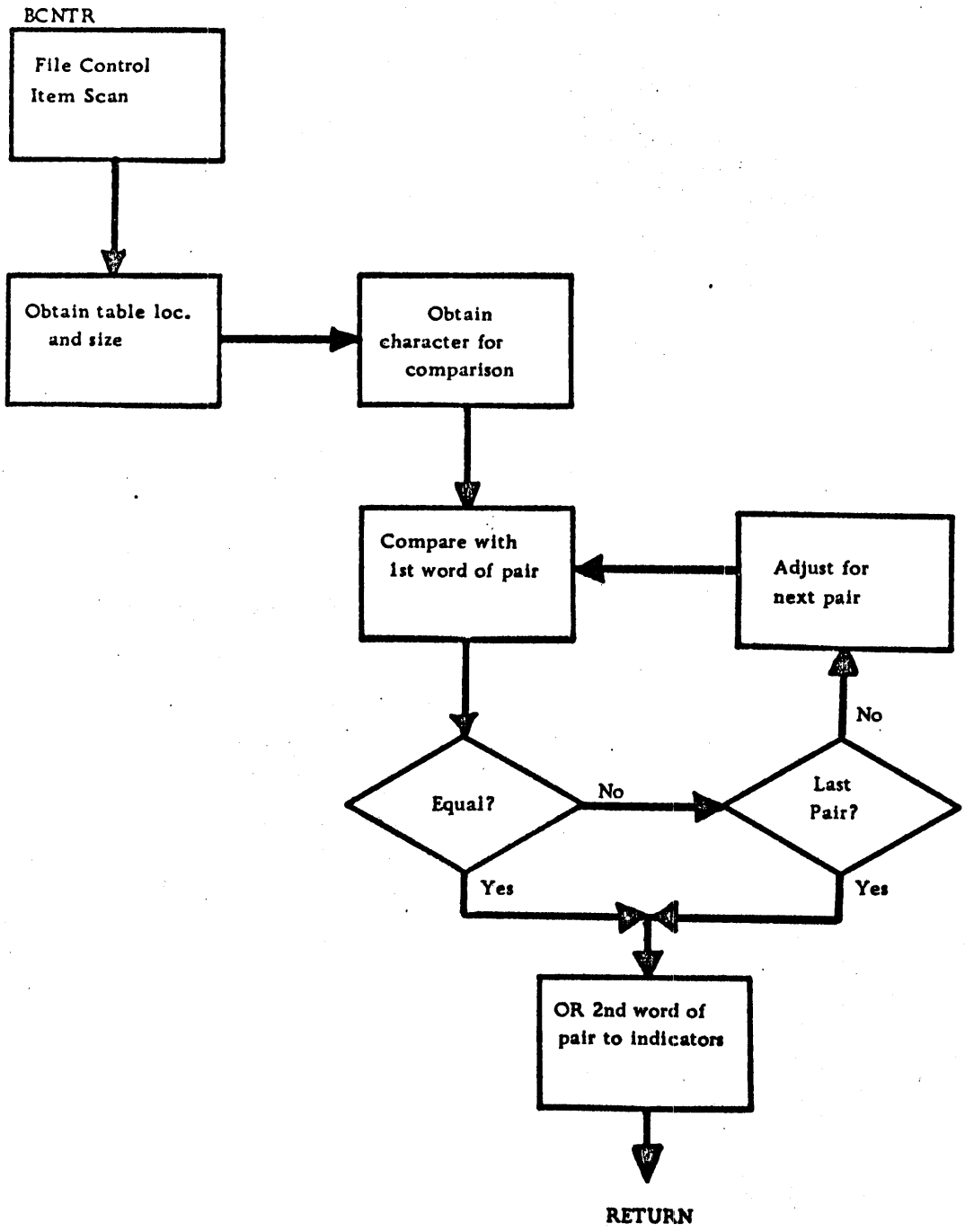
BCARD 1

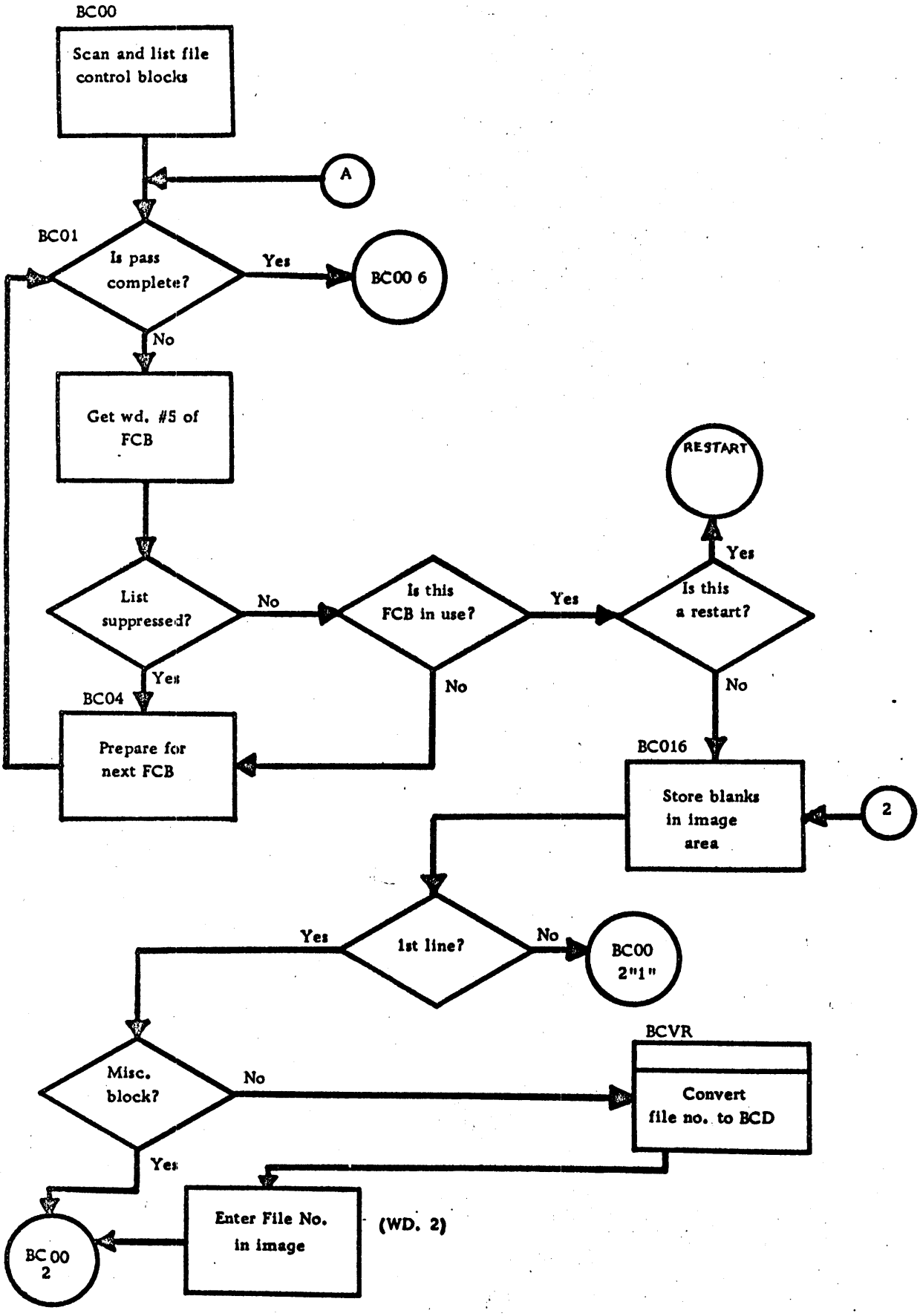

BCARD
2
A
Read one
card
1ST CARD
?
No
YES
1
B
No
Has
Trap occurred?
Yes
BCARD
3 "3"
CONTROL
CARD
EOF
Caused
by EOF?
No
BERR
BERR1
PRINT
ERROR
BCARD
3
PAUSE
STOP
BCARD
1"2"

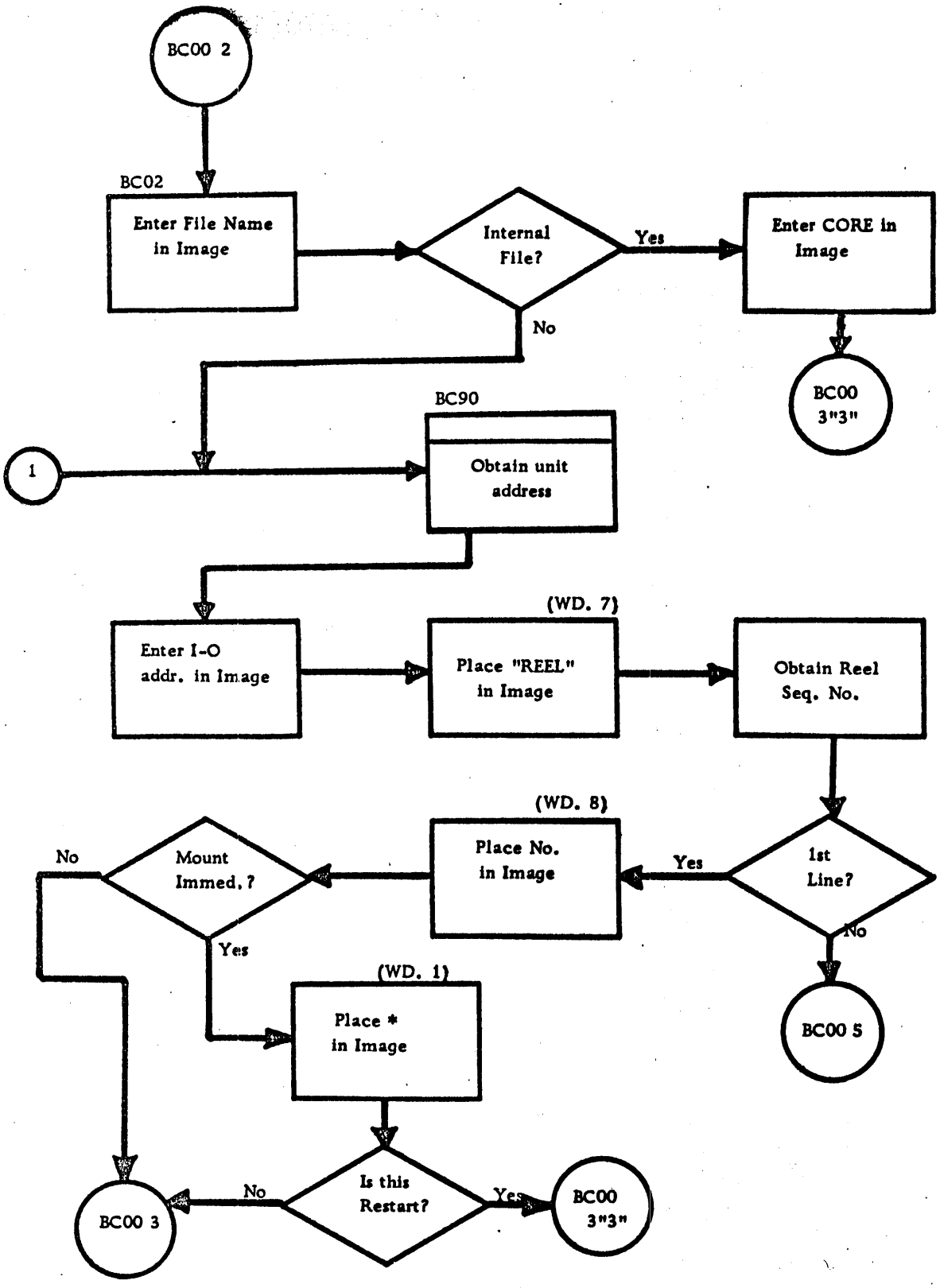
BCARD 2


BCARD
3
Caused by
Tape Check?
Yes
No
BERR
BERR 1
Print
Error
CONTROL
CARD
REDUNDANCY
Card
input?
Yes
No
Convert
6 columns
to BCD
Move word to
output buffer
2nd
word?
Yes
No
BCARD
1"2"
3
12th
word?
No
Yes
4
BB00

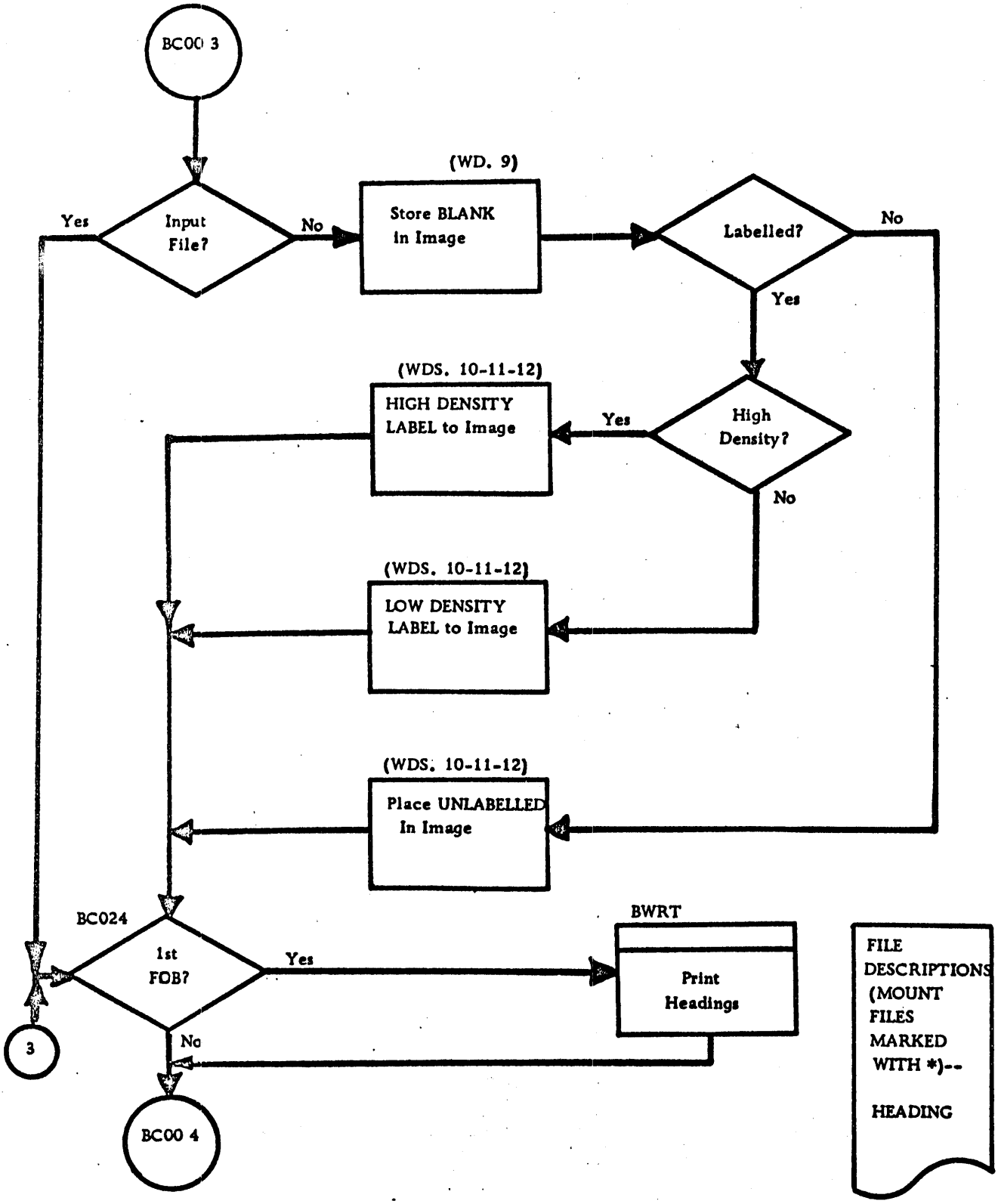
BCARD 3

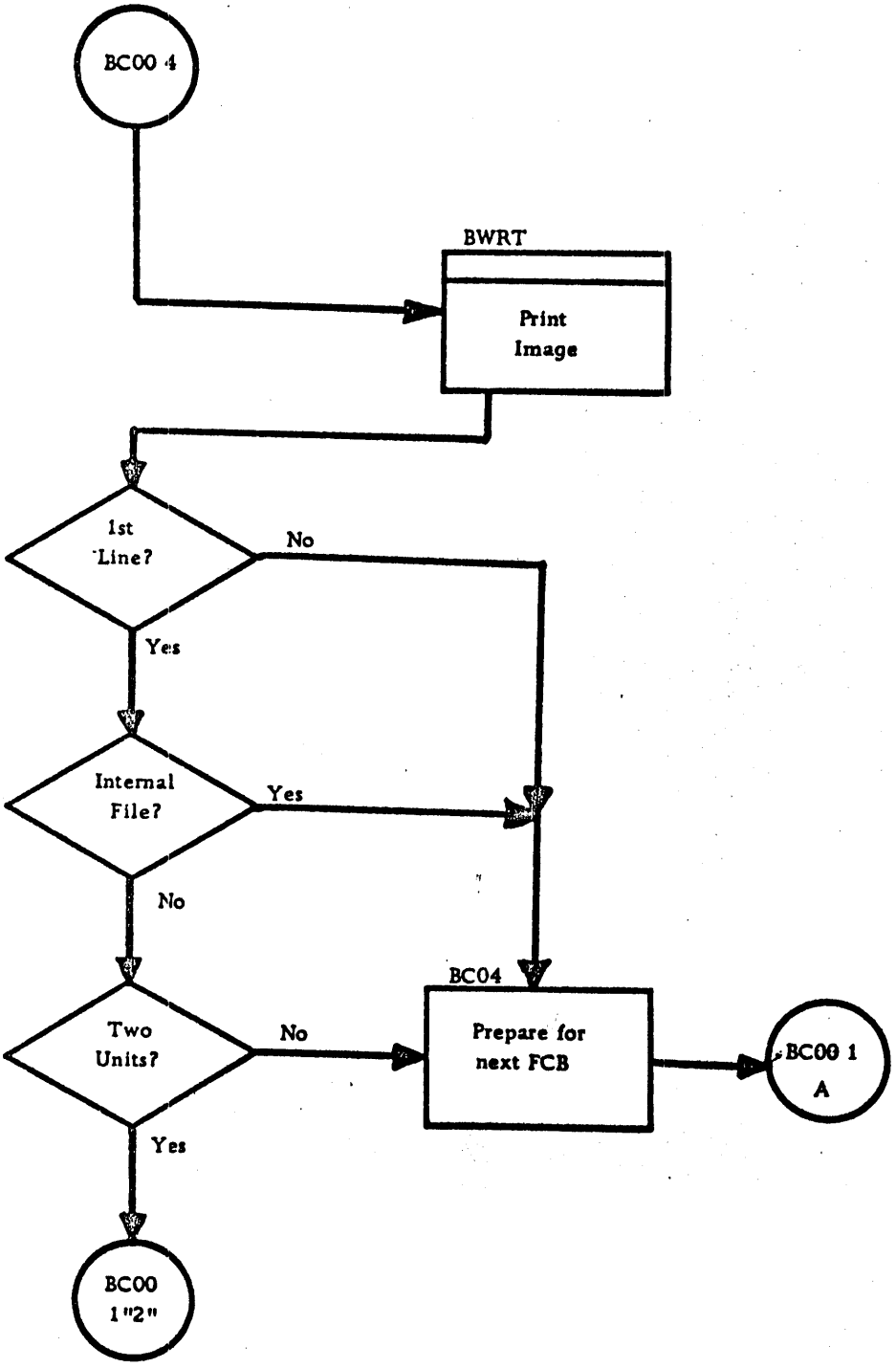

BCNTR
File Control Item Scan
Obtain table loc. and size
Obtain character for comparison
Compare with 1st word of pair
Adjust for next pair
No
Equal?
No
Last Pair?
Yes
Yes
OR 2nd word of pair to indicators
RETURN

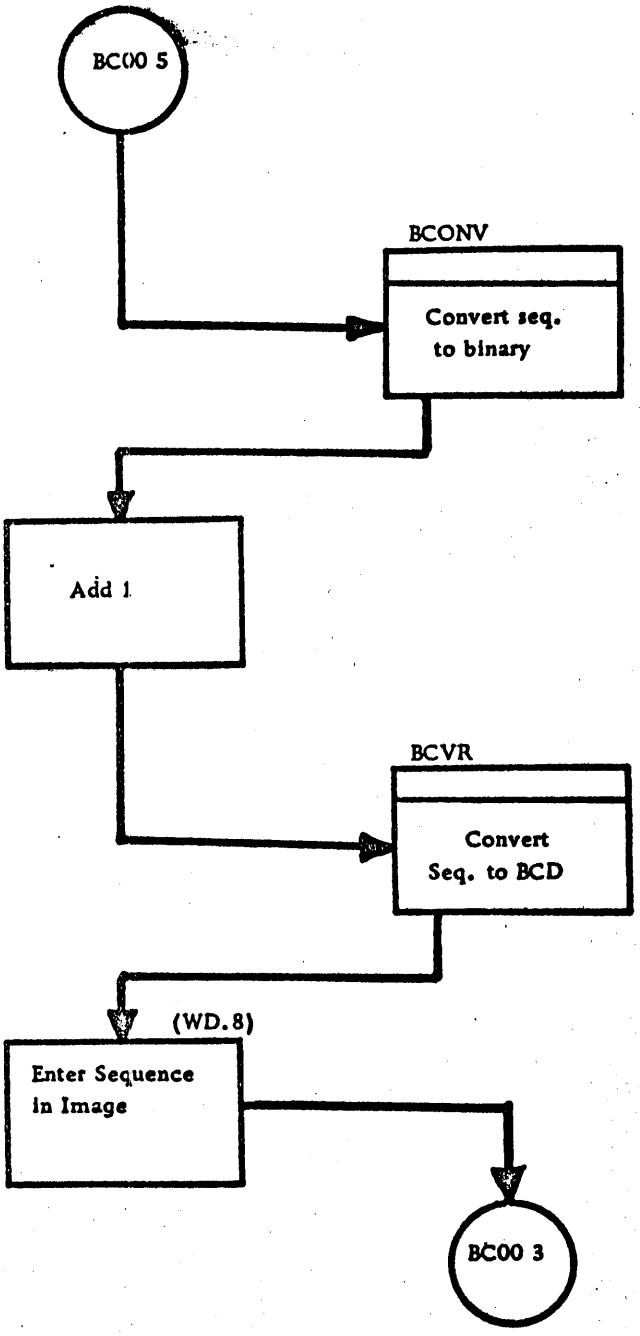

BC00
Scan and list file control blocks
A
BC01
Is pass complete?
Yes
BC00 6
No
Get wd. #5 of FCB
RESTART
List suppressed?
No
Is this FCB in use?
Yes
Is this a restart?
Yes
No
Yes
No
BC04
Prepare for next FCB
BC016
Store blanks in image area
2
Yes
1st line?
No
BC00 2"1"
BCVR
Misc. block?
No
Convert file no. to BCD
Yes
Enter File No. in image
(WD. 2)
BC00 2

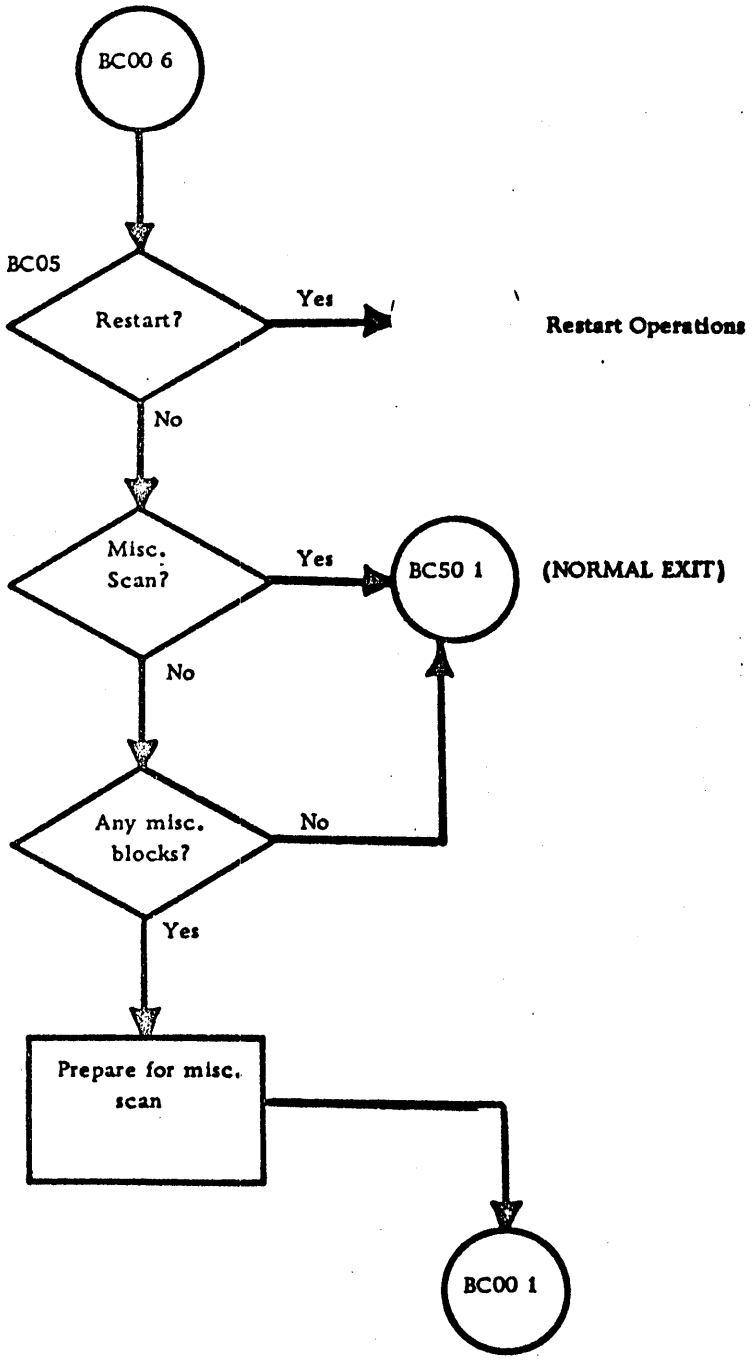

BC00 2
BC02
Enter File Name in Image
Internal File?
Yes
Enter CORE in Image
No
BC00 3"3"
BC90
1
Obtain unit address
(WD. 7)
Enter I-O addr. in Image
Place "REEL" in Image
Obtain Reel Seq. No.
(WD. 8)
Place No. in Image
Yes
1st Line?
No
No
Mount Immed.?
Yes
BC00 5
(WD. 1)
Place * in Image
No
Is this Restart?
Yes
BC00 3
BC00 3"3"

BC00 2

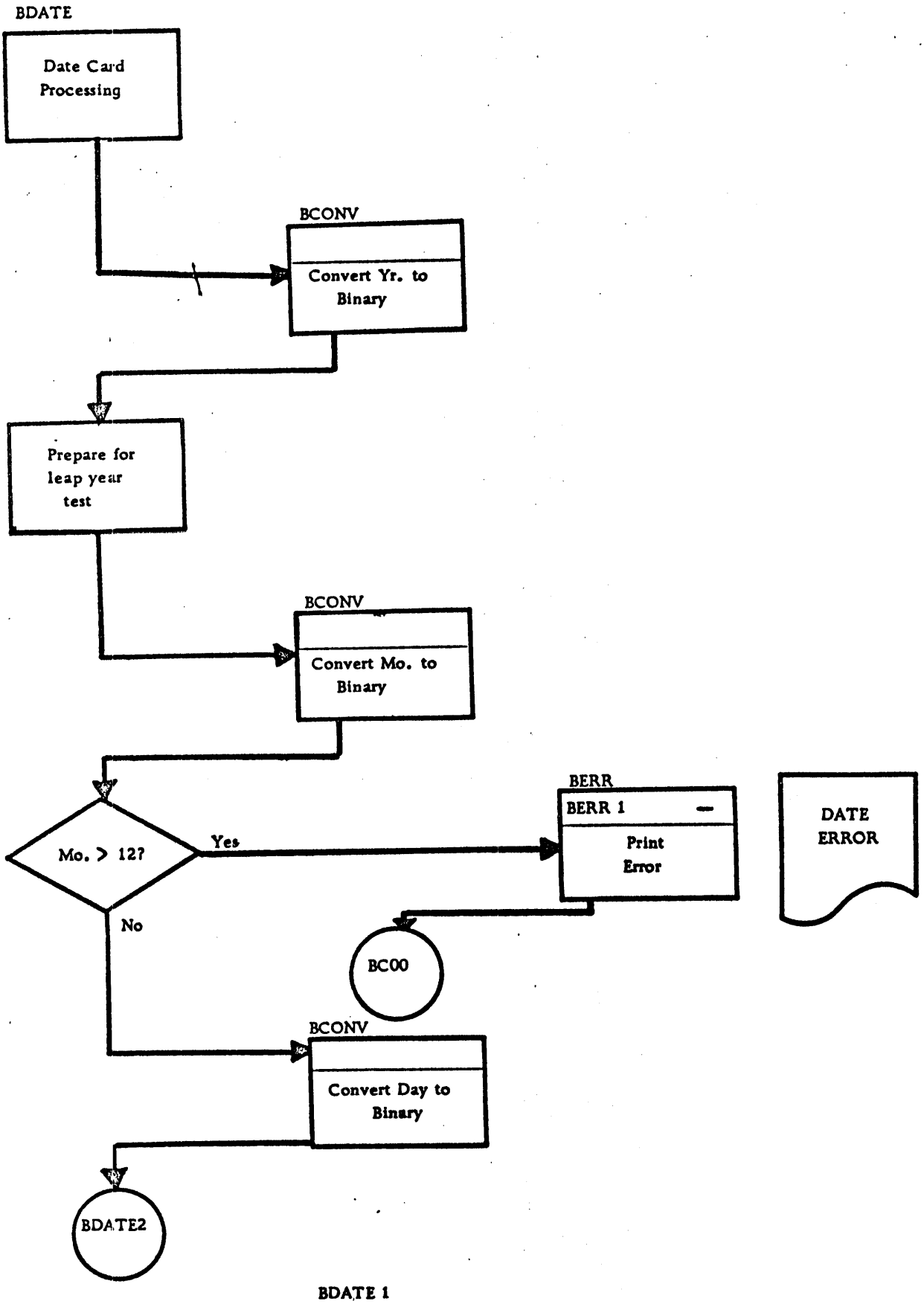

BC00 3
(WD. 9)
Store BLANK
in Image
Input
File?
Yes
No
Labelled?
No
Yes
(WDS. 10-11-12)
HIGH DENSITY
LABEL to Image
Yes
High
Density?
No
(WDS. 10-11-12)
LOW DENSITY
LABEL to Image
(WDS. 10-11-12)
Place UNLABELLED
In Image
BC024
1st
FOB?
Yes
No
BWRT
Print
Headings
3
BC00 4
FILE
DESCRIPTIONS
(MOUNT
FILES
MARKED
WITH *)--
HEADING


BC00 4
BWRT
Print
Image
1st
Line?
No
Yes
Internal
File?
Yes
No
BC04
Two
Units?
No
Prepare for
next FCB
BC00 1
A
Yes
BC00
1"2"

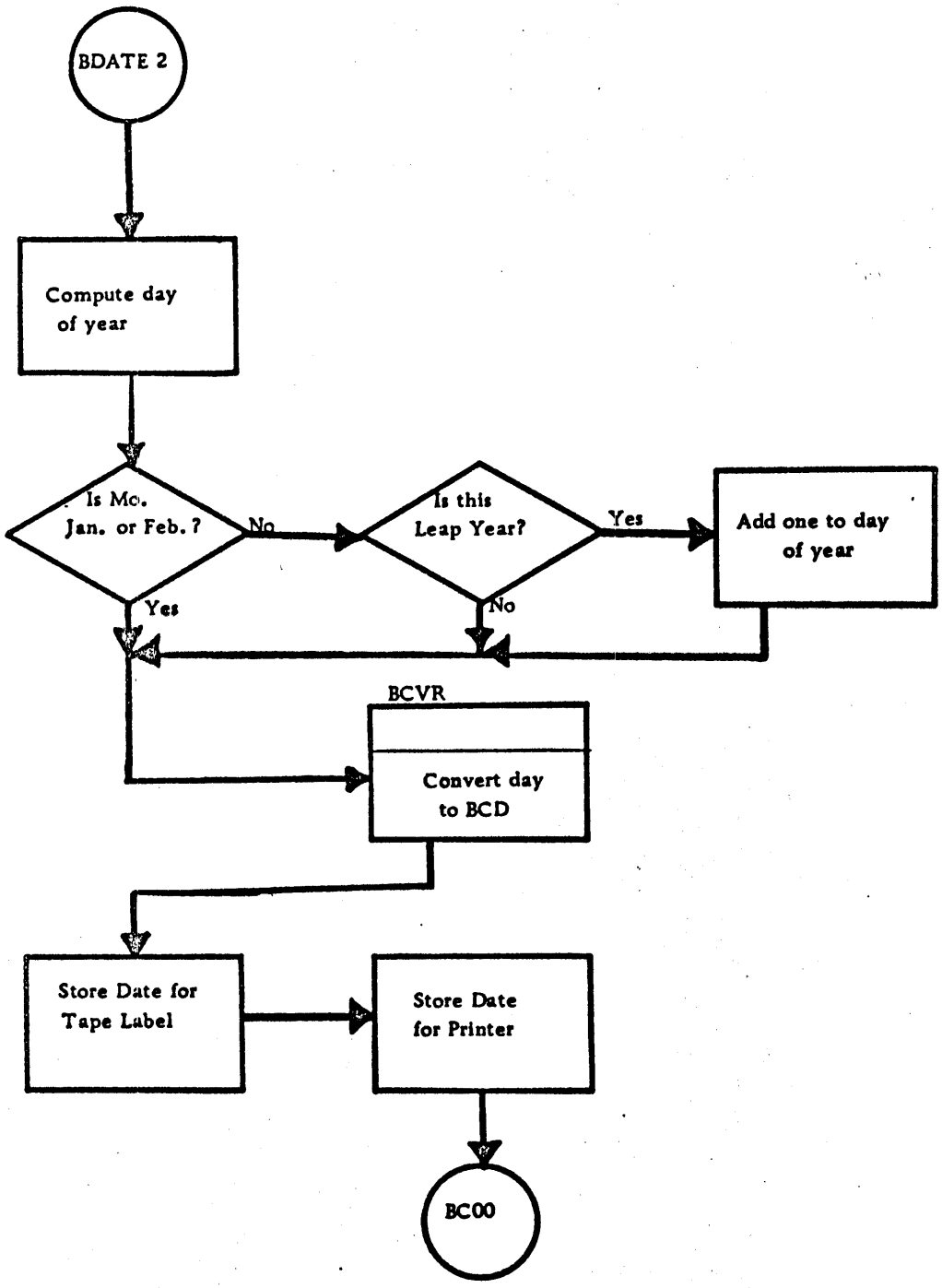

BC00 5
BCONV
Convert seq.
to binary
Add 1
BCVR
Convert
Seq. to BCD
(WD.8)
Enter Sequence
in Image
BC00 3


BC00 6
BC05
Restart?
Yes
Restart Operations
No
Misc. Scan?
Yes
BC50 1
(NORMAL EXIT)
No
Any misc. blocks?
No
Yes
Prepare for misc. scan
BC00 1

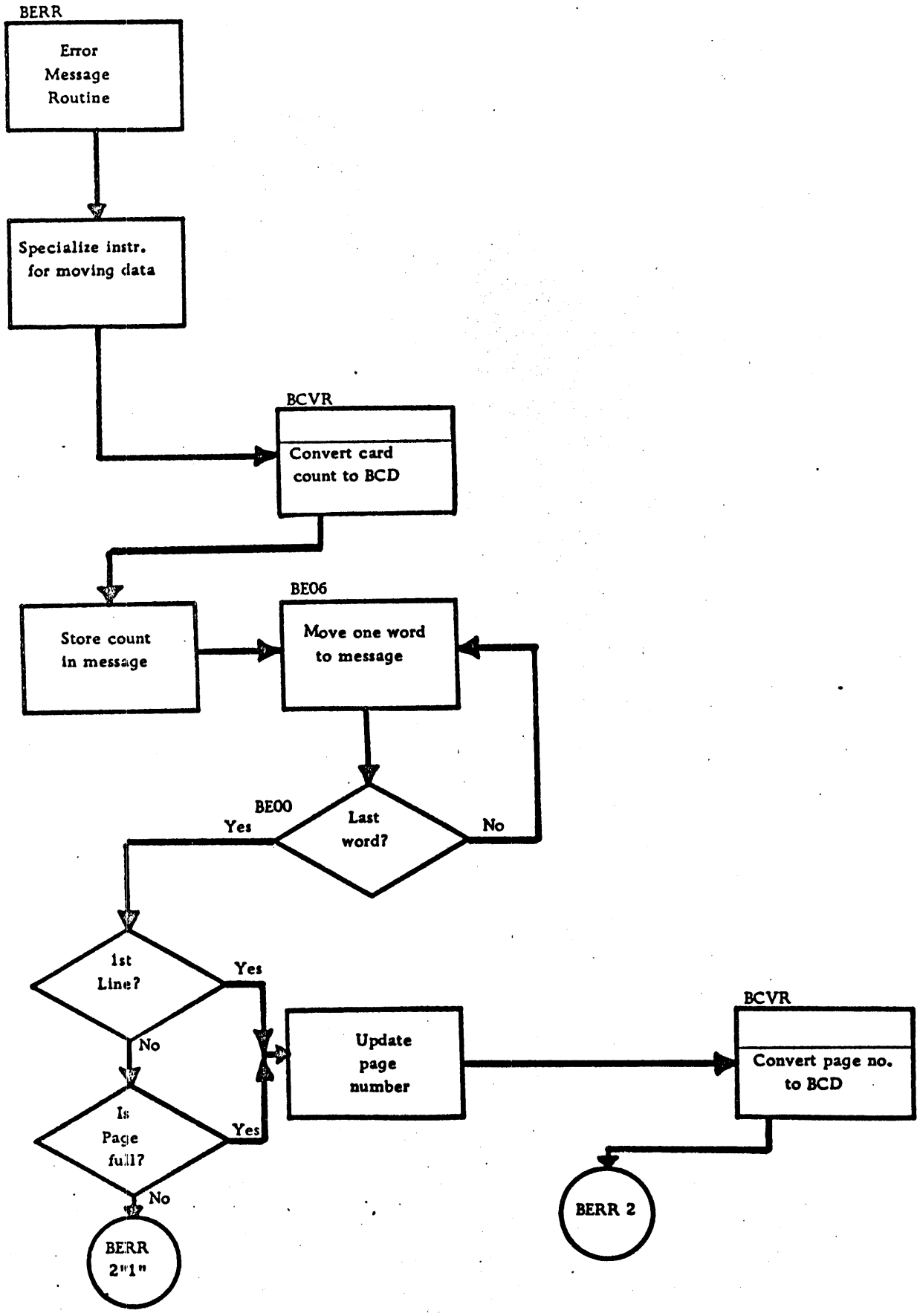
BC00 6


BDATE
Date Card Processing
BCONV
Convert Yr. to Binary
Prepare for leap year test
BCONV
Convert Mo. to Binary
Mo. > 12?
Yes
No
BERR
BERR 1
Print Error
DATE ERROR
BC00
BCONV
Convert Day to Binary
BDATE2

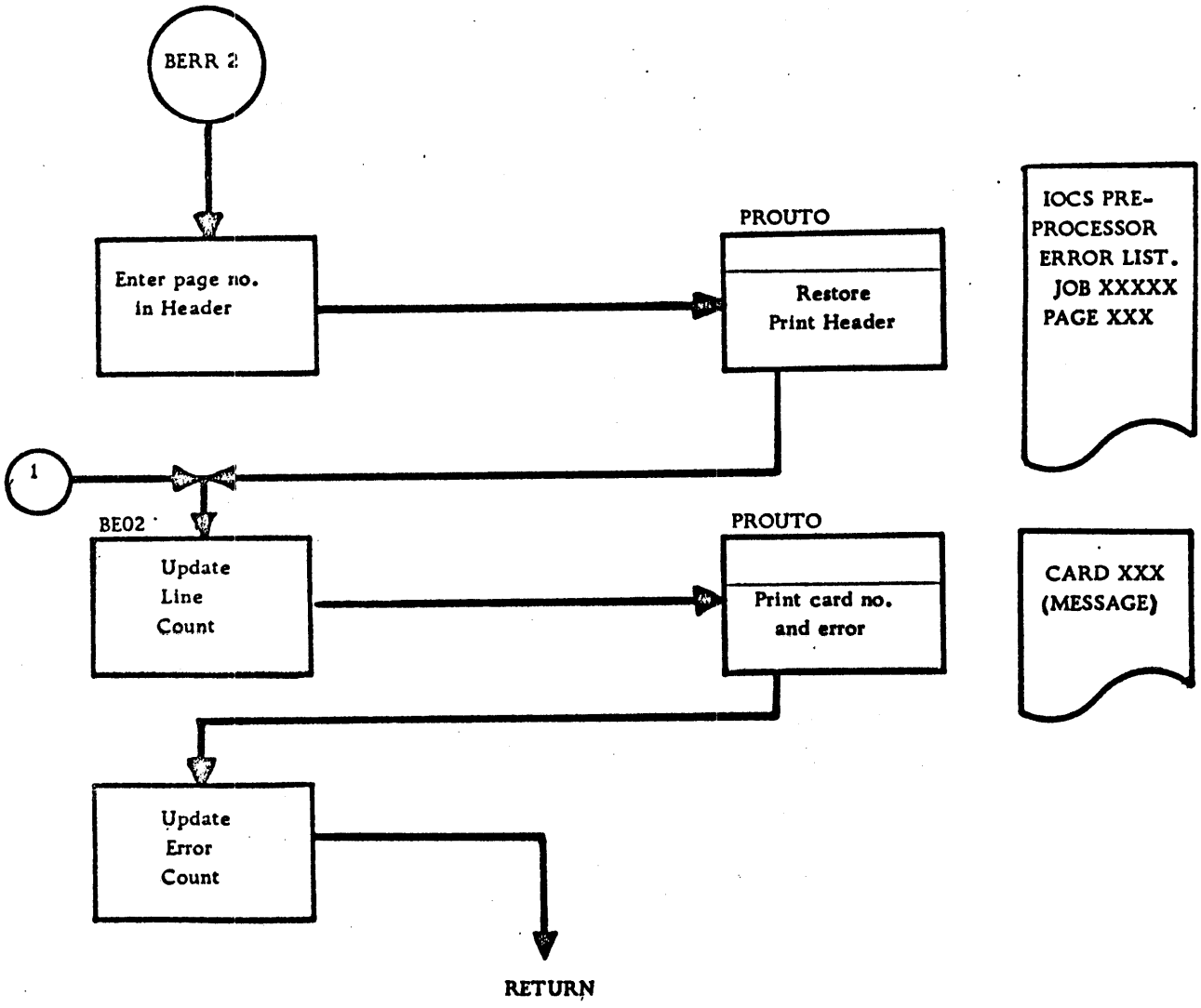
BDATE 1


BDATE 2
Compute day of year
Is Mo. Jan. or Feb.?
No
Is this Leap Year?
Yes
Add one to day of year
Yes
No
BCVR
Convert day to BCD
Store Date for Tape Label
Store Date for Printer
BC00

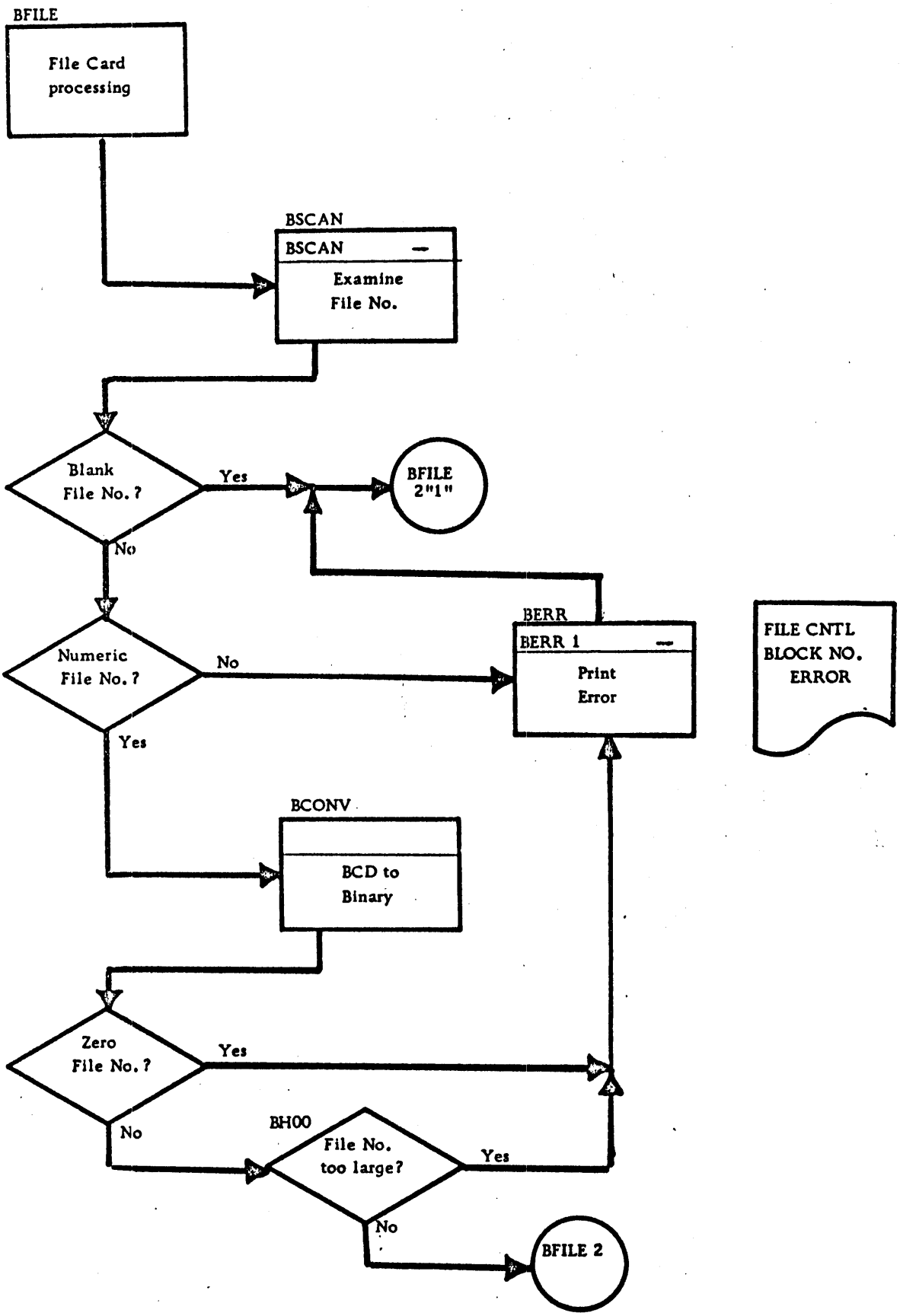
BDATE 2


BERR
Error Message Routine
Specialize instr. for moving data
BCVR
Convert card count to BCD
Store count in message
BE06
Move one word to message
BE00
Last word?
Yes
No
1st Line?
Yes
No
Update page number
BCVR
Convert page no. to BCD
Is Page full?
Yes
No
BERR 2
BERR 2"1"

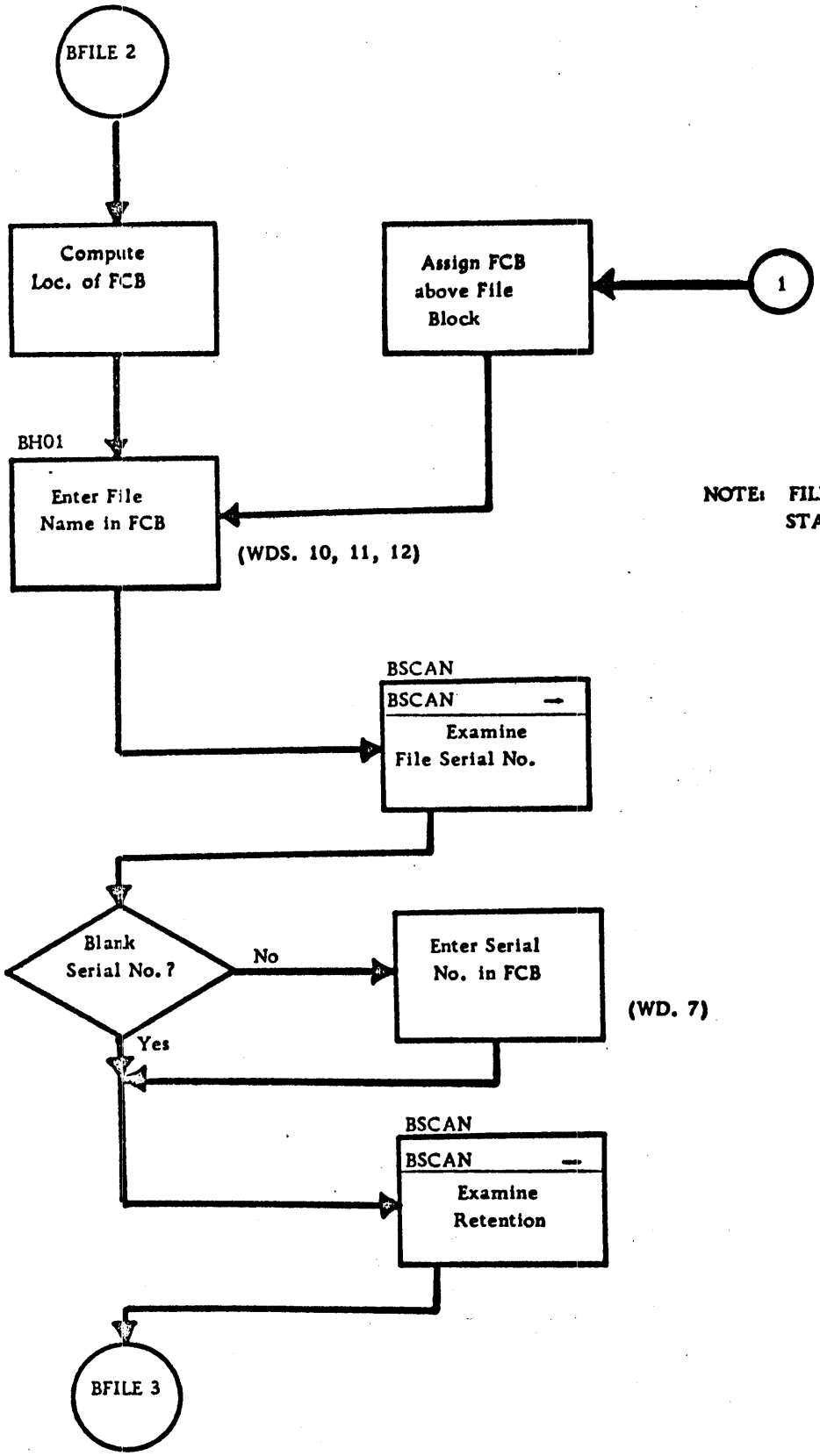
BERR 1


BERR 2
Enter page no. in Header
PROUTO
Restore Print Header
IOCS PRE-PROCESSOR ERROR LIST. JOB XXXXX PAGE XXX
1
BE02
Update Line Count
PROUTO
Print card no. and error
CARD XXX (MESSAGE)
Update Error Count
RETURN

