IBM

System/360

OS BTAM Coding

Education Guide

Education Development - Poughkeepsie, New York

Copies of this publication can be obtained through IBM Branch Offices.
Address comments concerning the contents of this publication to:
IBM DPD Education Development, Education Center, Poughkeepsie, New York 12602

TABLE OF CONTENTS

Introduction

Section 1

INTRODUCTION

This outline has been written in a modular form to facilitate updating and addition of new material.

It consists of a basic module covering device independent topics. Following this basic module will be device dependent modules. These will be added as they are developed for the various categories of devices, e.g., 1050, 2260, binary synchronous. They should in no way affect the basic module.

The basic module covers:

Basic operation of BTAM
BTAM buffering
BTAM macros
General channel programming considerations

Essentially the device dependent modules cover:

Hardware
Line control
Channel programs
Programming example
Special considerations related to the device
Class problem

It is recommended the instructor uses the basic module and foils as the basis for his instructions. However, he should incorporate into his presentation examples and materials from the various device modules. The interests of the students should dictate what material is used from the various modules.

The best BTAM class will be one where the instructor has a good knowledge of BTAM and can run the class as a workshop where there is adequate machine time to do the selected class problems.

Course Description

Section 2

Course Description

S/360 OS BTAM CODING

Course Code K3663 62

Duration 3 to 5 customer-student days *

Audience This course is intended for Systems Programmers - customer and IBM.

Prerequisites S/360 OS Coding (S3660 65)

Data Communications Concepts (U1900 62)

Objectives Upon successful completion of the course the student is able to:

1. Understand the functions of BTAM and where it fits in the Operating System.
2. To code using BTAM macros in order to service the communication lines connected to the S/360.

* Depends on amount of machine time.

Copies of this publication can be obtained through IBM Branch Offices.
Address comments concerning the contents of this publication to:
IBM DPD Education Development, Education Center, Poughkeepsie, New York

Material Requirements

Student Materials

Title	Form No.	Abstract Ref.
S/360 OS BTAM Version II	C30-2004	*
2701 Data Adapter Unit - Principles of Operation	A22-6864	*

Instructor Materials

S/360 OS BTAM - PLM	Y30-2001	*
S/360 OS BTAM Coding Education Guide		See below

* System/360 Bibliography (A22-6822, N20-0360)

Abstract

R20-4105 Education Guide

8 1/2" x 11" Looseleaf Instructor Outline
172 pages (Brown Cover)

This guide contains a detailed course outline, with teaching notes to be used by a qualified instructor. Included in this guide are: references to supporting information, paper masters of overhead foils, sample problems and class problems.

General Course Outline

Section 3

GENERAL COURSE OUTLINE

OS BTAM DEVICE INDEPENDENT TOPICS

A. Basic Operation of BTAM

1. Relationship between user's program and supervisor
2. Relationship between user's program, BTAM, and supervisor
    a. At execution of OPEN
    b. Additional OPEN functions
    c. Execution of program

B. BTAM Buffering

1. Why buffering
2. Why BTAM buffering
3. Format of buffer pool
4. Ways of building buffer pool
    a. GETMAIN and BUILD
    b. Area defined at assembly and BUILD
    c. GETPOOL
    d. At BTAM OPEN
5. Ways of obtaining buffer from pool
    a. For READ
    b. For WRITE
6. Dynamic Buffering
    a. Why dynamic buffering
    b. Operation

C. BTAM Macro Considerations

1. Data Control Block (DCB) macro
2. Line Error Block (LERB) macro

3. Assemble Translate Table (ASMTRTAB) macro

4. Define Terminal List (DFTRMLST) macro

5. Change Terminal Entry (CHGNTRY) macro

6. OPEN and CLOSE macro

7. LOPEN macro

8. READ and WRITE macros

9. Reset Polling or Reset Line (RESETPL) macro

10. Translate (TRNSLATE) macro

11. Line Error Print (LERPRT) macro

12. Request Buffer (REQBUF) macro

13. Release Buffer (RELBUF) macro

14. TWAIT macro

D. Error Recovery Procedures

1. Functions

a. Basic capabilities

b. Other capabilities that can be specified

2. Operation

a. Dynamic buffering

b. Regular buffering

E. Channel Programs

1. Purpose

2. Users concern with channel programs

3. Basic types of channel programs

a. Read Initial and Read Initial with Reset

b. Read Continue

c. Read Repeat

d. Other methods to terminate transmission

e. Write Initial and Write Initial with Reset

f. Write Continue

g. Examples of sequences

F. General OS BTAM Considerations

1. EOB in last position of buffer

2. Reserved areas in DECB

3. WR TN after WR TI for switched lines

4. Same features within line groups

5. All skip bits on for line

6. No circle D on Read TP

7. Translate macro limit -256 bytes

8. Read or Write Conversational use dial list

9. S parameter for terminal list entry on Read TI

10. Don't use S parameter for terminal list entry on switched line.

11. Falling through TWAIT on previously completed I/O

12. Read Initial with reset always resets

13. Location of IOB

IBM 2260 Module For OS BTAM

A. Introduction

1. Remote environment

2. Contents of module

3. Bibliography

B. 2260 Hardware For Remote Operation

1. 2848 Display Control

a. Function

b. 3 basic models

c. Optional features

d. Operator controls and indicators for 2848

2. 2260 Display Station

a. Two models

b. Standard features

c. Optional features

d. Operator controls

e. Control symbols

f. Control keys

g. Example of keyboard operations

C. ASCII Line Control

1. Requirements of line control

2. Start-Stop line control

a. Six control characters

b. Two additional characters for control

c. Modes of operation

d. Control status

3. Addressing sequence

4. Similarity with 1050 line control

D. Data Flow, Associated Hardware, and Code Sets

1. Between 2848 and 2701

2. Between S/360 Channel and 2701

3. Remote 2848

a. Data set adapter interface with data set

b. 2848 Data Set adapter

c. Common Control

E. Commands, Sequence and Responses, and Channel Programs

1. Sequences and Responses

2. Eight commands can be executed

   a. Specific Poll to 2260

   b. Specific Poll to Printer (1053)

   c. General Poll

   d. Read Full DS Buffer

   e. Write Addressed DS or 1053

   f. Erase/Write and Line Address Write

3. Effect of Dynamic Buffering

   a. PCI occurs

   b. PCI appendage receiver control

   c. Chain of 4 CCW's perform read and write

   d. This chain replaces read's and write's described in channed programs.

F. Programming Considerations and Sample Program

1. Purpose of program

2. Programming considerations

   a. JCL

   b. Special considerations

   c. DSECT

   d. Buffer Pool

   e. Addressing and Polling Lists

   f. Read's and Write's

   g. Sequence of I/O Operations

   h. I/O return codes and completion codes

i. Polling considerations

j. Translate

k. Terminal table

l. ERROPT in DCB

G. CLASS PROBLEM

Detailed Course Outline

Section 4

OS BTAM Device Independent Considerations

Outline

Objectives: This topic has been written independent of a device and describes:

1. How BTAM fits into the OS system.
2. BTAM buffering.
3. The basic operation of the various BTAM macros.
4. Error Recovery.
5. Channel programs.
6. General OS BTAM considerations.

The instructor will draw on the device dependent topics to describe in detail how BTAM operates for a particular device.

* * * * *

Suggested Sequence:

A. Basic Operation of BTAM

NOTE: Foils 1-V-1,2,3,4, and 5 show how BTAM fits into the system and what functions it performs. Foil 1-V-16 can be used if desired to further discuss BTAM's operation; however, these functions could have been discussed in the sequence of foils V-2,3, and 5.

Foils 1-V-17,18,19,20,21, and 22 are layouts of the control blocks and are included if the instructor should get a detailed question concerning their contents.

1. Relationship between User's Program and Supervisor 1-V-1

a. Supervisor

1) UCB - Unit Control Block

a) Generated at SYSGEN time when rest of system is generated.

b) Will have UCB for each communication line.

c) Describes the nature of the devices on the line, e.g.,

Physical device address
Type of TCU
Type of adapter
Terminal, e.g., 2260
Optional features, e.g., auto call

2) IOS - Input Output Supervisor

a) Function - two prime functions
To service all I/O requests, i.e., start and supervise I/O operations for programs requesting use of an I/O device.

To process I/O interruptions resulting from the execution of the channel programs.

b) Two major program sections within IOS:
EXCP supervisor - handles I/O requests
Input/Output supervisor

b. User's program

1) User has I/O activities he wishes to perform

2) How does user interface with IOS

a) EXCP
Programmer can use Execute Channel Program (EXCP) macro that interfaces directly with IOS.

Programmer would have to construct channel programs and program for all other functions which would be performed by an access method.

b) Access method - in this case BTAM
The user in his program at assembly time will generate control blocks to be used for passing parameters to BTAM.

The DECB will be used to pass parameters for the reads and writes.

At execution time an OPEN macro is executed.

2. Relationship between User's Program, BTAM, and Supervisor
1-V-2

a. At execution of OPEN

1) Initializes the Data Control Block (DCB) for a line group.

a) DCB serves three functions

Identifies access method
Specifies features of access method, e.g., buffering
Provides pointers to other control routines and read/write routines

b) All lines within a line group must observe following requirements

All lines must be switched or nonswitched.
All terminals must be the same type.
All lines must share the same buffer pool if one is defined.
All lines must have same hardware features.

c) How do we determine which lines are to be associated with which DCB's?

The DCB has a DDNAME parameter which references a DD card or cards in the JCL. These DD cards will then specify the line or lines in the line group. 1-V-3

NOTE: 1-V-3 shows two examples of DD statements. In the first GRP1 would be referenced by one DCB while GRP2 would be referenced by a second DCB.

The references to GROUPONE and GROUPTWO are names defined at SYSGEN which refer to groups of lines.

The second example shows DD cards which indicate by actual address which lines will be associated with the DCB.

2) Build Data Extent Block (DEB) 1-V-4

a) One per line group

b) Main function is to indicate extent of data set associated with related DCB.

Will point to each UCB associated with line group. Determined which to point to on the basis of DD cards in JCL which were pointed to by DDNAME parameter in DCB.

3) Builds Input/Output Block (IOB)

a) One per line

b) Used by BTAM to pass parameters to IOS. Therefore, it performs the function for BTAM that the DECB performs for the user, i.e., passing of I/O parameters.

4) Loads

a) BTAM Read/Write Routines

b) Buffering routines

5) Builds buffers if requested

b. Additional OPEN functions 1-V-5

1) Load I/O modules

a) An I/O module loaded for each type of device.

NOTE: 1-V-17 is a foil of an I/O module layout. It can be used at the instructors option.

b) Contains

Channel Command Words (CCW's) for channel programs
Transmission control characters

2) Load Appendages

a) Appendages are routines provided by the access method at OPEN time that allow additional control over I/O operations during channel program execution.

b) IOS will utilize these appendages as if they were extensions of IOS.

c) Two appendages available with BTAM
Program Controlled Interrupt (PCI) appendage, utilized if dynamic buffering specified.
Channel End/Abnormal End appendage, entered when Channel End interrupt occurs or when abnormal end, e.g., Unit Check, occurs.

c. Execution of program

NOTE: Instructor can continue to use foil 1-V-5 or use 1-V-16.

1) Read execution time

a) READ in user's program causes control to be passed to BTAM Read/Write routine (this address in DCB)

b) BTAM Read/Write routine builds channel program dynamically and places it in IOB (Uses I/O module to build program)

c) Control passed to IOS at this point

d) IOS issues START I/O

e) Control passed back to user's program. User at this point must check return code to be sure operation started properly.

f) If user cannot do processing before read completes, must go into wait for I/O completion.

g) Channel program execution is continuing concurrently with user's program during this time.

h) On Channel End (CE) or abnormal end IOS regains control and enters appendage.

i) Action appendage takes depends on status bytes set.

If I/O complete, post completion in an event control block (first word of DECB) and return control to user.

If polling and receive negative response, reinitializes channel program and restarts it.

If error and Error Recovery Procedures specified, IOS will pass control to these routines for diagnosis and recovery.

NOTE: If dynamic buffering employed, IOS may have to service a PCI interrupt. In this case control is passed to the PCI appendage which handles the requirements imposed by dynamic buffering.

j) If I/O complete when user's program regains control must check completion to see if it was good transmission.

This completion code is found in the Event Control Block (ECB) which is the first word of the DECB.

At completion BTAM will place status and sense bytes in DECB.

2) Translate

a) Message received may have to be translated from terminal transmission code to EBCDIC.

b) Messages being sent have to be translated from EBCDIC to terminal code.

c) User will employ translate macro to perform this operation.

d) Assemble translate table macro is employed to generate the required translate tables.

3) Line Error Block (LERB)

a) The user can employ a facility of BTAM to keep a record of transmission errors, namely,
Interventions Required
Time-outs
Number of transmissions

b) User specifies where to keep these counts with LERB macro.

c) User can print these counts out at any time employing the LERPRT macro.

4) Define terminal list

The BTAM channel programs require information from the users program concerning:

Addressing characters, i.e., what stations and their components are to be addressed on a line.
Polling characters, i.e., what stations and their components are to be polled on a line.
If auto call is employed what is the telephone number of the station being called.

This is all information required by BTAM to control the line and only the user can provide BTAM with this data. This is the function of the DFTRMLST macro.

BTAM will actually transmit characters from the users terminal list when polling and addressing stations.

5) Release buffer

a) When using the buffering facility of BTAM after a read it is necessary to return to the pool whatever buffers were required.

b) The user employs the RELBUF macro to do this.

6) Close

User uses CLOSE macro to conclude BTAM operation when the job is completed.

NOTE: All BTAM macros have not been convered at this point. The purpose of the preceding was to provide a feeling for the operation as a whole and some of the macros most frequently used.

B. BTAM Buffering

1. Why buffering

   a. To achieve concurrent operations

      1) Many I/O devices (lines) transferring data to and from CPU simultaneously.

      2) All this I/O occurring concurrently with CPU processing.

   b. Therefore buffering permits optimum usage of the computer system.

2. Why BTAM buffering

   a. Utilizing BTAM's buffering facility a pool of buffers can be formed to service a number of lines.

   b. As a buffer is required for a line, it is drawn from the pool. As soon as there is no further requirement for it, it should be returned to the pool.

   c. With this type of operation less space is required for buffers than would be required if each line was to have its seperate buffers.

   d. BTAM relieves the user of having to perform a similar operation himself.

NOTE: It is possible to utilize BTAM without calling on the buffering facilities. In this case all I/O operations will specify an address of a user's defined area and a count of characters to be transferred to or from this area.

3. Format of buffer pool 1-V-6

   a. Any buffer pool utilized by BTAM buffer management routines must have the format indicated in 1-V-6.

   b. Buffer control block

      1) Address of available buffer contains the address of the first buffer in the pool which is available for use in I/O operations.

      2) Number of available buffers
         Indicates how many buffers in the pool are available.

      3) Length of buffers
         Indicates the length of the buffers. All buffers will be same length.

c. Buffers
Each buffer will have an address pointing to the next available buffer. The last available buffer will have 0's in this field.

4. Ways of building a buffer pool

a. GETMAIN and BUILD 1-V-7

1) User uses OS GETMAIN and BUILD macros prior to OPEN to build pool.

NOTE: Could use DCB exit from OPEN to build pool in this fashion

2) GETMAIN obtains space for pool.
Address of area obtained returned in register 1.

3) This address is stored by user in the DCB field reserved for pointer to buffer control block. Could store this address in more than one DCB if one wishes them to share same pool between line groups.

NOTE: By sharing a pool between more than one DCB should be able to utilize buffers more efficiently. Should require fewer buffers than would be required if a buffer pool for each DCB.

4) The address returned from GETMAIN given to BUILD macro.
BUILD will then build pool consisting of the number of buffers and length specified.

NOTE: The BUILD need not be given before the OPEN. However, GETMAIN and storing of address in DCB must be done before OPEN or on OPEN exit.

5) Storing BUFL in DCB
It is the programmer's responsibility to assure buffer length is stored in DCB.
Include as parameter in DCB macro.

NOTE: When obtaining the buffer pool in this fashion buffer number will never appear in the DCB, the reason being that BTAM has no need for it to be there.

b. Area defined at assembly and BUILD 1-V-8

1) This is basically the same as the preceding except instead of a GETMAIN the user defines the area at assembly time with DS or DC statement.

2) User should put address of this area in DCB at assembly time along with the buffer length.

c. GETPOOL 1-V-9

1) Combination of GETMAIN and BUILD

2) Buffer control block address and buffer number inserted in DCB by GETPOOL

3) GETPOOL should be issued before OPEN

4) Buffer length parameter should be included in DCB macro for assembly.

NOTE: Even though GETPOOL places buffer number in DCB, it is not required by BTAM.

5) GETPOOL will place buffer control block address in a particular DCB. Therefore, if the user wished to share the pool between more than one DCB, it would be up to him to get the address out of the one DCB and place it in the other DCB's.

d. At BTAM OPEN 1-V-10

1) Simpliest method to employ

2) At assembly time buffer length and buffer number are included as parameters.

3) At OPEN time
Buffer pool is constructed.
Buffer control block address will be placed into the DCB at this time.

4) Buffer pool associated with only that DCB. However, after OPEN user could obtain address of buffer control block from that DCB and place it in other DCB's.

If the user were to do this he would want to be sure that the other DCB's did not have buffer number parameter at assembly time. Also these DCB's could not be OPENed at the same time as the one with BUFNO.

NOTE: There are three conditions BTAM checks before building a buffer pool.

1. No buffer control block address is in the DCB at OPEN time.
2. A buffer number is in the DCB at OPEN time.
3. A buffer length is in the DCB at OPEN time.

NOTE: The construction and format of a buffer pool is the same for dynamic buffering as no dynamic buffering.

It is suggested the instructor now discusses the use of the buffer pool when dynamic buffering is not employed. Following that discuss use of the pool when dynamic buffering is employed.

5. Ways of obtaining a buffer from the pool

a. Obtaining buffer for READ 1-V-11

1) 'S' parameters for Area and Length
The user can specify 'S' for Area and Length parameters in which case at Read execution time buffer management will go to the buffer pool for a single buffer. This will be the first available buffer (BUFFA). The length of the buffer is obtained from the **data control block. Both parameters** are placed in the DECB.

The buffer control block is updated to point to the next available buffer.

A block of the message will be read into the buffer starting at the first byte thus overwriting the link address which once existed in the first four bytes. This address is no longer required, since the buffer is not part of the pool at this time.

The block length must not exceed the length of one buffer.

NOTE: Again the instructor is reminded dynamic buffering is not being considered at this point.

NOTE: Even though buffer management is available, the user may specify an actual address of some area other than a buffer from the pool into which he wishes to read. However, the length of this area must be the same as specified for the buffer pool.

2) REQBUF
The user can request a buffer from the pool using the REQBUF macro. The user must then pass the address of the buffer to the READ via a register. He can also supply the length or still use the 'S' parameter for length in which case BTAM will get the length from the **data control block.**

NOTE: For reading, this last method of obtaining a buffer does not seem to buy the user that much.

3) RELBUF
Regardless of which method is used to obtain a buffer it is the user's responsibility to return it to the buffer pool.

The RELBUF macro is used to return the buffer to the pool. This buffer is returned to the head of the pool since the pool works on a last in first out (LIFO) basis. The buffer control block will be modified to point to this buffer as the next available, and the buffer formerly pointed to by the buffer control block will be pointed to by the one just returned to the pool.

NOTE: Prior to releasing the buffer back to the pool, the user must be sure the first 4 bytes of the buffer contain 0. It is possible to release a string of buffers back to the pool with each buffer pointing to the next with the last containing 0's in the pointer field.

b. Obtaining buffer for WRITE

1) Since the user is the only one in the position to know where the output message is located, it is his responsibility to provide BTAM with its location. Therefore, the 'S' parameter is not valid.

However, the length parameter can be 'S' in which case BTAM will go to the DCB for the length. Therefore, the 'S' parameter is used only when the output is the same length as the buffers in the pool.

2) REQBUF
With the REQBUF macro the user requests a buffer from the pool. He then fills it with the data to be transmitted and writes it out. The length of the message block is limited to the length of the buffer.

3) RELBUF
Once finished with the buffer it should be returned to the pool with the RELBUF macro.

NOTE: Frequently it may be desirable to write out a message which has been assembled as part of the program, e.g., an opening message. In this case the user need only indicate the address in his program where the message is located and its length.

NOTE: To provide more insight into the use and operation of the buffer pool, it might be pointed out that it is actually possible to release a buffer to the pool which had never been part of the pool. In other words once a buffer is taken from the pool, the pool has no idea it ever existed. Therefore, anything returned to the pool is automatically assumed to have come out of the pool.

The user should not do this in his program. It is only pointed out to further illustrate the operation of the pool.

6. Dynamic buffering

NOTE: The concept of buffer pools is the same for dynamic buffering. Therefore, the pool is built using one of the methods already described.

a. Why dynamic buffering

1) Handles variable length blocks

a) Provides capability to receive or transmit a block of data whose length may exceed the size of the buffer.

b) On Read continuous stream of data read into noncontiguous buffers.

c) On Write continuous stream of data written out of a chain of noncontiguous buffers.

2) With this capability the user can easily handle variable length blocks where it is not practical to use a buffer length for the maximum size block.

NOTE: The dynamic buffering facility of BTAM is a powerful feature and should be used when the block length is variable, particularly when there is a big difference between the minimum and maximum size message.

The user gets the dynamic buffering facility by indicating BFTEK=D in DCB.

b. Operation

1) Dynamic buffering with READ

a) Use of 'S' parameters
By using 'S' for the area and length parameters the user tells BTAM to go to the buffer pool for a buffer. While a buffer is filling, BTAM has gone to the buffer pool for another buffer and has prepared to use it if the block exceeds the one presently being filled. This process continues until enough buffers have been obtained to hold the entire block.

b) Format of buffers taken from pool 1-V-12
Since the block just read may extend over more than one buffer it is important that the user knows where each one is. The DECB points to the first buffer. Each buffer in turn will point to the next buffer in the first four bytes. The last buffer will have 0's in this pointer field. Therefore, the data is not read into the first four bytes.

The high order byte of each buffer is used to post completion when that particular buffer is full. Once the entire block has been read the primary ECB in the DECB is posted complete. It is possible for the user to wait on the ECB's in the buffers and in that manner process one buffer while another is filling.

NOTE: The problem with the above is that a buffer may be prematurely processed. That is, a buffer is posted complete and processed; however, at the end of the message LRC's don't check. Therefore, buffers which may have contained an error have been processed.

c) Effect on buffer pool 1-V-13
The reference foil indicates what happens to the pool when there are two operations occurring concurrently. Line 1 requires a buffer at times 1, 3, and 5. Line 2 requires a buffer at times 2 and 4.

NOTE: On a read the user can specify an address other than the pool; but, regardless of the length he specifies, BTAM goes to the buffer control block for the length.

NOTE: Once the user has indicated dynamic buffering is to be employed, all I/O operations will be treated as if the buffers had a dynamic buffering format, i.e., the first 4 bytes a pointer field. Therefore, if the user is doing nothing more than writing out an assembled opening message, it must have a pointer field, all 0's if it is the only buffer.

2) Dynamic buffering with WRITE

a) Area
The user must specify to BTAM the buffers to be transmitted. It operates the same as the READ, i.e., the buffers must all be linked together with 0's in the pointer field of the last.

The user obtains the buffers with a REQBUF or writes out of some area he specified at assembly time.

b) Length
The user can use the 'S' parameter which will result in buffers being transmitted until an End of Block character is encountered.

The user can indicate the actual length of the message if he desires.

c) RELBUF
Any buffers which have been obtained from the pool, by REQBUF or 'S' parameter in READ, must be returned to the pool with RELBUF.

RELBUF can release a chain of buffers back to the pool. However, the user must be sure each points to the next and there are O's in the pointer field of the last.

d) Buffering summary 1-V-14
The referenced foil summarizes the different methods of obtaining a buffer pool and the methods by which a READ or WRITE can utilize the pool.

e) Effect of dynamic buffering on Translate
Dynamic buffering effect the TRNSLATE macro in that it will translate all buffers as dynamic buffers, i.e., assumes the first 4 bytes provide a pointer field.

C. BTAM Macro Considerations

NOTE: The main purpose of this section is to discuss these macros which have not been mentioned and to point out parameters of macros which warrant special discussion.

It is recommended the instructor does not spend too much time discussing those macros and parameters which are well covered in the SRL.

Foils 1-V-23 to 1-V-36 are provided for instructors use if necessary.

1. Data Control Block (DCB) macro

   a. DSORG
      Always CX

   b. MACRF
      Indicates whether access to line group is to be READ, WRITE, or both.

   c. DDNAME

      1) Symbol which will appear in JCL DD card read at execute time.

      2) This DD card (or cards) will actually define which lines are associated with the DCB.

   d. BUFNO
      Must be entered by user when buffer pool is to be built at OPEN time.

   e. BUFL
      Must be entered by user when buffer pool is to be built at OPEN time.

      Also entered when user builds own buffer pool.

   f. BUFCB
      Address of buffer pool control block. If buffer pool built at OPEN time BTAM provides BUFCB.

      If user builds pool BUFCB, he must fill in at execution time.

      If user utilizing an area he defined at assembly time for pool, he could fill in this value at assembly time.

   g. EXLST
      Normally not used. Only the DCB exit possible. At that time user could build buffer pool.

h. BFTEK
Use D is dynamic buffering employed. This is the only way to indicate to BTAM that dynamic buffering is to be included.

i. LERB
Specifies address of line error block. This parameter used only when C included in ERROPT.

j. ERROPT
E - basic error recovery procedures (ERP)
R - read text errors to be retried, not applicable with dynamic buffering
W - write text errors to be retried, not applicable with dynamic buffering
C - indicates LERB to be kept
N - no ERP to be included
T - terminal test facilities to be included

2. Line Error Block (LERB) macro

a. Symbol
Address referenced by LERB entry in DCB.

b. NLINES
Number of lines associated with line group.

c. TRANSMCT
Threshhold for transmission count
Default option 255

NOTE: Whenever any of the threshhold values are reached, the counts are printed out. The counts are then added to the accumlative totals.

d. DATACK
Threshhold for data check count
Default option 10

e. INTREQ
Threshhold for interventions required
Default option 5

f. NOTTO
Threshhold count for non-text time outs.
Default option 5

g. Example

```
ALERB LERB 10,,,(200,20,,7),,,(240,20,25,10)
```

Two lines will have default options of 255, 10,5, and 5.
Three lines will have 200,20,5 (default option), and 7.
Five lines will have 240,20,25, and 10.

NOTE: User may indicate only "NLINES" and default options will be taken for each line.

3. Assemble Translation Table (ASMTRTAB) macro SRL contains names of all tables.

4. Define Terminal List (DFTRMLST) macro

NOTE: Terminal list is very much device dependent and the instructor is referred to the device modules of this outline for examples. This section will discuss in general terminal list considerations.

a. Purpose
The BTAM channel programs require information from the users program concerning:

Addressing characters, i.e., what stations and their components are to be addressed on a line.
Polling characters, i.e., what stations and their components are to be polled on a line.
If auto call is employed what is the telephone number of the station being called.

This is all information required by 'BTAM to control the line and only the user can provide BTAM with this data. This is the function of the DFTRMLST macro.

BTAM will actually transmit characters from the users terminal list when polling and addressing stations.

b. Parameters

1) Symbol
Name of terminal list

2) Listype

NOTE: There are a number of possible entries for list type which vary depending on the type of terminal and the features of the Transmission Control Unit. These options can be found in the SRL and the device dependent modules of this outline. However, this module discusses **three of the more important list** types to provide a basis for understanding the others.

a) OPENLST
Used for polling or addressing. On polling BTAM will poll each entry in the list until one responds with a block of data. If all terminals in list polled with no positive response, polling stops on reaching the last entry.

For addressing a terminal, an OPENLST is required.

NOTE: **For an addressing list the user will probably have a separate DFTRMLST macro for each terminal. For a polling list there may be one DFTRMLST macro for all the terminals on a line.**

b) WRAPLST
The WRAPLST is only used for polling. As soon as the polling list has been exhausted by receiving negative responses, BTAM goes back to the start of the list to continue polling.

NOTE: In use of the WRAPLST, the user must realize that BTAM will continue recycling through this list until a positive response is received or until a RESTPL macro is given.

c) DIALST
Used with 1050's on switched lines. Presence of telephone number indicates calling list and absence of telephone number indicates answering list.

3) Device dependent operands
User must be sure to use characters in the transmission code to be employed.
User must be sure he uses the characters specified by the CE's at installation time.

c. Format of expansion 1-V-15

1) OPENLST
Each entry in the list is expanded to include the addressing bytes and a control character for each pair of addressing bytes.

This control character provides control information to BTAM.

NOTE: Some terminals may only have a single addressing byte; however, they still have a control byte.

2) WRAPLST
This list will have basically the same format as the OPENLST except in this case the last item in the list will be a negative offset value which BTAM will use to get back to the beginning of the list.

3) DIALST
Basically same format as an OPENLST. However, in case of calling list has a count of the number of digits in the number followed by the number
In answering list count is 0 and no number is given.

5. Change Terminal Entry (CHGNTRY) macro

a. Operation

As just described BTAM uses characters from the terminal list to address and poll terminals. Prior to addressing or polling a station, BTAM checks the control byte associated with the characters to see if a skip bit has been set. If the skip bit is set, the terminal is skipped.

The CHGNTRY macro provides the means of setting the bit (skip) or turning the bit off (activate) for any entry in the addressing and polling lists.

b. Parameters

1) list
Address of start of list

2) listype
Describes the type of list

3) position
Position of entry in list

4) numchars
Number of polling or addressing characters per entry.

5) action
SKIP - stop polling or addressing
ACTIVATE - start polling or addressing

6. OPEN and CLOSE macros

a. Example of OPEN using standard form of macro

NOTE: Format is same for OPEN and CLOSE, therefore examples show only the OPEN.

1) Open one DCB

a) Using symbolic address
OPEN (DCB1)

b) Using register
OPEN ((r))

2) Open two or more DCB's

a) Using symbolic addresses
OPEN (DCB1,,DCB2,,....)

b) Using registers
OPEN ((R1),,(R2),,.....)

NOTE: This last example is least likely to be employed.

b. Example of OPEN using L and E forms of macro

```
          OPEN      MF=(E,DCBLIST)
           .
           .
           .
DCBLIST   OPEN      (DCB1,,DCB2),MF=L
```

In this manner the OPEN and CLOSE can refer to the same list.

NOTE: If the list should contain a DCB for a line(s) which is not available, the user simply does not enter a DD card in the JCL for that line group and it is not opened.

7. LOPEN macro

a. Purpose

At OPEN time for many devices the OPEN routine will try to issue a SAD or ENABLE command to the line. If a line error occurs at this time, the line is not opened. The user is not aware of this until a READ or WRITE macro is given for that line. At that time he gets a return code (x'14') indicating the line error occurred at OPEN. He can now try to overcome this problem by issuing an LOPEN to open just that line since the rest of the line group is open.

b. Format

Indicates DECB address for that line.

c. Return codes

There will be return codes for this macro indicating whether line was opened or not.

NOTE: For some devices, e.g., 2848, the LOPEN does not apply.

8. READ and WRITE macros

a. Form

1) L form

User will employ this in a constants area to generate the list, i.e., DECB which will be used to pass parameters to BTAM.

2) E form

User will employ this form in his executable coding. It expands to executable coding which fills in the DECB with the parameters specified by the user. If a parameter isn't specified the one already in the table is passed to BTAM.

3) Expansion of L form

Since the L form results in the DECB being built in the constants area, it is a simple matter for the user to expand the DECB to include additional information he wishes to maintain concerning a line and its related terminals. The need for this type of information becomes more apparent when the user has a system consisting of multiple communication lines with multiple terminals on each line.

b. Parameters

NOTE: Emphasize that all parameters but "optype" can be passed in registers and in actual practice this will normally be the case.

1) decb
Address of DECB associated with line

2) optype
Specific type of operation. In L form of macro can use any "optype" or just "T".

3) dcb
Address of DCB for line group

4) area - address of first byte of I/O area

a) Read
Can use symbolic address or 'S'. Use of 'S' tells BTAM to get buffer from pool.

NOTE: Keep in mind that when employing dynamic buffering the data is actually found starting in the fifth byte of the buffer. The first four bytes are used as an ECB and pointer to next buffer.

b) Write
User must provide the address for the write.

5) length

a) Read
User can specify a count or 'S'. If 'S', BTAM gets length from DCB.

NOTE: With dynamic buffering, length includes four bytes for the link field.

NOTE: If user is not using dynamic buffering, he must be sure the buffer is large enough to contain the text and the control bytes. For example, the first block consists of a start of text character followed by text which is concluded with an end of block character (EOB). These two control characters are read into the buffer.

NOTE: Receiving of EOB terminates the READ regardless of the size of the count.

b) Write
User may use actual count or 'S'.

NOTE: The user must be sure the count is at least great enough to include the EOB character. As soon as this is transmitted the WRITE is completed regardless of the count indicated in the length parameter. If dynamic buffering with 'S' parameter, blocks are written until EOB encountered.

6) entry

a) Non-switched line
For OPENLST:

Parameter must specify an address of an entry in the list

For WRAPLST (i.e., a poll list):
Parameter can specify an address of any entry or an 'S'. When using the 'S' parameter polling will start with the address of the next entry if a RESETPL macro was used to stop polling previously. If RESETPL was not used, polling will start with the last entry polled.

NOTE: The DECB has a field which points to the current entry in the poll list. If polling should be stopped because of a RESETPL macro this current entry pointer is incremented to point to the next entry. Therefore, when polling is restarted by the next read, the 'S' parameter tells BTAM to use this polling pointer field in the DECB to start polling. If the previous read happened to be terminated by a positive response to polling, i.e., a message was received, this polling pointer field would not be incremented. Therefore, when the next read is issued the 'S' parameter causes it to start polling with the last terminal polled. In effect it will "poll to exhaustion", i.e., poll a station until it has no more messages to send.

NOTE: If user employs 'S' parameter, he must be sure that an initial entry gets into the DECB for the first read. This can be done in two ways:

- In L form of macro put address of beginning of list.
- By the user's program placing the starting address of the list in the polling pointer field (DECPOLPT).

NOTE: The disadvantage of the first method, which the second overcomes, is there may be a WRITE given before the READ in which case the "entry" parameter placed in the L form of the macro is clobbered by the WRITE's entry.

b) Switched line
User specifies address of beginning of list.

7) RLN
Specifies relative line number within group.

Determination of the relative line number is completely dependent on the DD cards for the line group which are read at execute time. Examples are the best way to illustrate this.

Example 1

```
//TERM     DD   UNIT=020
//         DD   UNIT=022
//         DD   UNIT=023
```

In this example RLN 1 will be 020, RLN 2 will be 022, and RLN 3 will be 023.

Example 2

```
//TERM     DD   UNIT=020
//         DD   UNIT=023
//         DD   UNIT=022
```

In this example RLN 1 will be 020, RLN 2 will be 023, and RLN 3 will be 022. Note the significance of the order of the DD cards.

Example 3

```
//TERM    DD   UNIT=(GRPONE,3)
```

In this case GRPONE was a name assigned to lines at SYSGEN time. Assuming the macro at SYSGEN was:

```
          UNITNAME  UNIT=(020,030,040)
                    NAME=GRPONE
```

then the user would use RLN 1 for 020, RLN 2 for 030, and RLN 3 for 040.

NOTE: There is a DECB for every line. Therefore, it is quite likely the user will indicate relative line number in only the L form of the macro and the E forms need never give that parameter.

8) MF
Already discussed

c. Return codes
Following the issuing of a READ or WRITE macro, the user must check return codes in register 15 to see if the operation was started all right and if not why it wasn't.

Following is the information provided by the return codes:

Started OK
Line busy
Invalid "rln"
Invalid "optype"
Skip bits on for all entries (see CHGNTRY macro)
Line error during OPEN (see LOPEN macro)
Buffers not available
No buffer pool
No buffer routine

NOTE: See SRL for additional details on return and completion codes.

d. Completion codes
As soon as the I/O operation is complete, it is the users responsibility to check how it is completed.

There are 3 possible completions:

Normal
I/O error
Enable command halted or I/O operation purged

9. Reset Polling or Reset Line (RESETPL) macro

a. Purpose
The value of the RESETPL macro is that it permits the user to terminate polling on a line. For example, if polling with a WRAPLST is in progress and there is something to send on the line, the RESETPL can be issued to terminate polling so a WRITE can be issued.

To restart polling the user merely issues another READ initial macro.

1) Operation non-switched line
If polling in process and negative response received, polling terminated and operation posted complete with X '7F'.

If positive response or time-out occurring, operation will proceed to normal completion.

NOTE: See discussion on "entry" in Read macro to see results of RESETPL on polling pointer.

2) Operation switched line

NOTE: With switched lines BTAM must see that an ENABLE command is issued to the line before any calls can be received. The channel program will not start polling until the connection is completed.

If the connection has not been made, the RESETPL causes IOS to issue a HALT I/O to the line. A completion code of x '48' is posted in this case.

If the connection has been made, the operation of the RESETPL is the same as described for non-switched lines.

b. Parameters

1) decb
Address of data event control block

2) POLLING or ANSRING
POLLING

Macro generated will handle non-switched lines.

ANSRING
Macro generated will handle switched lines.

NOTE: If neither POLLING or ANSRING specified, instructions will be generated to determine at execution time which situation exists.

c. Return codes
Following this macro there are return codes which must be checked.

NOTE: See SRL for these codes.

10. Translate (TRNSLATE) macro

NOTE: When using the TRNSLATE macro and dynamic buffering has been specified, it is important the user makes sure all information to be translated adheres to the dynamic buffering format. This includes messages that have been assembled in core and are written out, e.g. opening messages.

a. Parameters

1) dcb
Address of DCB. Uses DCB to obtain length in same cases and to see if dynamic buffering specified.