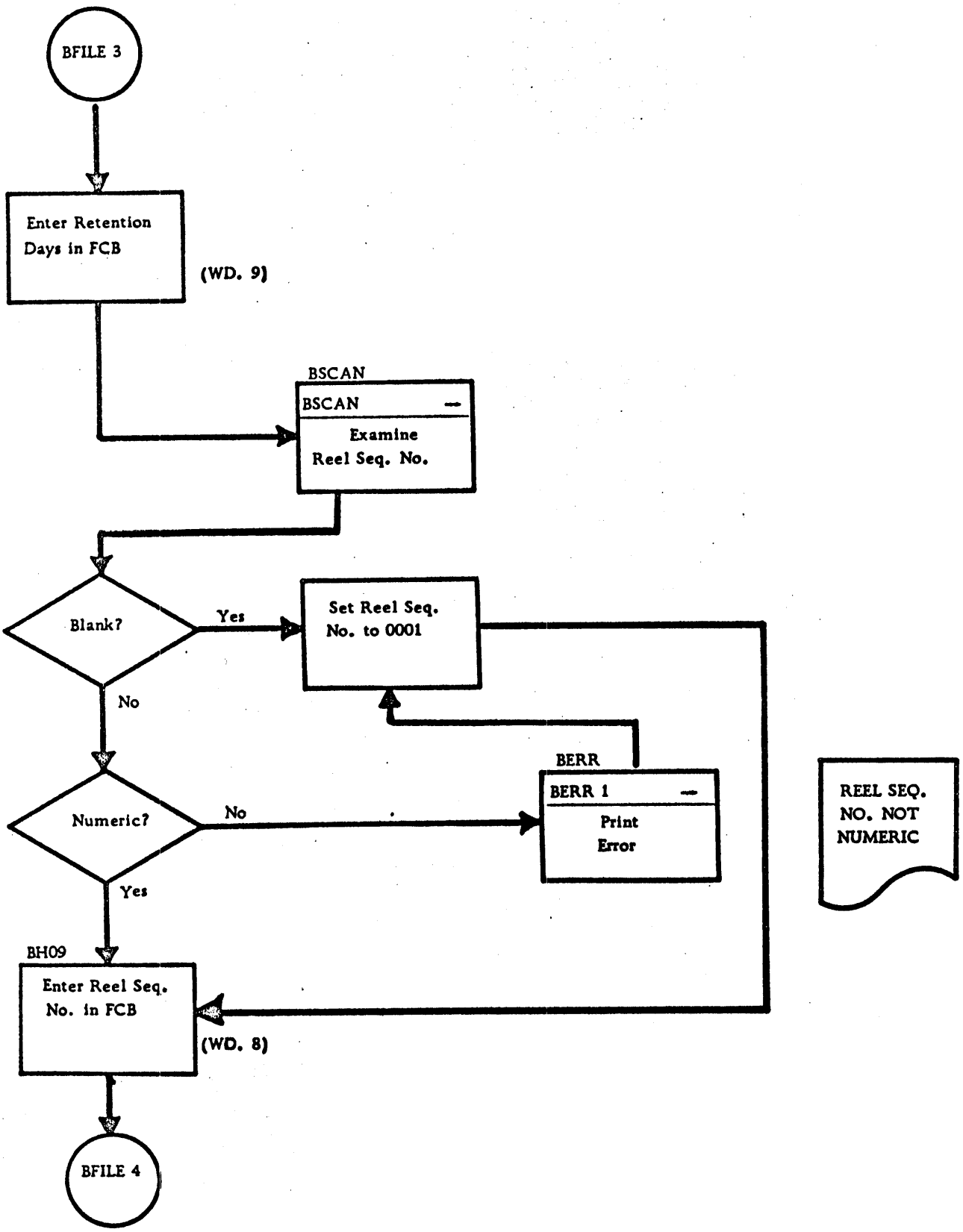
BERR 2


BFILE
File Card processing
BSCAN
BSCAN —
Examine File No.
Blank File No.?
Yes
BFILE 2"1"
No
BERR
BERR 1 —
Print Error
FILE CNTL BLOCK NO. ERROR
Numeric File No.?
No
Yes
BCONV
BCD to Binary
Zero File No.?
Yes
No
BH00
File No. too large?
Yes
No
BFILE 2

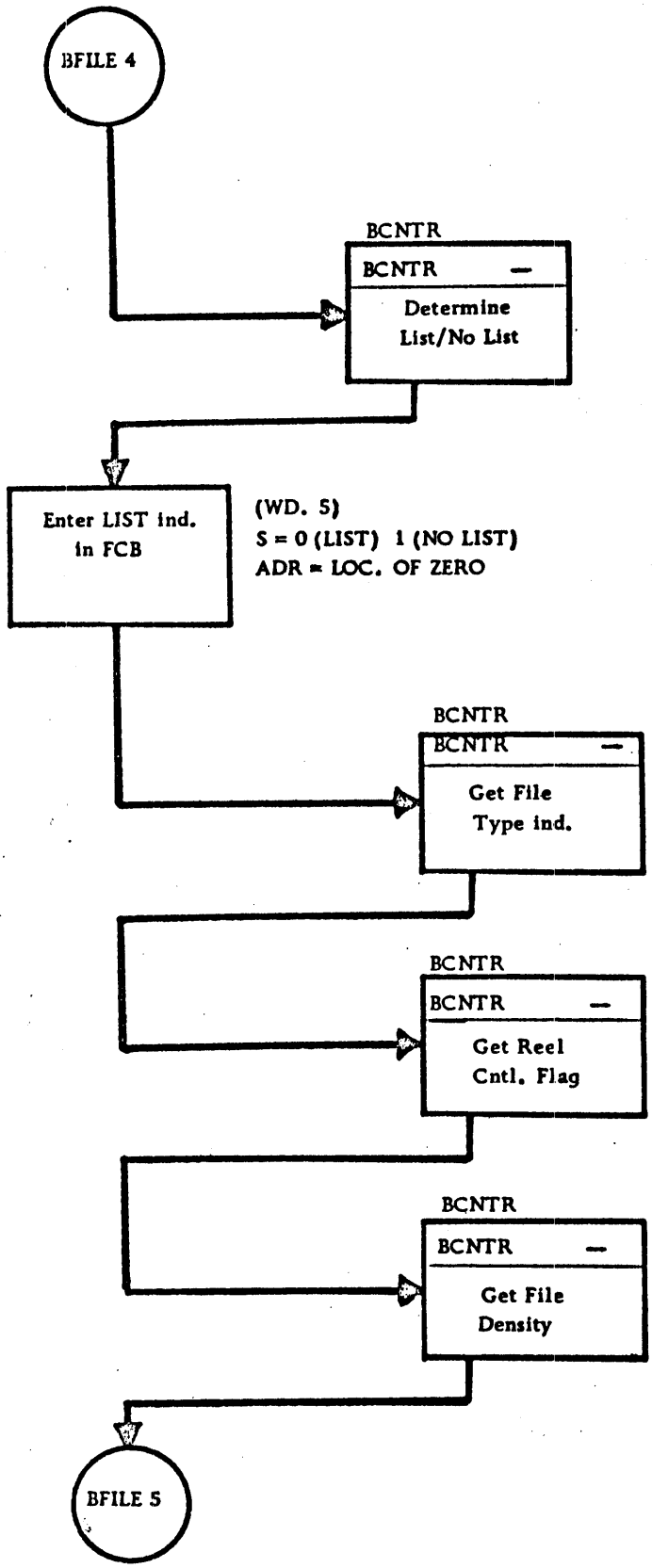
BFILE 1


BFILE 2
Compute Loc. of FCB
Assign FCB above File Block
1
BH01
Enter File Name in FCB
(WDS. 10, 11, 12)
NOTE: FILE BLOCK IS USING STANDARD ORIGIN.
BSCAN
BSCAN
Examine File Serial No.
Blank Serial No.?
No
Enter Serial No. in FCB
(WD. 7)
Yes
BSCAN
BSCAN
Examine Retention
BFILE 3

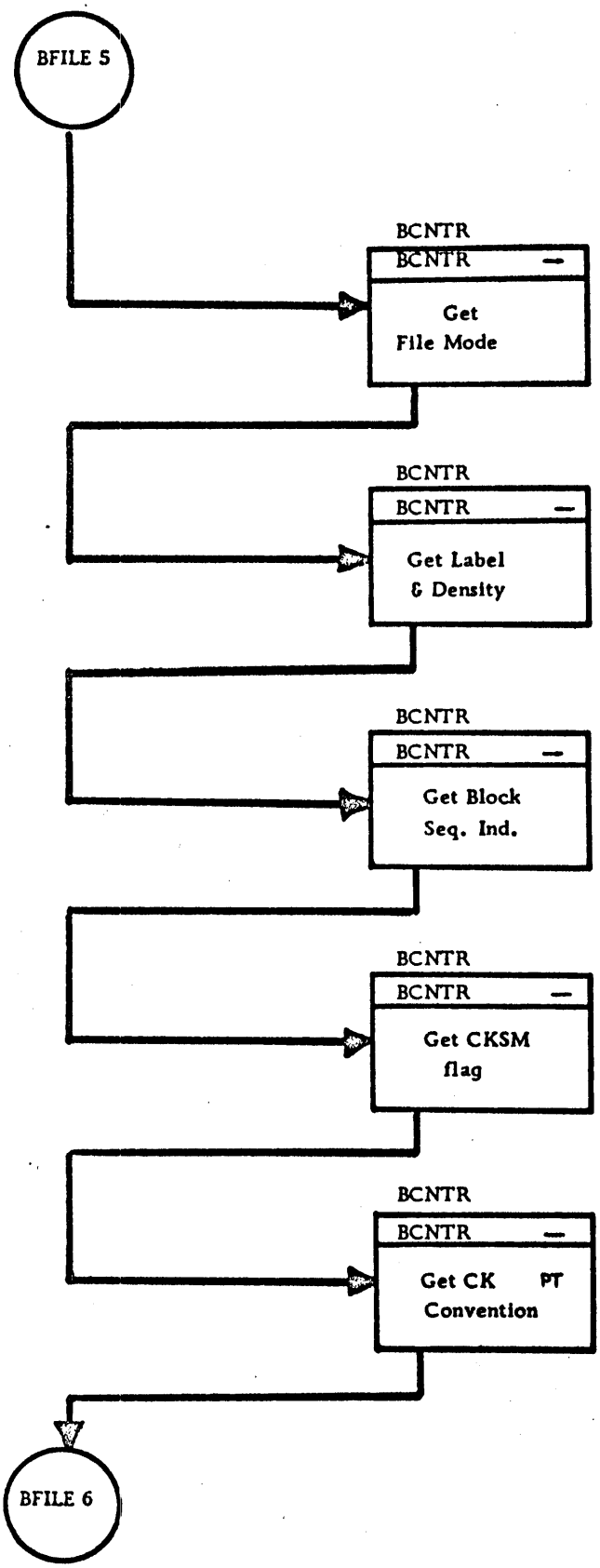
BFILE 2


BFILE 3
Enter Retention Days in FCB
(WD. 9)
BSCAN
BSCAN —
Examine Reel Seq. No.
Blank?
Yes
Set Reel Seq. No. to 0001
No
BERR
BERR 1 —
Print Error
Numeric?
No
REEL SEQ. NO. NOT NUMERIC
Yes
BH09
Enter Reel Seq. No. in FCB
(WD. 8)
BFILE 4

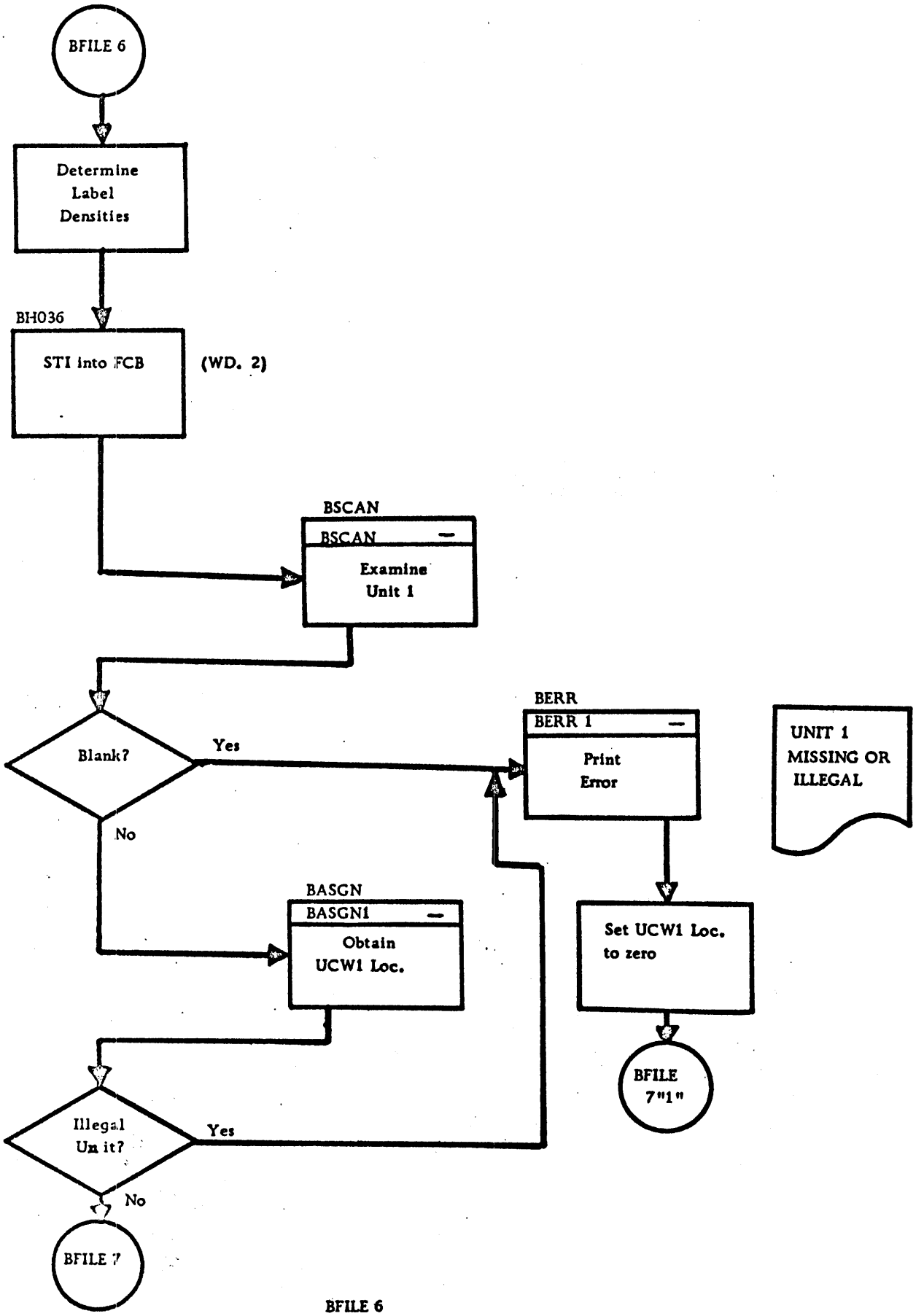
BFILE 3


BFILE 4
BCNTR
BCNTR —
Determine
List/No List
Enter LIST ind.
in FCB
(WD. 5)
S = 0 (LIST) 1 (NO LIST)
ADR = LOC. OF ZERO
BCNTR
BCNTR —
Get File
Type ind.
BCNTR
BCNTR —
Get Reel
Cntl. Flag
BCNTR
BCNTR —
Get File
Density
BFILE 5

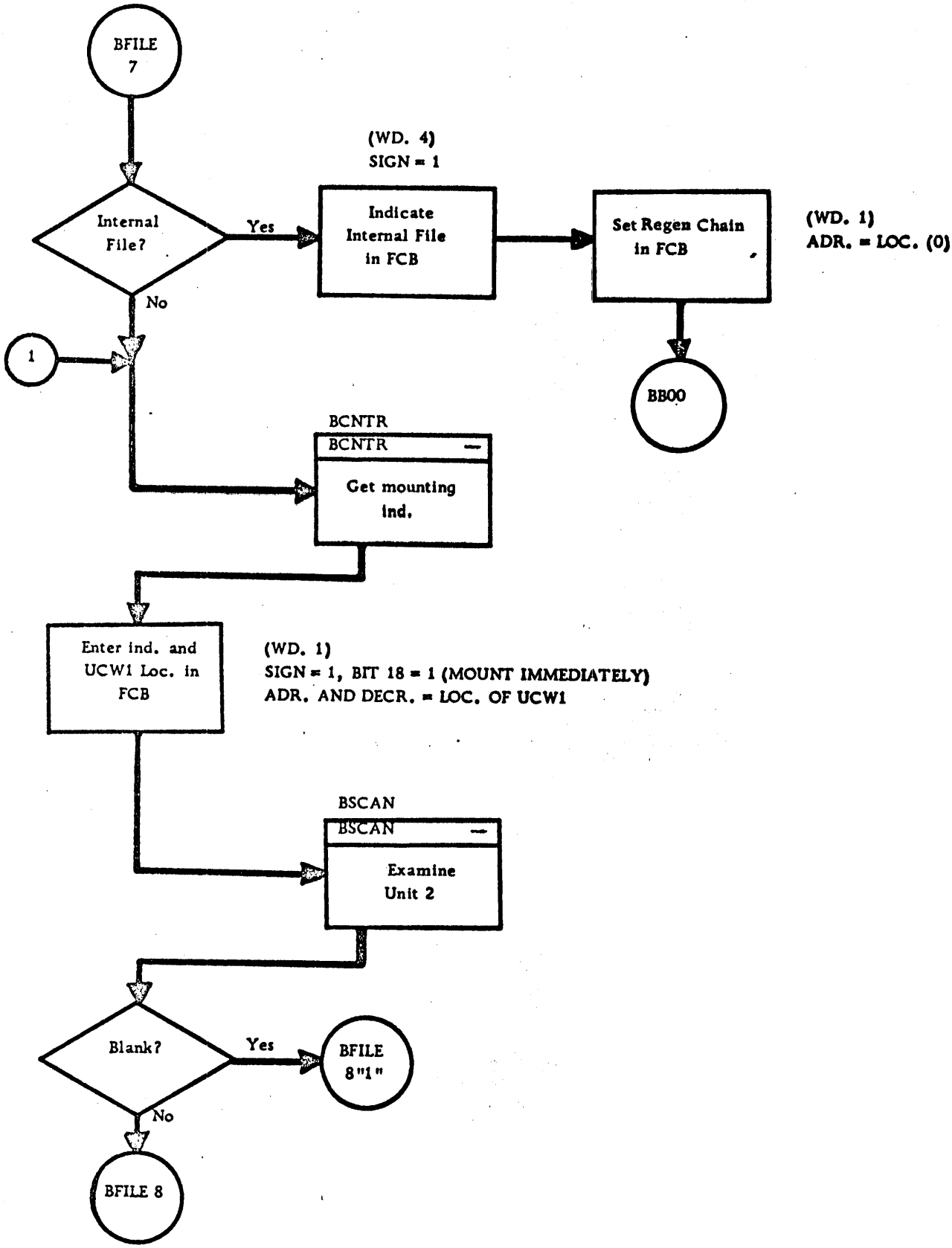
BFILE 4


BFILE 5
BCNTR
BCNTR
Get
File Mode
BCNTR
BCNTR
Get Label
& Density
BCNTR
BCNTR
Get Block
Seq. Ind.
BCNTR
BCNTR
Get CKSM
flag
BCNTR
BCNTR
Get CK PT
Convention
BFILE 6

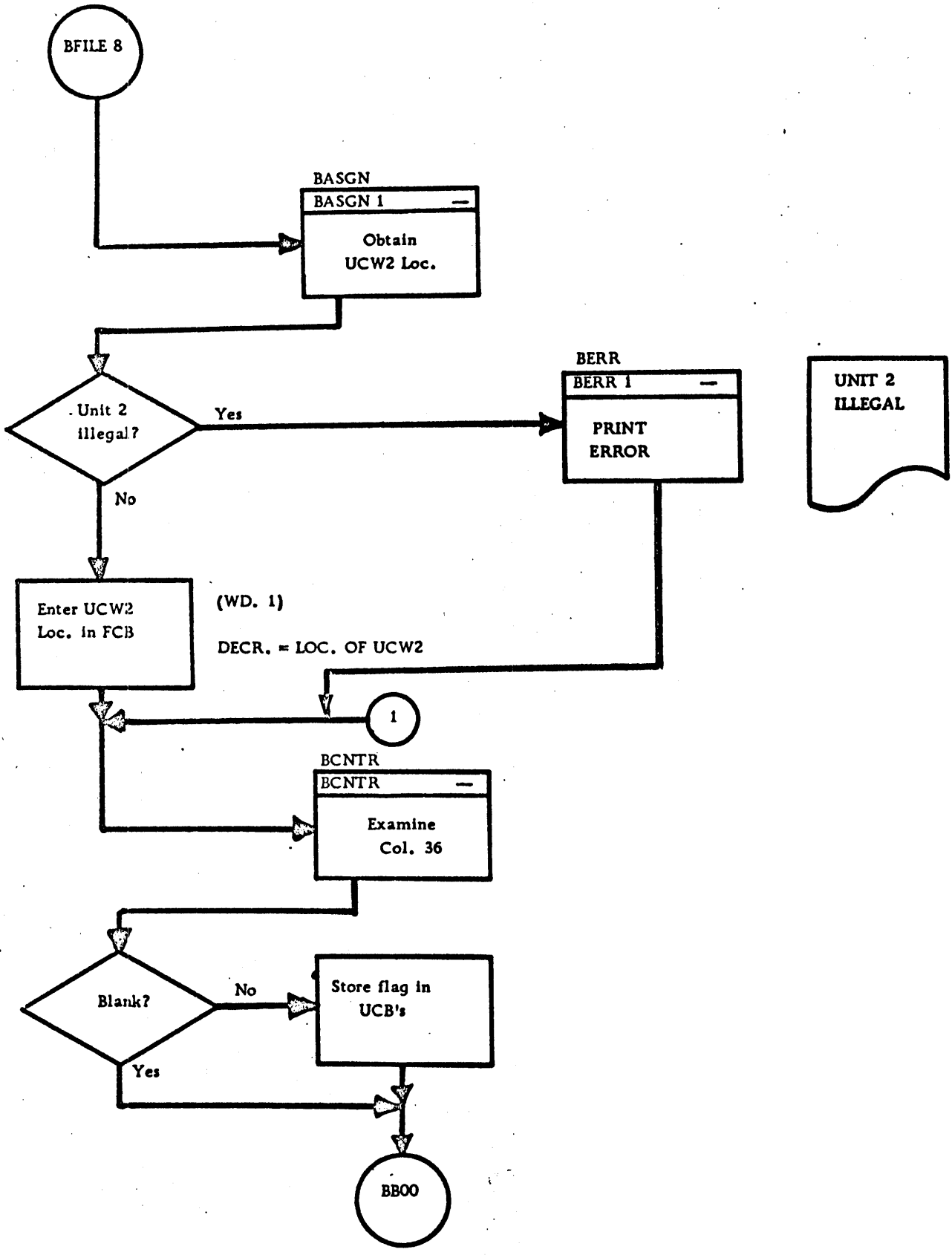
BFILE 5


BFILE 6
Determine Label Densities
BH036
STI into FCB
(WD. 2)
BSCAN
BSCAN
Examine Unit 1
Blank?
Yes
No
BERR
BERR 1
Print Error
UNIT 1 MISSING OR ILLEGAL
BASGN
BASGN1
Obtain UCW1 Loc.
Set UCW1 Loc. to zero
BFILE 7"1"
Illegal Unit?
Yes
No
BFILE 7

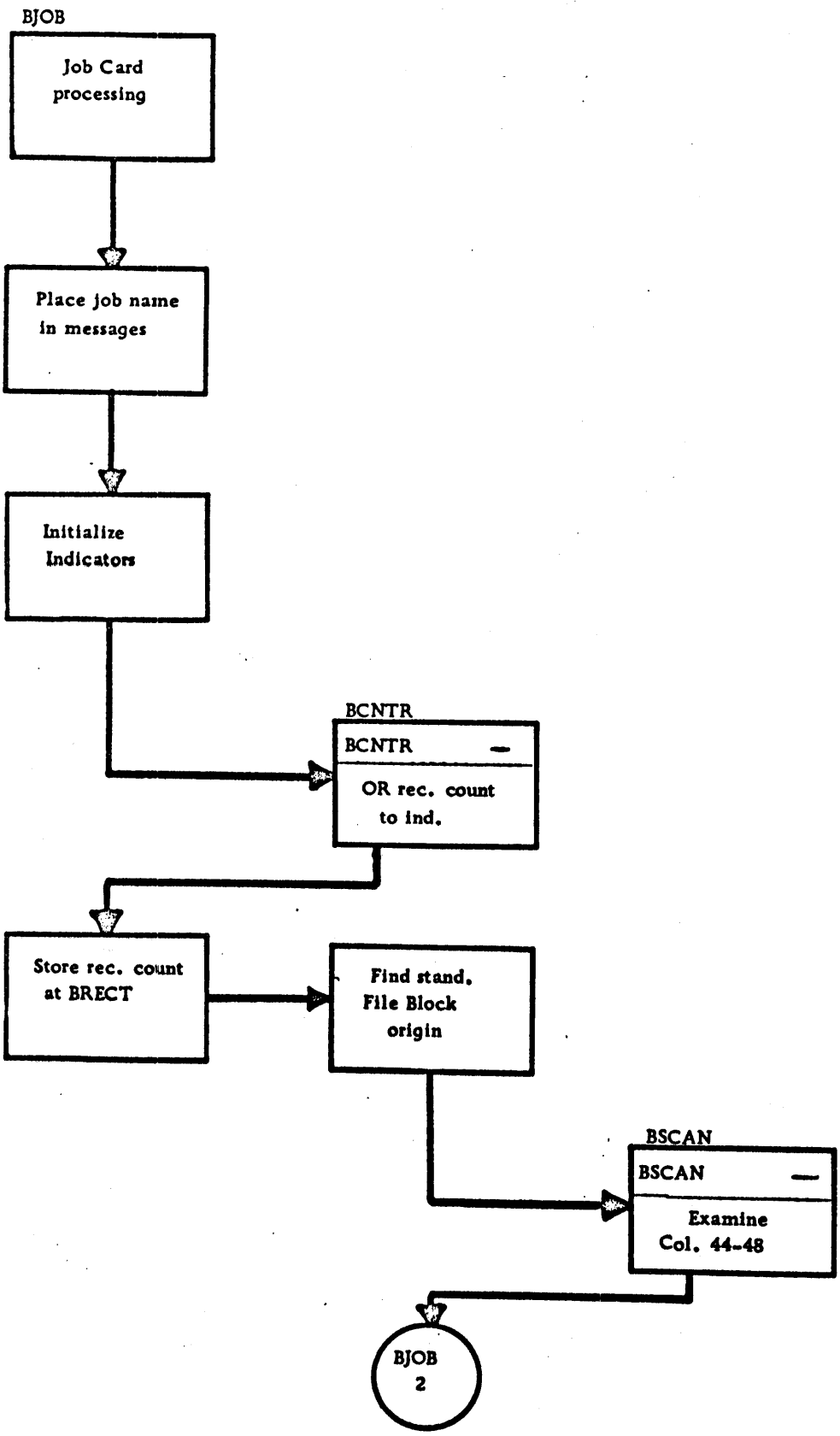
BFILE 6


BFILE 7
Internal File?
Yes
No
(WD. 4)
SIGN = 1
Indicate Internal File in FCB
Set Regen Chain in FCB
(WD. 1)
ADR. = LOC. (0)
BB00
1
BCNTR
BCNTR
Get mounting ind.
Enter ind. and UCW1 Loc. in FCB
(WD. 1)
SIGN = 1, BIT 18 = 1 (MOUNT IMMEDIATELY)
ADR. AND DECR. = LOC. OF UCW1
BSCAN
BSCAN
Examine Unit 2
Blank?
Yes
No
BFILE 8 "1"
BFILE 8

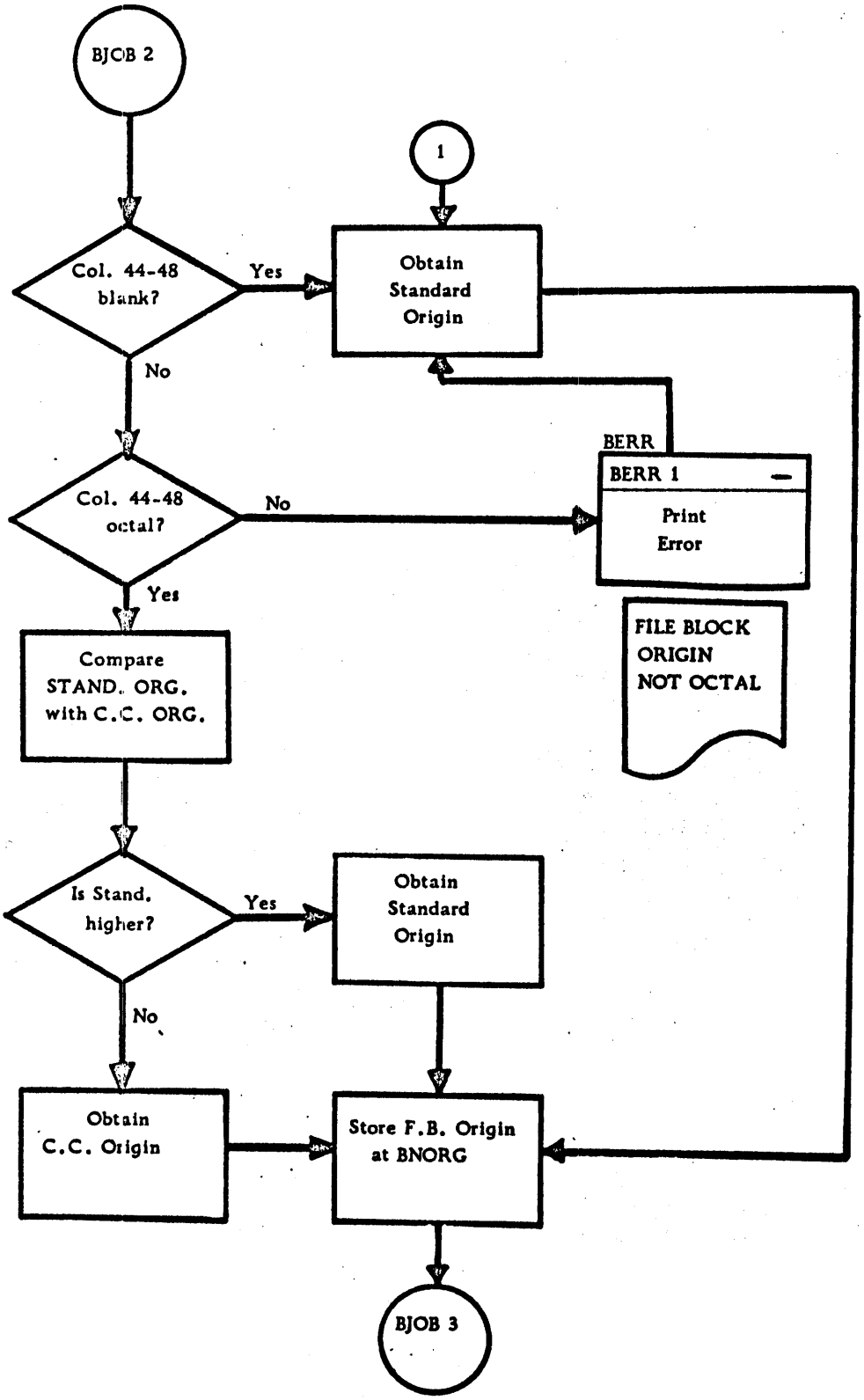
BFILE 7


BFILE 8
BASGN
BASGN 1
Obtain
UCW2 Loc.
BERR
BERR 1
PRINT
ERROR
UNIT 2
ILLEGAL
Unit 2
illegal?
Yes
No
Enter UCW2
Loc. in FCB
(WD. 1)
DECR. = LOC. OF UCW2
1
BCNTR
BCNTR
Examine
Col. 36
Blank?
No
Store flag in
UCB's
Yes
BB00

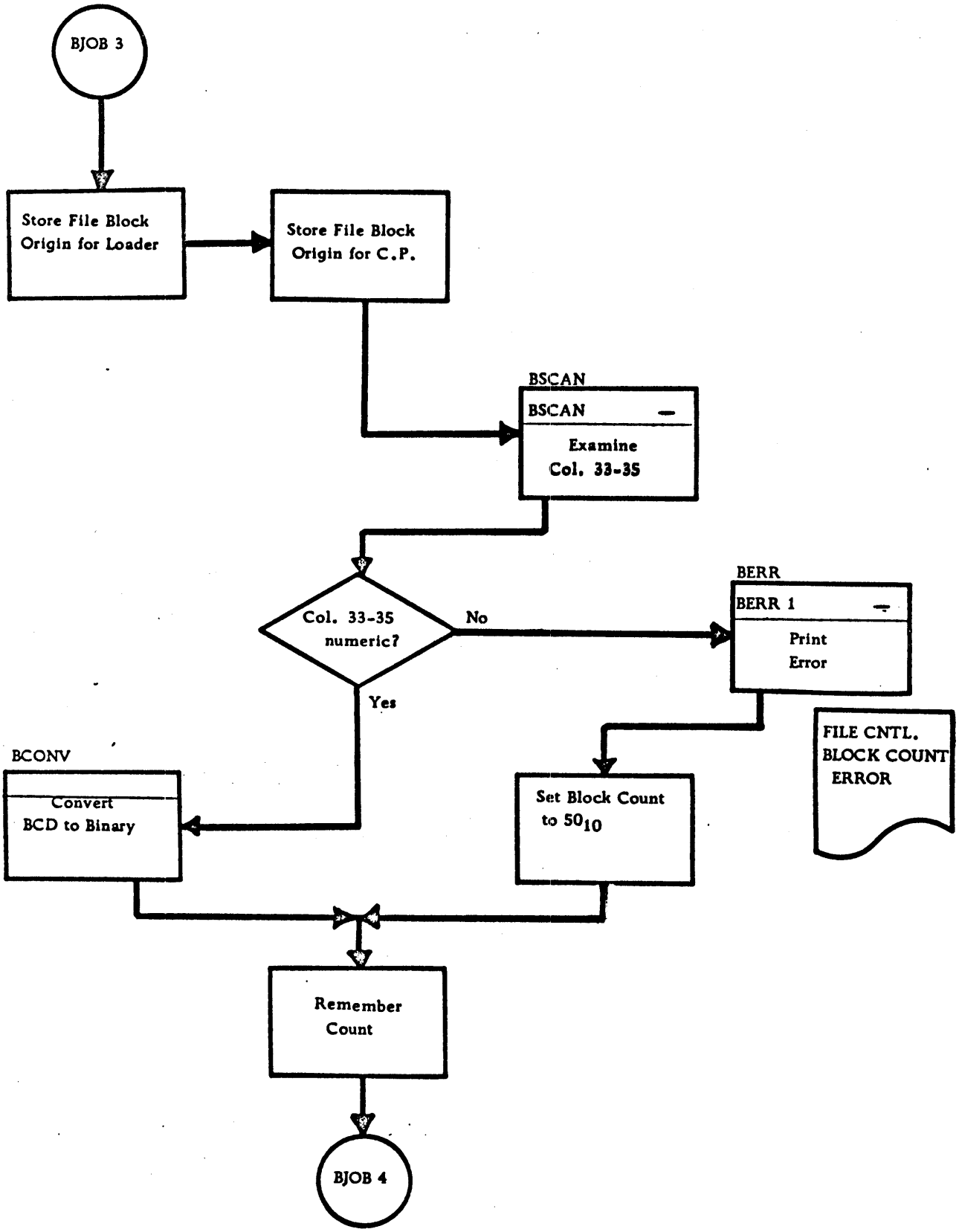
BFILE 8


BJOB
Job Card processing
Place job name in messages
Initialize Indicators
BCNTR
BCNTR —
OR rec. count to ind.
Store rec. count at BRECT
Find stand. File Block origin
BSCAN
BSCAN —
Examine Col. 44-48
BJOB 2

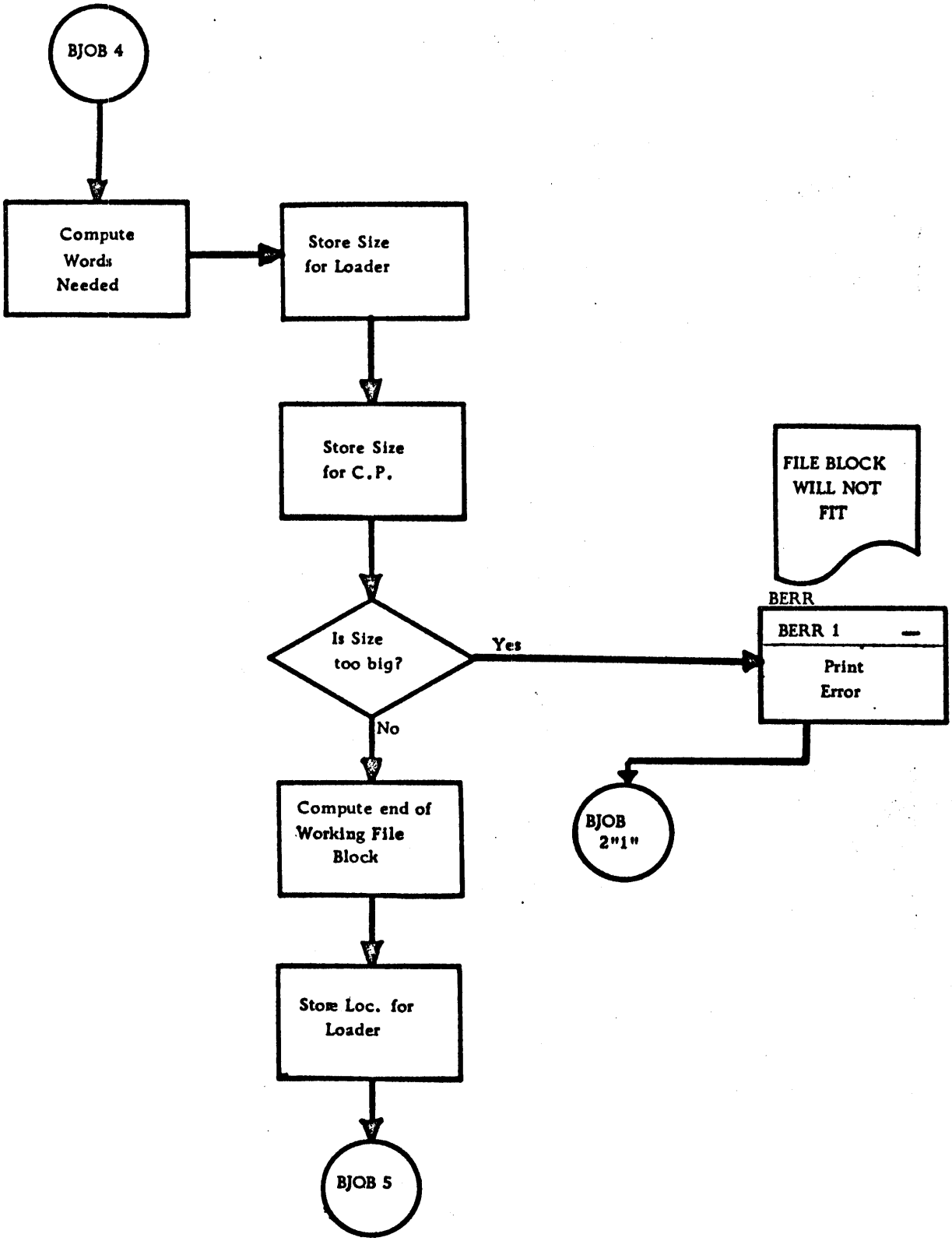
BJOB 1


BJOB 2
1
Col. 44-48 blank?
Yes
Obtain Standard Origin
No
BERR
BERR 1
Print Error
Col. 44-48 octal?
No
Yes
FILE BLOCK ORIGIN NOT OCTAL
Compare STAND. ORG. with C.C. ORG.
Is Stand. higher?
Yes
Obtain Standard Origin
No
Obtain C.C. Origin
Store F.B. Origin at BNORG
BJOB 3

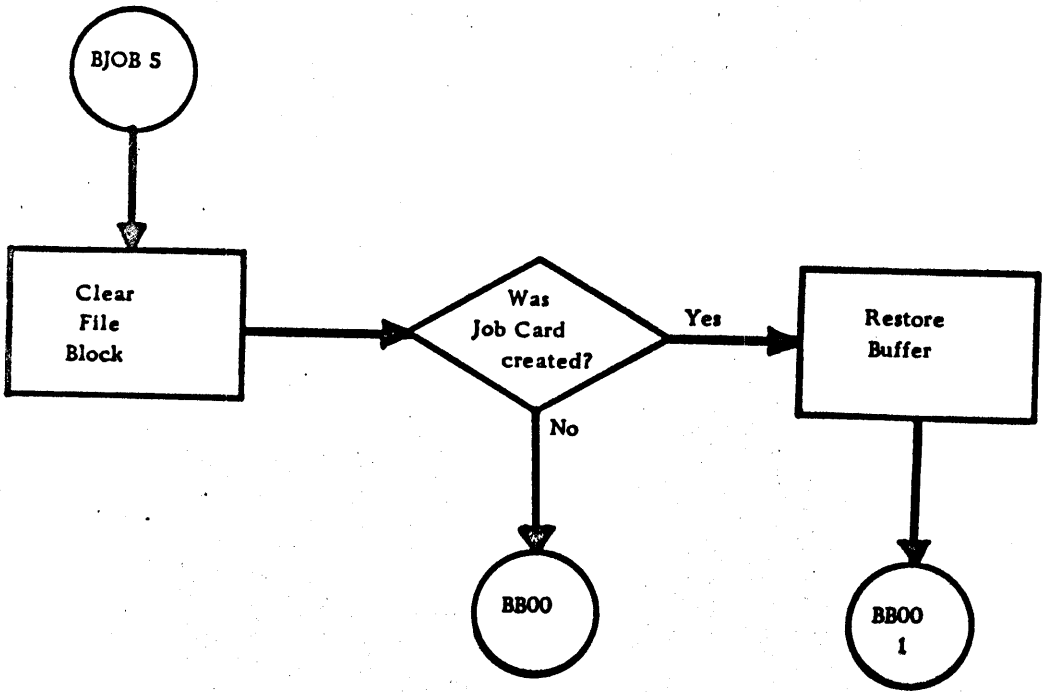
BJOB 2

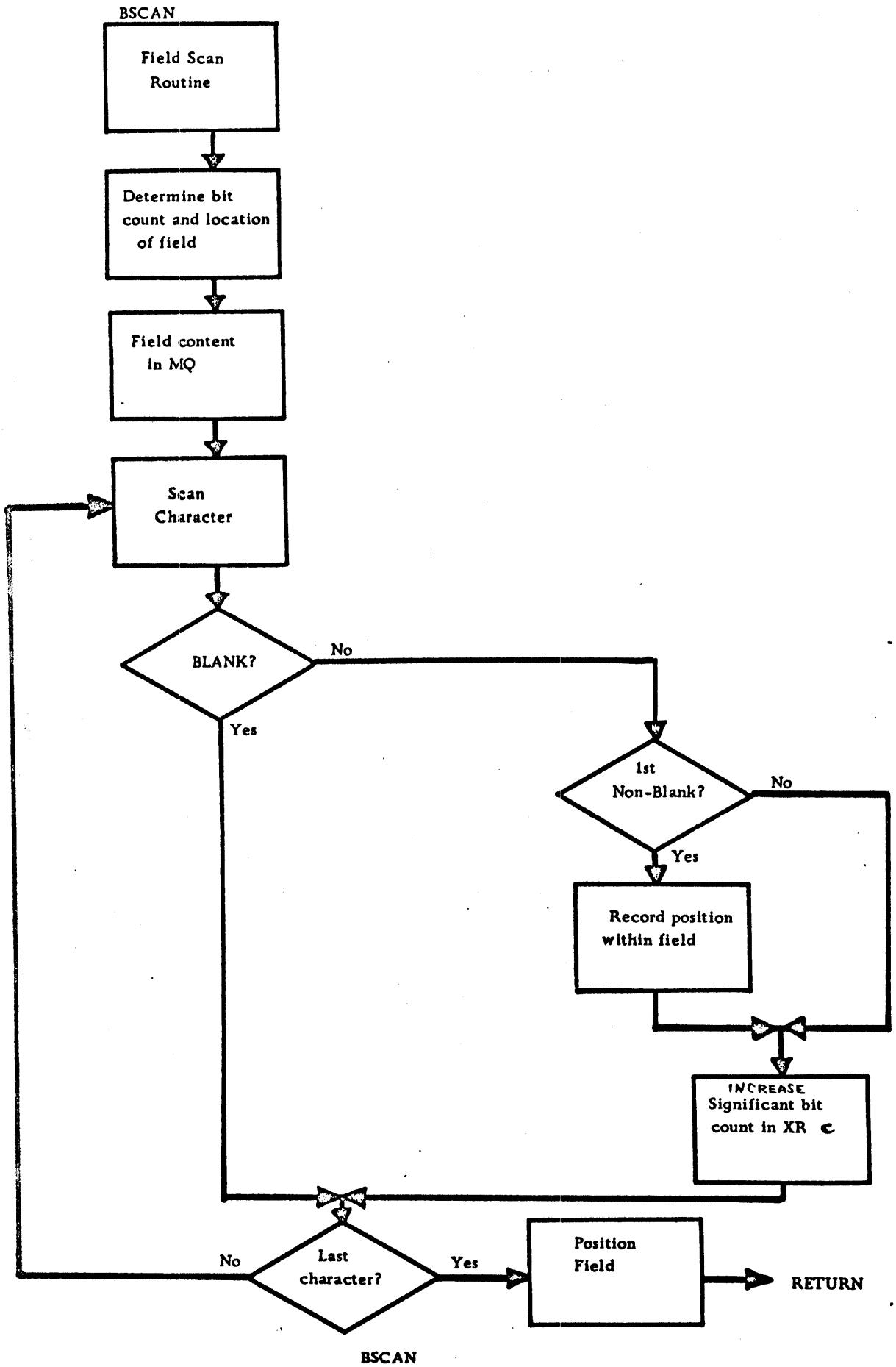

BJOB 3
Store File Block Origin for Loader
Store File Block Origin for C.P.
BSCAN
BSCAN
Examine Col. 33-35
Col. 33-35 numeric?
No
Yes
BERR
BERR 1
Print Error
FILE CNTL. BLOCK COUNT ERROR
BCONV
Convert BCD to Binary
Set Block Count to 50_{10}
Remember Count
BJOB 4

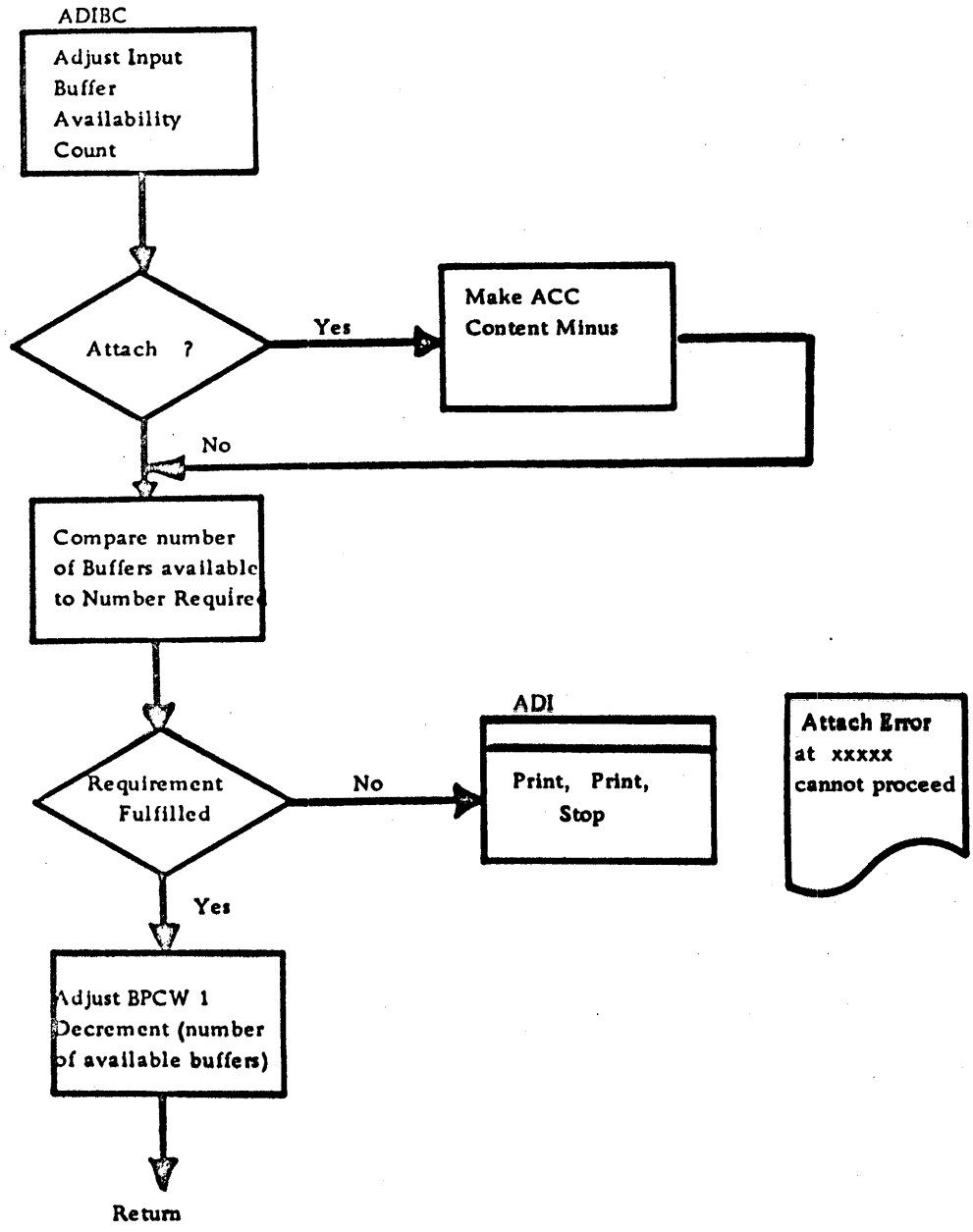
BJOB 3

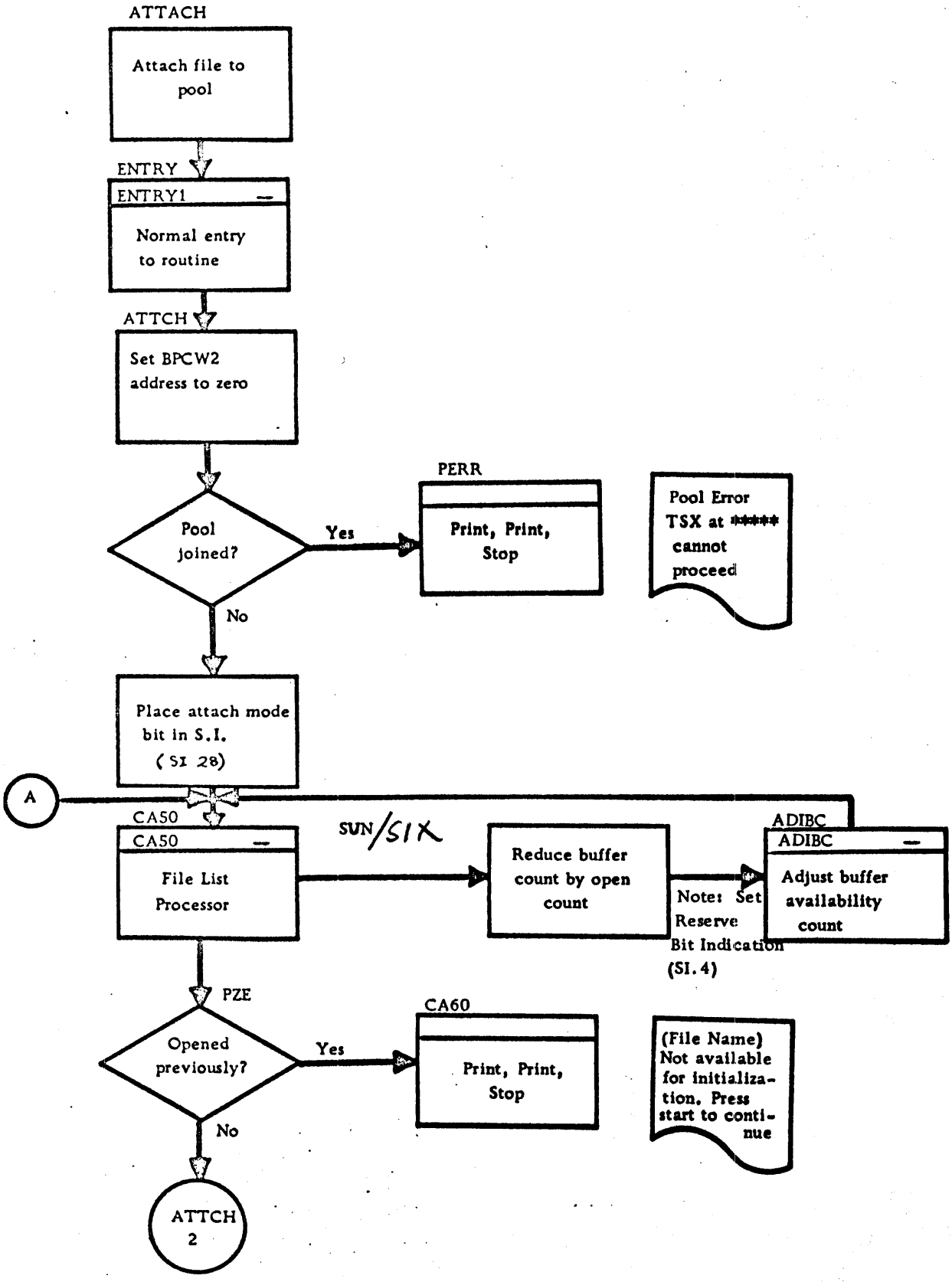

BJOB 4
Compute Words Needed
Store Size for Loader
Store Size for C.P.
Is Size too big?
Yes
No
FILE BLOCK WILL NOT FIT
BERR
BERR 1
Print Error
BJOB 2"1"
Compute end of Working File Block
Store Loc. for Loader
BJOB 5