2) table
Specifies name of translate table
Will be name used in ASMTRTAB macro.

If register notation is employed, user loads register with IECTXXXX where XXXX is name of table.

3) area
Specifies starting address of buffer to be translated.

When dynamic buffering is employed,the TRNSLATE will translate the chain of buffers using the link addresses to go from one to another. The last must have O's in the link address.

4) length
Specifies number of bytes to be translated.

'S' parameter indicates chain of buffers and length is available in DCB.

If register notation is employed, an O in the register has the same effect as the 'S' parameter.

11. Line Error Print (LERPRT) macro

a. Purpose
If the user is keeping transmission and error counts (kept in LERB), when any thresholds are reached the counts are printed. However, the user at any time, particularly at the end of the program, can call for a print out of these counts even if the thresholds have not been reached. The LERPRT macro does this.

Before the print out of these counts under either circumstance, the threshold counters are added to the accumulative counters and the accumulative counters are printed out. The threshold counts are then set back to O (optional when LERPRT used) and as transmissions continue these threshold counts are added to until one of them becomes as great as the thresholds specified in LERB or until another LERPRT is given.

b. Parameters

1) dcb
Address of DCB which contains LERB address

2) rln
Specific line for which counts are to be printed.

If O all non zero entries in LERB printed.

3) CLEAR

a) NO
Specifies threshold counters and accumulators are not to be reset to O after LERPRT

b) YES
Specifies error threshold counts and accumulators are to be set to O after LERPRT.

12. Request Buffer (REQBUF) macro

a. Purpose
Obtains one or more buffers from buffer pool. Last buffer contains O's in pointer field.

Normally this macro will be employed when a buffer(s) is required for a block to be transmitted.

On reads the read routine itself will issue the REQBUF if the user has indicated the 'S' parameter for 'area'.

b. Parameters

1) dcb
Address of DCB

2) (r)
Specifies general register into which address of first buffer is to be placed.

3) count
Specifies number of buffers requested.

NOTE: A buffer pool must have been assigned to DCB in order for this macro to work.

c. Return codes
Following request buffer, user must check return codes to see if:

Buffers obtained
Some obtained but not all
No buffers obtained
No buffer pool exists
No buffer routine exists

13. Release Buffer (RELBUF) macro

a. Purpose

It is always the users responsibility to return buffers to the pool. He does this with RELBUF macro.

Can return 1 buffer or a chain of buffers. Last buffer must always have 0's in pointer address. If only 1 buffer returned, it must have 0's in pointer address whether dynamic buffering employed or not.

b. Parameters

1) dcb
Specifies DCB address

2) (r)
Specifies register into which address of first buffer must be placed.

c. Return codes
Following RELBUF user must check register 15 for return code which will indicate one of the following:

Buffers released
No buffers returned because first buffer in chain had been returned previously
No buffer pool
No buffer routine

14. TWAIT

a. Purpose
Following a read or write the user continues processing until nothing more can be done because he must wait for an I/O operation to complete. At this time he issues a wait macro.

This wait can be entered with the regular OS WAIT macro or the TWAIT macro which is a facility of BTAM.

The TWAIT macro utilized the regular wait facility of OS, but the macro expansion provides additional coding which determines which event out of a number of events has completed. Therefore, the user must define a list of events on which he will wait. The completion of any of them will complete the wait and the TWAIT macro will furnish him with the address of the event completed.

The TWAIT is also utilized for the on-line terminal test facility.

b. Parameters

1) (r)
User specified register which will contain address of ECB posted complete

2) TERMTST
Specifies on-line terminal test facility is provided. Must also have ERROPT=T in DCB.

NOTE: If on-line terminal test requested, as soon as an I/O operation is completed the TWAIT routine passes control to the on-line terminal test routine to see if the user is trying to exercise this facility. If he isn't the users program regains control.

3) ECBLIST
Specifies address of user-created list of ECB addresses.

High order byte of each entry is set to 0 except for last entry which is set to 1.

c. Example

```
     TWAIT (5), ECBLIST=LIST
        .
        .
        .
LIST DS    OF
     DC    A(DECB1)
     DC    A(DECB2)
     DC    X'80'      END BIT
     DC    AL3(DECB3)
```

NOTE: The register the user provides for the address of the ECB posted complete should be the one he is using for the register to contain the DECB address since the ECB address is the same as the DECB address. The reason for using this register becomes evident when he starts to write a program for a multiple line system.

NOTE: When one of the events complete, the user should be sure to clear the ECB of the complete event prior to returning to the TWAIT to wait on the next event completing. If he doesn't the same event will cause him to drop through the wait.

This last precaution is not necessary should the user issue another I/O operation on the completed DECB prior to returning to the TWAIT.

D. Error Recovery Procedures (ERP)

1. Functions - set of routines designed to diagnose and recover, if possible, from all errors encountered during transmission.

   a. Basic capabilities

      1) Automatic retry of all errors not involving data transfer.

      2) Statistical recording of temporary and permanent unit check errors.

      3) Error messages to operator for all permanent errors.

      NOTE: If ERP not wanted can indicate such in DCB.

   b. Other capabilities that can be specified.

      1) Maintain a count in the line error block (specified by LERB macro) for data checks, time outs, and intervention required errors, and the number of transmissions.

      2) Retry write operations when errors in data transfer.

      3) Retry read operations when error in data transfer.

2. Operation

   a. User must indicate in LERB and ERROPT parameters of DCB which functions are to be performed.

   b. IOS passes control to ERP.

   c. ERP checks condition codes, sense bytes, and status bytes in CSW to determine type of error and passes control to proper module.

   NOTE: The BTAM II SRL (C30-2004) provides a description of the actions taken for the various conditions.

3. Special considerations

   a. Dynamic buffering

      1) Text retry
         When using dynamic buffering, ERP will not retry text I/O errors.

2) Users solution to text retries.

With little effort the user can retry transmitting the message. First he must check the completion code to determine an error occurred. Also he can check the sense byte in the DECB to be sure it was a text parity error, i.e., data check (READ) or bus out check (WRITE).

Next, for READ, release buffers for bad operations. This protects against blocks being different size on retry.

Lastly issue Read Repeat or Write to retry.

NOTE: On a Read Repeat following a Read Initial the user may not find a start of text as the first character. Therefore, he must compensate for the characters all being shifted over by one. Whether this problem occurs depends on the type of device and the component being polled.

b. Regular buffering

If the user specifies text retries, ERP will perform them. However, one caution should be observed. If an error occurs ERP will retry without the user becoming involved. If the retry is successful, the operation is posted complete with a good code, i.e., X'7F'. If this repeat should be following a Read Initial the start of text character may not appear in the repeat, but the user, not knowing a retry was made, would expect the first character to be a start of text character.

E. Channel programs

1. Purpose
BTAM generates the channel programs to perform the I/O operation requested in the users I/O macro.

There are a number of possible channel programs which can be called. Model programs are contained in the Device I/O module and are modified to reflect the addresses specified by the user for I/O areas, terminal lists, and byte counts.

2. Users concern with channel programs
The channel programs besides transmitting text are also responsible for exercising the line control required to control the devices on the line.

These devices are dependent on receiving the proper control characters in the proper sequence. If these rules are not observed problems will be caused on the line.

As mentioned there are a number of possible I/O operations (channel programs) each resulting in the transmission of certain line control characters and frequently text.

By specifying 'optype' in the read and write macros the user is also specifying the line control characters to be transmitted or received. Therefore, the sequence of I/O macros controls the sequence of line control characters. Should a macro not be given or given in the wrong sequence it will result in certain line control characters not being given or given in the wrong sequence, thus causing transmisstion problems.

3. Basic types of channel programs

NOTE: The instructor is referred to the I/O modules for specific channel programs. This section is provided to indicate in general what channel programs exist and at what time they are used.

a. Read Initial and Read Initial with Reset

1) Purpose
Used to initiate polling of a line.

2) Operation

a) Puts line in control mode
b) Transmits polling characters
c) Reads response to poll which can be:
   - Negative response - in which case polling will be terminated or the next terminal will be polled (BTAM will take care of this)

NOTE: The action taken depends on the entry in the poll list being polled and the construction of the polling list.

- A positive response followed by the first block of the message when the end of block character is received the read is terminated and the user receives control again.

3) Users next action
The terminal just read expects a response from the CPU. This can be achieved in a number of ways.

a) If the user expected only a single block message, he could have issued a Read Initial with Reset which would send a positive response to the terminal and place it back in control mode. He could also follow a Read Initial with a Write Positive Acknowledgement to achieve same result.

b) If the user expected additional blocks he would issue a Read Continue.

b. Read Continue

1) Purpose

Issued following last read to indicate to terminal it was a good transmission and to read next block from terminal

2) Operation

a) Sends positive response to last read

b) Reads text of next block from terminal

3) Users next action
Terminal again expects indication of how the Transmission went

a) Read Continue could be issued to again see if terminal has another block

b) Read Continue with Reset will cause the Read Continue to conclude with a positive response and control characters to put the line back in control mode.

c. Read Repeat

1) Purpose
Issued if last read resulted in error

2) Operation

a) Read Repeat sends out a negative response so the terminal will retransmit

b) Next reads the retransmitted block

3) Users next action

Must indicate to terminal how transmission went.

a) Read continue

b) Read Repeat with Reset would transmit positive response and place line in control mode.

d. Other methods to terminate transmission

1) EOT from terminal
Normally when a Read Continue is employed the terminal will finally terminate by sending an end of transmission character.

2) Reset from CPU
Already discussed

3) Write Positive Acknowledgement
Will cause CPU to transmit a positive response and then control characters to put line in control mode.

4) Write Negative Acknowledgement
CPU would use this to indicate a bad transmission and then put line into control mode.

e. Write Initial and Write Initial with Reset

1) Purpose
Used to select a particular terminal or terminals and transmit a block to it.

2) Operation

a) Establishes control mode on line

b) Transmit addressing characters

c) Read response to see if terminal ready. If not operation is terminated.

d) If positive response, write block

e) Read response from terminal.

3) Users next action

a) If only one block to be sent, user could have specified Write Initial with Reset in which case the line is put back into control mode following receiving of positive response on transmission from terminal.

b) If another block is to be transmitted the user would issue a Write Continue

f. Write Continue and Write Continue with Reset

1) Purpose
Used following Write Initial to transmit next block.

2) Operation

a) Writes message block

b) Read response from terminal

c) If reset specified will write control characters to put line back in control mode.

g. Examples of sequences

1) Reading a single block message
Read Initial with Reset **or**
Read Initial **and Write Positive Acknowledgement**

2) Reading multiple block messages
Read Initial
Read Continues until EOT from terminal

3) Reading a single block with error
Read Initial with Reset
Read Repeat with Reset

4) Reading multiple block message with error
Read Initial
Read Continue
Read Repeat
Read Continues until EOT from terminal

5) Write a single block message
Write Initial with Reset

6) Write multiple block message
Write Initial
Write Continue
Write Continue with Reset

NOTE: It is important to realize this section is concerned with only discussing channel programs in general and that variations will exist for different devices.

F. General OS BTAM Considerations

1. When using BTAM dynamic buffering if the EOB character comes into the last position of the buffer, this will cause BTAM to pick up an extra buffer and not release it. It is up to the user to release that additional buffer before writing out of that buffer area or before processing them. This consideration is written up in the BTAM SRL.

2. Any areas of DECB which are labeled reserved must not be used by the user as BTAM will use them. For example, the area labeled DECAREA has one byte labeled reserved and three bytes used for the address of the message area or the address of the first buffer. If the user stores the address in here himself, he must be sure that the reserved byte is cleared to zero as BTAM expects to find it that way.

3. When using switched lines after a write initial use a write negative acknowledgement to send the EOT sequence before disabling the line. If this not done, the terminal will hang up with a data check.

4. All lines in the same line group must have exactly the same features. The OPEN expects the feature portion of the UCB's to look exactly alike.

5. When all terminals in a terminal list have their skip bits on you will get into an endless loop in BTAM.

6. When processing a message read in via the read repeat macro, the user must remember that the first character of this message will not be a circle D.

7. The translate macro limits the buffer size to a maximum of 256 bytes.

8. The entry operand in a read or write conversational macro must point to a dial list. The OS BTAM SRL says this should point to an open list or the user must do the address arithmetic on a dial list himself. Read and write conversational work satisfactorily if the user puts an answering type dial list as the entry operand.

9. If you use an S parameter for terminal list entry on a read initial, BTAM will utilize the poll pointer in the DECB. This means that it is up to the user to initialize that poll pointer field before executing the read initial for the first time.

10. Do not use the S parameter for the terminal list entry on a switched line. After the first time the pointer will always be in the wrong place.

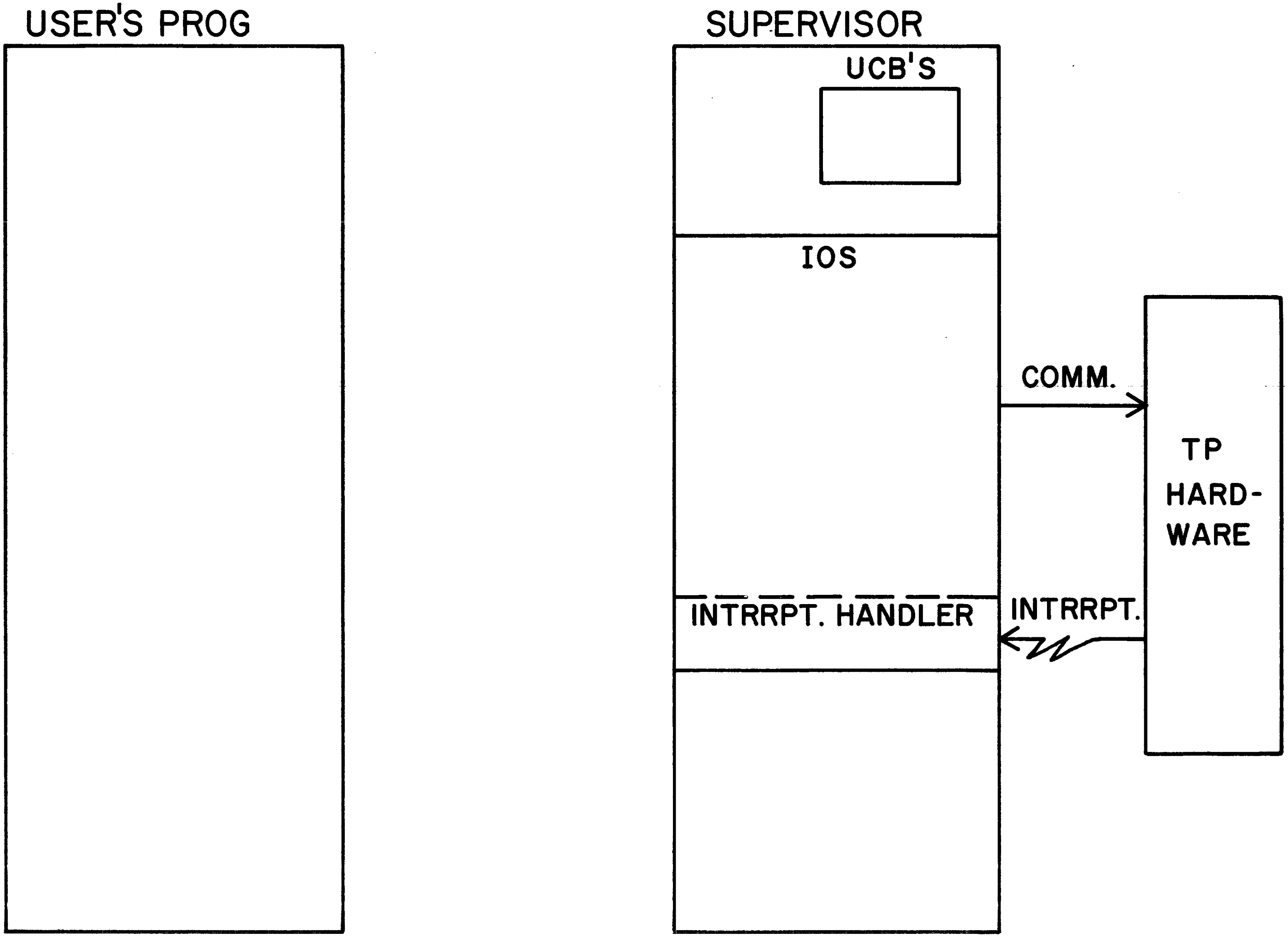
11. EXCP resets the ECB. If you go back to the TWAIT without starting I/O the previous completion code will still be set and you will fall through the TWAIT on a previous I/O completion.

12. The read initial with reset macro and the write initial with reset macro both post normal completion in the ECB even if a data check occurs.

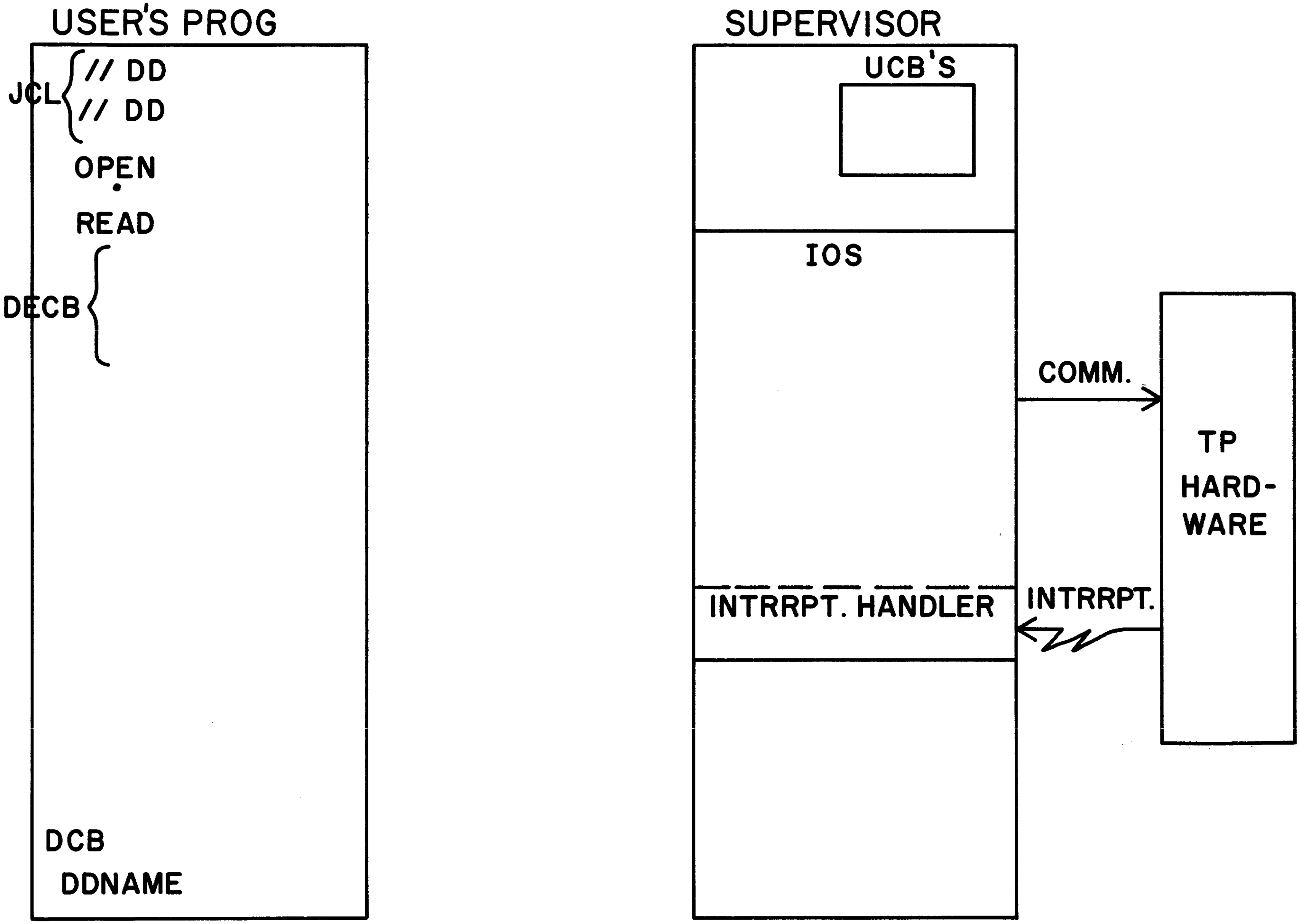
13. The location of the IOB is offset by the size of the IOB and its largest channel program.

NOTE: Many of these statements will always pertain to BTAM. However, a few may not pertain to latter releases of OS. This list references release 13.

SYSTEM OPERATION



SYSTEM OPERATION

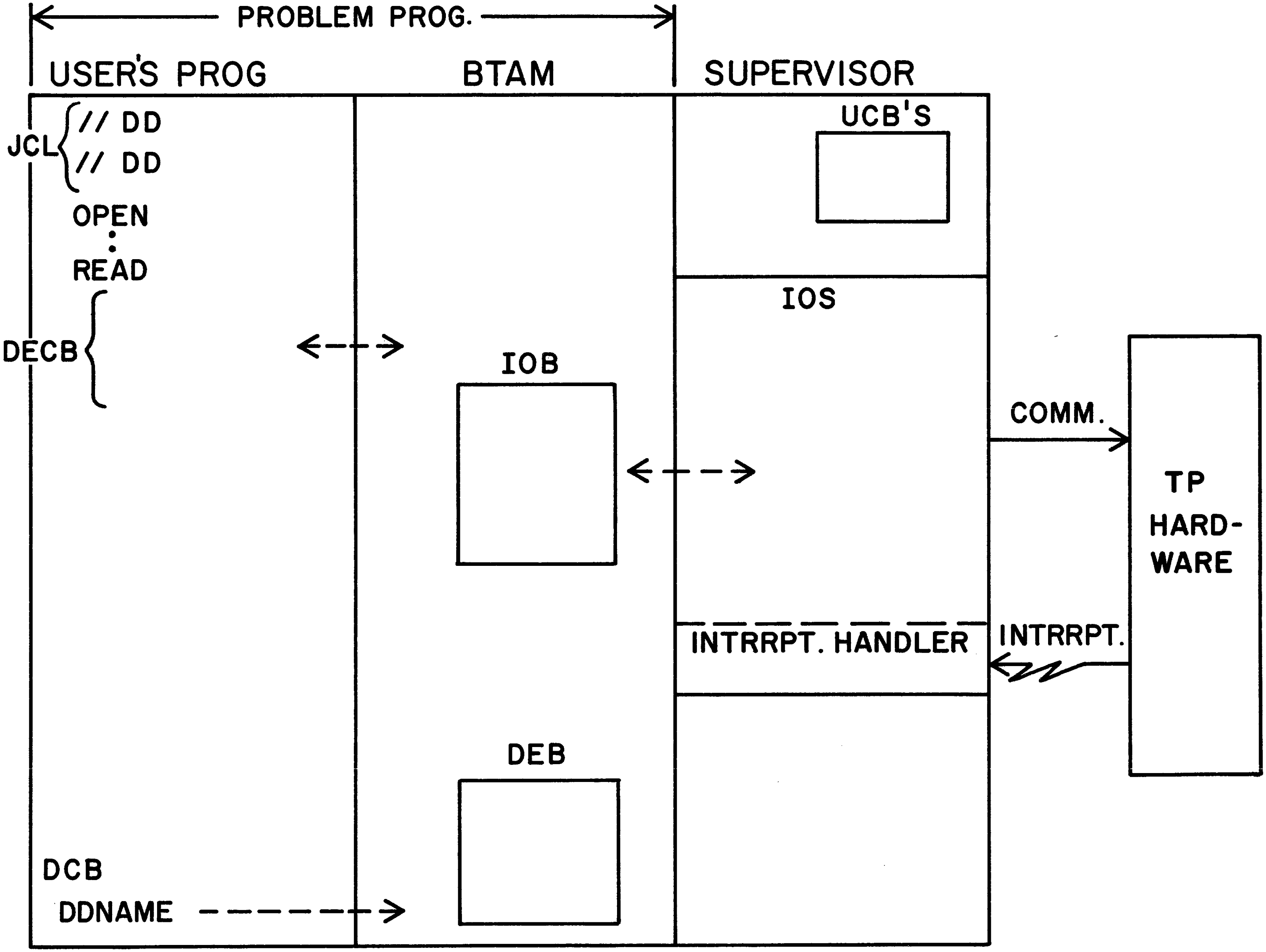


DD STATEMENTS

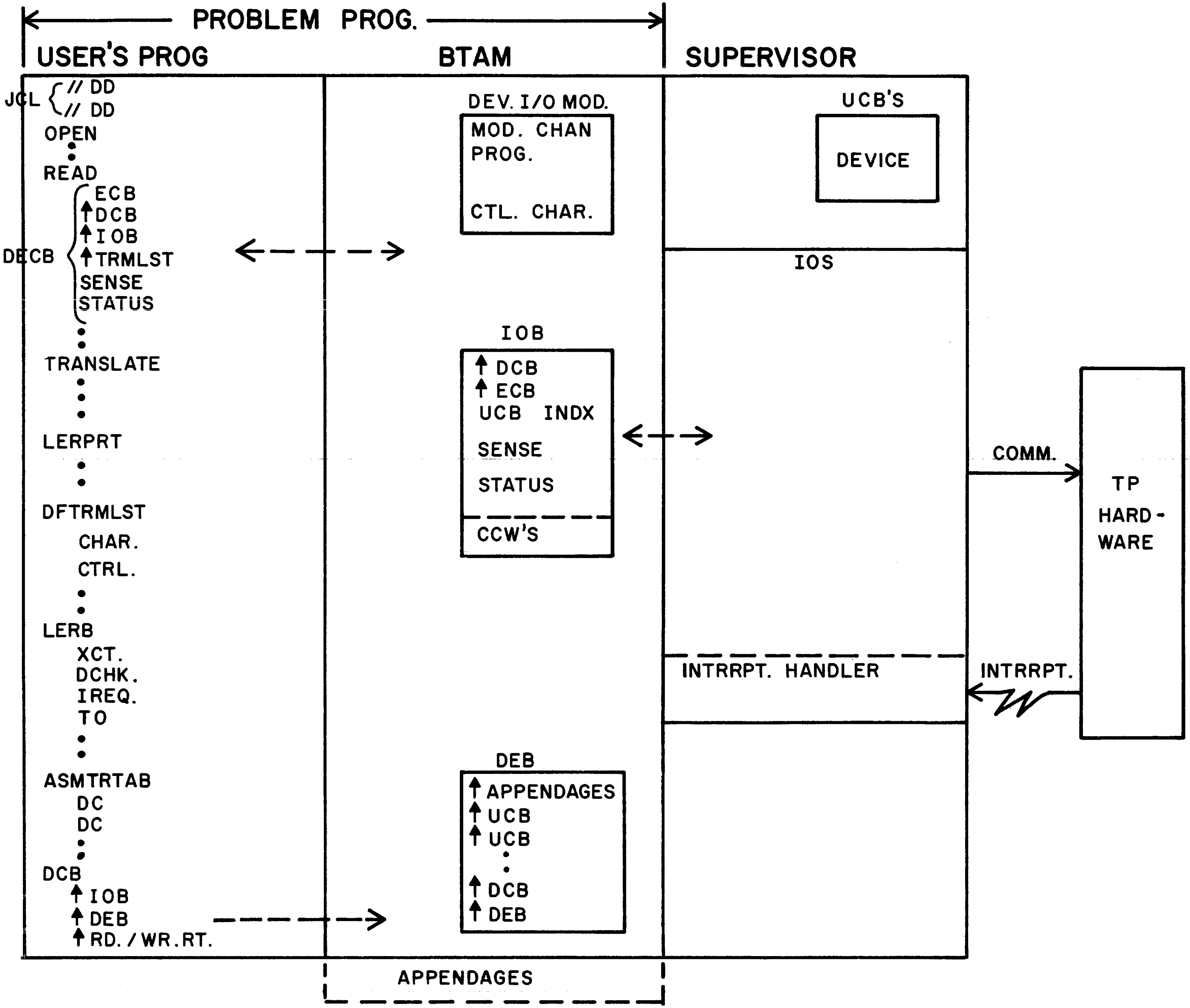
```
//GRP1    DD   UNIT=(GROUPONE,3)
//GRP2    DD   UNIT=(GROUPTWO,2)
           :
          ETC.
```

```
//GRP1    DD   UNIT=021
//        DD   UNIT=022
//        DD   UNIT=031
```

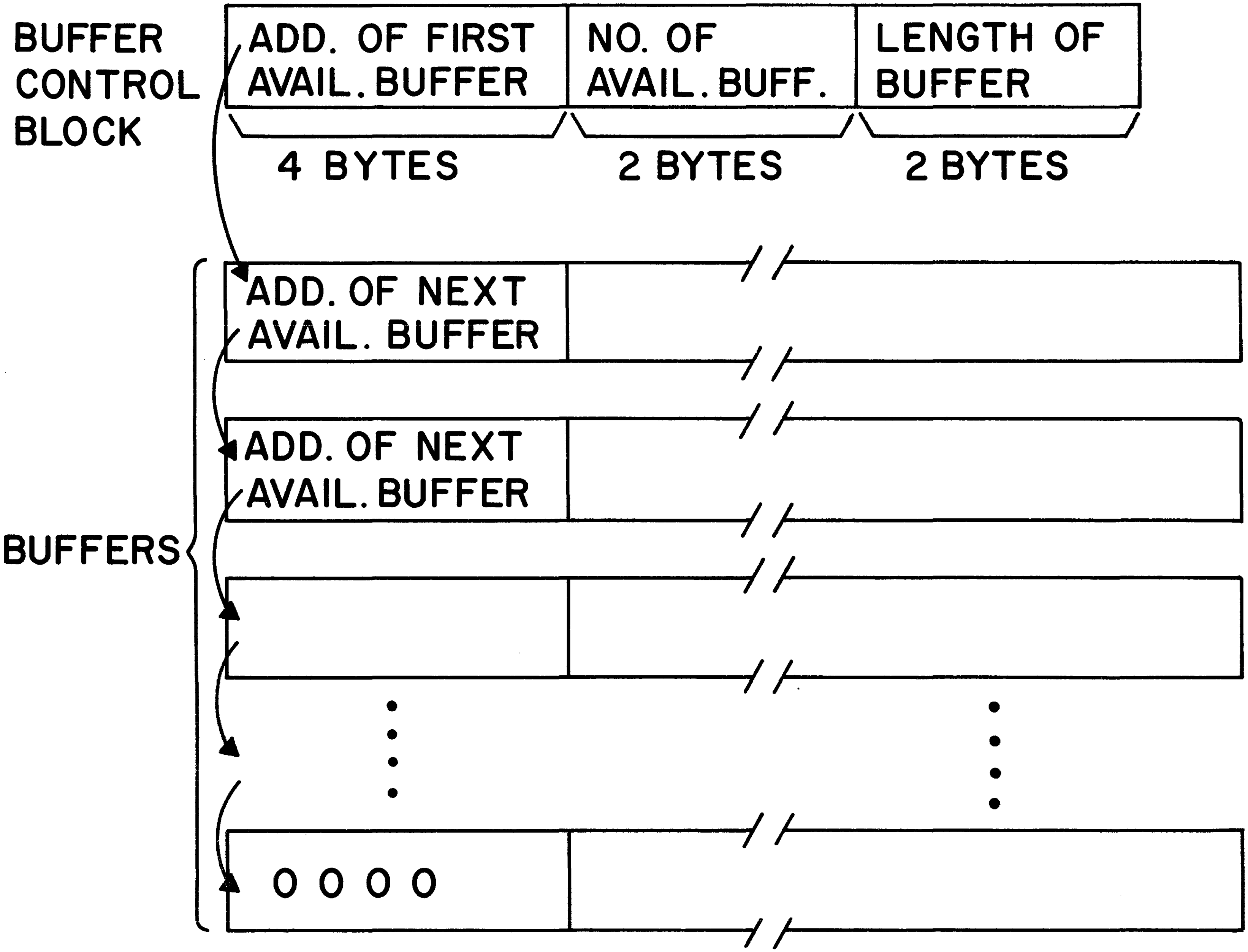
SYSTEM OPERATION



SYSTEM OPERATION



FORMAT OF BTAM BUFFER POOL



BUFFER CONTROL BLOCK

ADD. OF FIRST AVAIL. BUFFER

NO. OF AVAIL. BUFF.

LENGTH OF BUFFER

4 BYTES

2 BYTES

2 BYTES

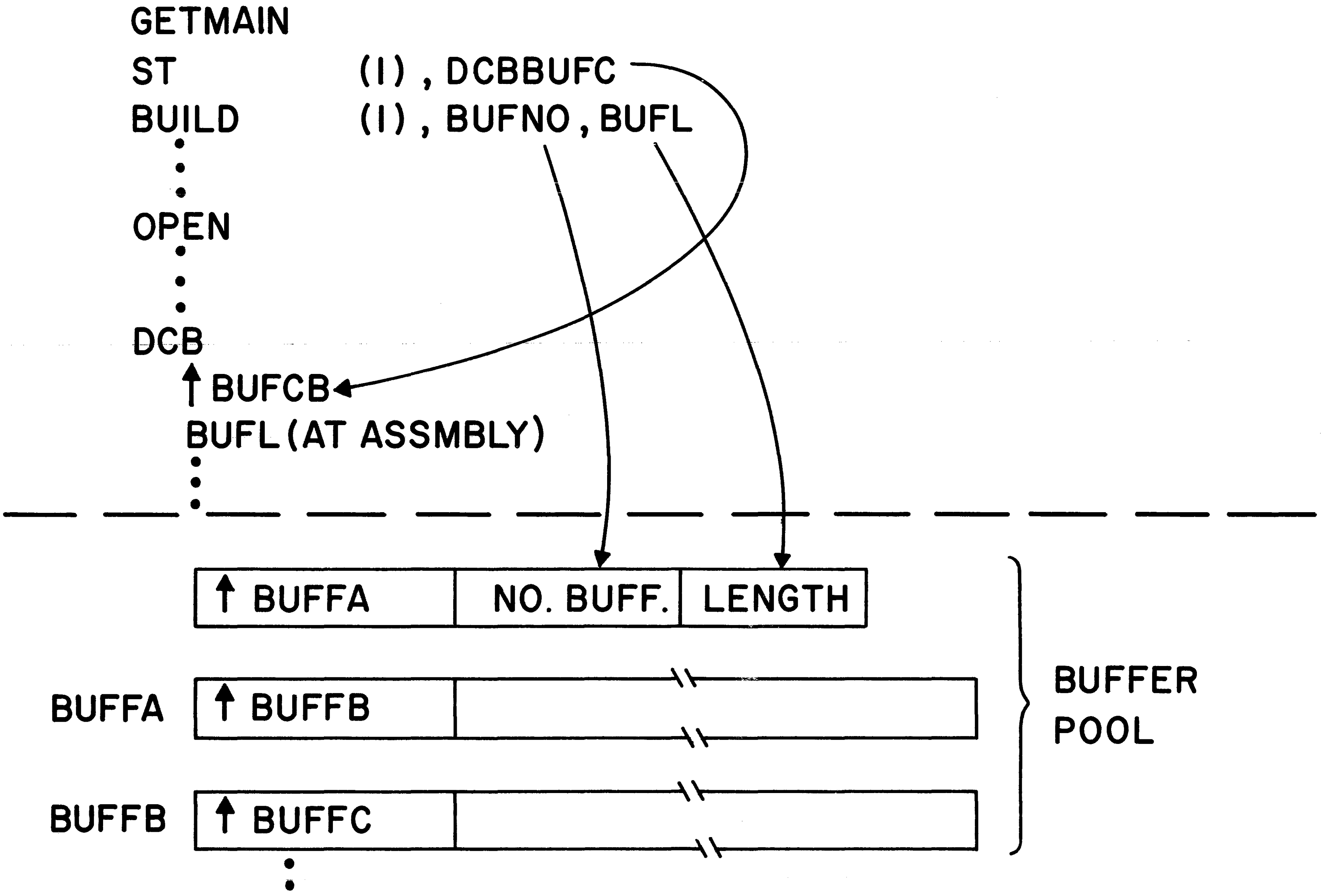
BUFFERS

ADD. OF NEXT AVAIL. BUFFER

ADD. OF NEXT AVAIL. BUFFER

0 0 0 0

BUFFER PROVIDED BY GETMAIN & BUILD



```
GETMAIN
ST              (1), DCBBUFC
BUILD           (1), BUFNO, BUFL
  .
  .
  .
OPEN
  .
  .
  .
DCB
   ↑BUFCB
   BUFL (AT ASSMBLY)
  .
  .
  .
```

↑ BUFFA	NO. BUFF.	LENGTH

BUFFA | ↑ BUFFB |

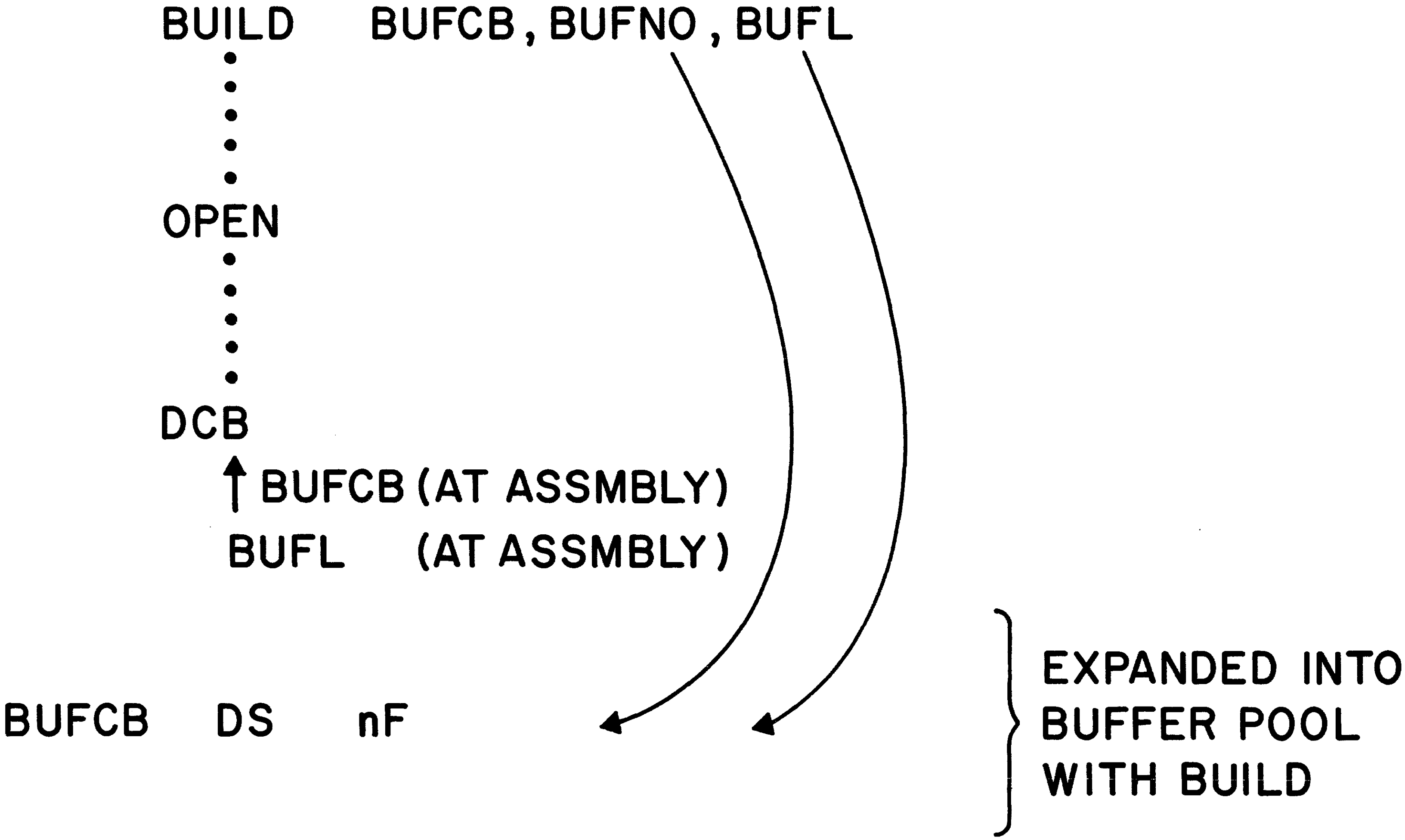
BUFFB | ↑ BUFFC |

.