BJOB 4

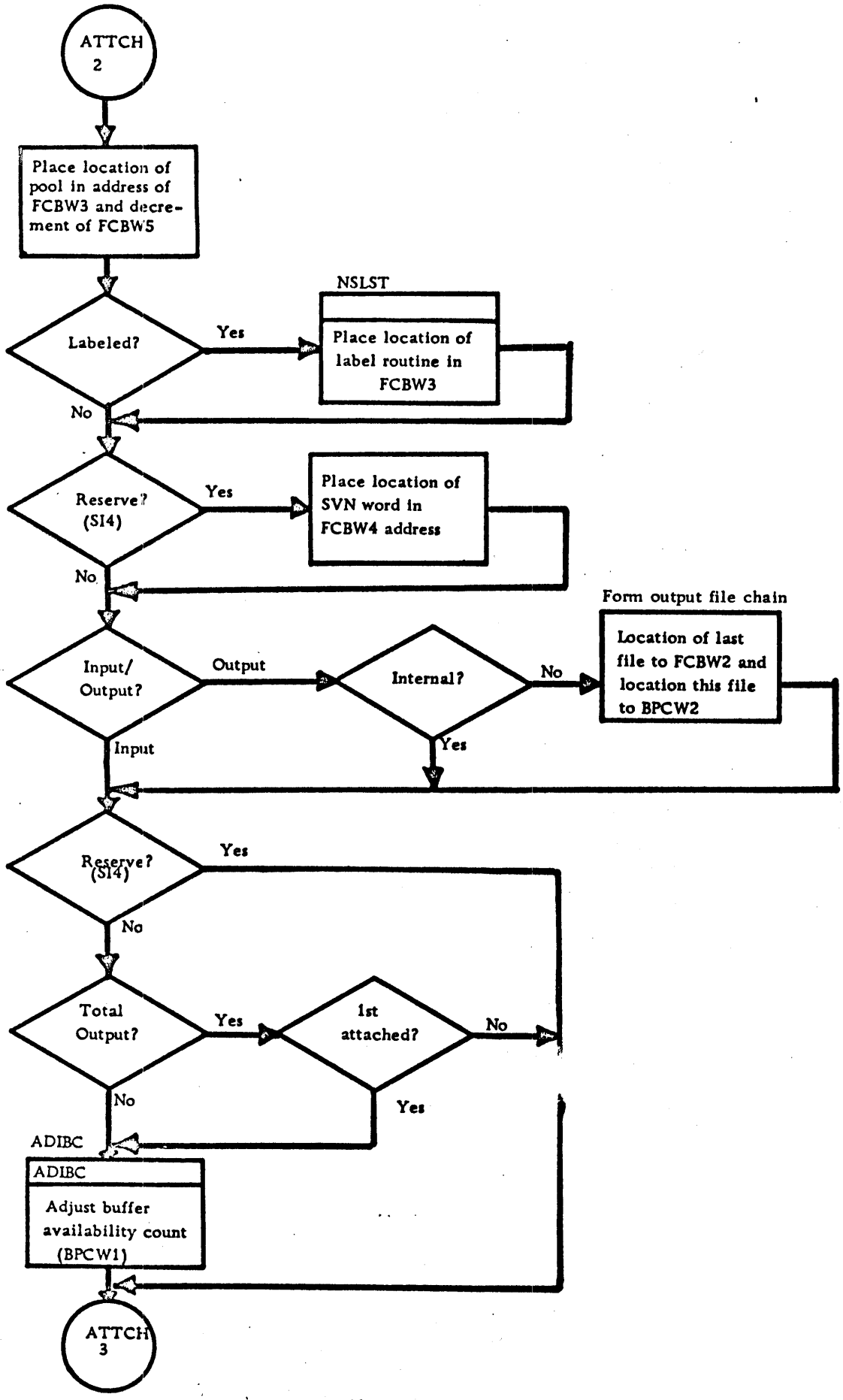

BJOB 5
Clear File Block
Was Job Card created?
Yes
No
Restore Buffer
BB00
BB00 1


BSCAN
Field Scan Routine
Determine bit count and location of field
Field content in MQ
Scan Character
BLANK?
No
Yes
1st Non-Blank?
No
Yes
Record position within field
INCREASE Significant bit count in XR c
Last character?
No
Yes
Position Field
RETURN
BSCAN

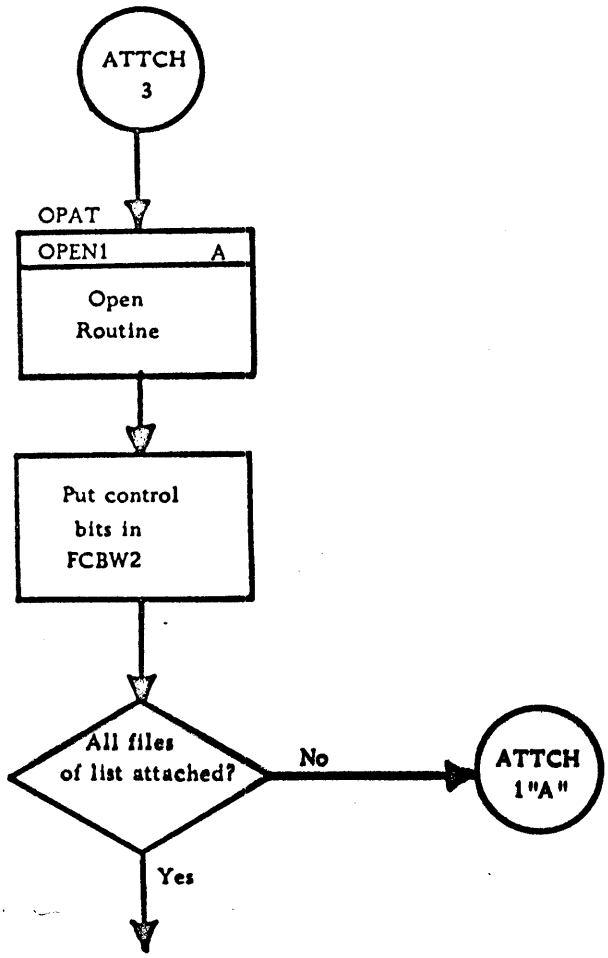

ADIBC
Adjust Input Buffer Availability Count
Attach ?
Yes
Make ACC Content Minus
No
Compare number of Buffers available to Number Required
Requirement Fulfilled
No
ADI
Print, Print, Stop
Attach Error at xxxxx cannot proceed
Yes
Adjust BPCW 1 Decrement (number of available buffers)
Return


ATTACH
Attach file to pool
ENTRY
ENTRY1
Normal entry to routine
ATTCH
Set BPCW2 address to zero
Pool joined?
Yes
PERR
Print, Print, Stop
Pool Error TSX at ***** cannot proceed
No
Place attach mode bit in S.I. (SI 28)
A
CA50
CA50
File List Processor
SUN/SIX
Reduce buffer count by open count
ADIBC
ADIBC
Adjust buffer availability count
Note: Set Reserve Bit Indication (SI.4)
PZE
Opened previously?
Yes
CA60
Print, Print, Stop
(File Name) Not available for initialization. Press start to continue
No
ATTCH 2

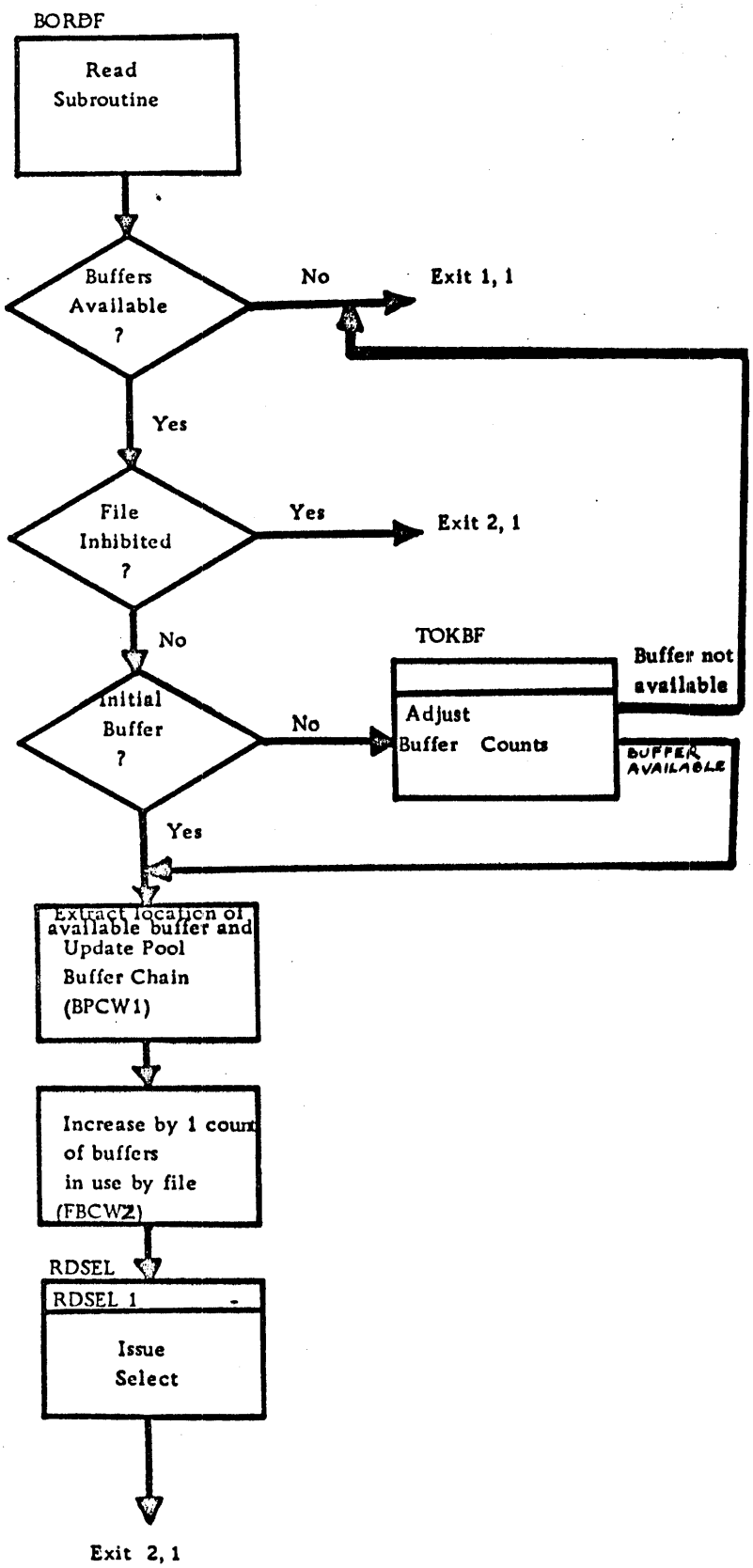
ATTCH 1

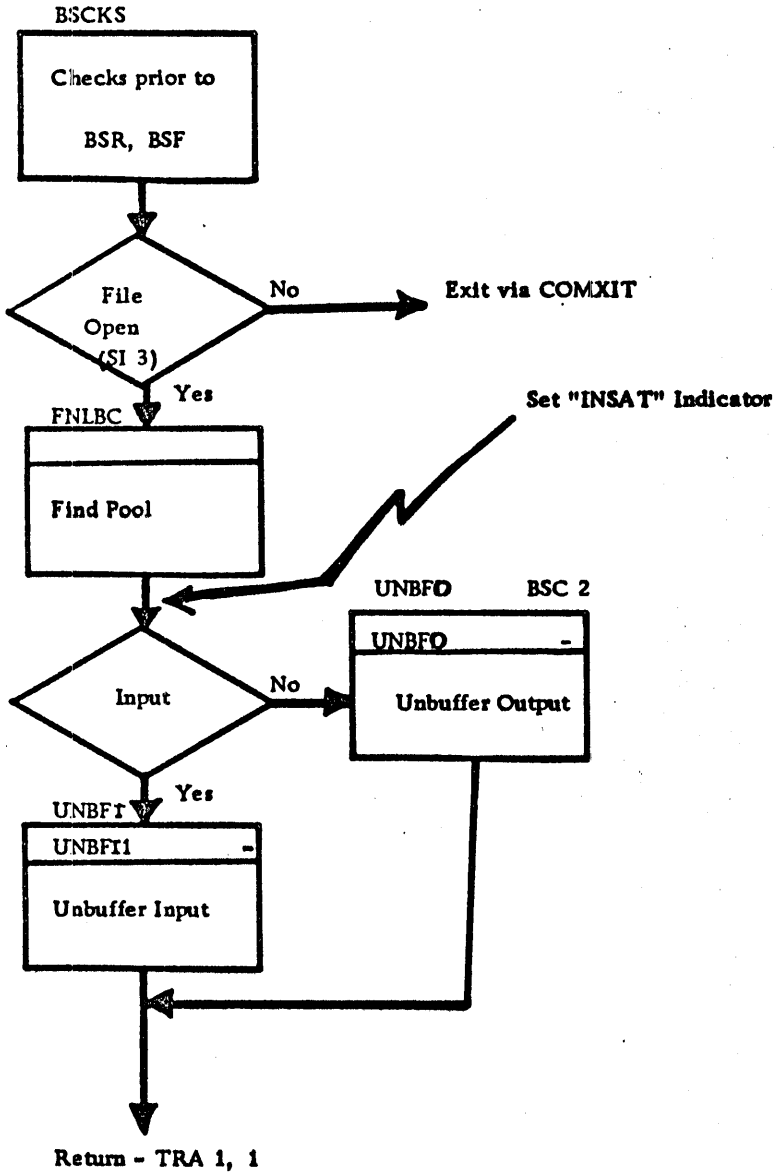

ATTCH 2
Place location of pool in address of FCBW3 and decrement of FCBW5
NSLST
Labeled?
Yes
Place location of label routine in FCBW3
No
Reserve? (SI4)
Yes
Place location of SVN word in FCBW4 address
No
Form output file chain
Input/ Output?
Output
Internal?
No
Location of last file to FCBW2 and location this file to BPCW2
Input
Yes
Reserve? (SI4)
Yes
No
Total Output?
Yes
1st attached?
No
No
Yes
ADIBC
ADIBC
Adjust buffer availability count (BPCW1)
ATTCH 3

ATTCH2

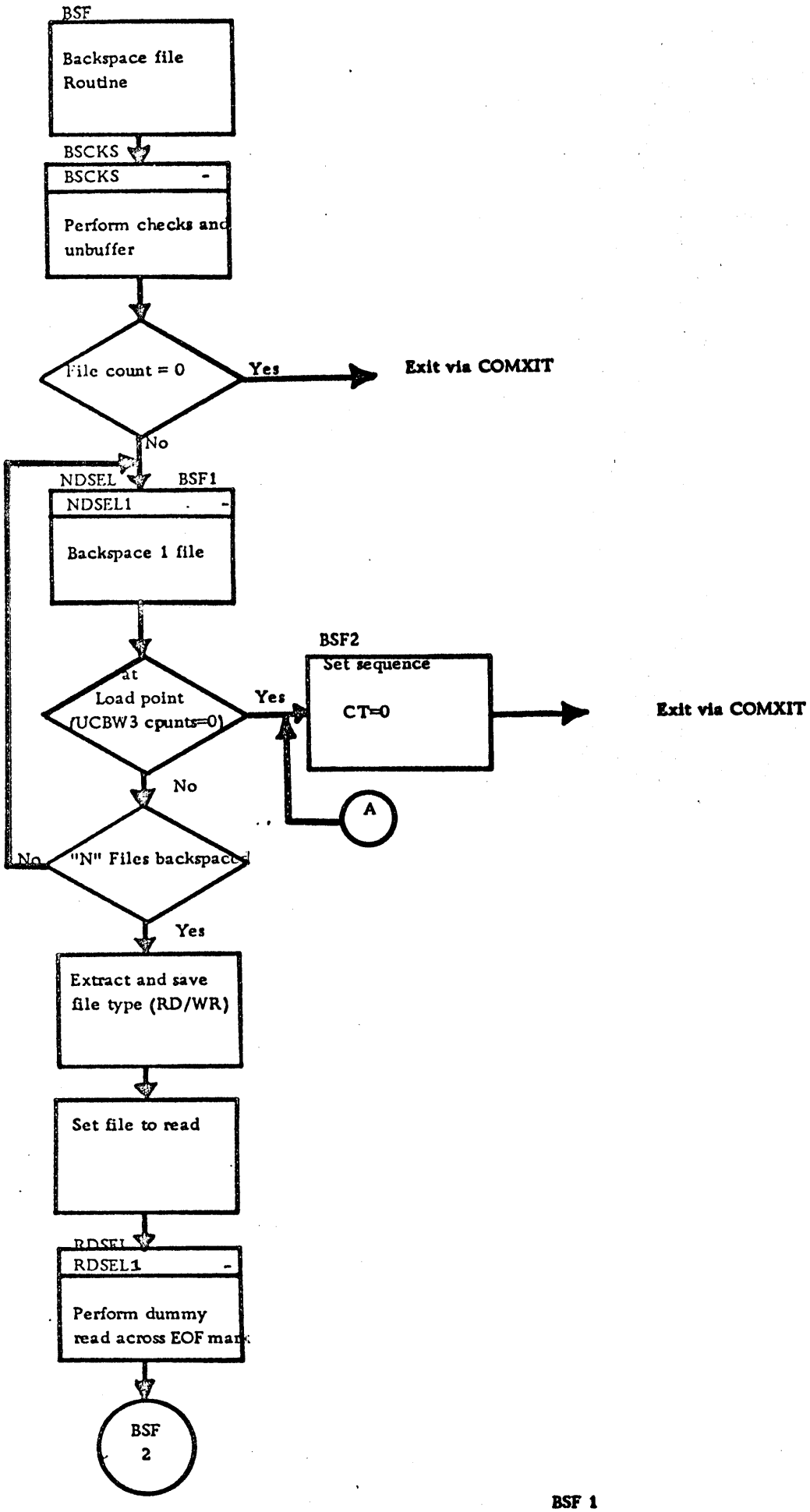


RETURN TO PROGRAM VIA "COMXIT" "A" ENTRY

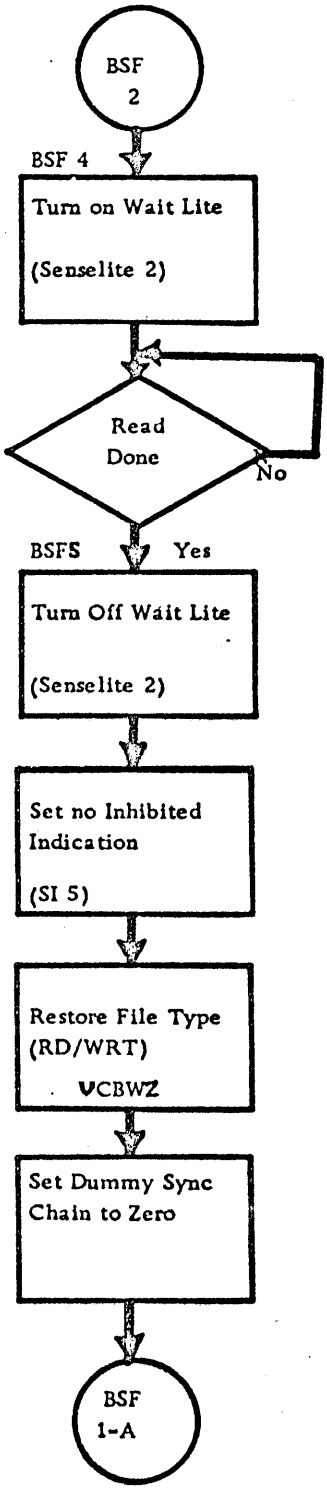

BORBF
Read Subroutine
Buffers Available ?
No
Exit 1, 1
Yes
File Inhibited ?
Yes
Exit 2, 1
No
Initial Buffer ?
No
TOKBF
Adjust Buffer Counts
Buffer not available
BUFFER AVAILABLE
Yes
Extract location of available buffer and Update Pool Buffer Chain (BPCW1)
Increase by 1 count of buffers in use by file (FBCW2)
RDSEL
RDSEL 1
Issue Select
Exit 2, 1

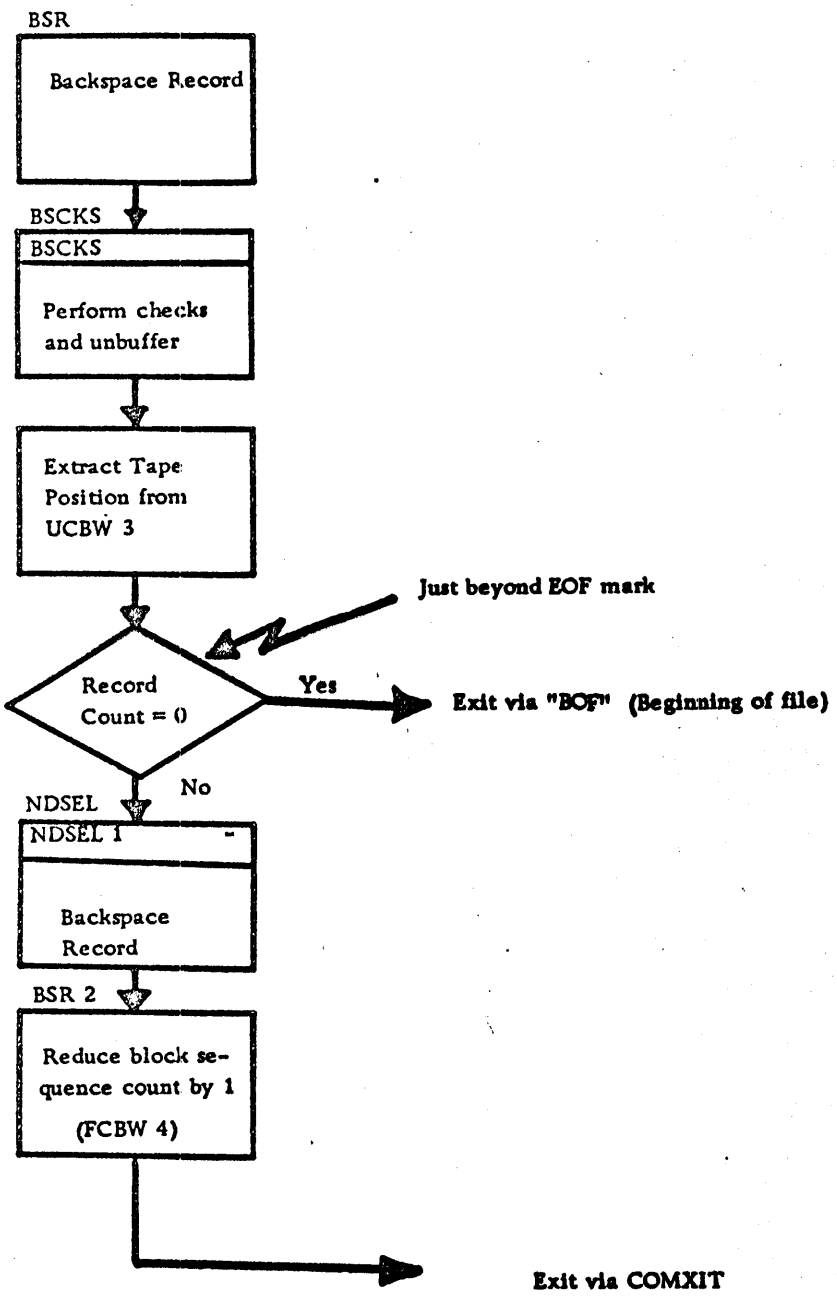

BSCKS
Checks prior to
BSR, BSF
File
Open
(SI 3)
No
Exit via COMXIT
Yes
FNLBC
Find Pool
Set "INSAT" Indicator
UNBFO
BSC 2
UNBFO
Unbuffer Output
Input
No
Yes
UNBFI
UNBFI1
Unbuffer Input
Return - TRA 1, 1

BSCKS

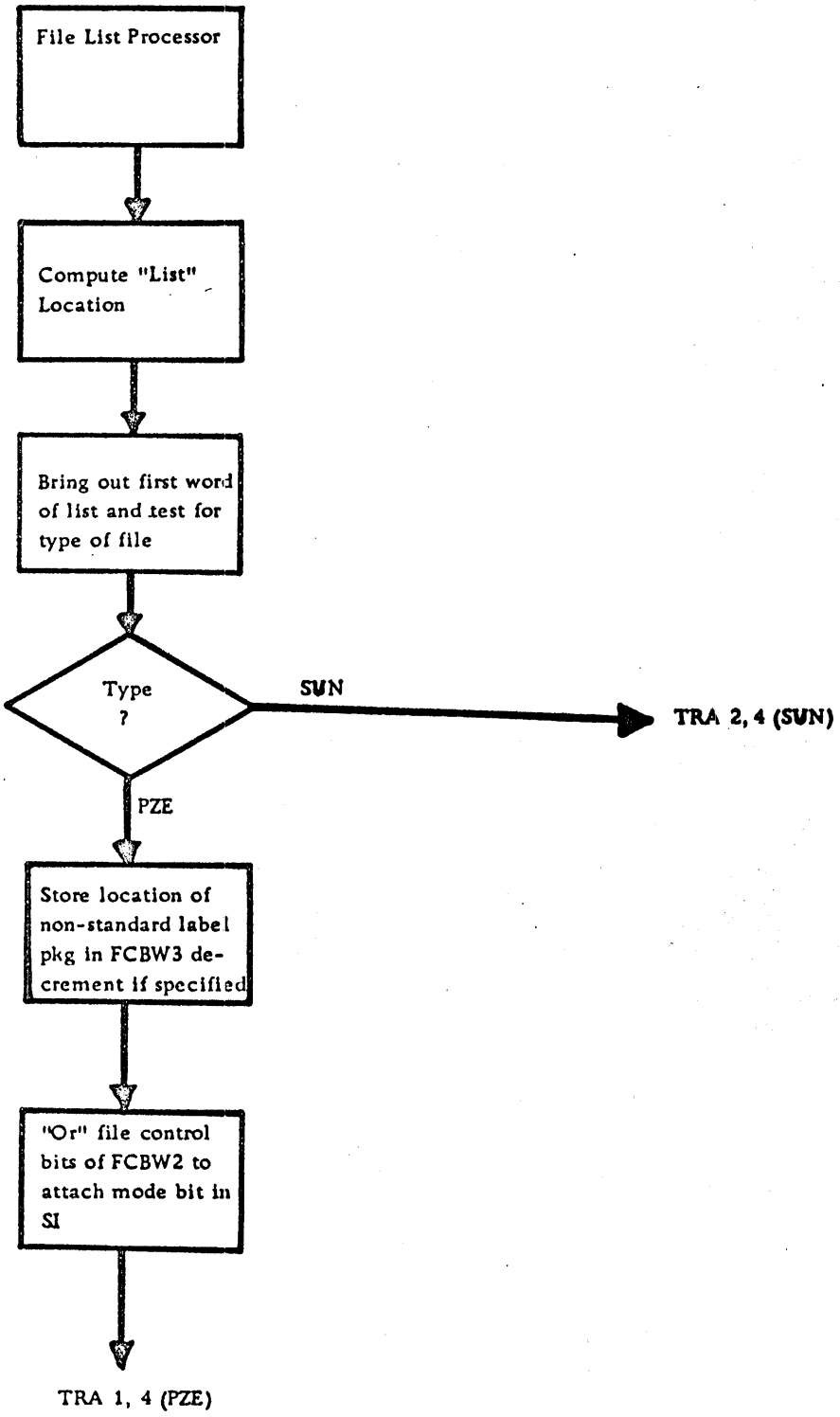

BSF
Backspace file Routine
BSCKS
BSCKS -
Perform checks and unbuffer
File count = 0
Yes
Exit via COMXIT
No
NDSEL
BSF1
NDSEL1 -
Backspace 1 file
BSF2
Set sequence
at Load point (UCBW3 cpunts=0)
Yes
CT=0
Exit via COMXIT
No
A
No
"N" Files backspaced
Yes
Extract and save file type (RD/WR)
Set file to read
RDSEL
RDSEL1 -
Perform dummy read across EOF mark
BSF 2

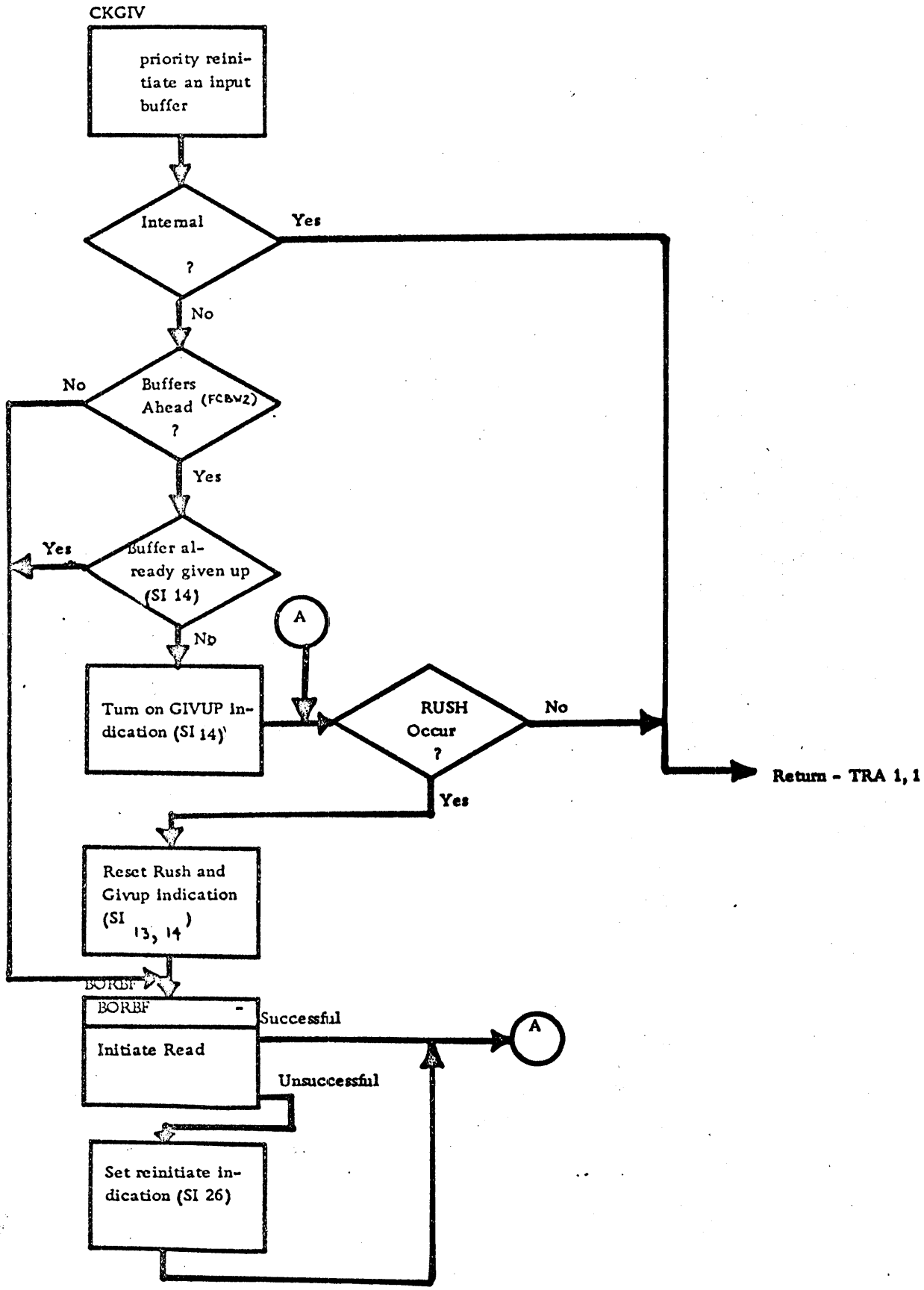
BSF 1

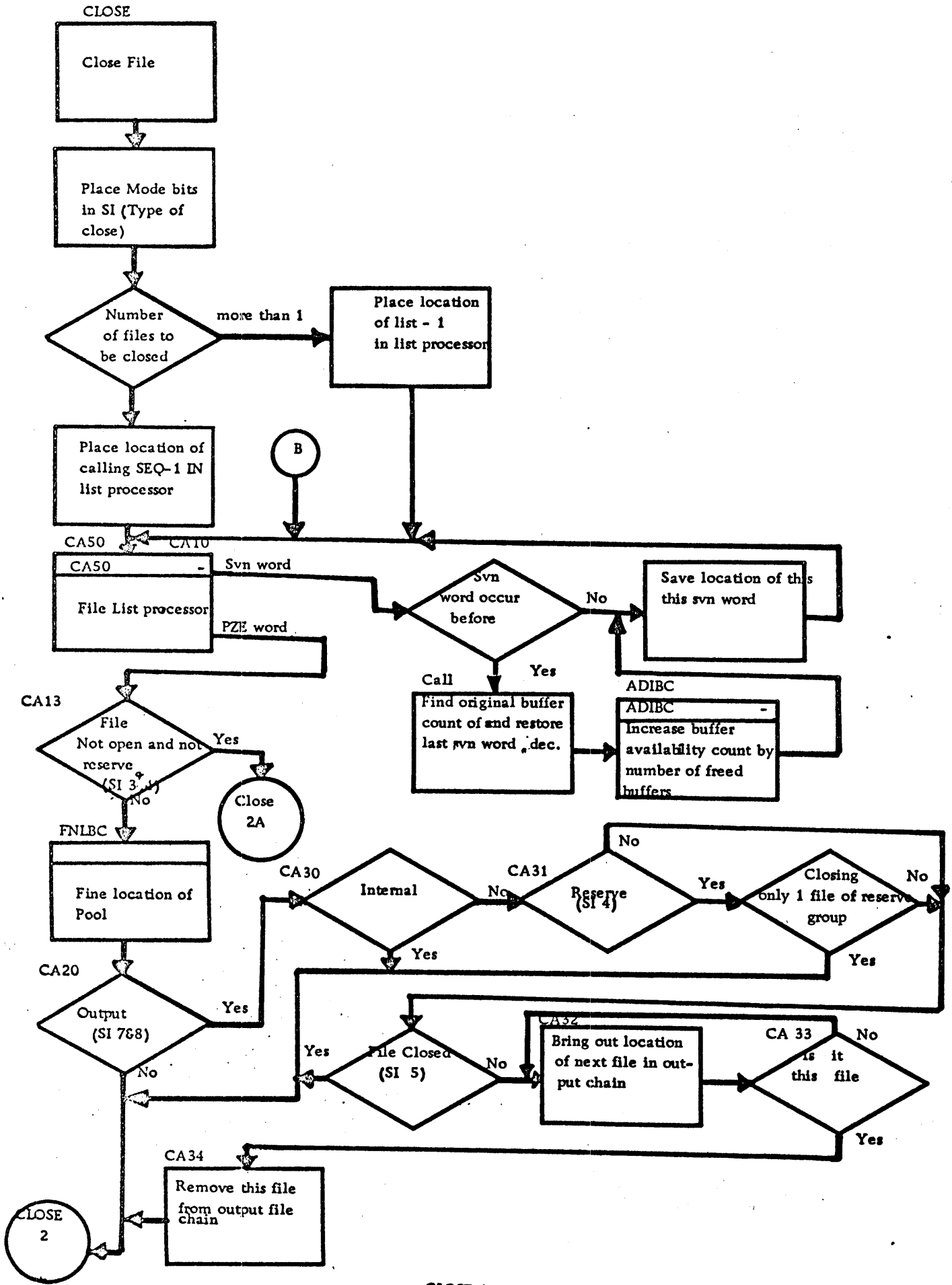

BSF
2
BSF 4
Turn on Wait Lite
(Senselite 2)
Read
Done
No
BSF5
Yes
Turn Off Wait Lite
(Senselite 2)
Set no Inhibited
Indication
(SI 5)
Restore File Type
(RD/WRT)
VCBWZ
Set Dummy Sync
Chain to Zero
BSF
1-A


BSR
Backspace Record
BSCKS
BSCKS
Perform checks and unbuffer
Extract Tape Position from UCBW 3
Just beyond EOF mark
Record Count = 0
Yes
Exit via "BOF" (Beginning of file)
No
NDSEL
NDSEL 1
Backspace Record
BSR 2
Reduce block sequence count by 1 (FCBW 4)
Exit via COMXIT

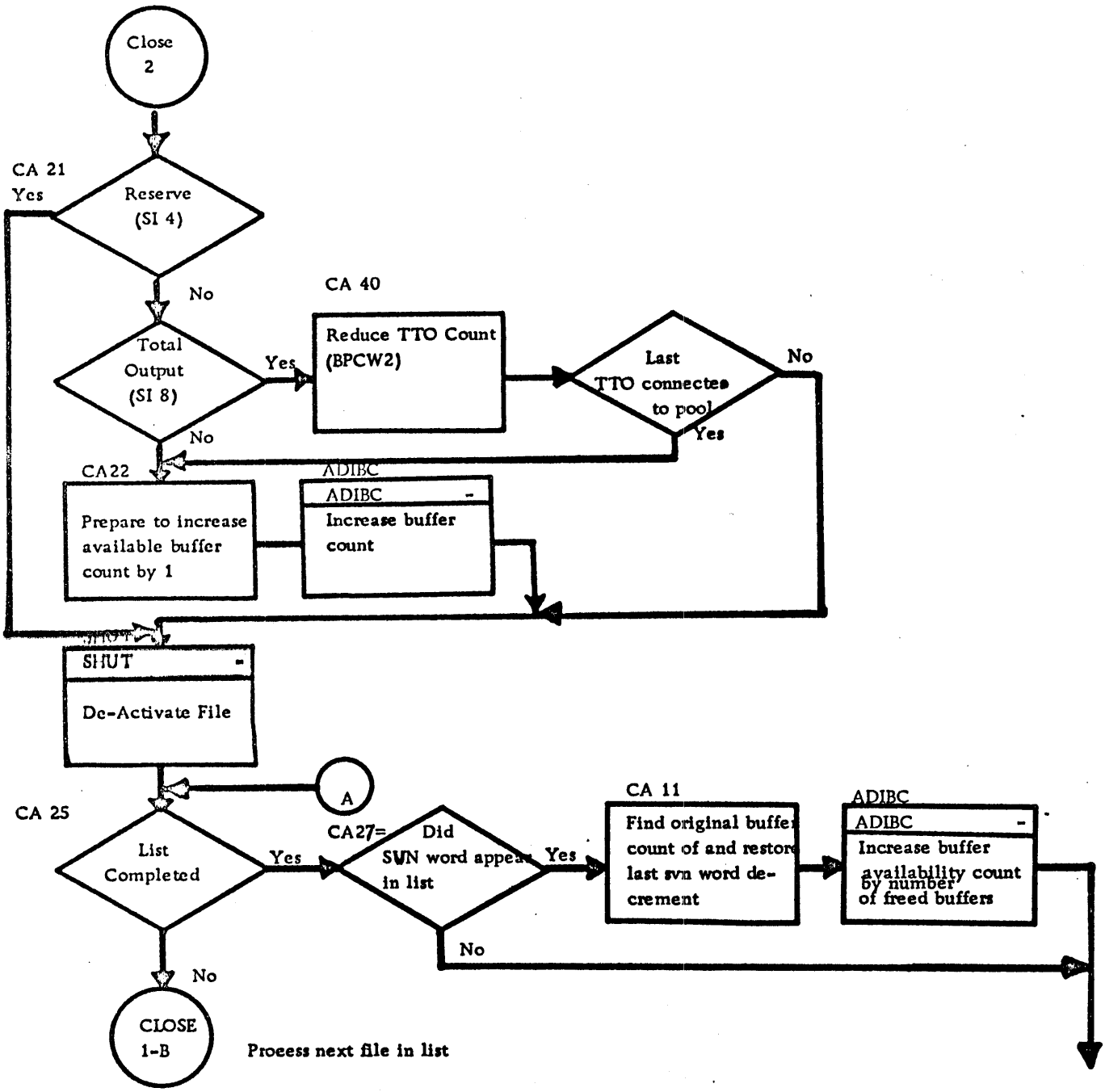
CA50

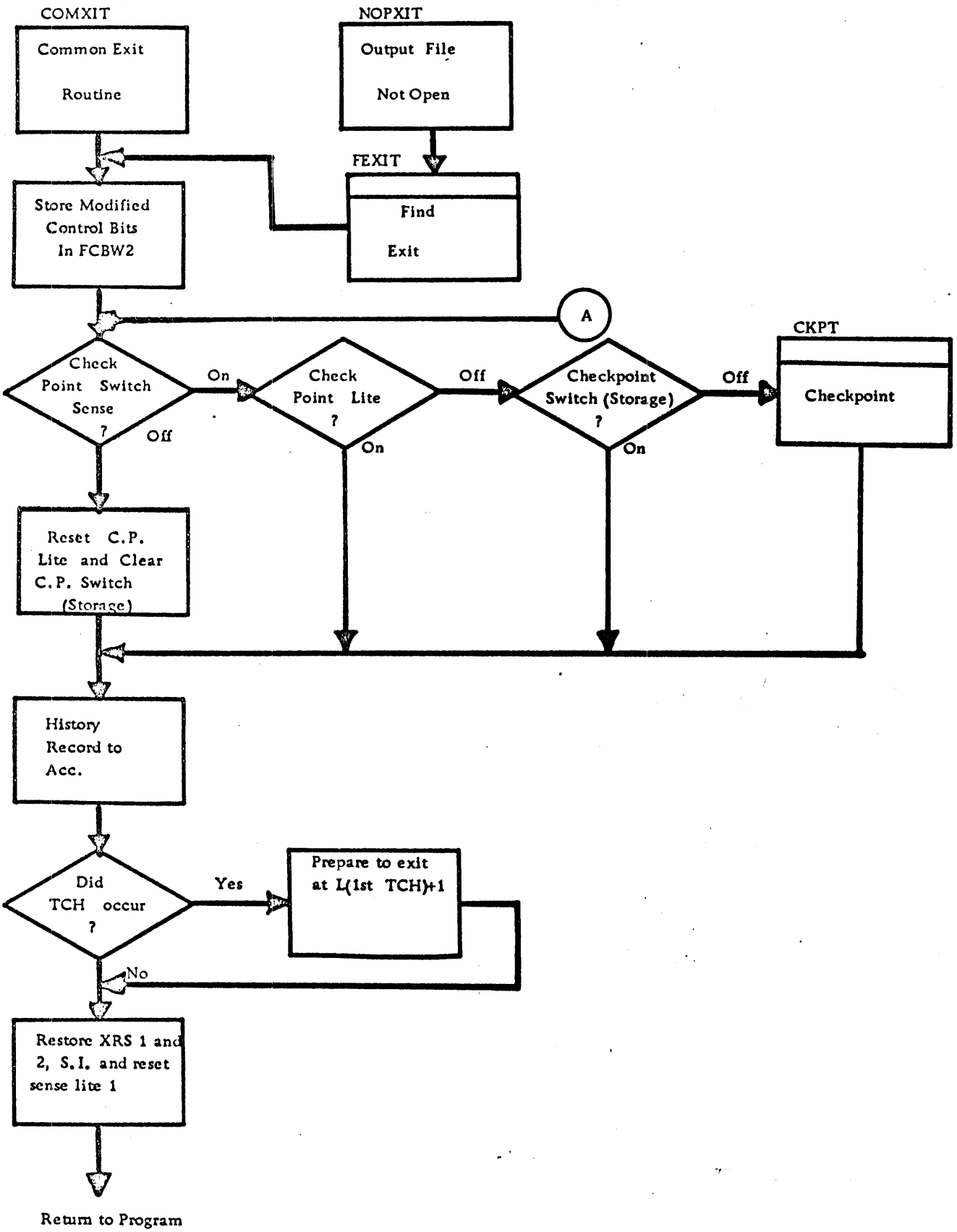

File List Processor
Compute "List" Location
Bring out first word of list and test for type of file
Type ?
SUN
TRA 2, 4 (SUN)
PZE
Store location of non-standard label pkg in FCBW3 decrement if specified
"Or" file control bits of FCBW2 to attach mode bit in SI
TRA 1, 4 (PZE)

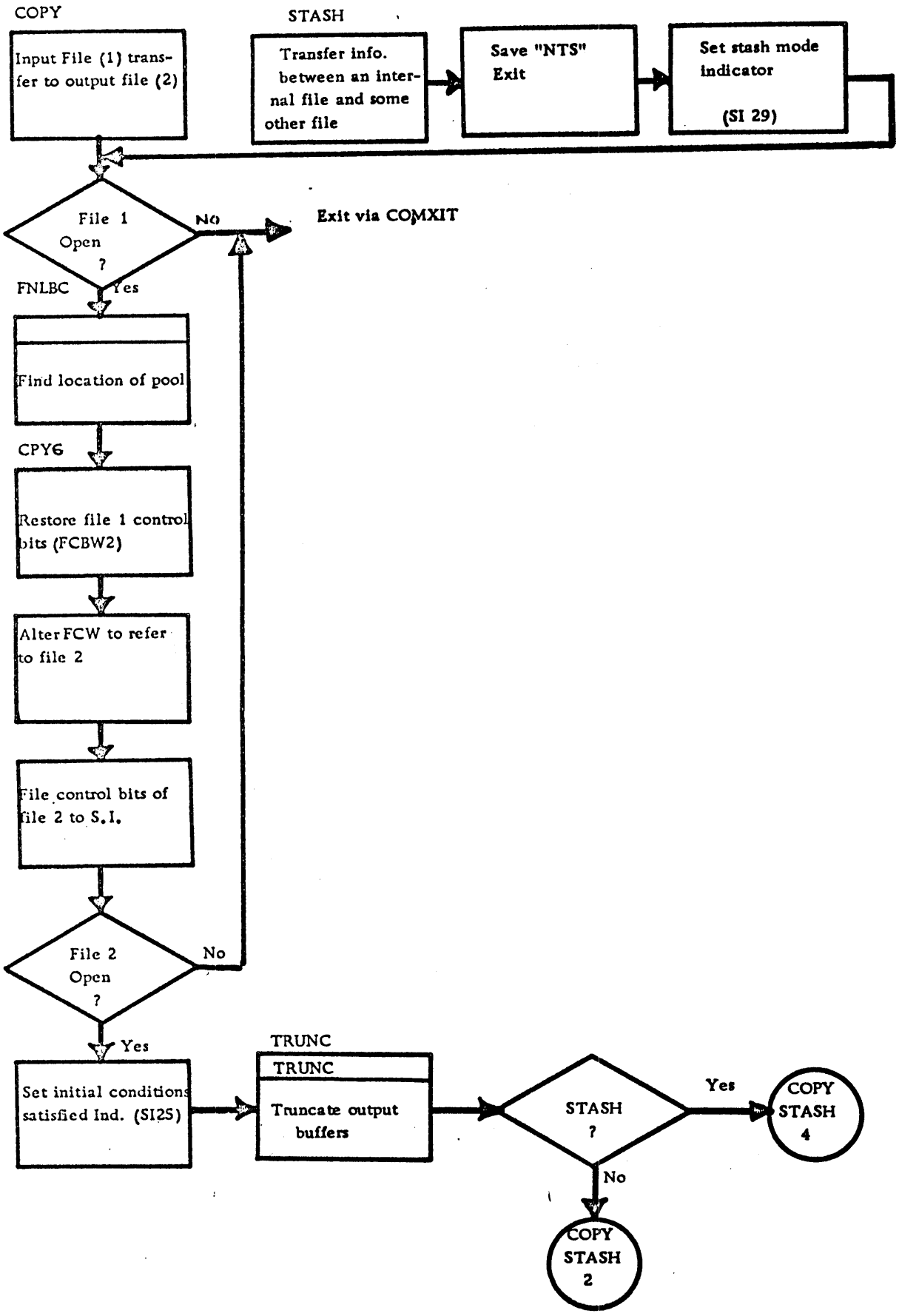

CKGIV
priority reinitiate an input buffer
Internal ?
Yes
No
Buffers Ahead (FCBW2) ?
No
Yes
Buffer already given up (SI 14)
Yes
No
A
Turn on GIVUP indication (SI 14)
RUSH Occur ?
No
Return - TRA 1, 1
Yes
Reset Rush and Givup indication (SI 13, 14)
BORBF
BORBF -
Initiate Read
Successful
Unsuccessful
A
Set reinitiate indication (SI 26)


CLOSE
Close File
Place Mode bits in SI (Type of close)
Number of files to be closed
more than 1
Place location of list - 1 in list processor
Place location of calling SEQ-1 IN list processor
B
CA50
CA10
CA50 -
File List processor
Svn word
PZE word
Svn word occur before
No
Yes
Save location of this this svn word
Call
Find original buffer count of and restore last svn word dec.
ADIBC
ADIBC -
Increase buffer availability count by number of freed buffers
CA13
File Not open and not reserve (SI 3 4)
Yes
No
Close 2A
FNLBC
Fine location of Pool
CA30
Internal
No
Yes
CA31
Reserve (SI 4)
No
Yes
Closing only 1 file of reserve group
No
Yes
CA20
Output (SI 7&8)
Yes
No
File Closed (SI 5)
Yes
No
CA32
Bring out location of next file in output chain
CA 33
Is it this file
No
Yes
CA34
Remove this file from output file chain
CLOSE 2

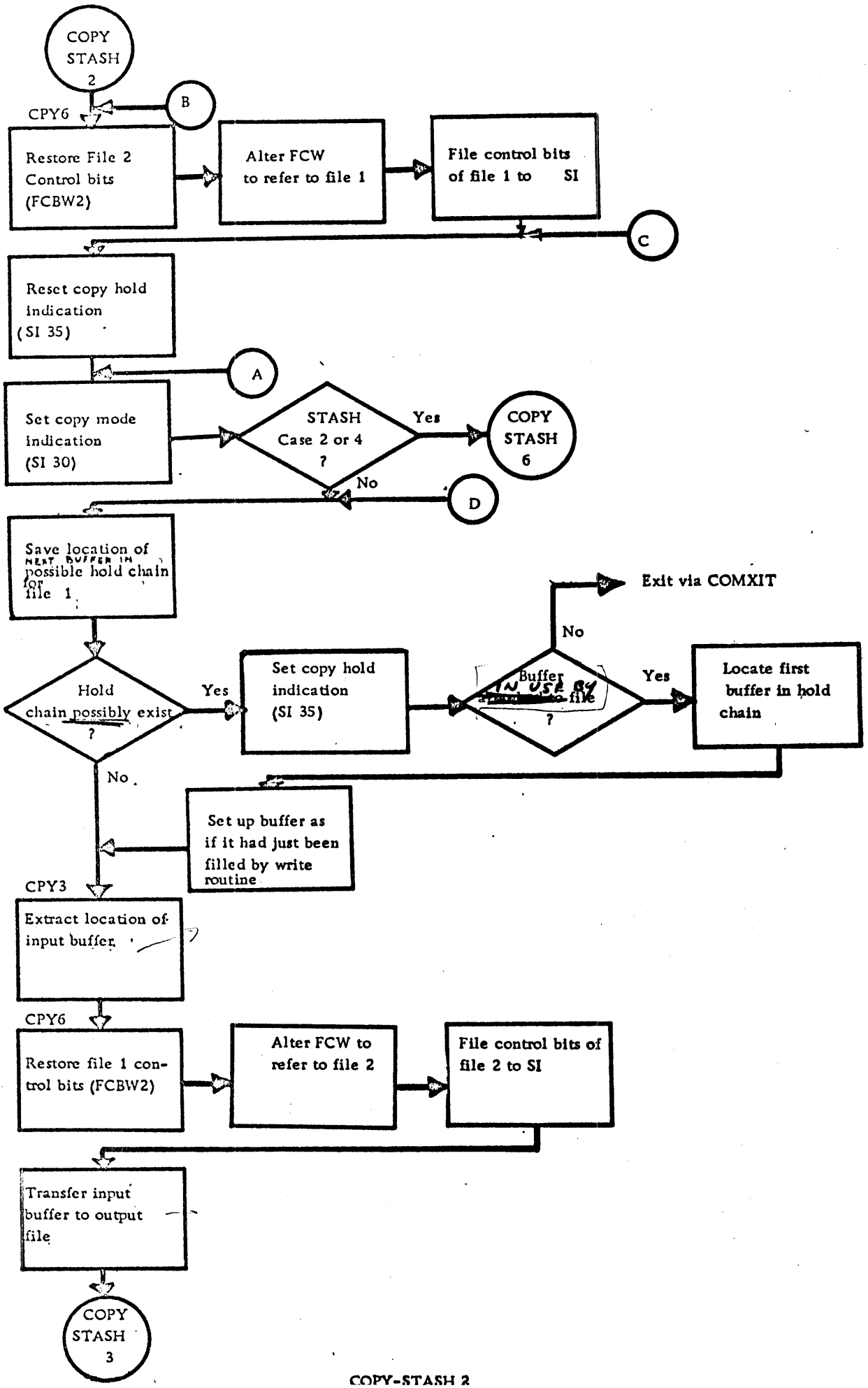
CLOSE 1


Close 2
CA 21
Yes
Reserve (SI 4)
No
Total Output (SI 8)
Yes
CA 40
Reduce TTO Count (BPCW2)
Last TTO connected to pool
No
Yes
No
CA22
Prepare to increase available buffer count by 1
ADIBC
ADIBC -
Increase buffer count
SHUT -
De-Activate File
A
CA 25
List Completed
Yes
CA27=
Did SVN word appear in list
Yes
CA 11
Find original buffer count of and restore last svn word de-crement
ADIBC
ADIBC -
Increase buffer availability count by number of freed buffers
No
No
CLOSE 1-B
Proeess next file in list
Exit via COMXIT

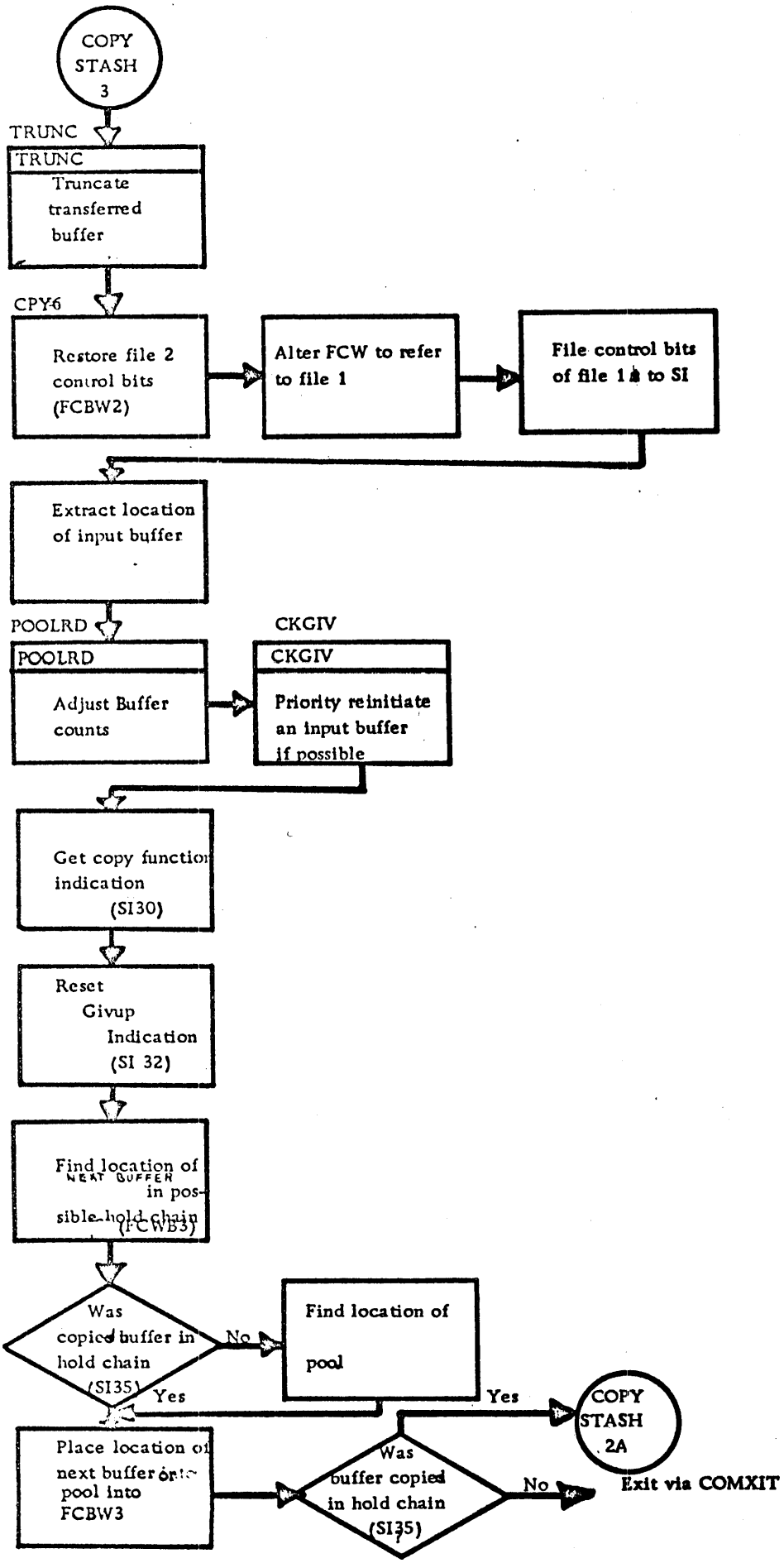

COMXIT
Common Exit Routine
NOPXIT
Output File Not Open
FEXIT
Find Exit
Store Modified Control Bits In FCBW2
A
CKPT
Check Point Switch Sense ?
On
Off
Check Point Lite ?
Off
On
Checkpoint Switch (Storage) ?
Off
On
Checkpoint
Reset C.P. Lite and Clear C.P. Switch (Storage)
History Record to Acc.
Did TCH occur ?
Yes
Prepare to exit at L(1st TCH)+1
No
Restore XRS 1 and 2, S.I. and reset sense lite 1
Return to Program


COPY
Input File (1) transfer to output file (2)
STASH
Transfer info. between an internal file and some other file
Save "NTS" Exit
Set stash mode indicator
(SI 29)
File 1 Open ?
NO
Exit via COMXIT
Yes
FNLBC
Find location of pool
CPY6
Restore file 1 control bits (FCBW2)
Alter FCW to refer to file 2
File control bits of file 2 to S.I.
File 2 Open ?
No
Yes
Set initial conditions satisfied Ind. (SI25)
TRUNC
TRUNC
Truncate output buffers
STASH ?
Yes
COPY STASH 4
No
COPY STASH 2

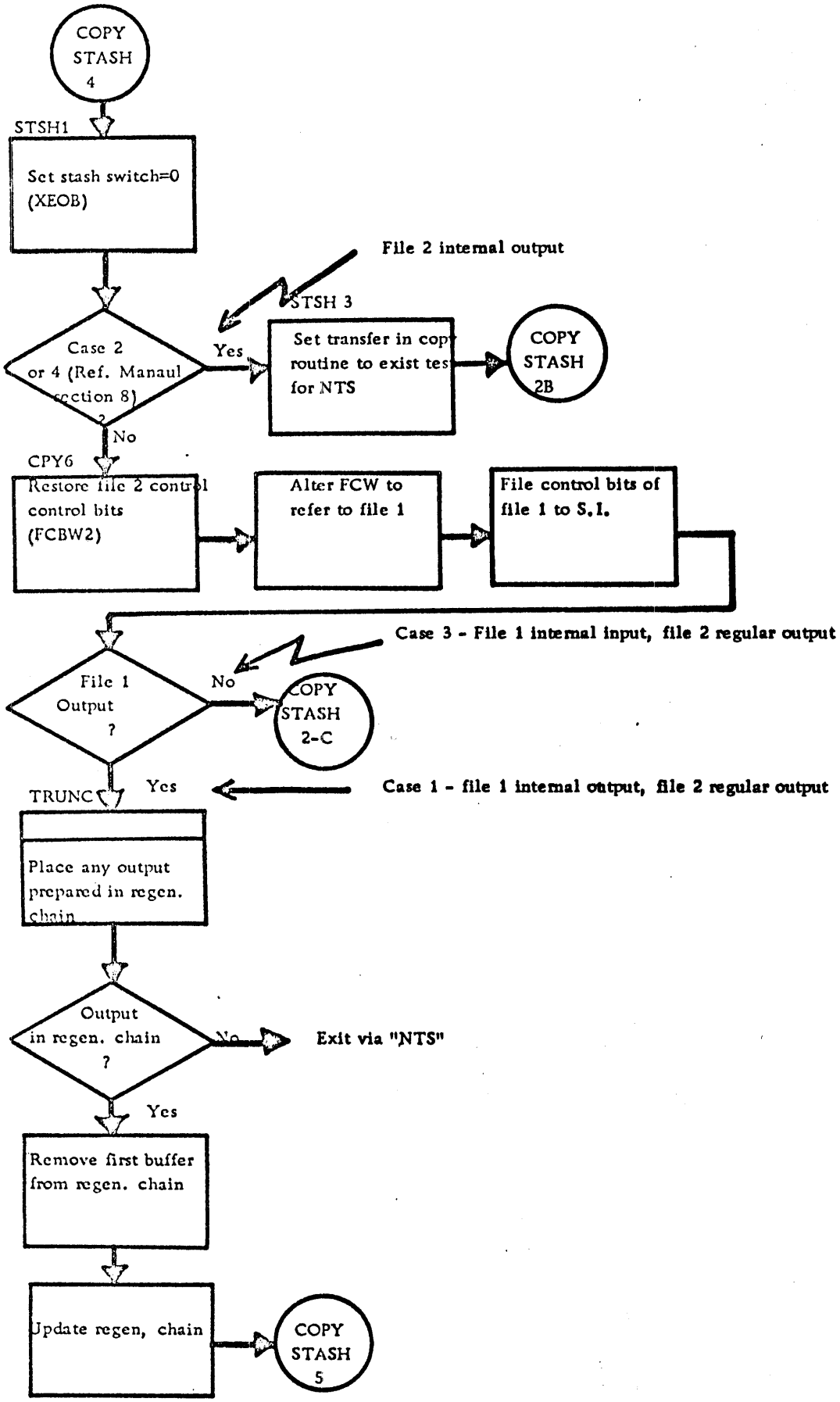
COPY - STASH 1


COPY STASH 2
B
CPY6
Restore File 2 Control bits (FCBW2)
Alter FCW to refer to file 1
File control bits of file 1 to SI
C
Reset copy hold indication (SI 35)
A
Set copy mode indication (SI 30)
STASH Case 2 or 4 ?
Yes
COPY STASH 6
No
D
Save location of NEXT BUFFER IN possible hold chain for file 1
Exit via COMXIT
No
Hold chain possibly exist ?
Yes
Set copy hold indication (SI 35)
Buffer IN USE BY file ?
Yes
Locate first buffer in hold chain
No
Set up buffer as if it had just been filled by write routine
CPY3
Extract location of input buffer
CPY6
Restore file 1 control bits (FCBW2)
Alter FCW to refer to file 2
File control bits of file 2 to SI
Transfer input buffer to output file
COPY STASH 3

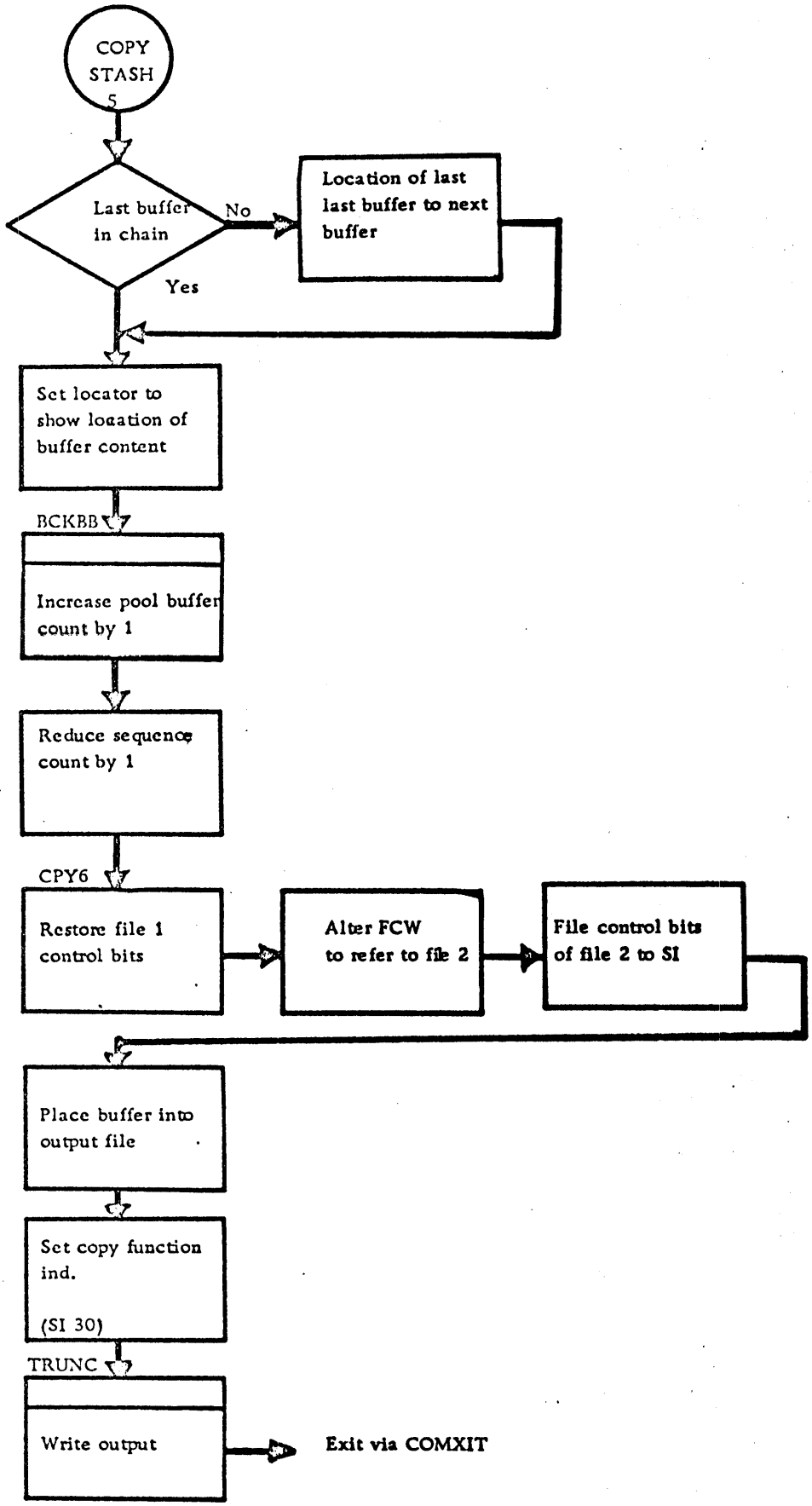
COPY-STASH 2


COPY STASH 3
TRUNC
TRUNC
Truncate transferred buffer
CPY6
Restore file 2 control bits (FCBW2)
Alter FCW to refer to file 1
File control bits of file 1 to SI
Extract location of input buffer
POOLRD
POOLRD
Adjust Buffer counts
CKGIV
CKGIV
Priority reinitiate an input buffer if possible
Get copy function indication (SI30)
Reset Givup Indication (SI 32)
Find location of NEXT BUFFER in possible hold chain (FCWB3)
Was copied buffer in hold chain (SI35)
No
Find location of pool
Yes
Yes
COPY STASH 2A
Place location of next buffer or pool into FCBW3
Was buffer copied in hold chain (SI35)
No
Exit via COMXIT

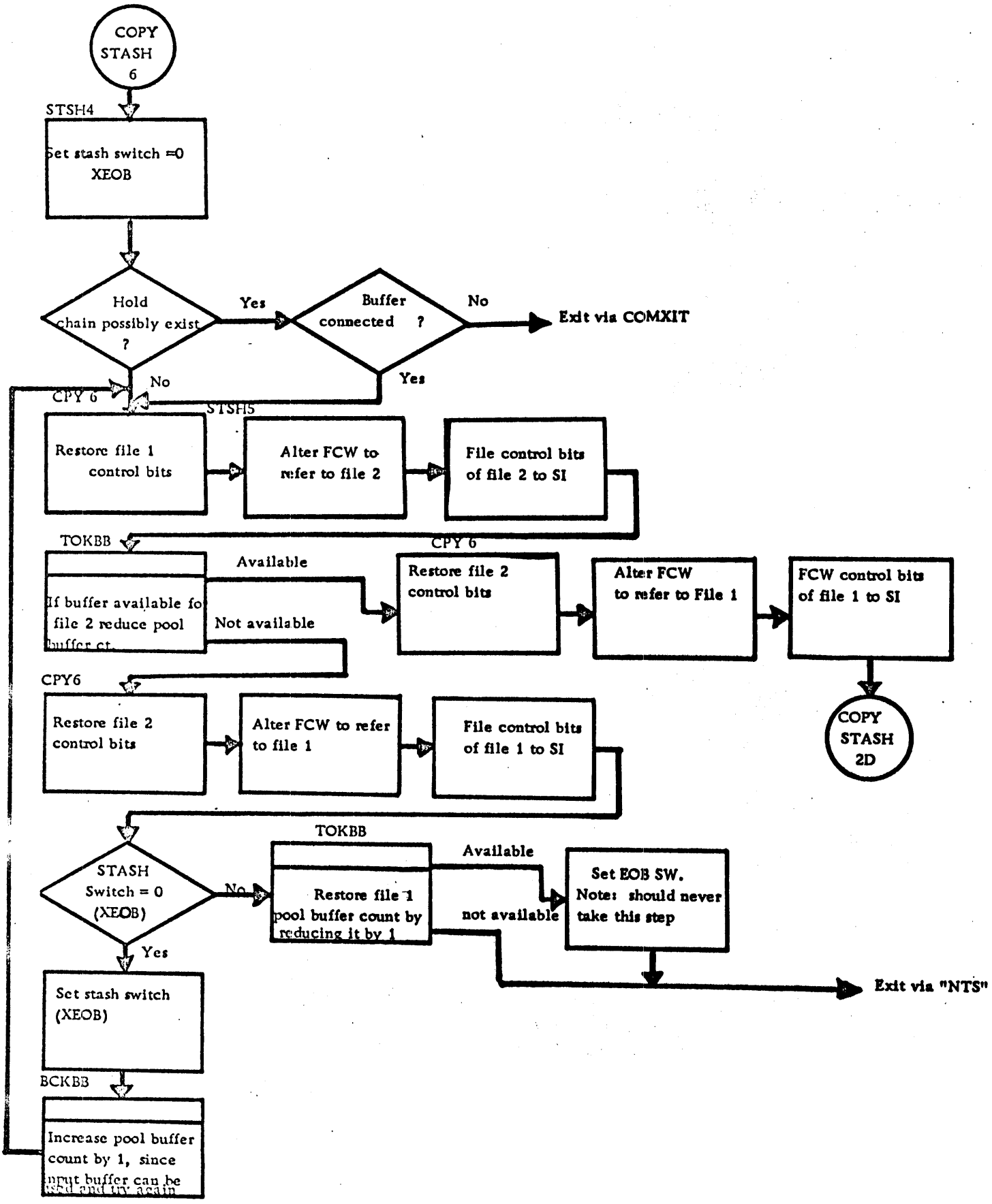
COPY - STASH 3

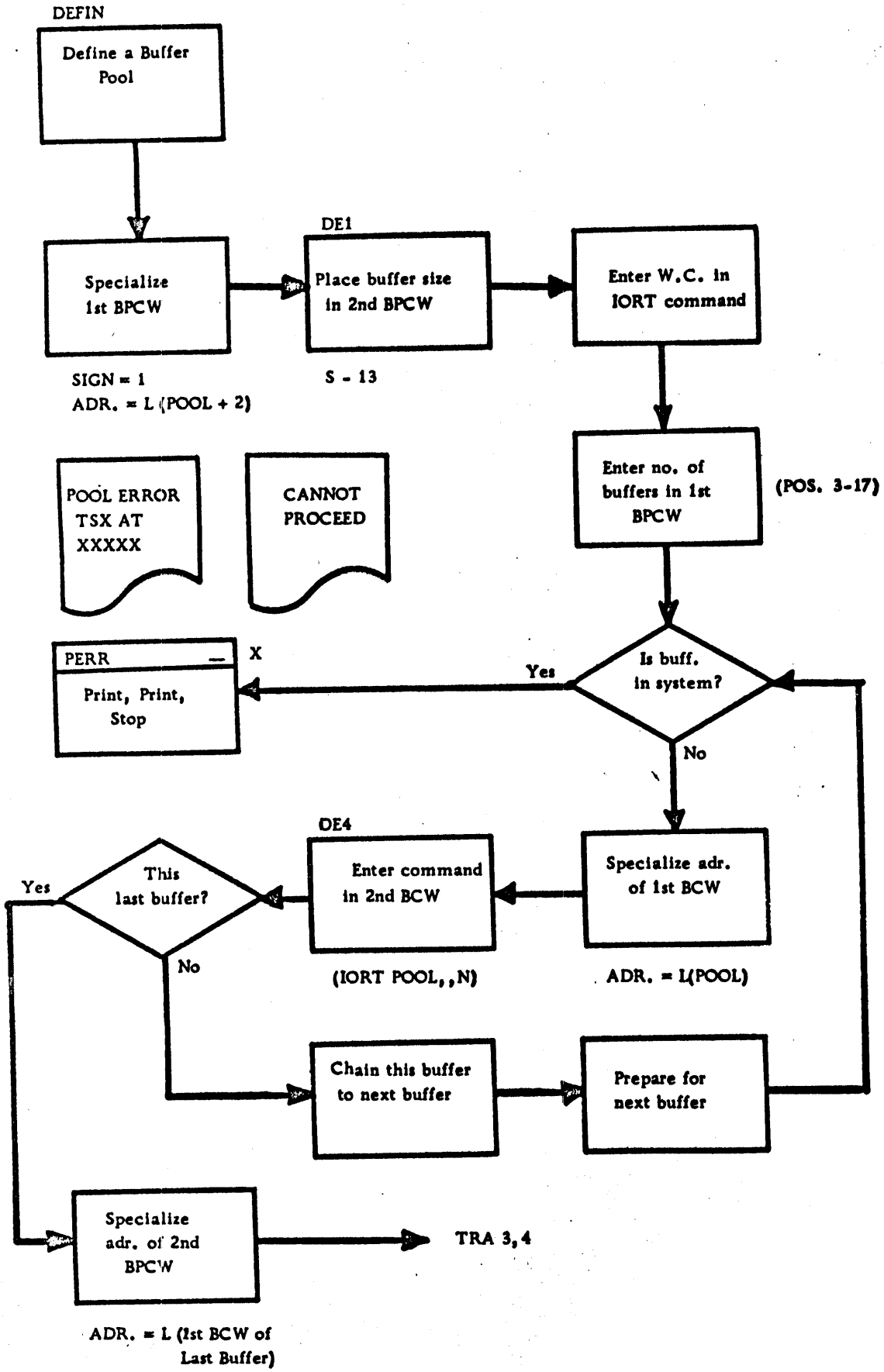

COPY STASH 4
STSH1
Set stash switch=0 (XEOB)
File 2 internal output
STSH 3
Case 2 or 4 (Ref. Manaul section 8)
Yes
Set transfer in copy routine to exist test for NTS
COPY STASH 2B
No
CPY6
Restore file 2 control control bits (FCBW2)
Alter FCW to refer to file 1
File control bits of file 1 to S.I.
Case 3 - File 1 internal input, file 2 regular output
File 1 Output ?
No
COPY STASH 2-C
TRUNC
Yes
Case 1 - file 1 internal output, file 2 regular output
Place any output prepared in regen. chain
Output in regen. chain ?
No
Exit via "NTS"
Yes
Remove first buffer from regen. chain
Update regen, chain
COPY STASH 5

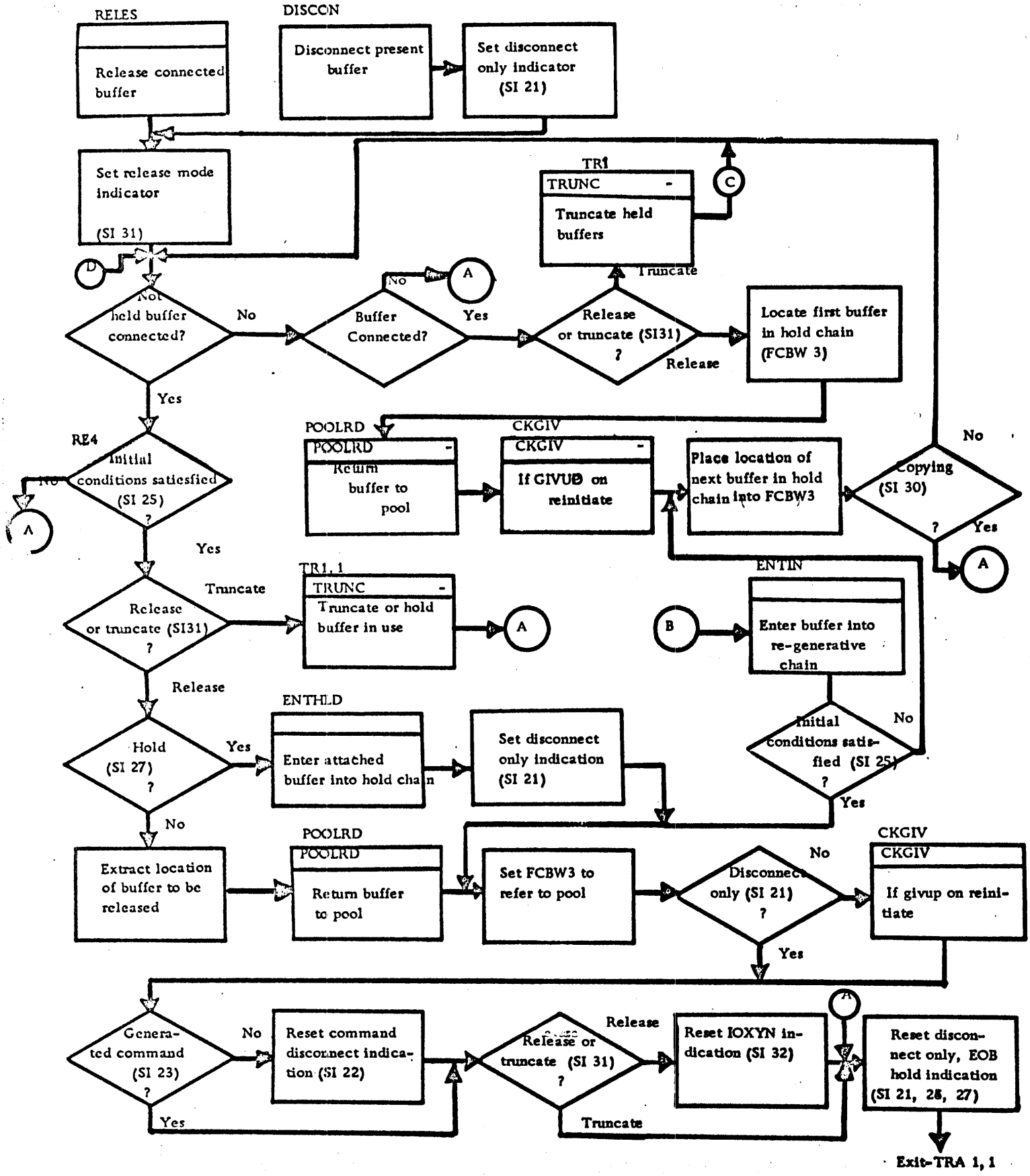
COPY - STASH 4


COPY STASH 5
Last buffer in chain
No
Location of last last buffer to next buffer
Yes
Set locator to show location of buffer content
BCKBB
Increase pool buffer count by 1
Reduce sequence count by 1
CPY6
Restore file 1 control bits
Alter FCW to refer to file 2
File control bits of file 2 to SI
Place buffer into output file
Set copy function ind.
(SI 30)
TRUNC
Write output
Exit via COMXIT

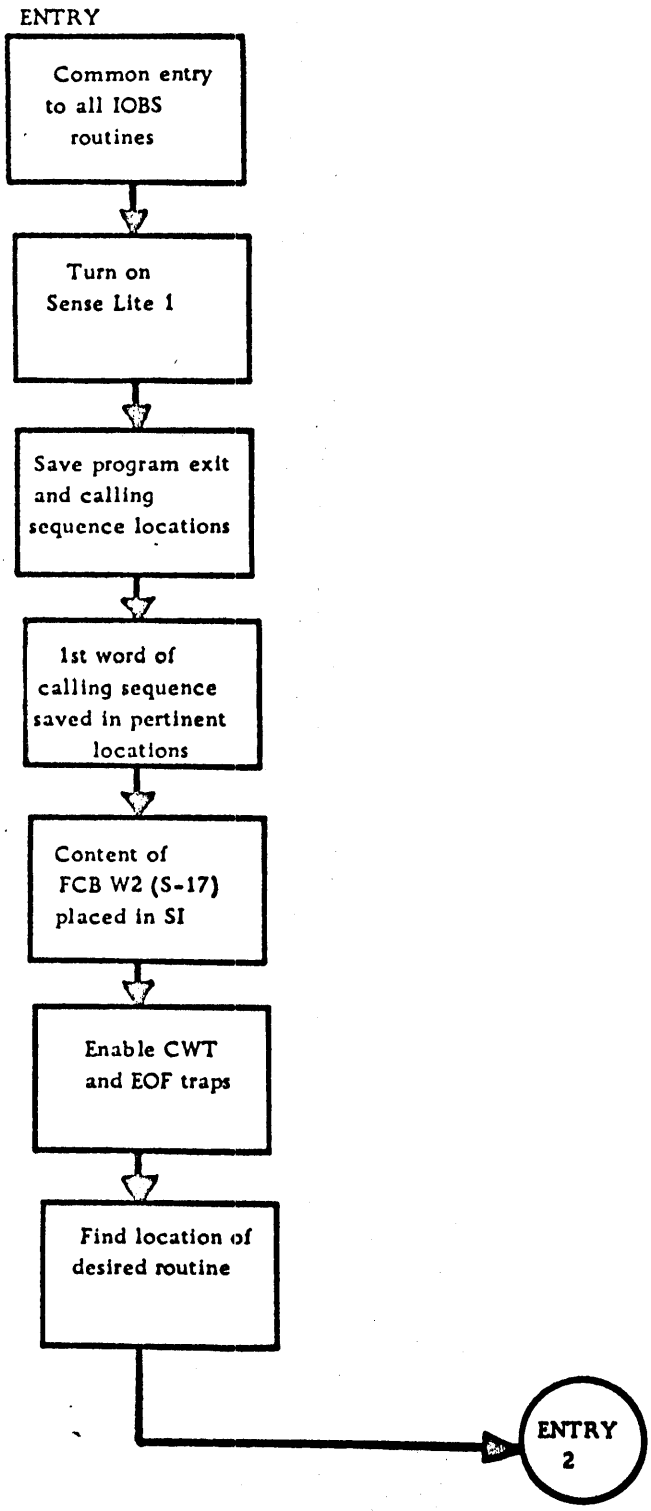
COPY-STASH 5


COPY STASH 6
STSH4
Set stash switch =0
XEOB
Hold chain possibly exist ?
Yes
Buffer connected ?
No
Exit via COMXIT
Yes
No
CPY 6
STSH5
Restore file 1 control bits
Alter FCW to refer to file 2
File control bits of file 2 to SI
TOKBB
If buffer available for file 2 reduce pool buffer ct.
Available
Not available
CPY 6
Restore file 2 control bits
Alter FCW to refer to File 1
FCW control bits of file 1 to SI
COPY STASH 2D
CPY6
Restore file 2 control bits
Alter FCW to refer to file 1
File control bits of file 1 to SI
TOKBB
STASH Switch = 0 (XEOB)
No
Restore file 1 pool buffer count by reducing it by 1
Available
not available
Set EOB SW. Note: should never take this step
Exit via "NTS"
Yes
Set stash switch (XEOB)
BCKBB
Increase pool buffer count by 1, since input buffer can be used and try again

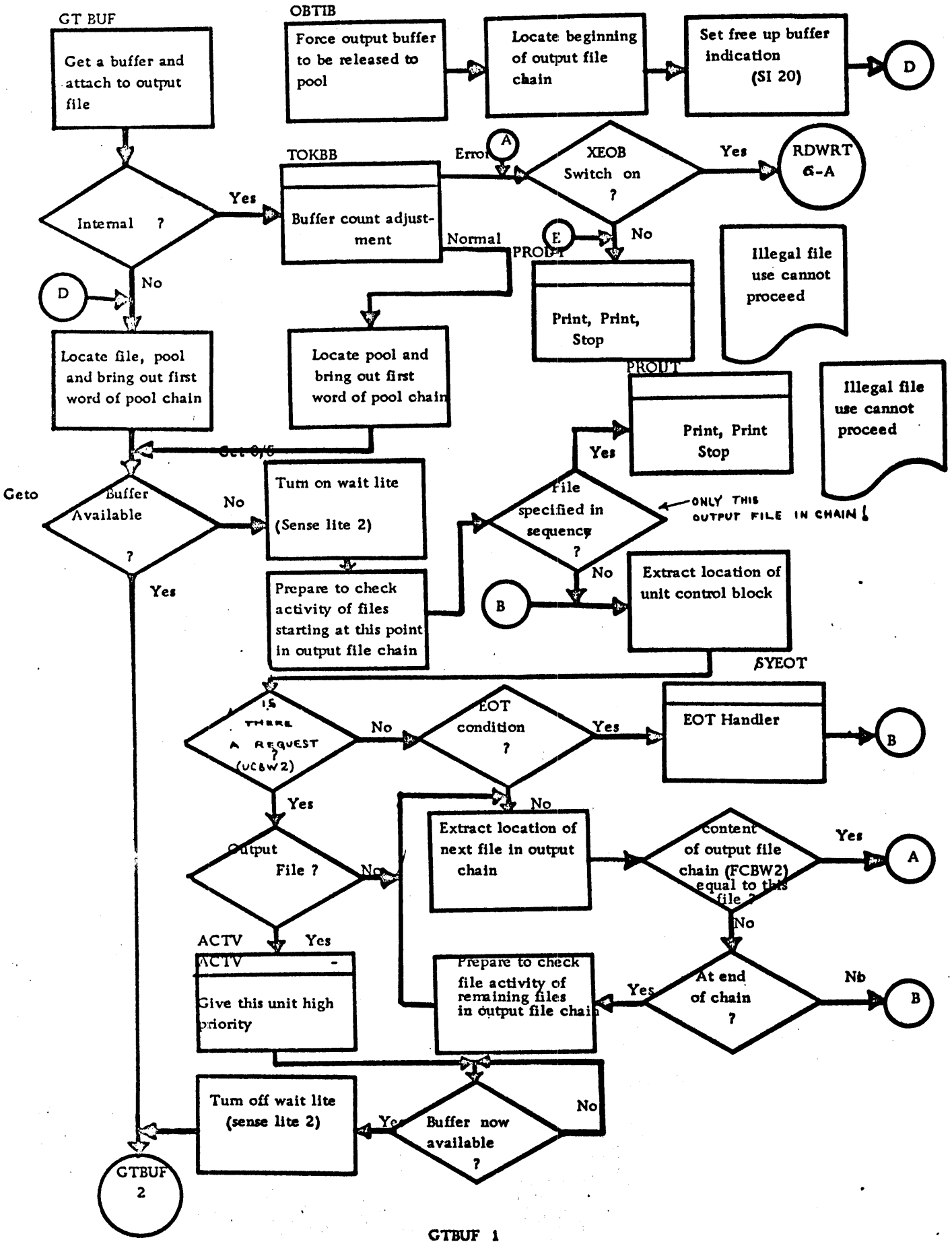

DEFIN
Define a Buffer Pool
Specialize 1st BPCW
SIGN = 1
ADR. = L (POOL + 2)
DE1
Place buffer size in 2nd BPCW
S - 13
Enter W.C. in IORT command
Enter no. of buffers in 1st BPCW
(POS. 3-17)
POOL ERROR TSX AT XXXXX
CANNOT PROCEED
PERR
Print, Print, Stop
X
Yes
Is buff. in system?
No
DE4
Enter command in 2nd BCW
(IORT POOL,,N)
Specialize adr. of 1st BCW
ADR. = L(POOL)
Yes
This last buffer?
No
Chain this buffer to next buffer
Prepare for next buffer
Specialize adr. of 2nd BPCW
ADR. = L (1st BCW of Last Buffer)
TRA 3,4



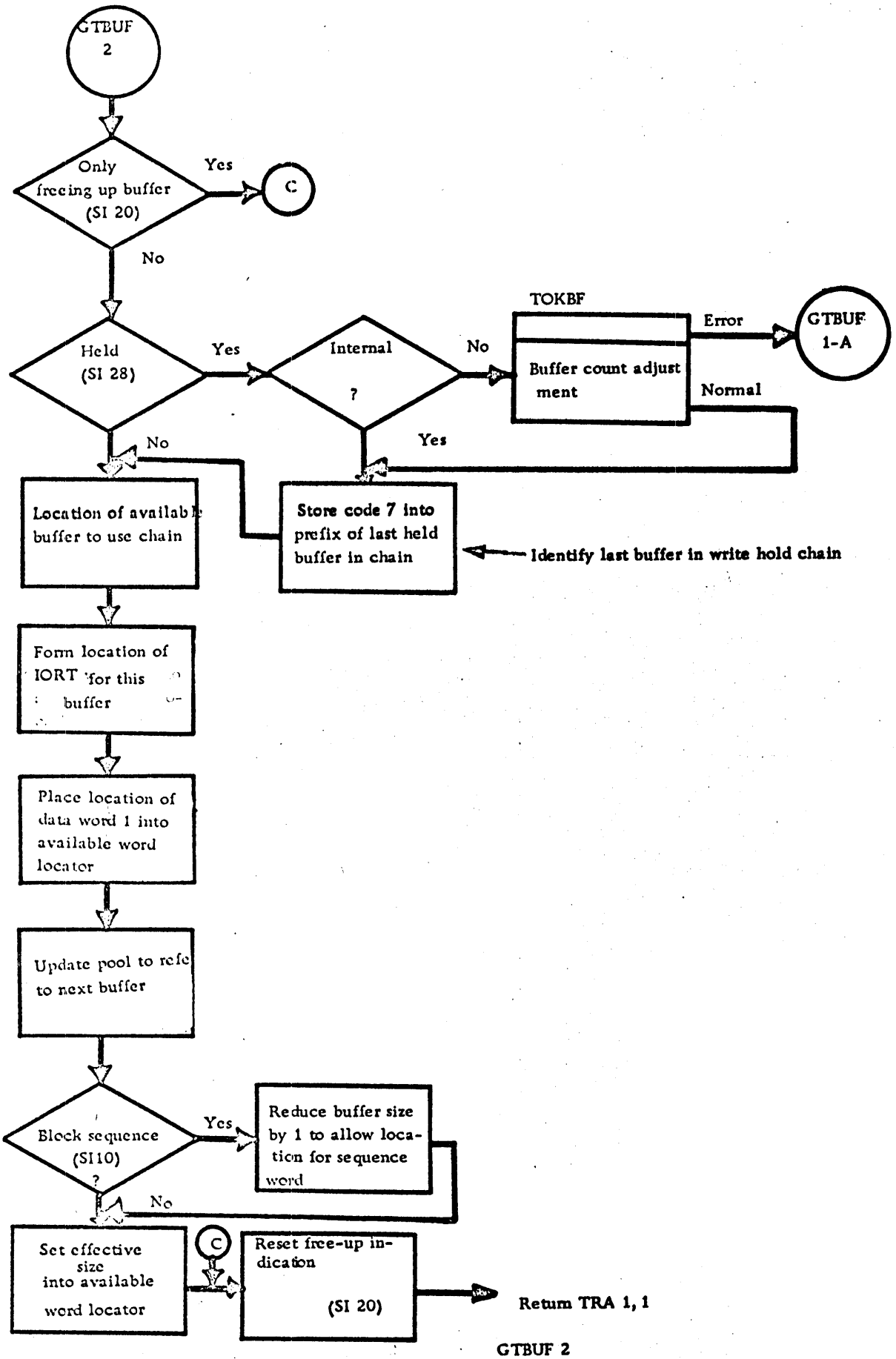
DISCON-RELES


ENTRY
Common entry to all IOBS routines
Turn on Sense Lite 1
Save program exit and calling sequence locations
1st word of calling sequence saved in pertinent locations
Content of FCB W2 (S-17) placed in SI
Enable CWT and EOF traps
Find location of desired routine
ENTRY 2

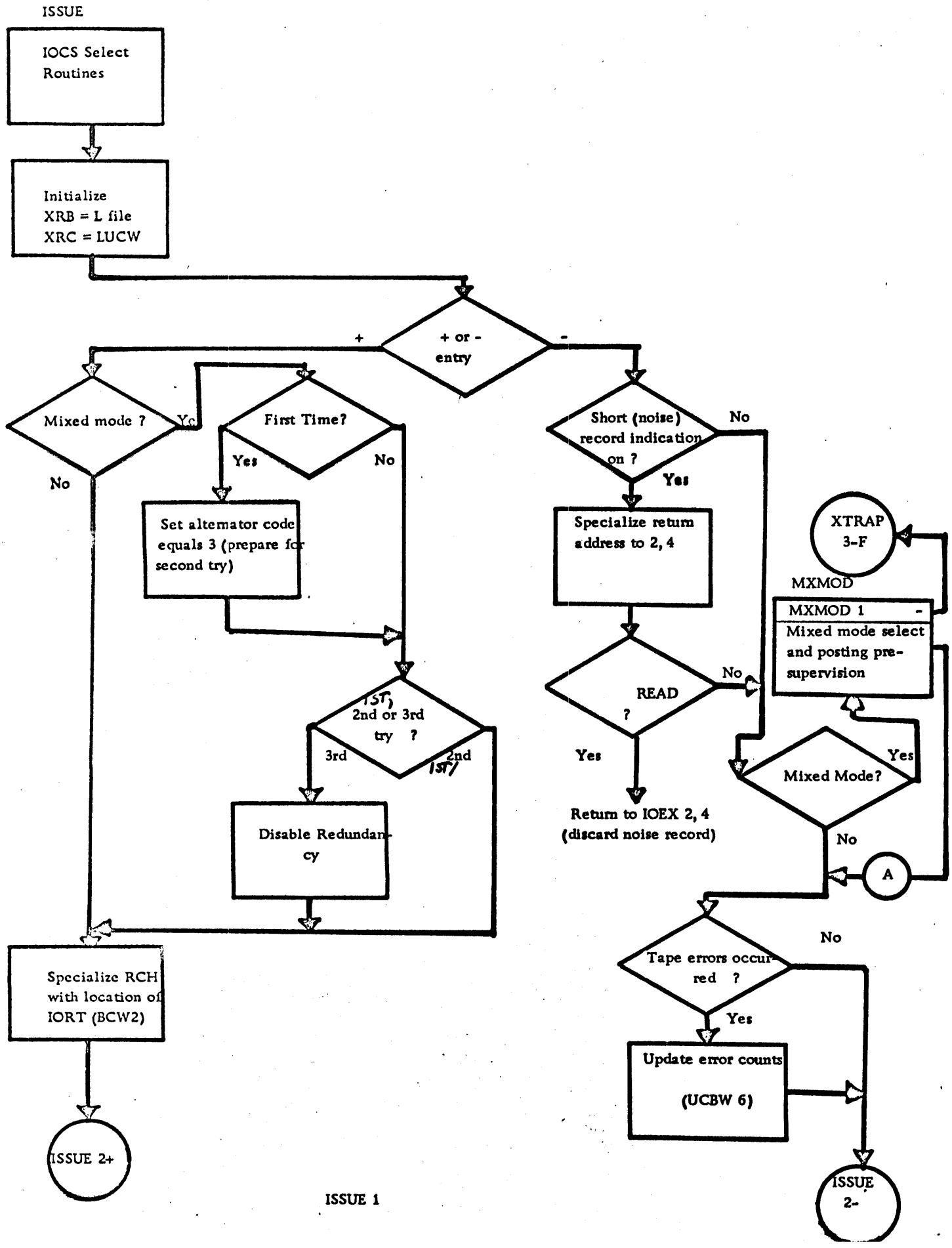
ENTRY 1


GT BUF
Get a buffer and attach to output file
Internal ?
Yes
No
D
Locate file, pool and bring out first word of pool chain
Buffer Available ?
Geto
No
Yes
Turn on wait lite (Sense lite 2)
Prepare to check activity of files starting at this point in output file chain
IS THERE A REQUEST ? (UCBW2)
No
Yes
Output File ?
No
ACTV
Yes
ACTV
Give this unit high priority
Turn off wait lite (sense lite 2)
Yes
Buffer now available ?
No
GTBUF 2
OBTIB
Force output buffer to be released to pool
Locate beginning of output file chain
Set free up buffer indication (SI 20)
D
TOKBB
Buffer count adjustment
Error
A
XEOB Switch on ?
Yes
RDWRT 6-A
Normal
E
No
PRODT
Print, Print, Stop
Illegal file use cannot proceed
Locate pool and bring out first word of pool chain
Yes
PRODT
Print, Print Stop
Illegal file use cannot proceed
File specified in sequence ?
ONLY THIS OUTPUT FILE IN CHAIN !
No
B
Extract location of unit control block
SYEOT
EOT condition ?
No
Yes
EOT Handler
B
No
Extract location of next file in output chain
Content of output file chain (FCBW2) equal to this file ?
Yes
A
No
Prepare to check file activity of remaining files in output file chain
At end of chain ?
Yes
No
B

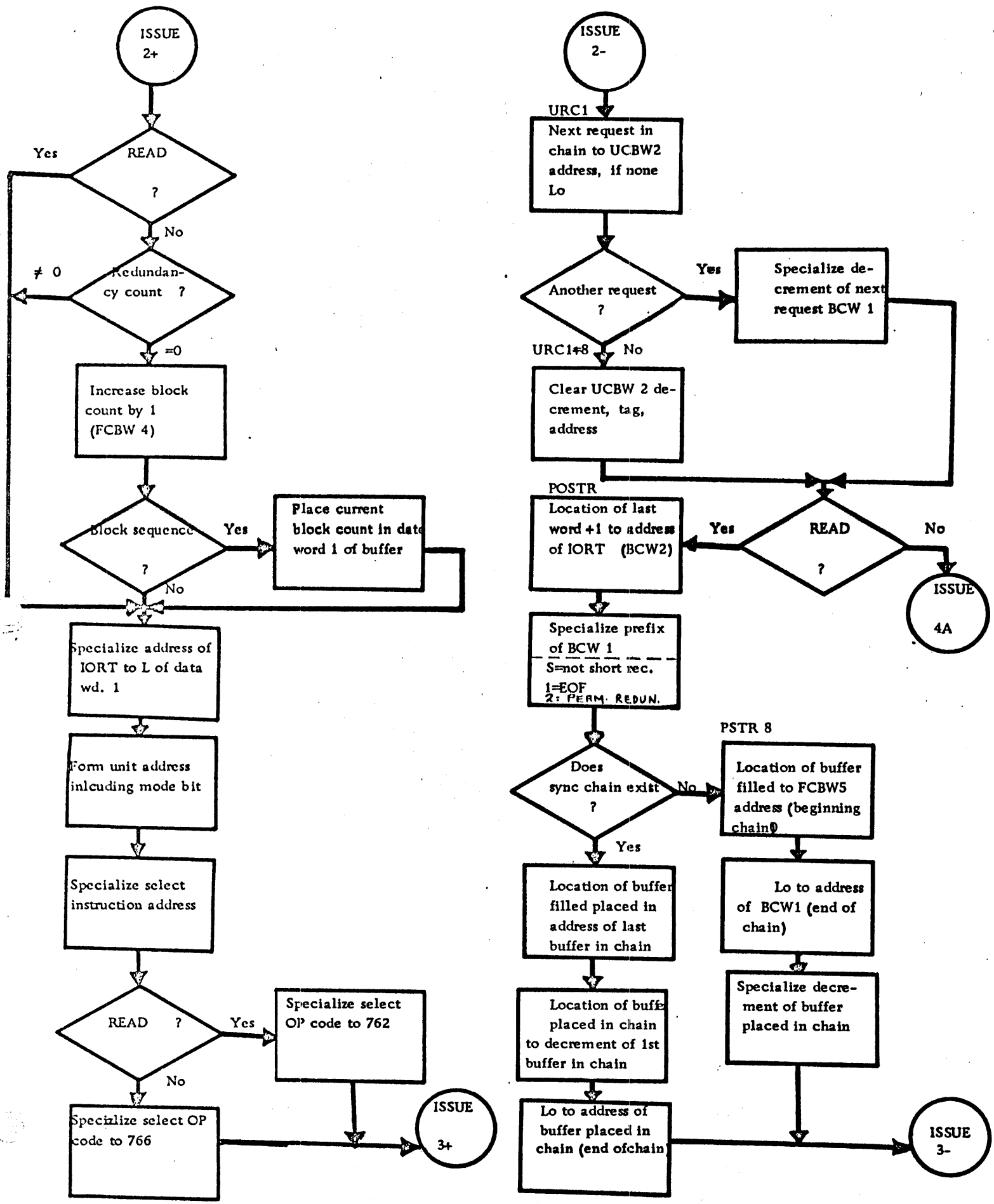
GTBUF 1


GTBUF 2
Only freeing up buffer (SI 20)
Yes
C
No
Held (SI 28)
Yes
Internal ?
No
TOKBF
Buffer count adjust ment
Error
GTBUF 1-A
Normal
Yes
No
Location of availab buffer to use chain
Store code 7 into prefix of last held buffer in chain
Identify last buffer in write hold chain
Form location of IORT for this buffer
Place location of data word 1 into available word locator
Update pool to refe to next buffer
Block sequence (SI10) ?
Yes
Reduce buffer size by 1 to allow location for sequence word
No
Set effective size into available word locator
C
Reset free-up indication (SI 20)
Return TRA 1, 1