.

BUFFER POOL

BUFFER PROVIDED BY DS & BUILD



BUILD BUFCB, BUFNO, BUFL

OPEN

DCB

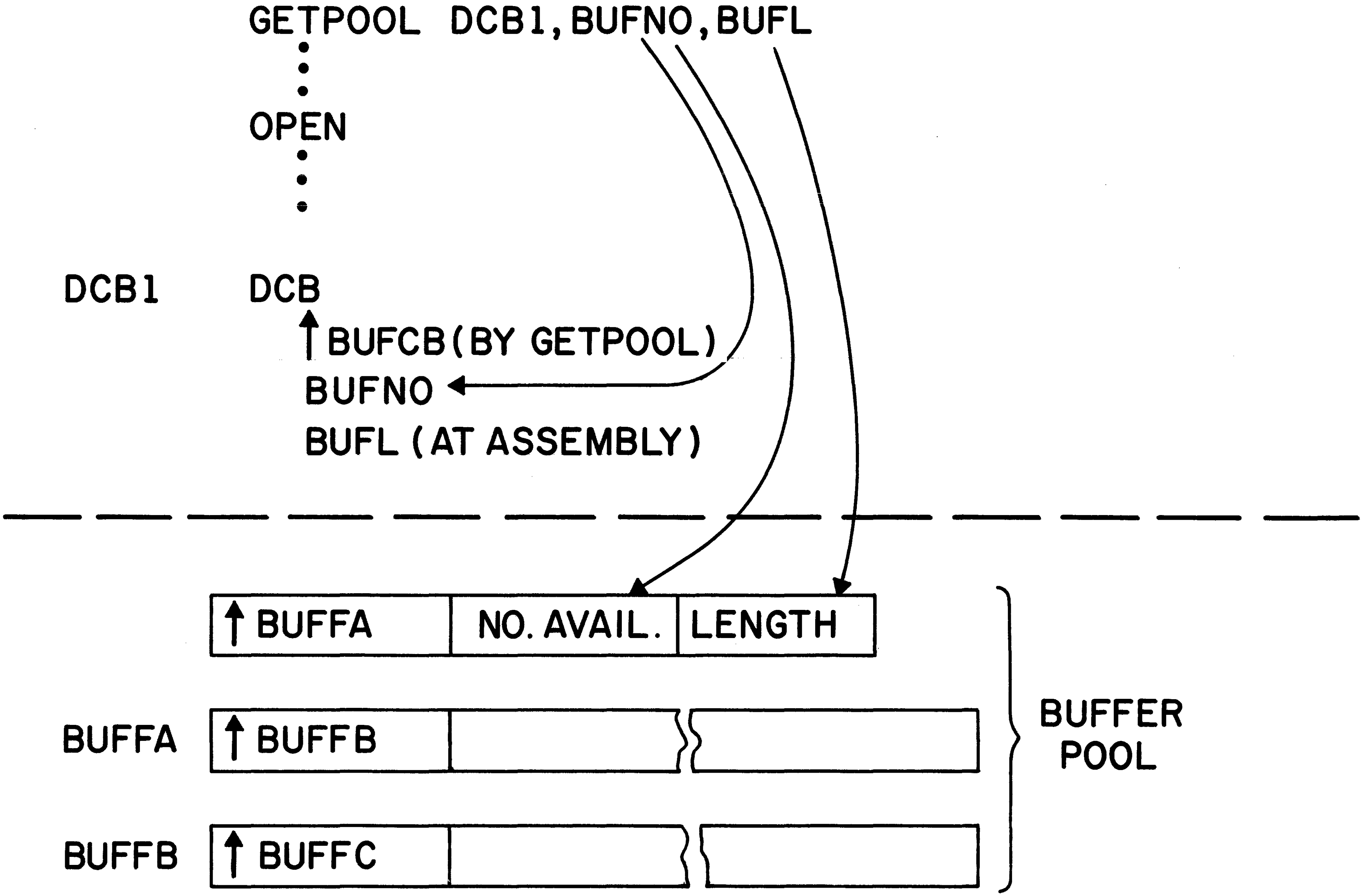
BUFCB (AT ASSMBLY)

BUFL (AT ASSMBLY)

BUFCB DS nF

EXPANDED INTO
BUFFER POOL
WITH BUILD

BUFFER PROVIDED BY GETPOOL



```
         GETPOOL  DCB1,BUFNO,BUFL
            •
            •
            •
         OPEN
            •
            •
            •

DCB1     DCB
           ↑BUFCB(BY GETPOOL)
           BUFNO
           BUFL (AT ASSEMBLY)
```

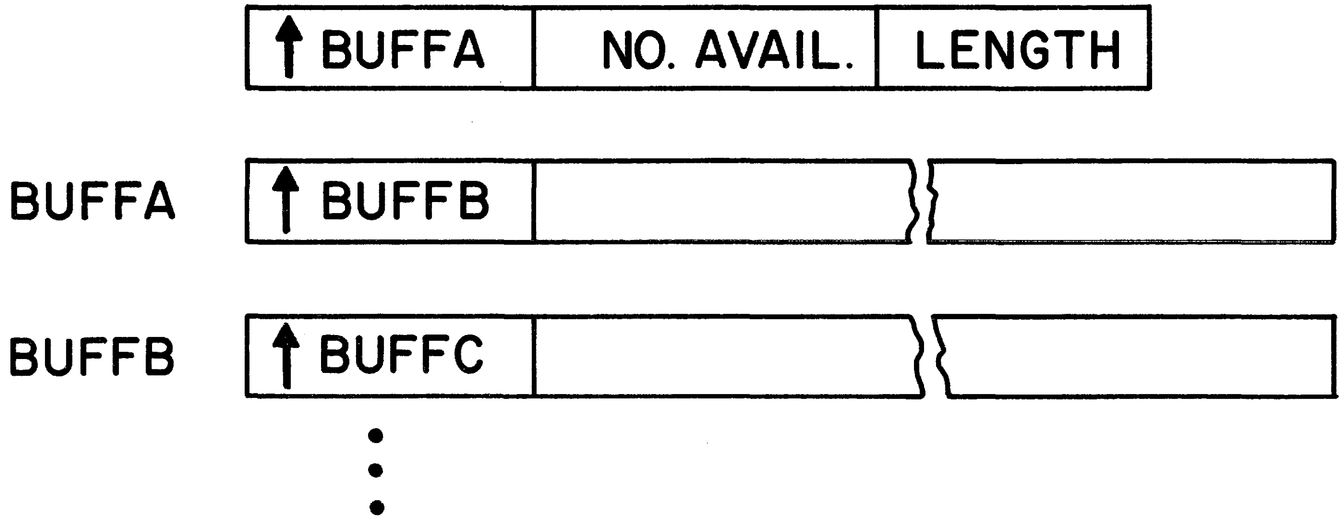
	↑ BUFFA	NO. AVAIL.	LENGTH
BUFFA	↑ BUFFB		
BUFFB	↑ BUFFC		

BUFFER POOL

BUFFER PROVIDED BY OPEN

```
        OPEN DCB1
          .
          .
          .
          .
          .
          .

DCB1    DCB
         ↑BUFCB  (AT OPEN EXEC)
         BUFNO   (AT ASSMBLY)
         BUFL    (AT ASSMBLY)
```



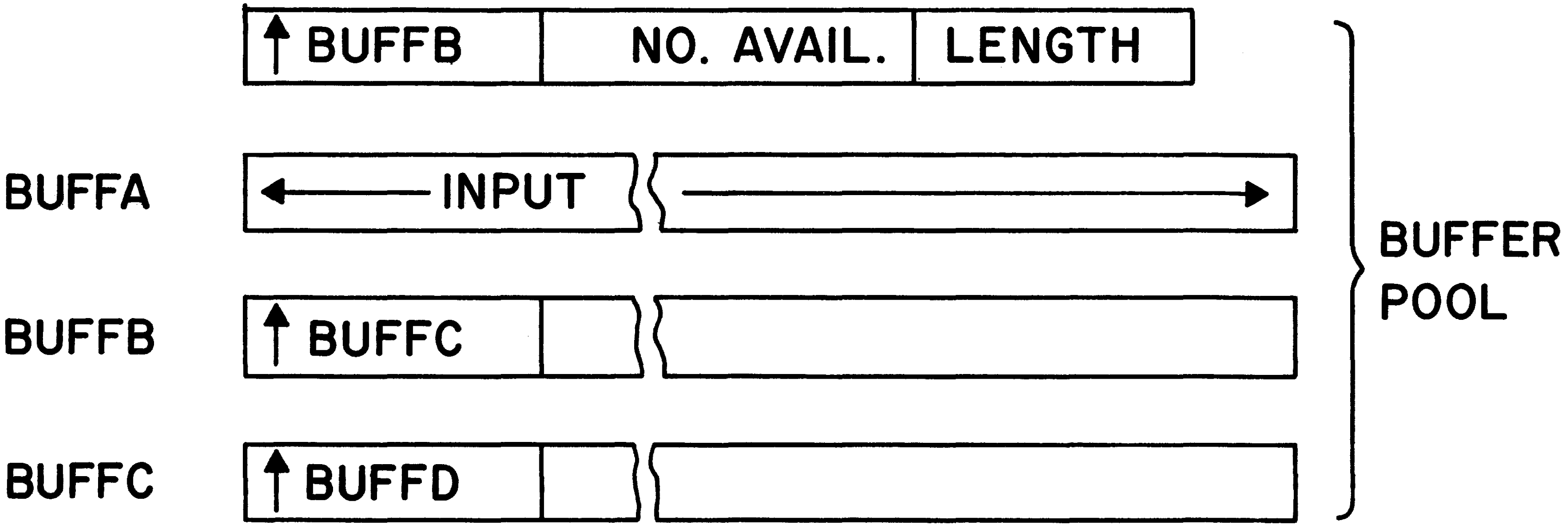
↑BUFFA	NO. AVAIL.	LENGTH

BUFFA ↑BUFFB

BUFFB ↑BUFFC

BUFFER POOL AFTER READ

```
READ DECB1,TI,,'S','S',................
   .
   .
   .
   .
   .
   .
   .
   .
   .
   .
   .
```



↑ BUFFB NO. AVAIL. LENGTH

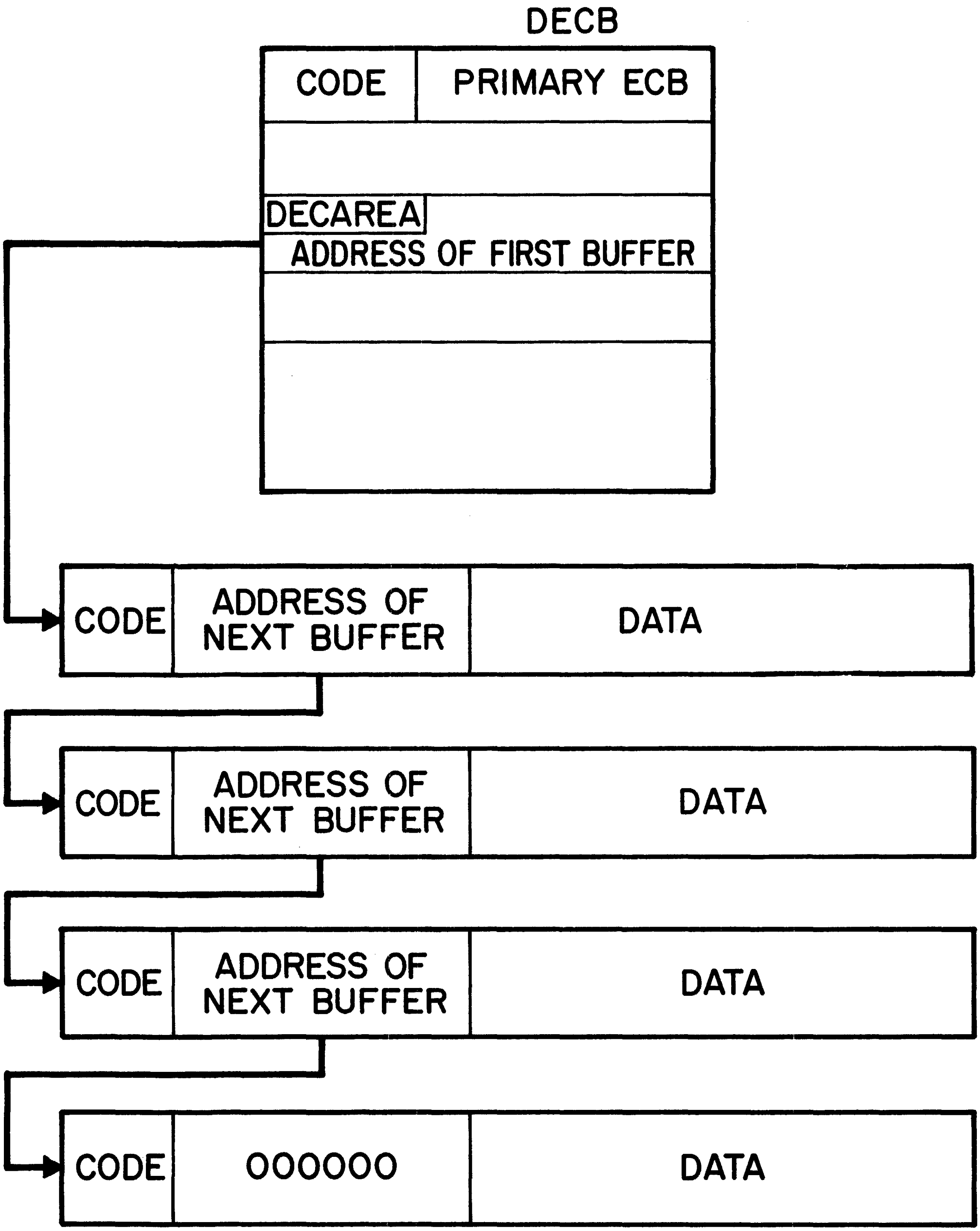
BUFFA ← INPUT →

BUFFB ↑ BUFFC

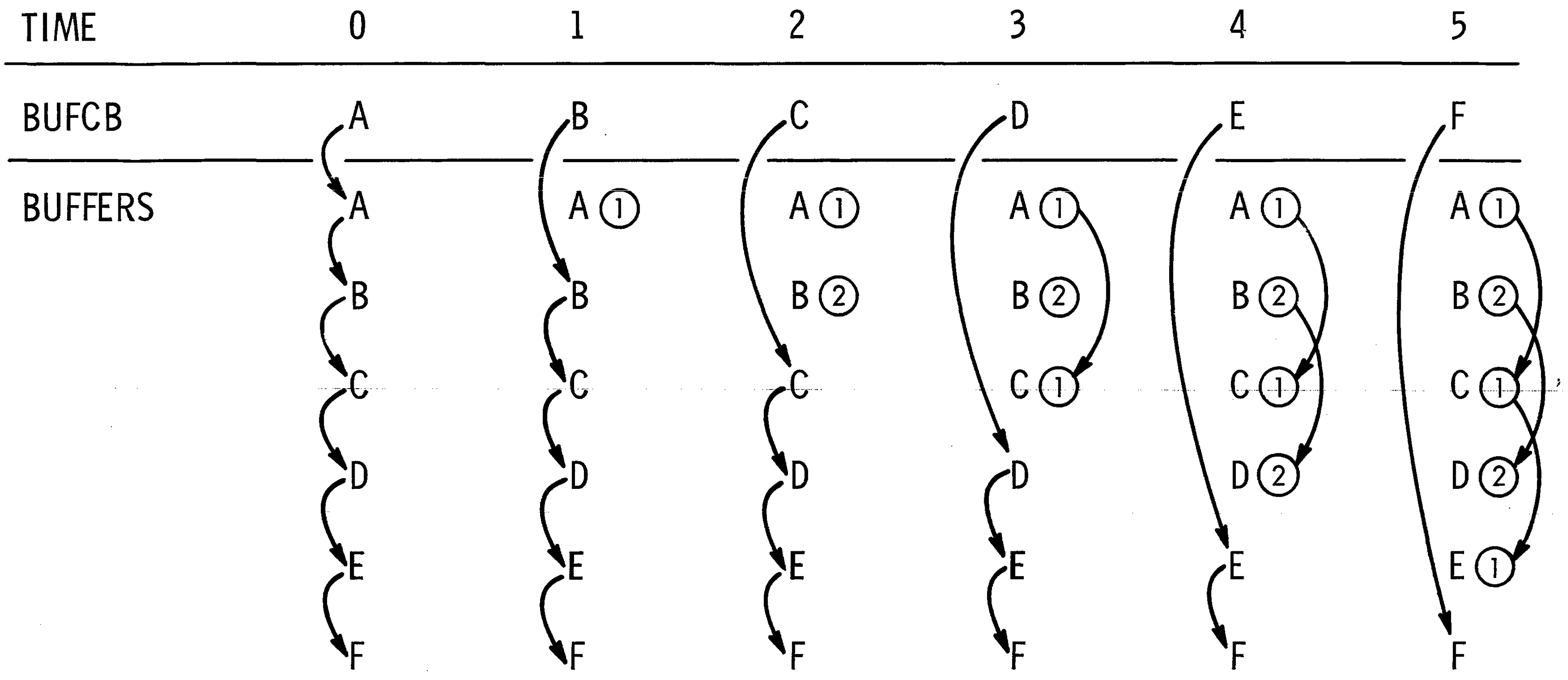
BUFFC ↑ BUFFD

BUFFER POOL

DECB/BUFFER CHAIN RELATIONSHIP



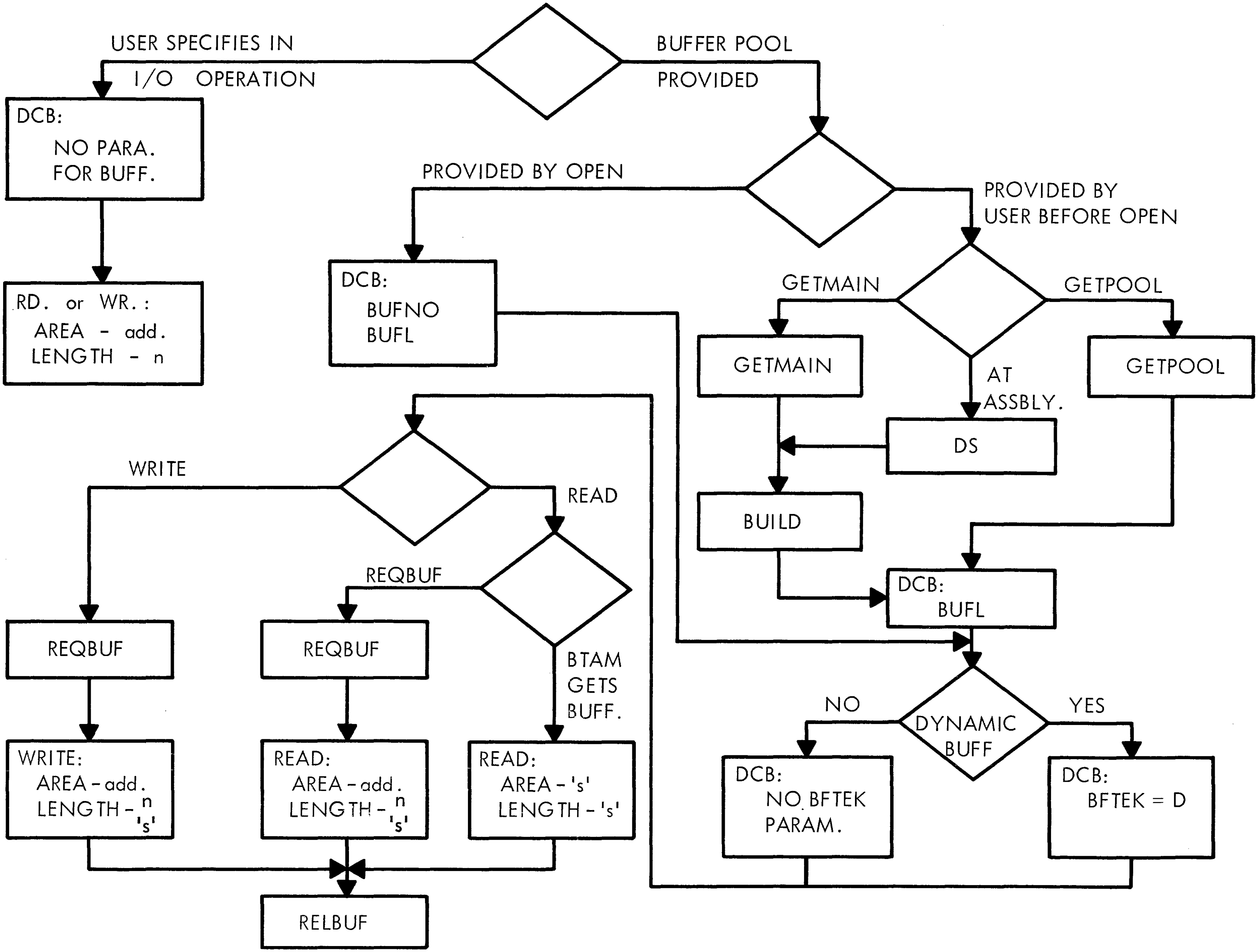
DYNAMIC BUFFERING EXAMPLE



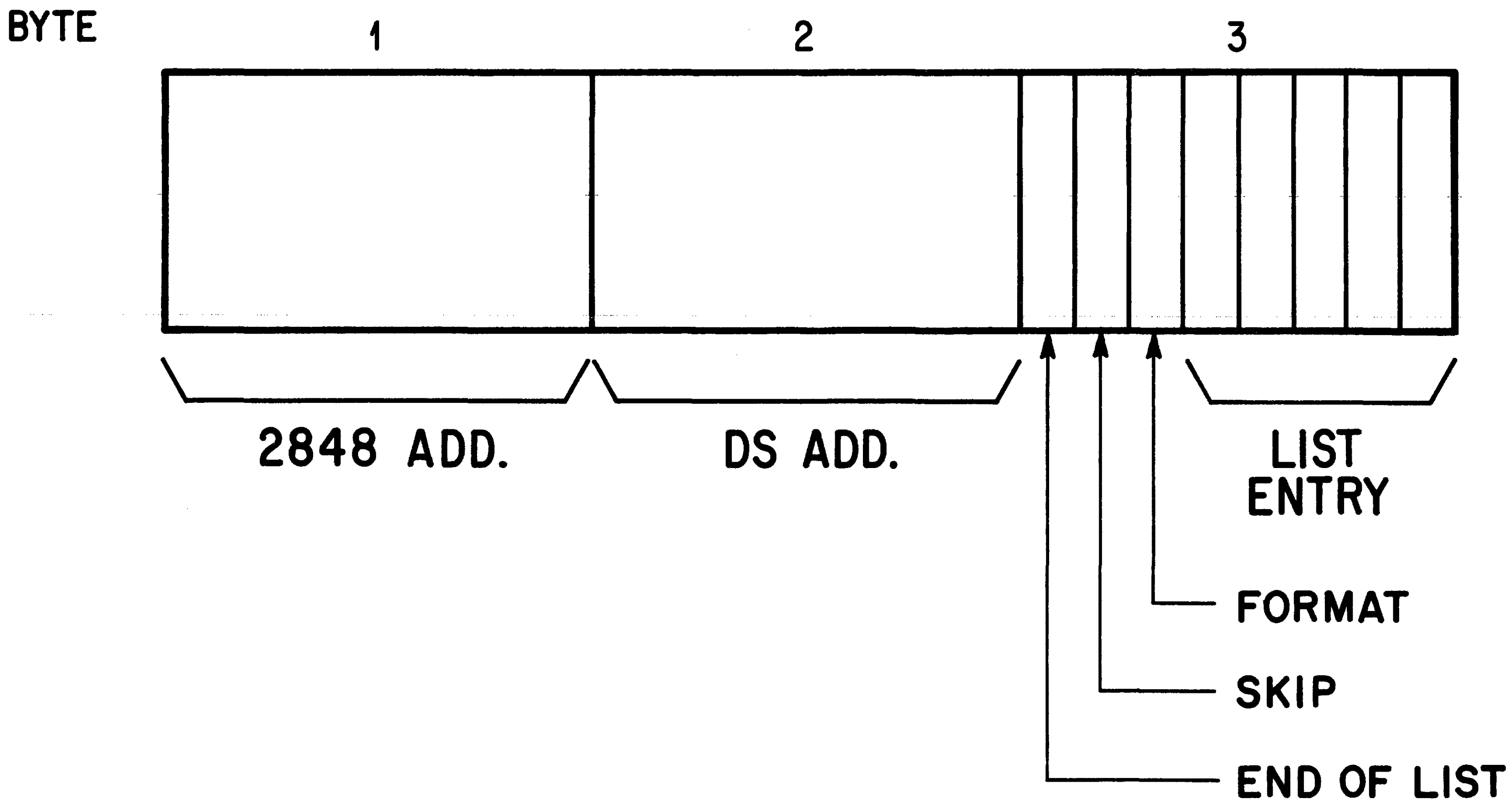
① Buffers for Line 1

② Buffers for Line 2

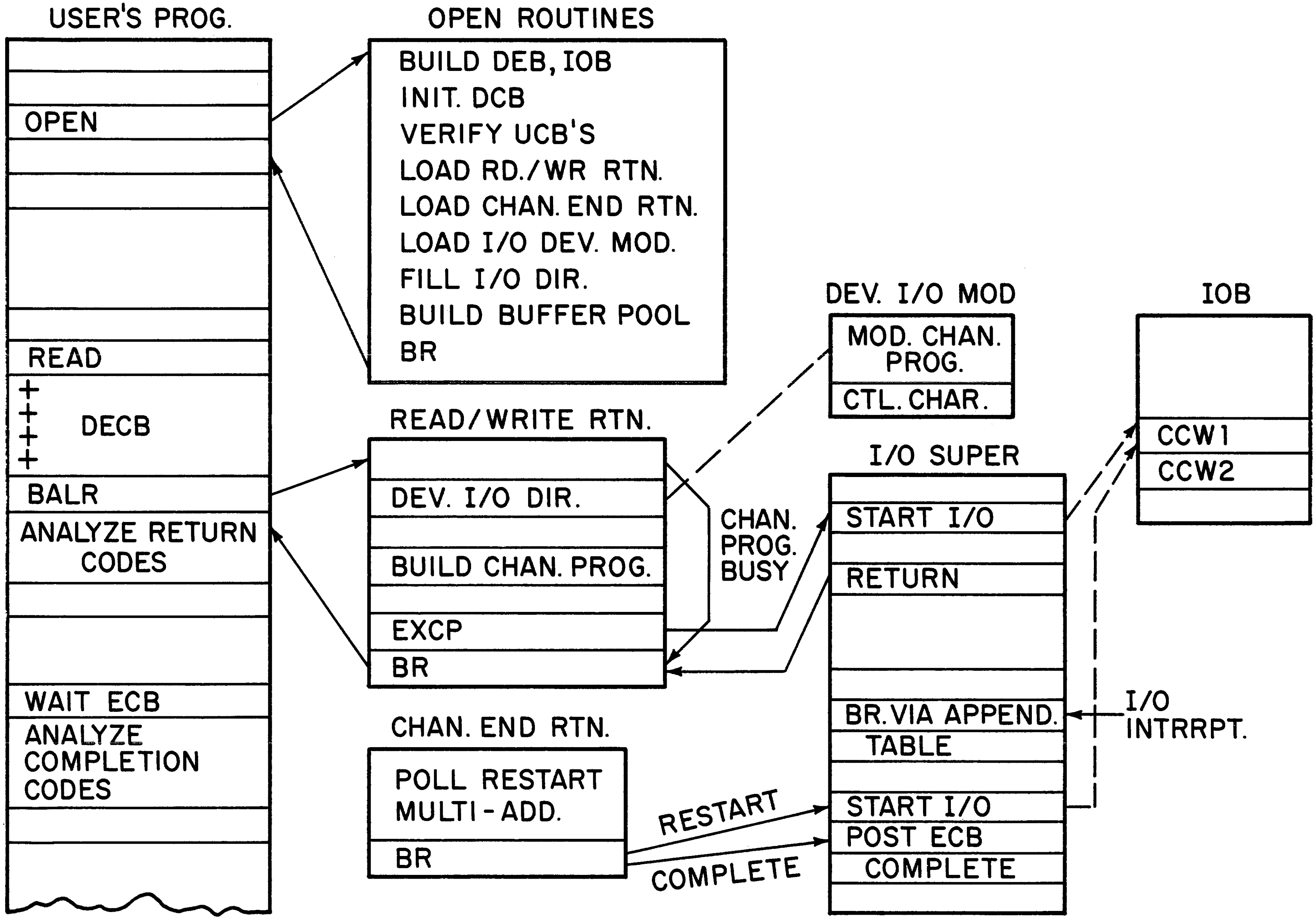
BTAM BUFFERING GUIDE



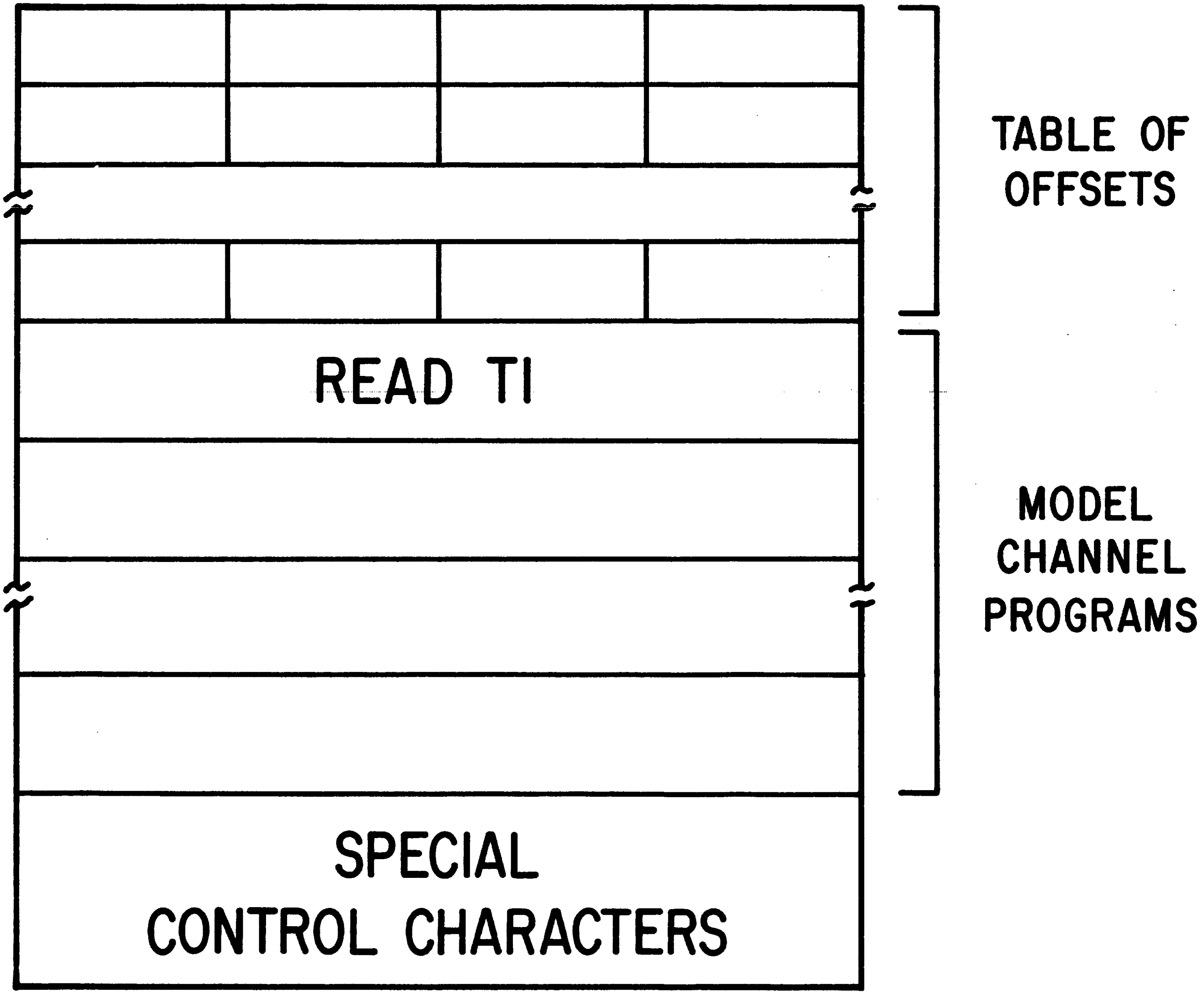
TRMLST POLLING OR ADDRESSING ENTRY



BASIC FACILITIES OF BTAM



DEVICE I/O MODULE



READ TI

SPECIAL CONTROL CHARACTERS

TABLE OF OFFSETS

MODEL CHANNEL PROGRAMS

DATA CONTROL BLOCK (DCB)

0			
20	NO. OF BUFF.	BUFF. CTRL. BLK. ADD.	
	BUFF. LNGTH.		DSORG
28		IOB ADD.	
	BFTEK	ERROPT	
36	IOB INDEX		

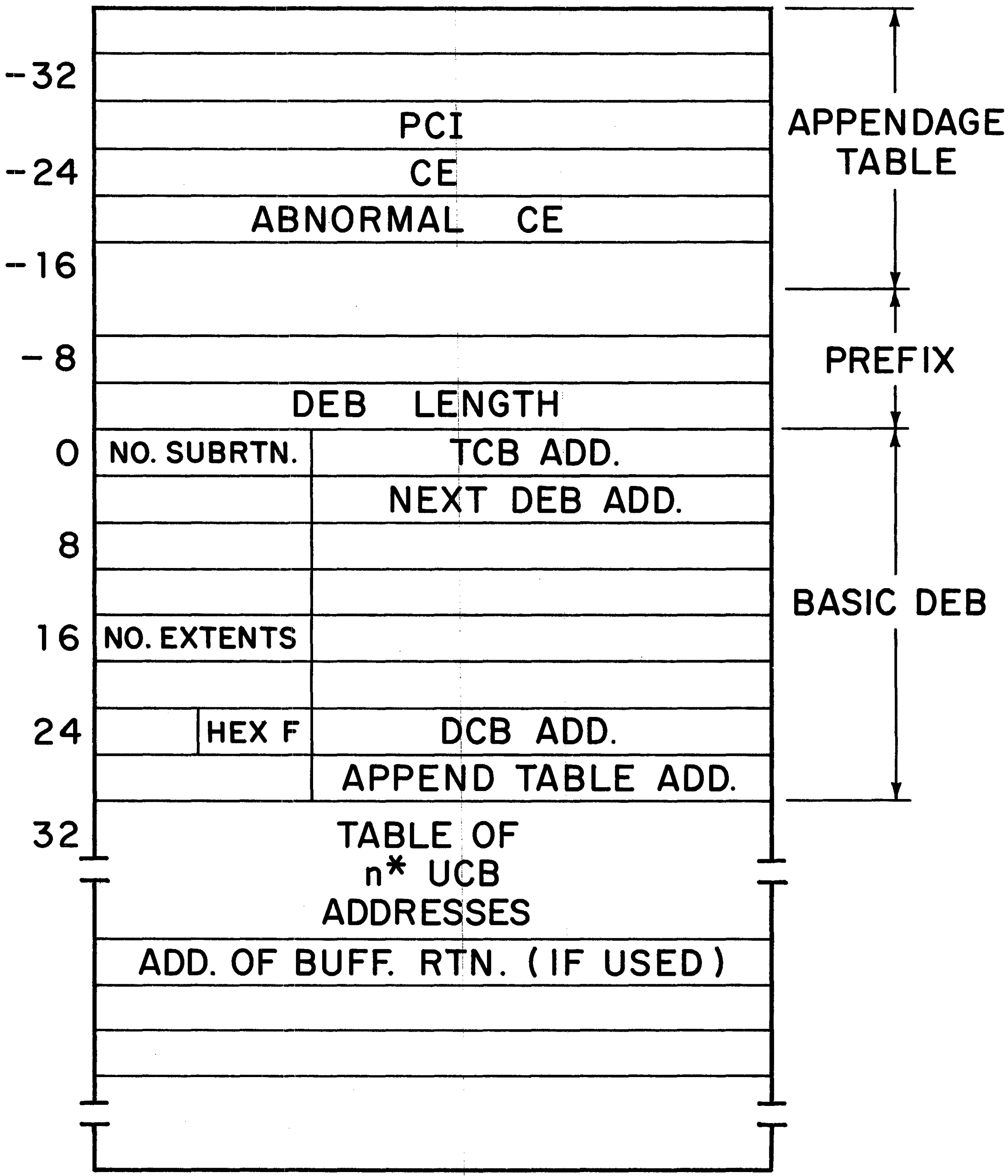
40	DDNAME		
48			

FOUNDATION BEFORE OPEN

40	TIOT		MACRF
		DEB ADD.	
48		RD.-WR. MOD. ADD.	
	LN. ERROR BLK. ADD.		
56			

FOUNDATION AFTER OPEN

DATA EXTENT BLOCK (DEB)