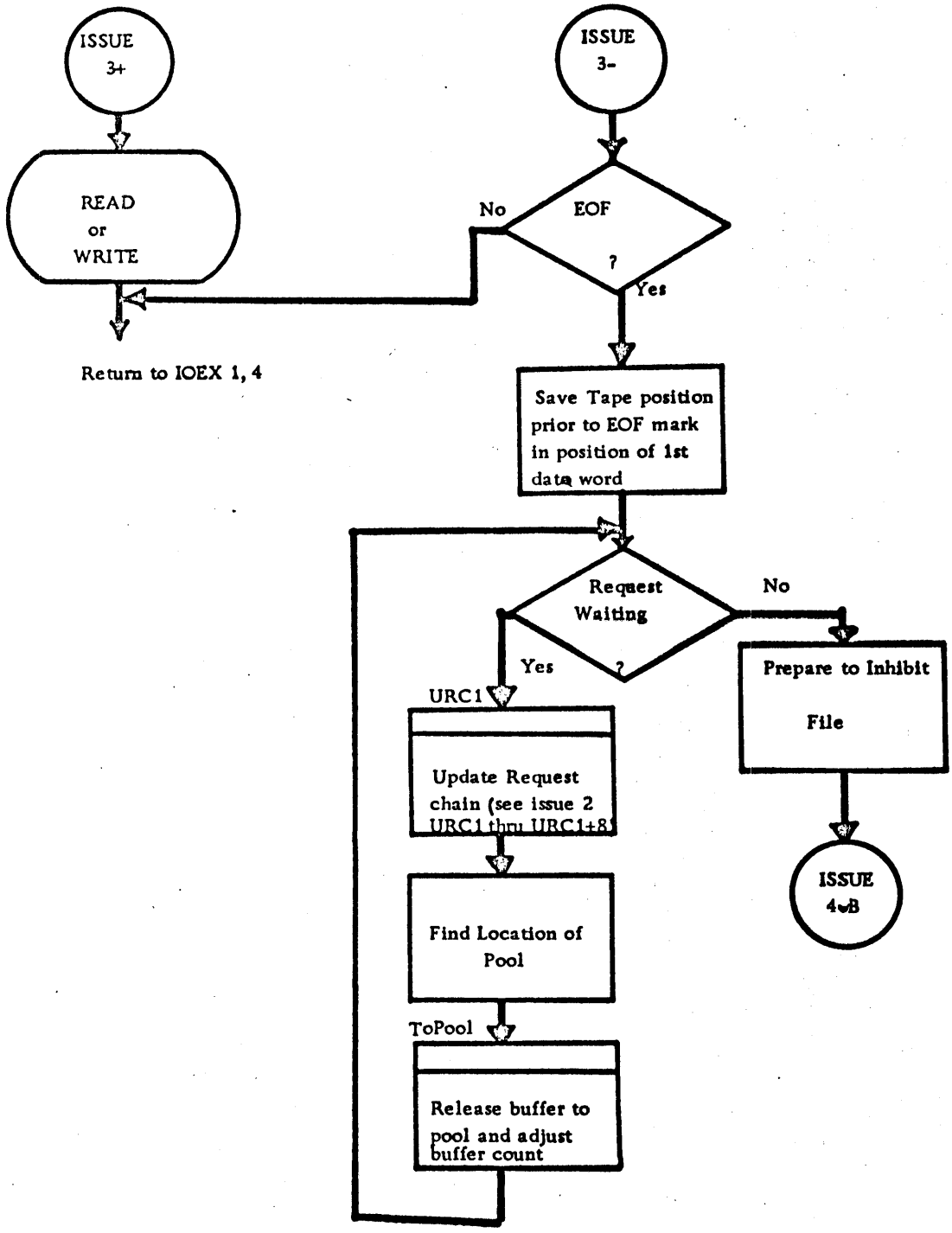
GTBUF 2

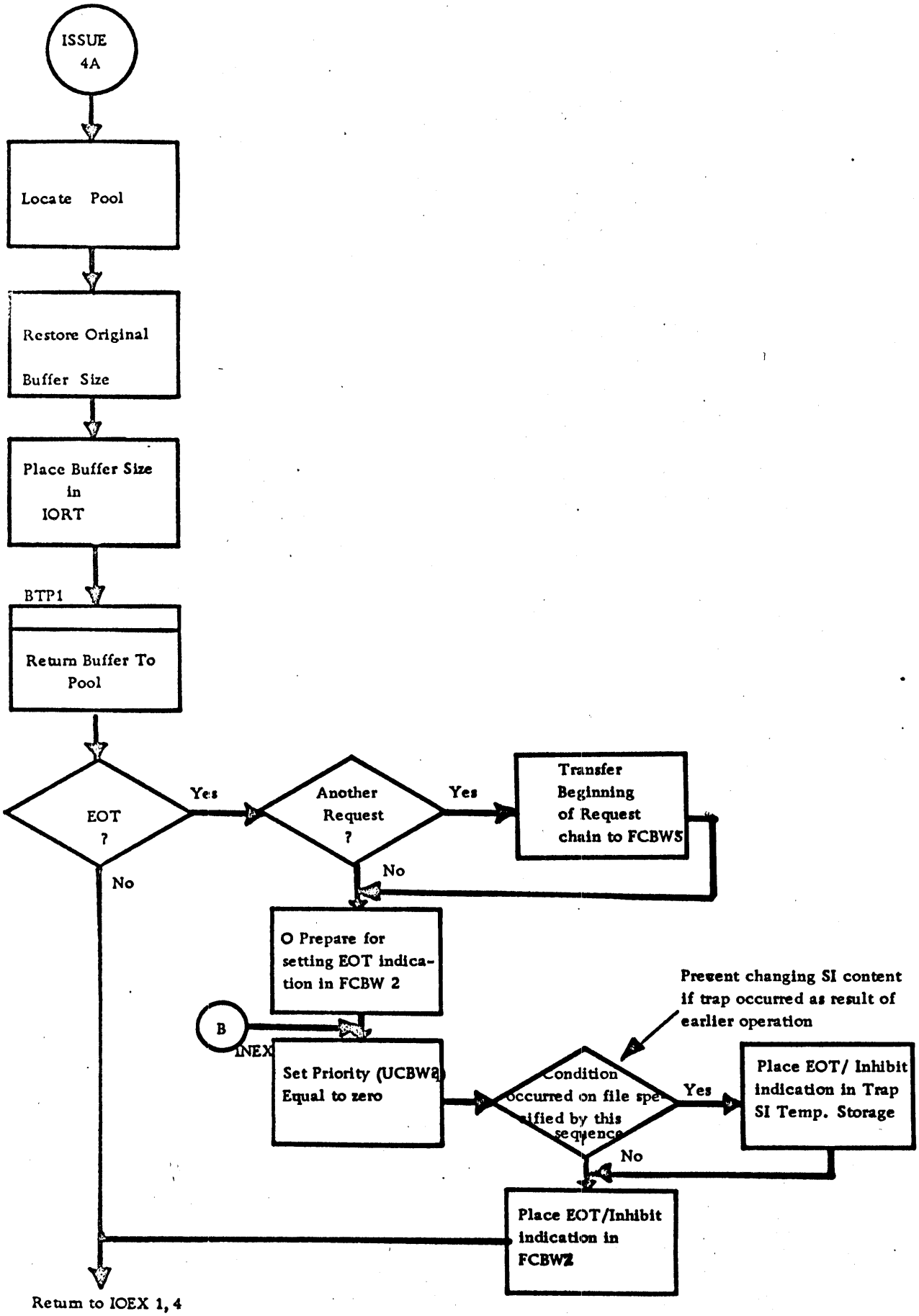

ISSUE
IOCS Select Routines
Initialize
XRB = L file
XRC = LUCW
+ or - entry
+
-
Mixed mode ?
Yes
No
First Time?
Yes
No
Set alternator code equals 3 (prepare for second try)
2nd or 3rd try ?
(1ST)
3rd
2nd
(1ST)
Disable Redundancy
Specialize RCH with location of IORT (BCW2)
ISSUE 2+
Short (noise) record indication on ?
No
Yes
Specialize return address to 2, 4
READ ?
No
Yes
Return to IOEX 2, 4 (discard noise record)
XTRAP 3-F
MXMOD
MXMOD 1 - Mixed mode select and posting pre-supervision
Mixed Mode?
Yes
No
A
Tape errors occurred ?
No
Yes
Update error counts (UCBW 6)
ISSUE 2-

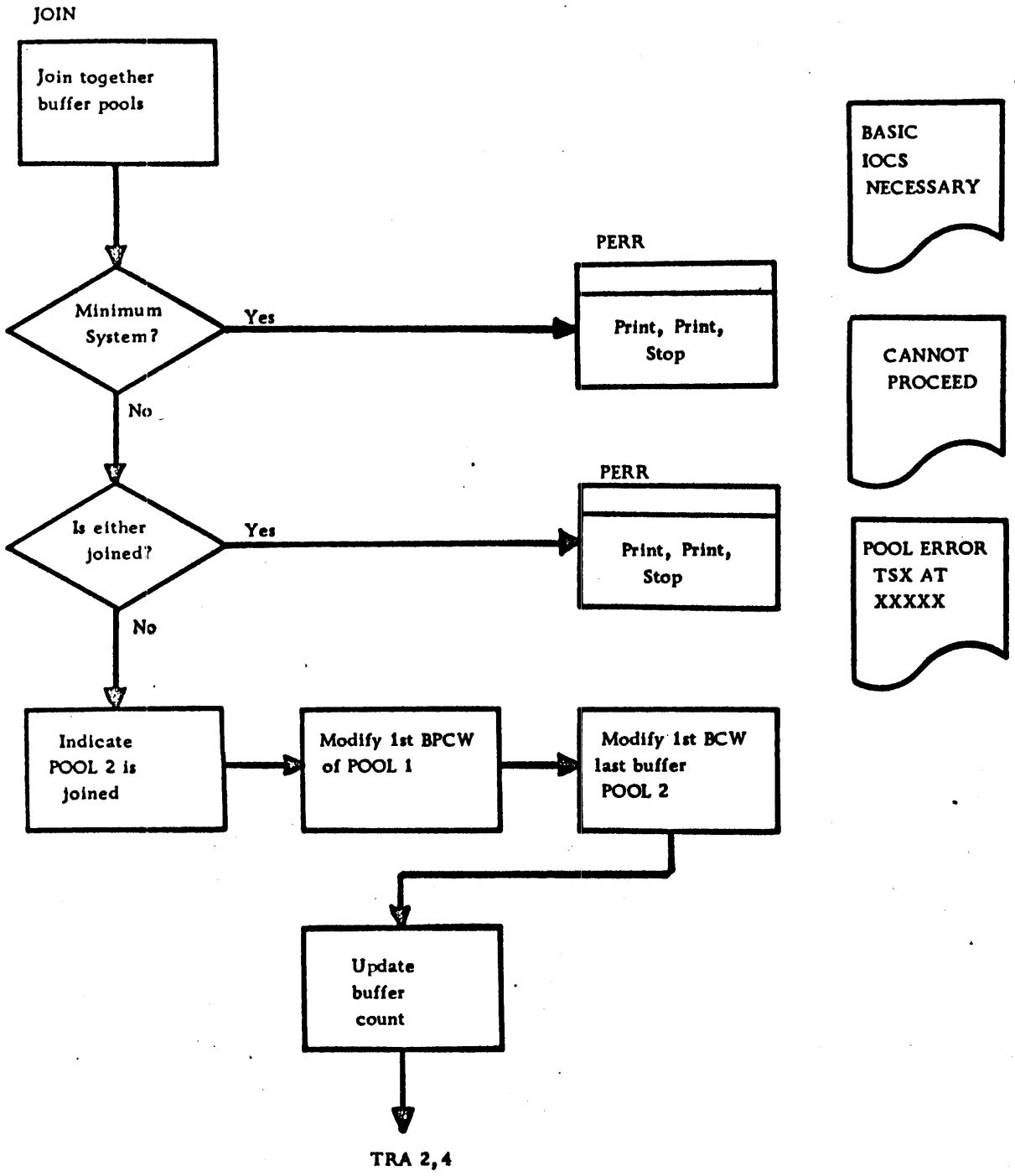
ISSUE 1

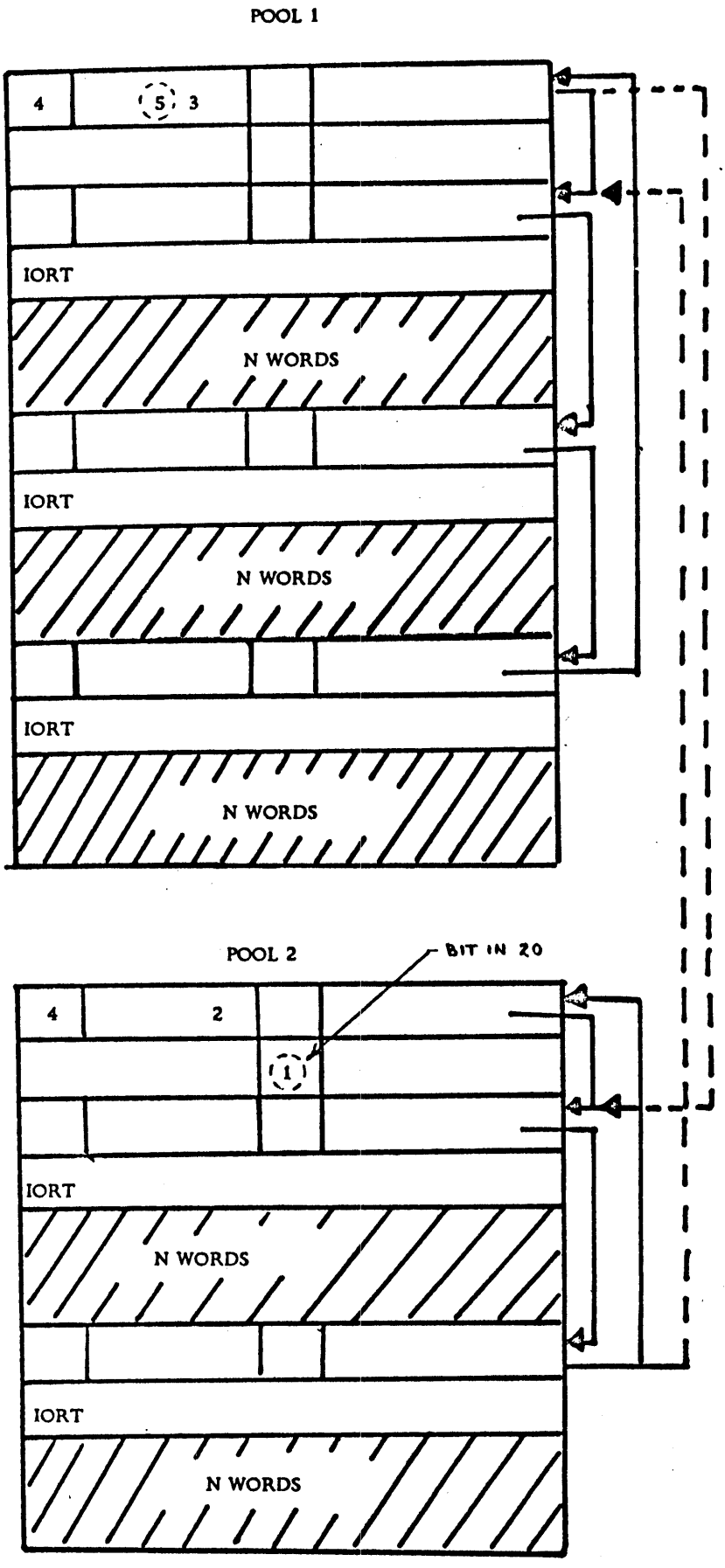

ISSUE 2+
READ ?
Yes
No
≠ 0
Redundancy count ?
=0
Increase block count by 1 (FCBW 4)
Block sequence ?
Yes
Place current block count in data word 1 of buffer
No
Specialize address of IORT to L of data wd. 1
Form unit address inlcuding mode bit
Specialize select instruction address
READ ?
Yes
Specialize select OP code to 762
No
Specialize select OP code to 766
ISSUE 3+
ISSUE 2-
URC1
Next request in chain to UCBW2 address, if none Lo
Another request ?
Yes
Specialize decrement of next request BCW 1
URC1≠8
No
Clear UCBW 2 decrement, tag, address
POSTR
Location of last word +1 to address of IORT (BCW2)
Yes
READ ?
No
ISSUE 4A
Specialize prefix of BCW 1
S=not short rec.
1=EOF
2: PERM. REDUN.
Does sync chain exist ?
No
PSTR 8
Location of buffer filled to FCBW5 address (beginning chain)
Yes
Location of buffer filled placed in address of last buffer in chain
Lo to address of BCW1 (end of chain)
Location of buffer placed in chain to decrement of 1st buffer in chain
Specialize decrement of buffer placed in chain
Lo to address of buffer placed in chain (end ofchain)
ISSUE 3-

ISSUE 2

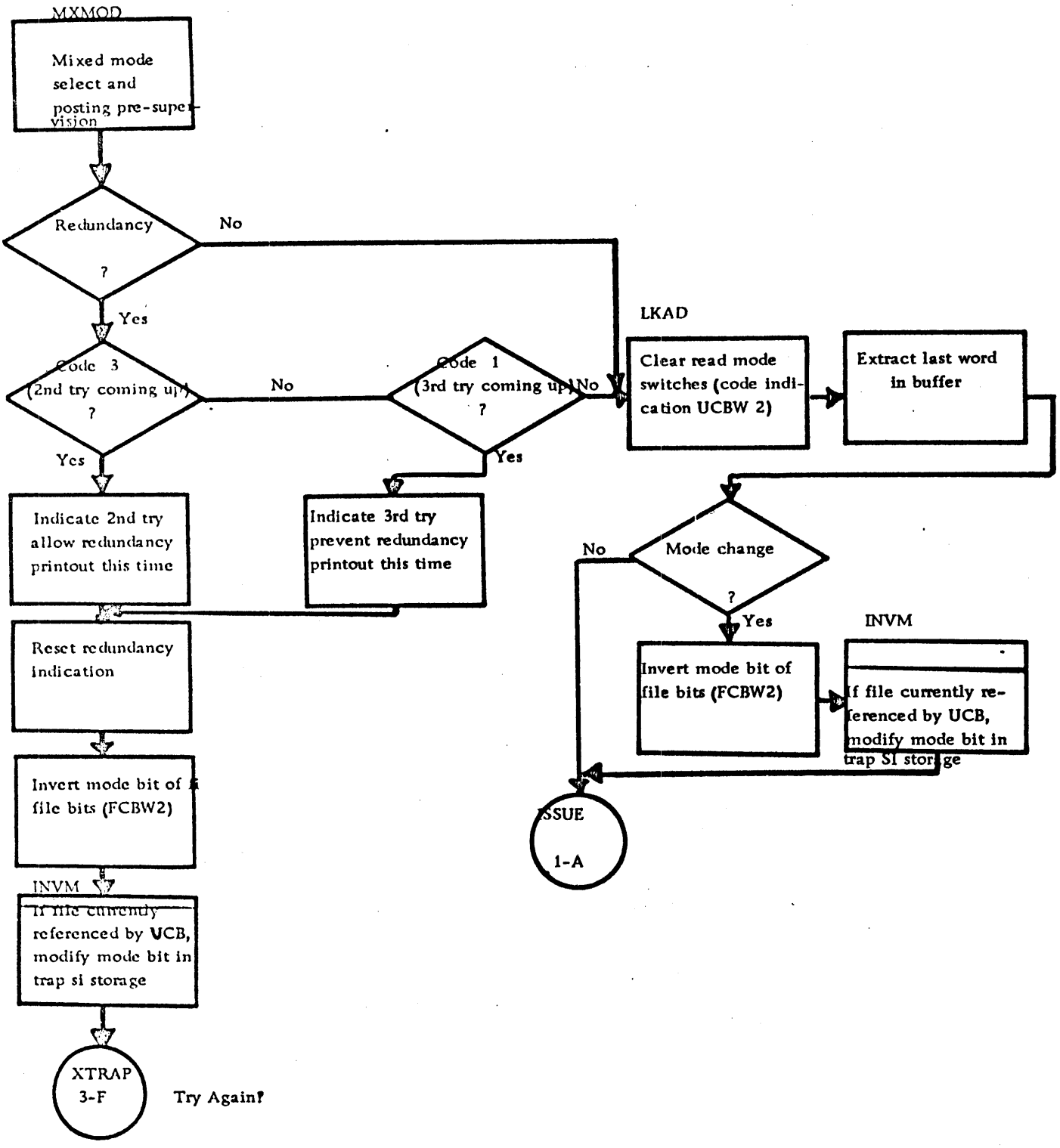

ISSUE 3+
READ or WRITE
Return to IOEX 1, 4
ISSUE 3-
EOF ?
No
Yes
Save Tape position prior to EOF mark in position of 1st data word
Request Waiting ?
No
Yes
URC1
Update Request chain (see issue 2 URC1 thru URC1+8)
Find Location of Pool
ToPool
Release buffer to pool and adjust buffer count
Prepare to Inhibit File
ISSUE 4+B

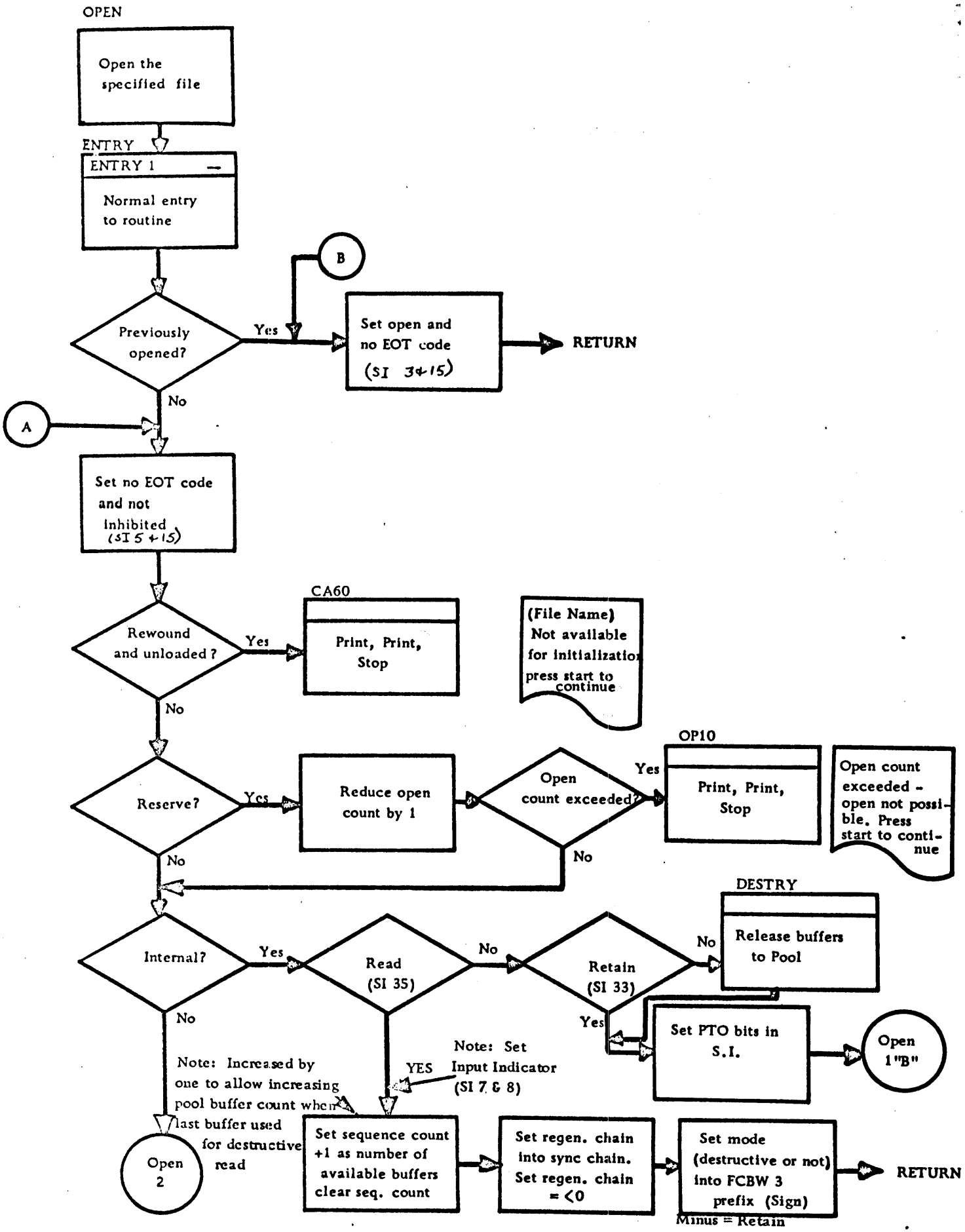

ISSUE 4A
Locate Pool
Restore Original Buffer Size
Place Buffer Size in IORT
BTP1
Return Buffer To Pool
EOT ?
Yes
Another Request ?
Yes
Transfer Beginning of Request chain to FCBW5
No
No
O Prepare for setting EOT indication in FCBW 2
B
INEX
Set Priority (UCBW2) Equal to zero
Prevent changing SI content if trap occurred as result of earlier operation
Condition occurred on file specified by this sequence
Yes
Place EOT/ Inhibit indication in Trap SI Temp. Storage
No
Place EOT/Inhibit indication in FCBW2
Return to IOEX 1, 4

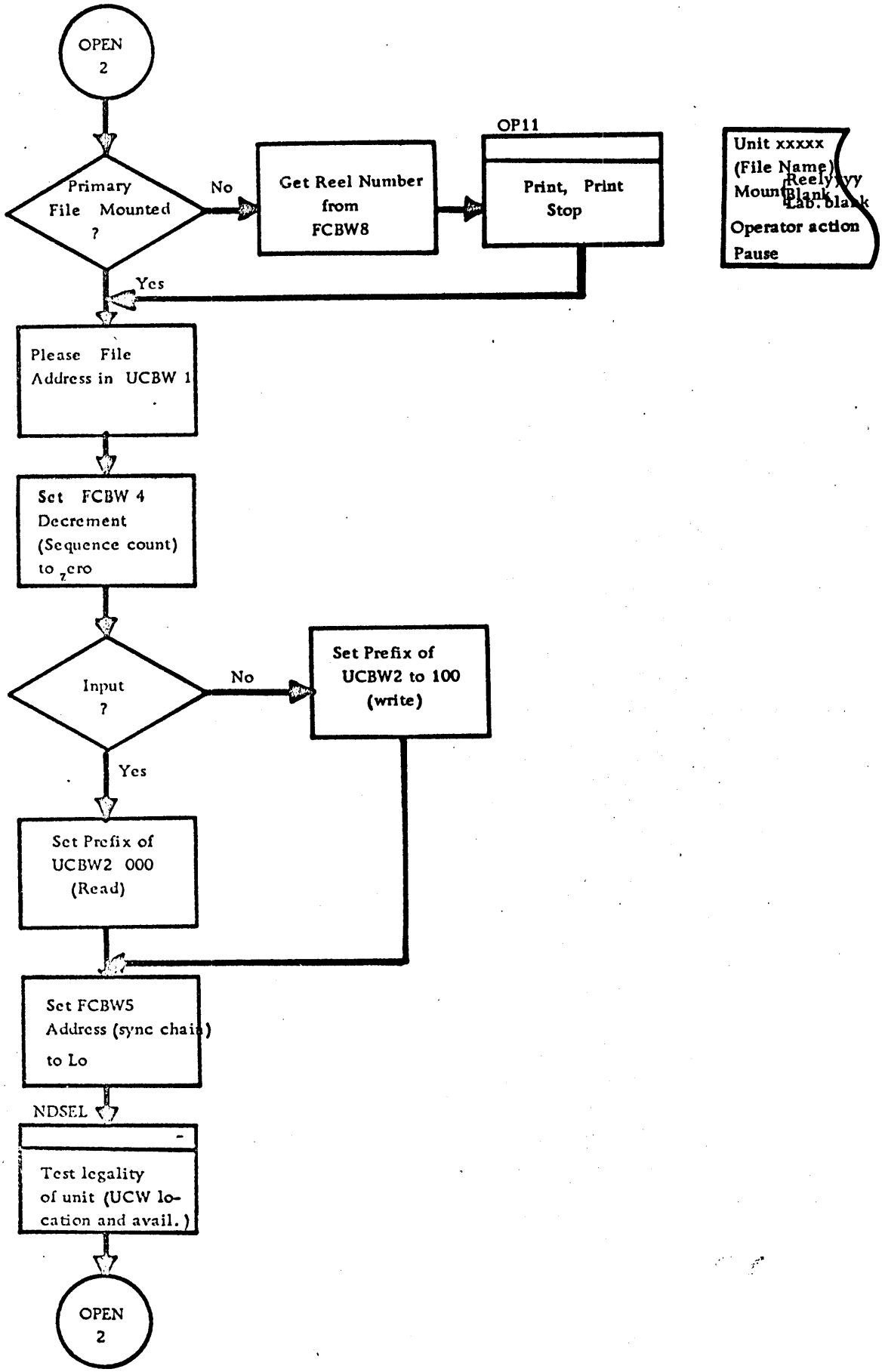

JOIN
Join together buffer pools
Minimum System?
Yes
No
PERR
Print, Print, Stop
BASIC IOCS NECESSARY
CANNOT PROCEED
Is either joined?
Yes
No
PERR
Print, Print, Stop
POOL ERROR TSX AT XXXXX
Indicate POOL 2 is joined
Modify 1st BPCW of POOL 1
Modify 1st BCW last buffer POOL 2
Update buffer count
TRA 2,4


POOL 1
4
5
3
IORT
N WORDS
IORT
N WORDS
IORT
N WORDS
POOL 2
BIT IN 20
4
2
1
IORT
N WORDS
IORT
N WORDS

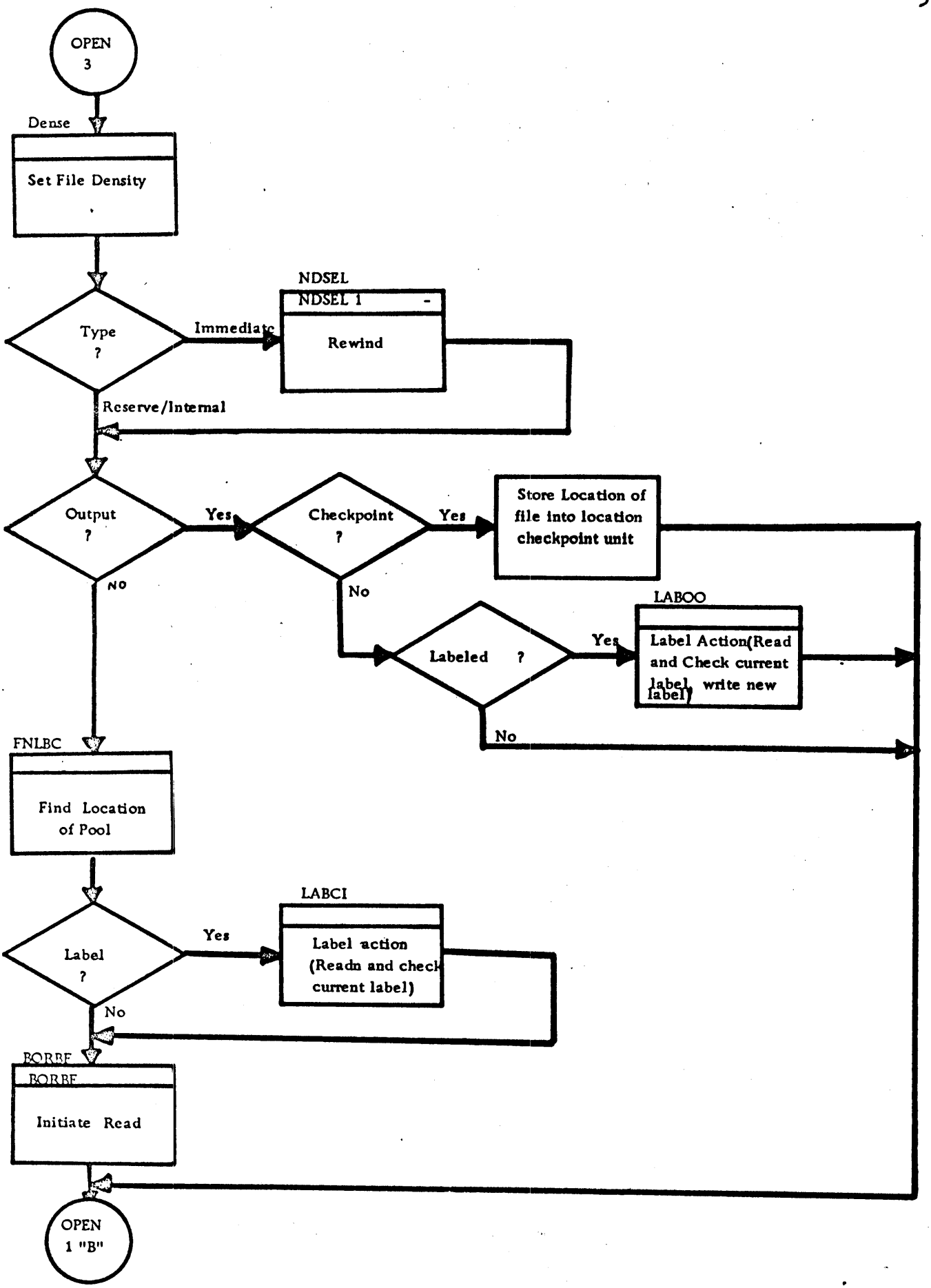
JOIN. 2

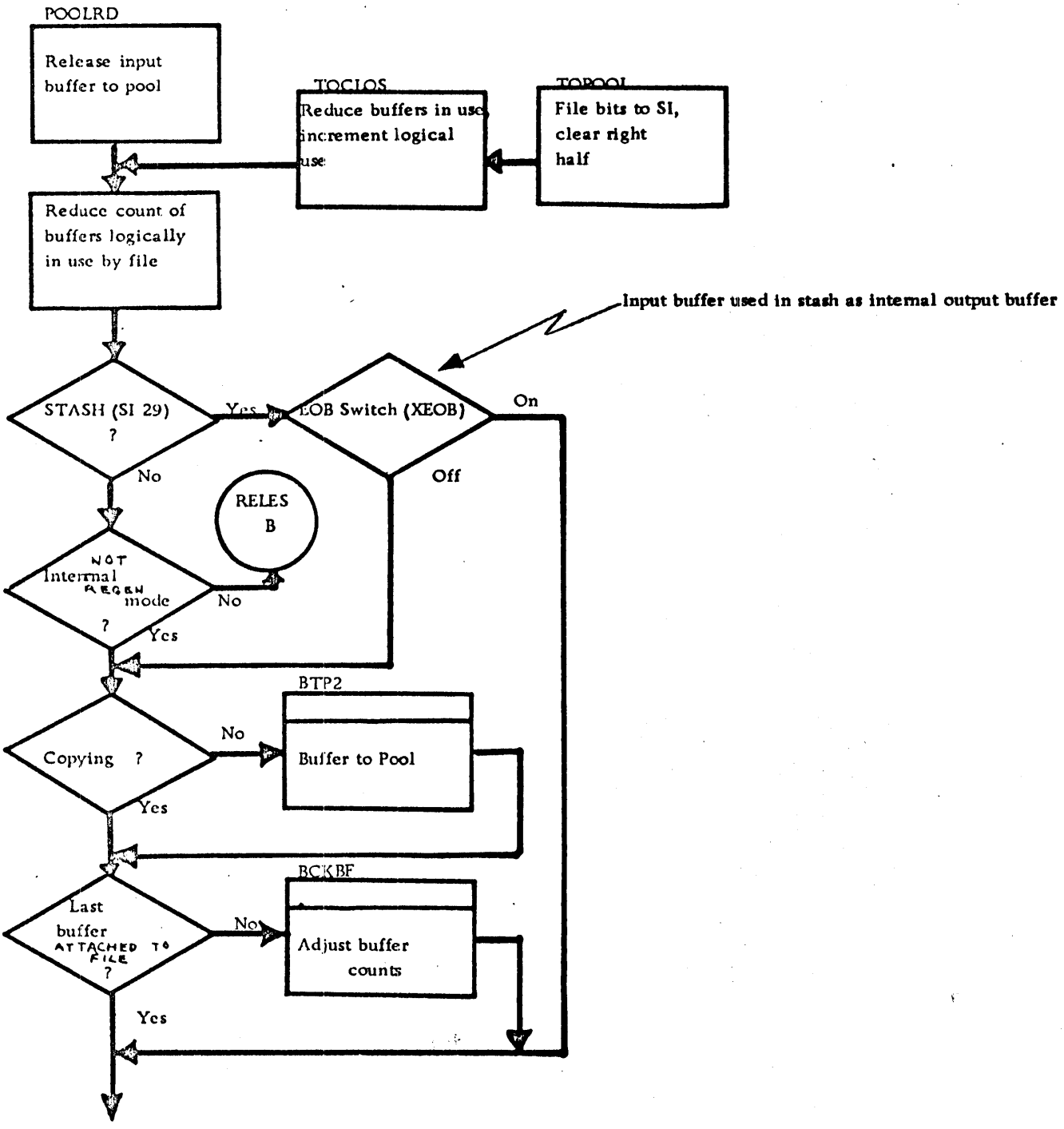

MXMOD
Mixed mode select and posting pre-supervision
Redundancy ?
No
Yes
Code 3 (2nd try coming up) ?
No
Code 1 (3rd try coming up) ?
No
Yes
Yes
LKAD
Clear read mode switches (code indication UCBW 2)
Extract last word in buffer
Indicate 2nd try allow redundancy printout this time
Indicate 3rd try prevent redundancy printout this time
Mode change ?
No
Yes
INVM
Invert mode bit of file bits (FCBW2)
If file currently referenced by UCB, modify mode bit in trap SI storage
Reset redundancy indication
Invert mode bit of file bits (FCBW2)
INVM
If file currently referenced by UCB, modify mode bit in trap si storage
ISSUE 1-A
XTRAP 3-F
Try Again?

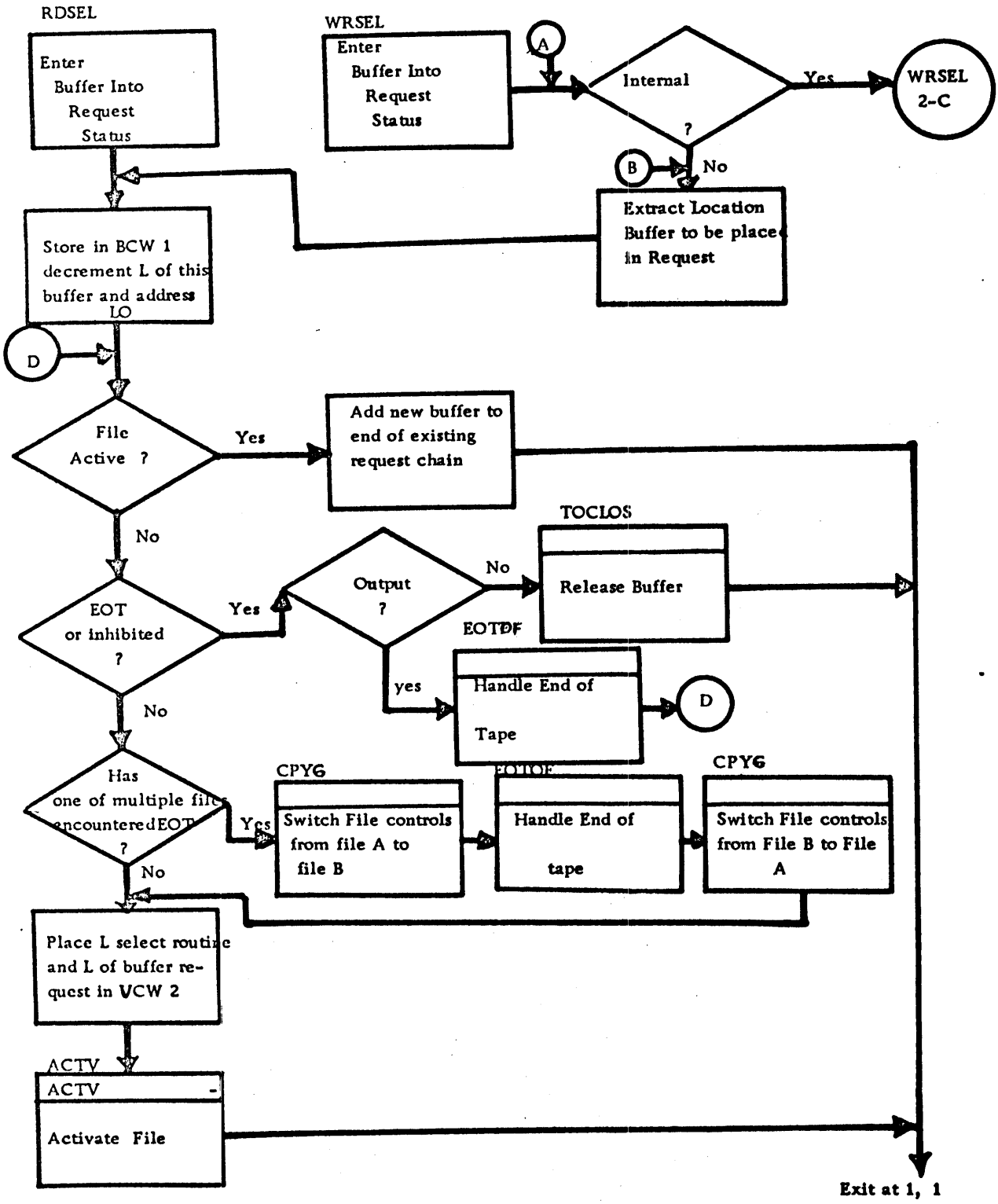

OPEN
Open the specified file
ENTRY
ENTRY 1
Normal entry to routine
B
Previously opened?
Yes
Set open and no EOT code (SI 3+15)
RETURN
No
A
Set no EOT code and not Inhibited (SI 5 + 15)
Rewound and unloaded?
Yes
CA60
Print, Print, Stop
(File Name) Not available for initialization press start to continue
No
Reserve?
Yes
Reduce open count by 1
Open count exceeded?
Yes
OP10
Print, Print, Stop
Open count exceeded - open not possible. Press start to continue
No
No
DESTRY
Internal?
Yes
Read (SI 35)
No
Retain (SI 33)
No
Release buffers to Pool
Yes
Set PTO bits in S.I.
Open 1 "B"
No
Note: Increased by one to allow increasing pool buffer count when last buffer used for destructive read
YES
Note: Set Input Indicator (SI 7 & 8)
Open 2
Set sequence count +1 as number of available buffers clear seq. count
Set regen. chain into sync chain. Set regen. chain = <0
Set mode (destructive or not) into FCBW 3 prefix (Sign)
RETURN
Minus = Retain


OPEN
2
Primary
File Mounted
?
No
Get Reel Number
from
FCBW8
OP11
Print, Print
Stop
Unit xxxxx
(File Name)
Mount Reelyyyy
Blank
Lab. blank
Operator action
Pause
Yes
Please File
Address in UCBW 1
Set FCBW 4
Decrement
(Sequence count)
to zero
Input
?
No
Set Prefix of
UCBW2 to 100
(write)
Yes
Set Prefix of
UCBW2 000
(Read)
Set FCBW5
Address (sync chain)
to Lo
NDSEL
Test legality
of unit (UCW lo-
cation and avail.)
OPEN
2

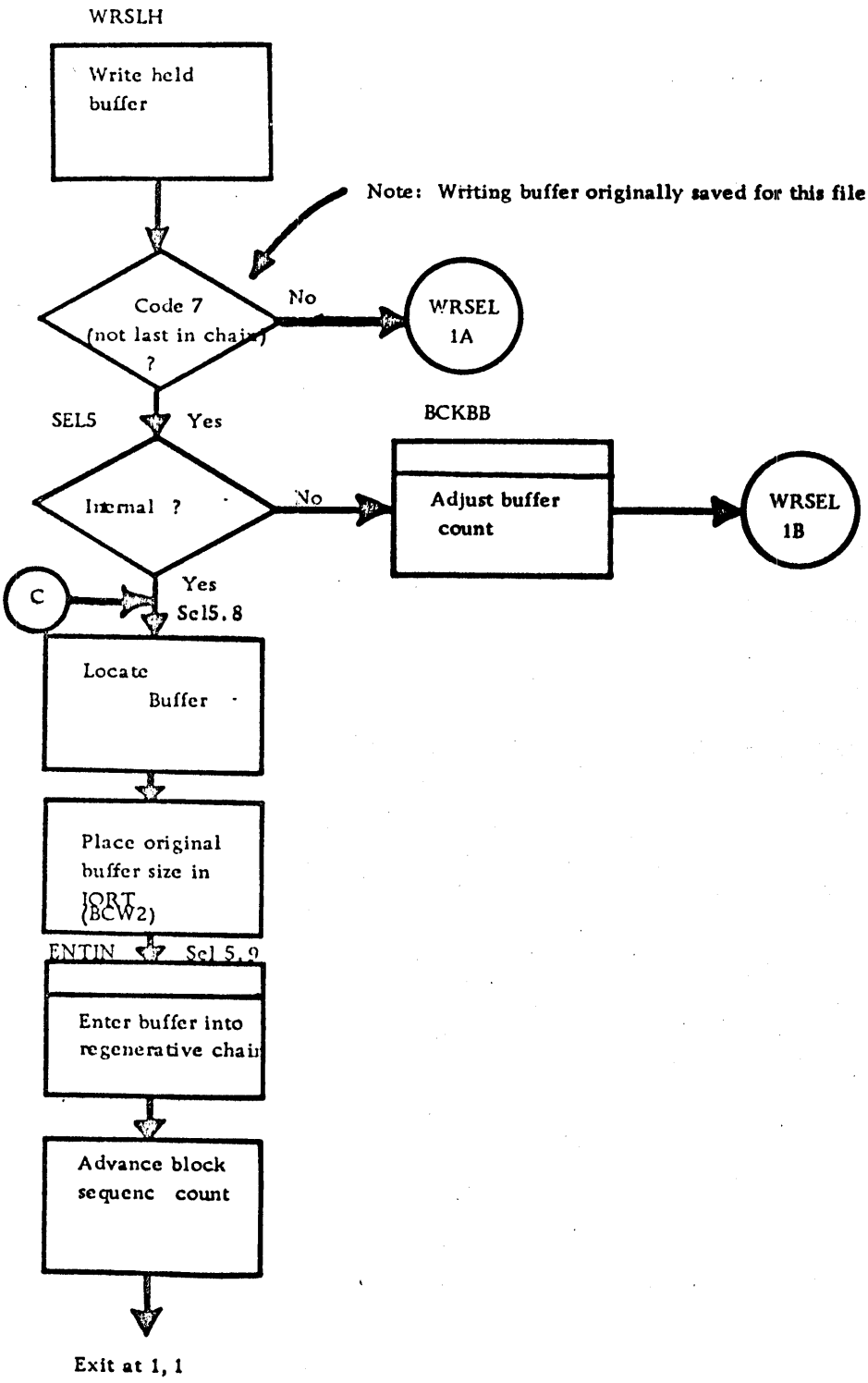
OPEN 2

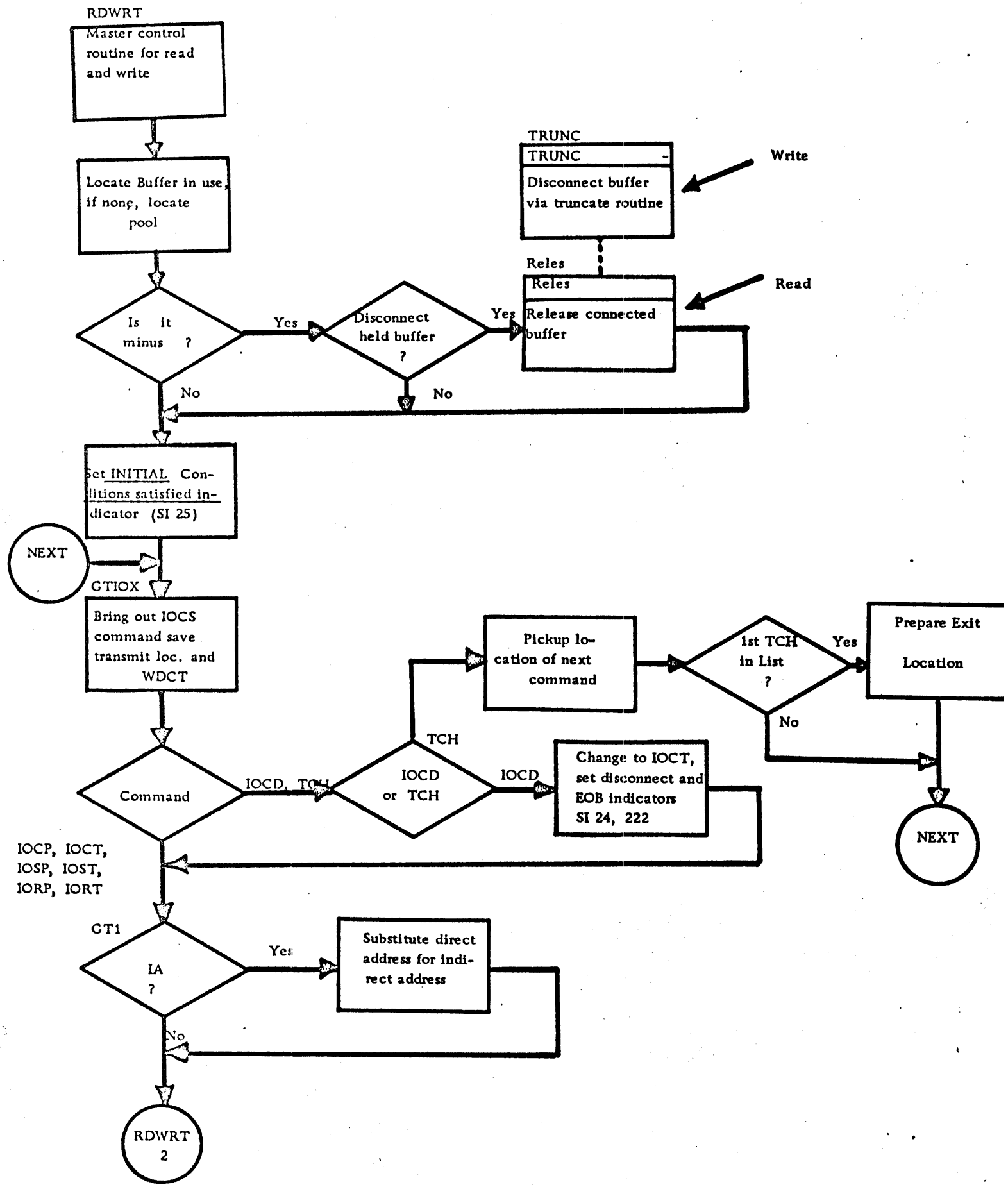

OPEN
3
Dense
Set File Density
NDSEL
NDSEL 1
Type
?
Immediate
Rewind
Reserve/Internal
Output
?
Yes
Checkpoint
?
Yes
Store Location of file into location checkpoint unit
NO
No
LABOO
Labeled ?
Yes
Label Action(Read and Check current label, write new label)
No
FNLBC
Find Location of Pool
LABCI
Label
?
Yes
Label action (Readn and check current label)
No
BORBF
BORBF
Initiate Read
OPEN
1 "B"