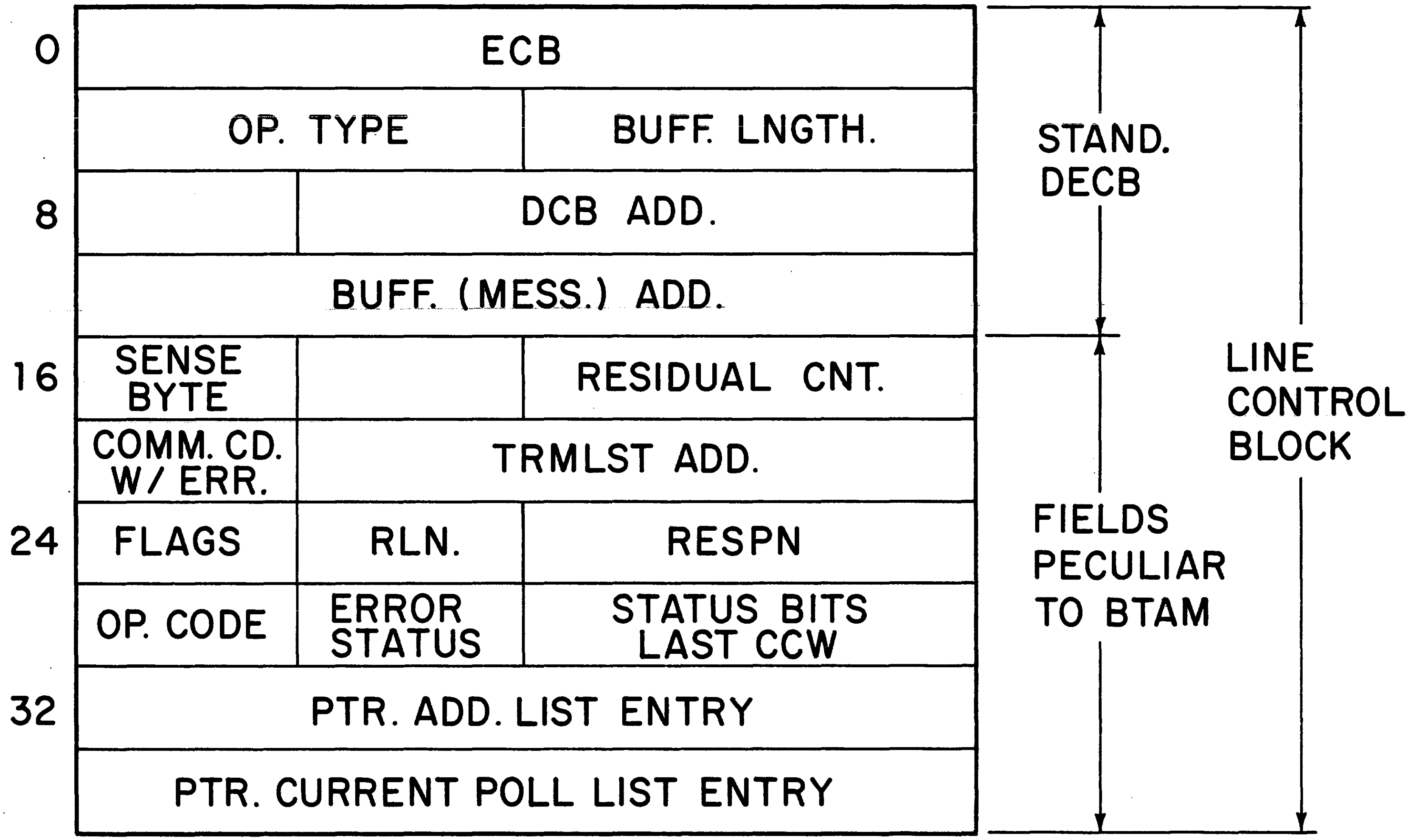


* Number of Extents (Lines)

UNIT CONTROL BLOCK (UCB)

<table>
<tr><td>0</td><td colspan="2"></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>CHAN ADD.</td><td>UNIT ADD.</td><td>FLAGS</td><td></td></tr>
<tr><td>8</td><td colspan="2"></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="2"></td><td></td><td colspan="2">UNIT NAME</td></tr>
<tr><td>16</td><td colspan="5">DEVICE TYPE</td></tr>
<tr><td></td><td colspan="3">LAST I/O REQ. PTR.</td><td colspan="2">SENSE BYTES</td></tr>
</table>

DATA EVENT CONTROL BLOCK (DECB)



0	ECB		
	OP. TYPE		BUFF. LNGTH.
8		DCB ADD.	
	BUFF. (MESS.) ADD.		
16	SENSE BYTE		RESIDUAL CNT.
	COMM. CD. W/ ERR.	TRMLST ADD.	
24	FLAGS	RLN.	RESPN
	OP. CODE	ERROR STATUS	STATUS BITS LAST CCW
32	PTR. ADD. LIST ENTRY		
	PTR. CURRENT POLL LIST ENTRY		

STAND. DECB

FIELDS PECULIAR TO BTAM

LINE CONTROL BLOCK

INPUT/OUTPUT BLOCK (IOB)

0	FLAG1	FLAG2	SENSE BYTE
	ECB ADD		
8		CSW	
16	CC	1ST. CCW ADD.	
		DCB ADD.	
24	CCW RESTART ADD.		
32	UCB INDEX	ERROR RTN. WORK AREA	
40	CCW AREA USED BY ERROR REC. RTN.		
48	CHANNEL PROGRAM		

DATA CONTROL BLOCK MACRO (DCB)

```
                                 ⎧ (R)   ⎫
[symbol]   DCB   DSORG=CX, MACRF=⎨ (W)   ⎬ [,DDNAME=symbol]
                                 ⎩ (R,W) ⎭
                 [,BUFNO=absexp][,BUFL=absexp][,BUFCB=relexp]
                 [,EXLST=relexp][,BFTEK=D][,LERB=relexp]
                 [,ERROPT=code]*
```

* Code can be:

1. Any combination of E,R,W,C,T, e.g., ERROPT=ERWCT
2. N
3. N,T

LINE ERROR BLOCK MACRO (LERB)

```
symbol      LERB      nlines-absexp,
                      [{[transmct-absexp],
                      [datack-absexp],
                      [intreq-absexp],
                      [notto-absexp],}....]
```

ASSEMBLE TRANSLATION TABLE MACRO (ASMTRTAB)

ASMTRTAB $\{$table code,$\}$....

DEFINE TERMINAL LIST MACRO
(DFTRMLST)

```
                                      ⎧ OPENLST  ⎫
[symbol]  DETRMLST  listtype-         ⎨ WRAPLST  ⎬ ,
                                      ⎪ DIALST   ⎪
                                      ⎩ IDLST    ⎭

                          (device dependent operands)
```

CHANGE TERMINAL ENTRY MACRO
(CHGNTRY)

[symbol] CHGNTRY $\left\{\begin{array}{c}\text{list-relexp}\\ \text{(r)}\end{array}\right\}$,

listype – $\left\{\begin{array}{l}\text{OPENLST}\\ \text{WRAPLST}\\ \text{DIALST}\end{array}\right\}$,

$\left\{\begin{array}{c}\text{position-absexp}\\ \text{(r)}\end{array}\right\}$,

$\left\{\begin{array}{c}\text{numcnars-absexp}\\ \text{(r)}\end{array}\right\}$,

action – $\left\{\begin{array}{l}\text{SKIP}\\ \text{ACTIVATE}\end{array}\right\}$

OPEN AND CLOSE MACROS

```
[symbol]   {OPEN }   {dcb-relexp,},...
           {CLOSE}
                     [     {L                          }]
                     [MF=  {     {lstnam-addx}         }]
                     [     {(E,  {    (r)    } )       }]
                     [     {     {    (1)    }         }]
```

LINE OPEN MACRO (LOPEN)

$$[\text{symbol}]\quad \text{LOPEN}\quad \left\{\begin{array}{ll}\text{decb} & \text{symbol}\\ & \text{(r)}\end{array}\right\}$$

READ AND WRITE MACROS
(LFORM)

```
[symbol]  {READ }   decb-symbol, optype-code,
          {WRITE}

                    [dcb-relexp, area-relexp,
                    length-absexp, entry-relexp,
                    rln-absexp,] MF=L
```

RESET POLLING OR RESET LINE MACRO (RESETPL)

```
[symbol]  RESETPL  ⎧decb-relexp⎫ ⎡,POLLING⎤
                   ⎨    (r)    ⎬ ⎣,ANSRING⎦
                   ⎩    (1)    ⎭
```

TRANSLATE MACRO (TRNSLATE)

$$[\text{symbol}]\quad \text{TRNSLATE}\quad \begin{bmatrix}\text{dcb-addx}\\ (\text{r})\\ (1)\end{bmatrix}, \begin{Bmatrix}\text{table-addx}\\ (\text{r})\end{Bmatrix},$$

$$\begin{Bmatrix}\text{area-addx}\\ (\text{r})\end{Bmatrix},$$

$$\begin{Bmatrix}\text{length-absexp}\\ (\text{r})\\ (0)\\ \text{'S'}\end{Bmatrix}$$

LINE ERROR PRINT MACRO
(LERPRT)

```
[symbol]    LERPRT    {dcb-symbol}
                      {    (r)   }
                      {    (1)   }

                      [,rln-absexp]
                      [    ,(r)   ]
                      [    ,(0)   ]

                      [,CLEAR=NO ]
                      [,CLEAR=YES]
```

REQUEST BUFFER MACRO
(REQBUF)

$$[\text{symbol}] \quad \text{REQBUF} \quad \left\{\begin{array}{c}\text{dcb-addx}\\ (\text{r})\\ (1)\end{array}\right\},(\text{r}),\left\{\begin{array}{c}\text{count-absexp}\\ (\text{r})\\ (0)\end{array}\right\}$$

RELEASE BUFFER MACRO (RELBUF)

$$[\text{symbol}] \quad \text{RELBUF} \quad \begin{Bmatrix} \text{dcb-addx} \\ (r) \\ (1) \end{Bmatrix}, (r)$$

TWAIT MACRO

```
[symbol]     TWAIT      (r), [TERMTST,]
                                 ⎧ addx ⎫
                        ECBLIST= ⎨ (r)  ⎬
                                 ⎩ (1)  ⎭
```

IBM 2260 Module for OS BTAM

Outline

A. INTRODUCTION

1. Remote Environment 2-V-1, 2-V-2
   a. 2260's connected to S/360 via a communication facility.

      2701 is the transmission control unit employed.

   b. Data rates
      1200 or 2400 bits per second.

2. Contents of module
   a. Hardware for remote 2260's

   b. Operation of remote system
      1) Line control
      2) Data flow
      3) Commands, responses, and channel programs
      4) Sample program
      5) Class problem

3. Bibliography

   A27-2700 Hardware Description of 2848 and 2260

B. 2260 HARDWARE FOR REMOTE OPERATION

1. IBM 2848 Display Control 2-V-3

a. Function - Control unit for 2260

1) Storage and Control Logic for Interface with: Communications Facilities - Data Set

Instead of Channel Adapter remote 2848 has one of two Data Set Adapters (depends on line speed).

These adapters are essentially functional entities; therefore, have little effect on basic 2848 when attached in lieu of Channel Adapter.

2) Character Generation
Provides character generation for all characters projected on 2260's CRT.

3) Buffer Storage

a) To maintain displays

b) To store keyboard information

c) To store printer information for a 1053 Model 4 printer which can be attached to the 2848.

4) Timing and Control Logic for Operations
Provides timing and control logic for the various 2260 operation.

INTRUCTOR'S NOTE:

Don't go into any detail on these points since they will be covered in more detail under data flow.

b. Three basic models of 2848 2-V-4

1) Model 1

a) Size of display
6 rows
40 characters per row

b) Number of 2260's

NOTE: There are two models of the 2260, Model 1 and 2. The only difference is the Type of CRT used.

° Model 2 of 2260 used with Model 1 of 2848

- Basic Model 1 of 2848 capable of operating two Display Adapters each servicing two 2260's.
- Expansion Capabilities of Model 1 of 2848

  A maximum of twenty-four 2260's

2) Model 2

   a) Size of display
   12 rows
   40 characters per row

   b) Number of 2260's

   - Model 2 of 2260's used with Model 2 of 2848.
   - Basic Model 2 of 2848 drives one Display adapter which services two 2260's.
   - Expansion Capabilities of Model 2 of 2848.

     A maximum of sixteen 2260's

3) Model 3

   a) Size of display
   12 rows
   80 characters per row

   b) Number of 2260's

   - Model 1 of 2260's used with Model 3 of 2848
   - Basic Model 3 of 2848 drives one Display Adapter which services two 2260's.
   - Expansion Capabilities of Model 3 of 2848.

     A maximum of eight 2260's

4) Data Set Adapters common to all basic 2848 Models. Choose one of the following:

   a) 9012 Data Set Adapter - 1200 Bits per second for remote applications.

   b) 9013 Data Set Adapter - 2400 Bits per second for remote applications.

c. Optional Features

1) Printer Feature

   a) Can be used by all keyboard 2260's associated with the 2848 and the System/360. Provides hard copy of display information.

   b) IBM 1053 Model 4 printer which attached to 2848 via the Printer Adapter. Adapter provides buffer and control for the printer

   c) Can be up to 2000 ft. from 2848.

2) Line Addressing Feature
Permits System/360 to select a particular line on the CRT of the 2260 to start displaying information.

3) Nondestructive Cursor and Nondestructive Cursor Adapter

If the Nodestructive Cursor is attached to a system, all display adapters must be provided with a Nondestructive Cursor Adapter.

d. Operator Controls and Indicators for 2848 2-V-5

1) Power On switch/indicator
Turns power on and resets 2848 controls.

2) Power Off switch
Removes Power.

3) Local/Remote switch
When in Local position the 2848 is logically disconnected from the channel. When in Remote position the 2848 is logically connected to the data set.

4) Data Set Ready Indicator
Indicates Power On/Off status of associated data set.

2. IBM 2260 Display Station

a. Two models of 2260 - only difference in CRT.

1) Model 1 used with model 3 of 2848.
Original 2260. Yellow phosphor. Regenerates 30 times per second.

2) Model 2 used with model 1 and 2 of 2848.
Longer life. Blue-white phosphor. Regenerates 60 times per second.

b. Standard Features

1) Operates without modification with any of the three models of 2848.

2) Anti-reflective display screen -- Minimizes glare.

3) Brightness Control -- Operation can adjust intensity.

4) Adjustable Character size.

Permits height and width of characters displayed to be adjusted to individual requirements.
Must be made by CE.

5) Adjustable character spacing.
Permits adjusting of spaces between characters.
Must be made by CE.

6) Total of 64 characters can be displayed on 2260. — 2-V-6, 2-V-7
26 alphabetic characters
10 numerical characters
25 special symbols
3 control symbols

NOTE: Character is 5 by 7 dot matrix.

c. Optional features

1) Alphanumeric Keyboard (4766) — 2-V-8
26 alphabetic characters
10 numerical characters
Special symbol keys
Control keys

2) Numeric Keyboard (4767) — 2-V-9
10 numerical characters
Special symbol keys
Control keys (same as with 4766)

NOTE: Attachment of numeric keyboard does not restrict the number of different characters that can be displayed on the 2260 by the System/360.

d. Operator Controls
Power On/Off switch and character brightness control. Control contained in a single rotary control located on the right side of each 2260.

e. Control Symbols — 2-V-10, 2-V-11

INSTRUCTOR'S NOTE:

Foil 2-V-11 is provided so instructor can show pictorially use of keys and control symbols.

1) Cursor - two types (hardware inserted).

a) General function of cursor.

° Visual marker

° Indicates display position on screen which next character entered will occupy and/or

° Marks end of message.

b) Destructive cursor - standard on all models.

- ° Appears as heavy horizontal bar (▬). e.g., J. SMITH ▬

- ° With various keys to be described cursor can be removed on the screen.

- ° Destructive cursor erases character contained in position to which cursor is advanced.

c) Nondestructive cursor -- optional.

- ° Replaces destructive cursor and appears as a vertical bar (▮) below the character line and immediately to the left of the next displayable position. e.g., J. SMIT▮

- ° Can be moved about the screen with various keys.

- ° Since the cursor does not occupy a displayable position, it can be moved freely about the screen without erasing other characters.

NOTE: Destructive cursor will still appear on screen as End of Message symbol. Covered under use of Enter Key.

2) Start Manual Input (Start MI) Symbol

a) Appears as ▶ on screen to indicate start of data to be transferred to the channel. e.g., Assume at top of screen ▶ J. SMITH

NOTE: When sent from S/360 it provides a means of telling operator where to key in data.

b) Entered by Start Key or sent from System/360.

3) New Line (NL) Symbol

a) Appears as a ◣ on the screen.

b) Prohibits the transfer of all data displayed between the NL Symbol and the end of the display line e.g., Assume.

J. SMITH◣	NAME
555555◣	MAN NO

c) Can be employed going to or from 2260.

4) Check Symbol

a) Appears as a ■ on the screen.

b) Used to indicate detection of a parity error in transfer of data from the System/360. Check symbol is displayed in place of the character in error, e.g., J. SM■TH

f. Control Keys

1) Basic considerations

a) On keys with 2 functions, that function below line is performed when shift key not depressed. Function above line is performed when shift key depressed at same time is control key.

b) Whenever destructive cursor is moved into a position where a character is displayed, that character is erased. With nondestructive cursor the character is not changed.

c) When key with only upper case function is depressed without depressing shift, a check symbol is entered into the cursor position.

2) Space/Erase
Advance

a) Destructive Cursor

° No Shift (i.e., Advance)
Blank in position of cursor
Advance cursor to next display position.
If end of line goes to start of next.
e.g., Before: AB▬C
After: AB▬

° Shift (i.e., Space/Erase)
Same as no shift

b) Nondestructive Cursor

° No shift
Advances cursor one display position.
No modification of display. e.g.
Before: AB▮C
After: ABC▮

° Shift
Erase position to right of cursor.
Advance cursor one display position. e.g.,
Before: ABC▮D
After: ABC▮

3) Backspace

a) Destructive Cursor

- No Shift
  Back cursor one display position. Erase character in that position. If cursor in first display position moves to last display position. e.g., Before: AB▬
  After: A▬

- Shift
  Same as no shift

b) Nondestructive Cursor

- No Shift
  Backspace cursor one position. No erasure. If in first position goes to last display position. e.g., Before: AB▮
  After: AB▮

- Shift
  Same as no shift.

4) Erase Display

a) Destructive Cursor

- Shift
  Erase entire display
  Keyboard restored
  Locates cursor in upper left hand corner of display (first displayable position)

b) Nondestructive Cursor
Same as destructive cursor

5) Start
Up

a) Destructive Cursor

- No Shift
  Blank in position of cursor
  Cursor moves one display position
  e.g., Before: AB▬
  After: AB▬

- Shift
  If no Start MI symbol displayed, Start MI Symbol placed in position of cursor and cursor advanced one position.
  e.g., Before: NAME▬
  After: NAME▶▬

If Start MI Symbol displayed, all displayed data between Start MI Symbol and Cursor, except data to right of a new line character, is erased. Cursor moves to first display position following Start MI Symbol.

```
e.g., Before: ▶ J.SMITH◣   NAME
                55▬
      After:▶▬          ◣   NAME
```

NOTE: This provides a good way to erase bad data which has been entered from the keyboard.

b) Nondestructive Cursor

° No Shift
Moves cursor up one line on display. If at top line goes to last, lateral position remains the same.

NOTE: Useful to move cursor rapidly for corrections.

° Shift
Same as shift with destructive cursor.

6) Enter

a) Destructive Cursor

° Shift
All data between the Start MI symbol and cursor (EOM in this case) transferred to computer (when computer issues read), except data between NL symbol and the end of line.

After successful transfer Start MI symbol is deleted. Cursor remains in its previous position. e.g.,

```
Before ▶ J. SMITH◢       NAME
         555555◢▬        MAN NO.
After    J. SMITH◢       NAME
         555555◢▬        MAN NO.

         read into computer
```

b) Nondestructive Cursor

° Shift
EOM symbol (▬) placed in; cursor position.
Cursor not advanced.
From that point proceeds same as destructive cursor.

7) Print - Used with print feature.

a) Destructive Cursor

° Shift
All data between first display position in first line and cursor, except data to right of NL symbols, printed.

If no print feature, blank in cursor position and move cursor one position.

NOTE: In this case Start MI symbol not required.

b) Nondestructive Cursor

° Shift
Same as destructive cursor except prior to data transfer an EOM (▬) is placed in cursor position.

8) <u>New Line</u>
Down

a) Destructive Cursor

° No Shift
Blank in cursor position
Cursor moves to next display position

° Shift
Place NL symbol in cursor position.
Cursor moves to first display position of next lower line. If already at bottom goes to top.

Data between NL symbol and end of line left undisturbed.
e.g. Before: J. SMITH▬ NAME
After: ▬ J. SMITH◢ NAME

b) Nondestructive Cursor

° No Shift
Cursor moves down one line
Lateral position maintained

NOTE: Useful for corrections

° Shift
Same as destructive cursor.

9) Single - Character Keys

a) Destructive Cursor

- ° No Shift
  Symbol displayed
  Cursor moved to next display position

- ° Shift
  Puts Check symbol in cursor position
  Cursor moves to next position

b) Nondestructive Cursor
Same as destructive cursor.

10) Double-Character Keys

a) Destructive Cursor

- ° No Shift
  Lower half character displayed
  Cursor moved to next display position

- ° Shift
  Upper half character displayed
  Cursor moved to next display position

b) Nondestructive Cursor
Same as destructive cursor.

g. Example of Keyboard Operations

1) Manual Input to be read by System/360.

a) Position cursor where message is to start.

b) Press shift key and depress Start key to enter MI symbol.

c) Key in message.

d) After keying in and verifying message press shift key and Enter key. All keys locked except Erase. Signals message is ready to be read.

e) If message successful Start MI symbol deleted and keyboard restored. If error Start MI symbol remains and keyboard locked. Erase will restore.

C. ASCII LINE CONTROL

1. Requirements

   a. Must be able to utilize the communications facilities for controlling the remote system and at the same time maintain line discipline.

   b. Since multi-drop environment, must be able to address a particular 2848.

   c. Once contacting a particular 2848 must be able to specify a particular 2260.

   d. Once contacting specific 2260 must be able to give it a command, e.g., read or write.

2. Start-Stop line control. 2-V-12

INSTRUCTOR'S NOTE:

It is suggested that these characters only be mentioned at this point and written on board for reference. Once written on board will use list to compare with 1050 start-stop control characters, e.g., Ⓒ, Ⓓ, Ⓝ, Ⓨ, Ⓑ, etc. The foil could be used if desired.

   a. Six ASCII communication control characters.

      1) STX - Start of Text.
      Preceedes a sequence of characters that is treated as an entity concluded with an ETX.

      2) ETX - End of Text
      Terminates a sequence of characters started with STX.

      3) ACK - Acknowledge
      Affirmative response from receiver to sender.

      4) NAK - Negative Acknowledge
      Negative response from receiver to sender.

      5) SOH - Start of Heading
      Used at beginning of a sequence of characters which constitute a machine-sensible address.

      6) EOT - End of Transmission
      Indicates conclusion of a transmission.

b. Two additional characters used for control purposes.

1) An ASC II control character
CAN - Cancel
Sent by 2848 following text and before ETX if data in text is in error. This error would have been detected by 2848 as it is read from the display buffer.

2) An ASC II format effector.
LF - Line Freed (New line)
Included as only part of the text. Causes New Line symbol to be displayed on 2260 and cursor moved to first display position of next line.

LF received by channel indicates following byte is first character of next 2260 display line.

c. Modes of operation

1) Remote 2848 operates in one of two modes:

a) Control mode
State of 2848 prior to receiving addressing or control data. Control mode established by SOH, EOT, and ETX control characters.

b) Text Mode
State of 2848 when receiving or transmitting text data. This mode is established only upon receipt of a STX control character.

NOTE: In multi-drop environment only one 2848 can be in text mode at a given time.

d. Control Status - At any time 2848 can be in one of the following statuses or a combination of them.

1) Non-Selected Status
The status the 2848 must be in prior to it being addressed by a command from the S/360. Established by SOH or EOT.

In other words a 2848 not indicated by an addressing sequence.

NOTE: Addressing sequence will be discussed shortly. It is a four byte sequence which is sent by the S/360 to the remote 2848 instructing it to perform some command.

2) Select Status
Once a remote 2848 is instructed to perform some operation, i.e., to receive or transmit data other than an addressing sequence or control data, it is in select status. In a multi-drop environment only one 2848 can be in select status.

3) Transmit Status
Once the 2848 has been instructed to transmit something other than ACK, NAK, or EOT it is considered to be in transmit status. It must also be in select status.

4) Receive Status
A 2848 not in transmit status is in receive status. Can also be in non-selected or selected status.

3. Addressing Sequence

a. Each command from the channel to the 2848 starts with a 4 byte Addressing Sequence.

1) 1st byte

a) Must be SOH or EOT

b) Establishes control mode

c) Puts 2848 in non-select status.

2) 2nd byte

a) Contains address of 2848.

b) 96 possible addresses.

3) 3rd byte

a) Contains address of the device

b) Addresses assigned in consecutive order with 1053 having highest position.

4) 4th byte

a) Command to be executed by the selected 2848, 2260 or 1053.

4. Similarity with other start-stop line control, i.e., Ⓒ, Ⓓ, Ⓑ, Ⓨ, Ⓝ.

a. The other line control is used for a different transmission code, i.e., 6 bit characters.

b. Transmission to the 2848 is via 7 bit ASC II code.

c. Nevertheless the following comparisons can be made:

ASC II (2260-2848)	6 bit BCD
ETX, SOH, EOT	Ⓒ
STX	Ⓓ
ACK	Ⓨ
NAK	Ⓝ
ETX	Ⓑ

NOTE: In the last case Ⓑ does not establish control mode whereas the ETX does. In 2260-2848 operation the message will be a maximum of one display station buffer of information. Therefore, this buffer is read or written with one block of data (message).

d. Addressing similarities

ASC II (2260-2848)	6 bit BCD
EOT	Ⓒ
2848 address	Station address
2260 address	Device address
Command	

Note that the main difference here is the line control for the transmission of 6 bit characters does not include a command. The 2848 receives a command to tell it what operation to perform.

D. DATA FLOW, ASSOCIATED HARDWARE, AND CODE SETS

1. Between 2848 and 2701 Transmission Control Unit

a. One of two data sets required

1) 202D or equivalent for transmission at 1200 baud

2) 201B or equivalent for transmission at 2400 baud

3) Half-duplex operation over four wire leased line

b. Transfer of data between 2701 and remote 2848 is in ASC II code (American Standard Code for Information Interchange). 2-V-13

1) 7 bit code

a) 128 possible characters

b) 71 and for data interchange

2) On communications line the 7 bits are accompanied by Start, VRC, and Stop bits, i.e., a 10 bit for mat

2. Between S/360 Channel and 2701

a. Transfer of data between S/360 channel and 2701 must be in an 8 bit code. Therefore,

1) 2701 provides translation between 7 bit ASC II code and a modified 8 bit version of ASC II (ASC II-8)

2) The ASC II - 8 code is formed by adding an x-bit between bits 5 and 6, i.e., 76 x 54321. This x-bit is the same as bit 7, e.g.
ASC II - 8 for A is 1010 0001
ASC II for A is 1000001

NOTE: In references foil the characters within the dotted line are ASCII-8 codes for lower case of alphabetic characters. These codes are converted to upper case by the 2848 if received during a write operation. End of message (▬) displayed as exclamation mark (!) on 1053 printer. Check symbol (■) displayed as quote (") on 1053 printer. Start MI (▶) displayed as cent sign (¢) on 1053 printer. New line symbol (◣) causes a carriage return and line feed on 1053 printer. 2-V-14

b. The 2701 requires a Type III Terminal adapter for the remote 2260-2848. Maximum of two Type III Adapter per 2701.

3. Remote 2848

NOTE: Reference back to 2-V-3 to indicate sections to be discussed.

a. Data Set Adapter of 2848 interfaces with data set.

1) Data Set Adapter 9011 interfaces with 202D or equivalent

2) Data Set Adapter 9012 interfaces with 201B or equivalent

b. 2848 Data Set Adapter 2-V-15

1) Write to 2848

a) Data received from communications facilities bit serially in serializer - deserializer register (SERDES register)

b) Once complete character formed in SERDES, the 7 data bits and parity transferred to character register

c) Outputs of character register sampled at ASCII decoder to determine if control character indicating start of message (SOH or EOT)

d) If SOH or EOT detected, next two bytes containing 2848 address and 2260 address are decoded to determine if for this 2848 and, if so, which is 2260.

e) Fourth byte routed to command decoder to determine operation to be performed. ACK response sent to channel.

f) Data is then received in SERDES and transferred to character register. Only text bytes used to update LRC accumulator.

g) Text bytes converted to 6 bit 2848 code (BCD) and transferred to serializer and into delay line buffer.

h) Data extracted from delay line deserialized and sent to common buffer in common control section of display control to be stored and displayed.

2) Read 2848

a) Data transferred to common buffer register to serializer in 7 bit ASCII code and into delay line buffer.

b) Almost immediately data extracted and loaded into output register and then to the character register.

c) Parity (VRC) is assigned and LRC updated.

d) Character transferred to SERDES register where start and stop bits are picked up prior to its transfer to communications facility.

c. Common control 2-V-16

1) Common buffer (7 bit register) acts as central point for data transfers between:

a) Data Set Adapter and 2260 buffer storage.

b) Data Set Adapter and printer buffer.

c) 2260 keyboard and 2260 buffer storage.

d) 2260 buffer storage and printer buffer.

2) Character generator and code translator

a) Consists of 64 core storage matrices, one for each of the 64 displayable characters.

b) Primary Function
Converts 6 bit BCD characters to 5 seven bit video bytes.

Convert video bytes to 6 bit BCD characters.

c) Other coding functions include converting:

Six-bit BCD code to 1053.
Tilt-and-Rotate code.
Six-bit BCD code to 7 bit ASCII code.

NOTE: The Common Buffer and Character Generator are part of the main control portion of the 2848 sometimes referred to as common control. Some of the other functions performed by this section are:

Checks for control characters (Start MI, NL, EOM)

Serializes data for writing into display buffers (delay lines)

Assigns parity to data stored in display buffers.
Checks parity data coming from display buffers.
Decodes address to select 2260.
Determines keyboard priority when more than one requires common controls.

Decodes keyboard commands.

Deserializes data, i.e., data received from delay lines is bit serially and must be assembled into bytes.

3) Display Adapter

a) Contains delay line buffer storage and control logic to service 2260's, e.g., read data from delay lines or write data into delay lines.

NOTE: These delay line buffers will be referred to as display station buffers.

b) Receives bits serially from common buffer. Continuously displayed and regenerated until erased or replaced.

NOTE: Position of bits in line determine portion of screen where cathode-ray beam will be intensified to cause dot.

Character is a group of dots.

c) Data stored in delay line can be read by computer.

4) Common Bus and MI Control

a) Accepts data and commands (e.g. Erase, Enter) generated from 2260 keyboard.

Data sent to common buffer.
Commands directed to control areas.

5) Printer Adapter

a) Buffers data to be printed. Converted to tilt-rotate prior to entry in buffer.

b) Controls print operation.

E. COMMANDS, SEQUENCES AND RESPONSES, AND CHANNEL PROGRAMS

1. Sequences and Responses.

a. Once the command is given to the 2848 in the addressing sequence it is necessary for the S/360 to control and monitor the operation by exchanges between the channel and remote 2848. These exchanges are called sequences and responses and are performed by the channel programs.

b. These sequences and responses will vary from application to application.

2. Eight commands can be executed.

NOTE: The following paragraphs discuss various commands, responses and sequences, and the channel programs necessary to perform the various. Only the more likely sequences are covered. Foils 2-V-27 to 2-V-30 are provided for reference concerning other possible sequences for some of the more common operations. Some operations show SOH being utilized. BTAM does not use this character since in effect it is the same as the EOT.

The commands to the remote 2848 perform the same operations as the commands for local operation.

a. Specific Poll to a 2260 Display Station 2-V-17

1) Purpose
Tests for presence of manually entered message awaiting transfer to channel. Causes message to be transferred to S/360.

NOTE: Message ready for transfer if Enter Key has been depressed and Start symbol is displayed.

2) Operation
Data from Start symbol to cursor, excluding that between an NL symbol and the end of the line, transferred.

3) Sequence and Response and Channel Programs

NOTE: TABLE in channel programs reference Device I/O module. TRMLST in channel programs reference terminal lists defined by user's DFTRMLST macro.

CD and CC refer to data chaining and command chaining. Whenever CD or CC is specified there will be no interrupt as long as the operation ends normally, i.e., no Unit Check or Unit Exception occurs.

a) Read Initial - performs following

° Sends addressing sequence to remote, i.e., EOT's to establish control mode, Terminal (2848) and Device (2260) addresses, Command for 2848 to execute.

NOTE: 2260 addresses depicted in foil 2-V-18, 2-V-19 2-V-18 are typical and may vary from installation to installation. Check with CE. 2848 addresses also vary.

° If 2260 has message, it is sent (STX establishes text mode) and read into core buffer. First Read reads only 2 characters but chains to second Read which completes filling of buffer.

NOTE: If no Start symbol would not receive text, only STX, ETX, and LRC

If CAN is generated, it indicates error detected in reading display buffer. Will cause CE, DE, and UC with Equipment Check in sense byte.

- ° If 2260 hasn't a message (i.e., Enter key hasn't been depressed), will receive EOT causing CE, DE, and UE. In this case first read would not chain to second read.

- ° If no response will cause time out resulting in Channel End, Device End, Unit Check, and Time Out in sense byte.

NOTE: No response can be caused by:

- ° Illegal command
- ° Illegal 2848 address
- ° Illegal 2260 address or printer address
- ° Parity error in addressing sequence
- ° Parity error in line control character
- ° Defective communication facility

b) Channel replies

- ° Read Repeat
  If transmission in error user would issue Read TP to send NAK and cause 2848 to retransmit message.

- ° Write Positive Acknowledgement
  Transmits STX and EOT's to 2848 to indicate good transmission and put system back into control mode.

  Read Initial with reset would have added a 6th step to the Read Initial channel program which would have done the same thing.

- ° Other possible replies but less likely to be used would be:
  Read Continues which would send ACK and ask **2848** for next block from the 2260. However, 2260 could not be expected to have another block yet and 2848 would send EOT's.

  Write Negative Acknowledgement
  Indicates bad transmission but doesn't ask for retransmission.

b. Specific Poll to the 1053 Printer 2-V-20

1) Purpose
Causes 2848 to test readiness and busy status of printer

2) Sequence and Response

a) Read Initial
Basically operation is same as poll of specific 2260. However, if printer is available message received will only be printer address. Printer is reserved at remote site until channel responds with message.

NOTE: Command is same but address of device indicates printer is being polled.

If printer not ready will receive NAK from 2848 which sets printer request in control unit.

If printer busy receive EOT and printer request set.

NOTE: Printer request indicates the channel has a message for the printer when it becomes available.

b) Write Continue
If the printer is available, the user would respond at this point with a Write Continue to transmit the message.

The channel will then read the response into an area of the DECB.

c) Write Negative Acknowledgement
If the message was received correctly and there is no more to print, the user would want to put the system back into control mode. This could have been done with a Write Continue with reset or by issuing a Write Negative Acknowledgement macro.

NOTE: Write Negative Acknowledgement does not have to mean message was received incorrectly. Can be used to put system back into control mode in some cases.

If the message was received in error, the Write Continue could be reissued.

If there is another block to send, another Write Continue could be issued.

c. General Poll 2-V-21

1) Operation

a) Causes 2848 to test each 2260 for presence of manually entered message pending transfer.

b) If printer attached determines Printer Request status to see if printer has been requested previously.

c) 2848 checks printer first and then Enter key circuit of each 2260.

2) Sequence and Response

a) Read Initial
Basically same as other Read Initials Except device address is a hex FF.

If printer request set will receive STX, device address, and ETX but no text.

If printer has not been requested and 2260 has something to send, will receive the text, i.e., all data between Start symbol and EOM.

If no message receive EOT.

No response - time out

b) Write Continue
If message from printer would issue Write Continue to send message to printer.

c) Read Continue
Once a Read Initial using a general poll is initiated, the general poll continues to remain in effect until all buffers have been checked or until the channel issues an STX or EOT, e.g., STX is issued when writing the printer.

The user can keep reading stations by issuing a Read Continue. When no more stations have messages or the last one has been checked, the 2848 sends an EOT which terminates the operation.

d) Other responses

Read Repeat - if message is received in error a Read TP can be issued to cause retransmission of message.

Reset - at any point in time Read Initial or Read Continue can be issued with reset to terminate the operation.

The same effect can be obtained by issuing a Write Positive Acknowledgement.

d. Read Full DS Buffer 2-V-22

1) Read Buffer
Mostly for special purpose, e.g., diagnostics.

All data in 2260 buffer transferred to channel.

2) Reset
Would follow up with Write Positive Acknowledgement or could have used reset on the Read Buffer.

e. Write Addressed DS or 1053 2-V-23

1) Operation
To transfer data from channel for display on selected 2260 screen.

2) Sequences and Responses

a) Write Initial

° Addressing sequence to remote, i.e., EOT's, Terminal and Device addresses, and Write command.

° Response read into area in DECB.
ACK - printer or 2260 ready to receive.
Time out - remote not on
NAK or EOT - indicate printer not ready or busy. Printer request would be set.

° On receiving ACK channel sends message preceded by STX

° Remote responds with ACK if received correctly.
NAK would indicate Data Check EOT would indicate loss data.

b) Write Negative Aknowledgement
Once the remote has received the message correctly the channel in most cases will send an EOT to put system back into control mode. The 2260 would not be ready to receive another block. The Write Initial could have been issued with reset to achieve the same result.

c) Write Continue
Only the printer would be ready to receive a second block and that could be done with a Write Continue. Eventually a Write TN or Write with reset would be issued.

Note: With 1053 could have 2848 indicate buffer overrun. Printer buffer holds 1223 characters. If this is exceeded, a buffer overrun condition occurs and print operation is terminated.

f. Erase/Write and Line Address Write

1) Operation

a) Erase/Write
Essentially an Erase command combined with a Write. Screen is erased and message is written starting in first position of buffer.

b) Write DS Line Address
Permits selection of a particular line as beginning of message.

2) Sequences and Responses 2-V-23

a) Write Erase
Sequence of commands same as Write Initial. Only difference is Command sent to remote will cause erase and then write.
Since this is only applicable with 2260 will issue Write with reset or follow up with Write Negative Acknowledgement.

b. Write at Line Address
Sequence of commands same as Write Initial. Only difference is Command sent to remote will cause write to start at a particular line.
In text itself the first 2-V-24
character will be the line at which to start.

3. Effect of Dynamic Buffering

a. Each read and write CCW causes a PCI (Program controlled interrupt).

b. The PCI appendage receives control and sets up buffer addresses and alters existing CCW's.

NOTE: Normally the PCI occurs within one character time after CCW is fetched. In this manner, while the rest of the buffer is filling or emptying there is time to set up the CCW's for the next buffer. However, since the time of this interrupt is still somewhat unpredictable and the amount of data being transferred may be small, precautions are taken in case the next buffer is not set up in time.

c. A chain of 4 CCW's perform the read and write.

1) Read 2-V-25
Two read commands in loop are followed by read with PCI and SKIP flags on. If PCI fails to occur in time, read skip is performed to read data from line but not into memory. At termination retransmission is requested.

2) Write 2-V-26
Preceding condition easier to handle for write. Invalid TIC is issued causing program check which causes abnormal channel end appendage to be entered. Abnormal channel end then gets next buffer address and restarts on next write command.

d. This sequence of CCW's would replace the CCW indicated in the previously described channel programs which reads or writes the message into or out of an INAREA and OUTAREA.

F. PROGRAMMING CONSIDERATIONS AND SAMPLE PROGRAM

INSTRUCTOR'S NOTE:

The following discussion relates to the programming example included in the appendix. This program doesn't cover all possibilities but does provide a basis for various programming considerations.

1. Purpose of Program

   a. Remote 2260's or remote 1050

      1) Can be operated with two remote 2260's on the same communication line.

         a) Line 1

         b) 2848 control unit with ASCII address of hex. A0

         c) 2260's with ASCII addresses of hex. A0 and A1

      2) Can be operated with 1 remote 1050

         a) Line 1

         b) Nonswitched

         c) 1050 with 1050 coded address of hex. 62 (A)

         d) Write to printer with 1050 coded address of hex.02 (1)

         e) Read from keyboard with 1050 coded address of hex.OB (5)

      NOTE: Program will not operate both lines at once. User will specify from console typewriter on System/360 which device(s) are on the system. The console message will state:
      "CHOOSE 1-2260 2-1050 3-BOTH"
      Third choice is for expansion,

      Reply: REPLY 00,'1' or REPLY 00,'2'

b. Gives three choices of operation to user on terminal

1) By entering a 1 he can display a file in core called COREFILE.

2) By entering a 2 followed by a message he can change the data in COREFILE. A new opening message will be sent to the terminal once the new message is received.

3) By entering a 3 he can terminate the job.

NOTE: On the 2260, the user will enter a START MI symbol followed immediately by the number selected. ENTER key is then depressed.

When using the 2260 if the user requests a display of COREFILE, he must terminate the display be entering START MI and the number 4. A new opening message will then be returned. On the 1050, the user will enter the selected number immediately following the opening message. The whole message should be entered before depressing the EOB key. The user should not enter an EOT.

The user when entering a new message into COREFILE should follow each carriage return with at least 3 blanks. In this way, when the user requests display of this message it should type out properly, i.e., allow for time on carriage **return on 1050**

2. Programming Considerations

a. Job Control Language (JCL)

1) The JCL will vary depending on the procedures employed by the installation.

2) TELCMLIB contains certain BTAM routines, therefore, at Link Edit time, the user must on the //SYSLIB DD card specify DSNAME=SYS1.TELCMLIB.

3) DD cards for units
Be sure at execute time to include the DD cards specifying the line(s) for the line group(s). In the DCB the program specifies DSNAME=DS2260.

b. Special Considerations
This program was written and checked out on a pre-release system of OS release 11. At that time, two compensations had to be made for problems in BTAM OPEN and TRNSLATE macros.

1) BUFL is stored by program in DCB. User need only specify this parameter in DCB at assembly time.

2) The user should not enter a block of data exceeding 256 bytes, because the TRNSLATE macro will only translate blocks of less than 256 bytes.

NOTE: At the time this outline was written APAR's exist for both of these problems.

c. Dummy Section (DSECT)
DSECT's available for both DCB and DECB. This provides a simplified method for the user to address fields in the DCB and DECB. The address of the DCB and DECB is designated in USING statements. Whenever one of the fields in the DSECT's are references, the base register will be the DECB or DCB address and the displacement will be that indicated by the DSECT. The user is responsible for loading the base register with the proper address prior to referencing one of the fields in the DSECT.

d. Buffer Pool

1) For READ's
The sample program has utilized **GETMAIN followed** by BUILD to construct a buffer pool. Note the program stores the address of the pool in the DCB. If other DCB's and line groups existed, the buffer pool address should also be stored in those DCB's. In this manner, the pool can be shared.

On READ's dynamic buffering is valuable since the data blocks received from the remote 2260's may vary from 1 to 960 characters. Frequently the blocks transferred will be short. However, the program must be able to handle the larger blocks when they are entered.

2) For WRITE's
All WRITE's in this program are of areas in core, i.e., COREFILE and opening message. However, these areas will still be treated by BTAM routines as dynamic buffers since that is specified by the DCB. Therefore, care must be taken to assure they look like dynamic buffers to WRITE and TRNSLATE macros by placing 0's in the first word where link address would be found.

Also, be sure to start buffers on a full word boundary.

Be sure the buffer counts include 4 bytes which must must be set aside for the link address.

e. Addressing and Polling Lists

1) Addressing List
This program has an address list for each terminal, including the printer although it is not used at this time.

2) Polling Lists

a) For 2260's
For the 2260's, the polling list indicates a general poll which will permit acceptance of message from any station requesting service.

b) For 1050
For the 1050, the keyboard is polled.

f. READ's and WRITE's

1) READ's
READ Initial is employed to read messages from the remote terminals. For area and length operands 'S' parameter is indicated. Therefore, length and area will be obtained from BUFCB and DCB. Since dynamic buffering is indicated, the data will fill as many 48 character buffers as required to contain the message.

Could have READ Initial with reset or use WRITE Positive Acknowledgement to return positive response and deselect characters on a good transmission.

The sample program chose the latter. Also a READ Continue would have issued a positive response and read the next message if the other 2260 required servicing. However, in this program the writer did not wish to receive a message from another 2260 before the previous one was processed.

2) WRITE's
All WRITE's were of data from areas assembled in the user's program. Therefore, did not have to request buffers. However, since dynamic buffering was specified in DCB, must be careful to have output areas adhere to dynamic buffering formats.

Counts and area locations are furnished in the Writes.

Note that with the 2260, the WRITE Erase is used since one normally wishes to erase the screen and position the cursor prior to writing out a message. The write also indicates reset to deselect the terminal. This should not be necessary since the next instruction will be a READ Initial which also transmits deselect characters. In fact, the WRITE Initial for the 1050 in this program does not use reset.

g. Sequence of I/O Operations
The sequence of I/O operations, e.g., when to use READ Continue after READ Initial, or when to use WRITE Continue after WRITE and READ Initial, or when to reset, are decisions dependent on the application. The user must be constantly aware of the total sequence of transmissions and response which must be made and be sure that no terminal is left hanging that will result in timeouts or other abnormal completions. Therefore, it is highly recommended that as the program is written he constantly refers to the charts showing the sequence of transmission and responses.

Also the programmer will find constant reference will have to be made to the channel programs to be sure just which of the required operations each channel program is performing.

h. I/O Return Codes and Completion Codes

1) Return Codes
Following READ's, WRITE's, REQBUF, and RELBUF return codes are passed back to the user in register 15.

With READ's and WRITE's these return codes indicate whether the I/O operation was started.

With RELBUF and REQBUF, the codes indicate whether the buffers were released or acquired as requested.

In both cases, it is the user's responsibility to be sure that these codes are checked. In the case of the sample program, all these codes are indicated, but a NOP is used where a branch to some subroutine would be required to take some action. In these situations the sample program terminates, following a message to the operator.

NOTE: When using the technique illustrated in the program to get to the proper routine, the user must be sure each branch is 4 bytes away from the start of the last. In the example the 1st NOP is not 4 bytes beyond the start of the BR instruction but instead 2 bytes beyond. Therefore, a filler of 1 half word should be inserted following the BR and before the 1st. NOP.

2) Completion codes
After an I/O Operation is completed the user must check the completion code placed in the ECB to determine whether it completed normally. The user should provide a line analysis routine to check completion, e.g., to examine the flags in the DECB.

The sample program indicates the completion codes which can be received but does not go into a line analysis routine.

3) Completion codes on dynamic buffering
When dynamic buffering is employed, completion codes are placed in the first character of each buffer when it is filled.

The user can check these individual ECB's for each buffer or wait on the primary ECB in the DECB. By waiting on the buffers' ECB's, it may be possible to save time by processing a buffer while another is being filled. However, the user must realize that the buffers may be processed as filled, but the LRC will not be checked until End of Block (ETX). At this time, if LRC's do not compare, bad data has been processed by processing buffers prematurely.

i. Polling considerations

1) What ends polling?

a) Positive response is received from the terminal

b) An error condition is detected

c) RESETPL macro

d) End of an open list reached

NOTE: With wrap list polling starts again at beginning of list.

2) Operation posted complete when error, RESETPL, end of OPENLST or transmission is completed.

3) Must issue another READ Initial to restart polling.

4) General Poll for 2260
Essentially this works in the same manner as polling for other terminals, except that one need not have an entry in the list for each 2260. The presence of a general poll will result in accepting a message from any 2260 which requests service.

A wrap list is still employed, which simply says the 2848 will continue to check all 2260's until one requests service.

j. Translate

1) Following READ Initial
In the sample program once the READ Initial is completed all the buffers containing the input are translated.

The length parameter is passed in LNGTHREG to the TRNSLATE macro. In this case a length of 0 is used to indicate dynamic buffering.

Since dynamic buffering is used, the macro will proceed through all buffers using the link address. Translation stops when the TRNSLATE finds 0's in the link address of a buffer.

2) Translates preceding WRITE Initial's

In all these cases the data is being transmitted out of a buffer specified at assembly. However, the TRNSLATE macro checks the DCB to see if dynamic buffering is employed. Since it is, the output buffers must also have a link address as the first 4 bytes. Therefore, the sample program is careful to include a full word of 0's at the start of each of these buffers.

k. Terminal Table
The SRL for BTAM makes reference to user terminal tables. In an application with a number of lines, the user will find it useful to have a table to contain information concerning each of the lines, e.g.,

- To indicate which terminals are on line, i.e., terminal status.
- Pointers to various addressing lists for the terminals
- Any other control information the user wishes to keep on a line and its terminals

1. ERROPT in DCB
   The sample program shows ERROPT=ERWC. In this program the R and W are meaningless since text errors on read's and write's will not be retired when dynamic buffering is employed.

G. CLASS PROBLEM

Problem 1

Your company is installing a TP system with a single line with a 2260.

Read a message not to exceed 48 characters from the 2260 into a core buffer 48 characters in length.

Let BTAM supply buffers at OPEN.

Translate message to EBCDIC.

Translate message back to ASCII code.

Send message back to 2260.

Use ERP and LERB.

Check for normal return codes and completion codes, but do not get involved with error analysis.

Problem 2

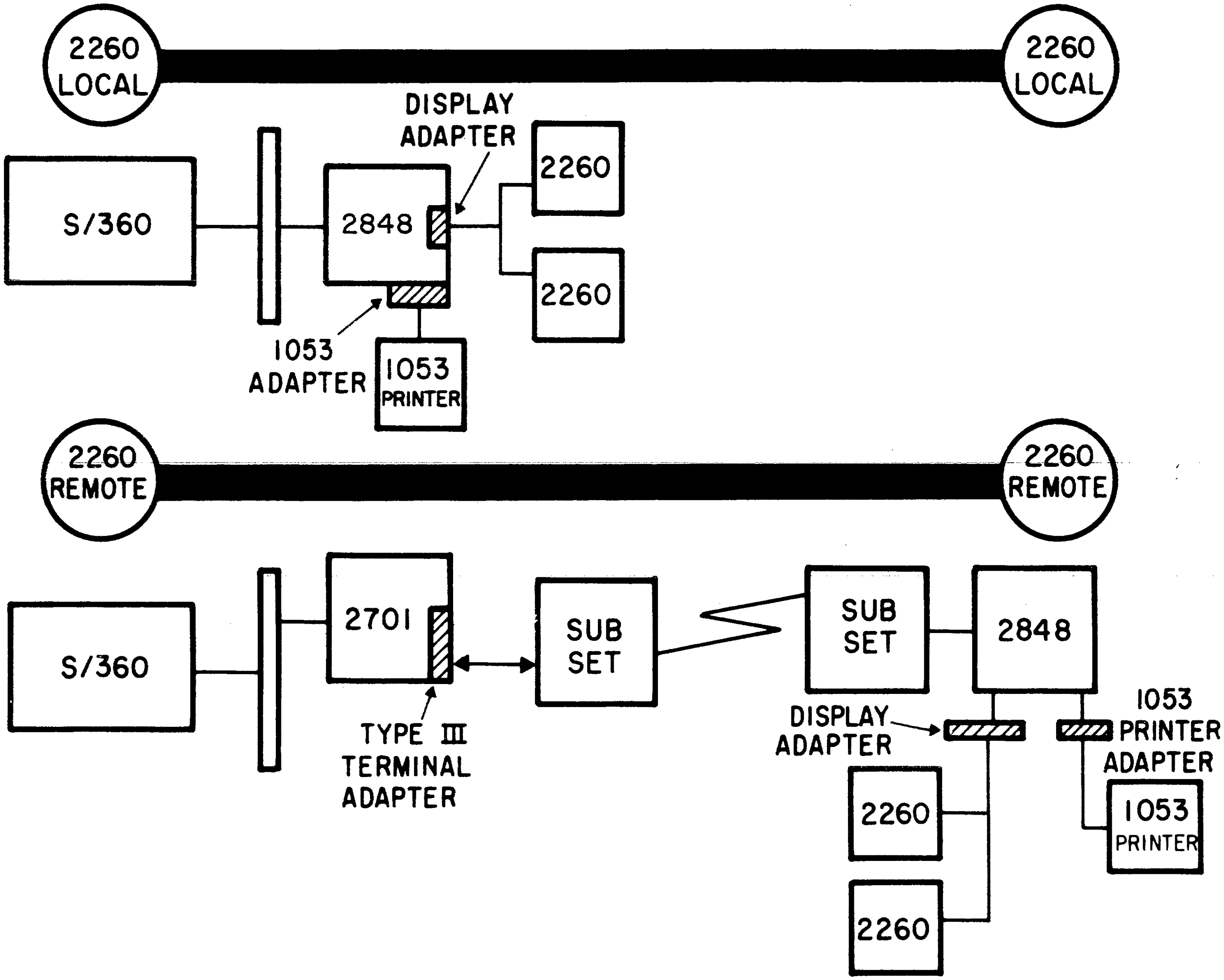
Your company has order a second 2260. In addition you find the message blocks will vary from 1 to 960 characters in length.

Still utilizing 48 character buffers, modify your program to handle message blocks up to 960 characters in length.

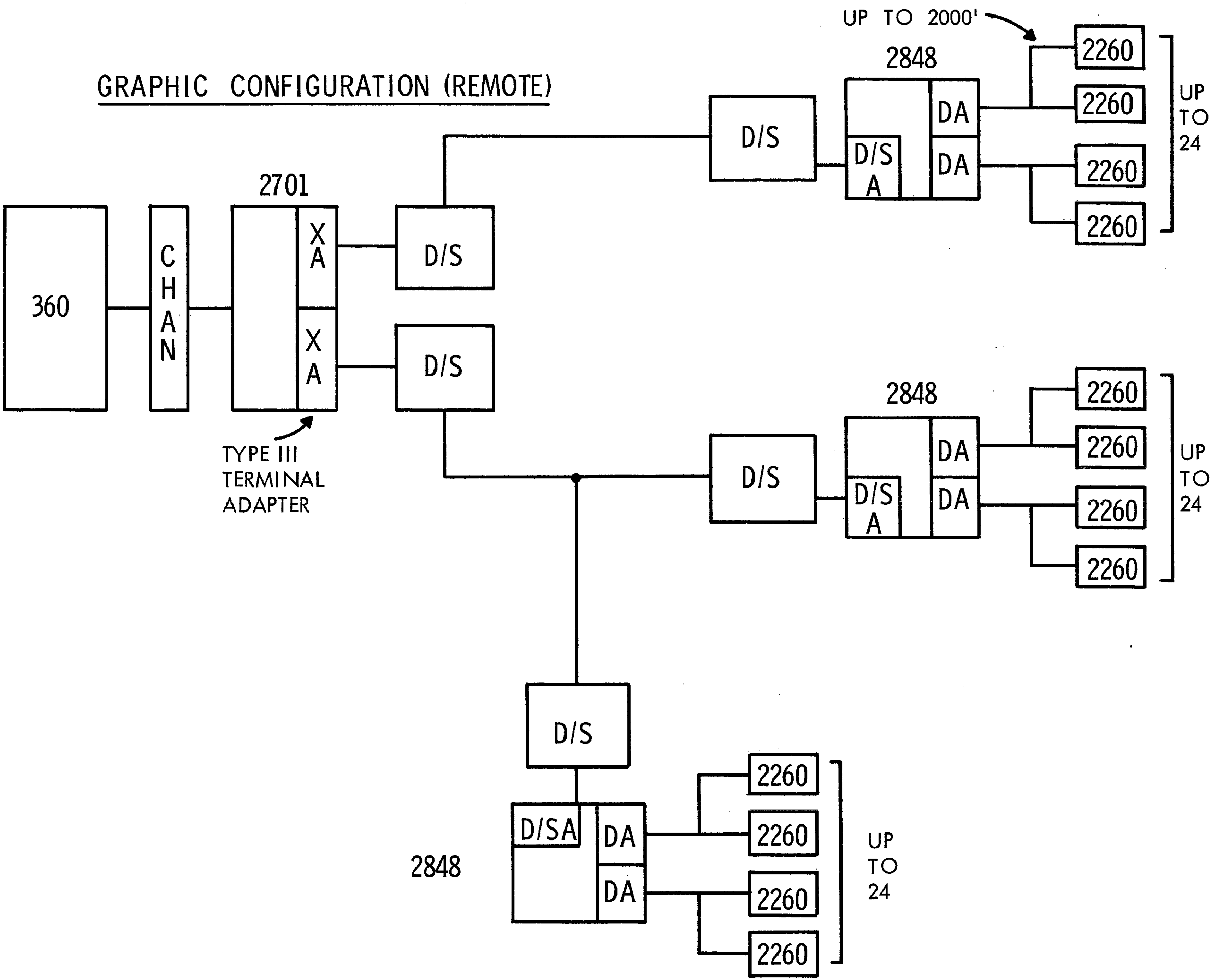
Also include on-line terminal test in your system.

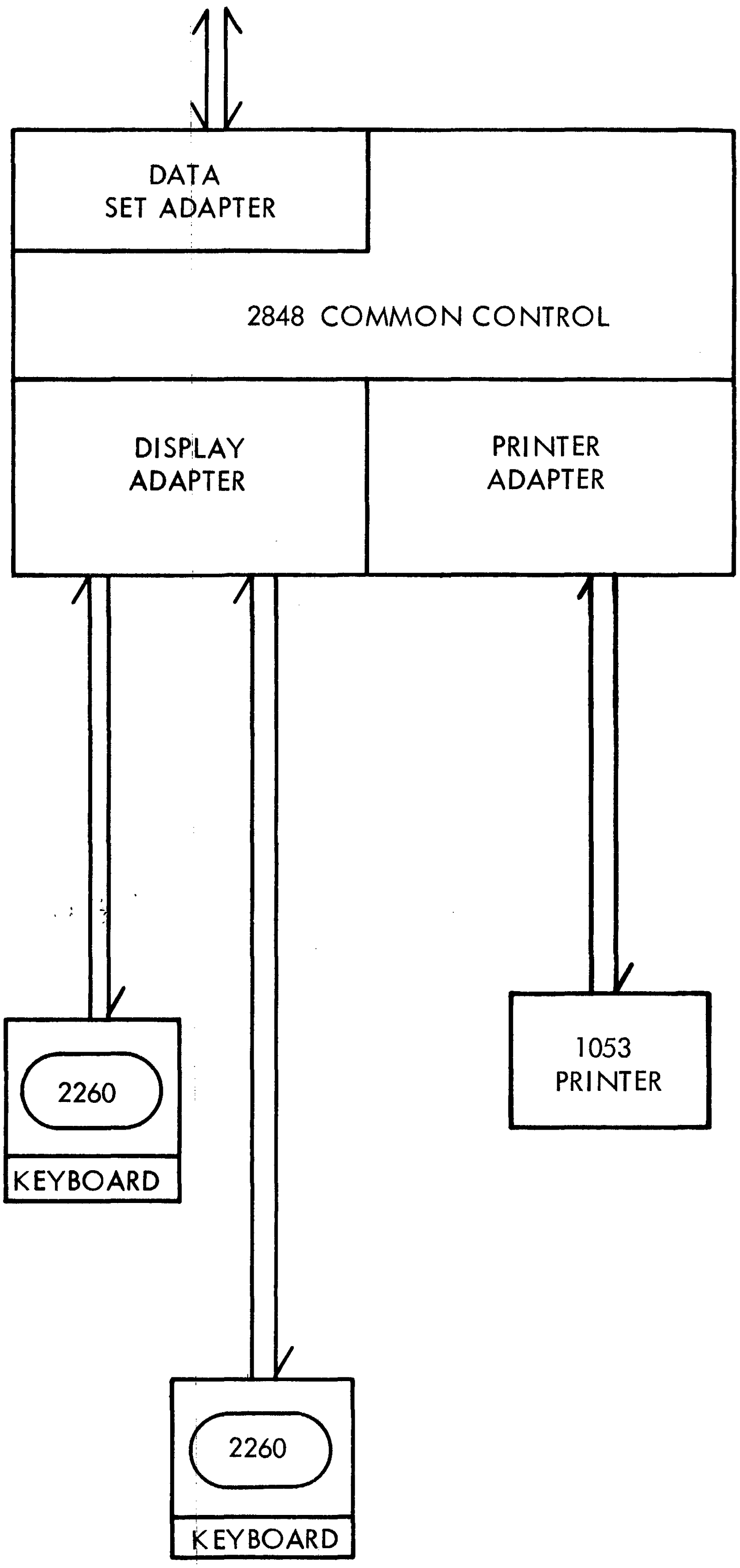
Expand your program to read a message from one 2260 and send it out to the other 2260.

2260 LOCAL OR REMOTE



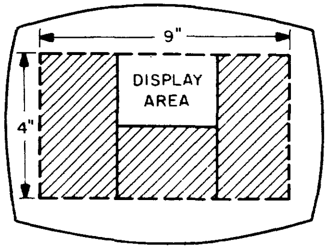
GRAPHIC CONFIGURATION (REMOTE)

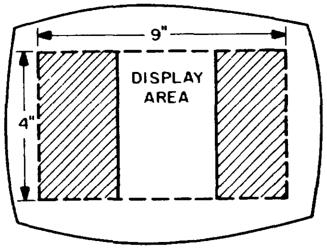


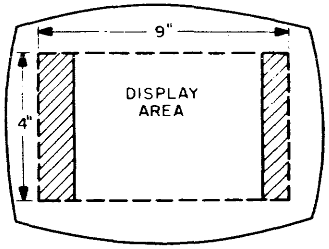

DATA
SET ADAPTER
2848 COMMON CONTROL
DISPLAY
ADAPTER
PRINTER
ADAPTER
2260
KEYBOARD
2260
KEYBOARD
1053
PRINTER

2260/2848-MODEL SUMMARY/CAPACITY

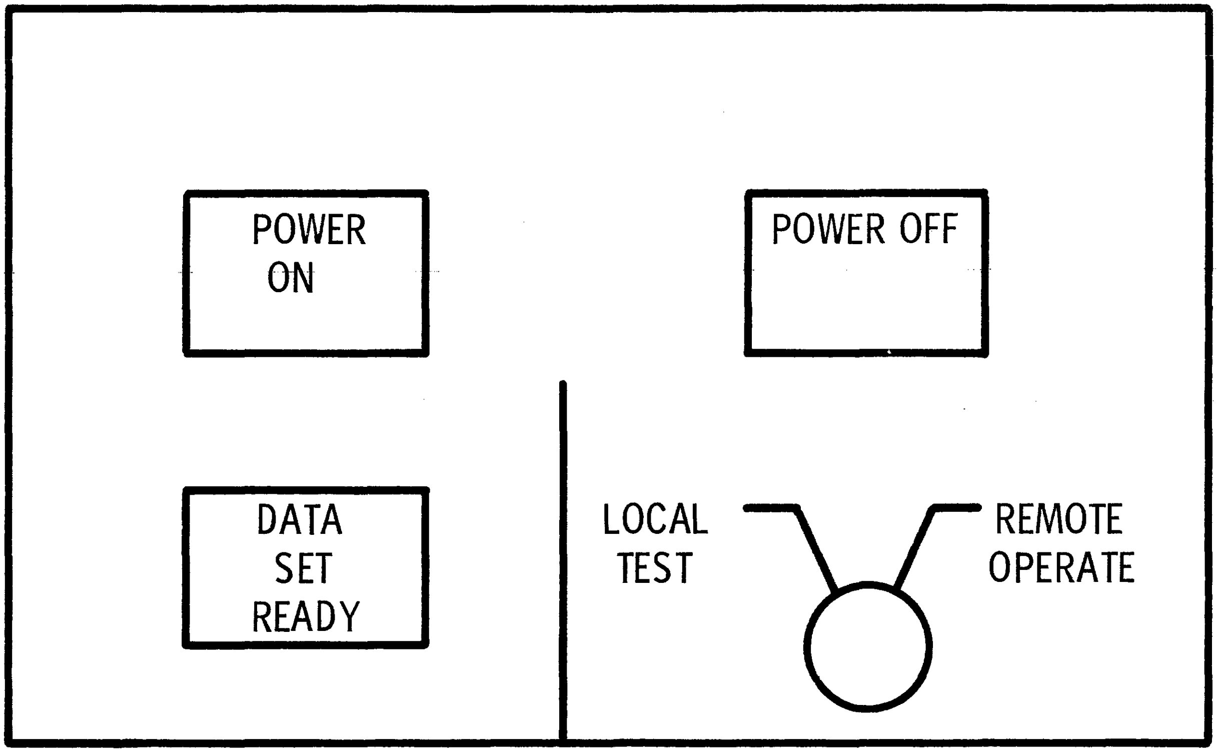
2848	BASIC D.S. CAPACITY	EXPANDED D.S. CAPACITY (TOT.)	CHARACTER CAPACITY
MODEL 1	4-2260 D.S.'s	24-2260 D.S.'S + 1-1053 MOD.4 PRINTER	6 ROWS 40 CHAR. 240 CHAR. (TOT.)
MODEL 2	2-2260 D.S.'s	16-2260 D.S.'s + 1-1053 MOD.4 PRINTER	12 ROWS 40 CHAR. 480 CHAR. (TOT.)
MODEL 3	2-2260 D.S.'s	8-2260 D. S.'s + 1-1053 MOD.4 PRINTER	12 ROWS 80 CHAR. 960 CHAR. (TOT.)



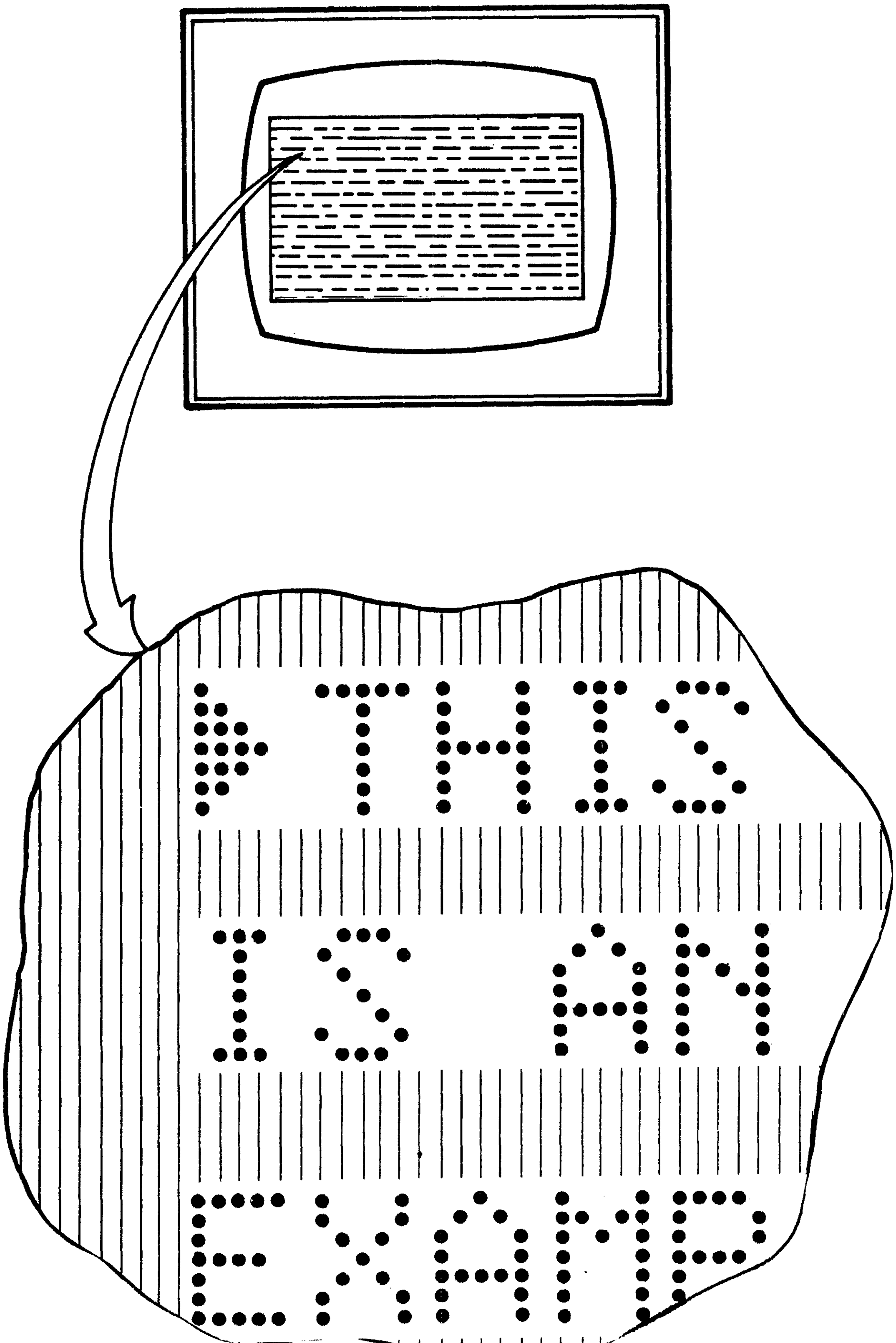




IBM 2848 OPERATOR'S PANEL



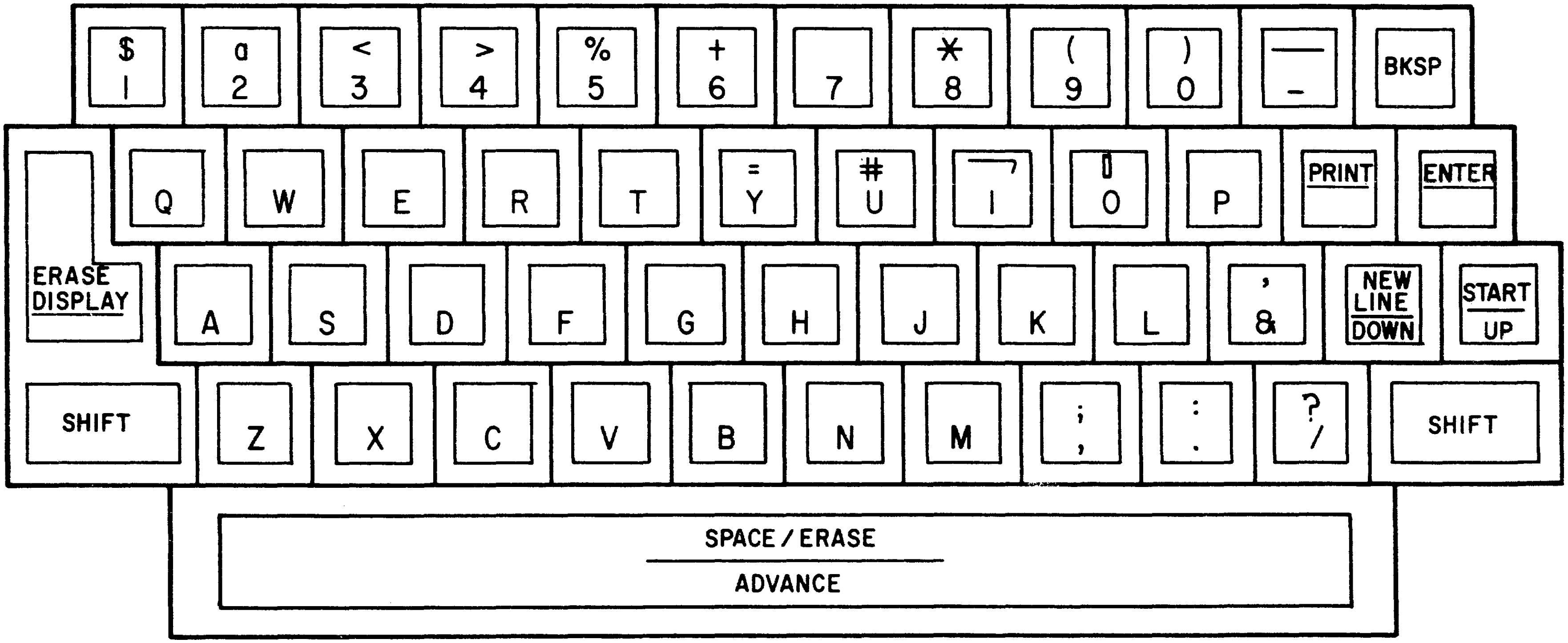
2260 DISPLAY DETAIL



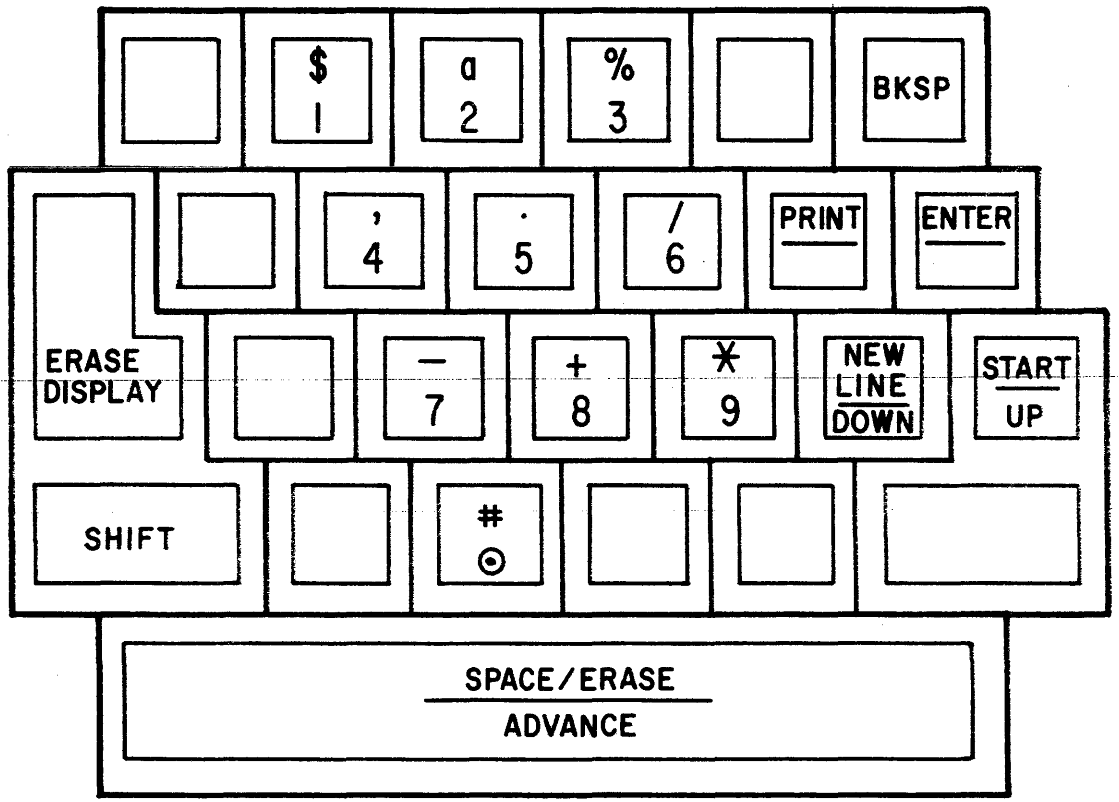
2260 DISPLAYABLE CHARACTERS

A C1	I C9	Q D8	Y E8	6 F6	< 4C Less Than	¬ 5F Logical Not	@ 7C At Symbol
B C2	J D1	R D9	Z E9	7 F7	(4D Left Paren.	% 6C Percent	' 7D Quote
C C3	K D2	S E2	0 F0 Zero	8 F8	+ 4E Plus	_ 6D Under-Score	= 7E Equals
D C4	L D3	T E3	1 F1	9 F9	| 4F Vertical Bar	> 6E Greater than	█ 7F Check Symbol
E C5	M D4	U E4	2 F2	& 50	$ 5B Dollar sign	? 6F	▬ 6A Cursor EOM
F C6	N D5	V E5	3 F3	- 60 Hyphen Minus	* 5C Asterisk	, 6B Comma	► 4A Start MI
G C7	O D6	W E6	4 F4	/ 61 Slash	) 5D Right Paren.	: 7A Colon	◢ 15 New Line
H C8	P D7	X E7	5 F5	. 4B Period	; 5E Semi-Colon	# 7B Number Sign	40 Space