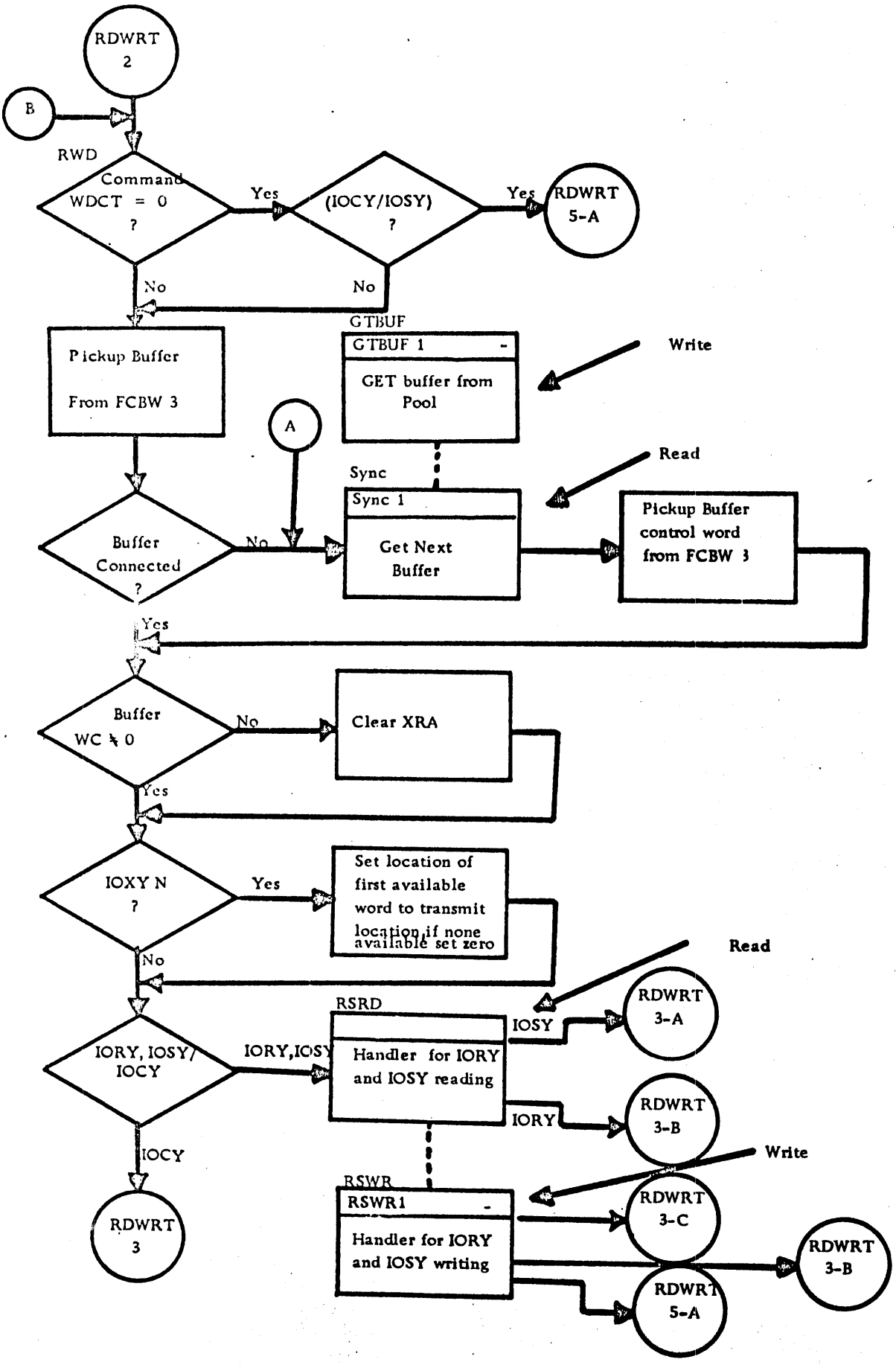

POOLRD
Release input buffer to pool
TOCLOS
Reduce buffers in use, increment logical use
TOPOOL
File bits to SI, clear right half
Reduce count of buffers logically in use by file
Input buffer used in stash as internal output buffer
STASH (SI 29) ?
Yes
EOB Switch (XEOB)
On
Off
No
RELES B
NOT Internal REGEN mode ?
No
Yes
BTP2
Copying ?
No
Buffer to Pool
Yes
BCKBF
Last buffer ATTACHED TO FILE ?
No
Adjust buffer counts
Yes
Return - TRA 1, 1

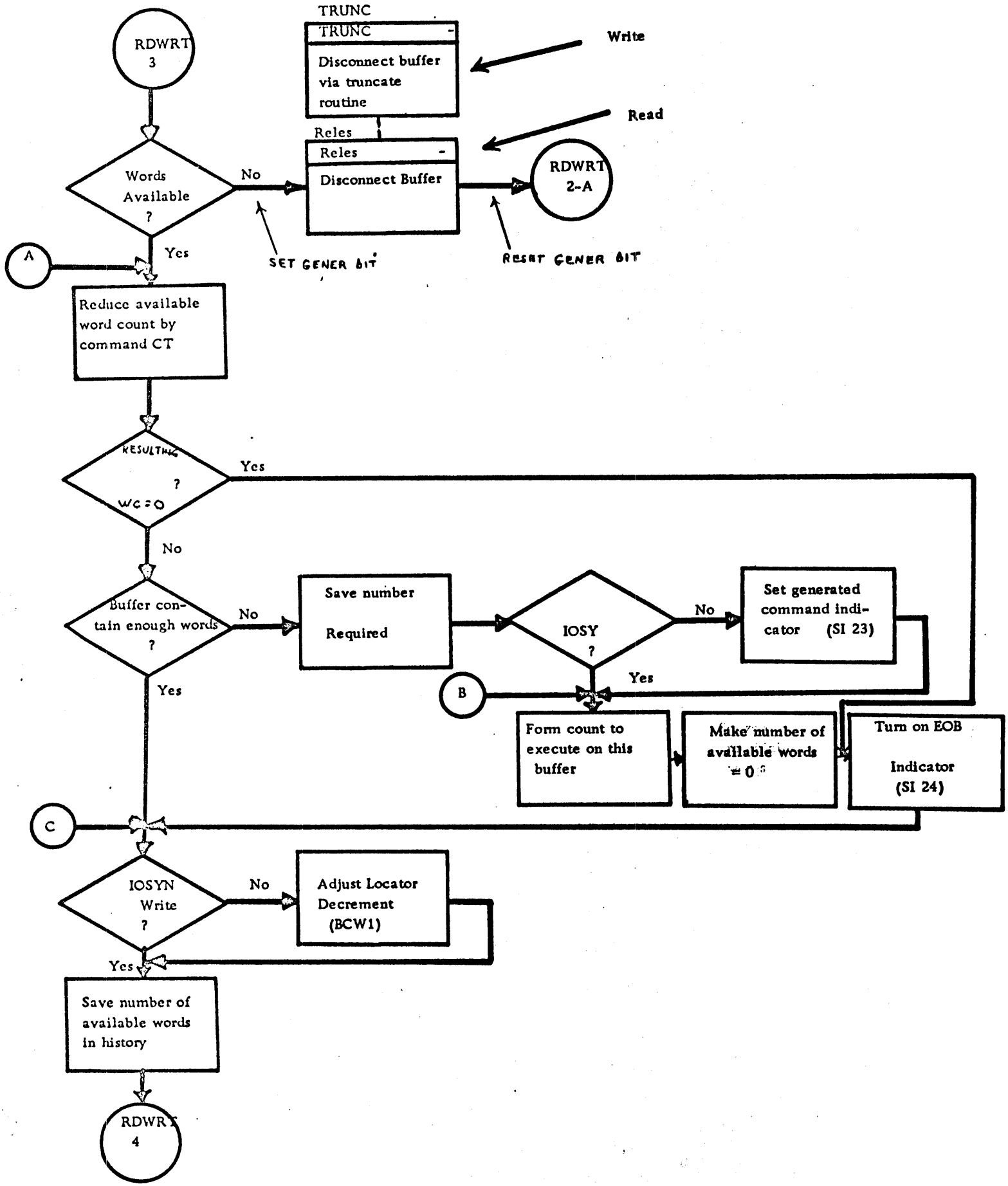


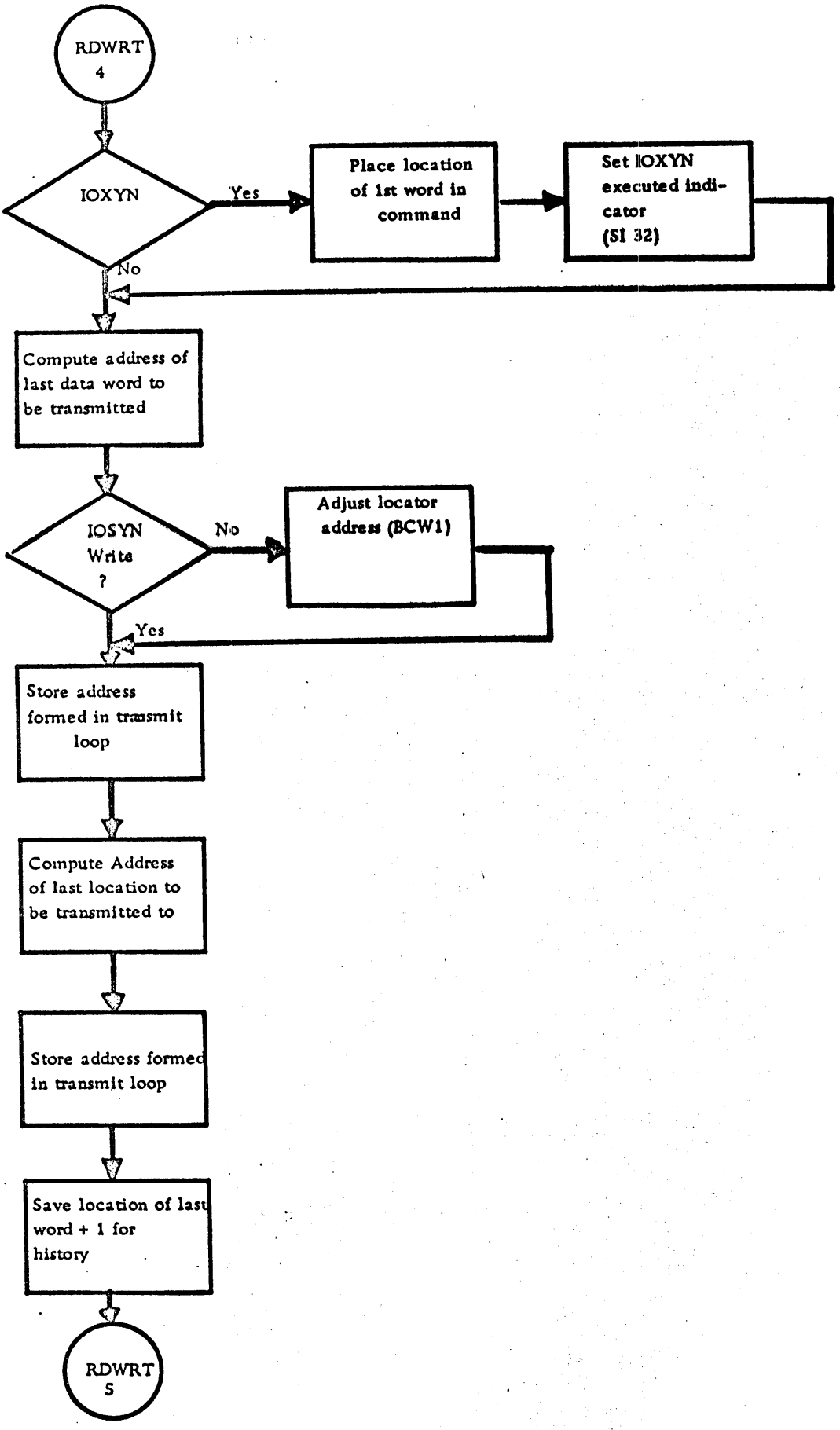
RD SEL = WRSEL 1

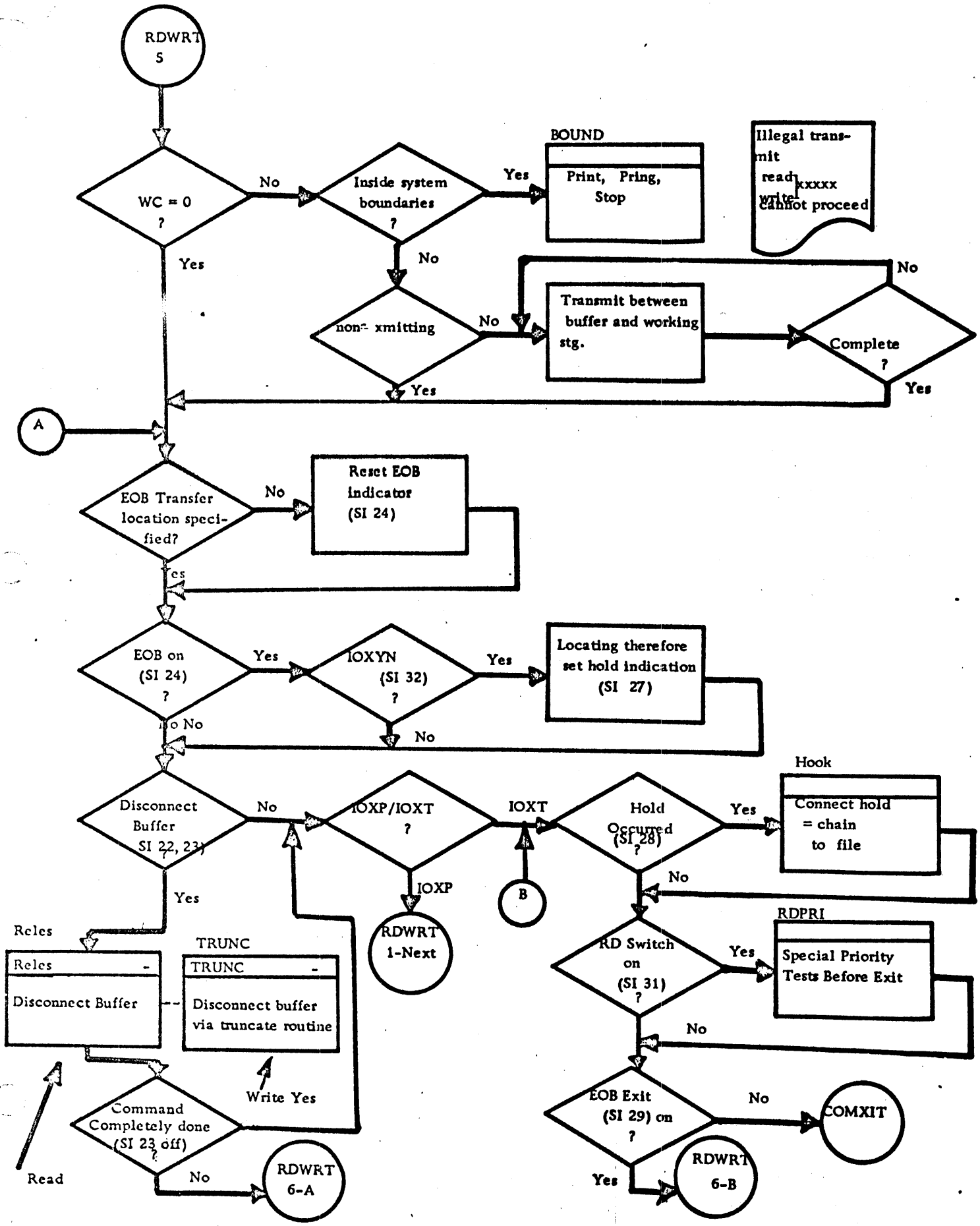

WRSLH
Write held buffer
Note: Writing buffer originally saved for this file will not affect buffer count
Code 7 (not last in chain) ?
No
WRSEL 1A
SEL5
Yes
BCKBB
Internal ?
No
Adjust buffer count
WRSEL 1B
C
Yes
Sel5.8
Locate Buffer
Place original buffer size in IORT (BCW2)
ENTIN
Sel 5.9
Enter buffer into regenerative chain
Advance block sequenc count
Exit at 1, 1


RDWRT
Master control routine for read and write
Locate Buffer in use, if none, locate pool
Is it minus ?
Yes
Disconnect held buffer ?
Yes
No
No
TRUNC
TRUNC
Disconnect buffer via truncate routine
Write
Reles
Reles
Release connected buffer
Read
Set INITIAL Conditions satisfied indicator (SI 25)
NEXT
GTIOX
Bring out IOCS command save transmit loc. and WDCT
Command
IOCD, TCH
IOCD or TCH
TCH
IOCD
Pickup location of next command
1st TCH in List ?
Yes
No
Prepare Exit Location
Change to IOCT, set disconnect and EOB indicators SI 24, 222
NEXT
IOCP, IOCT, IOSP, IOST, IORP, IORT
GT1
IA ?
Yes
No
Substitute direct address for indirect address
RDWRT 2

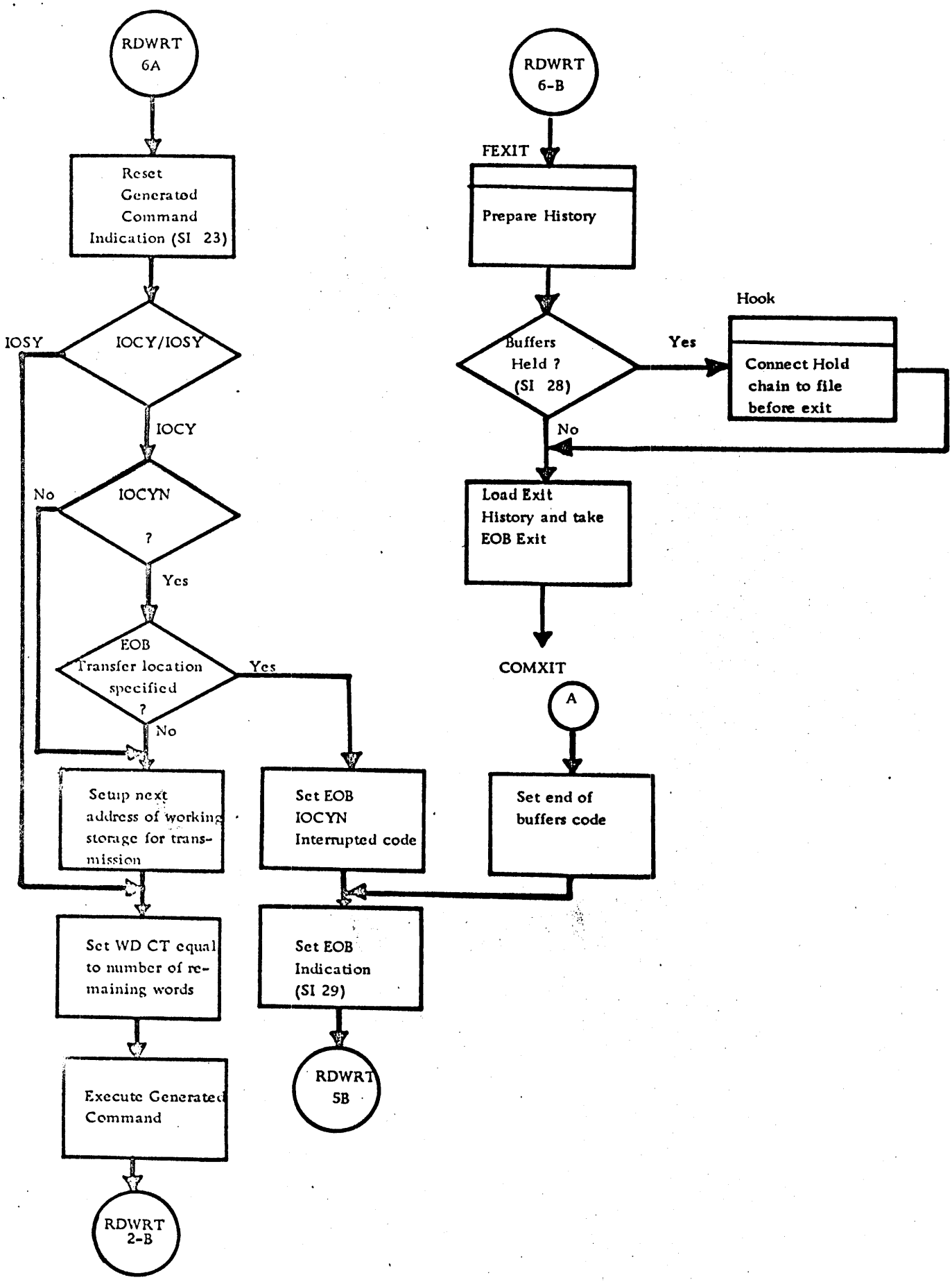

RDWRT 2
B
RWD
Command WDCT = 0 ?
Yes
(IOCY/IOSY) ?
Yes
RDWRT 5-A
No
No
GTBUF
GTBUF 1
GET buffer from Pool
Write
Pickup Buffer From FCBW 3
A
Sync
Sync 1
Get Next Buffer
Read
Pickup Buffer control word from FCBW 3
Buffer Connected ?
No
Yes
Buffer WC ≠ 0
No
Clear XRA
Yes
IOXY N ?
Yes
Set location of first available word to transmit location if none available set zero
No
Read
RSRD
IORY, IOSY/ IOCY
IORY,IOSY
Handler for IORY and IOSY reading
IOSY
RDWRT 3-A
IORY
RDWRT 3-B
Write
IOCY
RSWR
RSWR1
Handler for IORY and IOSY writing
RDWRT 3-C
RDWRT 3-B
RDWRT 5-A
RDWRT 3

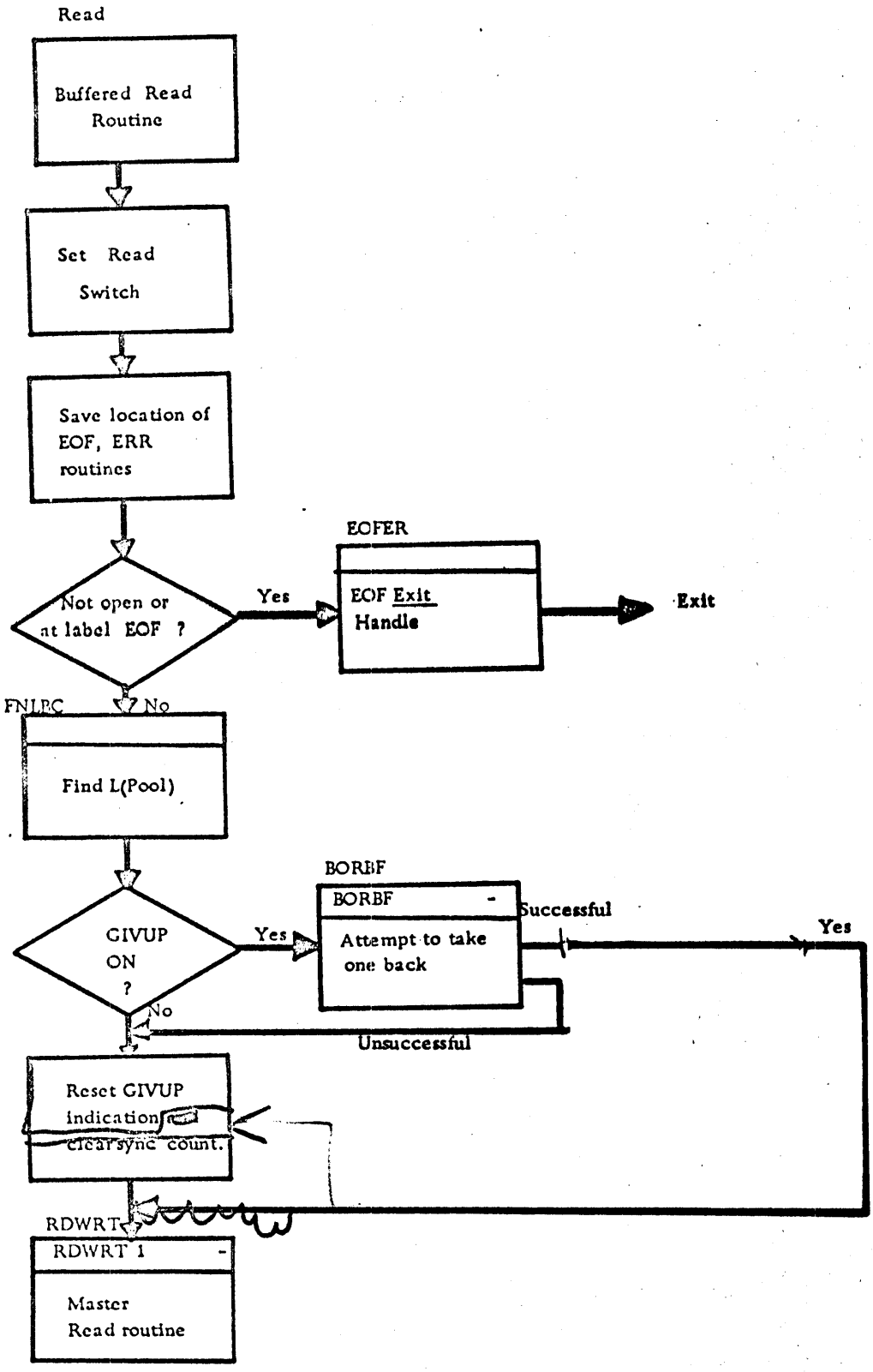

RDWRT 3
TRUNC
TRUNC -
Disconnect buffer via truncate routine
Write
Read
Reles
Reles -
Disconnect Buffer
RDWRT 2-A
Words Available ?
No
Yes
A
SET GENER BIT
RESET GENER BIT
Reduce available word count by command CT
RESULTING WC = 0 ?
Yes
No
Buffer contain enough words ?
No
Save number Required
IOSY ?
No
Set generated command indicator (SI 23)
Yes
B
Form count to execute on this buffer
Make number of available words = 0
Turn on EOB Indicator (SI 24)
C
IOSYN Write ?
No
Adjust Locator Decrement (BCW1)
Yes
Save number of available words in history
RDWRT 4


RDWRT
4
IOXYN
Yes
Place location of 1st word in command
Set IOXYN executed indicator (SI 32)
No
Compute address of last data word to be transmitted
IOSYN Write ?
No
Adjust locator address (BCW1)
Yes
Store address formed in transmit loop
Compute Address of last location to be transmitted to
Store address formed in transmit loop
Save location of last word + 1 for history
RDWRT
5

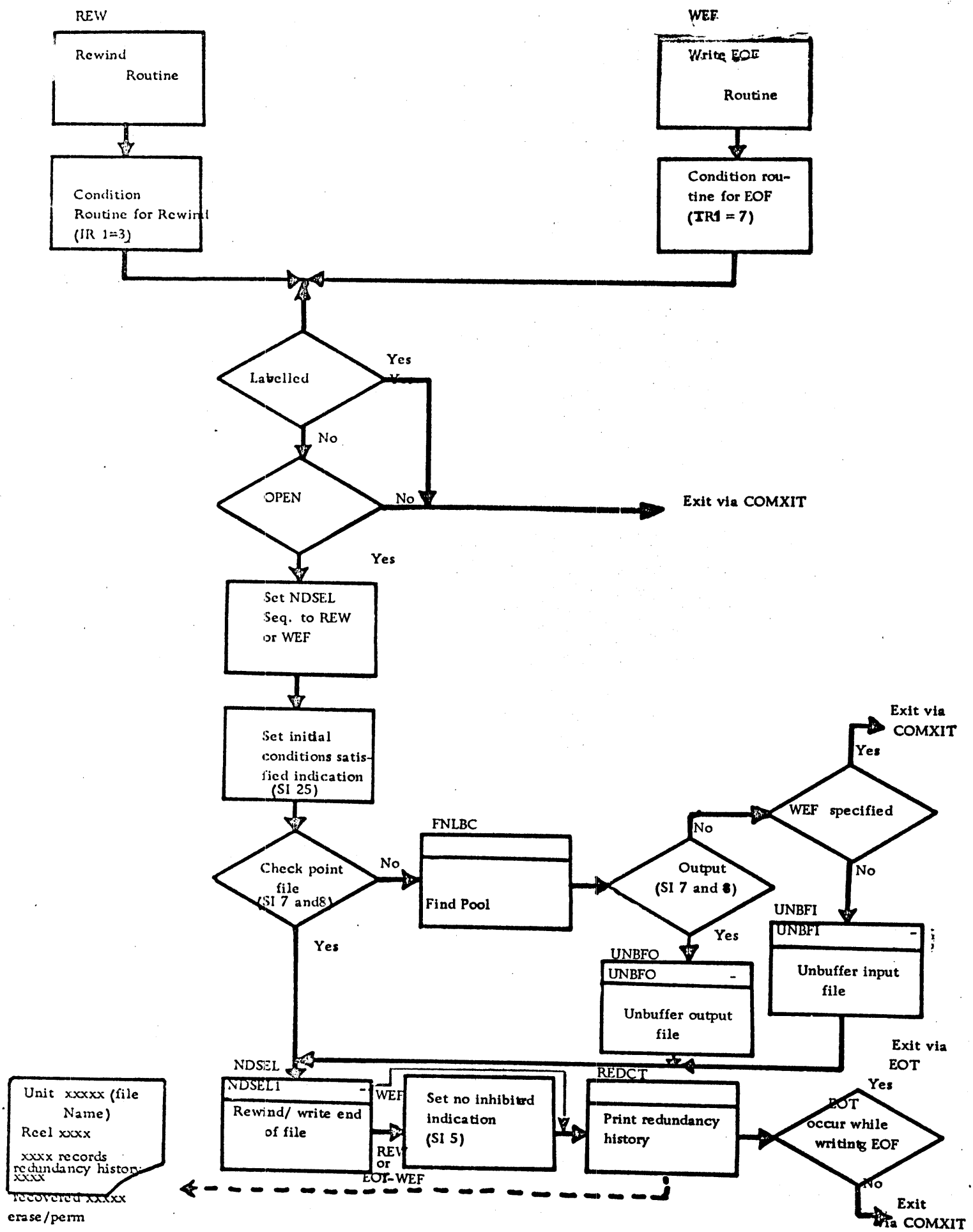

RDWRT 5
WC = 0 ?
No
Yes
Inside system boundaries ?
Yes
BOUND
Print, Pring, Stop
Illegal transmit read xxxxx write cannot proceed
No
non- xmitting
No
Yes
Transmit between buffer and working stg.
Complete ?
No
Yes
A
EOB Transfer location specified?
No
Reset EOB indicator (SI 24)
Yes
EOB on (SI 24) ?
Yes
IOXYN (SI 32) ?
Yes
Locating therefore set hold indication (SI 27)
No
No
Disconnect Buffer SI 22, 23 ?
No
IOXP/IOXT ?
IOXT
Hold Occurred (SI 28) ?
Yes
Hook
Connect hold = chain to file
No
Yes
IOXP
RDWRT 1-Next
B
Reles
Reles
Disconnect Buffer
TRUNC
TRUNC
Disconnect buffer via truncate routine
RDPRI
RD Switch on (SI 31) ?
Yes
Special Priority Tests Before Exit
No
Write Yes
Command Completely done (SI 23 off) ?
No
RDWRT 6-A
Read
EOB Exit (SI 29) on ?
No
COMXIT
Yes
RDWRT 6-B

RDWRT 5

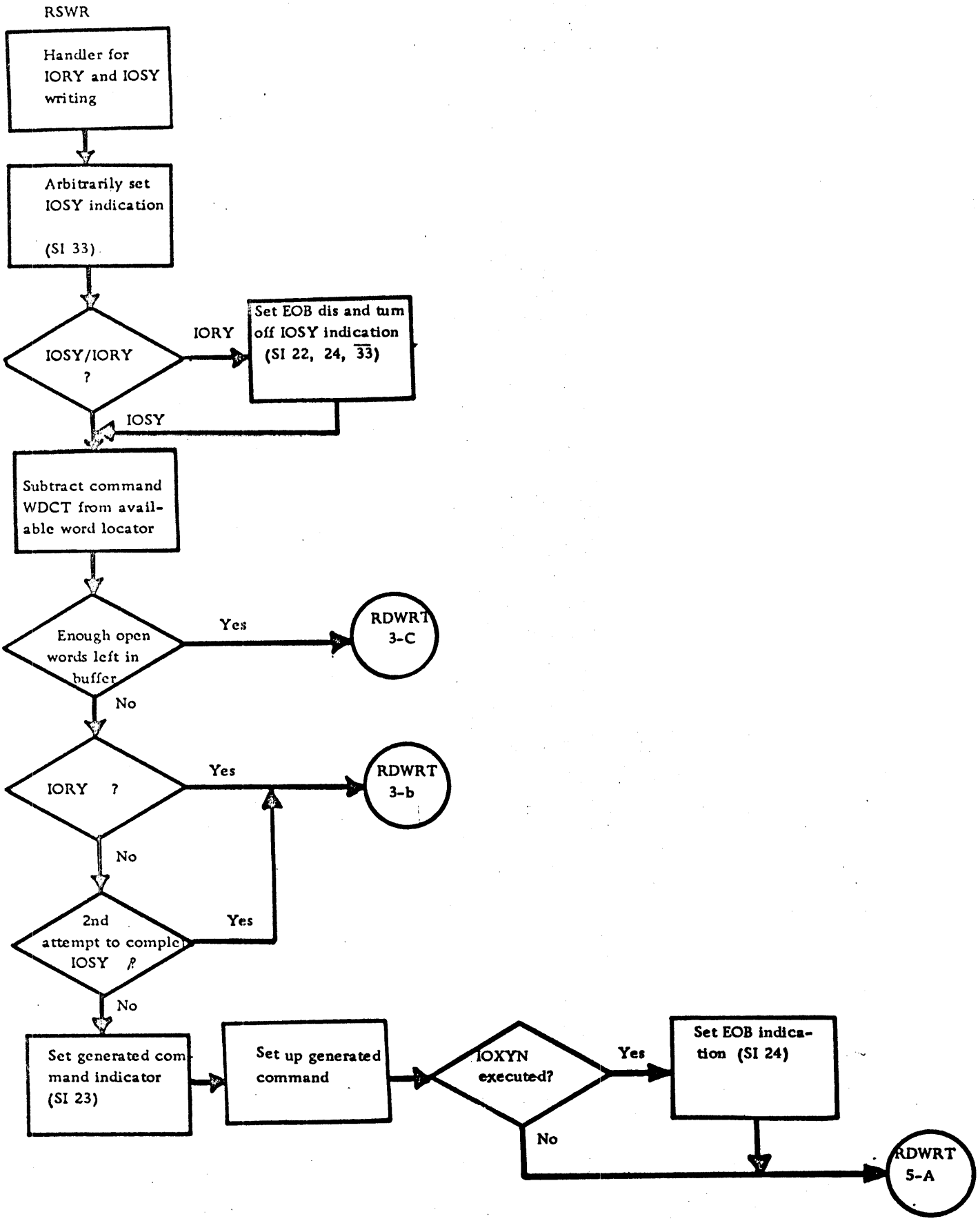

RDWRT 6A
Reset Generated Command Indication (SI 23)
IOSY
IOCY/IOSY
IOCY
No
IOCYN ?
Yes
EOB Transfer location specified ?
Yes
No
Setup next address of working storage for transmission
Set WD CT equal to number of remaining words
Execute Generated Command
RDWRT 2-B
Set EOB IOCYN Interrupted code
Set EOB Indication (SI 29)
RDWRT 5B
RDWRT 6-B
FEXIT
Prepare History
Buffers Held ? (SI 28)
Yes
Hook
Connect Hold chain to file before exit
No
Load Exit History and take EOB Exit
COMXIT
A
Set end of buffers code


Read
Buffered Read Routine
Set Read Switch
Save location of EOF, ERR routines
Not open or at label EOF ?
Yes
EOFER
EOF Exit Handle
Exit
No
FNLPC
Find L(Pool)
BORBF
BORBF -
Attempt to take one back
GIVUP ON ?
Yes
Successful
Yes
No
Unsuccessful
Reset GIVUP indication and clearsync count.
RDWRT
RDWRT 1 -
Master Read routine

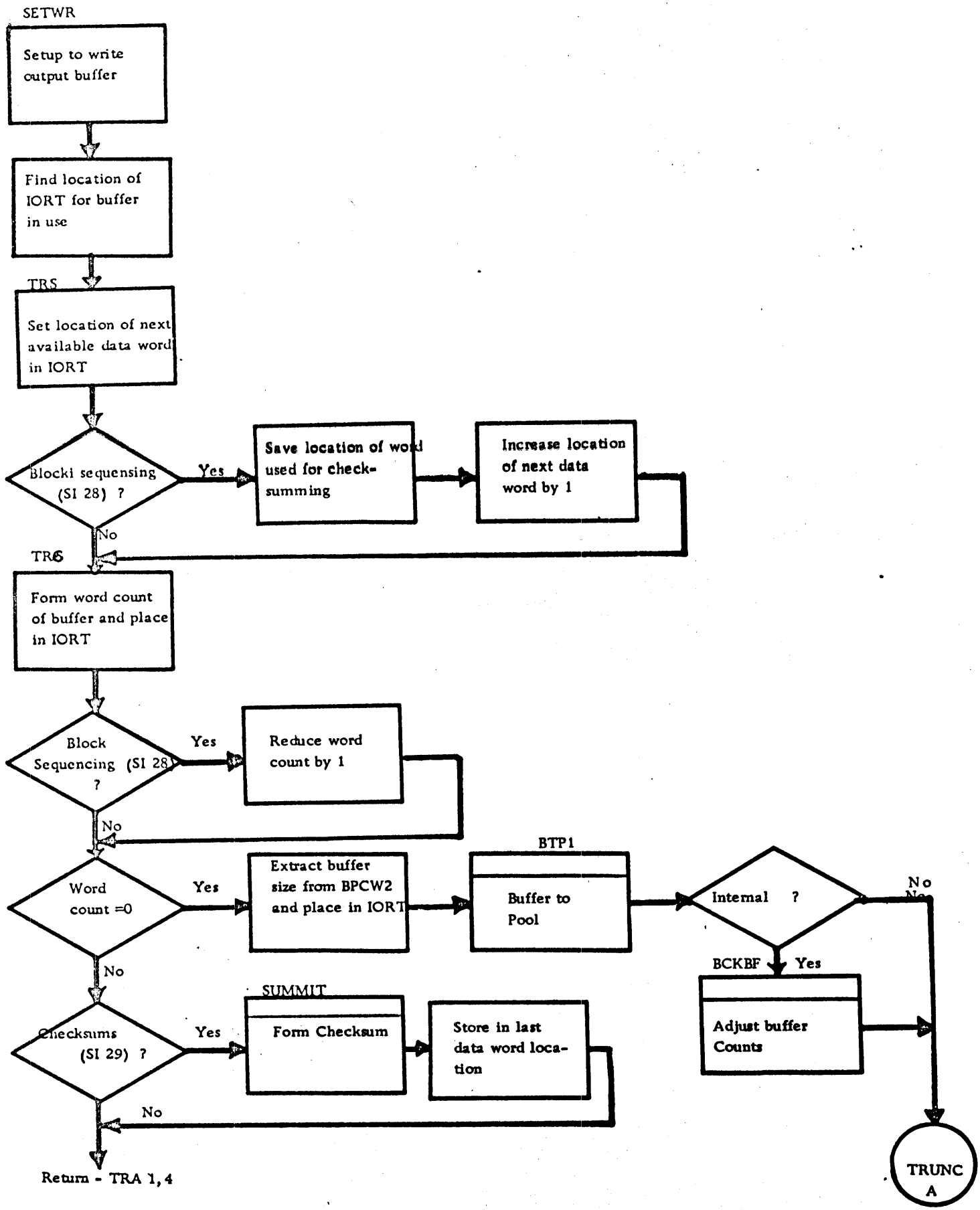
READ

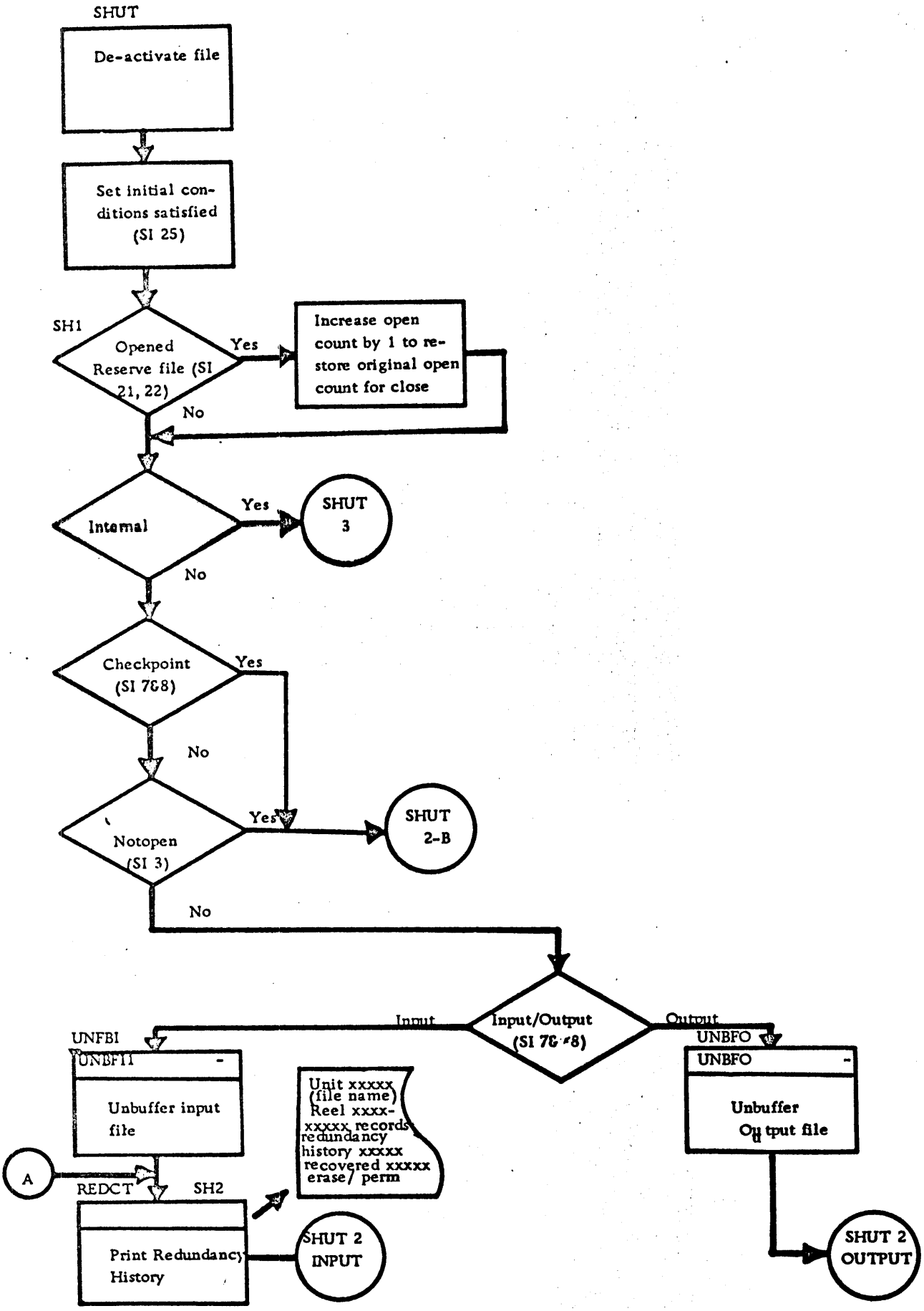

REW
Rewind Routine
Condition Routine for Rewind (IR 1=3)
WEF
Write EOF Routine
Condition routine for EOF (IR1 = 7)
Labelled
Yes
No
OPEN
No
Exit via COMXIT
Yes
Set NDSEL Seq. to REW or WEF
Set initial conditions satisfied indication (SI 25)
Check point file (SI 7 and8)
No
FNLBC
Find Pool
Output (SI 7 and 8)
No
Yes
WEF specified
Yes
Exit via COMXIT
No
UNBFI
UNBFI -
Unbuffer input file
UNBFO
UNBFO -
Unbuffer output file
Yes
Exit via EOT
NDSEL
NDSEL1 -
Rewind/ write end of file
WEF
REW or EOT-WEF
Set no inhibited indication (SI 5)
REDCT
Print redundancy history
EOT occur while writing EOF
Yes
No
Exit via COMXIT
Unit xxxxx (file Name)
Reel xxxx
xxxx records
redundancy histor
xxxx
recovered xxxxx
erase/perm

REW-WEF

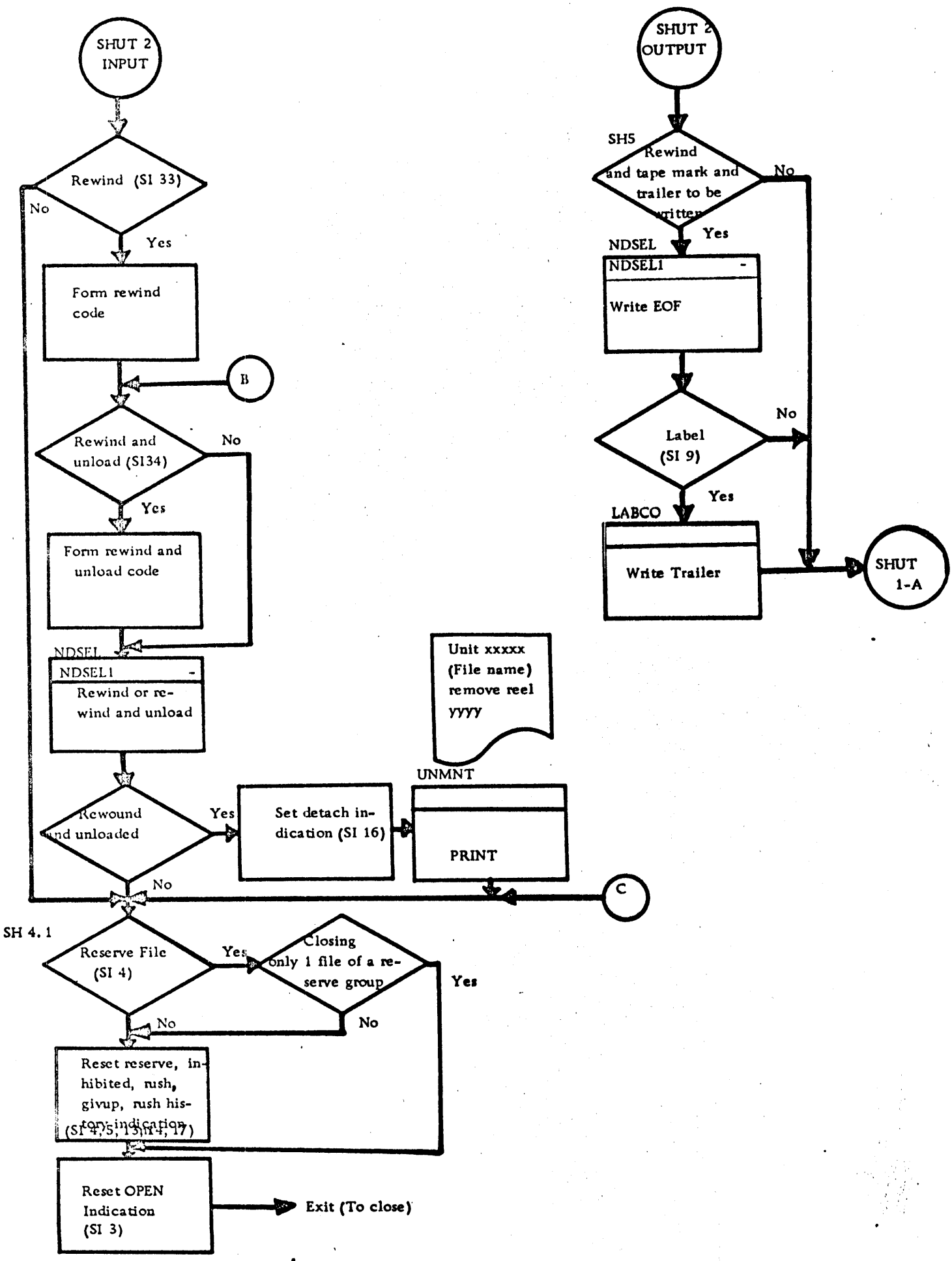

RSWR
Handler for IORY and IOSY writing
Arbitrarily set IOSY indication (SI 33)
IOSY/IORY ?
IORY
Set EOB dis and turn off IOSY indication (SI 22, 24, 33)
IOSY
Subtract command WDCT from available word locator
Enough open words left in buffer
Yes
RDWRT 3-C
No
IORY ?
Yes
RDWRT 3-b
No
2nd attempt to complete IOSY ?
Yes
No
Set generated command indicator (SI 23)
Set up generated command
IOXYN executed?
Yes
Set EOB indication (SI 24)
No
RDWRT 5-A