2260 ALPHAMERIC KEYBOARD



2260 NUMERIC KEYBOARD



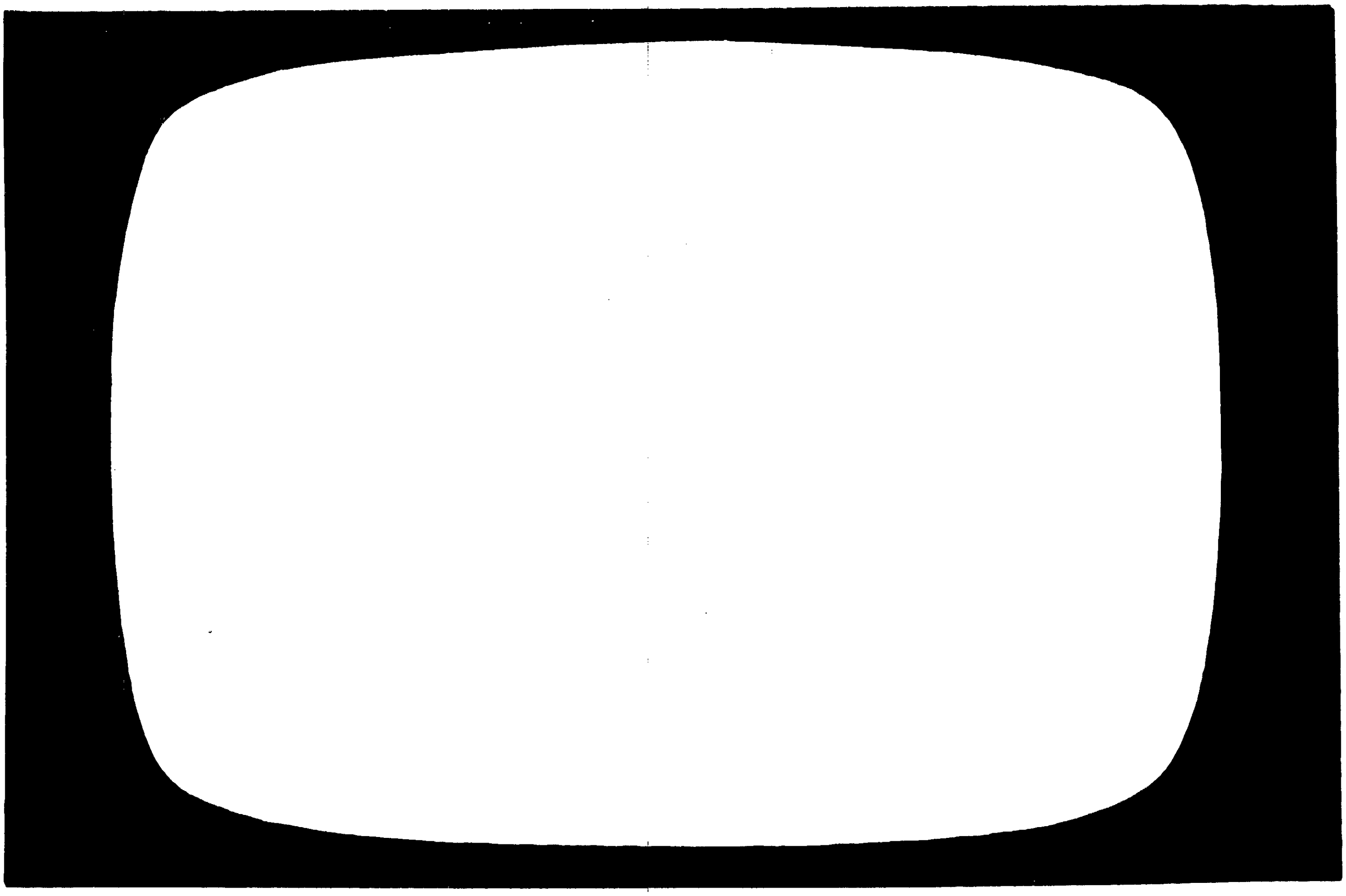
2260 D.S./CONTROL KEY FUNCTION TABLE

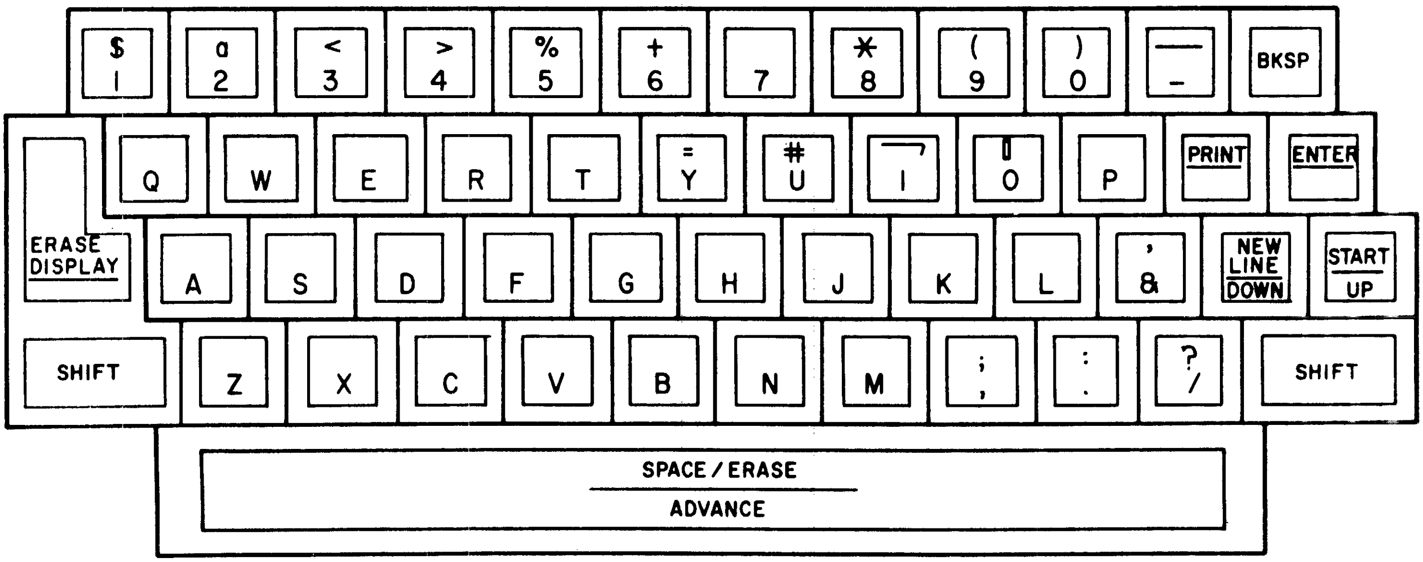
CONTROL KEY	BOTH TYPES OF CURSORS	
	NO SHIFT	SHIFT
1. ERASE DISPLAY	FROM: AB▬ TO AB■▬	ERASE DISPLAY (CURSOR TO 1st DISPLAY POSITION)
2. ENTER (TO C.P.U.)	"	FROM: ▶AB▬◢ TO: AB▬◢
3. PRINT (TO PRINTER)	"	"

LEGEND

▬	= DESTRUCTIVE CURSOR
▮	= NON-DESTRUCTIVE CURSOR
■	= CHECK SYMBOL
▶	= START MI SYMBOL
◢	= NL (NEW LINE) SYMBOL

	DESTRUCTIVE CURSOR		NON–DESTRUCTIVE CURSOR	
	NO SHIFT	SHIFT	NO SHIFT	SHIFT
4. START UP	FROM: AB▬ TO: AB ▬	NO START MI: FROM: AB ▬ TO: AB ▶ ▬ WITH START MI: FROM: ▶AB ▬ ◢ TO: ▶ ▬ ◢	▮ = UP ONE LINE	SAME AS DEST. CURSOR
5. NEW LINE DOWN	"	FROM: ▶AB ▬ TO: ▶AB◢ ▬	▮ = DOWN ONE LINE	"
6. SPACE/ERASE ADVANCE	FROM: AB▬C TO AB ▬	SAME ←	FROM: AB▮C TO: ABC ▮	FROM: AB▮C TO: AB ▮
7. BKSP	FROM: AB ▬ TO A ▬	SAME ←	FROM: AB▮ TO: A▮B	SAME ←




$ 1
a 2
< 3
> 4
% 5
+ 6
7
* 8
(9
) 0
— –
BKSP
ERASE DISPLAY
Q
W
E
R
T
= Y
# U
I
O
P
PRINT
ENTER
A
S
D
F
G
H
J
K
L
' &
NEW LINE DOWN
START UP
SHIFT
Z
X
C
V
B
N
M
; ,
: .
? /
SHIFT
SPACE / ERASE
ADVANCE

CONTROL CHARACTERS

STX - START OF TEXT

ETX - END OF TEXT

CAN - CANCEL

ACK - ACKNOWLEDGE

NAK - NEGATIVE ACKNOWLEDGE

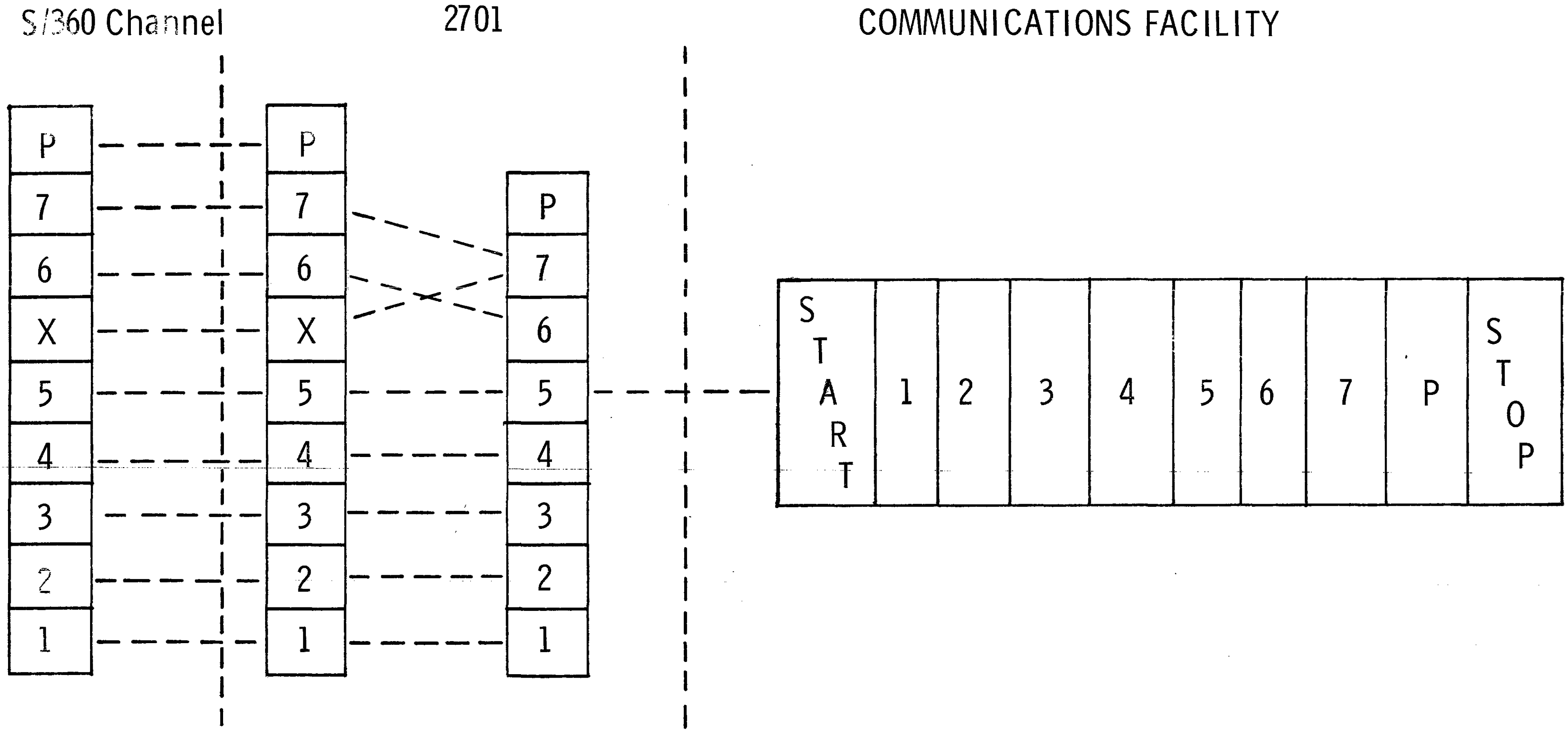
SOH - START OF HEADING

EOT - END OF TRANSMISSION

LF - LINE FEED

CONTROL MODE ESTABLISHED BY	TEXT MODE ESTABLISHED BY
SOH	STX
EOT	
ETX	

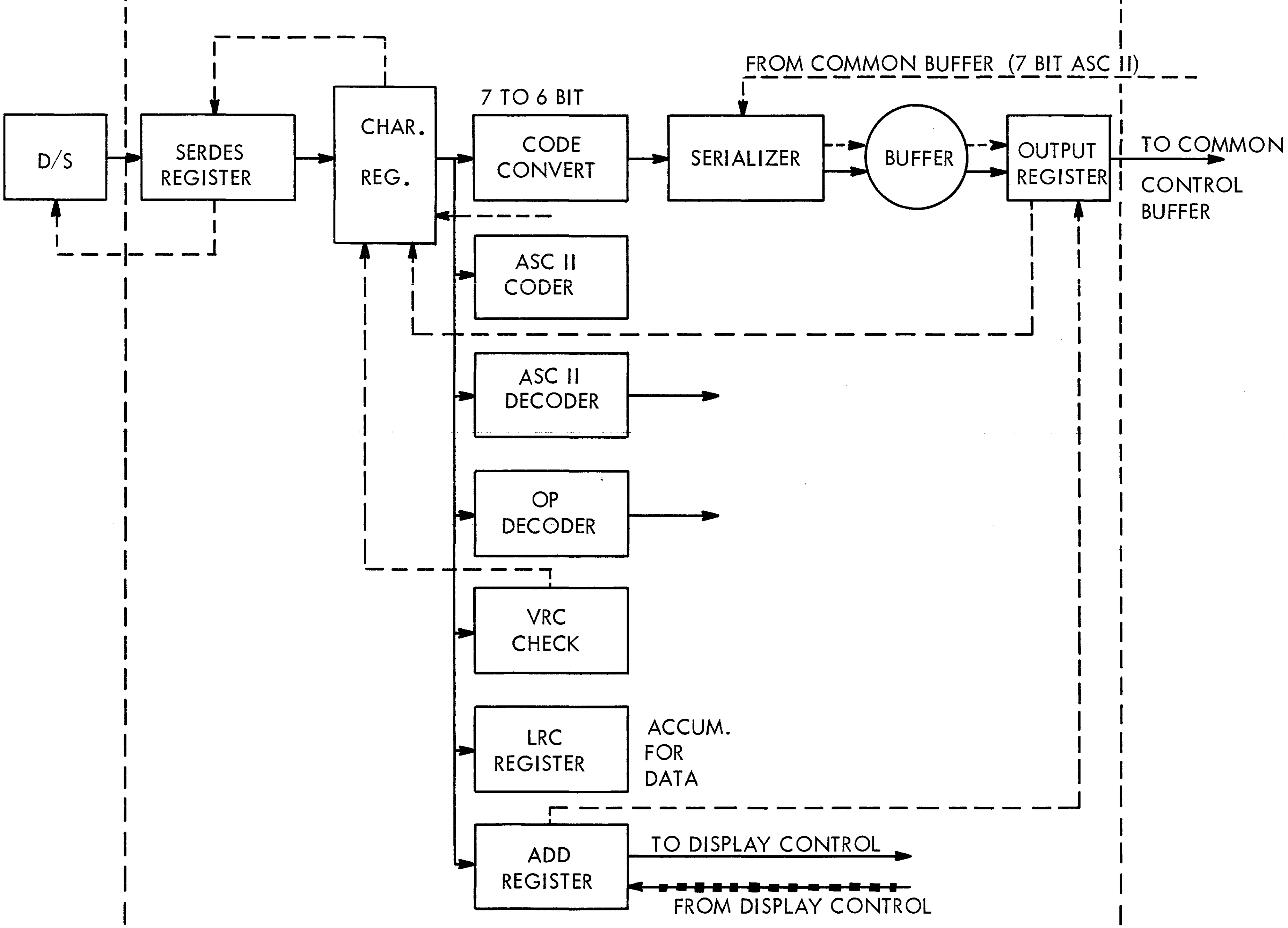
ASC II CODE SETS



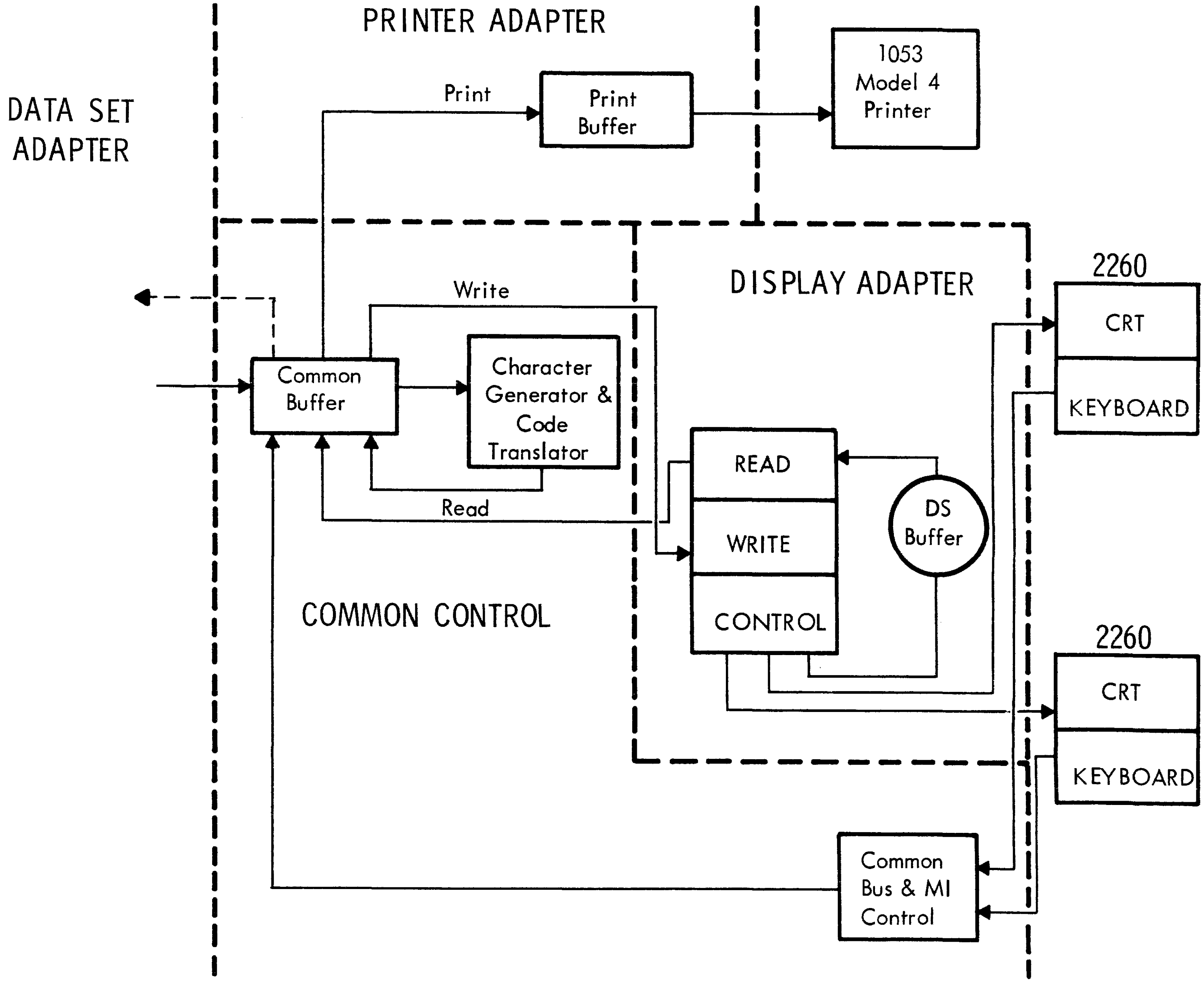
ASC II-8 CODE SET

Bits				B7 →	0	0	0	0	1	1	1	1
				B6 →	0	0	1	1	0	0	1	1
				BX →	0	0	0	0	1	1	1	1
				B5 →	0	1	0	1	0	1	0	1
B4	B3	B2	B1	Col Row	0	1	2	3	4	5	6	7
0	0	0	0	0			SP	0		P	@	P
0	0	0	1	1	SOH		—	1	A	Q	A	Q
0	0	1	0	2	STX		■	2	B	R	B	R
0	0	1	1	3	ETX		#	3	C	S	C	S
0	1	0	0	4	EOT		$	4	D	T	D	T
0	1	0	1	5		NAK	%	5	E	U	E	U
0	1	1	0	6	ACK		&	6	F	V	F	V
0	1	1	1	7			'	7	G	W	G	W
1	0	0	0	8		CAN	(	8	H	X	H	X
1	0	0	1	9			)	9	I	Y	I	Y
1	0	1	0	10	◢		*	:	J	Z	J	Z
1	0	1	1	11			+	;	K		K	▐
1	1	0	0	12			,	<	L		L	¬
1	1	0	1	13			-	=	M	▶	M	
1	1	1	0	14			.	>	N		N	|
1	1	1	1	15			/	?	O	-	O	

DATA FLOW FOR DATA SET ADAPTER



DATA FLOW FOR COMMON CONTROL AND ADAPTERS



SPECIFIC POLL OF 2260

	CHAN. PROG.		DATA FLOW
			S/360 → REMOTE
	RD TI		
1. WR	TABLE	CD	EOT, EOT, EOT →
2. WR	TRMLST	CD	2260 ADD , TERM. ADD. →
3. WR	TABLE	CC	COMMAND (RD MI) →
4. RD	INAREA	CD	← STX, 2260 ADD.
5. RD	INAREA+2		← TXT (CAN), ETX, LRC
	RD TP		
1. WR	TABLE	CC	NAK →
2. RD	INAREA		← STX , 2260 ADD. TXT, ECT.
	WR TA		
1. WR	TABLE		EOT, EOT , EOT, STX →

REMOTE DEVICE ADDRESS ASSIGNMENTS

ADDRESS BYTE (ASCII-8)								DEVICE NUMBER (2260 DISPLAY STATION OR 1053 MODEL 1 PRINTER*)
7	6	X	5	4	3	2	1	
1	0	1	0	0	0	0	0	DEVICE 1
1	0	1	0	0	0	0	1	DEVICE 2
		CONTINUED SEQUENTIALLY						
1	0	1	0	1	1	1	1	DEVICE 16
1	0	1	1	0	0	0	0	DEVICE 17
		CONTINUED SEQUENTIALLY						
1	0	1	1	1	0	0	0	DEVICE 25

*HIGHEST ADD. MUST BE 1053

2848-2260 REMOTE COMMANDS

Command	ASCII-8 Code Structure	Feature Required
COMMAND	7 6 X 5 4 3 2 1	
Specific Poll - 2260	0 1 0 0 0 0 0 0	Keyboard Feature
Specific Poll - Printer	0 1 0 0 0 0 0 0	Printer Feature
General Poll*	0 1 0 0 0 0 0 0	Keyboard Feature or Printer Feature
Read Addressed Full DS Buffer	0 1 0 1 0 0 0 0	
Write 2260	1 0 1 0 0 0 0 0	
Write Printer	1 0 1 0 0 0 0 0	Printer Feature
Line Address Write	1 0 1 1 0 0 0 0	Line Addressing Feature
Erase/Write	1 1 1 0 0 0 0 0	

*3rd. Char. of Add. Seq. All 1's.

SPECIFIC POLL OF 1053 PRINTER

CHAN. PROG.				DATA FLOW S/360		REMOTE
RD TI						
1.	WR	TABLE	CD		EOT, EOT, EOT	→
2.	WR	TRMLST	CD		1053 ADD., TERM. ADD.	→
3.	WR	TABLE	CC		COMMAND (RD MI)	→
4.	RD	INAREA	CD	←	STX, PRNTR. ADD	
5.	RD	INAREA + 2		←	EXT, LRC	
WR TT						
1.	WR	OUTAREA	CC		LRC, ETX, TXT, STX	→
2.	RD	RESPN + 1		←	ACK	
WR TN						
1.	WR	TABLE			EOT, EOT, EOT	→

GENERAL POLL

CHAN. PROG.			DATA FLOW	
RD TI			S/360	REMOTE
1. WR	TABLE	CD	EOT, EOT, EOT →	
2. WR	TRMLST	CD	X'FF', TERM. ADD. →	
3. WR	TABLE	CC	COMMAND (RD MI) →	
4. RD	INAREA	CD	← STX, DEV. ADD.	
5. RD	INAREA+2		← (TXT),(CAN), ETX, LRC	

WR TT (IF PRINTER)			
1. WR	OUTAREA	CC	LRC, ETX, TXT, STX →
2. RD	RESPN + 1		← ACK

RD TT (IF 2260)			
1. WR	TABLE	CC	ACK →
2. RD	INAREA		STX, DEV. ADD., TXT, ETC

READ FULL DS BUFFER

CHAN. PROG.			DATA FLOW	
	RD TB		S/360	REMOTE
1. WR	TABLE	CD	EOT, EOT, EOT →	
2. WR	TRMLST	CD	2260 ADD, TERM. ADD →	
3. WR	TABLE	CC	COMMAND (RD. BUFF.) →	
4. RD	INAREA		← STX, 2260 ADD, TXT, CAN, ETX, LRC	

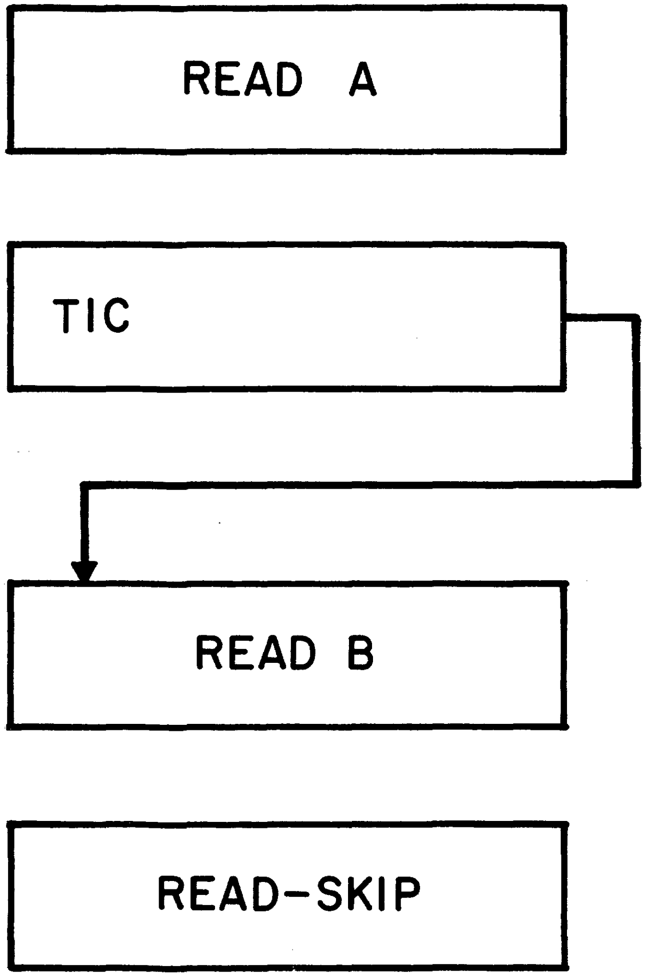
WRITE DS OR 1053 PRINTER

CHAN. PROG.			DATA FLOW
WR TI			S/360 → REMOTE
1. WR	TABLE	CD	EOT, EOT, EOT →
2. WR	TRMLST	CD	DEV. ADD, TERM. ADD →
3. WR	TABLE	CC	COMMAND (WR) →
4. RD	RESPN	CC	← ACK
5. WR	TABLE	CD	STX →
6. WR	OUTAREA	CC	LRC, ETX, TXT →
7. RD	RESPN + 1		← ACK
WR TN OR RESET			
WR	TABLE		EOT, EOT, EOT →
WR TT			
1. WR	OUTAREA	CC	LRC, ETX, TXT, STX →
2. RD	RESPN + 1		← ACK

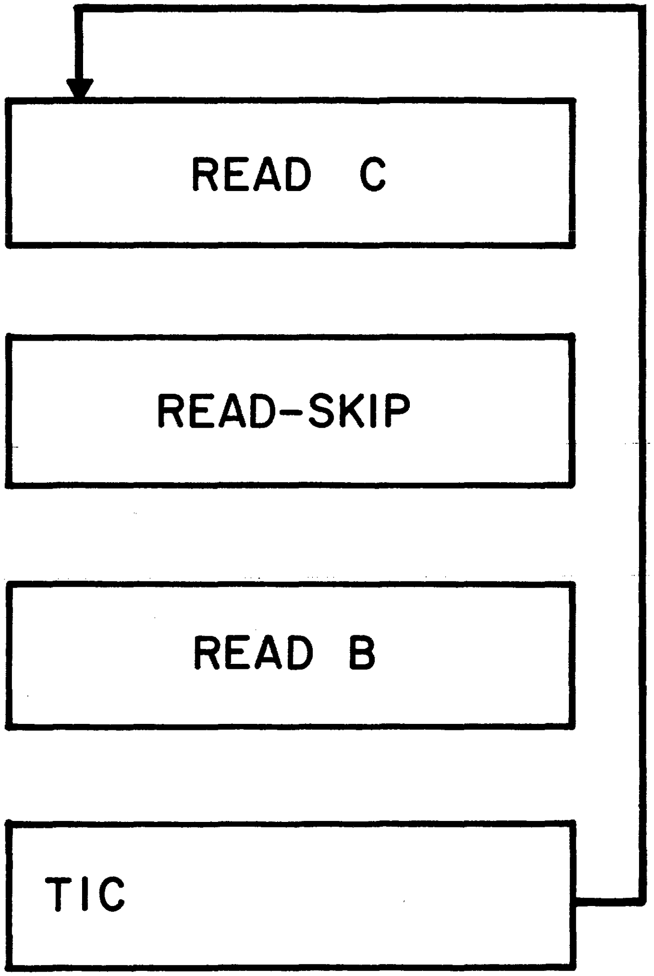
DISPLAY LINE ADDRESSES

ASCII-8 DATA BYTE FORMAT								SELECTED LINE		
BITS										
7	6	X	5	4	3	2	1			
0	1	0	1	0	0	0	0	ONE	2848 DC MODEL 1	2848 DC MODELS 2 AND 3
0	1	0	1	0	0	0	1	TWO		
0	1	0	1	0	0	1	0	THREE		
0	1	0	1	0	0	1	1	FOUR		
0	1	0	1	0	1	0	0	FIVE		
0	1	0	1	0	1	0	1	SIX		
0	1	0	1	0	1	1	0	SEVEN		
0	1	0	1	0	1	1	1	EIGHT		
0	1	0	1	1	0	0	0	NINE		
0	1	0	1	1	0	0	1	TEN		
0	1	0	1	1	0	1	0	ELEVEN		
0	1	0	1	1	0	1	1	TWELVE		

DYNAMIC BUFFERING READ CCW's

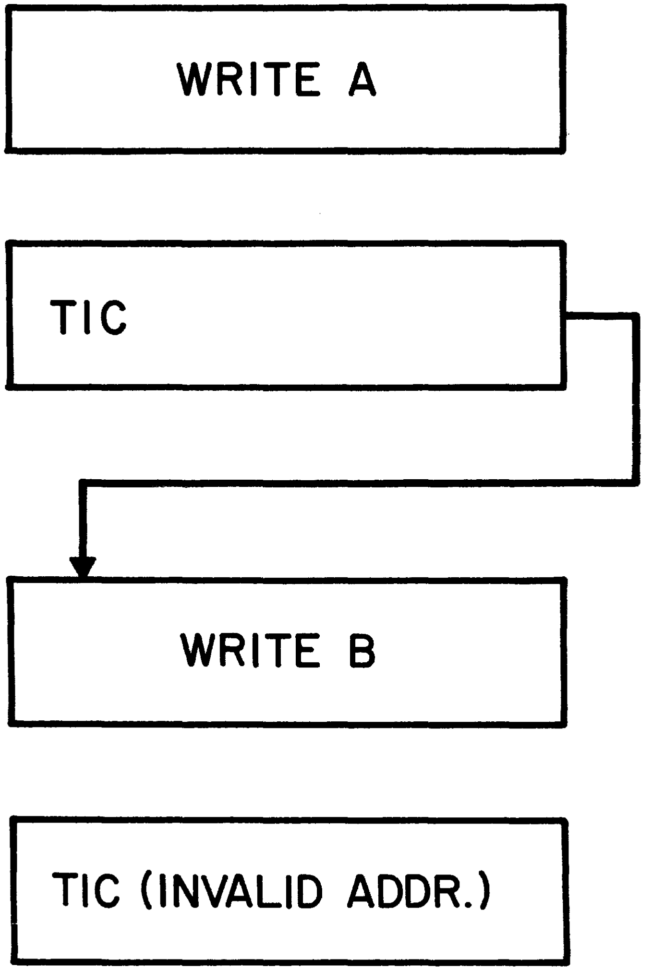


RD. INTO 1ST BUFF.
AFTER PCI ON A
BEFORE PCI ON B



RD. INTO 2ND BUFF.
AFTER PCI ON B
BEFORE PCI ON C

DYNAMIC BUFFERING WRITE CCW's



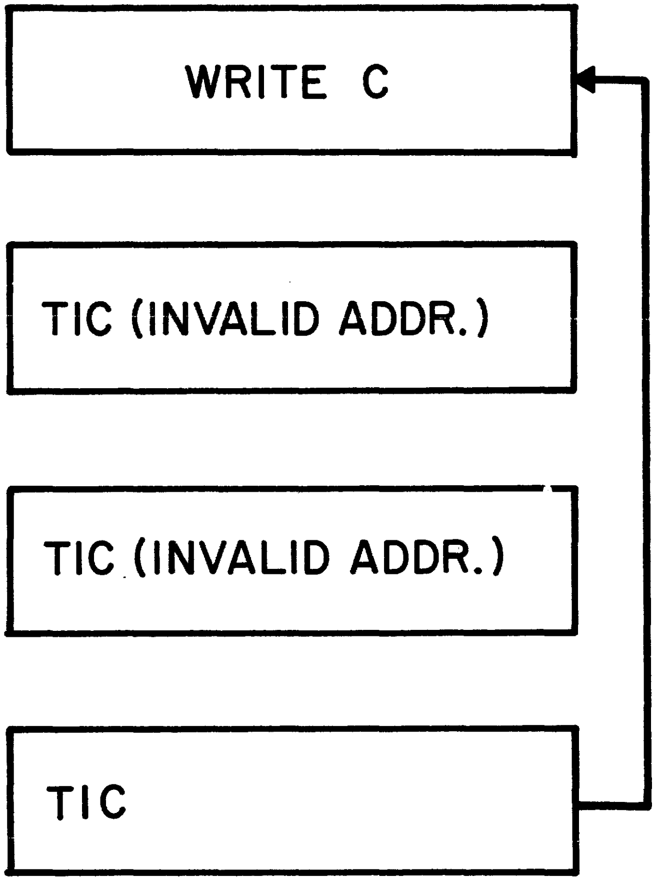
WRITE A

TIC

WRITE B

TIC (INVALID ADDR.)

WR. OUT OF 1ST BUFF.
AFTER PCI ON A
BEFORE PCI ON B



WRITE C

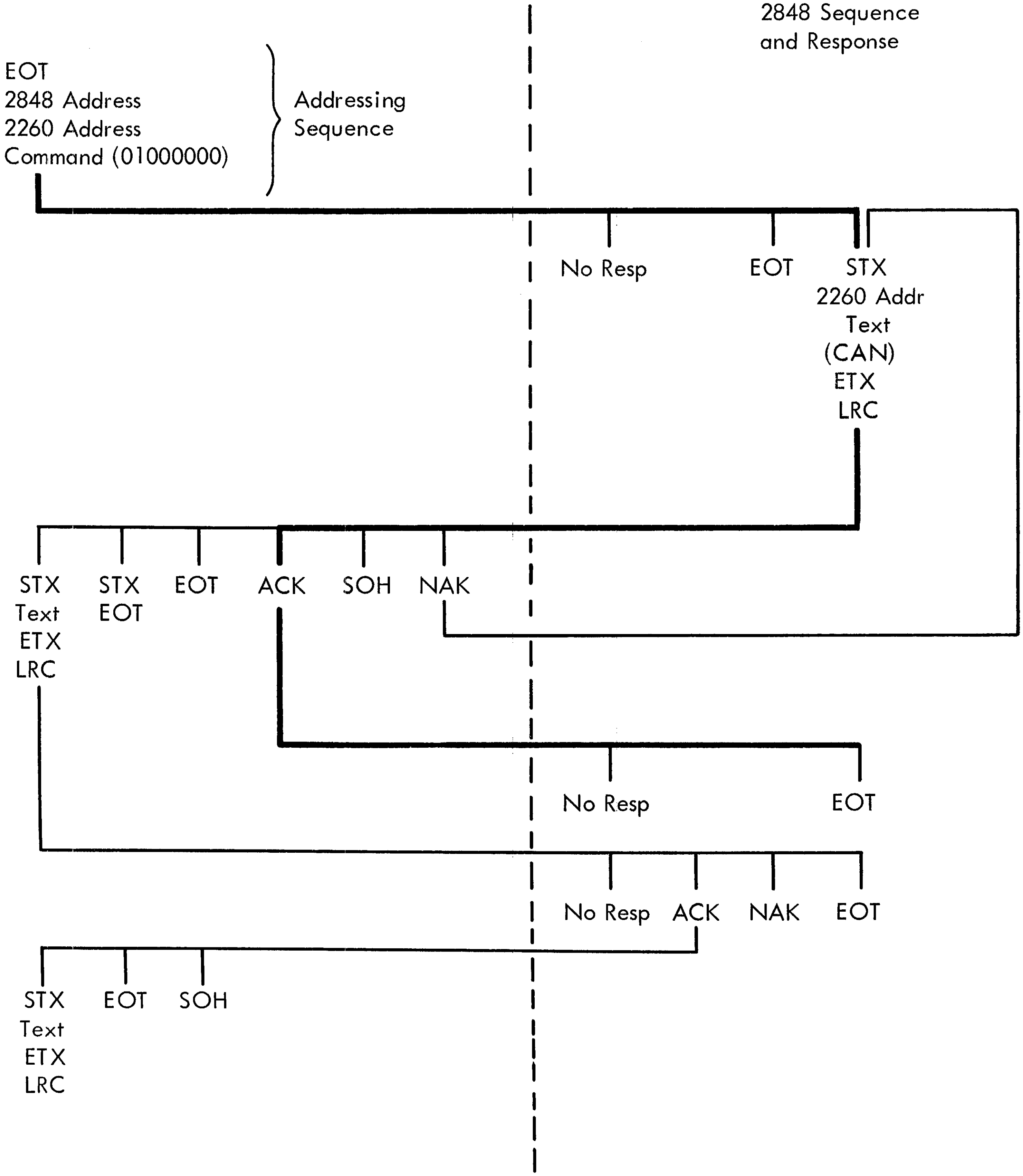
TIC (INVALID ADDR.)

TIC (INVALID ADDR.)

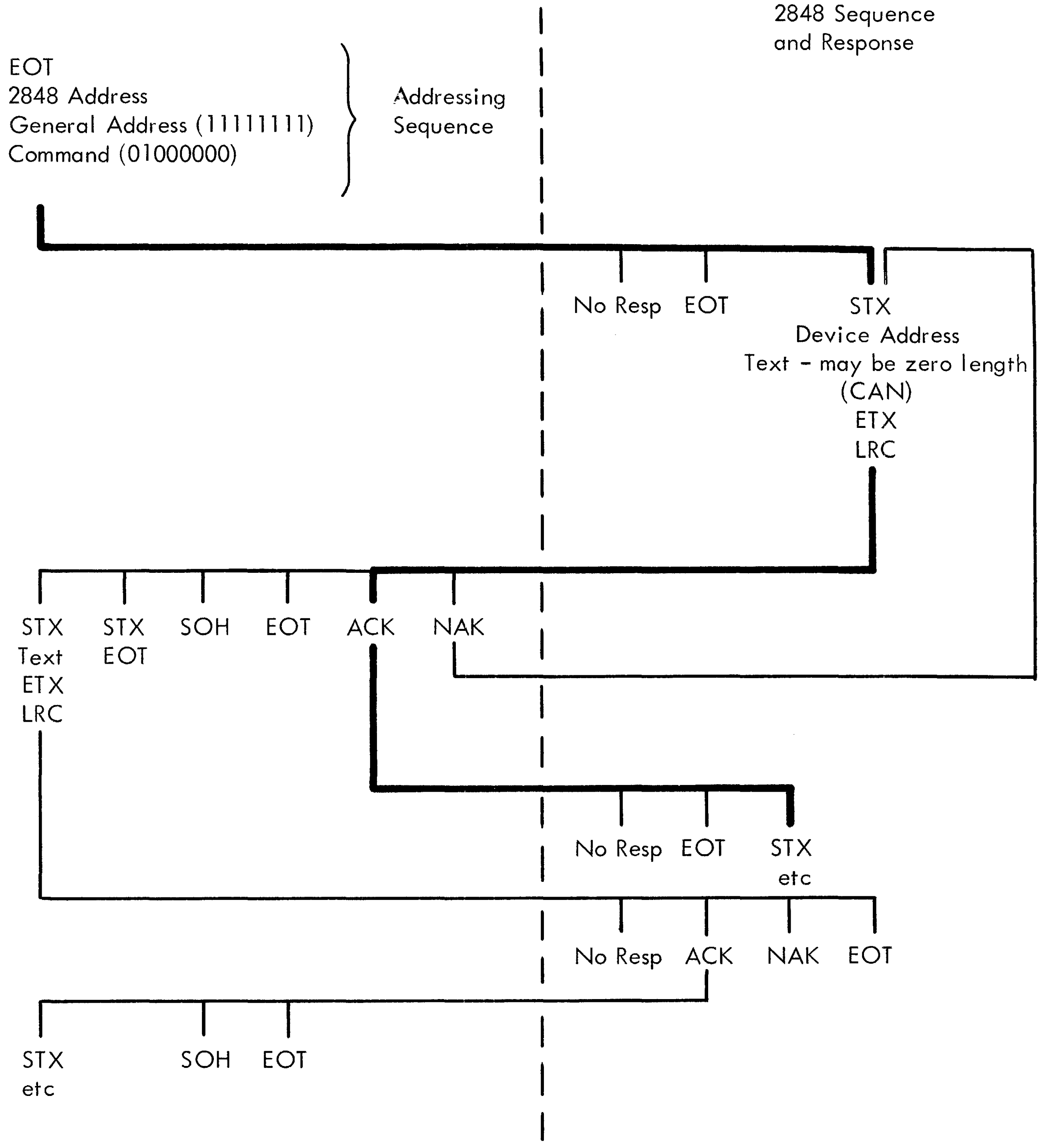
TIC

WR. OUT OF 2ND BUFF.
AFTER PCI ON B
BEFORE PCI ON C

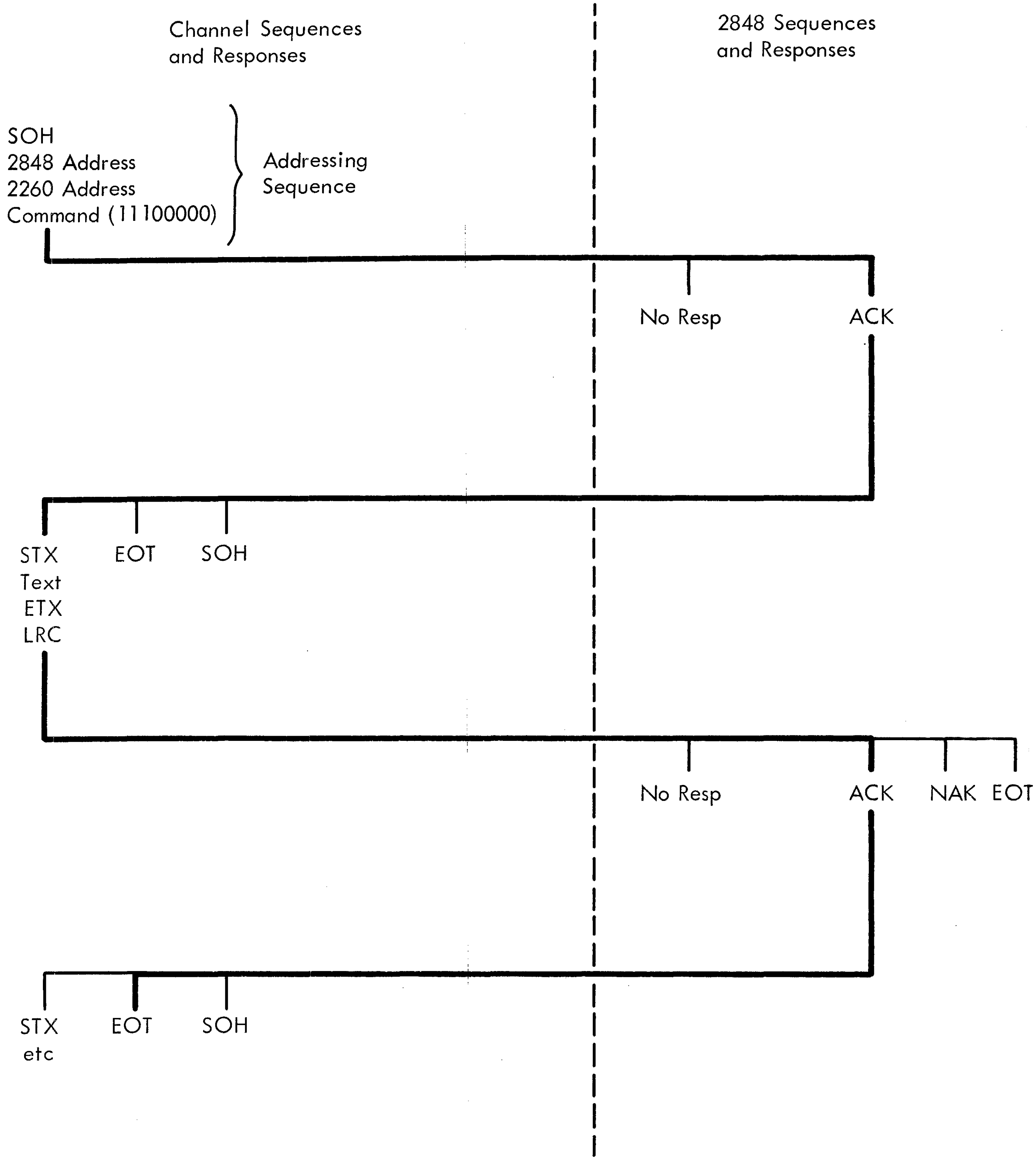
SPECIFIC POLL TO 2260 DS



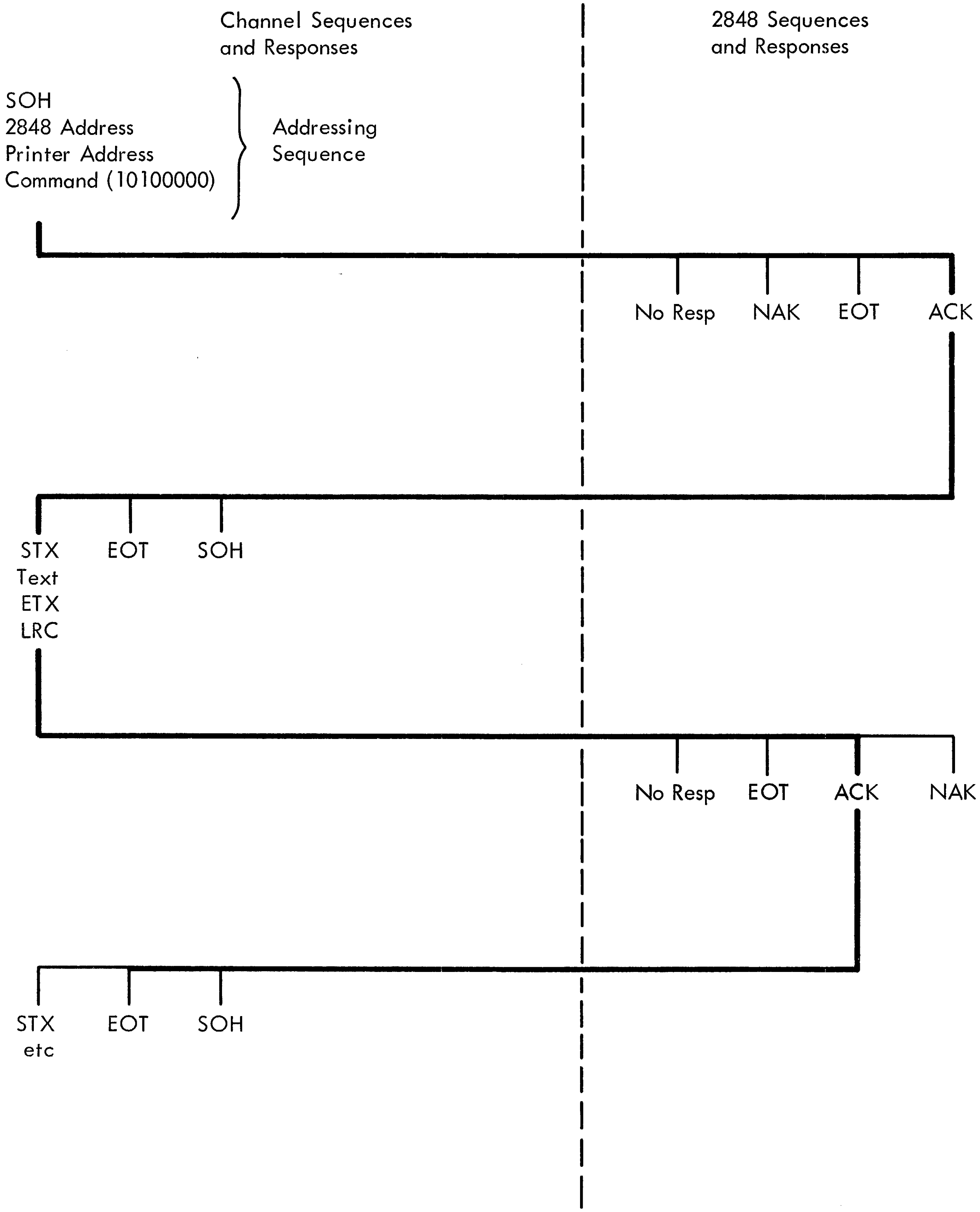
GENERAL POLL



ERASE/WRITE


Channel Sequences and Responses
2848 Sequences and Responses
SOH
2848 Address
2260 Address
Command (11100000)
Addressing Sequence
No Resp
ACK
STX
Text
ETX
LRC
EOT
SOH
No Resp
ACK
NAK
EOT
STX
etc
EOT
SOH

WRITE PRINTER



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01JAN67    5/01/67

000000                                 1          CSECT
                                       2          SAVE  (14,12) REGISTERS SAVED IN CALLING ROUTINES SAVE AREA
000000                                 3+         DS    0H
000000 90EC D00C             0000C     4+         STM   14,12,12(13) SAVE REGISTERS
000004 05C0                            5          BALR  BASEREG1,0
000006                                 6          USING *,BASEREG1
000000                                 7          USING IHADCB,DCBREG        BASE REG. FOR DSECT
000000                                 8          USING IECTDECB,DECBREG    BASE REG. FOR DSECT
000006 50D0 C7FA             00800     9          ST    13,SAVE+4  ADD. OF CALLING ROUT. SAVE AREA TO MY SAVE
00000A 41A0 C7F6             007FC    10          LA    10,SAVE    ADD. MY SAVE AREA TO REG 10
00000E 50A0 D008             00008    11          ST    10,8(0,13) ADD. OF MY SAVE AREA TO CALLER'S SAVE AREA
000012 18DA                           12          LR    13,10      ADD. OF MY SAVE AREA TO REG. 13
000014 4140 C512             00518    13          LA    DCBREG,DCB1  DCB1 ADD TO REG
                                      14          GETMAIN R,LV=968   GET STORAGE FOR BUFFERS
000018 4100 03C8             003C8    15+         LA    0,968(0,0) LOAD LENGTH
00001C 4510 C01A             00020    16+         BAL   1,*+4 INDICATE GETMAIN
000020 0A0A                           17+         SVC   10 ISSUE GETMAIN SVC
000022 5010 4014             00014    18          ST    1,DCBBUFCB   ADD. OF POOL TO DCB1
                                      19          BUILD (1),20,48   20 BUFF OF 48 BYTES. AVAIL. TO ANY DCB
000026 47F0 C028             0002E    20+         B     *+8 BRANCH AROUND VALUES
00002A 0014                           21+         DC    AL2(20) NUMBER OF BUFFERS
00002C 0030                           22+         DC    AL2(48) LENGTH OF BUFFERS
00002E D203 1000 C024  00000 0002A    23+         MVC   0(4,1),*-4 MOVE INTO LIST
000034                                24+         CNOP  0,4
000034 45F0 C042             00048    25+         BAL   15,*+20 LOAD SUP.PARAMLIST ADR
000038 00000040                       26+         DC    A(*+8) ADDR OF EP PARAMETER
00003C 00000000                       27+         DC    A(0) DCB ADDR PARAMETER
000040 C9C5C3C2C2C6C2F1               28+         DC    CL8'IECBBFB1' EP PARAMETER
000048 0A06                           29+         SVC   6 ISSUE LINK SVC
00004A 4180 0030             00030    30          LA    WRKREG1,48     BUFFER LENGTH TO REG
00004E 4080 4018             00018    31          STH   WRKREG1,DCBBUFL   STORE BUFFER LENGTH IN DCB
000052 9200 C846       0084C          32          MVI   RPLYECB,X'00'
                                      33          WTOR  'CHOOSE  1-2260 2-1050 3-BOTH',REPLY,1,RPLYECB
000056 0700                           34+         CNOP  0,4
000058 4510 C07E             00084    35+         BAL   1,IHB0007A BRANCH AROUND LIST
00005C 01                             36+         DC    AL1(1) REPLY LENGTH
00005D 000864                         37+         DC    AL3(REPLY) REPLY ADDRESS
000060 0000084C                       38+         DC    A(RPLYECB) ECB ADDRESS
000064 0020                           39+         DC    AL2(IHB0007-*) MESSAGE LENGTH
000066 0000                           40+         DC    AL2(0)
000068 C3C8D6D6E2C54040               41+         DC    C'CHOOSE  1-2260 2-1050 3-BOTH' MESG
000084                                42+IHB0007  EQU   *
000084                                43+IHB0007A DS    0H
000084 0A23                           44+         SVC   35 ISSUE WTOR SVC
                                      45          WAITR ECB=RPLYECB
000086 4110 C846             0084C    46+         LA    1,RPLYECB LOAD PARAMETER REG 1
00008A 4100 0001             00001    47+         LA    0,1(0,0) COUNT OMITTED,1 USED
00008E 1300                           48+         LCR   0,0 INDICATE WAITR MACRO
000090 0A01                           49+         SVC   1 LINK TO WAIT ROUTINE
000092 95F2 C85E       00864          50          CLI   REPLY,X'F2'
000096 4740 C09C             000A2    51          BL    DSON    BRANCH IF 2260 ON SYSTEM
00009A 4780 C24E             00254    52          BE    TEN50   BRANCH IF 1050 ON SYSTEM
00009E 4720 C226             0022C    53          BH    DONE      NOT CODED AT THIS TIME
0000A2 4160 CB62             00B68    54 DSON     LA    TBLREG,IECTSSCI  LOAD ADD OF TRANS TABLE INTO REG
0000A6 4130 C57A             00580    55          LA    LISTREG,ADLST1 TRMLST ADD FOR ADDRESSING 1ST DS TO REG
0000AA 4170 C592             00598    56 B1       LA    AREAREG,OPNMESS  ADD OF OPEN MESSAGE TO REG
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          FC1JAN67   5/01/67

000AE 4180 0081             00081    57             LA    LNGTHREG,129  LENGTH OF OPEN MESSAGE TO REG
0000B2 45E0 C4F0            004F6    58             BAL   RTREG,TRANSL  BRANCH TO TRANSLATE ROUTINE
0000B6 45E0 C450            00456    59             BAL   RTREG,OPEN    BRANCH TO OPEN FOR 2260'S
0000BA 4120 C552            00558    60             LA    DECBREG,DECB1  DECB ADD TO REG
0000BE 4170 C592            00598    61             LA    AREAREG,OPNMESS  ADD OF OPEN MESSAGE TO REG
0000C2 4150 0001            00001    62             LA    LINEREG,1      LINE NUMBER WITHIN LINE GROUP
0000C6 95F2 C85E      00864          63             CLI   REPLY,X'F2'
0000CA 4780 C25A            00260    64             BE    B2             BRANCH IF 1050
0000CE 45E0 C32E            00334    65             BAL   RTREG,WRTS  BRANCH TO ERASE AND WRITE FOR DS
                                     66             WAIT  ECB=DECB1   WAIT FOR 1ST WRITE TO COMPLETE
0000D2 4110 C552            00558    67+            LA    1,DECB1 LOAD PARAMETER REG 1
0000D6 4100 0001            00001    68+            LA    0,1(0,0) COUNT OMITTED,1 USED
0000DA 0A01                          69+            SVC   1 LINK TO WAIT ROUTINE
0000DC 45E0 C416            0041C    70             BAL   RTREG,CKICCOMP  BRANCH TO CHECK COMPLETION CODE
0000E0 4130 C57D            00583    71             LA    LISTREG,ADLST2  TRMLST OF 2ND DS TO REG.
0000E4 45E0 C32E            00334    72             BAL   RTREG,WRTS    BRANCH TO WRITE ERASE FOR 2260
                                     73             WAIT  ECB=DECB1
0000E8 4110 C552            00558    74+            LA    1,DECB1 LOAD PARAMETER REG 1
0000EC 4100 0001            00001    75+            LA    0,1(0,0) COUNT OMITTED,1 USED
0000F0 0A01                          76+            SVC   1 LINK TO WAIT ROUTINE
0000F2 45E0 C416            0041C    77             BAL   RTREG,CKICCOMP  BRANCH TO CHECK I/O COMPLETION
0000F6 4160 C982            00988    78 CONTPOLL LA    TBLREG,IECTRSCI  ASCII TO EBCDIC
0000FA 4130 C586            0058C    79             LA    LISTREG,POLLST1 ADD OF POLL LIST FOR DS GEN POLL TO REG
0000FE 45E0 C3EE            003F4    80 B3          BAL   RTREG,RDTI  BRANCH TO READ INITIAL TO START POLLING
000102 41A0 C616            0061C    81             LA    WRKREG3,COREFILE   ADD OF COREFILE TO REG
000106 1B88                          82             SR    WRKREG1,WRKREG1  ZERO OUT REG
                                     83             WAIT  ECB=DECB1
000108 4110 C552            00558    84+            LA    1,DECB1 LOAD PARAMETER REG 1
00010C 4100 0001            00001    85+            LA    0,1(0,0) COUNT OMITTED,1 USED
000110 0A01                          86+            SVC   1 LINK TO WAIT ROUTINE
000112 45E0 C416            0041C    87             BAL   RTREG,CKIOCOMP
000116 5870 200C            0000C    88             L     AREAREG,DECAREA ADD OF 1ST BUFF TO REG.
00011A 5070 C83E            00844    89             ST    AREAREG,RELAREA STORE AREA ADD FOR RELEASING BUFFERS
00011E 45E0 C3D2            003D8    90             BAL   RTREG,WRTA    BRANCH TO WRITE POSITIVE ACK
                                     91             WAIT  ECB=DECB1
000122 4110 C552            00558    92+            LA    1,DECB1 LOAD PARAMETER REG 1
000126 4100 0001            00001    93+            LA    0,1(0,0) COUNT OMITTED,1 USED
00012A 0A01                          94+            SVC   1 LINK TO WAIT ROUTINE
00012C 45E0 C416            0041C    95             BAL   RTREG,CKICCOMP
000130 95F2 C85E      00864          96             CLI   REPLY,X'F2'
000134 4780 C278            0027E    97             BE    B4          BRANCH IF 1050 SYSTEM
000138 95A1 7005      00005          98             CLI   5(AREAREG),X'A1' CHECK TO SEE WHICH DS RESPONDED TO POLL
00013C 4780 C142            00148    99             BE    DSTWO   BRANCH IF DS AT ADD. A1 SENT MESSAGE
000140 4130 C57A            00580   100             LA    LISTREG,ADLST1  ADD. OF A0 ADDRESSING LIST TO REG
000144 47F0 C146            0014C   101             BC    15,B5
000148 4130 C57D            00583   102 DSTWO       LA    LISTREG,ADLST2  ADD. OF A1 ADDRESSING LIST TO REG
00014C 5880 C862            00868   103 B5          L     LNGTHREG,ZERO1  INDICATES DYNAMIC BUFFERING TO TRANSLATE
000150 45E0 C4F0            004F6   104             BAL   RTREG,TRANSL  BRANCH TO TRANSLATE ROUTINE
000154 95F2 C85E      00864         105             CLI   REPLY,X'F2'
000158 4780 C280            00286   106             BE    B6           BRANCH IF 1050
00015C 95F2 7006      00006         107             CLI   6(AREAREG),X'F2'  SEE WHAT OPERATION REQUESTED
000160 4740 C166            0016C   108             BL    DSPLY    BRANCH TO DISPLAY COREFILE
000164 4780 C1C6            001CC   109             BE    CHANGE   BRANCH TO CHANGE COREFILE
000168 4720 C212            00218   110             BH    CONTCOMP   BRANCH TO CONTINUE COMPARE
00016C D203 C616 C862 0061C 00868   111 DSPLY       MVC   COREFILE(4),ZERO1  ZEROS TO FIRST WORD OF COREFILE
000172 5870 C83E            00844   112             L     AREAREG,RELAREA  ADD. OF INPUT AREA TO REG
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01JAN67   5/01/67

000176 45E0 C4A6            004AC   113            BAL   RTREG,RELBUFF   BRANCH TO RELEASE BUFFERS USED FOR RD
00017A 4170 C616            0061C   114            LA    AREAREG,CCREFILE ACD. OF MESS. TO DS TO REG.
00017E 58B0 C866            0086C   115            L     LNGTHREG,CCREFLN  LENGTH CF COREFILE TO REG.
000182 4160 C882            00888   116            LA    TBLREG,IECTSSCI      EBCDIC TC ASCII TABLE ADD. TO REG.
000186 45E0 C4F0            004F6   117            BAL   RTREG,TRANSL  BRANCH TC TRANSLATE ROUTINE
00018A 45E0 C32E            00334   118            BAL   RTREG,WRTS    BRANCH TO WRITE ERASE
                                    119            WAIT  ECB=DECB1
00018E 4110 C552            00558   120+           LA    1,DECB1 LCAD PARAMETER REG 1
000192 4100 0001            00001   121+           LA    0,1(0,0) CCUNT OMITTEC,1 USEC
000196 0A01                         122+           SVC   1 LINK TO WAIT ROUTINE
000198 45E0 C416            0041C   123            BAL   RTREG,CKICCCMP
00019C D203 C616 C862 0061C 00868   124            MVC   COREFILE(4),ZERO1    ZEROS TO FIRST WORD CF COREFILE
0001A2 4160 C982            00988   125            LA    TBLREG,IECTRSCI  ASCII TO EBCIDIC TABLE ACD TO REG
0001A6 45E0 C4F0            004F6   126            BAL   RTREG,TRANSL  TRANSLATE COREFILE BACK TO EBCDIC
0001AA 47F0 C0F0            000F6   127            BC    15,CONTPCLL
0001AE 41B0 0081            00081   128 CONT       LA    LNGTHREG,129    LENGTH CF OPENMESSAGE TO REG
0001B2 4170 C592            00598   129            LA    AREAREG,OPNMESS    ACC OF OPEN MESS TO REG
0001B6 45E0 C32E            00334   130            BAL   RTREG,WRTS   BRANCH TC WRITE ERASE- OPEN MESS
                                    131            WAIT  ECB=DECB1
0001BA 4110 C552            00558   132+           LA    1,DECB1 LCAD PARAMETER REG 1
0001BE 4100 0001            00001   133+           LA    0,1(0,0) CCUNT OMITTEC,1 USEC
0001C2 0A01                         134+           SVC   1 LINK TO WAIT ROUTINE
0001C4 45E0 C416            0041C   135            BAL   RTREG,CKICCCMP
0001C8 47F0 C0F0            000F6   136            BC    15,CCNTPCLL  BRANCH TC CONTINUE PCLLING
0001CC D228 A004 7007 00004 00007   137 CHANGE     MVC   4(41,WRKREG3),7(AREAREG)  FIRST BUFF TO COREFILE
0001D2 5A80 C84E            00854   138            A     WRKREG1,BUFFL48     48 TO COUNT IN REG
0001D6 5080 C866            0086C   139            ST    WRKREG1,CCREFLN    STCRE COUNT IN ACCUMULATOR
0001DA D502 7001 C84A 00001 0085C   140            CLC   1(3,AREAREG),ZERC  SEE IF LAST BUFF OF MESS
0001E0 4780 C206            0020C   141            BE    A1          BRANCH IF LAST BUFFER OF MESS TRANSFERRED
0001E4 5AA0 C852            00858   142            A     WRKREG3,BUFFLTH5   ADJUST COREFILE ACD CONTAINED IN REG
0001E8 5877 0000            0000C   143 CHANGEA    L     AREAREG,0(AREAREG) ADC. OF NEXT BUFF TO AREAREG
0001EC D22B A000 7004 00000 00004   144            MVC   0(44,WRKREG3),4(AREAREG)  MOVE NEXT BUFFER TO COREFILE
0001F2 5A80 C85A            0086C   145            A     WRKREG1,BUFFLTH4   INCREASE COUNT OF CHAR MOVED BY 44
0001F6 5080 C866            0086C   146            ST    WRKREG1,CCREFLN  STCRE COUNT IN ACCUMULATCR
0001FA D502 7001 C84A 00001 00850   147            CLC   1(3,AREAREG),ZERC   SEE IF LAST BUFF
000200 4780 C206            0020C   148            BE    A1          BRANCH IF LAST BUFFER
000204 5AA0 C85A            00860   149            A     WRKREG3,BUFFLTH4   ACJUST ADC. CF COREFILE IN REG
000208 47F0 C1E2            001E8   150            BC    15,CHANGEA    BRANCH TC TRANSFER NEXT BUFFER TO COREFILE
00020C 5870 C83E            00844   151 A1         L     AREAREG,RELAREA    ACC. CF FIRST BUFFER TC REG
000210 45E0 C4A6            004AC   152            BAL   RTREG,RELBUFF     BRANCH TO RELEASE BUFFERS
000214 47F0 C1A8            001AE   153            BC    15,CCNT
000218 95F3 7006      00006         154 CONTCOMP   CLI   6(AREAREG),X'F3'
00021C 4780 C226            0022C   155            BE    DONE        IF 2260 OPERATCR ENTERED 3 BRANCH TO DONE
000220 5870 C83E            00844   156            L     AREAREG,RELAREA  TC HERE IF 4 ENTEREC
000224 45E0 C4A6            004AC   157            BAL   RTREG,RELBUFF   BRANCH TO RELEASE INPUT BUFFERS
000228 47F0 C1A8            001AE   158            BC    15,CCNT   BRANCH TO CONT THAT IS PUT OUT OPEN MESS
00022C 45E0 C48A            00490   159 DONE       BAL   RTREG,LERPRT   IF DONE PRINT OUT VALUES IN LREB
000230 45E0 C46E            00474   160            BAL   RTREG,CLCSE
000234 5810 4014            00014   161            L     REG1,DCBBUFCB   ADC. CF AREA TO BE FREED WITH FREEMAIN
                                    162            FREEMAIN R,LV=968,A=(REG1)
000238 4100 03C8            003C8   163+           LA    0,968(0,0) LCAD LENGTH
00023C 1811                         164+           LR    1,REG1 LCAD AREA ACCRESS
00023E 0A0A                         165+           SVC   10 ISSUE FREEMAIN SVC
                                    166            ABEND 12,DUMP
000240                              167+           CNOP  0,4
000240 47F0 C242            00248   168+           B     *+8 BRANCH ARCUND CCNSTANT
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01JAN67   5/01/67

000244 80                          169+          DC     AL1(128) DUMP/STEP CODE
000245 00000C                      170+          DC     AL3(12) COMPLETION CODE
000248 5810 C23E             00244 171+          L      1,*-4 LOAD CODES INTO REG 1
00024C 0A0D                        172+          SVC    13 LINK TO ABEND ROUTINE
                                   173           RETURN (14,12)
00024E 98EC D00C             0000C 174+          LM     14,12,12(13) RESTORE THE REGISTERS
000252 07FE                        175+          BR     14 RETURN
000254 4160 CA82             00A88 176 TEN50     LA     TBLREG,IECTSD50  LOAD ADD. OF TRANS. TABLE TO REG
000258 4130 C583             00589 177           LA     LISTREG,ADLST4    TRMLST  ADD. FOR ADDRESSING 105C TO REG
00025C 47F0 C0A4             000AA 178           BC     15,B1
000260 45E0 C3A2             003A8 179 B2        BAL    RTREG,WRTI     BRANCH TO WRITE INITIAL
                                   180           WAIT   ECB=DECB1
000264 4110 C552             00558 181+          LA     1,DECB1 LOAD PARAMETER REG 1
000268 4100 0001             00001 182+          LA     0,1(0,0) COUNT OMITTED,1 USED
00026C 0A01                        183+          SVC    1 LINK TO WAIT ROUTINE
00026E 45E0 C416             0041C 184           BAL    RTREG,CKICCOMP
000272 4130 C58B             00591 185           LA     LISTREG,POLLST2   ADD. OF 1050 POLL LIST TO REG
000276 4160 C882             00888 186           LA     TBLREG,IECTRF50  1050 TO EBCDIC
00027A 47F0 C0F8             000FE 187           BC     15,B3
00027E 4160 C882             00888 188 B4        LA     TBLREG,IECTRF50     1050 TO EBCDIC TABLE ADD
000282 47F0 C146             0014C 189           BC     15,B5
000286 95F2 7005       00005       190 B6        CLI    5(AREAREG),X'F2'  SEE WHAT 1050 OPERATOR REQUESTED
00028A 4740 C290             00296 191           BL     DSPLY1    BRANCH TO DISPLAY COREFILE
00028E 4780 C2E2             002E8 192           BE     CHANGE1   BRANCH TO CHANGE COREFILE
000292 4720 C226             0022C 193           BH     DONE
000296 D203 C616 C862 0061C 00868 194 DSPLY1    MVC    COREFILE(4),ZERO1  ZEROS TO FIRST WORD OF COREFILE
00029C 5870 C83E             00844 195           L      AREAREG,RELAREA   ADD. OF FIRST INPUT BUFF TO REG
0002A0 45E0 C4A6             004AC 196           BAL    RTREG,RELBUFF   BRANCH TO RELEASE BUFFERS
0002A4 4170 C616             0061C 197           LA     AREAREG,COREFILE   ADD. OF COREFILE TO REG
0002A8 5880 C866             0086C 198           L      LNGTHREG,COREFLN   LENGTH OF COREFILE TO REG
0002AC 4160 CA82             00A88 199           LA     TBLREG,IECTSD50  EBCDIC TO 1050 TABLE TO REG
0002B0 45E0 C4F0             004F6 200           BAL    RTREG,TRANSL
0002B4 4130 C583             00589 201           LA     LISTREG,ADLST4   ADD. OF 1050 ADDRESSING LIST TO REG
0002B8 45E0 C3A2             003A8 202           BAL    RTREG,WRTI
                                   203           WAIT   ECB=DECB1
0002BC 4110 C552             00558 204+          LA     1,DECB1 LOAD PARAMETER REG 1
0002C0 4100 0001             00001 205+          LA     0,1(0,0) COUNT OMITTED,1 USED
0002C4 0A01                        206+          SVC    1 LINK TO WAIT ROUTINE
0002C6 45E0 C416             0041C 207           BAL    RTREG,CKICCOMP
0002CA D203 C616 C862 0061C 00868 208           MVC    COREFILE(4),ZERO1
0002D0 4160 C882             00888 209           LA     TBLREG,IECTRF50   1050 TO EBCDIC TABLE ADD TO REG
0002D4 45E0 C4F0             004F6 210           BAL    RTREG,TRANSL BR TO TRANSLATE COREFILE BACK TO EBCDIC
0002D8 4130 C583             00589 211 CONT1     LA     LISTREG,ADLST4
0002DC 4180 0081             00081 212           LA     LNGTHREG,129
0002E0 4170 C592             00598 213           LA     AREAREG,OPNMESS  ADD OF OPEN MESSAGE TO REG
0002E4 47F0 C25A             0026C 214           BC     15,B2
0002E8 D229 A004 7006 00004 00006 215 CHANGE1   MVC    4(42,WRKREG3),6(AREAREG)  MOVE FIRST BUFF TO COREFILE
0002EE 5A80 C84E             00854 216           A      WRKREG1,BUFFL48
0002F2 5080 C866             0086C 217           ST     WRKREG1,COREFLN
0002F6 D502 7001 C84A 00001 00850 218           CLC    1(3,AREAREG),ZERO   SEE IF THIS WAS LAST BUFFER
0002FC 4780 C322             00328 219           BE     A2       BRANCH IF LAST BUFFER
000300 5AA0 C856             0085C 220           A      WRKREG3,BUFFLTH6   ADJUST COREFILE ADD. FOR NEXT MOVE
000304 5877 0000             0000C 221 CHANGE1A L      AREAREG,0(AREAREG)     ADD OF NEXT BUFFER TO REG
000308 D228 A000 7004 00000 00004 222           MVC    0(44,WRKREG3),4(AREAREG)  MOVE INPUT TO COREFILE
00030E 5A80 C85A             00860 223           A      WRKREG1,BUFFLTH4
000312 5080 C866             0086C 224           ST     WRKREG1,COREFLN
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              FC1JAN67   5/01/67