RSWR 1

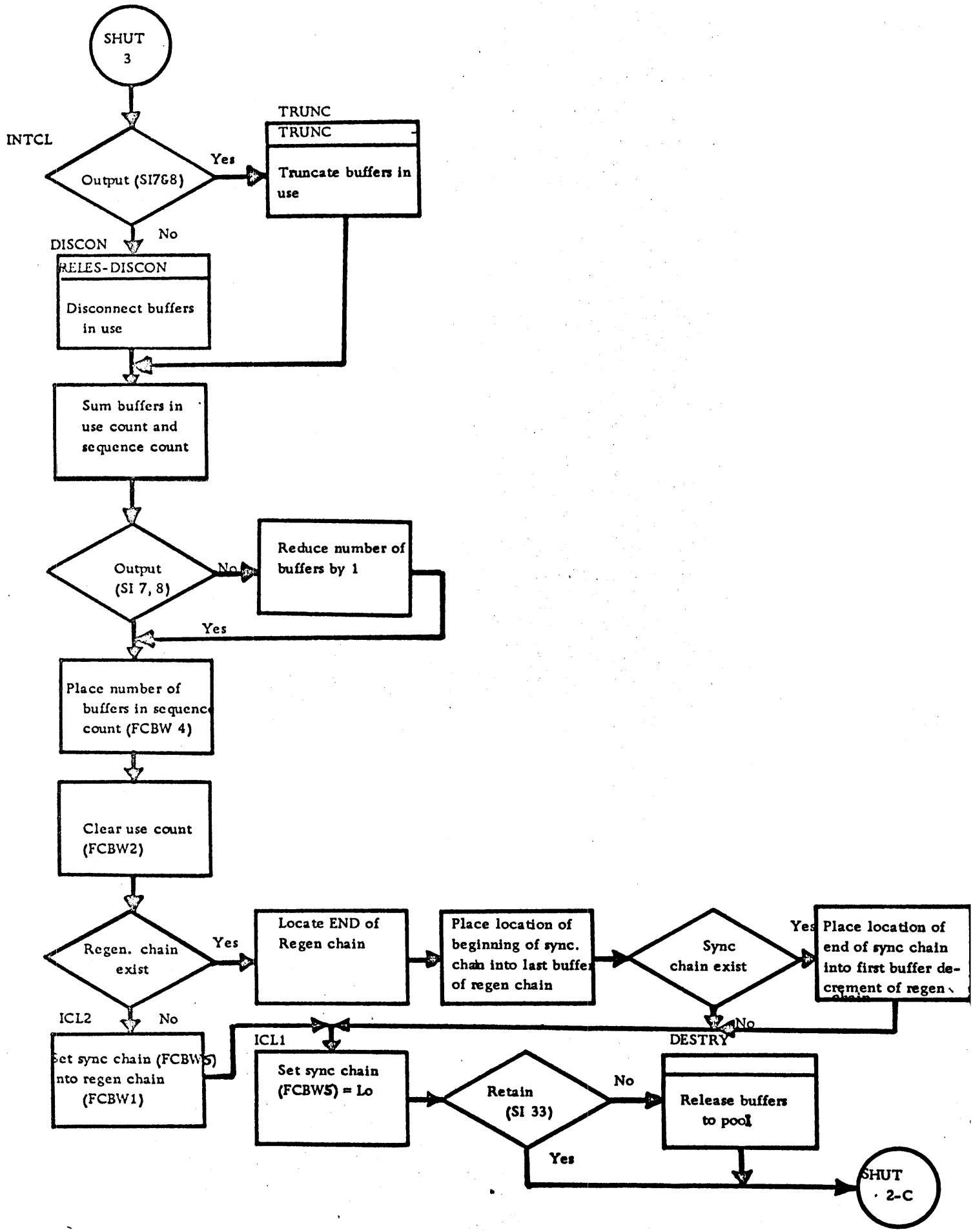

SETWR
Setup to write output buffer
Find location of IORT for buffer in use
TRS
Set location of next available data word in IORT
Block sequensing (SI 28) ?
Yes
Save location of word used for check-summing
Increase location of next data word by 1
No
TRS
Form word count of buffer and place in IORT
Block Sequencing (SI 28) ?
Yes
Reduce word count by 1
No
Word count =0
Yes
Extract buffer size from BPCW2 and place in IORT
BTP1
Buffer to Pool
Internal ?
No
BCKBF
Yes
Adjust buffer Counts
No
Checksums (SI 29) ?
Yes
SUMMIT
Form Checksum
Store in last data word location
No
Return - TRA 1, 4
TRUNC A

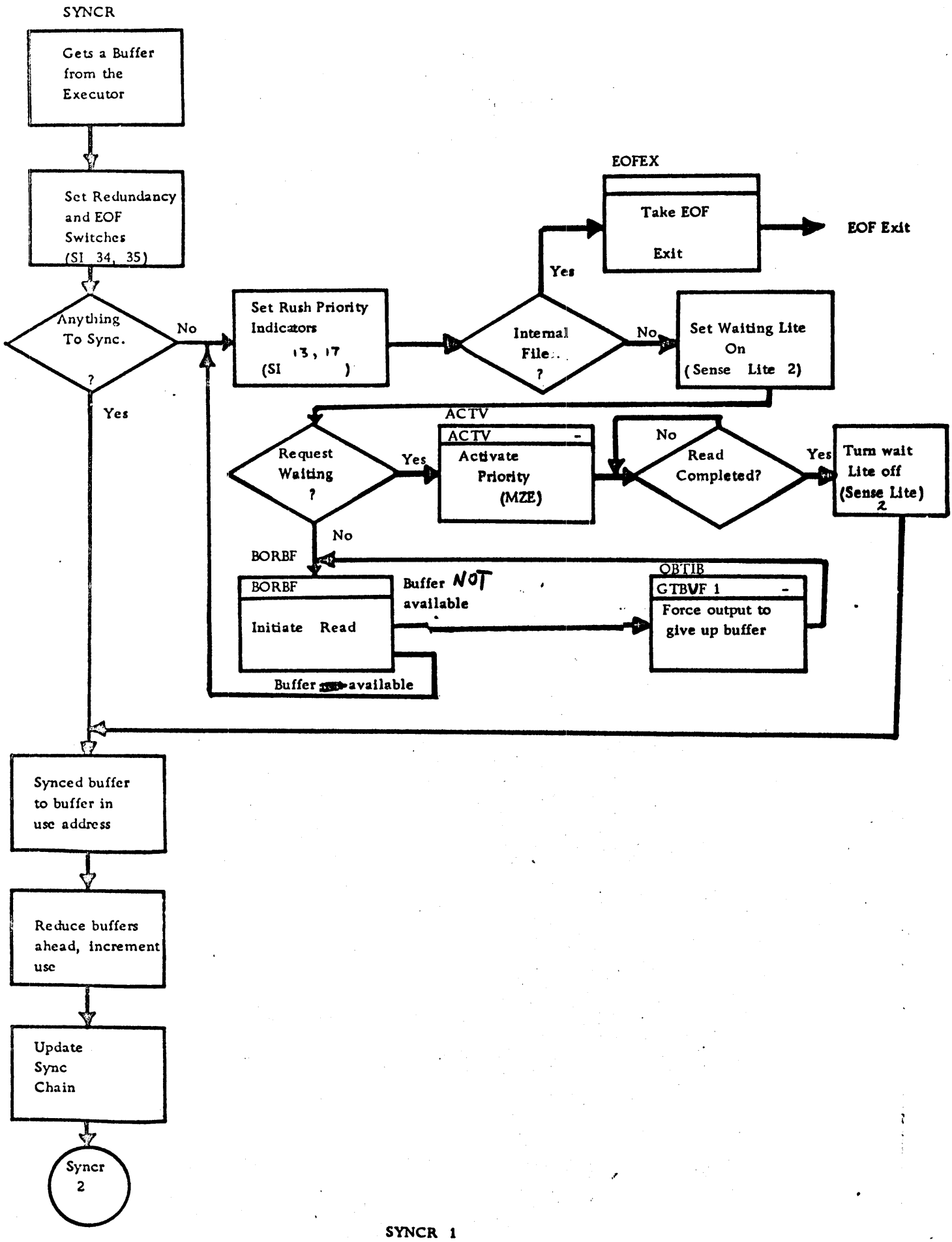

SHUT
De-activate file
Set initial conditions satisfied (SI 25)
SH1
Opened Reserve file (SI 21, 22)
Yes
Increase open count by 1 to restore original open count for close
No
Internal
Yes
SHUT 3
No
Checkpoint (SI 7&8)
Yes
No
Notopen (SI 3)
Yes
SHUT 2-B
No
Input/Output (SI 7&8)
Input
Output
UNFBI
UNBF11
Unbuffer input file
A
REDCT
SH2
Print Redundancy History
Unit xxxxx (file name) Reel xxxx-xxxxx records redundancy history xxxxx recovered xxxxx erase/ perm
SHUT 2 INPUT
UNBFO
UNBFO
Unbuffer Output file
SHUT 2 OUTPUT

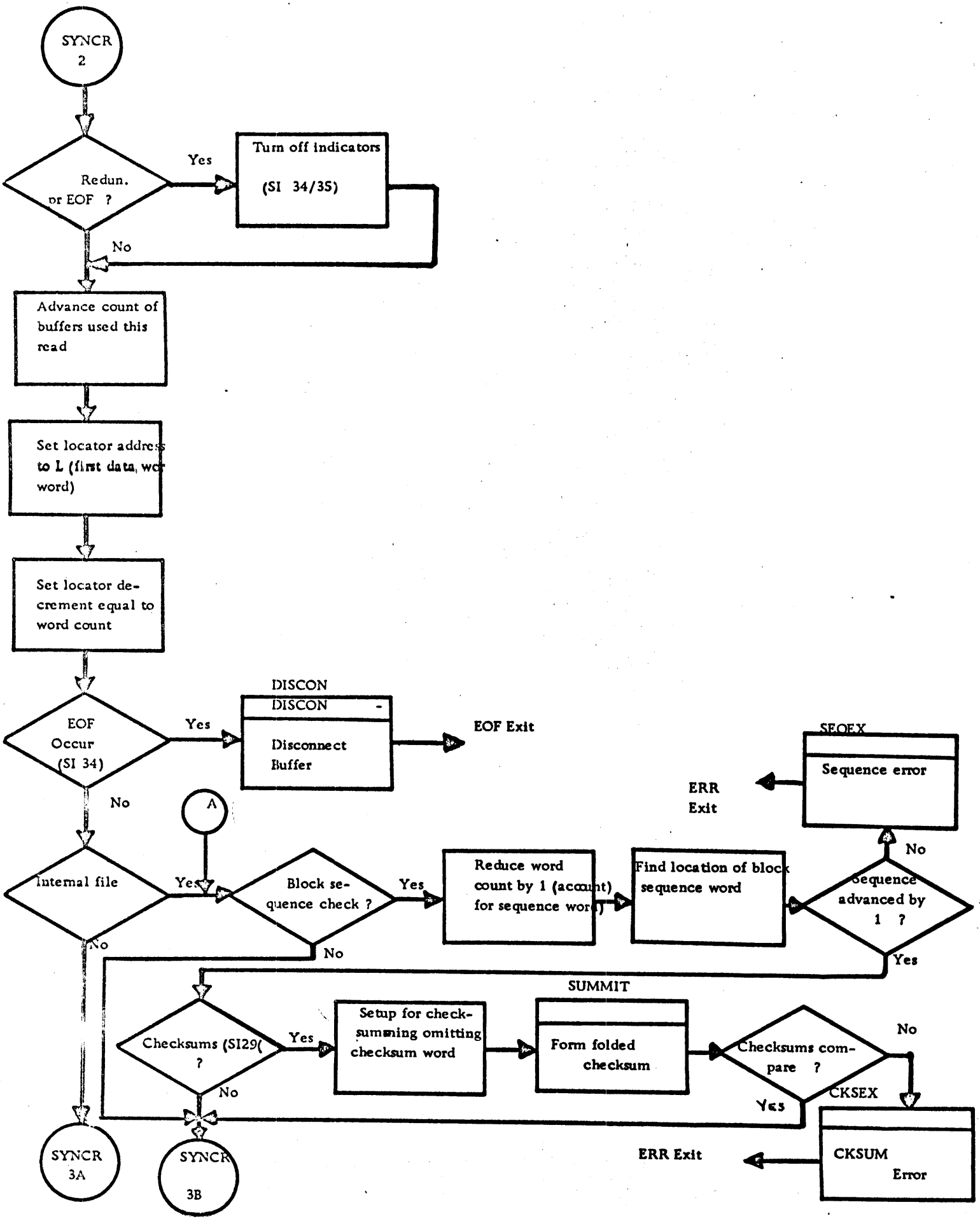
SHUT 1

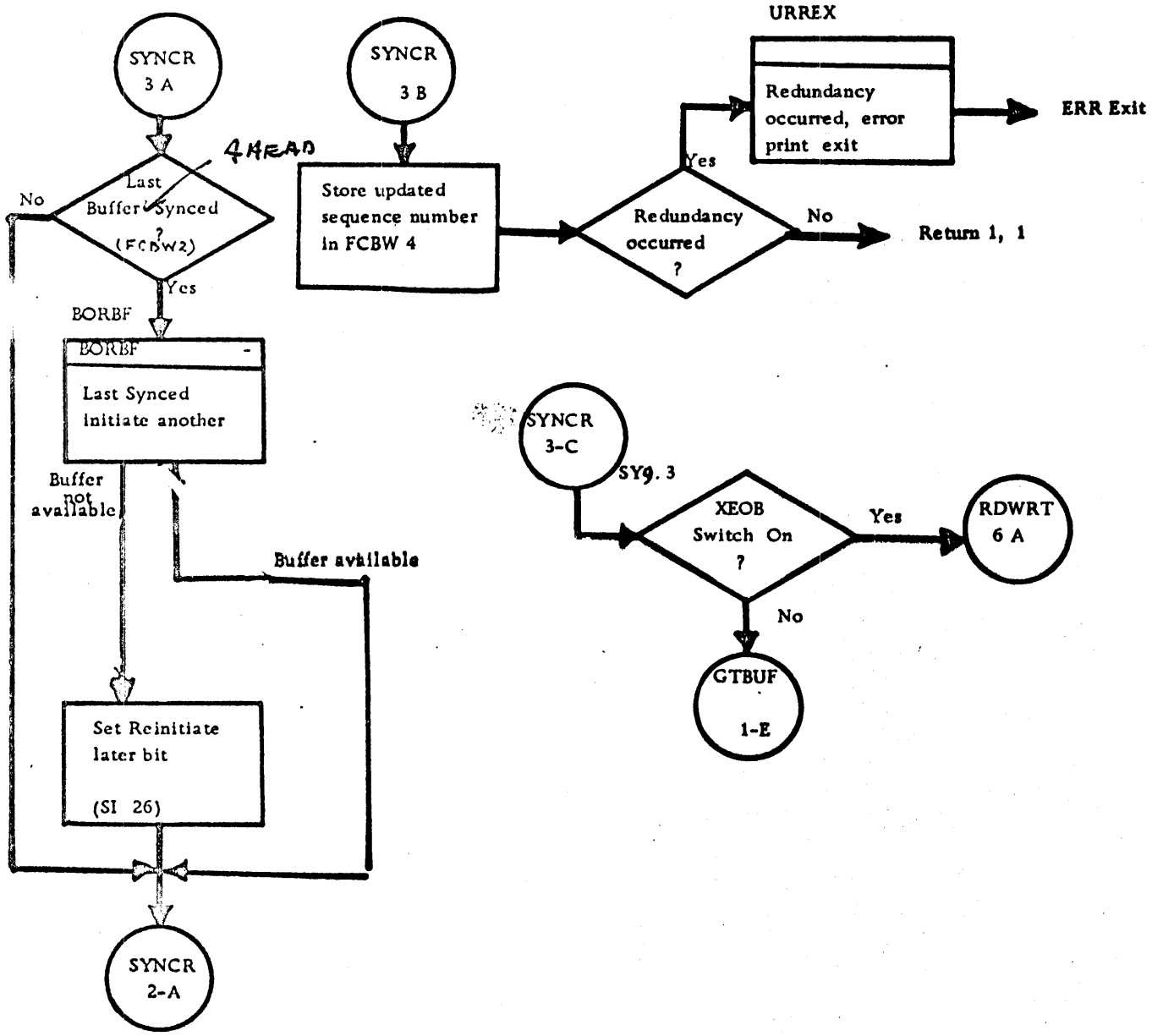

SHUT 2 INPUT
Rewind (SI 33)
No
Yes
Form rewind code
B
Rewind and unload (SI34)
No
Yes
Form rewind and unload code
NDSEL
NDSEL1 -
Rewind or rewind and unload
Unit xxxxx (File name) remove reel yyyy
UNMNT
Rewound and unloaded
Yes
Set detach indication (SI 16)
PRINT
No
C
SH 4.1
Reserve File (SI 4)
Yes
Closing only 1 file of a reserve group
Yes
No
No
Reset reserve, inhibited, rush, givup, rush history indication (SI 4,5,13,14,17)
Reset OPEN Indication (SI 3)
Exit (To close)
SHUT 2 OUTPUT
SH5
Rewind and tape mark and trailer to be written
No
Yes
NDSEL
NDSEL1 -
Write EOF
Label (SI 9)
No
Yes
LABCO
Write Trailer
SHUT 1-A

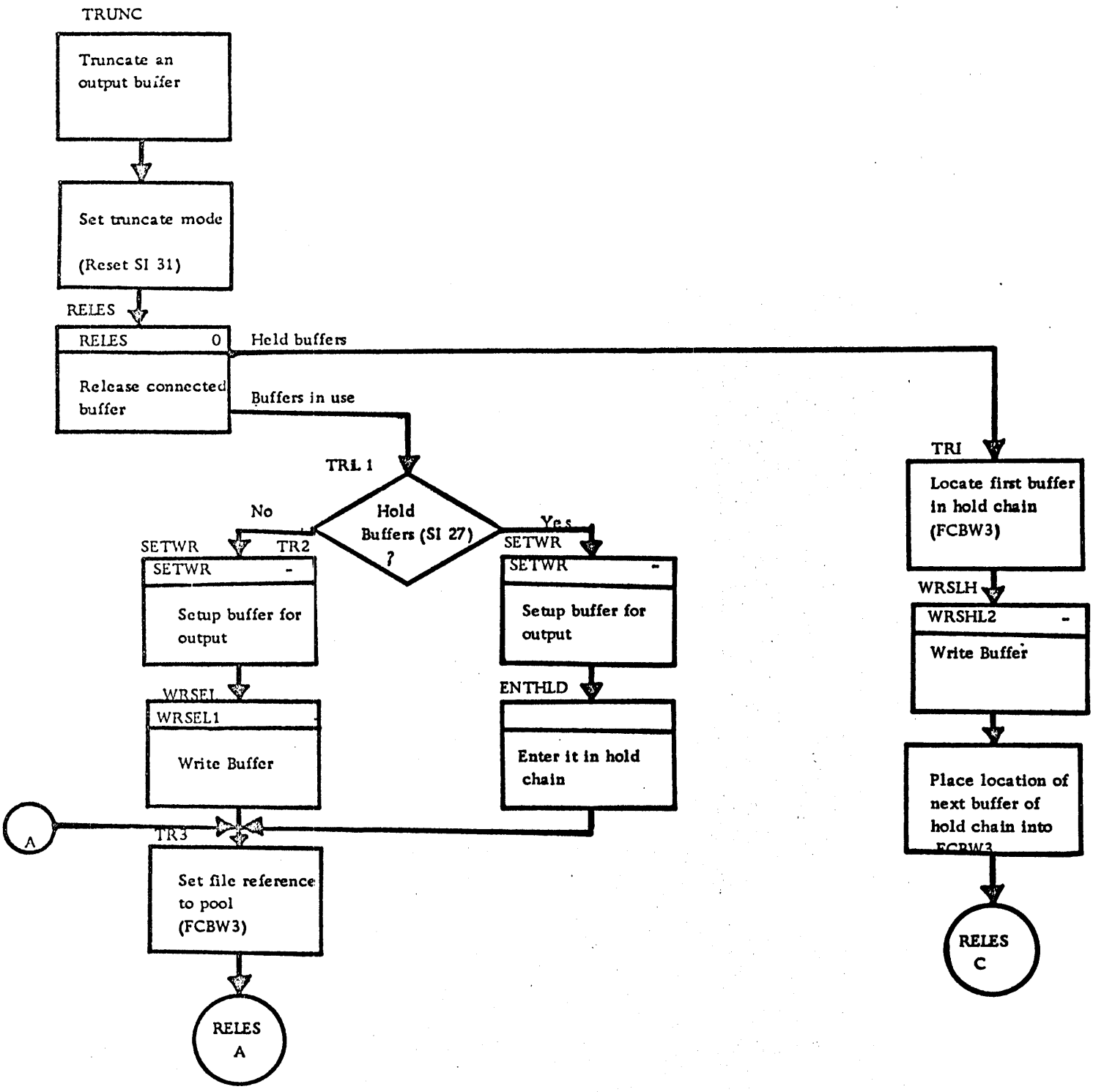
SHUT 2


SHUT 3
INTCL
Output (SI7&8)
Yes
TRUNC
TRUNC
Truncate buffers in use
No
DISCON
RELES-DISCON
Disconnect buffers in use
Sum buffers in use count and sequence count
Output (SI 7, 8)
No
Reduce number of buffers by 1
Yes
Place number of buffers in sequence count (FCBW 4)
Clear use count (FCBW2)
Regen. chain exist
Yes
Locate END of Regen chain
Place location of beginning of sync. chain into last buffer of regen chain
Sync chain exist
Yes
Place location of end of sync chain into first buffer decrement of regen
No
DESTRY
ICL2
No
Set sync chain (FCBW5) into regen chain (FCBW1)
ICL1
Set sync chain (FCBW5) = Lo
Retain (SI 33)
No
Release buffers to pool
Yes
SHUT 2-C

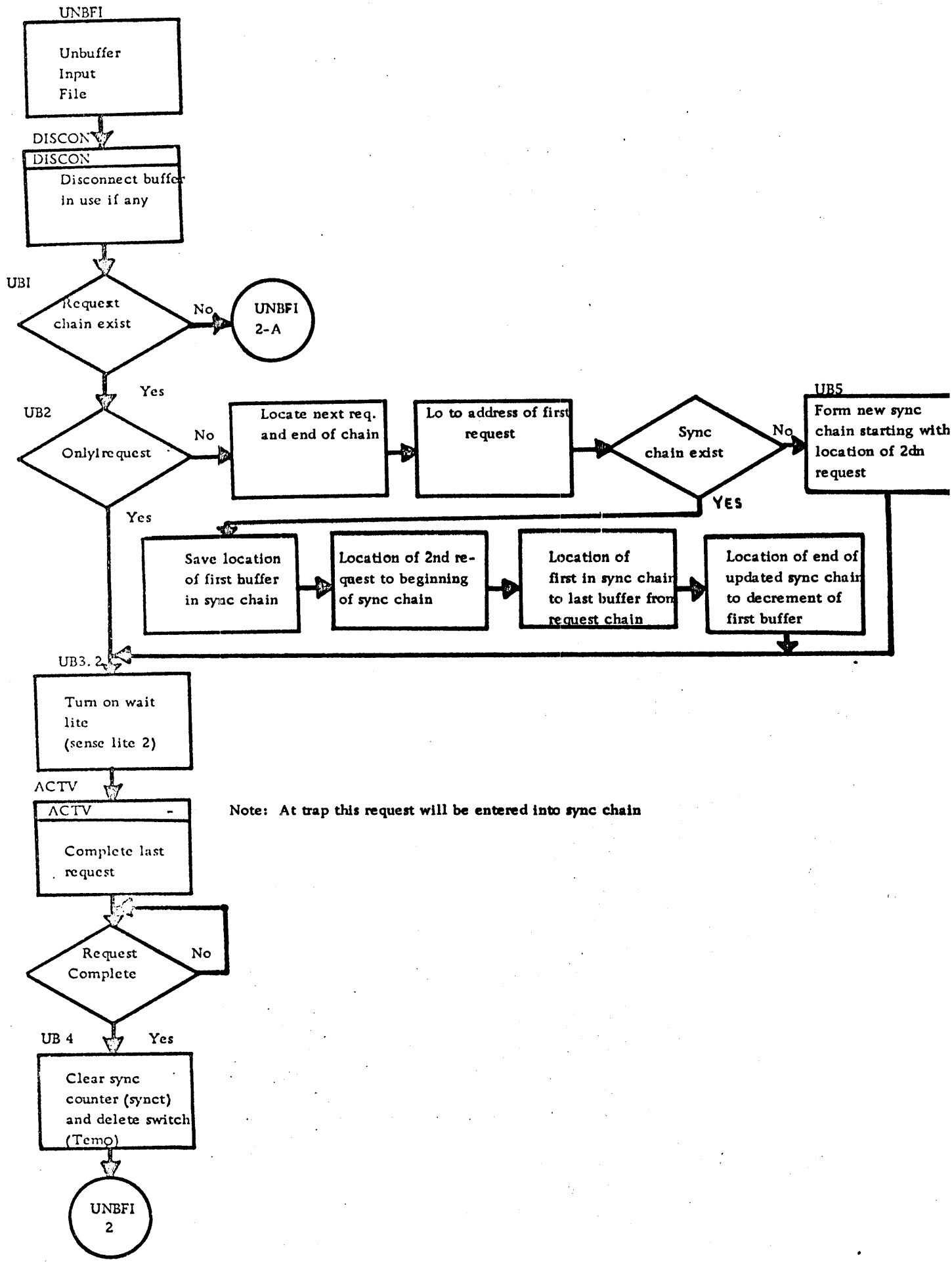

SYNCR
Gets a Buffer from the Executor
Set Redundancy and EOF Switches (SI 34, 35)
Anything To Sync. ?
No
Yes
Set Rush Priority Indicators (SI 13, 17)
Internal File ?
Yes
No
EOFEX
Take EOF
Exit
EOF Exit
Set Waiting Lite On (Sense Lite 2)
ACTV
Request Waiting ?
Yes
No
ACTV -
Activate Priority (MZE)
Read Completed?
Yes
No
Turn wait Lite off (Sense Lite) 2
BORBF
BORBF
Initiate Read
Buffer NOT available
Buffer available
OBTIB
GTBUF 1 -
Force output to give up buffer
Synced buffer to buffer in use address
Reduce buffers ahead, increment use
Update Sync Chain
Syncr 2


SYNCR 2
Redun. or EOF ?
Yes
Turn off indicators (SI 34/35)
No
Advance count of buffers used this read
Set locator address to L (first data word)
Set locator decrement equal to word count
EOF Occur (SI 34)
Yes
DISCON
DISCON
Disconnect Buffer
EOF Exit
No
A
Internal file
Yes
No
Block sequence check ?
Yes
No
Reduce word count by 1 (account for sequence word)
Find location of block sequence word
Sequence advanced by 1 ?
No
Yes
SEQEX
Sequence error
ERR Exit
SUMMIT
Checksums (SI29(?
Yes
No
Setup for checksumming omitting checksum word
Form folded checksum
Checksums compare ?
No
Yes
CKSEX
CKSUM Error
ERR Exit
SYNCR 3A
SYNCR 3B

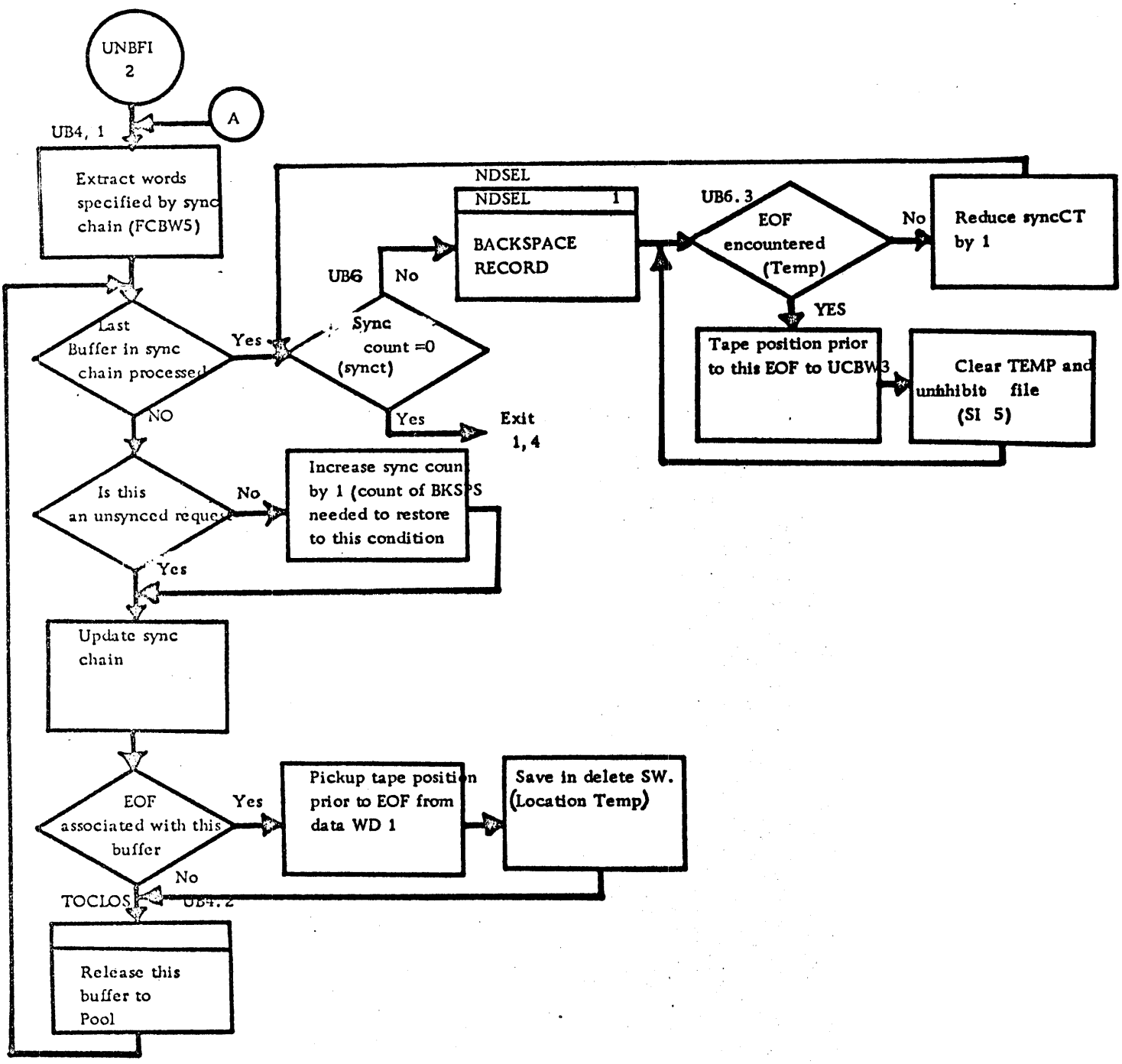

SYNCR 3 A
SYNCR 3 B
4 READ
Last Buffer Synced (FCBW2) ?
No
Yes
Store updated sequence number in FCBW 4
Redundancy occurred ?
Yes
No
URREX
Redundancy occurred, error print exit
ERR Exit
Return 1, 1
BORBF
BORBF
Last Synced initiate another
Buffer not available
Buffer available
Set Reinitiate later bit
(SI 26)
SYNCR 2-A
SYNCR 3-C
SY9. 3
XEOB Switch On ?
Yes
No
RDWRT 6 A
GTBUF 1-E

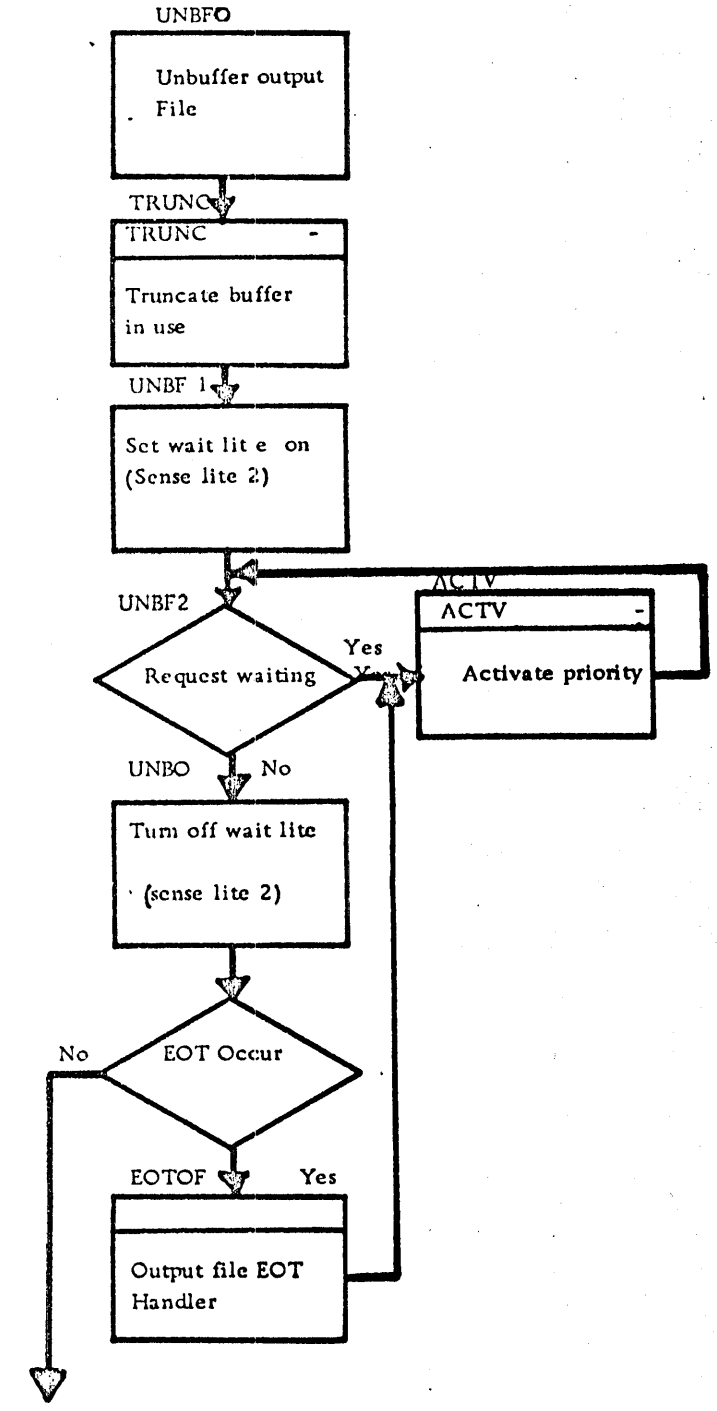

TRUNC
Truncate an output buffer
Set truncate mode
(Reset SI 31)
RELES
RELES 0
Release connected buffer
Held buffers
Buffers in use
TRI
Locate first buffer in hold chain (FCBW3)
WRSLH
WRSHL2 -
Write Buffer
Place location of next buffer of hold chain into FCBW3
RELES C
TR1 1
Hold Buffers (SI 27) ?
No
Yes
SETWR
TR2
SETWR -
Setup buffer for output
WRSEL
WRSEL1
Write Buffer
SETWR
SETWR -
Setup buffer for output
ENTHLD
Enter it in hold chain
A
TR3
Set file reference to pool (FCBW3)
RELES A

TRUNC

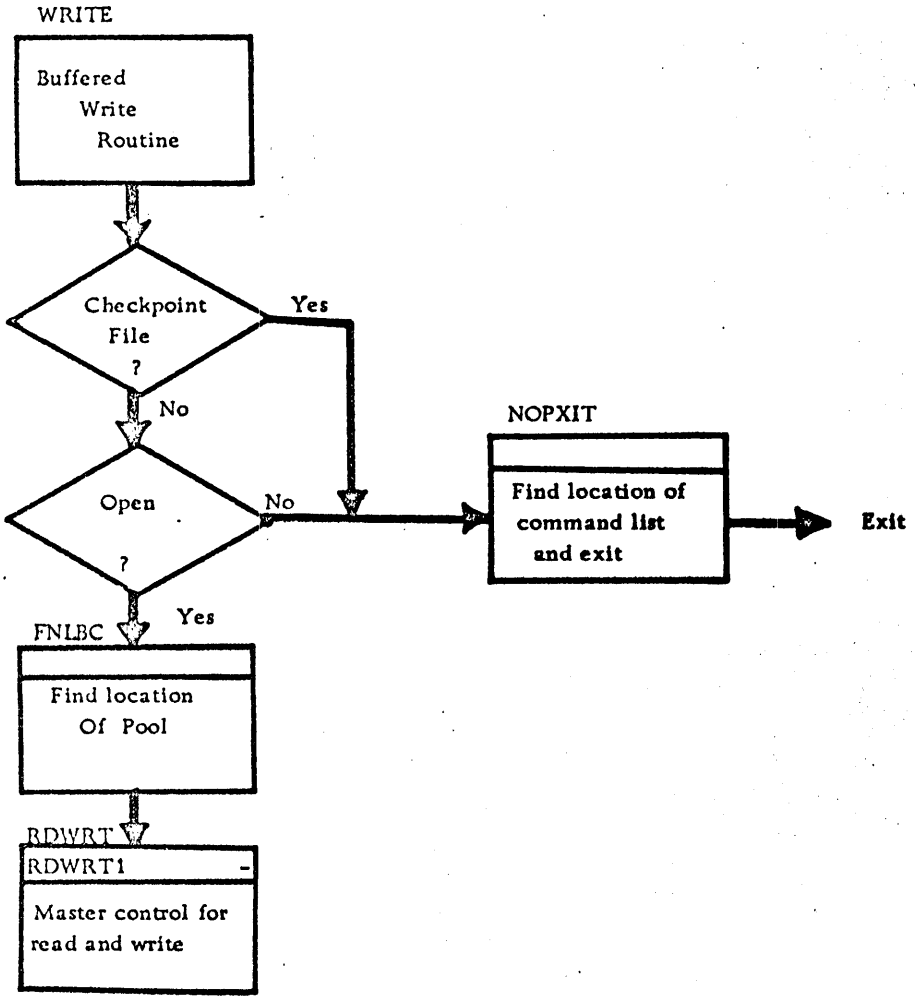


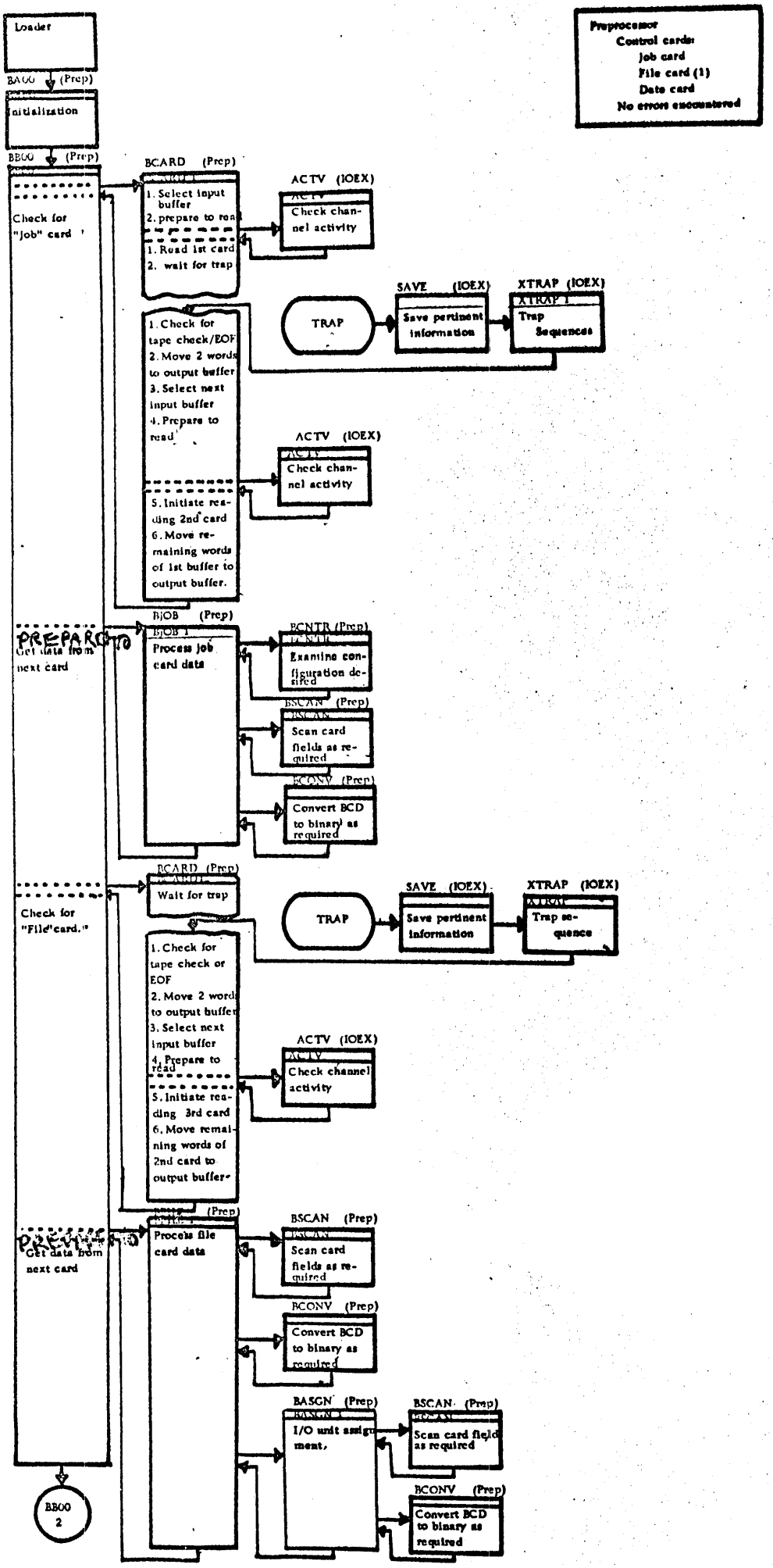
Note: At trap this request will be entered into sync chain

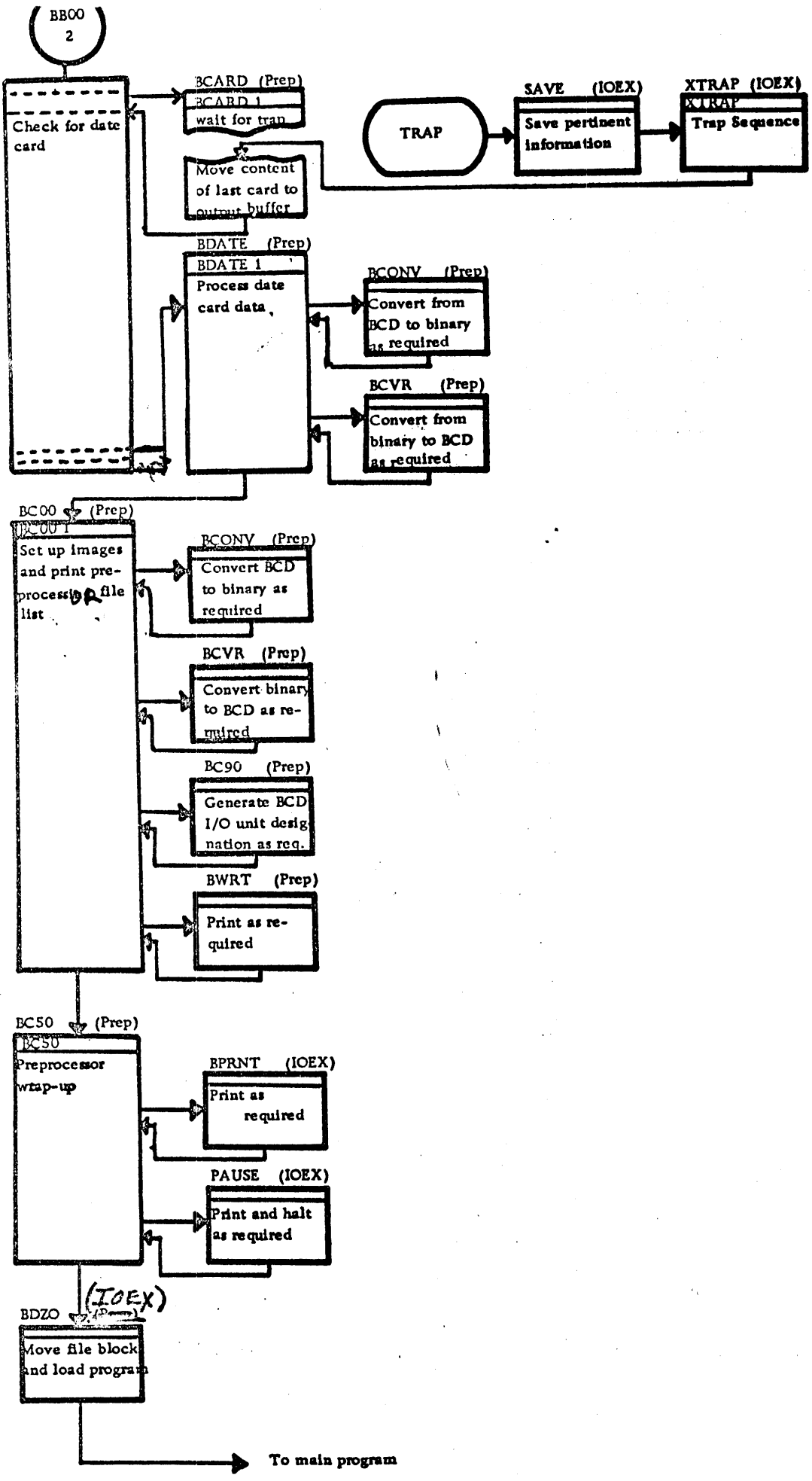

UNBFI
2
A
UB4, 1
Extract words specified by sync chain (FCBW5)
Last Buffer in sync chain processed
Yes
NO
UB6
No
Sync count =0 (synct)
Yes
Exit 1, 4
NDSEL
NDSEL 1
BACKSPACE RECORD
UB6.3
EOF encountered (Temp)
No
Reduce syncCT by 1
YES
Tape position prior to this EOF to UCBW3
Clear TEMP and unhhibit file (SI 5)
Is this an unsynced request
No
Increase sync coun by 1 (count of BKSPS needed to restore to this condition
Yes
Update sync chain
EOF associated with this buffer
Yes
Pickup tape position prior to EOF from data WD 1
Save in delete SW. (Location Temp)
No
TOCLOS
UB4.2
Release this buffer to Pool

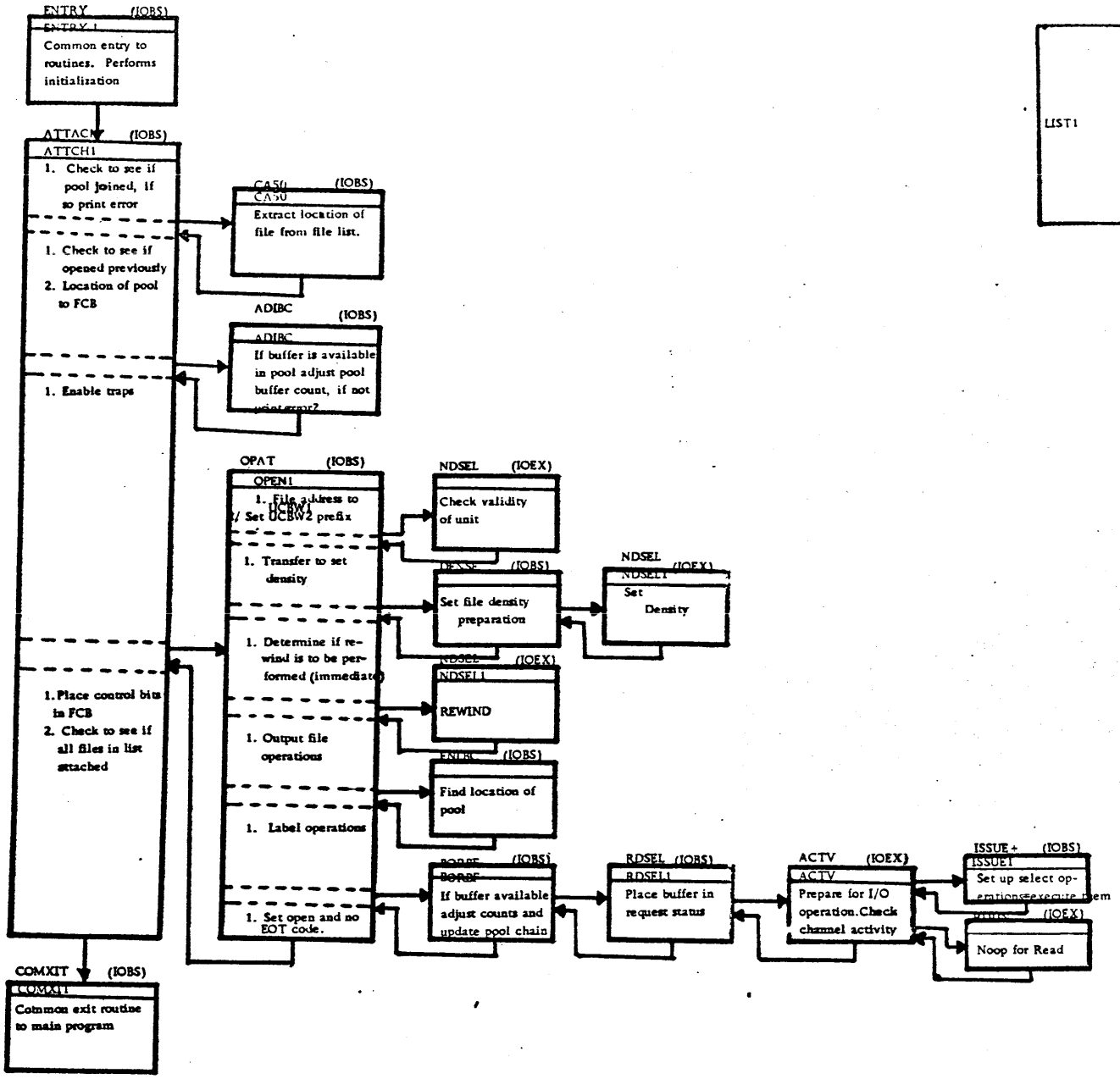

UNBFO
Unbuffer output File
TRUNC
TRUNC
Truncate buffer in use
UNBF 1
Set wait lit e on (Sense lite 2)
ACTV
UNBF2
ACTV
Request waiting
Yes
Activate priority
UNBO
No
Turn off wait lite (sense lite 2)
No
EOT Occur
EOTOF
Yes
Output file EOT Handler

Exit 1, 4

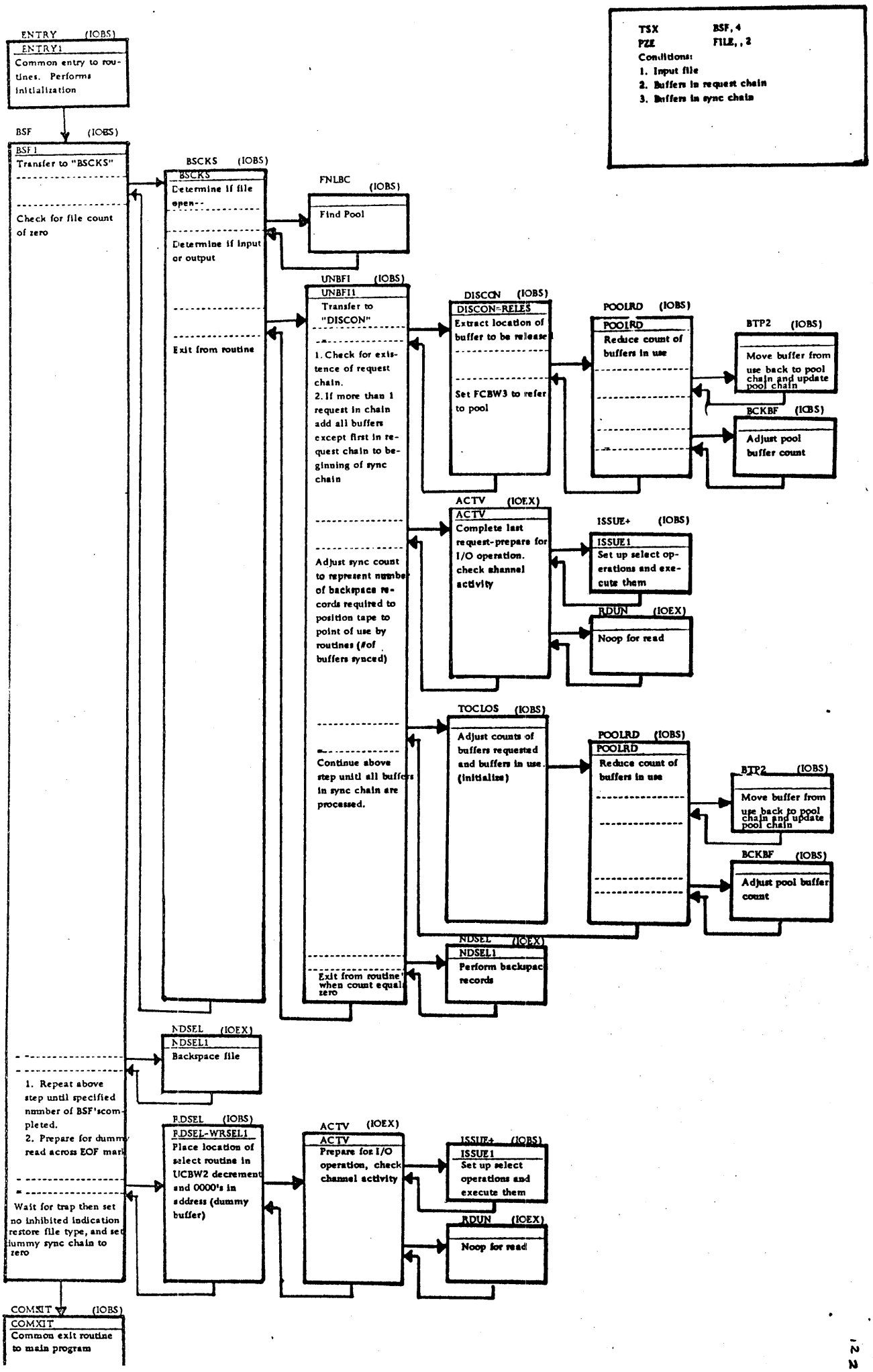

WRITE
Buffered
Write
Routine
Checkpoint
File
?
Yes
No
Open
?
No
NOPXIT
Find location of
command list
and exit
Exit
Yes
FNLBC
Find location
Of Pool
RDWRT
RDWRT1
Master control for
read and write