000316 D502 7001 C84A 00001 00850   225            CLC   1(3,AREAREG),ZERO
00031C 4780 C322            00328   226            BE    A2
000320 5AA0 C85A            00860   227            A     WRKREG3,BUFFLTH4    ADJUST COREFILE ADDRESS
000324 47F0 C2FE            00304   228            BC    15,CHANGE1A
000328 5870 C83E            00844   229 A2         L     AREAREG,RELAREA
00032C 45E0 C4A6            004AC   230            BAL   RTREG,RELBUFF
000330 47F0 C2D2            002D8   231            BC    15,CONT1
000334 50E0 C842            00848   232 WRTS       ST    RTREG,SAVERET
                                    233            WRITE (DECBREG),TSR,(DCBREG),(AREAREG),(LNGTHREG),(LISTREG),  X
                                                         (LINEREG),MF=E
000338 1812                         234+           LR    1,DECBREG LOAD DECB ADDRESS
00033A 9200 1004      00004         235+           MVI   4(1),0
00033E 928E 1005      00005         236+           MVI   5(1),142
000342 4080 1006            00006   237+           STH   LNGTHREG,6(0,1) STORE LENGTH
000346 5040 1008            00008   238+           ST    DCBREG,8(0,1) STORE DCB ADDRESS
00034A 5070 100C            0000C   239+           ST    AREAREG,12(0,1) STORE AREA ADDRESS
00034E 5030 1014            00014   240+           ST    LISTREG,20(0,1) STORE TERM. LIST ADDR.
000352 4050 1018            00018   241+           STH   LINEREG,24(0,1) STORE LINE NUMBER
000356 58F0 4030            00030   242+           L     15,48(0,DCBREG) LOAD RDWR ROUT. ADDR.
00035A 05EF                         243+           BALR  14,15
00035C 58E0 C842            00848   244            L     RTREG,SAVERET
000360 47FF C35E            00364   245            BC    15,CKIOST(15)    BRANCH TO SEE IF I/O STARTED OK.
000364 07FE                         246 CKIOST     BR    RTREG  RETURN CODE X'00'. OP STARTED OK.
000366 4700 0000            00000   247            NOP   0        RETURN CODE X'04'
00036A 4700 0000            00000   248            NOP   0        RETURN CODE X'08'
00036E 4700 0000            00000   249            NOP   0        RETURN CODE X'0C'
000372 4700 0000            00000   250            NOP   0        RETURN CODE X'14'
000376 4700 0000            00000   251            NOP   0        RETURN CODE X'10'
00037A 4700 0000            00000   252            NOP   0        RETURN CODE X'18'
00037E 4700 0000            00000   253            NOP   0        RETURN CODE X'1C'
000382 4700 0000            00000   254            NOP   0        RETURN CODE X'20'
                                    255            WTO   'I/O DID NOT START'
000386 0700                         256+           CNOP  0,4
000388 4510 C39C            003A2   257+           BAL   1,IHB0031A BRANCH AROUND MESSAGE
00038C 0015                         258+           DC    AL2(IHB0031-*) MESSAGE LENGTH
00038E 0000                         259+           DC    AL2(0)
000390 C961D640C4C9C440             260+           DC    C'I/O DID NOT START' MESSAGE
0003A1                              261+IHB0031    EQU   *
0003A2                              262+IHB0031A   DS    0H
0003A2 0A23                         263+           SVC   35 ISSUE SVC
0003A4 47F0 C226            0022C   264            BC    15,DONE
0003A8 50E0 C842            00848   265 WRTI       ST    RTREG,SAVERET
                                    266            WRITE (DECBREG),TI,(DCBREG),(AREAREG),(LNGTHREG),(LISTREG),   X
                                                         (LINEREG),MF=E
0003AC 1812                         267+           LR    1,DECBREG LOAD DECB ADDRESS
0003AE 9200 1004      00004         268+           MVI   4(1),0
0003B2 9202 1005      00005         269+           MVI   5(1),2
0003B6 4080 1006            00006   270+           STH   LNGTHREG,6(0,1) STORE LENGTH
0003BA 5040 1008            00008   271+           ST    DCBREG,8(0,1) STORE DCB ADDRESS
0003BE 5070 100C            0000C   272+           ST    AREAREG,12(0,1) STORE AREA ADDRESS
0003C2 5030 1014            00014   273+           ST    LISTREG,20(0,1) STORE TERM. LIST ADDR.
0003C6 4050 1018            00018   274+           STH   LINEREG,24(0,1) STORE LINE NUMBER
0003CA 58F0 4030            00030   275+           L     15,48(0,DCBREG) LOAD RDWR ROUT. ADDR.
0003CE 05EF                         276+           BALR  14,15
0003D0 58E0 C842            00848   277            L     RTREG,SAVERET
0003D4 47FF C35E            00364   278            BC    15,CKIOST(15)
```

```
  LOC  OBJECT CODE        ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                 FC1JAN67   5/01/67

0003D8 50E0 C842                00848  279 WRTA      ST     RTREG,SAVERET
                                       280           WRITE  (DECBREG),TA,(DCBREG),,,,,MF=E
0003DC 1812                            281+          LR     1,DECBREG LOAD DECB ADDRESS
0003DE 9208 1005          00005        282+          MVI    5(1),8
0003E2 5040 1008                00008  283+          ST     DCBREG,8(0,1) STORE DCB ADDRESS
0003E6 58F0 4030                00030  284+          L      15,48(0,DCBREG) LOAD RDWR ROUT. ADDR.
0003EA 05EF                            285+          BALR   14,15
0003EC 58E0 C842                00848  286           L      RTREG,SAVERET
0003F0 47FF C35E                00364  287           BC     15,CKICST(15)
0003F4 50E0 C842                00848  288 RDTI      ST     RTREG,SAVERET
                                       289           READ   (DECBREG),TI,(DCBREG),'S','S',(LISTREG),(LINEREG),MF=E
0003F8 1812                            290+          LR     1,DECBREG LOAD DECB ADDRESS
0003FA 9203 1004          00004        291+          MVI    4(1),3
0003FE 9201 1005          00005        292+          MVI    5(1),1
000402 5040 1008                00008  293+          ST     DCBREG,8(0,1) STORE DCB ADDRESS
000406 5030 1014                00014  294+          ST     LISTREG,20(0,1) STORE TERM. LIST ADDR.
00040A 4050 1018                00018  295+          STH    LINEREG,24(0,1) STORE LINE NUMBER
00040E 58F0 4030                00030  296+          L      15,48(0,DCBREG) LOAD RDWR ROUT. ADDR.
000412 05EF                            297+          BALR   14,15
000414 58E0 C842                00848  298           L      RTREG,SAVERET
000418 47FF C35E                00364  299           BC     15,CKICST(15)
00041C 50E0 C842                00848  300 CKIOCOMP  ST     RTREG,SAVERET
000420 957F 2000          00000        301           CLI    0(DECBREG),X'7F'
000424 4780 C44A                00450  302           BE     COMPOK
                                       303           WTO    'BAD COMPLETICN CODE CN I/O'
000428                                 304+          CNOP   0,4
000428 4510 C444                0044A  305+          BAL    1,IHB0038A BRANCH AROUND MESSAGE
00042C 001E                            306+          DC     AL2(IHB0038-*) MESSAGE LENGTH
00042E 0000                            307+          DC     AL2(0)
000430 C2C1C440C3D6D4D7                308+          DC     C'BAD CCMPLETICN CODE ON I/O' MESSAGE
00044A                                 309+IHB0038   EQU    *
00044A                                 310+IHB0038A  DS     0H
00044A 0A23                            311+          SVC    35 ISSUE SVC
00044C 47F0 C226                0022C  312           BC     15,DCNE
000450 58E0 C842                00848  313 COMPOK    L      RTREG,SAVERET
000454 07FE                            314           BR     RTREG
000456 50E0 C842                00848  315 OPEN      ST     RTREG,SAVERET
                                       316           OPEN   ((DCBREG))
00045A 0700                            317+          CNOP   0,4
00045C 4510 C45E                00464  318+          BAL    1,*+8 LCAD REG1 W/LIST ACCR.
000460 00000000                        319+          DC     A(0) CPT BYTE AND DCB ACCR.
000464 5041 0000                00000  320+          ST     DCBREG,0(1,0) STCRE INTO LIST
000468 9280 1000          00000        321+          MVI    0(1),128 MCVE IN OPTION BYTE
00046C 0A13                            322+          SVC    19 ISSUE CPEN SVC
00046E 58E0 C842                00848  323           L      RTREG,SAVERET
000472 07FE                            324           BR     RTREG
000474 50E0 C842                00848  325 CLOSE     ST     RTREG,SAVERET
                                       326           CLOSE  ((DCBREG))
000478                                 327+          CNOP   0,4
000478 4510 C47A                00480  328+          BAL    1,*+8 BRANCH ARCUND LIST
00047C 00000000                        329+          DC     A(0) CPTICN AND DCB ACCRESS
000480 5041 0000                00000  330+          ST     DCBREG,0(1,0) STCRE DCB ACCRESS
000484 9280 1000          00000        331+          MVI    0(1),128 MCVE IN CPTICN BYTE
000488 0A14                            332+          SVC    20 ISSUE CLCSE SVC
00048A 58E0 C842                00848  333           L      RTREG,SAVERET
00048E 07FE                            334           BR     RTREG
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                       FC1JAN67   5/01/67

000490 50E0 C842              00848   335 LERPRT   ST    RTREG,SAVERET
                                      336          LERPRT (DCBREG),(LINEREG)
000494                                337+         CNOP  0,4
000494 45F0 C496              0049C   338+         BAL   15,*+8
000498 80                             339+         DC    AL1(128)
000499 000000                         340+         DC    VL3(IECTLERP)
00049C 1814                           341+         LR    1,(DCBREG)
00049E 1805                           342+         LR    0,(LINEREG)
0004A0 58FF 0000              0000C   343+         L     15,0(15)
0004A4 05EF                           344+         BALR  14,15
0004A6 58E0 C842              00848   345          L     RTREG,SAVERET
0004AA 07FE                           346          BR    RTREG
0004AC 50E0 C842              00848   347 RELBUFF  ST    RTREG,SAVERET
                                      348          RELBUF (DCBREG),(AREAREG)
0004B0                                349+         DS    0H
0004B0 1814                           350+         LR    1,DCBREG
0004B2 4107 0000              00000   351+         LA    0,0(AREAREG)
0004B6 0A3A                           352+         SVC   58
0004B8 88F0 0018              00018   353+         SRL   15,24
0004BC 58E0 C842              00848   354          L     RTREG,SAVERET
0004C0 47FF C4BE              004C4   355          BC    15,CKRELB(15)
0004C4 07FE                           356 CKRELB   BR    RTREG  RETURN CODE X'00'-NORMAL
0004C6 4700 0000              00000   357          NOP   0                  X'04'
0004CA 4700 0000              00000   358          NOP   0                  X'08'
0004CE 4700 0000              00000   359          NOP   0                  X'0C'
0004D2 4700 0000              00000   360          NOP   0                  X'10'
                                      361          WTO   'RELBUF PROBLEMS'
0004D6 0700                           362+         CNOP  0,4
0004D8 4510 C4EA              004F0   363+         BAL   1,IHB0043A BRANCH AROUND MESSAGE
0004DC 0013                           364+         DC    AL2(IHB0043-*) MESSAGE LENGTH
0004DE 0000                           365+         DC    AL2(0)
0004E0 D9C5D3C2E4C640D7               366+         DC    C'RELBUF PROBLEMS' MESSAGE
0004EF                                367+IHB0043  EQU   *
0004F0                                368+IHB0043A DS    0H
0004F0 0A23                           369+         SVC   35 ISSUE SVC
0004F2 47F0 C226              0022C   370          BC    15,DONE
0004F6 50E0 C842              00848   371 TRANSL   ST    RTREG,SAVERET
                                      372          TRNSLATE (DCBREG),(TBLREG),(AREAREG),(LNGTHREG)
0004FA                                373+         DS    0H
0004FA 1808                           374+         LR    0,LNGTHREG
0004FC                                375+         CNOP  0,4
0004FC 4510 C506              005CC   376+         BAL   1,*+16
000500 00000000                       377+         DC    A(0)
000504 00000000                       378+         DC    A(0)
000508 00000000                       379+         DC    A(0)
00050C 5041 0000              00000   380+         ST    DCBREG,0(1)
000510 5061 0004              00004   381+         ST    TBLREG,4(1)
000514 5071 0008              00008   382+         ST    AREAREG,8(1)
000518 58F0 C51A              00520   383+         L     15,*+8
00051C 47F0 C51E              00524   384+         B     *+8
000520 00000000                       385+         DC    V(IECTTRNS)
000524 05EF                           386+         BALR  14,15
000526 58E0 C842              00848   387          L     RTREG,SAVERET
00052A 07FE                           388          BR    RTREG
000000                                389 REG0     EQU   0
000001                                390 REG1     EQU   1
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          FO1JAN67   5/01/67

000002                                    391 DECBREG    EQU    2          REG FOR ADD. OF DECB
000003                                    392 LISTREG    EQU    3          REG. FOR ADD. OF POLLING/ADDRESSING LIST
000004                                    393 DCBREG     EQU    4          REG. FOR ADD. OF DCB
000005                                    394 LINEREG    EQU    5          REG. FOR RELATIVE LINE NUMBER WITHIN DCB
000006                                    395 TBLREG     EQU    6          REG. FOR ADD. OF TRANSLATE TABLE
000007                                    396 AREAREG    EQU    7          REG. FOR ADD. OF I/O AREA
000008                                    397 WRKREG1    EQU    8
000009                                    398 WRKREG2    EQU    9
00000A                                    399 WRKREG3    EQU    10
00000B                                    400 LNGTHREG   EQU    11
00000C                                    401 BASEREG1   EQU    12
00000E                                    402 RTREG      EQU    14         REG. FOR RETURN ADD.
                                          403 DCB1       DCB    DSORG=CX,MACRF=(R,W),DDNAME=DS2260,BFTEK=D,LERB=ALERB,    X
                                                                ERROPT=ERWC
                                          404                   *,***  IHB063  DDNAME SHORT-PADDED TO 8 CHAR


                                          406+*                          DATA CONTROL BLOCK
                                          407+*
000518                                    408+           ORG    *-20 TO ELIMINATE UNUSED SPACE
000518                                    409+DCB1       DS     0F ORIGIN ON WORD BOUNDRY
00052C                                    410+           ORG    *+20 TO ORIGIN GENERATION

                                          412+*                          COMMON ACCESS METHOD INTERFACE

00052C 00                                 414+           DC     AL1(0) BUFNO
00052D 000001                             415+           DC     AL3(1) BUFCB
000530 0000                               416+           DC     AL2(0) BUFL
000532 1000                               417+           DC     BL2'0001000000000000' DSORG
000534 00000001                           418+           DC     A(1) IOBAD

                                          420+*                          FOUNDATION EXTENSION

000538 08                                 422+           DC     BL1'00001000' BFTEK,BFALN
                                          423                   *,*** IHB072 LERB REQUESTED-ERROPT=C ASSUMED
000539 0A                                 424+           DC     BL1'00001110' BTAM ERROPT CODE
00053A 0000                               425+           DC     H'0'
00053C 00                                 426+           DC     BL1'00000000' RECFM
00053D 000000                             427+           DC     AL3(0) EXLST

                                          429+*                          FOUNDATION BLOCK

000540 C4E2F2F2F6F04040                   431+           DC     CL8'DS2260' DDNAME
000548 02                                 432+           DC     BL1'00000010' OFLGS
000549 00                                 433+           DC     BL1'00000000' IFLG
00054A 2020                               434+           DC     BL2'0010000000100000' MACR
                                          435+*          BTAM INTERFACE
00054C 00000870                           436+           DC     A(ALERB) DCBLERB
000550 0000000000000000                   437+           DC     2F'0'
                                          438            WRITE DECB1,T,DCB1,,,,1,MF=L
000558                                    439+           DS     0F
000558 00000000                           440+DECB1      DC     A(0) EVENT CONTROL BLOCK
00055C 00                                 441+           DC     BL1'000'
00055D 00                                 442+           DC     AL1(0) TYPE FIELD
00055E 0000                               443+           DC     AL2(0) LENGTH
000560 00000518                           444+           DC     A(DCB1) DCB ADDRESS
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                           F01JAN67   5/01/67

000564 00000000                     445+         DC     A(0) AREA ADDRESS
000568 00000000                     446+         DC     A(0) ERROR INFC. FIELD ADDR
00056C 00000000                     447+         DC     A(0) TERMINAL LIST ADDRESS
000570 0001                         448+         DC     AL2(1) LINE NUMBER
000572 0000                         449+         DC     AL2(0) RESPONSE FIELD
000574 00                           450+         DC     AL1(0) TP-OP CODE
000575 00                           451+         DC     AL1(0) ERRCR STATUS
000576 0000                         452+         DC     AL2(0) CSW STATUS
000578 00000000                     453+         DC     AL4(0) CURRENT ADDR LIST PTR
00057C 00000000                     454+         DC     AL4(0) CURRENT ADDR PCLL PTR
                                    455 ADLST1   DFTRMLST OPENLST,A0A0  FOR ADDRESSING DS A0
000580                              456+ADLST1   DS     0C
000580 A0A0                         457+         DC     X'A0A0' TERMINAL LIST ENTRY
000582 81                           458+         DC     AL1(129) PROCEDURE FLAGS
                                    459 ADLST2   DFTRMLST OPENLST,A0A1  FOR ADDRESSING DS A1
000583                              460+ADLST2   DS     0C
000583 A0A1                         461+         DC     X'A0A1' TERMINAL LIST ENTRY
000585 81                           462+         DC     AL1(129) PROCEDURE FLAGS
                                    463 ADLST3   DFTRMLST OPENLST,A0A2  FOR ADDRESSING PRINTER
000586                              464+ADLST3   DS     0C
000586 A0A2                         465+         DC     X'A0A2' TERMINAL LIST ENTRY
000588 81                           466+         DC     AL1(129) PROCEDURE FLAGS
                                    467 ADLST4   DFTRMLST OPENLST,6202 FOR ADDRESSING 1050 K/B
000589                              468+ADLST4   DS     0C
000589 6202                         469+         DC     X'6202' TERMINAL LIST ENTRY
00058B 81                           470+         DC     AL1(129) PROCEDURE FLAGS
                                    471 POLLST1  DFTRMLST WRAPLST,A0FF  POLL LIST FOR DS. GENERAL POLL
00058C                              472+POLLST1  DS     0C
00058C A0FF                         473+         DC     X'A0FF' TERMINAL LIST ENTRY
00058E 21                           474+         DC     AL1(33) PROCEDURE FLAGS
00058F FFFD                         475+         DC     HL2'-3'
                                    476 POLLST2 DFTRMLST WRAPLST,620B POLL LIST FOR 1050 POLL
000591                              477+POLLST2  DS     0C
000591 620B                         478+         DC     X'620B' TERMINAL LIST ENTRY
000593 21                           479+         DC     AL1(33) PROCEDURE FLAGS
000594 FFFD                         480+         DC     HL2'-3'
000598                              481          DS     0F
000598 00000000                     482 OPNMESS  DC     F'0'
00059C C7D6D6C440D4D6D9             483          DC     C'GOOD MORNINGN'
0005A9 404040D7D3C5C1E2             484          DC     C'   PLEASE SELECT JOB DESIRED BYN'
0005C9 404040C5D5E3C5D9             485          DC     C'   ENTERING LINE NUMBERN'
0005E1 404040F14B40C4C9             486          DC     C'   1. DISPLAY COREFILEN'
0005F8 404040F24B40C3C8             487          DC     C'   2. CHANGE COREFILEN'
00060E 404040F34B40C4D6             488          DC     C'   3. DONEW'
00061C                              489          DS     0F
00061C 00000000                     490 COREFILE DC     1F'0'
000620 C7C5E340E6C9E3C8             491          DC     C'GET WITH IT.HOW ABOUT DOING SOMETHING YOURSELFW'
00064F 4040404040404040             492          DC     256C' '
00074F 4040404040404040             493          DC     173C' '
0007FC                              494 SAVE     DS     18F
000844                              495 RELAREA  DS     1F
000848                              496 SAVERET  DS     1F
00084C                              497 RPLYECB  DS     1F
000850 000000                       498 ZERO     DC     X'000000'
000853 00
000854 00000030                     499 BUFFL48  DC     1F'48'
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                   FO1JAN67   5/01/67

000858 0000002D                    500 BUFFLTH5 DC    1F'45'
00085C 0000002E                    501 BUFFLTH6 DC    1F'46'
000860 0000002C                    502 BUFFLTH4 DC    1F'44'
000864 40                          503 REPLY    DC    1C' '
000865 000000
000868 00000000                    504 ZERO1    DC    1F'0'
00086C 00000034                    505 COREFLN  DC    1F'52'
                                   506 ALERB    LERB  1,,,,
000870                             507+ALERB    DC    0F'0' ALIGN TABLE AND ATTACH NAME

                                   509+*        ACCUMULATORS
000870 00000000                    510+         DC    F'0' TRANSMISSIONS
000874 0000                        511+         DC    H'0' DATA CHECKS
000876 0000                        512+         DC    H'0' INTERVENTIONS
000878 0000                        513+         DC    H'0' NONTEXT TIMEOUTS
                                   514+*        COUNTERS
00087A 00                          515+         DC    X'0' TRANSMISSIONS
00087B 00                          516+         DC    X'0' DATA CHECKS
00087C 00                          517+         DC    X'0' INTERVENTIONS
00087D 00                          518+         DC    X'0' NONTEXT TIMEOUTS
                                   519+*        THRESHOLDS
00087E FF                          520+         DC    YL1(255) TRANSMISSIONS
00087F 0A                          521+         DC    YL1(10) DATA CHECKS
000880 05                          522+         DC    YL1(5) INTERVENTIONS
000881 05                          523+         DC    YL1(5) NONTEXT TIMEOUTS
                                   524+*        RESERVED
000882 0000                        525+         DC    XL2'0'

000888                             527          DS    0D
                                   528          ASMTRTAB RSCI,SSCI,RF50,SD50
000888                             529+IECTRF50 EQU   *
                                   530+*           0 1 2 3 4 5 6 7 8 9 A B C D E F
000888 3F40F13FF23F3FF3            531+         DC    X'3F40F13FF23F3FF3F43F3FF53FF6F73F' 0
000898 F83F3FF93FF07B3F            532+         DC    X'F83F3FF93FF07B3F3F34353F363F3F37' 1
0008A8 7C3F3F613FE2E33F            533+         DC    X'7C3F3F613FE2E33F3FE4E53FE63F3FE7' 2
0008B8 3FE8E93FE03F3F6B            534+         DC    X'3FE8E93FE03F3F6B243F3F253F26273F' 3
0008C8 603F3FD13FD2D33F            535+         DC    X'603F3FD13FD2D33F3FD4D53FD63F3FD7' 4
0008D8 3FD8D93FD03F3F5B            536+         DC    X'3FD8D93FD03F3F5B143F3F153F16173F' 5
0008E8 3F50C13FC23F3FC3            537+         DC    X'3F50C13FC23F3FC3C43F3FC53FC6C73F' 6
0008F8 C83F3FC93FC04B3F            538+         DC    X'C83F3FC93FC04B3F3F04053F063F3F07' 7
000908 3F407E3F4C3F3F5E            539+         DC    X'3F407E3F4C3F3F5E7A3F3F6C3F7D6E3F' 8
000918 5C3F3F4D3F5D7F3F            540+         DC    X'5C3F3F4D3F5D7F3F3F34353F363F3F3F' 9
000928 4A3F3F6F3FE2E33F            541+         DC    X'4A3F3F6F3FE2E33F3FE4E53FE63F3FE7' A
000938 3FE8E93F3F3F3F4F            542+         DC    X'3FE8E93F3F3F3F4F243F3F253F26273F' B
000948 6D3F3FD13FD2D33F            543+         DC    X'6D3F3FD13FD2D33F3FD4D53FD63F3FD7' C
000958 3FD8D93F3F3F3F5A            544+         DC    X'3FD8D93F3F3F3F5A143F3F153F16173F' D
000968 3F4EC13FC23F3FC3            545+         DC    X'3F4EC13FC23F3FC3C43F3FC53FC6C73F' E
000978 C83F3FC93F3F5F3F            546+         DC    X'C83F3FC93F3F5F3F3F04053F063F3F07' F
000988                             547+IECTRSCI EQU   *
                                   548+*           0 1 2 3 4 5 6 7 8 9 A B C D E F
000988 3F3F0203373F2E3F            549+         DC    X'3F3F0203373F2E3F3F3F153F3F3F3F3F' 0
000998 3F3F3F3F3F3D3F3F            550+         DC    X'3F3F3F3F3F3D3F3F183F3F3F3F3F3F3F' 1
0009A8 3F3F3F3F3F3F3F3F            551+         DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 2
0009B8 3F3F3F3F3F3F3F3F            552+         DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 3
0009C8 406A1A7B5B6C507D            553+         DC    X'406A1A7B5B6C507D4D5D5C4E6B604B61' 4
0009D8 F0F1F2F3F4F5F6F7            554+         DC    X'F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F' 5
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

0009E8 3F3F3F3F3F3F3F3F             555+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 6
0009F8 3F3F3F3F3F3F3F3F             556+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 7
000A08 3F3F3F3F3F3F3F3F             557+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 8
000A18 3F3F3F3F3F3F3F3F             558+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' 9
000A28 3FC1C2C3C4C5C6C7             559+          DC    X'3FC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6' A
000A38 D7D8D9E2E3E4E5E6             560+          DC    X'D7D8D9E2E3E4E5E6E7E8E93F3F3F3F6D' B
000A48 3F3F3F3F3F3F3F3F             561+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F0A3F3F' C
000A58 3F3F3F3F3F3F3F3F             562+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' D
000A68 7C3F3F3F3F3F3F3F             563+          DC    X'7C3F3F3F3F3F3F3F3F3F3F3F3F3F3F3F' E
000A78 3F3F3F3F3F3F3F3F             564+          DC    X'3F3F3F3F3F3F3F3F3F3F3F3F5F3F4F3F' F
000A88                              565+IECTSD5C EQU    *
                                    566+*                 0 1 2 3 4 5 6 7 8 9 A B C D E F
000A88 5E88163D797A7C7F             567+          DC    X'5E88163D797A7C7F88888888885B3858' 0
000A98 88888888585B5D5E             568+          DC    X'88888888585B5D5E8888888888888888' 1
000AA8 88888888383B3D3E             569+          DC    X'88888888383B3D3E8888888888888888' 2
000AB8 88885E88191A1C1F             570+          DC    X'88885E88191A1C1F8888888888888888' 3
000AC8 0188888888888888             571+          DC    X'01888888888888888888A0768493E1B7' 4
000AD8 6188888888888888             572+          DC    X'61888888888888888888D757909587F6' 5
000AE8 4023888888888888             573+          DC    X'4023888888888888888888378BC08EA3' 6
000AF8 8888888888888888             574+          DC    X'888888888888888888888816208C8296' 7
000B08 88626467686B6D6E             575+          DC    X'88626467686B6D6E7073888888888888' 8
000B18 88434546494A4C4F             576+          DC    X'88434546494A4C4F5152888888888888' 9
000B28 88882526292A2C2F             577+          DC    X'88882526292A2C2F3132888888888888' A
000B38 8888888888888888             578+          DC    X'88888888888888888888888888888888' B
000B48 75E2E4E7E8EBEDEE             579+          DC    X'75E2E4E7E8EBEDEEF0F3888888888888' C
000B58 54C3C5C6C9CACCCF             580+          DC    X'54C3C5C6C9CACCCFD1D2888888888888' D
000B68 3488A5A6A9AAACAF             581+          DC    X'3488A5A6A9AAACAFB1B2888888888888' E
000B78 15020407080B0D0E             582+          DC    X'15020407080B0D0E1013888888888888' F
000B88                              583+IECTSSCI EQU    *
                                    584+*                 0 1 2 3 4 5 6 7 8 9 A B C D E F
000B88 5A5A02035A5A5A5A             585+          DC    X'5A5A02035A5A5A5A5A5A5A5A5A0A5A5A' 0
000B98 5A5A5A5A5A0A5A5A             586+          DC    X'5A5A5A5A5A0A5A5A5A5A425A5A5A5A5A' 1
000BA8 5A5A5A5A5A5A035A             587+          DC    X'5A5A5A5A5A5A035A5A5A5A5A5A5A065A' 2
000BB8 5A5A5A5A5A5A5A04             588+          DC    X'5A5A5A5A5A5A5A045A5A5A5A5A155A5A' 3
000BC8 405A5A5A5A5A5A5A             589+          DC    X'405A5A5A5A5A5A5A5A5A5A4E5C484BFE' 4
000BD8 465A5A5A5A5A5A5A             590+          DC    X'465A5A5A5A5A5A5A5A5A5A444A495BFC' 5
000BE8 4D4F5A5A5A5A5A5A             591+          DC    X'4D4F5A5A5A5A5A5A5A5A414C45BF5E5F' 6
000BF8 5A5A5A5A5A5A5A5A             592+          DC    X'5A5A5A5A5A5A5A5A5A5A5A43E0475D5A' 7
000C08 5AA1A2A3A4A5A6A7             593+          DC    X'5AA1A2A3A4A5A6A7A8A95A5A5A5A5A5A' 8
000C18 5AAAABACADAEAFB0             594+          DC    X'5AAAABACADAEAFB0B1B25A5A5A5A5A5A' 9
000C28 5A5AB3B4B5B6B7B8             595+          DC    X'5A5AB3B4B5B6B7B8B9BA5A5A5A5A5A5A' A
000C38 5A5A5A5A5A5A5A5A             596+          DC    X'5A5A5A5A5A5A5A5A5A5A5A5A5A5A5A5A' B
000C48 5AA1A2A3A4A5A6A7             597+          DC    X'5AA1A2A3A4A5A6A7A8A95A5A5A5A5A5A' C
000C58 5AAAABACADAEAFB0             598+          DC    X'5AAAABACADAEAFB0B1B25A5A5A5A5A5A' D
000C68 5A5AB3B4B5B6B7B8             599+          DC    X'5A5AB3B4B5B6B7B8B9BA5A5A5A5A5A5A' E
000C78 5051525354555657             600+          DC    X'505152535455565758595A5A5A5A5A5A' F
                                    601           DCBD  DSORG=BX

                                    603+*                          DCB SYMBCLIC DEFINITICN FOR
                                    604+*                          BTAM LINE GRCUP

000000                              606+IHADCB    DSECT

                                    608+*                          BTAM LINE GRCUP INTERFACE
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                   FC1JAN67   5/01/67

000014                                     610+           ORG    IHADCB+20
000014                                     611+DCBBUFNO DS     0AL1
000014                                     612+DCBBUFCB DS     A
000018                                     613+DCBBUFL  DS     AL2
00001A                                     614+DCBDSORG DS     BL2
00001C                                     615+DCBIOBAD DS     A
000020                                     616+DCBBFTEK DS     BL1
000021                                     617+DCBERROP DS     AL1(0)
000022                                     618+         DS     AL2(0)
000024                                     619+DCBEXLST DS     A
00001C                                     620+         ORG    IHADCB+28
00001C                                     621+DCBDEVTP DS     BL1
000024                                     622+         ORG    IHADCB+36
000024                                     623+DCBEIOBX DS     BL1


                                           626+*                            FCUNDATICN BEFCRE OPEN

000028                                     628+         ORG    IHADCB+40
000028                                     629+DCBDDNAM DS     CL8
000030                                     630+DCBOFLGS DS     BL1
000031                                     631+DCBIFLG  DS     BL1
000032                                     632+DCBMACR  DS     BL2

                                           634+*                            FCUNDATICN AFTER OPEN

000028                                     636+         ORG    IHADCB+40
000028                                     637+DCBTIOT  DS     BL2
00002A                                     638+DCBMACRF DS     BL2
00002C                                     639+DCBIFLGS DS     0BL1
00002C                                     640+DCBDEBAD DS     A
000030                                     641+DCBREAD  DS     0A
000030                                     642+DCBWRITE DS     0A


000034                                     645+         ORG    IHADCB+52
000034                                     646+DCBLERB  DS     A(0)
000038                                     647+DCBONLTT DS     2F'0'
                                           648          IECTDECB
000000                                     649+IECTDECB DSECT  DECB DUMMY SECTICN
                                           650+*        -----------------------------------------
                                           651+*        +                                       +
                                           652+*      0 +      STANDARD EVENT CCNTRCL BLOCK     +
                                           653+*        +                                       +
                                           654+*        +---------------------------------------+
                                           655+*        +                  +                    +
                                           656+*      4 +  OPERATICN TYPE  +     AREA LENGTH    +
                                           657+*        +                  +                    +
                                           658+*        +---------------------------------------+
                                           659+*        + ON-LINE +                             +
                                           660+*      8 +TERMINAL +        ADDRESS CF DCB       +
                                           661+*        +  TEST   +                             +
                                           662+*        +---------------------------------------+
                                           663+*        +         +                             +
                                           664+*     12 +RESERVED +        ADDRESS CF AREA      +
                                           665+*        +         +
LOC  OBJECT CODE    ADDR1 ADDR2  STMT    SOURCE STATEMENT                                   F01JAN67   5/01/67

                                 666+*        +-----------------------------------------+
                                 667+*        +                    +                    +
                                 668+*    16  +    SENSE BYTES     +  RESIDUAL COUNT    +
                                 669+*        +    1     &    2    +                    +
                                 670+*        +-----------------------------------------+
                                 671+*        +          +                              +
                                 672+*    20  + COMMAND     ADDRESS OF TERMINAL LIST    +
                                 673+*        +   CODE   +                              +
                                 674+*        +-----------------------------------------+
                                 675+*        +          + RELATIVE+         +          +
                                 676+*    24  +  STATUS  +   LINE  + ADDRESS + VRC/LRC  +
                                 677+*        +  FLAGS   + NUMBER  + RESPONSE+ RESPONSE +
                                 678+*        +-----------------------------------------+
                                 679+*        +          +         +                    +
                                 680+*    28  +  TP-OP   + ERROR   +    CSW STATUS      +
                                 681+*        +   CODE   + STATUS  +                    +
                                 682+*        +-----------------------------------------+
                                 683+*        +          +                              +
                                 684+*    32  +RESERVED  +      ADDRESS OF CURRENT      +
                                 685+*        +          +       ADDRESSING ENTRY       +
                                 686+*        +-----------------------------------------+
                                 687+*        +          +                              +
                                 688+*    36  +RESERVED  +      ADDRESS OF CURRENT      +
                                 689+*        +          +        POLLING ENTRY         +
                                 690+*        -------------------------------------------
```

4.2.51

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                        F01JAN67   5/01/67

000000                              692+DECSDECB DS    1F STATUS FLAG + ACDRESS OF TFE TCB

000004                              694+DECTYPE  DS    1H CPERATICN TYPE

000006                              696+DECLNGTH DS    1H AREA LENGTH

000008                              698+DECONLTT DS    OCL1 RESERVED FOR CN-LINE TERMINAL TEST
000008                              699+DECDCBAD DS    1F ADDRESS CF DCB

00000C                              701+DECAREA  DS    1F ADDRESS OF AREA

000010                              703+DECSENS0 DS    1C 1ST SENSE BYTE

000011                              705+DECSENS1 DS    1C 2ND SENSE BYTE

000012                              707+DECCOUNT DS    1H RESIDUAL CCUNT

000014                              709+DECCMCOD DS    OCL1 CCMMAND CCDE
000014                              710+DECENTRY DS    1F ADDRESS CF TERMINAL LIST


000018                              713+DECFLAGS DS    1C STATUS FLAGS

000019                              715+DECRLN   DS    1C RELATIVE LINE NUMBER

00001A                              717+DECRESPN DS    1H RESPONSE FIELDS

00001C                              719+DECTPCOD DS    1C TP-OP CODE

00001D                              721+DECERRST DS    1C ERRCR STATUS

00001E                              723+DECCSWST DS    1H CSW STATUS

000020                              725+DECADRPT DS    1F ADDRESS CF CURRENT ACDRESSING ENTRY

000024                              727+DECPOLPT DS    1F ADDRESS CF CURRENT POLLING ENTRY

                                    729          END
          *** ERROR ***
                                    730+***
```

Printed in U.S.A.

R20-4105-0

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]