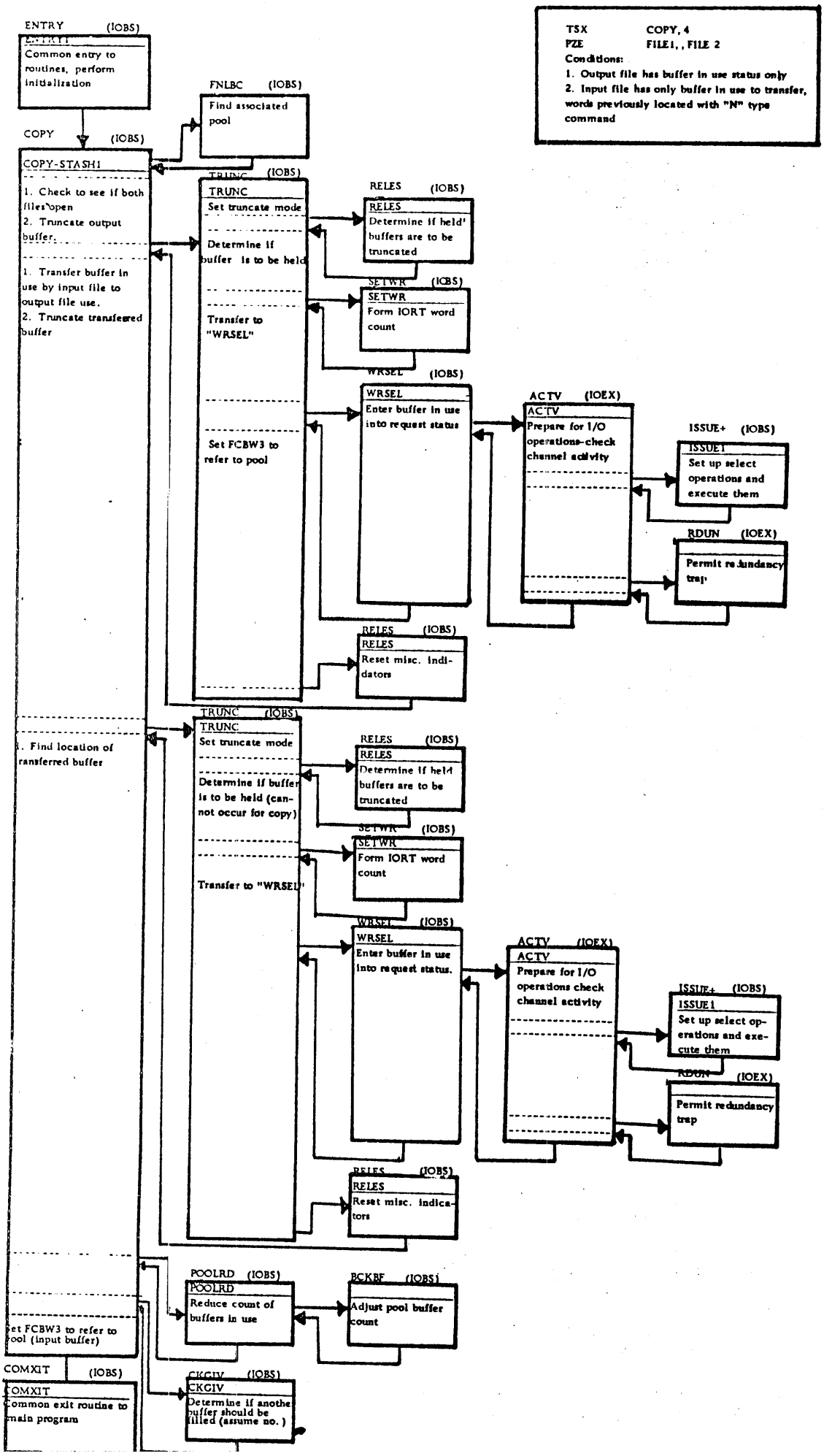

Loader
BA00 (Prep)
Initialization
BB00 (Prep)
Check for "job" card
BCARD (Prep)
1. Select input buffer
2. prepare to read
1. Read 1st card
2. wait for trap
ACTV (IOEX)
Check channel activity
TRAP
SAVE (IOEX)
Save pertinent information
XTRAP (IOEX)
Trap Sequences
1. Check for tape check/EOF
2. Move 2 words to output buffer
3. Select next input buffer
4. Prepare to read
5. Initiate reading 2nd card
6. Move remaining words of 1st buffer to output buffer.
ACTV (IOEX)
Check channel activity
PREPARE TO
Get data from next card
BJOB (Prep)
Process job card data
BCNTR (Prep)
Examine configuration defined
BSCAN (Prep)
Scan card fields as required
BCONV (Prep)
Convert BCD to binary as required
BCARD (Prep)
Wait for trap
TRAP
SAVE (IOEX)
Save pertinent information
XTRAP (IOEX)
Trap sequence
Check for "File" card.
1. Check for tape check or EOF
2. Move 2 words to output buffer
3. Select next input buffer
4. Prepare to read
5. Initiate reading 3rd card
6. Move remaining words of 2nd card to output buffer
ACTV (IOEX)
Check channel activity
(Prep)
Process file card data
BSCAN (Prep)
Scan card fields as required
BCONV (Prep)
Convert BCD to binary as required
BASGN (Prep)
I/O unit assignment,
BSCAN (Prep)
Scan card field as required
BCONV (Prep)
Convert BCD to binary as required
BB00 2
Preprocessor
Control cards:
job card
File card (1)
Data card
No errors encountered

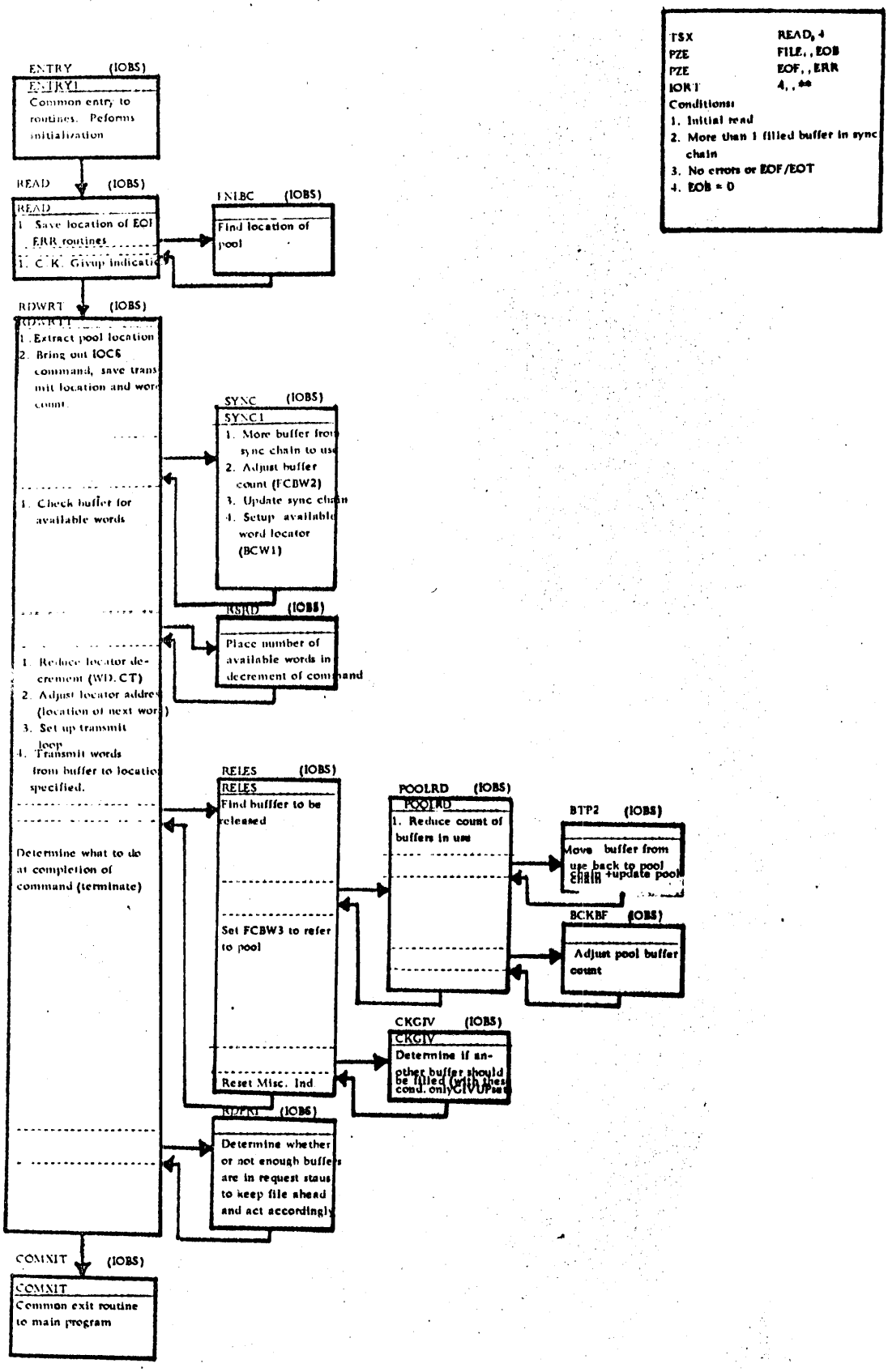

BB00
2
Check for date card
BCARD (Prep)
BCARD 1
wait for trap
Move content of last card to output buffer
TRAP
SAVE (IOEX)
Save pertinent information
XTRAP (IOEX)
XTRAP
Trap Sequence
BDATE (Prep)
BDATE 1
Process date card data
BCONV (Prep)
Convert from BCD to binary as required
BCVR (Prep)
Convert from binary to BCD as required
BC00 (Prep)
BC00 1
Set up images and print pre-processor file list
BCONV (Prep)
Convert BCD to binary as required
BCVR (Prep)
Convert binary to BCD as required
BC90 (Prep)
Generate BCD I/O unit designation as req.
BWRT (Prep)
Print as required
BC50 (Prep)
BC50
Preprocessor wrap-up
BPRNT (IOEX)
Print as required
PAUSE (IOEX)
Print and halt as required
(IOEX)
BDZO
Move file block and load program
To main program

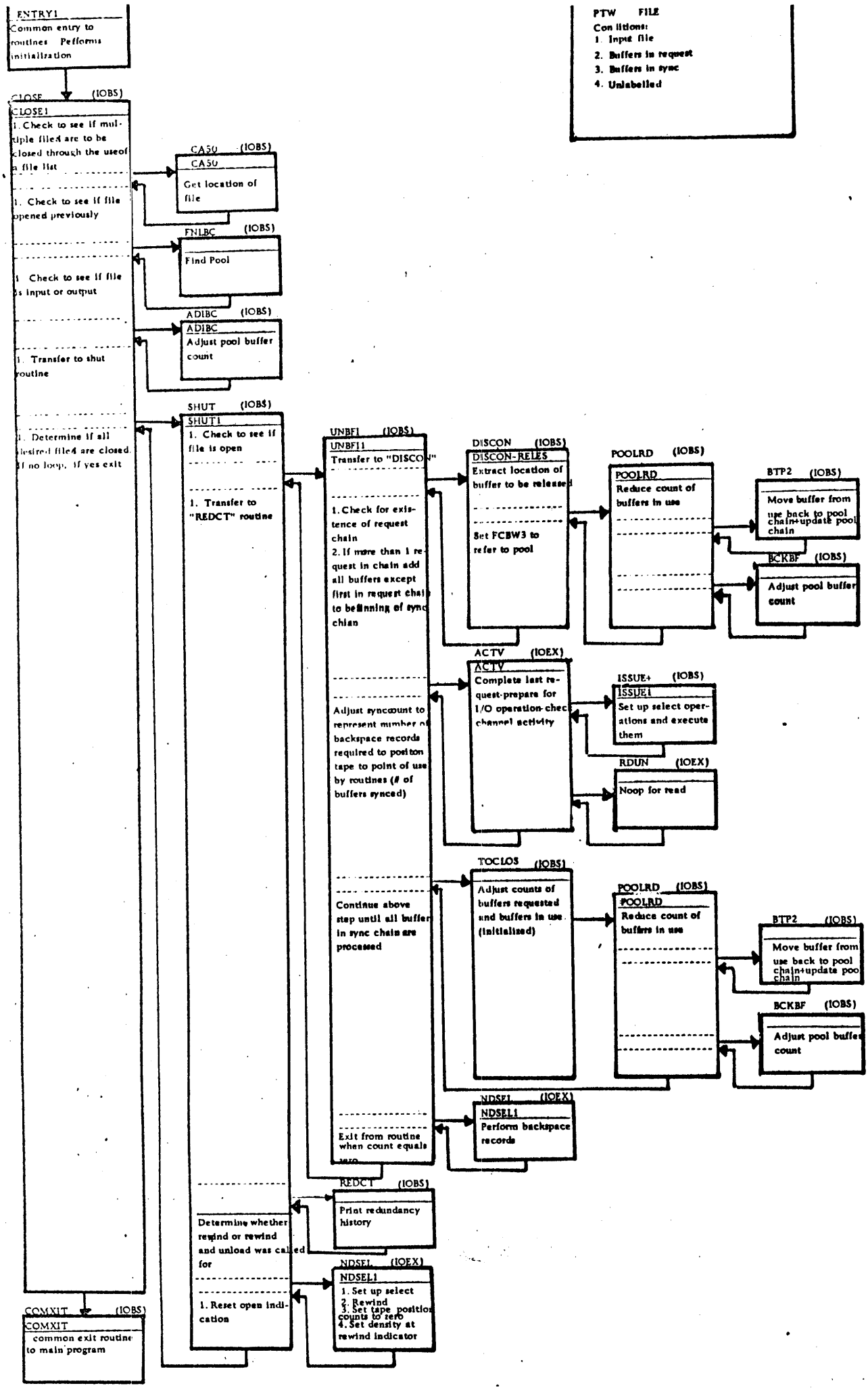


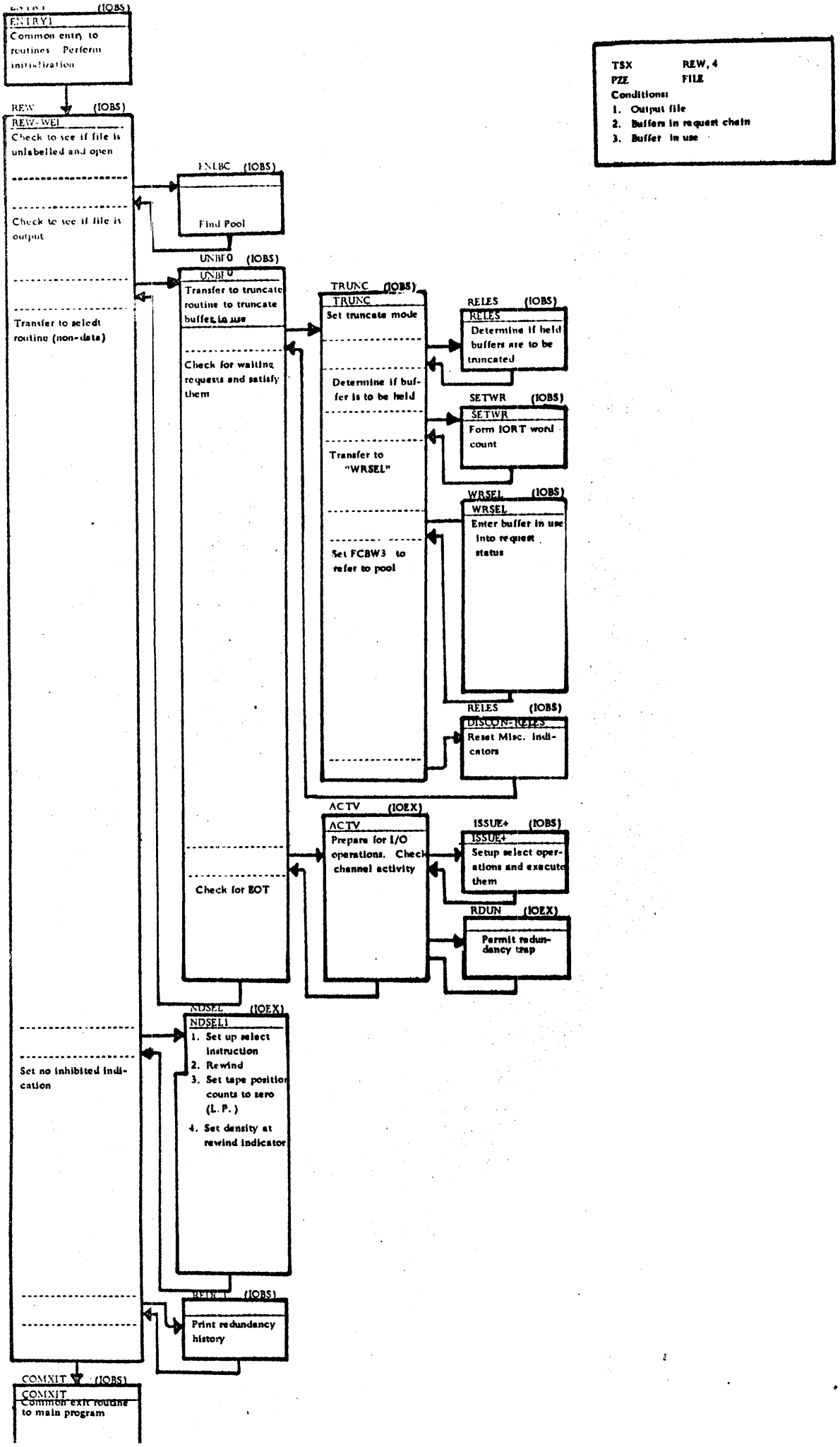

```
        TSX     ATTACH,4
        PZE     POOL
        PZE     LIST 1,,1
LIST1   PZE     FILE 1

Note:   FILE 1 is unlabeled, input and mounted
```

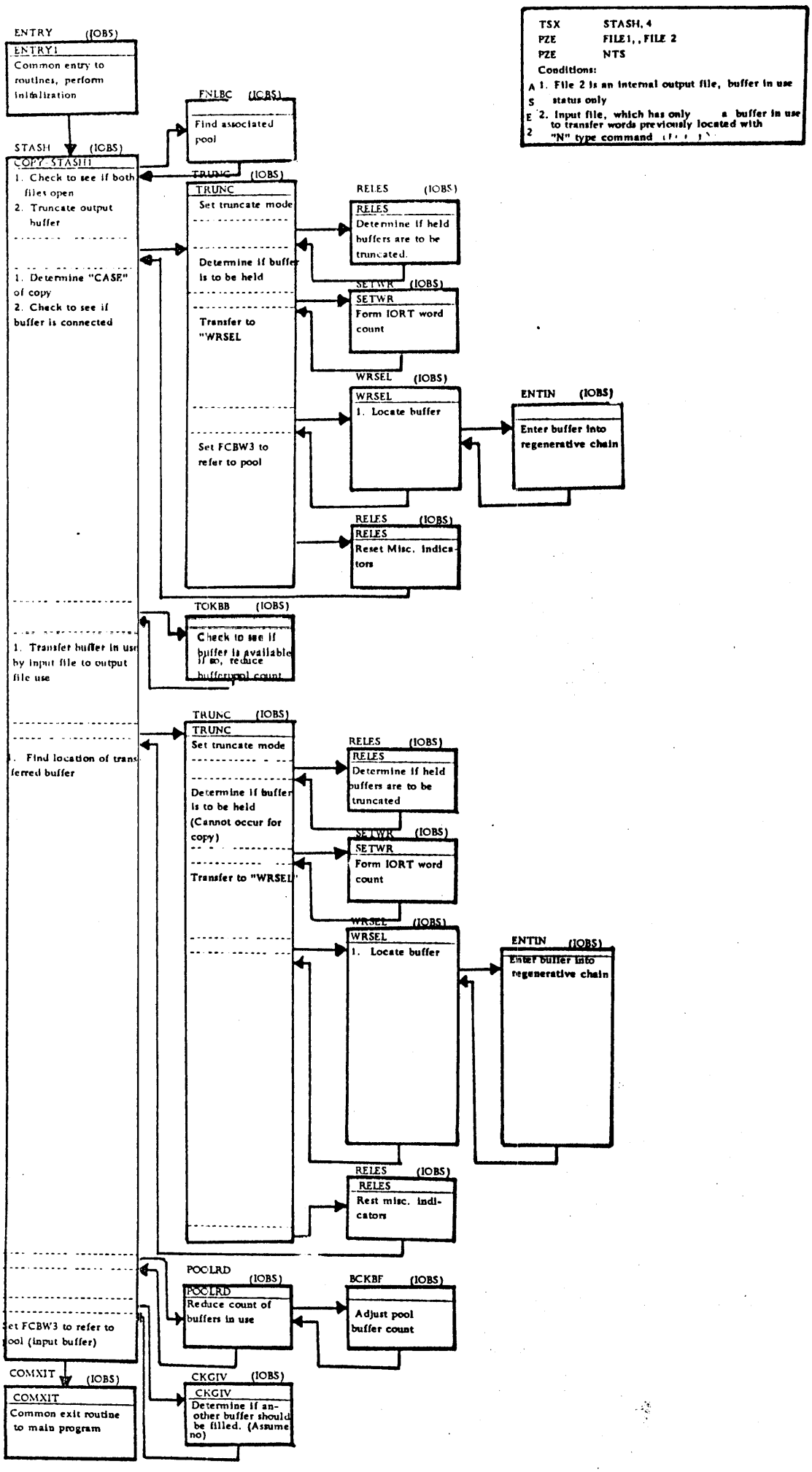

TSX BSF, 4
PZE FILE, , 2
Conditions:
1. Input file
2. Buffers in request chain
3. Buffers in sync chain
ENTRY (IOBS)
ENTRY1
Common entry to routines. Performs initialization
BSF (IOBS)
BSF1
Transfer to "BSCKS"
Check for file count of zero
1. Repeat above step until specified number of BSF's completed.
2. Prepare for dummy read across EOF mark
Wait for trap then set no inhibited indication restore file type, and set dummy sync chain to zero
COMXIT (IOBS)
COMXIT
Common exit routine to main program
BSCKS (IOBS)
BSCKS
Determine if file open--
Determine if input or output
Exit from routine
FNLBC (IOBS)
Find Pool
UNBFI (IOBS)
UNBFI1
Transfer to "DISCON"
1. Check for existence of request chain.
2. If more than 1 request in chain add all buffers except first in request chain to beginning of sync chain
Adjust sync count to represent number of backspace records required to position tape to point of use by routines (#of buffers synced)
Continue above step until all buffers in sync chain are processed.
Exit from routine when count equals zero
DISCON (IOBS)
DISCON-RELES
Extract location of buffer to be released
Set FCBW3 to refer to pool
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BTP2 (IOBS)
Move buffer from use back to pool chain and update pool chain
BCKBF (IOBS)
Adjust pool buffer count
ACTV (IOEX)
ACTV
Complete last request-prepare for I/O operation. check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Noop for read
TOCLOS (IOBS)
Adjust counts of buffers requested and buffers in use. (initialize)
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BTP2 (IOBS)
Move buffer from use back to pool chain and update pool chain
BCKBF (IOBS)
Adjust pool buffer count
NDSEL (IOEX)
NDSEL1
Perform backspace records
NDSEL (IOEX)
NDSEL1
Backspace file
RDSEL (IOBS)
RDSEL-WRSEL1
Place location of select routine in UCBW2 decrement and 0000's in address (dummy buffer)
ACTV (IOEX)
ACTV
Prepare for I/O operation, check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Noop for read

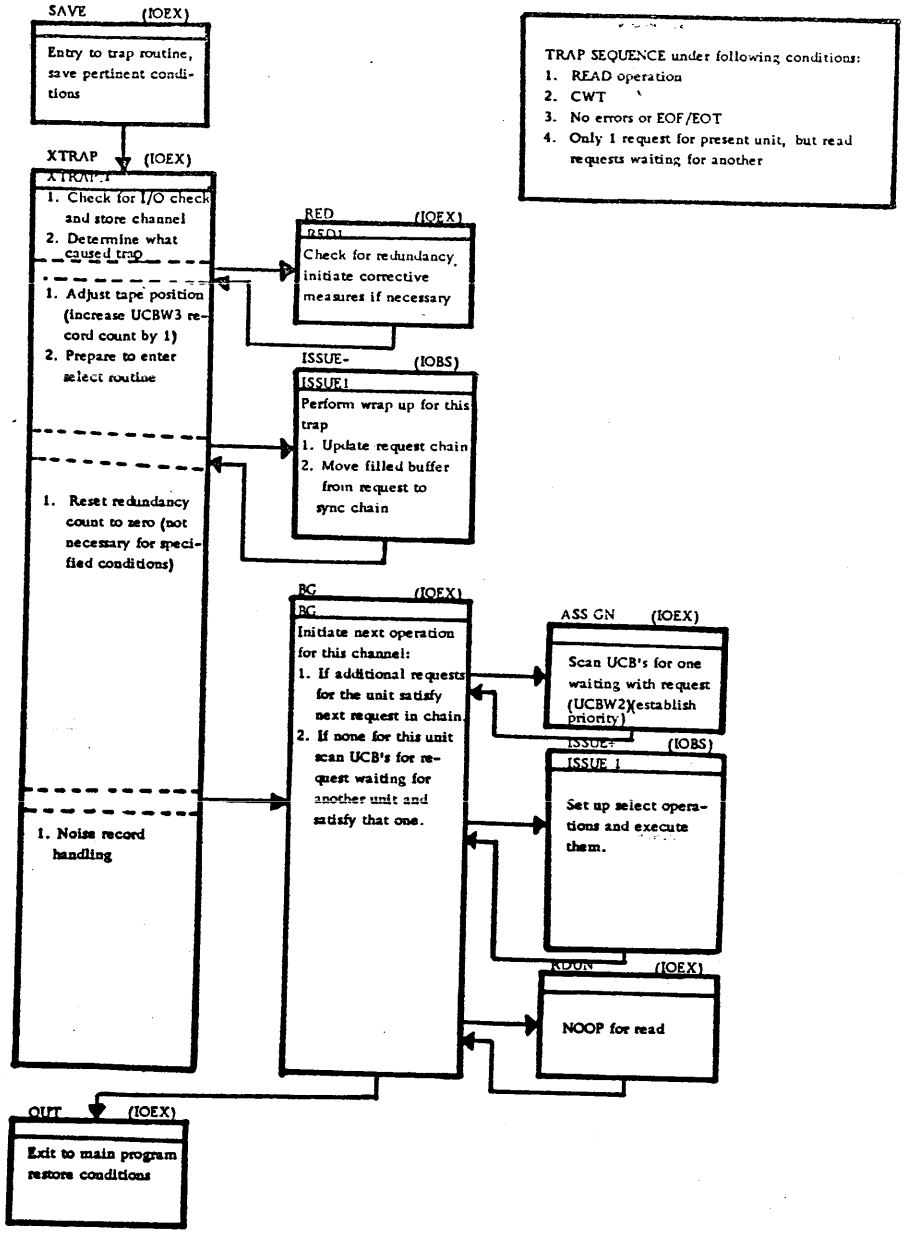

TSX COPY, 4
PZE FILE1, , FILE 2
Conditions:
1. Output file has buffer in use status only
2. Input file has only buffer in use to transfer, words previously located with "N" type command
ENTRY (IOBS)
ENTRY1
Common entry to routines, perform initialization
FNLBC (IOBS)
Find associated pool
COPY (IOBS)
COPY-STASH1
1. Check to see if both files open
2. Truncate output buffer.
1. Transfer buffer in use by input file to output file use.
2. Truncate transfeered buffer
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held
Transfer to "WRSEL"
Set FCBW3 to refer to pool
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
Enter buffer in use into request status
ACTV (IOEX)
ACTV
Prepare for I/O operations-check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Permit redundancy trap
RELES (IOBS)
RELES
Reset misc. indicators
1. Find location of transferred buffer
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held (cannot occur for copy)
Transfer to "WRSEL"
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
Enter buffer in use into request status.
ACTV (IOEX)
ACTV
Prepare for I/O operations check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Permit redundancy trap
RELES (IOBS)
RELES
Reset misc. indicators
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BCKBF (IOBS)
Adjust pool buffer count
et FCBW3 to refer to ool (input buffer)
COMXIT (IOBS)
COMXIT
Common exit routine to main program
CKGIV (IOBS)
CKGIV
Determine if anothe uffer should be filled (assume no.)

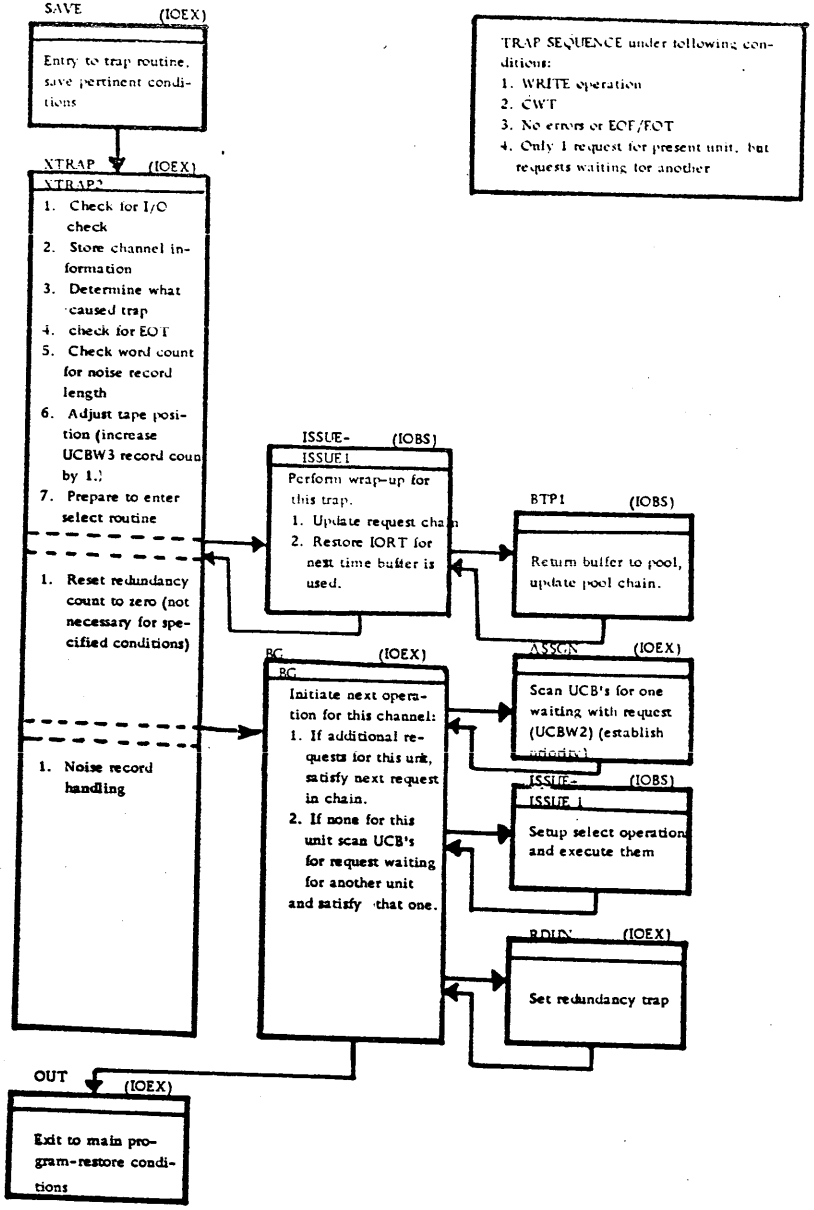

TSX READ, 4
PZE FILE, , EOB
PZE EOF, , ERR
IORT 4, , **
Conditions:
1. Initial read
2. More than 1 filled buffer in sync chain
3. No errors or EOF/EOT
4. EOB = 0
ENTRY (IOBS)
ENTRY1
Common entry to routines. Peforms initialization
READ (IOBS)
READ
1. Save location of EOF, ERR routines
2. C.K. Givup indicator
ENIBC (IOBS)
Find location of pool
RDWRT (IOBS)
RDWRT1
1. Extract pool location
2. Bring out IOCS command, save transmit location and word count
1. Check buffer for available words
1. Reduce locator decrement (WD.CT)
2. Adjust locator address (location of next word)
3. Set up transmit loop
4. Transmit words from buffer to location specified.
Determine what to do at completion of command (terminate)
SYNC (IOBS)
SYNC1
1. More buffer from sync chain to use
2. Adjust buffer count (FCBW2)
3. Update sync chain
4. Setup available word locator (BCW1)
RSIZD (IOBS)
Place number of available words in decrement of command
RELES (IOBS)
RELES
Find bufffer to be released
Set FCBW3 to refer to pool
Reset Misc. Ind
POOLRD (IOBS)
POOLRD
1. Reduce count of buffers in use
BTP2 (IOBS)
Move buffer from use back to pool chain +update pool
BCKBF (IOBS)
Adjust pool buffer count
CKGIV (IOBS)
CKGIV
Determine if another buffer should be filled (with these cond. only) GIVUP set
RDFKI (IOBS)
Determine whether or not enough buffers are in request staus to keep file ahead and act accordingly
COMXIT (IOBS)
COMXIT
Common exit routine to main program

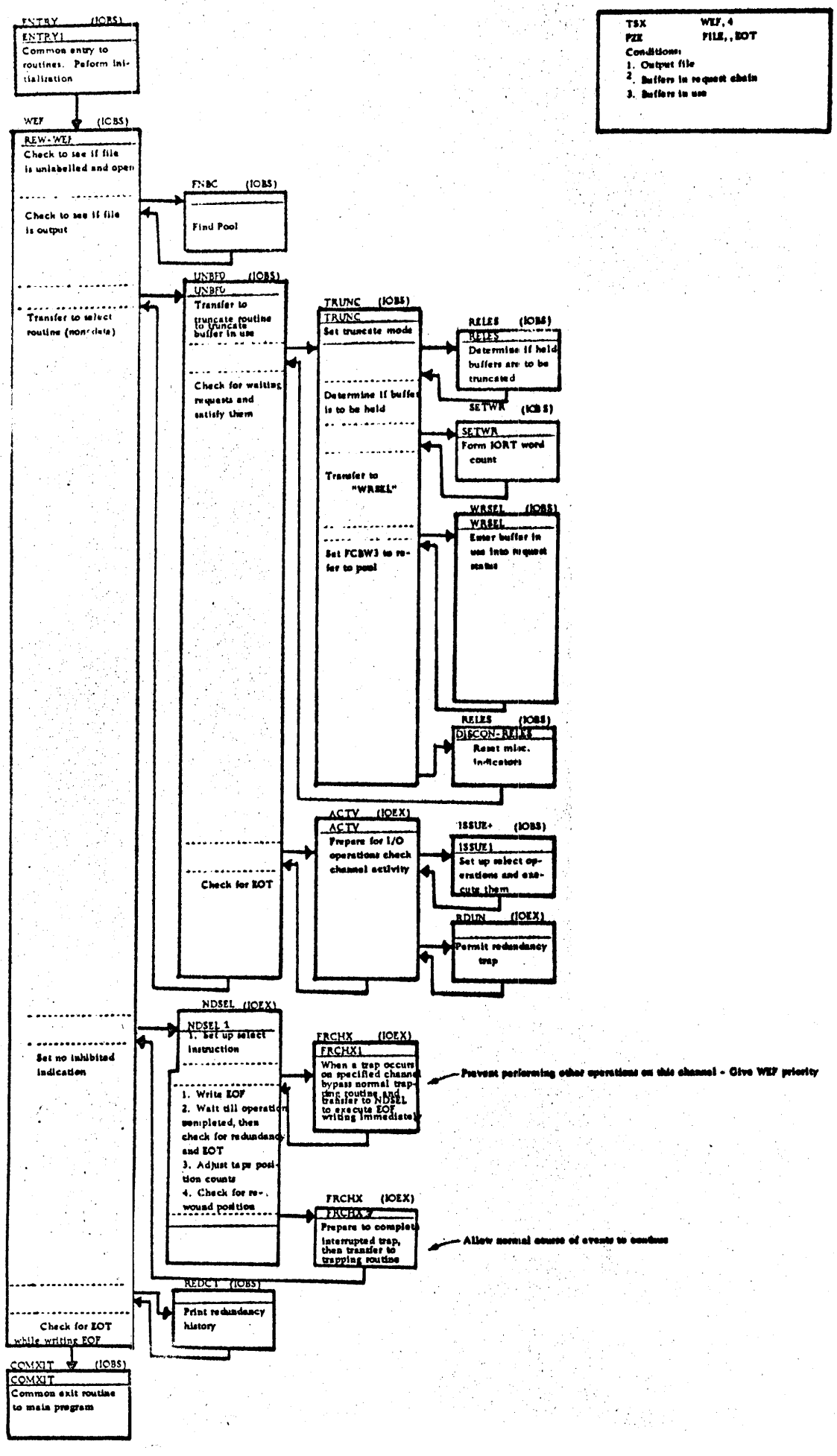

ENTRY1
Common entry to routines Performs initialization
CLOSE (IOBS)
CLOSE1
1. Check to see if multiple files are to be closed through the use of a file list
1. Check to see if file opened previously
1. Check to see if file is input or output
1. Transfer to shut routine
1. Determine if all desired files are closed. If no loop, if yes exit
CA50 (IOBS)
CA50
Get location of file
FNLBC (IOBS)
Find Pool
ADIBC (IOBS)
ADIBC
Adjust pool buffer count
SHUT (IOBS)
SHUT1
1. Check to see if file is open
1. Transfer to "REDCT" routine
UNBF1 (IOBS)
UNBF11
Transfer to "DISCON"
1. Check for existence of request chain
2. If more than 1 request in chain add all buffers except first in request chain to beginning of sync chian
Adjust syncount to represent number of backspace records required to position tape to point of use by routines (# of buffers synced)
Continue above step until all buffers in sync chain are processed
Exit from routine when count equals zero
DISCON (IOBS)
DISCON-RELES
Extract location of buffer to be released
Set FCBW3 to refer to pool
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BTP2 (IOBS)
Move buffer from use back to pool chain+update pool chain
BCKBF (IOBS)
Adjust pool buffer count
ACTV (IOEX)
ACTV
Complete last request-prepare for I/O operation-check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Noop for read
TOCLOS (IOBS)
Adjust counts of buffers requested and buffers in use (initialized)
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BTP2 (IOBS)
Move buffer from use back to pool chain+update pool chain
BCKBF (IOBS)
Adjust pool buffer count
NDSEL (IOEX)
NDSEL1
Perform backspace records
REDCT (IOBS)
Print redundancy history
Determine whether rewind or rewind and unload was called for
1. Reset open indication
NDSEL (IOEX)
NDSEL1
1. Set up select
2. Rewind
3. Set tape position counts to zero
4. Set density at rewind indicator
COMXIT (IOBS)
COMXIT
common exit routine to main program
PTW FILE
Conditions:
1. Input file
2. Buffers in request
3. Buffers in sync
4. Unlabelled

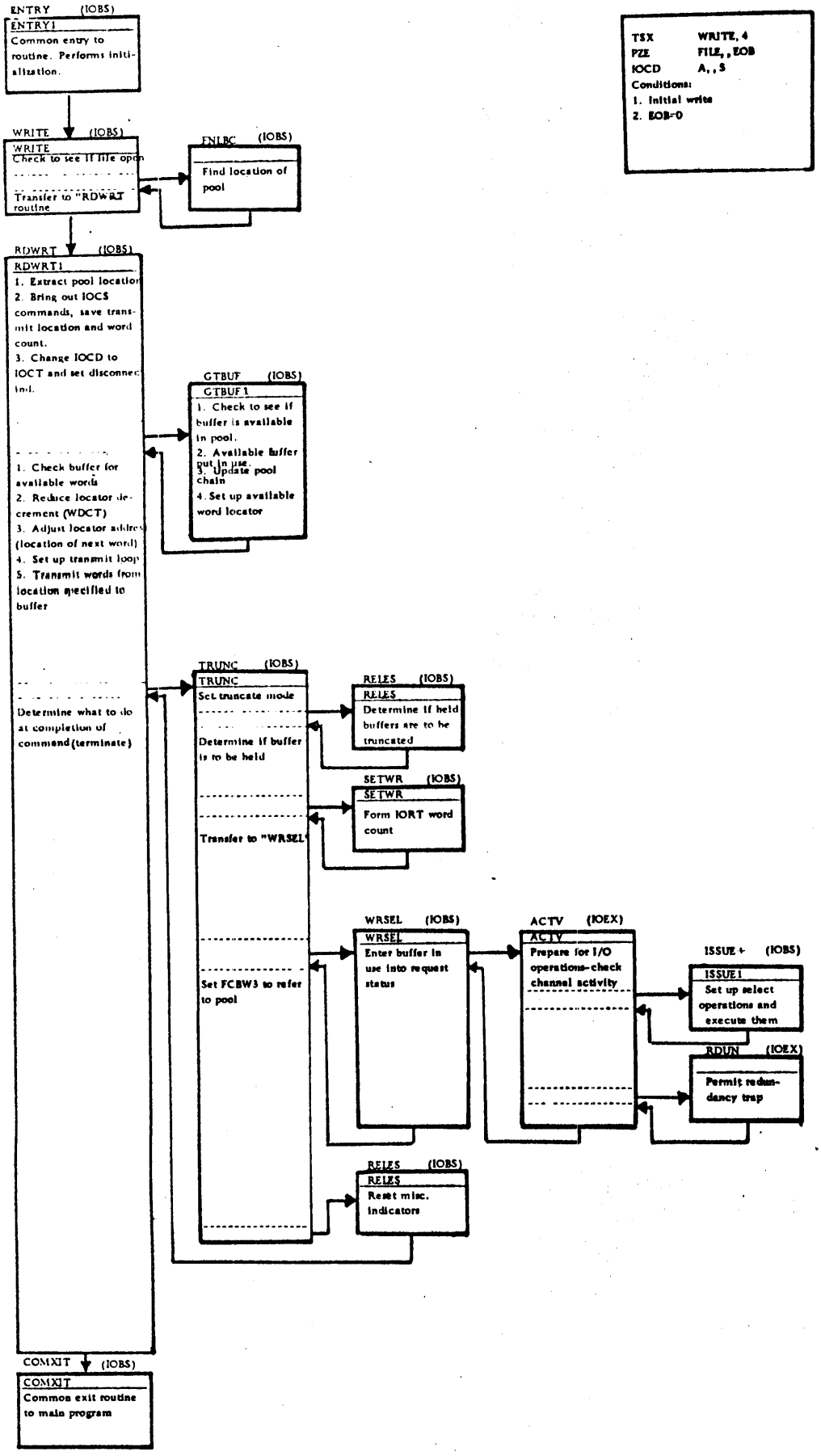

ENTRY1 (IOBS)
Common entry to routines Perform initialization
REW (IOBS)
REW-WEI
Check to see if file is unlabelled and open
Check to see if file is output
Transfer to seledt routine (non-data)
FNLBC (IOBS)
Find Pool
UNBF0 (IOBS)
UNBF0
Transfer to truncate routine to truncate buffer in use
Check for waiting requests and satisfy them
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held
Transfer to "WRSEL"
Set FCBW3 to refer to pool
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
Enter buffer in use into request status
RELES (IOBS)
DISCON-RELES
Reset Misc. Indicators
ACTV (IOEX)
ACTV
Prepare for I/O operations. Check channel activity
ISSUE+ (IOBS)
ISSUE+
Setup select operations and execute them
RDUN (IOEX)
Permit redundancy trap
Check for EOT
NDSEL (IOEX)
NDSEL1
1. Set up select instruction
2. Rewind
3. Set tape position counts to zero (L.P.)
4. Set density at rewind indicator
Set no inhibited indication
REDP1 (IOBS)
Print redundancy history
COMXIT (IOBS)
COMXIT
Common exit routine to main program
TSX REW, 4
PZE FILE
Conditions:
1. Output file
2. Buffers in request chain
3. Buffer in use


TSX STASH, 4
PZE FILE1, , FILE 2
PZE NTS
Conditions:
A 1. File 2 is an internal output file, buffer in use
S status only
E 2. Input file, which has only a buffer in use
2 to transfer words previously located with
"N" type command
ENTRY (IOBS)
ENTRY1
Common entry to routines, perform initialization
ENLBC (IOBS)
Find associated pool
STASH (IOBS)
COPY STASH1
1. Check to see if both files open
2. Truncate output buffer
1. Determine "CASE" of copy
2. Check to see if buffer is connected
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held
Transfer to "WRSEL
Set FCBW3 to refer to pool
RELES (IOBS)
RELES
Determine if held buffers are to be truncated.
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
1. Locate buffer
ENTIN (IOBS)
Enter buffer into regenerative chain
RELES (IOBS)
RELES
Reset Misc. Indicators
TOKBB (IOBS)
Check to see if buffer is available if so, reduce buffer pool count
1. Transfer buffer in use by input file to output file use
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held (Cannot occur for copy)
Transfer to "WRSEL"
1. Find location of transferred buffer
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
1. Locate buffer
ENTIN (IOBS)
Enter buffer into regenerative chain
RELES (IOBS)
RELES
Rest misc. indicators
POOLRD (IOBS)
POOLRD
Reduce count of buffers in use
BCKBF (IOBS)
Adjust pool buffer count
Set FCBW3 to refer to pool (input buffer)
COMXIT (IOBS)
COMXIT
Common exit routine to main program
CKGIV (IOBS)
CKGIV
Determine if another buffer should be filled. (Assume no)


SAVE (IOEX)
Entry to trap routine, save pertinent conditions
XTRAP (IOEX)
XTRAP1
1. Check for I/O check and store channel
2. Determine what caused trap
1. Adjust tape position (increase UCBW3 record count by 1)
2. Prepare to enter select routine
1. Reset redundancy count to zero (not necessary for specified conditions)
1. Noise record handling
RED (IOEX)
RED1
Check for redundancy, initiate corrective measures if necessary
ISSUE- (IOBS)
ISSUE1
Perform wrap up for this trap
1. Update request chain
2. Move filled buffer from request to sync chain
BG (IOEX)
BG
Initiate next operation for this channel:
1. If additional requests for the unit satisfy next request in chain.
2. If none for this unit scan UCB's for request waiting for another unit and satisfy that one.
ASS GN (IOEX)
Scan UCB's for one waiting with request (UCBW2)(establish priority)
ISSUE- (IOBS)
ISSUE 1
Set up select operations and execute them.
RDUN (IOEX)
NOOP for read
OUT (IOEX)
Exit to main program restore conditions
TRAP SEQUENCE under following conditions:
1. READ operation
2. CWT
3. No errors or EOF/EOT
4. Only 1 request for present unit, but read requests waiting for another


SAVE (IOEX)
Entry to trap routine, save pertinent conditions
XTRAP (IOEX)
XTRAP2
1. Check for I/O check
2. Store channel information
3. Determine what caused trap
4. check for EOT
5. Check word count for noise record length
6. Adjust tape position (increase UCBW3 record coun by 1.)
7. Prepare to enter select routine
1. Reset redundancy count to zero (not necessary for specified conditions)
1. Noise record handling
ISSUE- (IOBS)
ISSUE 1
Perform wrap-up for this trap.
1. Update request chain
2. Restore IORT for next time buffer is used.
BTP1 (IOBS)
Return buffer to pool, update pool chain.
BG (IOEX)
BG
Initiate next operation for this channel:
1. If additional requests for this unit, satisfy next request in chain.
2. If none for this unit scan UCB's for request waiting for another unit and satisfy that one.
ASSGN (IOEX)
Scan UCB's for one waiting with request (UCBW2) (establish priority)
ISSUE- (IOBS)
ISSUE 1
Setup select operation and execute them
RDUN (IOEX)
Set redundancy trap
OUT (IOEX)
Exit to main program-restore conditions
TRAP SEQUENCE under following conditions:
1. WRITE operation
2. CWT
3. No errors or EOF/EOT
4. Only 1 request for present unit, but requests waiting for another


TSX WEF, 4
PZE FILE, , EOT
Conditions:
1. Output file
2. Buffers in request chain
3. Buffers in use
ENTRY (IOBS)
ENTRY1
Common entry to routines. Peform initialization
WEF (IOBS)
REW-WEF
Check to see if file is unlabelled and open
Check to see if file is output
Transfer to select routine (non-data)
FNBC (IOBS)
Find Pool
UNBFO (IOBS)
UNBFO
Transfer to truncate routine to truncate buffer in use
Check for waiting requests and satisfy them
Check for EOT
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held
Transfer to "WRREL"
Set FCBW3 to refer to pool
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRREL (IOBS)
WRREL
Enter buffer in use into request status
RELES (IOBS)
DISCON-RELES
Reset misc. indicators
ACTV (IOEX)
ACTV
Prepare for I/O operations check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Permit redundancy trap
NDSEL (IOEX)
NDSEL 1
1. Set up select instruction
Set no inhibited indication
1. Write EOF
2. Wait till operation completed, then check for redundancy and EOT
3. Adjust tape position counts
4. Check for rewound position
FRCHX (IOEX)
FRCHX1
When a trap occurs on specified channel bypass normal trapping routine and transfer to NDSEL to execute EOF writing immediately
Prevent performing other operations on this channel - Give WEF priority
FRCHX (IOEX)
FRCHX2
Prepare to complete interrupted trap, then transfer to trapping routine
Allow normal course of events to continue
REDCT (IOBS)
Print redundancy history
Check for EOT while writing EOF
COMXIT (IOBS)
COMXIT
Common exit routine to main program


ENTRY (IOBS)
ENTRY1
Common entry to routine. Performs initialization.
WRITE (IOBS)
WRITE
Check to see if file open
Transfer to "RDWRT routine
ENLBC (IOBS)
Find location of pool
RDWRT (IOBS)
RDWRT1
1. Extract pool location
2. Bring out IOCS commands, save transmit location and word count.
3. Change IOCD to IOCT and set disconnect ind.
1. Check buffer for available words
2. Reduce locator decrement (WDCT)
3. Adjust locator address (location of next word)
4. Set up transmit loop
5. Transmit words from location specified to buffer
Determine what to do at completion of command (terminate)
GTBUF (IOBS)
GTBUF1
1. Check to see if buffer is available in pool.
2. Available buffer put in use.
3. Update pool chain
4. Set up available word locator
TRUNC (IOBS)
TRUNC
Set truncate mode
Determine if buffer is to be held
Transfer to "WRSEL"
Set FCBW3 to refer to pool
RELES (IOBS)
RELES
Determine if held buffers are to be truncated
SETWR (IOBS)
SETWR
Form IORT word count
WRSEL (IOBS)
WRSEL
Enter buffer in use into request status
ACTV (IOEX)
ACTV
Prepare for I/O operations-check channel activity
ISSUE+ (IOBS)
ISSUE1
Set up select operations and execute them
RDUN (IOEX)
Permit redundancy trap
RELES (IOBS)
RELES
Reset misc. indicators
COMXIT (IOBS)
COMXIT
Common exit routine to main program
TSX WRITE, 4
PZE FILE, , EOB
IOCD A, , S
Conditions:
1. Initial write
2. EOB=0